Data Visualization Notes

Luis Moreno

12/27/2020

Introduction to Tidyverse

The process of structuring datasets to facilitate analysis is called "data tidying" by Hadley Wickham.

Requisites to classify a dataset as tidy:

1. Each variable forms a column
2. Each observation forms a row
3. Each type of observational unit forms a table.

More information on tidy data can be found at:

```
vignette("tidy-data")
```

To use the "tidyverse" commands the package must be installed first and then loaded with:

```
library(tidyverse)
```

Importing Data

Read CSV

The difference between the base R read.csv and tidyverse read_csv is the output, this latter results in a "tibble" class object.

"**Tibble**": a specialized version of a data frame that is setup to work better with the tidyverse.

More information on "tibbles" can be found at:

```
vignette("tibble")
```

```
file <- "/Users/luism/Documents/R/data-viz-intro/data/week 2/cces_sample_coursera.csv"

cces_data <- read_csv(file)

# Show the class of an R object
class(cces_data)

# Switch back and forth between tibble and dataframe
cces_dataframe <- as.data.frame(cces_data)
cces_tibble <- as_tibble(cces_dataframe)

# Drop rows with missing data from a dataset
cces_data <- drop_na(cces_data)

# Filter data using conditionals
```

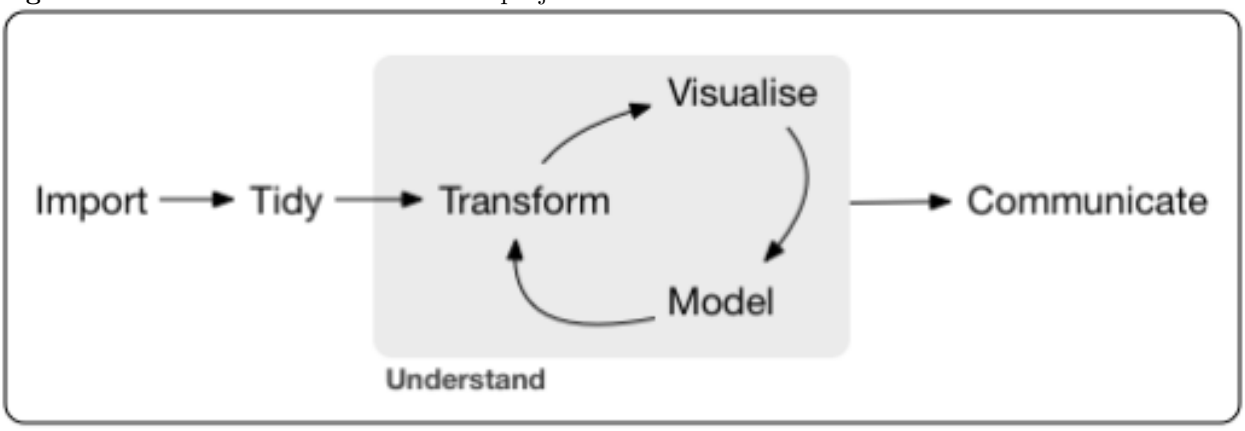
```r
## Filter data for only women
women_data <- filter(cces_data, gender==2)

# Display the number of observations and variables of a dataset
dim(cces_data)
dim(women_data)

# Contingency table of the counts at each combination of factors
## Number of observations by each type of gender
table(cces_data$gender)

# Filter data with logical operators
## Filter data for just republican women
republican_women <- filter(cces_data,gender==2 & pid7>4)

# Show first entries of a dataset
head(republican_women)

# Keep the specified columns from a dataset
## Keep "educ" and "employ" columns from the full dataset
select(cces_data, "educ", "employ")

# Combine multiple commands with "pipe" operator
## Filter republican women and keep only two variables from the original dataset
women_republicans_educ_employ <- cces_data %>% filter(gender==2 & pid7>4) %>% select("educ","employ")

# Recode variables
## Replace numeric values based on their position, and character values by their name
party <- recode(cces_data$pid7,`1`="Democrat",`2`="Democrat",`3`="Democrat",`4`="Independent",`5`="Repul

cces_data$party <- party

rec_sen1_01 <- recode(cces_data$CC18_310b,`1`=0,`5`=0,`2`=1,`3`=1,`4`=1)

rec_sen2_01 <- recode(cces_data$CC18_310c,`1`=0,`5`=0,`2`=1,`3`=1,`4`=1)

cces$rec_sen1_01<- rec_sen1_01

# Rename variables by name, not position
## Modify the name of variable CC18_308a by "trump_approval"
test <- rename(cces_data,trump_approval=CC18_308a)

## Assign the modified dataset to its original name to save the changes
cces_data <- test

# Reorder rows by column values
## Reorder by variable name
sort_by_gender_and_party <- cces_data %>% arrange(gender,pid7)

## Reorder data by a variable and in descending order
sorted_by_gender_and_party <- cces_data %>% arrange(gender,desc(pid7))

# Group data by variables
```

```r
## Group data by gender and political party
grouped_gender_pid7 <- cces_data %>% group_by(gender,pid7)

# Ungroup data by variables
ungroup(grouped_gender_pid7)

# Reduce multiple values down to a single summary
## Summarize by average "pid7" and "faminc_new"
cces_data %>% summarise(mean_pid7=mean(pid7),mean_faminc=mean(faminc_new))

## Summarize by group
cces_data %>% group_by(gender) %>% summarise(mean_pid7=mean(pid7), mean_faminc=mean(faminc_new))
```

R for Data Science

1. Introduction

Figure 1. Tools needed for a data science project



Data Science Project Model

1. Import data
    - File,
    - Database,
    - Web API.
2. Tidy data
    - Store data in a consistent form,

Functions Glossary

Data Manipulation Functions

read_csv()
as.data.frame()
as_tibble()
drop_na()
filter()
dim()

table()
head()
select()
filter()
recode()
rename()
arrange()
group_by()
ungroup()
summarise()

Definitions Glossary

pipes
tidy data