tidyr

Easily reshape your data

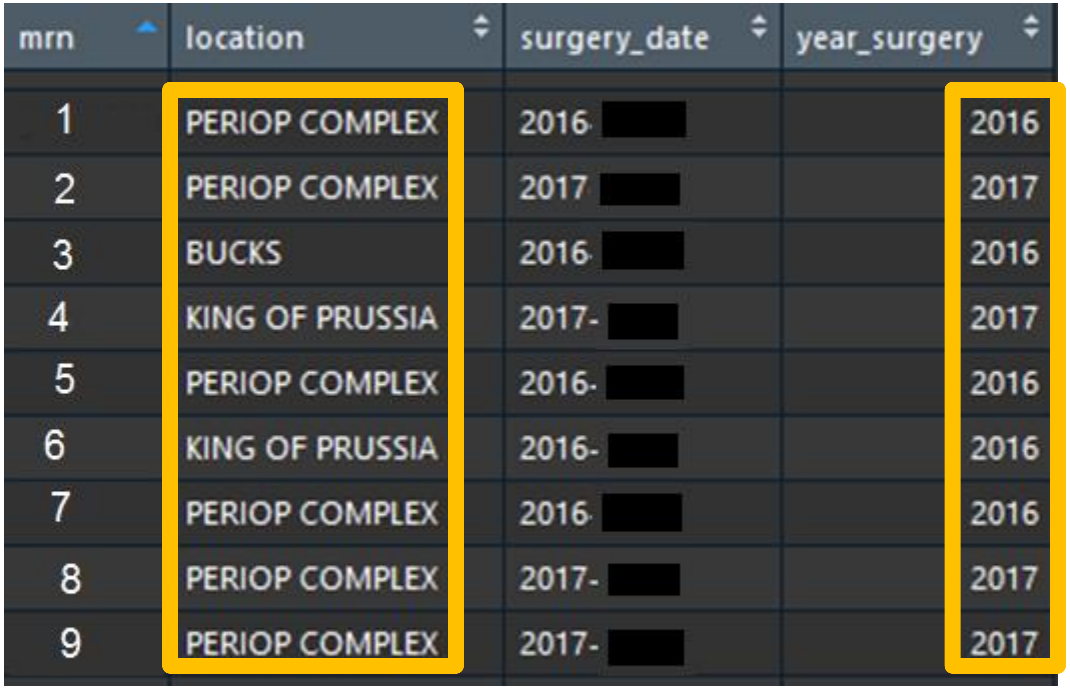
tidyr::reshaping your data

- **spread()**: create a column for each unique value of a single variable
  - long format to wide
  - no similar function in Netezza, just a ton of max(case when…) statements

- **gather()**: put the values of several columns into rows
  - wide format to long
  - no similar function in Netezza, would be replaced with tedious unions

- **comparison to excel**: not limited to numeric values!

tidyr::why use it?

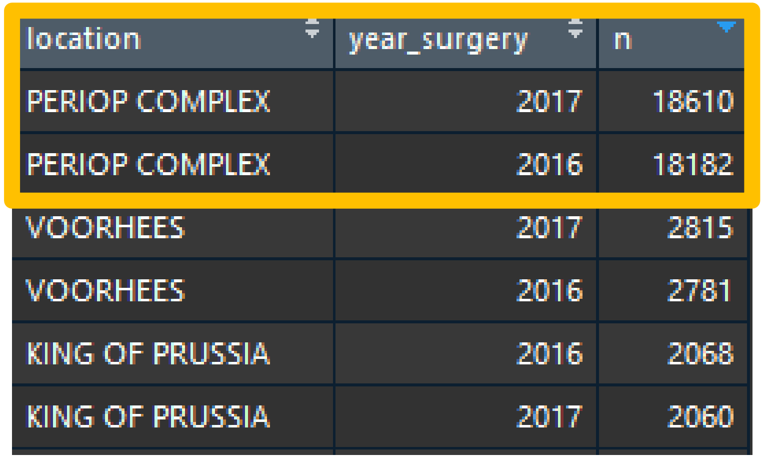


mrn	location	surgery_date	year_surgery
1	PERIOP COMPLEX	2016 [redacted]	2016
2	PERIOP COMPLEX	2017 [redacted]	2017
3	BUCKS	2016 [redacted]	2016
4	KING OF PRUSSIA	2017- [redacted]	2017
5	PERIOP COMPLEX	2016- [redacted]	2016
6	KING OF PRUSSIA	2016- [redacted]	2016
7	PERIOP COMPLEX	2016 [redacted]	2016
8	PERIOP COMPLEX	2017- [redacted]	2017
9	PERIOP COMPLEX	2017- [redacted]	2017

`spread()` and `gather()` can help us look at year over year comparisons for each surgical location

tidyr::spread

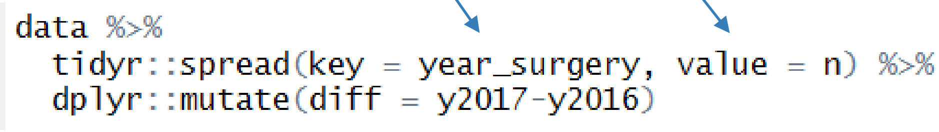
from long format to wide



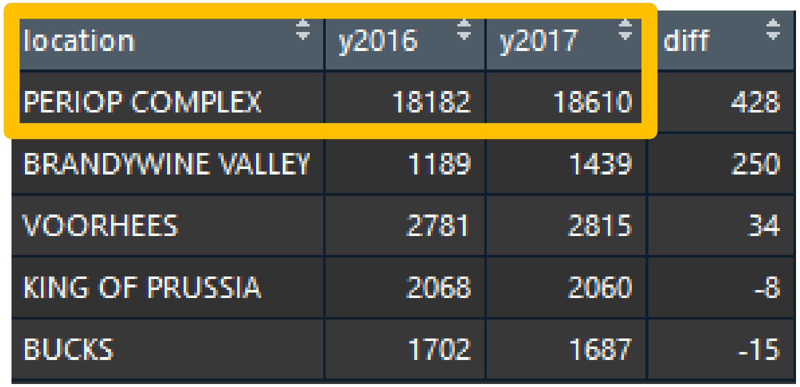
location	year_surgery	n
PERIOP COMPLEX	2017	18610
PERIOP COMPLEX	2016	18182
VOORHEES	2017	2815
VOORHEES	2016	2781
KING OF PRUSSIA	2016	2068
KING OF PRUSSIA	2017	2060

the name of the column to pivot

the name of the column associated with each column



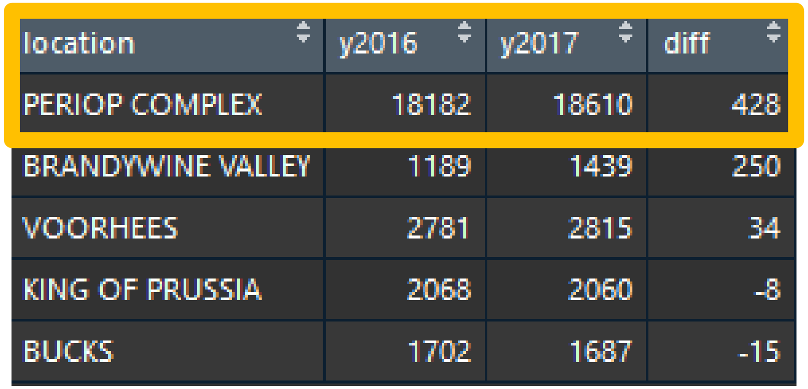
```
data %>%
  tidyr::spread(key = year_surgery, value = n) %>%
  dplyr::mutate(diff = y2017-y2016)
```



location	y2016	y2017	diff
PERIOP COMPLEX	18182	18610	428
BRANDYWINE VALLEY	1189	1439	250
VOORHEES	2781	2815	34
KING OF PRUSSIA	2068	2060	-8
BUCKS	1702	1687	-15

tidyr::gather

from wide format to long



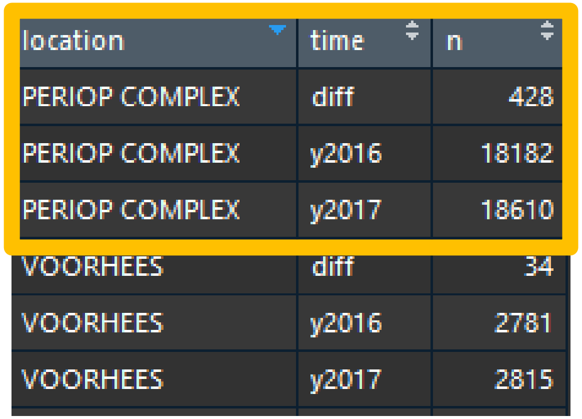
location	y2016	y2017	diff
PERIOP COMPLEX	18182	18610	428
BRANDYWINE VALLEY	1189	1439	250
VOORHEES	2781	2815	34
KING OF PRUSSIA	2068	2060	-8
BUCKS	1702	1687	-15

what to include/exclude, here I’m excluding location from the pivot

the new name for that will hold the column headers

the new name for the column that will hold the column values

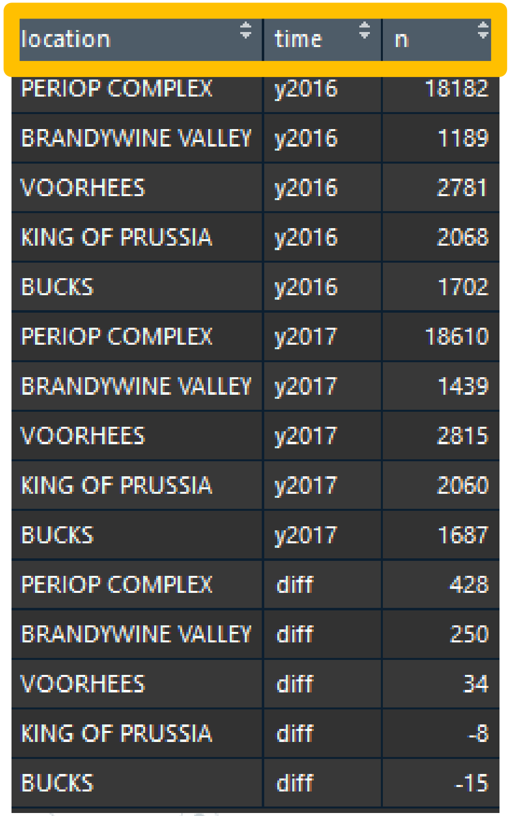
```
data %>%
  tidyr::gather(key = time, value = n, -location)
```



location	time	n
PERIOP COMPLEX	diff	428
PERIOP COMPLEX	y2016	18182
PERIOP COMPLEX	y2017	18610
VOORHEES	diff	34
VOORHEES	y2016	2781
VOORHEES	y2017	2815

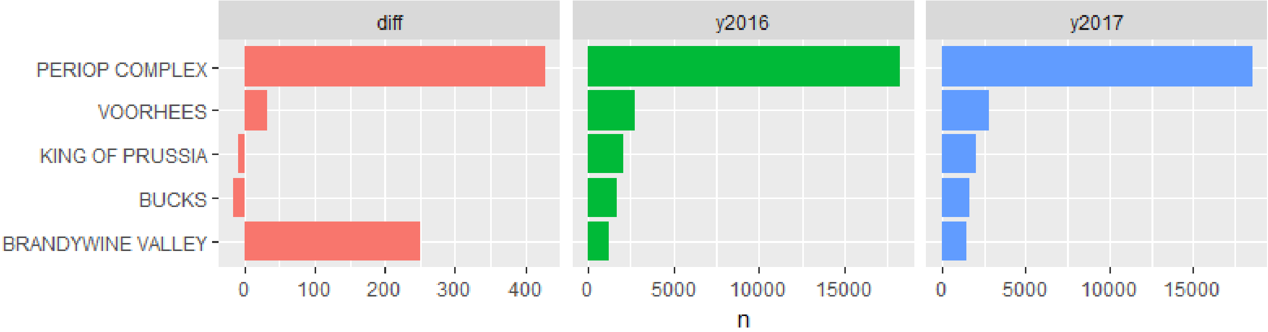
you might want a dataset like for ggplot

tidyr::plotting reshaped data



location	time	n
PERIOP COMPLEX	y2016	18182
BRANDYWINE VALLEY	y2016	1189
VOORHEES	y2016	2781
KING OF PRUSSIA	y2016	2068
BUCKS	y2016	1702
PERIOP COMPLEX	y2017	18610
BRANDYWINE VALLEY	y2017	1439
VOORHEES	y2017	2815
KING OF PRUSSIA	y2017	2060
BUCKS	y2017	1687
PERIOP COMPLEX	diff	428
BRANDYWINE VALLEY	diff	250
VOORHEES	diff	34
KING OF PRUSSIA	diff	-8
BUCKS	diff	-15

```
ggplot(cases_long) +
    geom_col(aes(x = location, y = n, fill = time)) +
    facet_grid(~time, scales = "free") +
    coord_flip()
```



tidyr::comparisons

both have the same first 2 arguments

Arguments	Gather wide to long	Spread long to wide
key =	new column name (was column headers)	name of column to pivot into headers
value =	new column name (was column values)	name of column that has the values that will go below the new columns
…	need to specify any variables that shouldn’t be included (or explicitly say which to include) as third argument	fill = … (0, “unknown”, “ ”) will prevent NA’s in your data can be some other value if pivoting text

tidyr::exercise

- Find the total number of central line uses by culture source by department
- From this, create a wide-form dataset having 1 row per department and a column for each culture source to show the totals, use a zero for any NAs. Store this as a new data frame.
- Pivot the data back into a long-form dataset that has the same # of rows and columns as step 1. Store this as a new data frame.