COMMERCE DATA SERVICE

Intro to Data Analysis with R

Pri Oberoi

5/16/2016

Make sure you have all the content in this github repo:
www.github.com/prioberoi/R_intro_to_data_analysis

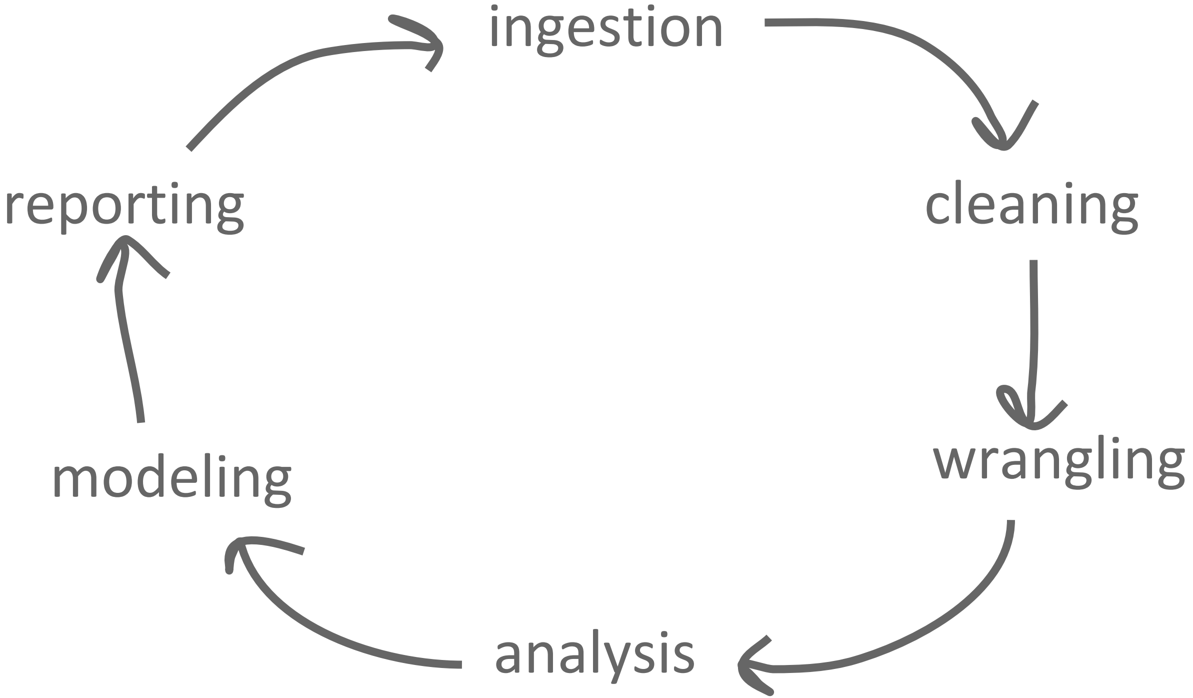
Pri Oberoi (poberoi@doc.gov)
Data Scientist, Commerce Data Service
US Department of Commerce

Goals

Walk away with the foundations for

- The role of data analysis is in the data science pipeline
- R markdown
- Data visualizations
- Clean data
- Aggregate and summarize data
- Statistical tests

The Data Science Pipeline



ingestion

cleaning

wrangling

analysis

modeling

reporting

Why do data analysis?

clean

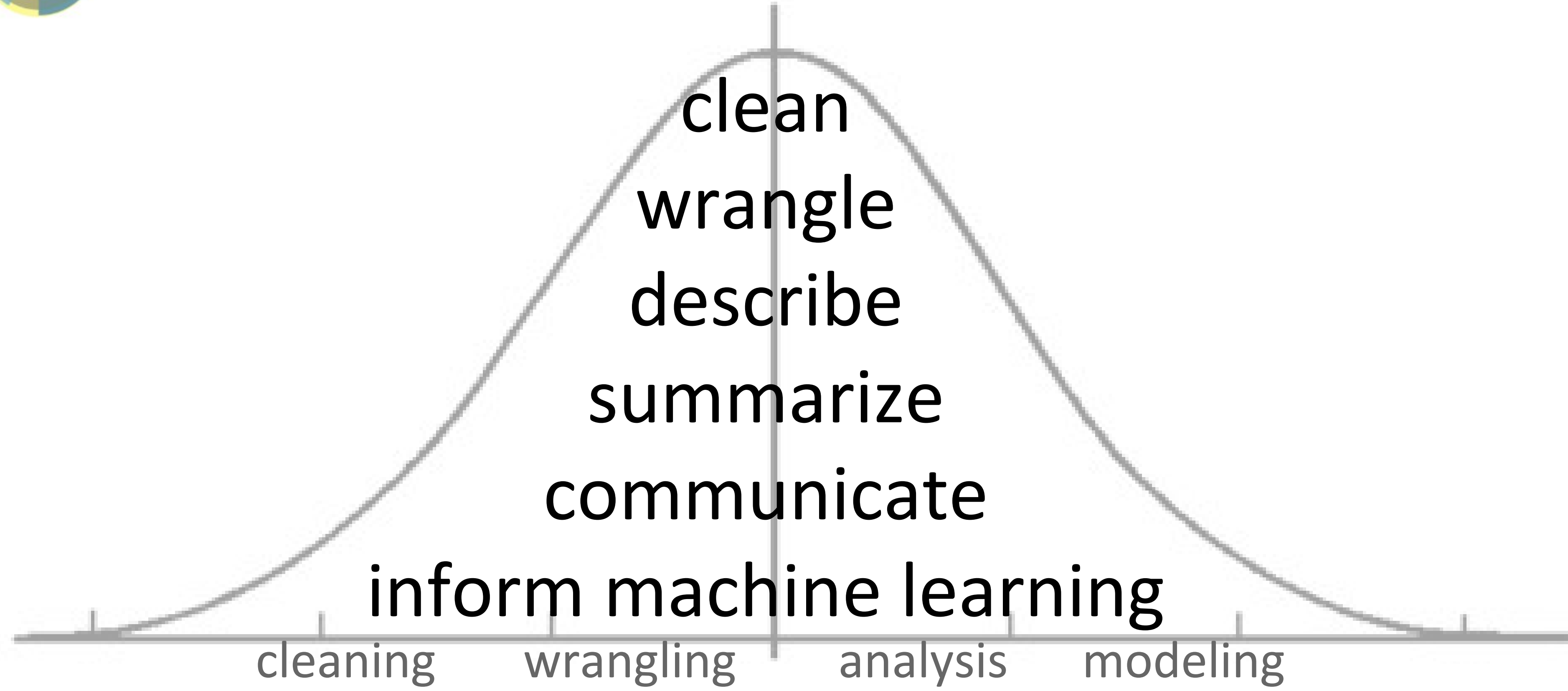
wrangle

describe

summarize

communicate

inform machine learning



clean

wrangle

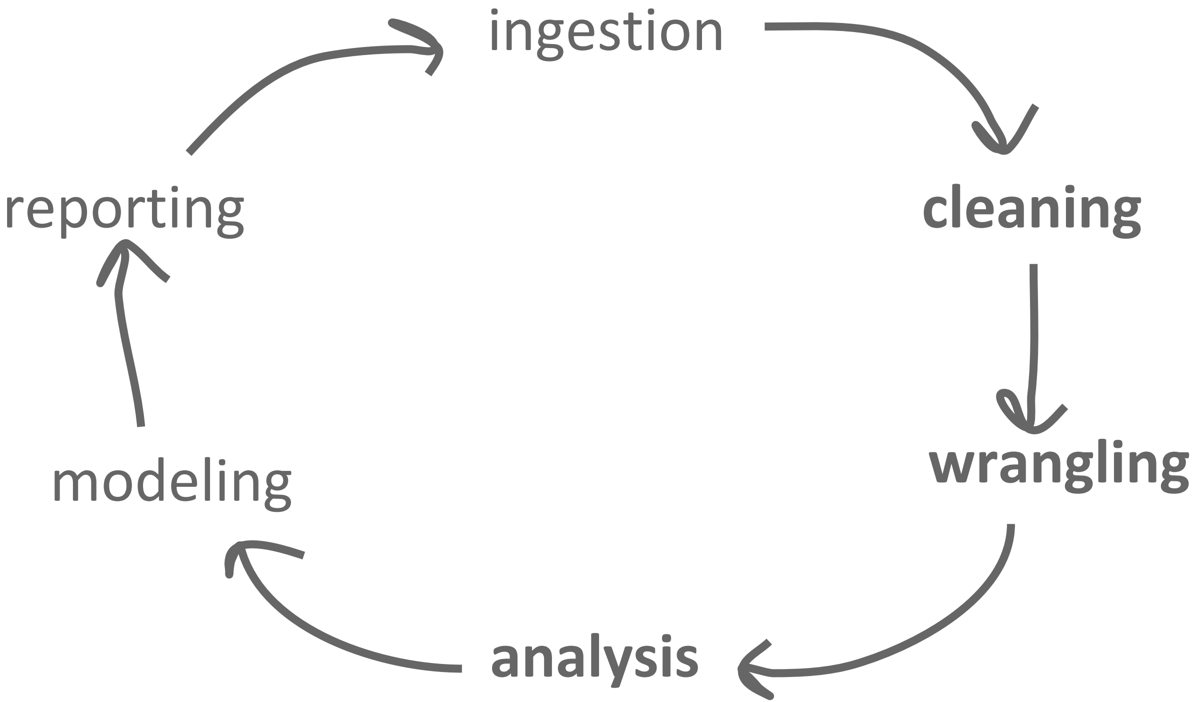
describe

summarize

communicate

inform machine learning

The Pipeline



ingestion

cleaning

wrangling

analysis

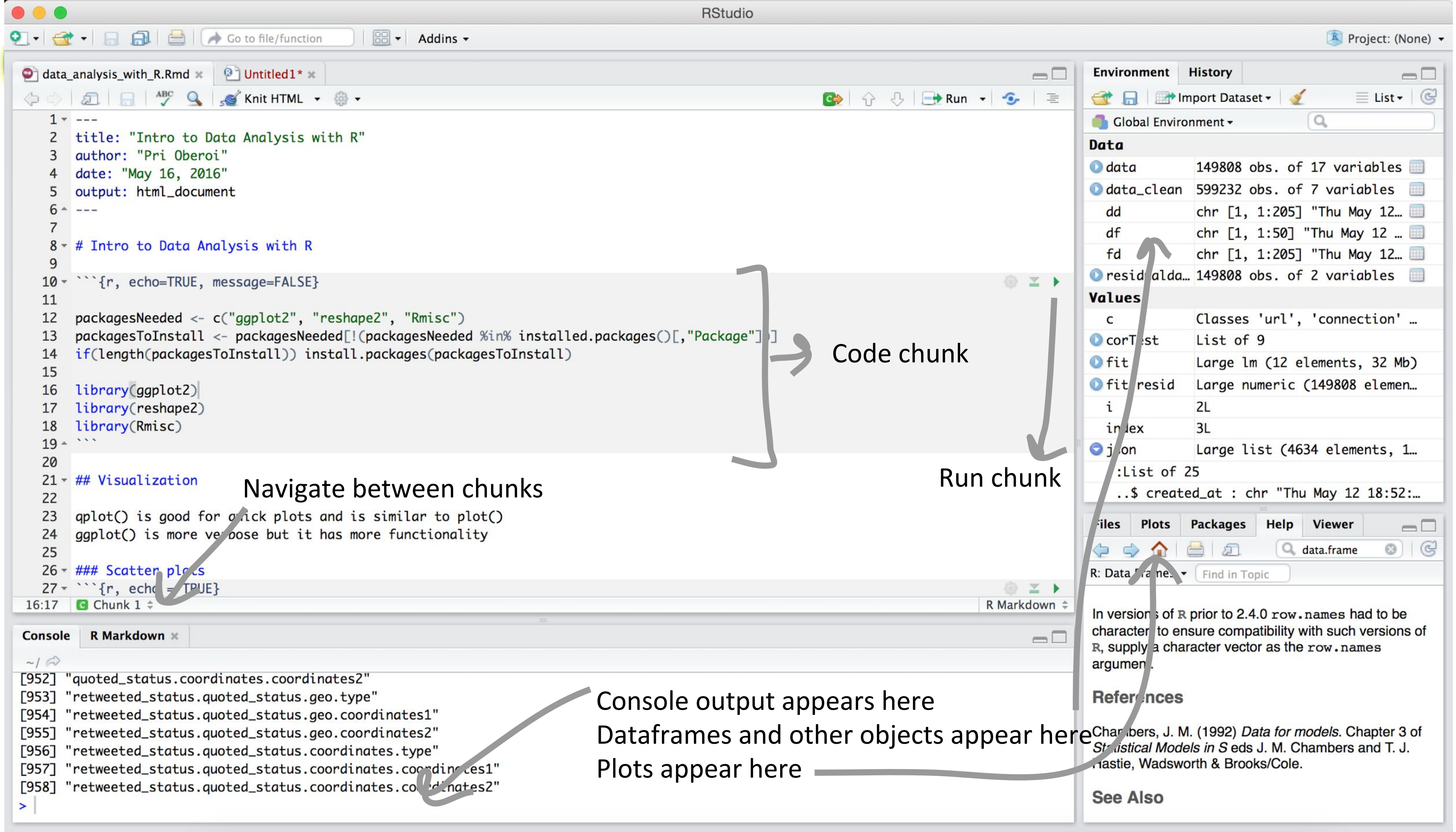
modeling

reporting

R Markdown

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



~~~
---
title: "Intro to Data Analysis with R"
author: "Pri Oberoi"
date: "May 16, 2016"
output: html_document
---

# Intro to Data Analysis with R

```{r, echo=TRUE, message=FALSE}

packagesNeeded <- c("ggplot2", "reshape2", "Rmisc")
packagesToInstall <- packagesNeeded[!(packagesNeeded %in% installed.packages()[,"Package"])]
if(length(packagesToInstall)) install.packages(packagesToInstall)

library(ggplot2)
library(reshape2)
library(Rmisc)
~~~

Visualization

qplot() is good for quick plots and is similar to plot()
ggplot() is more verbose but it has more functionality

Scatter plots
```{r, echo = TRUE}
```

```
Data
data          149808 obs. of 17 variables
data_clean    599232 obs. of 7 variables
dd            chr [1, 1:205] "Thu May 12…
df            chr [1, 1:50] "Thu May 12 …
fd            chr [1, 1:205] "Thu May 12…
residualda…   149808 obs. of 2 variables
Values
c             Classes 'url', 'connection' …
corTest       List of 9
fit           Large lm (12 elements, 32 Mb)
fit_resid     Large numeric (149808 elemen…
i             2L
index         3L
json          Large list (4634 elements, 1…
  :List of 25
  ..$ created_at : chr "Thu May 12 18:52:…
```

In versions of R prior to 2.4.0 row.names had to be character to ensure compatibility with such versions of R, supply a character vector as the row.names argument.

References

Chambers, J. M. (1992) *Data for models.* Chapter 3 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

See Also

```
~/
[952] "quoted_status.coordinates.coordinates2"
[953] "retweeted_status.quoted_status.geo.type"
[954] "retweeted_status.quoted_status.geo.coordinates1"
[955] "retweeted_status.quoted_status.geo.coordinates2"
[956] "retweeted_status.quoted_status.coordinates.type"
[957] "retweeted_status.quoted_status.coordinates.coordinates1"
[958] "retweeted_status.quoted_status.coordinates.coordinates2"
>
```

Code chunk

Run chunk

Navigate between chunks

Console output appears here

Dataframes and other objects appear here

Plots appear here

Analysis Toolkit

Visualization

Statistics

Aggregation

Data Visualization

ggplot2

qplot()
“quick plot”

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

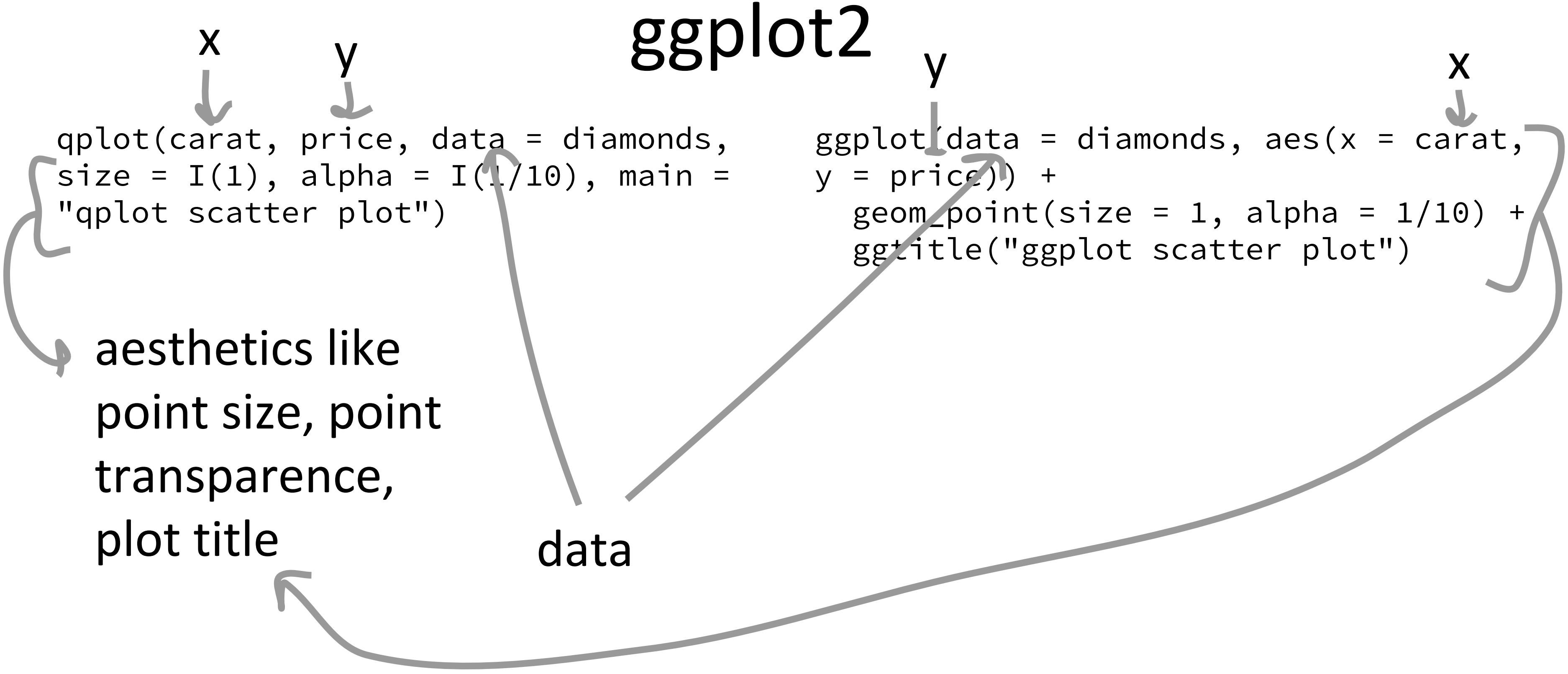
```
qplot(carat, price, data = diamonds,
size = I(1), alpha = I(1/10), main =
"qplot scatter plot")
```

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

COMMERCE DATA SERVICE

Your turn (10 mins)

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

Your turn (10 mins)

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

NTIA Broadband Data Example

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

Hypothesis Testing

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

Your turn (10 mins)

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)

10 min break

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 ? to read the documentation on a function
```

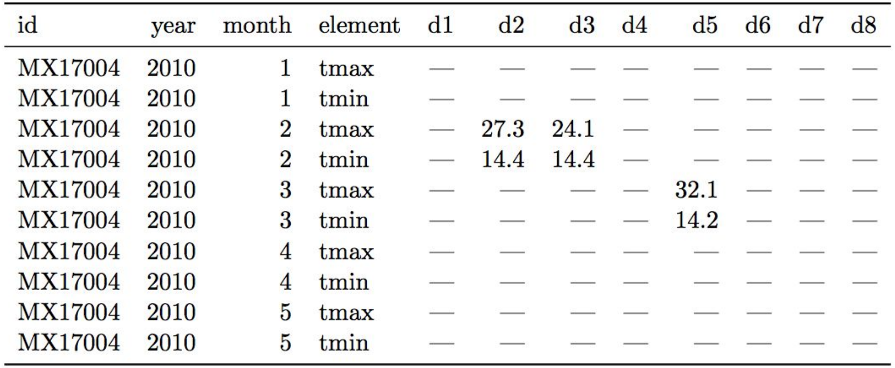
Multiple variables are stored in one column

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

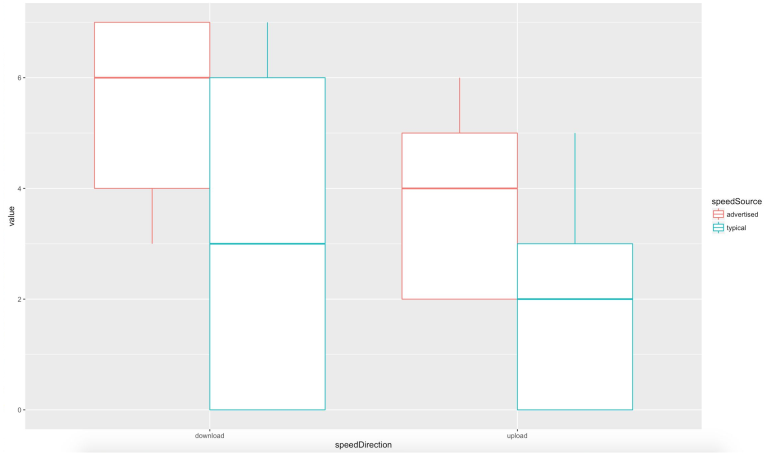
Data values for a single variable are found across tables

```
# 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 using an identifier column
ldply() #from the plyr package reads multiple csvs into one dataframe
```

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 (5 mins)

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

10 min break

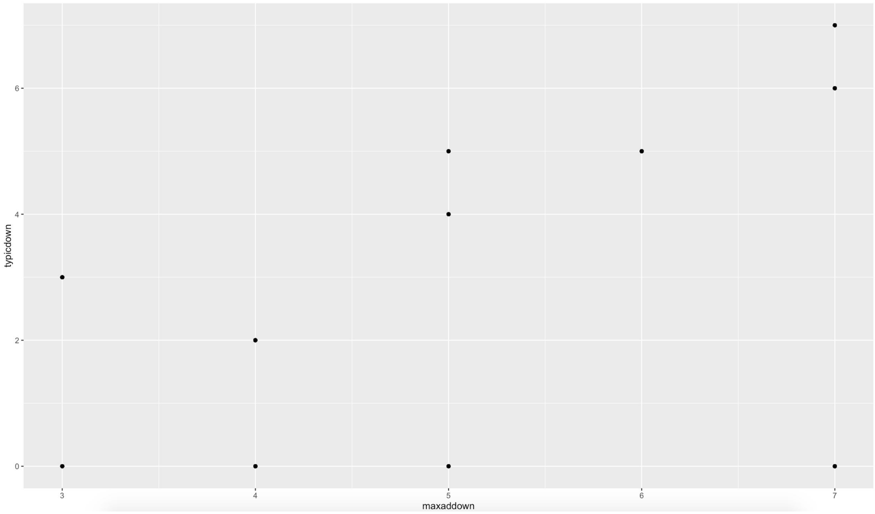
Correlation

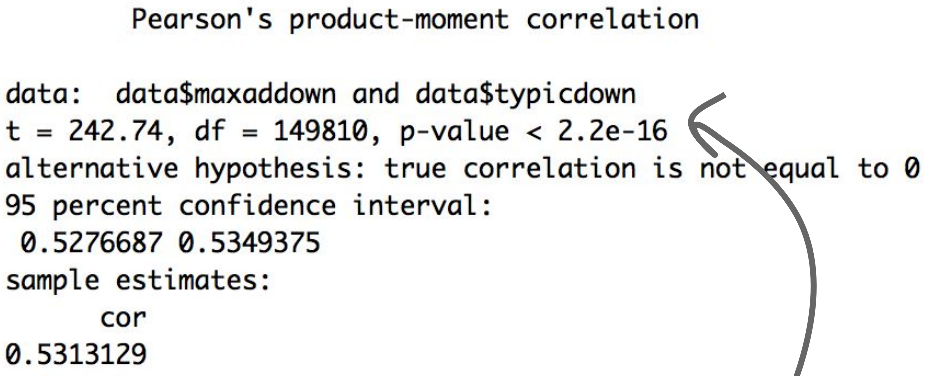
Correlation

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
```

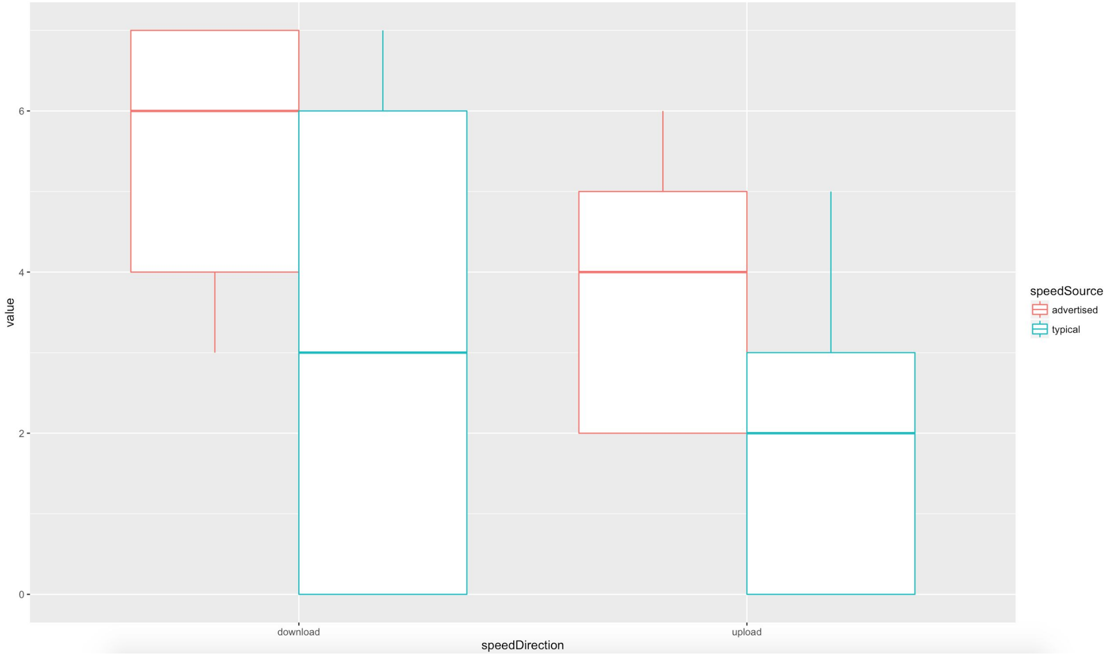
p-value < 0.05
not surprising

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

COMMERCE
DATA SERVICE

Comparing Samples



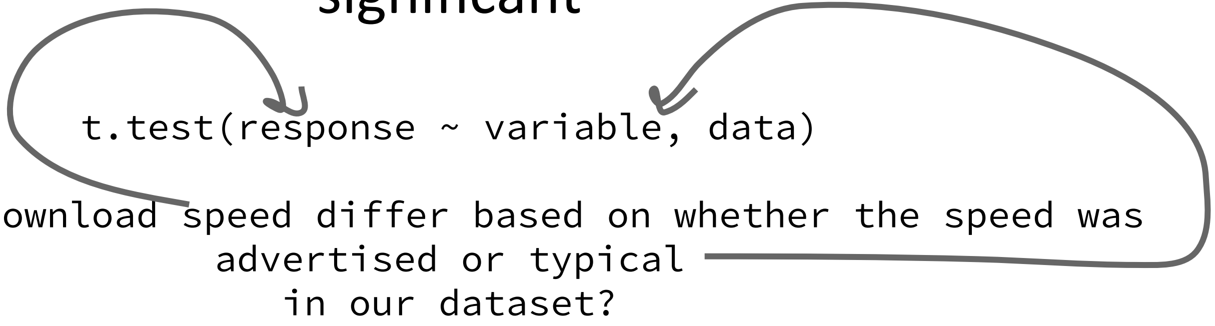
Null hypothesis: the typical download speeds for broadband providers in Washington, DC are the same as the advertised speeds

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



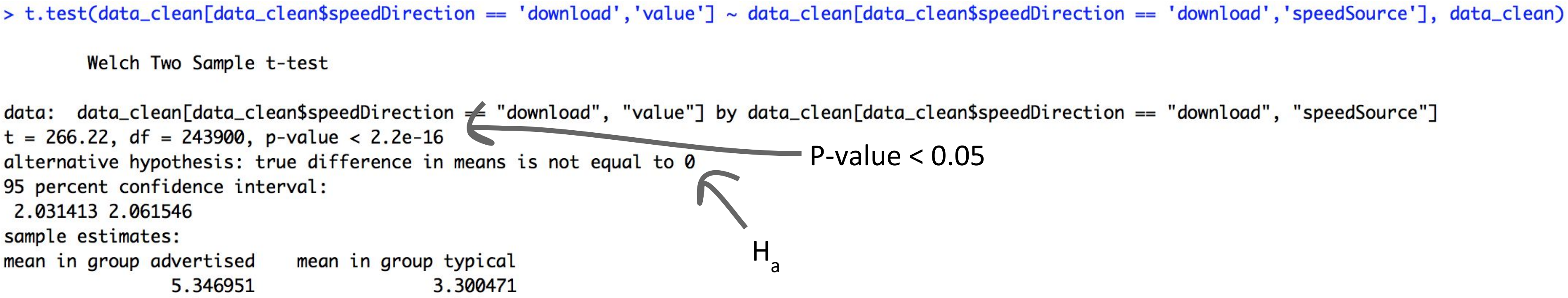
```
                    t.test(response ~ variable, data)

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

COMMERCE DATA SERVICE

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! (10 mins)

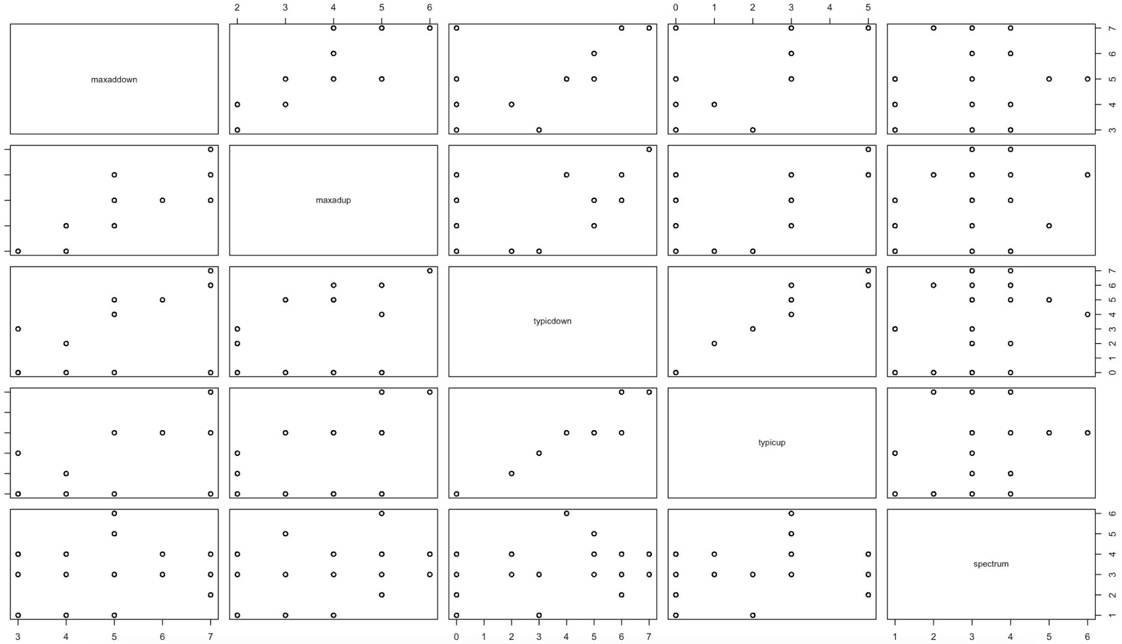
update the t-test code (chunk 10) to see if the typical upload speed is different than the advertised upload speed

10 min break

Analysis for Model Selection and Feature Selection

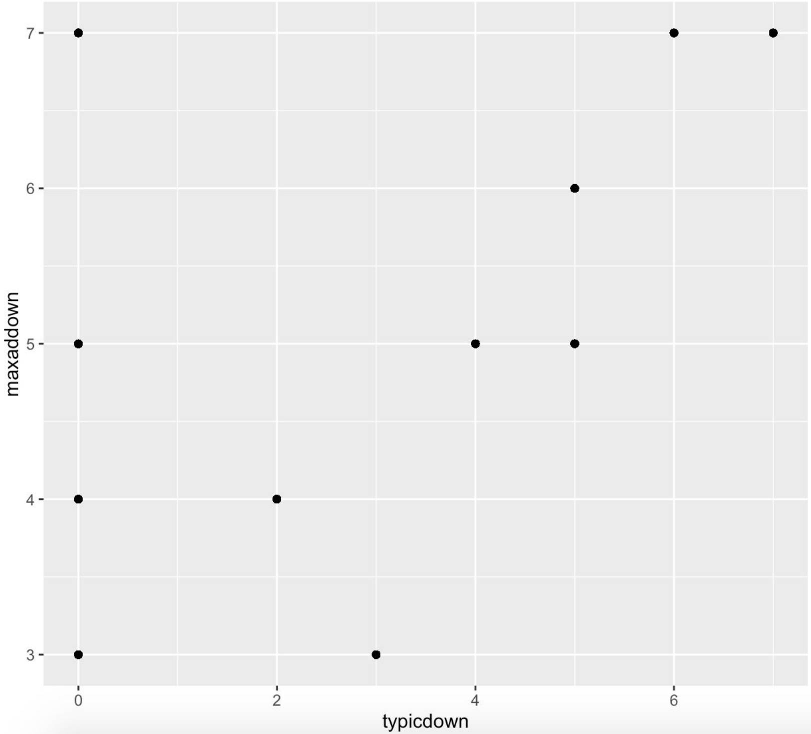
Scatterplot matrices

Relationship between continuous variables in your data

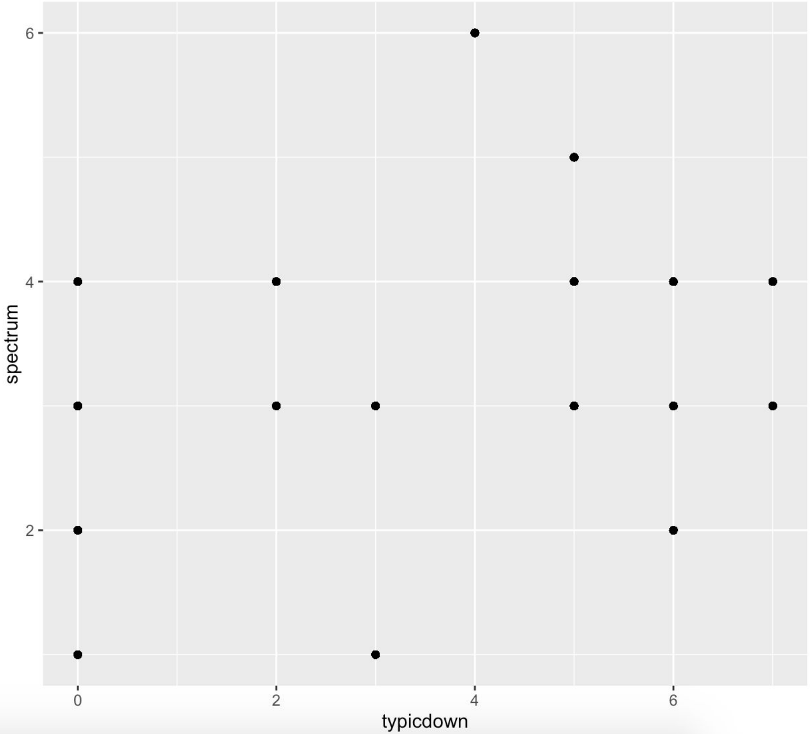


Linear regression

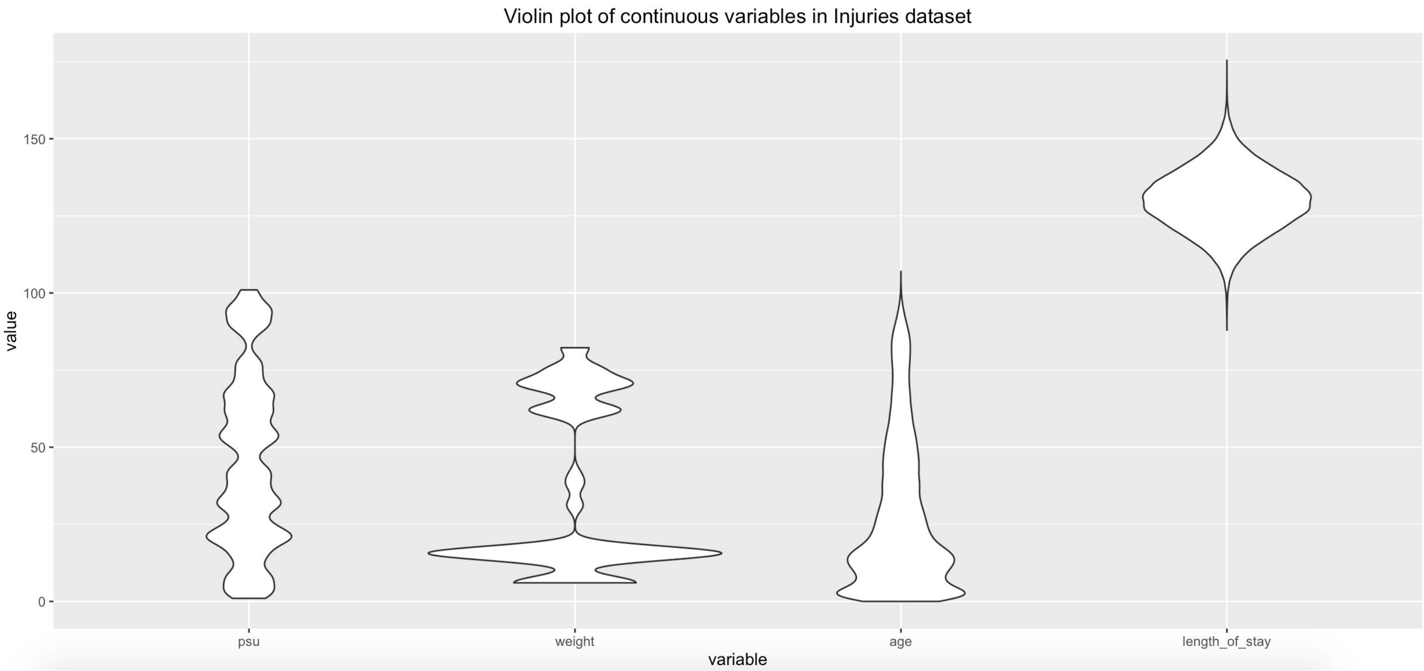
Scatterplot of typicdown and maxaddown



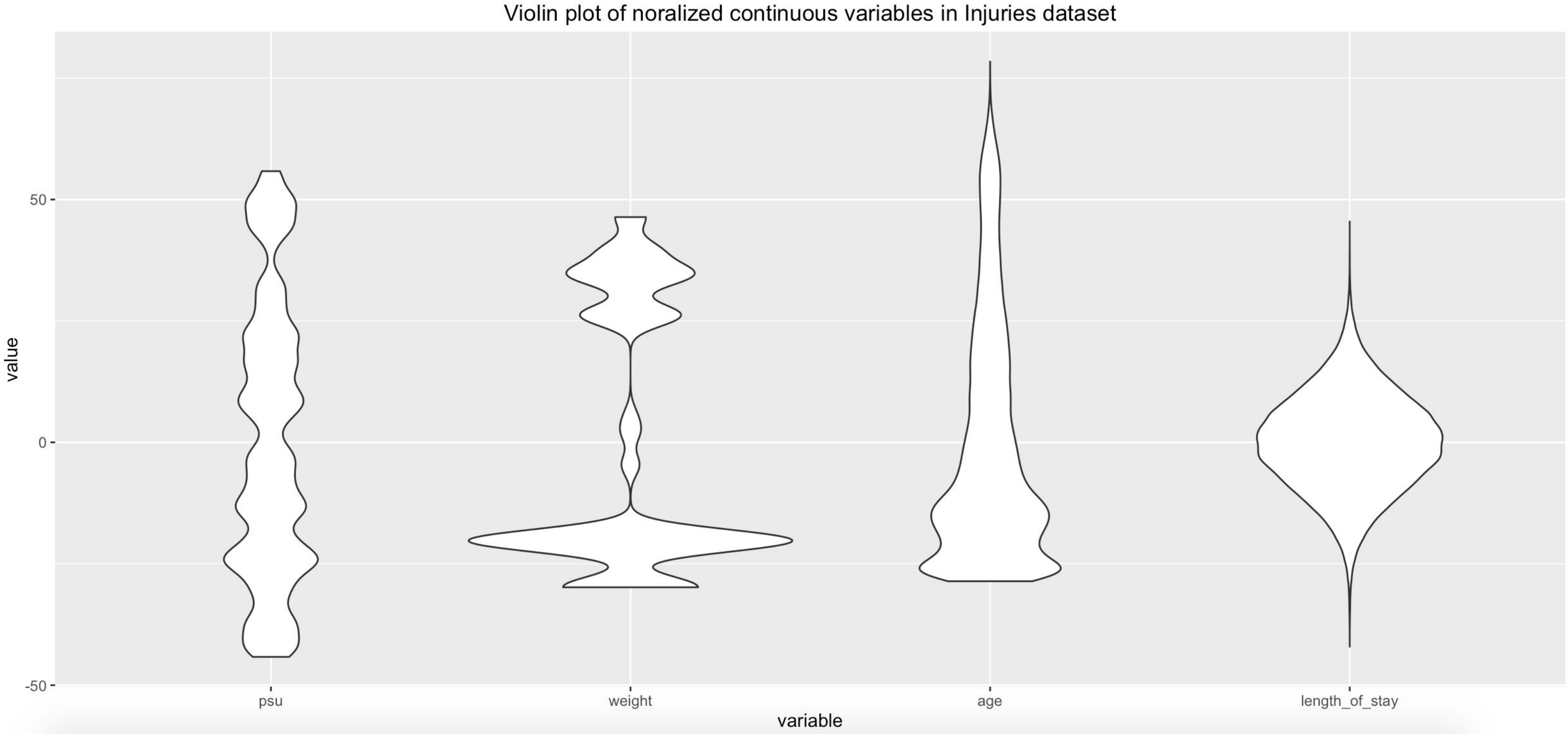
Scatterplot of typicdown and spectrum



Normalizing data and missing data



Normalized

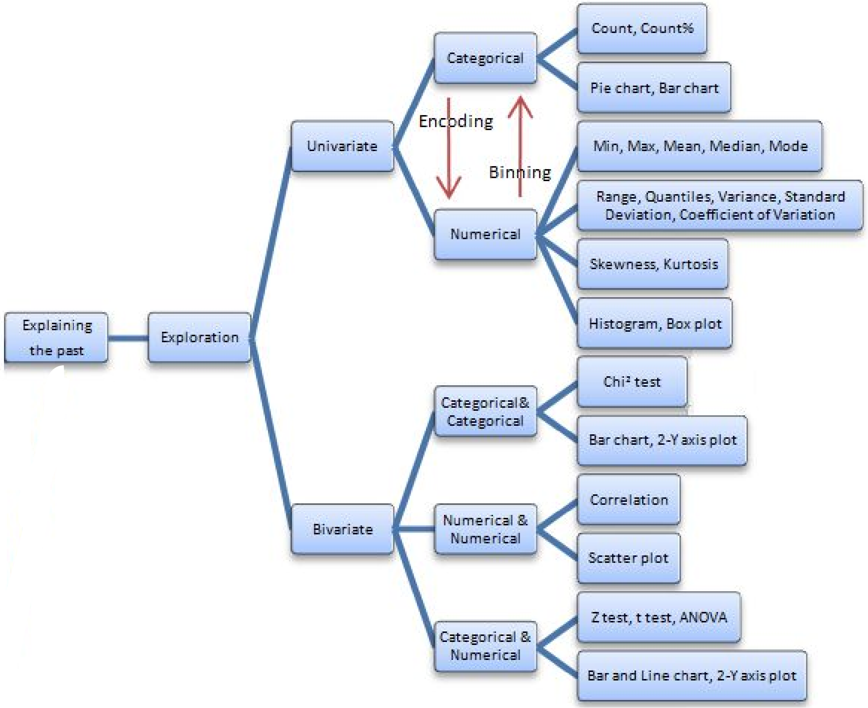

Violin plot of noralized continuous variables in Injuries dataset
value
50
0
-50
psu
weight
age
length_of_stay
variable

From statistical tests to statistical learning

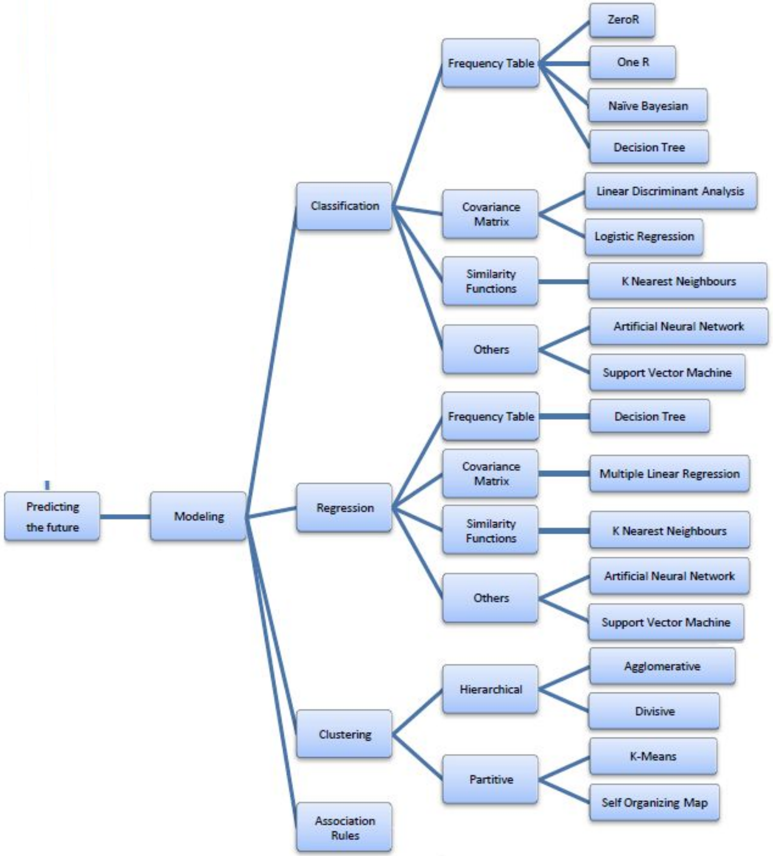
Statistical tests describe your data

Statistical learning allows you to make predictions about the future

COMMERCE DATA SERVICE



- Explaining the past
  - Exploration
    - Univariate
      - Categorical
        - Count, Count%
        - Pie chart, Bar chart
      - Encoding
      - Binning
      - Numerical
        - Min, Max, Mean, Median, Mode
        - Range, Quantiles, Variance, Standard Deviation, Coefficient of Variation
        - Skewness, Kurtosis
        - Histogram, Box plot
    - Bivariate
      - Categorical & Categorical
        - Chi² test
        - Bar chart, 2-Y axis plot
      - Numerical & Numerical
        - Correlation
        - Scatter plot
      - Categorical & Numerical
        - Z test, t test, ANOVA
        - Bar and Line chart, 2-Y axis plot



- Predicting the future
  - Modeling
    - Classification
      - Frequency Table
        - ZeroR
        - One R
        - Naïve Bayesian
        - Decision Tree
      - Covariance Matrix
        - Linear Discriminant Analysis
        - Logistic Regression
      - Similarity Functions
        - K Nearest Neighbours
      - Others
        - Artificial Neural Network
        - Support Vector Machine
    - Regression
      - Frequency Table
        - Decision Tree
      - Covariance Matrix
        - Multiple Linear Regression
      - Similarity Functions
        - K Nearest Neighbours
      - Others
        - Artificial Neural Network
        - Support Vector Machine
    - Clustering
      - Hierarchical
        - Agglomerative
        - Divisive
      - Partitive
        - K-Means
        - Self Organizing Map
    - Association Rules

Saed Sayad's data mining map

Statistical learning allows us to ask questions like:

Can we predict what the typical upload and download speeds are, for a given provider, if we know the advertised upload and download speeds?

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/

Saed Sayad's data mining map: http://www.saedsayad.com/data_mining_map.htm

See github repo for more resources!