# Labo 4.1: Régression linéaire à la main

Visseho Adjiwanou, PhD.

10 June 2021

## Travaux pratiques

Les données du Tableau ci-dessous provenant de Data Bank donnent le poids corporel (lb) et la longueur corporelle (cm) des louves :

| Observation | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Poids (lb) | 57 | 84 | 90 | 71 | 77 | 68 | 73 |
| Longueur (cm) | 123 | 129 | 143 | 125 | 122 | 125 | 122 |

1. Entrer les données dans R https://www.dummies.com/programming/r/how-to-create-a-data-frame-from-scratch-in-r/

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
## Warning: package 'tibble' was built under R version 3.6.2
```

```
## Warning: package 'tidyr' was built under R version 3.6.2
```

```
## Warning: package 'readr' was built under R version 3.6.2
```

```
## Warning: package 'purrr' was built under R version 3.6.2
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
## Warning: package 'forcats' was built under R version 3.6.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
poids <- c(57, 84, 90, 71, 77, 68, 73)
poids
```

```
## [1] 57 84 90 71 77 68 73
```

```
longueur <- c(123, 129, 143, 125, 122, 125, 122)
longueur
```

```
## [1] 123 129 143 125 122 125 122
```

```
louve <- data_frame(poids, longueur)
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

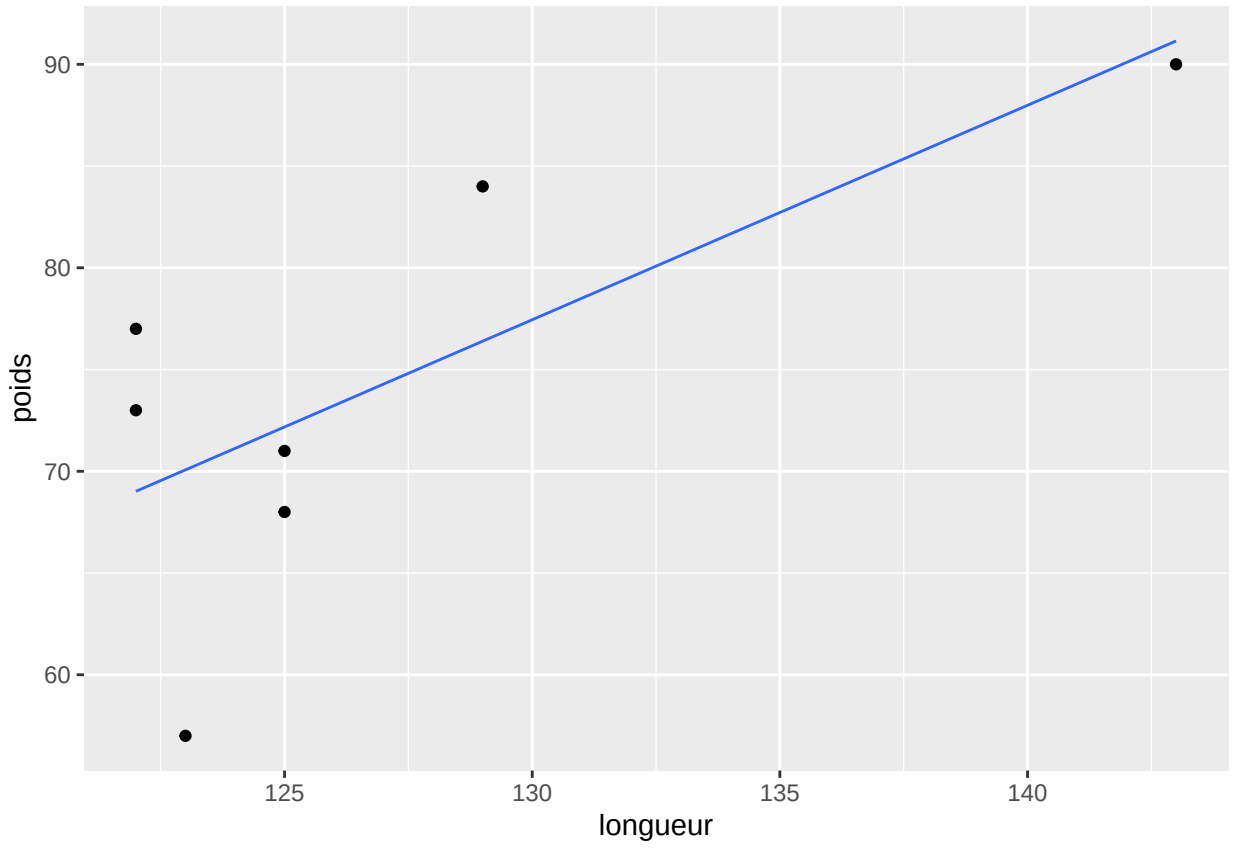
```
louve
```

```
## # A tibble: 7 x 2
##   poids longueur
##   <dbl>    <dbl>
## 1    57      123
## 2    84      129
## 3    90      143
## 4    71      125
## 5    77      122
## 6    68      125
## 7    73      122
```

2. Présenter un graphique montrant la relation entre le poids (variable dépendante) et la taille (variable indépendante)

```
ggplot(louve) +
  geom_point(aes(x = longueur, y = poids)) +
  geom_smooth(aes(x = longueur, y = poids), method = "lm", se = FALSE, size = .5)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



3. Quelle est le sens de cette relation?

Une relation positive. C'est à dire, qu'une plus grande taille est associée à un plus grand poids.

4. En estimant que cette relation est linéaire, calculer les paramètres $\alpha$ et $\beta$.

```
louve <-
  louve %>%
  mutate(dec_longueur = longueur - mean(longueur),
         dec_longueur_sq = dec_longueur ^ 2,
         dec_poids = poids - mean(poids),
         prod_croise = dec_poids*dec_longueur)

louve
```

```
## # A tibble: 7 x 6
##   poids longueur dec_longueur dec_longueur_sq dec_poids prod_croise
##   <dbl>    <dbl>        <dbl>           <dbl>     <dbl>       <dbl>
## 1    57      123           -4              16    -17.3        69.1
## 2    84      129            2               4      9.71       19.4
## 3    90      143           16             256     15.7       251.
## 4    71      125           -2               4     -3.29        6.57
## 5    77      122           -5              25      2.71      -13.6
## 6    68      125           -2               4     -6.29       12.6
## 7    73      122           -5              25     -1.29        6.43
```

```
coef_reg <-
  louve %>%
  summarise(beta = sum(prod_croise)/sum(dec_longueur_sq),
            alpha = mean(poids) - beta*mean(longueur))

coef_reg
```

```
## # A tibble: 1 x 2
##    beta alpha
##   <dbl> <dbl>
## 1  1.05 -59.6
```

5. Calculé le poids prédit

```
a <- coef_reg[2]
a
```

```
## # A tibble: 1 x 1
##   alpha
##   <dbl>
## 1 -59.6
```

```
b <- coef_reg[1]
b
```

```
## # A tibble: 1 x 1
##    beta
##   <dbl>
## 1  1.05
```

```
louve <-
  louve %>%
  mutate(poids_pred = -59.5586 + 1.053892   *longueur)
```

```
louve
```

```
## # A tibble: 7 x 7
##   poids longueur dec_longueur dec_longueur_sq dec_poids prod_croise poids_pred
##   <dbl>    <dbl>        <dbl>           <dbl>     <dbl>       <dbl>      <dbl>
## 1    57      123           -4              16    -17.3        69.1        70.1
## 2    84      129            2               4      9.71       19.4        76.4
## 3    90      143           16             256     15.7       251.         91.1
## 4    71      125           -2               4     -3.29        6.57       72.2
## 5    77      122           -5              25      2.71      -13.6        69.0
## 6    68      125           -2               4     -6.29       12.6        72.2
## 7    73      122           -5              25     -1.29        6.43       69.0
```

6. Calculé le résidu

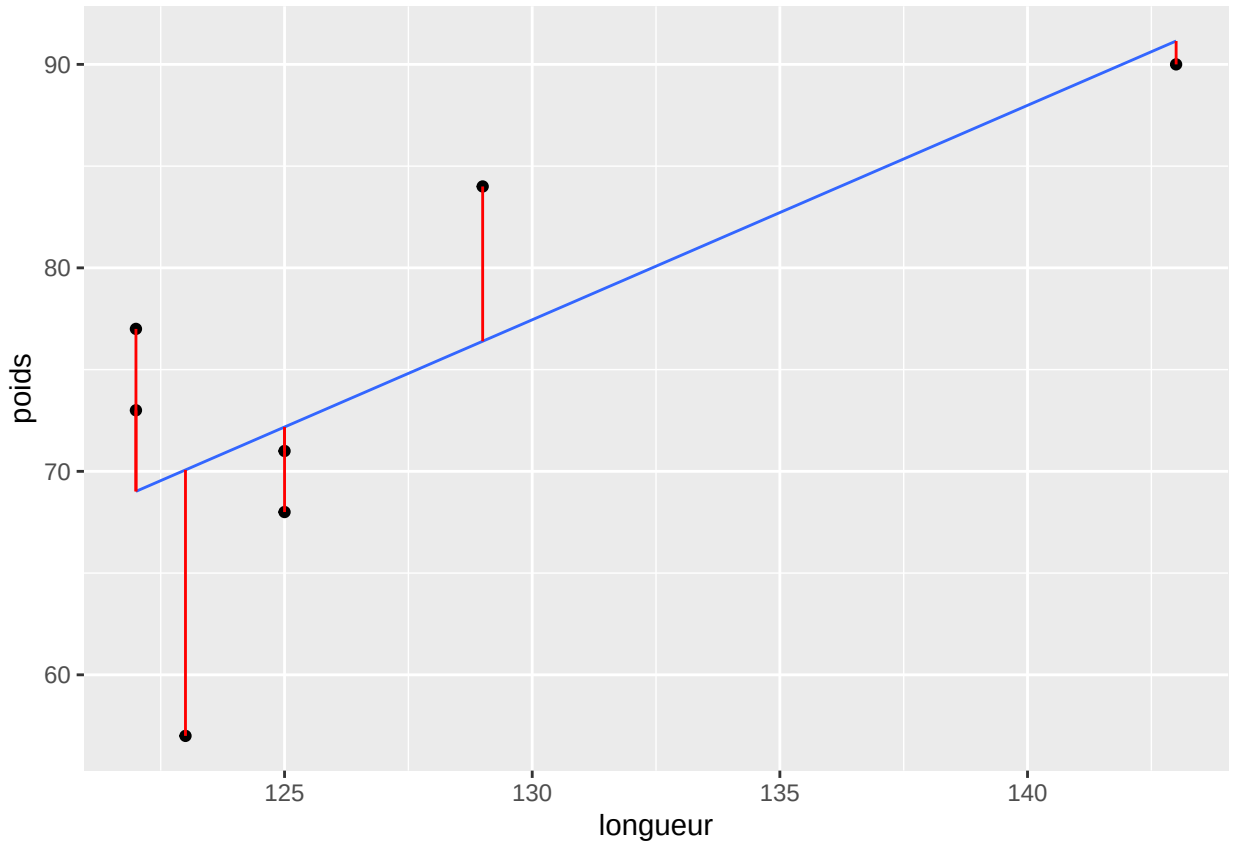
```
louve <-
  louve %>%
  mutate( residu = poids - poids_pred)

louve
```

```
## # A tibble: 7 x 8
##   poids longueur dec_longueur dec_longueur_sq dec_poids prod_croise poids_pred
##   <dbl>    <dbl>        <dbl>           <dbl>     <dbl>       <dbl>      <dbl>
## 1    57      123           -4              16    -17.3        69.1        70.1
## 2    84      129            2               4      9.71       19.4        76.4
## 3    90      143           16             256     15.7       251.         91.1
## 4    71      125           -2               4     -3.29        6.57       72.2
## 5    77      122           -5              25      2.71      -13.6        69.0
## 6    68      125           -2               4     -6.29       12.6        72.2
## 7    73      122           -5              25     -1.29        6.43       69.0
## # ... with 1 more variable: residu <dbl>
```

Graphique

```
ggplot(louve) +
  geom_point(aes(x = longueur, y = poids)) +
  geom_smooth(aes(x = longueur, y = poids), method = "lm", se = FALSE, size = .5) +
  geom_segment(aes(x = longueur, y = poids, xend = longueur, yend = poids_pred), color = "red")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




**Régardons ce qu'on observe en utilisant directement la fonction de regression de r**

```
library(broom)
```

```
## Warning: package 'broom' was built under R version 3.6.2
```

```
reg1 <- lm(formula =  poids ~ longueur , data = louve)
reg1
```

```
##
## Call:
## lm(formula = poids ~ longueur, data = louve)
##
## Coefficients:
## (Intercept)     longueur
##     -59.559        1.054
```

```
summary(reg1)
```

```
##
## Call:
## lm(formula = poids ~ longueur, data = louve)
##
## Residuals:
##       1       2       3       4       5       6       7
## -13.070   7.607  -1.148  -1.178   7.984  -4.178   3.984
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -59.5586    56.4062  -1.056   0.3393
## longueur      1.0539     0.4435   2.376   0.0634 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.105 on 5 degrees of freedom
## Multiple R-squared:  0.5304, Adjusted R-squared:  0.4365
## F-statistic: 5.647 on 1 and 5 DF,  p-value: 0.06345
```

```
tidy(reg1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)   -59.6     56.4       -1.06  0.339
## 2 longueur        1.05     0.443      2.38  0.0634
```

```
glance(reg1)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.530         0.436  8.11      5.65  0.0634     1  -23.4  52.8  52.6
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
results <- augment(reg1)

louve <-
  louve %>%
  mutate(longueur1 = longueur - 122)

reg2 <- lm(formula =  poids ~ longueur1 , data = louve)
reg2
```

```
##
## Call:
## lm(formula = poids ~ longueur1, data = louve)
##
## Coefficients:
## (Intercept)    longueur1
##      69.016        1.054
```

Rappel:

- Coéficients estimés :

$$\hat{\alpha} = \bar{Y} - \hat{\beta}\bar{X}$$

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

- Valeur prédite de la variable dépendante:

$$\hat{Y} = \hat{\alpha} + \hat{\beta}X$$

- Résidue:

$$\hat{\epsilon} = Y - \hat{Y}$$