Properties of Data

Digging into Data: Jordan Boyd-Graber

University of Maryland

February 11, 2013

COLLEGE OF
INFORMATION
STUDIES

Roadmap

- Munging 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
- CaBi

Caveat

- This is super important—everything else we do in the class won't work if the data aren't preprocessed well
- What I'm doing is not optimal
  - Probably not efficient to add columns in R, python, and Google spreadsheets
  - I'm doing it to show the breadth of options
  - Pick your poison and do what you need to do
  - You can (and should) use different tools: excel, SQL, java, perl, text editor

Outline

1. Data Terminology
2. Testbed: Capital Bikeshare
3. Adding Columns in R
4. Visualizing and Summarizing Data in Rattle
5. Adding Columns in Python
6. ggplot2
7. Wrapup

(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

- 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
- Why?

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

Quiz

- Height
- Gender
- Location

Quiz

- Height
    - Numeric
- Gender
- Location

Quiz

- Height
  - Numeric
- Gender
  - Categorical
- Location

Quiz

- Height
  - Numeric
- Gender
  - Categorical
- Location
  - Zip codes are numbers
  - Latitude and altitude are great numerical predictors of temperature

Outline

1. Data Terminology
2. **Testbed: Capital Bikeshare**
3. Adding Columns in R
4. Visualizing and Summarizing Data in Rattle
5. Adding Columns in Python
6. ggplot2
7. Wrapup

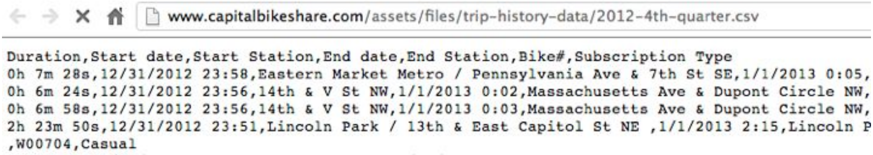
Capital Bikeshare

- 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`



www.capitalbikeshare.com/assets/files/trip-history-data/2012-4th-quarter.csv

```
Duration,Start date,Start Station,End date,End Station,Bike#,Subscription Type
0h 7m 28s,12/31/2012 23:58,Eastern Market Metro / Pennsylvania Ave & 7th St SE,1/1/2013 0:05,
0h 6m 24s,12/31/2012 23:56,14th & V St NW,1/1/2013 0:02,Massachusetts Ave & Dupont Circle NW,
0h 6m 58s,12/31/2012 23:56,14th & V St NW,1/1/2013 0:03,Massachusetts Ave & Dupont Circle NW,
2h 23m 50s,12/31/2012 23:51,Lincoln Park / 13th & East Capitol St NE ,1/1/2013 2:15,Lincoln P
,W00704,Casual
```

What story do you want to tell?

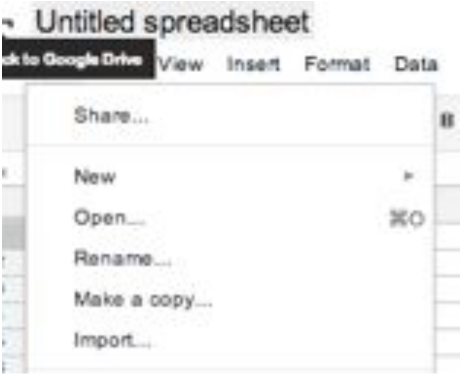
- What data are there?
- What information do you want?
- How to get from point A to point B?

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

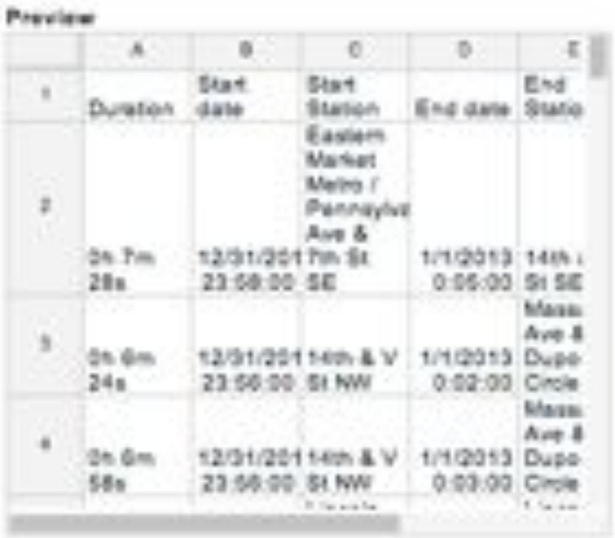
Adding it to Google Docs

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 0:06:00	14th St SE
3	0h 6m 24s	12/31/201 23:56:00	14th & V St NW	1/1/2013 0:02:00	Massa Ave & Dupo Circle
4	0h 6m 58s	12/31/201 23:56:00	14th & V St NW	1/1/2013 0:03:00	Massa Ave & Dupo Circle

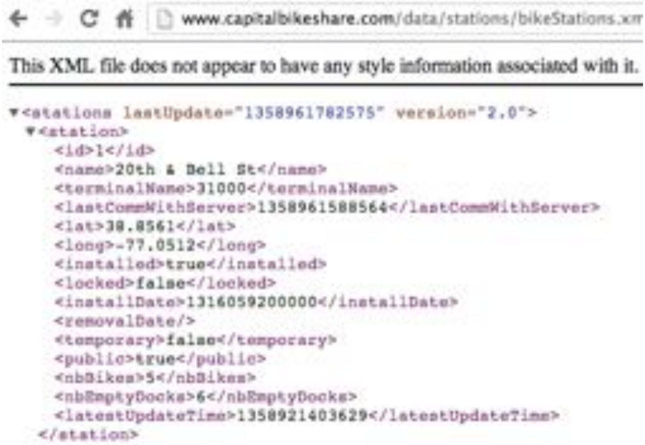
Adding it to Google Docs

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

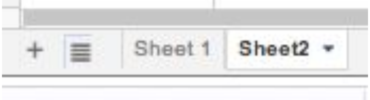


This XML file does not appear to have any style information associated with it.

```
▼<stations lastUpdate="1358961782575" version="2.0">
  ▼<station>
     <id>1</id>
     <name>20th & Bell St</name>
     <terminalName>31000</terminalName>
     <lastCommWithServer>1358961588564</lastCommWithServer>
     <lat>38.8561</lat>
     <long>-77.0512</long>
     <installed>true</installed>
     <locked>false</locked>
     <installDate>1316059200000</installDate>
     <removalDate/>
     <temporary>false</temporary>
     <public>true</public>
     <nbBikes>5</nbBikes>
     <nbEmptyDocks>6</nbEmptyDocks>
     <latestUpdateTime>1358921403629</latestUpdateTime>
   </station>
```

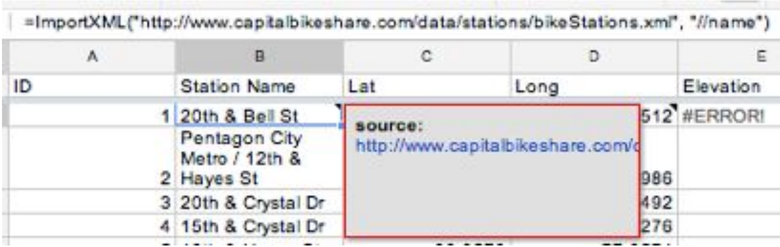
Adding it to Google Docs

Creating a new sheet just for stations



Adding it to Google Docs

Load columns from the xml file



=ImportXML("http://www.capitalbikeshare.com/data/stations/bikeStations.xml", "//name")

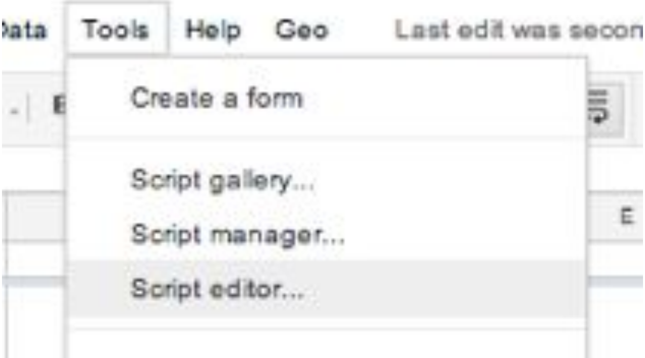
A	B	C	D	E
ID	Station Name	Lat	Long	Elevation
1	20th & Bell St		512	#ERROR!
2	Pentagon City Metro / 12th & Hayes St		986	
3	20th & Crystal Dr		492	
4	15th & Crystal Dr		276	

source:
http://www.capitalbikeshare.com/c

We now have columns for lat, long for every station

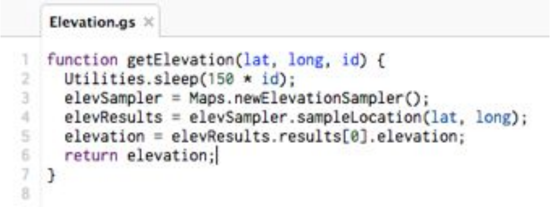
Adding it to Google Docs

Create a script to look up elevation



Adding it to Google Docs

Write the script

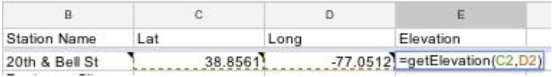


```
function getElevation(lat, long, id) {
  Utilities.sleep(150 * id);
  elevSampler = Maps.newElevationSampler();
  elevResults = elevSampler.sampleLocation(lat, long);
  elevation = elevResults.results[0].elevation;
  return elevation;
}
```

Now we can call this function in the spreadsheet to make a new elevation column for each station

Adding it to Google Docs

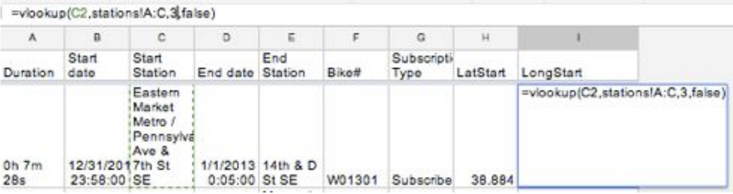
Call the script



B	C	D	E
Station Name	Lat	Long	Elevation
20th & Bell St	38.8561	-77.0512	=getElevation(C2,D2)

Adding it to Google Docs

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

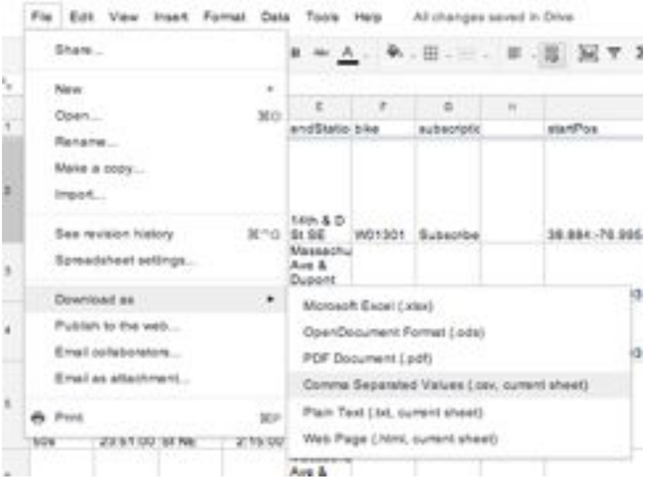


```
=vlookup(C2,stations!A:C,3,false)
```

A	B	C	D	E	F	G	H	I
Duration	Start date	Start Station	End date	End Station	Bike#	Subscripti Type	LatStart	LongStart
0h 7m 28s	12/31/201 23:58:00	Eastern Market Metro / Pennsylva Ave & 7th St SE	1/1/2013 0:05:00	14th & D St SE	W01301	Subscribe	38.884	=vlookup(C2,stations!A:C,3,false)

Adding it to Google Docs

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



Outline

1. Data Terminology
2. Testbed: Capital Bikeshare
3. **Adding Columns in R**
4. Visualizing and Summarizing Data in Rattle
5. Adding Columns in Python
6. ggplot2
7. Wrapup

Loading a dataset

```
rides <- read.csv("data/cabi-sample-rides.filtered.c
```

- 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

Functions in R

- Defined using the command “function”
- Can take untyped arguments
- Return a value with the return command
- Assigned to a variable name

Functions in R

```
earthDistance <- function(loc1, loc2) {
  leftFields = strsplit(loc1, ":")
  rightFields = strsplit(loc2, ":")

  lat1 = as.numeric(leftFields[[1]][1]) * PI / 180.0
  lat2 =  as.numeric(rightFields[[1]][1]) * PI / 180.0

  lon1 = as.numeric(leftFields[[1]][2]) * PI / 180.0
  lon2 =  as.numeric(rightFields[[1]][2]) * PI / 180.0

  x = (lon2-lon1) * cos((lat1+lat2)/2);
  y = (lat2-lat1);
  d = sqrt(x*x + y*y) * EARTH_RADIUS;

  return(d)
}
```

Adding columns in R

```
rides$elevation <- apply(rides, 1, function(row)
        elevationDistance(row[’startPos’], row[’endPos’]
rides$distance <- apply(rides, 1, function(row)
        earthDistance(row[’startPos’], row[’endPos’]))
rides$duration <- apply(rides, 1, function(row)
        timeDistance(row[’startDate’], row[’endDate’]))
rides$startHour <- apply(rides, 1, function(row)
        dayHour(row[’startDate’]))
```

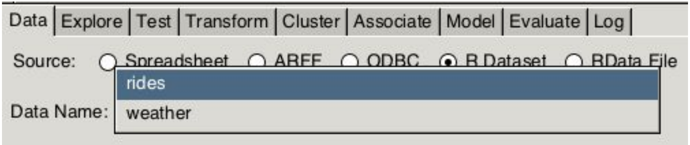
- Adds additional columns to the dataframe
- apply function works on the dataset we loaded from the csv
- You can download the script from the source page to see more function examples (earthDistance is the most complicated)
- “apply” works on the dataset **rides**’s rows (1) to apply the specified function to each row (based on the input accessed from the columns)

Loading a modified dataframe in Rattle

- Read in the data

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

- Do what you need to do (i.e. add columns)
- Choose "R Dataset" as our source



Source: ○ Spreadsheet ○ ARFF ○ ODBC ● R Dataset ○ RData File

Data Name: rides

Writing out the csv

```
write.csv(rides, "data/cabi-rides.ext.cvs")
```

- In case you want to do something else in a spreadsheet
- For future reference
- To get help

Outline

1. Data Terminology
2. Testbed: Capital Bikeshare
3. Adding Columns in R
4. **Visualizing and Summarizing Data in Rattle**
5. Adding Columns in Python
6. ggplot2
7. Wrapup

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
```

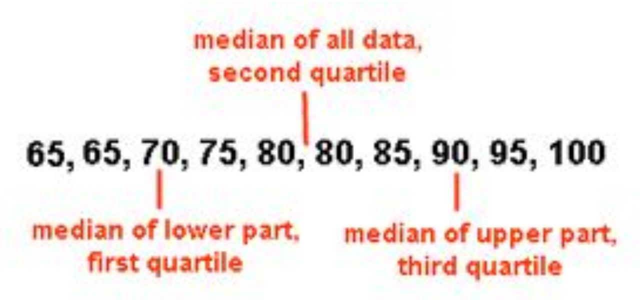
Summarizing Data

Getting Output Directly

- “Explore” tab
- Type: “summary”

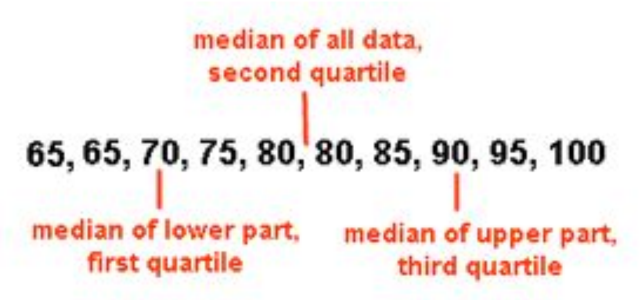
```
                              endStation      distance         startHour
 Massachusetts Ave & Dupont Circle NW: 148   Min.   :    0.0   Min.   : 0.1333
 15th & P St NW                      : 103   1st Qu.:  921.5   1st Qu.:10.5500
 Thomas Circle                       :  94   Median : 1515.5   Median :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



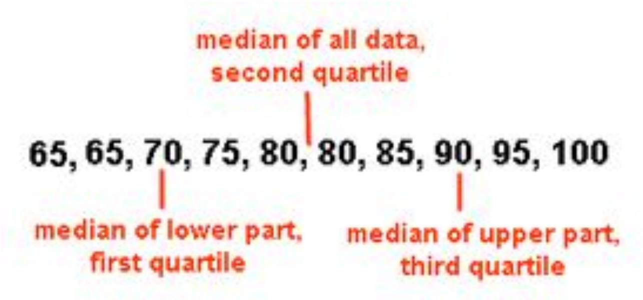
- Order your data
- Find the middle data point - this is your median
  - If even number of data points, average points in the middle
- Repeat on two halves on either side of median - these are your first and third quartiles

Descriptive Statistics



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

Descriptive Statistics



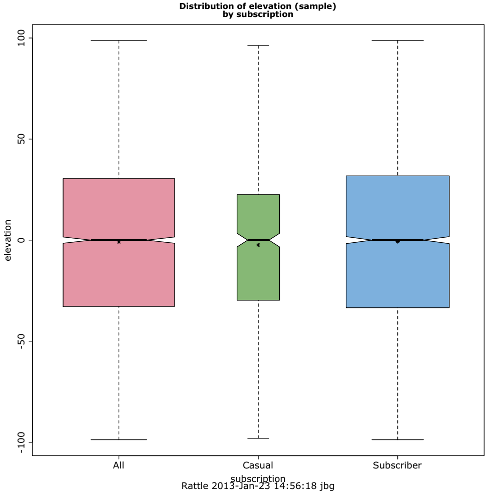
- min - smallest data point
- max - largest data point
- mean - sum of all data divided by number of data points

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

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?

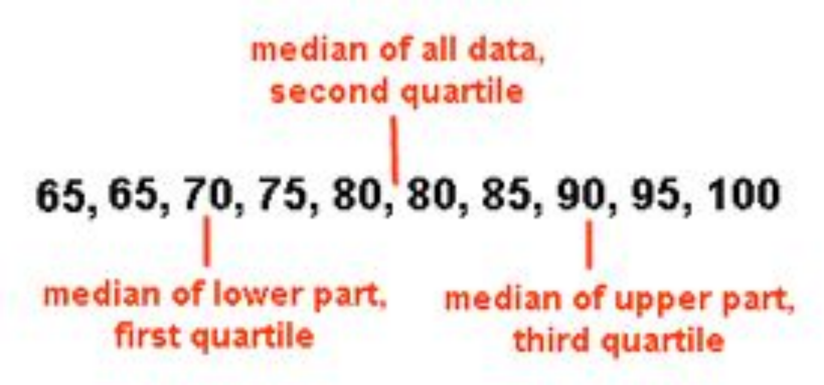
Box Plots



- 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”

Box Plots

What would this box plot look like?



median of all data, second quartile

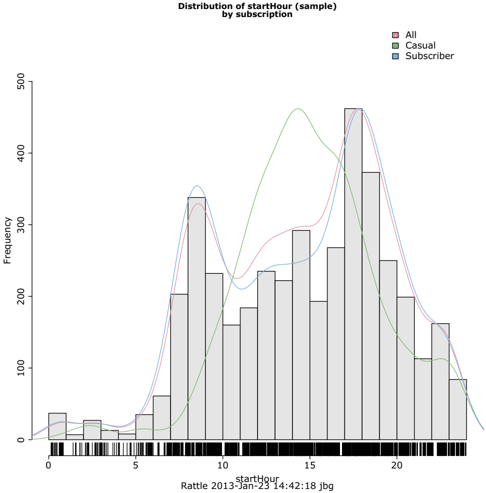
65, 65, 70, 75, 80, 80, 85, 90, 95, 100

median of lower part, first quartile

median of upper part, third quartile

Histogram

- Chop your range into bins (art to this)
- Count how many data fall in each bin
- Gives a better sense of shape of distribution
- Under “Explore”, “Histogram”



Outline

1. Data Terminology
2. Testbed: Capital Bikeshare
3. Adding Columns in R
4. Visualizing and Summarizing Data in Rattle
5. **Adding Columns in Python**
6. ggplot2
7. Wrapup

Python support for csv

- Python has built in DictReader and DictWriter classes
- Easy to add columns
- Example on course webpage
  - Counts up check outs and check ins per station
  - Bins time into human-readable categories (e.g. early morning, afternoon)
  - Output csv also available

Outline

1 Data Terminology

2 Testbed: Capital Bikeshare

3 Adding Columns in R

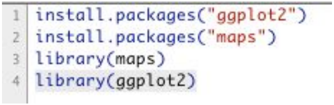
4 Visualizing and Summarizing Data in Rattle

5 Adding Columns in Python

6 **ggplot2**

7 Wrapup

ggplot2



```
install.packages("ggplot2")
install.packages("maps")
library(maps)
library(ggplot2)
```

- Library created by Hadley Wickham
- Load it by using “library(ggplot2)”
- Creates very attractive plots
- Very easy to customize

ggplot2 maps

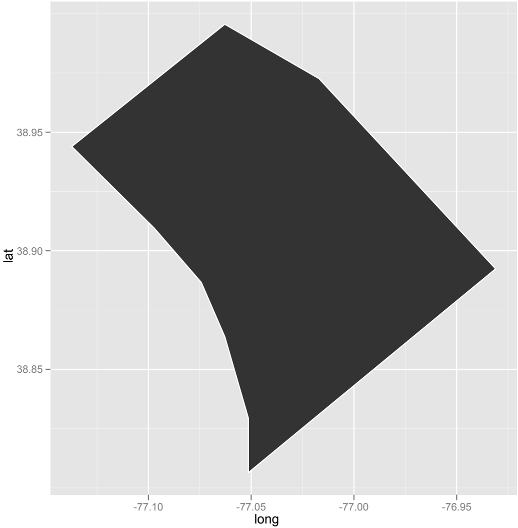
Get an outline of DC

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

Draw it

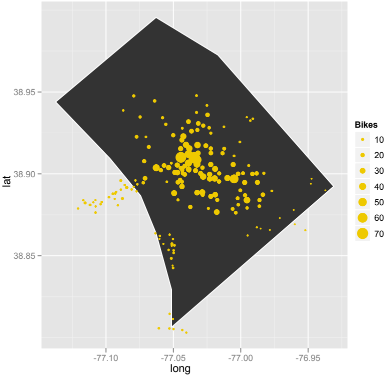
```
p <- ggplot(stations)
p <- p + geom_polygon( data=states, aes(x=long, y=lat))
```

ggplot2 maps



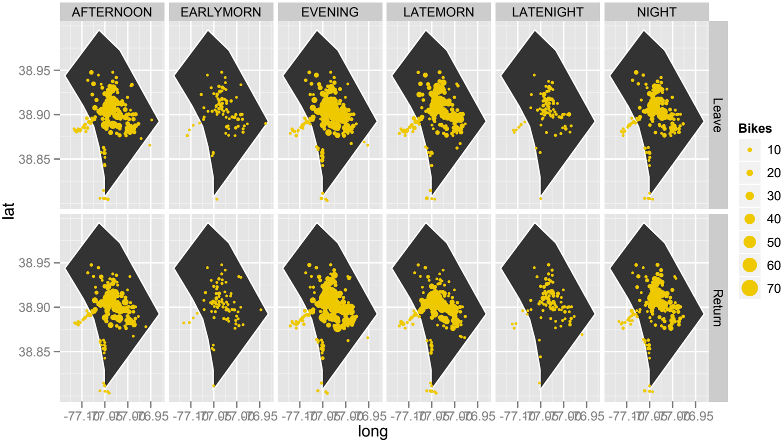
ggplot2 maps

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



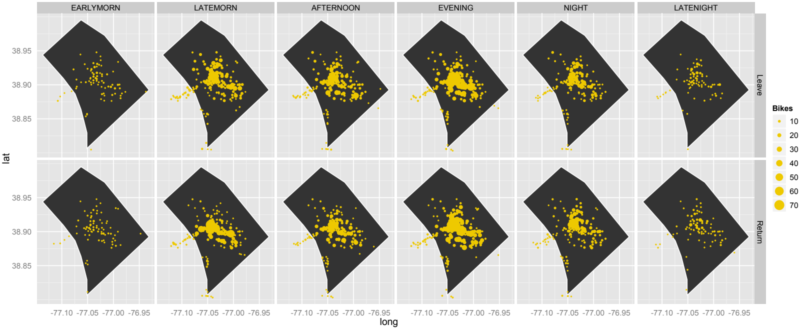
ggplot2 facets

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



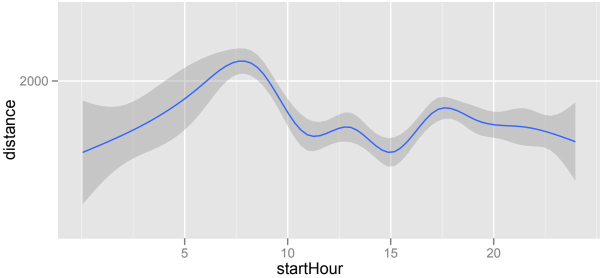
ggplot2 facets (resorted)

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



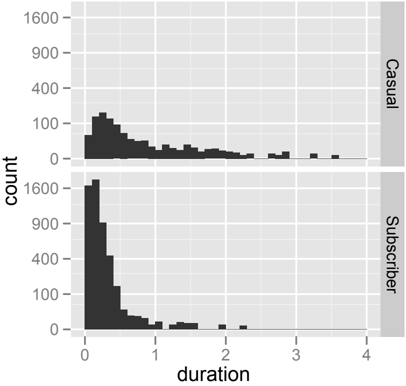
ggplot2 scatterplots

```
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

1. Data Terminology
2. Testbed: Capital Bikeshare
3. Adding Columns in R
4. Visualizing and Summarizing Data in Rattle
5. Adding Columns in Python
6. ggplot2
7. **Wrapup**

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

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!)