

Logiciel R et programmation

Exercices



Partie 4 : Graphiques

Exercice 1 (Créer un graphique simple, modifier son aspect)

1. Charger le *package* `ggplot2`, et utiliser la fonction `data()` pour charger en mémoire le jeu de données `economics`. Consulter la page d'aide de ce jeu de données pour prendre connaissance de son contenu ;

```
library(ggplot2)
data(economics)
?economics
```

2. À l'aide de la fonction `ggplot()`, représenter les dépenses personnelles de consommation (`pce`) en fonction de la date (`date`). Les observations doivent être connectées par une ligne.

```
ggplot(data = economics, aes(x = date, y = pce)) +
  geom_line()
```

3. Modifier le graphique de la question précédente de manière à ce que la couleur de la ligne soit `dodger blue` et définir la taille de la ligne à 0.5. Stocker le résultat dans un objet que l'on appellera `p_1` ;

```
p_1 <- ggplot(data = economics, aes(x = date, y = pce)) +
  geom_line(colour = "dodger blue", size = 0.5)
```

4. Ajouter une couche au graphique `p_1` pour modifier les titres des axes (les retirer), et ajouter le titre suivant : "Personal Consumption Expenditures (billions dollars)".

1. ewen.gallic[at]gmail.com

```
p_1 <-
  p_1 +
  xlab("") + ylab("") +
  ggtitle("Personal Consumption Expenditures (billions dollars)")
```

- Utiliser la fonction `date_breaks()` du *package* `scales` pour modifier l'échelle des abscisses de `p_1`, afin que les étiquettes des marques soient affichées tous les 5 ans; à l'aide de la fonction `date_format()`, modifier le format de ces étiquettes pour que seule l'année des dates s'affiche;

```
library(scales)
p_1 +
  scale_x_date(breaks = date_breaks("5 year"),
              labels = date_format("%Y"))
```

Exercice 2 (Créer un graphique avec plusieurs courbes, modifier son aspect)

- Charger le *package* `ggplot2`, et utiliser la fonction `data()` pour charger en mémoire le jeu de données `economics`. Consulter la page d'aide de ce jeu de données pour prendre connaissance de son contenu;

```
library(ggplot2)
data(economics)
?economics
```

- Charger les *packages* `dplyr` et `tidyr`. Sélectionner les variables `date`, `psavert` et `uempmed` dans le tableau de données `economics` et utiliser la fonction `gather()` sur le résultat pour obtenir un tableau dans lequel chaque ligne indiquera la valeur (`value`) pour une variable donnée (`key`) à une date donnée (`date`). Stocker le résultat dans un objet que l'on appellera `df`;

```
library(dplyr)
library(tidyr)
df <-
  economics %>%
  select(date, psavert, uempmed) %>%
  gather(key, value, -date)
```

- Sur un même graphique, représenter à l'aide de lignes, l'évolution dans le temps du taux d'épargne personnelle (`psavert`) et de la durée médiane en semaines du chômage (`uempmed`). Stocker le graphique dans un objet que l'on appellera `p_2`;

```
p_2 <- ggplot(data = df,
              aes(x = date, y = value)) +
  geom_line(aes(group = key))
p_2
```

- Modifier le code ayant servi à construire le graphique `p_2` pour que le type de ligne soit différent pour chacune des deux séries représentées. Les deux lignes doivent être tracées en bleu. Stocker le graphique dans un objet que l'on appellera `p_3`;

```
p_3 <- ggplot(data = df,
              aes(x = date, y = value)) +
  geom_line(aes(linetype = key), colour = "blue")
p_3
```

5. À présent, modifier le code ayant servi à construire `p_3` pour qu'à la fois la couleur et le type de ligne servent à différencier les deux séries. Stocker le graphique dans un objet que l'on appellera `p_4`;

```
p_4 <- ggplot(data = df,
              aes(x = date, y = value)) +
  geom_line(aes(linetype = key, colour = key))
p_4
```

6. Modifier le graphique `p_4` en ajoutant une couche d'échelle de couleur pour que le taux d'épargne personnelle (`psavert`) soit représenté en `dodger blue`, et que la durée de chômage (`uempmed`) soit représentée en rouge. De plus, retirer le titre de la légende;

```
p_4 <-
p_4 +
  scale_colour_manual("", values = c("psavert" = "dodger blue",
                                     "uempmed" = "red"))
p_4
```

7. Modifier le graphique `p_4` en ajoutant une couche d'échelle de type de ligne pour que le taux d'épargne personnelle (`psavert`) soit représenté par des tirets, et que la durée de chômage (`uempmed`) soit représentée par une ligne pleine. De plus, retirer le titre de la légende;

```
p_4 <-
p_4 +
  scale_linetype_manual("", values = c("psavert" = "dashed",
                                       "uempmed" = "solid"))
p_4
```

8. Créer le tableau de données `df_2`, une copie de `df`, dans lequel la variable `key` doit être un facteur dont les niveaux sont `uempmed` et `psavert`;

```
df_2 <- df %>%
  mutate(key = factor(key, levels = c("uempmed", "psavert")))
```

9. Créer le vecteur `etiq` suivant :

```
etiq <- c("psavert" = "Pers. Saving Rate",
          "uempmed" = "Median Duration of Unemployment (weeks)")
```

Ce vecteur contient des valeurs d'étiquettes pour la légende du graphique qu'il va falloir créer. Réaliser un graphique à l'aide des données contenues dans le tableau `df_2`, représentant l'évolution dans le temps du taux d'épargne personnelle ainsi que l'évolution de la durée médiane du chômage

en semaines. La courbe taux d'épargne personnelle doit être composée de tirets et être de couleur `dodger blue`; la courbe de la durée médiane du taux de chômage doit être une ligne rouge. La légende ne doit pas comporter de titre, et ses étiquettes doivent être modifiées pour que "Pers. Saving Rate" s'affiche à la place de "psavert", et pour que "Median Duration of Unemployment (weeks)" s'affiche à la place de "uempmed". Stocker le graphique dans un objet que l'on appellera `p_5`;

Note : il s'agit de reprendre le code ayant servi à créer le graphique `p_4`, en modifiant légèrement les échelles de couleur et de ligne pour prendre en compte les étiquettes proposées dans le vecteur `etiq`.

```
p_5 <-
  ggplot(data = df_2,
         aes(x = date, y = value)) +
  geom_line(aes(linetype = key, colour = key)) +
  scale_colour_manual("", values = c("psavert" = "dodger blue", "uempmed" = "red"),
                    labels = etiq) +
  scale_linetype_manual("", values = c("psavert" = "dashed", "uempmed" = "solid"),
                       labels = etiq)
p_5
```

10. Modifier `p_5` pour lui ajouter une couche permettant de déplacer la légende en bas du graphique (utiliser la fonction `theme()`);

```
p_5 <-
  p_5 +
  theme(legend.position = "bottom", text = element_text(size = 20))
p_5
```

11. Ajouter une couche au graphique `p_5` qui permet de définir un thème. Utiliser le thème minimal (`theme_minimal()`). Que se passe-t-il pour la légende? Repositionner la légende en dessous, et retirer les titres des axes;

Lorsqu'on utilise une fonction définissant un thème, toute modification effectuée par la fonction `theme()` doit être faite après, pour venir écraser les informations pré-définies par le thème utilisé.

```
p_5 <-
  p_5 + theme_minimal() +
  theme(legend.position = "bottom", text = element_text(size = 20)) +
  xlab("") + ylab("")
p_5
```

12. Sauvegarder le graphique `p_5` au format PDF en précisant une largeur de 12 et une hauteur de 8.

```
ggsave(filename = "graph_test.pdf", p_5, width = 12, height = 8)
```

Exercice 3 (Différentes représentations graphiques)

1. Charger le jeu de données `mpg` contenu dans le *package* `ggplot2` en mémoire, puis consulter la page d'aide du jeu de données pour en prendre connaissance;

```
data(mpg)
?mpg
```

2. Représenter à l'aide d'un nuage de points la relation entre la consommation sur autoroute des véhicules de l'échantillon (`hwy`) et la cylindrée de leur moteur (`displ`)

```
ggplot(data = mpg, aes(x = hwy, y = displ)) + geom_point()
```

3. Reprendre le code du graphique précédent et modifier la forme des points pour les changer en symbole +; modifier la couleur des "plus" de manière à la faire dépendre du nombre de cylindres (`cyl`);

```
ggplot(data = mpg, aes(x = hwy, y = displ)) +
  geom_point(aes(colour = factor(cyl)), shape = 3)
```

4. À présent, représenter la distribution de la consommation sur autoroute des véhicules (`hwy`) pour chacune des valeurs possibles du nombre de cylindres (`cyl`) à l'aide de boîtes à moustaches;

```
ggplot(data = mpg) + geom_boxplot(aes(x = factor(cyl), y = hwy))
```

5. Charger le jeu de données `economics` contenu dans le *package* `ggplot2` en mémoire, puis consulter la page d'aide du jeu de données pour en prendre connaissance. Ensuite, ajouter au tableau les variables `u_rate` et `e_rate`, respectivement le taux de chômage et le taux d'emploi;

```
data(economics)
economics <- economics %>%
  mutate(u_rate = unemploy / pop,
         e_rate = 1- u_rate)
economics
```

6. Représenter à l'aide de barres l'évolution dans le temps du taux de chômage, et remplir les barres avec la couleur rouge;

```
ggplot(data = economics, aes(x = date, y = u_rate)) +
  geom_bar(stat = "identity", fill = "red")
```

7. Reprendre le code du graphique précédent et ajouter une couche permettant de modifier les limites de l'axe des abscisses pour afficher les valeurs uniquement sur la période "2012-01-01" à "2015-01-01" (utiliser la fonction `coord_cartesian()`). Stocker le graphique dans un objet que l'on appellera `p`;

```
p <- ggplot(data = economics, aes(x = date, y = u_rate)) +
  geom_bar(stat = "identity", fill = "red") +
  coord_cartesian(xlim = c(as.Date("2012-01-01"),
                          as.Date("2015-01-01")))
```

8. Dans le tableau de données `economics`, sélectionner les variables `date`, `u_rate` et `e_rate`, puis utiliser la fonction `gather()` pour obtenir un tableau dans lequel chaque ligne correspond à la valeur d'une des variables (taux de chômage ou taux d'emploi) à une date donnée. Stocker le résultat dans un objet que l'on appellera `df_3`;

```
df_3 <-
  economics %>%
  select(date, u_rate, e_rate) %>%
  gather(key, value, -date)
```

9. Utiliser le tableau de données `df_3` pour représenter graphiquement à l'aide de barres les taux de chômage et taux d'emploi par mois sur la période "2012-01-01" à "2015-01-01". Sur le graphique, les barres représentant le taux de chômage et celles représentant le taux d'emploi devront être superposées.

Note : il s'agit de modifier légèrement le code ayant permis de réaliser le graphique p.

```
ggplot(data = economics %>%
  select(date, u_rate, e_rate) %>%
  gather(key, value, -date),
  aes(x = date, y = value)) +
  geom_bar(stat = "identity", aes(fill = key)) +
  coord_cartesian(xlim = c(as.Date("2012-01-01"),
    as.Date("2015-01-01")))
```

Exercice 4 (Facettes)

1. Charger le *package* `WDI` (l'installer si nécessaire), puis en utilisant la fonction `WDI()`, récupérer les données de PIB par tête (`NY.GDP.PCAP.PP.KD`, PPP, constant 2005 international \$) et de taux de chômage (`SL.UEM.TOTL.ZS`, total, % of total labor force) pour la France, l'Allemagne et le Royaume Uni, pour la période allant de 1990 à 2015. Ces données doivent être stockées dans un tableau que l'on appellera `df`;

```
library(WDI)
# NY.GDP.PCAP.PP.KD (GDP per capita, PPP (constant 2005 international $))
# SL.UEM.TOTL.ZS (Unemployment, total (% of total labor force))
df <- WDI(country = c("FR", "GB", "DE"),
  indicator = c("NY.GDP.PCAP.PP.KD", "SL.UEM.TOTL.ZS"),
  start = 1990, end = 2015)
head(df)
```

2. Transformer le tableau `df` afin que chaque ligne indique : l'état (`country`), l'année (`year`), le nom de la variable (`variable`) et la valeur (`value`). La colonne `variable` doit être de type `factor`, et les étiquettes des variables `NY.GDP.PCAP.PP.KD` et `SL.UEM.TOTL.ZS` doivent être `GDP` et `Unemployment` respectivement;

```
df <-
df %>%
select(-iso2c) %>%
gather(variable, valeur, NY.GDP.PCAP.PP.KD, SL.UEM.TOTL.ZS) %>%
mutate(variable = factor(variable,
levels = c("NY.GDP.PCAP.PP.KD", "SL.UEM.TOTL.ZS"),
labels = c("GDP", "Unemployment")))
```

3. Représenter graphiquement l'évolution du PIB et du taux de chômage pour les trois pays. Utiliser la fonction `facet_wrap()` afin de regrouper les variables par type de variable : les observations des valeurs du PIB d'un côté, et celles du taux de chômage de l'autre. Utiliser une représentation en ligne, en faisant dépendre la couleur du pays ;

```
ggplot(data = df, aes(x = year, y = valeur, colour = country)) +
geom_line() +
facet_wrap(~variable) +
ylab("") + xlab("")
```

4. Reprendre le code du graphique précédent en le modifiant légèrement afin de libérer les axes des ordonnées ;

```
ggplot(data = df, aes(x = year, y = valeur, colour = country)) +
geom_line() +
facet_wrap(~variable, scales = "free_y") +
ylab("") + xlab("")
```

5. Modifier les paramètres esthétiques du graphique afin de faire dépendre le type de ligne des pays de la manière suivante : des points pour la France, des tirets pour l'Allemagne, des tirets longs pour le Royaume Uni. Définir l'épaisseur des lignes à 1.5 ;

```
ggplot(data = df, aes(x = year, y = valeur, colour = country)) +
geom_line(aes(linetype = country), size = 1.5) +
scale_linetype_manual("country",
values = c("France" = "dotted",
"Germany" = "dashed",
"United Kingdom" = "longdash")) +
facet_wrap(~variable, scales = "free_y") +
ylab("") + xlab("")
```

6. Modifier légèrement le code ayant permis de réaliser le graphique de la question précédente pour que la direction ne soit non plus horizontale (par défaut), mais verticale ;

```
ggplot(data = df, aes(x = year, y = valeur, colour = country)) +
  geom_line(aes(linetype = country), size = 1.5) +
  scale_linetype_manual("country",
                        values = c("France" = "dotted",
                                   "Germany" = "dashed",
                                   "United Kingdom" = "longdash")) +
  facet_wrap(~variable, scales = "free_y", dir = "v") +
  ylab("") + xlab("")
```

7. En utilisant la fonction `facet_wrap()`, représenter chaque série dans un des panels, de sorte qu'un panneau représente l'évolution d'une série pour un pays donné ;

```
ggplot(data = df, aes(x = year, y = valeur)) +
  geom_line() +
  facet_wrap(country~variable, scales = "free_y") +
  ylab("") + xlab("")
```

8. À présent, utiliser la fonction `facet_grid()` pour créer une grille de graphiques dans laquelle les lignes correspondent aux pays et les colonnes aux variables. Prendre soin de libérer les échelles ;

```
ggplot(data = df, aes(x = year, y = valeur)) +
  geom_line() +
  facet_grid(country~variable, scales = "free") +
  ylab("") + xlab("")
```

9. Reprendre la question précédente en faisant cette fois une grille dans laquelle les lignes correspondent aux variables et les colonnes aux pays ;

```
ggplot(data = df, aes(x = year, y = valeur)) +
  geom_line() +
  facet_grid(variable~country, scales = "free") +
  ylab("") + xlab("")
```

Exercice 5 (Annotations)

1. En utilisant la fonction `WDI()` du *package* `WDI`, récupérer les séries de 2010 du PIB par tête (`NY.GDP.PCAP.PP.KD`, PPP, constant 2005 international \$) et de l'espérance de vie à la naissance (`SP.DYN.LE00.IN`, total, years) pour tous les pays. Les données seront stockées dans un tableau que l'on nommera `df` ;

```
# NY.GDP.PCAP.PP.KD (GDP per capita, PPP (constant 2005 international $))
# SP.DYN.LE00.IN (Life expectancy at birth, total (years))
df <- WDI(country = c("all"),
          indicator = c("NY.GDP.PCAP.PP.KD", "SP.DYN.LE00.IN"),
          start = 2010, end = 2010)
```

2. Représenter par un nuage de points l'espérance de vie à la naissance en fonction du PIB par tête. Retirer les titres des axes, et ajouter le titre suivant (sur deux lignes) : Life Expectancy at birth (years) VS. Real GDP per Capita (PPP, contant 2005 international \$) ;


```
library(stringr)
ggplot(data = df, aes(x = NY.GDP.PCAP.PP.KD, y = SP.DYN.LE00.IN)) +
  geom_point() +
  ggtitle(str_c("Life Expectancy at birth (years) VS.",
"Real GDP per Capita (PPP, contant 2005 international $)", sep = "\n")) +
  xlab("") + ylab("")
```

Note : ici, on utilise la fonction `str_c()` uniquement parce que la chaîne est trop longue pour tenir dans le document de correction. Il est tout fait possible de fournir une chaîne du type `xxx\nxxx`.

- Ajouter une courbe de tendance obtenue par lissage Loess (en utilisant une fonction du *package* `ggplot2`);

```
ggplot(data = df, aes(x = NY.GDP.PCAP.PP.KD, y = SP.DYN.LE00.IN)) +
  geom_point() +
  ggtitle(str_c("Life Expectancy at birth (years) VS.",
"Real GDP per Capita (PPP, contant 2005 international $)", sep = "\n")) +
  xlab("") + ylab("") +
  geom_smooth()
```

- Modifier le tableau `df` pour lui ajouter la variable `drapeau` qui prendra la valeur `TRUE` si l'état de l'observation est soit France, soit Luxembourg, et `FAUX` sinon;

```
df <-
df %>%
mutate(drapeau = ifelse(country %in% c("France", "Luxembourg"), yes = TRUE, no = FALSE))
```

- Modifier légèrement le code ayant permis de réaliser le précédent graphique, pour faire dépendre la couleur des points de la variable `drapeau` (rouge pour `TRUE` et noir pour `FALSE`);

```
ggplot(data = df, aes(x = NY.GDP.PCAP.PP.KD, y = SP.DYN.LE00.IN)) +
  ggtitle(str_c("Life Expectancy at birth (years) VS.",
"Real GDP per Capita (PPP, contant 2005 international $)", sep = "\n")) +
  xlab("") + ylab("") +
  scale_colour_manual("", values = c("TRUE" = "red", "FALSE" = "Black")) +
  guides(colour = FALSE) +
  geom_smooth() +
  geom_point(aes(colour = drapeau))
```

- Créer un tableau de données que l'on appellera `df_fleche` qui contient les observations du tableau `df` pour la France et le Luxembourg uniquement;

```
df_fleches <- df %>%
  filter(country %in% c("France", "Luxembourg"))
```

- Reprendre le code du graphique précédent pour le modifier de façon à ajouter deux flèches : l'une montrant les points de la France et l'autre du Luxembourg. Le nom du pays devra être inscrit

à l'origine des deux flèches (utiliser la fonction `annotate()` et se servir du tableau de données `df_fleche`);

```
library(grid)
ggplot(data = df, aes(x = NY.GDP.PCAP.PP.KD, y = SP.DYN.LE00.IN)) +
  ggtitle(str_c("Life Expectancy at birth (years) VS.",
"Real GDP per Capita (PPP, contant 2005 international $)", sep = "\n")) +
  xlab("") + ylab("") +
  scale_colour_manual("", values = c("TRUE" = "red", "FALSE" = "Black")) +
  guides(colour = FALSE) +
  geom_smooth() +
  geom_point(aes(colour = drapeau)) +
  annotate(geom = "segment",
  x = df_fleches$NY.GDP.PCAP.PP.KD + 1e4,
  xend = df_fleches$NY.GDP.PCAP.PP.KD,
  y = df_fleches$SP.DYN.LE00 - 10,
  yend = df_fleches$SP.DYN.LE00,
  arrow = arrow(length = unit(0.5, "cm")), colour = "red") +
  annotate(geom = "text",
  x = df_fleches$NY.GDP.PCAP.PP.KD + 1e4,
  y = df_fleches$SP.DYN.LE00 - 11,
  label = df_fleches$country, colour = "red")
```

Exercice 6 (Annotations)

1. À l'aide de la fonction `WDI` du *package* `WDI`, récupérer la série de l'inflation (`FP.CPI.TOTL.ZG`, consumer prices, annual %) en France sur la période 1960–2015, et stocker ces données dans un tableau que l'on appellera `inflation_fr`;

```
# FP.CPI.TOTL.ZG (Inflation, consumer prices (annual %))
inflation_fr <- WDI(country = "FR", indicator = "FP.CPI.TOTL.ZG",
  start = 1960, end = 2015)
```

2. Représenter par une ligne l'évolution du taux annuel d'inflation en France sur la période 1960–2015. Retirer les titres d'axes et ajouter le titre suivant : `Inflation in France (annual %)`;

```
p <- ggplot(data = inflation_fr, aes(x = year, y = FP.CPI.TOTL.ZG)) +
  geom_line() + xlab("") + ylab("") +
  ggtitle("Inflation in France (annual %)")
```

3. Soient les deux vecteurs suivants, qui définissent les dates de début et fin de la période inflationniste et celle de grande modération :

```
per_1 = c(1970, 1976) # Periode inflationniste
per_2 = c(1976, 1991) # Periode de grande moderation
```

Reprendre le code du graphique précédent et ajouter deux rectangles : un premier, rouge, pour mettre en valeur la période inflationniste et un second, bleu, mettant en valeur la période de grande modération. Modifier le paramètre de transparence pour le fixer à 0.2;

```
p <-
p +
  annotate(geom = "rect", xmin = per_2[1], xmax = per_2[2],
          ymin = -Inf, ymax = Inf,
          alpha = .2, fill = "blue") +
  annotate(geom = "rect", xmin = per_1[1], xmax = per_1[2],
          ymin = -Inf, ymax = Inf,
          alpha = .2, fill = "red")
```

- Ajouter au graphique précédent des lignes grises verticales en tirets pour les années 1973, 1979 et 2008. Modifier également les *breaks* de l'échelle des abscisses à l'aide de la fonction `pretty_breaks()` du *package* `scales`;

```
library(scales)
p +
  geom_vline(xintercept = c(1973, 1979, 2008),
            col = "grey40", linetype = "dashed", size = 0.9) +
  scale_x_continuous(breaks = pretty_breaks(n=10))
```

Exercice 7 (Carte simple)

- À l'aide de la fonction `WDI` du *package* `WDI`, récupérer la série du PIB par tête (`NY.GDP.PCAP.PP.KD`, PPP, constant 2005 international \$) pour tous les pays disponibles pour l'année 2010, et stocker ces données dans un tableau que l'on appellera `gdp_capita`;

```
# NY.GDP.PCAP.PP.KD (GDP per capita, PPP (constant 2005 international $))
gdp_capita <-
  WDI(country = c("all"), indicator = c("NY.GDP.PCAP.PP.KD"),
      start = 2010, end = 2010)
```

- Dans le tableau `gdp_capita`, modifier la valeur de la variable `country` pour l'observation de la Slovaquie, pour qu'elle vaille `Slovakia` au lieu de `Slovak Republic`;

```
gdp_capita <-
gdp_capita %>%
  mutate(country = ifelse(country == "Slovak Republic",
                          yes = "Slovakia", no = country))
```

- Filtrer les observations du tableau `gdp_capita` pour ne conserver que les observations des pays membres de l'Union Européenne dont les noms sont contenus dans le vecteur `membres_ue`. Stocker le résultat dans un tableau que l'on nommera `gdp_capita_eu`;

```
membres_ue <-c("Austria", "Belgium", "Bulgaria", "Cyprus", "Croatia",
              "Czech Republic", "Denmark", "Estonia", "Finland", "France",
              "Germany", "Greece", "Hungary", "Ireland", "Italy", "Latvia",
              "Lithuania", "Luxembourg", "Malta", "Netherlands", "Poland",
              "Portugal", "Romania", "Slovakia", "Slovenia", "Spain", "Sweden",
              "United Kingdom")
```

```
gdp_capita_eu <-
  gdp_capita %>%
  filter(country %in% membres_ue)
```

4. Utiliser le *package* `rworldmap` pour récupérer les données nécessaires à la réalisation d'une carte du monde ;

```
library(rworldmap)
world_map <- getMap()
world_df <- fortify(world_map)
```

5. Afficher une carte du monde à l'aide des fonctions contenues dans le *package* `ggplot2` ;

```
ggplot(data = world_df, aes(x = long, y = lat, group = group)) +
  geom_polygon() +
  coord_quickmap()
```

6. Modifier les échelles des axes pour faire figurer les méridiens de -60 à 60 par pas de 30 et les parallèles de -180 à 180 par pas de 45 . Modifier également la projection cartographique pour choisir la projection orthographique, à l'aide de la fonction `coord_map()` ;

```
ggplot(data = world_df, aes(x = long, y = lat, group = group)) +
  geom_polygon() +
  scale_y_continuous(breaks = (-2:2) * 30) +
  scale_x_continuous(breaks = (-4:4) * 45) +
  coord_map("orthographic", orientation = c(30, 20, 0))
```

7. Joindre les informations contenues dans le tableau `gdp_capita_eu` au tableau contenant les données permettant la réalisation des cartes ;

```
world_df_eu <-
  world_df %>%
  left_join(gdp_capita_eu, by = c("id" = "country"))
```

8. Réaliser une carte choroplèthe reflétant pour chaque pays membre de l'Union Européenne la valeur du PIB par tête de 2012 ;

```
ggplot(data = world_df_eu,
  aes(x = long, y = lat, group = group, fill = NY.GDP.PCAP.PP.KD)) +
  geom_polygon() +
  scale_y_continuous(breaks = (-2:2) * 30) +
  scale_x_continuous(breaks = (-4:4) * 45) +
  coord_quickmap(xlim = c(-15, 40), ylim = c(35, 70))
```

9. Modifier les couleurs de l'échelle continue de la carte précédente, pour que les faibles valeurs du PIB par tête soient représentées en jaune, et les valeurs les plus hautes en rouge ;

```
ggplot(data = world_df_eu,  
  aes(x = long, y = lat, group = group, fill = NY.GDP.PCAP.PP.KD)) +  
  geom_polygon() +  
  coord_quickmap(xlim = c(-15, 40), ylim = c(35, 70)) +  
  scale_fill_gradient(low = "yellow", high = "red")
```

10. Modifier les ruptures de l'échelle de couleur pour qu'elles aillent de 10000 à 100000 ; modifier également l'étiquette de ces ruptures de sorte que 35000 soit affiché comme 35k, 60000 comme 60k, etc. Enfin, ajouter un titre au graphique et retirer les titres d'axes.

```
range(world_df_eu$NY.GDP.PCAP.PP.KD, na.rm=TRUE)  
library(stringr)  
ggplot(data = world_df_eu,  
  aes(x = long, y = lat, group = group, fill = NY.GDP.PCAP.PP.KD)) +  
  geom_polygon() +  
  coord_quickmap(xlim = c(-15, 40), ylim = c(35, 70)) +  
  scale_fill_gradient("",  
    low = "yellow", high = "red",  
    breaks = seq(1e4, 1e5, by = 25e3),  
    labels = str_c(seq(1e4, 1e5, by = 25e3) %/% 1e3, "k")) +  
  ggtitle("2012 GDP per capita, PPP (constant 2005 international $)") +  
  xlab("") + ylab("")
```