Bio 724: Data wrangling

Paul Magwene and Greg Wray

Real-world data is often messy

Data files you generate or will be given may...

- Be poorly organized
- Have missing values
- Contain extraneous information
- Lack headers (variable names)
- Confound variables and labels
- Use different encoding schemes
- Use unfamiliar conventions for dates, decimal separators, etc.
- Include empty columns – used for visual organization in spreadsheet, but interferes with analysis
- Include meta data and comments

Tidy data

To facilitate downstream analyses, data should be organized in a manner such that...

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.

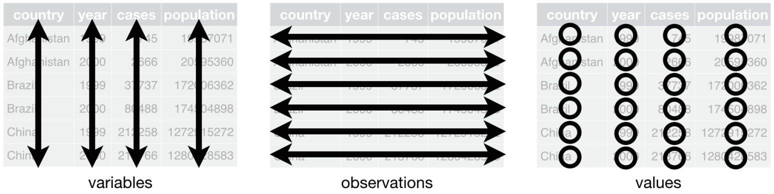


Figure 1: Visual representation of tidy data (from R4DS2e).

dplyr and tidyr to the rescue

The tidyverse packages `dplyr` and `tidyr` provide many useful tools for wrangling data into a tidy form.

dplyr functions that facilitate wrangling

- `select`
- `filter`
- `mutate`
- `rename`
- `arrange`

dplyr functions introduced in this lecture

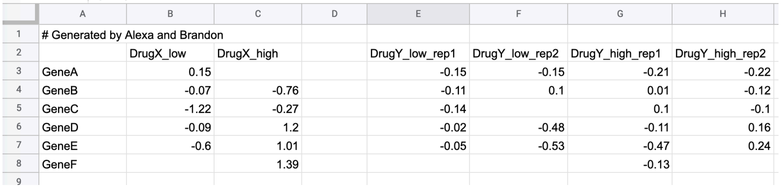
- `left_join`
- `full_join`
- `inner_join`

key functions introduced by tidyr

- `pivot_longer` - reshape column data into rows
- `pivot_wider` - reshape row data into columns
- `separate_wider_delim`, `separate_wider_position`, `separate_wider_regex` - turns a single column into multiple columns
- `unite` - turns multiple columns in a single column

Example: Starting messy data

```
# Generated by Alexa and Brandon
,DrugX_low,DrugX_high,,DrugY_low_rep1,DrugY_low_rep2,DrugY_high_rep1,DrugY_high_rep2
GeneA,0.15,,,-0.15,-0.15,-0.21,-0.22
GeneB,-0.07,-0.76,,-0.11,0.1,0.01,-0.12
GeneC,-1.22,-0.27,,-0.14,,0.1,-0.1
GeneD,-0.09,1.2,,-0.02,-0.48,-0.11,0.16
GeneE,-0.6,1.01,,-0.05,-0.53,-0.47,0.24
GeneF,,1.39,,,,-0.13,
```



	A	B	C	D	E	F	G	H
1	# Generated by Alexa and Brandon							
2		DrugX_low	DrugX_high		DrugY_low_rep1	DrugY_low_rep2	DrugY_high_rep1	DrugY_high_rep2
3	GeneA	0.15			-0.15	-0.15	-0.21	-0.22
4	GeneB	-0.07	-0.76		-0.11	0.1	0.01	-0.12
5	GeneC	-1.22	-0.27		-0.14		0.1	-0.1
6	GeneD	-0.09	1.2		-0.02	-0.48	-0.11	0.16
7	GeneE	-0.6	1.01		-0.05	-0.53	-0.47	0.24
8	GeneF		1.39				-0.13	
9								

Figure 2: In what ways is this data non-tidy?

Naively reading the data produces poor results

```r
messy <- read_csv("~/Downloads/small-messy-data.csv")
```

Generated by Alexa and Brandon
,DrugX_low,DrugX_high,,DrugY_low_rep1,DrugY_low_rep2,DrugY_high_rep1,DrugY_high_rep2
GeneA,0.15,, ,-0.15,-0.15,-0.21,-0.22
GeneB,-0.07,-0.76,,-0.11,0.1,0.01,-0.12
GeneC,-1.22,-0.27,,-0.14,,0.1,-0.1
GeneD,-0.09,1.2,,-0.02,-0.48,-0.11,0.16
GeneE,-0.6,1.01,,-0.05,-0.53,-0.47,0.24
GeneF
,1.39
,
,-0.13

Explore options of your reader function(s)

Example: filtering comment lines

```
messy <- read_csv("~/Downloads/small-messy-data.csv", comment="#")
```

...1	DrugX_low	DrugX_high	...4	DrugY_low_rep1	DrugY_low_rep2	DrugY_high_rep1	DrugY_high_rep2
GeneA	0.15	NA	NA	-0.15	-0.15	-0.21	-0.22
GeneB	-0.07	-0.76	NA	-0.11	0.10	0.01	-0.12
GeneC	-1.22	-0.27	NA	-0.14	NA	0.10	-0.10
GeneD	-0.09	1.20	NA	-0.02	-0.48	-0.11	0.16
GeneE	-0.60	1.01	NA	-0.05	-0.53	-0.47	0.24
GeneF	NA	1.39	NA	NA	NA	-0.13	NA

Renaming columns using dplyr::rename

```r
messy <-
  messy |>
  rename(Gene = "...1")
```

Gene	DrugX_low	DrugX_high	...4	DrugY_low_rep1	DrugY_low_rep2	DrugY_high_rep1	DrugY_high_rep2
GeneA	0.15	NA	NA	-0.15	-0.15	-0.21	-0.22
GeneB	-0.07	-0.76	NA	-0.11	0.10	0.01	-0.12
GeneC	-1.22	-0.27	NA	-0.14	NA	0.10	-0.10
GeneD	-0.09	1.20	NA	-0.02	-0.48	-0.11	0.16
GeneE	-0.60	1.01	NA	-0.05	-0.53	-0.47	0.24
GeneF	NA	1.39	NA	NA	NA	-0.13	NA

Dropping columns using select

```r
messy |>
  select(-"...4")
```

Gene	DrugX_low	DrugX_high	DrugY_low_rep1	DrugY_low_rep2	DrugY_high_rep1	DrugY_high_rep2
GeneA	0.15	NA	-0.15	-0.15	-0.21	-0.22
GeneB	-0.07	-0.76	-0.11	0.10	0.01	-0.12
GeneC	-1.22	-0.27	-0.14	NA	0.10	-0.10
GeneD	-0.09	1.20	-0.02	-0.48	-0.11	0.16
GeneE	-0.60	1.01	-0.05	-0.53	-0.47	0.24
GeneF	NA	1.39	NA	NA	-0.13	NA

Dropping columns using select and where

If you had many columns it might not be feasible to specify the names directly. The `where` helper function can be used to specify a function to determine column selection.

```r
messy <-
  messy |>
  select(-where( function(x) all(is.na(x)) ))
```

Gene	DrugX_low	DrugX_high	DrugY_low_rep1	DrugY_low_rep2	DrugY_high_rep1	DrugY_high_rep2
GeneA	0.15	NA	-0.15	-0.15	-0.21	-0.22
GeneB	-0.07	-0.76	-0.11	0.10	0.01	-0.12
GeneC	-1.22	-0.27	-0.14	NA	0.10	-0.10
GeneD	-0.09	1.20	-0.02	-0.48	-0.11	0.16
GeneE	-0.60	1.01	-0.05	-0.53	-0.47	0.24
GeneF	NA	1.39	NA	NA	-0.13	NA

Reshaping a data frame using pivoting

`tidyr::pivot_longer` collapses multiple columns into a single column, and create a new column from the collapse columns headers:

```r
long_messy <-
  messy |>
  pivot_longer(cols = -Gene,
               names_to="Drug_Dosage_Rep",
               values_to="Expression")
```

Gene	Drug_Dosage_Rep	Expression
GeneA	DrugX_low	0.15
GeneA	DrugX_high	NA
GeneA	DrugY_low_rep1	-0.15
GeneA	DrugY_low_rep2	-0.15
GeneA	DrugY_high_rep1	-0.21
GeneA	DrugY_high_rep2	-0.22
GeneB	DrugX_low	-0.07
GeneB	DrugX_high	-0.76
GeneB	DrugY_low_rep1	-0.11
GeneB	DrugY_low_rep2	0.10

Extract a column in multiple columns

`tidyr::separate_wider_delim` splits a single column into multiple columns based on a character delimiter.

```r
split_long_messy <-
  long_messy |>
  separate_wider_delim(cols = Drug_Dosage_Rep,
                       delim = "_",
                       names = c("Drug", "Dosage", "Replicate"),
                       too_few = "align_start")
```

Gene	Drug	Dosage	Replicate	Expression
GeneA	DrugX	low	NA	0.15
GeneA	DrugX	high	NA	NA
GeneA	DrugY	low	rep1	-0.15
GeneA	DrugY	low	rep2	-0.15
GeneA	DrugY	high	rep1	-0.21
GeneA	DrugY	high	rep2	-0.22
GeneB	DrugX	low	NA	-0.07
GeneB	DrugX	high	NA	-0.76
GeneB	DrugY	low	rep1	-0.11
GeneB	DrugY	low	rep2	0.10

Filling missing data

`tidyr::replace_na` can be used to replace `NA` values with a default:

```r
tidy_data <-
  split_long_messy |>
  replace_na(list(Replicate = "rep1"))
```

Gene	Drug	Dosage	Replicate	Expression
GeneA	DrugX	low	rep1	0.15
GeneA	DrugX	high	rep1	NA
GeneA	DrugY	low	rep1	-0.15
GeneA	DrugY	low	rep2	-0.15
GeneA	DrugY	high	rep1	-0.21
GeneA	DrugY	high	rep2	-0.22
GeneB	DrugX	low	rep1	-0.07
GeneB	DrugX	high	rep1	-0.76
GeneB	DrugY	low	rep1	-0.11
GeneB	DrugY	low	rep2	0.10

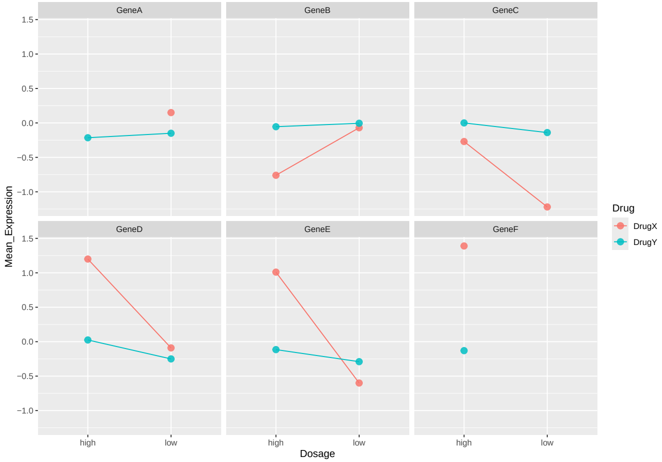
Tidy data enables complex summaries

```r
tidy_summary <-
  tidy_data |>
  group_by(Gene, Drug, Dosage) |>
  summarize(Mean_Expression = mean(Expression, na.rm=TRUE))
```

Gene	Drug	Dosage	Mean_Expression
GeneA	DrugX	high	NaN
GeneA	DrugX	low	0.150
GeneA	DrugY	high	-0.215
GeneA	DrugY	low	-0.150
GeneB	DrugX	high	-0.760
GeneB	DrugX	low	-0.070
GeneB	DrugY	high	-0.055
GeneB	DrugY	low	-0.005
GeneC	DrugX	high	-0.270
GeneC	DrugX	low	-1.220

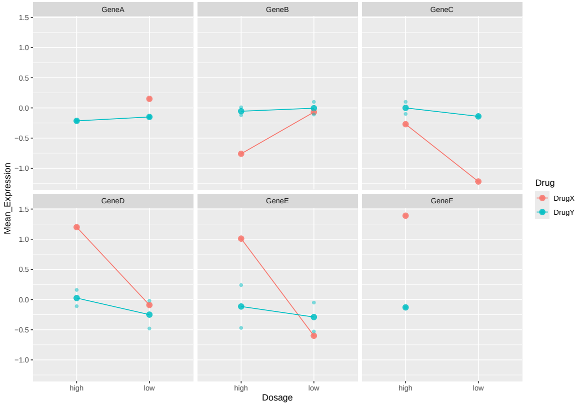
Tidy data enables complex plotting

```
tidy_summary |>
  ggplot(aes(x = Dosage, y = Mean_Expression, color=Drug)) +
  geom_point(alpha=0.85, size = 3) +
  geom_line(aes(group=Drug)) +
  facet_wrap(~Gene)
```



Summary and raw values as different ggplot layers

```r
tidy_summary |>
  ggplot(aes(x = Dosage, y = Mean_Expression, color=Drug)) +
  geom_point(alpha=0.85, size=3) +
  geom_line(aes(group=Drug)) +
  geom_point(data=tidy_data,
             mapping=aes(x = Dosage, y = Expression, color=Drug),
             alpha = 0.5, size = 1.5,
             inherit.aes = FALSE) +
  facet_wrap(~Gene)
```



Widening

`tidyr::pivot_wider` creates new columns and headers from specified columns:

```r
tidy_wide <-
  tidy_summary |>
  pivot_wider(names_from = Gene, values_from = Mean_Expression)
```

Drug	Dosage	GeneA	GeneB	GeneC	GeneD	GeneE	GeneF
DrugX	high	NaN	-0.760	-0.27	1.200	1.010	1.39
DrugX	low	0.150	-0.070	-1.22	-0.090	-0.600	NaN
DrugY	high	-0.215	-0.055	0.00	0.025	-0.115	-0.13
DrugY	low	-0.150	-0.005	-0.14	-0.250	-0.290	NaN

```r
corr_from_wide <-
  tidy_summary |>
  ungroup() |>  # ungroup important here
  pivot_wider(names_from = Gene, values_from = Mean_Expression) |>
  select(-c(Drug,Dosage,GeneF)) |>
  cor(use = "pairwise.complete.obs" )
```

	GeneA	GeneB	GeneC	GeneD	GeneE
GeneA	1.0000000	-0.5464173	-0.9980371	-0.0734459	-0.9807466
GeneB	-0.5464173	1.0000000	-0.1197536	-0.9929120	-0.9472035
GeneC	-0.9980371	-0.1197536	1.0000000	0.1652294	0.4313106
GeneD	-0.0734459	-0.9929120	0.1652294	1.0000000	0.9572841
GeneE	-0.9807466	-0.9472035	0.4313106	0.9572841	1.0000000