Introduction to R

Shelley Knuth, Research Computing, University of Colorado-Boulder

shelley.knuth@colorado.edu

Questions? #RC_Meetups

Link to survey on this topic: http://goo.gl/forms/8VidcwOhRT

Slides: https://github.com/ResearchComputing/Final_Tutorials

Outline

- Intended for those with little to no knowledge of R

- Why use R?
- Getting R
- Layout and GUIs
- Data structures
  - Data Frames
- Reading in data
- Manipulating data
- Plotting data

What is R and why is it useful?

- Emerged from the “S” language
- In use by statisticians and data scientists
- Programming environment within which statistical analysis and visualization is conducted
- Open source
  - Many user-written packages to perform lots of statistical analysis
  - Source code and list of packages available for download is located as part of the Comprehensive R Archive Network (CRAN)

Advantages and Disadvantages

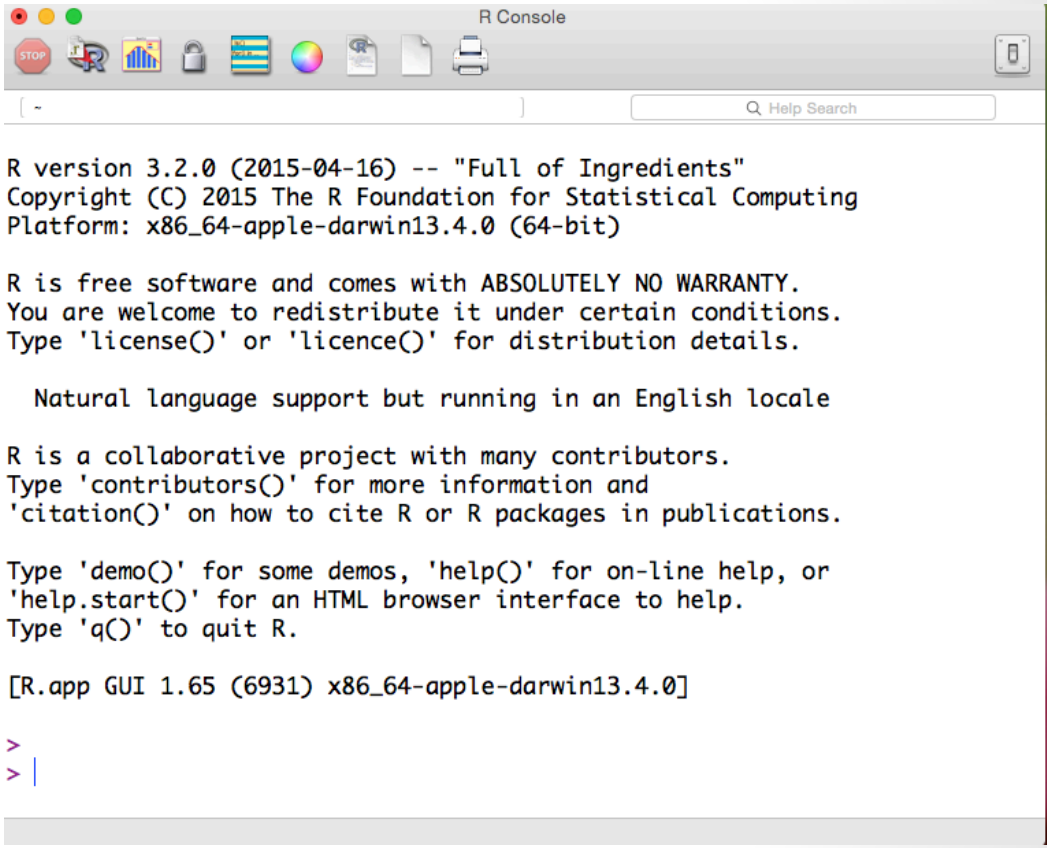
- R – Advantages
  - Free!!
  - Written by statisticians, for statisticians
  - Active community generating many statistics packages
  - Lots of online support
- R – Disadvantages
  - Written by statisticians, for statisticians
  - Can be complex to learn
  - Not really a programming language
    - Has a language but isn't a language
    - If it doesn't look like a traditional language can be more difficult to figure it out

How do I get R?

- Base installation:
  - Go to: https://cran.r-project.org/
  - Download the latest version
- For package add-ons:
  - Go to: https://cran.r-project.org/ and select “Contributed extension packages”
  - Here you will find a list of packages, including their description and their source files
  - Can also add them from the R Gui

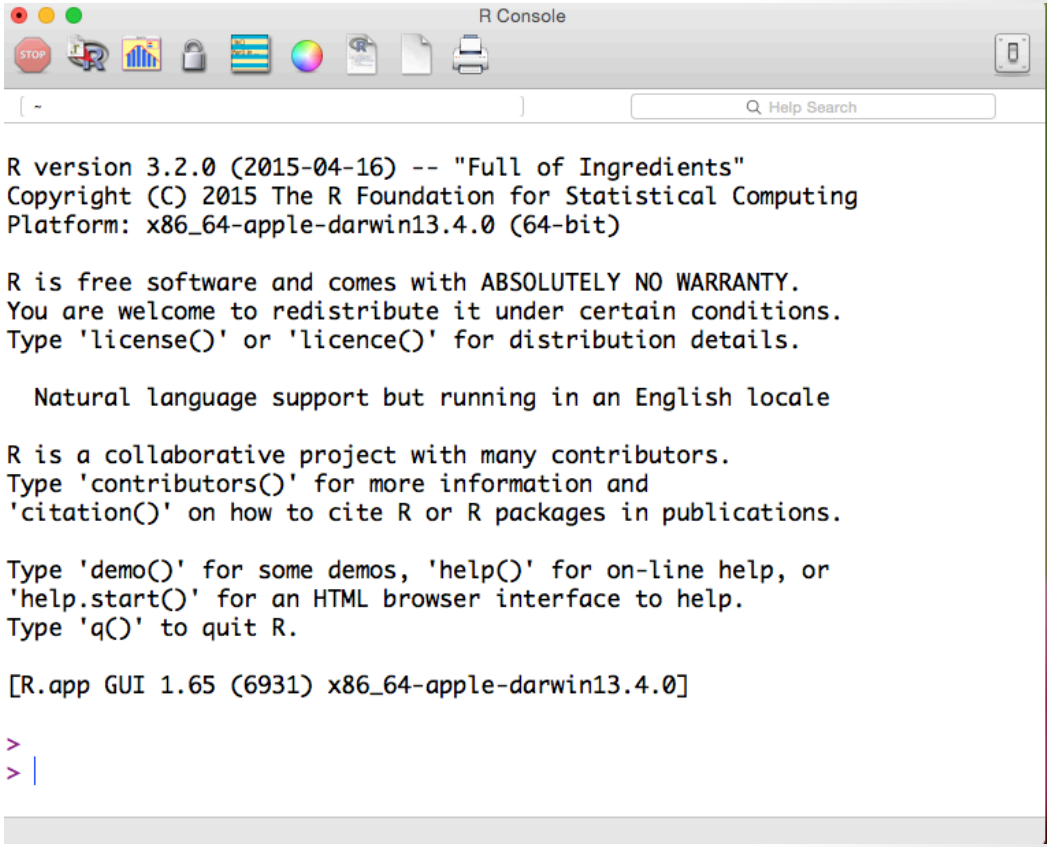
R Layout

- Can work with R using just the command line or with some sort of user interface
- You can open R a number of ways:
  - Typing “R” from a terminal window
  - From the Start Menu (Windows) or Applications Folder (Mac)



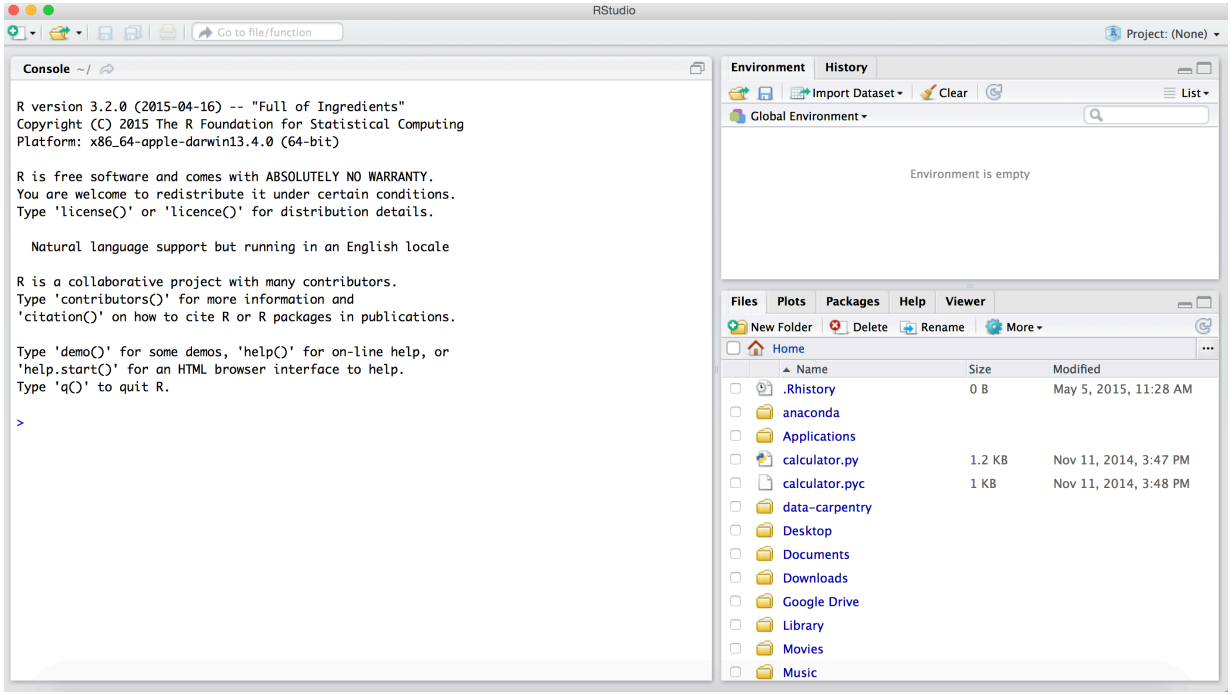
R Layout - Console

- Where you enter R commands
- Displays command history
- Results of commands
- Appears when R is launched



RStudio

- Rather than use the console, many people use the RStudio user interface
  - Highlighting, project management, etc.



R - General Syntax

	R
Element index	1
Comment	#
Print variable contents to screen	print(x) or just x
Print string	“Hello Everyone!”
Find help on a function	help(sum)
Script file extension	.R
Import library functions	R CMD INSTALL pkg –l /dir/ Or Install.packages(“pkg”, lib=/dir/
Assignment operator	<- or =
Line continuation	+

Working with Data

- One of the primary need-to-knows of any language
- Many ways to create data, read in data, etc.
- Experimental data is often held in tables, like in Excel
  - R can deal with tabular data relatively easily
- Five general data structures:
  - Vector
  - Matrix
  - Array
  - List
  - Data Frame
- Let's explore:
  - Data creation
  - Reading in data

Creating Vectors

- Several ways that you can create data in R
  - C command (combine data into a vector)

```
> newdata=c(1,7:9)
> newdata
[1] 1 7 8 9
```

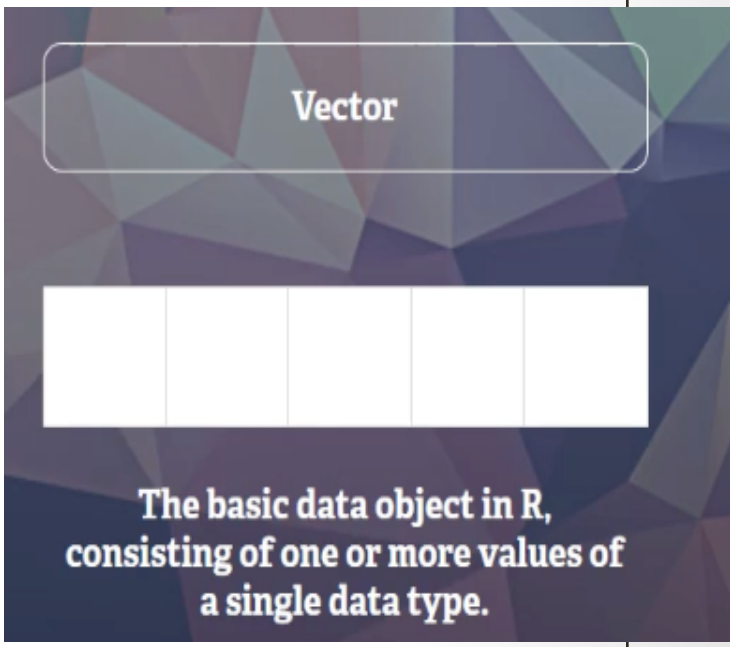
  - Seq command

```
> seq(0,5,0.5)
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

  - Simple assignment

```
> y
[1] 1 2 3 4 5
```

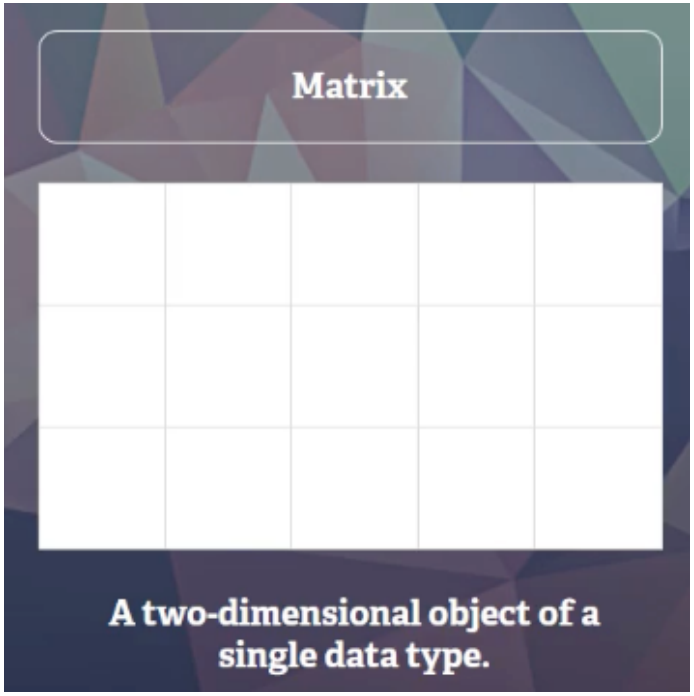
Can use this for strings, numerical values, or a combination of the two



From DataCamp

Data Creation - Matrices

- Matrix command (array command is similar)
- All values must be of the same type and have the same length



From DataCamp

Data Creation - Matrices

```
matrix(data, nrow, ncol)
```

```
> x=matrix(0,6,4)
> x
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
[4,]    0    0    0    0
[5,]    0    0    0    0
[6,]    0    0    0    0
```

```
> x=matrix(1:4,6,4)
> x
     [,1] [,2] [,3] [,4]
[1,]    1    3    1    3
[2,]    2    4    2    4
[3,]    3    1    3    1
[4,]    4    2    4    2
[5,]    1    3    1    3
[6,]    2    4    2    4
```

```
> x=matrix(newdata,6,4)
> x
     [,1] [,2] [,3] [,4]
[1,]    1    8    1    8
[2,]    7    9    7    9
[3,]    8    1    8    1
[4,]    9    7    9    7
[5,]    1    8    1    8
[6,]    7    9    7    9
```

Column/Row names

- You can name columns and rows in a matrix using the functions:

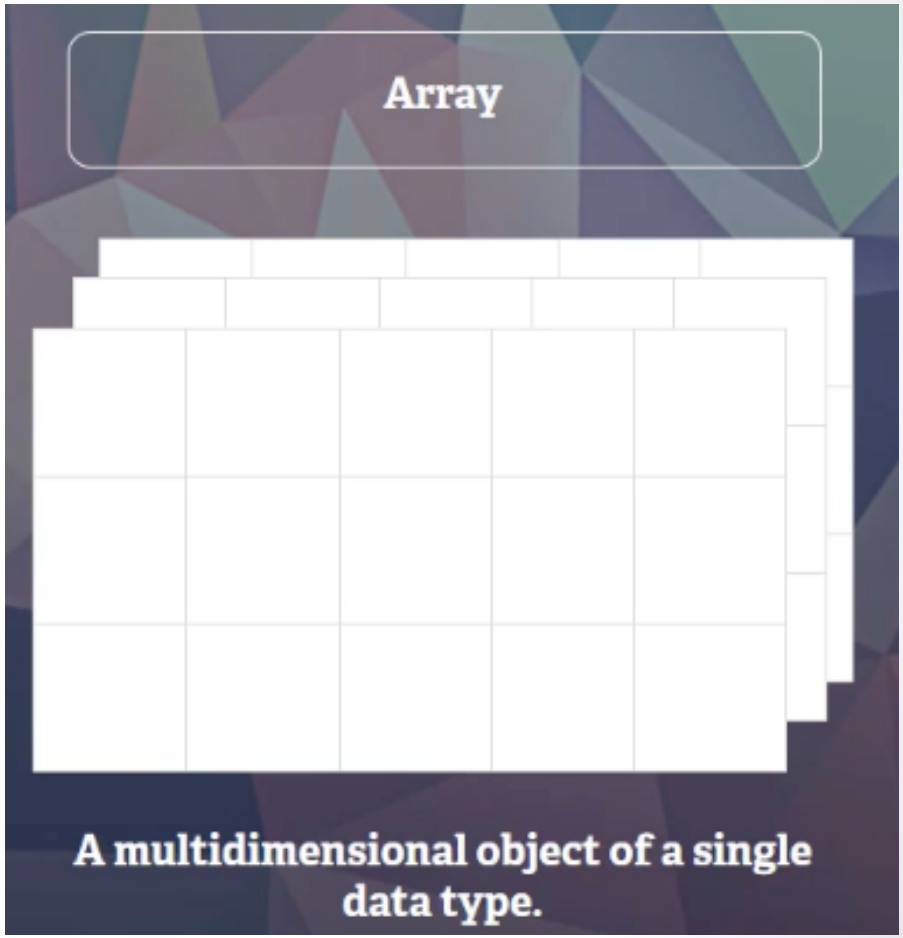
```
colnames(data, do.NULL, prefix)
rownames(data, do.NULL, prefix)
```

```
> colnames(x)=colnames(x,do.NULL=FALSE,prefix="col")
> rownames(x)=rownames(x,do.NULL=FALSE,prefix="row")
> x
     col1 col2 col3 col4
row1    1    2    3    4
row2    5    6    1    2
row3    3    4    5    6
row4    1    2    3    4
row5    5    6    1    2
row6    3    4    5    6
```

Arrays

- Similar to matrices except can have more than just two dimensions
- All values must be of the same type and have the same length

From DataCamp



Lists

- Ordered collection of components that allow you gather a variety of objects under one name
- Good for inputs from maybe a questionnaire, or keeping information about a person (name, address, birthdate, height, weight, etc)

From DataCamp

Lists

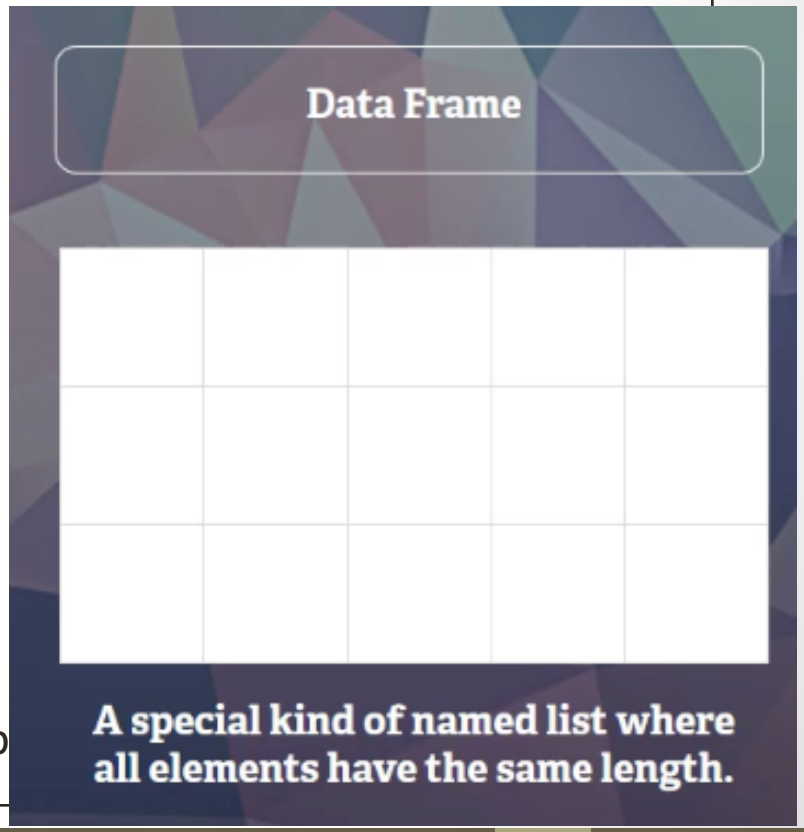
- Here's a pretty great way to demonstrate how to create a list and access an element

```
> awesome=list(name="Shelley", age=25)
> awesome$name
[1] "Shelley"
> awesome["name"]
$name
[1] "Shelley"
```

Data Creation – Data Frames

- Primary data structure in R
- Table where elements in a column are all measures of the same variable
- Elements of the same row are measures of the same case
- Different columns can have different data types (string, numeric, etc)
- All elements have to be the same length

From DataCamp



Data Creation – Data Frames

- Different columns can have different data types (string, numeric, etc)

```
> a=c(1,2,3,4)
> b=c("R","is","awesome",NA)
> c=c(FALSE,TRUE,FALSE,FALSE)
> new_df=data.frame(a,b,c)
> new_df
  a       b     c
1 1       R FALSE
2 2      is  TRUE
3 3 awesome FALSE
4 4    <NA> FALSE
```

Data Frame or Table?

- Have 15 responses to a study, which was divided into three experimental groups: control, treatment 1, and treatment 2
- Table

contr	treat1	treat2
22	32	30
18	35	28
25	30	25
25	42	22
20	31	33

- Not a data frame because the responses have been divided up by columns, and the column name has no real attachment to the data, which is important for analysis

http://ww2.coastal.edu/kingw/statistics/R-tutorials/dataframes.html

Data Frame or Table?

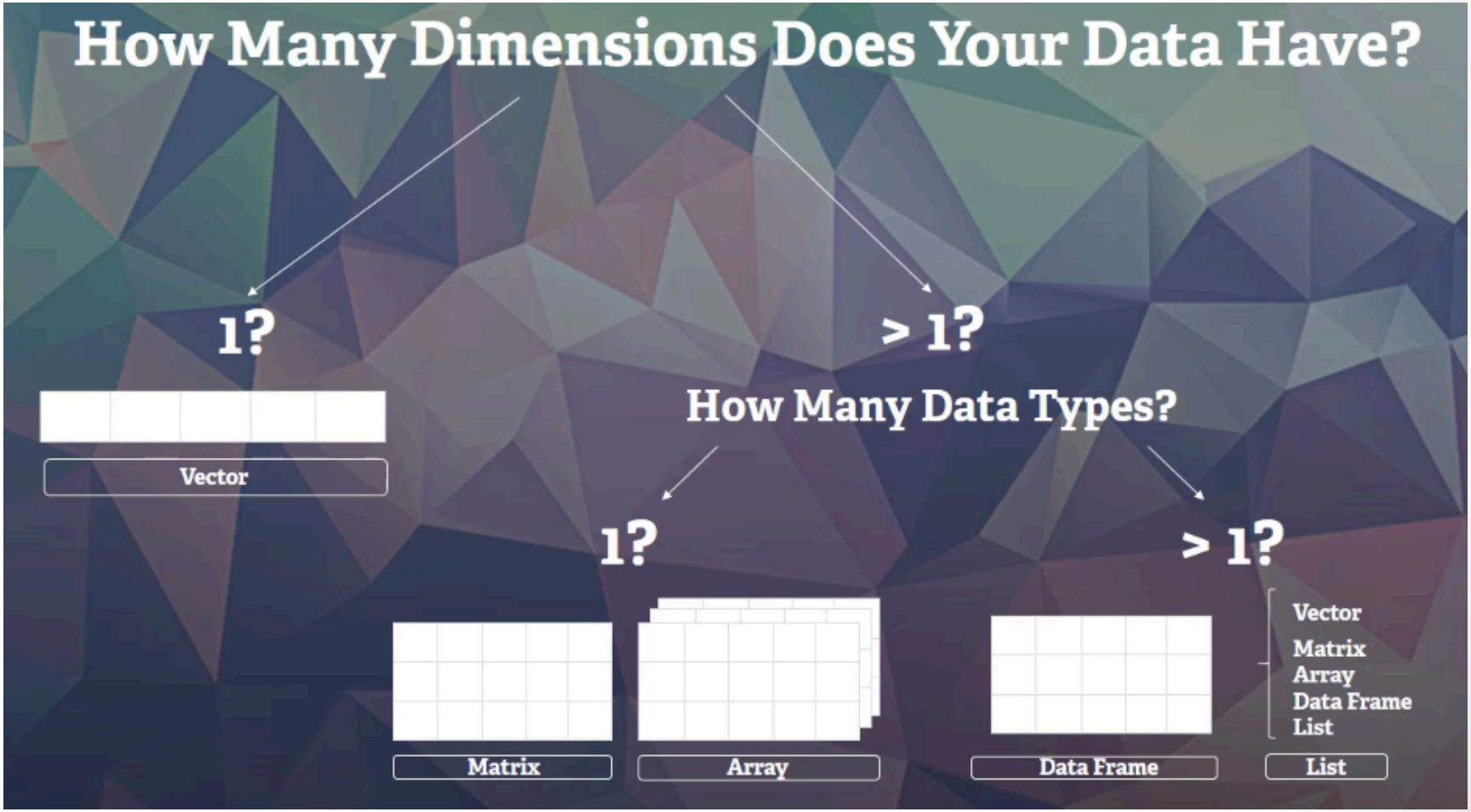
- A data frame has the variable name at the top, and values of the variable in the column under the variable name

scores	group
22	contr
18	contr
25	contr
25	contr
20	contr
32	treat1
35	treat1
30	treat1
42	treat1
31	treat1
30	treat2
28	treat2
25	treat2
22	treat2
33	treat2

http://ww2.coastal.edu/kingw/statistics/R-tutorials/dataframes.html

When do I use each structure?

From DataCamp



Data Frame vs. List vs. Matrix

- Data frames mimic behavior of matrices
  - You can do operations on rows
  - Cannot do that with a list
- Data frames are different from matrices because they can include heterogeneous data types
- Use data frames for any dataset where a row object is relational to a column

Accessing data in arrays/vectors

- Accessing data in R looks a little weird (at least to me)

```
> x
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    1    2
[3,]    3    4    5    6
[4,]    1    2    3    4
[5,]    5    6    1    2
[6,]    3    4    5    6
> x[,4]
[1] 4 2 6 4 2 6
> x[5,]
[1] 5 6 1 2
```

Reading in Data

- Many ways to read in data, depending on what you want to do:
- Read.table:

```
> aws_data=read.table("ftp://amrc.ssec.wisc.edu/pub/aws/10min/rdr/2015/089100415.r",skip=2)
> head(aws_data)
  V1 V2    V3    V4  V5  V6  V7  V8
1 91  1 -29.1 991.3 1.6  96  75 444
2 91  2 -29.2 991.3 1.6  96  74 444
3 91  3 -29.3 991.2 1.7  98  75 444
4 91  4 -29.3 991.1 3.8  98  74 444
5 91  5 -29.4 991.1 4.3  99  74 444
6 91  6 -29.4 991.2 3.9 100 444 444
```

- Reads it directly off the web

- Read in headers if they are there

```
> head(aws_data)
  V1 V2    V3    V4  V5  V6  V7  V8
1 91  1 -29.1 991.3 1.6  96  75 444
2 91  2 -29.2 991.3 1.6  96  74 444
3 91  3 -29.3 991.2 1.7  98  75 444
4 91  4 -29.3 991.1 3.8  98  74 444
5 91  5 -29.4 991.1 4.3  99  74 444
6 91  6 -29.4 991.2 3.9 100 444 444
```

- or set them:

```
> colnames(aws_data,do.NULL=FALSE)
[1] "V1" "V2" "V3" "V4" "V5" "V6" "V7" "V8"
> colnames(aws_data)=c("jday","10min_int","temp","pres","winds","windd","RH","VT")
> head(aws_data)
  jday 10min_int  temp  pres winds windd  RH  VT
1   91         1 -29.1 991.3   1.6    96  75 444
2   91         2 -29.2 991.3   1.6    96  74 444
3   91         3 -29.3 991.2   1.7    98  75 444
4   91         4 -29.3 991.1   3.8    98  74 444
5   91         5 -29.4 991.1   4.3    99  74 444
6   91         6 -29.4 991.2   3.9   100 444 444
```

Simple Statistics

- Use the summary function to get simple descriptive statistics on a data frame, including min and max, 1st and 3rd quartiles, median, and mean

```
> summary(aws_data)
      jday          10min_int            temp              pres            winds
 Min.   : 91.0   Min.   :  1.00   Min.   :-53.90   Min.   :444.0   Min.   :  0.000
 1st Qu.: 98.0   1st Qu.: 36.25   1st Qu.:-45.40   1st Qu.:964.5   1st Qu.:  0.800
 Median :105.0   Median : 72.00   Median :-41.50   Median :970.4   Median :  1.600
 Mean   :105.5   Mean   : 72.47   Mean   :-39.22   Mean   :969.7   Mean   :  3.005
 3rd Qu.:113.0   3rd Qu.:108.00   3rd Qu.:-34.40   3rd Qu.:977.0   3rd Qu.:  2.900
 Max.   :120.0   Max.   :144.00   Max.   :444.00   Max.   :991.3   Max.   :444.000
     windd              RH              VT
 Min.   :  0.0   Min.   :  0.0   Min.   :444
 1st Qu.:109.0   1st Qu.: 57.0   1st Qu.:444
 Median :170.0   Median : 66.0   Median :444
 Mean   :166.8   Mean   :150.6   Mean   :444
 3rd Qu.:226.0   3rd Qu.: 76.0   3rd Qu.:444
 Max.   :444.0   Max.   :444.0   Max.   :444
```

Statistics on rows/columns

- Use the `apply` function to apply a function to an entire array, matrix, or data frame or just each row/column

```
apply(data, margin, function)
```

```
> x
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    1    2
[3,]    3    4    5    6
[4,]    1    2    3    4
[5,]    5    6    1    2
[6,]    3    4    5    6
```

Apply to rows →

```
> apply(x,1,mean)
[1] 2.5 3.5 4.5 2.5 3.5 4.5
```

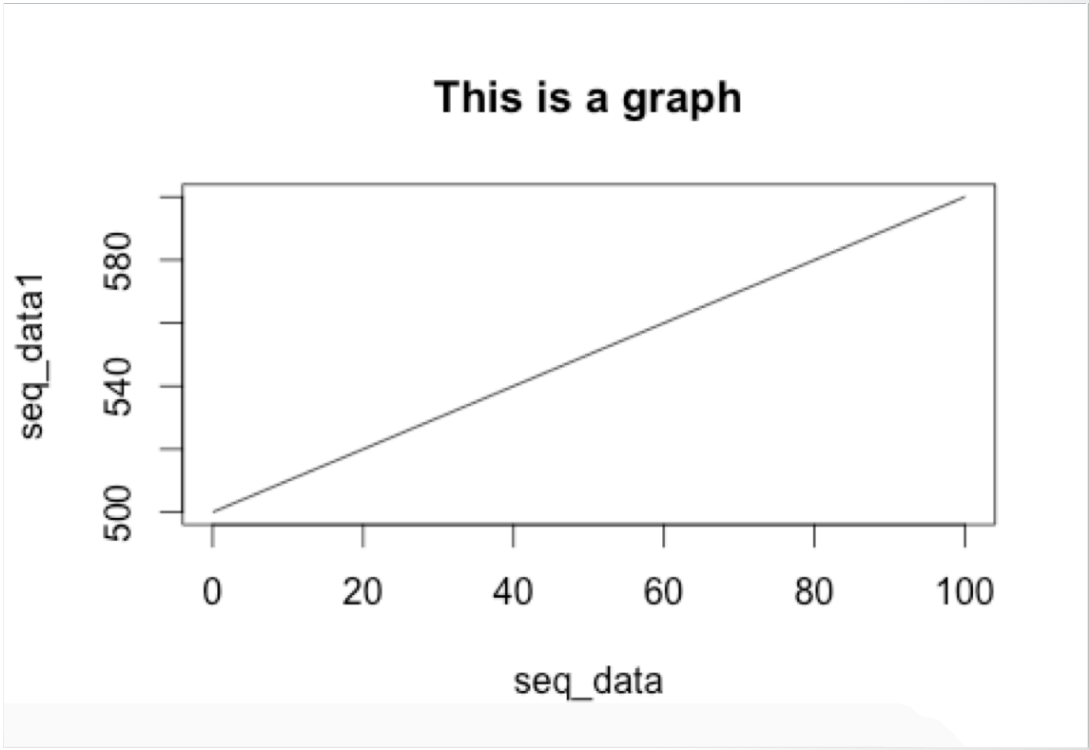
Apply to columns →

```
> apply(x,2,mean)
[1] 3 4 3 4
```

Simple Graphs

```
> seq_data=seq(0,100,1)
> seq_data1=seq(500,600,1)
> plot(seq_data,seq_data1,type="l")
> title("This is a graph")
```

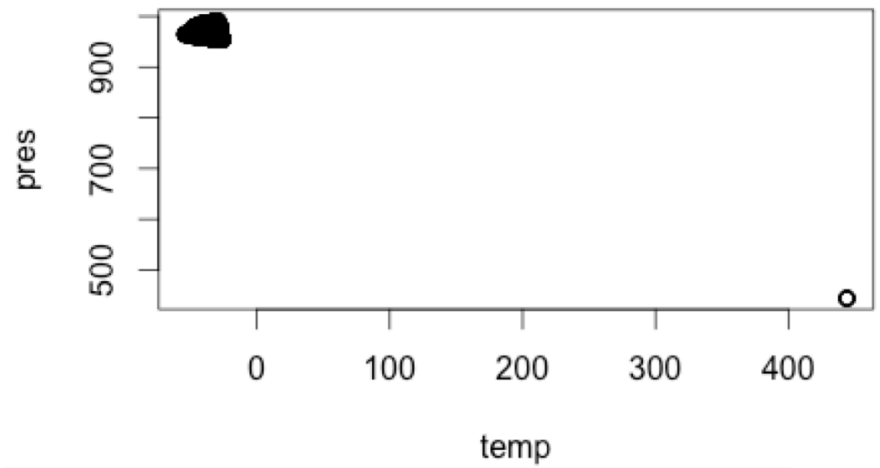
- We create two vectors using the sequence function
- The two vectors must be of equal size
- Then we plot that data, with "l" designating a line graph
- Then we add a title



NaN Values

- This dataset uses -444.0 as its missing value
- This is obvious if we do a scatterplot of temperature versus pressure:

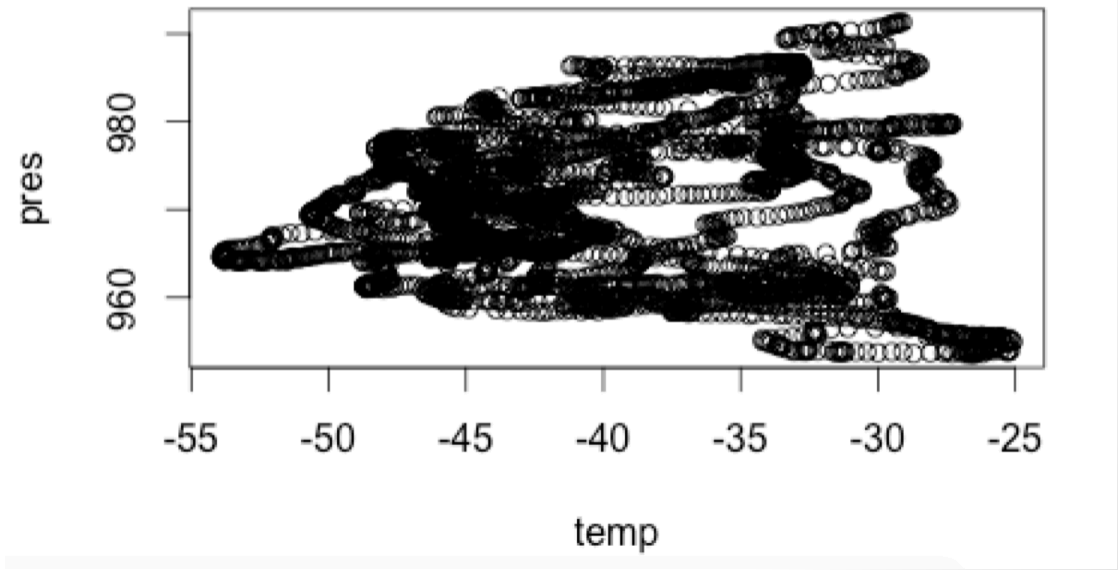
```
> pres=aws_data[,4]
> plot(temp,pres)
```



Set missing values

- Here we reassign the 444.0 values to be R's missing value, NA, and re-plot

```
> temp[temp==444.0] = NA
> pres[pres==444.0] = NA
> plot(temp,pres)
```



Basic Probability Distributions

- Normal Distribution
- t distribution
- Binomial distribution
- Chi-Squared distribution

- For every distribution there are four commands, prepended by a letter to indicate functionality
  - “d”: height of the probability density function
  - “p”: cumulative density function
  - “q”: quantiles
  - “r”: randomly generated numbers

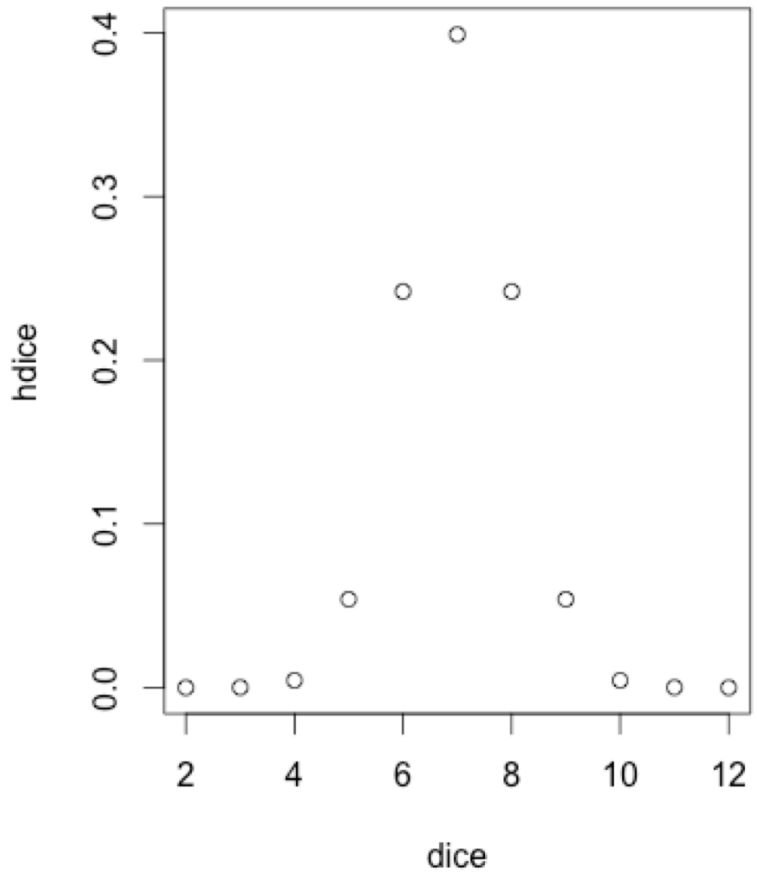
Basic Probability Distributions

- Normal Distribution
- For every distribution there are four commands, prepended by a letter to indicate functionality
  - “d”: height of the probability density function
- Let’s look at the dnorm function first
  - Returns the height of the probability distribution at each point

Basic Probability Distributions

- Use of dnorm
- In the casino game of craps, if a point has already rolled, a "7" will cause the shooter to lose their money (and the casino to win)
- What is the probability that the 7 will be rolled compared to other values?

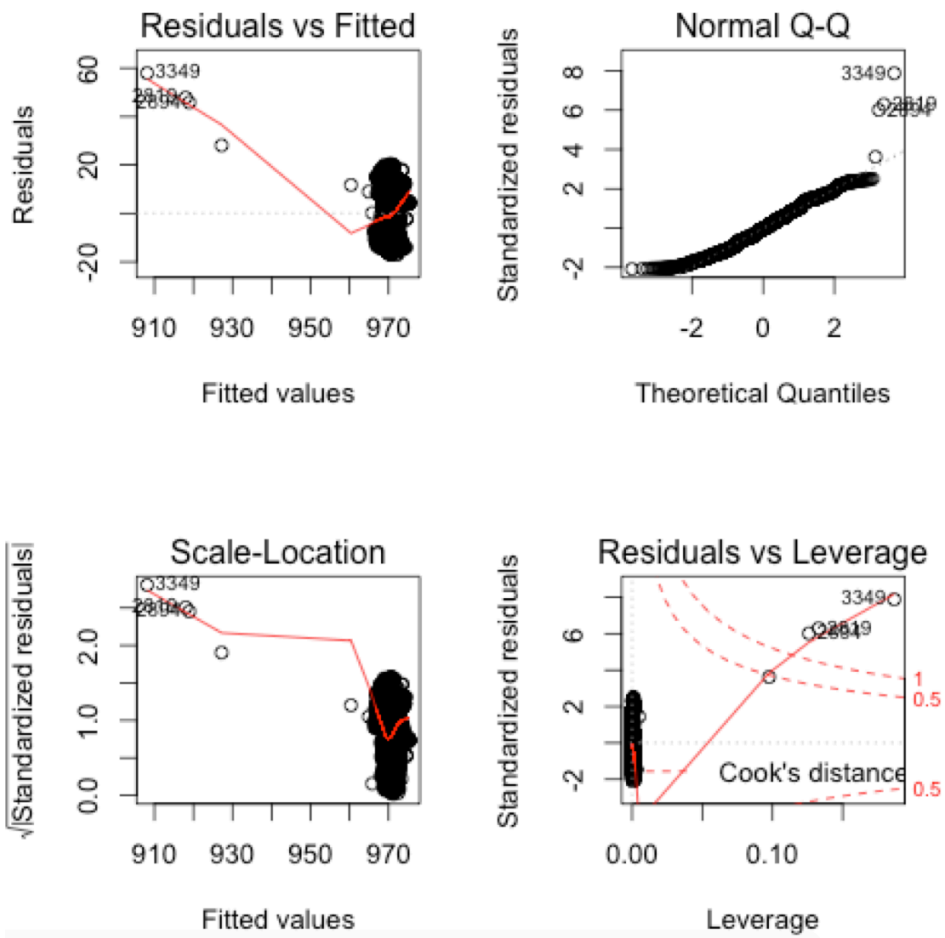
```
> dice=seq(2,12,1)
> hdice=dnorm(dice,mean=7)
> plot(dice,hdice)
```



Linear Models

- Fitting an ordinary linear model with pressure as the response and wind speed and temperature as the predictors

```
> winds=aws_data[,5]
> winds[winds==444.0] = NA
> lmfit=lm(pres~winds+temp)
> par(mfrow=c(2,2))
> plot(lmfit)
```



Thanks for Attending!

- Useful documentation: docs.python.org

- Email: rc-help@colorado.edu
- Shelley.knuth@colorado.edu
- Twitter: @shelley_knuth

- Survey: http://goo.gl/forms/8VidcwOhRT

References

- http://web.stanford.edu/group/ssds/cgi-bin/drupal/files/Guides/Using%20R%20for%20Windows%20and%20Macintosh_1.pdf
- https://www3.nd.edu/~mclark19/learn/Introduction_to_R.pdf
- https://www3.nd.edu/~steve/Rcourse/Lecture2v1.pdf
- http://data.princeton.edu/R/introducingR.pdf
- http://www.cyclismo.org/tutorial/R/probability.html

Questions?

- Email rc-help@colorado.edu
- Twitter: @CUBoulderRC

- Link to survey on this topic: http://goo.gl/forms/8VidcwOhRT

- Slides: https://github.com/ResearchComputing/Final_Tutorials

- Questions? #RC_Meetup