732A90: Computational Statistics

Computer lab5 - Group11

Sofie Jörgensen, Oriol Garrobé Guilera, David Hrabovszki

29 February 2020

Question 1: Hypothesis testing

1.

The aim of this task is to investigate whether or not a lottery is random. In this lottery, every day of the year `X` is given a draft number `Y`. The first approach is to make a scatter plot of `Y` versus `X` to see if these variables look correlated.

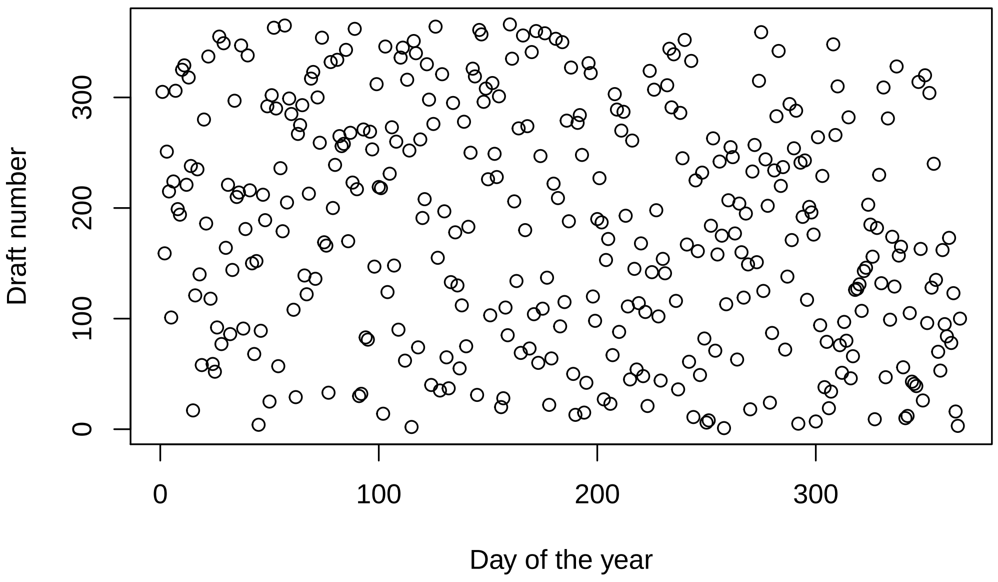

Figure 1: Scatterplot of Draft number versus Day of the year

The lottery looks random as there is no clear correlation between the variables in Figure 1.

2.

Using a loess smoother we can compute an estimate of `Y` as a function of `X` and plot the result. Therefore, we can state wheter the lottery still looks random.

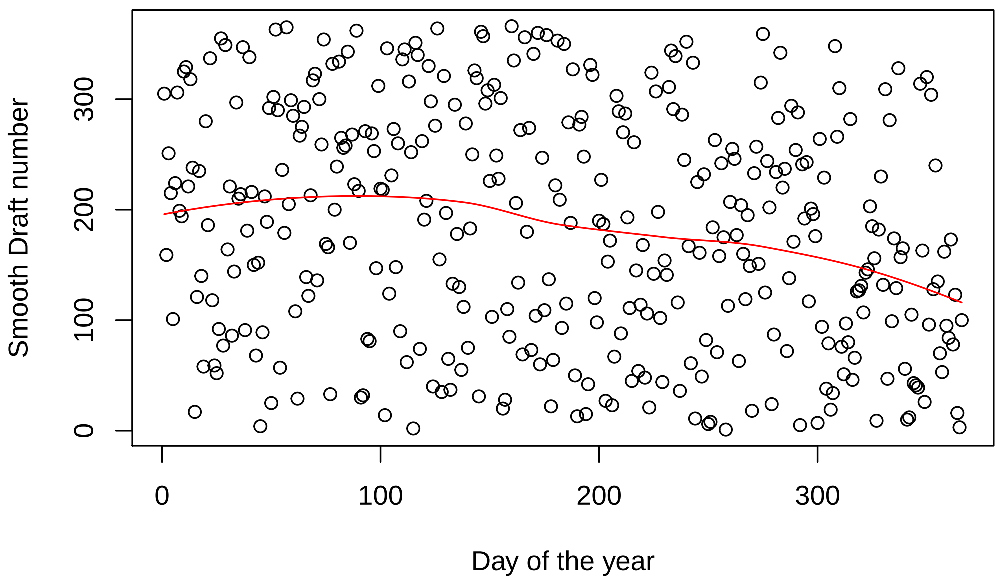

Figure 2: Scatterplot of Smooth Draft number versus Day of the year

In 2, the red smoothed line give an indication that the lottery does not look as random as before. The days at the beginning of the year have on average larger draft number that those at the end of the year.

3.

We can use a test statistic to conclude whether the test is random or not. We therefore use a non-parametric bootstrap with $B = 2000$ bootstraps and compute the p-value of the test. The code can be found in the Appendix.

Once performed the bootstrap, we can plot the result on an histogram to check if the statistic comes from a certain distribution. Also a normality test is performed and a Q-Q plot is given in Figure 3.

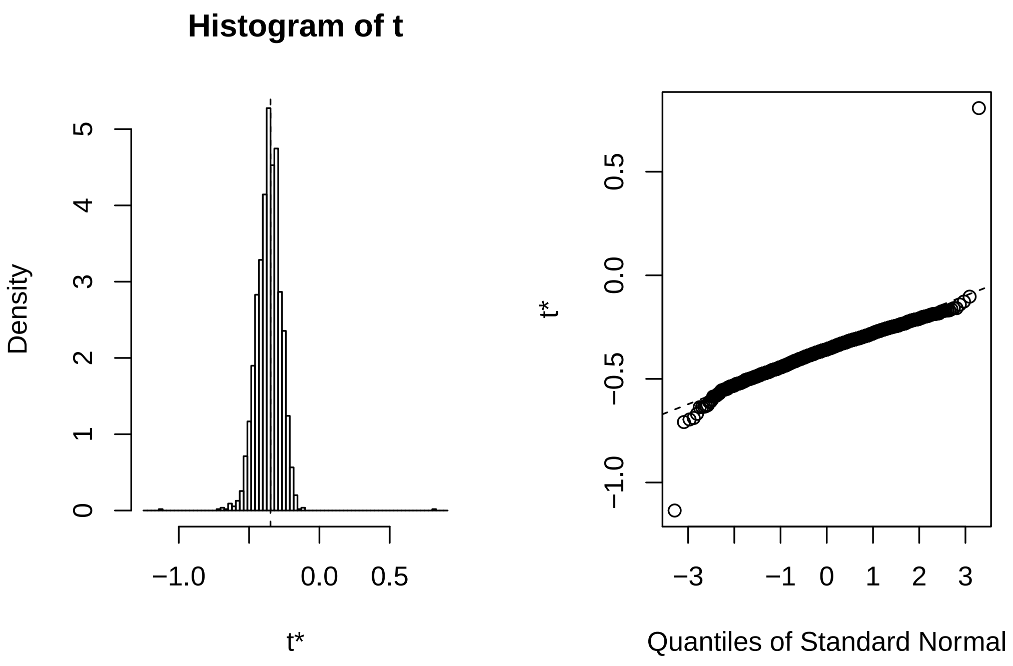

Figure 3: Histogram and Q-Q plot of the bootstrap

In Figure 3 we can see that the distribution is centered at -0.348 and reminds of a normal distribution. The Q-Q plot confirms it.

Finally, the p-value of the test is $5e-04$. This p-value is very small, therefore we can not reject the null hypothesis.

4.

In order to test the hypothesis,

$$H_0 : Lottery \quad is \quad random H_1 : Lottery \quad is \quad non-random$$

we use permutation test with statistic T, and return the p-value to decide the randomness of the draw. The code can be found in the Appendix.

The p-value of the permutation test is 0.0775. This p-value is large and we can reject the null hypothesis. The draw is not random.

5.

We want to estimate the power of the previous test. To do so we generate a new dataset using the same `X` as in the original dataset and `Y` follows the equation $Y(X) = max(0, min(\alpha x + \beta, 366))$, where $\alpha = 0.1$ and $\beta \sim N(183, sd = 10)$. The code can be found in the Appendix.

We now use the new dataset to run the previous test, and return the p-value of the test. The p-value of the power test is 1, this means that the test used never rejects the Null hypothesis when is actually true. The error type 2 is 0.

Finally we repeat the previous test but for $\alpha = 0.2, 0.3, ..., 10$ and return the p-values of each test. The p-values are the following:

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [77] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

We can see that the p-values are all 1. For all α the test never commits the Type 2 error, which means that the test never rejects the Null hypothesis when this one is true.

Question 2: Bootstrap, jackknife and confidence intervals

In this task we are going to estimate home prices in Albuquerque back in 1993, by using bootstrap, jackknife and confidence intervals. For this, we import the data set `prices1.csv` consisting of 110 observations, where `Price` is the variable of interest.

1.

First, we plot a histogram of the variable `Price`, to get an idea of its distribution.

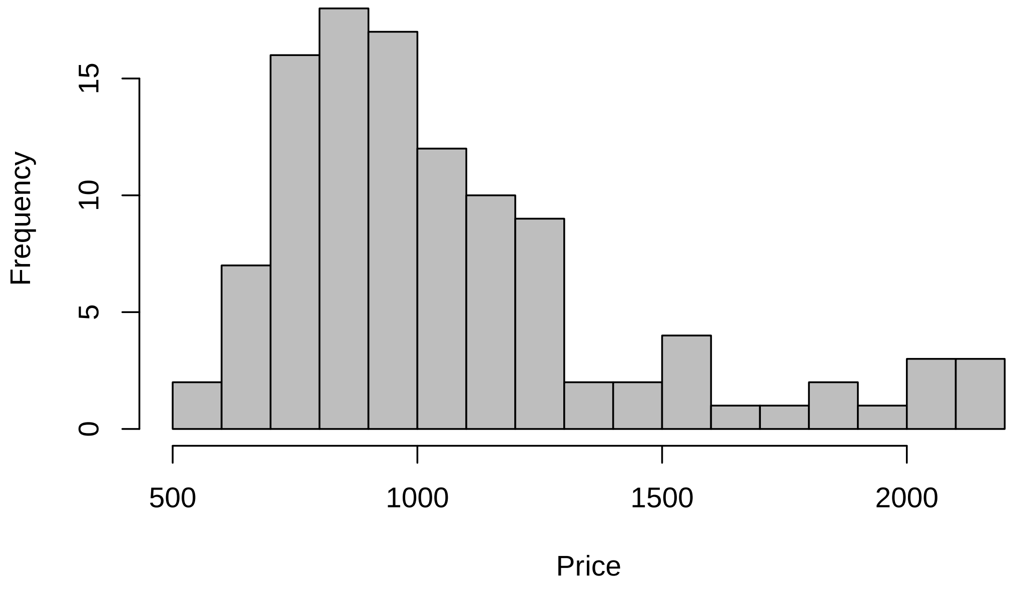

Figure 4: A histogram of the distribution of Price.

In Figure 4, we can observe that `Price` appears to be right-skewed, which reminds us of a Chi-squared distribution. The mean price of `Price` is computed to 1080.473.

2.

In the previous task, we presented a histogram of the price and computed the mean price. Now we are interested in estimating the distribution of the mean price by using bootstrap. The package `boot` provides appropriate functions for this purpose.

The following approach is being used: Draw $B = 2000$ bootstrap samples, i.e. resamples with replacement of size $n = 110$, and then compute the statistic, that is, the mean. Then we can form a bootstrap distribution that approximates the sampling distribution of the mean statistic.

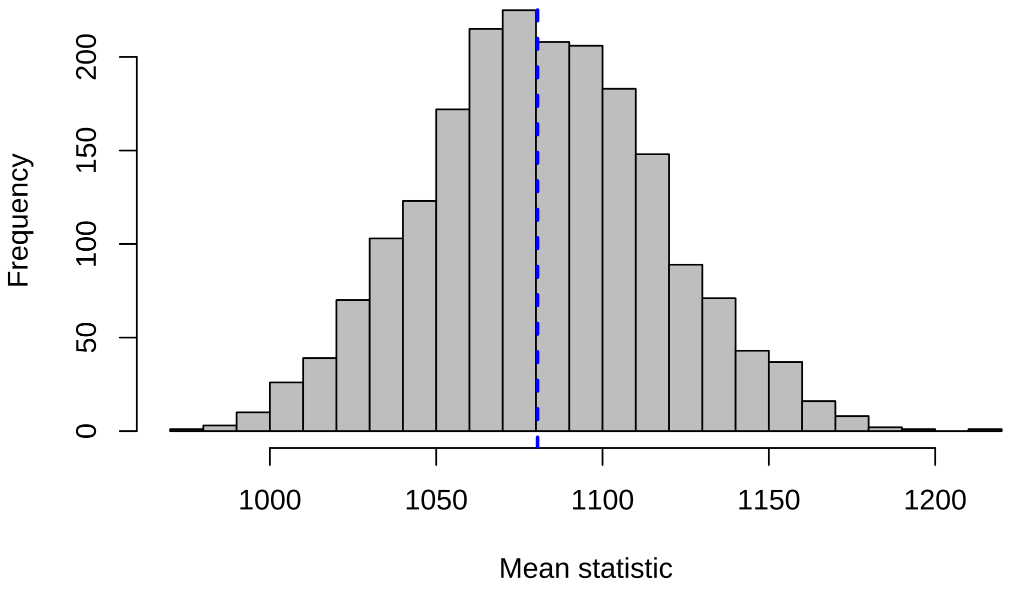

Figure 5: A histogram of the bootstrap distribution, with 2000 resamples, that approximates the sampling distribution of the mean statistic. The dotted vertical line indicates the original mean value of the price.

The distribution of the mean price is estimated using bootstrap can be viewed in Figure 5. From the histogram we can notice that the distribution looks normally distributed.

Further, we will determine the bootstrap bias–correction and the variance of the mean price. The bias-corrected estimate is computed to 1079.487, by taking two times the original mean price minus the mean of the bootstrap sample. The bootstrap variance of the mean price equals 1259.701.

Thereafter we wish to compute a 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first–order normal approximation, which are presented in Table 1.

Type	95% CI	Length
percentile	$(1013.583, 1154.444)$	140.8611
BCa	$(1015.234, 1156.600)$	141.3656
normal approx.	$(1009.923, 1149.050)$	139.1272

Table 1: *The 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first–order normal approximation.*

3.

Now let us consider an alternative way, the jackknife, of estimating the variance of the mean price. The jackknife estimation uses the leave-one-out method, which means that one observation of `Price` is omitted at each iteration. Since our data set consists of 110 observations, we will compute mean price of each sub-sample of size 109.

When estimating the variance of the mean price using the jackknife, we obtain a value of 12.23. This estimation is much lower compared to the estimated variance of the mean price, 1259.701, using bootstrap. Thus, it is clear that there is a big difference between the obtained estimations. One explanation is that only one observation differs between each sub-sample, and therefore the mean of each sub-sample will have a small variation among all the mean prices.

4.

Lastly, we are going to compare the confidence intervals obtained with respect to their length and the location of the estimated mean in these intervals. In Table 1, we can see that the intervals differ slightly depending on which type is being used, but the length of the interval is more similar. To illustrate the difference we present a histogram of the bootstrap distribution with its corresponding confidence interval.

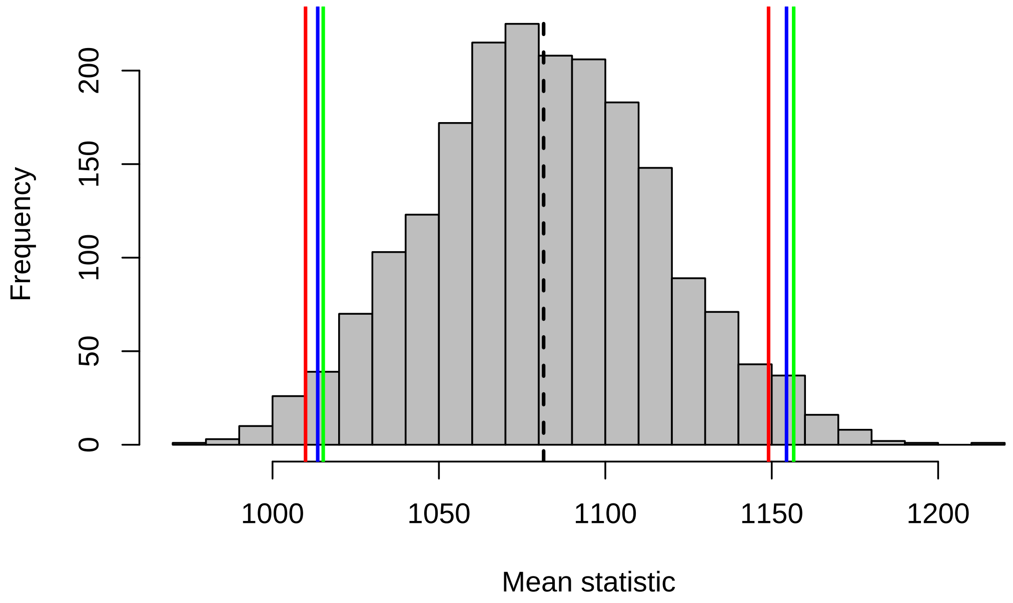

Figure 6: A histogram of the bootstrap distribution, with 2000 resamples, that approximates the sampling distribution of the mean statistic, with its corresponding 95 percent confidence interval. The dotted vertical line indicates the estimated mean price. The blue, green and red vertical lines corresponds to the confidence interval using bootstrap percentile, bootstrap BCa and first-order normal approximation, respectively.

The estimated mean is located around the center in these intervals. When we investigate the location even closer, the estimated mean is closest to the median of the interval using the first-order normal approximation, then comes the percentile, followed up by BCa.

Appendix

```r
knitr::opts_chunk$set(echo = FALSE)
# R version
RNGversion('3.5.1')
# Packages
library("boot")
# 1.1

# Import the data set
data <- read.csv2("lottery.csv")

Y <- data$Draft_No
X <- data$Day_of_year
```

```r
plot(X, Y, xlab = "Day of the year", ylab = "Draft number")

# 1.2
Y_loess <- loess(Draft_No ~ Day_of_year,data = data)
smoothed <- predict(Y_loess)

plot(X, Y, xlab = "Day of the year", ylab = "Smooth Draft number")
lines(smoothed, x=X, col = "red")

# 1.3
Xb <- which.max(smoothed)
Xa <- which.min(smoothed)

T_test <- (smoothed[Xb] - smoothed[Xa])/(Xb - Xa)
# computing bootstrap samples
stat1 <- function(data, ind){
  data1 <- data[ind,]# extract bootstrap sample
  Y_loess <- loess(Draft_No ~ Day_of_year, data = data1)
  smoothed <- predict(Y_loess)
  Xb <- which.max(smoothed)
  Xa <- which.min(smoothed)
  T_test <- (smoothed[Xb] - smoothed[Xa])/(data1$Day_of_year[Xb] - data1$Day_of_year[Xa])
  return(T_test)
}
# Seed
set.seed(1234567890)
res <- boot(data, stat1, R=2000) #make bootstrap
plot(res)
p_value <- sum(res$t >= 0)/(res$R)

#1.4
t_test <- function(data) {
  data2<-data
  loess <- loess(Draft_No ~ Day_of_year,data = data2)
  Y_hat <- predict(loess)
  #Y_hat <- sort(cbind(Y_hat, data2$id_col)[,2])
  Xb <- which.max(Y_hat)
  Xa <- which.min(Y_hat)

  t_stat <- (Y_hat[Xb] - Y_hat[Xa]) / (data2$Day_of_year[Xb] - data2$Day_of_year[Xa])
  #plot(data2$Day_of_year, data2$Draft_No, xlab = "Day of the year", ylab = "Smooth Draft number")
  #points(Y_hat, col = "red", type = "l")
  #lines(Y_hat, x=data2$Day_of_year, col = "red")
  return(t_stat)
}

permutation <- function(data, B) {
  # Seed
  set.seed(1234567890)

  stat <- numeric(B)
  n <- length(data[,1])
  for (b in 1:B) {
    id <- sample(n)
    Gb <- data[id,]
    Gb$Day_of_year <- seq(1,366) # modified by David
    stat[b] <- t_test(Gb)
  }
  stat0 <- t_test(data)
  return(sum(stat <= stat0)/B)
}



p_value_permutation <- permutation(data, 2000) #0.0775
#1.5

# a)

create_dataset <- function(X, alpha) {
  # Seed
  set.seed(1234567890)
  Y_set <- alpha*X+rnorm(366,183,10)
  Y_set[which(Y_set>366)] <- 366
  Y_set[which(Y_set<0)] <- 0
  dataset <- data.frame(Day_of_year=X, Draft_No=Y_set)
}


# b)
dataset <- create_dataset(X = X, alpha = 0.1)

p_value_powertest <- permutation(data = dataset, B = 200) #0.475
# c)

alphas <- seq(from=0.2, to=10, by=0.1)
p_values <- numeric(length(alphas))
for (e in alphas) {
  dataset <- create_dataset(X = X, alpha = e)
  p_values[which(alphas==e)] <- permutation(data=dataset, B=200)
}
p_values
# 2.1

# Import the data set
data <- read.csv2("prices1.csv")
# Compute the mean price
price_mean <- mean(data$Price)

# Histogram of Price
hist(data$Price, breaks = 20, col = "grey", xlab = "Price", main = "")

# 2.2

# Seed
set.seed(123456789)
```

```r
# Function that returns the mean statistic, that will be bootstrapped.
# 1st argument: data set to bootstrap
# 2nd argument: index vector of the data set
mean_statistic <- function(d, i){
  data <- d[i,] # use all observations
  return(mean(data$Price))
}

# Bootstrap
price_boot <- boot(data, mean_statistic, R = 2000)

# Mean of bootstrap samples

boot_mean <- mean(price_boot$t)
# Histogram of the bootstrap distribution
hist(price_boot$t[,1], main = "", xlab = "Mean statistic", col = 'grey', breaks = 20)
abline(v = price_mean, col = "blue", lwd = 2, lty = 2) # Original mean price

# Compute bootstrap estimated bias
bias <- boot_mean - price_mean

# Compute bias-correction, alt 1
price_mean - bias

# Compute bias-correction, alt 2
2*price_mean - boot_mean

#estimated variance
var(price_boot$t)
# 95% confidence interval for the mean price,
#   using bootstrap percentile, bootstrap BCa, and first-order normal approximation
ci_95 <- boot.ci(boot.out = price_boot, type = c("perc", "bca", "norm"))

# CI and their length
ci_perc <- ci_95$perc[ , c(4, 5)]
length_perc <- diff(ci_perc)

ci_bca <- ci_95$bca[ , c(4, 5)]
length_bca <- diff(ci_bca)


ci_norm <- ci_95$normal[ , c(2, 3)]
length_norm <- diff(ci_norm)

# 2.3
# Jackknife the mean
jack <- c()
for (i in seq_along(data$Price)){
  jack[i] <- mean(data$Price[-i])
}

# Variance of mean price
var(jack)
```

```r
# Median of CI
sum(ci_perc)/2
sum(ci_bca)/2
sum(ci_norm)/2
boot_mean
hist(price_boot$t[,1], main = "", xlab = "Mean statistic", col = 'grey', breaks = 20)
abline(v = boot_mean, lwd = 2, lty = 2) # estimated mean price
abline(v = ci_perc, col = 'blue',lwd = 2) # percentile CI
abline(v = ci_bca, col = 'green',lwd = 2) # BCa CI
abline(v = ci_norm, col = 'red',lwd = 2) # normal approx CI
```