These materials adapted by Amelia McNamara from the RStudio CC BY-SA materials Introduction to R (2014) and Master the Tidyverse (2017).

Introduction to R & RStudio:

deck 06: Importing data

Amelia McNamara

Visiting Assistant Professor of Statistical and Data Sciences

Smith College

January 2018

@AmeliaMN

Katie Leap

Kelly O'Briant

@b23kelly @RLadiesDC

Your turn

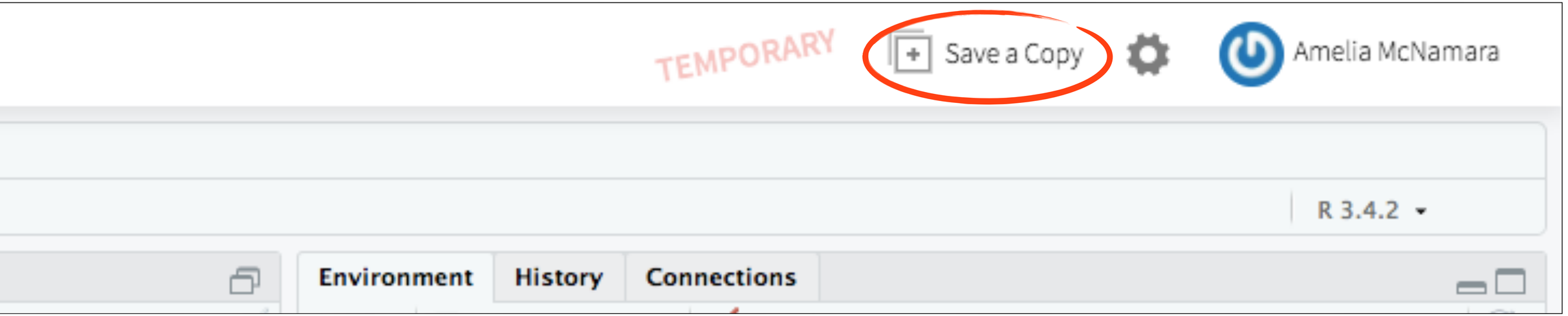
Introduce yourself to your nearest 2-4 neighbors.

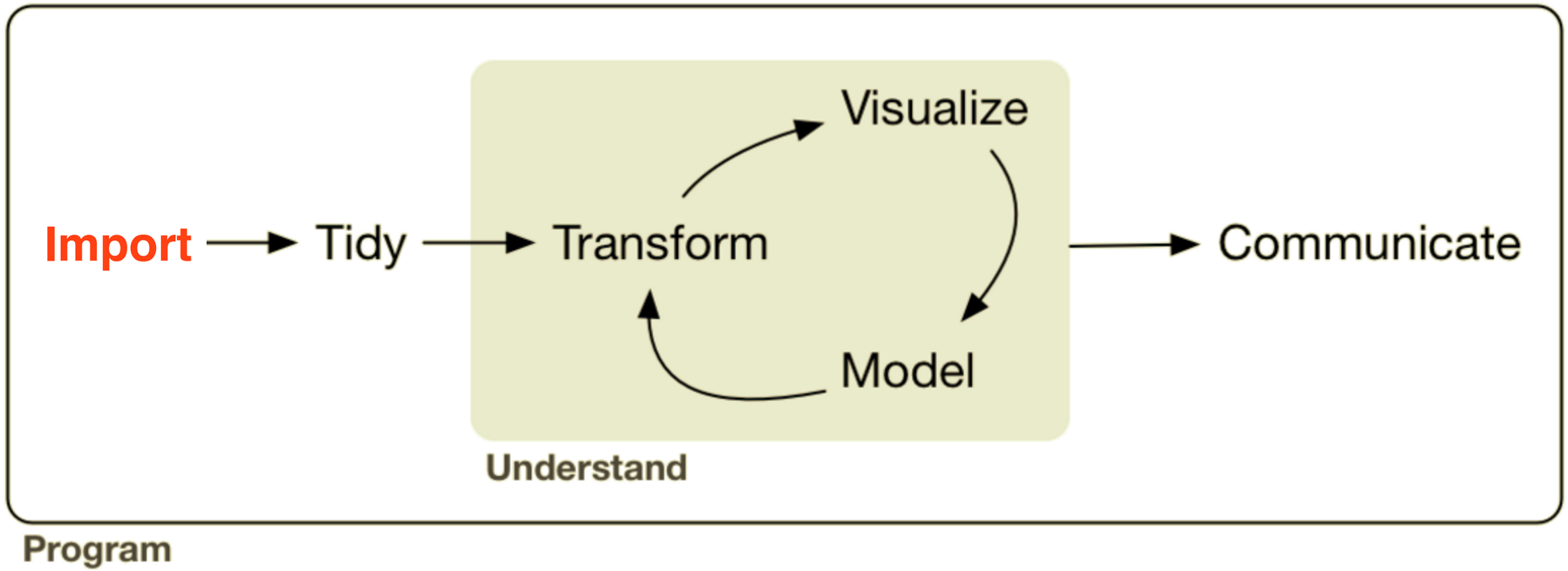
Schedule

9:00 -10:30	Import
10:30 - 11:00	Morning break (Seaport Foyer)
11:00 - 12:00	Best practices
12:00 - 1:00	Lunch (Grand Ballroom)
1:00 - 3:00	Transform and tidy
3:00 - 3:30	Afternoon break (Seaport Foyer)
3:30 - 5:00	Modeling and going forward

https://rstudio.cloud/project/13632

Save a Copy





From *R for Data Science* by Hadley Wickham and Garrett Grolemund.

Importing Data

nimbus.csv

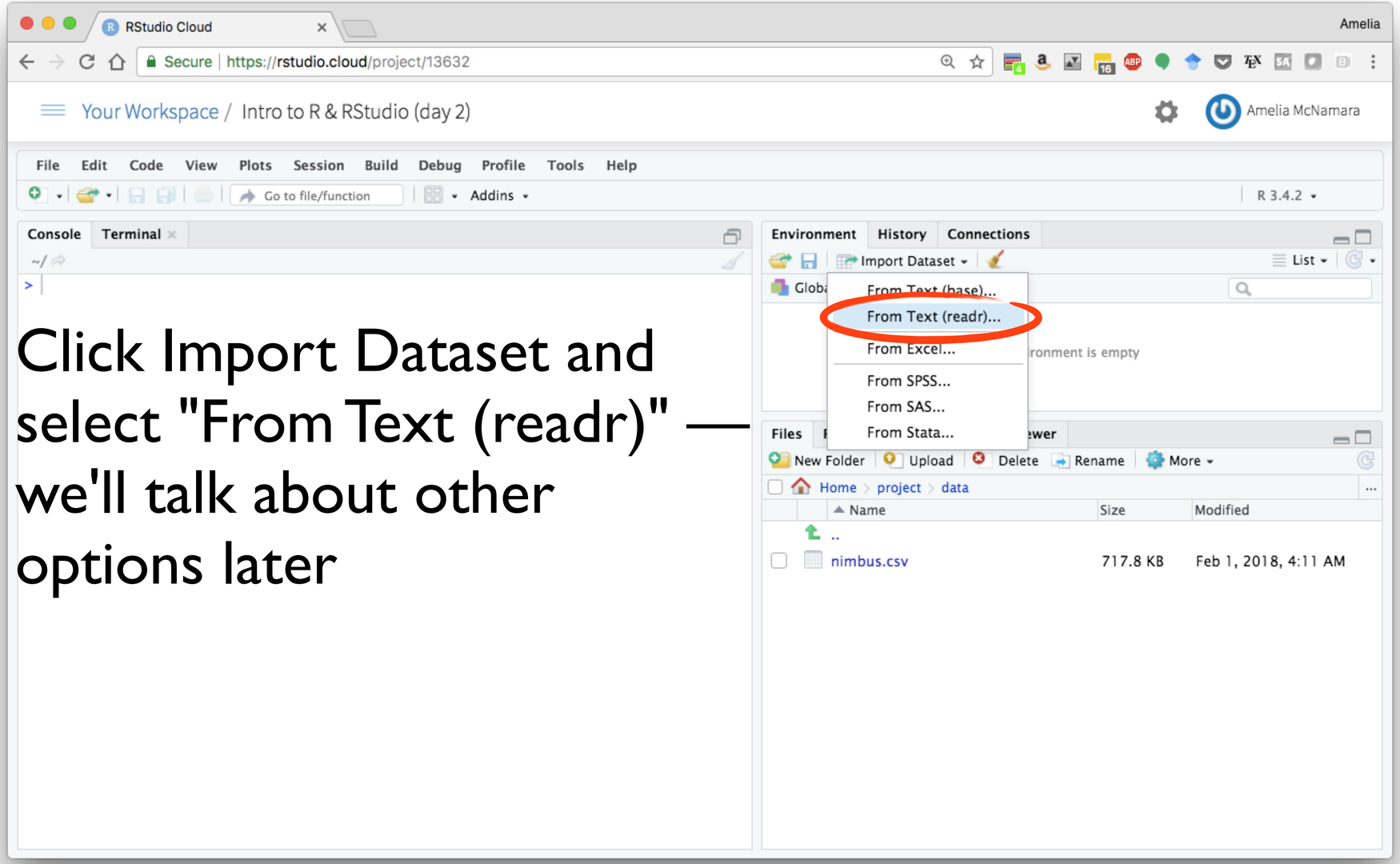
```
date,longitude,latitude,ozone
1985-10-01T00:00:00Z,-179.375,-87.5,.
1985-10-01T00:00:00Z,-178.125,-87.5,.
1985-10-01T00:00:00Z,-176.875,-87.5,.
1985-10-01T00:00:00Z,-175.625,-87.5,.
1985-10-01T00:00:00Z,-174.375,-87.5,.
1985-10-01T00:00:00Z,-173.125,-87.5,.
1985-10-01T00:00:00Z,-171.875,-87.5,.
1985-10-01T00:00:00Z,-170.625,-87.5,.
1985-10-01T00:00:00Z,-169.375,-87.5,.
```

nimbus.csv

```
date,longitude,latitude,ozone
1985-10-01T00:00:00Z,-179.375,-87.5,.
1985-10-01T00:00:00Z,-178.125,-87.5,.
1985-10-01T00:00:00Z,-176.875,-87.5,.
1985-10-01T00:00:00Z,-175.625,-87.5,.
1985-10-01T00:00:00Z,-174.375,-87.5,.
1985-10-01T00:00:00Z,-173.125,-87.5,.
1985-10-01T00:00:00Z,-171.875,-87.5,.
1985-10-01T00:00:00Z,-170.625,-87.5,.
1985-10-01T00:00:00Z,-169.375,-87.5,.
```

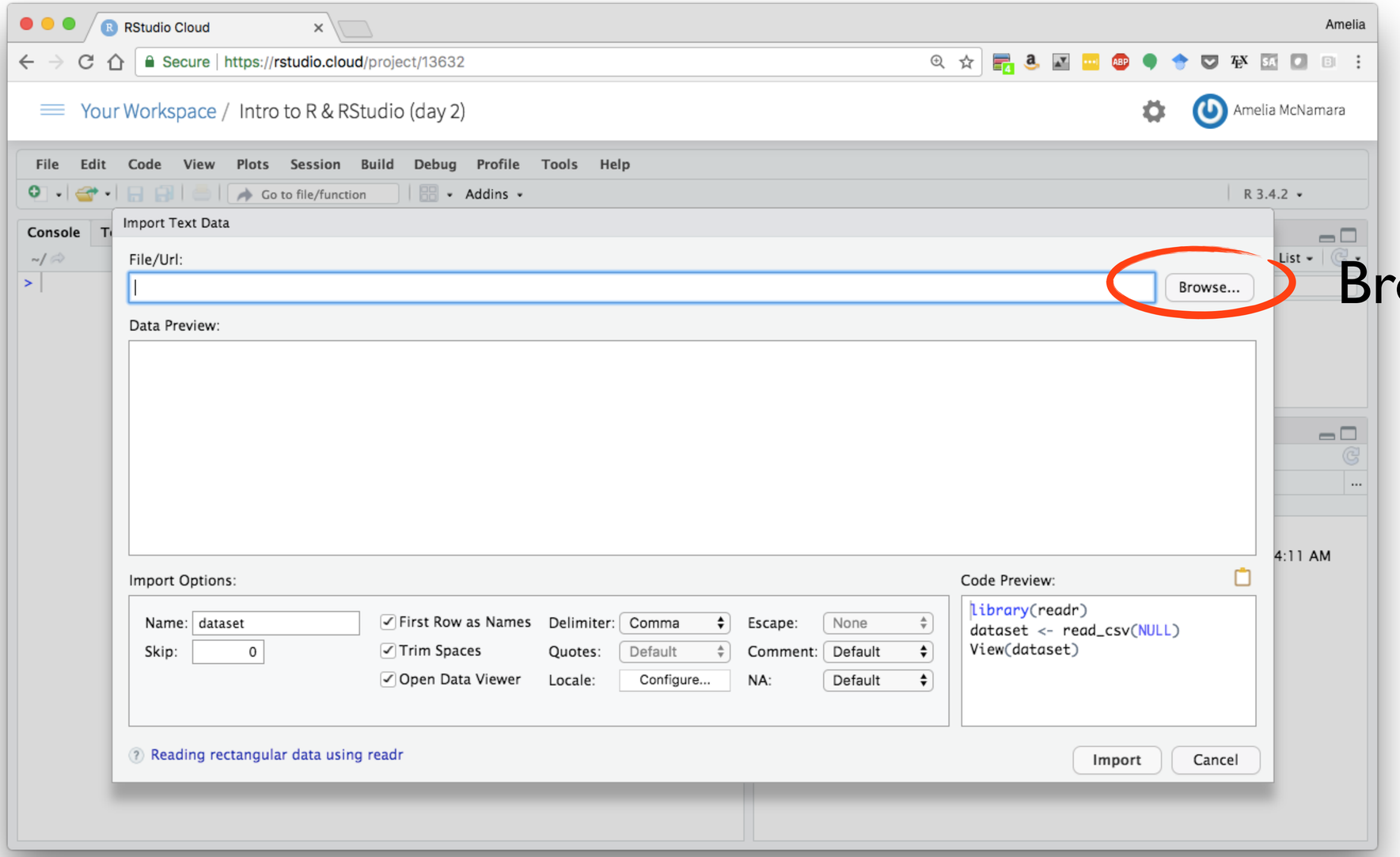
RStudio interface

Using the RStudio interface



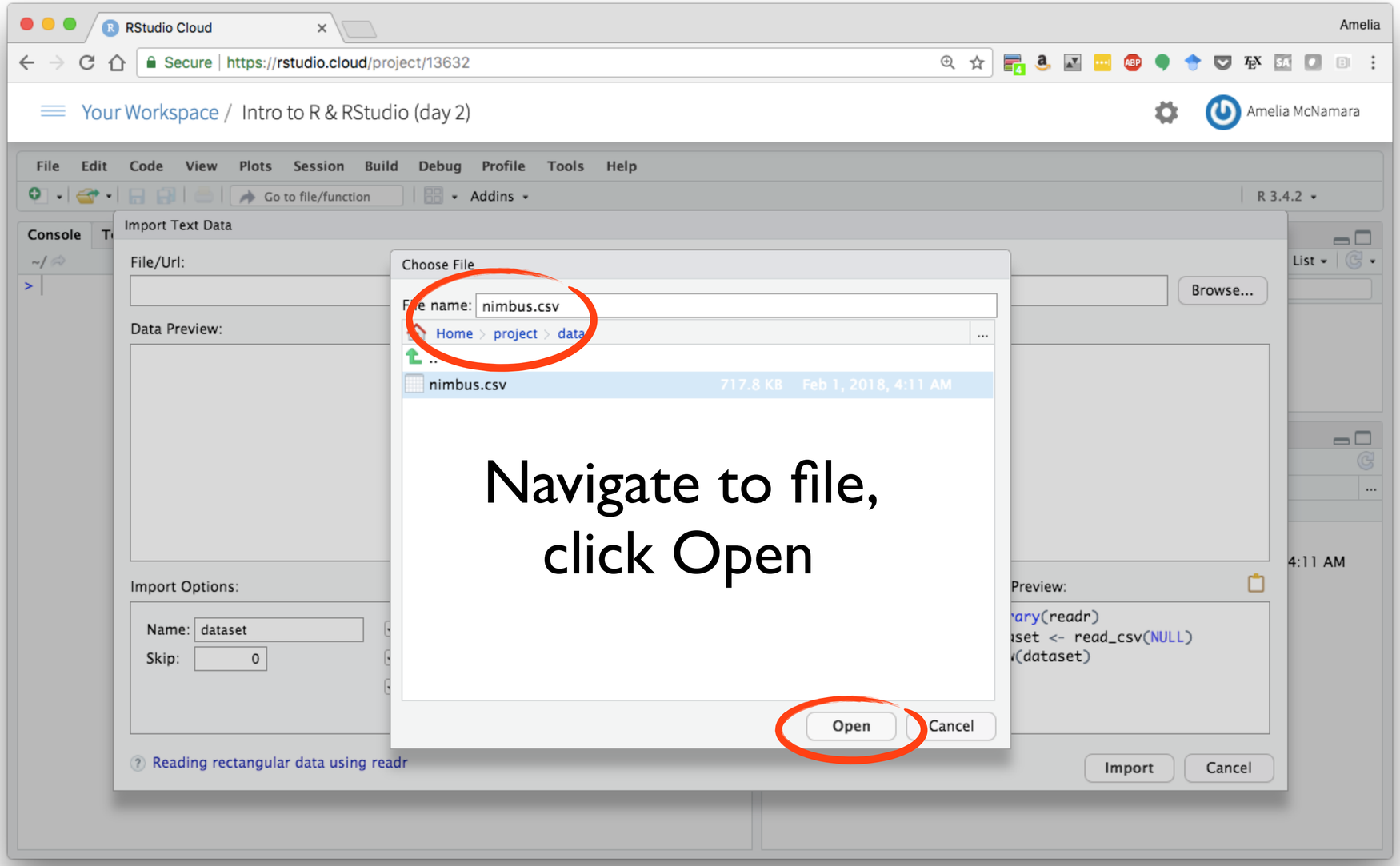
Click Import Dataset and select "From Text (readr)" — we'll talk about other options later

Using the RStudio interface

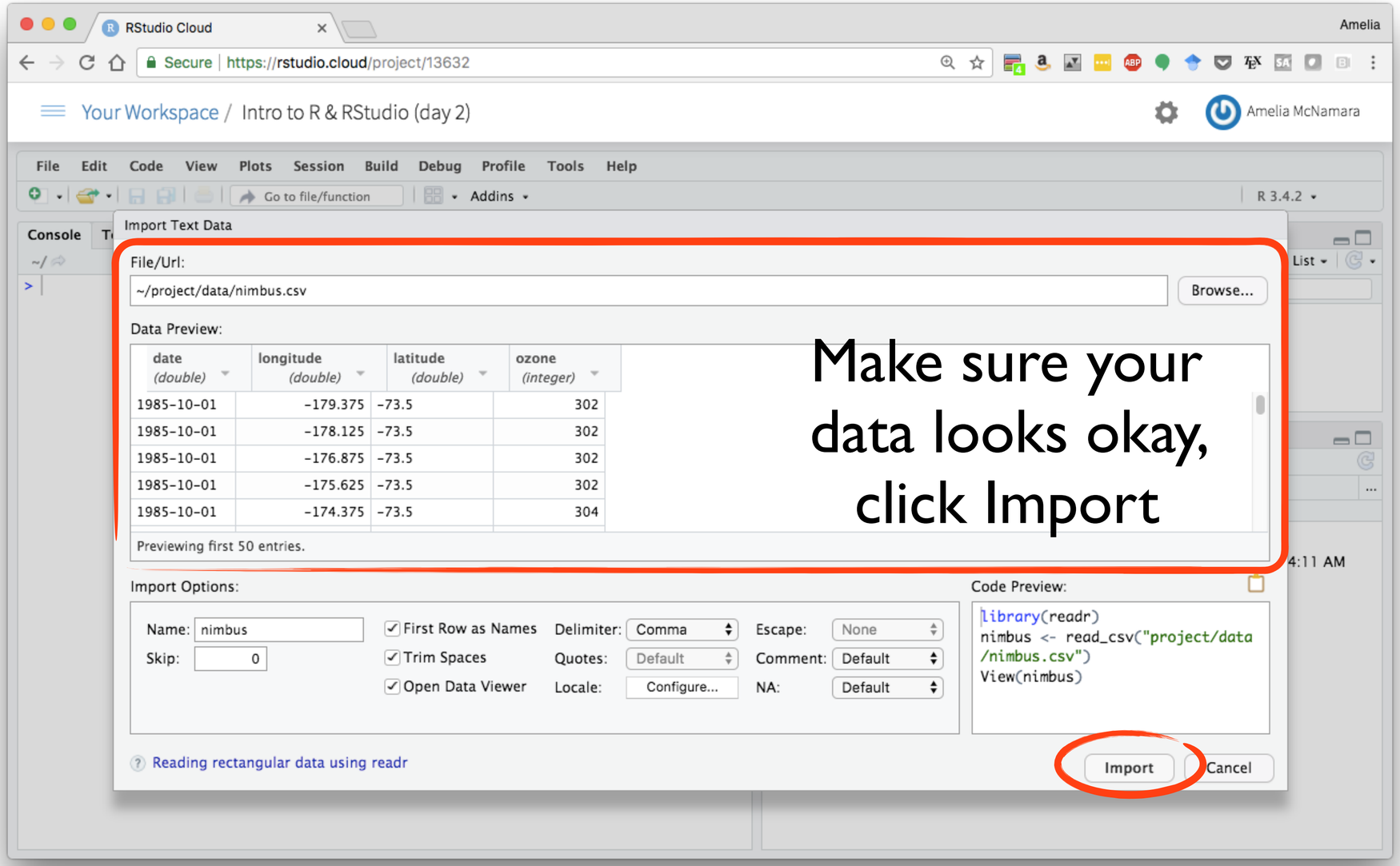


Browse files

Using the RStudio interface

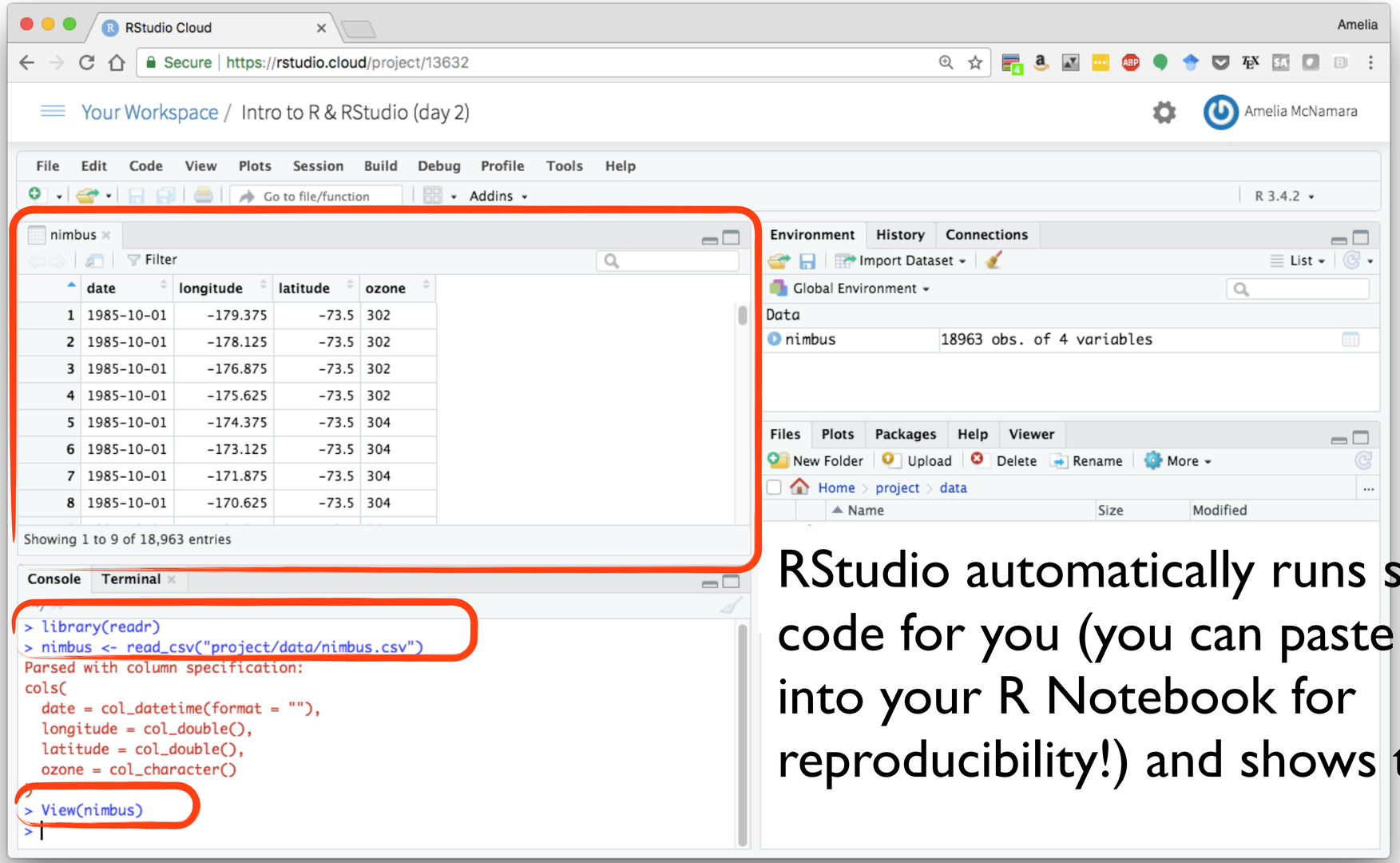


Using the RStudio interface



Using the RStudio interface

Another place where it looks like you got an error, but it's actually just a message



RStudio automatically runs some code for you (you can paste this into your R Notebook for reproducibility!) and shows the View

Your turn

Use the RStudio interface to import the GSS.csv dataset using readr

Your turn

Use the RStudio interface to import the GSS.csv dataset using readr

```
GSS <- read_csv("project/data/GSS.csv")
```

skimr

```
GSS %>%
     skim()
Skim summary statistics
 n obs: 2540
 n variables: 15

Variable type: character
              variable missing complete    n min max empty n_unique
                   Age       2     2538 2540   9  11     0       73
 ChildhoodFamilyIncome       2     2538 2540   7  17     0        7
           LaborStatus       2     2538 2540   5  16     0        9
         MaritalStatus       2     2538 2540   7  13     0        6
       OpinionOfIncome       2     2538 2540   7  17     0        7
        PoliticalParty       2     2538 2540   9  18     0       10
                  Race       2     2538 2540   5   5     0        3
      RespondentIncome       2     2538 2540   7  14     0       15
                   Sex       2     2538 2540   4   6     0        2
     SexualOrientation       2     2538 2540   8  27     0        6
     TotalFamilyIncome       2     2538 2540   7  14     0       14

Variable type: integer
               variable missing complete    n    mean     sd   p0     p25 median     p75
 HighestSchoolCompleted       3     2537 2540   13.7    3.07    0   12      14     16
                     ID       2     2538 2540 1271.22 734.76    1  635.25 1269.5 1908.75
            NumChildren      25     2515 2540    1.78   1.56    0    0       2      3
                   Year       2     2538 2540 2014      0    2014 2014    2014   2014
 p100     hist
   20 ▁▁▁▁▂▆▆▃▂
 2543 ▇▇▇▇▇▇▇▇
    7 ▆▂▃▆▂▁▁▁▁
 2014 ▁▁▁▁▁▇▁▁▁▁
```

It's always a good idea to start with summary statistics after you load in data, to do a sanity check for whether it looks right

Sometimes, you don't want all the summary statistics skimr gives you. You can use `skim_with()` to remove them

```
skim_with(integer = list(p25 = NULL, p75=NULL))
```

`skim_with()` can also be used to add additional summary statistics (e.g. a trimmed mean) but we won't discuss this

If you change your mind about your changes, use `skim_with_defaults()` to put them back

```
skim_with_defaults()
```

Your turn

Run the `skim_with()` command, and then try skimming GSS again to see how the output is different

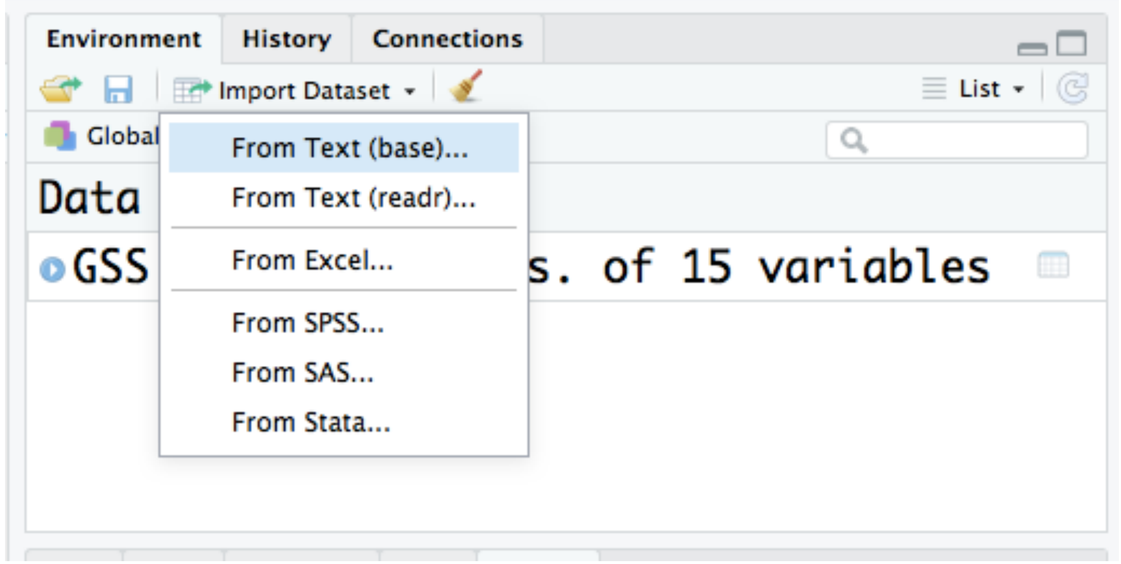
```
GSS %>%
     skim()
```

```
Skim summary statistics
 n obs: 2540
 n variables: 15

Variable type: character
              variable missing complete    n min max empty n_unique
                   Age       2     2538 2540   9  11     0       73
 ChildhoodFamilyIncome       2     2538 2540   7  17     0        7
           LaborStatus       2     2538 2540   5  16     0        9
         MaritalStatus       2     2538 2540   7  13     0        6
       OpinionOfIncome       2     2538 2540   7  17     0        7
        PoliticalParty       2     2538 2540   9  18     0       10
                  Race       2     2538 2540   5   5     0        3
      RespondentIncome       2     2538 2540   7  14     0       15
                   Sex       2     2538 2540   4   6     0        2
     SexualOrientation       2     2538 2540   8  27     0        6
     TotalFamilyIncome       2     2538 2540   7  14     0       14

Variable type: integer
               variable missing complete    n    mean     sd   p0 median p100     hist
 HighestSchoolCompleted       3     2537 2540   13.7    3.07    0   14     20 ▁▁▁▁▂▆▇▅▂
                     ID       2     2538 2540 1271.22 734.76    1 1269.5 2543 ▇▇▇▇▇▇▇▇
            NumChildren      25     2515 2540    1.78   1.56    0    2      7 ▇▃▅▃▂▁▁▁
                   Year       2     2538 2540 2014      0    2014 2014   2014 ▁▁▁▁▇▁▁▁▁
```

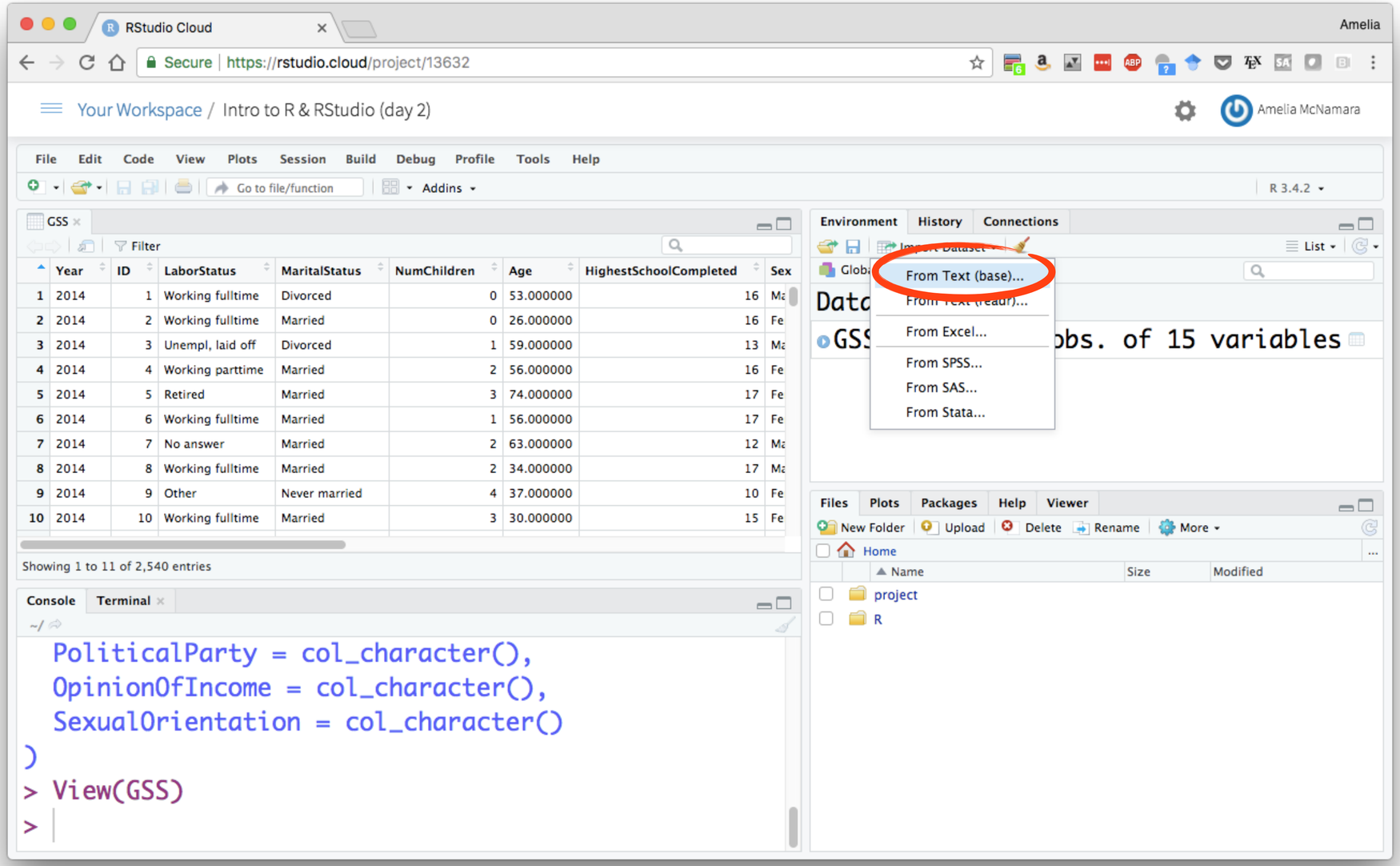
read.csv()

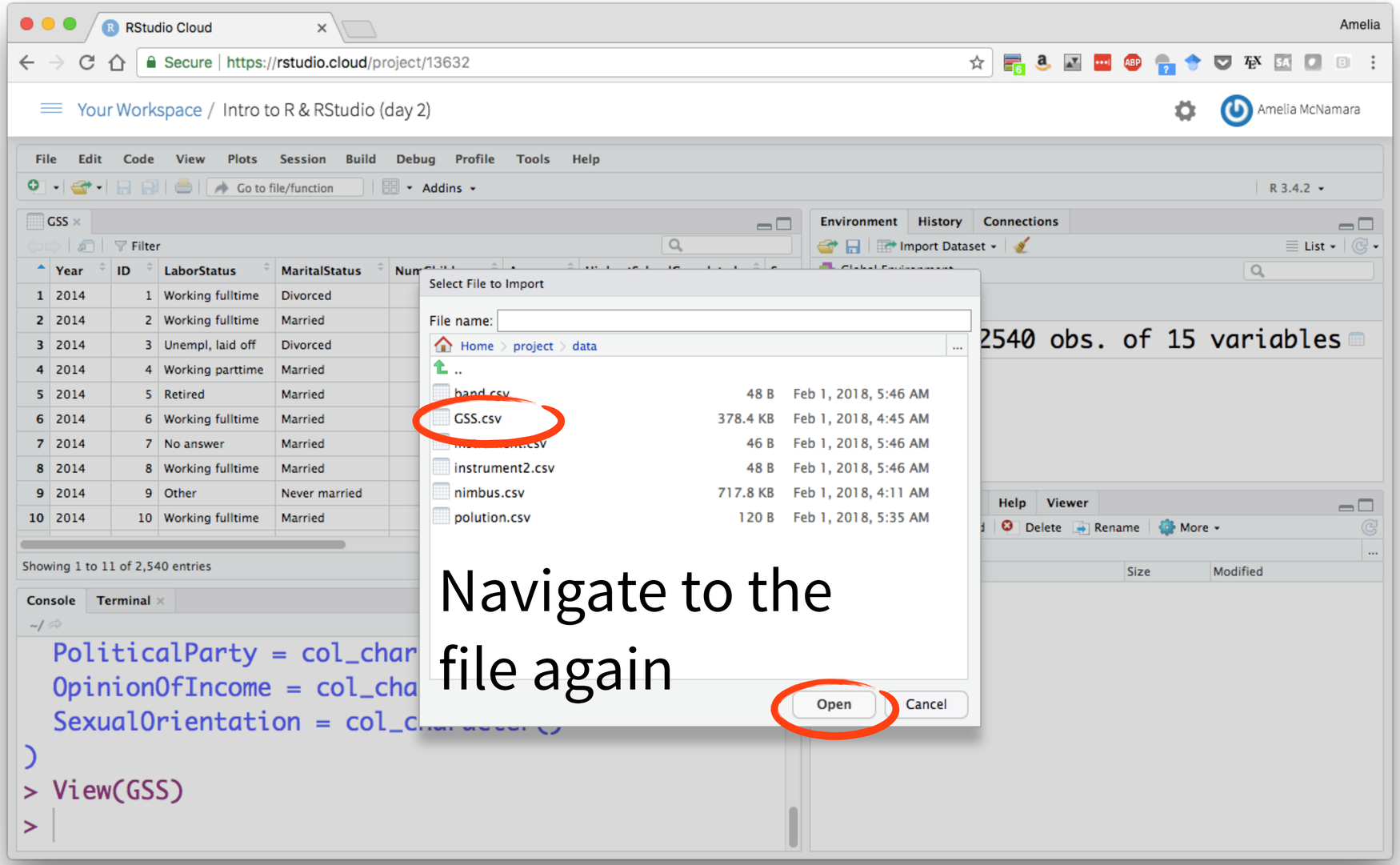
`read.csv()` is the base R version of `read_csv()`

I don't recommend using it, because its defaults are not as good, and it's much slower.



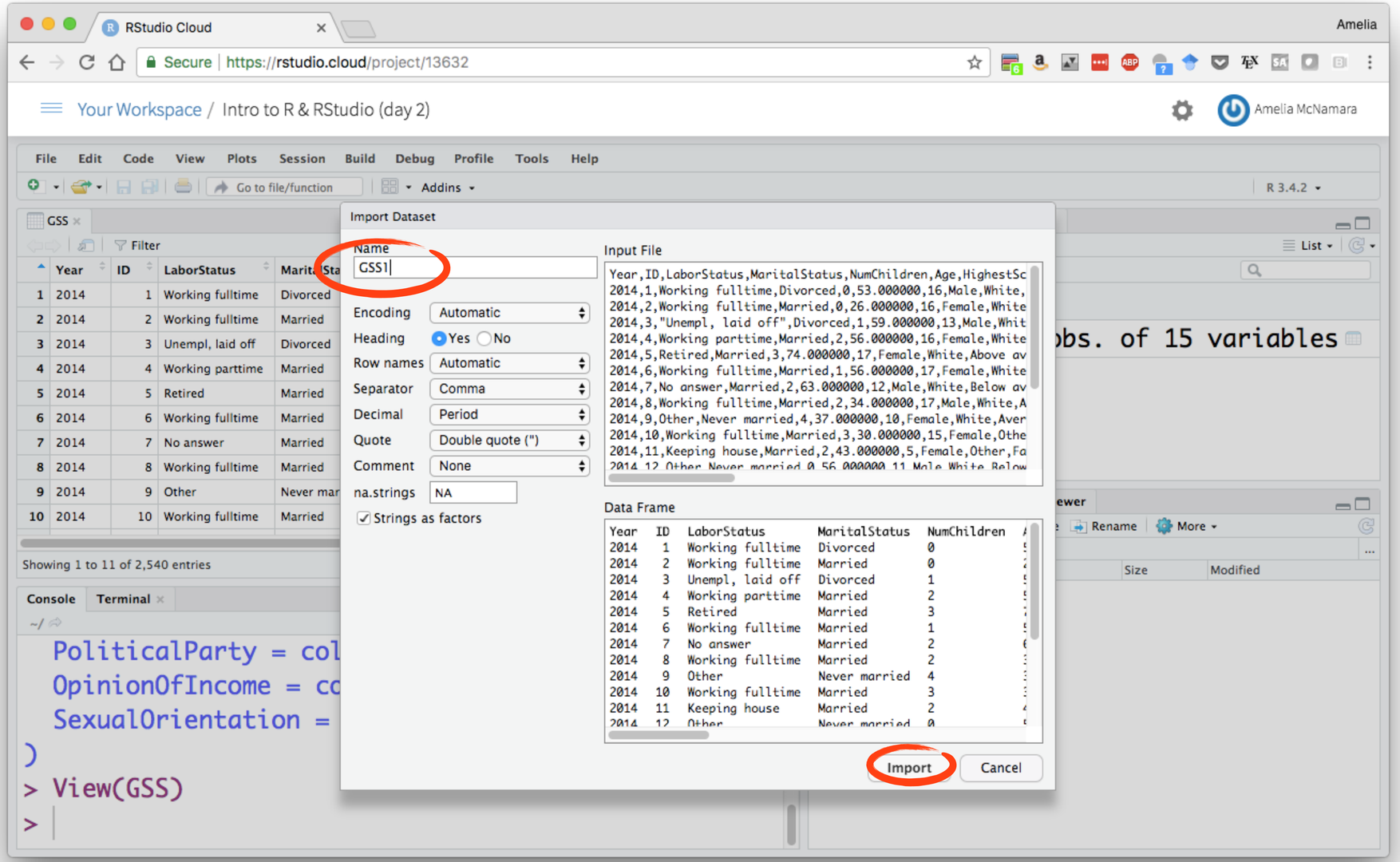
Let's compare how the GSS data looks if we read it in with `read.csv()` rather than `read_csv()`


RStudio Cloud
Amelia
Secure | https://rstudio.cloud/project/13632
Your Workspace / Intro to R & RStudio (day 2)
Amelia McNamara
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 3.4.2
GSS
Filter
Year ID LaborStatus MaritalStatus NumChildren Age HighestSchoolCompleted Sex
1 2014 1 Working fulltime Divorced 0 53.000000 16 Ma
2 2014 2 Working fulltime Married 0 26.000000 16 Fe
3 2014 3 Unempl, laid off Divorced 1 59.000000 13 Ma
4 2014 4 Working parttime Married 2 56.000000 16 Fe
5 2014 5 Retired Married 3 74.000000 17 Fe
6 2014 6 Working fulltime Married 1 56.000000 17 Fe
7 2014 7 No answer Married 2 63.000000 12 Ma
8 2014 8 Working fulltime Married 2 34.000000 17 Ma
9 2014 9 Other Never married 4 37.000000 10 Fe
10 2014 10 Working fulltime Married 3 30.000000 15 Fe
Showing 1 to 11 of 2,540 entries
Console Terminal
~/
PoliticalParty = col_character(),
OpinionOfIncome = col_character(),
SexualOrientation = col_character()
)
> View(GSS)
>
Environment History Connections
List
From Text (base)...
From Excel...
From SPSS...
From SAS...
From Stata...
obs. of 15 variables
Files Plots Packages Help Viewer
New Folder Upload Delete Rename More
Home
Name Size Modified
project
R


RStudio Cloud
Amelia
Secure | https://rstudio.cloud/project/13632
Your Workspace / Intro to R & RStudio (day 2)
Amelia McNamara
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 3.4.2
GSS
Filter
Year ID LaborStatus MaritalStatus
1 2014 1 Working fulltime Divorced
2 2014 2 Working fulltime Married
3 2014 3 Unempl, laid off Divorced
4 2014 4 Working parttime Married
5 2014 5 Retired Married
6 2014 6 Working fulltime Married
7 2014 7 No answer Married
8 2014 8 Working fulltime Married
9 2014 9 Other Never married
10 2014 10 Working fulltime Married
Showing 1 to 11 of 2,540 entries
Environment History Connections
Import Dataset
List
2540 obs. of 15 variables
Help Viewer
Delete Rename More
Size Modified
Console Terminal
~/
PoliticalParty = col_char
OpinionOfIncome = col_cha
SexualOrientation = col_c
)
> View(GSS)
>
Select File to Import
File name:
Home project data
..
band.csv 48 B Feb 1, 2018, 5:46 AM
GSS.csv 378.4 KB Feb 1, 2018, 4:45 AM
46 B Feb 1, 2018, 5:46 AM
instrument2.csv 48 B Feb 1, 2018, 5:46 AM
nimbus.csv 717.8 KB Feb 1, 2018, 4:11 AM
polution.csv 120 B Feb 1, 2018, 5:35 AM
Open
Cancel
Navigate to the file again

Again, make sure the data preview looks okay

Give it a different name to distinguish from our first dataset. I chose GSS1



Your turn

Try skimming GSS1 and see how the output is different from GSS

Lots of factors (which I've told you to watch out for!)

```
GSS1 %>%
     skim()
Skim summary statistics
 n obs: 2540
 n variables: 15

Variable type: factor
             variable missing complete    n n_unique                                  top_counts ordered
                  Age       2     2538 2540       73                               53.: 60, 56   FALSE
 ChildhoodFamilyIncome       2     2538 2540        7 Ave: 1118, Bel: 656, Abo: 403, Far: 271   FALSE
           LaborStatus       2     2538 2540        9 Wor: 1230, Ret: 460, Wor: 273, Kee: 263   FALSE
         MaritalStatus       2     2538 2540        6 Mar: 1158, Nev: 675, Div: 411, Wid: 209   FALSE
       OpinionOfIncome       2     2538 2540        7 Ave: 1118, Bel: 666, Abo: 483, Far: 179   FALSE
        PoliticalParty       2     2538 2540       10  Ind: 502, Str: 419, Not: 406, Ind: 337   FALSE
                  Race       2     2538 2540        3    Whi: 1890, Bla: 386, Oth: 262, NA: 2   FALSE
     RespondentIncome       2     2538 2540       15  $25: 951, Not: 923, $20: 143, $10: 111   FALSE
                   Sex       2     2538 2540        2             Fem: 1397, Mal: 1141, NA: 2   FALSE
    SexualOrientation       2     2538 2540        6   Het: 2195, Not: 188, Bis: 65, Gay: 45   FALSE
    TotalFamilyIncome       2     2538 2540       14 $25: 1684, $20: 157, $10: 150, Ref: 126   FALSE

Variable type: integer
             variable missing complete    n    mean     sd   p0     p25 median     p75 p100     hist
 HighestSchoolCompleted       3     2537 2540   13.7    3.07    0   12      14      16      20 ▁▁▂▅▃▂▁▁
                   ID       2     2538 2540 1271.22 734.76    1  635.25 1269.5 1908.75 2543 ▇▇▇▇▇▇▇▇
            NumChildren      25     2515 2540    1.78   1.56    0    0       2       3       7 ▇▃▅▂▂▁▁▁
                   Year       2     2538 2540 2014      0    2014 2014    2014    2014    2014 ▁▁▁▇▁▁▁▁
```

read.csv() vs. read_csv()

Console Terminal

```
50       Below average                    Bisexual
51       Above average    Heterosexual or straight
52       Above average    Heterosexual or straight
53       Below average    Heterosexual or straight
54       Above average    Heterosexual or straight
55       Above average    Heterosexual or straight
56       Below average    Heterosexual or straight
57   Far below average                    Bisexual
58             Average    Heterosexual or straight
59       Above average Gay, lesbian, or homosexual
60       Above average    Heterosexual or straight
61       Below average Gay, lesbian, or homosexual
62       Below average    Heterosexual or straight
63             Average    Heterosexual or straight
64       Above average    Heterosexual or straight
65       Above average    Heterosexual or straight
66       Above average    Heterosexual or straight
 [ reached getOption("max.print") -- omitted 2474 rows ]
>
```

Console Terminal

```
> GSS
# A tibble: 2,540 x 15
    Year    ID LaborStatus      MaritalStatus NumChildren Age
   <int> <int> <chr>            <chr>                <int> <chr>
 1  2014     1 Working fulltime Divorced                 0 53.000000
 2  2014     2 Working fulltime Married                  0 26.000000
 3  2014     3 Unempl, laid off Divorced                 1 59.000000
 4  2014     4 Working parttime Married                  2 56.000000
 5  2014     5 Retired          Married                  3 74.000000
 6  2014     6 Working fulltime Married                  1 56.000000
 7  2014     7 No answer        Married                  2 63.000000
 8  2014     8 Working fulltime Married                  2 34.000000
 9  2014     9 Other            Never married            4 37.000000
10  2014    10 Working fulltime Married                  3 30.000000
# ... with 2,530 more rows, and 9 more variables:
#   HighestSchoolCompleted <int>, Sex <chr>, Race <chr>,
#   ChildhoodFamilyIncome <chr>, TotalFamilyIncome <chr>,
#   RespondentIncome <chr>, PoliticalParty <chr>,
#   OpinionOfIncome <chr>, SexualOrientation <chr>
>
```

read_csv()

Why use read_csv()?

Compared to the base R functions, readr functions:

1. Are ~ 10 times faster
2. Return tibbles
3. Have more intuitive defaults. No row names, no strings as factors.

readr functions

function	reads
read_csv()	Comma separated values
read_csv2()	Semi-colon separated values
read_delim()	General delimited files
read_fwf()	Fixed width files
read_log()	Apache log files
read_table()	Space separated
read_tsv()	Tab delimited values

readr functions

function	reads
read_csv()	**Comma separated values**
read_csv2()	Semi-colon separated values
read_delim()	General delimited files
read_fwf()	Fixed width files
read_log()	Apache log files
read_table()	Space separated
read_tsv()	Tab delimited values

more programmatic

read_csv()

readr functions share a common syntax

```
df <- read_csv("path/to/file.csv", …)
```

object to save output into

path from working directory to file

Parsing

Missing values

In R, missing data is represented as NA, which is its own data type (like character, numeric, factor)

GSS

Filter

	Year	ID	LaborStatus	MaritalStatus	NumChildren	Age	HighestSchoolCompleted	Sex	Race	ChildhoodFamil
98	2014	98	Working fulltime	Never married	2	37.000000	12	Female	Black	Far below aver
99	2014	99	Working fulltime	No answer	1	No answer	17	Female	Black	Average
100	2014	100	Unempl, laid off	Widowed	NA	62.000000	11	Male	Black	Far below aver
101	2014	101	Working fulltime	Never married	3	43.000000	13	Female	Black	Above average
102	2014	102	Working fulltime	Never married	0	32.000000	18	Male	White	Average
103	2014	103	Working fulltime	Widowed	3	65.000000	12	Female	Black	Far below aver
104	2014	104	Working fulltime	Married	0	32.000000	16	Female	White	Average
105	2014	105	Working fulltime	Married	0	51.000000	17	Male	White	Average
106	2014	106	Working fulltime	Married	1	41.000000	16	Male	White	Above average
107	2014	107	Retired	Widowed	2	81.000000	16	Male	White	Below average

But, not all data comes with missing values represented that way

Ways I've seen missing data represented:

- " "
- "NA" (character string)
- 9999
- -9999
- 7777
- "n/a"
- "."
- … etc. People are creative!

Luckily, R has a way to specify missing data formatting when you read in files, so you don't have to worry about adjusting these when you are doing data cleaning

. = NA

```
nimbus
```

GSS × nimbus ×

Filter

	date	longitude	latitude	ozone
123	1985-10-01	58.125	-73.5	180
124	1985-10-01	59.375	-73.5	180
125	1985-10-01	70.625	-73.5	.
126	1985-10-01	71.875	-73.5	.
127	1985-10-01	73.125	-73.5	.
128	1985-10-01	74.375	-73.5	.

read_csv()

readr functions share a common syntax

```
nimbus <- read_csv("nimbus.csv", na = ".")
```

object to save output into

path from working directory to file

Value(s) to convert to NA

Your Turn

Reread in **nimbus.csv**. But this time convert the "."'s to NA's. How many NA's are in the ozone column?

Your Turn

Reread in **nimbus.csv**. But this time convert the "."'s to NA's. How many NA's are in the ozone column?

```
nimbus <- read_csv("nimbus.csv", na = ".")
View(nimbus)
nimbus %>%
 skim(ozone)
Skim summary statistics
 n obs: 18963
 n variables: 4

Variable type: integer
 variable missing complete     n   mean    sd  p0 p25 median p75 p100     hist
    ozone     155    18808 18963 299.41 42.61 180 272    296 332  417
```

Quiz

What "type" of column is ozone?

```
nimbus <- read_csv("nimbus.csv", na = ".")
```

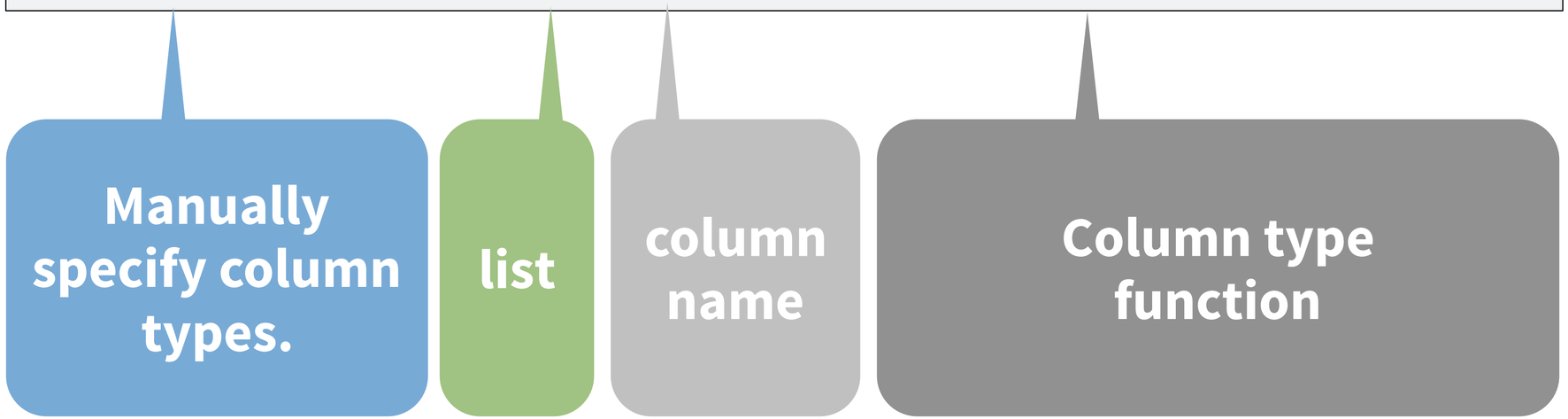
date <S3: POSIXct>	longitude <dbl>	latitude <dbl>	ozone <chr>
1985-10-01	-179.375	-87.5	*NA*
1985-10-01	-178.125	-87.5	*NA*
1985-10-01	-176.875	-87.5	*NA*
1985-10-01	-175.625	-87.5	*NA*
1985-10-01	-174.375	-87.5	*NA*
1985-10-01	-173.125	-87.5	*NA*
1985-10-01	-171.875	-87.5	*NA*
1985-10-01	-170.625	-87.5	*NA*
1985-10-01	-169.375	-87.5	*NA*
1985-10-01	-168.125	-87.5	*NA*

<chr> stands for character string (not a number)

read_csv()

readr functions share a common syntax

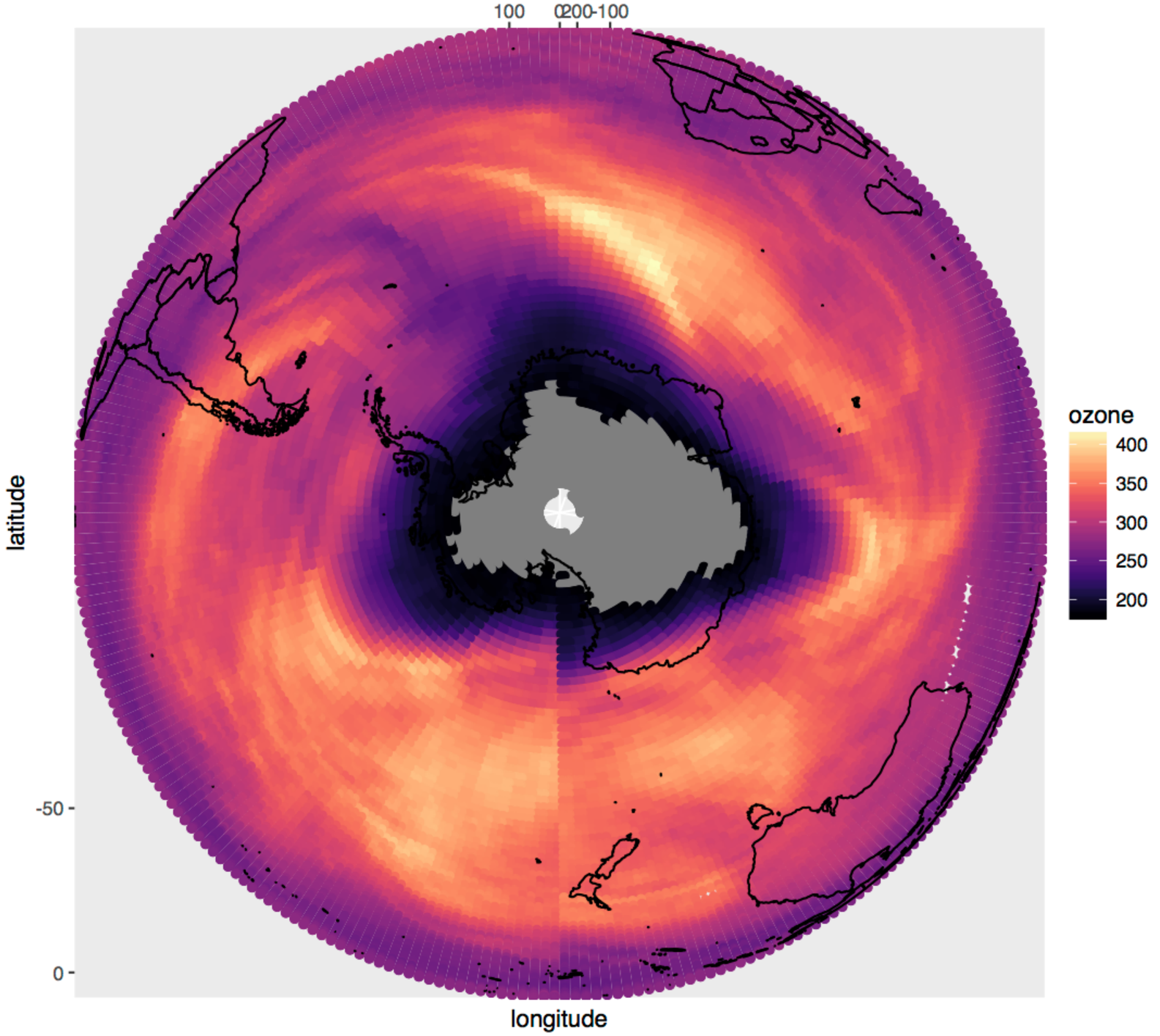
```
nimbus <- read_csv("project/data/nimbus.csv", na =
"."
  col_types = list(ozone = col_double()))
```



type function	data type
col_character()	character
col_date()	Date
col_datetime()	POSIXct (date-time)
col_double()	double (numeric)
col_factor()	factor
col_guess()	let readr guess (default)
col_integer()	integer
col_logical()	logical
col_number()	numbers mixed with non-number characters
col_numeric()	double or integer
col_skip()	do not read
col_time()	time

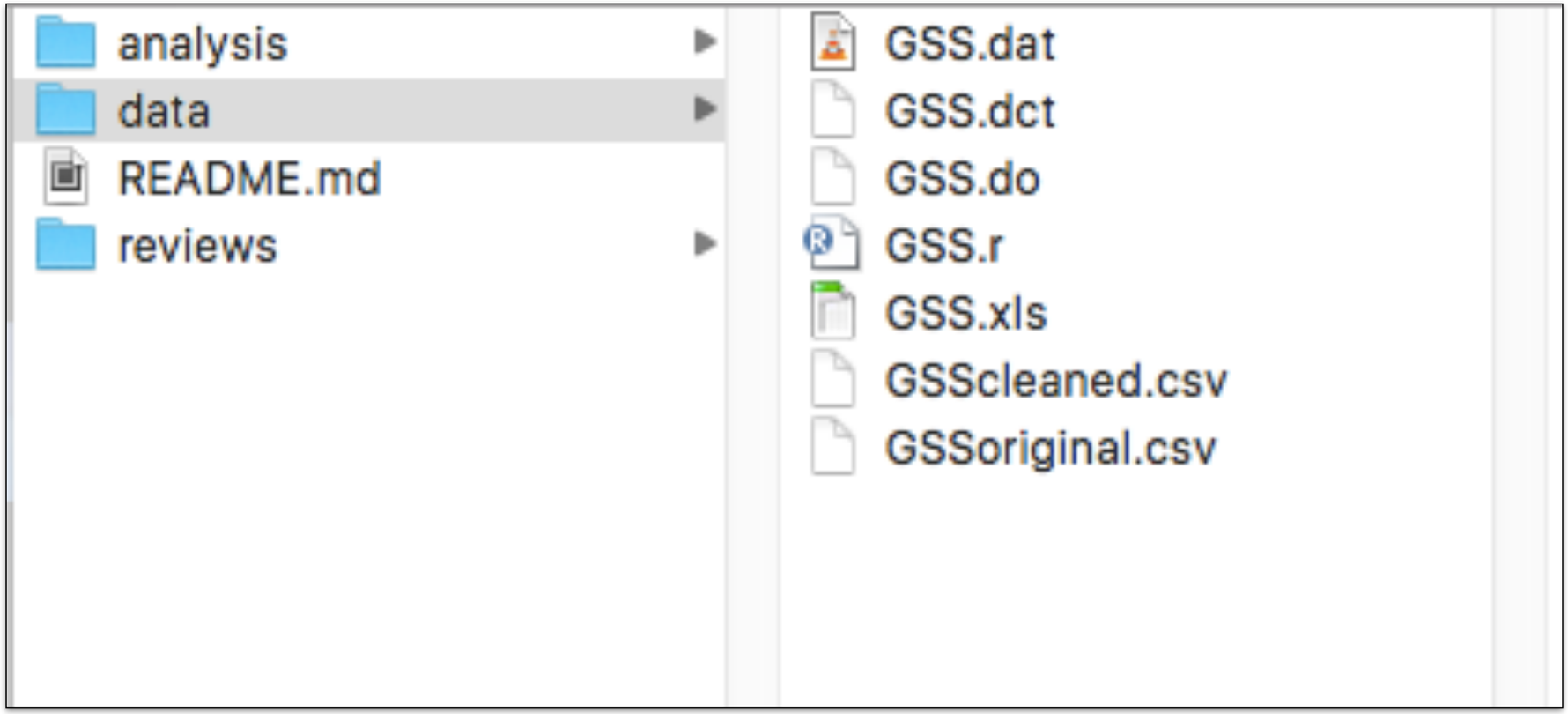
type function	data type
col_character()	character
col_date()	Date
col_datetime()	POSIXct (date-time)
col_double()	**double (numeric)**
col_factor()	factor
col_guess()	let readr guess (default)
col_integer()	integer
col_logical()	logical
col_number()	numbers mixed with non-number characters
col_numeric()	double or integer
col_skip()	do not read
col_time()	time

```
nimbus <- read_csv("nimbus.csv", na = ".",
  col_types = list(ozone = col_double()))

# this code requires the installation of the map and mapproj packages
# install.packages(c("map", "mapproj"))

library(viridis)
world <- map_data(map = "world")
nimbus %>%
  ggplot() +
    geom_point(aes(longitude, latitude, color = ozone)) +
    geom_path(aes(long, lat, group = group), data = world) +
    coord_map("ortho", orientation=c(-90, 0, 0)) +
    scale_color_viridis(option = "A")
```


100
0
200
-100
latitude
-50
0
longitude
ozone
400
350
300
250
200

Writing

Best practice is to create an R Notebook that begins with reading in your raw data, and then includes all the code to clean it up. This is reproducible! But, sometimes you don't want to have to run that code every time you begin your analysis. So, sometimes it makes sense to write out a file with your cleaned data.



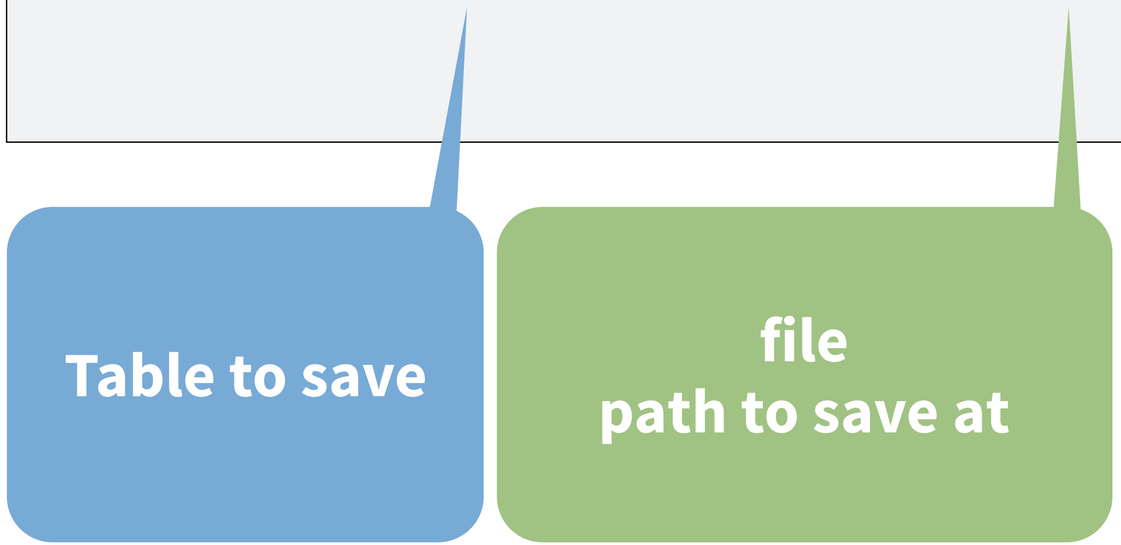
readr functions

function	writes
write_csv()	Comma separated values
write_excel_csv()	CSV intended for opening in Excel
write_delim()	General delimited files
write_file()	Single string, written as is
write_lines()	Vector of strings, one element per line
write_tsv()	Tab delimited values

write_csv()

Saves data set as a csv on your computer.

```
write_csv(nimbus, path = "nimbus2.csv")
```



Other types of data

package	accesses
haven	SPSS, Stata, and SAS files
readxl	excel files (.xls, .xlsx)
jsonlite	json
xml2	xml
httr	web API’s
rvest	web pages (web scraping)
DBI	databases
sparklyr	data loaded into spark

Spreadsheets

‹ All Collections

Collection idea for us?

Practical Data Science for Stats - a PeerJ Collection

Data Science | Statistics | Scientific Computing and Simulation | Computer Education | Computational Science | Social Computing | Software Engineering | Science and Medical Education | Computational Biology | Human-Computer Interaction | Anthropology | Programming Languages | Visual Analytics | Graphics | Data Mining and Machine Learning

Practical Data Science for Stats

The "Practical Data Science for Stats" Collection contains preprints focusing on the practical side of data science workflows and statistical analysis. Curated by Jennifer Bryan and Hadley Wickham.

Karl Broman, Kara Woo. Data organization in spreadsheets. PeerJ preprint and The American Statistician.
https://peerj.com/preprints/3183/

Abstract: Spreadsheets are widely used software tools for data entry, storage, analysis, and visualization. Focusing on the data entry and storage aspects, this paper offers practical recommendations for organizing spreadsheet data to reduce errors and ease later analyses. The basic principles are: be consistent, write dates like YYYY-MM-DD, don't leave any cells empty, put just one thing in a cell, organize the data as a single rectangle (with subjects as rows and variables as columns, and with a single header row), create a data dictionary, don't include calculations in the raw data files, don't use font color or highlighting as data, choose good names for things, make backups, use data validation to avoid data entry errors, and save the data in plain text file.

Your Turn (by request)

Find a dataset on your computer (or the internet) that you would like to import. Use the appropriate function to load it in, and skim() to see if it's being treated appropriately. Ask questions if you get stuck!