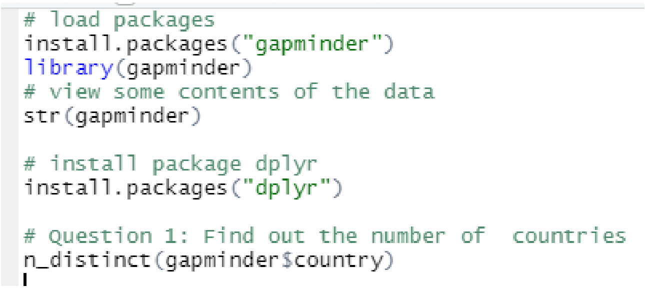
PHP2510 Homework 2

Kuan-Min Lee

Question 1:

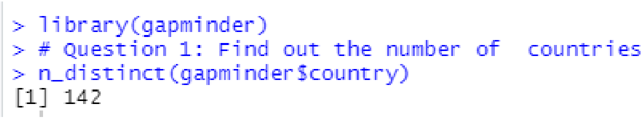
The following codes are implemented to generate the outcomes:



```
# load packages
install.packages("gapminder")
library(gapminder)
# view some contents of the data
str(gapminder)

# install package dplyr
install.packages("dplyr")

# Question 1: Find out the number of  countries
n_distinct(gapminder$country)
```



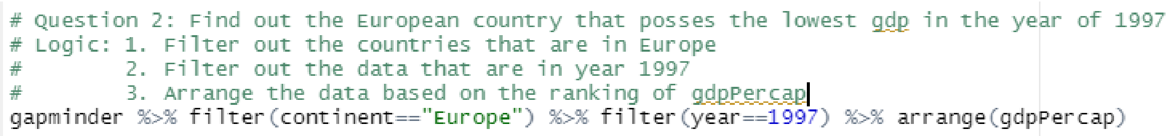
```
> library(gapminder)
> # Question 1: Find out the number of  countries
> n_distinct(gapminder$country)
[1] 142
```

Figure 1: Snapshot of the Data Summarization

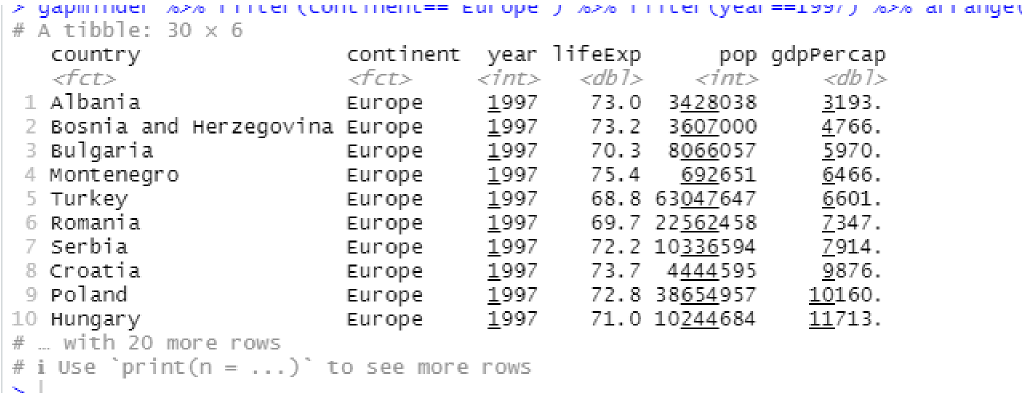
There are 142 countries in the dataset

Question 2:

The following codes are implemented to generate the outcomes. The comments of that are the logic behind this section of code implementation:



```
# Question 2: Find out the European country that posses the lowest gdp in the year of 1997
# Logic: 1. Filter out the countries that are in Europe
#        2. Filter out the data that are in year 1997
#        3. Arrange the data based on the ranking of gdpPercap
gapminder %>% filter(continent=="Europe") %>% filter(year==1997) %>% arrange(gdpPercap)
```



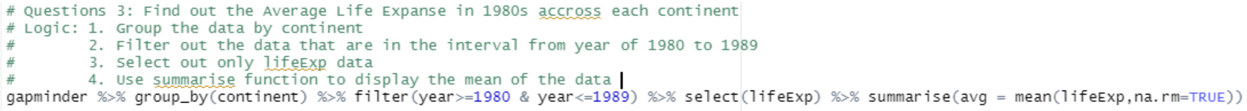
```
> gapminder %>% filter(continent=="Europe") %>% filter(year==1997) %>% arrange(
# A tibble: 30 × 6
   country                continent  year lifeExp      pop gdpPercap
   <fct>                  <fct>     <int>   <dbl>    <int>     <dbl>
 1 Albania                Europe     1997    73.0  3428038     3193.
 2 Bosnia and Herzegovina Europe     1997    73.2  3607000     4766.
 3 Bulgaria               Europe     1997    70.3  8066057     5970.
 4 Montenegro             Europe     1997    75.4   692651     6466.
 5 Turkey                 Europe     1997    68.8 63047647     6601.
 6 Romania                Europe     1997    69.7 22562458     7347.
 7 Serbia                 Europe     1997    72.2 10336594     7914.
 8 Croatia                Europe     1997    73.7  4444595     9876.
 9 Poland                 Europe     1997    72.8 38654957    10160.
10 Hungary                Europe     1997    71.0 10244684    11713.
# … with 20 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Figure 2: Snapshot of the Data of Arranged GDP

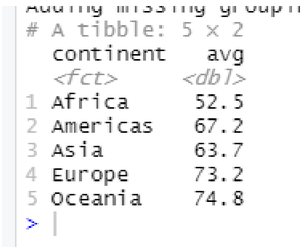
The country has the lowest GDP in 1997 is Albania with GDP per cap as 3193 dollars.

Question 3:

The following codes are implemented to generate the outcomes of the question. The comment section contains the logic behind this section of code:



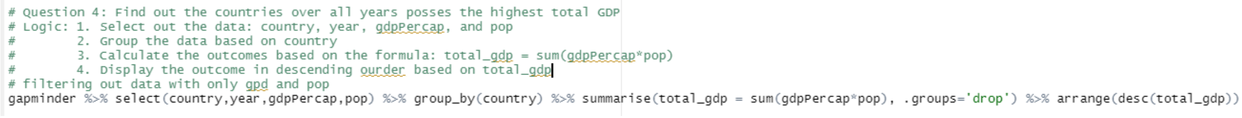
```
# Questions 3: Find out the Average Life Expanse in 1980s accross each continent
# Logic: 1. Group the data by continent
#        2. Filter out the data that are in the interval from year of 1980 to 1989
#        3. Select out only lifeExp data
#        4. Use summarise function to display the mean of the data
gapminder %>% group_by(continent) %>% filter(year>=1980 & year<=1989) %>% select(lifeExp) %>% summarise(avg = mean(lifeExp,na.rm=TRUE))
```



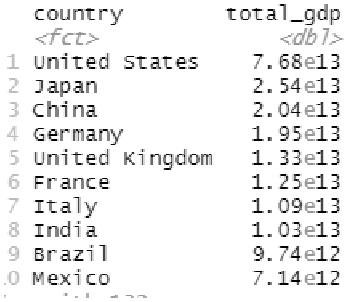
```
Adding missing grouping
# A tibble: 5 × 2
  continent   avg
  <fct>     <dbl>
1 Africa     52.5
2 Americas   67.2
3 Asia       63.7
4 Europe     73.2
5 Oceania    74.8
> 
```

Figure 3: Snapshot of the Table of Average Life Expend for Each Continent

Question 4:



```
# Question 4: Find out the countries over all years posses the highest total GDP
# Logic: 1. Select out the data: country, year, gdpPercap, and pop
#        2. Group the data based on country
#        3. Calculate the outcomes based on the formula: total_gdp = sum(gdpPercap*pop)
#        4. Display the outcome in descending ourder based on total_gdp
# filtering out data with only gpd and pop
gapminder %>% select(country,year,gdpPercap,pop) %>% group_by(country) %>% summarise(total_gdp = sum(gdpPercap*pop), .groups='drop') %>% arrange(desc(total_gdp))
```



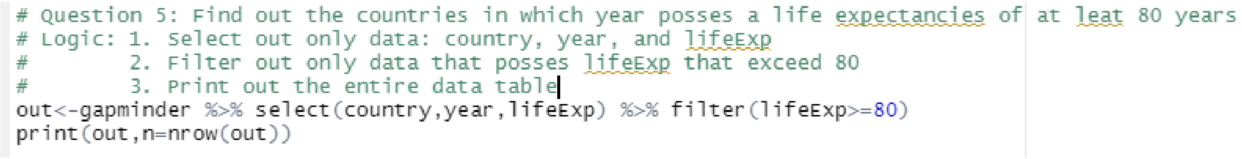
```
   country         total_gdp
   <fct>               <dbl>
 1 United States     7.68e13
 2 Japan             2.54e13
 3 China             2.04e13
 4 Germany           1.95e13
 5 United Kingdom    1.33e13
 6 France            1.25e13
 7 Italy             1.09e13
 8 India             1.03e13
 9 Brazil            9.74e12
10 Mexico            7.14e12
```

Figure 4: Snapshot of the Table of Total GDP

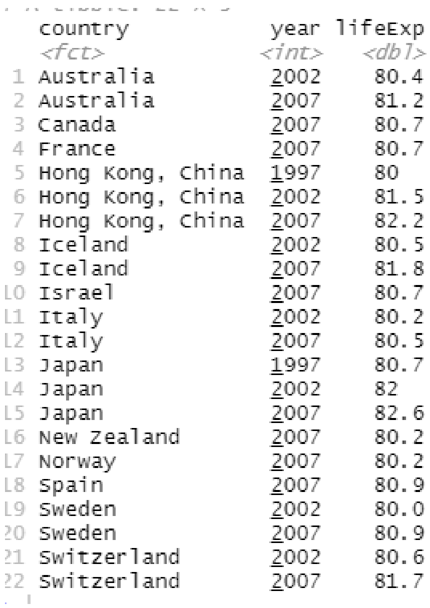
Top 5 Countries are: United States, Japan, China, Germany, United Kingdom

Question 5:

The following codes are implemented to generate the outcomes. The comments contain the logic behind the code:



```
# Question 5: Find out the countries in which year posses a life expectancies of at leat 80 years
# Logic: 1. Select out only data: country, year, and lifeExp
#        2. Filter out only data that posses lifeExp that exceed 80
#        3. Print out the entire data table
out<-gapminder %>% select(country,year,lifeExp) %>% filter(lifeExp>=80)
print(out,n=nrow(out))
```



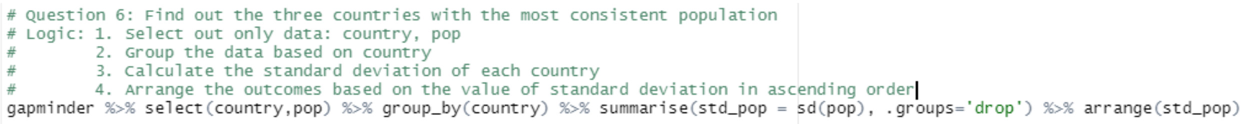
```
   country            year lifeExp
   <fct>             <int>   <dbl>
 1 Australia          2002    80.4
 2 Australia          2007    81.2
 3 Canada             2007    80.7
 4 France             2007    80.7
 5 Hong Kong, China   1997    80
 6 Hong Kong, China   2002    81.5
 7 Hong Kong, China   2007    82.2
 8 Iceland            2002    80.5
 9 Iceland            2007    81.8
10 Israel             2007    80.7
11 Italy              2002    80.2
12 Italy              2007    80.5
13 Japan              1997    80.7
14 Japan              2002    82
15 Japan              2007    82.6
16 New Zealand        2007    80.2
17 Norway             2007    80.2
18 Spain              2007    80.9
19 Sweden             2002    80.0
20 Sweden             2007    80.9
21 Switzerland        2002    80.6
22 Switzerland        2007    81.7
```

Figure 5: Snapshot of the Complete Table of Countries that Have Life Expand of at Least 80 Years

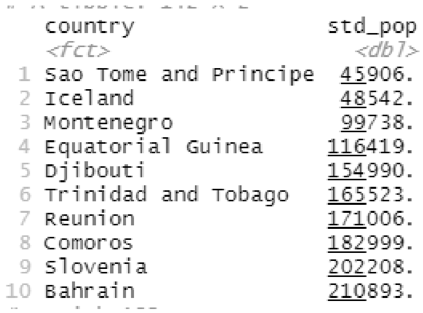
In total, there are 22 circumstances in this case.

Question 6:

The following codes are implemented to generate the outcomes. The comments are the logics behind this implementation:



```
# Question 6: Find out the three countries with the most consistent population
# Logic: 1. Select out only data: country, pop
#        2. Group the data based on country
#        3. Calculate the standard deviation of each country
#        4. Arrange the outcomes based on the value of standard deviation in ascending order
gapminder %>% select(country,pop) %>% group_by(country) %>% summarise(std_pop = sd(pop), .groups='drop') %>% arrange(std_pop)
```



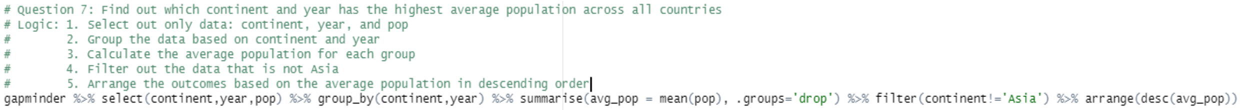
```
   country                std_pop
   <fct>                    <dbl>
 1 Sao Tome and Principe   45906.
 2 Iceland                 48542.
 3 Montenegro              99738.
 4 Equatorial Guinea      116419.
 5 Djibouti               154990.
 6 Trinidad and Tobago    165523.
 7 Reunion                171006.
 8 Comoros                182999.
 9 Slovenia               202208.
10 Bahrain                210893.
```

Figure 6: Snapshot of the Table of Population Standard Deviation Showed in Ascending Order

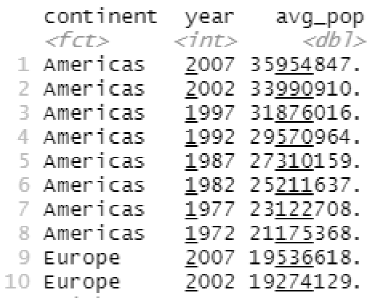
From the table, the top 3 are: Sao Tome and Principe, Iceland, and Montenegro

Question 7:

The following codes are utilized to generate the intended outcomes, and the comments are the logics behind this:



```
# Question 7: Find out which continent and year has the highest average population across all countries
# Logic: 1. Select out only data: continent, year, and pop
#        2. Group the data based on continent and year
#        3. Calculate the average population for each group
#        4. Filter out the data that is not Asia
#        5. Arrange the outcomes based on the average population in descending order
gapminder %>% select(continent,year,pop) %>% group_by(continent,year) %>% summarise(avg_pop = mean(pop), .groups='drop') %>% filter(continent!='Asia') %>% arrange(desc(avg_pop))
```



```
   continent  year   avg_pop
   <fct>     <int>     <dbl>
 1 Americas   2007 35954847.
 2 Americas   2002 33990910.
 3 Americas   1997 31876016.
 4 Americas   1992 29570964.
 5 Americas   1987 27310159.
 6 Americas   1982 25211637.
 7 Americas   1977 23122708.
 8 Americas   1972 21175368.
 9 Europe     2007 19536618.
10 Europe     2002 19274129.
```

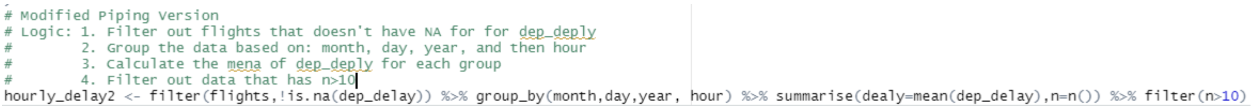
Figure 7: Snapshot of the Table of Average Population in Descending Order

From the table, Americas in 2007 has the highest average population for each country.

Question 8:

(a) The code is nested, and it nested inside a series of functions. It will be very difficult for the code reader to follow the logic behind this since it needs to be read from the very inner one onto the outer one.

(b)



```
# Modified Piping Version
# Logic: 1. Filter out flights that doesn't have NA for for dep_deply
#        2. Group the data based on: month, day, year, and then hour
#        3. Calculate the mena of dep_deply for each group
#        4. Filter out data that has n>10
hourly_delay2 <- filter(flights,!is.na(dep_delay)) %>% group_by(month,day,year, hour) %>% summarise(dealy=mean(dep_delay),n=n()) %>% filter(n>10)
```

```
hourly_delay2 <- filter(flights,!is.na(dep_delay)) %>%
group_by(month,day,year, hour) %>%
summarise(dealy=mean(dep_delay),n=n()) %>% filter(n>10)
```

Appendix: Source Code of the Homework

```
# load packages
install.packages("gapminder")
library(gapminder)
# view some contents of the data
str(gapminder)

# install package dplyr
install.packages("dplyr")

# Question 1: Find out the number of  countries
n_distinct(gapminder$country)

# Question 2: Find out the European country that posses the lowest gdp in the year of 1997
# Logic: 1. Filter out the countries that are in Europe
#        2. Filter out the data that are in year 1997
#        3. Arrange the data based on the ranking of gdpPercap
gapminder %>% filter(continent=="Europe") %>% filter(year==1997) %>% arrange(gdpPercap)

# Questions 3: Find out the Average Life Expanse in 1980s accross each continent
# Logic: 1. Group the data by continent
#        2. Filter out the data that are in the interval from year of 1980 to 1989
#        3. Select out only lifeExp data
#        4. Use summarise function to display the mean of the data
gapminder %>% group_by(continent) %>% filter(year>=1980 & year<=1989) %>%
select(lifeExp) %>% summarise(avg = mean(lifeExp,na.rm=TRUE))
```

```
# Question 4: Find out the countries over all years posses the highest total GDP
# Logic: 1. Select out the data: country, year, gdpPercap, and pop
#        2. Group the data based on country
#        3. Calculate the outcomes based on the formula: total_gdp = sum(gdpPercap*pop)
#        4. Display the outcome in descending ourder based on total_gdp
gapminder %>% select(country,year,gdpPercap,pop) %>% group_by(country) %>%
summarise(total_gdp = sum(gdpPercap*pop), .groups='drop') %>% arrange(desc(total_gdp))
```

```
# Question 5: Find out the countries in which year posses a life expectancies of at leat 80 years
# Logic: 1. Select out only data: country, year, and lifeExp
#        2. Filter out only data that posses lifeExp that exceed 80
#        3. Print out the entire data table
out<-gapminder %>% select(country,year,lifeExp) %>% filter(lifeExp>=80)
print(out,n=nrow(out))
```

```
# Question 6: Find out the three countries with the most consistent population
# Logic: 1. Select out only data: country, pop
#        2. Group the data based on country
#        3. Calculate the standard deviation of each country
#        4. Arrange the outcomes based on the value of standard deviation in ascending order
gapminder %>% select(country,pop) %>% group_by(country) %>% summarise(std_pop =
sd(pop), .groups='drop') %>% arrange(std_pop)
```

```
# Question 7: Find out which continent and year has the highest average population across all countries

# Logic: 1. Select out only data: continent, year, and pop

#        2. Group the data based on continent and year

#        3. Calculate the average population for each group

#        4. Filter out the data that is not Asia

#        5. Arrange the outcomes based on the average population in descending order

gapminder %>% select(continent,year,pop) %>% group_by(continent,year) %>% summarise(avg_pop = mean(pop), .groups='drop') %>% filter(continent!='Asia') %>% arrange(desc(avg_pop))


# Question 8

install.packages("nycflights13")

library(nycflights13)

# Original Code from Manual

hourly_delay <- filter(

  summarise(

    group_by(

      filter(

        flights,

        !is.na(dep_delay)

      ),

      month, day, year, hour

    ),

    delay=mean(dep_delay),

    n=n()

  ),

  n>10

)
```

```
# Modified Piping Version
# Logic: 1. Filter out flights that doesn't have NA for for dep_deply
#        2. Group the data based on: month, day, year, and then hour
#        3. Calculate the mena of dep_deply for each group
#        4. Filter out data that has n>10
hourly_delay2 <- filter(flights,!is.na(dep_delay)) %>% group_by(month,day,year, hour) %>%
summarise(dealy=mean(dep_delay),n=n()) %>% filter(n>10)
```