cstats_lab05

Thijs Quast (thiqu264), Anubhav Dikshit (Anudi287)

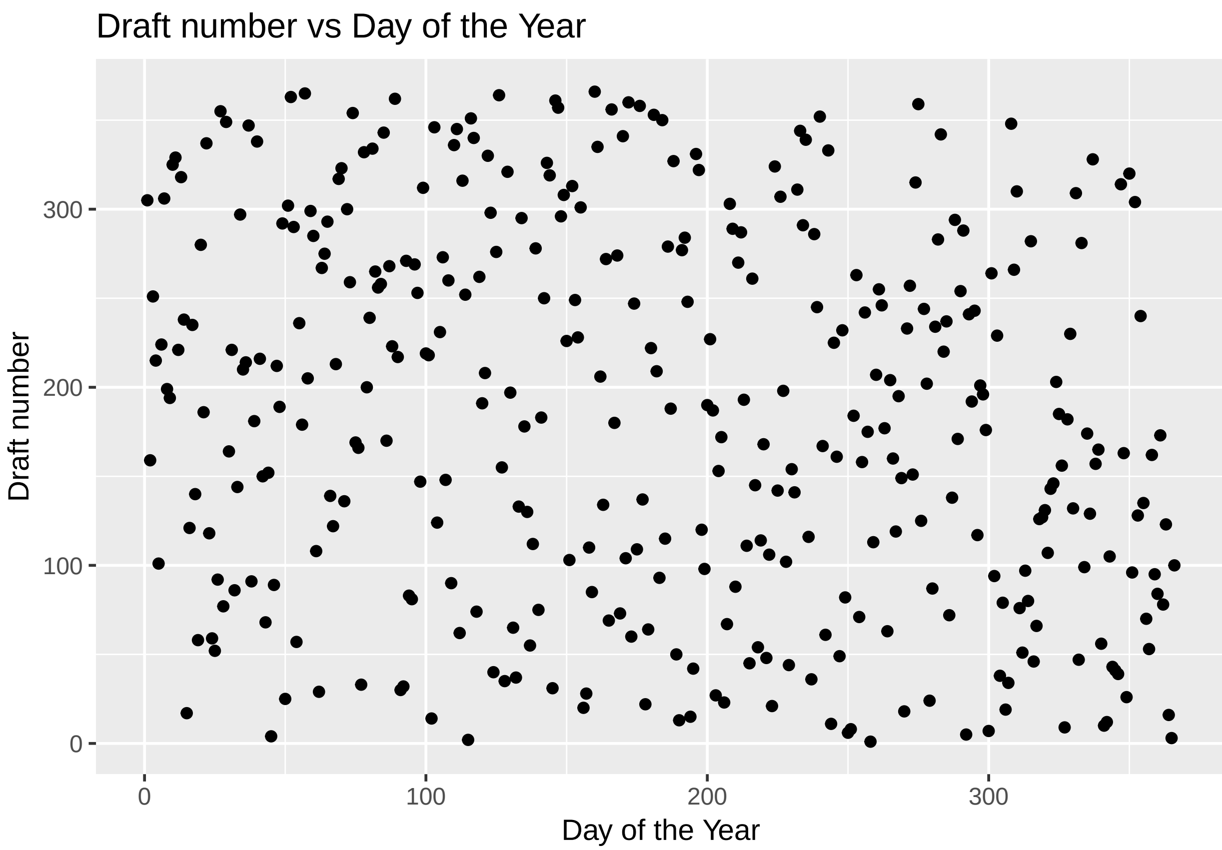
19-2-2019

Question 1 - Hypothesis testing

```
library(readxl)
lottery <- read_xls("lottery.xls")
Y <- lottery$Draft_No
X <- lottery$Day_of_year
```

1.1

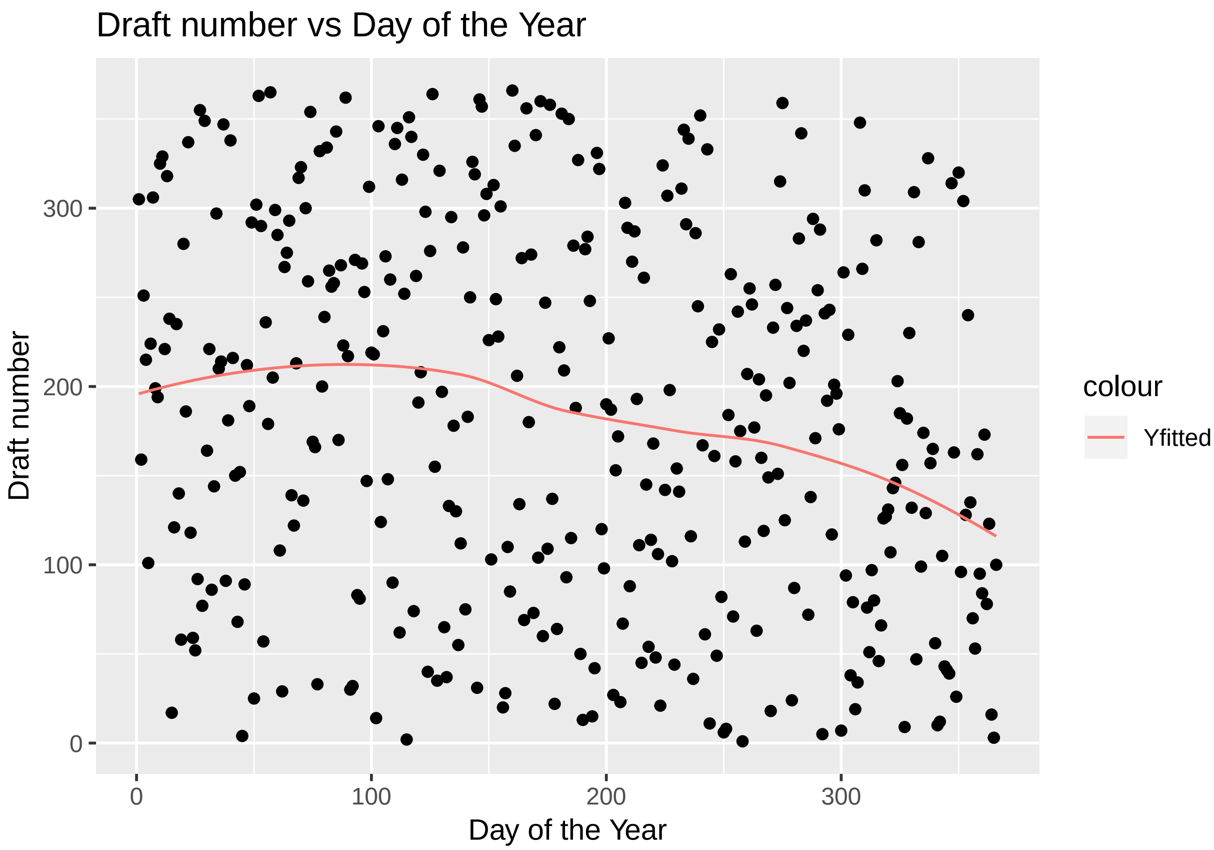
```
library(ggplot2)
plot <- ggplot(lottery, aes(x = X, y = Y)) + geom_point()
plot <- plot + xlab("Day of the Year") + ylab("Draft number") + ggtitle("Draft number vs Day of the Yea
plot
```



The lottery looks random, no pattern between Draft number and Day of teh Year seems to be present.

1.2

```
loess_model <- loess(Y ~ X, data = lottery)
Yfitted <- loess_model$fitted
plot <- plot + geom_line(aes(x = X, y =Yfitted, col = "Yfitted"))
plot
```



Still, largely the lottery looks random. However, if it would be perfectly random there would be an exact horizontal line, now there seems to be slight downward trend in the data. This implies that actually people who were born at the end of the year have a higher chance of getting a lower draft number and therefore being drafted earlier.

1.3

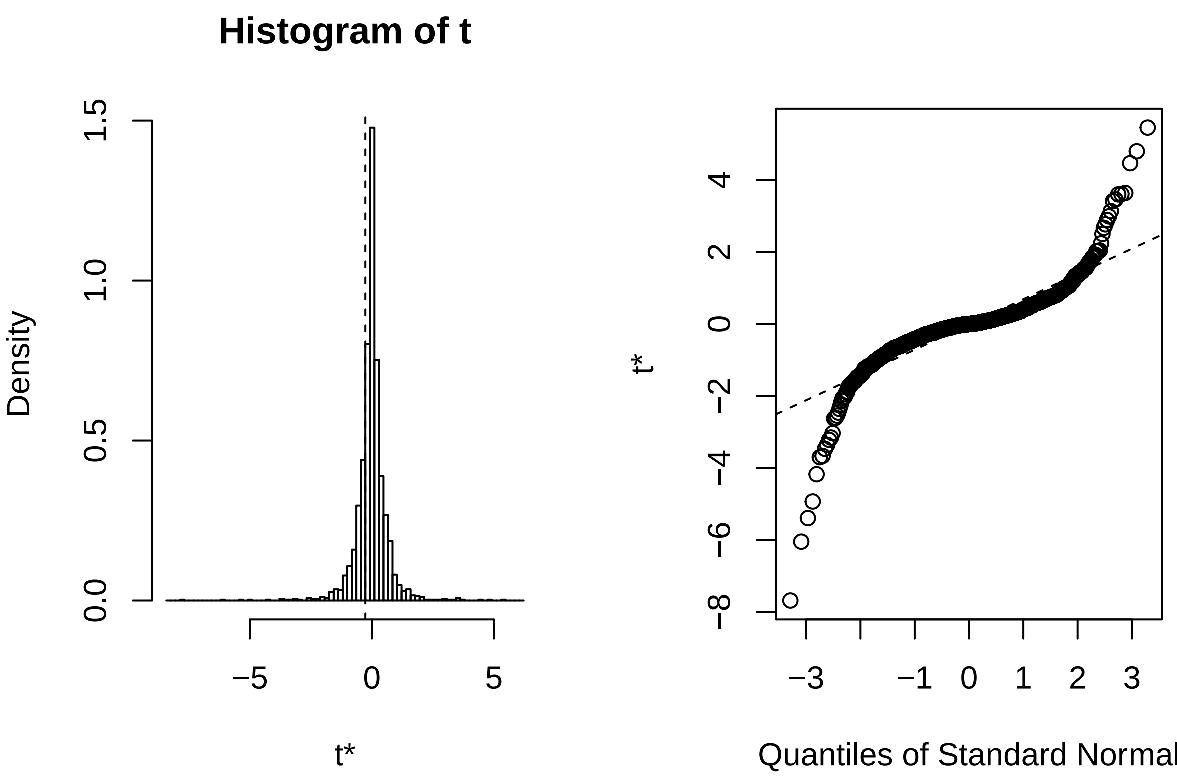
```
library(boot)

stat1<-function(data,vn){
    # Data frame for the bootstrap
    data<-as.data.frame(data[vn,])
    # Model and prediction
    loess_model2 <- loess(Draft_No ~ Day_of_year, data)
    prediction <- predict(loess_model2, data)

    # Statistics for T
    X_a <- data$Day_of_year[which.max(data$Draft_No)]
    X_b <- data$Day_of_year[which.min(data$Draft_No)]
```

```r
    Y_a <- prediction[X_a]
    Y_b <- prediction[X_b]

    # Output
    t <- ((Y_b - Y_a) / (X_b - X_a))
    return(t)

}

bootstrap <- boot(lottery,stat1,R=2000)
```

```r
print(boot.ci(bootstrap))
```

```
## Warning in boot.ci(bootstrap): bootstrap variances needed for studentized
## intervals

## Warning in norm.inter(t, adj.alpha): extreme order statistics used as
## endpoints

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootstrap)
##
## Intervals :
## Level      Normal              Basic
## 95%   (-1.8963,  0.8557 )   (-1.8722,  0.8234 )
##
## Level     Percentile            BCa
## 95%   (-1.3577,  1.3379 )   (-7.6864,  0.1453 )
## Calculations and Intervals on Original Scale
## Warning : BCa Intervals used Extreme Quantiles
## Some BCa intervals may be unstable
```

```r
plot(bootstrap)
```




```
t_final <- mean(bootstrap$t)
t_final
```

```
## [1] -0.01402395
```

```
p_value <- pt(t_final, df = (nrow(lottery)-1))
p_value
```

```
## [1] 0.4944093
```

We get a p-value of 0.40, meaning we cannot reject the null hypothesis. Meaning we cannot say the lottery is not random.

1.4

```
permutation <- function(data, B){

}
```

1.5

Assignment 2 - Bootstrap, jackknife and confidence intervals

2.1

```
prices <- read_excel("prices1.xls")
```

```r
hist(prices$Price, breaks = 50)
```

I would say this data shows the shape of a beta distribution.

```r
mean(prices$Price)
```

```
## [1] 1080.473
```

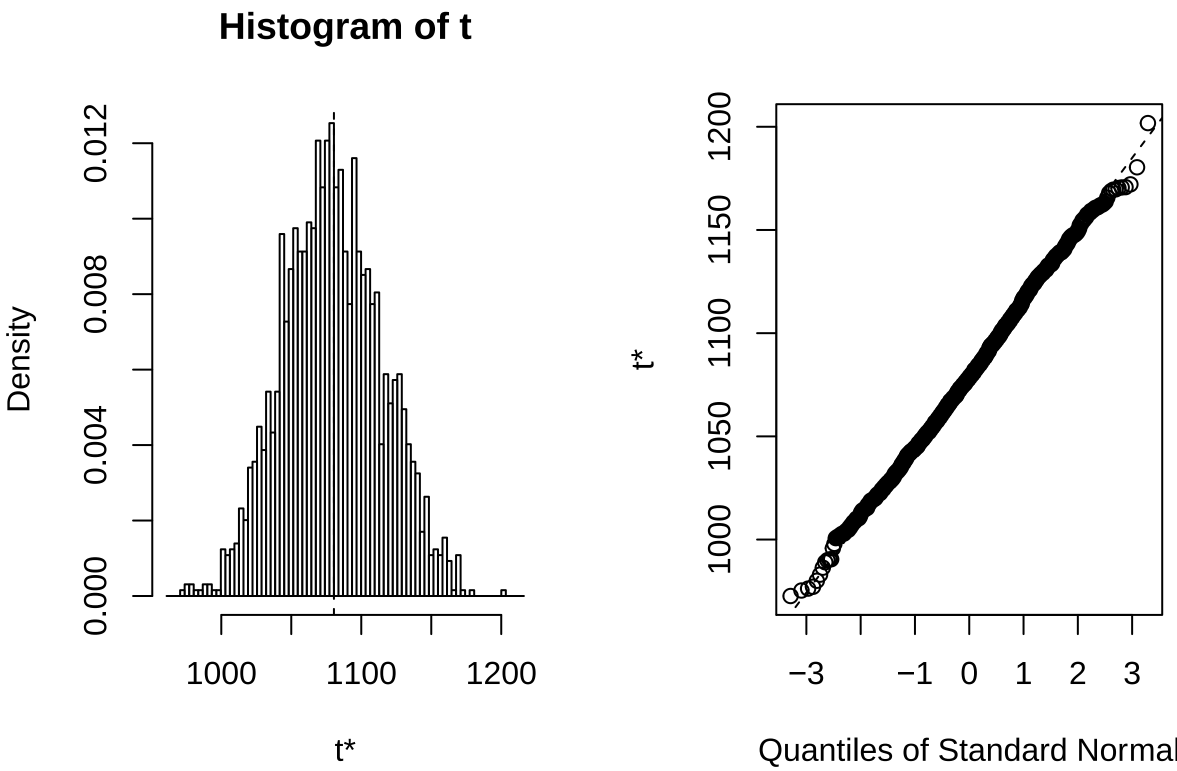
2.2

```r
library(boot)

stat2 <- function(data, vn){
  data <- as.data.frame(data[vn,])
  output <- mean(data$Price)
}

res <- boot(prices, stat2, R = 2000)
res
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = prices, statistic = stat2, R = 2000)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
```

```
## t1* 1080.473 -0.8658045    35.10489
```

```r
plot(res)
```




```r
var(res$t)
```

```
##          [,1]
## [1,] 1232.353
```

```r
boot.ci(res, type = "perc")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   (1014, 1148 )
## Calculations and Intervals on Original Scale
```

```r
boot.ci(res, type = "bca")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res, type = "bca")
##
## Intervals :
## Level       BCa
```

```
## 95%   (1020, 1158 )
## Calculations and Intervals on Original Scale
```

```r
boot.ci(res, type = "norm")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = res, type = "norm")
##
## Intervals :
## Level      Normal
## 95%   (1013, 1150 )
## Calculations and Intervals on Original Scale
```

2.3

```r
library(bootstrap)
res2 <- jackknife(prices$Price, mean)
summary(res2)
```

```
##             Length Class  Mode
## jack.se       1    -none- numeric
## jack.bias     1    -none- numeric
## jack.values 110    -none- numeric
## call          3    -none- call
```

```r
res2
```

```
## $jack.se
## [1] 36.34434
##
## $jack.bias
## [1] 0
##
## $jack.values
##   [1] 1071.578 1071.303 1070.661 1070.661 1072.046 1072.954 1073.872
##   [8] 1076.073 1077.083 1077.092 1077.771 1078.734 1078.917 1079.055
##  [15] 1079.651 1079.560 1079.789 1080.202 1079.936 1081.257 1081.257
##  [22] 1081.440 1081.440 1082.128 1081.578 1082.495 1074.835 1076.138
##  [29] 1078.917 1078.459 1081.028 1081.028 1081.927 1081.899 1082.138
##  [36] 1082.587 1082.349 1082.220 1082.404 1083.963 1083.780 1083.514
##  [43] 1083.679 1083.734 1084.239 1070.661 1075.716 1078.000 1078.468
##  [50] 1078.917 1079.018 1079.376 1080.064 1080.294 1080.477 1080.752
##  [57] 1080.761 1081.624 1081.817 1082.358 1082.229 1082.541 1082.725
##  [64] 1082.954 1083.000 1083.055 1083.505 1083.422 1083.505 1083.688
##  [71] 1083.697 1083.872 1084.239 1084.706 1078.505 1081.440 1081.771
##  [78] 1082.862 1083.229 1083.321 1083.963 1085.431 1080.569 1071.119
##  [85] 1083.734 1084.330 1085.064 1073.468 1075.890 1083.972 1079.743
##  [92] 1080.211 1080.028 1080.752 1080.798 1080.752 1081.028 1081.211
##  [99] 1080.936 1081.440 1081.670 1081.761 1081.945 1081.716 1082.367
## [106] 1082.385 1082.404 1082.413 1083.358 1083.606
##
## $call
```

```
## jackknife(x = prices$Price, theta = mean)
```

```
confidence_interval_jackknife <- c(mean(res2$jack.values - 1.96*res2$jack.se),
                                   mean(res2$jack.values + 1.96*res2$jack.se))
```