Week 7:

- Joins
- Plotting with ggplot2 (I)

Let’s recap

- What makes a dataset tidy?
- What is the count() fuction?
- When would we use the pivot_wider() fuction?
- When would we use the pivot_longer() fuction?
- When would we use the separate() fuction?

Today

- **Other useful functions**
  - **rename()**
  - **as.character()/ as.numeric()/ as.factor()**
  - **recode()**
  - **glimpse()**

- Transform
  - Join ()

- Visualize
  - ggplot()
    - Data
    - Aesthetics
    - Geometries

Let's get set

- Create an R project for this session and name it “week_7”
- Open the script file and rename it
- Load the tidyverse package
- Import the Sinai covid dataset

Manual column names cleaning

- rename() uses the style NEW = OLD (the new column name is given before the old column name)

Automatic column names cleaning

- The function clean_names() converts all names to consist of only underscores, numbers, and letters

as.character/ as.numeric/ as.Date

```
sinai_covid <- sinai_covid %>%
        mutate (copd = as.factor(copd))

sinai_covid <- sinai_covid %>%
mutate_at (vars (sex:cancer_flag), as.factor)
```

Recode()

```
sinai_covid %>%
        mutate(asthma = fct_recode(asthma, "Yes" = "1", "No" = "0"))
```

glimpse()

- glimpse (sinai_covid)

Your turn! Exercise 1

Using the pipe, create a new dataset from sinai_covid in which:

- The sex variable is named gender and the levels are woman/man
- deceased_indicator is a factor

Take a glimpse at the new dataset

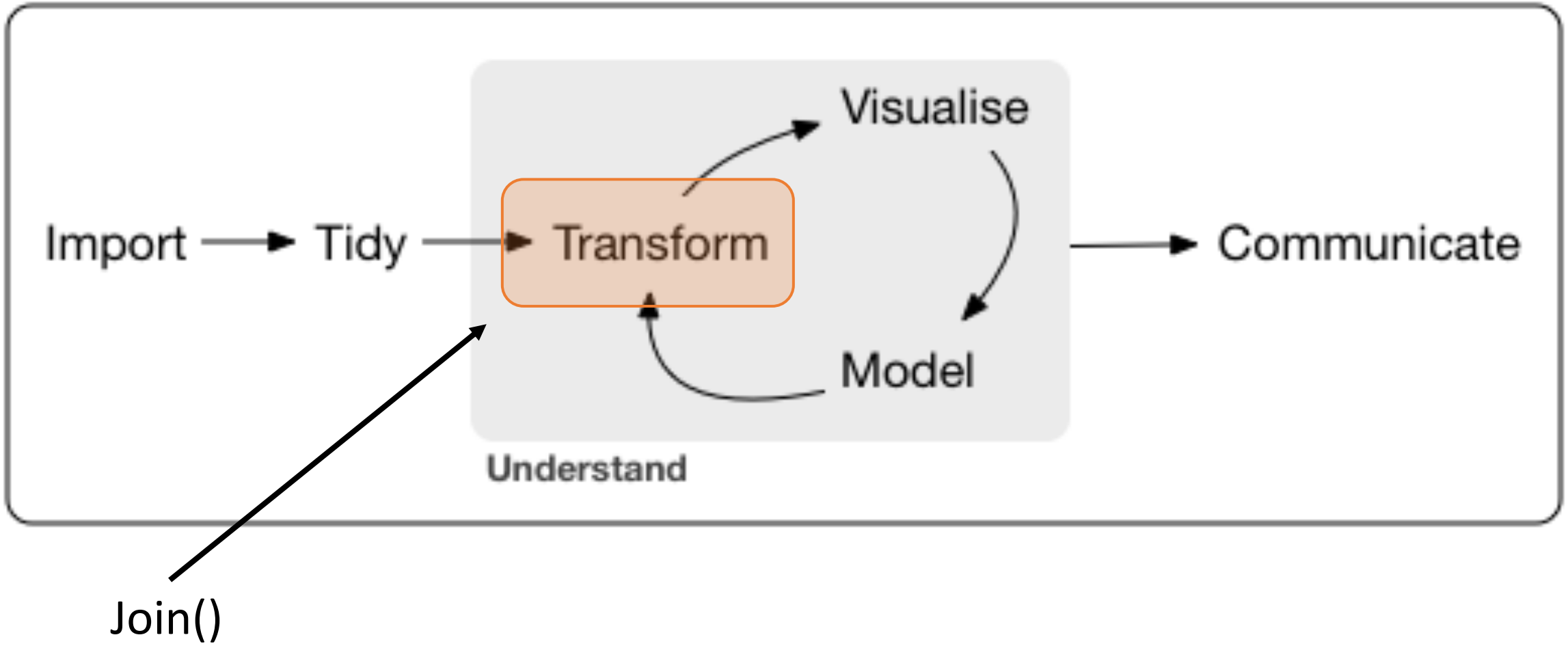
Today

- Other useful functions
  - rename()
  - as.character()/ as.numeric()/ as.factor()
  - recode()
  - glimpse()

- **Transform**
  - **Join ()**

- Visualize
  - ggplot()
    - Data
    - Aesthetics
    - Geometries

A typical data science project :



join()

- Create 2 tibbles

```
data1 <- tibble(ID = 1:3,
                age = c("23", "25", "21"))

data2 <- tibble(ID = 2:4,
                sex = c(0,1,0),
                heart_disease = c(FALSE, FALSE, TRUE))
```

ID	age
1	23
2	25
3	21

ID	sex	heart_disease
2	0	FALSE
3	1	FALSE
4	0	TRUE

inner_join()

data_inner <- inner_join (data1, data2, by = "ID")

ID	age
1	23
2	25
3	21

ID	sex	heart_disease
2	0	FALSE
3	1	FALSE
4	0	TRUE

ID	age	sex	heart_disease
2	25	1	FALSE
3	21	0	FALSE

left_join()

data_left <- left_join (data1, data2, by = "ID")

ID	age
1	23
2	25
3	21

ID	sex	heart_disease
2	0	FALSE
3	1	FALSE
4	0	TRUE

ID	age	sex	heart_disease
1	23	*NA*	*NA*
2	25	1	FALSE
3	21	0	FALSE

right_join()

data_right <- right_join (data1, data2, by = "ID")

ID	age
1	23
2	25
3	21

ID	sex	heart_disease
2	0	FALSE
3	1	FALSE
4	0	TRUE

ID	age	sex	heart_disease
2	25	0	FALSE
3	21	1	FALSE
4	*NA*	0	TRUE

full_join()

data_full <- full_join (data1, data2, by = "ID")

ID	age
1	23
2	25
3	21

ID	sex	heart_disease
2	0	FALSE
3	1	FALSE
4	0	TRUE

ID	age	sex	heart_disease
1	23	*NA*	*NA*
2	25	0	FALSE
3	21	1	FALSE
4	*NA*	0	TRUE

Your turn! Exercise 1

- Create this tibble and store it in tibble1

ID	SES
12	Low
13	Medium
14	High

- Create this tibble and store it in tibble2

ID	htn	IQ
11	yes	102
14	yes	95

- Join the tibbles so that the result look like that:

ID	SES	htn	IQ
11			
12			
13			
14			

Your turn! Exercise 2

- Import the sinai covid dataset
- Import the covid2 dataset

- Join both datasets so that all information is kept in the new dataset

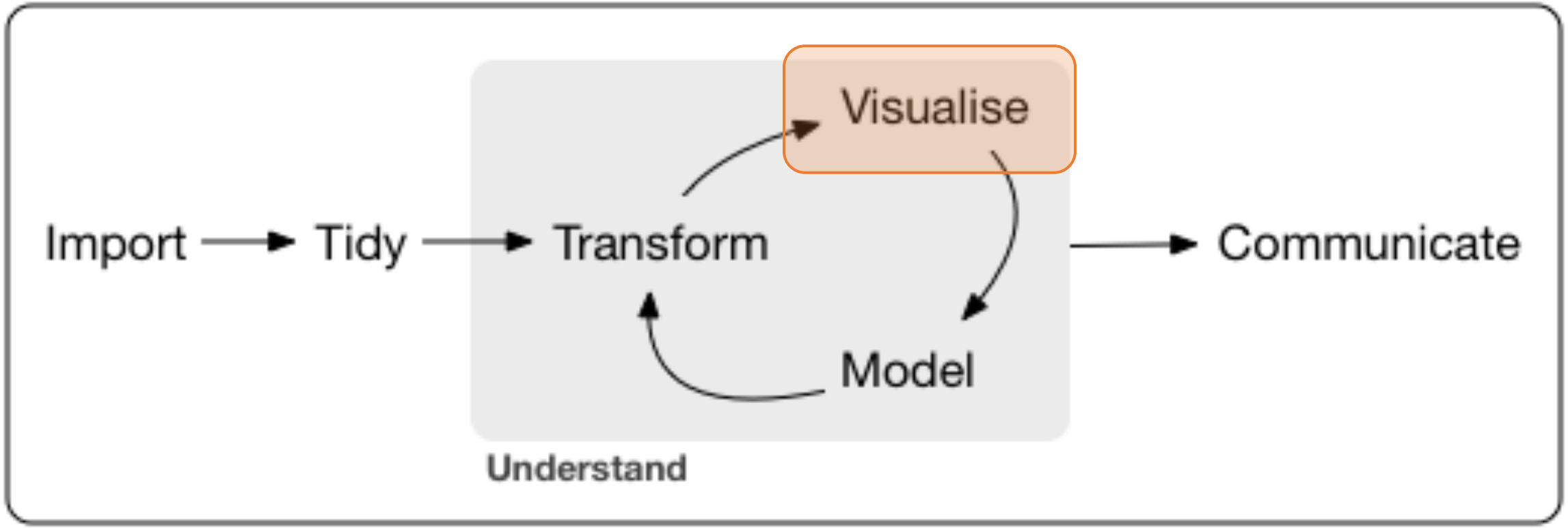
Today

- Other useful functions
  - rename()
  - as.character()/ as.numeric()/ as.factor()
  - recode()
  - glimpse()

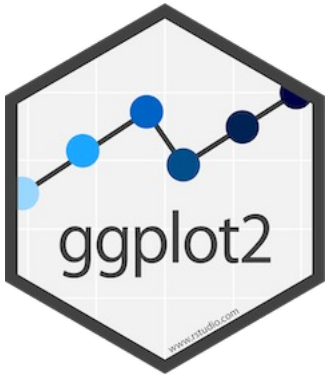
- Transform
  - Join ()

- **Visualize**
  - **ggplot()**
    - **Data**
    - **Aesthetics**
    - **Geometries**

A typical data science project :

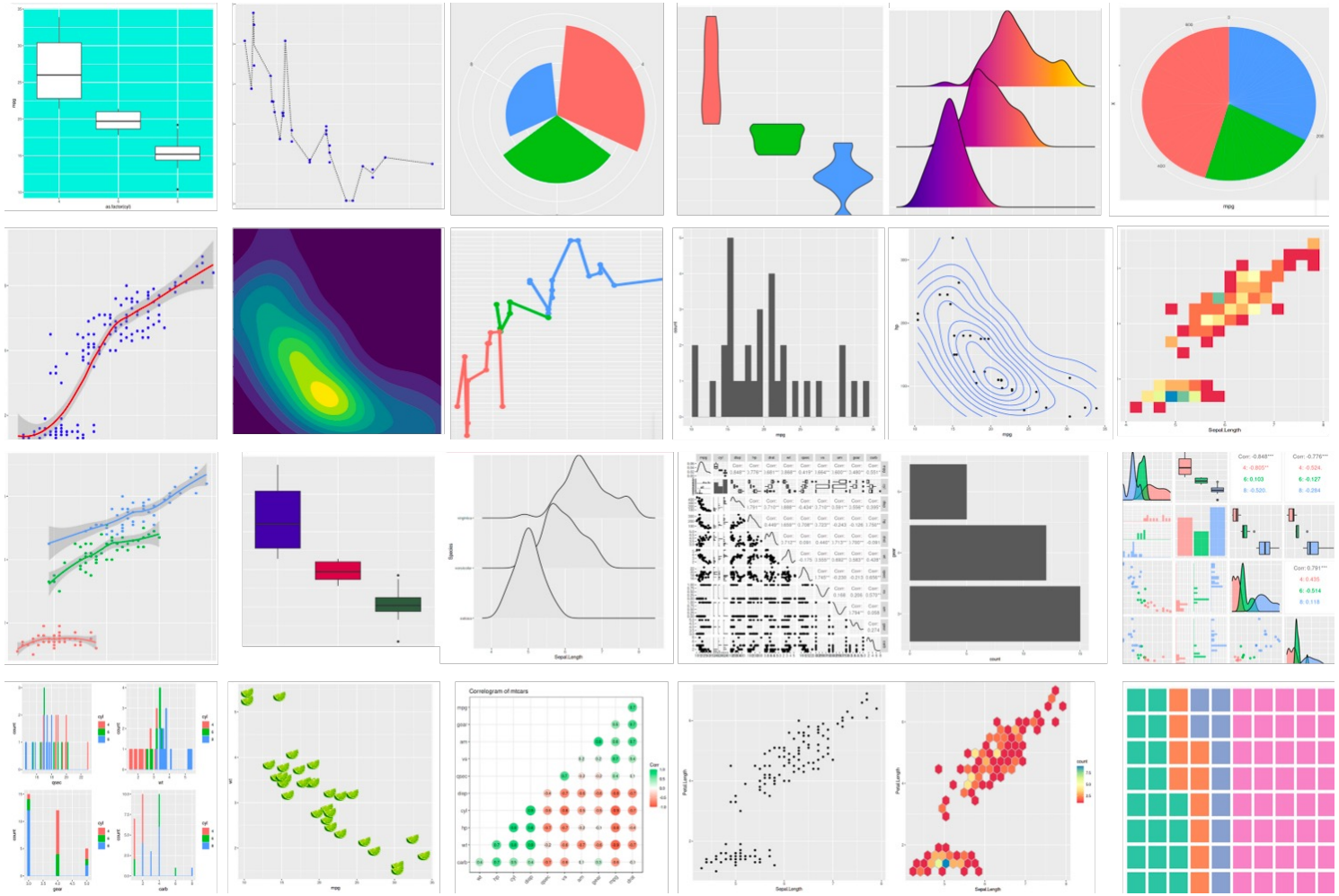


The ggplot2 package



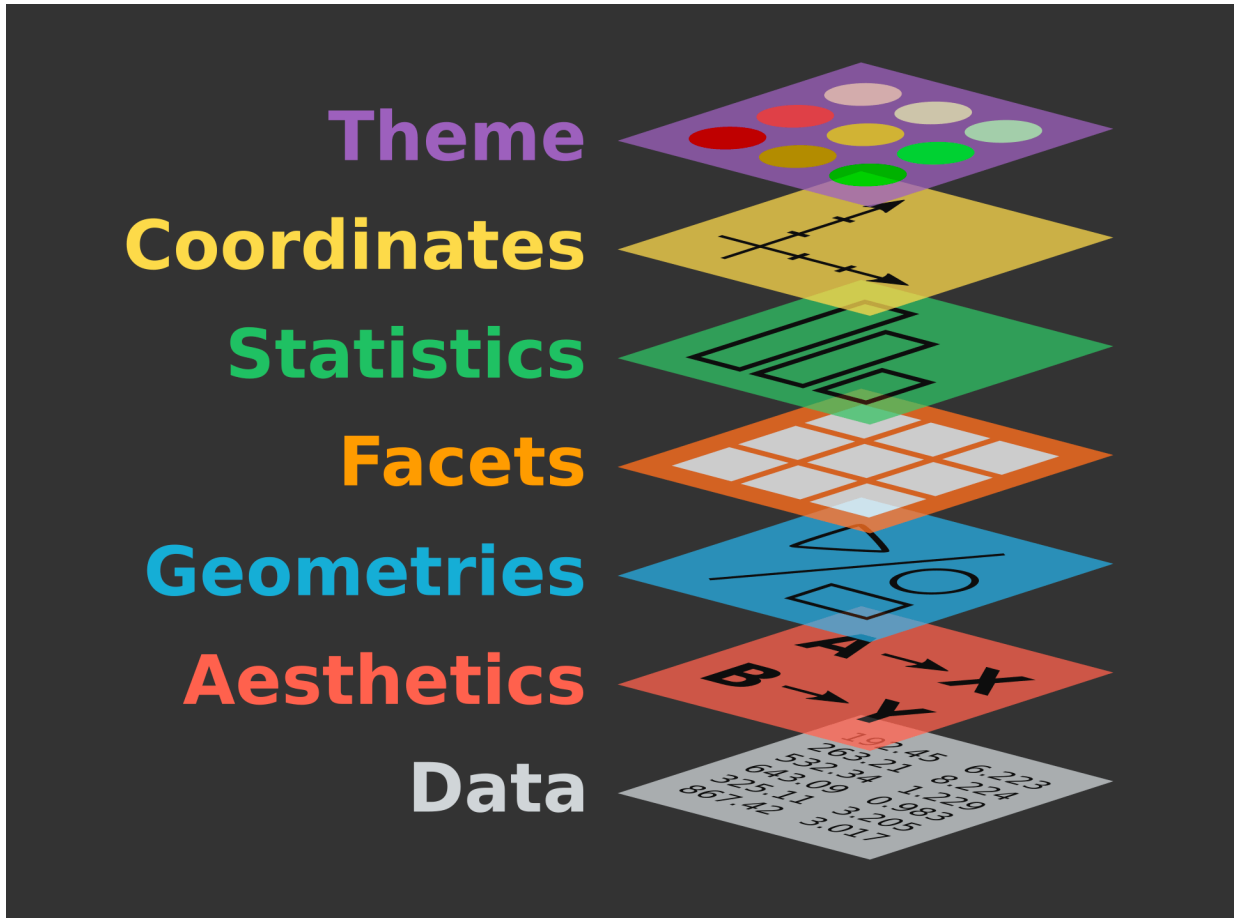
- ggplot2 is a system for creating graphics
- Based on The Grammar of Graphics (Leland Wilkinson, 2000)
- Structured syntax is based on layers
- You provide the data, tell ggplot2 how to map variables to plot, what type of plot to use, and it takes care of the details.

What can ggplot2 do? Highly flexible



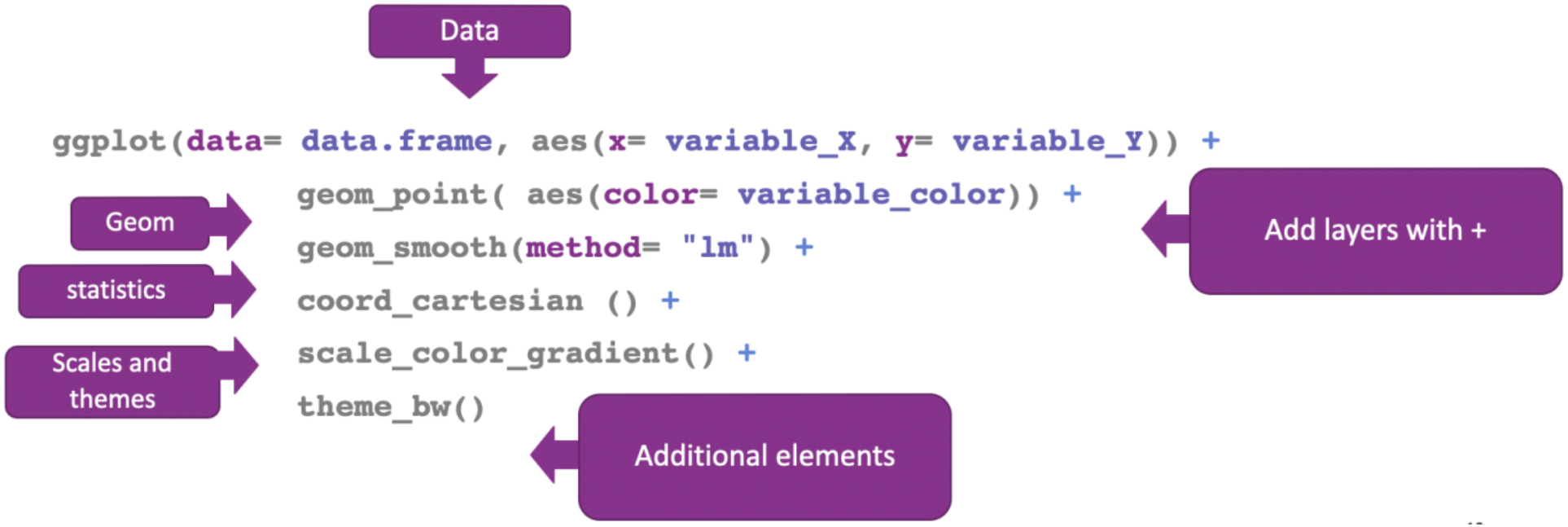
- Image source: https://www.analyticsvidhya.com/blog/2022/03/a-comprehensive-guide-on-ggplot2-in-r/

Major Components



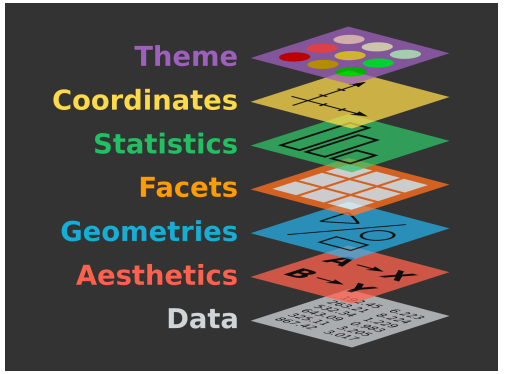
- Image source: https://r.qcbs.ca/workshop03/book-en/grammar-of-graphics-gg-basics.html

ggplot2 syntax



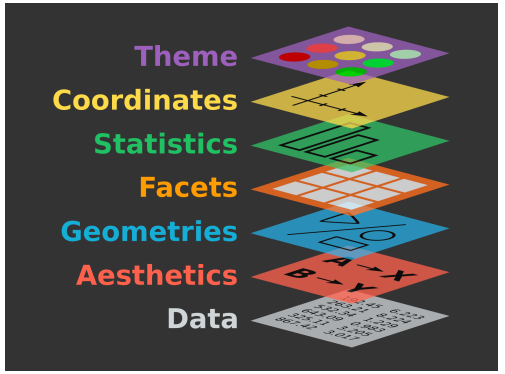
Let’s initialize a basic ggplot based on the sinai covid dataset.

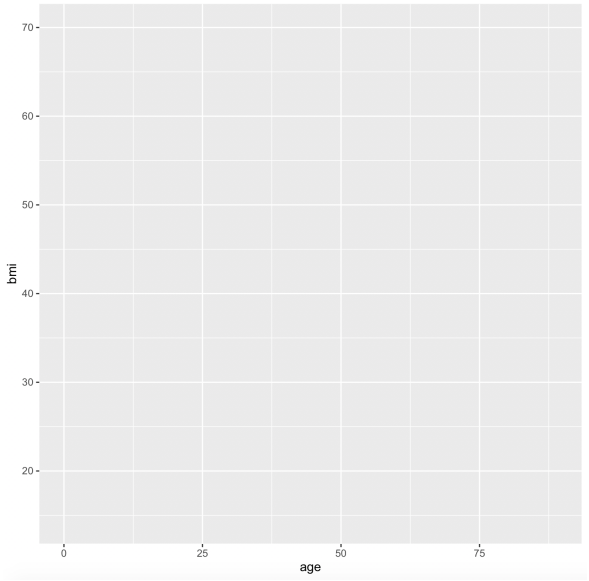
ggplot (data = sinai_covid)



Let’s initialize a basic ggplot based on the sinai covid dataset.

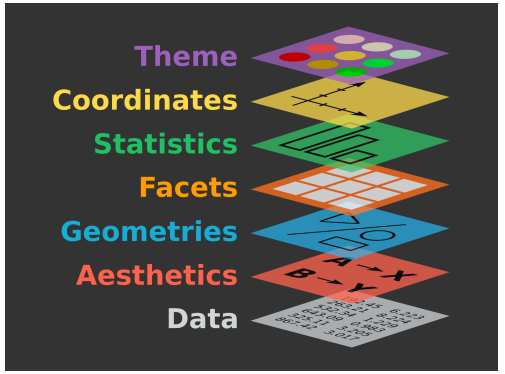
```
ggplot (sinai_covid,
        aes(x = age, y = bmi))
```

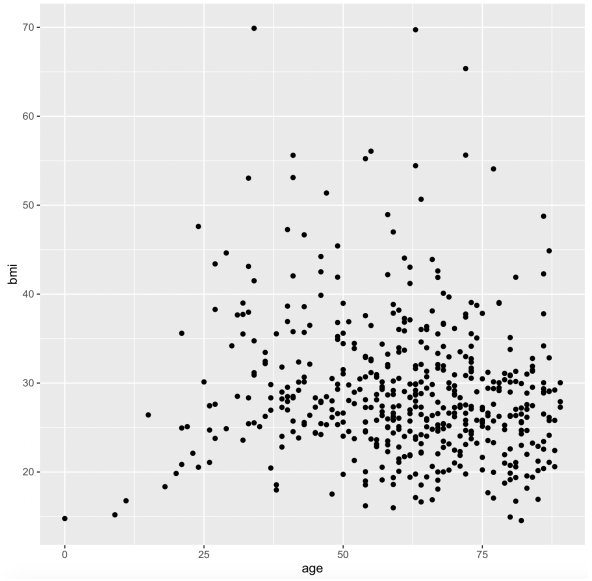




Let’s initialize a basic ggplot based on the sinai covid dataset.

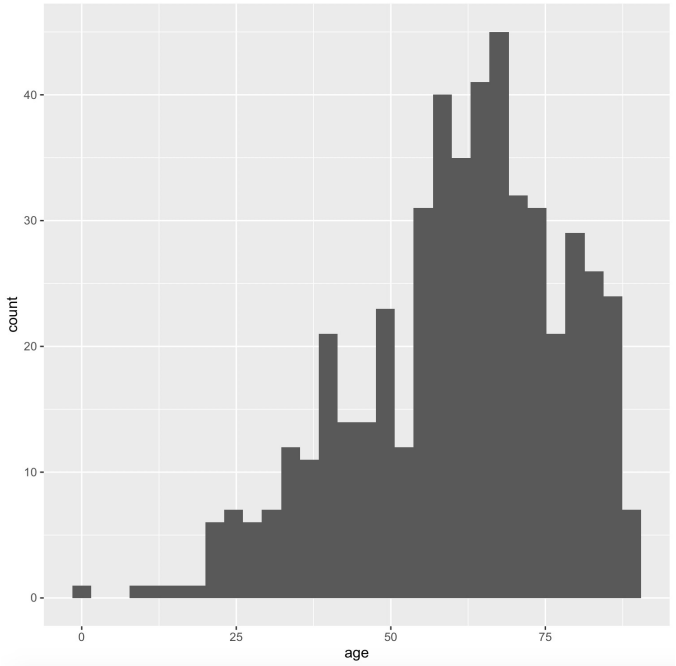
```
ggplot (sinai_covid,
        aes(x = age, y = bmi)) +
        geom_point()
```





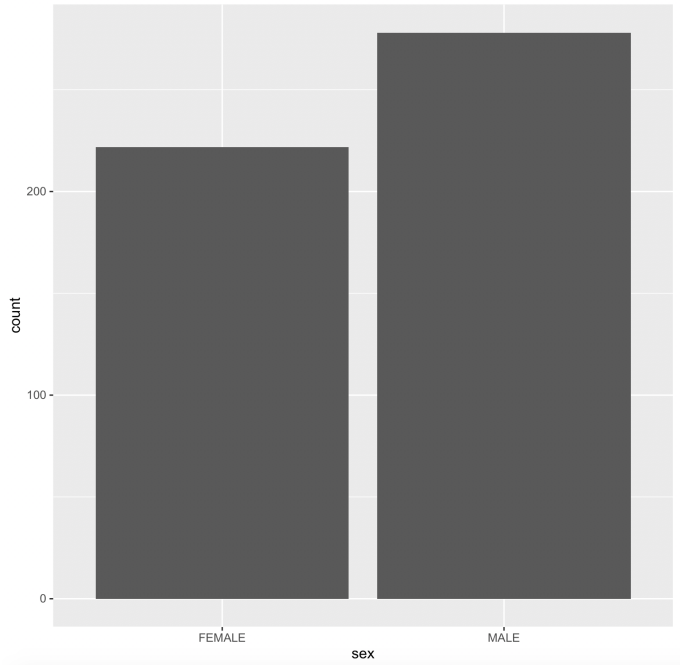
Plotting 1 variable: Numeric variable

```
ggplot(sinai_covid, aes(x = age)) +
  geom_histogram()
```



Plotting 1 variable: Categorical variable

```
ggplot(sinai_covid, aes(x = sex)) +
  geom_bar()
```



Your turn!

Exercise 1

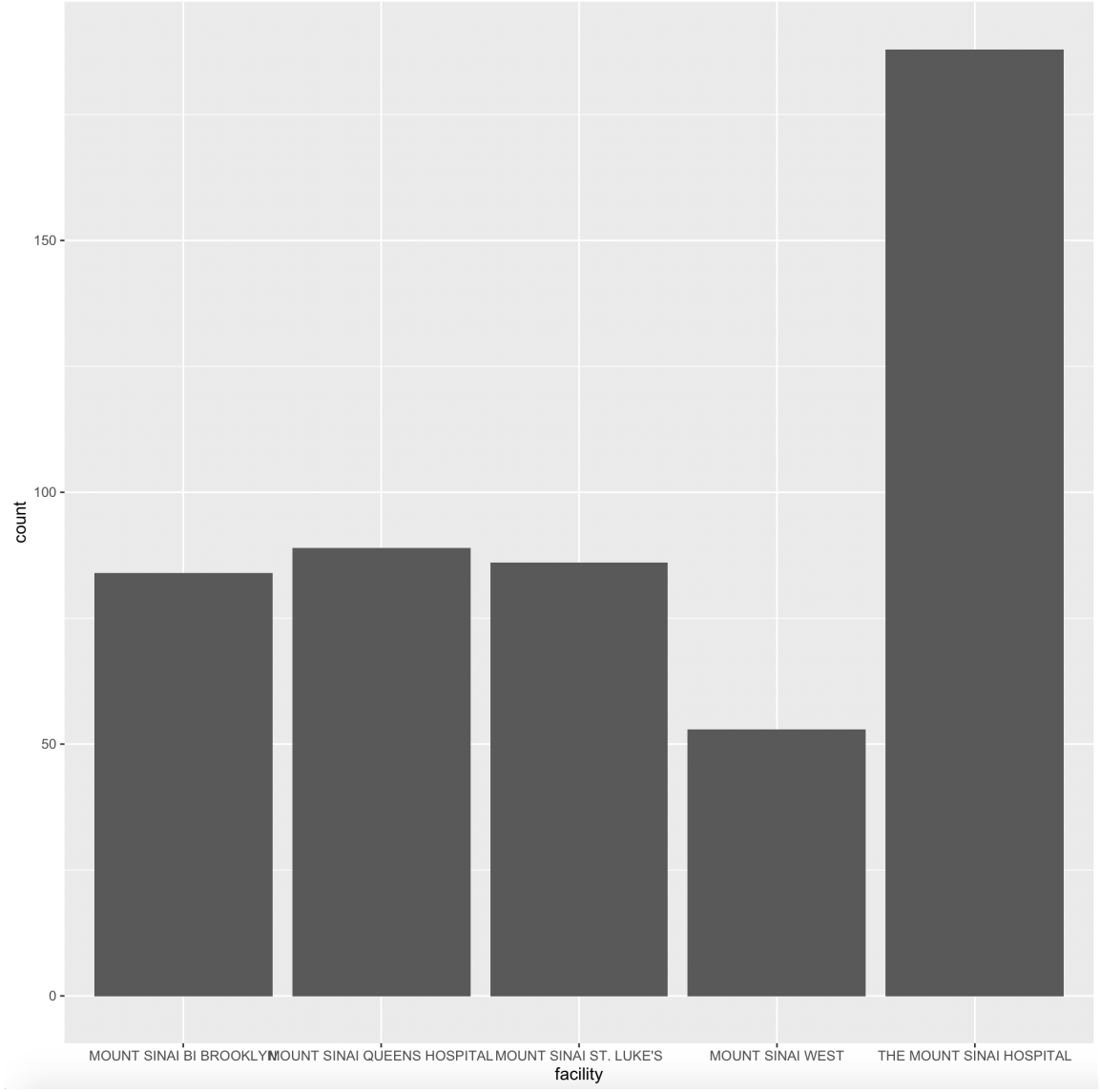
- Plot the distribution of BMI

Exercise 2

- Create a plot showing the number of patients who are smokers, non-smokers, or quit

Your turn! Exercise 3

- Recreate this plot:



Plotting 2 variables: Numeric - Numeric

```
ggplot (sinai_covid,
        aes(x = age, y = systolic_bp)) +
        geom_point()
```

Plotting 2 variables: Categorical - Numeric

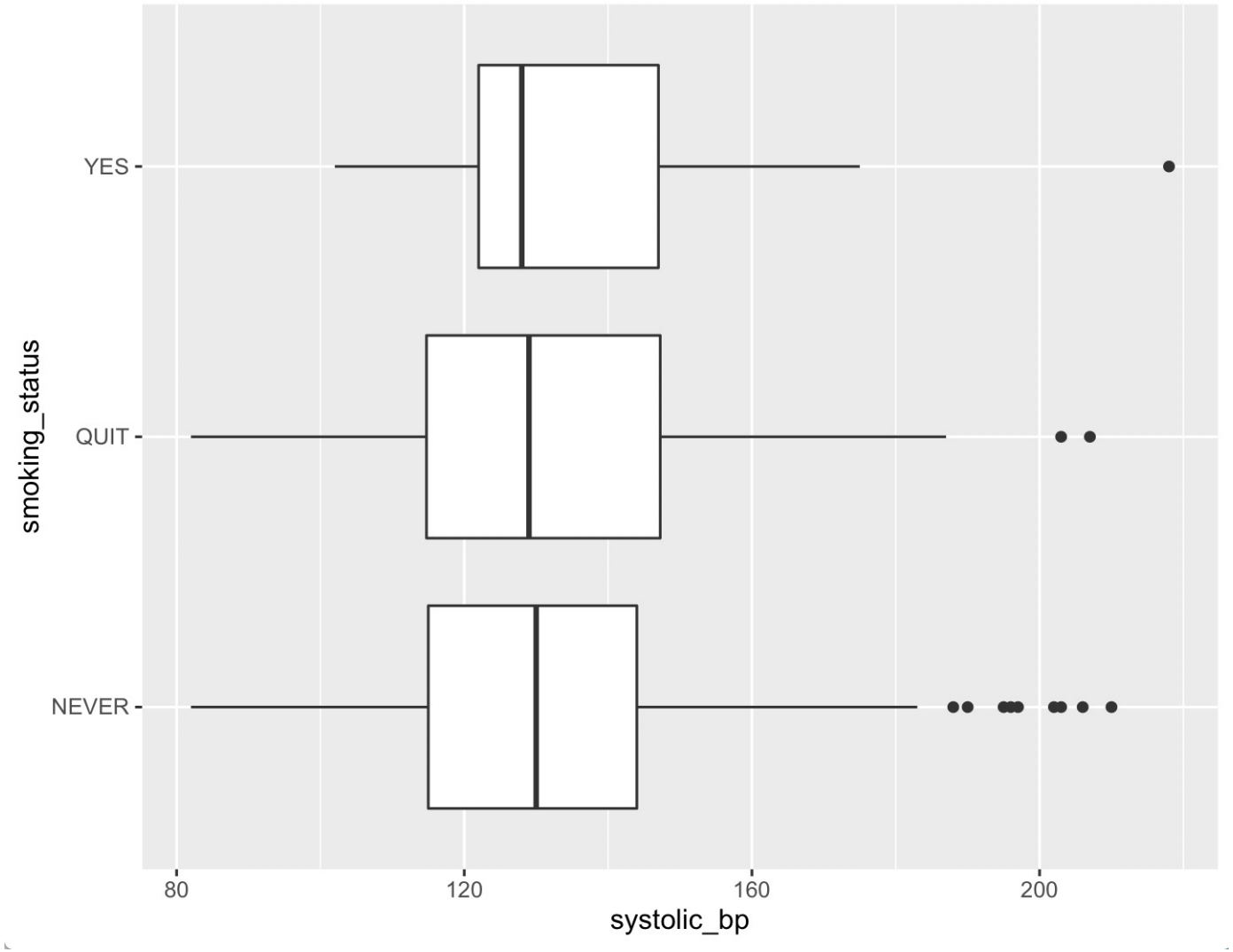
```
ggplot (sinai_covid,
        aes(x = age, y = smoking_status)) +
        geom_boxplot()
```

Your turn! Exercise 4

- Plot BMI vs diastolic_bp
- Plot systolic_bp vs sex
- Plot age vs IQ
- Plot systolic_bp vs diastolic_bp

Your turn! Exercise 5

- Recreate this plot:

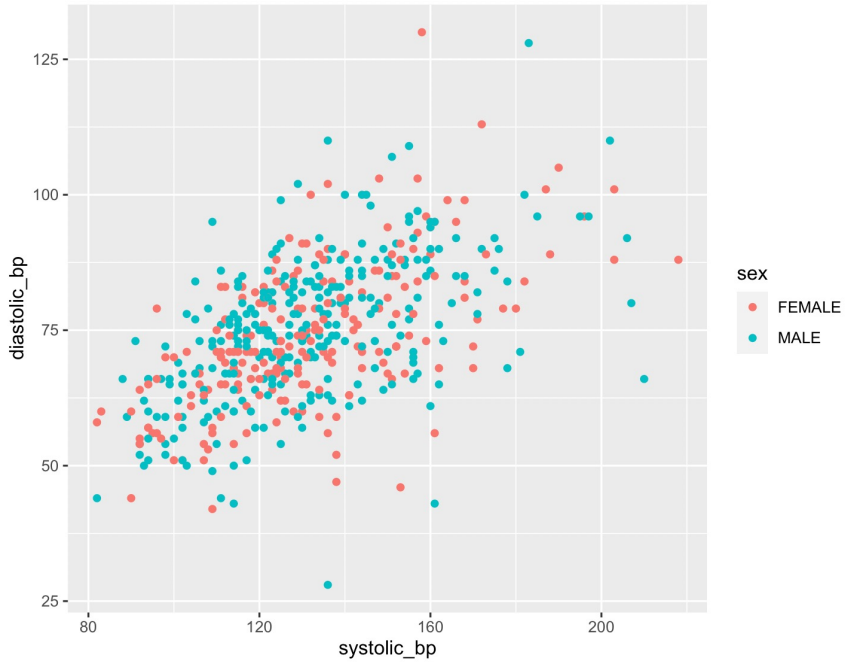


How can we plot 3 variables?

- Color
- Shape
- Size
- Fill
- Alpha (opacity)

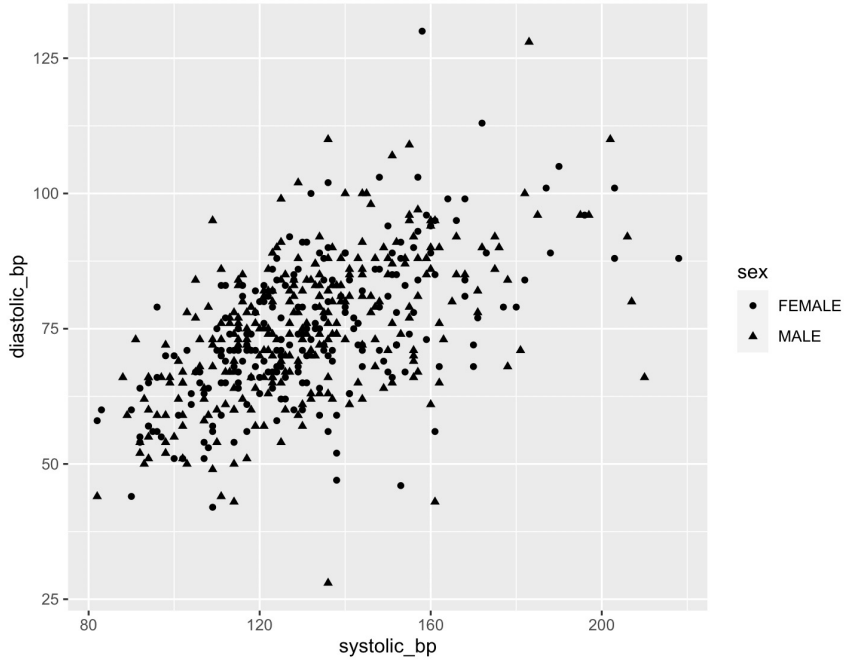
Plotting 3 variables: Color

```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = diastolic_bp, color = sex)) +
  geom_point()
```



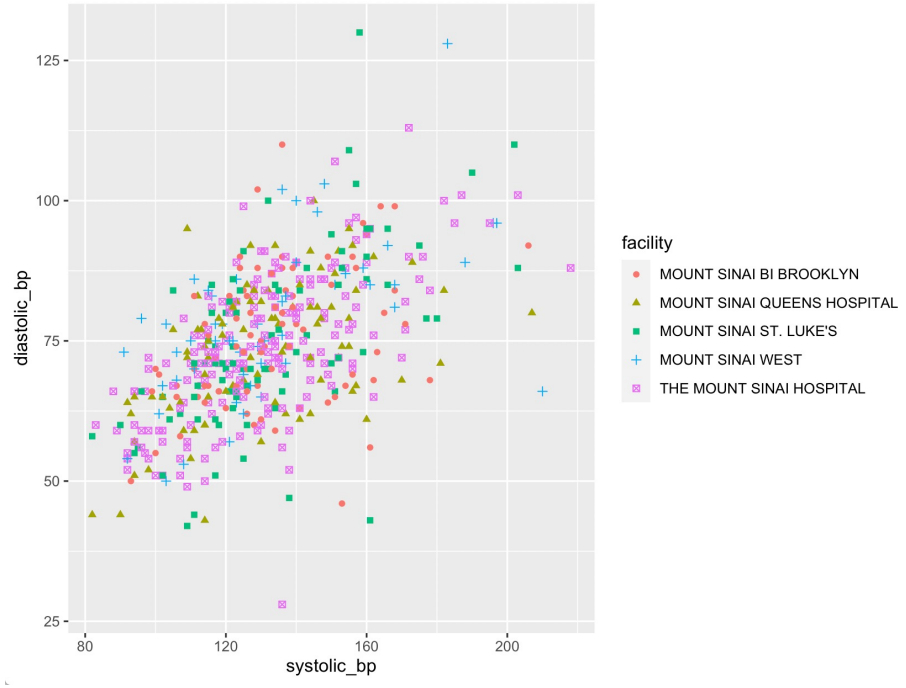
Plotting 3 variables: shape

```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = diastolic_bp, shape = sex)) +
  geom_point()
```



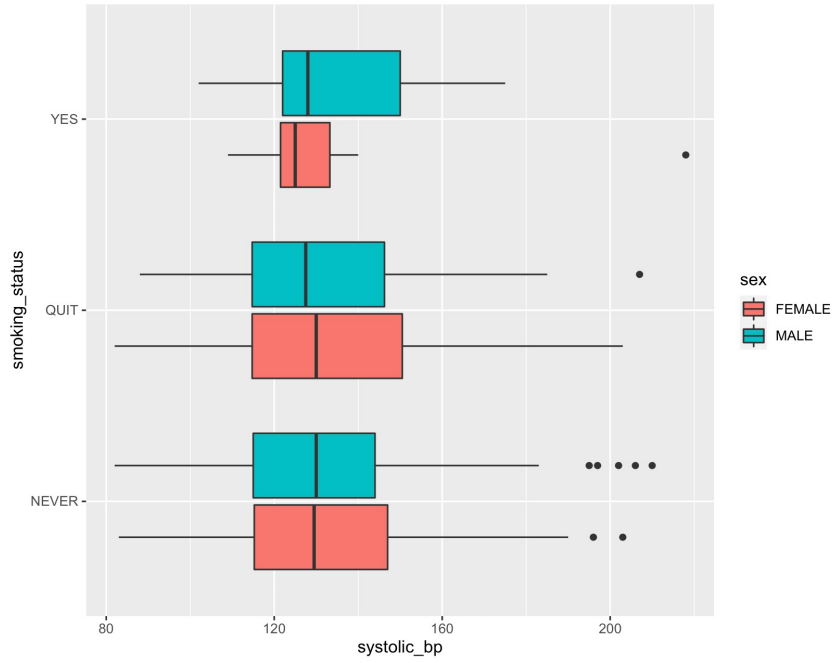
Plotting 3 variables: color + shape

```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = diastolic_bp,
           color = facility, shape = facility)) +
  geom_point()
```



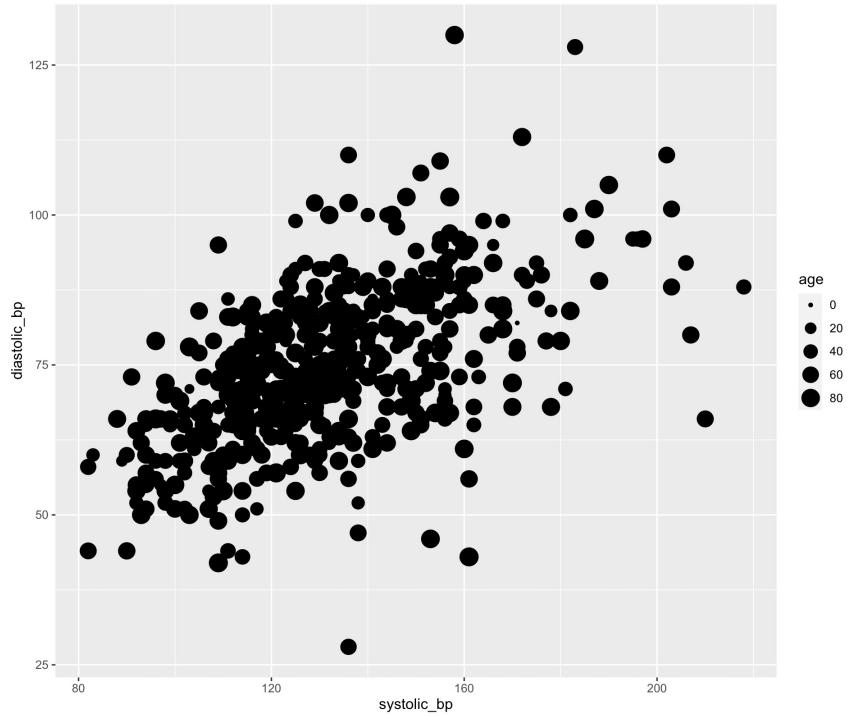
Plotting 3 variables: fill

```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = smoking_status, fill = sex)) +
  geom_boxplot()
```



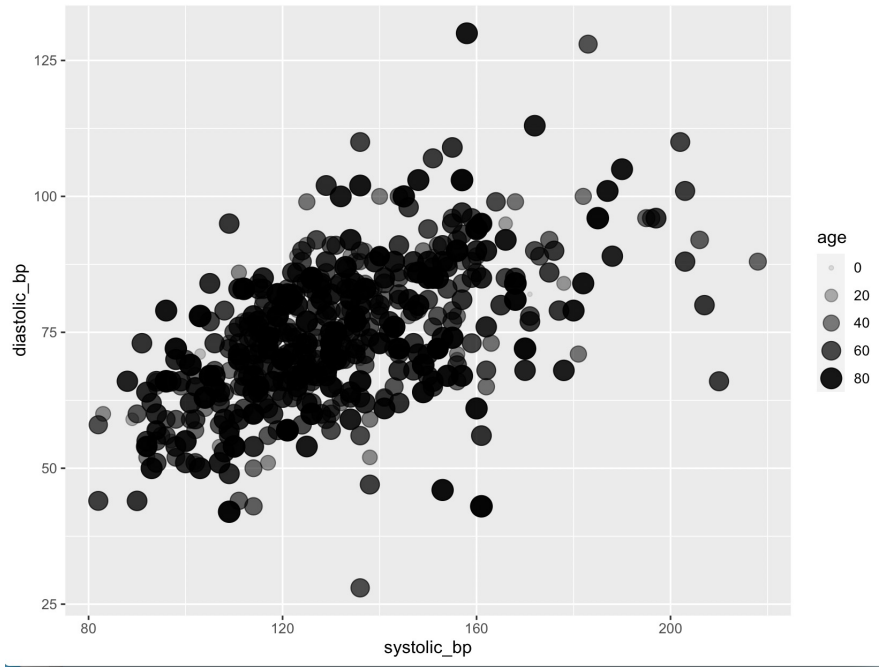
Plotting 3 variables: size

```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = diastolic_bp, size = age)) +
  geom_point()
```



Plotting 3 variables: alpha

```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = diastolic_bp,
           size = age, alpha = age)) +
  geom_point()
```



Your turn! Exercise 6

- Plot the relationship between age (x) and Systolic BP (y), and indicate the smoking status with color

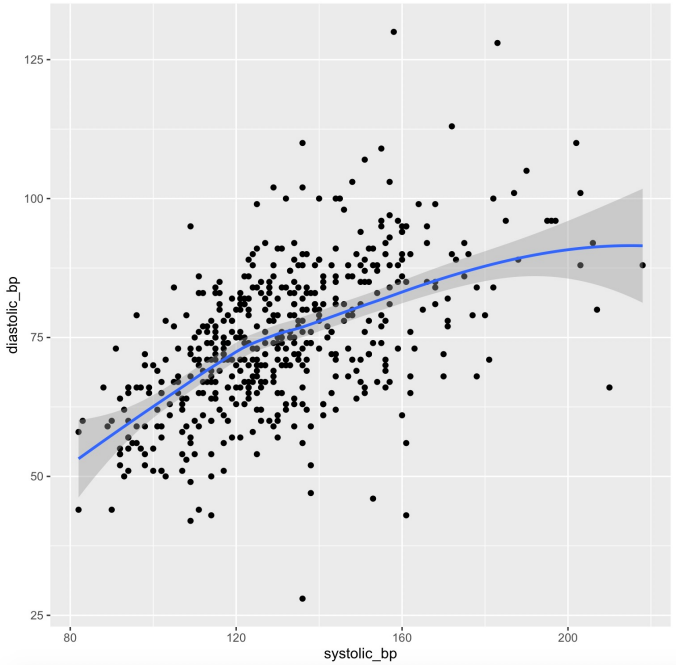
Your turn! Exercise 7

- Plot the relationship between sex (x) and BMI(y), and color the the boxplot by smoking status

Adding a smooth line fitted to the data

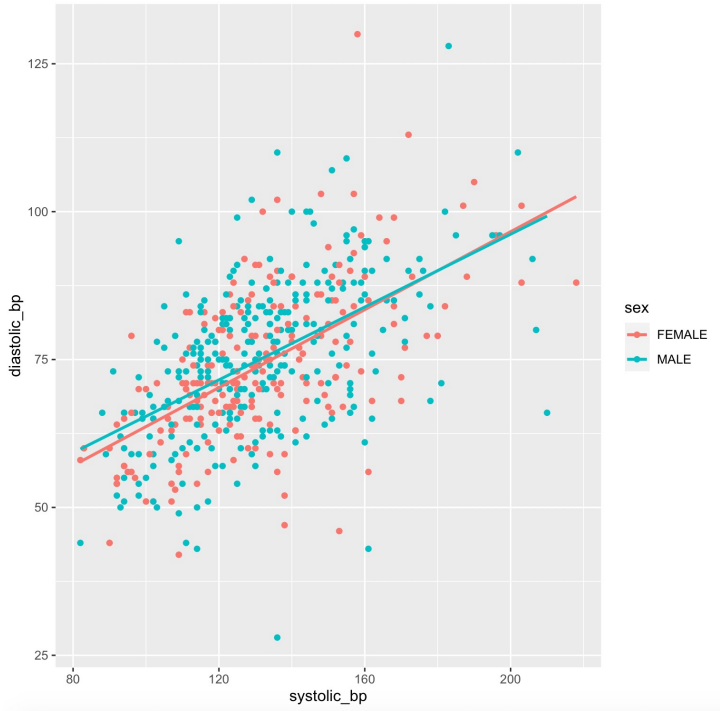
- It can be hard to view trends with just points alone. Many times we wish to add a smoothing line in order to see what the trends look like.
- geom_smooth ()

```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = diastolic_bp)) +
  geom_point() +
  geom_smooth()
```



Adding a smooth line + color

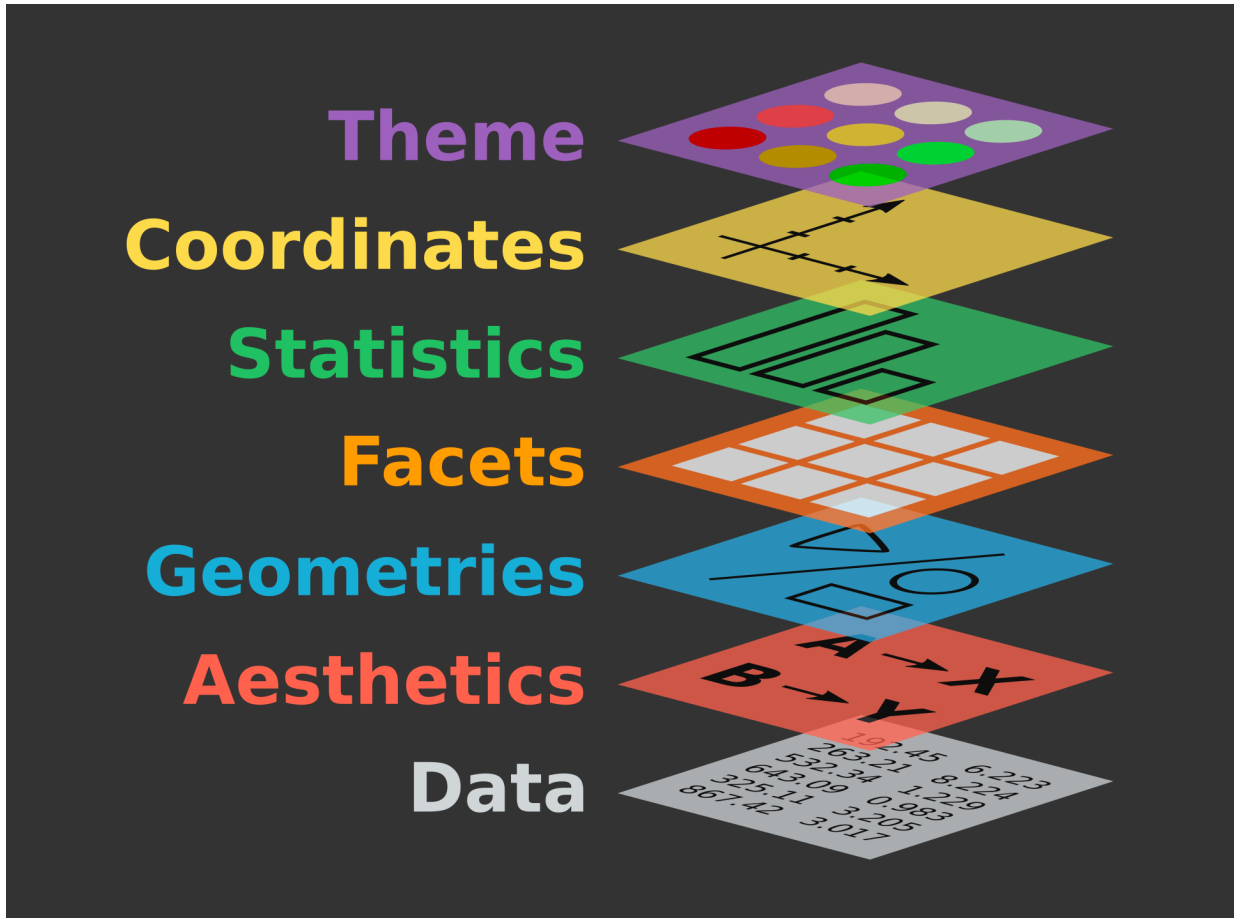
```
ggplot(sinai_covid,
       aes(x = systolic_bp, y = diastolic_bp, color = sex)) +
  geom_point() +
  geom_smooth(method = lm, se = F)
```



Your turn! Exercise 8

- Plot the relationship between systolic_bp and diastolic_bp, adding a linear fitted line, without displaying the confidence interval
- Add color by facility

Major Components



- Image source: https://r.qcbs.ca/workshop03/book-en/grammar-of-graphics-gg-basics.html

Final project

- Pairs
- 10 minutes presentation + 5 minutes for questions
- Import, tidy, transform, visualize, and model
- Judges