DATA 106 - Lab 2

Jillian Morrison

September 16, 2019

General rules

- For some questions, the needed methods may not have been covered in class. For them, please do some research to solve them.
- You must show your work in order to get points. Providing correct answers without supporting codes or intermediate steps does not receive full credit.
- You must submit both the R file as a .R file and the Assignment file as a PDF. For the Assignment file include the code, the output and explanations (if necesssary).

Questions

1. Using the `Cars93` dataset in the `MASS` package, do the following:

a. Create two new data frames for USA and non-USA cars. Name the new datasets `USA` and `Non` respectively. You can use the filter function in `{dplyr}` package.

```r
library(dplyr)
library(MASS)
#filter - selects entries whith specified origin
USA<-Cars93%>%filter(Origin=="USA")
Non<-Cars93%>%filter(!(Origin=="USA"))
```

b. Find the cheapest US car and non-US car. Here, consider using the `filter()` function along with `min()` and `max()` functions to find minimum and maximun values.

```r
##Cheapest Car
#filter - selects entry with minimum price
USA%>%filter(Price==min(Price))
```

```
##   Manufacturer   Model  Type Min.Price Price Max.Price MPG.city
## 1         Ford Festiva Small       6.9   7.4       7.9       31
##   MPG.highway AirBags DriveTrain Cylinders EngineSize Horsepower  RPM
## 1          33    None      Front         4        1.3         63 5000
##   Rev.per.mile Man.trans.avail Fuel.tank.capacity Passengers Length
## 1         3150             Yes                 10          4    141
##   Wheelbase Width Turn.circle Rear.seat.room Luggage.room Weight Origin
## 1        90    63          33             26           12   1845    USA
##           Make
## 1 Ford Festiva
```

```r
#Cheapest Non USA car
#filter - selects entry with minimum price
Non%>%filter(Price==min(Price))
```

```
##   Manufacturer Model  Type Min.Price Price Max.Price MPG.city MPG.highway
## 1      Hyundai Excel Small       6.8     8       9.2       29          33
##   AirBags DriveTrain Cylinders EngineSize Horsepower  RPM Rev.per.mile
## 1    None      Front         4        1.5         81 5500         2710
##   Man.trans.avail Fuel.tank.capacity Passengers Length Wheelbase Width
## 1             Yes               11.9          5    168        94    63
##   Turn.circle Rear.seat.room Luggage.room Weight  Origin          Make
## 1          35             26           11   2345 non-USA Hyundai Excel
```

c. find the most expensive USA and non-USA Car

```r
#Most expensive USA car
#filter - selects entry with maximum price
USA%>%filter(Price==max(Price))
```

```
##   Manufacturer   Model    Type Min.Price Price Max.Price MPG.city
## 1     Cadillac Seville Midsize      37.5  40.1      42.7       16
##   MPG.highway            AirBags DriveTrain Cylinders EngineSize
## 1          25 Driver & Passenger      Front         8        4.6
##   Horsepower  RPM Rev.per.mile Man.trans.avail Fuel.tank.capacity
## 1        295 6000         1985              No                 20
##   Passengers Length Wheelbase Width Turn.circle Rear.seat.room
## 1          5    204       111    74          44             31
##   Luggage.room Weight Origin             Make
## 1           14   3935    USA Cadillac Seville
```

```r
#Most expensive non-USA car
#filter - selects entry with maximum price
Non%>%filter(Price==max(Price))
```

```
##    Manufacturer Model    Type Min.Price Price Max.Price MPG.city
## 1 Mercedes-Benz  300E Midsize      43.8  61.9        80       19
##   MPG.highway            AirBags DriveTrain Cylinders EngineSize
## 1          25 Driver & Passenger       Rear         6        3.2
##   Horsepower  RPM Rev.per.mile Man.trans.avail Fuel.tank.capacity
## 1        217 5500         2220              No               18.5
##   Passengers Length Wheelbase Width Turn.circle Rear.seat.room
## 1          5    187       110    69          37             27
##   Luggage.room Weight  Origin               Make
## 1           15   3525 non-USA Mercedes-Benz 300E
```

d. The `Type` variable classifies the type of market the car is aimed at. Find the cheapest car in each type, and the car with the greatest fuel efficiency. (Hint: In part a, you separated by a specific variable and b and c, you filtered to find the cheapest car in each group. You will need to combine both in this part. However, instead of using filter to separate by a specific variable (part a), consider using group_by() in `{dplyr}`. You will also want to use piping (`%>%`) to make this easier.)

```
#group by Type - separates the dataset by the different Types
#filter - selects the minimum price for each of the different types we grouped by
Cars93%>%group_by(Type)%>%filter(Price==min(Price))
```

```
## # A tibble: 6 x 27
## # Groups:   Type [6]
##   Manufacturer Model Type  Min.Price Price Max.Price MPG.city MPG.highway
##   <fct>        <fct> <fct>     <dbl> <dbl>     <dbl>    <int>       <int>
## 1 Chevrolet    Lumi~ Van        14.7  16.3      18         18          23
## 2 Chrylser     Conc~ Large      18.4  18.4      18.4       20          28
## 3 Ford         Fest~ Small       6.9   7.4       7.9       31          33
## 4 Hyundai      Scou~ Spor~       9.1  10        11         26          34
## 5 Hyundai      Sona~ Mids~      12.4  13.9      15.3       20          27
## 6 Pontiac      Sunb~ Comp~       9.4  11.1      12.8       23          31
## # ... with 19 more variables: AirBags <fct>, DriveTrain <fct>,
## #   Cylinders <fct>, EngineSize <dbl>, Horsepower <int>, RPM <int>,
## #   Rev.per.mile <int>, Man.trans.avail <fct>, Fuel.tank.capacity <dbl>,
## #   Passengers <int>, Length <int>, Wheelbase <int>, Width <int>,
## #   Turn.circle <int>, Rear.seat.room <dbl>, Luggage.room <int>,
## #   Weight <int>, Origin <fct>, Make <fct>
```

e. Compute the mean horsepower for each type. (Hint: Still using piping (`%>%`), try using `group_by()` and `summarize()`. See: https://datacarpentry.org/R-genomics/04-dplyr.html for more info on summarize. Note: `na.rm=TRUE` removes missing values from the dataset before making calculations.).

```
#group by Type - separates the dataset by the different Types
#summarize - summarizes (in this case the mean and standard deviation) by the different types we groupe
Summary<-Cars93%>%group_by(Type)%>%summarize(mean=mean(Horsepower), sd=sd(Horsepower))
Summary
```

```
## # A tibble: 6 x 3
##   Type     mean    sd
##   <fct>   <dbl> <dbl>
## 1 Compact  131   22.8
## 2 Large    179.  21.8
## 3 Midsize  173.  52.5
## 4 Small     91   21.2
## 5 Sporty   160.  74.4
## 6 Van      149.  19.2
```

f. Save the resulting table in part e to a .csv file called `Summary.csv`. You will upoad this file to moodle along with your R script and pdf. (Hint: Remember to set your working directory so you know where your file is saved. Also make sure that you save your table as on object in R so you can save it to a csv.)

```
#setwd("C:/Users/jmorrison/OneDrive - The College of Wooster/College of Wooster/Fall2019/Data Analytics
#write.csv(Summary, "Summary.csv")
```

2.Using the `gapminder` dataset from the `{gapminder}` package, do the following:

a. Save the dataset to an object called "gap" and convert it to a dataframe

```
#install.packages("gapminder")
library("gapminder")
gap<-data.frame(gapminder)
```

b. How many different countries are covered by the data. List them.

```
#Counting the unique countries
length(unique(gap$country))
```

```
## [1] 142
```

```
#Listing the unique countries
unique(gap$country)
```

```
##   [1] Afghanistan              Albania
##   [3] Algeria                  Angola
##   [5] Argentina                Australia
##   [7] Austria                  Bahrain
##   [9] Bangladesh               Belgium
##  [11] Benin                    Bolivia
##  [13] Bosnia and Herzegovina   Botswana
##  [15] Brazil                   Bulgaria
##  [17] Burkina Faso             Burundi
##  [19] Cambodia                 Cameroon
##  [21] Canada                   Central African Republic
##  [23] Chad                     Chile
##  [25] China                    Colombia
##  [27] Comoros                  Congo, Dem. Rep.
##  [29] Congo, Rep.              Costa Rica
##  [31] Cote d'Ivoire            Croatia
##  [33] Cuba                     Czech Republic
##  [35] Denmark                  Djibouti
##  [37] Dominican Republic       Ecuador
##  [39] Egypt                    El Salvador
##  [41] Equatorial Guinea        Eritrea
##  [43] Ethiopia                 Finland
##  [45] France                   Gabon
##  [47] Gambia                   Germany
##  [49] Ghana                    Greece
##  [51] Guatemala                Guinea
##  [53] Guinea-Bissau            Haiti
##  [55] Honduras                 Hong Kong, China
##  [57] Hungary                  Iceland
##  [59] India                    Indonesia
##  [61] Iran                     Iraq
##  [63] Ireland                  Israel
##  [65] Italy                    Jamaica
##  [67] Japan                    Jordan
##  [69] Kenya                    Korea, Dem. Rep.
##  [71] Korea, Rep.              Kuwait
##  [73] Lebanon                  Lesotho
##  [75] Liberia                  Libya
##  [77] Madagascar               Malawi
##  [79] Malaysia                 Mali
##  [81] Mauritania               Mauritius
##  [83] Mexico                   Mongolia
##  [85] Montenegro               Morocco
##  [87] Mozambique               Myanmar
##  [89] Namibia                  Nepal
##  [91] Netherlands              New Zealand
##  [93] Nicaragua                Niger
##  [95] Nigeria                  Norway
##  [97] Oman                     Pakistan
##  [99] Panama                   Paraguay
## [101] Peru                     Philippines
## [103] Poland                   Portugal
## [105] Puerto Rico              Reunion
## [107] Romania                  Rwanda
## [109] Sao Tome and Principe    Saudi Arabia
## [111] Senegal                  Serbia
## [113] Sierra Leone             Singapore
## [115] Slovak Republic          Slovenia
## [117] Somalia                  South Africa
## [119] Spain                    Sri Lanka
## [121] Sudan                    Swaziland
## [123] Sweden                   Switzerland
## [125] Syria                    Taiwan
## [127] Tanzania                 Thailand
## [129] Togo                     Trinidad and Tobago
## [131] Tunisia                  Turkey
## [133] Uganda                   United Kingdom
## [135] United States            Uruguay
## [137] Venezuela                Vietnam
## [139] West Bank and Gaza       Yemen, Rep.
## [141] Zambia                   Zimbabwe
## 142 Levels: Afghanistan Albania Algeria Angola Argentina ... Zimbabwe
```

c. Extract all the 2002 life expectancies for African countries

(Note: the `select()` function is available in both `{dplyr}` and `{MASS}` packages. To specify you want to use the `{dplyr}` package, use instead `dplyr::select()`)

(Other Note: here you have 2 conditions - Africa and 2002)

```r
#filter - selects the dataset that satisfies both Africa and 2002
#select - pulls only the country and lifeExp columns
gap%>%filter(continent=="Africa"& year=="2002")%>%dplyr::select(country, lifeExp)
```

```
##                      country lifeExp
## 1                    Algeria  70.994
## 2                     Angola  41.003
## 3                      Benin  54.406
## 4                   Botswana  46.634
## 5               Burkina Faso  50.650
## 6                    Burundi  47.360
## 7                   Cameroon  49.856
## 8  Central African Republic  43.308
## 9                      Chad  50.525
## 10                  Comoros  62.974
## 11         Congo, Dem. Rep.  44.966
## 12              Congo, Rep.  52.970
## 13            Cote d'Ivoire  46.832
## 14                 Djibouti  53.373
## 15                    Egypt  69.806
## 16        Equatorial Guinea  49.348
## 17                  Eritrea  55.240
## 18                 Ethiopia  50.725
## 19                    Gabon  56.761
## 20                   Gambia  58.041
## 21                    Ghana  58.453
## 22                   Guinea  53.676
## 23            Guinea-Bissau  45.504
## 24                    Kenya  50.992
## 25                  Lesotho  44.593
## 26                  Liberia  43.753
## 27                    Libya  72.737
## 28               Madagascar  57.286
## 29                   Malawi  45.009
## 30                     Mali  51.818
## 31               Mauritania  62.247
## 32                Mauritius  71.954
## 33                  Morocco  69.615
## 34               Mozambique  44.026
## 35                  Namibia  51.479
## 36                    Niger  54.496
## 37                  Nigeria  46.608
## 38                  Reunion  75.744
## 39                   Rwanda  43.413
## 40    Sao Tome and Principe  64.337
## 41                  Senegal  61.600
## 42             Sierra Leone  41.012
## 43                  Somalia  45.936
## 44             South Africa  53.365
## 45                    Sudan  56.369
## 46                Swaziland  43.869
## 47                 Tanzania  49.651
## 48                     Togo  57.561
## 49                  Tunisia  73.042
## 50                   Uganda  47.813
## 51                   Zambia  39.193
## 52                 Zimbabwe  39.989
```

```
##OR

o<-gap%>%filter(continent=="Africa"& year=="2002")
dplyr::select(o, country, lifeExp)
```

```
##                     country lifeExp
## 1                   Algeria  70.994
## 2                    Angola  41.003
## 3                      Benin  54.406
## 4                   Botswana  46.634
## 5               Burkina Faso  50.650
## 6                    Burundi  47.360
## 7                   Cameroon  49.856
## 8   Central African Republic  43.308
## 9                       Chad  50.525
## 10                   Comoros  62.974
## 11          Congo, Dem. Rep.  44.966
## 12               Congo, Rep.  52.970
## 13             Cote d'Ivoire  46.832
## 14                  Djibouti  53.373
## 15                     Egypt  69.806
## 16         Equatorial Guinea  49.348
## 17                   Eritrea  55.240
## 18                  Ethiopia  50.725
## 19                     Gabon  56.761
## 20                    Gambia  58.041
## 21                     Ghana  58.453
## 22                    Guinea  53.676
## 23             Guinea-Bissau  45.504
## 24                     Kenya  50.992
## 25                   Lesotho  44.593
## 26                   Liberia  43.753
## 27                     Libya  72.737
## 28                Madagascar  57.286
## 29                    Malawi  45.009
## 30                      Mali  51.818
## 31                Mauritania  62.247
## 32                 Mauritius  71.954
## 33                   Morocco  69.615
## 34                Mozambique  44.026
## 35                   Namibia  51.479
## 36                     Niger  54.496
## 37                   Nigeria  46.608
## 38                   Reunion  75.744
## 39                    Rwanda  43.413
## 40     Sao Tome and Principe  64.337
## 41                   Senegal  61.600
## 42              Sierra Leone  41.012
## 43                   Somalia  45.936
## 44              South Africa  53.365
## 45                     Sudan  56.369
## 46                 Swaziland  43.869
## 47                  Tanzania  49.651
## 48                      Togo  57.561
## 49                   Tunisia  73.042
## 50                    Uganda  47.813
## 51                    Zambia  39.193
## 52                  Zimbabwe  39.989
```

```
#Checking the unique entries of year
#unique(gap$year)
```

d. Extract the 2005 population for African countries

```
#filter - selects the dataset that satisfies both Africa and 2005
#select - pulls only the country and population columns
gap%>%filter(continent=="Africa"& year=="2005")%>%dplyr::select(country, pop)
```

```
## [1] country pop
## <0 rows> (or 0-length row.names)
```

```
#Checking the unique entries of year
#unique(gap$year)
```

There is no data for 2005.

e. Extract the country with the highest gdp value for 2007 for each continent.

```
#Filter by year - gets the dataset but only for year 2007
#group by continent - separates the 2007 dataset by the different continents
#filter gdp per capita - chooses the maximum gdp within each group of continent.
one<-gap%>%filter(year=="2007")%>%group_by(continent)%>%filter(gdpPercap==max(gdpPercap))
one
```

```
## # A tibble: 5 x 6
## # Groups:   continent [5]
##   country       continent  year lifeExp       pop gdpPercap
##   <fct>         <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Australia     Oceania    2007    81.2  20434176    34435.
## 2 Gabon         Africa     2007    56.7   1454867    13206.
## 3 Kuwait        Asia       2007    77.6   2505559    47307.
## 4 Norway        Europe     2007    80.2   4627926    49357.
## 5 United States Americas   2007    78.2 301139947    42952.
```

```
#Checking the unique continents
#unique(gap$continent)
#gap%>%group_by(continent)%>%slice(1)
```