```
# Description -------------------------------------------------------------

# In this script, we are going to dive deeper into the
# plotting options



# Header ----------------------------------------------------------------

# The header contains all the preparatory stuff

# Load the packages
library(tidyverse)
library(forcats)

# Start with a clean sheet
rm(list = ls())

# Load the data that we prepared in 2_DataBasics.R
load("data/TradeEx_tidy.RData")


# Plotting basics ----------------------------------------------------------------

# What (statistical) variables do we have?
names(D)

# A simple scatterplot
ggplot(D) +
  geom_point(aes(gEUR, Exp_All_R))


# Give separate color to observations before 2007 ------------------

# Here is a possible solution

# Create a new so-called "indicator variable".
# For this, the ifelse() function is very convenient.
D <- D %>%
  mutate(Period = ifelse(year<2007, 1, 0))

ggplot(D) +
  geom_point(aes(gEUR, Exp_All_R, color = Period))


# Factors and visualizing group membership -------------------------------------------------


# It does not look too bad, but the legend reveals that
# R ggplot does not quite exactly what we want...

class(D$Period)
# Period is a numerical variable
# We actually want Period to be a categorical variable.
# That's something else than just character.
# As you may imagine, there is a data type for this.

# For this, we use the forcats package (see header).
# Read about this in Chapter 12.

D <- D %>%
  mutate(Period = factor(Period))

ggplot(D) + geom_point(aes(gEUR, Exp_All_R, color = Period))

# This looks more like what we wanted!


# Let's make the legend nicer
# Factors have the options to give labels to the categories. It's those
# labels that are used for the legend!
D <- D %>%
  mutate(Period = factor(Period,
            labels = c("Before 2007", "After 2007")))



ggplot(D) + geom_point(aes(gEUR, Exp_All_R, color = Period))




# Assignment: Axis labels --------------------------------------------

# Google how to make axis labels nicer!

ggplot(D) +
  #scatterplot
  geom_point(aes(gEUR, Exp_All_R, color = Period)) +

  #labels
  labs(x = "Change of CHF/Euro exchange rate (in %)",
  y = "Change in total exports (in %)")





# Visualizing statistical relationship between two variables ------------------

xlab = "Change of CHF/Euro exchange rate (in %)"
ylab =  "Change in total exports (in %)"

ggplot(D) +
  # scatterplot
  geom_point(aes(gEUR, Exp_All_R), color = "blue") +
```

```r
  #add curve that shows statistical relationship between variables
  geom_smooth(aes(gEUR, Exp_All_R), color = "red3") +
  labs(x = xlab, y = ylab)

# In the above plot, the geom_smooth curve has the default type
# "loess". This refers to a so-called non-parametric statistical
# method to gauge the relationship between two variables. Loosly
# speaking, the curve provides a "best guess" of an estimate for the
# y variable, given the x variable, under a method that allows quite
# flexibly for "locally" diverse behavior of the relationship.

# The antipode is a classical linear regression with
# an intercept and slope:

ggplot(D) +
  # scatterplot
  geom_point(aes(gEUR, Exp_All_R), color = "blue") +

  #add curve that shows statistical relationship between variables
  geom_smooth(aes(gEUR, Exp_All_R), method = "lm" ,color = "red3") +
  labs(x = xlab, y = ylab)


# If you want to add a title

ggplot(D) +
  # scatterplot
  geom_point(aes(gEUR, Exp_All_R), color = "blue") +

  #add curve that shows statistical relationship between variables
  geom_smooth(aes(gEUR, Exp_All_R), method = "lm" ,color = "red3") +

  # labels, title, caption
  labs(x = xlab, y = ylab,
       title = "Surprisingly little effect",
       subtitle = "Swiss exports and the CHF/euro exchange rate",
       caption = "Data source: Swiss National Bank")


# Check out
# http://ggplot2.tidyverse.org/reference/geom_smooth.html



# Changing the background and other features ------------------------------------------------



# If you do not like the background:


ggplot(D) +
  # scatterplot
  geom_point(aes(gEUR, Exp_All_R), color = "blue") +
  #add curve that shows statistical relationship between variables
  geom_smooth(aes(gEUR, Exp_All_R), method = "lm" ,color = "red3") +

  # labels, title, caption
  labs(x = xlab, y = ylab,
       title = "Surprisingly little effect",
       subtitle = "Swiss exports and the CHF/euro exchange rate",
       caption = "Data source: Swiss National Bank") +

  # background
  theme_bw() +

  # customizing the caption
  theme(plot.caption = element_text(color = "#999999", size = 7, hjust = 1))
  # See http://sharpsightlabs.com/blog/format-titles-and-axes-in-ggplot2/


# Google which themes there are for background.
# For instance:
# http://www.sthda.com/english/wiki/ggplot2-themes-and-background-colors-the-3-elements

# for some cool themes, you need an additional package ggthemes. See
# https://cran.r-project.org/web/packages/ggthemes/vignettes/ggthemes.html


# Saving plots ------------------------------------------------------------


# First create a new folder Plots inside your standard working directory!



ggplot(D) +
  # scatterplot
  geom_point(aes(gEUR, Exp_All_R), color = "blue") +

  #add curve that shows statistical relationship between variables
  geom_smooth(aes(gEUR, Exp_All_R), method = "lm" ,color = "red3") +

  # labels, title, caption
  labs(x = xlab, y = ylab,
       title = "Surprisingly little effect",
       subtitle = "Swiss exports and the CHF/euro exchange rate",
       caption = "Data source: Swiss National Bank") +

  # background
  theme_bw() +

  # customizing the caption
  theme(plot.caption = element_text(color = "#999999", size = 7, hjust = 1))
# See http://sharpsightlabs.com/blog/format-titles-and-axes-in-ggplot2/
```

```
ggsave("Plots/Exp_All_R_vs_gEUR.png")

ggsave("Plots/Exp_All_R_vs_gEUR.pdf")




# Regressions -------------------------------------------------------------



reg = lm(Exp_All_R ~ gEUR + gUSD, data = D)
reg
summary(reg)

# We will talk more about regression output in the context of R Markdown
```