Practicum 1

Problem 1

1 / Predicting Life Expectancy

Data Exploration

```
## Loading required package: ggplot2
```

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

```
## Rows: 2938 Columns: 22
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Country, Status
## dbl (20): Year, Life expectancy, Adult Mortality, infant deaths, Alcohol, pe...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## tibble [2,938 x 19] (S3: tbl_df/tbl/data.frame)
##  $ Life expectancy               : num [1:2938] 65 59.9 59.9 59.5 59.2 58.8 58.6 58.1 57.5 57.3 ...
##  $ Adult Mortality               : num [1:2938] 263 271 268 272 275 279 281 287 295 295 ...
##  $ infant deaths                 : num [1:2938] 62 64 66 69 71 74 77 80 82 84 ...
##  $ Alcohol                       : num [1:2938] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.03 0.02 0.03 .
##  $ percentage expenditure        : num [1:2938] 71.3 73.5 73.2 78.2 7.1 ...
##  $ Hepatitis B                   : num [1:2938] 65 62 64 67 68 66 63 64 63 64 ...
##  $ Measles                       : num [1:2938] 1154 492 430 2787 3013 ...
##  $ BMI                           : num [1:2938] 19.1 18.6 18.1 17.6 17.2 16.7 16.2 15.7 15.2 14.7 .
##  $ under-five deaths             : num [1:2938] 83 86 89 93 97 102 106 110 113 116 ...
##  $ Polio                         : num [1:2938] 6 58 62 67 68 66 63 64 63 58 ...
##  $ Total expenditure             : num [1:2938] 8.16 8.18 8.13 8.52 7.87 9.2 9.42 8.33 6.73 7.43 ..
##  $ Diphtheria                    : num [1:2938] 65 62 64 67 68 66 63 64 63 58 ...
##  $ HIV/AIDS                      : num [1:2938] 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ GDP                           : num [1:2938] 584.3 612.7 631.7 670 63.5 ...
##  $ Population                    : num [1:2938] 33736494 327582 31731688 3696958 2978599 ...
##  $ thinness  1-19 years          : num [1:2938] 17.2 17.5 17.7 17.9 18.2 18.4 18.6 18.8 19 19.2 ...
##  $ thinness 5-9 years            : num [1:2938] 17.3 17.5 17.7 18 18.2 18.4 18.7 18.9 19.1 19.3 ...
##  $ Income composition of resources: num [1:2938] 0.479 0.476 0.47 0.463 0.454 0.448 0.434 0.433 0.415
##  $ Schooling                     : num [1:2938] 10.1 10 9.9 9.8 9.5 9.2 8.9 8.7 8.4 8.1 ...
```

```
##  Life expectancy Adult Mortality infant deaths       Alcohol
##  Min.   :36.30   Min.   :  1.0   Min.   :   0.0   Min.   : 0.0100
##  1st Qu.:63.10   1st Qu.: 74.0   1st Qu.:   0.0   1st Qu.: 0.8775
##  Median :72.10   Median :144.0   Median :   3.0   Median : 3.7550
##  Mean   :69.22   Mean   :164.8   Mean   :  30.3   Mean   : 4.6029
##  3rd Qu.:75.70   3rd Qu.:228.0   3rd Qu.:  22.0   3rd Qu.: 7.7025
##  Max.   :89.00   Max.   :723.0   Max.   :1800.0   Max.   :17.8700
##  NA's   :10      NA's   :10                       NA's   :194
##  percentage expenditure  Hepatitis B       Measles              BMI
```

```
##  Min.   :    0.000      Min.   : 1.00   Min.   :     0.0   Min.   : 1.00
##  1st Qu.:    4.685      1st Qu.:77.00   1st Qu.:     0.0   1st Qu.:19.30
##  Median :   64.913      Median :92.00   Median :    17.0   Median :43.50
##  Mean   :  738.251      Mean   :80.94   Mean   :  2419.6   Mean   :38.32
##  3rd Qu.:  441.534      3rd Qu.:97.00   3rd Qu.:   360.2   3rd Qu.:56.20
##  Max.   :19479.912      Max.   :99.00   Max.   :212183.0   Max.   :87.30
##                         NA's   :553                        NA's   :34
##  under-five deaths     Polio       Total expenditure   Diphtheria
##  Min.   :   0.00   Min.   : 3.00   Min.   : 0.370    Min.   : 2.00
##  1st Qu.:   0.00   1st Qu.:78.00   1st Qu.: 4.260    1st Qu.:78.00
##  Median :   4.00   Median :93.00   Median : 5.755    Median :93.00
##  Mean   :  42.04   Mean   :82.55   Mean   : 5.938    Mean   :82.32
##  3rd Qu.:  28.00   3rd Qu.:97.00   3rd Qu.: 7.492    3rd Qu.:97.00
##  Max.   :2500.00   Max.   :99.00   Max.   :17.600    Max.   :99.00
##                    NA's   :19      NA's   :226       NA's   :19
##     HIV/AIDS           GDP              Population        thinness  1-19 years
##  Min.   : 0.100   Min.   :      1.68   Min.   :3.400e+01   Min.   : 0.10
##  1st Qu.: 0.100   1st Qu.:   463.94   1st Qu.:1.958e+05   1st Qu.: 1.60
##  Median : 0.100   Median :  1766.95   Median :1.387e+06   Median : 3.30
##  Mean   : 1.742   Mean   :  7483.16   Mean   :1.275e+07   Mean   : 4.84
##  3rd Qu.: 0.800   3rd Qu.:  5910.81   3rd Qu.:7.420e+06   3rd Qu.: 7.20
##  Max.   :50.600   Max.   :119172.74   Max.   :1.294e+09   Max.   :27.70
##                   NA's   :448         NA's   :652         NA's   :34
##  thinness 5-9 years Income composition of resources   Schooling
##  Min.   : 0.10      Min.   :0.0000                  Min.   : 0.00
##  1st Qu.: 1.50      1st Qu.:0.4930                  1st Qu.:10.10
##  Median : 3.30      Median :0.6770                  Median :12.30
##  Mean   : 4.87      Mean   :0.6276                  Mean   :11.99
##  3rd Qu.: 7.20      3rd Qu.:0.7790                  3rd Qu.:14.30
##  Max.   :28.60      Max.   :0.9480                  Max.   :20.70
##  NA's   :34         NA's   :167                     NA's   :163
```

```
## # A tibble: 6 x 19
##   `Life expectancy` `Adult Mortality` `infant deaths` Alcohol
##               <dbl>             <dbl>           <dbl>   <dbl>
## 1              65                 263              62    0.01
## 2              59.9               271              64    0.01
## 3              59.9               268              66    0.01
## 4              59.5               272              69    0.01
## 5              59.2               275              71    0.01
## 6              58.8               279              74    0.01
## # i 15 more variables: `percentage expenditure` <dbl>, `Hepatitis B` <dbl>,
## #   Measles <dbl>, BMI <dbl>, `under-five deaths` <dbl>, Polio <dbl>,
## #   `Total expenditure` <dbl>, Diphtheria <dbl>, `HIV/AIDS` <dbl>, GDP <dbl>,
## #   Population <dbl>, `thinness  1-19 years` <dbl>, `thinness 5-9 years` <dbl>,
## #   `Income composition of resources` <dbl>, Schooling <dbl>
```

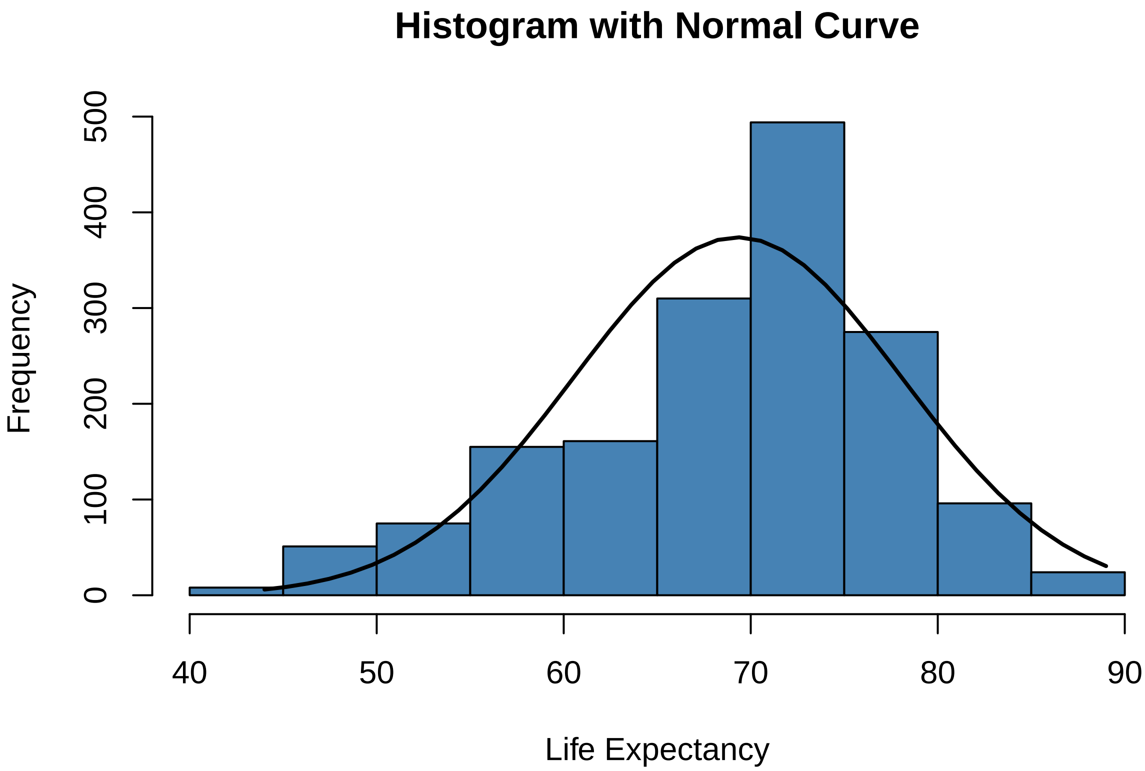
```
## # A tibble: 6 x 19
##   `Life expectancy` `Adult Mortality` `infant deaths` Alcohol
##               <dbl>             <dbl>           <dbl>   <dbl>
## 1              44.6               717              28    4.14
## 2              44.3               723              27    4.36
## 3              44.5               715              26    4.06
## 4              44.8                73              25    4.43
## 5              45.3               686              25    1.72
```

```
## 6              46                 665             24    1.68
## # i 15 more variables: `percentage expenditure` <dbl>, `Hepatitis B` <dbl>,
## #   Measles <dbl>, BMI <dbl>, `under-five deaths` <dbl>, Polio <dbl>,
## #   `Total expenditure` <dbl>, Diphtheria <dbl>, `HIV/AIDS` <dbl>, GDP <dbl>,
## #   Population <dbl>, `thinness  1-19 years` <dbl>, `thinness 5-9 years` <dbl>,
## #   `Income composition of resources` <dbl>, Schooling <dbl>
```

1.1 / Analysis of Data Distribution

Create a histogram of column "Life expectancy" column and overlay a normal curve.

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v stringr   1.5.0
## v forcats   1.0.0     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()     masks data.table::between()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x lubridate::hour()    masks data.table::hour()
## x lubridate::isoweek() masks data.table::isoweek()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x purrr::lift()        masks caret::lift()
## x lubridate::mday()    masks data.table::mday()
## x lubridate::minute()  masks data.table::minute()
## x lubridate::month()   masks data.table::month()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second()  masks data.table::second()
## x purrr::transpose()   masks data.table::transpose()
## x lubridate::wday()    masks data.table::wday()
## x lubridate::week()    masks data.table::week()
## x lubridate::yday()    masks data.table::yday()
## x lubridate::year()    masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```



The data apprears to be approximately normally distributed, but with a slight left skew. Most of life expectancy values are concentrated around the 67-70 range. Since it is slightly left skewed, it means that a majority of the countries have a life expectancy of around 67-70 or above.

```
##
##  Shapiro-Wilk normality test
##
## data:  life_exp_data$`Life expectancy`
## W = 0.96396, p-value < 2.2e-16
```

```
## Warning in ks.test.default(life_exp_data$`Life expectancy`, "pnorm",
## mean(life_exp_data$`Life expectancy`), : ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  life_exp_data$`Life expectancy`
## D = 0.11563, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

I performed a Shapiro-Wilk test and a Kolmogorov-Smirnov test to check the normality of the "Life expectancy" data. The Shapiro-Wilk test statistic was 0.96396, and the p-value was less than 2.2e-16. The Kolmogorov-Smirnov test statistic was 0.11563, and the p-value was also less than 2.2e-16.

Based on these p-values, we reject the null hypothesis that the data are normally distributed at a significance level of 0.05. This means that the "Life expectancy" data are not normally distributed.

However, it's important to note that these tests are sensitive to large sample sizes, and minor deviations from normality can lead to a significant p-value. Furthermore, the K-S test gave a warning about ties in the data, which can affect the reliability of the test results. Therefore, while these test results suggest non-normality, we should also consider the results from our histogram and other analyses, as well as the practical implications of the data distribution.

1.2 / Identification of Outliers

Identify any outliers for the columns using a Z-score deviation approach, i.e., consider any values that are more than 2.5 standard deviations from the mean as outliers.

```r
data<-life_exp_data
for (i in seq_along(data)) {
  if (is.numeric(data[[i]])) {
    mean_data <- mean(data[[i]], na.rm = TRUE)
    sd_data <- sd(data[[i]], na.rm = TRUE)
    zscore <- abs((data[[i]] - mean_data) / sd_data)
    data[zscore > 2.5, i] <- NA
    print(names(data)[i])
    print(sum(!is.na(data[, i])))
  }
}
```

```
## [1] "Life expectancy"
## [1] 1620
## [1] "Adult Mortality"
## [1] 1603
## [1] "infant deaths"
## [1] 1611
## [1] "Alcohol"
## [1] 1635
## [1] "percentage expenditure"
## [1] 1589
## [1] "Hepatitis B"
## [1] 1522
## [1] "Measles"
## [1] 1614
## [1] "BMI"
## [1] 1649
## [1] "under-five deaths"
## [1] 1614
## [1] "Polio"
## [1] 1547
## [1] "Total expenditure"
## [1] 1634
## [1] "Diphtheria"
## [1] 1556
## [1] "HIV/AIDS"
## [1] 1599
## [1] "GDP"
## [1] 1583
## [1] "Population"
## [1] 1635
## [1] "thinness  1-19 years"
## [1] 1587
## [1] "thinness 5-9 years"
## [1] 1585
## [1] "Income composition of resources"
## [1] 1601
```

```
## [1] "Schooling"
## [1] 1625
```

```
# Determinig the min, max, sd and median of column 'Life Expectancy'
print(paste("The min of Life Expectancy column is =", min(life_exp_data$`Life expectancy`, na.rm = TRUE
```

```
## [1] "The min of Life Expectancy column is = 44"
```

```
print(paste("The max of Life Expectancy column is =", max(life_exp_data$`Life expectancy`, na.rm = TRUE
```

```
## [1] "The max of Life Expectancy column is = 89"
```

```
print(paste("The SD of Life Expectancy column is =", sd(life_exp_data$`Life expectancy`, na.rm = TRUE))
```

```
## [1] "The SD of Life Expectancy column is = 8.7968341352386"
```

```
print(paste("The Median of Life Expectancy column is =", median(life_exp_data$`Life expectancy`, na.rm
```

```
## [1] "The Median of Life Expectancy column is = 71.7"
```

I identified outliers for each numeric column in the data using the Z-score method. For each column, I calculated the mean and standard deviation, and then considered any data points more than 2.5 standard deviations away from the mean as outliers.

The number of non-outlier data points identified for each column was printed out in the R console during the analysis. The specific outliers can be retrieved from the original data by using the `is.na()` function on the modified data to identify the locations of the outliers.

In terms of handling the outliers, my strategy would depend on the specifics of the data and the analysis. If I believe these outliers are due to errors or inconsistencies in the data collection, I might consider excluding them from the analysis. Alternatively, if the outliers could represent important phenomena, I might want to investigate them separately. Another approach could be imputing the outliers with the median of the rest of the data.

The maximum, minimum, standard deviation, and median of the "Life expectancy" column are 89, 44, 8.796, and 71.7 respectively.

A trimmed mean might be helpful for this data if it has significant outliers that are affecting the mean. The proportion of data identified as outliers could be used as a starting point for deciding how much of the data to trim when calculating the trimmed mean.

1.3 / Data Preparation

```
# Define a function to normalize a column with z-score standardization
normalize <- function(x) {
  return ((x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE))
}

# Apply the function to all numeric columns except the first three
life_exp_data_norm <- as.data.frame(lapply(life_exp_data[,], normalize))

# Check the results
summary(life_exp_data_norm)
```

```
##  Life.expectancy   Adult.Mortality   infant.deaths         Alcohol
##  Min.   :-2.8763   Min.   :-1.3344   Min.   :-0.26937   Min.   :-1.1226
##  1st Qu.:-0.5573   1st Qu.:-0.7279   1st Qu.:-0.26110   1st Qu.:-0.9241
##  Median : 0.2726   Median :-0.1613   Median :-0.24455   Median :-0.1845
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.6477   3rd Qu.: 0.4691   3rd Qu.:-0.08733   3rd Qu.: 0.6966
```

```
##  Max.   : 2.2392   Max.   : 4.4273   Max.   :12.97049   Max.   : 3.3100
##  percentage.expenditure  Hepatitis.B        Measles             BMI
##  Min.   :-0.3973        Min.   :-3.0158   Min.   :-0.2206   Min.   :-1.8289
##  1st Qu.:-0.3760        1st Qu.:-0.2038   1st Qu.:-0.2206   1st Qu.:-0.9430
##  Median :-0.3148        Median : 0.3821   Median :-0.2191   Median : 0.2820
##  Mean   : 0.0000        Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.1078        3rd Qu.: 0.6554   3rd Qu.:-0.1836   3rd Qu.: 0.8946
##  Max.   :10.3809        Max.   : 0.7726   Max.   :12.8117   Max.   : 1.9728
##  under.five.deaths      Polio          Total.expenditure    Diphtheria
##  Min.   :-0.27146   Min.   :-3.5885   Min.   :-2.26840   Min.   :-3.80715
##  1st Qu.:-0.26532   1st Qu.:-0.1142   1st Qu.:-0.67232   1st Qu.:-0.09988
##  Median :-0.24690   Median : 0.4203   Median :-0.05042   Median : 0.36353
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.:-0.09343   3rd Qu.: 0.5984   3rd Qu.: 0.65847   3rd Qu.: 0.59524
##  Max.   :12.62004   Max.   : 0.6875   Max.   : 3.66797   Max.   : 0.68792
##     HIV.AIDS            GDP             Population       thinness..1.19.years
##  Min.   :-0.3123   Min.   :-0.48487   Min.   :-0.20797   Min.   :-1.0329
##  1st Qu.:-0.3123   1st Qu.:-0.44475   1st Qu.:-0.20525   1st Qu.:-0.7068
##  Median :-0.3123   Median :-0.34624   Median :-0.18782   Median :-0.4024
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:-0.2128   3rd Qu.:-0.07385   3rd Qu.:-0.09927   3rd Qu.: 0.4891
##  Max.   : 8.0592   Max.   : 9.89959   Max.   :18.15496   Max.   : 4.8594
##  thinness.5.9.years Income.composition.of.resources   Schooling
##  Min.   :-1.0331    Min.   :-3.4494                 Min.   :-2.83320
##  1st Qu.:-0.6893    1st Qu.:-0.6694                 1st Qu.:-0.65103
##  Median :-0.3670    Median : 0.2264                 Median : 0.06443
##  Mean   : 0.0000    Mean   : 0.0000                 Mean   : 0.00000
##  3rd Qu.: 0.4711    3rd Qu.: 0.6524                 3rd Qu.: 0.67258
##  Max.   : 5.0050    Max.   : 1.6628                 Max.   : 3.06938
```

```
head(life_exp_data_norm)
```

```
##   Life.expectancy Adult.Mortality infant.deaths   Alcohol
## 1      -0.4890742       0.7563994     0.2436708 -1.122607
## 2      -1.0688282       0.8202408     0.2602207 -1.122607
## 3      -1.0688282       0.7963003     0.2767705 -1.122607
## 4      -1.1142991       0.8282210     0.3015952 -1.122607
## 5      -1.1484023       0.8521615     0.3181451 -1.122607
## 6      -1.1938732       0.8840823     0.3429698 -1.122607
##   percentage.expenditure Hepatitis.B     Measles        BMI under.five.deaths
## 1             -0.3568005  -0.5552780 -0.10613873 -0.9632673         0.2380623
## 2             -0.3555250  -0.6724442 -0.17177555 -0.9885784         0.2564787
## 3             -0.3556980  -0.5943334 -0.17792281 -1.0138894         0.2748951
## 4             -0.3528757  -0.4771673  0.05577204 -1.0392004         0.2994504
## 5             -0.3932838  -0.4381119  0.07817978 -1.0594492         0.3240056
## 6             -0.3520258  -0.5162227 -0.02334908 -1.0847602         0.3546997
##        Polio Total.expenditure Diphtheria   HIV.AIDS        GDP Population
## 1 -3.4549068         0.9585497 -0.8876720 -0.3122939 -0.4341074  0.2708311
## 2 -1.1387060         0.9672477 -1.0266948 -0.3122939 -0.4316294 -0.2033205
## 3 -0.9605367         0.9455027 -0.9340130 -0.3122939 -0.4299695  0.2423782
## 4 -0.7378251         1.1151133 -0.7949901 -0.3122939 -0.4266396 -0.1555011
## 5 -0.6932828         0.8324290 -0.7486492 -0.3122939 -0.4794826 -0.1656963
## 6 -0.7823674         1.4108445 -0.8413311 -0.3122939 -0.4368026 -0.1670507
##   thinness..1.19.years thinness.5.9.years Income.composition.of.resources
## 1             2.685095           2.662846                      -0.8332094
```

```
## 2             2.750323          2.750323                      -0.8495949
## 3             2.793808          2.748798                      -0.8823659
## 4             2.837294          2.813262                      -0.9205987
## 5             2.902522          2.856238                      -0.9697552
## 6             2.946008          2.899214                      -1.0025262
##    Schooling
## 1 -0.7225799
## 2 -0.7583531
## 3 -0.7941263
## 4 -0.8298995
## 5 -0.9372192
## 6 -1.0445388
```

```
# Create a new column 'outlook'
life_exp_data <- life_exp_data %>% mutate(outlook = ifelse(`Life expectancy` >= 70, 'good', 'not good'))

# Check the first few rows of the updated dataframe
head(life_exp_data)
```

```
## # A tibble: 6 x 20
##   `Life expectancy` `Adult Mortality` `infant deaths` Alcohol
##               <dbl>             <dbl>           <dbl>   <dbl>
## 1              65                 263              62    0.01
## 2              59.9               271              64    0.01
## 3              59.9               268              66    0.01
## 4              59.5               272              69    0.01
## 5              59.2               275              71    0.01
## 6              58.8               279              74    0.01
## # i 16 more variables: `percentage expenditure` <dbl>, `Hepatitis B` <dbl>,
## #   Measles <dbl>, BMI <dbl>, `under-five deaths` <dbl>, Polio <dbl>,
## #   `Total expenditure` <dbl>, Diphtheria <dbl>, `HIV/AIDS` <dbl>, GDP <dbl>,
## #   Population <dbl>, `thinness  1-19 years` <dbl>, `thinness 5-9 years` <dbl>,
## #   `Income composition of resources` <dbl>, Schooling <dbl>, outlook <chr>
```

```
# Drop the 'Life expectancy' column
life_exp_data <- life_exp_data %>% select(-`Life expectancy`)

# Check the first few rows of the updated dataframe
head(life_exp_data)
```

```
## # A tibble: 6 x 19
##   `Adult Mortality` `infant deaths` Alcohol percentage expenditu~1 `Hepatitis B`
##               <dbl>           <dbl>   <dbl>                  <dbl>         <dbl>
## 1               263              62    0.01                  71.3             65
## 2               271              64    0.01                  73.5             62
## 3               268              66    0.01                  73.2             64
## 4               272              69    0.01                  78.2             67
## 5               275              71    0.01                   7.10            68
## 6               279              74    0.01                  79.7             66
## # i abbreviated name: 1: `percentage expenditure`
## # i 14 more variables: Measles <dbl>, BMI <dbl>, `under-five deaths` <dbl>,
## #   Polio <dbl>, `Total expenditure` <dbl>, Diphtheria <dbl>, `HIV/AIDS` <dbl>,
## #   GDP <dbl>, Population <dbl>, `thinness  1-19 years` <dbl>,
## #   `thinness 5-9 years` <dbl>, `Income composition of resources` <dbl>,
## #   Schooling <dbl>, outlook <chr>
```

1.4 / Sampling Training and Validation Data

```
# Set seed for reproducibility
set.seed(123)

# Randomly shuffle the data
life_exp_data <- life_exp_data %>% sample_frac(1)

# Check the number of rows for each 'outlook' type
life_exp_data %>% group_by(outlook) %>% summarise(rows = n())
```

```
## # A tibble: 2 x 2
##   outlook   rows
##   <chr>    <int>
## 1 good       895
## 2 not good   754
```

```
# Split the data with 15% of each 'outlook' type going into the validation dataset
index <- createDataPartition(life_exp_data$outlook, p = 0.15, list = FALSE)

# Create the training and validation datasets
training_data <- life_exp_data[-index,]
validation_data <- life_exp_data[index,]

# Verify the number of rows in the validation dataset for each 'outlook' type
validation_data %>% group_by(outlook) %>% summarise(rows = n())
```

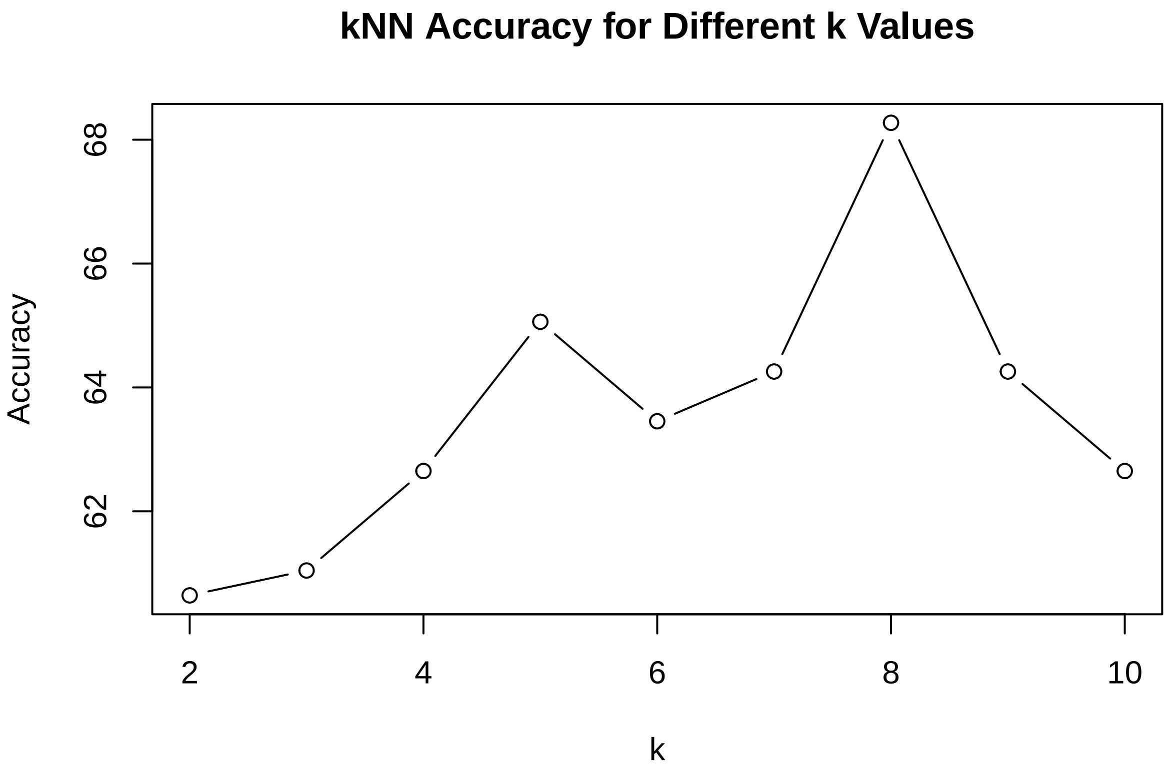
```
## # A tibble: 2 x 2
##   outlook   rows
##   <chr>    <int>
## 1 good       135
## 2 not good   114
```

1.5 / Predictive Modelling

```
## [1] good
## Levels: good not good
```

I replaced missing values in our data with median values, then used a method called k-Nearest Neighbors (kNN) to predict the quality of life for a new data point. This algorithm is effective as it uses the 'k' most similar data points to make its prediction, which in our case was 5 (k=5). The prediction from the kNN algorithm indicates that the quality of life for the given data point is classified as "good". This means that, based on the specific attributes of the data point and its similarity to other data points in the training set, the model predicts a positive quality of life outcome.

1.6 / Model Accuracy



The plot that was created visualizes the accuracy of a k-Nearest Neighbors (kNN) model as we vary the number of neighbors considered (k), ranging from 2 to 10. Each point on the plot indicates the percentage of correct classifications (accuracy) achieved by the kNN model for a specific value of k.

From the plot, we see that the accuracy fluctuates as we increase the value of k. The highest accuracy of around 68% is achieved when k is 7. Thus, through this analysis, it would be reasonable to choose k=7 for the final model as it provides the highest prediction accuracy according to the validation data.

Problem 2

2 / Predicting Age of Abalones using Regression kNN

```
library(readr)
library(dplyr)

data_url <- "https://s3.us-east-2.amazonaws.com/artificium.us/datasets/abalone.csv"

abalone_data <- read_csv(data_url)
```

```
## Rows: 4177 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Sex
## dbl (8): Length, Diameter, Height, Whole weight, Shucked weight, Viscera wei...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

2.1 / Save the values of the "Rings" column in a separate vector called target_data

```
target_data <- abalone_data$Rings

train_data <- select(abalone_data, -Rings)
```

2.2 / Encoding Categorical Variables

```
train_data <- mutate(train_data,
                     Sex_M = ifelse(Sex == "M", 1, 0),
                     Sex_F = ifelse(Sex == "F", 1, 0),
                     Sex_I = ifelse(Sex == "I", 1, 0)) %>%
  select(-Sex)
```

I'm using one-hot encoding for the categorical feature 'Sex', because this method results in binary vectors that are easy to compute, and doesn't imply any order (which is appropriate for the 'Sex' feature)

2.3 / Normalize all the columns in train_data using min-max normalization

```
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

train_data <- as.data.frame(lapply(train_data, normalize))
```

2.4 / Build (write) a function called knn.reg

```
knn.reg <- function(new_data, target_data, train_data, k) {

  # Euclidean distances between new_data and train_data
  distances <- apply(train_data, 1, function(x) sqrt(sum((x - new_data)^2)))

  # Find the k nearest neighbors
  neighbors <- order(distances)[1:k]

  # Define weights
  weights <- c(2, 1.5, rep(1, k - 2))

  # Calculate the weighted average of the Rings values
  predicted_value <- sum(target_data[neighbors] * weights) / sum(weights)

  return(predicted_value)
}
```

2.5 / Forecast the number of Rings of this new abalone

```
# Define new abalone data
new_abalone <- data.frame(
  Length = 0.34,
  Diameter = 0.491,
  Height = 0.245,
  WholeWeight = 0.4853,
  ShuckedWeight = 0.2532,
```

```r
  VisceraWeight = 0.0887,
  ShellWeight = 0.19,
  Sex_M = 1,
  Sex_F = 0,
  Sex_I = 0
)

# Normalizing new_abalone data
new_abalone_normalized <- as.data.frame(lapply(new_abalone, normalize))

# Predict the Rings value for the new abalone data using knn.reg
predicted_rings <- knn.reg(new_abalone_normalized, target_data, train_data, k = 3)
print(predicted_rings)
```

```
## [1] 11
```

2.6 / Calculate the Mean Squared Error (MSE)

```r
# Split data into train and test datasets
set.seed(123)
train_index <- sample(1:nrow(abalone_data), nrow(abalone_data)*0.85)

# Prepare train and test datasets
train_data_mse <- train_data[train_index, ]
target_data_mse <- target_data[train_index]

test_data <- train_data[-train_index, ]
test_target_data <- target_data[-train_index]

# Predict the Rings values for the test data
predicted_rings_mse <- apply(test_data, 1, knn.reg, target_data = target_data_mse, train_data = train_da

# Calculate the Mean Squared Error (MSE)
mse <- mean((test_target_data - predicted_rings_mse)^2)
print(mse)
```

```
## [1] 5.40408
```

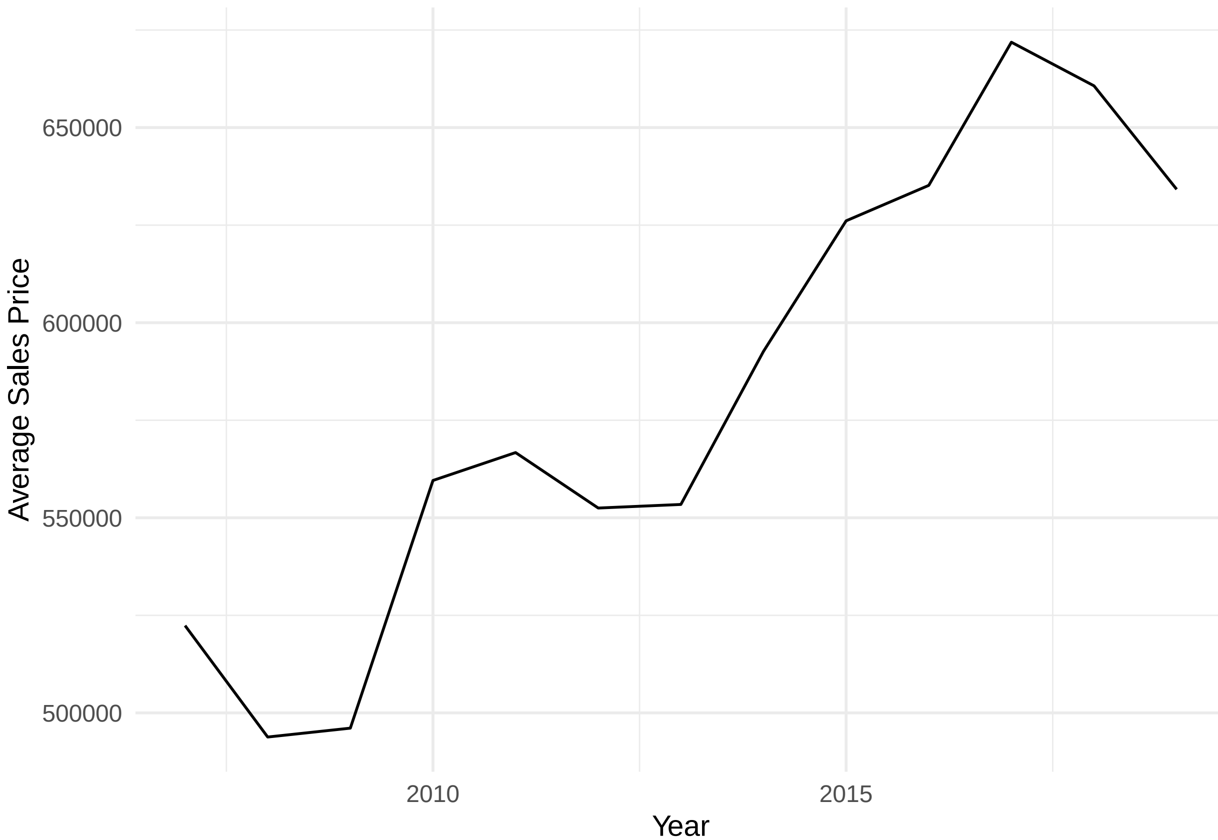
Problem 3

3 / Forecasting Future Sales Price

```
## 3 / Forecasting Future Sales Price

## We obtained a data set containing 29580 sales transactions for the years 2007 to 2019 .

## The mean sales price for the entire time frame was $ 609736.3  (sd =  281707.9 ).

## Broken down by year, we have the following average sales prices per year:

## # A tibble: 13 x 2
##     year avg_price
##    <dbl>     <dbl>
##  1  2007   522377.
##  2  2008   493814.
##  3  2009   496092.
```

```
##  4  2010   559565.
##  5  2011   566715.
##  6  2012   552501.
##  7  2013   553416.
##  8  2014   592654.
##  9  2015   626101.
## 10  2016   635185.
## 11  2017   671881.
## 12  2018   660701.
## 13  2019   634184.
```

```
##
## As the graph below shows, the average sales price per year has been  increasing .
```




```
##
## Using a weighted moving average forecasting model that averages the prior 3 years (with weights of 4
## we predict next year's average sales price to be around $ 662976.2 .
```