EDA of Four Horses in a Race

150000760

Executive Summary

Our report finds that the maximum amount of winnings possible only betting before the race on the winning horse is 11976.60 GBP, taking a lay position that would cover the back stake, if the winning horse had actually lost. The total lay and back stakes were 70.28 + 1303.47 = 1373.75, so a total percentage profit of 770%.

The expectation and variance shows that gamblers price things somewhat correctly, that is they give each horse an equal shot, and have tremendous anxiety.

Relevant plots are redone in SAS, to prove repeatability.

Introduction

This report analyzes betting exchange data for three randomly chosen horses and the winner. After data cleaning and Exploratory Data Analysis we determine the maximum amount of profit that could be made using an arbitrage position for the winner and the losers.

Backing a horse is betting that that horse will win. Laying a horse is betting that that horse will lose. Given n horses in a race, n-1 of them will lose, so the probability that a horse loses is higher than the probability that a given horse wins. A stake is the amount of money you will place on a bet, which is capped by the total volume in the market.

Odds are the multiple offered on a stake for a given bet, laying or backing. Odds given are intrinsically related to the probability that a certain event will occur. This is because the goal of a fair bet is to make the expected value zero.

A beautiful analysis on whether such gambling is worth doing is Daniel Bernoulli's "Exposition of a New Theory on the Measurement of Risk" from 1738. It is available on JSTOR.

Exploratory Data Analysis

With any dataset, it is good to see summary stats.

```
##       time                      marketStatus         inplay
##  Min.   :2018-06-20 15:29:02   Length:111276      Mode :logical
##  1st Qu.:2018-06-20 15:38:13   Class :character   FALSE:96030
##  Median :2018-06-20 15:48:34   Mode  :character   TRUE :15246
##  Mean   :2018-06-20 15:48:34
##  3rd Qu.:2018-06-20 15:58:43
##  Max.   :2018-06-20 16:08:45
##    competitor       competitorStatus    backPrice1
##  Min.   :  868018   Length:111276      Length:111276
##  1st Qu.: 8864290   Class :character   Class :character
##  Median :11162146   Mode  :character   Mode  :character
##  Mean   :10210545
##  3rd Qu.:11931592
##  Max.   :13373355
##  backVolume1         layPrice1          layVolume1
##  Length:111276      Length:111276      Length:111276
```

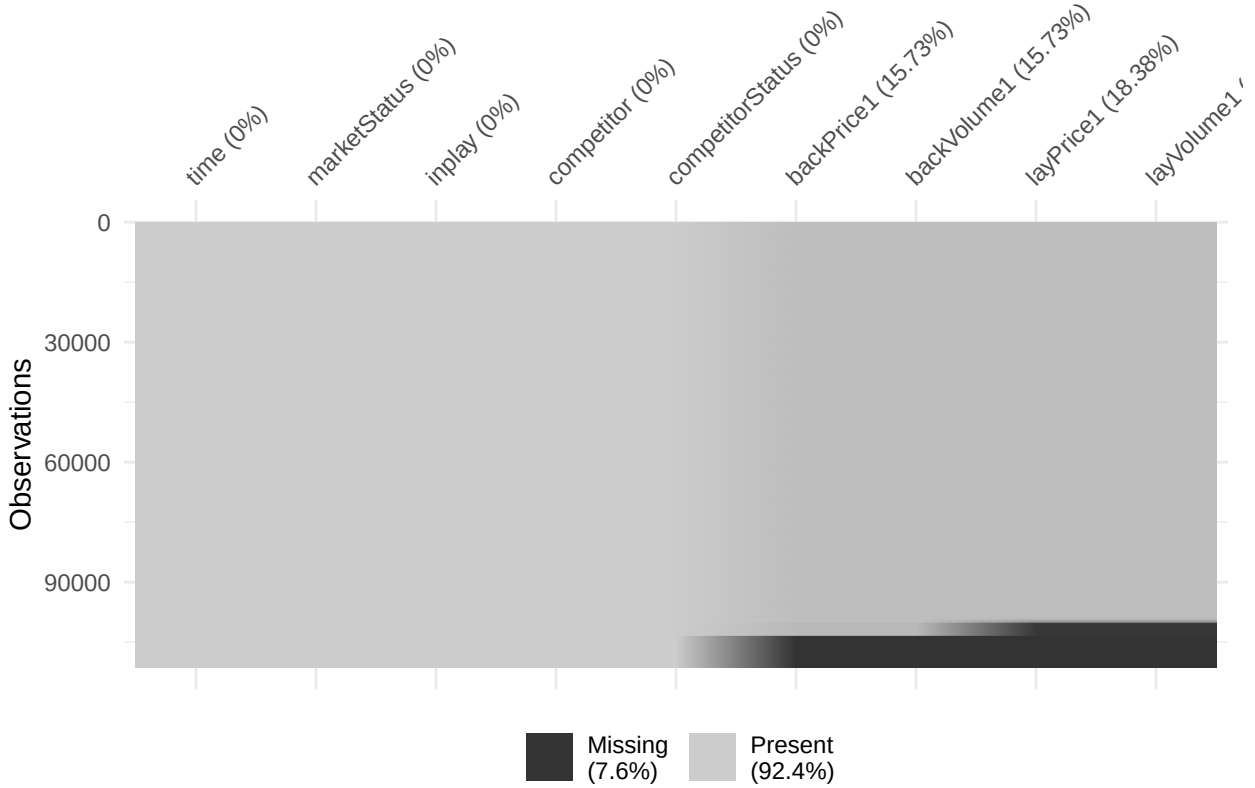
```
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

The data is not nice! We would like Price and Volume columns to be numeric so that we may manipulate such calculations.

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion
```

There is quite a number of NaNs! Can we visualize them?

```
library(naniar)
vis_miss(HorseData, warn_large_data = FALSE)
```



For Ease of analysis, we would like to remove NaN values, but important information - whether the Horse won or lost - is in these columns. A Column is formed of all False Booleans, then the competitor 11473056 with the winning horse is changed to true.

```
HorseData$Winner <- FALSE
HorseData$Winner[HorseData$competitor == 11473056] <- TRUE
```

Now the NaNs are removed.

```
HorseData <- na.omit(HorseData)
```

And finally, two new columns are added for ease of analysis

```
HorseData$backVP <- HorseData$backPrice1 * HorseData$backVolume1
HorseData$layVP <- HorseData$layPrice1 * HorseData$layVolume1
```

```
summary(HorseData)
```

```
##       time                       marketStatus         inplay
##  Min.   :2018-06-20 15:29:02   Length:90721       Mode :logical
##  1st Qu.:2018-06-20 15:37:15   Class :character   FALSE:87300
##  Median :2018-06-20 15:46:13   Mode  :character   TRUE :3421
##  Mean   :2018-06-20 15:46:29
##  3rd Qu.:2018-06-20 15:55:34
##  Max.   :2018-06-20 16:04:43
##    competitor       competitorStatus     backPrice1      backVolume1
##  Min.   :  868018   Length:90721       Min.   :  1.03   Min.   :   2.00
##  1st Qu.: 8846010   Class :character   1st Qu.: 19.50   1st Qu.:  33.11
##  Median :11295111   Mode  :character   Median : 29.00   Median :  97.01
##  Mean   :10146425                      Mean   : 64.09   Mean   : 213.11
##  3rd Qu.:11985643                      3rd Qu.: 70.00   3rd Qu.: 226.82
##  Max.   :13373355                      Max.   :670.00   Max.   :4005.50
##    layPrice1         layVolume1        Winner            backVP
##  Min.   :   1.04   Min.   :   2.00   Mode :logical   Min.   :    5.1
##  1st Qu.:  20.00   1st Qu.:  15.25   FALSE:87690     1st Qu.: 1718.9
##  Median :  30.00   Median :  54.16   TRUE :3031      Median : 3928.9
##  Mean   :  72.81   Mean   : 153.06                   Mean   : 5266.9
##  3rd Qu.:  80.00   3rd Qu.: 153.90                   3rd Qu.: 6801.1
##  Max.   :1000.00   Max.   :3063.73                   Max.   :50068.8
##      layVP
##  Min.   :    2.08
##  1st Qu.:  942.54
##  Median : 2256.76
##  Mean   : 4014.92
##  3rd Qu.: 4961.00
##  Max.   :41360.36
```

Variance and Expectation analysis

There are in total, 33 unique competitors in the two races.

```
unique(HorseData$competitor)
```

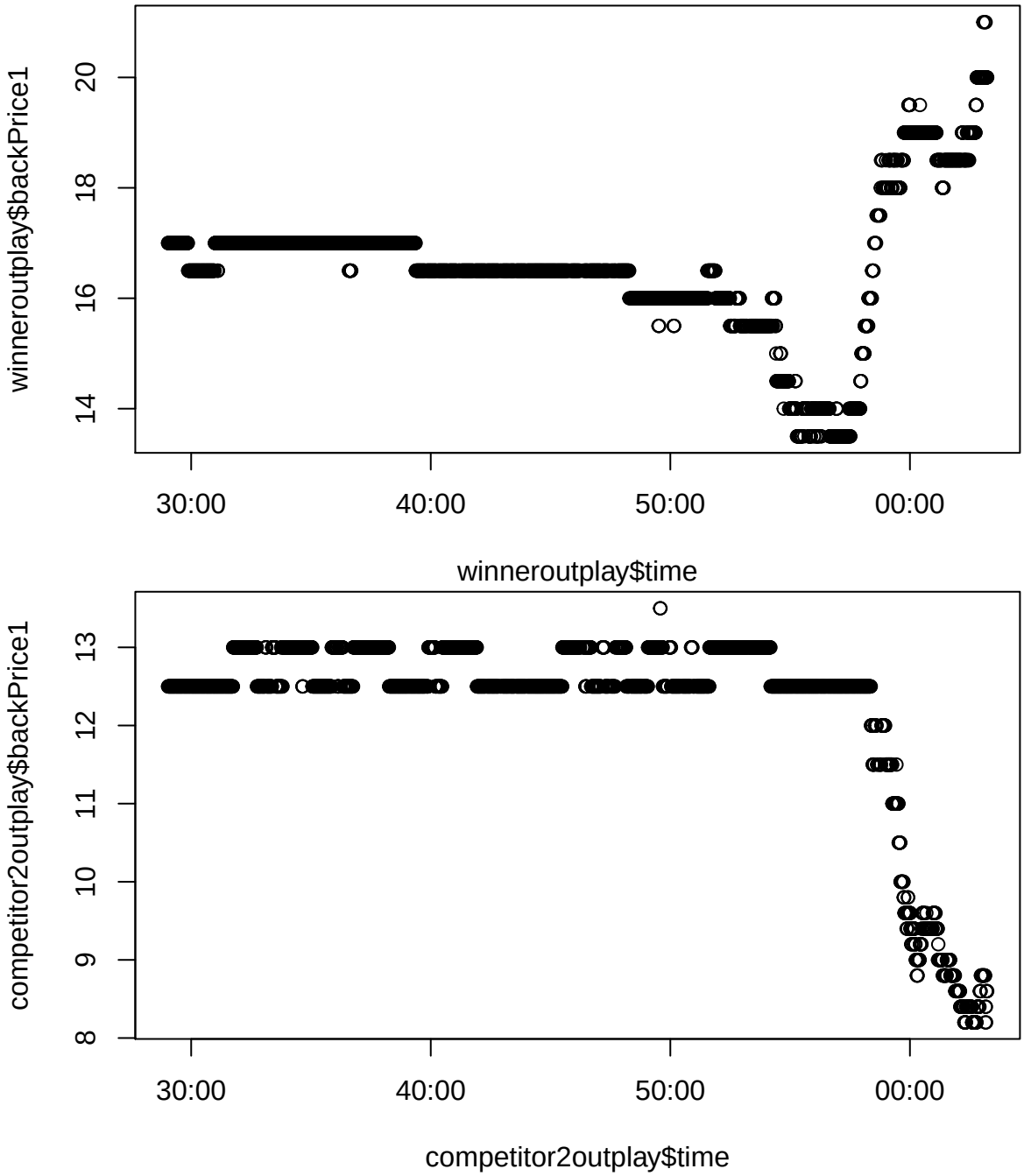
```
##  [1] 11817468 11538828  6044572   868018 12886815 10308467 11931592
##  [8] 11473056  8846010 13373355 12666330 12452848 12192132 10058972
## [15] 11314111  5521783  9175949  8565296  8692300 12150386  7579136
## [22]  9748889  9977366  8864290 12886814 11295113 11295111 11985643
## [29]  8528919 10339376
```

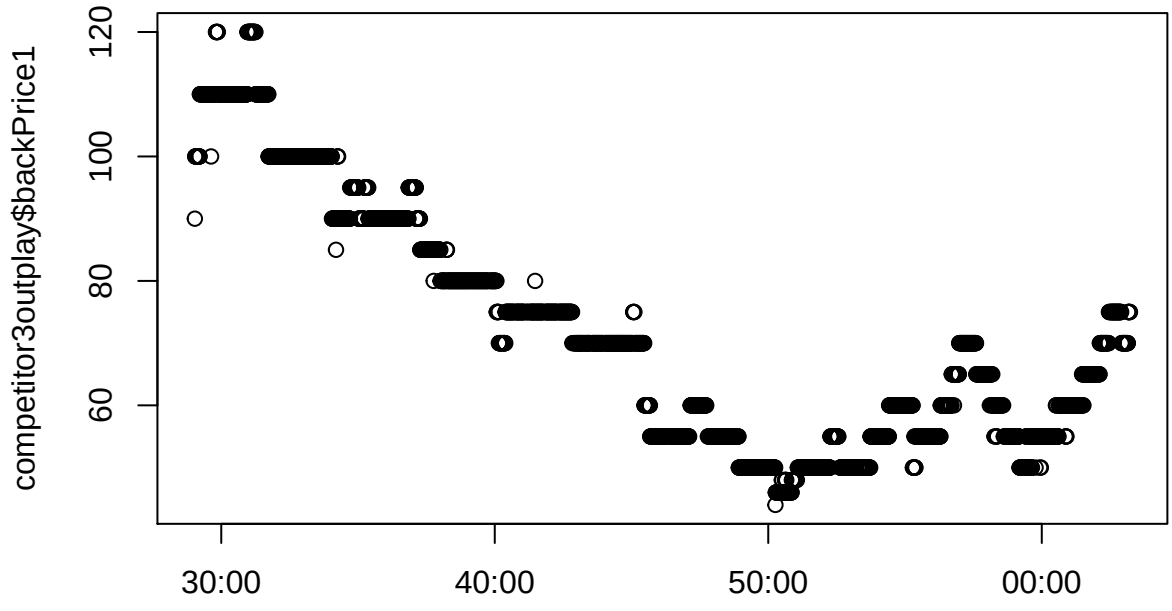
```
sorted_data <- HorseData[order(HorseData$competitor, HorseData$time),]
```

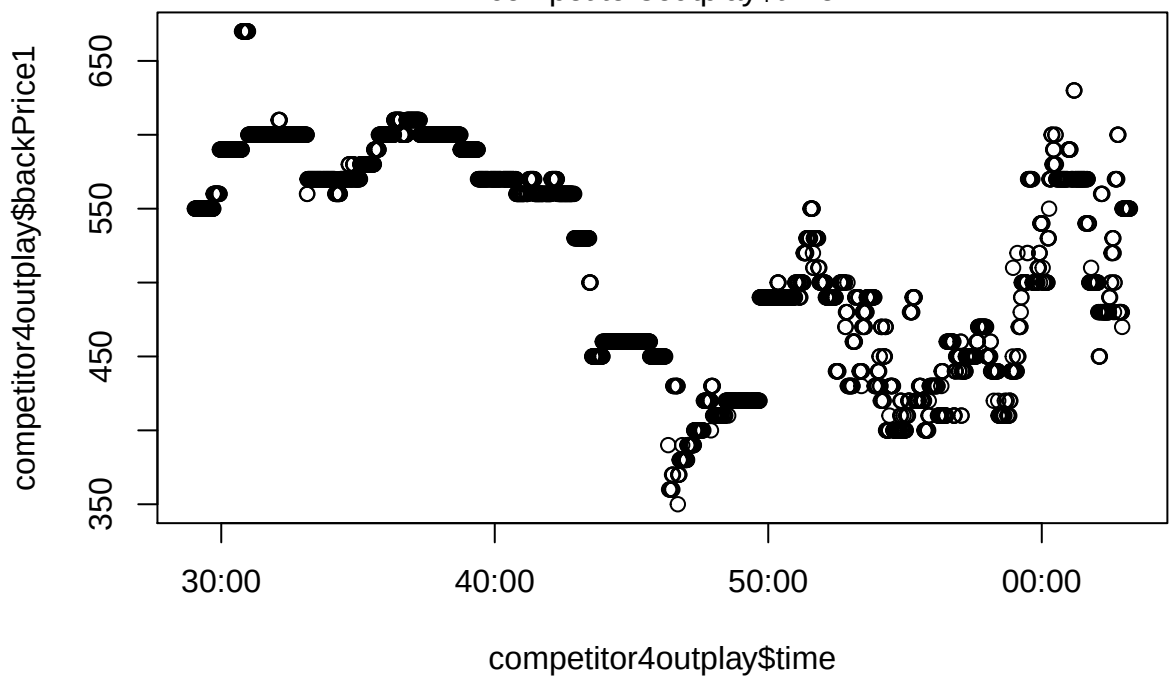
The following four competitors were randomly chosen. For ease of data replication, the extracted datasets were exported to CSV.

```
winner<-sorted_data[which(sorted_data$competitor==11473056), ]
write.csv(winner,"./winner.csv", row.names=FALSE)
competitor2<-sorted_data[which(sorted_data$competitor==11538828), ]
write.csv(competitor2,"./competitor2.csv", row.names=FALSE)
competitor3<-sorted_data[which(sorted_data$competitor==9977366), ]
write.csv(competitor3,"./competitor3.csv", row.names=FALSE)
competitor4<-sorted_data[which(sorted_data$competitor==10339376), ]
write.csv(competitor4,"./competitor4.csv", row.names=FALSE)
```

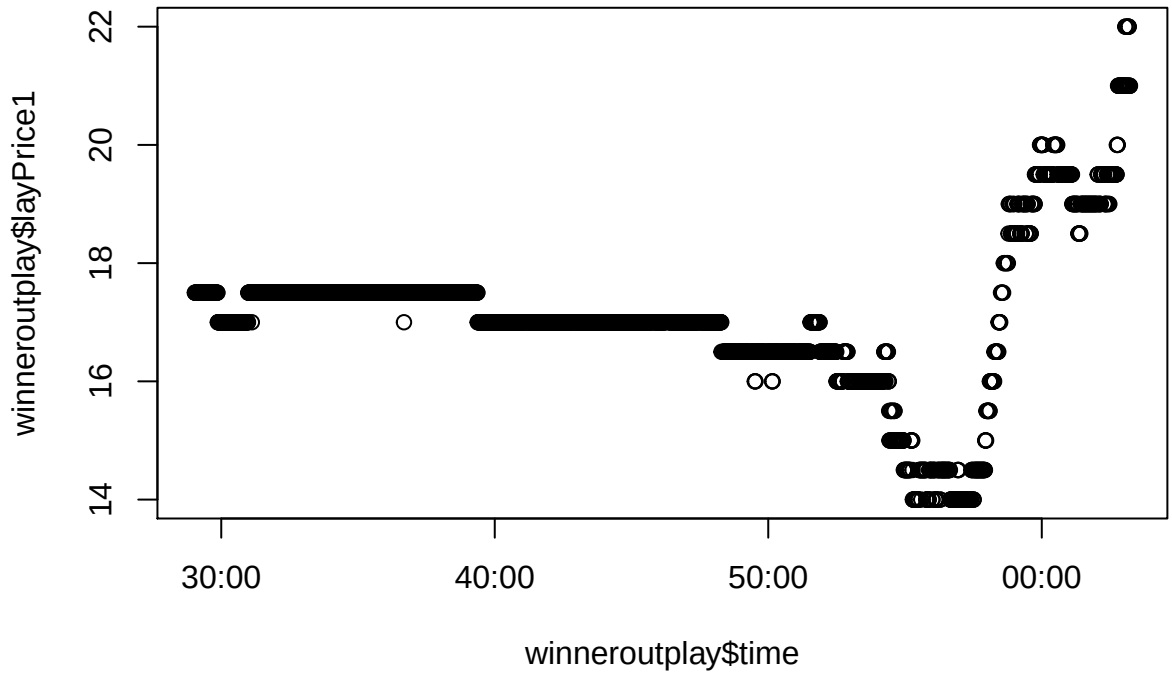
Back and Lay odds trace for pre-inplay data.



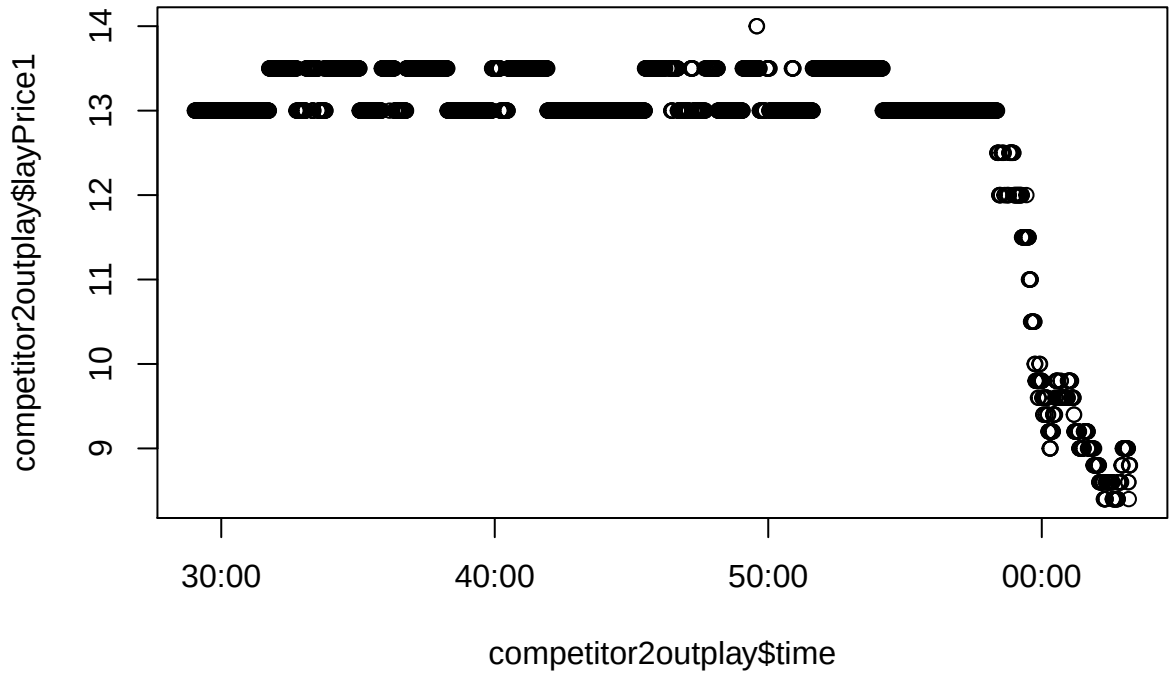




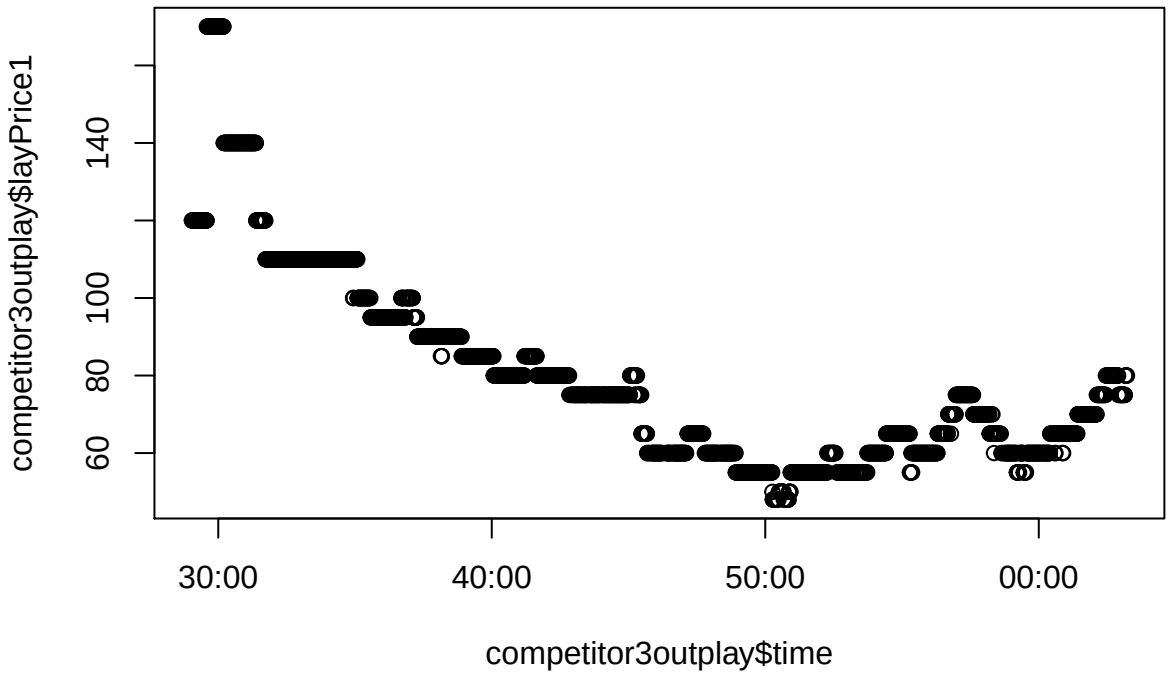

```
plot(winneroutplay$time,winneroutplay$layPrice1)
```



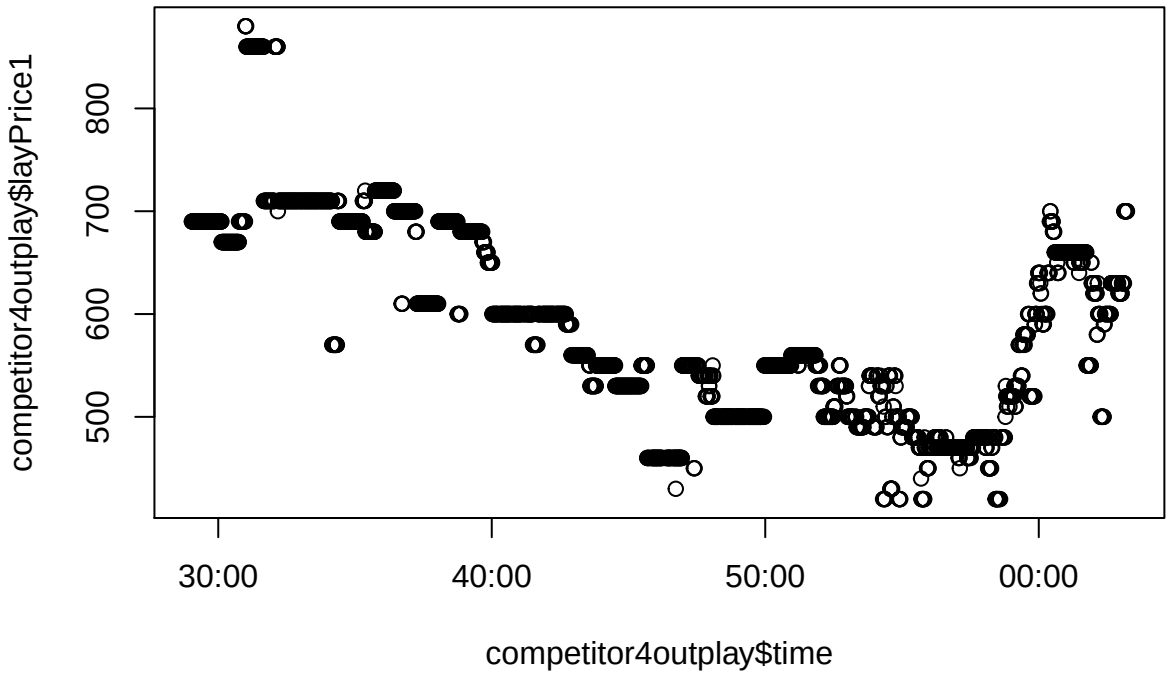

```
plot(competitor2outplay$time,competitor2outplay$layPrice1)
```




```
plot(competitor3outplay$time,competitor3outplay$layPrice1)
```




```
plot(competitor4outplay$time,competitor4outplay$layPrice1)
```



###

Mean/Variance for each player without normalisation.

Winner mean and variance

```
mean(winneroutplay$layPrice1)
```

```
## [1] 17.07234
```

```
var(winneroutplay$layPrice1)
```

```
## [1] 1.701624
```

```
mean(winneroutplay$backPrice1)
```

```
## [1] 16.55447
```

```
var(winneroutplay$backPrice1)
```

```
## [1] 1.625169
```

Competitor 2 mean and variance

```
mean(competitor2outplay$layPrice1)
```

```
## [1] 12.7268
```

```
var(competitor2outplay$layPrice1)
```

```
## [1] 1.630962
```

```
mean(competitor2outplay$backPrice1)
```

```
## [1] 12.25127
```

```
var(competitor2outplay$backPrice1)
```

```
## [1] 1.41603
```

Competitor 3 mean and variance

```
mean(competitor3outplay$layPrice1)
```

```
## [1] 80.86976
```

```
var(competitor3outplay$layPrice1)
```

```
## [1] 645.597
```

```
mean(competitor3outplay$backPrice1)
```

```
## [1] 72.63265
```

```
var(competitor3outplay$backPrice1)
```

```
## [1] 370.5041
```

Competitor 4 mean and variance

```
mean(competitor4outplay$layPrice1)
```

```
## [1] 594.0034
```

```
var(competitor4outplay$layPrice1)
```

```
## [1] 9043.953
```

```
mean(competitor4outplay$backPrice1)
```

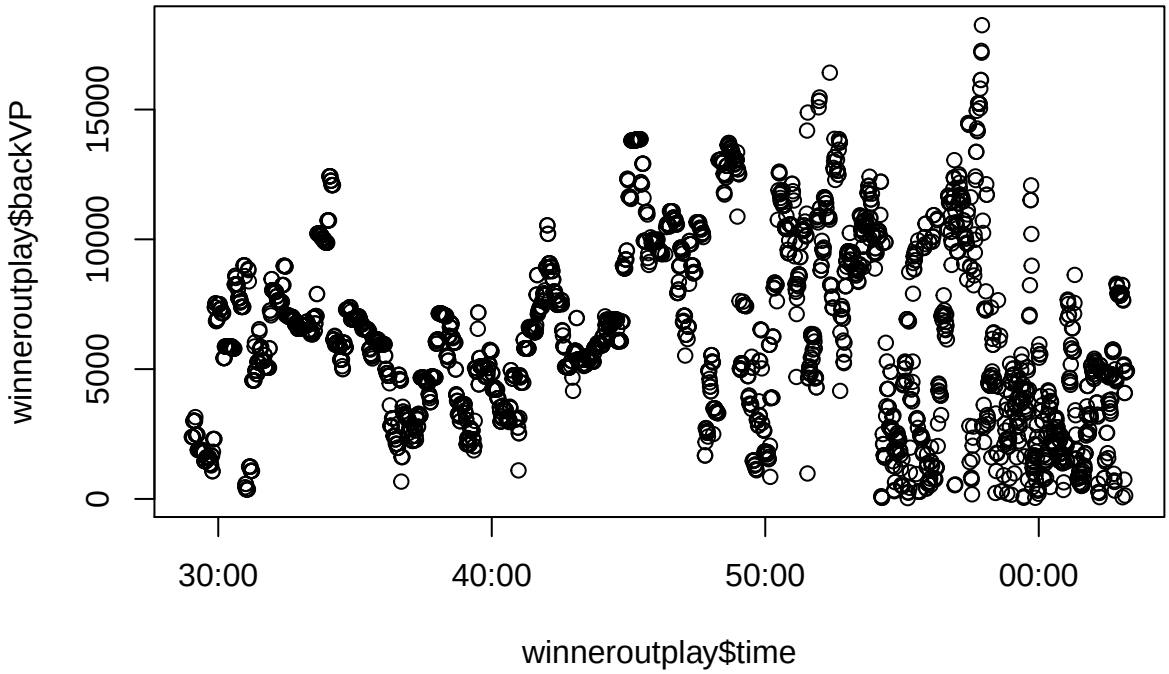
```
## [1] 517.4983
```

```
var(competitor4outplay$backPrice1)
```

```
## [1] 5010.515
```

Mean and variance do not tell you much. However competitor 3 and 4 appear to be more volatile, and reflects uncertainity in the markets.

How much money could you have made on the winner?




```
## [1] 18248.58
```

```
## [1] "2018-06-20 15:57:55 UTC"
```

```
## [1] 1303.47
```

So the most money you could have made is by backing the winner at 15:56:50.

What was the stake you needed?

```
## [1] 1303.47
```

So could you have found a lay, so that the 437.14 stake would have been recovered if you lost?

```
## [1] "2018-06-20 15:29:58 UTC"
```

```
## [1] 70.28
```

```
## [1] 17
```

So your overall earnings if you backed and laid at the time indicated above would be 11976.2

```
## [1] 11976.6
```

Your overall return would be 11976.2 on your initial investment. Not so bad!