Rédaction d’un rapport

Fabien Haury

2022-09-07

1 Librairies

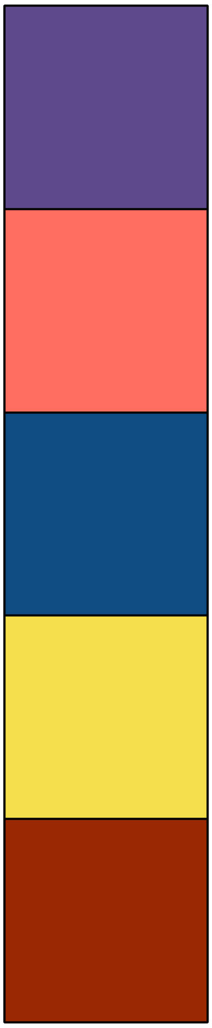
```r
library(plyr)
library(tidyverse)
library(scales)
library(ggthemes)
library(finalfit)
library(summarytools)
library(crosstable)
library(flextable)
library(vcd)
library(ggsankey)
library(ggalluvial)
library(ggpubr)
library(nortest)
library(xtable)
library(clipr)

library(extrafont)
loadfonts(device = "win")
```

2 Spécification police et thème

```r
theme_set(theme_minimal(base_family = "serif")) # Police Serif
theme_update(axis.line = element_line(color='black')
,
          plot.background = element_blank(),
          panel.grid.major = element_blank(),# Suppression des lignes de grilles principales
          panel.grid.minor = element_blank(),# Suppression des lignes de grilles mineures
          panel.border = element_blank(),
          axis.title = element_text(size = 15), # Changement de la taille de police des titres des axes de graphique
          axis.text.x = element_text(size = 12), # Changement de la taille de police des labels de l'axe x
          axis.text.y = element_text(size = 12)) # Changement de la taille de police des labels de l'axe y
```

3 Spécification code couleur

```r
# Code couleur pour chaque année et ainsi que pour le jeu de Donnes fusionné si besoin
my_color <- c("#5f4b8b", "#ff6f61", "#0f4c81", "#f5df4d", "#992900")
show_col(my_color, ncol = 1, labels = F)
```



4 Création fonction

```
# Pour transformer les langages en 0 ou 1
transformation_langage <- function(dset, colonne, observation) {
  if_else(dset[, colonne] == observation, 1, 0)
}


# Exploration complexe (fill et facet_wrap) des Donnes
eda_complexe <- function(dataframe, xcol, fill, target_facet) {
  ggplot(data = dataframe, aes({{ xcol }}, fill = {{ fill }})) +
    geom_bar(color = "black", position = "dodge") +
    facet_wrap(vars({{ target_facet }}), scales = "free_y")
}
```

5 Import et préparations des Donnes

5.1 2018

5.1.1 Import année 2018

```
ks_2018 <- read_csv("jobs/kaggle_survey_2018/kaggle_survey_2018_responses.csv",
                    na = "NA") %>%
           mutate(Annee = 2018)

ks_2018_question <- ks_2018[1, ] # Selection de la première ligne du df contenant les questions
ks_2018 <- ks_2018[2:nrow(ks_2018),] # Selection du reste des lignes
```

5.1.2 Préparation année 2018

```
# Selection des variables d'interets
ks_2018 <- ks_2018 %>% select(Q1, Q2, Q3, Q4 , Q5, Q9, Q16_Part_1:Q16_Part_16, Annee)

# Transformation des variables
ks_2018 <- ks_2018 %>% mutate(Q1 = case_when(Q1 == "Female" ~ "Femme",
               Q1 == "Male" ~ "Homme",
               TRUE ~ Q1),
     Q4 = str_replace_all(Q4, c("(^Some.*)|(^No.*)|(^I.*)" = "Autre")),
     Q2 = case_when(Q2 %in% c("18-21", "22-24", "25-29") ~ "18-29",
               Q2 %in% c("30-34", "35-39") ~ "30-39",
               Q2 %in% c("40-44", "45-49") ~ "40-49",
               Q2 %in% c("50-54", "55-59") ~ "50-59",
               Q2 %in% c("60-69") ~ "60-69",
               TRUE ~ "70+"),
     Q3 = case_when(Q3 == "United States of America" ~ "U.S.A",
               Q3 == "Iran, Islamic Republic of..." ~ "Iran",
               Q3 == "Hong Kong (S.A.R.)" ~ "Honk Kong",
               Q3 == "United Kingdom of Great Britain and Northern Ireland" ~ "UK",
               Q3 == "United Arab Emirates" ~ "U.A.E",
               Q3 == "Republic of Korea" ~ "South Korea",
               Q3 == "I do not wish to disclose my location" |
                Q3 == "Other" ~ "Inconnu",
               TRUE ~ Q3),
     Q9 = str_replace_all(Q9, c("(^Some.*)|(^No.*)|(^I.*)" = "Inconnu")),
     Q9 = case_when(Q9 == "I do not wish to disclose my approximate yearly compensation" ~ "Inconnu",
               Q9 == "$0 ($USD)" |
                Q9 == "$0-999" | Q9 == "0-10,000" |
                Q9 == "5,000-7,499" | Q9 == "4,000-4,999" |
                Q9 == "2,000-2,999" | Q9 == "7,500-9,999" |
                Q9 == "3,000-3,999" | Q9 == "1,000-1,999" ~ "0-10K",
               Q9 == "10-20,000" | Q9 == "10,000-14,999" |
                Q9 == "15,000-19,999" ~ "10K-20K",
               Q9 == "20-30,000" | Q9 == "20,000-24,999" |
                Q9 == "25,000-29,999" ~ "20K-30K",
               Q9 == "30-40,000" | Q9 == "30,000-39,999" ~ "30K-40K",
               Q9 == "40-50,000" | Q9 == "40,000-49,999" ~ "40K-50K",
               Q9 == "50-60,000" | Q9 == "50,000-59,999" ~ "50K-60K",
               Q9 == "60-70,000" | Q9 == "60,000-69,999" ~ "60K-70K",
               Q9 == "70-80,000" | Q9 == "70,000-79,999" ~ "70K-80K",
               Q9 == "80-90,000" | Q9 == "80,000-89,999" ~ "80K-90K",
               Q9 == "90-100,000" | Q9 == "90,000-99,999" ~ "90K-100K",
               Q9 == "125-150,000" | Q9 == "100-125,000" |
                Q9 == "150-200,000" | Q9 == "125,000-149,999" |
                Q9 == "100,000-124,999" | Q9 == "150,000-199,999" |
                Q9 == "200-250,000"  |  Q9 =="200,000-249,999" ~ "100K-250K",
               Q9 == "300,000-500,000" | Q9 == "300,000-499,999" |
                Q9 == "250-300,000" | Q9 == "300-400,000" |
                Q9 == "400-500,000" | Q9 == "250,000-299,999"  ~ "250K-500K",
               Q9 == "500,000+" | Q9 == "> $500,000" |
                Q9 == "$500,000-999,999" ~ "500K +",
               TRUE ~ Q9)) %>% rename(Q2 = Q1, Q1 = Q2, Q6 = Q9)
```

5.1.3 Transformation langage

```r
# Tranformation des langages de programmation en 0 ou 1
ks_2018$Q16_Part_1 <- transformation_langage(ks_2018, "Q16_Part_1", "Python")
ks_2018$Q16_Part_2 <- transformation_langage(ks_2018, "Q16_Part_2", "R")
ks_2018$Q16_Part_3 <- transformation_langage(ks_2018, "Q16_Part_3", "SQL")
ks_2018$Q16_Part_4 <- transformation_langage(ks_2018, "Q16_Part_4", "Bash")
ks_2018$Q16_Part_5 <- transformation_langage(ks_2018, "Q16_Part_5", "Java")
ks_2018$Q16_Part_6 <- transformation_langage(ks_2018, "Q16_Part_6", "Javascript/Typescript")
ks_2018$Q16_Part_7 <- transformation_langage(ks_2018, "Q16_Part_7", "Visual Basic/VBA")
ks_2018$Q16_Part_8 <- transformation_langage(ks_2018, "Q16_Part_8", "C/C++")
ks_2018$Q16_Part_9 <- transformation_langage(ks_2018, "Q16_Part_9", "MATLAB")
ks_2018$Q16_Part_10 <- transformation_langage(ks_2018, "Q16_Part_10", "Scala")
ks_2018$Q16_Part_11 <- transformation_langage(ks_2018, "Q16_Part_11", "Julia")
ks_2018$Q16_Part_12 <- transformation_langage(ks_2018, "Q16_Part_12", "Go")
ks_2018$Q16_Part_13 <- transformation_langage(ks_2018, "Q16_Part_13", "C#/.NET")
ks_2018$Q16_Part_14 <- transformation_langage(ks_2018, "Q16_Part_14", "PHP")
ks_2018$Q16_Part_15 <- transformation_langage(ks_2018, "Q16_Part_15", "Ruby")
ks_2018$Q16_Part_16 <- transformation_langage(ks_2018, "Q16_Part_16", "SAS/STATA")

langage <- c("Q16_Part_1", "Q16_Part_2", "Q16_Part_3", "Q16_Part_4", "Q16_Part_5", "Q16_Part_6",
    "Q16_Part_7", "Q16_Part_8", "Q16_Part_9", "Q16_Part_10", "Q16_Part_11", "Q16_Part_12",
    "Q16_Part_13", "Q16_Part_14", "Q16_Part_15", "Q16_Part_16")

ks_2018$Langage <- rowSums(ks_2018[, langage]
)
ks_2018 <- ks_2018[,!(names(ks_2018) %in% langage)]
rm(langage)
```

5.2 2019

5.2.1 Import année 2019

```r
ks_2019 <- read_csv("jobs/kaggle_survey_2019/kaggle_survey_2019_responses.csv",
            na = "NA") %>%
  mutate(Annee = 2019
)

ks_2019_question <- ks_2019[1, ] # Selection de la première ligne du df contenant les questions
ks_2019 <- ks_2019[2:nrow(ks_2019),] # Selection du reste des lignes
```

5.2.2 Préparation année 2019

```r
# Selection des variables d'interets
ks_2019 <- ks_2019 %>% select(Q1, Q2, Q3, Q4 , Q5, Q10, Q18_Part_1:Q18_Part_10, Annee)

# Transformation des variables
ks_2019 <- ks_2019 %>%
  mutate(Q2 = case_when(Q2 == "Female" ~ "Femme",
                        Q2 == "Male" ~ "Homme",
                        TRUE ~ Q2),
         Q4 = str_replace_all(Q4, c("(^Some.*)|(^No.*)|(^I.*)" = "Autre")),
         Q1 = case_when(Q1 %in% c("18-21", "22-24", "25-29") ~ "18-29",
                        Q1 %in% c("30-34", "35-39") ~ "30-39",
                        Q1 %in% c("40-44", "45-49") ~ "40-49",
                        Q1 %in% c("50-54", "55-59") ~ "50-59",
                        Q1 %in% c("60-69") ~ "60-69",
                        TRUE ~ "70+"),
         Q3 = case_when(Q3 == "United States of America" ~ "U.S.A",
                        Q3 == "Iran, Islamic Republic of..." ~ "Iran",
                        Q3 == "Hong Kong (S.A.R.)" ~ "Honk Kong",
                        Q3 == "United Kingdom of Great Britain and Northern Ireland" ~ "UK",
                        Q3 == "United Arab Emirates" ~ "U.A.E",
                        Q3 == "Republic of Korea" ~ "South Korea",
                        Q3 == "I do not wish to disclose my location" |
                          Q3 == "Other" ~ "Inconnu",
                        TRUE ~ Q3),
         Q10 = str_replace_all(Q10, c("(^Some.*)|(^No.*)|(^I.*)" = "Inconnu")),
         Q10 = case_when(Q10 == "I do not wish to disclose my approximate yearly compensation" ~ "Inconnu",
                         Q10 == "$0 ($USD)" |
                           Q10 == "$0-999" | Q10 == "0-10,000" |
                           Q10 == "5,000-7,499" | Q10 == "4,000-4,999" |
                           Q10 == "2,000-2,999" | Q10 == "7,500-9,999" |
                           Q10 == "3,000-3,999" | Q10 == "1,000-1,999" ~ "0-10K",
                         Q10 == "10-20,000" | Q10 == "10,000-14,999" |
                           Q10 == "15,000-19,999" ~ "10K-20K",
                         Q10 == "20-30,000" | Q10 == "20,000-24,999" |
                           Q10 == "25,000-29,999" ~ "20K-30K",
                         Q10 == "30-40,000" | Q10 == "30,000-39,999" ~ "30K-40K",
                         Q10 == "40-50,000" | Q10 == "40,000-49,999" ~ "40K-50K",
                         Q10 == "50-60,000" | Q10 == "50,000-59,999" ~ "50K-60K",
                         Q10 == "60-70,000" | Q10 == "60,000-69,999" ~ "60K-70K",
                         Q10 == "70-80,000" | Q10 == "70,000-79,999" ~ "70K-80K",
                         Q10 == "80-90,000" | Q10 == "80,000-89,999" ~ "80K-90K",
                         Q10 == "90-100,000" | Q10 == "90,000-99,999" ~ "90K-100K",
                         Q10 == "125-150,000" | Q10 == "100-125,000" |
                          Q10 == "150-200,000" | Q10 == "125,000-149,999" |
                          Q10 == "100,000-124,999" | Q10 == "150,000-199,999" |
                          Q10 == "200-250,000"  |  Q10 =="200,000-249,999" ~ "100K-250K",
                        Q10 == "300,000-500,000" | Q10 == "300,000-499,999" |
                          Q10 == "250-300,000" | Q10 == "300-400,000" |
                          Q10 == "400-500,000" | Q10 == "250,000-299,999" ~ "250K-500K",
                        Q10 == "500,000+" | Q10 == "> $500,000" |
                          Q10 == "$500,000-999,999" ~ "500K +",
                         TRUE ~ Q10)) %>% rename(Q6 = Q10)
```

5.2.3 Transformation langage

```
# Tranformation des langages de programmation en 0 ou 1
ks_2019$Q18_Part_1 <- transformation_langage(ks_2019, "Q18_Part_1", "Python")
ks_2019$Q18_Part_2 <- transformation_langage(ks_2019, "Q18_Part_2", "R")
ks_2019$Q18_Part_3 <- transformation_langage(ks_2019, "Q18_Part_3", "SQL")
ks_2019$Q18_Part_4 <- transformation_langage(ks_2019, "Q18_Part_4", "C")
ks_2019$Q18_Part_5 <- transformation_langage(ks_2019, "Q18_Part_5", "C++")
ks_2019$Q18_Part_6 <- transformation_langage(ks_2019, "Q18_Part_6", "Java")
ks_2019$Q18_Part_7 <- transformation_langage(ks_2019, "Q18_Part_7", "Javascript")
ks_2019$Q18_Part_8 <- transformation_langage(ks_2019, "Q18_Part_8", "TypeScript")
ks_2019$Q18_Part_9 <- transformation_langage(ks_2019, "Q18_Part_9", "Bash")
ks_2019$Q18_Part_10 <- transformation_langage(ks_2019, "Q18_Part_10", "Matlab")


langage <- c("Q18_Part_1", "Q18_Part_2", "Q18_Part_3", "Q18_Part_4", "Q18_Part_5", "Q18_Part_6",
             "Q18_Part_7", "Q18_Part_8", "Q18_Part_9", "Q18_Part_10")

ks_2019$Langage <- rowSums(ks_2019[, langage]
)
ks_2019 <- ks_2019[,!(names(ks_2019) %in% langage)]
rm(langage)
```

5.3 2020

5.3.1 Import année 2020

```
ks_2020 <- read_csv("jobs/kaggle_survey_2020/kaggle_survey_2020_responses.csv",
                    na = "NA") %>%
  mutate(Annee = 2020
)

ks_2020_question <- ks_2020[1:21, ] # Selection de la première ligne du df contenant les questions
ks_2020 <- ks_2020[2:nrow(ks_2020),] # Selection du reste des lignes
```

5.3.2 Préparation année 2020

```
# Selection des variables d'interets
ks_2020 <- ks_2020 %>% select(Q1, Q2, Q3, Q4 , Q5, Q24, Q7_Part_1:Q7_Part_11, Annee)

# Transformation des variables
ks_2020 <- ks_2020 %>%
  mutate(Q2 = case_when(Q2 == "Woman" ~ "Femme",
                    Q2 == "Man" ~ "Homme",
                    TRUE ~ Q2),
       Q4 = str_replace_all(Q4, c("(^Some.*)|(^No.*)|(^I.*)" = "Autre")),
       Q1 = case_when(Q1 %in% c("18-21", "22-24", "25-29") ~ "18-29",
                    Q1 %in% c("30-34", "35-39") ~ "30-39",
                    Q1 %in% c("40-44", "45-49") ~ "40-49",
                    Q1 %in% c("50-54", "55-59") ~ "50-59",
                    Q1 %in% c("60-69") ~ "60-69",
                    TRUE ~ "70+"),
       Q3 = case_when(Q3 == "United States of America" ~ "U.S.A",
                    Q3 == "Iran, Islamic Republic of..." ~ "Iran",
                    Q3 == "Hong Kong (S.A.R.)" ~ "Honk Kong",
                    Q3 == "United Kingdom of Great Britain and Northern Ireland" ~ "UK",
                    Q3 == "United Arab Emirates" ~ "U.A.E",
                    Q3 == "Republic of Korea" ~ "South Korea",
                    Q3 == "I do not wish to disclose my location" |
                     Q3 == "Other" ~ "Inconnu",
                    TRUE ~ Q3),
       Q24 = str_replace_all(Q24, c("(^Some.*)|(^No.*)|(^I.*)" = "Inconnu")),
       Q24 = case_when(Q24 == "I do not wish to disclose my approximate yearly compensation" ~ "Inconnu",
                     Q24 == "$0 ($USD)" |
                      Q24 == "$0-999" | Q24 == "0-10,000" |
                      Q24 == "5,000-7,499" | Q24 == "4,000-4,999" |
                      Q24 == "2,000-2,999" | Q24 == "7,500-9,999" |
                      Q24 == "3,000-3,999" | Q24 == "1,000-1,999" ~ "0-10K",
                     Q24 == "10-20,000" | Q24 == "10,000-14,999" |
                      Q24 == "15,000-19,999" ~ "10K-20K",
                     Q24 == "20-30,000" | Q24 == "20,000-24,999" |
                      Q24 == "25,000-29,999" ~ "20K-30K",
                     Q24 == "30-40,000" | Q24 == "30,000-39,999" ~ "30K-40K",
                     Q24 == "40-50,000" | Q24 == "40,000-49,999" ~ "40K-50K",
                     Q24 == "50-60,000" | Q24 == "50,000-59,999" ~ "50K-60K",
                     Q24 == "60-70,000" | Q24 == "60,000-69,999" ~ "60K-70K",
                     Q24 == "70-80,000" | Q24 == "70,000-79,999" ~ "70K-80K",
                     Q24 == "80-90,000" | Q24 == "80,000-89,999" ~ "80K-90K",
                     Q24 == "90-100,000" | Q24 == "90,000-99,999" ~ "90K-100K",
                     Q24 == "125-150,000" | Q24 == "100-125,000" |
                      Q24 == "150-200,000" | Q24 == "125,000-149,999" |
                      Q24 == "100,000-124,999" | Q24 == "150,000-199,999" |
                      Q24 == "200-250,000"  |  Q24 =="200,000-249,999" ~ "100K-250K",
                    Q24 == "300,000-500,000" | Q24 == "300,000-499,999" |
                      Q24 == "250-300,000" | Q24 == "300-400,000" |
                      Q24 == "400-500,000" | Q24 == "250,000-299,999" ~ "250K-500K",
                    Q24 == "500,000+" | Q24 == "> $500,000" |
                      Q24 == "$500,000-999,999" | Q24 == ">$1,000,000" ~ "500K +",
                     TRUE ~ Q24))%>% rename(Q6 = Q24)
```

5.3.3 Transformation langage

```
# Tranformation des langages de programmation en 0 ou 1
ks_2020$Q7_Part_1 <- transformation_langage(ks_2020, "Q7_Part_1", "Python")
ks_2020$Q7_Part_2 <- transformation_langage(ks_2020, "Q7_Part_2", "R")
ks_2020$Q7_Part_3 <- transformation_langage(ks_2020, "Q7_Part_3", "SQL")
ks_2020$Q7_Part_4 <- transformation_langage(ks_2020, "Q7_Part_4", "C")
ks_2020$Q7_Part_5 <- transformation_langage(ks_2020, "Q7_Part_5", "C++")
ks_2020$Q7_Part_6 <- transformation_langage(ks_2020, "Q7_Part_6", "Java")
ks_2020$Q7_Part_7 <- transformation_langage(ks_2020, "Q7_Part_7", "Javascript")
ks_2020$Q7_Part_8 <- transformation_langage(ks_2020, "Q7_Part_8", "Julia")
ks_2020$Q7_Part_9 <- transformation_langage(ks_2020, "Q7_Part_9", "Swift")
ks_2020$Q7_Part_10 <- transformation_langage(ks_2020, "Q7_Part_10", "Bash")
ks_2020$Q7_Part_11 <- transformation_langage(ks_2020, "Q7_Part_11", "Matlab")


langage <- c("Q7_Part_1", "Q7_Part_2", "Q7_Part_3", "Q7_Part_4", "Q7_Part_5", "Q7_Part_6",
             "Q7_Part_7", "Q7_Part_8", "Q7_Part_9", "Q7_Part_10", "Q7_Part_11")

ks_2020$Langage <- rowSums(ks_2020[, langage]
)
ks_2020 <- ks_2020[,!(names(ks_2020) %in% langage)]
rm(langage)
```

5.4 2021

5.4.1 Import année 2021

```
ks_2021 <- read_csv("jobs/kaggle_survey_2021/kaggle_survey_2021_responses.csv",
                    na = "NA") %>%
  mutate(Annee = 2021
)

ks_2021_question <- ks_2021[1, ] # Selection de la première ligne du df contenant les questions
ks_2021 <- ks_2021[2:nrow(ks_2021),] # Selection du reste des lignes
```

5.4.2 Préparation année 2021

```
# Selection des variables d'interets
ks_2021 <- ks_2021 %>% select(Q1, Q2, Q3, Q4 , Q5, Q25, Q7_Part_1:Q7_Part_11, Annee)

# Transformation des variables
ks_2021 <- ks_2021 %>%
  mutate(Q2 = case_when(Q2 == "Woman" ~ "Femme",
                       Q2 == "Man" ~ "Homme",
                       TRUE ~ Q2),
        Q4 = str_replace_all(Q4, c("(^Some.*)|(^No.*)|(^I.*)" = "Autre")),
        Q1 = case_when(Q1 %in% c("18-21", "22-24", "25-29") ~ "18-29",
                       Q1 %in% c("30-34", "35-39") ~ "30-39",
                       Q1 %in% c("40-44", "45-49") ~ "40-49",
                       Q1 %in% c("50-54", "55-59") ~ "50-59",
                       Q1 %in% c("60-69") ~ "60-69",
                       TRUE ~ "70+"),
        Q3 = case_when(Q3 == "United States of America" ~ "U.S.A",
                       Q3 == "Iran, Islamic Republic of..." ~ "Iran",
                       Q3 == "Hong Kong (S.A.R.)" ~ "Honk Kong",
                       Q3 == "United Kingdom of Great Britain and Northern Ireland" ~ "UK",
                       Q3 == "United Arab Emirates" ~ "U.A.E",
                       Q3 == "Republic of Korea" ~ "South Korea",
                       Q3 == "I do not wish to disclose my location" |
                        Q3 == "Other" ~ "Inconnu",
                       TRUE ~ Q3),
        Q25 = str_replace_all(Q25, c("(^Some.*)|(^No.*)|(^I.*)" = "Inconnu")),
        Q25 = case_when(Q25 == "I do not wish to disclose my approximate yearly compensation" ~ "Inconnu",
                        Q25 == "$0 ($USD)" |
                          Q25 == "$0-999" | Q25 == "0-10,000" |
                          Q25 == "5,000-7,499" | Q25 == "4,000-4,999" |
                          Q25 == "2,000-2,999" | Q25 == "7,500-9,999" |
                          Q25 == "3,000-3,999" | Q25 == "1,000-1,999" ~ "0-10K",
                        Q25 == "10-20,000" | Q25 == "10,000-14,999" |
                          Q25 == "15,000-19,999" ~ "10K-20K",
                        Q25 == "20-30,000" | Q25 == "20,000-24,999" |
                          Q25 == "25,000-29,999" ~ "20K-30K",
                        Q25 == "30-40,000" | Q25 == "30,000-39,999" ~ "30K-40K",
                        Q25 == "40-50,000" | Q25 == "40,000-49,999" ~ "40K-50K",
                        Q25 == "50-60,000" | Q25 == "50,000-59,999" ~ "50K-60K",
                        Q25 == "60-70,000" | Q25 == "60,000-69,999" ~ "60K-70K",
                        Q25 == "70-80,000" | Q25 == "70,000-79,999" ~ "70K-80K",
                        Q25 == "80-90,000" | Q25 == "80,000-89,999" ~ "80K-90K",
                        Q25 == "90-100,000" | Q25 == "90,000-99,999" ~ "90K-100K",
                        Q25 == "125-150,000" | Q25 == "100-125,000" | Q25 == "100,000-124,999" |
                          Q25 == "150-200,000" | Q25 == "125,000-149,999" | Q25 == "200,000-249,999" |
                          Q25 == "100,000-125,999" | Q25 == "150,000-199,999" |
                          Q25 == "200-250,000"  |  Q25 =="200,000-259,999" ~ "100K-250K",
                        Q25 == "300,000-500,000" | Q25 == "300,000-499,999" |
                          Q25 == "250-300,000" | Q25 == "300-400,000" |
                          Q25 == "400-500,000" | Q25 == "250,000-299,999" ~ "250K-500K",
                        Q25 == "500,000+" | Q25 == "> $500,000" |
                          Q25 == "$500,000-999,999" | Q25 == ">$1,000,000" ~ "500K +",
                        TRUE ~ Q25)) %>% rename(Q6 = Q25)
```

5.4.3 Transformation langage

```r
# Tranformation des langages de programmation en 0 ou 1
ks_2021$Q7_Part_1 <- transformation_langage(ks_2021, "Q7_Part_1", "Python")
ks_2021$Q7_Part_2 <- transformation_langage(ks_2021, "Q7_Part_2", "R")
ks_2021$Q7_Part_3 <- transformation_langage(ks_2021, "Q7_Part_3", "SQL")
ks_2021$Q7_Part_4 <- transformation_langage(ks_2021, "Q7_Part_4", "C")
ks_2021$Q7_Part_5 <- transformation_langage(ks_2021, "Q7_Part_5", "C++")
ks_2021$Q7_Part_6 <- transformation_langage(ks_2021, "Q7_Part_6", "Java")
ks_2021$Q7_Part_7 <- transformation_langage(ks_2021, "Q7_Part_7", "Javascript")
ks_2021$Q7_Part_8 <- transformation_langage(ks_2021, "Q7_Part_8", "Julia")
ks_2021$Q7_Part_9 <- transformation_langage(ks_2021, "Q7_Part_9", "Swift")
ks_2021$Q7_Part_10 <- transformation_langage(ks_2021, "Q7_Part_10", "Bash")
ks_2021$Q7_Part_11 <- transformation_langage(ks_2021, "Q7_Part_11", "Matlab")


langage <- c("Q7_Part_1", "Q7_Part_2", "Q7_Part_3", "Q7_Part_4", "Q7_Part_5", "Q7_Part_6",
             "Q7_Part_7", "Q7_Part_8", "Q7_Part_9", "Q7_Part_10", "Q7_Part_11")

ks_2021$Langage <- rowSums(ks_2021[, langage]
)
ks_2021 <- ks_2021[,!(names(ks_2021) %in% langage)]
rm(langage)
```

6 Fusion des jeux de Donnes

```r
# Fusion des data frames et renommage des questions/varibales cible
ks_fusion <- rbind.fill(ks_2018, ks_2019, ks_2020, ks_2021) %>%
  select("Annee", everything()) %>% filter(Q1 != "" & Q2 != "" & Q3 != "" &
                                         Q4 != "" & Q5 != "" & Q6 != "") %>%
  rename(Age = Q1, Genre = Q2, Pays = Q3, Education = Q4, Secteur = Q5, Salaire = Q6)
```

6.1 Création des variables temporaires des continents

```r
europe <- c("France", "Germany", "Netherlands", "UK", "Austria", "Sweden", "Portugal", "Poland",
            "Ireland", "Greece", "Ukraine", "Italy", "Czech Republic", "Spain", "Hungary", "Norway",
            "Denmark", "Belgium", "Romania", "Finland", "Switzerland", "Belarus")
asia <- c("Russia", "Kazakhstan", "China", "Hong Kong", "Japan", "South Korea","Taiwan",
          "India", "Bangladesh", "Sri Lanka", "Nepal", "Indonesia", "Singapore", "Thailand",
          "Viet Nam", "Malaysia", "Philippines")
northamericas <- c("U.S.A", "Canada", "Mexico")
southamericas <- c("Chile", "Argentina", "Colombia", "Peru", "Ecuador", "Brazil")
africa <- c("Morocco", "Egypt", "Tunisia", "Algeria", "Nigeria", "Kenya", "South Africa", "Ghana",
            "Uganda", "Ethiopia")
oceania <- c("New Zealand", "Australia")
middleeast <- c("Turkey", "Israel", "Iran", "Saudi Arabia", "Iraq", "U.A.E")
```

6.2 Transformations des Donnes

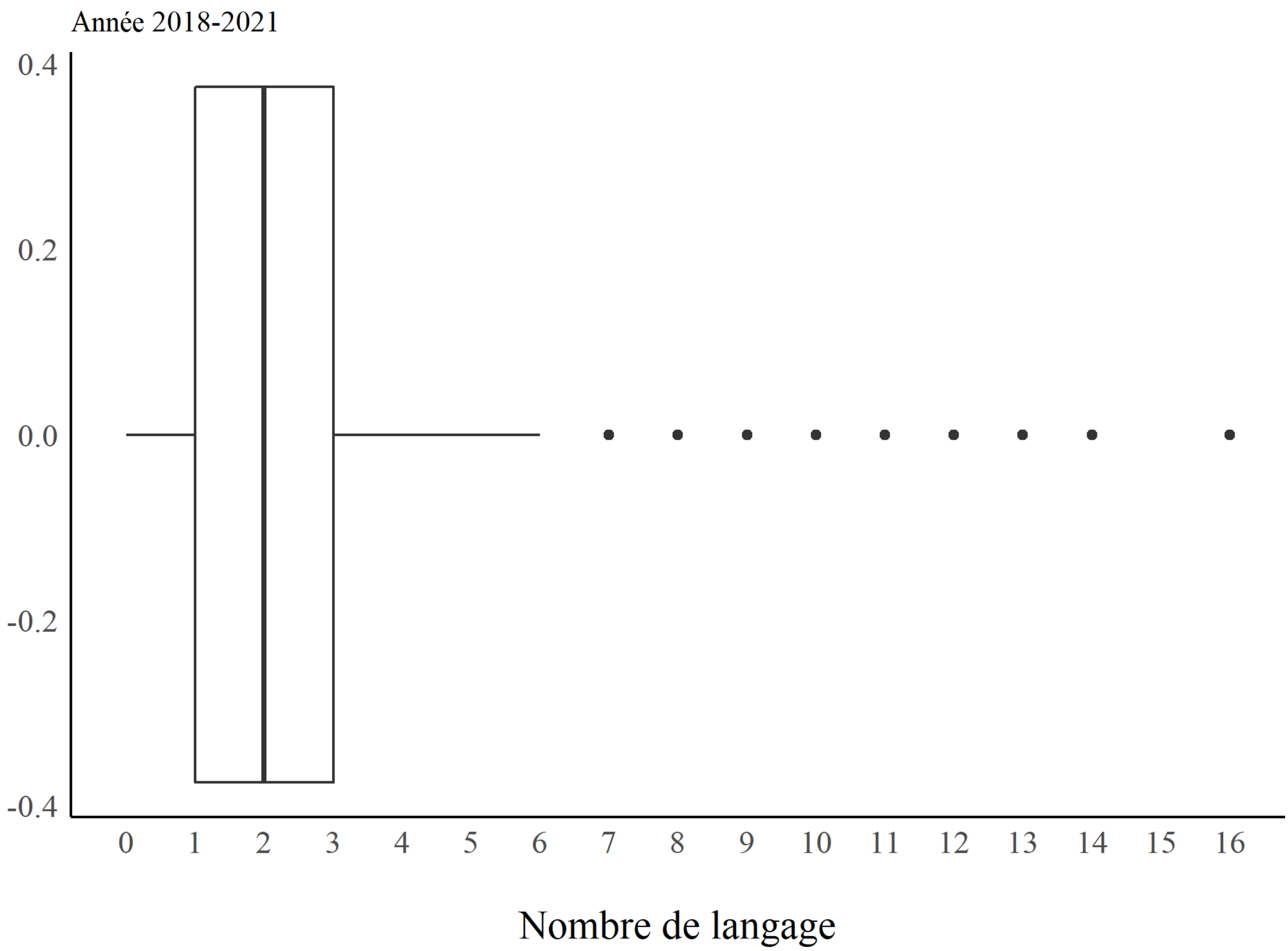
```
# Transformation des variables
ks_fusion <- ks_fusion %>% mutate(Education = case_when(Education == "Bachelor’s degree" ~ "Licence",
                                                        Education == "Master’s degree" ~ "Master",
                                                        Education == "Doctoral degree" ~ "Doctorat",
                                                        Education == "Professional degree" ~ "Doctorat pro",
                                                        Education == "Professional doctorate" ~ "Diplome pro",
                                                        TRUE ~ "Autre"),
                                  Secteur  = case_when(Secteur == "Software Engineer" |
                                                         Secteur == "Engineering (non-computer focused)"  |
                                                          Secteur == "Computer science (software engineering, etc.)" ~ "Ingenierie",
                                                         Secteur == "Mathematics or statistics" | Secteur == "Statistician" ~ "Maths/Stats",
                                                         Secteur == "Physics or astronomy" ~ "Phys. Astro.",
                                                         Secteur == "Information technology, networking, or system administration" ~ "Informatique",
                                                         Secteur == "A business discipline (accounting, economics, finance, etc.)" ~ "Business",
                                                         Secteur == "Environmental science or geology" ~ "Sc. de la Terre",
                                                         Secteur == "Medical or life sciences (biology, chemistry, medicine, etc.)" ~ "Sc. Vie/Médicale",
                                                         Secteur == "Social sciences (anthropology, psychology, sociology, etc.)" ~ "Sc. Sociales",
                                                         Secteur == "Humanities (history, literature, philosophy, etc.)" ~ "Sc. Humaines",
                                                         Secteur == "Data Scientist" | Secteur == "Data Analyst" |
                                                          Secteur == "Business Analyst" | Secteur == "Data Engineer" |
                                                          Secteur == "DBA/Database Engineer" |
                                                          Secteur == "Machine Learning Engineer" ~ "Sc. des Donnes",
                                                        Secteur == "Product/Project Manager" |
                                                        Secteur == "Program/Project Manager" |
                                                          Secteur == "Product Manager" ~ "Manageur",
                                                        TRUE ~ "Autre"),
                                 Continent = case_when(Pays %in% europe ~ "Europe",
                                                       Pays %in% asia ~ "Asie",
                                                       Pays %in% northamericas ~ "Amerique du Nord",
                                                       Pays %in% southamericas ~ "Amerique du Sud",
                                                       Pays %in% africa ~ "Afrique",
                                                       Pays %in% oceania ~ "Oceanie",
                                                       Pays %in% middleeast ~ "Moyen Orient",
                                                       TRUE ~ "Inconnu"),
                                  Genre = case_when(Genre == "Prefer not to say" | Genre == "Nonbinary" |
                                                      Genre == "Prefer to self-describe" ~ "Autres",
                                                    TRUE ~ Genre))

ks_fusion$Genre <- as_factor(ks_fusion$Genre)
```

6.3 Suppression des dataframes/variables intermédiaires

```
rm(europe, asia, middleeast, northamericas, southamericas, oceania, africa, ks_2018,
   ks_2018_question, ks_2019, ks_2019_question, ks_2020, ks_2020_question, ks_2021,
   ks_2021_question)
```

6.4 Detection outliers

```
ks_fusion %>%
  ggplot(aes(Langage)) +
  geom_boxplot() +
  scale_x_continuous(breaks = seq(0,20,1)) +
  labs(x = "\nNombre de langage",
       y = "",
       subtitle = "Année 2018-2021") +
  theme(axis.text.y = element_text(angle = 0),
        plot.caption = element_text(size = 10),
        legend.position = c(.85, .86),
        legend.background = element_rect(fill = "transparent"))
```



6.5 Suppression outliers

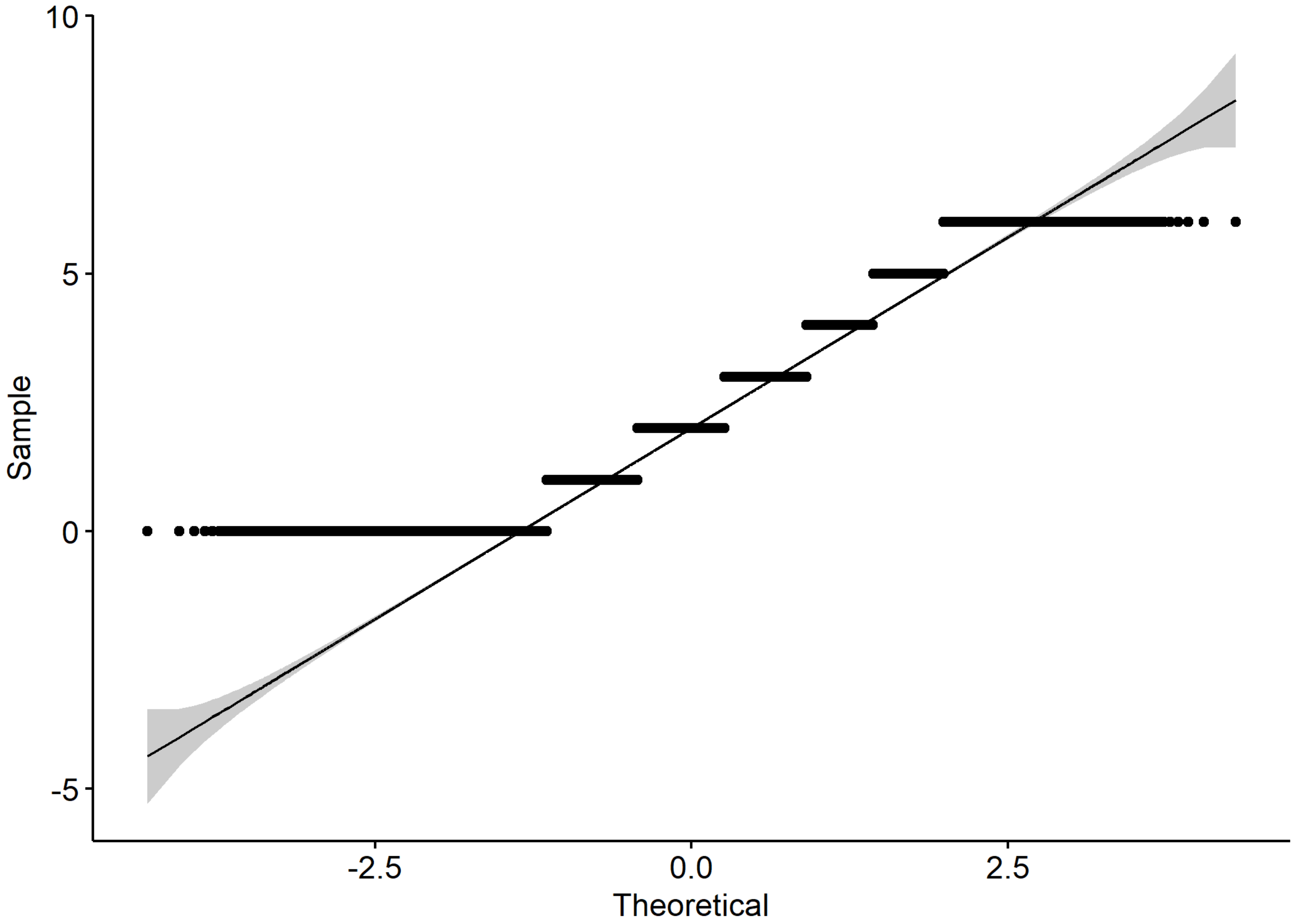
```
ks_fusion <- ks_fusion %>% filter(Langage < 7)
```

6.6 Vérification de la normalité de la variable dépendante

```
ad.test(ks_fusion$Langage)
```

```
##
##  Anderson-Darling normality test
##
## data:  ks_fusion$Langage
## A = 1283.5, p-value < 2.2e-16
```

```
ggqqplot(ks_fusion$Langage)
```



6.7 Summary du dataframe ks_fusion

```
print(dfSummary(ks_fusion,
          varnumbers   = FALSE,
          valid.col    = FALSE,
          graph.magnif = 0.76),
          method = 'render')
```

Data Frame Summary

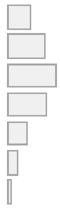
ks_fusion

Dimensions: 57605 x 9

Duplicates: 18344

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Annee [numeric]	Mean (sd) : 2019.4 (1.2) min ≤ med ≤ max: 2018 ≤ 2019 ≤ 2021 IQR (CV) : 3 (0)	2018 : 19434 (33.7%) 2019 : 12361 (21.5%) 2020 : 10615 (18.4%) 2021 : 15195 (26.4%)		0 (0.0%)
Genre [factor]	1. Homme 2. Femme 3. Autres	47571 (82.6%) 9117 (15.8%) 917 (1.6%)		0 (0.0%)

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Age [character]	1. 18-29 2. 30-39 3. 40-49 4. 50-59 5. 60-69 6. 70+	26922 (46.7%) 17249 (29.9%) 8200 (14.2%) 3739 (6.5%) 1221 (2.1%) 274 (0.5%)		0 (0.0%)
Pays [character]	1. India 2. U.S.A 3. Inconnu 4. Brazil 5. China 6. Russia 7. Japan 8. UK 9. Germany 10. Spain [58 others]	11457 (19.9%) 9460 (16.4%) 3425 (5.9%) 2132 (3.7%) 2080 (3.6%) 1986 (3.4%) 1975 (3.4%) 1632 (2.8%) 1546 (2.7%) 1283 (2.2%) 20629 (35.8%)		0 (0.0%)
Education [character]	1. Autre 2. Diplome pro 3. Doctorat 4. Doctorat pro 5. Licence 6. Master	3657 (6.3%) 283 (0.5%) 8915 (15.5%) 1408 (2.4%) 16791 (29.1%) 26551 (46.1%)		0 (0.0%)
Secteur [character]	1. Sc. des Donnes 2. Ingenierie 3. Autre 4. Maths/Stats 5. Manageur 6. Business 7. Phys. Astro. 8. Informatique 9. Sc. Vie/Médicale 10. Sc. Sociales [2 others]	20223 (35.1%) 16996 (29.5%) 9681 (16.8%) 3338 (5.8%) 2293 (4.0%) 1551 (2.7%) 967 (1.7%) 859 (1.5%) 763 (1.3%) 476 (0.8%) 458 (0.8%)		0 (0.0%)

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Salaire [character]	1. 0-10K 2. 100K-250K 3. 10K-20K 4. Inconnu 5. 20K-30K 6. 30K-40K 7. 40K-50K 8. 50K-60K 9. 60K-70K 10. 70K-80K [4 others]	19521 (33.9%) 6728 (11.7%) 5831 (10.1%) 4555 (7.9%) 4119 (7.2%) 3066 (5.3%) 2858 (5.0%) 2783 (4.8%) 2235 (3.9%) 2025 (3.5%) 3884 (6.7%)		0 (0.0%)
Langage [numeric]	Mean (sd) : 2.2 (1.5) min ≤ med ≤ max: 0 ≤ 2 ≤ 6 IQR (CV) : 2 (0.7)	0 : 7265 (12.6%) 1 : 11966 (20.8%) 2 : 15425 (26.8%) 3 : 12421 (21.6%) 4 : 6160 (10.7%) 5 : 3021 (5.2%) 6 : 1347 (2.3%)		0 (0.0%)
Continent [character]	1. Afrique 2. Amerique du Nord 3. Amerique du Sud 4. Asie 5. Europe 6. Inconnu 7. Moyen Orient 8. Oceanie	2884 (5.0%) 11319 (19.6%) 3641 (6.3%) 21430 (37.2%) 11309 (19.6%) 4252 (7.4%) 1843 (3.2%) 927 (1.6%)		0 (0.0%)

Generated by summarytools 1.0.1 (R version 4.0.1)
2022-09-07

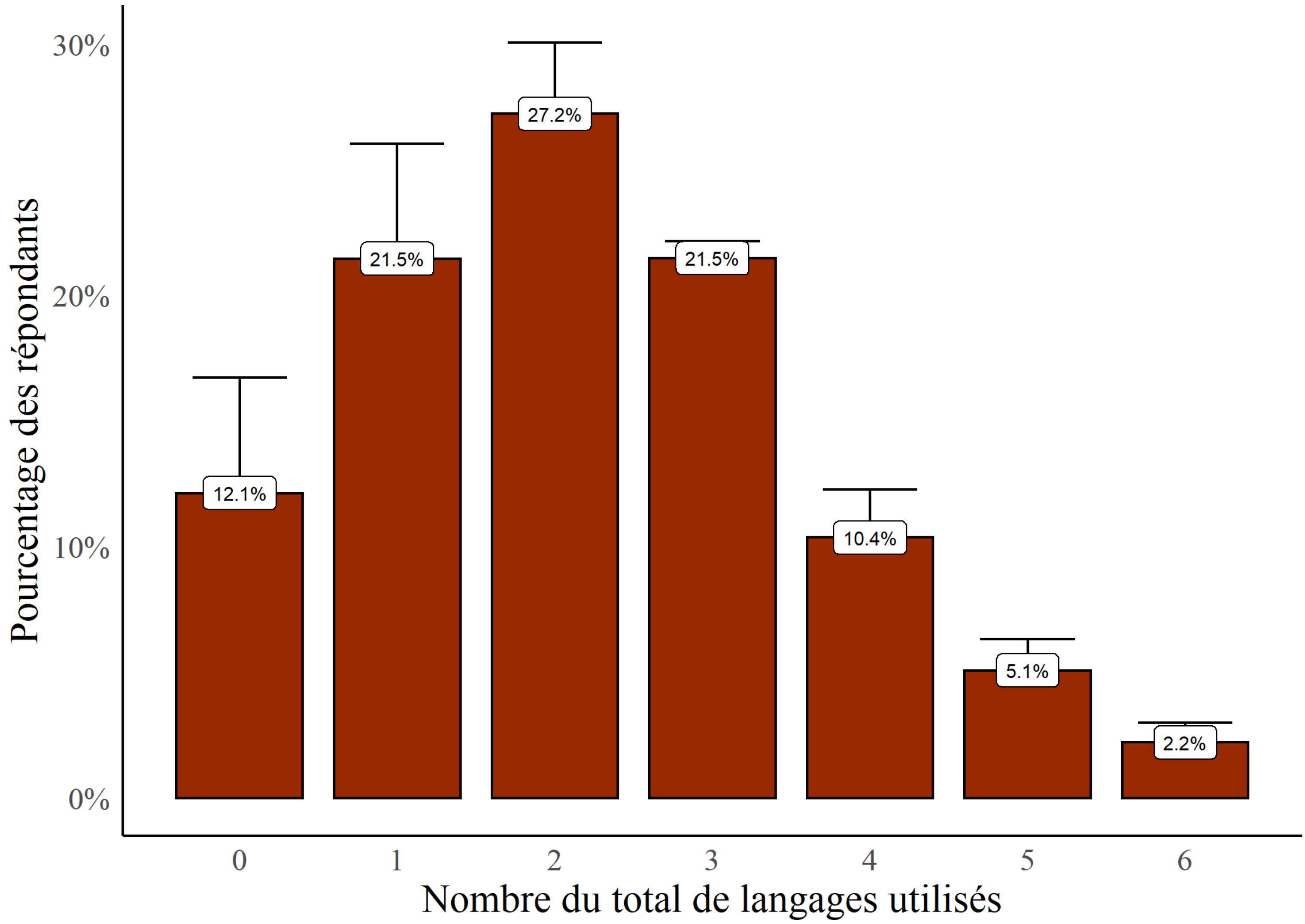
7 EDA

7.1 Graphiques

7.1.1 EDA simple

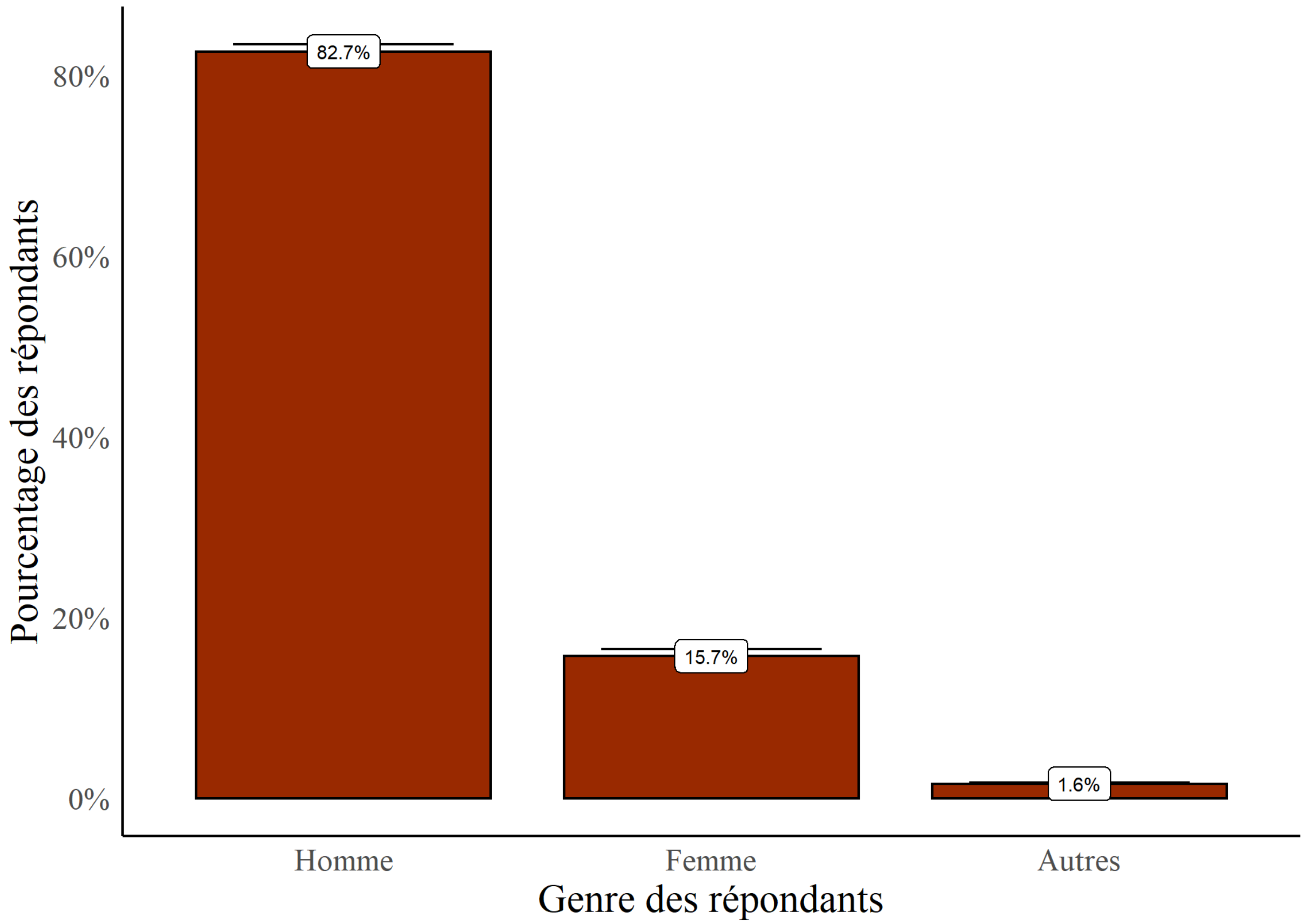
7.1.2 Distribution Langage avec écart type

```r
# Distribution de la variable cible (Langage) avec écart-type
temp_count_annee <- ks_fusion %>% count(Annee) %>% rename(total_annee = n)

temp_count_langage_annee <- ks_fusion %>% count(Annee, Langage) %>% rename(total_langage_annee = n)

temp_count_full <- full_join(temp_count_annee, temp_count_langage_annee, by = "Annee") %>%
  mutate(freq = total_langage_annee / total_annee,
         mean = mean(freq, na.rm = TRUE),
         sd = sd(freq, na.rm = TRUE))

temp_count_full_test <- ks_fusion %>% count(Langage) %>% rename(Count = n)

temp_count <- full_join(temp_count_full, temp_count_full_test, by = "Langage")

temp_count %>%
  ddply(~Langage, summarise, mean = mean(freq, na.rm = TRUE), sd = sd(freq, na.rm = TRUE)) %>%
  ggplot(aes(Langage, mean)) +
  geom_col(fill = my_color[5], color = "black", width = 0.8) +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.6) +
  geom_label(aes(label = label_percent(accuracy = 0.1)(mean)), size = 2.5) +
  scale_x_continuous(breaks = seq(0, 20, 1)) +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Nombre du total de langages utilisés",
     y = "Pourcentage des répondants") +
  theme(axis.text.y = element_text(angle = 0),
      plot.caption = element_text(size = 10))
```



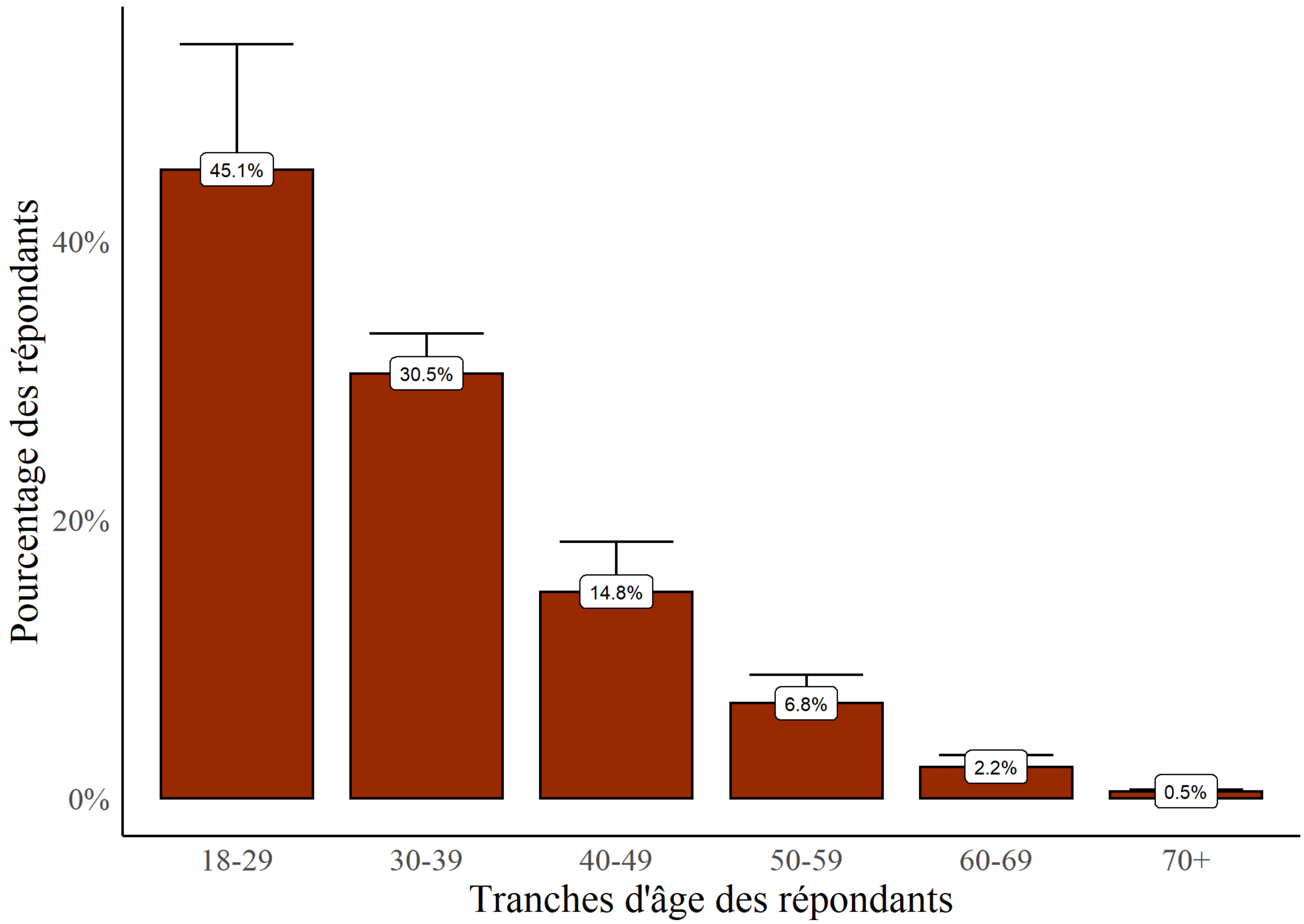
7.1.3 Distribution Genre avec écart type

```r
# Distribution de la variable Genre avec écart-type
temp_count_annee <- ks_fusion %>% count(Annee) %>% rename(total_annee = n)

temp_count_Genre_annee <- ks_fusion %>% count(Annee, Genre) %>% rename(total_Genre_annee = n)
temp_count_full <- full_join(temp_count_annee, temp_count_Genre_annee, by = "Annee") %>%
  mutate(freq = total_Genre_annee / total_annee)
temp_count_full_test <- ks_fusion %>% count(Genre) %>% rename(Count = n)
temp_count <- full_join(temp_count_full, temp_count_full_test, by = "Genre")


temp_count %>%
  ddply(~Genre, summarise, mean = mean(freq, na.rm = TRUE), sd = sd(freq, na.rm = TRUE)) %>%
  ggplot(aes(Genre, mean))
+
  geom_col(fill = my_color[5], color = "black", width = 0.8) +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.6) +
  geom_label(aes(label = label_percent(accuracy = 0.1)(mean)), size = 2.5) +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Genre des répondants",
       y = "Pourcentage des répondants") +
  theme(axis.text.y = element_text(angle = 0),
        plot.caption = element_text(size = 10))
```



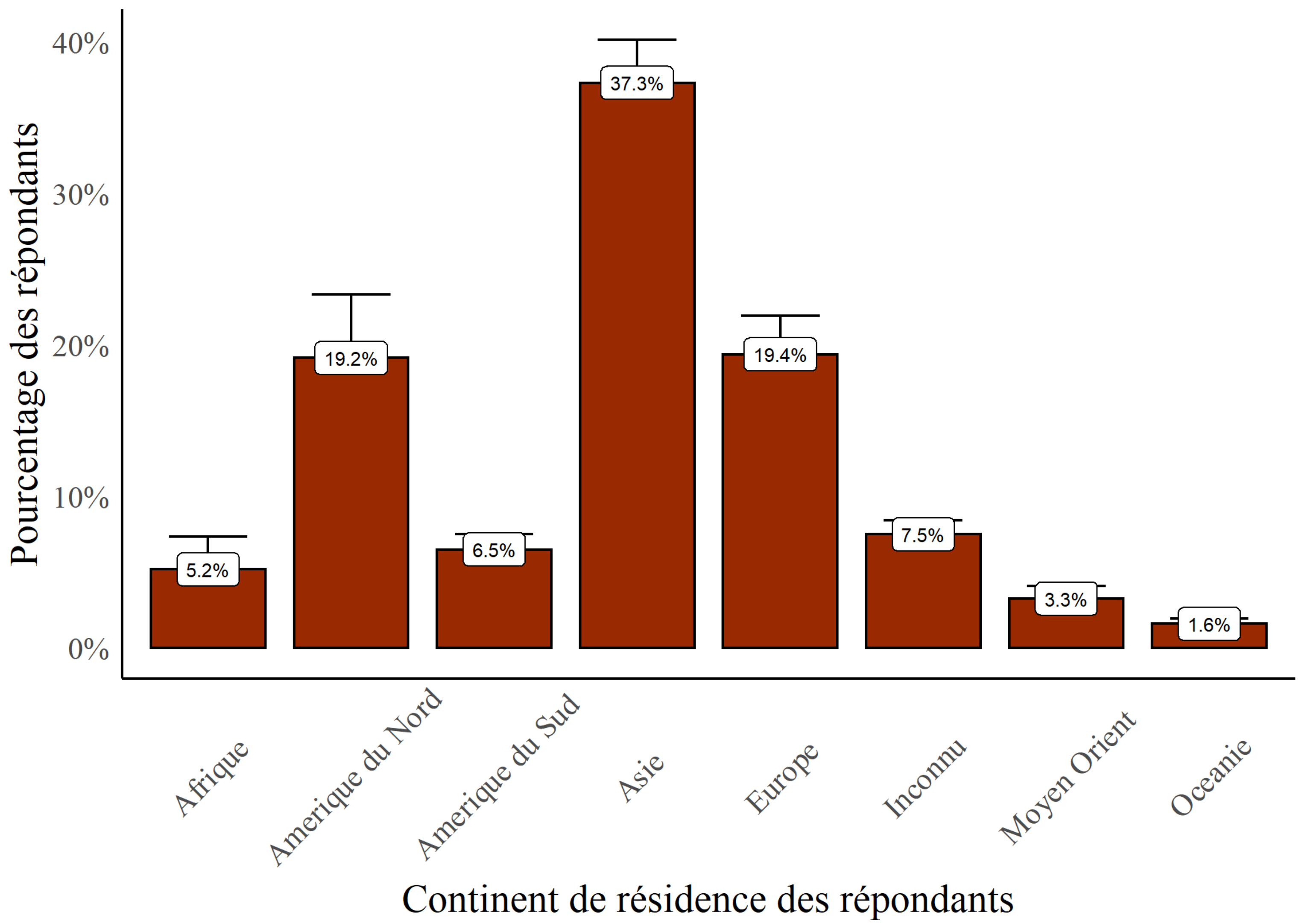
7.1.4 Distribution Age avec écart type

```r
# Distribution de la variable Age avec écart-type
temp_count_annee <- ks_fusion %>% count(Annee) %>% rename(total_annee = n)

temp_count_Age_annee <- ks_fusion %>% count(Annee, Age) %>% rename(total_Age_annee = n)
temp_count_full <- full_join(temp_count_annee, temp_count_Age_annee, by = "Annee") %>%
  mutate(freq = total_Age_annee / total_annee)
temp_count_full_test <- ks_fusion %>% count(Age) %>% rename(Count = n)
temp_count <- full_join(temp_count_full, temp_count_full_test, by = "Age")


temp_count %>%
  ddply(~Age, summarise, mean = mean(freq, na.rm = TRUE), sd = sd(freq, na.rm = TRUE)) %>%
  ggplot(aes(Age, mean))
+
  geom_col(fill = my_color[5], color = "black", width = 0.8) +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.6) +
  geom_label(aes(label = label_percent(accuracy = 0.1)(mean)), size = 2.5) +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Tranches d'âge des répondants",
     y = "Pourcentage des répondants") +
  theme(axis.text.y = element_text(angle = 0),
      plot.caption = element_text(size = 10))
```



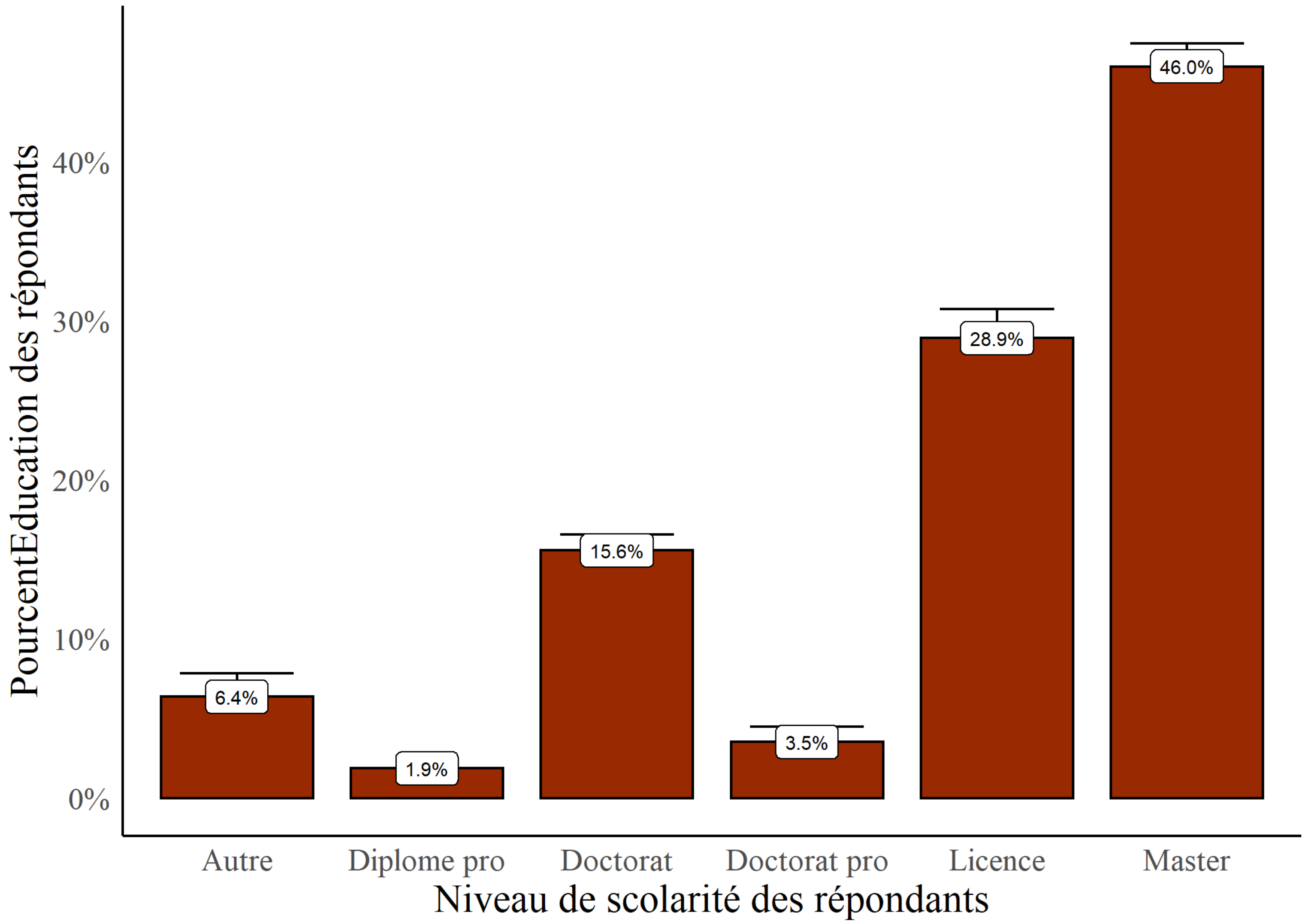
7.1.5 Distribution Continent avec écart type

```r
# Distribution de la variable Continent avec écart-type
temp_count_annee <- ks_fusion %>% count(Annee) %>% rename(total_annee = n)

temp_count_Continent_annee <- ks_fusion %>% count(Annee, Continent) %>% rename(total_Continent_annee = n)
temp_count_full <- full_join(temp_count_annee, temp_count_Continent_annee, by = "Annee") %>%
  mutate(freq = total_Continent_annee / total_annee)
temp_count_full_test <- ks_fusion %>% count(Continent) %>% rename(Count = n)
temp_count <- full_join(temp_count_full, temp_count_full_test, by = "Continent")


temp_count %>%
  ddply(~Continent, summarise, mean = mean(freq, na.rm = TRUE), sd = sd(freq, na.rm = TRUE)) %>%
  ggplot(aes(Continent, mean))
+
  geom_col(fill = my_color[5], color = "black", width = 0.8) +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.55) +
  geom_label(aes(label = label_percent(accuracy = 0.1)(mean)), size = 2.5) +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Continent de résidence des répondants",
      y = "Pourcentage des répondants") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.55),
       axis.text.y = element_text(angle = 0),
       plot.caption = element_text(size = 10))
```



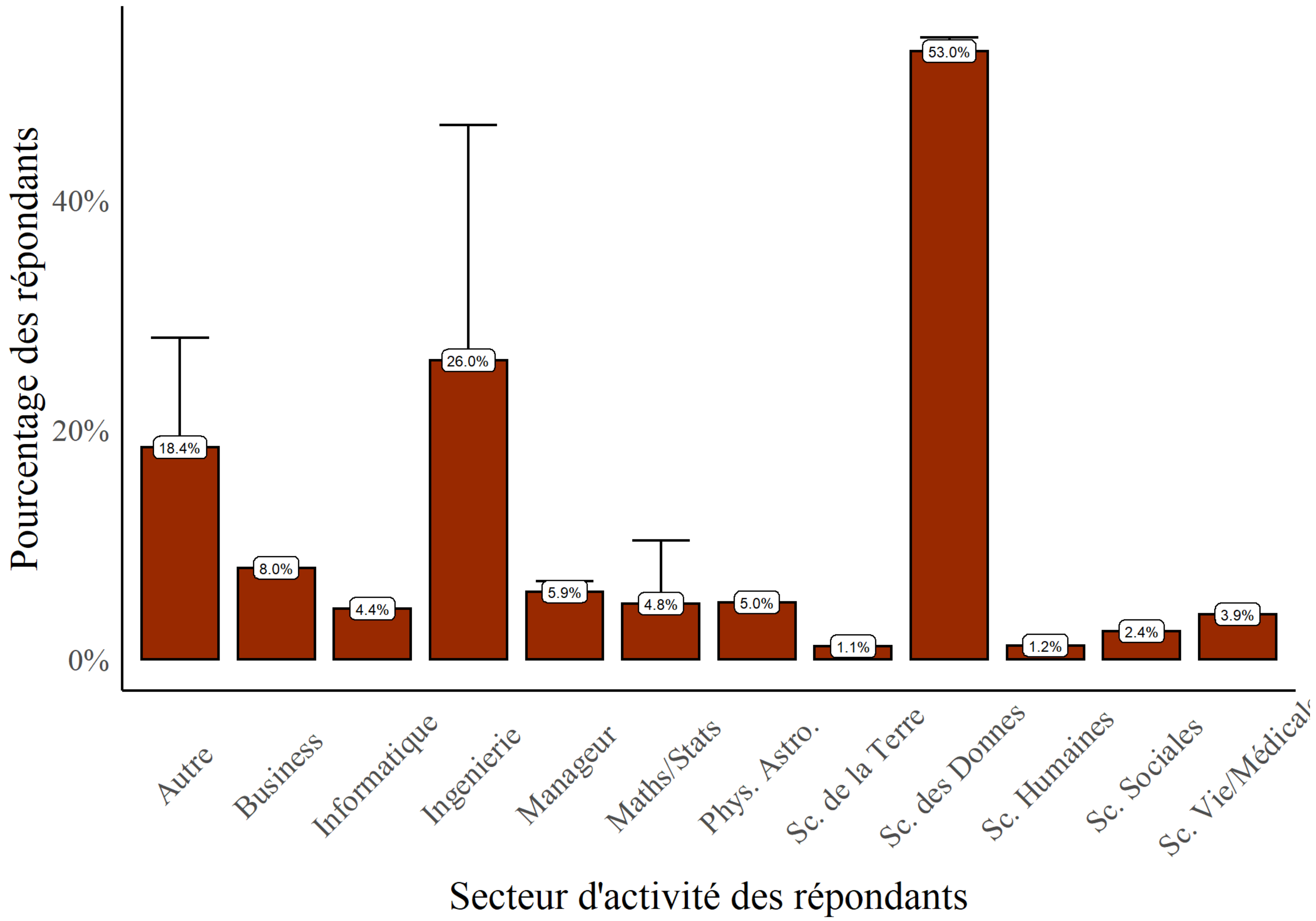
7.1.6 Distribution Education avec écart type

```
# Distribution de la variable Education avec écart-type
temp_count_annee <- ks_fusion %>% count(Annee) %>% rename(total_annee = n)

temp_count_Education_annee <- ks_fusion %>% count(Annee, Education) %>% rename(total_Education_annee = n)
temp_count_full <- full_join(temp_count_annee, temp_count_Education_annee, by = "Annee") %>%
  mutate(freq = total_Education_annee / total_annee)
temp_count_full_test <- ks_fusion %>% count(Education) %>% rename(Count = n)
temp_count <- full_join(temp_count_full, temp_count_full_test, by = "Education")


temp_count %>%
  ddply(~Education, summarise, mean = mean(freq, na.rm = TRUE), sd = sd(freq, na.rm = TRUE)) %>%
  ggplot(aes(Education, mean)) +
  geom_col(fill = my_color[5], color = "black", width = 0.8) +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.6) +
  geom_label(aes(label = label_percent(accuracy = 0.1)(mean)), size = 2.5) +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Niveau de scolarité des répondants",
      y = "PourcentEducation des répondants") +
  theme(axis.text.y = element_text(angle = 0),
       plot.caption = element_text(size = 10))
```



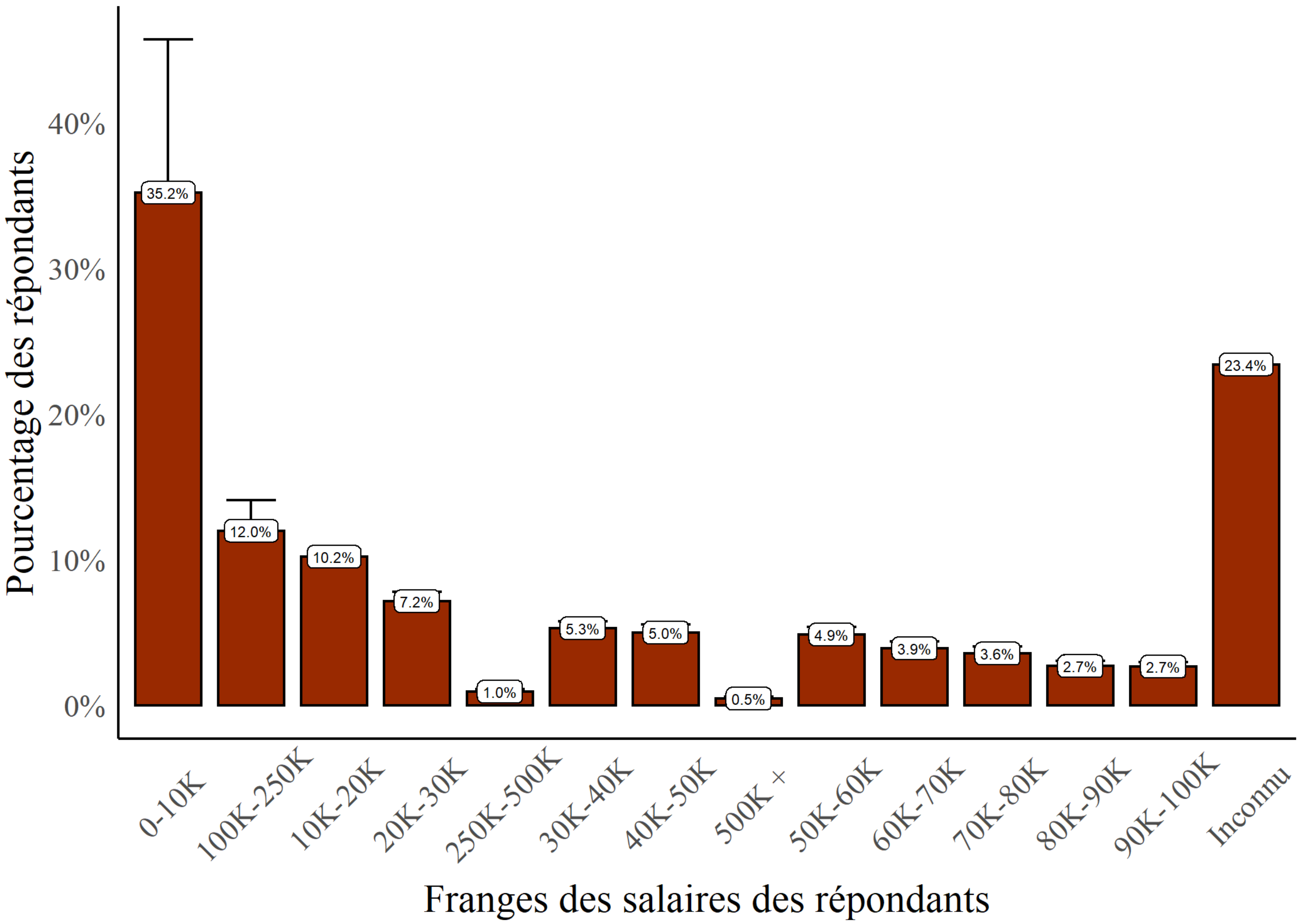
7.1.7 Distribution Secteur avec écart type

```r
# Distribution de la variable Secteur avec écart-type
temp_count_annee <- ks_fusion %>% count(Annee) %>% rename(total_annee = n)

temp_count_Secteur_annee <- ks_fusion %>% count(Annee, Secteur) %>% rename(total_Secteur_annee = n)
temp_count_full <- full_join(temp_count_annee, temp_count_Secteur_annee, by = "Annee") %>%
  mutate(freq = total_Secteur_annee / total_annee)
temp_count_full_test <- ks_fusion %>% count(Secteur) %>% rename(Count = n)
temp_count <- full_join(temp_count_full, temp_count_full_test, by = "Secteur")


temp_count %>%
  ddply(~Secteur, summarise, mean = mean(freq, na.rm = TRUE), sd = sd(freq, na.rm = TRUE)) %>%
  ggplot(aes(Secteur, mean)) +
  geom_col(fill = my_color[5], color = "black", width = 0.8) +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.6) +
  geom_label(aes(label = label_percent(accuracy = 0.1)(mean)), size = 2,
             label.padding = unit(0.15, "lines")) +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Secteur d'activité des répondants",
       y = "Pourcentage des répondants") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.6),
        axis.text.y = element_text(angle = 0),
        plot.caption = element_text(size = 10))
```



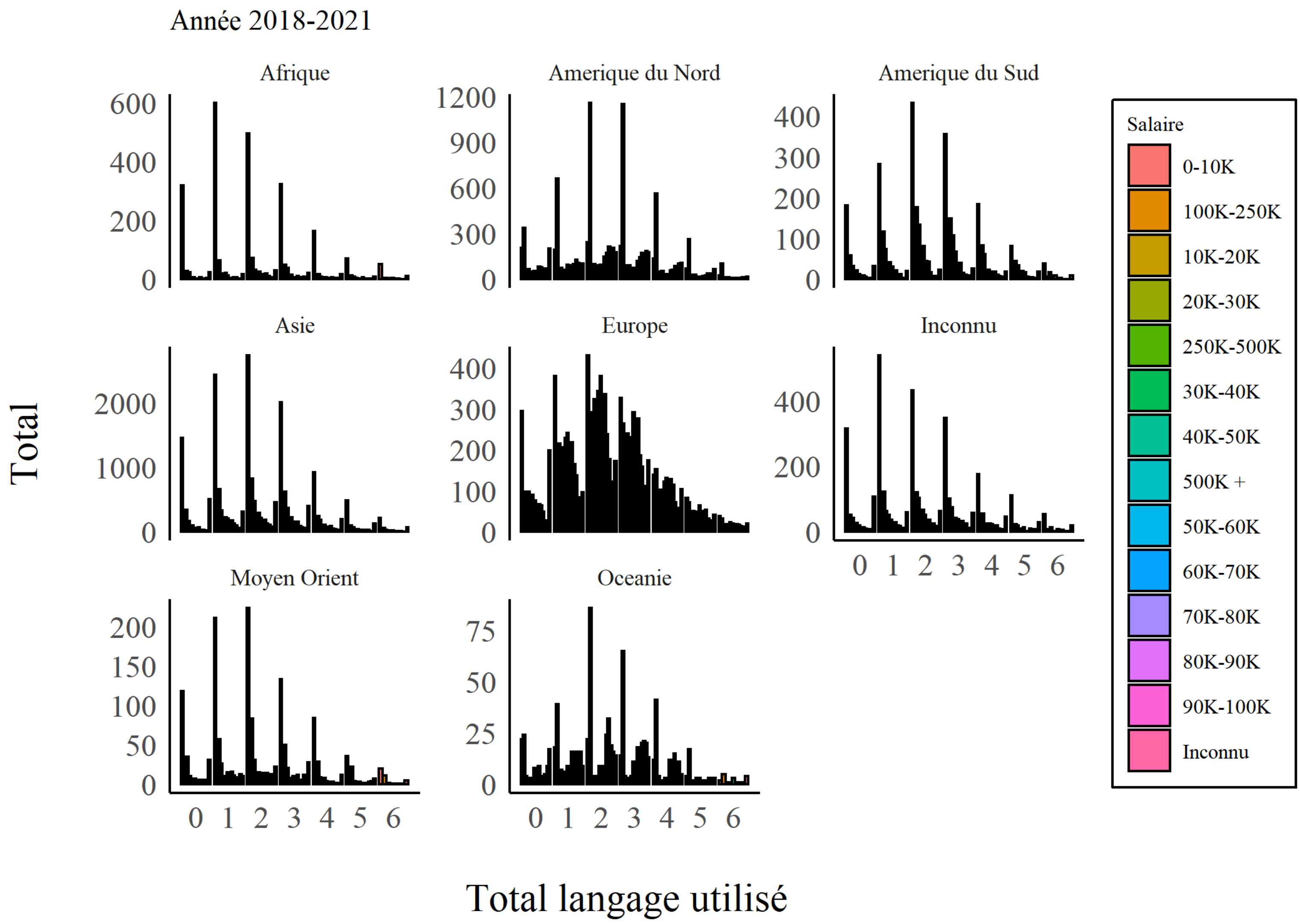
7.1.8 Distribution Salaire avec écart type

```r
# Distribution de la variable Salaire avec écart-type
temp_count_annee <- ks_fusion %>% count(Annee) %>% rename(total_annee = n)

temp_count_Salaire_annee <- ks_fusion %>% count(Annee, Salaire) %>% rename(total_Salaire_annee = n)
temp_count_full <- full_join(temp_count_annee, temp_count_Salaire_annee, by = "Annee") %>%
  mutate(freq = total_Salaire_annee / total_annee)
temp_count_full_test <- ks_fusion %>% count(Salaire) %>% rename(Count = n)
temp_count <- full_join(temp_count_full, temp_count_full_test, by = "Salaire")


temp_count %>%
  ddply(~Salaire, summarise, mean = mean(freq, na.rm = TRUE), sd = sd(freq, na.rm = TRUE)) %>%
  ggplot(aes(Salaire, mean)) +
  geom_col(fill = my_color[5], color = "black", width = 0.8) +
  geom_errorbar(aes(ymin = mean, ymax = mean + sd), width = 0.6) +
  geom_label(aes(label = label_percent(accuracy = 0.1)(mean)), size = 2,
             label.padding = unit(0.15, "lines")) +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Franges des salaires des répondants",
       y = "Pourcentage des répondants") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.6),
        axis.text.y = element_text(angle = 0),
        plot.caption = element_text(size = 10))
```

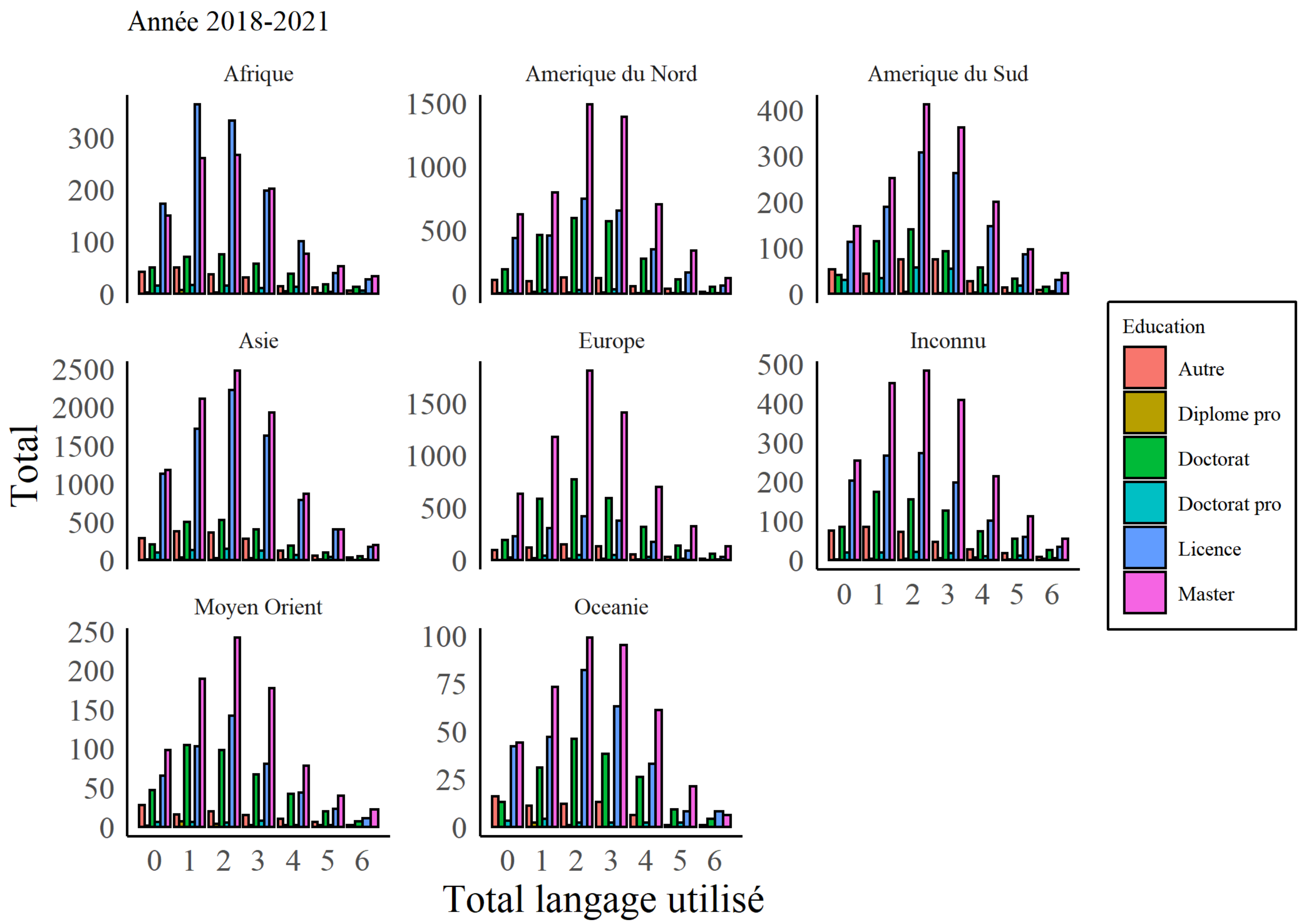


7.1.9 EDA complexe

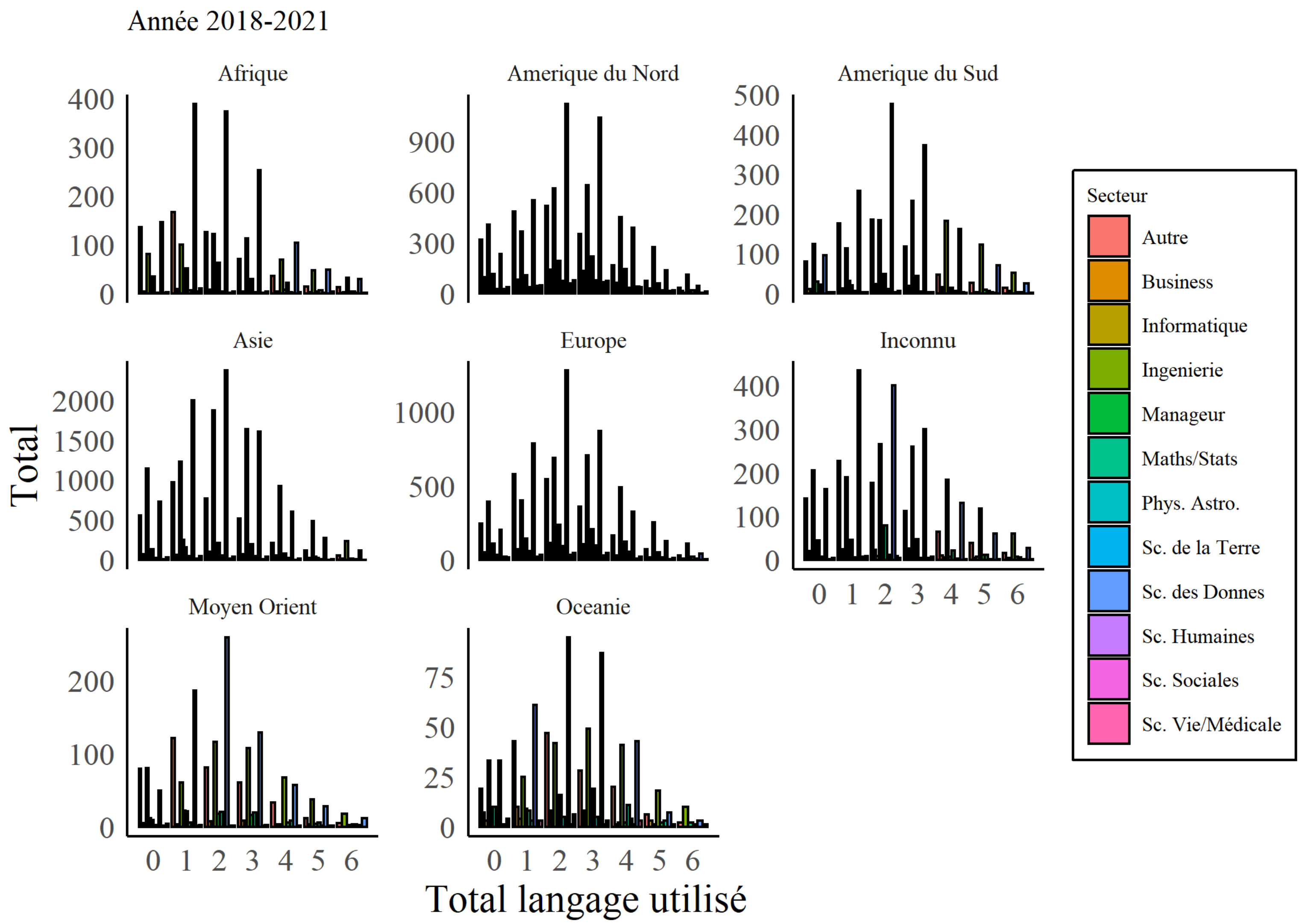
```r
# Permet de visualiser des combinaisons de variables
eda_complexe(ks_fusion, Langage, Salaire, Continent) +
  scale_x_continuous(breaks = seq(0, 20, 1)) +
  labs(x = "\nTotal langage utilisé",
       y = "Total\n",
       subtitle = "Année 2018-2021") +
  theme(axis.text.y = element_text(angle = 0),
        plot.caption = element_text(size = 10),
        legend.background = element_rect(fill = "transparent"),
        legend.title = element_text(size = 8),
        legend.text = element_text(size = 8),
        legend.position = "right")
```




```r
eda_complexe(ks_fusion, Langage, Education, Continent) +
  scale_x_continuous(breaks = seq(0, 20, 1)) +
  labs(x = "Total langage utilisé",
       y = "Total",
       subtitle = "Année 2018-2021") +
  theme(axis.text.y = element_text(angle = 0),
        plot.caption = element_text(size = 10),
        legend.background = element_rect(fill = "transparent"),
        legend.title = element_text(size = 8),
        legend.text = element_text(size = 8),
        legend.position = "right")
```

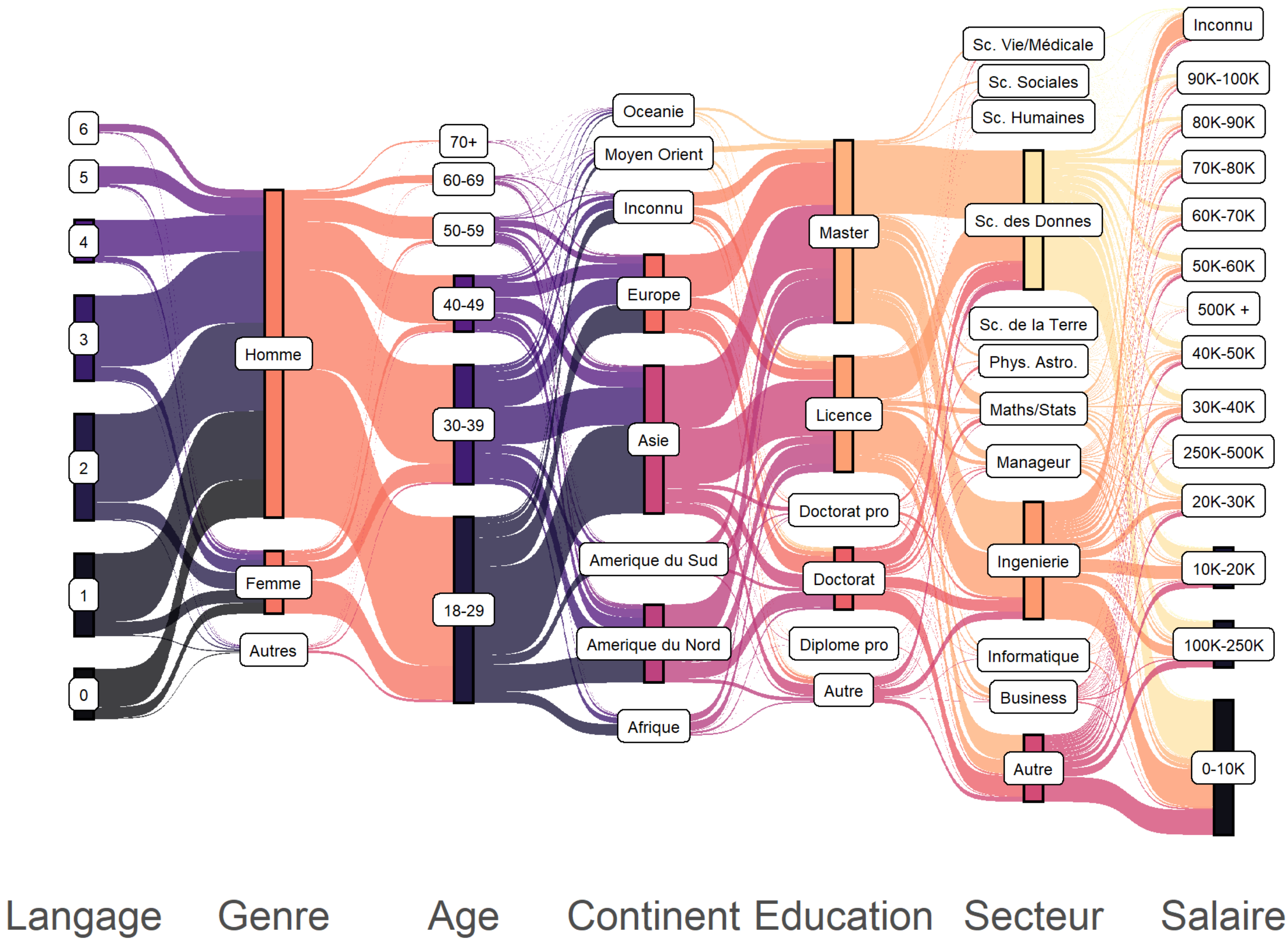



```r
eda_complexe(ks_fusion, Langage, Secteur, Continent) +
  scale_x_continuous(breaks = seq(0, 20, 1)) +
  labs(x = "Total langage utilisé",
       y = "Total",
       subtitle = "Année 2018-2021") +
  theme(axis.text.y = element_text(angle = 0),
        plot.caption = element_text(size = 10),
        legend.background = element_rect(fill = "transparent"),
        legend.title = element_text(size = 8),
        legend.text = element_text(size = 8),
        legend.key.size = unit(0.6, "cm"))
```



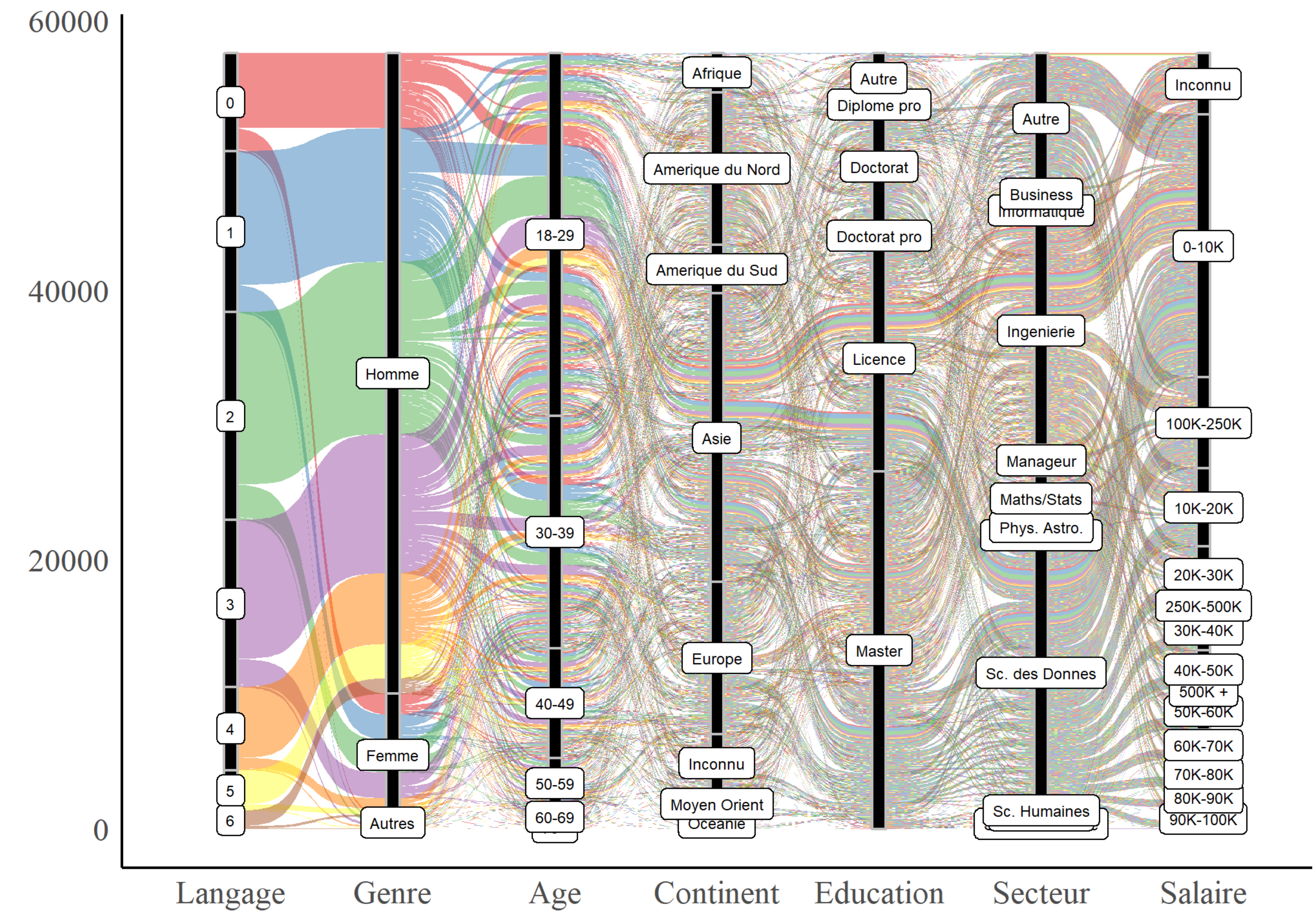
7.2 Sankey

```r
# Permet de voir la distribution entre toutes les variables utilisées
sankey <- ks_fusion %>% make_long(Langage, Genre, Age, Continent, Education, Secteur, Salaire)

ggplot(sankey, aes(x = x,
                   next_x = next_x,
                   node = node,
                   next_node = next_node,
                   fill = factor(node),
                   label = node)) +
  geom_sankey(flow.alpha = 0.75, node.color = 1) +
  geom_sankey_label(size = 2, color = 1, fill = "white") +
  scale_fill_viridis_d(option = "A", alpha = 0.95) +
  theme_sankey(base_size = 18) +
  labs(x = NULL) +
  theme(legend.position = "none",
      plot.title = element_text(hjust = .5))
```



7.3 Graphique alluvial

```
# Permet de voir le cheminement de la variable cible (Langage) A TRAVERS l'ensemble des autres variables
ks_fusion_copy <- ks_fusion
ks_fusion_copy$Langage <- as_factor(ks_fusion_copy$Langage)

sankey <- ks_fusion_copy %>%
  select(Langage, Genre, Age, Continent, Education, Secteur, Salaire) %>%
  drop_na() %>%
  group_by(Langage, Genre, Age,  Continent, Education, Secteur, Salaire) %>%
  summarize(Count = n()) %>%
  ungroup()


ggplot(sankey, aes(y = Count, axis1 = Langage, axis2 = Genre, axis3 = Age, axis4 = Continent,
                axis5 = Education, axis6 = Secteur, axis7 = Salaire)) +
  geom_alluvium(aes(fill = Langage), curve_type = "sigmoid", width = 1/12) +
  geom_stratum(width = 1/12, fill = "black", color = "grey") +
  geom_label(stat = "stratum", size = 2, aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("Langage", "Genre", "Age", "Continent", "Education", "Secteur", "Salaire"),
                expand = c(.11, .01)) +
  scale_fill_brewer(palette = "Set1") +
  theme(legend.position = "none") +
  labs(x = "",
       y = "",
       fill = "")
```




```
rm(ks_fusion_copy)
```

7.4 Suppression des variables temporaires

```
rm(temp_count, temp_count_Age_annee, temp_count_annee, temp_count_Continent_annee, temp_count_Education_annee, temp_count_full, temp_cou
```

8 Test statistiques

8.1 Langage/Age

8.1.1 Data frame temporaire

```
temp_la <- ks_fusion %>% select(Age, Langage)
```

8.1.2 Test statistiques

```
# Table résumant des Donnes statistiques (moyenne, deviation standard, etc.)
crosstable(temp_la, c(Age, Langage),
        by = Age, total = "both",
        percent_pattern = "{n} ({p_col})",
        showNA = "ifany") %>%
  as_flextable()
```

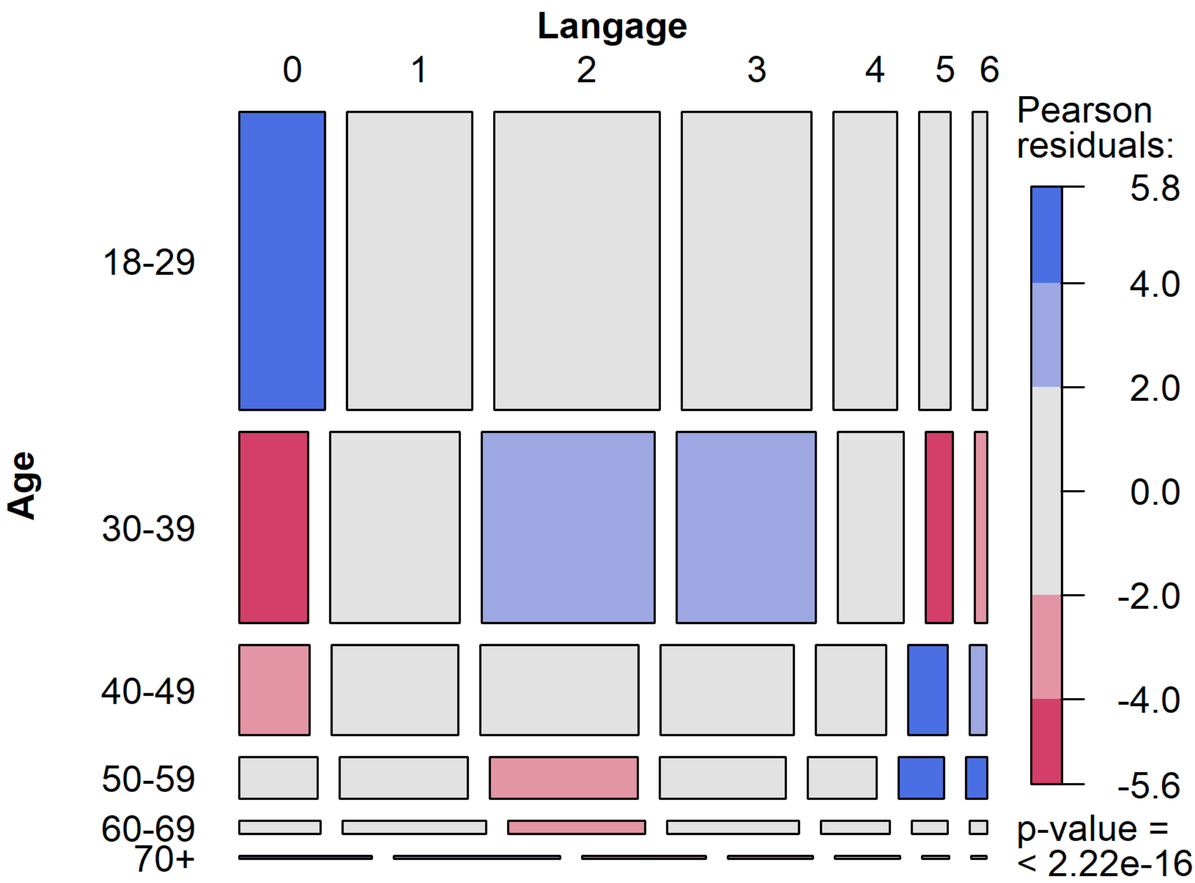
```r
# Table de contingence
temp_la %>%
  sjPlot::sjtab(fun = "xtab", var.labels = c("Age", "Langage"), show.row.prc = TRUE,
          show.col.prc = TRUE, show.summary = TRUE, show.legend = TRUE)
```

Age	*Langage* 0	1	2	3	4	5	6	***Total***
18-29	3727 13.8 % 51.3 %	5486 20.4 % 45.8 %	7247 26.9 % 47 %	5674 21.1 % 45.7 %	2796 10.4 % 45.4 %	1372 5.1 % 45.4 %	620 2.3 % 46 %	26922 100 % 46.7 %
30-39	1912 11.1 % 26.3 %	3650 21.2 % 30.5 %	4848 28.1 % 31.4 %	3901 22.6 % 31.4 %	1845 10.7 % 30 %	762 4.4 % 25.2 %	331 1.9 % 24.6 %	17249 100 % 29.9 %
40-49	932 11.4 % 12.8 %	1692 20.6 % 14.1 %	2105 25.7 % 13.6 %	1781 21.7 % 14.3 %	935 11.4 % 15.2 %	526 6.4 % 17.4 %	229 2.8 % 17 %	8200 100 % 14.2 %
50-59	474 12.7 % 6.5 %	779 20.8 % 6.5 %	899 24 % 5.8 %	765 20.5 % 6.2 %	418 11.2 % 6.8 %	278 7.4 % 9.2 %	126 3.4 % 9.4 %	3739 100 % 6.5 %
60-69	161 13.2 % 2.2 %	285 23.3 % 2.4 %	271 22.2 % 1.8 %	262 21.5 % 2.1 %	137 11.2 % 2.2 %	71 5.8 % 2.4 %	34 2.8 % 2.5 %	1221 100 % 2.1 %
70+	59 21.5 % 0.8 %	74 27 % 0.6 %	55 20.1 % 0.4 %	38 13.9 % 0.3 %	29 10.6 % 0.5 %	12 4.4 % 0.4 %	7 2.6 % 0.5 %	274 100 % 0.5 %
Total	7265 12.6 % 100 %	11966 20.8 % 100 %	15425 26.8 % 100 %	12421 21.6 % 100 %	6160 10.7 % 100 %	3021 5.2 % 100 %	1347 2.3 % 100 %	57605 100 % 100 %

χ^2=292.104 · df=30 · Cramer's V=0.032 · p=0.000

observed values
% within Age
% within Langage

```r
# Test chi2 et V de Cramer
chisq_la <- chisq.test(table(temp_la$Age, temp_la$Langage))
chisq_lav <- questionr::cramer.v(table(temp_la$Age, temp_la$Langage))

# Mosaic plot
mosaic_la <- mosaic(~ Age + Langage, data = temp_la, spacing = spacing_equal(c(0.5, 0.5)),
          shade = TRUE, legend = TRUE,
          labeling = labeling_border(rot_labels=c(0,0,0,0),
                         just_labels=c("left","right"),
                         offset_varnames = c(0, 0, 0, 3)),
          margins = c(0, 0, 0, 3))
```




```
# GLM
mod_la <- glm(Langage ~ Age, temp_la, family = "poisson")
anova(mod_la)
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	57604	66484.35
Age	5	83.13416	57599	66401.22

```
summary(mod_la)
```

```
##
## Call:
## glm(formula = Langage ~ Age, family = "poisson", data = temp_la)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1560  -0.8969  -0.1255   0.5231   2.3815
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.780620   0.004125 189.235  < 2e-16 ***
## Age30-39     0.015116   0.006571   2.300  0.02142 *
## Age40-49     0.059106   0.008347   7.081 1.43e-12 ***
## Age50-59     0.062735   0.011493   5.458 4.80e-08 ***
## Age60-69     0.020342   0.019613   1.037  0.29965
## Age70+      -0.147641   0.044215  -3.339  0.00084 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66484  on 57604  degrees of freedom
## Residual deviance: 66401  on 57599  degrees of freedom
## AIC: 203382
##
## Number of Fisher Scoring iterations: 5
```

8.2 Langage/Genre

8.2.1 Data frame temporaire

```
temp_lg <- ks_fusion %>% select(Genre, Langage)
```

8.2.2 Test statistiques

```
# Table résumant des Donnes statistiques (moyenne, deviation standard, etc.)
crosstable(temp_lg, c(Genre, Langage),
         by = Genre, total = "both",
         percent_pattern = "{n} ({p_col})",
         showNA = "ifany") %>%
  as_flextable()
```

```
# Table de contingence
temp_lg %>%
  sjPlot::sjtab(fun = "xtab", var.labels = c("Genre", "Langage"), show.row.prc = TRUE,
             show.col.prc = TRUE, show.summary = TRUE, show.legend = TRUE)
```

Genre	*Langage*							***Total***
	0	1	2	3	4	5	6	
Homme	5565 11.7 % 76.6 %	9924 20.9 % 82.9 %	12802 26.9 % 83 %	10329 21.7 % 83.2 %	5255 11 % 85.3 %	2556 5.4 % 84.6 %	1140 2.4 % 84.6 %	47571 100 % 82.6 %
Femme	1555 17.1 % 21.4 %	1880 20.6 % 15.7 %	2418 26.5 % 15.7 %	1904 20.9 % 15.3 %	785 8.6 % 12.7 %	400 4.4 % 13.2 %	175 1.9 % 13 %	9117 100 % 15.8 %

Autres	145 15.8 % 2 %	162 17.7 % 1.4 %	205 22.4 % 1.3 %	188 20.5 % 1.5 %	120 13.1 % 1.9 %	65 7.1 % 2.2 %	32 3.5 % 2.4 %	917 100 % 1.6 %
Total	7265 12.6 % 100 %	11966 20.8 % 100 %	15425 26.8 % 100 %	12421 21.6 % 100 %	6160 10.7 % 100 %	3021 5.2 % 100 %	1347 2.3 % 100 %	57605 100 % 100 %

χ^2=276.803 · df=12 · Cramer's V=0.049 · p=0.000

observed values
% within Genre
% within Langage

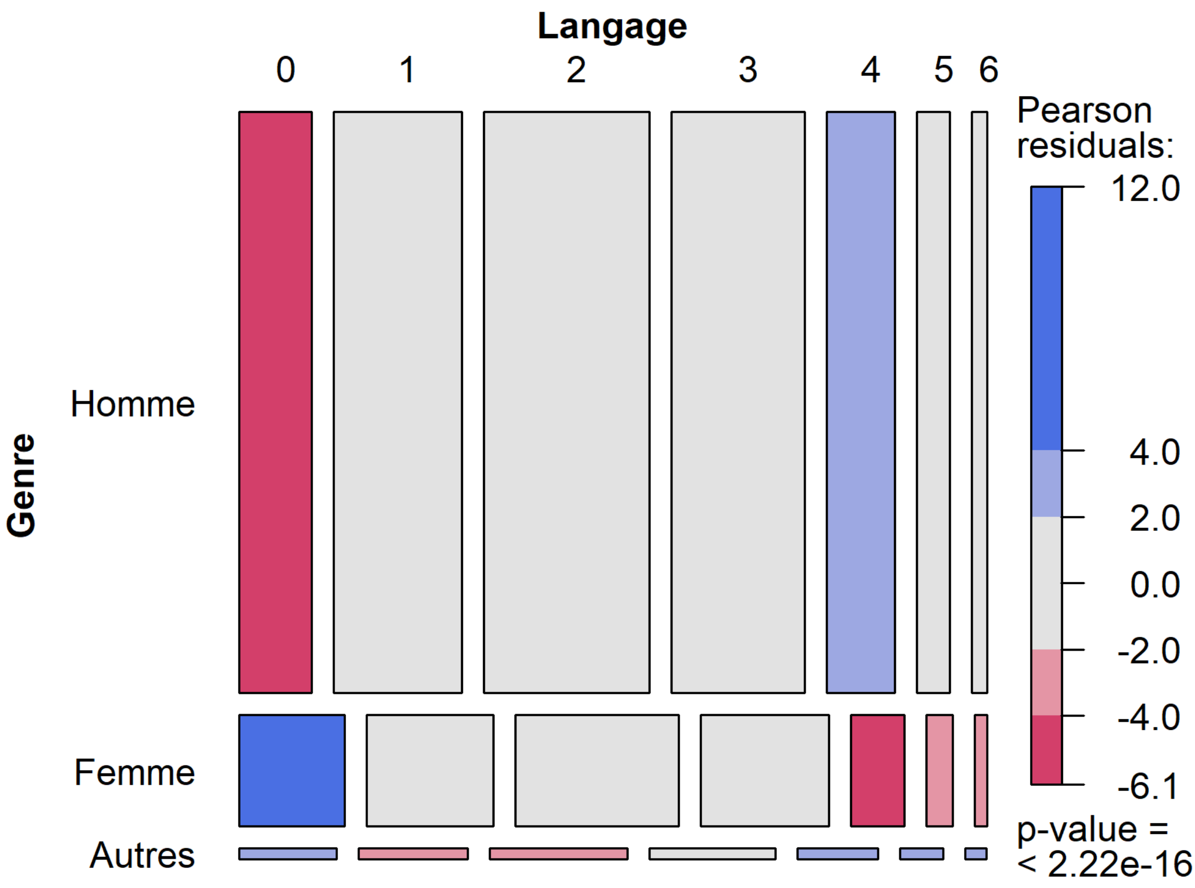
```
# Test chi2 et V de Cramer
chisq.test(table(temp_lg$Genre, temp_lg$Langage))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(temp_lg$Genre, temp_lg$Langage)
## X-squared = 276.8, df = 12, p-value < 2.2e-16
```

```
questionr::cramer.v(table(temp_lg$Genre, temp_lg$Langage))
```

```
## [1] 0.04901628
```

```
# Mosaic plot
mosaic(~ Genre + Langage,
       data = temp_lg,
       shade = TRUE, legend = TRUE, spacing = spacing_equal(c(0.5, 0.5)),
       labeling = labeling_border(rot_labels=c(0,0,0,0),
                                  just_labels=c("left","right"),
                                  offset_varnames = c(0, 0, 0, 3)),
       margins = c(0, 0, 0, 3))
```




```r
# GLM
mod_lg <- glm(Langage ~ Genre, temp_lg, family = "poisson")
anova(mod_lg)
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	57604	66484.35
Genre	2	160.694	57602	66323.66

```r
summary(mod_lg)
```

```
##
## Call:
## glm(formula = Langage ~ Genre, family = "poisson", data = temp_lg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1569  -0.9386  -0.1716   0.4737   2.2400
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.812053   0.003055 265.822   <2e-16 ***
## GenreFemme  -0.098065   0.007940 -12.351   <2e-16 ***
## GenreAutres  0.032124   0.021867   1.469    0.142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66484  on 57604  degrees of freedom
## Residual deviance: 66324  on 57602  degrees of freedom
## AIC: 203298
##
## Number of Fisher Scoring iterations: 5
```

8.3 Langage/Education

8.3.1 Data frame temporaire

```
temp_le <- ks_fusion %>% select(Education, Langage)
```

8.3.2 Test statistiques

```
# Table résumant des Donnes statistiques (moyenne, deviation standard, etc.)
crosstable(temp_le, c(Education, Langage),
          by = Education, total = "both",
          percent_pattern = "{n} ({p_col})",
          showNA = "ifany") %>%
  as_flextable()
```

```
# Table de contingence
temp_le %>%
  sjPlot::sjtab(fun = "xtab", var.labels = c("Education", "Langage"), show.row.prc = TRUE,
              show.col.prc = TRUE, show.summary = TRUE, show.legend = TRUE)
```

Education	*Langage* 0	1	2	3	4	5	6	***Total***
Autre	703 19.2 % 9.7 %	800 21.9 % 6.7 %	852 23.3 % 5.5 %	715 19.6 % 5.8 %	321 8.8 % 5.2 %	178 4.9 % 5.9 %	88 2.4 % 6.5 %	3657 100 % 6.3 %
Diplome pro	18 6.4 % 0.2 %	89 31.4 % 0.7 %	58 20.5 % 0.4 %	55 19.4 % 0.4 %	40 14.1 % 0.6 %	12 4.2 % 0.4 %	11 3.9 % 0.8 %	283 100 % 0.5 %

	822	2032	2399	1939	1012	485	226	8915
Doctorat	9.2 %	22.8 %	26.9 %	21.7 %	11.4 %	5.4 %	2.5 %	100 %
	11.3 %	17 %	15.6 %	15.6 %	16.4 %	16.1 %	16.8 %	15.5 %
	217	282	323	292	160	98	36	1408
Doctorat pro	15.4 %	20 %	22.9 %	20.7 %	11.4 %	7 %	2.6 %	100 %
	3 %	2.4 %	2.1 %	2.4 %	2.6 %	3.2 %	2.7 %	2.4 %
	2385	3453	4521	3453	1733	870	376	16791
Licence	14.2 %	20.6 %	26.9 %	20.6 %	10.3 %	5.2 %	2.2 %	100 %
	32.8 %	28.9 %	29.3 %	27.8 %	28.1 %	28.8 %	27.9 %	29.1 %
	3120	5310	7272	5967	2894	1378	610	26551
Master	11.8 %	20 %	27.4 %	22.5 %	10.9 %	5.2 %	2.3 %	100 %
	42.9 %	44.4 %	47.1 %	48 %	47 %	45.6 %	45.3 %	46.1 %
	7265	11966	15425	12421	6160	3021	1347	57605
Total	12.6 %	20.8 %	26.8 %	21.6 %	10.7 %	5.2 %	2.3 %	100 %
	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %

χ^2=415.988 · df=30 · Cramer's V=0.038 · p=0.000

observed values
% within Education
% within Langage

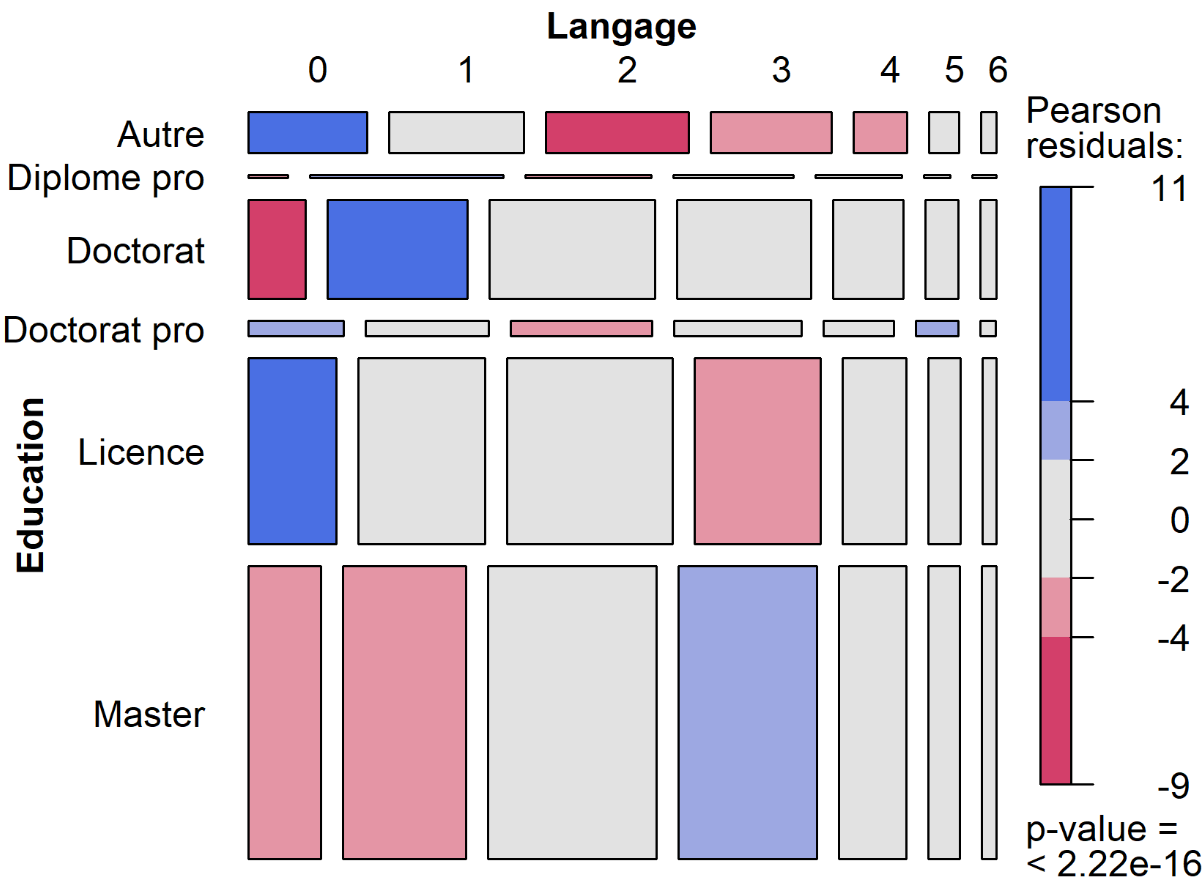
```
# Test chi2 et V de Cramer
chisq.test(table(temp_le$Education, temp_le$Langage))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(temp_le$Education, temp_le$Langage)
## X-squared = 415.99, df = 30, p-value < 2.2e-16
```

```
questionr::cramer.v(table(temp_le$Education, temp_le$Langage))
```

```
## [1] 0.03800364
```

```
# Mosaic plot
mosaic(~ Education + Langage,
          data = temp_le,
          shade = TRUE, legend = TRUE, spacing = spacing_equal(c(0.5, 0.5)),
          labeling = labeling_border(rot_labels=c(0,0,0,0),
                       just_labels=c("left","right"),
                       offset_varnames = c(0, 0, 0, 3)),
          margins = c(0, 0, 0, 3))
```




```r
# GLM
mod_le <- glm(Langage ~ Education, temp_le, family = "poisson")
anova(mod_le)
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	57604	66484.35
Education	5	135.75	57599	66348.60

```r
summary(mod_le)
```

```
##
## Call:
## glm(formula = Langage ~ Education, family = "poisson", data = temp_le)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1532  -0.9402  -0.1734   0.5342   2.2678
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.69819    0.01166  59.862  < 2e-16 ***
## EducationDiplome pro  0.14252    0.04075   3.498 0.000469 ***
## EducationDoctorat     0.13332    0.01360   9.806  < 2e-16 ***
## EducationDoctorat pro 0.10704    0.02130   5.026 5.00e-07 ***
## EducationLicence      0.07531    0.01279   5.890 3.87e-09 ***
## EducationMaster       0.11510    0.01236   9.313  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66484  on 57604  degrees of freedom
## Residual deviance: 66349  on 57599  degrees of freedom
## AIC: 203329
##
## Number of Fisher Scoring iterations: 5
```

8.4 Langage/Secteur

8.4.1 Data frame temporaire

```
temp_lr <- ks_fusion %>% select(Secteur, Langage)
```

8.4.2 Test statistiques

```
# Table résumant des Donnes statistiques (moyenne, deviation standard, etc.)
crosstable(temp_lr, c(Secteur, Langage),
          by = Secteur, total = "both",
          percent_pattern = "{n} ({p_col})",
          showNA = "ifany") %>%
  as_flextable()
```

```
# Table de contingence
temp_lr %>%
  sjPlot::sjtab(fun = "xtab", var.labels = c("Secteur", "Langage"), show.row.prc = TRUE,
             show.col.prc = TRUE, show.summary = TRUE, show.legend = TRUE)
```

Secteur	*Langage* 0	1	2	3	4	5	6	***Total***
Autre	1573 16.2 % 21.7 %	2770 28.6 % 23.1 %	2444 25.2 % 15.8 %	1616 16.7 % 13 %	746 7.7 % 12.1 %	362 3.7 % 12 %	170 1.8 % 12.6 %	9681 100 % 16.8 %
Business	257 16.6 % 3.5 %	272 17.5 % 2.3 %	406 26.2 % 2.6 %	367 23.7 % 3 %	147 9.5 % 2.4 %	75 4.8 % 2.5 %	27 1.7 % 2 %	1551 100 % 2.7 %

Informatique	148 17.2 % 2 %	131 15.3 % 1.1 %	190 22.1 % 1.2 %	184 21.4 % 1.5 %	110 12.8 % 1.8 %	54 6.3 % 1.8 %	42 4.9 % 3.1 %	859 100 % 1.5 %
Ingenierie	2466 14.5 % 33.9 %	2484 14.6 % 20.8 %	3905 23 % 25.3 %	3744 22 % 30.1 %	2411 14.2 % 39.1 %	1359 8 % 45 %	627 3.7 % 46.5 %	16996 100 % 29.5 %
Manageur	365 15.9 % 5 %	580 25.3 % 4.8 %	557 24.3 % 3.6 %	443 19.3 % 3.6 %	196 8.5 % 3.2 %	116 5.1 % 3.8 %	36 1.6 % 2.7 %	2293 100 % 4 %
Maths/Stats	478 14.3 % 6.6 %	555 16.6 % 4.6 %	866 25.9 % 5.6 %	781 23.4 % 6.3 %	420 12.6 % 6.8 %	167 5 % 5.5 %	71 2.1 % 5.3 %	3338 100 % 5.8 %
Phys. Astro.	100 10.3 % 1.4 %	168 17.4 % 1.4 %	258 26.7 % 1.7 %	237 24.5 % 1.9 %	123 12.7 % 2 %	48 5 % 1.6 %	33 3.4 % 2.4 %	967 100 % 1.7 %
Sc. de la Terre	28 12.7 % 0.4 %	43 19.5 % 0.4 %	62 28.2 % 0.4 %	43 19.5 % 0.3 %	28 12.7 % 0.5 %	12 5.5 % 0.4 %	4 1.8 % 0.3 %	220 100 % 0.4 %
Sc. des Donnes	1647 8.1 % 22.7 %	4664 23.1 % 39 %	6377 31.5 % 41.3 %	4663 23.1 % 37.5 %	1814 9 % 29.4 %	754 3.7 % 25 %	304 1.5 % 22.6 %	20223 100 % 35.1 %
Sc. Humaines	38 16 % 0.5 %	44 18.5 % 0.4 %	60 25.2 % 0.4 %	53 22.3 % 0.4 %	32 13.4 % 0.5 %	9 3.8 % 0.3 %	2 0.8 % 0.1 %	238 100 % 0.4 %
Sc. Sociales	61 12.8 % 0.8 %	96 20.2 % 0.8 %	116 24.4 % 0.8 %	117 24.6 % 0.9 %	55 11.6 % 0.9 %	21 4.4 % 0.7 %	10 2.1 % 0.7 %	476 100 % 0.8 %
Sc. Vie/Médicale	104 13.6 % 1.4 %	159 20.8 % 1.3 %	184 24.1 % 1.2 %	173 22.7 % 1.4 %	78 10.2 % 1.3 %	44 5.8 % 1.5 %	21 2.8 % 1.6 %	763 100 % 1.3 %
Total	7265 12.6 % 100 %	11966 20.8 % 100 %	15425 26.8 % 100 %	12421 21.6 % 100 %	6160 10.7 % 100 %	3021 5.2 % 100 %	1347 2.3 % 100 %	57605 100 % 100 %

χ^2=2693.015 · df=66 · Cramer's V=0.088 · p=0.000

observed values

% within Secteur

% within Langage

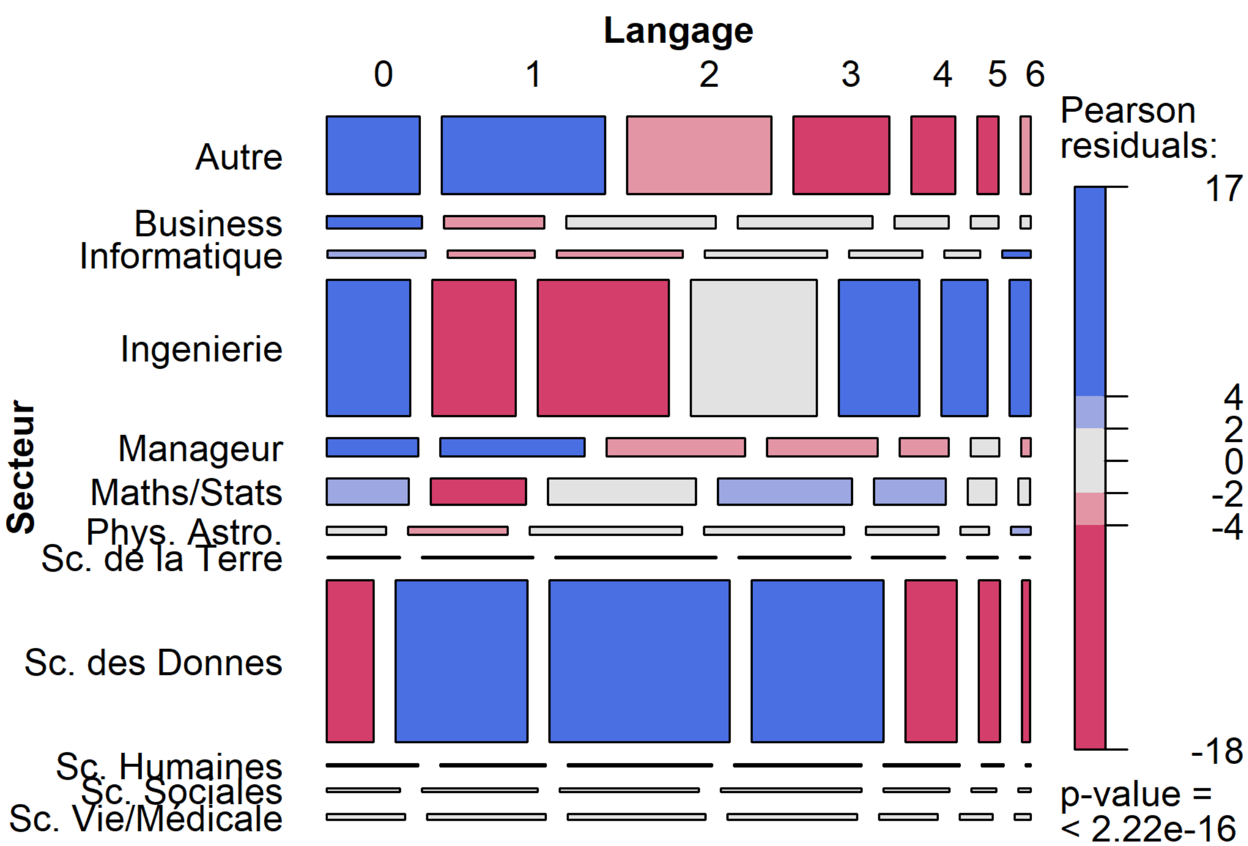
```r
# Test chi2 et V de Cramer
chisq.test(table(temp_lr$Secteur, temp_lr$Langage))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(temp_lr$Secteur, temp_lr$Langage)
## X-squared = 2693, df = 66, p-value < 2.2e-16
```

```
questionr::cramer.v(table(temp_lr$Secteur, temp_lr$Langage))
```

```
## [1] 0.08827011
```

```
# Mosaic plot
mosaic(~ Secteur + Langage, data = temp_lr, spacing = spacing_equal(c(0.5, 0.5)),
       shade = TRUE, legend = TRUE,
       labeling = labeling_border(rot_labels=c(0,0,0,0),
                                  just_labels=c("left","right"),
                                  offset_varnames = c(0, 0, 0, 5)),
       margins = c(0, 0, 0, 4))
```




```
# GLM
mod_lr <- glm(Langage ~ Secteur, temp_lr, family = "poisson")
anova(mod_lr)
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	57604	66484.35
Secteur	11	990.5155	57593	65493.84

```r
summary(mod_lr)
```

```
##
## Call:
## glm(formula = Langage ~ Secteur, family = "poisson", data = temp_lr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2159  -0.9003  -0.1293   0.5191   2.3732
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.637828   0.007388  86.331  < 2e-16 ***
## SecteurBusiness        0.120220   0.018887   6.365 1.95e-10 ***
## SecteurInformatique    0.219728   0.023418   9.383  < 2e-16 ***
## SecteurIngenierie      0.260342   0.008863  29.374  < 2e-16 ***
## SecteurManageur        0.059019   0.016487   3.580 0.000344 ***
## SecteurMaths/Stats     0.181125   0.013663  13.257  < 2e-16 ***
## SecteurPhys. Astro.    0.239449   0.022016  10.876  < 2e-16 ***
## SecteurSc. de la Terre 0.167023   0.045685   3.656 0.000256 ***
## SecteurSc. des Donnes  0.145365   0.008785  16.546  < 2e-16 ***
## SecteurSc. Humaines    0.120383   0.044979   2.676 0.007441 **
## SecteurSc. Sociales    0.166545   0.031535   5.281 1.28e-07 ***
## SecteurSc. Vie/Médicale 0.165648   0.025327   6.540 6.13e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66484  on 57604  degrees of freedom
## Residual deviance: 65494  on 57593  degrees of freedom
## AIC: 202486
##
## Number of Fisher Scoring iterations: 5
```

8.5 Langage/Continent

8.5.1 Data frame temporaire

```r
temp_lc <- ks_fusion %>% select(Continent, Langage)
```

8.5.2 Test statistiques

```r
# Table résumant des Donnes statistiques (moyenne, deviation standard, etc.)
crosstable(temp_lc, c(Continent, Langage),
          by = Continent, total = "both",
          percent_pattern = "{n} ({p_col})",
          showNA = "ifany") %>%
  as_flextable()
```

```r
# Table de contingence
temp_lc %>%
  sjPlot::sjtab(fun = "xtab", var.labels = c("Continent", "Langage"), show.row.prc = TRUE,
             show.col.prc = TRUE, show.summary = TRUE, show.legend = TRUE)
```

	Langage							
Continent	0	1	2	3	4	5	6	***Total***

Afrique	432 15 % 5.9 %	768 26.6 % 6.4 %	728 25.2 % 4.7 %	499 17.3 % 4 %	246 8.5 % 4 %	126 4.4 % 4.2 %	85 2.9 % 6.3 %	2884 100 % 5 %
Amerique du Nord	1380 12.2 % 19 %	1852 16.4 % 15.5 %	2991 26.4 % 19.4 %	2771 24.5 % 22.3 %	1408 12.4 % 22.9 %	662 5.8 % 21.9 %	255 2.3 % 18.9 %	11319 100 % 19.6 %
Amerique du Sud	379 10.4 % 5.2 %	631 17.3 % 5.3 %	991 27.2 % 6.4 %	845 23.2 % 6.8 %	450 12.4 % 7.3 %	244 6.7 % 8.1 %	101 2.8 % 7.5 %	3641 100 % 6.3 %
Asie	2903 13.5 % 40 %	4874 22.7 % 40.7 %	5751 26.8 % 37.3 %	4381 20.4 % 35.3 %	2043 9.5 % 33.2 %	1005 4.7 % 33.3 %	473 2.2 % 35.1 %	21430 100 % 37.2 %
Europe	1173 10.4 % 16.1 %	2250 19.9 % 18.8 %	3206 28.3 % 20.8 %	2564 22.7 % 20.6 %	1277 11.3 % 20.7 %	595 5.3 % 19.7 %	244 2.2 % 18.1 %	11309 100 % 19.6 %
Inconnu	635 14.9 % 8.7 %	998 23.5 % 8.3 %	1005 23.6 % 6.5 %	801 18.8 % 6.4 %	430 10.1 % 7 %	255 6 % 8.4 %	128 3 % 9.5 %	4252 100 % 7.4 %
Moyen Orient	245 13.3 % 3.4 %	425 23.1 % 3.6 %	511 27.7 % 3.3 %	349 18.9 % 2.8 %	178 9.7 % 2.9 %	93 5 % 3.1 %	42 2.3 % 3.1 %	1843 100 % 3.2 %
Oceanie	118 12.7 % 1.6 %	168 18.1 % 1.4 %	242 26.1 % 1.6 %	211 22.8 % 1.7 %	128 13.8 % 2.1 %	41 4.4 % 1.4 %	19 2 % 1.4 %	927 100 % 1.6 %
Total	7265 12.6 % 100 %	11966 20.8 % 100 %	15425 26.8 % 100 %	12421 21.6 % 100 %	6160 10.7 % 100 %	3021 5.2 % 100 %	1347 2.3 % 100 %	57605 100 % 100 %

χ^2=653.941 · df=42 · Cramer's V=0.043 · p=0.000

observed values
% within Continent
% within Langage

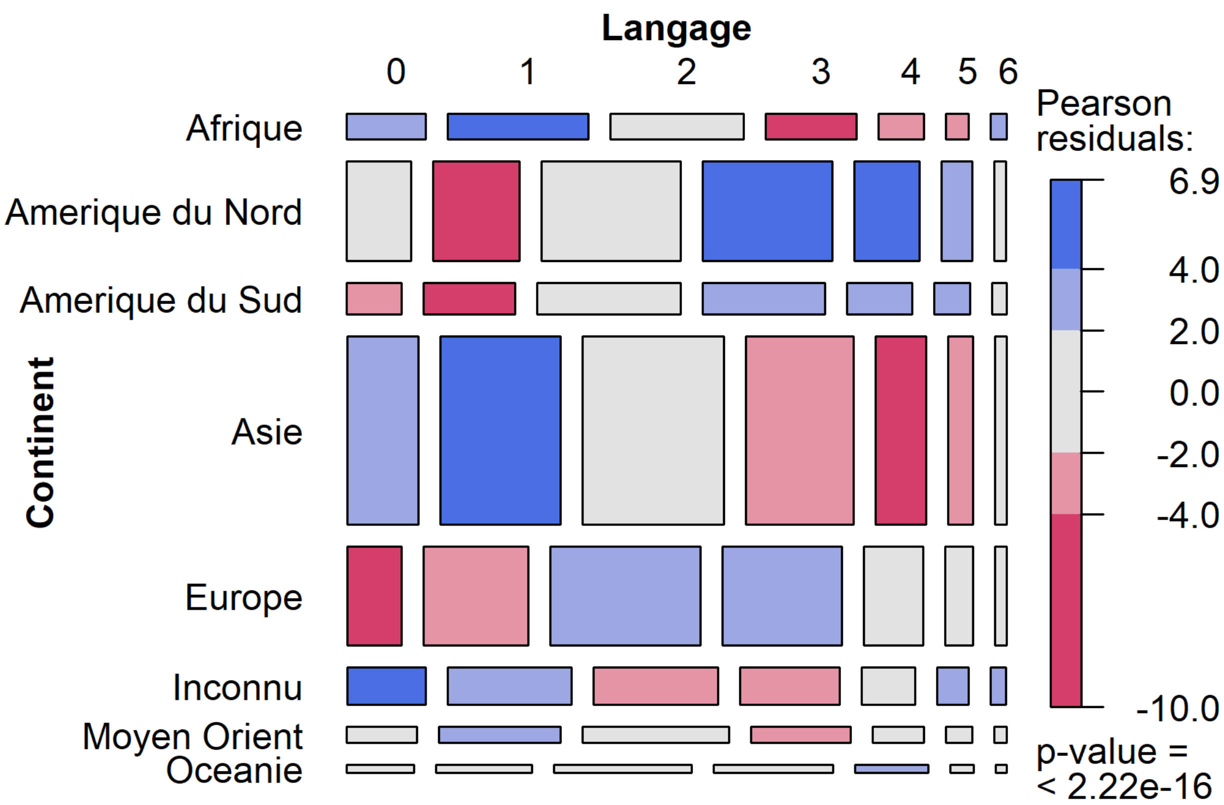
```
# Test chi2 et V de Cramer
chisq.test(table(temp_lc$Continent, temp_lc$Langage))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(temp_lc$Continent, temp_lc$Langage)
## X-squared = 653.94, df = 42, p-value < 2.2e-16
```

```
questionr::cramer.v(table(temp_lc$Continent, temp_lc$Langage))
```

```
## [1] 0.04349742
```

```
# Mosaic plot
mosaic(~ Continent + Langage,
       data = temp_lc,
       shade = TRUE, legend = TRUE, spacing = spacing_equal(c(0.5, 0.5)),
       labeling = labeling_border(rot_labels=c(0,0,0,0),
                                  just_labels=c("left","right"),
                                  offset_varnames = c(0, 0, 0, 5)),
       margins = c(0, 0, 0, 5))
```




```
# GLM
mod_lc <- glm(Langage ~ Continent, temp_lc, family = "poisson")
anova(mod_lc)
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	57604	66484.35
Continent	7	322.38	57597	66161.97

```
summary(mod_lc)
```

```
##
## Call:
## glm(formula = Langage ~ Continent, family = "poisson", data = temp_lc)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.19535  -0.86210  -0.08709   0.56432   2.25336
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.70641    0.01308  54.007  < 2e-16 ***
## ContinentAmerique du Nord  0.14873    0.01444  10.297  < 2e-16 ***
## ContinentAmerique du Sud   0.17313    0.01688  10.254  < 2e-16 ***
## ContinentAsie              0.04770    0.01389   3.433 0.000597 ***
## ContinentEurope            0.12227    0.01448   8.444  < 2e-16 ***
## ContinentInconnu           0.06258    0.01674   3.739 0.000185 ***
## ContinentMoyen Orient      0.04905    0.02064   2.377 0.017470 *
## ContinentOceanie           0.11892    0.02537   4.687 2.77e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66484  on 57604  degrees of freedom
## Residual deviance: 66162  on 57597  degrees of freedom
## AIC: 203147
##
## Number of Fisher Scoring iterations: 5
```

8.6 Langage/Salaire

8.6.1 Data frame temporaire

```r
temp_ls <- ks_fusion %>% select(Salaire, Langage)
```

8.6.2 Test statistiques

```r
# Table résumant des Donnes statistiques (moyenne, deviation standard, etc.)
crosstable(temp_ls, c(Salaire, Langage),
         by = Salaire, total = "both",
         percent_pattern = "{n} ({p_col})",
         showNA = "ifany") %>%
  as_flextable()
```

```r
# Table de contingence
temp_ls %>%
  sjPlot::sjtab(fun = "xtab", var.labels = c("Salaire", "Langage"), show.row.prc = TRUE,
              show.col.prc = TRUE, show.summary = TRUE, show.legend = TRUE)
```

Salaire	*Langage*							***Total***
	0	1	2	3	4	5	6	
0-10K	2912 14.9 % 40.1 %	4659 23.9 % 38.9 %	5023 25.7 % 32.6 %	3729 19.1 % 30 %	1815 9.3 % 29.5 %	938 4.8 % 31 %	445 2.3 % 33 %	19521 100 % 33.9 %

	546	1144	1827	1722	884	422	183	6728
100K-250K	8.1 %	17 %	27.2 %	25.6 %	13.1 %	6.3 %	2.7 %	100 %
	7.5 %	9.6 %	11.8 %	13.9 %	14.4 %	14 %	13.6 %	11.7 %
	672	1307	1639	1261	569	269	114	5831
10K-20K	11.5 %	22.4 %	28.1 %	21.6 %	9.8 %	4.6 %	2 %	100 %
	9.2 %	10.9 %	10.6 %	10.2 %	9.2 %	8.9 %	8.5 %	10.1 %
	446	806	1200	943	434	205	85	4119
20K-30K	10.8 %	19.6 %	29.1 %	22.9 %	10.5 %	5 %	2.1 %	100 %
	6.1 %	6.7 %	7.8 %	7.6 %	7 %	6.8 %	6.3 %	7.2 %
	60	86	131	124	73	47	18	539
250K-500K	11.1 %	16 %	24.3 %	23 %	13.5 %	8.7 %	3.3 %	100 %
	0.8 %	0.7 %	0.8 %	1 %	1.2 %	1.6 %	1.3 %	0.9 %
	301	649	912	667	325	140	72	3066
30K-40K	9.8 %	21.2 %	29.7 %	21.8 %	10.6 %	4.6 %	2.3 %	100 %
	4.1 %	5.4 %	5.9 %	5.4 %	5.3 %	4.6 %	5.3 %	5.3 %
	242	649	823	623	295	157	69	2858
40K-50K	8.5 %	22.7 %	28.8 %	21.8 %	10.3 %	5.5 %	2.4 %	100 %
	3.3 %	5.4 %	5.3 %	5 %	4.8 %	5.2 %	5.1 %	5 %
	60	55	38	57	29	12	10	261
500K +	23 %	21.1 %	14.6 %	21.8 %	11.1 %	4.6 %	3.8 %	100 %
	0.8 %	0.5 %	0.2 %	0.5 %	0.5 %	0.4 %	0.7 %	0.5 %
	258	569	788	667	328	121	52	2783
50K-60K	9.3 %	20.4 %	28.3 %	24 %	11.8 %	4.3 %	1.9 %	100 %
	3.6 %	4.8 %	5.1 %	5.4 %	5.3 %	4 %	3.9 %	4.8 %
	212	462	629	491	279	118	44	2235
60K-70K	9.5 %	20.7 %	28.1 %	22 %	12.5 %	5.3 %	2 %	100 %
	2.9 %	3.9 %	4.1 %	4 %	4.5 %	3.9 %	3.3 %	3.9 %
	173	422	585	463	231	115	36	2025
70K-80K	8.5 %	20.8 %	28.9 %	22.9 %	11.4 %	5.7 %	1.8 %	100 %
	2.4 %	3.5 %	3.8 %	3.7 %	3.8 %	3.8 %	2.7 %	3.5 %
	137	278	438	394	179	86	33	1545
80K-90K	8.9 %	18 %	28.3 %	25.5 %	11.6 %	5.6 %	2.1 %	100 %
	1.9 %	2.3 %	2.8 %	3.2 %	2.9 %	2.8 %	2.4 %	2.7 %
	131	255	430	387	211	91	34	1539
90K-100K	8.5 %	16.6 %	27.9 %	25.1 %	13.7 %	5.9 %	2.2 %	100 %
	1.8 %	2.1 %	2.8 %	3.1 %	3.4 %	3 %	2.5 %	2.7 %
	1115	625	962	893	508	300	152	4555
Inconnu	24.5 %	13.7 %	21.1 %	19.6 %	11.2 %	6.6 %	3.3 %	100 %
	15.3 %	5.2 %	6.2 %	7.2 %	8.2 %	9.9 %	11.3 %	7.9 %

	7265	11966	15425	12421	6160	3021	1347	57605
Total	12.6 %	20.8 %	26.8 %	21.6 %	10.7 %	5.2 %	2.3 %	100 %
	100 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %

χ^2=1662.905 · df=78 · Cramer's V=0.069 · p=0.000

observed values
% within Salaire
% within Langage

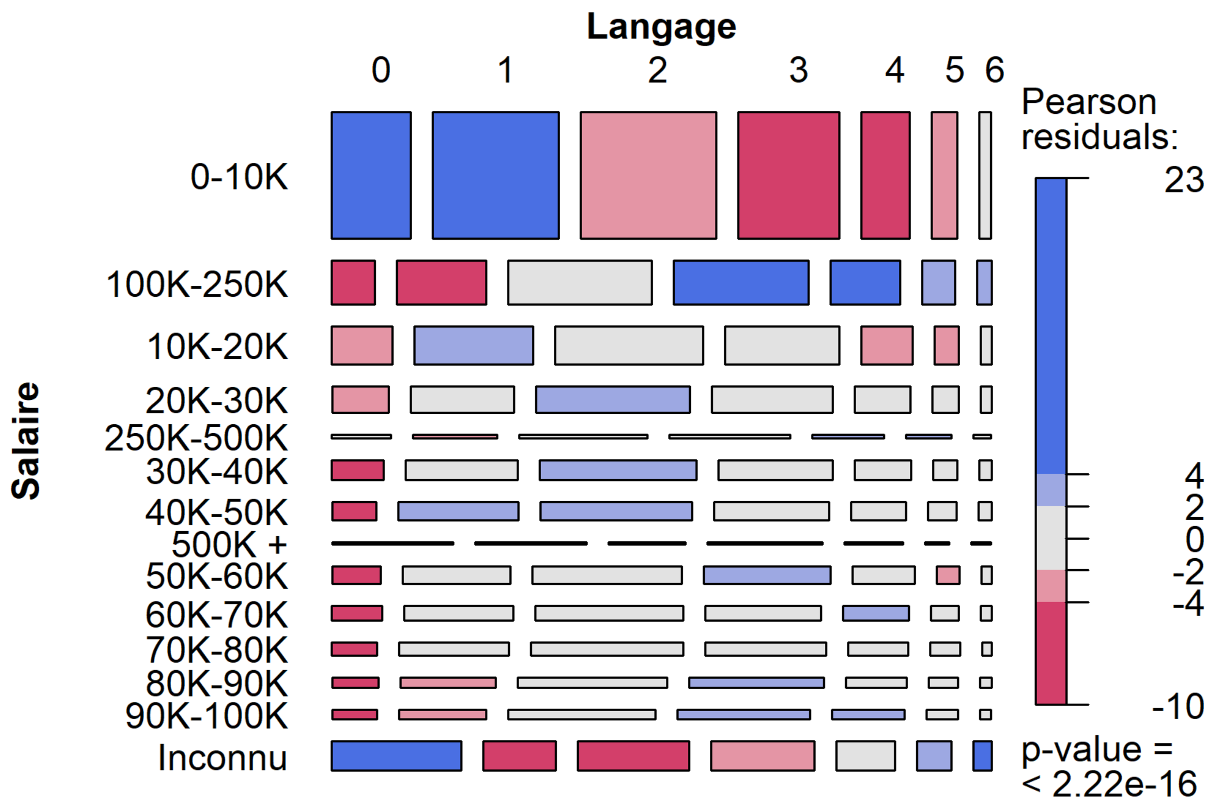
```
# Test chi2 et V de Cramer
chisq.test(table(temp_ls$Salaire, temp_ls$Langage))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(temp_ls$Salaire, temp_ls$Langage)
## X-squared = 1662.9, df = 78, p-value < 2.2e-16
```

```
questionr::cramer.v(table(temp_ls$Salaire, temp_ls$Langage))
```

```
## [1] 0.06936302
```

```
# Mosaic plot
mosaic(~ Salaire + Langage, data = temp_ls,spacing = spacing_equal(c(0.5, 0.5)),
       shade = TRUE, legend = TRUE,
       labeling = labeling_border(rot_labels=c(0,0,0,0),
                                  just_labels=c("left","right"),
                                  offset_varnames = c(0, 0, 0, 5)),
       margins = c(0, 0, 0, 5))
```




```
# GLM
mod_ls <- glm(Langage ~ Salaire, temp_ls, family = "poisson")
anova(mod_ls)
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	57604	66484.35
Salaire	13	527.3562	57591	65956.99

```
summary(mod_ls)
```

```
##
## Call:
## glm(formula = Langage ~ Salaire, family = "poisson", data = temp_ls)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.24228  -0.86066  -0.08551   0.56602   2.22343
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.730107   0.004968 146.953  < 2e-16 ***
## Salaire100K-250K  0.179502   0.009194  19.523  < 2e-16 ***
## Salaire10K-20K    0.046178   0.010178   4.537 5.71e-06 ***
## Salaire20K-30K    0.084942   0.011495   7.389 1.48e-13 ***
## Salaire250K-500K  0.191734   0.027617   6.943 3.85e-12 ***
## Salaire30K-40K    0.081909   0.013019   6.292 3.14e-10 ***
## Salaire40K-50K    0.098166   0.013324   7.368 1.73e-13 ***
## Salaire500K +    -0.006769   0.043398  -0.156    0.876
## Salaire50K-60K    0.098747   0.013474   7.329 2.32e-13 ***
## Salaire60K-70K    0.107169   0.014777   7.252 4.10e-13 ***
## Salaire70K-80K    0.111034   0.015415   7.203 5.90e-13 ***
## Salaire80K-90K    0.137781   0.017217   8.003 1.22e-15 ***
## Salaire90K-100K   0.168219   0.017009   9.890  < 2e-16 ***
## SalaireInconnu    0.022902   0.011317   2.024    0.043 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66484  on 57604  degrees of freedom
## Residual deviance: 65957  on 57591  degrees of freedom
## AIC: 202954
##
## Number of Fisher Scoring iterations: 5
```

8.7 Suppression dataframe temporaire

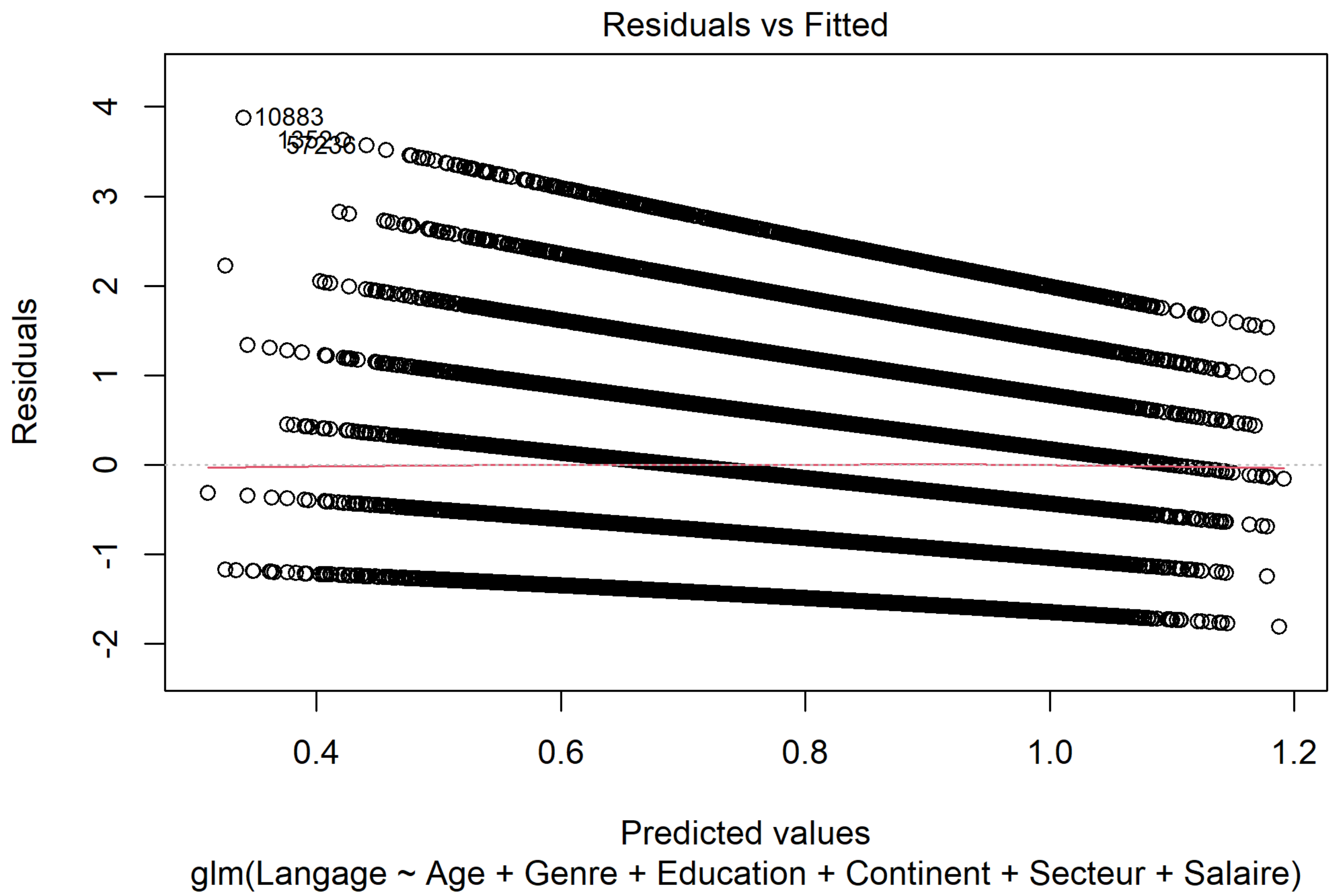
```r
rm(temp_la, temp_lg, temp_le, temp_lr, temp_lc, temp_ls)
```

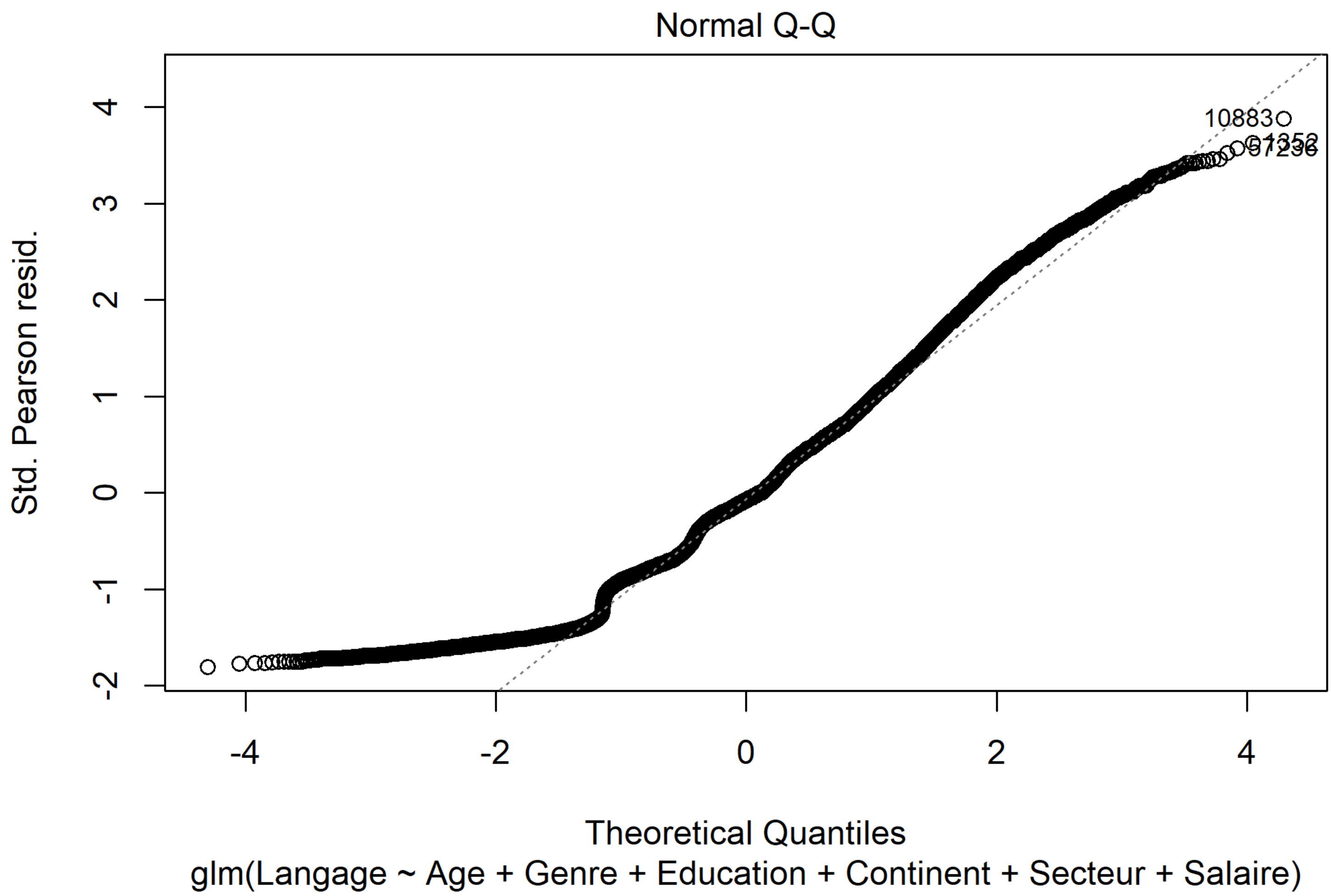
9 Modèle linéaire généralisé

```r
# GLM sur l'ensemble des variables en fonction de la variable cible
mod <- glm(Langage ~ Age + Genre + Education + Continent + Secteur + Salaire,
           ks_fusion, family = "poisson")

summary(mod)
```

```
##
## Call:
## glm(formula = Langage ~ Age + Genre + Education + Continent +
##     Secteur + Salaire, family = "poisson", data = ks_fusion)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.56083  -0.82987  -0.08161   0.57300   2.86796
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.464918   0.019026  24.437  < 2e-16 ***
## Age30-39                  -0.014546   0.007127  -2.041 0.041243 *
## Age40-49                   0.032129   0.009134   3.518 0.000435 ***
## Age50-59                   0.028610   0.012296   2.327 0.019976 *
## Age60-69                  -0.012973   0.020179  -0.643 0.520313
## Age70+                    -0.160100   0.044557  -3.593 0.000327 ***
## GenreFemme                -0.081191   0.008062 -10.071  < 2e-16 ***
## GenreAutres                0.055176   0.021959   2.513 0.011980 *
## EducationDiplome pro       0.222960   0.040895   5.452 4.98e-08 ***
## EducationDoctorat          0.137187   0.013909   9.863  < 2e-16 ***
## EducationDoctorat pro      0.100597   0.021379   4.705 2.53e-06 ***
## EducationLicence           0.064801   0.012894   5.026 5.02e-07 ***
## EducationMaster            0.100523   0.012471   8.061 7.59e-16 ***
## ContinentAmerique du Nord  0.043425   0.015616   2.781 0.005421 **
## ContinentAmerique du Sud   0.133265   0.017070   7.807 5.87e-15 ***
## ContinentAsie              0.006923   0.014026   0.494 0.621580
## ContinentEurope            0.039295   0.015148   2.594 0.009485 **
## ContinentInconnu           0.021747   0.016869   1.289 0.197323
## ContinentMoyen Orient      0.008969   0.020769   0.432 0.665869
## ContinentOceanie           0.019297   0.025983   0.743 0.457667
## SecteurBusiness            0.130325   0.019208   6.785 1.16e-11 ***
## SecteurInformatique        0.278071   0.023707  11.730  < 2e-16 ***
## SecteurIngenierie          0.298112   0.009302  32.048  < 2e-16 ***
## SecteurManageur            0.041942   0.016688   2.513 0.011961 *
## SecteurMaths/Stats         0.200852   0.013890  14.460  < 2e-16 ***
## SecteurPhys. Astro.        0.217767   0.022229   9.797  < 2e-16 ***
## SecteurSc. de la Terre     0.184353   0.045803   4.025 5.70e-05 ***
## SecteurSc. des Donnes      0.159034   0.009003  17.665  < 2e-16 ***
## SecteurSc. Humaines        0.106653   0.045116   2.364 0.018079 *
## SecteurSc. Sociales        0.165241   0.031743   5.206 1.93e-07 ***
## SecteurSc. Vie/Médicale    0.174525   0.025533   6.835 8.19e-12 ***
## Salaire100K-250K           0.148740   0.011349  13.106  < 2e-16 ***
## Salaire10K-20K             0.020656   0.010294   2.007 0.044801 *
## Salaire20K-30K             0.054240   0.011745   4.618 3.87e-06 ***
## Salaire250K-500K           0.168583   0.028470   5.921 3.19e-09 ***
## Salaire30K-40K             0.052288   0.013393   3.904 9.46e-05 ***
## Salaire40K-50K             0.074658   0.013821   5.402 6.59e-08 ***
## Salaire500K +             -0.018095   0.043639  -0.415 0.678400
## Salaire50K-60K             0.073194   0.014036   5.215 1.84e-07 ***
## Salaire60K-70K             0.084067   0.015425   5.450 5.04e-08 ***
## Salaire70K-80K             0.088844   0.016151   5.501 3.78e-08 ***
## Salaire80K-90K             0.112597   0.017964   6.268 3.66e-10 ***
## Salaire90K-100K            0.134378   0.017877   7.517 5.62e-14 ***
## SalaireInconnu            -0.064841   0.011939  -5.431 5.60e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66484  on 57604  degrees of freedom
## Residual deviance: 64375  on 57561  degrees of freedom
## AIC: 201432
##
## Number of Fisher Scoring iterations: 5
```

```r
plot(mod, which = c(1, 2))
```


Residuals vs Fitted
Residuals
Predicted values
glm(Langage ~ Age + Genre + Education + Continent + Secteur + Salaire)
10883


Normal Q-Q
Std. Pearson resid.
Theoretical Quantiles
glm(Langage ~ Age + Genre + Education + Continent + Secteur + Salaire)
10883

10 Export Donnes

10.1 Export copyboard

```
clipboard_rapport <- addmargins(table(ks_fusion$Genre, ks_fusion$Langage))
write_clip(clipboard_rapport)
```

10.2 Export csv

```
table_export <- ks_fusion %>% select(Annee, Genre, Age)
table_export <- table_export[1:50, ]
write_csv(table_export ,"~/R things/DU_TPE/UE_4/table_exportcsv.csv")
```