Data Visualisation: Assignment 4

Name: Marcel Aguilar Garcia Student Id: 20235620 Class: 2021-CT5136

Part 1: Visualisation decisions

Introduction

This visualisation shows the reading performance of 15-year old students of OECD countries in 2018. The countries are represented in axis x while the average reading performance of the students is represented in axis y. Students variable has been split into two categories, boys and girls, which are represented by two different shapes. It is important to note that Ireland is the only country highlighted in red which is probably relevant to the context of the article. Finally, the average of all countries has been added and helps to give a better understanding of which countries are below or above this average.

The axes

Axis x: Axis x represents the member countries from OECD. Therefore x is a nominal categorical variable (there is no intrinsic ordering for these countries). However, axis x has been sorted ascendingly by boys performance. An adjustment has been done to set country labels to 45° as they take too much horizontal space. The visualisation is probably focus on Ireland which has been highlighted in red. In a similar way, the label representing the average is in bold to differentiate it from country labels. Finally, and aligned with the ink-ratio theory the title, line and ticks for axis x have been removed.

Axis y: Axis y represents the average reading performance score per country and therefore is associated to a continuous variable. The author has decided to limit axis y to the range 340 to 560 as there are no countries with scores outside these limits. This range has been broken in intervals of length 20 which are used as labels for axis y. Again, the title, line and ticks for axis y have been removed. A subtle difference between both axes is that, axis y is inside the background while axis x is placed outside.

The background and gridlines

The colour of the background is light blue. Horizontal white gridlines are used to identify dots with reading performance (axis y). In a similar way, vertical white gridlines are used to identify dots with countries (axis x). Vertical gridlines stop in the dots representing the first category and a grey line is then used to connect it to the other one. This grey line helps to visualise the difference of girls and boys averages and compare it between countries.

The legend

The legend has been positioned in the left bottom of the visualisation. It simple but informative. It has no background or title and shows the dot shape for each of the categories, boys and girls.

Colour

As seen with the background and dots, the main colour of this visualisation is blue. However, the background has a very light blue and does not affect the visibility of the dots. Red and black has been used to highlight specific labels and dots (Ireland and Average).

Part 2: Visualisation reproduction

Explanation

Aesthetics

The aesthetics used for both categories were set using *geom_points*, *scale_shape_manual*, *scale_colour_manual*, and *scale_fill_manual*. The shape was set to be 21 (boys) and 23 (girls), the colour and fill were specified to be the colours matching the original plot (see more in colour section). The colour for shape 23 was set to be the same from the background so the grey gridlines would not be visible inside the shape. The dots for OECD-Average and Ireland were overridden by using *geom_point* again and restricting the data to each of the points setting colour and fill to black (average) and red (Ireland).

Axes

The axes were automatically defined when assigning variable x and y in the plot creation using *ggplot*. The scales of both axes were modified to look as the original plot by using *scale_x_discrete* and *scale_y_continuous*. The labels of axis y were overridden to slightly different numbers as, in this design, the breaks do not match the exact position where they are. Final details were changed in the *theme* such as the angle of the labels and highlighting specific labels (*axis.text*), removing ticks (*axis.ticks*), line (*axis.line*) and title (*axis.title*).

Panel

panel.background was used to set background colour to #dee8ef. The background has a blue line on the top that was set using *annotate* with *geom* set to segment.

Gridlines

Axis x gridlines were removed in the *theme* by setting *panel.grid.major.x* and *panel.grid.minor.x* to *element_blank()*. The desired behaviour was achieved using *geom_segment* twice to create the grey line and white line and setting the limits manually to fit the dots. The gridline is defined before *geom_points* as otherwise the line would be above the filling (passing inside dot shape 23).

Legend

Most of legend changes were done in the *theme*. To have the legend on the left bottom corner *legend.position* was set to 'bottom' and *legend.justification* to 'left'. In order to have both categories in the same line I set *legend.direction* to be 'horizontal'. The background and key colours of the legend was set to be white. The title was hidden. Finally, by using *guides* I overrode the colour of the shape assigned to 'Girls' to be white.

Colours

The main colours were extracted using `https://labs.tineye.com/color/`. The background colour was set to be #dee8ef and the colour of the shapes to #608098. Colours used to highlight were set to their names ("red" and "black").

Title

The title was defined using *ggtitle* and the font, size and face were defined in the theme using *plot.title*. However, I had to use *annotations* to add the other two texts. While I had access to most of available fonts using *extrafont* library, I was unable to match the exact font used in the original plot.

Margins

I did not have to modify the margins but I struggled trying to find a way to expand the background to axis y. The desired behaviour was finally achieved by using *ggplotGrob* and modifying the *grob* corresponding to y-axis labels.

Plot

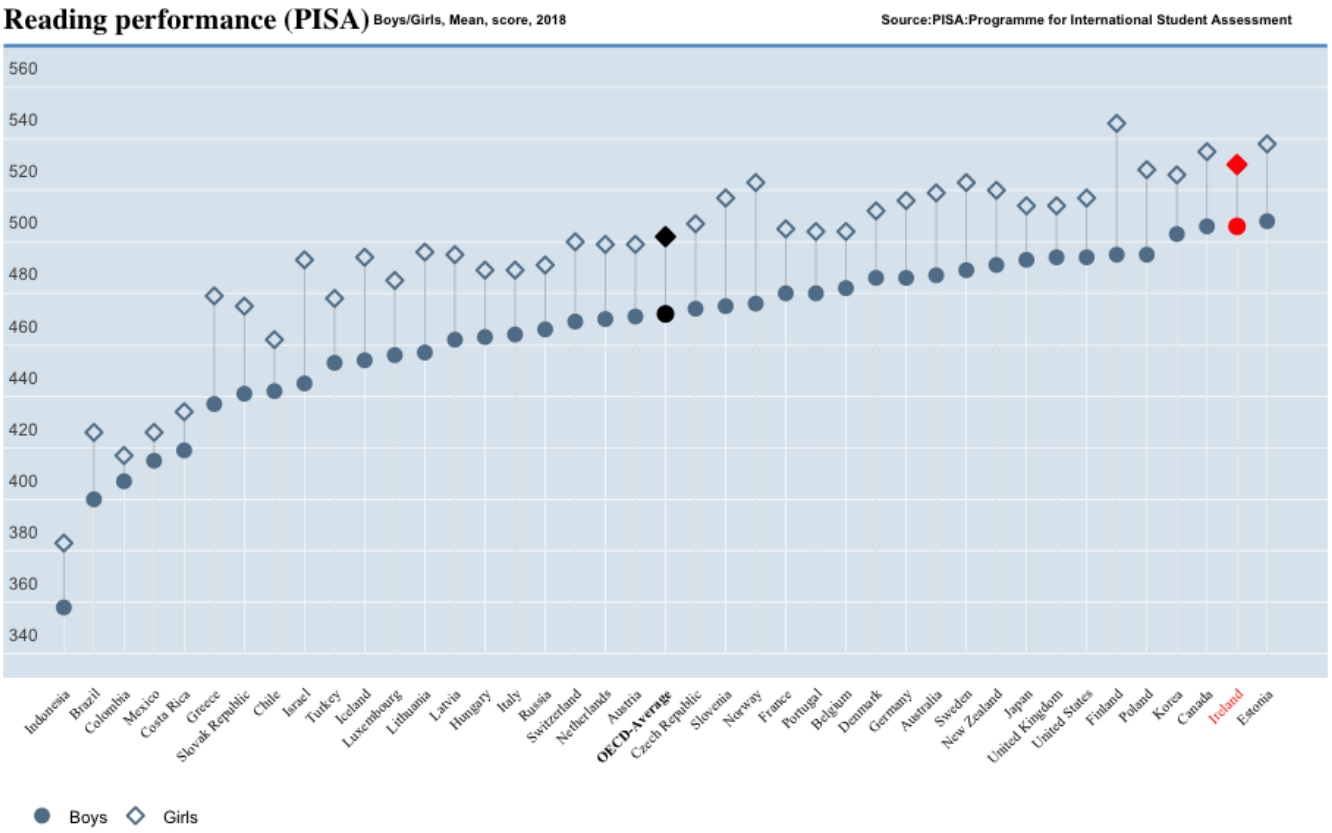

Figure 1: Reproduction of Reading Performance (PISA) visualisation

Code

```
#Loading all required R libraries

library(readxl)
library(tidyverse)
library(ggplot2)
library(dplyr)
library(grid)
library(gtable)
library(extrafont)

loadfonts(device="win")

#I start by loading the data

data <- read.csv('OECD_PISA.csv',fileEncoding="UTF-8-BOM") %>%

  #There are just four columns that are required to create this plot
  #which I have renamed to a more appropiate name: Country, Gender, Year and AverageScore

  select('LOCATION','SUBJECT','TIME','Value') %>%

  dplyr::rename(Country = 'LOCATION',Gender = 'SUBJECT',Year = 'TIME',AverageScore = 'Value') %>%

  #This plot is using the data from 2018. In this step I filter all rows from 2018.
  #In a similar way, I am just interested with the average of Girls and Boys and therefore,
  #I am filtering out all data related to totals

  filter(Year==2018 & Gender!='TOT') %>%

  #The country names are using a short name and I have updated them with their real name. When I
      read that country code package in R can do that was too late and had all the names typed. I
      decided to use the recoding as seen below:

  mutate(Country=recode(Country,"IDN" = "Indonesia", "BRA" = "Brazil","COL" ="Colombia",
                        "MEX" ="Mexico","CRI" ="Costa Rica","GRC" ="Greece","SVK" ="Slovak Republic",
                        "CHL" ="Chile","ISR" ="Israel","TUR" ="Turkey","ISL" ="Iceland",
                        "LUX" ="Luxembourg","LTU" ="Lithuania","LVA" ="Latvia","HUN" ="Hungary",
                        "ITA" ="Italy","RUS" ="Russia","CHE" ="Switzerland","NLD" ="Netherlands",
                        "AUT" ="Austria","OAVG" ="OECD-Average","CZE" ="Czech Republic","SVN"
                            ="Slovenia",
                        "NOR" ="Norway","FRA" ="France","PRT" ="Portugal","BEL" ="Belgium","DEU"
                            ="Denmark",
                        "DNK" ="Germany","AUS" ="Australia","SWE" ="Sweden","NZL" ="New
                            Zealand","JPN" ="Japan",
                        "GBR" ="United Kingdom","USA" ="United States","FIN" ="Finland","POL"
                            ="Poland",
                        "KOR" ="Korea","CAN" ="Canada","IRL" ="Ireland","EST" ="Estonia"))%>%

  #Finally, I update the categories Boy and Girl to be the plural (as seen in the original plot)

  mutate(Gender = recode(Gender,"BOY" = "Boys","GIRL" = "Girls"))

#The data is ordered by Gender, Average Score and finally, Country (alphabetically). I used
#factors to make sure that the this order will be used by ggplot.

data <- data %>% arrange(Gender,desc(AverageScore),desc(Country))
```

```
data$Country <- factor(data$Country,levels = rev(unique(data$Country)))
#I am defining the texts that will be used next to the title in here

Text1 = textGrob("Boys/Girls, Mean, score, 2018", gp = gpar(col = "black", fontsize = 7,fontface =
    "bold"))
Text2 = textGrob("Source:PISA:Programme for International Student Assessment", gp = gpar(col =
    "black", fontsize = 7,fontface = "bold"))

#I finally use ggplot to initialise the plot, variable x is set to be the Country and variable y
#is set to be the AverageScore. The shape, colour and fill will will be different
#depending on the Gender variable (Boys and Girls)

g <- ggplot(data, aes(x=Country, y=AverageScore,shape=Gender,colour=Gender,fill=Gender))

#I start by defining the vertical gridlines. The reason for that is that later on
#I will want the filling of one of the shapes to override the line so, the line
#will not be seen inside the shape. I have used different paramaters to make sure
#they look as closed as possible to the original plot.

g <- g + geom_segment(data = data %>% filter(Gender == "Girls"),aes(xend = Country), yend = 340,
    colour="grey", size=0.3)+
  geom_segment(data = data %>% filter(Gender == "Boys"), aes(xend = Country), yend = 340,
      colour="white", size=0.3)

#Finally, I add all points into the plot and define the shapes
#colours and fillings per category (boys/girls).

g <- g + geom_point(size=2.5, stroke = 1)+
  scale_shape_manual(values = c(21, 23))+
  scale_colour_manual(values = c( "#608098","#608098"))+
  scale_fill_manual(values = c("#608098","#dee8ef"))

#Dots that are highlighted (average and Ireland) are defined in here with the
#required colours (black and red respectively).

g <- g + geom_point(data = data %>% filter(Country == "Ireland" & Gender == "Boys"),
                    aes(x=Country, y=AverageScore,shape=Gender,colour=Gender,fill=Gender),
                    colour="red", size=2.5,shape=21,stroke=1,fill="red")+
  geom_point(data = data %>% filter(Country == "Ireland" & Gender == "Girls"),
             aes(x=Country, y=AverageScore,shape=Gender,colour=Gender,fill=Gender),
             colour="red", size=2.5,shape=23,stroke=1,fill="red")+
  geom_point(data = data %>% filter(Country == "OECD-Average" & Gender == "Boys"),
             aes(x=Country, y=AverageScore,shape=Gender,colour=Gender,fill=Gender),
             colour="black", size=2.5,shape=21,stroke=1,fill="black")+
  geom_point(data = data %>% filter(Country == "OECD-Average" & Gender == "Girls"),
             aes(x=Country, y=AverageScore,shape=Gender,colour=Gender,fill=Gender),
             colour="black", size=2.5,shape=23,stroke=1,fill="black")

#The scale/limits of axis x and y are a bit different from the expected one.
#I have done some changes using expand and limits. Additionally,
#The labels of axis y are overridened with different numbers (as seen in original plot).

g <-  g + scale_x_discrete(expand = c(0.05,0))+
  scale_y_continuous(limits = c(350,570),breaks = seq(350, 570, len = 12),
                     labels=seq(340, 560, len = 12))

#Title and annotations (title subtitles and line on the top) are added in here.
#I need to change the limits of the plot and use clip=off in order to be able
#to go out of these limits.
```

```
g <- g + coord_cartesian(ylim = c(342,565),clip = "off")+
  annotation_custom(grob = Text1, xmin = "Luxembourg", xmax = "Russia", ymin = 583, ymax = 590)+
  annotation_custom(grob = Text2, xmin = "Australia", xmax = "Ireland", ymin = 583, ymax = 590)+
  annotate(geom = 'segment',y = Inf,yend = Inf,x = -Inf,xend = Inf,colour = "steelblue3",size =
      0.8)+
  ggtitle("Reading performance (PISA)")

#Finally, all information related to axis, legend, and other remaining details are modified in the
    theme

g<- g + theme(panel.background = element_rect(fill = "#dee8ef"),
              axis.line.x = element_blank(),
              axis.ticks.x = element_blank(),
              axis.title.x = element_blank(),
              axis.text.x =
                element_text(angle = 45, vjust = 1, hjust = 1,size = 7.5,family = "Times New Roman",
                             face  = ifelse(levels(data$Country) == "OECD-Average",'bold','plain'),
                             colour = ifelse(levels(data$Country) == "Ireland","red","black")),
              axis.line.y = element_blank(),
              axis.ticks.y = element_blank(),
              axis.title.y = element_blank(),
              panel.grid.major.x = element_blank(),
              panel.grid.minor.x = element_blank(),
              panel.grid.major.y = element_blank(),
              legend.justification="left",
              legend.position = "bottom",
              legend.direction="horizontal",
              legend.background = element_blank(),
              legend.title = element_blank(),
              legend.key = element_rect(fill = "white"),
              plot.title = element_text(face="bold",family = "serif",size=15),
              axis.text.y = element_text(hjust = 0,vjust=1))+
  guides(colour = guide_legend(override.aes=list(fill=c("#608098","white"))))

#In order to expand the background over axis y I had to use the trick below I was able to find this
#information in
#https://stackoverflow.com/questions/46626326/ggplot2-y-axis-labels-left-justified-inside-plot-area
#The main idea is focus on updating the ggplot to a ggplotGrob and changing the label grob related
#to y axis

gp <- ggplotGrob(g)
y.label.grob <- gp$grobs[[which(gp$layout$name == "axis-l")]]$children$axis
gp$grobs[[which(gp$layout$name == "axis-l")]] <- zeroGrob()
gp$widths[gp$layout$l[which(gp$layout$name == "axis-l")]] <- unit(0, "cm")
new.y.label.grob <- gtable(heights = unit(1, "npc"))
new.y.label.grob <- gtable_add_cols(new.y.label.grob,widths = y.label.grob[["widths"]][2])
new.y.label.grob <- gtable_add_grob(new.y.label.grob,y.label.grob[["grobs"]][[2]],t = 1, l = 1)
new.y.label.grob <- gtable_add_cols(new.y.label.grob,widths = y.label.grob[["widths"]][1])
new.y.label.grob <- gtable_add_grob(new.y.label.grob,y.label.grob[["grobs"]][[1]],t = 1, l = 2)
new.y.label.grob <- gtable_add_cols(new.y.label.grob,widths = unit(1, "null"))
gp <- gtable_add_grob(gp,new.y.label.grob,t = gp$layout$t[which(gp$layout$name == "panel")],
                      l = gp$layout$l[which(gp$layout$name == "panel")])
grid.draw(gp)
```

Part 3: Performance subset of countries

Explanation

I have selected the subset of Nordic countries: Denmark, Finland, Iceland, Norway, and Sweden. The aim of this visualisation is to compare the yearly trends on the reading performance for these countries. It is immediate to see that Finland always have the highest performance among these countries. However, it has a clear trend as boys and girls performance is going down. On the other hand, it easy to see that the only country that has seen a performance increase in recent years has been Sweden. Finally, it is important to note that this visualisation shows how boys and girls yearly trend is similar in each country.

Aesthetics: *geom_line* and *facet_grid* have been used to represent each country as a line in different grids. On top of that, *aes* have been used to have different colours to represent boys and girls.

Axes: Variable 'Year' has been encoded to be a date. As the scores for PISA are done every three years, I have specified the axis x labels to be these years using *scale_x_date*. I was happy with the labels of axis y and there was no need to use *scale_y_continuous* to modify them.

Panel: Similar background features from Figure 1 have been used for this plot. An annotation line have been added on the top to follow the same style as in Part 2.

Gridlines: I felt that was unnecessary to use vertical gridlines in order to visualise the performance through the years. I used just horizontal lines (*panel.grid.major.y*). As in the previous plot, all other gridlines are hidden in the *theme*.

Legend: As in Figure 1, the legend was place on the bottom left corner of the plot. Again, l *legend.position* was set to 'bottom' and *legend.justification* to 'left'. In order to have both categories in the same line I set *legend.direction* to be 'horizontal'.

Colours: The colours of the lines are from Okabe and Ito Palette (#0072B2,#009E73) and were overrided by using *scale_color_manual*.

Title: The title was set using *ggtitle* and the style was defined in the *theme*. I used the same features as in Figure 1.

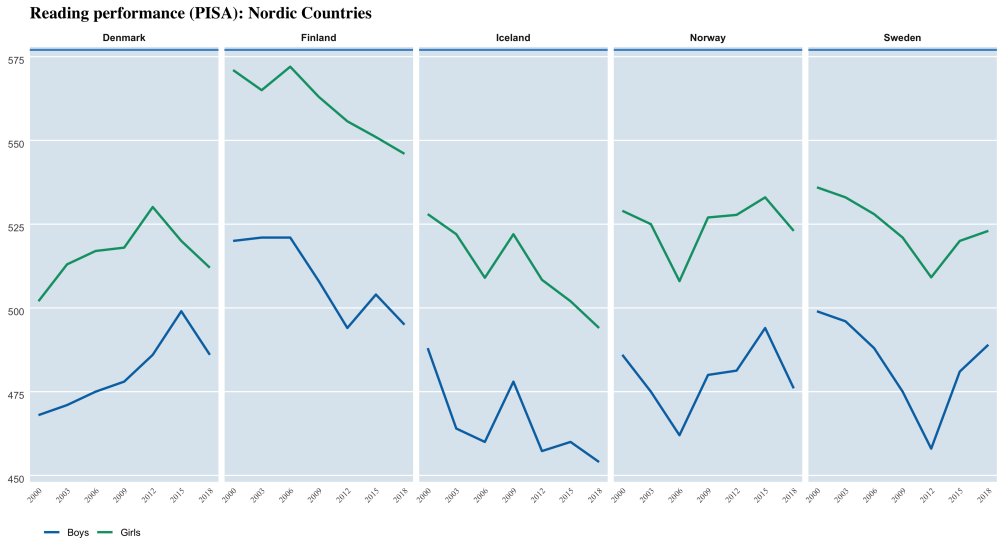

Figure 2: Reading Performance of Comparison of Nordic Countries

Code

```
#Loading all required R libraries

library(readxl)
library(tidyverse)
library(ggplot2)
library(dplyr)
library(grid)
library(gtable)
library(extrafont)
library(ggridges)
library(scales)
library(lubridate)

#I start by loading the data

data <- read.csv('OECD_PISA.csv',fileEncoding="UTF-8-BOM") %>%

  #There are just four columns that are required to create this plot
  #which I have renamed to a more appropiate name: Country, Gender, Year and AverageScore

  select('LOCATION','SUBJECT','TIME','Value') %>%

  dplyr::rename(Country = 'LOCATION',Gender = 'SUBJECT',Year = 'TIME',AverageScore = 'Value') %>%

  #This plot is using the data from 2018. In this step I filter all rows from 2018.
  #In a similar way, I am just interested with the average of Girls and Boys and therefore,
  #I am filtering out all data related to totals

  filter(Gender!='TOT') %>%
  #The country names are using a short name and I have updated them with their real name.

  mutate(CountryLong=recode(Country,"IDN" = "Indonesia", "BRA" = "Brazil","COL" ="Colombia",
                            "MEX" ="Mexico","CRI" ="Costa Rica","GRC" ="Greece","SVK" ="Slovak
                                Republic",
                            "CHL" ="Chile","ISR" ="Israel","TUR" ="Turkey","ISL" ="Iceland",
                            "LUX" ="Luxembourg","LTU" ="Lithuania","LVA" ="Latvia","HUN" ="Hungary",
                            "ITA" ="Italy","RUS" ="Russia","CHE" ="Switzerland","NLD" ="Netherlands",
                            "AUT" ="Austria","OAVG" ="OECD-Average","CZE" ="Czech Republic","SVN"
                                ="Slovenia",
                            "NOR" ="Norway","FRA" ="France","PRT" ="Portugal","BEL" ="Belgium","DEU"
                                ="Denmark",
                            "DNK" ="Germany","AUS" ="Australia","SWE" ="Sweden","NZL" ="New
                                Zealand","JPN" ="Japan",
                            "GBR" ="United Kingdom","USA" ="United States","FIN" ="Finland","POL"
                                ="Poland",
                            "KOR" ="Korea","CAN" ="Canada","IRL" ="Ireland","EST" ="Estonia"))%>%

  #Finally, I update the categories Boy and Girl to be the plural (as seen in the original plot)

  mutate(Gender = recode(Gender,"BOY" = "Boys","GIRL" = "Girls")) %>%
  mutate(Year=paste0(Year, "-01-01"))%>%
  mutate(Year= as.Date(Year, format="%Y-%m-%d"))

#The subset of countries (Nordic Countries) is filtered from the whole data set
NordicCountries <- c("Denmark","Finland","Iceland","Norway","Sweden")
data <- data %>% filter(CountryLong %in% NordicCountries)
```

```
#The plot is initialize with axis x representing the years and axis y the average score
g <- ggplot(data, aes(x=Year, y=AverageScore))

#In order to follow a simlar style than the plot from plot 2, a blue top line is added as
#an annotation
g<-g+geom_segment(y = 577,yend = 577,x = -Inf,xend = Inf,colour = "steelblue3",size = 0.8)


#By using facets, I split the plot per country using to visualise the score per gender
g<-g+ facet_grid(cols = vars(CountryLong))+
  geom_line(size = 1, aes(colour=Gender))

#I override the colours defaulted by ggplot to a more color blind friendly option
g <- g+scale_colour_manual(values=c("#0072B2","#009E73"))

#Axis x is changed to show labels for every 3 years (being aligned with the data)
#There was no need to change axis y and I have left the default values (I am happy
#with the breaks happening every 25 units)
g<-g+ scale_x_date(breaks = seq(as.Date("2000-01-01"),
                          as.Date("2018-01-01"), by="3 years"),
              labels=date_format("%Y"),
              expand = expansion(mult = c(0.2, 0.1)))

g <- g+ggtitle("Reading performance (PISA): Nordic Countries")
#Finally, a similar team than in the plot from part 2 is used
g<-g+
  theme(
    panel.background = element_rect(fill = "#dee8ef"),
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.x = element_blank(),
    axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1,size = 7.5,family = "Times New
        Roman"),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_blank(),
    legend.justification="left",
    legend.position = "bottom",
    legend.direction="horizontal",
    legend.background = element_blank(),
    legend.title = element_blank(),
    legend.key = element_rect(fill = "white"),
    plot.title = element_text(face="bold",family = "serif",size=15),
    strip.background = element_blank(),
    strip.text.x = element_text(size=9, face="bold"),
    axis.text.y = element_text(hjust = 0,vjust=1))


ggsave('part3_v2.png')
```