Lecture 13: Oct 1, 2018

EDA

- *Exploratory Data Analysis*
- *Graphing in R*
- *ggplot2*
- *Making Graphs*
    - *scatterplot*, *line graphs*, *bar plots*, *histogram*, and *box plots*
- *Themes*
- *Resources*

James Balamuta
STAT 385 @ UIUC

Announcements

- **hw05** is due **Friday, Oct 5th, 2018** at **6:00 PM**
- **John Lee's Office Hours** are now from **4 - 5 PM** on **WF**
- **Quiz 06** covers Week 5 contents @ CBTF.
  - Window: Oct 2nd - 4th
  - Sign up: https://cbtf.engr.illinois.edu/sched
- Want to review your homework or quiz grades?
  Schedule an appointment.
- Got caught using GitHub's web interface in hw01 or hw02?
  Let's chat.

Lecture Objectives

- Apply quantitative and visual EDA techniques to data
- Deduce and interpret patterns revealed in the EDA format
- Explain the underlying grammar of graphics.
- Create common statistical graphs using ggplot2.

Exploratory Data Analysis

Definition:
Exploratory Data Analysis (EDA) is a philosophy for the beginning of an analysis that describes a variety of techniques that are *quantitative* and *visual* in nature to look for patterns in data.

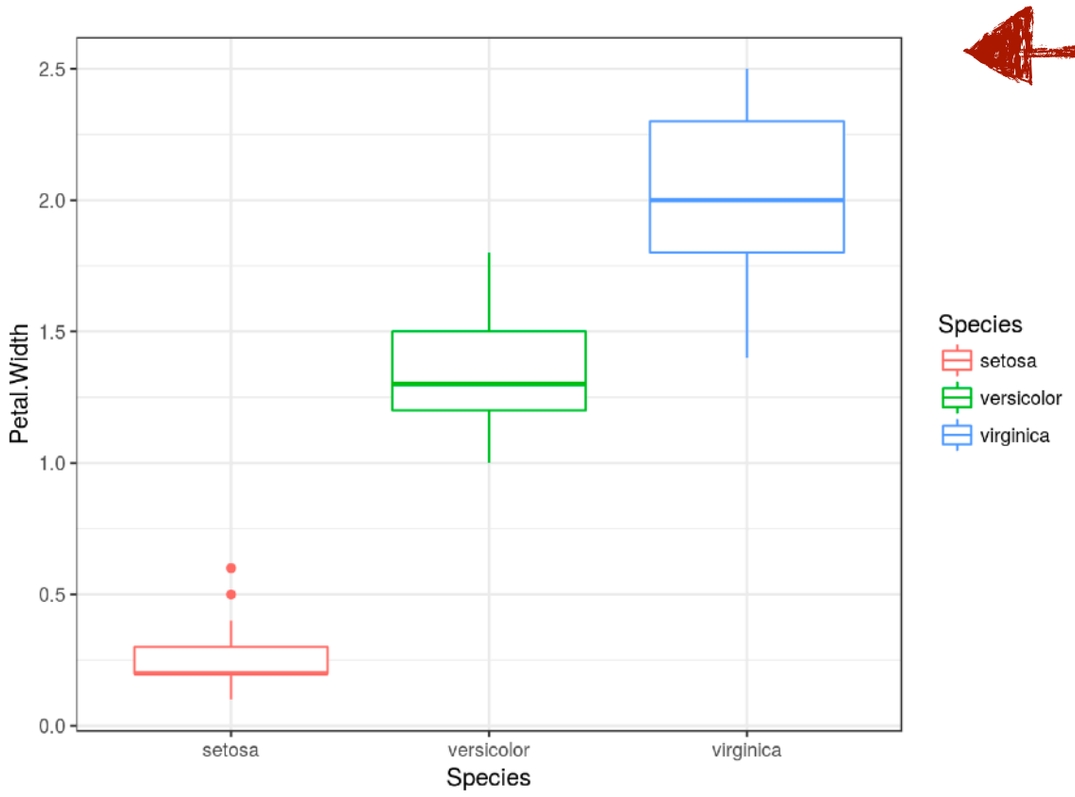
Source

Types of EDA

… quantitative vs. visual …

```
 Petal.Width          Species
Min.   :0.100   setosa    :50
1st Qu.:0.300   versicolor:50
Median :1.300   virginica :50
Mean   :1.199
3rd Qu.:1.800
Max.   :2.500
```

```
# Obtain summary information
summary(iris[, 4:5])
```

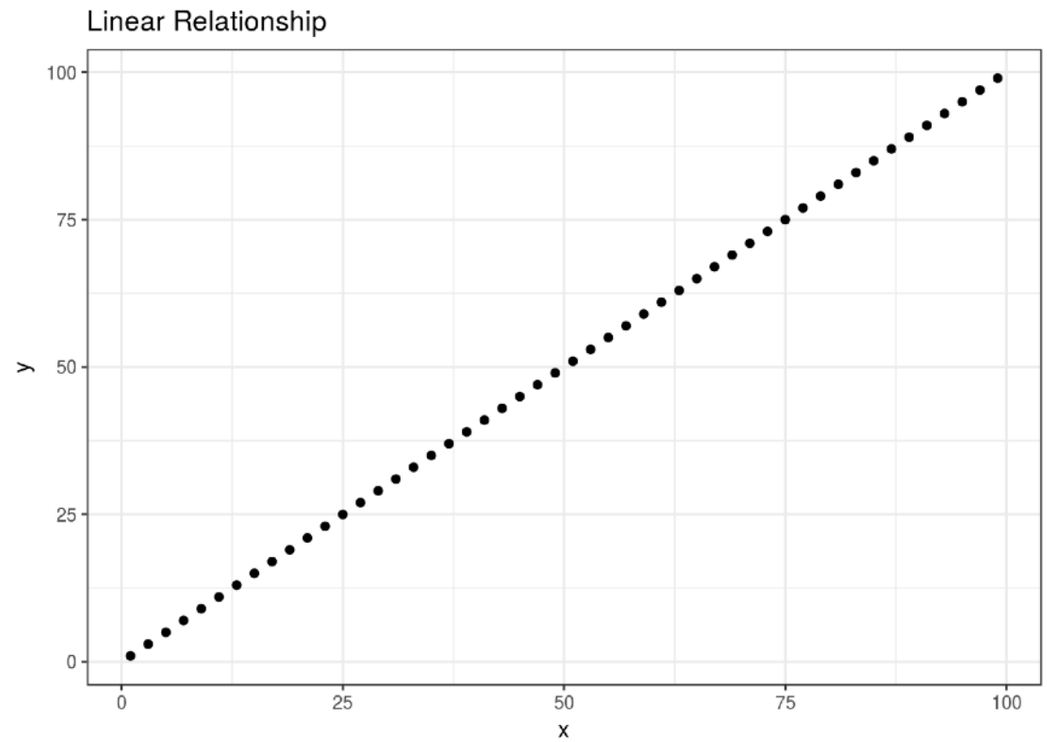


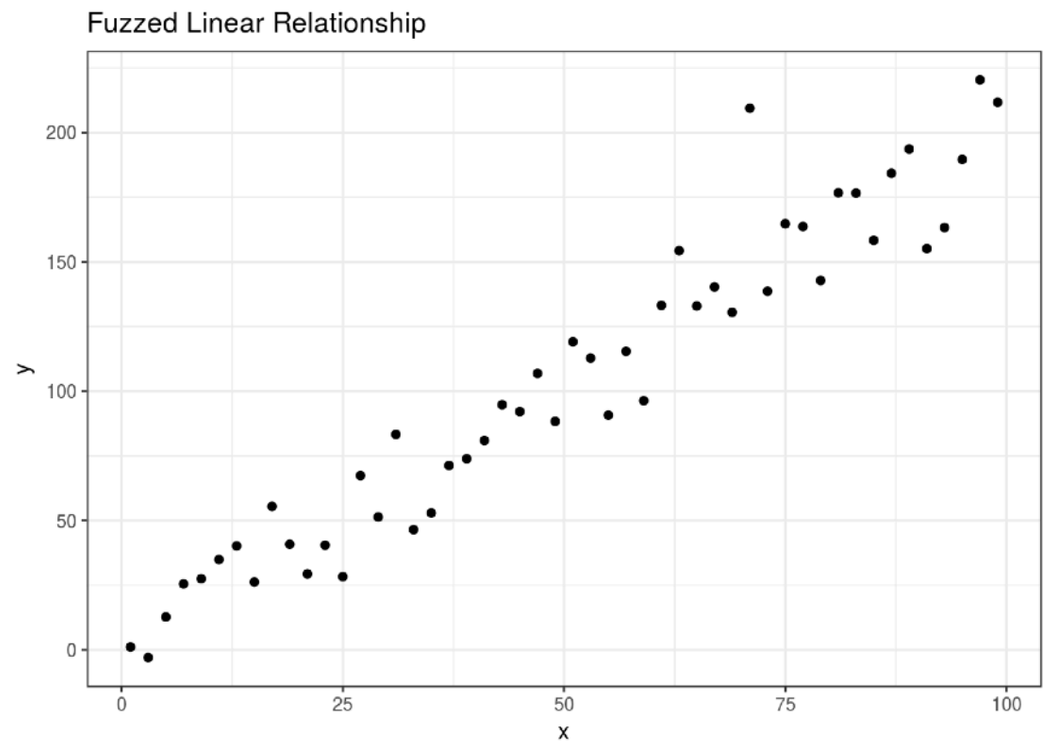

```
# Visualize information with
# a box plot
ggplot(iris) +
  geom_boxplot(
    aes(x = Species,
        y = Petal.Width,
        color = Species ))
```

Patterns

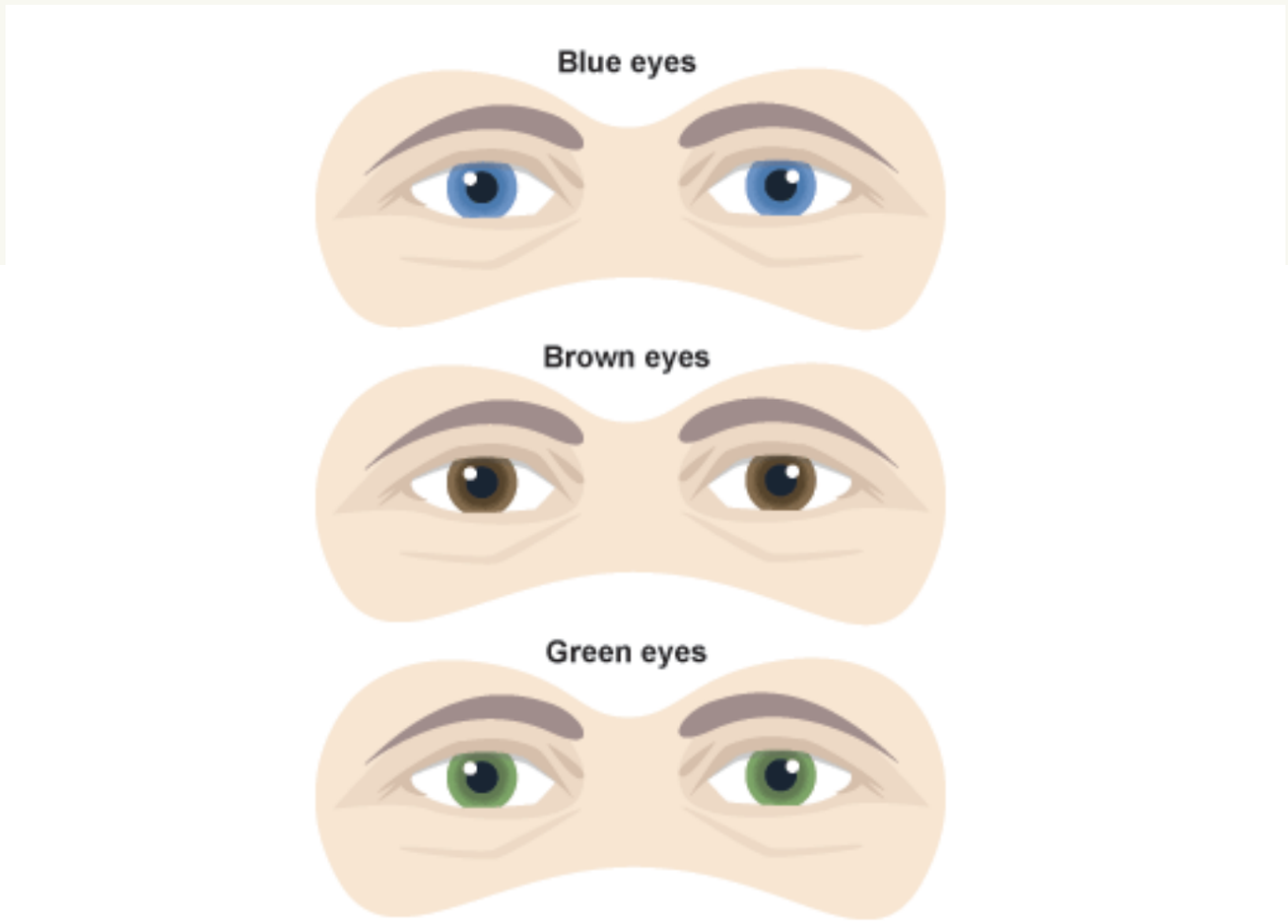
… detecting, analyzing, and communicating …

1. What kind of **relationship in the data** is present?
2. What's the **level of strength** of the relationship?
3. Any **confounding variables** that might be behind it?
4. What **happens if we look at subsets** of the data?



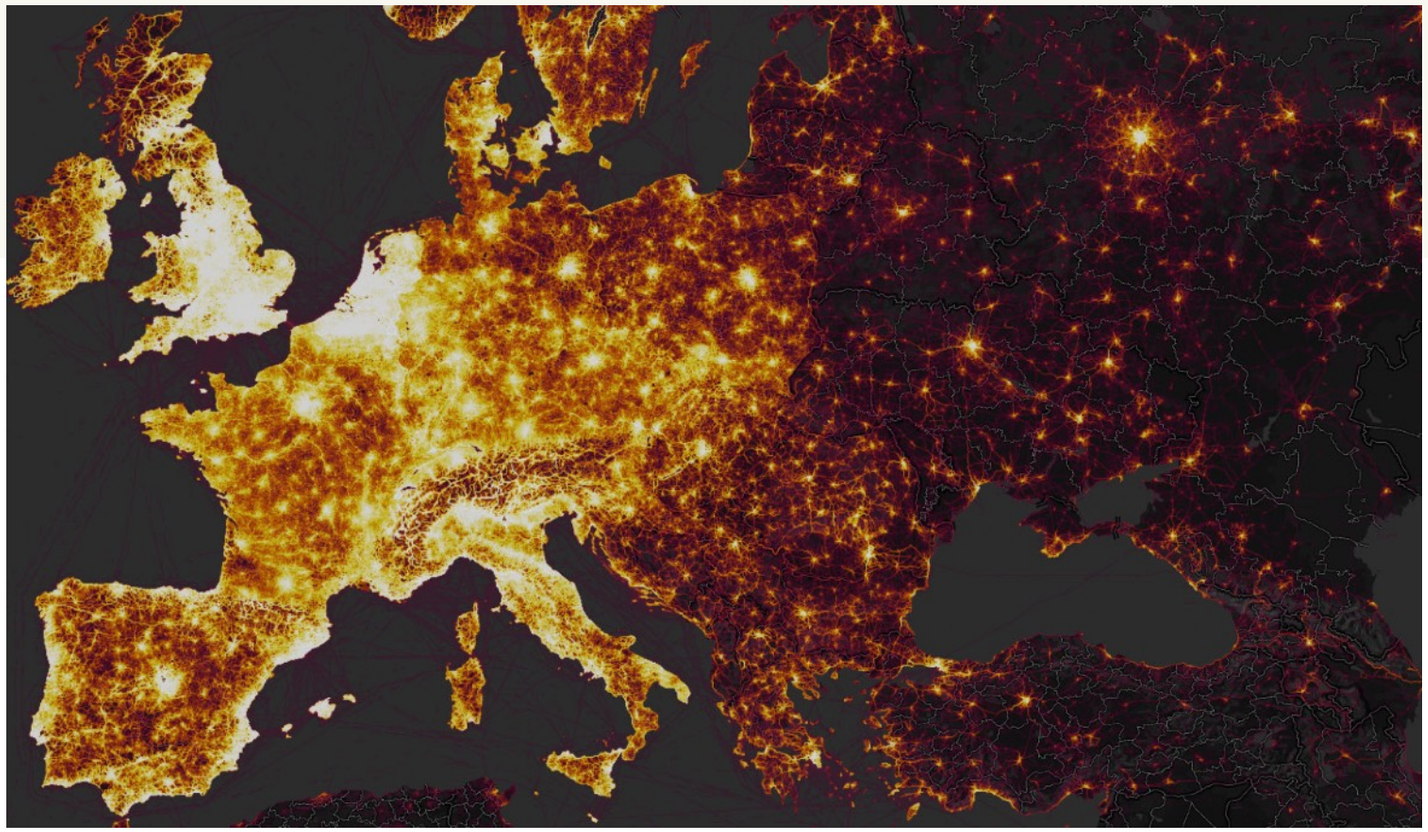


Definition:
Variation is the difference between observations in **one variable**.



Source

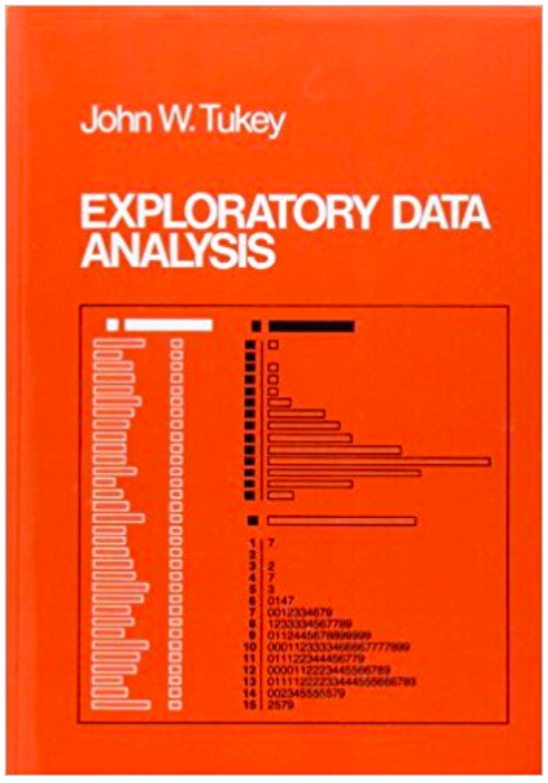
Definition:
Covariation is the difference among observations between **two or more** variables.



Source

"...make **both** calculations **and** graphs. Both sorts of output should be studied; each will contribute to understanding."

– F. J. Anscombe, 1973

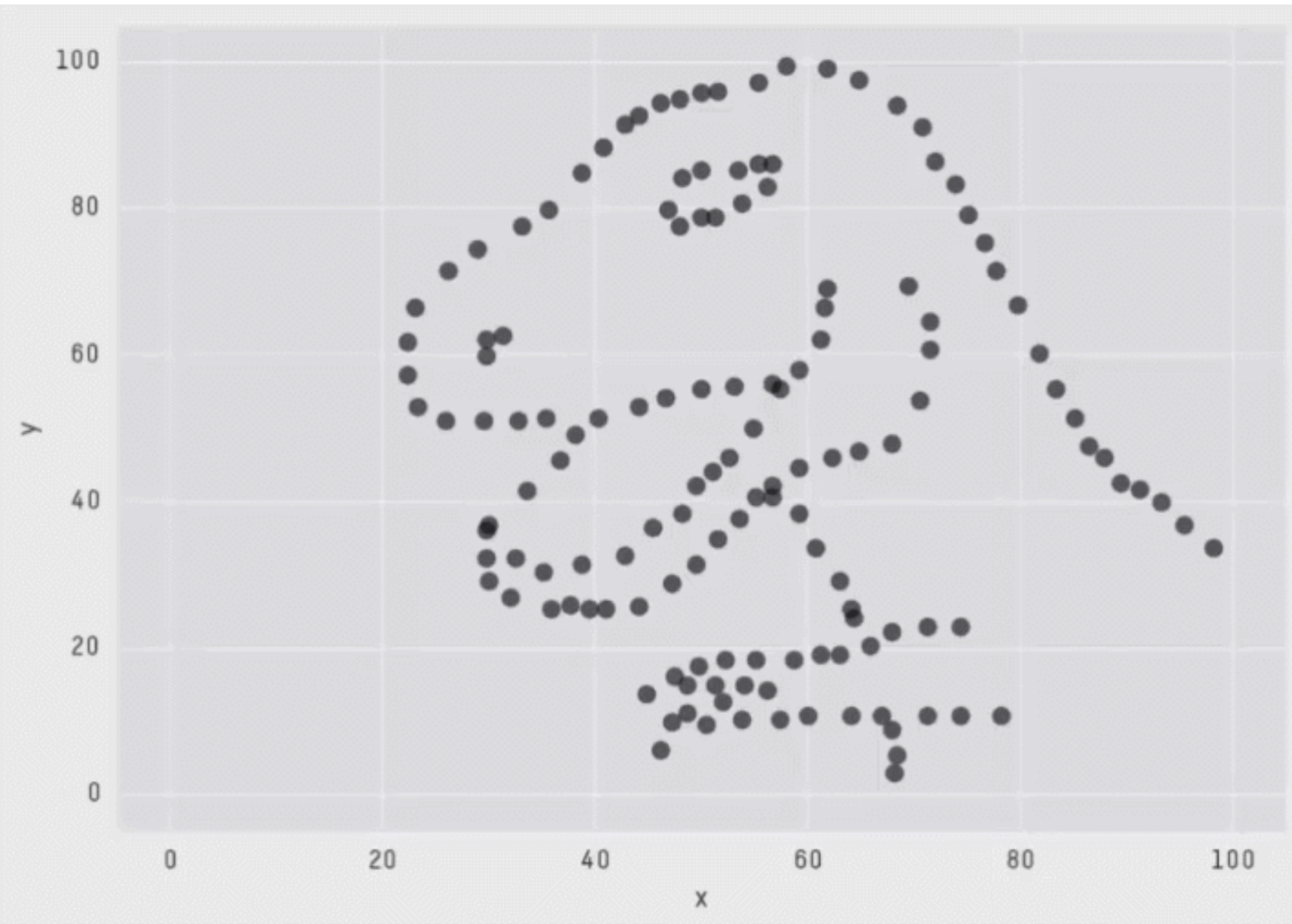


“The greatest value of a picture is when it forces us to notice what we never expected to see.”

–John Turkey in Exploratory Data Analysis (1977)

Datasaurus

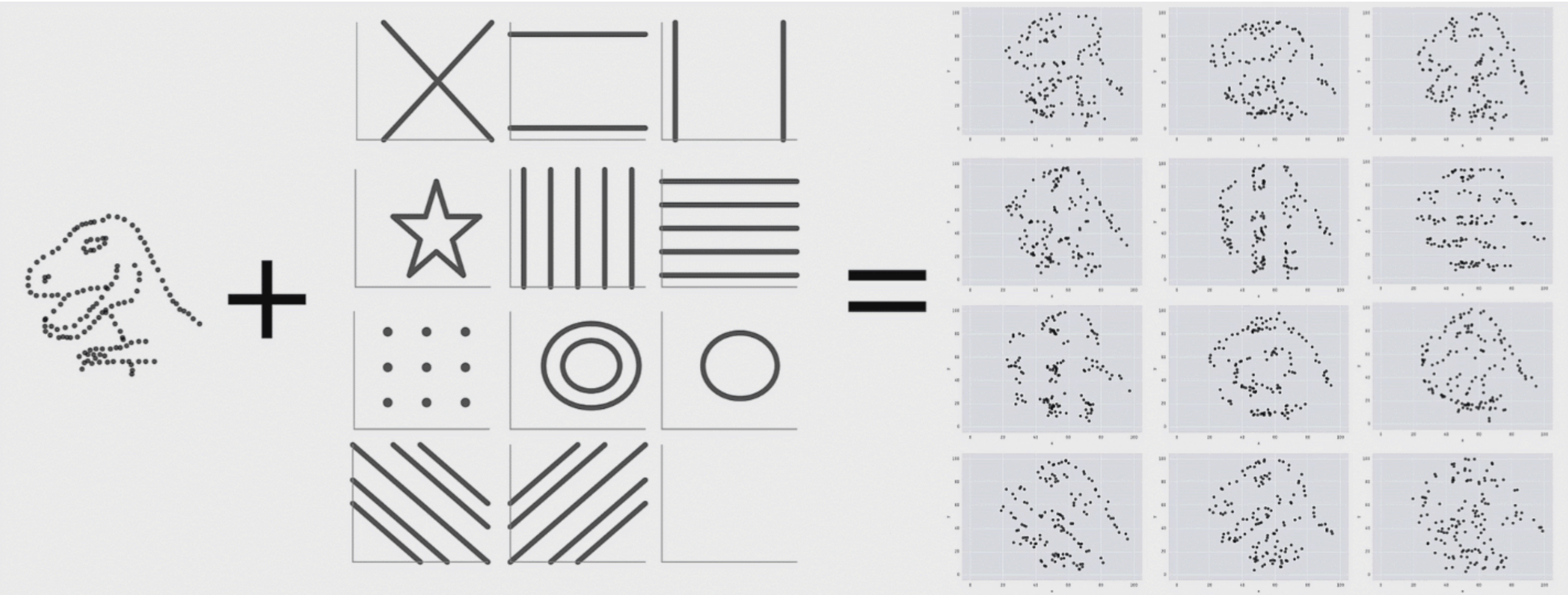
... the new anscombe's quartet ...



X Mean: 54.2659224
Y Mean: 47.8313999
X SD : 16.7649829
Y SD : 26.9342120
Corr. : -0.0642526

Source

Huh?



Source

Generated Data

... hiding a message in plain sight ...

```r
set.seed(2018)                                  # Set seed for reproducibility
n = 1e4                                         # Number of observations

a = matrix(rnorm(n * 2), ncol = 2)              # Generate values
rng = runif(n, 0, 2 * pi)
b = 0.5 * cbind(sin(rng), cos(rng))

o = rbind(a, b)                                 # Combine generate data

my_data = as.data.frame(o[sample(nrow(o)), ])   # Randomly sample data

colnames(my_data) = c("x", "y")
```

Source

Quantitative EDA

… trying to discern a pattern …

```
# Dimensions of Data
dim(my_data)
nrow(my_data) # Number of Rows
ncol(my_data)  # Number of Columns

# Summarize data
summary(my_data)

# First observations in data
head(my_data)

# Last observations in data
tail(my_data)

# Kind of data
class(my_data)

# Layout of data
str(my_data)

# Any missing values?
is.na(my_data)
```

Pattern

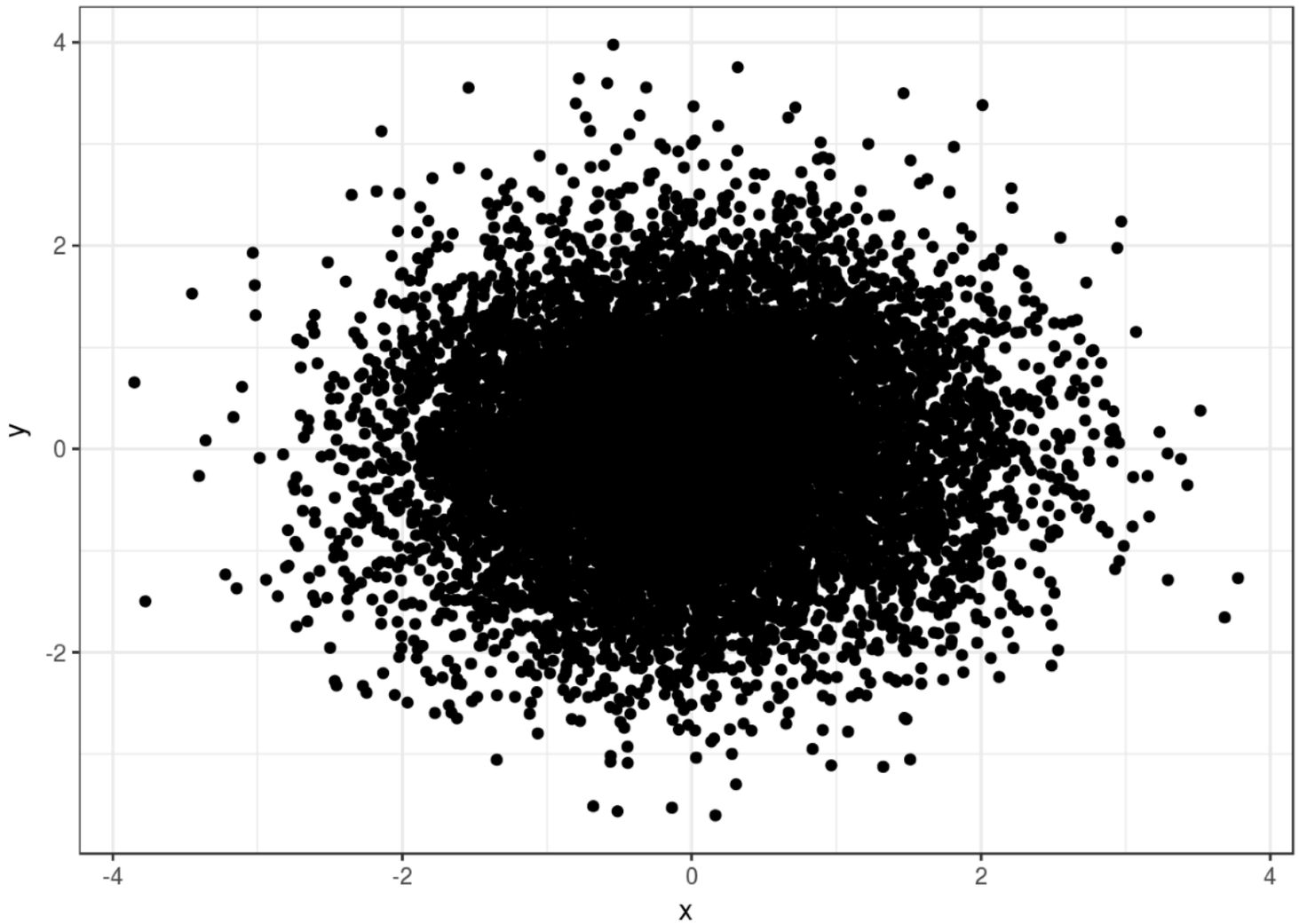
… what do you see in **numbers**???

```
summary(my_data)
```

```
       x                    y             
 Min.   :-3.851009   Min.   :-3.602245  
 1st Qu.:-0.433924   1st Qu.:-0.429797  
 Median :-0.001436   Median : 0.003054  
 Mean   : 0.003383   Mean   : 0.001575  
 3rd Qu.: 0.431648   3rd Qu.: 0.434031  
 Max.   : 3.777314   Max.   : 3.975975  
```

Pattern

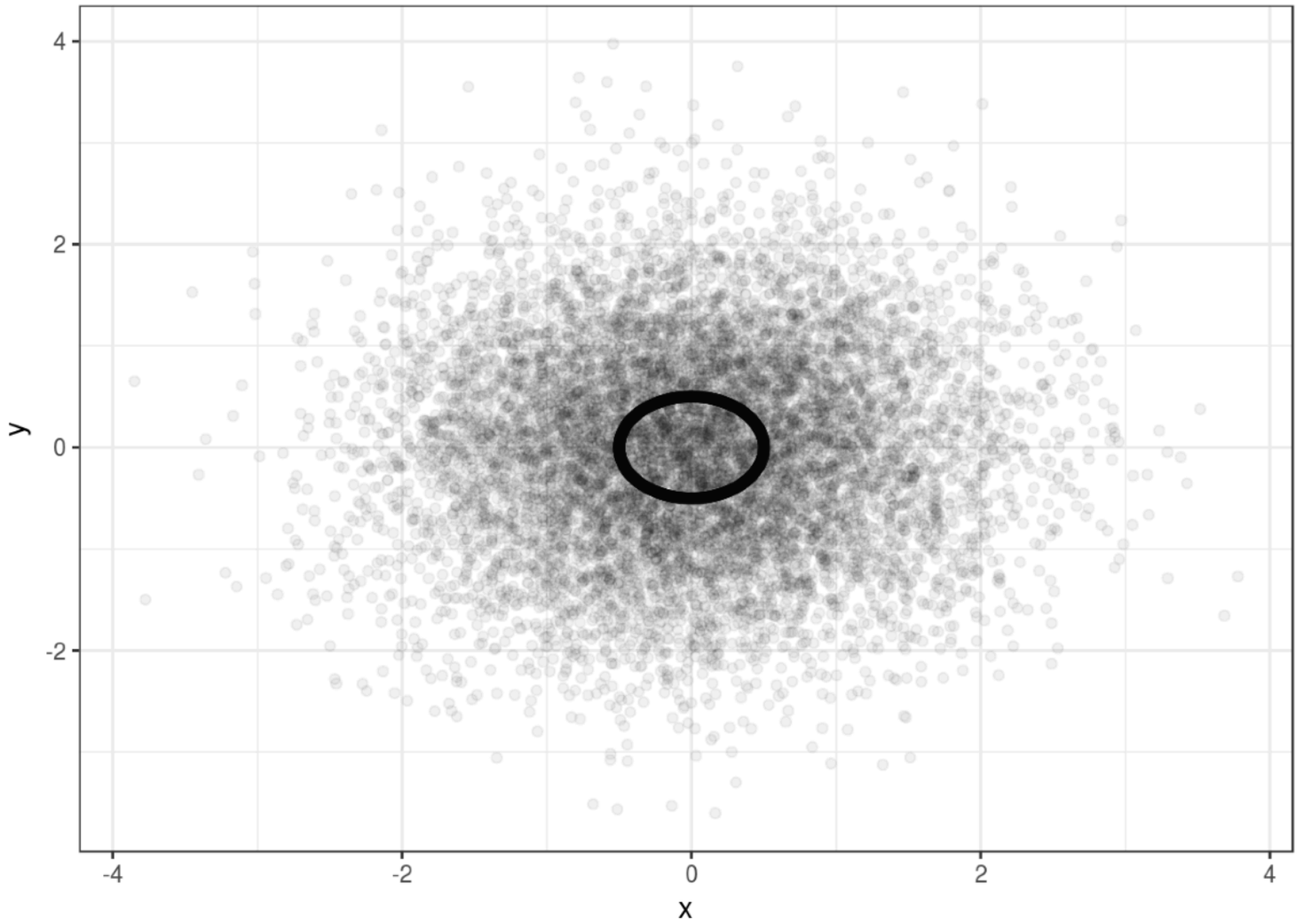
… what do you see in a **graph**???




```
ggplot(my_data) +
  geom_point(aes(x = x, y = y))
```

Pattern

… what do you see in a **transparency graph**???

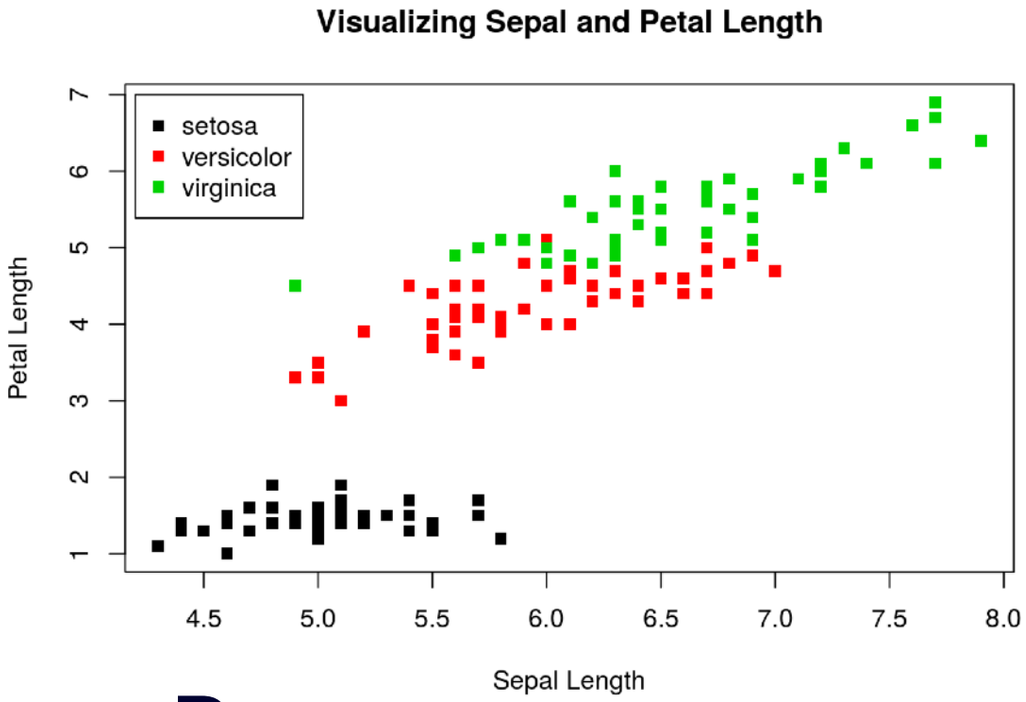



```
ggplot(my_data) +
  geom_point(aes(x = x, y = y), alpha = 0.05)
```

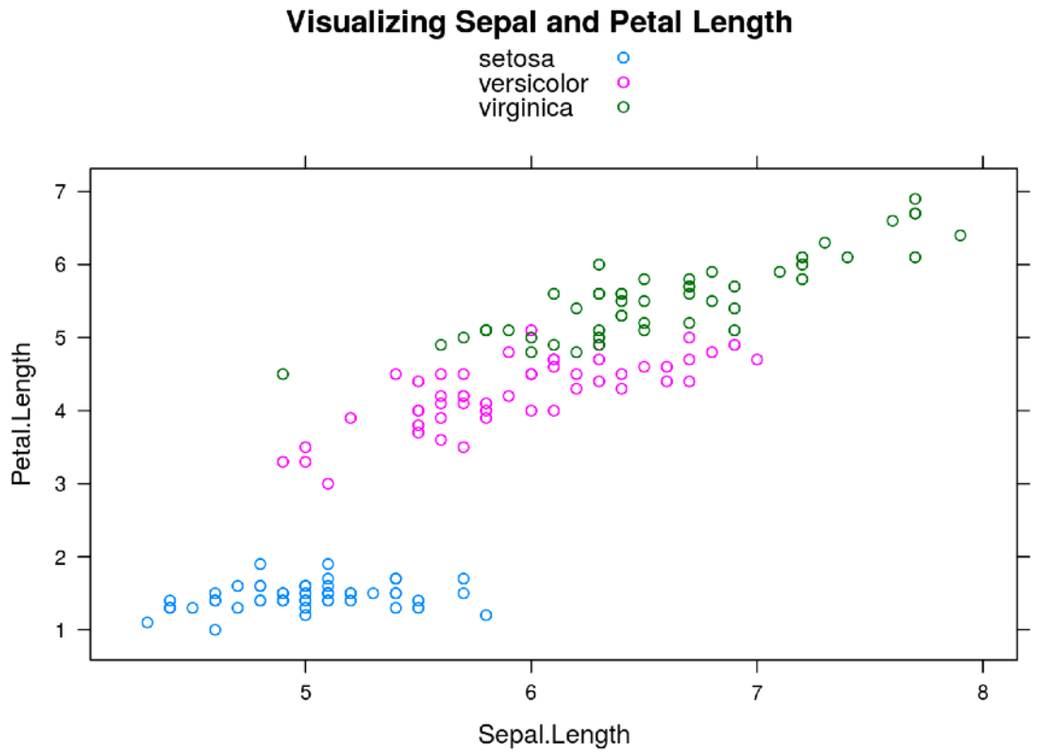
Graphing in R

Graphics Systems

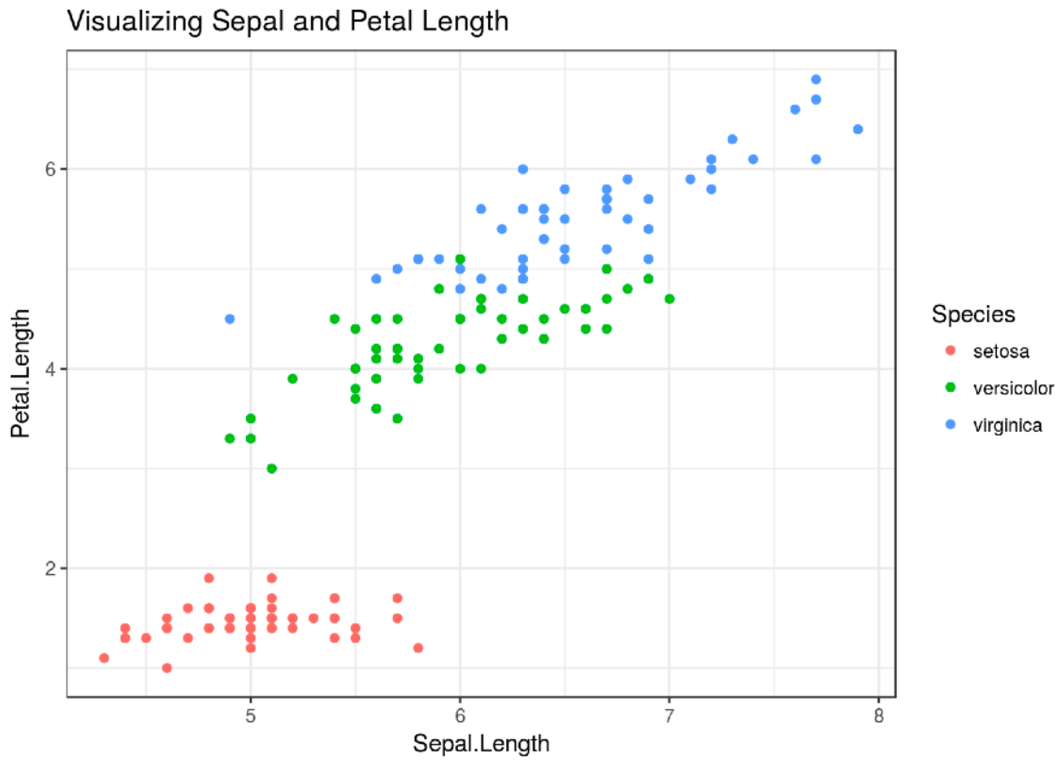
… different graphics systems …



Base R



lattice



ggplot2

Goldilocks Scenario

… what graph system to choose ???

Base R

Highly customizable

lattice

Formula-based limitations

ggplot2

Grammar of Graphics

Images Source

ggplot2

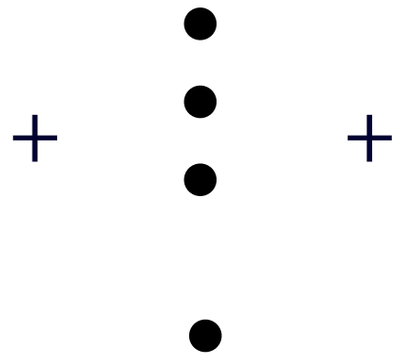
… grammar of graphics …

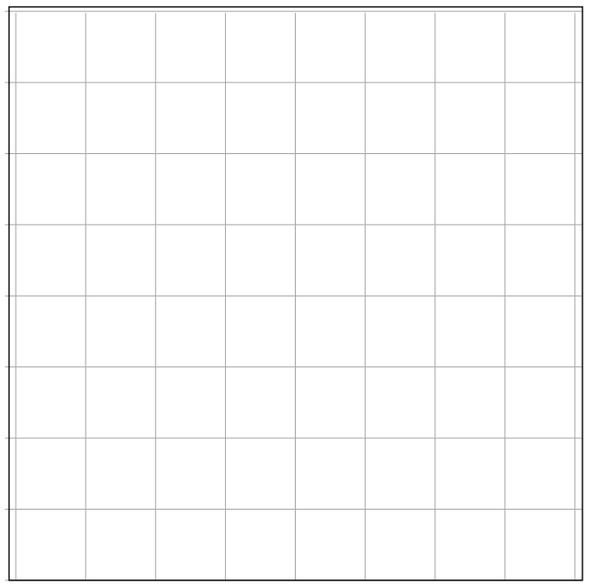
- **ggplot2** is an R Package by Hadley Wickham
  - Implementation of the pivotal theory of the Grammar of Graphics (1999) by Leland Wilkinson.
- **Grammar of Graphics**
  - Each graph shares a *common structure*.
  - The difference between graphs is different component layers and rules

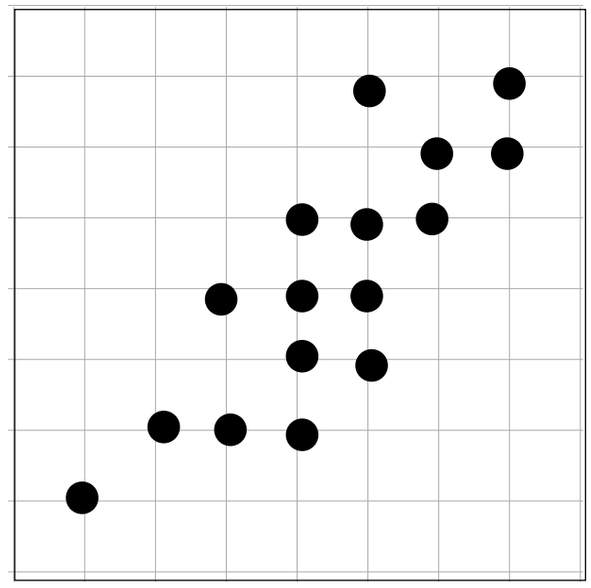
Grammar of a Graph

… graphs broken down …

A	B	C







Data

Geometric Object

Coordinate System

Plot

---

* Grammar of graphics relies on a layering system. Each of these objects are connected together by adding individual components together.

ggplot2 terminology

… formalizing layering …

Grammar of Graphics emphasizes the….

- **mapping** of data onto a **facet** and **coordinate system**
- with **aesthetic** values (e.g. color, shape, size, …)
- using **geometric** objects (e.g. points, lines, bars, …)

Base Template

… starting point for creating graphs with ggplot2 …

Data Set
Data to be visualised

Layers
Adding components to the plot

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION> ( mapping = aes( <MAPPINGS> ) )
```

Geometric Object
How the data will be displayed on the graph

Aesthetics
Visual properties of the geometric object

Adding Layers

… basis for making ggplot2 graphs …

Layers

Adding components to the plot

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION> ( mapping = aes( <MAPPINGS> ) )
```

Adding Layers Mistake

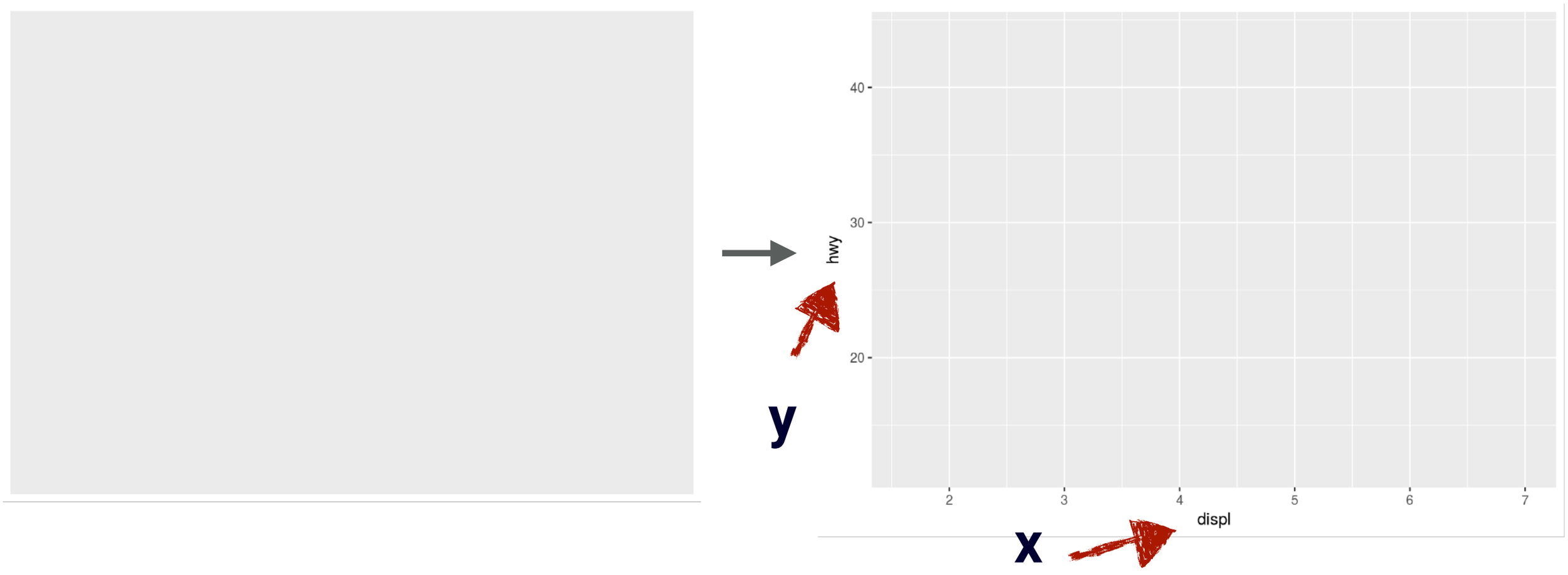
… add sign belongs at END of a line **not** START of a new line …

Layers
Adding components
to the plot

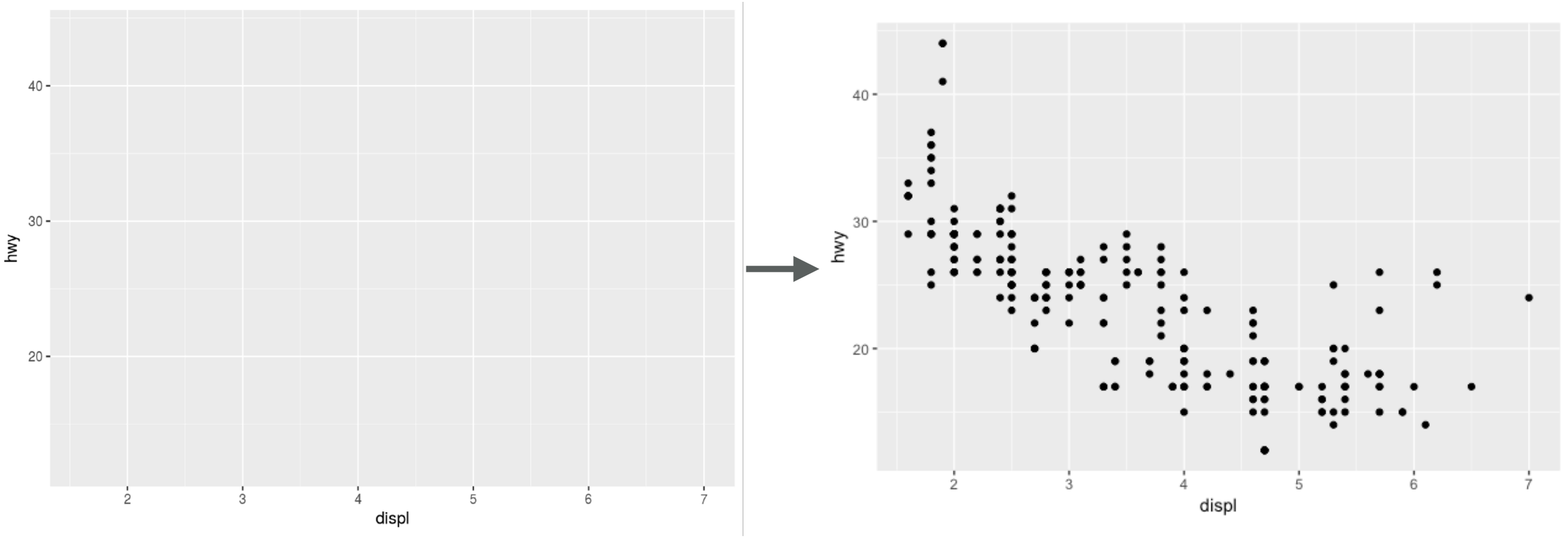
```
ggplot(data = <DATA>)
 + <GEOM_FUNCTION> ( mapping = aes( <MAPPINGS> ) )
```

```
# Building a Graph
# Base component of a Graph with ggplot2

g = ggplot()
g
```

```
# Specifying Aesthetics
# Describing the data with aes()...

g = ggplot(mpg) +
      aes(x = displ, y = hwy)

# Read as displ is mapped to x and hwy is mapped to y

g
```



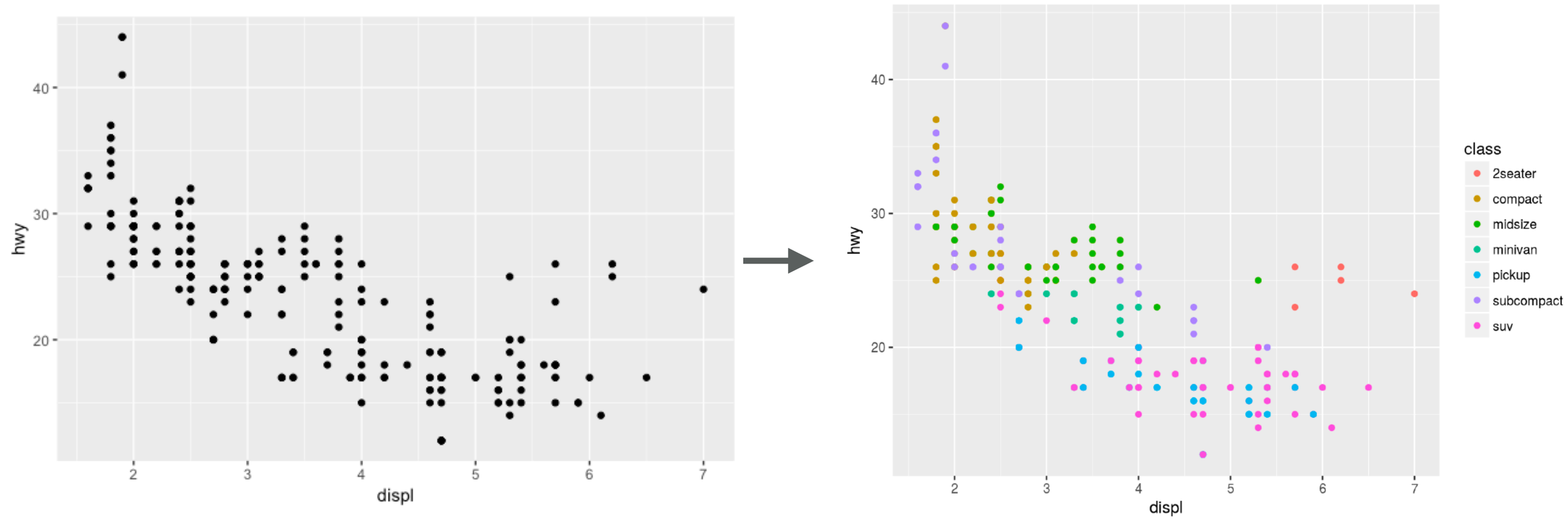
```
# Geometric Objects
# How data should be represented...
# Given by: geom_*()

g = ggplot(mpg) +
      aes(x = displ, y = hwy) +
      geom_point()

g
```



Improving Aesthetics

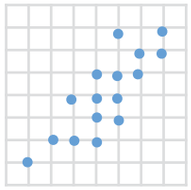
Adding color to make points stand out

```
g = ggplot(mpg) +
      aes(x = displ, y = hwy, colour = class) +
      geom_point()

g
```

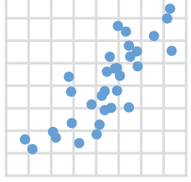


Common Geometries

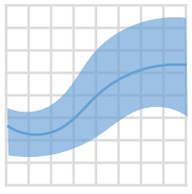
… kinds of graphs …



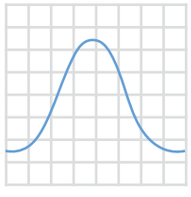
geom_point()



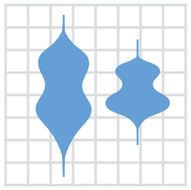
geom_jitter()



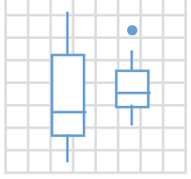
geom_smooth()



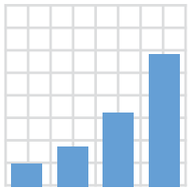
geom_density()



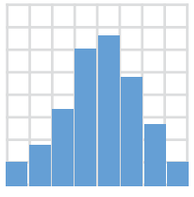
geom_violin()



geom_boxplot()



geom_bar()

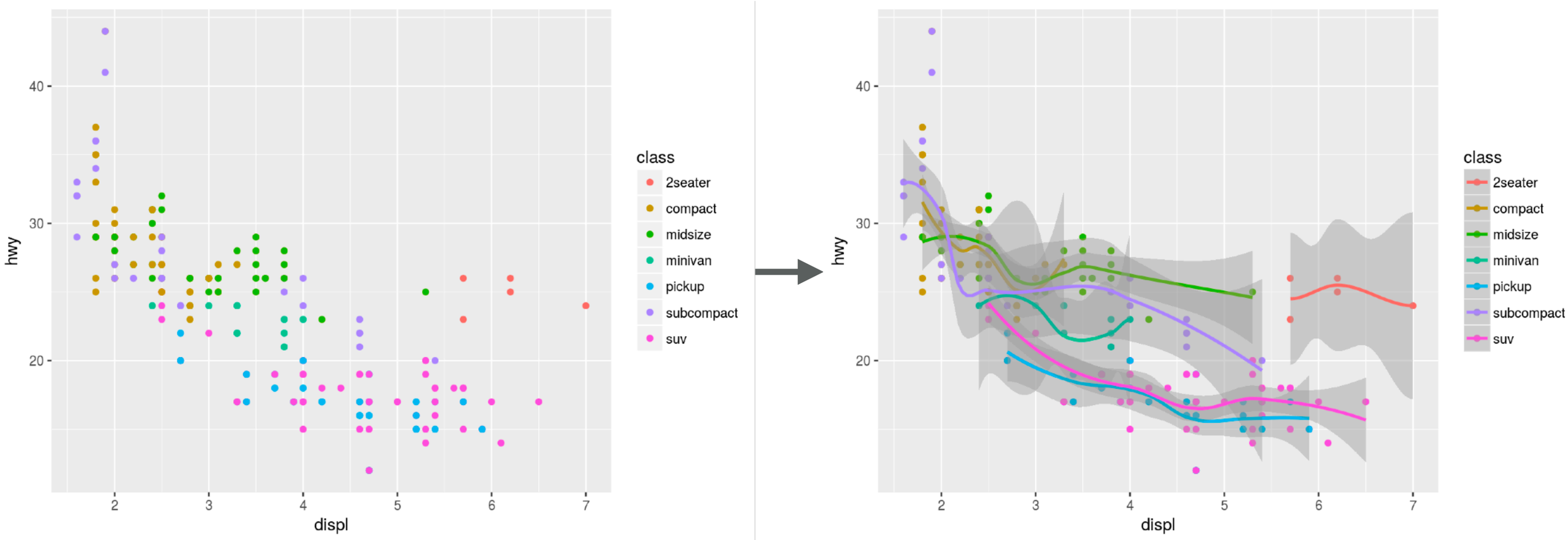


geom_histogram(binwidth = 5)

 # Multiple Geometric Objects

Layers affords the ability to overlay information

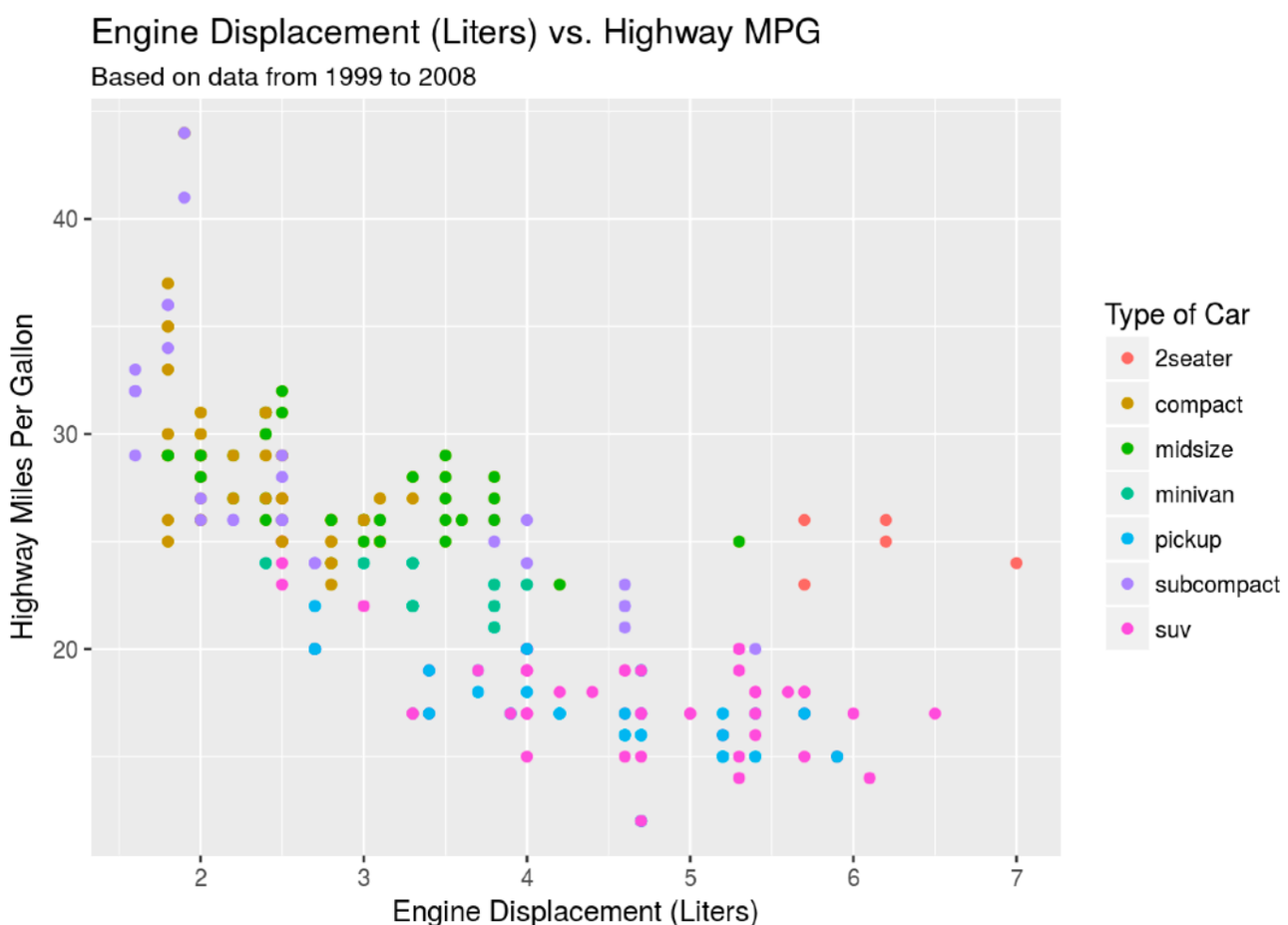
```
g = ggplot(mpg) +
      aes(x = displ, y = hwy, colour = class) +
      geom_point() +
      geom_smooth()
```



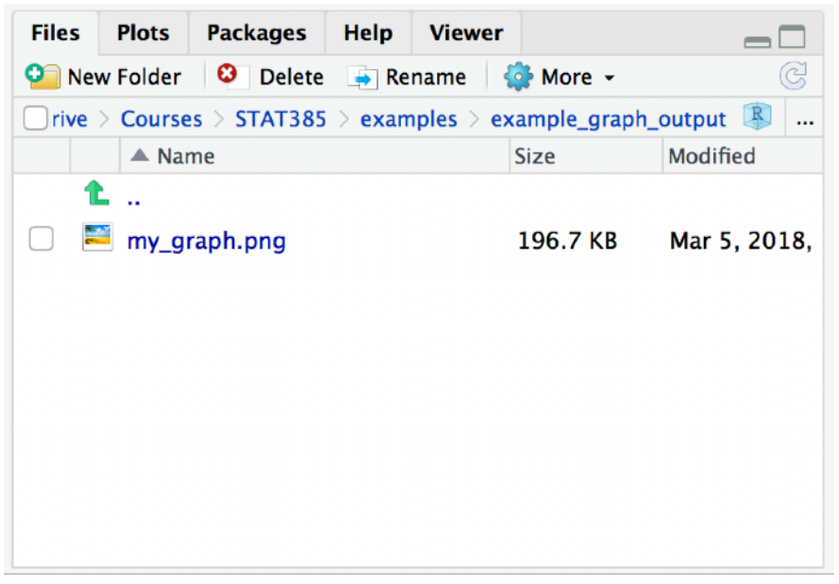
```r
# Layering without "Global" AES
# Mappings are important

g = ggplot(mpg) +
      geom_point(aes(x = displ, y = hwy, colour = class)) +
      geom_smooth()
g
```

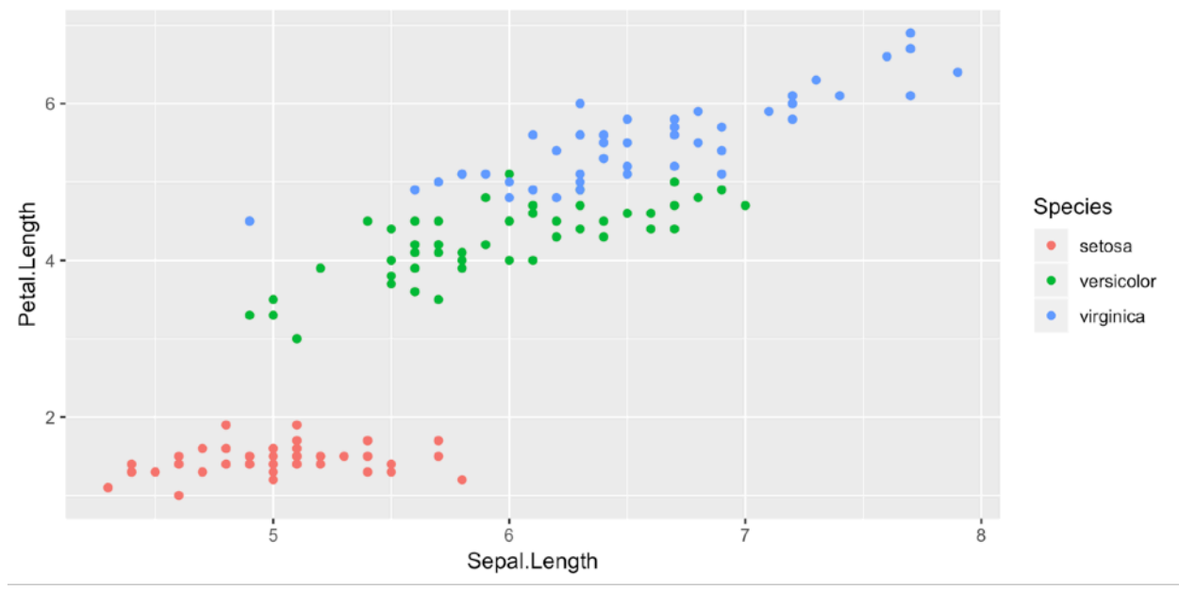
```
`geom_smooth()` using method = 'loess'
Error: stat_smooth requires the following missing aesthetics: x, y
```

```r
# Describing a Graph
# Adding labels and titles ...

g = ggplot(mpg) +
    aes(x = displ, y = hwy, color = class) +
    geom_point() +
    labs(title = "Engine Displacement (Liters) vs. Highway MPG",
            subtitle = "Based on data from 1999 to 2008",
            x = "Engine Displacement (Liters)",
            y ="Highway Miles Per Gallon",
            color = "Type of Car")
```



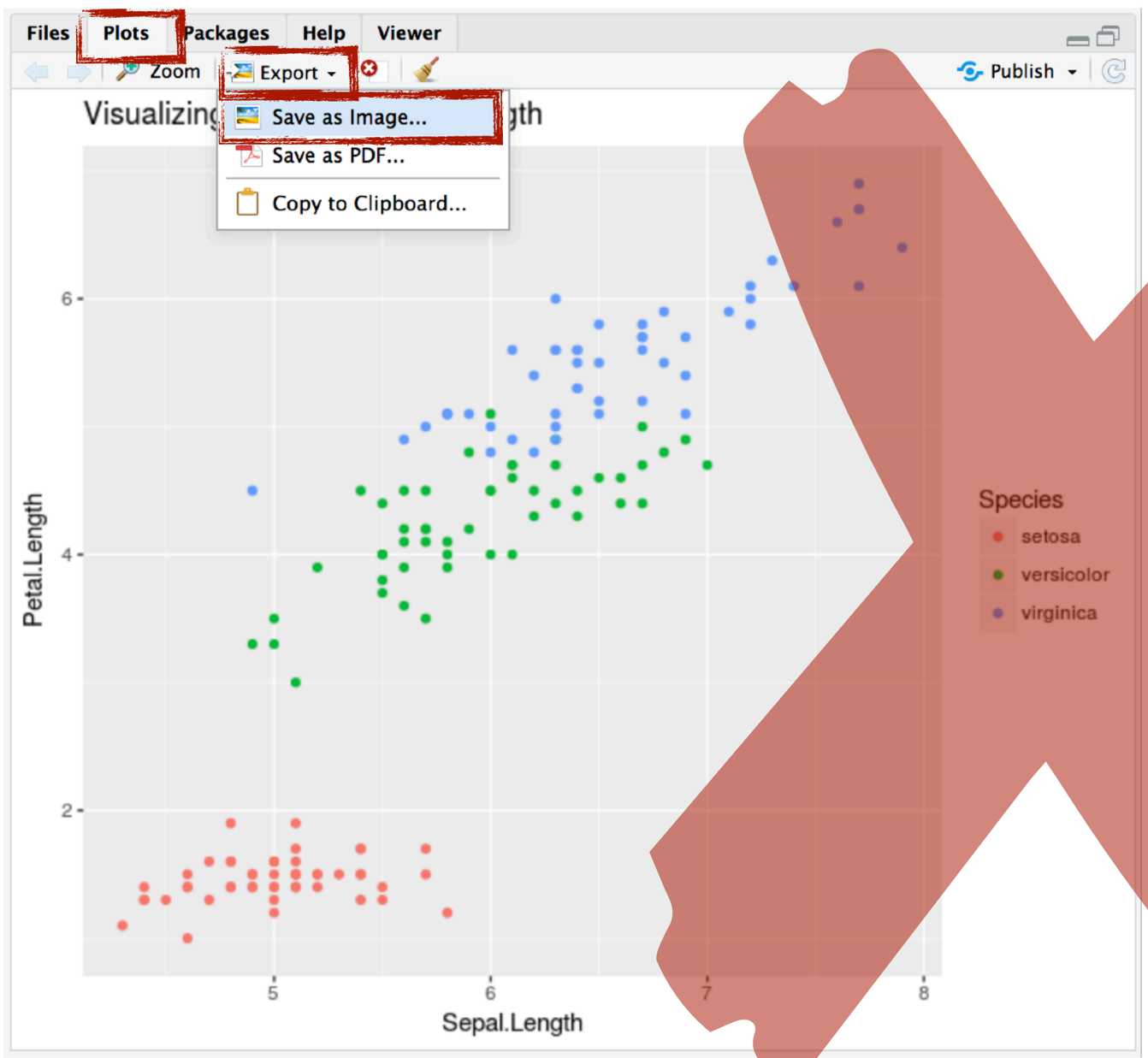
```r
# Exporting a graph via CLI
# Save a graph for use outside of R

# Create plot
g = ggplot(iris,
                mapping = aes(x = Sepal.Length, y = Petal.Length,
                              colour = Species)) +
   geom_point()

ggsave("my_graph.png", plot = g)
```

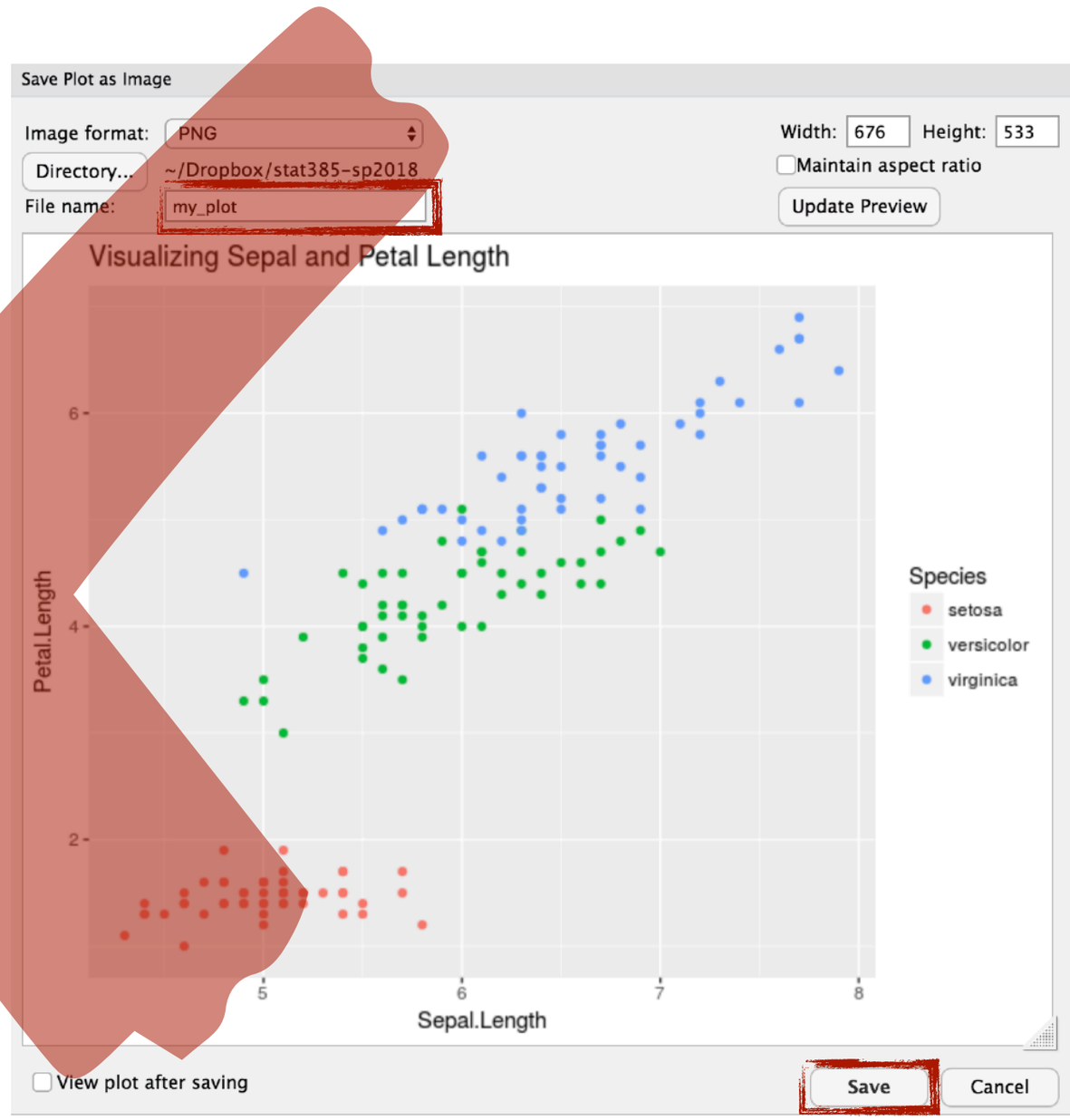




Exporting a Graph

… saving a graph for use elsewhere …





Making Graphs

Graphing Outline

… when to use a graph …

Variables

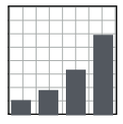
One Variable (Variation)

aes(**x = <?>**)

Continuous

Histogram
geom_histogram(binwidth = 5)

Categorical or Discrete

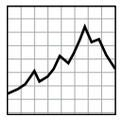


Barplot
geom_bar()

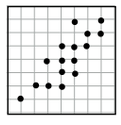
Two Variables (Covariation)

aes(**x = <?> , y = <?>**)

Indep: Continuous, Dep: Numerical

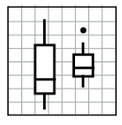


Line Graph
geom_line()



Scatterplot
geom_point()

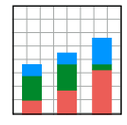
Indep: Categorical, Dep: Numerical



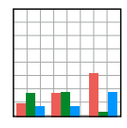
Boxplot
geom_boxplot()

Violin Graph
geom_violin()

Indep: Categorical, Dep: Categorical



Stacked Barplot
geom_bar(position="stack")



Side-by-side Barplot
geom_bar(position="dodge")

Independent
Dependent

Motivating Question

… what to explore ???

Source

How many bikes are rented daily?

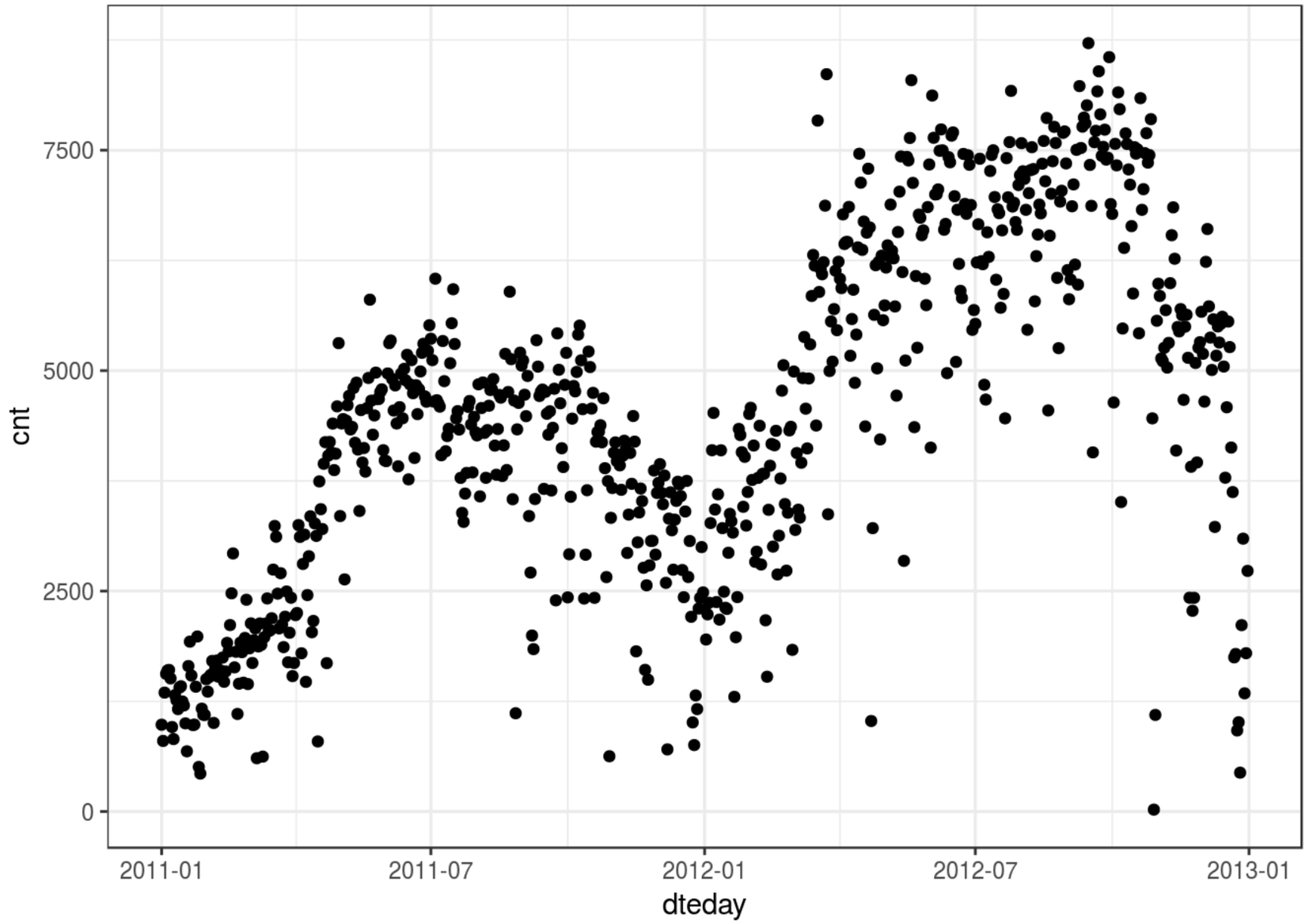
Example Data

… bike_sharing_daily data in ucidata …

```
# install.packages("devtools")
# devtools::install_github("coatless/ucidata")
library("ucidata")

head(bike_sharing_daily)
```

Scatterplot

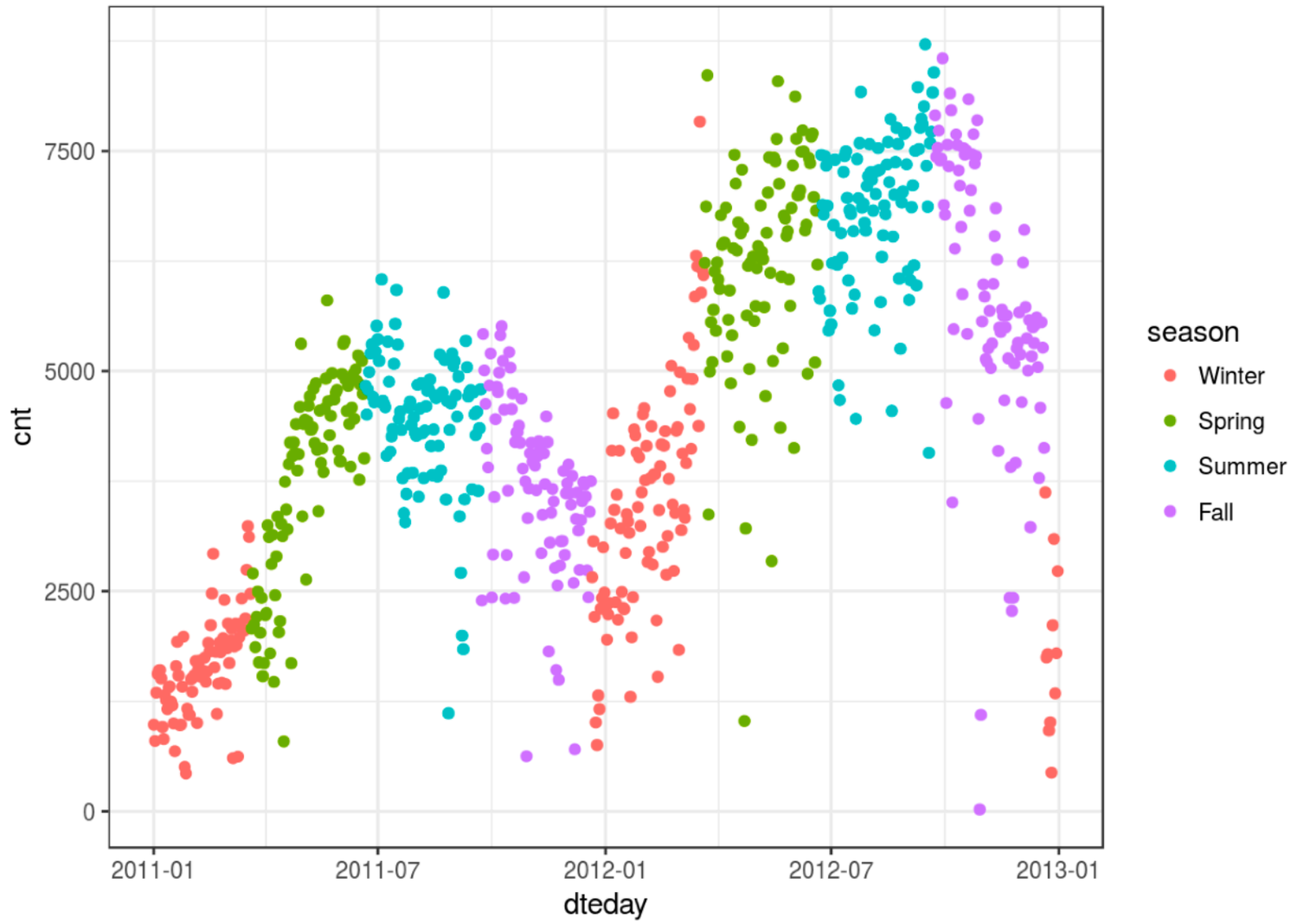
… *two* **numerical** variables as points to observe *covariation* …




```
ggplot(data = bike_sharing_daily,
       mapping = aes( x = dteday, y = cnt)) + geom_point( )
```

Colored Scatterplot

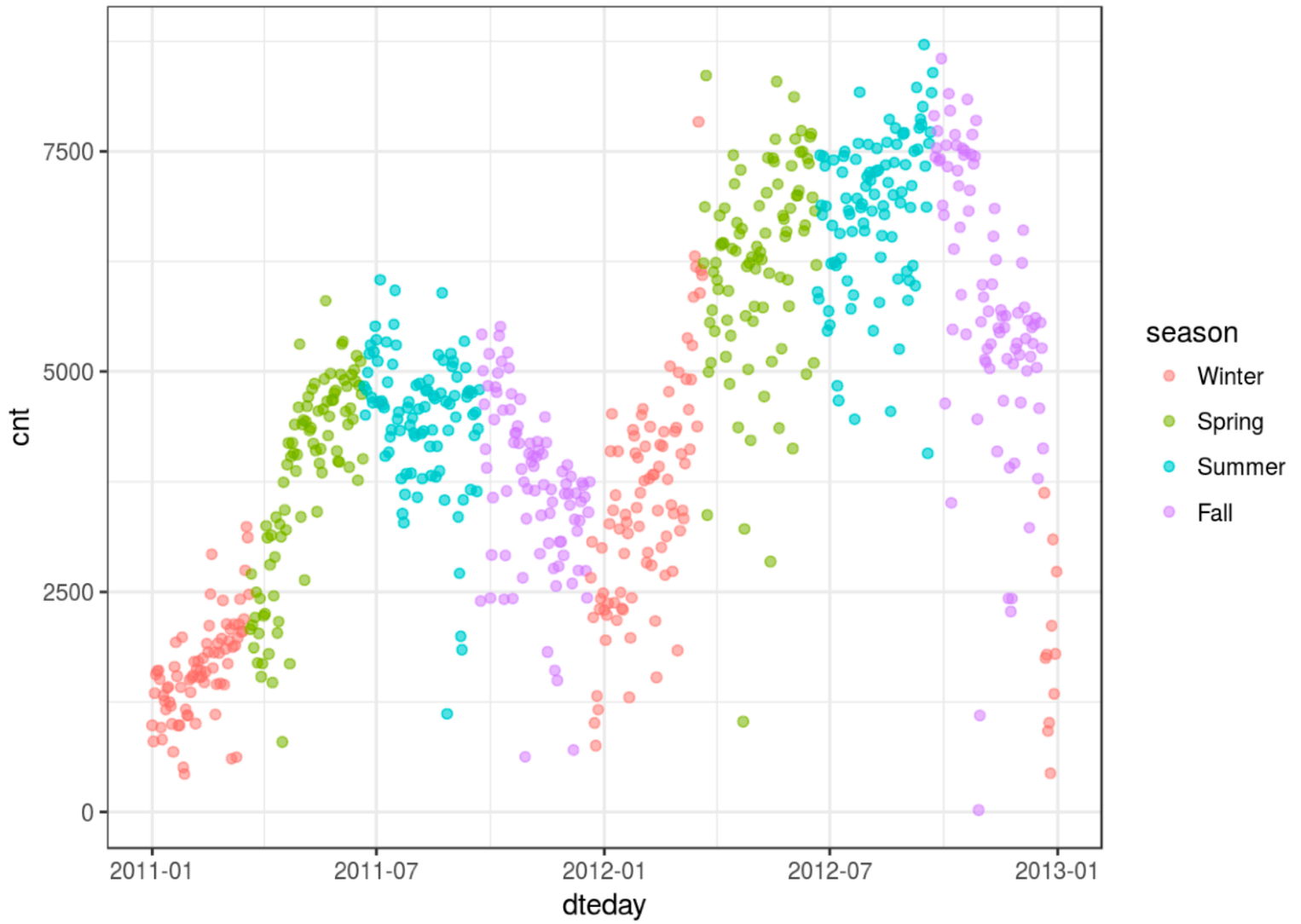
… *two* **numerical** variables with colored groupings …




```
ggplot(bike_sharing_daily) +
        aes( x = dteday, y = cnt, color = season) +
        geom_point()
```

Redux Colored Scatterplot

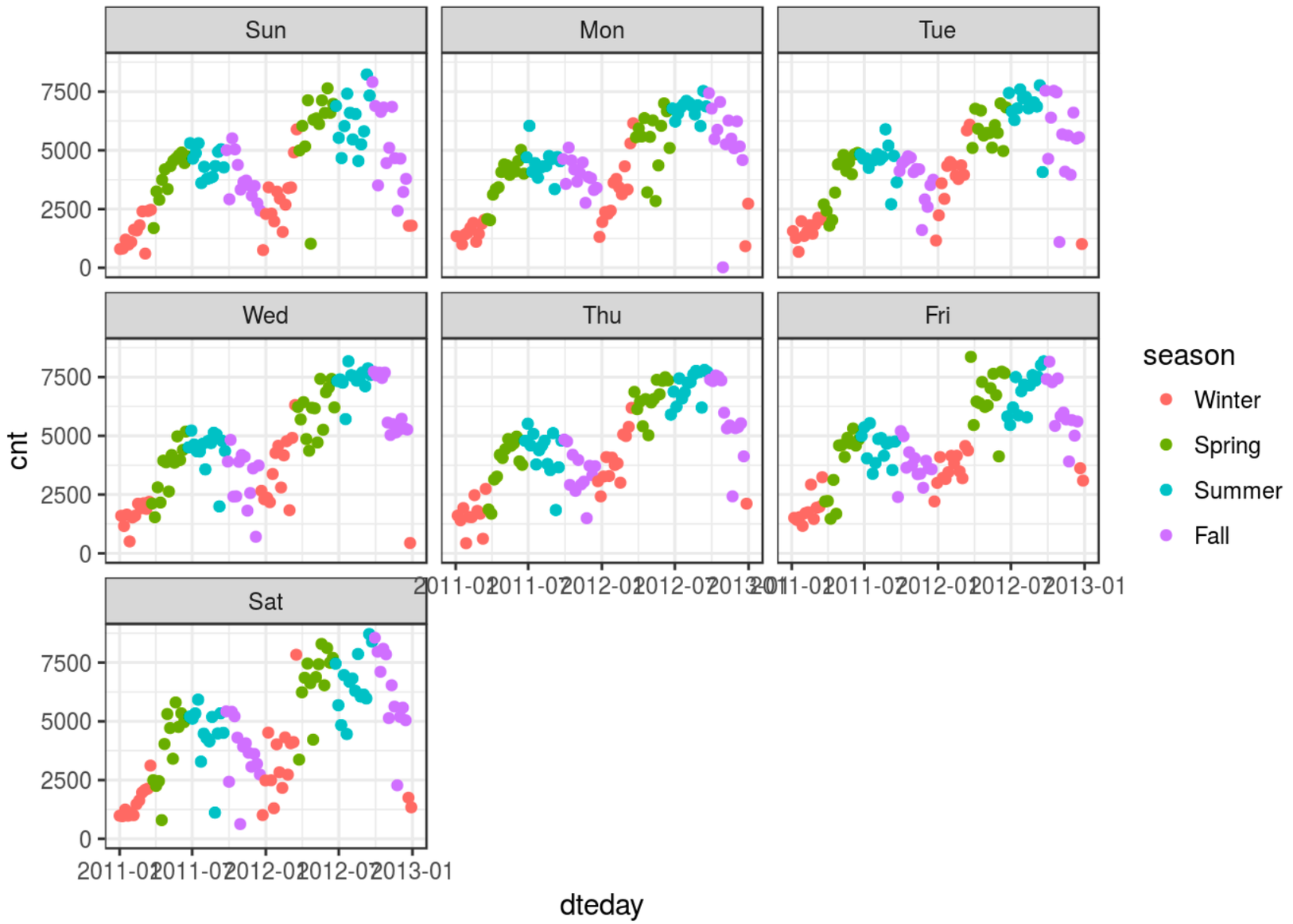
… what changed between plots ???




```
ggplot(bike_sharing_daily) +
        aes( x = dteday, y = cnt, color = season) +
        geom_point( alpha = 0.50 )
```

Facetting

… conditionally subsetting and plotting data ..




```
ggplot(data = bike_sharing_daily) +
        aes( x = dteday, y = cnt, color = season) +
        geom_point( ) + facet_wrap( ~ weekday)
```

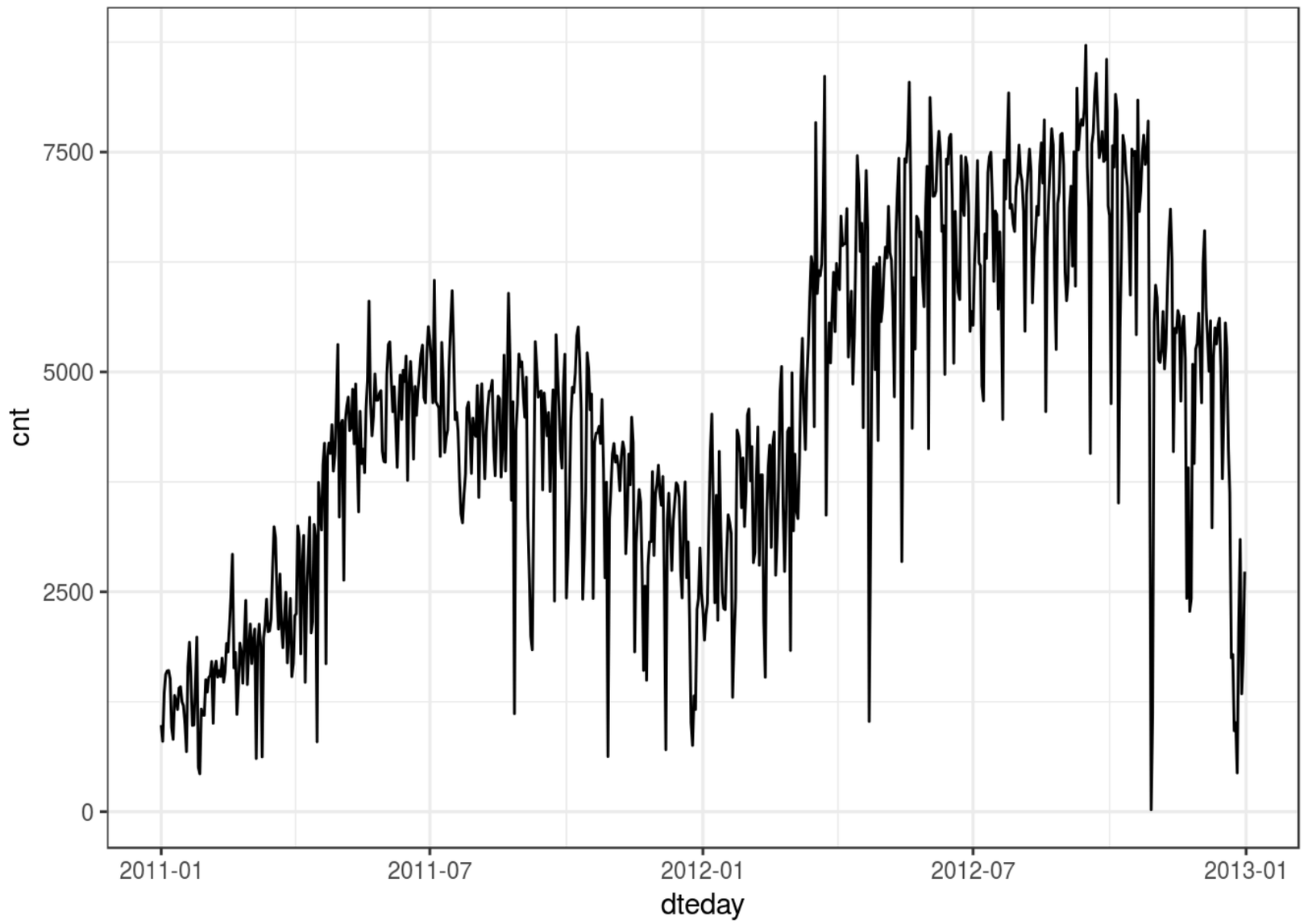
Your Turn

Modify the following so that it creates a scatter plot of
dteday vs. **atemp**
without a transparency

```
ggplot(bike_sharing_daily) +
  aes( x = dteday, y = cnt, color = season) +
  geom_point( alpha = 0.05)
```

Line Graph

… connecting points together on a plot …




```
ggplot(data = bike_sharing_daily) +
        aes( x = dteday, y = cnt) + geom_line( )
```

Your Turn

Modify the following so that it creates a **line** plot of **dteday** vs. **temp** without a transparency

```
ggplot(bike_sharing_daily) +
        aes( x = dteday, y = cnt, color = season) +
        geom_point( alpha = 0.05)
```

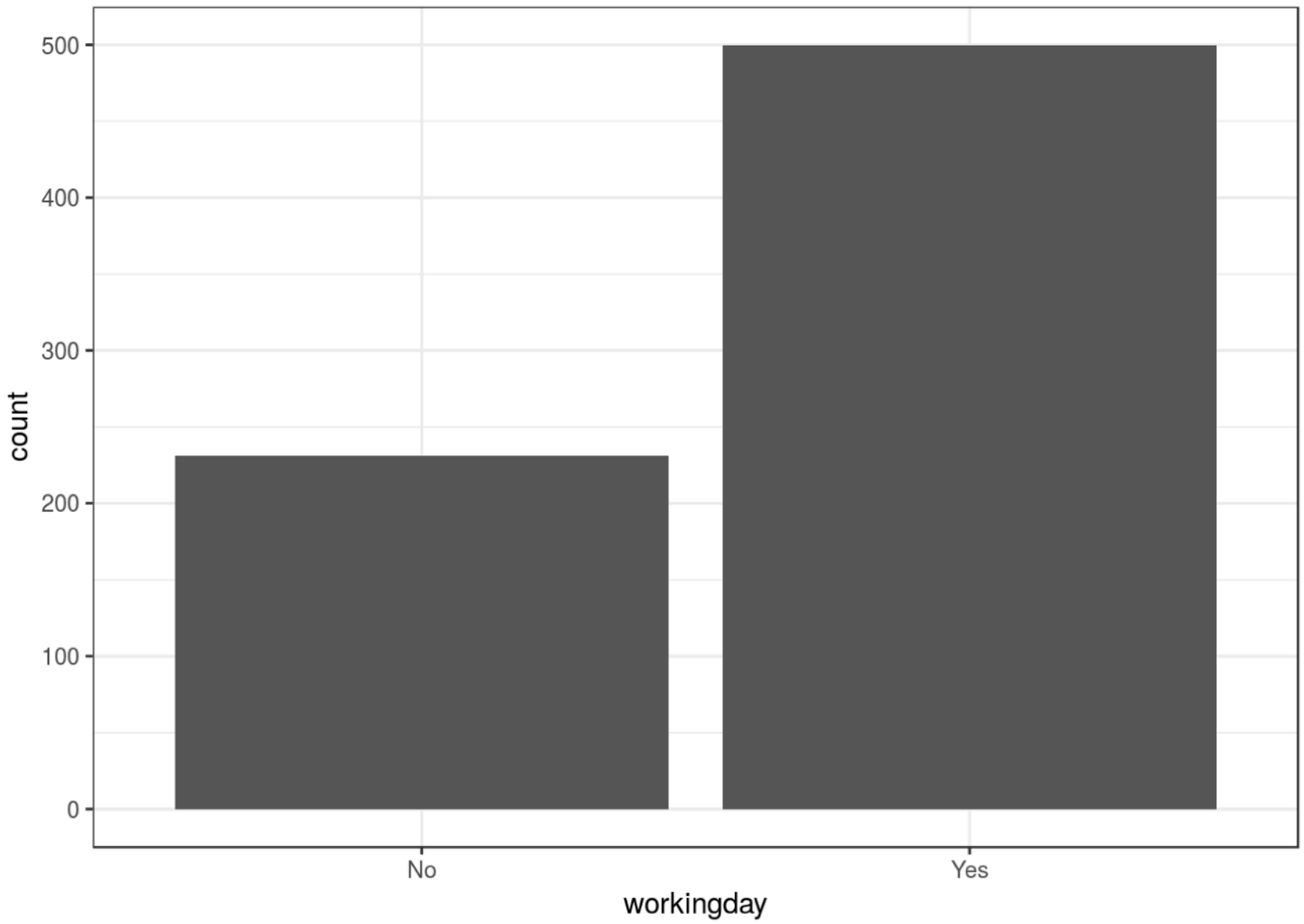
Tabulating Frequency

… comparing *variation* of one **categorical** …

```
table(bike_sharing_daily$workingday)
#   No Yes
# 231 500
```

Bar Plot

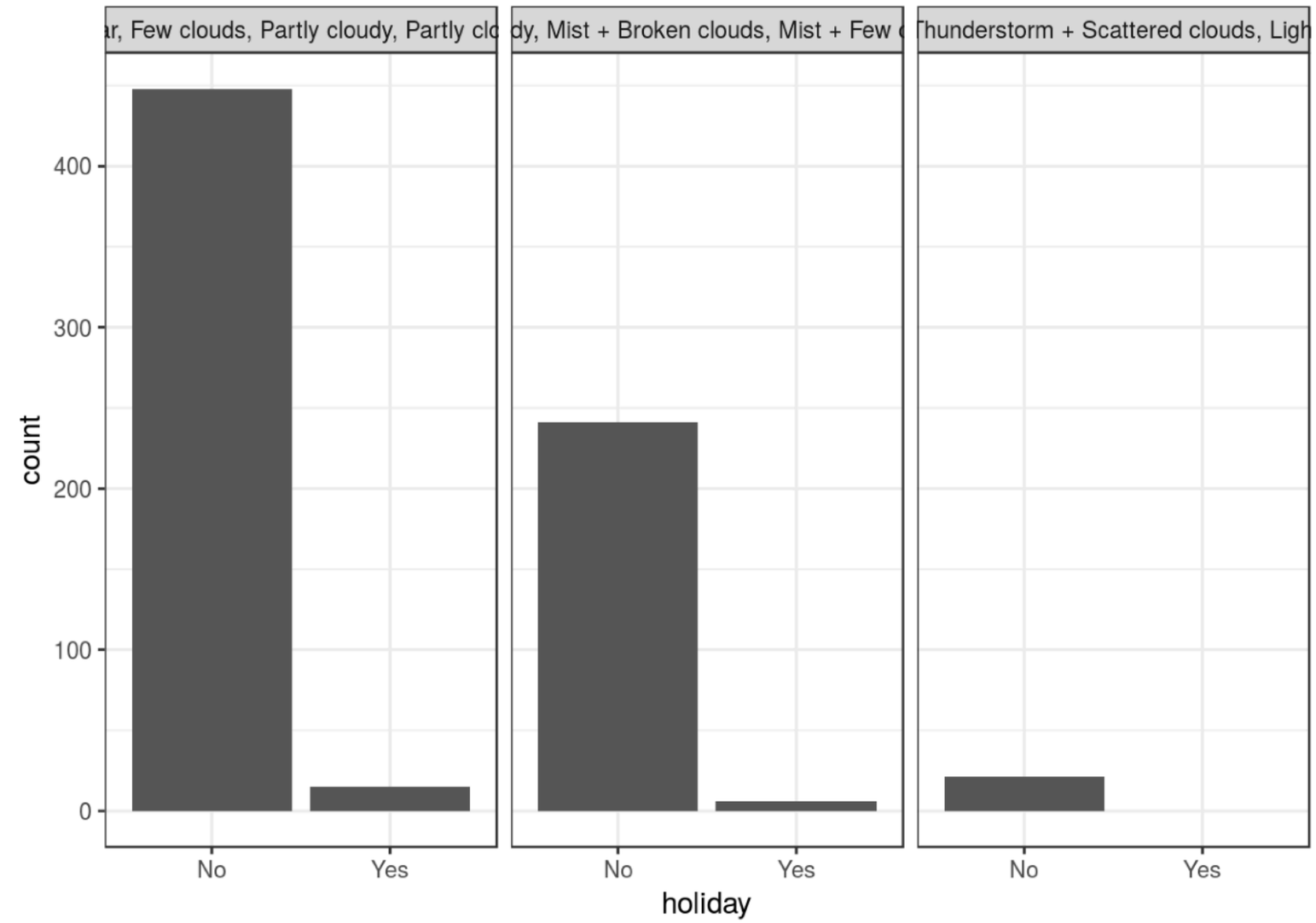
… comparing *variation* of one **categorical** …




```
ggplot(bike_sharing_daily) +
  aes( x = workingday ) +
  geom_bar( )
```

Facetted Bar Plot

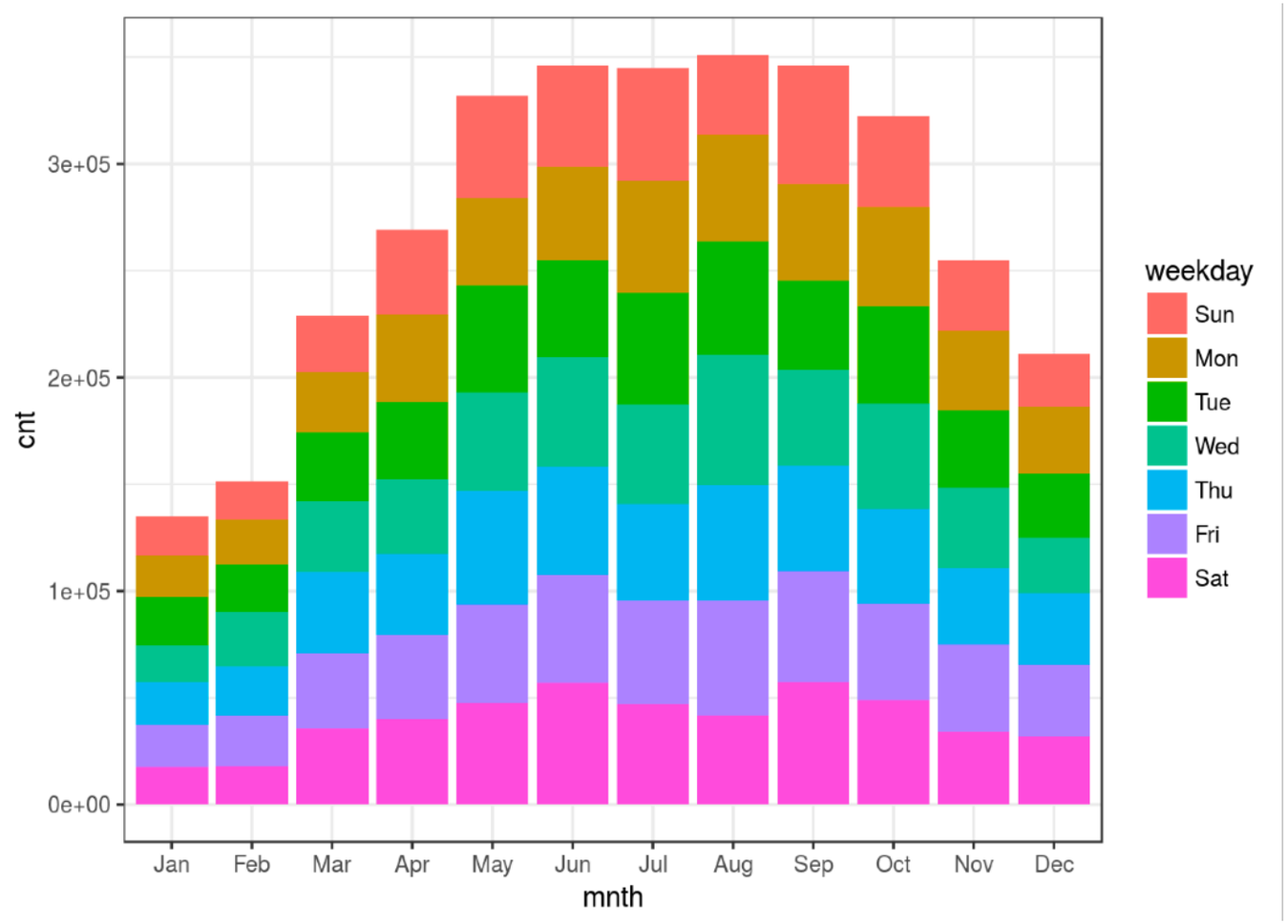
… comparing categorical frequency between two variables …




```
ggplot(data = bike_sharing_daily) +
  aes( x = holiday ) +
  geom_bar( ) + facet_wrap( ~ weathersit)
```

Stacked Bar Plot

… comparing amounts across groups …




```
ggplot(data = bike_sharing_daily) +
  aes( x = mnth, y = cnt, fill = weekday) +
  geom_bar( stat = "identity" )
```

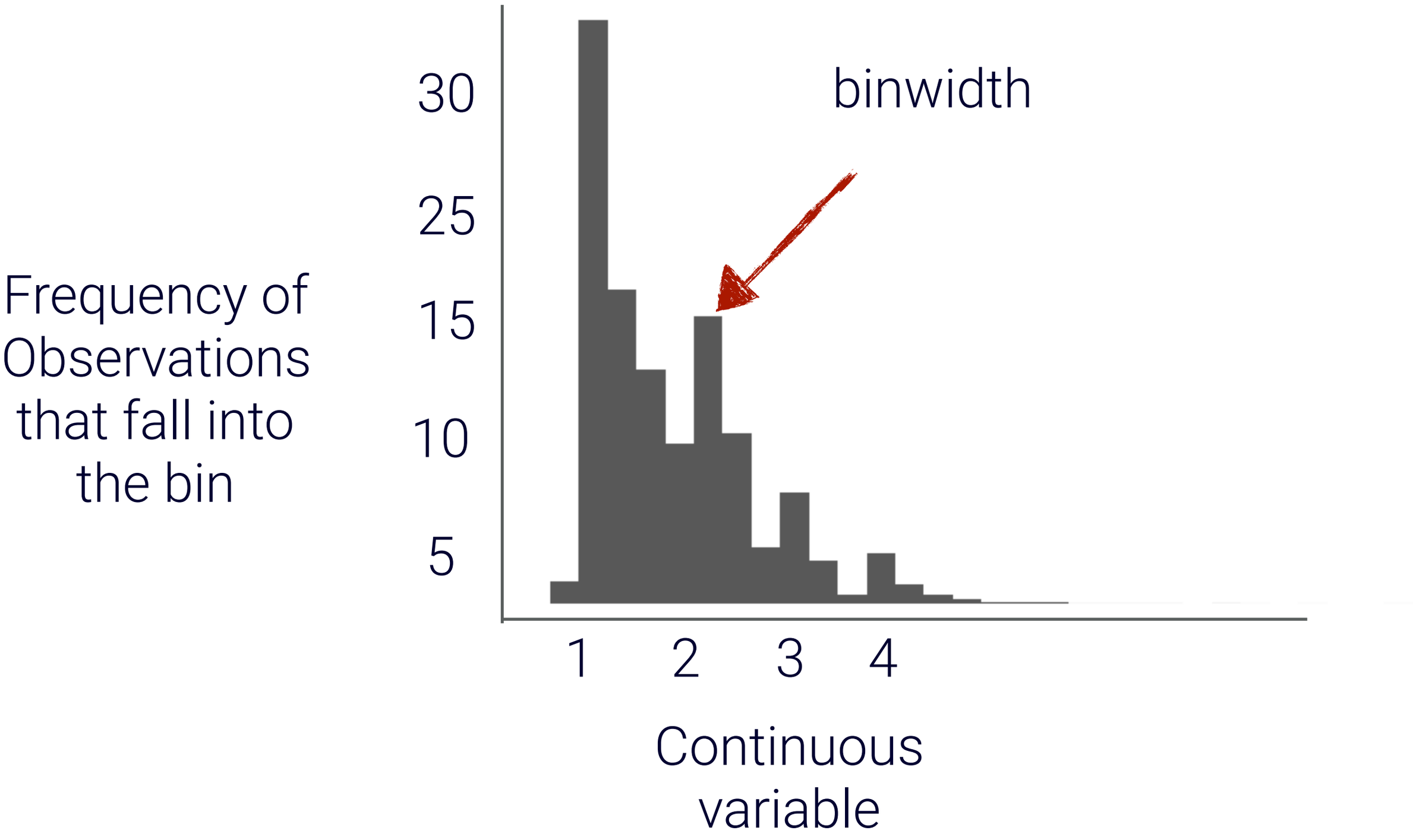
Your Turn

Modify the following code so that the bar plot is **facetted** by **yr**

```
ggplot(data = bike_sharing_daily) +
  aes( x = mnth, y = cnt, fill = weekday) +
  geom_bar( stat = "identity" )
```

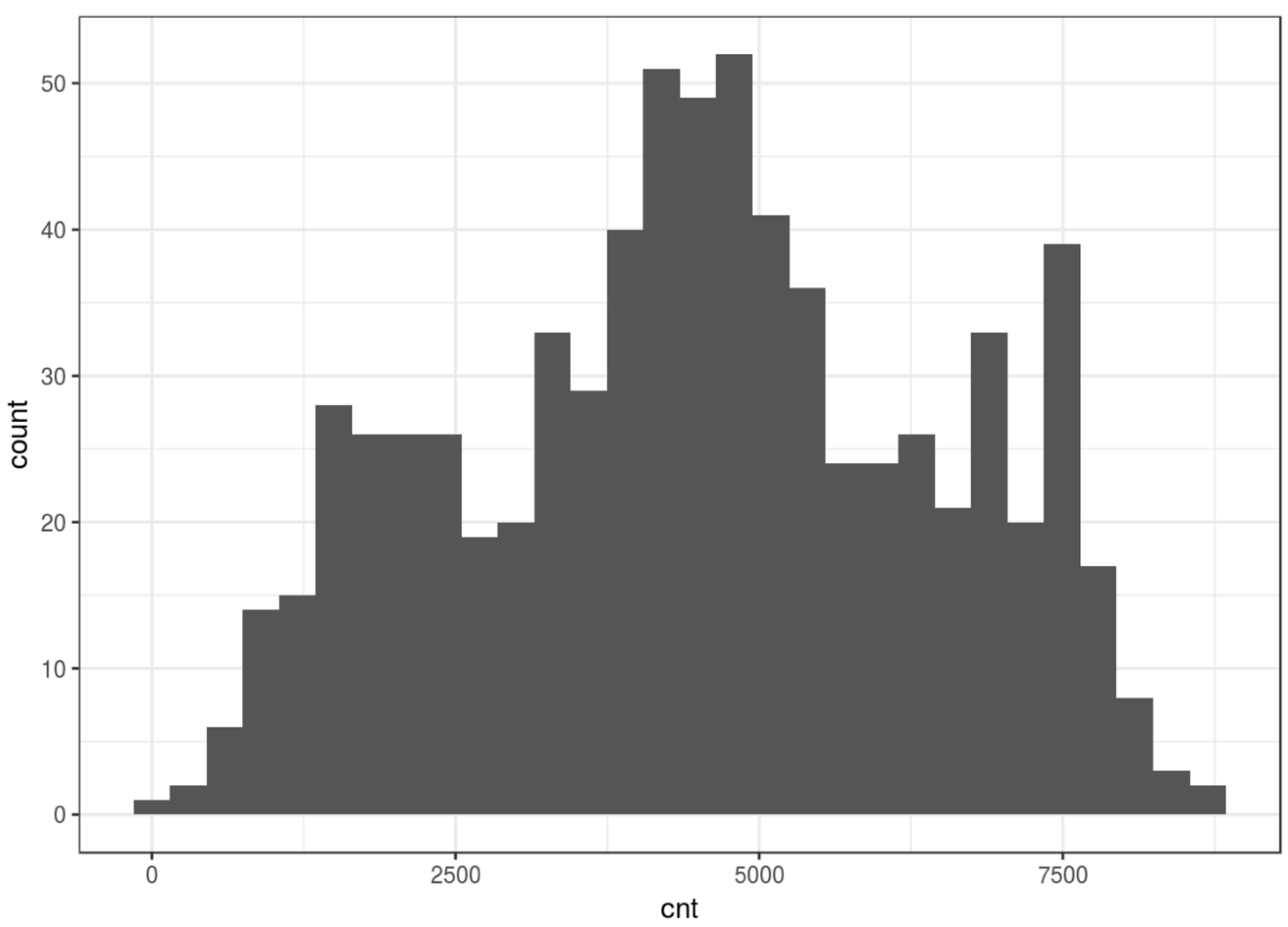
Histogram

… **continuous** data for one variable …



Histogram

… **continuous** data for one variable …

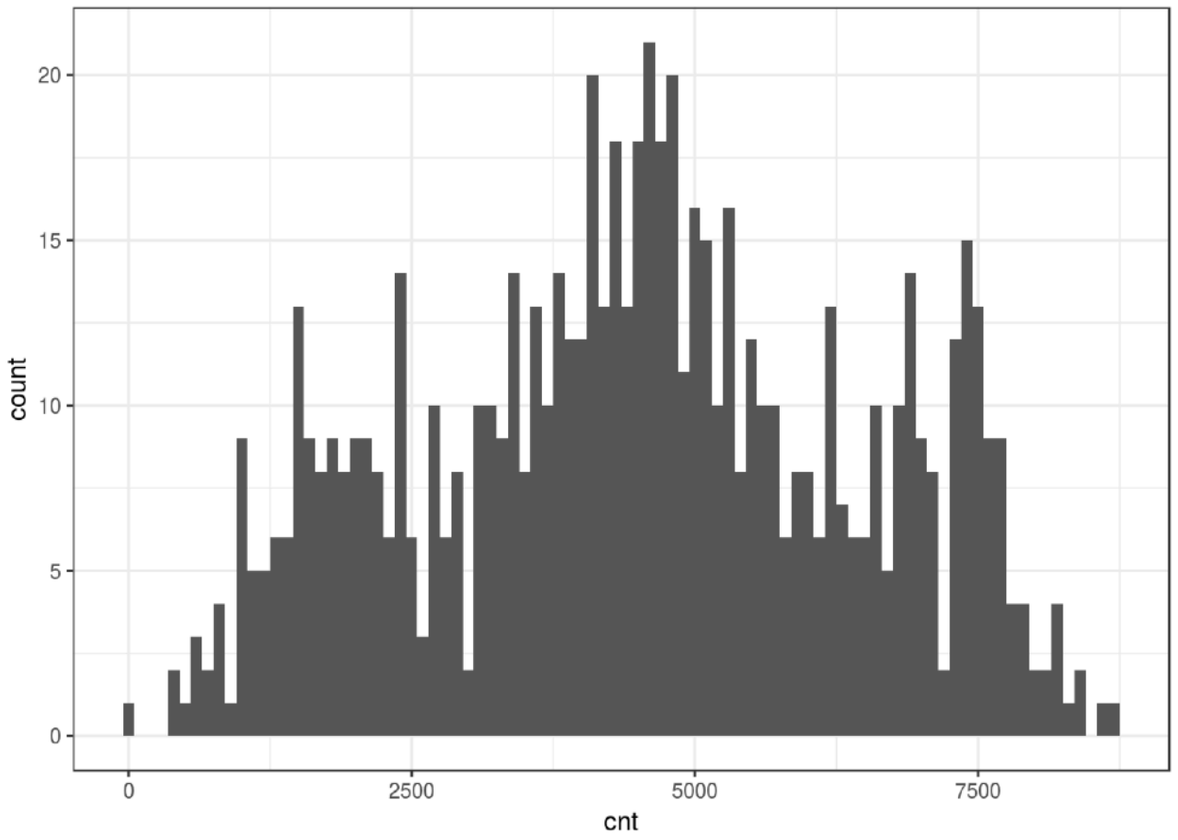



```
ggplot(data = bike_sharing_daily,
       mapping = aes( x = cnt ) ) +
  geom_histogram( )
```

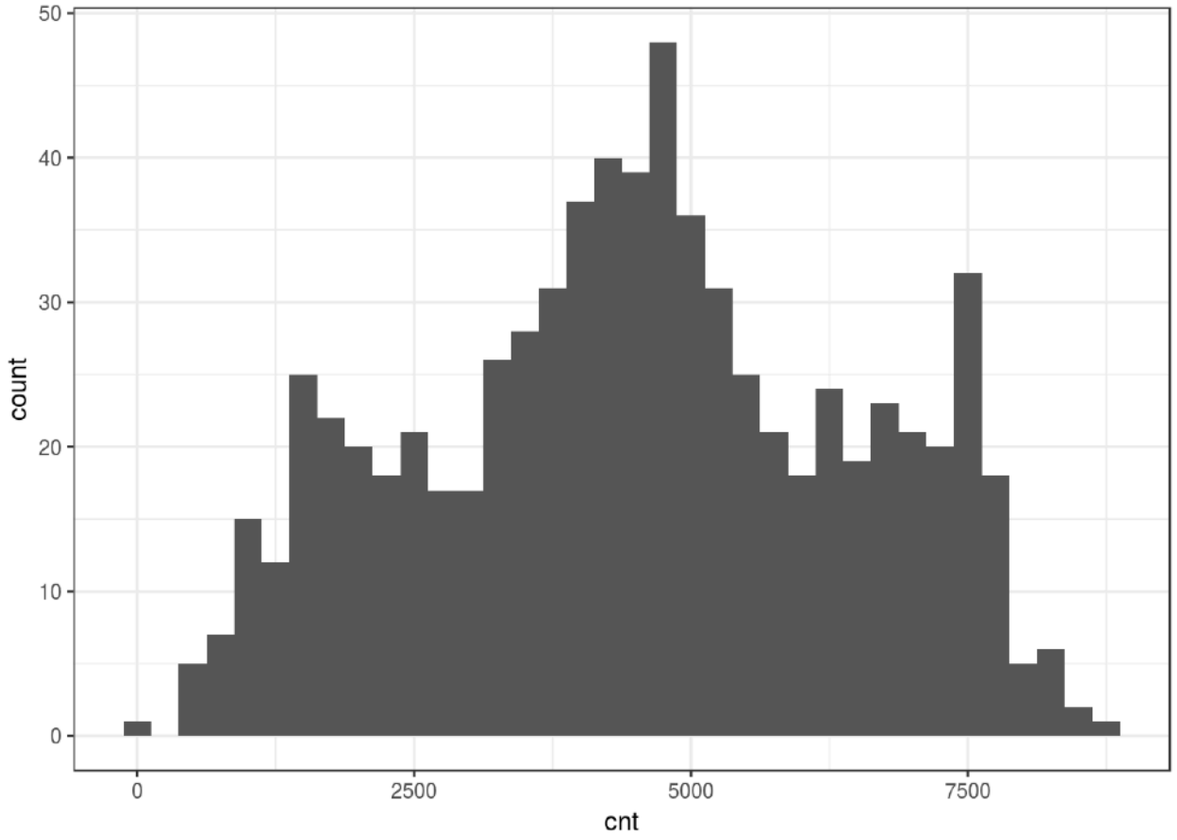
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Binwidth Changes

… **continuous** data for one variable …



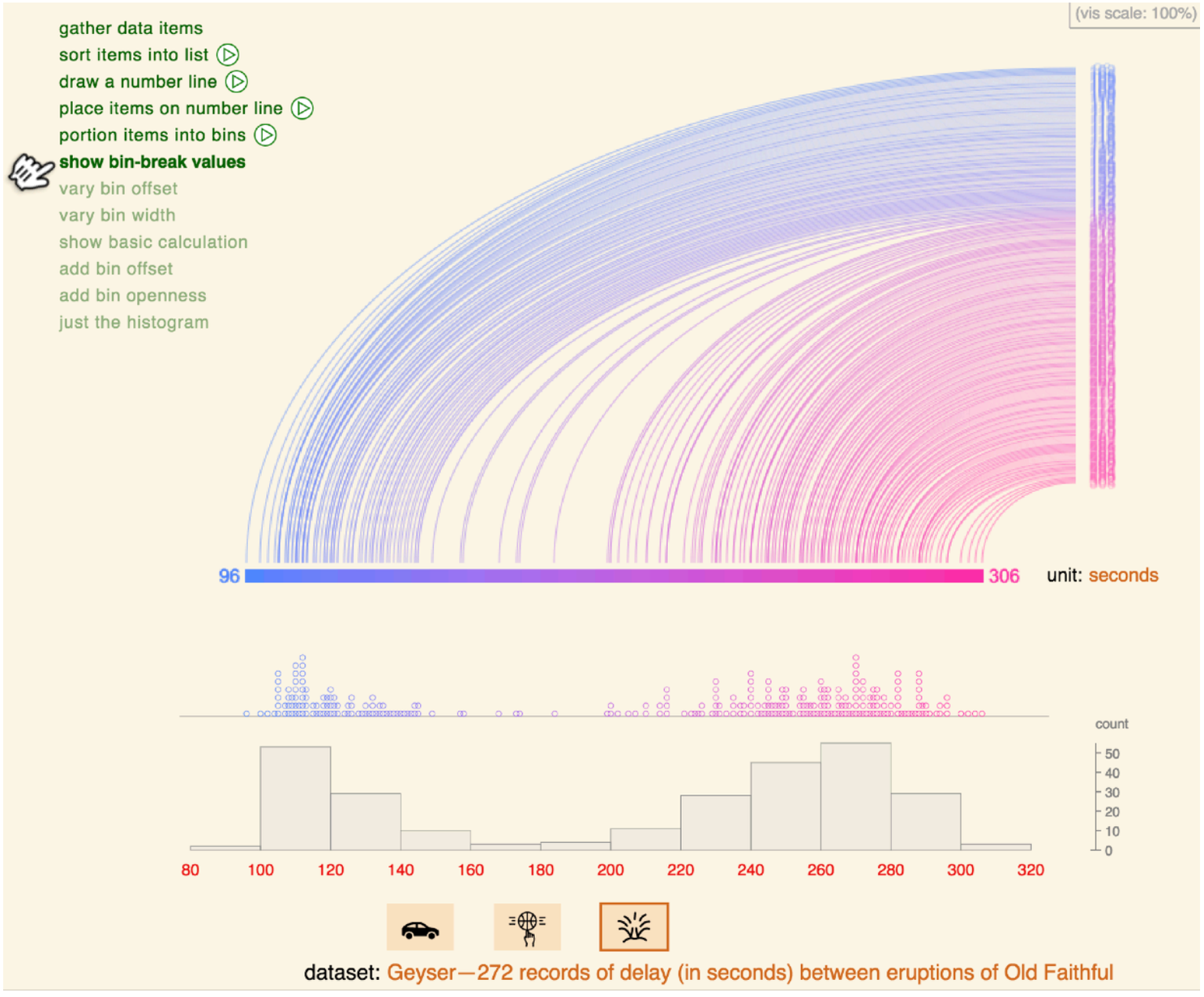
geom_histogram(**binwidth = 100**)



geom_histogram(**binwidth = 250**)

Histograms

imagined "bins" for continuous data



https://tinlizzie.org/histograms/

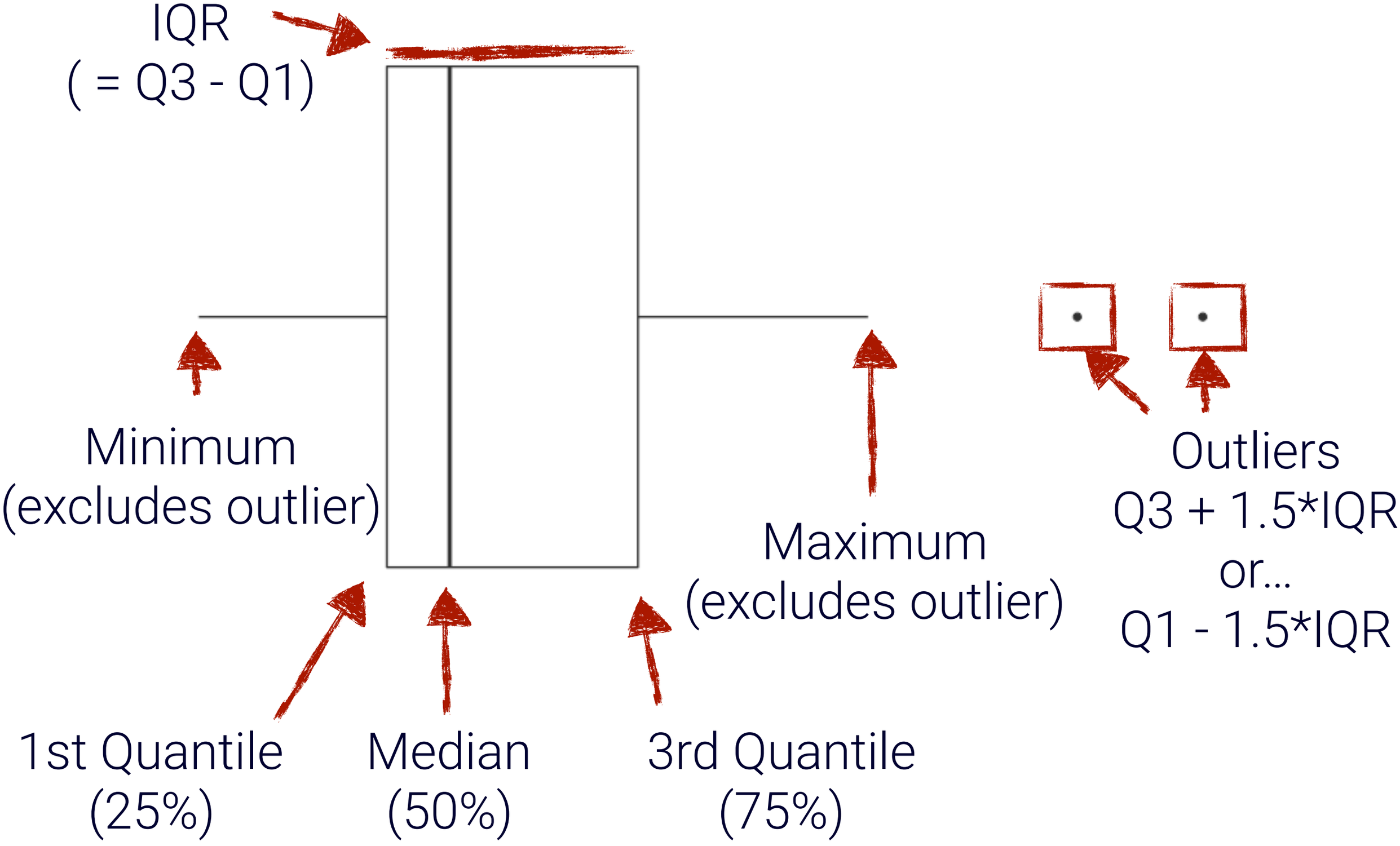
Your Turn

Create a histogram that shows what the
atemp is per **season**
by using the **fill** aesthetic

What happens if you use **color**?

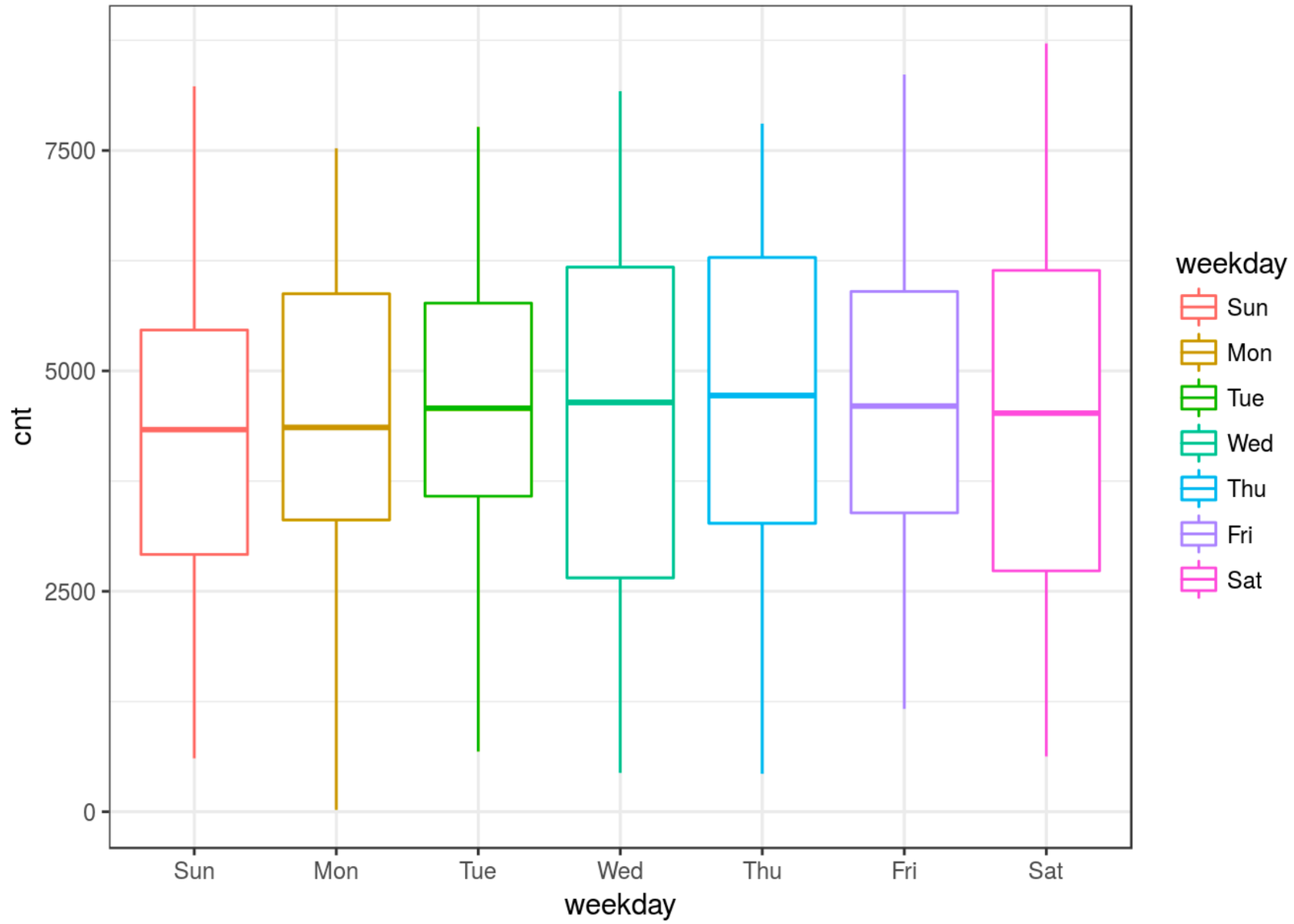
Boxplot

... **categorical** paired with **numerical** to observe *covariation*...



Boxplot

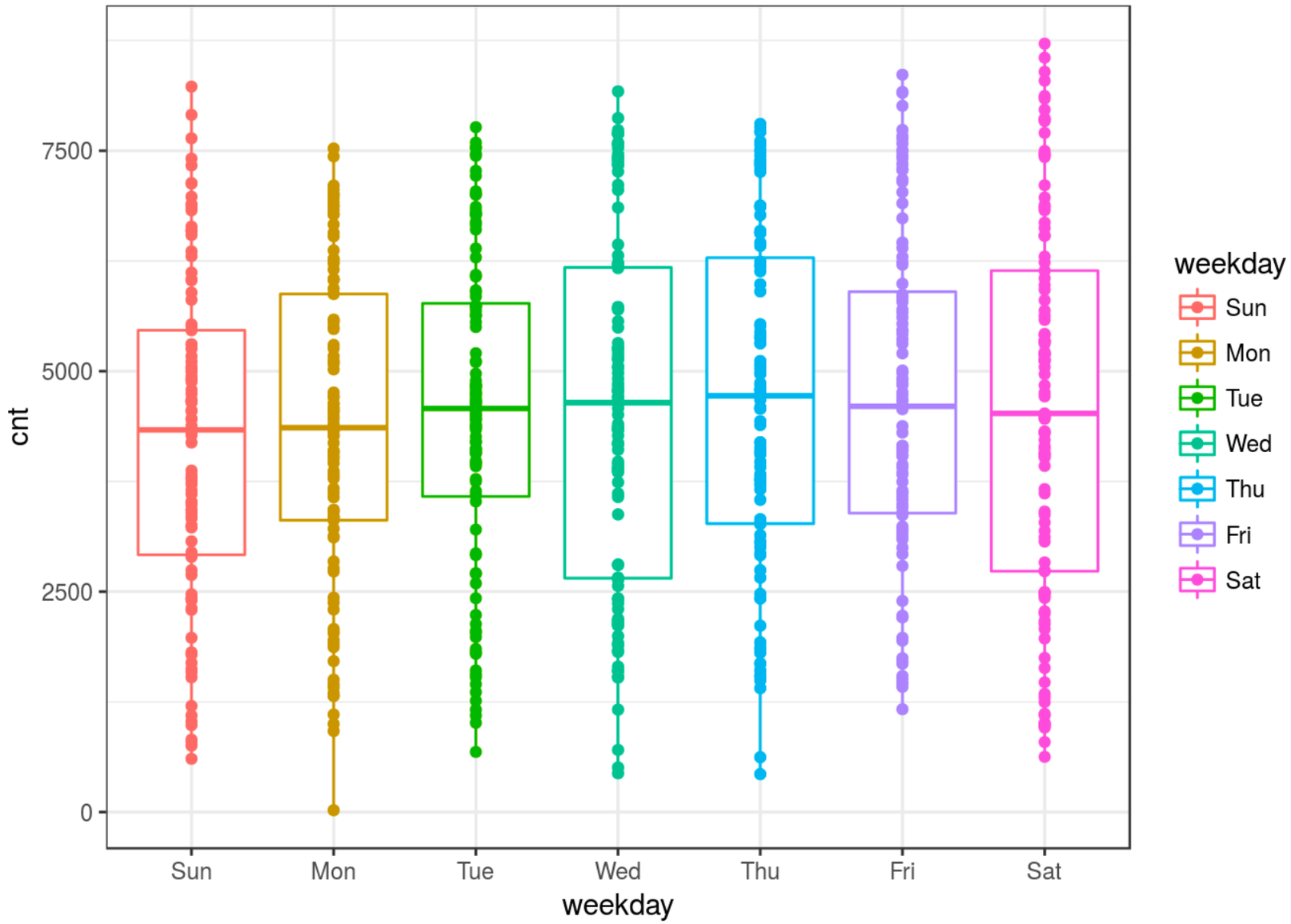
… **categorical** paired with **numerical** to observe *covariation*…




```
ggplot(data = bike_sharing_daily) +
  aes( x = weekday, y = cnt, color = weekday)+
  geom_boxplot( )
```

Boxplot with Points

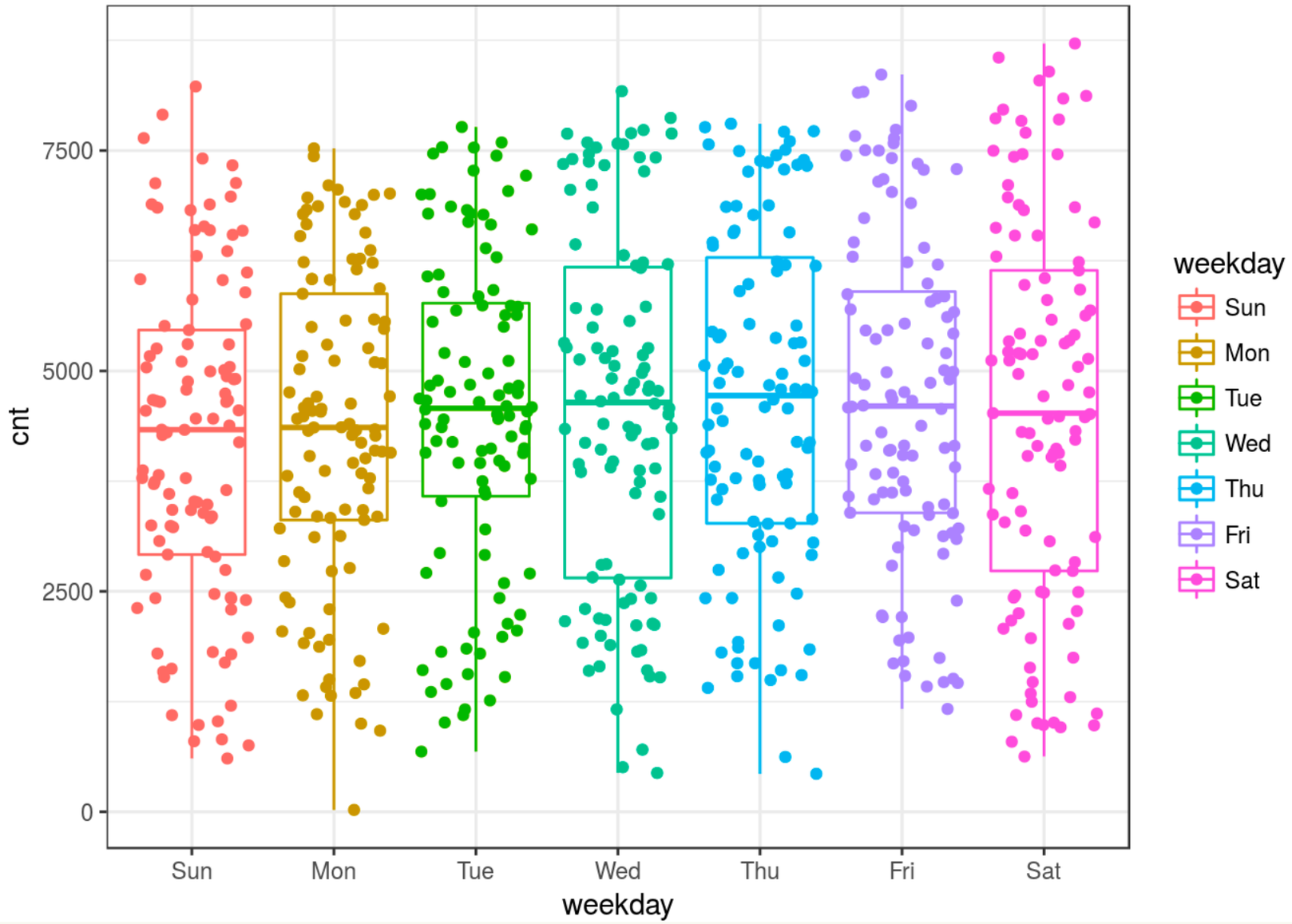
… **categorical** paired with **numerical** to observe *covariation*…




```
ggplot(data = bike_sharing_daily) +
  aes( x = weekday, y = cnt, color = weekday) +
  geom_boxplot( ) + geom_point()
```

Boxplot with Points ???

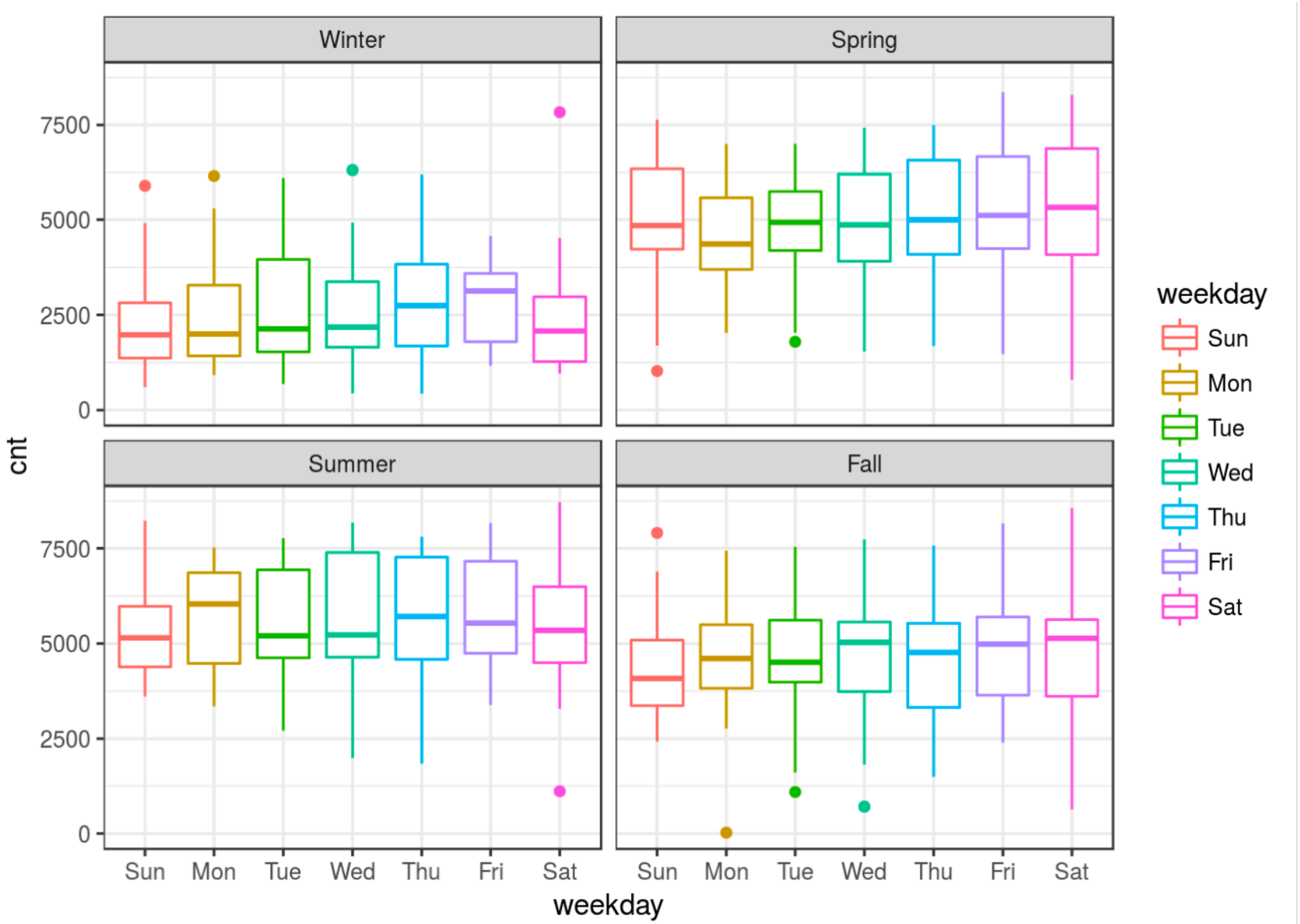
… **categorical** paired with **numerical** to observe *covariation*…




```
ggplot(data = bike_sharing_daily) +
  aes( x = weekday, y = cnt, color = weekday) +
  geom_boxplot( ) + geom_jitter()
```

Facetted Boxplots

… **categorical** paired with **numerical** split by a **categorical** …




```
ggplot(data = bike_sharing_daily) +
  aes( x = weekday, y = cnt, color = weekday) +
  geom_boxplot( ) + facet_wrap( ~ season)
```

Establishing a Theme

Importance of a Theme

… unified visualizations …

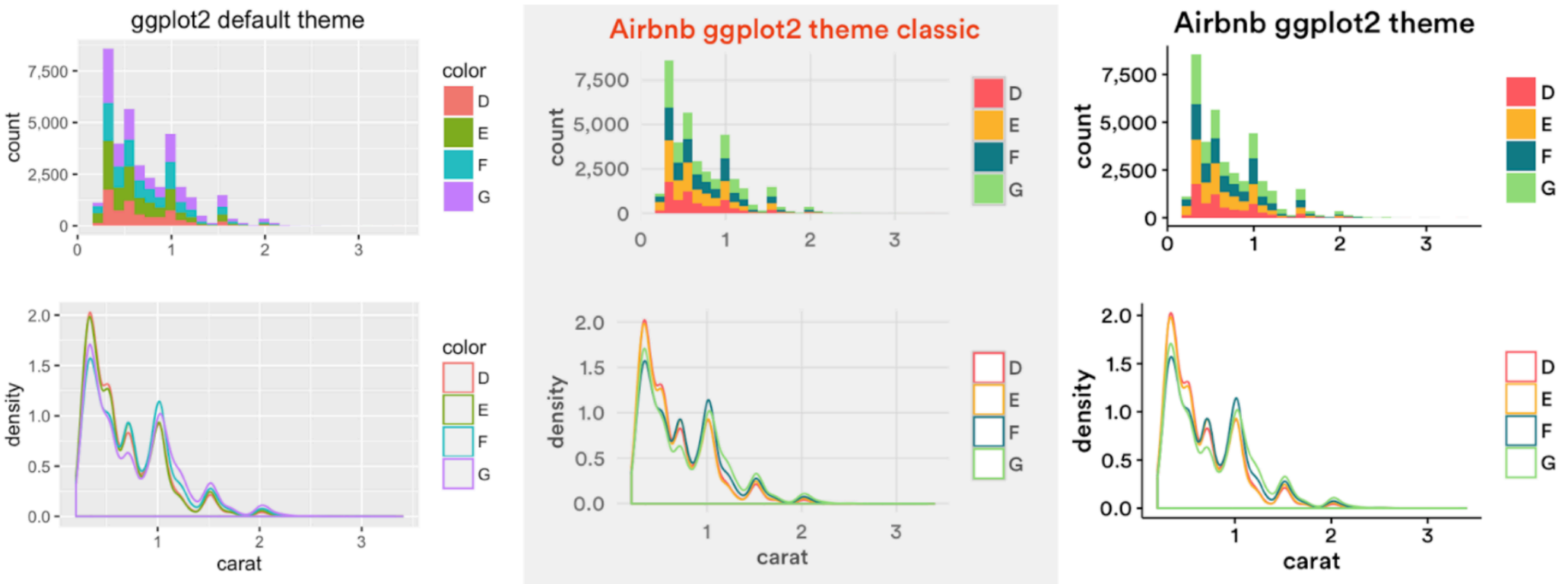
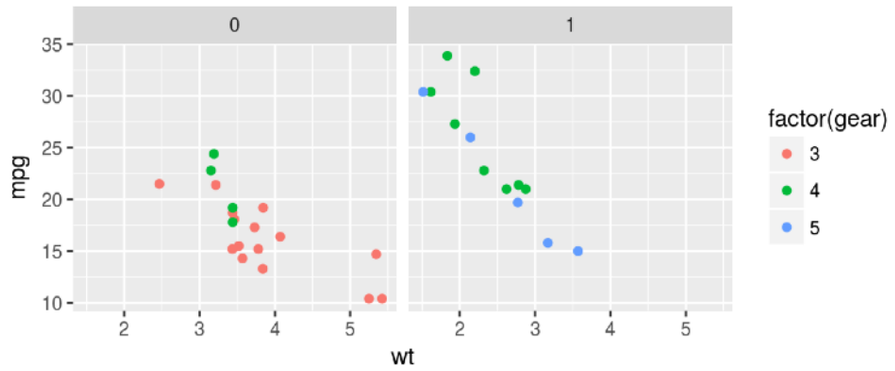


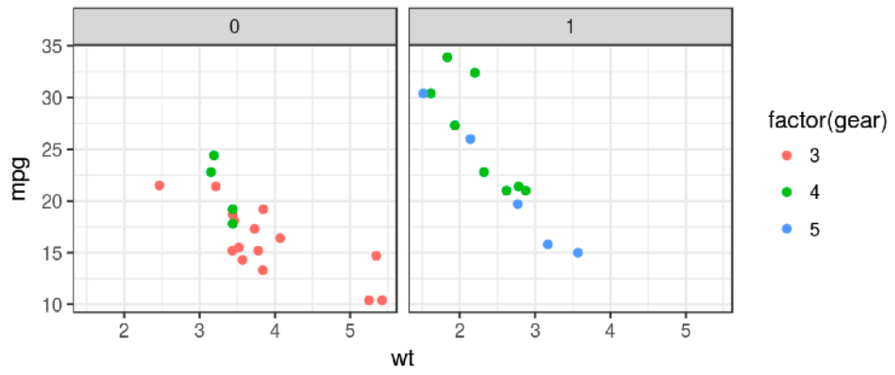
Figure 3: Example of Airbnb branded ggplot2 themes and scales to standardize our data visualizations. These extensions create a consistent internal data science brand, and can be found on Github. Taken with permission from 'Using R packages and education to scale data science at Airbnb.'

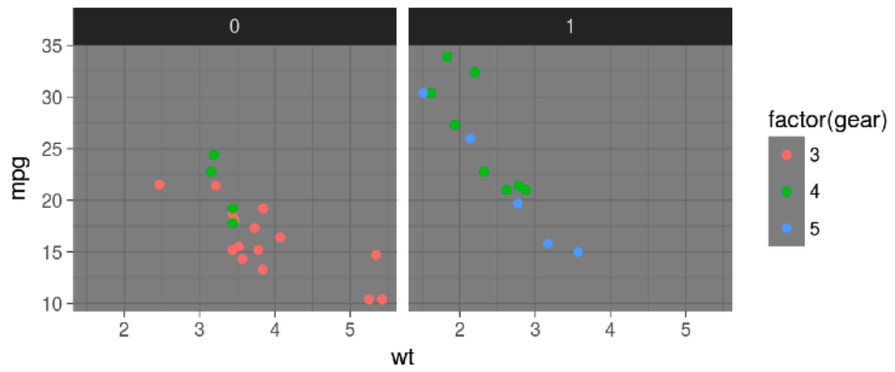
Practical Data Science Collection: How R Helps Airbnb Make the Most of Its Data

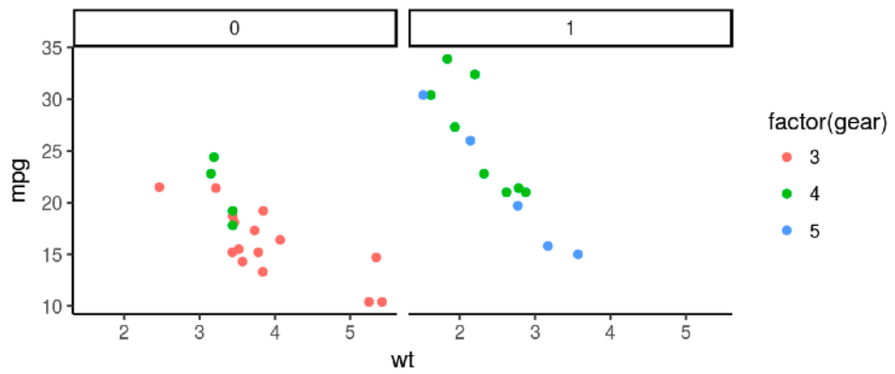
Preloaded Themes

… changing a plot's look …










```
# Base Plot
g = ggplot(mtcars) +
 geom_point(
 aes(x = wt, y = mpg,
       color = factor(gear))
  ) + facet_wrap(~am)

# Theme Options
g + theme_gray() # default
g + theme_dark()
g + theme_bw()
g + theme_classic()
```

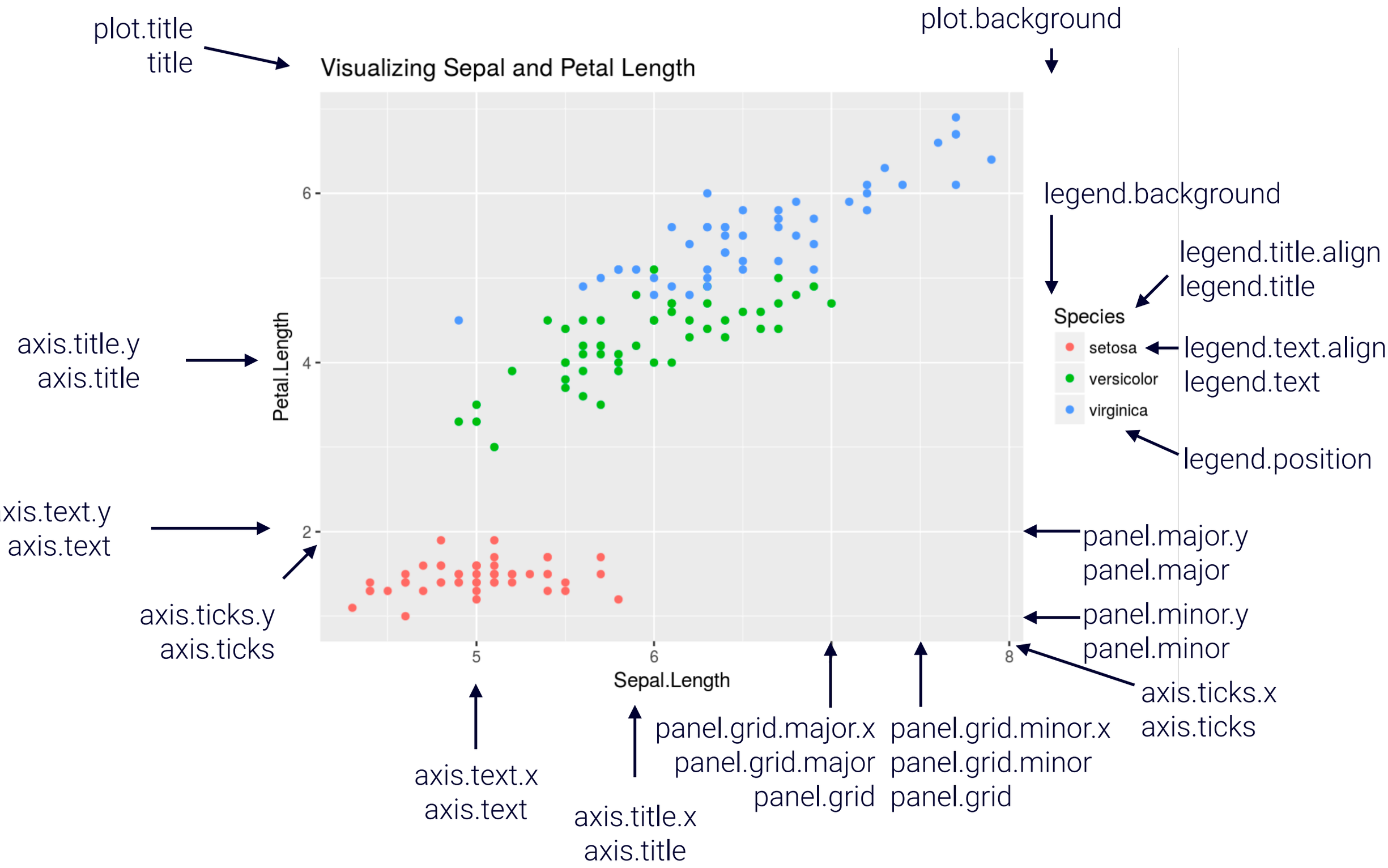
Theme Elements

... graph construction ..

Element	Description
element_blank()	Draws nothing and no space
element_rect()	Borders and Backgrounds
element_line()	Lines
element_text()	Text

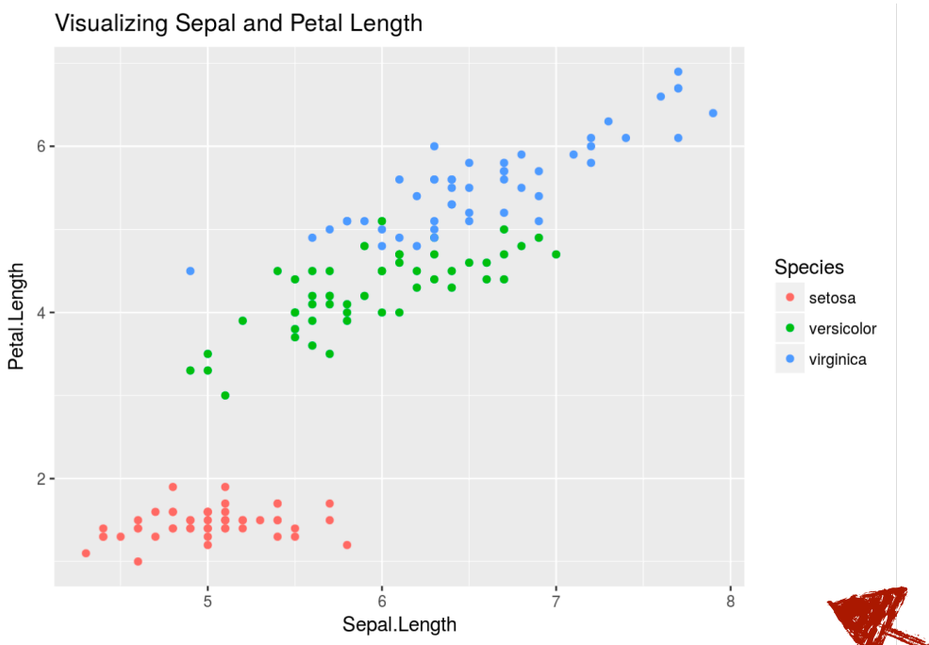
Theme Breakdown

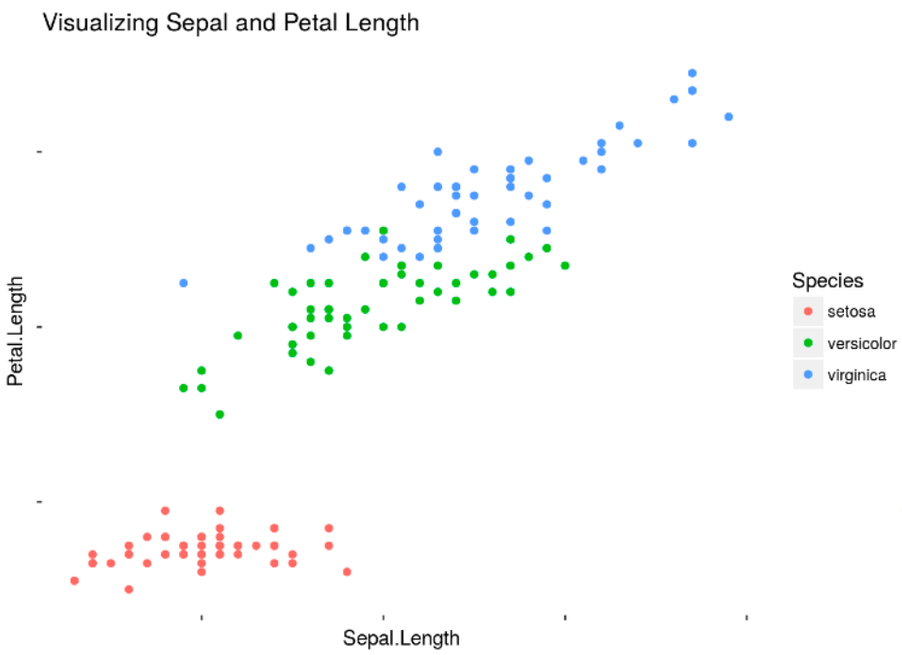
… what goes into a ggplot2 theme …



Changing a Theme

… creating custom theme …

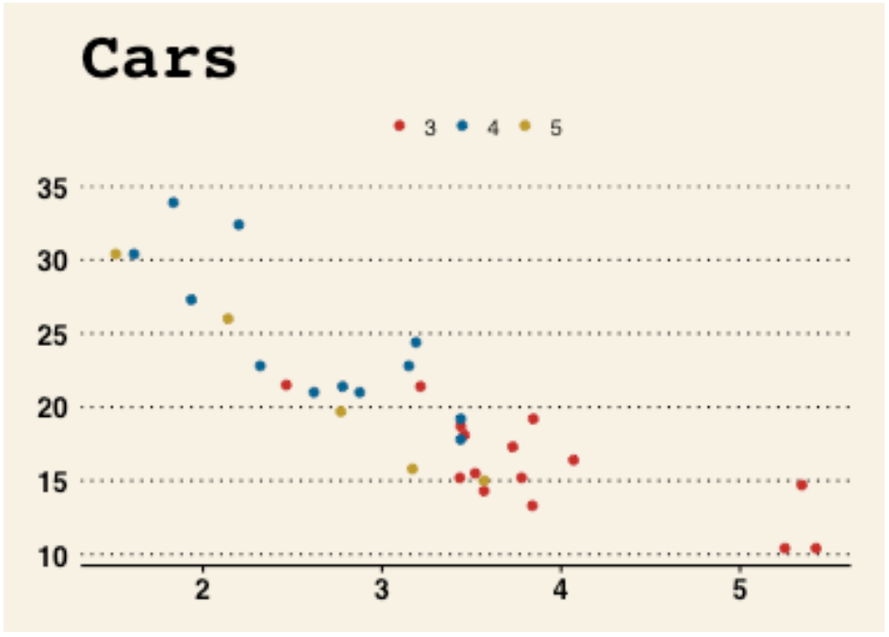




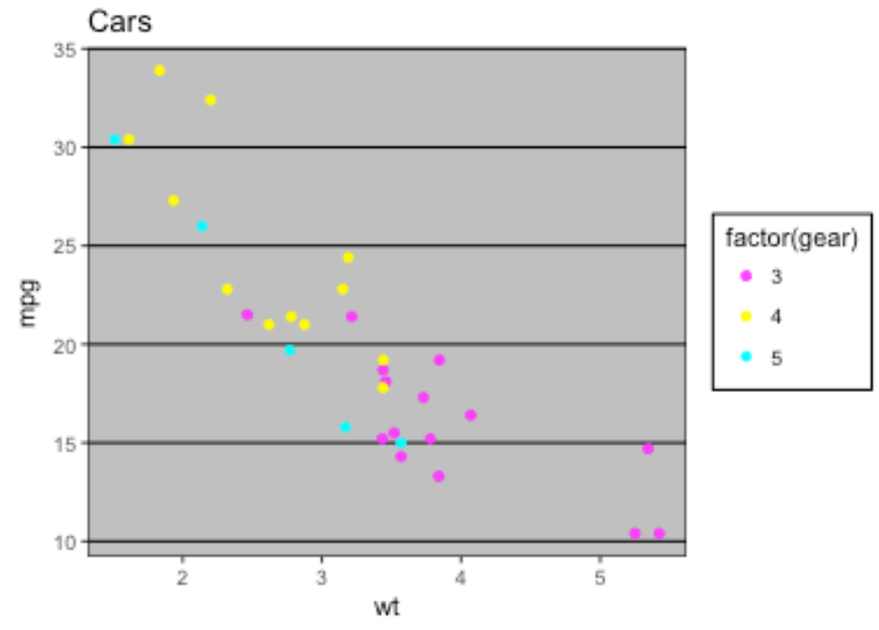

```
# Construct initial graph
g = ggplot(data = iris) +
  aes(x = Sepal.Length,
          y = Petal.Length,
          color = Species) +
  geom_point() +
  labs(
  title = "Visualizing Sepal and Petal
Length"
   )

# Display Graph
g

# Modify Graph
g +
  theme(
  panel.background = element_blank(),
  axis.text = element_blank()
)
```

ggthemes

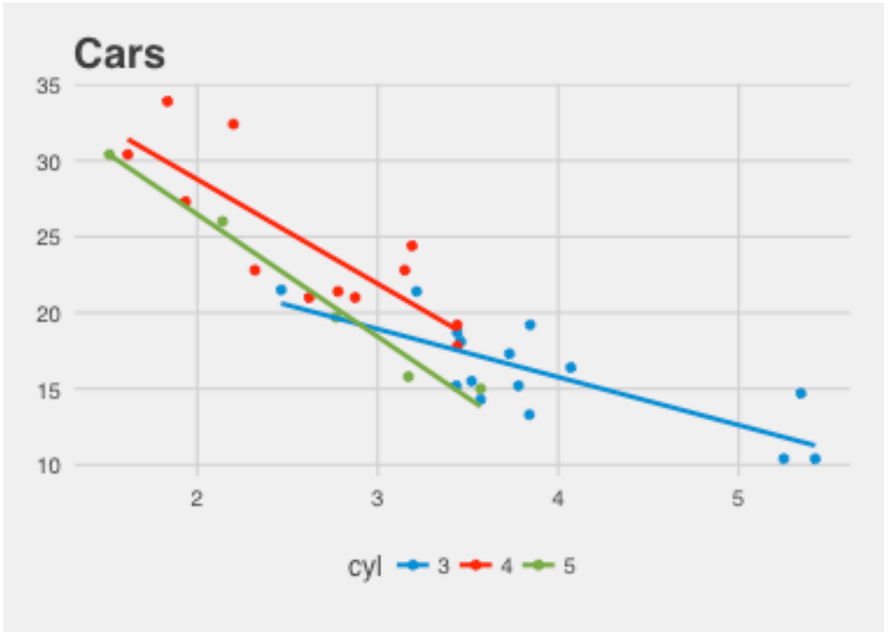
https://github.com/jrnold/ggthemes



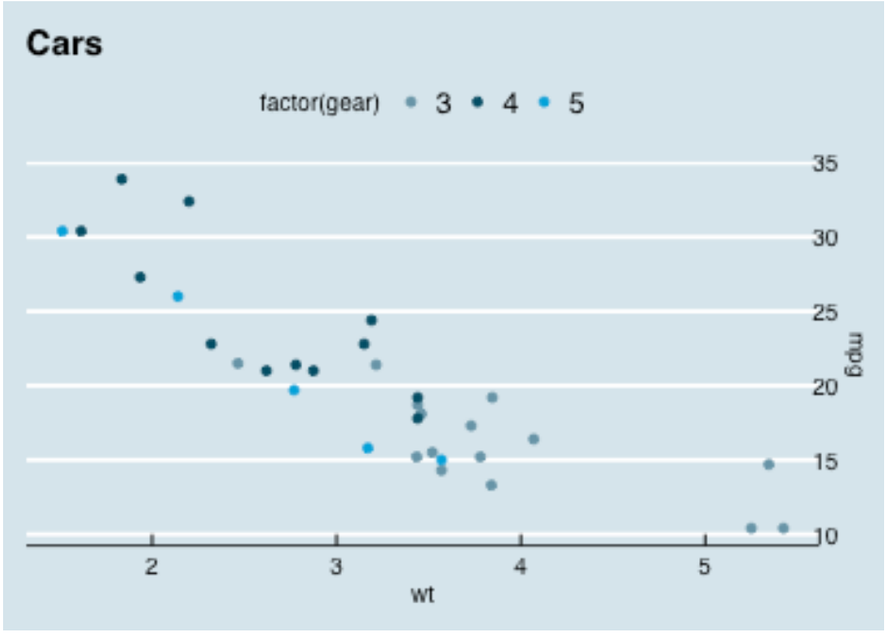
wsj



excel



fivethirtyeight

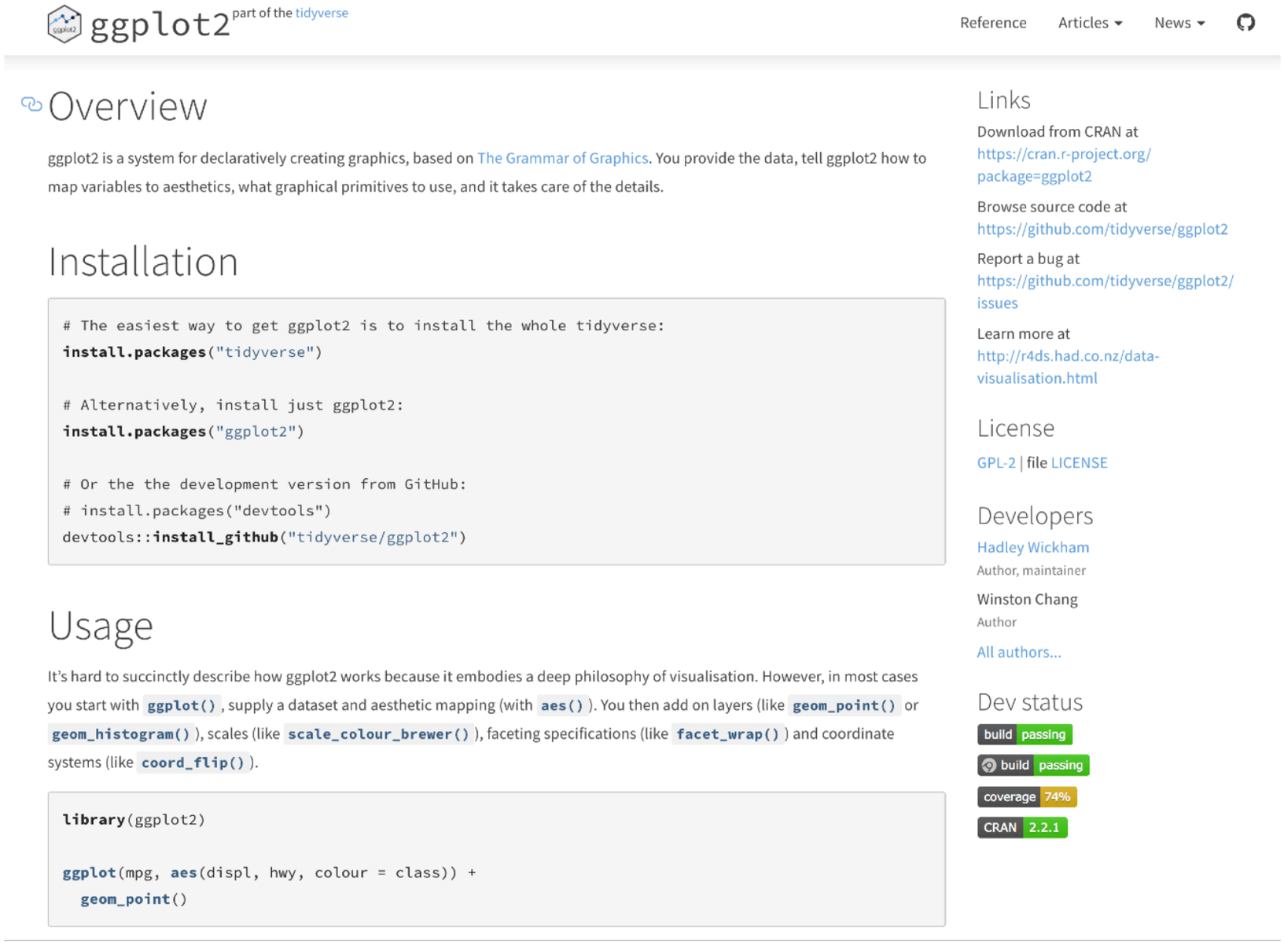


economist

Resources

ggplot2 website

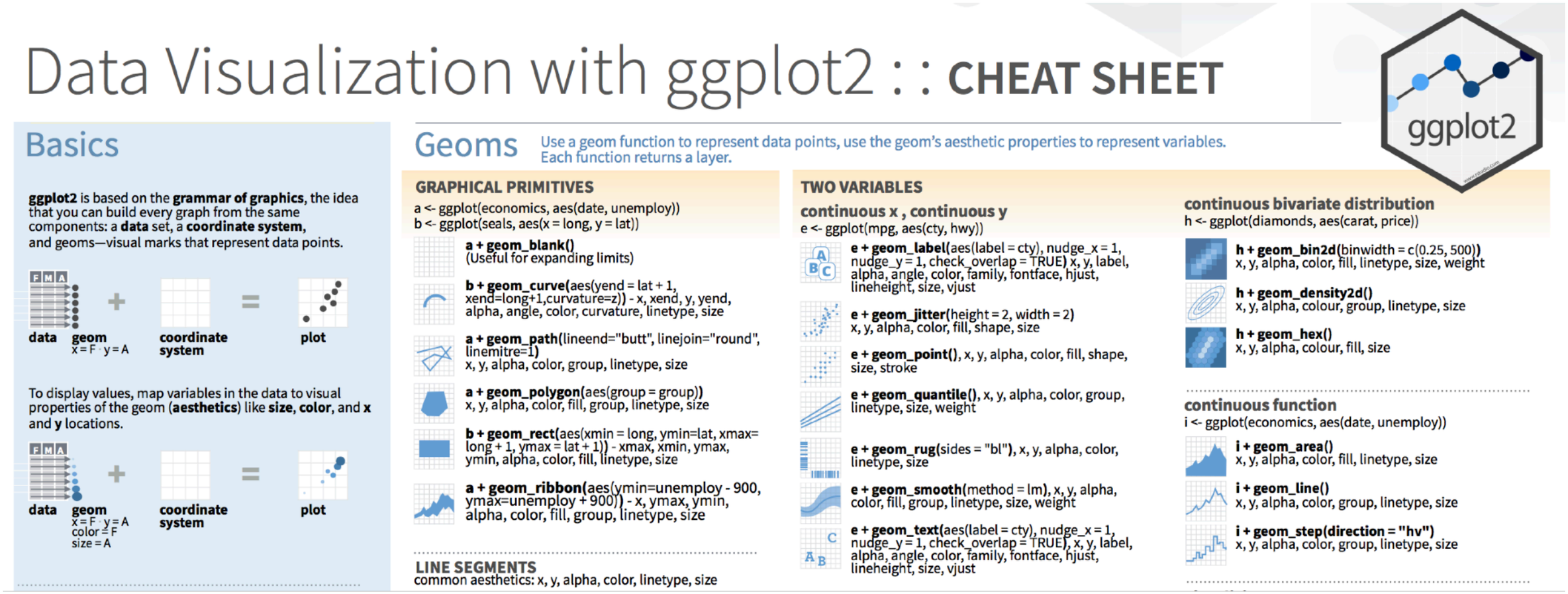
… online documentation with graphs …



ggplot2.tidyverse.org

Cheatsheet for Vis

… need a quick hint at how to make a graph???



Data Visualization with ggplot2 : : **CHEAT SHEET**

ggplot2

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and geoms—visual marks that represent data points.

data **geom** x = F · y = A + **coordinate system** = **plot**

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

data **geom** x = F · y = A color = F size = A + **coordinate system** = **plot**

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank()
(Useful for expanding limits)

b + geom_curve(aes(yend = lat + 1, xend=long+1,curvature=z)**)** - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend="butt", linejoin="round", linemitre=1**)**
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(group = group)**)**
x, y, alpha, color, fill, group, linetype, size

b + geom_rect(aes(xmin = long, ymin=lat, xmax= long + 1, ymax = lat + 1)**)** - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin=unemploy - 900, ymax=unemploy + 900)**)** - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS
common aesthetics: x, y, alpha, color, linetype, size

TWO VARIABLES

continuous x , continuous y
e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)** x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter(height = 2, width = 2**)**
x, y, alpha, color, fill, shape, size

e + geom_point(), x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile(), x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)**, x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)**, x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE**)**, x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

continuous bivariate distribution
h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500)**)**
x, y, alpha, color, fill, linetype, size, weight

h + geom_density2d()
x, y, alpha, colour, group, linetype, size

h + geom_hex()
x, y, alpha, colour, fill, size

continuous function
i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

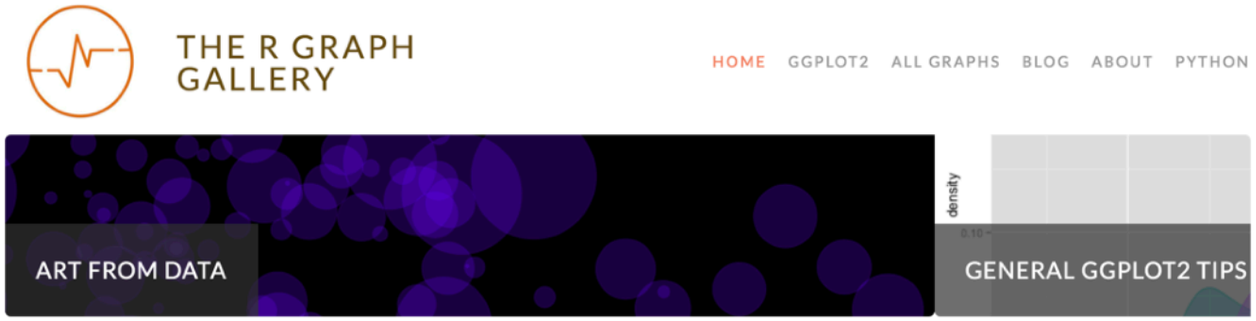
i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size

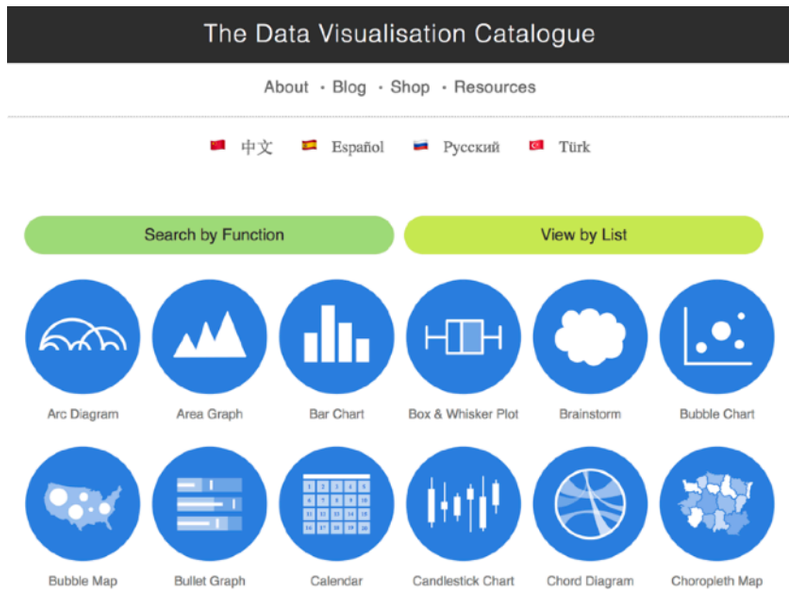
[Source]

Galleries for Vis

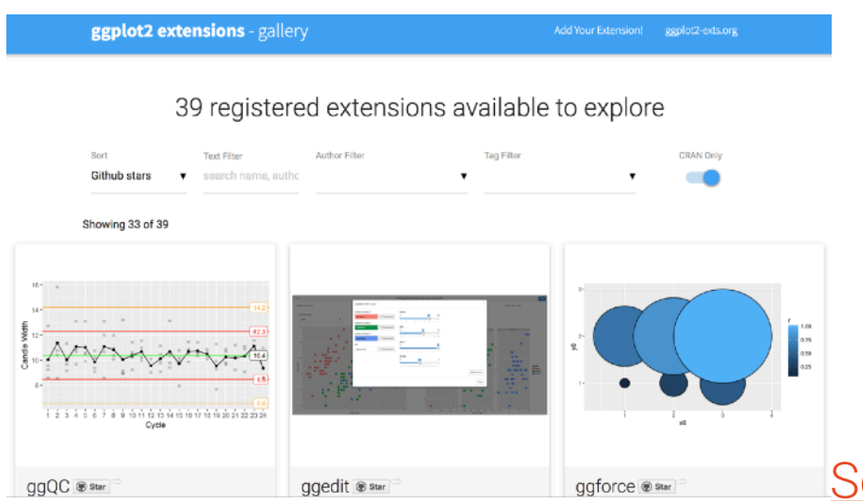
… need a quick hint at how to make a graph???



Source



Source



Source

Recap

- **Exploratory Data Analysis**
  - Search for patterns within the data *before* modeling
  - Quantitative vs. Visual
- **Graphing in R**
  - Three different systems: Base R, lattice, **ggplot2**
- **ggplot2**
  - Grammar of graphics implementation in *R*
  - Powerful system for quickly constructing EDA graphics on data