Data basics

Richard Layton

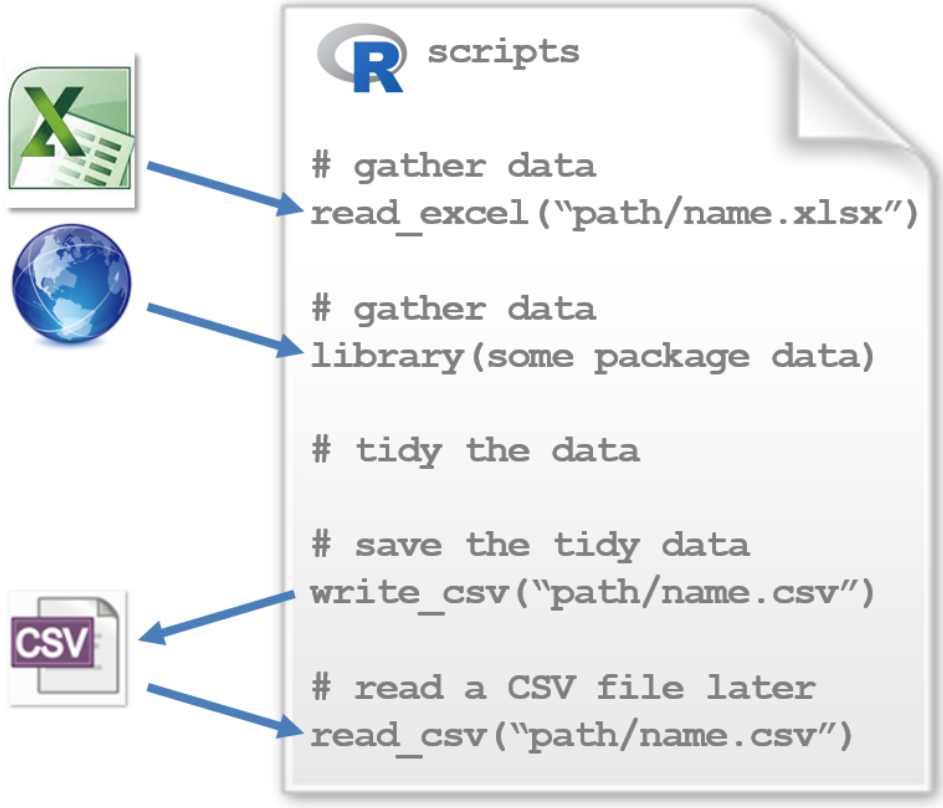
2017-09-05

R packages used in data basics

Package	For
tidyverse	ggplot2, dplyr, readr, tidyr
ggplot2	Graphing data
dplyr	Data manipulation
readr	Read rectangular data files (like csv or tsv)
tidyr	Set of functions that help you get to tidy data
knitr	Dynamic documents
readxl	Read .xls and .xslx files
VIM	Examine missing values in a data frame

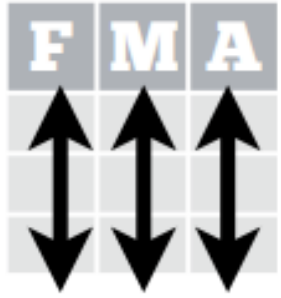
At the conclusion of the tutorial, you will be able to

- Define a "tidy" dataset
- Read data from an Excel file
- Obtain datasets included with R and R packages
- Read and write CSV data files



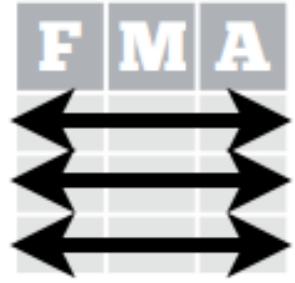
For data graphics success, data should be tidy

In a tidy data set:



Each **variable** is saved in its own **column**

&



Each **observation** is saved in its own **row**

Source: data-wrangling-cheatsheet, https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

Tidy data

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

table1

variables

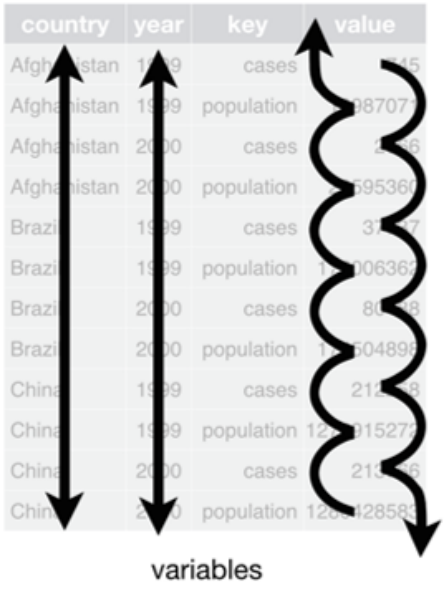
observations

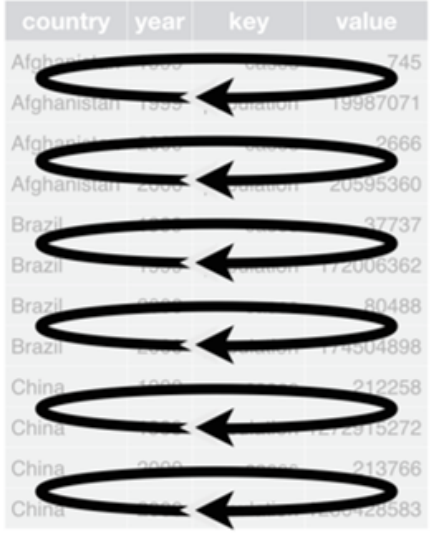
Source: Data Science with R by Garrett Grolemund, http://garrettgman.github.io/tidying/

Sadly, untidy data is common

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2





Sadly, untidy data is common

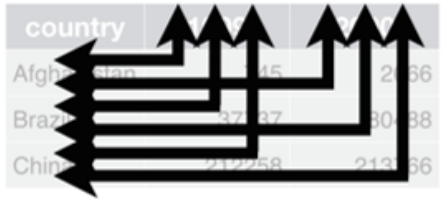
country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

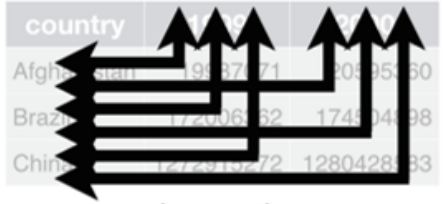
table4

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

table5

variables

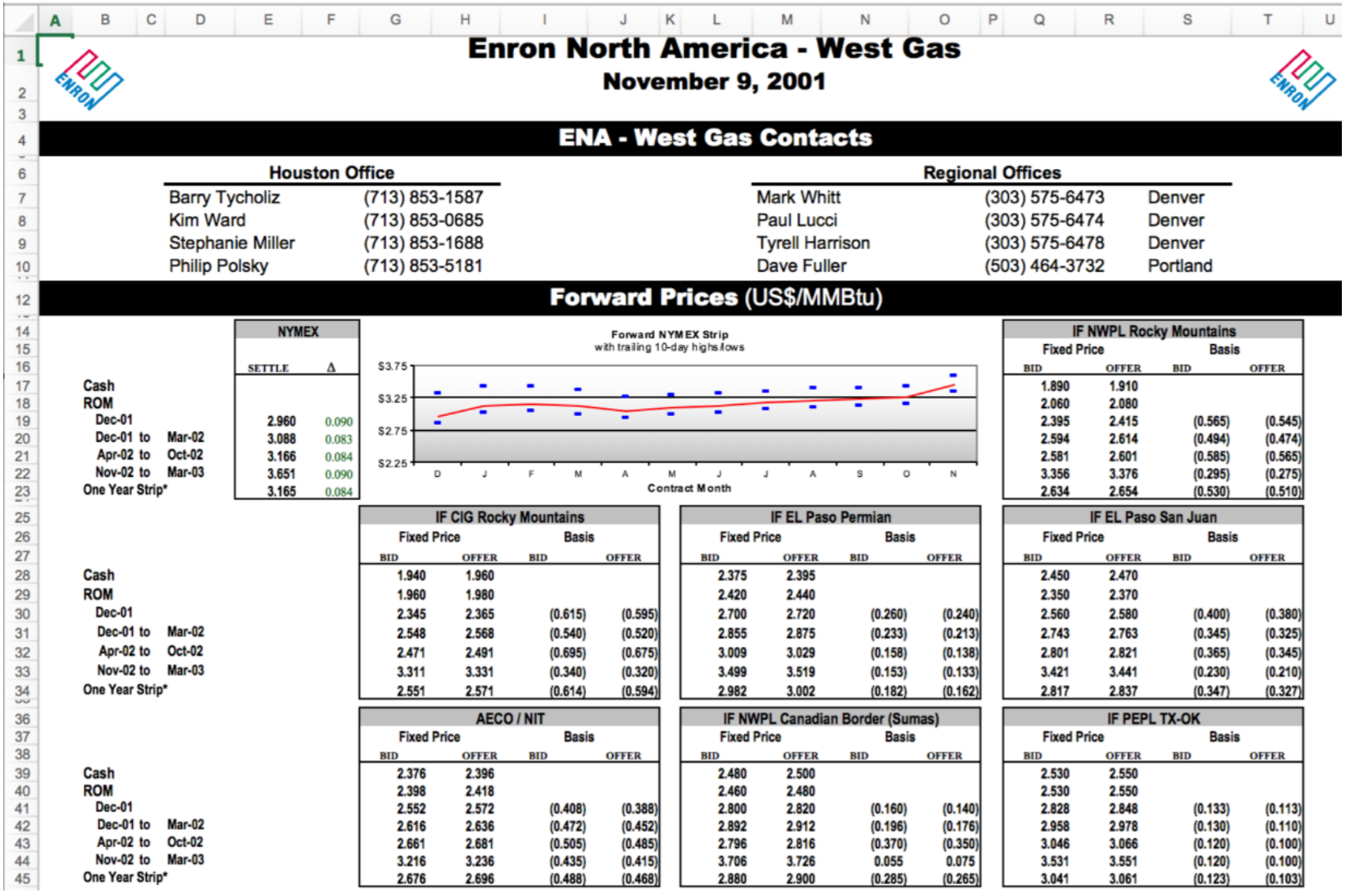




observations

Source: Data Science with R by Garrett Grolemund, http://garrettgman.github.io/tidying/

Some industry spreadsheets are really untidy



Enron North America - West Gas

November 9, 2001

ENA - West Gas Contacts

Houston Office		Regional Offices		
Barry Tycholiz	(713) 853-1587	Mark Whitt	(303) 575-6473	Denver
Kim Ward	(713) 853-0685	Paul Lucci	(303) 575-6474	Denver
Stephanie Miller	(713) 853-1688	Tyrell Harrison	(303) 575-6478	Denver
Philip Polsky	(713) 853-5181	Dave Fuller	(503) 464-3732	Portland

Forward Prices (US$/MMBtu)

	NYMEX		IF NWPL Rocky Mountains			
			Fixed Price		Basis	
	SETTLE	Δ	BID	OFFER	BID	OFFER
Cash			1.890	1.910		
ROM			2.060	2.080		
Dec-01	2.960	0.090	2.395	2.415	(0.565)	(0.545)
Dec-01 to Mar-02	3.088	0.083	2.594	2.614	(0.494)	(0.474)
Apr-02 to Oct-02	3.166	0.084	2.581	2.601	(0.585)	(0.565)
Nov-02 to Mar-03	3.651	0.090	3.356	3.376	(0.295)	(0.275)
One Year Strip*	3.165	0.084	2.634	2.654	(0.530)	(0.510)

	IF CIG Rocky Mountains				IF EL Paso Permian				IF EL Paso San Juan			
	Fixed Price		Basis		Fixed Price		Basis		Fixed Price		Basis	
	BID	OFFER	BID	OFFER	BID	OFFER	BID	OFFER	BID	OFFER	BID	OFFER
Cash	1.940	1.960			2.375	2.395			2.450	2.470		
ROM	1.960	1.980			2.420	2.440			2.350	2.370		
Dec-01	2.345	2.365	(0.615)	(0.595)	2.700	2.720	(0.260)	(0.240)	2.560	2.580	(0.400)	(0.380)
Dec-01 to Mar-02	2.548	2.568	(0.540)	(0.520)	2.855	2.875	(0.233)	(0.213)	2.743	2.763	(0.345)	(0.325)
Apr-02 to Oct-02	2.471	2.491	(0.695)	(0.675)	3.009	3.029	(0.158)	(0.138)	2.801	2.821	(0.365)	(0.345)
Nov-02 to Mar-03	3.311	3.331	(0.340)	(0.320)	3.499	3.519	(0.153)	(0.133)	3.421	3.441	(0.230)	(0.210)
One Year Strip*	2.551	2.571	(0.614)	(0.594)	2.982	3.002	(0.182)	(0.162)	2.817	2.837	(0.347)	(0.327)

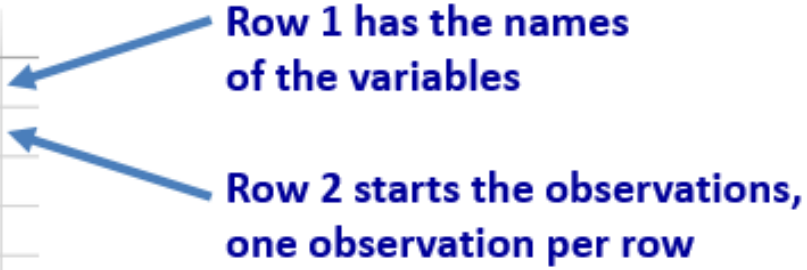
	AECO / NIT				IF NWPL Canadian Border (Sumas)				IF PEPL TX-OK			
	Fixed Price		Basis		Fixed Price		Basis		Fixed Price		Basis	
	BID	OFFER	BID	OFFER	BID	OFFER	BID	OFFER	BID	OFFER	BID	OFFER
Cash	2.376	2.396			2.480	2.500			2.530	2.550		
ROM	2.398	2.418			2.460	2.480			2.530	2.550		
Dec-01	2.552	2.572	(0.408)	(0.388)	2.800	2.820	(0.160)	(0.140)	2.828	2.848	(0.133)	(0.113)
Dec-01 to Mar-02	2.616	2.636	(0.472)	(0.452)	2.892	2.912	(0.196)	(0.176)	2.958	2.978	(0.130)	(0.110)
Apr-02 to Oct-02	2.661	2.681	(0.505)	(0.485)	2.796	2.816	(0.370)	(0.350)	3.046	3.066	(0.120)	(0.100)
Nov-02 to Mar-03	3.216	3.236	(0.435)	(0.415)	3.706	3.726	0.055	0.075	3.531	3.551	(0.120)	(0.100)
One Year Strip*	2.676	2.696	(0.488)	(0.468)	2.880	2.900	(0.285)	(0.265)	3.041	3.061	(0.123)	(0.103)

Source: Extract tables from messy spreadsheets with jailbreakr, http://blog.revolutionanalytics.com/2016/08/jailbreakr.html

For now, assume your data in Excel are tidy

DSR-table1.xlsx

	A	B	C	D
1	country	year	cases	population
2	Afghanistan	1999	745	19987071
3	Afghanistan	2000	2666	20595360
4	Brazil	1999	37737	172006362
5	Brazil	2000	80488	174504898
6	China	1999	212258	1272915272
7	China	2000	213766	1280428583



- *readxl* is the package
- *read_excel()* is the function

The data frame that results from *read_excel()*

```
library(readxl)
tidy_data <- read_excel(
        path = "data/DSR-table1.xlsx"
    , sheet = "DSR-table1"
    )
glimpse(tidy_data)
```

```
## Observations: 6
## Variables: 4
## $ country    <chr> "Afghanistan", "Afghanistan"...
## $ year       <dbl> 1999, 2000, 1999, 2000, 1999...
## $ cases      <dbl> 745, 2666, 37737, 80488, 212...
## $ population <dbl> 19987071, 20595360, 17200636...
```

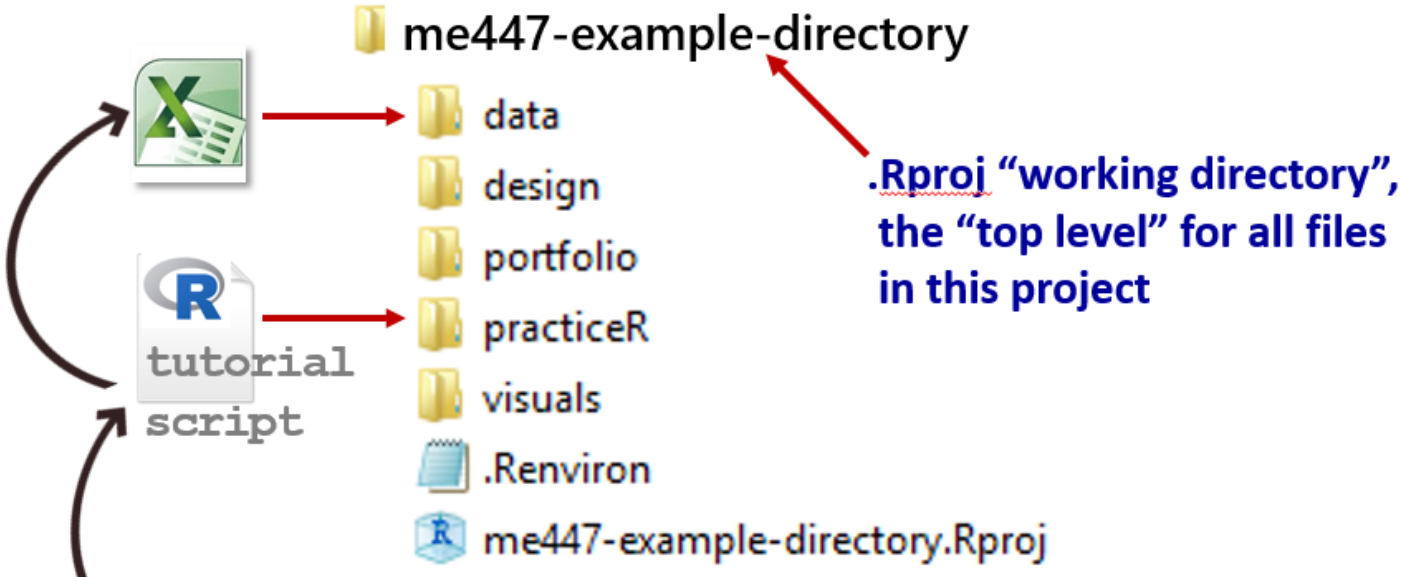
Or, if we print the data frame as a table using *knitr::kable()*

```
library(knitr)
kable(tidy_data)
```

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

Note the relative path argument in *read_excel()*

- Paths are relative to the current working directory
- In an R-project, the project folder is the working directory

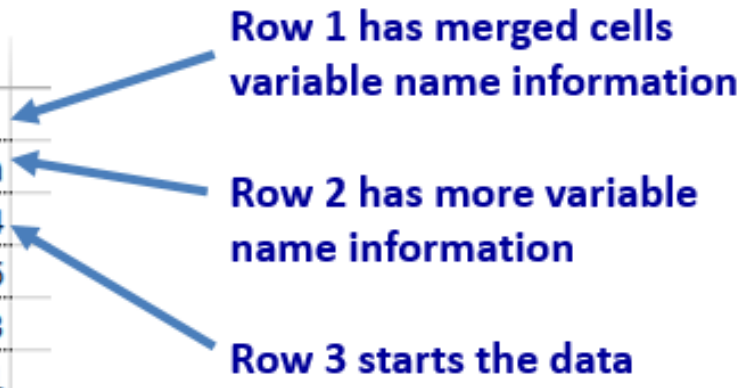



```
tidy_data <- read_excel(
        path = "data/DSR-table1.xlsx"
      , sheet = "DSR-table1"
      )
```

But, when your Excel data are not tidy

VADeaths.xlsx

	A	B	C	D	E
1		Rural		Urban	
2	Group	Men	Women	Men	Women
3	50-54	11.7	8.7	15.4	8.4
4	55-59	18.1	11.7	24.3	13.6
5	60-64	26.9	20.3	37	19.3
6	65-69	41	30.9	54.6	35.1
7	70-74	66	54.3	71.1	50



The *read_excel()* results can be weird

```
untidy_data <- read_excel("data/VADeaths.xlsx")
glimpse(untidy_data)
```

```
## Observations: 6
## Variables: 5
## $ X__1  <chr> "Group", "50-54", "55-59", "60-64...
## $ Rural <chr> "Men", "11.7", "18.1000000000000...
## $ X__2  <chr> "Women", "8.6999999999999993", "1...
## $ Urban <chr> "Men", "15.4", "24.3", "37", "54....
## $ X__3  <chr> "Women", "8.4", "13.6", "19.3", "...
```

- All the data have been coerced into character data

Or, if we print the result as a table

```
kable(untidy_data)
```

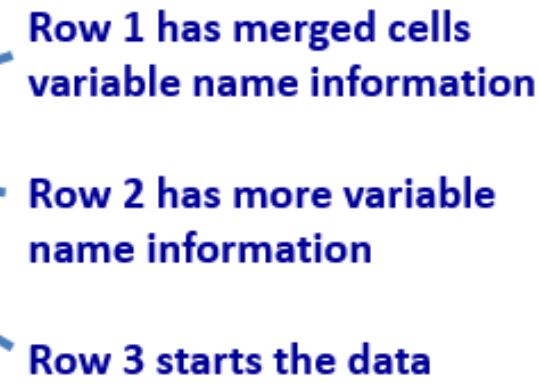
X___1	Rural	X___2	Urban
Group	Men	Women	Men
50-54	11.7	8.6999999999999993	15.4
55-59	18.100000000000001	11.7	24.3
60-64	26.9	20.3	37
65-69	41	30.9	54.6
70-74	66	54.3	71.09999999999

- The first data row is not actually an observation

However, *read_excel()* can skip lines

VADeaths.xlsx

	A	B	C	D	E
1		Rural		Urban	
2	Group	Men	Women	Men	Women
3	50-54	11.7	8.7	15.4	8.4
4	55-59	18.1	11.7	24.3	13.6
5	60-64	26.9	20.3	37	19.3
6	65-69	41	30.9	54.6	35.1
7	70-74	66	54.3	71.1	50




```
untidy_data <- read_excel("data/VADeaths.xlsx"
    , skip = 1)
```

But we might lose information

```
kable(untidy_data)
```

Group	Men	Women	Men___1	Women___1
50-54	11.7	8.7	15.4	8.4
55-59	18.1	11.7	24.3	13.6
60-64	26.9	20.3	37.0	19.3
65-69	41.0	30.9	54.6	35.1
70-74	66.0	54.3	71.1	50.0

- Try to work with tidy data only for the time being
- Later we will work on how to make untidy data tidy

Read and write CSV files

Write CSV. Given a data frame in your workspace

```
library(readr)
write_csv(tidy_data, "data/example_tidy_data.csv")
```

Read CSV. Given a CSV file in your data folder

```
new_copy <- read_csv("data/example_tidy_data.csv")
```

Graphing data and studying the data structure are complementary activities

Examine data frame characteristics

- *class()*
- *glimpse()*
- *kable()*
- *head()*
- *tail()*

Identifying missing values (details in the tutorial)

- *summary()*
- *VIM* package *aggr()* function

Accessing data in R

```
data()
```

```
Data sets in package 'datasets':

AirPassengers  Monthly Airline Passenger Numbers 1949-1960
BJsales        Sales Data with Leading Indicator
BOD            Biochemical Oxygen Demand
CO2            Carbon Dioxide Uptake in Grass Plants
ChickWeight    Weight vs age of chicks on different diets
etc.
```

Many of these R data sets widely used in R examples

Commonly encountered

- `mtcars`
- `iris`
- `Titanic`
- `barley` in the *lattice* package

From the tidyverse

- `starwars` in the *dplyr* package
- `economics` in the *ggplot2* package
- `who` in the *tidyr* package

And other packages

- *fivethirtyeight* package
- so many more

Review

package::function()

- readxl::read_excel()
- readr::write_csv()
- readr::read_csv()
- tibble::glimpse()
- knitr::kable()
- VIM::aggr()

base R functions

- data()
- class()
- head()
- tail()
- summary()
- unloadNamespace()
- search()

ggplot2

- geom_smooth(method = lm, se = FALSE)