Homework 02 - Statistical Methods for Data Science

Letícia Negrão Pinto

DSSC - 2019

Core Statistics (CS) - Chapter 3

Exercise 3.3

- Rewrite the following, replacing the loop with efficient code:

Answer: At first we have the code:

```
set.seed(2019)

# We increased n to highlight the time difference
n <- 100000000
# rnorm(n) will generate n numbers following a normal distribution, with mean 0 and sd 1
z <- rnorm(n)
zneg <- 0
j <- 1

# We can measure the time using the function 'Sys.time()'
start_time <- Sys.time()

for (i in 1:n)
{
  if (z[i]<0)
    {
      zneg[j] <- z[i]
      j <- j + 1
     }
}

end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 39.00332 secs
```

Replacing the loop with efficient code:

```
start_time <- Sys.time()

    zneg2 <- c(z[which(z<0)])

end_time <- Sys.time()
end_time - start_time
```

```
## Time difference of 2.725374 secs
```

Now it is necessary to validate the result:

```
all(zneg2 %in% zneg)
```

```
## [1] TRUE
```

```r
all(zneg %in% zneg2)
```

```
## [1] TRUE
```

For the validation, we used “%in%” to return a logical vector indicating if there is a match or not for its left operand.

In conclusion, our efficient method produces the same result as the previous one without using a loop for and in less time.

DAAG - Chapter 3

Exercise 11

- The following data represent the total number of aberrant crypt foci (abnormal growths in the colon) observed in seven rats that had been administered a single dose of the carcinogen azoxymethane and sacrificed after six weeks (thanks to RanjanaBird, Faculty of Human Ecology,University of Manitoba for the use of these data):

87 53 72 90 78 85 83

Enter these data and compute their sample mean and variance.

Answer:

In order to compute the sample mean and variance we can simply use the R functions mean() and var().

```r
# The total number abnormal growths in the colon observed in seven rats:
experimental_data = c(87, 53, 72, 90, 78, 85, 83)

mean(experimental_data)
```

```
## [1] 78.28571
```

```r
var(experimental_data)
```

```
## [1] 159.9048
```

- Is the Poisson model appropriate for these data? To investigate how the sample variance and sample mean differ under the Poisson assumption, repeat the following simulation experiment several times:

```r
set.seed(2019)

x <- rpois(7, 78.3)
mean(x)
```

```
## [1] 81
```

```r
var(x)
```

```
## [1] 85.66667
```

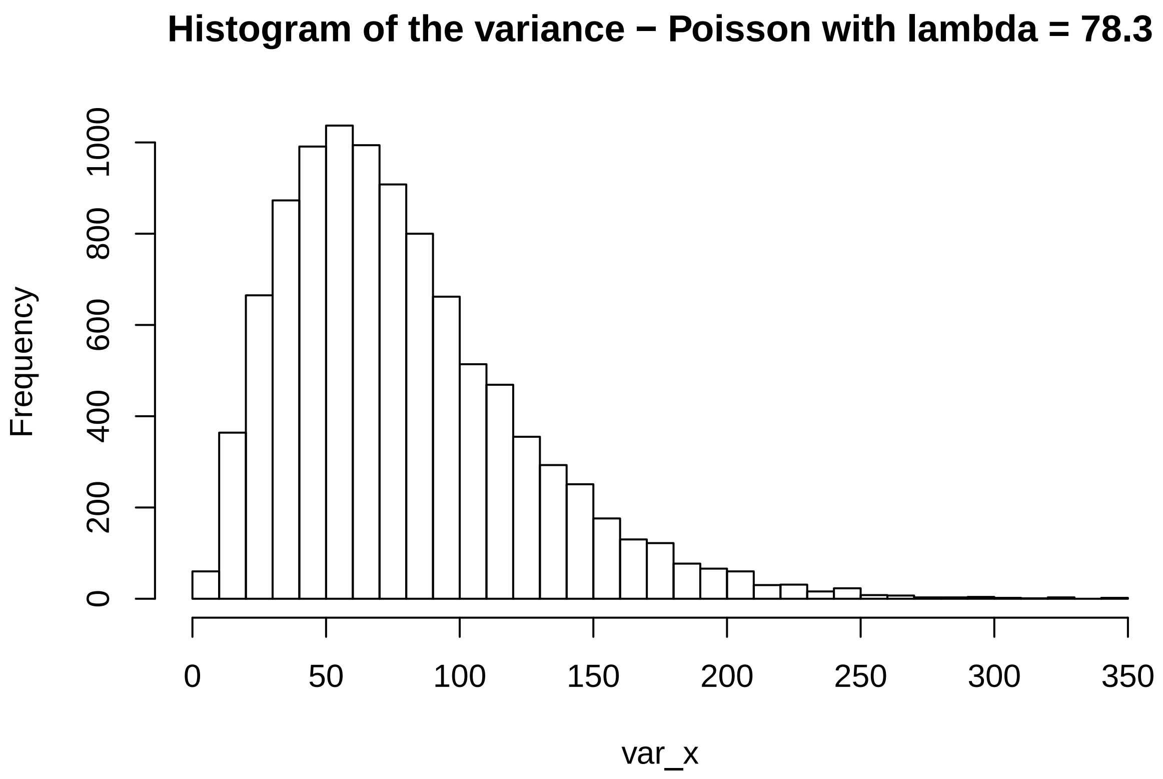
Answer:

Repeating the previous simulation experiment several times, we can find the following results:

```r
mean_x = c()
var_x = c()

for (i in 1:10000)
{
    x <- rpois(7, 78.3)
    mean_x[i] = mean(x)
    var_x[i] = var(x)
}

mean(mean_x)
```

```
## [1] 78.24583
```

```r
hist(var_x, breaks=30, main = 'Histogram of the variance - Poisson with lambda = 78.3')
```



Under the Poisson assumption, considering the parameter λ the mean of the experimental sample (mean(experimental_data)), the values that can be seen for the most frequent variance of the simulation (var_x) call our attention because they are extremely different from what we have obtained for our experiment (var(experimental_data)).

This difference between the variance of the distribution and the variance of experimental data is an indicator that the Poisson model might **not** be the most appropriate to describe the data.

DAAG - Chapter 4

Exercise 7

- Create a function that does the calculations in the first two lines of the previous exercise.Put the calculation in a loop that repeats 25 times. Calculate the mean and variance for each vector y that is returned. Store the 25 means in the vector av, and store the 25 variances in the vector v. Calculate the variance of av.

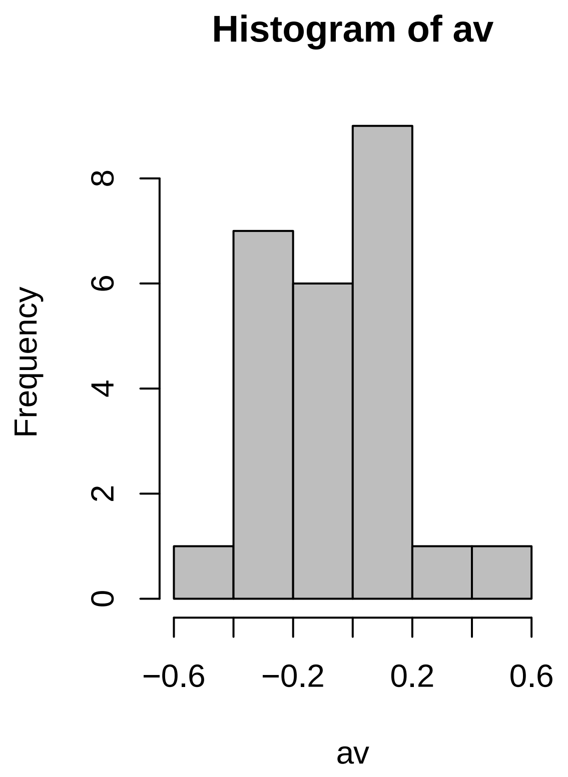
Answer:

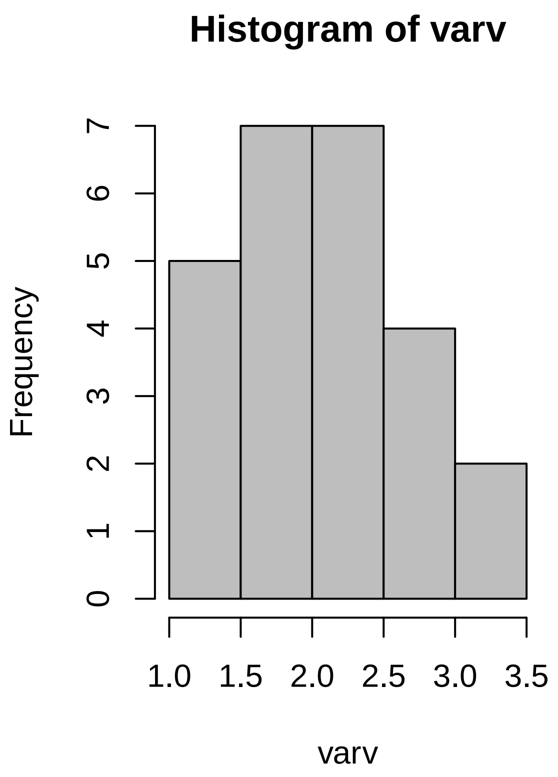
In the previous exercise we have:

```r
# Generation of 51 numbers following a normal distribution with mean 0 and sd 1
y1 <- rnorm(51)
# Sum of the vectors y1 without the first element with y1 without the last element
y <- y1[-1] + y1[-51]
```

We now construct a function:

```r
set.seed(2019)

 function_7 <- function(n = 51){
 y1 <- rnorm(n)
 y <- y1[-1] + y1[-n]
 y
 }

av <- c()
varv <- c()

for (i in 1:25) {
z <- function_7()
# Mean and variance for each vector y that is returned:
av[i] <- mean(z)
varv[i] <- var(z)
}

#histogram of av and varv:
par(mfrow = c(1,2))
hist(av, col = 400)
hist(varv, col = 400)
```





Finally, the variance of "av", as required:

```
# The variance of av:
 var(av)
```

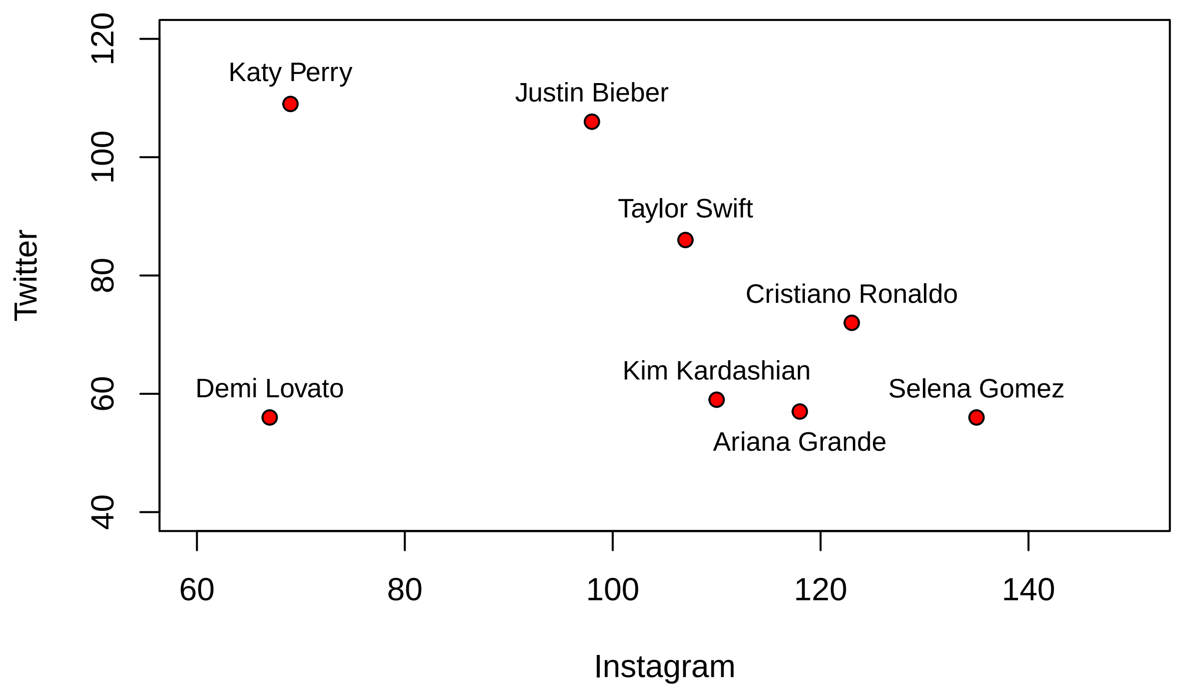
```
## [1] 0.05059898
```

Lab Exercises

Exercise 3

- Consider now some of the most followed Instagram accounts in 2018: for each of the owners, we report also the number of Twitter followers (in millions). Are the Instagram and Twitter account somehow associated? Perform a correlation test, compute the p-value and give an answer. Here is the data frame.

```
     Owners <- c( "Katy Perry", "Justin Bieber", "Taylor Swift", "Cristiano Ronaldo",
                  "Kim Kardashian", "Ariana Grande", "Selena Gomez", "Demi Lovato")
     Instagram <- c( 69, 98,107, 123, 110, 118, 135, 67)
     Twitter <- c( 109, 106, 86, 72, 59, 57, 56, 56)
     plot( Instagram, Twitter, pch=21, bg=2, xlim=c(60, 150), ylim=c(40, 120) )
     text( Instagram[-6], Twitter[-6]+5, Owners[-6], cex=0.8 )
     text( Instagram[6], Twitter[6]-5, Owners[6], cex=0.8 )
```



Answer:

In order to investigate if Instagram and Twitter accounts are somehow associated, we can begin stating what are our hypothesis:

$$H_0 : \rho = 0$$

$$H_1 : \rho \neq 0$$

where ρ stands for the correlation between Instagram and Twitter.

In order to investigate the relation between the variables, we can begin by computing the Pearson correlation:

```
# Computing the correlation and p-value:
results <- cor.test(Instagram,Twitter)
results
```

```
##
##  Pearson's product-moment correlation
##
## data:  Instagram and Twitter
## t = -1.1454, df = 6, p-value = 0.2957
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8689006  0.4006898
## sample estimates:
##        cor
## -0.4235845
```

The Pearson correlation coefficient is a measure of the strength and direction of association that exists between two variables. Its value can range from -1 for a perfect negative linear relationship to +1 for a perfect positive linear relationship. A value of 0 (zero) indicates no relationship between two variables.

In order to perform the computation, we used the R function 'cor.test()', a function that tests for association between paired samples using one of Pearson's product moment correlation coefficient. The function returns both the correlation coefficient and the significance level(or p-value) of the correlation.

Considering that the value obtained was approximately -0.4, we may interpret that the relationship between the Instagram and Twitter accounts are inversely proportional (negative correlation). In general, values between 0.3 and 0.7 (or -0.3 and -0.7) indicate a moderate linear relationship between the variables. However, when considering the p-value, we tend to say that there is **not** a relationship between the variables.

```
# Correlation:
results$estimate
```

```
##        cor
## -0.4235845
```

```
# P-value:
results$p.value
```

```
## [1] 0.2956669
```

From another point of view, we may think that there might be a linear relationship between the variables. In order to investigate this assumption we do:

```
# Considering a linear model:
# Twitter = intercept + beta*Instagram
linearMod <- lm(Twitter ~ Instagram)
linearMod
```

```
##
## Call:
## lm(formula = Twitter ~ Instagram)
##
## Coefficients:
## (Intercept)    Instagram
##    115.4246      -0.3898
```

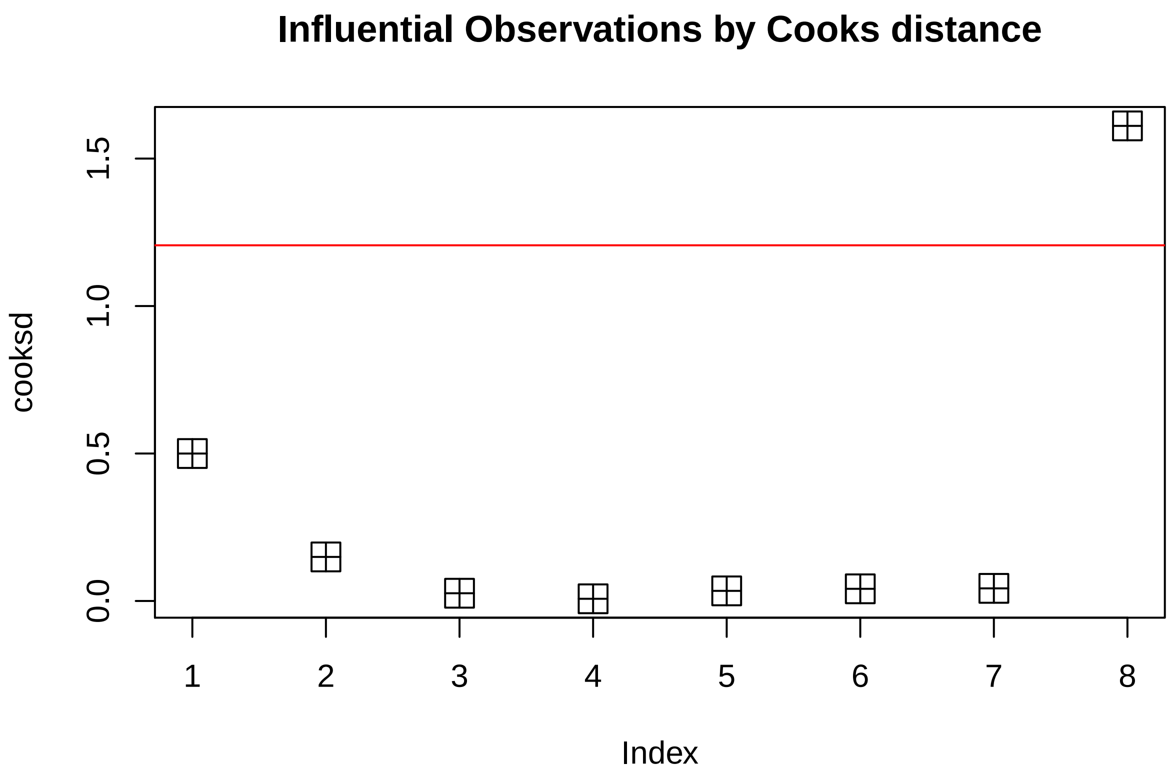
```
# Model summary:
summary(linearMod)
```

```
##
## Call:
## lm(formula = Twitter ~ Instagram)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.305 -12.703  -1.135  14.335  28.780
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 115.4246    36.0358   3.203   0.0185 *
## Instagram    -0.3898     0.3404  -1.145   0.2957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.03 on 6 degrees of freedom
## Multiple R-squared:  0.1794, Adjusted R-squared:  0.04266
## F-statistic: 1.312 on 1 and 6 DF,  p-value: 0.2957
```

Again, considering the values presented by the output of the function "summary()", we tend to say that there is **not** a relationship between the variables.

On the other hand, analysing the graph, one may think that the Demi Lovato's point is an outlier. In order to investigate this issue we might compute the Cook's distance:

```r
# Searching for outliers:
cooksd <- cooks.distance(linearMod)

# We have an indication of outliers - In general use, those observations that have a
# Cook's distance greater than 4 times the mean may be classified as influential.

# plot cook's distance:
plot(cooksd, pch=12, cex=2, main="Influential Observations by Cooks distance")
abline(h = 4*mean(cooksd, na.rm=T), col="red")  # cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4*mean(cooksd,
     na.rm=T),names(cooksd),""), col="red")  # labels
```



Cook's distance is the scaled change in fitted values, which is useful for identifying outliers. This value shows the influence of each observation on the fitted response values.

Identifying as an outlier and ignoring the Demi Lovato's data (the 8th observation), now we can compute again the correlation and p-value:

```r
# Ignoring Demi Lovato's data:
Instagram_without_DL <- Instagram[-length(Instagram)]# Demi Lovato is the last one
Twitter_without_DL <- Twitter[-length(Twitter)]

# Computing the correlation and p-value:
```

```
results <- cor.test(Instagram_without_DL,Twitter_without_DL)
results
```

```
##
##  Pearson's product-moment correlation
##
## data:  Instagram_without_DL and Twitter_without_DL
## t = -3.4438, df = 5, p-value = 0.01836
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9755887 -0.2324939
## sample estimates:
##        cor
## -0.8387096
```

```
# Considering a linear model:
# Twitter = intercept + beta*Instagram
linearMod <- lm(Twitter_without_DL ~ Instagram_without_DL)
linearMod
```

```
##
## Call:
## lm(formula = Twitter_without_DL ~ Instagram_without_DL)
##
## Coefficients:
##          (Intercept)  Instagram_without_DL
##             176.3210               -0.9069
```

```
# Model summary:
summary(linearMod)
```

```
##
## Call:
## lm(formula = Twitter_without_DL ~ Instagram_without_DL)
##
## Residuals:
##       1       2       3       4       5       6       7
##  -4.745  18.556   6.718   7.228 -17.562 -12.306   2.111
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          176.3210    29.0520   6.069  0.00175 **
## Instagram_without_DL  -0.9069     0.2633  -3.444  0.01836 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.63 on 5 degrees of freedom
## Multiple R-squared:  0.7034, Adjusted R-squared:  0.6441
## F-statistic: 11.86 on 1 and 5 DF,  p-value: 0.01836
```

Ignoring the outlier identified as Demi Lovato's data, it is possible to notice that the values for the correlation and p-value change completely. Considering now the strong correlation and low p-value, now we may conclude that the variables are indeed associated.

```
# Correlation without Demi Lovato's data:
results$estimate
```

```
##        cor
## -0.8387096
```

```
# P-value without Demi Lovato's data:
results$p.value
```

```
## [1] 0.0183603
```

In conclusion, we highlight that one must be cautious when considering the presented results. It is clear that more data is needed to assure whether or not the two variables are really associated, since we only have 8 values for each variable at our disposal.