PP4RS | R Module

Slot 6

Dora Simon

10.09.2018

Outline of the R-Module

Slot 1: Intro & Data Types

Slot 2: Conditionals and Functions & Loops

Slot 3: Read in Data

Slot 4: Data Manipulation

Slot 5: Regressions

Slot 6: Graphs

Slot 7: knitR

Now: Graphs

Not all graphs are created equal

Principles for nice graphs[1]

> Once upon a time, a picture was worth a thousand words. But with online news, blogs, and social media, a good picture can now be worth so much more. Economists who want to disseminate their research, both inside and outside the seminar room, should invest some time in thinking about how to construct compelling and effective graphics.

"Picture Superiority Effect": ability of humans to retain more information seen through pictures than through words

[1] Source: Schwabish, Jonathan A. 2014. "An Economist's Guide to Visualizing Data." Journal of Economic Perspectives, 28 (1): 209-34. Link to the Paper

Principles for nice graphs

1. Show the data
    - People look at graphs at first
    - Data are most important part of the graph

Principles for nice graphs

1. Show the data
    - People look at graphs at first
    - Data are most important part of the graph
2. Reduce the clutter
    - unnecessary or distracting elements should be avoided
    - examples of clutter:

Principles for nice graphs

1. Show the data
    - People look at graphs at first
    - Data are most important part of the graph
2. Reduce the clutter
    - unnecessary or distracting elements should be avoided
    - examples of clutter:
        - dark or heavy gridlines
        - unnecessary tick marks, labels, or text
        - unnecessary icons or pictures
        - ornamental shading and gradients
        - unnecessary dimensions

Principles for nice graphs

1. Show the data
    - People look at graphs at first
    - Data are most important part of the graph
2. Reduce the clutter
    - unnecessary or distracting elements should be avoided
    - examples of clutter:
        - dark or heavy gridlines
        - unnecessary tick marks, labels, or text
        - unnecessary icons or pictures
        - ornamental shading and gradients
        - unnecessary dimensions
3. Integrate the text and the graph
    - graphs should contain enough information to stand alone
    - integrated legends (right below the title, on the chart, at the end of a line) are better

Principles for nice graphs

1. Show the data
    - People look at graphs at first
    - Data are most important part of the graph
2. Reduce the clutter
    - unnecessary or distracting elements should be avoided
    - examples of clutter:
        - dark or heavy gridlines
        - unnecessary tick marks, labels, or text
        - unnecessary icons or pictures
        - ornamental shading and gradients
        - unnecessary dimensions
3. Integrate the text and the graph
    - graphs should contain enough information to stand alone
    - integrated legends (right below the title, on the chart, at the end of a line) are better

There is a Cheatseet for this.

Let's have a look at good and bad graphs in the paper itself!

Details

For more details, check the paper yourself. It includes topics like:

- Colors: tools for creating nice, consistent colors and that also work with greyscale when printed
- Fonts
- Maps
- Infographic tools

Website of the author with more data visualization info.

Graphs in R

How to graph in R

As you can imagine, there are lots of packages and functions around graphs.

Base functions:

- `plot()`
- `lines()`
- `barplot()`
- `hist()`
- `pie()`

How to graph in R

As you can imagine, there are lots of packages and functions around graphs.

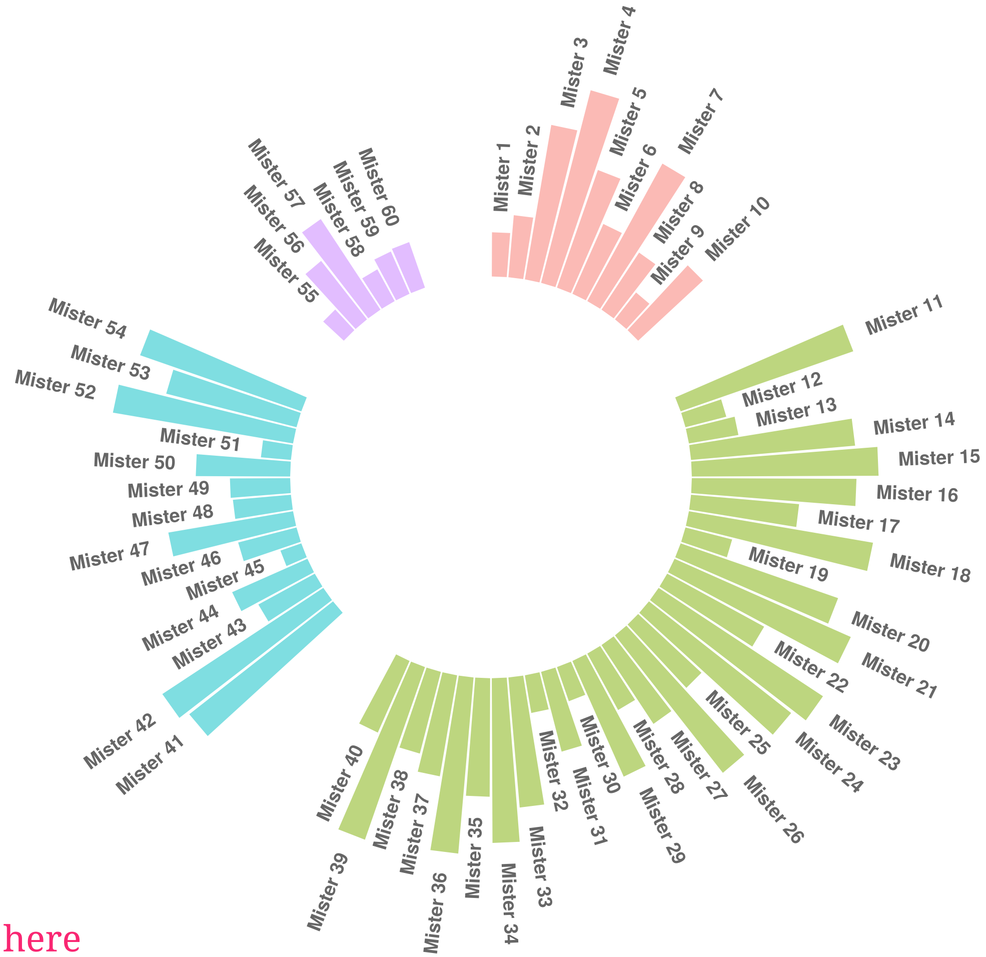
Base functions:

- `plot()`
- `lines()`
- `barplot()`
- `hist()`
- `pie()`

We will use `ggplot2`, the most common package for graphs.

- It can do everything that the base functions can do
- Has a nicer syntax
- Better output quality
- More options

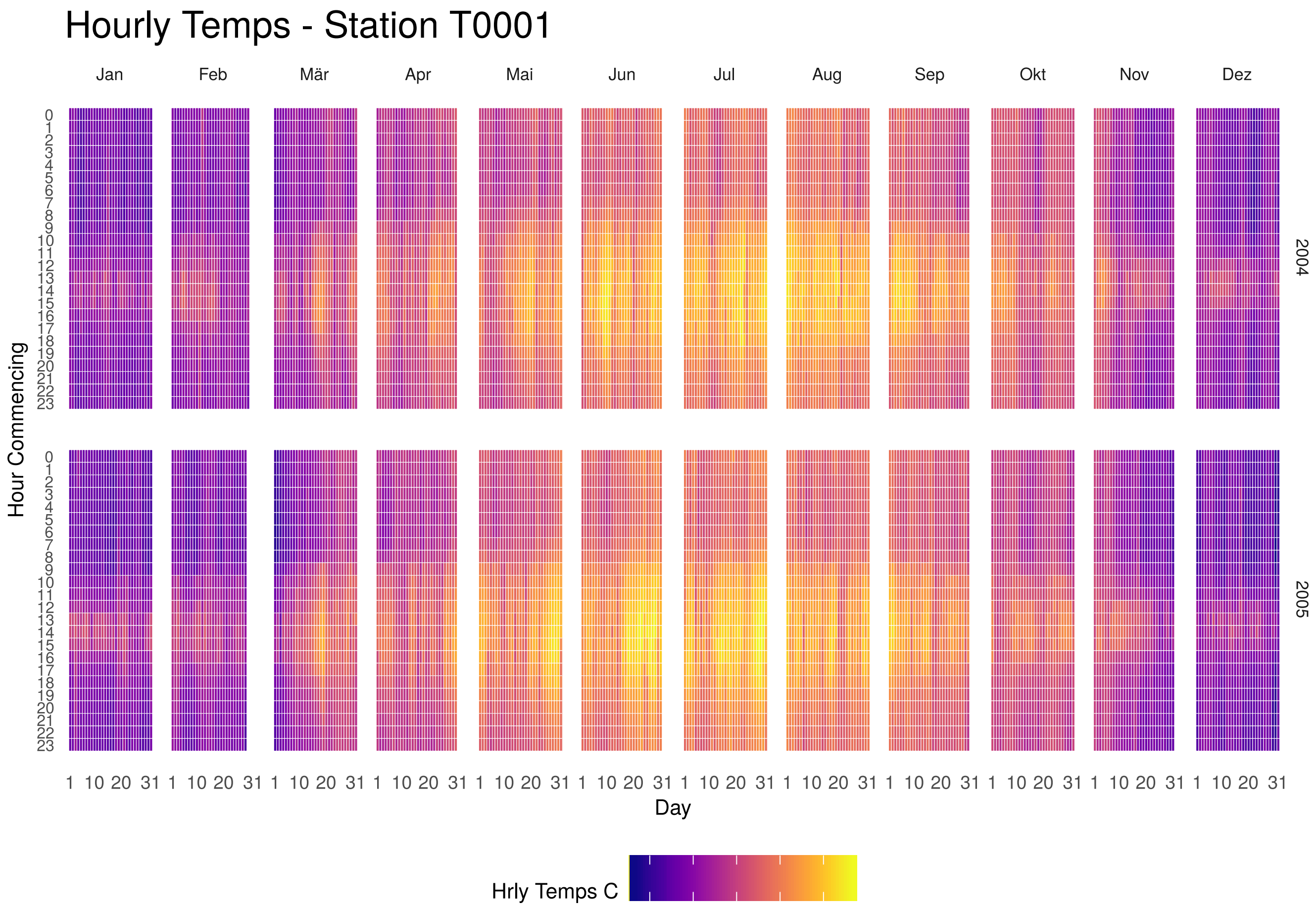
The power of ggplot



Code from here

.

The power of ggplot



Code from here

.

Resources

- base plots in R: Examples with code
- fancy graphs in R: R Graph Gallery
  - Look at fancy graphs and their code
  - All sorts of functions, not only ggplot
- ggplot2 cheatsheet
- ggplot2 references - better than the usual helpfiles!

Some examples are found in a different slideset.

Exercise

Exercise

We are going to replicate Figure 1 from the Income and Democracy paper of Friday.

Follow these steps:

1. Read in the raw data
    - Read in the sheet named "F1" from the raw data folder of Friday
    - Rename `lrgdpch` to `log_gdp_pc` and `fhpolrigaug` to `freedom_house`
2. Start the plot
    - Log GDP per Capita should be on the x-axis
    - the measure of democracy should be on the y-axis
3. Add a scatterplot and choose a good point size
4. Fit a smooth line to the data
    - choose the method to be used
    - you can adjust the appearance such as the color, the size and the transparency
5. Adjust the axis labels to be nicely readable
6. Add the country names as labels into the plot (hint: use `geom_text()`)
    - you can adjust their size as well

Additional Exercise

Recreate Figure 2 form the Income and Democracy paper of Friday.

Follow these steps:

1. Read in the raw data
   - Read in the sheet named "F2" from the raw data folder of Friday
   - Rename `s5fhpolrigaug` to `freedom_house_change` and `s5lrgdpch` to `log_gdp_pc_change`
2. Repeat the same steps as before for creating the plot
   - adjust the axis labels
3. Save your plot with `ggsave`

Solutions

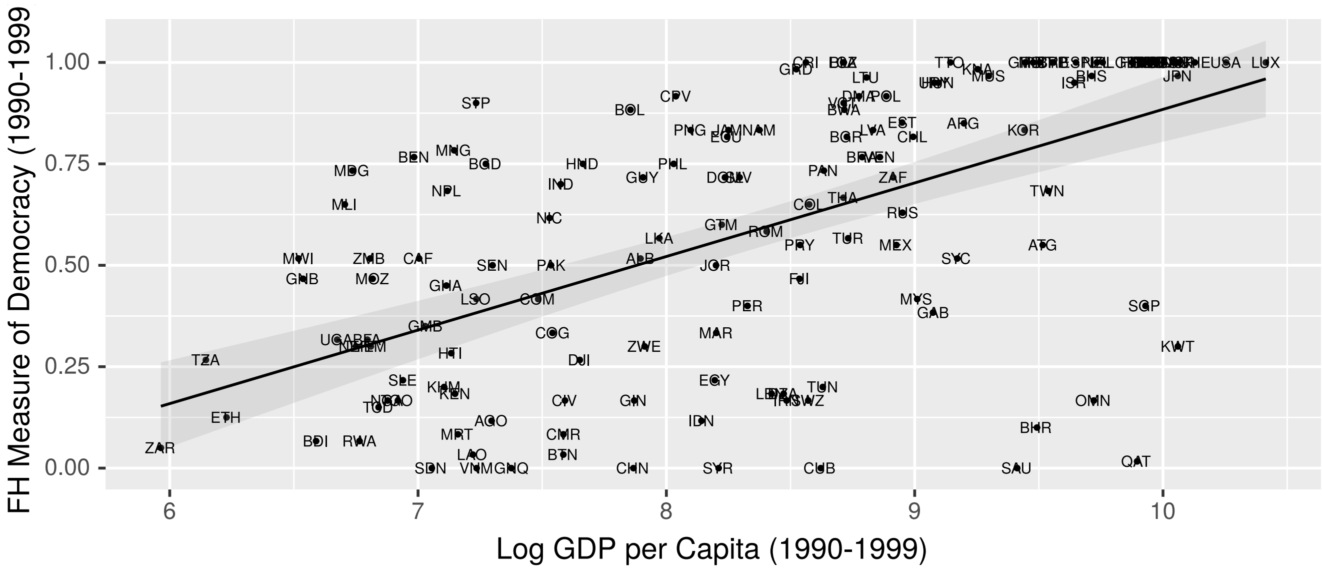
Read in the raw data

- Read in the sheet named "F1" from the raw data folder of Friday
- Rename `lrgdpch` to `log_gdp_pc` and `fhpolrigaug` to `freedom_house`

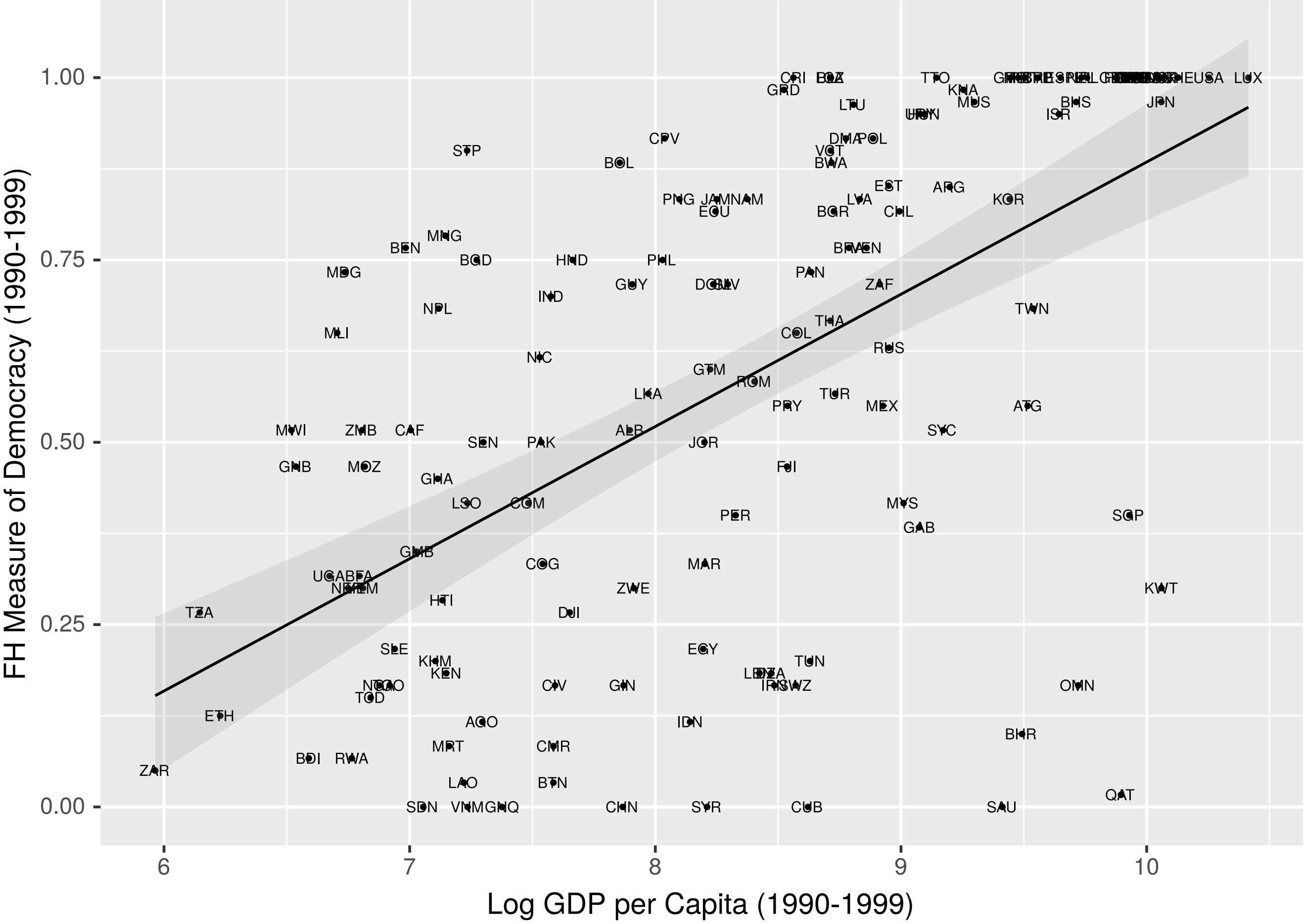
```r
library(readxl)
ajry_f1 <- read_xls("../day3/replicating-papers/raw_data/ajry.xls",
                   sheet = "F1") %>%
  rename(log_gdp_pc = lrgdpch,
         freedom_house = fhpolrigaug)
```

Solutions

```r
ggplot(ajry_f1, aes(x = log_gdp_pc, y = freedom_house)) +
  geom_point(size = 0.5) +
  geom_smooth(method = "lm", color = "black", size = 0.5,
              alpha = 0.2) +
  labs(x = "Log GDP per Capita (1990-1999)",
       y = "FH Measure of Democracy (1990-1999)") +
  geom_text(aes(label = code), size = 2)
```



Solutions


FH Measure of Democracy (1990-1999)
1.00
0.75
0.50
0.25
0.00
Log GDP per Capita (1990-1999)
6
7
8
9
10
CRI
GRD
TTO
KNA
MUS
USA
LUX
BHS
ISR
JPN
LTU
CPV
STP
BOL
DMA
POL
VCT
BWA
EST
ARG
KOR
PNG
JAM
NAM
ECU
LVA
BGR
CHL
MNG
BEN
BGD
HND
PHL
MDG
GUY
PAN
ZAF
IND
NPL
TWN
THA
MLI
COL
RUS
NIC
GTM
ROM
LKA
TUR
PRY
MEX
ATG
MWI
ZMB
CAF
ALB
SYC
SEN
PAK
JOR
GNB
MOZ
FJI
GHA
LSO
COM
MYS
PER
SGP
GAB
GMB
COG
MAR
UGA
BFA
ZWE
KWT
HTI
TZA
DJI
SLE
EGY
KHM
TUN
KEN
CIV
GIN
OMN
TCD
ETH
AGO
IDN
BHR
MRT
CMR
BDI
RWA
ZAR
LAO
BTN
QAT
SDN
VNM
GNQ
CHN
SYR
CUB
SAU

Solutions

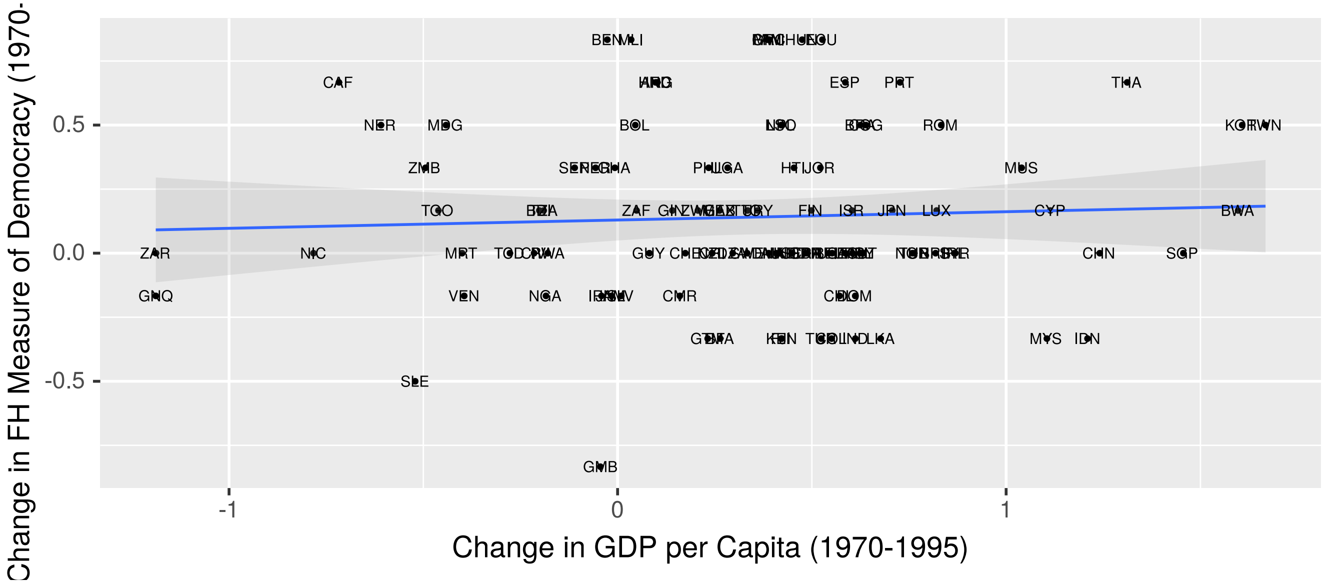
Read in the raw data

- Read in the sheet named "F1" from the raw data folder of Friday
- Rename `s5fhpolrigaug` to `freedom_house_change` and `s5lrgdpch` to `log_gdp_pc_change`

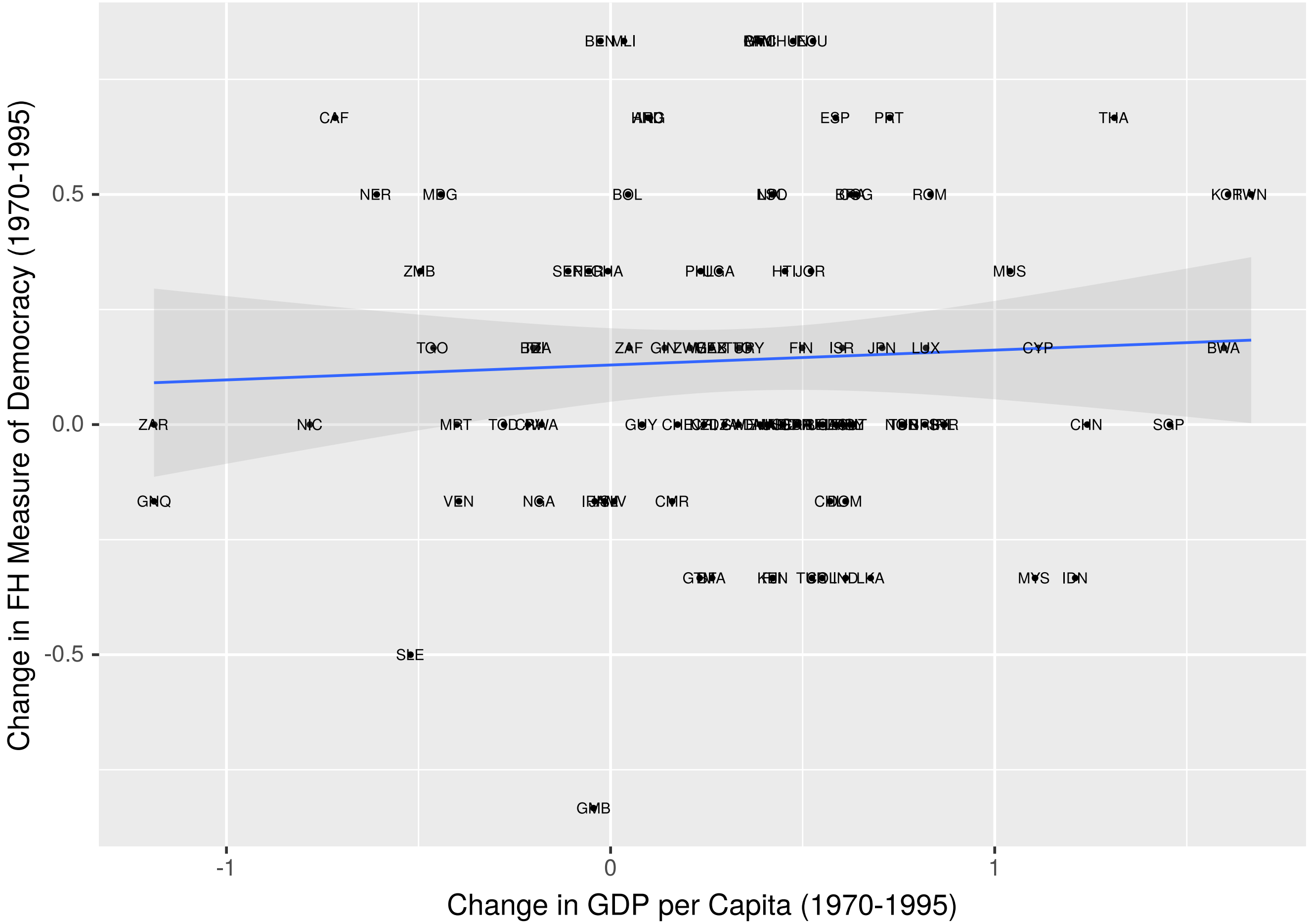
```r
library(readxl)
ajry_f2 <- read_xls("../day3/replicating-papers/raw_data/ajry.xls",
                   sheet = "F2") %>%
  rename(freedom_house_change = s5fhpolrigaug,
         log_gdp_pc_change = s5lrgdpch)
```

Solutions

```
ggplot(ajry_f2, aes(x=log_gdp_pc_change, y=freedom_house_change)) +
  geom_point(size = 0.5) +
  geom_smooth(method = "lm", size = 0.5, alpha = 0.2) +
  geom_text(aes(label = code), size = 2) +
  labs(x = "Change in GDP per Capita (1970-1995)",
       y = "Change in FH Measure of Democracy (1970-1995)")
```



Solutions


Change in FH Measure of Democracy (1970-1995)
Change in GDP per Capita (1970-1995)
0.5
0.0
-0.5
-1
0
1
BEN
MLI
CAF
ESP
PRT
THA
NER
MDG
BOL
ROM
ZMB
SEN
PHL
HTI
JOR
MUS
TGO
ZAF
FIN
ISR
JPN
LUX
CYP
BWA
ZAR
NIC
MRT
RWA
GUY
CHN
SGP
GNQ
VEN
NGA
CMR
DOM
KEN
IND
LKA
MYS
IDN
SLE
GMB

Solutions

Saving plots:

```
#my_plot <- ggplot() + geom_point()
#ggsave("my_plot.png", my_plot)
```

In order to view a plot, just print it (e.g. `my_plot` or `print(my_plot)`)