Nice effort of writing things in english ! ry to be more careful with some of the typos/ mispellings

TP1 - Data Mining

Lamyaa BOUZBIBA Eloise JULIEN Alexandre POUPEAU

Part one

Question a)

```
set.seed(0)

# creation of the vector
v = rnorm(6000*201, mean=0, sd=1)

# creation of the matrix 6000 lines and 201 columns
mat = matrix(data = v, nrow = 6000, ncol = 201)

# turn the matrix into a data frame
dataFrame = data.frame(mat)

# creation of the linear model
reg = lm(formula = X1 ~ . , data = dataFrame)

# uncomment that following part to see the summary of the regression
# summary(reg)
```

3.0 out of 3.0

Question b)

We assume that y and $\forall i$, μ_i is a vector of size 6000.

$$\begin{bmatrix} y & \mu_1 & \mu_2 & \cdots & \mu_{200} \end{bmatrix}$$

The associated equation is :

$$y = \beta_0 + \sum_{k=1}^{200} \beta_k \mu_k + \epsilon$$

We expect that all β_k are equals to 0 because there is no correlation between vectors. We expect that ϵ is the only thing different than 0, so ϵ is equal to y.

Here is an extract from the summary of the regression. Computed by “summary(reg)”. We only took the last part of it just to show that most of the p-value are not small, as expected, however some are not (they get like one star). The explanation of those one star (or plus) coefficients is present just after.

```
## X190         6.878e-03  1.322e-02   0.520  0.60301
## X191        -1.454e-02  1.304e-02  -1.115  0.26497
## X192        -6.778e-03  1.306e-02  -0.519  0.60365
## X193         6.833e-03  1.317e-02   0.519  0.60396
## X194         1.701e-02  1.299e-02   1.310  0.19021
## X195         1.372e-03  1.336e-02   0.103  0.91817
## X196        -2.030e-02  1.318e-02  -1.540  0.12372
## X197         5.899e-03  1.304e-02   0.452  0.65100
## X198         2.552e-02  1.339e-02   1.906  0.05676 .
## X199         1.480e-02  1.302e-02   1.136  0.25585
## X200         2.711e-02  1.332e-02   2.035  0.04190 *
## X201        -2.182e-02  1.296e-02  -1.683  0.09247 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.999 on 5799 degrees of freedom
## Multiple R-squared:  0.03336,    Adjusted R-squared:  2.157e-05
## F-statistic: 1.001 on 200 and 5799 DF,  p-value: 0.4848
```

Question c)

2.0 out of 3.0

Ok, you got the idea. However, you did not really explain the reason for this effect nor how one could try to correct it.

```
# stock in the vector coef all the p-values
coef <- summary(reg)$coefficients[, 4]

# here we select the p-values assessed as significantly non-zero at level 5%
coef[coef < 0.05]
```

```
##         X94        X125        X136        X141        X148        X169
## 0.022582923 0.026816252 0.040500240 0.038852768 0.004224608 0.020772670
##        X172        X177        X200
## 0.001814059 0.002914717 0.041895505
```

```
# count the number of those selected coefficients
length(coef[coef < 0.05])
```

```
## [1] 9
```

This result is not expected at all since we computed random independant variables from the classic Gaussian model $\mathcal{N}(0, 1)$. All the β_k should be equals to zero, however this is not the case because as there is many data, the linear model find some correlation between vectors that do not really exists. This is a problem because it happens in real study in compagny and we have to be aware of that effect in order to not misunderstand the result.

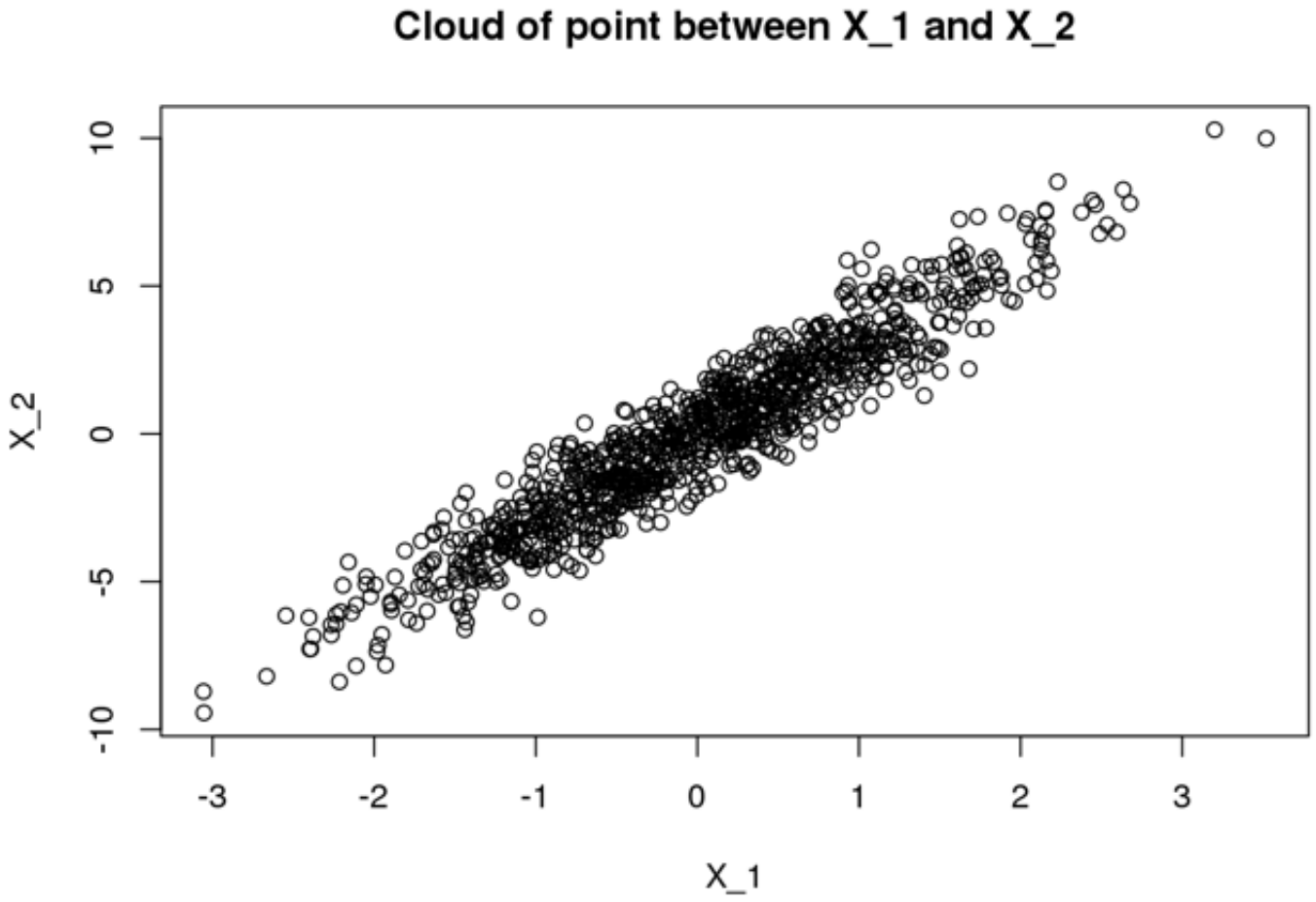
Part two

Question a)

1.0 out of 2.0

There's indeed a linear relation between the points, but we observe mainly an Elliptical form that should have been explained here.

```
set.seed(3)

# creation of three vectors
eps_1 <- rnorm(1000)
eps_2 <- rnorm(1000)
eps_3 <- rnorm(1000)

# creation of the model
X_1 = eps_1
X_2 = 3*X_1 + eps_2
Y = 2 + X_1 + X_2 + eps_3

# plot the cloud of points (X_1,X_2)
plot(X_1, X_2, main = "Cloud of point between X_1 and X_2")
```



The points are approximatively aligned, we obtain almost a positive linear straight line because there is a linear equation between X_1 and X_2. This is due to the linear fonction that link X_1 and X_2.

2.0 out of 2.0

Question b)

```
# creation of three new vectors
eps_1 <- rnorm(10)
eps_2 <- rnorm(10)
eps_3 <- rnorm(10)

# creation of the new model
X_1 = eps_1
X_2 = 3*X_1 + eps_2
Y = 2 + X_1 + X_2 + eps_3

# create a matrix
mat_reg1 <- matrix(data = c(Y, X_1), nrow = 10, ncol = 2)

# turn the matrix into a data frame
dataFrame1 = data.frame(mat_reg1)


# creation of the linear model
reg1 = lm(formula = X1 ~ . , data = dataFrame1)

# summary with every p-value and coefficients analysis
summary(reg1)
```

```
##
## Call:
## lm(formula = X1 ~ ., data = dataFrame1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3622 -0.4858  0.1965  0.6203  1.6274
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8004     0.3931   4.580   0.0018 **
## X2            3.7975     0.4214   9.011 1.84e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.172 on 8 degrees of freedom
## Multiple R-squared:  0.9103, Adjusted R-squared:  0.8991
## F-statistic: 81.19 on 1 and 8 DF,  p-value: 1.837e-05
```

The first one shows the correlation between Y and X_1 (written as $X2$ in the summary). We can clearly see that there is a link between those variables since the p-value of $X2$ and the p-value of the F-test are both small (approximately 10^-5).

```
# create a matrix
mat_reg2 <- matrix(data = c(Y, X_2), nrow = 10, ncol = 2)

# turn the matrix into a data frame
dataFrame2 = data.frame(mat_reg2)

# creation of the linear model
reg2 = lm(formula = X1 ~ . , data = dataFrame2)

# summary with every p-value and coefficients analysis
summary(reg2)
```

```
##
## Call:
## lm(formula = X1 ~ ., data = dataFrame2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3655 -0.5178  0.3155  0.4701  0.7793
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.67615    0.24524   6.835 0.000133 ***
## X2           1.30677    0.08913  14.661  4.6e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7415 on 8 degrees of freedom
## Multiple R-squared:  0.9641, Adjusted R-squared:  0.9596
## F-statistic: 214.9 on 1 and 8 DF,  p-value: 4.601e-07
```

The second one shows the correlation between Y and X_2 (written as $X2$ in the summary). We can notice that there is also a link between those two variables for the same reasons exposed just before. Plus, we can notice that both p-value in that case are smaller (approximately 10^-7). Therefore, we can deduce that Y might have a bigger correlation with X_2 than X_1.

2.0 out of 2.0

Question c)

```
# create a matrix
mat_reg3 <- matrix(data = c(Y, X_1, X_2), nrow = 10, ncol =3)

# turn the matrix into a data frame
dataFrame3 = data.frame(mat_reg3)

# creation of the linear model
reg3 = lm(formula = X1 ~ . , data = dataFrame3)

# summary with every p-value and coefficients analysis
summary(reg3)
```

```
##
## Call:
## lm(formula = X1 ~ ., data = dataFrame3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0842 -0.4256  0.0313  0.5550  0.7094
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7394     0.2293   7.586 0.000128 ***
## X2            1.1029     0.7052   1.564 0.161834
## X3            0.9610     0.2358   4.075 0.004717 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6824 on 7 degrees of freedom
## Multiple R-squared:  0.9734, Adjusted R-squared:  0.9658
## F-statistic: 128.1 on 2 and 7 DF,  p-value: 3.067e-06
```

Finally, when we analyze the summary of the regression of Y with X_1 (written as $X2$ in the summary) and X_2 (written as $X3$ in the summary), we can see that X_1 doesn't have a that small p-value. We were expected that both, X_1 and X_2, would have a small p-value, however only X_2 has. So what that this means ? One first explanation is that both X_1 and X_2 are useful to describe the model nevertheless X_2 contains the entire information contained in X_2 with a bit more. When we take a look at the model, this is exactly this : $X_2 = 3 * X_1 + \epsilon_2$. plot(density(Y), main = "Density Y") To explain that situation, we can talk about confounding factors. This can be explained because both X_1 and X_2 can be described with only X_1.

Those points of views are quiet consistent since we have the real model used to generate Y. So we can directly understand where the results come from. If there is a error we can analyze what's the main difference with the Y model.

0.5 out of 3.0 points

I see what you tried to do but it was not really this the result that we wanted. In fact, the right analytical form for the distribution of the coefficients is available in the course's booklet.

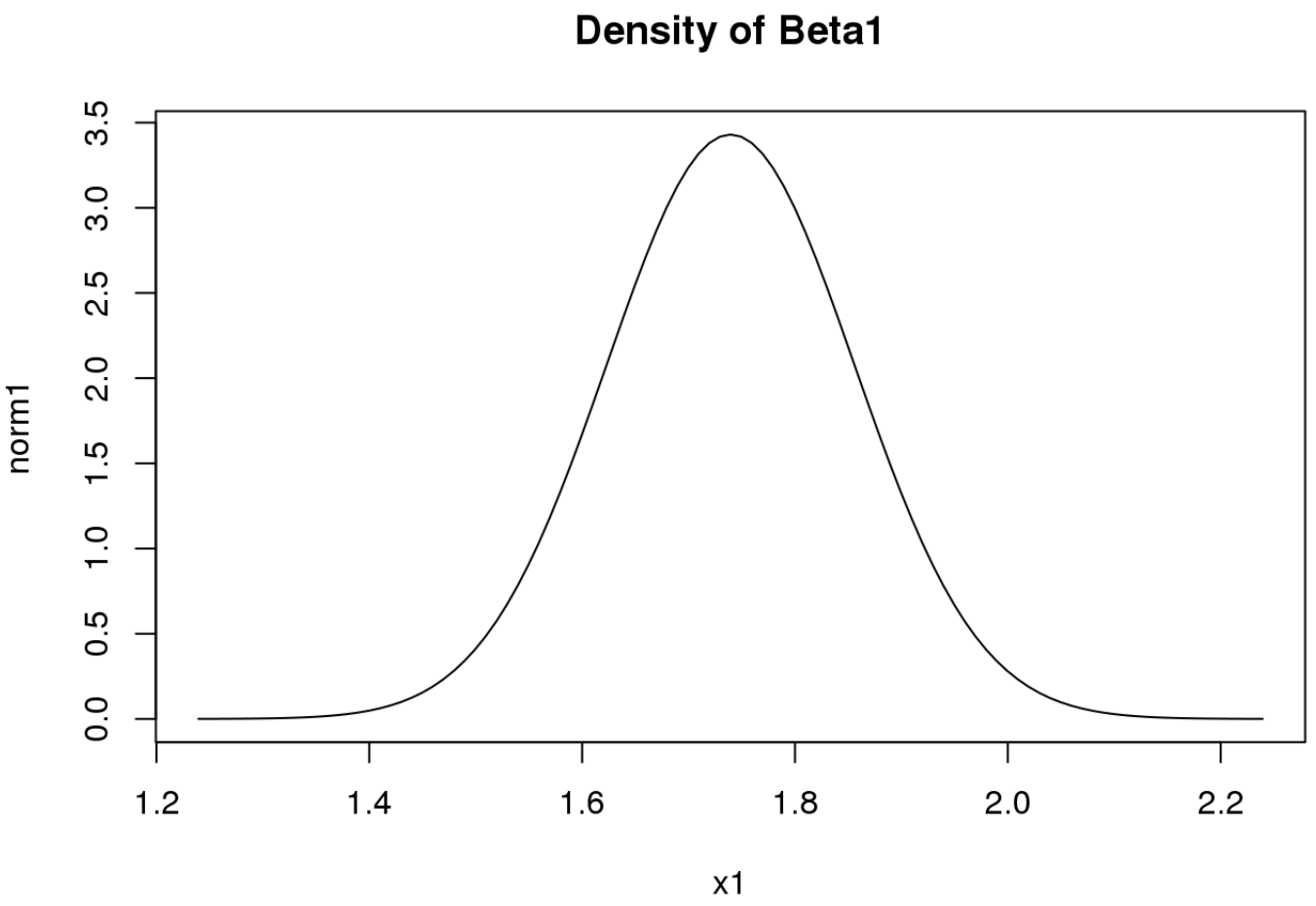
We don't see the lines with the quantiles in your figures ! Moreover, the most important conclusion for this question was seeing whether zero was part of the confidence intervals for each coefficient or not.

Question d)

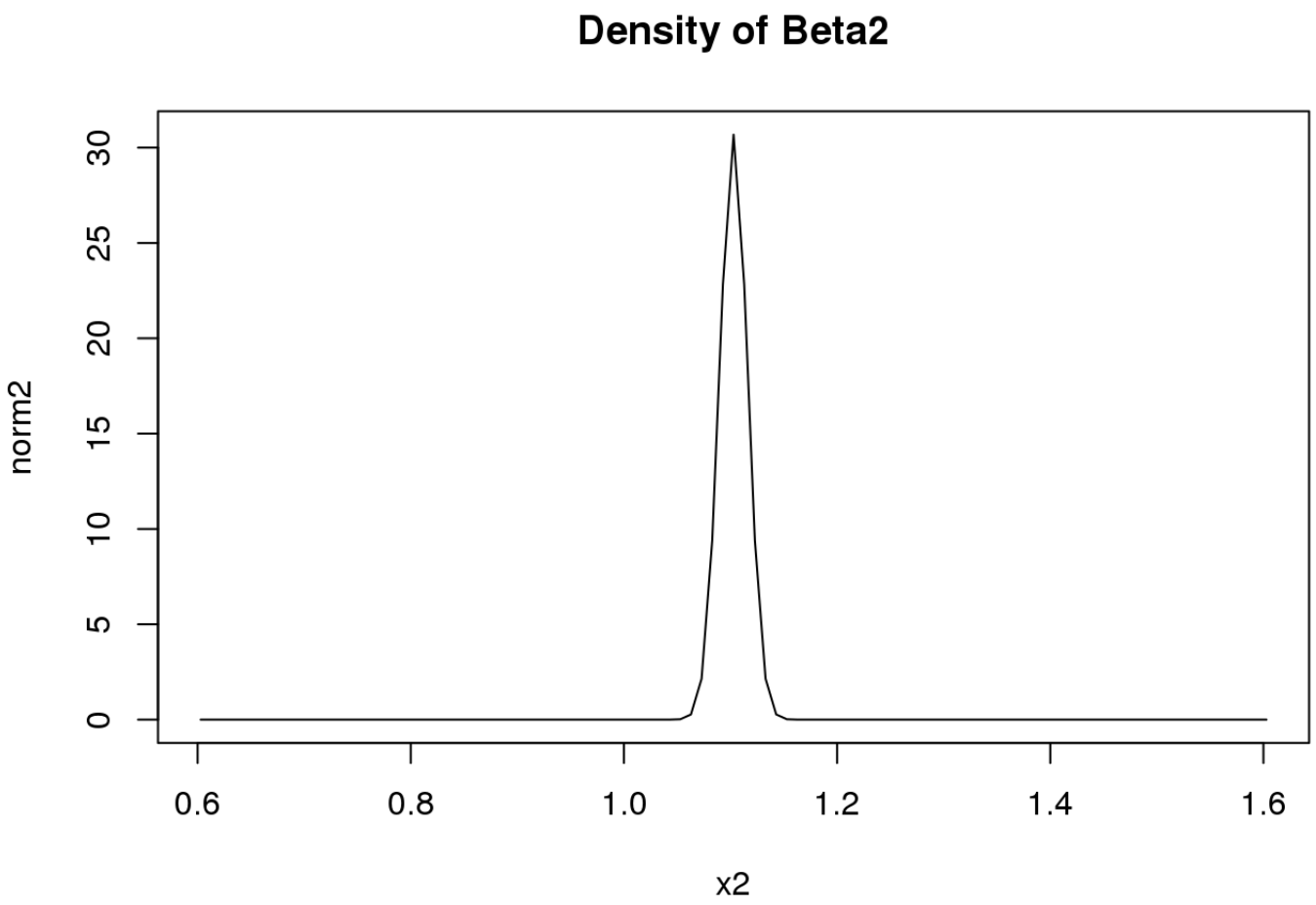
Definition in this type of model used for the next question :

$$\hat{\beta_1} = \mathcal{N}(\beta_1, \sigma^2/(n * s^2))$$

```
# for us
#plot(density(Y), main = "Density Y")

#plot(density(X_1), main = "Density X_1")

#plot(density(X_2), main = "Density X_2")

# for the question
beta1 <- reg3$coefficients[1]
beta2 <- reg3$coefficients[2]
n <- 10

x1 <- seq(beta1-0.5, beta1+0.5, 0.01)
norm1 <- dnorm(x1, mean = beta1, sd = 1/(n*var(X_1)))
plot(x1, norm1, main = "Density of Beta1", type = "l")
abline(v = quantile(norm1, 0.025))
abline(v = quantile(norm1, 0.975))
```




```
x2 <- seq(beta2-0.5, beta2+0.5, 0.01)
norm2 <- dnorm(x2, mean = beta2, sd = 1/(n*var(X_2)))
plot(x2, norm2, main = "Density of Beta2", type = "l")
abline(v = quantile(norm2, 0.025))
abline(v = quantile(norm2, 0.975))
```



We can notice that, as we expect in the precedent question, X_2 has a bigger link with Y than X_1 since we have less imprecision on β_2 and we have more information on its value.