sae_203_thierno_imany

Diallo Thierno | Imany Arango Catty

RESSOURCES: Statistiques descriptive 2 | Régression sur données reels

Table of Contents

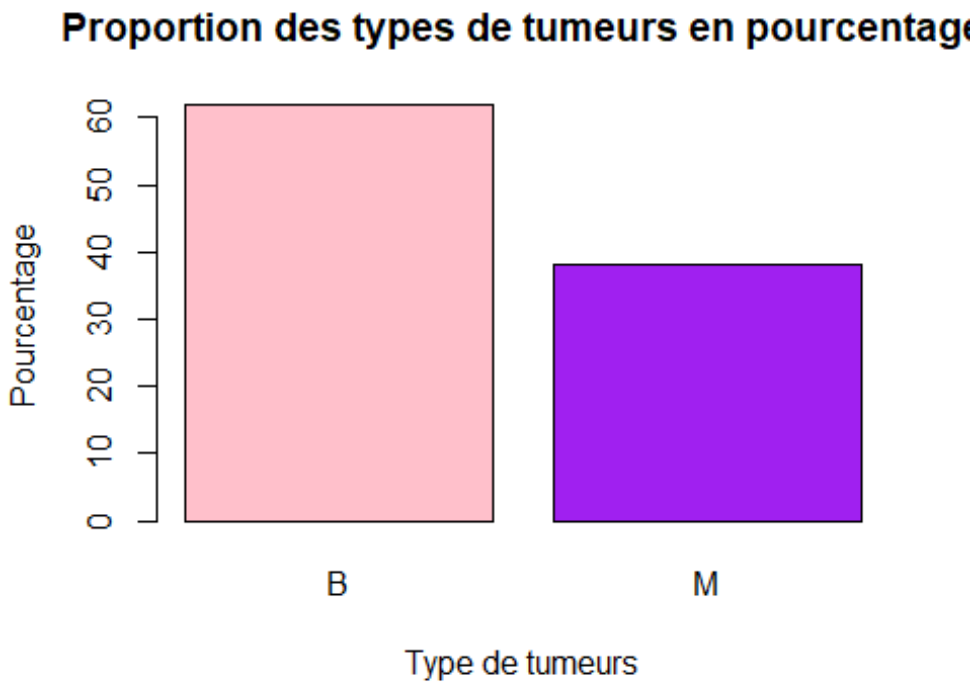
Lecture fichier et recupération des variables.

```
data = read.csv("data.csv")
data = data[, c('diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
                'area_mean', 'smoothness_mean', 'compactness_mean',
'concavity_mean', 'concave.points_mean', 'symmetry_mean',
                'fractal_dimension_mean')]

# pour supprimer les ligne avec des valers egales a 0 pour eviter les
probleme lorsque nous utiliserons log
data = data[data$concavity_mean != 0, ]
```

Partie 1

1. Proportion de tumeur maline et bénine.

```
#
barplot((table(data$diagnosis) / length(data$diagnosis)) * 100 ,
        col = c('pink', 'purple'),
        main = "Proportion des types de tumeurs en pourcentage",
        xlab = "Type de tumeurs",
        ylab = "Pourcentage"
        )
```

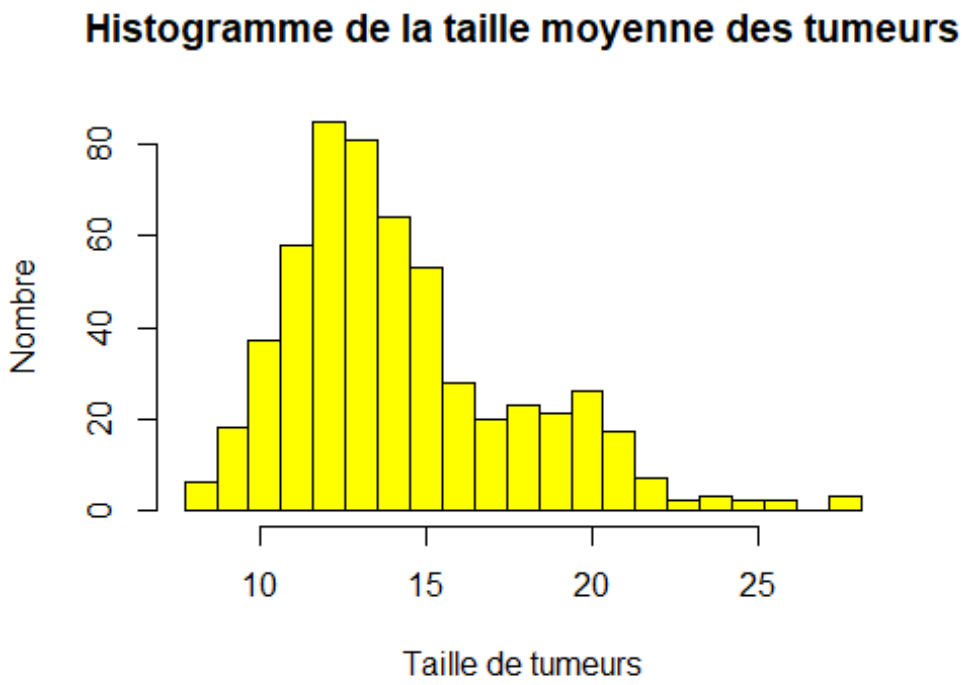


Ce graphique montre que la proportion de tumeurs malignes dans l'échantillon est d'environ 40% tandis que les tumeurs bénignes sont présentes à environ 60%. On constate donc qu'il y a une légère différence de population entre ces deux valeurs. Cependant, malgré

cette différence, l'echantillon permet de pouvoir détecter un discriminant qui expliquerait les tumeurs malignes. Il faut tout de même garder en tête que les tumeurs bénignes sont majoritaire. Il y a 1,5 fois plus de tumeurs bénignes que de tumeurs malignes

2. Histogramme du rayon moyen

```
summary(data$radius_mean)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.691  11.760  13.455  14.238  16.040  28.110

breaks = seq(from = min(data$radius_mean), to = max(data$radius_mean),
length = 22)
hist(data$radius_mean,
     main = "Histogramme de la taille moyenne des tumeurs",
     col = 'yellow',
     xlab = "Taille de tumeurs",
     ylab = "Nombre",
     breaks = breaks
     )
```



On remarque que la majorité des cellules ont un rayon entre 10 et 15 (50%), il y a également un faible nombre de cellules ayant des tailles moyennes assez importante jusqu'à 28 et d'autres ayant des tailles aux alentours de 15 et 20.

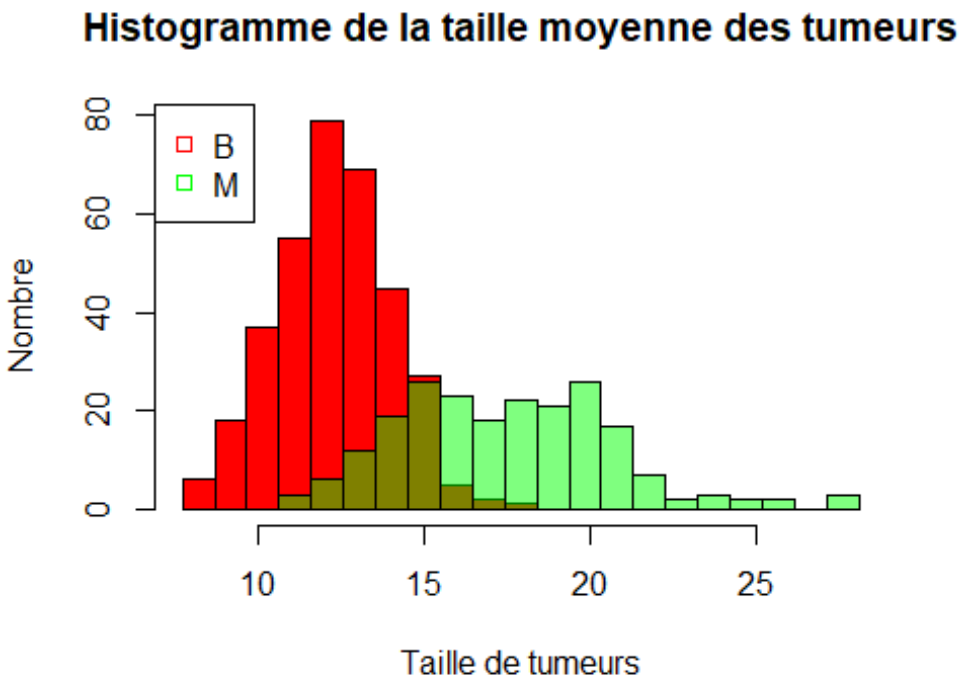
Donc dans l'ensemble la majorité des valeurs sont concentrées autour de la moyenne qui est d'environ 14,13. De plus 75% des valeurs est en dessous de 15 et par conséquent 25% au dessus.

On pourrait penser que les cellules avec une taille au dessus de 15 serait potentiellement etre malignes, car en sachant que les tumeurs bénignes sont majoritaires dans l'échantillon elles serait plus représentée en dessous de 15.

Et donc les tumeurs malignes serait quand à elles repartie majoritairement au dessus de 15. Sans oublié que forcément certaines de ces tumeurs sont également en dessous de 15.

3. Histogramme du rayon moyen par type de tumeurs

```
b = data[data$diagnosis == 'B', ]
m = data[data$diagnosis == 'M', ]

breaks = seq(from = min(data$radius_mean), to = max(data$radius_mean),
length = 22)

hist(b$radius_mean,
     main = "Histogramme de la taille moyenne des tumeurs",
     col = 'red',
     xlab = "Taille de tumeurs",
     ylab = "Nombre",
     breaks = breaks,
     xlim = c(min(data$radius_mean), max(data$radius_mean))
     )
hist(m$radius_mean,  add = T,
     col=rgb(0, 1, 0, 0.5),
     breaks = breaks,
     )

legend("topleft", c("B", "M"), pch = c(0), col = c("red", "green"))
```

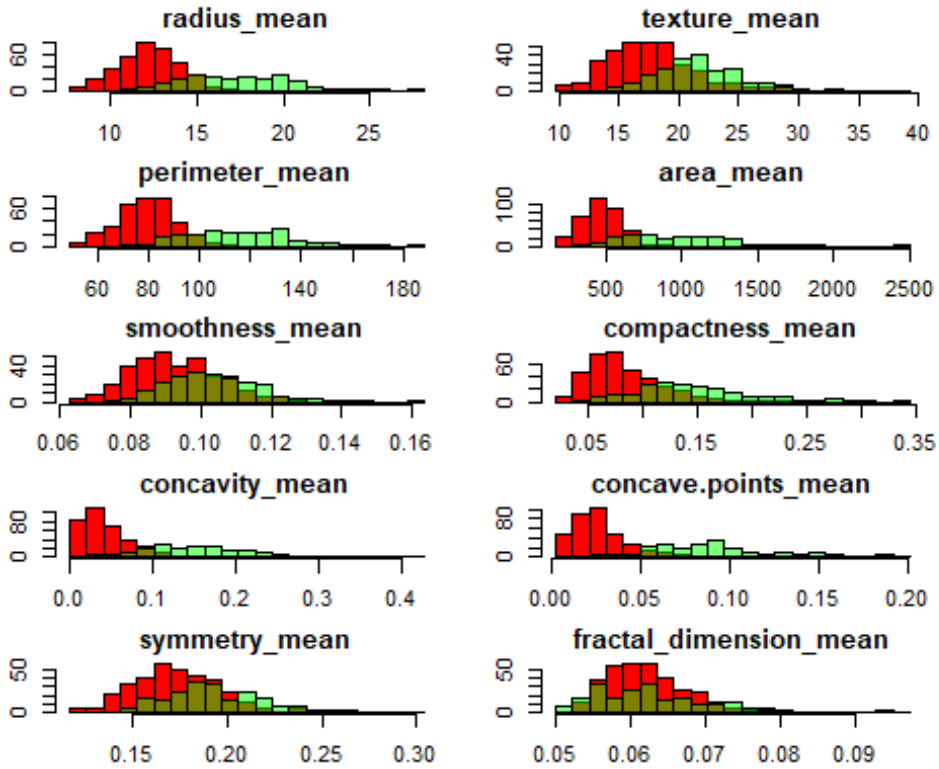


On remarque que les tumeurs bénignes sont beaucoup plus nombreuses que les tumeurs malignes et que leurs rayons est significativement plus petit que celui des tumeurs malignes. Tandis que les tumeurs malignes ont des tailles plus importantes pouvant aller jusqu'à 2 fois la taille d'un grand nombre de tumeurs bénignes.

4. Carte de graphique

```
vari = c('radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
'smoothness_mean', 'compactness_mean', 'concavity_mean',
'concave.points_mean', 'symmetry_mean', 'fractal_dimension_mean')

par(mfrow = c(5, 2), mar = c(2, 2, 2, 2))
for(i in vari){
  breaks = seq(from = min(data[, i]), to = max(data[, i]),  length = 22)
  hist(b[, i],
       main = i,
       col = 'red',
       xlab = i,
       ylab = "Nombre",
       breaks = breaks,
       xlim = c(min(data[, i]), max(data[, i]))
       )
  hist(m[, c(i)],  add = T,
       col=rgb(0, 1, 0, 0.5),
       breaks = breaks,
     )
```

```
  #legend("topright", c("B", "M"), pch = c(0), col = c("red", "green"))
}
```



5. Analyse et interpretation

- “radius_mean et perimeter_mean” : On remarque que les graphiques de perimeter_mean et radius_mean se comportent de manières similaires, car les repartitions des types de tumeurs reste quasiment la même et puis nous savons que nous pouvons trouver le périmètre en fonction du rayon, donc nous avons décidé de ne pas prendre en compte la variable”perimeter_mean” pour la suite ces analyse.

- “perimeter_mean” : on constate d’une part que les tumeurs bénignes sont concentrées dans l’intervalle d’environ 40 à 115 et le pic est aux alentours de 75-80, tandis que les tumeurs malignes sont beaucoup plus dispersés(de 75 jusqu’à environ 200). L’étendue est donc nettement plus large pour les malignes. Les tumeurs bénignes ont en général un périmetre inférieur à celui des malignes. Le seuil auquel nous pourrions identifié le type d’une tumeur serait environ 120.

- “area_mean” : Sur ce graphique les tumeurs bénignes sont assez sont assez concentrées entre 200 et 900 tandis que les malignes sont beaucoup plus étendue et se retrouvent entre 400 et 2500. Une valeur discriminante pour retrouver un type de tumeur serait 1100.

- “Smoothness_mean” : On remarque que pour ce graphique il y a une meilleure harmonie entre les des deux types de tumeurs. Les tumeurs bégnines ont une finesse moyenne entre 0.6 et 0.13 tandis que les malignes est entre 0.7 et 0.16, Leurs

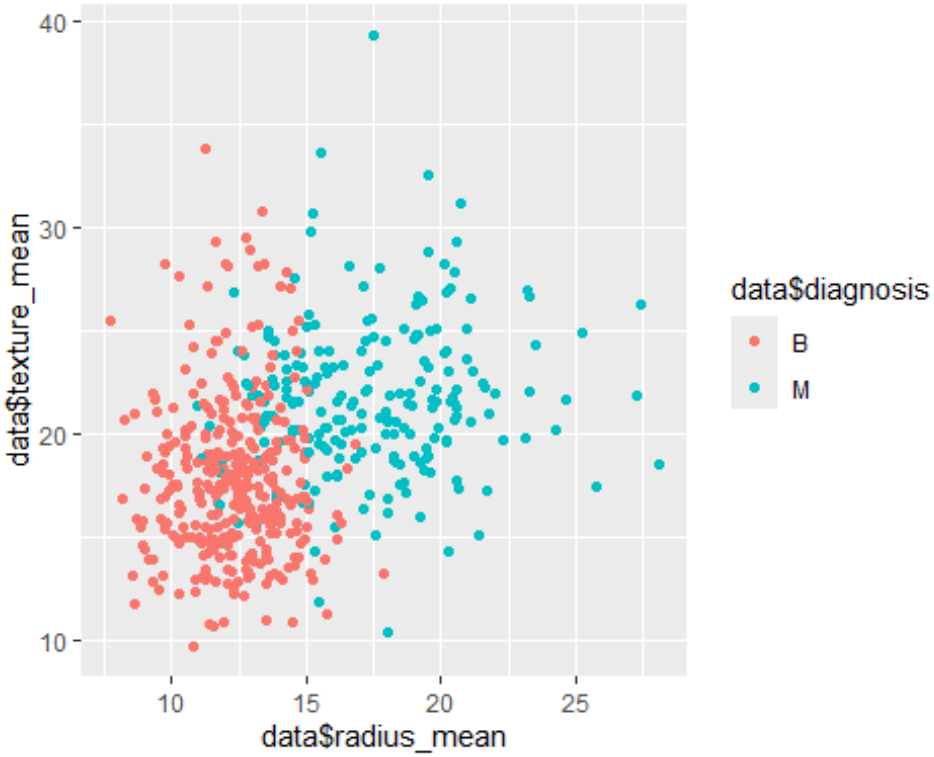
intervalles sont beaucoup plus superposés. Par conséquent on ne peut donc pas établir de valeur discriminante pour départager le type de tumeurs.

- “concavity_mean” : Sur ce graphique les tumeurs bénignes sont extrèmement concentrées vers la gauche dans un intervalle entre 0.0 et 0.12, alors que les malignes sont beaucoup plus étendues en ayant pour intervalle 0.09 à 0.23. Donc on pourrait choisir comme seuil pour distinguer les tumeurs la valeur 0.12.
- “texture_mean” : Les deux types de tumeurs semble se superposées sur une partie du graphique et ont un comportement similaire, le graphique ne permet pas de dégagé de valeur discriminante pour déterminer le type de tumeurs.
- “compactness_mean” : Les tumeurs bénignes sont plûtot concentrées sur la gauche aux alentour des valeurs proche de 0 avec une faible amplitude, tandis que les tumeurs malignes sont plus étendue et leurs valeurs plus importante. Au vu de ce graphique la valeur qui pourrait servir de déterminant serait environ 0.16.
- “concave.points_mean” : Dans le graphique concernant la moyenne des points concaves on distingue nettement la répartition entre les 2 types de tumeurs. Les bénignes sont concentrées autours de valeurs faible tandis que les malignes ont une étendue beaucoup plus large. La valeur discriminante pourrait être 0.07.
- symetrie_mean : Dans ce graphique les 2 types de tumeurs se comportent globalement de la même manière, c’est à dire que lorsque les tumeurs malignes sont nombreuses à une certaines symétrie, les tumeurs bénignes le sont également au même endroits. Pour les tumeurs benines, les valeurs partent d’environ 0.10 a 0.27 et que la majorité des valeurs sont comprise entre 0.15 et 0.20. Pour les tumeurs malines, les valeur prisent parteent de 0.15 a 0.30 et la majorité des valeur sont comprise entre 0.17 et 0.225. on constate que les deux histogramme sont tres similaire, et qu’on arrive pas a déterminer de valeur particuliaire (seuil) pour separe les deux categorie de tumeur.
- fractal_dimension_mean : d’une part, les tumeurs benines et malines on une etendu similaire, les valeur partent de 0.05 à 0.1. nous remarquons egalement que ces differente type de tumeur on a peu pres la meme distributon, la majorité des valeur sont en dessous de 0.07 et au dessus de 0.05. ainsi on ne peut pas determiner la categorie de la tumeur en fonction des de la valeur de la variable fractal_dimension_mean.
- Ainsi apres cette analyse, nous pouvons conserver les variable suivant avec les seuil suivant :
  - radius_mean : 15
  - perimeter_mean : 100
  - area_mean : 1100
  - compacness_mean : 0.16

- concavity_mean : 0.12
- concavity.point_mean : 0.07

6. Nuage de point entre texture et rayon

```
library(ggplot2)

fig = ggplot(data, aes(x = data$radius_mean, y = data$texture_mean)) +
  geom_point(aes(color = data$diagnosis))

fig
```



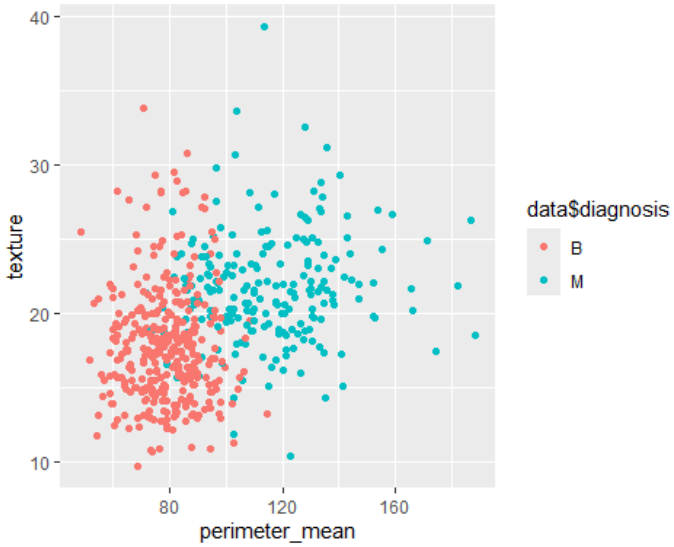
On peut voir ici qu'il y a une nette séparation entre les tumeurs bénignes et les tumeurs malignes notamment au niveau de la taille moyenne du rayon, la grande majorité des tumeurs bénignes sont en dessous du seuil de 15 pour la taille de rayon moyen tandis qu'inversement la plupart des tumeurs malignes sont au dessus de 15.

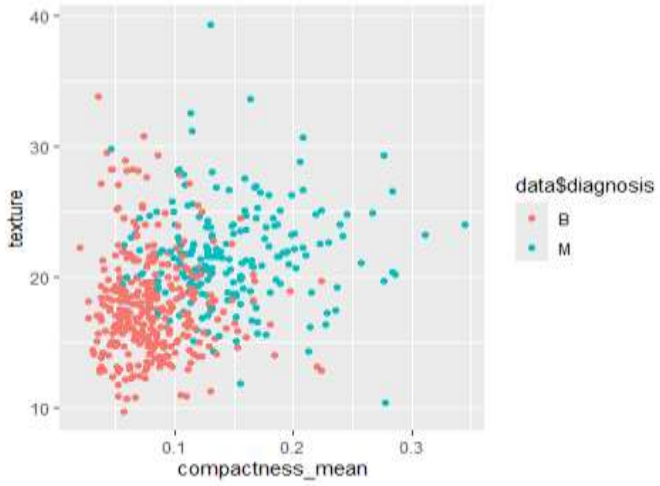
Pour ce qui est de la texture moyenne, il n'y a pas de grandes différences mais on peut quand même remarquer que les tumeurs bénignes ont tendance à avoir une texture globalement inférieur à celle des malignes puisqu'elles sont concentrent autour en dessous de 20 en valeur de texture moyenne ce qui n'est pas le cas des tumeurs malignes.

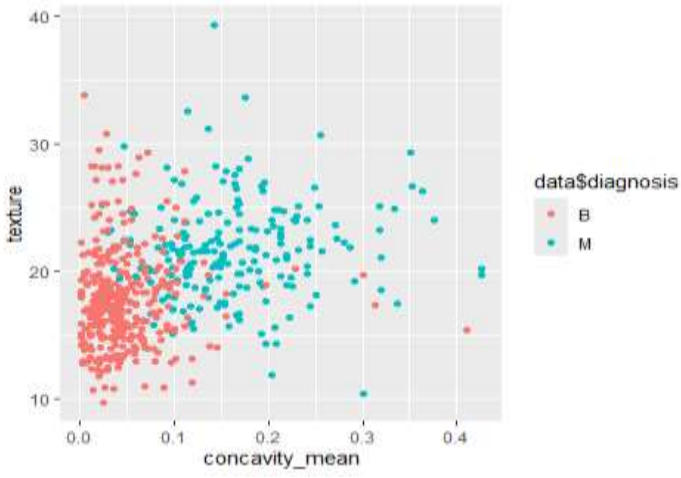
7. Analyses des variables conservé avec texture_mean

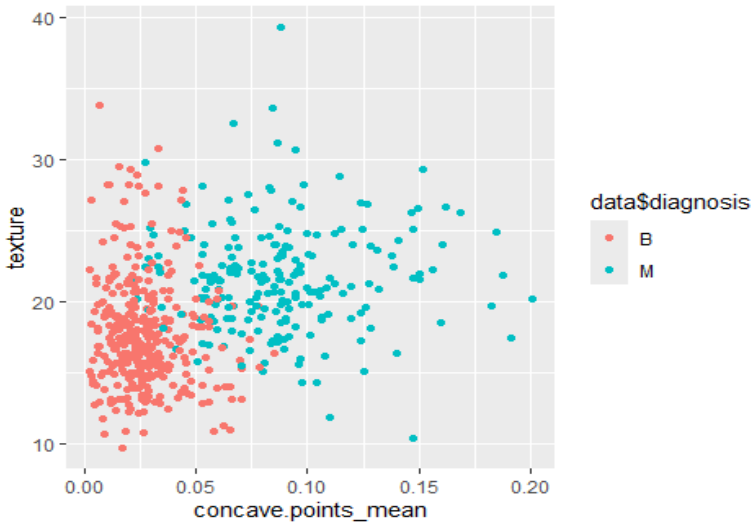
```
col = c('perimeter_mean', 'area_mean', 'compactness_mean', 'concavity_mean',
'concave.points_mean')
```

```
for(i in col){
  fig = ggplot(data,
               aes(x = data[, i],
                   y = data[, 'texture_mean'])) +
    geom_point(aes(color = data$diagnosis)) +
    labs(x = i, y = 'texture')
  print(fig)

}
```









- Nous avons réalisé tous ces nuages de points par rapport à texture_mean car on ne s'interresse qu'aux variables qui se trouve sur l'axe des abscisse, cela nous evite d'avoir des graphiques qui se repète et nous facilite l'interpretation.
- perimeter_mean : On observe sur ce nuage de point entre texture_mean et perimeter_mean que, les tumeur maline et benine sont bien separer a partir d'un rayon moyen de 100, en dessous de cette valeur, nous avons ne majorité de tumeur benine et presque pas de tumeur maline et inversement au dessus de 100
- area_mean : avec ce nuage de point, on peut voir, une séparation entre les tumeurs maline et benine au alentours de 740 pour l'aire moyen(area_mean), c'est a dire en dessous de cette valeur, nous avons ne majorité de tumeur benine et inversement au dessus.
- compactness_mean : d'une part, pour une valeur de compactness_mean superieur a 0.15, on constate qu'il y'a beaucoup de tumeur maline et presque pas de benine. D'autre part, en dessous de 0.1, il y'a de nombreuse tumeur benineet juste quelque tumeur maline. cependant entre 0.1 et 0.15, il y'a un peu de tous, mais ce sont les tumeur maline qui domine. ainsi on peut fixé comme seuil pour distinguer les benigne des maline 0.12
- concavity_mean : grace au nuage de point entre 'concavity_mean et concave.points_mean', on constate que, la majorité des tumeur maline on un concavity_mean superrieur à 0.1 et que la majorité des bénine on un concavity_mean inferieur a ce seuil. cela rejoint notre hypothese emise a la question 5
- concave.points_mean : grace au nuage de point entre texture et concave.points_mean, on constate que, la majorité des tumeur maline on un concave.points_mean superrieur à 0.05 et que la majorité des bénine on un concave.points_mean inferieur a ce seuil. cela rejoint notre hypothese emise a la question 5

- synthese : l'une des choses qui revient pour toute les varaibles est que les tumeurs maline on de plus grande valeur que les tumeurs benine. Nos nuages de point nous on permis de verifier toutes les hypothese s que nous avions emis a la question 5 et egalement d'etre un peu plus precis sur les valeurs seuil que nous avions detecter

8. proposition de model

Pour notre modèle, après avoir réperer les variables discriminante, nous avons egalement choisi des valeur seuil à partir des quelle ont pourra affirmer que une tumeur a plus de chance d'etre benigne que maligne. avec toutes les varables discriminante, nous calculons le seuil qui permet de faire la différence entre les types de tumeur et le score de chaque tumeur.

nb : les valeurs doivent etre normalisé en appliquant la formule suivante : $X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$

notre modèle est le suivant :

$$score = 0.22 * \text{area}_{\text{mean}} + 0.22 * \text{perimeter}_{\text{mean}} +$$

$$0.21 * \text{radius\_mean} + 0.15 * \text{concavity}_{\text{mean}} +$$

$$0.15 * \text{concave. points\_mean} + 0.05 * \text{compactness\_mean}$$

- Si $score < seuil$, alors le tumeur est bénine
- Si $score > seuil$ alors la tumeurs est maline

```
#fonction pour normaliser
# prend en parametre un vecteur et retourne un vecteur normalisé
normalise = function(vec){
  n = (vec - min(vec)) / (max(vec) - min(vec))
  return(n)
}

#calcule du seuil
#le seuil € [0;1]
area_mean = (700 - min(data$area_mean)) / (max(data$area_mean) -
min(data$area_mean))

perimeter_mean = (100 - min(data$perimeter_mean)) / (max(data$perimeter_mean)
- min(data$perimeter_mean))

radius_mean = (15 - min(data$radius_mean)) / (max(data$radius_mean) -
min(data$radius_mean))

concavity_mean = (0.1 - min(data$concavity_mean)) / (max(data$concavity_mean)
- min(data$concavity_mean))
```

```r
concave.points_mean = (0.05 - min(data$concave.points_mean)) /
(max(data$concave.points_mean) - min(data$concave.points_mean))

compactness_mean = (0.12 - min(data$compactness_mean)) /
(max(data$compactness_mean) - min(data$compactness_mean))


seuil = 0.22 * area_mean + 0.22 * perimeter_mean + 0.21 * radius_mean + 0.15
* concavity_mean + 0.15 * concave.points_mean + 0.05 * compactness_mean

# calcule de la précision de notre modele

# recuperation des variable discriminante
data_normal = data[c('area_mean', 'perimeter_mean',
'radius_mean','concavity_mean', 'concave.points_mean', 'compactness_mean')]

# normalisation des données
for(i in colnames(data_normal)){
  data_normal[, i] = normalise(data[, i])
}

# estimation si une tumeur est maline ou benine
data_normal['Estimation'] = ifelse((0.22 * data_normal[, 'area_mean'] +
                            0.22 * data_normal[, 'perimeter_mean'] +
                            0.21 * data_normal[, 'radius_mean'] +
                            0.15 * data_normal[, 'concavity_mean'] +
                            0.15 * data_normal[, 'concave.points_mean'] +
                            0.05 * data_normal[, 'compactness_mean']) >=
seuil, "M", "B")

# recuperation de l'etat reel des tumeurs
data_normal['diagnosis'] = data['diagnosis']

# calcul du score de notre modèle
score = sum(data_normal$Estimation == data_normal$diagnosis) /
dim(data_normal)[1]

print(paste("le score de notre modele est de ", round(score*100, digits = 2),
"%"))
```

```
## [1] "le score de notre modele est de  89.75 %"
```

Notre modèle a pu prédire correctement 89,75% des type de tumeur.

Partie 2

1. calcul des combinaisons possibles et calcule de correlation

A. calcule des combinaison possible

```
col = c('radius_mean', 'perimeter_mean', 'area_mean', 'compactness_mean',
'concavity_mean')

nb = factorial(length(col)) / (factorial(2) * factorial(length(col) - 2))
print(paste("il y'a ", nb, "cas possible"))
```

```
## [1] "il y'a  10 cas possible"
```

```
n = 1
vec_nu = matrix(rep(0, 20), nrow = 10, ncol = 2)
for(i in 1:length(col)){
  for(j in (i+1):length(col)){
    if(i < length(col)){
      vec_nu[n, c(1, 2)] = c(col[i], col[j])
      n = n + 1

    }
  }
}
```

les combinaison possible sont :

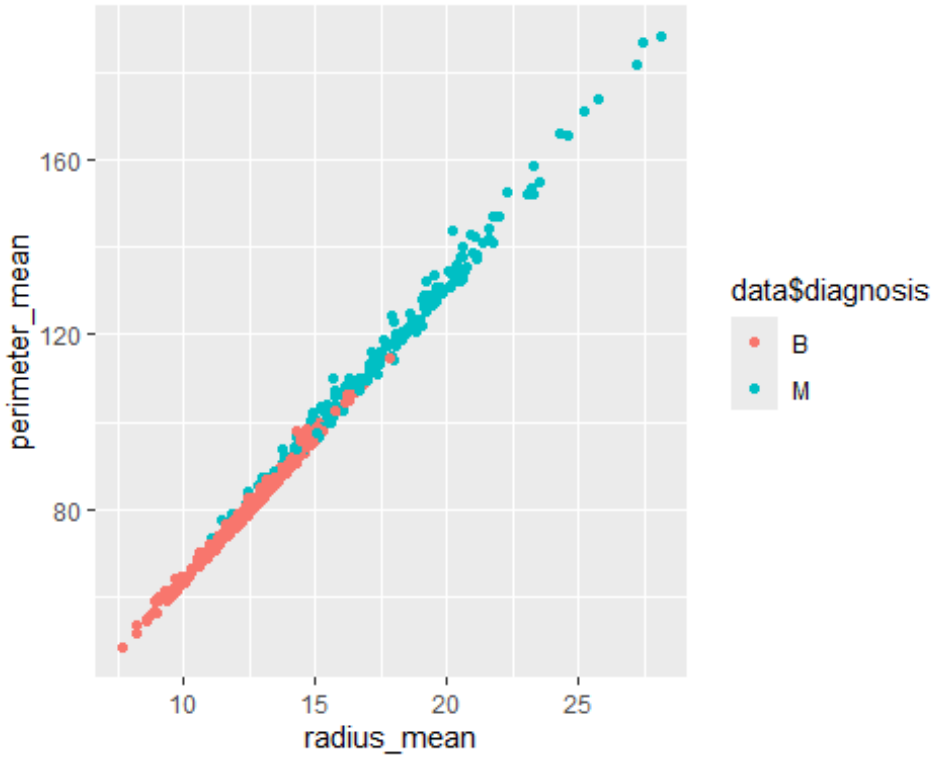
```
vec_nu
```

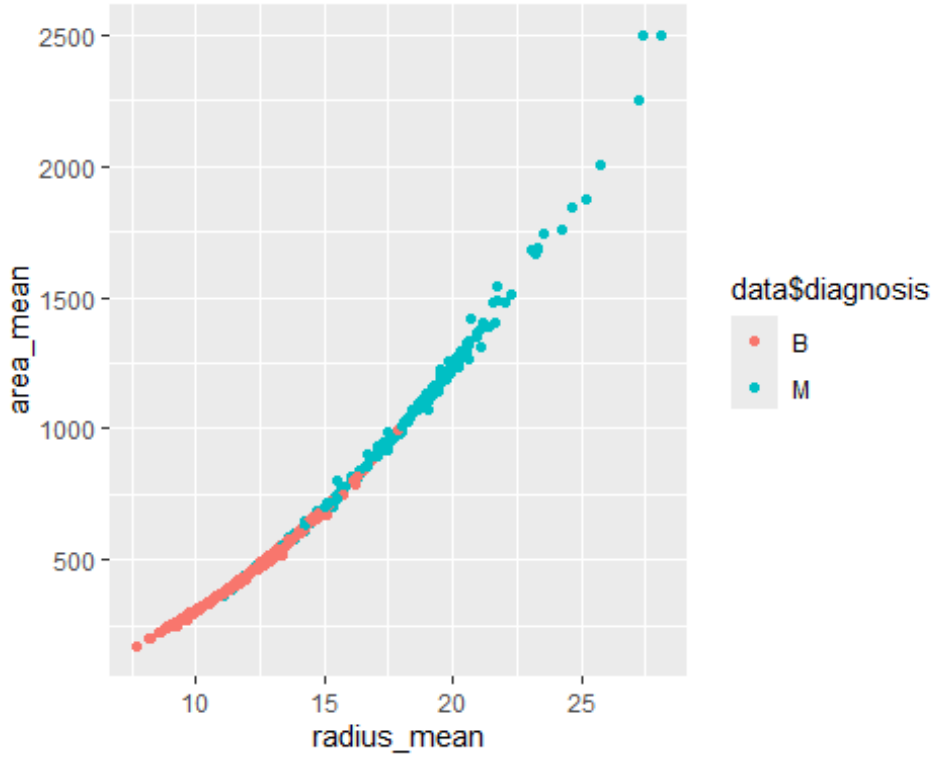
```
##       [,1]               [,2]
##  [1,] "radius_mean"      "perimeter_mean"
##  [2,] "radius_mean"      "area_mean"
##  [3,] "radius_mean"      "compactness_mean"
##  [4,] "radius_mean"      "concavity_mean"
##  [5,] "perimeter_mean"   "area_mean"
##  [6,] "perimeter_mean"   "compactness_mean"
##  [7,] "perimeter_mean"   "concavity_mean"
##  [8,] "area_mean"        "compactness_mean"
##  [9,] "area_mean"        "concavity_mean"
## [10,] "compactness_mean" "concavity_mean"
```

B. Nuage de point entre les variable

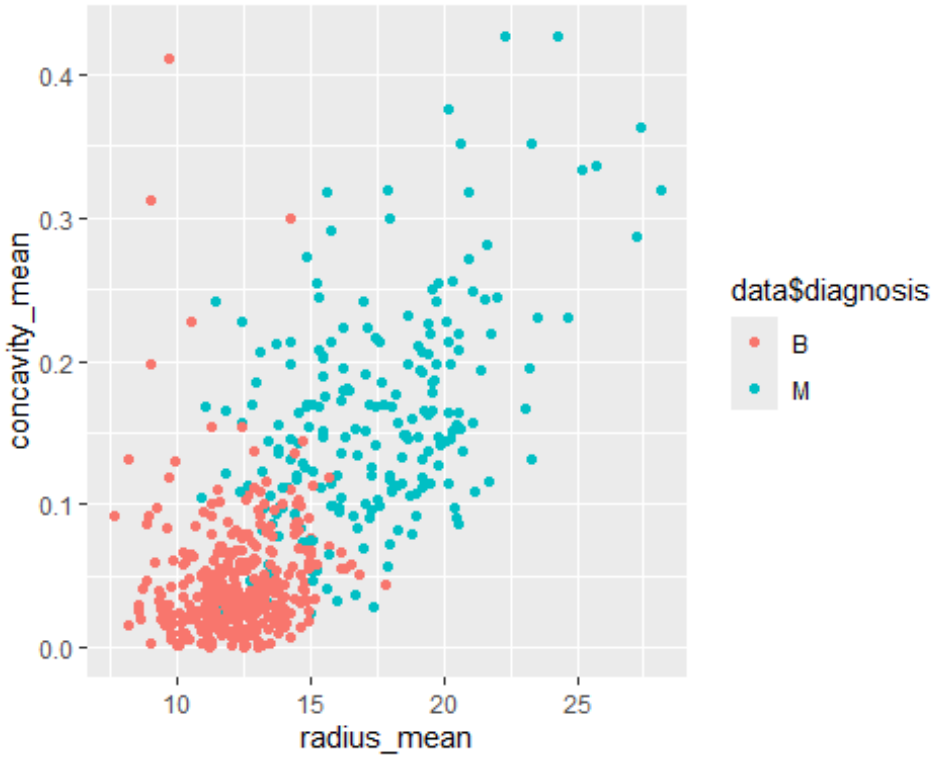
```
for(i in 1:10){
  fig = ggplot(data,
               aes(x = data[, vec_nu[i, ][1]],
                   y = data[, vec_nu[i, ][2]]
                  )
              ) +
  geom_point(aes(color = data$diagnosis)) +
  labs(x = vec_nu[i, ][1], y = vec_nu[i, ][2])
```

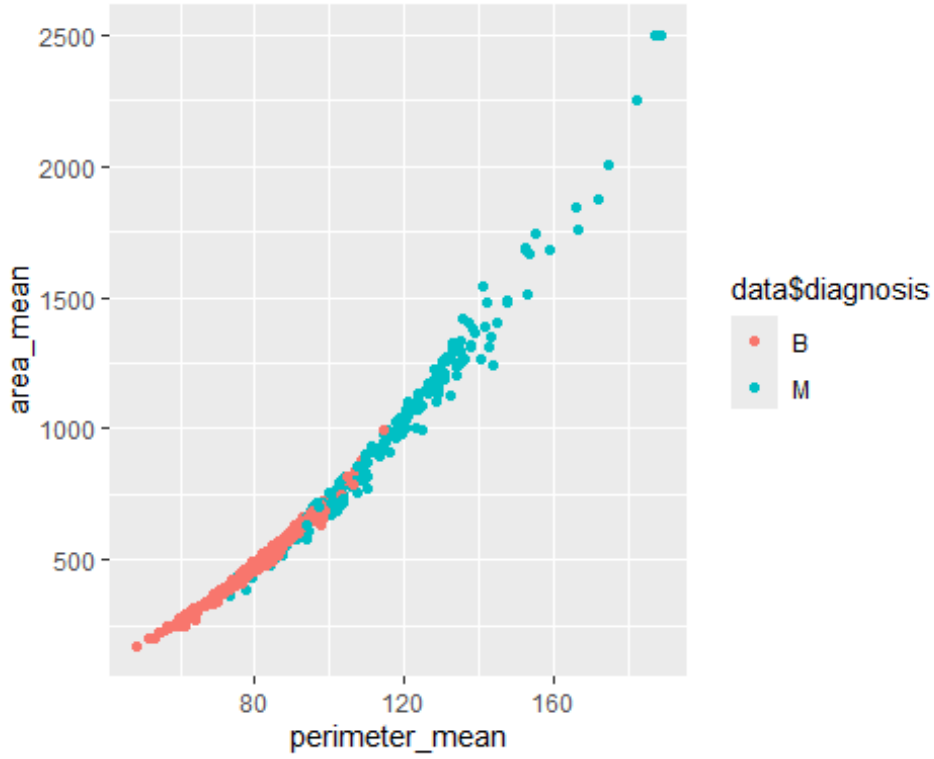
```
  print(fig)
}
```

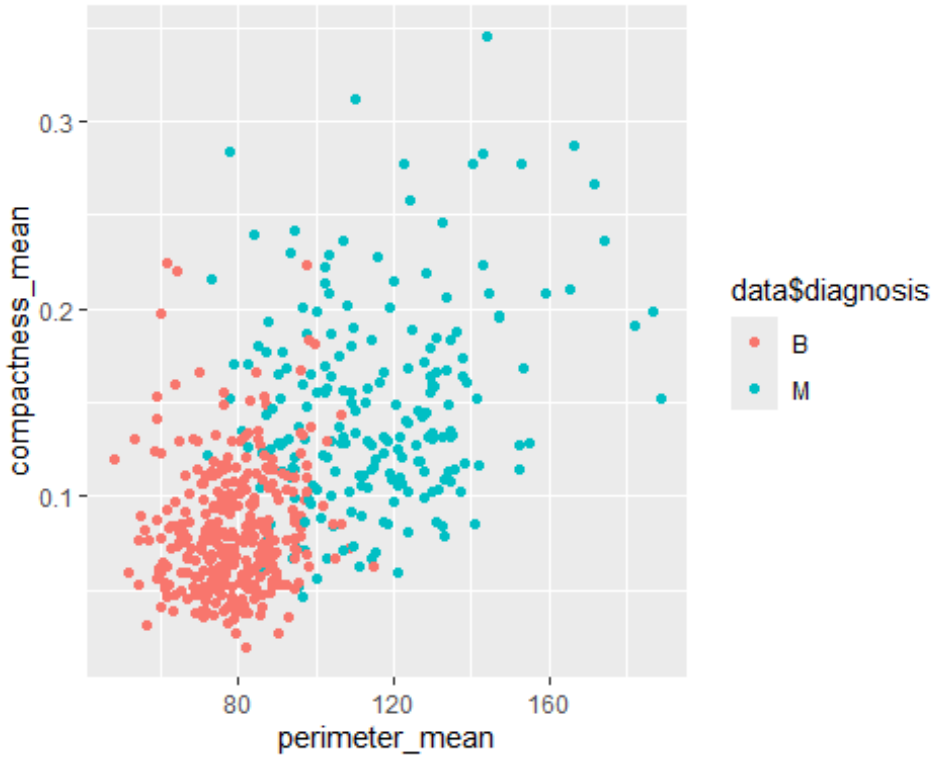

perimeter_mean
160
120
80
10
15
20
25
radius_mean
data$diagnosis
B
M

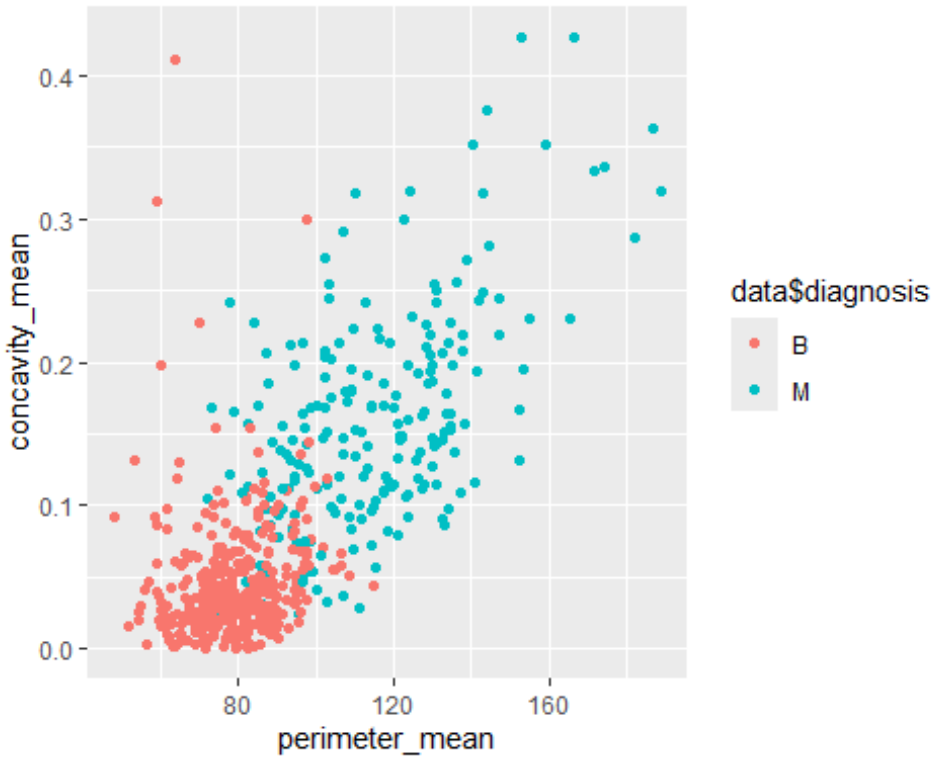

area_mean
2500
2000
1500
1000
500
10
15
20
25
radius_mean
data$diagnosis
B
M

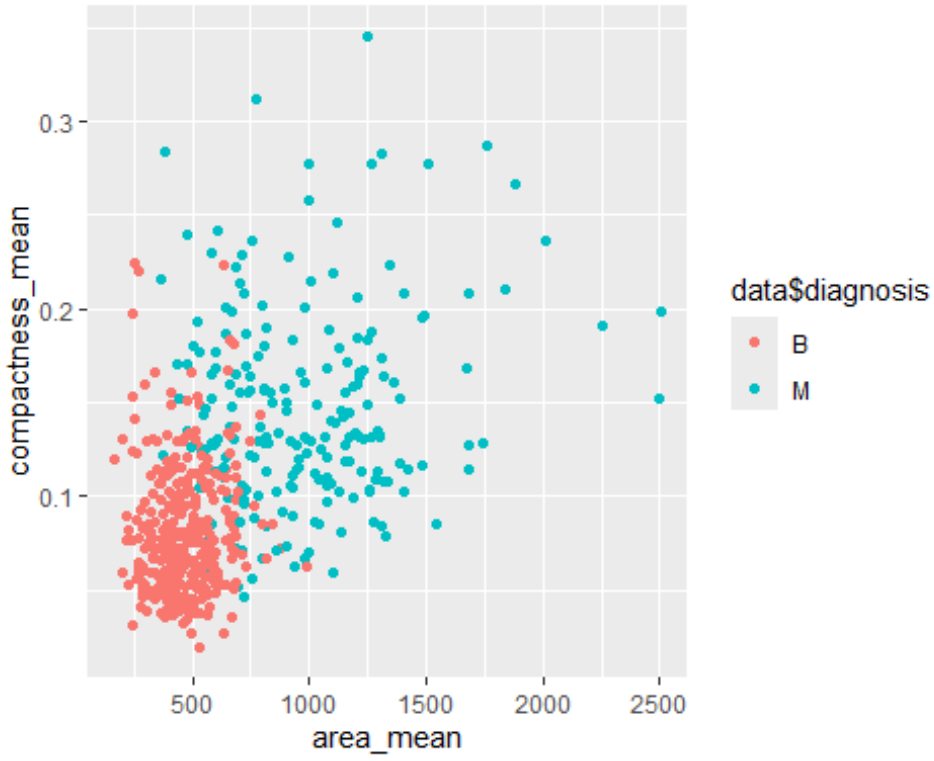

compactness_mean
0.3
0.2
0.1
10
15
20
25
radius_mean
data$diagnosis
B
M

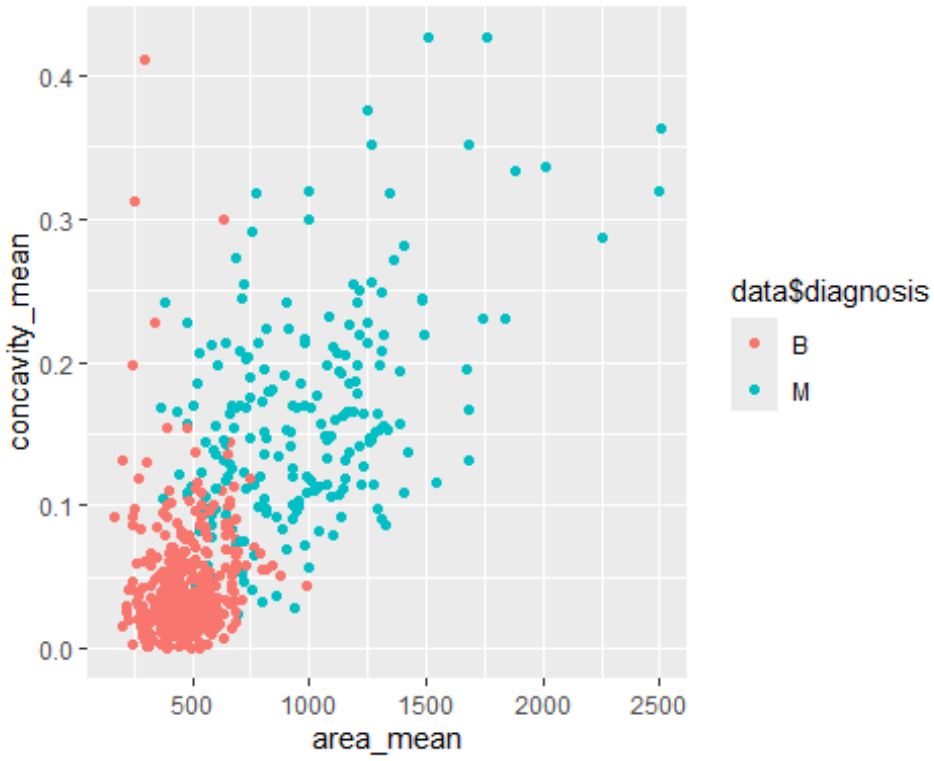

concavity_mean
0.4
0.3
0.2
0.1
0.0
10
15
20
25
radius_mean
data$diagnosis
B
M


2500
2000
1500
1000
500
area_mean
80
120
160
perimeter_mean
data$diagnosis
B
M


0.3
0.2
0.1
compactness_mean
80
120
160
perimeter_mean
data$diagnosis
B
M


concavity_mean
0.4
0.3
0.2
0.1
0.0
80
120
160
perimeter_mean
data$diagnosis
B
M


compactness_mean
0.3
0.2
0.1
500
1000
1500
2000
2500
area_mean
data$diagnosis
B
M



- **Variables Corrélées:**
  - On peut remarquer grâce au nuage de points entre radius_mean et perimeter_mean que les variables périmètre et rayon semblent corrélées car

les points semblent former une droite . Ce qui n'est pas surprenant étant donné qu'en géométrie le $périmètre = 2 * \pi * rayon$.

- Pour le nuage de points entre radius_mean et area_mean les variables ont l'air d'être corrélées car l'ensemble des points forment une courbe semblable à celle d'une fonction exponentielle.
- Le nuage de points entre perimeter_mean et et area_mean laisse présager une courbe semblable à celle de la fonction exponentielle. Donc on peut en déduire qu'elle pourrait être corrélée.
- Le nuage de points entre compactness_mean et concavity_mean est assez complexe car on remarque une dispersion des points assez importante mais on observe tout de même une sorte de courbe s'apparentant très légèrement à une fonction exponentielle.

- **Variables Non Corrélées:**

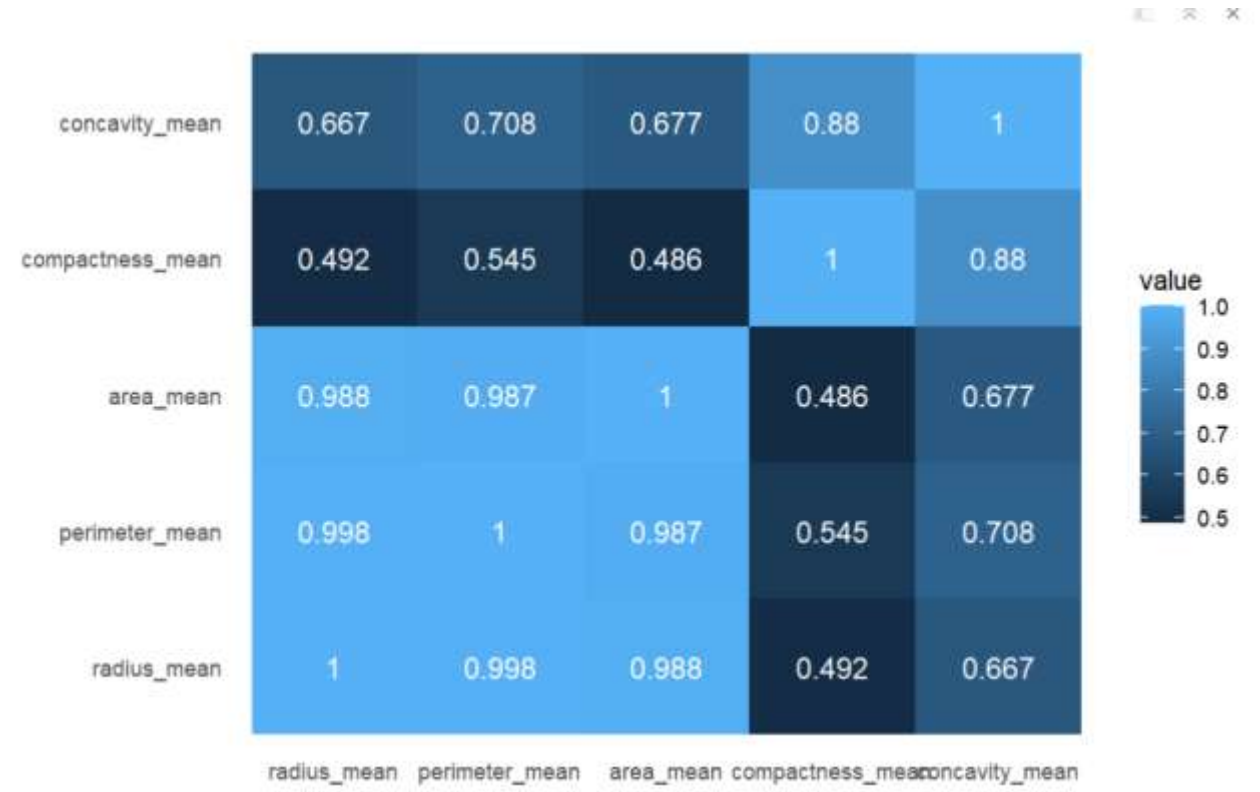
  Pour les variables non corrélées nous avons répertorié les nuages de points entre ces différentes variables:

  - radius_mean et compactness_mean
  - radius_mean et concavity_mean
  - perimeter_mean et compactness_mean
  - perimeter_mean et concavity_mean
  - area_mean et compactness_mean
  - area_mean et concavity_mean

2. Calcule de corrélation

```
library(reshape2)

df = data[, col]
correlation = round(cor(df), 3)

correlation <- melt(correlation)

fig = ggplot(data = correlation, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  #scale_fill_gradient2(low = 'blue', mid = 'white', high = 'red') +
  geom_text(aes(Var2, Var1, label = value), color = "white", size = 4) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
```

```
    panel.background = element_blank(),
    axis.ticks = element_blank())

fig
```

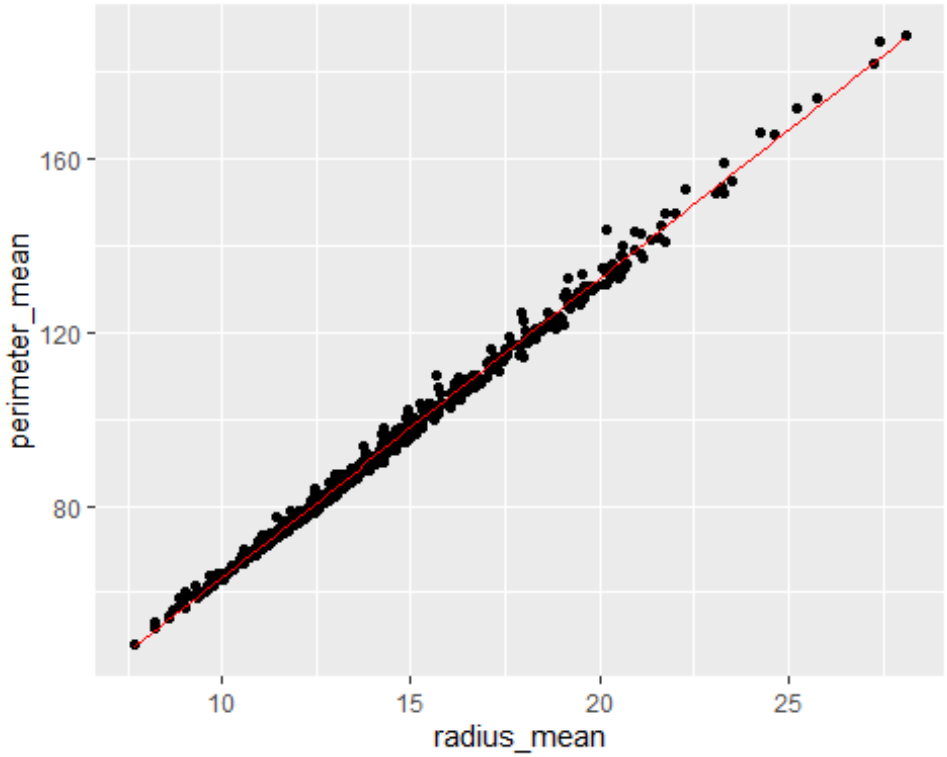


- Les variables area_mean, perimeter_mean et radius_mean sont très corrélées entres elles avec un coefficient corrélation de Pearson très proche de 1 (0.987 et 0.998).
- perimeter_mean et area_mean ont pour coefficient de corrélation 0.987 ce qui est très proche de 1 et témoigne d'un possible corrélation positive entre ces 2 variables
- la variable concavity_mean et la variable radius_mean ont un coefficient de corrélation de 0.677, on peut en déduire que la corrélation entre ces 2 variables est plus ou moins présente mais loin du niveau des variables précédentes
- compactness_mean et radius_mean ont pour coefficient de corrélation 0.506 ce qui veut dire que les variables semblent légèrement corrélés de façons positives mais elles restent inférieures au niveau de corrélation attendue. il s'agit de la variable la moins corrélées avec le rayon
- perimeter_mean et compactness ont 0.557 comme coefficient de corrélation, cette valeur rend compte d'une très légère corrélation avec perimeter_mean mais elle reste de loin la moins corrélés avec cette variable

- perimeter_mean et concavity_mean ont un coefficient de corrélation de 0.716 cette valeur peut être considérer comme élevé mais en raison du niveau de précision que nous souhaitons avoir nous dirons que ces variables ne sont pas fortement corrélés
- area_mean et compactnes_mean ont un coefficient de corrélation égale à 0.499 ce qui ne représente pas un haut degré de corrélation en vu de nos exigences
- Pour area_mean et concavity_mean le coefficient de corrélation est 0.686 cela ne répond pas à notre niveau d'exigence pour affirmer la corrélation entre 2 variables
- Enfin compactness_mean et concavity_mean sont des variables avec un coefficient de corrélation de 0.883 ce qui est relativement proche de 1 et peut donc témoigner d'une certaine corrélation positive
- Étant donné que nous souhaitons trouver un modèle efficace pour identifier le type de tumeur, sans perdre de l'information en laissant tombé certaine variable, nous exigeons un important niveau de corrélation dont nous plaçons le seuil à 0.8 ce qui est relativement proche de 1 et laisse transparaitre une assez forte corrélation.

A. Regression lineaire entre perimeter_mean et radius_mean

```
reg = summary(lm(perimeter_mean ~ radius_mean, data))
a = reg$coefficients['radius_mean', 'Estimate']
b = reg$coefficients['(Intercept)', 'Estimate']
r2 = reg$r.squared

func = function(x){
  a*x + b
}

fig = ggplot(data, aes(x = radius_mean, y = perimeter_mean )) +
  geom_point(aes()) +
  labs(x = "radius_mean", y = "perimeter_mean") +
  geom_function(fun = func, color = 'red')
print(fig)
```




```
print(paste("R² = ", r2))
```

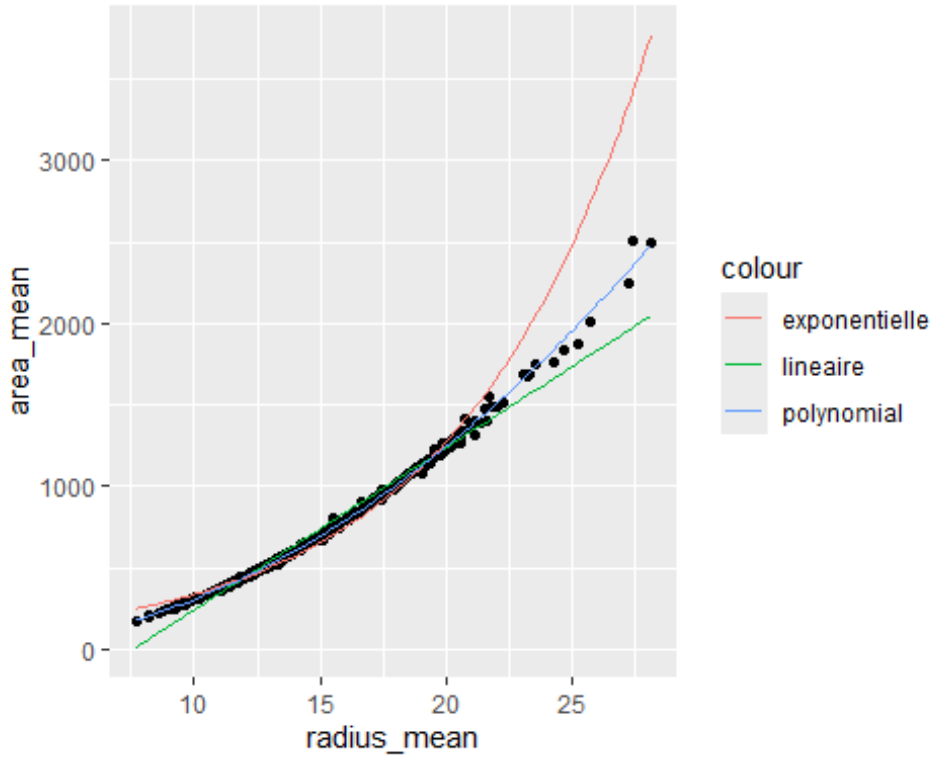
```
## [1] "R² =  0.995533361489872"
```

Ce graphique nous présente perimeter_mean en fonction de radius_mean. Comme dit precedament, les données semble formé une droite(corrélation lineaire). pour verifier cela, nous avons calculer le coefficient de corrélation et nous avons obtenu 99%. puis nous avons fait une regression linaire. ce modele de regression a un R^2 de 99,55%, ce qui signifie que notre modèle explique 99,55% de la variabilité, ce qui est très bien. On peu donc a partir de ce modèle estimé les valeurs de perimeter_mean en ayant radius_mean

B. Regression entre area_mean et radius_mean

```
# lineaire
reg = summary(lm(area_mean ~ radius_mean, data))
a_l = reg$coefficients['radius_mean', 'Estimate']
b_l = reg$coefficients['(Intercept)', 'Estimate']
r2_l = reg$r.squared

func_l = function(x){
  a_l*x + b_l
}

reg = summary(lm(log(area_mean) ~ radius_mean, data))
a_e = reg$coefficients['radius_mean', 'Estimate']
b_e = reg$coefficients['(Intercept)', 'Estimate']
r2_e = reg$r.squared
```

```
func_e = function(x){
  exp(b_e) * exp(a_e*x)
}

reg = summary(lm(log(area_mean) ~ log(radius_mean), data))
a_p = reg$coefficients['log(radius_mean)', 'Estimate']
b_p = reg$coefficients['(Intercept)', 'Estimate']
r2_p = reg$r.squared


func_p = function(x){
  exp(b_p) * x^a_p
}

fig = ggplot(data, aes(x = radius_mean, y = area_mean )) +
  geom_point() +
  labs(x = "radius_mean", y = "area_mean") +
  geom_function(fun = func_l, aes(colour = 'lineaire')) +
  geom_function(fun = func_e, aes(colour = 'exponentielle')) +
  geom_function(fun = func_p, aes(colour = 'polynomial'))

print(fig)
```




```
print(paste("R²_lineaire = ", r2_l))
```

```
## [1] "R²_lineaire =  0.976309576260783"
```

```
print(paste("R²_exponentiel = ", r2_e))
```

```
## [1] "R²_exponentiel =  0.975528744978367"
```

```
print(paste("R²_polynomial = ", r2_p))
```

```
## [1] "R²_polynomial =  0.99908987533294"
```

La première était avec la fonction exponentielle et nous constatons que ce modèle n'est pas très adapté car sur certaines parties la courbe n'est pas alignée avec le nuage de points

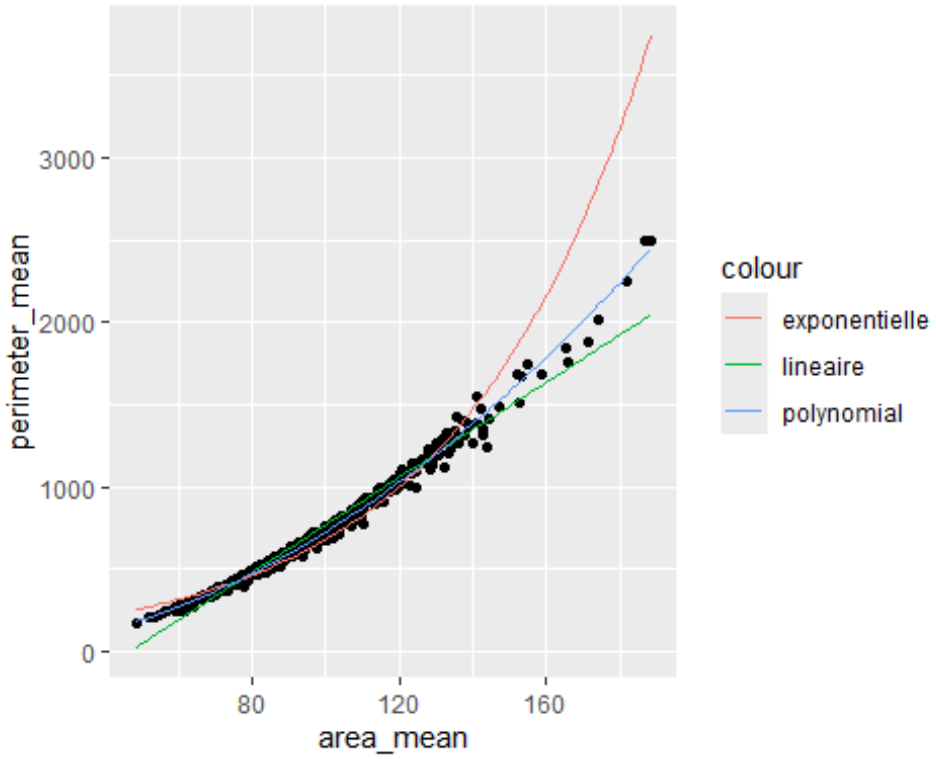
pour le modéle de droite linéaire il peut paraitre efficace au début mais vers la fin de la courbe il n'est plus aligné avec le nuage de points

Pour ce graphique le meilleur des 3 modèles est le modèle polynomial car la courbe suit parfaitement le nuage de point et le coefficient de détermination R^2 prend en explique 99.9% de la variationtandis que les 2 autres modèles sont moins adaptés en prenant pour l'exponentielle 97.5% et pour le linéaire 97.6%.

C. Regression entre area_mean et perimeter_mean

```
# lineaire
reg = summary(lm(area_mean ~ perimeter_mean, data))
a_l = reg$coefficients['perimeter_mean', 'Estimate']
b_l = reg$coefficients['(Intercept)', 'Estimate']
r2_l = reg$r.squared


func_l2 = function(x){
  a_l*x + b_l
}

reg = summary(lm(log(area_mean) ~ perimeter_mean, data))
a_e = reg$coefficients['perimeter_mean', 'Estimate']
b_e = reg$coefficients['(Intercept)', 'Estimate']
r2_e = reg$r.squared

func_e = function(x){
  exp(b_e) * exp(a_e*x)
}

reg = summary(lm(log(area_mean) ~ log(perimeter_mean), data))
a_p = reg$coefficients['log(perimeter_mean)', 'Estimate']
b_p = reg$coefficients['(Intercept)', 'Estimate']
r2_p = reg$r.squared


func_p = function(x){
  exp(b_p) * x^a_p
}
```

```r
fig = ggplot(data, aes(x = perimeter_mean, y = area_mean )) +
  geom_point() +
  labs(x = "area_mean", y = "perimeter_mean") +
  geom_function(fun = func_l, aes(colour = 'lineaire')) +
  geom_function(fun = func_e, aes(colour = 'exponentielle')) +
  geom_function(fun = func_p, aes(colour = 'polynomial'))

print(fig)
```




```r
print(paste("R²_lineaire = ", r2_l))
```

```
## [1] "R²_lineaire =  0.974539606999101"
```

```r
print(paste("R²_exponentiel = ", r2_e))
```

```
## [1] "R²_exponentiel =  0.966866617567231"
```

```r
print(paste("R²_exponentiel = ", r2_p))
```

```
## [1] "R²_exponentiel =  0.994241172071813"
```

Pour ce graphique nous avons abordé 3 approches différentes d'ajustement linéaires avec la méthode des moindres carrés.

La première était avec la fonction exponentielle et nous constatons que ce modèle n'est pas adapté car sur certaines parties la courbe n'est pas alignée avec le nuage de points

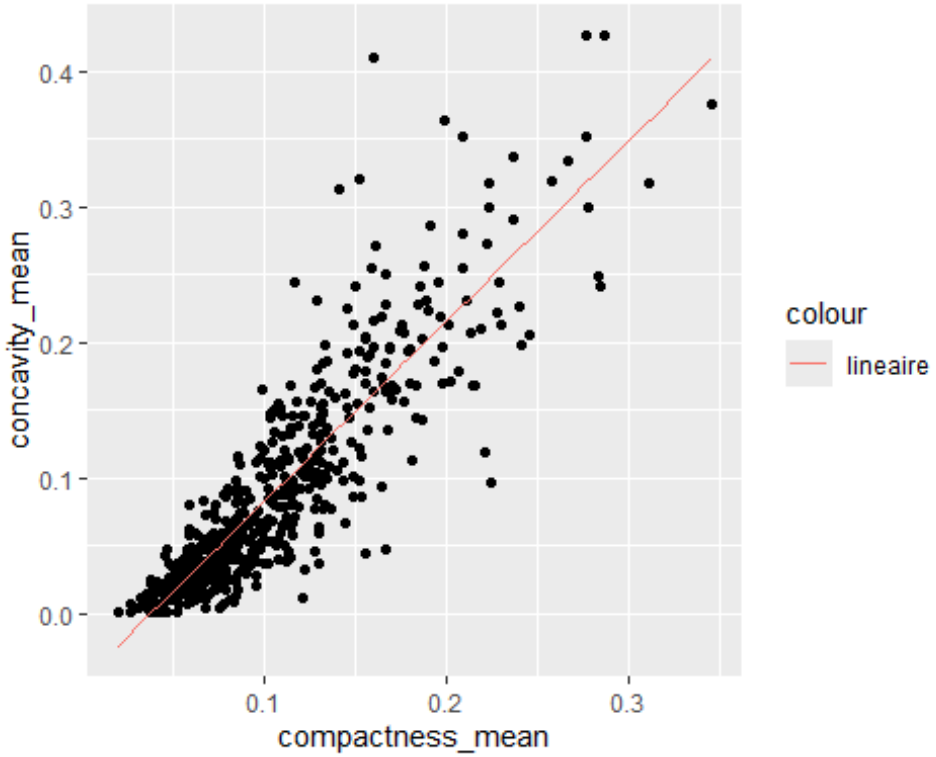
pour le modéle de droite linéaire il peut paraitre efficace au début mais vers la fin de la courbe il n'est plus aligné avec le nuage de points

et enfin le modéle polynomial est celui qui correspond le mieux car la courbe est aligné avec l'ensemble du nuage de point et le R^2 permet d'expliquer plus de 99% de la variable.

Pour ce graphique le meilleur des 3 modèles est le modèle polynomial car la courbe suit parfaitement le nuage de point et le coefficient de détermination R^2 prend en compte 99.4% de la variable tandis que les 2 autres modèles sont moins adaptés en prenant pour l'exponentielle 96.7% et pour le linéaire 97.5%

D. régression entre concavity_mean et compactness_mean

```
# lineaire
reg = summary(lm(concavity_mean ~ compactness_mean, data))
a_l = reg$coefficients['compactness_mean', 'Estimate']
b_l = reg$coefficients['(Intercept)', 'Estimate']
r2_l = reg$r.squared


func_l2 = function(x){
  a_l*x + b_l
}

fig = ggplot(data, aes(x = compactness_mean, y = concavity_mean )) +
  geom_point() +
  labs(x = "compactness_mean", y = "concavity_mean") +
  geom_function(fun = func_l, aes(colour = 'lineaire'))

print(fig)
```




```
print(paste("a_lineaire = ", a_l))
```

```
## [1] "a_lineaire =  1.32916529794442"
```

```
print(paste("b_lineaire = ", b_l))
```

```
## [1] "b_lineaire =  -0.0495890281818206"
```

```
print(paste("R²_lineaire = ", r2_l))
```

```
## [1] "R²_lineaire =  0.775268674250357"
```

Pour ce graphique nous pouvons voir que la courbe du modèle linéaire ne permet pas de bien expliqué la variation entre compactness et concavity

Le nuage de point ne suit pas la droite

D'autant plus que le coefficient de determination R^2 est de 77.5% ce qui ne permet pas de définir clairement le lien entre les variables

On choisira donc de garder ces 2 variables séparément

3. Conclusion

Après toutes ces analyses, nous avons vu que certaine variable n'etait pas discriminante, nous les avons donc laisser tombé en cours de route. Pour améliorer la précision de notre modèle, nous avons décidé d'examiner les différente correlation entre les variables restantes (radius, area, perimeter, compactness, concavity).

L'analyse des corréation entre ces variable nous on montré que les variables (area, radius, et perimeter) était tres bien corrélé entre elle, nous avons ainsi touver des modèle qui expliquait plus de 99% de la variabilité, grace a ces modèles, nous somme capable de retouver les valeurs des variable grace a radius_mean avec une assez grande precision. Nous avons donc décidé de garder la variable **radius_mean**.

Consernant compactness et concavity, il semblait avoir une corrélation lineaire apres le calcul du coefficient de pearson (88%). Nous avons donc chercher a trouver un modèle obtimal qui permetrait de predire les valeurs de l'un en fonction de l'autre. Sur le nuage de point, nous voyons bien qu'il ya une sorte de droite qui se desine, mais les valeur sont trop dispersé. en cherchant un modèle lineaire, qui semblais le plus approprier, nous avons trouver un modèle avec un $R^2 = 77.5\%$, ce qui n'est pas très convaincant, nous avons donc décidé de garder les deux variables pour le modèle final.

les variables conserver avec leurs seuil sont :

- radius mean : 15
- concavity_mean : 0.1
- compactness_mean : 0.12

le modèle final que nous proposons est le suivant :

$$score = \frac{2}{5} concavity\_mean + \frac{2}{5} radius\_mean + \frac{1}{5} compactness\_mean$$

- Si $score < seuil$, alors le tumeur est bénine
- Si $score > seuil$ alors la tumeurs est maline

nb : les valeurs doivent etre normalisé en appliquant la formule suivante : $X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}}$

```
#calcule du seuil
#le seuil € [0;1]
radius_mean = (15 - min(data$radius_mean)) / (max(data$radius_mean) -
min(data$radius_mean))

concavity_mean = (0.1 - min(data$concavity_mean)) / (max(data$concavity_mean)
- min(data$concavity_mean))

compactness_mean = (0.12 - min(data$compactness_mean)) /
(max(data$compactness_mean) - min(data$compactness_mean))


seuil = 0.4 * radius_mean + 0.4 * concavity_mean + 0.2 * compactness_mean

# calcule de la précision de notre modele
```

```r
# recuperation des variable discriminante
data_normal = data[c('radius_mean','concavity_mean', 'compactness_mean')]

# normalisation des données
for(i in colnames(data_normal)){
  data_normal[, i] = normalise(data[, i])
}

# estimation si une tumeur est maline ou benine
data_normal['Estimation'] = ifelse((0.4 * data_normal[, 'radius_mean'] +
                            0.4 * data_normal[, 'concavity_mean'] +
                            0.2 * data_normal[, 'compactness_mean']) >= seuil,
"M", "B")

# recuperation de l'etat reel des tumeurs
data_normal['diagnosis'] = data['diagnosis']

# calcul du score de notre modèle
score = sum(data_normal$Estimation == data_normal$diagnosis) /
dim(data_normal)[1]

print(paste("le score de notre modele est de ", round(score*100, digits = 2),
"%"))
```

```
## [1] "le score de notre modele est de  90.47 %"
```

On voit bien que ce modèle est meilleur que le précedant

- voici la fonction qui vous dira si une tumeur est benine ou maline

```
type_tumeur = function(rayon, concavite, compacite){
  radius_mean = (15 - min(data$radius_mean)) / (max(data$radius_mean) -
min(data$radius_mean))

  concavity_mean = (0.1 - min(data$concavity_mean)) /
(max(data$concavity_mean) - min(data$concavity_mean))

  compactness_mean = (0.12 - min(data$compactness_mean)) /
(max(data$compactness_mean) - min(data$compactness_mean))

  rayon = (rayon - min(data$radius_mean)) / (max(data$radius_mean) -
min(data$radius_mean))

  concavite = (concavite - min(data$radius_mean)) /
(max(data$radius_mean) - min(data$radius_mean))

  compacite = (compacite - min(data$radius_mean)) /
(max(data$radius_mean) - min(data$radius_mean))


  seuil = 0.4 * radius_mean + 0.4 * concavity_mean + 0.2 *
compactness_mean

  score = 0.4 * rayon + 0.4 * concavite + 0.2 * compacite

  if(score > seuil){
    return("la tumeur est maline")
  } else {
    return("la tumeur est benine")
  }

}
```