Computational Statistics - Lab 5

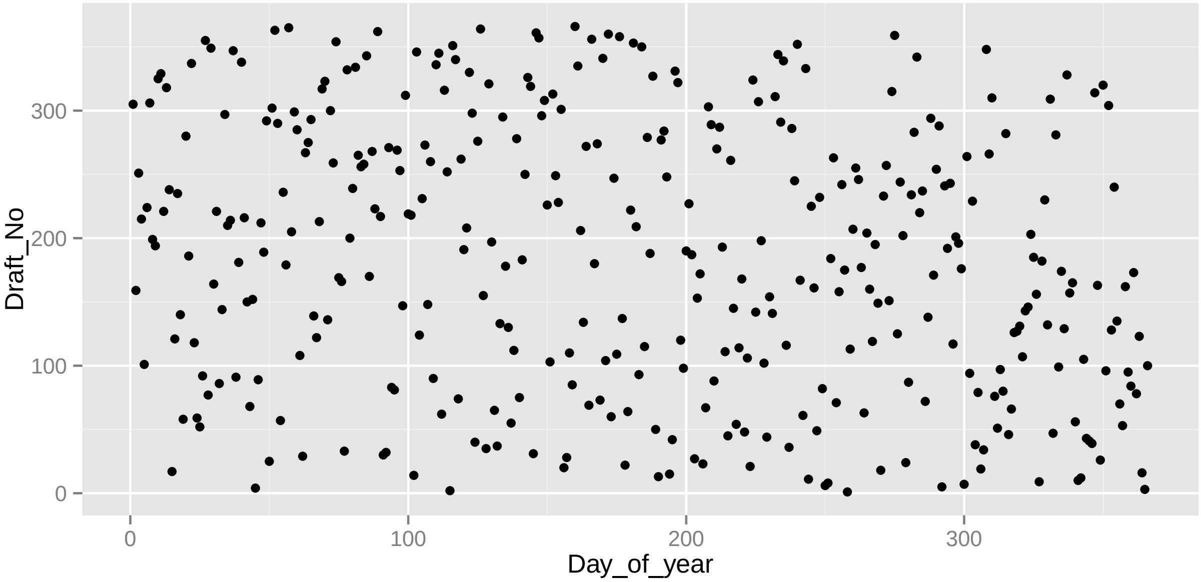
Gustav Sternelöv

March, 10, 2016

Assignment 1

1.1

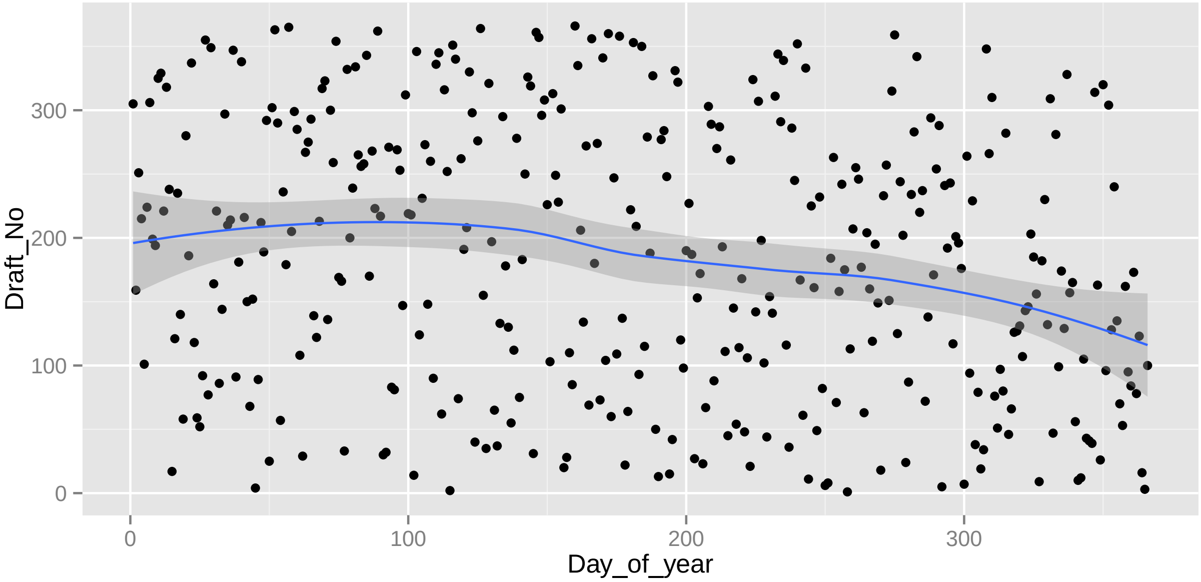
From the *lottery* data set are the variables day of year and draft number plotted against each other in order to investigate the randomness for the lottery.



By looking at the scatterplot it is hard to say anything else than that the lottery looks random. No trend or any other non-random pattern is thought to be seen in the plot. Instead, the values seem to be rather evenly spread out over the whole graph.

1.2

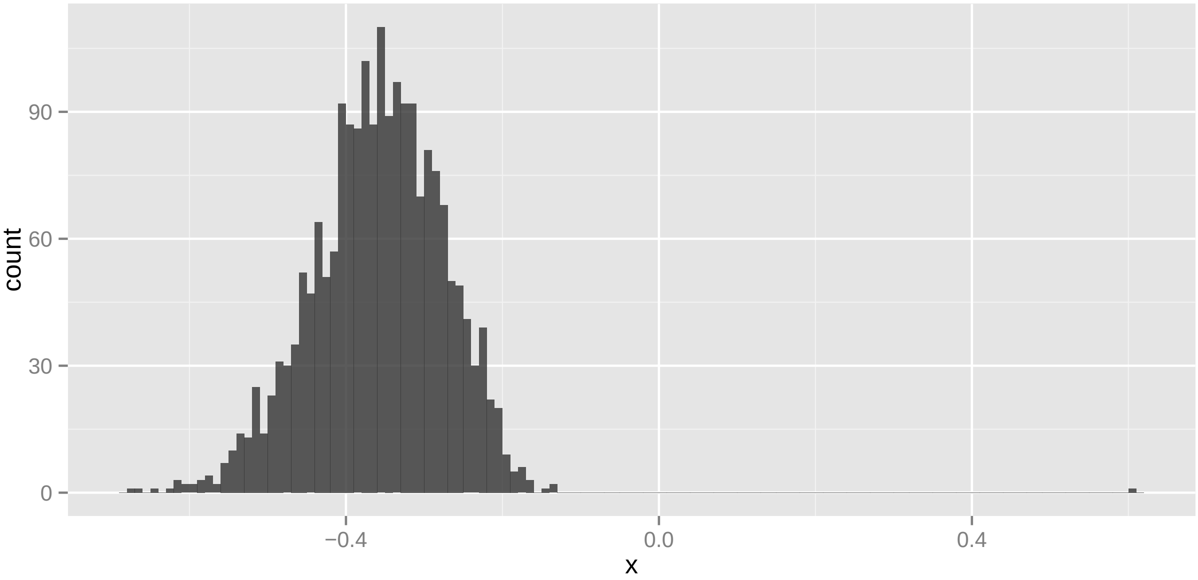
An estimate of $\hat{Y}$, the predicted draft number, is computed with the *loess* function and added to the graph that was presented in *1.1*.



There seem to be a negative trend from around the day 150 on the x-axis and so forth. This trend causes uncertainty over the randomness for the lottery procedure.

1.3

To further investigate the randomness of the lottery is a non-parametric bootstrap used for estimating the distribution of T, the test statistic of interest. T is specified in the following way:
$T = \frac{\hat{Y}(X_b)-\hat{Y}(X_a)}{X_b-X_a}, WhereX_b = argmax_X\hat{Y}, X_a = argmin_X\hat{Y}$
The generated T values are plotted in the histogram below. A T value that is significantly greater than zero is interpreted as there is a trend in the data, hence the lottery is not random.



The P-value of the test concerining if the lottery is random or not is conducted in the following way where B is the number of bootstrap samples, $\frac{\sum T\geq 0}{B}$
Thus, the estimated p-value is 0.0005 and the conclusion is that the lottery not is random.

1.4

The randomness of the lottery is checked with one more test. This time the test consists of the hypothesis H_0 : Lottery is random vs H_a : Lottery is not random.
The test is conducted by using a permutation test with the test statistic T. A function is used to implement the test and returned is the P-value of the test.

The returned p-value is 0.0725. Depending on the choice of significance level the conclusion about the randomness differ since the p-value is relatively low. However, if a the significance level is chosen to be 0.05, a common choice, the conclusion is that H_0 not can be rejected (the lottery is random).

1.5

a-b)

With α equal to 0.1 and B=200 the received p-value for the test is 0, so the null hypothesis is rejected.

c)

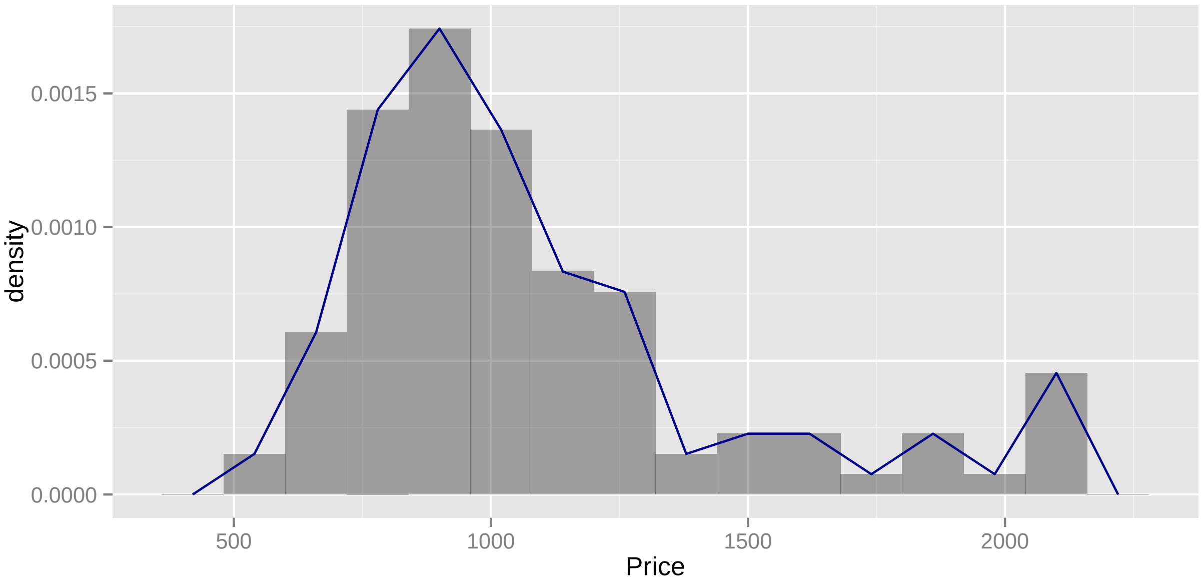
The p-values for α values from 0.2 to 10, by steps of 0.1, are presented below. Since all p-values are very close to 0 the power is equal to 1. This means that the quality of the test statistics is very good.

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [71] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Assignment 2

2.1

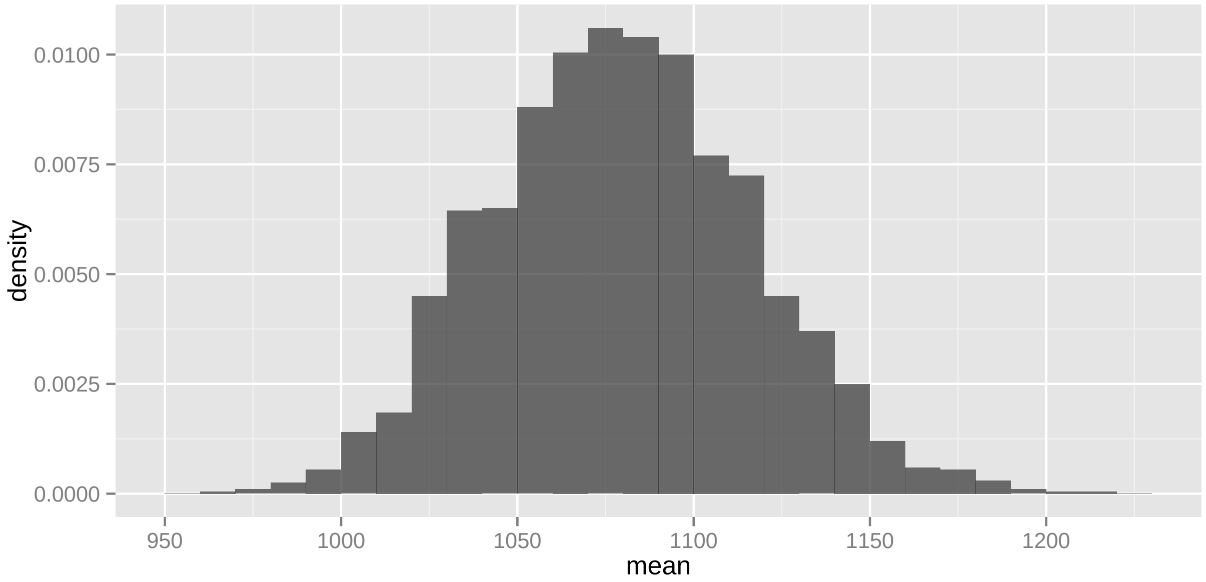
The histogram for the variable *Price* is plotted below.



It is not totally evident but the distribution might be thought to remind of a chi-square distribution.
The mean price is 1080.47.

2.2

Before bootstrap is used for estimating the distribution for the mean price of the houses it needs to be determined if it is a parametric or non-parametric bootstrap that best suites the problem. Since I not am entirely convinced of which distribution the histogram in *2.1* reminds of, a non-parametric bootstrap is chosen for estimating the distribution.
The distribution for the mean price obtained by a non-parametric bootstrap with B=2000 is plotted below.



The bootstrap bias-correction is obtained by first calculating the mean for the whole sample, 1080.47. Then, the mean of all the bootstrap means is computed, 1079.59. The bootstrap bias correction then is 2∗1080.47 -1079.59 = 1081.35.

The variance of the price of the mean is obtained by using a non-parametric bootstrap for the obtained estimates of mean and calculating the variance for each bootstrap sample. Again, B is set to 2000 and the estimated variance of the price of the mean is 1356.37.

The 95 % confidence interval for the mean price using bootstrap percentile, BCa and first-order normal approximation is presented by the table below. The *lower* and *upper* columns gives the boundiares of the intervals and the column *length* gives the length of the respective intervals. The notable differences between the confidence intervals is that the BCa interval has just a little higher values than the two other and that the percentile interval is the shortest.

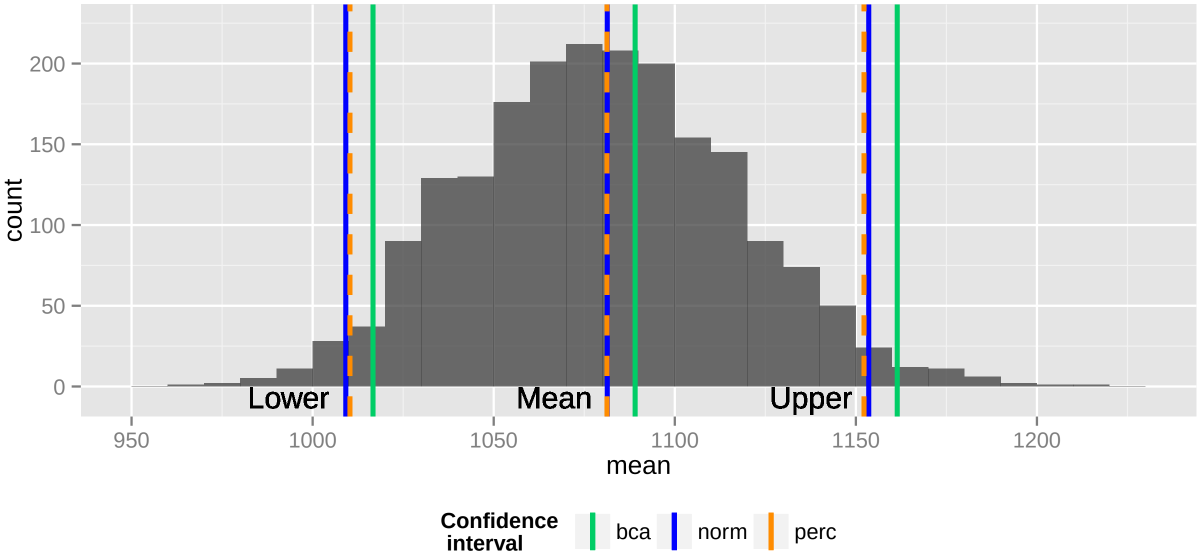
```
##      Lower    Upper   Length Method
## 1 1009.157 1153.551 144.3936   norm
## 2 1010.306 1152.248 141.9413   perc
## 3 1016.671 1161.423 144.7520    bca
```

2.3

The variance of the mean price with the jackknife method is 1320.91. Compared to the bootstrap estimate is the jackknife estimate of the variance clearly higher. This is not a surprising result since the variance often is overestimated with the jackknife method.

2.4

The following plot visualises a comparison of the confidence intervals presented in *2.2*. The upper and lower boundary as well as the location of the mean for each interval is shown in the plot.



The shortest interval is obtained for the percentile confidence interval, although the difference between the lengths of the intervals is very small. The estimated means are very similar for the normal and the percentile interval and a bit higher for the BCa interval. The confidence interval with the mean closest to the mean for the whole data set is the percentile interval with a mean of 1081.28. The mean for the normal confidence interval is very similar, 1081.35 and for the BCa is much higher, 1089.05.

Appendix

```
library(ggplot2)
require(XLConnect)
library(boot)
### Assignment 1 ###
wb = loadWorkbook("C:/Users/Gustav/Documents/Computational-Statistics/Lab5/lottery.xls")
lottery = readWorksheet(wb, sheet = "sheet1", header = TRUE)
# 1.1
ggplot(lottery, aes(y=Draft_No, x=Day_of_year)) + geom_point()
# 1.2
ggplot(lottery, aes(y=Draft_No, x=Day_of_year)) + geom_point() + geom_smooth(method="loess")
stat1<-function(data,n){
  data1=data[n,]
  loessM <- loess(Draft_No ~ Day_of_year, data=data1)
  Xa <- which.min(loessM$fitted)
  Xb <- which.max(loessM$fitted)
  y_Xa <- loessM$fitted[Xa]
  y_Xb <- loessM$fitted[Xb]
  T_val <- (y_Xb - y_Xa)/(data1$Day_of_year[Xb] - data1$Day_of_year[Xa])
  ret <- T_val
  return(ret)
}
set.seed(311015)
res1=boot(lottery,stat1,R=2000)
res1Dat <- data.frame(x=res1$t, index=1:2000)
ggplot(res1Dat, aes(x)) + geom_histogram(binwidth=0.01, alpha=.8)
options(scipen=999)
# 1.4
permFunc <- function(data, B){
  T_value=numeric(B)
  n=dim(data)[1]
  for(b in 1:B){
    Gb=sample(data$Day_of_year, n, replace = FALSE)
    data1 <- data.frame(Draft_No = data$Draft_No, day=Gb)
    loessM <- loess(Draft_No ~ day, data=data1)
    Xa <- which.min(loessM$fitted)
    Xb <- which.max(loessM$fitted)
    y_Xa <- loessM$fitted[Xa]
    y_Xb <- loessM$fitted[Xb]
    T_value[b] <- (y_Xb - y_Xa)/(data1$day[Xb] - data1$day[Xa])
  }
  loessMT <- loess(Draft_No ~ Day_of_year, data=data)
  Xa0 <- which.min(loessMT$fitted)
  Xb0 <- which.max(loessMT$fitted)
  y_Xa0 <- loessMT$fitted[Xa0]
  y_Xb0 <- loessMT$fitted[Xb0]
  T_0 <- (y_Xb0 - y_Xa0)/(Xb0 - Xa0)
  p_val <- mean(T_value > abs(T_0))
  return(p_val)
}
set.seed(311015)
lottery$Draft_No <- 0
```

```r
for(i in 1:366){
  lottery$Draft_No[i] <- max(0, min((0.1*lottery$Day_of_year[i] + rnorm(1, 183, 10)), 366))
}
alphas <- seq(0.2, 10, 0.1)
P_val <- 0
for(j in 1:length(alphas)){
  lottery$Draft_No <- 0
  for(i in 1:366){
    lottery$Draft_No[i] <- max(0, min(alphas[j]*lottery$Day_of_year[i] + rnorm(1, 183, 10), 366))
  }
  P_val[j] <- permFunc(lottery, 200)
}
P_val
wb2 = loadWorkbook("C:/Users/Gustav/Documents/Computational-Statistics/Lab5/prices1.xls")
prices = readWorksheet(wb2, sheet = "sheet1", header = TRUE)

ggplot(prices, aes(x=Price,..density..)) + geom_histogram(binwidth=120, alpha=0.4) +
  geom_freqpoly(binwidth=120, col="darkblue")
# 2.2
## Function for estimating the mean value
stat3<-function(data,n){
  data1=data[n,]
  res = mean(data1$Price)
  return(res)
}
set.seed(311015)
res3=boot(prices,stat3,R=2000)
priceBoot <- data.frame(mean = res3$t, index = 1:2000)
ggplot(priceBoot, aes(mean,..density..)) + geom_histogram(binwidth=10, alpha=0.7)
# Function for estimating the variance of the price of the mean
varBoot <-function(data,n){
  data1=data[n]
  res = (sum((data1 - mean(data1))^2)) * (1/(length(data1)-1))
  return(res)
}
set.seed(311015)
res4=boot(res3$t,varBoot,R=2000)
# 95 % C.I for the mean
CI <- boot.ci(res3, type=c("norm","perc", "bca"))
CIvals <- data.frame(rbind(CI$normal[2:3], CI$perc[4:5],CI$bca[4:5]), X3=c(CI$normal[3]-CI$normal[2], CI
names(CIvals) <- c("Lower", "Upper", "Length", "Method")
CIvals
## 2.3
T_star <- 0
for(j in 1:110){
  T_star[j] <- 110*mean(prices[,1]) - 109 * mean(prices[-j, 1])
}
J_T <- (1/110) * sum(T_star)
varJack <- sum((T_star-J_T)^2) / (110*109)
CIvals$Mean <- (CIvals$Lower+CIvals$Upper)/2

ggplot(priceBoot, aes(mean)) + geom_histogram(binwidth=10, alpha=0.7) + geom_vline(data=CIvals, aes(xin
  scale_linetype_manual(name="", values=c("solid", "solid","dashed"), guide=FALSE) +
  theme(legend.position = "bottom") +
  geom_text(aes(CIvals$Lower[1],0,label = "Lower", vjust = 1, hjust=1.2))+
  geom_text(aes(CIvals$Upper[1],0,label = "Upper", vjust = 1, hjust=1.2))+
  geom_text(aes(CIvals$Mean[1],0,label = "Mean", vjust = 1, hjust=1.2))+ ylim(-7, 225)
## NA
```