Getting / cleaning data 2

Tidying with `dplyr`

VADeaths data

For this example, I'll use the VADeaths dataset that comes with R.

This dataset gives the death rates per 1,000 people in Virginia in 1940. It gives death rates by age, gender, and rural / urban:

```r
data("VADeaths")
VADeaths
```

```
##       Rural Male Rural Female Urban Male Urban Female
## 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
```

VADeaths data

There are a few things that make this data untidy:

- One variable (age category) is saved as row names, rather than a column.
- Other variables (gender, rural / urban) are in column names.
- Once you gather the data, you will have two variables (gender, rural / urban) in the same column.

In the following slides, we'll walk through how to tidy this data.

Tidying the `VADeaths` data

(1) One variable (age category) is saved as row names, rather than a column.

To fix this, we need to convert the row names into a new column. We can do this using `mutate` (load `tibble` if needed):

```r
VADeaths %>%
  as.data.frame() %>% ## Convert from matrix to dataframe
  rownames_to_column(var = "age")
```

```
##     age Rural Male Rural Female Urban Male Urban Female
## 1 50-54       11.7          8.7       15.4          8.4
## 2 55-59       18.1         11.7       24.3         13.6
## 3 60-64       26.9         20.3       37.0         19.3
## 4 65-69       41.0         30.9       54.6         35.1
## 5 70-74       66.0         54.3       71.1         50.0
```

Tidying the VADeaths data

(2) Two variables (gender, rural / urban) are in column names.

Gather the data to convert column names to a new column:

```
VADeaths %>%
  as.data.frame() %>%
  rownames_to_column(var = "age") %>%
  pivot_longer(- age, names_to = "gender_loc", values_to = "mort
  slice(1:4)
```

```
## # A tibble: 4 x 3
##   age   gender_loc   mort_rate
##   <chr> <chr>            <dbl>
## 1 50-54 Rural Male        11.7
## 2 50-54 Rural Female       8.7
## 3 50-54 Urban Male        15.4
## 4 50-54 Urban Female       8.4
```

Tidying the VADeaths data

(3) Two variables (gender, rural / urban) in the same column.

Separate the column into two separate columns for “gender” and “loc” (rural / urban):

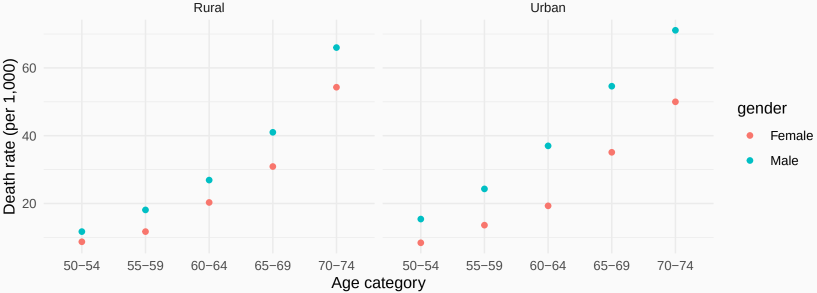
```
VADeaths %>%
  as.data.frame() %>%
  rownames_to_column(var = "age") %>%
  pivot_longer(- age, names_to = "gender_loc", values_to = "mort
  separate(col = gender_loc, into = c("gender", "loc"),
           sep = " ") %>%
  slice(1)
```

```
## # A tibble: 1 x 4
##   age   gender loc   mort_rate
##   <chr> <chr>  <chr>     <dbl>
## 1 50-54 Rural  Male       11.7
```

Tidying the VADeaths data

Now that the data is tidy, it's much easier to plot:

```
ggplot(VADeaths, aes(x = age, y = mort_rate,
                     color = gender)) +
  geom_point() +
  facet_wrap( ~ loc, ncol = 2) +
  xlab("Age category") + ylab("Death rate (per 1,000)") +
  theme_minimal()
```


Rural
Urban
Death rate (per 1,000)
Age category
gender
● Female
● Male