

workshop06

February 3, 2019

```
In [ ]: library(tidyverse)
        library(ggmap)

        # Plot size depending on your screen resolution to 5 x 3
        options(repr.plot.width=5, repr.plot.height=5)
```

1 Welcome to Workshop 6

1.0.1 Exercise 1

Load the file EconDataPop.csv. This file provides economic data across countries.

Here is a translation of region names: * Americas * Asia_pac (Asia-Pacific) * Eur_Asia (East Europe and Central Asia) * MENA (Middle East and North Africa) * SSA (Sub-Saharan Africa) * W_Eur (Europe)

Here is a translation of two of the variable names: * HDI (Human Development Index) * CPI (Corruption Perceptions Index)

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

Create a scatter plot between HDI and CPI:

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

So far, we always added colour or text to the points. Another possibility is to vary the size of the points. For this, add as aesthetic mapping colour=Region and size=Population. The result is called a bubble-chart.

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

Somehow the points are overlapping. For tuning of the plot, explore the transparency. You can do this with alpha=0.5 inside geom_point (values from 0-1). In addition, a scale can be included by adding + scale_size(range = c(0,15)) Try out a few values. Which one makes sense ?

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

Next, what kind of line should go through this data? Let's start with a linear regression. In the first stage, let's calculate it separately and add it as a column to the `econ.df`. To run a simple regression, the syntax is:

```
In [ ]: reg <- lm(HDI ~ CPI, data=econ.df)
        reg
```

To use this simple model for prediction, call `predict(reg)` and assign it to the data frame as column called `linearFitHDI`.

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

To plot this column, add

```
geom_line(aes(y = linearFitHDI, colour = NULL,
size = NULL))
```

Here we inherit the mappings from the previous layers and overwrite the values we want to change.

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

What happens? You will see that we have inherited settings for the legend but it changed. Let's fix this. Specify only the x and y in the first aes mapping and specify the geometric shape specific options `geom_point` separately (e.g. colour).

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

What do you think about this fit? Let's try a loess curve (local polynomial regression with a confidence interval pointwise). Replace the line with `geom_smooth(method = 'loess')`

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

Let's try this again with `method="lm, method = 'lm', se=FALSE"`

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

How does this graph compare to the previous one? By default the formula $y \sim x$ is used. You can modify it by specifying `formula=` as an option. Let's try log scale: `formula = y~log(x)`

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

To recap, we have learned how to build graphs by building it up from one data frame while making use of inheritance across the geometries. It is also possible to use only a subset of the data for some parts. Let's label each country that has the highest human development index by Region. To do this, group by Region first, then within each group we sort the data according to the HDI and pick the last element (the highest). After this, we filter all rows that are matching with our newly created variable:

```
In [ ]: subset <- econ.df %>%
        group_by(Region) %>%
        mutate(countrySelect=last(Country, order_by = HDI)) %>%
        filter(countrySelect==Country)

subset
```

Execute each line individually to see what is happening. For this copy it into a new cell.

Lets us this information and add it to our plot:

```
geom_text_repel(data=subset, aes(label=Country,
colour = Region), show.legend = FALSE)
```

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

1.0.2 Exercise 2

For the remainder of class, re-visit all plotting commands you have learnd so far and create visuillisations. What do you learn? Discuss it with your table. E.g Plot the name of the country with the lowest HDI index (Hint: use first instead of last)

```
In [ ]: # your code here
        fail() # No Answer - remove if you provide an answer
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs