Enhancing Poker Agents with Hand History Statistics

Verbesseren von Poker Agenten durch Hand History Statistiken

Bachelor-Thesis von Andreas Li aus Aachen

23.05.2013

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Data and Knowledge Engineering

Enhancing Poker Agents with Hand History Statistics
Verbessern von Poker Agenten durch Hand History Statistiken

Vorgelegte Bachelor-Thesis von Andreas Li aus Aachen

1. Gutachten: Prof. Dr. Johannes Fürnkranz
2. Gutachten: Eneldo Loza Mencía, Dr. Frederik Janssen

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 23.05.2013

____

(Andreas Li)

Abstract

Poker is one of the more recent topics in the research field of artificial intelligences. The increasing popularity of online poker has led to development of new software which helps online poker players to gain an advantage over other players. The software displays dynamically calculated statistics for each player which can help to make the right decisions at the poker table and can also help the player in finding weaknesses in their own play. In this work we want to evaluate the viability of these statistics in terms of usage in poker agents with the means of mathematical statistics and machine learning. Finally we summarize our findings and propose how these statistics can be used in poker agents.

Contents

1 Introduction & Motivation

The game of poker is one of the central topics in current research of artificial intelligence. Due to the fact that some cards in play are concealed, poker is a game of imperfect information. As a result, determining the outcome of the game becomes impossible and to bypass this problem researchers and scientists employ different methods, such as utilizing game theoretic approaches including the Nash equilibrium, computational algorithms like the Monte Carlo methods, probability theory, mathematical statistics and even artificial neural networks.

In recent years, especially at the time of the poker boom , poker has gained a huge amount of popularity. As part of this increase, online poker has become very popular as well. Online poker is fundamentally different from conventional poker in several aspects. One of the main differences is speed: while playing live poker at a casino table a rough estimate of playing speed is about 20 to 40 hands per hour - when playing poker online at a single table this number is usually two to three times higher. Another difference being that the players are able to effectively play at multiple poker tables at the same time, resulting in an enormous increase of hands played. It is not unusual for players to play a five-digit number of hands per month. Playing at multiple tables however has its disadvantages, as keeping track of the behavior of each opponent becomes nearly impossible.

We can look at an example: a player who plays 30,000 hands per month has to play a thousand hands per day. At a rate of 80 hands per hour it would take roughly 12 hours to play a thousand hands at a single table. Now if we assume that the player plays at 6 tables with the same average speed he or she would only need a bit more than 2 hours to play a thousand hands. This shows how easy (or hard, depending on the view) it is to reach a high volume of hands, but a central question remains: how do people actually play at multiple tables without losing money, even though they are at a theoretical disadvantage?

We can assume that the average player will not be able to memorize the behavior and habits of certain players and play accordingly when playing six or more tables at the same time. Most so-called multitablers[1] use external third-party software to aid them in their play. The software tracks each players actions and calculates player-specific statistics (e.g. how often does the player raise or bet) accordingly. These stats are shown at the game table in real-time for each player as a so-called heads-up display. Of course an (appropriate) interpretation of these statistics is required and this task is up to the player him/herself.

In this thesis we are interested in how we can use these player-specific statistics to gain knowledge about player-type characteristics (*what defines a winning or losing player?*, etc. as one of many possible questions) and how we can use this knowledge to improve poker bots. Multitabling and the playstyle connected to it is, in some ways, very similar to the general playstyle of a poker agent which makes decisions based on the given information (e.g a mathematical model, probabilities or computable information in general). In our case, player-specific statistics are an approach to quantify the playstyle and behavior of the players - and in terms of poker agents a new type of statistical information and player measure which could be used to make real-time in-game decisions.

Our approach is to look at a very large database of players (and their statistics) in order to analyze the characteristics of these measurements. By doing this we hope to deduce a player-model based on them which can in turn be used in various ways within the research of poker agents.

Organization

This work is organized as followed: Chapter 2 gives a very brief introduction to the game of poker in general and shows related work. In Chapter 3 we describe the programs used to conduct this research. Chapter 4 introduces and explains all player-specific statistics and their formulas. In Chapter 5 we describe the process of analyzing our data set and evaluate our findings. Chapter 6 reviews the analysis and proposes usage scenarios in poker agents. Chapter 7 concludes.

[1] Online poker players who play at multiple tables at the same time

2 General Information

2.1 Introduction

Poker is a game with a long unclear history [16]. There are many theories about the origin of poker and there is no universally accepted truth about its origin. Some people claim that poker originated in China around 900 A.D., while other people think it originates from the French game ”Poque” or its German variant ”Pochern”. The history of poker in America is a bit more clear as the common consensus is that it originated in the Mississippi River area [48][32].

Regardless of its origin, today poker describes a family of games which is played by millions of people around the world. There are many variants of poker and some of them even use dices [1], but most of them are played with a standard 52-card deck which is also known as a French deck. It contains 52 cards with thirteen descending ranks (A, K, Q, J, T or 10, 9, 8, 7, 6, 5, 4, 3, 2) in 4 suits (♠♥♣♦).

The rules vary for each poker variant, but almost all of them involve betting (e.g. play money, chips or real money) and usually forced first bets (namely ante or blinds). Players bet that their hand will be highest ranked and try to win the pot. A player's hand is usually made up of five cards (not counting exotic variants of poker) - how a hand is structured exactly depends on the poker variant. Card poker variants can be categorized into three major categories: *Draw Poker*, *Stud Poker* and *Community Card Poker*. The focus in this thesis will be on **(No-Limit) Texas Hold'em**, a member of the community card poker family. Texas Hold'em is the most played poker variant [18] on PokerStars.com [20] and is widely considered the ”standard” variant of poker today [49].

We expect the reader of this document to know the basic rules of No-Limit Texas Hold'em, namely its betting system and gameplay. If that is not the case we suggest reading [19] and [51]. For very in-depth information on the betting system of poker in general we suggest reading [46], although it should not be neccessary in order to understand this work.

2.2 Rules of No-Limit Texas Hold'em

The rules can be broken down in three parts: card distribution, hand ranking and betting system.

Card Distribution

At the beginning of the game every player, starting from the small blind (SB) is dealt one card, with the dealer (button) (BTN) being the last. This is repeated until every player on the table has *two cards*, which are called **hole cards**. In addition to that all players at the table share *five* **community cards** which are open to see for everyone. All cards are dealt on so-called **streets**: *preflop, flop, turn* and *river*. The hole cards are dealt preflop, while the community cards are dealt in the following three streets. On the flop three cards are dealt, while on turn and river only one card is dealt each. As an example:

Hand Ranking

Rank	Example
Royal Flush	A♠K♠Q♠J♠T♠
Straight Flush	T♥9♥8♥7♥6♥
Four of a Kind	Q♥Q♠Q♦Q♣5♥
Full House	3♣3♥3♦K♦K♣
Flush	K♦9♦7♦5♦2♦
Straight	8♦7♣6♦5♠4♥
Three of a Kind	Q♣Q♥Q♠7♣4♥
Two Pair	J♦J♠9♥9♦4♣
One Pair	K♥K♣9♦8♠4♥
High Card	A♠Q♣8♦5♣3♦

Table 2.1: Hand Rankings

The hand rankings are mostly the same for all poker variants. Table 2.1 shows the hand ranks and an example for each of them. The *Royal Flush* is the strongest hand, while *High Card* is the weakest. Starting from the top: a **Royal Flush** means that the is the strongest hand and denotes the five highest cards in sequence and in the same suit. The **Straight Flush** is the second strongest type of hand and is basically the same as the Royal Flush, but with the values do not have to be the five highest (they still need to be in the same suit and sequential though). A *Four of a Kind* is, as the name implies, a hand where four of the five cards have the same value. The **Full House** is a hand composed of three *and* two matching cards with the same value respectively. Below the Full House is the **Flush**, which is composed of five non-sequential cards of the same suit. A **Straight** is a hand of five sequential cards with at least two different suits (otherwise it would be a Straight Flush). **Three of a Kind** (also called *trips* or *set*, if two of the three cards are held as hole cards) is a hand with three cards of the same rank that contains three different ranks in total. A **Two Pair** is a hand which consists of two pairs (of two ranks) plus a card with a another hand rank. **One Pair** is a hand that contains two cards of the same rank and three additional cards which are not of this rank and all different from each other. A *High Card* is the last type of hand which does not fulfil any of the conditions described above.
When comparing two hands of the same hand rank (e.g One Pair), the hand which consists of higher ranked cards (for example a pair of aces vs. a pair of 8s) wins.

Betting System

The betting system of No-Limit Texas Hold'em (NLHE) is somewhat complex. First of all there are two *forced bets*, the so-called blinds, namely the small blind (SB) and the big blind (BB). The blinds are posted before any cards are dealt and since the players obviously do not know their cards at that point of game these bets are called *blinds*. In live poker some people might opt to *straddle* which is the "third" blind. The blinds decide the minimum bet amount of the game. Speaking in relative terms, the big blind is almost always twice the amount of the big blind, so if the small blind is $1, the big blind is $2. The smallest possible amount a player can *bet* is the big blind, which leads us to the actual actions the players can do. The money is collected in a pot, which upon the end of the game, is won by the player with the best hand.

The player have to act on each street, while the small blind is first (the only exception being the preflop street, where the player after the big blinds starts). The *dealer* (or also called *button*) is the last player to act. There are certain actions a player can do when its his/her turn.

Bet: *Betting* is the simple act of placing a bet, on condition that no one else has bet anything yet. As previously mentioned, the minimum amount of a bet must be of the same size as the big blind (with the exception of the player not having enough money, where he/she bet all of the remaining money). In no-limit Texas Hold'em the amount can be as high as the amount of money the player has at the table.

Raise: The act of increasing a previously placed bet (or raise), is called *raising*. The minimal amount is of a raise is the size of the last bet or, in case of a raise, the amount of the last raise. This might be ambiguous, so let us look at an example. Say player A bet $20 and player B raises by $50 to a total amount of $70. A common misconception is that player C can only raise by $70 to a total amount of $140, but this is not the case. The actual minimum amount is a raise of $50 (making the pot $120), since the last raise was in fact *by $50* and *not by $70*. The only exception to the rule is that if the player does not have enough chips[1] to double the previous bet or raise. In that case the player can go *all-in*, betting all of his/her remaining money.

[1] The "currency" of a poker game. Can be real-money, play-money or arbitrarily defined currency (e.g. tournament chips)

Call: When a player *calls*, he/she matches the last raise or bet. The next street is only dealt if all remaining players have called. In case no opponents call the bet (or raise), the last remaining player wins the pot.

Check: A player can *check* if no bets have been placed yet. Checking is basically the same as calling an amount of 0. In case a bet has been made, the players can no longer check. A special case is the preflop street, where the big blind may check if no one has raised beyond the amount of the big blind.

Fold: *Folding* is the act of discard the hole cards and forfeit the round. When a player folds he/she gives up the opportunity to win the pot and no further bets, calls or raises are required (nor possible).

Poker Jargon

In addition to the more or less complex rules, the poker community has developed an enormous amount of technical terms which some people might not be familiar with. Some of these terms are required in order to understand everything, but we will give an appropriate explanation of the terms when we first use them. The glossary at the end of the document will also contain all used poker terms including a short explanation for a fast lookup should that be required.

2.3 Related Work

The topic of this work is relatively undiscovered which is why we only found two works related to our research.

Anderson et al. [2] describe their approach on data mining a database of 170,000 hands of different players and 10,000 hands from a single player from PartyPoker.com [15]. They used PokerTracker to import the hand histories and create the database where they ran custom SQL[2] queries on to analyze the data. Two types of analysis were done. The first one was to find metrics (and their range) that define winners, the second part of the project was to develop an algorithm to estimate the hand held by the player when he/she has acted in a certain way.

They have found and defined two winning factors: the player's VPIP (Voluntarily Put Money in Pot %) and AF (Aggression Factor), in-depth explanations of these terms can be found in chapter 4, and suggested a range of values for each of them which would characterize a winner. Unfortunately their other approach in trying to determine the hole cards of certain players was not as successful. Their algorithm compared the players action on the flop with the actions of other players where the hole cards were known, but this did not yield good results.

The second related work was done by D. Bragonier [4]. In the work he analyzed the hand histories of his roommate which was made up of roughly 1.9 million hands played in between March 2006 and January 2010. Bragonier used different approaches to analyze the data including univariate, (logistic) regression, outlier analysis and ANOVA[3]. He looked at the data from several "viewpoints", mainly by grouping the data into specific categories like number of hands, (table) sessions and stakes. His results of the analysis were mixed, but can be summarized in that there were too many factors to be considered.

[2] Structured Query Language, a programming language specialized for managing relational databases
[3] Analysis of variance, see `http://en.wikipedia.org/wiki/Analysis_of_variance` for details

3 Programs

Due to the rise in popularity of online poker in the past decade(s) [50] a large number of new poker players have emerged. While most of them might be casual players, a few of them also play poker for profit and/or for a living. This usually requires playing a large amount of hands (i.e. 10,000 and more per month) and online poker has made this possible by enabling a new ”way” of playing poker, namely multitabling which we already mentioned.

Multitabling, as the name implies, describes the process of playing at multiple poker tables at the same time. The effective number of tables depends on the player's skill but can range from two to a few dozens. A small anecdote: the Guiness World Record for playing the most single table Sit & Goes[1] played in one hour is 62 [10], set by the French professional poker player Bertrand Grospellier[3]. As a consequence players are able to play a large amount of hands in a very short time, this however, does not come without downsides. As the number of tables (and so the number of players you are up against) increases, it becomes increasingly hard or even impossible to keep track of how your opponents play. Poker is a game of exploiting your opponents flaws while minimizing your own, so not being able to keep track of your opponents means playing at a (huge) disadvantage. Multitabling usually results in just ”playing your cards” while disregarding the opponents style.

Figure 3.1: An example of a Heads-Up Display in PokerTracker 4
Source: http://www.pokertracker.com/products/PT4/

In order to reduce the disadvantage of multitabling many players have begun to use external poker tools which show a heads-up display (HUD) for each player at each table. An example of a HUD can be seen in the picture above.

[1] A (online) poker tournament format which does not start at specific times, but rather when enough players have joined

Strictly speaking this does not only help multitabling players but generally all players who want more information on other players. There are two major commercial programs used by todays poker community namely Hold'em Manager (HM or HEM) [9] and PokerTracker (PT) [23]. In the next section we are going to take a look at their feature sets.

There is also a wide array of other poker-related tools, one of them being PokerStrategy.com's Equilab which we will also decribe briefly.

3.1 Tracking Software

Both Hold'em Manager 1 and 2 [11, 12] and PokerTracker 4 [25] have very similar feature sets. Their main feature are their respective heads-up displays as previously mentioned. PT as well as HM have diverse customization options. These options range from general stat selection to color-coding stats, up to customization of pop-ups which display detailed stats per street. The so called *Heat Maps*, which display the winrate[2] for each hand type, are also a common feature of the software.

The main defining feature for both programs resolves around hand histories, which are text-based log files of the events that happen on the poker table. A replayer, or hand history viewer, is a common feature found in both programs. However, the most powerful feature is the automatic extraction of information from these hand history files.

The information ranges from reports with various kinds of general information (overall reports, session reports and winnings summaries to name the most generic examples) to detailed opponent statistics. In HM1 and PT4 the user can also create their own custom reports.

All mentioned features are available for regular cash game poker as well as tournaments. We have only described some of the features that the software have to offer. For a full feature list of HM and PT please refer to [9] and [23].

Both programs require a local database to be set up in order to mine data from the hand histories. HM and PT use a local PostgreSQL database to store data, but they have vastly different approaches in doing so. PT stores all of the hand history data in the database, while HM uses a hybrid method, saving some of the data as normal files on the local file system. Due to these different approaches in saving the data, both programs have different strengths and weaknesses. Two of our main factors in deciding which software to use are (import) speed and features of utilizing the data. In the next sections we compare these factors for each program to each other.

3.1.1 Hold'em Manager

Figure 3.2: Logo of Hold'em Manager 2

We used both Hold'em Manager 1 [11] and 2 [12] with the newest version of PostgreSQL [29] (9.2 at the time of writing). Hold'em Manager 2 was the fastest in terms of import speed of our hand histories, going up to 400 hands per second, averaging at about 250 hands per second for our complete database. This is mainly due to the special storing structure that HM2 employs. It saves roughly 60% of the data in the PostgreSQL database while the remaining data (the actual hand histories of each player) is stored as highly compressed text files on the local file system. Simple file operations are generally faster than any kind of SQL queries which made HM2 the overall fastest software out of the three. This however resulted in a very restrictive usability from a third-party point of view. At the time of writing, HM2 lacks many features that PT4 or even its predecessor HM1 has to offer – the most interesting feature being custom reports. A picture of the *Opponents* tab is shown in Figure 3.3. Customization of the shown columns was not possible. The lack of customization options would not have been a problem if there was an API or documentation of how HM2 structures its database (so that running custom queries becomes possible), but unfortunately neither of them have been released as of yet. We have contacted the Hold'em Manager support and have received access to experimental beta builds of the software where a proprietary query language named Hold'em Manager Query Language (HMQL) was starting to get implemented. The language is developed with the intention of running custom queries on the proprietary

[2] Percentage of times won

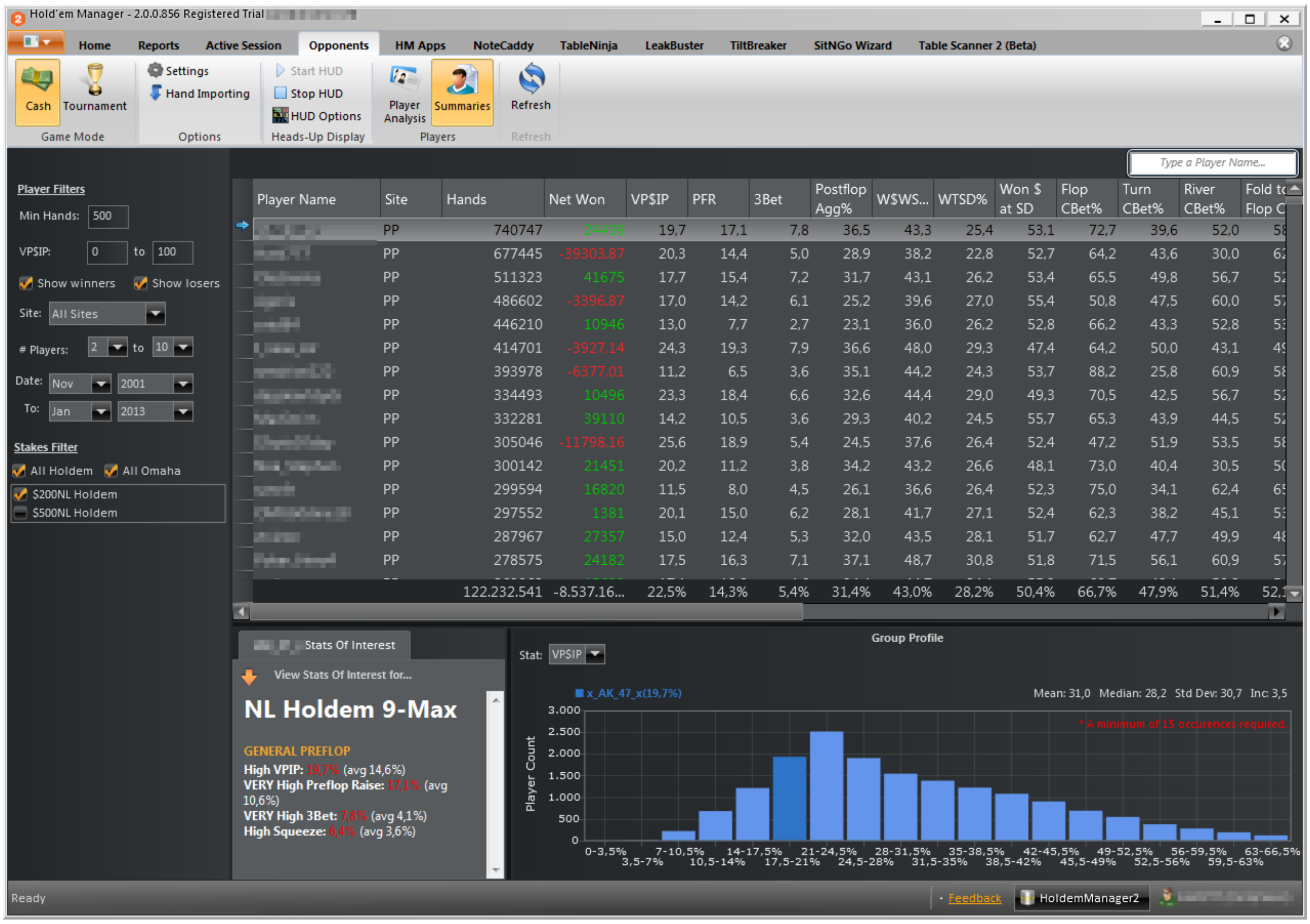

Figure 3.3: Hold'em Manager 2 showing the Opponents tab

database/storing format of HM2. The progress of implementation was still at the very beginning, making the usability of HMQL very limited and prone to bugs and crashes. We have also asked the Hold'em Manager support about the compressed hand history format but have not received an answer that helped in understanding or utilizing it.

Hold'em Manager 1 uses a different database structure and saves all data in the PostgreSQL database. This makes it slower than its successor but increases the usability of the database itself. The decrease in import speed is quite significant. We tried to import several batches of about 2 million hands, but only had peak speeds of about 150 hands per second. The average was roughly 40 to 50 hands per second which is extremely slow. Importing our database would have taken a few days at least.

3.1.2 PokerTracker

Figure 3.4: Logo of PokerTracker 4

PokerTracker 4 [25] utilizes a similar database structure as Hold'em Manager 1 but holds much more atomic statistics (such as the raw counts/numbers for each stat scenario, e.g. how many times the player had an opportunity to bet) which are more useful when trying to create custom stats. A documentation of the PT4 database schema was not available at the point of writing, but the database schema of PT3 is very similar and can be found at [24]. Unfortunately PokerTracker 4 has the huge drawback of being the slowest in terms of hand import speeds. In our initial import tests we only managed to peak at 120 hands per second with the average being only 30 to 40 hands per second, which is even slower than HM1.

With extensive optimization we managed to speed up this process significantly. This is described in detail in Section 5.1. Figure 3.5 shows an image of PokerTracker's equivalent of HM2's *Opponents* tab. As it can be seen on the right, the report is fully customizable and even supports self-created statistics.

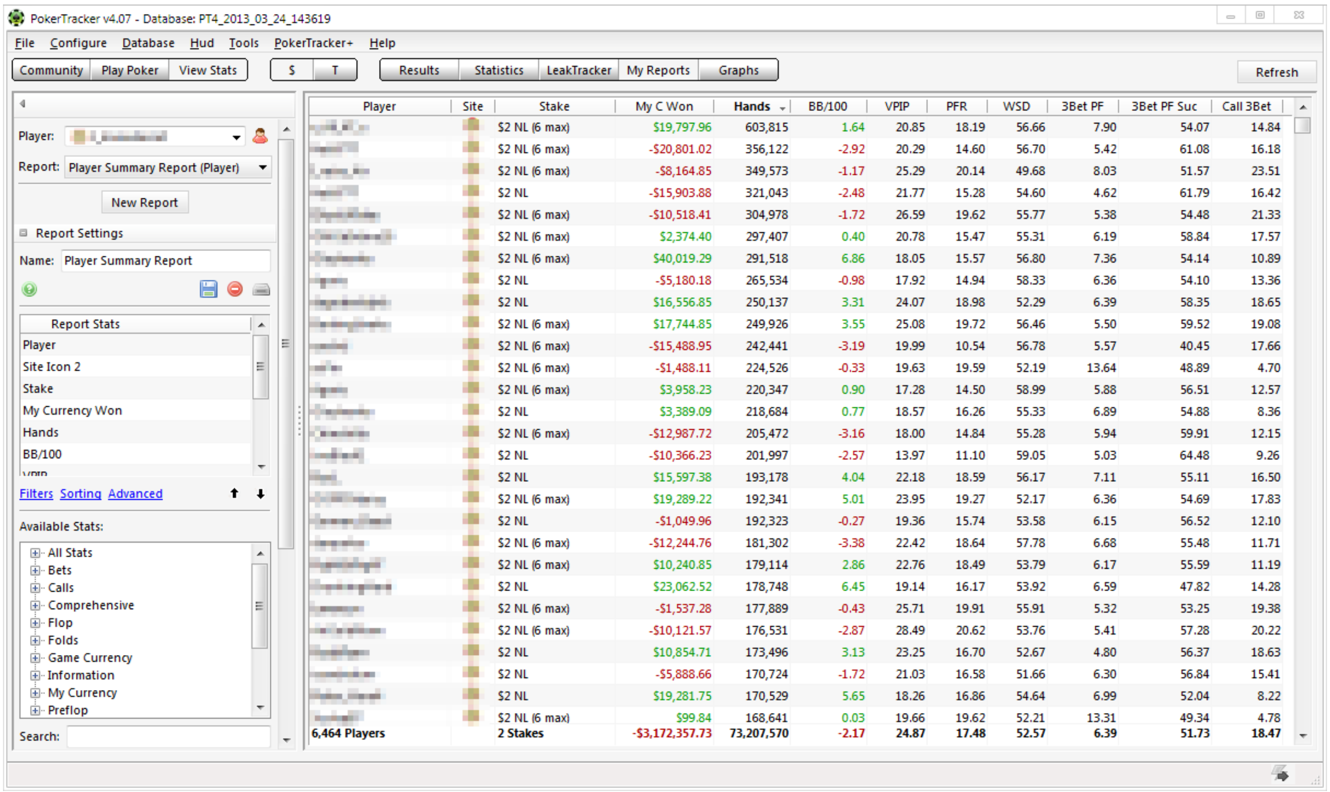

Figure 3.5: An example Player Summary Report in PokerTracker 4

While using Hold'em Manager 1 (because of its superior database usability in comparison to HM2) would have been an option, we ultimately decided to use PokerTracker 4 to create our database. The main drawback of HM1 is, that it is not a current-generation product, unlike PT4. The differences of customizable features between the two is negligible, but PT4 still has the advantage of being actively supported by the developer (and a large community).

3.2 Additional Software

PokerStrategy.com Equilab

Equilab [21] is a free poker software developed by PokerStrategy.com [22] and is used to calculate situation-specific (pot) equities by using either complete enumeration or Monte Carlo simulation. An equity of a hand is the average percentage of the pot the hand should have won based on the current situation. For example if we hold A♦K♦on the button against two random hands in the small blind[3] (SB) and big blind[4] (BB) we have a pot equity of about 50.73% - if the pot is 100$, we can expect to win 50.73$ on average in this particular situation. Another feature included is the ability to visualize hand ranges with an easy-to-use graphical user interface, which was our most used feature - a picture of the visualization is shown below. The *s* and *o* behind the hand indicates *suited* and *off-suited*, which means that the hand is of the same color or not.

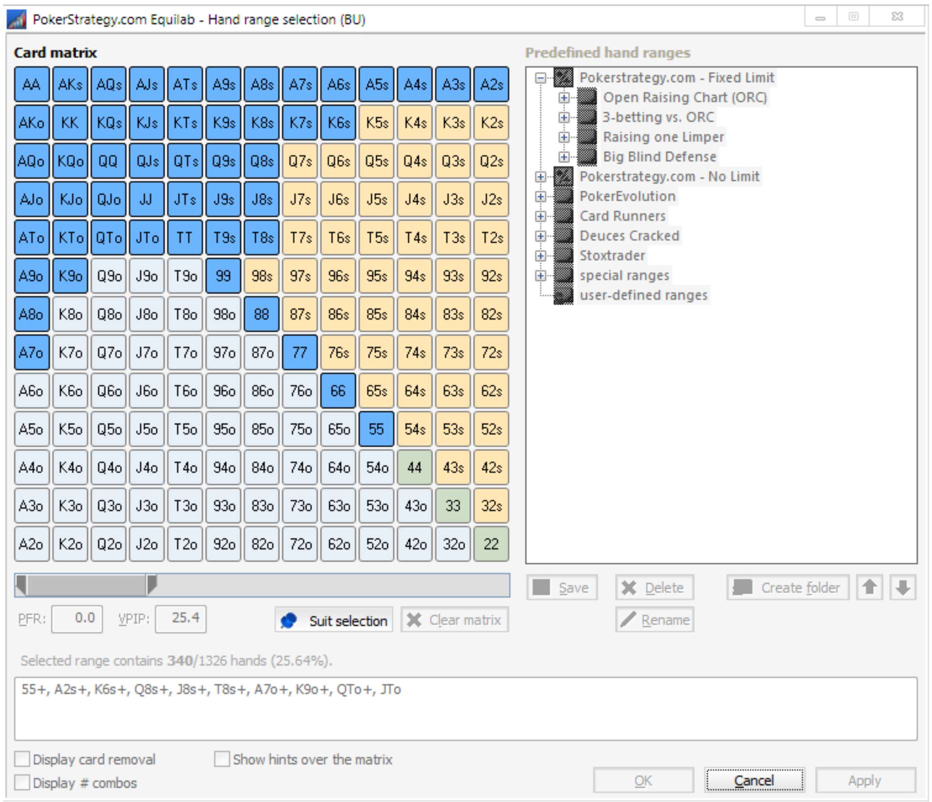

Figure 3.6: Visualization of a hand range in Equilab

Weka

Weka (Waikato Environment for Knowledge Analysis) [36] is a data mining software written in Java[5]. It is a collection of machine learning algorithms and is used for data mining and analysis. Weka also features visualization tools and a easy-to-use graphical user interface. We used Weka to plot histograms for our data, as well as to run classification and clustering algorithms on our data.

[3] the smaller one of the two forced bets made in the beginning of each played hand
[4] the larger one of the two forced bets, usually twice the size of the SB
[5] An object-oriented programming language

4 Player Statistics

In this chapter we introduce and describe a subset of player statistics (also abbreviated *stats*) used by the poker tracking software. We also analyze how they can be interpreted and which factors can play a role when evaluating them. We will discuss the most important stats thoroughly, while for some stats only their formula and a very brief explanation is given. Many formulas contain the *opportunity to do X* and *times X* which simply means that the player in question would have been able to do *X* and for the latter did *X*. As an example an opportunity to raise means that the player could have raised in the situation. Due to the rules of poker these opportunities usually overlap which means that opportunities to raise are also usually opportunities to fold or call. Information and knowledge has been compiled from several sources: [26, 5, 28, 7, 31, 33, 27, 13, 30, 34, 17]. We decided to reference everything here in order to increase the readability of the next sections.

4.1 Hand Histories in General

Before we go into the description of the statistics, we will briefly introduce *hand histories*, since they are an essential part of actually gathering them.

Hand histories serve various purposes and are generated by online poker networks. They are used to log player actions during games in case players encounter problems concerning the online game. Most poker site providers also enable the player to export their own hand histories where they can only see their own cards and the cards known to them (i.e. shown at showdown). These are most commonly used to review one's own game (e.g. posting them in forums for feedback on gameplay) or to be used in poker tracking software like PokerTracker or Hold'em Manager.

As hinted, there are different types of hand histories which can be grouped into three main categories: internal hand histories (used by the poker sites with all cards, including hole cards which did not go to the showdown, known and usually not available to the public), player specific (only own cards and shown cards are known) and data mined hand histories (mined from external providers where only the cards shown at showdown are known).

The exact format for a hand history differs from site so site, but all contain information about:

- Game ID
- Table ID/Name and type (e.g, fast table[1], zoom table[2] etc.)
- Date, time and timezone
- Limit and blinds (e.g. $1000 USD NL or NL $5/$10 USD)
- Type of money (e.g. real money, tournament chips or play money) and currency
- Number of players (2 up to 10 in Hold'em games)
- Location of the button
- All player names, positions and current stack sizes at the beginning of the hand (sometimes at the end of the hand as well)
- Hole cards if they are known
- Player actions (e.g. posting blinds/ante, fold, raise X, check, bet Y etc.)
- Cards shown at showdown and how much money won by which player

[1] A faster table where the players have less time to act
[2] A very fast poker format where, after folding the player is automatically transfered to another table where he can immediately play the next hand

This is an example hand history of a $0.25 buy-in FR tournament from PokerStars taken from [47] - it is a player-specific hand history from the view of the player on seat 4. We have anonymized the player names replacing them with variables:

```
PokerStars Game #27738602010: Tournament #160417233, $0.25+$0.00 Hold'em No Limit - Level XV (250/500) - 2008/03/01 15:36:40 ET
Table '160417233 3' 9-max Seat #8 is the button
Seat 1: A (9182 in chips)
Seat 2: B (25711 in chips) is sitting out
Seat 3: C (21475 in chips)
Seat 4: D (60940 in chips)
Seat 5: E (18044 in chips)
Seat 6: F (8338 in chips)
Seat 7: G (8353 in chips)
Seat 8: H (4404 in chips)
Seat 9: I (23553 in chips)
A: posts the ante 60
B: posts the ante 60
C: posts the ante 60
D: posts the ante 60
E: posts the ante 60
F: posts the ante 60
G: posts the ante 60
H: posts the ante 60
I: posts the ante 60
I: posts small blind 250
A: posts big blind 500
*** HOLE CARDS ***
Dealt to D [Qd Qc]
B: folds
C: folds
D: raises 2000 to 2500
E: raises 15484 to 17984 and is all-in
F: folds
G: calls 8293 and is all-in
H: folds
I: calls 17734
A: folds
D: raises 15484 to 33468
I: calls 5509 and is all-in
Uncalled bet (9975) returned to D
*** FLOP *** [2d 2c 3c]
*** TURN *** [2d 2c 3c] [8h]
*** RIVER *** [2d 2c 3c 8h] [4d]
*** SHOW DOWN ***
I: shows [9s 9h] (two pair, Nines and Deuces)
D: shows [Qd Qc] (two pair, Queens and Deuces)
B has returned
D collected 11018 from side pot-2
E: shows [5d 5h] (two pair, Fives and Deuces)
D collected 29073 from side pot-1
G: shows [Kh As] (a pair of Deuces)
D collected 34212 from main pot
*** SUMMARY ***
Total pot 74303 Main pot 34212. Side pot-1 29073. Side pot-2 11018. | Rake 0
Board [2d 2c 3c 8h 4d]
Seat 1: A (big blind) folded before Flop
Seat 2: B folded before Flop (didn't bet)
Seat 3: C folded before Flop (didn't bet)
Seat 4: D showed [Qd Qc] and won (74303) with two pair, Queens and Deuces
Seat 5: E showed [5d 5h] and lost with two pair, Fives and Deuces
Seat 6: F folded before Flop (didn't bet)
Seat 7: G showed [Kh As] and lost with a pair of Deuces
Seat 8: H (button) folded before Flop (didn't bet)
Seat 9: I (small blind) showed [9s 9h] and lost with two pair, Nines and Deuces
```

As we can see from the example, every action (and situation) on the table is logged, which makes it possible to derive certain information out of the hand histories - commonly called *player statistics* or simply *stats*. In the following sections we will describe these stats and how they are actually calculated.

4.2 Hands Played

Hands played (or *Hands*) is simply the number of hands played by the player in question and the most important of all stats. It serves as the concrete number for the sample size. To get a very rough picture of the player's behavior a minimum sample size of around 50 hands is required and even at this number the credibility of the stats will not be particularly good. Most stats show good convergence at a sample size of roughly 100 to 200 hands which is recommended and some might even require more (such as the statistics related to raising, where convergence starts at 200+). Estimating a good minimum number of hands is difficult, as it hugely depends on the occurence of certain events. For the following statistics we will only give very rough estimates for good sample sizes, if at all.

Generally speaking, a greater sample size means a better accuracy in terms of determining the play style of the player, but a large sample size is not without drawbacks. Poker is a game where the player skill usually increases and evolves the more experience the player gets and therefore time as well as timespan are factors to be considered when checking the quality of the sample. Suppose we have a very large sample size of a few million hands for a player over a timespan of five years. While the sheer number of hands will give quite accurate stats over his or her average play style in the last five years, it might not reflect the current playstyle (e.g. from the past year or so) very well which is much more interesting when trying to exploit this player based on his/her stats.

4.3 Amount Won

This value just shows the raw amount of money won or lost during all played hands and is used to calculate the more meaningful BB/100 stat described in Section 4.8, which includes the limit where the hands were played on. The raw amount of money won is an interesting statistic, but overall it does not hold much (useful) informational value. It does not take into account how many hands were played or what limit was played. For example a NLHE1000[3] player ($5/$10 blinds) who has played 1 hand and won $100 compared to an NLHE100 ($5/$10) player who has over a million hands and more or less breaks even with $100 won - there is no comparison to be made, because their total amount won will be the same, but the limit and number of hands is vastly different.

4.4 Voluntarily Put Money in Pot

Voluntarily Put Money in Pot % or abbreviated VPIP or VP$IP is the first stat of the so-called *holy trinity*[4] and is calculated as follows:

$$\text{VPIP} = \frac{\text{Total Times Voluntarily Put Money in the Pot}}{\text{Total Hands Played - Total Walks}} \times 100$$

Note that the SB and BB postings are not voluntary unless the respective players call/raise in case of the SB or raise, in case of the player posting the BB. Walks[5] are substracted because if everyone folds when the player sits on the BB, it technically counts as a hand played, but our player never had the chance to actually put money into the pot. The VPIP depends on a lot of different factors.

As positional play is one of the most important factors of poker this fact is also reflected in the VPIP, which can be split into values for each respective position (i.e. EP[6], MP[7], LP[8], SB, BB, CO[9], BTN[10] VPIP). As a rule of thumb the VPIP in later positions should be higher than in early positions. This is because the range of playable hands becomes greater the later your own turn is, therefore it becomes easier to bet or raise due to positional advantage [35]. We can use this rule as an indicator for an opponent's strength: if the VPIP value does not follow the rule, for example if it is constant throughout all positions, the player probably does not incorporate positional awareness in his play.

3 No-Limit Hold'em with a maximum stack-size (or buy-in) of $1000)
4 VPIP, Preflop Raise (PFR) and Aggression Factor (AF)
5 All players fold except the Big Blind, making him/her the winner
6 Early Position, after the blinds
7 Middle Position, after the EP
8 Late Position, the CO and BTN are considered LP
9 Cutoff, the player before the BTN
10 (Dealer) Button, the last player to act

VPIP	% of Hands	Hand Range
5	5.6	88+, ATs+, KQs, AKo
10	10.3	77+, A9s+, KTs+, QTs+, AJo+
15	15.1	77+, A7s+, K9s+, QTs+, JTs, ATo+, KTo+, QJo
20	20.4	66+, A4s+, K7s+, Q9s+, J9s+, T9s, A9o+, KTo+, QTo+, JTo
25	25.2	66+, A2s+, K6s+, Q8s+, J8s+, T8s+, A7o+, K9o+, QTo+, JTo
35	35.3	55+, A2s+, K3s+, Q5s+, J7s+, T7s+, 97s+, 87s, A4o+, K8o+, Q9o+, J9o+, T9o
50	50.7	33+, A2s+, K2s+, Q2s+, J4s+, T6s+, 96s+, 86s+, 76s, 65s, A2o+, K5o+, Q7o+, J7o+, T7o+, 98o
75	75.9	22+, A2s+, K2s+, Q2s+, J2s+, T2s+, 92s+, 83s+, 73s+, 63s+, 52s+, 43s, A2o+, K2o+, Q2o+, J4o+, T6o+, 96o+, 86o+, 75o+, 65o

Table 4.1: An example hand range chart in relation to VPIP

The number of players at the table is also a factor when looking at the VPIP. When playing FR[11] the VPIP value tends to be lower in comparison to playing at SH[12] tables or even HU[13]. The explanation for this is hand selection at the respective games. When playing FR games the playable hand range becomes smaller because of the simple fact that when playing against more people you need stronger hands to call, bet or raise in order to stay profitable (mathematically). SH games tend to be more aggressive, opening up the hand range and thus increasing the VPIP. The most extreme case is HU play where very high values between 60% and 85% are quite common because almost every hand played, since there is less chance that the opponent holds a strong hand. Implicitly VPIP gives us some information about the hand range of the player as seen in Table 4.1.

Looking at this table we can get a rough picture on how loose[14] or tight[15] the player is if we look at his or her VPIP. Players with extreme VPIP values (very high or very low) can be seen as potential targets to exploit (high) or avoid (low). If a player with a very low VPIP (5% or lower) bets or raises we can probably assume that he holds some kind of strong hand and it is better to avoid them and fold unless we have an equally strong hand. On the other hand players with a VPIP of 90% or higher will probably bet, call or raise with anything. This information however, is not perfect – it does not take sudden changes in playing style into consideration. As an example: a player has played several hundred hands tight-aggressively[16], but suddenly changes his style to loose-aggressive[17]. The VPIP will appear to be low, but the hand range is probably much wider than the VPIP would give away.

When we try to use VPIP as a measure to read hands a difference in the lower values (e.g. 5% vs. 20%) holds significantly more information than a difference in the higher VPIPs (e.g. 75% vs. 90%). This is because the hand range for 5% and 20% are significantly different and indicate different types of players while a difference between 75% and 90% does not tell us much because the player can still hold pretty much any type of hand. Please note that the table above only shows example hand ranges as each percentage of hands can also be made up of different hand combinations (e.g. for 25.2% of all hands we could remove the JTo/QTo+ and take more pocket pairs into our range instead of just 66+ and it would still be the same percentage).

[11] Full-Ring, a table with a maximum of 9 or 10 players
[12] Short-Hand, a table with a maximum of 6 players
[13] Heads-Up, a table with only two players
[14] A player who plays a lot of hands
[15] A player who plays very few hands
[16] A player who plays very few hands, but is very aggressive, i.e raises and bets often
[17] Analogous to tight-aggressive, but with more hands played

4.5 Preflop Raise

Preflop Raise % or PFR is the second stat of the ”holy trinity”. The formula for PFR is:

$$\text{PFR} = \frac{\text{Total hands Raised Preflop}}{\text{Total Hands Played}} \times 100$$

PFR is quite similar to VPIP and the two of them are almost always evaluated together. Like VPIP, PFR also depends on table size and position and can be split up to values for each position, which gives finer grained information about the player. In late position the PFR value should be highest, while in early and middle positions the value should be lower. Similar to VPIP this can be used as a hint if the player is playing positional aware poker or not. Generally speaking if a player's PFR in position[18] is under average, that player will tend to just limp[19] the pot. In short, PFR can be seen as the effective raising range of the player and we could potentially create a table similar to Table 4.1.

4.6 PFR/VPIP Ratio

The ratio between PFR and VPIP is (abbreviated PFR/VPIP), in our opinion, probably one of the best statistics to look at, as it combines two of the best stats into one metric.

$$\text{PFR/VPIP} = \frac{\text{Preflop Raise Percentage}}{\text{Voluntarily Put Money In Pot Percentage}}$$

As it is only the pure ratio between the two stats we obviously lose some information - for example two players with PFR and VPIP of 80/40 and 20/10 will have the same ratio, but are obviously different types of players. When looking at this stat we have to keep this in mind. The PFR/VPIP can obviously not cross 1 as raising preflop automatically implies putting money voluntarily into the pot.

4.7 Aggression Factor

Aggression Factor (sometimes also called Total Aggression (Factor), abbr. AF) is a postflop statistic and the last of the trinity of stats and probably the most complex out of the three. There are two formulas for the AF and both are called the same:

$$\text{AF}_1 = \frac{\text{Number of Bets} + \text{Number of Raises}}{\text{Number of Calls}}$$

$$\text{AF}_2 = \frac{\text{Bet Percentage} + \text{Raise Percentage}}{\text{Call Percentage}}$$

where the percentages are defined like this:

$$\text{Bet Percentage} = \frac{\text{Total Times Bet}}{\text{Total Opportunities to Bet}} \times 100$$

$$\text{Raise Percentage} = \frac{\text{Total Times Raised}}{\text{Total Opportunities to Raise}} \times 100$$

$$\text{Call Percentage} = \frac{\text{Total Times Call}}{\text{Total Opportunities to Call}} \times 100$$

These formulas are simple and seem to denote the same thing, but examples show that their numerical value can be very different. Basically both formulas are a ratio between aggressive actions (i.e. betting and raising) and passive actions (calling, checking and folding). Checking and folding does not influence the value of AF_1, but implicitly influences AF_2.

[18] *being in position* means being one of the last players to act, i.e. CO or BTN

[19] Calling the big blind, with the intention of seeing the flop cheaply

At every opportunity (postflop) where the player is able to bet he or she could also check or even fold, likewise at every opportunity where the player can call, raising and folding are also possible. When looking at the formulas it is apparent that betting and raising increases the result, while calling decreases it. The Aggression Factor can be calculated separately for each street by using the formulas below:

$$\mathrm{AF}_{1\mathrm{X}} = \frac{\text{Number of Bets X} + \text{Number of Raises X}}{\text{Number of Calls X}}, \text{ where } X \in \{\text{Preflop, Flop, Turn, River}\}$$

$$\mathrm{AF}_{2\mathrm{X}} = \frac{\text{Bet Percentage X} + \text{Raise Percentage X}}{\text{Call Percentage X}}, \text{ where } X \in \{\text{Preflop, Flop, Turn, River}\}$$

Even though there are two formulas for AF, Hold'em Manager 2 and PokerTracker 4 seem to use AF_1 rather than AF_2 which is why we put our focus on it. Aggression Factor is an extremely controversial stat where opinion, knowledge and assumptions seem to clash. A very interesting discovery made during our research is that many players do not seem to either know or care for the difference between the two formulas, even though using one or the other makes quite a difference.

A "commonly recommended" value for AF is in between 1.5 and 3.5 [14], while most players consider AFs below 1.0-1.5 as *passive* and anything above 3-3.5 *aggressive* – these values have been mentioned in connection with both AF_1 as well as AF_2, but assuming they are refering to the PokerTracker stat called *Total AF* they ultimately mean AF_1.

Aggression Factor on its own is not a very good stat to look at, which is why it is usually evaluated in conjunction with PFR and VPIP as it does not hold much information besides the ratio of aggressive to passive actions.

We can see this if we construct an example: say we have two players A and B with the same AF of 2.5, player A has a VPIP/PFR of 15/12 and player B 70/10. Judging from their PFR and VPIP player A is a solid tight player, while player B is extremely loose and does not raise a lot of hands preflop. Now in this example they both have the same AF. Player A does not depend heavily on making a hand on the flop as he or she mostly plays only very strong hands, so seeing this player bet and raise 2.5 times as much as he/she calls is not surprising. On the other hand player B has the exact same AF while being very loose on the flop which means that the player is very dependant on actually hitting something on the flop as his wide hand range will usually result in having a weak hand postflop. To maintain an AF of 2.5 while being this loose preflop requires the player to call very rarely while folding a lot (remember, folding does not affect the AF) in conjunction with betting/raising a lot (which is highly unlikely due to the postflop hand strength of a 70/10 player).

When slightly modifying this example we can see something different: player A remains at 15/12/2.5 while player B now has 70/10/1.0 (VPIP/PFR/AF) stats. We already said that the AF for player A is quite normal considering his preflop range of premium hands, but what about player B? An AF of 1.0 may seem low at first, but when considering his VPIP (and his resulting hand range) this view changes a bit. Player B will see a lot more flops than player A because of the high VPIP, this means that to maintain an AF of 1.0 he will have to bet and raise in a 1:1 ratio to calls. This does not sound like a very aggressive player, but considering the PFR of only 10 the player would need to raise and bet a lot on postflop streets to compensate for that, which in turn can be again considered aggressive.

These examples show how little AF alone says about the playstyle of certain players. Generally speaking the AF per street can be considered more useful, since it gives finer grained information on how the players behave on each street. This can result in significant differences between the AF value of each street, since the play style can vary on each street as well. There are also other factors which can influence the Aggression Factor such as the number of players on the table - in an 9- or 10-player game this would show most apparently on the preflop AF since betting and raising is more unlikely due to the tighter playstyle at FR games in general.

For the remainder of the document we will generally refer to AF_1 when we mention the stat AF.

Aggression Frequency and Aggression Percentage

In addition to AF, PokerTracker and Hold'em Manager use two different metrics called Aggression Frequency (AFq) and Aggression Percentage (Agg%). The AFq is calculated as shown below:

$$\mathrm{AFq} = \frac{\text{Total Times Bet} + \text{Total Times Raised}}{\text{Total Times Bet} + \text{Total Times Raised} + \text{Total Times Called} + \text{Total Times Folded}} \times 100$$

Aggression percentage only considers postflop actions and is best explained with an example: if a player bets the flop and the turn but folds or checks the river his Agg% is 66 because he made 2 out of 3 aggressive actions during each street. Bet, Raise and Check-Raise are considered aggressive actions, while Check and Fold are not. From this example we can derive the formula:

$$\text{Agg\%} = \frac{\text{Number of Aggressive Actions}}{\text{Number of (seen) Streets}}$$

If we compare the two it becomes apparent that they are quite different. A major difference is that we can calculate the AFq for each street separately while that is not possible for the Agg%. The other difference is that checking does not have any impact on AFq but is taken into consideration when calculating the Agg%.

4.8 BB/100, $/100 and $/Hand

Unlike *Amount Won*, which tells us nothing about the limit, this is the stat to look at if you want to know how good a player is at his/her limit (i.e. how much the player actually wins or loses in relation to the limit). In most cases *BB* means big bet (which is equivalent to 2 big blinds) but can mean a single big blind as well. BB/100 is affected by the limit the games are played on. As an example if a player has 2 BB/100 at NL200 ($1/$2 blinds) it means he is winning $4 per 100 hands played. Any value above 0 can be considered good, while anything below means that the player is losing money.

Very similar to BB/100 are $/100 (Money Won per 100 Hands) and $/Hand (Money Won per Hand) which are calculated as follows:

$$\$/100 = \frac{\text{Total Amount Won/Lost}}{\frac{\text{Total Hands Played}}{100}}$$

$$\$/\text{Hand} = \frac{\text{Total Amount Won/Lost}}{\text{Total Hands Played}}$$

$/100 shows how much raw money we have won per 100 hands, while $/Hand shows us how much money we have won or lost per hand played. These stats suffer from the same problem as Amount Won, since they do not contain any information on what limit the hands were played on.

4.9 Went to Showdown

Went to Showdown Percentage (abbr. WTSD or WTSD%) is the fraction of times seeing the showdown after seeing the flop. It is calculated with this formula:

$$\text{WTSD} = \frac{\text{Total Times Went to Showdown}}{\text{Total Times Saw The Flop}} \times 100$$

As the formula implies WTSD is a general statistic. This stat depends on a number of different factors like number of players, limit and play style. Seeing a showdown in FR games is less likely because of the lower VPIP and PFR which results in less flops seen overall. As stated previously the limit also plays a certain role. At higher limits it becomes less likely that players will go to showdown because of the more aggressive play style at higher limits, where the hands end before the showdown.

The style of the player also influences his WTSD. Generally, loose players will have a lower WTSD, while tighter players will have a higher WTSD value. This is because of their hand range. A loose player will have a bigger hand range pre-flop, but will need to hit the flop (or turn and/or river) in order to proceed to win the pot. If he or she did not hit anything on the flop the player will either have to fold or try to win the pot by playing aggressively with the intention of inducing a fold from the opponents. On the other hand a tight player with a much smaller hand range will try to maximize profit by value-betting as much as possible (trying keeping the remaining players in the pot in order to maximize its size) if he or she has made a strong hand on the flop. Tight players usually do not try to win the pot before the showdown, unless they are not confident that their hand will beat the others, but will try to go all the way to the showdown to increase the pot size as much as possible.

Putting things together when looking at the WTSD we can deduct following information if the value deviates from the average. If the player's WTSD is lower than average he or she is probably a weak player who folds a lot before the showdown, while if it is higher than average we can assume that the player is calling a lot. In both cases the player is probably bad at reading the opponents hand, overestimating or underestimating the hand strength. These players can be seen as possible targets and WTSD can be used as a steal- or value-bet-indicator. For example we can bluff the weak tight opponents out of the pot to win the hand before the showdown and value-bet the callers to increase pot size at the showdown to maximize profit.

4.10 Won $ at Showdown

Won Money at Showdown % (W$SD or WSD) describes the percentage of times the player won money when he/she saw the showdown and is calculated this way:

$$\text{W\$SD} = \frac{\text{Total Times Won Money at Showdown}}{\text{Total Times Went to Showdown}} \times 100$$

This stat is a percentage stat, it does not say how much raw money was won at the showdown, but how frequently the player has won money (at all) at the showdown.

4.11 Stealing

AttSB

Attempt to Steal Blinds % is a more specialized stat and is strictly connected to preflop play. It describes the ratio of actually trying to steal the blinds (making an open raise) when folded to (all players fold on their turn, leaving the player to be the first to act) as the CO, BTN or SB. A steal is an attempt to win the blinds, forcing the players who posted them to fold, by opening with a raise (usually using a hand of mediocre strength). The equation for AttSB is:

$$\text{AttSB} = \frac{\text{Total Attempts to Steal Blinds}}{\text{Total Opportunities to Steal Blinds}} \times 100$$

The scenario described in this stat usually does not happen very frequently, which is why the sample size for this stat needs to be around at least a few hundred in order to have enough occurences of this scenario. In SH games this stat tends to converge more quickly because you are playing more often in the positons of CO, BTN or SB and thus making the chance of having the opportunity to steal more likely in general.

AttSB can be seen as the specialized PFR stat for being in late position (with the exception of the SB case) and for good players this stat is almost always higher than their total PFR. The two stats are linearly correlated, as an attempt to steal is an open raise, which counts towards the PFR. As described in the previous Section 4.5, the PFR of a positionally aware player will increase the later his turn is. Players with an AttSB that is roughly the same as their PFR are most likely not playing their position, while players with a significantly higher AttSB are more likely to overplay their position (widening their hand range too much and betting with weak hands), making them targets for 3-betting[20] because their hand range will not hold against pressure, forcing them to fold.

Stat aware players will typically aim for a value between 25 and 40 for AttSB. Using Equilab with a relatively wide, but not exceptionally weak, hand range of 22+, A2s+, K9s+, Q9s+, J9s+, T9s, 98s, 87s, 76s, A2o+, KTo+, QTo+, J9o+, T8o+, 98o, 87o, 76o (35.14% of all hands) on the button against two random hands (on SB and BB) we still have an equity of roughly 42% even if the steal attempt fails.

One additional thing to note is that even if the player meant to raise a strong hand (e.g. AQs or AA) for value and the blinds fold, it will still count as an (successful) attempt to steal blinds.

[20] 3Bet. The third bet, also known as a re-raise. The raise is called the 2Bet, or second bet.

AttSB can split up further into two stats called AttSB (LP) and AttSB (SB), which are the stats for their respective position – namely late-position and small blind. The formulas are as follows:

$$\text{AttSB (LP)} = \frac{\text{Total Attempts to Steal Blinds from LP}}{\text{Total Opportunities to Steal Blinds from LP}} \times 100$$

$$\text{AttSB (SB)} = \frac{\text{Total Attempts to Steal Blinds from SB}}{\text{Total Opportunities to Steal Blinds from SB}} \times 100$$

These two stats are more specialized with the intention to show more detail.

Stole

The stat which describes the rate of success for stealing is simply called **Stole** and is calculated as follows:

$$\text{Stole} = \frac{\text{Total Times Successfully Stole}}{\text{Total Attempts to Steal Blinds}} \times 100$$

Reactions to Steals

Another set of statistics related to stealing are reactions to steals, namely folding, calling or (re-)raising steals. The formulas are:

$$\text{Fold to Steal} = \frac{\text{Total Times Folded to a Steal}}{\text{Total Opportunities to Fold to a Steal}} \times 100$$

$$\text{Call Steal} = \frac{\text{Total Times Called a Steal Attempt}}{\text{Total Opportunities to Call a Steal Attempt}} \times 100$$

$$\text{3Bet Steal} = \frac{\text{Total Times 3Bet a Steal}}{\text{Total Opportunities to 3Bet a Steal}} \times 100$$

There is also a separate statistic for 3-betting when the player faced a steal from the CO or BTN.

$$\text{3Bet Steal (LP)} = \frac{\text{Total Times 3Bet a Late-Position Steal}}{\text{Total Opportunities to 3Bet a Late-Position Steal}} \times 100$$

Since it is only possible to face a steal when the player is a blind, this spot is more rare than having the opportunity to steal. Consequently these stats also require a larger sample size.

4.12 Continuation Betting

Continuation Bet % or Cbet%/CBet/CB is a postflop stat which is calculated separately for each postflop street (flop, turn, river). The formula is:

$$\text{CBet X} = \frac{\text{Total Times Continuation Bet X}}{\text{Total Continuation Bet X Opportunities}} \times 100, \text{ where } X \in \{\text{Flop, Turn, River}\}$$

A continuation bet opportunity is when the player was the preflop aggressor (first to bet or raise) and is able to bet first on the flop (e.g. all players check or fold to him). Likewise this is the same for the turn and river CBet values: an opportunity for a turn/river continuation bet is when the player was able to continuation bet on the flop/turn. As this scenario implies, a very large sample size is required for this stat to be significant (increasingly so for each of the later streets), because opportunities to continuation bet will be quite rare.

This stat is mainly used at the flop because it has a reasonable sample size requirement (in comparison to later streets) and can give us some information about the opponents hand. A high value means that the player usually follows up his or her preflop aggression with a continuation bet, while a low value means that the player will not often make a continuation bet if he/she was the preflop aggressor.

A (very) high flop Cbet% can indicate that the player will make a continuation bet on the flop regardless of his hand strength but a continuation bet from a player with a relatively low Cbet% (e.g 20 to 60) implies that he or she will probably hold a strong hand. A concrete example for this can be if a player with a Cbet% of 40 raised preflop and checked postflop we can assume that he did not hit the flop (because of the low CBet% the player will most likely only bet if the hand is decently strong, the check indicates a weaker type of hand) and thus attempt to exploit this weakness by betting, trying to induce a fold.

Reactions to Continuation Bets

Similar to reactions to stealing there is also a set of statistics which cover the reactions to continuation bets. There are a total of 12 stats and all of them need a significant sample size of several hundred, better yet thousand, hands:

$$\text{Call CBet} = \frac{\text{Total Times Called a CBet}}{\text{Total Opportunities to Call a CBet}} \times 100$$

$$\text{Call CBet X} = \frac{\text{Total Times Called a X CBet}}{\text{Total Opportunities to Call a X CBet}} \times 100, \text{ where } X \in \{\text{Flop, Turn, River}\}$$

$$\text{Raise CBet} = \frac{\text{Total Times Raised a CBet}}{\text{Total Opportunities to Raise a CBet}} \times 100$$

$$\text{Raise CBet X} = \frac{\text{Total Times Raised a X CBet}}{\text{Total Opportunities to Raise a X CBet}} \times 100, \text{ where } X \in \{\text{Flop, Turn, River}\}$$

$$\text{Fold to CBet} = \frac{\text{Total Times Folded to a CBet}}{\text{Total Opportunities to Fold to a CBet}} \times 100$$

$$\text{Fold to CBet X} = \frac{\text{Total Times Folded to a X CBet}}{\text{Total Opportunities to Fold to a X CBet}} \times 100, \text{ where } X \in \{\text{Flop, Turn, River}\}$$

The utility of these stats is questionable, mainly because of their enormous sample size requirement.

4.13 Raising

As raising (3-betting, 4-betting, etc.) is an integral part of the game there is also a large set of statistics which are in fact about raising. For starters we have the two stats 3Bet Total and 3Bet per Street which are calculated as follows:

$$\text{3Bet Total} = \frac{\text{Total Times 3Bet}}{\text{Total Opportunities to 3Bet}} \times 100$$

$$\text{3Bet X} = \frac{\text{Total Times 3Bet X}}{\text{Total Opportunities to 3Bet X}} \times 100, \text{ where } X \in \{\text{Preflop, Flop, Turn, River}\}$$

Both stats simply denote the percentage the player 3bet on any given street (3Bet Total) when he or she had the chance to do so. The split up variant of this stat for each street is basically the same, but only considers the actions on the certain street. As raising and especially re-raising (3betting) is not done that often, the sample size requirement for these statistics are quite high. The preflop and flop variants probably have the highest utility and usability as they start to become useful after a sample of a few hundred hands while the two later streets (and the total value) require significantly more hands to converge.

Reactions to Raises

There is also a set of statistics about reactions to 3Bets analogous to reactions to steals and continuation bets. All of the following stats disregard previous actions, so only the raw counts of folds, calls and raises when facing a 3Bet are accounted for, given the player had the opportunity to do so.

$$\text{Fold to 3Bet} = \frac{\text{Total Times Folded to a 3Bet}}{\text{Total Opportunities to Fold to a 3Bet}} \times 100$$

$$\text{Call 3Bet} = \frac{\text{Total Times Called a 3Bet}}{\text{Total Opportunities to Call a 3Bet}} \times 100$$

The stat for **Raise 3Bet** does not exist explicitly, but rather it is grouped into the stat called **4+Bet**, which contains all further raises past the re-raise including the fourth bet (the "re-re-raise") and so on.

$$\text{4+Bet} = \frac{\text{Total Times 4+Bet}}{\text{Total Opportunities to 4+Bet}} \times 100$$

Reactions to 4+Bets are also grouped together. The raise reaction to a 4+Bet is the stat itself.

$$\text{Call 4+Bet} = \frac{\text{Total Times Called a 4+Bet}}{\text{Total Opportunities to Call a 4+Bet}} \times 100$$

$$\text{Fold to 4+Bet} = \frac{\text{Total Times Folded to a 4+Bet}}{\text{Total Opportunities to Fold to a 4+Bet}} \times 100$$

As all spots are exceedingly rare, the sample size requirements for all these stats are extremely high as well. These stats probably start converging after a few thousand hands minimum.

4.14 Other Statistics

This section shows some other (minor) stats and their formulas.

Win%

Win% is a simple stat which shows how many of the played hands were actually won. It does not include the potsize or winnings.

$$\text{Win\%} = \frac{\text{Total Times Won a Hand}}{\text{Total Hands Played}} \times 100$$

XR Total

XR Total or Check-Raise Total is the percentage of the times that the player checked and then raised when facing a bet. Only postflop street actions are taken into account for this stat.

$$\text{Check-Raise} = \frac{\text{Total Times Check-Raised}}{\text{Total Opportunities to Check-Raise}} \times 100$$

Bet

Bet is simply the percentage of times the player bet when he had the chance to do so. It can also be calculated per street.

$$\text{Bet Total} = \frac{\text{Total Times Bet}}{\text{Total Opportunities to Bet}} \times 100$$

$$\text{Bet X} = \frac{\text{Total Times Bet X}}{\text{Total Opportunities to Bet X}} \times 100, \text{ where } X \in \{\text{Flop, Turn, River}\}$$

Call vs BTN Open

As the name implies this stat is the percentage of times the player called an open raise made by the button when he had the chance to do so. This stat is a good indicator for positional awareness of the player, because facing a raise from the button is disadvantageous when in EP, unless the player is certain that the button raises with very weak hands.

$$\text{Call vs BTN Open} = \frac{\text{Total Times Called when Button Opened}}{\text{Total Opportunities To Call when Button Opened}} \times 100$$

Preflop Limp

This stat shows the percentage of times the player limped preflop, which means he or she just called the big blind, when the player had the chance to do so. While **Preflop Limp** is a very specialized stat, in our opinion it is a quite interesting stat to look at because limping is commonly considered weak play.

$$\text{Preflop Limp} = \frac{\text{Total Times Limped Preflop}}{\text{Total Opportunities to Limp Preflop}} \times 100$$

Other Limp-related stats

Bet Flop (Limped Pot), as the name implies, is the percentage of times a player bet the flop after all players limped preflop, given he/she had the opportunity to do so as the first.

$$\text{Bet Flop (Limped Pot)} = \frac{\text{Total Times Bet Flop when Limped Pot}}{\text{Total Opportunities to Bet when Limped Pot}} \times 100$$

The following three stats describe preflop reactions after the player limped. **Limp/X** means that the limping player faced a raise preflop where he/she then did X – this can be calling, folding or raising.

$$\text{Limp/Call} = \frac{\text{Total Times Limp Called Preflop}}{\text{Total Times Faced a Raise after Limping}} \times 100$$

$$\text{Limp/Fold} = \frac{\text{Total Times Limp Folded Preflop}}{\text{Total Times Faced a Raise after Limping}} \times 100$$

$$\text{Limp/Raise} = \frac{\text{Total Times Limp Raised Preflop}}{\text{Total Times Faced a Raise after Limping}} \times 100$$

Donkbet Flop

Another stat which is similarly related to bad play like limping is the **Donkbet** stat. A donkbet is a bet on the flop, turn or river before letting the (preflop) aggressor make his/her continuation bet. The name stems from "donkey" and implies that the action is usually not the best move to make.

$$\text{Donkbet X} = \frac{\text{Total Times Donk Bet the X}}{\text{Total Opportunities to Donk Bet the X}} \times 100, \text{ where } X \in \{\text{Flop, Turn, River}\}$$

Float

A **float** describes a scenario similar to the donk bet. It is a bet in position after the preflop aggressor has failed to make (i.e. did not) a continuation bet on the flop (or following streets). There is also a combined variant called **Float Total**.

$$\text{Float Total} = \frac{\text{Total Times Float Bet}}{\text{Total Opportunities to Float Bet}} \times 100$$

$$\text{Float X} = \frac{\text{Total Times Float Bet on the X}}{\text{Total Opportunities to Float Bet on the X}} \times 100, \text{ where } X \in \{\text{Flop, Turn, River}\}$$

Cold Call

Calling a raise without having put money into the pot before (blinds count as money as well) is called a *cold call* (abbreviated CC). A **Cold Call 2Bet** is the percentage of times a player called a 2bet (the first raise) without previously putting money into the pot. A **Cold Call 3+Bet** is the portion of times the player called a 3bet or higher without having made the previous raise. For example if the player called a 4bet and did not make the immediate 3bet previously, he/she cold called a 3bet+ preflop.

$$\text{CC 2Bet Preflop} = \frac{\text{Total Times Player Cold Called Preflop}}{\text{Total Opportunities to Cold Call Preflop}} \times 100$$

$$\text{CC 3Bet+ Preflop} = \frac{\text{Total Times Player Cold Called 3Bet or Higher Preflop}}{\text{Total Opportunities to Cold Call 3Bet or Higher Preflop}} \times 100$$

5 Analysis

This chapter compiles the results of the analysis, including general information on the problems we faced. The two main approaches in the research were an univariate analysis and machine learning techniques.

5.1 Approach

In this section we talk about how this research was conducted and which problems were faced during the progress. Our approach can be summarized into four steps:

1. Understanding hand histories and more importantly their contained stats
2. Gathering a significant sample size of hand histories
3. Converting the hand histories into a usable set of data
4. Analyzing the converted data set

Understanding the stats contained in the hand histories involved gathering information from various poker-related sites. While the official websites of PokerTracker and Hold'em Manager had well defined formulas for each statistic, this was not neccessarily the case for sites which explained how to interpret them. This was especially problematic with statistics where different formulas for the "same" statistic existed – the most prominent example being Aggression Factor (which is explained in Section 4.7). In addition to ambiguity, when talking about the topic of interpretation there is always the challenge of differentiating between opinions and facts. We have found many useful forum posts on how to interpret certain statistics and while it was often possible to see a common consensus on which values for each statistic are "good" or "bad", this kind of information always has to be viewed critically as the general opinion might not always be the right one. A common approach is to define player types, such as the classic player types *tight-aggressive, loose-aggressive, tight-passive and loose-passive*[37]. The main problem of this approach is defining the thresholds for each stat of each type - at what VPIP value can we consider someone *tight* or *loose*? Defining thresholds is very prone to bias and subjectivity.

Obtaining a large batch of hand histories proved to be quite difficult unless being a long-term multitabling poker player oneself, as players usually do not want to disclose their hand histories because they contain a lot of information about themselves. We contacted several third-party sites where hand histories are data mined. Poker sites have always frowned upon data mining hand histories and many third-party data mining sites were already contacted by the major poker site providers (namely PokerStars and PartyPoker) to stop their conduct. At this point we would like to thank David from HandHQ.com [8] who has provided us with 22 million hand histories for academic and analytical purposes which were mined from an anonymous site in between April 2009 and August 2010 (a timespan of 16 months). All hands are from NLHE200[1] ($1/$2 blinds). Our data set contains roughly 17 million 3 to 6 player hands and 5 million 7 to 10 player hands.

This data set might not be optimal as it only contains one limit (i.e. NL200), but larger datasets with multiple limits may contain more noise and variables. The size of the hand histories are roughly 1 GB per million. In the end we decided to only use the 3-6 player batch of 17 million hands in order to minimize the number of possible factors playing a role in the analysis.

During our search we also found an academic dataset labeled the *Billion Hand History Project*. We have tried contacting several sources (including the project initiator) in order to obtain this dataset, but unfortunately we were not able to do so. It is also questionable if that dataset would have met our needs – namely if the hand histories would have been importable into our tracking software as it was supposedly saved in a compressed proprietary format.

Converting the hand histories, or rather importing them in Hold'em Manager and PokerTracker was a problem which we largely underestimated, considering the massive size of our database. The cumulative size of our hand histories was roughly 20 GB of raw text files (made up of characters and whitespace). As we briefly outlined in Section 3.1 import speeds were terrible and some database structures were not really suitable for our cause.

[1] No-Limit Hold'em $200 buy-in

In the end we decided to prioritize a suitable feature set instead of focusing on import speeds. This lead to using PokerTracker 4 as it had the most well-fitted feature set for our purposes. Unfortunately PokerTracker 4 also had the worst import speeds of all tested software – peaking at only 35 hands per second on our initial test import. After several hours this speed continued to decrease after which we canceled our first test. After extensive research on how the hand import process worked we managed to optimize hand import speeds by 1.) tweaking our PostgreSQL database config and 2.) optimizing (the content of) our hand history files.

The tuning of our PostgreSQL database included using an older release version, 9.0 instead of 9.2 which was recommended to us by the PokerTracker Support Team, and tweaking of parameters in the config file (mostly related to buffering and writing content).

For hand history optimization we found that data mined hands are generally very slow to import due to the fact that the hand histories are not sorted by their game ID. Due to the database structure of PokerTracker 4 this resulted in very heavy slowdowns as the hand histories within the database are sorted (or inserted in a sorted manner). A factor that has to be considered is the used type of data storage. We used a traditional hard disk drive instead of a newer and faster solid state drive. On an hardware level the slowdowns can be explained by the construction of a hard disk drive - if the raw data is not sorted and has to be inserted in a sorted manner this results in a lot of read and write overhead because already written data has to be moved which takes time as the read/write head has to be moved quite frequently. This might be insignificant in smaller databases but when looking at databases like ours it results in noticable slowdowns. We bypassed this problem by modifying our data-mined hands beforehand using an external script. Besides from sorting all hands by their game ID we also removed all duplicates and corrupted hands (e.g. hands where the connection was lost or where other problems occured).

We managed to reach hand import speeds of roughly 130 hands per second after optimization and it took us about 50 to 60 hours to import our database of 3-6 player hands (around 16 million after hand history optimization) – considering the speed was still dropping down from 35 hands per second before, it saved us a lot of time as it would have probably taken 200 to 300 hours without optimization.

5.2 Dataset Composition

In this section we describe the general statistics of our sample, namely its composition. We split our sample into three disjoint classes of losers, breakeven players and winners. Of course, there is also the "Total" group which simply includes all players regardless of their class. Winners and losers are defined as players who have a BB/100 value greater or equal 1 or lower or equal -1 respectively. The breakeven class consists of all players who possess an BB/100 value in between -1 and 1 (exclusive). Having a BB/100 value in (-1,1) can be considered breakeven because of no significant losses or winnings.

We also used various filters where we used a threshold for the minimum number of played hands required for a player to appear in the sample. Our default filter is 500 minimum hands. We arbitarily chose this value because running queries on the database without a hand filter took too many resources. In Table 5.1 the general statistics of our sample are shown and each column pair consists of the raw count as well as the percentage. The outmost right column shows the trend whether the value is increasing or decreasing. With our filtering alone we can see a correlation between the class (namely winner, loser or breakeven player) and the number of hands played. This is most likely due to the fact that players with more hands have more experience and are thus, better overall. The general number of players without applying filters is 30,975.

Hand Filter	500		1000		2500		5000		10000		20000		
Losers	7401	67.46	4667	65.76	2339	59.26	1286	52.72	643	44.53	302	37.19	↘
Breakeven	482	4.39	395	5.56	311	7.88	246	10.09	188	13.02	127	15.64	↗
Winners	3088	28.15	2036	28.68	1297	32.86	907	37.19	613	42.45	383	47.17	↗
Total	10971		7098		3947		2439		1444		812		

Table 5.1: Statistics of the sample with various filters applied.

While we have talked about the composition of our data, we did not yet mention *what* is actually included in the data. Our database basically consists of a list of vectors or tuples and each of them represents a player. In its original form each vector consists of 72 attributes. Besides the mentioned statistics in Chapter 4, it also contains the name of the player. Depending on the analysis we have removed some of the attributes (for instance we did not include the names in the univariate analysis or for the histograms, while we exluded the BB/100 and WSD attribute, as they directly correlate with the class attribute, when we did our tests with machine learning algorithms) from the dataset.

The next thing we did was actually plotting the data with the use of histograms. For this we used the *Visualize All* feature of Weka. Histograms are a visualization of the frequency in which certain values appear, so the x-axis shows the value, while the y-axis shows its frequency. The histograms are colored by classes, winners are green, losers are red and breakeven players are orange (for better distinction). As we can see in Figure 5.1 some of our stats appear to be somewhat normally distributed with skewing (for instance *Turn AFq* or *WSD*). Because of our relatively large sample size of 10971 players most of the histograms are also quite fine-grained.

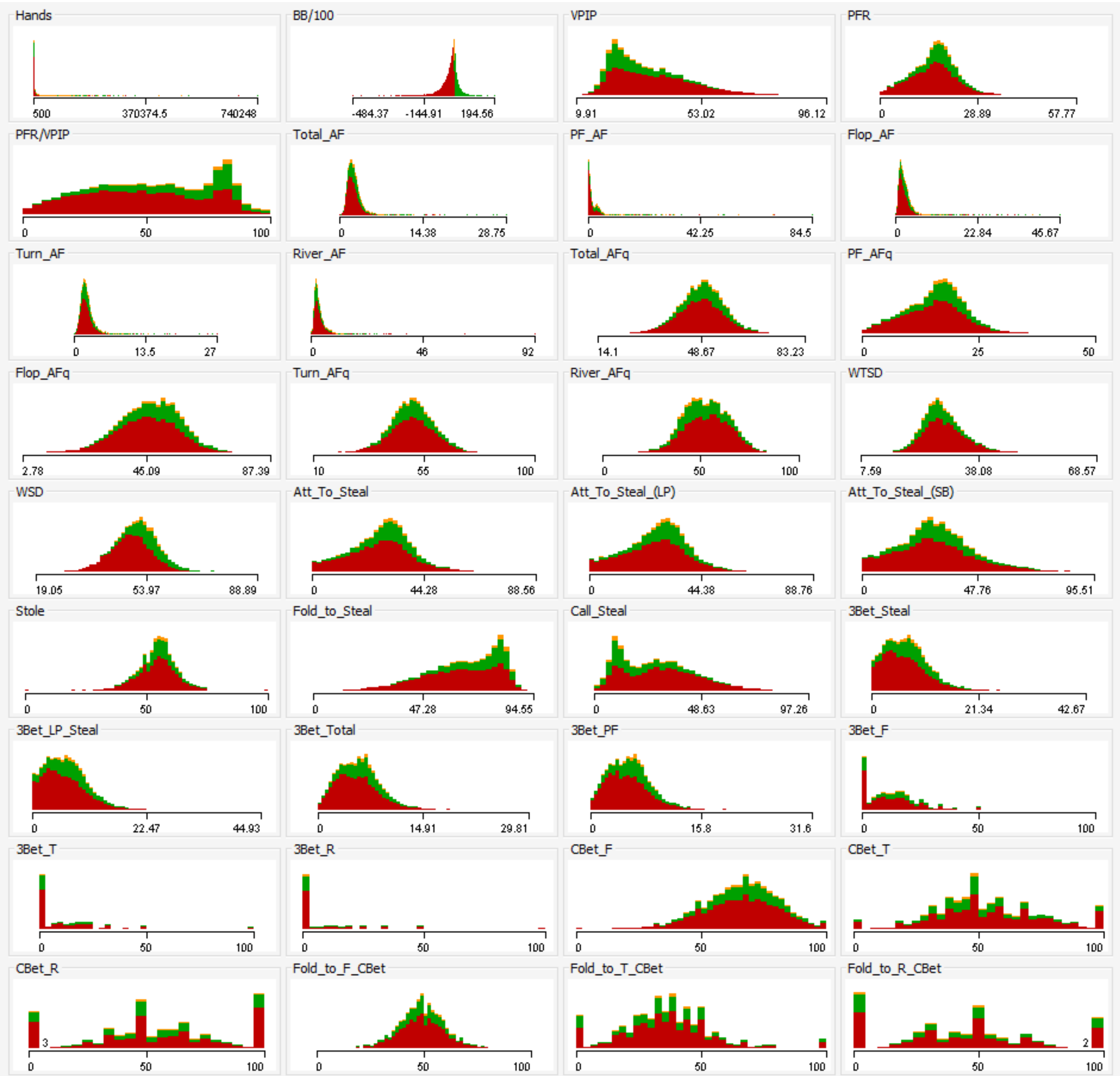

Figure 5.1: Histograms of our sample with default filter (Part 1)

The second set of histograms shown in Figure 5.2 mostly shows stats which require high sample sizes. Many of these stats have quite a few outliers and are most likely not normally distributed. We also created histograms with a smaller sample with the intention of A. removing outliers and B. see if we could observe more noticable differences between winners and losers as all of our green and red areas are of similar shape.

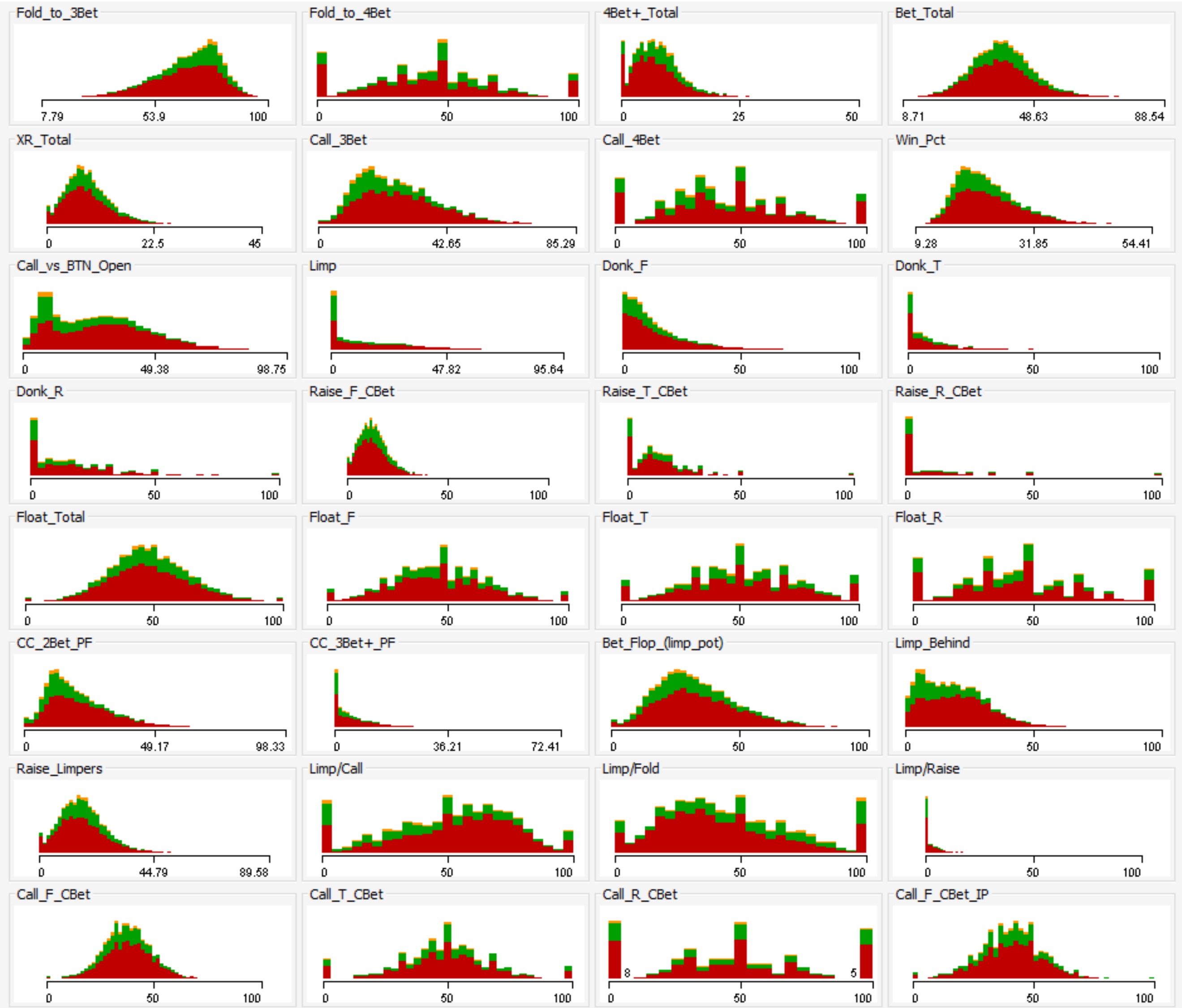

Figure 5.2: Histograms of our sample the default filter of 500 minimum hands (Part 2)

Obviously the resulting histograms shown in Figure 5.3 and 5.4 are a lot more rough but show practically no outliers. As already seen in Table 5.1 this filtered sample has nearly the equal amount of losers and winners. In the histograms the winners are usually more tightly packed together (in terms of spread on the x-axis) with more noticable peaks than losers. On the other hand, the distribution of losers is more spread out and flat-shaped (e.g. as seen in *Att to Steal, Float Total* or *Bet Flop (Limp Pot)*. Unfortunately that is the only noticeable difference between the two major groups as the overlapping is still very present.

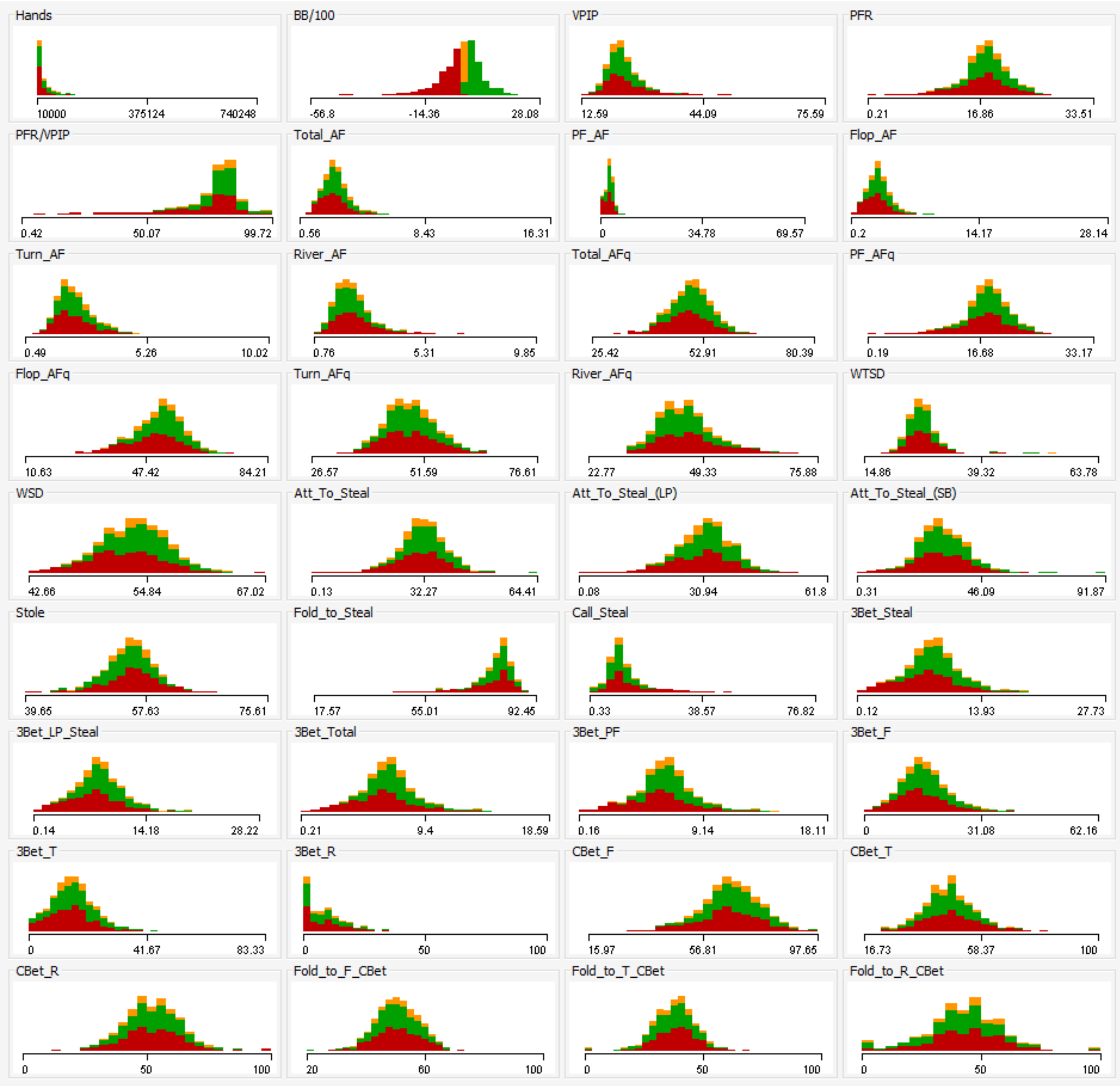

Figure 5.3: Histograms of our sample with a 10000 minimum hand filter (Part 1)
Unfortunately Weka does not show a scale for the y-axis.

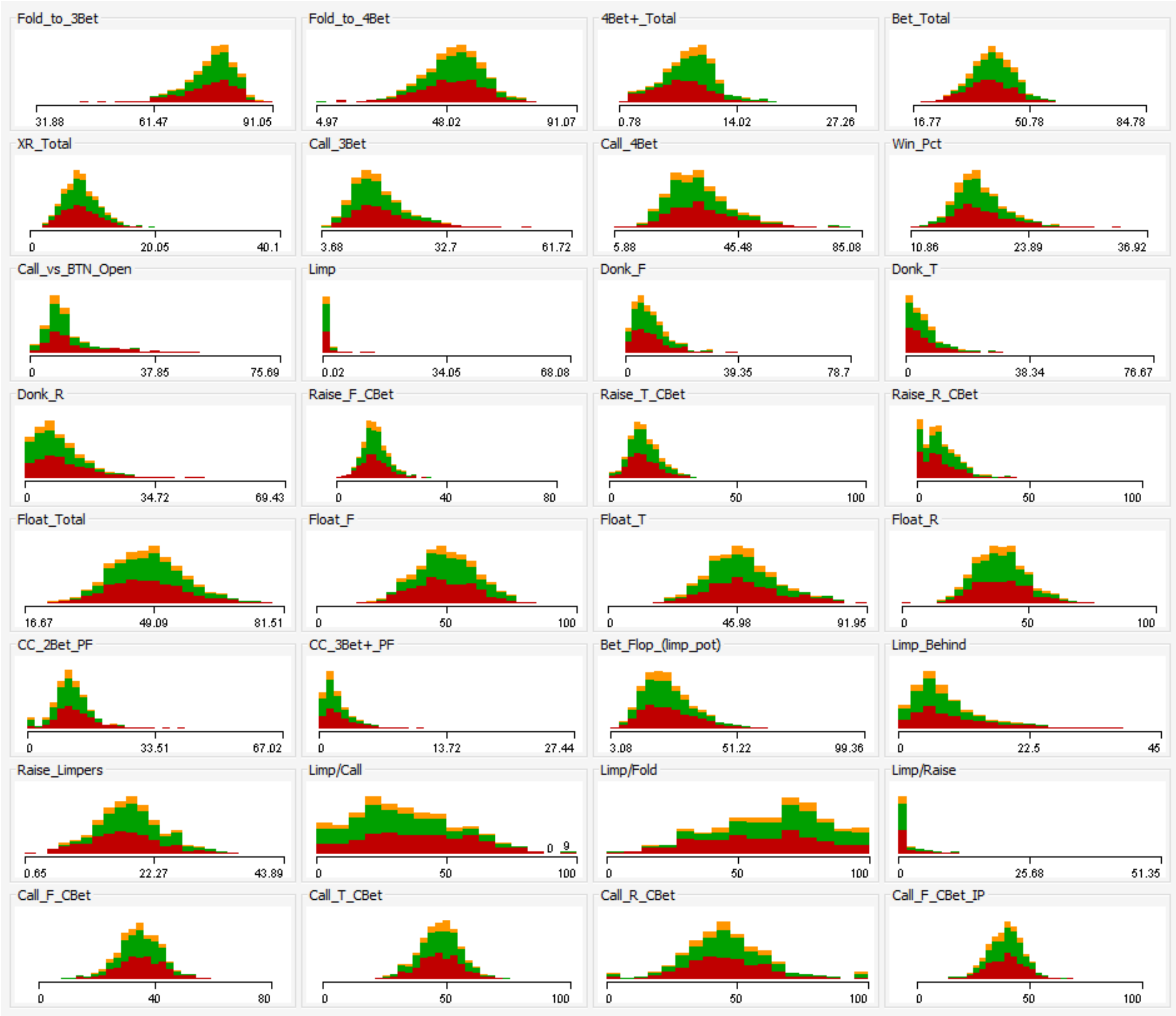

Figure 5.4: Histograms of our sample with a 10000 minimum hand filter (Part 2)

Before going to the next step (the univariate analysis) we proceeded to test our statistics for normality. The reason we did this is simple: parametric statistics rely on the assumption that the test data follows some kind of distribution (mostly the normal distribution, also known as Gaussian distribution), however as we have seen in our histograms this might not neccessarily be the case. Using parametric statistics on data which does not follow any known distribution can give misleading and/or inappropriate results.

In order to test the normality of our data we used R[2]. We used the Shapiro-Wilk test[3] on all of our variables to test for normality. One limitation of the Shapiro-Wilk test is, that the sample size must be between 3 and 5000, so we used the data set with a 2500 hand filter which consisted of 3947 instances. Using a loop we ran the test for all attributes/variables. As for the result, nearly all of our p-Values[4] were in the order of magnitude of -16 (i.e. values like 3×10^{-16}). The magnitude of our highest p-Value was -4. Using the commonly used significance level of $p \leq 0.05$ (or 95% confidentiality) we were extremely far from accepting the null hypothesis that our data is normally distributed. Since the result of the tests were not even remotely close to 0.05 (e.g 0.03 or 0.06) we did not run any further tests on normality and concluded that the data was not normal distributed.

[2] A free programming language for statistical compuation

[3] `http://en.wikipedia.org/wiki/Shapiro%E2%80%93Wilk_test`

[4] `http://en.wikipedia.org/wiki/P-Value`

As a consequence we decided to primarily use non-parametric methods and statistics, which have higher statistical power to describe the data. These include the median, mode, median absolute deviation, first and third quartile and implicitly the interquartile range. For completeness we have also included the mean and the standard deviation (which are considered non-robust and not optimal for non-parametric data). We will briefly explain the statistics we used in the next section.

5.3 Statistics of the univariate analysis

Mean

The **mean**, or also called arithmetic mean or average, is probably the most common statistic to describe a data set. In our case we are talking about the sample mean ($\overline{x}$ called "x bar") which is the sum of all values divided by the number of values. Unfortunately the mean is not a robust measure as a single outlier value can greatly influence it, but for the sake of completeness we still decided to use it.

Median

Unlike the mean, the **median** is a robust measure of central tendency. The median is defined as the value which splits the data sample in two equal sized halves. As an example a data set containing the numbers 1, 7, 8, 9, 11, 12, 13 has the median of 9. We will denote the median as $\tilde{x}$. For normal distributions the median is the same as the mean and mode.

Mode

The **mode** is the value which appears most often in the data set and is another robust measure. In normal distributions the mode is usually the same as the median and the mean, but in other distributions the mode is usually different from those. Basically the mode denotes the value with the highest frequency of occurence (and therefor highest peak) in a sample when plotted in an histogramm.

Standard Deviation

The **standard deviation** (commonly abbreviated *SD* or σ) is a non-robust measure of variabillity which shows the average distance of each sample to the sample mean. It is also defined as the square root of the variance σ^2. It has good statistical power for symmetrical (nearly) normal distributed samples, but suffers when the sample is skewed or very wide-spread.

Median Absolute Deviation

As an alternative to σ we use the **median absolute deviation** (*MAD*) which can be considered its robust counterpart. It is defined as the median of the absolute deviations from the sample median. The absolute deviation is the absolute difference between two elements of a data set. In this case the difference is calculated for each element of the data and the sample's median. Differences in the σ are squared which makes it sensitive to outliers, this is not the case for the MAD, which is why it is a robust measure.

Quartile and Interquartile Range

The last statistics we will use for our univariate analysis are **quartiles** and the interquartile range. Quartiles are the three points which split a data set into four equal parts. They are a type of quantiles. The first quartile (Q_1), also called lower quartile, is the 25th percentile of the data (the lowest 25%). The second quartile (Q_2) is also known as the median which is explained above. The third quartile (Q_3), the upper quartile is the point where the data is split into the lowest 75% and the highest 25%. The difference $Q_3 - Q_1$ is called the **interquartile range** (IQR) which consists of the middle 50% of the data set (leaving 25% on each side). Sometimes the IQR is also called *midspread* or *middle fifty*. Quartiles are a good method to exclude outliers from the data set and can be used as a robust measure of spread instead of the "classic" range (which is the difference between the highest and lowest value of the data set). Due to space reasons we will not show the IQR, but it can be easily calculated on-the-fly when looking at Q_1 and Q_3.

5.4 Univariate Analysis

All of the following tables are created from our data set with the base (500 hands) filter applied. For research purposes we have also taken a look at the same tables created with different filters (2500 and 10000). In short, as we have already somewhat seen from Table 5.1, the more hands the players have actually played, the more skill they have. Because of this the disparity between the groups decreased, making differentiation more difficult. The intention of using larger filters was mainly to reduce the standard deviation, which (in most cases) worked, but was not significant enough to warrant the trade-off of decreasing group disparity.

5.4.1 General Stats

VPIP	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	37.08	34.96	24.26	13.61	9.75	25.94	45.55
Breakeven	27.13	24.11	19.69	8.84	3.85	21.10	30.40
Winners	29.88	26.76	23.18	9.91	4.97	22.87	34.87
Total	34.61	31.53	23.95	12.99	8.40	24.23	42.58

PFR	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	15.78	15.94	16.67	7.09	4.51	10.95	20.03
Breakeven	16.61	16.94	17.11	4.98	2.63	14.36	19.58
Winners	17.41	17.73	19.71	5.58	3.07	14.50	20.68
Total	16.28	16.59	16.67	6.66	4.03	11.97	20.23

PFR/VPIP	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	47.61	46.94	50.00	23.86	19.01	28.67	67.22
Breakeven	65.87	75.41	25.00	22.25	10.49	50.04	82.22
Winners	63.02	70.30	50.00	22.52	13.94	46.08	81.54
Total	52.75	53.59	50.00	24.57	21.85	32.76	76.38

Hands	$\bar{x}$	$\tilde{x}$	Mo	MAD	Q_1	Q_3
Losers	4555	1383	502	750	777	3294
Breakeven	21897	5331	533	4598	1375	21990
Winners	10738	1705	527	1111	817	7002
Total	7057	1488	502	865	797	4280

The **VPIP** is one of the most common measure to used by real players to classify into the classic types of *loose* or *tight* players. The (relative) standard deviation for losers is much higher than that of breakeven or winning players (13.61 at an average of 37.08 – a relative deviation of 26.29% in comparison to 16.63% for winners). The IQR for losers also covers most of the midspread of winners which leads us to the fact that VPIP is a type of statistic where the center is one group, while the outliers form another. Due to the below average median and very low mode, the group of losers must have quite a few players with a very high VPIP in order to reach the high average of 37.08% – this also explains the high standard deviation. The last interesting thing to note is that the median for all groups is below the average which menas that more than half of the players are actually below the average, which indicates a skewed distribution.

The averages for **PFR** are not as clear cut between the groups in comparison to the averages of VPIP. One reason for this is that the value for PFR is generally lower with a total average of 16.28%. Another reason is that a small difference in PFR already denotes a big difference in playstyle. In general, winners and breakeven players are more aggressive than losers, which play more passive. This can be seen in the first quartile as well, where winners and breakeven players are around 4% higher than losers. The mode for winners is especially interesting because it is very high (19.71%) in comparison to the other values. As previously mentioned the difference between groups is not definite, but still an interesting (and somewhat useful) statistic to look at.

The next stat is the ratio between the two previous stats, namely **PFR/VPIP**. The difference in averages between losers and winners (plus breakeven players) is quite distinct. We still have a relative standard deviation of 30% to 50% depending on the group but even so this stat still provides quite a lot of information. The MAD is generally a bit lower than the SD, but still relatively high. Looking at the IQRs for losers and winners we can clearly see a difference: the upper and lower quartile for winners are much higher than for losers. From this we can conclude that winners are clustered more in the upper values between 50% and 80%, while losers are more centered towards the lower values of 40% to 60%. This is very good, but not optimal as there is still an overlap between the groups.

The number of **hands** is the last general stat which we looked at. All the values are to be viewed critically as we have already filtered our data set for this analysis by the number of hands (i.e. only players with more than 500 hands). This becomes apparent as we look at the mode for winners and losers which are quite close to our filter of 500 hands – most players in our sample have around 500 hands and not that many are above that threshold.

An interesting thing to see is the relation between the average of the groups. On average winners have twice as much hands as losers and breakeven players have twice as much hands as winners. This could be explained by multitabling or experience. Players who play a very large volume of hands are likely to play a large number of tables at the same time or play extremely frequently in order to reach that volume. In our opinion it is highly doubtable that human players are able to play at their maximum potential when grinding that many hands which explains that breakeven players have the highest number of hands on average. This leads to the next question as to why they are not losing money in the long run where the reason might be experience. Because those players play so many hands they *theoretically know* how to play but are likely failing to make crucial decisions that defines winning players (e.g. playing more aggressive). The low number of average hands for losers supports this theory as well – they lack experience because of their low volume of hands which results in losing money. Winning players are right between losers and breakeven players. They play a fairly high amount of hands but just enough to still be able to make decisions which earn them money. Please note that this interpetation only applies to human players and that the hand stat might not be that useful when facing computer poker agents.

5.4.2 Aggression

The next set of statistics are total and street-specific **Aggression Factor** and **Aggression Frequency**.

Total AF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	2.39	2.20	2.04	1.11	0.58	1.67	2.85
Breakeven	2.94	2.64	2.67	1.49	0.62	2.13	3.36
Winners	2.81	2.59	2.89	1.57	0.61	1.99	3.22
Total	2.53	2.32	2.41	1.29	0.60	1.77	2.99

Total AFq	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	48.34	48.53	50.00	7.90	5.03	43.37	53.44
Breakeven	50.09	50.40	50.75	6.53	4.45	45.44	54.43
Winners	50.17	50.25	50.00	6.91	4.34	45.81	54.50
Total	48.93	49.14	50.00	7.63	4.85	44.18	53.87

PF AF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	1.42	0.74	0.21	2.07	0.50	0.34	1.73
Breakeven	3.43	2.57	0.60	5.82	1.54	0.86	3.89
Winners	2.67	2.02	0.45	3.42	1.40	0.74	3.71
Total	1.86	0.99	0.28	2.85	0.72	0.41	2.75

PF AFq	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	15.27	15.63	10.34	6.83	4.52	10.44	19.63
Breakeven	16.51	17.05	17.60	4.94	2.62	14.25	19.61
Winners	17.24	17.70	19.62	5.51	3.14	14.39	20.59
Total	15.88	16.39	16.67	6.48	4.07	11.57	19.99

Flop AF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	2.36	1.98	2.00	1.70	0.74	1.35	2.95
Breakeven	3.39	2.84	2.52	2.82	0.96	1.93	4.01
Winners	3.12	2.68	2.00	2.62	0.95	1.78	3.71
Total	2.62	2.19	2.00	2.09	0.84	1.46	3.23

Flop AFq	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	45.11	45.45	50.00	10.56	7.18	38.16	52.54
Breakeven	49.38	50.00	50.00	9.15	6.05	43.70	55.77
Winners	49.02	50.00	50.00	9.81	6.38	42.56	55.64
Total	46.40	46.87	50.00	10.46	7.10	39.47	53.75

Turn AF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	2.40	2.17	2.00	1.37	0.62	1.60	2.86
Breakeven	2.68	2.40	3.00	1.32	0.53	1.93	3.00
Winners	2.63	2.35	2.00	1.56	0.58	1.84	3.00
Total	2.48	2.23	2.00	1.43	0.60	1.68	2.92

Turn AFq	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	50.17	50.26	50.00	9.49	6.10	44.21	56.41
Breakeven	50.53	50.00	50.00	8.06	5.32	45.06	55.53
Winners	50.61	50.41	50.00	8.57	5.26	45.24	55.77
Total	50.31	50.31	50.00	9.18	5.79	44.59	56.21

River AF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	3.55	2.92	2.00	2.75	1.02	2.05	4.28
Breakeven	2.92	2.50	3.33	1.74	0.67	1.90	3.33
Winners	3.25	2.60	2.00	2.30	0.83	1.91	3.83
Total	3.44	2.80	2.00	2.60	0.95	2.00	4.11

River AFq	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	53.51	53.81	50.00	11.57	8.28	45.44	61.90
Breakeven	49.44	47.19	50.00	10.68	6.76	41.55	56.29
Winners	51.16	50.00	50.00	11.27	7.66	43.10	58.92
Total	52.67	52.38	50.00	11.55	8.24	44.44	60.95

Overall **Aggression Frequency** is not a very good indicator for grouping players into categories. For all groups the postflop AFq is clearly at 50 on average. For postflop streets (and all groups) the mode and standard deviation are fairly close to each other at 50 and 10. Preflop AFq differs quite a lot from postflop AFq or total AFq. The main reason for this is that the preflop AFq is roughly equivalent to PFR – this also makes it the only usable AFq stat overall (and even then we already have PFR) since all other AFq values are too close together for all groups throughout all streets.

When looking at the **Aggression Factor** in general, breakeven players tend to have the highest AF on average except on the river, where losers are most aggressive. Overall losers tend to have the lowest AF and – interestingly enough – generally a lower standard deviation than winners, except at the river (all statistics we have looked at previously losers had the tendency to have the highest standard deviation). The AF for winners and losers tends to rise per street in exception for the turn where it stagnates or drops slightly. On the contrary the AF for breakeven players (slightly) decreases per street.

The first specific stat we will take a look at is the **Total Aggression Factor** or **Total AF**. One thing to note is that all group averages are very close together, the highest difference being 0.62 between losers and breakeven players. As a result the median and mode are very close together as well. These facts make total AF a bad stat to look at as we have no real way to differentiate between the groups. The IQR improves this slightly, as the winners and breakeven players are more skewed towards higher values, but even the MAD is still relatively high, making it difficult to give clear statements on group "membership".

For **Preflop AF** breakeven players have the highest group average, median, mode and standard deviation. Especially the mode value is interesting as it is significantly higher than the mode of other groups (3.22 vs 0.21 and 0.65). Overall losers have the lowest metrics. As a quick reminder: a low AF means that the players tends to play passive – calling and folding more often than betting or raising. A possible interpretation for the averages of losers and breakeven players might be that playing too aggressive or passive preflop does not pay for itself – that a balance has to be found. The average for winners being inbetween the average for the other groups supports this theory. Another thing to note is that the medians for all groups are below their respective averages, meaning that more than half of the players are actually below their group average. For breakeven players the difference between mean and median is the highest, which kind of explains the high variance.

When looking at the **Flop AF** it is quite clear that mean, median and mode for winners and losers are all quite a bit higher than their preflop counterparts. Breakeven players are more constant, although now their median is higher than their mode (it was reversed on the flop). Standard deviation however is lower for all groups while the "ranking" still remains – breakeven players have the highest, while losers have the lowest deviation.

Between preflop and flop AF, losers have the sharpest increase of averages. A possible interpretation for this is that losers tend to overestimate the strength of weak hands (e.g. a middle pair or TPWK) and play them too aggressively which results in an overall loss of chips. Still their AF is lower than that of winners or breakeven players – most likely because they tend to play a wider range of hands preflop which makes them weaker postflop. On the same line of thought winners tend to play tighter preflop (see VPIP) and thus usually have stronger hands where aggressive play is more justified and profitable. An important thing when looking at the flop AF is that the mode is quite often at 2.0 exactly which means that our sample size might not be large enough.

The AF for turn and river are not quite as usable as the preflop or flop AF as they suffer the same problem as total AF: the values are too close together (highest differences being 0.32 on the turn and 0.69 at the river). The modes are all discrete values as well which indicate that our sample size is not sufficient for the AF of later streets.

5.4.3 Showdown

WTSD	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	29.48	28.79	33.33	5.75	3.48	25.60	32.69
Breakeven	28.88	27.21	25.00	6.84	2.43	25.08	30.07
Winners	28.48	27.57	25.00	5.83	2.86	24.98	30.87
Total	29.17	28.33	33.33	5.84	3.30	25.37	32.13

WSD	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	48.76	48.91	50.00	6.69	4.25	44.48	53.09
Breakeven	53.32	53.47	50.00	5.23	3.03	50.37	56.37
Winners	54.97	54.68	50.00	6.37	3.49	51.25	58.25
Total	50.71	50.95	50.00	7.13	4.44	46.25	55.21

The average values for **Went to Showdown %** are all very close together in the 28.5 to 29.5 range. While the median and mode are probably more useful, they still have to be viewed critically, especially the mode which takes two discrete values for the group of losers and winners ($\frac{1}{3}$ and $\frac{1}{4}$ respectively). In general, losers tend to go to showdown more often than winners. The standard deviations are nearly the same for all three groups and overall WTSD is not the greatest stat to look at when considering player classification and our non-parametric metrics also support this fact.

Interestingly the mode for **Won $ at Showdown** is the same for all three groups at a value of exactly 50. Losers have a slightly lower winrate at the showdown while winners have a greater winrate as expected. We expected breakeven players to be closer to 50 but their average as well as their median are well above 50. This could be explained by the rake mechanic that the poker sites employ to earn money. Players actually have to win more than 50% of their showdowns to breakeven because in every pot they are involved in, they are paying a small rake to the poker site.

5.4.4 Stealing

Att To Steal	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	26.51	27.13	0.00	12.51	8.22	17.72	34.53
Breakeven	29.41	30.26	25.00	9.65	5.02	25.00	35.03
Winners	30.06	30.84	33.33	10.15	5.76	24.56	36.31
Total	27.63	28.70	0.00	11.94	7.41	19.93	35.17

Stole	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	53.94	54.72	50.00	9.34	5.17	48.91	59.38
Breakeven	54.94	55.57	50.00	6.74	3.08	52.38	58.40
Winners	55.48	55.56	50.00	7.78	3.85	51.59	59.19
Total	54.42	55.03	50.00	9.31	4.75	50.00	59.28

AttSB (LP)	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	25.73	26.28	0.00	12.40	8.44	16.86	34.07
Breakeven	28.70	29.72	27.27	9.58	4.88	24.69	34.54
Winners	29.56	30.45	33.33	10.20	5.98	23.87	35.85
Total	26.94	27.94	0.00	11.94	7.60	19.19	34.75

AttSB (SB)	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	29.46	28.48	0.00	16.06	10.53	17.95	38.99
Breakeven	32.07	30.53	20.00	14.46	7.69	23.81	39.53
Winners	31.85	30.84	33.33	14.01	8.11	23.35	39.47
Total	30.24	29.30	0.00	15.70	9.65	20.00	39.22

Fold to Steal	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	61.60	63.41	50.00	16.58	12.41	50.53	75.48
Breakeven	73.84	78.06	79.57	11.59	5.15	66.67	82.08
Winners	70.73	75.00	66.67	13.15	7.19	62.88	80.72
Total	64.70	67.55	50.00	16.14	11.69	54.22	78.39

Call Steal	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	31.72	30.53	50.00	17.17	13.03	17.23	43.35
Breakeven	18.22	13.19	33.33	13.02	5.67	8.77	26.75
Winners	21.35	16.52	40.00	14.43	8.15	10.04	30.94
Total	28.21	26.52	33.33	17.06	13.54	12.85	39.84

3Bet Steal	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	6.68	6.02	0.00	4.52	2.92	3.25	9.15
Breakeven	7.94	7.69	0.00	4.12	2.34	5.32	9.92
Winners	7.92	7.76	0.00	4.14	2.65	5.00	10.32
Total	7.09	6.67	0.00	4.52	2.97	3.67	9.62

3Bet LP Steal	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	6.42	5.67	0.00	4.45	2.85	3.08	8.87
Breakeven	7.62	7.28	0.00	4.09	2.28	5.07	9.60
Winners	7.64	7.45	0.00	4.14	2.69	4.66	10.05
Total	6.82	6.33	0.00	4.49	2.91	3.47	9.30

The next statistics are about stealing and steal defense which are important parts of the game. Winners tend to steal the most, while losers steal the least. The standard deviation is quite high for all groups, but the MAD is reasonable for most stats which don't require large sample sizes (such as 3Bet Steal). Interestingly the mode for losers is 0 and for winners 33.33 (or $\frac{1}{3}$) which shows a quite apparent difference between the playstyle of losers and winners. The steal success rate, namely **Stole**, does not seem to play a huge factor for group classification as all averages are very close together. In general, the first quartile is much higher for winners than for losers, which supports the fact that winning players are generally more aggressive.

As position plays a crucial role in poker, we have taken a look at the positional attempt to steal stats. Surprisingly the attempt to steal blinds as the small blind is higher than in late positions (cutoff and button) for all groups. This is quite counter-intuitive as stealing in late positions should be easier (and more profitable) than stealing in early position (as the SB). Even for winners, their steal attempts in LP are lower than their overall attempts to steal which made us doubt that the positional advantage is that significant when considering stealing. Obviously all stealing-related stats are for the preflop street only.

Statistics which measure the reactions to steals are also important to look at, these include **Fold to Steal**, **Call Steal** and **3Bet (LP) Steal**. As expected, losers fold less and call more often than winners with a significant margin of about 9%. While the standard deviation is relatively high at 12 to 16 we do believe that the "Fold to" and "Call" statistics are still quite usable. The difference between the groups is more obvious when looking at the mode and median where the disparity between losers and winners is even higher.

Unfortunately the difference in **3Bet (LP) Steal** is not quite as obvious as when looking at folding to or calling steals. All average values are quite close together with a highest difference of about 1.3% – and all modes are 0 which hints at an inadequate samplesize. We can only see small tendencies here, namely that winners are slightly more aggressive. When thinking in terms of hand range for 3-betting steals, a difference of 1.3% is only one extra type of hand which is included (or excluded) into the 3bet range (something like a KJs or AQo). The LP version of this stat shows slightly lower average values as one would expect (albeit not by much), as 3betting against a stealer in position will leave the player in a worse position postflop.

5.4.5 Raising

Next we take a look at stats which resolve around 3betting.

3Bet PF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	5.43	5.00	0.00	3.35	2.11	2.97	7.20
Breakeven	6.08	6.13	6.27	2.70	1.48	4.49	7.50
Winners	6.23	6.16	6.88	2.93	1.81	4.23	7.85
Total	5.68	5.42	0.00	3.25	2.07	3.31	7.44

3Bet F	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	13.05	11.11	0.00	11.86	7.88	3.33	19.05
Breakeven	15.61	14.12	0.00	11.09	5.89	8.78	20.38
Winners	15.90	14.40	0.00	12.30	7.73	6.78	22.22
Total	13.96	12.50	0.00	12.87	7.50	4.35	20.00

3Bet T	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	12.00	6.45	0.00	17.84	6.45	0.00	18.18
Breakeven	14.77	13.04	0.00	15.36	8.98	0.00	20.00
Winners	14.29	10.00	0.00	18.82	10.00	0.00	21.05
Total	12.76	7.69	0.00	17.57	7.69	0.00	20.00

3Bet R	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	8.88	0.00	0.00	23.09	0.00	0.00	11.11
Breakeven	9.45	2.01	0.00	18.05	2.01	0.00	13.11
Winners	10.02	0.00	0.00	25.02	0.00	0.00	12.50
Total	9.21	0.00	0.00	18.68	0.00	0.00	11.76

3Bet Total	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	5.68	5.26	0.00	3.31	2.06	3.28	7.45
Breakeven	6.32	6.31	7.41	2.75	1.52	4.70	7.68
Winners	6.49	6.41	4.71	2.92	1.80	4.51	8.10
Total	5.94	5.71	6.67	3.21	2.05	3.59	7.67

4Bet+ Total	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	7.39	6.67	0.00	4.88	3.09	3.83	10.00
Breakeven	8.20	8.23	0.00	3.95	2.36	5.61	10.38
Winners	8.33	8.18	0.00	4.47	2.94	4.94	10.97
Total	7.69	7.19	0.00	5.02	3.15	4.12	10.40

Call 3Bet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	27.86	26.18	33.33	13.43	9.26	17.48	36.26
Breakeven	19.79	17.54	16.67	9.89	5.54	12.66	25.41
Winners	22.24	19.40	25.00	11.59	6.90	13.85	28.85
Total	25.92	23.70	33.33	13.17	8.73	15.79	34.00

Fold to 3Bet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	66.14	67.86	66.67	13.69	9.37	57.42	76.47
Breakeven	73.53	75.84	77.78	9.97	5.36	68.53	80.00
Winners	70.98	73.36	75.00	11.43	6.69	64.71	79.01
Total	67.83	70.00	66.67	13.18	8.46	60.00	77.62

Call 4Bet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	44.98	44.44	50.00	22.31	15.87	28.57	60.00
Breakeven	37.71	34.01	0.00	19.17	11.15	25.00	50.00
Winners	39.26	34.78	0.00	21.89	14.78	25.00	50.00
Total	43.05	40.00	50.00	25.32	15.00	26.09	57.97

Fold to 4Bet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	41.97	42.31	0.00	21.81	16.51	25.00	57.14
Breakeven	48.19	50.00	50.00	18.39	10.00	37.50	57.79
Winners	45.78	50.00	50.00	19.58	12.50	33.33	58.33
Total	43.31	45.00	0.00	24.88	15.00	27.27	57.14

Similar to AF the per street version of this statistic is more valueable than its total value. Even with our large sample size the mode for all postflop streets is 0 which shows the enormous sample size requirement for these stats which is why the turn and river variant are not that useful in our analysis. Variance and standard deviation are also very high throughout all postflop streets which further limits the usability - for the preflop 3bet stat variance and standard deviation is lower, but still high if we consider the lower average values. The average for **3Bet PF** is signifcantly lower (between 4% and 10% difference) in comparison to the postflop 3bet percentages. Another thing that can be seen is that the average decreases per street for the postflop stats. All in all the statistics are very limited in their usability due to reasons explained above, also if looking at each stat individually the differences between the groups are not very distinct either.

Another aspect worth looking at are reactions to 3betting and 4betting as they are often crucial decisions that can determine whether you are winning or losing money. All stats concern all streets, preflop included.

First we take a look at the stats related to **3Bets**. On average, losers call more often and the difference between winners and losers is quite large at a relative margin of nearly 25% (5% in raw numbers). Mode and median show this difference even more clearly. The analogous argument can be made for the **Fold to 3Bet** stat – losers overestimate their hand strength and thus fold less to 3bets.

Breakeven players tend to be very close to winners, but it seems that playing too tight is making the difference as they have the lowest call 3bet and the highest fold to 3bet average. Variance and standard deviation still have to be considered as they are quite high, the mode is often at discrete values which hints at an insufficient sample size. Overall, in our opinion these two stats are worth looking at, but relying on them is not advised as they require a significant sample.

The stats concerning **4bets** take no consideration of past actions so they just use the raw number of 4bet calls, folds and their respective opportunities. As the opportunities to 4bet are even rarer than for 3bet these stats are only good to see some tendencies (the variance and standard deviation is just too high). Things we can see are just as one would have expected: losers call more often and fold less, while it is the exact opposite for winners. Again, breakeven players are even tighter than winners and have the lowest call and highest fold percentages.

Generally speaking all raise-related statistics have very limited use as they require extremely large sample sizes, and even then the usability is questionable, as we can see that even the non-parametric metrics like MAD are still extremely high.

5.4.6 Continuation Bets

Next up are the statistics related to continuation bets.

CBet F	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	65.30	66.10	66.67	15.39	10.37	55.56	76.27
Breakeven	66.97	67.82	71.43	13.30	8.44	58.56	75.81
Winners	66.87	67.78	66.67	13.72	8.69	58.49	76.00
Total	65.81	66.67	66.67	15.04	9.80	56.52	76.19

Fold to F CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	48.90	48.84	50.00	10.60	6.86	41.98	55.71
Breakeven	51.03	50.75	50.00	9.95	6.15	45.03	57.34
Winners	48.98	49.15	50.00	10.35	6.37	42.83	55.56
Total	49.02	49.06	50.00	10.57	6.68	42.31	55.63

CBet T	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	55.01	52.94	50.00	20.51	13.73	40.54	68.45
Breakeven	48.95	47.47	50.00	16.46	7.91	40.43	57.14
Winners	50.73	48.89	50.00	17.97	10.39	39.71	60.00
Total	53.53	50.00	50.00	21.00	12.50	40.00	66.67

Fold to T CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	36.75	36.36	50.00	16.16	10.04	26.30	46.15
Breakeven	36.01	36.89	50.00	14.18	8.32	28.09	44.66
Winners	36.21	35.80	0.00	16.51	10.76	25.00	46.15
Total	36.57	36.36	50.00	18.17	10.02	26.09	46.15

CBet R	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	55.80	54.55	100	24.16	20.45	37.50	75.00
Breakeven	53.57	53.13	100	19.99	13.13	40.74	66.67
Winners	54.79	52.90	100	22.17	13.78	40.00	69.30
Total	55.41	53.88	100.00	29.46	18.17	39.29	75.00

Fold to R CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	43.26	44.44	0.00	23.56	19.44	25.00	60.00
Breakeven	42.35	44.44	0.00	20.16	12.70	30.00	56.00
Winners	41.09	40.00	0.00	24.72	20.00	16.67	57.29
Total	42.63	42.86	0.00	29.94	17.86	23.08	60.00

Call F CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	38.10	38.00	33.33	11.02	7.35	30.77	45.45
Breakeven	34.69	34.61	25.00	9.64	6.04	28.81	41.02
Winners	36.39	36.25	33.33	10.59	6.61	29.66	42.86
Total	37.47	37.33	33.33	11.10	7.11	30.30	44.44

Raise F CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	13.00	12.00	0.00	7.71	4.84	7.45	17.20
Breakeven	14.28	13.33	0.00	7.28	3.99	9.77	17.74
Winners	14.63	13.83	0.00	8.04	4.48	9.51	18.48
Total	13.51	12.53	0.00	7.97	4.72	8.06	17.58

Call T CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	49.01	50.00	50.00	16.39	10.00	39.02	60.00
Breakeven	47.75	48.64	50.00	14.74	8.51	40.00	57.14
Winners	48.26	50.00	50.00	17.16	10.00	38.10	58.77
Total	48.75	50.00	50.00	19.04	10.00	38.71	60.00

Raise T CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	14.24	11.91	0.00	13.01	7.14	5.37	20.00
Breakeven	16.23	13.83	0.00	13.44	6.18	8.39	20.89
Winners	15.53	13.04	0.00	15.04	7.48	5.51	20.45
Total	14.68	12.50	0.00	14.21	7.50	5.56	20.00

Call R CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	46.98	50.00	0.00	24.47	16.67	28.57	66.67
Breakeven	46.37	44.83	0.00	22.22	13.58	33.33	60.00
Winners	46.82	48.54	0.00	25.09	18.13	25.00	66.67
Total	46.91	50.00	0.00	30.36	16.67	28.57	66.67

Raise R CBet	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	9.76	0.00	0.00	20.86	0.00	0.00	14.29
Breakeven	11.28	5.00	0.00	21.71	5.00	0.00	14.29
Winners	12.08	0.00	0.00	23.81	0.00	0.00	16.67
Total	10.46	0.00	0.00	18.69	0.00	0.00	14.29

It is very interesting to see that all group averages across all streets are really close together in terms of continuation betting. The most noticable difference is the second continuation bet at the turn where losers are around 10% higher in relation to breakeven or winning players (or around 5% in raw percentage). Fold to CBet behaves very much like CBet itself – very close averages across all streets. Both **Cbet X** as well as **Fold to X CBet** are lowest at the turn and highest at the flop with the average for the river being inbetween. The standard deviation and variance for the river are quite extreme and the mode is at discrete values for almost all stats as well, so the sample size is most likely not sufficient to draw a detailed conclusion. An interesting observation is that the mode of Fold to T CBet is 0 for winners, while it is at 50 for breakeven and losing players. This could be explained by the fact that good players will usually not call a flop continuation bet with a weak hand (or a flush draw for example) leaving them vulnerable to aggression on the turn which would force them to fold should they fail to improve their hand strength.

5.4.7 Donk Bet

Donk F	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	15.44	11.01	0.00	14.36	7.31	4.84	21.66
Breakeven	11.27	8.12	0.00	10.47	5.10	4.13	14.62
Winners	11.63	8.33	0.00	11.51	5.12	4.04	15.38
Total	14.19	10.00	0.00	13.66	6.50	4.55	19.70

Donk T	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	10.31	5.92	0.00	13.48	5.92	0.00	14.29
Breakeven	8.65	5.83	0.00	10.16	4.93	1.35	11.90
Winners	8.47	4.66	0.00	13.09	4.66	0.00	11.11
Total	9.72	5.56	0.00	12.78	5.56	0.00	13.33

Donk R	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	19.05	14.29	0.00	18.67	12.47	3.33	28.00
Breakeven	14.86	10.07	0.00	17.17	7.82	3.31	20.00
Winners	15.39	9.30	0.00	19.27	9.30	0.00	22.58
Total	17.84	12.63	0.00	19.31	12.63	0.00	25.58

The **Donk Bet** is one of the few stats where we expected to see a clear difference between winners and losers as it is considered a bad play by many players. While a difference can be seen, it is not as significant as we expected it to be. The relative difference between the average values for **Donk Flop** is nearly 50%, but when looking at the median or mode we can deduct that the difference is in fact not that significant. Reasons for this might be a too small sample size or that a donk bet can be used moderately as a tool to be successful at the table (e.g. donk betting in order to pressure the preflop aggressor). Comparing the quantiles we can see that anyone with a Donk F value between 16% and 22% is more likely to be classified as a loser rather than a winner.

5.4.8 Float

Float Total	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	47.95	47.27	50.00	15.33	10.20	37.50	57.89
Breakeven	48.12	47.78	50.00	13.13	8.36	39.64	56.74
Winners	48.42	47.86	50.00	14.56	9.29	38.89	57.27
Total	48.09	47.44	50.00	15.37	9.78	38.07	57.74

Float F	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	46.11	45.65	50.00	18.68	12.68	33.33	59.14
Breakeven	49.27	50.00	50.00	17.82	11.73	36.84	60.61
Winners	48.08	48.68	50.00	18.13	12.14	35.39	60.00
Total	46.80	46.67	50.00	19.69	13.33	33.33	59.82

Float T	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	51.69	50.00	50.00	21.40	16.67	36.36	66.67
Breakeven	49.88	50.00	50.00	18.35	11.54	37.62	60.85
Winners	51.22	50.00	50.00	20.97	14.29	36.41	66.67
Total	51.48	50.00	50.00	23.26	15.91	36.36	66.67

Float R	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	45.17	43.75	50.00	22.79	16.25	28.57	60.40
Breakeven	41.13	39.62	0.00	20.24	11.15	28.66	52.72
Winners	44.19	42.11	50.00	22.52	14.11	29.03	57.79
Total	44.72	42.86	50.00	26.15	15.59	28.57	60.00

Although the **Float** is considered a more advanced move on the poker table in terms of skill and strategy, the differences between the groups are surprisingly non-existent. A possible explanation for this might be that the first two actions which define a float are (1) calling a preflop raise and (2) betting, when the preflop aggressor does not make a continuation bet – both can be considered weak play and thus losers are likely to make these kind of plays as well which influences this stat. Overall we do not see how this stat can help in categorizing players, not only because of the previously mentioned insignificant differences between the groups, but also because of the high standard deviation of all statistics.

5.4.9 Limping

Limp	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	18.73	14.73	0.00	17.19	12.68	3.01	29.81
Breakeven	7.64	1.79	0.00	11.46	1.46	0.67	11.12
Winners	9.28	2.68	0.00	12.27	2.35	0.85	14.82
Total	15.58	10.10	0.00	16.37	9.24	1.49	25.58

Limp/Call	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	54.61	58.00	0.00	19.82	14.73	41.36	71.43
Breakeven	40.61	40.00	0.00	21.60	18.33	22.22	58.82
Winners	45.17	46.81	0.00	23.20	19.86	25.00	65.03
Total	51.42	54.17	0.00	25.04	16.88	34.21	69.57

Limp Behind	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	20.95	20.00	33.33	12.49	9.21	10.87	29.30
Breakeven	12.86	9.33	0.00	10.43	5.96	4.94	18.97
Winners	14.39	11.54	0.00	10.70	6.82	5.88	20.95
Total	18.75	17.24	0.00	12.45	9.30	8.33	27.03

Limp/Fold	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	41.40	37.29	100	23.29	14.56	24.13	54.49
Breakeven	55.92	55.56	100	24.31	19.44	36.60	75.00
Winners	51.44	50.00	100	25.68	20.59	30.10	71.43
Total	44.78	41.18	100	25.78	17.04	25.49	61.08

Raise Limpers	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	17.06	15.87	0.00	9.71	6.03	10.10	22.37
Breakeven	16.90	16.52	0.00	7.06	4.52	12.06	21.07
Winners	17.79	17.39	0.00	7.91	5.13	12.24	22.50
Total	17.26	16.39	0.00	9.40	5.73	10.84	22.32

Limp/Raise	$\bar{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	3.98	1.72	0.00	8.01	1.72	0.00	5.48
Breakeven	3.47	0.00	0.00	13.74	0.00	0.00	3.66
Winners	3.40	0.00	0.00	10.41	0.00	0.00	4.00
Total	3.80	1.04	0.00	7.36	1.04	0.00	5.10

Limp, **Limp/Call**, **Limp/Fold** and **Limp Behind** (also known as *Limp With Previous Limpers*) explicitly show the characteristics of losing players as we expected. Losers limp almost twice as much as winning players and are more prone to call on the flop without laying their hand down after they limped. Still, the majority of players tend to avoid limping, and if they do they usually fold on the flop, which becomes apparent when looking at the mode values. The majority of losers still limp behind a third of the time though. In terms of **Raise Limpers** or **Limp/Raise** (the former indicates a preflop limp, while the latter means that the player raised on the flop after limping preflop) there is not much difference between the groups. For all limping-related statistics the standard deviation is quite high and basically all modes are at discrete percentage values (0, $\frac{1}{6}$, $\frac{1}{3}$ and 1)

5.4.10 Cold Call

CC 2Bet PF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	22.33	20.00	16.67	12.56	8.01	12.86	29.49
Breakeven	13.76	11.98	0.00	8.40	4.16	8.28	17.15
Winners	15.79	13.58	0.00	9.36	4.84	9.56	20.00
Total	20.11	17.39	16.67	12.06	7.08	11.33	26.67

CC 3Bet+ PF	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	7.24	4.76	0.00	8.33	4.49	1.14	10.34
Breakeven	3.23	1.64	0.00	4.86	1.39	0.69	4.02
Winners	4.07	2.04	0.00	6.17	2.04	0.00	5.48
Total	6.17	3.65	0.00	7.70	3.65	0.61	8.73

As a reminder: if a player has not yet put money in the pot (including blinds) and calls a bet or raise, it is called cold call. Looking at above tables confirms our intuition that weak players will cold call more than winners. This is very apparent when looking at the mode where the difference is extreme (0 vs. 16.67 for 2Bet CCs), but the standard deviation is still quite high nonetheless.

5.4.11 Miscellaneous

For the last set of statistics we will look at those which did not quite fit into any of the previous categories.

Bet Total	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	38.97	38.35	25.00	10.14	6.26	32.23	44.81
Breakeven	40.31	40.28	37.87	8.40	5.47	34.20	44.97
Winners	40.18	39.79	40.00	8.64	5.48	34.52	45.43
Total	39.36	38.90	50.00	9.69	6.03	32.91	44.99

XR Total	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	8.63	7.91	0.00	4.96	3.08	5.08	11.32
Breakeven	8.63	7.88	0.00	4.18	2.42	5.87	10.76
Winners	9.14	8.48	0.00	4.78	2.79	5.94	11.58
Total	8.77	8.09	0.00	5.00	2.96	5.36	11.38

Call vs BTN Open	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	31.40	30.86	33.33	17.81	13.49	16.67	43.72
Breakeven	17.33	11.45	0.00	13.81	5.89	7.44	25.73
Winners	20.56	15.47	25.00	15.33	9.07	8.26	31.03
Total	27.73	26.32	33.33	17.82	14.47	11.30	40.00

Bet F (limp pot)	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	34.19	32.14	33.33	15.85	10.72	22.60	44.12
Breakeven	28.50	25.94	0.00	13.29	7.86	19.17	35.26
Winners	30.25	27.87	33.33	14.47	8.97	20.00	38.61
Total	32.83	30.56	33.33	15.70	10.19	21.52	42.42

Win %	$\overline{x}$	$\tilde{x}$	Mo	σ	MAD	Q_1	Q_3
Losers	23.69	22.69	25.86	6.40	4.20	18.91	27.45
Breakeven	19.94	18.98	18.63	4.34	2.61	16.76	22.46
Winners	21.81	20.90	18.78	5.08	2.98	18.25	24.46
Total	23.00	21.90	18.78	6.07	3.77	18.49	26.37

The first thing we notice when looking at **Bet Total** is that the average is basically much the same for all three groups and thus makes it a mediocre stat to look at when trying to categorize players into winners and losers. The mode is the only stat which lets us see a tendency – losers and breakeven players bet less often than winners.

Other stats which require an extreme amount of hands to be accurate but might still be interesting is **XR Total**. For our sample size the mode is still 0 across all groups which just shows how big the sample size has to actually relay some information. Again, with the median and average, we can only see a small tendency – winners are slightly more aggressive than losers or breakeven players. The relative standard deviation is extremely high at around 50% so this observation has to be taken with a huge grain of salt.

Call vs BTN Open is a quite good statistic to look at, although the SD and MAD are relatively high. Judging from the quantiles most players with a value higher than 30% can be considered losers. Calling when the button opens shows that the player is probably not positionally aware or has an extremely strong hand.

The last statistic is **Win%** which in theory is not really useful to see at the table. All averages are very close together in the 20 to 24 percentage range. It is very interesting to see that losers have the highest average win percentage while breakeven players have the lowest. When looking at the mode and median it becomes apparent that most players have a win percentage of below 20. As expected though, this statistic is not very helpful in categorizing players.

5.5 Machine Learning

In this section we use different methods of machine learning on our data and evaluate the results. We will however, not explain how the algorithms actually work – for more information on the implementation please refer to the Weka documentation [45] (and for information on classifiers [44]). Generally speaking, machine learning involves using algorithms and computers to analyze data in order to gain knowledge from it. This can, of course, be done in various ways - for example by classification or clustering which we will put our focus on. Classification is the process of assigning classes to instances of a dataset. To measure the accuracy we can also compare the results to an arbitarily pre-defined class, such as our class of breakeven players, winners and losers. Such an approach is called *supervised learning*, since we provide a pre-defined class for which the classifier tries to find an appropriate mapping based on the other attributes.

Clustering is another approach, where the data is ”drawn” into an n-dimensional graph and then separated into clusters. It can be both supervised (with a pre-defined class attribute), as well as unsupervised, where the clusterer tries to find an appropriate output on its own.

Cross-validation is another thing worth mentioning when talking about machine learning. It describes the process of splitting the data into n folds, for which the classifier is then trained separately. As an example we used a 10-fold cross-validation where it was possible, meaning that the data was split into ten equal parts for the training.

5.5.1 Datasets

We prepared six different data sets for this experiment with three different categorizations and two different filters each. The first pair of datasets uses our standard classification of losers, winners and breakeven players. Again, winners are defined as players who have a BB/100 ≥ 1, losers with a BB/100 ≤ -1 and all players in between as breakeven. These datasets are called LBW 500 and LBW 10000, while the number stands for the applied minimum hand filter. The second pair of datasets is basically the same as the first one, but all players who are considered breakeven are removed - we call them LW1 500 and LW1 10000. This is mainly done with the intention of observing a more apparent split when clustering. The last pair only uses two classes: winners and losers. Winners are defined as players with a BB/100 ≥ 0 while losers are all players with a BB/100 below 0. These datasets will be called LW 500 and LW 10000. We choose to use the filters of 10000 because of the trade-off between noise and number of instances.

The basic dataset (LBW 500) was used most frequently for all of our tests. The general motivation behind the different datasets is for the purpose of comparison. Other datasets were used if we felt that the performance could increase significantly.

Dataset	Losers		Breakeven		Winners		Total
LBW 500	7401	67.46%	482	4.39%	3088	28.15%	10971
LW1 500	7401	70.56%	–	–	3088	29.44%	10489
LW 500	7658	69.80%	–	–	3313	30.20%	10971
LBW 10000	643	44.53%	188	13.02%	613	42.45%	1444
LW1 10000	643	51.19%	–	–	613	48.81%	1256
LW 10000	745	51.59%	–	–	699	48.41%	1444

Table 5.2: Composition of the datasets

The percentage values serve as an indicator for *worst case* accuracy, which is when a classifier classifies all data as part of the group which has the highest share of the total sample. In this case, this is the group of losers which make up the majority in the samples. Accuracy in general is the ability of the classifier to correctly assign the each instance to the pre-defined class - the higher the accuracy, the more classes were correctly assigned by the classifier.

Each instance of the dataset is a vector of 69 attributes, which are the statistics described in Section 4. As previously mentioned in Section 5.2, we excluded the *BB/100* and *WSD* stats, because they are heavily correlated to the class attribute, leaving a total of 67 attributes. Specifically, our dataset consists of following attributes (including WSD and BB/100):

Hands	Total AFq	Fold to Steal	Fold to F CBet	Limp	CC 2Bet PF
BB/100	PF AFq	Call Steal	Fold to T CBet	Donk F	CC 3Bet+ PF
Class	Flop AFq	3Bet Steal	Fold to R CBet	Donk T	Bet Flop (limp pot)
VPIP	Turn AFq	3Bet LP Steal	Fold to 3Bet	Donk R	Limp Behind
PFR	River AFq	3Bet Total	Fold to 4Bet	Raise F CBet	Raise Limpers
PFR/VPIP	WTSD	3Bet PF	4Bet+ Total	Raise T CBet	Limp/Call
Total AF	WSD	3Bet F	Bet Total	Raise R CBet	Limp/Fold
PF AF	Att To Steal	3Bet T	XR Total	Float Total	Limp/Raise
Flop AF	Att To Steal (LP)	3Bet R	Call 3Bet	Float F	Call F CBet
Turn AF	Att To Steal (SB)	CBet F	Call 4Bet	Float T	Call T CBet
River AF	Stole	CBet T	Win Pct	Float R	Call R CBet
		CBet R	Call vs BTN Open		Call F CBet IP

Table 5.3: Attributes of the datasets

5.5.2 Classification

For classification we used multiple algorithms: logistic regression, support vector machines, rule learners and decision trees. Whenever possible, we used 10-fold cross validation. For a few tests we also trained the classifiers on a 10000 dataset, while using the respective 500 dataset as test data – note that this might not be optimal as the data sets are not disjunct (meaning that the first dataset contains entries from the second dataset). In the following paragraphs we will go into detail for the results of each tested algorithm. The field *Run* just serves as an identification and has nothing to do with how many runs were made. The idea behind multiple algorithms is again for comparative purposes. Some algorithms might be more suited for this type of data, while others might perform worse.

Logistic Regression

Weka has two implementations for logistic regression classification, namely **Logistic** [39] and **SimpleLogistic** [42], which we used both. The following two tables show the results of our test runs.

Run	Dataset	Ridge	Accuracy
1	LWB 500	1.00E-04	70.0939%
2	LWB 500	1.00E-08	70.0574%
3	LWB 500	1.00E-12	70.1303%
4	LW1 500	1.00E-12	73.2482%
5	LW 500	1.00E-12	72.2085%
6	LW1 10000	1.00E-12	64.7293%
7	LW1 10000	1.00E-12	71.4177%
8	LW 10000	1.00E-12	62.3269%
9	LW 10000	1.00E-12	70.2853%

Table 5.4: Results for the **Logistic** classifier. *Ridge* is a parameter for the used ridge estimator.

Logistic: [39] For all runs the parameters *useConjugateGradientDescent* and *MaxIts* were *False* and *-1* respectively. Table 5.4 shows the results of our runs. After a few initial runs we decided to let the *Ridge* parameter stay at 1.00E-12 while testing different datasets. Runs 7 and 9 used the shown dataset as training set, while using their 500 counterparts as test sets. While the accuracy in run 4 and 5 are the highest the worst-case accuracy is 70.56% for the datasets, which means that the classifier only managed to improve the accuracy by roughly 3%. However, looking at run 6 to 9 we can see an improvement in accuracy between 13 and 20%. For the runs 7 and 9 we used the respective 10000 dataset to train the classifier, while using the same 500 dataset as test data. Runs 6 and 8 were regular 10-fold cross-validated runs. A 20% improve over the worst-case accuracy is quite significant.

Run	Dataset	I	H	Accuracy
1	LBW 500	500	1	69.6564%
2	LBW 500	250	1	69.5288%
3	LBW 500	0	1	68.8269%
4	LBW 500	250	10	69.5288%
5	LBW 500	250	100	69.5288%
6	LW1 500	250	1	72.6571%
7	LW 500	250	1	71.6708%
8	LBW 10000	500	1	54.8476%
9	LBW 10000	250	1	54.9169%
10	LBW 10000	0	1	55.8864%
11	LW1 10000	500	1	62.8185%
12	LW1 10000	250	1	62.8981%
13	LW1 10000	0	1	62.9777%
14	LW 10000	500	1	61.4958%
15	LW 10000	250	1	61.3573%
16	LW 10000	0	1	59.8338%

Table 5.5: Results for the **SimpleLogistic** classifier. *I* is the number of maximum iterations, and *H* is the parameter for the heuristic stop. An *I* of 0 means that the algorithm will stop automatically.

SimpleLogistic: All runs in Table 5.5 used the following parameter configuration: *errorOnProbabilities=False, useAIC=False, useCV=true* and *weightTrimBeta=0*. The parameters *I* and *H* are number of iterations and heuristic stop. We tried changing both *H* and *I* while keeping one constant, which can be seen in runs 1 to 5. Since *H* did not change the accuracy in any way, we decided to keep it at 1 for all further runs and try different datasets instead. Overall we can see an increase of accuracy in between 2% (run 1-5) and 11% (run 11-13) compared to the worst-case accuracy.

Support Vector Machines

Weka offers two implementations of SVMs, the first one uses the **LibSVM** library [6, 38] and the second one, called **SMO**[43], uses a sequential minimal optimization algorithm to train a support vector classifier. We tried linear as well as polynomial kernels of different degrees. As a side note we would like to mention that the building time of the tested support vector classifiers was extremely long, even on a fairly powerful machine[5] – some would take more than 10 hours to build.

LibSVM: In order for LibSVM to run we had to remove all missing values from our dataset. For this we used the Weka filter *ReplaceMissingValues* which replaced all missing values with the mean value of the sample. For all runs we used following fixed parameters: *normalize=True, loss=0.1, epsilon=0.001, coefficient=0, shinkHeuristic=True, probabilityEstimate=False* and we used the C-SVC as our SVMType. Run 3 classified nearly all players as losers and run 4, 6 and 7 classified all players as losers. The result of run 5 is even worse than the accuracy in our defined worst-case (classifying all players as losers). For run 13 and 14 we used the shown sets as training sets, while the 500 versions of the sets were used as test sample. The results may seem high, but the worst-case accuracy for LW1 and LBW 500 are 70.56 and 67.46 respectively.

Run	Dataset	C	D	E	G	K	Accuracy
1	LBW 500	1	–	0.001	0	L	67.6511%
2	LBW 500	10	–	0.001	0	L	67.9610%
3	LBW 500	100	–	0.001	0	L	67.5052%
4	LBW 500	1	3	0.001	0	P	67.4597%
5	LBW 500	10	3	0.001	0	P	56.1845%
6	LBW 500	1	3	0.001	0.001	P	67.4597%
7	LBW 500	10	3	0.001	0.001	P	67.4597%
8	LBW 500	50	5	0.001	0.1	P	67.3867%
9	LW1 10000	10	3	0.001	1	P	63.6943%
10	LW1 10000	20	3	0.001	1	P	63.1369%
11	LW1 10000	5	3	0.001	1	P	56.0510%
12	LW1 10000	7.5	3	0.001	1	P	64.2516%
13	LW1 10000	7.5	3	0.001	1	P	70.6550%
14	LBW 10000	7.5	3	0.001	1	P	67.7149%

Table 5.6: Results for the **LibSVM** classifier
C is the cost, D is the degree of the polynomial kernel and E is the tolerance of the termination criterion. G is gamma and a coefficient used in the polynomial kernel (when G=0, 1/max_index is used instead). K denotes the used kernel, while L is linear and P is polynomial.

Run	Dataset	M	N-C	K	E	Accuracy
1	LBW 500	T	F	Poly	1	68.5717%
2	LBW 500	T	F	Poly	2	67.8790%
3	LBW 500	T	T	Poly	2	67.8790%
4	LBW 500	F	T	Poly	2	69.3282%
5	LBW 500	F	T	N.Poly	2	67.4688%

Table 5.7: Results for the **SMO** classifier. The parameter *M* denotes whether the algorithm should fit a logistic model to the SVM output or not. *N-C* stands for *no-checks*. *K* is the used kernel type, while *E* is the used exponent.

SMO: The tests with SMO were all done with *complexity=1, epsilon=1.0E-12, filterType=Normalize, CVFold=10* and *toleranceParam=0.001*. We only did a few runs because the results of the previous SVC were not very promising and the SMO was basically the same thing, method-wise.

[5] Intel Core i5 3570K @ 4.4 GHz and 8GB RAM

C4.5 Decision Tree

Run	Dataset	M	C	Accuracy
1	LBW 500	100	0.1	68.5535%
2	LBW 500	150	0.1	68.9181%
3	LBW 500	200	0.1	68.6720%
4	LBW 500	100	0.01	68.5170%
5	LBW 500	100	0.5	68.6355%
6	LBW 500	100	0.25	68.7175%
7	LW1 500	150	0.1	71.7609%
8	LW 500	150	0.1	70.6226%
9	LW 10000	150	0.1	69.0730%

Table 5.8: Results for the **J48** classifier. The parameter *M* is the minimum number of occurences per leaf and *C* is the confidence factor.

J48: All runs were done with fixed parameters: *binarySplits=False, numFolds=3, unpruned=False, subtreeRaising=True, collapseTree=True, useMDLcorrection=True, useLaplace=False* and *reducedErrorPruning=False*. *M* is the number of minimum occurences per leaf and *C* is the confidence factor. We also tried more configurations than shown in Table 5.8, like using different datasets as well as lower *M* values and disabling pruning, but the results were of similar nature, not exceeding the worst-case accuracy significantly. In run 9 we used the LW 10000 to train the classifier, while using the LW 500 dataset as the test set. The accuracy is only 1-2% better than the worst-case accuracy of the LW 500 set.

Rule Learner

Run	Dataset	F	N	O	Accuracy
1	LBW 500	3	0.001	2	68.5079%
2	LBW 500	3	0.01	2	68.5079%
3	LBW 500	3	0.1	2	68.5079%
4	LBW 500	3	1	2	68.5079%
5	LBW 500	3	10	2	68.1615%
6	LBW 500	3	100	2	68.6902%
7	LBW 500	3	250	2	68.0430%
8	LBW 500	3	500	2	67.4414%
9	LBW 500	3	1000	2	67.4597%
10	LBW 500	3	100	1	67.9940%
11	LBW 500	3	100	10	67.9428%
12	LBW 500	3	100	100	67.8881%
13	LW1 10000	3	100	2	60.3503%
14	LW 10000	3	100	2	60.5263%

Table 5.9: Results for the **JRip** classifier. *F* is the number of folds (in addition to cross-validation), *N* is the minimum total weight of the instances in a rule and *O* is the number of optimization iterations.

JRip: All runs were done with the fixed parameters: *binarySplits=False, numFolds=3, unpruned=False, subtreeRaising=True, collapseTree=True, useMDLcorrection=True, useLaplace=False* and *reducedErrorPruning=False*. *N* is the minimum total weight of the instances in a rule while *O* are the number of optimization iterations. *F* is the number of folds (in addition to our 10-fold cross validation). In runs 1 to 9 we tested different N's while keeping O constant. For run 10 to 12 we picked the "best" N while varying O. As we can see for the dataset LBW 500, JRip only managed to predict the classes roughly 1% better than our worst-case. For the smaller samples used in run 13 and 14 the accuracy gain is slightly higher (around 9%).

Classification Summary

The results of the classification algorithms were mixed. In general the accuracy gain is poor (between 1% and 4%) when using the larger dataset (with a 500 hand filter). Using the smaller dataset (with a worst-case accuracy of around 50%) the increase in accuracy is higher overall. Still, the gain *somewhat* depends on the used algorithm. Algorithms of the *logisitic regression* family for example, can show overall increases up to 20% (71% compared to the 51% worst-case accuracy), while rule learners seem to cap out at around 10%. Another factor which has to be considered is computational cost. Classifiers which use support vector machines are really costly to run on this kind of data. Some of our test runs with SVM classifiers ran for several hours before completion. In contrast, regression algorithms or rule learners ran for a few minutes at most.

5.5.3 Clustering

Clustering is an approach which can be visualized quite well. The basic idea is to plot the observations into a n-dimensional graph with one dimension for each attribute. After doing this the points are grouped by their location in the space. Visualizing a n-dimensional room is difficult, but methods like the principal component analysis (PCA) exist to bypass this problem. The goal of a PCA is to reduce the dimensionality of the data by finding of linearly uncorrelated principal components, which are linear combinations of existing attributes.

In terms of clustering algorithms we used **K-Means Clustering** [41] in conjunction with a **Principal Component Analysis (PCA)** [40] for better visualization. The goal of a PCA is to create linearly uncorrelated variables (called principal components), which are linear combinations of existing variables, while the number of resulting principal components is less or equal than the original number of variables. With the PCA, we reduced the dimensionality of our data to a maximum of 5 attributes which could be made up of arbitary linear combinations of all existing attributes. The variance covered by the PCA was 95%. We used the LW 500 dataset and ran the algorithm for 2 clusters.

The result can be seen in Figures 5.5 and 5.6. As we can see in Figure 5.5 there are principal components which are good at covering a large amount of variance for different clusters. On the right however, we can also see that the overlapping of the cluster assignments for different principal components is still very present. Figure 5.6 shows the class to cluster distribution. As we can see, both classes have a pretty equal amount of instances from both cluster assignments, so the clustering did not really manage to model our pre-defined classes of winner and loser.

We also tried different set-ups (using more principal components and/or more clusters), but the results were all very similar. Including, or excluding, the pre-defined class attribute also had a large influence on the results. In one of our test setups we ran the algorithm with $k = 4$ and tested it with both included (training set) and excluded (classes to cluster evaluation) class attribute. When including the clustering shows a relatively good separation where cluster 0 and 1 mainly consisted of winners and breakeven players, while cluster 2 and 3 was made up of losers and breakeven players. However, excluding the class attribute and using a class to cluster evaluation, the results were quite different, as all clusters had more or less equal parts of winners, losers and breakeven players.

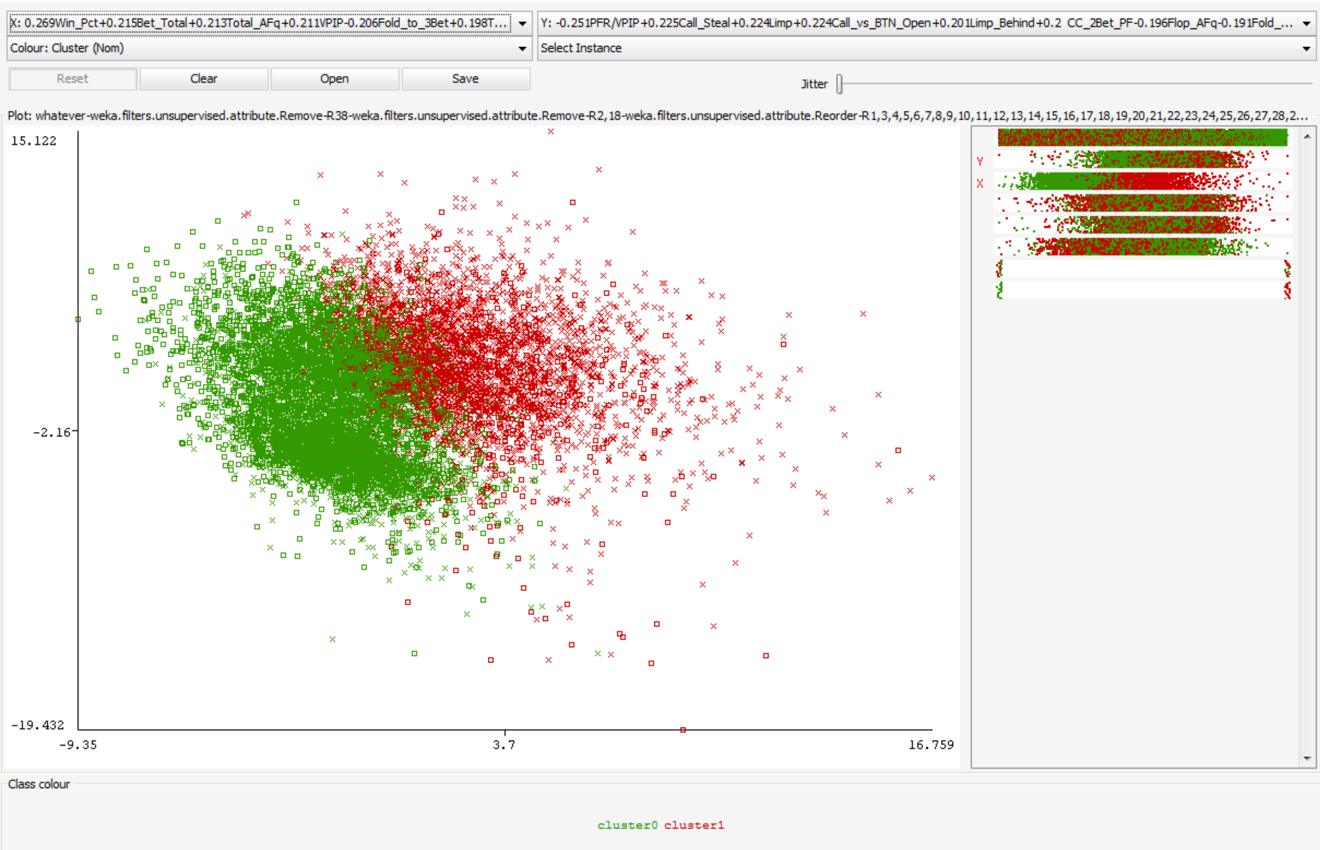

Figure 5.5: K-Means Clustering for 2 clusters on LW 500 with previous PCA

The x- and y-axes are different principal components which are linear combinations of existing attributes as seen on the top of the image. On the right is the cluster distribution for other principal components where the overlap is hinted at. The two bars on the bottom are the *class* and *cluster* distributions, hence the wide separation.

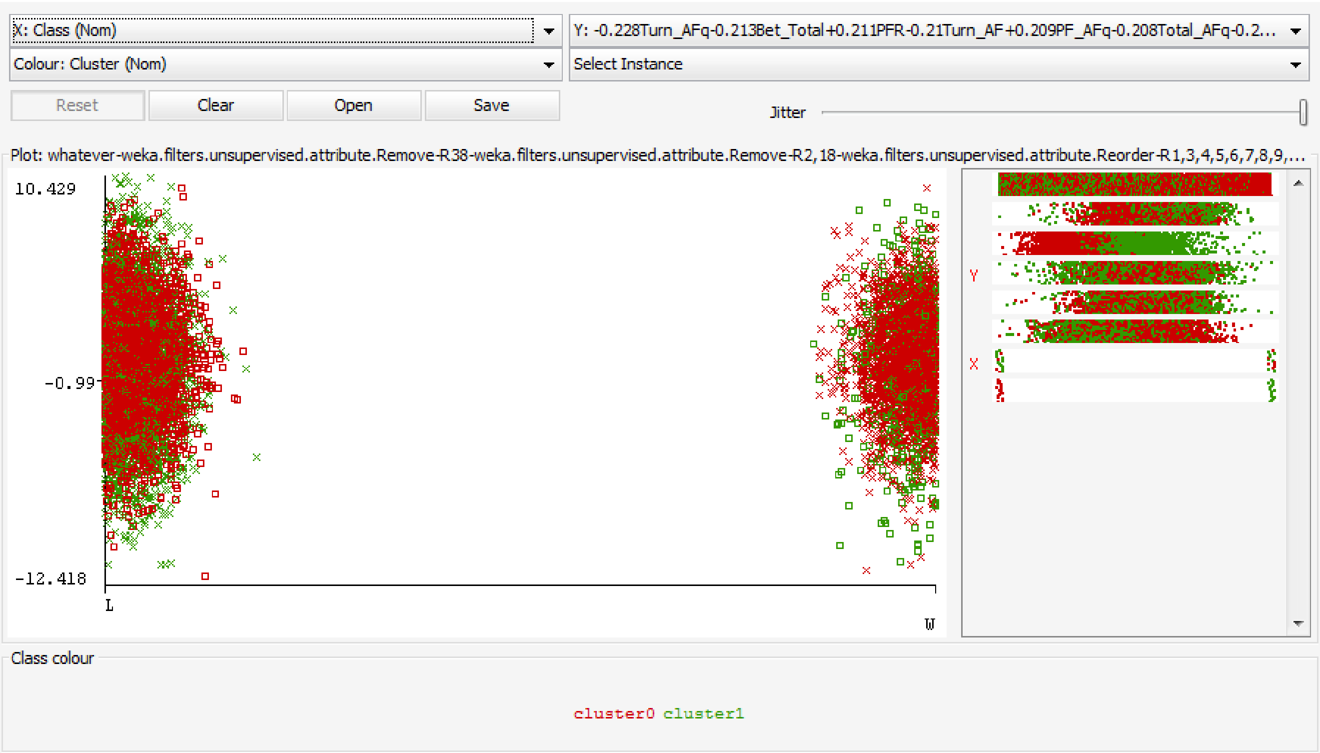

Figure 5.6: K-Means Clustering on LW 500, showing the cluster assignments and classes. Jittering is maximized to increase visibility of the distributions. The x-axis shows the original classes *Loser* and *Winner*, while the y-axis shows a principal component. In addition the points are colored by their cluster assignment. We have just taken an arbitrary principal component as the result is similar for all of them: the cluster assignments fail to match the pre-defined classes.

6 Summary and Evaluation of Player Statistics

In the last chapter we looked our data from three different perspectives. This chapter summarizes our findings, proposes different usage scenarios and gives an overview of ideas on how the statistics can be used in real poker agents. We also discuss some of the concerns related to this work.

6.1 Concerns

General Concerns

First of all we will discuss some general concerns. The game of poker has an enormous amount of factors to consider. Even though we have only looked at a small subset of possible features to describe the game we struggled to gain proper information out of this subset. We did not consider game type (Texas Hold'em vs. Omaha Hold'em, though they are vastly different games), limit type (Limit vs. Fixed-Limit vs. No-Limit vs. Pot-Limit), number of players (heads-up vs. 6-max vs. 9/10-max) and different limits (e.g. NL2 vs NL200 vs NL5000). All of our analyzed data is from one specific combination of the mentioned stats only (i.e No-Limit Texas Hold'em 6-Max with 1$/2$ Blinds), due to this fact we can only say something about this specific "configuration" of poker.

Another factor which we basically disregarded was time. A basic assumption which can be made is, that the players get better the longer (i.e the more) they play. Taking this into consideration it is almost certain that the time frame covered by the sample is a significant factor. Our sample covered a time frame of roughly 16 months. Giving a good number for a time frame is difficult and up to the subject, but we believe that period covered by our sample is neither really bad nor really good. Within 16 months the playing style can change drastically, but it is not a period where we would say that some of the data might be too old to give a good picture of the player.

A more specific concern is, that when looking at the statistics of players who play adaptively. By playing adaptive we mean that the player will adjust his or her style of play when facing certain types (which can be arbitarily defined by the player him/herself) of opponents. When looking at the statistics of such a player it will become very hard to determine the "true" play style as the stats will only tell us something about the average. This is still a problem when looking at non-adaptive players, but we assume that their style of play will be closer to their average more often than it is not.

Position is insanely important in poker as seen in Figure X. For many stats we have a "total" variant of the statistic which might not be optimal for the majority of the time. This is a minor concern as we can, at least partially, bypass this by ignoring the total variant and use the per street statistics instead. Another thing to take into consideration is, that even the per street statistics might not be detailed and/or specific enough, as splitting them up into extremely detailed statistics is still possible, namely by looking at the stat for each position (EP, MP, LP, CO, BTN, Blinds).

There is also the random factor of poker which can influence the variance quite a bit, although this can be somewhat neutralized by using only (very) large sample sizes.

Statistic-specific Concerns

As we have seen from our statistics analysis the amount of information gained out of it has to be evaluated critically. First of all for nearly all stats there is no clear polarization of the data, meaning that it is often not possible to give exact thresholds where the value defines a winner or a loser. Many stats suffer from either having a very wide range of values (such as Aggression Factor) or having a very high standard deviation, making the process of classification or clustering very difficult. A few statistics also suffer from the fact that they are the same for all groups which makes those stats practically useless for classification.

We could certainly do manual selection of statistics which seem promising, but we also have to consider that these statistics may show certain characteristics because of the way we selected our sample for example - other factors could be of influence as well. Some of these stats might show better differentiation with other sample configurations (e.g a sample of NL2 or NL5000 hands might show completely different characteristics for the same player statistic). This of course, is hard to verify but at this point we can not deny the possibility of that being the case.

Manual selection of ”good” or ”bad” statistics is also prone to being biased, especially for statistics which are borderline usable (i.e. stats that only show a small difference between winners and losers, *Donk F* for example) as it is hard to decide whether these stats are actually significant or not. By using formal methods such as regression or correlation analysis we can eliminate bias, but due to previously described problems like high standard deviation and low disparity the prediction accuracy still remains poor.

Human-specific Concerns

Although the following concerns are only related to human poker players we felt that they should at least be mentioned since our data set is from real players. Players tilting and/or losing focus is another challenge, similar to that of adaptive players, which has to be overcome. Neither states of mind are neccessarily reflected in the players statistics as it is highly unlikely that players will constantly play unfocused and/or in a state of tilt. When facing computerized poker agents it is safe to assume that they will neither tilt nor play ”unfocused”, but most often these states of mind represent the best opportunities to exploit a human player.

6.2 Usability in Poker Agents

There are several ways to use player-specific statistics in poker agents. In this section we describe a few possible usage scenarios. All of the introduced statistics can be used with varying degree of detail, ranging from very rough categorization into player types to analyzing the behavior of the player for extremely specific situations.

6.2.1 Classification into Player Types

First of all we will talk about how to use the statistics for classification (and clustering). Classification is the process of identifying to which group, category or class a new observation belongs to. In our case, players (and their statistics) are observations, while the class(es) can be arbitarily defined. Our analysis used the classes of losers, breakeven players and winners. Other schemes of classification are also possible, such as the ”classic” player types like tight-aggressive (TAG), loose-aggressive (LAG), tight-passive (*Rocks*) or loose-passive (*Fish* or *Calling Station*).

A general downside to all classification methods is that the information is being generalized into one prediction, which is the class. Due to this downside we feel that the approach of classification is not optimal, as it does not fully utilize the potential information the statistics can convey. The main question is, how much information do we actually gain by knowing which class the new observation (i.e. player) belongs to and what do we actually do with this piece of information? In general, the poker agent will have to be modified in a way so that the classification is incorporated into the decision making algorithms. We will not go into detail on how they work, because it does not really matter here, but if we reduce the dimensionality of the data to a single statistic (the class) the applications are limited. A very basic approach which comes to mind would be using different decision sets (or weights) for different types of players.

Machine Learning

A common approach is to utilize machine learning algorithms for classification. There are a lot of different algorithms based on various methods, such as rule-learning, bayes classifiers, neural networks, (decision) trees or regression analysis. All methods can be basically summarized by the attempt to find correlations between the attributes (in our case the statistics) of the observation and the class. The effectiveness (or accuracy) depends on the used algorithm and data. In Section 5.5 we have already tested quite a few machine learning algorithms using Weka. The results of our tests were mixed, but generally not very promising, which is why we would suggest using other ways to utilize the data.

Clustering

As already described in Section 5.5.3, clustering is a method which is quite suitable for visualization. There are different approaches to clustering, for example including or excluding a class attribute. In case of our data this class attribute would be the group the player belongs to (namely winning, losing or breakeven player). By including or exluding a class attribute, we can see if there is any correlation between arbitarily defined classes and the actual statistics of the players. Clustering is usually done on all available attributes, but this can be difficult to visualize due to the high dimensionality. To solve this, the also previously mentioned, PCA[1] is used.

[1] Principal Component Analysis

Vector Space Models

Another approach is vector space modeling [52] to classification. Strictly speaking it is also a method which is used in machine learning. In our case we can see each player as a vector with a separate dimension for each of their statistics. This can be seen as an analogy to text classification using TF-IDF (term frequency and inverse document frequency) vectors. Calculating an average or median vector of winning and losing players can serve as a foundation for the process of classification. By comparing each player to these vectors we can calculate whether they are closer to the class of winners or losers (or any other class definition). A finer-grained class split is also possible (for example splitting the group of winners into big, medium and small winners) as well as fine-tuning the vector itself by limiting it to certain statistics only (a process called attribute selection). Based on the resulting classification the poker agents can adapt it's decision strategies by knowing what type of player they are facing. The advantage of this approach is that it is very easy to do, while the downside being that it might not be very precise due to the non-normality of all stats.

6.2.2 Per Decision based Statistics Usage

In our opinion the best way to use the statistics is to incorporate them when making certain (single, or situation-based) decisions. Most stats are either situation or street specific (many are in fact are both), while there are very few which can be generally applied to all decisions. Stealing, limping and stats related to continuation bets or 3bets for example are very situation/street specific stats, while (total) Aggression Factor, PFR/VPIP or WTSD are more general statistics which can be applied more universally. Although poker is a game of imperfect information we can reliably and accurately gauge in what kind of situation the player is in if we just look at all the player actions.

For example if we are facing a raise on the flop we know that we are facing an opportunity to 3bet (the flop) - by using this knowledge we can look at the stats of the player who made the raise which are related to this particular situation. If his/her *Fold to F 3Bet* is higher than normal we could give (re-)raising (3betting) a higher weight or priority than we normally would.

This can be extended to pretty much any given situation (stealing, facing steals, limping, facing continuation bets and the list goes on) and is basically the way human players utilize the statistics shown on the HUDs at the online tables. We can even take this a step further by splitting the statistics up into position specific statistics which results in higher resolution data of each player.

The obvious advantage is that we can utilize the statistics to their full potential since we actually use them the way they are intended for. Unlike the classification approach we do not "lose" information by generalization but instead we look at each player's characteristics individually.

6.2.3 Hybrid and other approaches

The two proposed approaches are not disjunct, in fact, they can be combined freely. As an example we could use classification methods to define a baseline behavior for the agent against certain types of opponents (e.g. play more aggressively versus very tight players) and then use situation-specific evaluation of the opponent's stat to fine-tune the final decision.

Finding the agents own weaknesses

An approach that is not directly connected to exploiting the opponent is to find and correct the agents own weaknesses. All methods previously described can also be utilized when looking at the agents own statistics. Classifying the agent into the group of winners or losers will give a rough idea on how the performance of the poker agent is to be evaluated, but in order to solve problems or simply improve the play a detailed analysis of each statistic is most likely neccessary. Again, there are several ways to do that, for example comparing each stat to its total average/mean or average/mean within groups. After the analysis we still have to decide on how to rectify or improve the statistic which can be derived from the formula of each stat. We can take VPIP as a concrete example. Comparing its value to the total sample or certain groups is a trivial process. After the comparison we can now try to adjust this value: when it's too high we will play less hands and vice-versa if it's too low we let the agent play more hands preflop. For VPIP this adjustment is still easy to make, but for more complex statistics like Aggression Factor this can become more tricky. The downside or problem of this approach is still the high spread of all statistics. Finding the right threshold or optimal area for each statistic is difficult.

7 Conclusion and Outlook

Poker is a complex and challenging game and requires a wide array of (different) skills in order to gain an advantage over the opponents. In this thesis we have introduced a multitude of statistics which are used by online poker players to gain an informational advantage over their opponents. These metrics can be employed in poker agents as well. We analyzed the informational value of these statistics and used techniques from the field of mathematics and machine learning. The categorization and classification of poker players is a complex and not well-understood research.

Instead of analyzing the classic opponent types (*TAG, LAG, rock, fish/calling station*) we choose to group the players by their winrate, namely their BB/100 value. In our opinion this is the most objective and meaningful measure of a players success. We used three categories: players who lost money (losers), players who broke even (breakeven) and players who won money (winners). In hindsight this categorization may have increased the difficulty of actually finding out how the players in the respective groups play.

In order to grasp the play style of each group we analyzed over 60 player-specific metrics. The result of this was very mixed: while some statistics showed a good disparity between the groups of winners and losers, many stats did not show any difference between groups at all. Another problem we faced was the non-normality of nearly all statistics and some of them suffered from heavy outliers. Due to this we resorted mainly to non-parametric statistics, but even then the effective range (i.e. the IQR) the values can take was still quite large and in most cases overlapping with other classes as well.

We conclude that categorizing players into groups, using both classic opponent types and winner/loser type, is of limited use. Within our used categories the composition of play styles varied by a lot and altough slight trends could be seen, the overlapping between groups was simply too high to give a conclusive statement about what is a winning or losing play style. We think the best way to utilize statistics is to use them to make specific situation-based decisions, for example by using quantiles and determining outlier characteristics. As opposed to classification, this approach also has the advantage that less information is lost by generalization into classes since we would look at each statistic separately.

This work is intended to give a solid foundation for further research. At this time of writing the possibilities of trying out the the proposed methods of usage in poker agents are limited and work has to be done in order to advance. Poker frameworks would need to be extended or re-developed with appropriate features.

One method would require the ability of the poker framework to generate PokerTracker 4 compatible hand histories and just retrieve the results of the calculations done by PokerTracker. This is somewhat reliant on the PokerTracker development team, unless we run SQL queries directly on the PokerTracker database, since there is currently no released documentation and/or API. When choosing the route of accessing the database directly a good amount of reverse engineering will still be required as there is currently database schema documentation.

Another possibility would be to copy the functionality of PokerTracker (the ability to extract the numbers required for calculations from hand histories and implement all formulas) to a poker framework. While this approach requires a good amount of reverse engineering and programming to be done, it is most likely the most effective (and fastest) way to do it, because the database can be customized and is not reliant on any third parties.

Bibliography

[1] Erik Arneson. *Poker Dice Rules*. Guide. URL: http://boardgames.about.com/od/pokerdice/a/rules.htm (visited on 04/25/2013).

[2] Johan Karlsteen Aron Andersson and Rickard Andersson. "Data Mining Poker Hand Histories". In: (). Date unknown. URL: http://www8.cs.umu.se/kurser/TDBD15/VT05/report8.pdf.

[3] *Bertrand Grospellier*. Encyclopedic entry. 2013. URL: http://en.wikipedia.org/w/index.php?title=Bertrand_Grospellier&oldid=554664565 (visited on 02/2013).

[4] Daniel Bragonier. "Statistical Analysis of Texas Holdem Poker". In: (2010).

[5] User bravos1. *By request : Poker Tracker and stats*. Forum entry. 2008. URL: http://forumserver.twoplustwo.com/35/micro-stakes-limit/request-poker-tracker-stats-165898/ (visited on 02/2013).

[6] Chij-Jen Lin Chih-Chung Chang. *LibSVM – A Library for Support Vector Machines*. URL: http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (visited on 04/28/2013).

[7] Jordan D. *Poker HUD Guide*. Article. 2011. URL: http://www.myholdempokertips.com/hud-stats-guide (visited on 02/2013).

[8] *HandHQ.com*. URL: http://www.handhq.com (visited on 01/2013).

[9] *Hold'em Manager*. Product site. URL: http://www.holdemmanager.com (visited on 01/2013).

[10] Owen Laukkanen. *ElkY grinds to SNG world record*. Article. 2009. URL: http://www.pokerlistings.com/elky-grinds-to-sng-world-record-39706 (visited on 03/2013).

[11] Hold'em Manager. *Hold'em Manager 1*. Product site. URL: http://www.holdemmanager.com/store/downloads-manuals.php (visited on 02/2013).

[12] Hold'em Manager. *Hold'em Manager 2*. Product site. URL: http://www.holdemmanager.com/buy/211/holdem-manager-2 (visited on 01/2013).

[13] Hold'em Manager. *Stat Definitions*. Software manual. URL: http://faq.holdemmanager.com/questions/95/Stat+Definitions (visited on 02/2013).

[14] User mpethybridge. *Aggression Factor Categories*. Forum entry. 2009. URL: http://forumserver.twoplustwo.com/showpost.php?p=15540847&postcount=2487 (visited on 02/2013).

[15] *PartyPoker.com*. Product site. URL: http://www.partypoker.com (visited on 01/2013).

[16] Pokerpages. *The History of Poker*. Article. URL: http://www.pokerpages.com/pokerinfo/history.htm (visited on 02/2013).

[17] PokerProfile. *Opponen Profiling - Making Use of your Statistics*. Article. 2006. URL: http://www.hand-histories.com/en/Articles/1/Opponent-Profiling---Making-Use-of-your-Statistics (visited on 02/2013).

[18] PokerScout. *Statistics for Pokerstars.com*. URL: http://www.pokerscout.com/SiteDetail.aspx?site=PokerStars (visited on 03/2013).

[19] Pokerstars.com. *Poker Rules*. Article. URL: http://www.pokerstars.eu/poker/how-to-play/rules/ (visited on 03/2013).

[20] *PokerStars.com*. URL: http://www.pokerstars.com (visited on 01/2013).

[21] Pokerstrategy. *Equilab*. Product site. URL: http://www.pokerstrategy.com/software/10/ (visited on 02/2013).

[22] *Pokerstrategy*. URL: http://www.pokerstrategy.com (visited on 02/2013).

[23] *PokerTracker*. Product site. URL: http://www.pokertracker.com (visited on 01/2013).

[24] PokerTracker. *PokerTracker 3 Database Schema Documentation*. Software manual. 2012. URL: https://www.pokertracker.com/guides/PT3/databases/pokertracker-3-database-schema-documentation (visited on 02/2013).

[25] PokerTracker. *PokerTracker 4*. Product site. URL: http://www.pokertracker.com/products/PT4/ (visited on 01/2013).

[26] PokerTracker. *Statistical Reference Guide*. Software manual. 2012. URL: https://www.pokertracker.com/guides/PT3/general/statistical-reference-guide (visited on 02/2013).

[27] User Pokey. *An Unbelievable Long Guide to Hand-Reading*. Forum entry. 2007. URL: http://archives1.twoplustwo.com/showflat.php?Number=8629256 (visited on 02/2013).

[28] User Pokey. *How to use Poker Tracker*. Forum entry. 2006. URL: http://archives1.twoplustwo.com/showflat.php?Cat=0&Number=4946669 (visited on 02/2013).

[29] PostgreSQL. *PostgreSQL*. Product site. URL: http://www.postgresql.org/ (visited on 01/2013).

[30] RedJoker. *Reading and Interpreting Holdem Manager and PokerTracker Statistics*. Article. 2012. URL: http://www.bestofrakeback.com/eng/poker-articles/poker-help-tips/reading-interpreting-poker-statistics (visited on 02/2013).

[31] Daniel Skolovy. *How to Interpret Your Opponent's Poker Stats*. Article. 2010. URL: http://www.pokerzeit.com/poker-software-stats-und-wie-man-sie-interpretiert (visited on 02/2013).

[32] The Gamble Society. *The History Of Poker*. 2013. URL: http://www.thegamblesociety.com/online-poker/history/ (visited on 03/2013).

[33] User *Split*. *COTW: Configuring Our HUD Stats*. Forum entry. 2010. URL: http://forumserver.twoplustwo.com/showpost.php?p=16525918&postcount=23 (visited on 02/2013).

[34] User tigerjack89. *Difference between HM1 AFq and PT4 AFqX*. Forum entry. 2013. URL: https://www.pokertracker.com/forums/viewtopic.php?f=61&t=47542 (visited on 02/2013).

[35] Tightpoker.com. *Position in Poker*. Article. URL: http://www.tightpoker.com/position.html.

[36] Machine Learning Group at the University of Waikato. *Weka*. Book alternative: Data Mining: Practical Machine Learning Tools and Techniques by Witten, Frank and Hall. URL: http://www.cs.waikato.ac.nz/ml/weka/ (visited on 04/2013).

[37] Nick Wealthall. *Poker Player Types - Know Your Foe*. Article. 2008. URL: http://www.pokerplayer.co.uk/poker-strategy/cash-poker/6318/poker_player_types.html (visited on 04/25/2013).

[38] Weka. *Documentation: Class LibSVM*. Software documentation. URL: http://weka.sourceforge.net/doc.stable/weka/classifiers/functions/LibSVM.html (visited on 04/28/2013).

[39] Weka. *Documentation: Class Logistic*. Software documentation. URL: http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/Logistic.html (visited on 04/28/2013).

[40] Weka. *Documentation: Class PrincipalComponents*. Software documentation. URL: http://weka.sourceforge.net/doc.dev/weka/attributeSelection/PrincipalComponents.html (visited on 04/28/2013).

[41] Weka. *Documentation: Class SimpleKMeans*. Software documentation. URL: http://weka.sourceforge.net/doc.dev/weka/clusterers/SimpleKMeans.html (visited on 04/28/2013).

[42] Weka. *Documentation: Class SimpleLogistic*. Software documentation. URL: http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SimpleLogistic.html (visited on 04/28/2013).

[43] Weka. *Documentation: Class SMO*. Software documentation. URL: http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html (visited on 04/28/2013).

[44] *Weka Class AbstractClassifier*. Software documentation. URL: http://weka.sourceforge.net/doc.dev/weka/classifiers/AbstractClassifier.html (visited on 04/28/2013).

[45] *Weka Documentation*. Software documentation. URL: http://weka.sourceforge.net/doc.dev/overview-summary.html (visited on 04/28/2013).

[46] Wikipedia. *Betting in Poker*. Encyclopedic entry. URL: https://en.wikipedia.org/w/index.php?title=Betting_in_poker&oldid=541789992 (visited on 03/2013).

[47] Wikipedia. *Hand history*. Encyclopedic entry. URL: https://en.wikipedia.org/w/index.php?title=Hand_history&oldid=488871190 (visited on 03/2013).

[48] Wikipedia. *History of poker*. Encyclopedic entry. 2012. URL: http://en.wikipedia.org/w/index.php?title=History_of_poker&oldid=527813034 (visited on 03/2013).

[49] Wikipedia. *History of poker*. Encyclopedic entry. 2013. URL: http://en.wikipedia.org/w/index.php?title=Texas_hold_%27em&oldid=539239184#Popularity (visited on 02/2013).

[50] Wikipedia. *Poker boom*. Encyclopedic entry. URL: http://en.wikipedia.org/w/index.php?title=Poker_boom&oldid=544989328 (visited on 03/2013).

[51] Wikipedia. *Texas Hold'em Rules*. Encyclopedic entry. URL: http://en.wikipedia.org/w/index.php?title=Texas_hold_%27em&oldid=539239184#Rules (visited on 03/2013).

[52] Wikipedia. *Vector Space Model*. Encyclopedic entry. URL: http://en.wikipedia.org/w/index.php?title=Vector_space_model&oldid=548024109 (visited on 03/2013).