Assignment exercises. Unit 2.

Arnau García and Maria Lee

Problem 1: Regression modeling is one of the most popular statistical methods and, as such, its aspects have been widely studied. In this exercise we investigate the use of bootstrapping in regression and correlation.

(a) Select one dataset that is appropriate to study the correlation between two continuous variables, call them X and Y, and perform a simple bivariate descriptive analysis that includes a scatterplot and the correlation coefficient. Also do some bet about the data distribution.

We will be using a data set of characteristics about penguins. In this data set we have several categorical variables for the penguins: sex, species. And we have several continuous variables related with the physical characteristics of the penguins: the body mass (in grams), the bill length (in mm), the flipper length (in mm), etc.

We take this dataset from a public repository of the GitHub of Victor Peña (professor of Linear and Linear Generalized models of this master).

```r
peng = read.csv("https://vicpena.github.io/glm/datasets/penguins_train.csv")
# we convert the categorical variables in factors,
#as a good practice before analyze the data
peng$year = factor(peng$year)
peng$sex = factor(peng$sex)
```

See several observations of this dataset:

```r
head(peng)
```

```
    species    island bill_length_mm bill_depth_mm flipper_length_mm
1    Gentoo    Biscoe           49.5          16.1               224
2    Gentoo    Biscoe           55.1          16.0               230
3    Adelie     Dream           39.7          17.9               193
4    Adelie    Biscoe           35.0          17.9               190
5    Adelie Torgersen           40.6          19.0               199
```

```
6 Chinstrap     Dream          52.0          18.1                 201
  body_mass_g    sex year
1        5650   male 2009
2        5850   male 2009
3        4250   male 2009
4        3450 female 2008
5        4000   male 2009
6        4050   male 2007
```

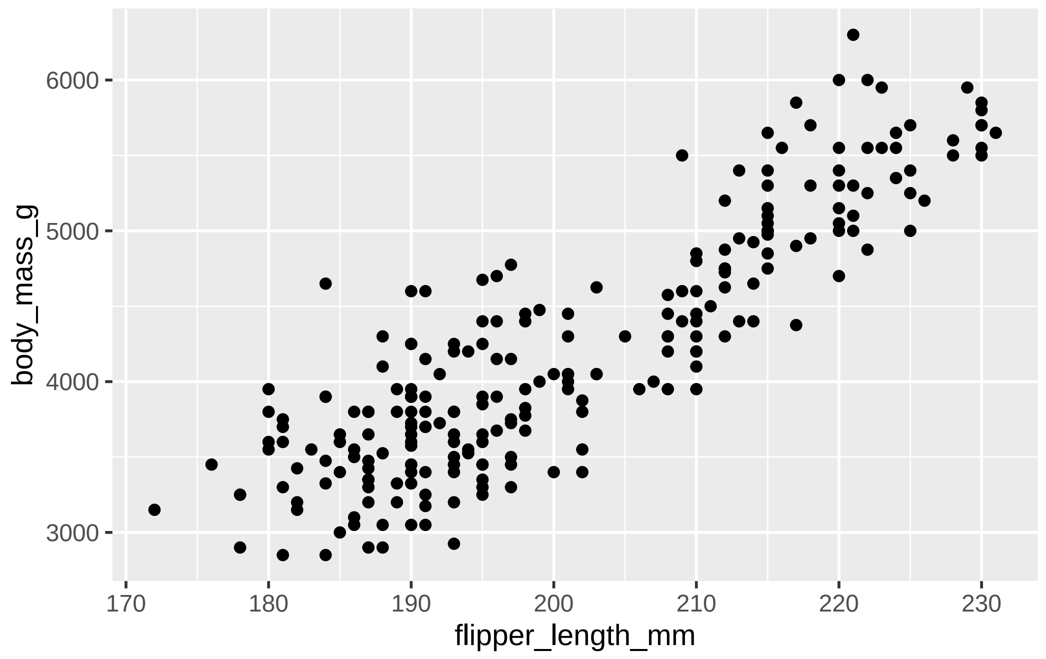
In addition, see the summary:

```
summary(peng)
```

```
   species             island          bill_length_mm  bill_depth_mm
 Length:233         Length:233         Min.   :32.10   Min.   :13.10
 Class :character   Class :character   1st Qu.:39.60   1st Qu.:15.60
 Mode  :character   Mode  :character   Median :45.10   Median :17.50
                                       Mean   :44.01   Mean   :17.21
                                       3rd Qu.:48.60   3rd Qu.:18.80
                                       Max.   :58.00   Max.   :21.50
 flipper_length_mm  body_mass_g       sex         year
 Min.   :172       Min.   :2850   female:113   2007:73
 1st Qu.:190       1st Qu.:3550   male  :120   2008:79
 Median :197       Median :4050                2009:81
 Mean   :201       Mean   :4210
 3rd Qu.:213       3rd Qu.:4800
 Max.   :231       Max.   :6300
```

We select for do the different problems the continuous variables X as *flipper_length_mm* and Y as *body_mass_g*. Thus, now we perform the simple bivariate analysis desired. The reader can observe the summary statistics of these variables in the previous summary. Now, we do an scatterplot:

```
library(ggplot2)
ggplot(peng) +
  aes(flipper_length_mm, body_mass_g)+
  geom_point()
```



And we compute the correlation coefficient:

```
cor_cof <- with(peng, cor(flipper_length_mm, body_mass_g))
cor_cof
```

```
[1] 0.872783
```

With the results obtained in the scatterplot and the correlation coefficient one can do some bet about the data distribution. We observe that we obtain a big correlation coefficient near to one, which indicates that there is a relationship between our variables X, Y. In addition, observing the scatterplot one can also imagine that there is a relationship between the variables. We observe that when the flipper length grows, so does the body mass.

(b) Investigate how to build a confidence interval for the correlation coefficient, ρ, using Fisher's transformation and build a 95% CI for its MOM estimator r.

First, we derive the MOM estimator r, and then using r we build a confidence interval using the Fisher's transformation.

Observe that we can write the correlation coefficient, ρ, as:

$$\rho = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - (E(X))^2}\sqrt{E(Y^2) - (E(Y))^2}}$$

From the last expression is esasy to deduce the MOM estimator:

$$r = \frac{\hat{\mu}^1_{X,Y} - \hat{\mu}^1_X \hat{\mu}^1_Y}{\sqrt{\hat{\mu}^2_X - (\hat{\mu}^1_X)^2}\sqrt{\hat{\mu}^2_Y - (\hat{\mu}^1_Y)^2}}$$

Where $\hat{\mu}^1_{X,Y}$ is the first sample moment of XY, $\hat{\mu}^1_X$ is the first sample moment of X (idem for Y). And $\hat{\mu}^2_X$ is the second sample moment of X (idem for Y).

Now, we build the desired confidence interval. First we should compute the MOM estimator r from our data, we will develop the calculus using the previous formula, and we are expecting the same result than the one obtained with the `cor` function previously.

```
r <- (mean(peng$body_mass_g*peng$flipper_length_mm) -
        mean(peng$body_mass_g)*mean(peng$flipper_length_mm))/
  (sqrt(mean(peng$body_mass_g*peng$body_mass_g)-
          mean(peng$body_mass_g)^2)*sqrt(mean(peng$flipper_length_mm*
          peng$flipper_length_mm)-mean(peng$flipper_length_mm)^2))
r
```

```
[1] 0.872783
```

Indeed, we have obtained the same result. The next step should be take the sample size:

```
n <- nrow(peng)
n
```

```
[1] 233
```

Now, it's time to use the Fisher's transformation which converts r into a z-score (it is a quantity standardized). The transformation is:

$$z = \frac{1}{2}\log\left(\frac{1+r}{1-r}\right)$$

We compute this using our data:

```
z<-0.5*log((1+r)/(1-r))
z
```

```
[1] 1.344643
```

Now we have to determine the standard error of the transformed correlation coefficient which is:

$$SE_z = \frac{1}{\sqrt{n-3}}$$

Thus, in our case is:

```
se <- 1/sqrt(n-3)
se
```

```
[1] 0.06593805
```

And, since we wish to construct a 95% confidence interval, we should take $\alpha = 0.05$. Thus we have:

```
z_halfalpha <- qnorm(1-0.05/2)
z_halfalpha
```

```
[1] 1.959964
```

And finally, the confidenc interval is:

$$CI = [z - z_{\alpha/2} \cdot SE_z, z + z_{\alpha/2} \cdot SE_z]$$

Substituting all the quantities, which we have obtained before, we obtain:

```
z-z_halfalpha*se
```

```
[1] 1.215407
```

```
z+z_halfalpha*se
```

```
[1] 1.473879
```

Hence:

$$CI = [1.215407, 1.473879]$$

And, o obtain the confidence interval for the original correlation coefficient (r), we back-transform the endpoints of the interval:

```r
(r_low <- tanh(1.215407))
```

```
[1] 0.8382941
```

```r
(r_up <- tanh(1.473879))
```

```
[1] 0.9003148
```

And we finally obtain the 95% confidence interval for the correlation coefficient:

$$(0.8382941, 0.9003148)$$

Which seems a good interval, because as we saw $r = 0.872783$, and this is a quantity near to the true correlation coefficient, and r is within the computed interval.

(c) Bootstrap the correlation. Report and accurate confidence interval for the correlation coefficient and compare it with the one obtained using Fisher's method. How would you decide which option is best?

We will be using the bootstrap confidence intervals explained in the theory lectures. Remember that we have the observed correlation coefficient saved in the variable `cor_cof`. Now, we code the bootstrap method in this case. We are going to program the bootstrap method by hand. It is, we will not use any package for this purpose. We think that it is interesting, in order to familiarize ourselves with the bootstraping method, to program it on our own. Notwithstanding, we know that probably there are packages better implemented and which are less computational costly. In addition, `R` provides other strategies for do these kind of computations, for instance parallel programming is a very interesting approach for develop these calculus. We will be using the `parallel` library for do the computations in parallel, and use the different cores of the computer to do the computations faster. We saw how to use this package in the subject of *Statistical Software*, and we think that is interesting to apply it here.

First we will define a function that computes the correlation coefficient from the two first variables of a data frame.

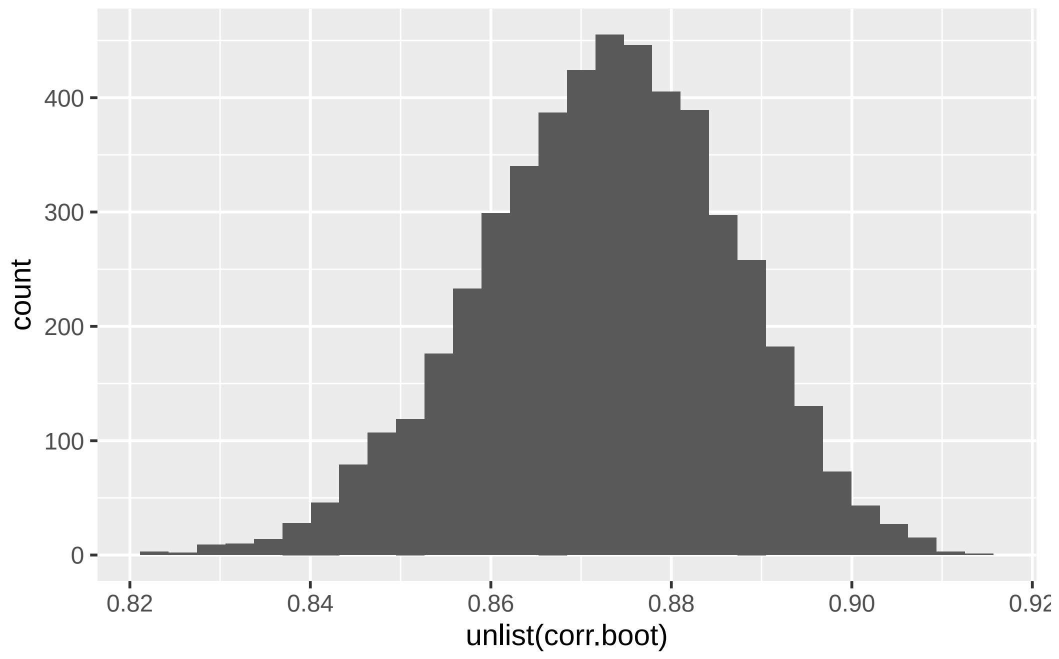
```r
cor_coeff <- function(df) {
  cor(df[, 1], df[, 2])
}
```

Now, we implement the bootstrap method. We have to take into account that in this case we should resample the indices. It is, in this case we have 233 observations, and two continuous variables for each observations. Thus, we can not resample one of the variables, or both by separately. Thus, the easiest way to do properly the resample of the bootstraping method is resampling the indices. In addition, for ensure the reproducibility of out results we will use the function `RNGkind`. In *Statistical Software* lectures we saw how to use this function for ensure the reproducibility of the results. Notice that in this case we can not use only the `set.seed` function, because we are using different cores of the computer. This is why we need to use the `RNGkind` function, together with the *"L'Ecuyer-CMRG"* parameter. Because we are parallelizing, we can be ambitious and take 5000 bootstrap observations.

```r
library(parallel)
RNGkind("L'Ecuyer-CMRG")
nCores <- detectCores()
df<-data.frame(peng$flipper_length_mm, peng$body_mass_g)
df_boot <- df
set.seed(1234)
corr.boot <- mclapply(X=1:5000, FUN=function(X){
  indices <- sample(1:nrow(df), replace = TRUE)
  k<-1
  for(i in indices){
    df_boot[k,] <- df[i,]
    k<-k+1
  }
  cor_coeff(df_boot)
}, mc.cores = nCores)
```

It is interesting to see how the bootstrap distribution behaves, we can study this using an histogram:

```r
ggplot() +
  aes(x=unlist(corr.boot))+
  geom_histogram()
```



Indeed, the bootstrap distribution is skewed to the right and centered near the true value of the correlation coefficient. Now, we finally give the 95% confidence interval using the bootstrap percentile intervals method:

```
confidence_interval <- quantile(unlist(corr.boot), c(0.025, 0.975))
confidence_interval
```

```
     2.5%     97.5%
0.8438128 0.8982653
```

We can observe that this confidence interval, [0.8428942, 0.8972064], contains the observed correlation coefficient which is $r = 0.872783$. In addition this confidence interval is narrower than te one obtained using the Fisher method. Then, it seems to be a better option than the interval obtained using Fisher's method.

I would decide which option is best taking into account different things: which is the narrowest one, which is the more easy to compute, and which is the less time consuming to compute with the computer. Both intervals are easy to compute, maybe the bootstrap one is easier if one understand how the bootsrap works. The bootstrap interval is more time consuming to compute with the computer. But, it is no necessary so much time, and using programming strategies this time can be improved. Finally, the bootsrap interval is narrower. So, personally, we would prefer the bootstrap interval, especially because the Fisher's transformation is a specific transformation that one need to remember, and on the other hand the bootstrap method is more flexible and can be applied with other purposes.

(d) Calculate the least-squares regression line to predict Y from X. What is the traditional confidence interval for the slope of the population regression line?

In order to calculate the least-square regression line we will use the `lm` function. We will print a `summary` in order to see the estimated intercept and coefficient of the linear model fitted.

```r
model <- lm(peng$body_mass_g~peng$flipper_length_mm)
summary(model)
```

```
Call:
lm(formula = peng$body_mass_g ~ peng$flipper_length_mm)

Residuals:
    Min      1Q  Median      3Q     Max 
-889.15 -265.44  -17.58  246.56 1281.99 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -5753.107    367.535  -15.65   <2e-16 ***
peng$flipper_length_mm    49.571      1.824   27.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 398.1 on 231 degrees of freedom
Multiple R-squared:  0.7618,	Adjusted R-squared:  0.7607 
F-statistic: 738.6 on 1 and 231 DF,  p-value: < 2.2e-16
```

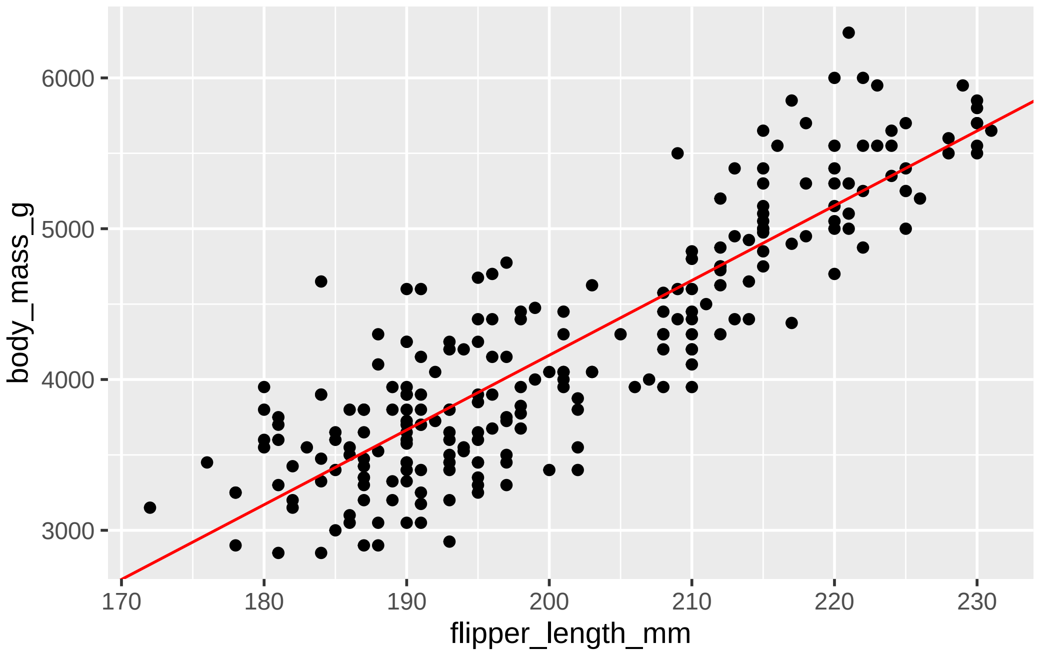
From this summary one can see that the regression line is

$$Y = -5753.107 + 49.571X.$$

In addition, we can see graphically the regression line obtained:

```r
interc <- coefficients(model)[1]
coeff <- coefficients(model)[2]

ggplot(peng) +
  aes(flipper_length_mm, body_mass_g)+
  geom_point() +
  geom_abline(intercept = interc, slope = coeff, color="red")
```



The traditional 95% confidence interval for the slope, let us denote the slope m, is the following one

$$CI(m) = \hat{m} \pm t_{\alpha/2,n-2} SE(m)$$

Where $\hat{m}$ is the estimated slope, $t_{\alpha/2,n-2}$ is the quantile of a t-student distribution with $n-2$ degrees of freedom, and $SE(m)$ is the standar error of the slope (we have the estimation of this standard error in the previous summary of the model). If we compute this confidence interval we obtain:

```
coefficients(model)[2]+qt(1-0.05/2,n-2)*1.824
```

```
peng$flipper_length_mm
               53.1651
```

```
coefficients(model)[2]-qt(1-0.05/2,n-2)*1.824
```

```
peng$flipper_length_mm
              45.97749
```

Hence, we have the confidence interval

$$CI_{95\%}(m) = [45.97749, 53.1651].$$

Notice that we can easily compute this confidence interval uisng the `confint` function:

```r
confint(model, level = 0.95)
```

```
                            2.5 %      97.5 %
(Intercept)            -6477.25580 -5028.95749
peng$flipper_length_mm    45.97742    53.16518
```

(e) Bootstrap the regression model. Give a 95% basic bootstrap confidence interval for the regression slope using the bootstrap.

Firstly, it is important to comment that one can compute the estimation of the slope using:

$$\hat{m} = \frac{Cov(X, Y)}{Var(X)}.$$

If we simply plug-in the covariance and variance estimators we obtain the estimation of the slope obtained in the previous section:

```r
cov_xy <- cov(peng$flipper_length_mm, peng$body_mass_g)
var_x <- var(peng$flipper_length_mm)
slope_estimate <- cov_xy / var_x
slope_estimate
```

```
[1] 49.5713
```

Then, we will be using this form to compute the estimation for bootstrapping. As we did in the section (c), we will be using the `parallel` library for develop the calculus in parallel. The bootstrap method structure will be the same, but now we are interested in estimate the slope of the regression model.

We first program a function that, given a data frame with two columns, compute the estimation of the slope. Notice that this function, for our purposes, expect the X variable in the first column, and the Y in the second.

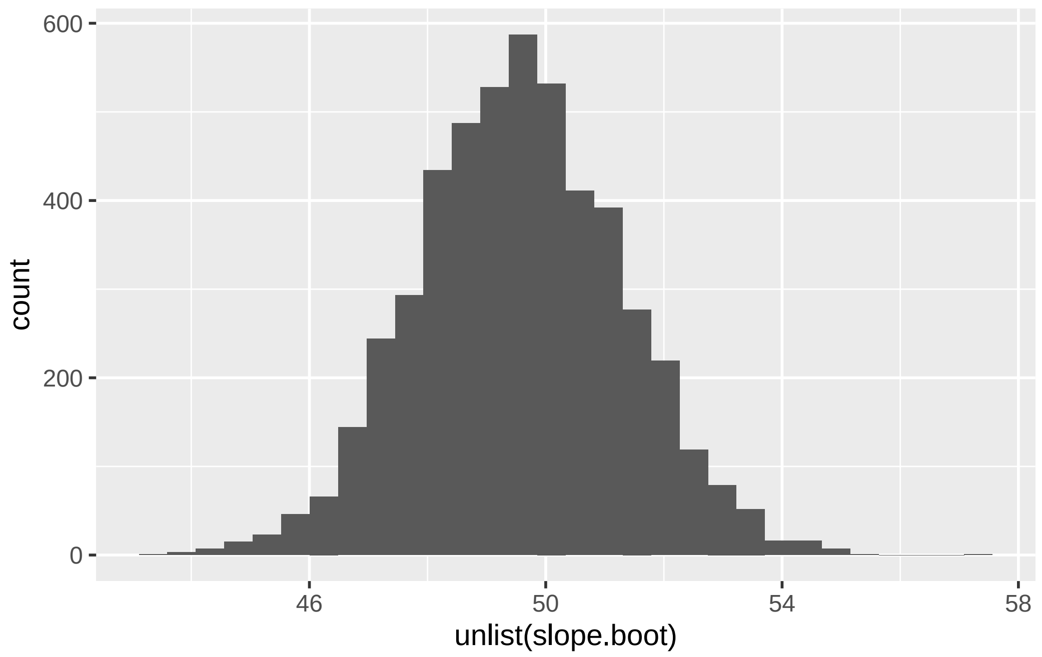
```r
slope_estim <- function(data){
  cov(data[,1], data[,2])/var(data[,1])
}
```

Now, again, we use the function `RNGkind` for the reprodicibility of the results, and again we use a similar scheme than the one used in (c) for bootstrapping:

```r
RNGkind("L'Ecuyer-CMRG")
nCores <- detectCores()
df<-data.frame(peng$flipper_length_mm, peng$body_mass_g)
df_boot <- df
set.seed(1235)
slope.boot <- mclapply(X=1:5000, FUN=function(X){
  indices <- sample(1:nrow(df), replace = TRUE)
  k<-1
  for(i in indices){
    df_boot[k,] <- df[i,]
    k<-k+1
  }
  slope_estim(df_boot)
}, mc.cores = nCores)
```

We are interested now in see how the bootsrap distribution is:

```r
ggplot() +
  aes(x=unlist(slope.boot))+
  geom_histogram()
```



The bootstrap distribution, as the reader can observe in the previous histogram, is centered at the true and behaves as one was expecting. Thus, we can now compute the bootstrap confidence interval, using again thr percentile method (as we did in the section (c)).

```r
conf_intervals <- quantile(unlist(slope.boot), c(0.025,0.975))
conf_intervals
```

```
    2.5%    97.5%
46.23904 53.04385
```

And in conclusion, the 95% confidence interval using bootstrapping is:

$$[46.23904, 53.04385]$$

(f) Which confidence interval do you think is a better option? How may your responses be different depending on the underlying distribution for X and Y. For instance, if we assume (bivariate) normality for X and Y or we do not.

Again the bootstrap interval is narrower than the other one. We can use similar arguments than the used in the section (c). In this case, we have to take into account that the traditional confidence interval for the slope is very extended and there are several options in the statistical software for compute it. Notwithstanding, the traditional method involves the computation of an standard error, which as we saw in the theory lectures is not an easy computation in general. Bootstrap is one tool that covers the problem of compute standard errors. Thus, if one need to do the computations without using existing packages, maybe is preferable to use bootstrapping. Because, for do the computation of the standard error, without using packages that implement this calculus, we may need also bootstrapping.

But, if we assume bivariate normality of for X, and Y, then we have that $aX + bY$ is normally distributed for all the possible constants $a, b \in \mathbb{R}$. In particular, if $X \sim N(\mu_x, \sigma_x^2), Y \sim N(\mu_y, \sigma_y^2)$, then $Y = a + mX \sim N(\mu_y, \sigma_y^2)$, and one can easily deduce the relations

$$\mu_y = a + m\mu_x, \quad \sigma_y^2 = m^2\sigma_x^2.$$

For derive the previous results one only need to apply the expectation and variance properties. Now, using the traditional methods for estimate the parameters of a normal, one can estimate the slope m. This traditional methods, based on the well-known properties of the normal distribution, will be easier to implement and also powerful. Thus, in this specific case, we should not use bootstrapping.

Exercise 2 Let $X \sim \mathcal{P}(\lambda)$, a Poisson distribution, and let $X_1, \dots, X_n$ be a simple random sample of X. We are interested in estimating $\theta = P(X = 0)$ from the sample.

(a) Find $\hat{\theta}_{MM}$ a method of moments estimate of θ. Is it unique?

We know that for a Poisson distribution with parameter λ, we have the probability function

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}.$$

We also know that the first moment of the population is

$$m_1 = E(X) = \lambda$$

and we want to estimate θ defined as

$$\theta = P(X = 0) = e^{-\lambda}.$$

Using the method of moments, we know we can use the sample mean $\bar{X}$ as an estimator for the population mean and write

$$\hat{\lambda} = \bar{X}$$

and then we can plug-in this estimator to estimate θ and obtain

$$\hat{\theta}_{MM} = e^{-\bar{X}}.$$

This estimator is not unique, as we can use other moments to approximate $\theta = h(m_1, ..., m_k)$, using different functions h. For example, we could use the second order moment instead of the first one. For the second population moment we have:

$$\begin{aligned} E(X^2) &= \sum_{k=0}^{\infty} k^2 \cdot \frac{e^{-\lambda} \cdot \lambda^k}{k!} = e^{-\lambda} \sum_{k=0}^{\infty} k \cdot \frac{k \cdot \lambda^k}{k!} \\ &= e^{-\lambda} \sum_{k=0}^{\infty} \frac{k \cdot \lambda^k}{(k-1)!} = e^{-\lambda} \lambda \sum_{k=1}^{\infty} \frac{\lambda^{k-1}}{(k-1)!} = e^{-\lambda} \cdot \lambda \cdot e^{\lambda} \\ &= \lambda^2 + \lambda \end{aligned}$$

and now equating to the second sample moment $\bar{X^2}$, we get

$$\hat{\lambda}^2 + \hat{\lambda} = \bar{X^2} \Longrightarrow \hat{\lambda} = \bar{X^2} - \bar{X}$$

and we get another estimator for θ:

$$\hat{\theta}_{MM} = e^{\bar{X^2} - \bar{X}}$$

thus proving that a method of moments estimate of θ is not unique.

(b) Find the asymptotic standard error (square root of the asymptotic variance) of $\hat{\theta}_{MM}, \sigma_{\hat{\theta}_{MM}}$ using the delta-method.

The delta method allows us to approximate the variance of a function of random variables. Let $g(\lambda) = e^{-\lambda}$ and $g(\bar{X}) = e^{-\bar{X}}$. We can write the variance of $\hat{\theta}_{MM}$ using the delta method as

$$\mathrm{Var}_\theta(\hat{\theta}_{MM}) \approx (g'(\lambda))^2 \cdot \mathrm{Var}(\bar{X}) = e^{-2\lambda} \cdot \frac{\lambda}{n}$$

where we have used that $\mathrm{Var}(\bar{X}) = \frac{\lambda}{n}$. Taking into account that $\sigma_{\hat{\theta}_{MM}}$ is the square root of the asymptotic variance of $\hat{\theta}_{MM}$ we get that

$$\sigma_{\hat{\theta}_{MM}} = \sqrt{e^{-2\lambda} \cdot \frac{\lambda}{n}}$$

(c) Find a second order approximation of the bias of $\hat{\theta}_{MM}, b_\theta\left(\hat{\theta}_{MM}\right)$. Is $\hat{\theta}_{MM}$ asymptotically unbiased?

First of all, recall that the bias of an estimator of θ is given by

$$b_\theta(\hat{\theta}) = E(\hat{\theta}) - \theta,$$

so we can write the bias of our estimator $\hat{\theta}_{MM}$ as

$$b_\theta(\hat{\theta}_{MM}) = E(\hat{\theta}_{MM}) - \theta = E(e^{-\bar{X}}) - e^{-\lambda}$$

To find a second-order approximation of the bias of $\hat{\theta}_{MM}$, we will expand this last expression using a Taylor series up to the second order around the expected value of $\bar{X}$, denoted as $E(\bar{X}) = \lambda$. The Taylor series expansion is:

$$\begin{aligned} e^{-\bar{X}} &= e^{-E(\bar{X})} - e^{-E(\bar{X})} \cdot (\bar{X} - E(\bar{X})) + \frac{e^{-E(\bar{X})}}{2} \cdot (\bar{X} - E(\bar{X}))^2 \\ &= e^{-\lambda} - e^{-\lambda} \cdot (\bar{X} - \lambda) + \frac{e^{-\lambda}}{2} \cdot (\bar{X} - \lambda)^2 \end{aligned}$$

Now let us take expectations and simplify:

$$\begin{aligned} E(e^{-\lambda}) &= e^{-\lambda} - e^{-\lambda} \cdot E(\bar{X} - \lambda) + \frac{E(e^{-\lambda})}{2} \cdot E((\bar{X} - \lambda)^2) \\ &= e^{-\lambda} - e^{-\lambda} \cdot 0 + \frac{E(e^{-\lambda})}{2} \cdot Var(\bar{X}) \\ &= e^{-\lambda} + \frac{e^{-\lambda}}{2} \cdot \frac{\lambda}{n} \end{aligned}$$

Hence, the bias becomes:

$$b_\theta(\hat{\theta}_{MM}) = e^{-\lambda} + \frac{e^{-\lambda}}{2} \cdot \frac{\lambda}{n} - e^{-\lambda} = \frac{e^{-\lambda}}{2} \cdot \frac{\lambda}{n}$$

Taking the limit $n \to \infty$ we can see that

$$\lim_{n\to\infty} b_\theta(\hat{\theta}_{MM}) = \lim_{n\to\infty} \frac{e^{-\lambda}}{2} \cdot \frac{\lambda}{n} = 0$$

so we can affirm that $\hat{\theta}_{MM}$ is asymptotically unbiased.

(d) Simulate a s.r.s of a Poisson distribution with $\lambda = 1$ and sample size $n = 15$ and compute $\hat{\theta}_{MM}$ and plug-in estimates of $\sigma_{\hat{\theta}_{MM}}$ and $b_\theta\left(\hat{\theta}_{MM}\right)$.

We first simulate a simple random sample of a Poisson distribution with $\lambda = 1$ and size $n = 15$:

```r
set.seed(69)
sample_pois <- rpois(15, 1)
```

Now, we compute $\hat{\theta}_{MM}$:

```r
mom <- exp(-mean(sample_pois))
mom
```

```
[1] 0.449329
```

Now, we compute the $\sigma_{\hat{\theta}_{MM}}$ using the plug-in method:

```r
std_err <- sqrt((exp(-2*mom)*mom)/15)
std_err
```

```
[1] 0.1104321
```

Finally we estimate $b_\theta\left(\hat{\theta}_{MM}\right)$ by using the expression derived in the previous section via Delta method:

```r
bias <- (1*exp(-1))/(2*15)
bias
```

```
[1] 0.01226265
```

(e) Use the bootstrap to estimate the standard error and the bias of $\hat{\theta}_{MM}$. Which estimates would you consider preferable? Why?

Here we will be using the same programming scheme than the one used in the first problem when we used bootstrapping. First we compute the standard error bootstrapping:

```r
RNGkind("L'Ecuyer-CMRG")
nCores <- detectCores()
set.seed(1234)
stderr.boot <- mclapply(X=1:5000, FUN=function(X){
  xnew <- sample(sample_pois, replace = TRUE)
  exp(-mean(xnew))
}, mc.cores = nCores)
sd(unlist(stderr.boot))
```

```
[1] 0.1061028
```

Again, we use this parallelization scheme to compute in this case the bias via bootstrapping:

```r
RNGkind("L'Ecuyer-CMRG")
nCores <- detectCores()
set.seed(1234)
```

```
bias.boot <- mclapply(X=1:5000, FUN=function(X){
  xnew <- sample(sample_pois, replace = TRUE)
  exp(-mean(xnew))
}, mc.cores = nCores)
mean(unlist(bias.boot))-exp(-mean(sample_pois))
```

```
[1] 0.01027549
```

Taking into account that both estimations are correct, we think that the estimates via bootstrap are preferable. This is because the estimations using the bootstrap method are smaller, and we are interested in little bias and standard error.