

---

# Introduction to Visualizations with R

Irina Kukuyeva, Ph.D.  
[www.KukuyevaConsulting.com](http://www.KukuyevaConsulting.com)

---

March 3, 2016



# Tutorial Outline

Preliminaries

Introduction to R

Visualizations

Working with R



# Part I

## Preliminaries



## 1 Motivation

## 2 Why R?

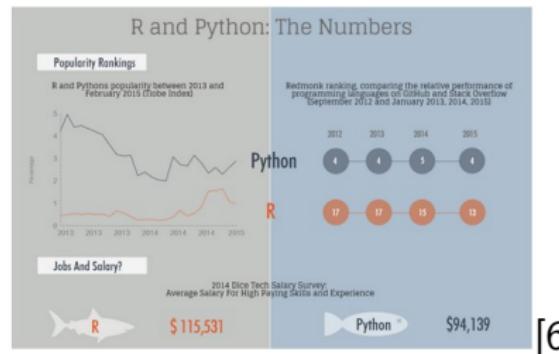
## 3 Why visualize?

## 4 Tips for Visualizations



## Motivation

- 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



## 1 Motivation

## 2 Why R?

## 3 Why visualize?

## 4 Tips for Visualizations



## 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).



## 1 Motivation

## 2 Why R?

## 3 Why visualize?

## 4 Tips for Visualizations



## Why visualize?

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



[8]

## 1 Motivation

## 2 Why R?

## 3 Why visualize?

## 4 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
- Color and font choices
- Add text/labels to figure



[7]



## Part II

# Introduction to R



## 5 Installing R and RStudio

## 6 Overview of R

## 7 Working with Data

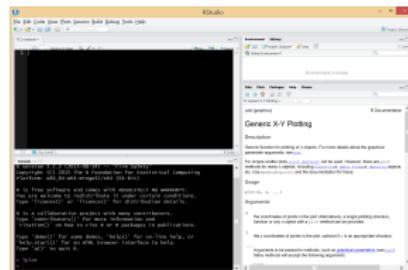
## 8 Adding Functionality to Base R



# 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).



## 5 Installing R and RStudio

## 6 Overview of R

- Creating Objects
- (Very) Brief Overview of Functions

## 7 Working with Data

## 8 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 in R

Objects can be created in different ways, via:

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



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

## 5 Installing R and RStudio

## 6 Overview of R

## 7 Working with Data

- Reading Data from File
- Working with Data Frames

## 8 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  data <- read.table(
6    file=filepath,
7    header=TRUE,
8    sep=","
9  )
```



## 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 data
```

	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(data, 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(data, 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(data)
```

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

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

```
1 dim(data)
```

```
[1] 8 6
```

- See the first 3 entries for variable 'gender' via `head()`:

```
1 head(data$gender)
```

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



## Working with Data Frames

# Working with Data Frames V

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

```
1 summary(data)
```

```
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
```



## 5 Installing R and RStudio

## 6 Overview of R

## 7 Working with Data

## 8 Adding Functionality to Base R



# Adding Functionality to Base R

- Base R is what you download off CRAN  
[www.cran.r-project.org](http://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



## 9 Data Set

## 10 Summary Plots

## 11 Time Series Plots

## 12 Geographical Plots



# Data Set for Tutorial I

Data set for this tutorial comes from State of California. Please download it 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)

To read it into R, type:

```
1 data <- read.csv(  
2   file = "Number_of_Selected_Inpatient_Medical  
      _Procedures__California__Hospitals__2005–2014.  
      csv",  
3   stringsAsFactors = FALSE  
4 )
```



# Data Set for Tutorial II

**CHHS Open Data**

Sign Up Sign In Accessibility

Home Catalog Tutorials Developers About

Number of Selected Inpatient Medical Procedures, California Hospitals, 2005-2014

This dataset contains the number (volume) of 6 selected inpatient procedures (Esophageal Resection, Pancreatic Resection, Abdominal Aortic Aneurysm (AAA) Repairs, Endovascular Endarterectomy, Coronary Artery Bypass Graft Surgery, Pneumotuberculous Coronary Interventions) performed in California hospitals from 2005 to 2014. This dataset does not include procedures performed in outpatient settings.

Year	County	Hospital Name	OSHPDID	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	Scripps Memorial Hospital La Jolla	106370771	CABG	367	-117.22279	32.88506	(3)
3 2014	Fresno	Adventist Medical Center Reedley	106100797	Pancreatic Resection	.	-119.45145	36.60789	(3)
4 2014	Fresno	Adventist Medical Center Reedley	106100797	AAA Repair (Rupture)	.	-119.45145	36.60789	(3)
5 2014	Orange	Saint Jude Medical Center	106301342	Esophageal Resection	1	-117.92851	33.89349	(3)
6 2014	San Diego	Scripps Memorial Hospital La Jolla	106370771	AAA Repair (Unruptured)	9	-117.22279	32.88506	(3)
7 2014	San Diego	Scripps Memorial Hospital La Jolla	106370771	AAA Repair (Rupture)	2	-117.22279	32.88506	(3)
8 2014	San Diego	Scripps Memorial Hospital La Jolla	106370771	AAA Repair (Rupture)	.	-117.22279	32.88506	(3)
9 2014	San Diego	Scripps Memorial Hospital La Jolla	106370771	AAA Repair	13	-117.22279	32.88506	(3)
10 2014	San Diego	Scripps Memorial Hospital La Jolla	106370771	Pancreatic Resection	1	-117.22279	32.88506	(3)
11 2014	Orange	Saint Jude Medical Center	106301342	PCI	235	-117.92851	33.89349	(3)
12 2014	Fresno	Adventist Medical Center Reedley	106100797	AAA Repair (Rupture)	450	-117.92851	36.60789	(3)

Manage More Views Filter Visualize Export Discuss Embed About

Home Catalog Terms of Use Privacy Policy Contact Us © 2015 California Health & Human Services Agency

## 9 Data Set

## 10 Summary Plots

- Scatterplot
- Histogram
- Exercise I

## 11 Time Series Plots

## 12 Geographical Plots

## Scatterplot

## Scatterplot (v0.1) |

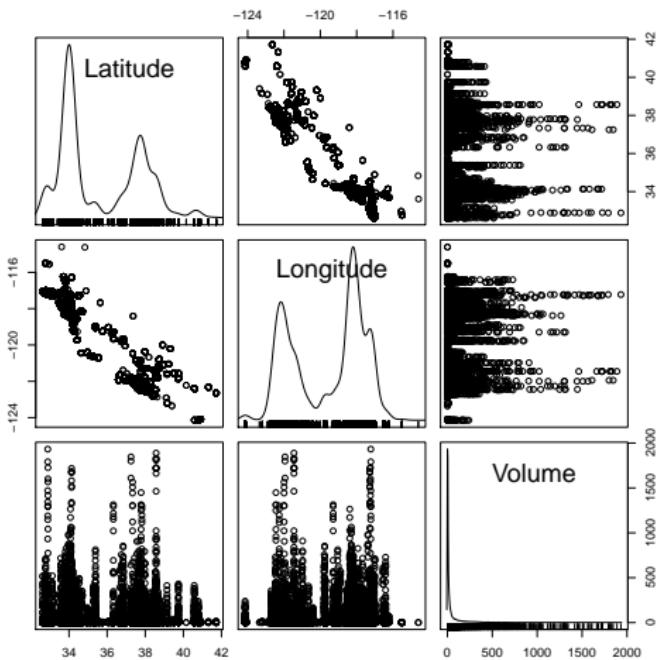
To get a quick peak into the numerical variables in the data  
`scatterplotMatrix()`:

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



## Scatterplot

## Scatterplot (v0.1) II



## Scatterplot

# Scatterplot (v0.2) |

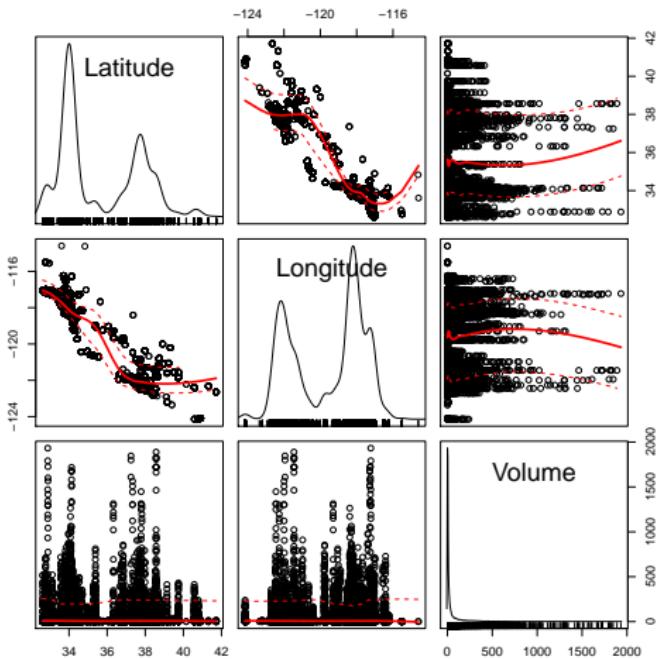
To get a quick peak into the numerical variables and the trends in the data, modify arguments of `scatterplotMatrix()`:

```
1 ?scatterplotMatrix # for documentation
2 library(car)
3 scatterplotMatrix(
4   x = data[, 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 = data[, 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

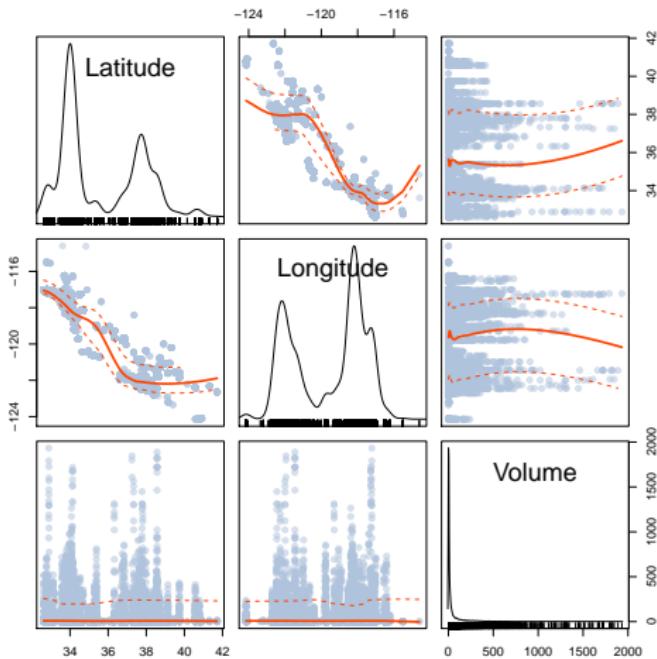
You can specify a color in R via:

- 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

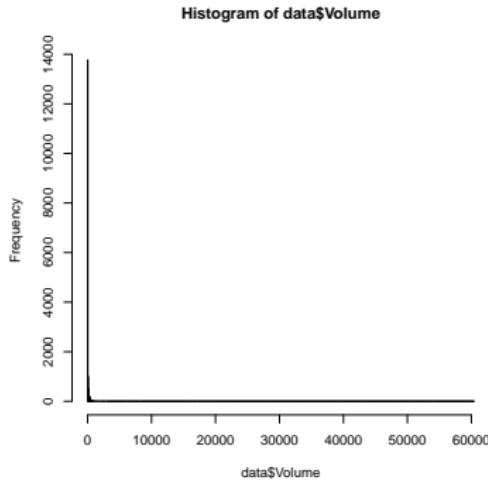


## Histogram

# Histogram I

To see the distribution of one variable, use `hist()`:

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



## Histogram

# Histogram II

Let's examine the potential outlier:

```
1 subset(  
2   x = data,  
3   Volume == max(Volume, na.rm=TRUE)  
4 )
```

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

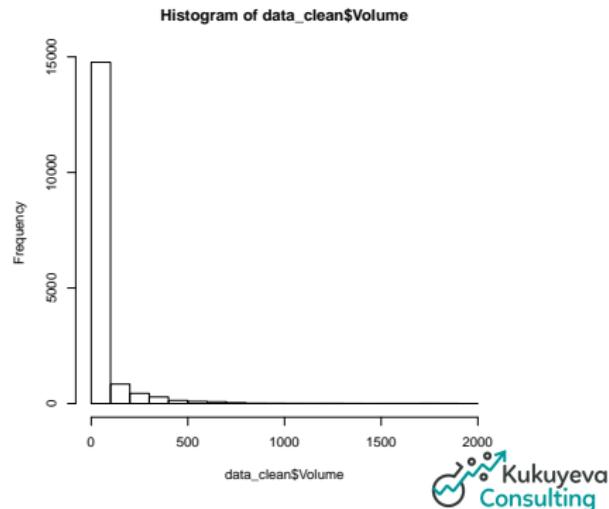


## Histogram

# Histogram III

Entry seems to be an aggregate number, not county/year level. To remove the outlier:

```
1 data_clean <- subset(
2   x = data,
3   County != "STATEWIDE"
4 )
5 hist(data_clean$Volume)
6 # Is anything 'off' now?  
# (Exercise.)
```



# 'Nicer' Histogram I

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

```
1 library(beanplot)
2 # Subset the data to LA and SF:
3 data_LA_SF <- subset(
4   x = data_clean,
5   County %in% c("Los Angeles", "San Francisco")
6 )
7
8 # Orient y-axis labels to be more readable:
9 op <- par(las=2)
10 beanplot(
11   Volume ~ as.factor(County),
12   data = data_LA_SF,
13   xlab = "",
14   log = "y",
```



## Histogram

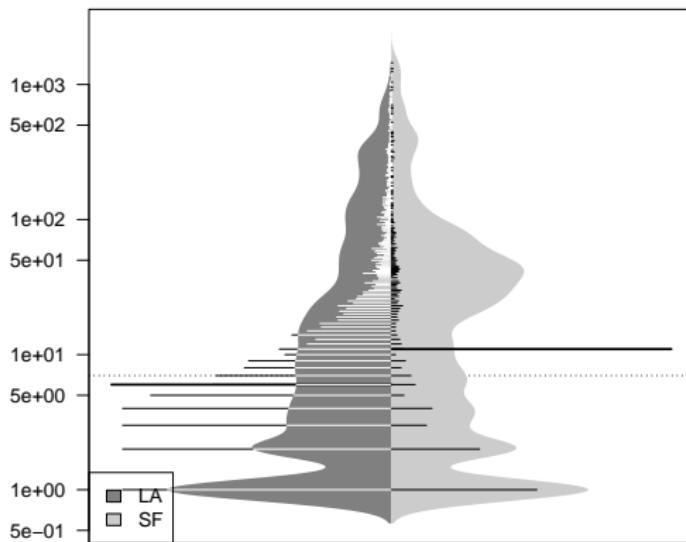
## 'Nicer' Histogram II

```
15     side = "both",
16     col = list( c( grey(0.5), "white"), grey(0.8) ),
17     border = NA,
18     overallline = "median",
19     ll = 0.005,
20     show.names=FALSE
21   )
22
23 legend(
24   x = "bottomleft",
25   fill=c( grey(0.5), grey(0.8) ),
26   legend=c( "LA", "SF" )
27 )
28 par(op)
```



## Histogram

## 'Nicer' Histogram 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?



**9** Data Set**10** Summary Plots**11** Time Series Plots

- Multivariate Plots: Approach 1
- Multivariate Plots: Approach 2
- Multivariate Plots: Approach 3
- Multivariate Plots: Approach 5
- Exercise II

**12** Geographical Plots

# Multivariate Time Series Plots (v0.1) |

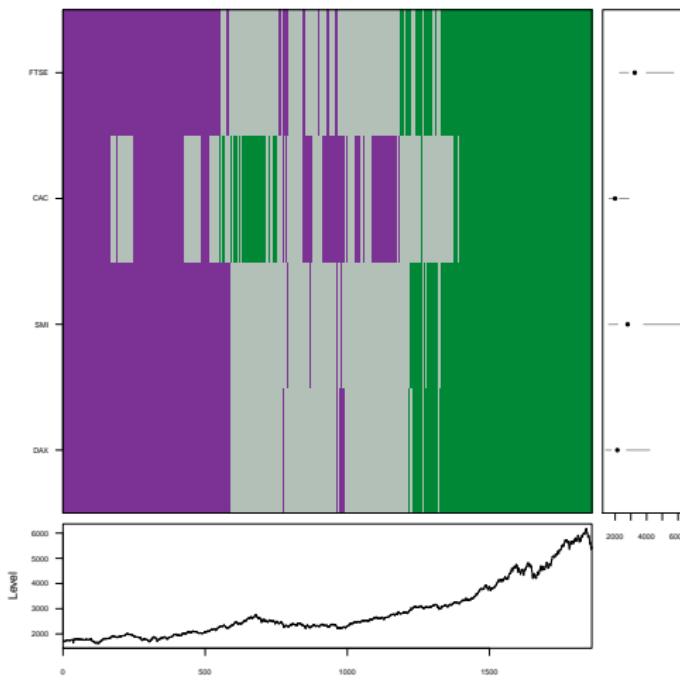
To plot more than variables one at a time, use `mvtspplot()`[5]:

```
1 library(mvtspplot)
2 # Data = Daily Closing of Stock Indices:
3 data(EuStockMarkets)
4 # Purple=low, grey=medium,
5 # green=high, white=missing values:
6 mvtspplot(EuStockMarkets)
```



## Multivariate Plots: Approach 1

## Multivariate Time Series Plots (v0.1) II



# Multivariate Time Series Plots (v0.2) |

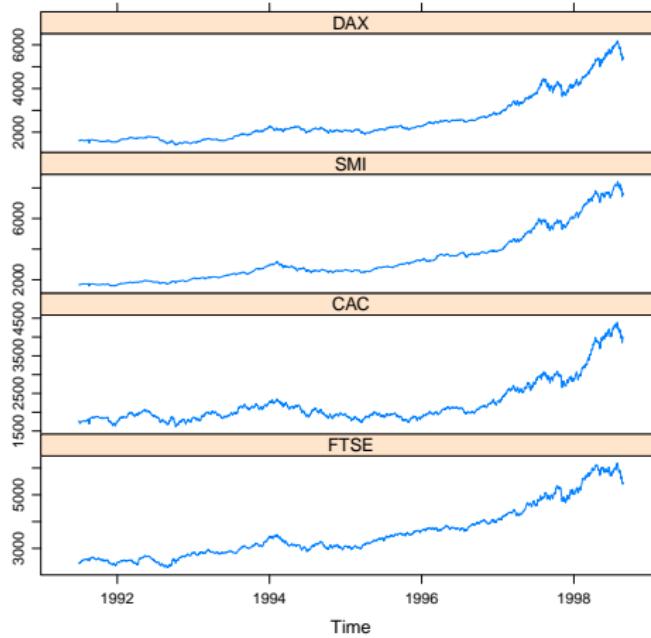
Another way to plot more than one variable at a time, is via `xyplot()`:

```
1 library(lattice)
2 data(EuStockMarkets)
3 xyplot(EuStockMarkets)
```



## Multivariate Plots: Approach 2

## Multivariate Time Series Plots (v0.2) II



# Multivariate Time Series Plots (v0.3) I

To customize the output, modify controls of `xyplot()`:

```
1 library(lattice)
2 library(reshape2)
3 data(EuStockMarkets)
4 # Stack data per [2]:
5 df = cbind.data.frame(
6   "id"=1:nrow(EuStockMarkets),
7   EuStockMarkets
8 )
9 df_stacked = melt(
10  df,
11  id="id"
12 )
13 # Adding labels per [3]:
```



## Multivariate Plots: Approach 2

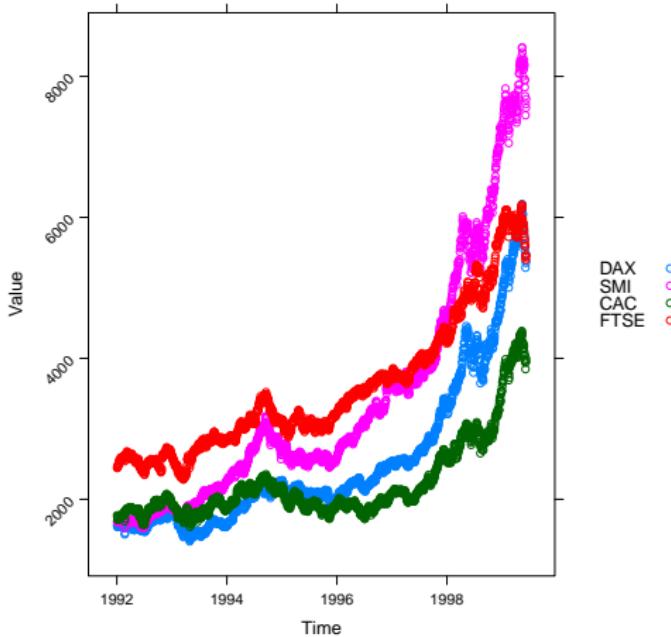
## Multivariate Time Series Plots (v0.3) II

```
14 xyplot( df_stacked$value ~ df_stacked$id ,  
15 groups=df_stacked$variable ,  
16 auto.key = list(space = "right") ,  
17 scales = list(y = list(rot = 45) ,  
18 x = list(  
19 at=c(0,500, 1000, 1500) ,  
20 labels=seq(from=1992, to=1998,  
21 by=2)  
22 )  
23 xlab="Time" ,  
24 ylab="Value"  
25 )
```



## Multivariate Plots: Approach 2

## Multivariate Time Series Plots (v0.3) III



# Multivariate Time Series Plots (v0.4) |

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

```
1 library(ggplot2)
2 library(reshape2)
3 data(EuStockMarkets)
4 df = cbind.data.frame(
5   "id"=1:nrow(EuStockMarkets),
6   EuStockMarkets
7 )
8 df_stacked = melt(
9   df,
10  id="id"
11 )
12 # Adding labels per [4]:
```



## Multivariate Plots: Approach 3

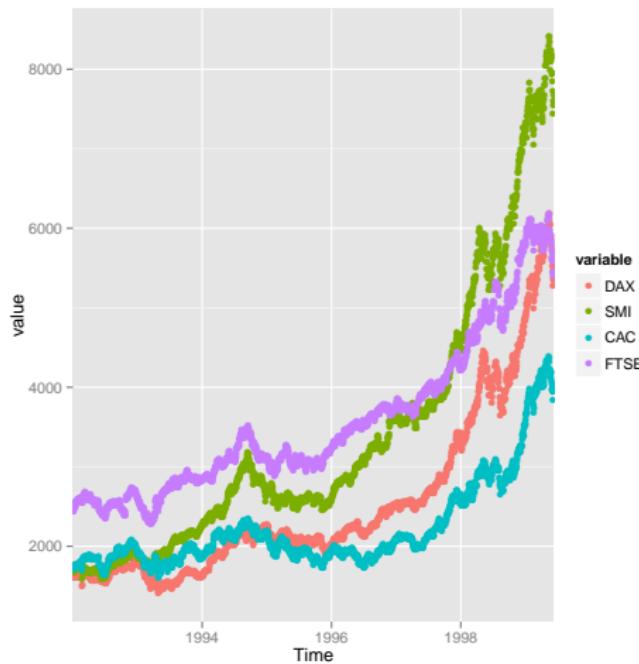
## Multivariate Time Series Plots (v0.4) II

```
13 ggplot(  
14   data = df_stacked,  
15   aes(x = id, y = value, color = variable)  
16   ) +  
17   geom_point() +  
18   scale_x_discrete(name="Time",  
19     breaks=seq(from=0, to=1500, by=500),  
20     labels=seq(from=1992, to=1998, by=2)  
21   )
```



## Multivariate Plots: Approach 3

## Multivariate Time Series Plots (v0.4) III



Multivariate Plots: Approach 5

# Multivariate Time Series Plots (v0.5) |



## Exercise II

Analyze the approval ratings of the President of the United States in the data set `presidents`. What stands out and why?

Hints:

- Load the data set `presidents`.
- Convert the data set to be in matrix format, one row per year.
- Plot the data set.
- Examine the result.



**9** Data Set**10** Summary Plots**11** Time Series Plots**12** Geographical Plots

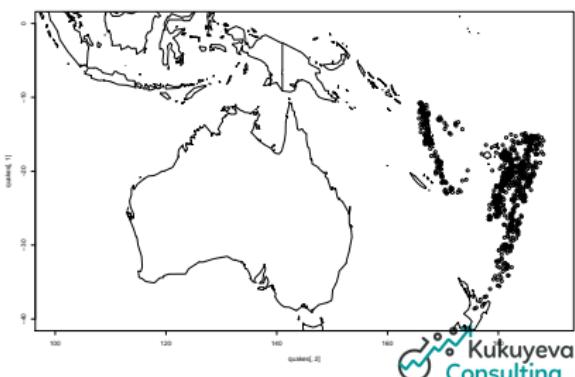
- Maps
- Projection Maps
- Exercise III

# Geographic Maps

## Map of Fiji Earthquakes Since 1964

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

```
1 data( quakes )
2 library( maps )
3 plot(
4           quakes$long ,
5           quakes$lat ,
6           xlim=c(100, 190) ,
7           ylim=c(-40, 0)
8         )
9 map("world" , add=TRUE)
```



# Geographic Maps I

## Map of Fiji Earthquakes Since 1964

To include information on magnitude of earthquake, use cex argument of the plot() function:

```
1 # Step 1: Recode magnitude to have 3
  categories only
2 ind<-which(quakes$mag<5)
3 ind2<-which(quakes$mag<6 & quakes$mag >=5)
4 ind3<-which(quakes$mag>=6)
5 color<-rep(NA, length(quakes$mag))
6 color[ind]<-1; color[ind2]<-2; color[ind3]<-3
7
8
9
10
```



# Geographic Maps II

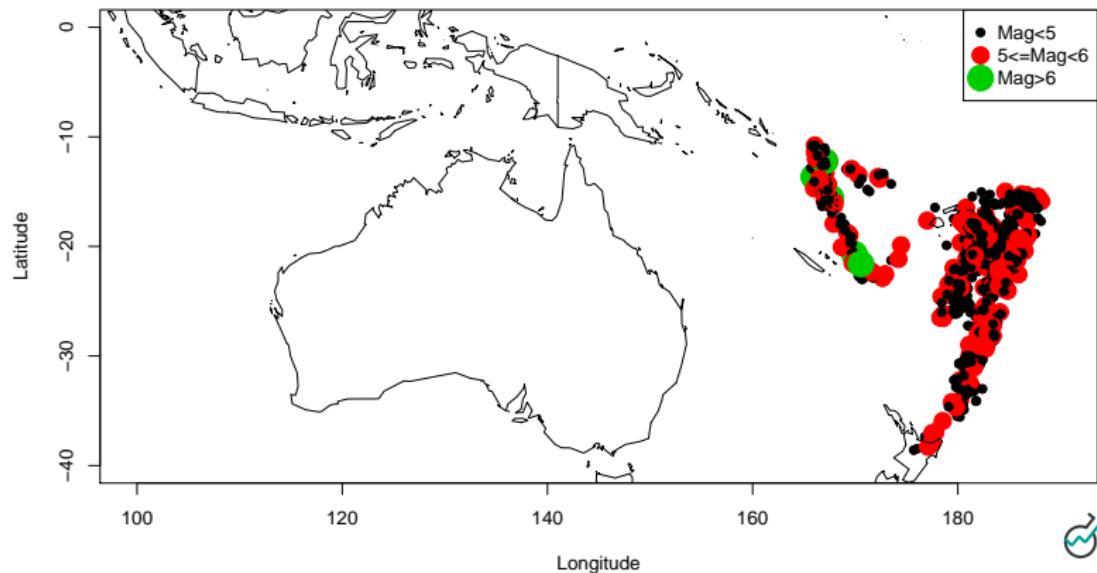
## Map of Fiji Earthquakes Since 1964

```
11  # Step 2: Plot
12  plot(quakes$long, quakes$lat, cex=color, pch
13    =19, col=color, xlim=c(100, 190), ylim=
14    c(-40,0), xlab="Longitude", ylab="Latitude
15    ")
16  legend("topright", pch=19, col=1:3, c("Mag<5",
17    "5<=Mag<6", "Mag>6"), pt.cex=1:3)
18  map("world", add=TRUE)
```



# Geographic Maps III

## Map of Fiji Earthquakes Since 1964



# Projection Maps I

Map of Fiji Earthquakes Since 1964

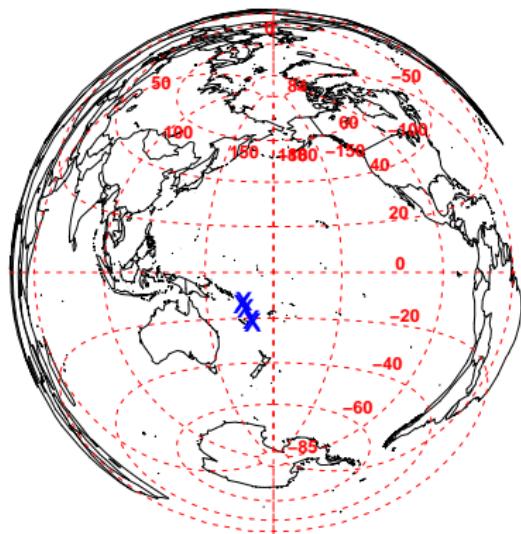
For a different perspective of a map, use `mapproject()`:

```
1 library(mapproj)
2 library(maps)
3 m <- map('world', plot=FALSE)
4 # Projection is Azimuthal with equal-area
5 map('world', proj='azequalarea', orient=c(
       longitude=0, latitude=180, rotation=0))
6 map.grid(m, col=2)
7 points(mapproject(list(
      y=quakes$lat[which(quakes$mag>=6)],
      x=quakes$long[which(quakes$mag>=6)])),
      col="blue", pch="x", cex=2)
```



# Projection Maps II

Map of Fiji Earthquakes Since 1964



## Exercise III

# Exercise III

For the ozone data set <sup>1</sup>, determine the region of the world that the observations correspond to and overlay the appropriate map.



---

<sup>1</sup><http://www.ats.ucla.edu/stat/R/faq/ozone.csv>

## Part IV

# Working with R



## 13 Installation

- Software Installation
- Saving Plots as a PDF

## 14 Common Bugs and Fixes

## 15 Getting R Help

## 16 Online Resources for R

## 17 References



# Saving Plots as a PDF

## For Static Plots

*Note:* The files will be saved in the folder specified with `setwd()`.

To save a static plot in R as a PDF, use function `pdf()`:

```
1 # To save the image to the desktop:  
2 setwd("~/Desktop")  
3 pdf("filename.pdf")  
4 wireframe(volcano, col.regions = terrain.  
           colors(100), asp = 1, color.key=TRUE,  
           drape=TRUE, scales = list(arrows = FALSE))  
5 dev.off()
```



## 13 Installation

## 14 Common Bugs and Fixes

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

## 15 Getting R Help

## 16 Online Resources for R

## 17 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 in Calling an Object

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



13 Installation

14 Common Bugs and Fixes

15 Getting R Help

16 Online Resources for R

17 References



# R Help: Approach 1

For help with any function in R, put a question mark before the function name to determine what arguments to use, examples and background information.

1    ?plot

plot (graphics)

Generic X-Y Plotting

R Documentation

**Description**

Generic function for plotting of R objects. For more details about the graphical parameter arguments, see [par](#).

For simple scatter plots, [plot.default](#) will be used. However, there are `plot` methods for many R objects, including [functions](#), [data.frames](#), [density](#) objects, etc. Use `methods(plot)` and the documentation for these.

**Usage**

`plot(x, y, ...)`

**Arguments**

`x` the coordinates of points in the plot. Alternatively, a single plotting structure, function or *any R object with a plot method* can be provided.

`y` the y coordinates of points in the plot, *optional* if `x` is an appropriate structure.

`...` Arguments to be passed to methods, such as [graphical parameters](#) (see [par](#)). Many methods will accept the following arguments:

`type`

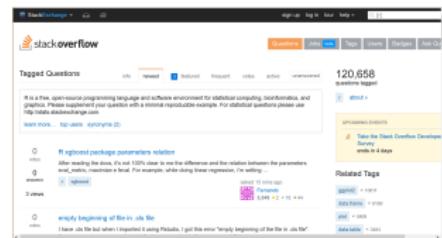
what type of plot should be drawn. Possible types are

- “`p`” for points,



# 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>



13 Installation

14 Common Bugs and Fixes

15 Getting R Help

16 Online Resources for R

17 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/>

More R tutorials:

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

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



# Online Resources for R II

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

JSS:

Stackoverflow:

Blogs:



13 Installation

14 Common Bugs and Fixes

15 Getting R Help

16 Online Resources for R

17 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](http://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/](http://flowingdata.com/2014/02/05/where-people-run/)
8. [https:  
//www.facebook.com/notes/facebook-engineering/  
visualizing-friendships/469716398919](https://www.facebook.com/notes/facebook-engineering/visualizing-friendships/469716398919)



Thank you.  
Any questions?

