Project 3

Courtney Hodge

2023-12-02

Introduction:

1. The dataset that I chose is called "Life expectancy & Socio-Economic (world bank)" and it comes from Kaggle.com. https://www.kaggle.com/datasets/mjshri23/life-expectancy-and-socio-economic-world-bank
2. There are 3306 obs. within this dataset.
3. There are 16 variables from the data and they are the following: Country Name, Country Code, Region, IncomeGroup, Year, Life Expectancy World Bank, Prevalence of Undernourishment, CO2, Health Expenditure, Education Expenditure, Unemployment, Corruption, Sanitation, Injuries, Communicable, and NonCommunicable.
4. Some interesting variables are the following: Corruption - Transparency,accountability, and corruption in the public sector assets Prevalence of Undernourishment (% of the population)- the percentage of the population whose habitual food consumption is insufficient to provide the dietary energy level that are required to maintain a normally active and healthy life. Health Expenditure - level of current health expenditure expressed as a percentage of GDP. Estimates of currenty health expenditures include healthcare goods and services consumed during each year. This idicator does not incldue capital health expenditures such as buildings, machinery, IT, and stocks of vaccines for emergencies or outbreaks.

Primary Research Questions:

Primary Question:

How does a country's characteristics reflect their quality of life and how does that compare to different countries from different regions. For example, how does Afghanistan's CO2, Unemployment, and Life expectancy reflect its quality of life for a given year and how does Afghanistan's quality of life compare to another Middle Eastern country? A European country? An Asian Country? So on and so forth.

Secondary Questions:

A) Do factors like corruption and unemployment rate impact life expectancy?
B) Does increase in CO2 emissions decrease life expectancy?
C) How does a country's income class affect their unemployment rate?

```
data = read.csv("C:/Users/hodge/OneDrive - Baylor University/Desktop/Computational Statistics/project da
```

Exploratory Data Analysis

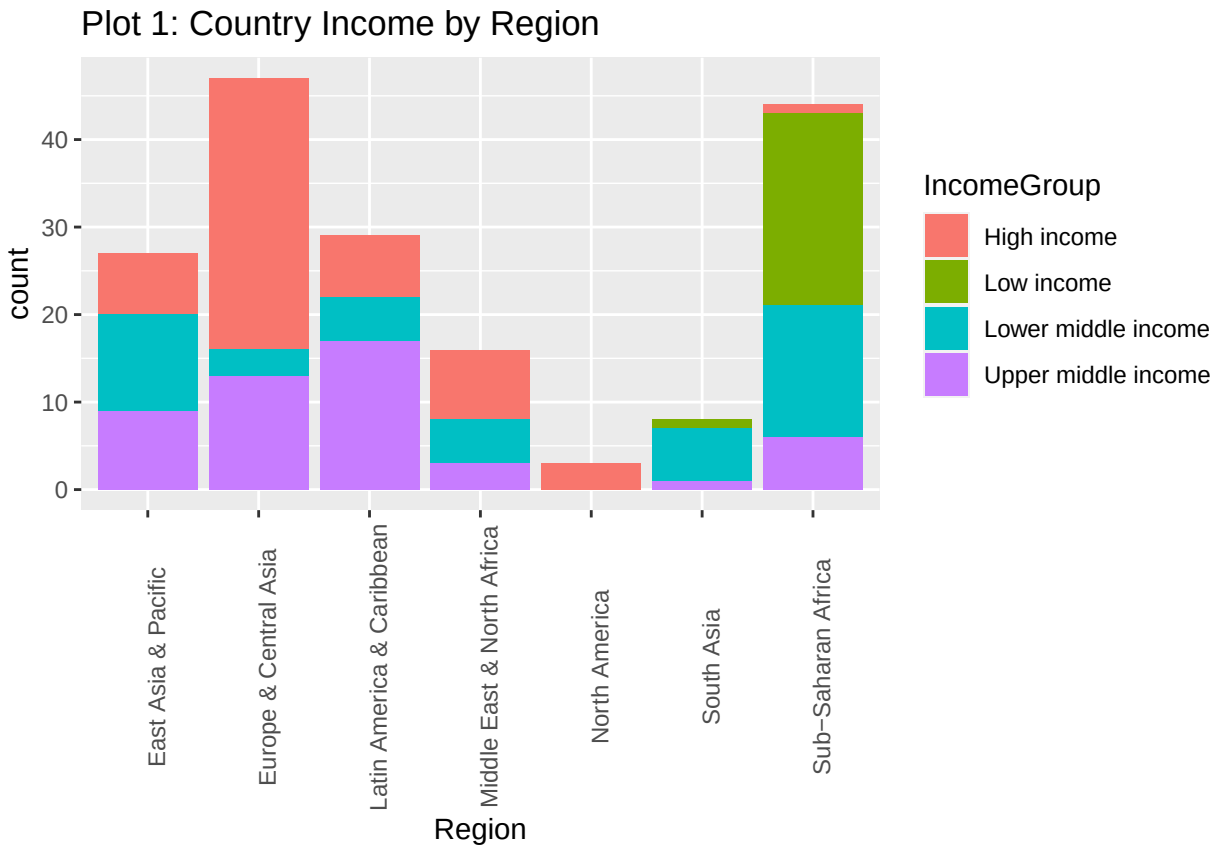
1. Make a plot involving at least one categorical variable. You can include other variables as well but it must include a categorical variable.

```
library(tidyverse)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
data |>
  filter(Year == "2019") |>
  ggplot(aes(x = Region, na.rm = TRUE, fill = IncomeGroup)) +
  geom_bar() +
  labs(title = "Plot 1: Country Income by Region") +
  theme(axis.text.x = element_text(angle = 90))
```



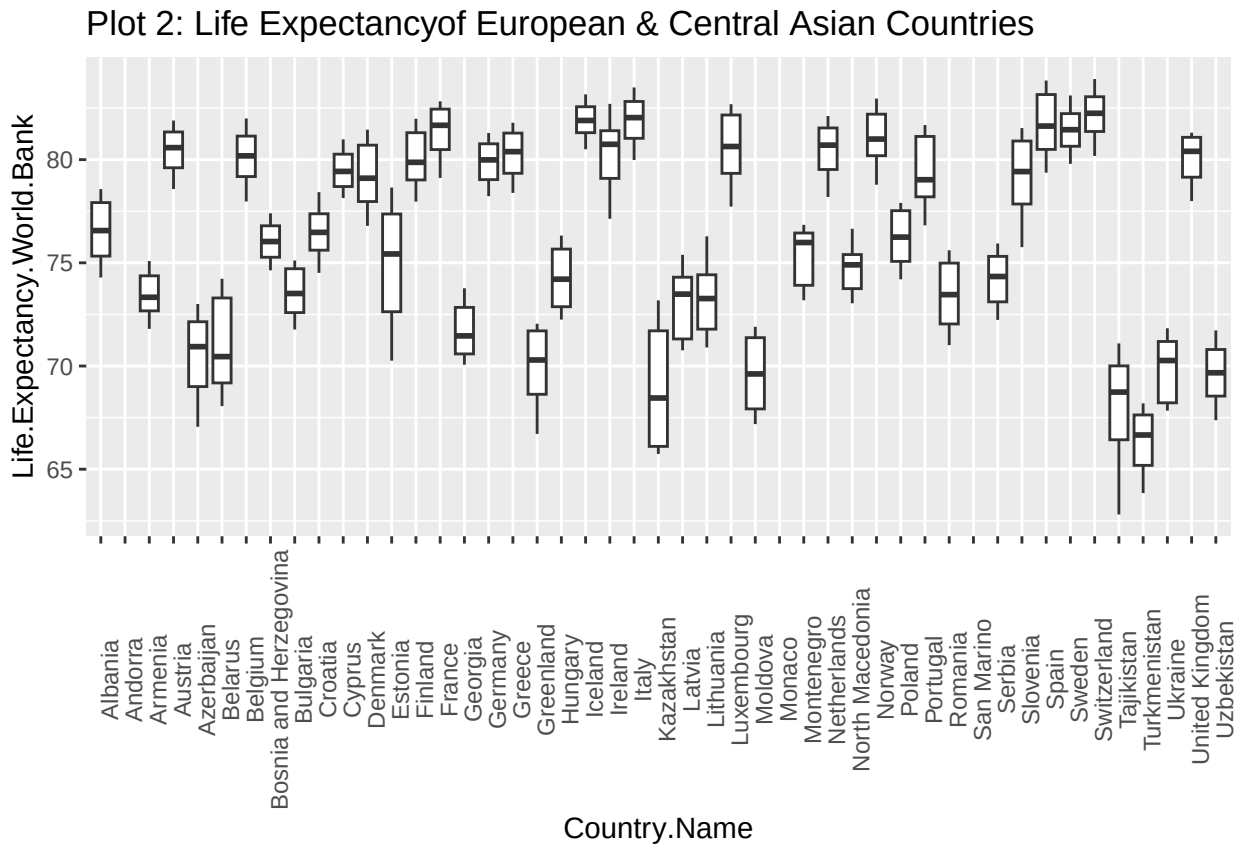
2. Comment on any pattern you see in this plot.

I made a stacked bar graph looking at country income by region. I filtered this bar graph to look at only one year, 2019. If the year was changed, the same bar graph would show so in reality, the year doesn't matter. Some of the patterns that this stacked bar graph shows is that in only South Asia and Sub-Saharan Africa do "Low income" countries appear in this data. This makes sense because usually these regions of the world are less wealthy than others. North America is the only region in the world that has a couple of "High income" countries. From this data, it would be helpful to know what the boundaries are for each Income Group so that the information is clearly understandable. Other patterns that are seen is that Europe & Central Asia have the largest amount of "High income" countries than all others; Latin America & Caribbean have the largest amount of "Upper middle income" countries; and lastly Sub-Saharan Africa has the largest amount of "Lower middle income" and "Low income" countries out of the other regions.

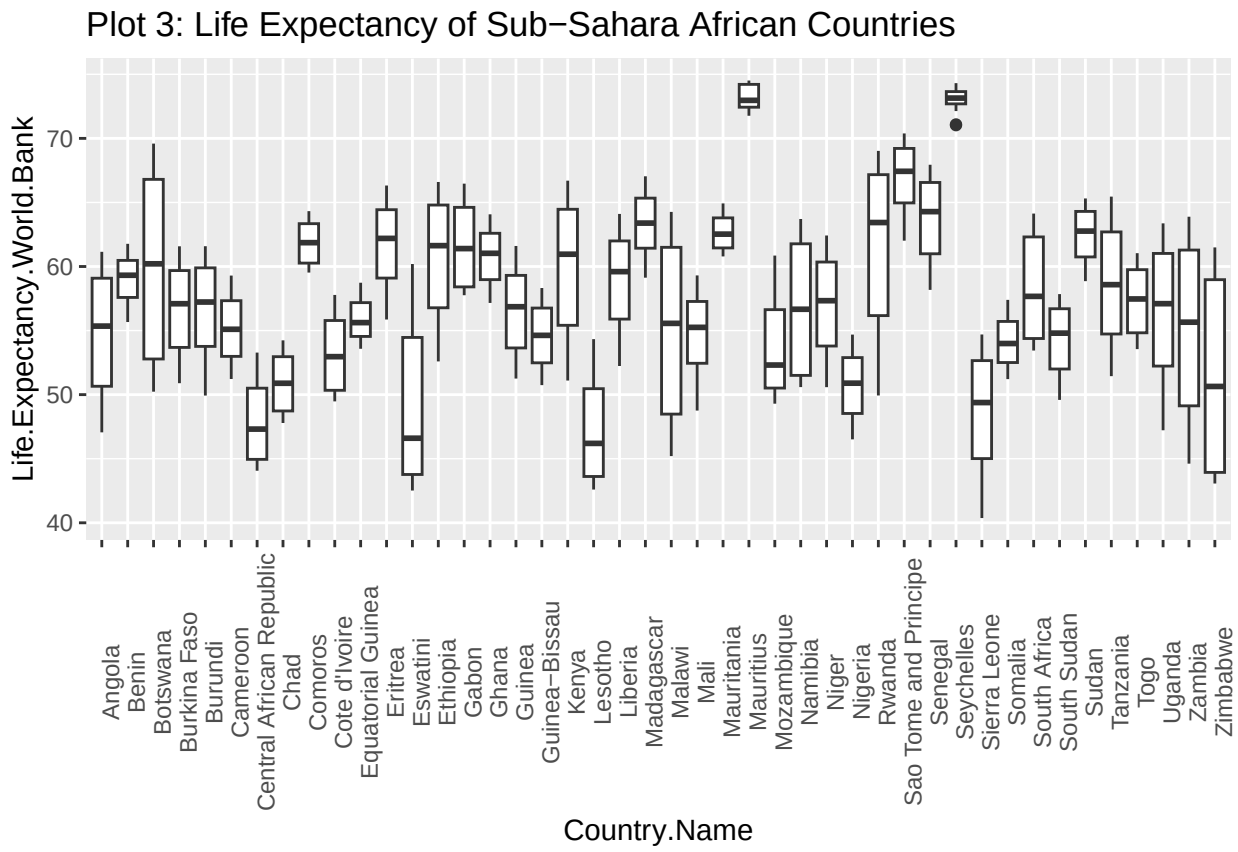
3. Make a second plot involving at least one quantitative variable. You can include other variables but it must include a quantitative variable. This plot must be different than the plot in part 1.

```
data |>
  filter(Region == "Europe & Central Asia", na.rm = TRUE) |>
  ggplot( aes(x = Country.Name, y = Life.Expectancy.World.Bank)) +
  geom_boxplot() +
  labs(title = "Plot 2: Life Expectancyof European & Central Asian Countries") +
  theme(axis.text.x = element_text(angle = 90))
```

```
## Warning: Removed 57 rows containing non-finite values (`stat_boxplot()`).
```



```r
data |>
  filter(Region == "Sub-Saharan Africa", na.rm = TRUE) |>
  ggplot( aes(x = Country.Name, y = Life.Expectancy.World.Bank)) +
  geom_boxplot() +
  labs(title = "Plot 3: Life Expectancy of Sub-Sahara African Countries") +
  theme(axis.text.x = element_text(angle = 90))
```



4. Comment on any pattern you see in this plot.

In this plot, I made a box plot for the recorded life expectancy for each European and Central Asian country represented in this data. There are 47 countries plotted and the range for the entire region is between 60 to 90 years old. One of the patterns observed are that there is not a single country from this region has an outlier life expectancy age from 2001 to 2019. Tajikistan has the lowest recorded life expectancy while Switzerland and Spain have the highest recorded life expectancies.

Just for comparison purposes, I made a boxplot for life expectancies for Sub-Sahara African countries. There are 44 countries represented in this region. The range for life expectancies from 2001 to 2019 fall between 40 to 75 years old, which is vaslty different from European and Central Asian countries. Sierra Leone has the lowest life expectancy out of all the countries and Mauritius has the highest life expectancy, followed by Seychelles. Most of the countries life expectancies fall between 50 and 70 years old. The only country that has an outlier life expectancy was Seychelles.

Regression Prediction

In this portion of the assignment, I will do regression to conduct a k-nearest neighbors on the data. The variable that I will choose to predict is 'Prevelance.of.Undernourishment', known in the later code chunks as 'Undernourished'. Undernourished is the percentage of the "population whose habitual food consumption is insufficient to provide the dietary energy levels that are required to maintain a normally active and healthy life". The values for this variable range between 2.5 to 26.1.

Aligning with the first question A form the research questions I created in HW 10, I chose to predict Undernourished against 'Life.Expectancy.World.Bank' (known as Life in the following code chunks) because after plotting several variables against Life, I found that there was a patter between these two variables. The plot below shows Undernourished v Life. Based on the data generated by all the regions of the world, we can see that the younger people are, the more sparse the undernourished values are. As you get closer to the bottom right corner, you can see that the undernourished amount tails off.

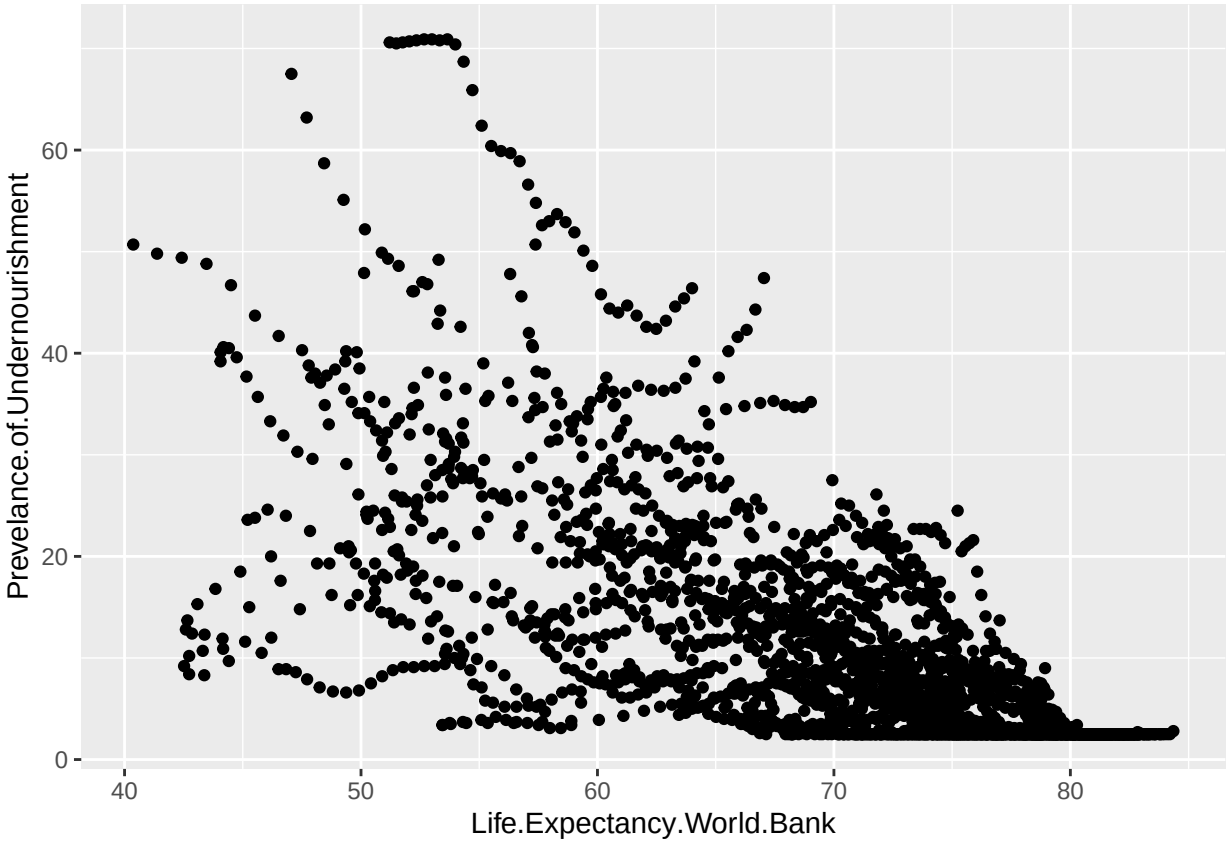
In this regression prediction section of the project, a regression using knn will be performed on the entire dataset, removing NAs from Life and Undernourished. See conclusion for a synopsis of what was done!

```
#this is very good!
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.1.1 --

## v broom        1.0.5     v rsample      1.2.0
## v dials        1.2.0     v tune         1.1.2
## v infer        1.0.5     v workflows    1.1.3
## v modeldata    1.2.0     v workflowsets 1.0.1
## v parsnip      1.1.1     v yardstick    1.2.0
## v recipes      1.0.8

## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
library(tidyverse)

data |>
  ggplot(aes(x = Life.Expectancy.World.Bank, y = Prevelance.of.Undernourishment)) +
  geom_point()
```

```
## Warning: Removed 702 rows containing missing values (`geom_point()`).
```




```
  #labs(title = "Plot 1: Country Income by Region") +
  #theme(axis.text.x = element_text(angle = 90))
```

```
data = data |>
  mutate(Life = Life.Expectancy.World.Bank) |>
  mutate(Undernourished = Prevelance.of.Undernourishment)

data = data |>
  filter(!is.na(Life)) |>
  filter(!is.na(Undernourished))
```

```
#split data
data_split = initial_split(data, prop = .75, strata = Undernourished)

data_train = training(data_split)
data_test = testing(data_split)
```

```
#set up data processing recipe
data_recipe = recipe(Undernourished~Life, data = data_train) |>
  step_normalize(all_predictors())
```

```
#set up model
data_model = nearest_neighbor(weight_func = "rectangular",
                              neighbors = 5) |>
```

```
  set_engine("kknn") |>
  set_mode("regression")
```

```
#combine recipe and model into workflow and fit to data
data_wf = workflow() |>
  add_recipe(data_recipe) |>
  add_model(data_model) |>
  fit(data = data_train)
```

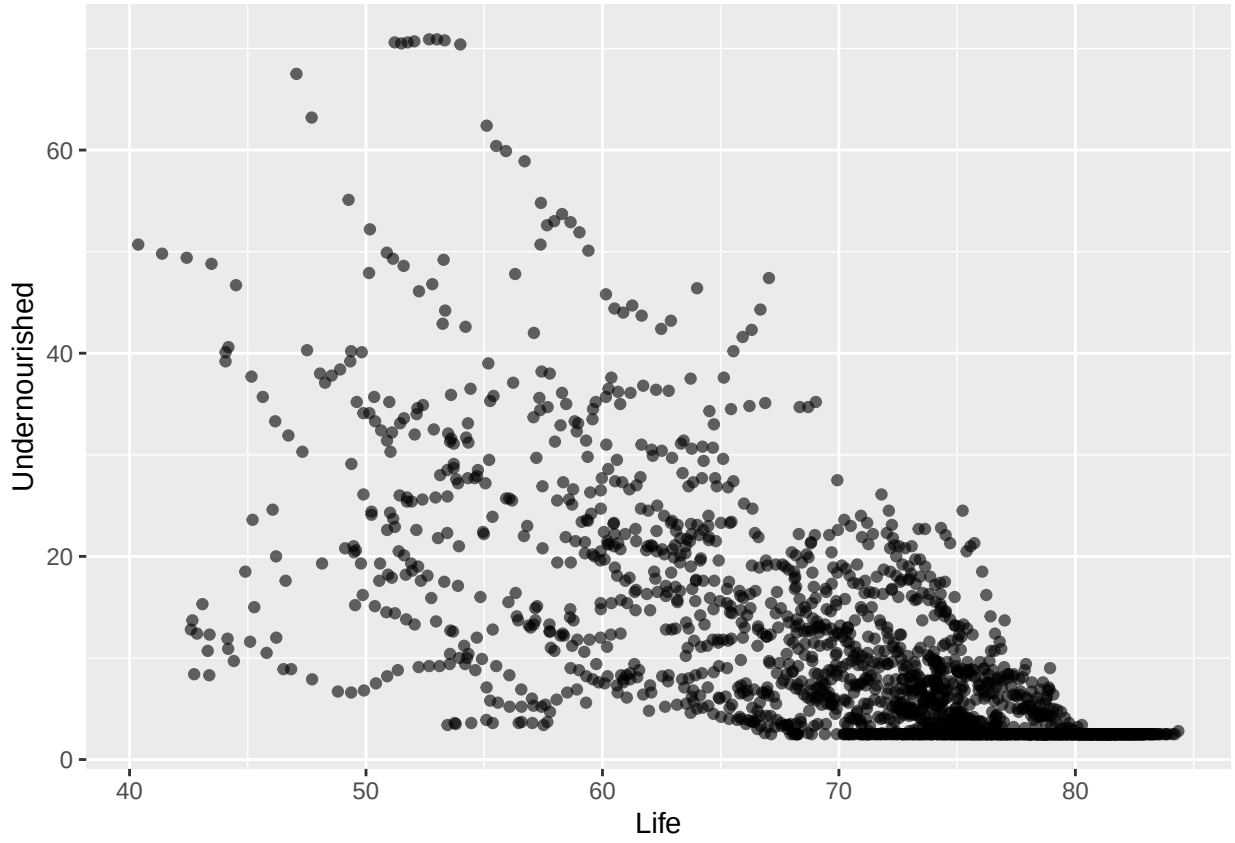
```
#predict train data
data_pred = predict(data_wf, data_train) |>
  bind_cols(data_train) |>
  metrics(truth = Undernourished, estimate = .pred)

data_pred
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       7.39
## 2 rsq     standard       0.576
## 3 mae     standard       4.61
```

```
#check test data
train_pred = predict(data_wf, data_test) |>
  bind_cols(data_test) |>
  metrics(truth = Undernourished, estimate = .pred)

train_pred
```

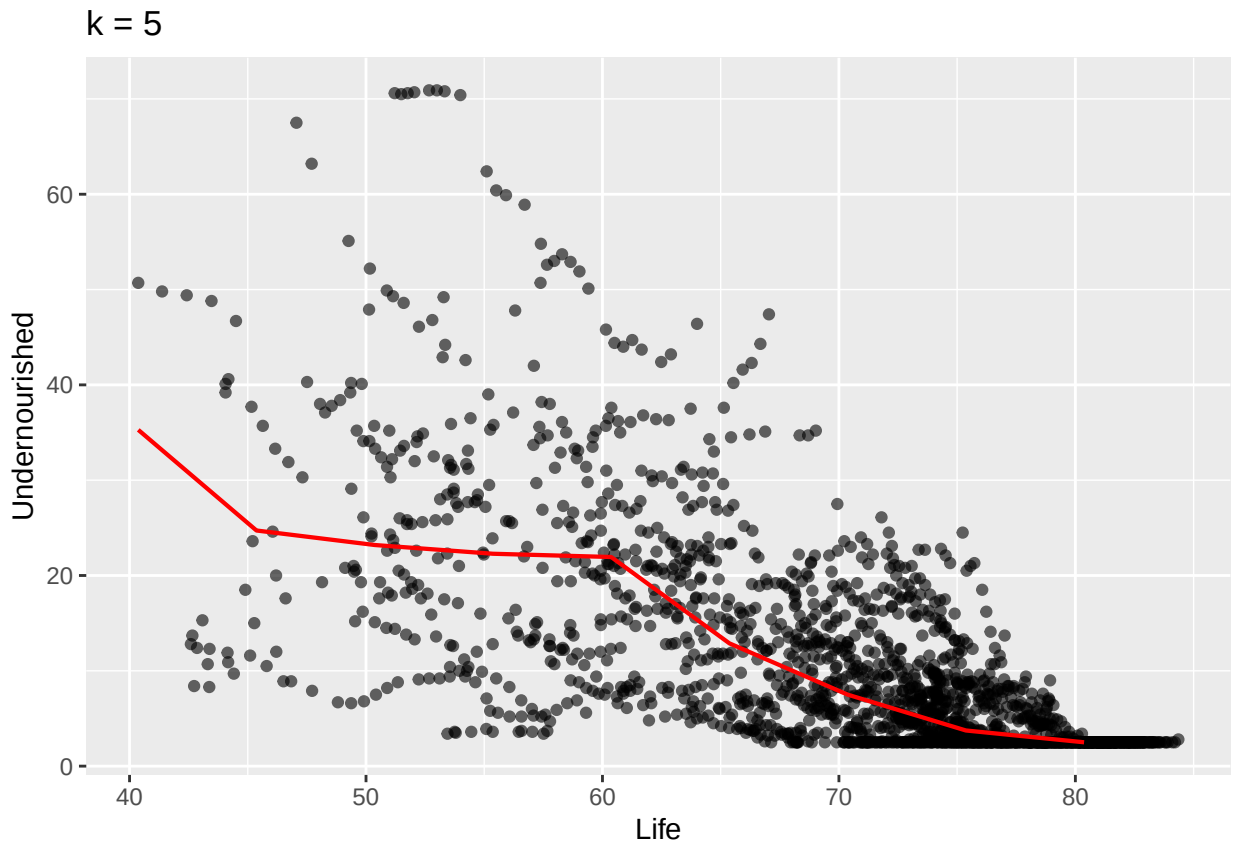
```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       8.84
## 2 rsq     standard       0.394
## 3 mae     standard       5.59
```

```
#plot entire dataset
data_train |>
  ggplot(aes(x = Life, y = Undernourished)) +
  geom_point(alpha = .6)
```




```
#make a sequence of values for Life to predict at
finegrid <- tibble(Life = seq(from = 40.369, to = 84.35634, by = 5))
```

```
#predict at those values
pred.func = predict(data_wf, finegrid) |>
  bind_cols(finegrid)

data_train |>
  ggplot(aes(x = Life, y = Undernourished)) +
  geom_point(alpha = .6) +
  geom_line(data = pred.func,
            aes(x = Life, y = .pred),
            col = "red", lwd = .75) +
  ggtitle("k = 5")
```

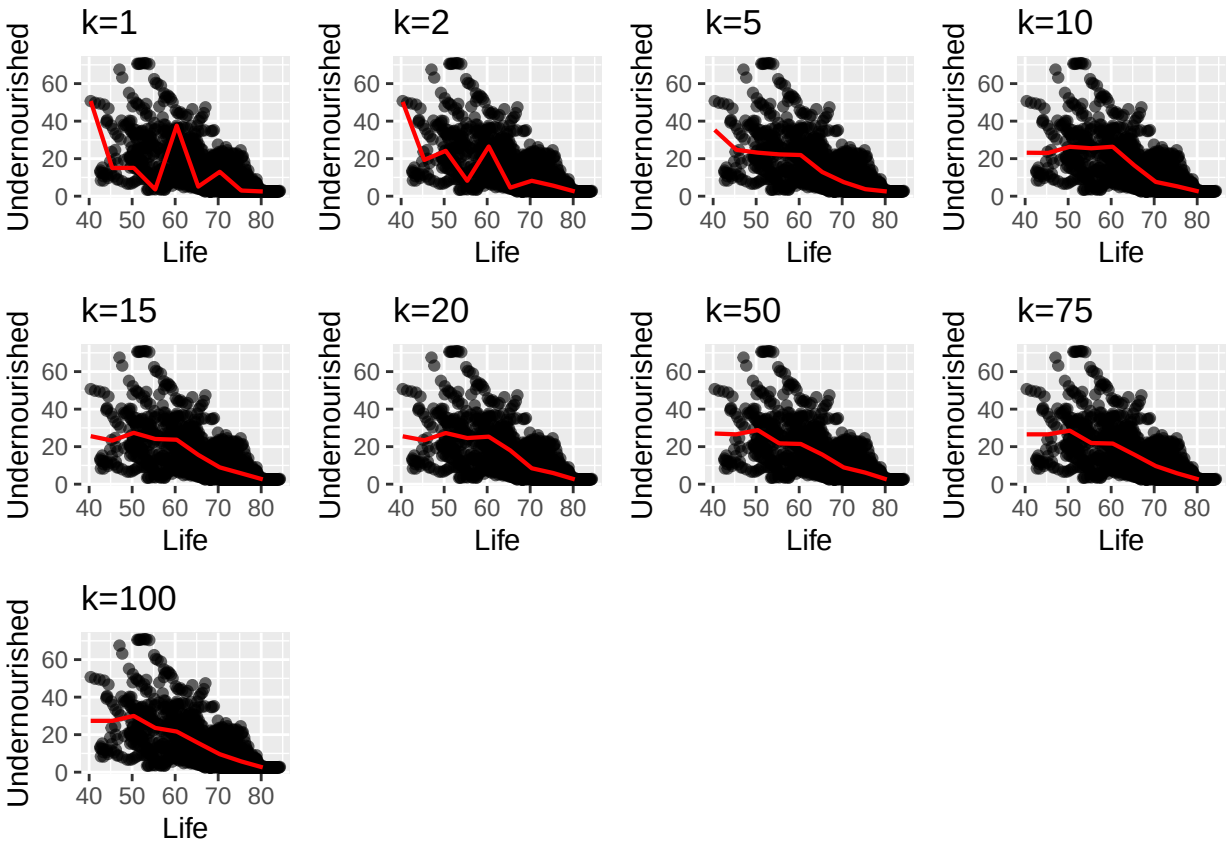



```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
plots = list()
ks = c(1,2,5,10,15,20,50,75,100)

for(i in 1:length(ks)){
  data_model = nearest_neighbor(weight_func = "rectangular",
                                neighbors = ks[i]) |>
    set_engine("kknn") |>
    set_mode("regression")

data_wf = workflow() |>
  add_recipe(data_recipe) |>
  add_model(data_model) |>
  fit(data = data_train)

pred.func = predict(data_wf, finegrid) |>
  bind_cols(finegrid)
```

```
plots[[i]] = data_train |>
  ggplot(aes(x = Life, y = Undernourished)) +
  geom_point(alpha = .6) +
  geom_line(data = pred.func,
            aes(x = Life, y = .pred),
            col = "red", lwd = .75)+
  ggtitle(paste0("k=", ks[i]))

}

grid.arrange(grobs = plots, ncol = 4)
```

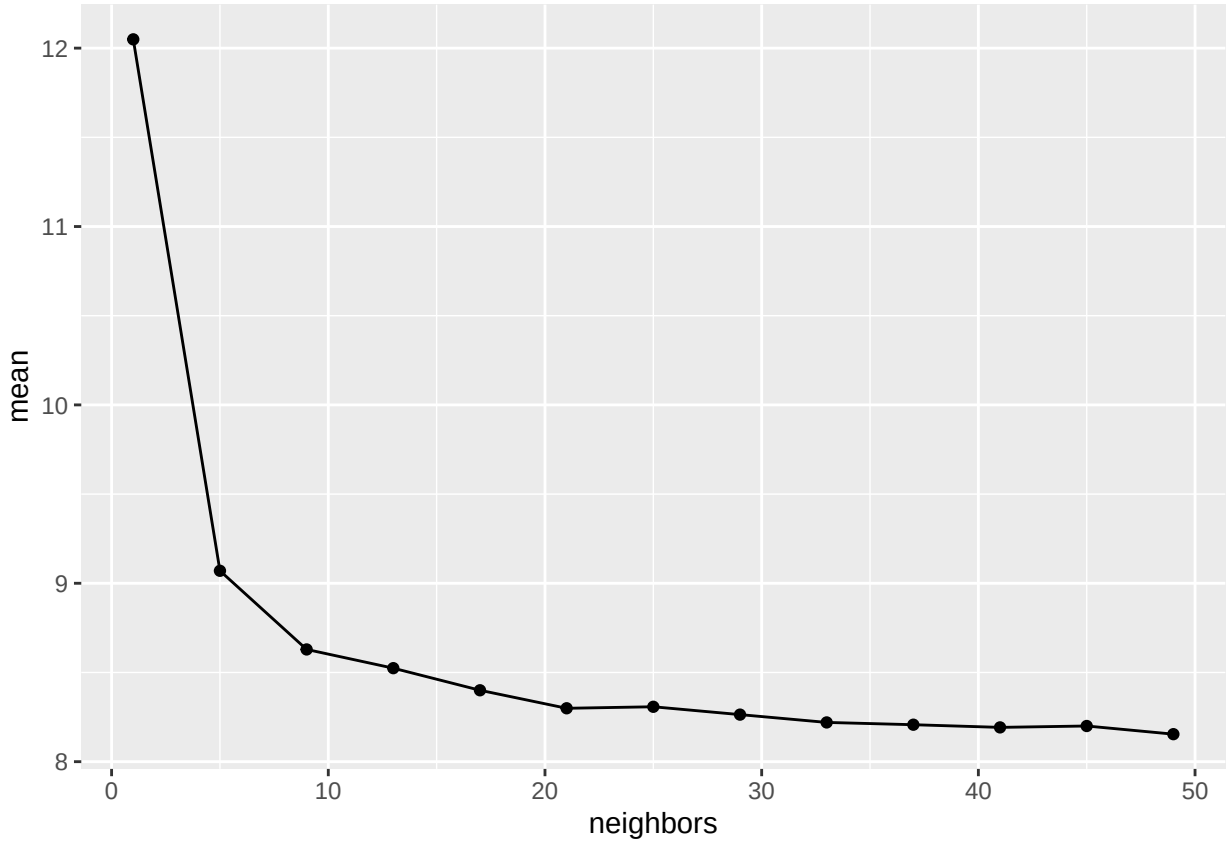



```
#find the optimal value of k using cross validation
data_vfold = vfold_cv(data_train, v = 5, strata = Undernourished)

data_recipe = recipe(Undernourished~Life, data = data_train) |>
  step_normalize(all_predictors())

data_model = nearest_neighbor(weight_func = "rectangular",
                              neighbors = tune()) |>
  set_engine("kknn") |>
  set_mode("regression")
```

```
gridvals = tibble(neighbors = seq(from = 1, to = 50, by = 4))

results = workflow() |>
  add_recipe(data_recipe) |>
  add_model(data_model) |>
  tune_grid(resamples = data_vfold, grid = gridvals) |>
  collect_metrics() |>
  filter(.metric == "rmse")

results
```

```
## # A tibble: 13 x 7
##    neighbors .metric .estimator  mean     n std_err .config
##        <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
##  1         1 rmse    standard   12.0      5   0.273 Preprocessor1_Model01
##  2         5 rmse    standard    9.07     5   0.269 Preprocessor1_Model02
##  3         9 rmse    standard    8.63     5   0.199 Preprocessor1_Model03
##  4        13 rmse    standard    8.52     5   0.203 Preprocessor1_Model04
##  5        17 rmse    standard    8.40     5   0.168 Preprocessor1_Model05
##  6        21 rmse    standard    8.30     5   0.137 Preprocessor1_Model06
##  7        25 rmse    standard    8.31     5   0.142 Preprocessor1_Model07
##  8        29 rmse    standard    8.26     5   0.142 Preprocessor1_Model08
##  9        33 rmse    standard    8.22     5   0.144 Preprocessor1_Model09
## 10        37 rmse    standard    8.21     5   0.140 Preprocessor1_Model10
## 11        41 rmse    standard    8.19     5   0.130 Preprocessor1_Model11
## 12        45 rmse    standard    8.20     5   0.144 Preprocessor1_Model12
## 13        49 rmse    standard    8.15     5   0.138 Preprocessor1_Model13
```

```
results |>
  ggplot(aes(x = neighbors, y = mean)) +
  geom_point() +
  geom_line()
```




```
kmin = results |>
  filter(mean == min(mean)) |>
  select(neighbors)
kmin
```

```
## # A tibble: 1 x 1
##   neighbors
##       <dbl>
## 1        49
```

```
#now, set up your model with that kmin value!
#fit train data with the value of k that minimized rmse
data_model = nearest_neighbor(weight_func = "rectangular",
                              neighbors = kmin) |>
  set_engine("kknn") |>
  set_mode("regression")

data_wf = workflow() |>
  add_recipe(data_recipe) |>
  add_model(data_model) |>
  fit(data = data_train)
```

```
#predict train data
data_pred = predict(data_wf, data_train) |>
  bind_cols(data_train) |>
```

```
  metrics(truth = Undernourished, estimate = .pred)

data_pred
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       8.08
## 2 rsq     standard       0.494
## 3 mae     standard       5.09
```

```
#check test data
train_pred = predict(data_wf, data_test) |>
  bind_cols(data_test) |>
  metrics(truth = Undernourished, estimate = .pred)

train_pred
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       8.13
## 2 rsq     standard       0.473
## 3 mae     standard       5.15
```

Conclusion

To conclude, the above code compares Life and Undernourished to find the ideal k value to be able to predict undernourished. In our regression model, the RMSE metric for training data ended up being 7.4. For the test data, the RMSE ended up being 8.7. When finding the ideal k value for knn, it's evident that from the mean v neighbors graph, the ideal k value to predict the undernourished value before the k values start to smooth out is k = 5. Please reference the data above for further inquiry into any of the other parts of the knn model.

As it relates to the previous research questions I created, I forsee this regression model being able to predict the Undernourished rates of the different regions world based on the Life expectancy. I also believe that it would be more impressive to predict Life expectancy based on the undernourished level, so maybe in a future demonstration this could be done. Even though its not explicitly answering one of my previous research questions, I believe that it falls in line well with question A. Although a regression model had to be created before this answer could be solved, I believe that this will be a great tool to be able to use for future research into how life expectancy is altered by undernourishment and vice versa. Thank you for reading through my report!