Be fruitful and dplyr

PROGRAMMING WITH DPLYR

Dr. Chester Ismay

Educator, Data Scientist, and R/Python Consultant

Course prerequisites

- Joining Data with dplyr
- Introduction to Writing Functions in R

Course outline

Chapter 1

- Refresh `dplyr` pipelines
- Choose columns based on patterns

Chapter 2

- Move columns around in your data
- Transform across multiple columns of data

Chapter 3

- Strengthen `dplyr` join knowledge
- Use set theory clauses to improve programming skills with multiple data sources

Chapter 4

- Create functions to wrap `dplyr` and `ggplot2` code
- Use the `rlang` package to decipher tidy evaluation

The world_bank_data tibble

country	region	year	infant_mortality_rate	fertility_rate	perc_rural_pop
Saudi Arabia	Western Asia	2013	13.3	2.64	17.260
Greece	Southern Europe	2014	3.7	1.54	22.298
Latvia	Northern Europe	2014	7.2	1.62	32.048
Romania	Eastern Europe	2014	10.1	1.43	46.100
Netherlands	Western Europe	2015	3.2	1.78	9.827

world_bank_data columns

```
names(world_bank_data)
```

```
 [1] "iso"                  "country"                "continent"
 [4] "region"               "year"                   "infant_mortality_rate"
 [7] "fertility_rate"       "perc_electric_access"   "perc_college_complete"
[10] "perc_cvd_crd_70"      "unemployment_rate"      "perc_rural_pop"
```

Select some columns from world_bank_data

```
world_bank_data %>%
    select(country, continent, region, year, perc_rural_pop, perc_college_complete)
```

```
# A tibble: 300 x 6
   country      continent region           year perc_rural_pop perc_college_complete
   <chr>        <fct>     <fct>           <dbl>          <dbl>                 <dbl>
 1 Portugal     Europe    Southern Europe  2000           45.6                  7.26
 2 Armenia      Asia      Western Asia     2001           35.6                 20.4
 3 Bulgaria     Europe    Eastern Europe   2001           30.8                 18.0
 4 Portugal     Europe    Southern Europe  2001           45.0                  7.57
 5 Qatar        Asia      Western Asia     2004            2.91                20.9
 6 Saudi Arabia Asia      Western Asia     2004           19.2                 14.9
 7 Pakistan     Asia      Southern Asia    2005           66.0                  3.92
# ... with 293 more rows
```

Filter rows to match continent values

```
continents_vector <- c("Africa", "Asia")
```

```
asia_africa_results <- world_bank_data %>%
    select(country, continent, region, year, perc_rural_pop, perc_college_complete) %>%
    filter(continent %in% continents_vector)
```

Results of row filter

```
asia_africa_results
```

```
# A tibble: 111 x 6
   country      continent region              year perc_rural_pop perc_college_complete
   <chr>        <fct>     <fct>              <dbl>          <dbl>                  <dbl>
 1 Armenia      Asia      Western Asia        2001          35.6                   20.4
 2 Qatar        Asia      Western Asia        2004           2.91                  20.9
 3 Saudi Arabia Asia      Western Asia        2004          19.2                   14.9
 4 Pakistan     Asia      Southern Asia       2005          66.0                    3.92
 5 Nigeria      Africa    Western Africa      2006          60.1                    9.04
 6 Pakistan     Asia      Southern Asia       2006          65.8                    6.30
 7 Singapore    Asia      South-Eastern Asia  2006           0                     19.6
 8 Azerbaijan   Asia      Western Asia        2007          47.2                   14.9
 9 Qatar        Asia      Western Asia        2007           2.08                  25.1
10 Singapore    Asia      South-Eastern Asia  2007           0                     20.1
# ... with 101 more rows
```

Mutate a new column

```
asia_africa_results <- asia_africa_results %>%
    mutate(perc_urban_pop = 100 - perc_rural_pop)
```

Results of mutate

```
# A tibble: 111 x 7
   country      continent region              year perc_rural_pop perc_college_complete perc_urban_pop
   <chr>        <fct>     <fct>               <dbl>          <dbl>                 <dbl>          <dbl>
 1 Armenia      Asia      Western Asia         2001          35.6                  20.4            64.4
 2 Qatar        Asia      Western Asia         2004           2.91                 20.9            97.1
 3 Saudi Arabia Asia      Western Asia         2004          19.2                  14.9            80.8
 4 Pakistan     Asia      Southern Asia        2005          66.0                   3.92           34.0
 5 Nigeria      Africa    Western Africa       2006          60.1                   9.04           39.9
 6 Pakistan     Asia      Southern Asia        2006          65.8                   6.30           34.2
 7 Singapore    Asia      South-Eastern Asia   2006           0                    19.6           100
 8 Azerbaijan   Asia      Western Asia         2007          47.2                  14.9            52.8
 9 Qatar        Asia      Western Asia         2007           2.08                 25.1            97.9
10 Singapore    Asia      South-Eastern Asia   2007           0                    20.1           100
# ... with 101 more rows
```

Analyze urban percentage across regions

```
asia_africa_results %>%
    group_by(region) %>%
    summarize(
      mean_urban = mean(perc_urban_pop)
    )
```

```
# A tibble: 9 x 2
  region                mean_urban
  <fct>                      <dbl>
1 Central Asia                49.2
2 Eastern Africa              19.5
3 Eastern Asia                74.2
4 Middle Africa               42.4
5 South-Eastern Asia          79.8
6 Southern Africa             64.8
7 Southern Asia               40.0
8 Western Africa              39.6
9 Western Asia                78.9
```

Let's practice!

PROGRAMMING WITH DPLYR

Lending a helper hand

PROGRAMMING WITH DPLYR

Dr. Chester Ismay

Educator, Data Scientist, and R/Python Consultant

starts_with()

- Looks for column names starting with a given substring when combined with `select()`
- In the `tidyselect` package, but automatically loaded when `library(dplyr)` is run

starts_with() example

```
world_bank_data %>%
    select(starts_with("perc"))
```

```
# A tibble: 300 x 4
   perc_electric_access perc_college_complete perc_cvd_crd_70 perc_rural_pop
                  <dbl>                 <dbl>           <dbl>          <dbl>
 1                  100                  7.26            15.8           45.6
 2                  100                 20.4             26.2           35.6
 3                  100                 18.0             28.1           30.8
 4                  100                  7.57            15.5           45.0
 5                   83.8                3.92            33.3           66.0
# ... with 295 more rows
```

Where and when are these results from!?

```
world_bank_data %>%
  select(country, year, starts_with("perc"))
```

```
# A tibble: 300 x 6
   country       year perc_electric_access perc_college_complete perc_cvd_crd_70 perc_rural_pop
   <chr>        <dbl>                <dbl>                  <dbl>           <dbl>          <dbl>
 1 Portugal      2000                100                     7.26            15.8           45.6
 2 Armenia       2001                100                    20.4             26.2           35.6
 3 Bulgaria      2001                100                    18.0             28.1           30.8
 4 Portugal      2001                100                     7.57            15.5           45.0
 5 Pakistan      2005                 83.8                   3.92            33.3           66.0
# ... with 295 more rows
```

ends_with()

```
world_bank_data %>%
    select(country, year, ends_with("rate"))
```

```
# A tibble: 300 x 5
   country       year infant_mortality_rate fertility_rate unemployment_rate
   <chr>        <dbl>                 <dbl>           <dbl>              <dbl>
 1 Portugal      2000                   5.5            1.47               3.81
 2 Armenia       2001                  25.3            1.2               10.9
 3 Bulgaria      2001                  17.1            1.2               19.9
 4 Portugal      2001                   5.2            1.46               3.83
 5 Pakistan      2005                  80              3.79               0.610
# ... with 295 more rows
```

Let's practice!

PROGRAMMING WITH DPLYR

Finding our perfect match

PROGRAMMING WITH DPLYR

Dr. Chester Ismay

Educator, Data Scientist, and R/Python Consultant

```
glimpse(world_bank_data)
```

```
Rows: 300
Columns: 12
$ iso                   <chr> "PRT", "ARM", "BGR", "PRT", "PRT", "PRT", "PRT...
$ country               <chr> "Portugal", "Armenia", "Bulgaria", "Portugal",...
$ continent             <fct> Europe, Asia, Europe, Europe, Europe, Europe, ...
$ region                <fct> Southern Europe, Western Asia, Eastern Europe,...
$ year                  <dbl> 2000, 2001, 2001, 2001, 2002, 2003, 2004, 2004...
$ infant_mortality_rate <dbl> 5.5, 25.3, 17.1, 5.2, 4.7, 4.3, 4.0, 9.2, 17.2...
$ fertility_rate        <dbl> 1.47, 1.20, 1.20, 1.46, 1.45, 1.44, 1.43, 2.70...
$ perc_electric_access  <dbl> 100.00000, 100.00000, 100.00000, 100.00000, 10...
$ perc_college_complete <dbl> 7.25665, 20.35655, 18.04557, 7.57332, 7.69954,...
$ perc_cvd_crd_70       <dbl> 15.8, 26.2, 28.1, 15.5, 15.0, 14.8, 14.0, 23.8...
$ unemployment_rate     <dbl> 3.81, 10.91, 19.92, 3.83, 4.50, 6.13, 6.32, 0....
$ perc_rural_pop        <dbl> 45.601, 35.615, 30.834, 44.956, 44.334, 43.713...
```

One way to select

```
world_bank_data %>%
    select(country, year, infant_mortality_rate, fertility_rate, unemployment_rate)
```

```
# A tibble: 300 x 5
   country       year infant_mortality_rate fertility_rate unemployment_rate
   <chr>        <dbl>                 <dbl>           <dbl>              <dbl>
 1 Portugal      2000                   5.5            1.47               3.81
 2 Armenia       2001                  25.3            1.2               10.9
 3 Bulgaria      2001                  17.1            1.2               19.9
 4 Portugal      2001                   5.2            1.46               3.83
 5 Pakistan      2005                  80              3.79               0.610
# ... with 295 more rows
```

Another way using contains()

- What is similar about `country`, `year`, `infant_mortality_rate`, `fertility_rate`, and `unemployment_rate`?

```
world_bank_data %>%
    select(contains("y"))
```

```
# A tibble: 300 x 5
   country       year infant_mortality_rate fertility_rate unemployment_rate
   <chr>        <dbl>                 <dbl>          <dbl>             <dbl>
 1 Portugal      2000                   5.5           1.47              3.81
 2 Armenia       2001                  25.3           1.2              10.9
 3 Bulgaria      2001                  17.1           1.2              19.9
 4 Portugal      2001                   5.2           1.46              3.83
 5 Pakistan      2005                  80             3.79              0.610
# ... with 295 more rows
```

A dip into regular expressions

Tokens:

- `|` - For matching two strings
- `^` - For specifying the start of a string
- `$` - For specifying the end of a string

matches()

- Use `matches()` when looking for a regular expression

```
world_bank_data %>%
    select(matches("y|perc"))
```

```
# A tibble: 300 x 9
   country       year infant_mortality_rate fertility_rate perc_electric_access
   <chr>        <dbl>                 <dbl>          <dbl>                 <dbl>
 1 Portugal      2000                   5.5           1.47                  100
 2 Armenia       2001                  25.3           1.2                   100
 3 Bulgaria      2001                  17.1           1.2                   100
 4 Portugal      2001                   5.2           1.46                  100
 5 Pakistan      2005                  80             3.79                   83.8
# ... with 295 more rows, and 4 more variables: perc_college_complete <dbl>,
#   perc_cvd_crd_70 <dbl>, unemployment_rate <dbl>, perc_rural_pop <dbl>
```

Alternatives to starts_with() and ends_with()

```
world_bank_data %>%
    select(matches("^co"))
```

```
# A tibble: 300 x 2
   country      continent
   <chr>        <fct>
 1 Portugal     Europe
 2 Armenia      Asia
 3 Bulgaria     Europe
 4 Portugal     Europe
 5 Pakistan     Asia
# ... with 295 more rows
```

```
world_bank_data %>%
    select(country, matches("on$"))
```

```
# A tibble: 300 x 2
   country      region
   <chr>        <fct>
 1 Portugal     Southern Europe
 2 Armenia      Western Asia
 3 Bulgaria     Eastern Europe
 4 Portugal     Southern Europe
 5 Pakistan     Southern Asia
# ... with 290 more rows
```

Let's practice!

PROGRAMMING WITH DPLYR