Quant II

Lab 1: Introduction: R Markdown, reproducibility, DAGs

Giacomo Lemoli

January 26, 2023

Ciao!

- Giacomo Lemoli, 5^{th} year PhD.

Ciao!

- Giacomo Lemoli, 5^{th} year PhD.
- I do research on CP and (H)PE

Ciao!

- Giacomo Lemoli, 5^{th} year PhD.
- I do research on CP and (H)PE
- I like: (i) politics of identity formation and language, (ii) statistics and causal inference

Ciao!

- Giacomo Lemoli, 5^{th} year PhD.
- I do research on CP and (H)PE
- I like: (i) politics of identity formation and language, (ii) statistics and causal inference
- Email: gl1759@nyu.edu

Ciao!

- Giacomo Lemoli, 5^{th} year PhD.
- I do research on CP and (H)PE
- I like: (i) politics of identity formation and language, (ii) statistics and causal inference
- Email: gl1759@nyu.edu
- Office: 404

Logistics (1): recitations

- Lab: Thursdays, 4pm - 6pm EST, Room 217

Logistics (1): recitations

- Lab: Thursdays, 4pm - 6pm EST, Room 217
- Lab materials will be posted on the lab's GitHub repo: https://github.com/giacomolemoli/quant2-labs-spring-2023

Logistics (1): recitations

- Lab: Thursdays, 4pm - 6pm EST, Room 217
- Lab materials will be posted on the lab's GitHub repo: https://github.com/giacomolemoli/quant2-labs-spring-2023
- For class material refer to Course web page

Logistics (1): recitations

- Lab: Thursdays, 4pm - 6pm EST, Room 217
- Lab materials will be posted on the lab's GitHub repo: https://github.com/giacomolemoli/quant2-labs-spring-2023
- For class material refer to Course web page
- Office hours: by appointment

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**
- Submit **PDF document** + **Code used**

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**
- Submit **PDF document** + **Code used**
  - Markdown recommended (more below)

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**
- Submit **PDF document** + **Code used**
  - Markdown recommended (more below)
  - Alternatively, PDF and separate R/Stata code

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**
- Submit **PDF document** + **Code used**
  - Markdown recommended (more below)
  - Alternatively, PDF and separate R/Stata code
- Code should be well commented (more below)

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**
- Submit **PDF document** + **Code used**
  - Markdown recommended (more below)
  - Alternatively, PDF and separate R/Stata code
- Code should be well commented (more below)
- Tables and plots format should be of high quality, irrespective of the editor used

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**
- Submit **PDF document** + **Code used**
  - Markdown recommended (more below)
  - Alternatively, PDF and separate R/Stata code
- Code should be well commented (more below)
- Tables and plots format should be of high quality, irrespective of the editor used
- Substantive answers should be presented in the written paragraphs

Logistics (2): homeworks

- Homework due via email to Cyrus and me by the indicated deadline
- Deadline is **strict**
- Submit **PDF document** + **Code used**
  - Markdown recommended (more below)
  - Alternatively, PDF and separate R/Stata code
- Code should be well commented (more below)
- Tables and plots format should be of high quality, irrespective of the editor used
- Substantive answers should be presented in the written paragraphs
- Ultimately, the goal is to learn how to **do** and **communicate** empirical research

Labs

- Strong applied focus

Labs

- Strong applied focus
- Clarification on important concepts: examples, occasional additional derivations, simulations

Labs

- Strong applied focus
- Clarification on important concepts: examples, occasional additional derivations, simulations
- Extensions

Labs

- Strong applied focus
- Clarification on important concepts: examples, occasional additional derivations, simulations
- Extensions
- Applications: examples and replications from published papers' data

Labs

- Strong applied focus
- Clarification on important concepts: examples, occasional additional derivations, simulations
- Extensions
- Applications: examples and replications from published papers' data
  - Code samples and packages useful for homeworks

Labs

- Strong applied focus
- Clarification on important concepts: examples, occasional additional derivations, simulations
- Extensions
- Applications: examples and replications from published papers' data
  - Code samples and packages useful for homeworks
  - Bridge theory (class) and practice (research papers)

Labs

- Strong applied focus
- Clarification on important concepts: examples, occasional additional derivations, simulations
- Extensions
- Applications: examples and replications from published papers' data
  - Code samples and packages useful for homeworks
  - Bridge theory (class) and practice (research papers)
- Software: R (primary) and Stata (secondary)

Labs

- Strong applied focus
- Clarification on important concepts: examples, occasional additional derivations, simulations
- Extensions
- Applications: examples and replications from published papers' data
  - Code samples and packages useful for homeworks
  - Bridge theory (class) and practice (research papers)
- Software: R (primary) and Stata (secondary)
  - Focus on R, but Stata may be useful for homeworks, so resources will be available in both languages

Today's plan

- R Markdown
- Doing reproducible work
- DAGs: conditioning, collider bias

Install R

- For MAC, go to http://cran.r-project.org/bin/macosx/

Install R

- For MAC, go to http://cran.r-project.org/bin/macosx/
- For Windows, go to http://cran.r-project.org/bin/windows/base/

Install R

- For MAC, go to http://cran.r-project.org/bin/macosx/
- For Windows, go to http://cran.r-project.org/bin/windows/base/
- You should choose an editor and learn how it works

Install R

- For MAC, go to http://cran.r-project.org/bin/macosx/
- For Windows, go to http://cran.r-project.org/bin/windows/base/
- You should choose an editor and learn how it works
  - One common choice is RStudio, but there is vim, emacs, Notepad++, etc.

Install R

- For MAC, go to http://cran.r-project.org/bin/macosx/
- For Windows, go to http://cran.r-project.org/bin/windows/base/
- You should choose an editor and learn how it works
  - One common choice is RStudio, but there is vim, emacs, Notepad++, etc.
- If you are Python/Data science people, you can also download and run RStudio from Anaconda

When things break

- Documentation - R: `?lm`. Stata: `help regress`

When things break

- Documentation - R: `?lm`. Stata: `help regress`
- Google

When things break

- Documentation - R: `?lm`. Stata: `help regress`
- Google
- CRAN (Reference manuals, vignettes, etc) - E.g.: http://cran.r-project.org/web/packages/AER/index.html

When things break

- Documentation - R: `?lm`. Stata: `help regress`
- Google
- CRAN (Reference manuals, vignettes, etc) - E.g.: http://cran.r-project.org/web/packages/AER/index.html
- JSS - E.g.: http://www.jstatsoft.org/v27/i02

When things break

- Documentation - R: `?lm`. Stata: `help regress`
- Google
- CRAN (Reference manuals, vignettes, etc) - E.g.: http://cran.r-project.org/web/packages/AER/index.html
- JSS - E.g.: http://www.jstatsoft.org/v27/i02
- Stack Overflow - http://stackoverflow.com/questions/tagged/r

When things break

- Documentation - R: `?lm`. Stata: `help regress`
- Google
- CRAN (Reference manuals, vignettes, etc) - E.g.: http://cran.r-project.org/web/packages/AER/index.html
- JSS - E.g.: http://www.jstatsoft.org/v27/i02
- Stack Overflow - http://stackoverflow.com/questions/tagged/r
- Listservs - http://www.r-project.org/mail.html

When things break

- Documentation - R: `?lm`. Stata: `help regress`
- Google
- CRAN (Reference manuals, vignettes, etc) - E.g.: http://cran.r-project.org/web/packages/AER/index.html
- JSS - E.g.: http://www.jstatsoft.org/v27/i02
- Stack Overflow - http://stackoverflow.com/questions/tagged/r
- Listservs - http://www.r-project.org/mail.html
- Stata - https://www.statalist.org/

Resources

- The Art of R Programming - N. Matloff

Resources

- The Art of R Programming - N. Matloff
- Modern Applied Statistics with S - W. Venables and B. Ripley

Resources

- The Art of R Programming - N. Matloff
- Modern Applied Statistics with S - W. Venables and B. Ripley
- Advanced R Programming - H. Wickham

Resources

- The Art of R Programming - N. Matloff
- Modern Applied Statistics with S - W. Venables and B. Ripley
- Advanced R Programming - H. Wickham
- The R Inferno - P. Burns

Resources

- The Art of R Programming - N. Matloff
- Modern Applied Statistics with S - W. Venables and B. Ripley
- Advanced R Programming - H. Wickham
- The R Inferno - P. Burns
- Rdataviz - a talk by P. Barberá on ggplot2

Resources

- The Art of R Programming - N. Matloff
- Modern Applied Statistics with S - W. Venables and B. Ripley
- Advanced R Programming - H. Wickham
- The R Inferno - P. Burns
- Rdataviz - a talk by P. Barberá on ggplot2
- Basic Intro to R - also by P. Barberá

Resources

- The Art of R Programming - N. Matloff
- Modern Applied Statistics with S - W. Venables and B. Ripley
- Advanced R Programming - H. Wickham
- The R Inferno - P. Burns
- Rdataviz - a talk by P. Barberá on ggplot2
- Basic Intro to R - also by P. Barberá
- Jamie Monogan

Resources

- Harvard Dataverse and other replication archives for scientific work

Resources

- Harvard Dataverse and other replication archives for scientific work
- Reproduction/Replication is the best way to learn how to work with data and to code

Resources

- Harvard Dataverse and other replication archives for scientific work
- Reproduction/Replication is the best way to learn how to work with data and to code
- If you are interested in using a technique, you should always look at the code and data of papers who use it

Resources

- Harvard Dataverse and other replication archives for scientific work
- Reproduction/Replication is the best way to learn how to work with data and to code
- If you are interested in using a technique, you should always look at the code and data of papers who use it
- A search engine for replication data: https://datasetsearch.research.google.com/

What is R Markdown?

- A tool within RStudio that allows you to write documents, presentations, or webpages, and combine written text with code

What is R Markdown?

- A tool within RStudio that allows you to write documents, presentations, or webpages, and combine written text with code
- The text in the document can be fully formatted

What is R Markdown?

- A tool within RStudio that allows you to write documents, presentations, or webpages, and combine written text with code
- The text in the document can be fully formatted
- You can choose whether to make your code visible or not

What is R Markdown?

- A tool within RStudio that allows you to write documents, presentations, or webpages, and combine written text with code
- The text in the document can be fully formatted
- You can choose whether to make your code visible or not
- Output documents can be PDFs, Word Documents, HTML pages, and other formats

Why R Markdown?

- It has all the advantages of LaTeX for writing and formatting scientific documents

Why R Markdown?

- It has all the advantages of LaTeX for writing and formatting scientific documents
- Directly embed code, code output, and text comments in one document

Why R Markdown?

- It has all the advantages of LaTeX for writing and formatting scientific documents
- Directly embed code, code output, and text comments in one document
    - All workflow (data analysis, graphs, tables, text) is in one place

Why R Markdown?

- It has all the advantages of LaTeX for writing and formatting scientific documents
- Directly embed code, code output, and text comments in one document
  - All workflow (data analysis, graphs, tables, text) is in one place
- Work is immediately reproducible and verifiable

Why R Markdown?

- It has all the advantages of LaTeX for writing and formatting scientific documents
- Directly embed code, code output, and text comments in one document
  - All workflow (data analysis, graphs, tables, text) is in one place
- Work is immediately reproducible and verifiable
- Besides Quant 2 homeworks, I recommend to use it for all class projects that involve replications or data analyses

What if I prefer Stata?

- You can still use Markdown via Stata Markdown

What if I prefer Stata?

- You can still use Markdown via Stata Markdown
  - Similar to R Markdown, you can write from your Stata editor

What if I prefer Stata?

- You can still use Markdown via Stata Markdown
  - Similar to R Markdown, you can write from your Stata editor
- Up-to-date Stata software is available on NYU desk machines

Getting started with R Markdown

- You need to download R and RStudio. For recent versions, that's enough

Getting started with R Markdown

- You need to download R and RStudio. For recent versions, that's enough
- You may need to install some packages the first time (`knitr`,`markdown`)

Getting started with R Markdown

- You need to download R and RStudio. For recent versions, that's enough
- You may need to install some packages the first time (`knitr`,`markdown`)
- If you want to generate PDF output, you also need a LaTex system (e.g. MiKTeX)

Getting started with R Markdown

Once you have done all that,

1. Open RStudio
2. Select File > New File > R Markdown
3. Enter Title, Author, Output Format (which can easily be changed later)
4. You are good to go! You will see that there is some example text that you can delete.

Markdown document

The documents typically have four different pieces: the **YAML**, the **formatted text**, **code chunks**, and **inline code**.

YAML

- At the beginning of any R Markdown script is a YAML header section enclosed by —.

YAML

- At the beginning of any R Markdown script is a YAML header section enclosed by —.
- Includes title, author, date, and type of document you want to produce -`beamer_presentation`, `pdf_document`, `html_document`, etc

YAML

- At the beginning of any R Markdown script is a YAML header section enclosed by —.
- Includes title, author, date, and type of document you want to produce -`beamer_presentation`, `pdf_document`, `html_document`, etc
  - An overview of the available output formats is here

YAML

- At the beginning of any R Markdown script is a YAML header section enclosed by —.
- Includes title, author, date, and type of document you want to produce -`beamer_presentation`, `pdf_document`, `html_document`, etc
  - An overview of the available output formats is here
- **NB**: rules in the header section will alter the whole document

YAML

You will always insert something like this at the top of your new .Rmd script:

```
---
title: "Quant II"
subtitle: "Lab 1: Introduction: R Markdown, reproducibility, I
author: "Giacomo Lemoli"
date: "January 26, 2023"
---
```

Code chunks

- The real payoff of R Markdown
- The code you enter is executed and the results are displayed in the document
- Syntax for code chunks:

````
```{r}
R code here
```
````

Code chunks

- Note that “‘ are backticks, not quotes or apostrophes. Everything that goes between these backticks should have R syntax

```
set.seed(1)

x <- runif(10)
y <- rnorm(10, x, 1)
df <- data.frame(x, y)
df
```

```
##             x           y
## 1  0.26550866 -0.55495972
## 2  0.37212390  0.85955295
## 3  0.57285336  1.31117807
## 4  0.90820779  1.48398914
## 5  0.20168193 -0.10370646
## 6  0.89838968  2.41017085
## 7  0.94467527  1.33451851
## 8  0.66079779  0.03955721
## 9  0.62911404 -1.58558584
## 10 0.06178627  1.18671719
```

Code chunk options

- You can specify options for each code chunk

```
' ' ' {r name, echo = TRUE, eval=TRUE, warning=FALSE, message=
R code here
' ' '
```

- name - A label for your code chunk

Code chunk options

- You can specify options for each code chunk

```
' ' ' {r name, echo = TRUE, eval=TRUE, warning=FALSE, message=
R code here
' ' '
```

- name - A label for your code chunk
- echo - Whether to display the code chunk or just show the results. If you don't want the reader of the document to see the code, but just the output, set echo=FALSE

Code chunk options

- You can specify options for each code chunk

```
' ' ' {r name, echo = TRUE, eval=TRUE, warning=FALSE, message=
R code here
' ' '
```

- name - A label for your code chunk
- echo - Whether to display the code chunk or just show the results. If you don't want the reader of the document to see the code, but just the output, set echo=FALSE
- eval - Whether to run the code in the code chunk

Code chunk options

- You can specify options for each code chunk

```
' ' ' {r name, echo = TRUE, eval=TRUE, warning=FALSE, message=
R code here
' ' '
```

- name - A label for your code chunk
- echo - Whether to display the code chunk or just show the results. If you don't want the reader of the document to see the code, but just the output, set echo=FALSE
- eval - Whether to run the code in the code chunk
- warning - Whether to display warning messages in the document

Code chunk options

- You can specify options for each code chunk

```
' ' ' {r name, echo = TRUE, eval=TRUE, warning=FALSE, message=
R code here
' ' '
```

- name - A label for your code chunk
- echo - Whether to display the code chunk or just show the results. If you don't want the reader of the document to see the code, but just the output, set echo=FALSE
- eval - Whether to run the code in the code chunk
- warning - Whether to display warning messages in the document
- message - Whether to display code messages in the document

Output visualization

```
iris <- read.csv("iris.csv", sep = ",")

head(iris)
```

```
##   X Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 1          5.1         3.5          1.4         0.2  setosa
## 2 2          4.9         3.0          1.4         0.2  setosa
## 3 3          4.7         3.2          1.3         0.2  setosa
## 4 4          4.6         3.1          1.5         0.2  setosa
## 5 5          5.0         3.6          1.4         0.2  setosa
## 6 6          5.4         3.9          1.7         0.4  setosa
```

Output visualization

```r
fit <- lm(Sepal.Length ~ Sepal.Width, iris)

summary(fit)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Sepal.Width, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5561 -0.6333 -0.1120  0.5579  2.2226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5262     0.4789   13.63   <2e-16 ***
## Sepal.Width  -0.2234     0.1551   -1.44    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8251 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

Output visualization

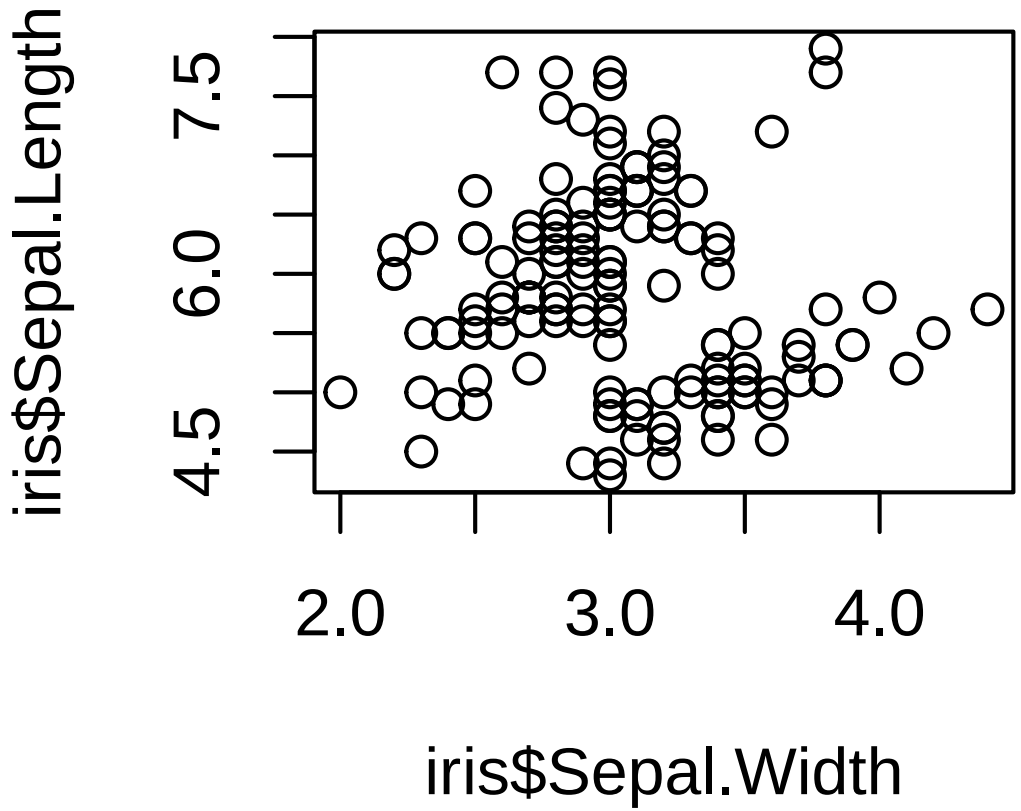
```
library(stargazer)

stargazer(fit, type = "text")
```

```
##
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                              Sepal.Length
## ------------------------------------------------
## Sepal.Width                    -0.223
##                                (0.155)
##
## Constant                      6.526***
##                                (0.479)
##
## ------------------------------------------------
## Observations                     150
## R2                              0.014
## Adjusted R2                     0.007
## Residual Std. Error       0.825 (df = 148)
## F Statistic              2.074 (df = 1; 148)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

Output visualization

```
plot(iris$Sepal.Width, iris$Sepal.Length)
```



Output visualization

```
![](screenparts.png)
```

```
---
title: "Quant II"
subtitle: "Lab 1: Intro to R Markdown, reproducibility, DAGs"
author: "Giacomo Lemoli"
date: "January 26, 2023"
output:
    beamer_presentation:
        theme: "Madrid"
        colortheme: "crane"
urlcolor: blue
---

# Ciao!
- Giacomo Lemoli, $5^{th}$ year PhD.
- I do research on CP and (H)PE
- I like: (i) politics of identity formation and language, (ii) statistics and causal inference
- Personal website: [giacomolemoli.com](https://giacomolemoli.com/)
```

Formatted text

It is very easy to format your text in Markdown.

- `*italics*` *italics*
- `**bold**` **bold**
- `~~strikethrough~~` ~~strikethrough~~
- `[nyu](https://www.nyu.edu/)` nyu
- `superscript^2^` superscript^2

You can find many more (e.g. headers, lists, etc.) here or Googling

Equations

- Use LaTex notation
- Inline equation (single dollar sign) We can use inline equations such as `$y_i = \alpha + \beta x_i + e_i$` which is displayed as $y_i = \alpha + \beta x_i + e_i$.
- Displayed equation (double dollar sign)

```
$$f(x) = \frac{e^{(-x-\mu)^{2}/(2\sigma^{2})}}
{\sigma \sqrt{2\pi}}$$
```

Which give this:

$$f(x) = \frac{e^{(-x-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}$$

Equations

For multiline equations, you need the `aligned` environment:

```
$$
\begin{aligned}
x & = 2*z + 3 \\
y & = 5*z + 6
\end{aligned}
$$
```

$$
\begin{aligned}
x & = 2*z + 3 \\
y & = 5*z + 6
\end{aligned}
$$

Inline code

- You can embed R output within text
- Use [r R-object] within the single backticks

```
mean <- mean(x)
```

The mean of x is bt-r mean-bt

The mean of x is 0.5515139

- Text gets updated automatically when code is updated too

Reproducibility

- One of our learning objectives in this class (and in other classes)

Reproducibility

- One of our learning objectives in this class (and in other classes)
- The importance of preserving reproducibility while working cannot be stressed enough

Reproducibility

- One of our learning objectives in this class (and in other classes)
- The importance of preserving reproducibility while working cannot be stressed enough
- It should be as easy as possible to reproduce all the steps that go from data to results

Reproducibility

- One of our learning objectives in this class (and in other classes)
- The importance of preserving reproducibility while working cannot be stressed enough
- It should be as easy as possible to reproduce all the steps that go from data to results
    - First and foremost for us and co-authors/collaborators, then for everyone else

Reproducibility

- One of our learning objectives in this class (and in other classes)
- The importance of preserving reproducibility while working cannot be stressed enough
- It should be as easy as possible to reproduce all the steps that go from data to results
  - First and foremost for us and co-authors/collaborators, then for everyone else
- Implications: structure of data files and folders, coding style

Files/folders

- Files should be set-up and organized in a coherent way

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder
- Unique directory for the class

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder
- Unique directory for the class
- Inside it, new directory for each new project

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder
- Unique directory for the class
- Inside it, new directory for each new project
- Within the project: use sub-directories

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder
- Unique directory for the class
- Inside it, new directory for each new project
- Within the project: use sub-directories
  - E.g. "Raw data" folder to store data, "Codes" folder for the scripts, "Output" folder for saving the results

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder
- Unique directory for the class
- Inside it, new directory for each new project
- Within the project: use sub-directories
  - E.g. "Raw data" folder to store data, "Codes" folder for the scripts, "Output" folder for saving the results
- From inside the script you can use the project folder as main directory

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder
- Unique directory for the class
- Inside it, new directory for each new project
- Within the project: use sub-directories
  - E.g. "Raw data" folder to store data, "Codes" folder for the scripts, "Output" folder for saving the results
- From inside the script you can use the project folder as main directory
  - R: `setwd("path_to_project_dir")`. Stata: `cd "path_to_project_dir"`

Files/folders

- Files should be set-up and organized in a coherent way
  - E.g. not keeping everything in the Desktop folder
- Unique directory for the class
- Inside it, new directory for each new project
- Within the project: use sub-directories
  - E.g. "Raw data" folder to store data, "Codes" folder for the scripts, "Output" folder for saving the results
- From inside the script you can use the project folder as main directory
  - R: `setwd("path_to_project_dir")`. Stata: `cd "path_to_project_dir"`
- Then your code pulls files from and saves files to the specific sub-directories

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)
- Code script should be concise and as short as possible

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)
- Code script should be concise and as short as possible
- Exhaustive and constant in-line comments

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)
- Code script should be concise and as short as possible
- Exhaustive and constant in-line comments
  - Every line (or sequence of lines that performs a self-contained task) should be preceded by a comment explaining what it does

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)
- Code script should be concise and as short as possible
- Exhaustive and constant in-line comments
  - Every line (or sequence of lines that performs a self-contained task) should be preceded by a comment explaining what it does
- Avoid redundancy

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)
- Code script should be concise and as short as possible
- Exhaustive and constant in-line comments
  - Every line (or sequence of lines that performs a self-contained task) should be preceded by a comment explaining what it does
- Avoid redundancy
  - If the same action is needed multiple times, better to write a callable function for it, or use loops

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)
- Code script should be concise and as short as possible
- Exhaustive and constant in-line comments
  - Every line (or sequence of lines that performs a self-contained task) should be preceded by a comment explaining what it does
- Avoid redundancy
  - If the same action is needed multiple times, better to write a callable function for it, or use loops
  - Copy-and-paste of code chunks makes errors more likely and is hard to read through

Code

- Coding style is like writing style: as personal as it can be, but should follow guidelines
  - Key principles are efficiency, clarity and transparency (mostly for ourselves and co-authors/collaborators)
- Code script should be concise and as short as possible
- Exhaustive and constant in-line comments
  - Every line (or sequence of lines that performs a self-contained task) should be preceded by a comment explaining what it does
- Avoid redundancy
  - If the same action is needed multiple times, better to write a callable function for it, or use loops
  - Copy-and-paste of code chunks makes errors more likely and is hard to read through
- Play around with replication packages to learn coding tricks

Resources

- Great resource: Gentzkow and Shapiro (2014), Code and Data for the Social Sciences: a Practitioner's Guide

DAGs

- Represent causal mechanisms, i.e. causal relationship between different variables

DAGs

- Represent causal mechanisms, i.e. causal relationship between different variables
- Important concepts:

DAGs

- Represent causal mechanisms, i.e. causal relationship between different variables
- Important concepts:
  - Backdoor path: non-causal dependency between D and Y

DAGs

- Represent causal mechanisms, i.e. causal relationship between different variables
- Important concepts:
  - Backdoor path: non-causal dependency between D and Y
  - Collider: variable caused by different variables

Conditioning in DAGs

Conditioning on some variable w in a DAG is equivalent to do the following steps:

- If w is a collider, link all pairs of parents of w by drawing an undirected edge between them

Conditioning in DAGs

Conditioning on some variable w in a DAG is equivalent to do the following steps:

- If w is a collider, link all pairs of parents of w by drawing an undirected edge between them
- For any ancestor of w, if this ancestor is itself a collider, link all pairs of parents of this ancestor with undirected edges to connote induced dependencies

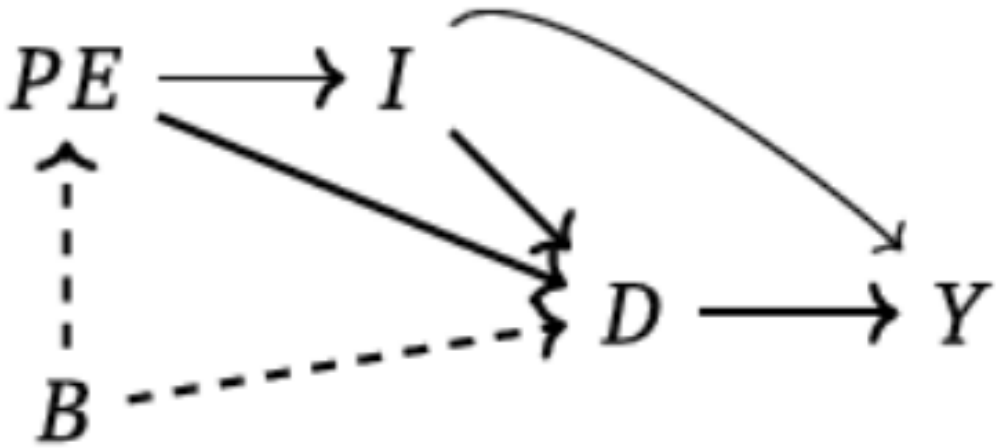
Conditioning in DAGs

Conditioning on some variable w in a DAG is equivalent to do the following steps:

- If w is a collider, link all pairs of parents of w by drawing an undirected edge between them
- For any ancestor of w, if this ancestor is itself a collider, link all pairs of parents of this ancestor with undirected edges to connote induced dependencies
- Erase w from the graph and all the edges connected with w

DAGs

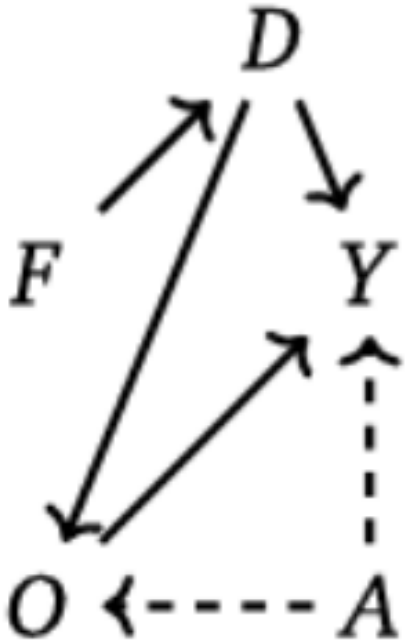
An example from the Mixtape



Collider bias

- Let's see these properties at work, with some simulations
- Examples and code from The Mixtape, (pp. 108-113)

Case 1: the effect of gender discrimination on women income



Collider bias

```
library(tidyverse)
library(stargazer)

# Set seed
set.seed(123)

# Simulate our data
tb <- tibble(
  female = ifelse(runif(10000) >= 0.5, 1, 0),
  ability = rnorm(10000),
  discrimination = female,
  occupation = 1 + 2*ability + 0*female - 2*discrimination + rnorm(10000),
  wage = 1-1*discrimination + 1*occupation + 2*ability + rnorm(10000)
)

# Estimate regressions
lm_1 <- lm(wage ~ female, tb)
lm_2 <- lm(wage ~ female + occupation, tb)
lm_3 <- lm(wage ~ female + occupation + ability, tb)
```

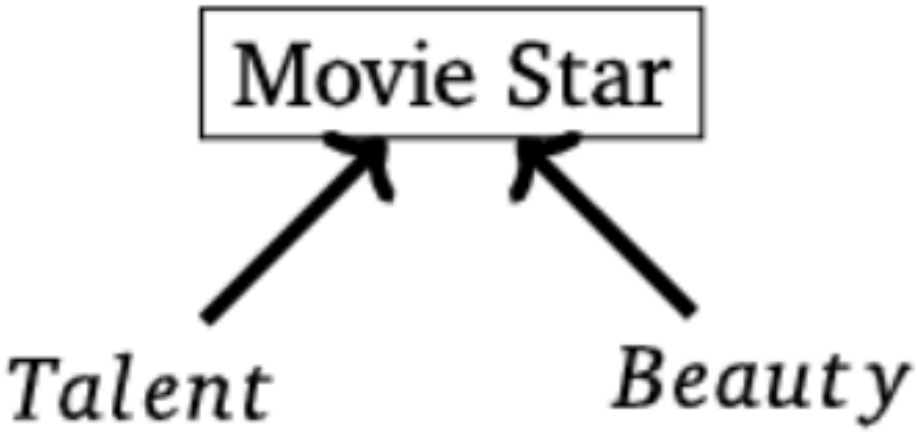
Collider bias

```
# Compare
stargazer(lm_1, lm_2, lm_3, type = "text",
          column.labels = c("Biased unconditional",
                            "Biased",
                            "Unbiased Conditional"))
```

```
##
## ================================================================================================
##                                                Dependent variable:
##                    ------------------------------------------------------------------------------
##                                                       wage
##                          Biased unconditional          Biased              Unbiased Conditional
##                                   (1)                  (2)                        (3)
## ------------------------------------------------------------------------------------------------
## female                          -3.066***            0.587***                   -1.050***
##                                  (0.085)             (0.030)                    (0.028)
##
## occupation                                           1.796***                   0.987***
##                                                      (0.006)                    (0.010)
##
## ability                                                                         2.033***
##                                                                                 (0.022)
##
## Constant                         2.023***            0.222***                   1.025***
##                                  (0.060)             (0.020)                    (0.017)
##
## ------------------------------------------------------------------------------------------------
## Observations                      10,000              10,000                     10,000
## R2                                 0.114               0.912                      0.952
## Adjusted R2                        0.114               0.912                      0.952
## Residual Std. Error          4.265 (df = 9998)   1.347 (df = 9997)          0.994 (df = 9996)
## F Statistic           1,292.306*** (df = 1; 9998) 51,551.530*** (df = 2; 9997) 65,927.470*** (df = 3; 9996)
## ================================================================================================
## Note:                                                                *p<0.1; **p<0.05; ***p<0.01
```

Collider bias

Case 2: Talent and beauty



DAG simulations

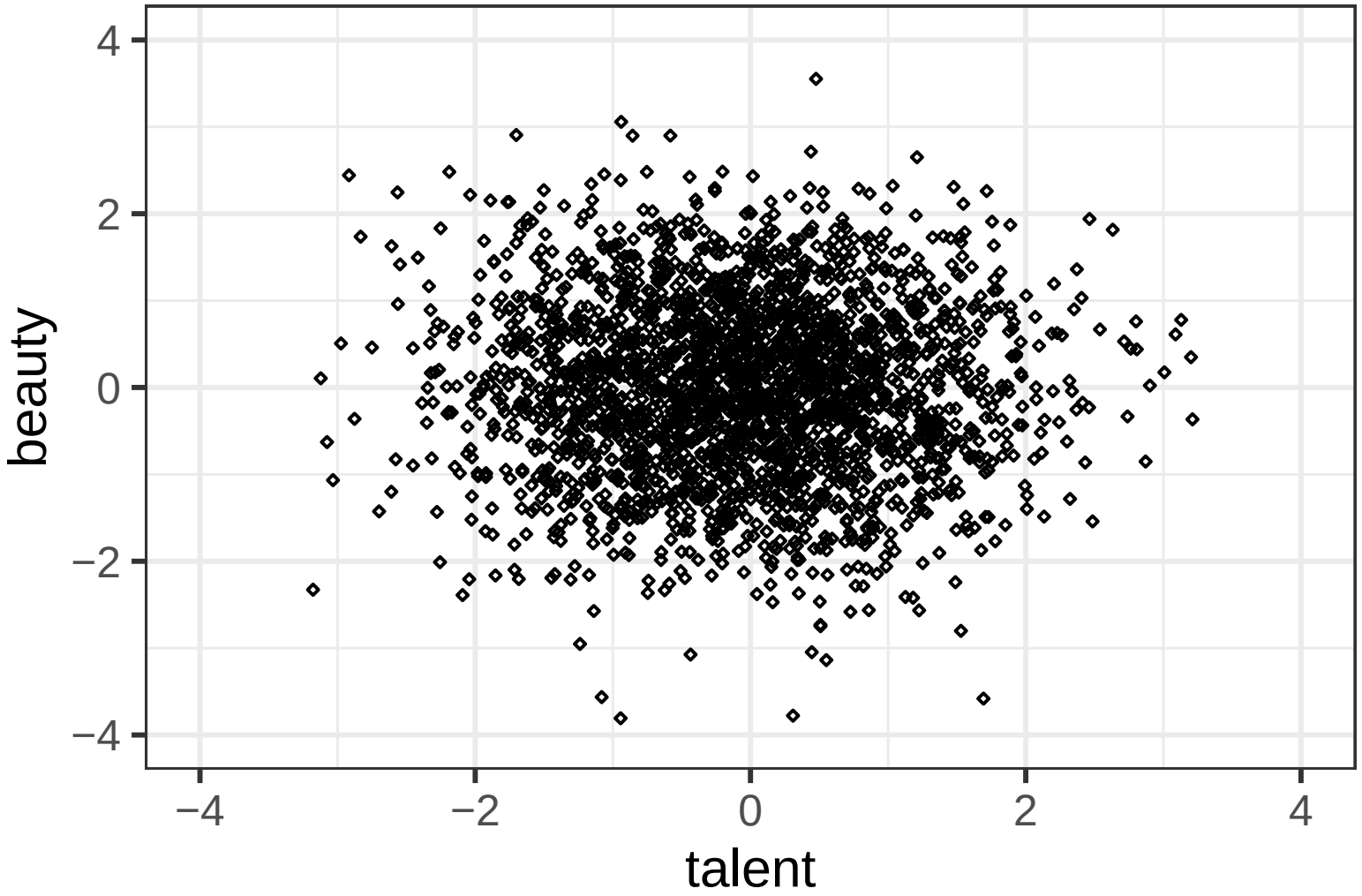
```
library(tidyverse)

# Set seed
set.seed(3444)

# Simulate data
star_is_born <- tibble(
  beauty = rnorm(2500),
  talent = rnorm(2500),
  score = beauty + talent,
  c85 = quantile(score, .85),
  star = ifelse(score >= c85, 1, 0)
)
```

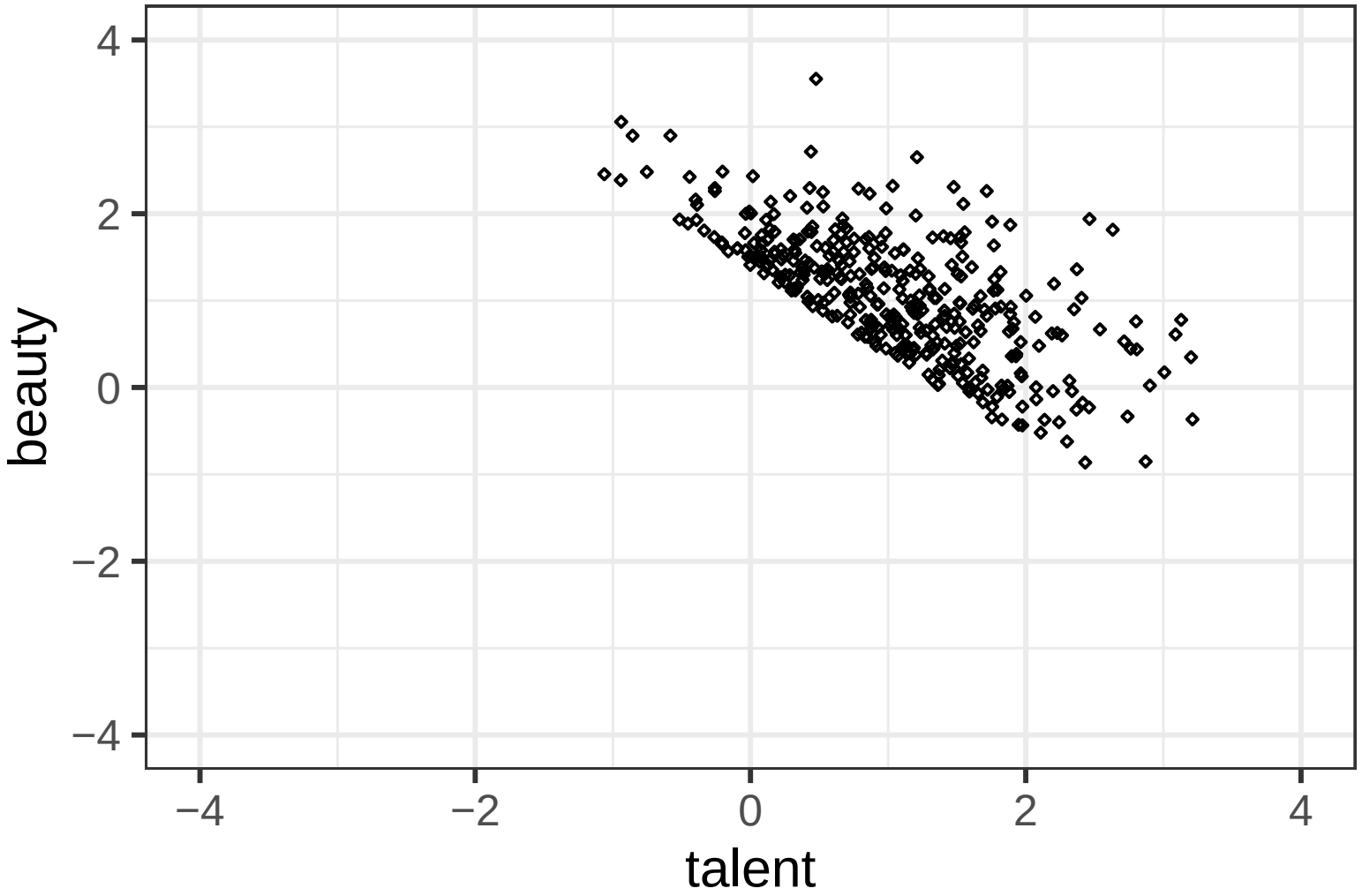
DAG simulations

```
star_is_born %>%
  lm(beauty ~ talent, .) %>%
  ggplot(aes(x = talent, y = beauty)) +
  geom_point(size = 0.5, shape = 23) + xlim(-4, 4) + ylim(-4, 4) +
  theme_bw()
```



DAG simulations

```
star_is_born %>%
  filter(star == 1) %>% lm(beauty ~ talent, .) %>%
  ggplot(aes(x = talent, y = beauty)) +
  geom_point(size = 0.5, shape = 23) + xlim(-4, 4) + ylim(-4, 4) +
  theme_bw()
```



Conclusion

- Don't control for/condition on colliders
  - Endogenous sample selection is a form of collider bias
  - See discussion in Knox et al (2020) on admin data

Additional resources:

- Elwert and Winship (2014), Endogenous selection bias: the problem of conditioning on a collider variable
- Knox, Lucas, and Cho (2022), Testing causal theories with learned proxies
- Schneider (2020), Collider bias in economic history research