R graphics using ggplot2

KY Park

2018 3 22

Introduction

R graphic system

- base graphic system:
  - plot(x, y), barplot(height), boxplot(formula, data)
  - lines(), axis(), points(), text(), title(), abline(), mtext()
- lattice: xyplot(), cloud()
- ggplot2
- ggvis, rCharts, googleVis, rbokeh, htmlwidgets

ggplot 2 is an enhanced data visualization package for R.

Create stunning multi-layered graphics with ease

http://docs.ggplot2.org/current

Why ggplot2? Advantages

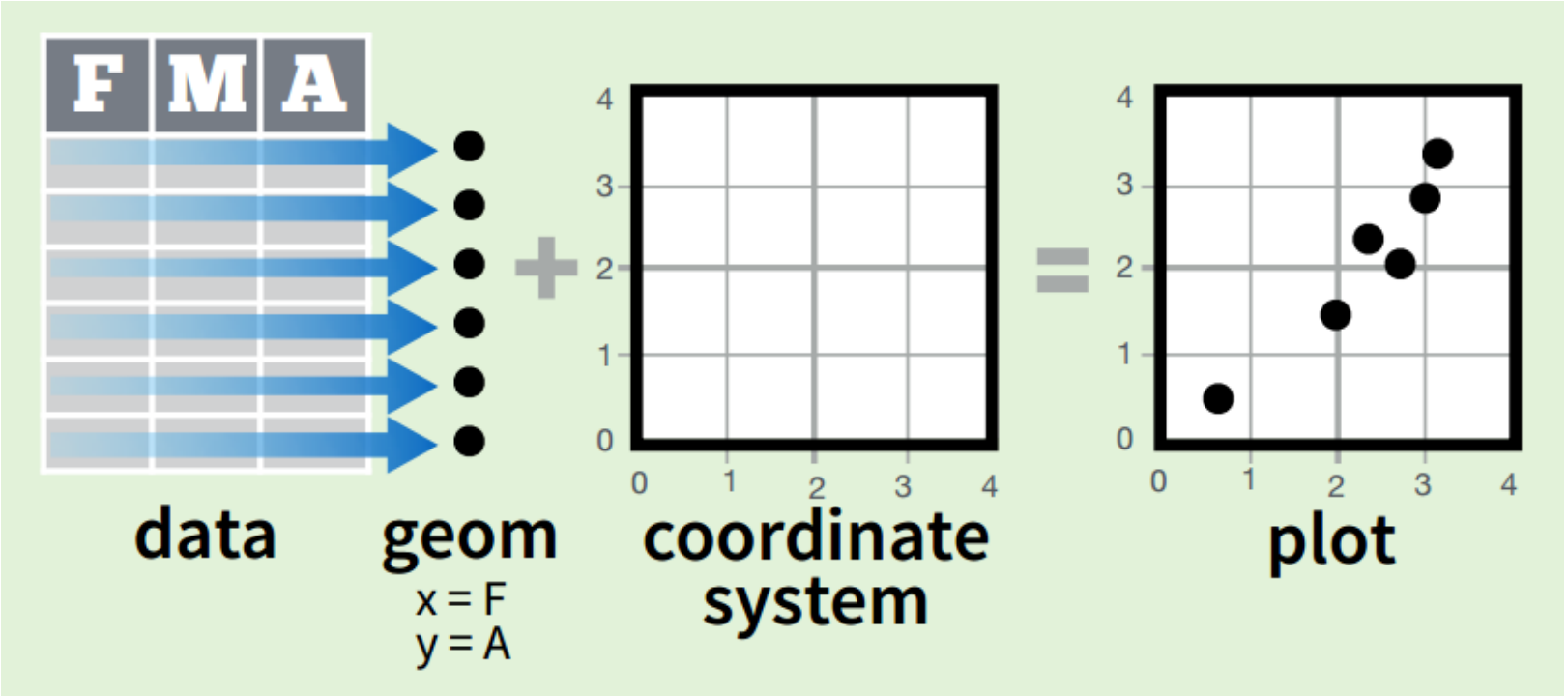
- Based on **Grammar of Graphics** (Wilkinson, 2005)
- Flexible
- Theme system for polishing plot appearance

Grammar of Graphics

- Independent specification of plotting blocks
  - Dataset
  - Aesthetic mapping
  - Geometric object
  - Statistical transformations
  - Scales
  - Coordinate system
  - Position adjustments
  - Faceting

Basics and Geoms I

Basic structure



Preparation

- install and/or update ggplot2 package
- read data file, db/VitDdb_example.csv
- don't forget to check working directory
- check the names of variables
- You can change the names.

```r
# install.packages("ggplot2")
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```r
# getwd()
db <- read.csv("db/VitDdb_example.csv", header=T)

# names(db)
# str(db)
db <- rename(db,
             TOAST = TOAST.classification,
             prev_mRS = mRS.admission,
             NIHSS = NIHSS.admission)
```

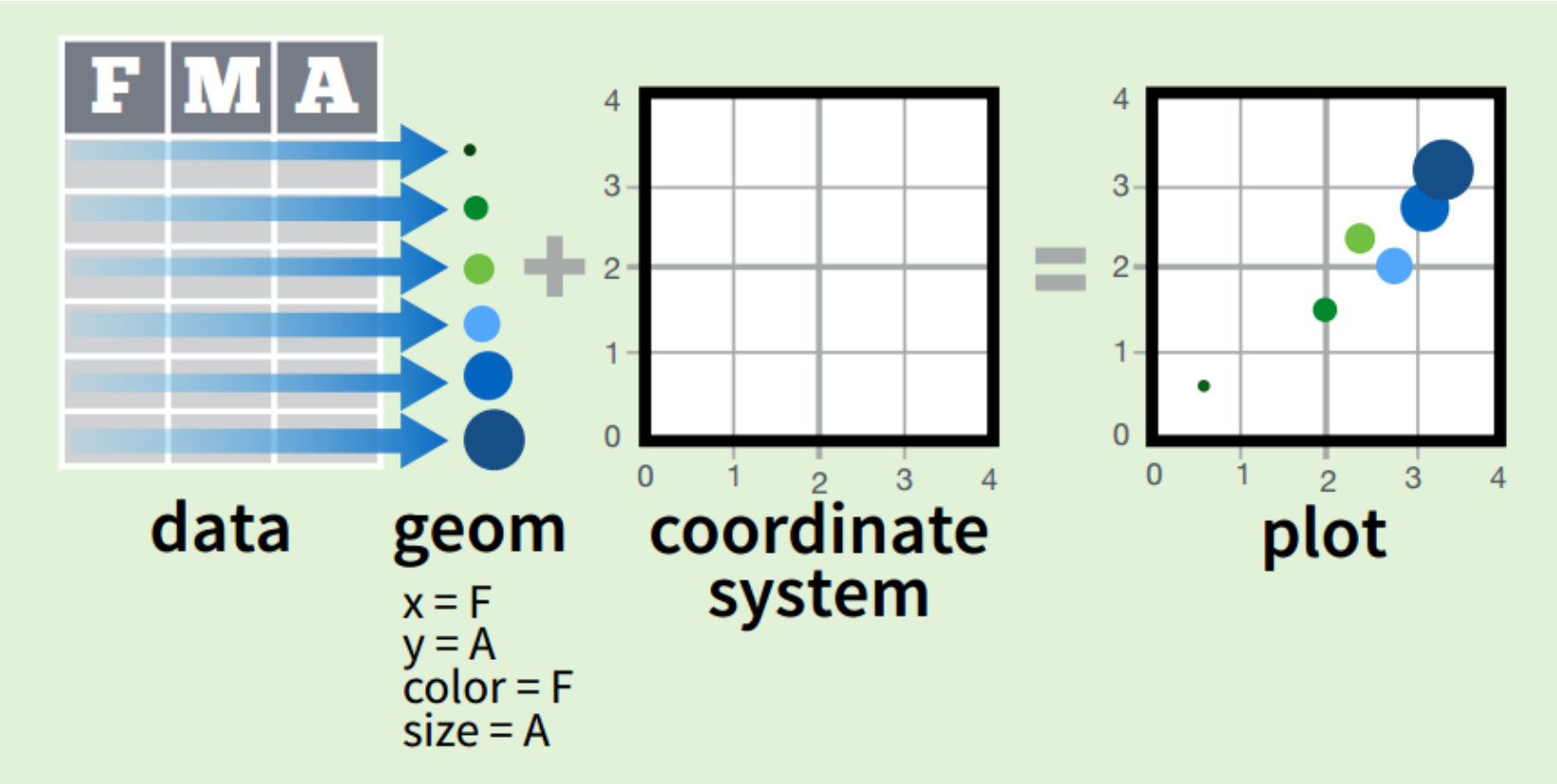
Geometric objects

- Geometric objects are the actual marks we put on a plot
- ~~A plot must have at least one geom~~
- Addition of geoms: +
- Examples
    - geom_point(): scatter plots
    - geom_boxplot()

Aesthetics

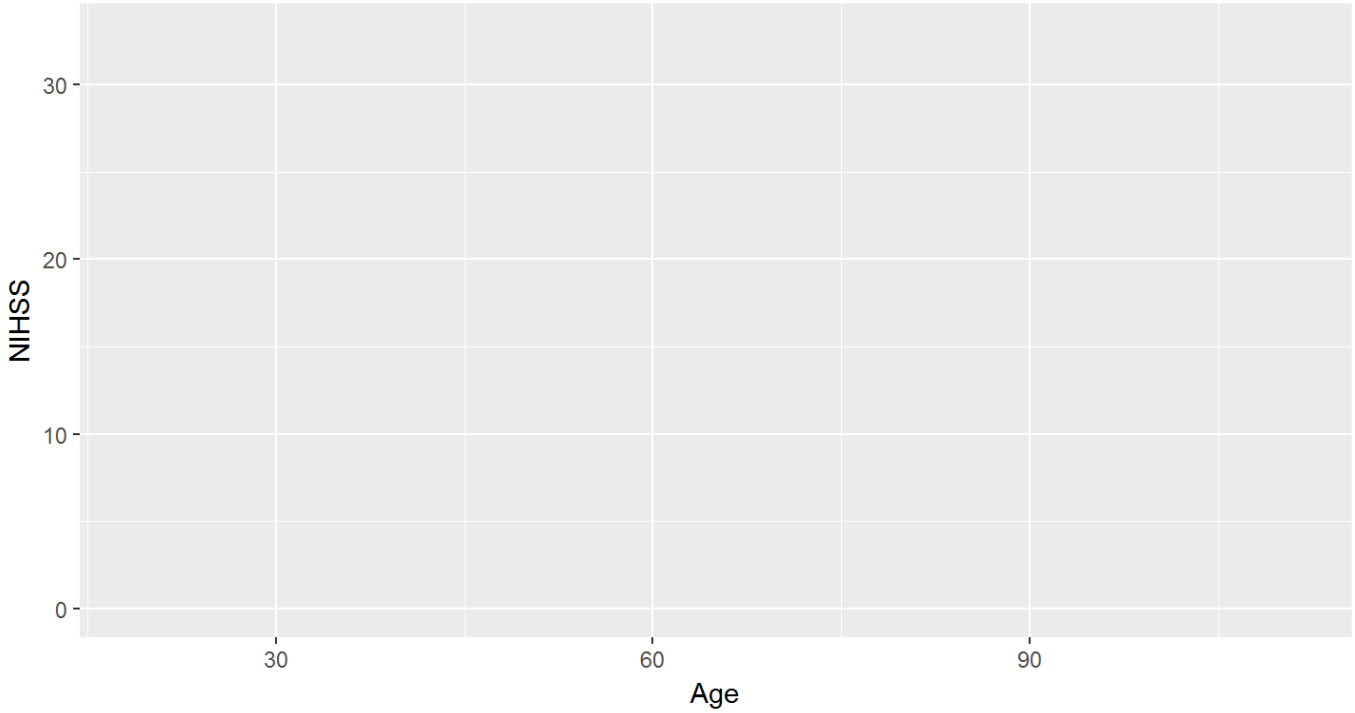
- ggplot(dataset, **aes**(x= , y= , color= , size= , alpha = , fill = , shape =))
- **Aesthetic** means *something you can see*.
    - Examples include:
        - position (on the x and y axes)
        - color (outside color)
        - fill(inside color)
        - shape (of point)
        - linetype
        - size
- Each type of geom accepts only a subset of all aesthetics.

Basic structure 2



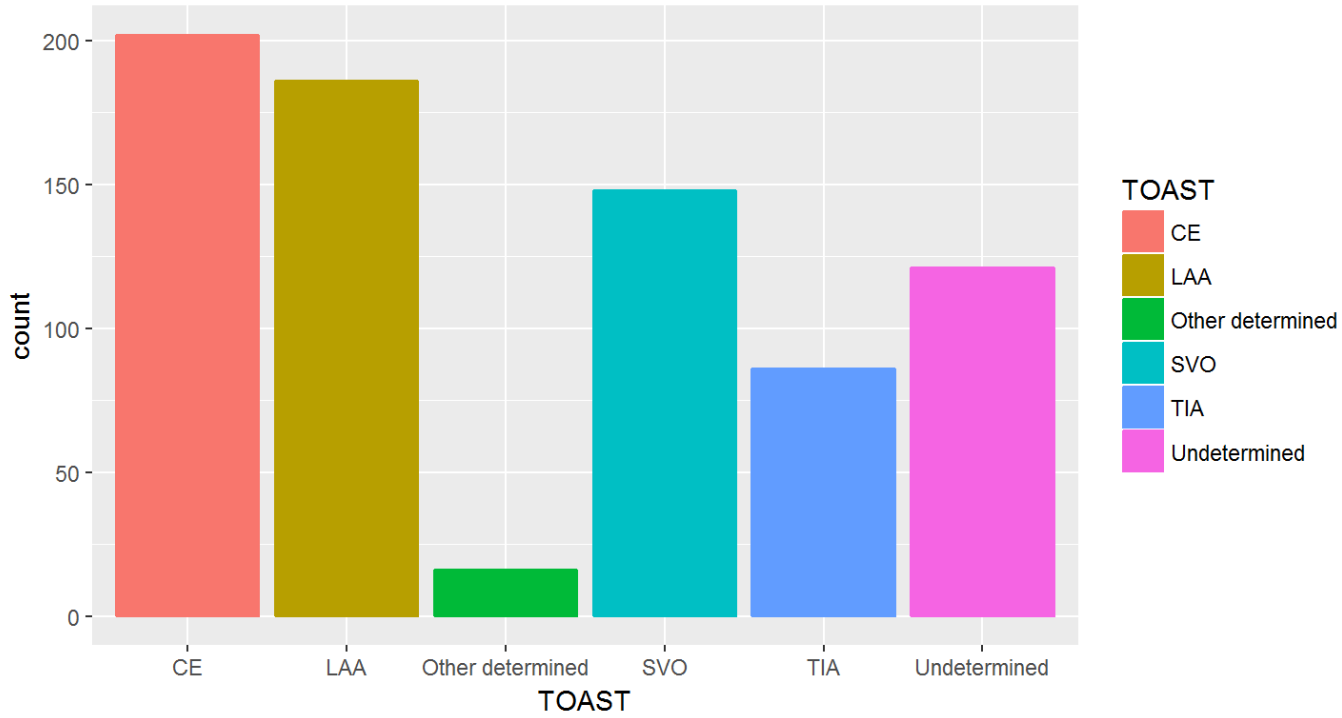
First example

```
p1 <- ggplot(db, aes(x=Age, y=NIHSS))
p1
```



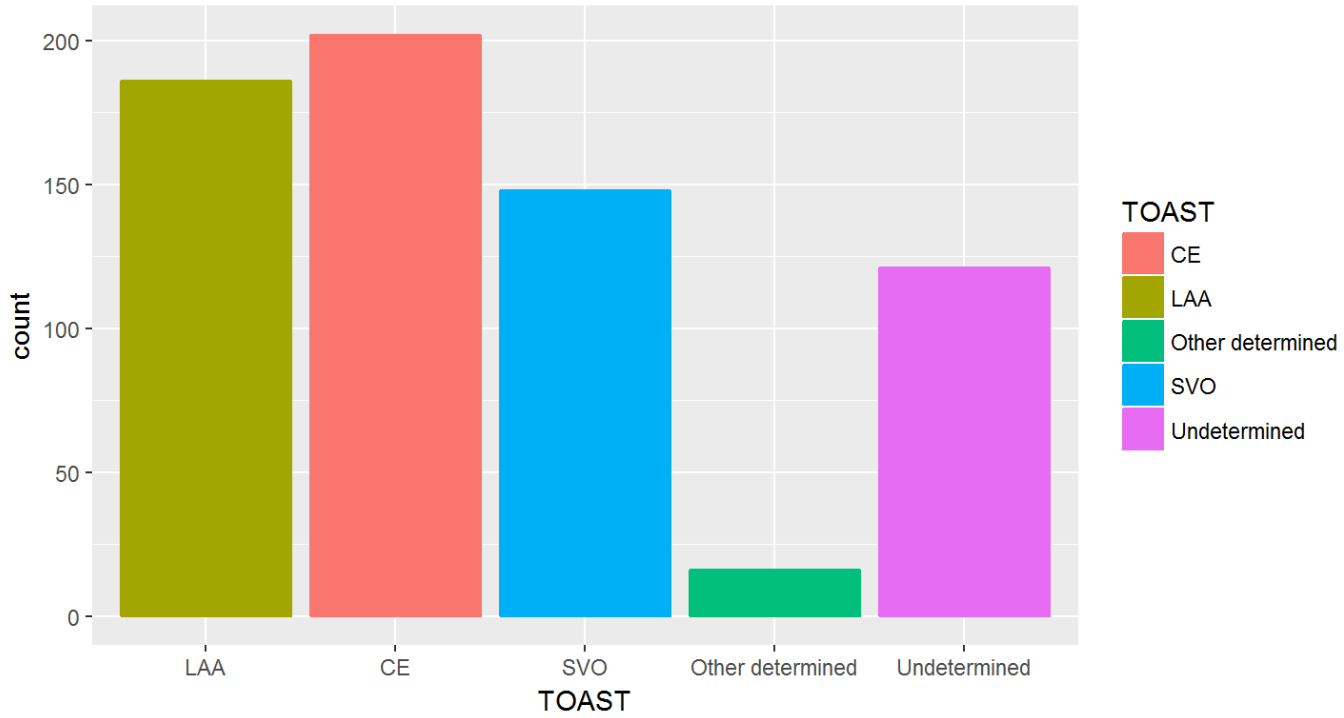
Categorical variable

```
p2 <- ggplot(db, aes(x=TOAST, fill=TOAST, color=TOAST))
p2 + geom_bar()
```



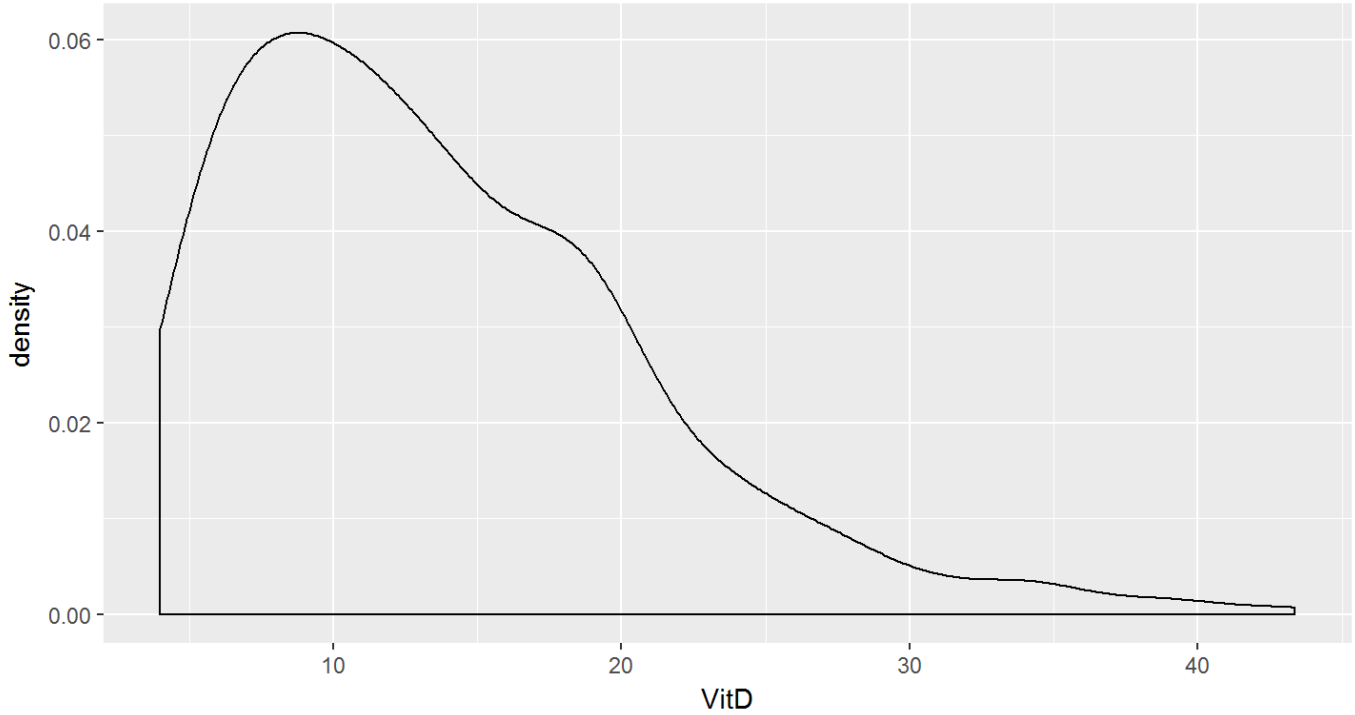
Categorical variable - change the order of items on X-axis

```
p2 + geom_bar() +
  scale_x_discrete(limits=c("LAA", "CE", "SVO", "Other determined", "Undetermined"))
```



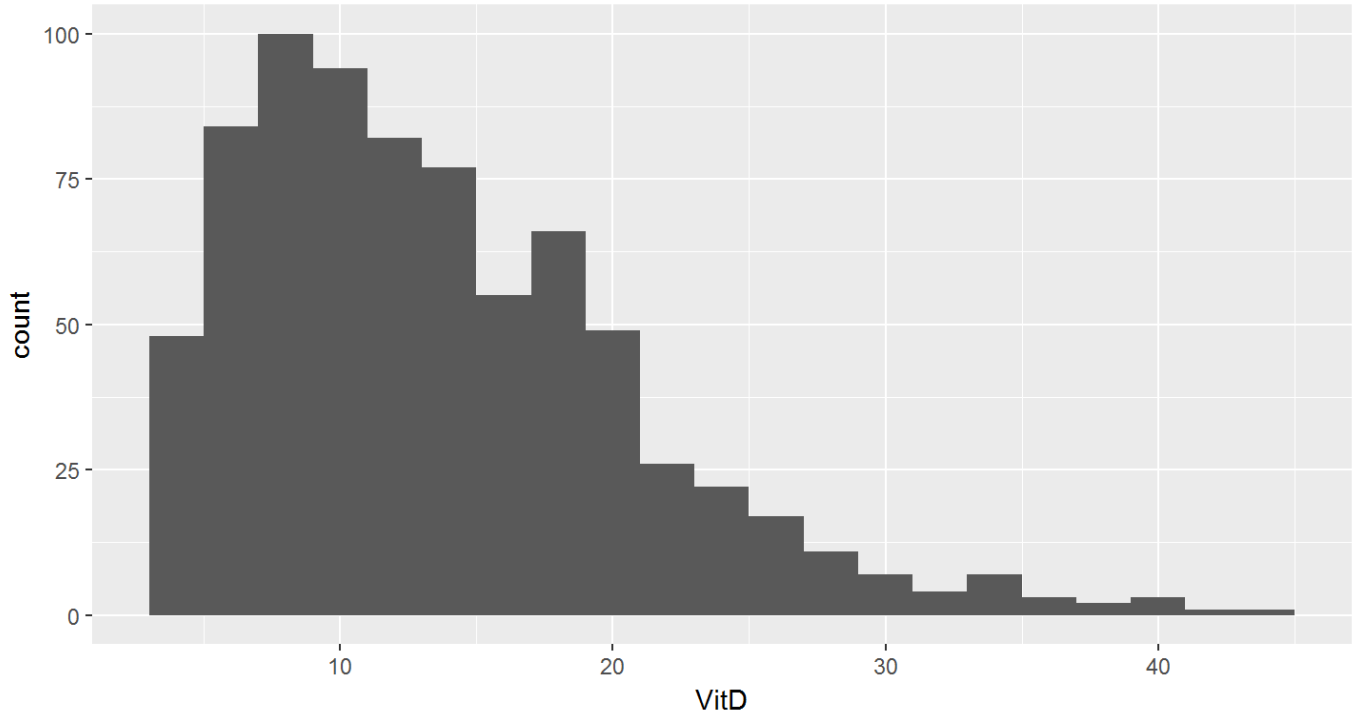
Continuous variable 1

```
p3 <- ggplot(db, aes(VitD))
p3 + geom_density()
```



Continuous variable 2

```
p3 + geom_histogram(binwidth = 2)
```



Save graph - simple

```
ggsave("exercise1.png", width = 5, height = 5)
```

Save graph 2

```
ppi = 300
tiff("Exercise.tiff", width = ppi*8, height = ppi*4, res = ppi)
p2 + geom_bar()
dev.off()
```

```
## png
##   2
```

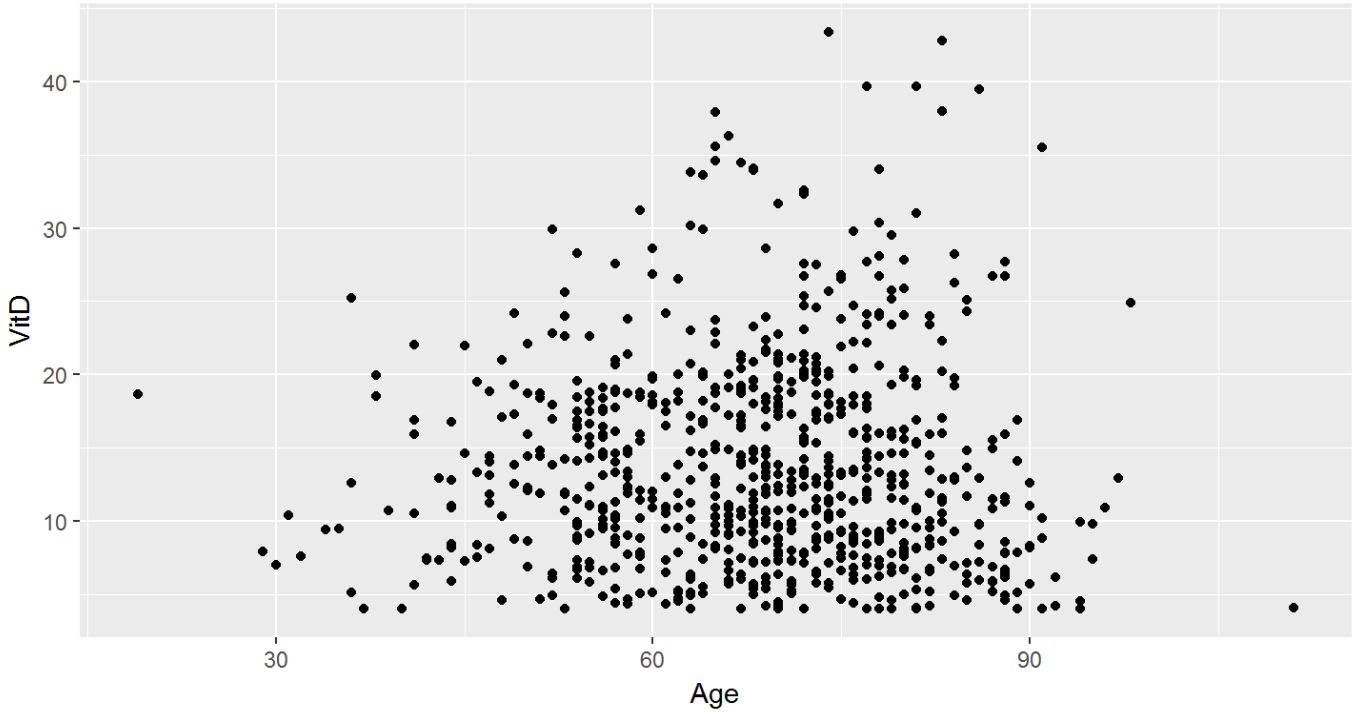
Exercise 1

- Make a presentation file
  - read your own db file
  - make bar plot of gender
  - you can change the order of items on x-axis
  - make histogram of age.

Geoms II

Continuous - continuous 1

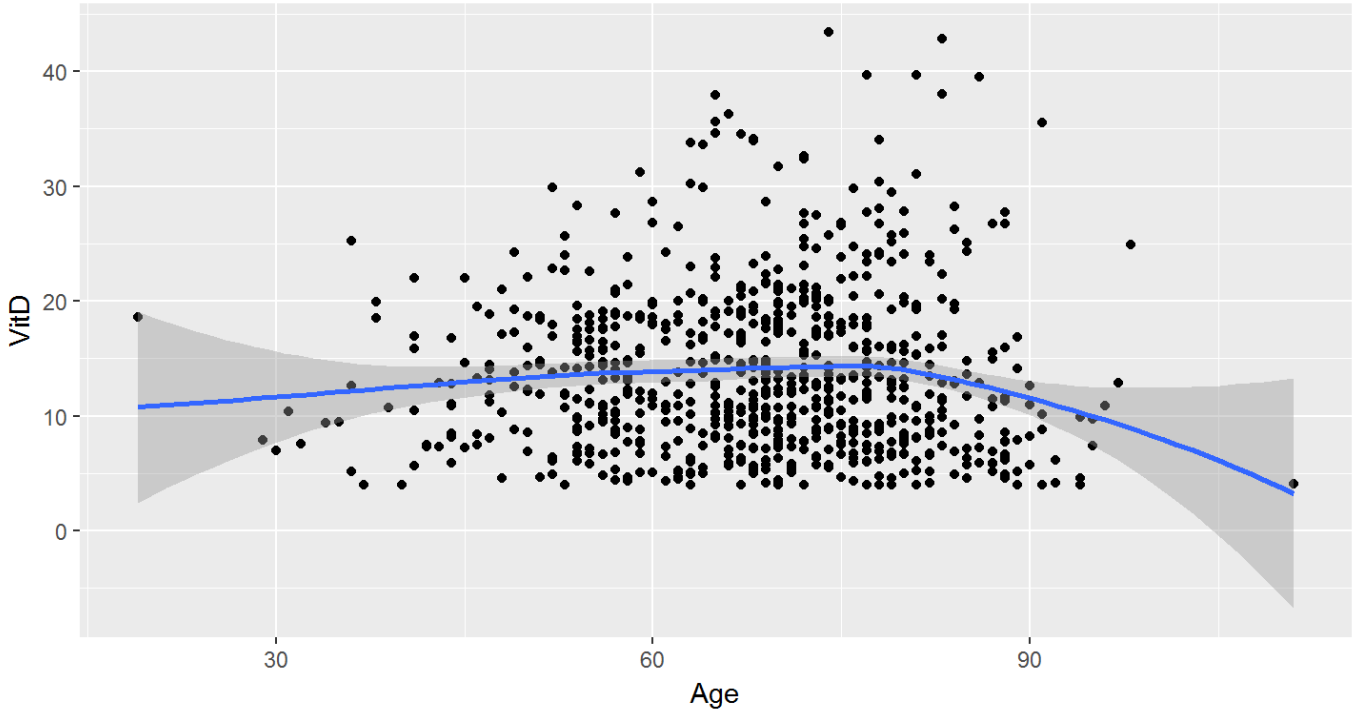
```
p4 <- ggplot(db, aes(x=Age, y=VitD))
p4 + geom_point()
```



Continuous - continuous 2

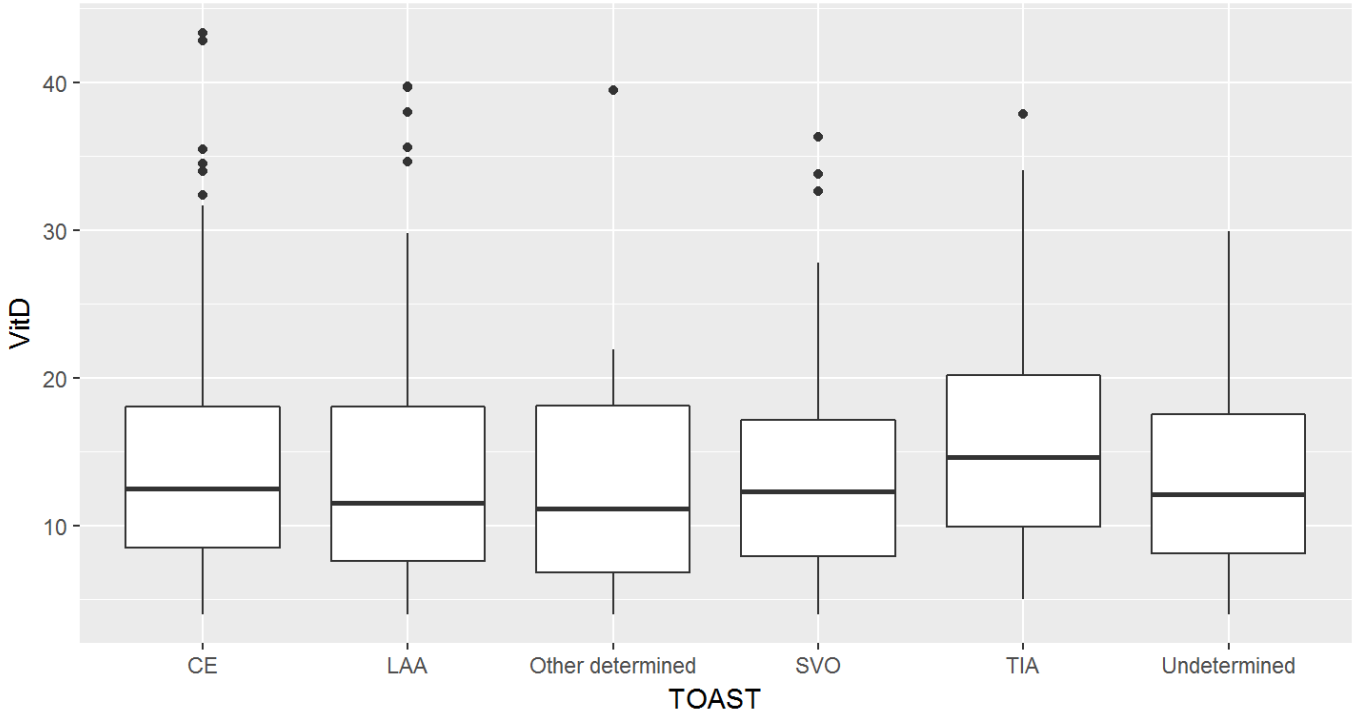
```
p4 + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



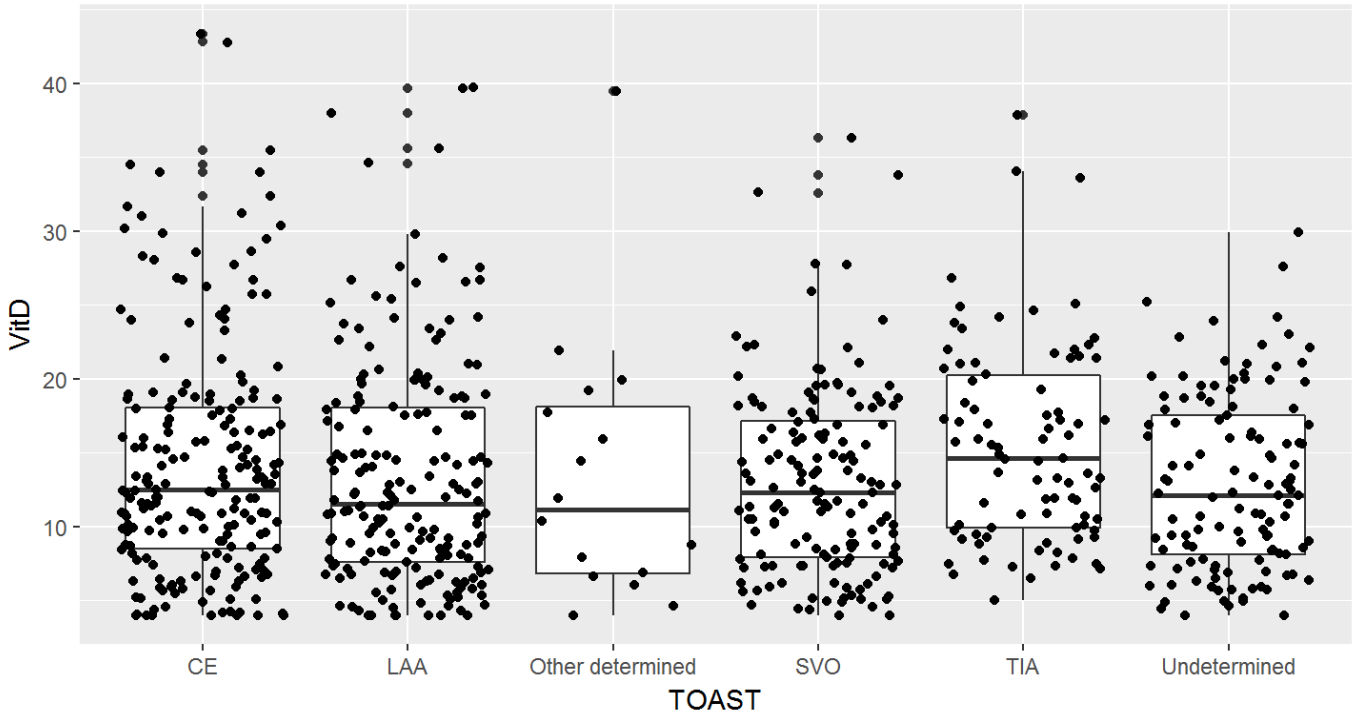
Categorical - continuous 1

```
p5 <- ggplot(db, aes(x=TOAST, y=VitD))
p5 + geom_boxplot()
```



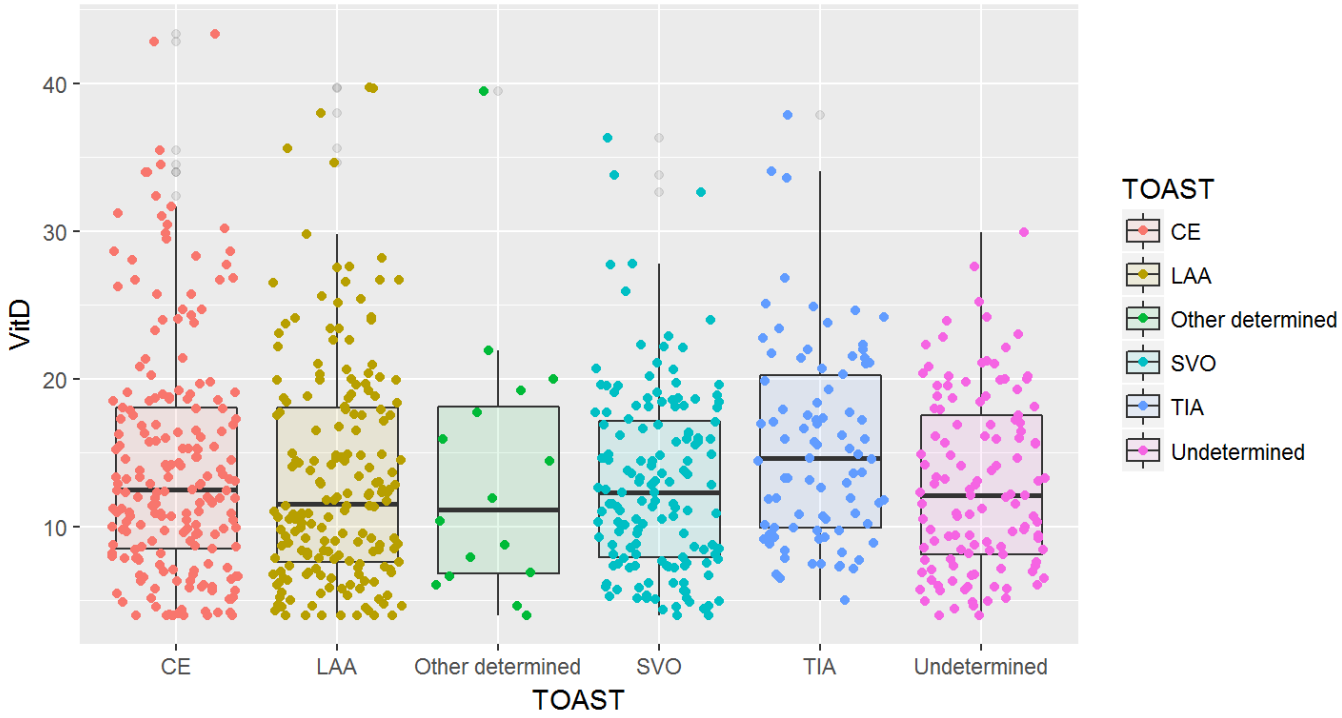
Categorical - continuous 1-1

```
p5 + geom_boxplot() + geom_jitter()
```



Categorical - continuous 1-2

```
p5 + geom_boxplot(aes(fill=TOAST), alpha=0.1) + geom_jitter(aes(color=TOAST))
```



Statistical transformation

Statistical transformation

- Some plot types like scatterplots do not require transformation
- Other plots such as boxplots, histograms, and prediction lines require statistical transformation

Scale

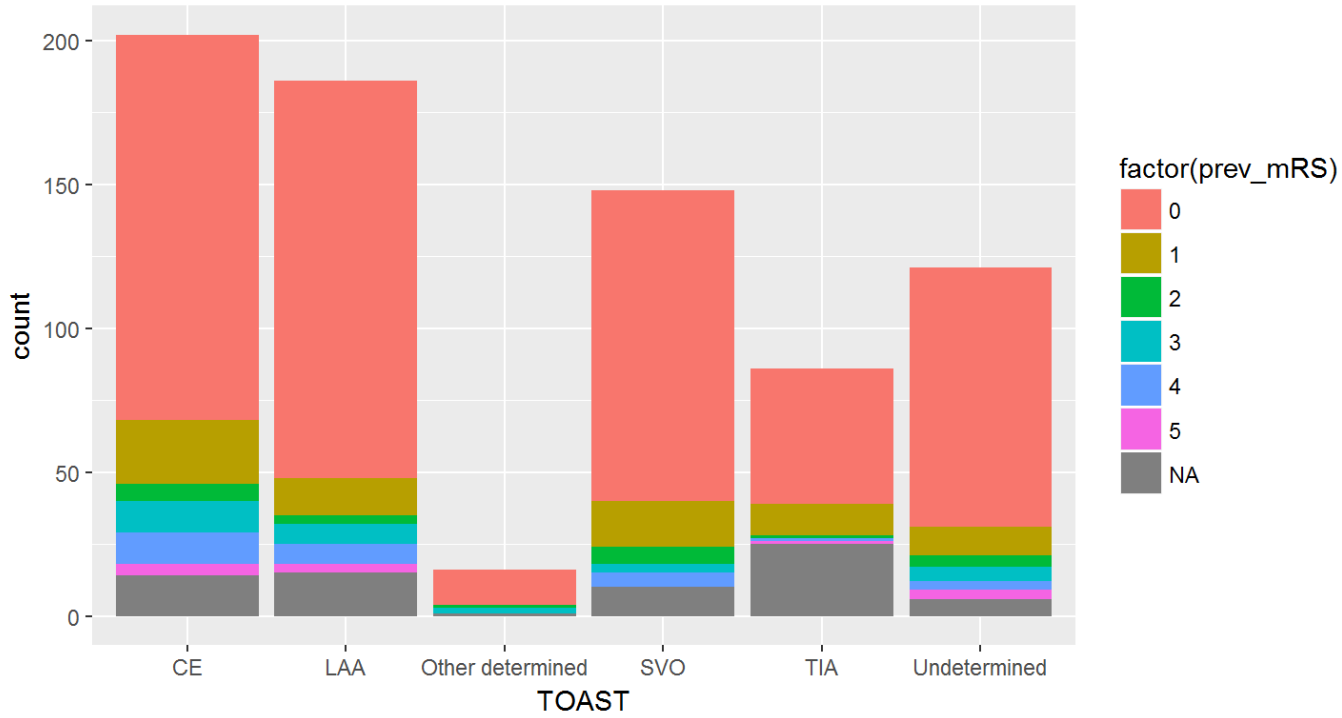
Scale

- Include
  - position, color, fill, size, shape, line type
- Arguments
  - name, limits, breaks, labels
- Example
  - scale_*aesthetic_type*

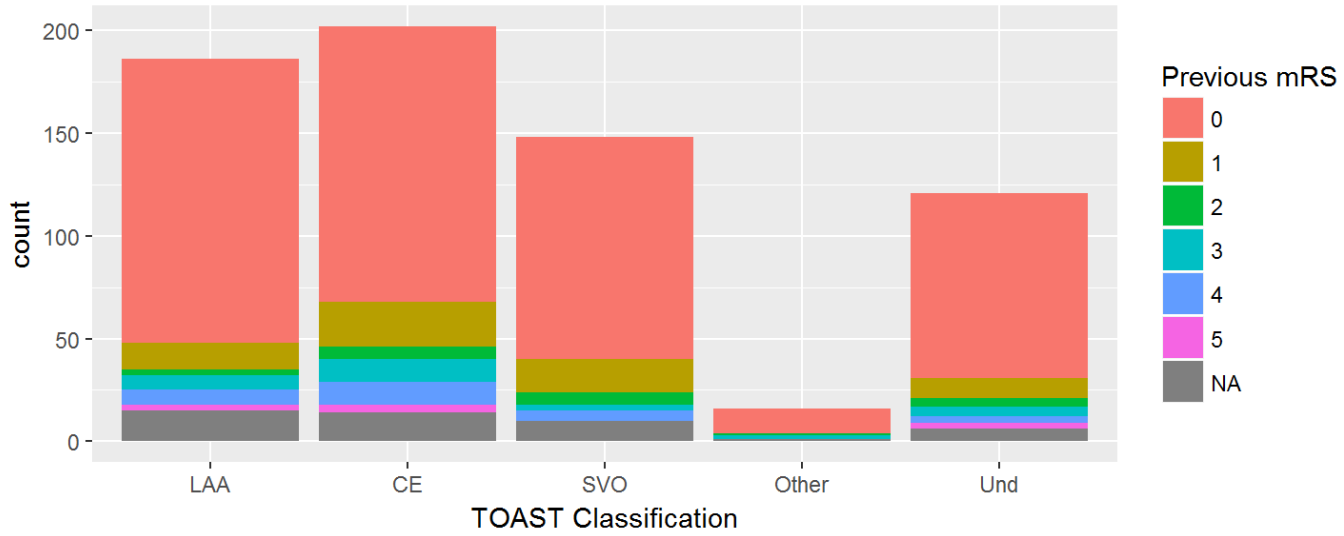
Scale	Types	Examples
scale_color_	identity	scale_fill_continuous
scale_fill_	manual	scale_color_discrete
scale_size_	continuous	scale_size_manual
	discrete	scale_size_discrete
scale_shape_	discrete	scale_shape_discrete
scale_linetype_	identity	scale_shape_manual
	manual	scale_linetype_discrete
scale_x_	continuous	scale_x_continuous
scale_y_	discrete	scale_y_discrete
	reverse	scale_x_log
	log	scale_y_reverse
	date	scale_x_date
	datetime	scale_y_datetime

Scale example

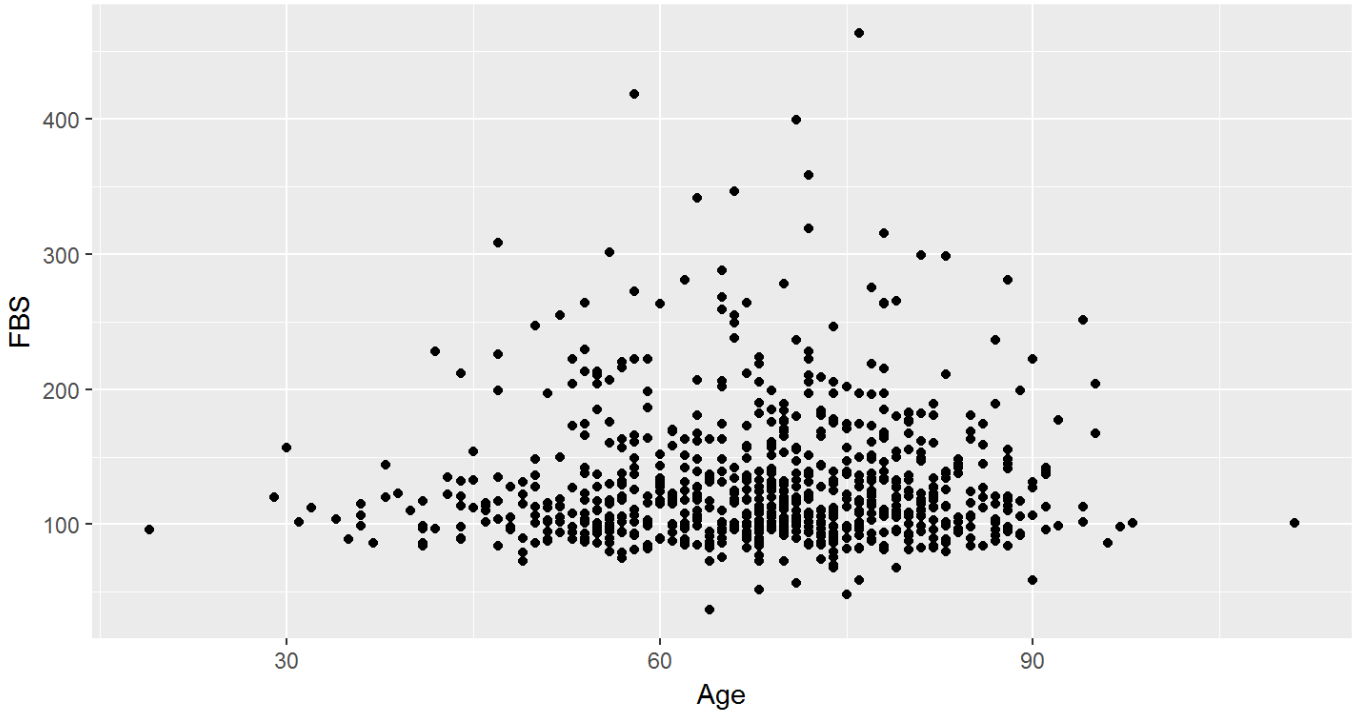
```
p6 <- ggplot(db, aes(TOAST))
p6 + geom_bar(aes(fill = factor(prev_mRS)))
```



```
p6 <- ggplot(db, aes(TOAST))
p6 + geom_bar(aes(fill = factor(prev_mRS))) +
  scale_fill_discrete(name = "Previous mRS") +
  scale_x_discrete(name="TOAST Classification",
                   limits=c("LAA", "CE", "SVO", "Other determined", "Undetermined"),
                   breaks=c("LAA", "CE", "SVO", "Other determined", "Undetermined"),
                   labels=c("LAA", "CE", "SVO", "Other", "Und"))
```

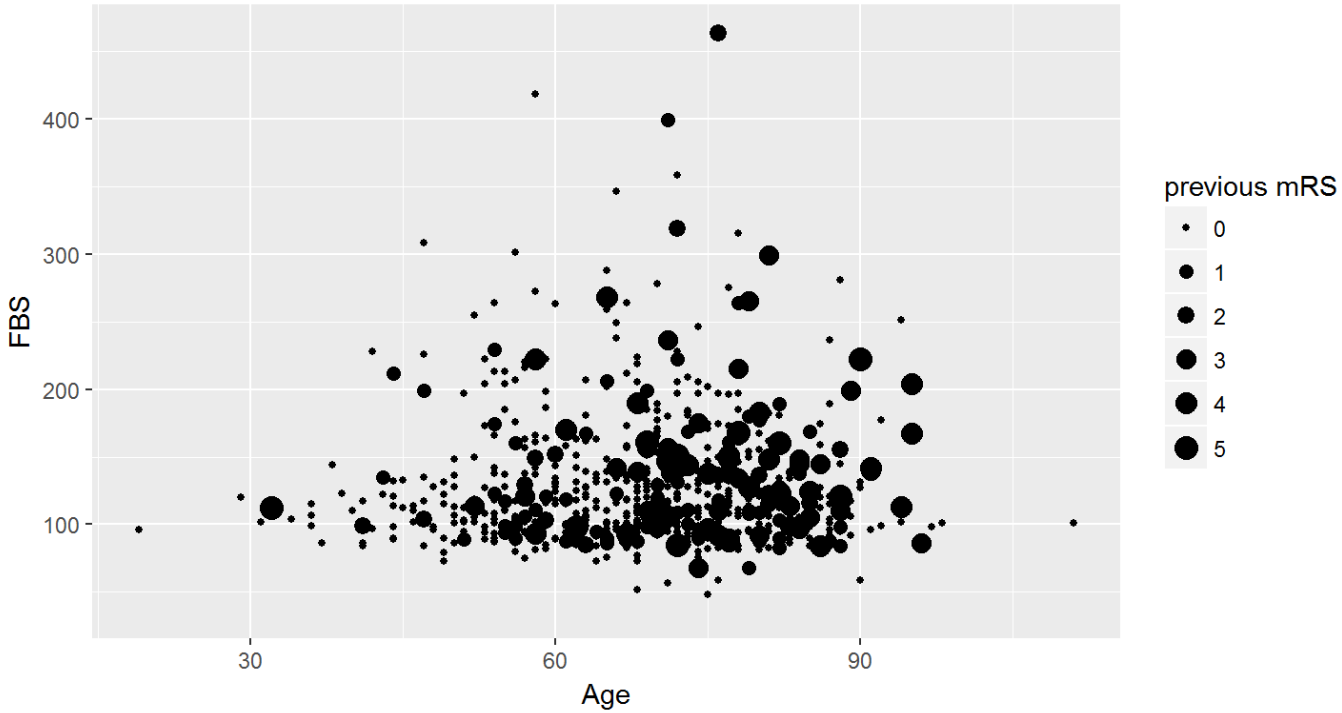


```
p7 <- ggplot(db, aes(x=Age, y=FBS))
p7 + geom_point()
```



```
p7 + geom_point(aes(size=prev_mRS)) +
  scale_size_continuous(name="previous mRS", range=c(1, 4))
```

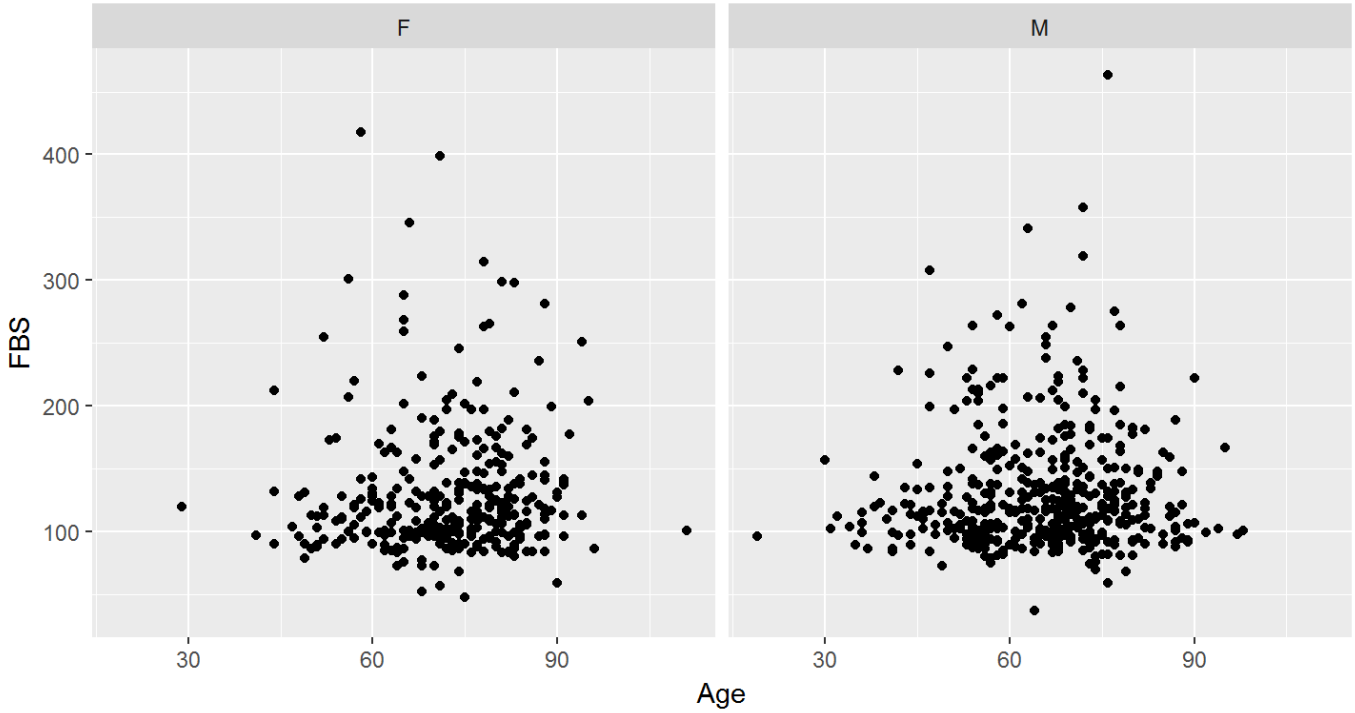
```
## Warning: Removed 71 rows containing missing values (geom_point).
```



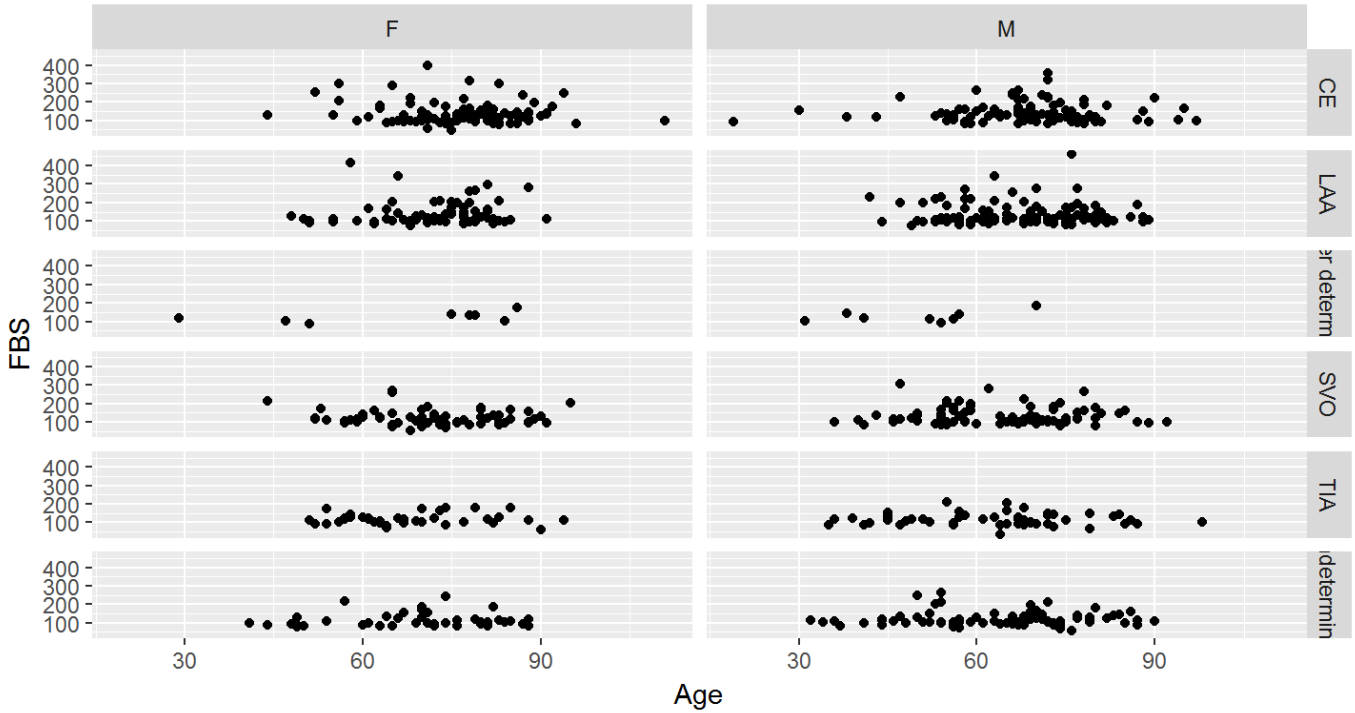
Faceting

Faceting

```
p7 <- ggplot(db, aes(x=Age, y=FBS))
p7 + geom_point() + facet_wrap(~Gender_F)
```



```
p7 + geom_point() + facet_grid(TOAST~Gender_F)
```

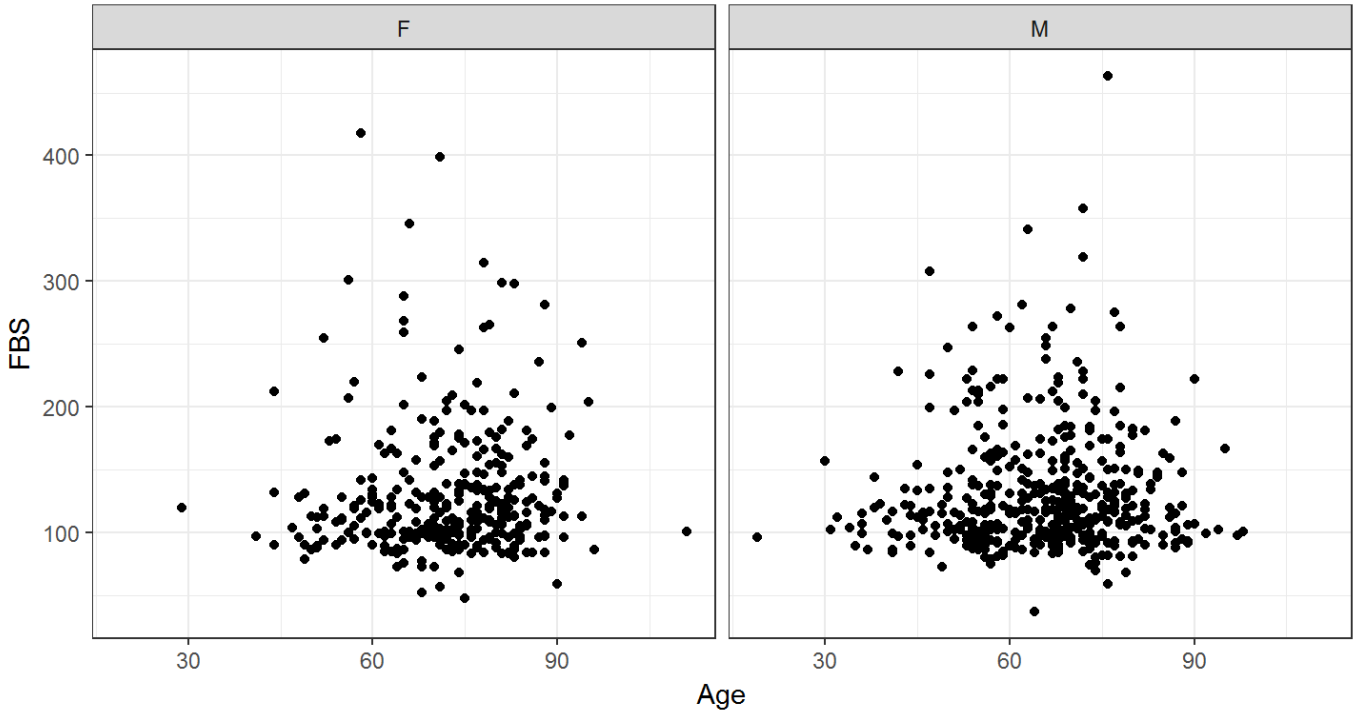


Theme and label

Theme

- The theme system handles non-data plot elements
  - Axis labels
  - Plot background
  - Facet label background
  - Legend appearance

```
p7 <- ggplot(db, aes(x=Age, y=FBS))
p7 + geom_point() + facet_wrap(~Gender_F) + theme_bw()
```

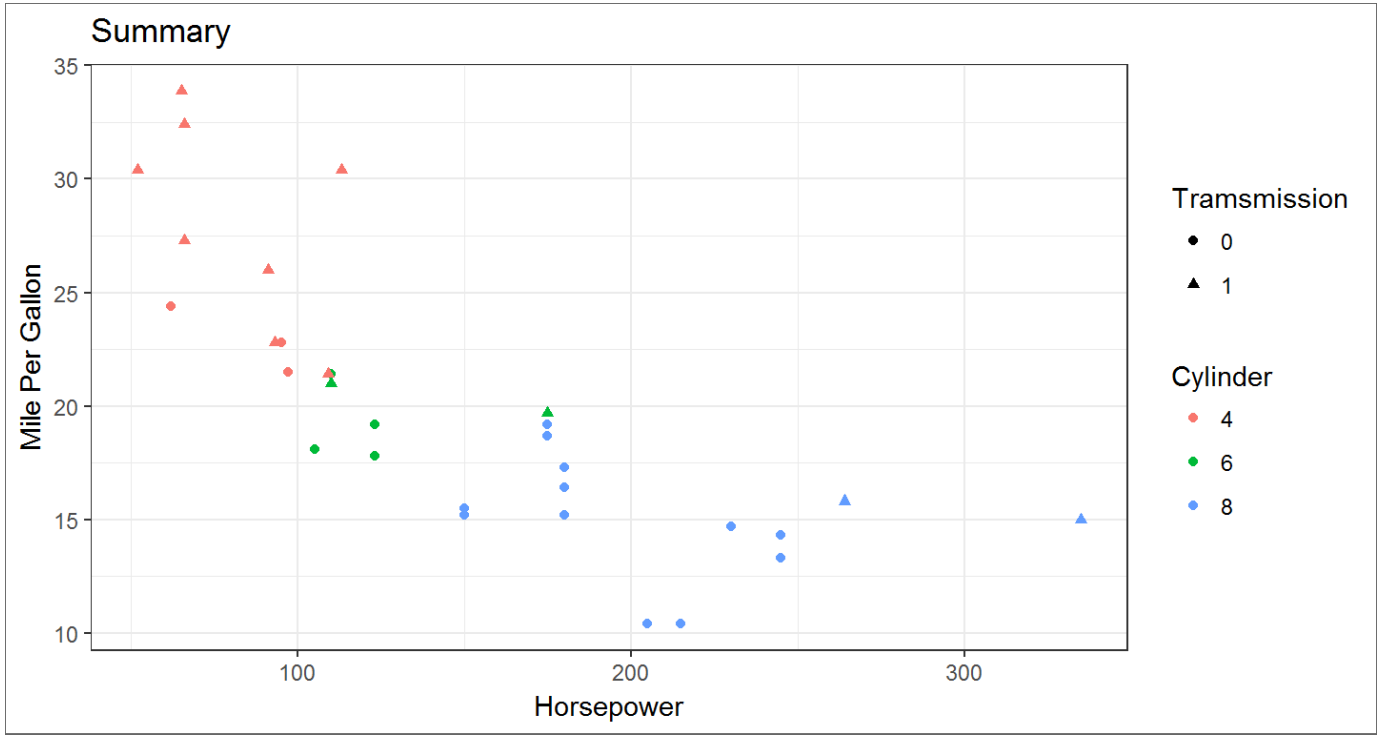


Label

- ggtitle("New title")
- xlab("New x label")
- ylab("New y label")

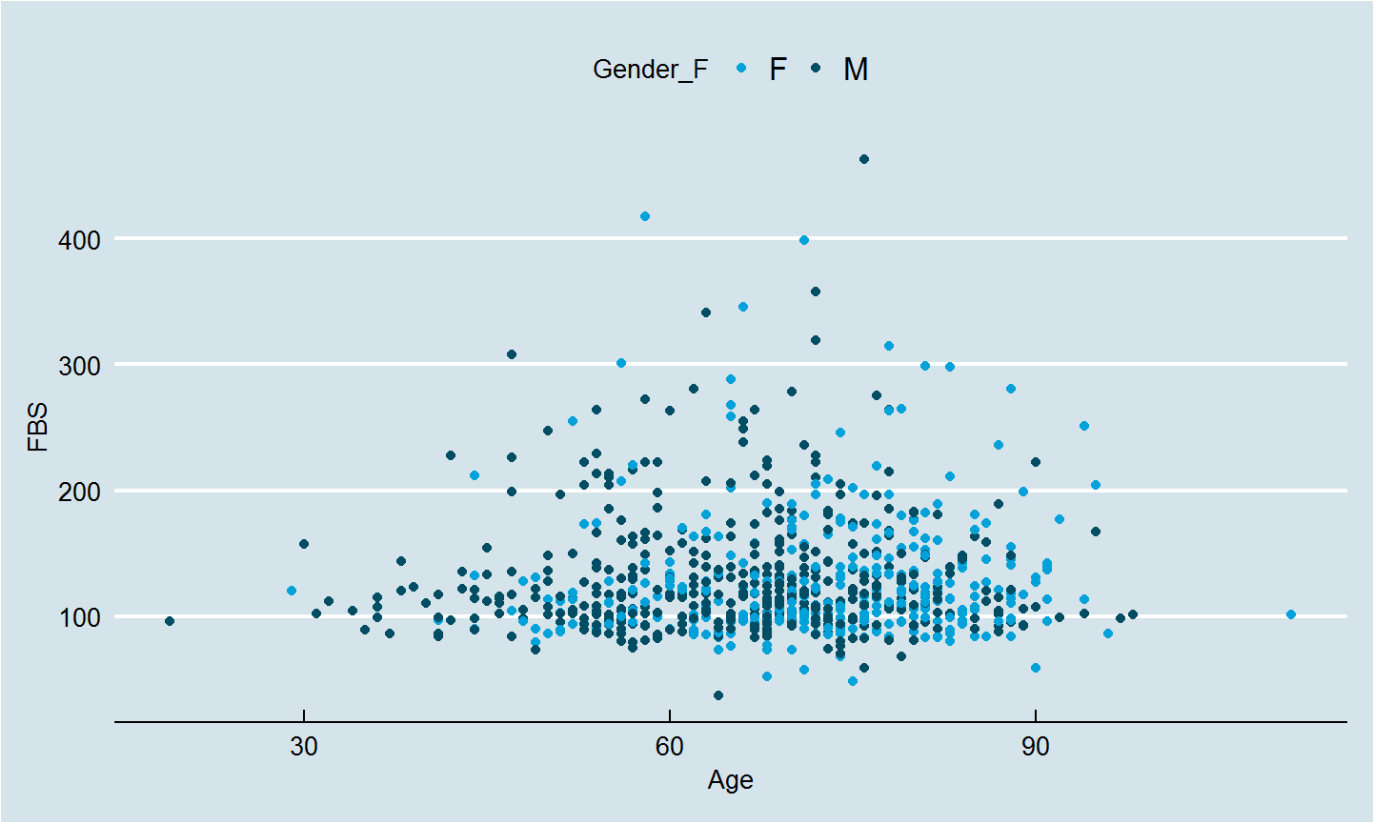
Exercise 2

- using db named mtcars or your own db file
- make the following graph

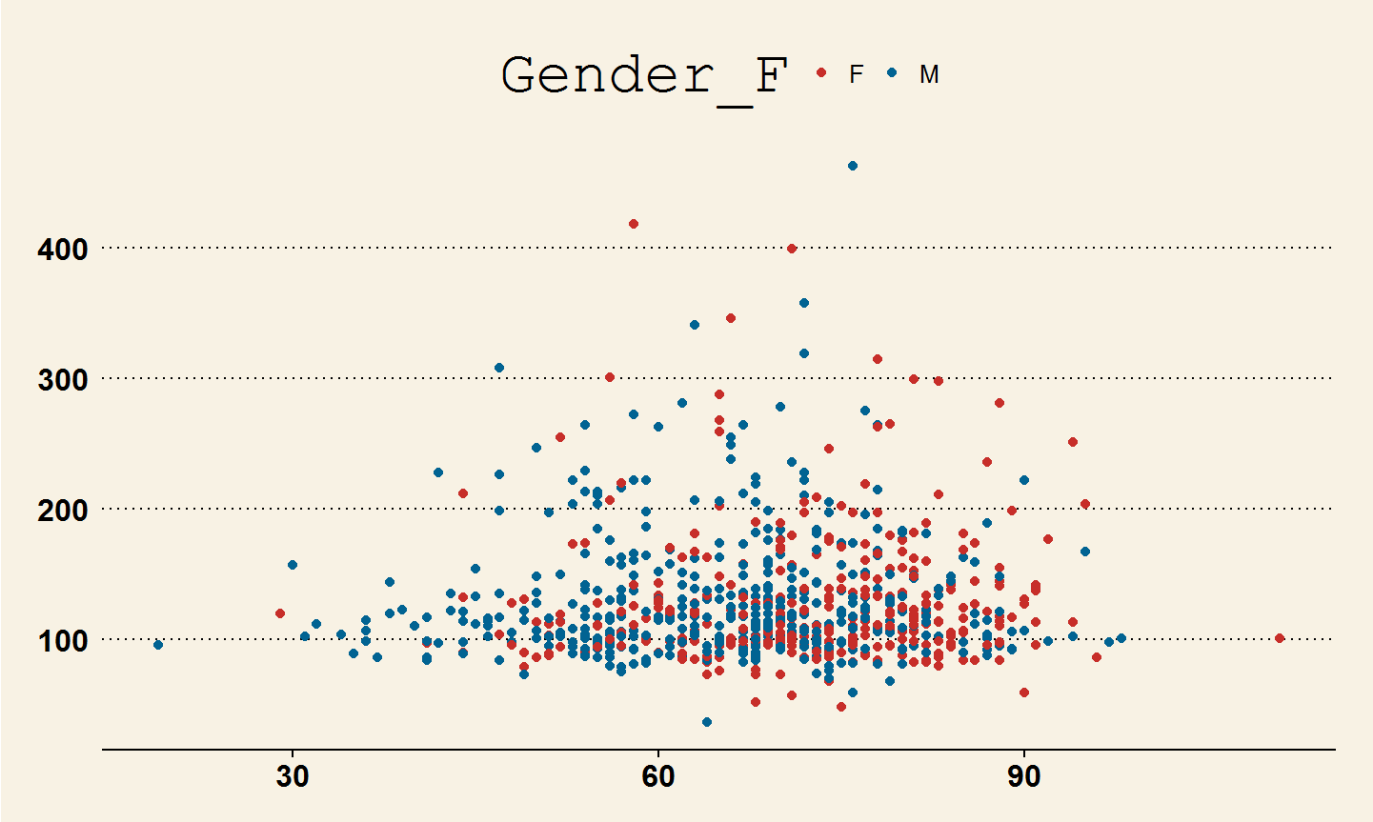


ggthemes

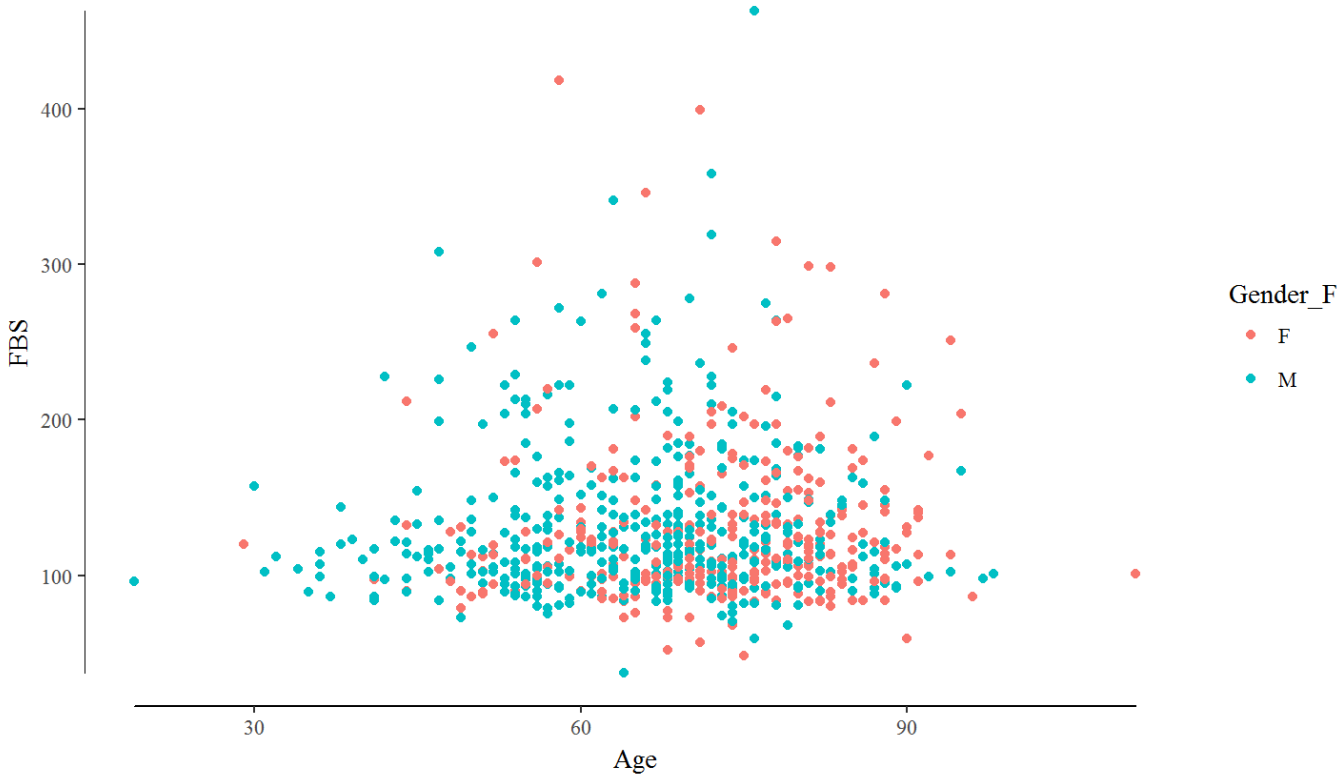
```
library(ggthemes)
p7 + geom_point(aes(color=Gender_F)) + theme_economist() + scale_color_economist()
```



```
p7 + geom_point(aes(color=Gender_F)) + theme_wsj() + scale_color_wsj("colors6")
```



```
p7 + geom_point(aes(color=Gender_F)) + geom_rangeframe() + theme_tufte()
```



```
p7 + geom_point(aes(color=Gender_F)) + theme_solarized()
```