

Analiza a castigurilor in jocul de Poker

Adrian Adam

Iancu Onescu

Mihail Feraru

05 February 2021

Abstract

O scurta analiza a rezultatelor mainilor de poker urmarind cat de mult se apropie de o distributie Gaussiana, respectiv influenta asupra planificarii bugetului si evaluarii performantei jucatorilor.

Aplicatia interactiva cu intreaga analiza: <https://seras3.shinyapps.io/PokerStatsApp/>

Cod sursa: <https://github.com/IancuOnescu/poker-stats>

Introducere

Jocul de poker se afla intr-un punct controversat intre sansa si strategie, astfel pentru o analiza consistenta asupra propriei performante, un jucator are nevoie de procedee matematice concrete de evaluare.

In articolul lui Mike Stein, *How far from Gaussian (normal) are poker results?*, acesta isi propune sa determine cat de departe sunt propriile sale rezultate fata de o distributie Gaussiana si cum poate folosi acest fapt pentru planificarea bugetara reducand riscul de “ruina”. In vederea generalizarii observatiilor, am decis sa extidem analiza la un set de date mai cuprinzator cu jucatori de aptitudini variate pe o perioada mult mai lunga de timp. De asemenea vom utiliza si alte procedee statistice pentru o mai buna intelegere a datelor.

Dataset-ul utilizat

Ca baza pentru analiza noastra am folosit 10 milioane de maini de poker inregistrate intre 1995 si 2001 pe mai multe canale de IRC dedicate jocului. Jucatorii sunt in general *computer geeks* ai anilor 90, studenti, matematicieni sau roboti, iar pariurile nu au fost realizate cu bani reali. Ne-am concentrat doar asupra variantei Texas Hold'em excluzand turneele.

Dataset-ul este organizat pe luni calendaristice si mese de joc (canale de IRC), datele fiind stocate in format text. De exemplu `data/holdem1/199601` reprezinta jocurile inregistrate pe canalul `#holdem1` in Ianuarie 1996. Fiecare astfel de folder contine:

- `hroster` - jucatorii prezenti la fiecare mana
- `hdb` - pariurile in fiecare etapa a jocului si cartile de pe masa
- `pdb/<nume>.pdb` - mainile jucate de `<nume>`, actiunile, cartile si pariurile sale

Procesarea datelor in format text este lenta si complicata in lipsa unui format standard, asa ca am utilizat o serie de script-uri in Python, Bash si R pentru a le converti in formatul Feather, dupa cum urmeaza:

- `decompress.py` si `parse.py` decompresaza si transforma datele in format CSV
- Urmatorul script Bash creeaza un director de iesire cu structura corecta:

```
find parsed_data -type d > dirs.txt
sed -i 's/parsed_data/dataframes/g' dirs.txt
xargs mkdir -p < dirs.txt
rm -f dirs.txt
```

- Transformare din CSV in Feather in maniera paralelizata:

```
convert_all_csvs_to_feather = function (
  path,
  INPUT_DIR_NAME = "parsed_data",
  OUTPUT_DIR_NAME = "dataframes"
) {
  csvs = list.files(path, pattern = "*.csv",
                    recursive = TRUE, full.names = TRUE)

  cl = makeForkCluster()
  pblapply(csvs, function(name) {
    df = read.csv(name)

    outname = gsub(INPUT_DIR_NAME, OUTPUT_DIR_NAME, name)
    outname = gsub("\\.csv", ".feather", outname)
    write_feather(df, outname)
    return ()
  }, cl = cl)
  stopCluster(cl)
}
```

Majoritatea mainilor au fost jucate de un set relativ mic de jucatori (sub 10%), astfel in analizele ulterioara ne vom concentra in special pe primii 50 de jucatori ca activitate.

```
library(PokerStats)
library(feather)

data = read_feather("./dataframes/all_hands.feather")
pfreq = player_hands_freq(data)
pstats = player_hands_stats(pfreq)

# Cuantile jucatori/maini jucate
print (pstats$quantiles)

##      10%      25%      50%      90%      95%      99%
##      5.00     22.00    107.00   3943.00  9207.50 36341.12

print (paste("Media nr. de maini jucate:", pstats$mean))

## [1] "Media nr. de maini jucate: 2014.318086627"
```

Legea lui Benford

Pentru a ne asigura ca datele nu au fost alterate in timpul colectarii sau in procesarile ulterioare am considerat relevant sa verificam daca variabile precum pariurile, castigurile si soldurile jucatorilor respecta o distributie Benford.

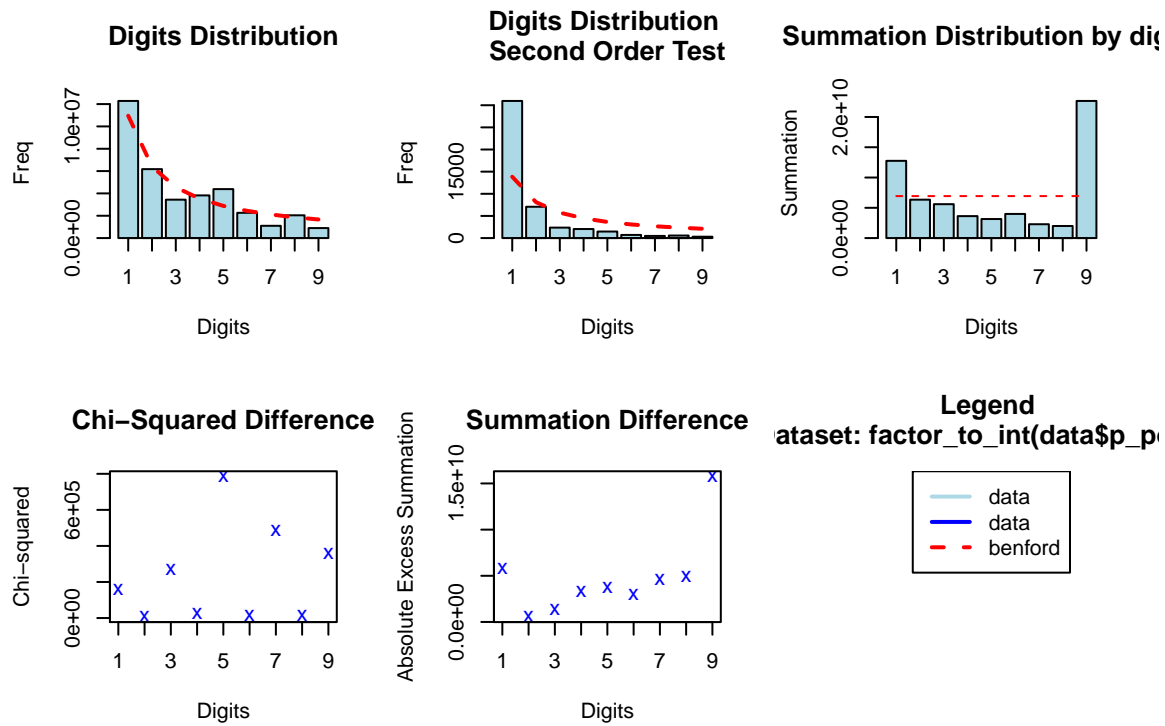
O observatie interesanta este ca o buna parte din cei mai activi jucatori este reprezentata de roboti (4/10 in top 10), insa comportamentul acestora nu se abate de la Legea lui Benford.

```
library(benford.analysis)

bf1 = benford(factor_to_int(data$p_pot_sz), number.of.digits = 1)
bf2 = benford(factor_to_int(data$p_bankroll), number.of.digits = 1)
bf3 = benford(factor_to_int(data$win), number.of.digits = 1)
```

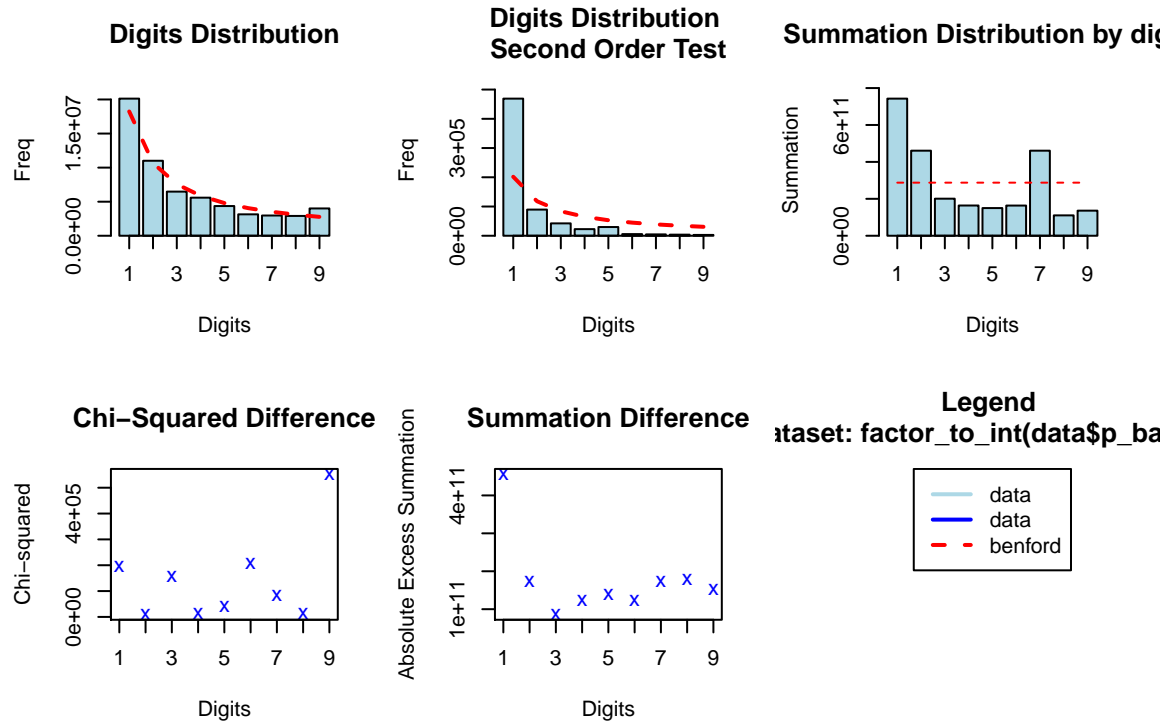
```
plotBf = function (bf) {
  plot(bf) # , except = c("mantissa", "abs diff", "second order", "summation", "abs diff", "ex summation")
}

plotBf(bf1); print(chisq(bf1))
```

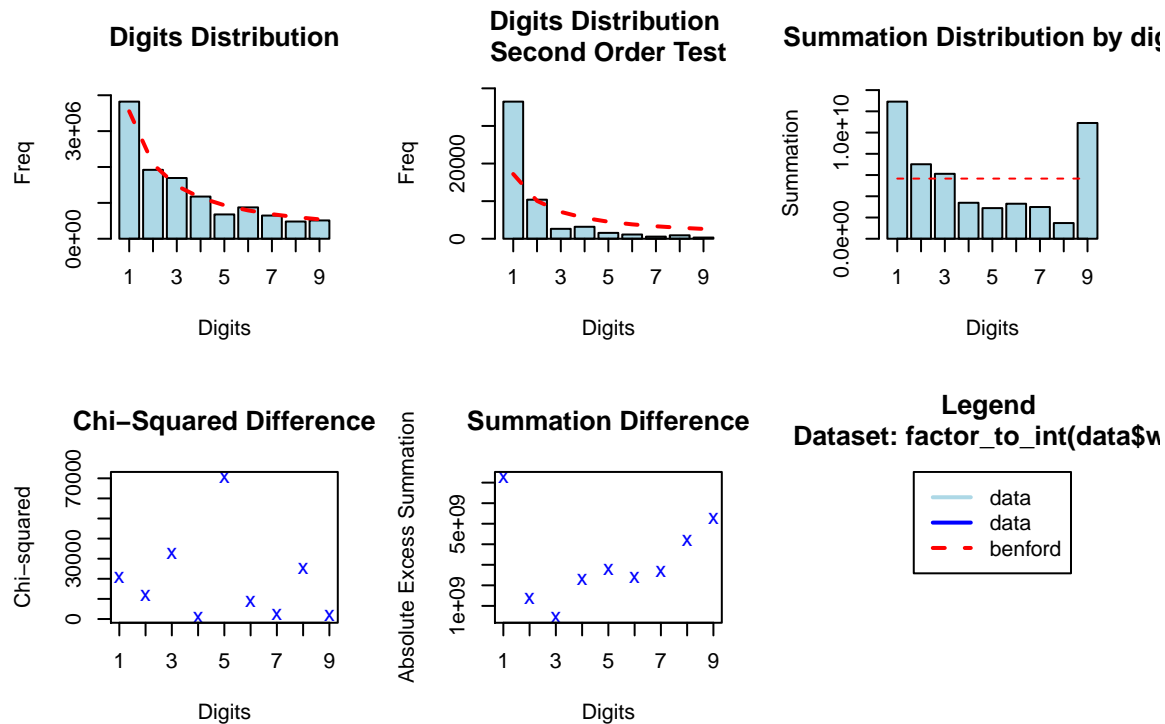


```
##
## Pearson's Chi-squared test
##
## data: factor_to_int(data$p_pot_sz)
## X-squared = 2118545, df = 8, p-value < 2.2e-16

plotBf(bf2); print(chisq(bf2))
```



```
##
## Pearson's Chi-squared test
##
## data: factor_to_int(data$p_bankroll)
## X-squared = 1268875, df = 8, p-value < 2.2e-16
plotBf(bf3); print(chisq(bf3))
```



```
##
## Pearson's Chi-squared test
##
## data: factor_to_int(data$win)
## X-squared = 174220, df = 8, p-value < 2.2e-16
```

Se observa atat grafic, cat si din testul statistic Chi-Square ca datele analizate sunt repartizate Benford.

Castiguri

Acum ca am incheiat cu prezentarea si validarea datelor vom trece la obiectul principal al acestui proiect si anume analiza. Ideea centrala a fost sa vedem daca distributia castigurilor unui jucator poate fi aproximata cu o curba gaussiana. In perioada curenta aceasta distributie starneste cel mai mare grad de atentie datorita proprietatilor ei. In sine, distributia probabilitatilor meciurilor de poker este discreta, nu continua. Insa Teorema de limita centrala ne spune ca folosind suficient de multe maini in final vom ajunge la rezultatul dorit.

Ne intereseaza sa vedem care este numarul de maini de la care media tinde catre cea normala si in final cat de aproape reusim sa ajungem de curba, pentru a aproxima si suma necesara sa mentinem aceasta distributie eleganta a castigurilor.

Din datasetul pentru fiecare jucator(mainile acestuia) alegem un numar de maini (100, 1.000 sau 10.000) insumand toate rezultatele. Alegem praguri atat de diferite pentru a putea observa mai usor modalitatea de convergenta. Facem asta pentru 100.000 de "pachete" de sample-uri si plotam rezultatele obtinute.

```
library(PokerStats)

fpfreq = filter(player_hands_freq(data), hands >= 10000)

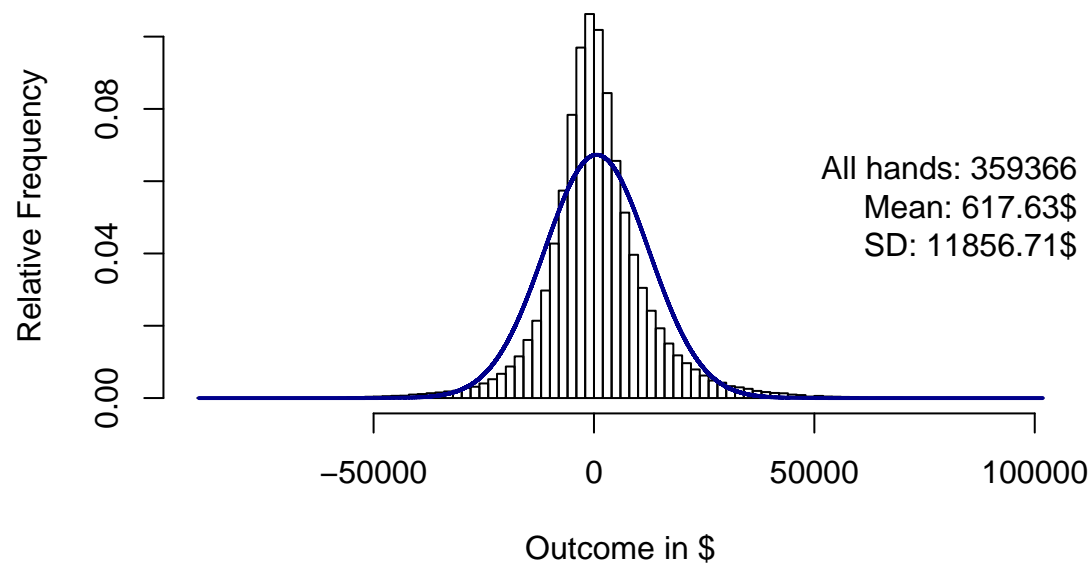
for(i in 2:5){
```

```

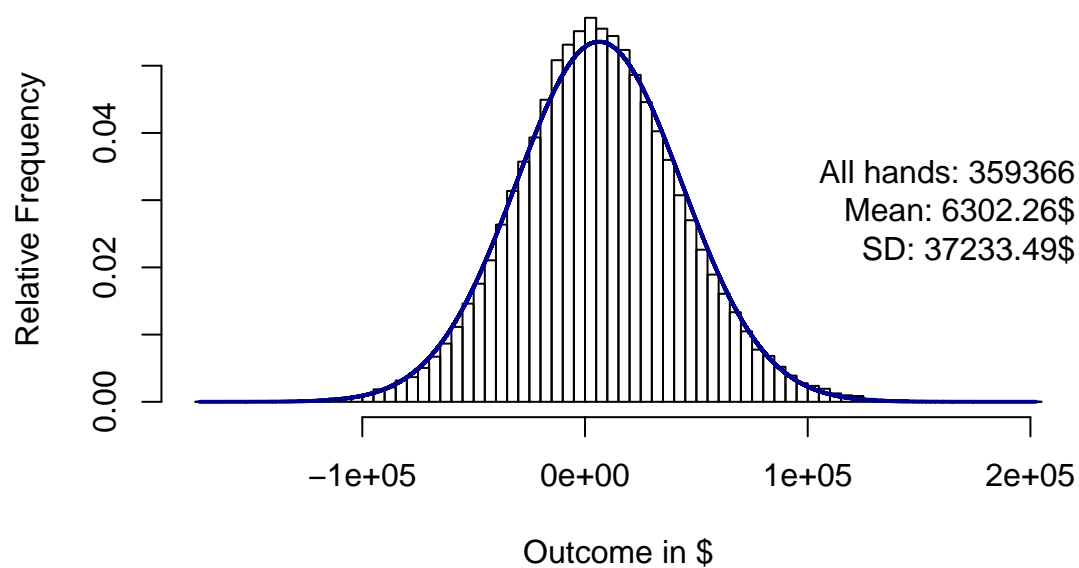
pdata = filter(data, p_name == fpfreq$p_name[i])
name = as.character(fpfreq$p_name[i])
plot_player_distr(pdata, name, 100, prefiltered = TRUE)
plot_player_distr(pdata, name, 1000, prefiltered = TRUE)
plot_player_distr(pdata, name, 10000, prefiltered = TRUE)
}

```

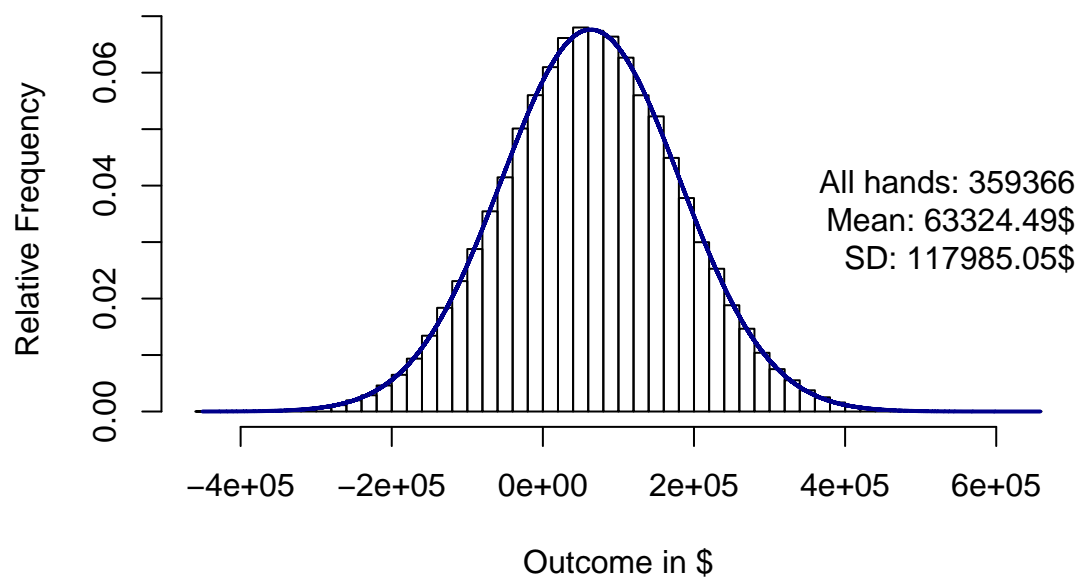
Sample distribution of 100 hands from sagerbot



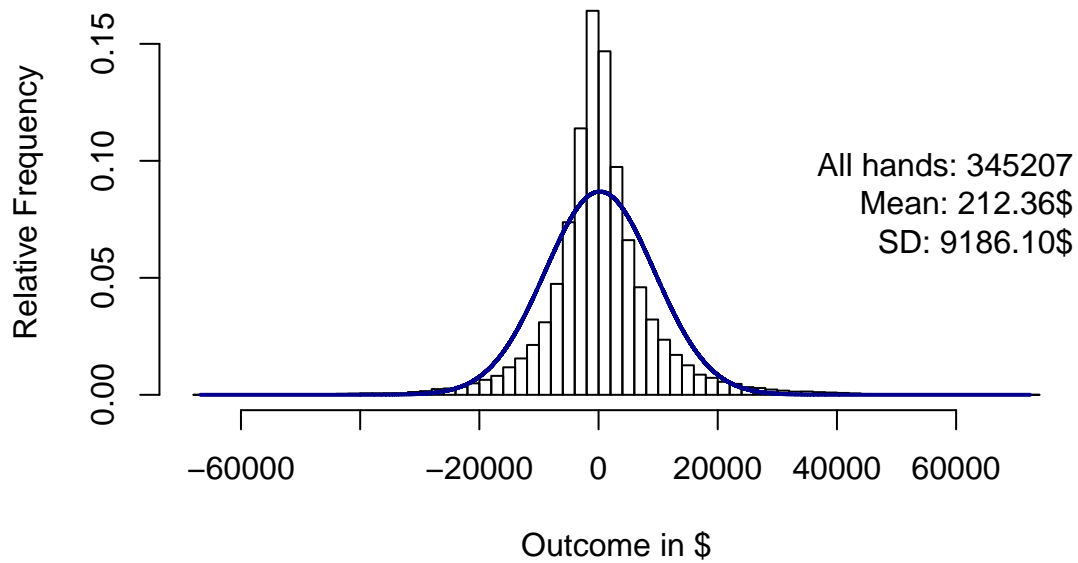
Sample distribution of 1000 hands from sagerbot



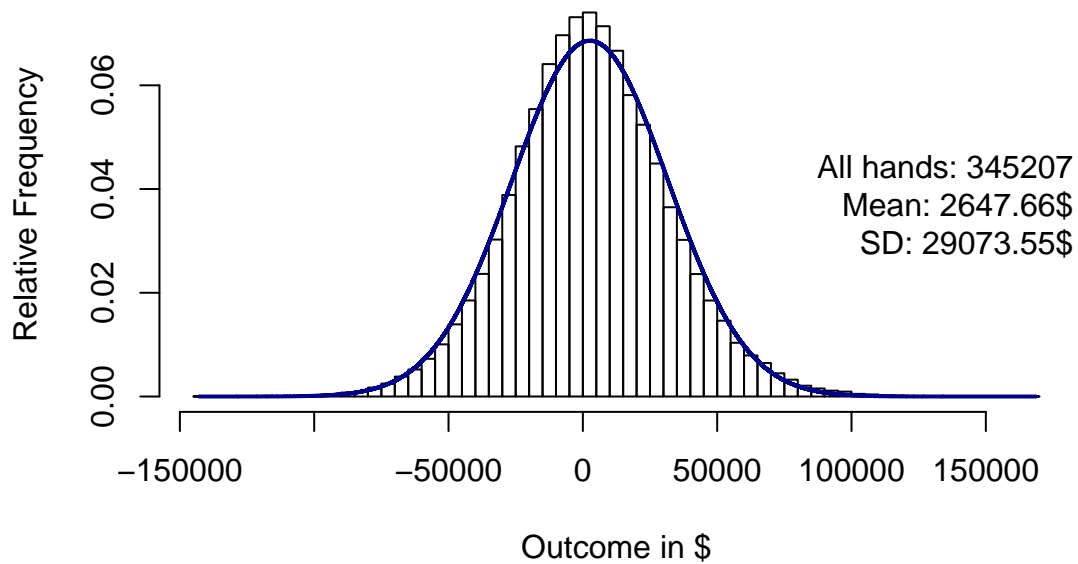
Sample distribution of 10000 hands from sagerbot



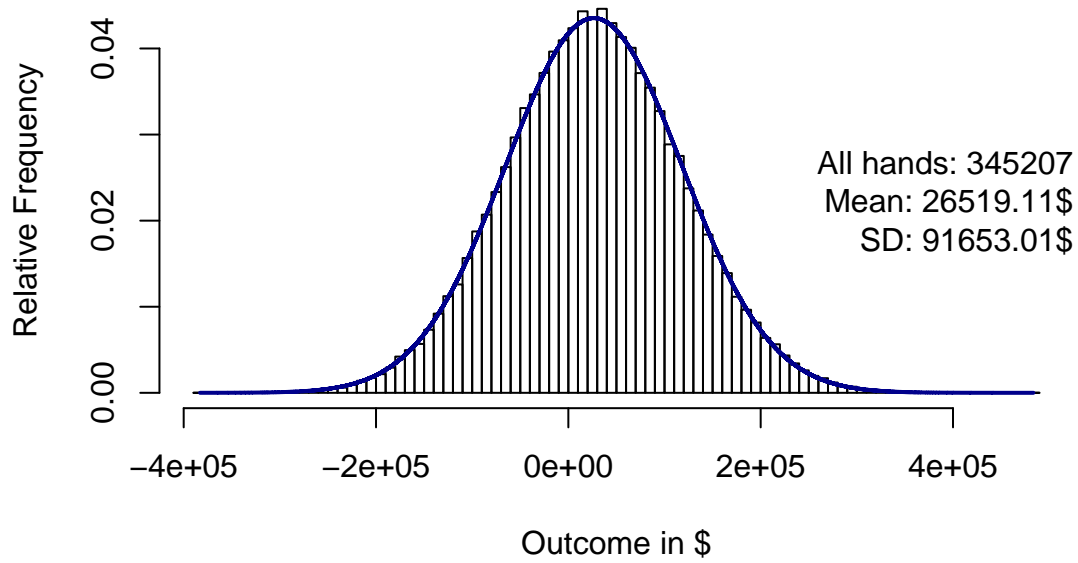
Sample distribution of 100 hands from kfish



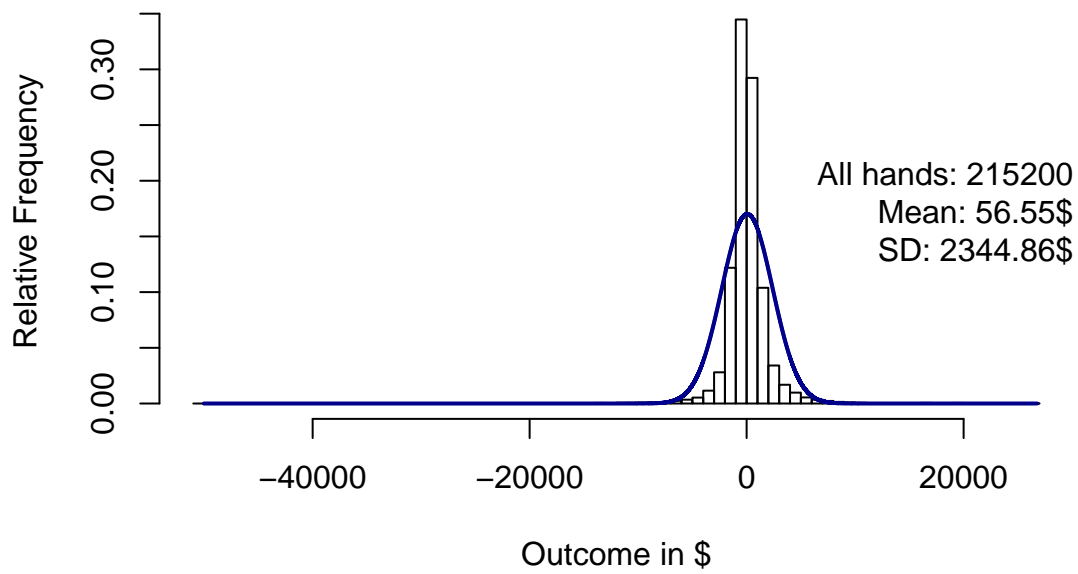
Sample distribution of 1000 hands from kfish



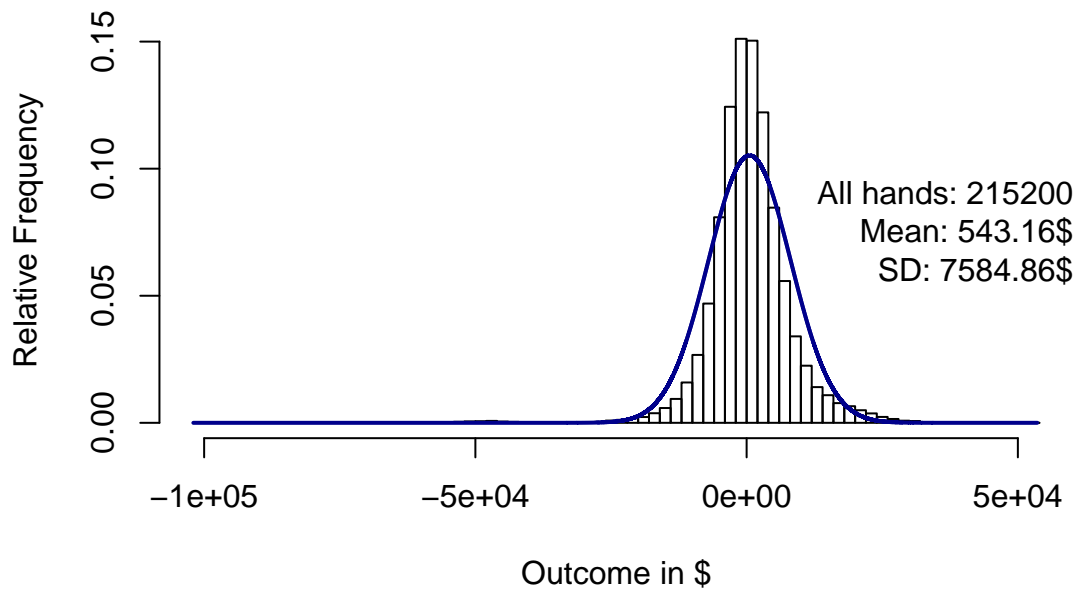
Sample distribution of 10000 hands from kfish



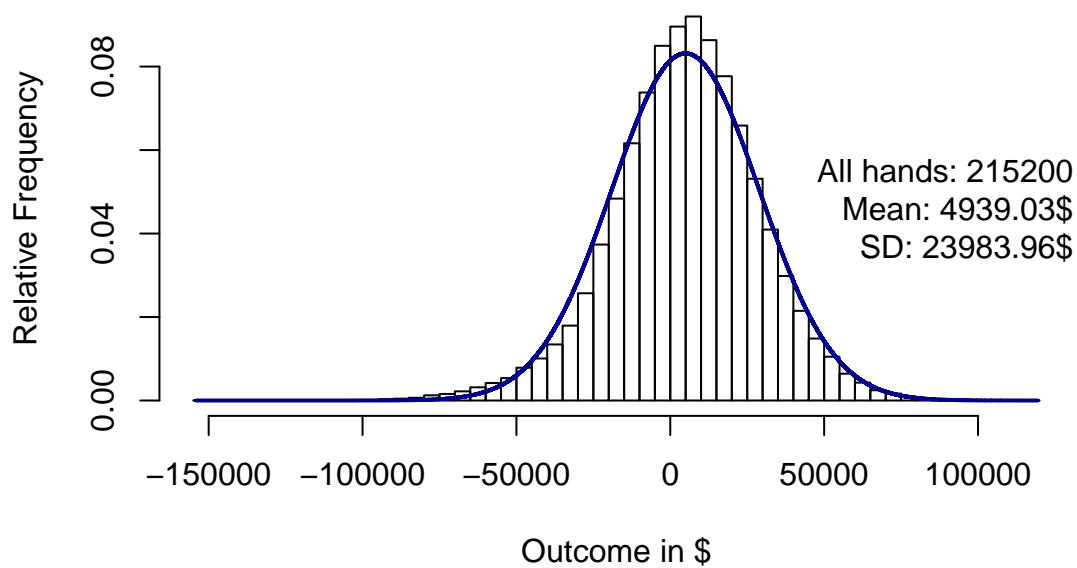
Sample distribution of 100 hands from BlackBart



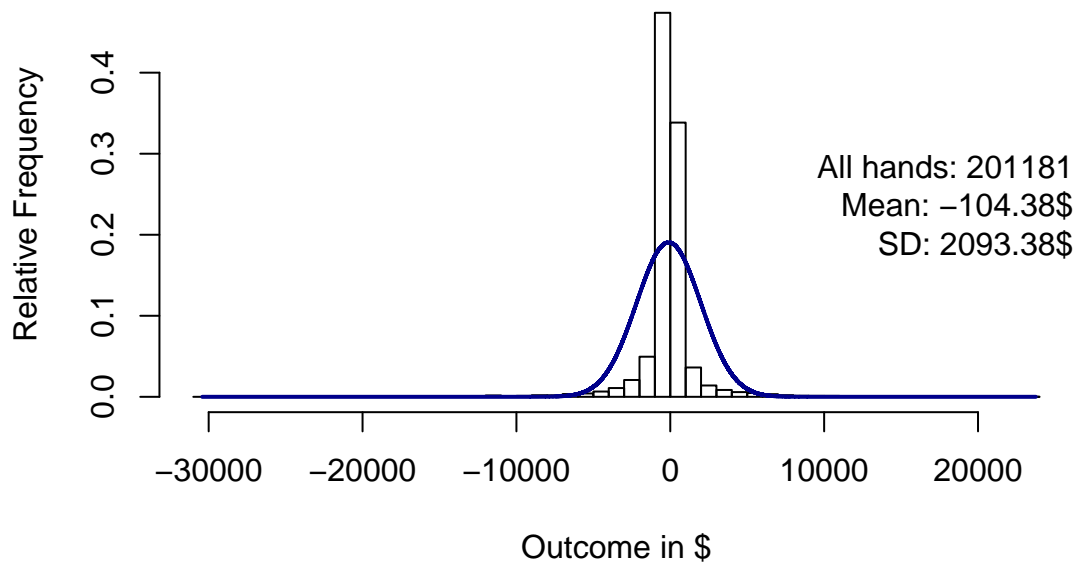
Sample distribution of 1000 hands from BlackBart



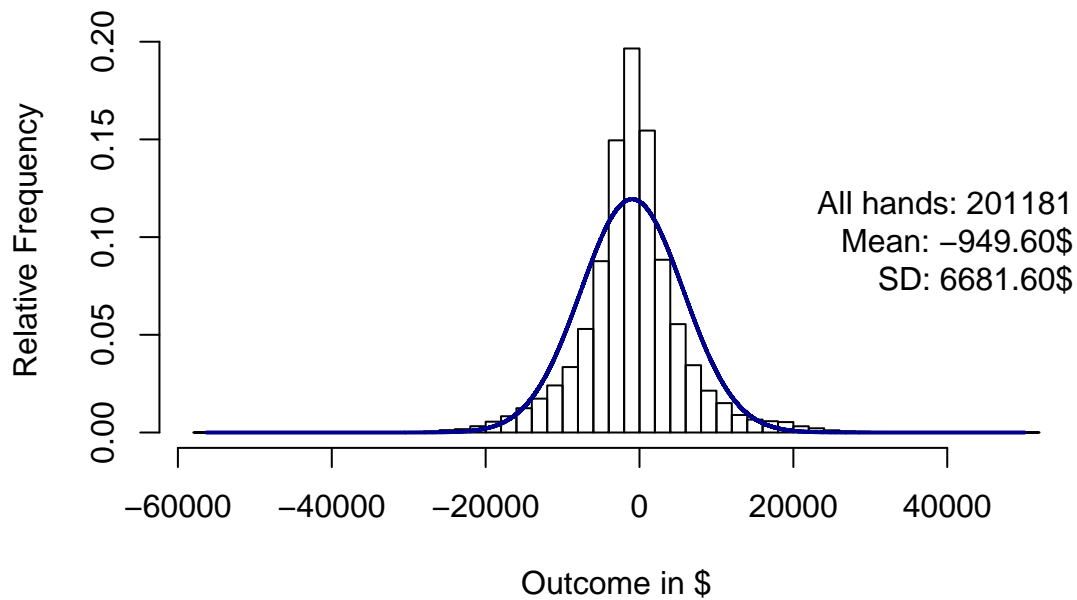
Sample distribution of 10000 hands from BlackBart



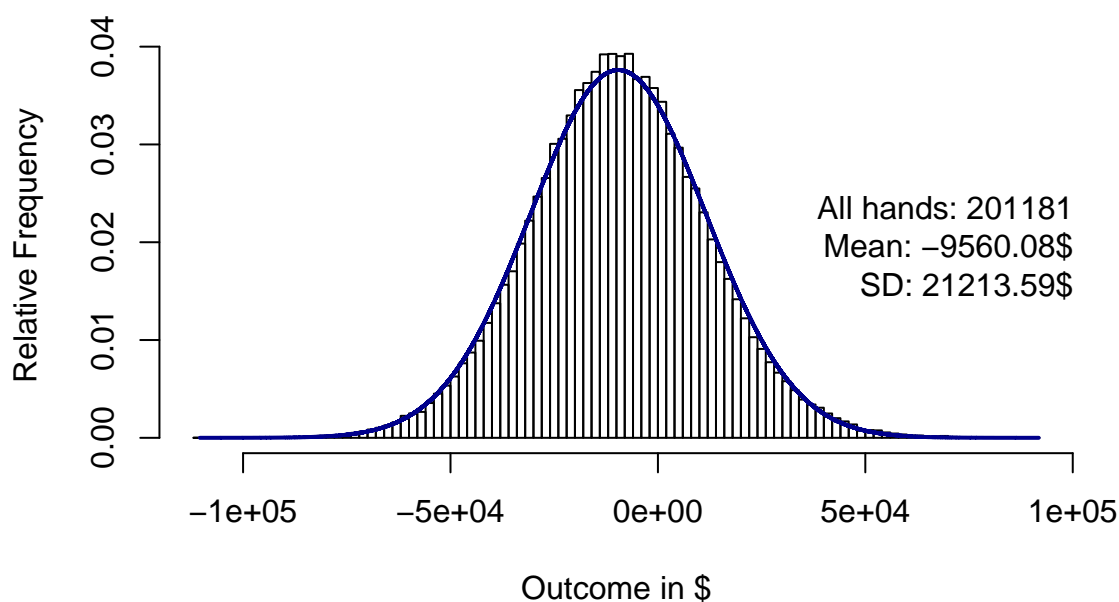
Sample distribution of 100 hands from kwAAkbot



Sample distribution of 1000 hands from kwAAkbot



Sample distribution of 10000 hands from kwAAkbot



Asa cum ne asteptam, pentru 10 maini nu suntem nici macar aproape, majoritatea *greutatatii* aflandu-se in punctul 0.

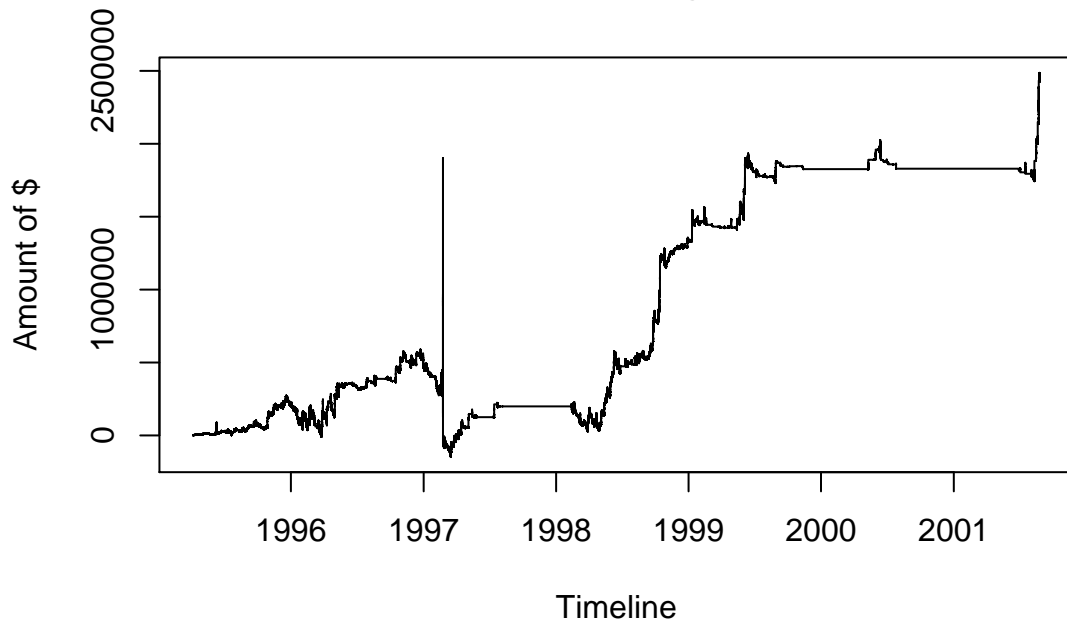
Pentru aproape toti jucatorilor deja de la pragul de 1000 de maini se poate vedea cum castigurile se apropie semnificativ de normala, iar de la pragul de 10.000 nu mai avem niciun dubiu. Observam si extremitatile mai accentuate ale graficelor, lucru care era de asteptat avand in vedere tendintele din jocul de poker. Jucatorii care se afla in posesia sumelor foarte mari tind sa parieze tot pentru a initimida sau chiar elimina un adversar. Pierderile de sume mari arata lipsa inspiratiei sau a norocului.

Pentru curiozitatea cititorului lasam mai jos cateva grafice cu castigurile jucatorilor analizati mai sus. Concluziile sunt lasate la latitudinea acestuia.

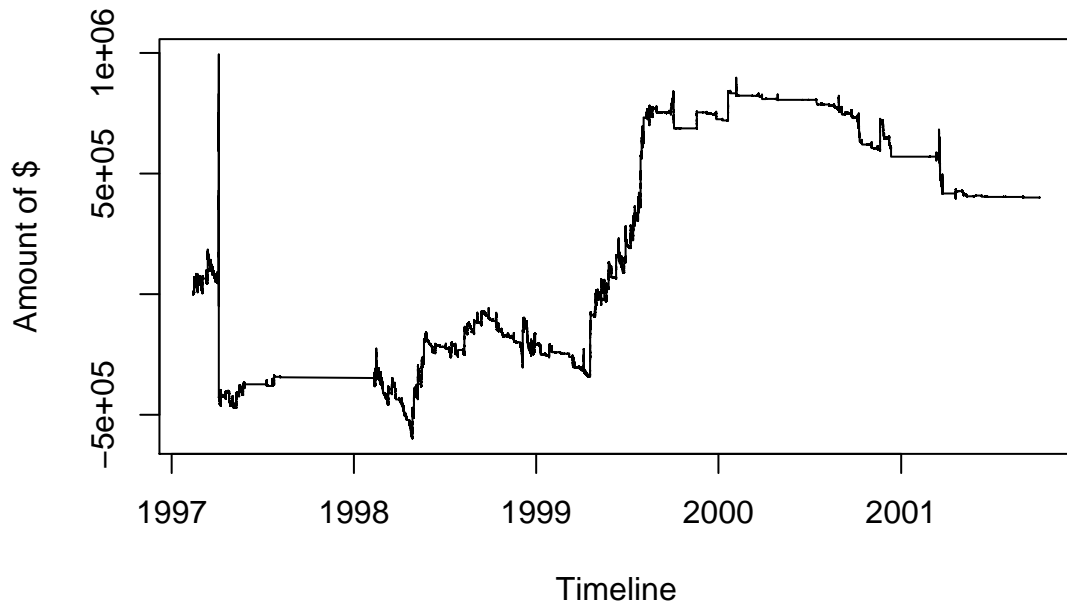
```
library(PokerStats)

for(i in 2:5){
  pdata = filter(data, p_name == fpfreq$p_name[i])
  name = as.character(fpfreq$p_name[i])
  plot_bankroll(pdata, name)
}
```

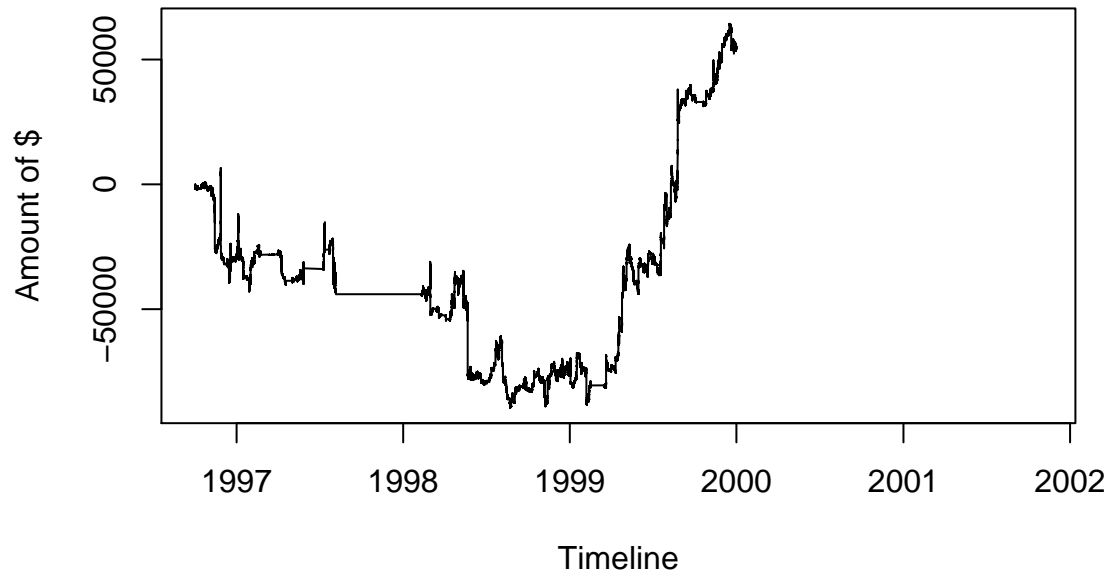
Bankroll of sagerbot



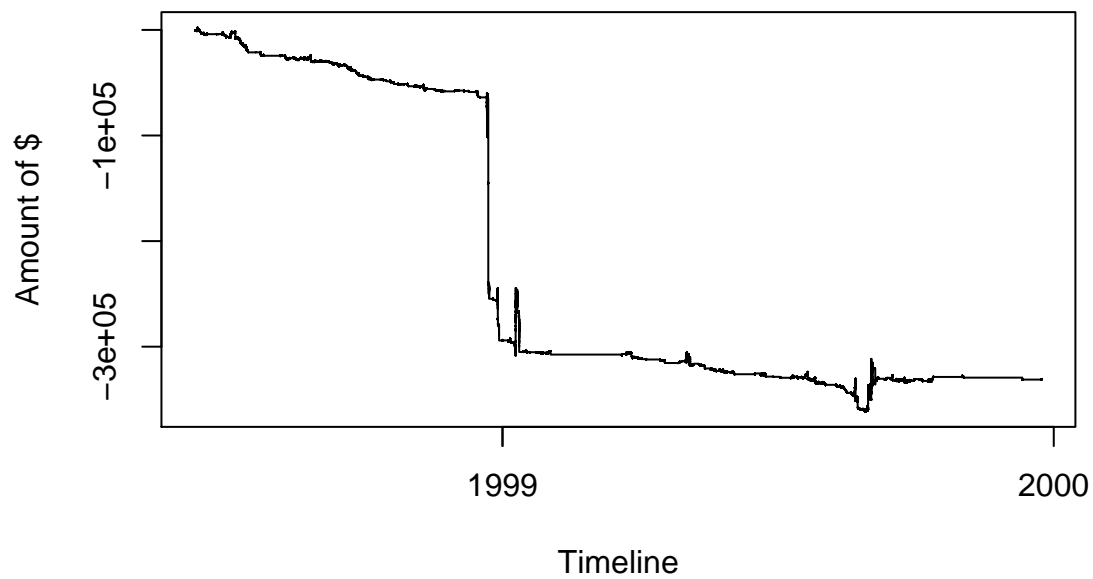
Bankroll of kfish



Bankroll of BlackBart



Bankroll of kwAAkbot



Sansa de ruina

Asa cum am observat distributia castigurilor converge catre o distributie normala, cu mici exceptii pentru cativa jucatori. Datele au fost analizate pentru jucatorii consistenti cu un numar ridicat de maini jucate. Dar

pentru a juca un numar mare de maini inraitii trebuie sa evite falimentul.

Asadar, am urmarit sa analizam si riscul ca un jucator sa ajunga falit, impiedicandu-l astfel sa isi proiecteze castigurile cu ajutorul curbei gaussiene. Desi mainile de poker nu urmeaza intocmai distributia normala, se apropie destul, cat sa putem folosi formula renumita propusa de Chen si Ankenman in cartea *The Mathematics of Poker* si anume:

$$P(B = 0) = \exp\left(\frac{-2\mu b}{\sigma^2}\right)$$

Pentru a testa validitatea acestei formule pentru setul nostru de date am realizat aprox 10.000 de simulari Monte-Carlo per suma initiala si am facut diferenta.

Rezultatele au fost surprinzatoare: Pe masura ce dimenensiunea bankroll-ului creste, creste si diferenta intre rezultate, care pentru sume mai mici era aproape nesemnificativa.

```
library(PokerStats)

for(i in 5:10){
  pdata = filter(data, p_name == fpfreq$p_name[i])
  df = table_RoR(pdata, as.character(fpfreq$p_name[i]))
  grid.table(df)
}
```

	Bankroll.size	Simulated.RoR	Gaussian.RoR	difference
1	10	0.9986	0.9998	0.0012
2	100	0.9976	0.9983	7e-04
3	500	0.9952	0.9917	0.0035
4	1000	0.9923	0.9835	0.0088
5	3000	0.9839	0.9519	0.032
6	5000	0.9759	0.922	0.0539
7	10000	0.9503	0.8538	0.0965
8	25000	0.8689	0.6941	0.1748
9	1e+05	0.3128	0.3431	0.0303

O posibila explicatie ar fi ca formula propusa in carte presupune ca jucatorul joaca o infinitate de maini, sau cel putin pana cand pierde toti banii. O a doua presupunere este ca suma se modifica doar pentru mainile jucate, daca este cazul, adica jucatorii nu pot extrage bani. Dar aceasta a doua proprietate a fost verificata. Asadar ramanem cu faptul ca jocurile reale de poker sunt limitate, deci drumul Gaussian va converge la zero mai repede, rezultand o probabilitate mai mare. Este posibilul si reversul, caci graficele arata o pondere mai mare catre extremele graficului, de unde simularile ar putea avea de suferit, desi nu a fost cazul in testele noastre.

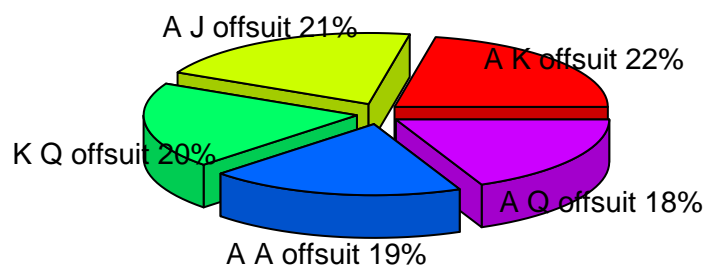
Maini castigatoare

Din perspectiva mainilor castigatoare ne-a interesat sa vedem daca jucatorii profita la maxim de cartile bune si daca au “maini preferate”. Rezulatele au fost destul de previzibile.

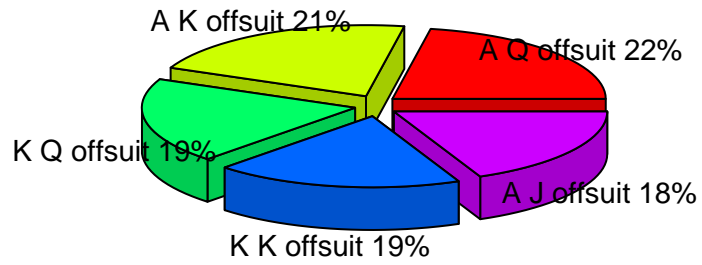
```
library(PokerStats)

for(i in 5:10){
  pdata = filter(data, p_name == fpfreq$p_name[i])
  plot_top_hands(pdata, as.character(fpfreq$p_name[i]), 5)
}
```

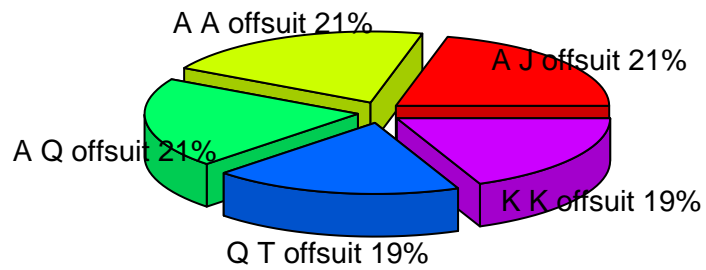
Top winning hands of kwAAkbot



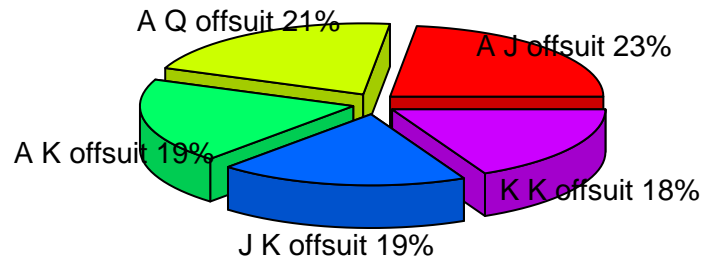
Top winning hands of pokibot1



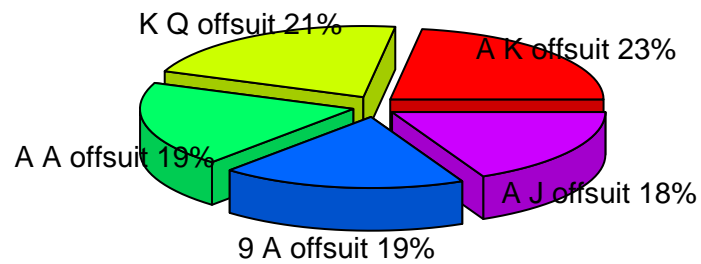
Top winning hands of Swagger



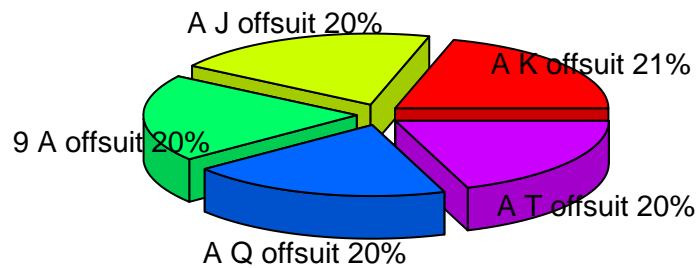
Top winning hands of MegaHertz



Top winning hands of going



Top winning hands of RiverRatt



In majoritatea topurilor se afla perechile cele mai puternice (AA, KK) si in general combinatii de A cu un kicker puternic (Q, J, 10, 9). Ceea ce a fost surprinzator este ca majoritatea acestor combinatii care s-au dovedit castigatoare au avut carti cu simbol diferit, reducand sansa jucatorilor sa obtina o chinta. Nu se pot observa “maini preferate” in particular, semn ca jucatorii adopta o tactica care sa maximizeze EV-ul. Este probabil si un motiv bun pentru care acestia au reusit sa joace nu numai atat de mare de maini, in jocul de poker flexibilitatea situationala face diferenta. Putem spune ca cei care reusesc sa converteasca un avantaj statistic in bani sunt cei care vor fi in joc cea mai lunga perioada de timp.

Concluzii

Per total proiectul a reprezentat o sansa foarte buna de a ne obisnui cu sintaxa limbajului si cu elementele referitoare la probabilitati si statistica.

Dataset-ul pe care l-am gasit s-a dovedit a fi mai mult decat satisfacator, dupa ce am reusit sa il aducem intr-un format usor de prelucrat si cu viteza mai mare de incarcare.

Contrar asteptarilor, in urma analizei datelor a numerosi jucatori am observat cum performantele acestora nu depind in proportii semnificative de calitatea cartilor dintr-o anumita partida. Balanta inclinata spre profit apartine jucatorilor cu o strategie dobandita in timp. Analiza are la baza o serie consistenta de date, motiv pentru care putem concluziona necesitatea unei strategii pentru o evolutie de succes in acest domeniu.

Referinte

Chen, William. 2006. *The Mathematics of Poker*.

Stein, Mike. 2011. “How Far from Gaussian (Normal) Are Poker Results,” February. <http://www.quantitativepoker.com/2011/02/how-far-from-gaussian-normal-are-poker.html>.