UNIVERSITY OF LINCOLN

Importing, transforming, and summarising your data

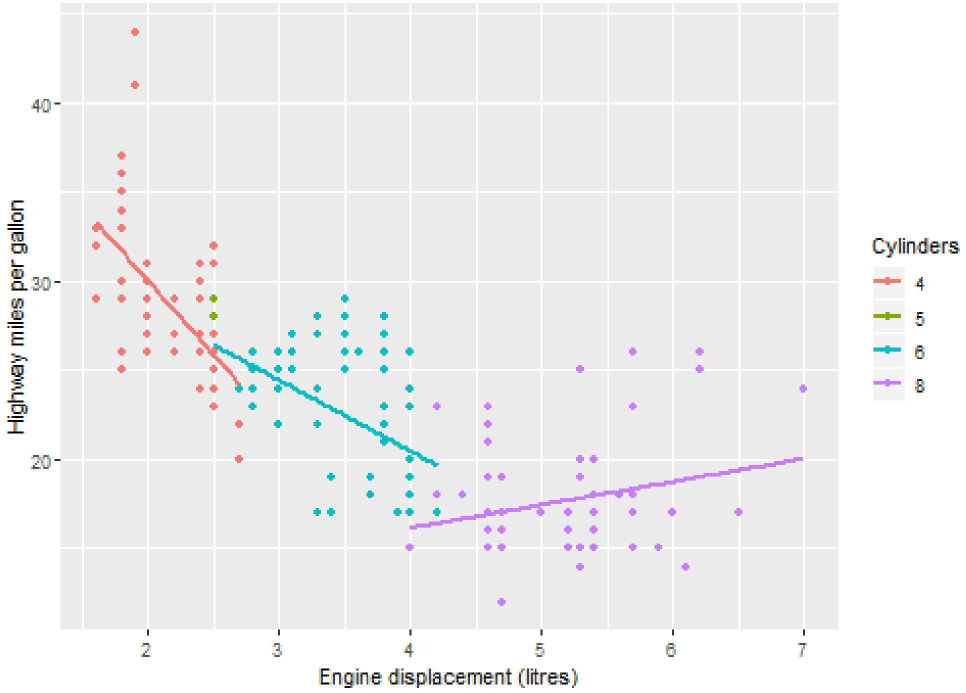
PSY9219M - Research Methods and Skills

Dr Matt Craddock

23/10/2018

Plotting using ggplot2

```r
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     colour = factor(cyl))) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Engine displacement (litres)",
       y = "Highway miles per gallon",
       colour = "Cylinders")
```



A quick reminder

For anyone that hasn't done this already, join the PSY9219M workspace on RStudio.cloud.

Link removed from the public version :)

Importing your data

Different types of file

Data comes in many different shapes, sizes, and formats.

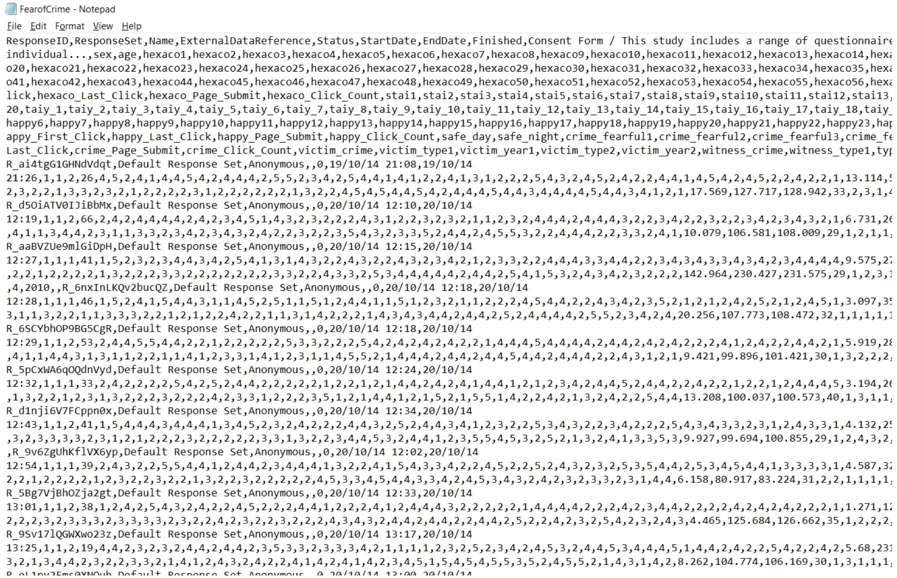
The most common file formats you'll deal with are either Excel files or text files, but you may also find dealing with SPSS files useful.

Fortunately, R has several functions and packages for importing data!

File formats	File extension	Functions	Package
SPSS	.sav	**read_sav()**	library(haven)
Excel	.xls, .xlsx	**read_excel()**	library(readxl)
Text	.csv, .txt, .*	**read_csv()**, **read_delim()**	library(readr)

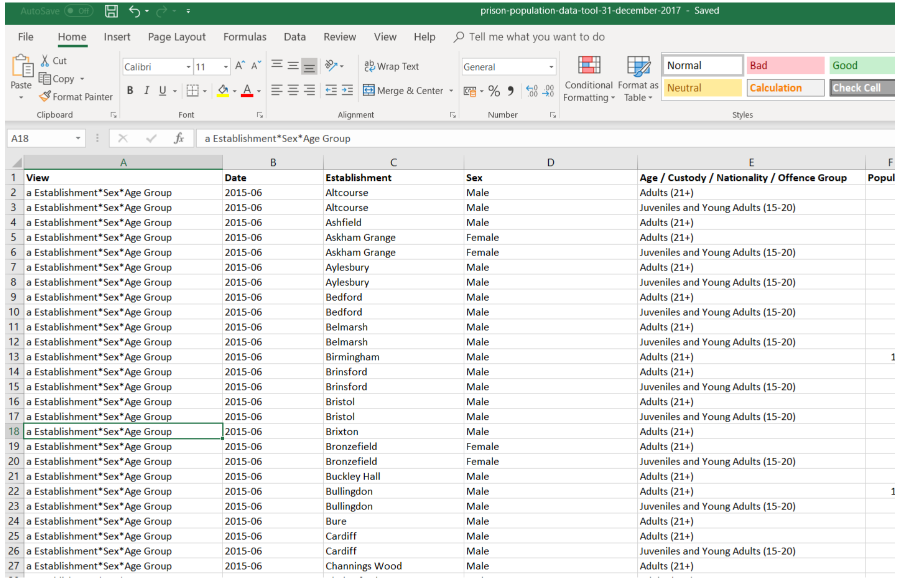
Importing data into R

Comma-separated values



```
ResponseID,ResponseSet,Name,ExternalDataReference,Status,StartDate,EndDate,Finished,Consent Form / This study includes a range of questionnaire
individual...,sex,age,hexaco1,hexaco2,hexaco3,hexaco4,hexaco5,hexaco6,hexaco7,hexaco8,hexaco9,hexaco10,hexaco11,hexaco12,hexaco13,hexaco14,hexa
o20,hexaco21,hexaco22,hexaco23,hexaco24,hexaco25,hexaco26,hexaco27,hexaco28,hexaco29,hexaco30,hexaco31,hexaco32,hexaco33,hexaco34,hexaco35,hexa
o41,hexaco42,hexaco43,hexaco44,hexaco45,hexaco46,hexaco47,hexaco48,hexaco49,hexaco50,hexaco51,hexaco52,hexaco53,hexaco54,hexaco55,hexaco56,hexa
lick,hexaco_Last_Click,hexaco_Page_Submit,hexaco_Click_Count,stai1,stai2,stai3,stai4,stai5,stai6,stai7,stai8,stai9,stai10,stai11,stai12,stai13,
20,taiy_1,taiy_2,taiy_3,taiy_4,taiy_5,taiy_6,taiy_7,taiy_8,taiy_9,taiy_10,taiy_11,taiy_12,taiy_13,taiy_14,taiy_15,taiy_16,taiy_17,taiy_18,taiy_
happy6,happy7,happy8,happy9,happy10,happy11,happy12,happy13,happy14,happy15,happy16,happy17,happy18,happy19,happy20,happy21,happy22,happy23,ha
appy_First_Click,happy_Last_Click,happy_Page_Submit,happy_Click_Count,safe_day,safe_night,crime_fearful1,crime_fearful2,crime_fearful3,crime_f
Last_Click,crime_Page_Submit,crime_Click_Count,victim_crime,victim_type1,victim_year1,victim_type2,victim_year2,witness_crime,witness_type1,ty
R_ai4tgG1GHNdVdqt,Default Response Set,Anonymous,,0,19/10/14 21:08,19/10/14
21:26,1,1,2,26,4,5,2,4,1,4,4,5,4,2,4,4,4,2,5,5,2,3,4,2,5,4,4,1,4,1,2,2,4,1,3,1,2,2,2,5,4,3,2,4,5,2,4,2,2,4,4,1,4,5,4,2,4,5,2,2,4,2,2,1,13.114,
2,3,2,2,1,3,3,2,3,2,1,2,2,2,2,3,1,2,2,2,2,2,2,1,3,2,2,4,5,4,5,4,4,5,4,2,4,4,4,5,4,4,3,4,4,4,4,5,4,4,3,4,1,2,1,17.569,127.717,128.942,33,2,3,1,
R_d5OiATV0IJiBbMx,Default Response Set,Anonymous,,0,20/10/14 12:10,20/10/14
12:19,1,1,2,66,2,4,2,4,4,4,4,2,4,2,3,4,5,1,4,3,2,3,2,2,2,4,3,1,2,2,3,2,3,2,1,1,2,3,2,4,4,4,2,4,4,4,3,2,2,3,4,2,2,3,2,2,3,4,2,3,4,3,2,1,6.731,2
,4,1,1,3,4,4,2,3,1,1,3,3,2,3,4,2,3,4,3,2,4,2,2,3,2,2,3,5,4,3,2,3,3,5,2,4,4,2,4,5,5,3,2,2,4,4,4,2,2,3,3,2,4,1,10.079,106.581,108.009,29,1,2,1,1,
R_aaBVZUe9mlGiDpH,Default Response Set,Anonymous,,0,20/10/14 12:15,20/10/14
12:27,1,1,1,41,1,5,2,3,2,3,4,4,3,4,2,5,4,1,3,1,4,3,2,2,4,3,2,2,4,3,2,3,4,2,1,2,3,3,2,2,4,4,4,3,3,4,4,2,2,3,4,3,4,3,3,4,3,4,2,3,4,4,4,4,9.575,2
,2,2,1,2,2,2,2,1,3,2,2,2,3,3,2,2,2,2,2,2,2,3,3,2,2,4,3,3,2,5,3,4,4,4,4,4,2,4,4,2,5,4,1,5,3,2,4,3,4,2,3,2,2,2,142.964,230.427,231.575,29,1,2,3,
,4,2010,,R_6nxInLKQv2bucQZ,Default Response Set,Anonymous,,0,20/10/14 12:18,20/10/14
12:28,1,1,1,46,1,5,2,4,1,5,4,4,3,1,1,4,5,2,5,1,1,5,1,2,4,4,1,1,5,1,2,3,2,1,1,2,2,2,4,5,4,4,2,2,4,3,4,2,3,5,2,1,2,1,2,4,2,5,2,1,2,4,5,1,3.097,3
3,1,1,3,2,2,1,1,3,3,3,2,2,1,2,1,2,2,4,2,2,1,1,3,1,4,2,2,2,1,4,3,4,3,4,4,2,4,4,2,5,2,4,4,4,4,2,5,5,2,3,4,2,4,20.256,107.773,108.472,32,1,1,1,1,
R_6SCYbhOP9BG5CgR,Default Response Set,Anonymous,,0,20/10/14 12:18,20/10/14
12:29,1,1,2,53,2,4,4,5,5,4,4,2,2,1,2,2,2,2,2,5,3,3,2,2,2,5,4,2,4,2,4,2,4,2,2,1,4,4,4,5,4,4,4,4,2,4,4,2,4,2,4,2,2,2,4,1,2,4,2,2,4,4,2,1,5.919,2
,4,1,1,4,4,3,1,3,1,1,2,2,1,1,4,1,2,3,3,1,4,1,2,3,1,1,4,5,5,2,1,4,4,4,2,4,4,2,4,4,5,4,4,2,4,4,4,2,2,4,3,1,2,1,9.421,99.896,101.421,30,1,3,2,2,2,
R_5pCxWA6qOQdnVyd,Default Response Set,Anonymous,,0,20/10/14 12:24,20/10/14
12:32,1,1,1,33,2,4,2,2,2,2,5,4,2,5,2,4,4,2,2,2,2,2,1,2,2,1,2,1,4,4,2,4,2,4,1,4,4,1,2,1,2,3,4,2,4,4,5,2,4,4,2,2,4,2,2,1,2,2,1,2,4,4,4,5,3.194,2
,1,3,2,2,1,2,3,1,3,2,2,2,3,2,2,2,4,2,3,3,1,2,2,2,3,5,1,2,1,4,4,1,2,1,5,2,1,5,5,1,4,2,2,4,2,1,3,2,4,2,2,5,4,4,13.208,100.037,100.573,40,1,3,1,1,
R_d1nji6V7FCppn0x,Default Response Set,Anonymous,,0,20/10/14 12:34,20/10/14
12:43,1,1,2,41,1,5,4,4,4,3,4,4,4,1,3,4,5,2,3,2,4,2,2,2,4,4,3,2,5,2,4,4,3,4,1,2,3,2,2,5,3,4,3,2,2,3,4,2,2,2,5,4,3,4,3,3,2,3,1,2,4,3,3,1,4.132,2
,3,2,3,3,3,3,2,3,1,2,1,2,2,2,3,2,2,2,2,2,3,3,1,3,2,2,3,4,4,5,3,2,4,4,1,2,3,5,5,4,5,3,2,5,2,1,3,2,4,1,3,3,5,3,9.927,99.694,100.855,29,1,2,4,3,2,
,R_9v6ZgUhKflVX6yp,Default Response Set,Anonymous,,0,20/10/14 12:02,20/10/14
12:54,1,1,1,39,2,4,3,2,2,5,5,4,4,1,2,4,4,2,3,4,4,4,1,3,2,2,4,1,5,4,3,3,4,2,2,4,5,2,2,5,2,4,3,2,3,2,5,3,5,4,4,2,5,3,4,5,4,4,1,3,3,3,3,1,4.587,3
2,2,1,2,2,2,2,1,2,3,2,2,3,2,2,1,3,2,2,3,2,2,2,2,2,4,5,3,3,4,5,4,4,3,3,4,2,4,5,3,4,3,2,4,2,3,2,3,3,2,3,1,4,4,6.158,80.917,83.224,31,2,2,1,1,1,1,
R_5Bg7VjBhOZja2gt,Default Response Set,Anonymous,,0,20/10/14 12:33,20/10/14
13:01,1,1,2,38,1,2,4,2,5,4,3,2,4,2,2,4,5,2,2,2,4,4,1,2,2,2,4,1,2,4,4,4,3,2,2,2,2,2,1,4,4,4,4,2,2,2,4,2,3,4,4,2,2,2,2,4,2,4,2,4,2,2,2,1,1.271,1
2,2,2,3,2,3,3,3,2,3,3,3,2,3,2,2,4,2,3,2,2,3,2,2,2,4,3,4,3,2,4,4,2,4,4,2,2,4,4,2,5,2,2,4,2,3,2,5,4,2,3,2,4,3,4.465,125.684,126.662,35,1,2,2,2,
R_9Sv17lQGWXwo23z,Default Response Set,Anonymous,,0,20/10/14 13:17,20/10/14
13:25,1,1,2,19,4,4,2,3,2,3,2,4,4,2,4,4,2,3,5,3,3,2,3,3,3,4,2,1,1,1,1,2,3,2,5,2,3,4,2,4,5,3,2,4,4,5,3,4,4,4,5,1,4,4,2,4,2,2,5,4,2,2,4,2,5.68,23
3,2,1,3,4,4,2,3,2,2,3,3,2,1,4,1,2,4,3,2,4,2,2,4,1,4,2,4,1,4,2,3,4,5,1,5,4,5,4,5,5,3,5,2,4,5,5,2,1,4,3,1,4,2,8.262,104.774,106.169,30,1,3,1,1,1,
R_oL1py2Fms0XNOuh,Default Response Set,Anonymous,,0,20/10/14 13:00,20/10/14
```

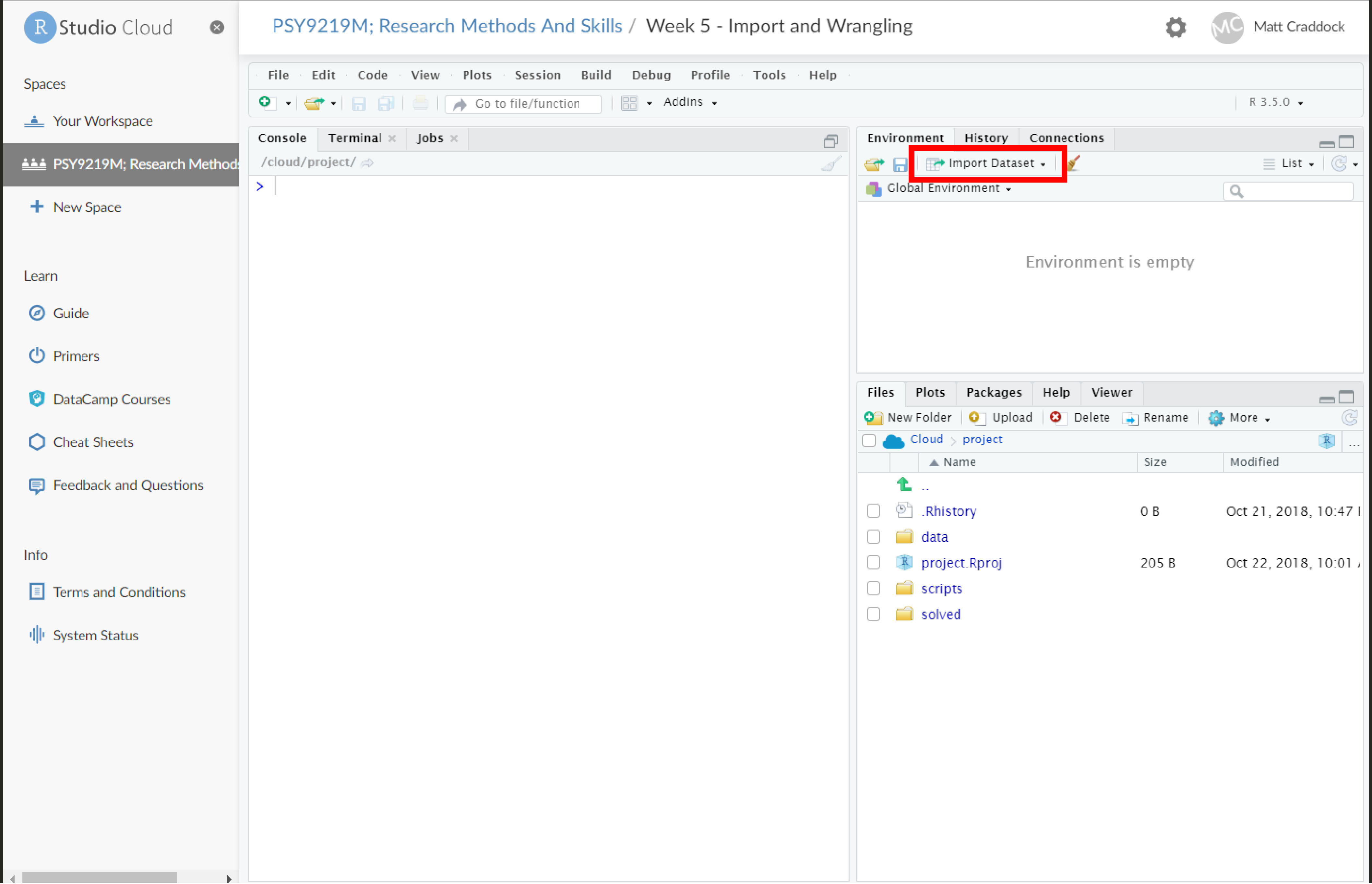
Excel spreadsheets



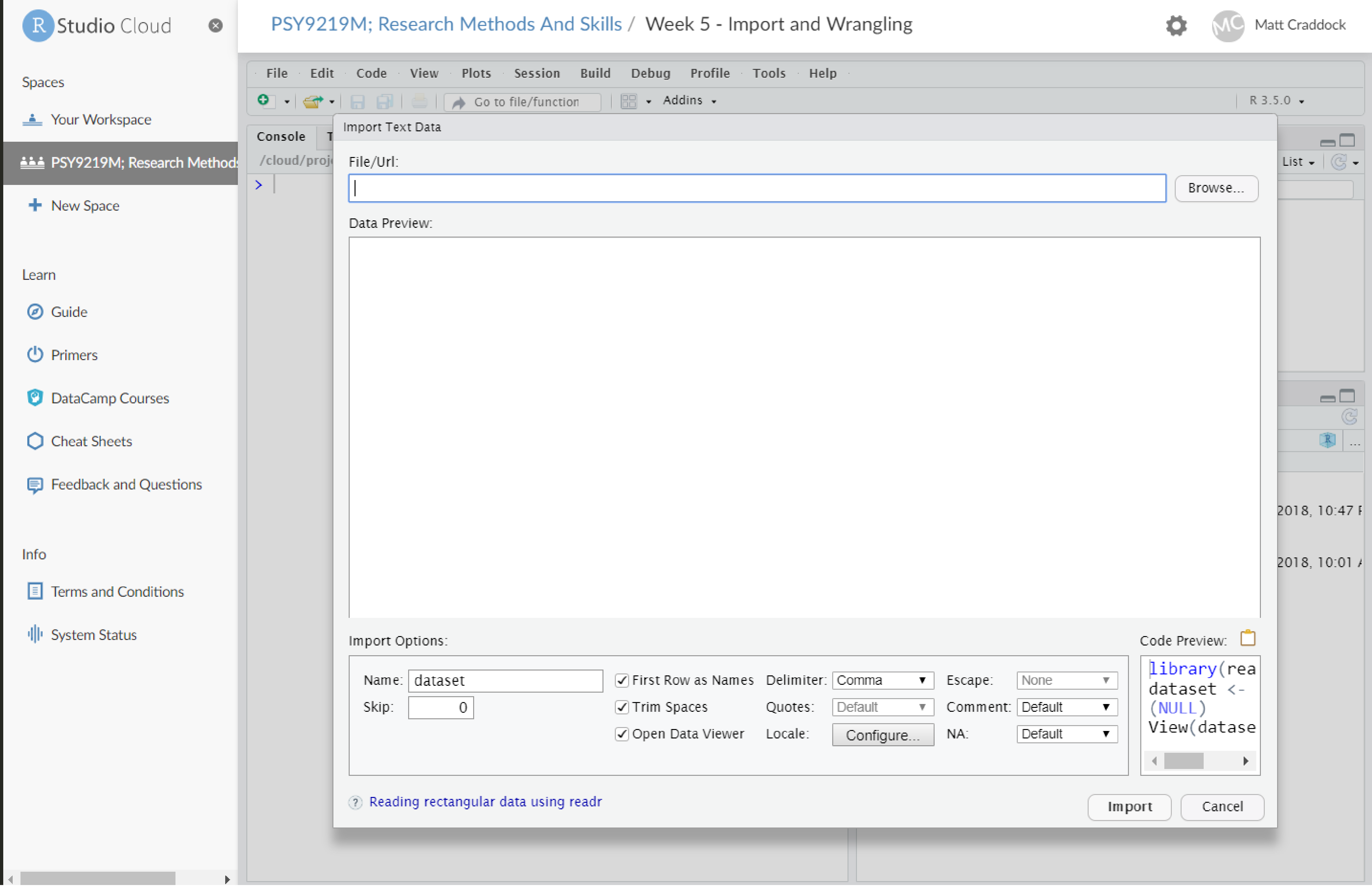
prison-population-data-tool-31-december-2017 - Saved

A18 | a Establishment*Sex*Age Group

	A	B	C	D	E	F
1	View	Date	Establishment	Sex	Age / Custody / Nationality / Offence Group	Popul
2	a Establishment*Sex*Age Group	2015-06	Altcourse	Male	Adults (21+)	
3	a Establishment*Sex*Age Group	2015-06	Altcourse	Male	Juveniles and Young Adults (15-20)	
4	a Establishment*Sex*Age Group	2015-06	Ashfield	Male	Adults (21+)	
5	a Establishment*Sex*Age Group	2015-06	Askham Grange	Female	Adults (21+)	
6	a Establishment*Sex*Age Group	2015-06	Askham Grange	Female	Juveniles and Young Adults (15-20)	
7	a Establishment*Sex*Age Group	2015-06	Aylesbury	Male	Adults (21+)	
8	a Establishment*Sex*Age Group	2015-06	Aylesbury	Male	Juveniles and Young Adults (15-20)	
9	a Establishment*Sex*Age Group	2015-06	Bedford	Male	Adults (21+)	
10	a Establishment*Sex*Age Group	2015-06	Bedford	Male	Juveniles and Young Adults (15-20)	
11	a Establishment*Sex*Age Group	2015-06	Belmarsh	Male	Adults (21+)	
12	a Establishment*Sex*Age Group	2015-06	Belmarsh	Male	Juveniles and Young Adults (15-20)	
13	a Establishment*Sex*Age Group	2015-06	Birmingham	Male	Adults (21+)	1
14	a Establishment*Sex*Age Group	2015-06	Brinsford	Male	Adults (21+)	
15	a Establishment*Sex*Age Group	2015-06	Brinsford	Male	Juveniles and Young Adults (15-20)	
16	a Establishment*Sex*Age Group	2015-06	Bristol	Male	Adults (21+)	
17	a Establishment*Sex*Age Group	2015-06	Bristol	Male	Juveniles and Young Adults (15-20)	
18	a Establishment*Sex*Age Group	2015-06	Brixton	Male	Adults (21+)	
19	a Establishment*Sex*Age Group	2015-06	Bronzefield	Female	Adults (21+)	
20	a Establishment*Sex*Age Group	2015-06	Bronzefield	Female	Juveniles and Young Adults (15-20)	
21	a Establishment*Sex*Age Group	2015-06	Buckley Hall	Male	Adults (21+)	
22	a Establishment*Sex*Age Group	2015-06	Bullingdon	Male	Adults (21+)	1
23	a Establishment*Sex*Age Group	2015-06	Bullingdon	Male	Juveniles and Young Adults (15-20)	
24	a Establishment*Sex*Age Group	2015-06	Bure	Male	Adults (21+)	
25	a Establishment*Sex*Age Group	2015-06	Cardiff	Male	Adults (21+)	
26	a Establishment*Sex*Age Group	2015-06	Cardiff	Male	Juveniles and Young Adults (15-20)	
27	a Establishment*Sex*Age Group	2015-06	Channings Wood	Male	Adults (21+)	



Name	Size	Modified
..		
.Rhistory	0 B	Oct 21, 2018, 10:47
data		
project.Rproj	205 B	Oct 22, 2018, 10:01
scripts		
solved		



Name: dataset

Skip: 0

☑ First Row as Names

☑ Trim Spaces

☑ Open Data Viewer

Delimiter: Comma

Quotes: Default

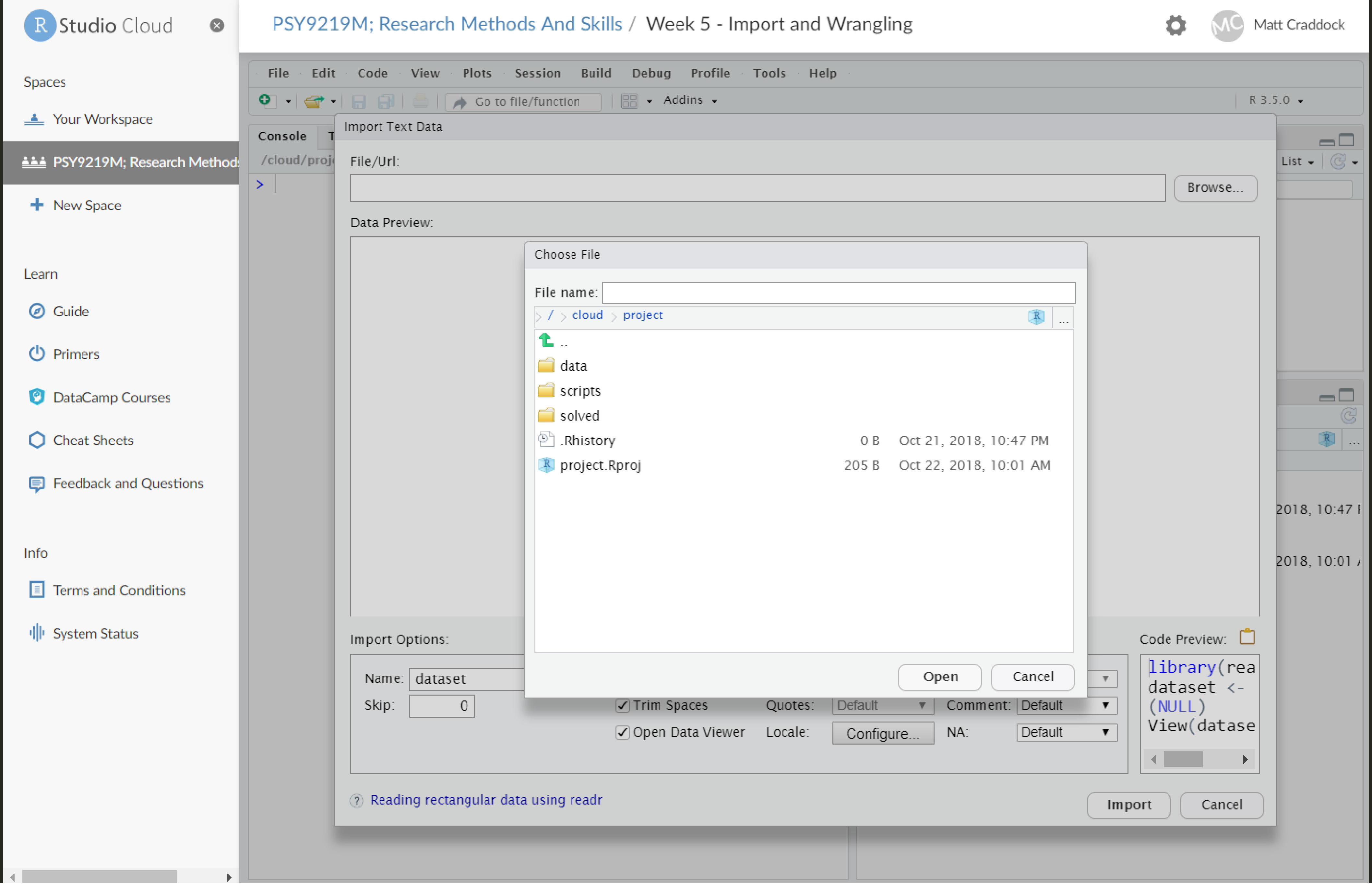
Locale: Configure...

Escape: None

Comment: Default

NA: Default

```
library(rea
dataset <-
(NULL)
View(datase
```



..		
data		
scripts		
solved		
.Rhistory	0 B	Oct 21, 2018, 10:47 PM
project.Rproj	205 B	Oct 22, 2018, 10:01 AM

Name: dataset

Skip: 0

☑ Trim Spaces

☑ Open Data Viewer

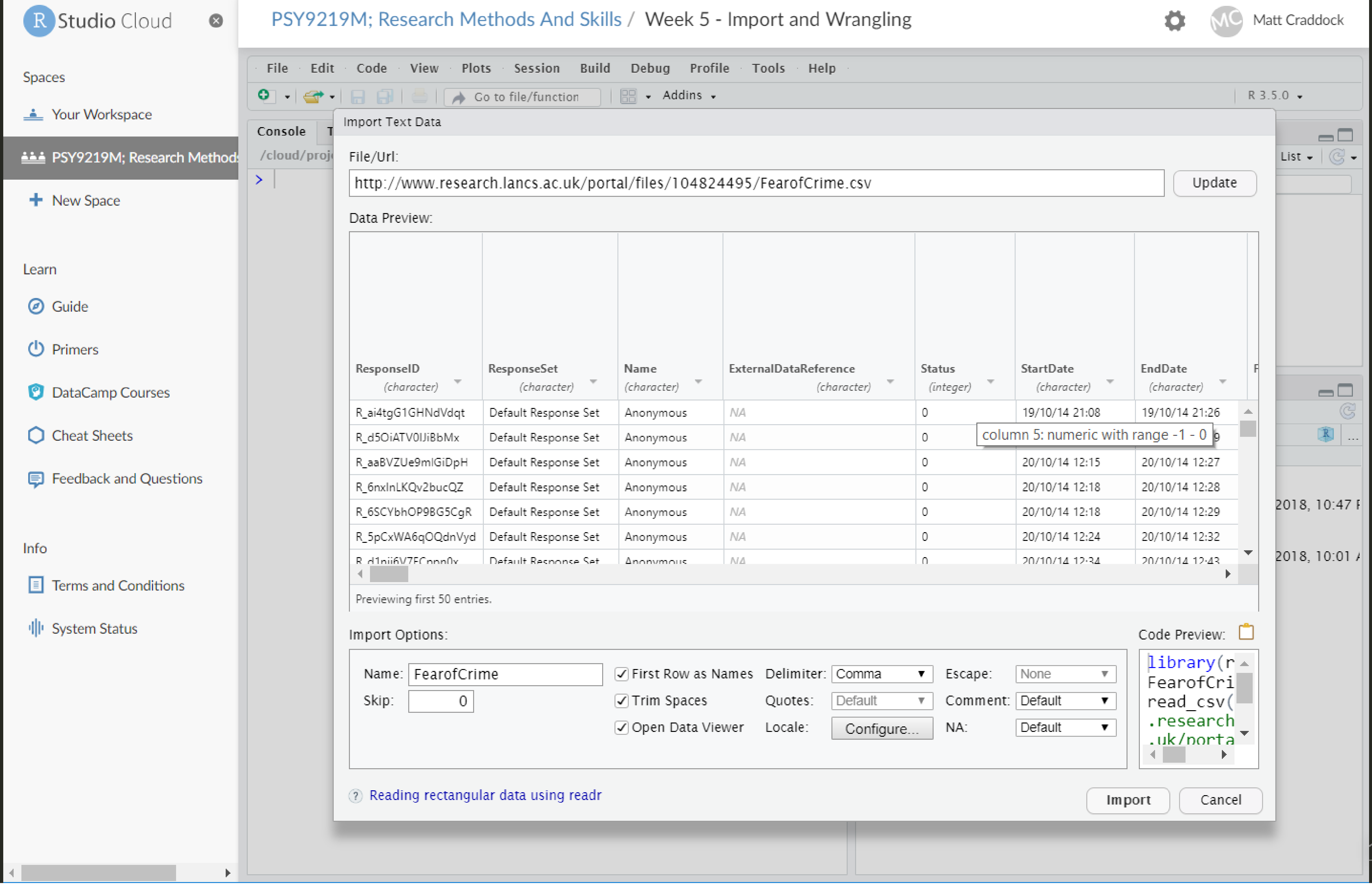
Quotes: Default

Locale: Configure...

Comment: Default

NA: Default

```
library(rea
dataset <-
(NULL)
View(datase
```



Import Text Data

File/Url: http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv

Data Preview:

ResponseID (character)	ResponseSet (character)	Name (character)	ExternalDataReference (character)	Status (integer)	StartDate (character)	EndDate (character)
R_ai4tgG1GHNdVdqt	Default Response Set	Anonymous	NA	0	19/10/14 21:08	19/10/14 21:26
R_d5OiATV0lJiBbMx	Default Response Set	Anonymous	NA	0		9
R_aaBVZUe9mlGiDpH	Default Response Set	Anonymous	NA	0	20/10/14 12:15	20/10/14 12:27
R_6nxInLKQv2bucQZ	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:28
R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:29
R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0	20/10/14 12:24	20/10/14 12:32
R_d1nji6V7FCnpn0x	Default Response Set	Anonymous	NA	0	20/10/14 12:34	20/10/14 12:43

column 5: numeric with range -1 - 0

Previewing first 50 entries.

Import Options:

Name: FearofCrime

Skip: 0

☑ First Row as Names

☑ Trim Spaces

☑ Open Data Viewer

Delimiter: Comma

Quotes: Default

Locale: Configure...

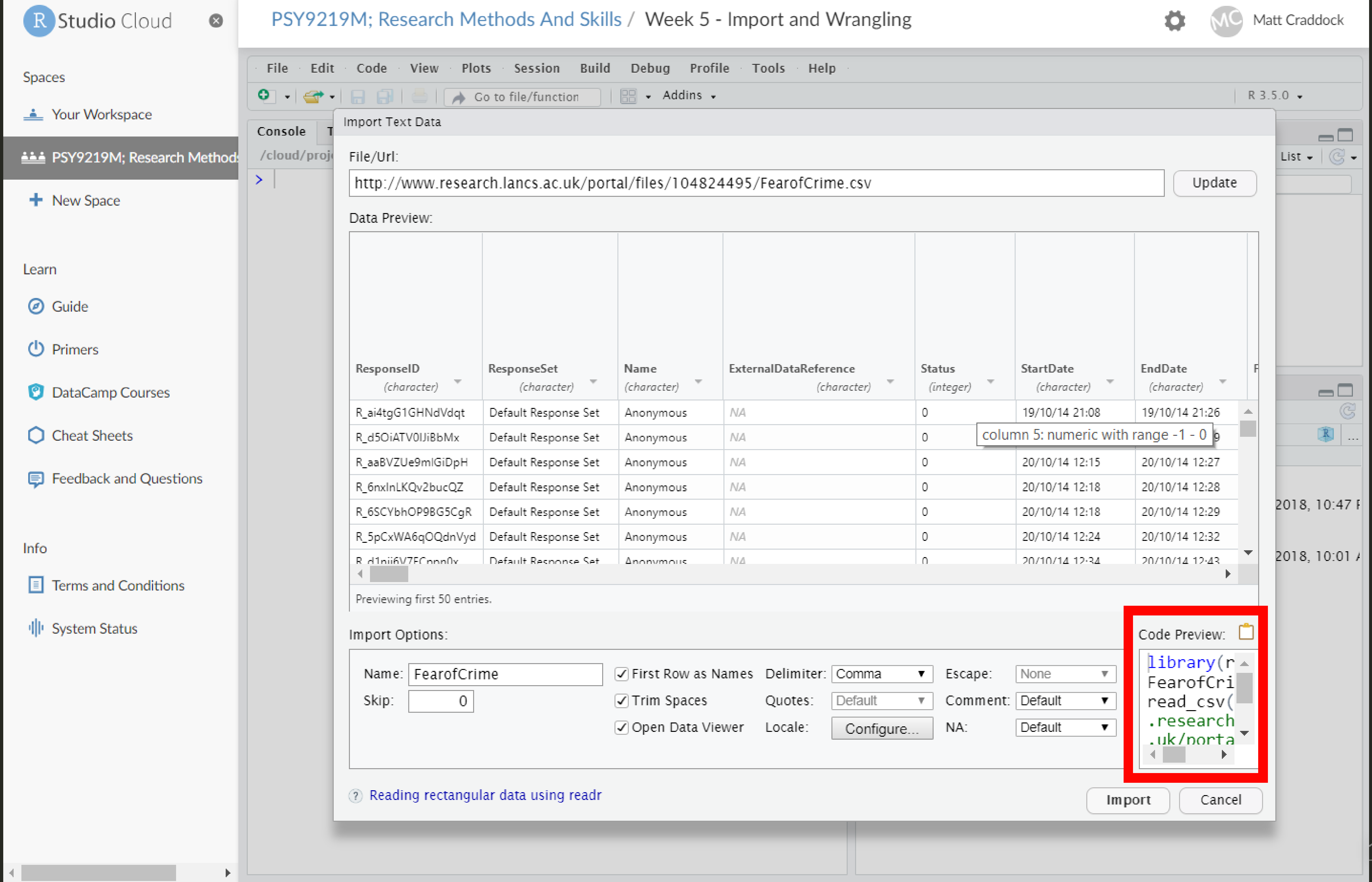
Escape: None

Comment: Default

NA: Default

Code Preview:

```
library(r
FearofCri
read_csv(
.research
.uk/porta
```



Import Text Data

File/Url: http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv

Update

Data Preview:

ResponseID (character)	ResponseSet (character)	Name (character)	ExternalDataReference (character)	Status (integer)	StartDate (character)	EndDate (character)
R_ai4tgG1GHNdVdqt	Default Response Set	Anonymous	NA	0	19/10/14 21:08	19/10/14 21:26
R_d5OiATV0IJiBbMx	Default Response Set	Anonymous	NA	0		9
R_aaBVZUe9mlGiDpH	Default Response Set	Anonymous	NA	0	20/10/14 12:15	20/10/14 12:27
R_6nxInLKQv2bucQZ	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:28
R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:29
R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0	20/10/14 12:24	20/10/14 12:32
R_d1nji6V7FCnnn0x	Default Response Set	Anonymous	NA	0	20/10/14 12:34	20/10/14 12:43

column 5: numeric with range -1 - 0

Previewing first 50 entries.

Import Options:

Name: FearofCrime

Skip: 0

☑ First Row as Names

☑ Trim Spaces

☑ Open Data Viewer

Delimiter: Comma

Quotes: Default

Locale: Configure...

Escape: None

Comment: Default

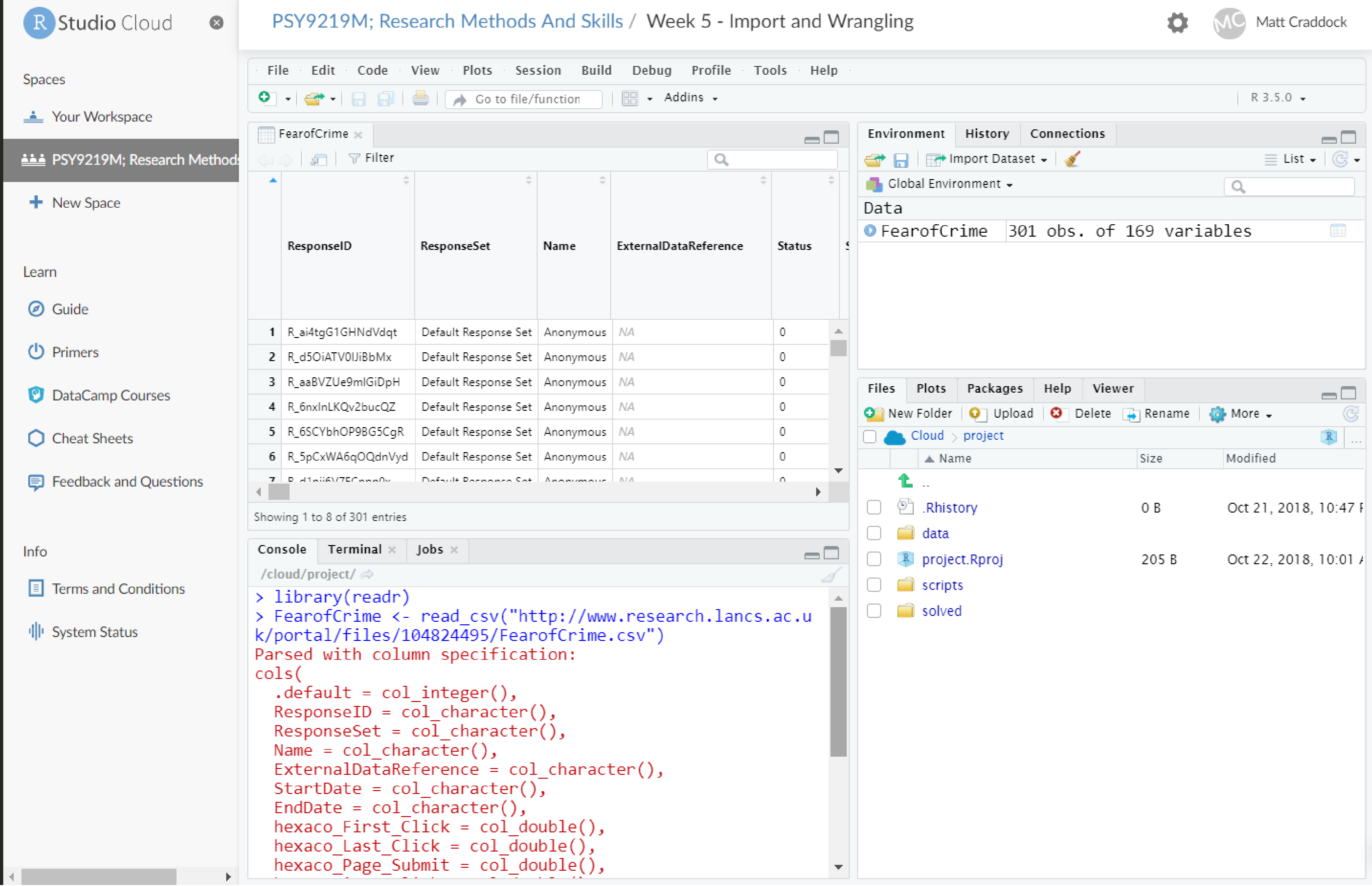
NA: Default

Code Preview:

```
library(r
FearofCri
read_csv(
.research
.uk/porta
```

? Reading rectangular data using readr

Import Cancel



	ResponseID	ResponseSet	Name	ExternalDataReference	Status
1	R_ai4tgG1GHNdVdqt	Default Response Set	Anonymous	NA	0
2	R_d5OiATV0IJiBbMx	Default Response Set	Anonymous	NA	0
3	R_aaBVZUe9mlGiDpH	Default Response Set	Anonymous	NA	0
4	R_6nxInLKQv2bucQZ	Default Response Set	Anonymous	NA	0
5	R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA	0
6	R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0

Showing 1 to 8 of 301 entries

Data

FearofCrime 301 obs. of 169 variables

Name	Size	Modified
..		
.Rhistory	0 B	Oct 21, 2018, 10:47
data		
project.Rproj	205 B	Oct 22, 2018, 10:01
scripts		
solved		

/cloud/project/

```
> library(readr)
> FearofCrime <- read_csv("http://www.research.lancs.ac.u
k/portal/files/104824495/FearofCrime.csv")
Parsed with column specification:
cols(
  .default = col_integer(),
  ResponseID = col_character(),
  ResponseSet = col_character(),
  Name = col_character(),
  ExternalDataReference = col_character(),
  StartDate = col_character(),
  EndDate = col_character(),
  hexaco_First_Click = col_double(),
  hexaco_Last_Click = col_double(),
  hexaco_Page_Submit = col_double(),
```

Prison population

Last week, we looked at some data regarding the UK's prison population.

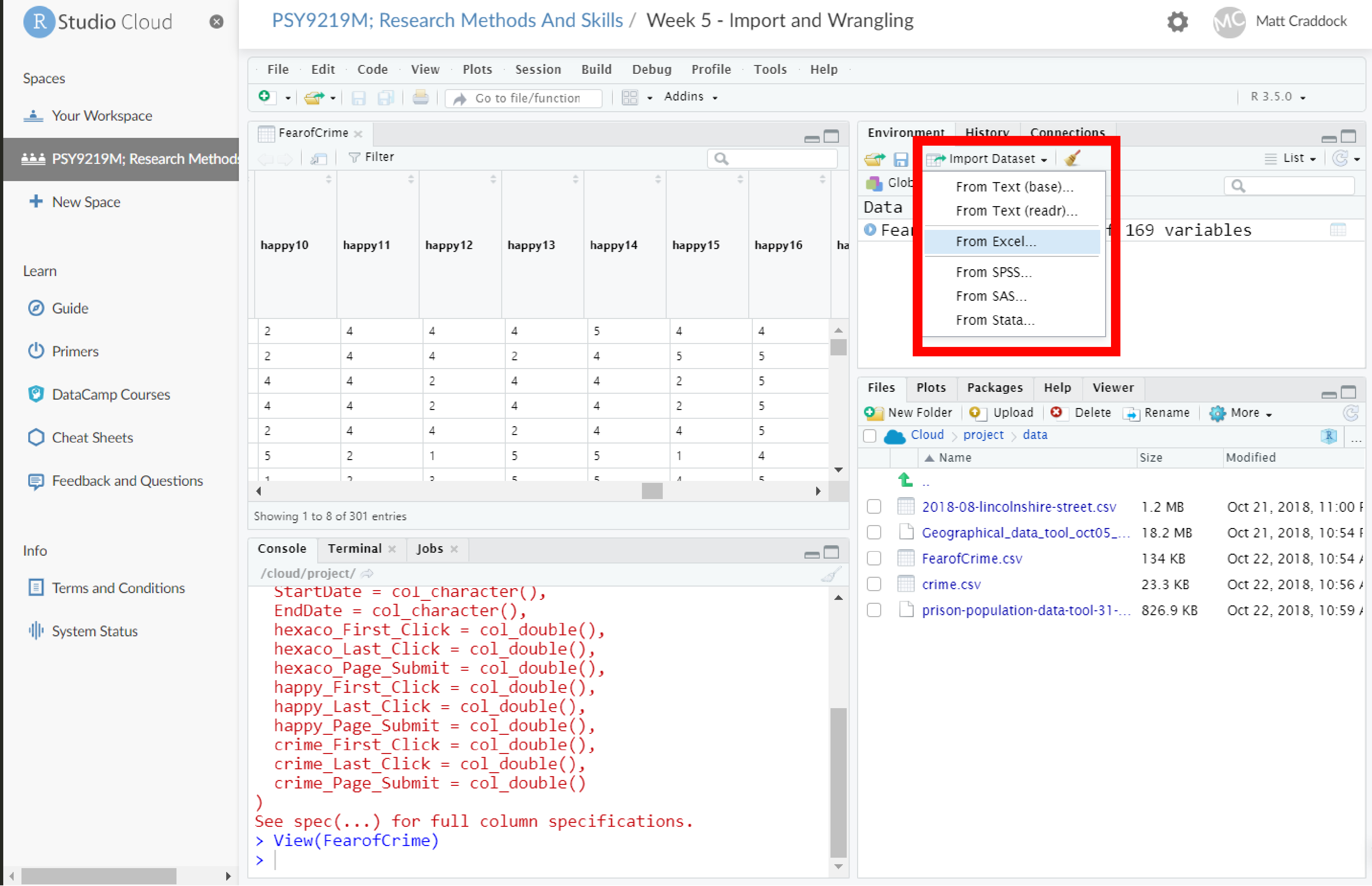
The data is contained in an Excel spreadsheet, downloaded from data.gov.uk.

```
library(readxl)
prison_pop <- read_excel("data/prison-population-data-tool-31-december-2017.xlsx",
                         sheet = "PT Data")
```

We use the **read_excel()** function to read Excel files.

Note how the file name and location come first, and then I specify a specific *sheet*.

Excel spreadsheets often have multiple sheets with different information.

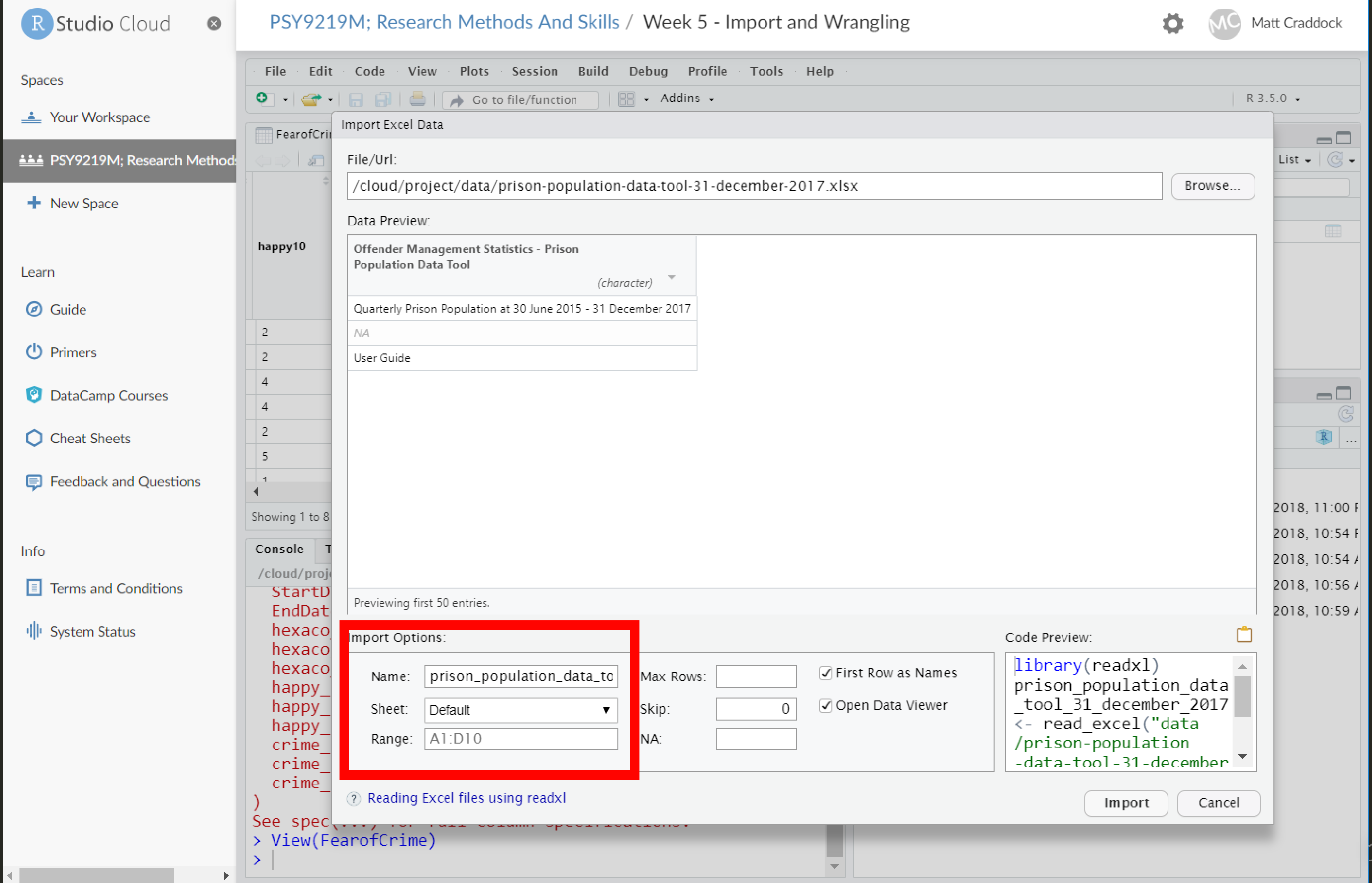


happy10	happy11	happy12	happy13	happy14	happy15	happy16	ha
2	4	4	4	5	4	4	
2	4	4	2	4	5	5	
4	4	2	4	4	2	5	
4	4	2	4	4	2	5	
2	4	4	2	4	4	5	
5	2	1	5	5	1	4	
1	2	3	5	5	4	5	

Showing 1 to 8 of 301 entries

Name	Size	Modified
..		
2018-08-lincolnshire-street.csv	1.2 MB	Oct 21, 2018, 11:00
Geographical_data_tool_oct05_...	18.2 MB	Oct 21, 2018, 10:54
FearofCrime.csv	134 KB	Oct 22, 2018, 10:54
crime.csv	23.3 KB	Oct 22, 2018, 10:56
prison-population-data-tool-31-...	826.9 KB	Oct 22, 2018, 10:59

```
  StartDate = col_character(),
  EndDate = col_character(),
  hexaco_First_Click = col_double(),
  hexaco_Last_Click = col_double(),
  hexaco_Page_Submit = col_double(),
  happy_First_Click = col_double(),
  happy_Last_Click = col_double(),
  happy_Page_Submit = col_double(),
  crime_First_Click = col_double(),
  crime_Last_Click = col_double(),
  crime_Page_Submit = col_double()
)
See spec(...) for full column specifications.
> View(FearofCrime)
> 
```



File/Url: /cloud/project/data/prison-population-data-tool-31-december-2017.xlsx

Data Preview:

Offender Management Statistics - Prison Population Data Tool (character)
Quarterly Prison Population at 30 June 2015 - 31 December 2017
NA
User Guide

Previewing first 50 entries.

Import Options:

Name: prison_population_data_to

Sheet: Default

Range: A1:D10

Max Rows:

Skip: 0

NA:

☑ First Row as Names

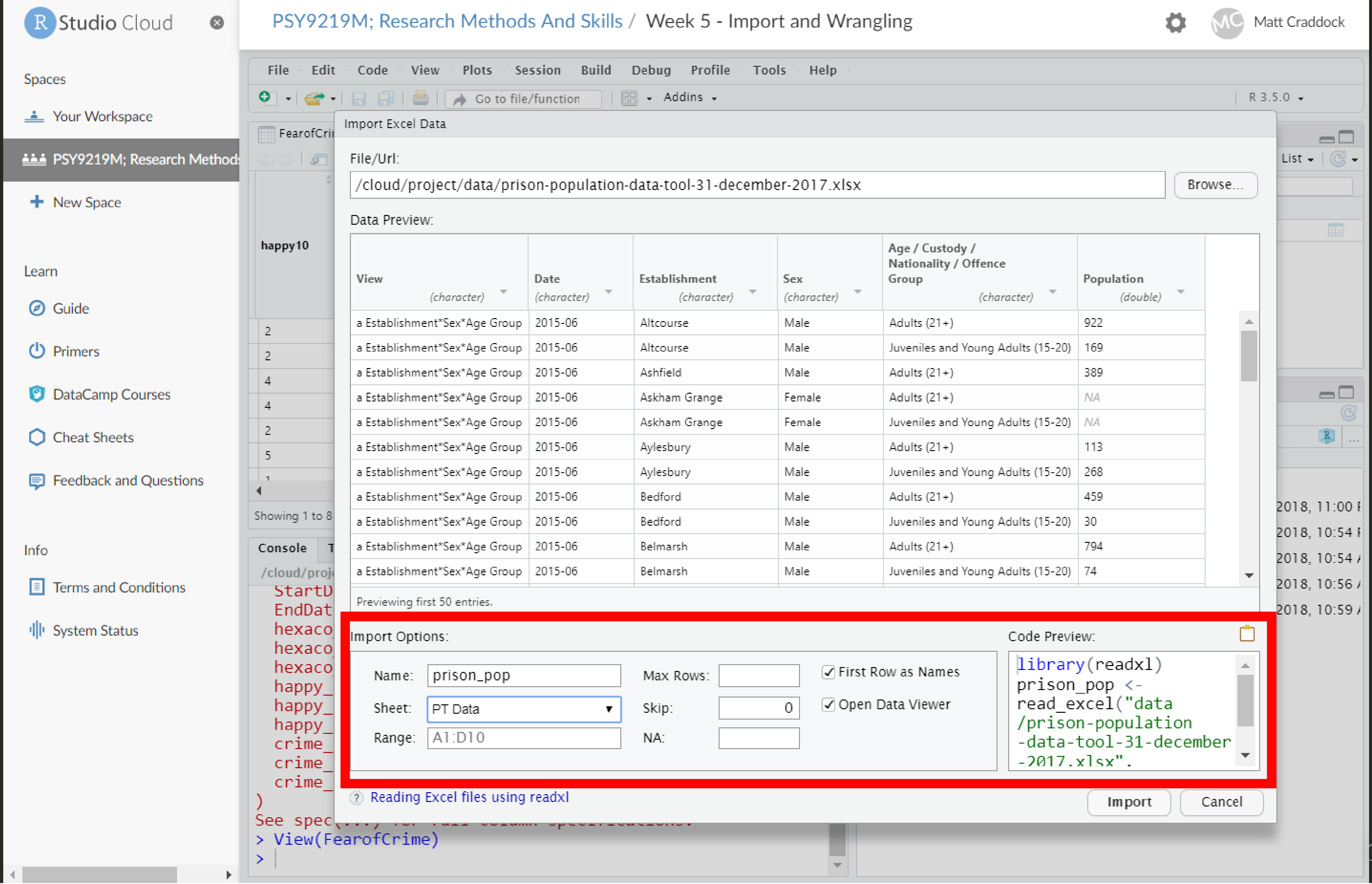
☑ Open Data Viewer

Code Preview:

```
library(readxl)
prison_population_data
_tool_31_december_2017
<- read_excel("data
/prison-population
-data-tool-31-december
```

Reading Excel files using readxl

```
/cloud/proj
  StartD
  EndDat
  hexaco
  hexaco
  hexaco
  happy_
  happy_
  happy_
  crime_
  crime_
  crime_
)
See spec(...) for full column specifications.
> View(FearofCrime)
>
```



Import Excel Data

File/Url: /cloud/project/data/prison-population-data-tool-31-december-2017.xlsx

Data Preview:

View (character)	Date (character)	Establishment (character)	Sex (character)	Age / Custody / Nationality / Offence Group (character)	Population (double)
a Establishment*Sex*Age Group	2015-06	Altcourse	Male	Adults (21+)	922
a Establishment*Sex*Age Group	2015-06	Altcourse	Male	Juveniles and Young Adults (15-20)	169
a Establishment*Sex*Age Group	2015-06	Ashfield	Male	Adults (21+)	389
a Establishment*Sex*Age Group	2015-06	Askham Grange	Female	Adults (21+)	NA
a Establishment*Sex*Age Group	2015-06	Askham Grange	Female	Juveniles and Young Adults (15-20)	NA
a Establishment*Sex*Age Group	2015-06	Aylesbury	Male	Adults (21+)	113
a Establishment*Sex*Age Group	2015-06	Aylesbury	Male	Juveniles and Young Adults (15-20)	268
a Establishment*Sex*Age Group	2015-06	Bedford	Male	Adults (21+)	459
a Establishment*Sex*Age Group	2015-06	Bedford	Male	Juveniles and Young Adults (15-20)	30
a Establishment*Sex*Age Group	2015-06	Belmarsh	Male	Adults (21+)	794
a Establishment*Sex*Age Group	2015-06	Belmarsh	Male	Juveniles and Young Adults (15-20)	74

Previewing first 50 entries.

Import Options:

Name: prison_pop
Sheet: PT Data
Range: A1:D10
Max Rows:
Skip: 0
NA:
☑ First Row as Names
☑ Open Data Viewer

Code Preview:

```
library(readxl)
prison_pop <-
read_excel("data
/prison-population
-data-tool-31-december
-2017.xlsx",
```

Reading Excel files using readxl

```
StartD
EndDat
hexaco
hexaco
hexaco
happy_
happy_
happy_
crime_
crime_
crime_
)
See spec(...) for full column specifications.
> View(FearofCrime)
>
```

Prison population

Once the data is imported, we have a *tibble*.

We can immediately see there are 6 columns with 22409 rows.

```
prison_pop
```

```
## # A tibble: 22,409 x 6
##    View        Date   Establishment Sex   `Age / Custody / Nati~ Population
##    <chr>       <chr>  <chr>         <chr> <chr>                       <dbl>
##  1 a Establis~ 2015-~ Altcourse     Male  Adults (21+)                  922
##  2 a Establis~ 2015-~ Altcourse     Male  Juveniles and Young A~        169
##  3 a Establis~ 2015-~ Ashfield      Male  Adults (21+)                  389
##  4 a Establis~ 2015-~ Askham Grange Fema~ Adults (21+)                   NA
##  5 a Establis~ 2015-~ Askham Grange Fema~ Juveniles and Young A~         NA
##  6 a Establis~ 2015-~ Aylesbury     Male  Adults (21+)                  113
##  7 a Establis~ 2015-~ Aylesbury     Male  Juveniles and Young A~        268
##  8 a Establis~ 2015-~ Bedford       Male  Adults (21+)                  459
##  9 a Establis~ 2015-~ Bedford       Male  Juveniles and Young A~         30
## 10 a Establis~ 2015-~ Belmarsh      Male  Adults (21+)                  794
## # ... with 22,399 more rows
```

We need to do more work to make this file useable…

Fear of Crime Dataset

Ellis & Renouf (2018) - the relationship between fear of crime and various personality measures.

Their data is openly available, stored as text in a *comma-separated-values* format (*.csv*).

Once again, we can use the import button or some code (with **read_csv()**)to load this data in and automatically format it into a *tibble*.

```r
library(readr)
FearofCrime <- read_csv("data/FearofCrime.CSV")
```

See also Ellis & Merdian, 2015, Frontiers in Psychology

Fear of Crime Dataset

Ellis & Renouf (2018) collected data online using Qualtrics.

The file contains one column for each question that the participants answered, for a total of 169(!) columns.

Each row is a single participant's answers, and their demographic information.

```
FearofCrime
```

```
## # A tibble: 301 x 169
##    ResponseID ResponseSet Name  ExternalDataRef~ Status StartDate EndDate
##    <chr>      <chr>       <chr> <lgl>             <dbl> <chr>     <chr>
##  1 R_ai4tgG1~ Default Re~ Anon~ NA                    0 19/10/14~ 19/10/~
##  2 R_d5OiATV~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
##  3 R_aaBVZUe~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
##  4 R_6nxInLK~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
##  5 R_6SCYbhO~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
##  6 R_5pCxWA6~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
##  7 R_d1nji6V~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
##  8 R_9v6ZgUh~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
##  9 R_5Bg7VjB~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
## 10 R_9Sv17lQ~ Default Re~ Anon~ NA                    0 20/10/14~ 20/10/~
## # ... with 291 more rows, and 162 more variables: Finished <dbl>, `Consent
```

dpylr and data transformation

Data transformation

With datasets like those we've loaded, there are often organisational issues.

For example, there could be many columns or rows we don't need, or the data would make more sense if it were sorted.

This is where **dplyr** comes in!

Function	Effect
filter()	Include or exclude observations (rows)
select()	Include or exclude variables (columns)
mutate()	Create new variables (columns)
summarise()	Aggregate or summarise groups of observations (rows)
arrange()	Change the order of observations (rows)

Removing unwanted rows

Filtering rows

The `prison_pop` dataset has 22409 rows, but we don't need (or want) them all!

```
unique(prison_pop$View)
```

```
## [1] "a Establishment*Sex*Age Group"     "b Establishment*Sex*Custody type"
## [3] "c Establishment*Sex*Nationality"   "d Establishment*Sex*Offence group"
```

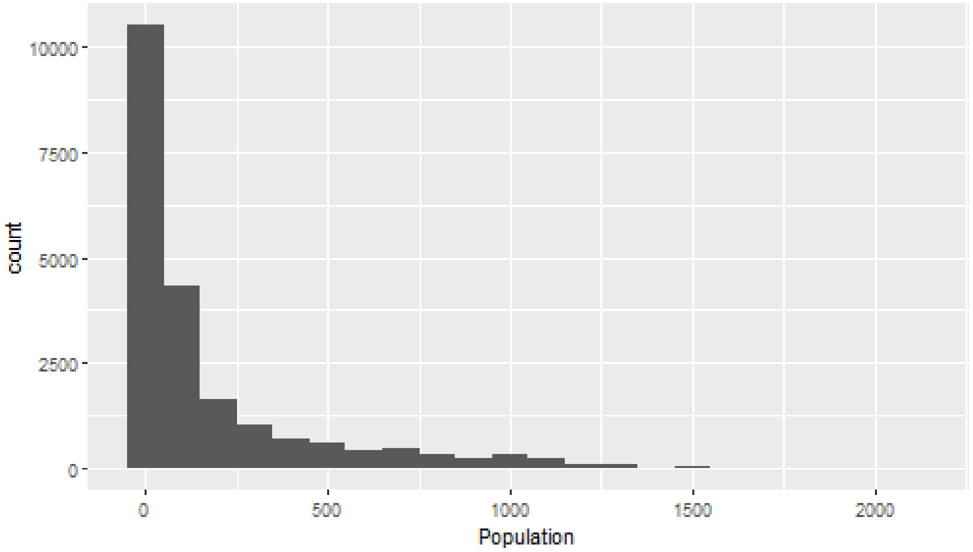
The data is actually *repeated* four times, but organised differently each time.

```
## # A tibble: 4 x 3
##   View                              total_pop num_entries
##   <chr>                                 <dbl>       <int>
## 1 a Establishment*Sex*Age Group        938760        2042
## 2 b Establishment*Sex*Custody type     939314        2740
## 3 c Establishment*Sex*Nationality      938841        3215
## 4 d Establishment*Sex*Offence group    936191       14412
```

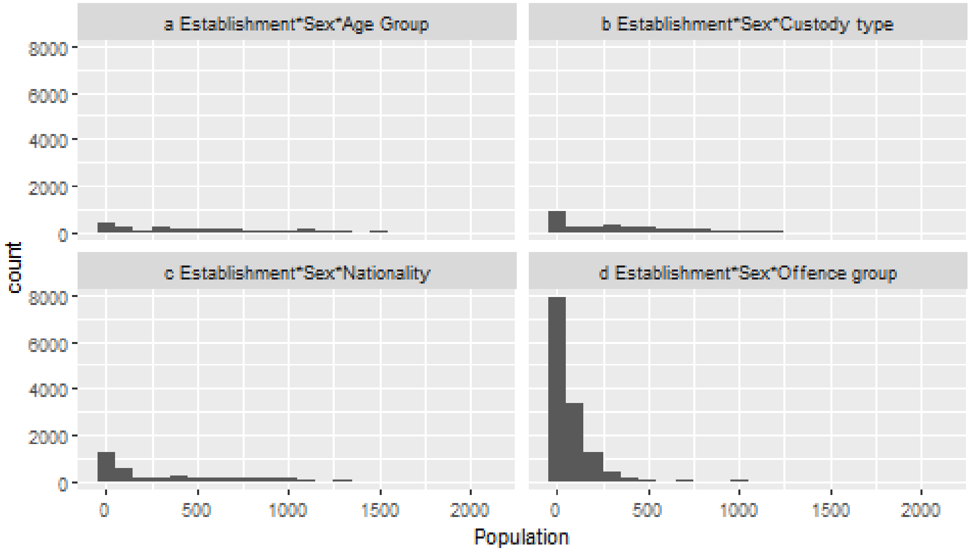
Filtering rows

If we just started investigating the data without accounting for this, it would be misleading.

```
ggplot(prison_pop, aes(x = Population)) +
  geom_histogram(binwidth = 100)
```



```
ggplot(prison_pop, aes(x = Population)) +
  geom_histogram(binwidth = 100) + facet_wrap
```



Filtering rows

We can use the **filter()** function to select only the rows we're interested in, using *logical conditions* and *relational operators*.

```
filter(prison_pop,
       View == "a Establishment*Sex*Age Group")
```

```
## # A tibble: 2,042 x 6
##    View         Date   Establishment Sex   `Age / Custody / Nati~ Population
##    <chr>        <chr>  <chr>         <chr> <chr>                       <dbl>
##  1 a Establis~ 2015-~ Altcourse     Male  Adults (21+)                  922
##  2 a Establis~ 2015-~ Altcourse     Male  Juveniles and Young A~        169
##  3 a Establis~ 2015-~ Ashfield      Male  Adults (21+)                  389
##  4 a Establis~ 2015-~ Askham Grange Fema~ Adults (21+)                   NA
##  5 a Establis~ 2015-~ Askham Grange Fema~ Juveniles and Young A~         NA
##  6 a Establis~ 2015-~ Aylesbury     Male  Adults (21+)                  113
##  7 a Establis~ 2015-~ Aylesbury     Male  Juveniles and Young A~        268
##  8 a Establis~ 2015-~ Bedford       Male  Adults (21+)                  459
##  9 a Establis~ 2015-~ Bedford       Male  Juveniles and Young A~         30
## 10 a Establis~ 2015-~ Belmarsh      Male  Adults (21+)                  794
## # ... with 2,032 more rows
```

Relational operators

Relational operators compare two (or more) things and return a **logical** value (i.e. TRUE/FALSE)

Operator	Meaning	Example
>	Greater than	5 > 4
>=	Greater than or equal to	4 >= 4
<	Less than	Population < 400
<=	Less than or equal to	Population <= 400
==	Exactly equal to	Sex == "Male"
!=	Not equal to	Establishment != "Ashfield"
%in%	Is contained in	Establishment %in% c("Bedford", "Oakwood")

Logical operators

Logical operators can be used to combine multiple relational operators or *negate* a relational operator.

Operator	Meaning	Example
&	AND	Population < 1000 & Sex == "Male"
|	OR	Population > 200 & Population < 500
!	NOT	!(Establishment %in% c("Bedford", "Oakwood"))

Filtering rows

We can have multiple *conditions* for selection with **filter()**.

Suppose we only wanted to include rows where Population is over 300 but under 600.

```
filter(prison_pop,
       View == "a Establishment*Sex*Age Group",
       Population > 300 & Population < 600)
```

```
## # A tibble: 487 x 6
##    View         Date   Establishment Sex   `Age / Custody / Nati~ Population
##    <chr>        <chr>  <chr>         <chr> <chr>                       <dbl>
##  1 a Establis~ 2015-~ Ashfield      Male  Adults (21+)                  389
##  2 a Establis~ 2015-~ Bedford       Male  Adults (21+)                  459
##  3 a Establis~ 2015-~ Brinsford     Male  Juveniles and Young A~        349
##  4 a Establis~ 2015-~ Bristol       Male  Adults (21+)                  553
##  5 a Establis~ 2015-~ Bronzefield   Fema~ Adults (21+)                  459
##  6 a Establis~ 2015-~ Buckley Hall  Male  Adults (21+)                  440
##  7 a Establis~ 2015-~ Coldingley    Male  Adults (21+)                  515
##  8 a Establis~ 2015-~ Deerbolt      Male  Juveniles and Young A~        311
##  9 a Establis~ 2015-~ Eastwood Park Fema~ Adults (21+)                  331
## 10 a Establis~ 2015-~ Erlestoke     Male  Adults (21+)                  514
## # ... with 477 more rows
```

Removing unneeded columns

Selecting columns

Sometimes only some columns are of interest.

The Fear of Crime dataset has 169 columns. Only some of them are useful.

```r
names(FearofCrime)[1:10]
```

```
##  [1] "ResponseID"
##  [2] "ResponseSet"
##  [3] "Name"
##  [4] "ExternalDataReference"
##  [5] "Status"
##  [6] "StartDate"
##  [7] "EndDate"
##  [8] "Finished"
##  [9] "Consent Form / This study includes a range of questionnaires collecting / demographic and indivi
## [10] "sex"
```

Selecting columns

Suppose that, first of all, we were only interested in the age and sex of our participants.

```
select(FearofCrime, age, sex)
```

```
## # A tibble: 301 x 2
##       age   sex
##     <dbl> <dbl>
##  1     26     2
##  2     66     2
##  3     41     1
##  4     46     1
##  5     53     2
##  6     33     1
##  7     41     2
##  8     39     1
##  9     38     2
## 10     19     2
## # ... with 291 more rows
```

Selecting columns

The HEXACO-PI-R is a personality questionnaire that aims to measure six factors - Honesty-Humility, Emotionality, Extraversion, Agreeableness, Conscientiousness, and Openness to Experience.

The Fear of Crime dataset has the participants answers to the 60 questions of the HEXACO-PI-R in 60 columns.

```
select(FearofCrime, hexaco1, hexaco2, hexaco3)
```

```
## # A tibble: 8 x 3
##   hexaco1 hexaco2 hexaco3
##     <dbl>   <dbl>   <dbl>
## 1       4       5       2
## 2       2       4       2
## 3       1       5       2
## 4       1       5       2
## 5       2       4       4
## 6       2       4       2
## 7       1       5       4
## 8       2       4       3
```

Selecting columns

Typing these out one by one would be ... *laborious*.

Fortunately, there are some shorthands.

The colon (:) operator can be used to say "everything between these columns (inclusive)".

```
select(FearofCrime, hexaco1:hexaco5)
```

```
## # A tibble: 301 x 5
##    hexaco1 hexaco2 hexaco3 hexaco4 hexaco5
##      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1       4       5       2       4       1
##  2       2       4       2       4       4
##  3       1       5       2       3       2
##  4       1       5       2       4       1
##  5       2       4       4       5       5
##  6       2       4       2       2       2
##  7       1       5       4       4       4
##  8       2       4       3       2       2
##  9       1       2       4       2       5
## 10       4       4       2       3       2
## # ... with 291 more rows
```

Selecting columns

There are also several helper functions that can be used within **select()** (see the cheat sheet!).

starts_with() will select any column that starts with the string you supply:

```
hex_only <- select(FearofCrime, starts_with("hexaco1"))
head(hex_only, 5)
```

```
## # A tibble: 5 x 11
##   hexaco1 hexaco10 hexaco11 hexaco12 hexaco13 hexaco14 hexaco15 hexaco16
##     <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1       4        2        4        4        4        2        5        5
## 2       2        2        3        4        5        1        4        3
## 3       1        4        2        5        4        1        3        1
## 4       1        1        1        4        5        2        5        1
## 5       2        1        2        2        2        2        2        5
## # ... with 3 more variables: hexaco17 <dbl>, hexaco18 <dbl>,
## #   hexaco19 <dbl>
```

Selecting columns

Note that you can also tell **select()** to *remove* columns using the minus (-) sign.

Here I tell it to remove a few columns that have no useful information.

```
FoC_removed <- select(FearofCrime, -ResponseSet, -Name,
                      -Status, -ExternalDataReference)
head(FoC_removed[, 1:7], 5)
```

```
## # A tibble: 5 x 7
##   ResponseID  StartDate  EndDate Finished `Consent Form / This~   sex   age
##   <chr>       <chr>      <chr>      <dbl>                 <dbl> <dbl> <dbl>
## 1 R_ai4tgG1G~ 19/10/14 ~ 19/10/~        1                     1     2    26
## 2 R_d5OiATV0~ 20/10/14 ~ 20/10/~        1                     1     2    66
## 3 R_aaBVZUe9~ 20/10/14 ~ 20/10/~        1                     1     1    41
## 4 R_6nxInLKQ~ 20/10/14 ~ 20/10/~        1                     1     1    46
## 5 R_6SCYbhOP~ 20/10/14 ~ 20/10/~        1                     1     2    53
```

Calculating new columns

Mutating columns

Many psychometric tests calculate scores by adding up the responses across questions.

For example, the State-Trait Anxiety Inventory (STAI) was collected in the Fear of Crime study.

The STAI is split into two parts of 20 questions, one for "state" anxiety (i.e. a person's generaly propensity towards anxiety), one for "trait" anxiety (i.e. how anxious a person is *right now*).

Although there are 20 items, for a demo I select the first 4 "state"" questions.

```r
FoC_stai <- select(FearofCrime, stai1:stai4)
FoC_stai <- mutate(FoC_stai, state_anxiety = stai1 + stai2 + stai3 + stai4)
FoC_stai["state_anxiety"]
```

```
## # A tibble: 301 x 1
##    state_anxiety
##            <dbl>
##  1             7
##  2             9
##  3            12
##  4             9
##  5            10
##  6            10
```

Creating summaries

Summarising rows

summarise() takes data frame columns and summarises them.

The authors of the Fear of Crime study helpfully also provide another version of their dataset that pre-calculates several of the measures.

```
head(crime, 8)
```

```
## # A tibble: 8 x 15
##   Participant sex     age victim_crime     H     E     X     A     C     O
##   <chr>       <chr> <dbl> <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 R_01TjXgC1~ male     55 yes            3.7   3     3.4   3.9   3.2   3.6
## 2 R_0dN5YeUL~ fema~    20 no             2.5   3.1   2.5   2.4   2.2   3.1
## 3 R_0DPiPYWh~ male     57 yes            2.6   3.1   3.3   3.1   4.3   2.8
## 4 R_0f7bSsH6~ male     19 no             3.5   1.8   3.3   3.4   2.1   2.7
## 5 R_0rov2RoS~ fema~    20 no             3.3   3.4   3.9   3.2   2.8   3.9
## 6 R_0wioqGER~ fema~    20 no             2.6   2.6   3     2.6   2.9   3.4
## 7 R_0wRO8lNe~ male     34 yes            3.2   2.5   3.2   2.8   4     3.2
## 8 R_116nEdFs~ fema~    19 no             2.9   4     3.9   4.2   3.7   1.9
## # ... with 5 more variables: SA <dbl>, TA <dbl>, OHQ <dbl>, FoC <dbl>,
## #   Foc2 <dbl>
```

Summarising rows

Here I calculate the mean, standard deviation, and variance of the State Anxiety variable.

This is a simple way to create some basic summary statistics for a given data frame.

Other possible summmary functions (other than **mean()**, **sd()**, or **var()**) include **max()**, **min()**, **IQR()**, or **median()**.

```
summarise(crime,
          mean = mean(SA),
          standard_dev = sd(SA),
          variance = var(SA))
```

```
## # A tibble: 1 x 3
##    mean standard_dev variance
##   <dbl>        <dbl>    <dbl>
## 1  1.92        0.554    0.307
```

Summarising rows

summarise() becomes much more useful when used with the **group_by()** function.

group_by() is used to organise data frames into groups according to categorical variables.

```
grouped_crime <- group_by(crime, sex, victim_crime)
summarise(grouped_crime,
          state_anxiety = mean(SA),
          sd_SA = sd(SA),
          var_SA = var(SA))
```

```
## # A tibble: 4 x 5
## # Groups:   sex [2]
##   sex    victim_crime state_anxiety sd_SA var_SA
##   <chr>  <chr>                <dbl> <dbl>  <dbl>
## 1 female no                    1.90 0.518  0.268
## 2 female yes                   1.98 0.643  0.413
## 3 male   no                    2.02 0.553  0.306
## 4 male   yes                   1.74 0.472  0.223
```

Arranging rows

Sometimes we want to view data sorted by a specific field.

For example, suppose that we were looking at the prison population data and wanted to quickly see which was the largest prison population recorded.

```r
pris_filt <- filter(prison_pop,
                    View == "a Establishment*Sex*Age Group",
                    `Age / Custody / Nationality / Offence Group` == "Adults (21+)",
                    Sex == "Male")
head(arrange(pris_filt, Population), 5)
```

```
## # A tibble: 5 x 6
##   View         Date  Establishment Sex   `Age / Custody / Natio~ Population
##   <chr>        <chr> <chr>         <chr> <chr>                        <dbl>
## 1 a Establish~ 2015~ Feltham       Male  Adults (21+)                     9
## 2 a Establish~ 2016~ Feltham       Male  Adults (21+)                    15
## 3 a Establish~ 2015~ Feltham       Male  Adults (21+)                    20
## 4 a Establish~ 2015~ Brinsford     Male  Adults (21+)                    24
## 5 a Establish~ 2016~ Brinsford     Male  Adults (21+)                    26
```

Arranging rows

Sometimes we want to view data sorted by a specific field.

For example, suppose that we were looking at the prison population data and wanted to quickly see which was the largest prison population recorded.

```
pris_filt <- filter(prison_pop,
                    View == "a Establishment*Sex*Age Group",
                    `Age / Custody / Nationality / Offence Group` == "Adults (21+)",
                    Sex == "Male")
head(arrange(pris_filt, desc(Population)), 5)
```

```
## # A tibble: 5 x 6
##   View         Date  Establishment Sex   `Age / Custody / Natio~ Population
##   <chr>        <chr> <chr>         <chr> <chr>                        <dbl>
## 1 a Establish~ 2017~ Oakwood       Male  Adults (21+)                  2090
## 2 a Establish~ 2017~ Oakwood       Male  Adults (21+)                  2082
## 3 a Establish~ 2017~ Oakwood       Male  Adults (21+)                  2082
## 4 a Establish~ 2017~ Oakwood       Male  Adults (21+)                  2067
## 5 a Establish~ 2016~ Oakwood       Male  Adults (21+)                  1913
```

Putting it all together

Pipes

Often you want to conduct several steps, one after the other.

You could do this using objects to store each intermediate step.

```
temp_pris <- filter(prison_pop,
                    View == "a Establishment*Sex*Age Group",
                    Date == "2015-06")
temp_pris <- group_by(temp_pris,
                      Sex,
                      `Age / Custody / Nationality / Offence Group`)
temp_pris <- summarise(temp_pris,
                       mean_pop = mean(Population, na.rm = TRUE),
                       median_pop = median(Population, na.rm = TRUE),
                       total_pop = sum(Population, na.rm = TRUE),
                       max_pop = max(Population, na.rm = TRUE))
```

Pipes

A simpler way is to use *pipes* **(%>%)**

pipes can be read as meaning "AND THEN"

```
prison_pop %>%
  filter(View == "a Establishment*Sex*Age Group",
         Date == "2015-06") %>%
  group_by(Sex, `Age / Custody / Nationality / Offence Group`) %>%
  summarise(mean_pop = mean(Population, na.rm = TRUE),
            median_pop = median(Population, na.rm = TRUE),
            total_pop = sum(Population, na.rm = TRUE),
            max_pop = max(Population, na.rm = TRUE))
```

```
## # A tibble: 4 x 6
## # Groups:   Sex [2]
##   Sex    `Age / Custody / Nationalit~ mean_pop median_pop total_pop max_pop
##   <chr>  <chr>                           <dbl>      <dbl>     <dbl>   <dbl>
## 1 Female Adults (21+)                    356          333      3560     480
## 2 Female Juveniles and Young Adults ~     18.6         19       167      35
## 3 Male   Adults (21+)                    717.         677     76730    1587
## 4 Male   Juveniles and Young Adults ~    101.          54      5559     490
```

Reading materials

Revision

For revision of this week's concepts, see Chapter 5 - Data transformation of R for Data Science.

For practice, use DataCamp's "Data manipulation in R with dplyr", and the "Work with Data" RStudio cloud primer.

Next week

Discovering Statistics using R (Field et al.)

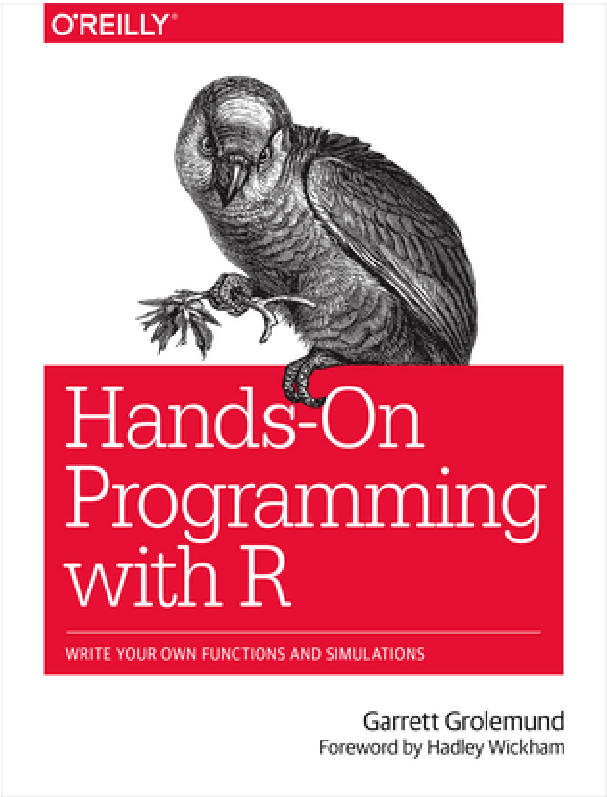
- Chapter 9, Comparing two means
- Chapter 5, Exploring assumptions (additional)

An additional recommendation...

Hands-on Programming with R

Basic R programming book, JUST MADE AVAILABLE ONLINE FOR FREE!

https://rstudio-education.github.io/hopr/index.html


O'REILLY
Hands-On Programming with R
WRITE YOUR OWN FUNCTIONS AND SIMULATIONS
Garrett Grolemund
Foreword by Hadley Wickham

UNIVERSITY OF LINCOLN