R introduction assignment

Oleksii Yehorchenkov

30 11 2020

This assignment is based on materials from Coursera course Introduction to Data Science in Python

Part 1

Preparing dataset

In the part 1 we are going to work with the olympics dataset (olympics.csv), which was derrived from the Wikipedia entry on All Time Olympic Games Medals

You can download the dataset by the link

At first you should do some basic clearning.

1. Read the file. File encoding is "UTF-8"
2. Give the 1st column name "Country"
3. Write a code for naming the next column:

- Remove from names "X.U.2116.." so "X.U.2116..Summer" will be "Summer"
- "X01.." change to "Gold" so "X01...1" will be "Gold.1"
- "X02.." and "X03.." change to "Silver" and "Bronze"

4. Clean the country's names to "Afghanistan", "Algeria", etc. Remove beginning and end spaces.
5. Add a new column "ID" with country code, for instance "AFG", "ALG", etc.
6. Save the tidy dataset to "olympics" variable.

```
knitr::opts_chunk$set(echo=TRUE, message=FALSE, warning=FALSE, fig.height=12)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
df <- read_csv("C:\\Users\\Soumy\\Documents\\data\\olympics.csv", skip = 1)
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Warning: Duplicated column names deduplicated: '01 !' => '01 !_1' [8], '02 !'
## => '02 !_1' [9], '03 !' => '03 !_1' [10], 'Total' => 'Total_1' [11], '01 !' =>
## '01 !_2' [13], '02 !' => '02 !_2' [14], '03 !' => '03 !_2' [15]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_character(),
##   `<U+2116> Summer` = col_double(),
##   `01 !` = col_double(),
##   `02 !` = col_double(),
##   `03 !` = col_double(),
##   Total = col_double(),
##   `<U+2116> Winter` = col_double(),
##   `01 !_1` = col_double(),
##   `02 !_1` = col_double(),
##   `03 !_1` = col_double(),
##   Total_1 = col_double(),
##   `<U+2116> Games` = col_double(),
##   `01 !_2` = col_double(),
##   `02 !_2` = col_double(),
##   `03 !_2` = col_double(),
##   `Combined total` = col_double()
## )
```

```
names(df) <- c("Country", "Summer", "Gold", "Silver", "Bronze", "Total", "Winter", "Gold.1", "Silver.1"
df <- df %>% tidyr::separate(Country, c("Country", "ID"), "[(]") %>% mutate(ID=gsub(")]","",ID))
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 1 rows [147].
```

```
olympics <- filter(df, !(df$Country == "Totals"))
```

Question 0 (example)

What is the first country in df? *Script should return a single string value.*

```
olympics$Country[1]
```

```
## [1] "Afghanistan "
```

Question 1

Which country has won the most gold medals in summer games?

Script should return a single string value.

```
olympics$Country[which.max(olympics$Gold)]
```

```
## [1] "United States "
```

Question 2

Which country had the biggest difference between their summer and winter gold medal counts?

Script should return a single string value.

```
olympics$Country[which.max(olympics$Gold - olympics$Gold.1)]
```

```
## [1] "United States "
```

Question 3

Which country has the biggest difference between their summer gold medal counts and winter gold medal counts relative to their total gold medal count?

$$\frac{Summer\ Gold - Winter\ Gold}{Total\ Gold}$$

Only include countries that have won at least 1 gold in both summer and winter.

Script should return a single string value.

```
difference <- olympics %>% mutate(diffPerc = abs(Gold - Gold.1)/Gold.2) %>% filter(Gold > 0) %>% filter
max <- which.max(difference$diffPerc)
difference$Country[[max]]
```

```
## [1] "Bulgaria "
```

Question 4

Write a function that creates a Series called "Points" which is a weighted value where each gold medal (`Gold.2`) counts for 3 points, silver medals (`Silver.2`) for 2 points, and bronze medals (`Bronze.2`) for 1 point.

Script should return a data frame of length 146 with 2 columns named "Country and"Points"

```
olympics_4<-mutate(olympics,Points=(3*Gold.2+2*Silver.2+Bronze.2))
olympics_4<-data.frame(olympics_4$Country,olympics_4$Points)
```

Part 2

For the next set of questions, we will be using census data from the United States Census Bureau. Counties are political and geographic subdivisions of states in the United States. This dataset contains population data for counties and states in the US from 2010 to 2015. See this document for a description of the variable names.

The census dataset (census.csv) should be loaded as census_df. Answer questions using this as appropriate.

Reading data

```
knitr::opts_chunk$set(echo=TRUE, message=FALSE, warning=FALSE, fig.height=12)
library(dplyr)
library(readr)
census_df <- read_csv("C:\\Users\\Soumy\\Documents\\data\\census.csv", skip = 1)
View(census_df)
```

Question 5

Which state has the most counties in it? (hint: consider the sumlevel key carefully! You'll need this for future questions too...)

Script should return a single string value.

```
census_df <- read_csv("C:\\Users\\Soumy\\Documents\\data\\census.csv")
census_counties <- census_df %>% filter(SUMLEV=="050")
census_counties <- table(census_counties$STNAME)
census_counties[which.max(census_counties)]
```

```
## Texas
##   254
```

Question 6

Only looking at the three most populous counties for each state, what are the three most populous states (in order of highest population to lowest population)? Use `CENSUS2010POP`.

Script should return a vector of string values.

```
census_counties <- census_df %>% filter(SUMLEV=="050")
census_counties$STNAME[order(census_counties$CENSUS2010POP, decreasing=TRUE)[1:3]]
```

```
## [1] "California" "Illinois"   "Texas"
```

Question 7

Which county has had the largest absolute change in population within the period 2010-2015? (Hint: population values are stored in columns POPESTIMATE2010 through POPESTIMATE2015, you need to consider all six columns.)

e.g. If County Population in the 5 year period is 100, 120, 80, 105, 100, 130, then its largest change in the period would be |130-80| = 50.

Script should return a single string value.

```
census_counties <- census_df %>% filter(SUMLEV=="050")
census_counties <- census_counties[,c("POPESTIMATE2010","POPESTIMATE2011","POPESTIMATE2012","POPESTIMATE
results<-(apply(census_counties,1,FUN=max)-apply(census_counties,1,FUN=min))
census<-census_df[which.max(results),]
census$CTYNAME
```

```
## [1] "Dallas County"
```

Question 8

In this datafile, the United States is broken up into four regions using the "REGION" column.

Create a query that finds the counties that belong to regions 1 or 2, whose name starts with 'Washington', and whose POPESTIMATE2015 was greater than their POPESTIMATE 2014.

Script function should return a 5x2 DataFrame with the columns "STNAME", "CTYNAME".

```
census_counties <- census_df %>% filter(SUMLEV=="050")
census_region <- census_counties %>% filter(REGION == 1 | REGION == 2)
census_washington <- census_region[(grep("Washington*",census$CTYNAME)), ]
census_grew <- subset(census,POPESTIMATE2015>POPESTIMATE2014)
census_8 <- data.frame(census_grew$STNAME, census_grew$CTYNAME)
```