

Projet1 analyse de donnée - Shuaibo Huang & Théophane Guiffant

Shuaibo Huang & Théophane Guiffant

2024-11-15

Contents

I. Choix de données	1
II. Description des données: Metadata	3
III. ANALYSE SUR R	5

I. Choix de données

0. Installation des packages.

```
available_packages <- installed.packages()[,1] # what's available ?

if (! 'pacman' %in% available_packages){
  install.packages("pacman")           # install if needed
}

pacman::p_load(tidyverse)    # meta-package
pacman::p_load(questionr)
pacman::p_load(ggstats)
pacman::p_load(BioStatR)
rm(available_packages)      # Ockham's razor

old_theme <- theme_set(theme_minimal(base_family = "Helvetica Neue"))

# permet de lire le fichier excel .xlsx
if (!require("readxl")) install.packages("readxl")

# Installer le package nortest si nécessaire
if (!require(nortest)) install.packages("nortest")

# Load necessary libraries
library(nortest)
library(readxl)
library(dplyr)
library(ggplot2)
library(forcats)
library(BioStatR)
library(readxl)
```

1. importation des données.

```
# Load the dataset  
bike_data <- read_excel("london_merged.xlsx")
```

2. Transformation des données.

```
#on transforme les colonnes season et weather_code pour plus de simplicité.  
bike_data <- bike_data %>%  
  mutate(  
    season = factor(season, labels = c("Spring", "Summer", "Fall", "Winter")),  
    weather_code = factor(weather_code,  
      levels = c(1, 2, 3, 4, 7, 10, 26, 94),  
      labels = c("Clear", "Few Clouds", "Broken Clouds",  
                "Cloudy", "Rain", "Thunderstorm",  
                "Snowfall", "Freezing Fog"))  
)
```

II. Description des données: Metadata

"cnt" - the count of a new bike shares

"t1" - real temperature in C

"t2" - temperature in C "feels like"

"hum" - humidity in percentage

"wind_speed" - wind speed in km/h

"weather_code" - category of the weather

"is_holiday" - booleanfield - 1 holiday / 0 non holiday

"is_weekend" - boolean field - 1if the day is weekend

"season" -category field meteorological seasons: 0-spring ; 1-summer; 2-fall; 3-winter. (cela a été transfor

"weather_code" category description:

1 = Clear ; mostly clear but have some values with haze/fog/patches of fog/ fog in vicinity

2 = scattered clouds / few clouds

3 = Broken clouds

4 = Cloudy

7 = Rain/ light Rain shower/ Light rain

10 = rain with thunderstorm

26 = snowfall

94 = Freezing Fog

(cela a été transformé dans le I)

```
# Inspect the data
colSums(is.na(bike_data)) # Check de la non presence de valeur na
```

```
##      timestamp          cnt          t1          t2          hum    wind_speed
##      0                  0                  0                  0                  0                  0
## weather_code  is_holiday  is_weekend      season
##      0                  0                  0                  0

summary(bike_data)

##      timestamp          cnt          t1
## Min.   :2015-01-04 00:00:00.00  Min.   : 0  Min.   :-1.50
## 1st Qu.:2015-07-04 20:15:00.00  1st Qu.: 257  1st Qu.: 8.00
## Median :2016-01-03 15:30:00.00  Median : 844  Median :12.50
## Mean   :2016-01-03 22:31:00.56  Mean   :1143  Mean   :12.47
## 3rd Qu.:2016-07-04 15:45:00.00  3rd Qu.:1672  3rd Qu.:16.00
## Max.   :2017-01-03 23:00:00.00  Max.   :7860  Max.   :34.00
##
##      t2          hum    wind_speed      weather_code
## Min.   :-6.00  Min.   : 20.50  Min.   : 0.00  Clear       :6150
## 1st Qu.: 6.00  1st Qu.: 63.00  1st Qu.:10.00  Few Clouds  :4034
## Median :12.50  Median : 74.50  Median :15.00  Broken Clouds:3551
## Mean   :11.52  Mean   : 72.32  Mean   :15.91  Rain        :2141
## 3rd Qu.:16.00  3rd Qu.: 83.00  3rd Qu.:20.50  Cloudy      :1464
## Max.   :34.00  Max.   :100.00  Max.   :56.50  Snowfall    : 60
##                                         (Other)     : 14
##
##      is_holiday  is_weekend      season
## Min.   :0.00000  Min.   :0.0000  Spring:4394
## 1st Qu.:0.00000  1st Qu.:0.0000  Summer:4387
## Median :0.00000  Median :0.0000  Fall   :4303
## Mean   :0.02205  Mean   :0.2854  Winter:4330
## 3rd Qu.:0.00000  3rd Qu.:1.0000
## Max.   :1.00000  Max.   :1.0000
##
```

Ce jeu de donnée décrit les emprunts de vélib à Londre heure par heure de 2015 à 2017. Pour offrir un service plus qualitatif, il serait intéressant de pouvoir prédire le nombre d'emprunt selon certaines des variables donnée. Pour cela nous analyserons d'abord certaines variables et leur lien avant de répondre à la question: Peut-on créer un modèle à partir des variables données pour prédire l'emprunt de vélo(cnt)?

III. ANALYSE SUR R

1. Analyse d'une variable qualitative (weather_code).

Déterminons tout d'abord la table statistique du code météo.

```
weather_dist<-bike_data %>% group_by(weather_code) %>%  
  count() %>%  
  ungroup() %>%  
  mutate(Pourcentage=100*n/sum(n)) %>%  
  arrange(desc(Pourcentage))  
  
weather_dist  
  
## # A tibble: 7 x 3  
##   weather_code     n Pourcentage  
##   <fct>       <int>     <dbl>  
## 1 Clear         6150     35.3  
## 2 Few Clouds    4034     23.2  
## 3 Broken Clouds 3551     20.4  
## 4 Rain          2141     12.3  
## 5 Cloudy        1464      8.41  
## 6 Snowfall       60      0.345  
## 7 Thunderstorm   14      0.0804
```

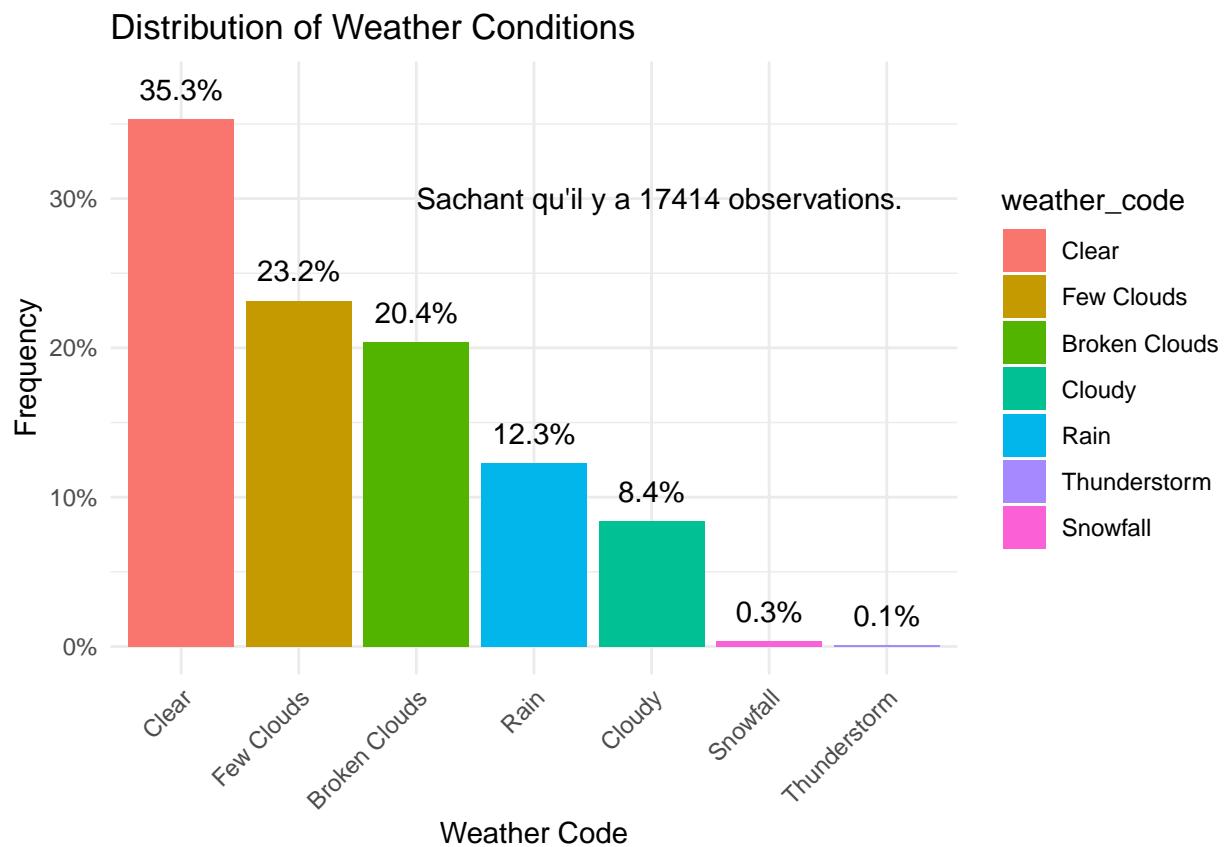
Nous observons l'absence de la ligne “Freezing Fog”, vérifions qu'il n'y a jamais eu ce code météo à Londres de 2015 à 2017.

```
bike_data %>% filter(weather_code == "Freezing Fog")  
  
## # A tibble: 0 x 10  
## # i 10 variables: timestamp <dttm>, cnt <dbl>, t1 <dbl>, t2 <dbl>, hum <dbl>,  
## #   wind_speed <dbl>, weather_code <fct>, is_holiday <dbl>, is_weekend <dbl>,  
## #   season <fct>
```

Maintenant, pour mieux visualiser la distribution des codes météo, représentons graphiquement celle-ci à l'aide d'un diagramme en barre.

```
#Nombre total d'observations  
N<-(weather_dist%>%  
  mutate(N=sum(n))%>%  
  select(N)%>%  
  distinct()  
 )$N
```

```
#resume graphique
ggplot(bike_data, aes(x=fct_infreq(weather_code), y=after_stat(prop), fill = weather_code)) +
  geom_bar(stat="prop") +
  geom_text(stat="prop", nudge_y=0.02) +
  labs(title="Distribution of Weather Conditions",
       x = "Weather Code",
       y = "Frequency") +
  scale_y_continuous(labels=scales::percent) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  annotate(
    "text",
    x = 3, y = 0.3, # Position de l'annotation, ajustez selon votre graphique
    label = paste("Sachant qu'il y a", N, "observations."),
    size = 4, color = "Black", hjust = 0
  )
```



On voit ici que certains code météo sont quasi-inexistant (snowfall et thunderstorm) voire complètement inexistant (freezing fog). Cela montre une distribution assez asymétrique pour les code météo extrême mais assez bien répartie pour les code météo plus modéré.

2. Analyse d'une variable quantitative (cnt).

Décrivons statistiquement en premier cnt.

```
summary(bike_data$cnt)

##      Min.   1st Qu.    Median     Mean   3rd Qu.   Max.
##        0       257      844     1143     1672    7860

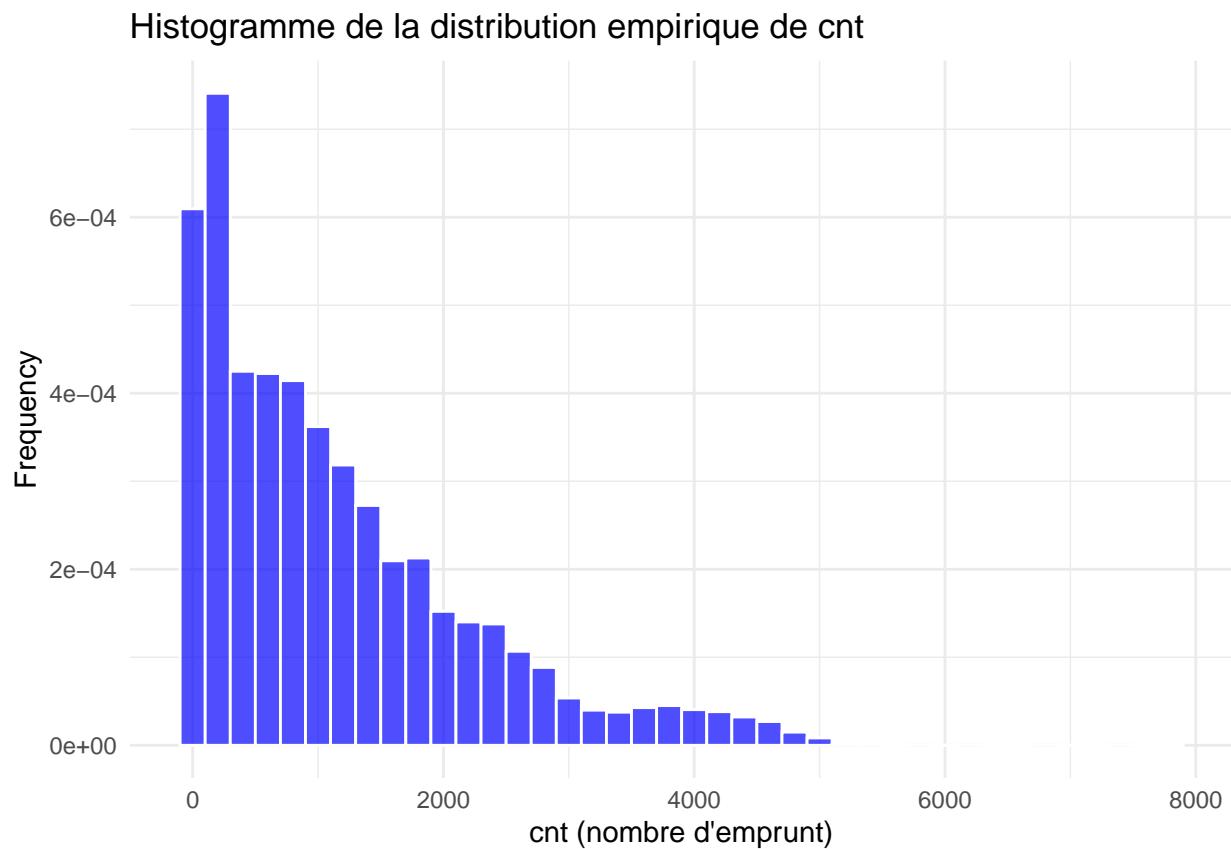
mean_data <- mean(bike_data$cnt)
var_data <- var(bike_data$cnt)
ecart_data <- sd(bike_data$cnt)
cat("Moyenne:", mean_data, "\nVariance:", var_data, "\nEcart-type:", ecart_data)

## Moyenne: 1143.102
## Variance: 1177460
## Ecart-type: 1085.108
```

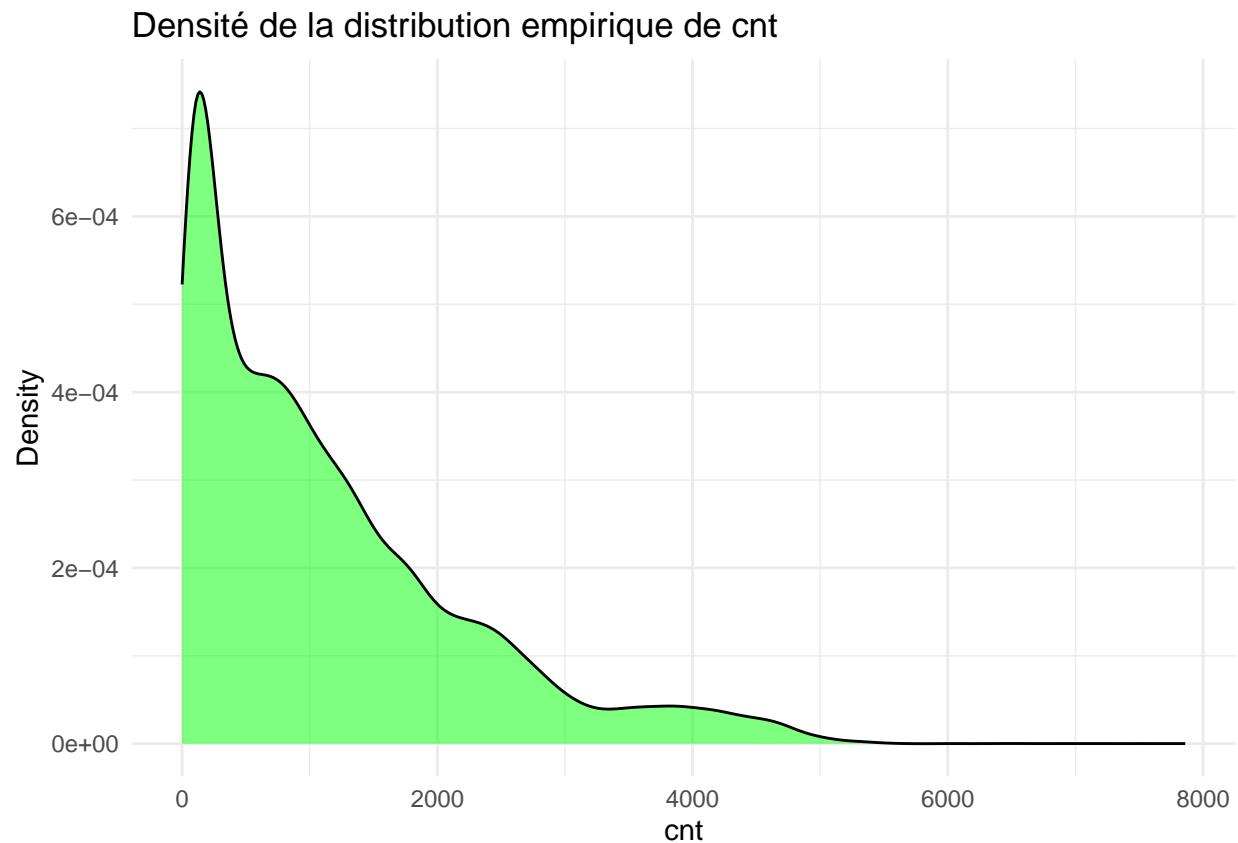
Ces résultats suggèrent une grande variabilité du nombre d'emprunts. En effet, la variance aussi élevée est le signe de périodes avec peu d'emprunts (comme tôt le matin) et des périodes avec beaucoup d'emprunt (comme les heures de pointes).

Pour mieux visualiser cela, Représentons graphiquement la distribution empirique de cnt.

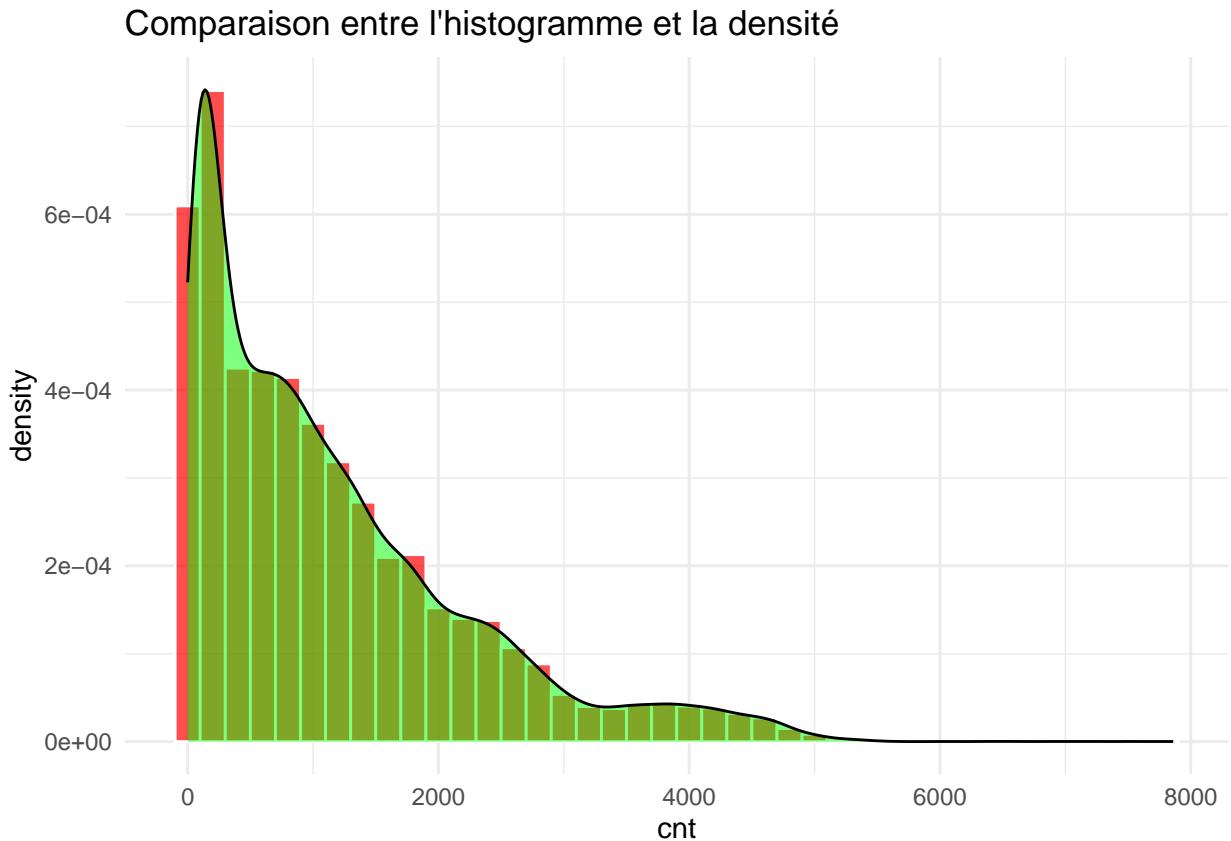
```
# Histogram of `cnt`
ggplot(bike_data, aes(x = cnt)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 200, fill = "blue", color = "white", alpha = 0.8)
  labs(title = "Histogramme de la distribution empirique de cnt", x = "cnt (nombre d'emprunt)", y = "Fréquence")
  theme_minimal()
```



```
# Density plot of `cnt`  
ggplot(bike_data, aes(x = cnt)) +  
  geom_density(fill = "green", alpha = 0.5) +  
  labs(title = "Densité de la distribution empirique de cnt", x = "cnt", y = "Density") +  
  theme_minimal()
```



```
ggplot(bike_data, aes(x = cnt)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 200, fill = "red", color = "white", alpha = 0.5) +
  geom_density(fill = "green", alpha = 0.5) +
  labs(title = "Comparaison entre l'histogramme et la densité", x = "cnt") +
  theme_minimal()
```



On voit une distribution asymétrique selon la moyenne avec une queue à droite indiquant quelques périodes avec un très grand nombre d'emprunts. C'est cohérent avec la forte variance trouvée précédemment.

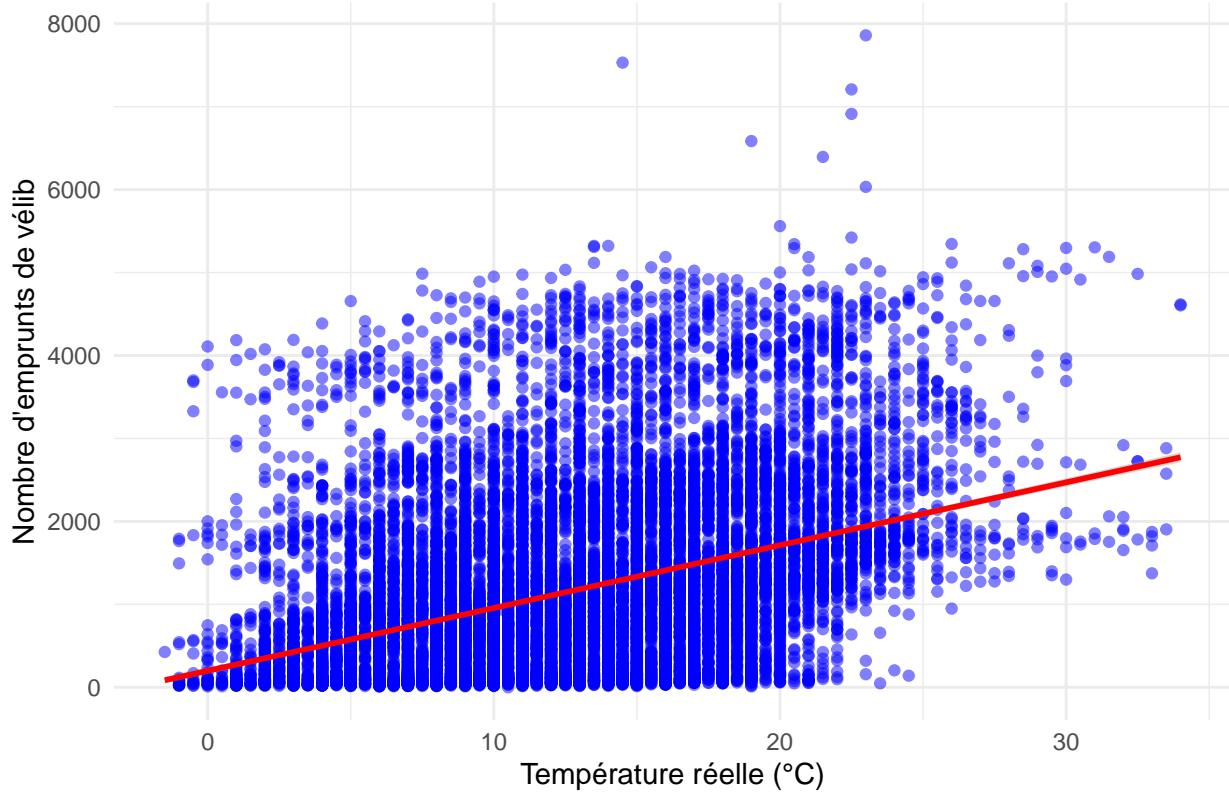
3. Analyse d'un lien entre deux variables quantitatives (cnt et t1).

Commençons par visualiser la relation entre t1 et cnt.

```
# Scatter plot: t1 (température) vs. cnt (comptes de vélos)
ggplot(bike_data, aes(x = t1, y = cnt)) +
  geom_point(alpha = 0.5, color = "blue") +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  labs(
    title = "Relation entre la température réelle et les emprunts de vélos",
    x = "Température réelle (°C)",
    y = "Nombre d'emprunts de vélib"
  ) +
  theme_minimal()
```

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

Relation entre la température réelle et les emprunts de vélos



On voit une augmentation du nombre d'emprunt selon la température mais ce n'est pas très claire. Faisons des tests pour confirmer le lien entre t1 et cnt.

Calculons le coefficient de corrélation entre t1 et cnt avec le test de Pearson.

```
## Test de corrélation de Pearson
cor_test <- cor.test(bike_data$t1, bike_data$cnt)

## Affichage du résultat
cor_test

##
##  Pearson's product-moment correlation
##
## data: bike_data$t1 and bike_data$cnt
## t = 55.685, df = 17412, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3761178 0.4013335
## sample estimates:
##        cor
## 0.3887985
```

Le p est très faible donc t1 et cnt ne sont pas indépendante. De plus, elles ont un coefficient de corrélation positif et pas trop élevé. C'est logique puisque l'on a pas de température extrême et plus il fait beau/chaud plus il y a d'emprunts de vélib.

4. Analyse d'un lien entre deux variables qualitative (season et weather_code).

Créons la table de contingence des variables season et weather_code.

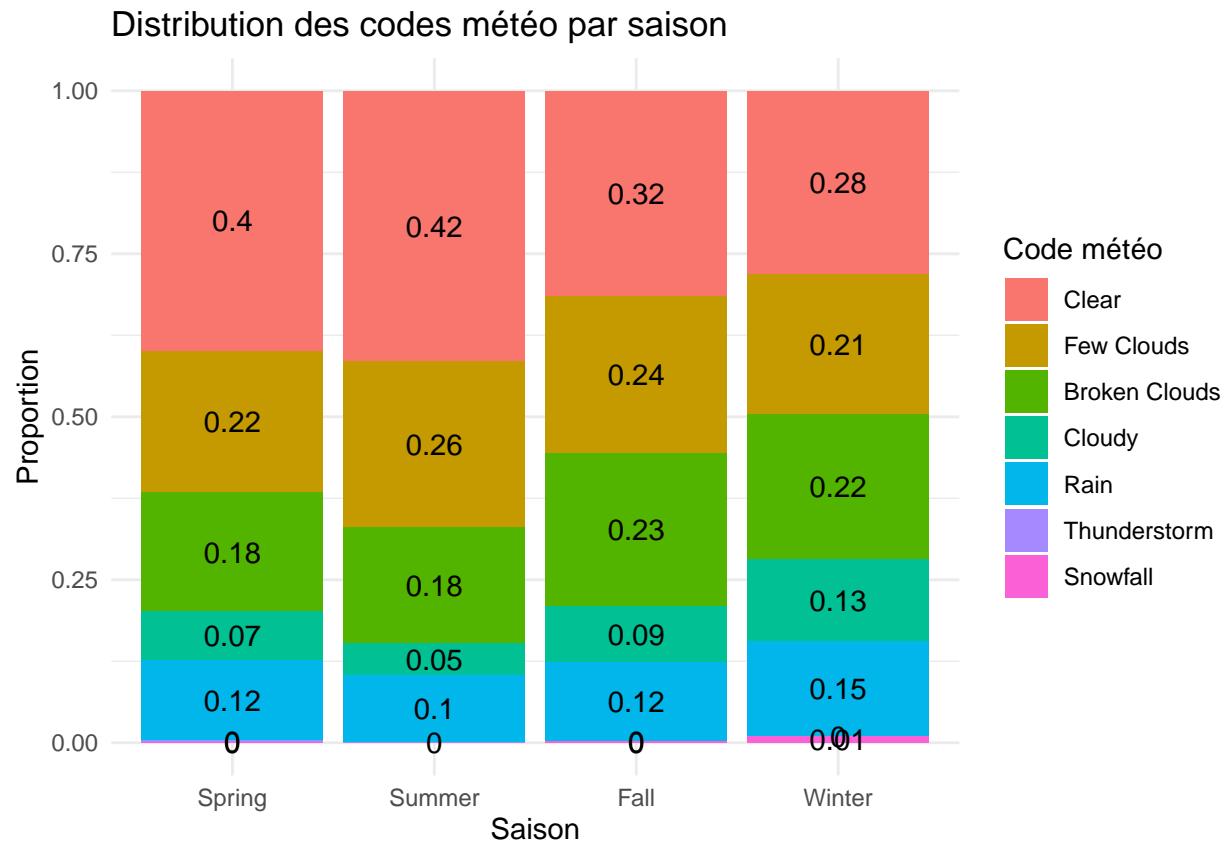
```
data1<-bike_data %>% group_by(weather_code,season) %>%
  count() %>%
  ungroup() %>%
  select(season,weather_code,n) %>%
  distinct() %>%
  group_by(season) %>%
  mutate(percent=n/sum(n))

data2<-pivot_wider(data1,
                     id_cols=season,
                     names_from=weather_code,
                     values_from=n
                    )
data2

## # A tibble: 4 x 8
## # Groups:   season [4]
##   season Clear `Few Clouds` `Broken Clouds` Cloudy Rain Thunderstorm Snowfall
##   <fct>    <int>      <int>          <int>   <int> <int>        <int>   <int>
## 1 Spring     1752       955          804    326   544         7      6
## 2 Summer     1822      1119          775    216   450         5     NA
## 3 Fall       1356      1034         1011    372   519         1     10
## 4 Winter     1220       926          961    550   628         1     44
```

Pour mieux visualiser ces résultats faisons un diagramme en barre illustrant la distribution du code météo sachant la saison.

```
# Graphique en barres empilées
ggplot(data1, aes(x = season, y=percent, fill = weather_code)) +
  geom_bar(stat='identity',position="stack") +
  geom_text(aes(label=round(percent,2)),position = position_stack(vjust = 0.5))+ 
  labs(
    title = "Distribution des codes météo par saison",
    x = "Saison",
    y = "Proportion",
    fill = "Code météo"
) +
  theme_minimal()
```



On observe un lien entre les saisons et le code météo. En effet, en été et au printemps, il fait plus beau alors qu'en hivers et en automne il y a plus de nuages.

Faisons un test du chi-2 pour tester l'indépendance des variables weather_code et saison.

```
data2[is.na(data2)] <- 0 #on remplace NA par 0

contingence_table <- as.matrix(data2[, -1]) # Enlever la colonne 'season'
rownames(contingence_table) <- data2$season

# Test du Chi-2 sur la table de contingence
chi2_test <- chisq.test(contingence_table)

## Warning in chisq.test(contingence_table): L'approximation du Chi-2 est
## peut-être incorrecte

# Résultat du test
print(chi2_test)

##
## Pearson's Chi-squared test
##
## data: contingence_table
## X-squared = 510.55, df = 18, p-value < 2.2e-16
```

Il semble que weather_code et season sont lié car le p est très faible. Cependant, plusieurs weather_code ont des fréquences trop faibles.

On va donc utiliser le test de Fisher pour vérifier ce résultat.

```
fisher.test(contingence_table, simulate.p.value = TRUE)

##
## Fisher's Exact Test for Count Data with simulated p-value (based on
## 2000 replicates)
##
## data: contingence_table
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

p est de valeur suffisamment faible pour conclure que les saisons ont un impact sur les types de conditions météorologique.

5. Un lien entre une variable quantitative (cnt) et une variable qualitative (weather_code).

```
weather_dist<-bike_data %>% group_by(weather_code) %>%
  summarize(nb_emprunt=sum(cnt),n=n())%>%
  ungroup() %>%
  mutate(frequence_apparition=100*n/sum(n)) %>%
  mutate(Pourcentage_emprunt=100*nb_emprunt/sum(nb_emprunt)) %>%
  arrange(desc(frequence_apparition))

weather_dist

## # A tibble: 7 x 5
##   weather_code nb_emprunt     n frequence_apparition Pourcentage_emprunt
##   <fct>          <dbl> <int>                <dbl>                  <dbl>
## 1 Clear            7146847    6150                35.3                 35.9
## 2 Few Clouds       6035580    4034                23.2                 30.3
## 3 Broken Clouds    4243887    3551                20.4                 21.3
## 4 Rain              1526461    2141                12.3                 7.67
## 5 Cloudy            929978    1464                 8.41                 4.67
## 6 Snowfall           15051      60                 0.345                0.0756
## 7 Thunderstorm      8168       14                 0.0804                0.0410
```

On voit que la fréquence d'apparition d'un code météo est équivalente au pourcentage d'emprunt que représente ce code. On peut donc se demander si la météo influence l'emprunt de vélib

Détails du nombre d'emprunt selon le code météo.

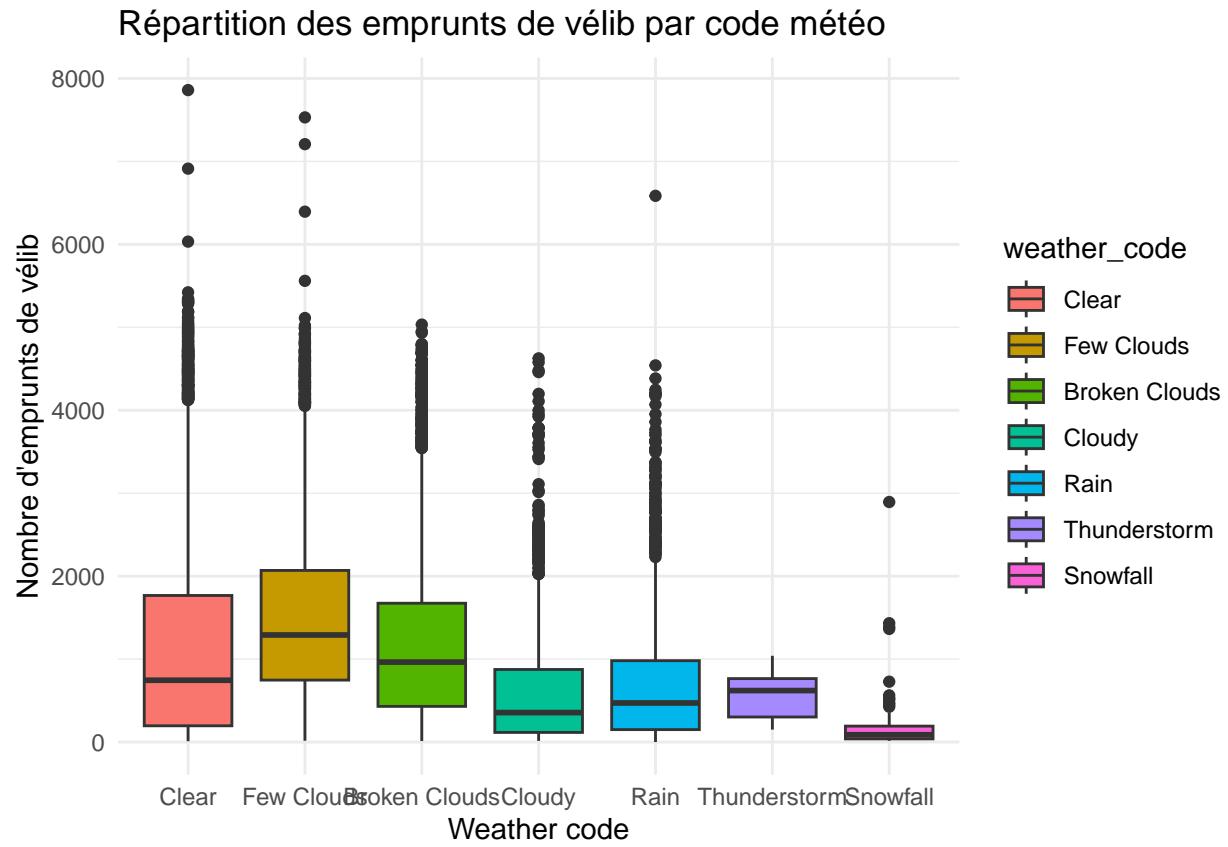
```
# Résumé de `cnt` par code météo
summary_by_weather <- bike_data %>%
  group_by(weather_code) %>%
  summarise(
    moyenne = mean(cnt),
    mediane = median(cnt),
    ecart_type = sd(cnt),
    minimum = min(cnt),
    maximum = max(cnt)
  )

print(summary_by_weather)

## # A tibble: 7 x 6
##   weather_code moyenne mediane ecart_type minimum maximum
##   <fct>        <dbl>    <dbl>      <dbl>    <dbl>    <dbl>
## 1 Clear         1162.     745       1187.     10      7860
## 2 Few Clouds    1496.    1291      1085.     16      7531
## 3 Broken Clouds 1195.    964       1015.     12      5033
## 4 Cloudy        635.     355       751.      14      4626
## 5 Rain           713.     471       765.      0       6585
## 6 Thunderstorm   583.    620.      283.     150     1040
## 7 Snowfall       251.     88        471.     15      2894
```

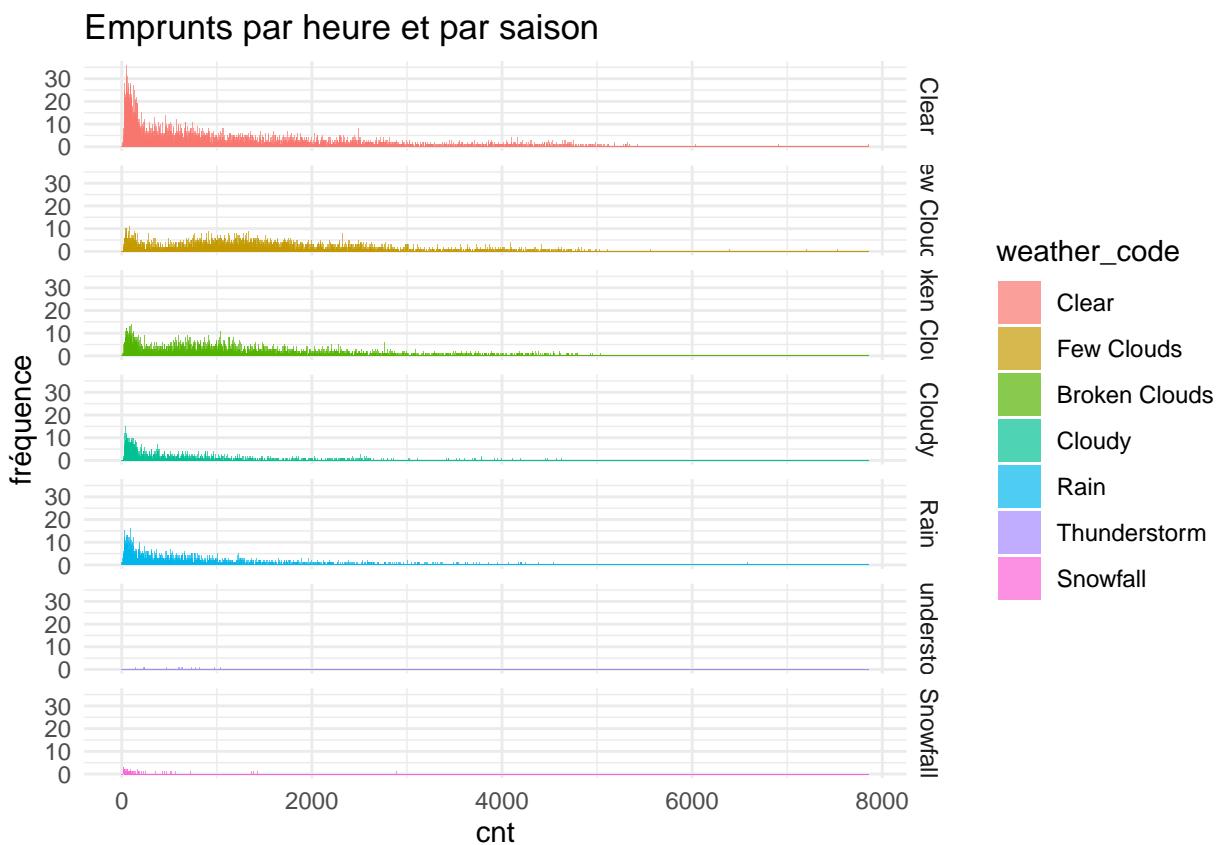
Visualisons cela mieux avec les Boxplots des distributions des emprunts (cnt) suivant la météo (weather_code).

```
ggplot(bike_data, aes(x = weather_code, y = cnt, fill = weather_code)) +
  geom_boxplot() +
  labs(
    title = "Répartition des emprunts de vélib par code météo",
    x = "Weather code",
    y = "Nombre d'emprunts de vélib"
  ) +
  theme_minimal()
```



Histogrammes représentant les distributions des emprunts (cnt) suivant la météo (weather_code).

```
ggplot(bike_data, aes(x =cnt,fill=weather_code)) +  
  geom_histogram(binwidth = 2, alpha = 0.7) +  
  facet_grid(weather_code ~ .)+  
  labs(title = "Emprunts par heure et par saison", x = "cnt", y = "fréquence") +  
  theme_minimal()
```



Ce graphe (bien que nous l'admettons difficilement lisible) nous montre quand même des différences entre les codes météo.

Calculons le rapport de corrélation empirique entre cnt et weather_code.

```
eta2(bike_data$cnt,bike_data$weather_code)

## [1] 0.06538532
```

La corrélation est assez faible. Maintenant regardons si oui ou non ces deux variables sont reliés.

```
anova_result <- aov(cnt ~ factor(weather_code), data = bike_data)
summary(anova_result)
```

```
##                               Df    Sum Sq   Mean Sq F value Pr(>F)
## factor(weather_code)      6 1.341e+09 223433638     203 <2e-16 ***
## Residuals                 17407 1.916e+10   1100850
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La valeur de p nous dit que le weather_code influe sur le nombre d'emprunt mais le coefficient de corrélation nous dit que le weather_code a une petite influence sur cnt.

6. Ajustement d'un modèle linéaire, avec l'étape de validation.

Construisons un modèle linéaire entre cnt et t1.

```
modele_lm <- lm(cnt ~ t1, data = bike_data)

summary(modele_lm)

## 
## Call:
## lm(formula = cnt ~ t1, data = bike_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1930.4  -680.0  -227.9   426.8  6234.0 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 199.04     18.57   10.72 <2e-16 ***
## t1          75.72      1.36   55.69 <2e-16 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 999.8 on 17412 degrees of freedom
## Multiple R-squared:  0.1512, Adjusted R-squared:  0.1511 
## F-statistic: 3101 on 1 and 17412 DF,  p-value: < 2.2e-16
```

p est suffisamment faible pour déduire qu'il y a un lien entre t1 et cnt. Cependant R2 est relativement faible ce qui laisse supposer qu'il y a d'autre variables qui influencent cnt.

On peut supposer qu'une variable comme l'heure de l'emprunt peut influencer cnt. Nous créerons donc une nouvelle variable hour.

```
bike_data<-bike_data%>%
  mutate(hour=hour(timestamp))

modele_lm <- lm(cnt ~ hour, data = bike_data)

summary(modele_lm)

## 
## Call:
## lm(formula = cnt ~ hour, data = bike_data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1645.8  -673.6  -301.0   395.1  6566.7 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 557.051    15.106   36.88 <2e-16 ***
## hour        50.902     1.125   45.26 <2e-16 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 1026 on 17412 degrees of freedom
```

```

## Multiple R-squared:  0.1053, Adjusted R-squared:  0.1052
## F-statistic:  2048 on 1 and 17412 DF,  p-value: < 2.2e-16

```

hour a donc une influence sur cnt. Cependant cette influence doit aussi dépendre des pics horaires lié à l'heure d'emprunt.

Pour prendre en compte les pics horaire étudions un modèle créé avec `poly(hour,2)`.

```
modele_lm <- lm(cnt ~ poly(hour,2), data = bike_data)
```

```
summary(modele_lm)
```

```

##
## Call:
## lm(formula = cnt ~ poly(hour, 2), data = bike_data)
##
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -1616.2 -609.7 -269.8  409.0 6235.3 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1143.102    6.828 167.43 <2e-16 ***
## poly(hour, 2)1 46453.886   900.979  51.56 <2e-16 ***
## poly(hour, 2)2 -64896.287   900.979 -72.03 <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 901 on 17411 degrees of freedom
## Multiple R-squared:  0.3107, Adjusted R-squared:  0.3106 
## F-statistic:  3923 on 2 and 17411 DF,  p-value: < 2.2e-16

```

On voit tout de suite que ce modèle est plus précis que les précédents car R2 est égal à 0.31.

Maintenant créons un modèle plus précis en utilisant t1 et `poly(hour,2)`.

```
modele_lm <- lm(cnt ~ t1+poly(hour,2), data = bike_data)
```

```
summary(modele_lm)
```

```

##
## Call:
## lm(formula = cnt ~ t1 + poly(hour, 2), data = bike_data)
##
## Residuals:
##    Min     1Q   Median     3Q    Max 
## -1891.9 -541.3 -237.7  351.5 6057.4 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 460.262    16.142  28.51 <2e-16 ***
## t1          54.767     1.187   46.14 <2e-16 ***
## poly(hour, 2)1 39660.457   863.158   45.95 <2e-16 ***
## poly(hour, 2)2 -58943.790   860.237  -68.52 <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 850.5 on 17410 degrees of freedom

```

```
## Multiple R-squared:  0.3858, Adjusted R-squared:  0.3857
## F-statistic:  3645 on 3 and 17410 DF,  p-value: < 2.2e-16
```

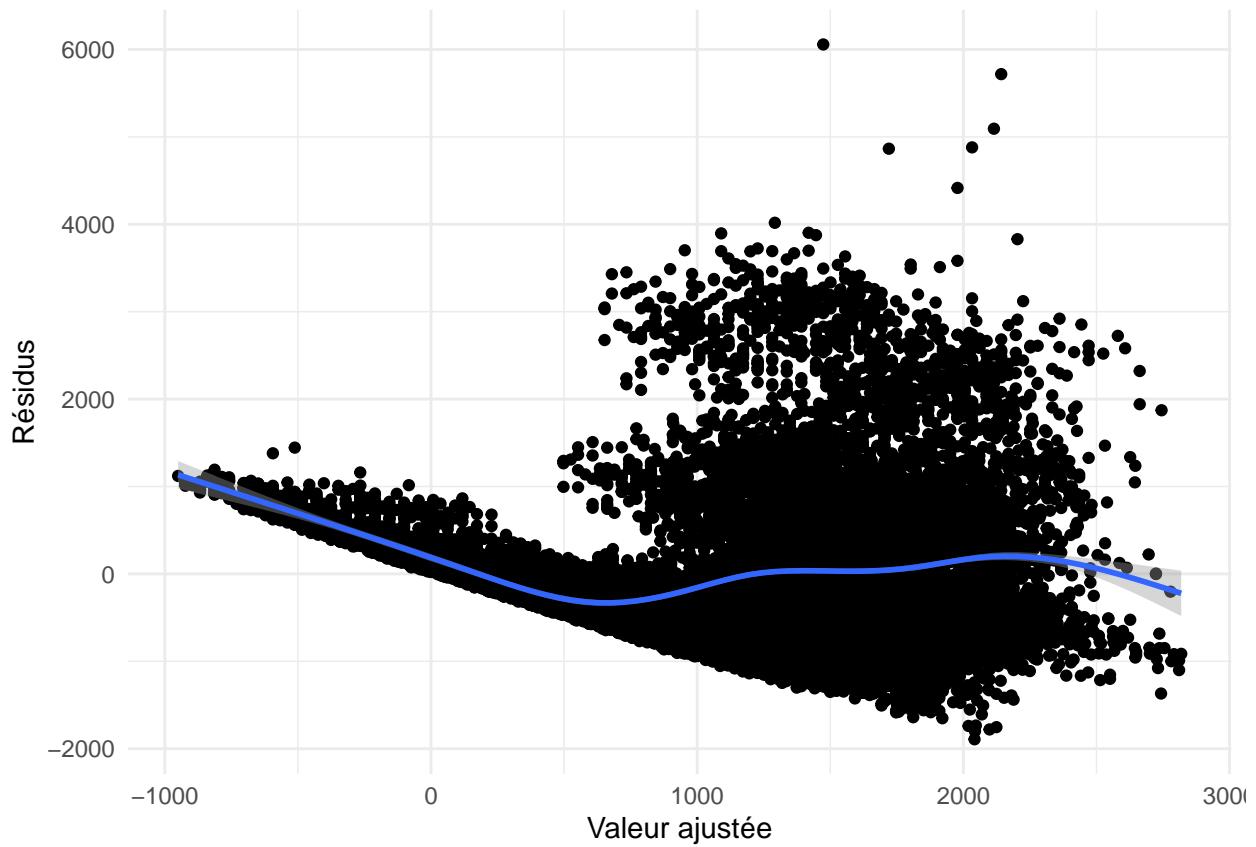
On voit que le meilleur modèle est le dernier. On a un R² plus important (0.15 avec juste t1 à 0.38 pour ce modèle), une amélioration sur les erreurs résiduels (999 à 850) et une étendue des résidus plus faible.

(si on ajoute les variables is_holiday+is_weekend+weather_code+wind_speed+season+hum, R² passe à 0.48 mais les résultats qui vont suivre sont les même donc pour plus de simplicité nous les avons enlevés)

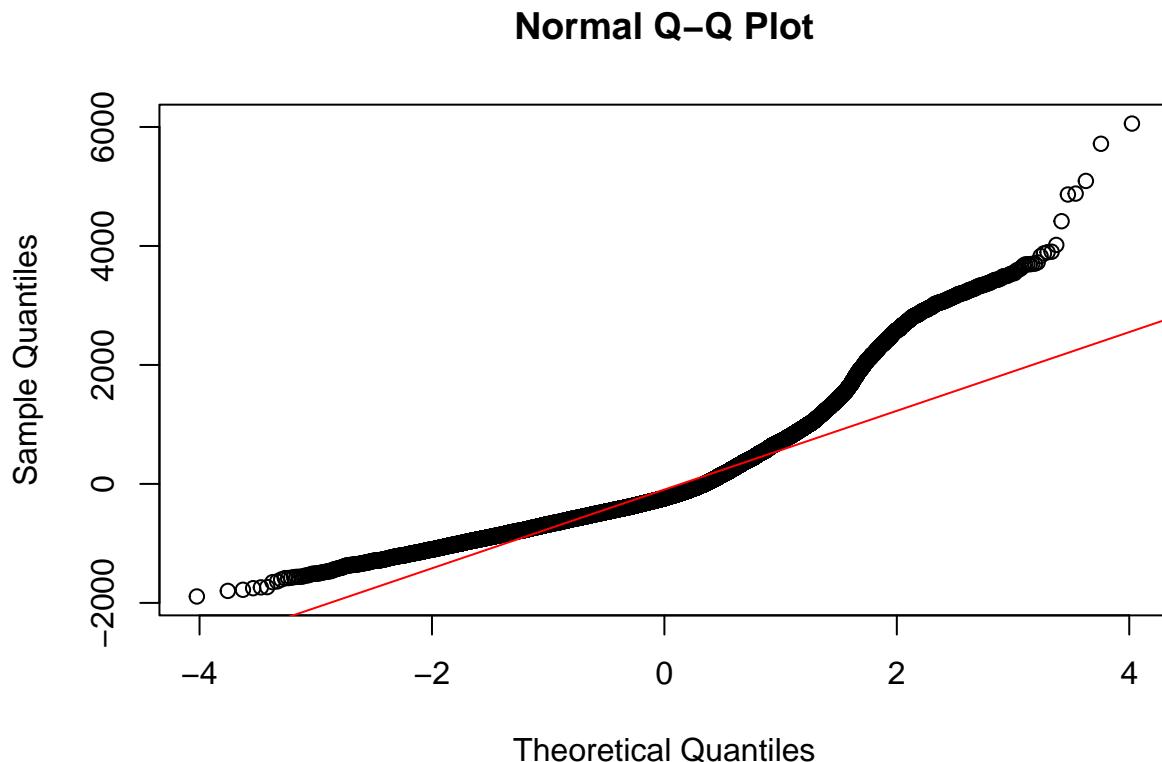
Voyons maintenant si ces résidus suivent une loi normale.

```
ggplot(modele_lm, aes(x=modele_lm$fitted.values,y=modele_lm$residuals)) +
  geom_point(alpha=1) +
  geom_smooth()+
  labs(y="Résidus",
       x="Valeur ajustée")+
  theme_minimal()

## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
qqnorm(residuals(modele_lm))
qqline(residuals(modele_lm), col="red")
```



On voit sur les 2 graphes que la normalité des résidus semble peu probable. En effet, les résidus fonction des valeurs ajustées forme un U et le QQ plot s'écarte trop de la droite

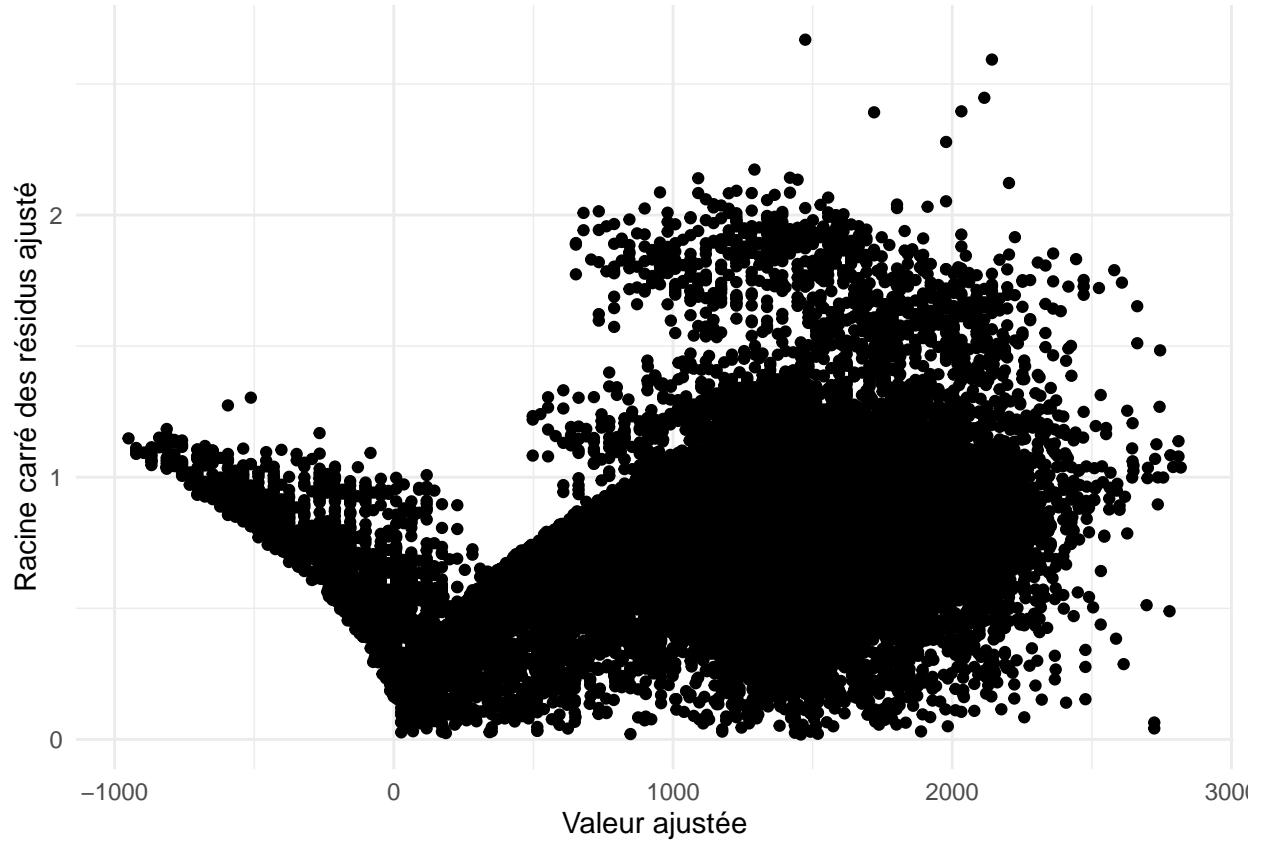
```
# Test d'Anderson-Darling
ad_test <- ad.test(residuals(modele_lm))
print(ad_test)
```

```
##
##  Anderson-Darling normality test
##
##  data:  residuals(modele_lm)
##  A = 620.35, p-value < 2.2e-16
```

La non normalité constatée sur les graphes est confirmée avec le test de Anderson-Darling.

Etudions l'homoscédaticité du modèle.

```
residu<-modele_lm$residuals  
valeur_ajuste<-modele_lm$fitted.values  
  
ecart_type<-sqrt(sum(residu^2)/length(residu))  
  
residu_normalise<-residu/ecart_type  
  
ggplot(data=data.frame(residu_normalise,valeur_ajuste), aes(x=valeur_ajuste,y=sqrt(abs(residu_normalise)))  
geom_point(alpha=1) +  
labs(y="Racine carré des résidus ajusté",  
x="Valeur ajustée") +  
theme_minimal()
```



On voit un patern et une bande. Cela montre une hétéroscédaticité.

Ce modèle a donc une hétéroscédaticité et une non normalité des résidus.

Essayons de créer un nouveau modèle avec le log de cnt. Cela permet d'amoindrir les valeurs extrêmes.

```
bike_data<-bike_data%>%
  mutate(cnt_log=log(cnt+1))

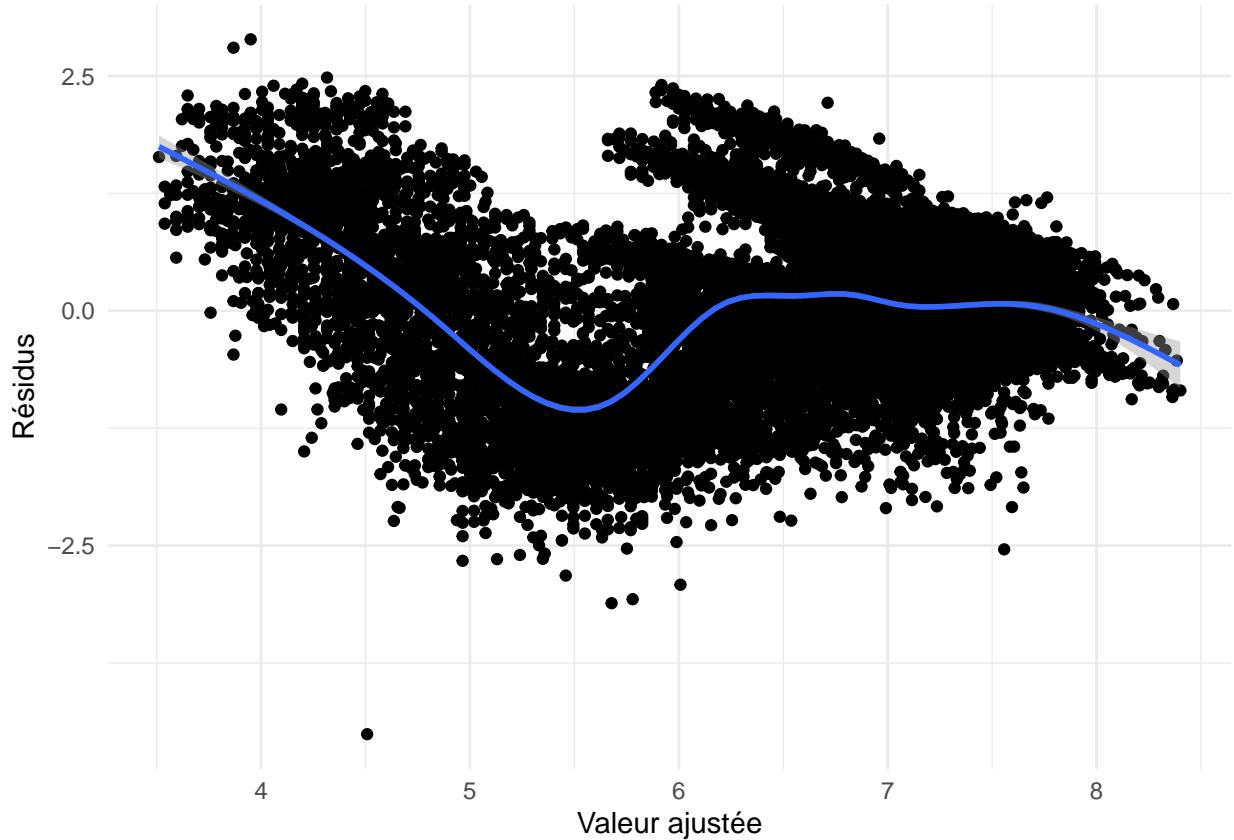
modele_lm <- lm(cnt_log ~ t1+poly(hour,2), data = bike_data)
#wind_speed+is_holiday+is_weekend+season+hum+weather_code
summary(modele_lm)

##
## Call:
## lm(formula = cnt_log ~ t1 + poly(hour, 2), data = bike_data)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -4.5078 -0.4478 -0.0099  0.4544  2.8893 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.754882  0.015713 366.24   <2e-16 ***
## t1          0.054869  0.001155  47.49   <2e-16 ***
## poly(hour, 2)1 84.184099  0.840229 100.19   <2e-16 ***
## poly(hour, 2)2 -77.079045  0.837385 -92.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8279 on 17410 degrees of freedom
## Multiple R-squared:  0.5835, Adjusted R-squared:  0.5835 
## F-statistic:  8131 on 3 and 17410 DF,  p-value: < 2.2e-16
```

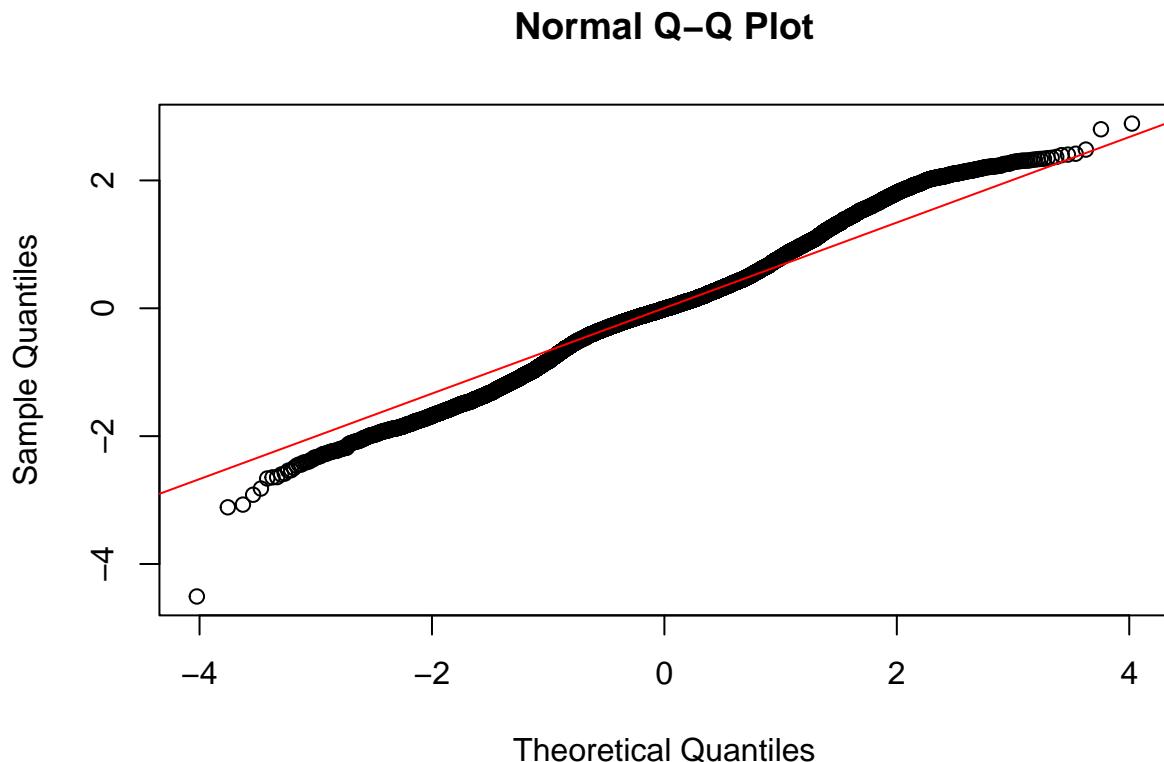
On voit que ce modèle est encore plus précis que le précédent.

Testons la normalité des résidus et l'homoscédasticité du modèle.

```
ggplot(modele_lm, aes(x=modele_lm$fitted.values,y=modele_lm$residuals)) +  
  geom_point(alpha=1) +  
  geom_smooth() +  
  labs(y="Résidus",  
       x="Valeur ajustée") +  
  theme_minimal()  
  
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
qqnorm(residuals(modele_lm))
qqline(residuals(modele_lm), col="red")
```



```
# Test d'Anderson-Darling
ad_test <- ad.test(residuals(modele_lm))
print(ad_test)

##
##  Anderson-Darling normality test
##
##  data:  residuals(modele_lm)
##  A = 66.261, p-value < 2.2e-16
```

```

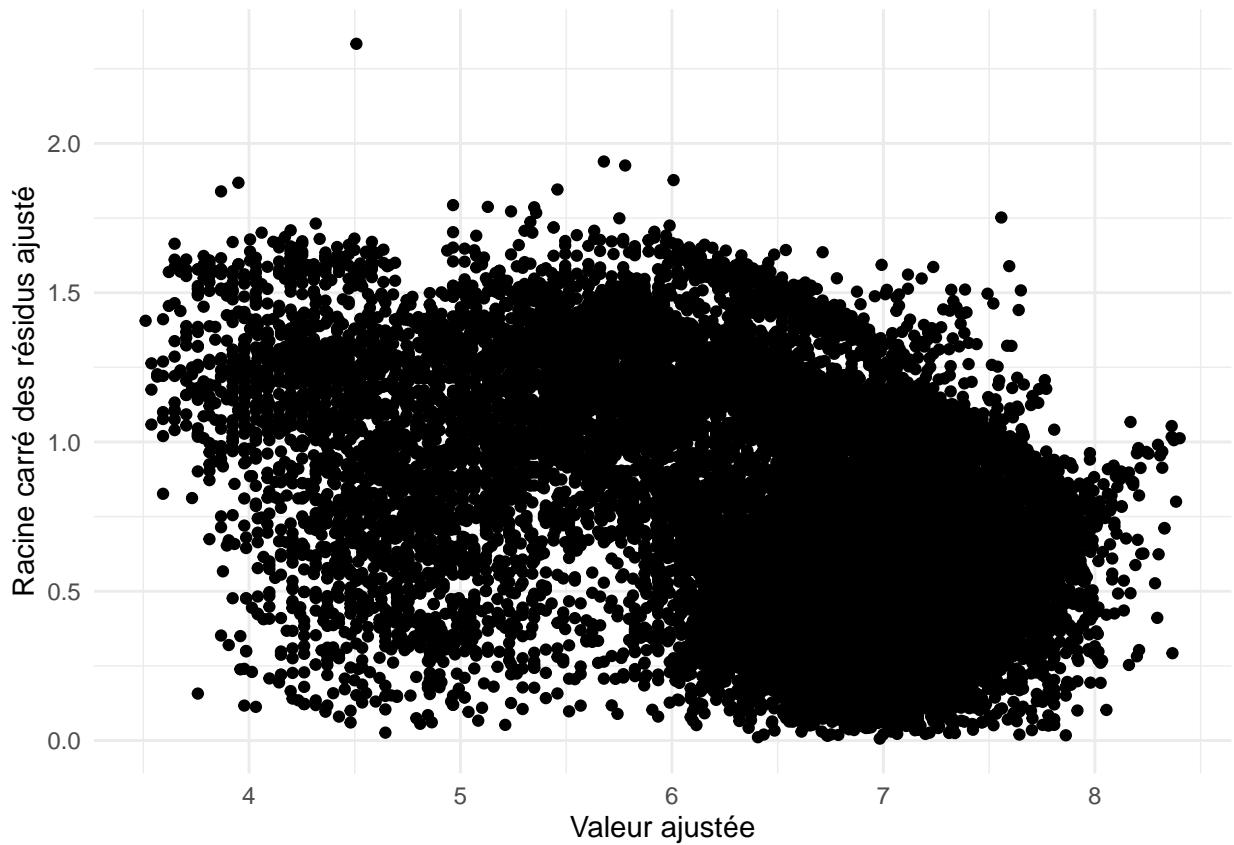
residu<-modele_lm$residuals
valeur_ajuste<-modele_lm$fitted.values

ecart_type<-sqrt(sum(residu^2)/length(residu))

residu_normalise<-residu/ecart_type

ggplot(data=data.frame(residu_normalise,valeur_ajuste), aes(x=valeur_ajuste,y=sqrt(abs(residu_normalise)))
geom_point(alpha=1) +
labs(y="Racine carré des résidus ajusté",
x="Valeur ajustée")+
theme_minimal()

```



On voit un patern et des bandes. Cela montre une hétérosécédaticité.

Notre modèle ayant un $R^2=0.5833$ capture une bonne part des variations de $\log(\text{cnt})$. Cependant, nous observons une non-normalité des résidus, vu sur les graphes et à l'aide du test d'Anderson-Darling, et une hétérosécédaticité, vu sur le graphe.

On peut en conclure que soit la relation entre $\log(\text{cnt})/\text{cnt}$ et les prédicteurs n'est pas linéaire soit il y a une hétérogénéité des données.