
Introduction to Visualizations with R for Exploratory Data Analysis

Irina Kukuyeva, Ph.D.
www.KukuyevaConsulting.com

March 3, 2016



Tutorial Outline

Preliminaries

Introduction to R

Visualizations

Working with R



Part I

Preliminaries



1 Motivation

- Why tutorial?
- Why R?
- Why visualize?

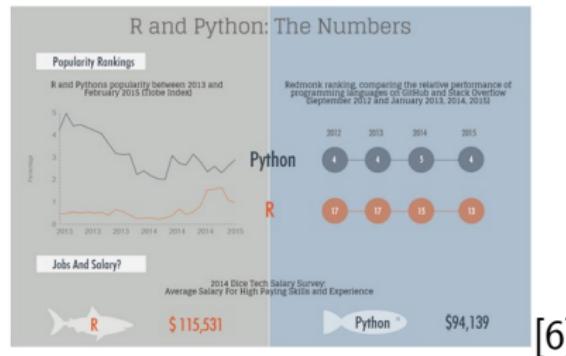
2 Tips for Visualizations



Why tutorial?

Why Tutorial?

- Derive and convey key insights from data
- Learn and use open source software
- Explore first/another programming language
- Gain ability to create insightful visualizations
- Be more marketable



Why R?

- Absolutely free!
- Used in industry and academia.
- Has a great community:
 - StackOverflow
 - Blogs
 - Meetup groups
 - MOOCs
 - many, many others
- Has over 7900 packages available for use (for free!).
- Transparent code (e.g. easier to check for bugs).

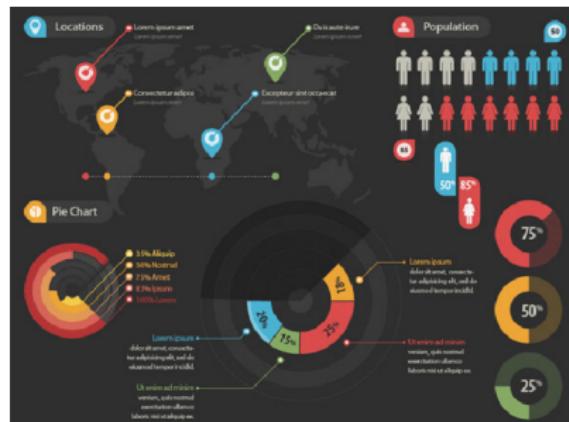


Why visualize?

Approach 1: Infographic

Infographic

- Manually created
- Customized to a data set
- Art



[10]



Approach 2: Data Visualization I

Data Visualization

- Automated via software
- Reproducible for a different data set
- (Less? of an) art

There are two types of data visualization:

1. Exploratory

- Capture uncertainty in the data
- Explore potential relationships/trends/missing patterns (or absence thereof)

Approach 2: Data Visualization II

2. Explanatory

- Convey key information



1 Motivation

2 Tips for Visualizations



Tips for visualizations

- Know your audience
- *15 second rule:* if your audience won't be able to understand the graphic in 15 seconds, simplify
- Layout [9] and font choices
- Color choices
 - Available colors in R:
 - <http://research.stowers-institute.org/efg/R/Color/Chart/>
 - Color blindness simulation [11]:
 - <http://vischeck.com>
 - <http://rsb.info.nih.gov/ij/>
- Add text/labels to figure



Part II

Introduction to R



3 Installing R and RStudio

4 Overview of R

5 Working with Data

6 Adding Functionality to Base R

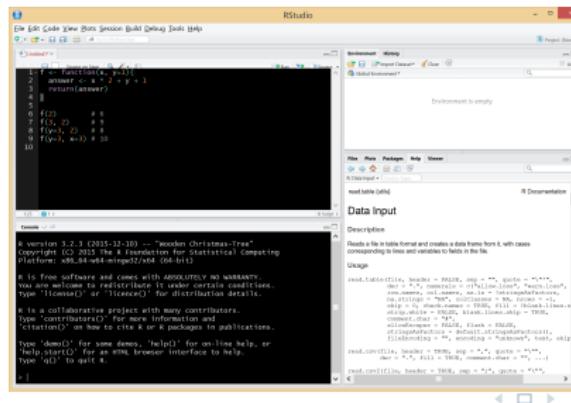


Installing R and RStudio

R: Go to <http://cran.r-project.org/>, select your operating system and download the latest version: 3.2.3 (Release 2015/12/10).

RStudio: Go to

<https://www.rstudio.com/products/rstudio/download/>, select your operating system and download the installer (if available).



3 Installing R and RStudio

4 Overview of R

- Creating Objects
- (Very) Brief Overview of Functions
- Comparisons
- (More) Notation

5 Working with Data

6 Adding Functionality to Base R



Overview of R per John Chambers [1]:

"To understand computations in R, two slogans are helpful:

- Everything that exists is an object.
- Everything that happens is a function call."



Creating Objects

Objects in R I

Creating objects in R

Objects can be created in different ways, via:

- equal sign: `x = 3`
- left arrow: `y <- 2 * x + 1`
- right arrow: `10 -> z`



Objects in R II

Examples of Objects

- Variables, which can be either discrete or continuous:

Discrete: e.g. number of people at the tutorial

Continuous: e.g. how long it took to get to the tutorial

- Data frames, which contain data sets with at least one variable:

EX 1: e.g. data set of commonly performed procedures at CA hospitals (variables include hospital name, procedure type, number of patients who had that procedure, etc.)

EX 2: e.g. data set of popular movies (variables include movie name, genre, lead actor/actress, etc.)

(Very) Brief Overview of Functions

Functions in R

Function by example

```
1 f <- function(x, y=1){  
2     answer <- x * 2 + y + 1  
3     return(answer)  
4 }  
5  
6 f(2)      # 6  
7 f(3, 2)    # 9  
8 f(y=3, 2)  # 8  
9 f(y=3, x=3) # 10
```

Components of a function:

- Assign function to a variable
- Add the 'function' keyword
- Specify arguments (if any) that function needs to compute result
- Specify any argument defaults

Comparisons

Making Comparisons

Comparisons return a TRUE or FALSE depending on the condition evaluated:

```
1 # Suppose x and y are:  
2 x = 3  
3 y = 5
```

- 1 What should the following comparisons return?
- 2 $x == 3$ # is x 3?
- 3 $x != y$ # are x and y different?
- 4 $x >= y$ # is x greater than or equal to y?
- 5 $(x==3) & (y==4)$ # is it true that x is 3 and y is 4?
- 6 $(x==3) | (y==4)$ # is it true that x is 3 or y is 4?



(More) Notation

Notation/Conventions I

`%>%`: 'pipe operator' which sends objects on the left-hand side to be processed by functions on the right-hand side [15]

- `x %>% f` is equivalent to `f(x)`

`c()`: 'concatenation' which combines objects together

```
1 x = 3
2 y = 5
3 z = c(x, y)
4 z
```

```
[1] 3 5
```



[\(More\) Notation](#)

Notation/Conventions II

Brackets: 'subsetting' which says what observation we are (not) interested in

```
1 # Example 1:  
2 z = c(3, 5)  
3 z[2]
```

```
[1] 5
```

```
1 # Example 2:  
2 z = c(3, 5)  
3 z[z > 3]
```

```
[1] 5
```



3 Installing R and RStudio

4 Overview of R

5 Working with Data

- Reading Data from File
- Working with Data Frames

6 Adding Functionality to Base R



Reading Data from File

Reading Data from File I

One approach is via `read.table()`. In the arguments of the function:

- `file`: specifies the (relative) location, file name and file extension
- `header`: if TRUE, tells R to include variables names when importing
- `sep`: tells R how the entries in the data set are separated
 - `sep=","`: when entries are separated by COMMAS
 - `sep="\t"`: when entries are separated by TAB
 - `sep=" "`: when entries are separated by SPACE



Reading Data from File

Reading Data from File II

```
1 filepath = "http://www.ats.ucla.edu/stat/data/test_
missing_comma.txt"
2 #### Other valid paths:
3 # filepath = "C:/Documents/test_missing_comma.txt"
4 # filepath = ".\test_missing_comma.txt"
5
6 df <- read.table(
7     file = filepath,
8     header = TRUE,
9     sep = ","
10 )
```



Working with Data Frames

Working with Data Frames I

To check that a data set has been read-in correctly:

- View the complete data set by typing the variable name.
(This is not recommended for large data sets.)

```
1 df
```

	prgtype	gender	id	ses	schtyp	level
1	general	0	70	4	1	1
2	vocati	1	121	4	NA	1
3	general	0	86	NA	NA	1
4	vocati	0	141	4	3	1
5	academic	0	172	4	2	1
6	academic	0	113	4	2	1
7	general	0	50	3	2	1
8	academic	0	11	1	2	1



Working with Data Frames

Working with Data Frames II

- View the first 5 lines of the dataset via `head()`:

```
1 head(df, 5)
```

	prgtype	gender	id	ses	schtyp	level
1	general	0	70	4	1	1
2	vocati	1	121	4	NA	1
3	general	0	86	NA	NA	1
4	vocati	0	141	4	3	1
5	academic	0	172	4	2	1



Working with Data Frames

Working with Data Frames III

- View the last 7 lines of the dataset via `tail()`:

```
1 tail(df, 7)
```

	prgtype	gender	id	ses	schtyp	level
2	vocati	1	121	4	NA	1
3	general	0	86	NA	NA	1
4	vocati	0	141	4	3	1
5	academic	0	172	4	2	1
6	academic	0	113	4	2	1
7	general	0	50	3	2	1
8	academic	0	11	1	2	1



Working with Data Frames

Working with Data Frames IV

- Check the variable names via `names()`:

```
1 names(df)
```

```
[1] "prgtype" "gender"   "id"        "ses"       "schtyp"    "level"
```

- Check the size of the data set via `dim()`:

```
1 dim(df)
```

```
[1] 8 6
```

- See the first 6 entries for variable 'gender' via `head()`:

```
1 head(df$gender)
```

```
[1] 0 1 0 0 0 0
```



Working with Data Frames

Working with Data Frames V

- Examine how many unique levels a variable has via `unique()`:

```
1 unique(df$gender)
```

```
[1] 0 1
```

- Examine the counts of levels that a variable has via `table()`:

```
1 table(df$gender, useNA='always')
```

0	1	<NA>
7	1	0

Working with Data Frames

Working with Data Frames VI

- View entries for a particular variable:

```
1 ## Returns variable as a row/vector:
2 df$gender # OR
3 df[, 'gender']
4 ## Returns variable as a column:
5 df[ 'gender' ]
```

```
[1] 0 1 0 0 0 0 # OR
```

```
gender
1      0
2      1
3      0
4      0
5      0
6      0
7      0
8      0
```



Working with Data Frames VII

- View entries for a few variables:

```
1 head( df[, c('gender', 'ses') ], 3)
```

	gender	ses
1	0	4
2	1	4
3	0	NA



Working with Data Frames

Working with Data Frames VIII

- Verify ranges and check for missing data via `summary()`:

```
1 summary(df)
```

```
prgtype      gender       id        ses        schtyp      level
vocati:2    Min.   :0.000  Min.   :11.0  Min.   :1.000  Min.   :1  Min.   :1
general:3   1st Qu.:0.000  1st Qu.:65.0  1st Qu.:3.500  1st Qu.:2  1st Qu.:1
academic:3  Median :0.000  Median :99.5  Median :4.000  Median :2  Median :1
              Mean   :0.125  Mean   :95.5  Mean   :3.429  Mean   :2  Mean   :1
              3rd Qu.:0.000  3rd Qu.:126.0 3rd Qu.:4.000  3rd Qu.:2  3rd Qu.:1
              Max.   :1.000  Max.   :172.0  Max.   :4.000  Max.   :3  Max.   :1
                               NA's   :1           NA's   :1           NA's   :2
```



3 Installing R and RStudio

4 Overview of R

5 Working with Data

6 Adding Functionality to Base R



Adding Functionality to Base R

- Base R is what you download off CRAN
www.cran.r-project.org
- Available packages are listed here:
<https://cran.r-project.org/web/packages/>
- Install an R package(s) via `install.packages()`:

```
1 install.packages("lattice")
2 install.packages("ggplot2")
3 # OR
4 install.packages( c("lattice", "ggplot2") )
```



Part III

Visualizations



7 Data Set

8 Plots of Counts

9 Time Series Plots

10 Geographical Plots

Data Set for Tutorial I

Data set for this tutorial comes from State of California. Please export it to CSV from:

[https://chhs.data.ca.gov/Healthcare/
Number-of-Selected-Inpatient-Medical-Procedures-Ca/
mdt8-gwyw](https://chhs.data.ca.gov/Healthcare/Number-of-Selected-Inpatient-Medical-Procedures-Ca/mdt8-gwyw)

 CHHS Open Data

Sign Up Sign In Accessibility

Home Catalog Tutorials Developers About

Find in this Dataset

Year	County	Hospital Name	OSHPID	Procedure	Volume	Longitude	Latitude	Local
1 2014	Orange	Saint Jude Medical Center	106301342	Pancreatic Resection	2	-117.92851	33.89349	(3)
2 2014	San Diego	Scripta Memorial Hospital La Jolla	106370771	CABG	367	-117.22279	32.88506	(3)
3 2014	Fresno	Adventist Medical Center Reedley	106100797	Pancreatic Resection	-	-119.45145	36.60780	(3)
4 2014	Reno	Adventist Medical Center Reedley	106100797	AAA Repair (Ruptu)	-	-119.45145	36.60780	(3)
5 2014	Orange	Saint Jude Medical Center	106301342	Esophageal Repair	1	-117.92851	33.89349	(3)
6 2014	San Diego	Scripta Memorial Hospital La Jolla	106370771	AAA Repair (Unrg)	9	-117.22279	32.88506	(3)
7 2014	San Diego	Scripta Memorial Hospital La Jolla	106370771	AAA Repair (Ruptu)	2	-117.22279	32.88506	(3)
8 2014	San Diego	Scripta Memorial Hospital La Jolla	106370771	AAA Repair (Ruptu)	-	-117.22279	32.88506	(3)
9 2014	San Diego	Scripta Memorial Hospital La Jolla	106370771	AAA Repair	13	-117.22279	32.88506	(3)
10 2014	San Diego	Scripta Memorial Hospital La Jolla	106370771	Pancreatic Resection	1	-117.22279	32.88506	(3)
11 2014	Orange	Saint Jude Medical Center	106301342	PO	235	-117.92851	33.89349	(3)
12 2014	Orange	Saint Jude Medical Center Reedley	106100797	AAA Repair (Unrg)	-	-117.92851	33.89349	(3)

Manage More Views Print Visualize Import Download Email Help

© 2015 California Health & Human Services Agency



Data Set for Tutorial II

To read it into R, type:

```
1 df <- read.table(  
2   "Number_of_Selected_Inpatient_Medical_Procedures  
     __California_Hospitals__2005–2014.csv",  
3   sep = ",",  
4   header=TRUE,  
5   stringsAsFactors = FALSE  
6 )
```

Check that the file was read-in correctly:

```
1 dim(df) # 45,438 rows and 9 columns
```

Rename the first variable to make it easier to work with:

```
1 names(df)[1] = 'Year'
```



7 Data Set

8 Plots of Counts

- Histogram
- Scatterplot
- Exercise I

9 Time Series Plots

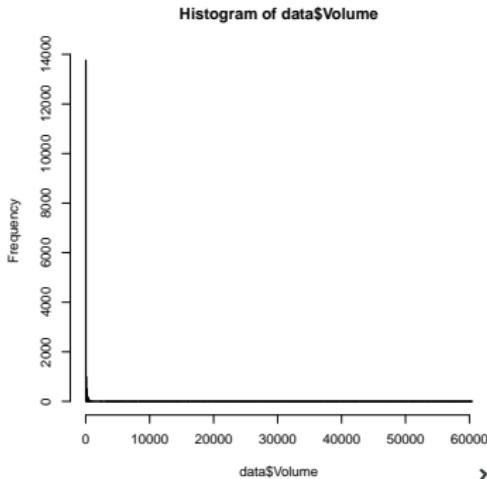
10 Geographical Plots

Histogram

Histogram I

To see the counts of values that a variable has (i.e. its distribution), use `hist()`:

```
1 hist(  
2   df$Volume,  
3   breaks=1000  
4 )  
5 # Is anything 'off'?
```



Histogram

Histogram II

Let's examine the potential outlier:

```
1  ### Step 0: Load required packages
2  library(dplyr)
3
4  ### Step 1: Examine entry when volume is
      largest in data set
5  df %>%
6    filter( Volume == max(Volume, na.rm=TRUE) )
```

	Year	County	Hospital.Name	OSHPDID	Procedure	Volume	Longitude	Latitude	Location
41783	2005	STATEWIDE	STATEWIDE	NA	PCI	60418	NA	NA	

Entry seems to be an aggregate number, not county/year level.



Histogram

Histogram III

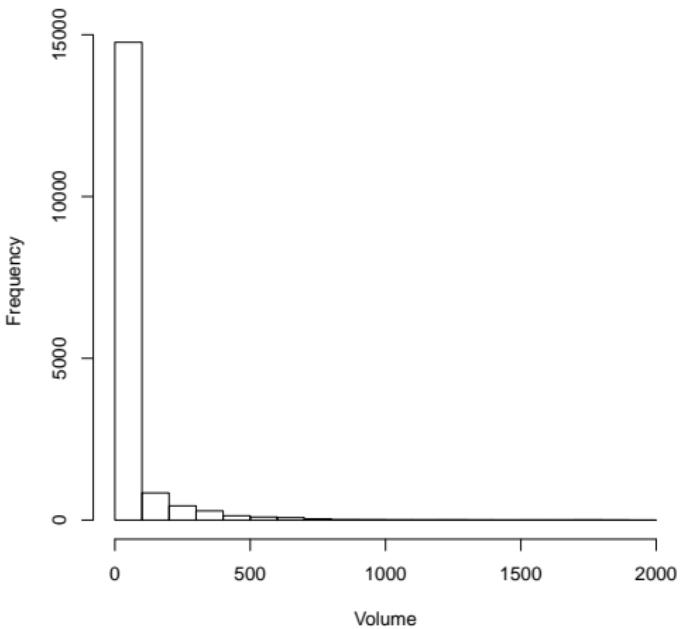
To remove any extraneous data points:

```
1 #### Step 0: Load required packages
2 library(dplyr)
3
4 #### Step 1: Remove any 'statewide' counts
5 data_clean <- df %>%
6   filter(County != "STATEWIDE")
7
8 #### Step 2: Visualize
9 hist(data_clean$Volume,
10       xlab = "Volume",
11       main = "")
12
13 # Is anything 'off' now? (Exercise.)
```



Histogram

Histogram IV



Side-by-side Histogram I

To compare distributions of two variables side-by-side, use `beanplot()`:

```
1  ### Step 0: Load required packages
2  library(beanplot)
3
4  ### Step 1: Subset data set to LA and SF
5  data_LA_SF <- data_clean %>%
6    filter(
7      (County == "Los Angeles") |
8      (County == "San Francisco"))
9  )
```



Side-by-side Histogram II

```
1 ### Step 3: Visualize
2 op <- par(las=2) # orient y-axis labels
3 beanplot(
4   Volume ~ as.factor(County),
5   data = data_LA_SF,
6   xlab = "",
7   log = "y",
8   side = "both",
9   col = list( c(grey(0.5), "white"), grey(0.8) ),
10  border = NA,
11  overallline = "median",
12  ll = 0.005,
13  show.names=FALSE
14 )
```



Histogram

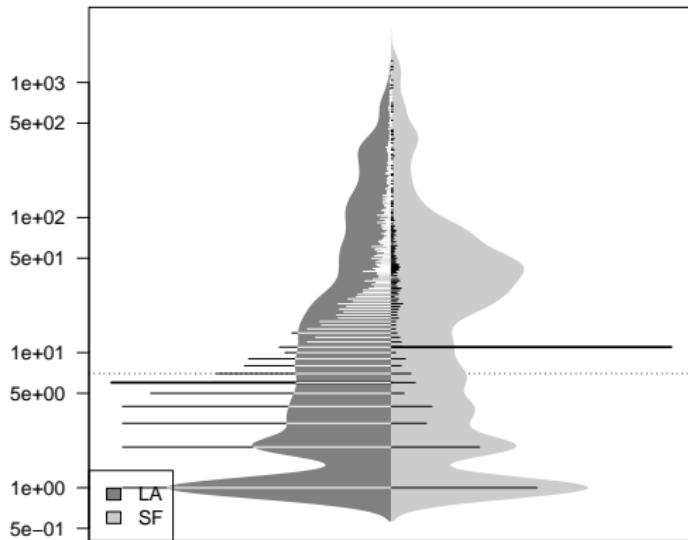
Side-by-side Histogram III

```
1 legend(  
2   x = "bottomleft",  
3   fill=c( grey(0.5), grey(0.8) ),  
4   legend=c( "LA", "SF" )  
5 )  
6 par(op)
```



Histogram

Side-by-side Histogram IV



Scatterplot

Scatterplot (v0.1) |

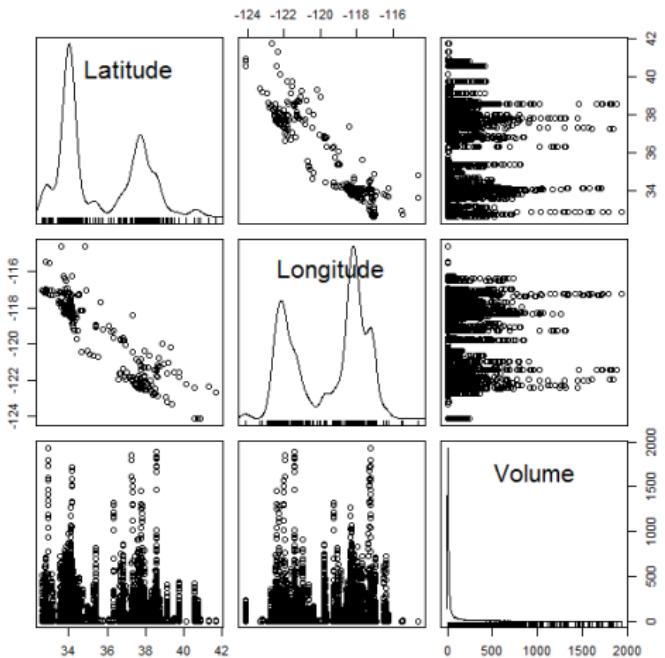
To compare a few variables in the data set at the same time, use `scatterplotMatrix()`:

```
1 # Load package 'car' to use scatterplotMatrix():
2 library(car)
3 scatterplotMatrix(
4   x = df[, c("Latitude", "Longitude", "Volume")],
5   smoother = FALSE,
6   reg.line = FALSE
7 )
```



Scatterplot

Scatterplot (v0.1) II



Scatterplot

Scatterplot (v0.2) |

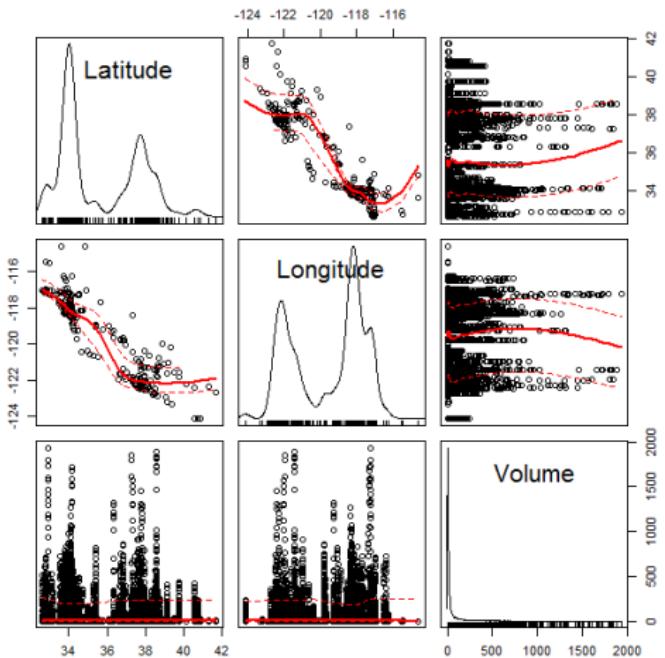
To compare a few variables in the data set and their trends at the same time, modify arguments of `scatterplotMatrix()`:

```
1 ?scatterplotMatrix # for documentation
2 library(car)
3 scatterplotMatrix(
4   x = df[, c("Latitude", "Longitude", "Volume")],
5   reg.line = FALSE
6 )
```



Scatterplot

Scatterplot (v0.2) II



Scatterplot

Scatterplot (v0.3) |

To improve the color scheme, modify arguments of `scatterplotMatrix()`.

```
1 library(car)
2 scatterplotMatrix(
3   x = df[, c("Latitude", "Longitude", "Volume")],
4   reg.line = FALSE,
5   col=c(3,
6     "orangered",
7     rgb(176/256, 196/256, 222/256, alpha=0.5)
8   ),
9   pch=19,
10  lwd=3
11 )
```



Scatterplot

Scatterplot (v0.3) II

You can specify a color in R via:

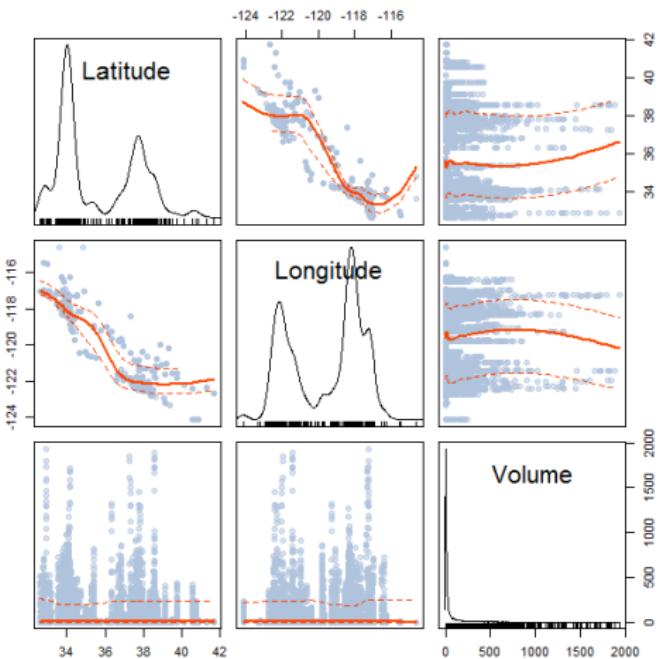
- Number (e.g. 1 = black, 2 = red, etc.)
 - Name (e.g. "black", "red", "dodgerblue")
 - RGB (e.g. $c(0,0,0)$ = black, $c(1,0,0)$ = red, etc.)
 - Please see <http://research.stowers-institute.org/efg/R/Color/Chart/> for color references.

Note the order of color arguments.



Scatterplot

Scatterplot (v0.3) III



Exercise I

In the healthcare data set, after we removed statewide patient admissions, is the largest volume of patients now seen at a Californian hospital a potential outlier? How do you know?



7 Data Set

8 Plots of Counts

9 Time Series Plots

- Univariate Time Series Plots
- Multivariate Time Series Plots
- Exercise II

10 Geographical Plots

Univariate Time Series Plots I

One way to plot one variable at a time (over time), is via `plot()`:

```
1  ### Step 0: Load any required packages:
2  library(dplyr)
3
4  ### Step 1: Get yearly counts across hospitals
and procedures
5  df_summary <- data_clean %>%
6      group_by(Year) %>%
7      summarise(Total.for.Year = sum(Volume, na.rm=TRUE))
```



Univariate Time Series Plots

Univariate Time Series Plots II

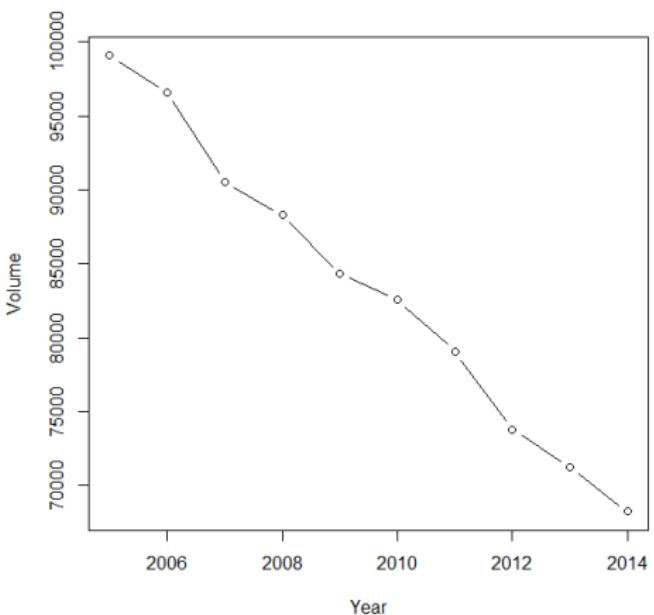
```
1  ### Step 3: Visualize
2  plot(
3    x=df_summary$Year,
4    y=df_summary$Total.for.Year,
5    xlab="Year",
6    ylab="Volume",
7    type="b",
8    main="Yearly Volume for 6 procedures in CA"
9  )
```

What are some limitations of the data set (and hence plot)?



Univariate Time Series Plots

Univariate Time Series Plots III



Multivariate Time Series Plots

Multivariate Time Series Plots: Approach 1 |

To plot more than one variable at a time, over time, you can `ttfamily xyplot()`[5]:

```
1  ### Step 0: Load any required packages
2  library(dplyr)
3  library(lattice)
4
5  ### Step 1: Get data for one procedure
6  df_LA_CABG <- df %>%
7    filter(
8      County == "Los Angeles") &
9      (Procedure == "CABG") &
10     (Volume > 0)
11   )
```



Multivariate Time Series Plots

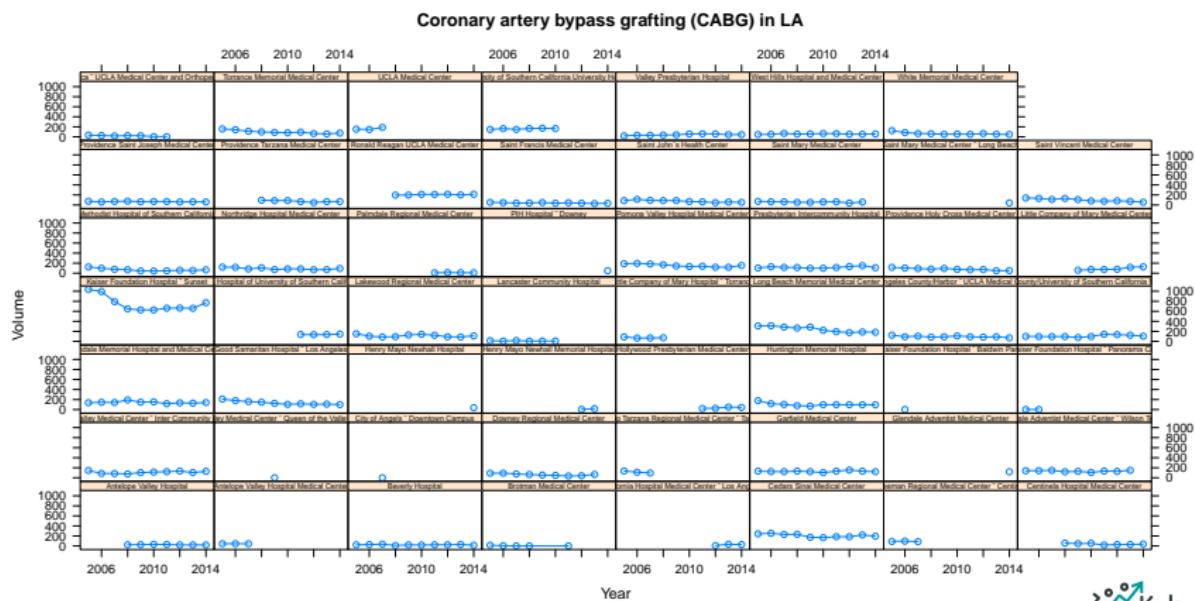
Multivariate Time Series Plots: Approach 1 II

```
1  ### Step 2: Visualize
2  xyplot( Volume ~ Year | Hospital.Name ,
3  data=df_LA_CABG ,
4  par.strip.text=list(cex=0.5) ,
5  type="b" ,
6  main="LA Coronary artery bypass grafting (CABG)"
7  )
```



Multivariate Time Series Plots

Multivariate Time Series Plots: Approach 1 III



Multivariate Time Series Plots

Multivariate Time Series Plots: Approach 2 |

Another way to plot more than one variable at a time, is via `ggplot()`:

```
1  ### Step 0: Load any required packages:  
2  library(dplyr)  
3  library(ggplot2)  
4  
5  ### Step 1: Get data for one hospital:  
6  df_Cedars <- df_clean %>%  
7    filter( Hospital.Name == 'Cedars Sinai  
          Medical Center')
```



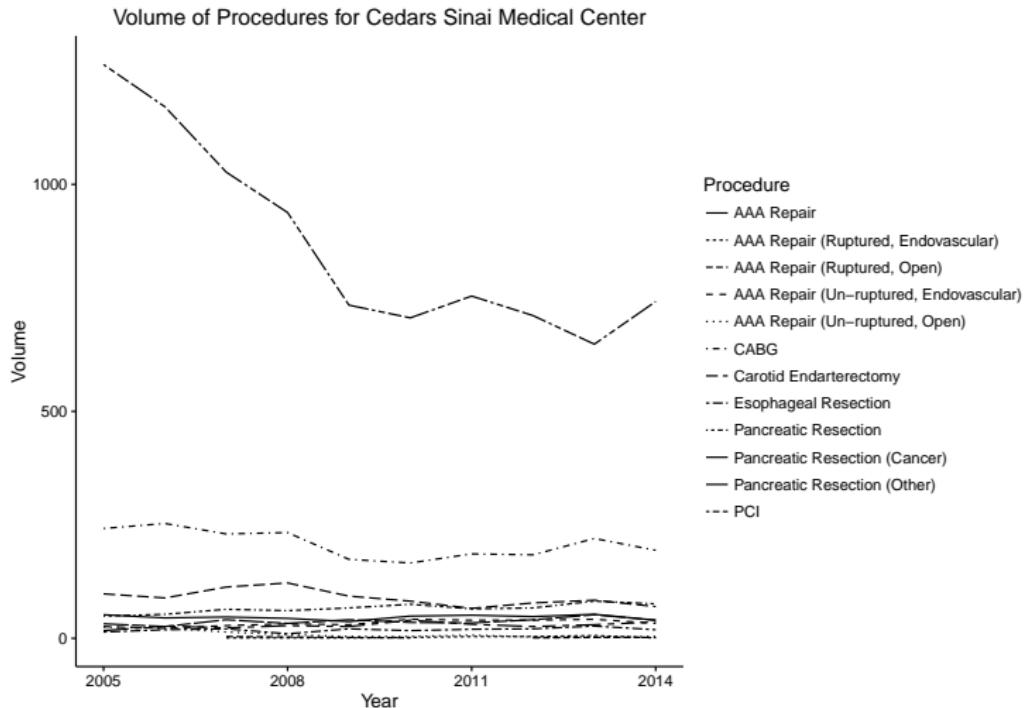
Multivariate Time Series Plots: Approach 2 II

```
1  ### Step 2: Visualize
2  ggplot( data = df_Cedars ) +
3      geom_line( aes(
4          x = Year,
5          y = Volume,
6          linetype = Procedure) ) +
7      scale_x_continuous( breaks=seq(
8          from=2005,
9          to=2014,
10         by=3) ) +
11     theme_classic() +
12     ggttitle( "Volume of Procedures for Cedars
13                 Sinai Medical Center" )
```



Multivariate Time Series Plots

Multivariate Time Series Plots: Approach 2 III



Exercise II

In the healthcare data set, recently what seems to be the most popular procedure in CA?

Hints:

- Should we aggregate the data? If so, to what level?
- What's one way to examine trends in the data?



7 Data Set

8 Plots of Counts

9 Time Series Plots

10 Geographical Plots

- Plots using Maps
- Interactive Geographical Plots
- Exercise III



Geographic Maps (v0.1) I

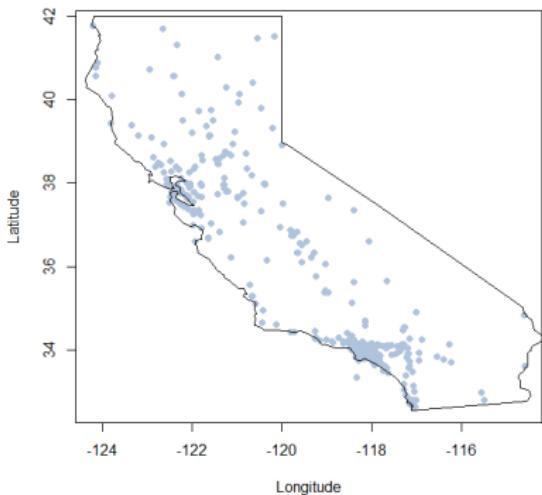
To overlay a map to a plot containing latitude and longitude, load the package `maps`:

```
1 library(maps)
2 plot(
3   x = df_clean$Longitude,
4   y = df_clean$Latitude,
5   xlab = "Longitude",
6   ylab = "Latitude",
7   pch = 16,
8   col = rgb(176/256, 196/256, 222/256, alpha=0.5)
9 )
10 map("state", "california", add=TRUE)
```



Plots using Maps

Geographic Maps (v0.1) II



What are some limitations of the plot?



Geographic Maps (v0.2) |

To include information on volume of procedure, use `cex` argument of the `plot()` function:

```
1  #### Step 1: Aggregate data to be at hospital level:  
2  df_hosp <- data_clean %>%  
3  group_by( Latitude, Longitude ) %>%  
4  summarise( Total.for.Hospital = sum(Volume, na.rm=  
      TRUE) )  
5  
6  #### Step 2: Recode 'Volume' to have 2 categories  
    only: high ( >100 cases ) and low ( <= 100 cases  
    ):  
7  df_hosp$volume_ind <- ifelse(  
8  test = df_hosp$Total.for.Hospital > 100,  
9  yes = 2,  
10 no = 1 )
```



Geographic Maps (v0.2) II

```
1  ### Step 3: Plot
2  plot(
3    x = df_hosp$Longitude,
4    y = df_hosp$Latitude,
5    pch = 19,
6    cex = df_hosp$volume_ind,
7    col = df_hosp$volume_ind,
8    xlab = "Longitude",
9    ylab = "Latitude",
10   main="Indicator of overall volume between
11   2005-2014"
11 )
12 map("state", "california", add=TRUE)
```



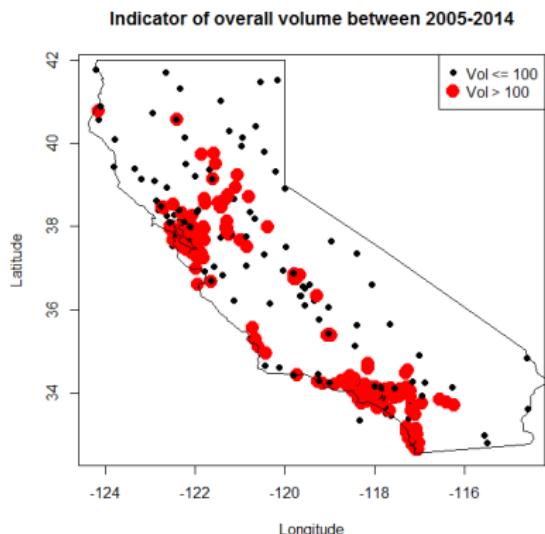
Plots using Maps

Geographic Maps (v0.2) III

```
1  ### Step 4: Add legend
2  legend("topright",
3    pch = 19,
4    pt.cex = 1:2,
5    col = 1:2,
6    c("Vol <= 100", "Vol > 100")
7  )
```



Geographic Maps (v0.2) IV



What are some limitations of the plot? (Exercise.)

Geographic Maps (v0.3) |

Another way to visualize geographical information is via shapefiles.
Please download shapefiles for California for 2014 from https://www.census.gov/geo/maps-data/data/cbf/cbf_tracts.html
and unzip the folder.

```
1 ### Step 1: Load in required packages
2 library(sp)
3 library(rgdal)
```



Geographic Maps (v0.3) II

```
1  ### Step 2: Read in shapefile and add latitude  
   and longitude coordinates to it:  
2  tracts = spTransform(  
3      readOGR(  
4          file.path("cb_2014_06_tract  
5              _500k"),  
6          layer = "cb_2014_06_tract_500k  
7              "  
8          ),  
9          CRS("+proj=longlat +datum=WGS84")  
10     )  
11  
10  ### Step 3: Visualize  
11  plot(tracts)
```



Geographic Maps (v0.3) III



Plot of the shapefile tracts.



Geographic Maps (v0.3) IV

Now let's add the locations of hospitals to the map [12]:

```
1  ### Step 1: Load required packages
2  library("rgdal")
3  library("maptools")
4  library("ggplot2")
5  library("plyr")
6
7  ### Step 2: Preprocess data ahead of plotting
8  tracts@data$id = rownames(tracts@data)
9  tracts.points = fortify(tracts, region="id")
10 tracts.df = join(
11   tracts.points,
12   tracts@data,
13   by="id"
14 )
```



Plots using Maps

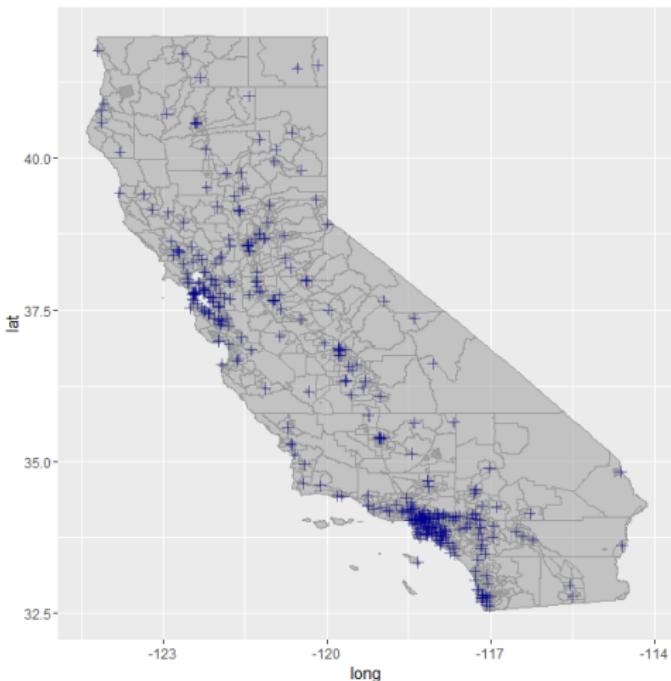
Geographic Maps (v0.3) V

```
1  ### Step 3: Visualize
2  ggplot() +
3    geom_polygon(data = tracts.df,
4                  aes(x = long, y = lat, group =
5                      group),
5                  fill = grey(0.6),
6                  color = grey(0.6),
7                  alpha = 0.5
8                  ) +
9    geom_point(data = df_hosp,
10               aes(x = Longitude, y = Latitude),
11               color = "blue4",
12               alpha = 0.5,
13               shape = 3,
14               size = 2 )
```



Plots using Maps

Geographic Maps (v0.3) VI



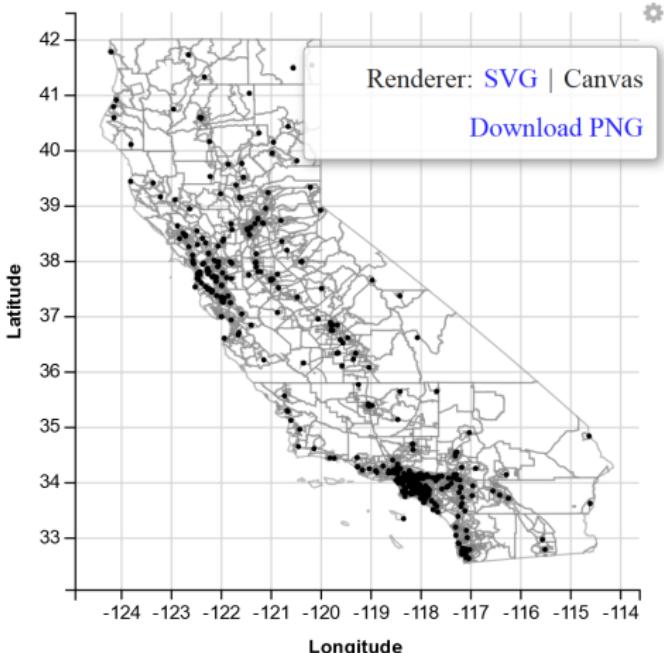
Geographical Plots via 'ggvis' |

One way to make high resolution and interactive images is via 'ggvis' [13]:

```
1 # Please note, it may take some time to load as the
   shapefile is high resolution:
2 library(ggvis)
3 tracts.points %>%
4   ggvis(~long, ~lat) %>%
5   group_by(group, id) %>%
6   layer_paths(strokeOpacity:=0.5, stroke:=grey(0.5))
7   %>%
8   layer_points(data=df_hosp, x=~Longitude, y=
    ~Latitude, size:=8) %>%
9   hide_legend("fill") %>%
  set_options(width=400, height=400, keep_aspect=
    TRUE)
```



Geographical Plots via 'ggvis' !!



Geographical Plots via 'ggvis' III

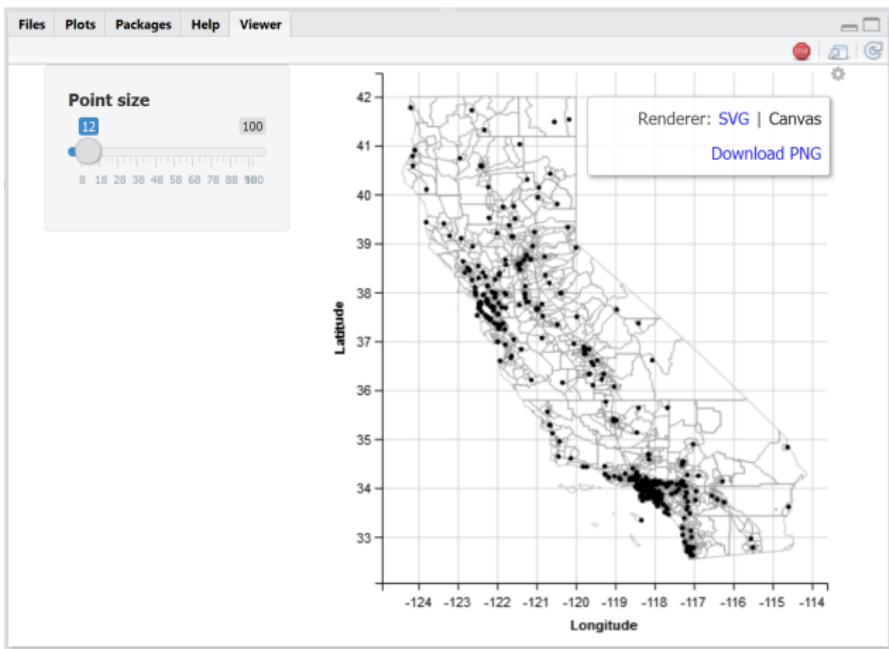
Now we can add (basic) interactivity by resizing the hospital locations:

```
1 library(ggvis)
2 tracts.points %>%
3   ggvis(~long, ~lat) %>%
4   group_by(group, id) %>%
5   layer_paths(
6     strokeOpacity:=0.5,
7     stroke:=grey(0.5)) %>%
8   layer_points(
9     data=df_hosp,
10    x=~Longitude,
11    y=~Latitude,
12    size:=input_slider(8, 100, value = 12, label='Point size')) %>%
13   hide_legend("fill") %>%
14   set_options(width=400,
15    height=400,
16    keep_aspect=TRUE)
```



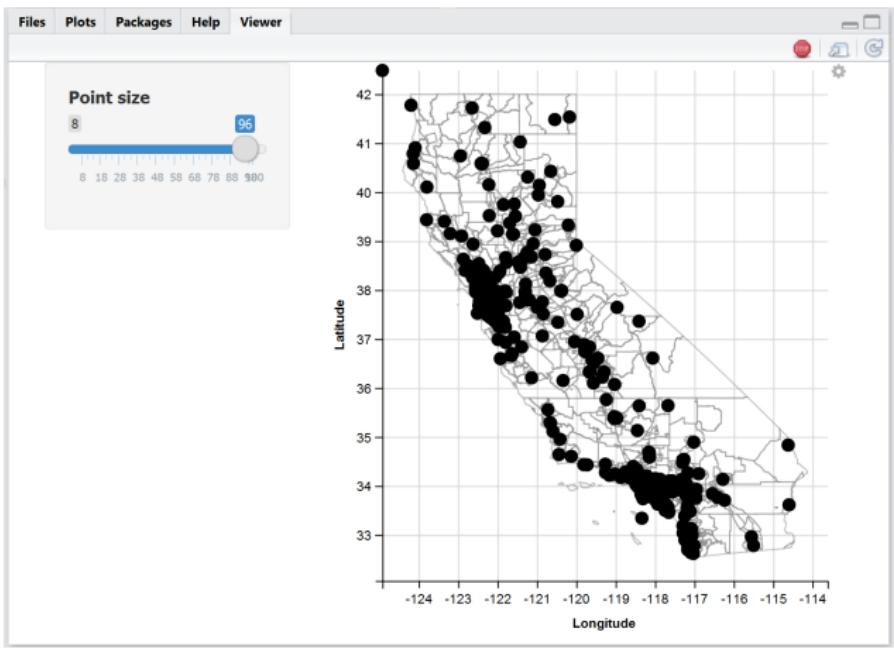
Interactive Geographical Plots

Geographical Plots via 'ggvis' IV



Interactive Geographical Plots

Geographical Plots via 'ggvis' V



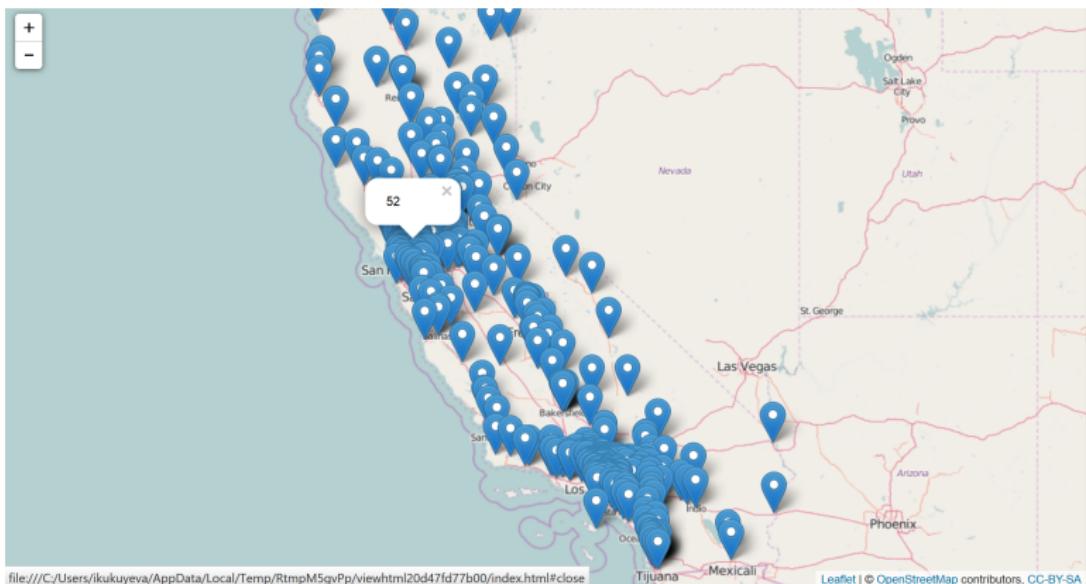
Geographical Plots via 'leaflet' I

Another way to make interactive images is via 'leaflet':

```
1 library(leaflet)
2 leaflet(data = df_hosp) %>%
3   addTiles() %>%
4   addMarkers(
5     ~Longitude ,
6     ~Latitude ,
7     popup = ~as.character(Total.for.Hospital)
8   )
```



Geographical Plots via 'leaflet' II



Exercise III

Exercise III

In the healthcare data set, what's one way to showcase higher volume procedures in the plot?



Part IV

Working with R



11 Common Bugs and Fixes

- Syntax Error
- Trailing +
- Error When Performing Operations
- Error in Calling an Object
- Silent Errors

12 Where to go from here?

13 Getting R Help

14 Online Resources for R

15 References





Error: syntax error

Possible causes:

- Misspelling the object's name
- Including a "+" when copying code from console/website, etc.
- Having an extra parenthesis at the end of a function
- Having an extra bracket when subsetting



Trailing +

Possible causes:

- Not closing a function call with a parenthesis
- Not closing brackets when subsetting
- Not closing a function you wrote with a squiggly brace





Error When Performing Operations

Error in ... : requires numeric matrix/vector arguments

Possible causes:

- ① Objects are data frames, not matrices
- ② Elements of the vectors are characters

Possible solutions:

- ① Coerce (a copy of) the data set to be a matrix, with the `as.matrix()` command
- ② Coerce (a copy of) the vector to have numeric entries, with the `as.numeric()` command





Error: ... object not found

Possible causes:

- ➊ Misspelling the object's name
- ➋ Package containing the object has not been loaded





Silent Errors

Most common silent errors:

- ① (Inadvertently) Creating a data set with no rows or columns.
- ② (Inadvertently) Recycling (and padding) of entries in a variable with a smaller number of observations than the one it is compared to.

Possible solutions:

- ① Always check the dimensionality of the data set after subsetting.
- ② Always check the lengths of variables ahead of comparison, especially if subsetting just took place.

For more caveats and solutions, read the "R Inferno":

http://www.burns-stat.com/pages/Tutor/R_inferno.pdf



11 Common Bugs and Fixes

12 Where to go from here?

13 Getting R Help

14 Online Resources for R

15 References



Where to go from here?

Part 1: Explore other visualizations

Other exploratory visualizations in R that we didn't get to cover:

- Sankey graphs
- (Interactive) dashboards via R package 'shiny'
- Creating animations with R
- ...



Where to go from here?

Part 2: Familiarize yourself with R (as it relates to EDA)

Other data-related topics that we didn't get to cover:

- Variety of ways to aggregate data
- Getting data via an API (e.g. meetup, yelp, etc.)
- Developing reproducible reports (via 'knitr' package)
- ...



Where to go from here?

Part 3: Familiarize yourself with HTML/JS/d3/…

Other visualization topics that we didn't get to cover:

- Ability to customize design of interactive graphics (e.g.
[http://datascience.la/
interactive-visualizations-from-r-using-rcharts/](http://datascience.la/interactive-visualizations-from-r-using-rcharts/)
and [http://www.slideshare.net/f0008/
javascriptbased-visualization-in-r](http://www.slideshare.net/f0008/javascriptbased-visualization-in-r))
- Ability to customize design of interactive dashboards (e.g.
<http://shiny.rstudio.com/articles/>)
- Ability to customize design of reproducible reports (e.g.
[http://shiny.rstudio.com/gallery/
download-knitr-reports.html](http://shiny.rstudio.com/gallery/download-knitr-reports.html))
- ...



Where to go from here?

Part 4: Familiarize yourself with data analysis models

We did not cover model building as a way to explain relationships in the data, such as:

- Different types of regression models for modeling numerical data
- Different types of decision trees for modeling numerical data
- Different types of models for analyzing text/image/video/audio data
- ...

Please see the 'Online Resources for R' section (below) for more information.



Where to go from here?

Part 5: Familiarize yourself with 'best practices'

We did not explicitly cover any best practices such as:

- Commenting code
- Clear variable names
- Version control
- ...

Please see the following for more information:

- Google's R style guide:
<https://google.github.io/styleguide/Rguide.xml>
- Joel's Test for writing better code: <http://www.joelonsoftware.com/articles/fog0000000043.html>
- Iliinsky and Steele's book on *Designing Data Visualizations*
- ...



Where to go from here?

Part 6: Connect with Other Data Scientists

- meetup.com (e.g. R Users' Group, Data Viz LA, etc.)
- LinkedIn
- Conferences: userR 2016, Big Data LA
-



11 Common Bugs and Fixes

12 Where to go from here?

13 Getting R Help

14 Online Resources for R

15 References



R Help: Approach 1

For help with any function in R, add a question mark before the function name to see the documentation (which includes explanation of the function's arguments/inputs, function outputs and example use cases).

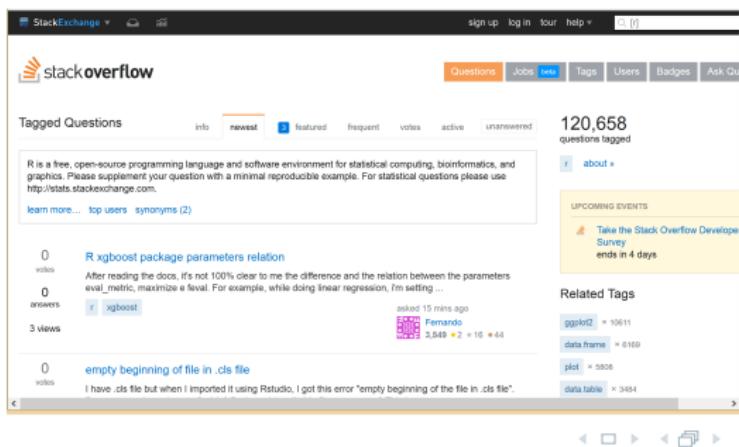
1 ?plot

The screenshot shows the R Documentation window for the `plot` function. The title bar says "R Documentation". The main content area has a yellow header bar with the function name "plot [graphics]" and a sub-header "Generic X-Y Plotting". Below this, the "Description" section states: "Generic function for plotting of R objects. For more details about the graphical parameter arguments, see [par](#)". It notes that for simple scatter plots, `plot.default` will be used. The "Usage" section shows the command `plot(x, y, ...)`. The "Arguments" section describes the parameters `x` and `y`, and mentions additional arguments like `type`. A note at the bottom indicates that `plot` is a generic function.



R Help: Approaches 2 and 3

- For help with any function in R, search answers on StackOverflow (SO).
- For help with any function in R, when all else fails, ask a question on StackOverflow. Don't forget to follow the SO tips: <http://stackoverflow.com/help/how-to-ask>



11 Common Bugs and Fixes

12 Where to go from here?

13 Getting R Help

14 Online Resources for R

15 References



Online Resources for R I

Download R: <http://cran.stat.ucla.edu/>

Download RStudio: <https://www.rstudio.com/>

R Reference Card:

<http://cran.r-project.org/doc/contrib/Short-refcard.pdf>

R Graphics Gallery:

<http://research.stowers-institute.org/efg/R/>

R Graph Gallery: <http://addictedtor.free.fr/graphiques/>

Stackoverflow: <http://stackoverflow.com/tags/r/info>

Blogs: <http://www.r-bloggers.com/>

JSS: <https://www.jstatsoft.org/index>



Online Resources for R II

More R tutorials:

Courses: Code School, Coursera, DataCamp, DataRobot, edX, RStudio, swirl

IK: <http://www.KukuyevaConsulting.com/tutorials>

UCLA IDRE: <http://www.ats.ucla.edu/stat/r/>

UCLA SCC: <http://scc.stat.ucla.edu/mini-courses/>



11 Common Bugs and Fixes

12 Where to go from here?

13 Getting R Help

14 Online Resources for R

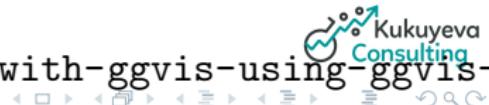
15 References



1. <http://adv-r.had.co.nz/>
2. [http://www.sixhat.net/
how-to-plot-multiple-data-series-with-ggplot.html](http://www.sixhat.net/how-to-plot-multiple-data-series-with-ggplot.html)
3. [http://stackoverflow.com/questions/17584248/
exact-axis-ticks-and-labels-in-r-lattice-xypot](http://stackoverflow.com/questions/17584248/exact-axis-ticks-and-labels-in-r-lattice-xypot)
4. [https://rstudio-pubs-static.s3.amazonaws.com/
3364_d1a578f521174152b46b19d0c83cbe7e.html](https://rstudio-pubs-static.s3.amazonaws.com/3364_d1a578f521174152b46b19d0c83cbe7e.html)
5. www.jstatsoft.org/v25/c01/paper
6. [http://www.kdnuggets.com/2015/05/
r-vs-python-data-science.html](http://www.kdnuggets.com/2015/05/r-vs-python-data-science.html)
7. <http://flowingdata.com/2014/02/05/where-people-run/>



8. https://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919
9. Iliinsky, N. and Steele, J. (2011) *Designing Data Visualizations* Sebastopol, CA: O'Reilly.
10. http://www.pixel-push.com/2013/09/24/ultimate-infographic-resource-kits-for-designers/
11. Heiberger, R. M. (2016, February). Design of Not-Simple Graphs. Talk presented at the meeting of the American Statistical Association Conference on Statistical Practice, San Diego, CA.
12. https://github.com/hadley/ggplot2/wiki/plotting-polygon-shapefiles
13. http://www.r-bloggers.com/making-static-interactive-maps-with-ggvis-using-ggvis-



14. <http://ggvis.rstudio.com/interactivity.html>
15. <https://github.com/smbache/magrittr>



Thank you.
Any questions?

