Computational Methods in Economics

Data

January 21, 2020

Data

- Data for statistical analysis
  - Cross section
  - Longitudinal data
- A data structure is a particular way of organizing information in a computer so that it can be used effectively.
  - Array
  - Linked list
  - Stack, Queue, Heap, Hashing, Graph
  - Vector, Matrix
  - Dataframe
    - Individual/Time identifiers
    - Variables

Data: Tools

- Serious data analysis requires to store and access information quickly.
  - Python
  - SQL
- Libraries dplyr and data.table are recommenced on R.

Data: Tools (0)

- dplyr works with pipes % > %
- Basic structure of a dataframe.
    - Each variable is in its own column
    - Each observation is in its own row

Data: Tools (1)

Logical and boolean operators

- $<$ and $<=$
- $>$ and $>=$
- is.na() and !is.na()
- %*in*%
- $|$ or *xor*() and &
- $==$ and $=$!

Data: Tools (2)

Manipulate cases

- **filter(.data.,...)**: Extract rows that meet a logical criteria.
- **distinct(.data.,...)**: Remove rows with duplicate values
- **slice(.data.,...)**: Select rows by position
- **arrange(.data.,...)**: Order rows by values of a column or columns (low to high).

Data: Tools (3)

Manipulate Variables

- **pull(.data.,var)**: Extract column values as a vector.
- **select(.data.,...)**: Extract columns as a table.
- **mutate(.data.,var)**: Compute new columns.
- **rename(.data.,...)**: rename columns.
- **add_column(.data.,...), add_count(), add_tally()**

Data: Tools (4)

Combine variables

- **bind_cols()**: return tables side by side.
- Mutating join: match two dataframes or tables.
  - **left_join(x,y,by=NULL)**: join matching values from y and x.
  - **right_join(x,y,by=NULL)**: join matching values from x and y.
  - **inner_join(x,y,by=NULL)**: join data. Retain only rows with matches.
  - **full_join(x,y,by=NULL)**: join data. Retain all values, all rows.
- **semi_join(x,y,by=NULL)**: WHAT WILL BE JOINED
- **anti_join(x,y,by=NULL)**: WHAT WILL NOT BE JOINED

Data: Visualize

Library ggplot.

- **ggplot (data = DATA)**
- **GEOM_FUNCTION**
  - (mapping = aes(MAPPINGS)
  - stat = STAT ,
  - position = POSITION)
- COORDINATE_FUNCTION
- FACET_FUNCTION
- SCALE_FUNCTION
- THEME_FUNCTION

-

```
dat <- read.dta13("nlsy_merged.dta")
names(dat)

dat1 = subset(dat,select=c(year,job_req_educ),!is.n

tab = dat1 %>% group_by(year,job_req_educ) %>%
summarise(n = n()) %>%
mutate(freq = n / sum(n))

tab



graph = ggplot(tab,aes(x=year,y=freq,color=as.facto
geom_line(size=2) + ylab("Rates") + xlab("Years") +

graph = graph +theme(axis.title = element_text(size
axis.text = element_text(size = rel(1.1)),
strip.text = element_text(size = 12),
```