# CSC/ST 442 Assignment 1

## Instruction

This assignment consists of 3 problems. The assignment is due on **Wednesday, September 1** at 11:59pm EDT. Please submit your assignment electronically through the moodle webpage. You are encouraged (but not required) to use RMarkdown to write up your homework solution. To start using Rmarkdown read

- Section 40.2 of Introduction to Data Science
- the RStudio tutorial
- the Rmarkdown cheatsheet.

## Problem 1 (30pts)

This problem uses a sample of data from the Gapminder foundation. If you haven't yet watched it, you might want to take a few minutes to watch the following Hans Rosling TeD talk

The data that we want to use is part of the *gapminder* library. We can install this library and load the dataset using the following code chunk.

```
install.packages("gapminder")
library(gapminder)
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
```

The variables `lifeExp`, `pop` and `gdpPercap` describe the life expectancy at birth, the total population, and the per-capita GDP for each country in the observed year.

Using this dataset, answer the following questions.

1. (5pts) How many observations do we have per continent ?
2. (5pts) How many (distinct) countries do we have for each continent ? Hint: Try the functions n_distinct().
3. (5pts) Create a new column named `gdp_ratio` whose values are the GDP for that country and year divided by the corresponding GDP for the United States that year. For example, the value of `gdp_ratio` for Afghanistan in 1952 is roughly 0.00298; that is to say, the GDP of Afghanistan in 1952 is approximately 0.00298 times that of the United States in 1952. Hint: First create a column for the `GDP` of each country. Then try the following code

```
gapminder %>% arrange(country,year) %>% mutate(gdp_ratio = GDP/GDP[country == "United States"])
```

4. (5pts) Now look at the countries in Asia. For every unique year in the dataset (namely $1952, 1957, 1962, ...$), which country has the lowest life Expectancy ? Which country has the highest lifeExpectancy ? Hint: use a **grouped** filter.
5. (10pts) For every continent, find the country that experienced the sharpest 5 year drop in life expectancy during the period from 1952 to 1997. Hint: the change in life expectancy for each country can be computed via the code chunk

```
library(dplyr)
library(gapminder)
gapminder %>% group_by(country) %>% mutate(lifeExp_change = lifeExp - lag(lifeExp)) %>%
  select(lifeExp_change, everything())
```

```
## # A tibble: 1,704 x 7
## # Groups:   country [142]
##    lifeExp_change country     continent  year lifeExp      pop gdpPercap
##             <dbl> <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1        NA      Afghanistan Asia       1952    28.8  8425333      779.
##  2         1.53   Afghanistan Asia       1957    30.3  9240934      821.
##  3         1.66   Afghanistan Asia       1962    32.0 10267083      853.
##  4         2.02   Afghanistan Asia       1967    34.0 11537966      836.
##  5         2.07   Afghanistan Asia       1972    36.1 13079460      740.
##  6         2.35   Afghanistan Asia       1977    38.4 14880372      786.
##  7         1.42   Afghanistan Asia       1982    39.9 12881816      978.
##  8         0.968  Afghanistan Asia       1987    40.8 13867957      852.
##  9         0.852  Afghanistan Asia       1992    41.7 16317921      649.
## 10         0.0890 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

Note that the `lifeExp_change` for the year 1952 is always missing as 1952 is the first year that has data collected.

## Problem 2 (30pts)

Using the *flights* dataset from the **nycflights13** library, answer the following question. Note that this question is slightly more open ended than the previous question.

1. (5pts) Which plane tailnum has the worst on-time record ? You might want to look only at planes with say at least 50 flights (because if the plane only fly once then the on-time record is not particularly accurate).
2. (5pts) Look at the proportion of cancelled flights (compared to the total number of flights) per day (let us define a cancelled flight as one for which the departure time or the arrival time is missing). Is the proportion of cancelled flights per day related to the average delay per day ?
3. (10pts) What time of day (morning, noon, afternoon, evening) should we fly if we want to avoid delay as much as possible ? Hint: You might want to look at the function case_when to convert the variable `sched_dep_time` into the time of day (morning, noon, afternoon, evening). For example

```
library(dplyr)
library(nycflights13)
data(flights)
flights_tod <- flights %>% mutate(time_of_day = case_when(
  sched_dep_time <= 1100 ~ "morning",
  between(sched_dep_time,1101,1400) ~ "noon",
  between(sched_dep_time,1401,1800) ~ "afternoon",
  sched_dep_time > 1801 ~ "evening"
)) %>% select(time_of_day, everything())
flights_tod
```

```
## # A tibble: 336,776 x 20
##    time_of_day  year month   day dep_time sched_dep_time dep_delay arr_time
##    <chr>       <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1 morning      2013     1     1      517            515         2      830
##  2 morning      2013     1     1      533            529         4      850
##  3 morning      2013     1     1      542            540         2      923
##  4 morning      2013     1     1      544            545        -1     1004
##  5 morning      2013     1     1      554            600        -6      812
##  6 morning      2013     1     1      554            558        -4      740
##  7 morning      2013     1     1      555            600        -5      913
##  8 morning      2013     1     1      557            600        -3      709
##  9 morning      2013     1     1      557            600        -3      838
## 10 morning      2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>, origin <chr>,
## #   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

4. (10pts) Departure delays are typically temporally correlated; once the problem that caused the initial delay has been resolved, later flights are delayed to allow earlier flights to leave. Using the lag function, explores how the delay of a flight (at a particular origin) is related to the delay of the immediately preceding flight (at the same origin). You might want to arrange the flights in order of month followed by day followed by dep$_t$ime before using lag.

# Problem 3 (10pts)

Choose either one of the following two problems

(a) Go on to your favorite job posting website and look at the job posting for say 10 or 20 data science jobs. Note down the keywords for the required and preferred skills listed in these posting. Separate these keywords into categories such as

- skills I already know/fluent
- skills I don't know but are interested in learning and
- I have no interests in ever acquiring these skills.

Bonus points if you can automate the above process.

b) If you are familiar with base R then take a look at this vignette. Next, try to solve either of problem 1 or problem 2 using base R. Compare and contrast your experience with using dplyr and using base **R**. NB. If you are not familiar with base R then fret not, we will learn more about base **R** as the semester progressed.