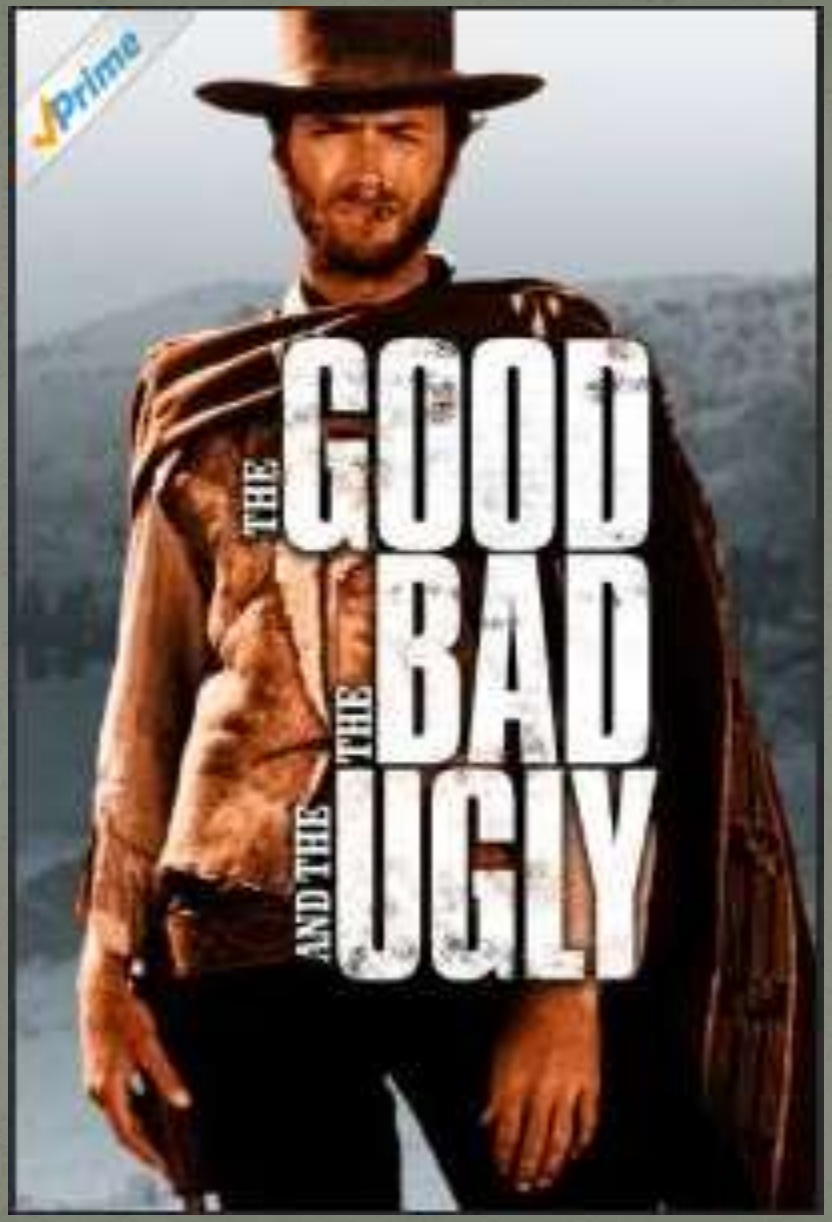
The Pirate's Code

Faron Kincheloe
Baylor University


Prime
THE GOOD
THE BAD
AND THE UGLY

Characteristics of R

- Free
- No warranty
- Open source - 7994 packages and growing
- No GUI
- Function/object based programming
- Nice, simple graphics
- Loops are slow -> Vectorize!!!

Characteristics of R

- Limitations on the types of data that R handles well
- Unable to handle large data sets quickly and efficiently
- No easy method for writing out tabular reports
- Available for several operating systems
- Popular in Academia

The Basics of R

The R Environment

- Console
  - Accepts commands
  - Displays results and messages(log)

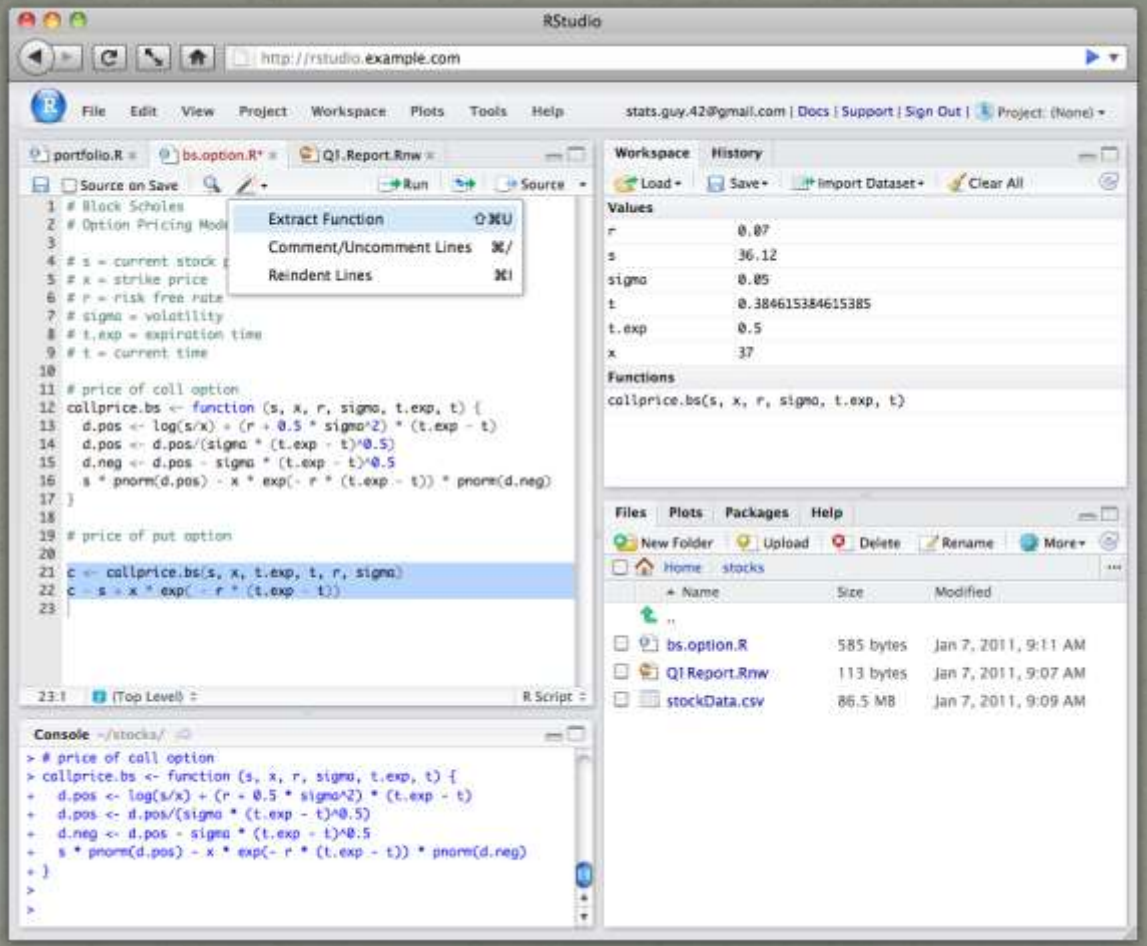
The R Environment

- Scripts
  - Save and retrieve your work (R “programs”)
  - Text files with .R extension
  - Use R Editor, Word, Emacs, or other text editor
  - Beware of Office quote marks & capitalization
  - Ctrl+r (Windows) or command+return (Mac) executes highlighted script commands

The R Environment

- The R Workspace
  - Collection of objects currently stored in R
  - Objects created or loaded during R session
  - objects() or ls() – display contents of workspace
  - May be saved as file – has RData extension
  - Saving when prompted at end of session creates both .Rdata and .Rhistory files
  - R reloads these saved files next time you run R

RStudio Interface



The R Environment

- Packages (libraries)
  - Extend functionality of R
  - May need to download and install package first
  - Use library() to show available packages
  - Use library(PackageName) to load
  - Use search() to see which packages are loaded
  - foreign – required for accessing external data such as SAS

Getting Help

- ?*functionname* – opens help page for function
- *functionname* – displays code of function
- example(*functionname*) – gives examples
- demo(*functionname*) – demo of some functions
- ??*keyword* – opens possible help pages
- ??"*multiple words*"
- PDF documents
- The Internet

Getting Help (Books)

- Michael J. Crawley. *The R Book*. (2012 or 2007) Wiley. ISBN: 978-0-470-97392-9 or 978-0-470-51024-7
- Kabacoff, Robert. *R in Action: Data Analysis and Graphics with R*. (2015) Manning Publications Co. ISBN: 978-1617291388
- Norman Matloff. *The Art of R Programming*. (2011) No Starch Press. ISBN: 978-1-59327-384-2.

Data Storage in R

- Named data structures (objects)
- Vector – series of data values
- Scalar – single value vector
- Matrix or array – multidimensional vectors of same data type (matrix: 2 dimensions)

Data Storage in R

- Factor – grouping by category
- Data frame – matrix-like structure with different data types
- Function – object containing program code
- Typically returned by class() function

Common R Data Types

- Numeric
  - Integer
  - Double
- Logical – True/False
- Character
- List – elements not of same type (also a structure)
- Typically returned by mode() function

Information About Objects

- class(*objectname*) – reveals object structure
- mode(*objectname*) – reveals data type
- summary(*objectname*) – additional info depending on class of object
- str(*objectname*) – **str**ucture of R object
- length(*objectname*) – number of values

R Operators

- Arithmetic and Assignment Operators

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/% y	integer division 5%/%2 is 2
x<-y or y->x	assignment; x gets y
:	create series (1:10)

R Operators

- Logical Operators

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
!x	Not x
x | y	x OR y
||	OR with IF
x & y	x AND y
&&	AND with IF
isTRUE(x)	test if X is TRUE

Command Syntax

- functionname(x,arg=0)
- x – positional argument
  - usually required
  - must be in expected location (order)
- arg – keyword argument
  - often optional
  - usually has a default value

Command Syntax

- How do you specify an argument with multiple values that are separated by a comma?
  - c(1,2)
  - Known as the combine function
  - plot(c(1,2,3,4,5),c(1,2,3,4,5))

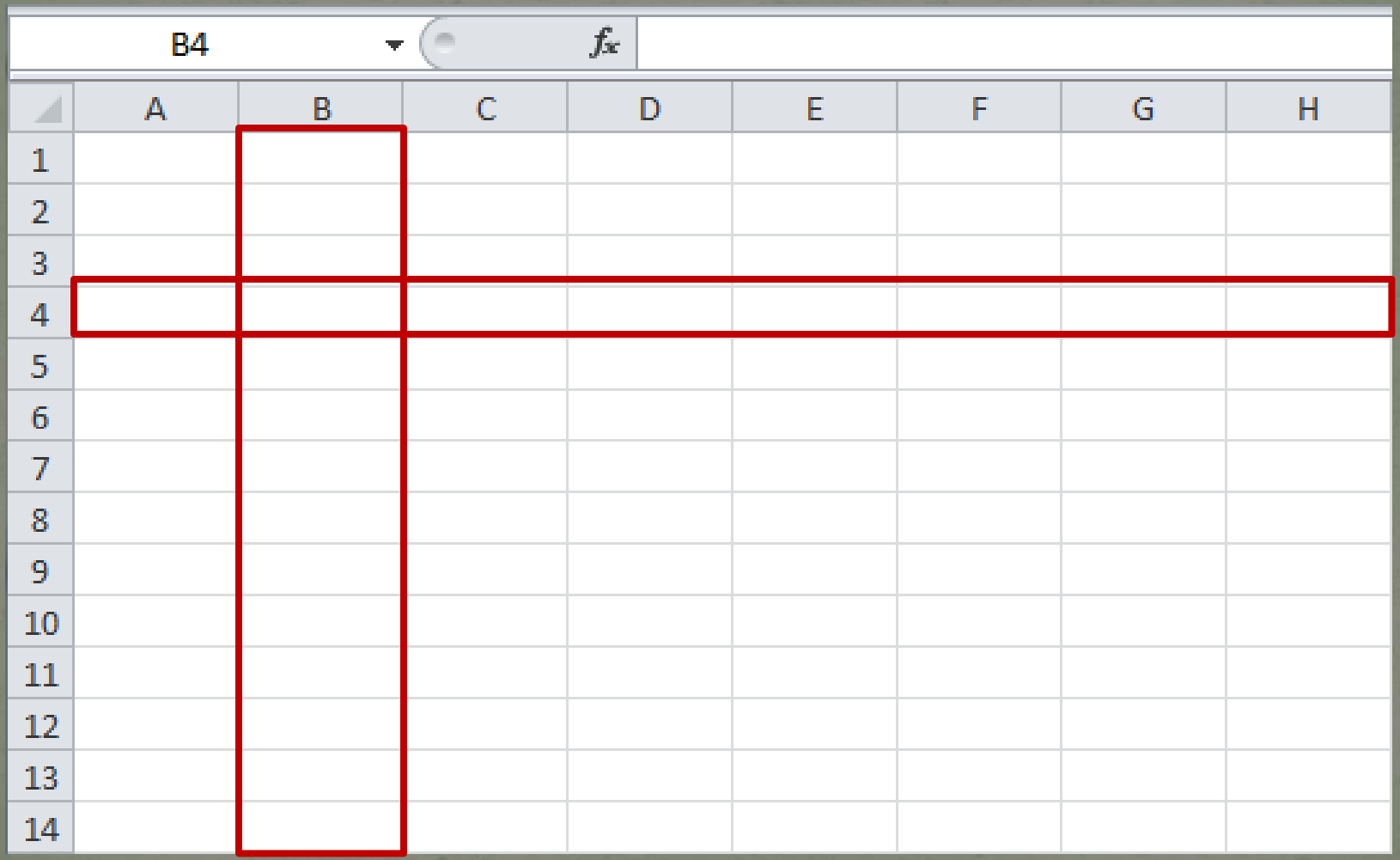
Some Commonly Used R Functions

- length()
- sum(), cumsum(), prod(), cumprod()
- mean(), sd(), var(), median(), min(), max(), range(), summary()
- exp(), log(), sin(), cos(), tan() [radians, not degrees]
- round(), ceiling(), floor(), signif()
- sort(), order(), rank(), rev()
- which(), which.max(), which.min()
- any(), all()
- apply(), tapply(), lapply()

Command Syntax

- The devil is in the details!
- This is different from a function!!!
  - dataobject[indices]
  - dataobject – name of data frame, vector, etc.
  - indices – vector, formula, or function to specify members to use

Working With Vectors

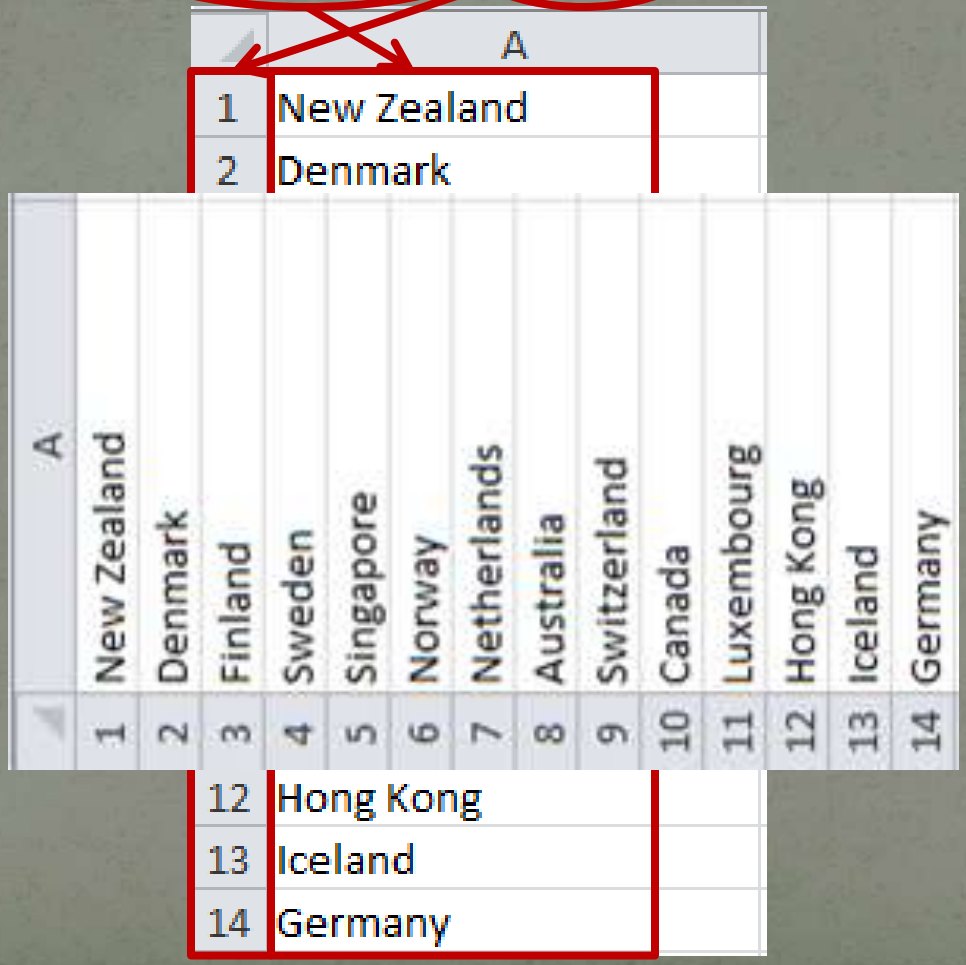


Working With Vectors

- A vector is a series of values
- Single dimension
- Not necessarily part of a data frame or matrix
- Frequently are a subset of data frame or matrix
- (V<-c(1:14))
- [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Working With Vectors

countries[1:14]



Working With Missing Data

- NA
- <NA> (among characters without quotes)
- Function arguments
  - na.strings= specify values to identify as missing
  - na.rm=T instructs function to remove missing
- Functions
  - na.omit() removes cases from action inside ()
  - is.na() tests to see if a value is missing
- NaN "Not a number" i.e.: Square root -4

Working With Vectors

1	New Zealand	9.5
2	Denmark	9.4
3	Finland	9.4
4	Sweden	9.3
5	Singapore	9.2
6	Norway	9
7	Netherlands	8.9
8	Australia	8.8
9	Switzerland	8.8
10	Canada	8.7
11	Luxembourg	8.5
12	Hong Kong	8.4
13	Iceland	8.3
14	Germany	8
15	Japan	8
16	Austria	7.8

Working with Data Frames

cpidf[??]

	Country	CPI
1	New Zealand	9.5
2	Denmark	9.4
3	Finland	9.4
4	Sweden	9.3
5	Singapore	9.2
6	Norway	9.0
7	Netherlands	8.9
8	Australia	8.8
9	Switzerland	8.8
10	Canada	8.7
11	Luxembourg	8.5
12	Hong Kong	8.4
13	Iceland	8.3
14	Germany	8.0
15	Japan	8.0
16	Austria	7.8

cpidf[row,col]

	countries	CPI
[1,]	"New Zealand"	"9.5"
[2,]	"Denmark"	"9.4"
[3,]	"Finland"	"9.4"
[4,]	"Sweden"	"9.3"
[5,]	"Singapore"	"9.2"
[6,]	"Norway"	"9"
[7,]	"Netherlands"	"8.9"
[8,]	"Australia"	"8.8"
[9,]	"Switzerland"	"8.8"
[10,]	"Canada"	"8.7"
[11,]	"Luxembourg"	"8.5"
[12,]	"Hong Kong"	"8.4"
[13,]	"Iceland"	"8.3"
[14,]	"Germany"	"8"
[15,]	"Japan"	"8"
[16,]	"Austria"	"7.8"

Working with Data Frames

- Treat a row as a vector: cpidf[row#,]
- Treat a column as a vector (all rows)
  - cpidf[,col#]
  - cpidf$ColName
  - attach(cpidf)
  - with(cpidf, ColName or function using Colname)
- Get all column names: names(cpidf)
- Get all row names: row.names(cpidf)

String Functions

- s <- c('apple','bee','cars','danish','egg')
- nchar(s) #length of each member
- substr(s,2,3) # substring second to third character
- grep('e',s) # search s for instances of 'e'
- grep('^e',s) # search with regular expressions
- sub('e','_',s) # replace first 'e' with '_'
- gsub('e','_',s) # globally replace 'e' with '_'
- toupper(s) #convert all letters to upper case

Reordering Values

- sort – returns actual values in desired order
- order – returns vector of indices in new order
- order(..., na.last = TRUE, decreasing = FALSE)
  - ... a sequence of numeric, complex, character or logical vectors, all of the same length, or a classed **R** object.
  - na.last= – where to put missing values (NA removes)
  - decreasing= – increasing order by default
- order(cpidf$Country, decreasing=T)

User Defined Functions

- Purpose – reusable code
- ?"function" for documentation
- Components
  - Name
  - Inputs (arguments or parameters)
  - Code that does something
  - Output (return)

Function Example

mysum<-function(a,b,c=0) a+b+c

mysum - *the name of my new function*

function(a,b,c=0) – *defines the structure*

a,b – *positional parameters*

c – *optional parameter with default value*

a+b+c – *code that returns sum of three inputs (class of returned value depends on inputs)*

- variables are local to the function

Importing Data

- See help on read.table for more info
- R recommends converting Excel, etc. to delimited text if possible – Know thy data
- read.csv(file, header = TRUE, sep = ",", quote="\"", dec=".", fill = TRUE, comment.char="", ...)

Basic Plotting Command

- plot(x, y, type="p", pch=2, col="red")
  - *x-y scatter plot*
  - *x: vector of horizontal values*
  - *y: vector of vertical values*

Basic Plotting Command

- plot(x, y, type="p", pch=2, col="red")
  - *type: how to plot*
    - *p – points*
    - *l – line*
    - *b - both*
  - *pch: vector of numbers defining the plot symbol.*
    - *R recycles*
  - *col: vector of any of three kinds of R color specs*
    - a color name (as listed by colors()) i.e. “maroon”
    - a hexadecimal string of the form "#rrggbb"
    - a positive integer i meaning palette()[i]

Output Devices

- ?Devices
  - *Provides a list and explanation of devices*
- Graphics functions opens Screen device if device not specified
  - *Use console menu to save results*
- Bitmap devices (single image)
  - png()
  - jpeg()
  - bmp()
  - tiff()

Output Devices

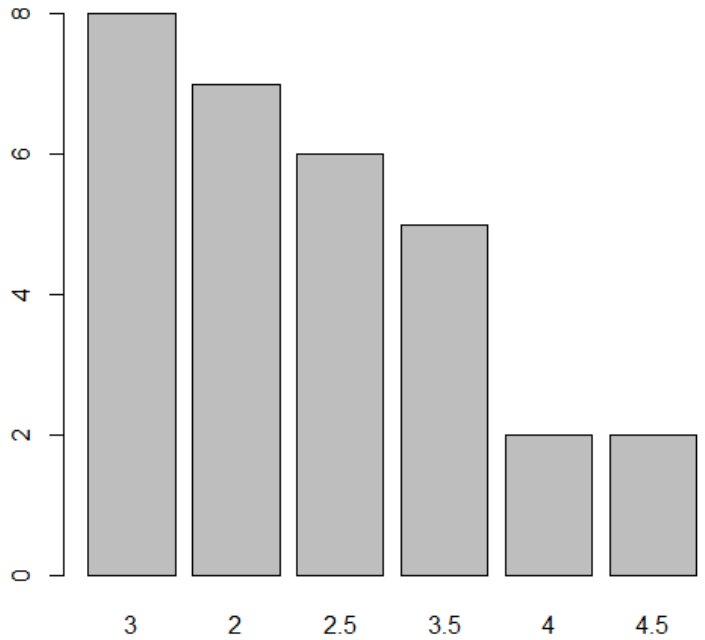
- File devices (multiple images)
  - postscript()
  - pdf()
  - pdf("C:/Rfiles/HW05.pdf")
- Close the device when finished
  - dev.off()
  - graphics.off()

Helpful Plotting Info

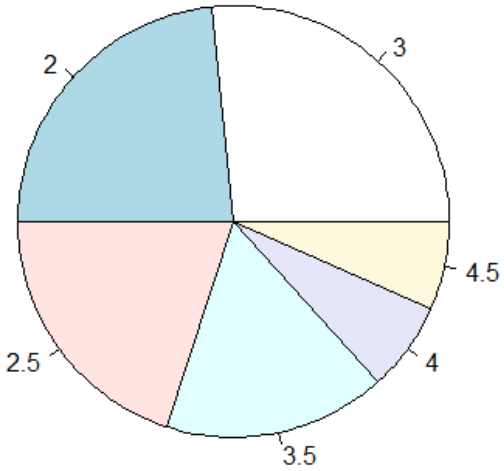
- colors()
  - *displays in the console a list of all color names*
- ?points
  - *help page that provides a definition of all the available plot characters (pch)*
- palette()
  - *displays vector of colors in the current palette*

Higher Level Graphics
For categorical variables:

Bar Plot

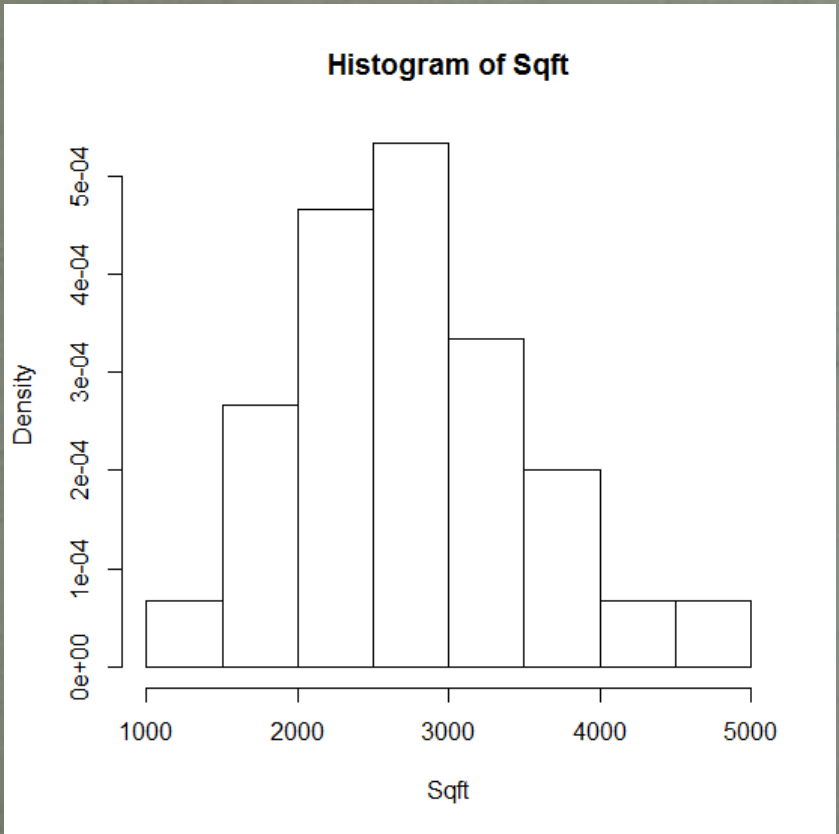


Pie Chart

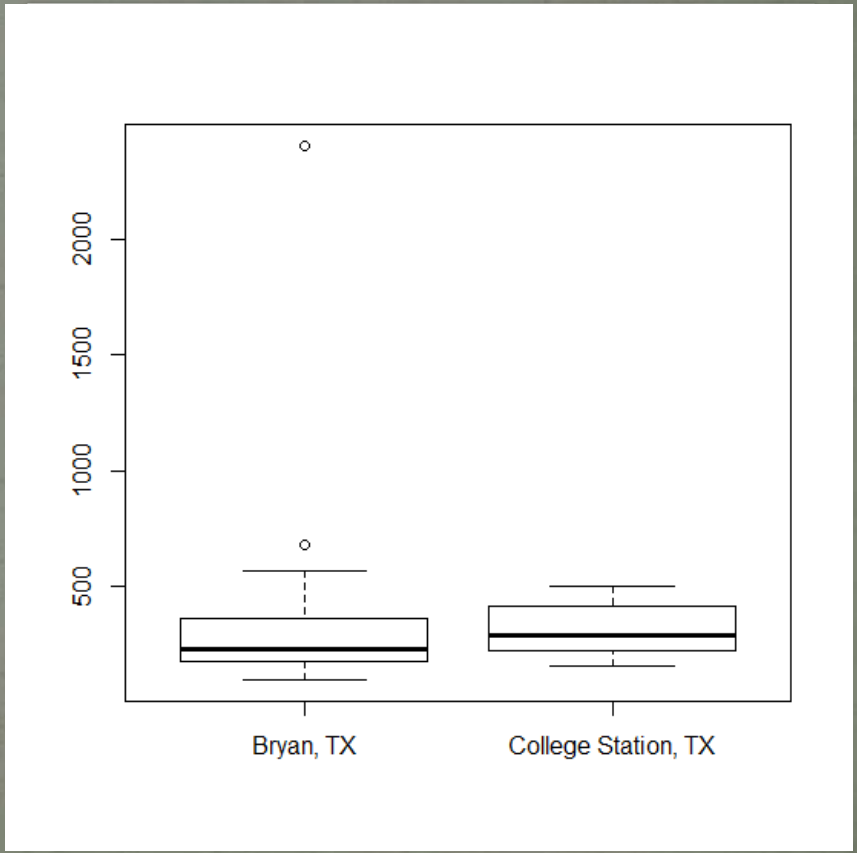


Higher Level Graphics
For continuous variables:

Histogram



Boxplot



Bar Plot

- barplot(height, width = 1, space = NULL, names.arg = NULL, beside = FALSE, horiz = FALSE)
  - height *vector or matrix describing the bars*
  - width *width of bars*
  - space *amount of space between bars*
  - names.arg *vector of names below bars*
  - beside *controls bar stacking*
  - horiz *orientation of bars*

Bar Plot Example

- barplot(sort(table(Baths),decreasing=T))
- table()
  - *uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels*

Pie Chart

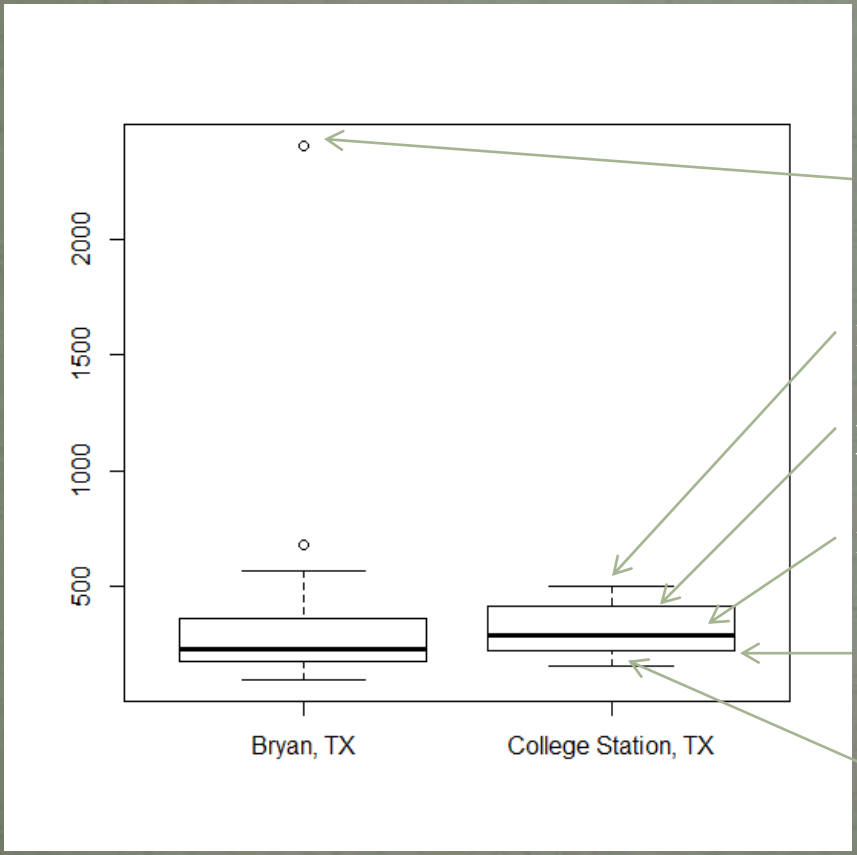
- pie(x, labels = names(x), clockwise = FALSE, init.angle = if(clockwise) 90 else 0)
  - x *vector of non-negative numerical quantities*
  - labels *names for slices*
  - clockwise *specifies order slices are drawn*
  - init.angle *specifies starting angle for slices*
- Pie charts are not an ideal method of displaying information.

Histogram

- hist(x, breaks = "Sturges", freq = NULL)
  - x *vector of values for which histogram is desired*
  - breaks
    - ✓ *a vector giving the breakpoints between histogram cells*
    - *a single number giving the number of cells for the histogram*
    - *a character string naming an algorithm to compute the number of cells (Sturges, Scott, Freedman-Diaconis/FD)*
    - *a function to compute the number of cells*
  - freq *choose counts or probability densities for y axis*

Boxplot

- 5 number summary



outlier

max

3rd Quartile

median

1st Quartile

min

Boxplot

- boxplot(formula, range = 1.5, notch = FALSE)
- boxplot(x, …)
  - formula *such as y ~ grp (y as a function of grp)*
  - x *vector or list of vectors (like data frame columns)*
  - range *position of whiskers*
    - *0: include all values*
    - *positive value n: limit to n times interquartile range*
  - notch *draw notch on each side of boxes*
- boxplot(Price ~ Location, range=0)

Adding Objects to Plots

- points(x ,y, col="blue")
  - *add more points to an existing plot*
- lines(x, y, col="blue")
  - *add more points connected by lines to an existing plot*
- polygon(x,y,col="blue")
  - *fill a polygon defined by the x and y values*

Adding Objects to Plots

- abline(a, b)
  - slope/intercept line as in y=b(x)+a
- abline(h=y)
  - *A horizontal line at the value y*
- abline(v=x)
  - *A vertical line at the value x*
- abline(lm(y~x))
  - *Fit line from linear model of control x and response y*
- summary(lm(y~x))
  - *Shows statistics for linear model*

Adding Objects to Plots

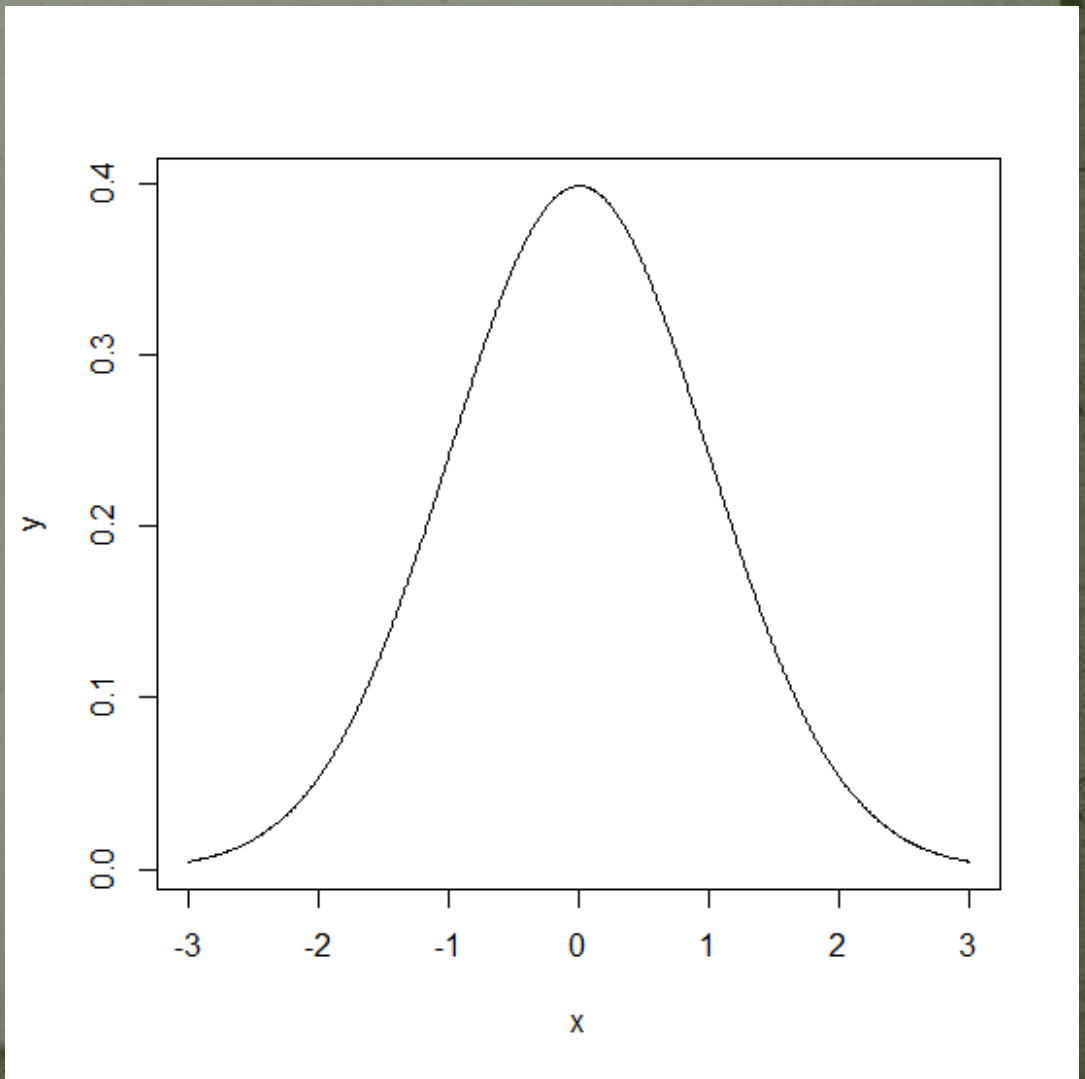
- text(x,y,s,cex=0.8)
  - *centers the string s at the values x and y where cex controls the size of the text*
- legend(locator(1),c("treatment", "control"),pch=c(1,1),col=c(2,4))
  - *locator=click to position the legend (or replace with x, y coordinates of upper left corner)*
  - *col=colors of plot objects*
  - *pch=plotting characters*

The Normal Distribution

- Bell curve
- 68 – 95 – 99.7% Rule
  - *68% of observations within 1 std. dev. of mean*
  - *95% of observations within 2 std. dev. of mean*
  - *99.7% of observations within 3 std. dev. of mean*
- dnorm(x, mean = 0, sd = 1)
  - x *vector of quantiles*
  - mean *vector of means*
  - sd *vector of standard deviations*

Normal Distribution Example

- x <- seq(-3,3,0.01)
  y <- dnorm(x)
  plot(x,y,type="l")
- Use lines to add to add to existing



Graphics Parameters

- Session parameters affecting graphics devices
- par() outputs the (long) list of options & current values
- Read ?par for all the details!
- Some can only be set with par()
- mfrow, mfcol: multiple plots per image
  - *par(mfrow=c(nr, nc))*
  - *mfrow vs. mfcol controls the order graphs appear*

Graphics Parameters

- mar, oma, mai, omi: margin controls
  - *par(oma=c(bottom, left, top, right))*
  - *mar/mai margins for the specific plot area*
  - *oma/omi overall (page) margins*
  - *parameters ending with i use inches as units*
  - *parameters not ending in i use lines as units*
- adj: text alignment control
  - *0 left justifies*
  - *default of 0.5 centers text*
  - *1 right justifies*

Mathematical Annotation in R

- plotmath
  - *not a function but a collection of features*
  - *imbed inside text functions and options*
  - ?plotmath
  - demo(plotmath)
  - example(plotmath)

Mathematical Annotation in R

- expression()
  - *no variables get evaluated*
  - *add plain text inside expression with paste()*
  - *text(4, 7, expression(bar(x) == sum(frac(x[i], n), i==1, n)))*
  - *xlab = expression(paste("Phase Angle ", phi))*

Mathematical Annotation in R

- bquote(expr1 .(expr2))
  - *evaluates anything inside .()*
  - *mu <- 8.25*
    ylab=bquote(mu== .(mu))

How to get R?

- cran.r-project.org
- www.rstudio.com

Questions?

Faron_Kincheloe@baylor.edu