

Relation

Mamadou Billo Diallo

2023-10-06

#Reflexion

En effet, sur ce chapitre nous étudions la relation entre les données tels que les doubles axes y pour pouvoir mettre ensemble 2 choses qui mesurent la même chose La corrélation pour savoir comment les données sont-elles corrélées. La régression pour pouvoir faire des lignes et observés les résidus entre cette ligne et les données; La prédiction aussi pour pouvoir prendre des décisions C'est un chapitre très complexe mais qui donne des bases solides pour pouvoir faire de la prédiction. ### Charger et nettoyer les données

Tout d'abord, nous chargeons les libraries que nous utiliserons :

```
library(tidyverse)
library(patchwork) # Pour combiner des graphiques ggplots
library(GGally)    # pour les matrices de scatterplot
library(broom)     # Pour convertir des objets modèles en jeu de données

result<- read_csv("data/results_2016.csv")
## Rows: 3158 Columns: 20
## — Column specification

```

```
## Delimiter: ","
## chr (3): state, state_po, county
## dbl (17): year, FIPS, totalvotes, D, O, R, percent_dem, percent_gop,
percent_other, total_population, percent_white, percent_black,
percent_asian,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

#3 Graphiques de visualisation des données

```
#Rangement des données
#resultat de vote par candidat selon l'Etat
result_vote_candidat<- result%>%
  group_by(state)%>%
  summarise(total1=sum(D),total2=sum(O),total3=sum(R),n=n())

#nous enlevons tous les na
result_vote_candidat <- na.omit(result_vote_candidat)
result_vote_candidat
## # A tibble: 48 x 5
```

```
##      state      total1 total2 total3      n
##      <chr>      <dbl>  <dbl>  <dbl> <int>
##  1 Alabama      729547   75570 1318250    67
##  2 Alaska       116454   38557  163387    41
##  3 Arizona      1161167  215188 1252401    15
##  4 Arkansas      380494   64530  684872    75
##  5 California    8753788  943997 4483810    58
##  6 Colorado      1338870  238866 1202484    64
##  7 Delaware      235603   22267  185127     3
##  8 District of Columbia 282830   15715   12723     1
##  9 Florida       4504975  297178 4617886    67
## 10 Georgia       1877963  147644 2089104   159
## # ... with 38 more rows
```

```
#nous integrons le roboto condensed
windowsFonts(`Roboto Condensed` = windowsFont("Roboto Condensed"))
```

#graphe 1

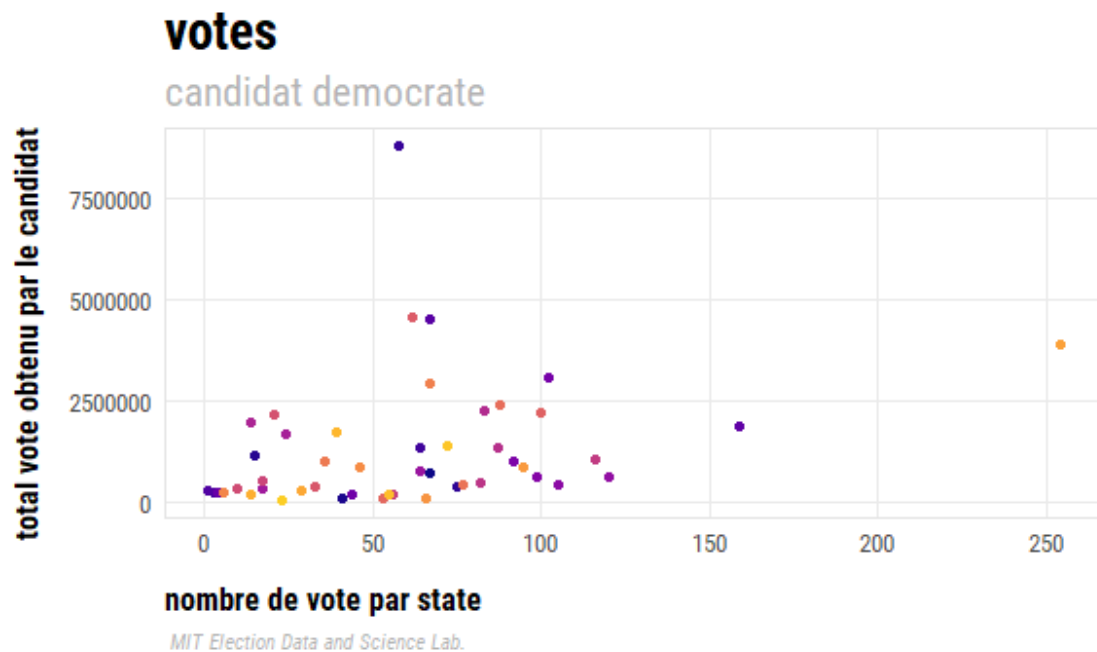
```
#graphe pour les resultats de vote du candidat democrate
CD1<-ggplot(result_vote_candidat,mapping = aes(x=n,y=total1))+
  geom_point(aes(colour = factor(state)),show.legend = FALSE)+
  # On utilise viridis
  scale_color_viridis_d(option = "plasma", end = 0.9) +
  labs(x = "nombre de vote par state", y = "total vote obtenu par le
candidat",
       color = "state", size = "total1",
       title = "votes ",
       subtitle = "candidat democrate",
       caption = " MIT Election Data and Science Lab.")

CD2<-CD1+
  theme_minimal(base_family = "Roboto Condensed", base_size = 12) +
  theme(panel.grid.minor = element_blank(),
        # Titre en gras et plus gros
        plot.title = element_text(face = "bold", size = rel(1.7)),
        # Sous-titre simple, légèrement plus grand et gris
        plot.subtitle = element_text(face = "plain", size = rel(1.3), color =
"grey70"),
        # Légende en italique, plus petite, grise et alignée à gauche
        plot.caption = element_text(face = "italic", size = rel(0.7),
                                     color = "grey70", hjust = 0),
        # Titres de légende en gras
        legend.title = element_text(face = "bold"),
        # Titres de facettes en gras, légèrement plus grands, alignés à
gauche pour des raisons de répétition
        strip.text = element_text(face = "bold", size = rel(1.1), hjust = 0),
        # Titres des axes en gras
        axis.title = element_text(face = "bold"),
        # Ajoutez un peu d'espace au-dessus du titre de l'axe des x et
alignez-le à gauche
```

```

axis.title.x = element_text(margin = margin(t = 10), hjust = 0),
# Ajoutez un peu d'espace à droite du titre de l'axe des ordonnées et
alignez-le en haut
axis.title.y = element_text(margin = margin(r = 10), hjust = 1))
CD<-CD2+
# Ajouter un fond gris clair aux titres des facettes, sans bordures
theme(strip.background = element_rect(fill = "grey90", color = NA),
# Ajoutez une fine bordure grise autour de tous les tracés pour lier
les titres des facettes
panel.border = element_rect(color = "grey90", fill = NA))
CD

```



#graphe 2

```

#graphe pour les resultats de vote pour les autres candidats
C01<-ggplot(result_vote_candidat,mapping = aes(x=n,y=total2))+
  geom_point(aes(colour = factor(state)),show.legend = FALSE)+
  # On utilise viridis
  scale_color_viridis_d(option = "plasma", end = 0.9) +
  labs(x = "nombre de vote par state", y = "total vote obtenu par les
candidat",
       color = "state", size = "total2",
       title = "votes ",
       subtitle = "autres candidats",
       caption = " MIT Election Data and Science Lab.")
C02<-C01+
  theme_minimal(base_family = "Roboto Condensed", base_size = 12) +
  theme(panel.grid.minor = element_blank(),

```

```

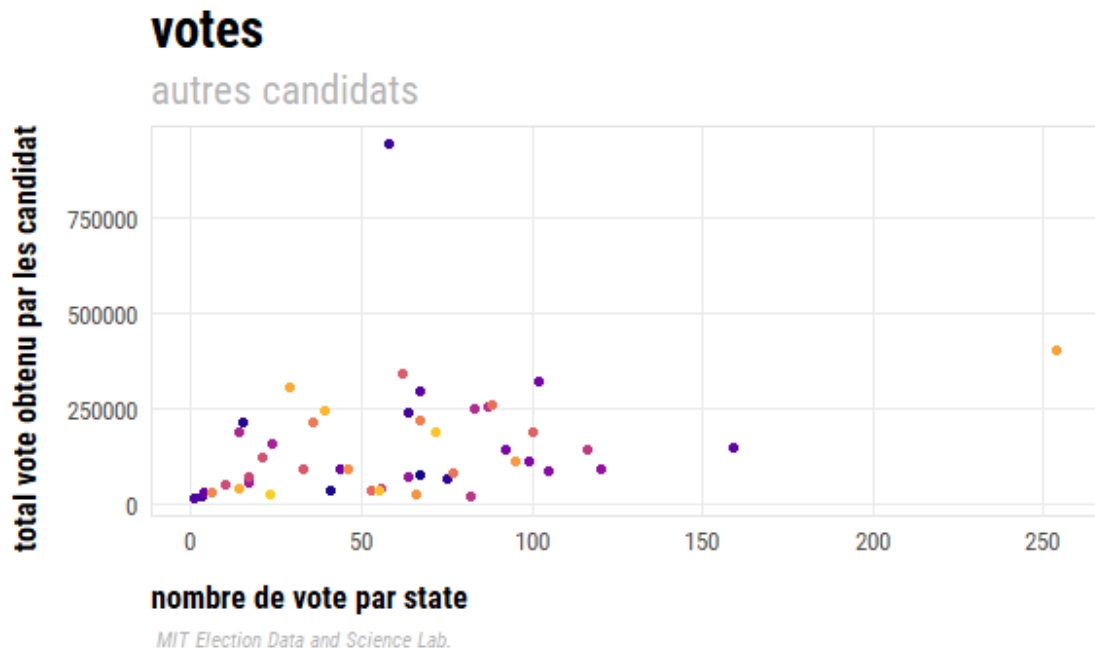
# Titre en gras et plus gros
plot.title = element_text(face = "bold", size = rel(1.7)),
# Sous-titre simple, légèrement plus grand et gris
plot.subtitle = element_text(face = "plain", size = rel(1.3), color =
"grey70"),
# Légende en italique, plus petite, grise et alignée à gauche
plot.caption = element_text(face = "italic", size = rel(0.7),
color = "grey70", hjust = 0),

# Titres de légende en gras
legend.title = element_text(face = "bold"),
# Titres de facettes en gras, légèrement plus grands, alignés à
gauche pour des raisons de répétition
strip.text = element_text(face = "bold", size = rel(1.1), hjust = 0),
# Titres des axes en gras
axis.title = element_text(face = "bold"),
# Ajoutez un peu d'espace au-dessus du titre de l'axe des x et
alignez-le à gauche
axis.title.x = element_text(margin = margin(t = 10), hjust = 0),
# Ajoutez un peu d'espace à droite du titre de l'axe des ordonnées et
alignez-le en haut
axis.title.y = element_text(margin = margin(r = 10), hjust = 1))

CO<-CO2+
# Ajouter un fond gris clair aux titres des facettes, sans bordures
theme(strip.background = element_rect(fill = "grey90", color = NA),
# Ajoutez une fine bordure grise autour de tous les tracés pour lier
les titres des facettes
panel.border = element_rect(color = "grey90", fill = NA))

CO

```



#graphe3

```
#graphe pour les resultats de vote pour les autres candidats
CR1<-ggplot(result_vote_candidat,mapping = aes(x=n,y=total3))+
  geom_point(aes(colour = factor(state)),show.legend = FALSE)+
  # On utilise viridis
  scale_color_viridis_d(option = "plasma", end = 0.9) +
  scale_y_continuous(labels = scales::label_number())+
  labs(x = "nombre de vote par state", y = "total vote obtenu par le candidat
repubicain ",
       color = "state", size = "total2",
       title = "votes ",
       subtitle = "candidat republicain",
       caption = " MIT Election Data and Science Lab.")

CR2<-CR1+
  theme_minimal(base_family = "Roboto Condensed", base_size = 12) +
  theme(panel.grid.minor = element_blank(),
        # Titre en gras et plus gros
        plot.title = element_text(face = "bold", size = rel(1.7)),
        # Sous-titre simple, légèrement plus grand et gris
        plot.subtitle = element_text(face = "plain", size = rel(1.3), color =
"grey70"),
        # Légende en italique, plus petite, grise et alignée à gauche
        plot.caption = element_text(face = "italic", size = rel(0.7),
                                     color = "grey70", hjust = 0),

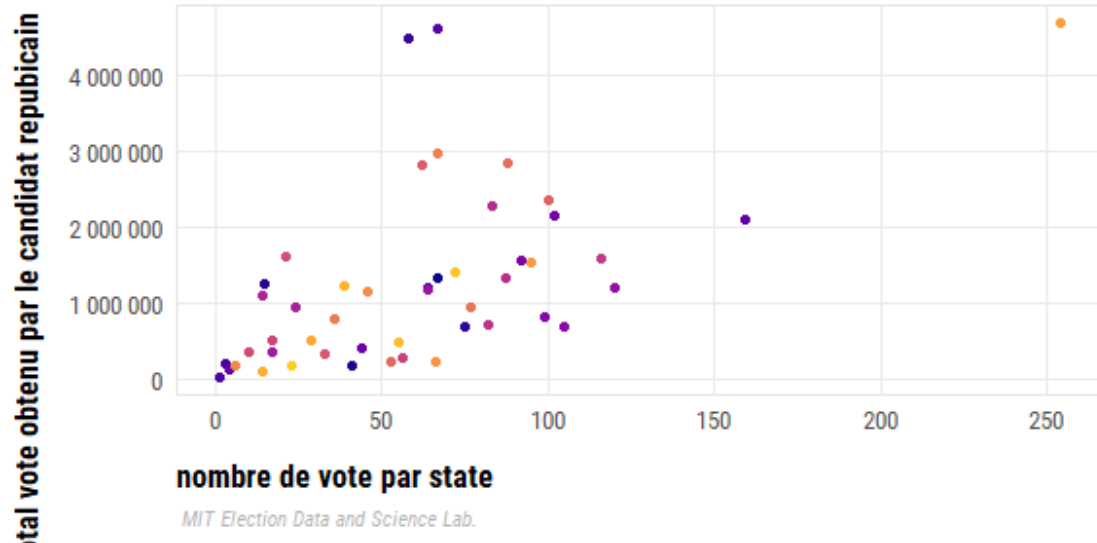
        # Titres de légende en gras
        legend.title = element_text(face = "bold"),
        # Titres de facettes en gras, légèrement plus grands, alignés à
gauche pour des raisons de répétition
        strip.text = element_text(face = "bold", size = rel(1.1), hjust = 0),
        # Titres des axes en gras
        axis.title = element_text(face = "bold"),
        # Ajoutez un peu d'espace au-dessus du titre de l'axe des x et
alignez-le à gauche
        axis.title.x = element_text(margin = margin(t = 10), hjust = 0),
        # Ajoutez un peu d'espace à droite du titre de l'axe des ordonnées et
alignez-le en haut
        axis.title.y = element_text(margin = margin(r = 10), hjust = 1))

CR<-CR2+
  # Ajouter un fond gris clair aux titres des facettes, sans bordures
  theme(strip.background = element_rect(fill = "grey90", color = NA),
        # Ajoutez une fine bordure grise autour de tous les tracés pour lier
les titres des facettes
        panel.border = element_rect(color = "grey90", fill = NA))

CR
```

votes

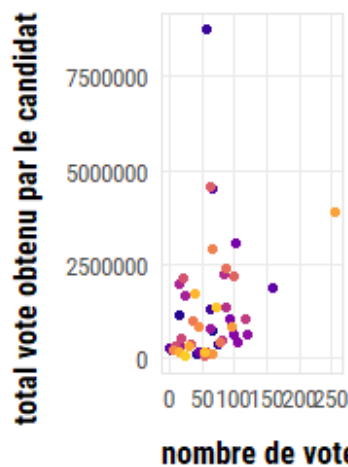
candidat republicain



```
patch<- CD+CO+CR
patch
```

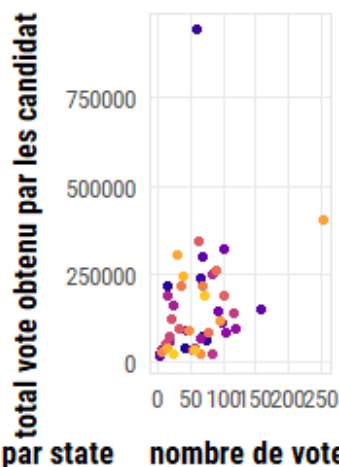
votes

candidat democrate



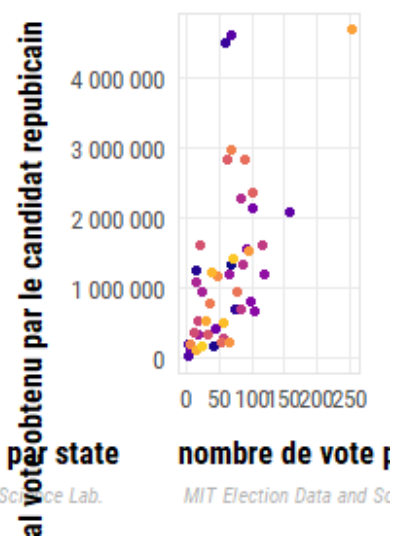
votes

autres candidats



votes

candidat repu



#predication

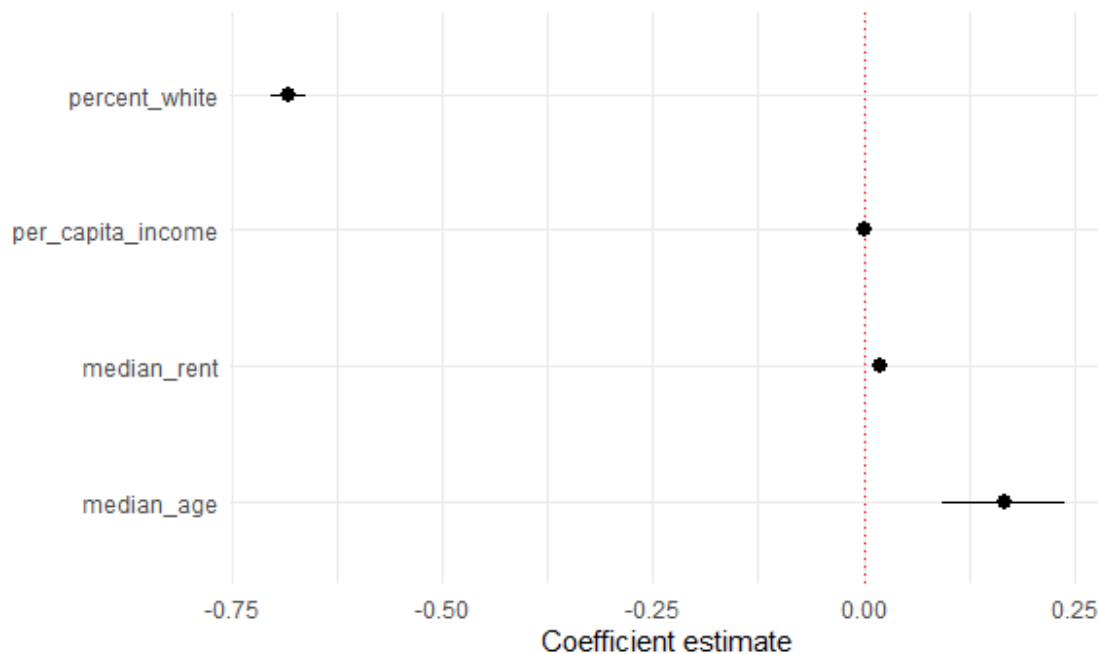
```
result_predict <- lm(percent_dem ~ median_age + percent_white +
  per_capita_income + median_rent + state,
  data = result)
result_coefs<- tidy(result_predict , conf.int = TRUE) %>%
  filter(!str_detect(term, "state"))%>%
```

```

filter(term != "(Intercept)") # Nous pouvons généralement ignorer le tracé
de l'ordonnée
result_coefs
## # A tibble: 4 × 7
##   term                estimate std.error statistic  p.value  conf.low
##   <chr>                <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 median_age           0.166    0.0368      4.52 6.33e- 6  0.0942
## 2 percent_white       -0.682    0.0110     -61.9  0      -0.704
## 3 per_capita_income    0.000271 0.0000497    5.46 5.18e- 8  0.000174
## 4 median_rent          0.0200    0.00163     12.3 6.23e-34  0.0168
## 5 median_age           0.238
## 6 percent_white        0.661
## 7 per_capita_income    0.000369
## 8 median_rent          0.0232

ggplot(result_coefs, aes(x = estimate, y = term)) +
  geom_vline(xintercept = 0, color = "red", linetype = "dotted") +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high)) +
  labs(x = "Coefficient estimate", y = NULL) +
  theme_minimal()

```



median_rent et median_age ont un impact positif en moyenne de 0.16625 et 0.02003 sur y
 per_capita_income n'a pas d'impact
 percent_white a un impact negatif en moyenne de - 0.68228 sur y

```

library(marginaleffects)
## Warning: Le package 'marginaleffects' a été compilé avec la version R
4.2.3

```

```

my_predictions <- predictions(
  result_predict,
  newdata = datagrid(median_rent = seq(9000, 60000, by = 100),
    state = "Georgia"))

my_predictions
##
##   median_rent   state Estimate Std. Error    z Pr(>|z|)      S 2.5 % 97.5 %
##       9000 Georgia      196      13.8 14.2  <0.001 149.0   169   223
##       9100 Georgia      198      14.0 14.1  <0.001 148.6   170   225
##       9200 Georgia      200      14.1 14.1  <0.001 148.1   172   227
##       9300 Georgia      202      14.3 14.1  <0.001 147.7   174   230
##       9400 Georgia      204      14.5 14.1  <0.001 147.3   175   232
## --- 501 rows omitted. See ?avg_predictions and ?print.marginalEffects ---
##      59600 Georgia     1209      96.2 12.6  <0.001 117.8  1021  1398
##      59700 Georgia     1211      96.4 12.6  <0.001 117.8  1022  1400
##      59800 Georgia     1213      96.6 12.6  <0.001 117.8  1024  1402
##      59900 Georgia     1215      96.7 12.6  <0.001 117.8  1026  1405
##      60000 Georgia     1217      96.9 12.6  <0.001 117.8  1027  1407
## Columns: rowid, estimate, std.error, statistic, p.value, s.value,
## conf.low, conf.high, percent_dem, median_age, percent_white,
## per_capita_income, median_rent, state
## Type: response

ggplot(my_predictions, aes(x = median_rent, y = estimate)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high),
    fill = "#BF3984", alpha = 0.5) +
  geom_line(size = 1, color = "#BF3984") +
  labs(x = "median_rent", y = "Predicted median_rent") +
  theme_minimal()

```

