

Data Wrangling with R: Day 1

Tidying data with **tidyverse**

Presented by Emi Tanaka

Department of Econometrics and Business Statistics



MONASH University

1st December 2020 @ Statistical Society of Australia | Zoom

Recall tidy data



Definition of a tidy data

- Each variable must have its own column
- Each observation must have its own row
- Each value must have its own cell

country	date	cases	deaths
USA	20/1/24	148,65	8,7
India	20/1/24	37,29	4,8
Brazil	20/1/24	16,603	3,4
France	20/1/24	4,52	5,0
Russia	20/1/24	25,73	3,1

variables

country	date	cases	deaths
USA	20/1/24	148,65	8,7
India	20/1/24	37,29	4,8
Brazil	20/1/24	16,603	3,4
France	20/1/24	4,52	5,0
Russia	20/1/24	25,73	3,1

observations

country	date	cases	deaths
USA	20/1/24	148,65	8,7
India	20/1/24	37,29	4,8
Brazil	20/1/24	16,603	3,4
France	20/1/24	4,52	5,0
Russia	20/1/24	25,73	3,1

values

Is this tidy data?

Part 1

Estimated residential population in December

state	2019	2018	2017
NSW	8130159	80366651	7919815
VIC	6655284	6528601	6387081
ACT	427892	423169	415874

Estimated residential population in December

state	year	population
NSW	2019	8130159
NSW	2018	80366651
NSW	2017	7919815
VIC	2019	6655284
VIC	2018	6528601
VIC	2017	6387081
ACT	2019	427892
ACT	2018	423169
ACT	2017	415874

Values adapted from Australian Bureau of Statistics. (2020). Table 04. Estimated Resident Population, States and Territories [Time series spreadsheet]. National, state and territory population, Australia Mar 2020. Retrieved Nov 24, 2020.
<https://www.abs.gov.au/statistics/people/population/national-state-and-territory-population/mar-2020/310104.xls>

Is this tidy data?

Part 2

Total weekly personal income by age and sex from ABS 2016 Census

33 What is the total of all income the person usually receives?

- Mark one box only.
- Do not deduct tax, superannuation contributions, amounts salary sacrificed, or any other automatic deductions.
- Include:
 - Wages and salaries
 - Regular overtime
 - Commissions and bonuses
 - Government pensions, benefits and allowances
 - Age pension
 - Family tax benefit
 - Parenting payment
 - Disability support pension
 - Newstart allowance
 - Youth and student allowances
 - Career allowance
 - Any other government pension/allowance
 - Profit or loss from
 - Unincorporated business/farm (e.g. sole traders, partnerships)
 - Rental properties
 - Other income
 - Income from superannuation
 - Private pensions
 - Child support
 - Interest
 - Dividends from shares
 - Workers' compensation
 - Any other income

Information from this question provides an indication of living standards in different areas.

[Go to census.abs.gov.au for more information.](http://census.abs.gov.au)

\$3,000 or more per week \$156,000 or more per year	109473	33797	253
\$2,000 - \$2,999 per week \$104,000 - \$155,999 per year	88338	31685	213
\$1,750 - \$1,999 per week \$91,000 - \$103,999 per year	63756	15815	8
\$1,500 - \$1,749 per week \$78,000 - \$90,999 per year	24566	7007	7
\$1,250 - \$1,499 per week \$65,000 - \$77,999 per year	36101	9806	5
\$1,000 - \$1,249 per week \$52,000 - \$64,999 per year	7040	1320	2
\$800 - \$999 per week \$41,600 - \$51,999 per year	2669	550	1
\$650 - \$799 per week \$33,800 - \$41,599 per year	5515	2251	Queensland
\$500 - \$649 per week \$26,000 - \$33,799 per year			South Australia
\$400 - \$499 per week \$20,800 - \$25,999 per year			Western Australia
\$300 - \$399 per week \$15,600 - \$20,799 per year			Tasmania
\$150 - \$299 per week \$7,800 - \$15,599 per year			Northern Territory
\$1 - \$149 per week \$1 - \$7,799 per year			Australian Capital Territory
Nil income			Other Territories
Negative income			

5_19_yrs	M_Neg_Nil_income_20_24_yrs	M_Neg_Nil_income_25_34_yrs	STE Code
109473	33797	253	
88338	31685	213	
63756	15815	8	
24566	7007	7	
36101	9806	5	
7040	1320	2	
2669	550	1	
5515	2251	Queensland	
		South Australia	
		Western Australia	
		Tasmania	
		Northern Territory	
		Australian Capital Territory	
		Other Territories	

Is this tidy data?

Part 3



Total weekly personal income by age and sex

state	group	count
New South Wales	M_Neg_Nil_income_15_19_yrs	109473
New South Wales	M_Neg_Nil_income_20_24_yrs	33797
New South Wales	M_Neg_Nil_income_25_34_yrs	25398
New South Wales	M_Neg_Nil_income_35_44_yrs	14472
New South Wales	M_Neg_Nil_income_45_54_yrs	15235
New South Wales	M_Neg_Nil_income_55_64_yrs	22736
New South Wales	M_Neg_Nil_income_65_74_yrs	15465
New South Wales	M_Neg_Nil_income_75_84_yrs	6576
New South Wales	M_Negtve_Nil_incme_85_yrs_ovr	2320
New South Wales	M_Neg_Nil_income_Tot	245480

Is this tidy data?

Part 4



Total weekly personal income by age and sex

state	sex	income_min	income_max	age_min	age_max	count
New South Wales	M	-Inf	0	15	19	109473
New South Wales	M	-Inf	0	20	24	33797
New South Wales	M	-Inf	0	25	34	25398
New South Wales	M	-Inf	0	35	44	14472
New South Wales	M	-Inf	0	45	54	15235
New South Wales	M	-Inf	0	55	64	22736
New South Wales	M	-Inf	0	65	74	15465
New South Wales	M	-Inf	0	75	84	6576
New South Wales	M	-Inf	0	85	Inf	2320
New South Wales	M	1	149	15	19	41865

Is this tidy data?

Part 5

U.S. historical crop yields by state

year	state	crop	yield
1900	Iowa	barley	28.5
1900	Kansas	barley	18.0
2000	Kansas	barley	35.0
1900	Iowa	wheat	14.4
1900	Kansas	wheat	18.2
2000	Iowa	wheat	47.0
2000	Kansas	wheat	37.0

U.S. historical crop yields by state

year	state	barley_yield	wheat_yield
1900	Iowa	28.5	14.4
1900	Kansas	18.0	18.2
2000	Kansas	35.0	37.0
2000	Iowa	NA	47.0

Is this tidy data?

Part 6

U.S. historical crop yields by state

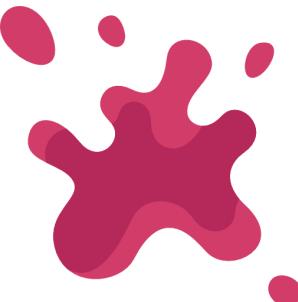
year	state	crop	metric	value
1900	Iowa	barley	yield	2.85e+01
1900	Iowa	barley	acres	6.20e+05
1900	Kansas	barley	yield	1.80e+01
1900	Kansas	barley	acres	1.27e+05
2000	Kansas	barley	yield	3.50e+01
2000	Kansas	barley	acres	7.00e+03
1900	Iowa	wheat	yield	1.44e+01
1900	Iowa	wheat	acres	1.45e+06
1900	Kansas	wheat	yield	1.82e+01
1900	Kansas	wheat	acres	4.29e+06

U.S. historical crop yields by state

year	state	barley_yield	wheat_yield	barley_acres	wheat_acres
1900	Iowa	28.5	14.4	620000	1450000
1900	Kansas	18.0	18.2	127000	4290000
2000	Kansas	35.0	37.0	7000	9400000
2000	Iowa	NA	47.0	NA	18000

year	state	crop	yield	acres
1900	Iowa	barley	28.5	620000
1900	Kansas	barley	18.0	127000
2000	Kansas	barley	35.0	7000
1900	Iowa	wheat	14.4	1450000
1900	Kansas	wheat	18.2	4290000
2000	Iowa	wheat	47.0	18000
2000	Kansas	wheat	37.0	9400000

Some signs of non-tidy data



- 1 Header has meaningful information
- 2 Cells have multiple values

Wide and long data formats

wide			
id	x	y	z
1	a	c	e
2	b	d	f

long		
id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

- Transforming data from wide to long or vice versa is referred to as **pivoting**
- Sometimes the long form is referred to as **molten data**
- Some people may refer to transforming data from wide to long as **melting** the data and vice-versa as **casting**

Evolving language

- Earlier efforts to transform data from wide to long was in the `reshape`, first released on CRAN in 2005-08-05
- It was then superseded by `reshape2` released on CRAN in 2010-09-10
- Then finally `tidyverse` released on CRAN in 2014-07-21 *v1.0.0 released 2019-09-12

Wide to long

- `reshape::melt` lifecycle retired
- `reshape2::melt` lifecycle retired
- `tidyverse::gather` lifecycle retired
- **`tidyverse::pivot_longer`***

Long to wide

- `reshape::cast` lifecycle retired
- `reshape2::dcast` lifecycle retired
- `tidyverse::spread` lifecycle retired
- **`tidyverse::pivot_wider`***

Pivoting data with `tidyverse`

Part 1

df_wide				
state	2019	2018	2017	
NSW	8130159	80366651	7919815	
VIC	6655284	6528601	6387081	
ACT	427892	423169	415874	

```
pivot_longer(df_wide, cols = `2019`:`2017`,  
             names_to = "year",  
             values_to = "population")
```

```
pivot_wider(df_long, id_cols = state,  
             names_from = year,  
             values_from = population)
```

df_long		
state	year	population
NSW	2019	8130159
NSW	2018	80366651
NSW	2017	7919815
VIC	2019	6655284
VIC	2018	6528601
VIC	2017	6387081
ACT	2019	427892
ACT	2018	423169
ACT	2017	415874

Pivoting data with `tidyverse`

Part 2

▶ table

```
data(census_2016_G17, package = "dwexercise")
```

wide format

```
as_tibble(census_2016_G17)

## # A tibble: 9 x 481
##   STE_CODE_2016 M_Neg_Nil_incom... M_Ne...
##   <int>          <int>
## 1 1              109473
## 2 2              88338
## 3 3              63756
## 4 4              24566
## 5 5              36101
## 6 6              7040
## 7 7              2669
## 8 8              5515
```

long format

```
pivot_longer(census_2016_G17,
              cols = -STE_CODE_2016,
              names_to = "group",
              values_to = "count")

## # A tibble: 4,320 x 3
##   STE_CODE_2016 group      count
##   <int> <chr>     <int>
## 1 1 M_Neg_Nil_income_1 109473
## 2 1 M_Neg_Nil_income_2 88338
## 3 1 M_Neg_Nil_income_2 63756
## 4 1 M_Neg_Nil_income_2 24566
## 5 1 M_Neg_Nil_income_2 36101
## 6 1 M_Neg_Nil_income_2 7040
## 7 1 M_Neg_Nil_income_2 2669
## 8 1 M_Neg_Nil_income_2 5515
```

Pivoting data with `tidyverse`

Part 3

yield_long

year	state	crop	yield
1900	Iowa	barley	28.5
1900	Kansas	barley	18.0
2000	Kansas	barley	35.0
1900	Iowa	wheat	14.4
1900	Kansas	wheat	18.2
2000	Iowa	wheat	47.0
2000	Kansas	wheat	37.0

yield_wide

year	state	barley_yield	wheat_yield
1900	Iowa	28.5	14.4
1900	Kansas	18.0	18.2
2000	Kansas	35.0	37.0
2000	Iowa	NA	47.0

yield_long → yield_wide

```
pivot_wider(yield_long,  
            id_cols = c(year, state),  
            names_from = crop,  
            values_from = yield,  
            names_glue = "{crop}_yield")
```

Pivoting data with `tidyverse`

Part 4

yield_long			
year	state	crop	yield
1900	Iowa	barley	28.5
1900	Kansas	barley	18.0
2000	Kansas	barley	35.0
1900	Iowa	wheat	14.4
1900	Kansas	wheat	18.2
2000	Iowa	wheat	47.0
2000	Kansas	wheat	37.0

yield_wide			
year	state	barley_yield	wheat_yield
1900	Iowa	28.5	14.4
1900	Kansas	18.0	18.2
2000	Kansas	35.0	37.0
2000	Iowa	NA	47.0

yield_wide → yield_long

```
pivot_longer(yield_wide,  
             cols = contains("yield"),  
             names_to = "crop",  
             names_pattern = "(.+)_yield",  
             values_to = "yield",  
             values_drop_na = TRUE)
```

Pivoting data with `tidyverse`

Part 5

`crop_long`

year	state	crop	metric	value
1900	Iowa	barley	yield	2.85e+01
1900	Iowa	barley	acres	6.20e+05
1900	Kansas	barley	yield	1.80e+01
1900	Kansas	barley	acres	1.27e+05
2000	Kansas	barley	yield	3.50e+01
2000	Kansas	barley	acres	7.00e+03
1900	Iowa	wheat	yield	1.44e+01
1900	Iowa	wheat	acres	1.45e+06
1900	Kansas	wheat	yield	1.82e+01
1900	Kansas	wheat	acres	4.29e+06

`crop_wide`

year	state	barley_yield	wheat_yield	barley_acres	wheat_acres
1900	Iowa	28.5	14.4	620000	1450000
1900	Kansas	18.0	18.2	127000	4290000
2000	Kansas	35.0	37.0	7000	9400000
2000	Iowa	NA	47.0	NA	18000

`crop_long` → `crop_wide`

```
pivot_wider(crop_long,  
            names_from = c(crop, metric),  
            values_from = value,  
            names_glue = "{crop}_{metric}")
```

Pivoting data with `tidyverse`

Part 6

crop_long				
year	state	crop	metric	value
1900	Iowa	barley	yield	2.85e+01
1900	Iowa	barley	acres	6.20e+05
1900	Kansas	barley	yield	1.80e+01
1900	Kansas	barley	acres	1.27e+05
2000	Kansas	barley	yield	3.50e+01
2000	Kansas	barley	acres	7.00e+03
1900	Iowa	wheat	yield	1.44e+01
1900	Iowa	wheat	acres	1.45e+06
1900	Kansas	wheat	yield	1.82e+01
1900	Kansas	wheat	acres	4.29e+06

crop_wide					
year	state	barley_yield	wheat_yield	barley_acres	wheat_acres
1900	Iowa	28.5	14.4	620000	1450000
1900	Kansas	18.0	18.2	127000	4290000
2000	Kansas	35.0	37.0	7000	9400000
2000	Iowa	NA	47.0	NA	18000

`crop_wide → crop_long`

```
pivot_longer(crop_wide,  
             names_to = c("crop", "metric"),  
             names_pattern = "(.+)_(.+)",  
             values_to = "value")
```

Pivoting data with `tidyverse`

Part 7

▶ table

STE_CODE_2016	M_Neg_Nil_income_15_19_yrs	M_Neg_Nil_income_20_24_yrs	M_Neg_Nil_income_25_34_yrs
1	109473	33797	253

```
census_2016_G17 %>%  
  pivot_longer(cols = -STE_CODE_2016,  
               names_to = c("sex", "income_min", "income_max", "age_min", "age_max"),  
               names_pattern = "^(F|M)_([\\d]+|Neg_Nil|Negtve_Nil|PI)_([\\d]+|more|inf|na)",  
               values_to = "count",  
               names_transform = list(  
                 income_min = function(x) case_when(str_detect(x, "Neg") ~ -Inf,  
                                              TRUE ~ as.numeric(x)),  
                 income_max = function(x) case_when(x %in% c("income", "incme") ~ Inf,  
                                              x %in% "more" ~ Inf,  
                                              TRUE ~ as.numeric(x)),  
                 age_min = as.numeric,  
                 age_max = function(x) ifelse(x=="", Inf, as.numeric(x)))) %>%
```

Extract values into multiple columns

Alternatively, we could pivot all column names except STE_CODE_2016 to a single column as in  and then extract the values from cells

```
census_2016_G17 %>%  
  pivot_longer(cols = -STE_CODE_2016,  
               names_to = "group",  
               values_to = "value") %>%  
  filter(!str_detect(group, "Tot"),  
         !str_starts(group, "P")) %>%  
  extract(group,  
          into = c("sex", "income_min", "income_max", "age_min", "age_max"),  
          regex = "^(F|M)_(\d+|Neg_Nil|Negtve_Nil|PI)_(\d+|more|income|incme|N  
          remove = TRUE) %>%  
  mutate(income_min = case_when(str_detect(income_min, "Neg") ~ -Inf,  
                                TRUE ~ as.numeric(income_min),  
                                income_max = case_when(income_max %in% c("income", "incme") ~ 0,
```

Separate values into columns

pkg_dat

package	maintainer
dplyr	Hadley Wickham
magrittr	Lionel Henry
rlang	Lionel Henry
RVerbalExpressions	Tyler Littlefield
stringr	Hadley Wickham
tibble	Kirill Müller
tidyr	Hadley Wickham
tidyselect	Lionel Henry

🎯 separate maintainer name
to columns, first name and
last name

```
pkg_dat %>%  
  separate(maintainer,  
           into = c("first_name", "last_name"),  
           sep = " ")  
  
## # A tibble: 8 x 3  
##   package     first_name last_name  
##   <chr>       <chr>      <chr>  
## 1 dplyr       Hadley      Wickham  
## 2 magrittr    Lionel      Henry  
## 3 rlang        Lionel      Henry  
## 4 RVerbalExpressions Tyler      Littlefield  
## 5 stringr     Hadley      Wickham  
## 6 tibble       Kirill      Müller  
## 7 tidyr        Hadley      Wickham  
## 8 tidyselect   Lionel      Henry
```

Separate values into rows

author_dat	
package	author
dplyr	Hadley Wickham, Romain François, Lionel Henry, Kirill Müller
magrittr	Lionel Henry, Stefan Milton Bache, Hadley Wickham
rlang	Lionel Henry, Hadley Wickham
RVerbalExpressions	Tyler Littlefield
stringr	Hadley Wickham
tibble	Kirill Müller, Hadley Wickham
tidyr	Hadley Wickham
tidyselect	Lionel Henry, Hadley Wickham

```
author_dat %>%  
  separate_rows(author, sep = ", ")  
  
## # A tibble: 16 x 2  
##   package     author  
##   <chr>       <chr>  
## 1 dplyr      Hadley Wickham  
## 2 dplyr      Romain François  
## 3 dplyr      Lionel Henry  
## 4 dplyr      Kirill Müller  
## 5 magrittr   Lionel Henry  
## 6 magrittr   Stefan Milton Bache  
## 7 magrittr   Hadley Wickham  
## 8 rlang      Lionel Henry  
## 9 rlang      Hadley Wickham  
## 10 RVerbalExpressions Tyler Littlefield  
## 11 stringr   Hadley Wickham  
## 12 tibble    Kirill Müller  
## 13 tibble    Hadley Wickham  
## 14 tidyr     Hadley Wickham  
## 15 tidyselect Lionel Henry  
## 16 tidyselect Hadley Wickham
```

**</> If you installed the `dwexercise` package,
run below in your R console**

```
learnr::run_tutorial("day1-exercise-04", package = "dwexercise")
```

 **If the above doesn't work for you, go [here](#).**
 **Questions or issues, let us know!**

15:00

Session Information

```
devtools::session_info()
```

```
## - Session info -----
##   setting  value
##   version  R version 4.0.1 (2020-06-06)
##   os        macOS Catalina 10.15.7
##   system   x86_64, darwin17.0
##   ui        RStudio
##   language (EN)
##   collate  en_AU.UTF-8
##   ctype    en_AU.UTF-8
##   tz       Australia/Melbourne
##   date     2020-11-26
##
## - Packages -----
##   package * version  date      lib source
##   agridat  * 1.17    2020-08-03 [1] CRAN (R 4.0.2)
##   anicon    0.1.0    2020-06-21 [1] Github (emitanaka/anicon@0b756df)
```

These slides are licensed under

