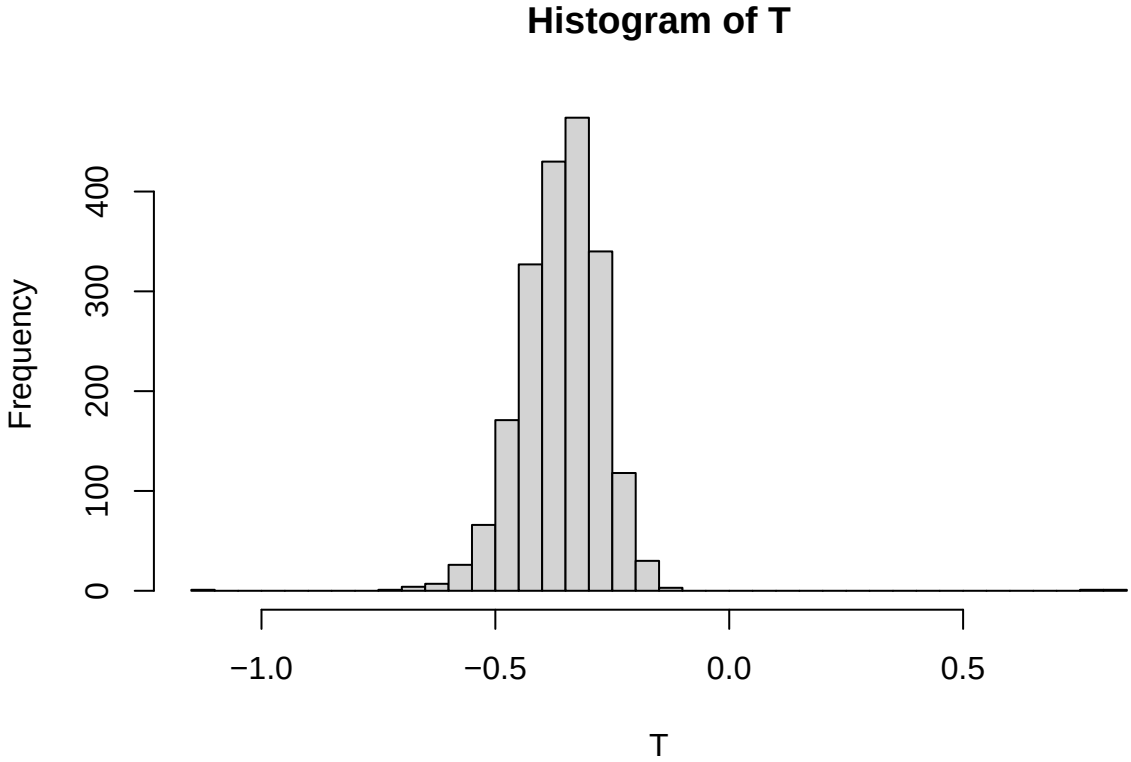
Computational Statistics (732A90) Lab05

Christoforos Spyretos, Marc Braun, Marketos Damigos, Patrick Siegfried Hiemsch & Prakhar

2021-12-07

Question 1

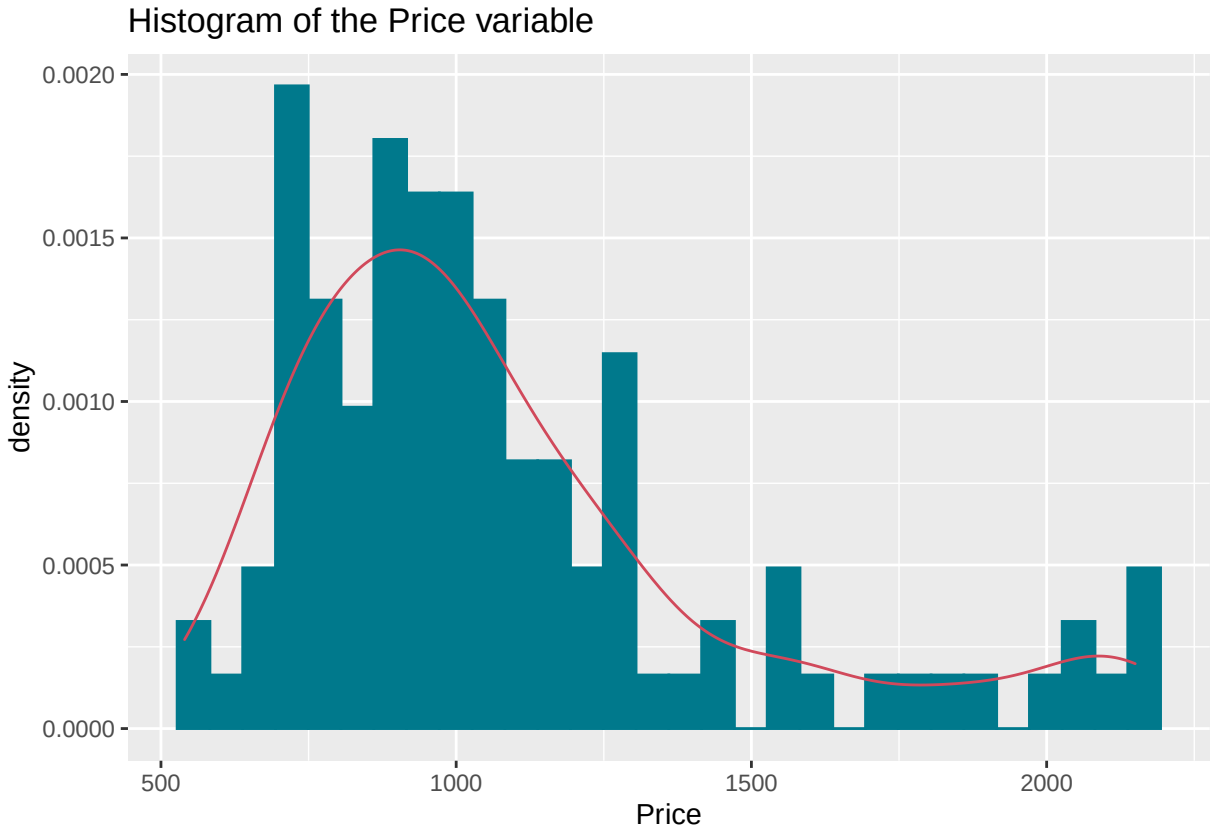
In the first step, 2000 randomly bootstrapped datasets are sampled from the original data. After that, a prediction model is trained using the `loess` function and the test statistic calculated for each bootstrapped dataset, resulting in the subsequent distribution of the statistic.



From the distribution, one can conclude that the value of the statistic seems to be skewed to the left of zero, leading to the conclusion that the "lottery" is not truly random.

Question 2

Task 1



The distribution of the price looks like gamma distribution.

The mean value of the price is 1080.473.

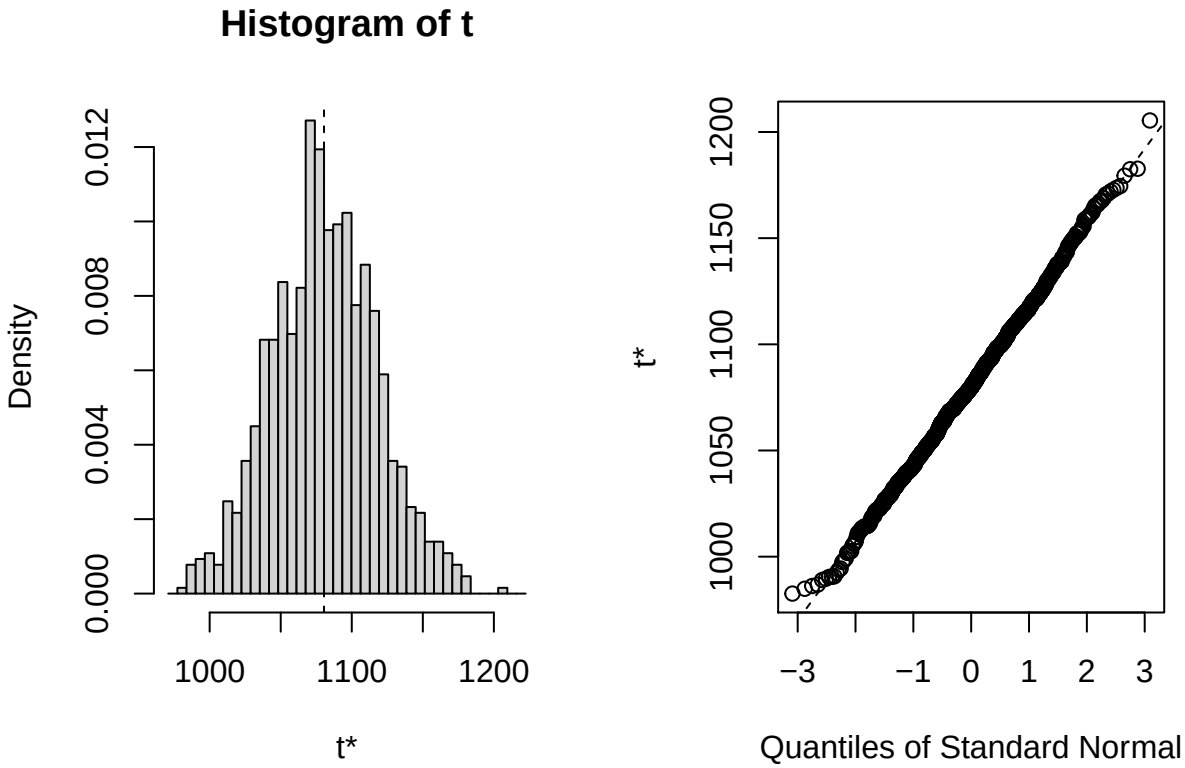
Task 2

The non-parametric bootstrap estimator of bias is given by:

$$\hat{T} = 2T(D) - \frac{1}{B}\sum_{i=1}^{B} T(D_i^*)$$

The variance of estimator is given by:

$$\widehat{Var\,[T(\cdot)]} = \frac{1}{B-1}\sum_{i=1}^{B}(T(D_i^*) - \overline{T(D^*)})^2$$



The bootstrap bias-correction is 1082.036 and the variance of the mean price is 1382.811.

Task 3

	Normal	Basic	Percentile	BCa
Low Bound	1007	1002	1011	1014
Upper Bound	1152	1150	1159	1161

The estimated mean is located in all confidence intervals.

Task 4

The variance of the mean price using the jackknife is given by $\widehat{Var[T(.)]} = \frac{1}{n(n-1)} \sum_{i=1}^{n} ((\mathrm{T_i^*}) - \mathrm{J(T)})^2$, where T_i^* is give by $T_i^* = nT(D) - (n-1)T(D_i^*)$ and $J(T)$ is given by $J(T) = \frac{1}{n} \sum_{i=1}^{n} T_i^*$.

The estimated variance of the mean price using the jackknife is 1320.911. Thus it could be assumed that bootstrap performs better compared to jackknife, because it has a higher value than the variance of the mean price using bootstrap (1382.811).

Appendix

```
library(formatR)
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 80), tidy = TRUE)
lottery <- read.csv("lottery.csv", sep = ";")
Y <- lottery$Draft_No
X <- lottery$Day_of_year

set.seed(12345)
bootstrap <- matrix(nrow = 2000, ncol = length(X))
bootstrap_Y <- matrix(nrow = 2000, ncol = length(X))
for (i in 1:2000) {
    bootstrap[i, ] <- sample(X, replace = TRUE)
    bootstrap_Y[i, ] <- Y[bootstrap[i, ]]
}

T <- c()
for (i in 1:nrow(bootstrap)) {
    temp_data <- data.frame(Draft_No = bootstrap_Y[i, ], Day_of_year = bootstrap[i,
        ])
    model <- loess(Draft_No ~ Day_of_year, data = temp_data)
    Xb <- bootstrap[i, max(which(predict(model, bootstrap[i, ]) == max(predict(model,
        bootstrap[i, ]))))]
    Xa <- bootstrap[i, min(which(predict(model, bootstrap[i, ]) == min(predict(model,
        bootstrap[i, ]))))]
    T <- append(T, (predict(model, Xb) - predict(model, Xa))/(Xb - Xa))
}

hist(T, breaks = 30)
price <- read.csv("prices1.csv", sep = ";")
library(ggplot2)
my_histogram <- ggplot(price, aes(x = Price)) + geom_histogram(bins = 30, color = "#00798c",
    fill = "#00798c", aes(y = ..density..)) + geom_density(colour = "#d1495b") +
    labs(title = "Histogram of the Price variable")
ylab("Density")

my_histogram
mean_value <- mean(price$Price)
library("boot")

# statistic
my_stat_fun <- function(data, indeces) {
    return(mean(data[indeces]))
}

# typical bootstrap replicates 100-2000
B <- 1000

bootstrap <- boot(data = price$Price, statistic = my_stat_fun, R = B)
mean_estimator <- 2 * bootstrap$t0 - mean(bootstrap$t)
variance_estimator <- (1/(B - 1)) * sum((bootstrap$t - mean(bootstrap$t))^2)
ci <- boot.ci(boot.out = bootstrap)  # conf=0.95 default
plot(bootstrap)
intervals <- data.frame(Normal = round(ci$normal[c(2, 3)]), Basic = round(ci$basic[c(4,
    5)]), Percentile = round(ci$percent[c(4, 5)]), BCa = round(ci$bca[c(4, 5)]))

rownames(intervals) <- c("Low Bound", "Upper Bound")
```

```r
knitr::kable(intervals)
n <- nrow(price)

ti <- list()

for (i in 1:n) {
    ti[i] <- n * mean(price$Price) - (n - 1) * mean(price[-i, 1])
}

jt <- mean(unlist(ti))

jackknife <- (1/(n * (n - 1))) * sum((unlist(ti) - jt)^2)
```