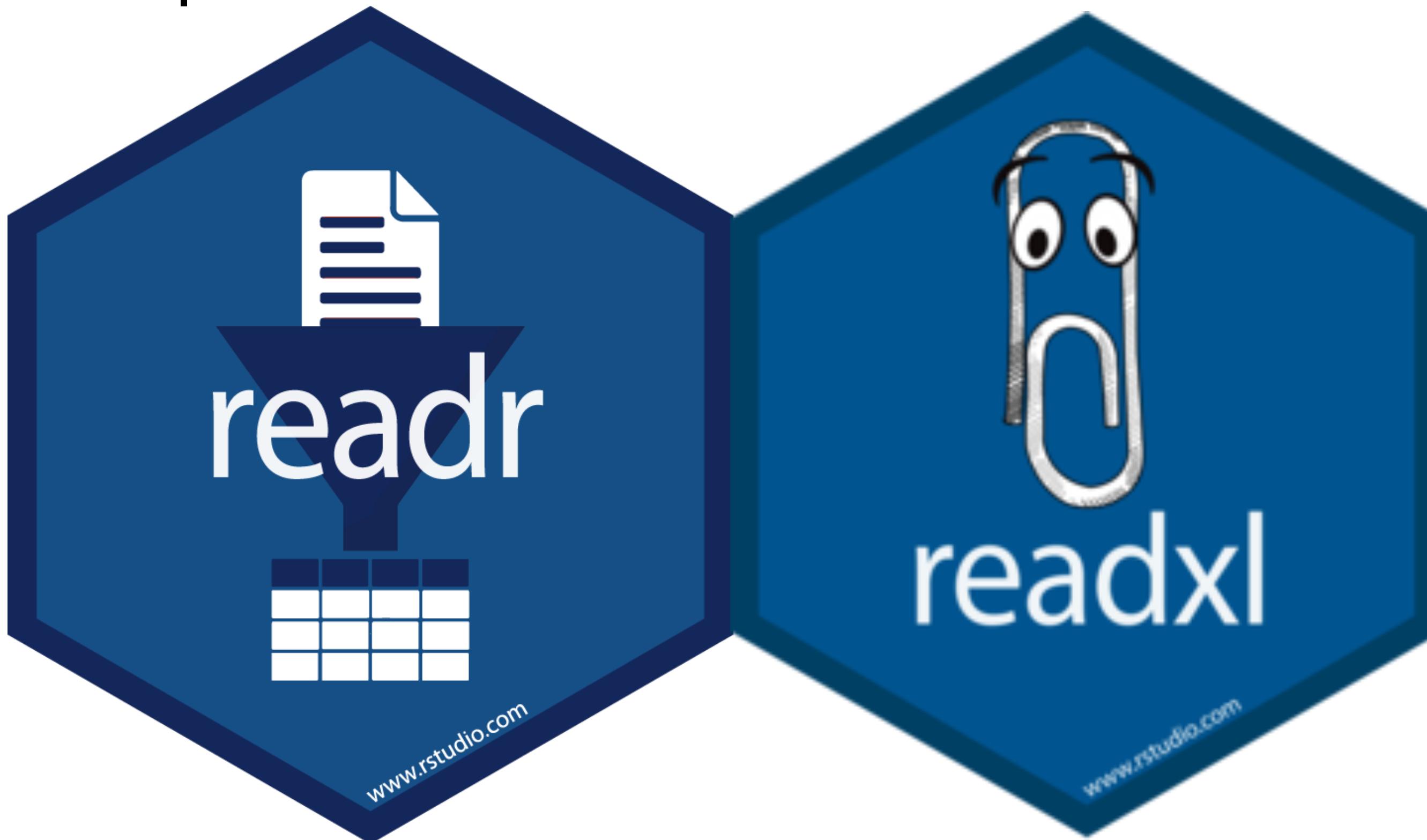


Import Data with

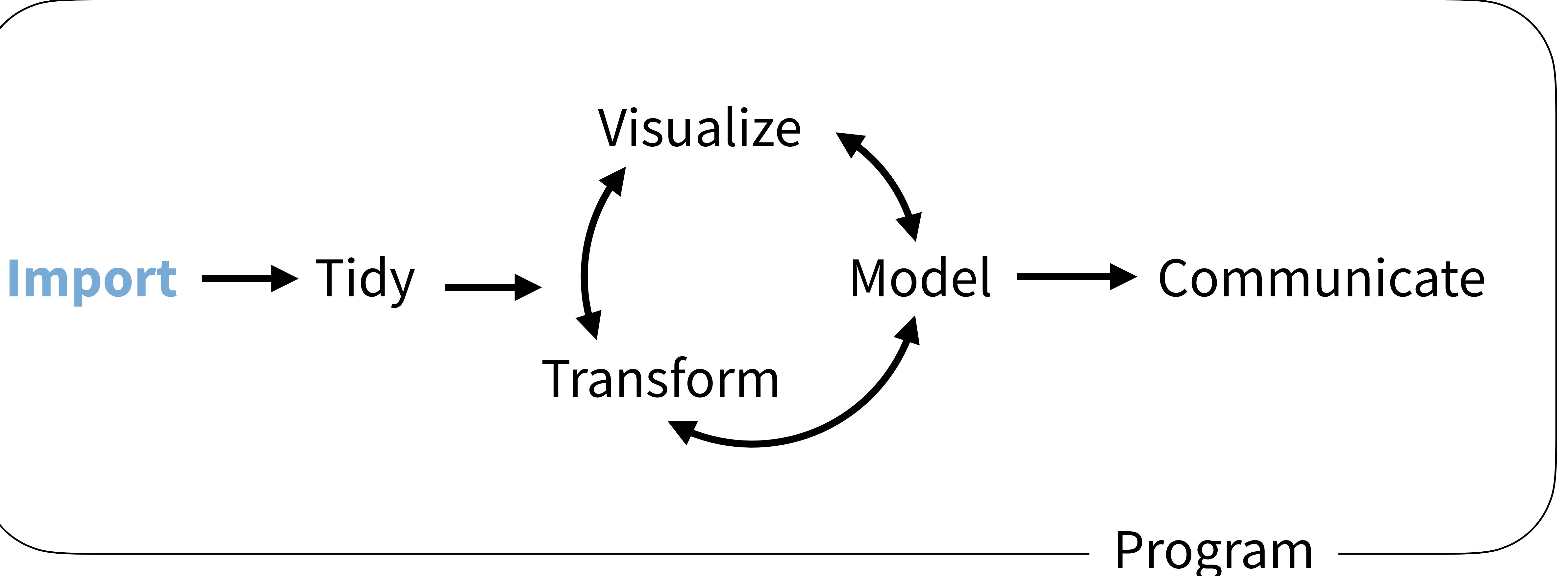


Exercise: Importing Data

- Split into groups and ask your partner:
 - *Where is the data that you want to analyze?*
 - *How do you currently access it?*



(Applied) Data Science



Data Import in the Tidyverse

A package for each storage type!

package	accesses
readr	csv, tsv, etc.
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



Reading in CSV Files



readr



Simple, consistent functions for working
with strings / csv data.

```
# install.packages("tidyverse")
library(readr)
```



Open Import-Data- Exercises.Rmd

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



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,.
```





`read_csv()`

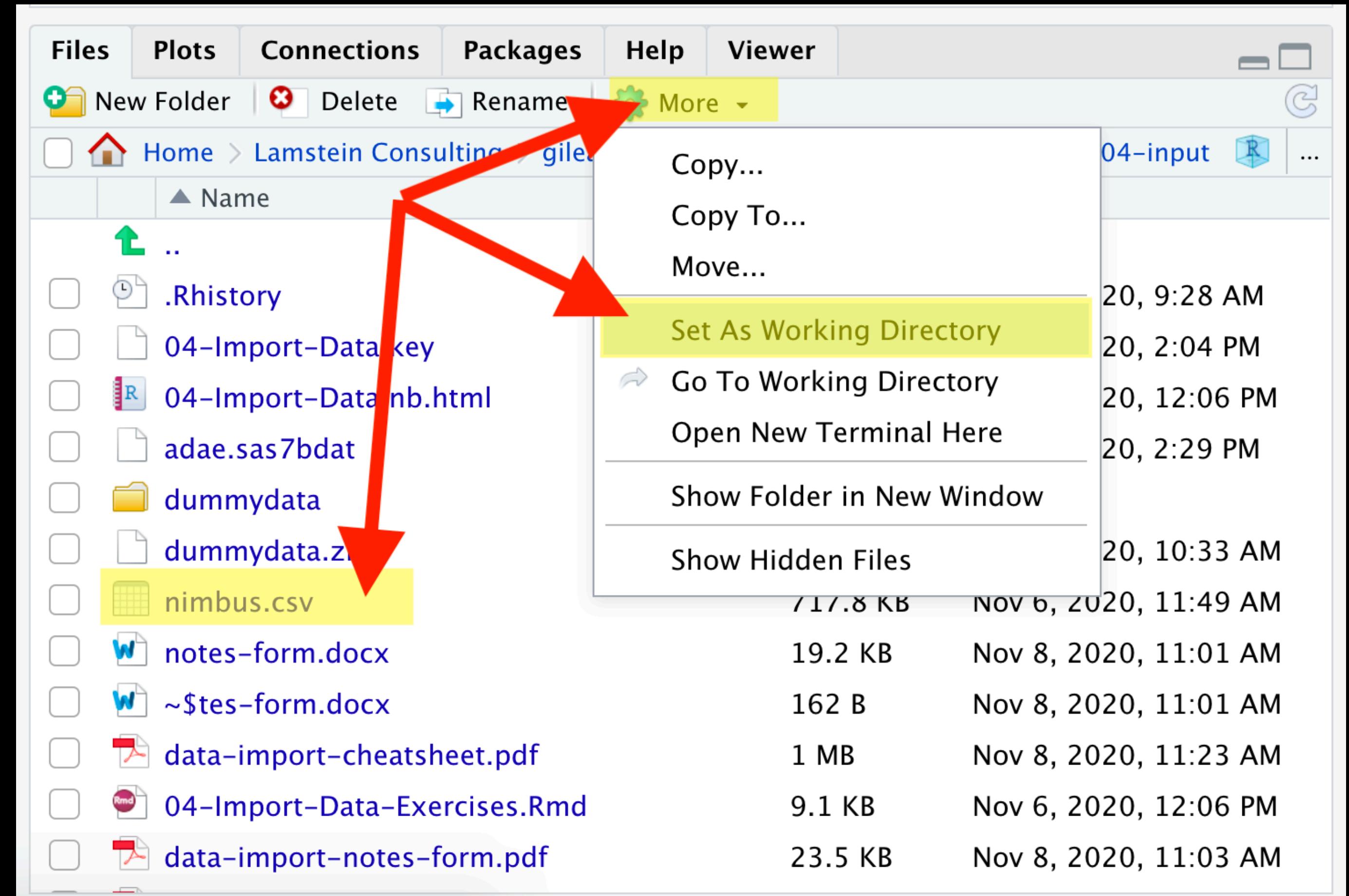
readr functions share a common syntax

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

object to save
output into

path from working
directory to file





Set Your Working Directory

Your Turn 1

Find **nimbus.csv** (in your working directory). Then read it into an object. Then view the results.

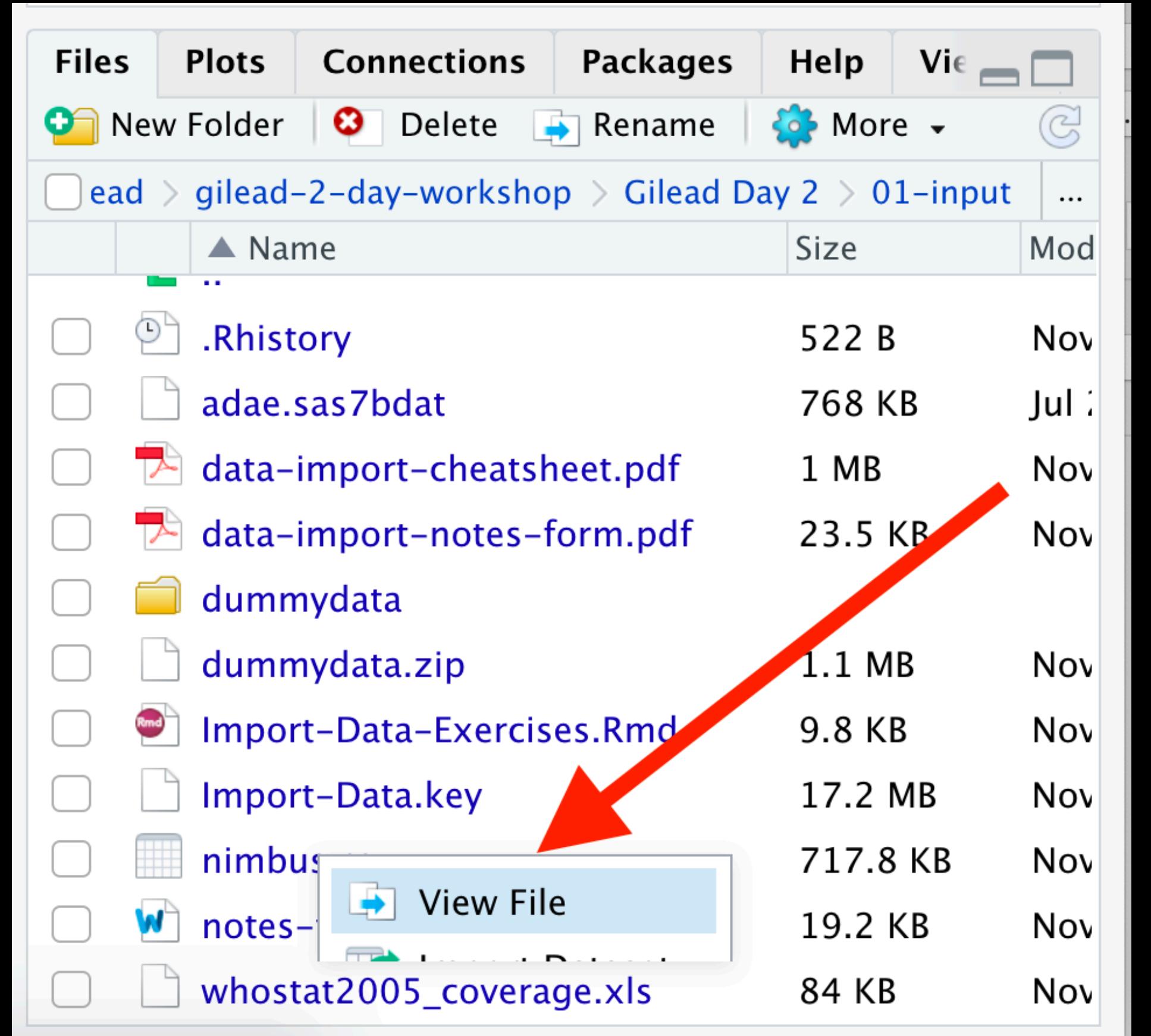


Your Turn 1

Find **nimbus.csv** (in your working directory). Then read it into an object. Then view the results.

```
nimbus <- read_csv("nimbus.csv")
```

```
nimbus
```



```
date,longitude,latitude,ozone
1985-10-01T00:00:00Z,-179.375,-73.5,302
1985-10-01T00:00:00Z,-178.125,-73.5,302
1985-10-01T00:00:00Z,-176.875,-73.5,302
1985-10-01T00:00:00Z,-175.625,-73.5,302
1985-10-01T00:00:00Z,-174.375,-73.5,304
1985-10-01T00:00:00Z,-173.125,-73.5,304
1985-10-01T00:00:00Z,-171.875,-73.5,304
1985-10-01T00:00:00Z,-170.625,-73.5,304
1985-10-01T00:00:00Z,-164.375,-73.5,287
1985-10-01T00:00:00Z,-163.125,-73.5,287
```

Viewing CSV files in RStudio

Parsing



Quiz

What class is ozone?

```
nimbus %>% pluck("ozone") %>% class()
```

```
nimbus %>% pluck("ozone") %>% class()
```

```
[1] "character"
```



```
nimbus %>% pluck("ozone") %>% unique()
```

```
[1] "302" "304" "287" "274" "264" "242" "211" "195" "197" "196" "198" "193" "187"  
[14] "190" "199" "194" "213" "218" "221" "229" "209" "186" "188" "191" "189" "184"  
[27] "180" "190" "215" "312" "319" "320" "311" "300" "290" "267" "226" "210" "200"  
[40] "203" "201" "192" "204" "206" "208" "205" "223" "232" "238" "243" "220" "202"  
[53] "185" "219" "222" "216" "324" "336" "333" "323" "308" "295" "244" "212" "237"  
[66] "248" "239" "241" "250" "249" "252" "234" "318" "313" "326" "335" "337" "316"  
[79] "266" "207" "227" "251" "253" "257" "261" "214" "228" "273" "285" "288" "291"  
[92] "270" "254" "317" "325" "332" "340" "344" "338" "297" "247" "217" "225" "231"  
[105] "235" "236" "262" "260" "265" "272" "278" "280" "279" "255" "245" "224" "181"  
[118] "240" "269" "296" "307" "315" "321" "306" "299" "298" "283" "327" "322" "328"  
[131] "331" "310" "275" "233" "258" "276" "281" "289" "330" "346" "305" "334" "359"  
[144] "347" "314" "301" "256" "263" "277" "284" "282" "271" "246" "183" "182" "230"  
[157] "349" "351" "350" "342" "329" "355" "371" "309" "303" "292" "259" "268" "341"  
[170] "343" "348" "345" "354" "361" "372" "382" "376" "356" "293" "286" "353" "351"  
[183] "358" "360" "363" "370" "384" "380" "294" "339" "362" "352" "368" "373" "377
```



. = NA

nimbus

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



read_csv()

readr functions share a common syntax

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

object to save
output into

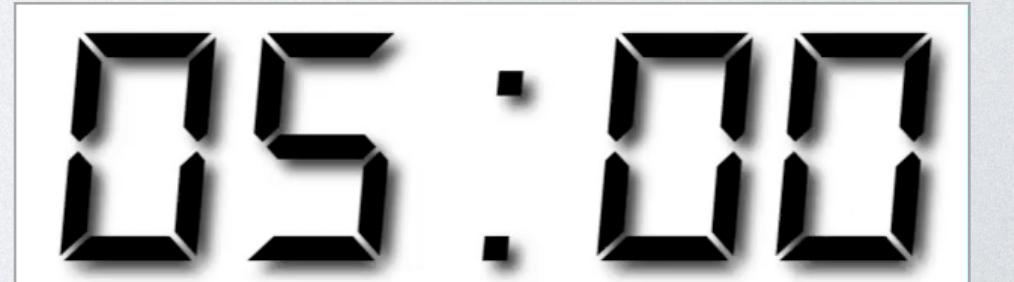
path from working
directory to file

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 = ".")  
nimbus %>%  
  filter(is.na(ozone)) %>%  
  summarize(n = n())  
##      n  
## 1 155
```

Quiz

What "type" of column is ozone?

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

date	longitude	latitude	ozone
<dtm>	<dbl>	<dbl>	<dbl>
1985-10-01 00:00:00	-179.	-73.5	302
1985-10-01 00:00:00	-178.	-73.5	302
1985-10-01 00:00:00	-177.	-73.5	302
1985-10-01 00:00:00	-176.	-73.5	302
1985-10-01 00:00:00	-174.	-73.5	302
1985-10-01 00:00:00	-173.	-73.5	302
1985-10-01 00:00:00	-172.	-73.5	304
1985-10-01 00:00:00	-171.	-73.5	304
1985-10-01 00:00:00	-164.	-73.5	287
1985-10-01 00:00:00	-163.	-73.5	287
... with 18,953 more rows			

<dbl> stands
for "double"



Suppose

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

date <small><S3: POSIXct></small>	longitude <small><dbl></small>	latitude <small><dbl></small>	ozone <small><chr></small>
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("nimbus.csv", na = ".",
  col_types = list(ozone = col_double()))
```

Manually
specify column
types.

list

column
name

Column type
function



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

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



Your Turn

Read in `nimbus.csv`. accounting for NA's and setting the `col_type` of ozone to a double. Then make this plot. What do you see?

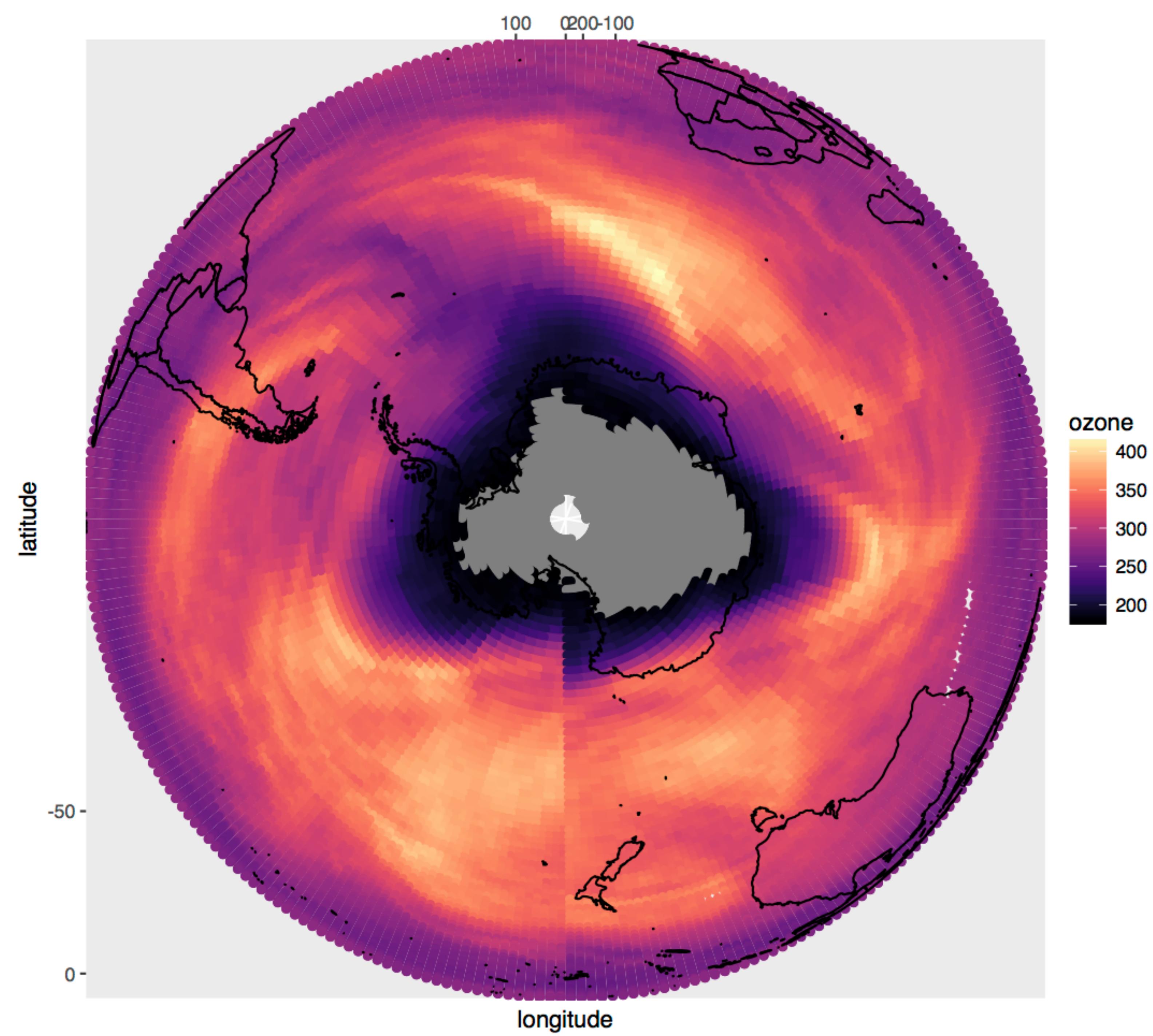
```
library(viridis)
world <- map_data(map = "world")
nimbus %>%
  ggplot() +
  geom_point(aes(longitude, latitude, color = ozone)) +
  geom_path(aes(long, lat, group = group), data = world) +
  coord_map("ortho", orientation=c(-90, 0, 0)) +
  scale_color_viridis(option = "A")
```



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

```
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")
```





Writing



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, file = "nimbus2.csv")
```

Table to save

file
path to save at



Reading in Excel Files



readxl



Reading in Excel Files

```
# install.packages("tidyverse")
library(readxl)
```



C37

Country	WHO region	Immunization coverage (%) among 1-year-olds ^a			Antenatal care coverage ^b (%)		Births attended by skilled health personnel (%)	
		Measles DTP3 HepB3			2003	2003	2003	2003
		2003	2003	2003	(%)	year	(%)	skilled health personnel (%)
Afghanistan	EMR	50	54	0	52	2003	52	14
Albania	EUR	93	97	97	81	2002	99	99
Algeria	AFR	84	87	0	79	2000	92	92
Andorra	EUR	96	99	84	x	x	x	x
Angola	AFR	62	46	0	x	x	x	45
Antigua and Barbuda	AMR	99	99	99	x	x	x	100
Argentina	AMR	97	88	0	x	x	x	99
Armenia	EUR	94	94	93	82	2000	97	97
Australia	WPR	93	92	95	x	x	x	100
Austria	EUR	79	84	44	x	x	x	...
Azerbaijan	EUR	98	97	98	70	2001	84	84
Bahamas	AMR	90	92	88	x	x	x	99
Bahrain	EMR	100	97	98	63	1995	98	98
Bangladesh	SEAR	77	85	0	39	2000	14	14
Barbados	AMR	90	86	91	89	2001	91	91
Belarus	EUR	99	86	99	x	x	x	100
Belgium	EUR	75	90	50	x	x	x	x
Belize	AMR	96	96	96	x	x	x	83
Benin	AFR	83	88	81	88	2001	66	66
Bhutan	SEAR	88	95	95	x	x	x	24
Bolivia	AMR	64	81	81	84	2001	65	65
Bosnia and Herzegovina	EUR	84	87	0	99	2000	100	100
Botswana	AFR	90	97	78	99	2001	94	94
Brazil	AMR	99	96	91	84	1996	88	88
Brunei Darussalam	WPR	99	99	99	x	x	x	99

who.xls

Country-Level Immunization Stats from the
World Health Organization

Quiz

Open up **who.xls** in Excel.

What problems might you encounter reading this data into R?

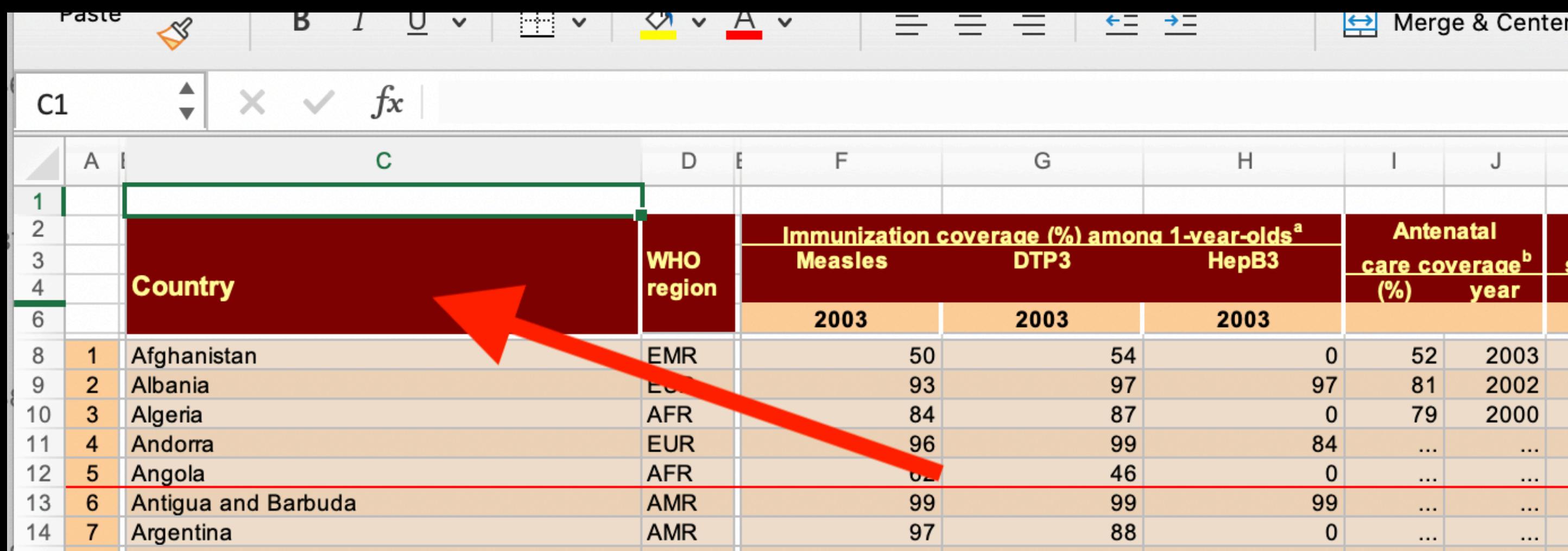
(Hint: Is this data "tidy")

Zambia	AFR	84	80	0
Zimbabwe	AFR	80	80	80
Region				
African Region	AFR	63	61	29
Region of the Americas	AMR	93	91	77
South-East Asia Region	SEAR	71	73	13
European Region	EUR	90	91	67
Eastern Mediterranean Region	EMR	75	77	44
Western Pacific Region	WPR	85	89	65
Figures computed by WHO to improve comparability where appropriate; they are not necessarily the official statistics.				
... data not available or not applicable.				
^a World Health Organization, Department of Immunization Vaccines and Biologicals, Vaccine Assessment and Monitoring System. (http://www.who.int/immunization_monitoring/vaccine)				
^b The World Health Report 2005: make every mother and child count. Geneva, World Health Organization, 2005. (http://www.who.int/whr)				
^c The WHO Global Roll Back Malaria database. (http://www.who.int/globalatlas/autologin/malaria_login.asp)				
^d WHO report 2005. Global Tuberculosis Control; Surveillance, Planning, Financing. Geneva, World Health Organization. (http://www.who.int/tb)				
^e The WHO Global Database on Child Growth and Malnutrition. (http://www.who.int/nutgrowthdb)				

Sheet != Table

2.7	50
x	50
x	45
x	23
x	28
x	50

2 NA values



A screenshot of Microsoft Excel showing a data table. The table has a header row (row 1) with various column titles. Row 2 contains numerical data. Row 3 contains country names. Row 4 is a multi-row column header where 'Country' spans two rows. Row 5 contains WHO region codes. Row 6 contains immunization coverage percentages for 2003. Row 7 contains antenatal care coverage percentages for 2003. Row 8 contains a year column. The table is styled with alternating colors for readability.

		C	D	E	F	G	H	I	J
1									
2									
3									
4		Country	WHO region		Immunization coverage (%) among 1-year-olds ^a		Antenatal care coverage ^b		
5					Measles	DTP3	HepB3		
6					2003	2003	2003		
7	1	Afghanistan	EMR		50	54	0	52	2003
8	2	Albania	EU		93	97	97	81	2002
9	3	Algeria	AFR		84	87	0	79	2000
10	4	Andorra	EUR		96	99	84
11	5	Angola	AFR		62	46	0
12	6	Antigua and Barbuda	AMR		99	99	99
13	7	Argentina	AMR		97	88	0

Multi-row columns

read_excel()

Read in an Excel File

```
df <- read_excel("who.xls", na = c("x", ""))
```

object to save
output into

path from working
directory to file

Value(s) to
convert to NA



read_excel()

Read in an Excel File

```
df <- read_excel("who.xls", na = c("x", ""), range="A1:B2")
```

Range of Cells to
read



Exercise: Reading in the Small Table

Read in the small table in **who.xls**.

Make it look like the image on the right

1. Set the **range** parameter set to read in just the second, smaller table.
2. Set the **NA** parameter appropriately.
3. Set **another option** ... so that the first row is not treated as a column name (see ?read_excel)

	...1	...2	...3	...4
1	African Region	AFR	NA	
2	Region of the Americas	AMR	NA	
3	South-East Asia Region	SEAR	NA	
4	European Region	EUR	NA	
5	Eastern Mediterranean Region	EMR	NA	
6	Western Pacific Region	WPR	NA	

Exercise: Reading in the Small Table

```
read_excel("who.xls",  
          na      = c("X", ""),  
          range   = "C204:U209",  
          col_names = FALSE)
```

	...1	...2	...3	...4
1	African Region	AFR	NA	
2	Region of the Americas	AMR	NA	
3	South-East Asia Region	SEAR	NA	
4	European Region	EUR	NA	
5	Eastern Mediterranean Region	EMR	NA	
6	Western Pacific Region	WPR	NA	

excel_sheets()

List sheets in an Excel file

```
excel_sheets("who.xls")
[1] "2.Health service coverage"
```

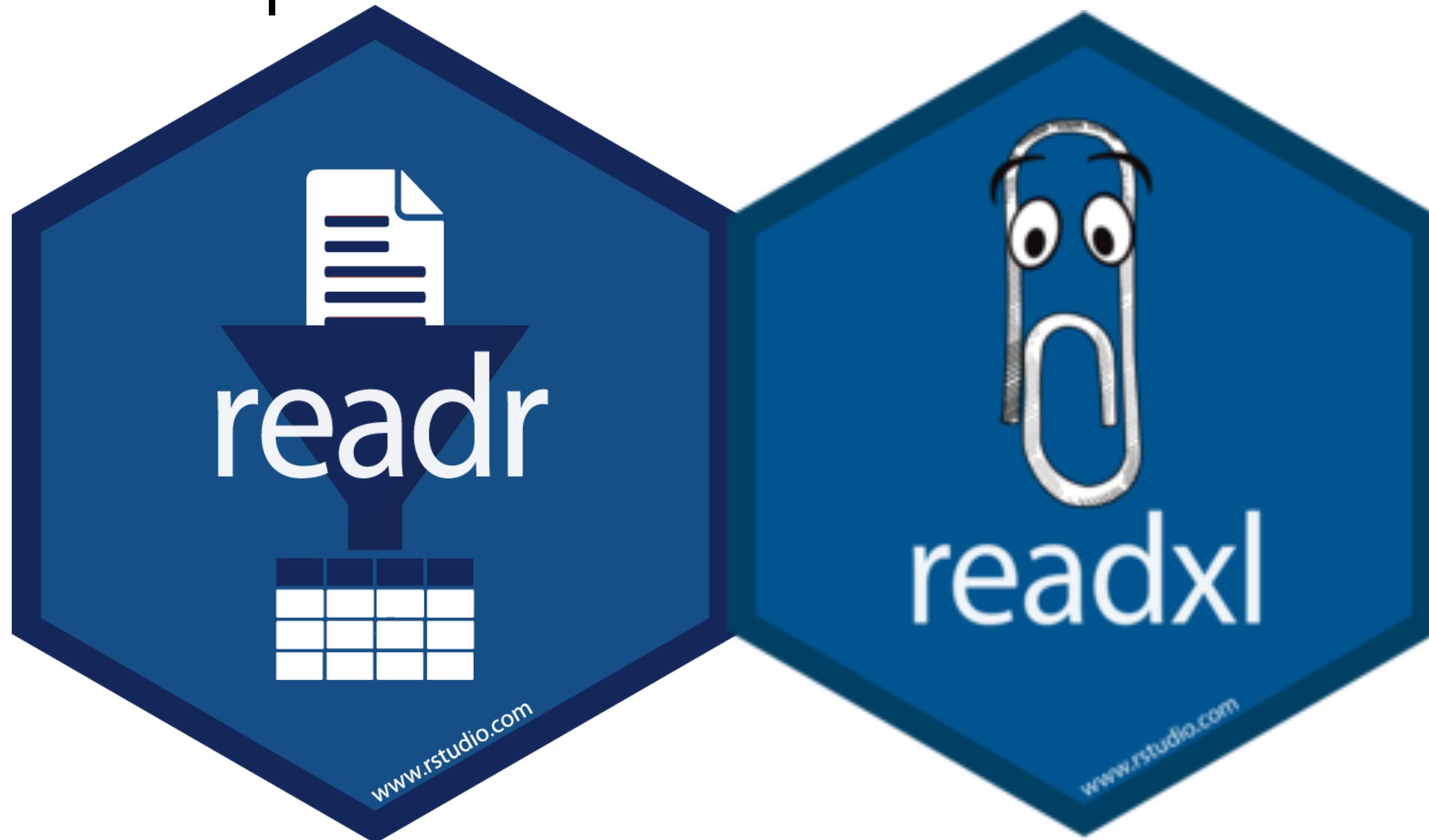
Data Import in the Tidyverse

A package for each storage type!

package	accesses
readr	csv, tsv, etc.
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



Import Data with



Data Import

Main Ideas	Notes

Notes form