Introduction to Data Science Using R Part IV

Justin Post

August 13-14, 2018

What do we want to be able to do?

- Read in data
- Manipulate data
- **Plot data**
- **Summarize data**
- Analyze data

Schedule

Day 2

- Logical Statements and Subsetting/Manipulating Data?
- **Numerical and Graphical Summaries**
- Basic Analyses

Numerical and Graphical Summaries

- How to summarize data?
- Depends on data type:
  - Categorical
  - Quantitative

Numerical and Graphical Summaries

- How to summarize categorical data?
- Numerically?
  - Tables (contingency tables)
    - Show frequency of categories
- Graphically?
  - Barplots
  - Piecharts (not recommended)

Numerical and Graphical Summaries

Categorical Data

- Data on titanic passengers in `titanic.csv`

```
titanicData <- read_csv("https://raw.githubusercontent.com/
                      jbpost2/DataScienceR/master/datasets/titanic.csv")
titanicData
```

```
## Parsed with column specification:
## cols(
##   pclass = col_integer(),
##   survived = col_integer(),
##   name = col_character(),
##   sex = col_character(),
##   age = col_double(),
##   sibsp = col_integer(),
##   parch = col_integer(),
##   ticket = col_character(),
##   fare = col_double(),
##   cabin = col_character(),
##   embarked = col_character(),
##   boat = col_character(),
##   body = col_integer(),
```

Numerical and Graphical Summaries

Categorical Data - `table()` function creates counts (see help)

- Summarize embarked (where journey started), survived (survive or not), and sex (binary here, Male or Female)

```
table(titanicData$embarked)
```

```
##
##   C   Q   S
## 270 123 914
```

```
table(titanicData$survived)
```

```
##
##   0   1
## 809 500
```

```
table(titanicData$sex)
```

```
##
## female   male
##    466    843
```

Numerical and Graphical Summaries

Categorical Data - Two-Way tables

```
table(titanicData$survived,
      titanicData$sex)
```

```
##
##     female male
##   0    127  682
##   1    339  161
```

```
table(titanicData$survived,
      titanicData$embarked)
```

```
##
##       C   Q   S
##   0 120  79 610
##   1 150  44 304
```

```
table(titanicData$sex,
      titanicData$embarked)
```

```
##
##            C   Q   S
##   female 113  60 291
##   male   157  63 623
```

8/150

Numerical and Graphical Summaries

Categorical Data - Three way table (order matters!)

```
table(titanicData$sex, titanicData$embarked, titanicData$survived)
```

```
## , ,  = 0
##
##
##            C   Q   S
##   female  11  23  93
##   male   109  56 517
##
## , ,  = 1
##
##
##            C   Q   S
##   female 102  37 198
##   male    48   7 106
```

Numerical and Graphical Summaries

Categorical Data

- Can obtain bivariate info from three way table

```
tab <- table(titanicData$sex, titanicData$embarked, titanicData$survived)

str(tab)
```

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

- Example of an array! 3 dimensions `[ , , ]`

10/150

Numerical and Graphical Summaries

Categorical Data

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#returns embarked vs survived table for females
tab[1, , ]
```

```
## 
##       0   1
##   C  11 102
##   Q  23  37
##   S  93 198
```

Numerical and Graphical Summaries

Categorical Data

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#returns embarked vs survived table for males
tab[2, , ]
```

```
## 
##       0   1
##   C 109  48
##   Q  56   7
##   S 517 106
```

Numerical and Graphical Summaries

Categorical Data

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#returns survived vs sex table for embarked "C"
tab[, 1, ]
```

```
##
##            0   1
##   female  11 102
##   male   109  48
```

Numerical and Graphical Summaries

Categorical Data

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#Survived status for males that embarked at "Q"
tab[2, 2, ]
```

```
##  0  1
## 56  7
```

Numerical and Graphical Summaries

Categorical Data

- Main plot: bar plot and variations on it
- `barplot()` function in base R can be used
- We'll use `ggplot2` in tidyverse! cheatsheet

Numerical and Graphical Summaries

`ggplot2` needs and syntax

Needs:

- Data Frame
- Aesthetic (aes) - maps variables to properties of geom
    - Ex: size, color, and x, y location(s)
- Geom layer(s) (visualizaton type(s))
- Coordinate system (mostly use Cartesian plane)
- Optional: Stat layer, titles, etc.

Numerical and Graphical Summaries

`ggplot2` needs and syntax

Needs:

- Data Frame
- Aesthetic (aes) - maps variables to properties of geom
- Geom layer(s) (visualizaton type(s))
- Optional: Stat layer, titles, etc.
- Syntax:

```
g <- ggplot(dataframe, aes(x = , y = , ...))
g + geom_type(...) +
  stat_type(...) +
  labs(...)
```

Numerical and Graphical Summaries

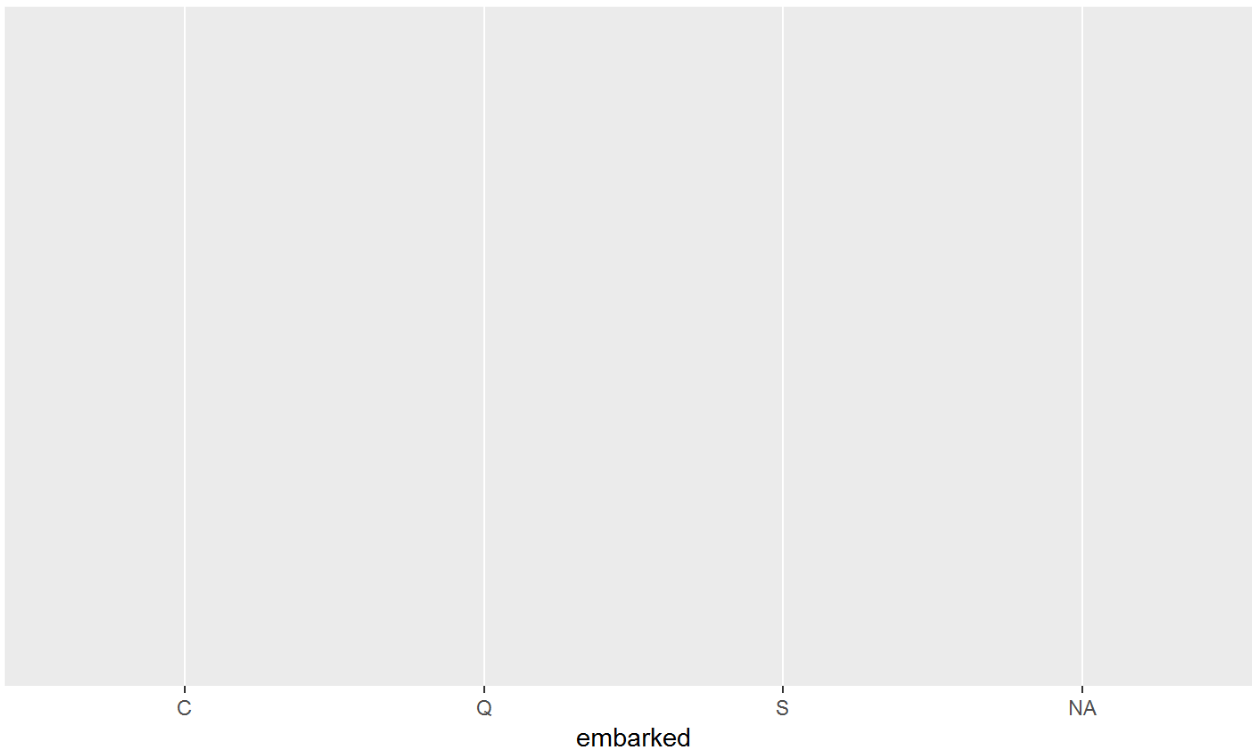
Categorical Data

- Example bar plot:

```
ggplot(data = titanicData, aes(x = embarked))
```

Numerical and Graphical Summaries

- Notice no plot is made
- Must add geom layer!



Numerical and Graphical Summaries

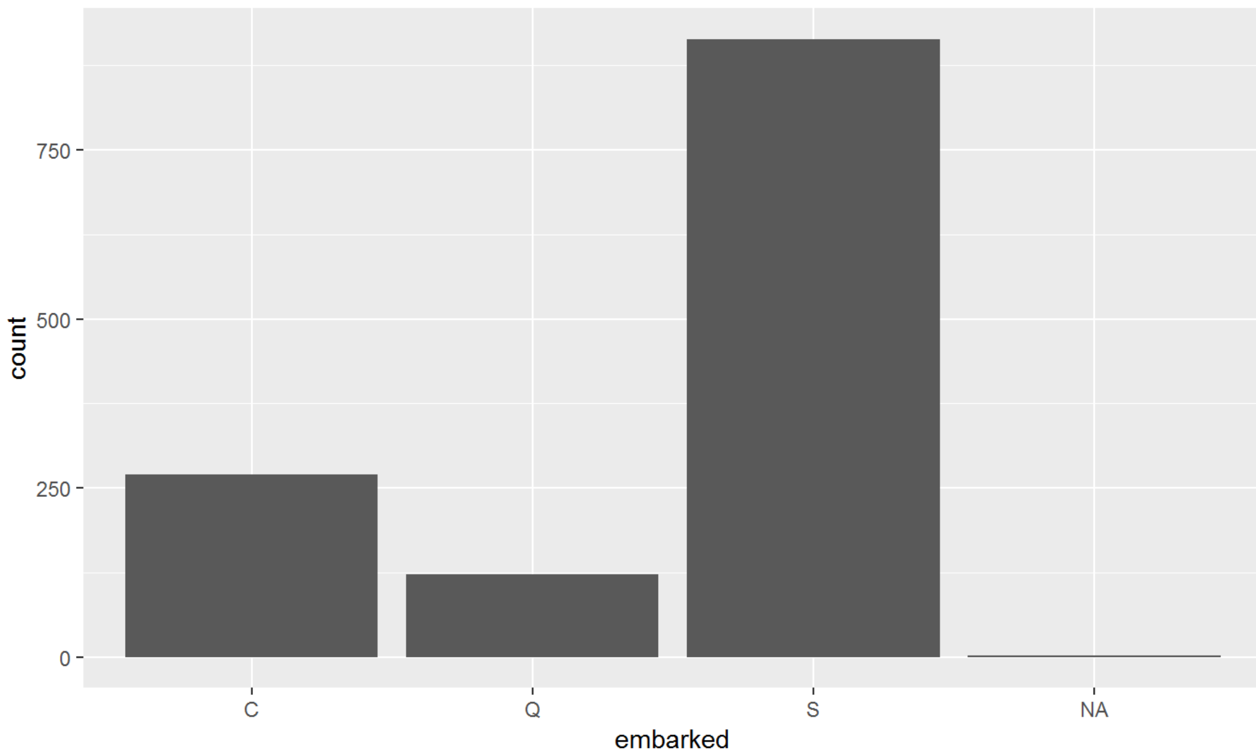
Categorical Data

- Idea: Save base object, then "add layers"

```
g <- ggplot(data = titanicData, aes(x = embarked))
g + geom_bar()
```

Numerical and Graphical Summaries

Categorical Data



Numerical and Graphical Summaries

Categorical Data

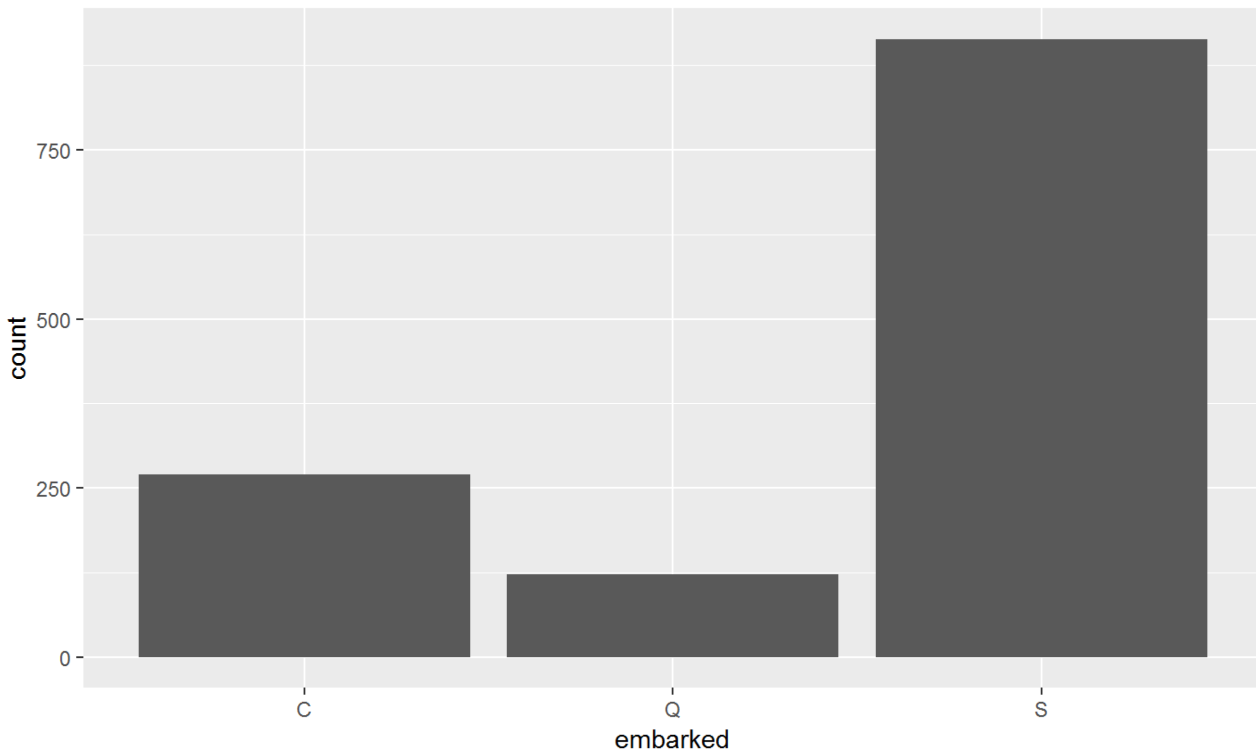
- How to improve this plot?
- Might remove `NA` category

```
titanicData <- titanicData %>% drop_na(embarked)
g <- ggplot(data = titanicData, aes(x = embarked))
g + geom_bar()
```

22/150

Numerical and Graphical Summaries

Categorical Data



Numerical and Graphical Summaries

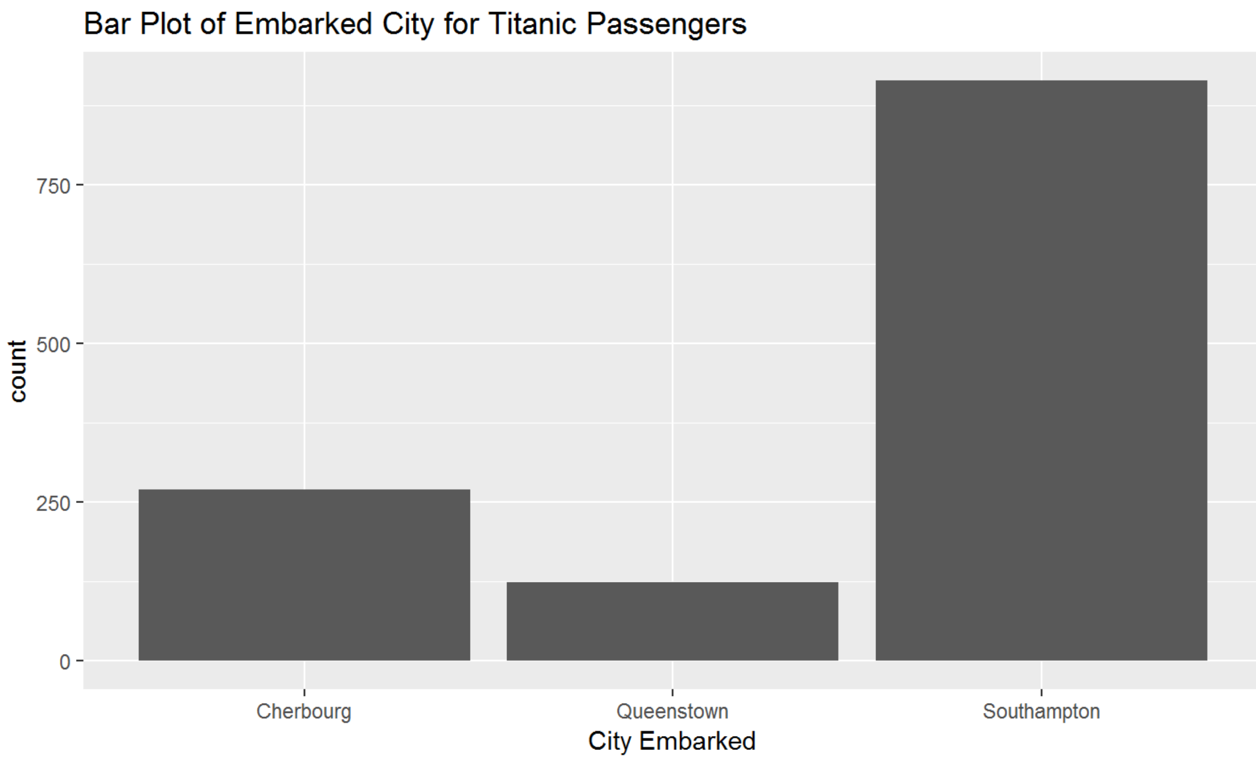
Categorical Data

- How to improve this plot?
- Might add better labels and a title

```
#Fix x axis, x axis label and give title
g + geom_bar() +
  labs(x = "City Embarked", title = "Bar Plot of Embarked City
       for Titanic Passengers") +
  scale_x_discrete(labels = c("Cherbourg","Queenstown","Southampton"))
```

Numerical and Graphical Summaries

Categorical Data



25/150

Numerical and Graphical Summaries

Categorical Data

- Previous plot visualized table for one variable
- How to visualize table for two?
  - Filled bar plot
  - Side-by-side bar plot (Requires work!)
- Process the same
  - Create base object
  - Add geoms
  - Use aes to specify aspects of the plot

Numerical and Graphical Summaries

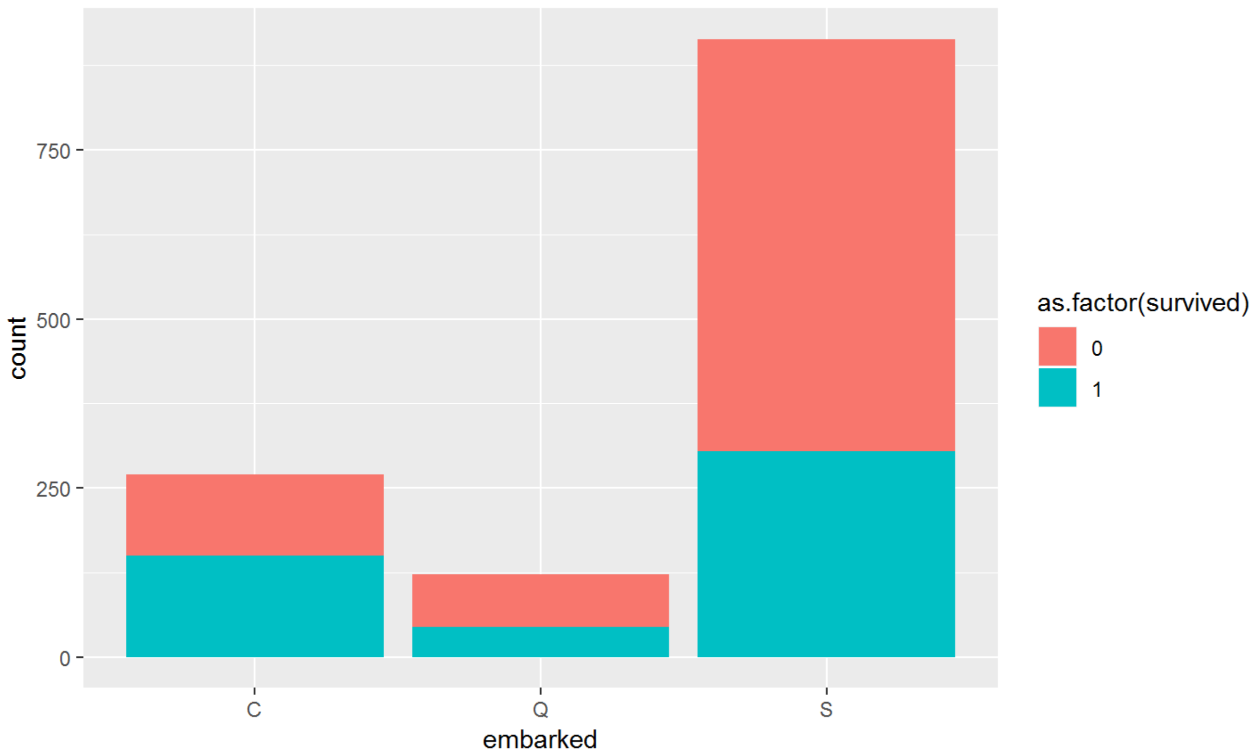
Categorical Data

- Filled bar plot

```
g + geom_bar(aes(fill = as.factor(survived)))
```

Numerical and Graphical Summaries

Categorical Data



Numerical and Graphical Summaries

Categorical Data

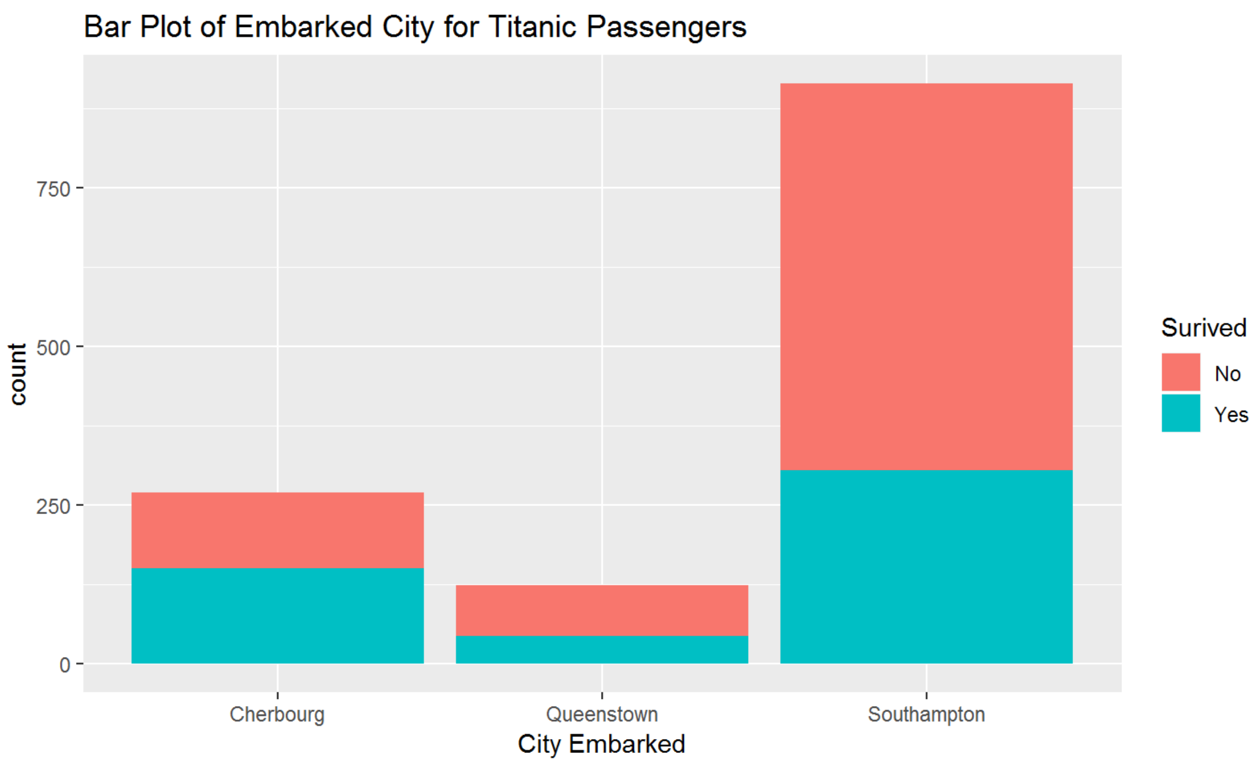
Add labels and such:

```
g + geom_bar(aes(fill = as.factor(survived))) +
  labs(x = "City Embarked",
       title = "Bar Plot of Embarked City for Titanic Passengers") +
  scale_x_discrete(labels = c("Cherbourg", "Queenstown", "Southampton")) +
  scale_fill_discrete(name = "Surived", labels = c("No","Yes"))
```

29/150

Numerical and Graphical Summaries

Categorical Data



Numerical and Graphical Summaries

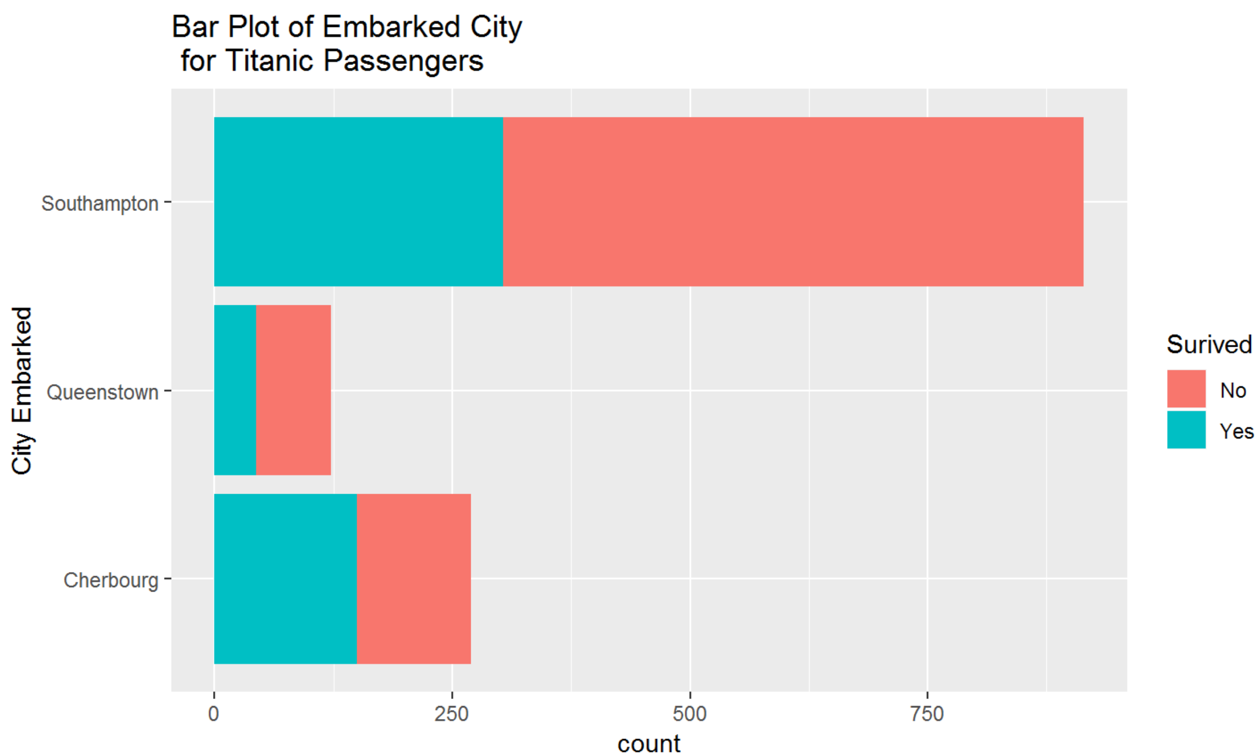
Categorical Data

- Can rotate it

```
g + geom_bar(aes(fill = as.factor(survived))) +
  labs(x = "City Embarked",
       title = "Bar Plot of Embarked City for Titanic Passengers") +
  scale_x_discrete(labels = c("Cherbourg", "Queenstown", "Southampton")) +
  scale_fill_discrete(name = "Surived",labels = c("No","Yes")) +
  coord_flip()
```

Numerical and Graphical Summaries

Categorical Data



32/150

Numerical and Graphical Summaries

Categorical Data

- Side-by-side bar plot
- First, create data frame with summary info

```
twoWayData <- titanicData %>% drop_na(embarked) %>% group_by(embarked, survived) %>%
  summarise(count = n())
```

```
## # A tibble: 6 x 3
## # Groups:   embarked [?]
##   embarked survived count
##   <chr>       <int> <int>
## 1 C               0   120
## 2 C               1   150
## 3 Q               0    79
## 4 Q               1    44
## 5 S               0   610
## 6 S               1   304
```

33/150

Numerical and Graphical Summaries

Categorical Data

- Side-by-side bar plot

```
g <- ggplot(data = twoWayData,
       aes(x = embarked,y = count, fill = as.factor(survived)))
```

Numerical and Graphical Summaries

Categorical Data

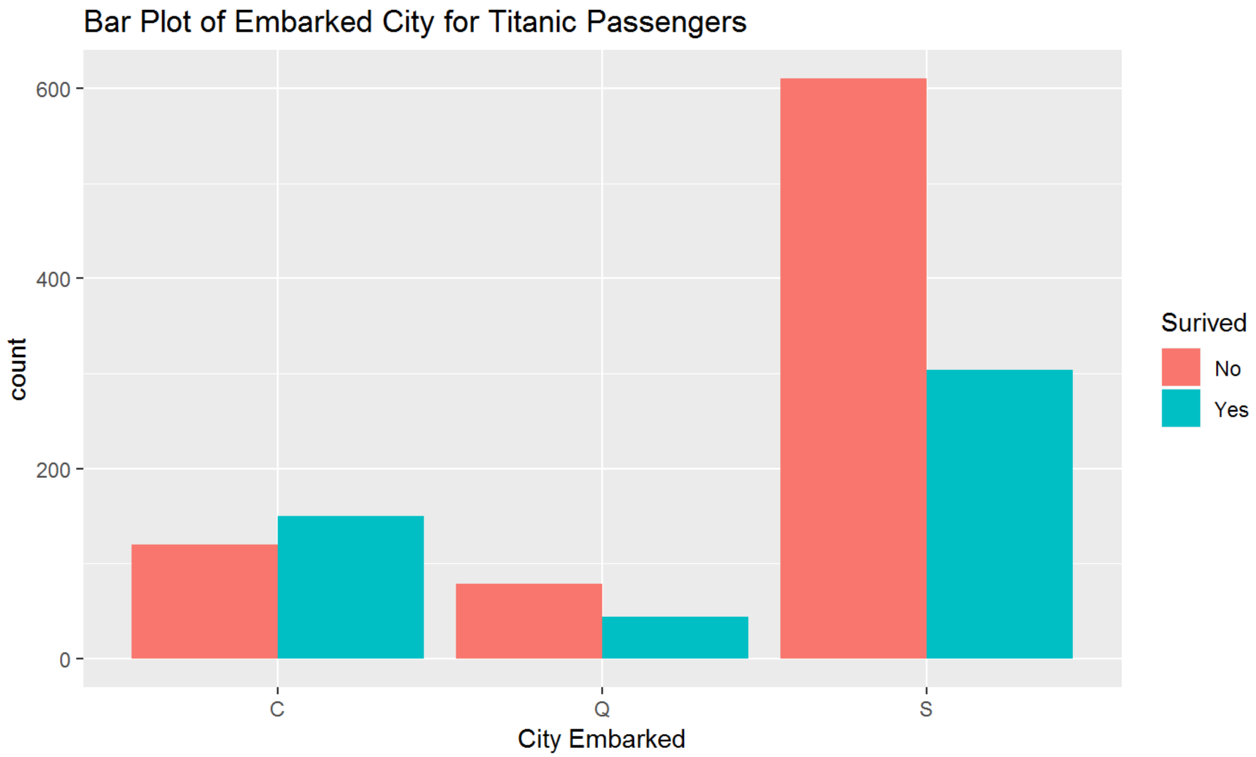
- Side-by-side bar plot

```
g <- ggplot(data = twoWayData,
      aes(x = embarked,
y = count
, fill = as.factor(survived)))

g + geom_bar(stat = "identity", position = "dodge") +
    labs(x = "City Embarked",
         title = "Bar Plot of Embarked City for Titanic Passengers") +
  scale_x_discrete(labels = c("Cherbourg", "Queenstown", "Southampton")) +
  scale_fill_discrete(name = "Surived", labels = c("No", "Yes"))
```

Numerical and Graphical Summaries

Categorical Data



36/150

Numerical and Graphical Summaries

Categorical Data

- How to save tables and graphs?
  - Save tables with write_csv!

```
tab <- tbl_df(table(titanicData$embarked, titanicData$survived))

names(tab) <- c("Embarked", "Survived", "Count")

write_csv(x = tab, path = "titanicTable.csv", col_names = TRUE)
```

Numerical and Graphical Summaries

Categorical Data

- How to save tables and graphs?
  - Save graphs with
    - ggsave()
    - 'Export' button

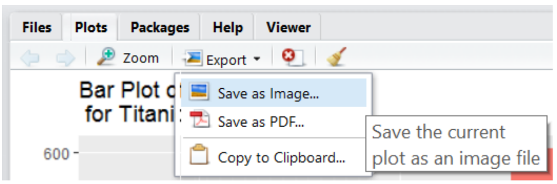
```
#by default ggsave saves last plot
#guesses file type by extension
ggsave(filename = "output/titanicBarPlot.png")
ggsave(filename = "output/titanicBarPlot.pdf")
```

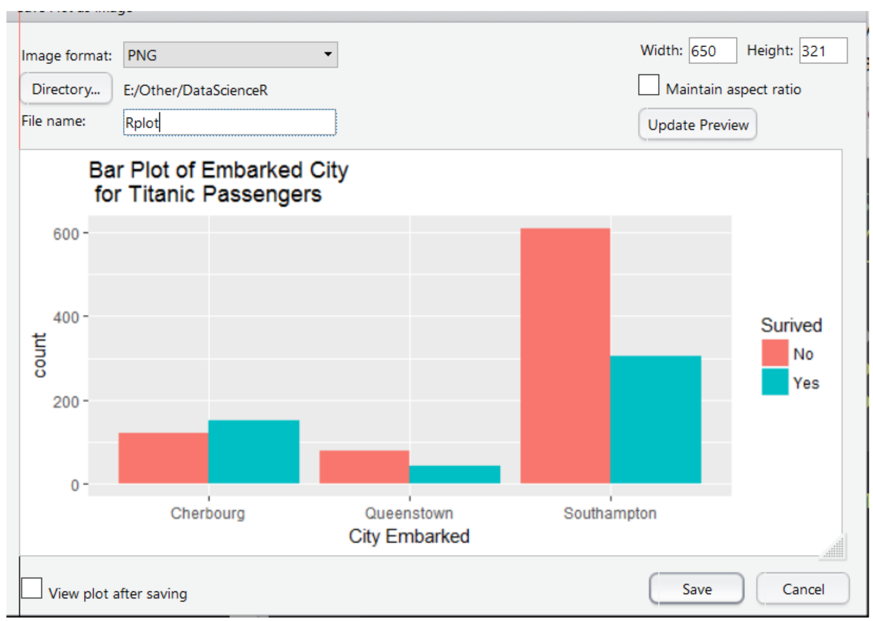
38/150

Numerical and Graphical Summaries

Categorical Data

- 'Export' button





Recap!

- How to summarize categorical data?
- Numerically?
  - Tables (contingency tables)
    - Show frequency of categories
- Graphically?
  - Barplots
- ggplot (create object, add layers)
  - Data Frame
  - Geoms (Vis type)
  - Aesthetic (aes)
  - Coordinate system, stat, labels, etc.

Activity

- **Categorical Plots Activity** instructions available on web
- Work in small groups
- Ask questions! TAs and I will float about the room
- Feel free to ask questions about anything you didn't understand as well!

Numerical and Graphical Summaries

- How to summarize data?
- Depends on data type:
  - Categorical
  - Quantitative

Numerical and Graphical Summaries

- How to summarize quantitative data?
- Numerically?
- One Variable:
  - Measures of center
    - Mean, Median
  - Measures of spread
    - Variance, Standard Deviation, Quartiles, IQR
- Two Variables:
  - Measures of linear relationship
    - Covariance, Correlation
  - Can do any of above for subgroups of data!

Numerical and Graphical Summaries

- Graphically?
- One Variable:
  - Dot Plot, Histogram, or Kernel Smoother
  - Empirical Cumulative Distribution Function
- Two Variables:
  - Scatter Plot (with trend lines/smoothers, rug)
  - Side-by-side boxplots/violin plots
  - Line plot
- Three Variables:
  - 3D scatter plot
- Can do any of above for subgroups of data!

Numerical and Graphical Summaries

Quantitative Data

- Look at CO2 uptake data set
  - Carbon Dioxide Uptake in Grass Plants

```
CO2 <- tbl_df(CO2)
CO2
```

```
## # A tibble: 84 x 5
##   Plant Type   Treatment   conc uptake
## * <ord> <fct>  <fct>      <dbl>  <dbl>
## 1 Qn1   Quebec nonchilled    95   16
## 2 Qn1   Quebec nonchilled   175   30.4
## 3 Qn1   Quebec nonchilled   250   34.8
## 4 Qn1   Quebec nonchilled   350   37.2
## 5 Qn1   Quebec nonchilled   500   35.3
## # ... with 79 more rows
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: measures of center

```
mean(CO2$uptake)
```

```
## [1] 27.2131
```

```
#note you can easily get a trimmed mean
mean(CO2$uptake, trim = 0.05) #5% trimmed mean
```

```
## [1] 27.25263
```

```
median(CO2$uptake)
```

```
## [1] 28.3
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: measures of spread

```
#quartiles and mean
summary(CO2$uptake)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    7.70   17.90   28.30   27.21   37.12   45.50
```

```
var(CO2$uptake)
```

```
## [1] 116.9515
```

```
sd(CO2$uptake)
```

```
## [1] 10.81441
```

```
IQR(CO2$uptake)
```

```
## [1] 19.225
```

```
quantile(CO2$uptake, probs = c(0.1, 0.2))
```

```
##   10%   20% 
## 12.36 15.64
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Covariance/Correlation

```
cov(CO2$conc, CO2$uptake)
```

```
## [1] 1552.687
```

```
cor(CO2$conc, CO2$uptake)
```

```
## [1] 0.4851774
```

Numerical and Graphical Summaries

Quantitative Data - Summaries on subgroups of data

- Get same uptake stats for uptake **by Treatment**
- `aggregate()` function pretty good
- We'll use `dplyr` (not as flexible)
- Combine
    - group_by
    - summarise

Numerical and Graphical Summaries

Quantitative Data - Summaries on subgroups of data

```
CO2 %>% group_by(Treatment) %>% summarise(avg = mean(uptake))
```

```
## # A tibble: 2 x 2
##   Treatment    avg
##   <fct>      <dbl>
## 1 nonchilled  30.6
## 2 chilled     23.8
```

```
CO2 %>% group_by(Treatment) %>% summarise(med = median(uptake))
```

```
## # A tibble: 2 x 2
##   Treatment    med
##   <fct>      <dbl>
## 1 nonchilled  31.3
## 2 chilled     19.7
```

```
CO2 %>% group_by(Treatment) %>% summarise(var = var(uptake))
```

Numerical and Graphical Summaries

Quantitative Data - Summaries on subgroups of data

- Can refine by more than one variable grouping

```
CO2 %>% group_by(Treatment, Type) %>% summarise(avg = mean(uptake))
```

```
## # A tibble: 4 x 3
## # Groups:   Treatment [?]
##   Treatment  Type          avg
##   <fct>      <fct>       <dbl>
## 1 nonchilled Quebec       35.3
## 2 nonchilled Mississippi  26.0
## 3 chilled    Quebec       31.8
## 4 chilled    Mississippi  15.8
```

Numerical and Graphical Summaries

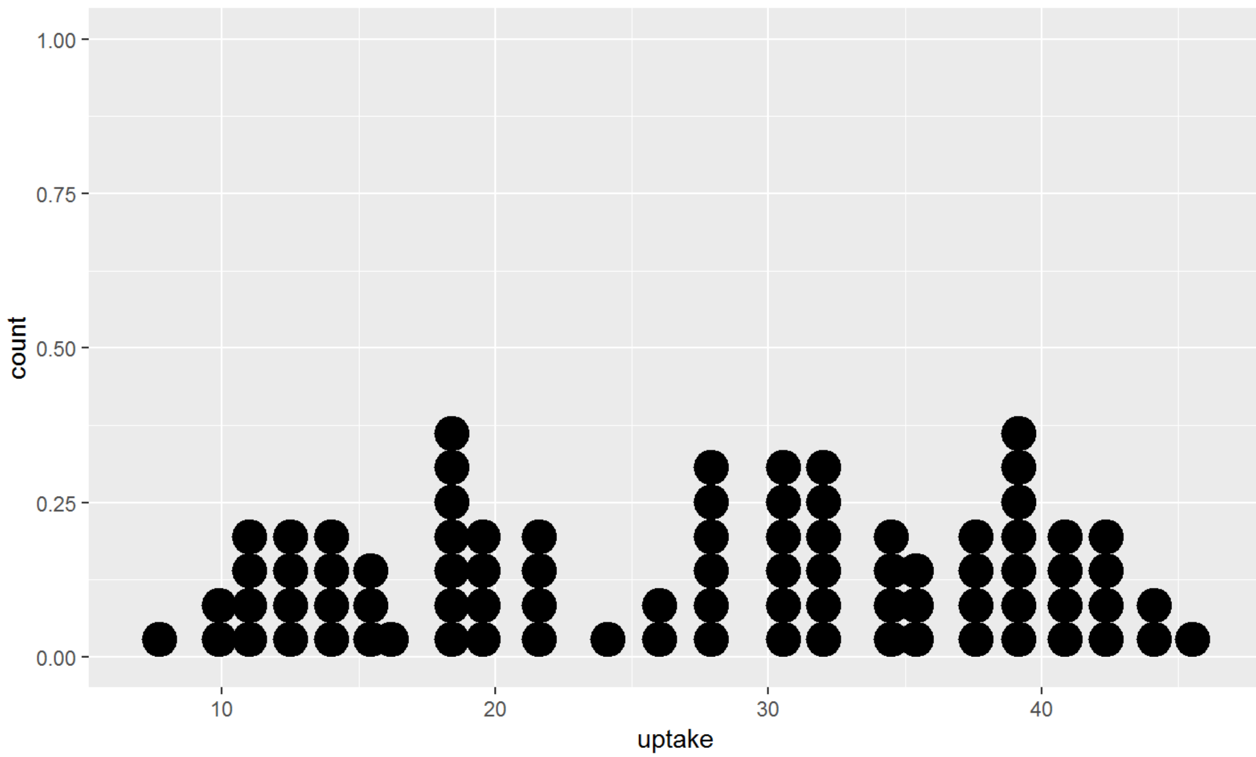
Quantitative Data - One Variable: Dot Plot

- Dot shown to represent each data point

```
g <- ggplot(CO2, aes(x = uptake))
g + geom_dotplot()
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: Dot Plot



Numerical and Graphical Summaries

Quantitative Data - One Variable: Dot Plot

- Color dots:
  - **Any attribute that depends on the data must go into aes**
  - If consistent can go outside aes (no legend needed)

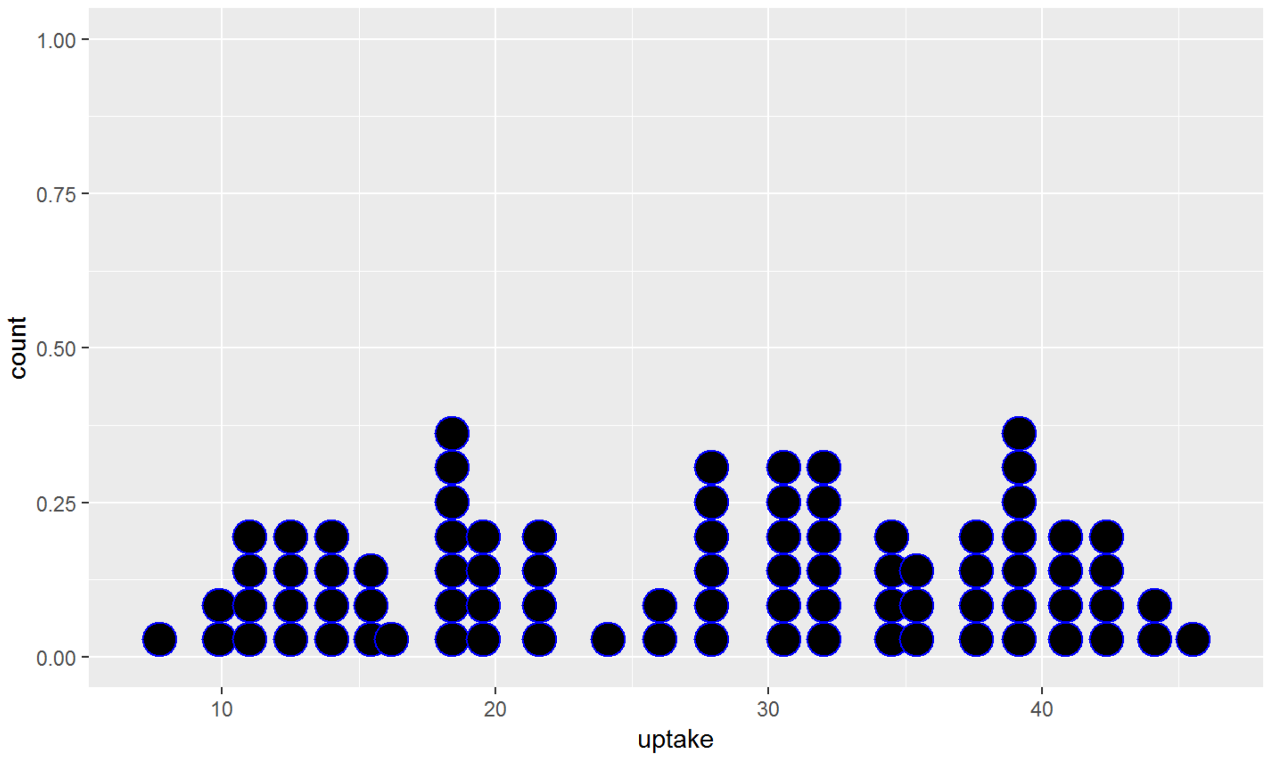
```
g + geom_dotplot(col = "Red")
```

```
#vs
```

```
g + geom_dotplot(aes(col = Treatment))
```

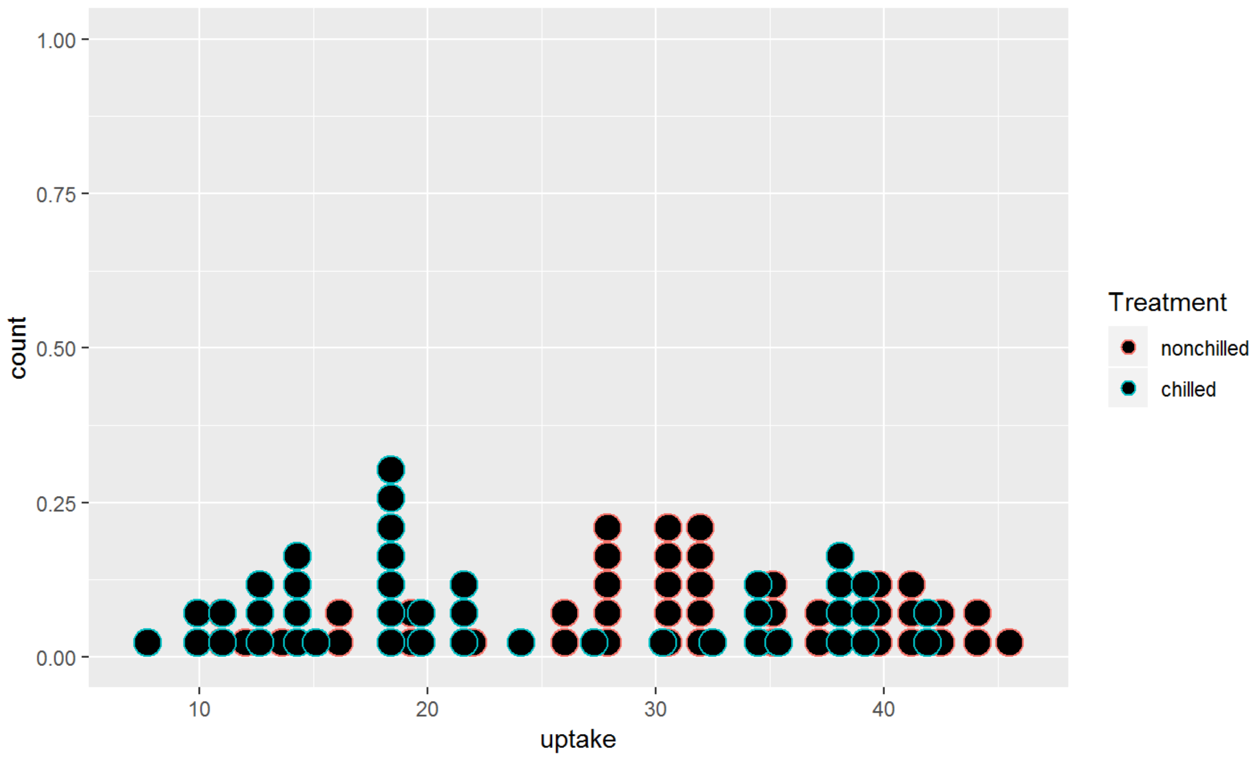
Numerical and Graphical Summaries

Quantitative Data - One Variable: Dot Plot



Numerical and Graphical Summaries

Quantitative Data - One Variable: Dot Plot



56/150

Numerical and Graphical Summaries

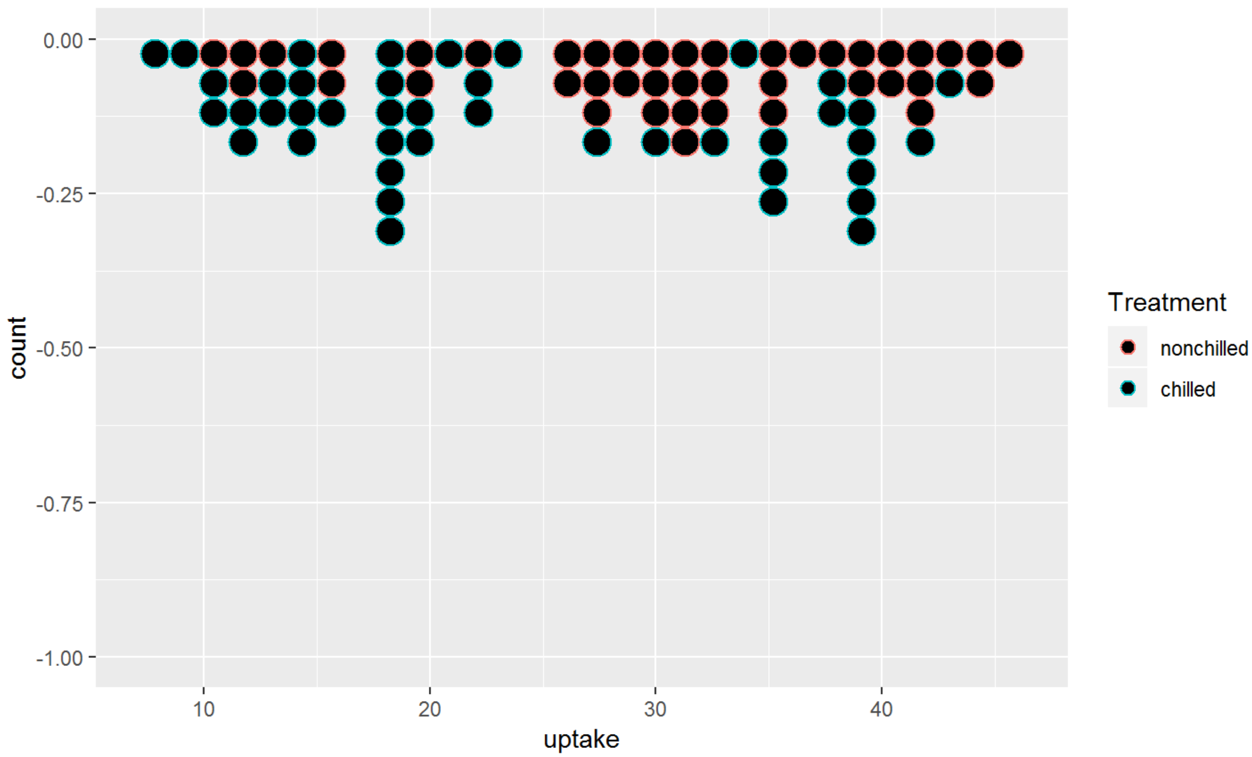
Quantitative Data - One Variable: Dot Plot

- Lots of options!

```
g + geom_dotplot(aes(col = Treatment),
               stackgroups = TRUE, method = "histodot",
               binpositions = "all",stackdir = "down")
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: Dot Plot



58/150

Numerical and Graphical Summaries

Quantitative Data - One Variable: Histogram

- Bins data to show distribution of observations

```
g + geom_histogram()
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: Histogram

Numerical and Graphical Summaries

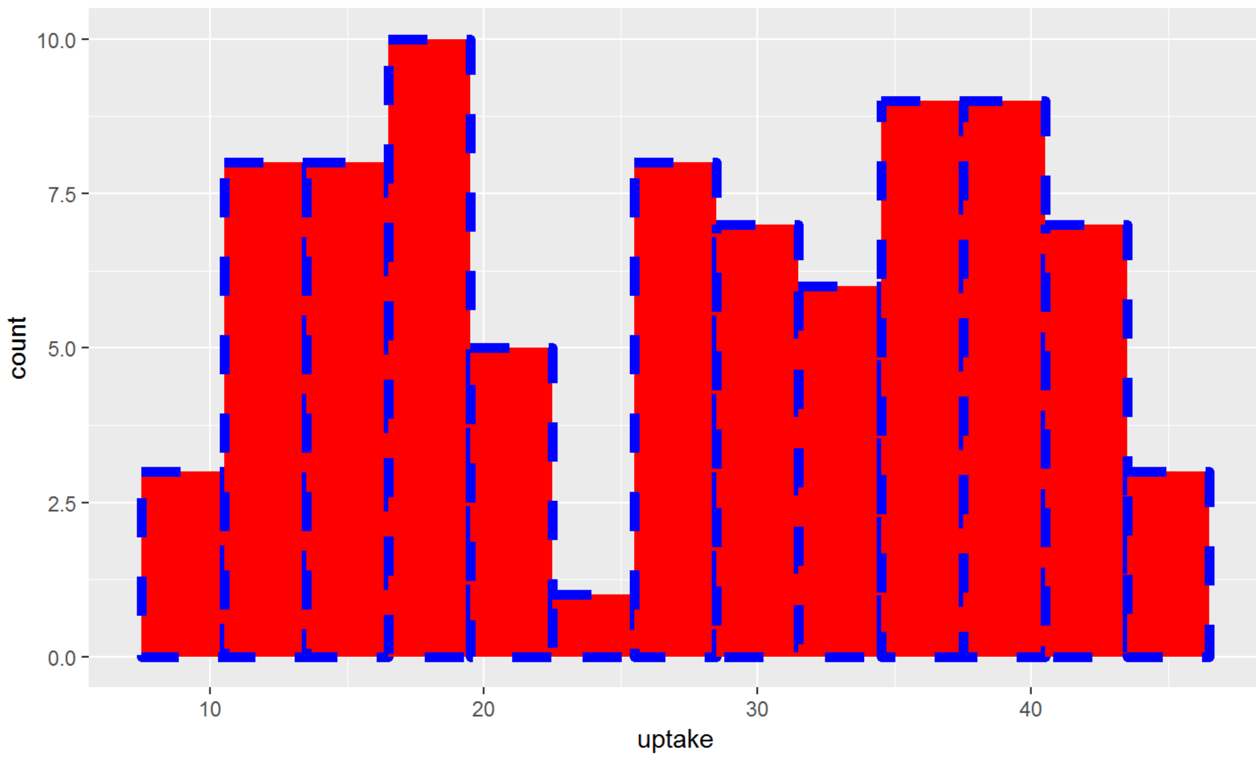
Quantitative Data - One Variable: Histogram

- Can improve the look

```
g + geom_histogram(color = "blue", fill = "red", linetype = "dashed",
                   size = 2, binwidth = 3)
```

Numerical and Graphical Summaries

Quantitative Data Hideous!



62/150

Numerical and Graphical Summaries

Quantitative Data - One Variable: Kernel Smoother

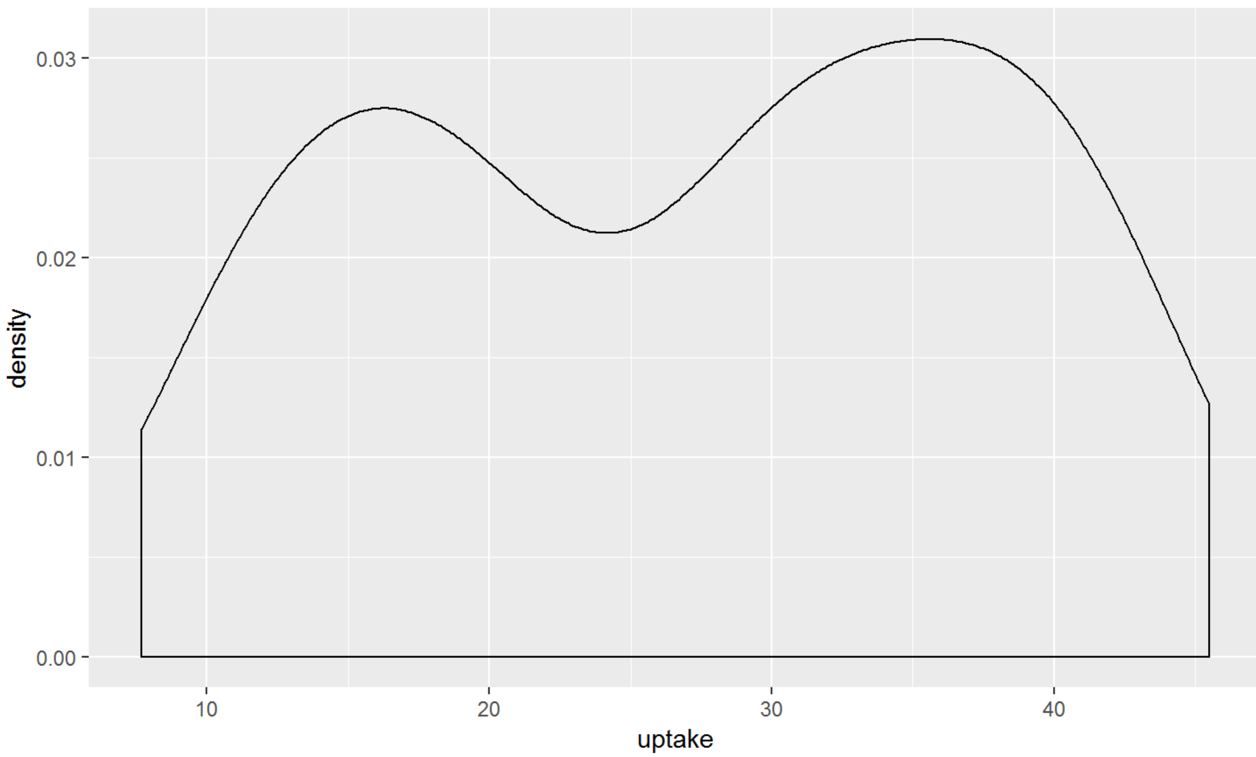
- Smoothed version of a histogram
- Kernel determines weight given to nearby points
  - Many options (see `help(density)`)

```
g + geom_density()
```

63/150

Numerical and Graphical Summaries

Quantitative Data - One Variable: Kernel Smoother



64/150

Numerical and Graphical Summaries

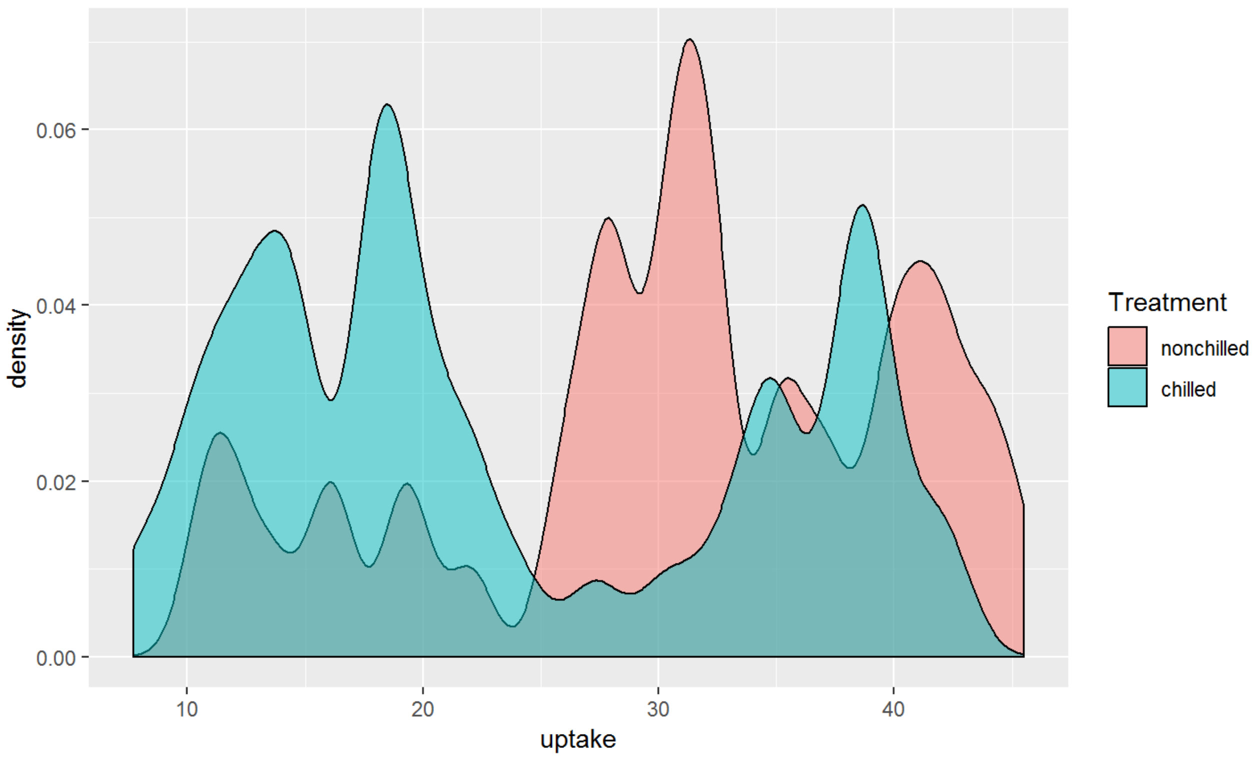
Quantitative Data - One Variable: Kernel Smoother

- Improve it with options!

```
g + geom_density(adjust = 0.25, alpha = 0.5, aes(fill = Treatment))
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: Kernel Smoother



66/150

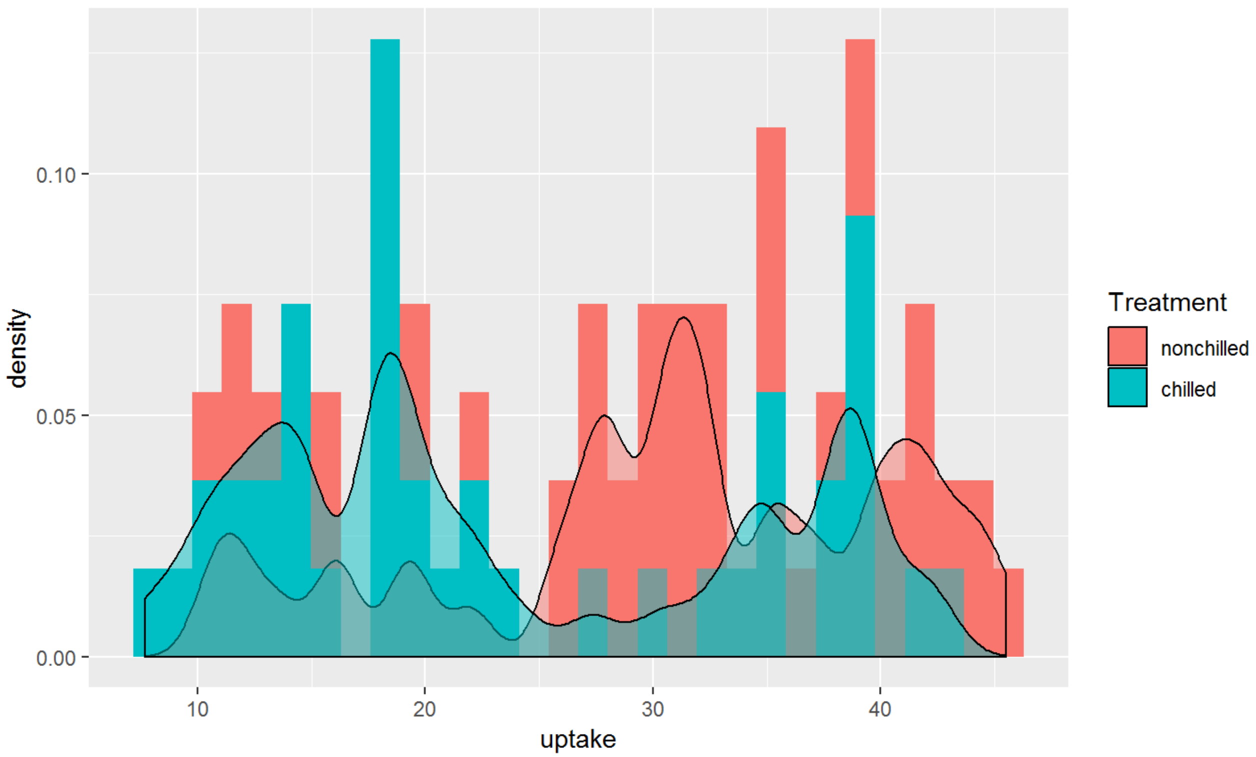
Numerical and Graphical Summaries

Quantitative Data - One Variable: Histogram and Kernel Smoother

```
g + geom_histogram(aes(y = ..density.., fill = Treatment)) +
  geom_density(adjust = 0.25, alpha = 0.5, aes(fill = Treatment))
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: Histogram and Kernel Smoother



Numerical and Graphical Summaries

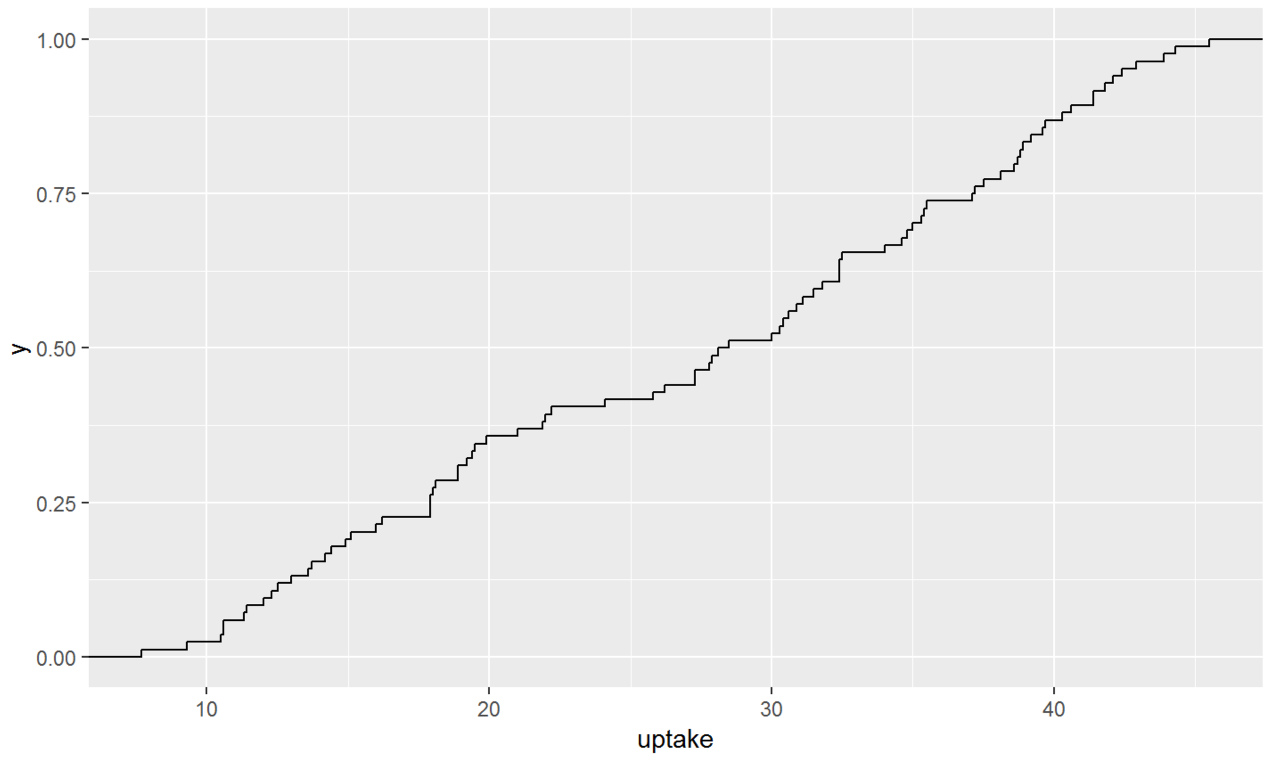
Quantitative Data - One Variable: ECDF

- Empirical Cumulative Distribution Function (ECDF)
- At each point x, gives the proportion of points at or below x

```
g + stat_ecdf(geom = "step")
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: ECDF



70/150

Numerical and Graphical Summaries

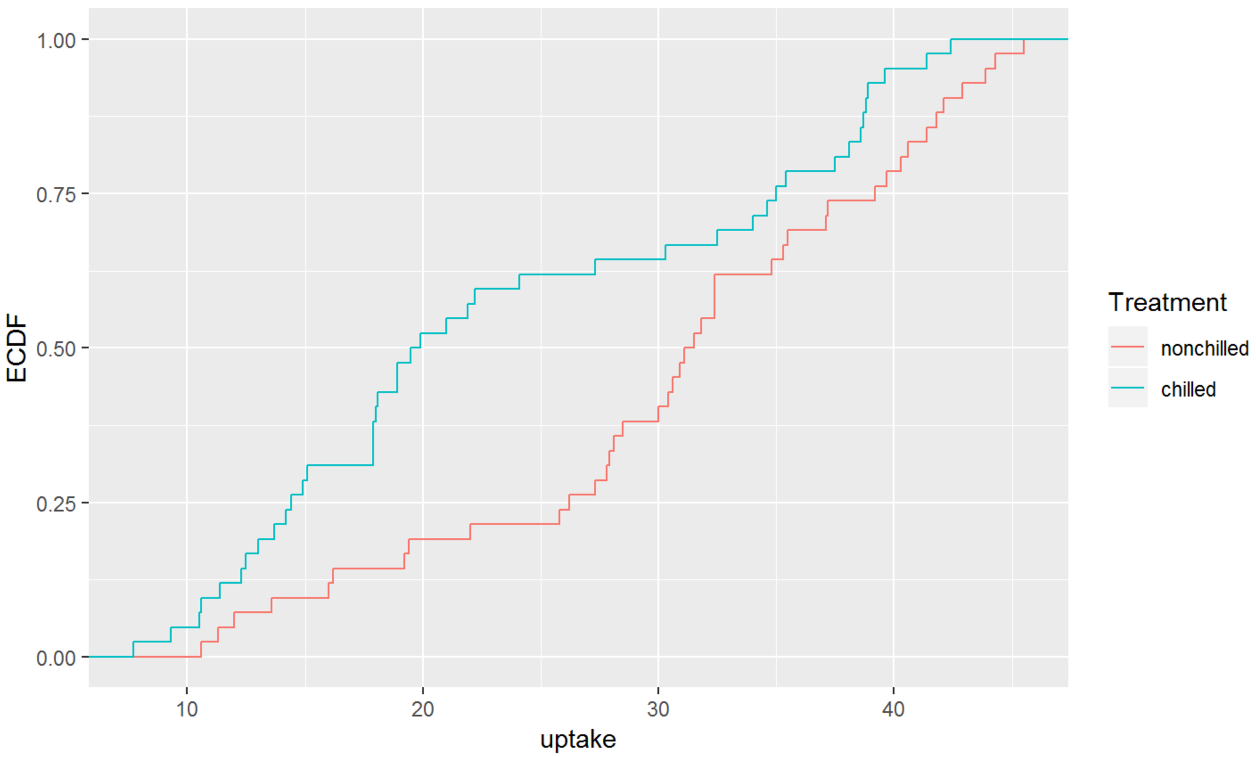
Quantitative Data - One Variable: ECDF

- Empirical Cumulative Distribution Function (ECDF)
- At each point x, gives the proportion of points at or below x
- For each treatment

```
g + stat_ecdf(geom = "step", aes(color = Treatment)) +
  ylab("ECDF")
```

Numerical and Graphical Summaries

Quantitative Data - One Variable: ECDF



Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot

- Probably most used graph!
- Shows a point corresponding to each observation

```
scoresFull<-read_csv("https://raw.githubusercontent.com/
                     jbpost2/DataScienceR/master/datasets/scoresFull.csv")
scoresFull
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   week = col_character(),
##   date = col_character(),
##   day = col_character(),
##   awayTeam = col_character(),
##   homeTeam = col_character(),
##   stadium = col_character(),
##   startTime = col_time(format = ""),
##   toss = col_character(),
##   roof = col_character(),
##   surface = col_character(),
```

Numerical and Graphical Summaries

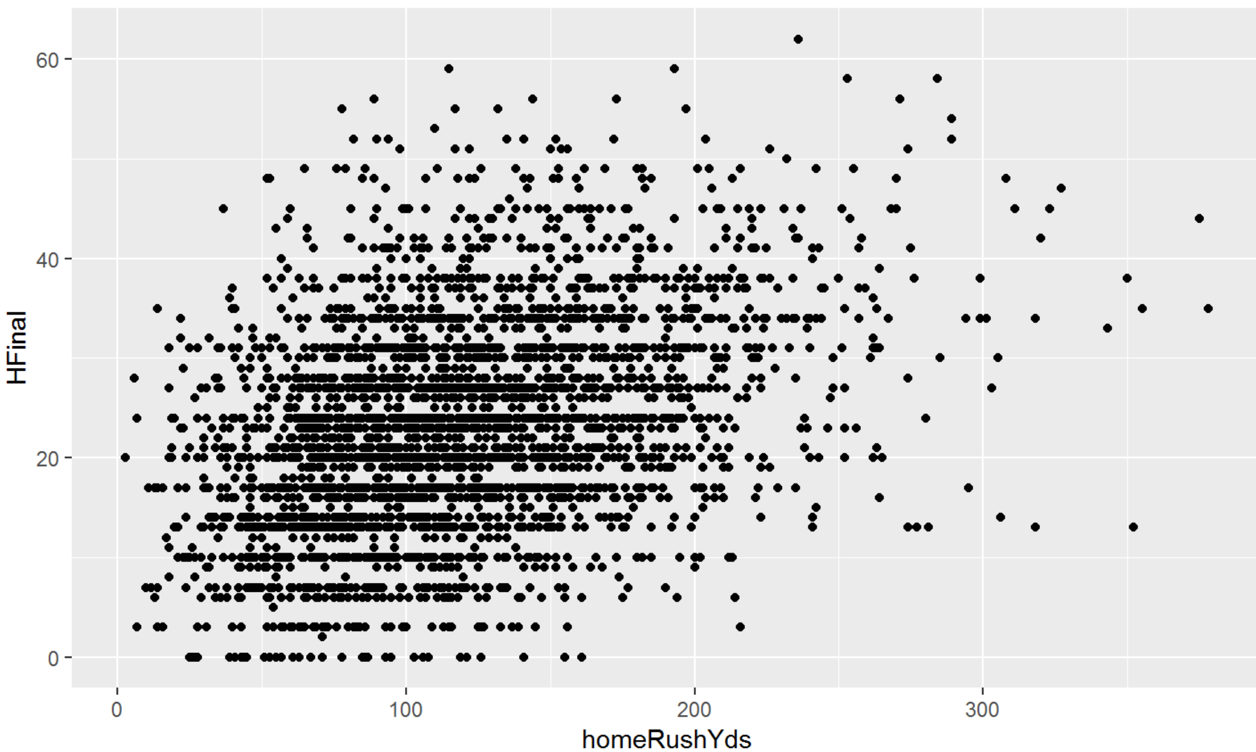
Quantitative Data - Two Variables: Scatter Plot

- Probably most used graph!
- Shows a point corresponding to each observation

```
g <- ggplot(scoresFull, aes(x = homeRushYds, y = HFinal))
g + geom_point()
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot



75/150

Numerical and Graphical Summaries

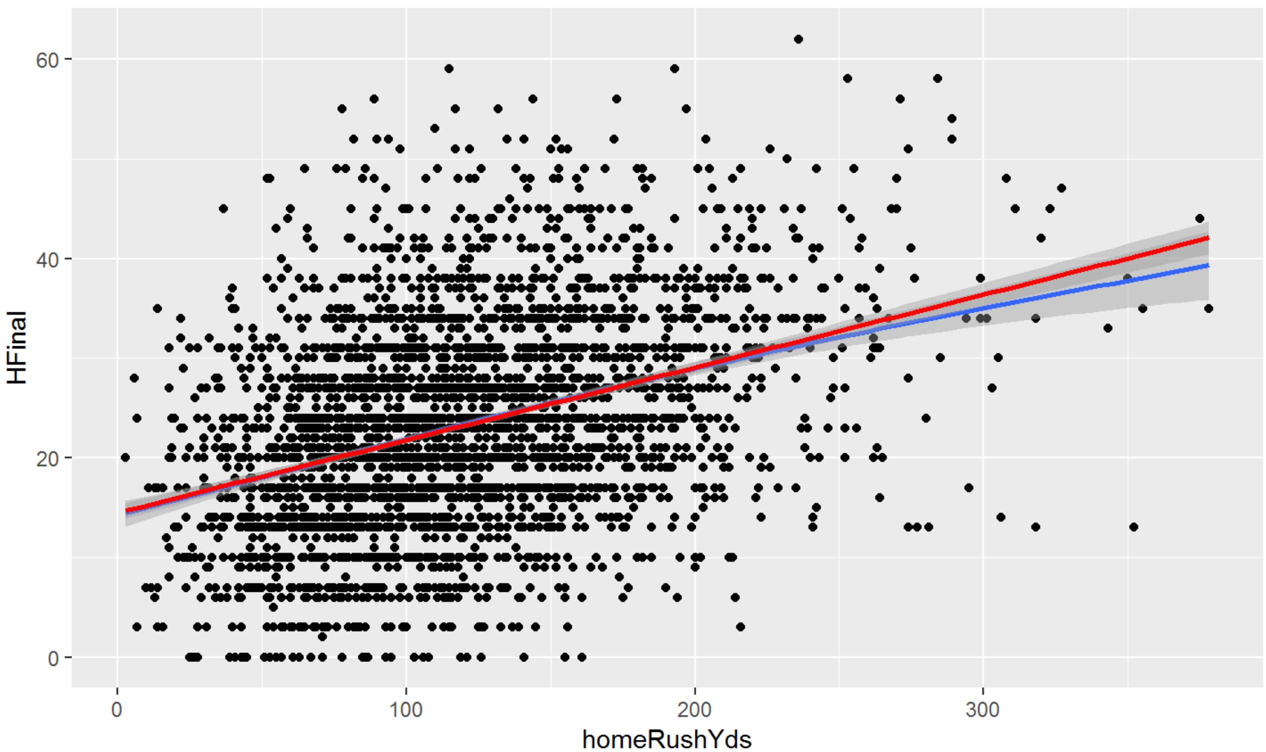
Quantitative Data - Two Variables: Scatter Plot

- Add trend lines (linear and gam - a fancy smoother)

```
g + geom_point() +
    geom_smooth() +
    geom_smooth(method = lm, col = "Red")
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot



77/150

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot

- May want to add text to plot
- Ex: Add value of correlation to plot

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot

- May want to add text to plot
- Ex: Add value of correlation to plot
- `paste()` and `paste0()` very useful! (see help)

```
paste("Hi", "What", "Is", "Going", "On", "?", sep = " ")
```

```
## [1] "Hi What Is Going On ?"
```

```
paste("Hi", "What", "Is", "Going", "On", "?", sep = ".")
```

```
## [1] "Hi.What.Is.Going.On.?"
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot

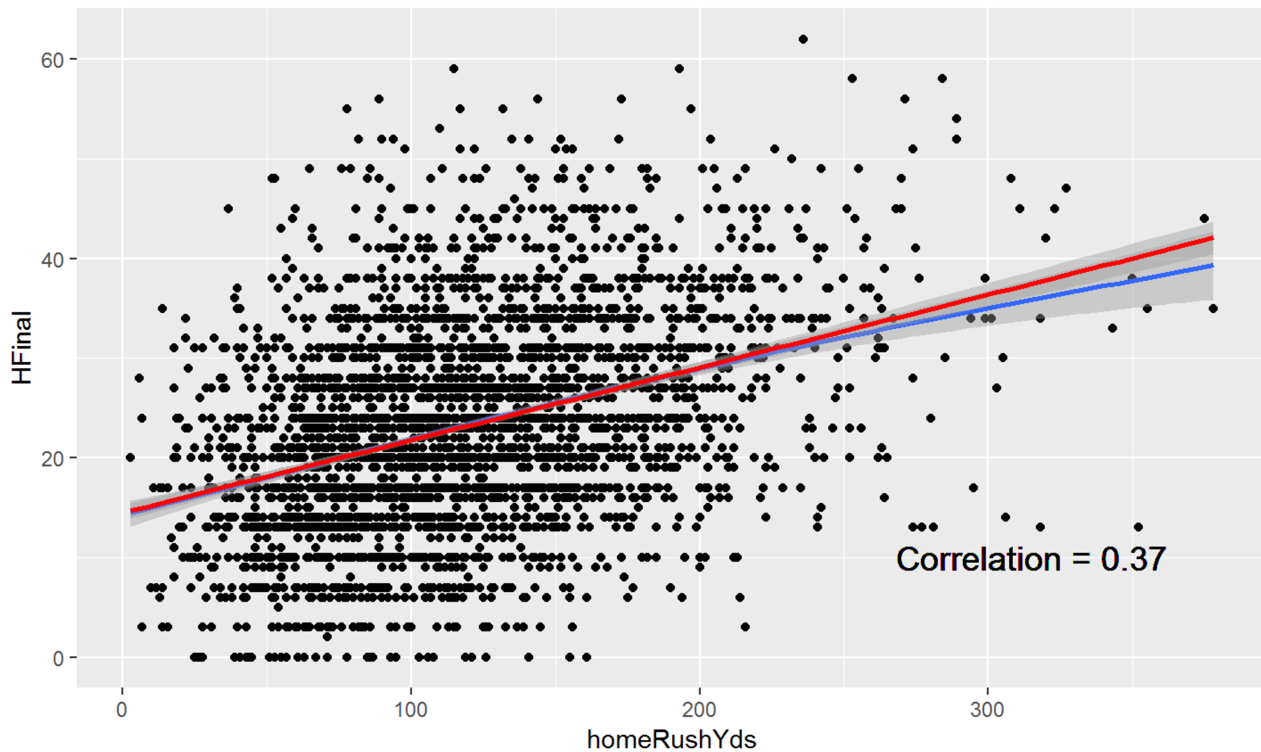
- Ex: Add value of correlation to plot

```
correlation <- cor(scoresFull$homeRushYds,scoresFull$HFinal)

g + geom_point() +
    geom_smooth() +
    geom_smooth(method = lm, col = "Red") +
    geom_text(x = 315, y = 10, size = 5, label = paste0("Correlation = ",
                                                 round(correlation, 2)))
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot



Numerical and Graphical Summaries

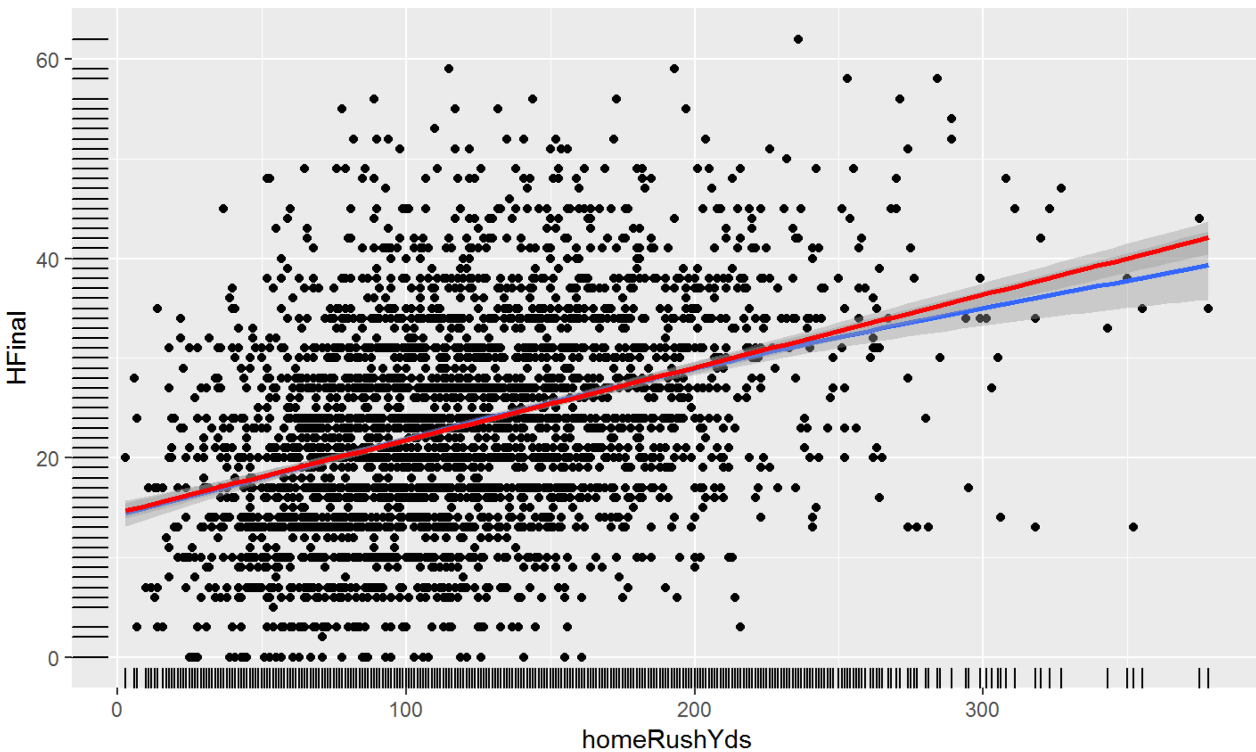
Quantitative Data - Two Variables: Scatter Plot

- Can add "rug"
- Gives idea about density of just x or just y variable

```
g + geom_point() +
    geom_smooth() +
    geom_smooth(method = lm, col = "Red") +
    geom_rug()
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot



83/150

Numerical and Graphical Summaries

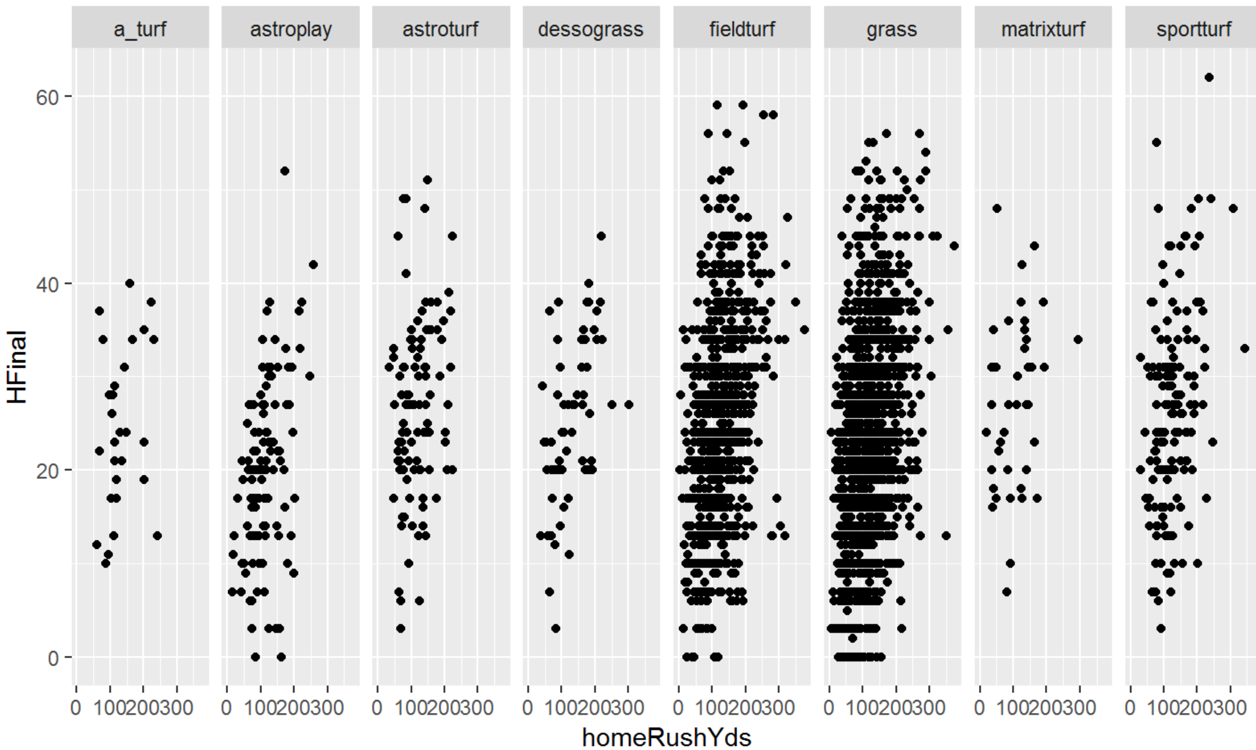
Quantitative Data - Two Variables: Scatter Plot

- Look at graphs broken down by other variables
- Use `facet_grid()`

```
g + geom_point() +
    facet_grid(. ~ surface)
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot



85/150

Numerical and Graphical Summaries

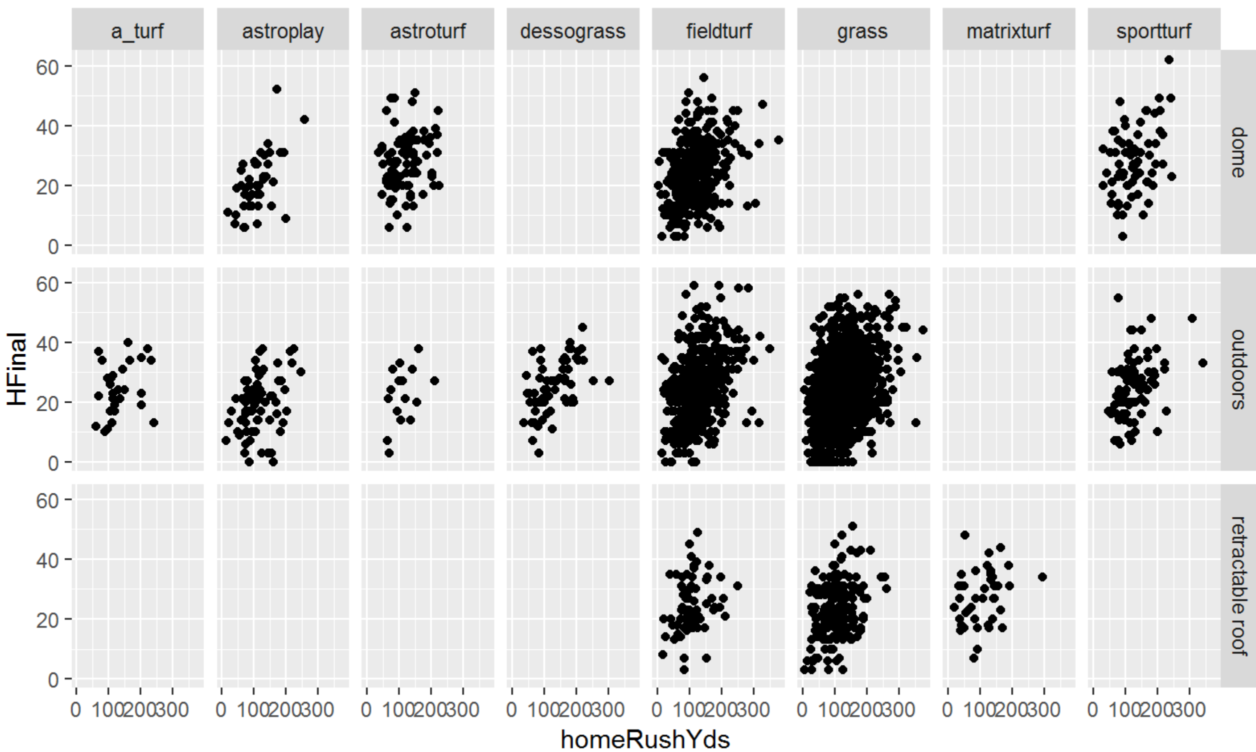
Quantitative Data - Two Variables: Scatter Plot

- Look at graphs broken down by other variables
- Use `facet_grid()`

```
g + geom_point() +
    facet_grid(roof ~ surface)
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot



Numerical and Graphical Summaries

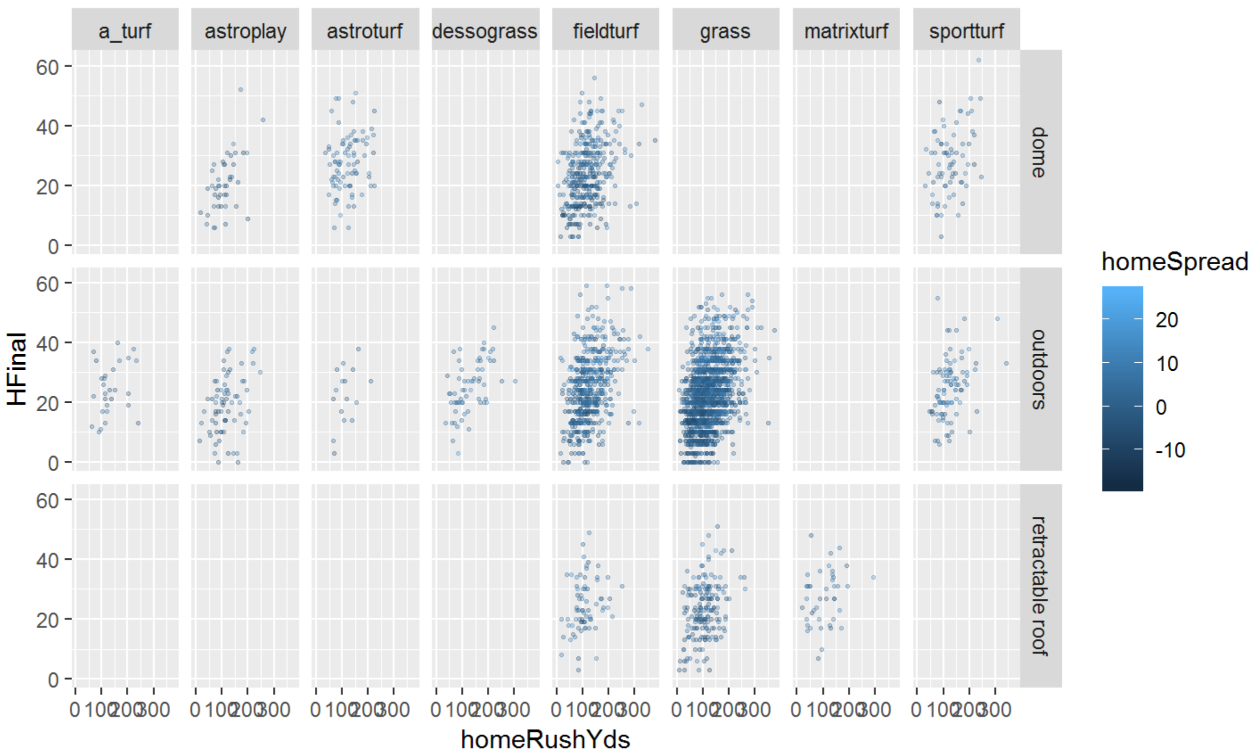
Quantitative Data - Two Variables: Scatter Plot

- Can still do all the fancy stuff to each plot!

```
g + geom_point(aes(col = homeSpread), alpha = 0.3, size = 0.5) +
    facet_grid(roof ~ surface)
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Scatter Plot



Numerical and Graphical Summaries

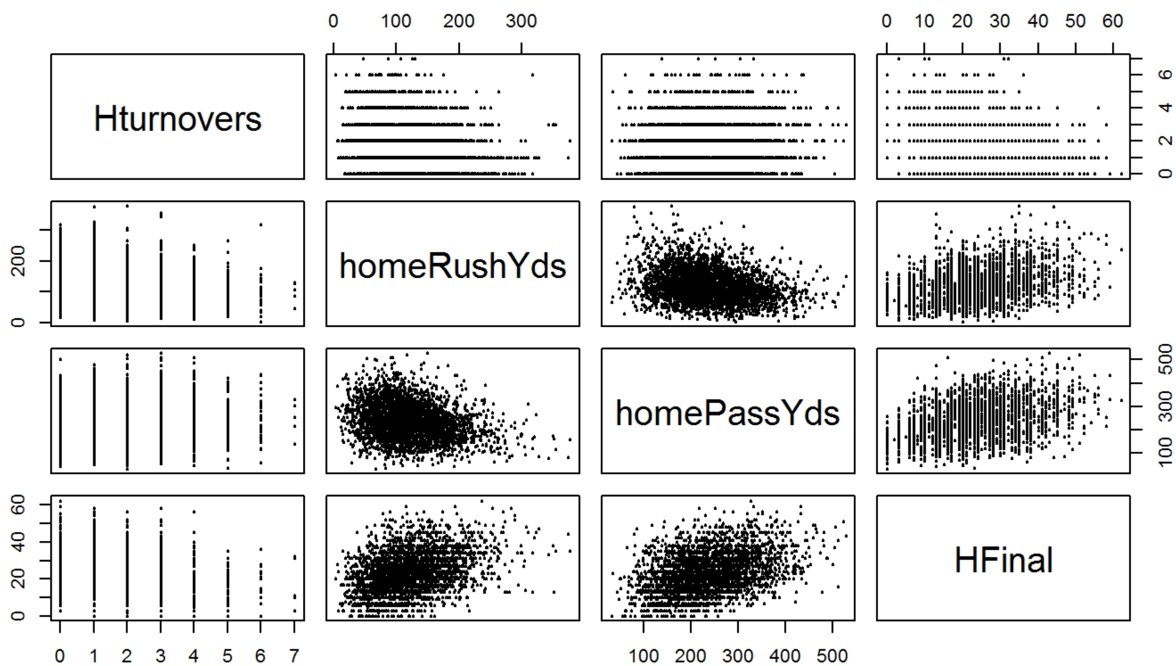
Quantitative Data - Many Variables: Scatter Plot

- Often want to look at relationships of all continuous variables
- `pairs()` gives all pairwise scatter plots

```
pairs(select(scoresFull, Hturnovers, homeRushYds,
             homePassYds, HFinal), cex = 0.3)
```

Numerical and Graphical Summaries

Quantitative Data - Many Variables: Scatter Plot



Numerical and Graphical Summaries

Quantitative Data - Many Variables: Correlation

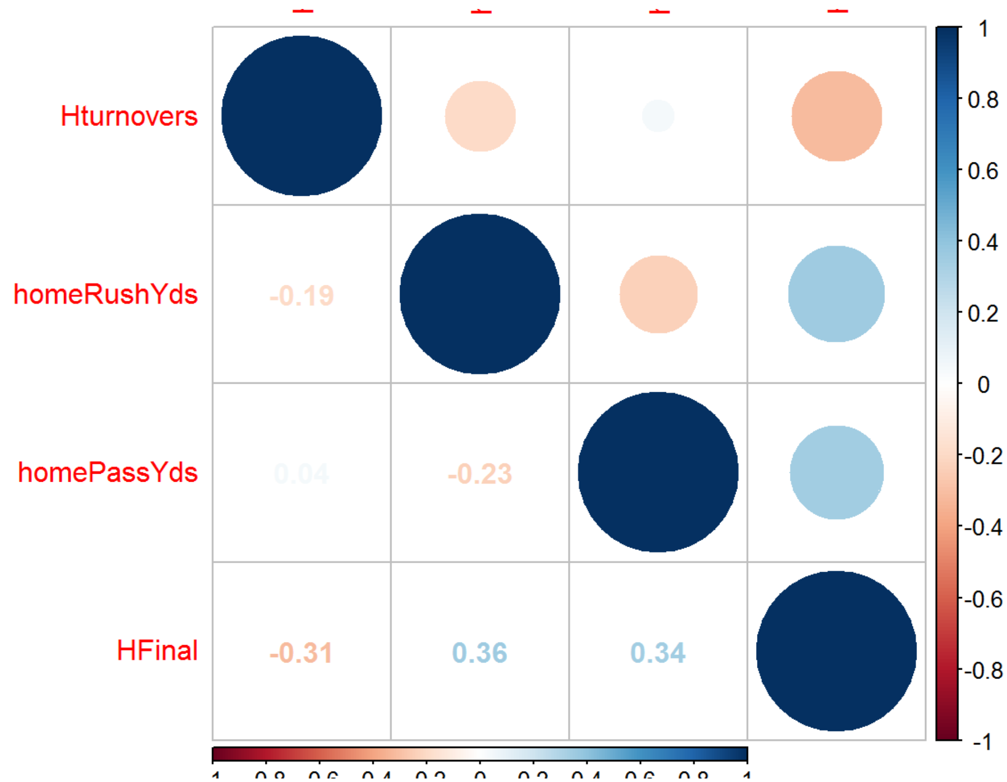
- Create a visual of the correlations

```
Correlation <- cor(select(scoresFull, Hturnovers, homeRushYds,
                        homePassYds, HFinal), method = "spearman")
#install corrplot library
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(Correlation, type = "upper",
         tl.pos = "lt")
corrplot(Correlation, type = "lower", method = "number",
         add = TRUE, diag = FALSE, tl.pos = "n")
```

Numerical and Graphical Summaries

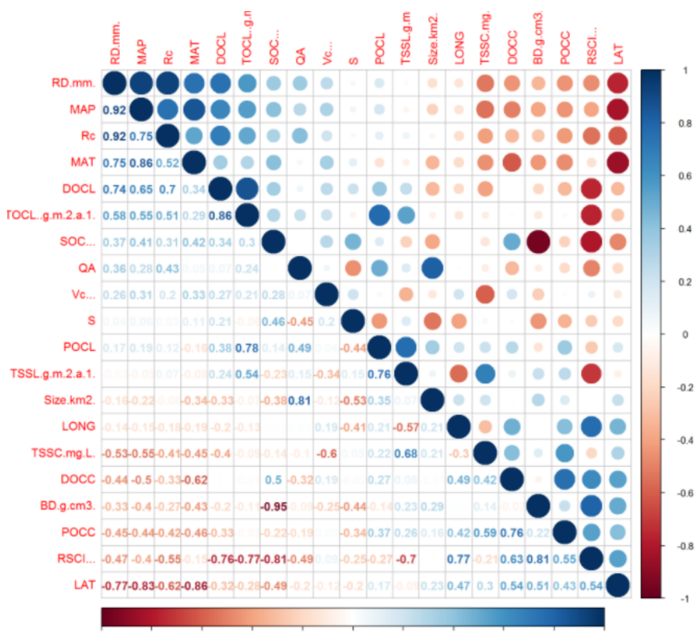


93/150

Numerical and Graphical Summaries

Quantitative Data - Many Variables: Correlation

- Really nice for many variables



Numerical and Graphical Summaries

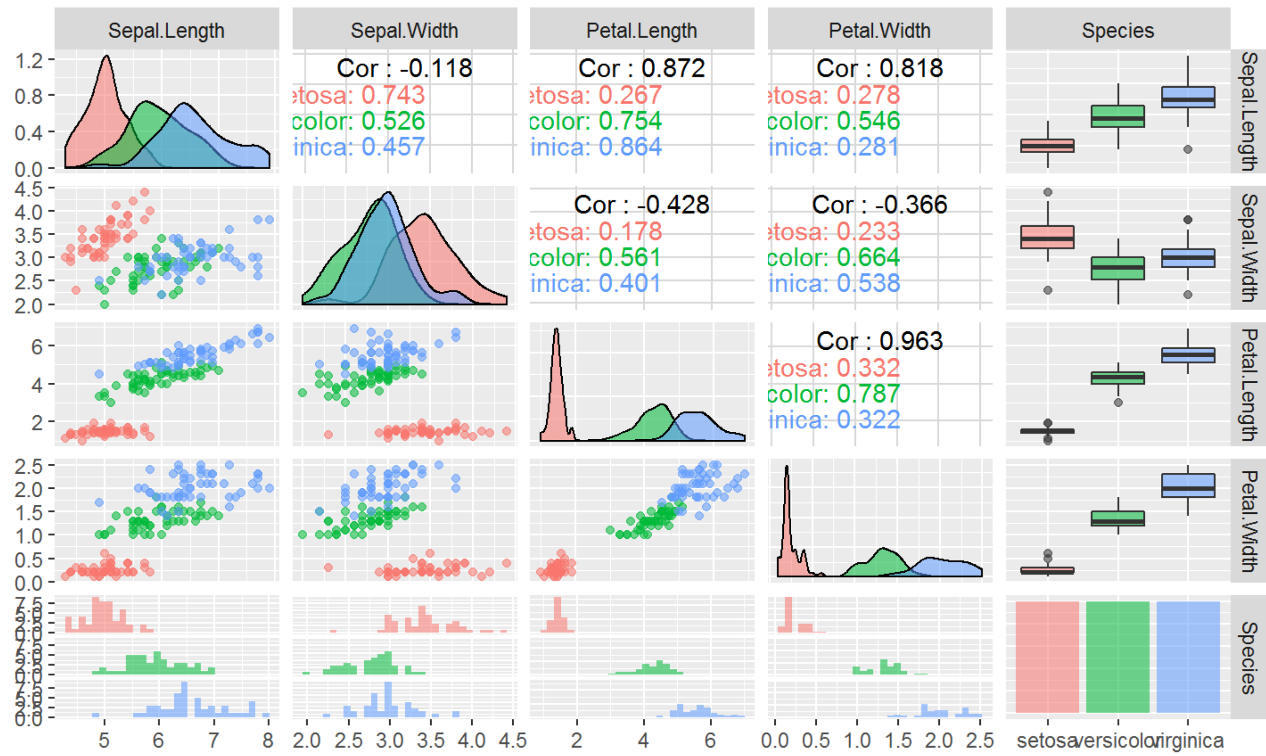
Quantitative Data - Many variables

- Even better, use `ggpairs()` from `GGally` package
- Allows for auto creation with both quantitative and categorical data

```
#install GGally
library(GGally)
```

```
ggpairs(iris, aes(colour = Species, alpha = 0.4))
```

Numerical and Graphical Summaries



Numerical and Graphical Summaries

Quantitative Data - Two Variables: Box/Violin Plot

- Boxplots provide the five number summary in a graph
  - min, Q1, median, Q3, max
  - Often show possible outliers as well

- Violin plots are fancy versions
  - Rotated kernel densities for sides

97/150

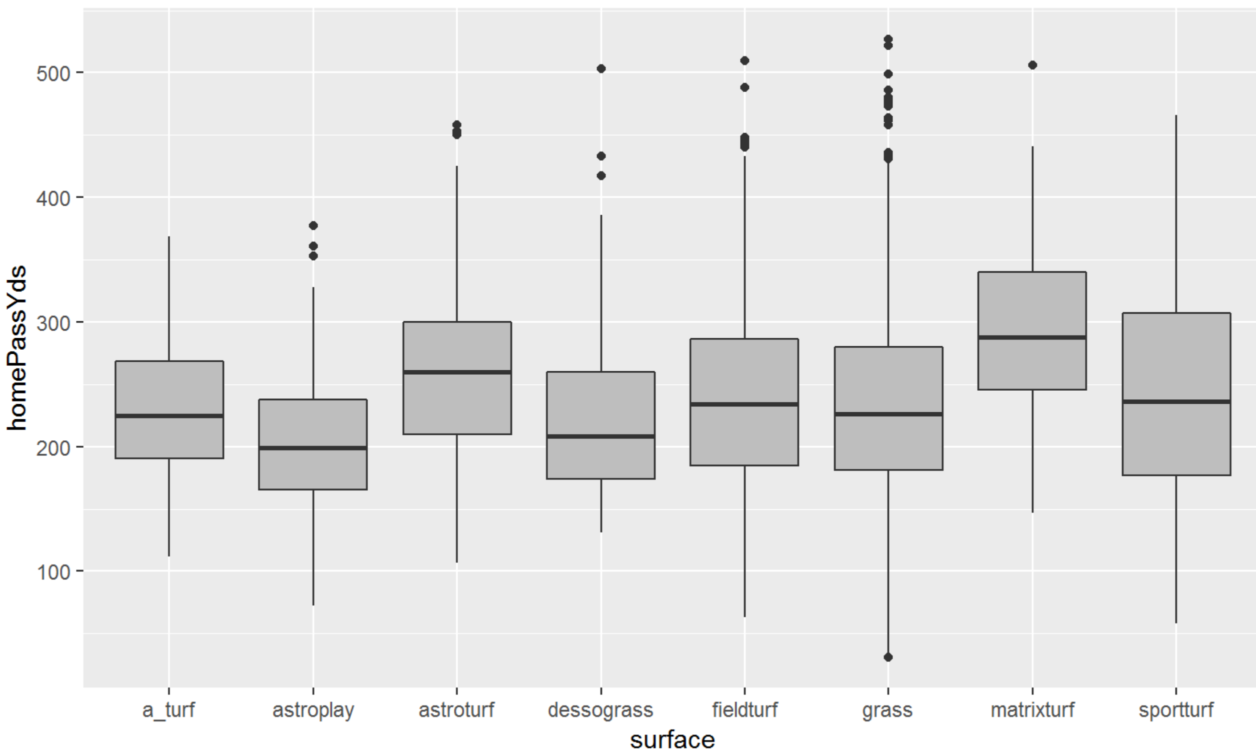
Numerical and Graphical Summaries

Quantitative Data - Two Variables: Box/Violin Plot

```
g <- ggplot(scoresFull, aes(x = surface, y = homePassYds))
g + geom_boxplot(fill = "grey")
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Box/Violin Plot



Numerical and Graphical Summaries

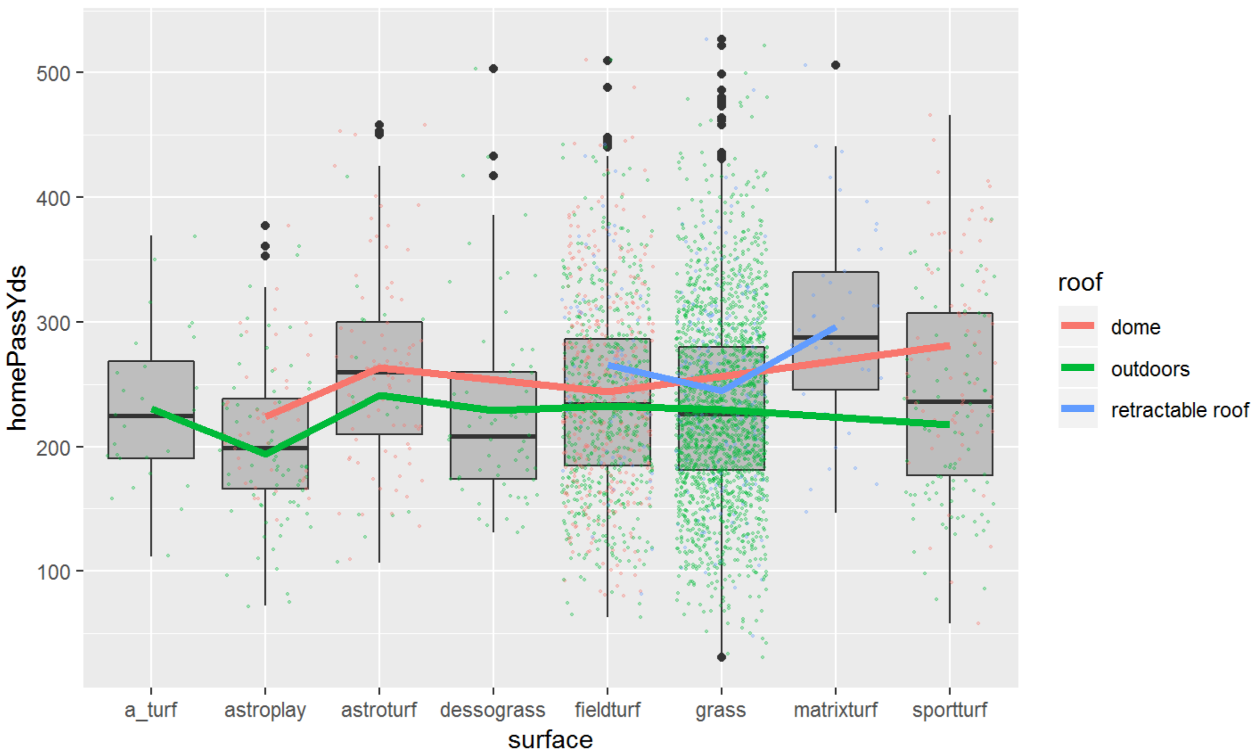
Quantitative Data - Two Variables: Box/Violin Plot

- Get very fancy
- Add in points, lines connecting means for each group

```
g + geom_boxplot(fill = "grey") +
    geom_jitter(aes(col = roof), alpha = 0.3, size = 0.3) +
    stat_summary(fun.y = mean, geom = "line",
                 lwd = 1.5, aes(group = roof, col = roof))
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Box/Violin Plot



101/150

Numerical and Graphical Summaries

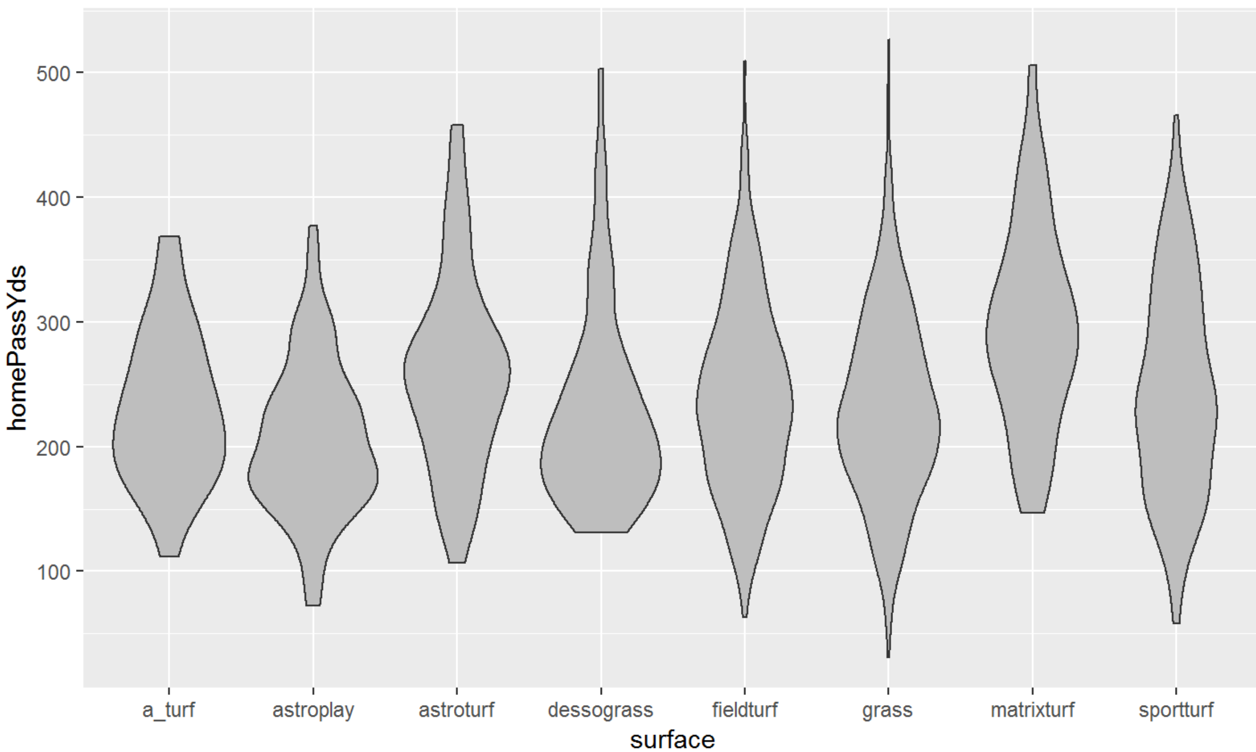
Quantitative Data - Two Variables: Box/Violin Plot

- Check how violin plots look

```
g + geom_violin(fill = "grey")
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Box/Violin Plot



103/150

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Line Plot

- Connects dots
- Most useful with a time type variable

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Line Plot

- Connects dots
- Most useful with a time type variable
- We'll just use season, but…
- Could combine date and season

```
oneDate<-paste(scoresFull$date[1], scoresFull$season[1], sep = "-")
oneDate
```

```
## [1] "5-Sep-2002"
```

Numerical and Graphical Summaries

Aside: Working with Dates

- `lubridate` package great for dates! (part of `tidyverse`)
- Dates can be added and subtracted
- Are # of days since Jan 1, 1970

```
library(lubridate)
as.Date(oneDate,"%d-%b-%Y")
```

```
## [1] "2002-09-05"
```

```
as.Date(oneDate,"%d-%b-%Y") + 1
```

```
## [1] "2002-09-06"
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Line Plot

- Create full date variable for fun
- Summarise total yards for AFC North team home games for each year

```
scoresFull$date <- paste(scoresFull$date, scoresFull$season, sep = "-") %>%
                   as.Date("%d-%b-%Y")

subScores <- scoresFull %>%
  filter(homeTeam %in% c("Pittsburgh Steelers", "Cleveland Browns",
                         "Baltimore Ravens", "Cincinnati Bengals")) %>%
  group_by(season, homeTeam) %>%
  summarise(homeAvgYds = mean(homePassYds + homeRushYds))
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Line Plot

```
subScores
```

```
## # A tibble: 52 x 3
## # Groups:   season [?]
##   season homeTeam            homeAvgYds
##    <int> <chr>                    <dbl>
## 1   2002 Baltimore Ravens          294
## 2   2002 Cincinnati Bengals        326
## 3   2002 Cleveland Browns          325.
## 4   2002 Pittsburgh Steelers       408.
## 5   2003 Baltimore Ravens          333.
## # ... with 47 more rows
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Line Plot

```
g <- ggplot(subScores, aes(x = season, y = homeAvgYds, color = homeTeam))
g + geom_line(lwd = 2)
```

Numerical and Graphical Summaries

Quantitative Data - Two Variables: Line Plot

110/150

Numerical and Graphical Summaries

Quantitative Data - Three Variables: 3D Scatter Plot

- A few packages can do this:
  - `scatterplot3d`
  - `plotly`
  - `plot3Drgl`

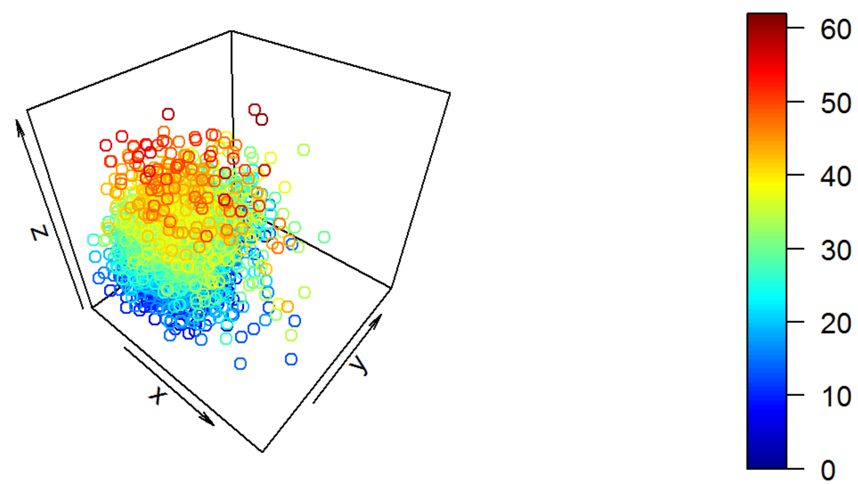
```
install.packages("plot3Drgl")
```

```
library(plot3Drgl)
```

111/150

Numerical and Graphical Summaries

```
scatter3D(x = scoresFull$homeRushYds, y = scoresFull$awayRushYds,
          z = scoresFull$HFinal)
```



Numerical and Graphical Summaries

Quantitative Data - Three Variables: 3D Scatter Plot

- Run the previous `scatter3D` code in your console
- Then run `plotrgl()`

Recap!

- How to summarize quantitative data?
- Numerically?
  - Many summary stats. Focus often on
    - Center, Spread, Linear Relationship
- Graphically?
  - One variable: Kernel Density/Histogram
  - Two variables: Scatter plots
- Important to view your data together **and** broken down into subsets!

Activity

- **Quantitative Plots Activity** instructions available on web
- Work in small groups
- Ask questions! TAs and I will float about the room
- Feel free to ask questions about anything you didn't understand as well!

What do we want to be able to do?

- Read in data
- Manipulate data
- Plot data
- Summarize data
- **Analyze data**

Schedule

Day 2

- Logical Statements and Subsetting/Manipulating Data?
- Numerical and Graphical Summaries
- **Basic Analyses**

Basic Analysis (Linear Regression)

- With multiple quantitative varibles can look at many types of relationships
- Investigated linear relationship between two with correlation
- Scatter Plot gives useful visualization
- Consider data on voting preferences by county
  - Inspect relationship with pct with college degree and income

Basic Analysis (Linear Regression)

- With multiple quantitative varibles can look at many types of relationships
- Investigated linear relationship between two with correlation
- Scatter Plot gives useful visualization
- Consider data on voting preferences by county
  - Inspect relationship with pct with college degree and income

```
voting <- tbl_df(read.csv("datasets/counties.csv", header = TRUE))
```

Basic Analysis (Linear Regression)

```
voting
```

```
## # A tibble: 3,141 x 20
##   region county state   msa  pmsa pop.density   pop pop.change age6574
##   <fct>  <fct>  <fct> <int> <int>       <int> <int>      <dbl>   <dbl>
## 1 South  Autau~ AL     5240    NA          61 34222      11.9     5.70
## 2 South  Baldw~ AL     5160    NA          67 98280      35.4     9.20
## 3 South  Barbo~ AL       NA    NA          29 25417       2       8.20
## 4 South  Bibb   AL       NA    NA          28 16576       9.20    6.70
## 5 South  Blount AL     1000    NA          62 39248      10.6     7.40
## # ... with 3,136 more rows, and 11 more variables: age75 <dbl>,
## #   crime <int>, college <dbl>, income <int>, farm <dbl>, democrat <dbl>,
## #   republican <dbl>, Perot <dbl>, white <dbl>, black <dbl>, turnout <dbl>
```

120/150

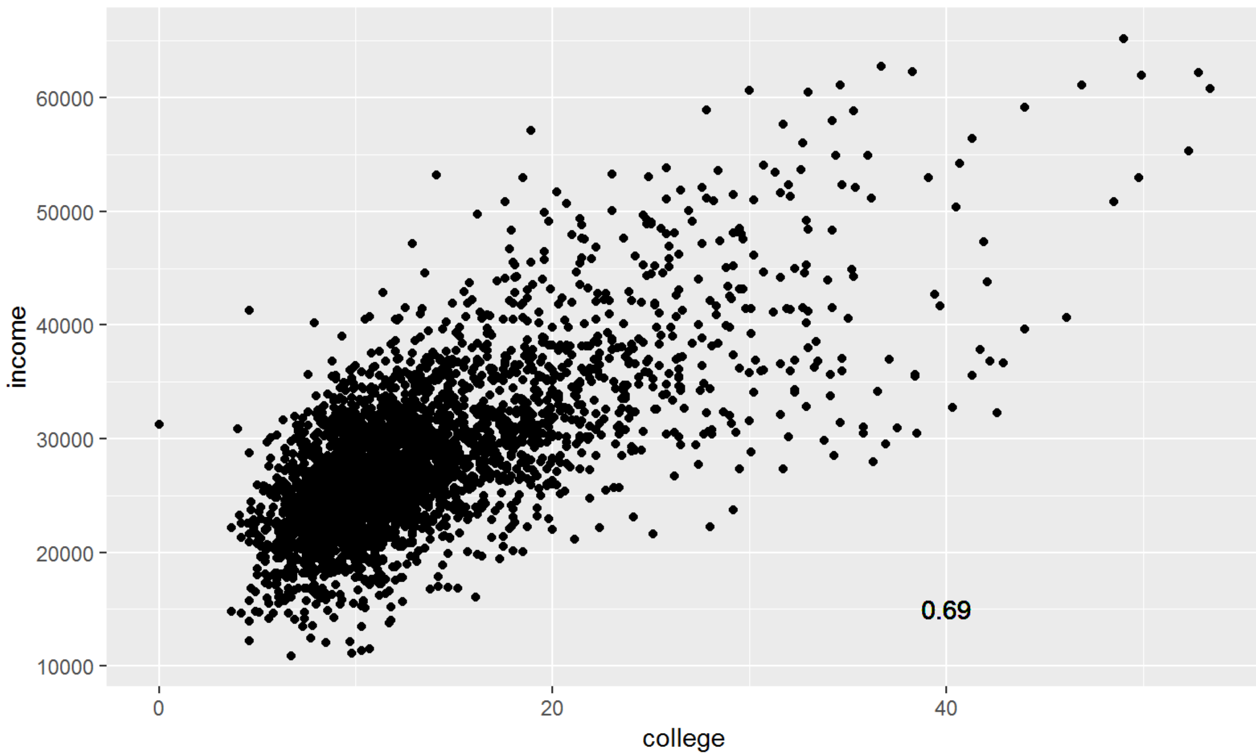
Basic Analysis (Linear Regression)

- Create a scatter plot, add correlation

```
votePlot <- ggplot(voting, aes(x = college, y = income))
```

```
votePlot +
  geom_point()+
  geom_text(x = 40, y = 15000,
            label = round(cor(voting$college, voting$income), 2))
```

Basic Analysis (Linear Regression)



Basic Analysis (Linear Regression)

- Often want to predict y variable using a value of x
- **Simple Linear Regression** allows for this
- Linear model for predicting income based on college:

cty_income = int + slope * (% college in county) + random_error

- Generally, response (Y) = function of predictors (features, X's) + error

$$Y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + E_i$$

123/150

Basic Analysis (Linear Regression)

- Fit simple linear regression model using `lm()`

```
lm(income ~ college, data = voting)
```

```
##
## Call:
## lm(formula = income ~ college, data = voting)
##
## Coefficients:
## (Intercept)      college
##     18305.5        752.6
```

Basic Analysis (Linear Regression)

- Save as object, check attributes

```
fit <- lm(income ~ college, data = voting)
attributes(fit)
```

```
## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"
```

125/150

Basic Analysis (Linear Regression)

- Three main ways to get at *many* of these attributes

```
fit[[1]]
```

```
## (Intercept)     college
##  18305.5294    752.6481
```

```
coefficients(fit)
```

```
## (Intercept)     college
##  18305.5294    752.6481
```

```
fit$coefficients
```

```
## (Intercept)     college
##  18305.5294    752.6481
```

Basic Analysis (Linear Regression)

- Three main ways to get at *many* of these attributes

```
fit[[2]]
residuals(fit)
fit$residuals
```

Basic Analysis (Linear Regression)

- Three main ways to get at *many* of these attributes

```
#no generic function for some things
rank(fit)
```

```
## Error in rank(fit): unimplemented type 'list' in 'greater'
```

```
y(fit)
```

```
## Error in y(fit): could not find function "y"
```

```
fit$rank
```

```
## [1] 2
```

128/150

Basic Analysis (Linear Regression)

- Statistical analysis found using `anova()` or `summary()`

```
#ANOVA table (F tests)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: income
##             Df     Sum Sq    Mean Sq F value    Pr(>F)
## college      1 7.6911e+10 7.6911e+10  2865.4 < 2.2e-16 ***
## Residuals 3139 8.4254e+10 2.6841e+07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Basic Analysis (Linear Regression)

- Statistical analysis found using `anova()` or `summary()`

```
#coefficient type III tests
summary(fit)
```

```
##
## Call:
## lm(formula = income ~ college, data = voting)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18062.3  -3146.8   -288.8   2890.7  24569.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18305.53     211.29   86.64   <2e-16 ***
## college       752.65      14.06   53.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5181 on 3139 degrees of freedom
## Multiple R-squared:  0.4772, Adjusted R-squared:  0.4771
## F-statistic:  2865 on 1 and 3139 DF,  p-value: < 2.2e-16
```

Basic Analysis (Linear Regression)

- Diagnostic plots easily found using `plot()`
- Run this code in console

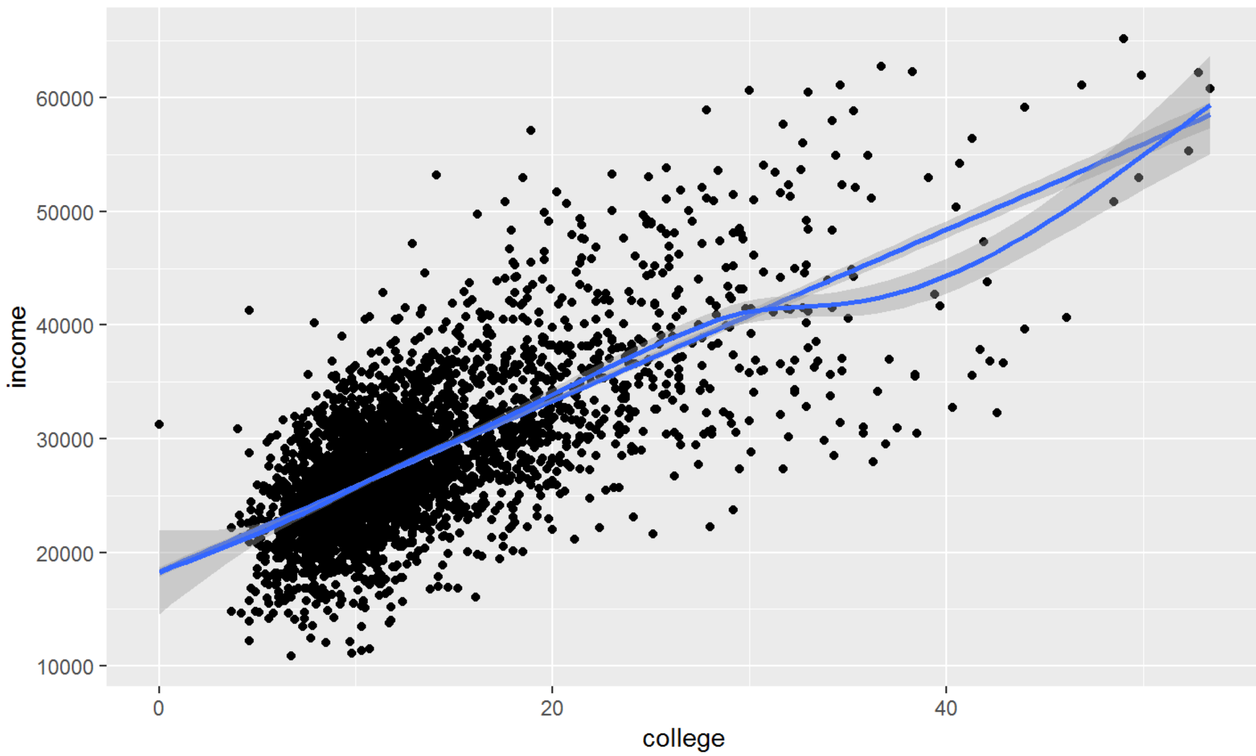
```
plot(fit)
```

Basic Analysis (Linear Regression)

- Add fit to our scatter plot
- Compare to a smoothed fit

```
votePlot +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_smooth()
```

Basic Analysis (Linear Regression)



133/150

Basic Analysis (Linear Regression)

- Now can predict y for a given x
- Check `help(predict)`, particularly `predict.lm()`

```
predict(fit, newdata = data.frame(college = c(40, 10)))
```

```
##        1        2
## 48411.45 25832.01
```

Basic Analysis (Linear Regression)

- Get SE for prediction

```
predict(fit, newdata = data.frame(college = c(40, 10)), se.fit = TRUE)
```

```
## $fit
##        1        2
## 48411.45 25832.01
##
## $se.fit
##        1        2
## 383.7192 104.8104
##
## $df
## [1] 3139
##
## $residual.scale
## [1] 5180.841
```

135/150

Basic Analysis (Linear Regression)

- Get confidence interval for mean response

```
predict(fit, newdata = data.frame(college = c(40, 10)),
        se.fit = TRUE, interval = "confidence")
```

```
## $fit
##        fit      lwr      upr
## 1 48411.45 47659.09 49163.82
## 2 25832.01 25626.51 26037.51
##
## $se.fit
##        1        2
## 383.7192 104.8104
##
## $df
## [1] 3139
##
## $residual.scale
## [1] 5180.841
```

Basic Analysis (Linear Regression)

- Get prediction interval for new response

```
predict(fit, newdata = data.frame(college = c(40, 10)),
        se.fit = TRUE, interval = "prediction")
```

```
## $fit
##        fit      lwr      upr
## 1 48411.45 38225.45 58597.45
## 2 25832.01 15671.75 35992.27
##
## $se.fit
##        1        2
## 383.7192 104.8104
##
## $df
## [1] 3139
##
## $residual.scale
## [1] 5180.841
```

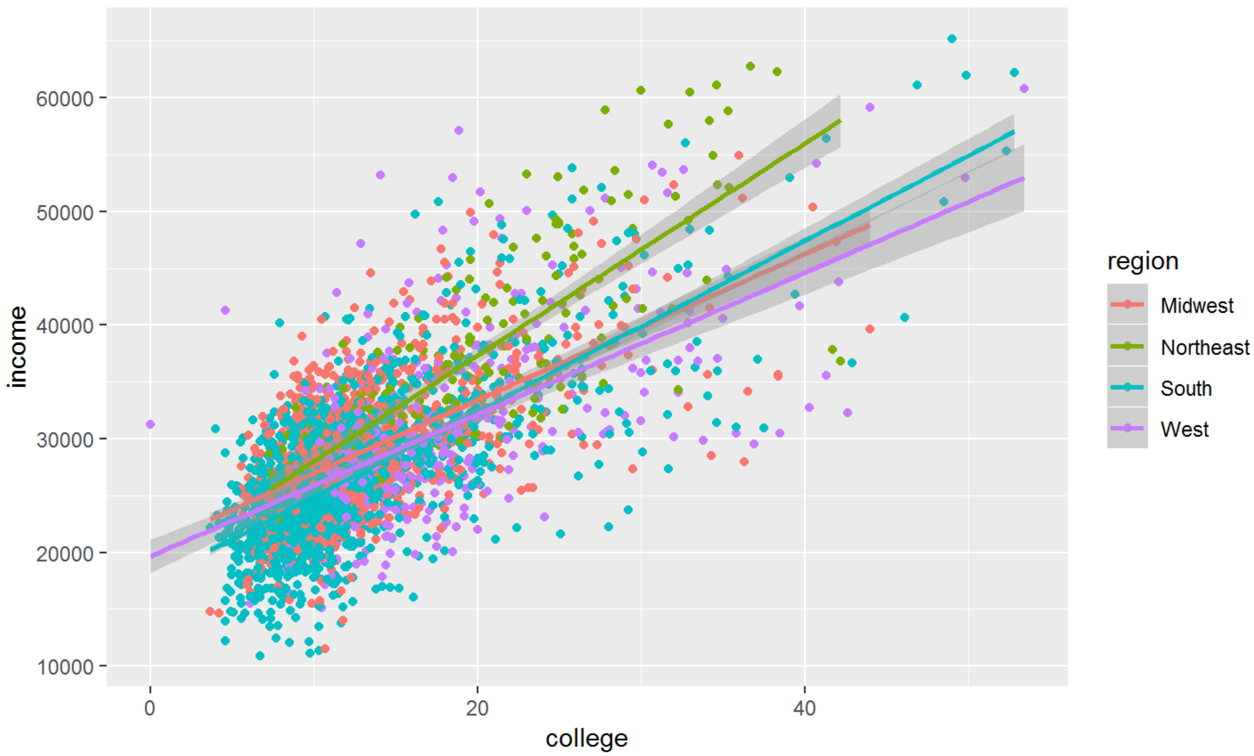
137/150

Basic Analysis (Linear Regression)

- May want separate fits by a variable
- Do separate SLR for each region setting

```
votePlot +
  geom_point(aes(col = region)) +
  geom_smooth(method = "lm", aes(col = region))
```

Basic Analysis (Linear Regression)



139/150

Basic Analysis (Linear Regression)

- Obtain fits for each grouping

```
fits <- voting %>% group_by(region) %>%
  do(model = lm(income ~ college, data = .))
names(fits)
```

```
## [1] "region" "model"
```

- Probably easier to just subset and then fit each model separately!

Basic Analysis (Linear Regression)

```
fits$model[[1]]
```

```
##
## Call:
## lm(formula = income ~ college, data = .)
##
## Coefficients:
## (Intercept)      college
##     20566.1        642.2
```

```
fits$model[[2]]
```

```
##
## Call:
## lm(formula = income ~ college, data = .)
##
## Coefficients:
## (Intercept)      college
##     18702.1        932.1
```

Basic Analysis (Linear Regression)

- Multiple Linear Regression (more than one x)
- In `lm()`
  - add "main effect" terms with `+ name`
  - interactions with `+ name1:name2`
  - all possible combinations with `+ name1*name2`

```
fit2<-lm(income ~ college + Perot, data = voting)
```

142/150

Basic Analysis (Linear Regression)

- Multiple Linear Regression (more than one x)

```
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: income
##             Df     Sum Sq    Mean Sq  F value    Pr(>F)
## college      1 7.4771e+10 7.4771e+10 3013.718 < 2.2e-16 ***
## Perot        1 2.0739e+09 2.0739e+09   83.591 < 2.2e-16 ***
## Residuals 3111 7.7185e+10 2.4810e+07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

143/150

Basic Analysis (Linear Regression)

```
summary(fit2)
```

```
##
## Call:
## lm(formula = income ~ college + Perot, data = voting)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -17192  -3205   -234   2909  21077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16167.90     310.46  52.077   <2e-16 ***
## college       728.46      13.71  53.136   <2e-16 ***
## Perot         119.73      13.10   9.143   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4981 on 3111 degrees of freedom
##   (27 observations deleted due to missingness)
## Multiple R-squared:  0.4989, Adjusted R-squared:  0.4986
## F-statistic:  1549 on 2 and 3111 DF,  p-value: < 2.2e-16
```

Basic Analysis (Linear Regression)

- Access elements of object just as before

```
coef(fit2)
```

```
## (Intercept)     college       Perot
##  16167.9032    728.4587    119.7262
```

```
fit2$rank
```

```
## [1] 3
```

Basic Analysis (Linear Regression)

- Basic diagnostic plots (run in console)

```
plot(fit2)
```

Basic Analysis (Linear Regression)

- Predict new for new values

```
predict(fit2, newdata = data.frame(college = 40, Perot = 20))
```

```
##        1
## 47700.77
```

Recap!

- Basic linear regression
  - Use `lm()`
- Statistical analysis
  - `anova()`
  - `summary()`
- Predict using `predict()`
- Add fits to scatter plot with `geom_smooth(method = "lm")`

What do we want to be able to do?

- Read in data
- Manipulate data
- Plot data
- Summarize data
- Analyze data

Activity

- **Basic Analysis Activity** instructions available on web
- Work in small groups
- Ask questions! TAs and I will float about the room
- Feel free to ask questions about anything you didn't understand as well!
- Thanks for coming!!