Data Wrangling with `dplyr()`

Abhinav Anand

```
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

```
#what is the format? wide or long?
```

Tibbles

It prints differently because it's a tibble. Tibbles are data frames, but slightly tweaked to work better in the tidyverse

1. `fct`
2. `dbl`
3. `int`

Other types: `chr`, `lgl`, `date` etc.

`dplyr()`: The Main Verbs

Pick observations by their values (filter()). Reorder the rows (arrange()). Pick variables by their names (select()). Create new variables with functions of existing variables (mutate()). Collapse many values down to a single summary (summarise()). These can all be used in conjunction with group_by() which changes the scope of each function from operating on the entire dataset to operating on it group-by-group. These six functions provide the verbs for a language of data manipulation.

All verbs work similarly:

The first argument is a data frame.

The subsequent arguments describe what to do with the data frame, using the variable names (without quotes).

The result is a new data frame.

Filter

Arrange

Select

Mutate

Summarise

Grouped Mutate