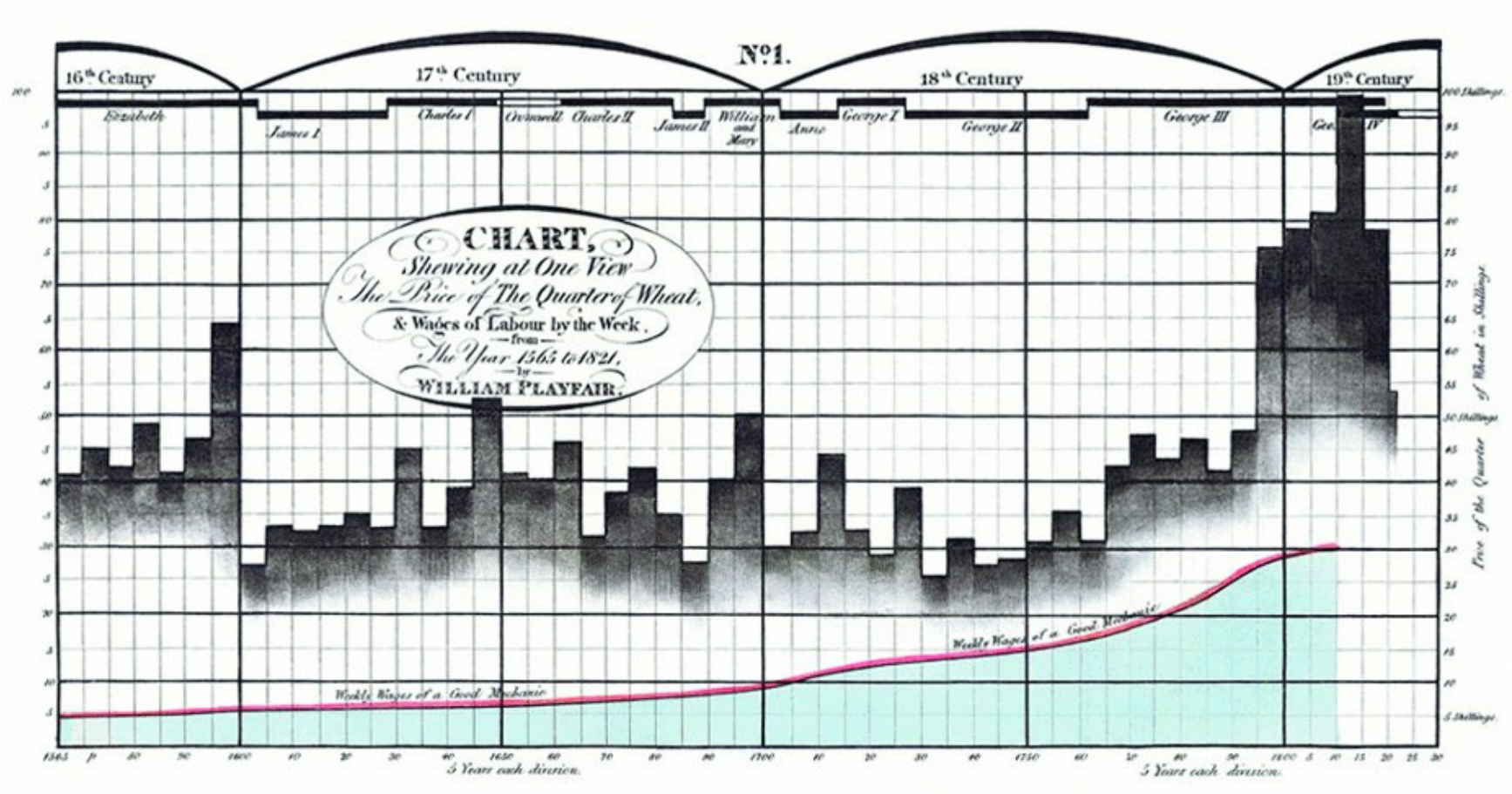
REPORTS IN R MARKDOWN & OTHER TOOLS

LECTURE 3 INTRO TO R PROGRAMMING

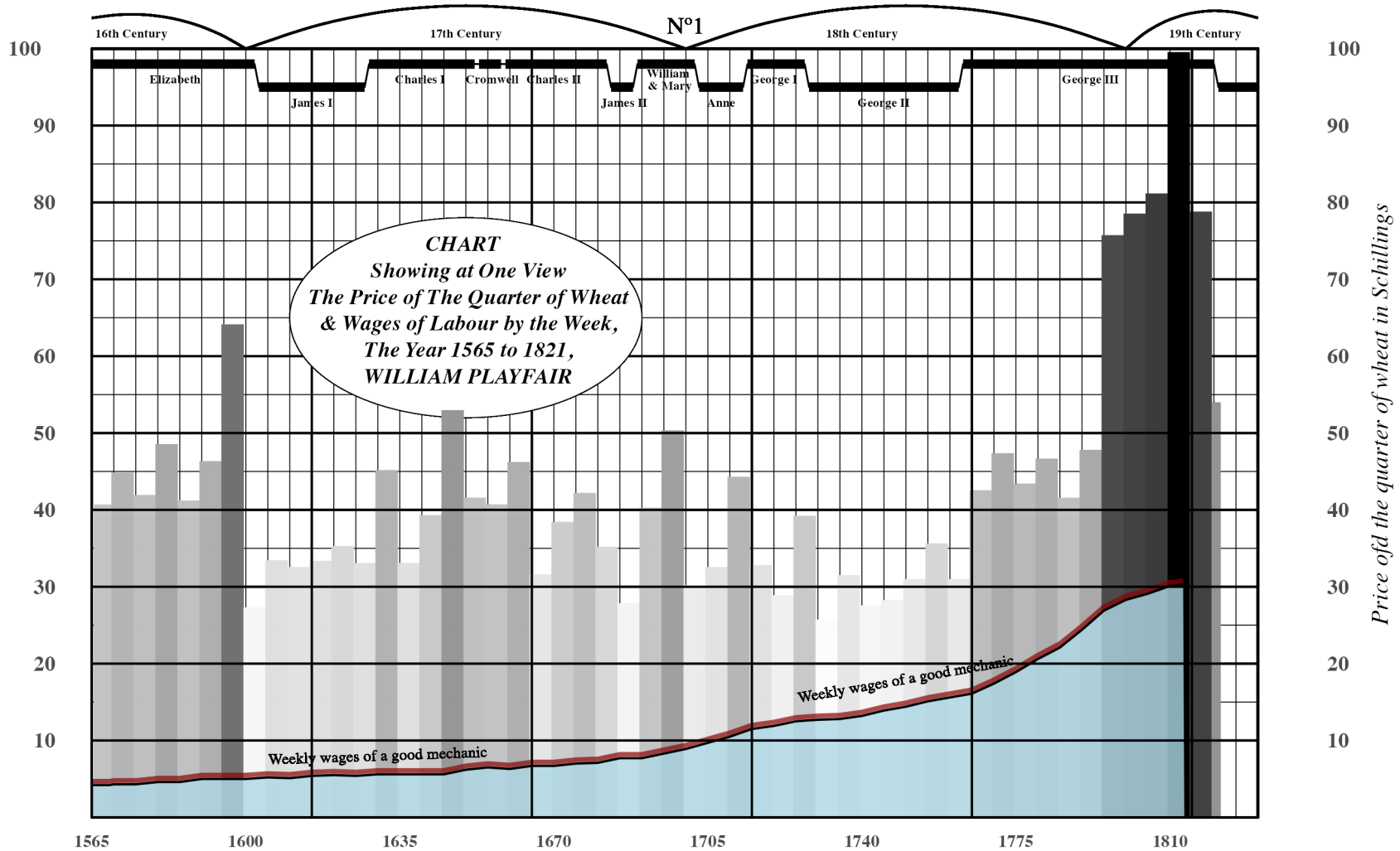
MARIA MONTOYA-AGUIRRE

M1 APE @ PARIS SCHOOL OF ECONOMICS

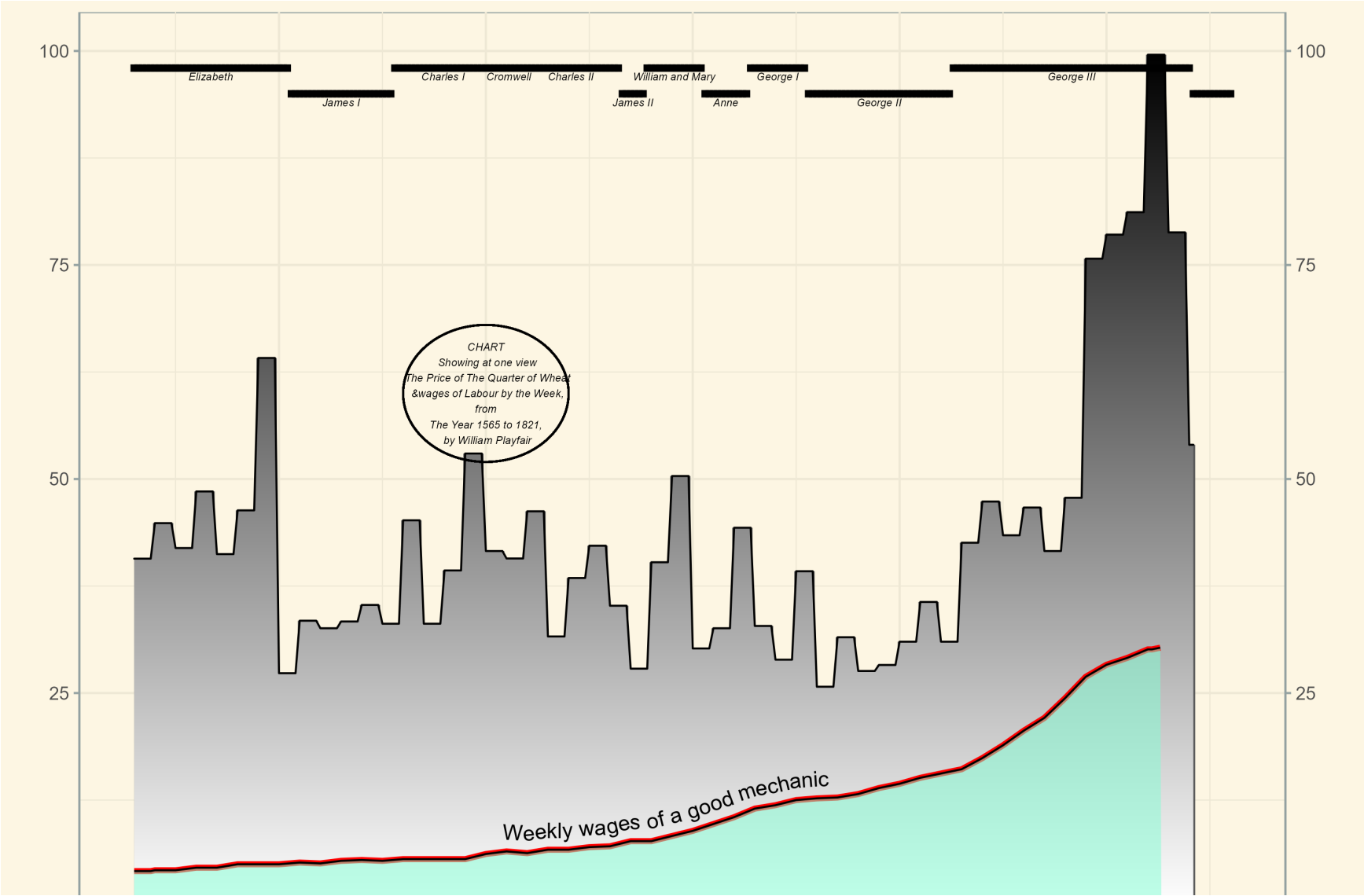
HOMEWORK REVIEW



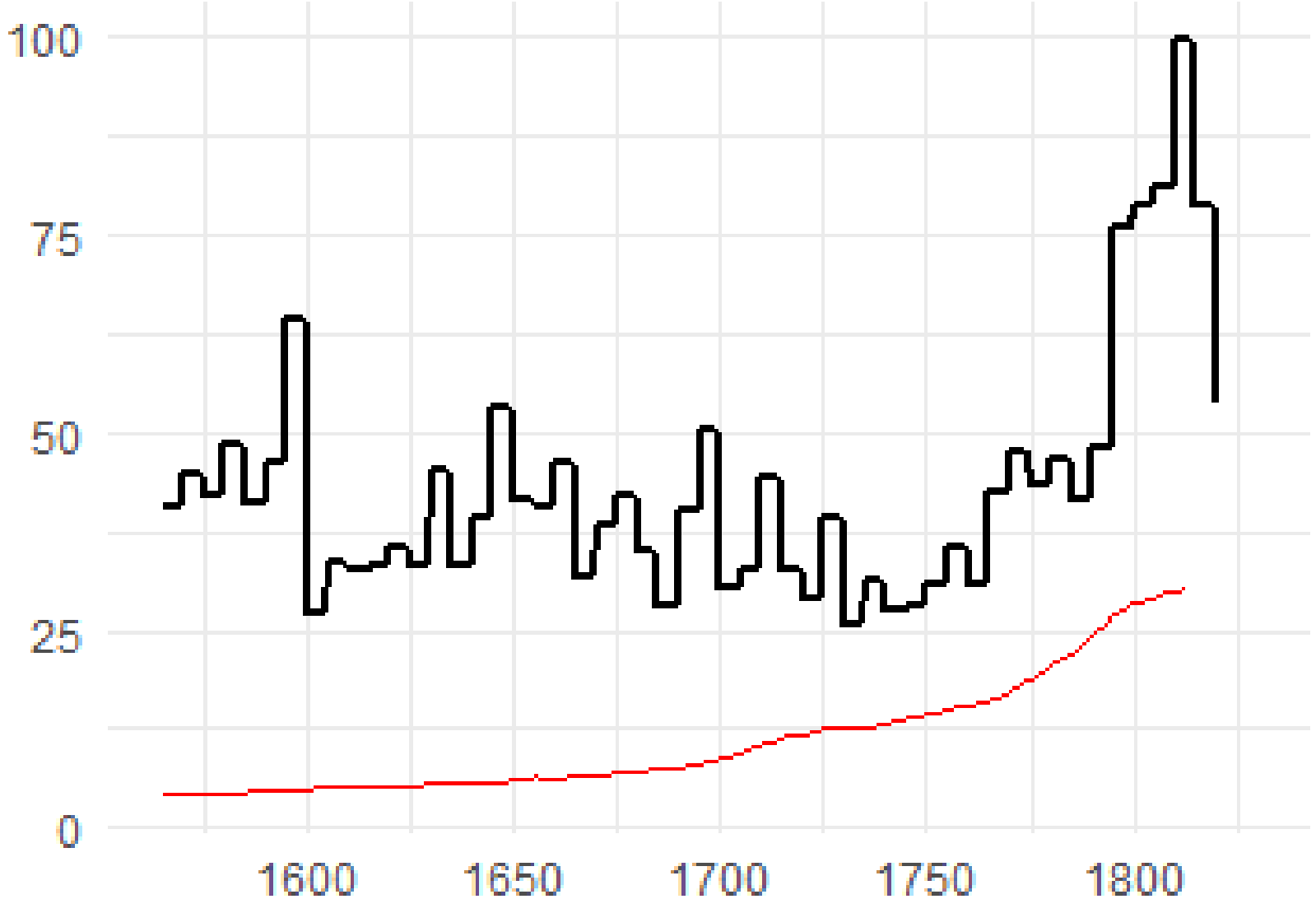
HOMEWORK REVIEW



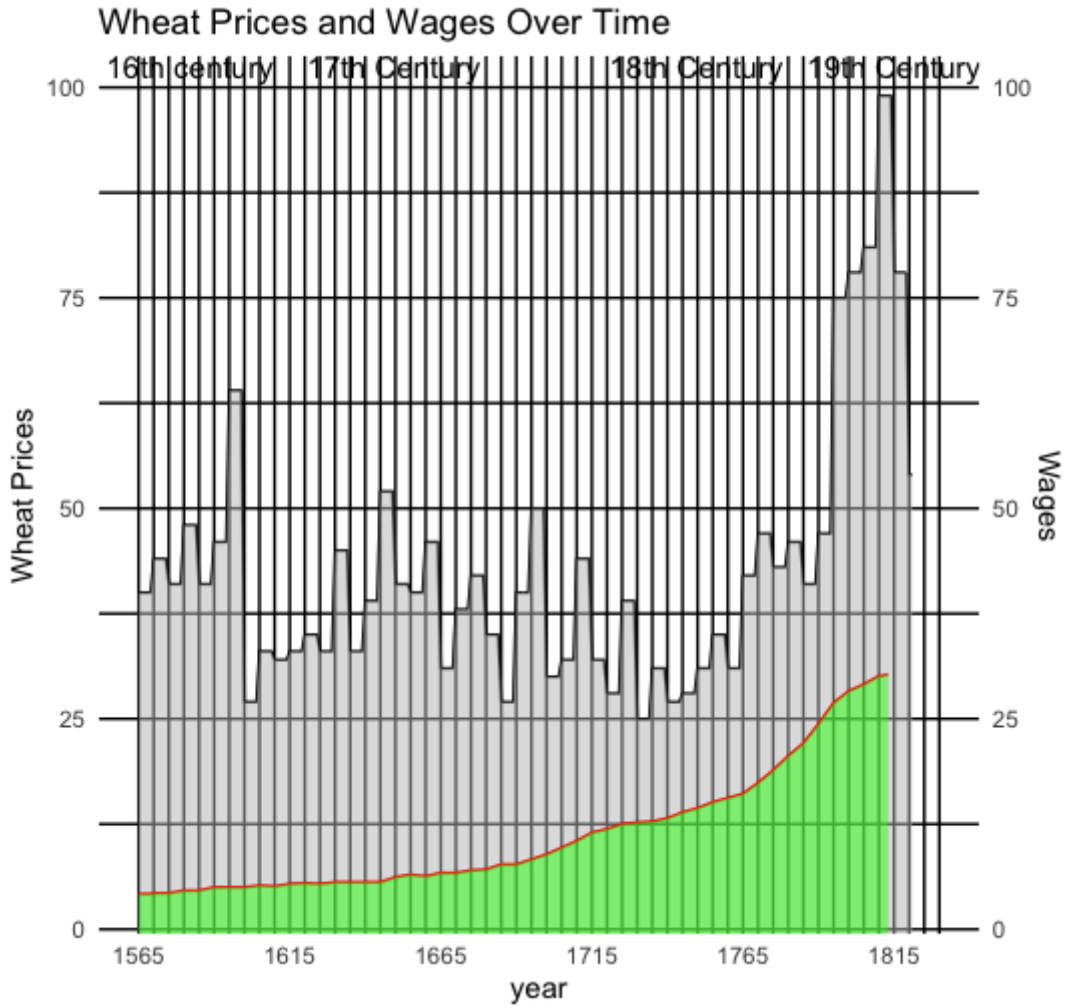
YOUR HOMEWORKS



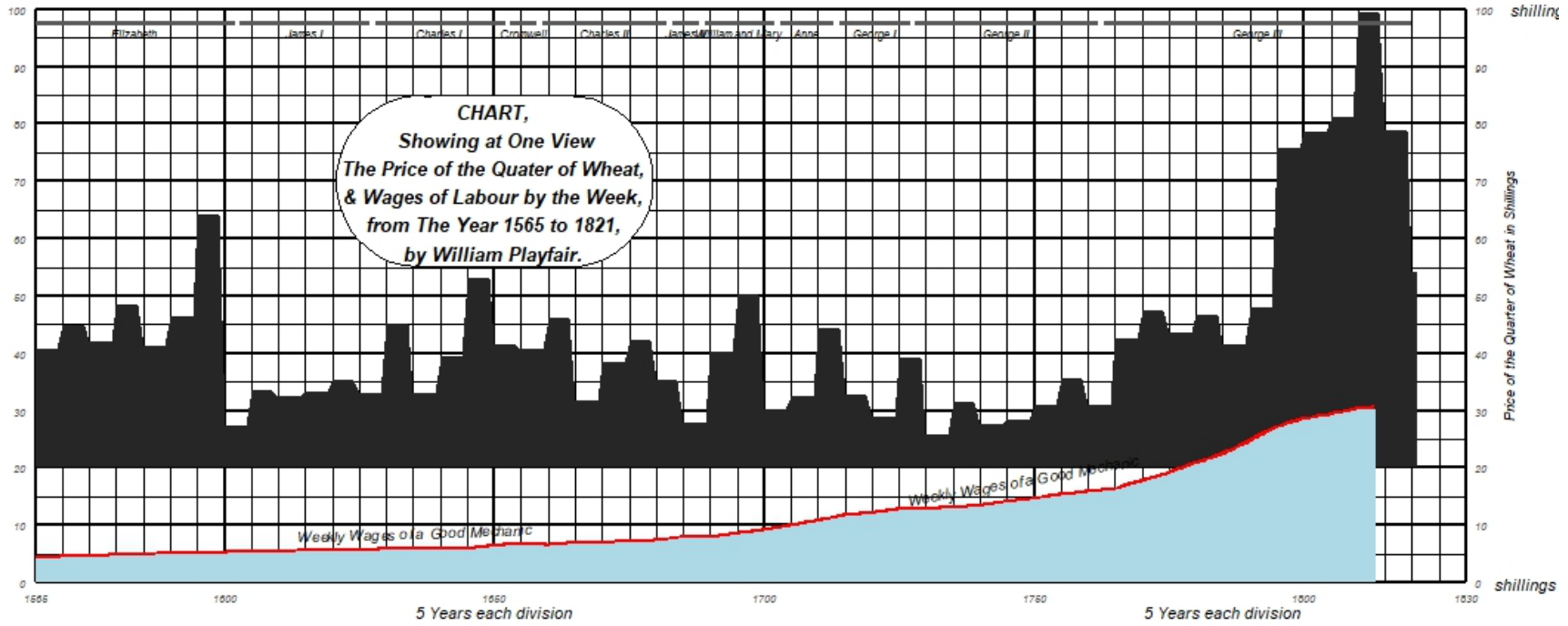
YOUR HOMEWORKS



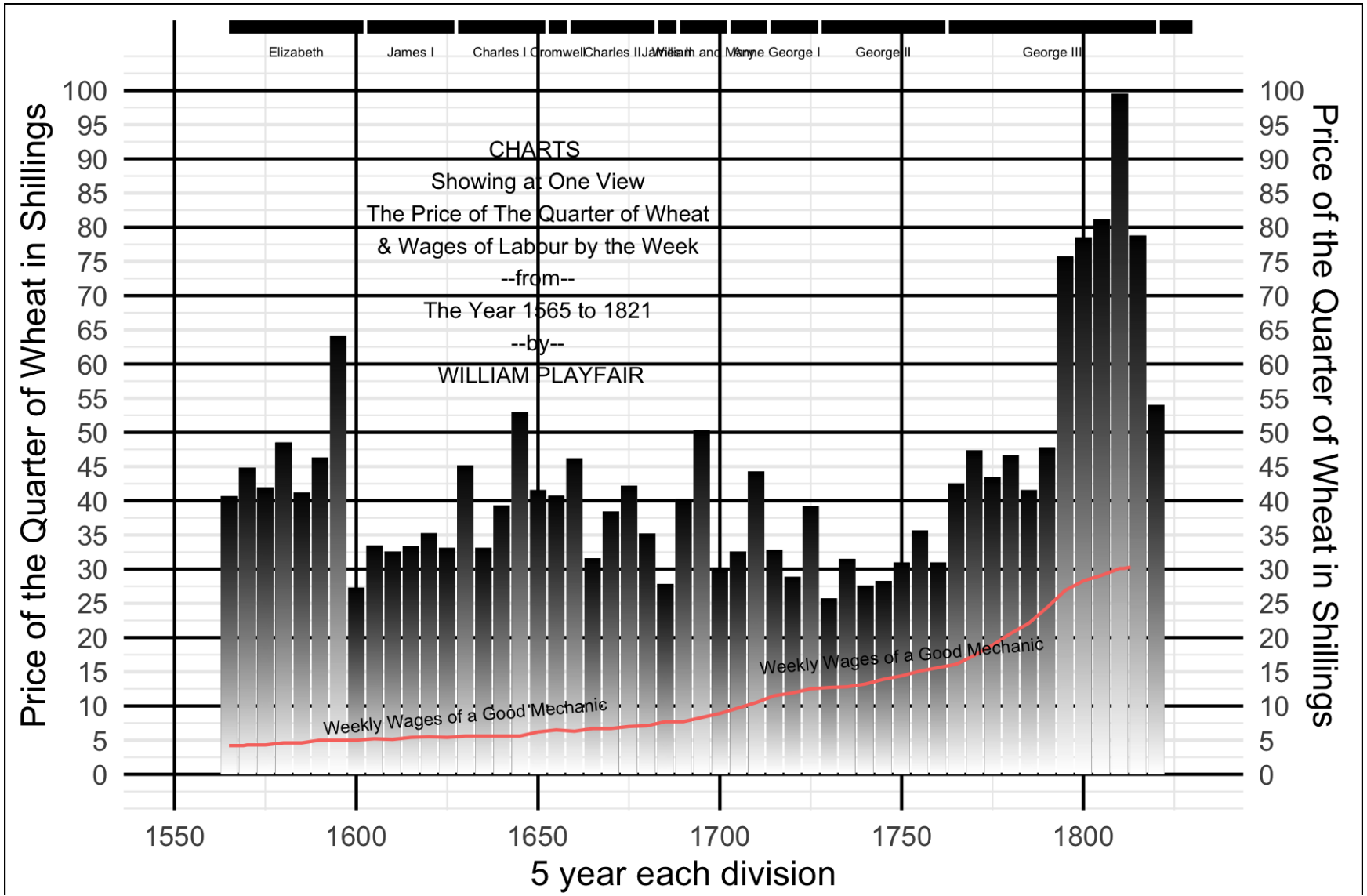
YOUR HOMEWORKS



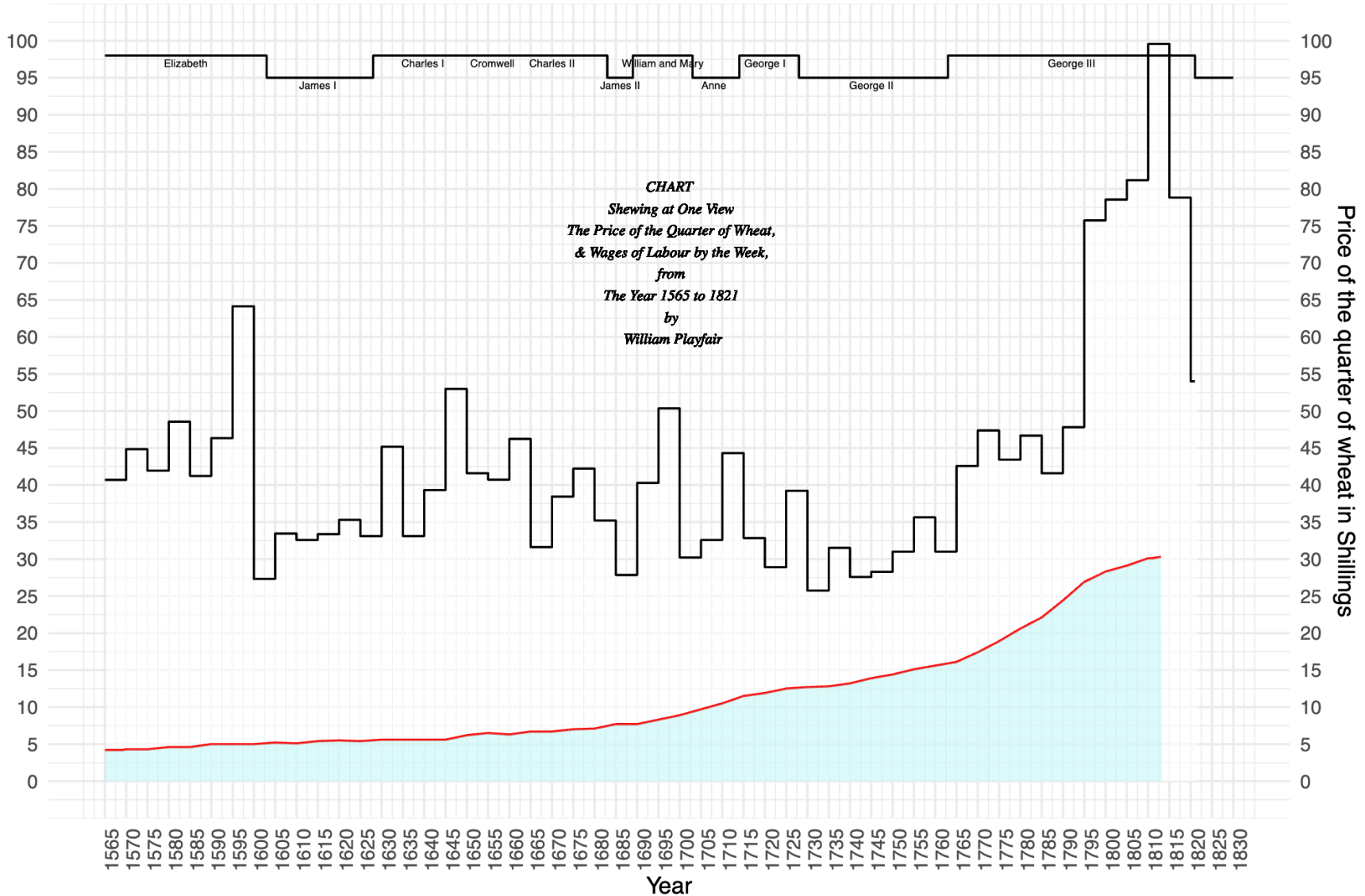
YOUR HOMEWORKS



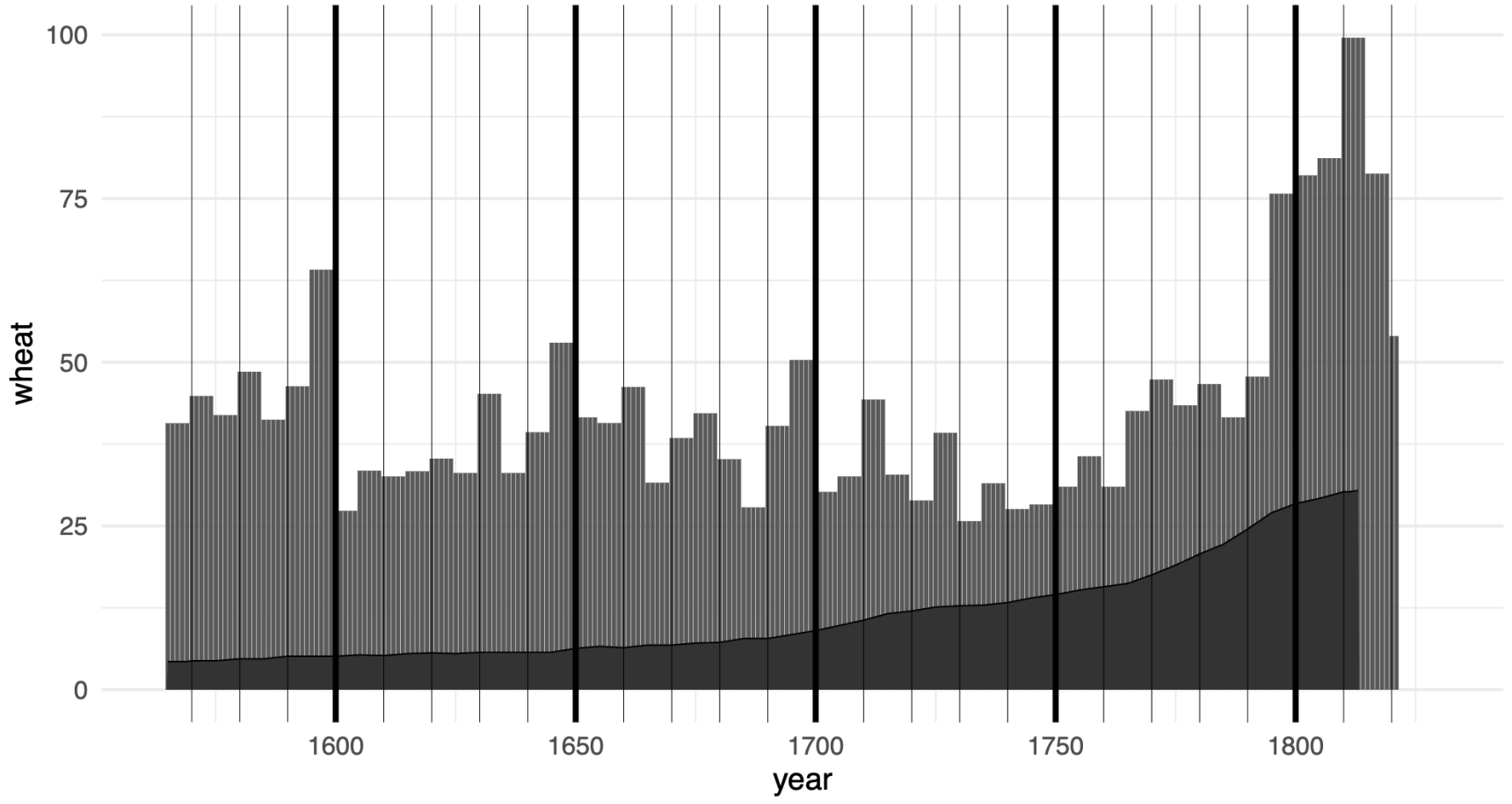
YOUR HOMEWORKS



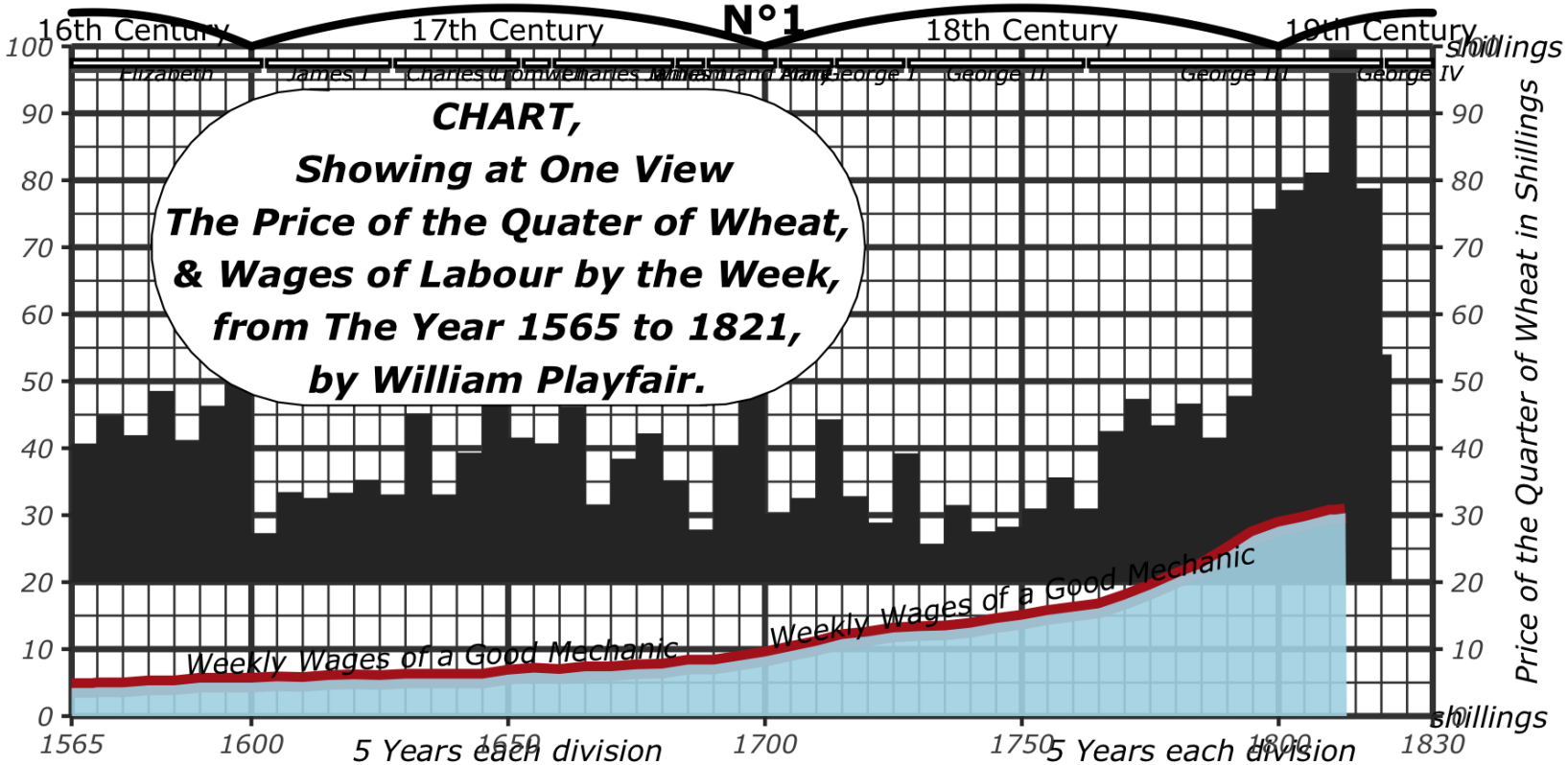
YOUR HOMEWORKS



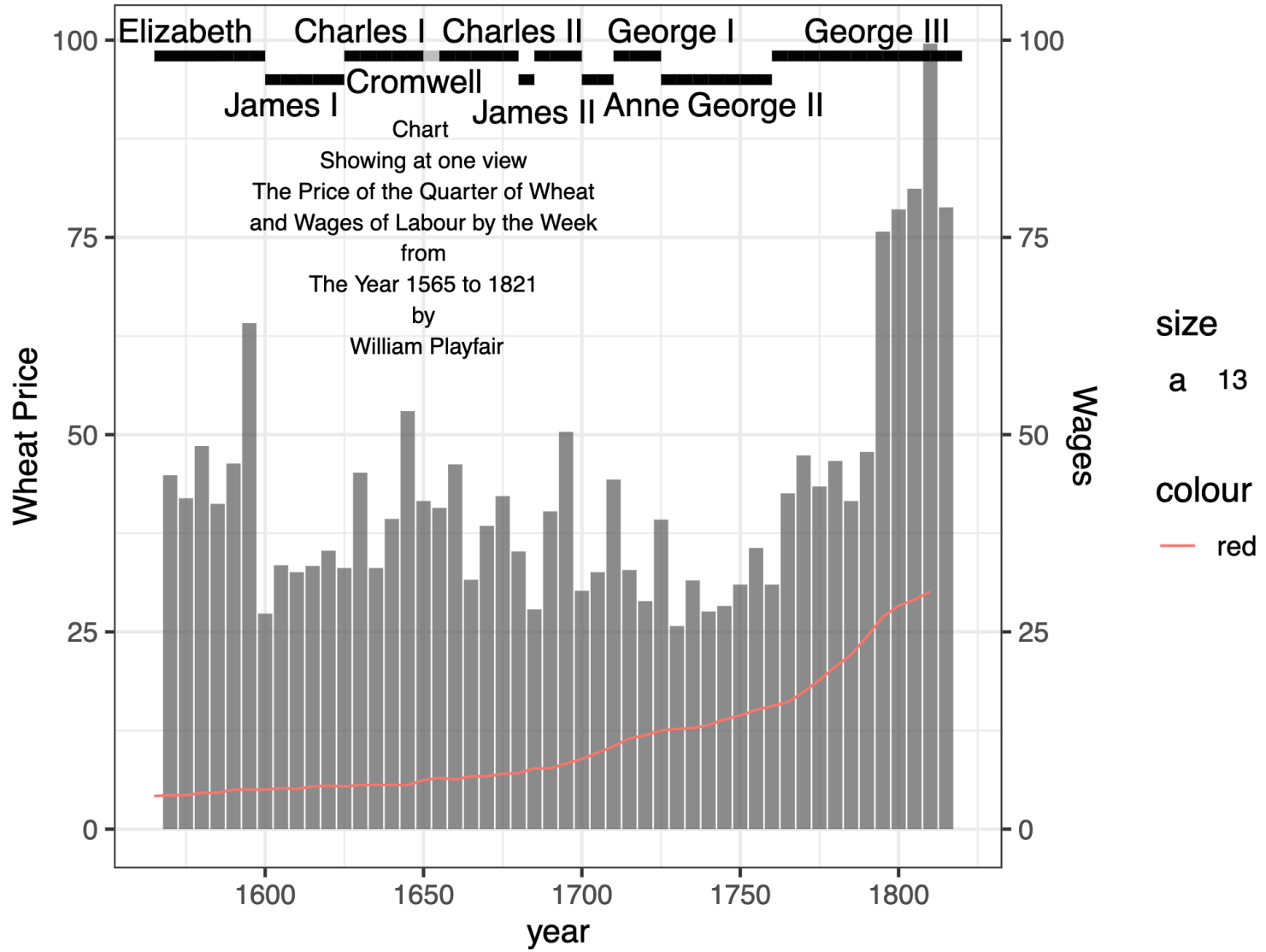
YOUR HOMEWORKS



YOUR HOMEWORKS



YOUR HOMEWORKS



WARM UP

- We are going to use the "Fichier des prénoms" data
  - This is where the INSEE reports the birth count associated with each first name in France
  - It is virtually exhaustive from 1946, when the INSEE was founded

```
names <- read.csv("../data/03_fichier_prenoms.csv", sep = ";", encoding = "UTF-8")
str(names)
```

```
## 'data.frame':    686538 obs. of  4 variables:
##  $ sexe    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ preusuel: chr  "_PRENOMS_RARES" "_PRENOMS_RARES" "_PRENOMS_RARES" "_PRENOMS_RARES" ...
##  $ annais  : chr  "1900" "1901" "1902" "1903" ...
##  $ nombre  : int  1249 1342 1330 1286 1430 1472 1451 1514 1509 1526 ...
```

- `sexe:` 1 for Male and 2 for Female
- `preusuel:` first name (_PRENOMS_RARES gathers rare first names for anonymity considerations)
- `annais:` birth year (XXXX groups unknown birth years)
- `nombre:` number of newborns for the corresponding sex/name/year

WARM UP

1) Recode the `sexe` variable with Male and Female instead of 1 and 2

2) Filter out observations for which `annais` is XXXX and convert annais to numeric

3) Summarise your data into the total number of births per year

4) Plot the evolution of the number of births over time using a line geometry

Don't forget to load the necessary packages!

10:00

ANSWERS

```
library(dplyr)
library(ggplot2)
```

1) Recode the `sexe` variable with Male and Female instead of 1 and 2

```
names %>%
  mutate(sex = ifelse(sexe == 1, "Male", "Female"))
```

2) Filter out observations for which `annais` is XXXX and convert annais to numeric

```
names %>%
  mutate(sex = ifelse(sexe == 1, "Male", "Female")) %>%
  filter(annais != "XXXX") %>%
  mutate(annais = as.numeric(annais))
```

ANSWERS

3) Summarise your data into the total number of births per year

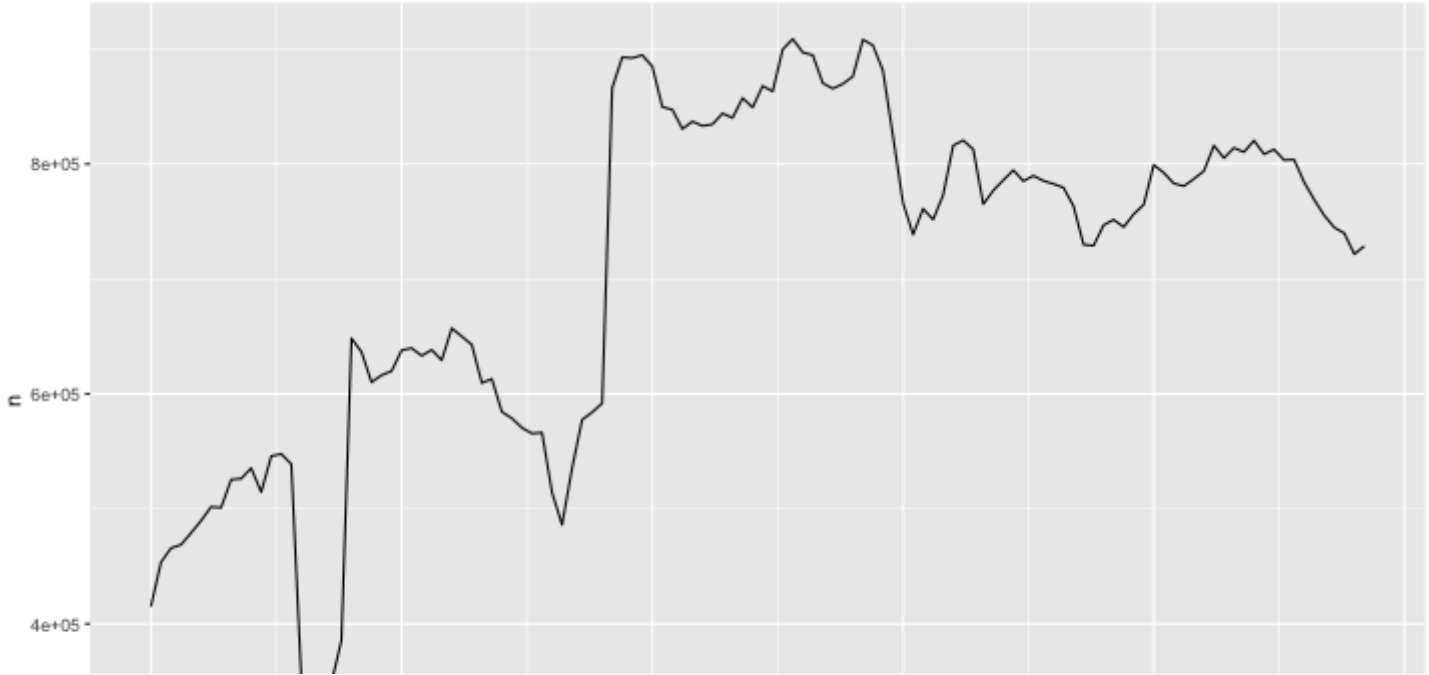
```
names %>%
  mutate(sex = ifelse(sexe == 1, "Male", "Female")) %>%
  filter(annais != "XXXX") %>%
  mutate(annais = as.numeric(annais)) %>%
  group_by(annais) %>%
  summarise(n = sum(nombre))
```

```
## # A tibble: 122 × 2
##    annais      n
##     <dbl>  <int>
##  1   1900 415040
##  2   1901 453456
##  3   1902 465791
##  4   1903 468810
##  5   1904 478962
##  6   1905 489697
##  7   1906 501745
##  8   1907 501025
##  9   1908 525490
## 10   1909 526233
## # ℹ 112 more rows
```

ANSWERS

4) Plot the evolution of the number of births over time using a line geometry

```
names %>%
  mutate(sex = ifelse(sexe == 1, "Male", "Female")) %>%
  filter(annais != "XXXX") %>%
  mutate(annais = as.numeric(annais)) %>%
  group_by(annais) %>%
  summarise(n = sum(nombre)) %>%
  ggplot(aes(x = annais, y  = n)) +
  geom_line()
```



AGENDA

- R MARKDOWN BASICS
- USEFUL FEATURES
- LATEX FOR EQUATIONS
- BEST CODING PRACTICES
- OTHER TOOLS
    1. GitHub
    2. Reference manager
    3. Coding library

R MARKDOWN BASICS

- A type of document (.Rmd) in which you can both **write/run R code** and edit text
- Produces **dynamic documents* that are generated by a script and updated automatically every time the script runs.
- Good for research transparency and simple documents that don't require a lot of format. No need of copying and pasting outputs into a document editor.
- You can use R Markdown to create different types of documents, presentations (like this one!), dashboards, and even books. See many examples in this gallery

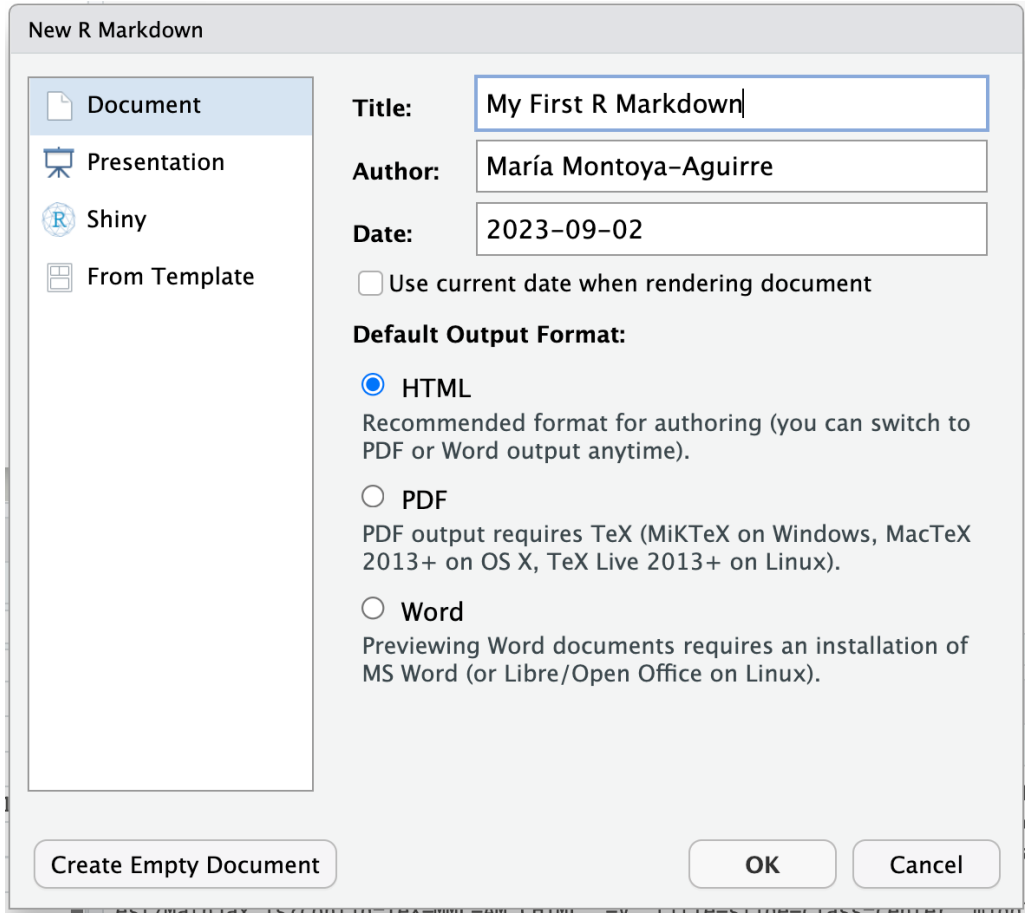
  --

- It is structured around 3 types of content:
  - **Code chunks** to run code and render its output
  - **Editable text** to display
  - **YAML metadata** to specify the process to build the R Markdown

LET'S START BY CREATING OUR FIRST R MARKDOWN

R MARKDOWN BASICS

Click on **File > New File > R Markdown ...**



1. Fill out the information and select HTML
2. Click **OK**

THE ANATOMY

It creates a template document with **3 types of content:**

````
---
title: "My First R Markdown"
author: "María Montoya-Aguirre"
date: "2023-09-02"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:
````

THE ANATOMY

It creates a template document with **3 types of content:**

YAML HEADER

```
---
title: "My First R Markdown"
author: "María Montoya-Aguirre"
date: "2023-09-02"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

Including Plots

You can also embed plots, for example:
```

THE ANATOMY

It creates a template document with **3 types of content:**

````
---
title: "My First R Markdown"
author: "María Montoya-Aguirre"
date: "2023-09-02"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.
For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any
embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

## Including Plots

You can also embed plots, for example:
````

CODE CHUNK 1

CODE CHUNK 2

THE ANATOMY

It creates a template document with **3 types of content:**

```
---
title: "My First R Markdown"
author: "María Montoya-Aguirre"
date: "2023-09-02"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

Including Plots

You can also embed plots, for example:
```

TEXT

YAML HEADER

- **YAML Ain't Markup Language -> It's configuration.** It gives instructions on how to *render* the document.

- Contains specific arguments:

    - Title (also subtitle)
    - Author
    - Date
    - Output type: `html_document`, `pdf_document`
    - We can also include other parameters that are referenced later in the document (more on this later)
- Specified at the **very beggining** of the document enclosed by **three dashes** `---`

TEXT/NARRATIVE

You can write paragraphs as you would do in a text editor.

- Text can be formatted using Markdown syntax (enough formatting to ensure readability but simple to use to focus on the content)
    - No clicking on buttons to create text formatting, instead we use **symbols** that need to be written along the text.
- Not all our narrative needs to be human-written. We'll see how our code can help generate parts of the text. For example, we can combine words into our list with the values of a variable, we can calculate a statistic directly in the text using inline R code, or write a bibliography.

SYNTAX

```
Plain text

End a line with two spaces for line break
*italics*
**bold**
***bold+italics***


# Header 1


## Header 2

[link](https://www.google.com)

> block quote

***
```

OUTPUT

Plain text

End a line with two spaces for line break
italics
bold
bold+italics

Header 1

Header 2

...
link

> block quote

```
* item 1
* item 2
  + sub-item
```

- item 1
- item 2
    - sub-item

```
1. ordered list
2. item 2    - sub-item
```

1. ordered list
2. item 2
    - sub-item

```
| Column 1  | Column 2 |
|-----------|----------|
| Row 1     | Row 2    |
```

Column 1	Column 2
Row 1	Row 2

```
| Column 1      | Column 2 |
|--------|----------|
| Row 1     |  Row 2    |
```

Column 1	Column 2
Row 1	Row 2

CODE CHUNKS

- The code in each chunk is run by R and its output is translated to the document. The document is kept in sync with the code! If the data we are working with, or the code we write changes, the document will too.

- A code chunk usually starts with ```` ```{} ```` and ends with ```` ``` ````. You can insert one by writing the fences or **chunk delimiters** or by using `Ctrl + Alt + i`

````
```{r}
x <- 5
x
```
````

- When the document renders, R will run the code, so both the code and the output will appear in the document:

```
x <- 5
x
```

```
## [1] 5
```

CODE CHUNKS

- You can customize the behavior and output of code chunks through chunk options (provided inside the curly brackets `{}`
- For example, to display only the output and not the code chunk, you set `echo = FALSE`

````
```{r, echo = F}
x <- 5
x
```
````

And the output will only be:

```
## [1] 5
```

CODE CHUNKS

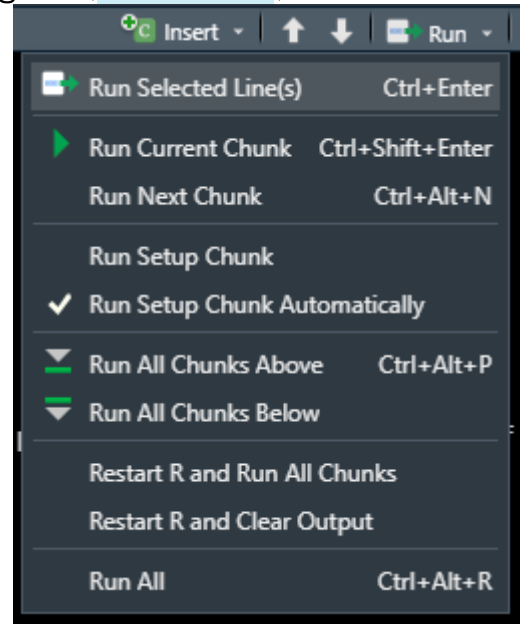
Option	Default	Effect
eval	TRUE	Evaluate the code and include its results?
echo	TRUE	Display the code and include its results?
warning	TRUE	Display warnings?
error	FALSE	Display errors?
message	TRUE	Display messages?
results	'markup'	How to render code output? 'hide' hides output
fig.width	7	Width in inches for plots created in the chunk
fig.height	7	Height in inches for plots created in the chiunk

RUN AND KNIT YOUR CODE

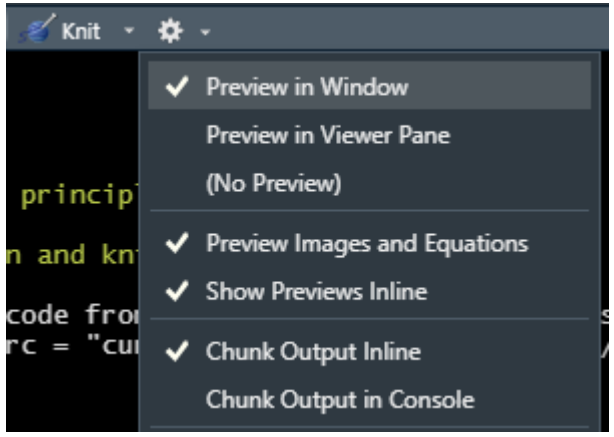
- To execute the content of a code chunk in R Markdown:
  - Click on the green play button at the top right of the chunk 

You can also:

- Run **all chunks above** the current chunk 
- Run **all chunks** from the Run dropdown menu at the top right (`Ctrl+Alt+R`)



- To choose where the output must be displayed while working on the .Rmd, click on the button



RUN AND KNIT YOUR CODE

To render an R Markdown file, click on the **knit button**

(`Ctrl + Shift + K`))

```
---
title: "My First R Markdown"
author: "Maria Montoya-Aguirre"
date: "04-28-1997"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```{r cars}
summary(cars)
```

Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```
```

My First R Markdown

Maria Montoya-Aguirre

04-28-1997

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

AGENDA

- R MARKDOWN BASICS
- USEFUL FEATURES
- LATEX FOR EQUATIONS
- BEST CODING PRACTICES
- OTHER TOOLS

    1. GitHub
    2. Reference manager
    3. Coding library

INLINE CODE

- A big advantage of R Markdown is that you can **automate** your reports
  - Imagine you figure out late in the process that you need to change something important that changes results! - With R Markdown you can just change the code to fix it and the tables and figures will **update automatically** But, what if you wrote some of the results within **paragraphs?**
- You might be able to update them automatically too!

Imagine there's a problem with the observations for which `dist > 100` and you need to remove them

````
---
title: "Report example"
author: "Maria Montoya-Aguirre"
date: "08-10-2023"
output: html_document
---

## Overview of the data
```{r}
# Omit if distance >= 100
cars <-
  cars %>%
  filter(dist < 100)

names(cars)
dim(cars)

mean(cars$speed)
mean(cars$dist)

```

The dataset contains two variables: speed and distance, and has 50 observations. The average speed value is 15.4 and the average distance value is 42.98.
````

Report example

Maria Montoya-Aguirre

08-10-2023

Overview of the data

```
# Omit if distance >= 100
cars <-
  cars %>%
  filter(dist < 100)

names(cars)
```

```
## [1] "speed" "dist"
```

```
dim(cars)
```

```
## [1] 49  2
```

```
mean(cars$speed)
```

```
## [1] 15.22449
```

```
mean(cars$dist)
```

```
## [1] 41.40816
```

All the results are updated but not the ones mentioned in the text. We can use **inline code** to make our paragraphs dynamic too.

Inline code allows to include the output of some **R code within text areas** of your report

- R code outside code chunks should be included between backticks
- Surroding text with backticks will change the **format** to that of the code chunk

- Adding the letter **r** after the first backtick will **show the output**

SYNTAX

```
`paste("a", "b", sep = "-")`
```

```
a-b
```

OUTPUT

`paste("a", "b", sep = "-")`

a-b

Setting the default chunk options

````
```{r}
knitr::opts_chunk$set(
  comment = "#>", echo = FALSE, fig.width = 6
```
````

```
---
title: "Report example"
author: "Maria Montoya-Aguirre"
date: "08-10-2023"
output: html_document
---
## Overview of the data
```{r}
# Omit if distance >= 100
cars <-
  cars %>%
  filter(dist < 100)

names(cars)
dim(cars)

mean(cars$speed)
mean(cars$dist)

```

The dataset contains two variables: speed and distance, and has `r dim(cars)[1]` observations. The average speed value is `r mean(cars$speed)` and the average distance value is `r mean(cars$dist)`.
```

Report example

Maria Montoya-Aguirre

08-10-2023

Overview of the data

```
# Omit if distance >= 100
cars <-
  cars %>%
  filter(dist < 100)

names(cars)
```

```
## [1] "speed" "dist"
```

```
dim(cars)
```

```
## [1] 49  2
```

```
mean(cars$speed)
```

```
## [1] 15.22449
```

```
mean(cars$dist)
```

```
## [1] 41.40816
```

TABLES

- Displaying a table as a raw output can be unpleasant to read

```
head(mtcars, 3)
```

```
##                mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4     21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710    22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

- Prettier tables are produced with the `kable()` function from the package `{knitr}`

```
library("knitr")

kable(head(mtcars, 3), caption = "First three rows of the dataset")
```

Table: First three rows of the dataset

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1

TABLES

For **big tables** you can use the `datatable()` function from the package `{DT}`

```
library("DT")

datatable(mtcars, options = list(pageLength = 5))
```

Show 5 entries Search:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2

Showing 1 to 5 of 32 entries Previous 1 2 3 4 5 6 7 Next

THEMES

- You can change the appearance of the document using a variaety of **preset themes**
- They need to be specified in the **YAML header** using a **theme argument**

```
---
title: "My First R Markdown"
author: "María Montoya-Aguirre"
date: "2023-08-01"
output:
  html_document:
    theme: cosmo
---
```

- See R Markdown theme gallery

PRACTICE

Reproduce the following html document using R markdown
Use `theme: cosmo`

Report on the first name MARIE

your name here

September 2023

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file

```
library(dplyr)
library(ggplot2)
```

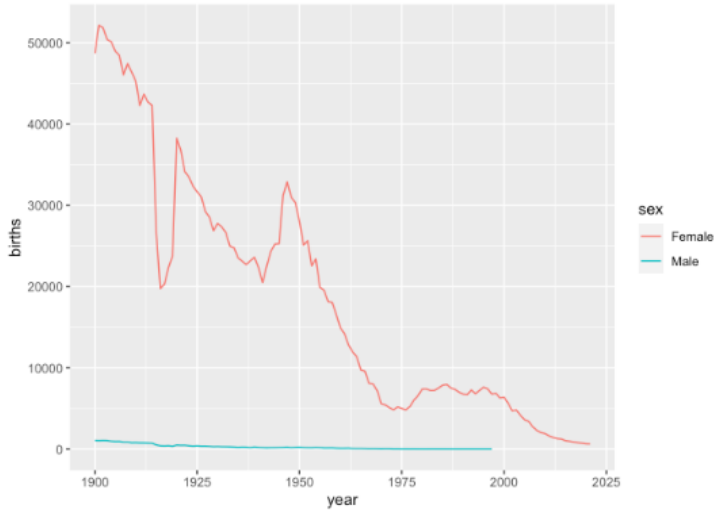
However, the command `install.packages()` must **not** be written in an R Markdown. It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("../data/03_fichier_prenoms.csv", sep = ";", encoding = "UTF-8") %>%
  mutate(sex = ifelse(sexe == 1, "Male", "Female")) %>%
  rename(year = annais, births = nombre, name = preusuel) %>%
  filter(year != "XXXX") %>%
  mutate(year = as.numeric(year))
```

3. Evolution of the first name MARIE over time

```
names %>%
  filter(name == "MARIE") %>%
  ggplot(aes(x = year, y = births,
             color = sex)) +
  geom_line()
```



641 children were born under the name MARIE in 2021. This statistic is written in **inline code** so it updates automatically.

USEFUL FEATURES

REPORT PARAMETERS

Sometimes it can be useful to produce separate reports for different groups in your data

- Country/state-specific reports
- Here, a different report for each first name

For this, we can use **YAML parameters**, which work similarly to an **object** in your environment

```
---
title: "Report on the first name `r params$name`"
author: "your name here"
date: "September 2023"
output:
  html_document:
    theme: cosmo
params:
  name: "MARIE"
---
```

REPORT PARAMETERS

````
### 3. Evolution of the first name `r params$name` over time

```{r}
  names %>%
    filter(name == params$name) %>%
    ggplot(aes(x = year, y = births,
               color = sex)) +
    geom_line()
```

`r sum(names[names$name== params$name & names$year == 2021, "births"])` children were born under the name `r params$name` in 2021. This statistic is written in **inline code** so it updates automatically.
````

Let's knit our .Rmd with a different name!

Report on the first name AHMED

your name here

September 2023

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file

```
library(dplyr)
library(ggplot2)
```

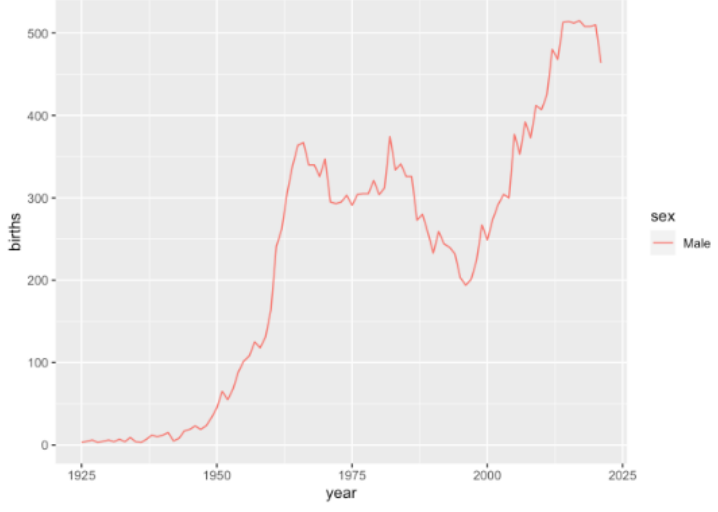
However, the command `install.packages()` must **not** be written in an R Markdown. It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("../data/03_fichier_prenoms.csv", sep = ";", encoding = "UTF-8") %>%
  mutate(sex = ifelse(sexe == 1, "Male", "Female")) %>%
  rename(year = annais, births = nombre, name = preusuel) %>%
  filter(year != "XXXX") %>%
  mutate(year = as.numeric(year))
```

3. Evolution of the first name AHMED over time

```
names %>%
  filter(name == params$name) %>%
  ggplot(aes(x = year, y = births,
             color = sex)) +
  geom_line()
```



464 children were born under the name AHMED in 2021. This statistic is written in **inline code** so it updates automatically.

Report on the first name CAMILLE

your name here

September 2023

1. Setup

The packages needed in an Rmd must *always* be loaded in a code chunk at the beginning of the file

```
library(dplyr)
library(ggplot2)
```

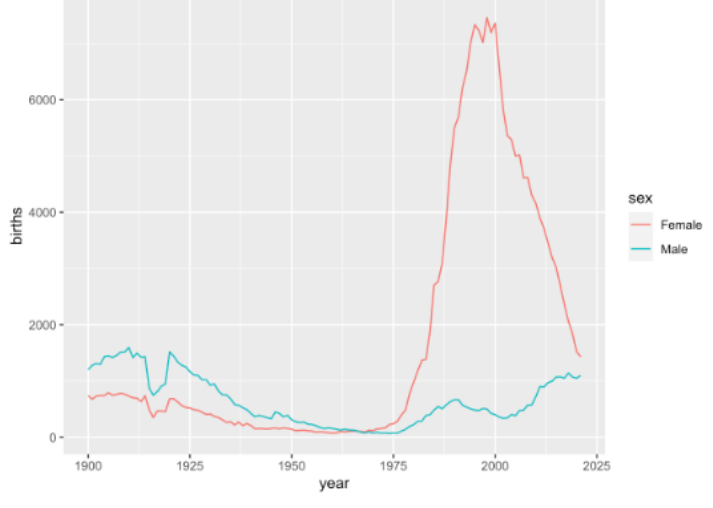
However, the command `install.packages()` must **not** be written in an R Markdown. It should be run only once in the console.

2. Data cleaning

```
names <- read.csv("../data/03_fichier_prenoms.csv", sep = ";", encoding = "UTF-8") %>%
  mutate(sex = ifelse(sexe == 1, "Male", "Female")) %>%
  rename(year = annais, births = nombre, name = preusuel) %>%
  filter(year != "XXXX") %>%
  mutate(year = as.numeric(year))
```

3. Evolution of the first name CAMILLE over time

```
names %>%
  filter(name == params$name) %>%
  ggplot(aes(x = year, y = births,
             color = sex)) +
  geom_line()
```



2524 children were born under the name CAMILLE in 2021. This statistic is written in **inline code** so it updates automatically.

USEFUL FEATURES

You can knit reports with different parameters on different documents on a separate script using the function `render()` `like this:

```
library(rmarkdown)
render(
  input = "../code/03_practice_report_params.Rmd",          # Specify the input .Rmd
  output_file = "C:/User/Documents/MARIE.html",             # Specify the output file
  params = list(name = "MARIE")                             # Specify the YAML parameter(s)
)
```

To avoid copy-pasting this command multiple times we can use a **loop**

1. First we should name the object that will successively take the value of each first name
2. Then indicate which values this object must successively take
3. Then indicate what to do at each iteration
4. And this should depend on the object that successively take each value

```
for (i in c("MARIE", "AHMED", "LOUIS", "CAMILLE")) {
  render(
    input = "../code/03_practice_report_params.Rmd",
    output_file = paste0("C:/User/Documents/", i, ".html"),
    params = list(name = i)
  )
}
```

LaTeX

- LaTeX is a document preparation system. It is similar to R Markdown in the sense that you:
    - Edit your text in a script using commands and symbols
    - Compile the script to get the output
- LaTeX is the preferred typesetting system for most academic fields mainly because:
    - Many things can be automated in LaTeX
    - It has a good way to typeset mathematical formulas

You can learn LaTeX in 30 minutes and easily use it on Overleaf (great for shared documents)

You can use it to write equations in R Markdown anywhere in the text, enclosed by `$$` or `$`

```
`$$\bar{x}=\frac{1}{N}\sum_{i=1}^Nx_i$$`
```

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

AGENDA

- R MARKDOWN BASICS
- USEFUL FEATURES
- LATEX FOR EQUATIONS
- BEST CODING PRACTICES
- OTHER TOOLS

BEST CODING* PRACTICES

Highlights from Reproducible Research Fundamentals course by DIME Analytics

Delicious meals are just as important as recipes!

1. Organize and name your files appropriately
2. Write the **full** recipe! (Write a master script and a README)
3. Write code that others (future you!) can read (comments, spacing & style)
4. Keep improving your skills (when your code works you're only half done)
5. Track your changes (Use a version control tool)

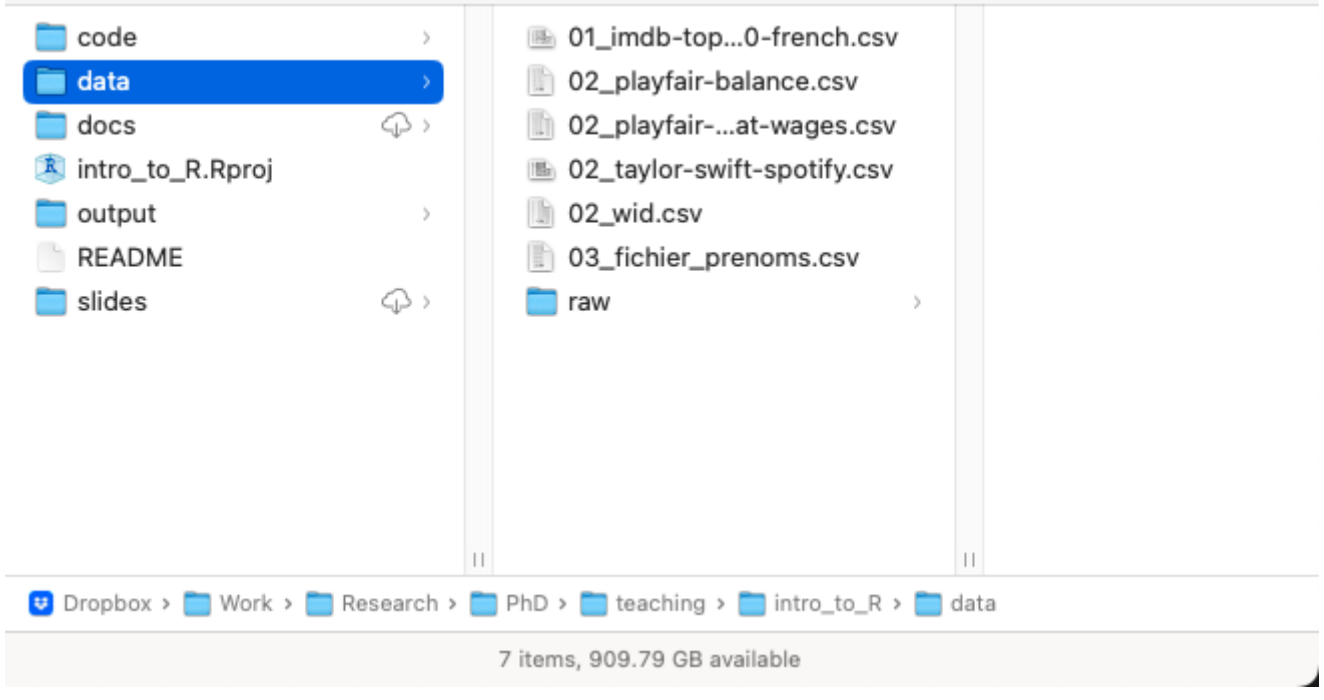
1. Organize and name your files appropriately!

Working directories

- The working directory is just a file path on your computer that sets the default location of any files you read into R, or save out of R. In other words, a working directory is like a little flag somewhere on your computer which is tied to a specific analysis project.
- If you ask R to import a dataset from a text file, or save a dataframe as a text file, it will assume that the file is inside of your working directory.
- To see your current working directory, use `getwd()` and `setwd()` to establish a different one
- When you are working with .Rmd files, the working directory is automatically set to the folder that contains the .Rmd file. For example, if the. file is: `~/Downloads/foo.Rmd`, under which R code chunks are evaluated is`~/Downloads/`.
    - This is useful! This also happens when you work with R Projects (File > New File > New Project). They are very useful for dividing work into multiple contexts and calling all you need at once

1. Organize and name your files appropriately!

Naming things



- Separate code for cleaning and analysis
- Naming your files (machine readable, human readable and fit for ordering). Use regular expressions.
  - `2023-08_WBLCHILDCARE_abstract.doc`
  - `2023-08_WBLCHILDCARE_submission_AEJ.doc`
  - `2023-10_WBLCHILDCARE_submission_QJE.doc`

2. Write the full recipe!

The master script Write down all the instructions from the raw data to the final report

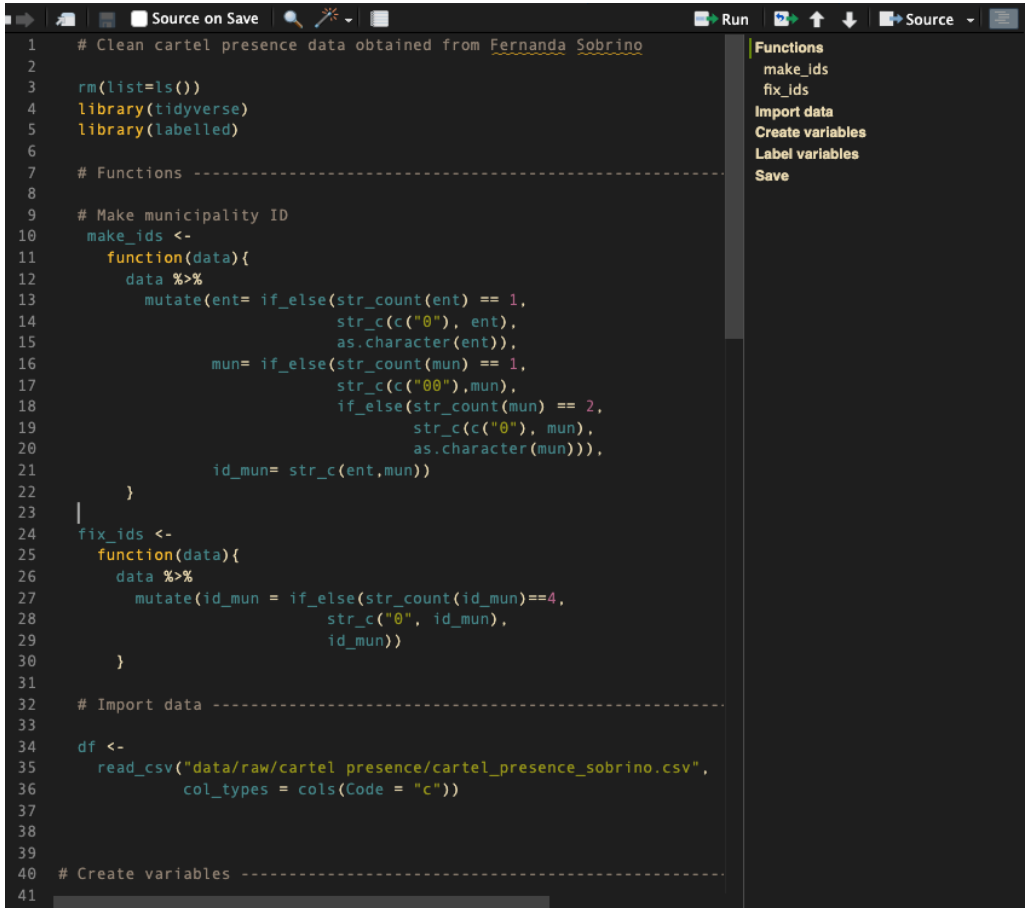
- It compactly and reproducibly runs all the scripts needed for the project Code starts to pile up quickly. Keep your scripts short and focused (cleaning each dataset, merging)
- It establishes an identical workspace between users by specifying settings, installing packages, and setting parameters
- It maps the files within the data folder and serves as a starting point to find data, scripts or outputs

Structure:

1. Intro header with main information about the project
2. Specify required packages to install and run user-created programs
3. Specify settings like the R version used, default theme for graphs and tables, etc.
4. Define units and assumptions. For example, conversion rates and units used, list of control variables to be used
5. Run scripts

```
1
2
3  # Employment data ----------------------------------------------------------
4
5  # Get individual indicators of employment and aggregate at the municipal-quarter
6  # level by combination of gender, age, and schooling
7  #   Input:  data/raw/ENOE
8  #   Output: data/employment_quarter_municipal_2005-2009_gender.csv
9  #           data/employment_quarter_municipal_2005-2009_age.csv
10 #           data/employment_quarter_municipal_2005-2009_schooling.csv
11 #           data/employment_quarter_municipal_2005-2009_gender_age_schooling.csv
12 #           data/nonresponse_quarter_municipal_2005-2009.csv
13 #   ID:     id_mun year trim group_gender group_age group_schooling
14   source("./code/clean/employment/0_clean_enoe_quarter.R")
15
16 # Get municipal-year estimates by group (gender, age, schooling) &
17 # municipality-year non response rates
18 #   Input: data/employment_quarter_municipal_2005-2009_gender.csv
19 #          data/employment_quarter_municipal_2005-2009_age.csv
20 #          data/employment_quarter_municipal_2005-2009_schooling.csv
21 #          data/employment_quarter_municipal_2005-2009_gender_age_schooling.csv
22 #          data/nonresponse_quarter_municipal_2005-2009.csv
23 #   Output: data/employment_year_municipal_2005-2009_gender.csv
24 #           data/employment_year_municipal_2005-2009_age.csv
25 #           data/employment_year_municipal_2005-2009_schooling.csv
26 #           data/employment_year_municipal_2005-2009_gender_age_schooling.csv
27 #           data/nonresponse_year_municipal_2005-2009.csv
28 #   ID:     id_mun year group_gender group_age group_schooling (employment)
29 #           id_mun year (nonresponse)
30   source("./code/clean/employment/1_get_enoe_yearly.R")
31
32 # Get MSA-year employment and nonresponse estimates by group
33 #   Input: data/employment_year_municipal_2005-2009_gender.csv
34 #          data/employment_year_municipal_2005-2009_age.csv
35 #          data/employment_year_municipal_2005-2009_schooling.csv
36 #          data/employment_year_municipal_2005-2009_gender_age_schooling.csv
```

3. Write code that others (future you!) can read



```
# Clean cartel presence data obtained from Fernanda Sobrino

rm(list=ls())
library(tidyverse)
library(labelled)

# Functions ---------------------------------------------------------

# Make municipality ID
 make_ids <-
   function(data){
     data %>%
       mutate(ent= if_else(str_count(ent) == 1,
                           str_c(c("0"), ent),
                           as.character(ent)),
              mun= if_else(str_count(mun) == 1,
                           str_c(c("00"),mun),
                           if_else(str_count(mun) == 2,
                                   str_c(c("0"), mun),
                                   as.character(mun))),
              id_mun= str_c(ent,mun))
     }

fix_ids <-
  function(data){
    data %>%
      mutate(id_mun = if_else(str_count(id_mun)==4,
                          str_c("0", id_mun),
                          id_mun))
    }

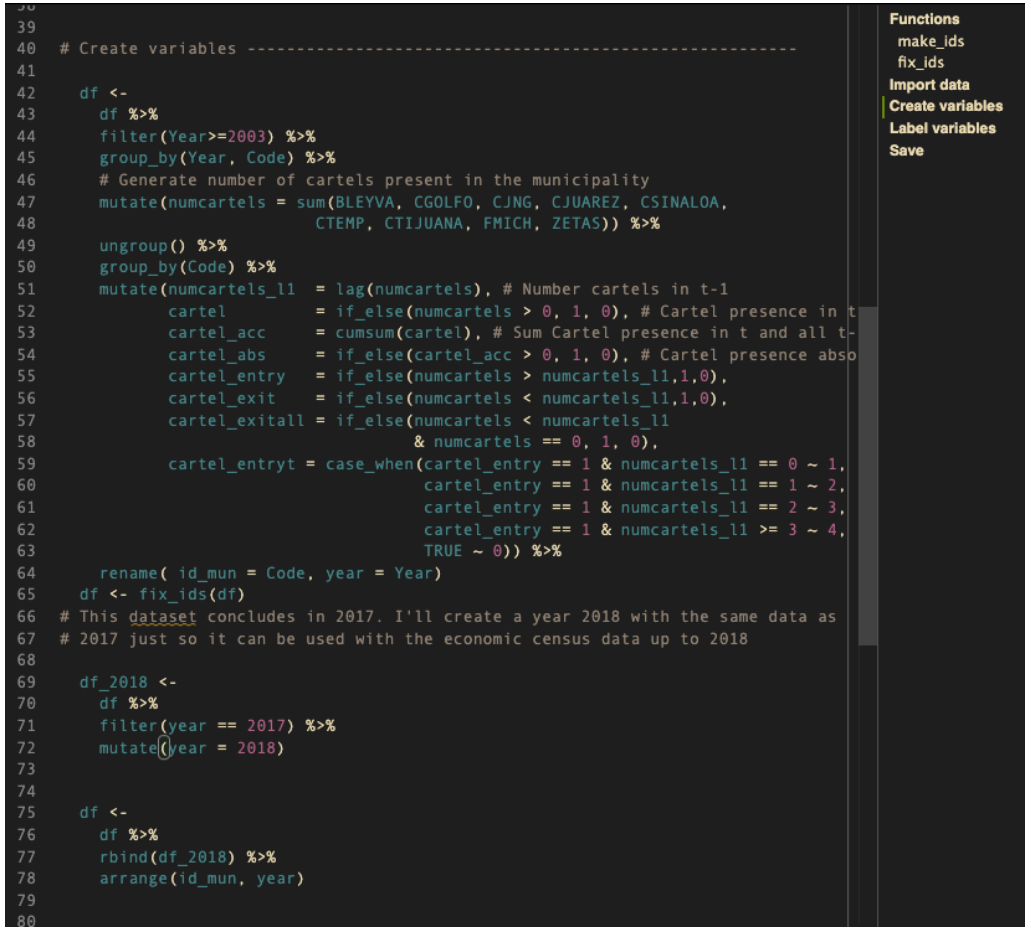
# Import data -------------------------------------------------------

df <-
  read_csv("data/raw/cartel presence/cartel_presence_sobrino.csv",
           col_types = cols(Code = "c"))



# Create variables --------------------------------------------------
```

Functions
make_ids
fix_ids
Import data
Create variables
Label variables
Save



```
# Create variables --------------------------------------------------------

  df <-
    df %>%
    filter(Year>=2003) %>%
    group_by(Year, Code) %>%
    # Generate number of cartels present in the municipality
    mutate(numcartels = sum(BLEYVA, CGOLFO, CJNG, CJUAREZ, CSINALOA,
                            CTEMP, CTIJUANA, FMICH, ZETAS)) %>%
    ungroup() %>%
    group_by(Code) %>%
    mutate(numcartels_l1  = lag(numcartels), # Number cartels in t-1
           cartel         = if_else(numcartels > 0, 1, 0), # Cartel presence in t
           cartel_acc     = cumsum(cartel), # Sum Cartel presence in t and all t-
           cartel_abs     = if_else(cartel_acc > 0, 1, 0), # Cartel presence abso
           cartel_entry   = if_else(numcartels > numcartels_l1,1,0),
           cartel_exit    = if_else(numcartels < numcartels_l1,1,0),
           cartel_exitall = if_else(numcartels < numcartels_l1
                                    & numcartels == 0, 1, 0),
           cartel_entryt = case_when(cartel_entry == 1 & numcartels_l1 == 0 ~ 1,
                                     cartel_entry == 1 & numcartels_l1 == 1 ~ 2,
                                     cartel_entry == 1 & numcartels_l1 == 2 ~ 3,
                                     cartel_entry == 1 & numcartels_l1 >= 3 ~ 4,
                                     TRUE ~ 0)) %>%
    rename( id_mun = Code, year = Year)
  df <- fix_ids(df)
# This dataset concludes in 2017. I'll create a year 2018 with the same data as
# 2017 just so it can be used with the economic census data up to 2018

  df_2018 <-
    df %>%
    filter(year == 2017) %>%
    mutate(year = 2018)


  df <-
    df %>%
    rbind(df_2018) %>%
    arrange(id_mun, year)
```

Functions
make_ids
fix_ids
Import data
Create variables
Label variables
Save

- White space makes everything better.
- Using `# Headers ---` and indenting creates a useful outline
- Follow The tidyverse style guide. Lines < 80 characters.

4. Keep improving your skills

When your code works you're only half done

```
data %>%
  mutate(
    province = str_to_title(province),
    region = str_to_title(region),
    commune = str_to_title(commune),
    district = str_to_title(district),
    name = str_to_title(name),
  )
```

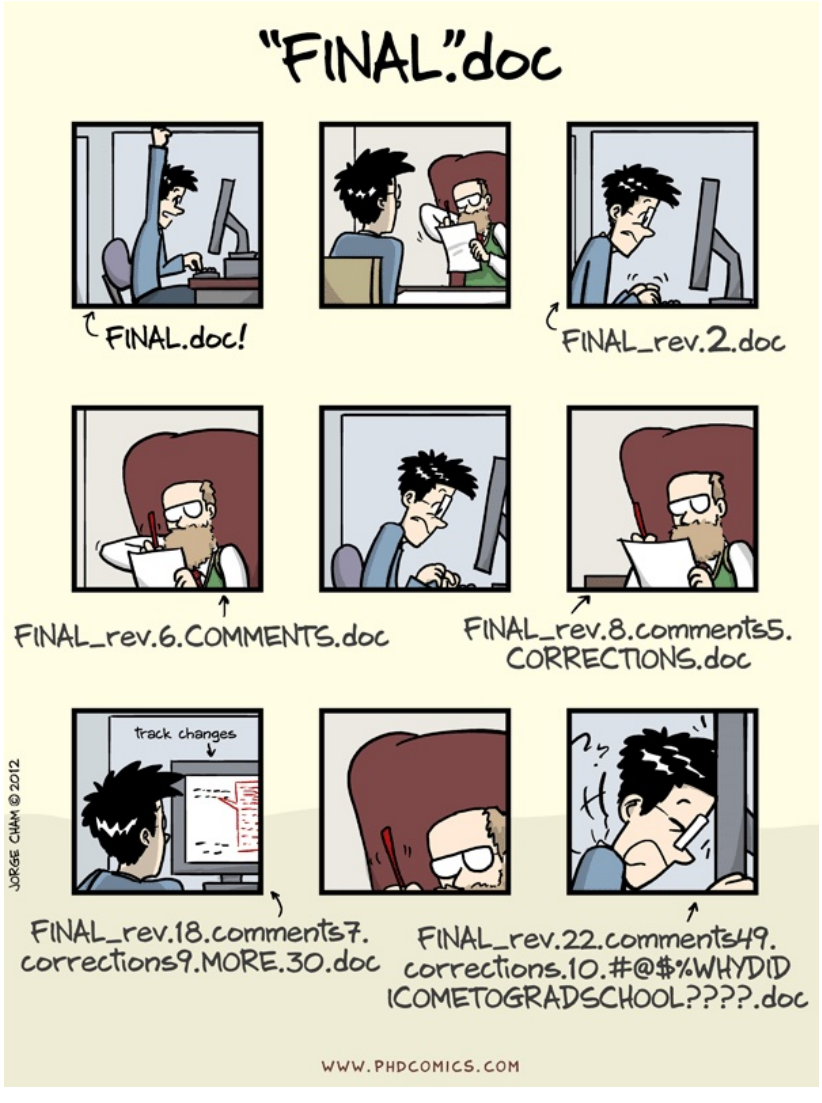
```
data %>%
  mutate(
    across(
      c(province, region, commune, district, name),
      ~str_to_title(.)
    )
  )
```

- Don't repeat yourself
- Don't write code that is too long (you're doing too many things at once)
- Defense programming: write tests to check your data.
- No magic constants: `data[x < 58.7,]` vs `data[x < mean(median(data$x))]`

5. Keep track of your work

- Your code will constantly change, but when you us a version control tool like Git/GitHub you will always have access to previous versions of your code
- Using file naming conventions (such adding dates and initials as suffixes) is better than no version control, but it can get out of control very quickly
- Syncing software (such as OneDrive and Dropbox) allow teams to revert to old version of a document, but not to track specific changes
- *git* is currently the best version control system out there as one can track changes and revert to old versions easily

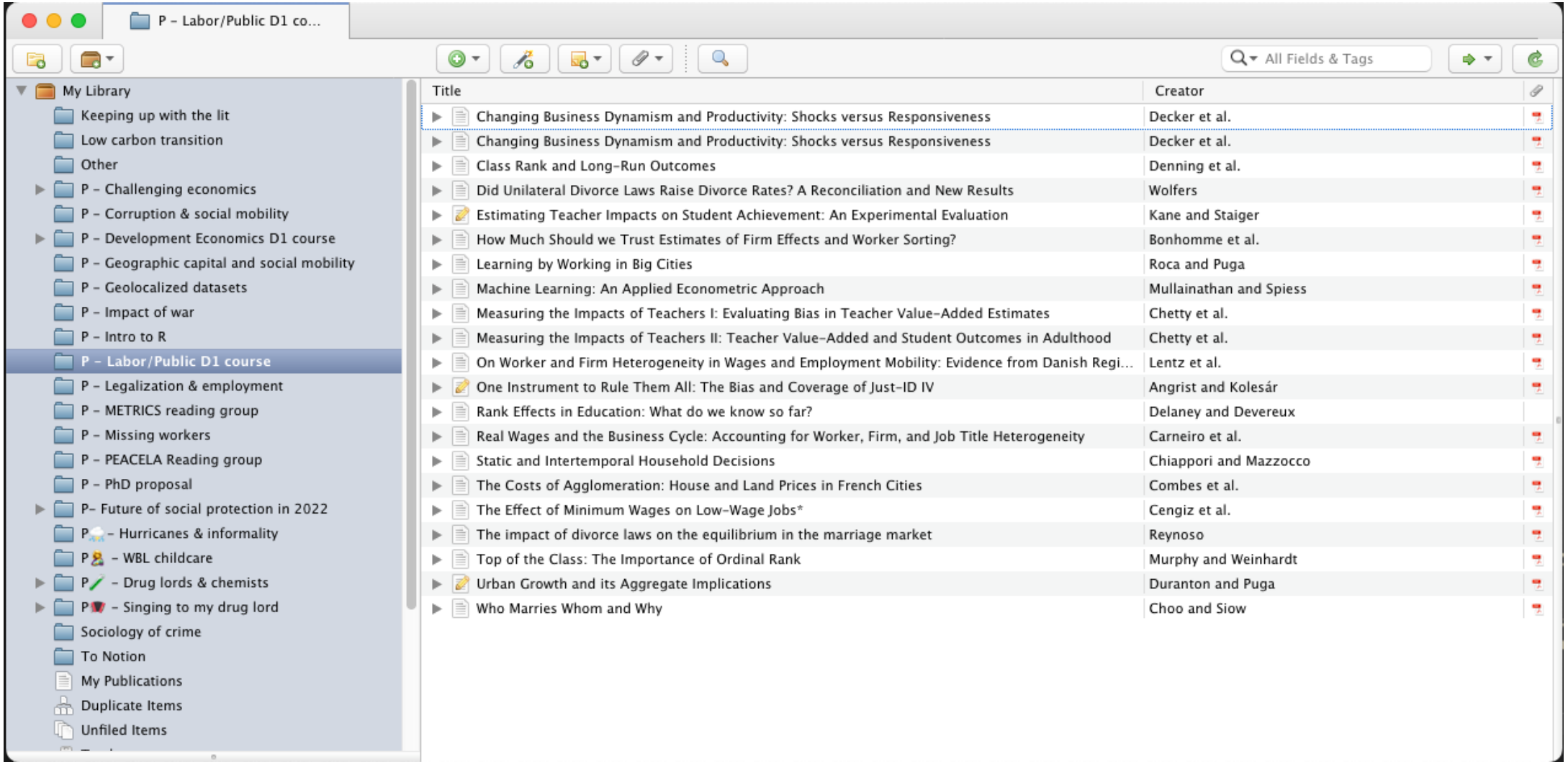
Learn how to use GitHub with DIME Analytics GitHub trainings. Start here



OTHER TOOLS

Reference manager

- Applications like Zotero or Mendeley allow you to collect, organize, and export bibliographic references or citations. You can store the actual PDFs there, annotate them and associate notes to them.
- Easily send a paper or book to your Zotero/Mendeley library with a browser extension.
- You can create a bibliography with two clicks (export a BibTex file) and add it to your report on R Markdown.


P – Labor/Public D1 co...
All Fields & Tags
My Library
Keeping up with the lit
Low carbon transition
Other
P – Challenging economics
P – Corruption & social mobility
P – Development Economics D1 course
P – Geographic capital and social mobility
P – Geolocalized datasets
P – Impact of war
P – Intro to R
P – Labor/Public D1 course
P – Legalization & employment
P – METRICS reading group
P – Missing workers
P – PEACELA Reading group
P – PhD proposal
P– Future of social protection in 2022
P – Hurricanes & informality
P – WBL childcare
P – Drug lords & chemists
P – Singing to my drug lord
Sociology of crime
To Notion
My Publications
Duplicate Items
Unfiled Items
Title
Creator
Changing Business Dynamism and Productivity: Shocks versus Responsiveness
Decker et al.
Changing Business Dynamism and Productivity: Shocks versus Responsiveness
Decker et al.
Class Rank and Long-Run Outcomes
Denning et al.
Did Unilateral Divorce Laws Raise Divorce Rates? A Reconciliation and New Results
Wolfers
Estimating Teacher Impacts on Student Achievement: An Experimental Evaluation
Kane and Staiger
How Much Should we Trust Estimates of Firm Effects and Worker Sorting?
Bonhomme et al.
Learning by Working in Big Cities
Roca and Puga
Machine Learning: An Applied Econometric Approach
Mullainathan and Spiess
Measuring the Impacts of Teachers I: Evaluating Bias in Teacher Value-Added Estimates
Chetty et al.
Measuring the Impacts of Teachers II: Teacher Value-Added and Student Outcomes in Adulthood
Chetty et al.
On Worker and Firm Heterogeneity in Wages and Employment Mobility: Evidence from Danish Regi...
Lentz et al.
One Instrument to Rule Them All: The Bias and Coverage of Just-ID IV
Angrist and Kolesár
Rank Effects in Education: What do we know so far?
Delaney and Devereux
Real Wages and the Business Cycle: Accounting for Worker, Firm, and Job Title Heterogeneity
Carneiro et al.
Static and Intertemporal Household Decisions
Chiappori and Mazzocco
The Costs of Agglomeration: House and Land Prices in French Cities
Combes et al.
The Effect of Minimum Wages on Low-Wage Jobs*
Cengiz et al.
The impact of divorce laws on the equilibrium in the marriage market
Reynoso
Top of the Class: The Importance of Ordinal Rank
Murphy and Weinhardt
Urban Growth and its Aggregate Implications
Duranton and Puga
Who Marries Whom and Why
Choo and Siow

OTHER TOOLS

Reference manager

- Applications like Zotero or Mendeley allow you to collect, organize, and export bibliographic references or citations. You can store the actual PDFs there, annotate them and associate notes to them.
- Easily send a paper or book to your Zotero/Mendeley library with a browser extension.
- You can create a bibliography with two clicks (export a BibTex file) and add it to your report on R Markdown.

Create your code library

- Whenever you learn to do something new in R, write a quick note about it and add a reusable snippet of code
- When working with certain data, you'll find yourself doing the same tasks over and over again. You don't want to write your code from scratch every time.
  - For example, I always have to create geographic IDs from state and municipality variables. I have a snippet in my code library (basically a function) that I just call to do this.

Resources

- Tidyverse style guide https://style.tidyverse.org/
- Checklist on regression output https://blogs.worldbank.org/impactevaluations/crowd-sourced-checklist-top-10-little-things-drive-us-crazy-regression-output
- Intro to R Markdown by DIME https://raw.githack.com/worldbank/dime-r-training/main/Presentations/07-r-markdown.html#1
- Spatial data https://raw.githack.com/worldbank/dime-r-training/main/Presentations/06-spatial-data.html#1
- Blog posts by David McKenzie https://blogs.worldbank.org/impactevaluations/curated-list-our-postings-technical-topics-your-one-stop-shop-methodology-0
  - DIME coding standards https://github.com/worldbank/dime-standards/tree/master/dime-coding-standards/checklists
  - DIME research standards https://github.com/worldbank/dime-standards/#dime-coding-standards
  - Writing reports in R markdown https://book.rwithoutstatistics.com/rmarkdown-chapter https://worldbank.github.io/dime-data-handbook/