

# Reproducible Research (2)

Camera-Ready plots

---

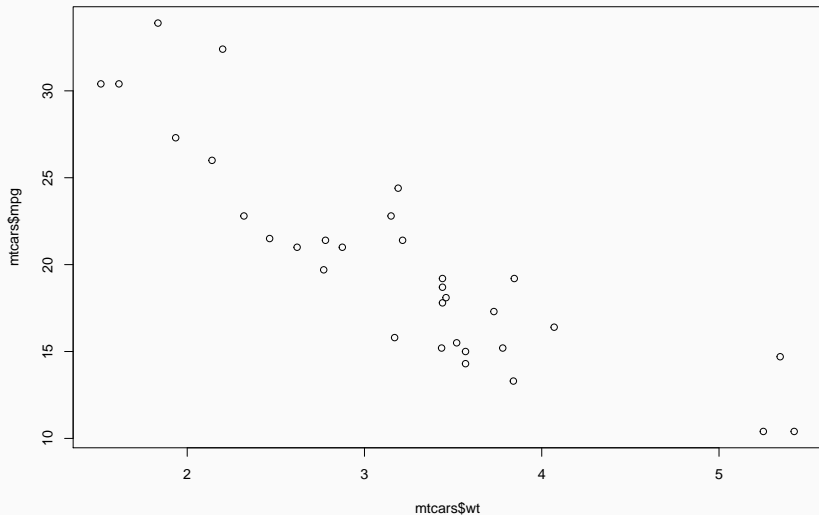
Zhuanghua Shi

12 May 2016

```
git clone https://github.com/strongway/  
seminar-reproducible-research.git
```

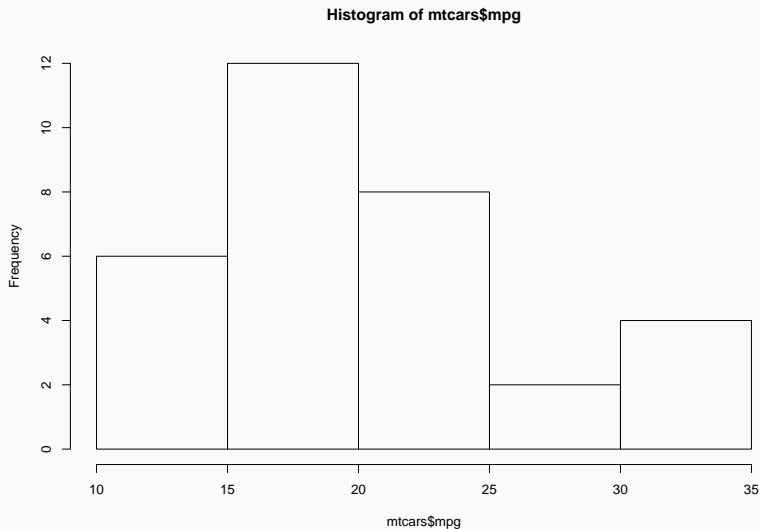
# R basic plots

```
plot(mtcars$wt, mtcars$mpg)
```



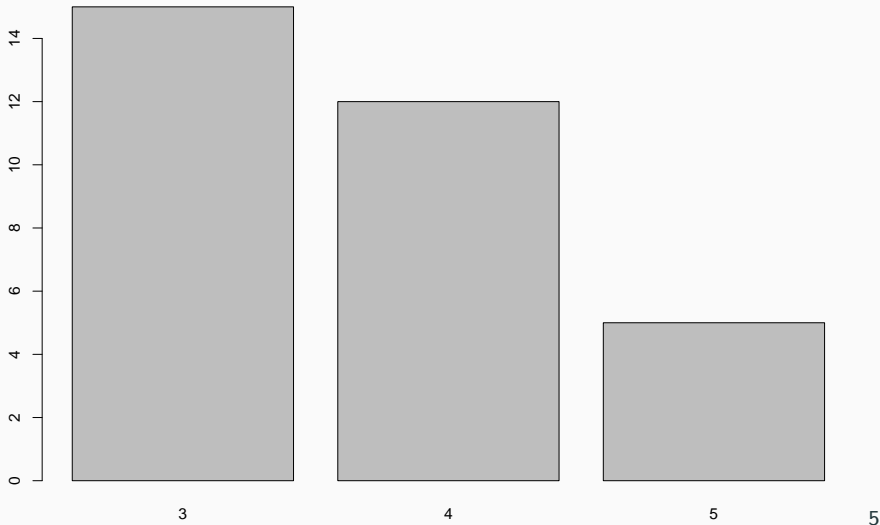
# R basic plots

```
hist(mtcars$mpg)
```



## R basic plots

```
barplot(table(mtcars$gear))
```



## Two powerful packages

- `ggplot2`
  - grammar of graphics
  - It takes care of many of the fiddle details

<http://docs.ggplot2.org/>

- `ggvis`
  - grammar of graphics
  - heavily use pipe `%>%`
  - Interactive

<http://ggvis.rstudio.com>

# Package - ggplot

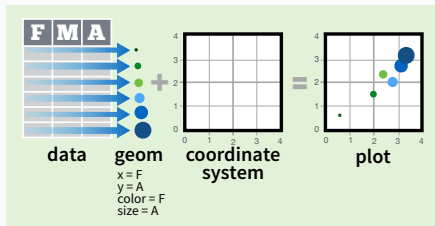
three key components

- a data set
- a *coordinate system*
- a set of *geoms*
  - layer by layer

graph = **mtcars** + **cartesian** + **points** + **fill =** + ...  
data coordinate system mark properties

```
ggplot(mtcars, aes(x=mpg, y = hp, color = gear)) +  
geom_point() + geom_smooth()
```

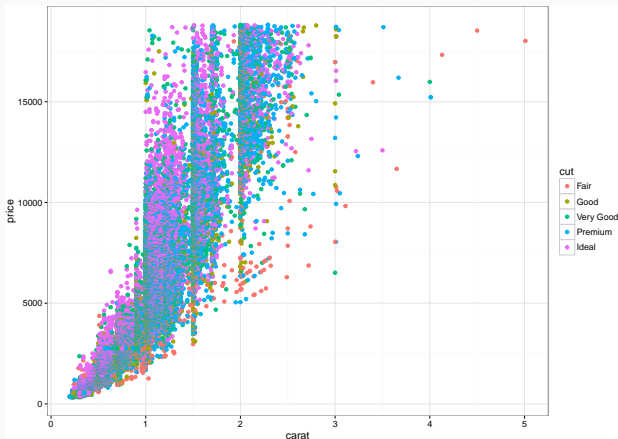
# ggplot grammar





# ggplot examples

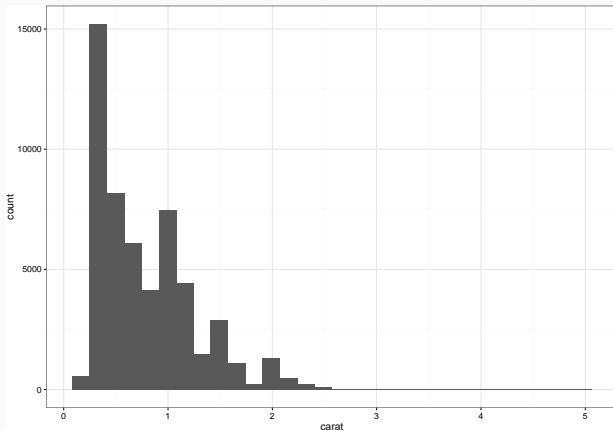
```
ggplot(diamonds, aes(x = carat, y = price, color = cut)) +  
  geom_point()
```



## ggplot examples

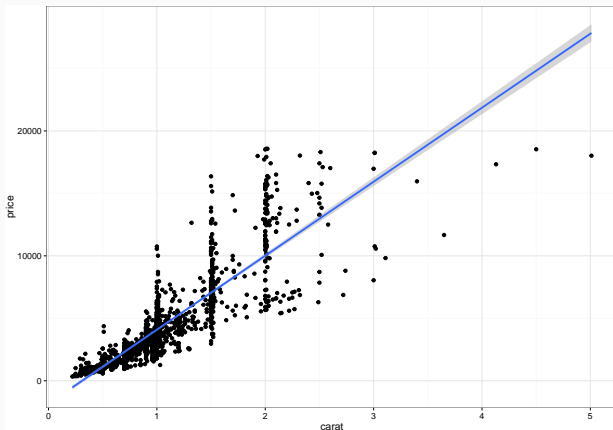
```
ggplot(diamonds, aes(x = carat)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `
```



## ggplot examples

```
diamonds %>% filter (cut == "Fair") %>% ggplot(., aes(x = carat, y = price)) +  
  geom_point() + geom_smooth(method = 'lm')
```



# Package ggvis

- relative young, still under development
- powerful for interactive
- better consistent grammars than ggplot
  - data
  - coordinate system,
  - properties: marks, colors, lines

Tutorial Video:

[https://campus.datacamp.com/courses/  
ggvis-data-visualization-r-tutorial/  
chapter-one-the-grammar-of-graphics?ex=4](https://campus.datacamp.com/courses/ggvis-data-visualization-r-tutorial/chapter-one-the-grammar-of-graphics?ex=4)

## Basic grammar of ggvis

```
dataset %>% ggvis(~x, ~y, fill = ~variable, ...) %>%  
layer_<mark>()
```

e.g.,

```
mtcars %>% ggvis(x = ~mpg, y = ~hp, fill = ~cyl) %>%  
layer_points() %>% layer_smooths() %>%  
layer_model_predictions(model='lm', stroke := 'red' )
```