MATH4068 Coursework

SID 20490457

2023-05-04

Project basic setup

```
# setup my working directory
knitr::opts_knit$set(root.dir = "/Users/davidhui/Documents/GitHub/uon/MATH4068 Applied Multivariate Sta

# centering figure
knitr::opts_chunk$set(fig.align = 'center')
```

```
.caption {
  margin: auto;
  text-align: center;
}
```

```
# import some commonly used libraries first
library("ggplot2")
library("plotly")
library("GGally")
library("tidyverse")
library("reshape2")
library("ggfortify")
library("dplyr")
```

Exploratory data analysis

This task is to analyse the dataset that contains the GDP per capita, and the life expectancy for 142 different countries from 1952 to 2007 (for every 5 years).

From the given introduction, for analysis purposes, I will work on **log** scale for GDP per capita, and original scale for the life expectancy.

```
gap.raw = read.csv("materials/gap.csv") # read the csv
gap = gap.raw
gap[,3:14] = log(gap.raw[,3:14])
```

We can make some simple plots to show how GDP and life expectancy have changed over the past 70 years. Because there are 142 different countries in the dataset, plotting all countries one by one with colors in the same plot is definitely messy. So here I calculate the mean GDP and mean life expectancy each year, and take a look at the mean changes.

```
# For plotting purposes only

# similar to the given instructions, but I have made some revise
gdp = gap[,3:14] # I am not going to directly follow the instruction, I will remain using log-scale to
```

```r
years = seq(1952, 2007,5)
gdp = cbind(gap[,1:2], gdp)
colnames(gdp) = c("continent", "country", years)

lifeExp = gap[,15:26]
lifeExp = cbind(gap[,1:2], lifeExp)
colnames(lifeExp) = c("continent", "country", years)


# prepare a dataframe for plotting:
# keep only year data
gdp_mean = t(as.data.frame(t(colMeans(gdp[,3:14]))))
gdp_mean = melt(gdp_mean,"Row.names")[,c(1,3)]
colnames(gdp_mean) = c("year", "log_gdp")

# prepare a dataframe for plotting:
# keep only year data
lifeExp_mean = t(as.data.frame(t(colMeans(lifeExp[,3:14]))))
lifeExp_mean = melt(lifeExp_mean, "Row.names")[,c(1,3)]
colnames(lifeExp_mean) = c("year", "lifeExp")
```

```r
plot(log_gdp~year, data = gdp_mean, type="l", main="mean log(GDP) against year", ylab="log(GDP)")
```

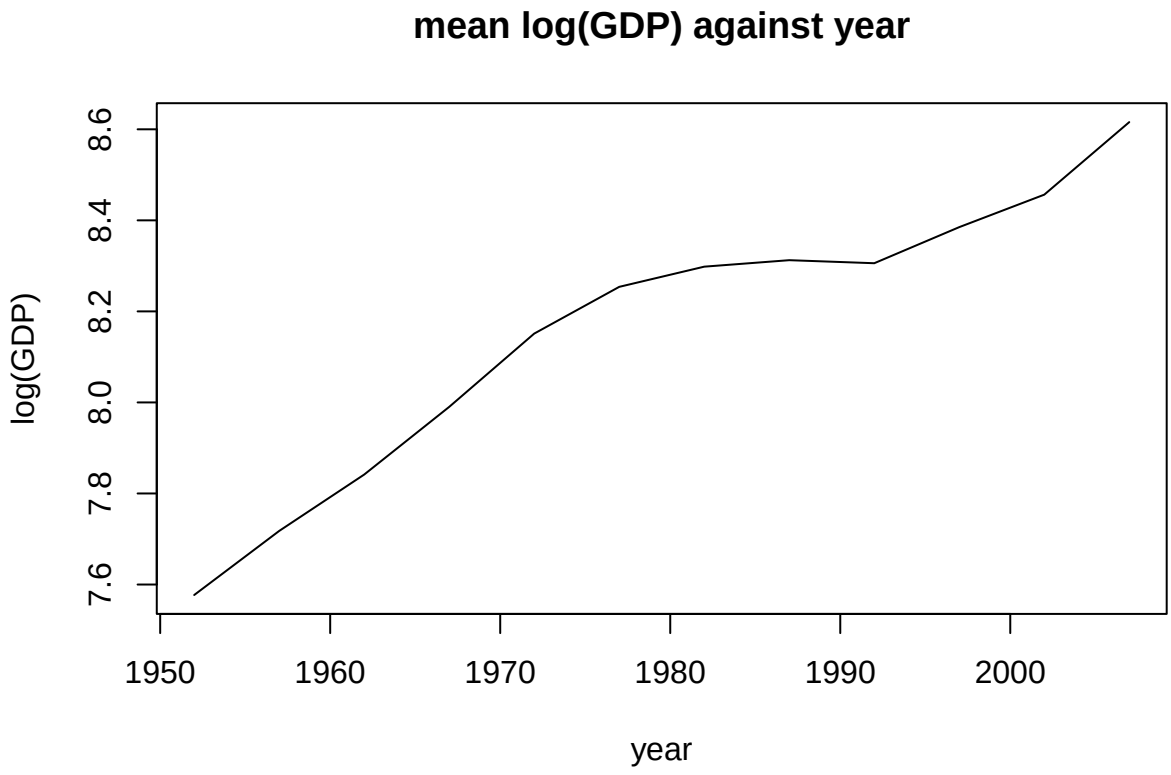


Figure 1: mean log(GDP) against year

```r
plot(lifeExp~year, data=lifeExp_mean, type="l", main="mean Life expectancy against year", ylab="lifeExp
```

From both simple plots, we can obviously see that the mean GDP is increasing as year increases. This is also true for mean life expectancy.

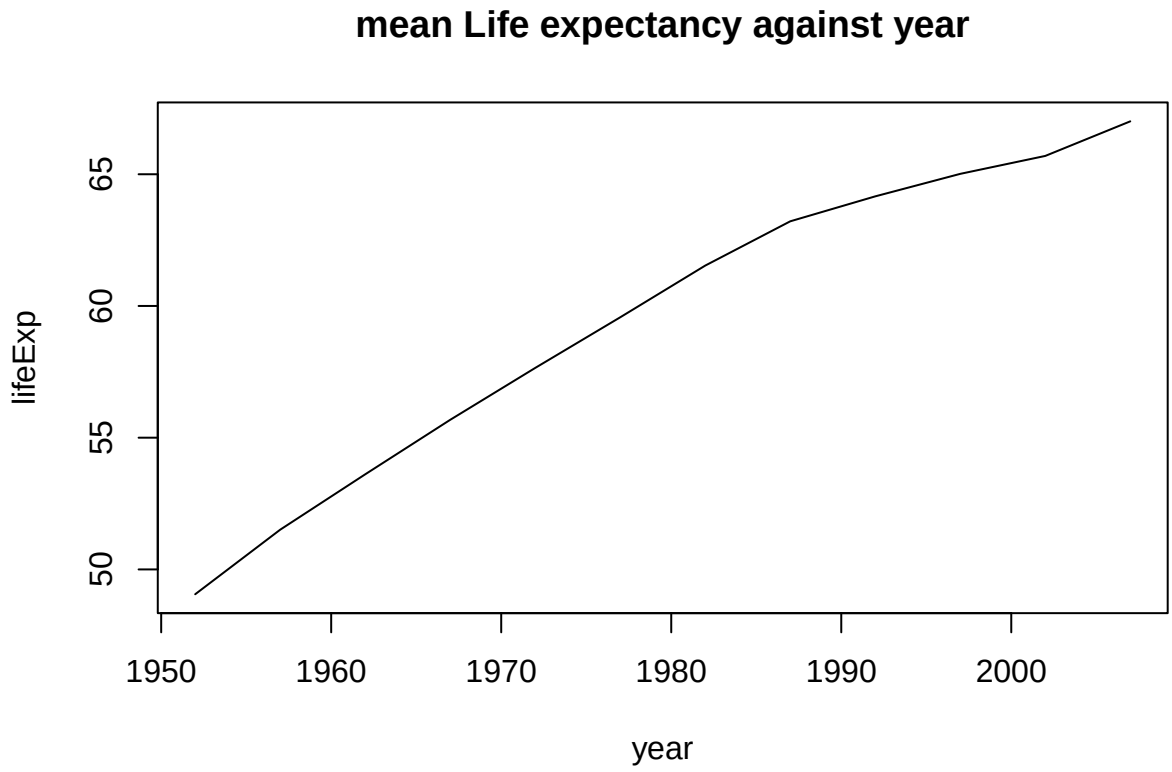


Figure 2: mean Life expectancy vs year

Because mean is sensitive to outlier, so simply look at the change by taking mean across all countries may miss some information. So instance, we may take a look at the mean across different continents.

```
gdp_continent_mean = gdp %>%
              select(! country) %>%
              group_by(continent) %>%
              summarise(across(everything(), mean), .groups = "drop") %>%
              as.data.frame(.)

# there are 5 continents
rownames(gdp_continent_mean) = gdp_continent_mean[,1]
gdp_continent_mean = gdp_continent_mean[,-1]

# prepare a dataframe for plottingg:
gdp_continent_mean = as.data.frame(t(gdp_continent_mean))
gdp_continent_mean$year = rownames(gdp_continent_mean)
#gdp_continent_mean

# now there are 5 continents columns; with last column as year
# to avoid over-engineering, I will just use these to make plot

min_gdp_continent_mean = min(as.matrix(gdp_continent_mean[,1:5]))
max_gdp_continent_mean = max(as.matrix(gdp_continent_mean[,1:5]))


plot(Africa~year, data = gdp_continent_mean,
     main="Continent mean log(GDP) vs year",
```

```
     ylab="log(GDP)", col=1, type="l",
     ylim=c(min_gdp_continent_mean/2, max_gdp_continent_mean))
lines(Americas~year, data = gdp_continent_mean, col=2)
lines(Asia~year, data = gdp_continent_mean, col=3)
lines(Europe~year, data = gdp_continent_mean, col=4)
lines(Oceania~year, data = gdp_continent_mean, col=5)

legend(x="bottomright", legend=c("Africa", "Americas", "Asia", "Eruope", "Oceania"),
       col=1:5,
       lty=1)
```

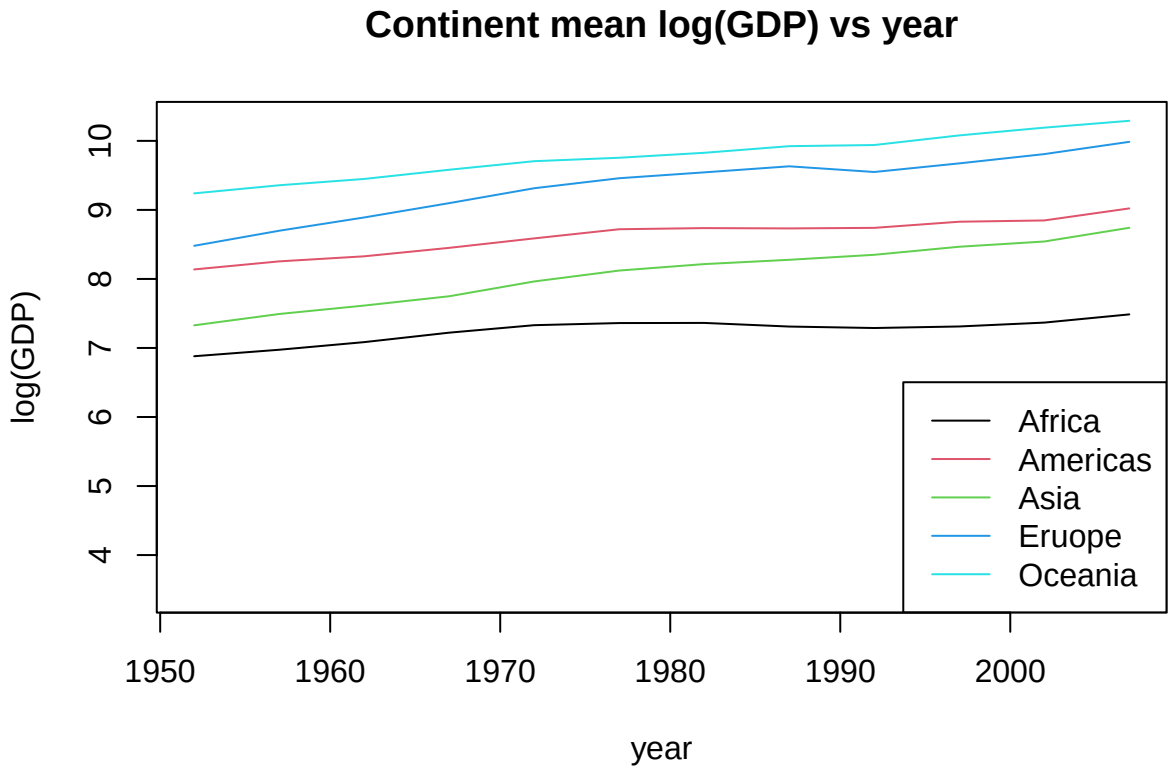


Figure 3: Continent mean log(GDP) vs year

```
lifeExp_continent_mean = lifeExp %>%
              select(! country) %>%
              group_by(continent) %>%
              summarise(across(everything(), mean), .groups = "drop") %>%
              as.data.frame(.)

# there are 5 continents
rownames(lifeExp_continent_mean) = lifeExp_continent_mean[,1]
lifeExp_continent_mean = lifeExp_continent_mean[,-1]

# prepare a dataframe for plotting:
lifeExp_continent_mean = as.data.frame(t(lifeExp_continent_mean))
lifeExp_continent_mean$year = rownames(lifeExp_continent_mean)
#lifeExp_continent_mean

# now there are 5 continents columns; with last column as year
```

```r
# to avoid over-engineering, I will just use these to make plot

min_lifeExp_continent_mean = min(as.matrix(lifeExp_continent_mean[,1:5]))
max_lifeExp_continent_mean = max(as.matrix(lifeExp_continent_mean[,1:5]))


plot(Africa~year, data = lifeExp_continent_mean,
     main="Continent mean Life Expectancy vs year",
     ylab="lifeExp", col=1, type="l",
     ylim=c(min_lifeExp_continent_mean/2, max_lifeExp_continent_mean))
lines(Americas~year, data = lifeExp_continent_mean, col=2)
lines(Asia~year, data = lifeExp_continent_mean, col=3)
lines(Europe~year, data = lifeExp_continent_mean, col=4)
lines(Oceania~year, data = lifeExp_continent_mean, col=5)

legend(x="bottomright", legend=c("Africa", "Americas", "Asia", "Eruope", "Oceania"),
       col=1:5,
       lty=1)
```

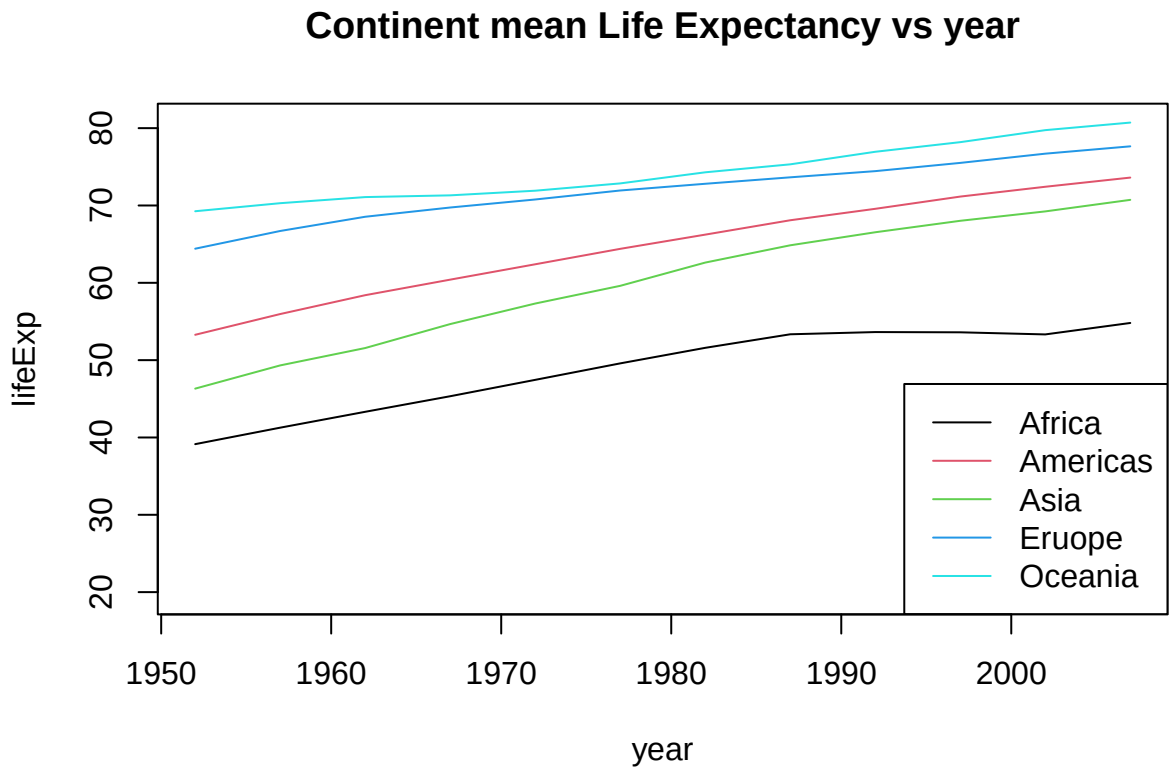


Figure 4: Continent mean life expectancy vs year

From both plots, in general, we can still identify a increasing trend for each continents, GDP and life expectancy are increasing by year in general.

Data Analysis

Next I will carry out the multivariate analysis according to the coursework requirement. So I will stick with the given data frames requirements, but making GDP in log-scale

```r
gdp <- gap[,3:14] # no need to take back exp(...), keep using log-scale
years <- seq(1952, 2007,5)
colnames(gdp) <- years
rownames(gdp) <- gap[,2]
lifeExp <- gap[,15:26]
colnames(lifeExp) <- years
rownames(lifeExp) <- gap[,2]
```

Principal Component Analysis (PCA)

This section will cover the use of PCA on the log(GDP) and on the life-expectancy data.

PCA is a useful tool for dimension reduction. This method aim at transforming the raw dataset, so we can use smaller number of variables to explain most of the variability of the dataset.

We can perform PCA on either sample covariance matrix **S** or the sample correlation matrix **R**. In general, we will use PCA based on R rather than S if the scales of the variables differs a lot. So before making any choice, first take a look at the sample covariance matrix first.

```r
cov(gdp)
```

```
##          1952     1957     1962     1967     1972     1977     1982     1987
## 1952 1.072040 1.095016 1.105962 1.123522 1.161495 1.163183 1.140939 1.150637
## 1957 1.095016 1.130072 1.145562 1.169295 1.212565 1.217069 1.197002 1.208559
## 1962 1.105962 1.145562 1.175347 1.209500 1.256451 1.265524 1.247702 1.259302
## 1967 1.123522 1.169295 1.209500 1.266375 1.319665 1.331957 1.316279 1.329465
## 1972 1.161495 1.212565 1.256451 1.319665 1.398623 1.416035 1.403404 1.422402
## 1977 1.163183 1.217069 1.265524 1.331957 1.416035 1.463876 1.458415 1.482573
## 1982 1.140939 1.197002 1.247702 1.316279 1.403404 1.458415 1.478826 1.512833
## 1987 1.150637 1.208559 1.259302 1.329465 1.422402 1.482573 1.512833 1.569160
## 1992 1.150938 1.203997 1.251459 1.321714 1.421097 1.481418 1.511967 1.577363
## 1997 1.148430 1.200641 1.250821 1.323253 1.421463 1.488542 1.521711 1.595212
## 2002 1.139307 1.192138 1.245796 1.319359 1.413837 1.485600 1.522338 1.600209
## 2007 1.146708 1.199793 1.253755 1.327363 1.418140 1.491849 1.528763 1.609362
##          1992     1997     2002     2007
## 1952 1.150938 1.148430 1.139307 1.146708
## 1957 1.203997 1.200641 1.192138 1.199793
## 1962 1.251459 1.250821 1.245796 1.253755
## 1967 1.321714 1.323253 1.319359 1.327363
## 1972 1.421097 1.421463 1.413837 1.418140
## 1977 1.481418 1.488542 1.485600 1.491849
## 1982 1.511967 1.521711 1.522338 1.528763
## 1987 1.577363 1.595212 1.600209 1.609362
## 1992 1.630302 1.659425 1.662041 1.670160
## 1997 1.659425 1.721054 1.733230 1.746747
## 2002 1.662041 1.733230 1.769460 1.793577
## 2007 1.670160 1.746747 1.793577 1.838674
```

```r
cov(lifeExp)
```

```
##          1952     1957     1962     1967     1972     1977     1982     1987
## 1952 149.4740 148.8163 146.1139 140.0125 133.2961 128.2374 121.3496 115.2509
## 1957 148.8163 149.6044 147.3309 142.0055 135.7643 130.7612 123.8030 117.7603
## 1962 146.1139 147.3309 146.3433 140.8698 135.0029 130.2947 123.5347 117.7853
## 1967 140.0125 142.0055 140.8698 137.3316 132.6094 128.3811 122.0879 116.8755
## 1972 133.2961 135.7643 135.0029 132.6094 129.5489 126.7057 120.2478 115.6296
```

```
## 1977 128.2374 130.7612 130.2947 128.3811 126.7057 126.0507 119.2315 115.2083
## 1982 121.3496 123.8030 123.5347 122.0879 120.2478 119.2315 116.0062 112.9900
## 1987 115.2509 117.7603 117.7853 116.8755 115.6296 115.2083 112.9900 111.4352
## 1992 114.8457 117.5331 117.7528 117.2470 116.5765 116.8622 115.6543 115.6677
## 1997 114.8698 117.5484 117.7097 117.2918 116.4533 116.6372 115.5977 115.9447
## 2002 118.0452 120.7149 120.7665 120.3362 119.2583 119.0025 117.5598 117.8361
## 2007 114.6120 117.1245 117.1912 116.8119 115.7111 115.2924 113.9993 114.3162
##          1992     1997     2002     2007
## 1952 114.8457 114.8698 118.0452 114.6120
## 1957 117.5331 117.5484 120.7149 117.1245
## 1962 117.7528 117.7097 120.7665 117.1912
## 1967 117.2470 117.2918 120.3362 116.8119
## 1972 116.5765 116.4533 119.2583 115.7111
## 1977 116.8622 116.6372 119.0025 115.2924
## 1982 115.6543 115.5977 117.5598 113.9993
## 1987 115.6677 115.9447 117.8361 114.3162
## 1992 126.0541 126.5995 128.2683 124.2956
## 1997 126.5995 133.6206 139.5559 135.4912
## 2002 128.2683 139.5559 150.7940 147.7072
## 2007 124.2956 135.4912 147.7072 145.7578
```

Here we see that the variabilities are not differs too much, so we can simply use sample covariance matrix S for our PCA (i.e., we **do not need to add the option scale=TRUE** in the **prcomp** command for PCA)

Next I will perform PCA on log(GDP) and life-expectancy separately.

PCA for log(GDP)

```
# run PCA for logGDP data
logGDP.pca = prcomp(gdp)

# total variability
logGDP.total_var = sum(logGDP.pca$sdev^2)

# proportion of variation explained by each of PC
summary(logGDP.pca)
```

```
## Importance of components:
##                            PC1    PC2     PC3    PC4     PC5     PC6     PC7
## Standard deviation     4.0608 0.8718 0.38139 0.2215 0.16930 0.11366 0.09388
## Proportion of Variance 0.9416 0.0434 0.00831 0.0028 0.00164 0.00074 0.00050
## Cumulative Proportion  0.9416 0.9850 0.99328 0.9961 0.99771 0.99845 0.99895
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.07325 0.06644 0.05813 0.05649 0.04419
## Proportion of Variance 0.00031 0.00025 0.00019 0.00018 0.00011
## Cumulative Proportion  0.99926 0.99951 0.99971 0.99989 1.00000
```

```
# then look at the scree plot
plot(1:12, logGDP.pca$sdev^2 / logGDP.total_var,
     xlab="Principal Component",
     ylab="Percentage of variance explained",
     ylim=c(0,1),
     main="Scree plot")
```

Since from the scree plot and the PCA summary, we can see that 94.16% of variability can be explained by PC1, and further 4.34% by PC2. So if I need to keep at least 95% variability of the data, I choose the first 2

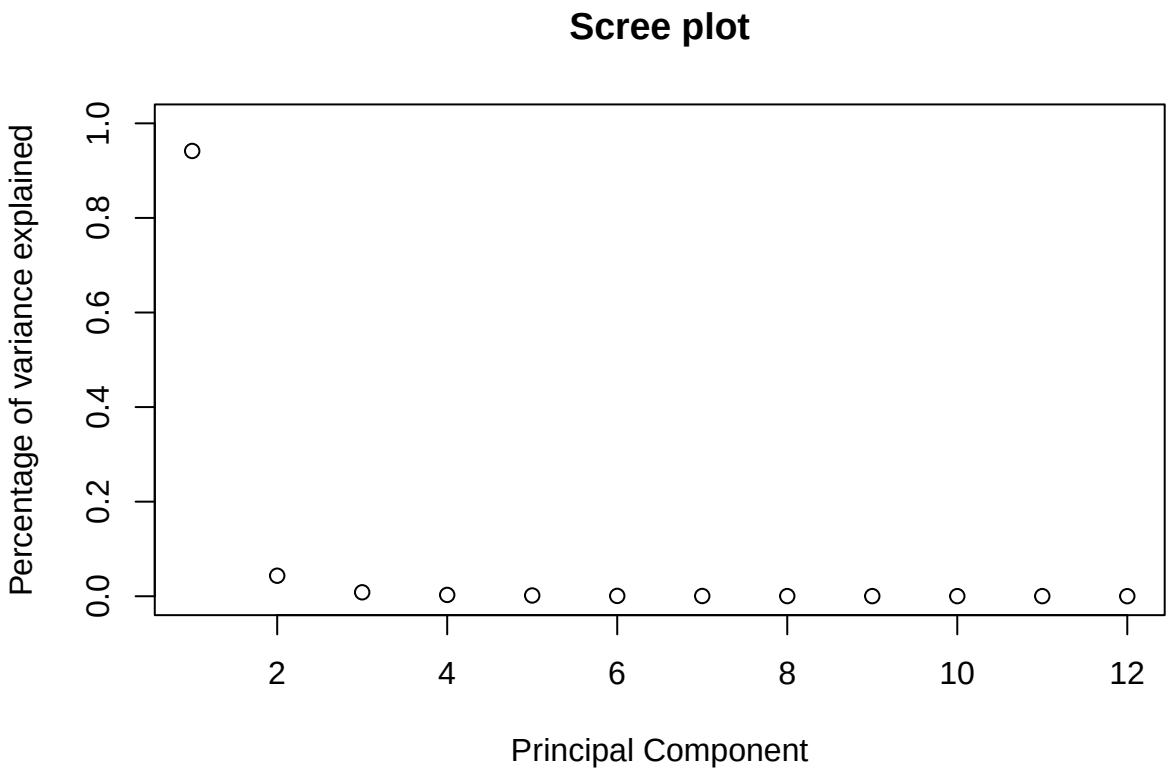


Figure 5: Scree plot for PCA on logGDP

PCs.

```
# see the data with first 2 PCs, only show the first 15 rows
head(cbind(logGDP.pca$x[,1:2], gdp), 15)
```

```
##                                PC1          PC2     1952     1957     1962
## Algeria                 -0.65612685  0.100124796 7.803438 8.011015 7.844169
## Angola                   0.18870103  1.433886716 8.166390 8.250082 8.359200
## Benin                    3.90718350  0.194259188 6.968617 6.866518 6.855935
## Botswana                -0.04307935 -2.378945139 6.746695 6.822451 6.891274
## Burkina Faso             5.00867078 -0.145594772 6.297579 6.425166 6.582734
## Burundi                  7.02641886  0.001440845 5.826874 5.939025 5.872690
## Cameroon                 2.46080175  0.226785963 7.067036 7.180107 7.243947
## Central African Republic 4.67703018  1.204169584 6.976638 7.082418 7.084284
## Chad                     3.95425271  0.628260356 7.072139 7.176633 7.236928
## Comoros                  3.55678729  0.986955237 7.005781 7.099324 7.248965
## Congo Dem. Rep.          6.36082589  1.944411780 6.659989 6.808885 6.798292
## Congo Rep.               0.31257931  0.332441066 7.661819 7.747189 7.809859
## Cote d'Ivoire            2.22886077  0.810788840 7.236048 7.313817 7.455223
## Djibouti                 1.09019808  1.409925038 7.889658 7.960313 8.013340
## Egypt                    0.74952875 -0.686833392 7.257583 7.285448 7.434456
##                             1967     1972     1977     1982     1987     1992
## Algeria                 8.085484 8.338704 8.499114 8.656113 8.644946 8.521826
## Angola                  8.616636 8.607635 8.009246 7.921882 7.795732 7.873920
## Benin                   6.942960 6.990069 6.936499 7.152972 7.111395 7.082723
## Botswana                7.102260 7.724717 8.075538 8.423134 8.733253 8.981444
## Burkina Faso            6.678124 6.750793 6.611217 6.693570 6.815709 6.837068
```

```
## Burundi                  6.023393 6.140099 6.320954 6.327228 6.432649 6.448414
## Cameroon                 7.318840 7.429014 7.486295 7.769794 7.864291 7.491737
## Central African Republic 7.035318 6.975426 7.011551 6.863545 6.739190 6.617277
## Chad                     7.087415 7.006789 7.033493 6.681993 6.858971 6.964196
## Comoros                  7.536913 7.569194 7.066981 7.144486 7.182338 7.128422
## Congo Dem. Rep.          6.758783 6.807820 6.679294 6.512856 6.511411 6.126256
## Congo Rep.               7.892803 8.075008 8.089231 8.492800 8.343124 8.298101
## Cote d'Ivoire            7.626595 7.774100 7.831116 7.864309 7.676453 7.407362
## Djibouti                 8.013029 8.214523 8.033256 7.965361 7.965581 7.773660
## Egypt                    7.503775 7.612835 7.932180 8.161583 8.264997 8.241375
##                              1997     2002     2007
## Algeria                  8.475808 8.573203 8.736066
## Angola                   7.730676 7.927789 8.475794
## Benin                    7.117185 7.224664 7.273290
## Botswana                 9.064984 9.305978 9.439057
## Burkina Faso             6.852554 6.944709 7.104171
## Burundi                  6.137976 6.101223 6.063950
## Cameroon                 7.435047 7.567352 7.621732
## Central African Republic 6.607334 6.604879 6.559639
## Chad                     6.912704 7.052878 7.440771
## Comoros                  7.067847 6.980831 6.893806
## Congo Dem. Rep.          5.743607 5.485485 5.626008
## Congo Rep.               8.155984 8.155954 8.197692
## Cote d'Ivoire            7.487882 7.407804 7.342617
## Djibouti                 7.546983 7.553948 7.641316
## Egypt                    8.336434 8.466869 8.627156
```

From the data above, for PC1, some give large positive value if the logGDP each year are "low"; whilst large negative value if the logGDP each year are "high". For example, for **Burundi** (PC1=7.026), generally the logGDP each year are between 5 to 7; for **Gabon** (PC1=-3.818), the logGDP each year are above 8.

So PC1 can be interpreted as to differentiate wealthy country and poorer country.

For PC2, seems can be explained by the changes over the years. For example, **Botswana** (PC2=-2.378) increases its logGDP value from 6.746 to 9.439 over these years. **Equatorial Guinea** (PC2=-2.590) also behave similar (increase from 5.928 to 9.405). In contract, **Central African Republic** with PC2 1.294 decrease its logGDP value from 6.976 to 6.559. **Congo Dem. Rep.** with PC2=1.944 decrease from 6.659989 to 5.626008;

So PC2 can be viewed as the change (negative for increasing; positive for decreasing).

Next is a scatter plots for the first 2 principal component scores. Although the question asked for the first three, I think 2-d plot is easier to visualize so I will keep using 2 PC scores only.

```
gdp_with_continent = data.frame(cbind(gap[,1:2], gdp)) %>% rename(continent=1, country=2)

autoplot(logGDP.pca, data = gdp_with_continent, colour="continent", label=TRUE,scale=FALSE)
```

Based on this plot, **Congo Dem. Rep.** is have having both large PCs; whilst **Hong Kong China** and **Singapore** are low in PCs. Those seemed support my observation, as **Hong Kong China** and **Singapore** are also wealthy countries with GDP increasing over these years; But **Congo Dem. Rep.** is among the poorest nations in the world. In addition, **Kuwait** on the top-left, from the data, it shows that **Kuwait** is wealthy but has decreasing log(GDP), which also supports my inspection.

From the plot above with continent colored, we can also see that the top right of the plot (large PC1 and large PC2) are mainly **Africa**, and mainly **Europe** on the left (low PC1, with PC2 around 0); and some **Asia** countries on the bottom left (low PC1, low PC2). These can be explained as: over these years, most **Africa** countries are under-developed, some **Asia** country is developing, and most **Europe** countries developed.

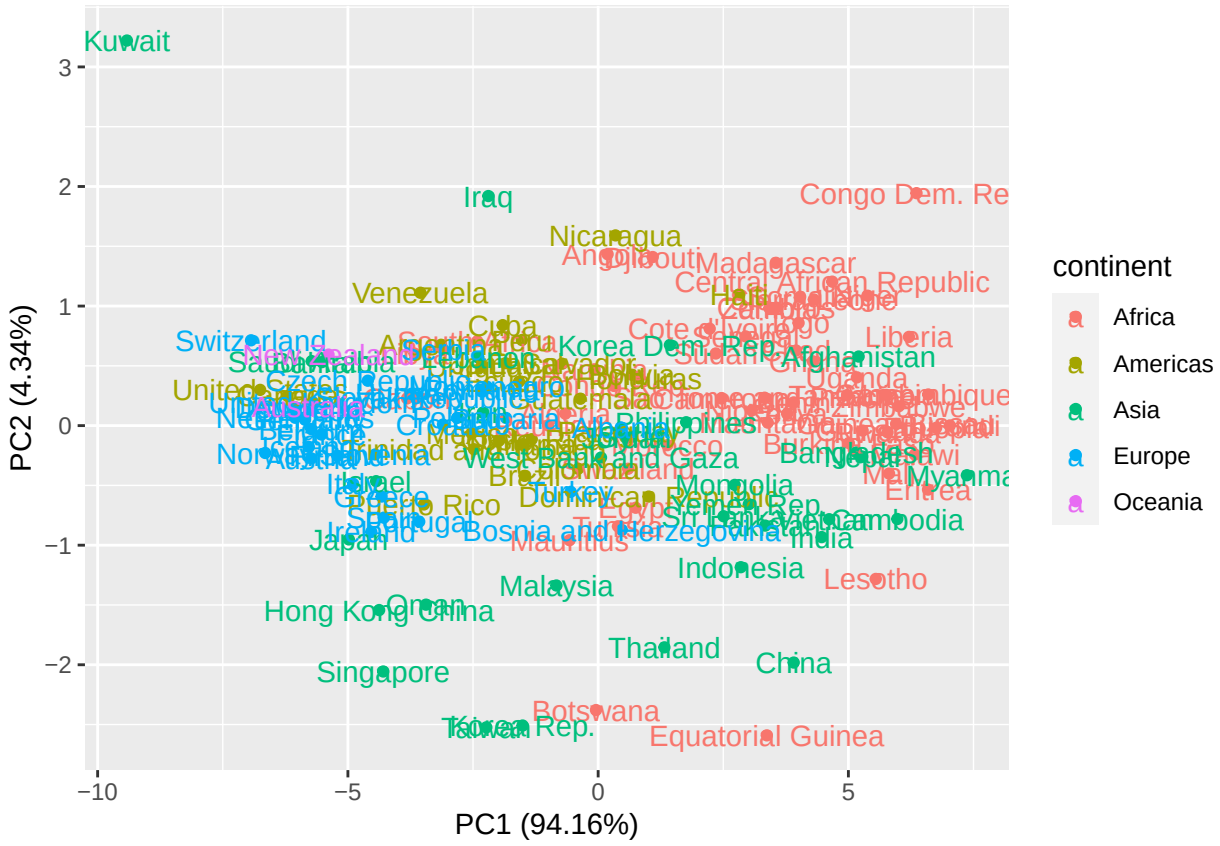


Figure 6: Scatter plots for the first 2 Principal component scores on logGDP

PCA for Life-Expectancy

```
# run PCA for lifeExp data
lifeExp.pca = prcomp(lifeExp)

# total variability
lifeExp.total_var = sum(lifeExp.pca$sdev^2)

# proportion of variation explained by each of PC
summary(lifeExp.pca)
```

```
## Importance of components:
##                            PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     38.6350 9.83955 4.50873 2.52549 1.39085 1.27733 1.01896
## Proportion of Variance  0.9203 0.05969 0.01253 0.00393 0.00119 0.00101 0.00064
## Cumulative Proportion   0.9203 0.97994 0.99247 0.99641 0.99760 0.99860 0.99924
##                            PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.79691 0.48377 0.43319 0.33836 0.23486
## Proportion of Variance 0.00039 0.00014 0.00012 0.00007 0.00003
## Cumulative Proportion  0.99964 0.99978 0.99990 0.99997 1.00000
```

```
# then look at the scree plot
plot(1:12, lifeExp.pca$sdev^2 / lifeExp.total_var,
     xlab="Principal Component",
     ylab="Percentage of variance explained",
     ylim=c(0,1),
     main="Scree plot")
```

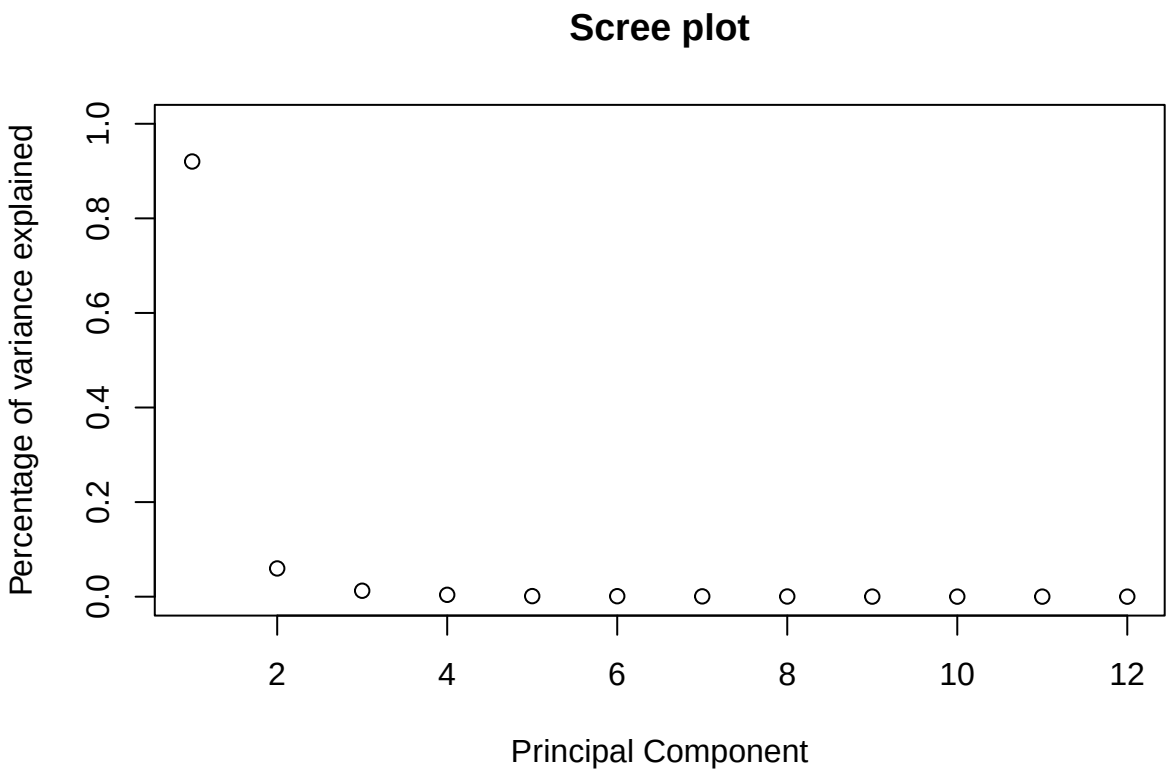


Figure 7: Scree plot for PCA on lifeExp

Since from the scree plot and the PCA summary, we can see that 92.03% of variability can be explained by PC1, and further 5.969% by PC2. So if I need to keep at least 95% variability of the data, I choose the first 2 PCs.

```
# see the data with the first 2 PCs, only show the first 15 rows
head(cbind(lifeExp.pca$x[,1:2], lifeExp), 15)
```

```
##                               PC1         PC2   1952   1957   1962   1967
## Algeria                  -1.900204 -14.4755608 43.077 45.685 48.303 51.407
## Angola                  -74.579717   8.5322851 30.015 31.999 34.000 35.985
## Benin                   -37.043948   0.2682935 38.223 40.358 42.618 44.885
## Botswana                -16.963419  18.1542772 47.622 49.618 51.520 53.298
## Burkina Faso            -51.290904  -0.6935283 31.975 34.906 37.814 40.697
## Burundi                 -50.503052  11.3943585 39.031 40.533 42.045 43.548
## Cameroon                -39.374537   6.0140441 38.523 40.428 42.643 44.799
## Central African Republic -54.056255  10.4663905 35.463 37.464 39.475 41.478
## Chad                    -43.936419   5.2874420 38.092 39.881 41.716 43.601
## Comoros                 -24.610126  -7.5999295 40.715 42.460 44.467 46.472
## Congo Dem. Rep.         -51.432538  15.4812922 39.143 40.652 42.122 44.056
## Congo Rep.              -24.117165   9.6450091 42.111 45.053 48.435 52.040
## Cote d'Ivoire           -38.181126  13.7781797 40.477 42.469 44.930 47.350
## Djibouti                -45.385363  -1.5745993 34.812 37.328 39.693 42.074
## Egypt                   -11.361785 -14.3346790 41.893 44.444 46.992 49.293
##                            1972   1977   1982   1987   1992   1997   2002
## Algeria                  54.518 58.014 61.368 65.799 67.744 69.152 70.994
## Angola                   37.928 39.483 39.942 39.906 40.647 40.963 41.003
## Benin                    47.014 49.190 50.904 52.337 53.919 54.777 54.406
## Botswana                 56.024 59.319 61.484 63.622 62.745 52.556 46.634
## Burkina Faso             43.591 46.137 48.122 49.557 50.260 50.324 50.650
## Burundi                  44.057 45.910 47.471 48.211 44.736 45.326 47.360
## Cameroon                 47.049 49.355 52.961 54.985 54.314 52.199 49.856
## Central African Republic 43.457 46.775 48.295 50.485 49.396 46.066 43.308
## Chad                     45.569 47.383 49.517 51.051 51.724 51.573 50.525
## Comoros                  48.944 50.939 52.933 54.926 57.939 60.660 62.974
## Congo Dem. Rep.          45.989 47.804 47.784 47.412 45.548 42.587 44.966
## Congo Rep.               54.907 55.625 56.695 57.470 56.433 52.962 52.970
## Cote d'Ivoire            49.801 52.374 53.983 54.655 52.044 47.991 46.832
## Djibouti                 44.366 46.519 48.812 50.040 51.604 53.157 53.373
## Egypt                    51.137 53.319 56.006 59.797 63.674 67.217 69.806
##                            2007
## Algeria                  72.301
## Angola                   42.731
## Benin                    56.728
## Botswana                 50.728
## Burkina Faso             52.295
## Burundi                  49.580
## Cameroon                 50.430
## Central African Republic 44.741
## Chad                     50.651
## Comoros                  65.152
## Congo Dem. Rep.          46.462
## Congo Rep.               55.322
## Cote d'Ivoire            48.328
## Djibouti                 54.791
## Egypt                    71.338
```

From the data above, for PC1, some give large positive value if the lifeExp each year are "high"; whilst large negative value if the lifeExp each year are "low". For example, for **Angola** (PC1=-74.579), generally the lifeExp each year are around 40; for **New Zealand** (PC1=50.569), the lifeExp each year are above 70 except the first year is 69.39.

So PC1 can be interpreted as to differentiate high and low lifeExp.

For PC2, seems harder to explain easily. One may also using the "trend" to explain. Smaller PC2 means the For example, **Oman** has PC2=-26.2570 increases its lifeExp value from 40.080 to 75.640 over these years. for **Zimbabwe** having PC2=27.834, its lifeExp is 48.451 for year 1952 to 43.487 for year 2007. So the changes is not large.

So PC2 can be viewed as the changes. Small value means change rapidly; whilst large value means the lifeExp are stable over these years.

Next is a scatter plots for the first 2 principal component scores. Although the question asked for the first three, I think 2-d plot is easier to visualize so I will keep using 2 PC scores only.

```
lifeExp_with_continent = data.frame(cbind(gap[,1:2], lifeExp)) %>% rename(continent=1, country=2)

autoplot(lifeExp.pca, data = lifeExp_with_continent, colour="continent", label=TRUE,scale=FALSE)
```

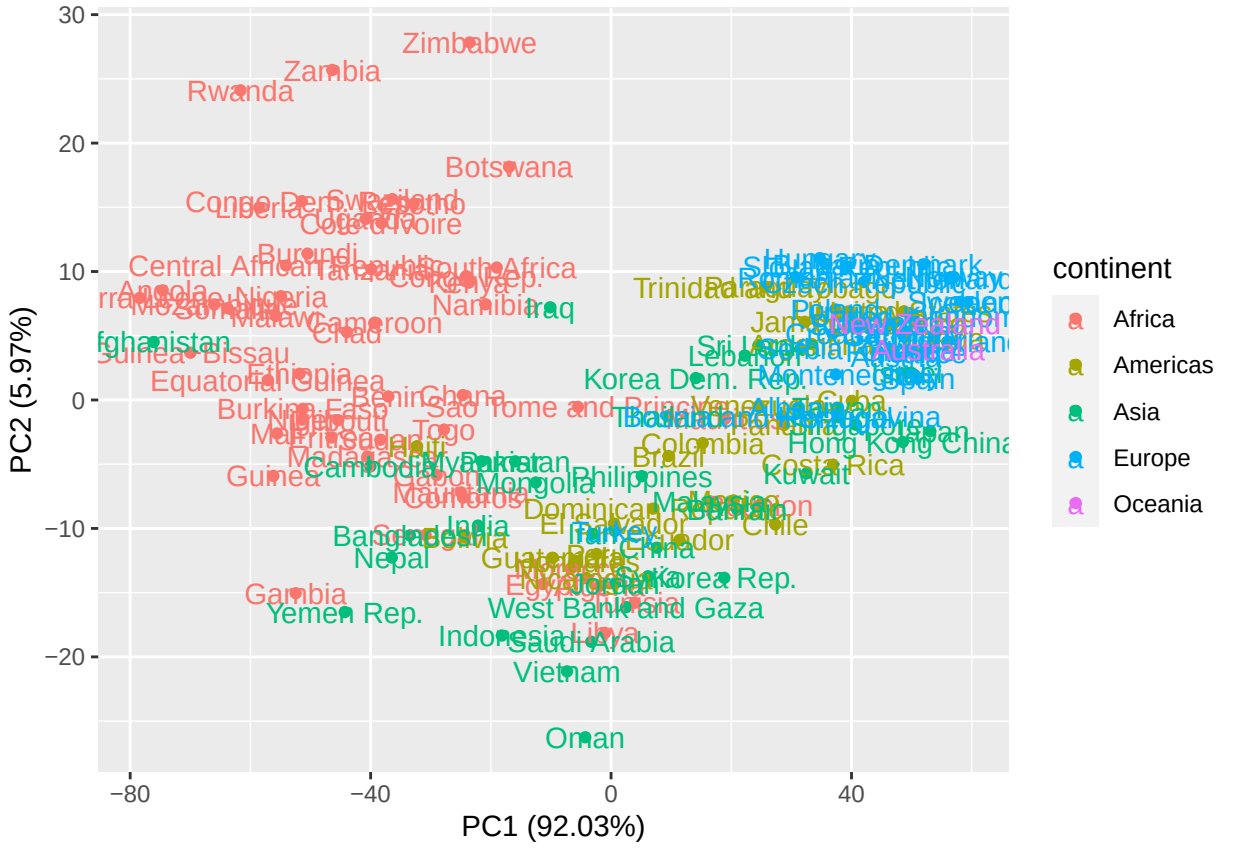


Figure 8: Scatter plots using the first 2 Principal component on lifeExp

Based on this plot, for instance, we can see that **Denmark** has both large PC1 and PC2, where from my inspection, has greater life expectancy and the not changes largely over these years. Among the top left countries, for example, **Zambia**, has low PC1 and high PC2, which also satisfied my explanation that the life expectancy is relatively low and not changes a lot (mostly around 45). In addition for **Oman** which has about 0 PC1 and lowest PC2, which the life expectancy are moderate (average is around 60 over these years),

and has a rapid change (increase from 37.578 to 75.640, which is a rapid increase).

In terms of the continents, we can see that most **Europe** countries on the top right; **Asia** in the bottom middle; and **Africa** countries on the top left. There are many possible explanation to describe this distribution, but maybe we can use **living condition** to explain this phenomenon. Because most like **Europe** countries has a better education and medical condition, so most people will live longer; on the other hand, **Africa** countries are mainly under-developed or developing countries, so most of them are suffering from various fetal diseases. In addition, because most **Asia** countries are developing rapidly after world war II, so they have a boost of economy which making them having more money to improve their living condition over these years.

Multidimensional scaling (MDS)

This section will carry out MDS using the combined dataset of log(GDP) and life-expectancy.

```
# use the combined dataset
combined_data = gap[,3:26]

# calculate the distiance matrix
combined_data.dist = dist(combined_data)

# perform MDS, keep only dimension k = 2
combined_data.mds = as.data.frame(cmdscale(combined_data.dist, k=2))

combined_data_with_continent.mds = cbind(gap[,1:2], combined_data.mds)
colnames(combined_data_with_continent.mds) = c("continent", "country", "X1", "X2")

# plot the graph with colored continent
ggplot(combined_data_with_continent.mds, aes(x=X1, y=X2, colour=continent, label=country))+
  geom_text(aes(label=country))+ ggtitle("MDS for GAP dataset")
```

This plot also looks the same as the previous plot: PCA on life-expectancy for the first 2 principal components. To interpret solely on this plot, we can claim that log(GDP) and life-expectancy are highly similar within continent. For example, **Africa** countries are mainly arranged on the left; **Europe** on the right; and **Asia** are in the bottom middle.

Hypothesis Test

In this section, we would like to perform the following multivariate hypothesis test:

1. to test whether there was a statistically significant difference between the mean log(GDP) and life expectancy of Asian and European countries in the year 2007
2. to test whether there was a statistically significant difference between continents in 1952

Test for year 2007 data

Let $\mu_{Asia} = (\mu_{Asia,1}, \mu_{Asia,2})^T$ and $\mu_{Europe} = (\mu_{Europe,1}, \mu_{Europe,2})^T$

Where $\mu_{continent}$ is a 2-dim vector with the first element $\mu_{continent,1}$ is the mean of logGDP for that continent; and the second element $\mu_{continent,2}$ is the mean of life-expectancy.

Our multivariate test is aim at the if there are significant difference in the mean vector.

$$H0 : \mu_{\mathbf{Asia}} = \mu_{\mathbf{Europe}}$$
$$H1 : \mu_{\mathbf{Asia}} \neq \mu_{\mathbf{Europe}}$$

And equivalently, we test if the difference is a zero-vector.

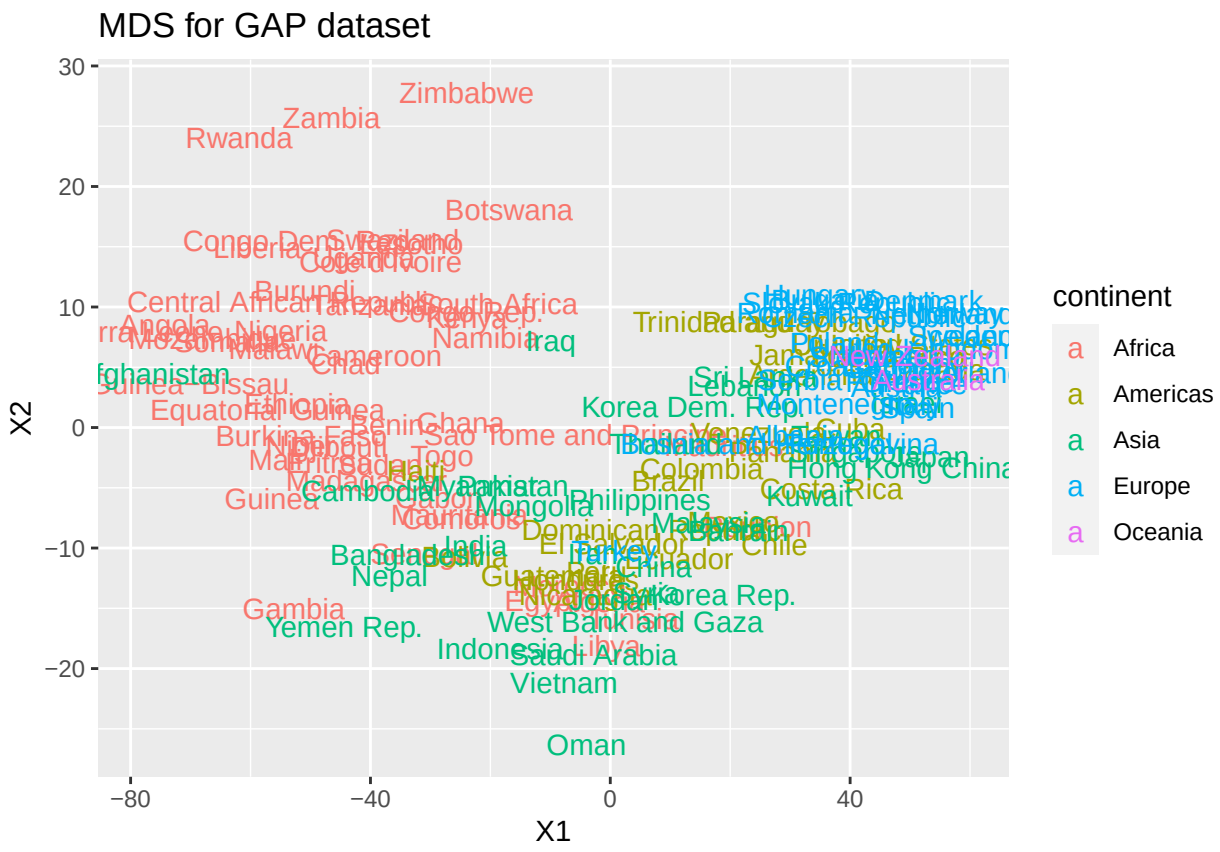


Figure 9: MDS for GDP dataset

let $\mu = \mu_{Asia}$ - μ_{Europe}, then test

$$H0 : \mu = \mathbf{0_2}$$
$$H1 : \mu \neq \mathbf{0_2}$$

Assumption:

1. data for **Asia** and **Europe** are independent
2. for both continent, each sample data are from corresponding multivariate normal distribution $N_2(\mu_{continent}, \Sigma_{continent})$
3. common population covariance matrix, i.e. $\Sigma_{Asia} = \Sigma_{Europe}$

To carry out the test, first prepare 2007 data

```
# first column of gap is continent; then need 2007 log(GDP) and life-expectancy
data_2007 = data.frame(continent=gap[,1],
                       logGDP_2007=gdp$"2007",
                       lifeExp_2007=lifeExp$"2007")

# filter out only Asia and Europe, in separate dataframe
Asia_2007 = data_2007 %>% filter(continent == "Asia") %>% select(! continent)
Europe_2007 = data_2007 %>% filter(continent == "Europe") %>% select(! continent)

n = nrow(Asia_2007)
m = nrow(Europe_2007)

n;m
```

```
## [1] 33
```

```
## [1] 30
```

We can take a look at assumption 3, to see if the sample covariance matrix are approximately equal or not

```
print(cov(Asia_2007))
```

```
##              logGDP_2007 lifeExp_2007
## logGDP_2007     1.527396     7.877177
## lifeExp_2007    7.877177    63.420907
```

```
print(cov(Europe_2007))
```

```
##              logGDP_2007 lifeExp_2007
## logGDP_2007    0.3472175     1.467651
## lifeExp_2007   1.4676508     8.879283
```

Although it is in-doubt that the covariance matrix are equal as the variance of lifeExp are differs a lot. I may still go on complete the hypothesis as requested.

Then we can make use of R command to test this, using library **ICSNP** and **HotellingsT2**. and for significance level $\alpha = 0.05$, we reject H0 if the test statistic returned from R command is greater than $F_{2,n+m-2-1,\alpha}$, where n = sample size of **Asia** data; m = sample size of **Europe** data

Using **HotellingsT2(data1, data2)** to compute the test statistic

```
# library ICSNP provide HotellingsT2 command
library("ICSNP")


HotellingsT2(Asia_2007, Europe_2007)
```

```
##
##  Hotelling's two sample T2-test
##
## data:  Asia_2007 and Europe_2007
## T.2 = 12.681, df1 = 2, df2 = 60, p-value = 2.55e-05
## alternative hypothesis: true location difference is not equal to c(0,0)
```

We can the test statistic $\delta^2 = 12.681$, with p-value = 2.55e-05

Since with significance level $\alpha = 0.05$, the critical value = $F_{2,n+m-2-1,\alpha} = F_{2,33+30-2-1,0.05} = F_{2,60,0.05} = 3.1504113$

Since the test statistic $\delta^2 = 12.681 > 3.1504113$, so we can reject H0 and conclude that the population mean for **Asia** and **Europe** are significantly different in 2007.

We can also see the plot below, we can see that the points for both continents are quite differs, which also supports our result of hypothesis test.

```r
# min/max for the x-y limit of the plot
min_2007 = apply(rbind(Asia_2007, Europe_2007), 2, min)
max_2007 = apply(rbind(Asia_2007, Europe_2007), 2, max)

# calculate the mean for plotting
Asia_2007_mean = apply(Asia_2007, 2, mean)
Europe_2007_mean = apply(Europe_2007, 2, mean)


# scatter plot
plot(lifeExp_2007~logGDP_2007, data = Asia_2007,
     xlim=c(min_2007[[1]], max_2007[[1]]),
     ylim=c(min_2007[[2]], max_2007[[2]]))
points(lifeExp_2007~logGDP_2007, data = Europe_2007, pch=2, col=2)

# mark the mean onto the plot
points(Asia_2007_mean[[1]], Asia_2007_mean[[2]], pch=3, col=1, cex=3, lwd=2)
points(Europe_2007_mean[[1]], Europe_2007_mean[[2]], pch=3, col=2, cex=3, lwd=2)

legend(x="bottomright", legend=c("Asia", "Europe"),
       col=c(1,2),
       pch=c(1,2))
```

Test for difference between continents in 1952

The question asked "Were the continents more similar in the year 1952?".

Firstly, because there are not specifying which continents, so I guess I should test treat all continents instead just **Asia** and Europe. Secondly, I think the term "more similar" is not easy to setup a test of hypothesis because this seemed not asking for testing the difference in mean for each continents.

Anyway, I think I should carry out both in order to trying to match the question criteria:

1. perform testing on **Asia** and **Europe** for 1952 data
2. perform testing on all continenets on 1952 data

perform testing on Asia and Europe for 1952 data just carry out the same analysis like 2007 data.

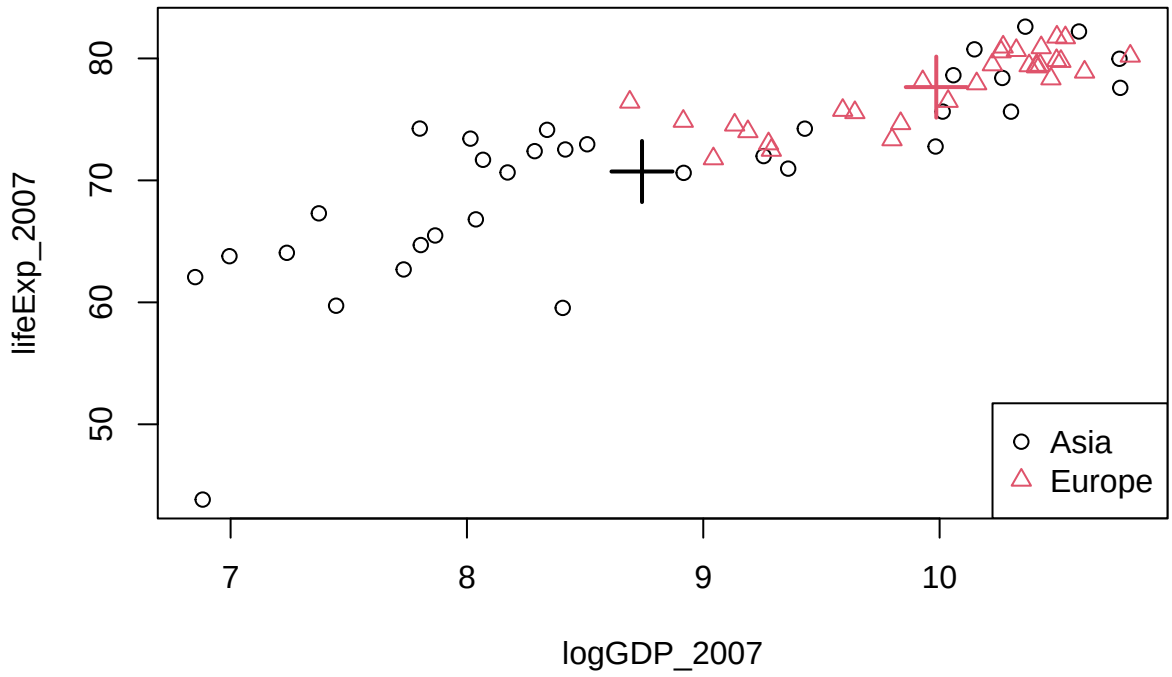


Figure 10: 2007 lifeExp vs logGDP on Asia and Europe

$$H0 : \mu_{\textbf{Asia}} = \mu_{\textbf{Europe}}$$
$$H1 : \mu_{\textbf{Asia}} \neq \mu_{\textbf{Europe}}$$

Prepare the data

```
# first column of gap is continent; then need 1952 log(GDP) and life-expectancy
data_1952 = data.frame(continent=gap[,1],
                       logGDP_1952=gdp$"1952",
                       lifeExp_1952=lifeExp$"1952")

# filter out only Asia and Europe, in separate dataframe
Asia_1952 = data_1952 %>% filter(continent == "Asia") %>% select(! continent)
Europe_1952 = data_1952 %>% filter(continent == "Europe") %>% select(! continent)

n = nrow(Asia_1952)
m = nrow(Europe_1952)

n;m
```

```
## [1] 33
```

```
## [1] 30
```

Check if assumption 3 hold:

```
print(cov(Asia_1952))
```

```
##              logGDP_1952 lifeExp_1952
## logGDP_1952     1.312206     5.458348
## lifeExp_1952    5.458348    86.336631
```

```
print(cov(Europe_1952))
```

```
##              logGDP_1952 lifeExp_1952
## logGDP_1952    0.3734262     3.362349
## lifeExp_1952   3.3623494    40.463444
```

The covariance matrix still a little bit different, but not too much

Scatter plot:

```
# min/max for the x-y limit of the plot
min_1952 = apply(rbind(Asia_1952, Europe_1952), 2, min)
max_1952 = apply(rbind(Asia_1952, Europe_1952), 2, max)

# calculate the mean for plotting
Asia_1952_mean = apply(Asia_1952, 2, mean)
Europe_1952_mean = apply(Europe_1952, 2, mean)


# scatter plot
plot(lifeExp_1952~logGDP_1952, data = Asia_1952,
     xlim=c(min_1952[[1]], max_1952[[1]]),
     ylim=c(min_1952[[2]], max_1952[[2]]))
points(lifeExp_1952~logGDP_1952, data = Europe_1952, pch=2, col=2)

# mark the mean onto the plot
points(Asia_1952_mean[[1]], Asia_1952_mean[[2]], pch=3, col=1, cex=3, lwd=2)
points(Europe_1952_mean[[1]], Europe_1952_mean[[2]], pch=3, col=2, cex=3, lwd=2)

legend(x="bottomright", legend=c("Asia", "Europe"),
       col=c(1,2),
       pch=c(1,2))
```

Test of hypothesis: Using **HotellingsT2(data1, data2)** to compute the test statistic

```
HotellingsT2(Asia_1952, Europe_1952)
```

```
##
##  Hotelling's two sample T2-test
##
## data:  Asia_1952 and Europe_1952
## T.2 = 39.347, df1 = 2, df2 = 60, p-value = 1.21e-11
## alternative hypothesis: true location difference is not equal to c(0,0)
```

We can the test statistic $\delta^2 = 39.347$, with p-value = 1.21e-11

Since with significance level $\alpha = 0.05$, the critical value $= F_{2,n+m-2-1,\alpha} = F_{2,33+30-2-1,0.05} = F_{2,60,0.05} = 3.1504113$

Since the test statistic $\delta^2 = 39.347 > 3.1504113$, so we can reject H0 and conclude that the population mean for **Asia** and **Europe** are significantly different in 1952.

perform testing on all continents for 1952 data I think the question could be asking to test the "similarity" of continents in 1952, i.e. to test of hypothesis for the equality of the population mean:

$$\mu_{Africa} = \mu_{Americas} = \mu_{Asia} = \mu_{Europe} = \mu_{Oceania}$$

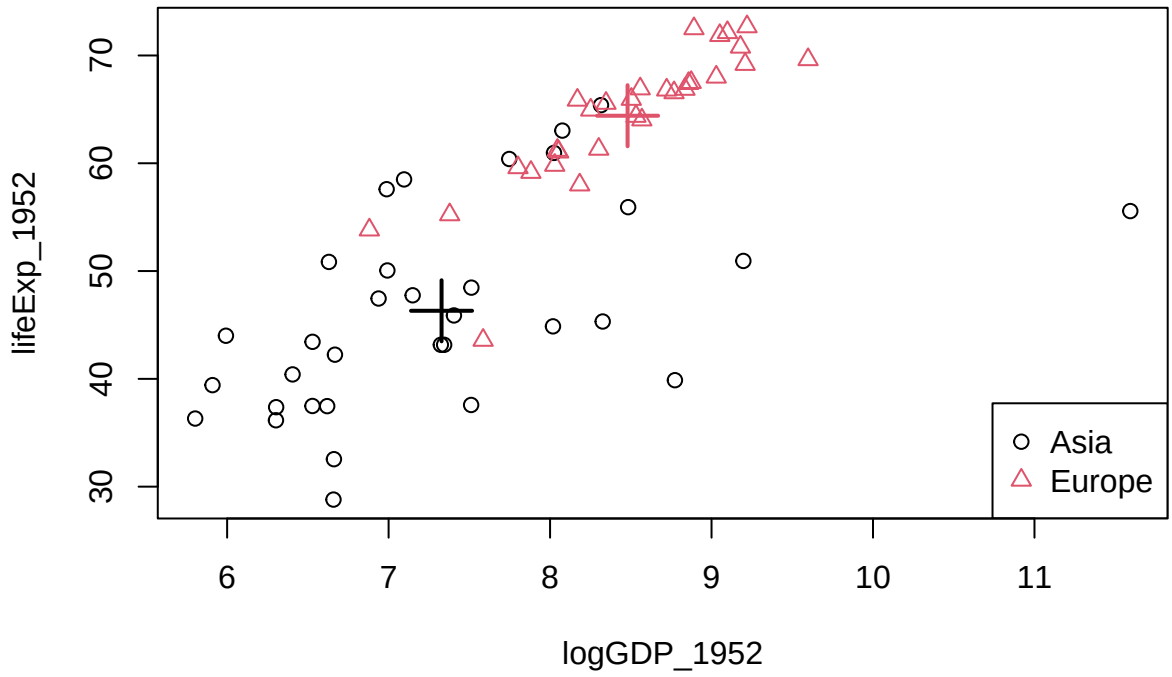


Figure 11: 1952 lifeExp vs logGDP on Asia and Europe

However, this module do not cover multivariate version of ANOVA (MANOVA), so I simply find online resources to take a look how to perform.

[Tutorial for MANOVA] (http://www.sthda.com/english/wiki/manova-test-in-r-multivariate-analysis-of-variance)

First take a look at the scatter plot

```
# get 1952 data
data_1952 = data.frame(continent=gap[,1], logGDP_1952=gdp$"1952", lifeExp_1952=lifeExp$"1952")

# separated by continent
Asia_1952 = data_1952 %>% filter(continent == "Asia") %>% select(! continent)
Europe_1952 = data_1952 %>% filter(continent == "Europe") %>% select(! continent)
Oceania_1952 = data_1952 %>% filter(continent == "Oceania") %>% select(! continent)
Africa_1952 = data_1952 %>% filter(continent == "Africa") %>% select(! continent)
Americas_1952 = data_1952 %>% filter(continent == "Americas") %>% select(! continent)

# calculate the mean for each continent
Asia_1952_mean = apply(Asia_1952, 2, mean)
Europe_1952_mean = apply(Europe_1952, 2, mean)
Oceania_1952_mean = apply(Oceania_1952, 2, mean)
Africa_1952_mean = apply(Africa_1952, 2, mean)
Americas_1952_mean = apply(Americas_1952, 2, mean)


# prepare data for x,y limit
min_1952 = apply(data_1952[,2:3], 2, min)
max_1952 = apply(data_1952[,2:3], 2, max)

# scatter plots for different continents
```

```r
# with marked the mean
plot(lifeExp_1952~logGDP_1952, data = Asia_1952,
     xlim=c(min_1952[[1]], max_1952[[1]]),
     ylim=c(min_1952[[2]], max_1952[[2]]))
points(lifeExp_1952~logGDP_1952, data = Europe_1952, pch=2, col=2)
points(lifeExp_1952~logGDP_1952, data = Oceania_1952, pch=2, col=3)
points(lifeExp_1952~logGDP_1952, data = Africa_1952, pch=2, col=4)
points(lifeExp_1952~logGDP_1952, data = Americas_1952, pch=2, col=5)

points(Asia_1952_mean[[1]], Asia_1952_mean[[2]], pch=3, col=1, cex=3, lwd=2)
points(Europe_1952_mean[[1]], Europe_1952_mean[[2]], pch=3, col=2, cex=3, lwd=2)
points(Oceania_1952_mean[[1]], Oceania_1952_mean[[2]], pch=3, col=3, cex=3, lwd=2)
points(Africa_1952_mean[[1]], Africa_1952_mean[[2]], pch=3, col=4, cex=3, lwd=2)
points(Americas_1952_mean[[1]], Americas_1952_mean[[2]], pch=3, col=5, cex=3, lwd=2)

legend(x="bottomright", legend=c("Asia", "Europe", "Oceania", "Africa", "Americas"),
       col=1:5,
       pch=1:5)
```

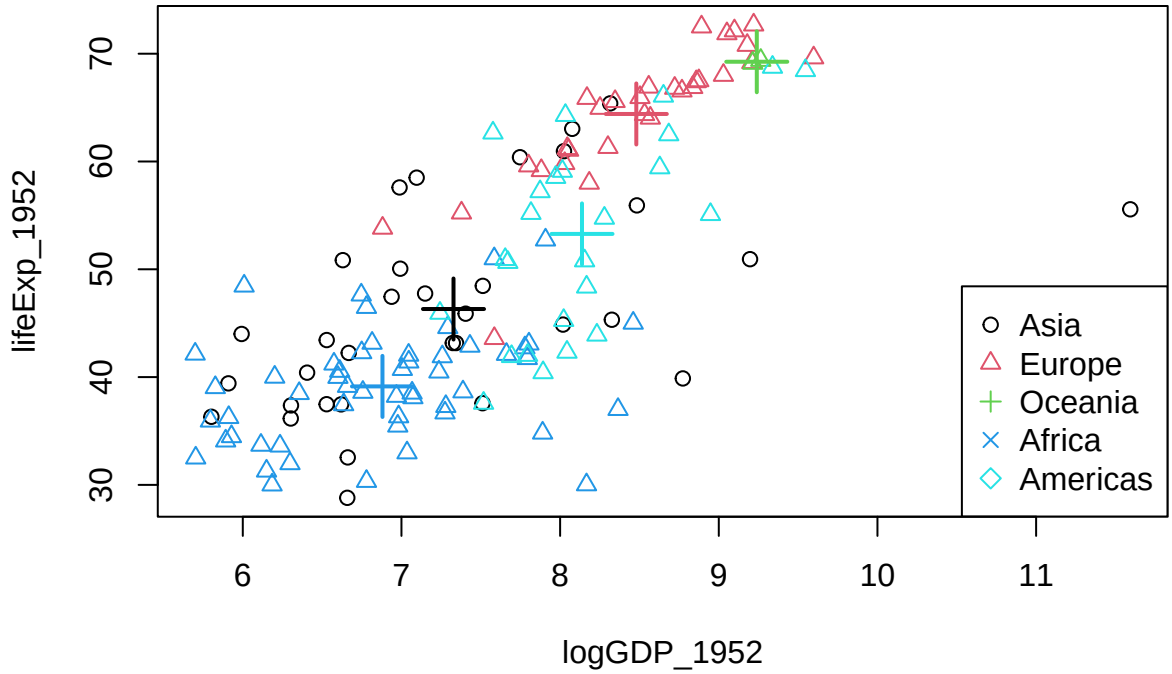


Figure 12: 1952 lifeExp vs logGDP on all continents

Simply looking at the plot, we can see that the mean are quite different for all continents.

for MANOVA part, to perform the test, each $\mu_{continent}$ is a 2d vector, with elements mean of logGDP and lifeExp, respectively.

$$H0 : \mu_{Africa} = \mu_{Americas} = \mu_{Asia} = \mu_{Europe} = \mu_{Oceania}$$
$$H1 : otherwise$$

We can simply use **manova** command to perform this test.

```r
# perform MANOVA test
manova_out = manova(cbind(logGDP_1952, lifeExp_1952) ~ continent, data = data_1952)

summary(manova_out)
```

```
##            Df  Pillai approx F num Df den Df    Pr(>F)
## continent   4 0.69528   18.252      8    274 < 2.2e-16 ***
## Residuals 137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the p-value of the test $<$ 2.2e-16, we can reject H0 and conclude that not all mean vectors are equal.

Although there are several assumptions that stated in the tutorial link, I am not going to perform diagnostic as I think MANOVA test for this question is just an extra work for us to explore the extension of ANOVA.

Discriminant analysis

This section I will use discriminant analysis to perform classification. This part required to use log(GDP) and life-expectancy to predict (classify) the continent of each country. And there are 2 questions:

1. using Linear Discriminant analysis (LDA) to classify continent
2. Give a plot of the 2d projection of the data onto the first two eigenvectors found by Fisher's discriminant analysis approach

using LDA for classification

First prepare the data, with train-test split

```r
# to confirm I can reproduce the result, I use set.seed here, but in practice this is not required
set.seed(1234)

# required continent, lifeExp and log(GDP) for each year
# directly use "gap" dataframe is enough
# the "-2" exclude "country" column
df = gap[, -2]

n = nrow(df)

# use 70%-30% split for train-test split
shuffled_df= df[sample(1:n), ]
n_train = floor(0.7*n)
n_test = n - n_train

print(n_train)
```

```
## [1] 99
```

```r
print(n_test)
```

```
## [1] 43
```

```r
# get the required rows according to the n_train and n_test as specified
training_set = shuffled_df[1:n_train,]
test_set = shuffled_df[(n_train+1): n,]
```

Then we can build the classifier, using **lda** command

```r
# use MASS library to provide lda command
library("MASS")


# use all columns except country, to classify continent
continent.lda = lda(continent ~ . , data = training_set)
```

```r
# use continent.lda to predict the data
continent.predict = predict(continent.lda, test_set[, 2:25])


# calculate the prediction accuracy
print(paste("The predictive accuracy is ", sum(continent.predict$class== test_set$continent)/dim(test_s
```

```
## [1] "The predictive accuracy is  55.8139534883721 %"
```

```r
# row label is from prediction class
# column label is from test_set class
table(continent.predict$class, test_set$continent)
```

```
##
##             Africa Americas Asia Europe
##    Africa       11        0    1      0
##    Americas      1        3    1      1
##    Asia          4        4    3      0
##    Europe        0        3    1      7
##    Oceania       0        1    0      2
```

From the result here, we see that the predictive accuracy is not high (only 55% correctly classified). One main problem for this classification is that, we use randomly splitting the data, while the dataset itself is imbalance (there are only 3 **Oceania** data, so it is possible that, all **Oceania** are split into a specific dataset).

Back to the table again, we see that **Africa** and **Europe** are classify quite well. **Americas** and **Asia** are just moderate. And there are no **Oceania** data in the test set, but still 3 data classify as Oceania.

We may claim that, continents are not linearly separable by using log(GDP) and life-expectancy.

using Fisher's discriminant analysis

Now we back to consider the whole dataset instead of splitting into training and test set

```r
library("vcvComp")


# ignoring "country" column
# here the first column (1) is the continent; the rest (2:25) are log(GDP) and life-expectancy for each
df = gap[, -2]

# between-group covariance matrix
B = cov.B(df[,2:25], df[,1])

# within-group covariance matrix
W = cov.W(df[,2:25], df[,1])

# compute the eigen-decomposition
df.eigen = eigen(solve(W)%*%B)
```

The first 2 eigenvectors are:

```
V = df.eigen$vectors[, 1:2]
```

Next we can see the 2-dimensional projection of the data

```
Z = as.matrix(df[,2:25])%*% V
continent_factor = as.factor(df[,1])
ggplot2::qplot(as.numeric(Z[,1]), as.numeric(Z[,2]),
               colour=continent_factor, xlab='LD1', ylab='LD2')
```

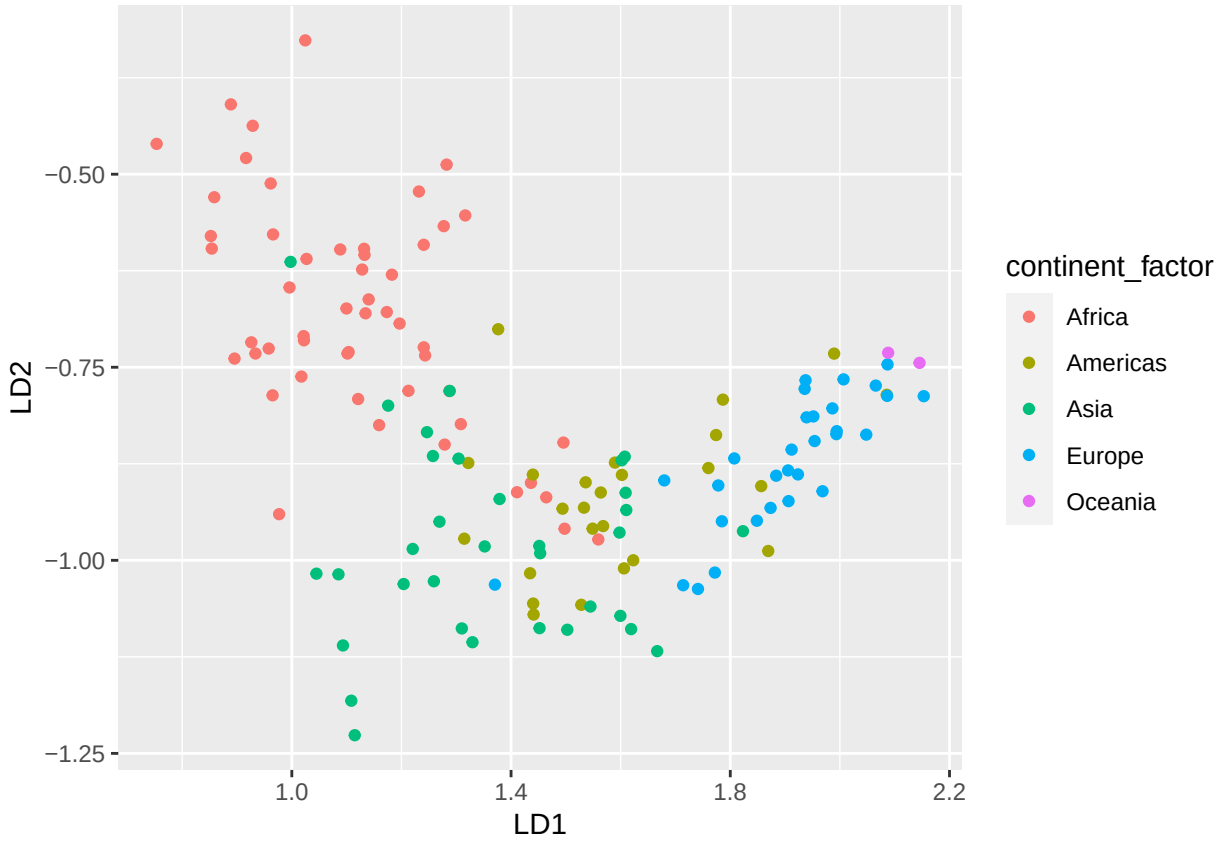


Figure 13: 2d projection of the data onto the first 2 eigenvectors

From this plot, we see that the plot is very similar to the PCA plot using the first 2 PCs on lifeExp (see Figure 3.4: Scatter plots using the first 2 Principal component on lifeExp).

Clustering

This section focus on clustering the data. Cluster analysis is a kind of unsupervised learning method, and this method aim at grouping cases into "clusters"

The coursework requirement asked to complete the following tasks:

1. K-means clustering
2. agglomerative hierarchical clustering
3. discuss the similarity of the clusters using both methods

K-means clustering

Because there are 5 continents, so the most intuitive number of clusters should be 5. However, because **Oceania** records has only 3, we may also expect that there would be fewer than 5 clusters to be grouped. So firstly starts with K=3 clusters, and then increase the number of clusters to find "optimal" number.

```
# for reproducible purpose
set.seed(1234)

# the data that contains all continent, country, logGDP and lifeExp data
data = gap

# using the logGDP and lifeExp data only
data2 = data[,3:26]

data.k = kmeans(data2, centers=3, nstart=25)
```

Then we can visualize the output

```
library("factoextra")
fviz_cluster(data.k, data = data2, geom="point")
```

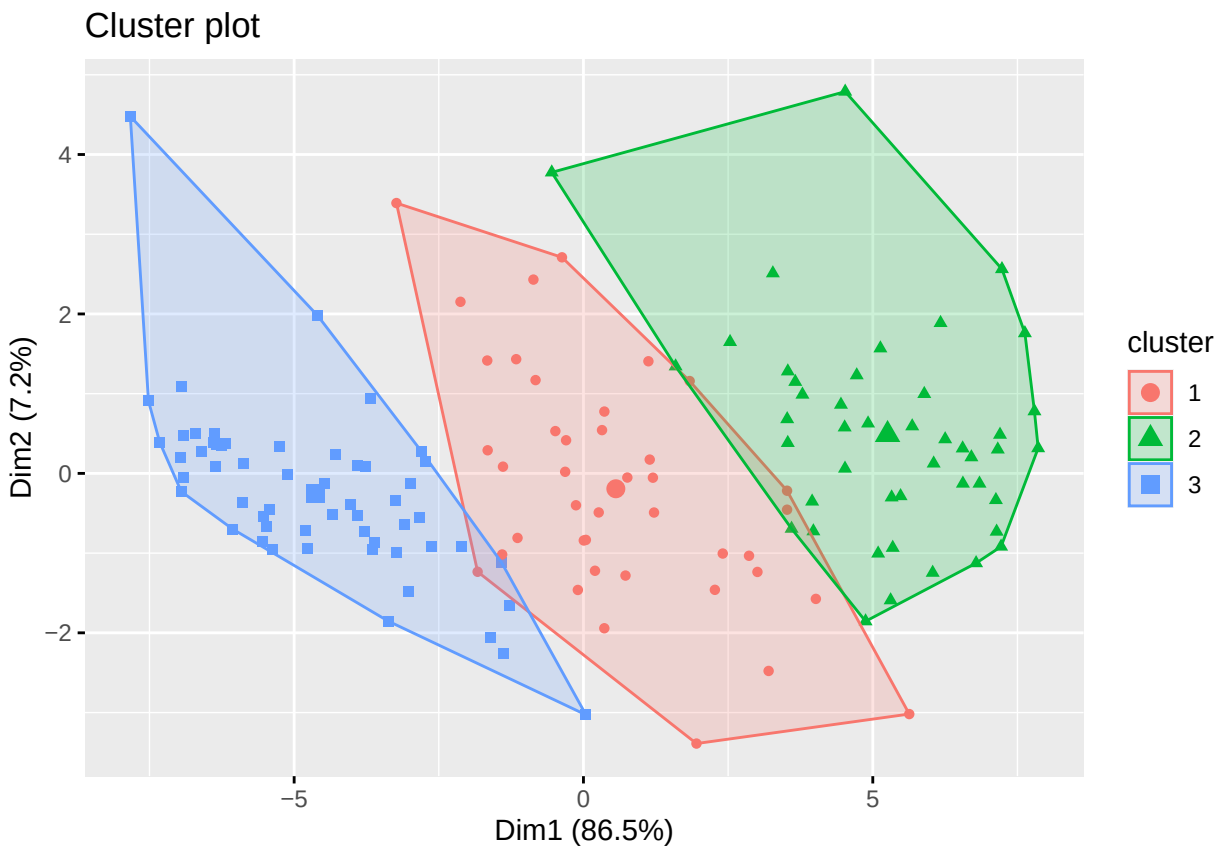


Figure 14: K-means clustering using K=3 clusters

And we can see how clusters found using K-means with the continent label to see if they are similar.

```
table(data[,1], data.k$cluster)
```

```
##
```

```
##              1  2  3
##   Africa    11 39  2
##   Americas  10  1 14
##   Asia      19  5  9
##   Europe     1  0 29
##   Oceania    0  0  2
```

Because there are lots of choice can be made, we can make use of R command to "choose" number of clusters

```
fviz_nbclust(data2, kmeans, method="wss")
```

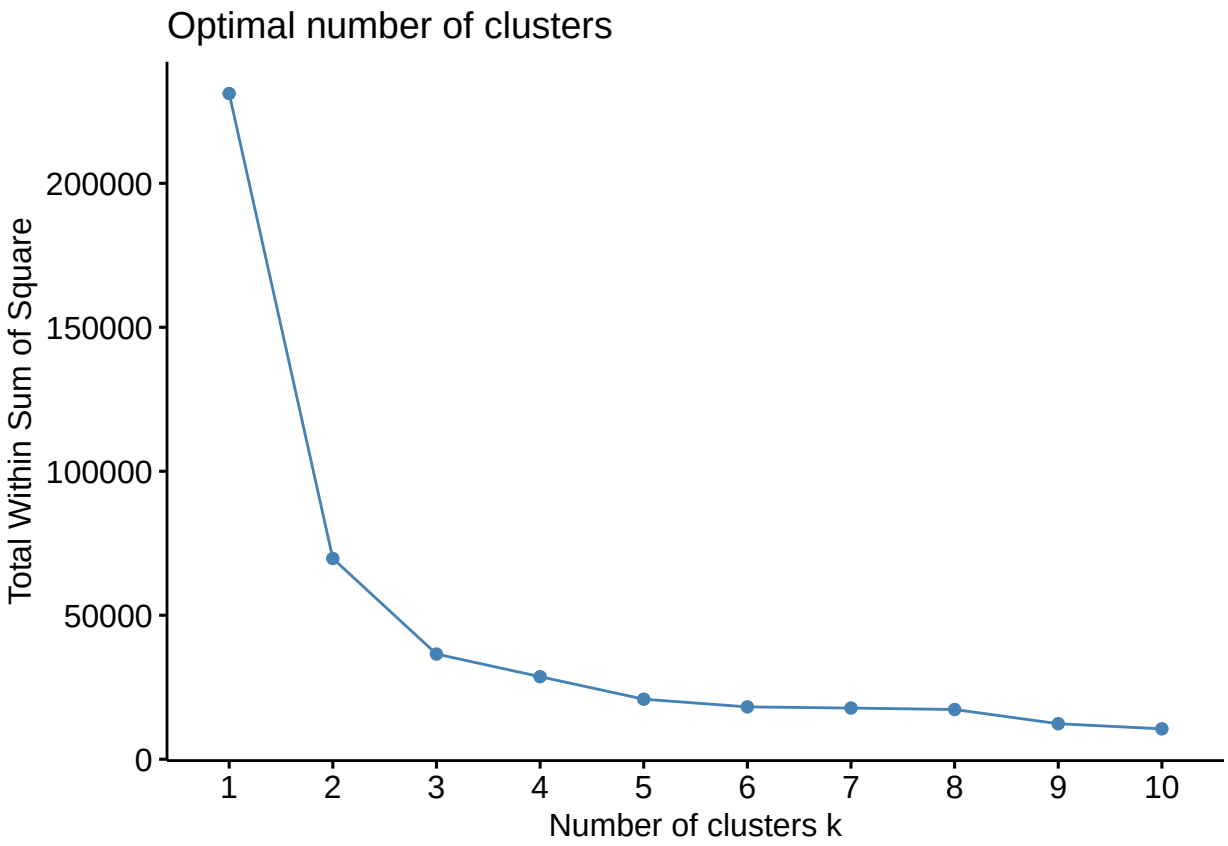
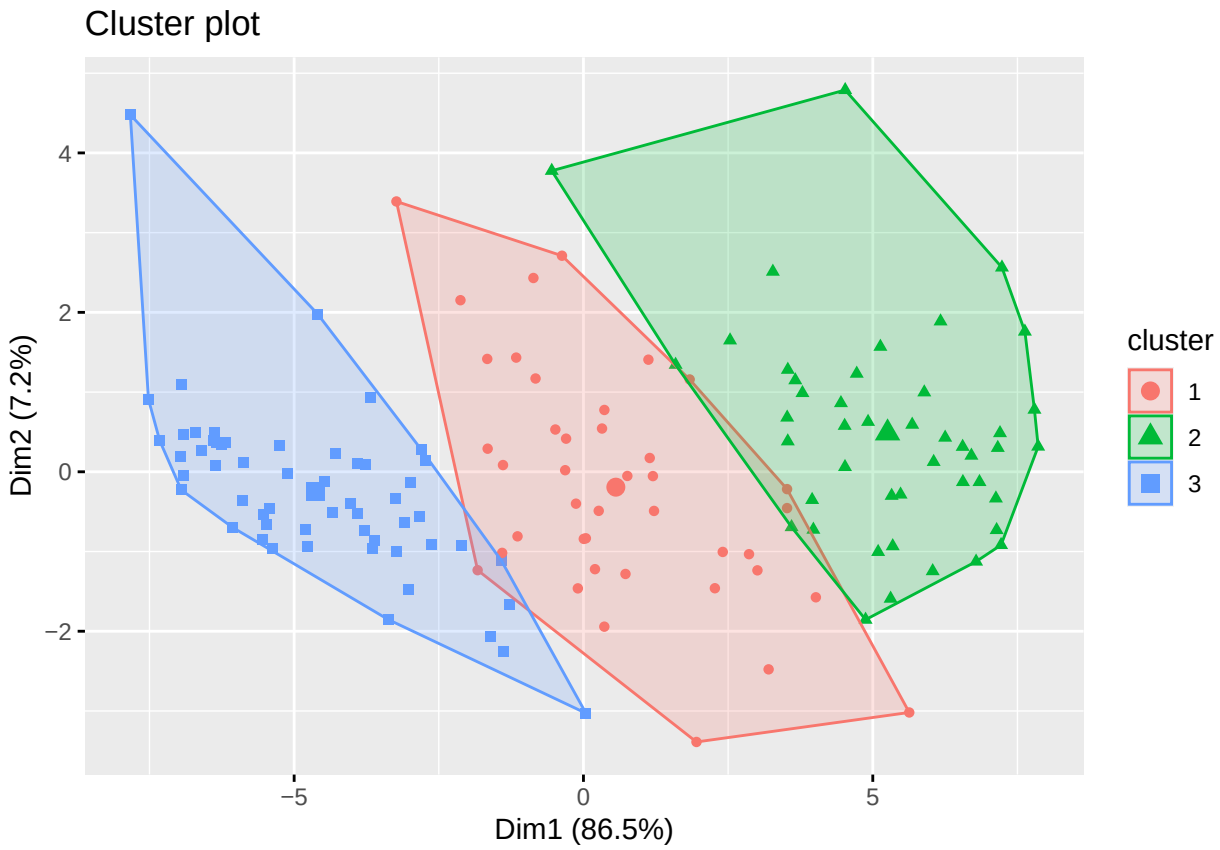


Figure 15: Within-cluster sum of squares vs number of clusters

From this plot, we see that there is a reasonable decrease in Within Sum of square when moving from 2 to 3, but only yields a minor decrease from 3 to 4. So based on this observation, I may decide to choose number of clusters = 3. So the plot final clusters can be used is

```
fviz_cluster(data.k, data = data2, geom="point")
```



Agglomerative hierarchical clustering

By the coursework requirement, I need to work on scaled data

```
# scaled data
gap.scaled = gap
gap.scaled[,3:26] = scale(gap[,3:26])

# get the distance matrix, using Euclidean distance
D = dist(gap.scaled[,3:26], method="euclidean")
```

Because there are 3 types of determining "Distances between clusters", namely

1. single linkage (SL)
2. complete linkage (CL)
3. group average (GA)

However, we can simply use R command **hclust** to help work on these

```
D.sl = hclust(D, method="single")

plot(D.sl)
```

Single linkage (SL) using Single linkage, I tried different choice of **cutree**, with k = 3, 4, 5; or h > 1, but the clustering are not performing well, as it would intended to group almost all records into a single group, with extra groups only contain 1-2 cases. So I think Single linkage is not helpful in this dataset.

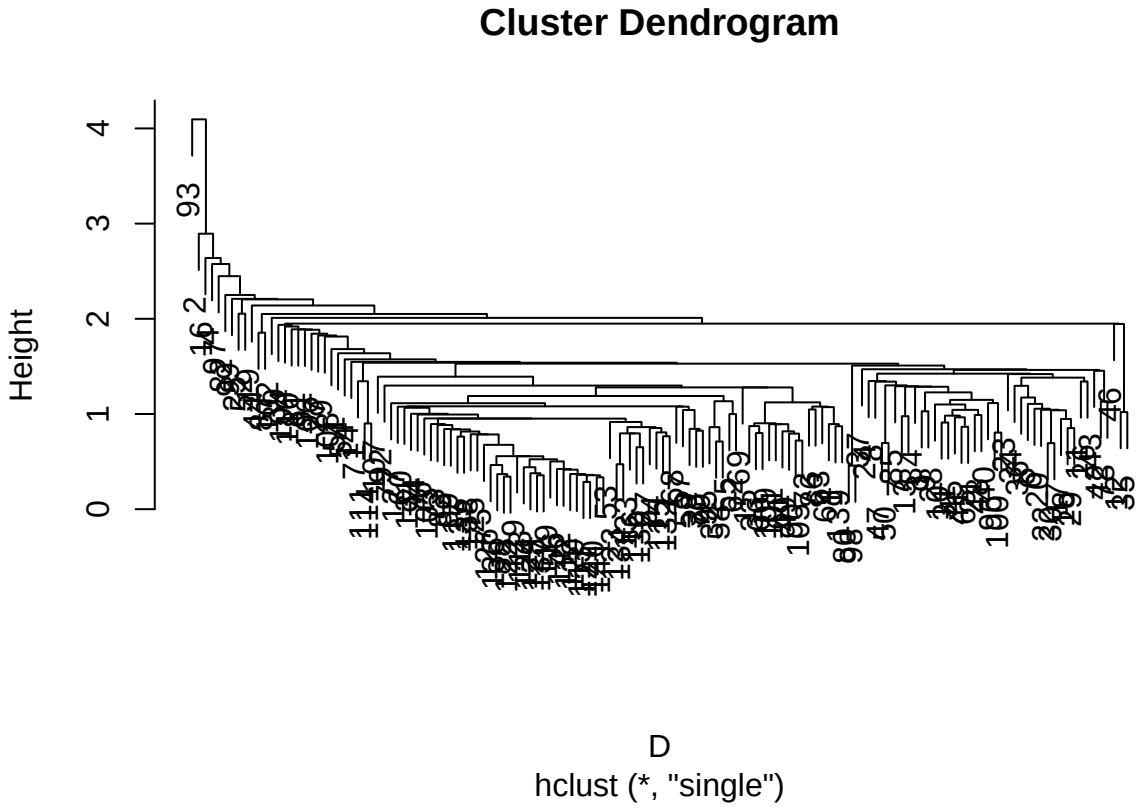


Figure 16: Cluster Dendrogram using SL

```
cutree(D.sl, k=3)
```

```
##   [1] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
table(cutree(D.sl, k=3), gap[,1])
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     51       25   32     30       2
##   2      1        0    0      0       0
##   3      0        0    1      0       0
```

```
D.cl = hclust(D, method="complete")

plot(D.cl)
```

```
# create a function to prevent replications for different k
cl = function(k){
  cutree_cl_k = cutree(D.cl, k=k)
  table(cutree(D.cl, k=k), gap[,1])
}
```

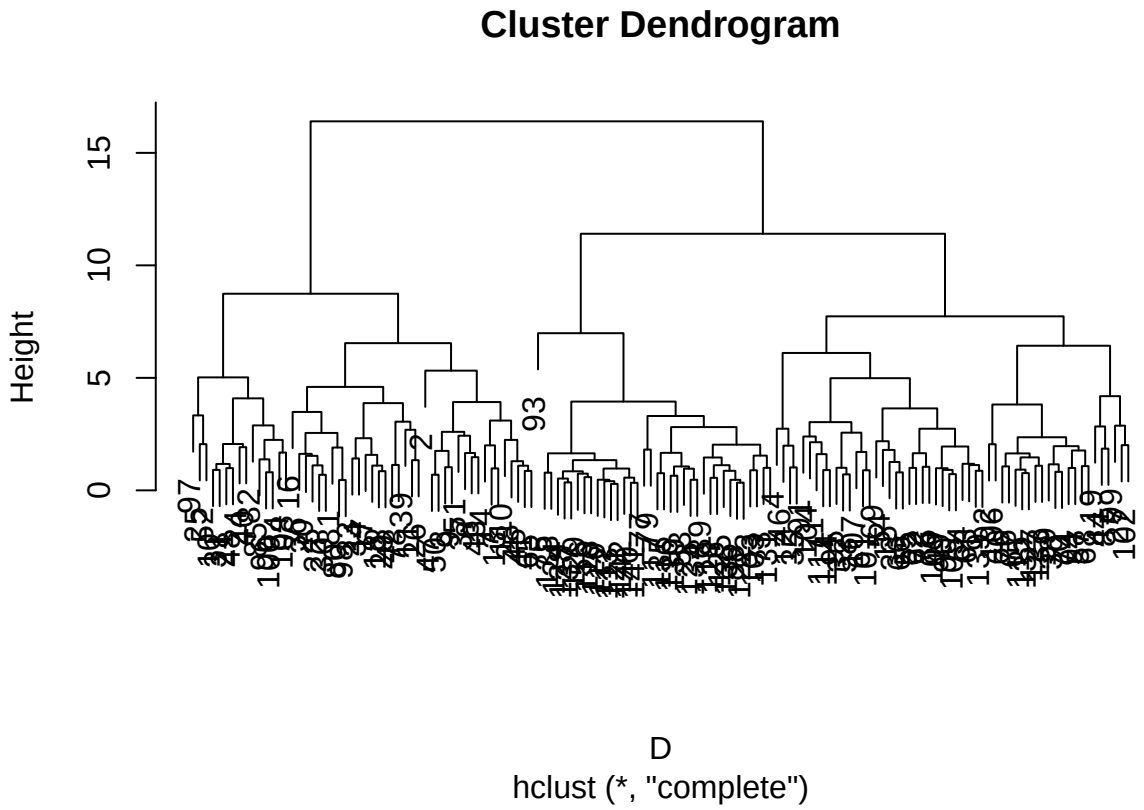


Figure 17: Cluster Dendrogram using CL

```
cl(3)
```

Complete linkage (CL)

```
##
##     Africa Americas Asia Europe Oceania
##   1     13       19   15      7       0
##   2     39        1   12      0       0
##   3      0        5    6     23       2
```

```
cl(4)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     13       19   15      7       0
##   2     31        1    5      0       0
##   3      8        0    7      0       0
##   4      0        5    6     23       2
```

```
cl(5)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1      9       11    9      3       0
##   2     31        1    5      0       0
##   3      8        0    7      0       0
##   4      4        8    6      4       0
```

```
##   5      0        5    6     23       2
```

```
cl(6)
```

```
##
##      Africa Americas Asia Europe Oceania
##   1       9       11    9      3       0
##   2      31        1    5      0       0
##   3       8        0    7      0       0
##   4       4        8    6      4       0
##   5       0        5    5     23       2
##   6       0        0    1      0       0
```

Although as an unsupervised learning method, we should not know the actual label and class that it should be "grouped" or "clustered". Anyway, when using this Complete Linkage method, this can group our data into several cluster, that each cluster contain sufficient number of cases. So I think this method is useful, but the number of clusters is depending.

Since k=6 will produce a new cluster with only 1 case in it, so k = 3 to 5 are also reasonable.

```
D.ga = hclust(D, method="average")
plot(D.ga)
```

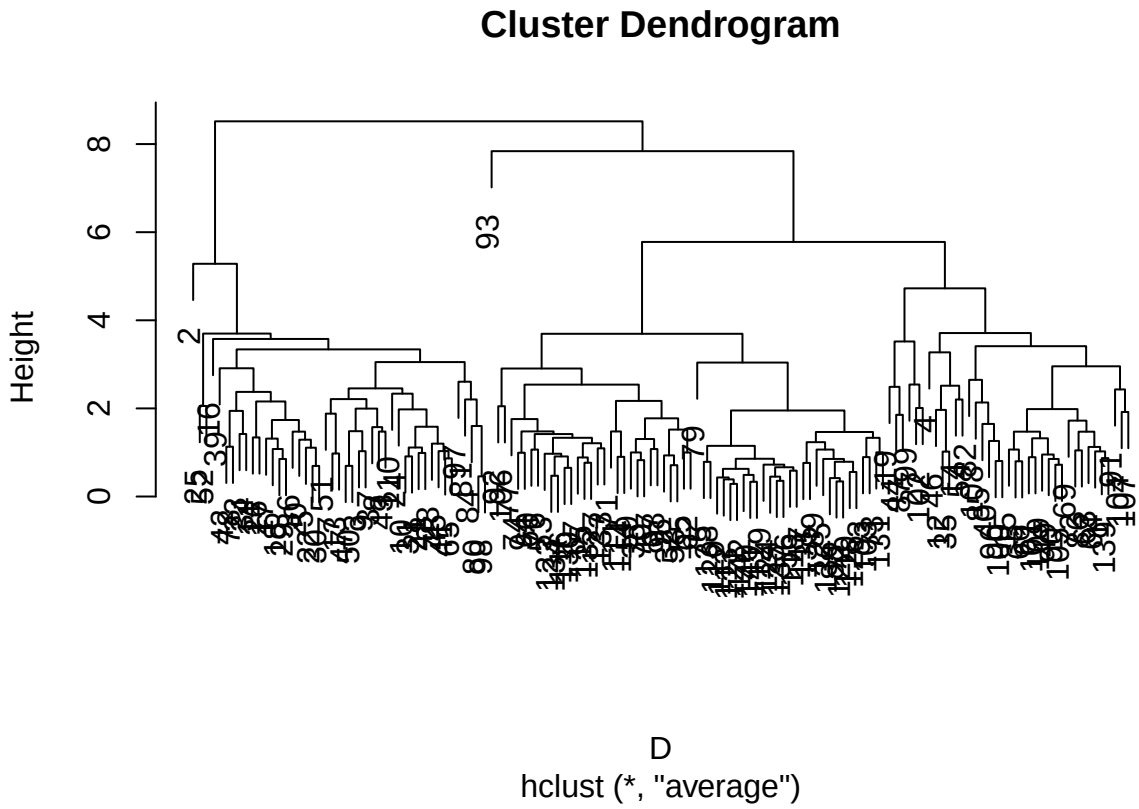


Figure 18: Cluster Dendrogram using GA

Group Average Then try to inspect number of clusters to be kept

```
# create a function to prevent replications for different k
ga = function(k){
```

```
  cutree_ga_k = cutree(D.ga, k=k)
  table(cutree(D.ga, k=k), gap[,1])
}

ga(3)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     15       24   25     30       2
##   2     37        1    7      0       0
##   3      0        0    1      0       0
```

```
ga(4)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     13        7   16      1       0
##   2     37        1    7      0       0
##   3      2       17    9     29       2
##   4      0        0    1      0       0
```

```
ga(5)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     13        7   16      1       0
##   2      1        0    0      0       0
##   3     36        1    7      0       0
##   4      2       17    9     29       2
##   5      0        0    1      0       0
```

```
ga(6)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     10        7   13      1       0
##   2      1        0    0      0       0
##   3     36        1    7      0       0
##   4      3        0    3      0       0
##   5      2       17    9     29       2
##   6      0        0    1      0       0
```

```
ga(8)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1      5        6   13      1       0
##   2      1        0    0      0       0
##   3     34        1    7      0       0
##   4      5        1    0      0       0
##   5      3        0    3      0       0
##   6      2        0    0      0       0
##   7      2       17    9     29       2
##   8      0        0    1      0       0
```

```
ga(10)
```

```
##
##       Africa Americas Asia Europe Oceania
##   1        5        6   13      1       0
##   2        1        0    0      0       0
##   3       33        1    7      0       0
##   4        5        1    0      0       0
##   5        1        0    0      0       0
##   6        3        0    3      0       0
##   7        2        0    0      0       0
##   8        2       14    4     10       0
##   9        0        3    5     19       2
##   10       0        0    1      0       0
```

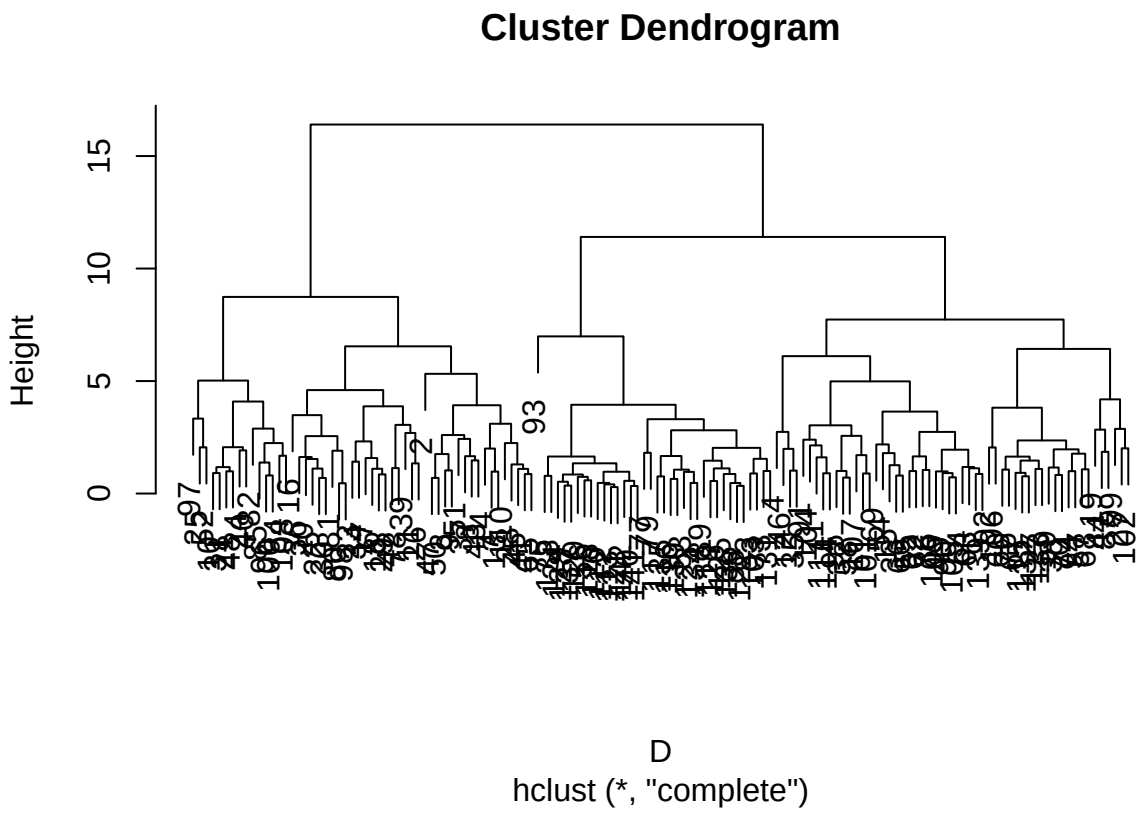
As I increase number of clusters for different **cutree** (I tried from 3 to 10), we can still see that some clusters will always contain very little cases. It may give evidence that using Group Average may not group our data very well.

Decision of different hierarchical clustering method

I would say using Complete Linkage is the best for this dataset. Not only is it can group our data into different clusters, with each contains sufficient number of cases; but also it seems have a little power to "classify" the continent (although this is not the major use-case of cluster analysis).

The dendrogram is

```
plot(D.cl)
```



and what I think is natural number of clusters is k=5 because for k = 4

```
cl(4)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     13       19   15      7       0
##   2     31        1    5      0       0
##   3      8        0    7      0       0
##   4      0        5    6     23       2
```

there are a large cluster in cluster1. And if I move on to k = 5

```
cl(5)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1      9       11    9      3       0
##   2     31        1    5      0       0
##   3      8        0    7      0       0
##   4      4        8    6      4       0
##   5      0        5    6     23       2
```

it breaks down the original cluster1, but still seem keep reasonable number of cases in each cluster. So I will choose k = 5 in this case.

Discuss the similarity of the clusters

From my findings, I choose k = 3 for k-means clustering; and k = 5 for agglomerative hierarchical clustering using Complete Linkage method.

```
# for k-means clustering with k = 3
t(table(data[,1], data.k$cluster))
```

```
##
##     Africa Americas Asia Europe Oceania
##   1     11       10   19      1       0
##   2     39        1    5      0       0
##   3      2       14    9     29       2
```

```
# hierarchical clustering using Complete Linkage
cl(5)
```

```
##
##     Africa Americas Asia Europe Oceania
##   1      9       11    9      3       0
##   2     31        1    5      0       0
##   3      8        0    7      0       0
##   4      4        8    6      4       0
##   5      0        5    6     23       2
```

Basically because I choose k = 5 for hierarchical clustering rather than 3, so we can see the clusters are breaking down a bit than k-means. However, we can still see a bit similarity between these 2, for example, both cluster2 also clustered the same distribution of continents (although we can't know whether the countries are equal based on this table). In addition, both cluster1 behave similar.

Also, we can see that both clustering method does not seem to cluster the data by continent.

Anyway as emphasized, because clustering analysis is an unsupervised learning, it intended to group "similar" records into clusters, although it is natural, we cannot assume / ensure that using continents to see the clustering is valid. Maybe the clustering method is just clustering the data by different year (say 1950-1960; 1970-1980; . . . , as the year clusters).

Linear regression

This last section required to use linear regression method to predict the 2007 life-expectancy using the GDP values. Although the question ask "GDP values" instead of "log(GDP) values", I think keep using **log** scale should be better as it makes the scale for both response and covariates more compatible.

From the lecture notes, there are 3 types of method can be used:

1. Ordinary least squares (OLS)
2. Principal Component regression (PCR)
3. Ridge regression (use of Shrinkage methods)

firstly, we should cannot use OLS in this case as after we split our data into training and test dataset, the number of records would be not large enough but there are 12 covariates, so the regression would not fit well.

To compare between PCR and Ridge regression, both are capable of fitting in a dataset with number of records is small but large number of covariates. Also, both have ability to tackle correlated covariates, which is obviously happened in our dataset.

There are still pros-and-cons between the 2 methods, I would choose Ridge regression over PCR as ridge regression keeps all covariates into the model (by the nature of the shrinkage, some parameter estimates would shrink towards 0). PCR helps to transform the dataset and use the principal components to predict the response, but it is harder to interpret than Ridge regression.

fitting Ridge regression

Ridge regression required a constant λ to control the size of the parameter estimates

```
# to confirm I can reproduce the result, I use set.seed here, but in practice this is not required
set.seed(1234)

df = cbind(lifeExp$"2007", gdp)

n = nrow(df)

# use 70%-30% split for train-test split
shuffled_df= df[sample(1:n), ]
n_train = floor(0.7*n)
n_test = n - n_train

# glmnet not supporting dataframe
training_set = as.matrix(shuffled_df[1:n_train,])
test_set = as.matrix(shuffled_df[(n_train+1): n,])
```

```
lambdas = 10^seq(3, -4, by=-.1)


# glmnet library
library("glmnet")

# response is the 2007 life-expectancy (first column)
# "covariates" are the log(GDP) values (from 2:13)
lifeExp_2007.train_ridge = glmnet(training_set[,2:13], training_set[,1], alpha = 0, lambda = lambdas)
```

Then we can check the parameter estimates change as the size of λ changes

```
plot(lifeExp_2007.train_ridge, xvar="lambda")
```

Then we can check which lambda works best for minimizing prediction error for the training data

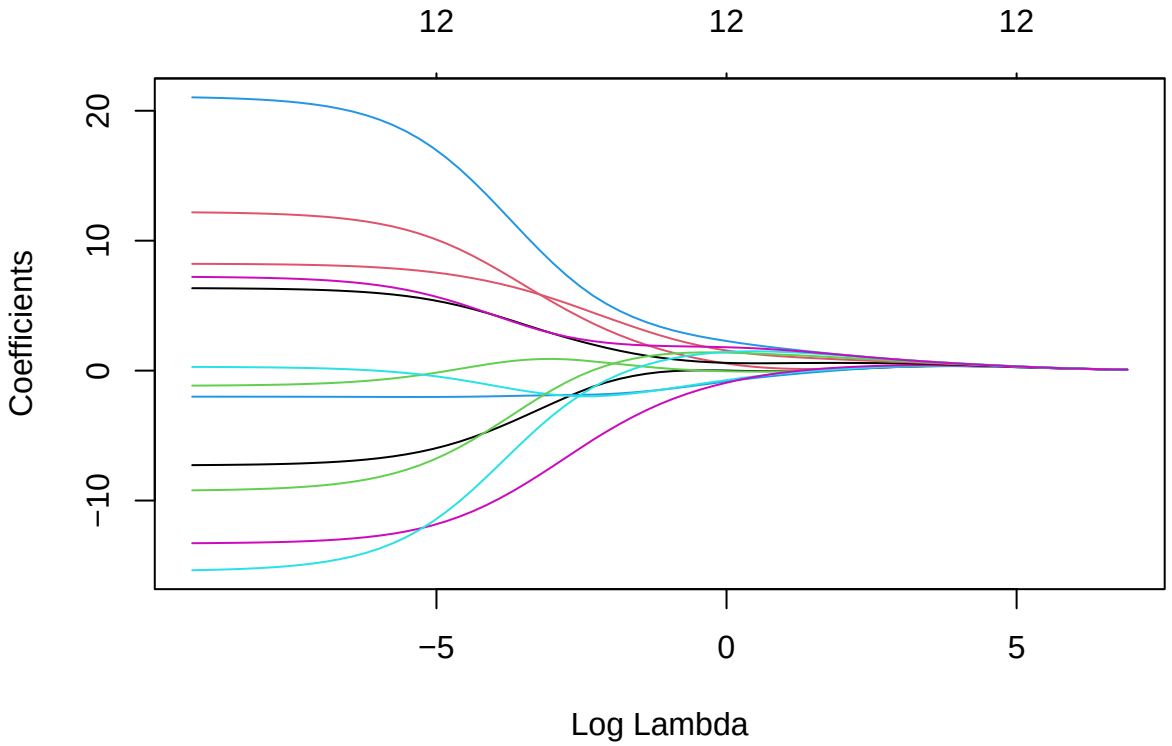


Figure 19: Ridge regression coefficient changes against log lambda

```
cv_fit <- cv.glmnet(training_set[,2:13], training_set[,1], alpha = 0, lambda = lambdas)
plot(cv_fit)
```

And the value of lambda that minimizes the prediction error is

```
cv_fit$lambda.min
```

```
## [1] 1.584893
```

Hence, I can use this lambda value 1.5848932‘ for my Ridge regression

```
lifeExp_2007.ridge = glmnet(training_set[,2:13], training_set[,1], alpha = 0, lambda = cv_fit$lambda.mi
```

and the prediction error

```
predicted_value = predict(lifeExp_2007.ridge, test_set[,2:13])

result = cbind(test_set[,1], predicted_value)

colnames(result) = c("true value", "predicted value")

result
```

```
##                       true value predicted value
## Congo Dem. Rep.           46.462        48.44538
## Puerto Rico               78.746        76.73740
## Turkey                    71.777        70.26659
## Japan                     82.603        80.48826
## Saudi Arabia              72.777        77.90822
## Mexico                    76.195        73.30790
```

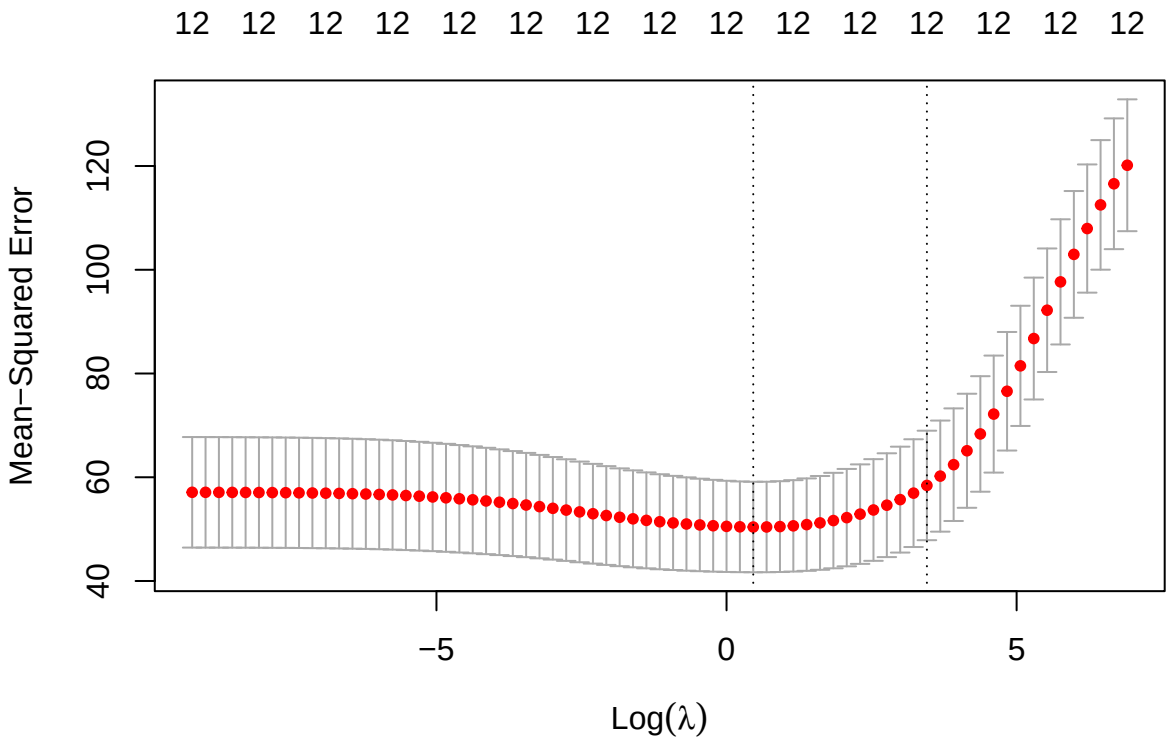


Figure 20: MSE against log(lambda)

```
## Swaziland                39.613        66.43180
## Costa Rica               78.782        70.60399
## Haiti                    60.916        58.87188
## Slovenia                 77.926        77.73522
## Malawi                   48.303        53.88750
## Bosnia and Herzegovina   74.852        68.21361
## Korea Rep.               78.623        77.34262
## Uruguay                  76.384        72.21522
## South Africa             49.339        71.13994
## Sudan                    58.556        60.84495
## Nigeria                  46.859        60.22078
## Kenya                    54.110        58.95706
## United Kingdom           79.425        80.07904
## Finland                  79.313        79.83823
## Paraguay                 71.752        67.05001
## Egypt                    71.338        67.81556
## Netherlands              79.762        80.94436
## Norway                   80.196        83.20257
## Guinea-Bissau            46.388        53.75427
## Cameroon                 50.430        61.63508
## Rwanda                   46.242        54.97672
## Colombia                 72.889        69.64845
## Congo Rep.               55.322        66.19768
## Morocco                  71.164        65.18940
## Iran                     70.964        71.39685
## United States            78.242        82.17212
## Oman                     75.640        78.22920
```

```
## Austria                  79.829        80.93992
## Ethiopia                 52.947        52.28715
## Iceland                  81.757        80.94201
## Nicaragua                72.899        62.19927
## Bulgaria                 73.005        70.89053
## Sri Lanka                72.396        64.38082
## Guatemala                70.259        67.63266
## Mozambique               42.082        51.96164
## Trinidad and Tobago      69.819        73.33130
## Mauritania               64.164        59.48970
```

with Mean square error (MSE)

```
MSE = sum((result[,1]-result[,2])^2) / nrow(result)
MSE
```

```
## [1] 56.49671
```

and the root mean square error (RMSE) is

```
RMSE = sqrt(MSE)
RMSE
```

```
## [1] 7.516429
```

Generally low RMSE value indicates that the model is good. Since the RMSE value is 6.75 and the true value of our data is approximately ranging from 40 to 80. So it may be a good sign that our ridge regression model is good.

And finally, the parameter estimates of our regression model is

```
coef(lifeExp_2007.ridge)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  8.15345903
## 1952        -0.01639041
## 1957         0.31252012
## 1962        -0.07548056
## 1967        -0.59152059
## 1972        -0.47521231
## 1977        -0.51196352
## 1982         0.56271466
## 1987         1.28917815
## 1992         1.32889659
## 1997         1.98432924
## 2002         1.48811011
## 2007         1.72028977
```