Hw02

Cody

September 22, 2017

```
library(gapminder)
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

Smell test

- Is it a data.frame, a matrix, a vector, a list?

```
str(gapminder)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1704 obs. of  6 variables:
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 16317921 22
##  $ gdpPercap: num  779 821 853 836 740 ...
```

A: Gapminder is a data.frame.

- What's its class?

```
class(gapminder)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

A: It is actually a tibble, a particular kind of data.frame favorable in the tidyverse.

- How many variables/columns?

```
ncol(gapminder)
```

```
## [1] 6
```

A: 6

- How many rows/observations?

```
nrow(gapminder)
```

```
## [1] 1704
```

A: 1704

- Can you get these facts about "extent" or "size" in more than one way? Can you imagine different functions being useful in different contexts?

```
dim(gapminder)
```

```
## [1] 1704    6
```

A: Yes, there are redundancies in some of these functions and theres good reason for it! Perhaps different inputs are useful for one function over another, the speed of computation could be better for a particular data type in one function or even the outputs of the functions can be used in different manors.

- What data type is each variable?

```
head(gapminder)
```

```
## # A tibble: 6 x 6
##       country continent  year lifeExp      pop gdpPercap
##        <fctr>    <fctr> <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan      Asia  1952  28.801  8425333  779.4453
## 2 Afghanistan      Asia  1957  30.332  9240934  820.8530
## 3 Afghanistan      Asia  1962  31.997 10267083  853.1007
## 4 Afghanistan      Asia  1967  34.020 11537966  836.1971
## 5 Afghanistan      Asia  1972  36.088 13079460  739.9811
## 6 Afghanistan      Asia  1977  38.438 14880372  786.1134
```

A: Country and Continent- Factors, Year and Population- Integers, Life Expectancy and GDP per Capita- Double.

Explore individual Variables

Categorical Variable= Country:

- What are possible values (or range, whichever is appropriate) of each variable?
- What values are typical? What's the spread? What's the distribution? Etc., tailored to the variable at hand.
- Feel free to use summary stats, tables, figures. We're NOT expecting high production value (yet).

Quantitative Variable= Life Expectancy:

- What are possible values (or range, whichever is appropriate) of each variable?
- What values are typical? What's the spread? What's the distribution? Etc., tailored to the variable at hand.
- Feel free to use summary stats, tables, figures. We're NOT expecting high production value (yet).

Explore various plot types

Make a few plots, probably of the same variable you chose to characterize numerically. Try to explore more than one plot type. Just as an example of what I mean:

- A scatterplot of two quantitative variables.
- A plot of one quantitative variable. Maybe a histogram or densityplot or frequency polygon.
- A plot of one quantitative variable and one categorical. Maybe boxplots for several continents or countries.

You don't have to use all the data in every plot! It's fine to filter down to one country or small handful of countries.

Use filter(), select() and %>%

Use filter() to create data subsets that you want to plot.

Practice piping together filter() and select(). Possibly even piping into ggplot().

But I want to do more!

Evaluate this code and describe the result. Presumably the analyst's intent was to get the data for Rwanda and Afghanistan. Did they succeed? Why or why not? If not, what is the correct way to do this?

```
filter(gapminder, country == c("Rwanda", "Afghanistan"))
```

```
## # A tibble: 12 x 6
##        country continent  year lifeExp      pop gdpPercap
##         <fctr>    <fctr> <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan      Asia  1957  30.332  9240934  820.8530
##  2 Afghanistan      Asia  1967  34.020 11537966  836.1971
##  3 Afghanistan      Asia  1977  38.438 14880372  786.1134
##  4 Afghanistan      Asia  1987  40.822 13867957  852.3959
##  5 Afghanistan      Asia  1997  41.763 22227415  635.3414
##  6 Afghanistan      Asia  2007  43.828 31889923  974.5803
##  7       Rwanda    Africa  1952  40.000  2534927  493.3239
##  8       Rwanda    Africa  1962  43.000  3051242  597.4731
##  9       Rwanda    Africa  1972  44.600  3992121  590.5807
## 10       Rwanda    Africa  1982  46.218  5507565  881.5706
## 11       Rwanda    Africa  1992  23.599  7290203  737.0686
## 12       Rwanda    Africa  2002  43.413  7852401  785.6538
```

Present numerical tables in a more attractive form, such as using knitr::kable().

(SEE HW2)

Reflection

Reflect on what was hard/easy, problems you solved, helpful tutorials you read, etc. What things were hard, even though you saw them in class? What was easy(-ish) even though we haven't done it in class?