COMMERCE DATA SERVICE

Intro to Data Analysis with R

Pri Oberoi

5/16/2016

While you are waiting, go here to get started:

www.github.com/prioberoi/R_intro_to_data_analysis

COMMERCE DATA SERVICE

Pri Oberoi (poberoi@doc.gov)
Data Scientist, Commerce Data Service
US Department of Commerce

COMMERCE DATA SERVICE

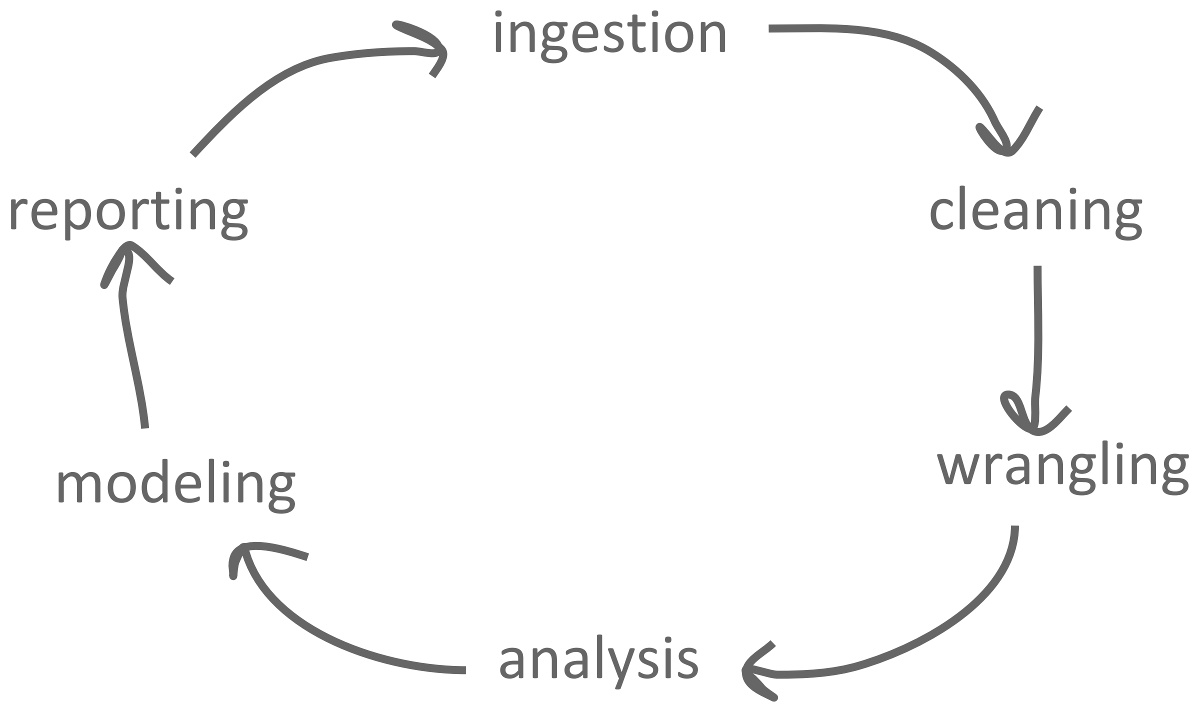
Goals

Our goals for the class

- Import, clean, visualize data in R
- Role of data analysis in the data science pipeline
- Use data analysis to summarize,
-

COMMERCE DATA SERVICE

The Data Science Pipeline



COMMERCE DATA SERVICE

Why do data analysis?

COMMERCE
DATA SERVICE

clean

wrangle

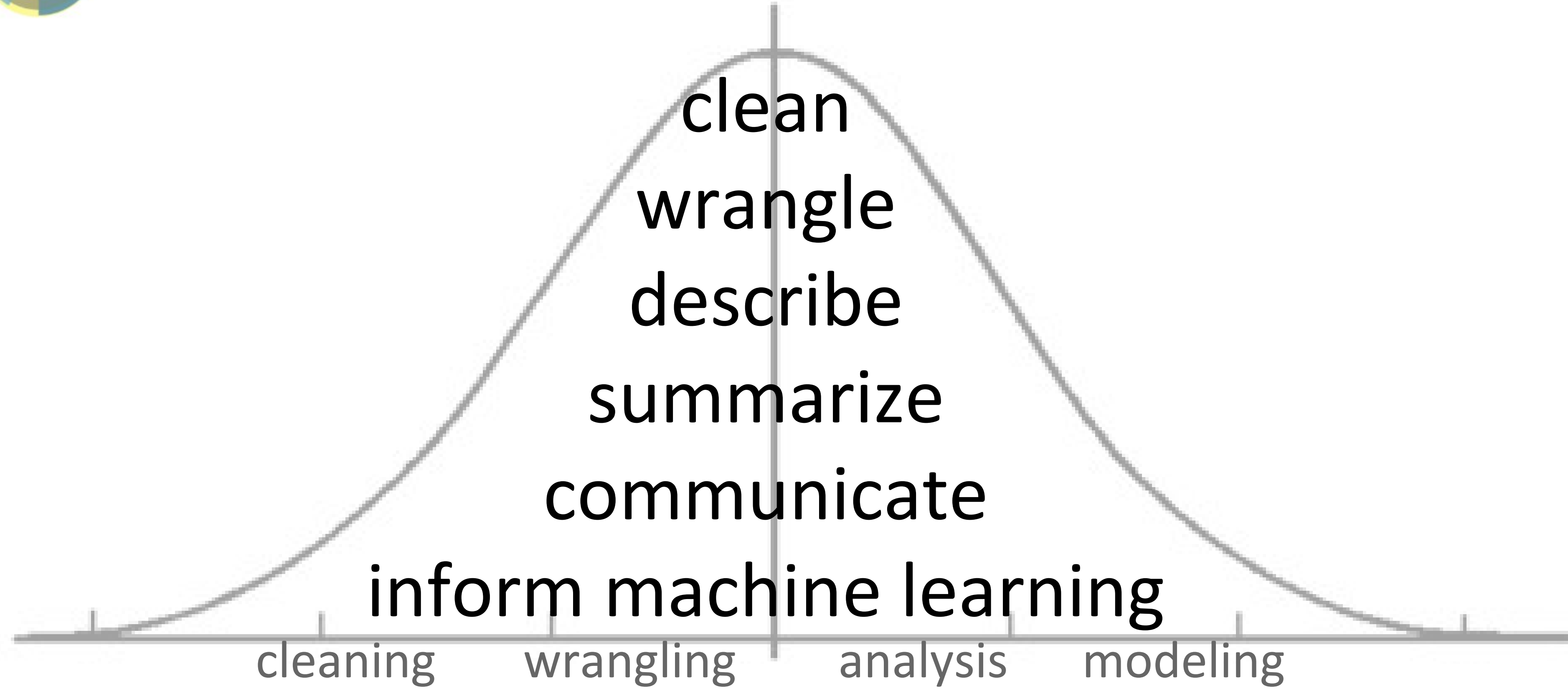
describe

summarize

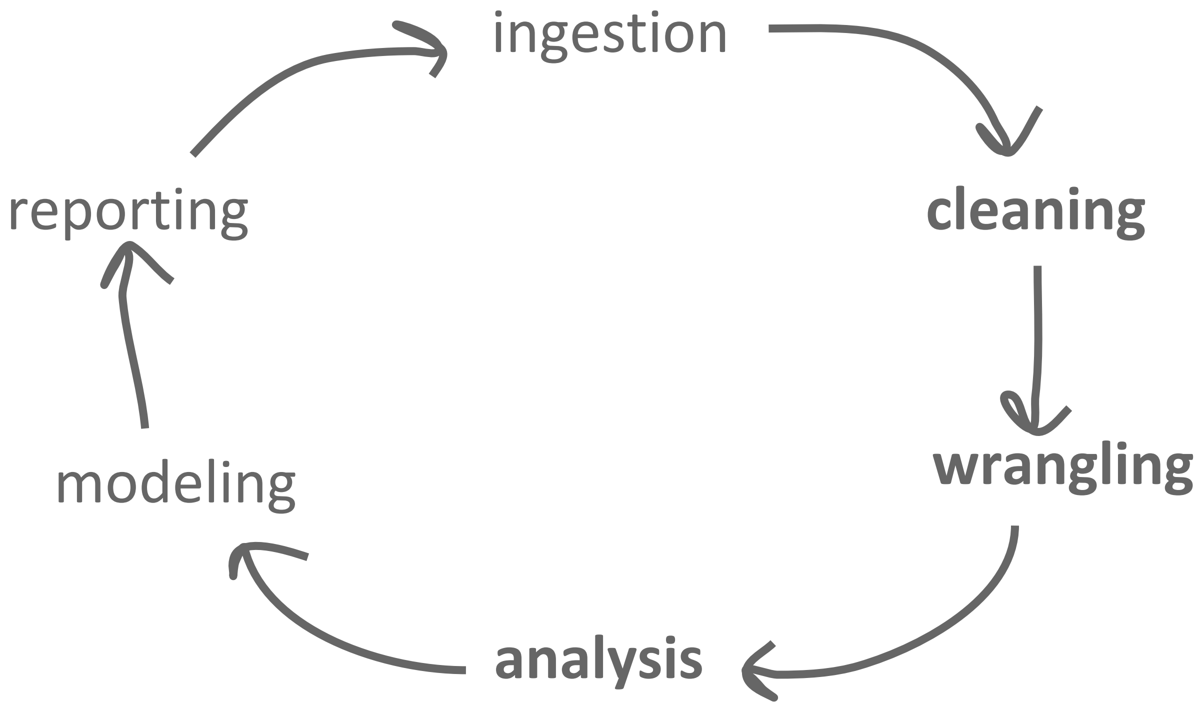
communicate

inform machine learning

clean
wrangle
describe
summarize
communicate
inform machine learning



The Pipeline



R Markdown

COMMERCE
DATA SERVICE

Output formats:
HTML, PDF, MS Word, HTML5 slides,
books, dashboards, websites

Benefits:
Easy to create
Embedded R code chunks
(which can be visible or not on the final output)
Allows you to add a narrative through your code
Reproducible

COMMERCE
DATA SERVICE

PUT SCREENSHOT OF R MARKDOWN FOR CLASS HERE.
ANNOTATE.

COMMERCE DATA SERVICE

Analysis Toolkit

Visualization

Statistics

Aggregation

COMMERCE DATA SERVICE

Data Visualization

COMMERCE
DATA SERVICE

ggplot2

qplot()
"quick plot"

- similar to plot() from base R
- less typing
- less customizable

```
qplot(carat, price, data = diamonds,
size = I(1), alpha = I(1/10), main =
"qplot scatter plot")
```

ggplot()

- more customizable
- more functionality

```
ggplot(data = diamonds, aes(x = carat,
y = price)) +
  geom_point(size = 1, alpha = 1/10) +
  ggtitle("ggplot scatter plot")
```

ggplot2

x y

```
qplot(carat, price, data = diamonds,
size = I(1), alpha = I(1/10), main =
"qplot scatter plot")
```

y x

```
ggplot(data = diamonds, aes(x = carat,
y = price)) +
  geom_point(size = 1, alpha = 1/10) +
  ggtitle("ggplot scatter plot")
```

aesthetics like point size, point transparence, plot title

data

Your turn

Run the code in chunk 2: **Scatterplots**

Update ggplot() code so the color of the scatterplot points varies based on the value of ‘cut’

You can do this by adding a ‘colour =’ argument to the aes() mapping

COMMERCE
DATA SERVICE

Your turn

Run the code in chunk 3: **Histograms and Bar Charts**
Note that you can set the ‘binwidth’ for histograms

Run the code in chunk 4: **Boxplots and Violin Plots**
Box plots: more widely interpretable
Violin plots: useful for non-normal distributions and to scale to number of observations

COMMERCE
DATA SERVICE

NTIA Broadband Data Example

NTIA’s broadband data from June, 2014 for Washington, DC

COMMERCE DATA SERVICE

Hypothesis Testing

Null hypothesis: the typical upload and download speeds for broadband providers in Washington, DC are the same as the advertised speeds

COMMERCE DATA SERVICE

Your turn

Import the data by running chunk 5

Look at the dataframe

```
View(data)
dim(data)
names(data)
str(data)
summary(data)
```

Create a histogram of max advertised download speeds (maxaddown)

COMMERCE DATA SERVICE

10 min break

COMMERCE DATA SERVICE

Cleaning

Messy Data

This content is from Hadley Wickham's paper on tidy data

Signs you have messy data:

- Column headers are values, not variable names
- Multiple variables are stored in one column
- Variables are stored in both rows and columns
- Multiple types of observational units are stored in the same table
- A single observational unit is stored in multiple tables

Column headers are values, not variable names

We will be looking at the maxaddown, maxadup, typicdown, typicup variables

They are stored as different columns/variables, rather than different values.

```
# Run chunk 6

melt() # this is a function that converts columns into rows
?melt # use ?melt to read the documentation on this function
```

Multiple variables are stored in one column

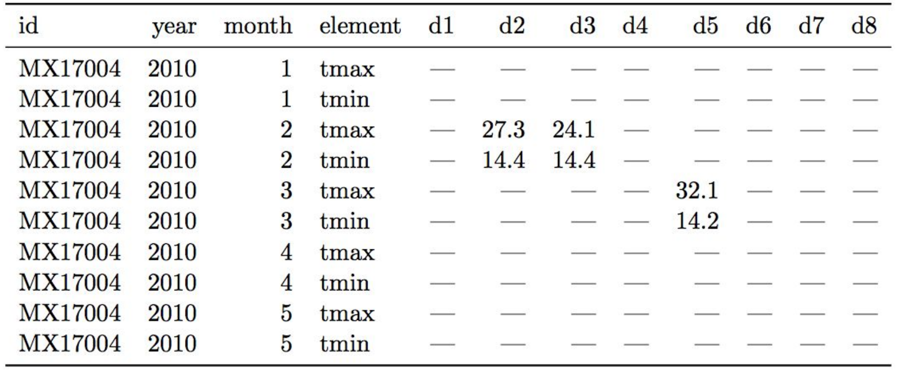
data_clean now has one column named ‘variable’ that contains the variable indicating if this is advertised or typical as well as whether this speed is for uploads or downloads.

```
# Run chunk 7 and look at the resulting data_clean dataframe

data_clean$speedDirection <- "download"
data_clean$speedDirection[data_clean$variable %in% c("maxadup","typicup")]
<- "upload"
data_clean$speedSource <- "advertised"
data_clean$speedSource[data_clean$variable %in% c("typicup","typicdown")]
<- "typical"
data_clean$variable <- NULL
```

Variables are stored in both rows and columns

We don’t have this problem in our dataset, but here is an example:



id	year	month	element	d1	d2	d3	d4	d5	d6	d7	d8
MX17004	2010	1	tmax	—	—	—	—	—	—	—	—
MX17004	2010	1	tmin	—	—	—	—	—	—	—	—
MX17004	2010	2	tmax	—	27.3	24.1	—	—	—	—	—
MX17004	2010	2	tmin	—	14.4	14.4	—	—	—	—	—
MX17004	2010	3	tmax	—	—	—	—	32.1	—	—	—
MX17004	2010	3	tmin	—	—	—	—	14.2	—	—	—
MX17004	2010	4	tmax	—	—	—	—	—	—	—	—
MX17004	2010	4	tmin	—	—	—	—	—	—	—	—
MX17004	2010	5	tmax	—	—	—	—	—	—	—	—
MX17004	2010	5	tmin	—	—	—	—	—	—	—	—

Table 11: Original weather dataset. There is a column for each possible day in the month. Columns `d9` to `d31` have been omitted to conserve space.

Variables are individual columns (id, year, month), spread across columns (day, d1–d31) and across rows (tmin, tmax)

Multiple types of observational units are stored in the same table

data_clean has data on different observational units, the provider/holding company, broadband speeds, location

Repeating values in a column are a result.

Similar to database normalization.

Note, some data analysis tools work with denormalized data

A single observational unit is stored in multiple tables

Data values for a single variable are found across tables

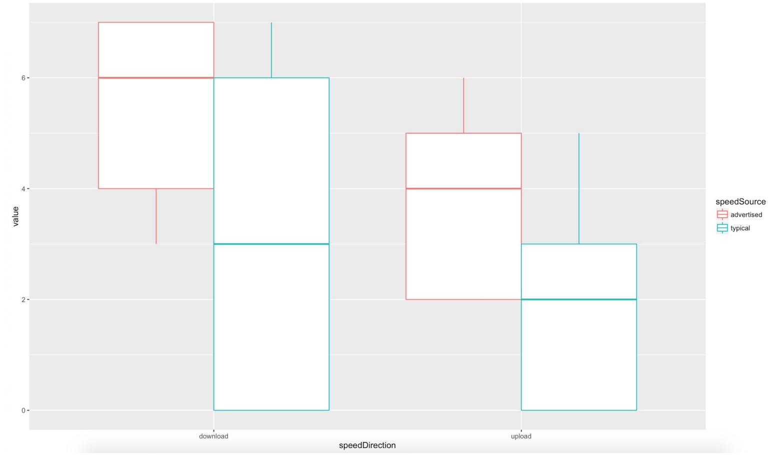
```
# the following functions are helpful
rbind() #add dataframe as rows, must have same number of columns
cbind() #add dataframe as columns, must have same number of rows
merge() #merge two data frames by common columns or row names
ldply() #from the plyr package reads multiple csvs into one dataframe
```

COMMERCE DATA SERVICE

Summarizing Data Within Groups

Summarizing Data Within Groups

This boxplot indicates advertised and typical speeds differ.



```
# aggregate() will run functions over a group you specify

# what does this do:
# data_clean[data_clean$speedDirection == 'download','value']

aggregate(data_clean[data_clean$speedDirection == 'download','value'], list
(data_clean[data_clean$speedDirection == 'download', 'speedSource']), mean)
```

Your turn

Update chunk 8 to show the mean upload speeds by speedSource (advertised or typical)

COMMERCE
DATA SERVICE

10 min break

COMMERCE
DATA SERVICE

Correlation

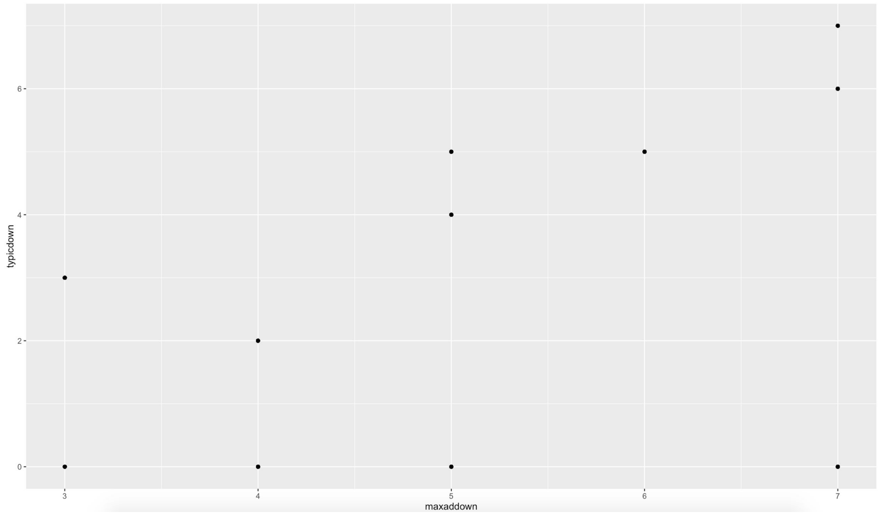
COMMERCE DATA SERVICE

Correlation

We expect that the advertised and typical speeds are correlated

Run chunk 9

Correlation



```
                 Pearson's product-moment correlation

data:  data$maxaddown and data$typicdown
t = 242.74, df = 149810, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5276687 0.5349375
sample estimates:
      cor
0.5313129
```

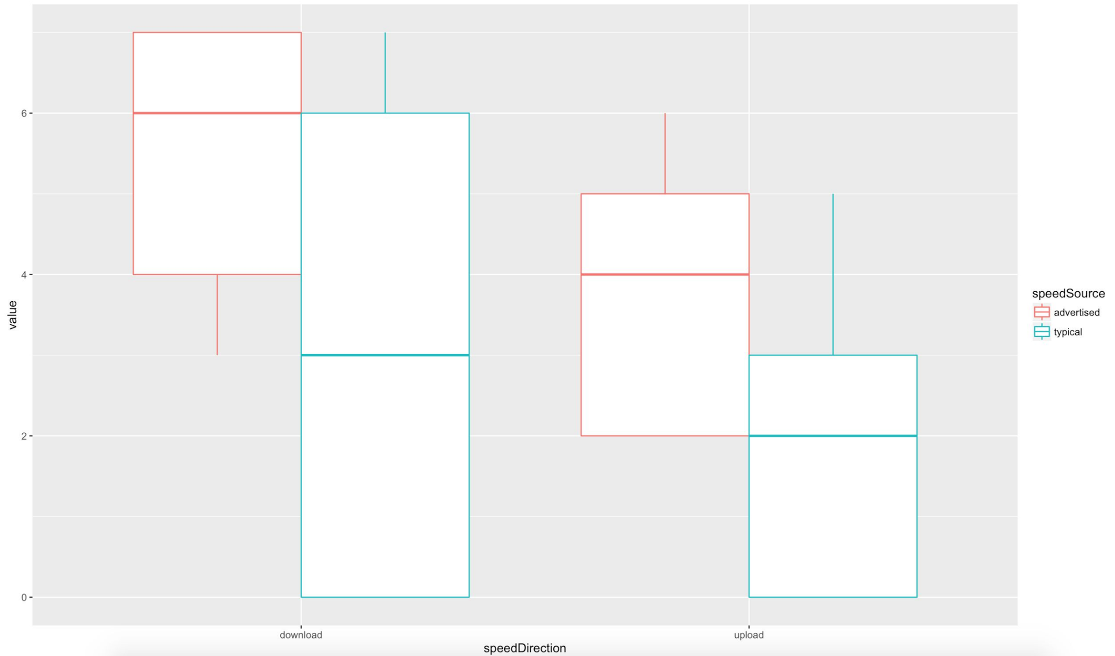
p-value < 0.05
not surprising

Revisit our null hypothesis

Null hypothesis: the typical upload and download speeds for broadband providers in Washington, DC are the same as the advertised speeds

Comparing Samples

Null hypothesis: the typical upload and download speeds for broadband providers in Washington, DC are the same as the advertised speeds



Null hypothesis: the typical download speeds for broadband providers in Washington, DC are the same as the advertised speeds

Let’s do a quick t-test to see if that difference in statistically significant

```
t.test(response ~ variable, data)
```

Null hypothesis: the typical download speeds for broadband providers in Washington, DC are the same as the advertised speeds

Let’s do a quick t-test to see if that difference in statistically significant

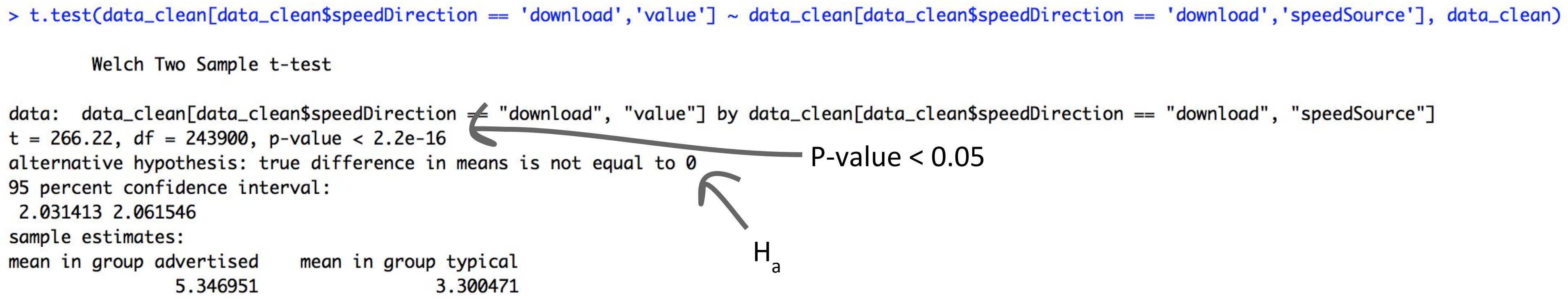
```
t.test(response ~ variable, data)

#does the download speed differ based on whether the speed was
advertised or typical
in our dataset?
```

Null hypothesis: the typical mean upload and download speeds for broadband providers in Washington, DC are the same as the advertised speeds

H_0

Run chunk 10



```
> t.test(data_clean[data_clean$speedDirection == 'download','value'] ~ data_clean[data_clean$speedDirection == 'download','speedSource'], data_clean)

	Welch Two Sample t-test

data:  data_clean[data_clean$speedDirection == "download", "value"] by data_clean[data_clean$speedDirection == "download", "speedSource"]
t = 266.22, df = 243900, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.031413 2.061546
sample estimates:
mean in group advertised    mean in group typical
                5.346951                 3.300471
```

P-value < 0.05

H_a

Your turn!

update the t-test code (chunk 10) to see if the typical upload speed is different than the advertised upload speed

COMMERCE DATA SERVICE

10 min break

COMMERCE DATA SERVICE

From statistical tests to statistical learning

Can we predict what the typical upload and download speeds are if given the advertised upload and download speeds?

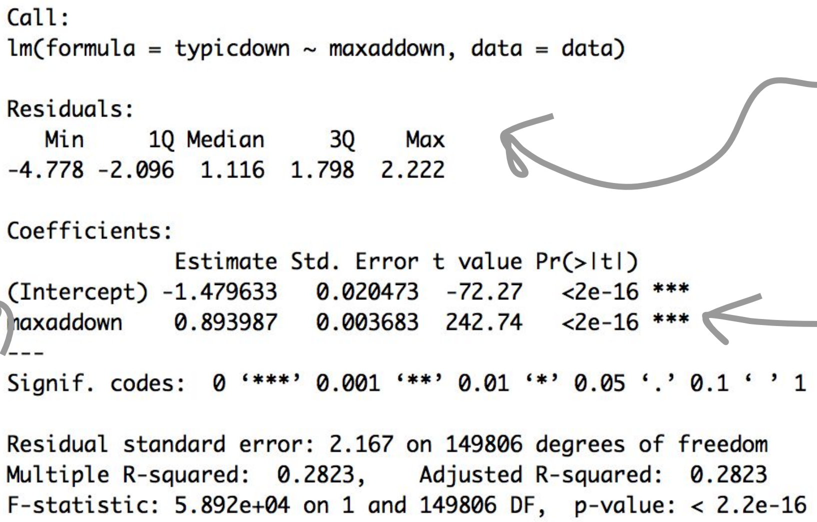
COMMERCE
DATA SERVICE

Linear Regression

```
fit <- lm(response ~ predictors, data)
```

Linear Regression

```
# run chunk 11
fit <- lm(typicdown ~ maxaddown, data)
```



```
Call:
lm(formula = typicdown ~ maxaddown, data = data)

Residuals:
   Min     1Q Median     3Q    Max 
-4.778 -2.096  1.116  1.798  2.222 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.479633   0.020473  -72.27   <2e-16 ***
maxaddown    0.893987   0.003683  242.74   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.167 on 149806 degrees of freedom
Multiple R-squared:  0.2823,	Adjusted R-squared:  0.2823 
F-statistic: 5.892e+04 on 1 and 149806 DF,  p-value: < 2.2e-16
```

residuals: how far off were the predicted values from the observed value?

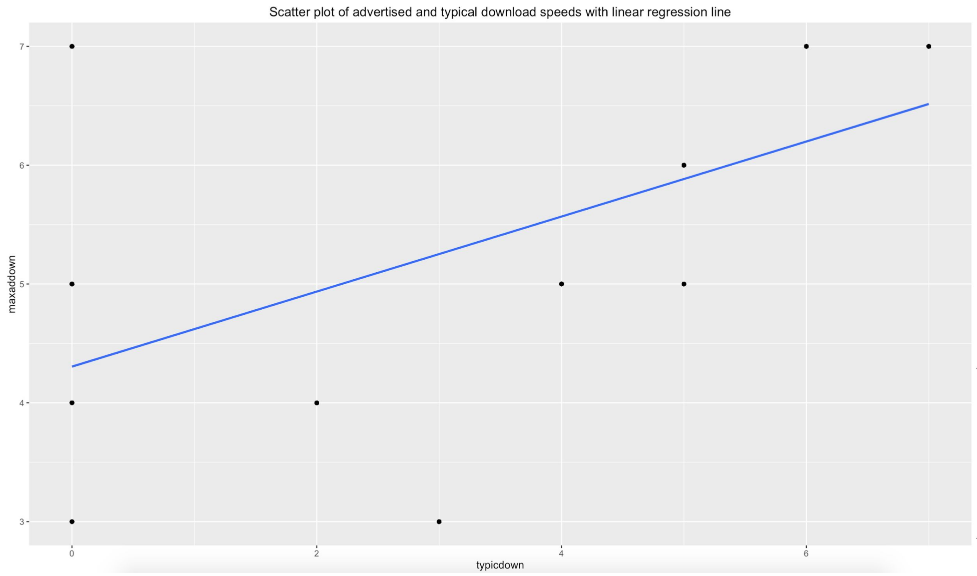
our predictors and their significance

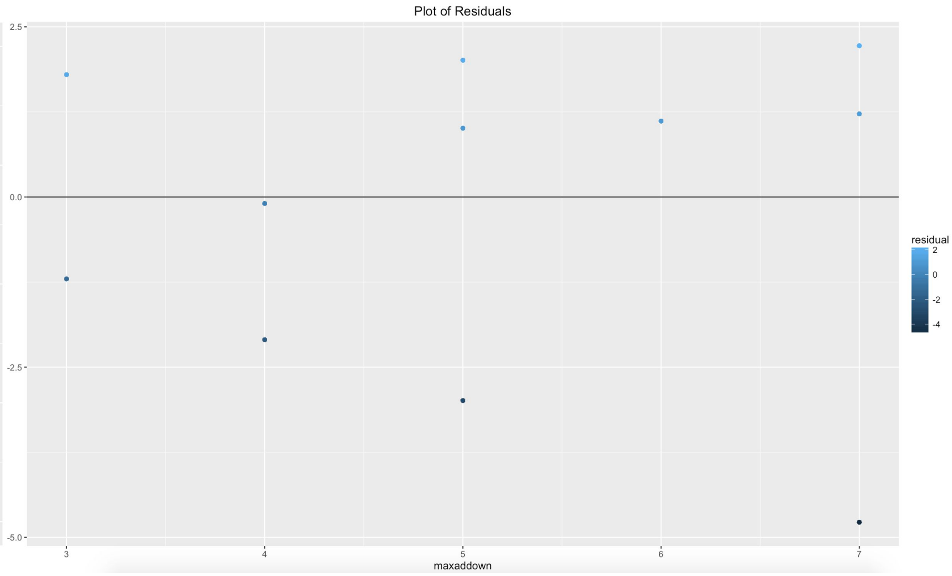
p-value < 0.05

hypothetical typical download speed for an advertised download speed of 0

COMMERCE DATA SERVICE

Linear Regression





COMMERCE
DATA SERVICE

Resources

R-bloggers: http://www.r-bloggers.com/
FlowingData: http://flowingdata.com/category/tutorials/
Google's R Style Guide: https://google.github.io/styleguide/Rguide.xml
R Markdown: http://rmarkdown.rstudio.com/