### INTRODUCTION TO R

Bruin Actuarial Society



#### AGENDA

- Introduction
- Getting Started
- Data Manipulation with dplyr
- Plotting with ggplot2 package



## INTRODUCTION



#### MHX K5

- R is a popular statistical programming language
  - Free
  - Great capabilities: from simple regression to machine learning
- Excel has its limitation:
  - Hard to implement statistical models
  - Graphing in Excel can also be challenging at times
- R can solve all the above problems and perform the task you can do in Excel faster



### GETTING STARTED



#### STARTING R

• We will use RStudio, an Integrated Development Environment, to program





#### STARTING R

 To actually write and save your code, you will need a R script. We can add an R script under "File", or Ctrl + Shift + N (Window)/Cmd + Shift + N (Mac)

#### C:/Users/gavin/Desktop/BAS/R workshop/Demo - RStudio

Edit	Code	View	Plots	Session	Build	Debug	Profile	Tools	Help	
New	File				)		R Script		Ctrl+Shift+N	
New	Project						R Notebook			
Oper	File			Ctrl+O						
Recei	nt Files				)		K Markdov Shiny Web	vn		
Oper	Project						Shiriy Web	Арр		
Oper	Project i	n New S	ession				Text File			
Recei	nt Projects	s			,		C++ File			
Impo	rt Dataset	t			,		R Sweave			
				0.1.0			R HTML R Presentation R Documentation			
Save As			Ctrl+S							
Save All				Ctrl+Alt+	S	fo	or on-line help, or			
Print						terf	ace to	help	•	
Close	2			Ctrl+W						
Close All Close All Except Current		Ctrl+Shift+W Ctrl+Alt+Shift+W								
Close	Project									
Ouit	Session			Ctrl+Q						

#### 🚯 C:/Users/gavin/Desktop/BAS/R workshop/Demo - RStudio



**BRUIN ACTUARIAL SOCIETY** 

#### ASSIGNMENT

- Use "<-" for assignment
- Use "c(...)" to create vector

C:/Users/gavin/Desktop/BAS/R workshop/Demo - RStudio					
File Edit Code View Plots Session Build Debug Profile Tools Help					
💽 🗸 🧐 🥣 🖌 🕞 🔚 📥 🛛 🥕 Go to file/function 🔤 🗄 👻 Addins 👻					
Untitled1* ×		Environment	History	Connections	
🗇 🖒 🔎 🔚 🗌 Source on Save 🛛 🔍 🎢 🖌 📋	🕣 📊 🖙 Import Dataset 👻 💉				
1 db] <- 2.65		🔩 Global Environment 👻			
2 Int <- 1L 3 char <- 'a'		Values			
4 vec1 <- c(1,2,3.5)		char		"a"	
5 vec2 <- c(db1, int, char)		db1		2.65	
0		int		1L	
		vec1		num [1:3] 1 2 3.5	
		vec2		chr [1:3] "2.65" "1" "a"	





- R does element-wise operation by default
- It also recycle through vector if their length does not match

```
> c(1,2,3)+c(4,5,6)
[1] 5 7 9
> c(1,2,3)*c(4,5,6)
[1] 4 10 18
> c(1,2,3,4) - c(5,6)
[1] -4 -4 -2 -2
>
```



#### SUBSETTING

- You can subset data table (matrix/dataframe/tibble) using indices
  - d[1, 3] represent 1<sup>st</sup> row 3<sup>rd</sup> column of d
- You can also subset using column name of data table with "\$"
- You can also subset using row name and column name

```
1:1
       (Top Level) $
Console
         Terminal ×
C:/Users/gavin/Desktop/BAS/Workshop/R w
> d
     col1 col2 col3
row1
         1
              4
                    7
         2 5
row2
                    8
         3
               6
row3
                    9
> d[1,3]
[1] 7
> d$col1
[1] 1 2 3
> d["row1", ]
     col1 col2 col3
row1
               4
                    7
> d[, "col1"]
[1] 1
      2 3
```



# PRACTICE TIME!





- 1. Create a vector from 1 to 10 and assign it the variable "v"
- Increase the <u>odd</u> elements of v and decrease the <u>even</u> elements by 2 (Hint: vector cycling)
- 3. We will practice subsetting with the built-in dataset "iris"
  - a. Type in iris on your console/script to examine the data
  - b. Subset the 4<sup>th</sup> row and 2<sup>nd</sup> column of iris
  - c. Subset the Petal Length column



## BASIC DATA MANIPULATION WITH DPLYR





- We can install and load "packages" from internet in R
- We will load "tidyverse" package, which includes the "dplyr" package and "ggplot2" package



### READING DATA

- R has build-in functions for reading data, but generally they are slow
- The "tidyverse" package contains the "readr" package that reads data quickly
- Usually, data are in ".csv" type





#### READING DATA SIMPLIFIED

• Importing CSV files through "Import Dataset" tab and browse to the desired file





#### READING DATA SIMPLIFIED

#### • Click "Import" and the dataset will appear in the global environment

1								Environment History Connections	
Import Text D	ata								
File/Url:									
~/Download	ls/grades.csv								Browse
Data Previev	v:						1		
Last name	First name	SSN	Test1	Test2	Test3	Test4	Final	Grade	
(character)	▼ (character) ▼	(character) 🦷	(character) 🦷	(double) 🦷	(double) 🦷	(double)	(character) 🦷	(character)	
Alfalfa	Aloysius	123-45-6789	40.0	90	100	83	49.0	D-	
Alfred	University	123-12-1234	41.0	97	96	97	48.0	D+	
Gerty	Gramma	567-89-0123	41.0	80	60	40	44.0	с	
Android	Electric	087-65-4321	42.0	23	36	45	47.0	В-	
Bumpkin	Fred	456-78-9012	43.0	78	88	77	45.0	A-	
Rubble	Betty	234-56-7890	44.0	90	80	90	46.0	C-	
Noshow	Cecil	345-67-8901	45.0	11	-1	4	43.0	F	
Buff	Bif	632-79-9939	46.0	20	30	40	50.0	B+	
Airpump	Andrew	223-45-6789	49.0 1.0	90	100	83	А	NA	
Backus	Jim	143-12-1234	48.0	1	97	96	97.0	A+	
Carnivore	Art	565-89-0123	44.0	1	80	60	40.0	D+	
Dandy	Jim	087-75-4321	47.0	1	23	36	45.0	C+	
Elephant	Ima	456-71-9012	45.0	1	78	88	77.0	B-	
Previewing f	irst 50 entries. 1 parsing	errors.							
Import Optio	ons:						Code Prev	iew:	Ċ
							librarv	(readr)	
Name: g	rades	✓ First Row as	Names Delimite	r: Comma	Escape:	None	grades	<- read_csv("grades.csv")	
Skip:	0	Trim Spaces	Quotes:	Default	<ul> <li>Comment:</li> </ul>	: Default	View(gr	ades)	
		✓ Open Data V	iewer Locale:	Configure	NA:	Default	•		
? Reading	rectangular data using	readr							Import Cancel
ouucu n om	, bonneouus, mbac	9 <u>1</u>						-	



### FILTER() FUNCTION

- The filter() function extracts rows by their value
- The syntax for filter is filter(data, logical\_expression\_for\_extracting\_rows)
- The filter function ignore rows that evaluates to FALSE and NA

Console	Terminal ×					
C:/Users/	gavin/Desktop/8	BAS/Workshop/R	workshop/Demo/	\$		
> filte	er(iris, Pet	tal.Width >	0.3)			
sep	al.Length s	Sepal.Width	Petal.Length	Petal.Width	Speches	
1	5.4	3.9	1.7	0.4	setosa	
2	5.7	4.4	1.5	0.4	setosa	
3	5.4	3.9	1.3	0.4	setosa	
4	5.1	3.7	1.5	0.4	setosa	
5	5.1	3.3	1.7	0.5	setosa	
6	5.0	3.4	1.6	0.4	setosa	
7	5.4	3.4	1.5	0.4	setosa	
8	5.0	3.5	1.6	0.6	setosa	
9	5.1	3.8	1.9	0.4	setosa	
10	7.0	3.2	4.7	1.4	versicolor	
11	6.4	3.2	4.5	1.5	versicolor	
12	6.9	3.1	4.9	1.5	versicolor	



### SELECT() FUNCTION

- The select() function extracts columns by their value
- The syntax for select() is select(data, col\_names\_condition)
- There are helpers function function for select():
  - start\_with(): select name that start with whatever you specify
  - ends\_with(): select name that ends with whatever you specify
  - contains(): select names that contain a specific string



#### SELECT() FUNCTION

Console	Terminal ×
C:/Users/	/gavin/Desktop/BAS/Workshop/R workshop/Demo/
> seled	ct(iris, Petal.Width)
Pet	tal.width
1	0.2
2	0.2
3	0.2
4	0.2
5	0.2
6	0.4
7	0.3
8	0.2
9	0.2
10	0.1
11	0.2
12	0.2

Console	Terminal ×	
C:/Users/	/gavin/Desktop/BAS/\	Norkshop/R workshop/De
> seled	ct(iris, ends_w	<pre>rith("Length"))</pre>
Sep	oal.Length Peta	l.Length
1	5.1	1.4
2	4.9	1.4
3	4.7	1.3
4	4.6	1.5
5	5.0	1.4
6	5.4	1.7
7	4.6	1.4
8	5.0	1.5
9	4.4	1.4
10	4.9	1.5
11	5.4	1.5
12	4.8	1.6
13	4.8	1.4



#### MUTATE() FUNCTION

- Compute new variable from existing ones and add it to the data table
- Syntax: mutate(data, var\_name = [operation w/ existing columns])

> m	utate(iris, lpw_p =	= Petal.Le	ngth/Petal.Widt	:h)			
	Sepal.Length Sepal	l.Width Pet	tal.Length Peta	l.Width	Species	lpw_p	
1	5.1	3.5	1.4	0.2	setosa	7.000000	
2	4.9	3.0	1.4	0.2	setosa	7.000000	
3	4.7	3.2	1.3	0.2	setosa	6.500000	
4	4.6	3.1	1.5	0.2	setosa	7.500000	
5	5.0	3.6	1.4	0.2	setosa	7.000000	
6	5.4	3.9	1.7	0.4	setosa	4.250000	
7	4.6	3.4	1.4	0.3	setosa	4.666667	
8	5.0	3.4	1.5	0.2	setosa	7.500000	
9	4.4	2.9	1.4	0.2	setosa	7.000000	
10	4.9	3.1	1.5	0.1	setosa	15.000000	
11	5.4	3.7	1.5	0.2	setosa	7.500000	
12	4.8	3.4	1.6	0.2	setosa	8.000000	
13	4.8	3.0	1.4	0.1	setosa	14.000000	
14	4.3	3.0	1.1	0.1	setosa	11.000000	
15	5.8	4.0	1.2	0.2	setosa	6.000000	
16	5.7	4.4	1.5	0.4	setosa	3.750000	
17	5.4	3.9	1.3	0.4	setosa	3.250000	
18	5.1	3.5	1.4	0.3	setosa	4.666667	
19	5.7	3.8	1.7	0.3	setosa	5.666667	



# PRACTICE TIME!





- 1. Using the filter() function, capture all the rows that has sepal width greater than 1 in the dataset iris
- 2. Using the select() function and its helper functions, capture all the columns that start with the letter "s"
- 3. Create a separate column in iris that contains the sum of the sepal width and length



## PLOTTING WITH GGPLOT2



#### WHAT IS GGPLOT2?

- ggplot2 is a R package that is very useful for data visualization
- "gg" stands for grammars of graphics
- It provides a systematic way to create visually-pleasing graphs
- Basic syntax
  - ggplot(data = data\_name, aes(x=..., y=...)) + geom\_...



#### SCATTERPLOT

- To create a scatterplot, we use the geom\_point() function
  - ggplot(data = data\_name, aes(x=..., y=...)) + geom\_point()







#### LINE PLOT

- To create a line plot, we use the geom\_line() function
  - ggplot(data = data\_name, aes(x=..., y=...)) + geom\_line()



```
3 library(tidyverse)
4 |
5 plot2 <- ggplot(data = economics,
6 aes(date, unemploy))+
7 geom_line()+
8 plot2</pre>
```



#### HISTOGRAM

- To create a histogram, we use the geom\_histogram() function
  - ggplot(data = data\_name, aes(x=..., y=...)) + geom\_histogram()





# PRACTICE TIME!





- 1. Type on mtcars on your script or console and take a look at the data. We will be using this dataset
- 2. Create a scatterplot for "disp" and "mpg" and save it as splot
- 3. Create a line graph for "wt" and "qsec" and save it as line\_graph
- 4. Create histogram for "cyl" and save it as histo



#### HOW TO MAKE PLOT PRETTIER

- We can specify additional elements in the ggplot to make plots look nicer!
- Example
  - Rename axis
  - Add a title
  - Color
  - Themes
  - Add a smoother function
  - And more!



#### LINE PLOT REVISITED

```
library(tidyverse)
 3
 4
 5
    install.packages("ggthemes")
    library(ggthemes)
 6
 7
    plot2_nice <- ggplot(data = economics,</pre>
 8
9
                     aes(date, unemploy))+
               geom_line(size = 1, color = "blue")+
               xlab("Year")+
10
11
               ylab("Unemployment")+
12
               ggtitle("Unemployment over Time")+
13
               theme_economist()
14
    plot2_nice
15
```





#### GGPLOT2 IS VERY POWERFUL



Parallel Coordinate Plot for the Iris Data



BRUIN ACTUARIAL SOCIETY



# **Question?**