Properties of Data

Digging into Data

University of Maryland

February 11, 2013

COLLEGE OF
INFORMATION
STUDIES

ggplot2 material adapted from Karthik Ram

Roadmap

- Getting and cleaning data
  - Unavoidable step
  - Example of how I do it
- Goal
  - Not to teach you how
  - What end results you need to tell stories from data
  - Telling those stories with pictures
  - Same thing necessary for making predictions and clustering
  - Homework 1
- ggplot2
- CaBi

Outline

(Confusing) Terminology

- A dataset has different components
- Input: what you always know
  - Sometimes called independent variable
  - Sometimes called regressor
  - Sometimes called feature
- Output: what you're trying to learn
  - Sometimes called independent variable
  - Sometimes called the regressand
  - Sometimes called the response variable
  - Sometimes called the "label"

(Confusing) Terminology

- A dataset has different components
- Input: what you always know
  - Sometimes called independent variable
  - Sometimes called regressor
  - Sometimes called feature
- Output: what you're trying to learn
  - Sometimes called independent variable
  - Sometimes called the regressand
  - Sometimes called the response variable
  - Sometimes called the "label"
  - Does not exist for **unsupervised** learning

Terminology

- But not all data are usable
- Most data also have an **identifier**
- Could also be metadata
  - When data was collected
  - Who collected it
  - How much it cost
- Often important to exclude such data from your algorithms

Terminology

Discrete Data

- Also called categoric
- Bins that you group data into
- There is no "in between"
- You can ask most frequent value

Continuous Data

- Also called numeric
- Numeric values that represent data
- There is an "in between"
- You can take the average
- It makes sense to ask questions like what if this were 10% more X

Outline

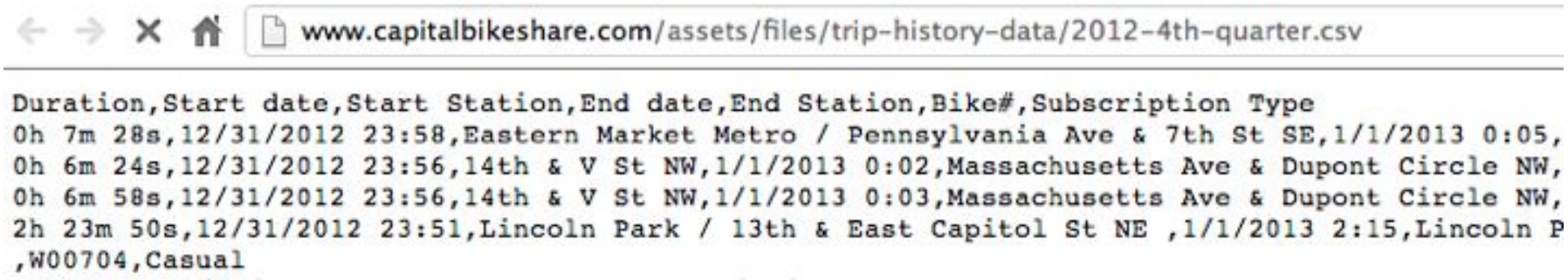
Capital Bikeshare

- Until this year, largest bikeshare system in US
- Publicly share data
- Important problems:
  - Where should new stations be?
  - Rebalancing
  - Pricing
  - Coordinating with other transit

Downloading CaBi Data

CSV File

http://www.capitalbikeshare.com/trip-history-data



What story do you want to tell?

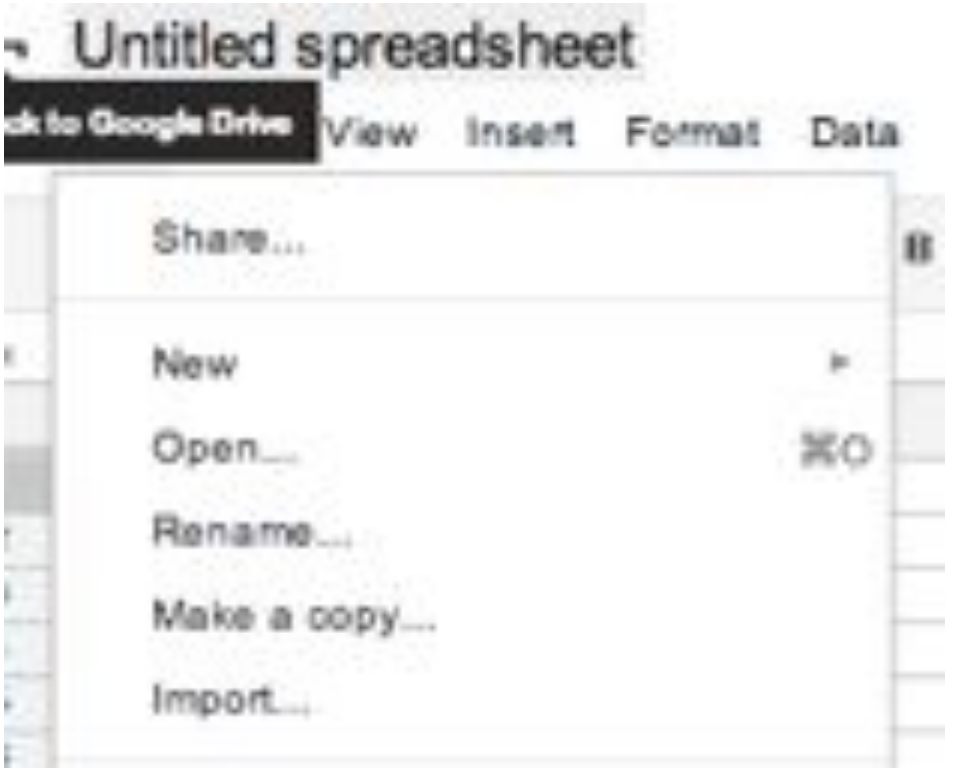
- What data are there?
- What information do you want?
- How to get from point A to point B?

What story do you want to tell?

- What data are there?
- What information do you want?
- How to get from point A to point B?
  - More art than science
  - No right answers

Adding it to Google Docs

Import into Google Spreadsheet



Adding it to Google Docs

Loads nicely into columns

Preview

	A	B	C	D	E
1	Duration	Start date	Start Station	End date	End Statio
2	0h 7m 28s	12/31/201 23:58:00	Eastern Market Metro / Pennsylva Ave & 7th St SE	1/1/2013 0:05:00	14th & St SE
3	0h 6m 24s	12/31/201 23:56:00	14th & V St NW	1/1/2013 0:02:00	Mass Ave & Dupo Circle
4	0h 6m 58s	12/31/201 23:56:00	14th & V St NW	1/1/2013 0:03:00	Mass Ave & Dupo Circle

Adding it to Google Docs

It would be nice to have more

- Real world locations
- Elevation
- CaBi has some of this information
- Google (Maps) knows the rest . . .

Adding it to Google Docs

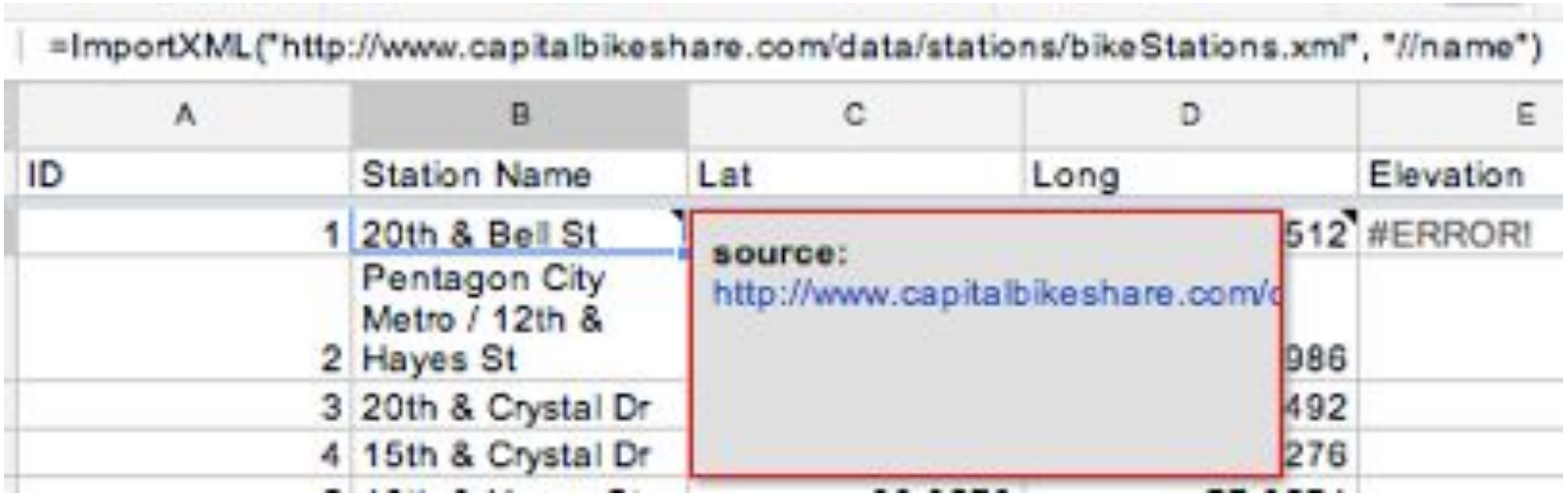
http://www.capitalbikeshare.com/data/stations/bikeStations.xml

Adding it to Google Docs

Creating a new sheet just for stations

Adding it to Google Docs

Load columns from the xml file



We now have columns for lat, long for every station

Adding it to Google Docs

Now we can attach a location to each row in the original sheet

Adding it to Google Docs

Now we've added neat new columns to the spreadsheet; time to download

Loading a dataset

```
rides <- read.csv("data/cabi-rides.ext.csv")
```

- Creates a "data frame"
- This is the basic unit of R data (Rattle creates these automatically for you)
- Very easy to add columns
- Use the $ to access columns

Outline

Summarizing Data

Getting Output Directly

- "Explore" tab
- Click: "summary"

```
   duration                                                    startStation
Min.   : 0.0000   Massachusetts Ave & Dupont Circle NW              : 116
1st Qu.: 0.1000   15th & P St NW                                    :  97
Median : 0.1667   Columbus Circle / Union Station                   :  94
Mean   : 0.2418   Thomas Circle                                    :  79
3rd Qu.: 0.2667   Eastern Market Metro / Pennsylvania Ave & 7th St SE:  74
Max.   :13.5667   17th & Corcoran St NW                            :  70
NA's   : 2.0000   (Other)                                          :3629
```

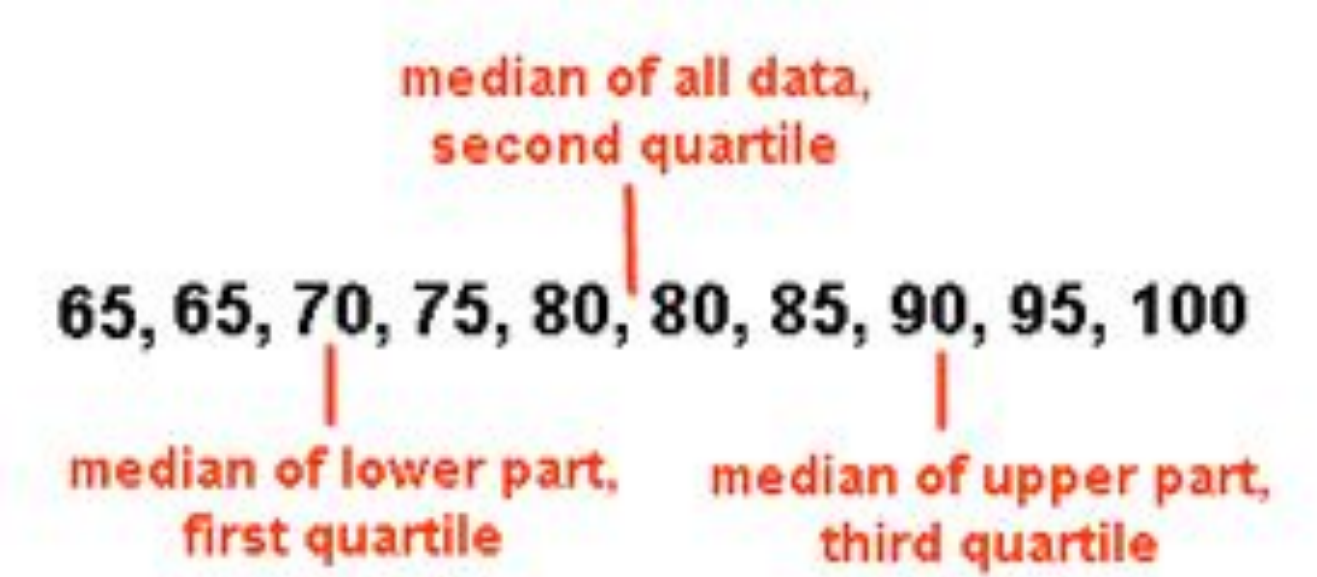
Summarizing Data

Getting Output Directly

- "Explore" tab
- Type: "summary"

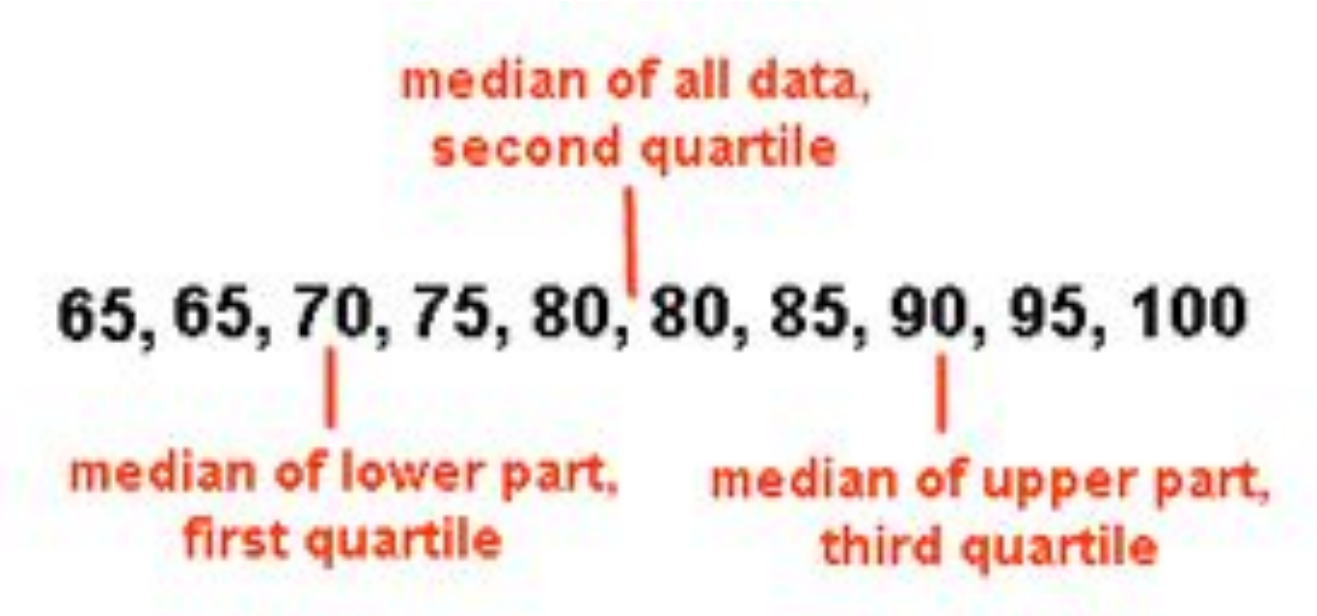
```
                                endStation      distance         startHour
Massachusetts Ave & Dupont Circle NW: 148   Min.   :    0.0   Min.   : 0.1333
15th & P St NW                      : 103   1st Qu.:  921.5   1st Qu.:10.5500
Thomas Circle                       :  94   Median : 1515.5   Median :15.1500
17th & Corcoran St NW               :  86   Mean   : 1785.3   Mean   :14.6237
Columbus Circle / Union Station     :  82   3rd Qu.: 2402.2   3rd Qu.:18.3500
North Capitol St & F St NW          :  74   Max.   :13166.5   Max.   :23.9667
(Other)                             :3572                     NA's   : 1.0000
```

Descriptive Statistics: Quartiles



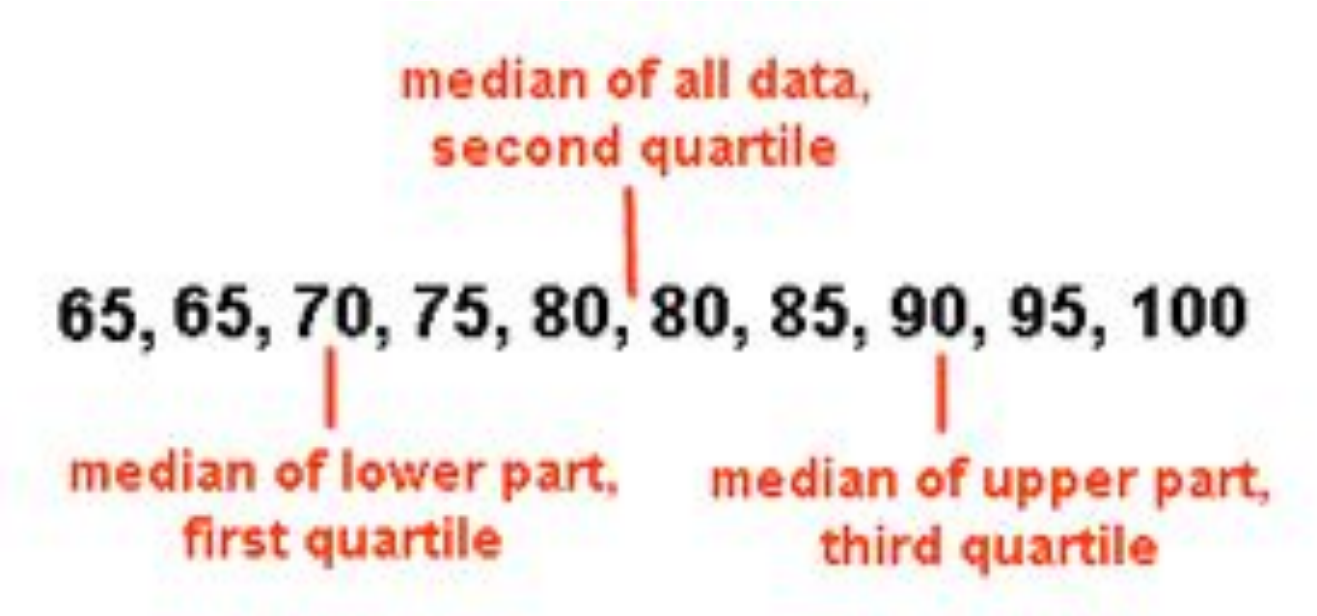
- Order your data
- Find the middle data point - this is your median
  - If even number of data points, average points in the middle
- Repeat on two halves on either side of median - these are your first and third quartiles

Descriptive Statistics



- min - smallest data point
- max - largest data point
- mean - sum of all data divided by number of data points

Descriptive Statistics



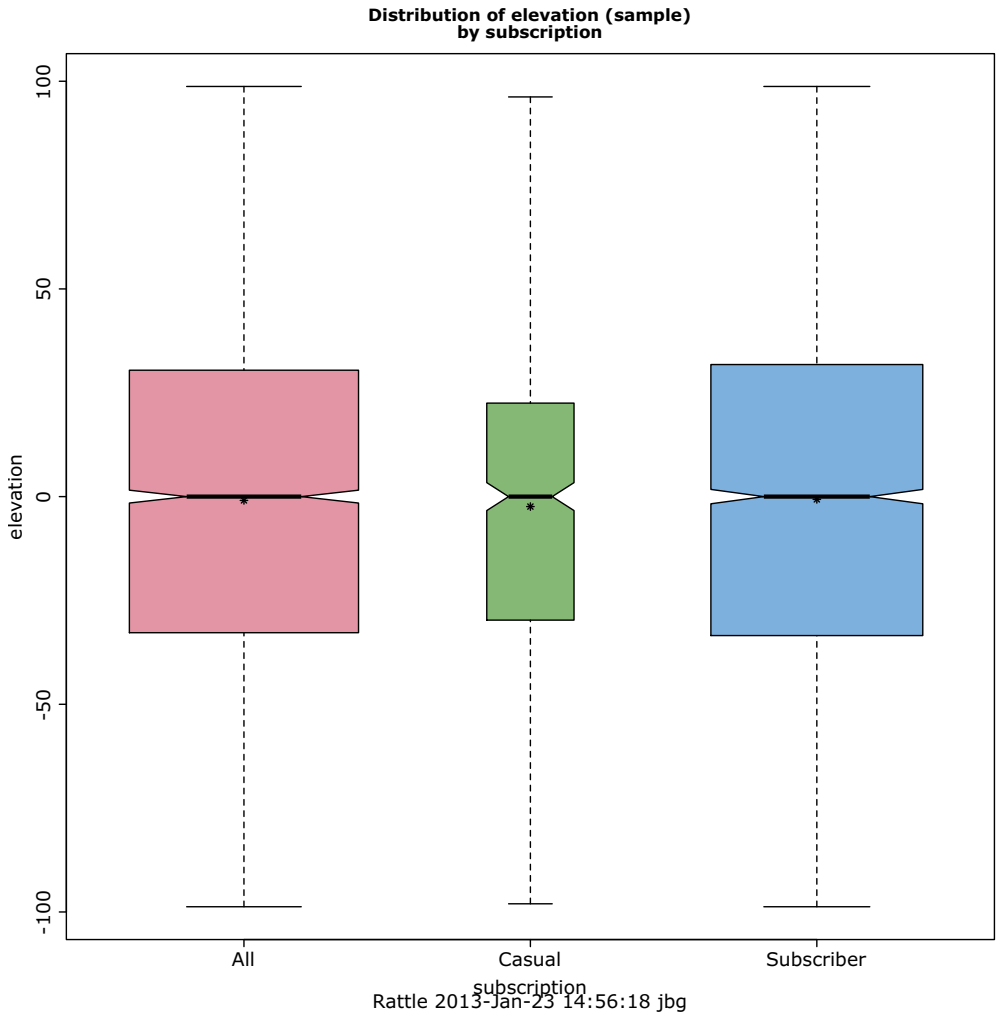
- min - smallest data point
- max - largest data point
- mean - sum of all data divided by number of data points

$$\mu = \sum_i x_i / N \tag{1}$$

What to look for . . .

- Are the min / max reasonable?
- Is there a lot of missing data (NA)?
- Do the most frequent levels for categorical data make sense?

Box Plots



- Show median, mean, Q1, Q2, max and min
- Show if distributions are skewed
- Easier to see than reading off numbers
- Introduced by Tukey
- Under "Explore", "Distributions"

Outline

Some housekeeping

Install some packages (make sure you also have recent copies of reshape2 and plyr)

```
install.packages("ggplot2", dependencies = TRUE)
```

Base graphics

- Ugly, laborious, and verbose
- There are better ways to describe statistical visualizations.

Why ggplot2?

- Follows a grammar, just like any language.
- It defines basic components that make up a sentence. In this case, the grammar defines components in a plot.
- Grammar of graphics originally coined by Lee Wilkinson

Why ggplot2?

- Supports a continuum of expertise.
- Get started right away but with practice you can effortless build complex, publication quality figures.
- Common pitfal:
  - Never use `qplot` - short for quick plot.
  - You'll end up unlearning and relearning a good bit.

Some terminology

- **ggplot** - The main function where you specify the dataset and variables to plot
- **geoms** - geometric objects
  - geom_point(), geom_bar(), geom_density(), geom_line(), geom_area()
- **aes** - aesthetics
  - shape, transparency (alpha), color, fill, linetype.
- **scales** Define how your data will be plotted
  - *continuous*, *discrete*, *log*

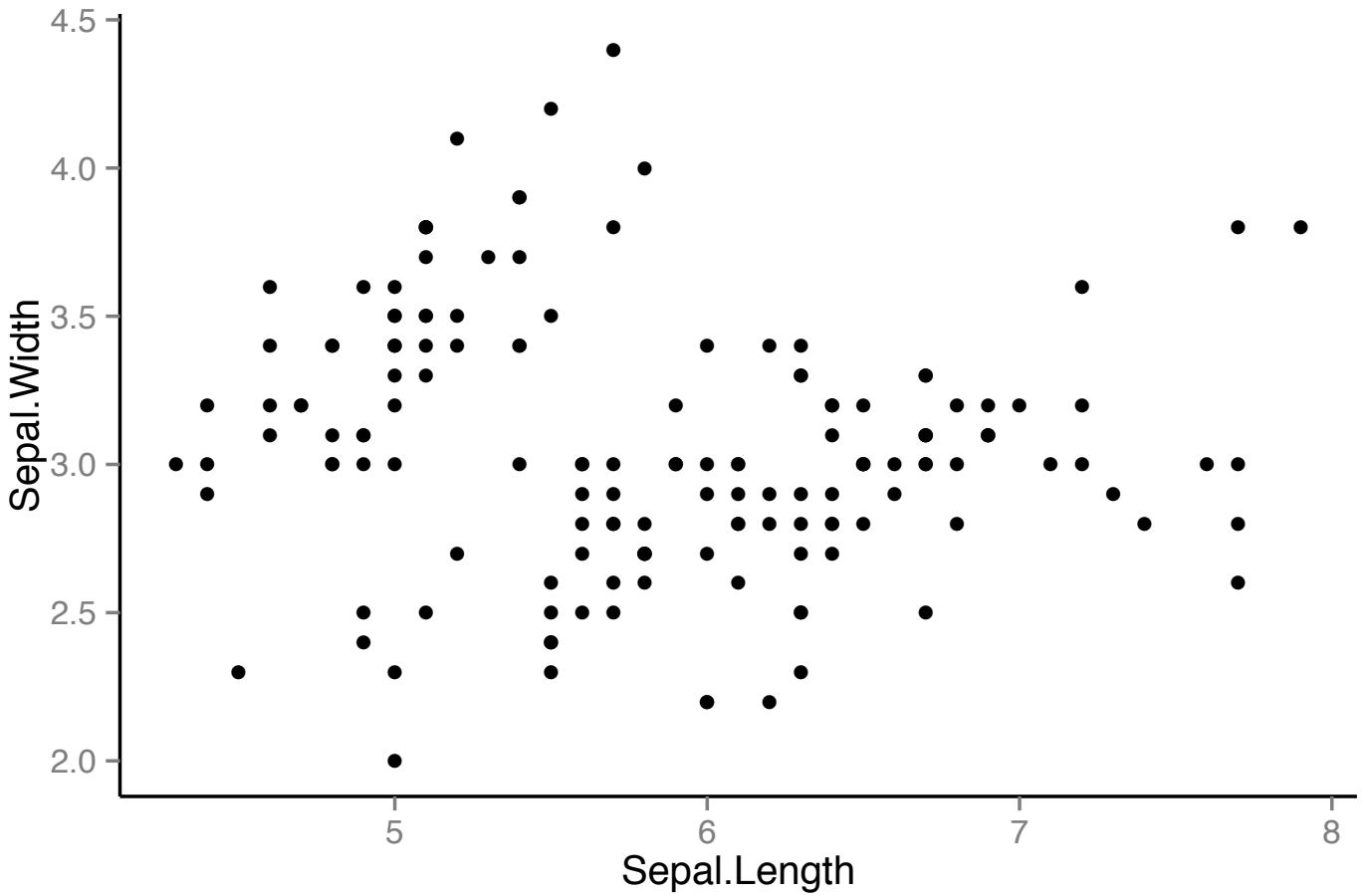
The iris dataset

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Let's try an example

```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +
geom_point()
```



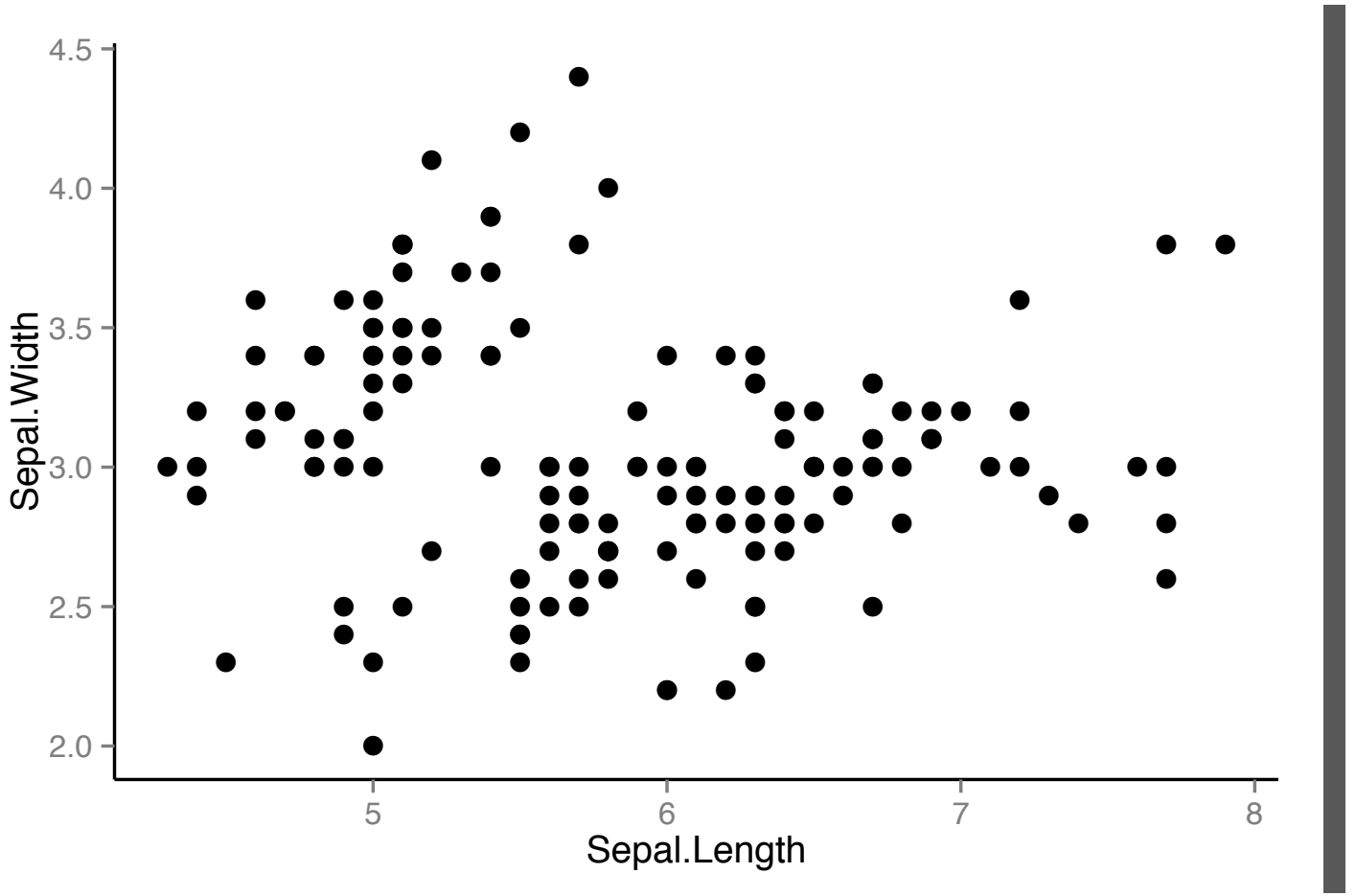
Basic structure

```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))
 + geom_point()
myplot <- ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))
myplot + geom_point()
```

- Specify the data and variables inside the ggplot function.
- Anything else that goes in here becomes a global setting.
- Then add layers of geometric objects, statistical models, and panels.

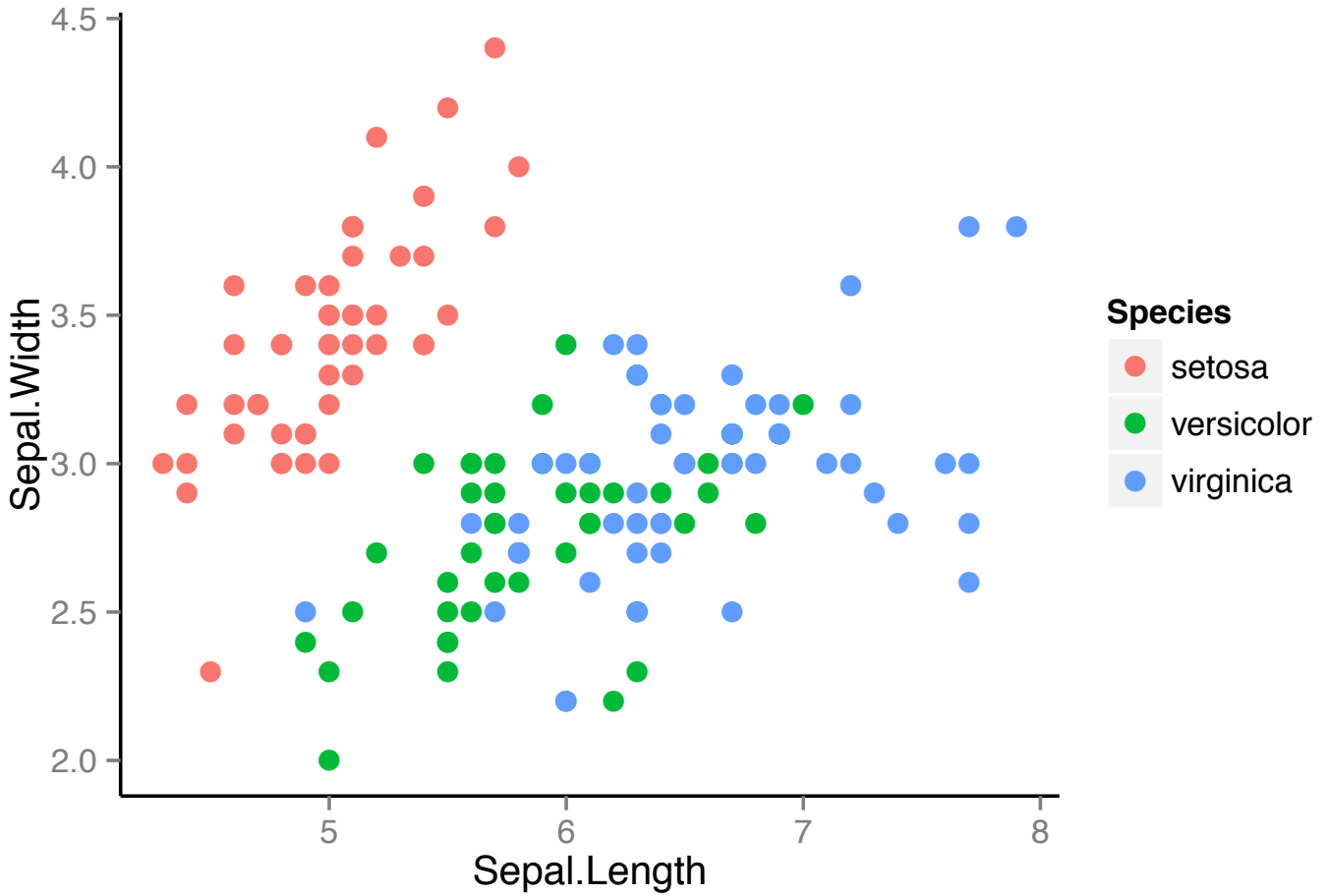
Scatter Plots: Increase the size of points

```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +
geom_point(size = 3)
```



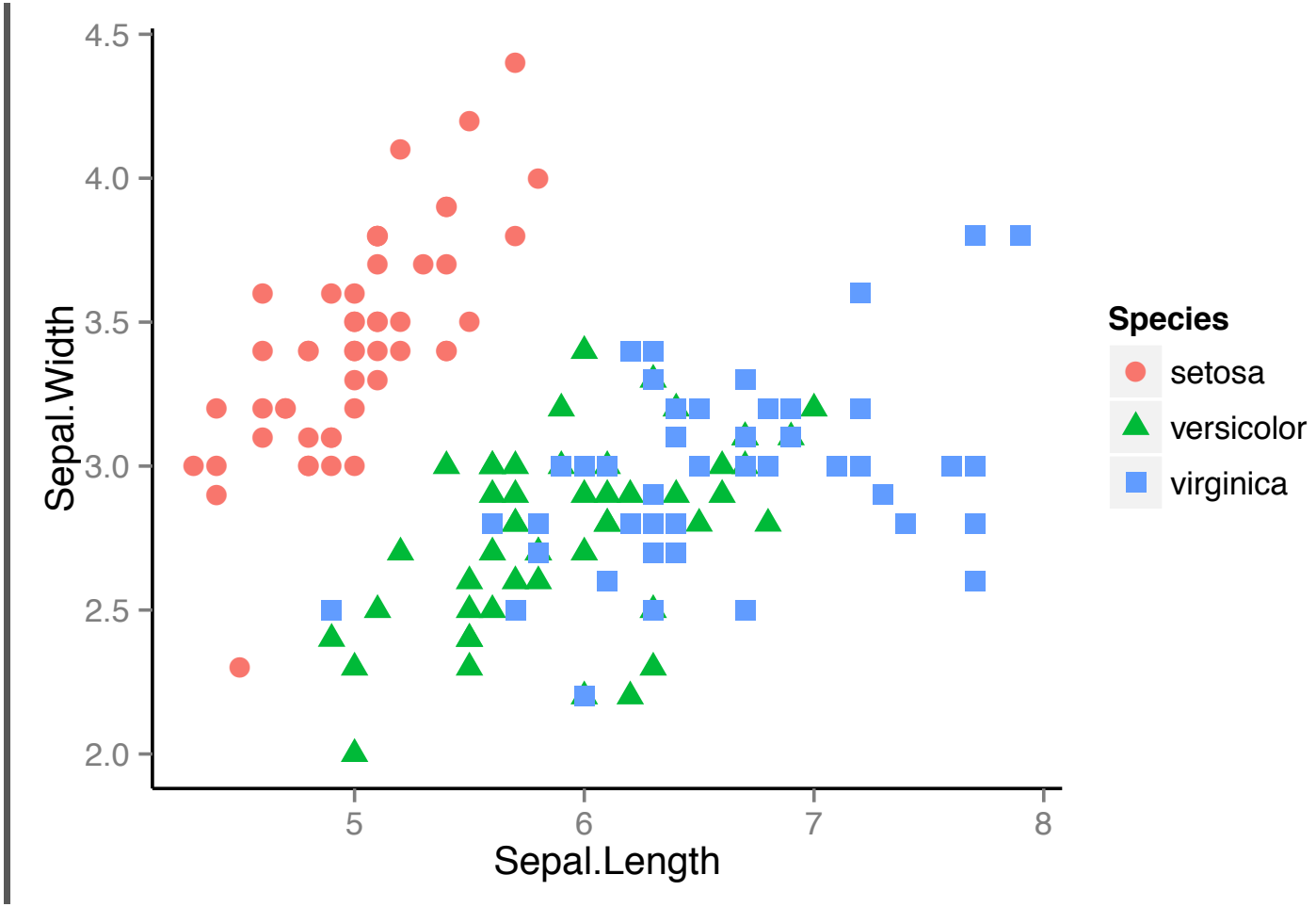
Scatter Plots: Add some color

```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
geom_point(size = 3)
```



Scatter Plots: Differentiate points by shape

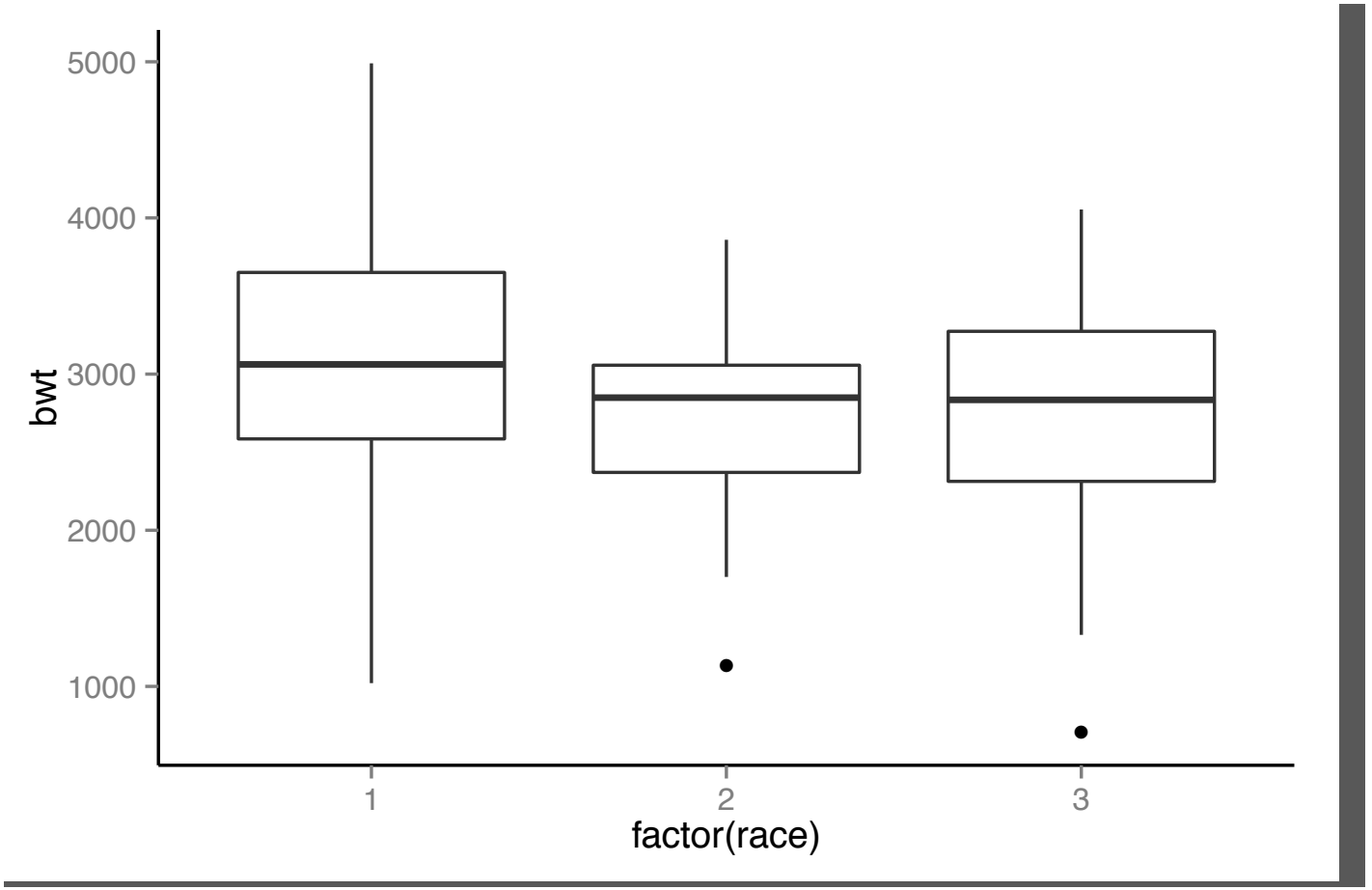
```
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
geom_point(aes(shape = Species), size = 3)
```



Boxplots

See ?geom_boxplot for list of options

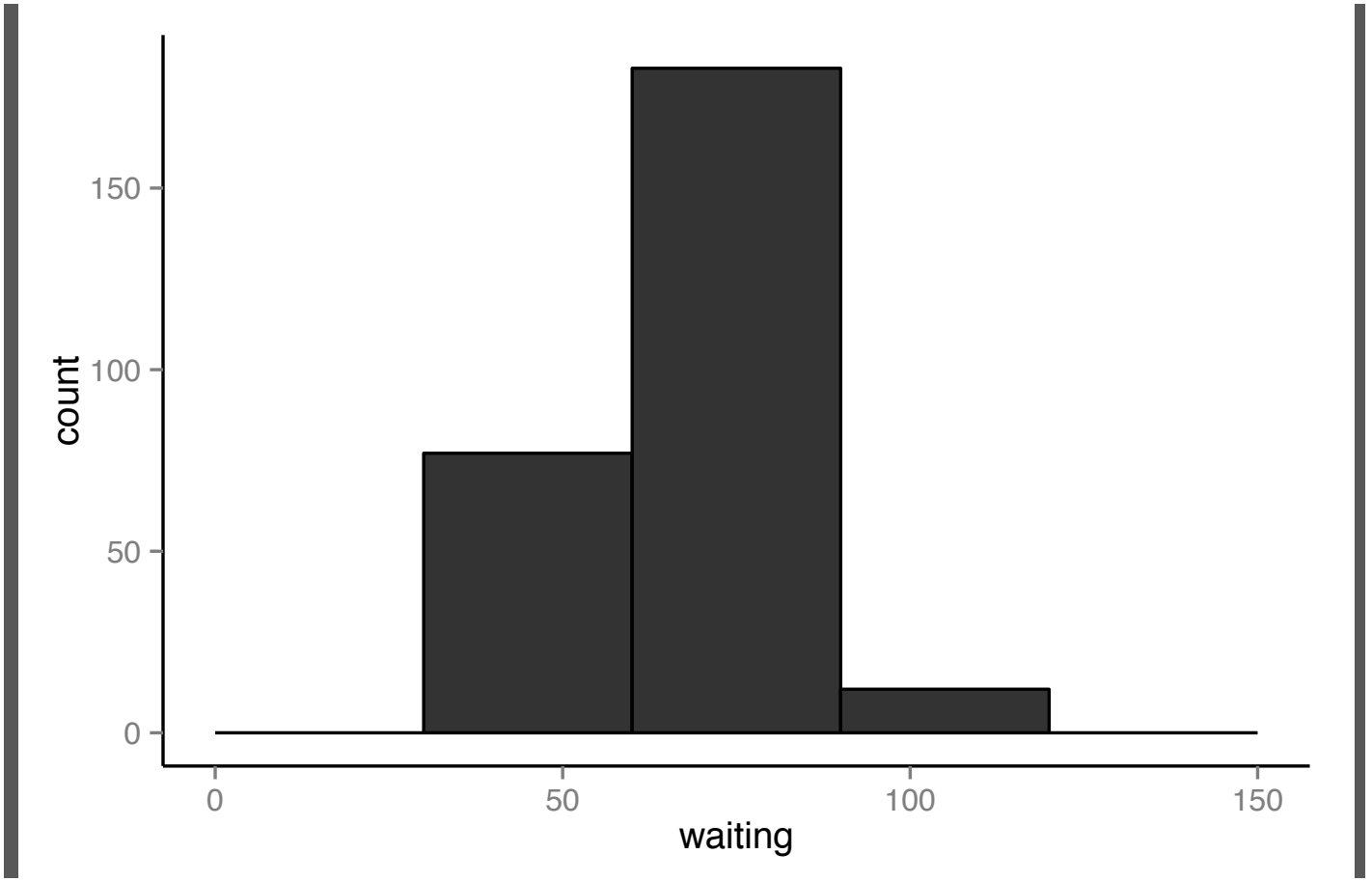
```
library(MASS)
ggplot(birthwt, aes(factor(race), bwt)) + geom_boxplot()
```



Histograms

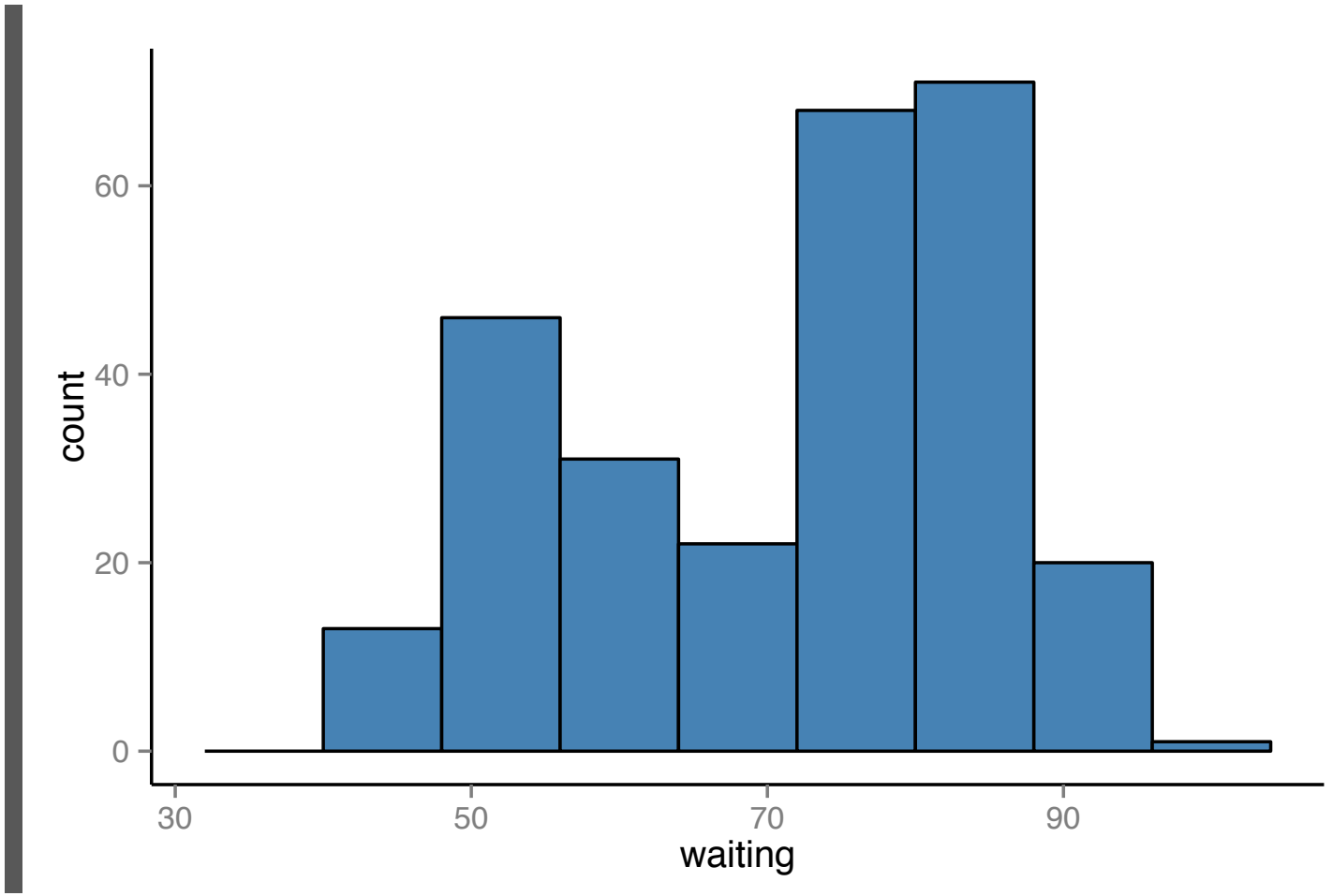
See ?geom_histogram for list of options

```
h <- ggplot(faithful, aes(x = waiting))
h + geom_histogram(binwidth = 30, colour = "black")
```



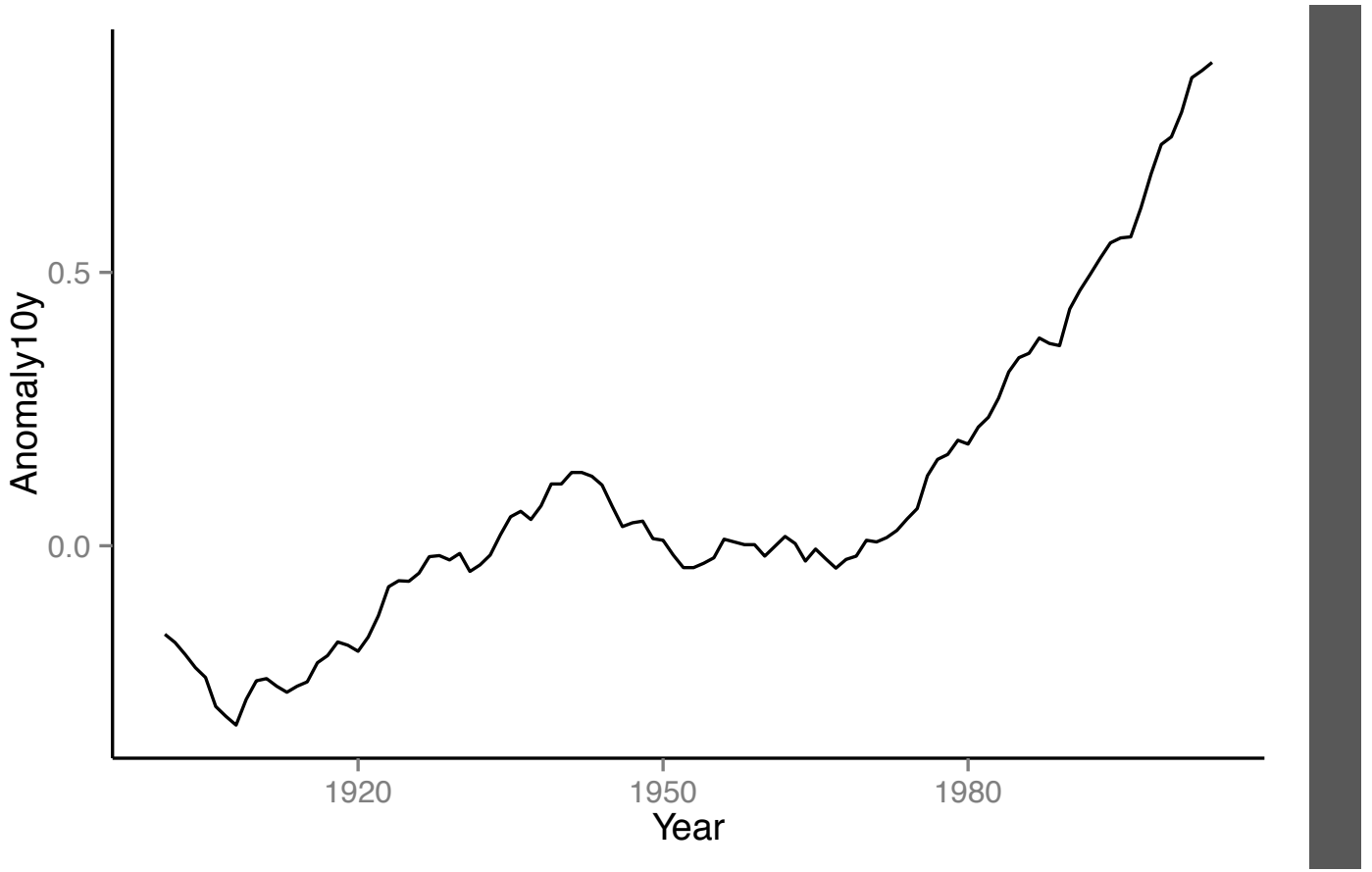
Histograms

```
h <- ggplot(faithful, aes(x = waiting))
h + geom_histogram(binwidth = 8, fill = "steelblue",
colour = "black")
```



Line Plot

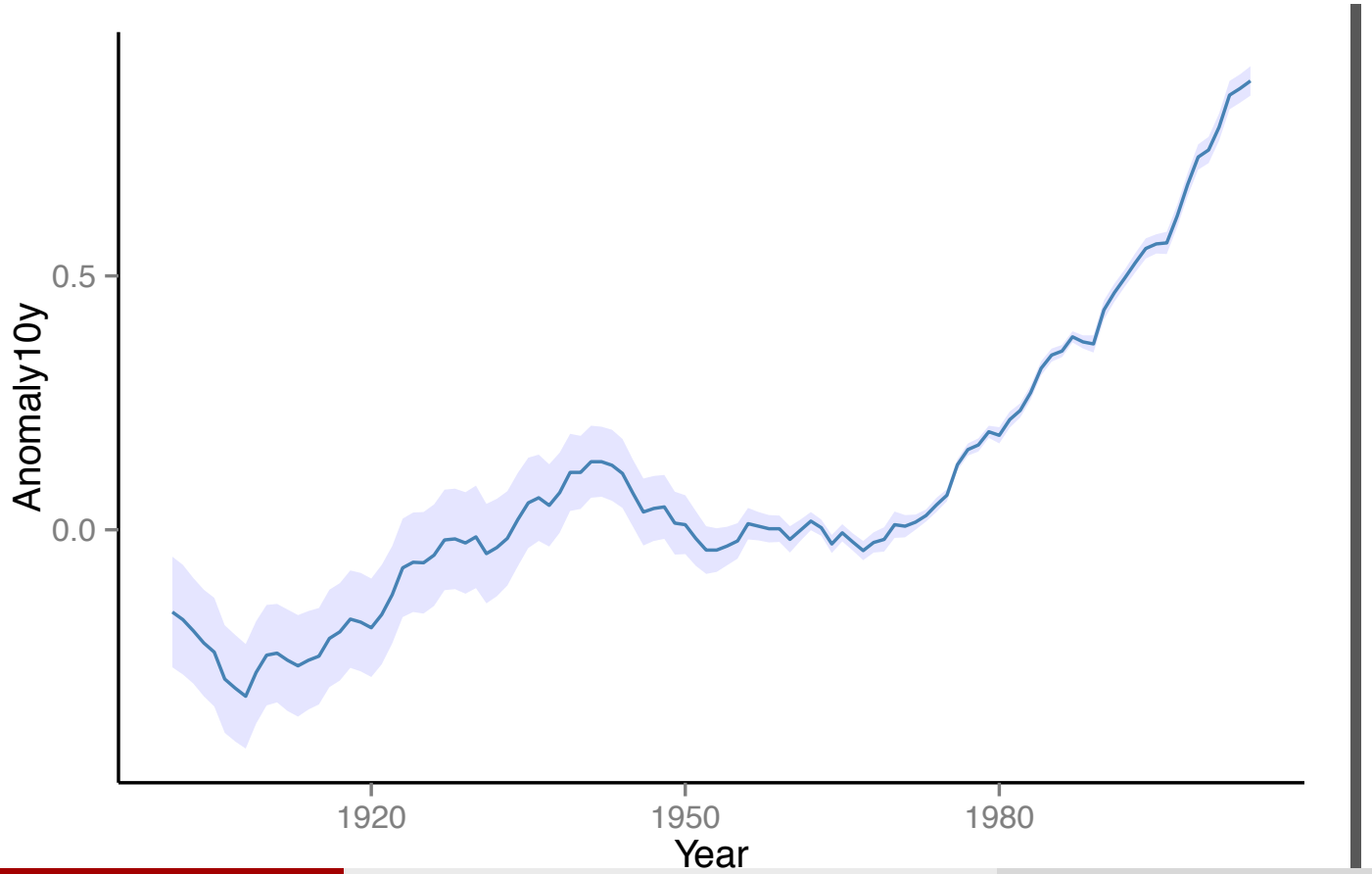
```
climate <- read.csv("climate.csv", header = T)
ggplot(climate, aes(Year, Anomaly10y)) +
geom_line()
```




```
climate <- read.csv(text =
```

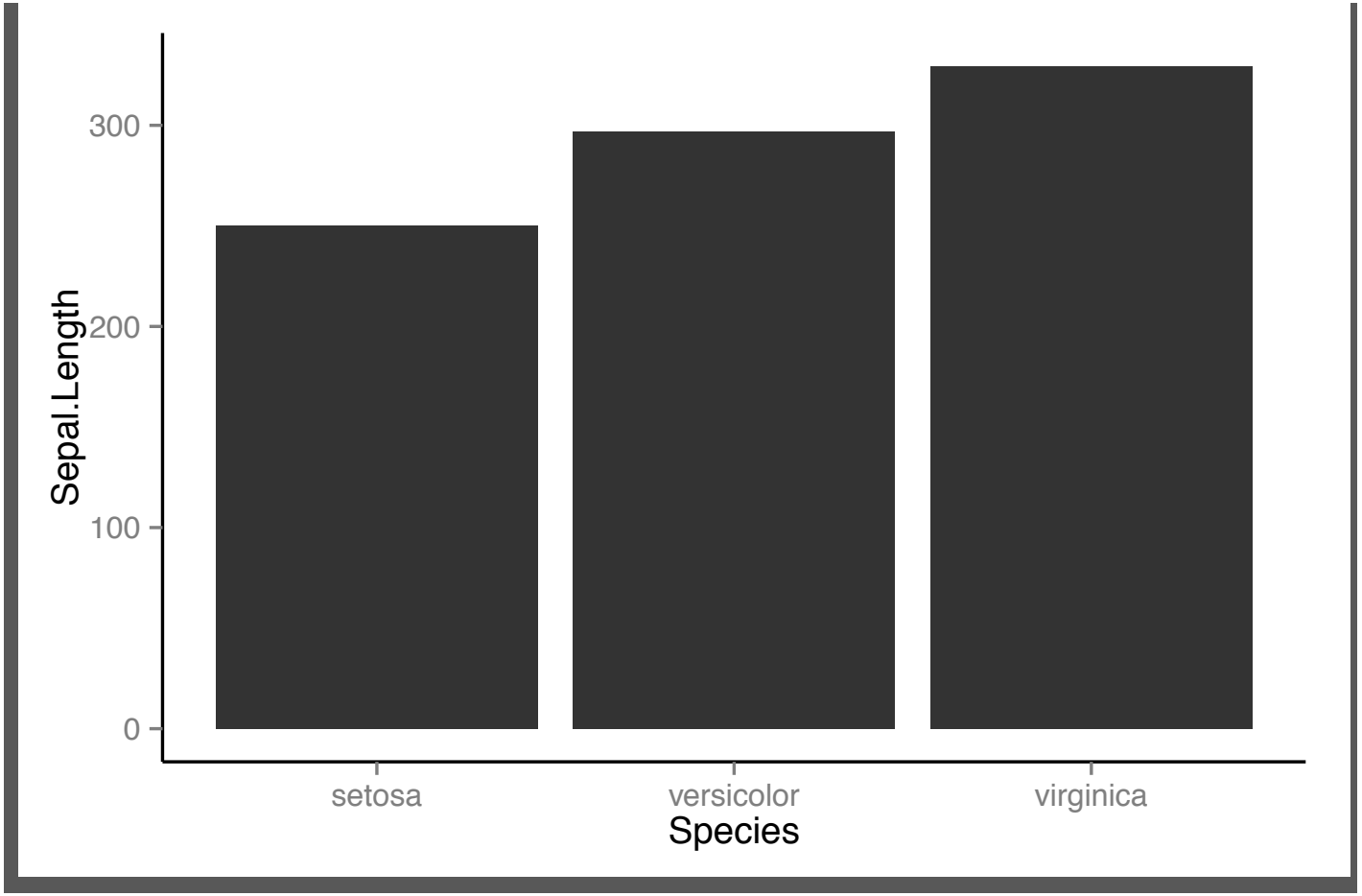
Line Plot: Confidence Regions

```
ggplot(climate, aes(Year, Anomaly10y)) +
geom_ribbon(aes(ymin = Anomaly10y - Unc10y,
ymax = Anomaly10y + Unc10y),
fill = "blue", alpha = .1) +
geom_line(color = "steelblue")
```



Bar Plots

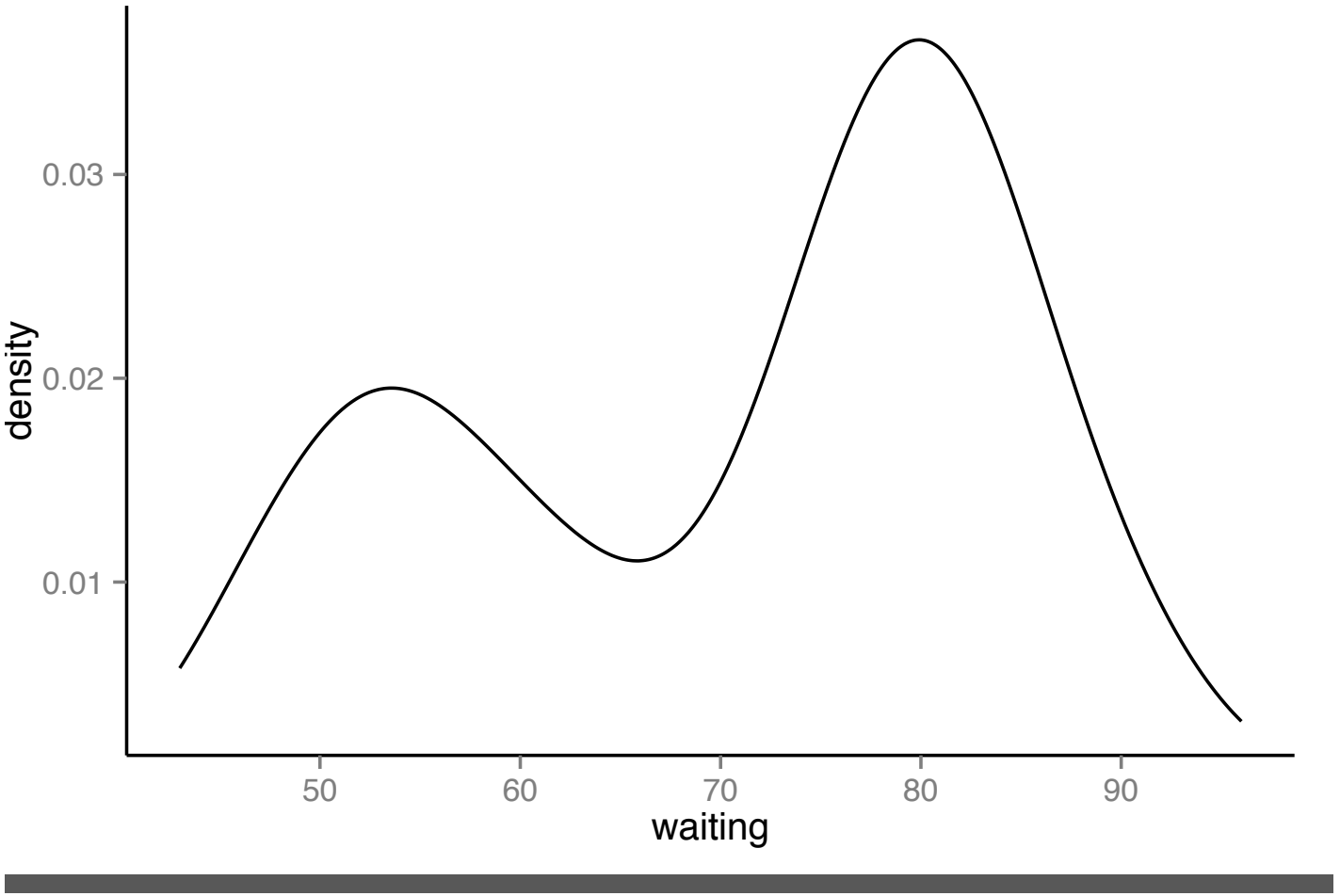
```
ggplot(iris, aes(Species, Sepal.Length)) +
geom_bar(stat = "identity")
```



Density vs. Line Plots

```
ggplot(faithful, aes(waiting)) +
geom_density(fill = "blue", alpha = 0.1)
```

```
ggplot(faithful, aes(waiting)) +
geom_line(stat = "density")
```



Publication Quality Figures

- Raster graphics (bmp, jpeg, png) don't scale well
- Preparing graphics for publication requires vector graphics (pdf, eps)
- Much easier to provide publication-quality images with ggplot2

Saving Plots

- If the plot is on your screen

```
ggsave("~/path/to/figure/filename.png")
```

- If your plot is assigned to an object

```
ggsave(plot1, file = "~/path/to/figure/filename.png")
```

- Specify a size

```
ggsave(file = "/path/to/figure/filename.png", width = 6,
height =4)
```

- or any format (pdf, png, eps, svg, jpg)

```
ggsave(file = "/path/to/figure/filename.eps")
ggsave(file = "/path/to/figure/filename.jpg")
ggsave(file = "/path/to/figure/filename.pdf")
```

Outline

ggplot2 maps

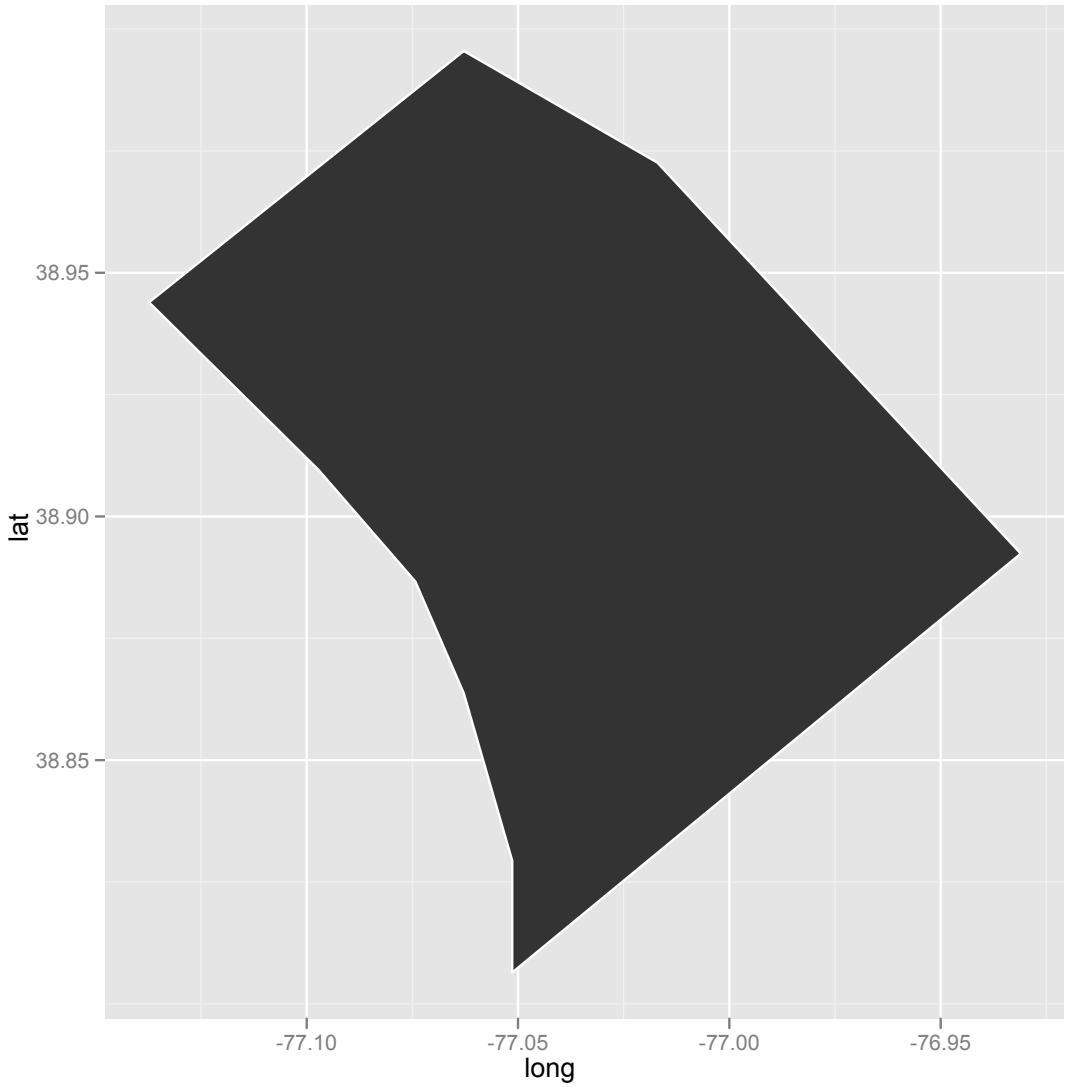
Get an outline of DC

```
all_states <- map_data("state")
states <- subset(all_states, region %in%
                          c( "district of columbia" ) )
```

Draw it

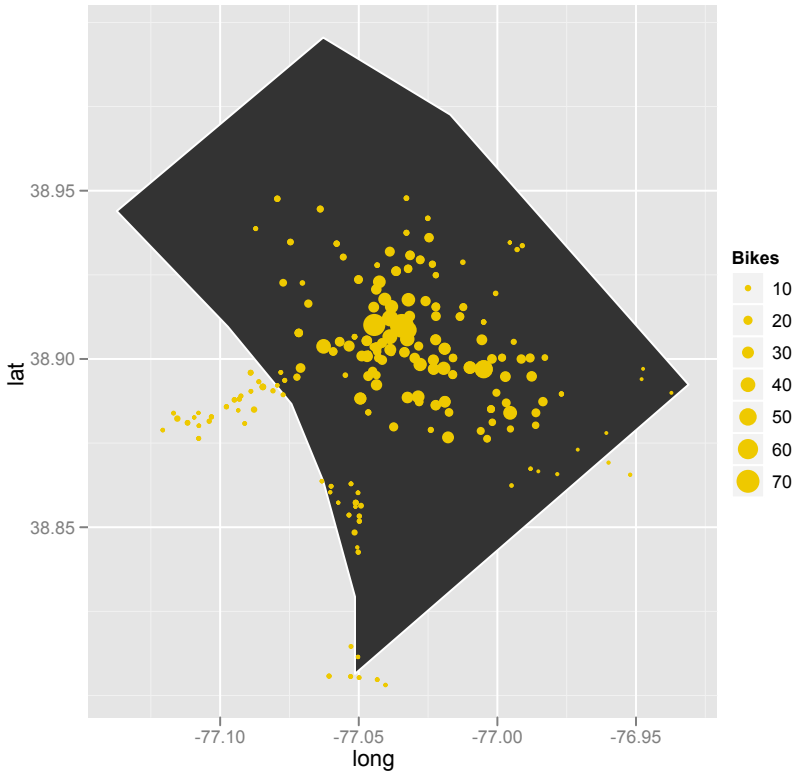
```
p <- ggplot(stations)
p <- p + geom_polygon( data=states, aes(x=long, y=lat))
```

ggplot2 maps



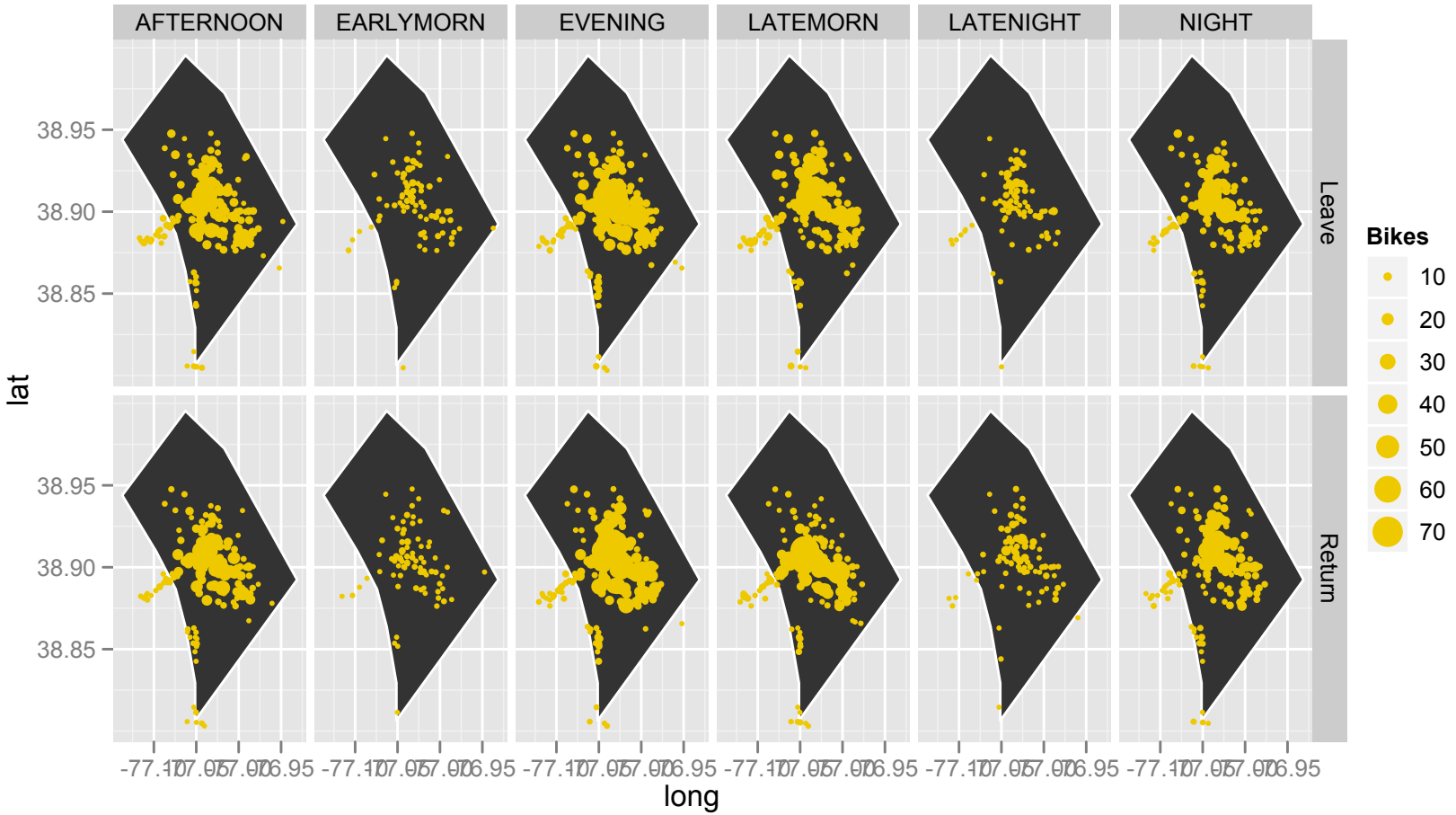
ggplot2 maps

```
p <- p + geom_point( data=stations,
                     aes(x=long, y=lat, size = count),
                          color="gold2") +
    scale_size(name="Bikes")
```



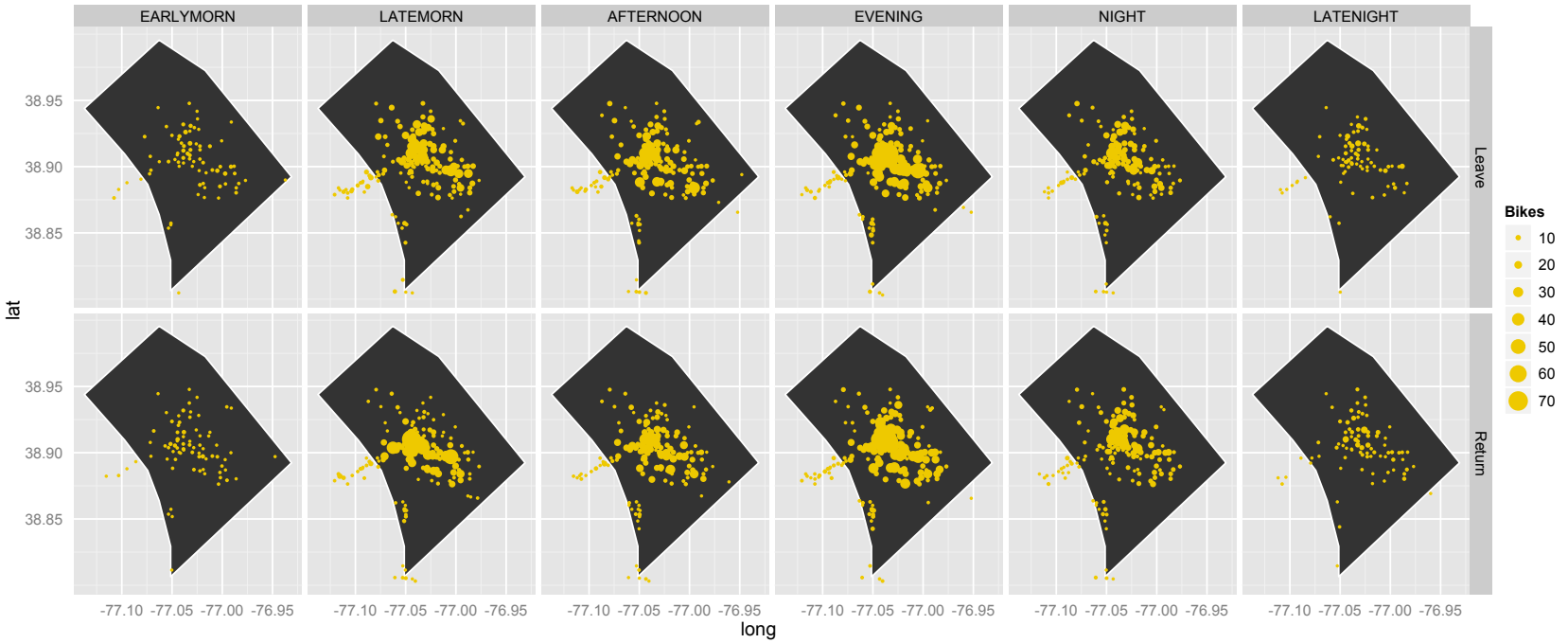
ggplot2 facets

```
p <- p + facet_grid(type ~ time)
```



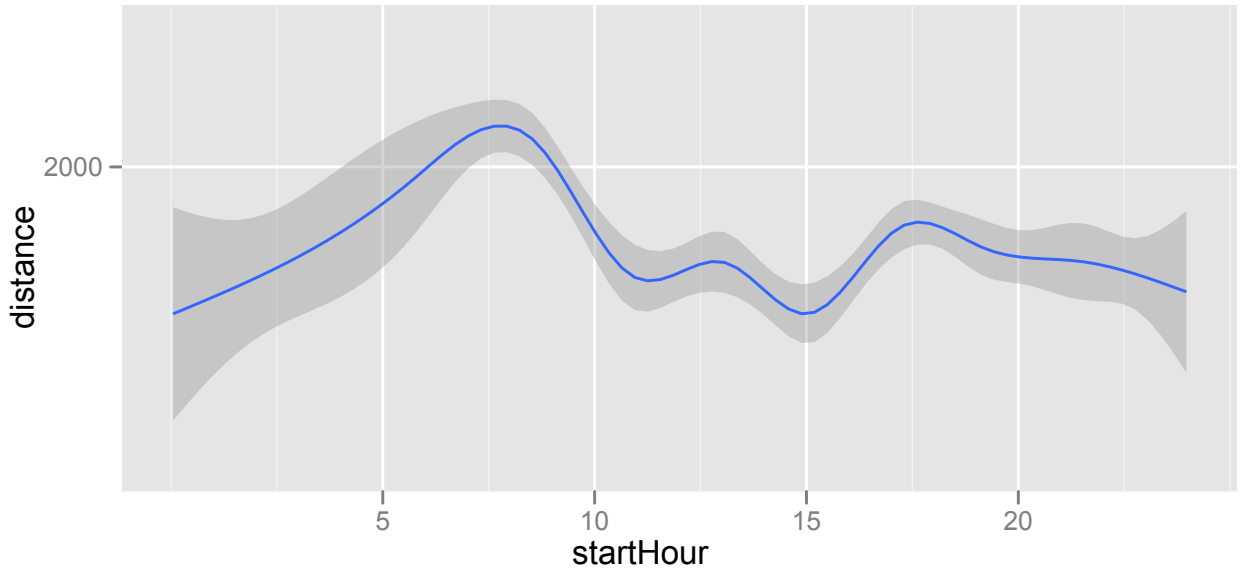
ggplot2 facets (resorted)

```
stations$time <- factor(stations$time, levels =
                c("EARLYMORN","LATEMORN","AFTERNOON",
                  "EVENING", "NIGHT", "LATENIGHT"))
```



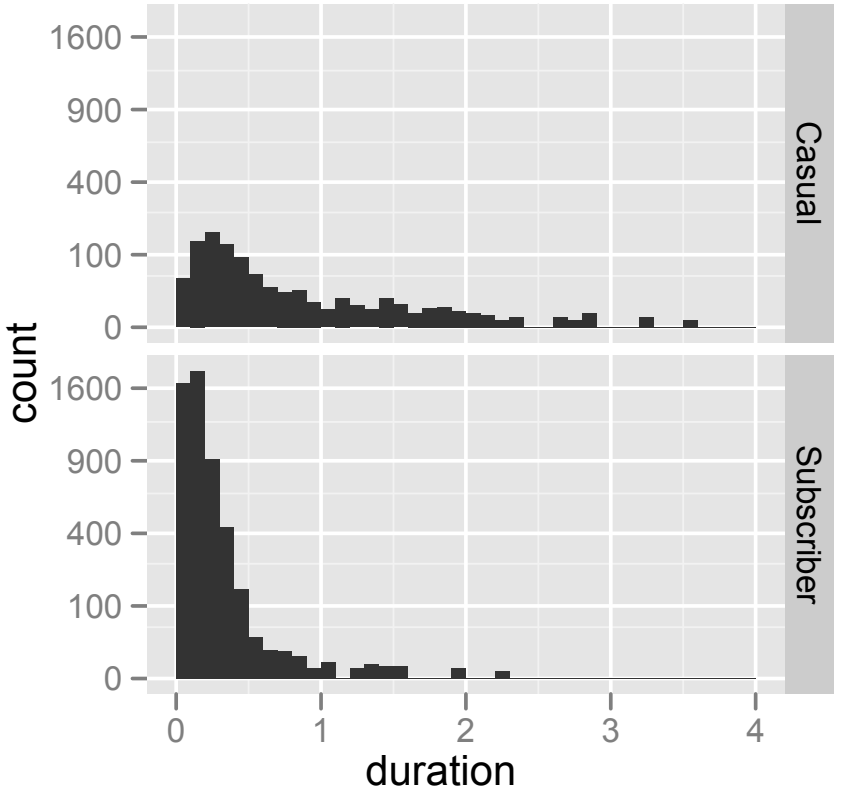
ggplot2 scatterplots

```
p <- ggplot(rides)
p <- p + geom_smooth(aes(x=startHour, y=distance))
p <- p + coord_cartesian(ylim=c(1000,2500))
```



ggplot2 histograms

```
p <- ggplot(rides)
p <- p + geom_histogram(aes(x=duration), binwidth = .1)
p <- p + scale_y_sqrt()
p <- p + facet_grid(subscription ~ .)
p <- p + scale_x_continuous(limits=c(0, 4))
```



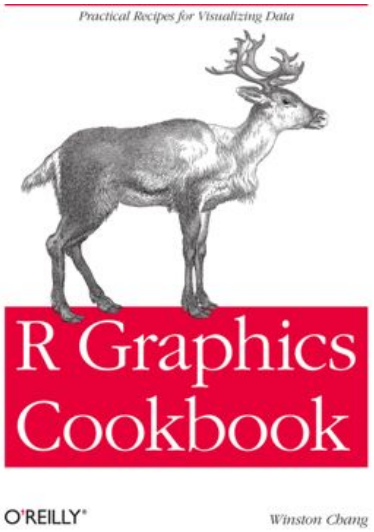
Outline

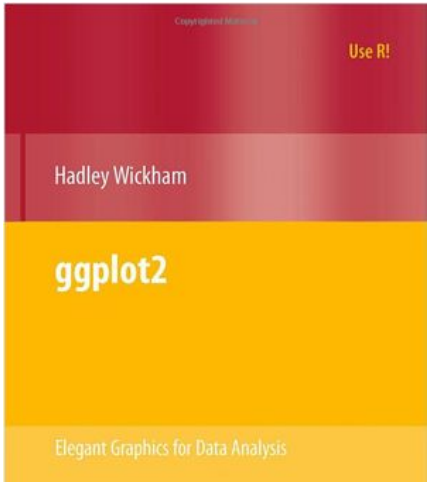
We've done a lot

- You don't have to be able to do everything we did today
- You have to be able to do some of it
- Play around with the way of manipulating data you feel most comfortable with

Further help

- You've just scratched the surface with ggplot2.
- Practice
- Read the docs (either locally in R or at `http://docs.ggplot2.org/current/`)
- Work together





First assignment

- Find some data
- Edit it so it is in a usable form
- Find interesting relationships in your data
- Use Rattle/ggplot2 to display those relationships (be creative and thorough!)