Society of Actuaries in Ireland

Data Processing with R

10th September 2018

Disclaimer

The views expressed in this presentation are those of the presenter(s) and not necessarily of the Society of Actuaries in Ireland

Agenda

Part 1

- Types of Data Processing
- Data Manipulation
- Data Generation
- Data Analysis
- Other Solutions

Part 2

- Introduction to dplyr
- Tips & Tricks
- Further Support

Types of Data Processing

- **Data Manipulation** – Restructure/adjusting existing data sets
- **Data Generation** – Creation of new data sets
- **Data Analysis** – Summarise the messages from existing data sets

Agenda

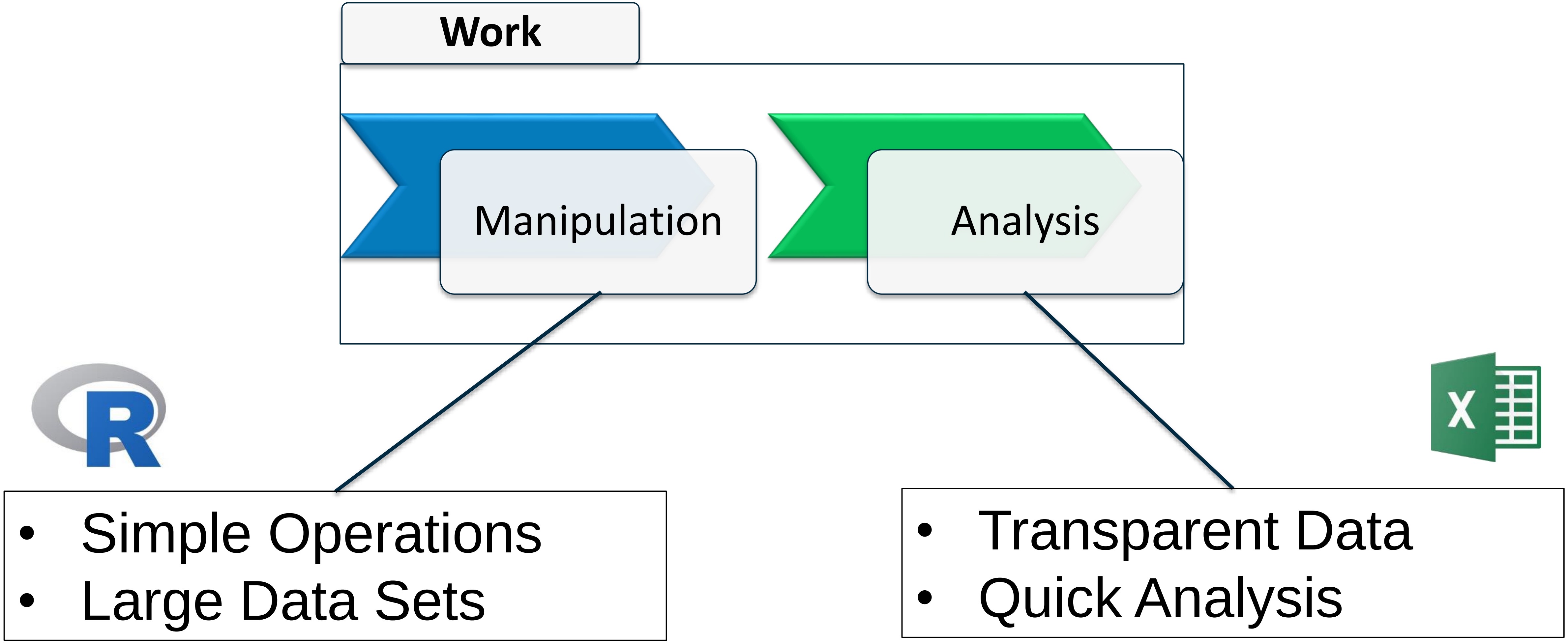
Part 1

- Types of Data Processing
- **Data Manipulation**
- Data Generation
- Data Analysis
- Other Solutions

Part 2

- Introduction to dplyr
- Tips & Tricks
- Further Support

Data Manipulation



Work

Manipulation

Analysis

- Simple Operations
- Large Data Sets

- Transparent Data
- Quick Analysis

Data Manipulation – Benefits of R

	R
Data Volumes	✓
Robust	✓
Audit Trail	✓
Development Time	?

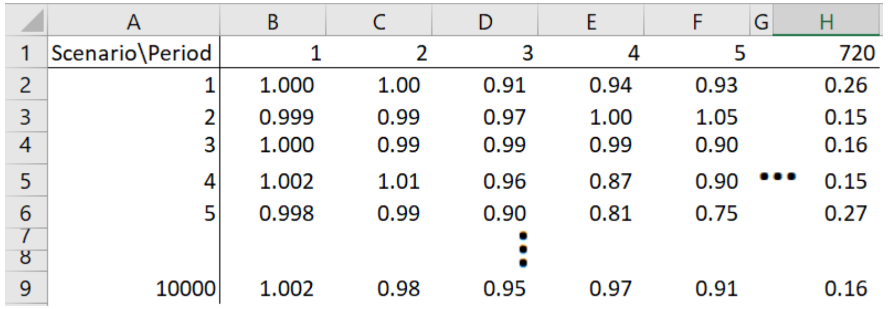
Example 1 – Data Manipulation – The Problem

- We required discounted cashflows from the cashflow model
- Our cashflow model produces undiscounted cashflows for each scenarios
- Large number of files (48) totalling 4GB in file size

- Need to simplify the data set, discount cashflows, average values over the scenarios

Example 1 – Data Manipulation – Excel Solution

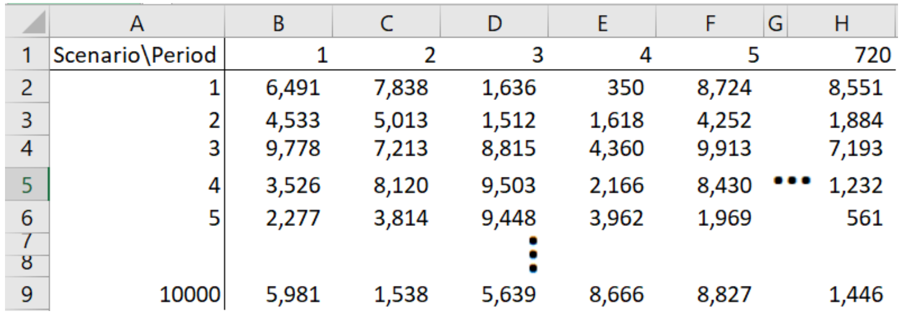
Discount:



	A	B	C	D	E	F	G	H
1	Scenario\Period	1	2	3	4	5		720
2	1	1.000	1.00	0.91	0.94	0.93		0.26
3	2	0.999	0.99	0.97	1.00	1.05		0.15
4	3	1.000	0.99	0.99	0.99	0.90		0.16
5	4	1.002	1.01	0.96	0.87	0.90	…	0.15
6	5	0.998	0.99	0.90	0.81	0.75		0.27
7				⋮				
8								
9	10000	1.002	0.98	0.95	0.97	0.91		0.16

Multiple By

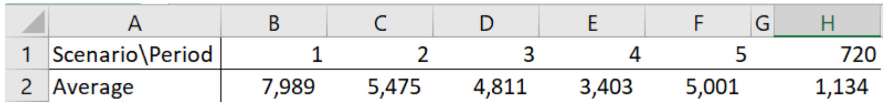
Cashflow:



	A	B	C	D	E	F	G	H
1	Scenario\Period	1	2	3	4	5		720
2	1	6,491	7,838	1,636	350	8,724		8,551
3	2	4,533	5,013	1,512	1,618	4,252		1,884
4	3	9,778	7,213	8,815	4,360	9,913		7,193
5	4	3,526	8,120	9,503	2,166	8,430	…	1,232
6	5	2,277	3,814	9,448	3,962	1,969		561
7				⋮				
8								
9	10000	5,981	1,538	5,639	8,666	8,827		1,446

Average of Columns

Output:

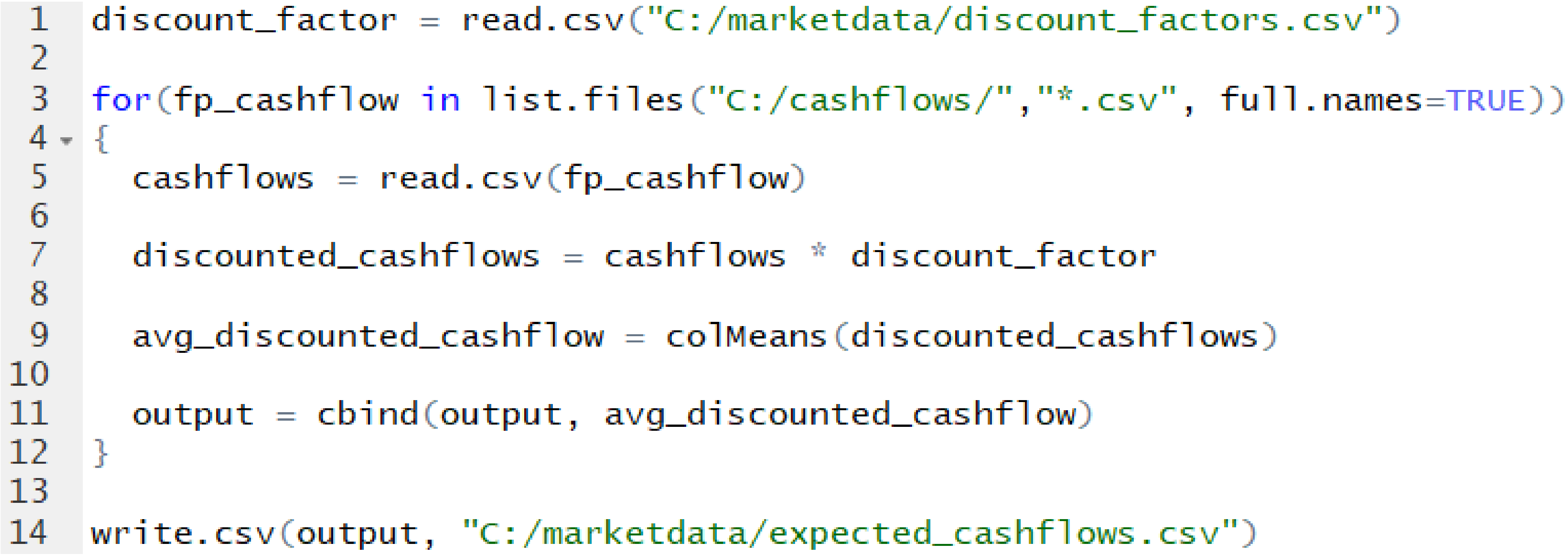


	A	B	C	D	E	F	G	H
1	Scenario\Period	1	2	3	4	5		720
2	Average	7,989	5,475	4,811	3,403	5,001		1,134

Possible Excel Setup:

1. Use VBA to process the data
2. Read in a cashflow
3. Recalculate worksheets
4. Save averages to an output sheet

Example 1 – Data Manipulation – R Solution



```r
discount_factor = read.csv("C:/marketdata/discount_factors.csv")

for(fp_cashflow in list.files("C:/cashflows/","*.csv", full.names=TRUE))
{
  cashflows = read.csv(fp_cashflow)

  discounted_cashflows = cashflows * discount_factor

  avg_discounted_cashflow = colMeans(discounted_cashflows)

  output = cbind(output, avg_discounted_cashflow)
}

write.csv(output, "C:/marketdata/expected_cashflows.csv")
```

When to Use VBA and Why

“VBA programming is a powerful solution, but it is not always the optimal approach.

Sometimes it makes sense to use other ways to achieve your aims.”   Microsoft

https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office#when-to-use-vba-and-why

Agenda

Part 1

- Types of Data Processing
- Data Manipulation
- **Data Generation**
- Data Analysis
- Other Solutions

Part 2

- Introduction to dplyr
- Tips & Tricks
- Further Support

Data Generation

- Generate new data sets based on parameter inputs
- Often based on random number generation
- Leverage R’s statistical capabilities - “*R is a language and environment for statistical computing*”
- R solutions are quite practical
  - Set.seed reproducible in a single R version
  - RNGversion reproducible across different versions

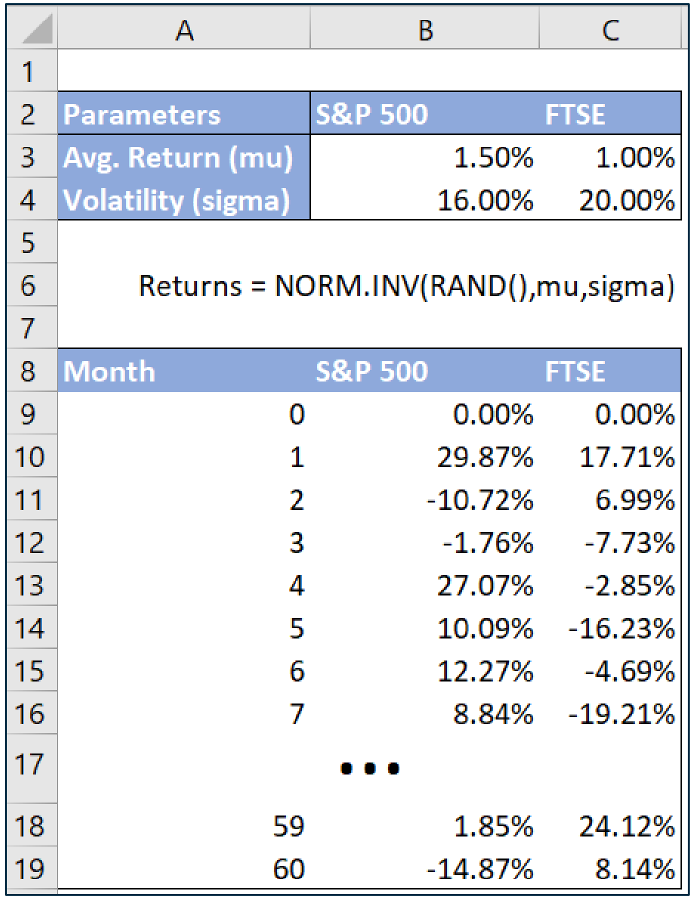
Data Generation – Benefits of R

	R
Statistical Packages	✓
Performance	✓
Automation	✓

Example 2 – Data Generation – The Problem

- Simulate returns on a number of indexes (eg S&P500, FTSE100)
- Assume returns are normally distributed & correlated
- Require 60 years of output & 10k scenarios

Example 2 – Data Generation – Excel Solution

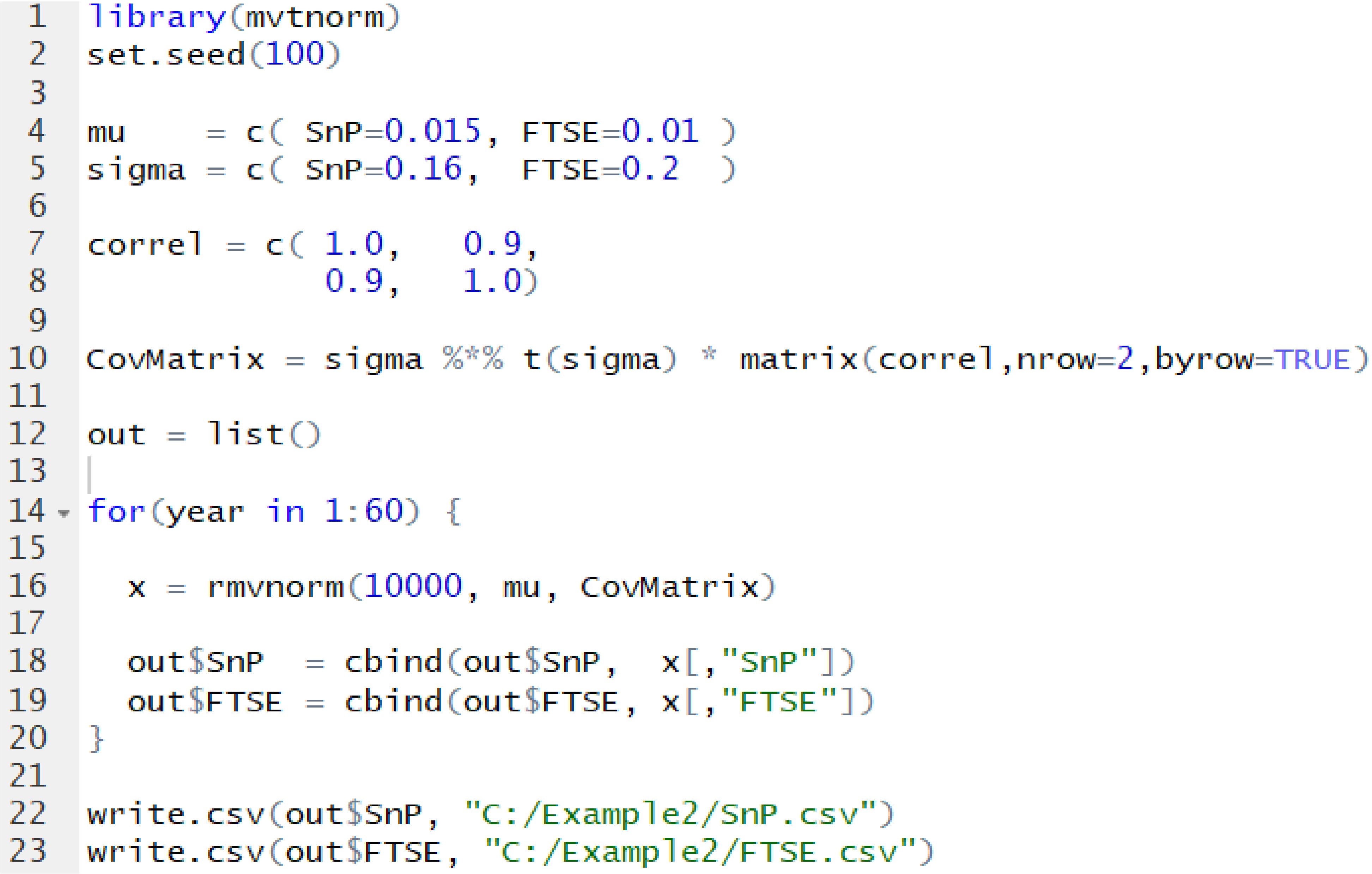


	A	B	C
1			
2	Parameters	S&P 500	FTSE
3	Avg. Return (mu)	1.50%	1.00%
4	Volatility (sigma)	16.00%	20.00%
5			
6	Returns = NORM.INV(RAND(),mu,sigma)		
7			
8	Month	S&P 500	FTSE
9	0	0.00%	0.00%
10	1	29.87%	17.71%
11	2	-10.72%	6.99%
12	3	-1.76%	-7.73%
13	4	27.07%	-2.85%
14	5	10.09%	-16.23%
15	6	12.27%	-4.69%
16	7	8.84%	-19.21%
17	• • •		
18	59	1.85%	24.12%
19	60	-14.87%	8.14%

Possible Excel Setup:

1. Use Excel to simulate returns
2. Use VBA to loop over scenarios and save to file
3. What about correlations?
4. What about repeatable random numbers?
5. Performance?
6. How scalable will our solution be?

Example 2 – Data Generation – R Solution



```r
library(mvtnorm)
set.seed(100)

mu    = c( SnP=0.015, FTSE=0.01 )
sigma = c( SnP=0.16,  FTSE=0.2  )

correl = c( 1.0,   0.9,
            0.9,   1.0)

CovMatrix = sigma %*% t(sigma) * matrix(correl,nrow=2,byrow=TRUE)

out = list()

for(year in 1:60) {

  x = rmvnorm(10000, mu, CovMatrix)

  out$SnP  = cbind(out$SnP,  x[,"SnP"])
  out$FTSE = cbind(out$FTSE, x[,"FTSE"])
}

write.csv(out$SnP, "C:/Example2/SnP.csv")
write.csv(out$FTSE, "C:/Example2/FTSE.csv")
```

Agenda

Part 1

- Types of Data Processing
- Data Manipulation
- Data Generation
- **Data Analysis**
- Other Solutions

Part 2

- Introduction to dplyr
- Tips & Tricks
- Further Support

Data Analysis

- Summarise existing data sets (averages, standard deviations, percentiles etc)
- Produce statistical analysis on data sets (eg p-tests)
- Visualising data sets
- Fitting models

Data Analysis – Benefits of R

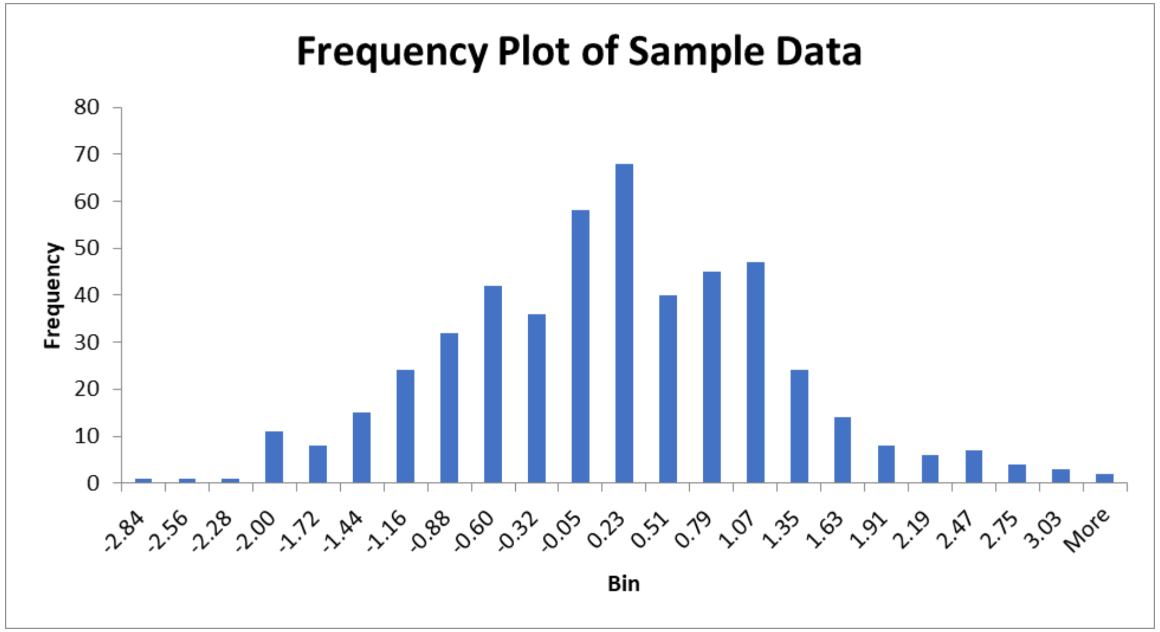
	R
Statistical Packages	✓
Graphing Capabilities	✓
Online Support	✓
Data Transparency	?

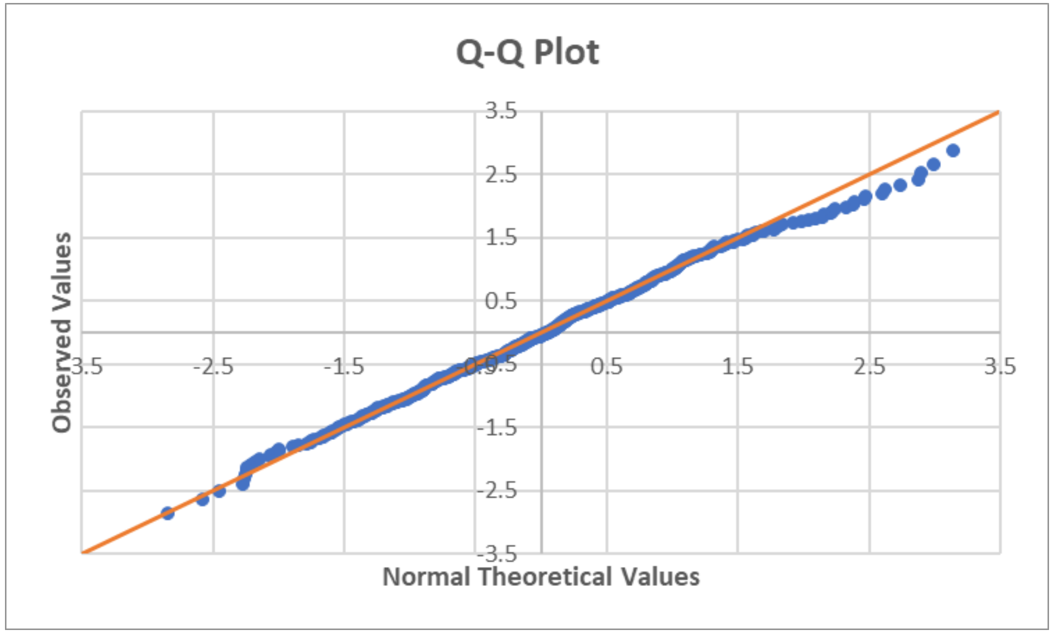
Example 3 – Data Analysis – The Problem

- Perform a normality test on a data set (500 observations)

Example 3 – Data Analysis – Excel Solution

- Graph the frequency using a histogram (Data Analysis add-in)
- Produce a Q-Q plot (Scatter Plot + Line)
- Perform a Hypothesis Tests (P-Value approach). How?

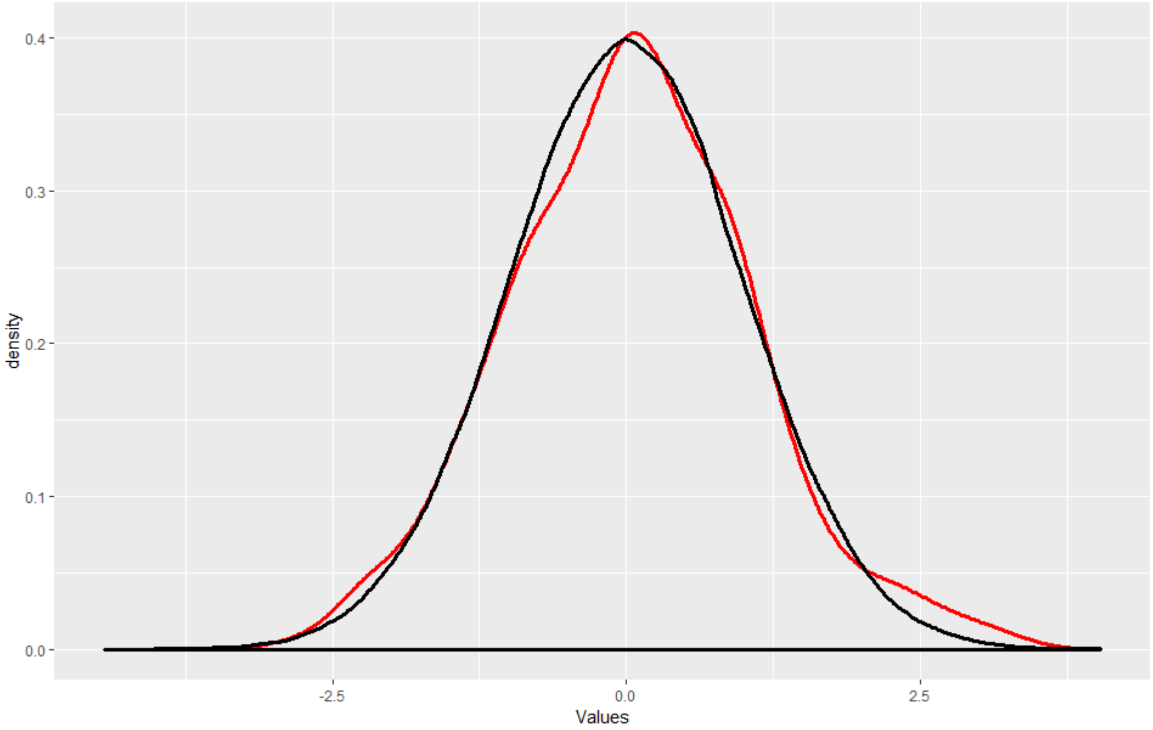




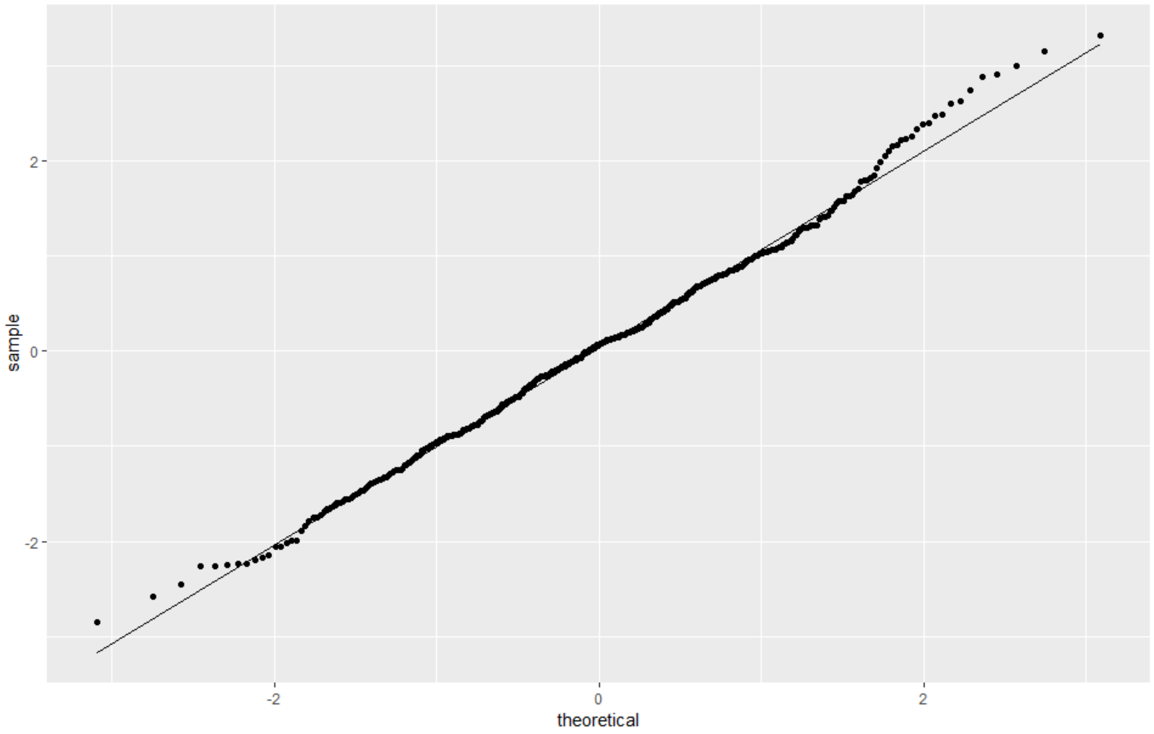
Example 3 – Data Analysis – R Solution

- Graph the density plot (using ggplot2)
- Produce a Q-Q plot (using ggplot2)
- Perform a Hypothesis Tests (P-Value approach)

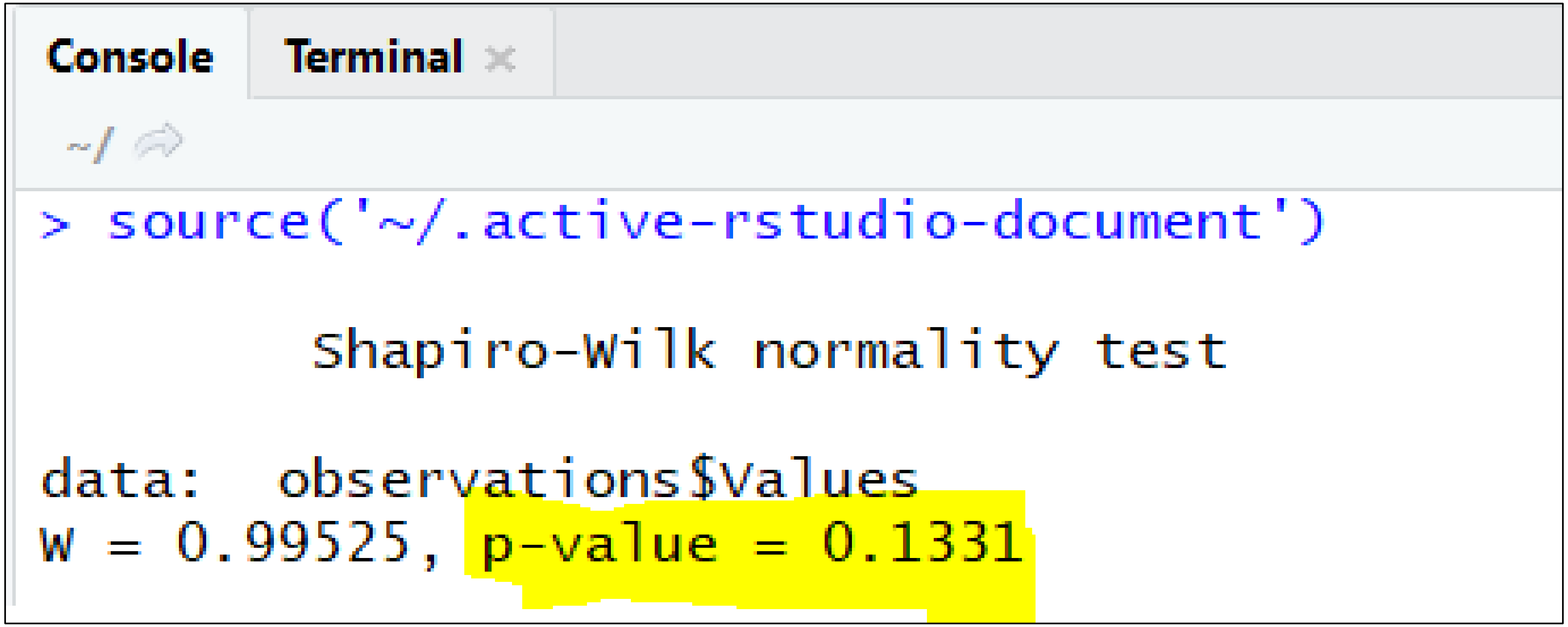
Density Graph



Q-Q Plot



Example 3 – Data Analysis – R Solution

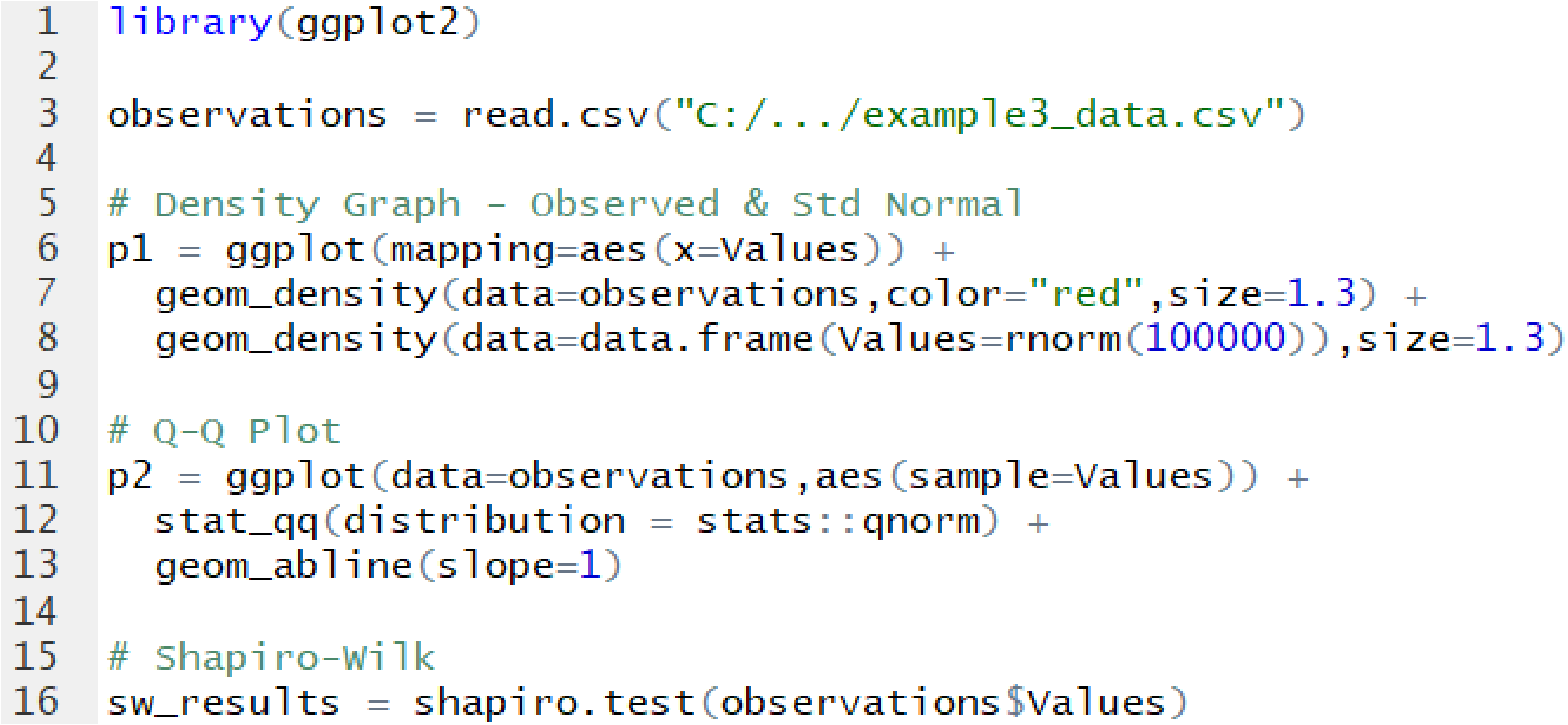


```
> source('~/.active-rstudio-document')

        Shapiro-Wilk normality test

data:  observations$Values
W = 0.99525, p-value = 0.1331
```

Example 3 – Data Analysis – R Solution



```r
library(ggplot2)

observations = read.csv("C:/.../example3_data.csv")

# Density Graph - Observed & Std Normal
p1 = ggplot(mapping=aes(x=Values)) +
  geom_density(data=observations,color="red",size=1.3) +
  geom_density(data=data.frame(Values=rnorm(100000)),size=1.3)

# Q-Q Plot
p2 = ggplot(data=observations,aes(sample=Values)) +
  stat_qq(distribution = stats::qnorm) +
  geom_abline(slope=1)

# Shapiro-Wilk
sw_results = shapiro.test(observations$Values)
```

Example 3 – Data Analysis – Remark

- There is a unintentional mistake in the Excel Q-Q chart
- Expected/Observations axis are the wrong way around
- Error not realised until compared against results from R

Model & Code Review

Agenda

Part 1

- Types of Data Processing
- Data Manipulation
- Data Generation
- Data Analysis
- **Other Solutions**

Part 2

- Introduction to dplyr
- Tips & Tricks
- Further Support

Other Solutions (Non-Excel)

?

Other Solutions

- Level of Industry Knowledge
- Online Support
- Interoperability
- Extensibility
- Cost

Drawbacks of R

⛔

R

- Low Industry Experience (eg Code Review)
- Steep Learning Curve
- Language Inconsistencies / Multiple Syntaxes
- Performance

Agenda

Part 1

- Types of Data Processing
- Data Manipulation
- Data Generation
- Data Analysis
- Other Solutions

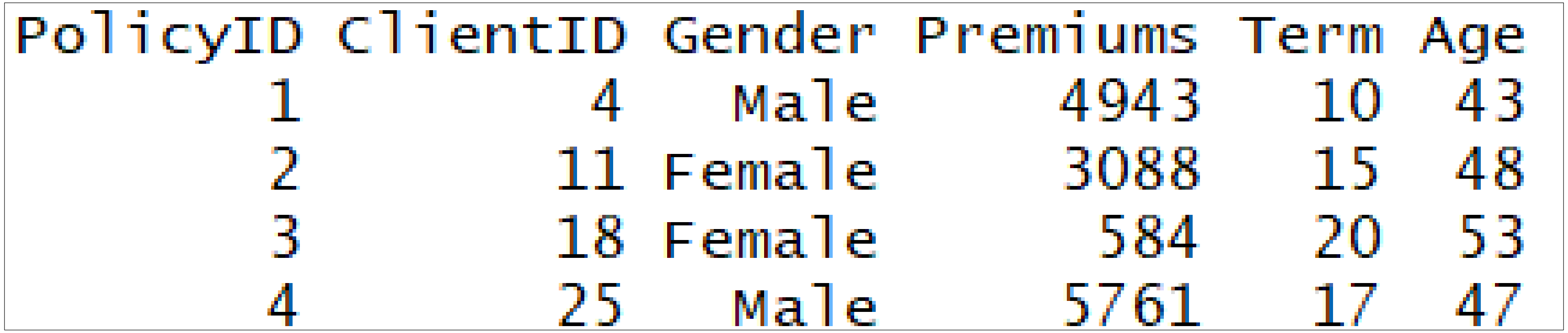
Part 2

- **Introduction to dplyr**
- Tips & Tricks
- Further Support

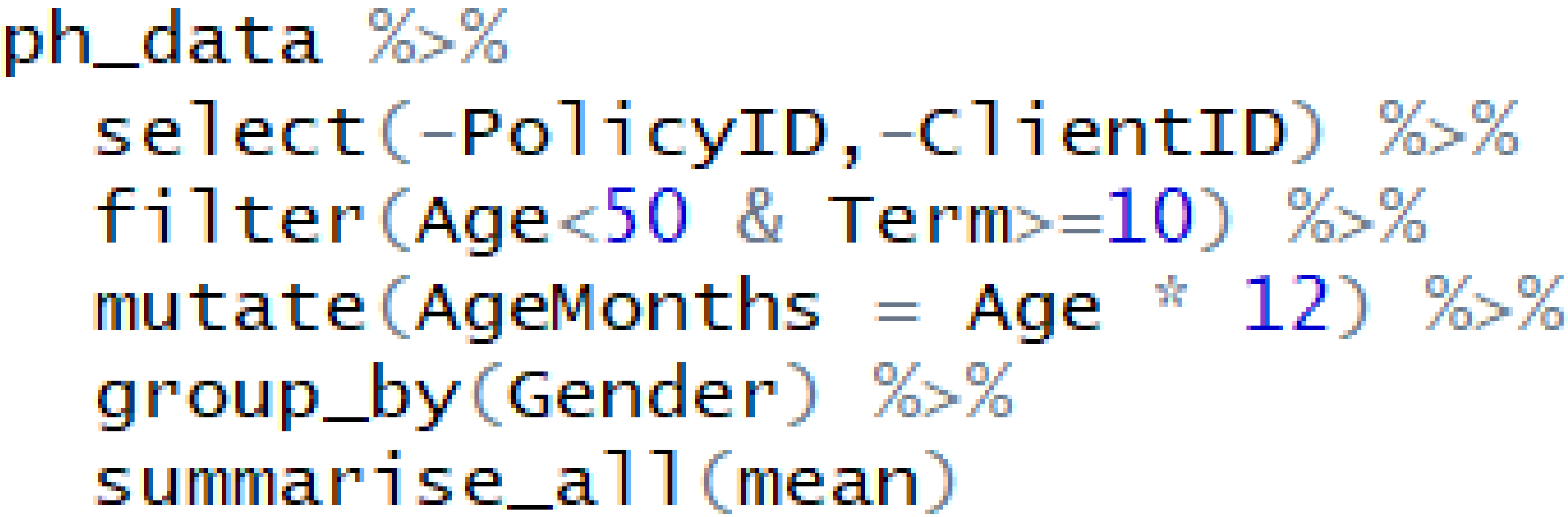
Introduction to dplyr

- dpylr is a package you can download for R
  - install.packages(“dpylr”)
  - Rstudio > Tools > Install Packages > search for dpylr
- Provides SQL like abilities to query and modify tables, all within R
- dplyr is optimised for data analysis
  - Very rich & powerful commands
  - Syntax is intuitive
- Very good document and online support

Introduction to dplyr - Example

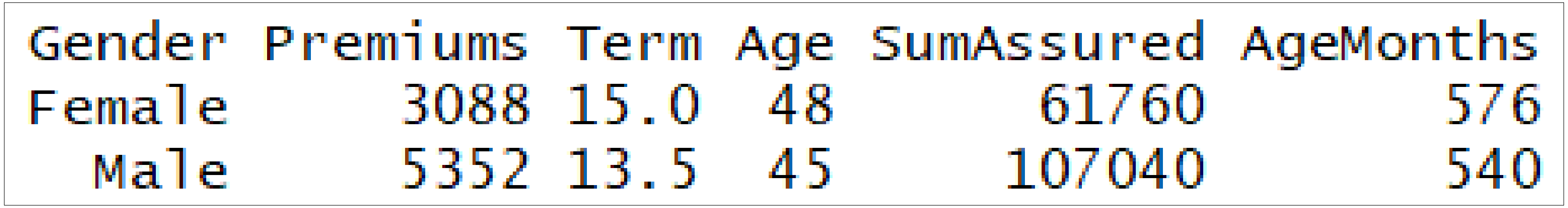


```
PolicyID ClientID Gender Premiums Term Age
       1        4   Male     4943   10  43
       2       11 Female     3088   15  48
       3       18 Female      584   20  53
       4       25   Male     5761   17  47
```



```
ph_data %>%
  select(-PolicyID,-ClientID) %>%
  filter(Age<50 & Term>=10) %>%
  mutate(AgeMonths = Age * 12) %>%
  group_by(Gender) %>%
  summarise_all(mean)
```

Introduction to dplyr - Example



```
Gender Premiums Term Age SumAssured AgeMonths
Female     3088 15.0  48      61760       576
  Male     5352 13.5  45     107040       540
```

Agenda

Part 1

- Types of Data Processing
- Data Manipulation
- Data Generation
- Data Analysis
- Other Solutions

Part 2

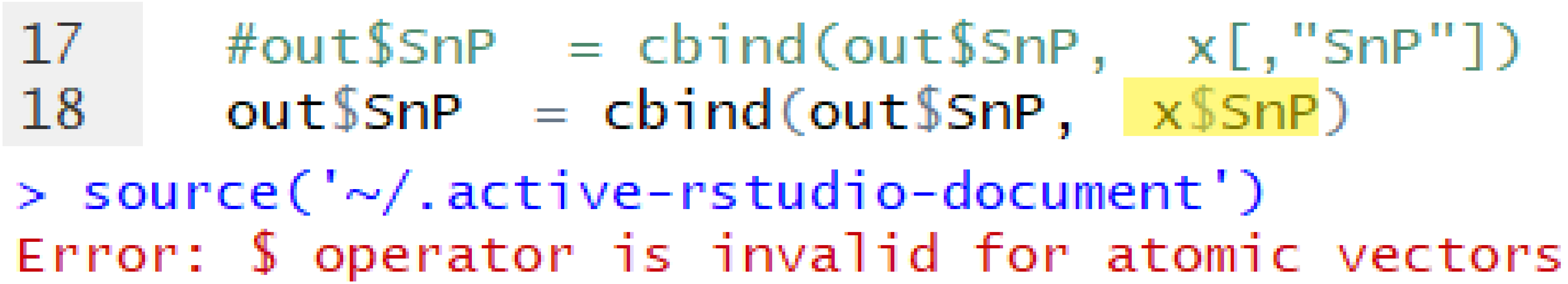
- Introduction to dplyr
- **Tips & Tricks**
- Further Support

Tips & Tricks

R Command	Description
rm(list=ls())	Put at start of script. Removes all variables. Ensures no unintentional picking up of variables
class/str	Inspect the data type of a variable
head, tail, dim	Inspect the beginning/end of the data. Dim prints the dimension (how many rows, cols)
set.seed	Ensures random numbers are reproducible
stringAsFactors=FALSE	Factors may not be what you
data.table	**Very** fast reading (fread) and writing (fwrite) to CSV files
traceback	Shows the line of code what caused the error

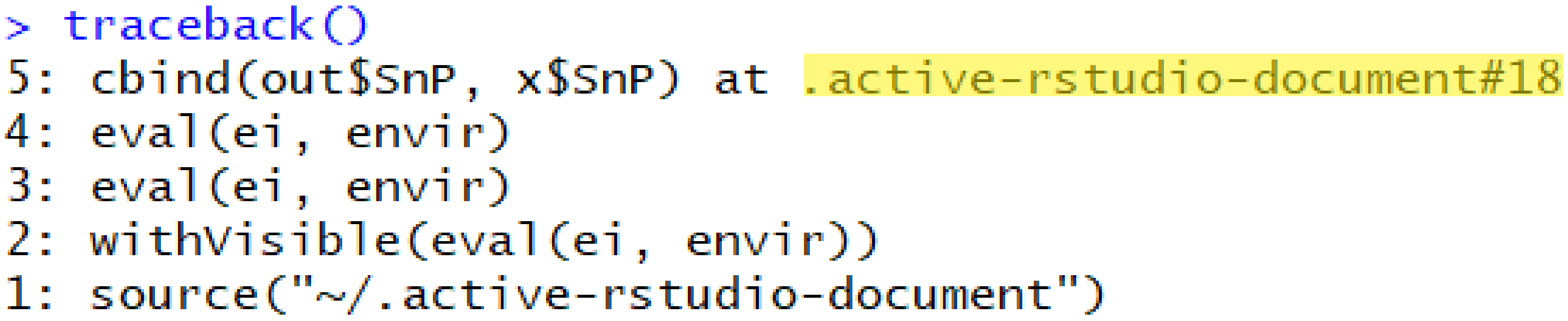
Tips & Tricks - Traceback

Introduce an error into Example 3:



```
17    #out$SnP  = cbind(out$SnP,  x[,"SnP"])
18    out$SnP  = cbind(out$SnP,  x$SnP)
> source('~/.active-rstudio-document')
Error: $ operator is invalid for atomic vectors
```

Traceback:



```
> traceback()
5: cbind(out$SnP, x$SnP) at .active-rstudio-document#18
4: eval(ei, envir)
3: eval(ei, envir)
2: withVisible(eval(ei, envir))
1: source("~/.active-rstudio-document")
```

Agenda

Part 1

- Types of Data Processing
- Data Manipulation
- Data Generation
- Data Analysis
- Other Solutions

Part 2

- Introduction to dplyr
- Tips & Tricks
- **Further Support**

Further Support

- Online tutorials: datacamp.com (paid & free)
- Classroom based support: companies in Dublin running courses
- Individual Questions: google / stackoverflow.com

Recap

Part 1

- Types of Data Processing
- Data Manipulation
- Data Generation
- Data Analysis
- Other Solutions

Part 2

- Introduction to dplyr
- Tips & Tricks
- Further Support

Q&A