Measurement 1 - tidyverse

Anna Yorozuya

University of Tokyo

June 2, 2022

Table of Contents

- Handling (missing) data
- Introduction to `ggplot2`
- Today's in-class assignment: `gay-marriage-revisited`

Section 1

Handling (missing) data

Handling (missing) data

arrange(): reordering data

reordering the rows from low to high (low to high with desc())

```
afghan_age <- select(afghan, province, district, age)
head(arrange(afghan_age, age))
```

```
##   province  district age
## 1    Kunar Ghaziabad  15
## 2    Logar    Khoshi  16
## 3    Logar Puli Alam  16
## 4    Kunar  Asadabad  16
## 5    Kunar  Asadabad  16
## 6    Kunar  Asadabad  16
```

```
head(arrange(afghan_age, desc(age)))
```

```
##   province       district age
## 1  Uruzgan Shahidi Hassas  80
## 2    Kunar      Ghaziabad  78
## 3    Logar         Khoshi  75
## 4    Kunar     Chapa Dara  75
## 5    Logar   Baraki Barak  73
## 6    Logar   Baraki Barak  73
```

Handling (missing) data

drop_na(): listwise deletion (base R: na_omit())

Remove all observations with at least one missing value from a data frame.

```
num1 <- c(2, 4, 3, NA)
num2 <- c(5, NA, 9, 8)
num <- tibble(num1, num2)
drop_na(num)
```

```
## # A tibble: 2 x 2
##    num1  num2
##   <dbl> <dbl>
## 1     2     5
## 2     3     9
```

Example for arrange()

```
## Table for non-missing values of ISAF and Taliban
afghan %>%
  filter(!is.na(violent.exp.ISAF), !is.na(violent.exp.taliban)) %>%
  group_by(violent.exp.ISAF, violent.exp.taliban) %>%
  count() %>%
  ungroup() %>%
  mutate(prop = n / sum(n)) %>%
  arrange(prop) # compare to arrange(desc(prop))
```

```
## # A tibble: 4 x 4
##   violent.exp.ISAF violent.exp.taliban     n  prop
##              <int>               <int> <int> <dbl>
## 1                0                   1   354 0.132
## 2                1                   0   475 0.177
## 3                1                   1   526 0.196
## 4                0                   0  1330 0.495
```

Example for drop_na()

```r
# check how many NAs will be omitted!
nrow(afghan) # original
```

```
## [1] 2754
```

```r
afghan.sub.2 <- drop_na(afghan)
nrow(afghan.sub.2) # NAs omitted
```

```
## [1] 2554
```

```r
afghan %>%
  drop_na(income) %>%
  nrow() # NAs in income omitted
```

```
## [1] 2600
```

Section 2

Introduction to ggplot2

Introduction to ggplot2

What is ggplot2?

- A package in tidyverse, which allows visualization of data in a more intuitive way.
- To make a plot, you assign the data and aesthetics (mapping) first, and add layers to tell the `ggplot` function what you want the figure to look like.
- The package name is **ggplot2**, while the function is **`ggplot()`**.

The advantage of using ggplot2

- Intuitive: very simple grammar
- Flexibility: you can build everything with the grammar
- Very nice-looking graphs!

Basics of ggplot2

Basic syntax

ggplot(data = **DATA**) +
GEOM_FUNCTION(mapping = aes(**MAPPINGS**)) +
ADDITIONAL_FUNCTIONS

* Add the components with "+"

Elements

- **DATA**: specify the dataset to use in the graph
- **GEOM_FUNCTION**: starting from "`geom_`", where you specify the types of figures such as bar plot or histogram.
- **MAPPINGS**: defines how variables in your dataset are mapped to visual properties. Commonly used arguments are `x` and `y` to specify which variables to map to each axis.
- **ADDITIONAL_FUNCTIONS**: additional layers

ggplot2: GEOM_FUNCTION

Plots

- geom_point(): scatterplot
- geom_histogram(): histogram
- geom_bar(): bar plot
- geom_boxplot(): box plot
- geom_line(): line chart
- geom_smooth(): smooth line, mainly for regression
- geom_ribbon(): show confidence intervals

Line

- geom_abline(): intercept, slope
- geom_hline(): yintercept
- geom_vline(): xintercept

ggplot2: aes(MAPPINGS)

Aesthetics

- x = variable: values for x axis
- y = variable: values for y axis
- color = variable: assign unique color for each value of the variable
- fill = variable: assign the unique color to fill in for each value
- size = variable: assign unique size for each value of the variable
- alpha = variable: control the level of transparency for each value
- shape = variable: change the shape of points in geom_point
- position =: fill for stacking, dodge for avoiding overlapping, jitter for solving overplotting

*All of those above should be within aes()

- If it goes outside of aes(), the above arguments will be applied to all the variables, regardless of the value, unless specified.

ggplot2: ADDITIONAL_FUNCTIONS

Scales

Map the data values to visual values

`scale_!_!!`, where

!: aesthetic to adjust (x, y, fill, etc.)

!!: prepackaged scale to use

- `scale_x_discrete()`: set discrete values for visualization
- `scale_x_continuous()`: set continuous values for visualization
- `scale_fill_discrete()`: fill the plot with discrete values

Labels

- `labs(title = "", x = "", y = "")`: label for x-axis, y-axis, and title
- `ylab("label")`: label for y-axis
- `xlab("label")`: label for x-axis
- ‘ggtitle(“title”): title

ggplot2: `ADDITIONAL_FUNCTIONS`

Limits

- `xlim()`: limits for x-axis
- `ylim()`: limits for y-axis

Themes

- `theme_classic()`
- `theme_bw()`: white background with grid lines
- `theme_gray()`: grey background (default)
- `theme_void()`: empty theme

ggplot2: ADDITIONAL_FUNCTIONS

Facets

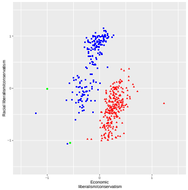
- facet_wrap(~ a): facet by a single variable (a in the example)
- facet_grid(b ~ c): facet by two variables

Coordinate systems

- coord_flip(): switch x and y axis
- coord_quickmap(): set the aspect ratio correctly for maps
- coord_fixed(): fix the aspect ratio to square

Example: Scatterplot

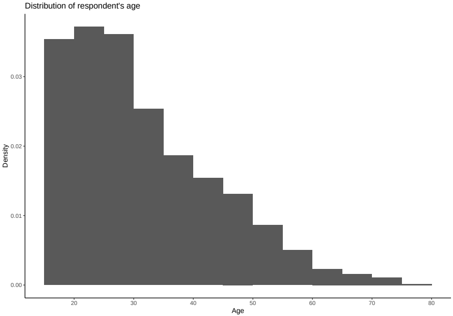
```
ggplot(data = filter(congress, congress == 80),
       aes(x = dwnom1, y = dwnom2)) +
  geom_point(aes(shape = party, color = party),
             show.legend = FALSE) +
  scale_color_manual(values = c(Democrat = "blue",
                                Republican = "red",
                                Other = "green")) +
  scale_shape_manual(values = c(Democrat = "square",
                                Republican = "triangle",
                                Other = "circle")) +
  scale_y_continuous("Racial liberalism/conservatism", limits = c(-1.5, 1.5)) +
  scale_x_continuous("Economic\n liberalism/conservatism", limits = c(-1.5, 1.5)) +
  coord_fixed()
```



- scale_color_manual(): specify which colors are used for which value
- scale_shape_manual(): specify which shape is used for which value
- scale_y_continuous()/ scale_x_continuous(): add title, change the limits
- coord_fixed(): squared aspect ratio

Example: Histogram (basic)

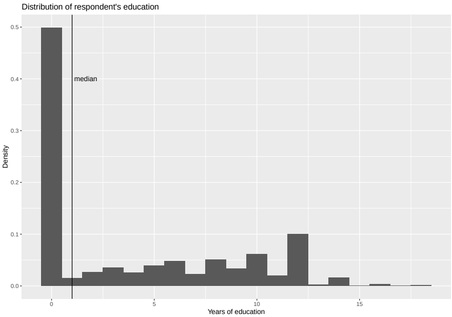
```
ggplot(afghan, aes(x = age)) + # the data and initial aes()
  geom_histogram(aes(y = ..density..), # histogram, additional aes()
                 binwidth = 5, # how wide for each bin
                 boundary = 0) + # bin position
  scale_x_continuous(breaks = seq(20, 80, by = 10)) +
  labs(title = "Distribution of respondent's age",
       y = "Density", x = "Age") +
  theme_classic()
```



- `aes(y = ..density..)`: y-axis shows the density, not the count
- `aes(binwidth = 5)`: set the width of each bin
- `aes(boundary = 0)`: the position of bins
- `scale_x_continuous()`: change the ticks of x-axis

Example: Histogram (advanced)

```
ggplot(afghan, aes(x = educ.years, y = ..density..)) +
  geom_histogram(binwidth = 1, center = 0) +
  geom_vline(xintercept = median(afghan$educ.years)) +
  annotate(geom = "text", x = median(afghan$educ.years),
           y = 0.4,
           label = "median",
           hjust = -0.1) +
  labs(title = "Distribution of respondent's education",
       x = "Years of education",
       y = "Density")
```



- geom_vline(): add a vertical line
- annotate(): add text to the plot. specify the position and text

How to save/print graphs

ggsave

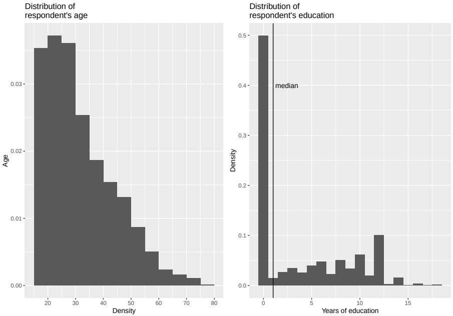
- ggsave(path, filename, extension)
- for example, if you want to wave the figure as a pdf in the result_fingures directory, ggsave("results_figures/education_by_province.pdf")

gridExtra

- save multiple plots into a single file
- first, load the package with library(gridExtra)
- use the grid_arrange()

Example: gridExtra

```r
library(gridExtra)
## The age histogram
age_hist <- ggplot(afghan, aes(x = age)) +
  geom_histogram(aes(y = ..density..), binwidth = 5, boundary = 0) +
  scale_x_continuous(breaks = seq(20, 80, by = 10)) +
  labs(title = "Distribution of \nrespondent's age", y = "Age", x = "Density")
## The education histogram
educ_hist <- ggplot(afghan, aes(x = educ.years, y = ..density..)) +
  geom_histogram(binwidth = 1, center = 0) +
  geom_vline(xintercept = median(afghan$educ.years)) +
  annotate(geom = "text", x = median(afghan$educ.years), y = 0.4, label = "median", hjust = -0.1) +
  labs(title = "Distribution of \nrespondent's education", x = "Years of education", y = "Density")
## Put the plots side-by-side
grid.arrange(age_hist, educ_hist, ncol = 2)
```



References

- “Quantitative Social Science: An Introduction” - Kosuke Imai
- “Quantitative Social Science: An Introduction in Tidyverse” - Kosuke Imai and Nora Webb Williams
- R for data science - H.Wickham and G.Grolemund