Statistics with Recitation: TA Session

Danny Po-Hsien Kang (康柏賢)

October 14, 2025

Midterm Week

- Midterm: 9:30 AM - 12:00 PM, Oct 22 (Wed)
- You can bring: One cheat sheet, Calculator
- **No questions about R will appear in midterm exam!**
- No TA Session in midterm week.
    - TA office hour: 16:30-19:00 (Tue.)

Today's agenda

1. Draw Graphs
    - `geom_vline()`
    - `geom_hline()`
    - `facet_wrap()`
    - `ggsave()`

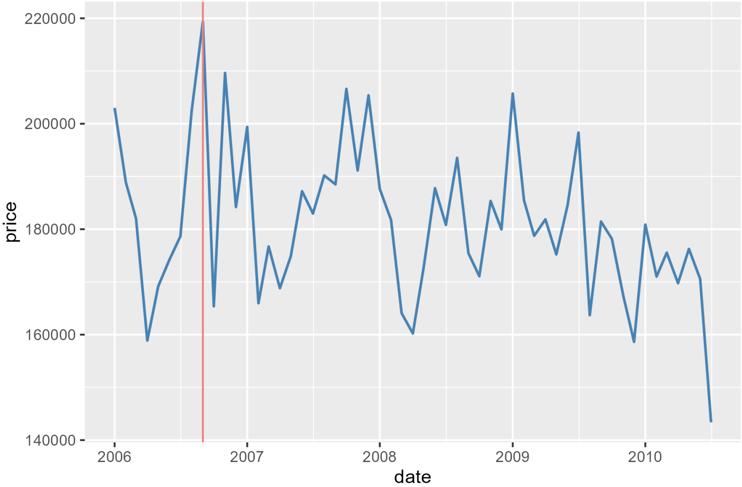
Today's Dataset

- Please download the following dataset:
  - `ames.csv` (example data)
  - `loans_full_schema.csv`

  from the TA Session's Website or the OpenIntro website.
- Please import the following datasets into RStudio.

```r
housing <- read.csv("data/ames.csv")
loan <- read.csv("loans_full_schema.csv")
```

Draw Vertical Line: `geom_vline()`



Draw Vertical Line: geom_vline()

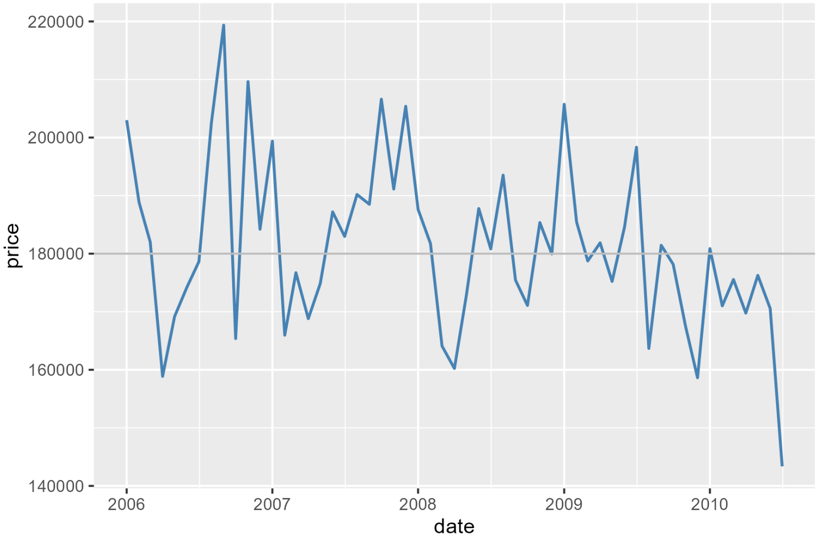
- **Syntax:**

```
Your_graph +
  geom_vline(xintercept = ..., color = ...)
```

- **Example:**

```
ggplot(housing, aes(x = date, y = price)) +
    geom_line(stat = "summary",
              fun = "mean",
              linewidth = 0.7,
              color = "steelblue") +
    geom_vline(xintercept = as.Date("2006-09-01"),
               color = "lightcoral")
```

Draw Horizontal Line: geom_hline()



Draw Horizontal Line: geom_hline()

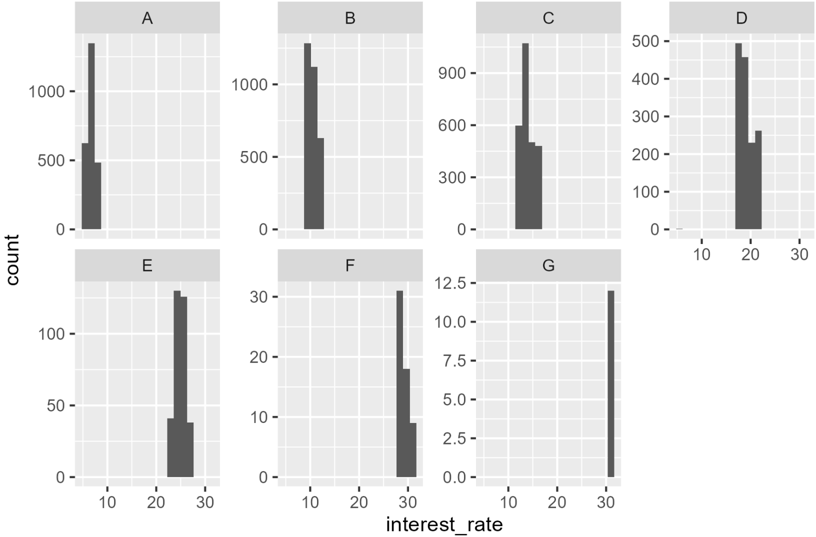
- **Syntax:**

```
Your_graph +
  geom_hline(yintercept = ..., color = ...)
```

- **Example:**

```
ggplot(housing, aes(x = date, y = price)) +
    geom_line(stat = "summary",
              fun = "mean",
              linewidth = 0.7,
              color = "steelblue") +
    geom_hline(yintercept = 180000,
               color = "gray")
```

Splitting Graph into Groups: facet_wrap()



Splitting Graph into Groups: facet_wrap()

- **Syntax:**

```
Your_graph +
  facet_wrap(~variable,
  scales = "free_y",
  nrow = ...)
```

- **Example:**

```
ggplot(loan, aes(x = interest_rate)) +
  geom_histogram(bins = 20) +
  facet_wrap(~grade,
             scales = "free_y",
             nrow = 2)
```

Splitting Graph into Groups: facet_wrap()

- **Example:**

```
ggplot(loan, aes(x = interest_rate)) +
  geom_histogram(bins = 20) +
  facet_wrap(~grade,
             scales = "free_y",
             nrow = 2)
```

- ~grade: Facet by grade. Each unique grade value gets its own panel.
  - Works with multiple variables. e.g. grade + disbursement_method.
- scales = "free_y": Scales setting for x, y axes.
  - "free_y" means x scales vary per panel and y fixed.
  - Other type: fixed (default), free_x, free.
- nrow = 2: Force 2 rows in the panel.
  - ncol works as well.

Save Your Graph into File: ggsave()

- **Syntax:**

```
ggsave(filename = ...,
       plot = ...,
       path = ...,
       width = ...,
       height = ...,
       dpi = ...)
```

- **Example:**

```
vline_plot <- ggplot(housing, aes(x = date, y = price)) +
    geom_line(stat = "summary",
              fun = "mean",
              linewidth = 0.7,
              color = "steelblue") +
    geom_vline(xintercept = as.Date("2006-09-01"),
                     color = "lightcoral")
```

ggsave(): Which plot is saved?

- By default, ggsave() saves the newest plot showed in the plot window.

```
print(plot_1)
print(plot_2)
ggsave(filename = ...) #It saves plot_2!
```

- Use plot = ... to assign which plot you want to save. (Or just put ggsave() right after your plot produced.)
- Use filename = ... to set the file names and the output type (.png, .jpeg ...).
- **Example:**

```
ggsave(filename = "vline_plot.png",
       plot = "vline_plot")
```

ggsave(): Where is my plot saved?

- By default, ggsave() saves your plot to the working directory.
    - It saves to the project folder if you have opened one.
    - But it is better to save in the plots folder!
- Two ways to assign the saved path:
    - Type the path inside `filename = ...` (absolute or relative).
    - Or use `path = ...`
- **Example:**

```
ggsave(filename = "C:/Users/you/figures/vline_plot.png")
ggsave(filename = "figures/vline_plot.png")
ggsave(filename = "vline_plot.png", path = "figures")
```

ggsave(): How big is my plot?

- By default, ggsave() uses the current plot size in the RStudio Plots window.
    - Which is not good because you may produce difference size of plots every time!
- Use `width = ...` and `height = ...` to control the size.
- Use `dpi = ...` to control the quality of the plot (default $= 300$).
- **Example:**

```
ggsave(filename = "vline_plot.png",
       width = 6,
       height = 4,
       dpi = 300)
```

ggsave(): Putting everything together

- **Example:**

```
ggsave(filename = "vline_plot.png",
       plot = vline_plot,
       path = "figures",
       width = 6,
       height = 4,
       dpi = 300)
```