Ggplot2 : Grammaire des graphique de la base au maîtrise

ASRI Ayoub
04/01/2020

Outline

1 Introduction

2 Grammaire des graphiques

3 La base de données utilisée

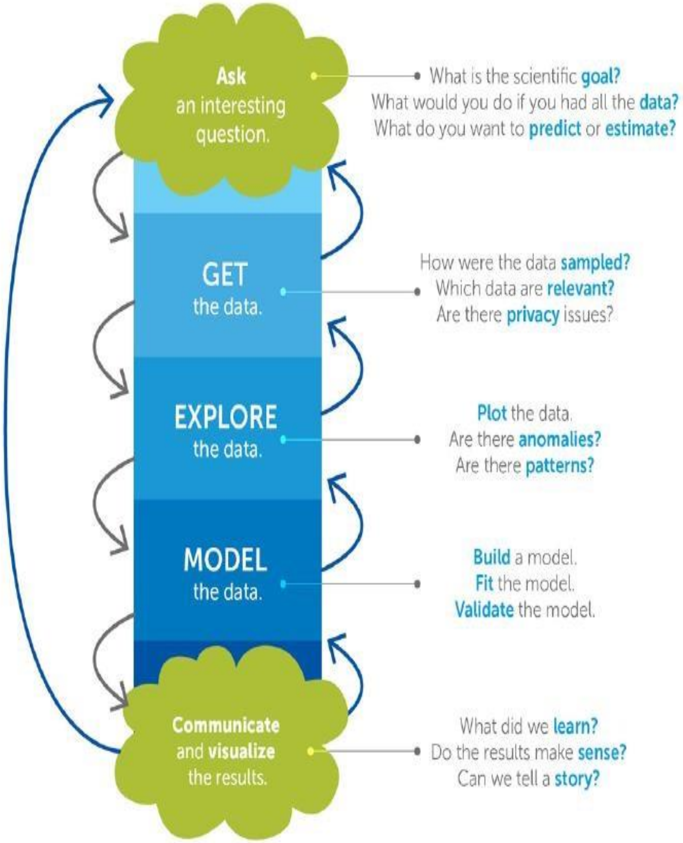
4 Les packages complémentaires

5 Utilisation des graphiques

1

Introduction

Le processus de data science

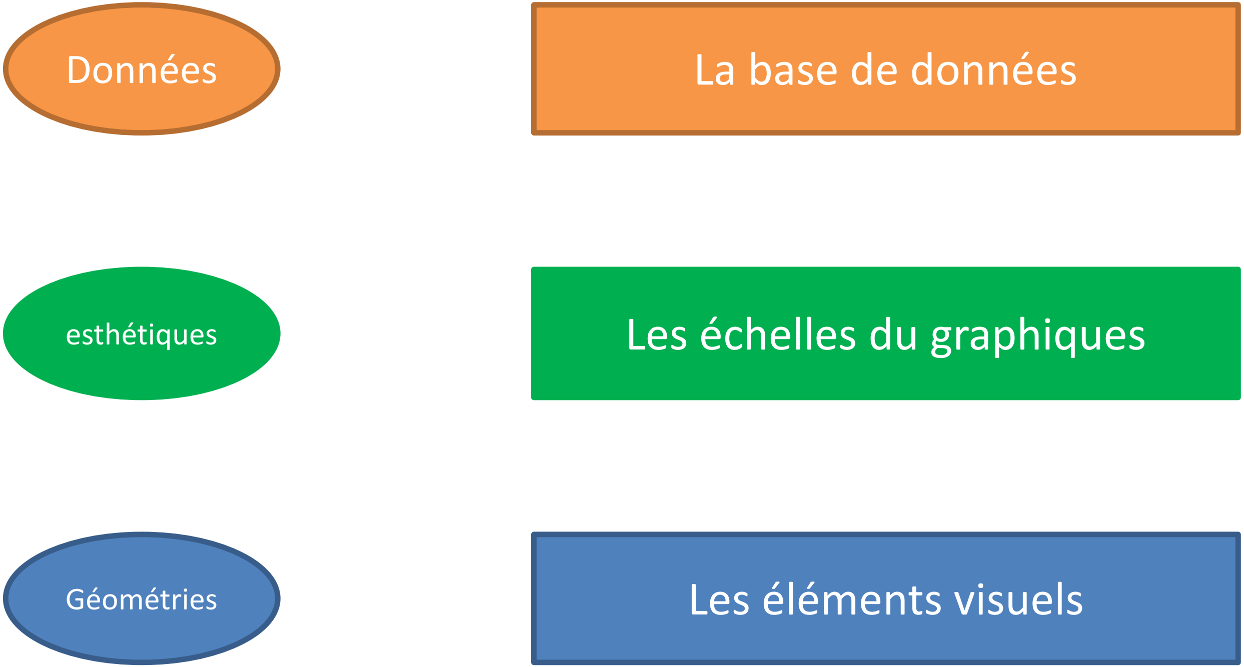


Source : https://academy.vertabelo.com/blog/agile-data-science-improve-workflow-with-scrum/

2

Grammaire des graphiques

Les éléments de Grammaire essentiels

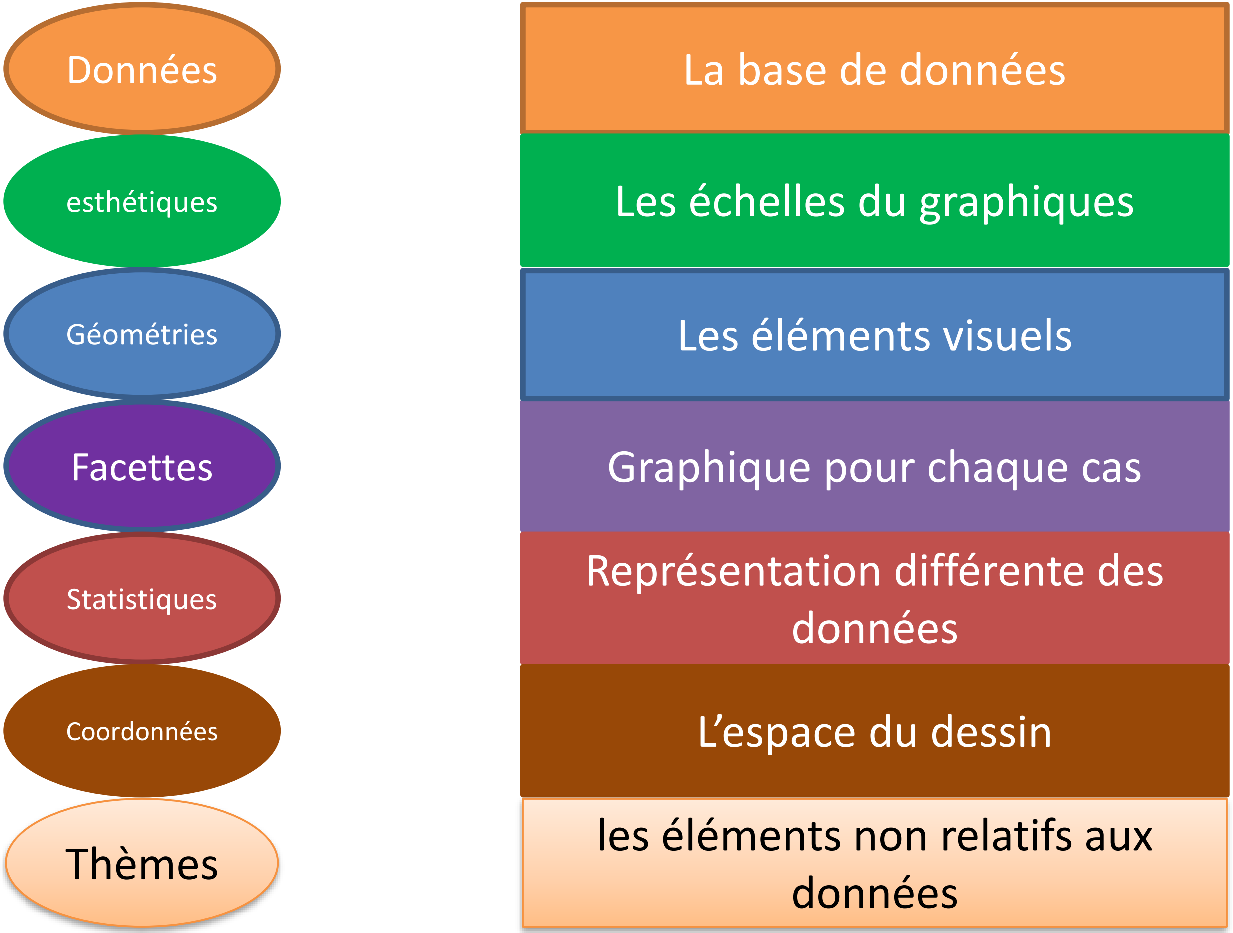


Données — La base de données

esthétiques — Les échelles du graphiques

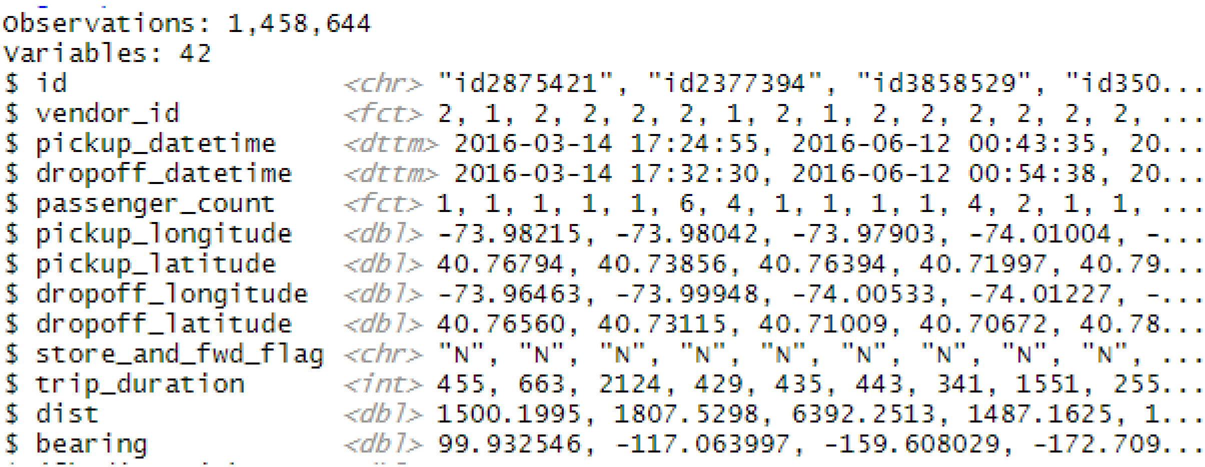
Géométries — Les éléments visuels

Tous les éléments de Grammaire



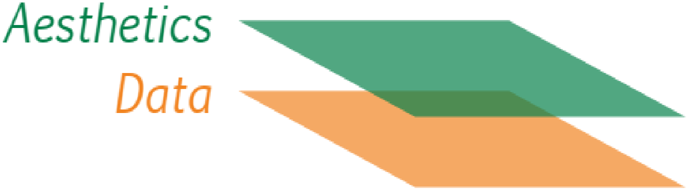
Données	La base de données
esthétiques	Les échelles du graphiques
Géométries	Les éléments visuels
Facettes	Graphique pour chaque cas
Statistiques	Représentation différente des données
Coordonnées	L'espace du dessin
Thèmes	les éléments non relatifs aux données

Couche : données (Data)



```
Observations: 1,458,644
Variables: 42
$ id                 <chr> "id2875421", "id2377394", "id3858529", "id350...
$ vendor_id          <fct> 2, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 2, 2, 2, ...
$ pickup_datetime    <dttm> 2016-03-14 17:24:55, 2016-06-12 00:43:35, 20...
$ dropoff_datetime   <dttm> 2016-03-14 17:32:30, 2016-06-12 00:54:38, 20...
$ passenger_count    <fct> 1, 1, 1, 1, 1, 6, 4, 1, 1, 1, 1, 4, 2, 1, 1, ...
$ pickup_longitude   <dbl> -73.98215, -73.98042, -73.97903, -74.01004, -...
$ pickup_latitude    <dbl> 40.76794, 40.73856, 40.76394, 40.71997, 40.79...
$ dropoff_longitude  <dbl> -73.96463, -73.99948, -74.00533, -74.01227, -...
$ dropoff_latitude   <dbl> 40.76560, 40.73115, 40.71009, 40.70672, 40.78...
$ store_and_fwd_flag <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N", ...
$ trip_duration      <int> 455, 663, 2124, 429, 435, 443, 341, 1551, 255...
$ dist               <dbl> 1500.1995, 1807.5298, 6392.2513, 1487.1625, 1...
$ bearing            <dbl> 99.932546, -117.063997, -159.608029, -172.709...
```

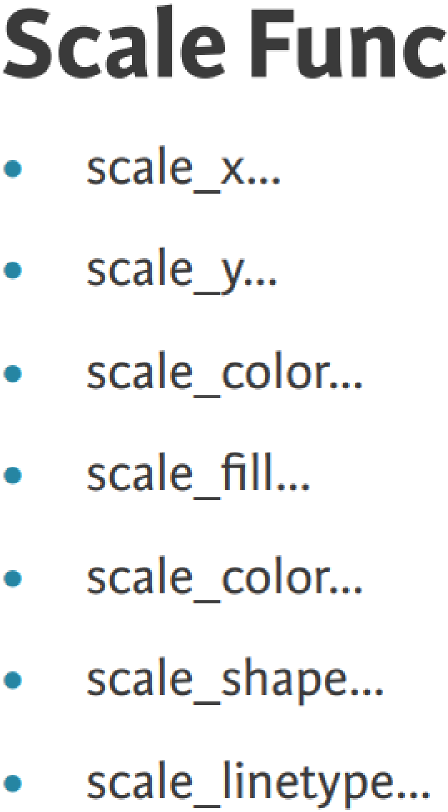
Couche : esthétiques (Aesthetics)



Couche : esthétiques (Aesthetics)

Esthétiques	Discription
x	Position de l’axe X
Y	Position de l’axe Y
color, col, colour	Couleur des points ou des autres formes
fill	Couleur de remplissage
size	Diamètre des points, épaisseur des lignes
alpha	Transparence
linetype	Style d’une ligne
Labels	Texte sur le graphe ou sur les axes
shape	Forme

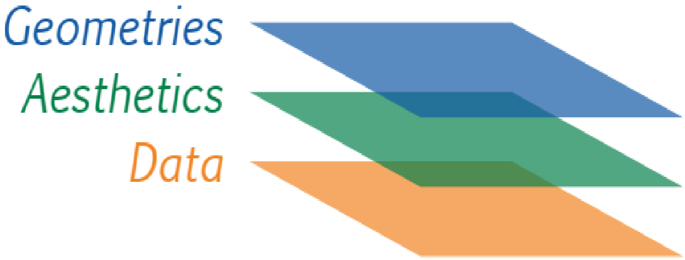
Couche : esthétiques (Aesthetics)



Scale Functions

- scale_x...
- scale_y...
- scale_color...
- scale_fill...
- scale_color...
- scale_shape...
- scale_linetype...

Couche : géométries (Geometries)



Couche : géométries (Geometries)

abline

area

bar

bin2d

blank

boxplot

contour

crossbar

density

density2d

dotplot

errorbar

errorbarh

freqpoly

hex

histogram

hline

jitter

line

linerange

map

path

point

pointrange

polygon

quantile

raster

rect

ribbon

rug

segment

smooth

step

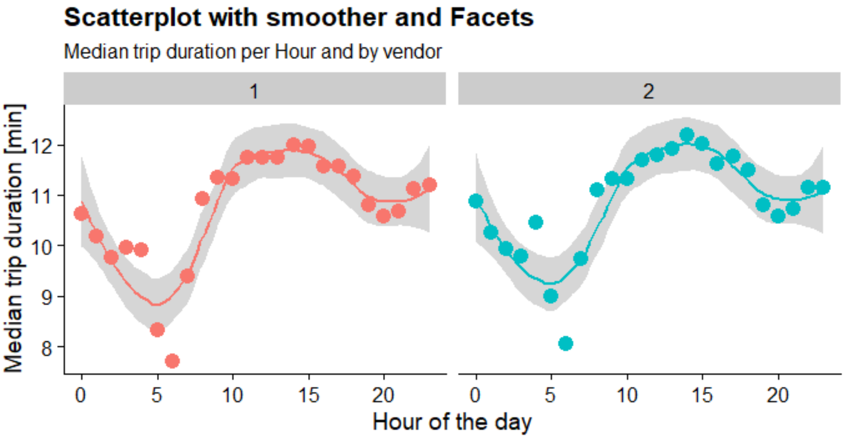
text

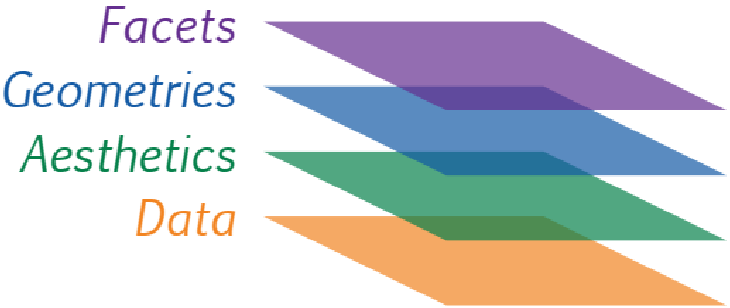
tile

violin

vline

Couche : facettes (Facets)



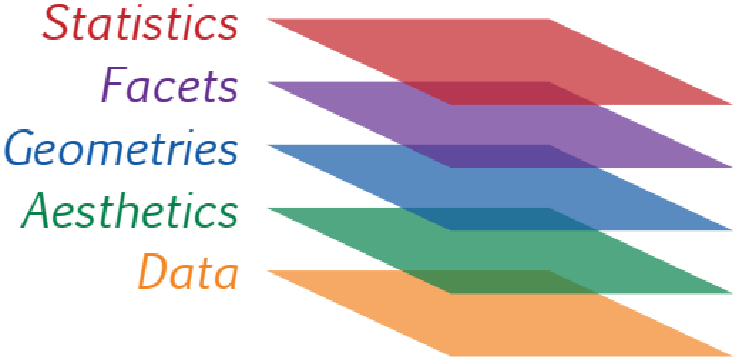

Facets

Geometries

Aesthetics

Data

Couche : statistiques (Statistics)

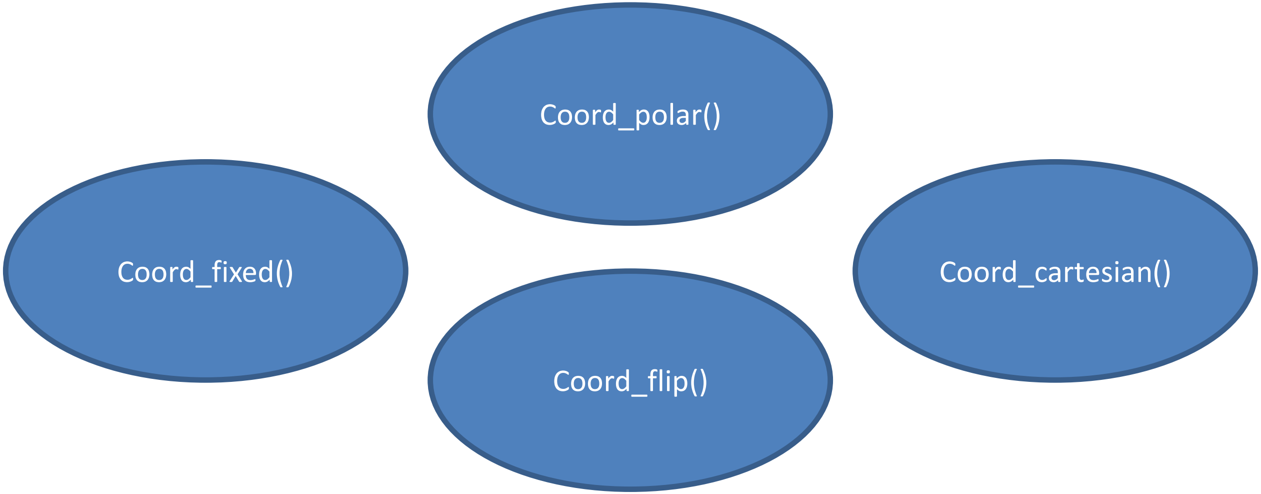


Couche : statistiques (Statistics)

stat_	geom_
stat_bin()	geom_histogram(), geom_bar(), geom_freqpoly()
stat_smooth()	geom_smooth()
stat_bindot()	geom_dotplot()
stat_boxplot()	geom_boxplot()
stat_bin2d()	geom_bin2d()
stat_binhx()	geom_hex()

Stat_	Description
stat_summary()	Statistiques sommaires des valeurs de y pour des valeurs choisies de x
stat_function()	Calcule des valeurs de y à partir d'une fonction des valeurs de x
stat_qq()	Calculs pour qq-plot

Couche : coordonnées (Coordinates)

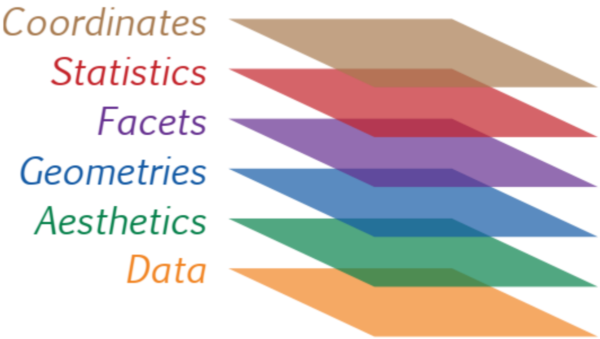


Coord_polar()

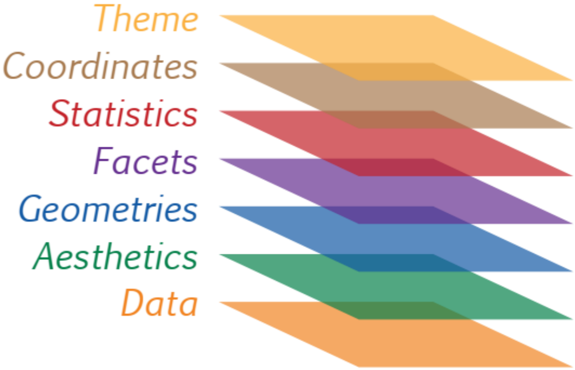
Coord_fixed()

Coord_cartesian()

Coord_flip()

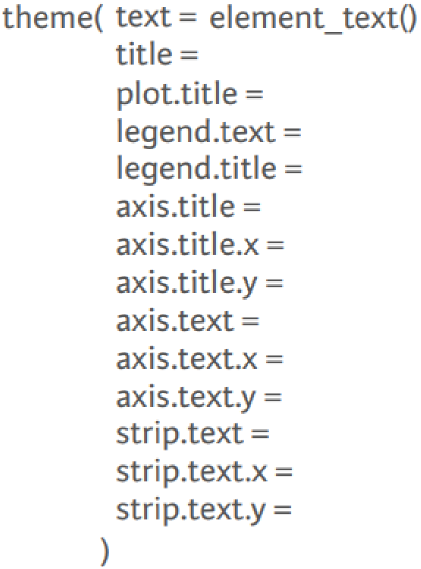


Couche : thèmes (Theme)

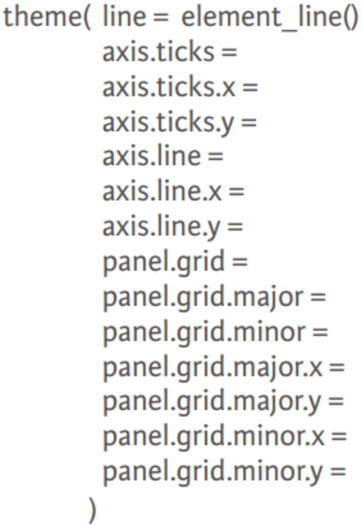


Couche : thèmes (Theme)

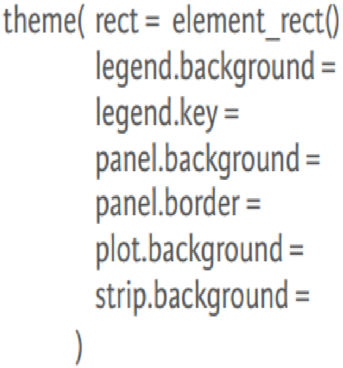
text	element_text()
line	element_line()
rectangle	element_rect()



```
theme( text = element_text()
       title =
       plot.title =
       legend.text =
       legend.title =
       axis.title =
       axis.title.x =
       axis.title.y =
       axis.text =
       axis.text.x =
       axis.text.y =
       strip.text =
       strip.text.x =
       strip.text.y =
     )
```



```
theme( line = element_line()
       axis.ticks =
       axis.ticks.x =
       axis.ticks.y =
       axis.line =
       axis.line.x =
       axis.line.y =
       panel.grid =
       panel.grid.major =
       panel.grid.minor =
       panel.grid.major.x =
       panel.grid.major.y =
       panel.grid.minor.x =
       panel.grid.minor.y =
     )
```



```
theme( rect = element_rect()
       legend.background =
       legend.key =
       panel.background =
       panel.border =
       plot.background =
       strip.background =
     )
```

3

La base de données utilisée

NYC taxi trip duration dataset

1,5 Millions observations sur une période 6 mois

Compétition kaggle

11 variables dans la base principale

Pickup/dropoff time, passenger count, pickup/dropoff location, trip duration, vendor id, …

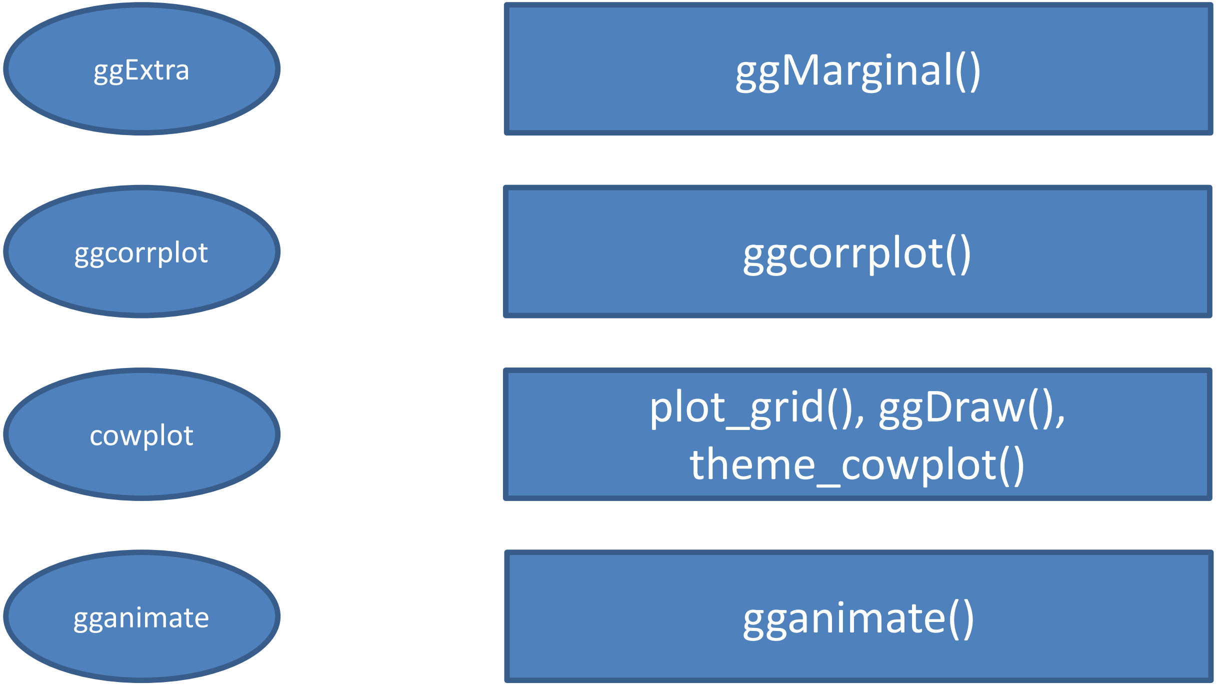
Plusieurs données externes combinées, 42 variables

Weather, fastest routes, feature engineering

4

Les packages complémentaires

packages

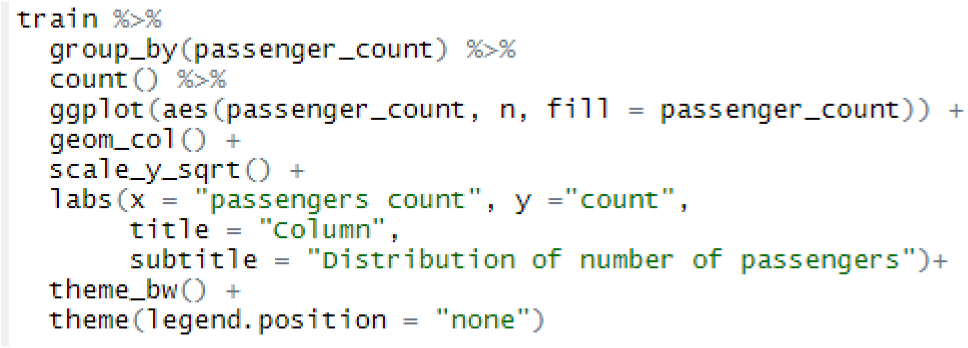


ggExtra	ggMarginal()
ggcorrplot	ggcorrplot()
cowplot	plot_grid(), ggDraw(), theme_cowplot()
gganimate	gganimate()

5

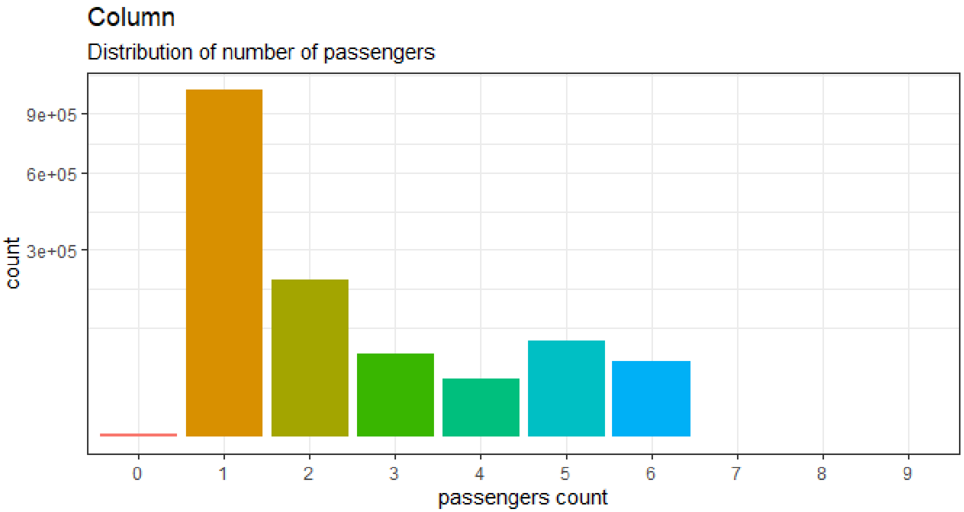
Application : utilisation des graphiques sur la base de données

Diagramme en bâton

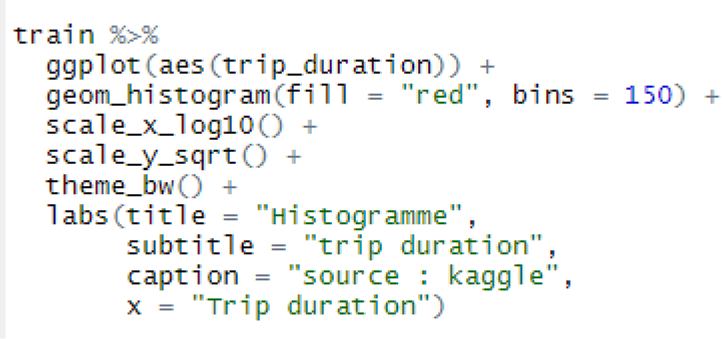


```
train %>%
  group_by(passenger_count) %>%
  count() %>%
  ggplot(aes(passenger_count, n, fill = passenger_count)) +
  geom_col() +
  scale_y_sqrt() +
  labs(x = "passengers count", y ="count",
       title = "Column",
       subtitle = "Distribution of number of passengers")+
  theme_bw() +
  theme(legend.position = "none")
```

- **geom_col()**
- **scale_y_sqrt()**
- **theme_bw()**

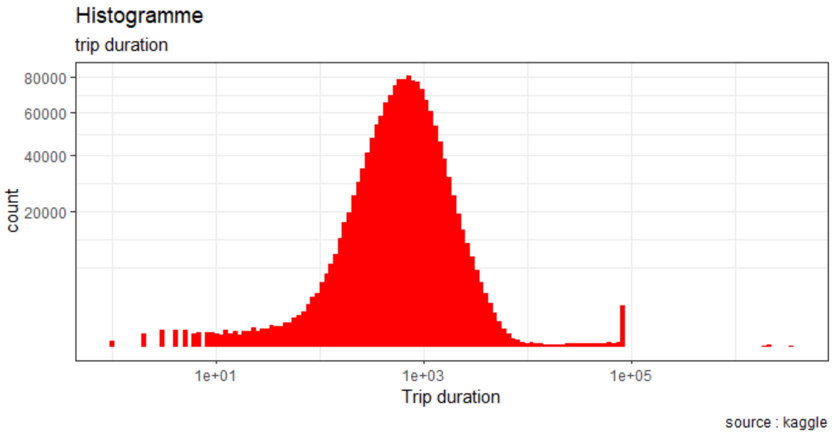


Histogramme

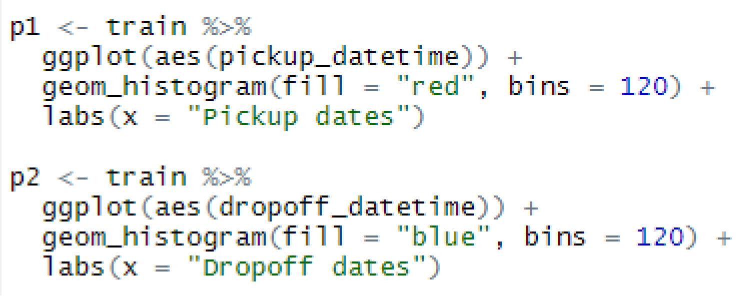


```
train %>%
  ggplot(aes(trip_duration)) +
  geom_histogram(fill = "red", bins = 150) +
  scale_x_log10() +
  scale_y_sqrt() +
  theme_bw() +
  labs(title = "Histogramme",
       subtitle = "trip duration",
       caption = "source : kaggle",
       x = "Trip duration")
```

- **geom_histogram()**
- **scale_y_log10()**

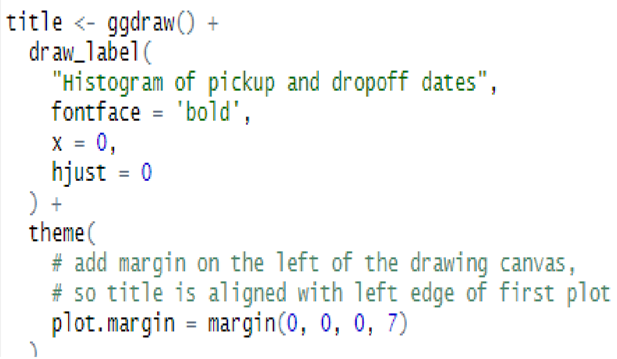


Histogramme (II)

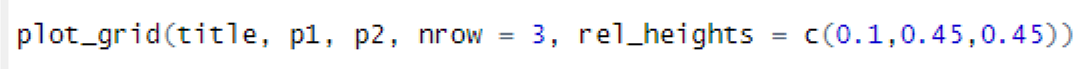


```r
p1 <- train %>%
  ggplot(aes(pickup_datetime)) +
  geom_histogram(fill = "red", bins = 120) +
  labs(x = "Pickup dates")

p2 <- train %>%
  ggplot(aes(dropoff_datetime)) +
  geom_histogram(fill = "blue", bins = 120) +
  labs(x = "Dropoff dates")
```

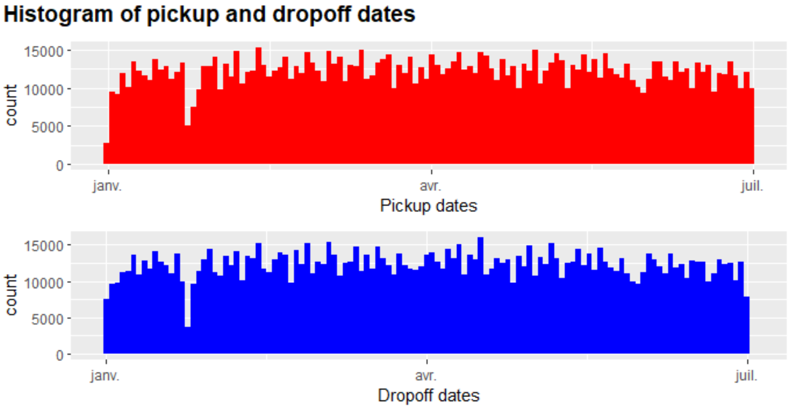


```r
title <- ggdraw() +
  draw_label(
    "Histogram of pickup and dropoff dates",
    fontface = 'bold',
    x = 0,
    hjust = 0
  ) +
  theme(
    # add margin on the left of the drawing canvas,
    # so title is aligned with left edge of first plot
    plot.margin = margin(0, 0, 0, 7)
  )
```

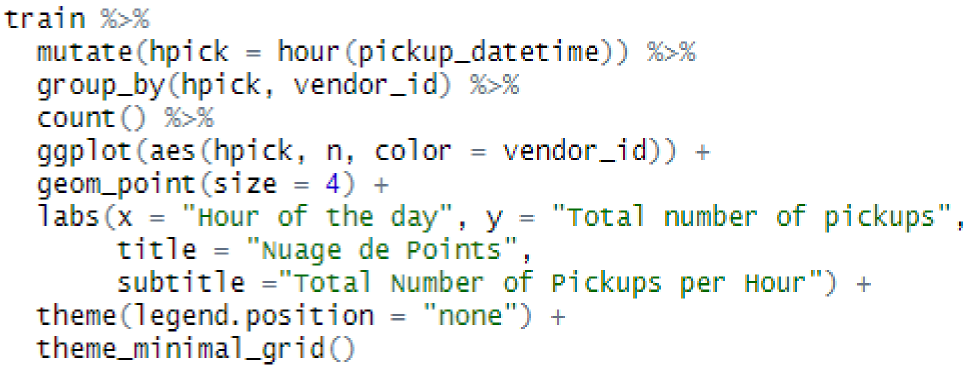


```r
plot_grid(title, p1, p2, nrow = 3, rel_heights = c(0.1,0.45,0.45))
```

- **ggDraw()**
- **Plot_grid()**



Nuage de Points



```
train %>%
  mutate(hpick = hour(pickup_datetime)) %>%
  group_by(hpick, vendor_id) %>%
  count() %>%
  ggplot(aes(hpick, n, color = vendor_id)) +
  geom_point(size = 4) +
  labs(x = "Hour of the day", y = "Total number of pickups",
       title = "Nuage de Points",
       subtitle ="Total Number of Pickups per Hour") +
  theme(legend.position = "none") +
  theme_minimal_grid()
```

- **geom_point()**
- **theme_minial_grid()**

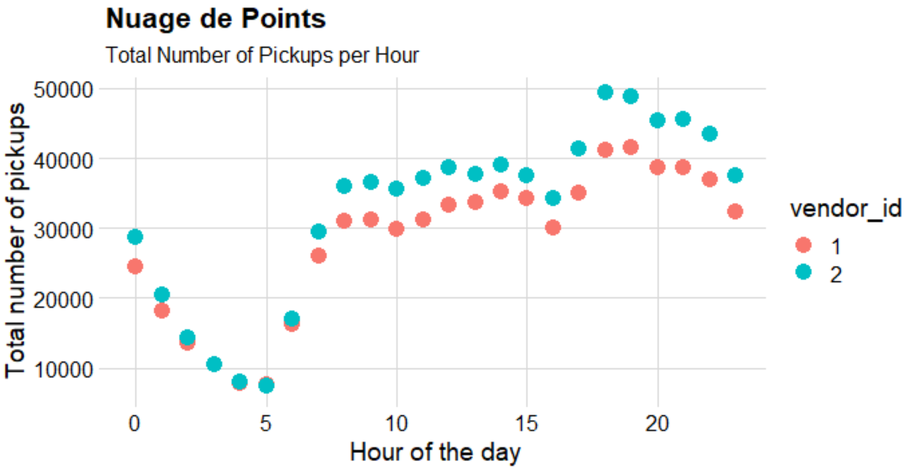
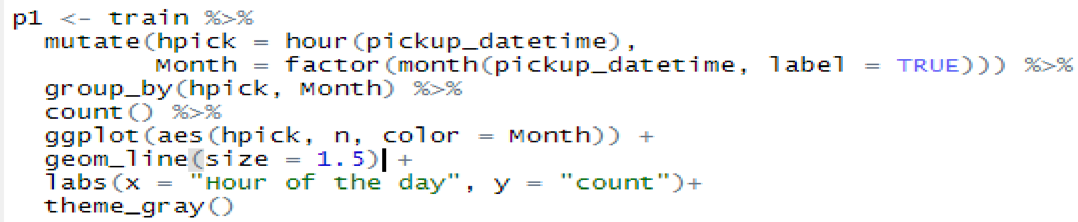
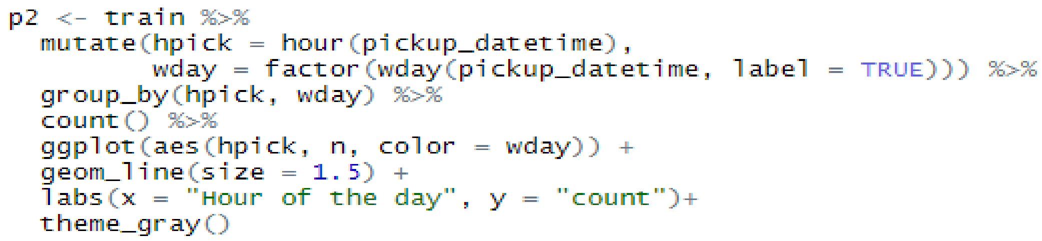


Diagramme linéaire

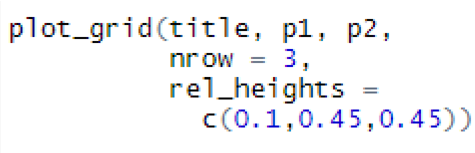


```
p1 <- train %>%
  mutate(hpick = hour(pickup_datetime),
         Month = factor(month(pickup_datetime, label = TRUE))) %>%
  group_by(hpick, Month) %>%
  count() %>%
  ggplot(aes(hpick, n, color = Month)) +
  geom_line(size = 1.5) +
  labs(x = "Hour of the day", y = "count")+
  theme_gray()
```

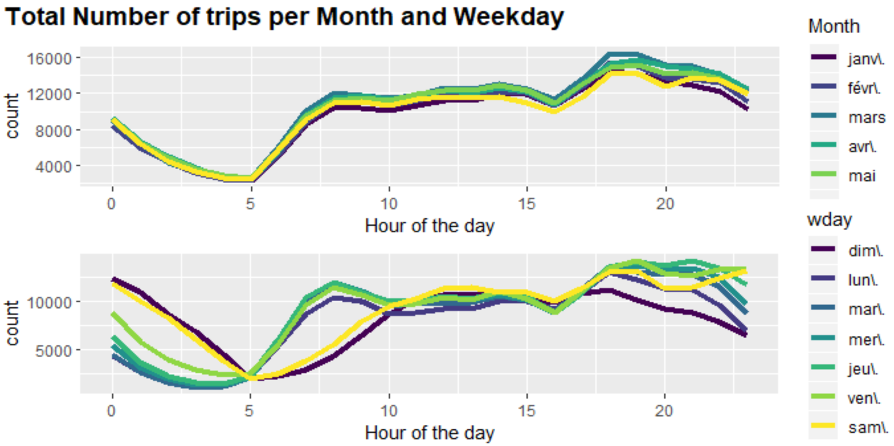


```
p2 <- train %>%
  mutate(hpick = hour(pickup_datetime),
         wday = factor(wday(pickup_datetime, label = TRUE))) %>%
  group_by(hpick, wday) %>%
  count() %>%
  ggplot(aes(hpick, n, color = wday)) +
  geom_line(size = 1.5) +
  labs(x = "Hour of the day", y = "count")+
  theme_gray()
```

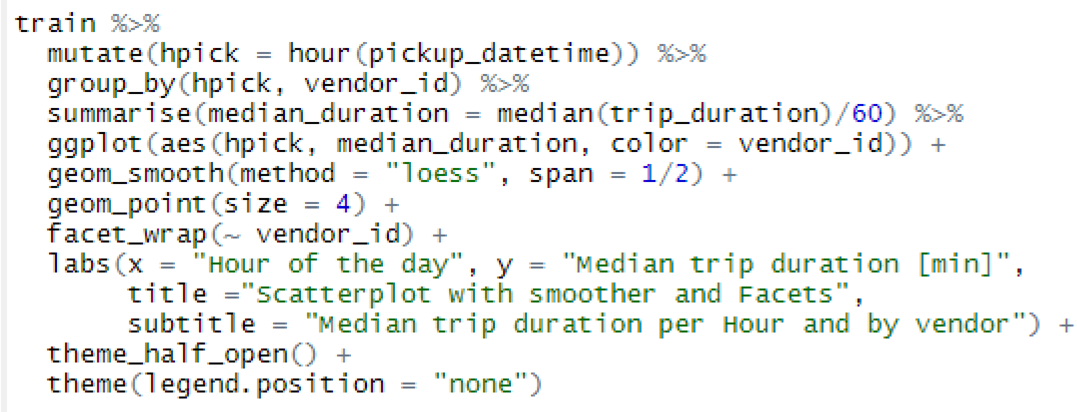
- **geom_line()**
- **theme_gray()**



```
plot_grid(title, p1, p2,
          nrow = 3,
          rel_heights =
            c(0.1,0.45,0.45))
```

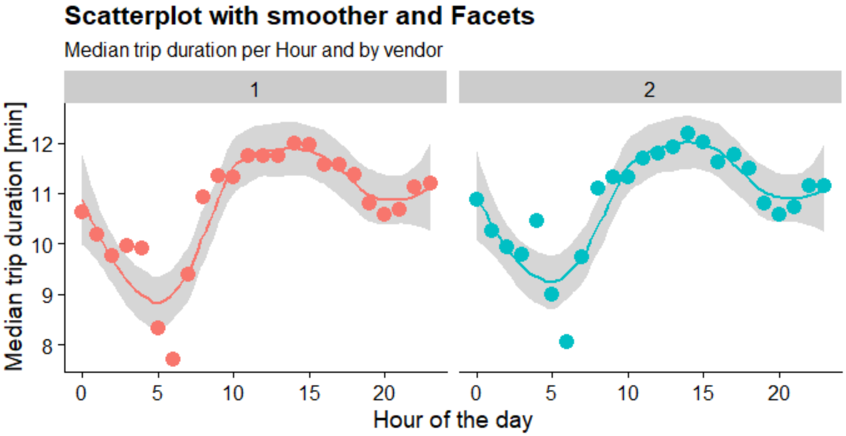


Statistiques et Facettes



```
train %>%
  mutate(hpick = hour(pickup_datetime)) %>%
  group_by(hpick, vendor_id) %>%
  summarise(median_duration = median(trip_duration)/60) %>%
  ggplot(aes(hpick, median_duration, color = vendor_id)) +
  geom_smooth(method = "loess", span = 1/2) +
  geom_point(size = 4) +
  facet_wrap(~ vendor_id) +
  labs(x = "Hour of the day", y = "Median trip duration [min]",
       title ="Scatterplot with smoother and Facets",
       subtitle = "Median trip duration per Hour and by vendor") +
  theme_half_open() +
  theme(legend.position = "none")
```

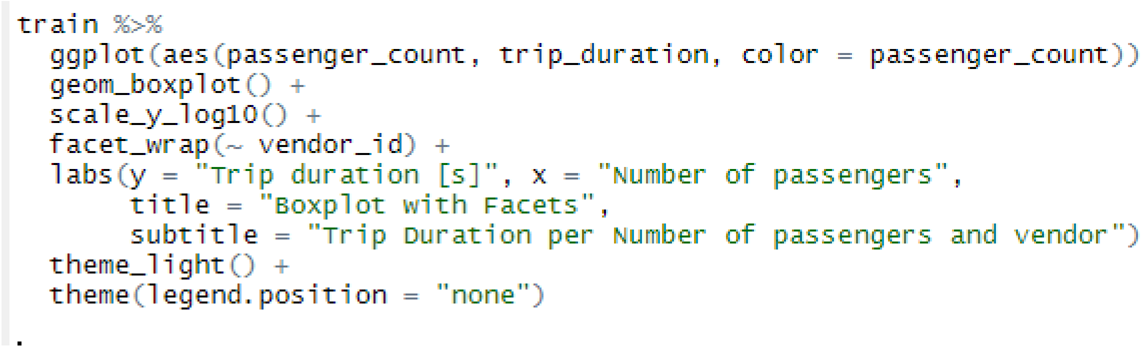
- **geom_smooth()**
- **facet_wrap()**
- **Theme_half_open()**



Scatterplot with smoother and Facets

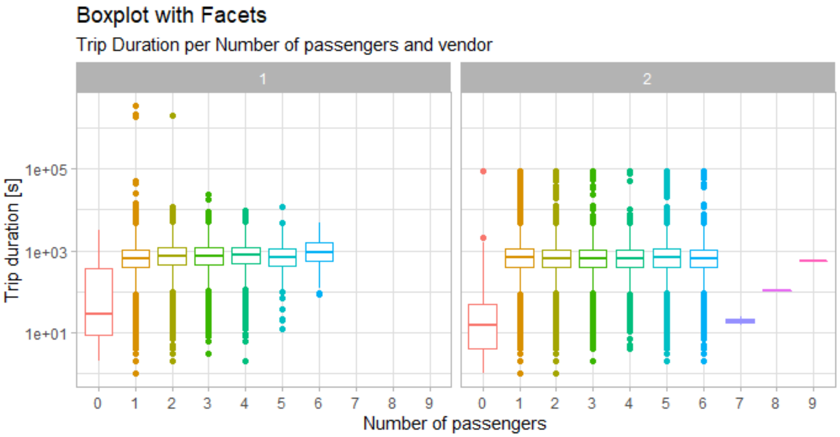
Median trip duration per Hour and by vendor

Boite à Moustaches (Boxplot)

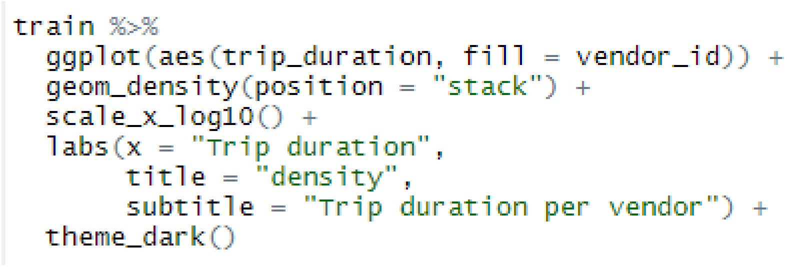


```
train %>%
  ggplot(aes(passenger_count, trip_duration, color = passenger_count))
  geom_boxplot() +
  scale_y_log10() +
  facet_wrap(~ vendor_id) +
  labs(y = "Trip duration [s]", x = "Number of passengers",
       title = "Boxplot with Facets",
       subtitle = "Trip Duration per Number of passengers and vendor")
  theme_light() +
  theme(legend.position = "none")
```

- **geom_boxplotl()**
- **theme_light()**

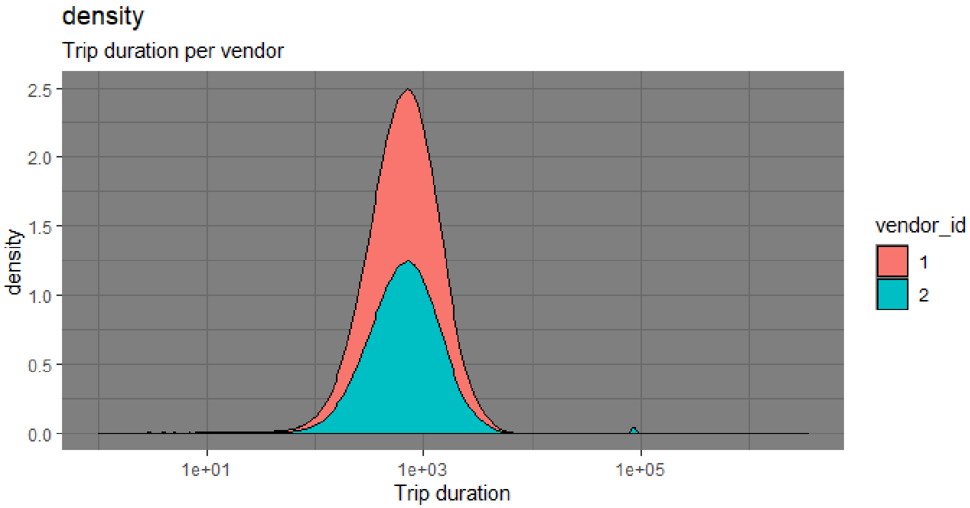


Densités

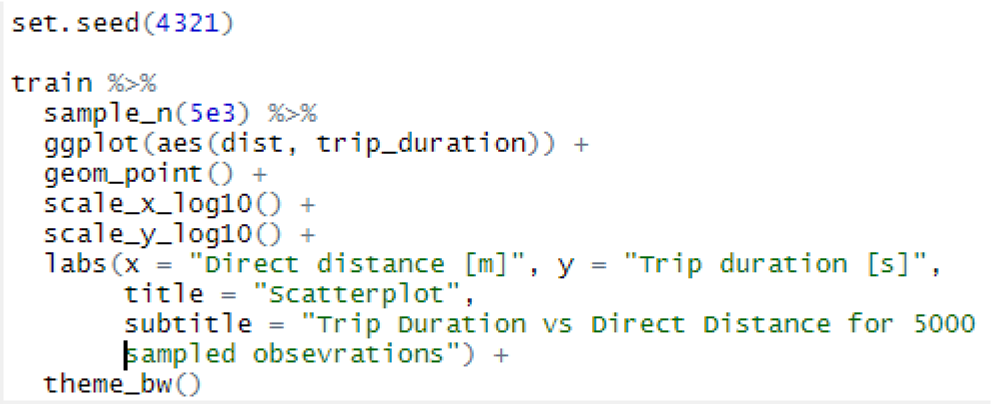


```
train %>%
  ggplot(aes(trip_duration, fill = vendor_id)) +
  geom_density(position = "stack") +
  scale_x_log10() +
  labs(x = "Trip duration",
       title = "density",
       subtitle = "Trip duration per vendor") +
  theme_dark()
```

- **geom_density()**
- **theme_dark()**



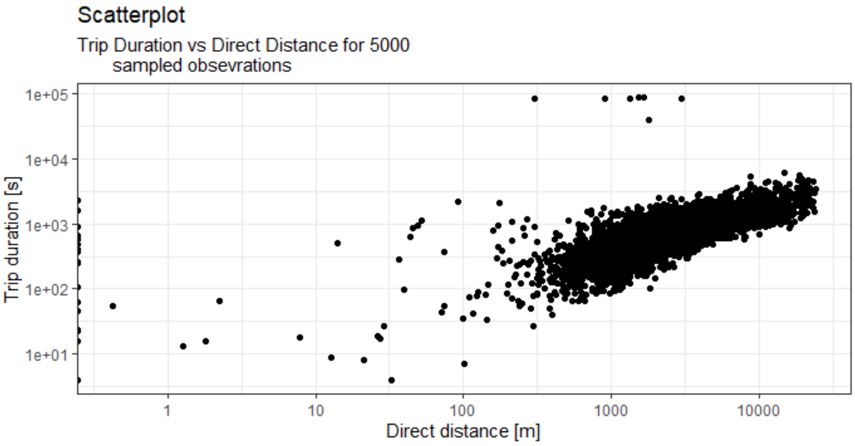
Nuage de Points



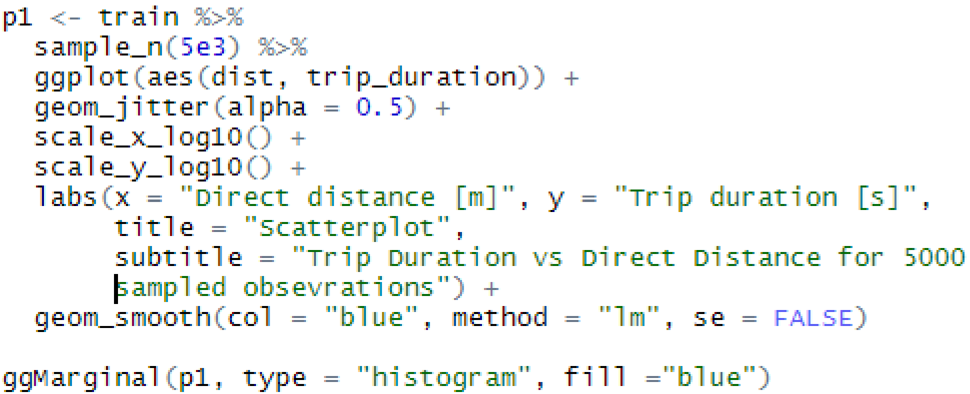
```
set.seed(4321)

train %>%
  sample_n(5e3) %>%
  ggplot(aes(dist, trip_duration)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Direct distance [m]", y = "Trip duration [s]",
       title = "Scatterplot",
       subtitle = "Trip Duration vs Direct Distance for 5000
       sampled obsevrations") +
  theme_bw()
```

- **geom_point()**
- **theme_bw()**



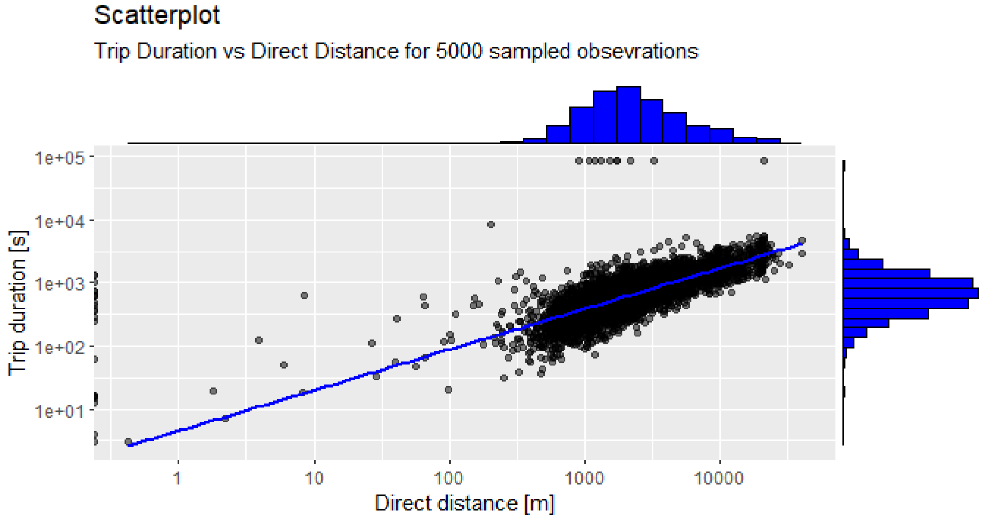
Nuage de Points (Jitter) à distribution marginale



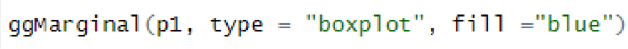
```
p1 <- train %>%
  sample_n(5e3) %>%
  ggplot(aes(dist, trip_duration)) +
  geom_jitter(alpha = 0.5) +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Direct distance [m]", y = "Trip duration [s]",
       title = "Scatterplot",
       subtitle = "Trip Duration vs Direct Distance for 5000
       sampled obsevrations") +
  geom_smooth(col = "blue", method = "lm", se = FALSE)

ggMarginal(p1, type = "histogram", fill ="blue")
```

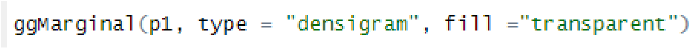
- **geom_jitter()**
- **ggmarginal()**



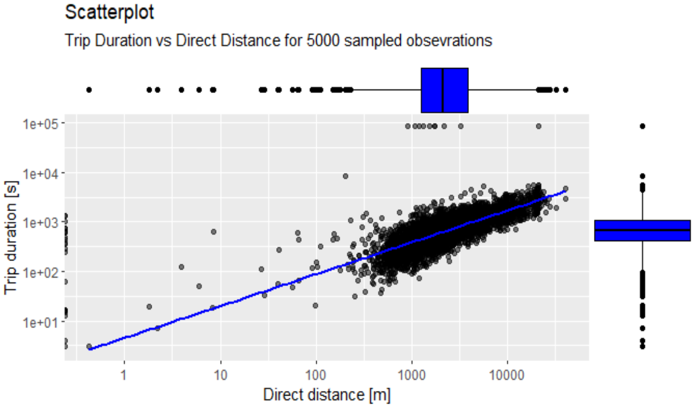
Nuage de Points (Jitter) à distribution marginale (II)

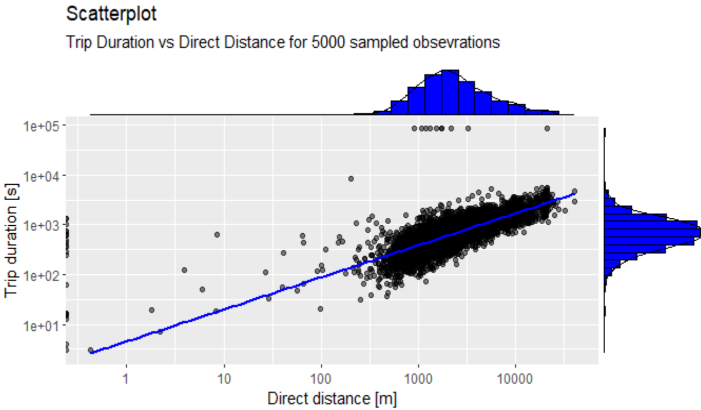


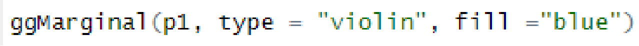
```
ggMarginal(p1, type = "boxplot", fill ="blue")
```



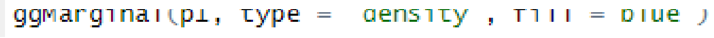
```
ggMarginal(p1, type = "densigram", fill ="transparent")
```



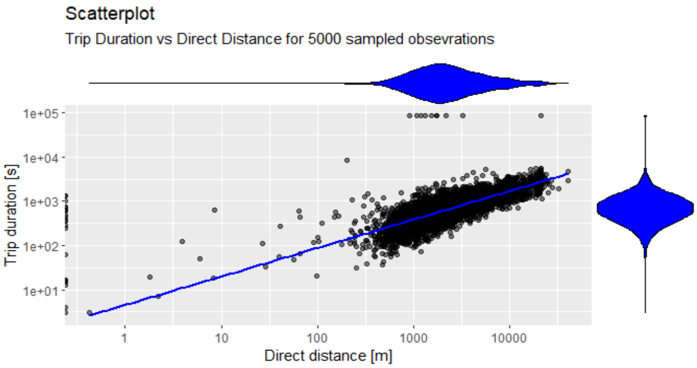


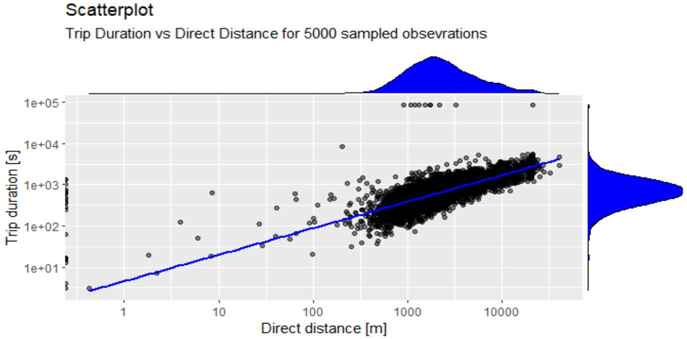


```
ggMarginal(p1, type = "violin", fill ="blue")
```

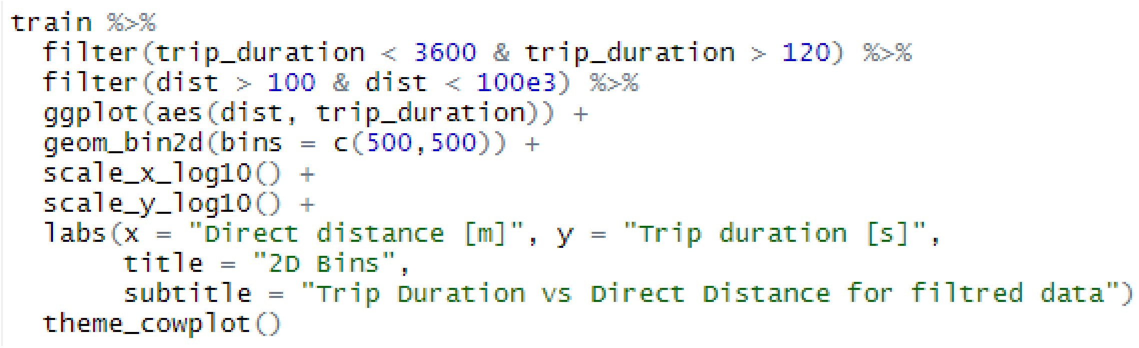


```
ggMarginal(p1, type = "density", fill = "blue")
```



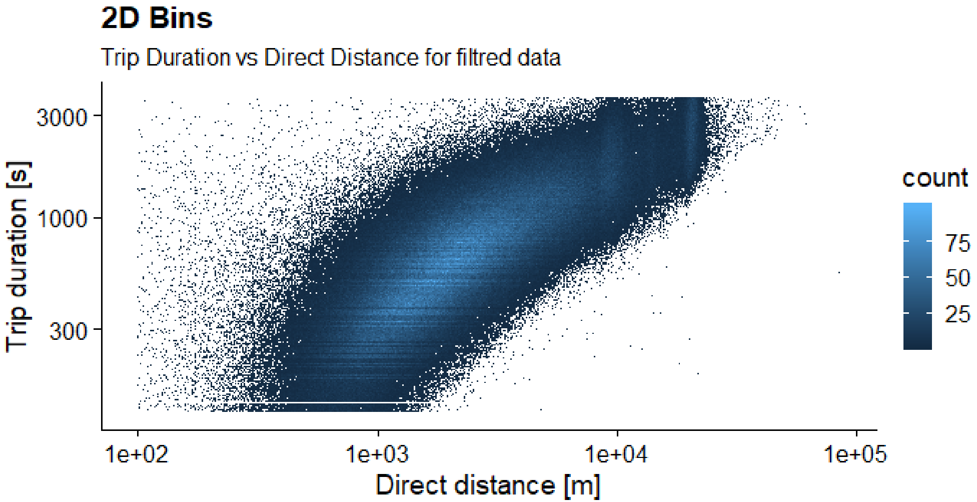


Alternatives au nuage de points : bins

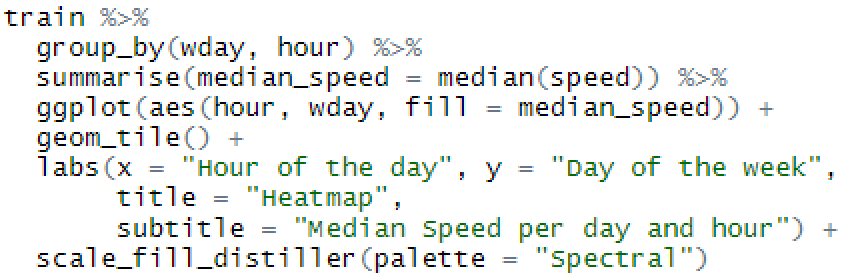


```
train %>%
  filter(trip_duration < 3600 & trip_duration > 120) %>%
  filter(dist > 100 & dist < 100e3) %>%
  ggplot(aes(dist, trip_duration)) +
  geom_bin2d(bins = c(500,500)) +
  scale_x_log10() +
  scale_y_log10() +
  labs(x = "Direct distance [m]", y = "Trip duration [s]",
       title = "2D Bins",
       subtitle = "Trip Duration vs Direct Distance for filtred data")
  theme_cowplot()
```

- **geom_bin2d()**
- **theme_cowplot()**



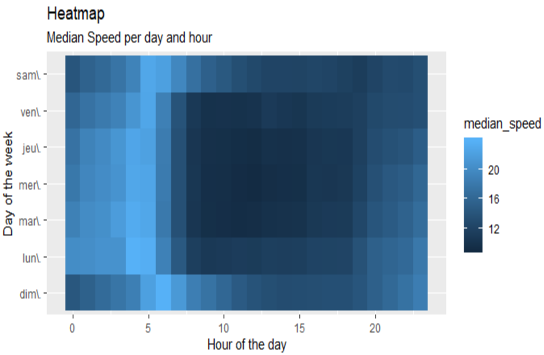
Cartes thématiques (Heat maps)

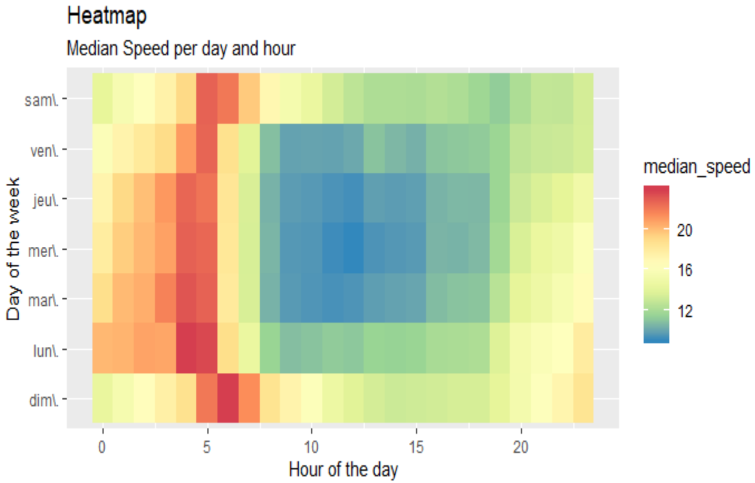


```
train %>%
  group_by(wday, hour) %>%
  summarise(median_speed = median(speed)) %>%
  ggplot(aes(hour, wday, fill = median_speed)) +
  geom_tile() +
  labs(x = "Hour of the day", y = "Day of the week",
       title = "Heatmap",
       subtitle = "Median Speed per day and hour") +
  scale_fill_distiller(palette = "Spectral")
```

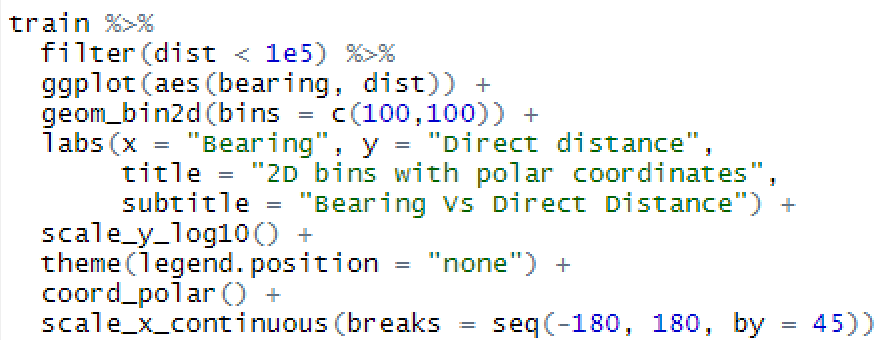
- **geom_tile()**
- **Scale_fill_distiller()**

- **Scale_fill_continuous()**



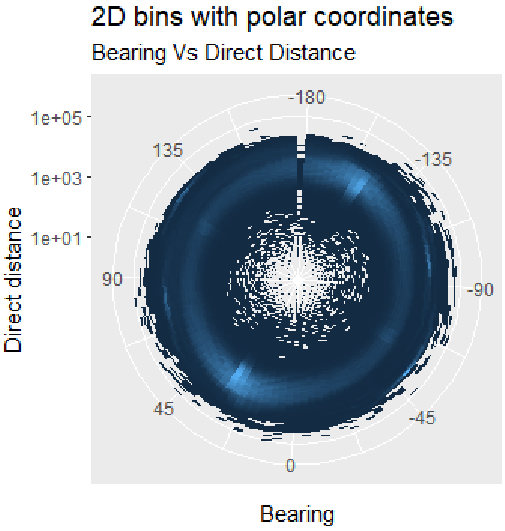


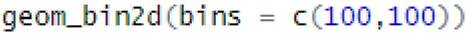
Utilisation des coordonnées : polaires



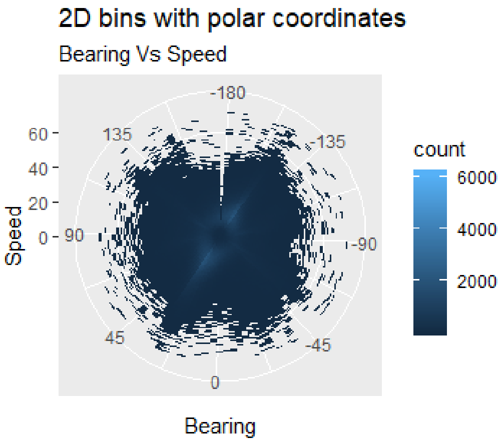
```
train %>%
  filter(dist < 1e5) %>%
  ggplot(aes(bearing, dist)) +
  geom_bin2d(bins = c(100,100)) +
  labs(x = "Bearing", y = "Direct distance",
       title = "2D bins with polar coordinates",
       subtitle = "Bearing Vs Direct Distance") +
  scale_y_log10() +
  theme(legend.position = "none") +
  coord_polar() +
  scale_x_continuous(breaks = seq(-180, 180, by = 45))
```

- **scale_x_continuous()**
- **coord_polar()**

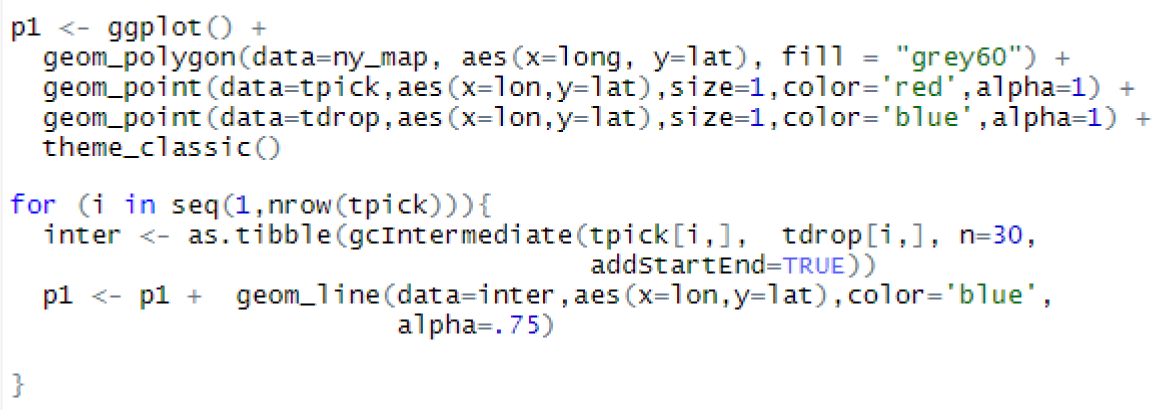




```
geom_bin2d(bins = c(100,100))
```



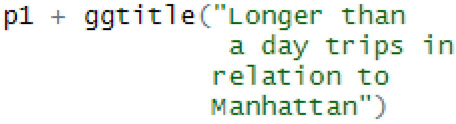
Utilisation des boucles pour dessiner un graphique des différents itinéraires



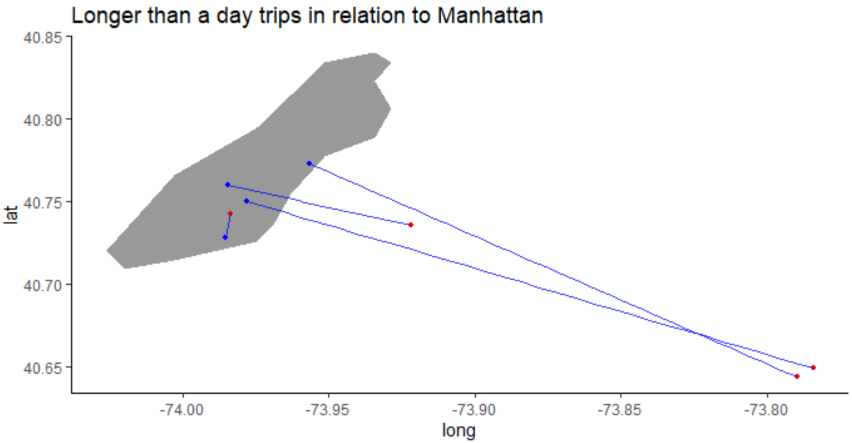
```
p1 <- ggplot() +
  geom_polygon(data=ny_map, aes(x=long, y=lat), fill = "grey60") +
  geom_point(data=tpick,aes(x=lon,y=lat),size=1,color='red',alpha=1) +
  geom_point(data=tdrop,aes(x=lon,y=lat),size=1,color='blue',alpha=1) +
  theme_classic()

for (i in seq(1,nrow(tpick))){
  inter <- as.tibble(gcIntermediate(tpick[i,],  tdrop[i,], n=30,
                                  addStartEnd=TRUE))
  p1 <- p1 +  geom_line(data=inter,aes(x=lon,y=lat),color='blue',
                        alpha=.75)

}
```

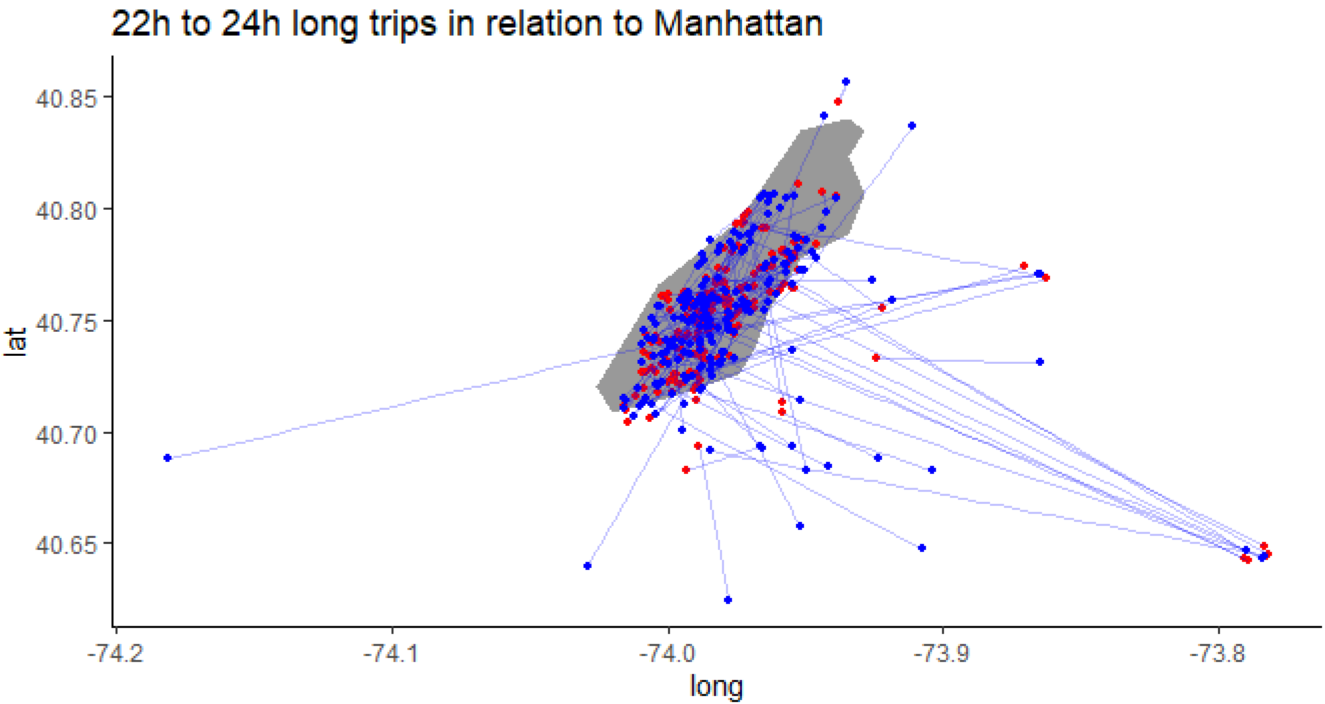


```
p1 + ggtitle("Longer than
              a day trips in
             relation to
             Manhattan")
```

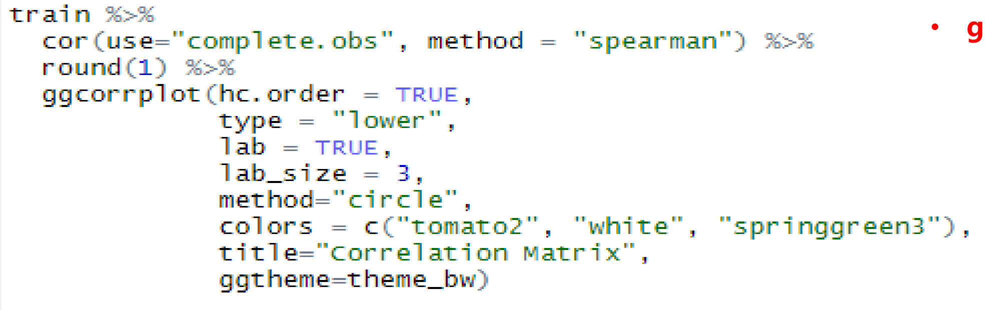


- **geom_polygon()**
- **Theme_classic()**
- **ggtitle()**

Utilisation des boucles pour dessiner un graphique des différents itinéraires (II)

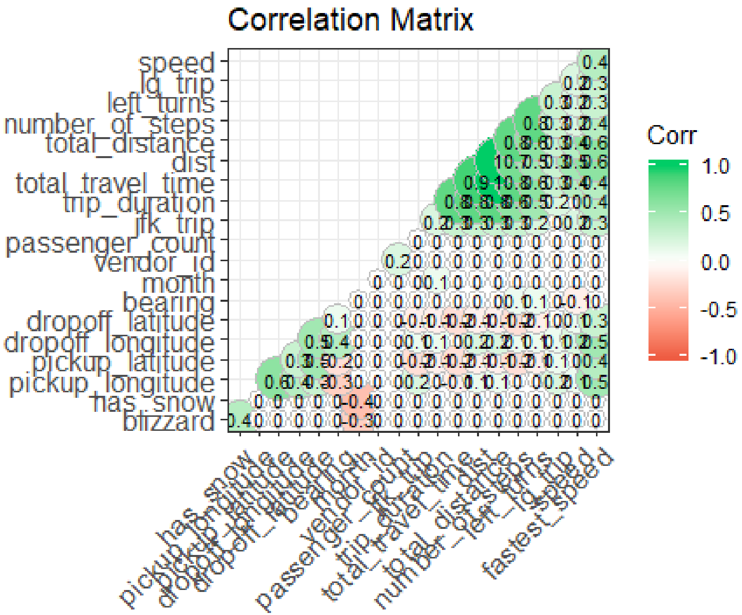


Matrice de Corrélation



```
train %>%
  cor(use="complete.obs", method = "spearman") %>%
  round(1) %>%
  ggcorrplot(hc.order = TRUE,
             type = "lower",
             lab = TRUE,
             lab_size = 3,
             method="circle",
             colors = c("tomato2", "white", "springgreen3"),
             title="Correlation Matrix",
             ggtheme=theme_bw)
```

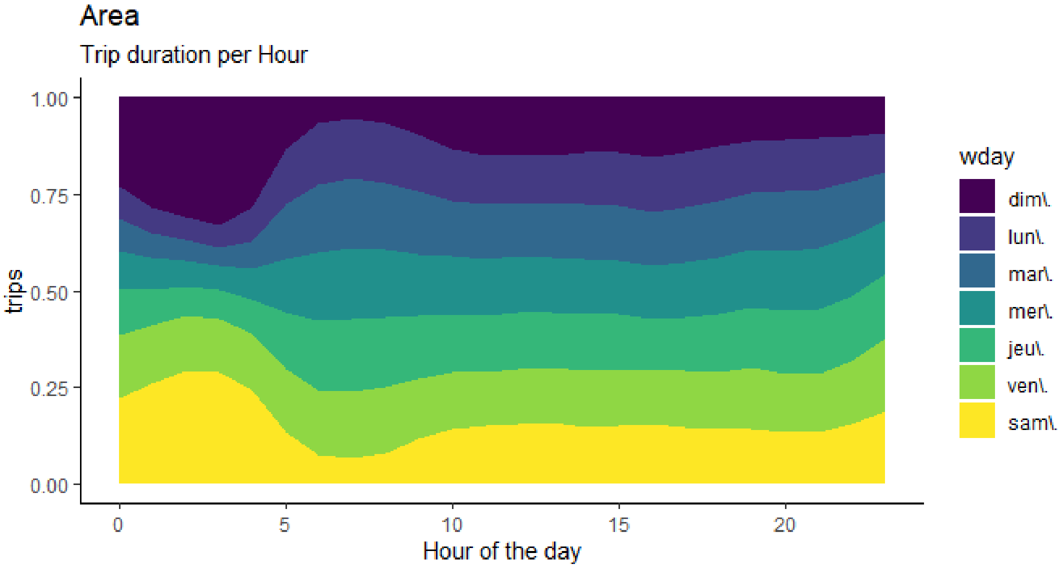
- **ggcorrplot()**



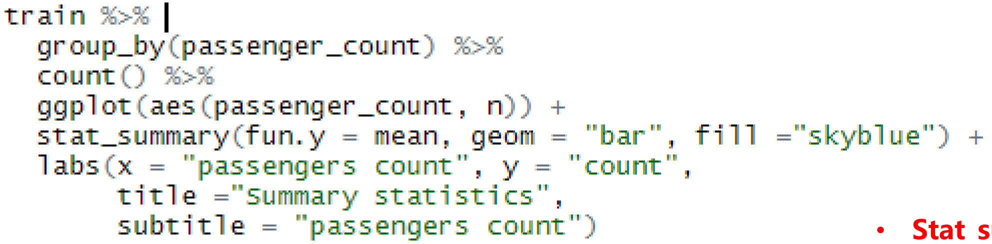
Surface

```
train %>%
  mutate(hpick = hour(pickup_datetime),
         wday = factor(wday(pickup_datetime, label = TRUE))) %>%
  group_by(hpick, wday) %>%
  count() %>%
  ggplot(aes(hpick, n, fill = wday)) +
  geom_area(position = "fill") +
  labs(x = "Hour of the day", y = "trips",
       title = "Area",
       subtitle = "Trip duration per Hour")+
  theme_classic()
```

- **geom_area()**

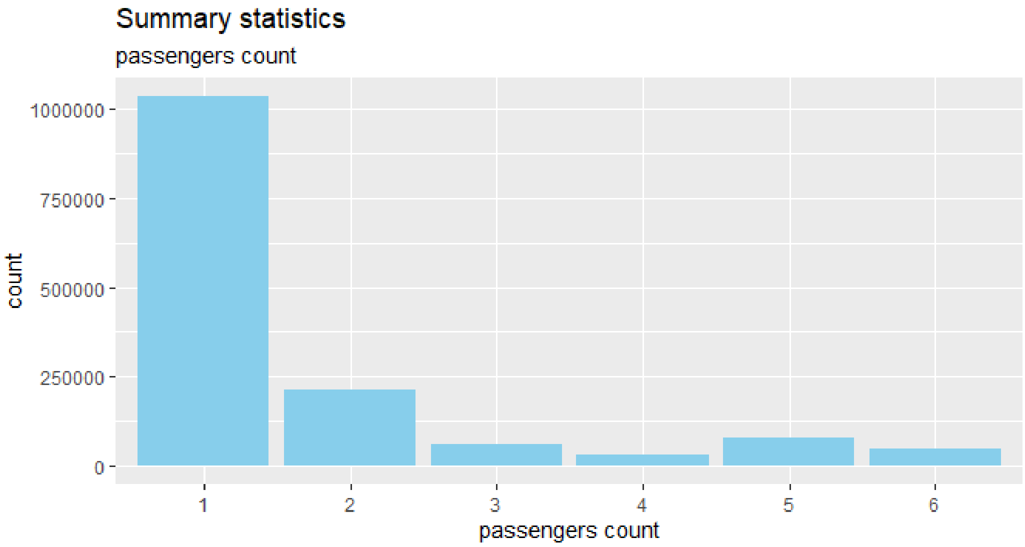


Utilisation de la couche « statistiques »



```
train %>%
  group_by(passenger_count) %>%
  count() %>%
  ggplot(aes(passenger_count, n)) +
  stat_summary(fun.y = mean, geom = "bar", fill ="skyblue") +
  labs(x = "passengers count", y = "count",
       title ="Summary statistics",
       subtitle = "passengers count")
```

- **Stat_summary()**



Thanks

Q&A