Lab2 - Assignment2

Fei Xie

Lab2- Part1: 2a, 2b

```r
library(readr)
library(car)
```

```
## Loading required package: carData
```

```r
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(gridExtra)
library(MASS)
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 3.0-2
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```r
library(tidyverse)
```

```
## -- Attaching packages -------------------------------------------------------------------------
```

```
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v purrr   0.3.3     v forcats 0.4.0
```

```
## -- Conflicts ----------------------------------------------------------------------------------
## x dplyr::combine() masks gridExtra::combine()
## x tidyr::expand()  masks Matrix::expand()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
## x tidyr::pack()    masks Matrix::pack()
## x dplyr::recode()  masks car::recode()
## x dplyr::select()  masks MASS::select()
## x purrr::some()    masks car::some()
## x tidyr::unpack()  masks Matrix::unpack()
```

```r
library(dplyr) # sample_n(), sample_frac(), arrang(), summerise(), %>% (pipe) (ref:https://datacarpentr
```

Lab2a. Measures of Central Tendency/Histograms/ Data Manipulation:

Generate Central Tendency values for EPI and DALY variable

Note: I used the EPI/EPI_data.csv under https://aquarius.tw.rpi.edu/html/DA/EPI/

```r
data <- read_csv("EPI_data.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   ISO3V10 = col_character(),
##   Country = col_character(),
##   EPI_regions = col_character(),
##   GEO_subregion = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```r
# data %>% glimpse()
attach(data)
```

```r
# summary() shows the mean, median, and quantiles for numeric variables in a data frame
summary(EPI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   32.10   48.60   59.20   58.37   67.60   93.50      68
```

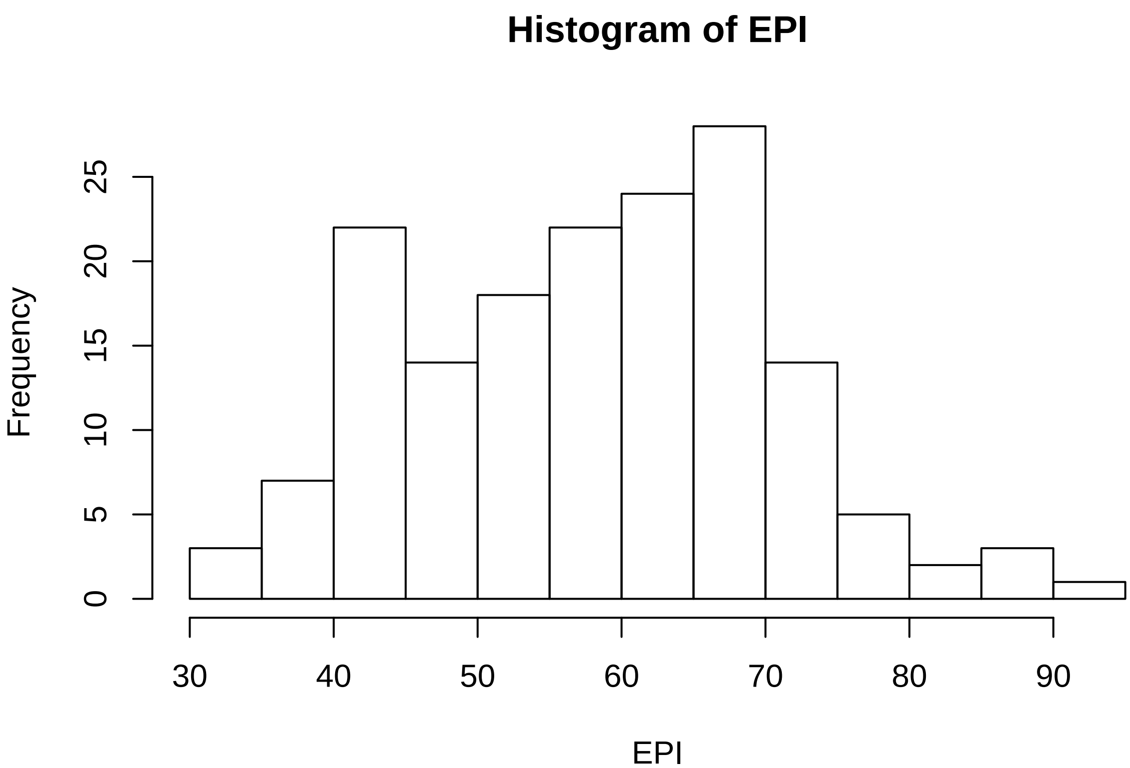
```r
summary(DALY)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   37.19   60.35   53.94   71.97   91.50      39
```

Generate the Histogram for EPI and DALY variables

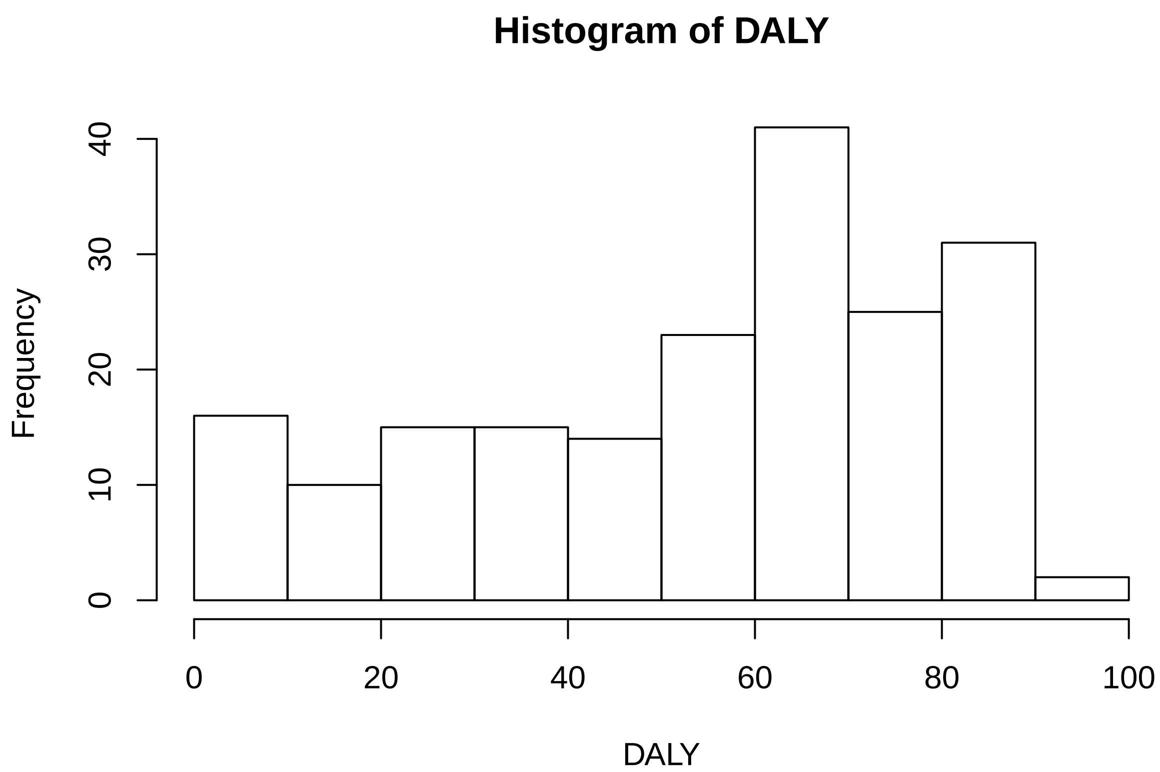
Generate the Histogram for EPI variable

```r
hist(EPI)
```



####
Generate the Histogram for DALY variable

```r
hist(DALY)
```



Data Manipulation with Dplyr

```
df_EPI = data.frame(EPI)
df_DALY = data.frame(DALY)
# (1) sample_n() ==> pick random number of rows that we wish to choose:
# random 5 rows
sample_n(df_EPI, 5)
```

```
##    EPI
## 1 63.5
## 2 58.0
## 3 71.4
## 4 86.0
## 5 33.3
```

```
sample_n(df_DALY, 5)
```

```
##    DALY
## 1 33.86
## 2 63.34
## 3 86.86
## 4 58.50
## 5    NA
```

```r
# (2) sample_frac() ==> pick a percentage of rows
# sample with a 10% of rows from the total number of rows
sample_frac(df_EPI, 0.1)
```

```
##     EPI
## 1  60.4
## 2  72.5
## 3  68.2
## 4  56.4
## 5  49.9
## 6  55.3
## 7    NA
## 8    NA
## 9  47.0
## 10 62.9
## 11 48.9
## 12 51.1
## 13 69.2
## 14 67.3
## 15 33.3
## 16   NA
## 17 64.6
## 18 48.3
## 19   NA
## 20 65.6
## 21 93.5
## 22   NA
## 23 49.0
```

```r
sample_frac(df_DALY, 0.1)
```

```
##     DALY
## 1  10.17
## 2  86.86
## 3  84.77
## 4  39.85
## 5  27.06
## 6  54.28
## 7  73.01
## 8     NA
## 9  58.50
## 10 71.63
## 11 27.75
## 12 91.50
## 13 89.10
## 14 31.43
## 15 47.21
## 16 39.35
## 17 67.82
## 18    NA
## 19 80.96
## 20 19.76
```

```
## 21 66.64
## 22 43.04
## 23 16.40
```

```r
# (4)  arrange() and desc() ==> arrange values in the descending order in the EPI and DALY
new_decs_EPI  <- arrange( data, desc(EPI) )
new_decs_DALY <- arrange( data, desc(DALY) )
```

```r
new_decs_EPI
```

```
## # A tibble: 231 x 160
##     code ISO3V10 Country EPI_regions GEO_subregion GDPCAP07 Population07
##    <dbl> <chr>   <chr>   <chr>       <chr>            <dbl>        <dbl>
##  1   352 ISL     Iceland Europe      Western Euro~   36118.      310997
##  2   756 CHE     Switze~ Europe      Western Euro~   37581.     7550077
##  3   188 CRI     Costa ~ Latin Amer~ Meso America    10239.     4462193.
##  4   752 SWE     Sweden  Europe      Western Euro~   34090.     9148092
##  5   578 NOR     Norway  Europe      Western Euro~   49359.     4709153
##  6   480 MUS     Maurit~ Sub-Sahara~ Western Indi~   10668.     1260692
##  7   250 FRA     France  Europe      Western Euro~   31625.    61707072
##  8    40 AUT     Austria Europe      Western Euro~   35537.     8315427
##  9   192 CUB     Cuba    Latin Amer~ Caribbean        9100     11257013.
## 10   170 COL     Colomb~ Latin Amer~ South America    8109.    43987000
## # ... with 221 more rows, and 153 more variables: Landarea <dbl>,
## #   PopulationDensity <dbl>, Landlock <dbl>, No_surface_water <dbl>,
## #   Desert <dbl>, High_Population_Density <dbl>, EPI <dbl>, ENVHEALTH <dbl>,
## #   ECOSYSTEM <dbl>, DALY <dbl>, AIR_H <dbl>, WATER_H <dbl>, AIR_E <dbl>,
## #   WATER_E <dbl>, BIODIVERSITY <dbl>, FORESTRY <dbl>, FISHERIES <dbl>,
## #   AGRICULTURE <dbl>, CLIMATE <dbl>, DALY_pt <dbl>, ACSAT_pt <dbl>,
## #   ACSAT_pt_imp <dbl>, WATSUP_pt <dbl>, WATSUP_pt_imp <dbl>, INDOOR_pt <dbl>,
## #   PM10_pt <dbl>, SO2_pt <dbl>, NOX_pt <dbl>, NMVOC_pt <dbl>, OZONE_pt <dbl>,
## #   WQI_pt <dbl>, WQI_pt_imp <dbl>, `WQI_pt_GEMS station data` <dbl>,
## #   WSI_pt <dbl>, WATSTR_pt <dbl>, PACOV_pt <dbl>, MPAEEZ_pt <dbl>,
## #   AZE_pt <dbl>, FORGRO_pt <dbl>, FORCOV_pt <dbl>, MTI_pt <dbl>,
## #   EEZTD_pt <dbl>, AGWAT_pt <dbl>, AGSUB_pt <dbl>, AGPEST_pt <dbl>,
## #   GHGCAP_pt <dbl>, GHGCAP_pt_imp <dbl>, GHGIND_pt <dbl>, CO2KWH_pt <dbl>,
## #   CO2KWH_pt_imp <dbl>, DALY_raw <dbl>, ACSAT_raw <dbl>, ACSAT_raw_imp <dbl>,
## #   WATSUP_raw <dbl>, WATSUP_raw_imp <dbl>, INDOOR_raw <dbl>, PM10_raw <dbl>,
## #   OZONE_raw <dbl>, WQI_raw <dbl>, WQI_raw_imp <dbl>, `WQI_raw_GEMS station
## #   data` <dbl>, SO2_raw <dbl>, NOX_raw <dbl>, NMVOC_raw <dbl>, WSI_raw <dbl>,
## #   WATSTR_raw <dbl>, PACOV_raw <dbl>, AZE_raw <dbl>, MPAEEZ_raw <dbl>,
## #   FORGRO_raw <dbl>, FORCOV_raw <dbl>, MTI_raw <dbl>, EEZTD_raw <dbl>,
## #   AGWAT_raw <dbl>, AGSUB_raw <dbl>, AGPEST_raw <dbl>, GHGCAP_raw <dbl>,
## #   GHGCAP_raw_imp <dbl>, GHGIND_raw <dbl>, CO2KWH_raw <dbl>,
## #   CO2KWH_raw_imp <dbl>, DALY_w <dbl>, ACSAT_w <dbl>, WATSUP_w <dbl>,
## #   INDOOR_w <dbl>, PM10_w <dbl>, OZONE_w <dbl>, SO2_w <dbl>, NOX_w <dbl>,
## #   NMVOC_w <dbl>, WSI_w <dbl>, WATSTR_w <dbl>, PACOV_w <dbl>, AZE_w <dbl>,
## #   MPAEEZ_w <dbl>, FORGRO_w <dbl>, FORCOV_w <dbl>, MTI_w <dbl>, EEZTD_w <dbl>,
## #   AGWAT_w <dbl>, ...
```

```r
new_decs_DALY
```

```
## # A tibble: 231 x 160
```

```
##     code ISO3V10 Country EPI_regions GEO_subregion GDPCAP07 Population07
##    <dbl> <chr>   <chr>   <chr>       <chr>            <dbl>        <dbl>
##  1   352 ISL     Iceland Europe      Western Euro~   36118.      310997
##  2   376 ISR     Israel  Middle Eas~ Western Euro~   24824.     7180100
##  3   784 ARE     United~ Middle Eas~ Arabian Peni~   51586.     4364746.
##  4   756 CHE     Switze~ Europe      Western Euro~   37581.     7550077
##  5   414 KWT     Kuwait  Middle Eas~ Arabian Peni~   45152.     2662966.
##  6   634 QAT     Qatar   Middle Eas~ Arabian Peni~   99100      1137553
##  7   702 SGP     Singap~ East Asia ~ South East A~   47497.     4588600
##  8    40 AUT     Austria Europe      Western Euro~   35537.     8315427
##  9    96 BRN     Brunei~ East Asia ~ South East A~   47407.      389252.
## 10   124 CAN     Canada  North Amer~ North America   36260.    32976000
## # ... with 221 more rows, and 153 more variables: Landarea <dbl>,
## #   PopulationDensity <dbl>, Landlock <dbl>, No_surface_water <dbl>,
## #   Desert <dbl>, High_Population_Density <dbl>, EPI <dbl>, ENVHEALTH <dbl>,
## #   ECOSYSTEM <dbl>, DALY <dbl>, AIR_H <dbl>, WATER_H <dbl>, AIR_E <dbl>,
## #   WATER_E <dbl>, BIODIVERSITY <dbl>, FORESTRY <dbl>, FISHERIES <dbl>,
## #   AGRICULTURE <dbl>, CLIMATE <dbl>, DALY_pt <dbl>, ACSAT_pt <dbl>,
## #   ACSAT_pt_imp <dbl>, WATSUP_pt <dbl>, WATSUP_pt_imp <dbl>, INDOOR_pt <dbl>,
## #   PM10_pt <dbl>, SO2_pt <dbl>, NOX_pt <dbl>, NMVOC_pt <dbl>, OZONE_pt <dbl>,
## #   WQI_pt <dbl>, WQI_pt_imp <dbl>, `WQI_pt_GEMS station data` <dbl>,
## #   WSI_pt <dbl>, WATSTR_pt <dbl>, PACOV_pt <dbl>, MPAEEZ_pt <dbl>,
## #   AZE_pt <dbl>, FORGRO_pt <dbl>, FORCOV_pt <dbl>, MTI_pt <dbl>,
## #   EEZTD_pt <dbl>, AGWAT_pt <dbl>, AGSUB_pt <dbl>, AGPEST_pt <dbl>,
## #   GHGCAP_pt <dbl>, GHGCAP_pt_imp <dbl>, GHGIND_pt <dbl>, CO2KWH_pt <dbl>,
## #   CO2KWH_pt_imp <dbl>, DALY_raw <dbl>, ACSAT_raw <dbl>, ACSAT_raw_imp <dbl>,
## #   WATSUP_raw <dbl>, WATSUP_raw_imp <dbl>, INDOOR_raw <dbl>, PM10_raw <dbl>,
## #   OZONE_raw <dbl>, WQI_raw <dbl>, WQI_raw_imp <dbl>, `WQI_raw_GEMS station
## #   data` <dbl>, SO2_raw <dbl>, NOX_raw <dbl>, NMVOC_raw <dbl>, WSI_raw <dbl>,
## #   WATSTR_raw <dbl>, PACOV_raw <dbl>, AZE_raw <dbl>, MPAEEZ_raw <dbl>,
## #   FORGRO_raw <dbl>, FORCOV_raw <dbl>, MTI_raw <dbl>, EEZTD_raw <dbl>,
## #   AGWAT_raw <dbl>, AGSUB_raw <dbl>, AGPEST_raw <dbl>, GHGCAP_raw <dbl>,
## #   GHGCAP_raw_imp <dbl>, GHGIND_raw <dbl>, CO2KWH_raw <dbl>,
## #   CO2KWH_raw_imp <dbl>, DALY_w <dbl>, ACSAT_w <dbl>, WATSUP_w <dbl>,
## #   INDOOR_w <dbl>, PM10_w <dbl>, OZONE_w <dbl>, SO2_w <dbl>, NOX_w <dbl>,
## #   NMVOC_w <dbl>, WSI_w <dbl>, WATSTR_w <dbl>, PACOV_w <dbl>, AZE_w <dbl>,
## #   MPAEEZ_w <dbl>, FORGRO_w <dbl>, FORCOV_w <dbl>, MTI_w <dbl>, EEZTD_w <dbl>,
## #   AGWAT_w <dbl>, ...
```

```r
# (5) mutate() ==> create new columns (ref: https://www.sharpsightlabs.com/blog/add-a-column-to-a-dataf
# (ref: https://cengel.github.io/R-data-wrangling/dplyr.html)

# in additing to selecting sets of existing columns in the dataframe, sometimes
# we need to add new columns that are functions of existing columns in the dataframe.
# we can use the mutate() function to do that.
data %>% mutate(double_EPI = EPI * 2) %>% head() %>% glimpse()
```

```
## Observations: 6
## Variables: 161
## $ code                          <dbl> 533, 4, 24, 660, 8, 20
## $ ISO3V10                       <chr> "ABW", "AFG", "AGO", "AIA", "ALB", "AND"
## $ Country                       <chr> "Aruba", "Afghanistan", "Angola", "Angu...
## $ EPI_regions                   <chr> "Latin America and Caribbean", "South A...
## $ GEO_subregion                 <chr> "Caribbean", "South Asia", "Southern Af...
```

```
## $ GDPCAP07                   <dbl> NA, NA, 4875.36, NA, 6811.38, NA
## $ Population07               <dbl> 104176, NA, 17554585, NA, 3132458, 82180
## $ Landarea                   <dbl> 189.12, 634924.74, 1251895.62, 82.83, 2...
## $ PopulationDensity          <dbl> 550.85, NA, 14.02, NA, 110.51, 177.19
## $ Landlock                   <dbl> 0, 1, 0, 0, 0, 1
## $ No_surface_water           <dbl> 0, 0, 0, 0, 0, 0
## $ Desert                     <dbl> 0, 1, 0, 1, 0, 0
## $ High_Population_Density    <dbl> 1, 0, 0, 1, 0, 1
## $ EPI                        <dbl> NA, NA, 36.3, NA, 71.4, NA
## $ ENVHEALTH                  <dbl> NA, 11.55, 18.29, NA, 69.93, 90.21
## $ ECOSYSTEM                  <dbl> NA, NA, 54.40, NA, 72.92, NA
## $ DALY                       <dbl> NA, 0.00, 0.00, NA, 65.50, 84.77
## $ AIR_H                      <dbl> NA, 35.49, 43.47, NA, 52.97, 91.28
## $ WATER_H                    <dbl> 100.00, 10.72, 29.70, NA, 95.73, 100.00
## $ AIR_E                      <dbl> 33.13, 72.03, 40.13, 86.54, 49.16, 52.41
## $ WATER_E                    <dbl> NA, 57.43, 64.76, NA, 91.24, NA
## $ BIODIVERSITY               <dbl> 0.23, 3.11, 58.43, 0.26, 77.02, 57.16
## $ FORESTRY                   <dbl> 100.00, 22.63, 94.79, 100.00, 100.00, 1...
## $ FISHERIES                  <dbl> 92.86, NA, 86.74, NA, 62.54, NA
## $ AGRICULTURE                <dbl> 40.00, 39.59, 54.55, 40.00, 54.55, 40.00
## $ CLIMATE                    <dbl> NA, NA, 53.85, NA, 68.97, NA
## $ DALY_pt                    <dbl> NA, 0.00000, 0.00000, NA, 65.50225, 84....
## $ ACSAT_pt                   <dbl> NA, 21.43659, 43.88328, NA, 96.63300, 1...
## $ ACSAT_pt_imp               <dbl> 0, 0, 0, 0, 0, 0
## $ WATSUP_pt                  <dbl> 100.00000, 0.00000, 15.51724, NA, 94.82...
## $ WATSUP_pt_imp              <dbl> 0, 0, 0, 0, 0, 0
## $ INDOOR_pt                  <dbl> NA, 9.168421, 49.747368, NA, 47.368421,...
## $ PM10_pt                    <dbl> NA, 61.81838, 37.18680, NA, 58.56530, 8...
## $ SO2_pt                     <dbl> 17.63125, 80.49462, 56.46289, 100.00000...
## $ NOX_pt                     <dbl> 17.02643, 92.98855, 40.77051, 66.43237,...
## $ NMVOC_pt                   <dbl> 28.83952, 58.79643, 30.60573, 52.78802,...
## $ OZONE_pt                   <dbl> 100.00000, 38.89569, 0.00000, 100.00000...
## $ WQI_pt                     <dbl> 48.00000, 44.80000, 51.80000, NA, 82.47...
## $ WQI_pt_imp                 <dbl> 1, 1, 1, 0, 0, 1
## $ `WQI_pt_GEMS station data`  <dbl> NA, NA, NA, NA, 82.47194, NA
## $ WSI_pt                     <dbl> NA, 100, 100, NA, 100, NA
## $ WATSTR_pt                  <dbl> NA, 40.17494, 55.35011, NA, 100.00000, NA
## $ PACOV_pt                   <dbl> 0.306, 4.145, 98.368, 0.000, 96.279, 57...
## $ MPAEEZ_pt                  <dbl> 0.00683877, NA, 36.96774581, 1.03444056...
## $ AZE_pt                     <dbl> NA, 0, 0, NA, NA, NA
## $ FORGRO_pt                  <dbl> NA, 41.6748, 95.8012, NA, 100.0000, NA
## $ FORCOV_pt                  <dbl> 100.000000, 3.576983, 93.779160, 100.00...
## $ MTI_pt                     <dbl> 100.000, NA, 98.961, NA, 100.000, NA
## $ EEZTD_pt                   <dbl> 85.72373, NA, 74.51304, 95.29017, 25.08...
## $ AGWAT_pt                   <dbl> NA, 47.95721, 100.00000, NA, 100.00000, NA
## $ AGSUB_pt                   <dbl> 100, 100, 100, 100, 100, 100
## $ AGPEST_pt                  <dbl> 0.000000, 0.000000, 9.090909, 0.000000,...
## $ GHGCAP_pt                  <dbl> NA, 93.3000, 37.8481, NA, 70.5000, NA
## $ GHGCAP_pt_imp              <dbl> 0, 1, 0, 0, 1, 0
## $ GHGIND_pt                  <dbl> NA, 100.00000, 100.00000, NA, 66.91523, NA
## $ CO2KWH_pt                  <dbl> NA, NA, 39.68988, NA, 68.00976, NA
## $ CO2KWH_pt_imp              <dbl> 0, 0, 0, 0, 0, 0
## $ DALY_raw                   <dbl> NA, 255, 288, NA, 29, 16
## $ ACSAT_raw                  <dbl> NA, 30, 50, NA, 97, 100
```

```
## $ ACSAT_raw_imp               <dbl> 0, 0, 0, 0, 0, 0
## $ WATSUP_raw                  <dbl> 100, 22, 51, NA, 97, 100
## $ WATSUP_raw_imp              <dbl> 0, 0, 0, 0, 0, 0
## $ INDOOR_raw                  <dbl> NA, 86.29, 47.74, NA, 50.00, 5.00
## $ PM10_raw                    <dbl> NA, 41.26848, 65.85132, NA, 43.89564, 2...
## $ OZONE_raw                   <dbl> 0.00000e+00, 1.83308e+05, 1.36433e+09, ...
## $ WQI_raw                     <dbl> 48.00000, 44.80000, 51.80000, NA, 82.47...
## $ WQI_raw_imp                 <dbl> 1, 1, 1, 0, 0, 1
## $ `WQI_raw_GEMS station data` <dbl> NA, NA, NA, NA, 82.47194, NA
## $ SO2_raw                     <dbl> 27.566208, 0.065268, 0.658351, 0.000109...
## $ NOX_raw                     <dbl> 26.278820, 0.019452, 2.760811, 0.241785...
## $ NMVOC_raw                   <dbl> 6.374132, 0.420557, 5.430189, 0.725463,...
## $ WSI_raw                     <dbl> NA, 0, 0, NA, 0, NA
## $ WATSTR_raw                  <dbl> NA, 11.28, 5.50, NA, 0.00, NA
## $ PACOV_raw                   <dbl> 0.0306, 0.4145, 9.8368, 0.0000, 9.6279,...
## $ AZE_raw                     <dbl> NA, 0, 0, NA, NA, NA
## $ MPAEEZ_raw                  <dbl> 0.000164, NA, 1.426495, 0.025115, 0.586...
## $ FORGRO_raw                  <dbl> NA, 0.854187, 0.989503, NA, 1.035620, NA
## $ FORCOV_raw                  <dbl> 0.0, -3.1, -0.2, 0.0, 0.6, 0.0
## $ MTI_raw                     <dbl> 0.025715, NA, -0.000354, NA, 0.018862, NA
## $ EEZTD_raw                   <dbl> 14.276271, NA, 25.486959, 4.709826, 74....
## $ AGWAT_raw                   <dbl> NA, 35.140, 0.141, NA, 2.541, NA
## $ AGSUB_raw                   <dbl> 0, 0, 0, 0, 0, 0
## $ AGPEST_raw                  <dbl> 0, 0, 2, 0, 2, 0
## $ GHGCAP_raw                  <dbl> NA, 3.20000, 16.16991, NA, 6.40000, NA
## $ GHGCAP_raw_imp              <dbl> 0, 1, 0, 0, 1, 0
## $ GHGIND_raw                  <dbl> NA, 0.00000, 16.06677, NA, 72.75947, NA
## $ CO2KWH_raw                  <dbl> NA, NA, 153.4030, NA, 42.5599, NA
## $ CO2KWH_raw_imp              <dbl> 0, 0, 0, 0, 0, 0
## $ DALY_w                      <dbl> NA, 5.388905, 5.388905, NA, 3.367296, 2...
## $ ACSAT_w                     <dbl> NA, 30, 50, NA, 97, 100
## $ WATSUP_w                    <dbl> 100, 42, 51, NA, 97, 100
## $ INDOOR_w                    <dbl> NA, 86.29, 47.74, NA, 50.00, 5.00
## $ PM10_w                      <dbl> NA, 3.720099, 4.187399, NA, 3.781815, 3...
## $ OZONE_w                     <dbl> 0.000000, 12.118929, 19.833180, 0.00000...
## $ SO2_w                       <dbl> 3.3165907, -2.7292534, -0.4180171, -9.1...
## $ NOX_w                       <dbl> 3.26876329, -3.93980539, 1.01552448, -1...
## $ NMVOC_w                     <dbl> 1.8522479, -0.8661753, 1.6919739, -0.32...
## $ WSI_w                       <dbl> NA, 0, 0, NA, 0, NA
## $ WATSTR_w                    <dbl> NA, 2.507972, 1.871802, NA, 0.000000, NA
## $ PACOV_w                     <dbl> 0.0306, 0.4145, 9.8368, 0.0000, 9.6279,...
## $ AZE_w                       <dbl> NA, 0, 0, NA, NA, NA
## $ MPAEEZ_w                    <dbl> 0.000163987, NA, 0.886447829, 0.0248048...
## $ FORGRO_w                    <dbl> NA, 0.854187, 0.989503, NA, 1.035620, NA
## $ FORCOV_w                    <dbl> 0.0, -3.1, -0.2, 0.0, 0.6, 0.0
## $ MTI_w                       <dbl> 0.025715, NA, -0.000354, NA, 0.018862, NA
## $ EEZTD_w                     <dbl> 14.276271, NA, 25.486959, 4.709826, 74....
## $ AGWAT_w                     <dbl> NA, 3.5874003, 0.1319051, NA, 1.2644092...
## $ AGSUB_w                     <dbl> 0, 0, 0, 0, 0, 0
## $ AGPEST_w                    <dbl> 0, 0, 2, 0, 2, 0
## $ GHGCAP_w                    <dbl> NA, 1.423393, 2.843158, NA, 2.008084, NA
## $ GHGIND_w                    <dbl> NA, 0.000000, 2.837133, NA, 4.300809, NA
## $ CO2KWH_w                    <dbl> NA, NA, 5.033068, NA, 3.750912, NA
## $ DALY_tr                     <dbl> NA, 5.541264, 5.662960, NA, 3.367296, 2...
```

```
## $ PM10_tr                    <dbl> NA, 3.720099, 4.187399, NA, 3.781815, 3...
## $ OZONE_tr                   <dbl> 0.000000, 12.118929, 21.033929, 0.00000...
## $ SO2_tr                     <dbl> 3.3165907, -2.7292534, -0.4180171, -9.1...
## $ NOX_tr                     <dbl> 3.26876329, -3.93980539, 1.01552448, -1...
## $ NMVOC_tr                   <dbl> 1.8522479, -0.8661753, 1.6919739, -0.32...
## $ WATSTR_tr                  <dbl> NA, 2.507972, 1.871802, NA, 0.000000, NA
## $ MPAEEZ_tr                  <dbl> 0.000163987, NA, 0.886447829, 0.0248048...
## $ AGWAT_tr                   <dbl> NA, 3.5874003, 0.1319051, NA, 1.2644092...
## $ GHGCAP_tr                  <dbl> NA, 1.423393, 2.843158, NA, 2.008084, NA
## $ GHGIND_tr                  <dbl> NA, 0.000000, 2.837133, NA, 4.300809, NA
## $ CO2KWH_tr                  <dbl> NA, NA, 5.033068, NA, 3.750912, NA
## $ DALY_t                     <dbl> 10, 10, 10, 10, 10, 10
## $ ACSAT_t                    <dbl> 100, 100, 100, 100, 100, 100
## $ WATSUP_t                   <dbl> 100, 100, 100, 100, 100, 100
## $ INDOOR_t                   <dbl> 0, 0, 0, 0, 0, 0
## $ PM10_t                     <dbl> 20, 20, 20, 20, 20, 20
## $ OZONE_t                    <dbl> 0, 0, 0, 0, 0, 0
## $ SO2_t                      <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01
## $ NOX_t                      <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01
## $ NMVOC_t                    <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01
## $ WSI_t                      <dbl> 0, 0, 0, 0, 0, 0
## $ WATSTR_t                   <dbl> 0, 0, 0, 0, 0, 0
## $ PACOV_t                    <dbl> 10, 10, 10, 10, 10, 10
## $ AZE_t                      <dbl> 100, 100, 100, 100, 100, 100
## $ MPAEEZ_t                   <dbl> 10, 10, 10, 10, 10, 10
## $ FORGRO_t                   <dbl> 1, 1, 1, 1, 1, 1
## $ FORCOV_t                   <dbl> 0, 0, 0, 0, 0, 0
## $ MTI_t                      <dbl> 0, 0, 0, 0, 0, 0
## $ EEZTD_t                    <dbl> 0, 0, 0, 0, 0, 0
## $ AGWAT_t                    <dbl> 10, 10, 10, 10, 10, 10
## $ AGSUB_t                    <dbl> 0, 0, 0, 0, 0, 0
## $ AGPEST_t                   <dbl> 22, 22, 22, 22, 22, 22
## $ GHGCAP_t                   <dbl> 2.5, 2.5, 2.5, 2.5, 2.5, 2.5
## $ GHGIND_t                   <dbl> 36.3, 36.3, 36.3, 36.3, 36.3, 36.3
## $ CO2KWH_t                   <dbl> 10, 10, 10, 10, 10, 10
## $ DALY_ttr                   <dbl> 2.302585, 2.302585, 2.302585, 2.302585,...
## $ PM10_ttr                   <dbl> 2.995732, 2.995732, 2.995732, 2.995732,...
## $ OZONE_ttr                  <dbl> 0, 0, 0, 0, 0, 0
## $ SO2_ttr                    <dbl> -4.60517, -4.60517, -4.60517, -4.60517,...
## $ NOX_ttr                    <dbl> -4.60517, -4.60517, -4.60517, -4.60517,...
## $ NMVOC_ttr                  <dbl> -4.60517, -4.60517, -4.60517, -4.60517,...
## $ WATSTR_ttr                 <dbl> 0, 0, 0, 0, 0, 0
## $ MPAEEZ_ttr                 <dbl> 2.397895, 2.397895, 2.397895, 2.397895,...
## $ AGWAT_ttr                  <dbl> 2.397895, 2.397895, 2.397895, 2.397895,...
## $ GHGCAP_ttr                 <dbl> 1.252763, 1.252763, 1.252763, 1.252763,...
## $ GHGIND_ttr                 <dbl> 3.618993, 3.618993, 3.618993, 3.618993,...
## $ CO2KWH_ttr                 <dbl> 2.302585, 2.302585, 2.302585, 2.302585,...
## $ double_EPI                 <dbl> NA, NA, 72.6, NA, 142.8, NA
```

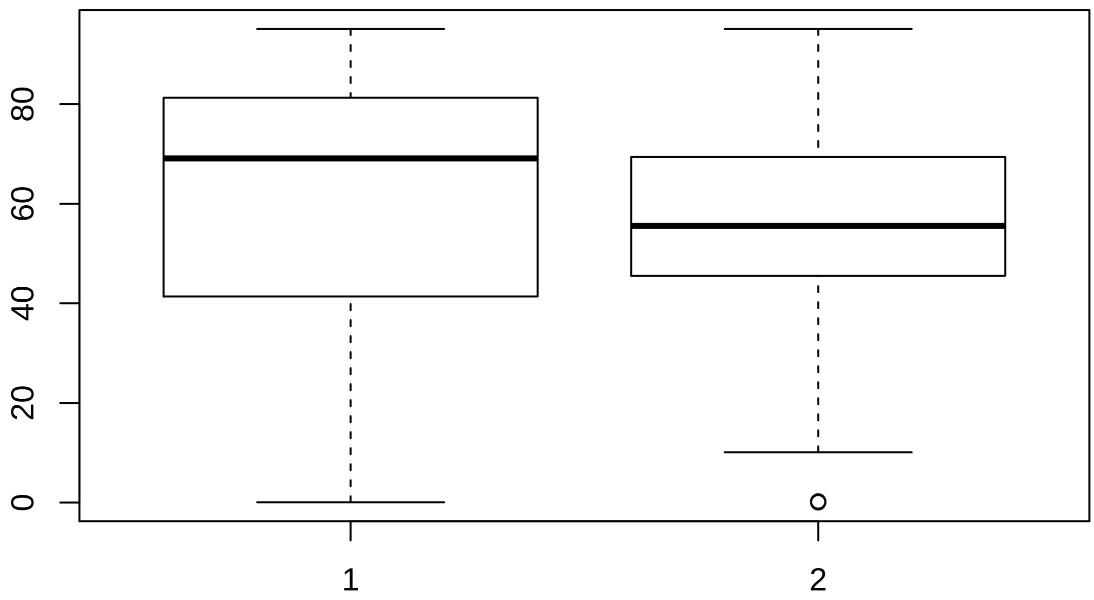
```r
# If you only want to see the new column instead of calling the mutate, you can
# use the transmute() fuction.
# The difference between the mutate() and transmute() is that mutate() function returns
# the entire dataframe along with the new column and the transmute() shows only the new column.
data %>% transmute(double_DALY = DALY * 2) %>% glimpse()
```

```
## Observations: 231
## Variables: 1
## $ double_DALY <dbl> NA, 0.00, 0.00, NA, 131.00, 169.54, NA, 178.20, 143.26,...
```

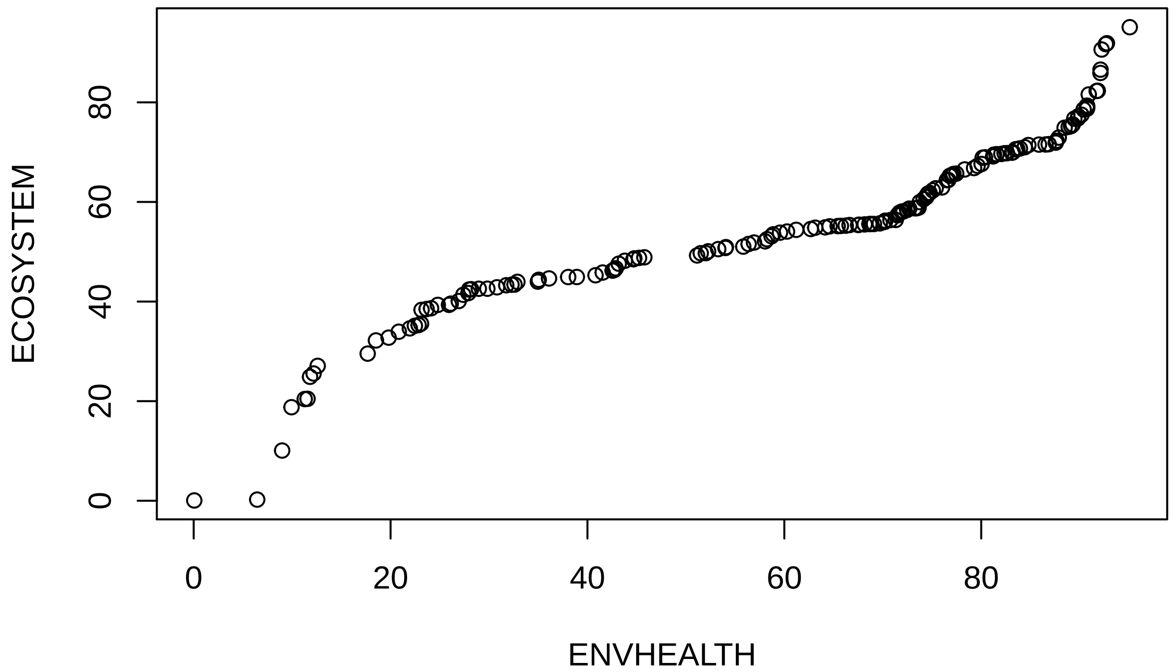
```r
# (6) summaries() and mean ==> summarize the data frame into a single row using another aggrigate funct
data %>% summarise(mean_EPI = mean(EPI, na.rm = TRUE), mean_DALY = mean(DALY, na.rm = TRUE)) %>% glimps
```

```
## Observations: 1
## Variables: 2
## $ mean_EPI  <dbl> 58.37055
## $ mean_DALY <dbl> 53.94313
```

```r
# (7) draw boxplot and qqplot
boxplot(ENVHEALTH,ECOSYSTEM)
```




```r
qqplot(ENVHEALTH,ECOSYSTEM)
```



Lab2b Regression

Using the EPI (under /EPI on web) dataset find the single most important factor in increasing the EPI in a given region ### Linear and Least-Squares

```
# (1) create a multilinear regression model
lmENVH <- lm(ENVHEALTH~DALY+AIR_H+WATER_H)
```

```
# (2) display the mode
lmENVH
```

```
##
## Call:
## lm(formula = ENVHEALTH ~ DALY + AIR_H + WATER_H)
##
## Coefficients:
## (Intercept)          DALY        AIR_H      WATER_H
##  -2.673e-05    5.000e-01    2.500e-01    2.500e-01
```

1) sinece DALY has the largest coefficient, which could mean that DALY has the largest effect on increaseing EPI in a given region
2) all three factors are significant based on their p-values

```
summary( lmENVH )
```

```
##
## Call:
## lm(formula = ENVHEALTH ~ DALY + AIR_H + WATER_H)
##
```

```
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0072734 -0.0027299  0.0001145  0.0021423  0.0055205
##
## Coefficients:
##               Estimate Std. Error   t value Pr(>|t|)
## (Intercept) -2.673e-05  6.377e-04    -0.042    0.967
## DALY         5.000e-01  1.922e-05 26020.669   <2e-16 ***
## AIR_H        2.500e-01  1.273e-05 19645.297   <2e-16 ***
## WATER_H      2.500e-01  1.751e-05 14279.903   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003097 on 178 degrees of freedom
##   (49 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.983e+09 on 3 and 178 DF,  p-value: < 2.2e-16
```

```
cENVH<-coef(lmENVH)
cENVH
```

```
##   (Intercept)          DALY         AIR_H       WATER_H
## -2.673362e-05  5.000401e-01  2.499968e-01  2.499781e-01
```

Predict

```
# keep copies
origin_DALY <- DALY
origin_AIR_H <- AIR_H
origin_WATER_H <- WATER_H
```

```
DALY <- c( seq(5, 95, 5) )
AIR_H <- c( seq(5, 95, 5) )
WATER_H <- c( seq(5, 95, 5) )
NEW <- data.frame( DALY, AIR_H, WATER_H )
```

```
 pENV<- predict(lmENVH,NEW,se.fit = TRUE,interval="prediction",na.action = na.pass)
```

```
cENV<-predict(lmENVH,NEW,se.fit = TRUE,interval="confidence",na.action = na.pass)
```

reference: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/predict.lm

Repeat for AIR_E

```
DALY <-origin_DALY
AIR_H <- origin_AIR_H
WATER_H <- origin_WATER_H
# (1) create a multilinear regression model
lmAIR_E <- lm(AIR_E~DALY+AIR_H+WATER_H)
```

```r
# (2) display the mode
lmAIR_E
```

```
##
## Call:
## lm(formula = AIR_E ~ DALY + AIR_H + WATER_H)
##
## Coefficients:
## (Intercept)         DALY        AIR_H      WATER_H
##     59.2903      -0.1248       0.1686      -0.1798
```

```r
summary( lmAIR_E )
```

```
##
## Call:
## lm(formula = AIR_E ~ DALY + AIR_H + WATER_H)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.708  -7.328  -1.739   8.117  38.182
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 59.29025    2.55759  23.182  < 2e-16 ***
## DALY        -0.12482    0.07707  -1.620  0.10710
## AIR_H        0.16863    0.05104   3.304  0.00115 **
## WATER_H     -0.17982    0.07021  -2.561  0.01126 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.42 on 178 degrees of freedom
##   (49 observations deleted due to missingness)
## Multiple R-squared:  0.1803, Adjusted R-squared:  0.1664
## F-statistic: 13.05 on 3 and 178 DF,  p-value: 9.654e-08
```

```r
cAIR_E<-coef(lmAIR_E)
cAIR_E
```

```
## (Intercept)        DALY       AIR_H     WATER_H
##  59.2902524  -0.1248238   0.1686255  -0.1798231
```

Predict

```r
DALY <- c( seq(5, 95, 5) )
AIR_H <- c( seq(5, 95, 5) )
WATER_H <- c( seq(5, 95, 5) )
NEW <- data.frame( DALY, AIR_H, WATER_H )
```

```r
pAIR_E<- predict(lmAIR_E,interval="prediction")
```

```
## Warning in predict.lm(lmAIR_E, interval = "prediction"): predictions on current data refer to _future
```

```r
cAIR_E-predict(lmAIR_E,interval="prediction")
```

```
## Warning in predict.lm(lmAIR_E, interval = "prediction"): predictions on current data refer to _future

## Warning in cAIR_E - predict(lmAIR_E, interval = "prediction"): longer object
## length is not a multiple of shorter object length

##              fit       lwr        upr
## 2    -4.05681417 -38.26518 -28.970071
## 3   -61.40447961 -36.50292 -86.361042
## 5   -42.66329819  41.25097 -67.455941
## 6   -46.29858949 -21.56547 -70.976711
## 8    20.60174588 -13.49201  -4.426121
## 9   -44.67895399 -20.07082 -69.342091
## 10  -46.15677710  37.62988 -70.821804
## 12  -50.69237764 -25.95216 -75.377599
## 13   12.14455694 -22.28006 -12.552451
## 14  -44.78040900 -20.14135 -69.474468
## 15  -50.48275383  33.22437 -75.068246
## 16  -58.20044248 -33.16690 -83.178987
## 17   12.18665854 -22.26844 -12.479869
## 18  -56.66839334 -31.99303 -81.398751
## 19  -55.67283222  28.34874 -80.572780
## 20  -46.54171452 -21.40266 -71.625773
## 21   15.23566854 -19.14555  -9.504736
## 22  -41.85089702 -17.13668 -66.620112
## 23  -47.41809150  36.38902 -72.103572
## 24  -46.02671891 -21.32791 -70.670529
## 25    7.43974701 -26.77302 -17.469110
## 26  -56.60144068 -31.83660 -81.421279
## 28  -50.87113898  32.87776 -75.498412
## 29  -53.10916470 -28.38727 -77.776064
## 30   13.61961802 -20.85616 -11.026226
## 31  -42.59165056 -17.90089 -67.337413
## 32  -58.10030273  25.80564 -82.884620
## 33  -51.39104717 -26.54766 -76.179437
## 34    2.57467139 -31.80245 -22.169835
## 35  -47.00963755 -22.35216 -71.722117
## 36  -45.35569390  38.48330 -70.073063
## 37  -45.52850132 -20.81573 -70.186277
## 38   13.61035997 -20.75039 -11.150522
## 39  -58.12527845 -33.34041 -82.965143
## 40  -53.21595467  30.71590 -78.026180
## 41  -61.23156571 -36.32450 -86.083630
## 42    5.03289582 -29.30721 -19.748626
## 43  -50.16207353 -25.52478 -74.854369
## 44  -51.73465678  32.03718 -76.384866
## 45  -53.13472209 -28.46320 -77.751247
## 46    5.57448641 -28.76135 -19.211308
## 47  -45.55653986 -20.97249 -70.195589
## 48  -49.86208013  33.94174 -74.544272
## 50  -43.60275621 -18.83307 -68.317442
## 51   11.15156794 -23.30092 -13.517573
```

```
## 52  -47.51517380 -22.88396 -72.201383
## 53  -54.01848742  30.03105 -78.946403
## 54  -48.25208174 -23.57012 -72.879046
## 55   11.66897838 -22.77333 -13.010337
## 56  -53.63044191 -28.93441 -78.381470
## 57  -47.20489935  36.52383 -71.812002
## 58  -51.56954760 -26.85111 -76.232984
## 59   13.41094752 -21.04327 -11.256464
## 60  -59.15129102 -34.16482 -84.192765
## 62  -44.94344698  38.85258 -69.617844
## 63  -49.98459934 -25.16730 -74.746901
## 64   -1.36023897 -35.43264 -26.409466
## 65  -47.51517380 -22.88396 -72.201383
## 66  -57.34211762  27.10094 -82.663550
## 68  -47.57017310 -22.82897 -72.256382
## 71   -0.06283985 -34.35856 -24.888746
## 72  -47.74609777 -23.12178 -72.425413
## 73  -42.70108234  41.17548 -67.456016
## 74  -56.76054841 -32.00781 -81.458289
## 76    4.35635145 -29.97395 -20.434975
## 78  -47.22518200 -22.17167 -72.333697
## 79  -56.75896173  27.22888 -81.625175
## 80  -71.08468288 -45.27458 -96.839782
## 81   14.30713634 -20.15033 -10.357023
## 82  -47.35128092 -22.76357 -71.993989
## 84  -42.94751986  41.14813 -67.921542
## 87  -52.50808802 -27.72856 -77.232612
## 89   10.46138090 -24.03648 -14.162386
## 90  -46.28803990 -21.70736 -70.923721
## 91  -58.00791618  26.02045 -82.914652
## 92  -49.58857342 -24.79916 -74.322987
## 93   10.33575404 -24.07179 -14.378333
## 95  -51.82421281 -27.23024 -76.473189
## 96  -46.97706999  36.84156 -71.674077
## 97  -48.77156828 -24.11858 -73.369554
## 98    5.93284365 -28.19714 -19.058804
## 99  -46.43045527 -21.72407 -71.191840
## 100 -44.12193183  39.74776 -68.869995
## 101 -45.77971086 -21.03161 -70.472811
## 102  14.52422651 -19.91117 -10.162003
## 103 -47.32570338 -22.77466 -71.931748
## 104 -44.96754233  38.84605 -69.659503
## 105 -52.75814357 -27.63566 -77.825630
## 106   0.48315846 -33.91423 -24.241075
## 107 -50.87095858 -26.18287 -75.614045
## 108 -55.48346906  28.37605 -80.221364
## 110 -50.14782317 -25.38046 -74.860184
## 111  10.95397549 -23.51871 -13.694967
## 112 -39.69835073 -14.82423 -64.627473
## 113 -52.82383968  31.04512 -77.571177
## 114 -47.41372954 -22.66674 -72.105719
## 115  -5.03293030 -39.03644 -30.151048
## 116 -47.30687508 -22.65870 -72.010052
## 117 -48.15093768  35.59158 -72.771832
```

```
## 119 -44.55311611 -19.46422 -69.587010
## 120   6.48594446 -27.93836 -18.211382
## 121 -54.04285139 -29.36791 -78.772796
## 122 -47.22172458  36.58611 -71.907933
## 123 -52.39773324 -27.63656 -77.103911
## 125   4.52513621 -29.80112 -20.270235
## 126 -46.56615259 -21.91805 -71.269251
## 127 -50.32055636  33.39154 -74.911023
## 128 -61.85665462 -36.69909 -86.959224
## 129   4.21045459 -30.20203 -20.498683
## 130 -47.73581430 -23.16821 -72.358416
## 132 -47.03166050  36.70947 -71.651163
## 133 -53.07069400 -27.84040 -78.245992
## 134  12.84892356 -21.56961 -11.854174
## 135 -45.34112321 -20.37291 -70.364337
## 136 -46.20752303  38.00559 -71.299004
## 138 -61.87701624 -36.87302 -86.826018
## 139   2.33845238 -31.98794 -22.456784
## 142 -49.68147464 -25.05007 -74.367882
## 143 -53.94755254  30.29611 -79.069591
## 144 -49.60162633 -24.87164 -74.276609
## 146   8.96966962 -25.50190 -15.680390
## 148 -59.45217376 -34.48594 -84.473402
## 150 -61.54543301  22.41424 -86.383478
## 151 -52.82884075 -27.98016 -77.622521
## 152  10.78261512 -23.57099 -13.985409
## 153 -44.68935125 -20.04952 -69.384183
## 154 -47.22172458  36.58611 -71.907933
## 155 -53.25423150 -28.57733 -77.876129
## 157  11.89990235 -22.53552 -12.786306
## 158 -44.76586766 -19.98635 -69.600384
## 159 -44.48612008  39.62256 -69.473175
## 160 -48.01704446 -23.32550 -72.653594
## 162  11.51924144 -22.95569 -13.127458
## 163 -49.95644191 -25.40363 -74.564252
## 165 -59.88541127  24.26535 -84.914543
## 166 -50.08453761 -25.39078 -74.723300
## 168  14.37156963 -19.94518 -10.433309
## 169 -47.53651700 -22.93328 -72.194753
## 170 -45.95164054  38.03516 -70.816814
## 173 -42.60312027 -17.78375 -67.367489
## 175   6.50415941 -27.93802 -18.175286
## 176 -53.86420218 -28.91630 -78.867102
## 177 -60.87220917  23.09108 -85.713876
## 178 -44.66423658 -19.90577 -69.367704
## 179   8.77554070 -25.63904 -15.931506
## 180 -49.71712443 -24.79186 -74.697391
## 181 -54.49991811  29.34461 -79.222814
## 182 -43.60812799 -18.81917 -68.342087
## 185   1.93562266 -32.06789 -23.182492
## 186 -62.08045345 -37.24863 -86.967278
## 187 -49.95269637  33.75411 -74.537872
## 189 -64.51699844 -39.50672 -89.472280
## 191   1.20106138 -33.22802 -23.491482
```

```
## 192 -51.17336642 -26.60494 -75.796797
## 193 -48.78202167  35.04161 -73.484031
## 194 -46.11174487 -21.41454 -70.753952
## 195  12.40543861 -22.00371 -12.307041
## 196 -58.20633520 -33.56847 -82.899204
## 197 -50.65601048  33.13508 -75.325479
## 198 -45.44583966 -20.72851 -70.108169
## 200   0.76561753 -33.39290 -24.197489
## 201 -59.93779123 -35.12796 -84.802619
## 202 -44.06228387  39.92233 -68.925270
## 203 -53.47607314 -28.76674 -78.130407
## 205   2.85995751 -31.60346 -21.798250
## 207 -43.26283590 -18.53659 -68.044084
## 208 -43.63467543  40.20061 -68.348333
## 209 -50.07066697 -25.39535 -74.690986
## 210  11.64485665 -22.86442 -12.967490
## 213 -60.27327261 -35.55834 -85.043207
## 214 -60.01577708  23.87177 -84.781700
## 215 -52.33745029 -27.41582 -77.204083
## 216  18.77087813 -15.43749  -6.142380
## 217 -47.83208591 -23.21946 -72.499708
## 218 -47.88616326  35.93836 -72.589064
## 221 -52.40476556 -27.64593 -77.108601
## 224  13.99467392 -20.38253 -10.749745
## 225 -54.74781835 -29.67228 -79.878354
## 228 -55.80074369  27.97105 -80.450907
## 229 -57.25062757 -32.31323 -82.133030
## 230   1.87881048 -32.46296 -22.901044
## 231 -56.32190028 -31.61676 -81.082041
```

Repeat for CLIMATE

```
DALY <-origin_DALY
AIR_H <- origin_AIR_H
WATER_H <- origin_WATER_H
# (1) create a multilinear regression model
lmCLIMATE <- lm(CLIMATE~DALY+AIR_H+WATER_H)
```

```
# (2) display the mode
lmCLIMATE
```

```
##
## Call:
## lm(formula = CLIMATE ~ DALY + AIR_H + WATER_H)
##
## Coefficients:
## (Intercept)         DALY        AIR_H      WATER_H
##     75.3487      -0.1732       0.0181      -0.1538
```

```
summary( lmCLIMATE )
```

```
##
## Call:
## lm(formula = CLIMATE ~ DALY + AIR_H + WATER_H)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.578  -9.768   1.165   9.164  44.434
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 75.34874    3.01412  24.999   <2e-16 ***
## DALY        -0.17323    0.09050  -1.914   0.0573 .
## AIR_H        0.01810    0.05919   0.306   0.7602
## WATER_H     -0.15385    0.08161  -1.885   0.0611 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.15 on 168 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.255,  Adjusted R-squared:  0.2417
## F-statistic: 19.17 on 3 and 168 DF,  p-value: 9.704e-11
```

```
cCLIMATE<-coef(lmCLIMATE)
cCLIMATE
```

```
## (Intercept)        DALY       AIR_H     WATER_H
##  75.3487356  -0.1732265   0.0180960  -0.1538496
```

```
DALY <- c( seq(5, 95, 5) )
AIR_H <- c( seq(5, 95, 5) )
WATER_H <- c( seq(5, 95, 5) )
NEW <- data.frame( DALY, AIR_H, WATER_H )
```

```
pCLIMATE<- predict(lmCLIMATE,NEW,interval="prediction")
```

```
cCLIMATE<-predict(lmCLIMATE,NEW,interval="confidence")
```

Lab2- Part2: 2a, 2b

MultiLinear Regression

```
df = read_csv( "dataset_multipleRegression.csv" )
```

```
## Parsed with column specification:
## cols(
##   YEAR = col_double(),
##   ROLL = col_double(),
##   UNEM = col_double(),
##   HGRAD = col_double(),
##   INC = col_double()
## )
```

```r
# attach data frame
attach(df)
```

```r
# create a linear model using lm(FORMULA, DATAVAR)
# predict the fall enrollment (ROLL) using the unemployment rate (UNEM) and number of spring high schoo
twoPredictorModel <- lm( ROLL ~ UNEM + HGRAD, df )
# display model
twoPredictorModel
```

```
##
## Call:
## lm(formula = ROLL ~ UNEM + HGRAD, data = df)
##
## Coefficients:
## (Intercept)         UNEM        HGRAD
##  -8255.7511     698.2681       0.9423
```

```r
# the expected fall enrollment (ROLL) given this year's unemployment rate
# (UNEM) of 7% and spring high school graduating class (HGRAD) of 90,000 is:
ans1 <- -8255.7511 + 698.2681 * 7 + 0.9423 * 90000
```

```r
# Repeat and add per capita income (INC) to the model. Predict ROLL if INC=$25,000
# Summarize and compare the two models.
# Comment on significance
threePredictorModel <- lm( ROLL ~ UNEM + HGRAD + INC, df )
# display model
threePredictorModel
```

```
##
## Call:
## lm(formula = ROLL ~ UNEM + HGRAD + INC, data = df)
##
## Coefficients:
## (Intercept)         UNEM        HGRAD          INC
##  -9153.2545     450.1245       0.4065       4.2749
```

```r
# the expected fall enrollment (ROLL) given this year's unemployment rate (UNEM) of 9%, spring high sch
ans2 <- -9153.2545 + 450.1245 * 9 + 0.4065 * 100000 + 4.2749 * 30000
ans2
```

```
## [1] 163794.9
```

```r
# generate model summaries
summary(twoPredictorModel)
```

```
##
## Call:
## lm(formula = ROLL ~ UNEM + HGRAD, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2102.2  -861.6  -349.4   374.5  3603.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.256e+03  2.052e+03  -4.023  0.00044 ***
## UNEM         6.983e+02  2.244e+02   3.111  0.00449 **
## HGRAD        9.423e-01  8.613e-02  10.941 3.16e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1313 on 26 degrees of freedom
## Multiple R-squared:  0.8489, Adjusted R-squared:  0.8373
## F-statistic: 73.03 on 2 and 26 DF,  p-value: 2.144e-11
```

```
summary(threePredictorModel)
```

```
##
## Call:
## lm(formula = ROLL ~ UNEM + HGRAD + INC, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1148.84  -489.71    -1.88   387.40  1425.75
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.153e+03  1.053e+03  -8.691 5.02e-09 ***
## UNEM         4.501e+02  1.182e+02   3.809 0.000807 ***
## HGRAD        4.065e-01  7.602e-02   5.347 1.52e-05 ***
## INC          4.275e+00  4.947e-01   8.642 5.59e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 670.4 on 25 degrees of freedom
## Multiple R-squared:  0.9621, Adjusted R-squared:  0.9576
## F-statistic: 211.5 on 3 and 25 DF,  p-value: < 2.2e-16
```

kNN

```
abalone <- read.csv( "abalone.csv" )
abalone <- read.csv("abalone.csv", header =T, na.strings=c("","NA"))
suppressWarnings(suppressMessages(library(dplyr)))
#create an sex column that is numeric
abalone <- abalone %>%
  mutate(sex_num =case_when(
    Sex %in% 'M' ~ 0,
    Sex %in% "F"  ~ 1,
    Sex %in% "I" ~ 2
  ))
#create an age column
abalone <- abalone %>%
```

```r
  mutate(age=case_when(
    Rings %in% 1:5 ~ "young",
    Rings %in% 6:13 ~ "adult",
    Rings %in% 14:30 ~ "old"
  ))
# remove rings, sex
abalone <- abalone[c(-1, -9)]
str(abalone)
```

```
## 'data.frame':    4177 obs. of  9 variables:
##  $ Length        : num  0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...
##  $ Diameter      : num  0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...
##  $ Height        : num  0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...
##  $ Whole.weight  : num  0.514 0.226 0.677 0.516 0.205 ...
##  $ Shucked.weight: num  0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ Viscera.weight: num  0.101 0.0485 0.1415 0.114 0.0395 ...
##  $ Shell.weight  : num  0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...
##  $ sex_num       : num  0 0 1 0 2 2 1 1 0 1 ...
##  $ age           : chr  "old" "adult" "adult" "adult" ...
```

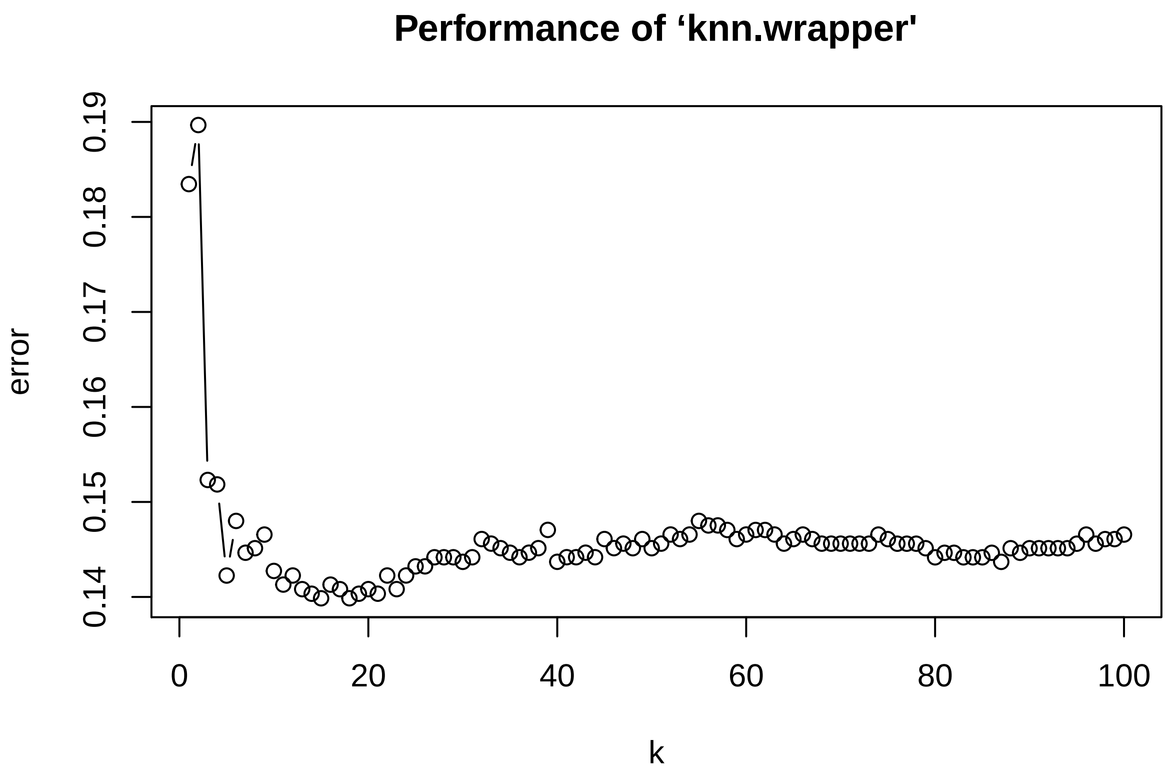
```r
### the dependent variable is age , with the different values young adult old
### standardize the predictors
set.seed(100)
abalone_scale <- data.frame(scale(abalone[1:8]))
### add the target variable to the data set abalone_scale
abalone$age <- as.factor(abalone$age)
abalone_scale  <- cbind(abalone_scale, age = abalone$age)
i <- sample(4177, 2088)
abalone_train <- abalone_scale[i,]
abalone_test <- abalone_scale[-i,]
```

The value of K is important in the KNN algorithm, because the prediction accuracy in the test set depends on it. The optimal value of K is the value that leads to the highest prediction accuracy.

```r
### we use the tune.knn function in the e1071 package to determine a good K number
### this function performs a 10-fold cross-validation
library(e1071)
t_knn <- tune.knn(abalone_train[,-9], factor(abalone_train[,9]), k = 1:100)
t_knn # names(t_knn) to see the list of variables
```

```
##
## Parameter tuning of 'knn.wrapper':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##   k
##  15
##
## - best performance: 0.1398601
```

```
plot(t_knn)
```




```
# Run the prediction
library(class)
age <- abalone_train$age
pred <- knn(train = abalone_train[,-9], test = abalone_test[,-9], cl = age, k = t_knn$best.parameters)
```

```
### get the prediction accuracy in the test set
mean(pred == abalone_test$age)
```

```
## [1] 0.8697942
```

```
table(pred,abalone_test$age)
```

```
##
## pred    adult  old young
##   adult  1727  218    34
##   old       9   30     0
##   young    11    0    60
```

Kmeans (Clustering)

```
data("iris")
iris_dataset<-iris
view(iris_dataset)
```

Splitting the data into training and testing Sets

```r
# Load the Caret package which allows us to partition the data
library(caret)
# We use the dataset to create a partition (80% training 20% testing)
index <- createDataPartition(iris_dataset$Species, p=0.80, list=FALSE)
# select 20% of the data for testing
testset <- iris_dataset[-index,]
# select 80% of data to train the models
trainset <- iris_dataset[index,]
```

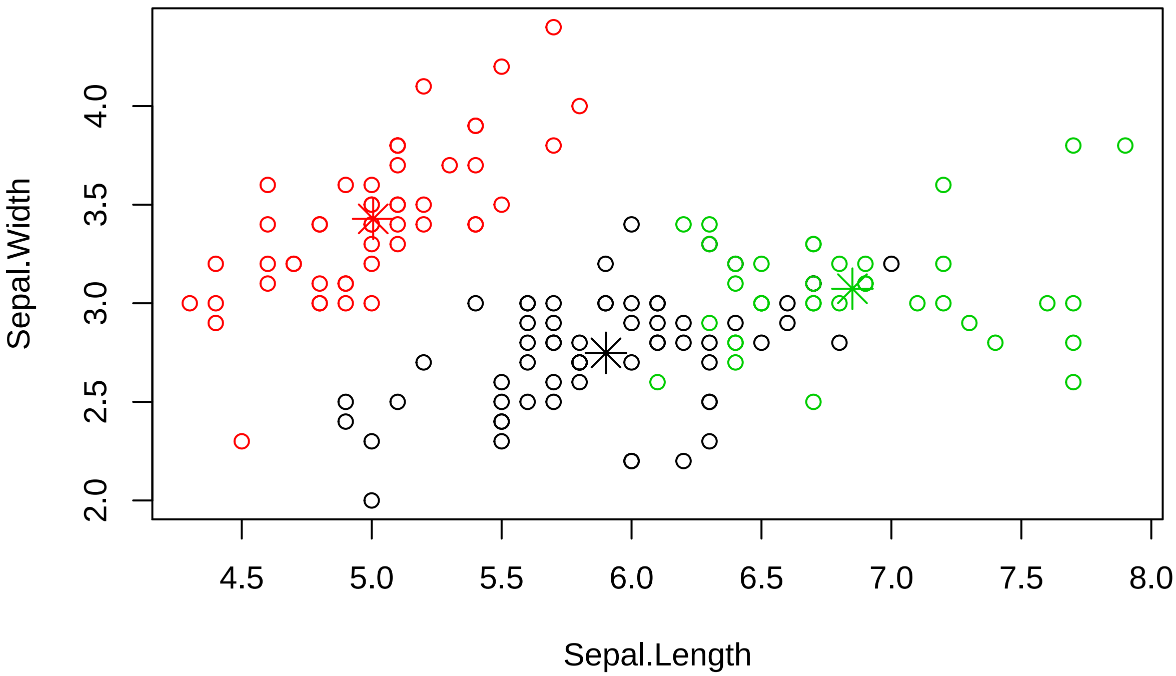
```r
# Since Kmeans is a random start algo, we need to set the seed to ensure reproduceability
set.seed(1000)
irisCluster <- kmeans(iris[, 1:4], centers = 3, nstart = 1000)
irisCluster
```

```
## K-means clustering with 3 clusters of sizes 62, 50, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.901613    2.748387     4.393548    1.433871
## 2     5.006000    3.428000     1.462000    0.246000
## 3     6.850000    3.073684     5.742105    2.071053
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 3 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 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 3 3 3 1 3 3 3 3
## [112] 3 3 1 1 3 3 3 3 1 3 1 3 1 3 3 1 1 3 3 3 3 3 1 3 3 3 3 1 3 3 3 1 3 3 3 1 3
## [149] 3 1
##
## Within cluster sum of squares by cluster:
## [1] 39.82097 15.15100 23.87947
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
table(irisCluster$cluster, iris$Species)
```

```
##
##     setosa versicolor virginica
##   1      0         48        14
##   2     50          0         0
##   3      0          2        36
```

```r
plot(iris[c("Sepal.Length", "Sepal.Width")], col=irisCluster$cluster)
points(irisCluster$centers[,c("Sepal.Length", "Sepal.Width")], col=1:3, pch=8, cex=2)
```


Sepal.Width
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Sepal.Length