Final Project

Code

Cheng Ye

2022-11-27

Abstract

Begun in 2000, the World Health Organization (WHO) has been tracking records of human life expectancy globally. The measurement of life expectancy yields important information regarding the overall health of a country. It summarizes the trends in mortality of given countries. This project analyzes the life expectancy around the world and tends to examine the relationship between life expectancy and factors not limited to education, illness, economy, mortality, and etc. To achieve this goal, we studied the integrated data of global life expectancy from 2000~2015 along with the components mentioned earlier, compared the effectiveness of elements, and utilized various models to verify the colinearity of factors. Our results showed some correlation between factors. After testing multiple models, we ultimately went with the best performing model based on our methods. However, additional research is needed to present a credible result, as the integrated data lacks information related to contemporary date. Besides the lack of recent dates, lack of information from certain countries undermine the generality of this research as it limits the variables and data used. In addition, the trade off between variance and bias within the modelling process may yield fluctuations in credibility. Further research may be combining data within a longer magnitude of time and integrating the data along with different sources of investigation instead of relying only on one source of data. In addition, alterations in test/train percentage should be included to examine the generality of results.

Introduction

Life expectancy tells us the average number of years of life that a person can expect to live within a given country. In other words, the higher the life expectancy of a given country, the higher the quality of living for average citizens of that country. Life expectancy is also one of the most used indicators for evaluating the overall health of a population. Because life expectancy is one nebulous concept, as multiple components may or may not affect it,the purpose of this project is to generate a model that will predict the relationship

between life expectancy and factors including education, illness, economy, mortality, and etc. Understanding the correlation between life expectancy and such factors allows policy makers to implement better policies to increase life quality of its citizens. The data of this project comes from the World Health Organization and United Nations. It is integrated by a non-profit third party Kaggle Datasets. Among all categories of health-related factors within the dataset, only those critical factors were chosen which are more representative.In this research, we will implement multiple techniques to yield the most accurate model for this project. In this project, we hypothesize that life expectancy is correlated with the development status of the given country This project is divided into the following sections:An abstract part concluding the origins, processes, and results of the research; a introduction part that explains the project with more details; an Exploratory Data Analysis section where we explain the usage of variables; a model building section where we set up the base components of our model; a model comparison section where we ascertain the best performing model; a conclusion part that discusses the outcomes of the models we used in this research and comments regarding the research.

Source of Data

The data-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation website.Apart from deriving directly from the WHO data source, the integrated data also includes other social/mortality factors.
The dataset comes from World Health Organization,available at https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-life-expectancy-and-healthy-life-expectancy (https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-life-expectancy-and-healthy-life-expectancy) The integrated data sources came from Non-profit Website Kaggle, available at https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who (https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who)

Loading necessary packages

Exploratory Data Analysis

Before we could start training our data or apply any models we need to view our data. Not everything is perfect and ready, in other words, there may exist missing statistics among the data which needs to be cleaned or there are some variables needed to be tweaked to avoid over-fitting. In this section, there will be processes where we need to clean/tidy our data;analyze key variables through graphing and comparing.
Processing Raw Data

Hide

```
Raw_data<- read.csv(file="C:/Cheng Ye/UCSB/PSTAT 231/Final Project/Life Expectancy Data.csv")
# Raw_data
```

From the raw dataset we could observe that there are 2938 observations and 22 variables. However, on initial visual inspection of the data showed some missing values,most of the missing data for variables occurred in population, Hepatitis B and GDP while other missing data were from less known countries. In order to decrease the error in further training of data, we try to exclude the missing statistics.

Hide

```
Life_Expectancy_World <- read.csv(file="C:/Cheng Ye/UCSB/PSTAT 231/Final Project/Life Expe
ctancy Data.csv")%>% na.omit()
# Life_Expectancy_World
dim(Life_Expectancy_World)
```

```
## [1] 1649   22
```

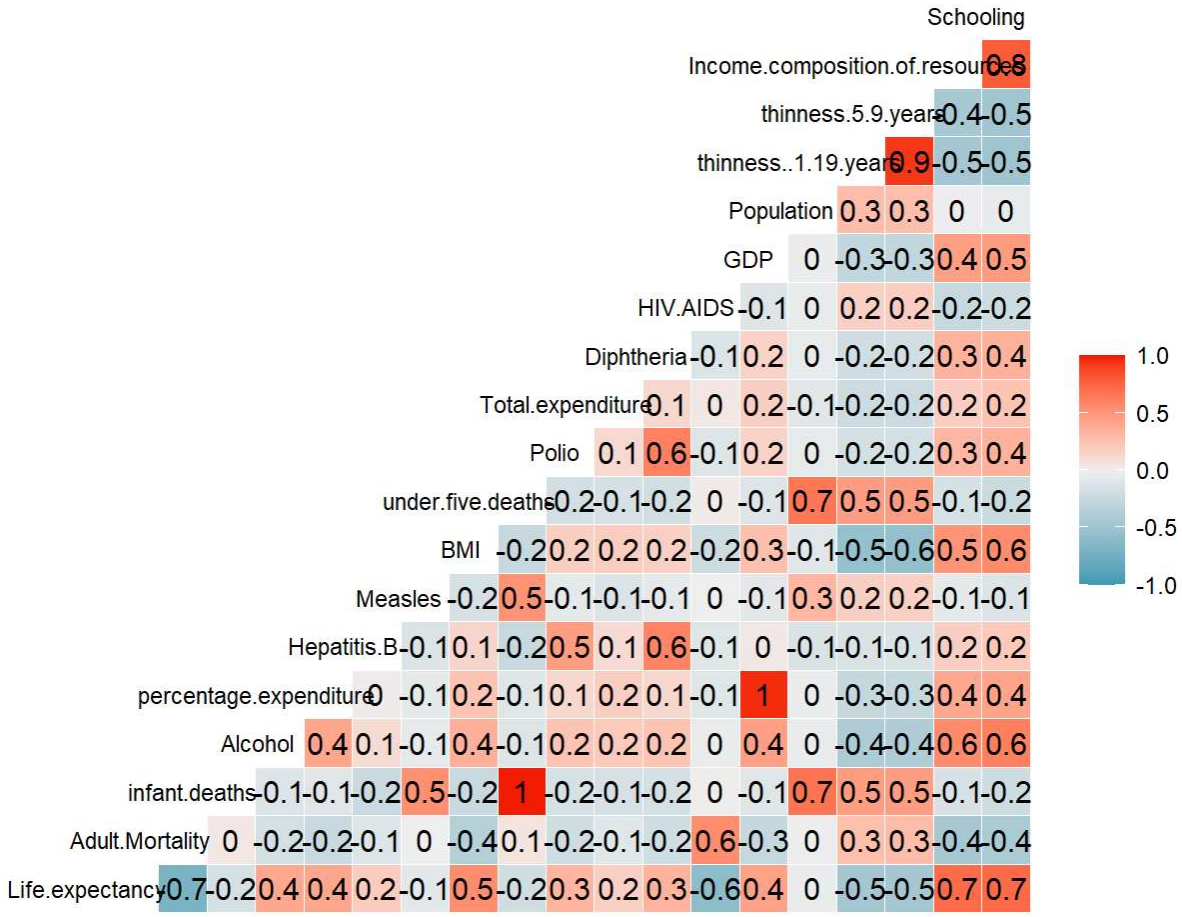
After excluding missing values we could observe that there are 1649 observations with 22 variables. ### Understanding our variables Now let's take a look of our integrated dataset, we could conclude the variables in the following categories: 1. Country: The name of our given country waiting to be analyzed 2. Timeframe: Our dataset contains time values of certain countries from 2000 to 2015 3. Status: This indicates whether our given country is developed or developing, which is crucial as one of the hypothesis in this project before all modeling is that "life expectancy is correlated with the development status of the given country". 4. Death related figures:Adult Mortality, Infant Deaths, Under-five deaths where it takes into account of the people died (out of per 1000 people within the same category). 5. Habits related figures:Alcohol, Thinness 1~19yrs, Thinness 5~9yrs, and BMI. "Alcohol" is the measurement of the amount of average alcohol intake of population within that year while Thinness X~XX yrs represent the percentage of thinness presented in the given age range. BMI is average body fitness percentage of population of that given country 6. Population: This part is self explanatory. It is the population of the given country 7. Broad Economic factors:GDP, Total expenditure, Income composition of resources. Such three figures represent the allocation of resources(unit is billion dollars) 8. Immunization coverage related factors:Hepatitis B, Polio, Diphtheria. Again, very self-explanatory. The immunization coverage percentage of such disease within their respective population 9. Disease related factors: HIV/AIDS/Measles. The percentage of people infected with such disease within the given population 10. Education related figures: Schooling. The percentage of people being able to go to school within the given population

Visualizing Variables

After we've taken a moment to know what each variable means, we should begin visualizing our variables. We start with life Expectancy as it is the core of our project. #### Variable Correlation Plot First, let's do a correlation heat map of the numeric variables to get a general idea of their relationship.

Hide

```
life_relation <- Life_Expectancy_World %>%
  select(-Country, -Year, -Status)
life_cor <- cor(life_relation)
ggcorr(life_relation ,
  cor_matrix = life_cor,
    label = T,
    hjust = 0.8,
    angle = 0,
    size = 3,
    layout.exp = 2
) #Excluding irrelevant variables such as Year and Country and or Developed/Devloping stat
us as category variables and timeframe won't do us any good in this part.
```



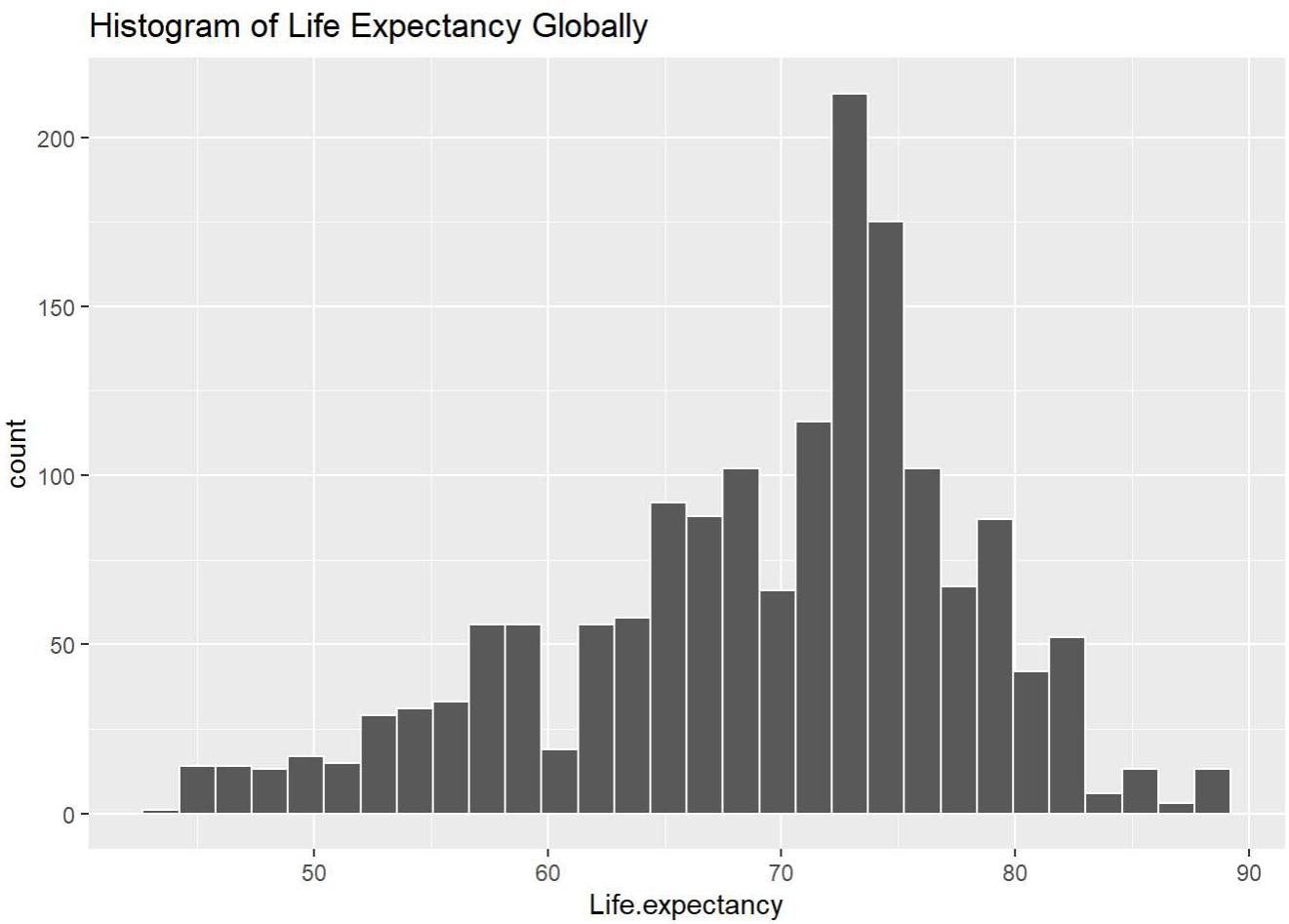
At first glance, it is surprising that certain predictors have great correlation with others while certain predictors have little to none correlation.But after further analysis between each variable, it makes more sense. Like how Infant deaths is correlated to under five deaths, as if one dies when he/she is an infant, it also qualifies under five death by default.

Life Expectancy

Hide

```
ggplot(Life_Expectancy_World, aes(`Life.expectancy`)) +
  geom_histogram(color = "white") +
  labs(title = "Histogram of Life Expectancy Globally")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



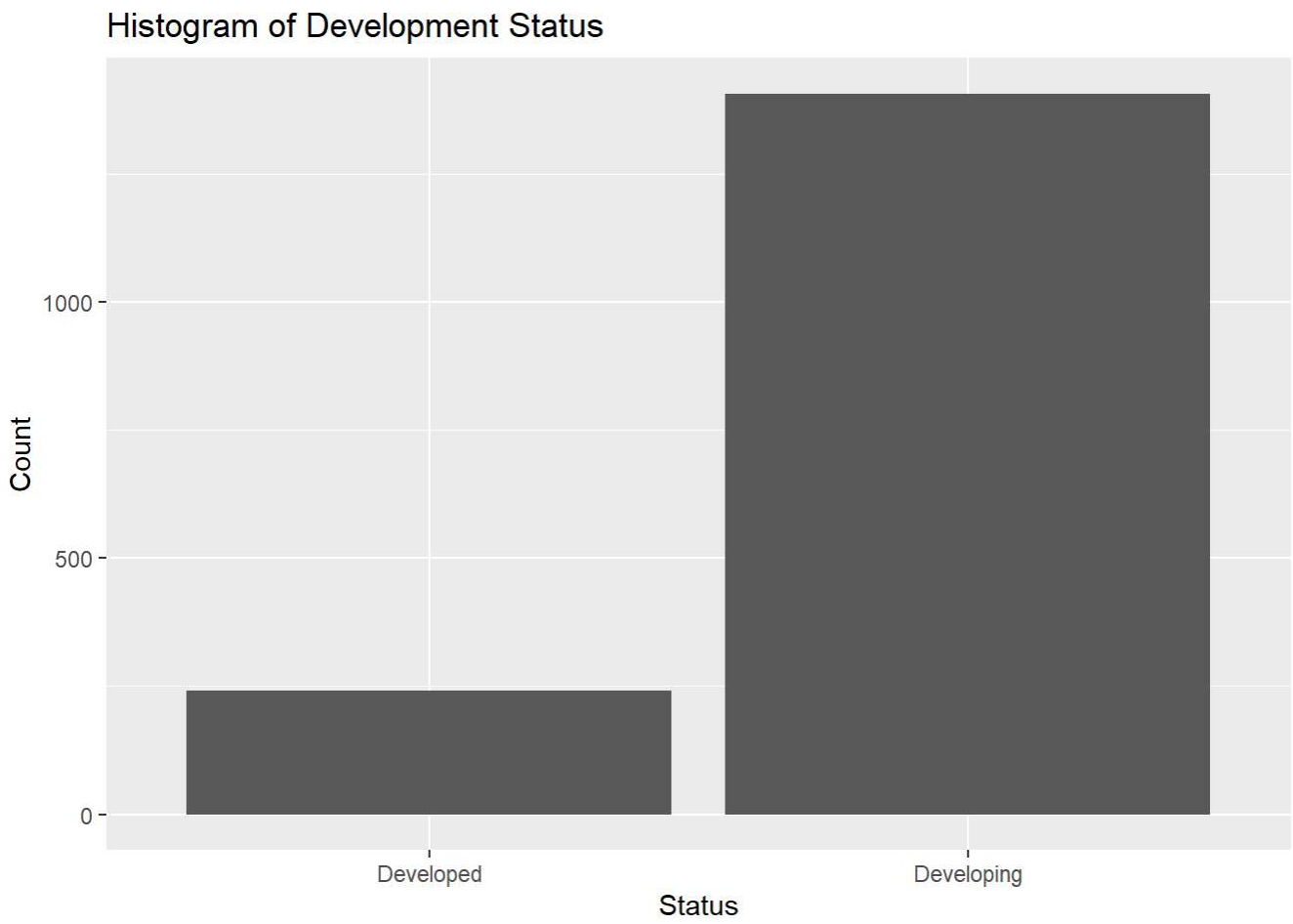
From the histogram of life expectancy we could observe that it is rather right-leaned which indicates that most countries have their average life expectancy at around 70-75 years

Development Status

Does development status affect the life expectancy of given country? Let's find out!

Hide

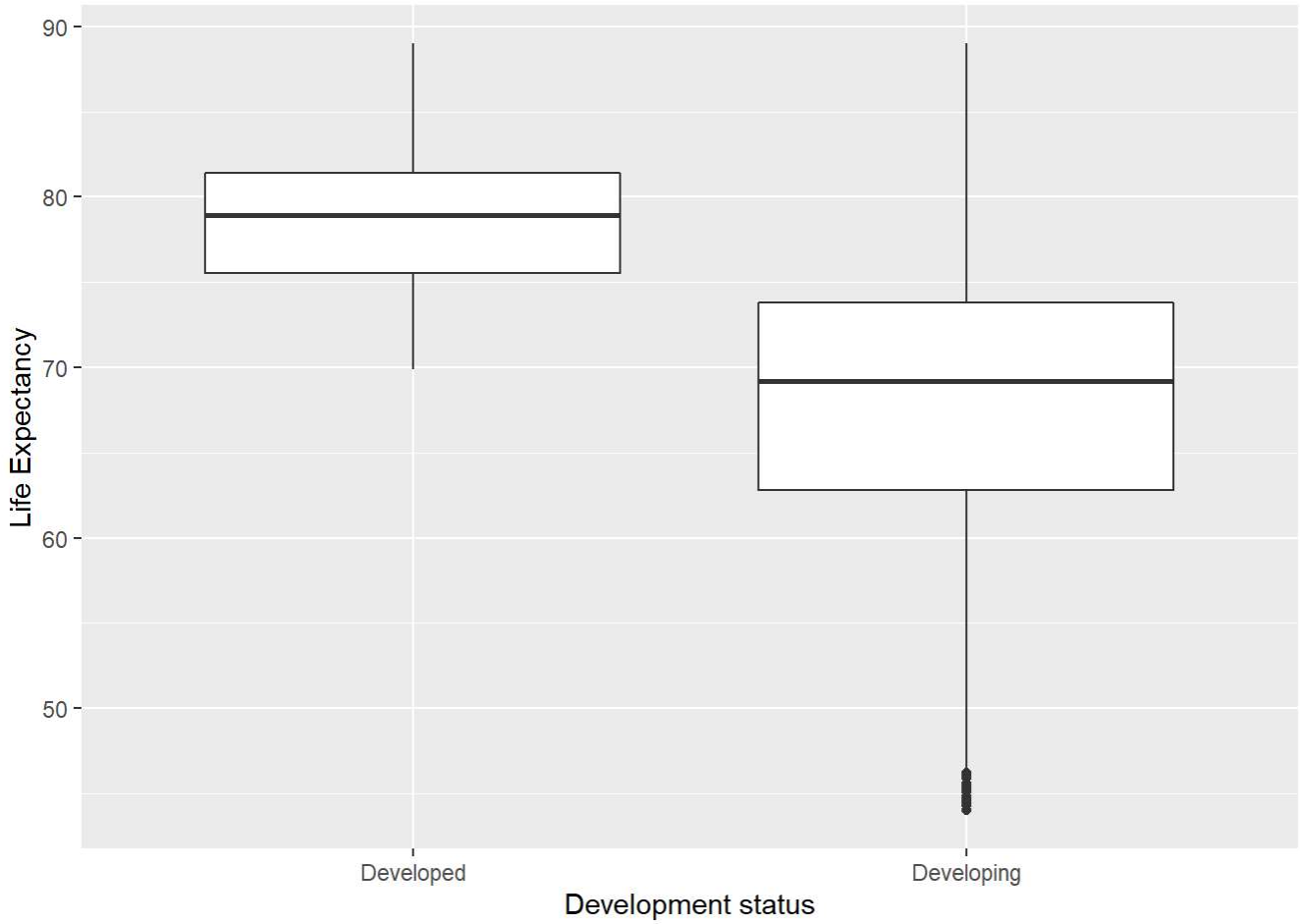
```
status <- Life_Expectancy_World %>%
    group_by(Status) %>%
    summarise(Count = n())
ggplot(data = status, aes(x = Status,y=Count)) +
  geom_histogram(bins = 30, stat = "identity") +
  labs(title="Histogram of Development Status")
```



From the histogram of development status, we could observe that the majority of countries in the dataset are developing countries, but is there a relationship between life expectancy and development status of countries? It may not be obivous as only viewing the histogram of development status doesn't tell us anything. We might be able to deduce a result through drawing a boxplot to see whether developed countries have higher boundaries as well as higher average of life expectancy than that of developing countries.

Hide

```
boxplot <- ggplot(Life_Expectancy_World, aes(x=Status, y= Life.expectancy)) + geom_boxplot
() + xlab("Development status") + ylab("Life Expectancy")
boxplot
```



From this boxplot, we could visualize the following results: Developed countries usually have higher boundaries of life expectancy than that of developing countries. Life expectancy data is less scattered in developed countries than that of developing countries, although this may be the effect of lesser data count in developed countries. The mean and median life expectancy of developed countries appears to be higher than that of the developing countries (based on visualization alone), From the results, we could conclude that the development status of countries have an impact on life expectancy. Generally, people in developed countries have higher life expectancy than that of people in developing countries.

Country

In this part, we will visualize how life expectancy is different among each country.

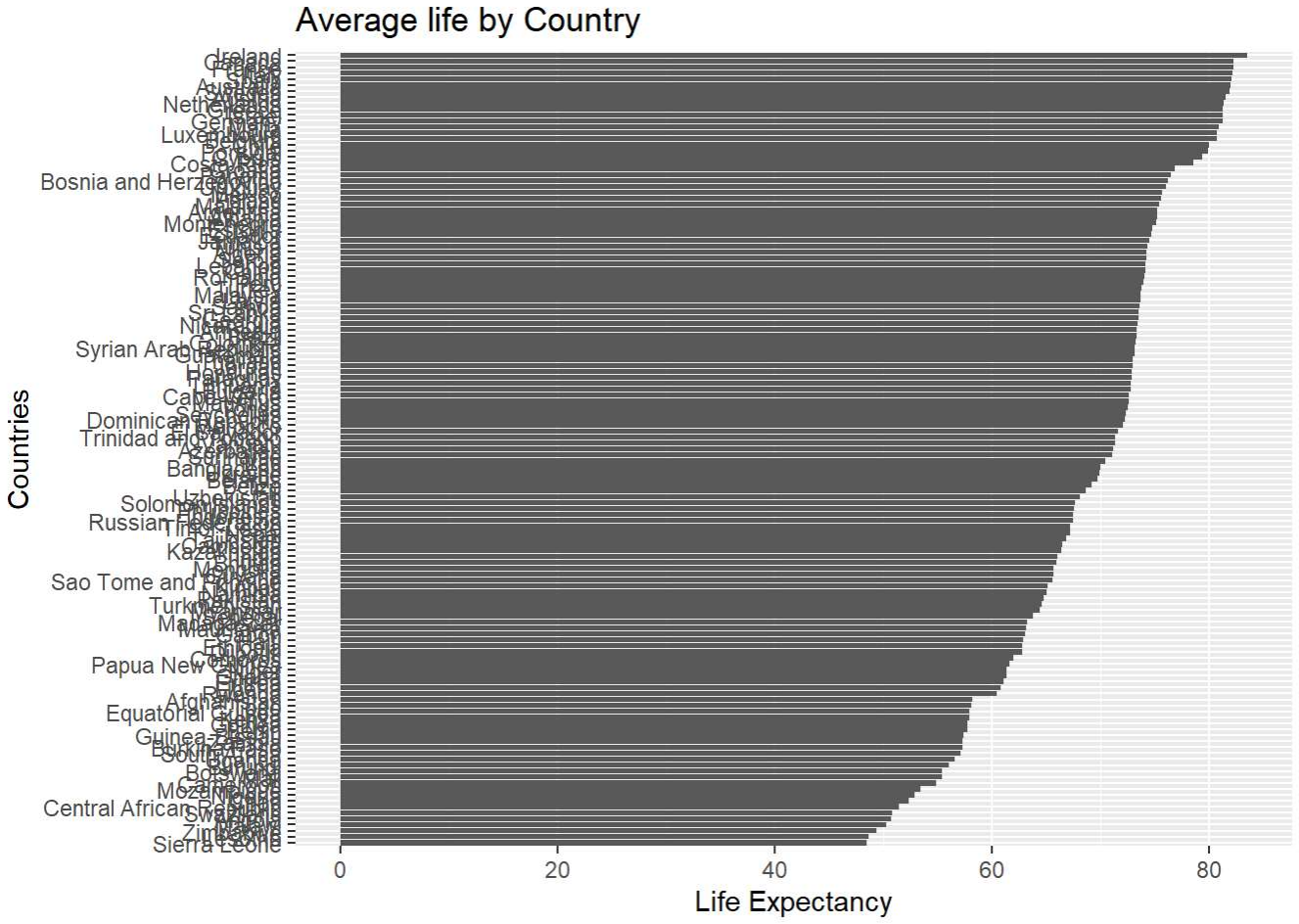
Hide

```
life_avg <-Life_Expectancy_World %>%
  group_by(Country) %>%
  summarize(mean = mean(Life.expectancy))
life_avg # Finding out the average life of citizens by country over 2000~2015
```

```
## # A tibble: 133 x 2
##    Country      mean
##    <chr>       <dbl>
##  1 Afghanistan  58.2
##  2 Albania      75.2
##  3 Algeria      74.2
##  4 Angola       50.7
##  5 Argentina    75.2
##  6 Armenia      73.3
##  7 Australia    81.9
##  8 Austria      81.5
##  9 Azerbaijan   71.1
## 10 Bangladesh   70.0
## # ... with 123 more rows
```

Hide

```
life_avg %>%
  ggplot(aes(x = mean, y = reorder(Country, mean))) +
  geom_bar(stat="identity") +
  labs(title = "Average life by Country",
       x = 'Life Expectancy',
       y = 'Countries')
```



Hide

```
life_avg %>%
  arrange(desc(mean)) #arranging the data in descending order to have better understanding
```

```
## # A tibble: 133 x 2
##    Country      mean
##    <chr>       <dbl>
##  1 Ireland      83.4
##  2 Canada       82.2
##  3 France       82.2
##  4 Italy        82.2
##  5 Spain        82.0
##  6 Australia    81.9
##  7 Sweden       81.9
##  8 Austria      81.5
##  9 Netherlands  81.3
## 10 Greece       81.2
## # ... with 123 more rows
```

From our data, we could observe that certain countries have high mean life expectancy than others: Countries such as Ireland, Canada, France, Italy, and Spain have significant higher life expectancy than that of Zimbabwe, Lesotho, and Sierra Leone.

Relationship between Life expectancy and other variables

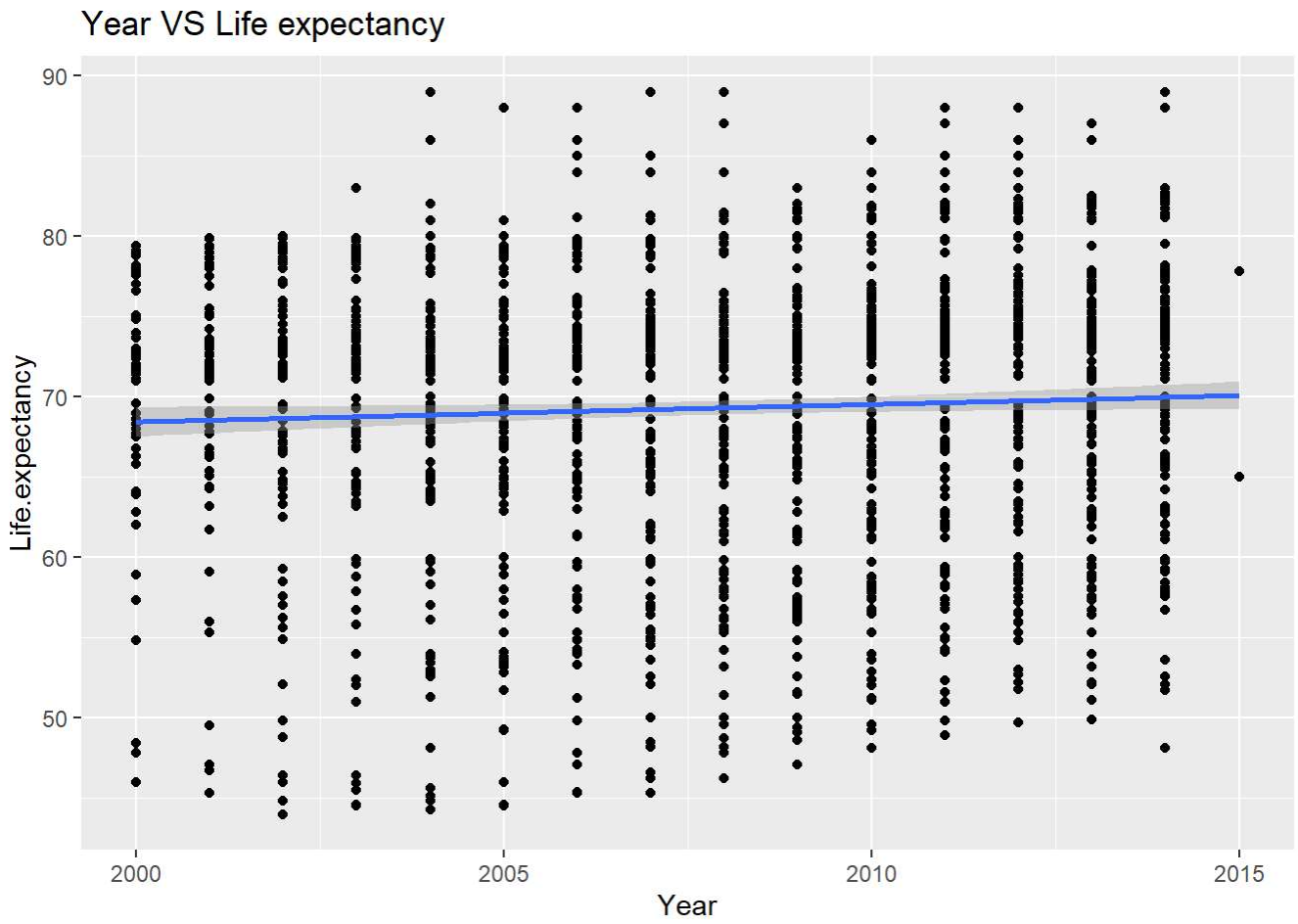
In this sub-section we will explore the differences between life expectancy and variables such as timeframe, death related figures, habits related figures, population, broad economic factors, immunization coverage factors, education related factors, and disease spread factors. Although we have drawn correlation maps in the previous section, it should be noted that other methods to visualize correlation of variables should also be used to strengthen the credibility of results. Because we are focused on life expectancy then we need to examine correlation of it with other components. We will generally use scatter plots to illustrate the correlation between each component with life expectancy, as using scatter plots would provide visual and statistical means to test the strength of a relationship between two variables. However, scatter plots could only show correlation, correlation is not equivalent to causation, hence scatterplots cannot provide the precise extent of association.We cannot use Scatter diagrams to show the relation of more than two variables. As intersectionality exist in real life cases, what we could achieve here is merely inferring the data out of scatter plots.

Timeframe

Hide

```
ggplot(Life_Expectancy_World, aes(x=Year, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Year VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

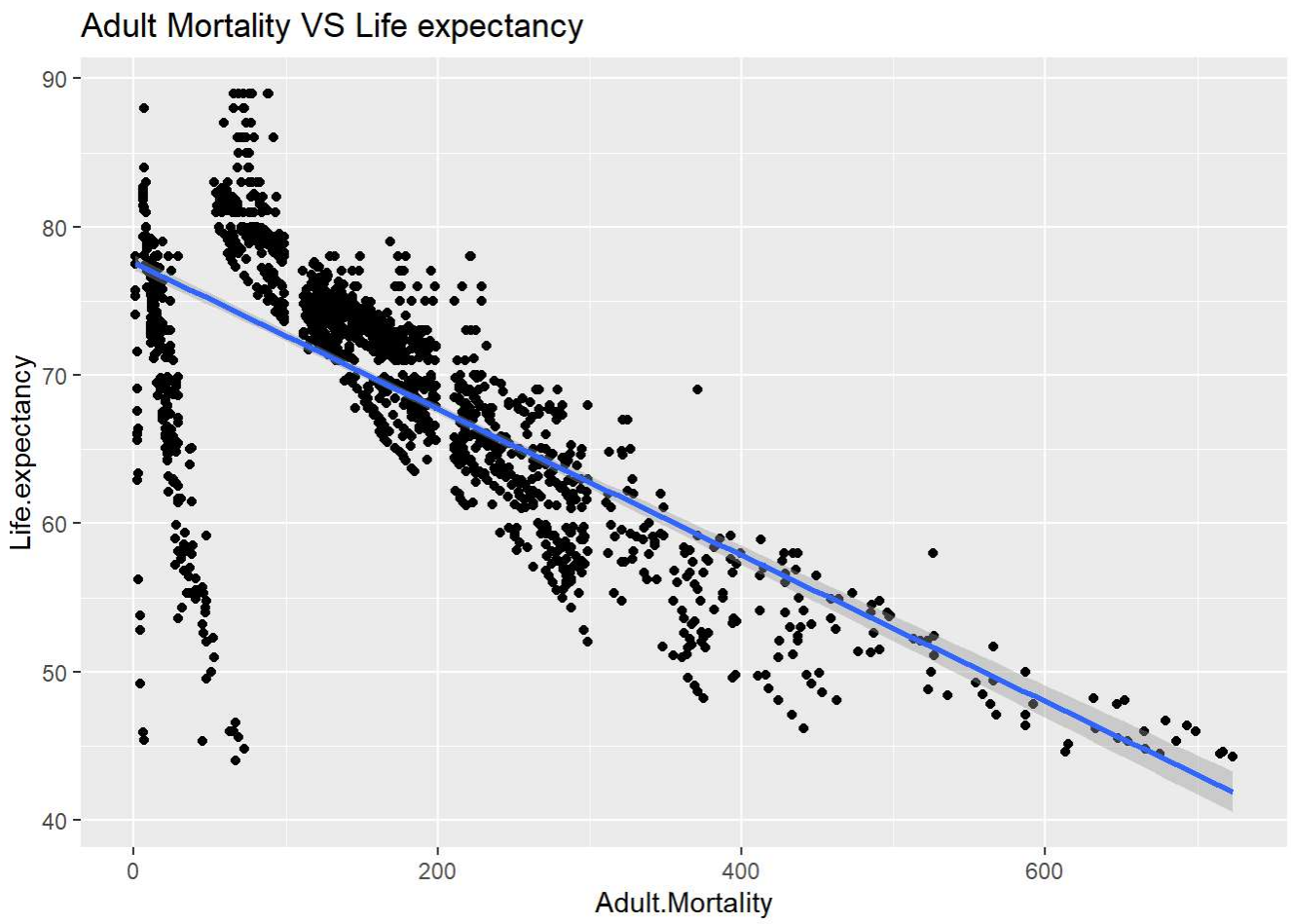


Here, we could observe that year and life expectancy have a slight relationship. Although as the years increases, life expectancy tends to “increase”, we are unable to determine it being strongly correlated. #### Death Related Figures ##### Adult Mortality

Hide

```
ggplot(Life_Expectancy_World, aes(x=Adult.Mortality, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Adult Mortality VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



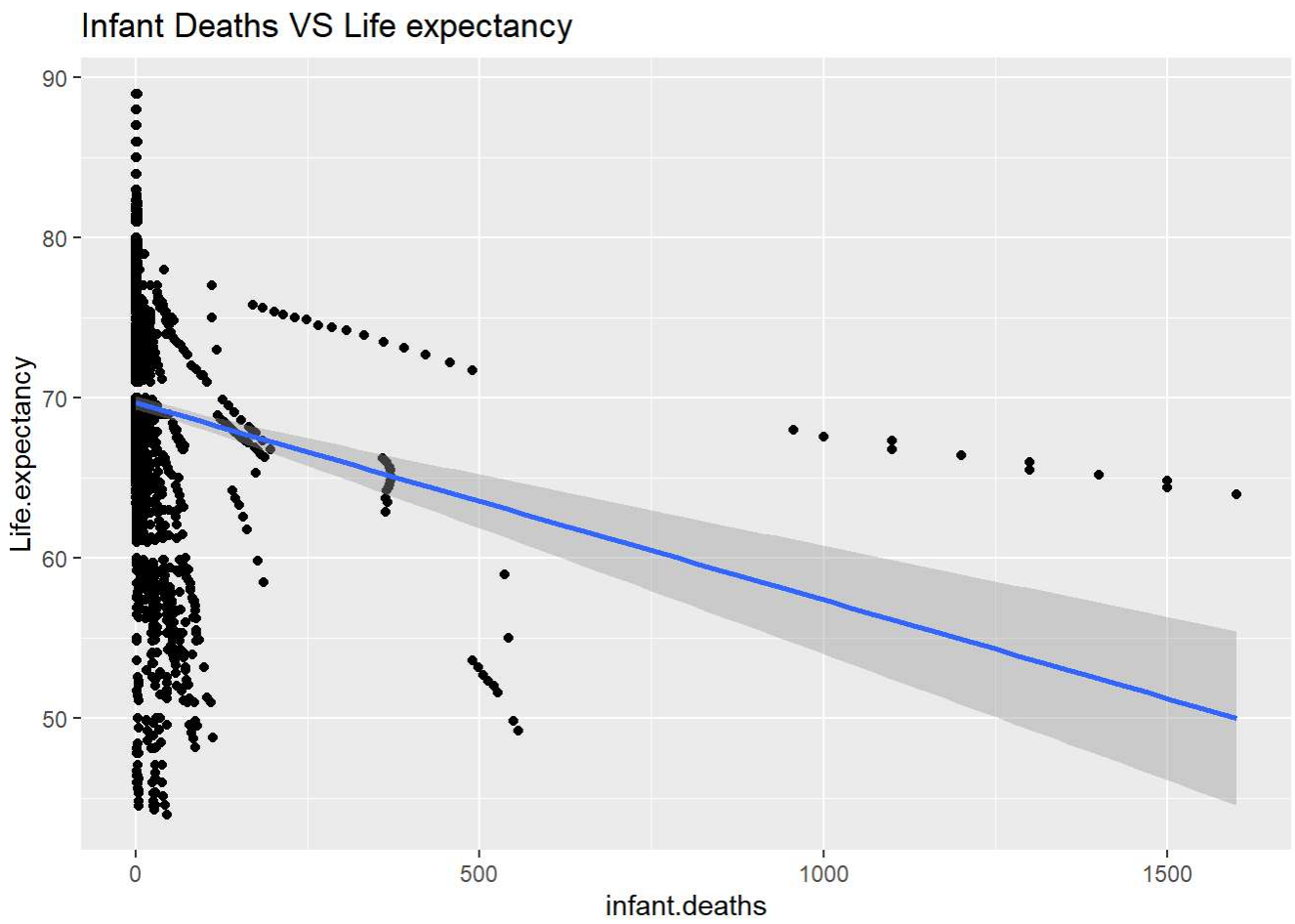
From inspecting the scatter plot of adult mortality and life expectancy we could see a strong negative relationship between it. This also fits common sense: as adult mortality increases, more people are dead hence the life expectancy of overall population would decrease.

Infant Deaths

Hide

```
ggplot(Life_Expectancy_World, aes(x=infant.deaths, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Infant Deaths VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



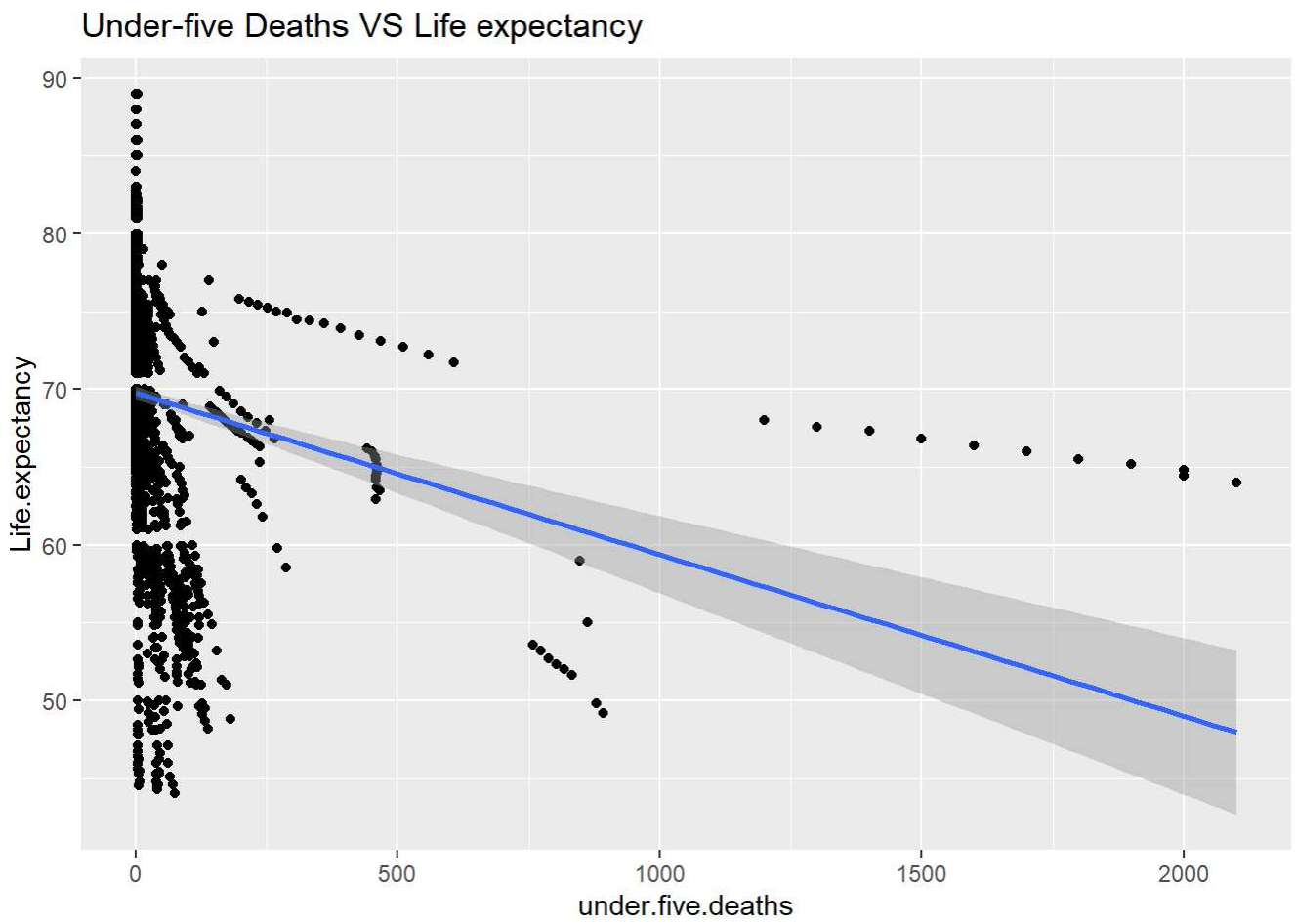
From inspecting the scatter plot of infant deaths and life expectancy we could see a strong negative relationship between it. Again, this applies to common sense. We could also observe that most of the data points are scattered around 0, meaning that most countries within the research have little to none infant deaths within population.

Under-five Deaths

Hide

```
ggplot(Life_Expectancy_World, aes(x=under.five.deaths, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Under-five Deaths VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



There is strong negative correlation between under-five deaths and life expectancy. The graphs are similar to that of infant deaths vs life expectancy. These two categories overlap in someway and is proven throughout the correlation heat map in previous sections.

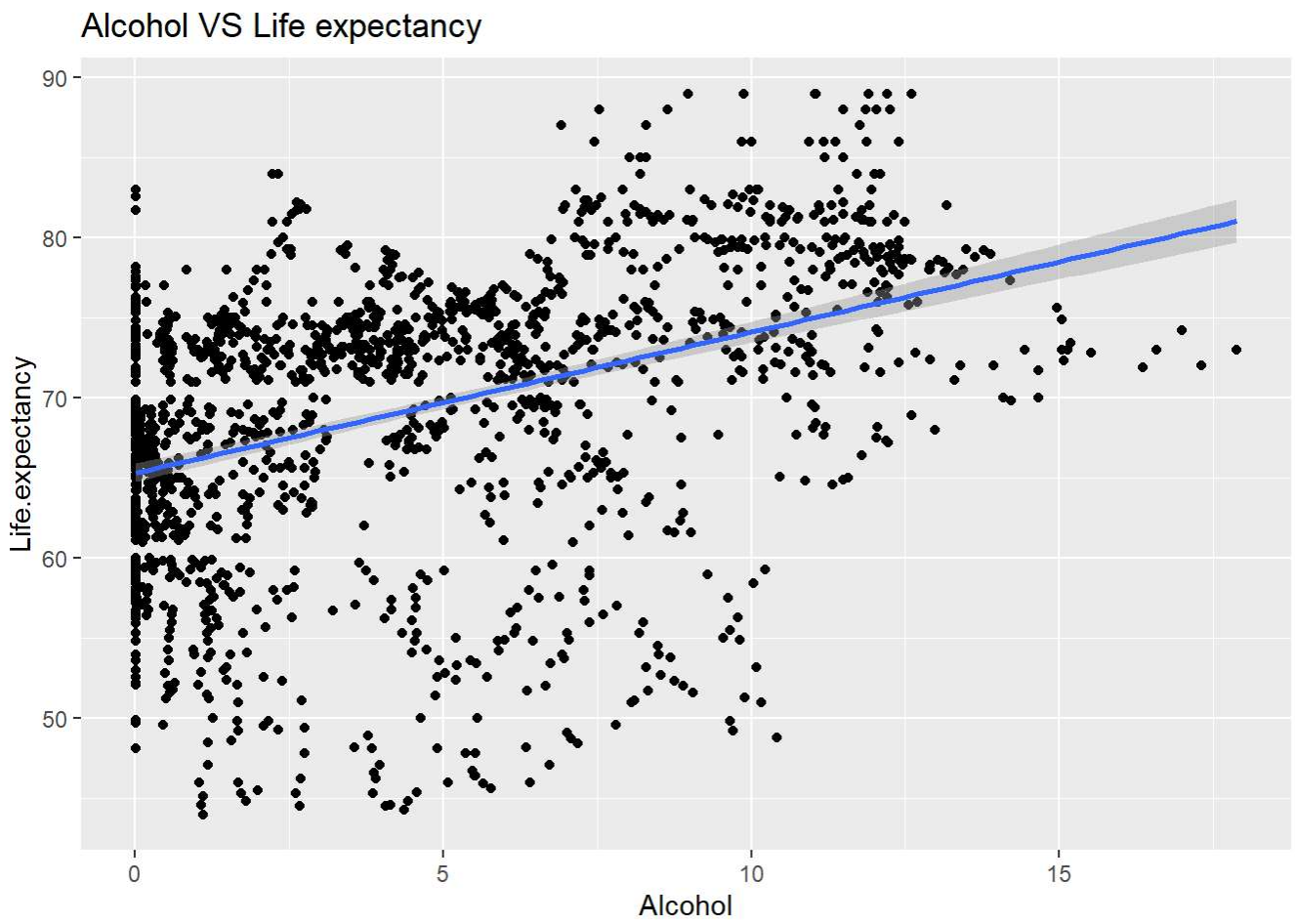
Habit Related Figures

Alcohol

Hide

```
ggplot(Life_Expectancy_World, aes(x=Alcohol, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Alcohol VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



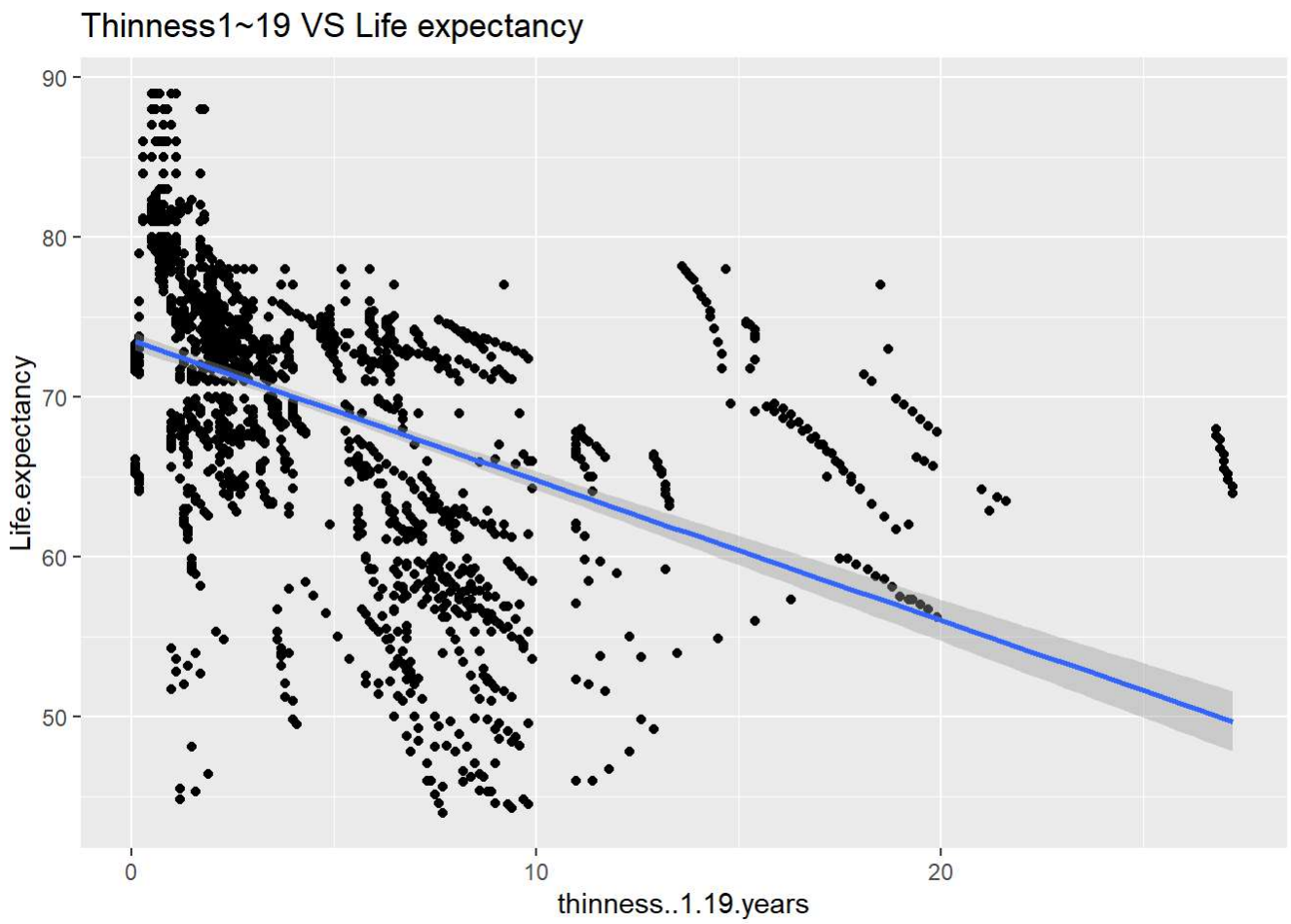
From the results show a positive relationship when it comes to alcohol and life expectancy. However, this is counter-intuitive and somewhat contradicts common sense. We hypothesize that there might be some mistakes within the original dataset that led to this weird result.

Thinness Related

Hide

```
ggplot(Life_Expectancy_World, aes(x=thinness..1.19.years, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Thinness1~19 VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Hide

```
ggplot(Life_Expectancy_World, aes(x=thinness.5.9.years, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Thinness5~9 VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

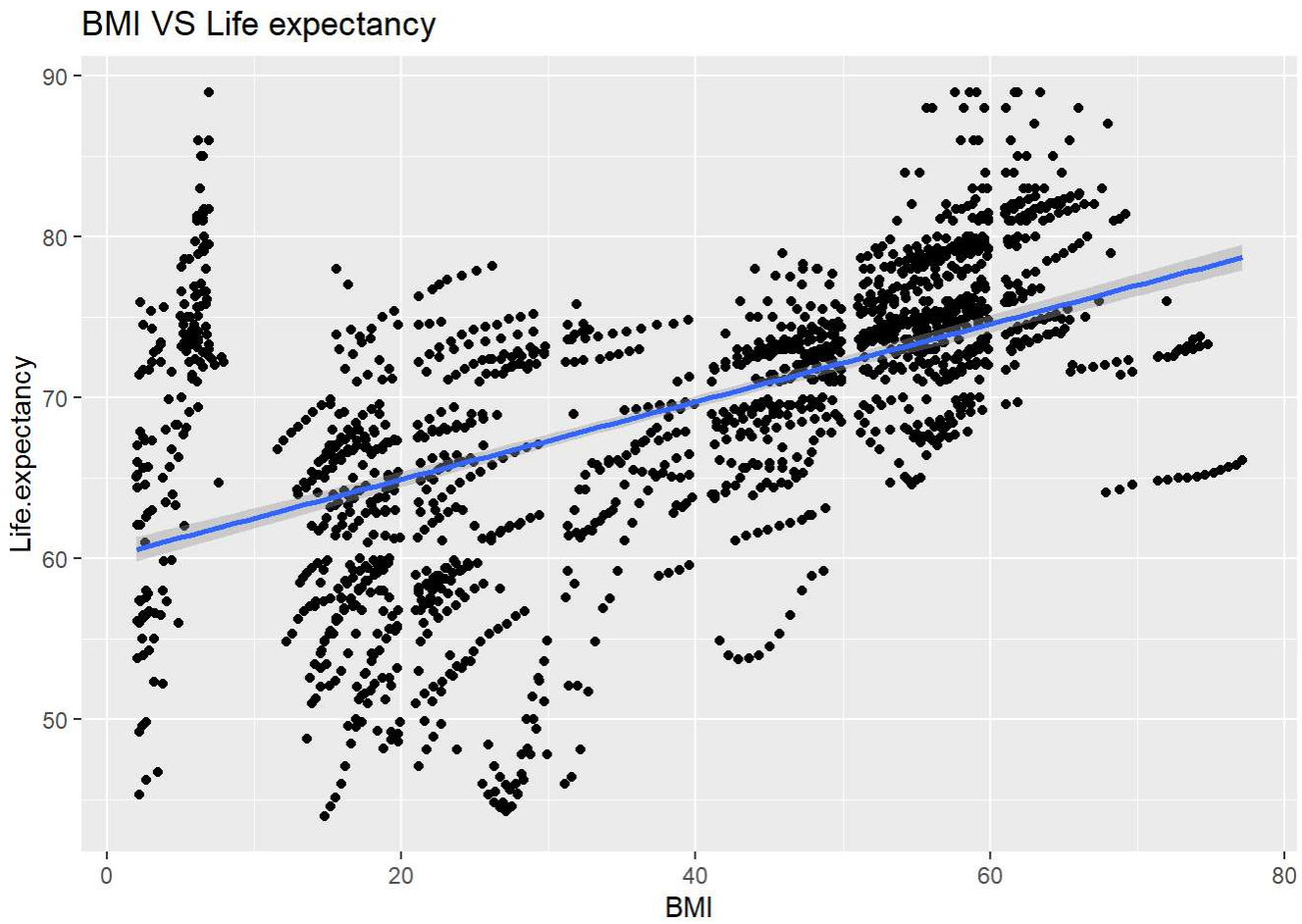
When observing both thinness of different ages vs life expectancy, we could find out that there is a strong negative relationship between them. This applies to both cases. However,thinness 5~9 appears to be a subset data of thinness 1~19

BMI

Hide

```
ggplot(Life_Expectancy_World, aes(x=BMI, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "BMI VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Here, we could observe a positive relationship between BMI and Life expectancy. As the fitness percentage of population increases, life expectancy tends to increase.

Population

Hide

```
ggplot(Life_Expectancy_World, aes(x=Population, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Population VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

By inspecting the outcomes we could see that population has little to none correlation with life expectancy. This also fits the result from the heat map drawn in previous sections.

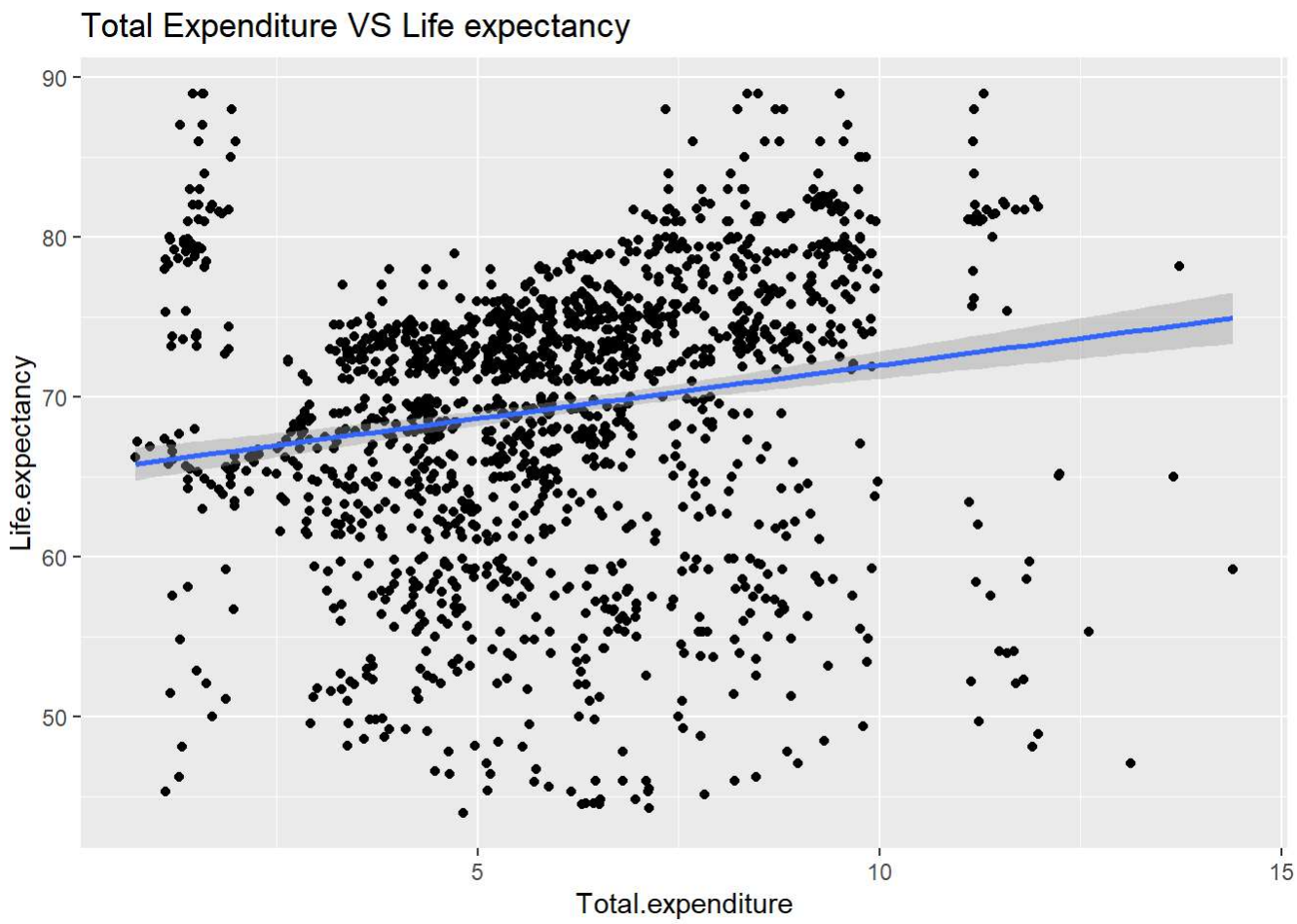
Broad Economic Factors

Total Expenditure

Hide

```
ggplot(Life_Expectancy_World, aes(x=Total.expenditure, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Total Expenditure VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

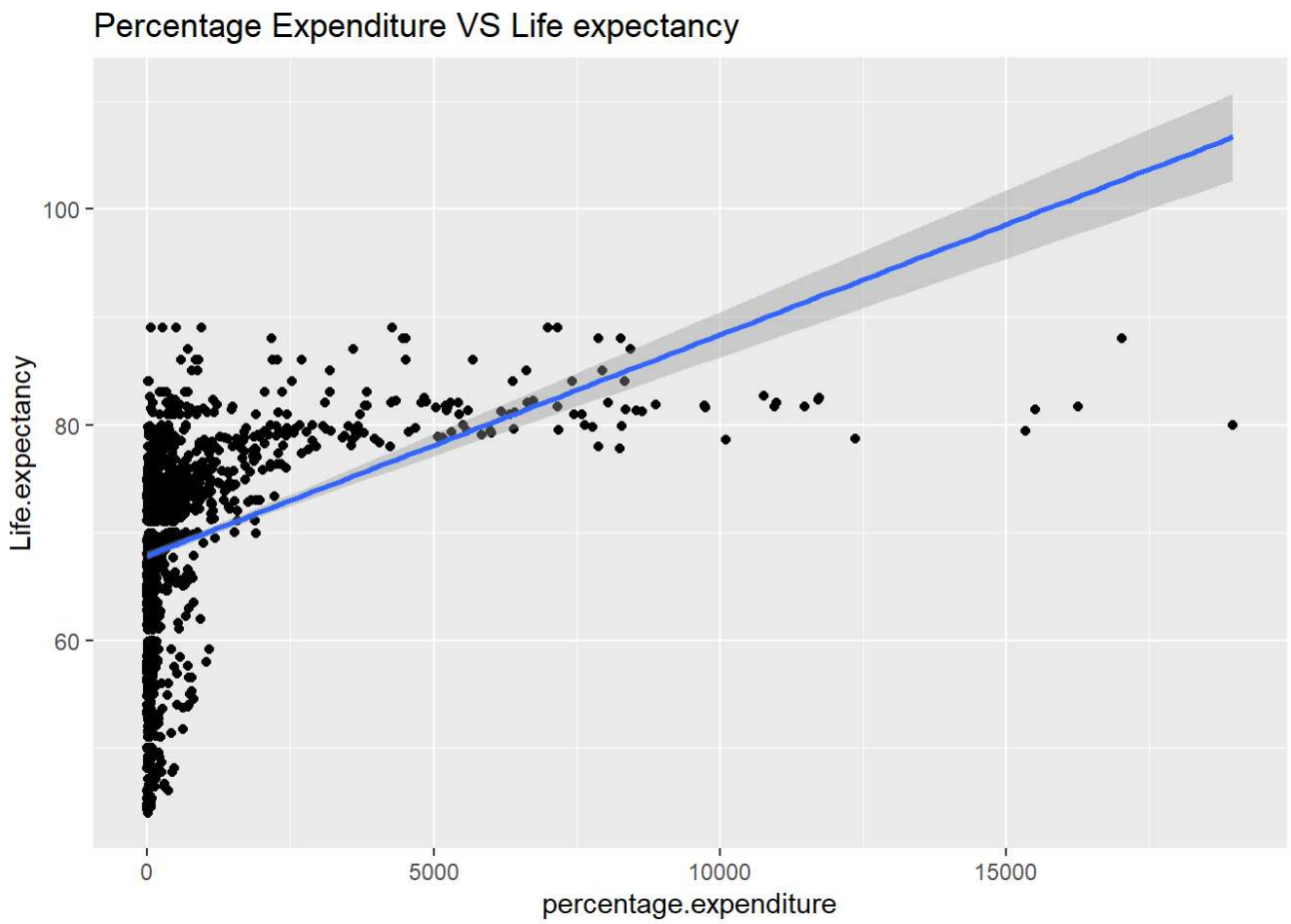


From the graph we could observe a positive relationship between total expenditure and life expectancy.
Percentage Expenditure

Hide

```
ggplot(Life_Expectancy_World, aes(x=percentage.expenditure, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Percentage Expenditure VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

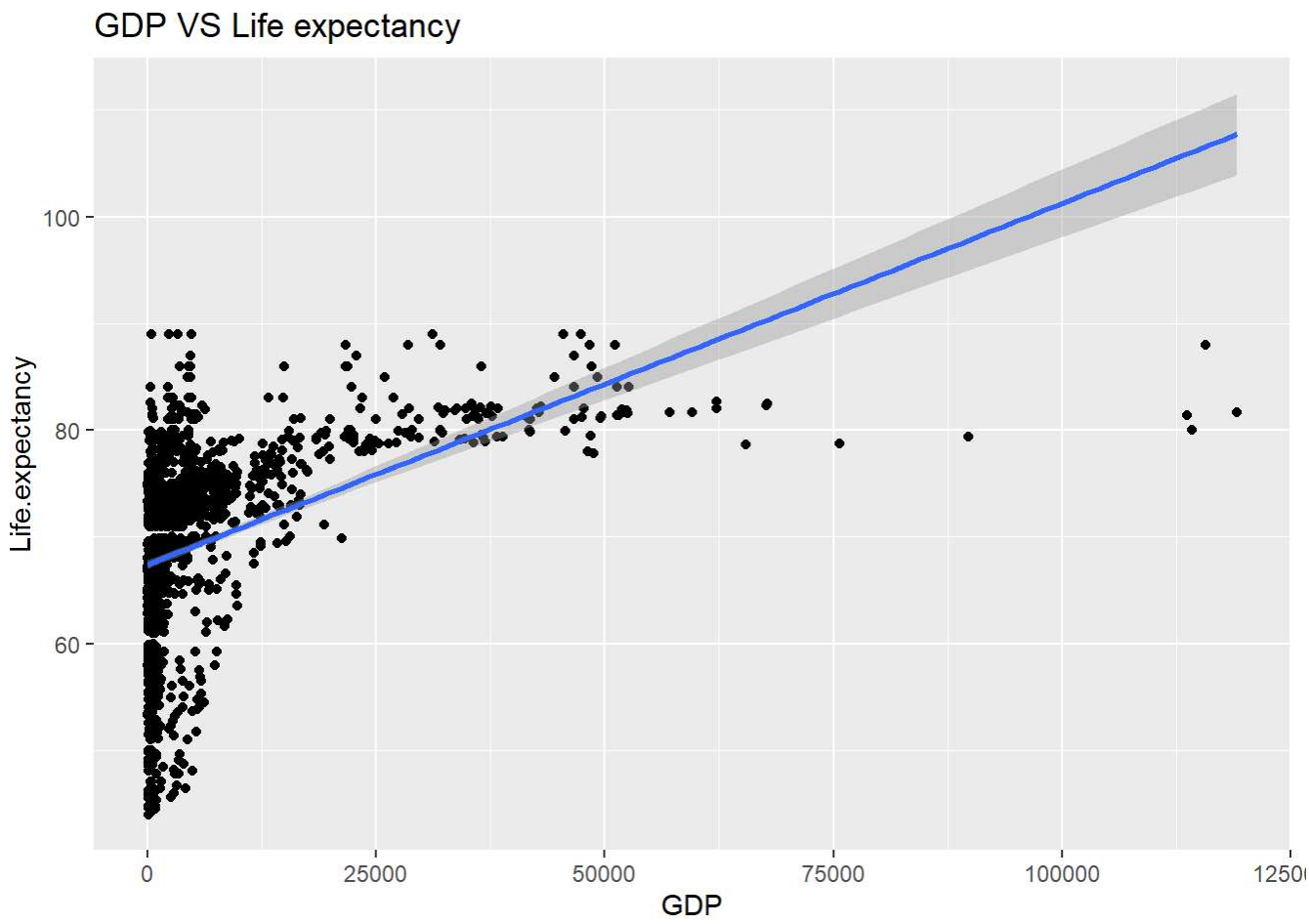


From the graph we could observe a strong positive relationship between total expenditure and life expectancy. ##### GDP

Hide

```
ggplot(Life_Expectancy_World, aes(x=GDP, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "GDP VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



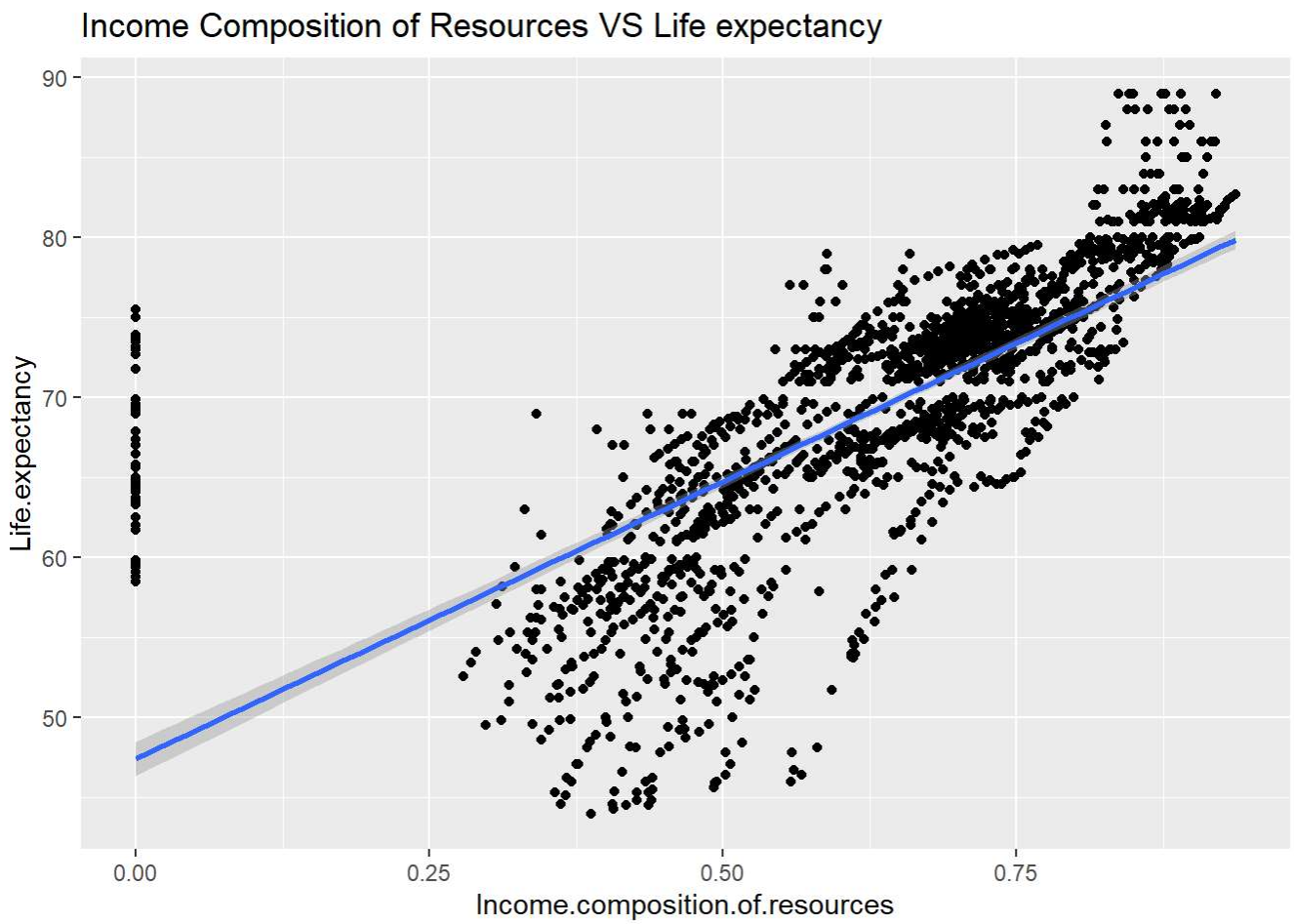
From the graph we could observe a positive relationship between GDP and life expectancy.

Income Composition of Resources

Hide

```
ggplot(Life_Expectancy_World, aes(x=Income.composition.of.resources, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Income Composition of Resources VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



This suggests that income composition of resources have very strong relationship with life expectancy. As percentages of income composition of resources increases, life expectancy increases drastically.

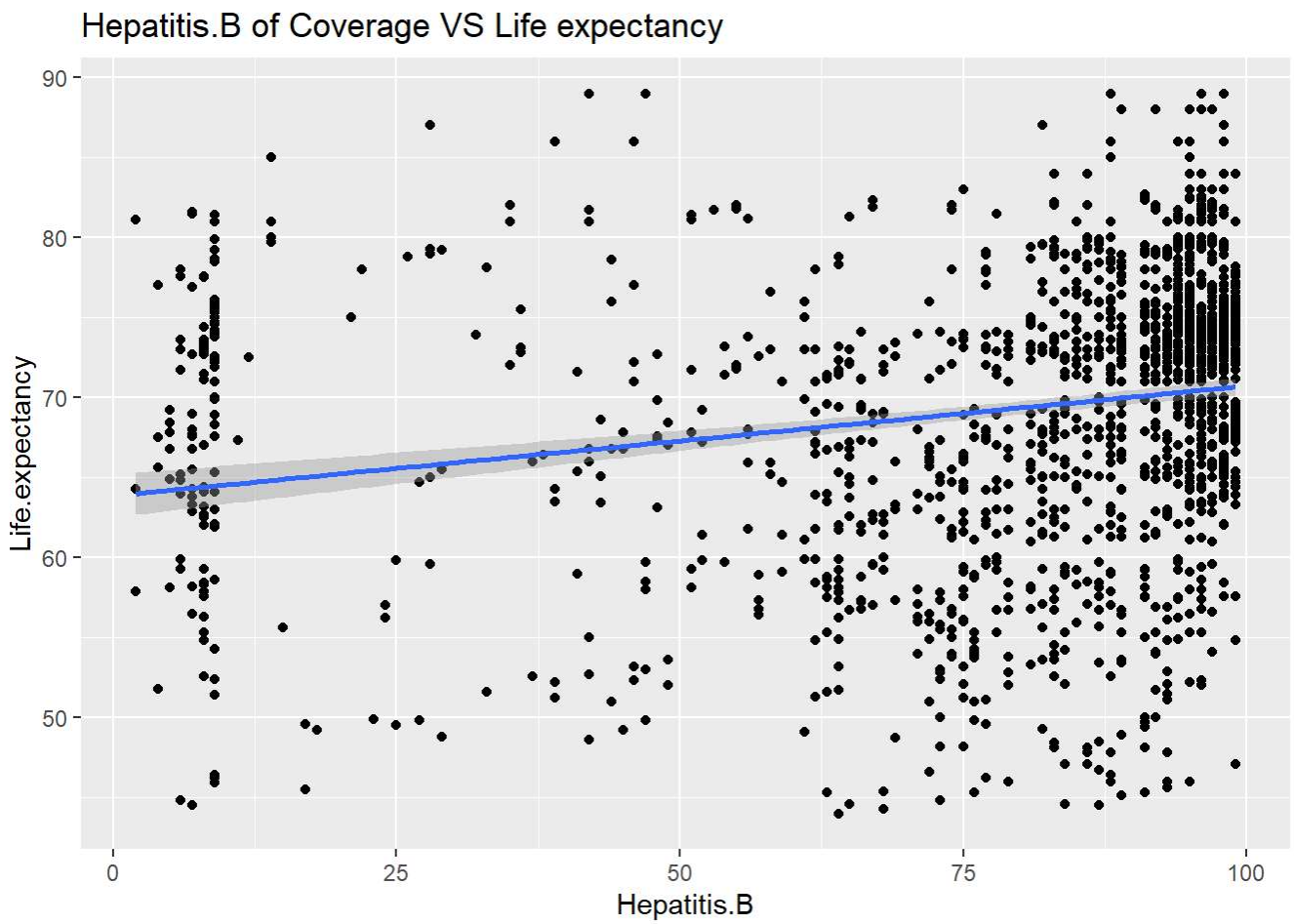
Immunization Related Figures

Hepatitis B

Hide

```
ggplot(Life_Expectancy_World, aes(x=Hepatitis.B, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Hepatitis.B of Coverage VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



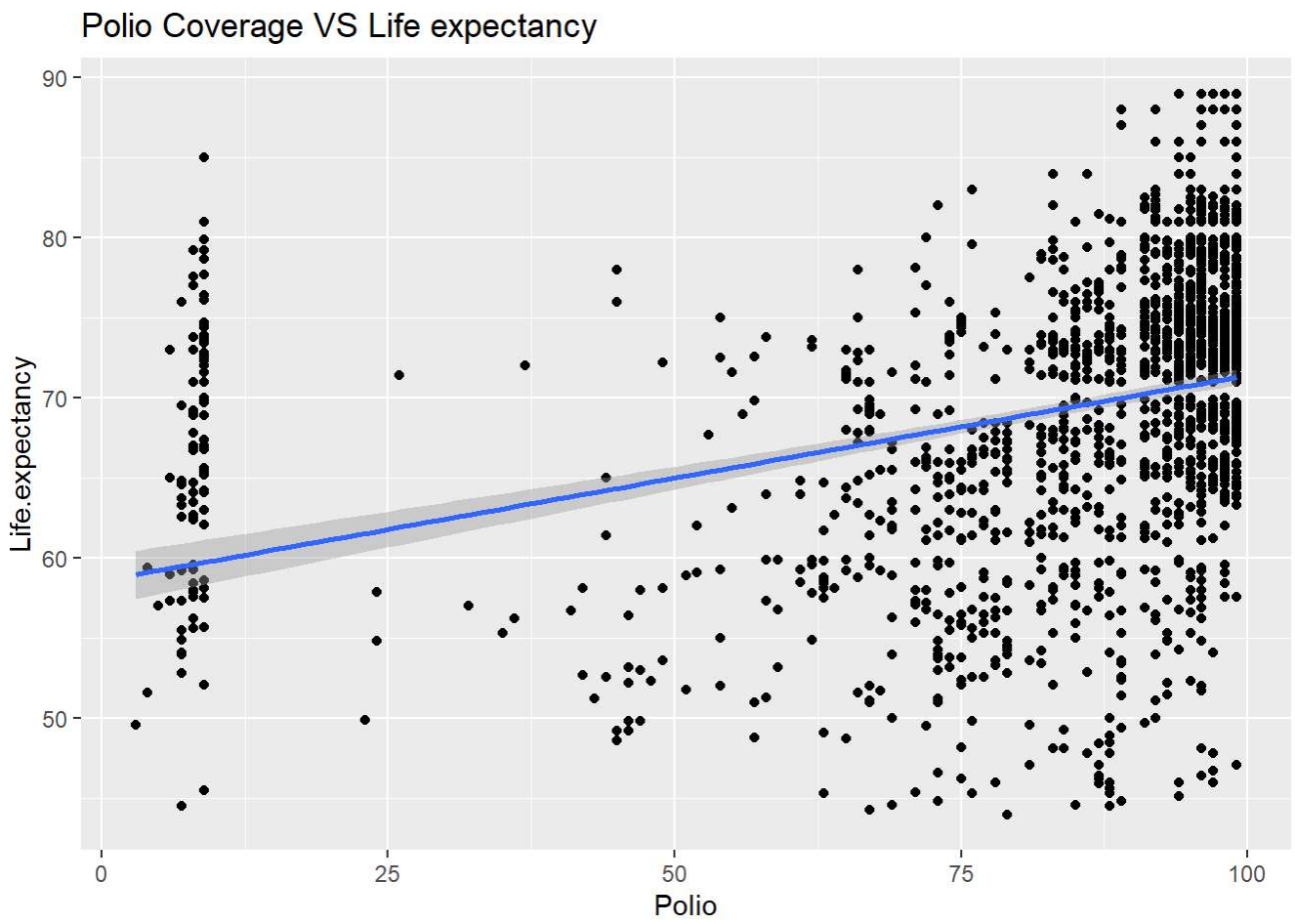
From observing the graph, we could deduce a weak positive relationship between Hepatitis B coverage and life expectancy.

Polio

Hide

```
ggplot(Life_Expectancy_World, aes(x=Polio, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Polio Coverage VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



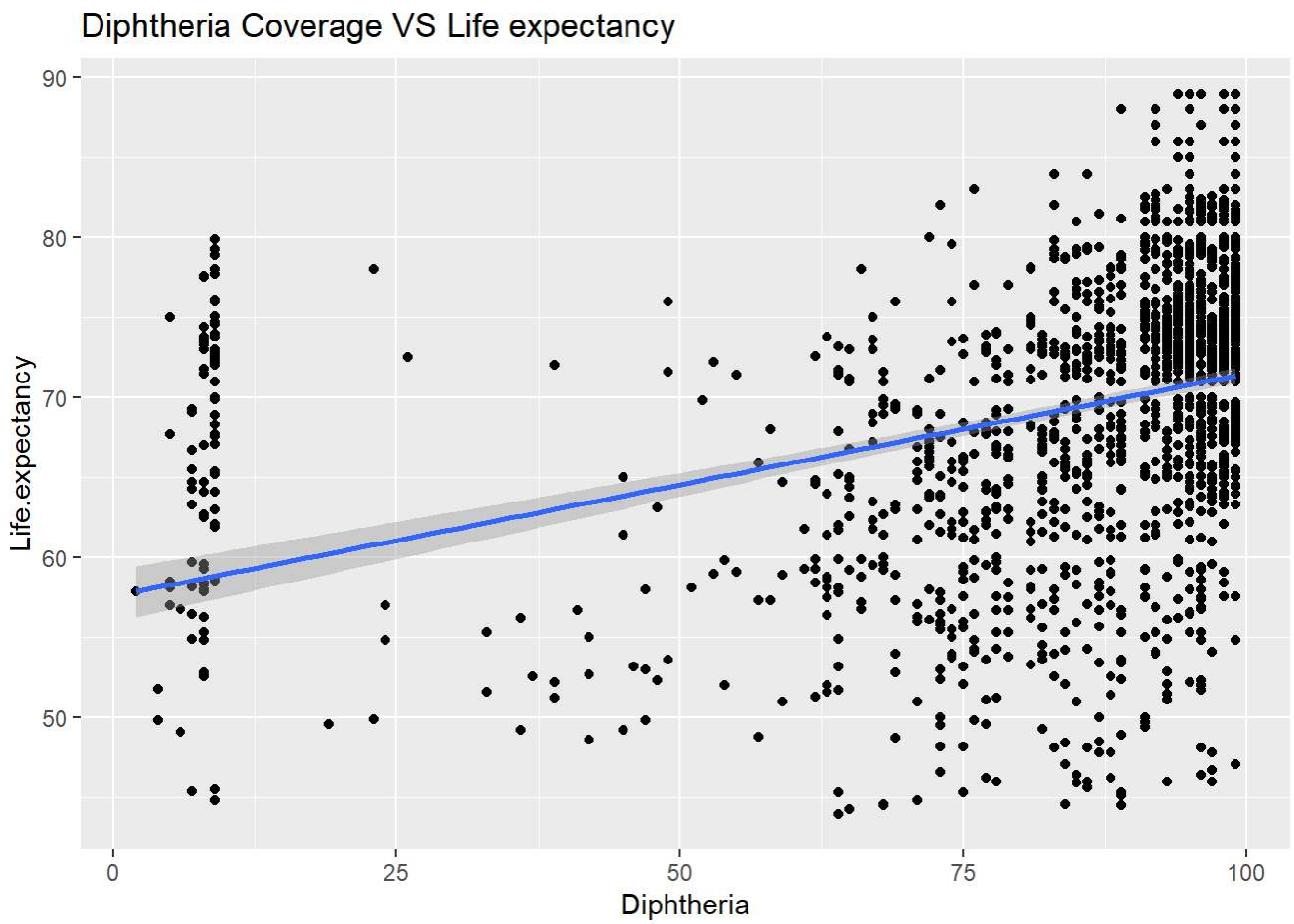
From observing the graph, we could deduce a positive relationship between Polio coverage and life expectancy.

Diptheria

Hide

```
ggplot(Life_Expectancy_World, aes(x=Diphtheria, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Diphtheria Coverage VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



From observing the graph, we could deduce a positive relationship between Diphtheria coverage and life expectancy.

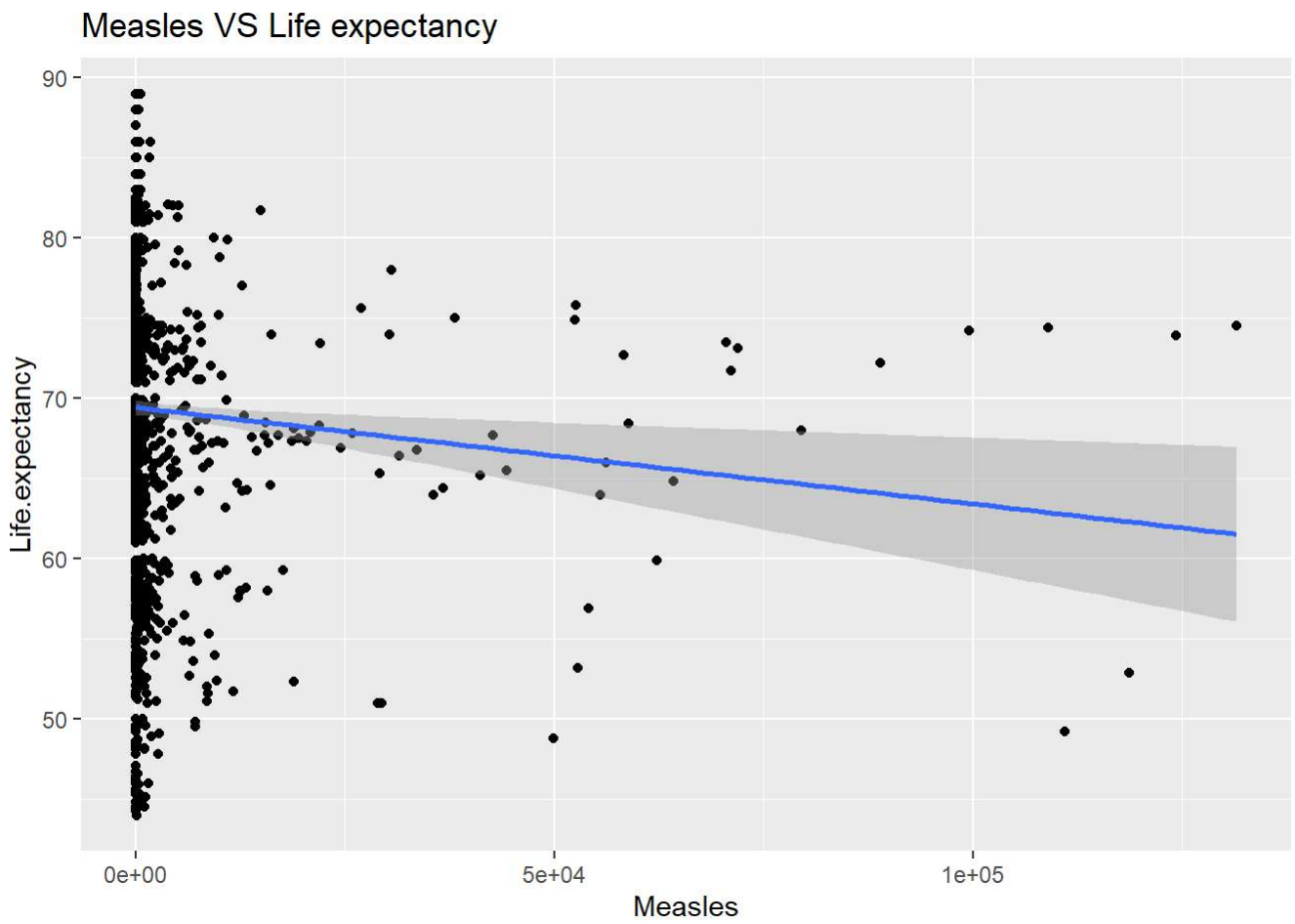
Disease Related Factors

Measles

Hide

```
ggplot(Life_Expectancy_World, aes(x=Measles, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Measles VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



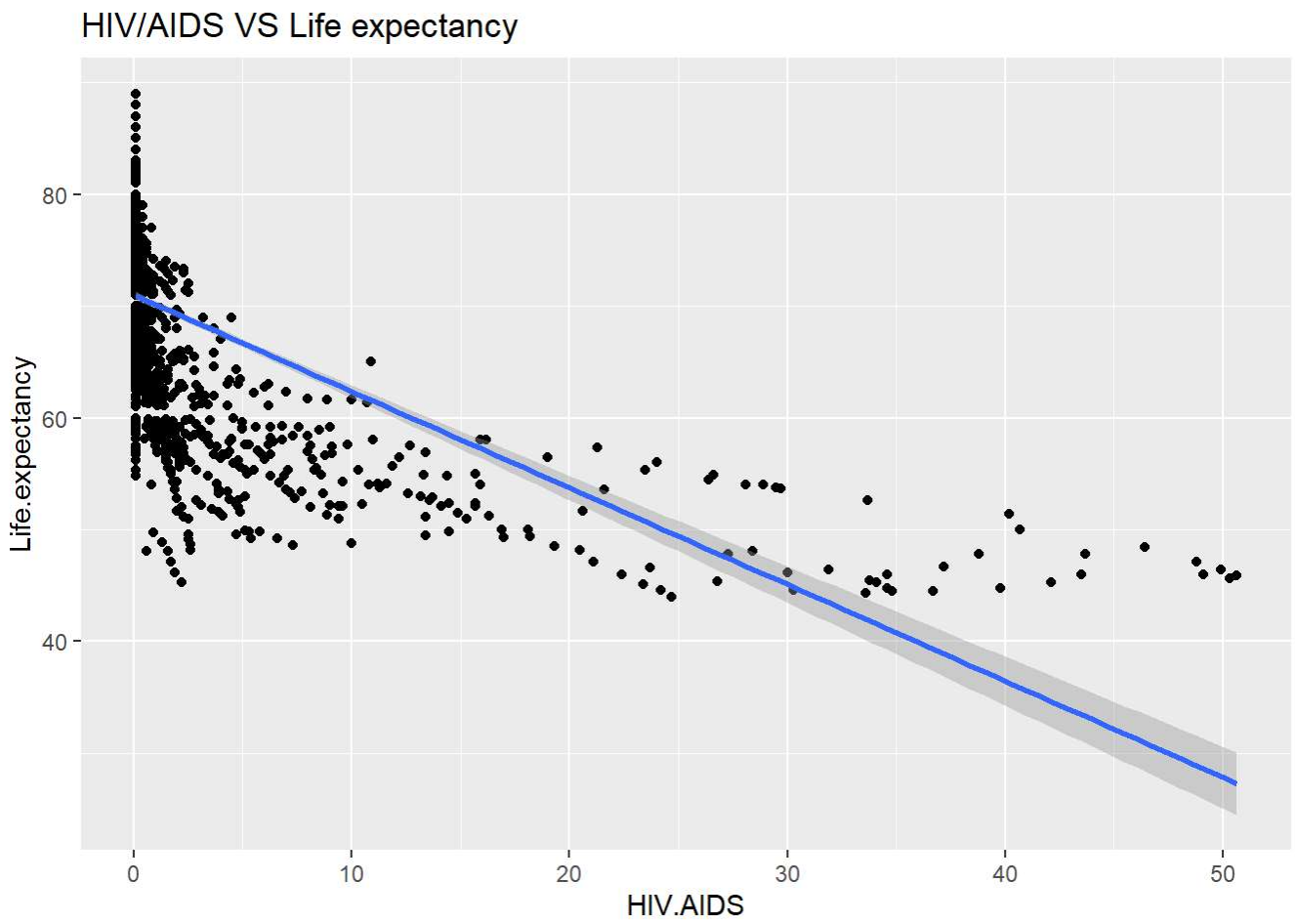
Based on inspection, measles seemed to have a negative relationship with life expectancy.

HIV/AIDS

Hide

```
ggplot(Life_Expectancy_World, aes(x=HIV.AIDS, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "HIV/AIDS VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



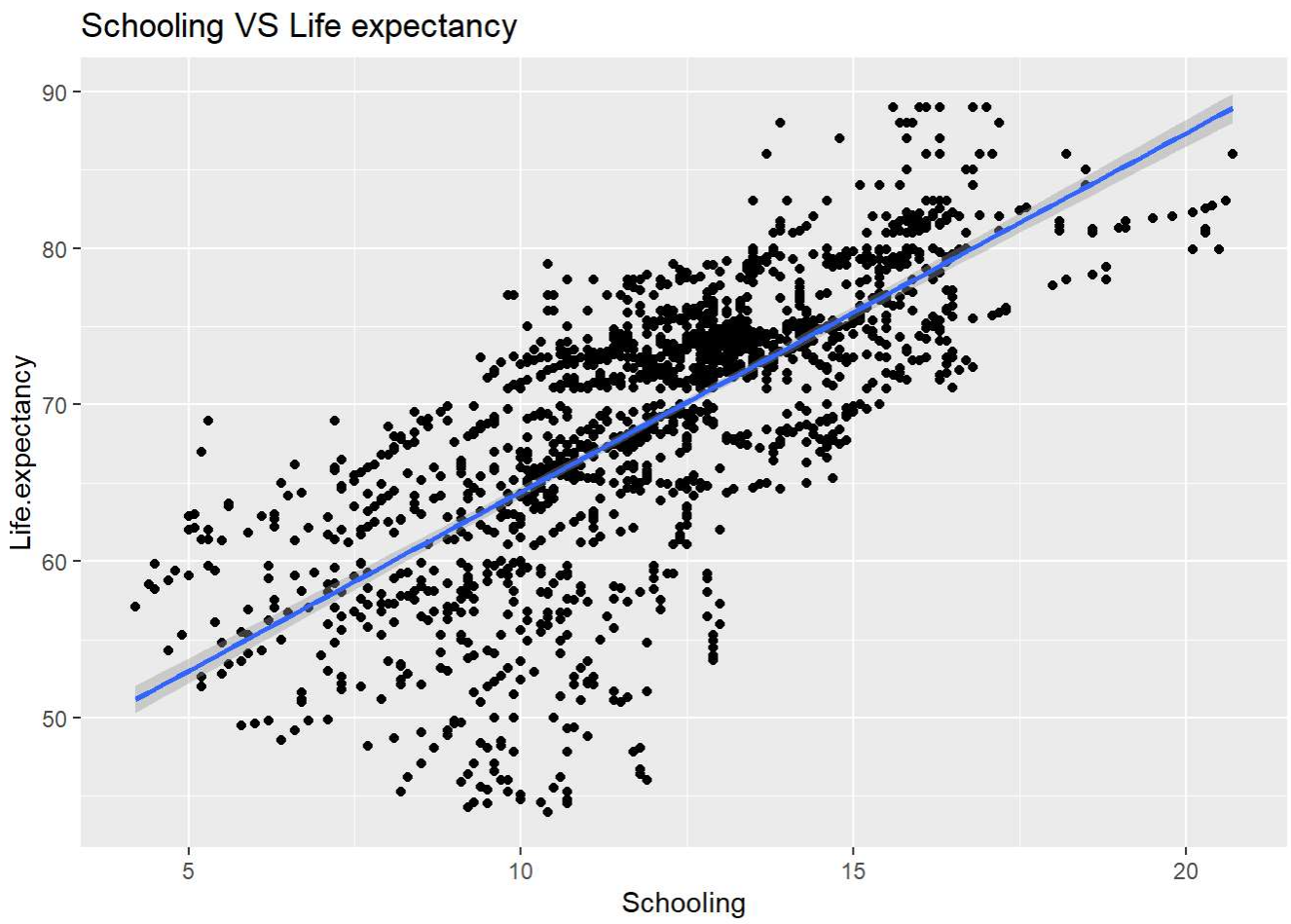
It seems that Hiv/AIDS have strong negative relationship with life expectancy, as percentage of population with AIDS increases, life expectancy decreases drastically.

Education

Hide

```
ggplot(Life_Expectancy_World, aes(x=Schooling, y=Life.expectancy)) +
  geom_point()+
  geom_smooth(method=lm)+
  labs(title = "Schooling VS Life expectancy")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



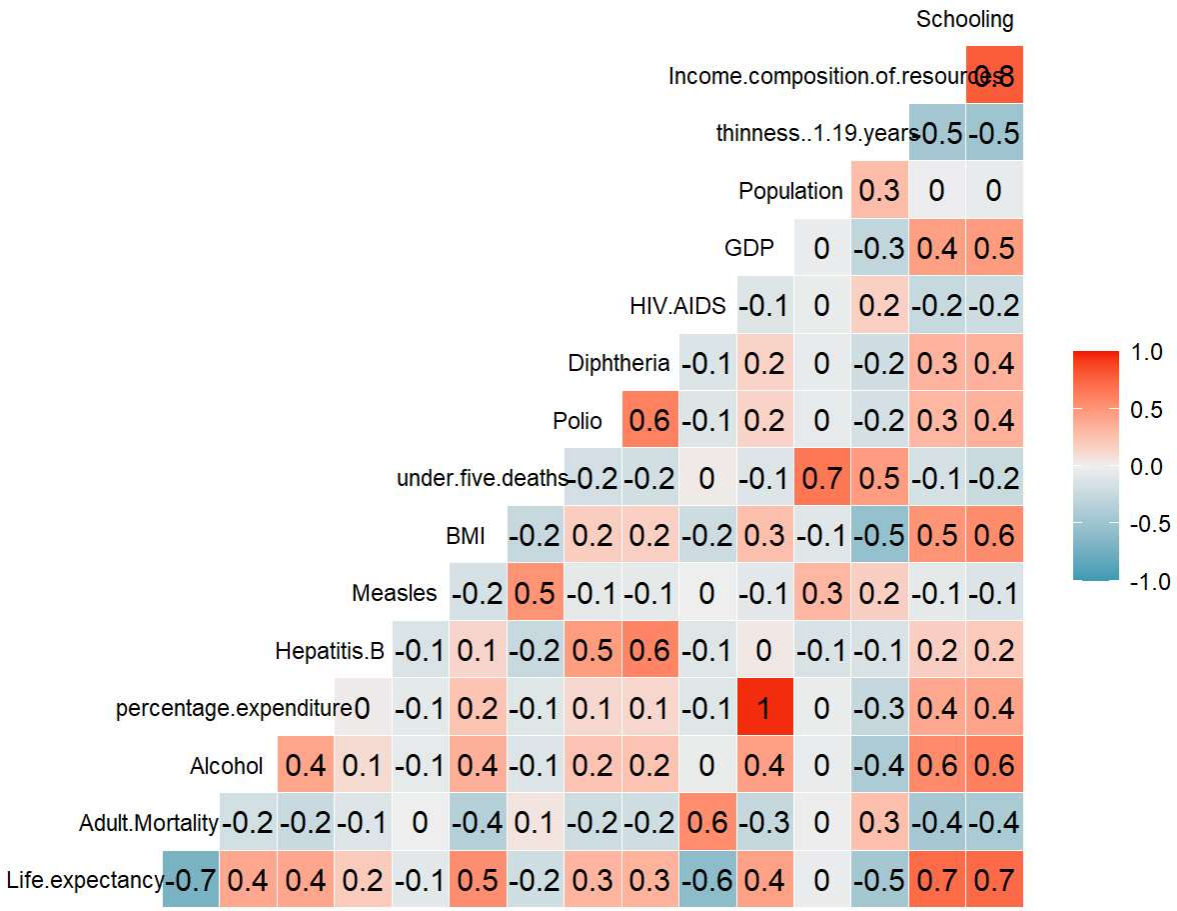
This suggests that schooling have very strong relationship with life expectancy. As percentages of schooling increases, life expectancy increases drastically.

More Analysis

After realizing the overlap of data throughout the analysing process, we could then draw a new heat map excluding the overlapping variables such as thinness 5~9 years, infant deaths, total expentiture

Hide

```
life_relation <- Life_Expectancy_World %>%
  select(-Country, -Year, -Status, -infant.deaths,-Total.expenditure,-thinness.5.9.years)
life_cor <- cor(life_relation)
ggcorr(life_relation ,
  cor_matrix = life_cor,
    label = T,
    hjust = 0.8,
    angle = 0,
    size = 3,
    layout.exp = 2
)
```



To wrap things up, in this section we processed the raw data and explained the usage of each variables as well as investigated itself correlation with other variables. It is worth Noting that BMI, Schooling,percentage expenditure, and Income composition of resources have strong positive relationship with life expectancy while HIV/AIDS and Adult mortality have drastic negative relations with life expectancy. We also learned that the status of development correlates to life expectancy as countries that are categorized as developed countries tend to have higher life expectancy than that of developing countries. Another discovery is that time and population tends not to affect life expectancy or having really weak relationship. However, Alcohol and life expectancy has a counter-intuitive positive relationship which contradicts common sense, we hypothesize this might be the cause of wrong data inputs of the original dataset. From analyzing the dataset, we also discovered that most countries have their average life expectancy at around 70-75 years while differences in life expectancy tends to be huge, as countries such as Ireland have significant higher life expectancy than countries like Zimbabwe. Almost twice the amount.

Model Building

In this section, we will be mainly focused on creating models to for this project. As previously mentioned in earlier sections, we will be applying various machine learning techniques using the same recipe. ### Training/Testing Split Before creating our recipe, we first need to perform a training/testing split on our data. We eventually will use 80% of the data for training and 20% of the data for testing, as this percentage is commonly used and tends to yield a persuasive result. We need to perform such process to avoid overfitting of models. We also set a random seed to ensure that the training and testing split is the same so that it would always generate the same result so that we don't have to deal with different results every time we run the code. We stratify on our variable "status"

Hide

```
set.seed(23145) #set random seed in order to repeat the result could be any number
Life_split <- initial_split(Life_Expectancy_World, strata = "Status", prop = 0.8)
Life_Train <- training(Life_split)
Life_Test <- testing(Life_split)
dim(Life_Train)
```

```
## [1] 1318   22
```

Hide

```
dim(Life_Test)
```

```
## [1] 331  22
```

As the outcome variable is imbalanced for this dataset, using stratified sampling method for this data allows every subgroup in the population receives proper representation. ### Recipe Building Now comes the important part, because in the modeling building section we will be using various models based on the same recipe, so the coding of recipe matters greatly. It is like cooking:we know we want to make some dish, but which one? What are we accomplishing with that dish? What purpose is that dish for? We are using data from WHO combined with UN on life expectancy, and not all variables will be presented in the recipe process, we will exclude those variables that overlap or are hard to perceive such as infant deaths, total expenditure, thinness 5~9years old, and Alcohol. Doing so is to prevent overfitting of model and increase accuracy of models. Hence, what we will be including in the recipe would be life expectancy, status, adult mortality, Hepatitis B, measles, BMI, under five deaths, Polio, Diphtheria, HIV/AIDS, GDP, population, thinness 1-19years, income composition of resources,and schooling.

Hide

```
life_recipe<-
  recipe(formula= Life.expectancy ~ Status + percentage.expenditure +Adult.Mortality +unde
r.five.deaths + thinness..1.19.years + BMI + GDP +Income.composition.of.resources +Hepatit
is.B + Polio + Diphtheria +Measles + HIV.AIDS + Schooling, data = Life_Train) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_normalize(all_predictors()) %>%
  step_novel(all_nominal_predictors()) %>%
  step_zv(all_nominal_predictors())
summary(life_recipe)
```

```
## # A tibble: 15 x 4
##    variable                        type    role      source
##    <chr>                           <chr>   <chr>     <chr>
##  1 Status                          nominal predictor original
##  2 percentage.expenditure          numeric predictor original
##  3 Adult.Mortality                 numeric predictor original
##  4 under.five.deaths               numeric predictor original
##  5 thinness..1.19.years            numeric predictor original
##  6 BMI                             numeric predictor original
##  7 GDP                             numeric predictor original
##  8 Income.composition.of.resources numeric predictor original
##  9 Hepatitis.B                     numeric predictor original
## 10 Polio                           numeric predictor original
## 11 Diphtheria                      numeric predictor original
## 12 Measles                         numeric predictor original
## 13 HIV.AIDS                        numeric predictor original
## 14 Schooling                       numeric predictor original
## 15 Life.expectancy                 numeric outcome   original
```

K-Fold Cross Validation

Hide

```
life_folds <- vfold_cv(Life_Train, v=10, strata = Status)
```

We are training the data by spliting it into 10 folds to evaluate the model's ability when given new data.The K-fold Cross-Validation is a method we use to estimate skill of machine learning models, the method is that it split the dataset into K number of folds and is used to evaluate the model's ability when given new data.Using K-fold Cross-Validation method instead of simple fit the model helps us avoid overfitting and gives the model the opportunity to train on multiple train-test splits.

Running the Models

In this part, we will be using 4 different models: Ridge Regression model, Lasso Regression model, Decision Tree model, and Boosted Tree model. Before the modelling begins, we predict Decision Tree model would perform the best,as decision trees have advantages in interpretation, less Data Preparation, is non-Parametric, and versatile. #### Ridge Regression Model As for the first model, we decided to go with Ridge regression as ridge regression model is capable of reducing standard error by adding bias in the regression process. Again it is a trade-off between vairance and bias.

Hide

```
rg_reg <- linear_reg(penalty = tune(), mixture = 0) %>%
  set_mode("regression") %>%
  set_engine("glmnet")

rg_wkflow <- workflow() %>%
  add_recipe(life_recipe) %>%
  add_model(rg_reg)

penalty_grid <- grid_regular(penalty(range = c(-4, 4)), levels = 10)

tune_res <- tune_grid(
  rg_wkflow,
  resamples = life_folds,
  grid = penalty_grid
)
```

```
## ! Fold01: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold02: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold03: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold04: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold05: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold06: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold07: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

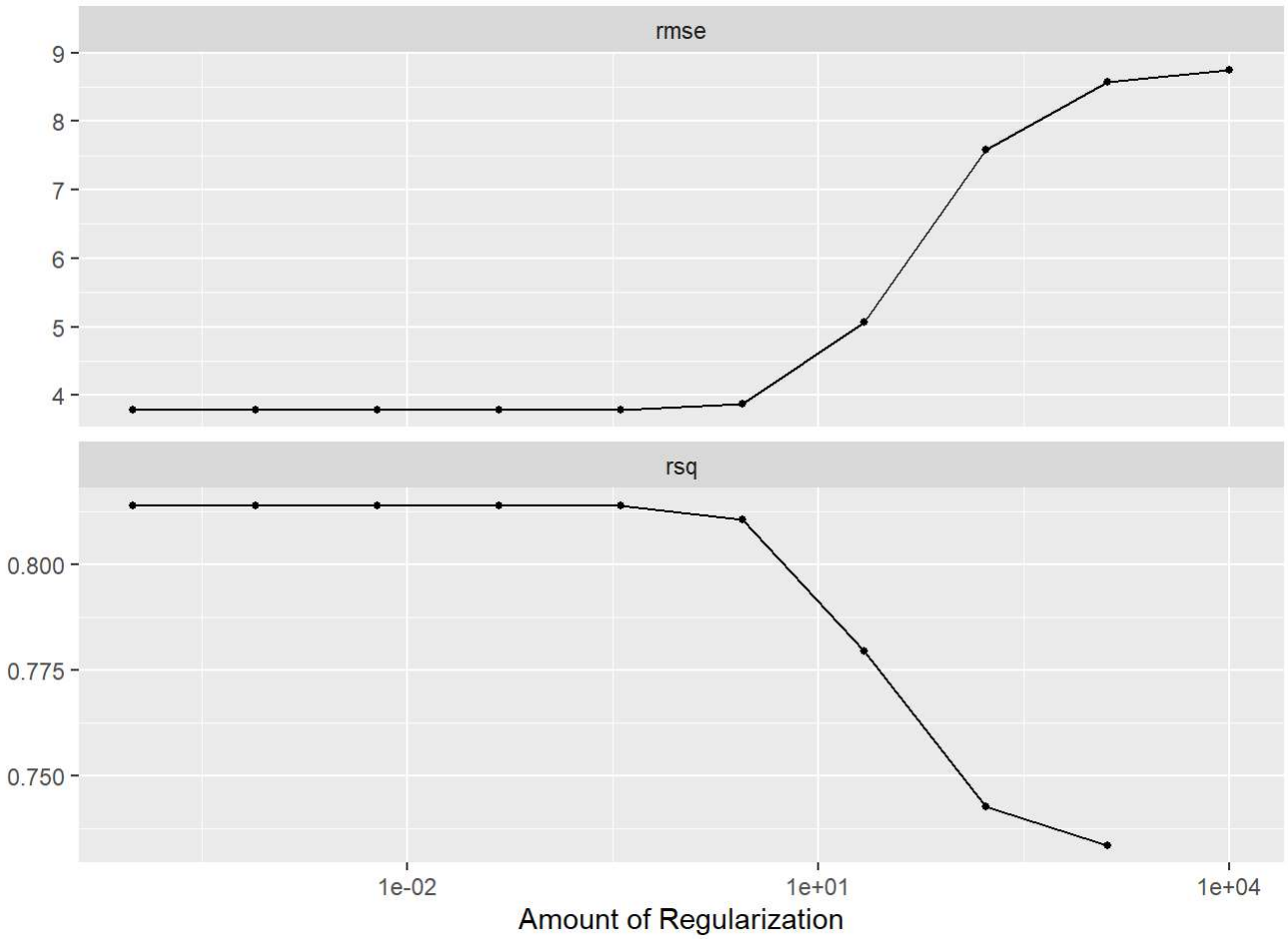
```
## ! Fold08: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold09: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold10: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

Hide

```
autoplot(tune_res)
```



Here we see that the amount of regularization affects the performance metrics differently.

Hide

```
collect_metrics(tune_res)
```

```
## # A tibble: 20 x 7
##         penalty .metric .estimator    mean     n  std_err .config
##           <dbl> <chr>   <chr>        <dbl> <int>    <dbl> <chr>
##  1     0.0001   rmse    standard     3.78     10  0.0766  Preprocessor1_Model01
##  2     0.0001   rsq     standard     0.814    10  0.00664 Preprocessor1_Model01
##  3     0.000774 rmse    standard     3.78     10  0.0766  Preprocessor1_Model02
##  4     0.000774 rsq     standard     0.814    10  0.00664 Preprocessor1_Model02
##  5     0.00599  rmse    standard     3.78     10  0.0766  Preprocessor1_Model03
##  6     0.00599  rsq     standard     0.814    10  0.00664 Preprocessor1_Model03
##  7     0.0464   rmse    standard     3.78     10  0.0766  Preprocessor1_Model04
##  8     0.0464   rsq     standard     0.814    10  0.00664 Preprocessor1_Model04
##  9     0.359    rmse    standard     3.78     10  0.0766  Preprocessor1_Model05
## 10     0.359    rsq     standard     0.814    10  0.00664 Preprocessor1_Model05
## 11     2.78     rmse    standard     3.87     10  0.0750  Preprocessor1_Model06
## 12     2.78     rsq     standard     0.811    10  0.00652 Preprocessor1_Model06
## 13    21.5      rmse    standard     5.06     10  0.0980  Preprocessor1_Model07
## 14    21.5      rsq     standard     0.780    10  0.00716 Preprocessor1_Model07
## 15   167.       rmse    standard     7.58     10  0.152   Preprocessor1_Model08
## 16   167.       rsq     standard     0.743    10  0.00839 Preprocessor1_Model08
## 17  1292.       rmse    standard     8.57     10  0.169   Preprocessor1_Model09
## 18  1292.       rsq     standard     0.733    10  0.00871 Preprocessor1_Model09
## 19 10000        rmse    standard     8.75     10  0.172   Preprocessor1_Model10
## 20 10000        rsq     standard   NaN         0 NA       Preprocessor1_Model10
```

We need to find the best values of this

Hide

```
best_penalty <- select_best(tune_res, metric = "rmse")
best_penalty
```

```
## # A tibble: 1 x 2
##   penalty .config
##     <dbl> <chr>
## 1  0.0001 Preprocessor1_Model01
```

Hide

```
rg_final <- finalize_workflow(rg_wkflow, best_penalty)

rg_final_fit <- fit(rg_final, data = Life_Train)

rsq_acc<-augment(rg_final_fit, new_data = Life_Test) %>%
  rsq(truth = Life.expectancy, estimate = .pred)
rsq_acc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.849
```

From the results, we could observe the maximum rmse is 0.8485144

Lasso Regression Model

we decided to choose lasso regression due to the fact that lasso regression allows us to minimize coefficient of variables to prevent overfitting. As our original data contains varaibles that overlap, it will inevitably cause overlap

Hide

```
ls_reg <- linear_reg(penalty = tune(),mixture = 1) %>%
  set_mode("regression") %>%
  set_engine("glmnet")

ls_wkflow <- workflow() %>%
  add_recipe(life_recipe) %>%
  add_model(ls_reg)

penalty_grid_2<- grid_regular(penalty(range = c(-2, 2)), levels = 10)

tune_ls <-tune_grid(
  ls_wkflow,
  resamples = life_folds,
  grid = penalty_grid_2,
  )
```

```
## ! Fold01: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold02: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold03: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold04: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold05: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold06: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold07: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

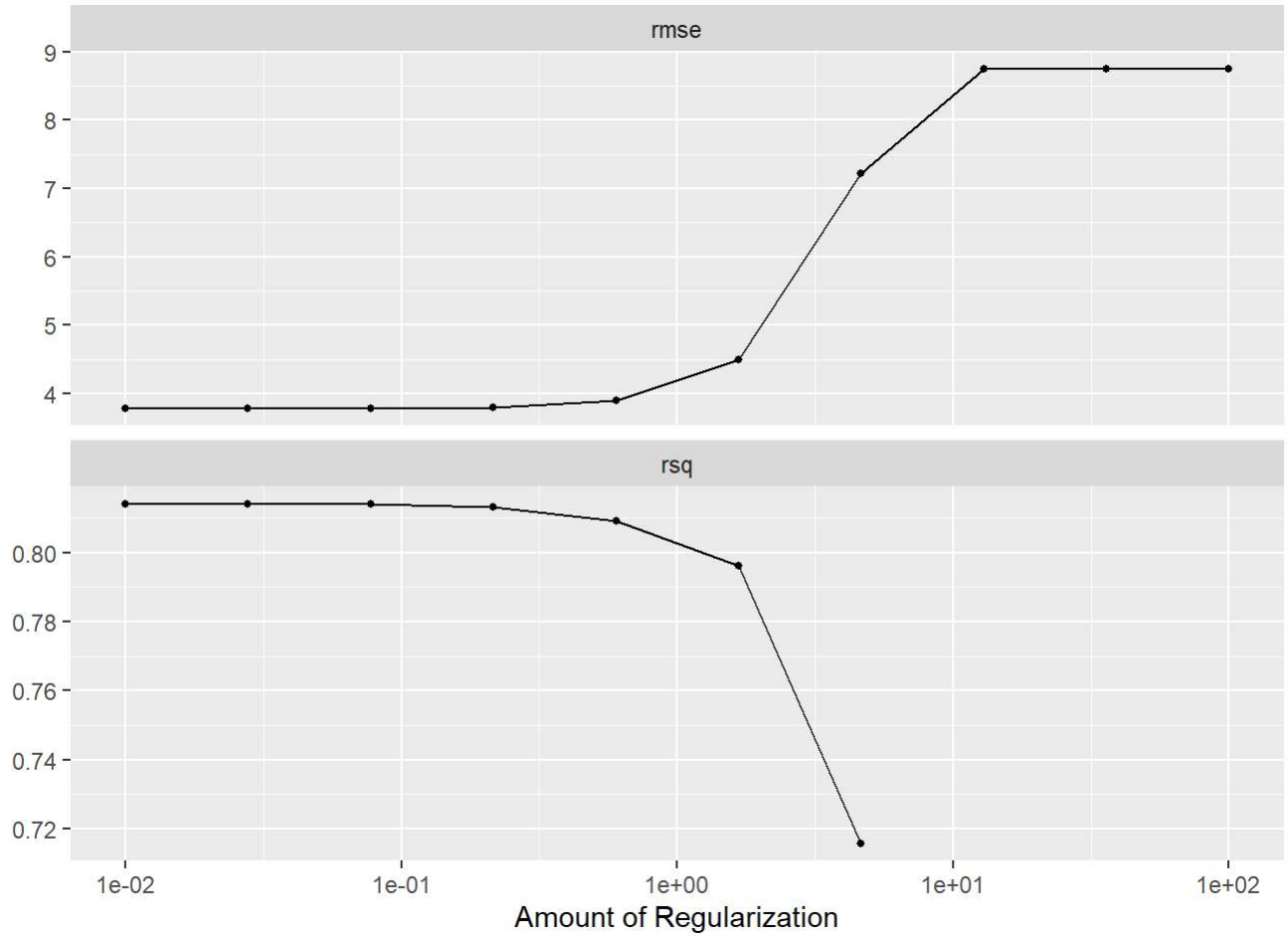
```
## ! Fold08: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold09: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold10: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

Hide

```
autoplot(tune_ls)
```



We need to find the best values of this

Hide

```
best_penalty <- select_best(tune_res, metric = "rsq")

ls_final <- finalize_workflow(ls_wkflow, best_penalty)

ls_final_fit <- fit(ls_final, data = Life_Train)

ls_acc<-augment(ls_final_fit, new_data = Life_Test) %>%
  rsq(truth = Life.expectancy, estimate = .pred)
ls_acc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.850
```

From the results, we could observe the maximum rsq is 0.8499223

Decision Tree Model

For the third model presented in the project, we decided to use Decision tree model as it is capable of dissolving complex data into more concrete parts.

Hide

```
tree_spec <- decision_tree()%>%
  set_engine("rpart")
reg_tree_spec <- tree_spec %>%
  set_mode("regression")
reg_tree_wkflow <- workflow() %>%
  add_model(reg_tree_spec %>% set_args(cost_complexity = tune())) %>%
  add_recipe(life_recipe)

param_grid <- grid_regular(cost_complexity(range = c(-4, 4)), levels = 10)

tune_res_tree <- tune_grid(
  reg_tree_wkflow,
  resamples = life_folds,
  grid = param_grid,
)
```

```
## ! Fold01: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold02: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold03: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold04: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold05: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold06: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold07: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

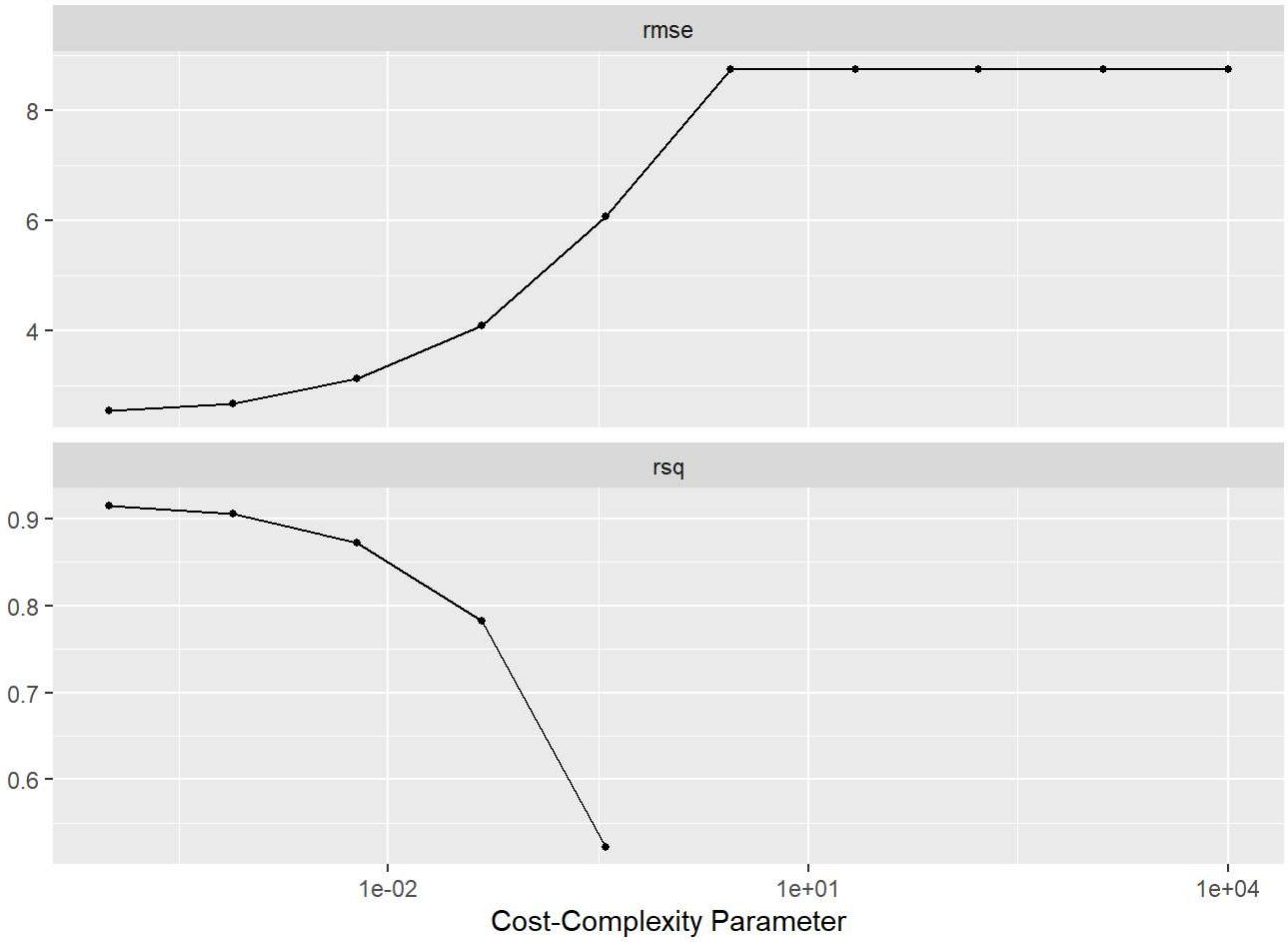
```
## ! Fold08: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold09: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

```
## ! Fold10: internal: A correlation computation is required, but `estimate` is constant a
nd ha...
```

Hide

```
autoplot(tune_res_tree)
```



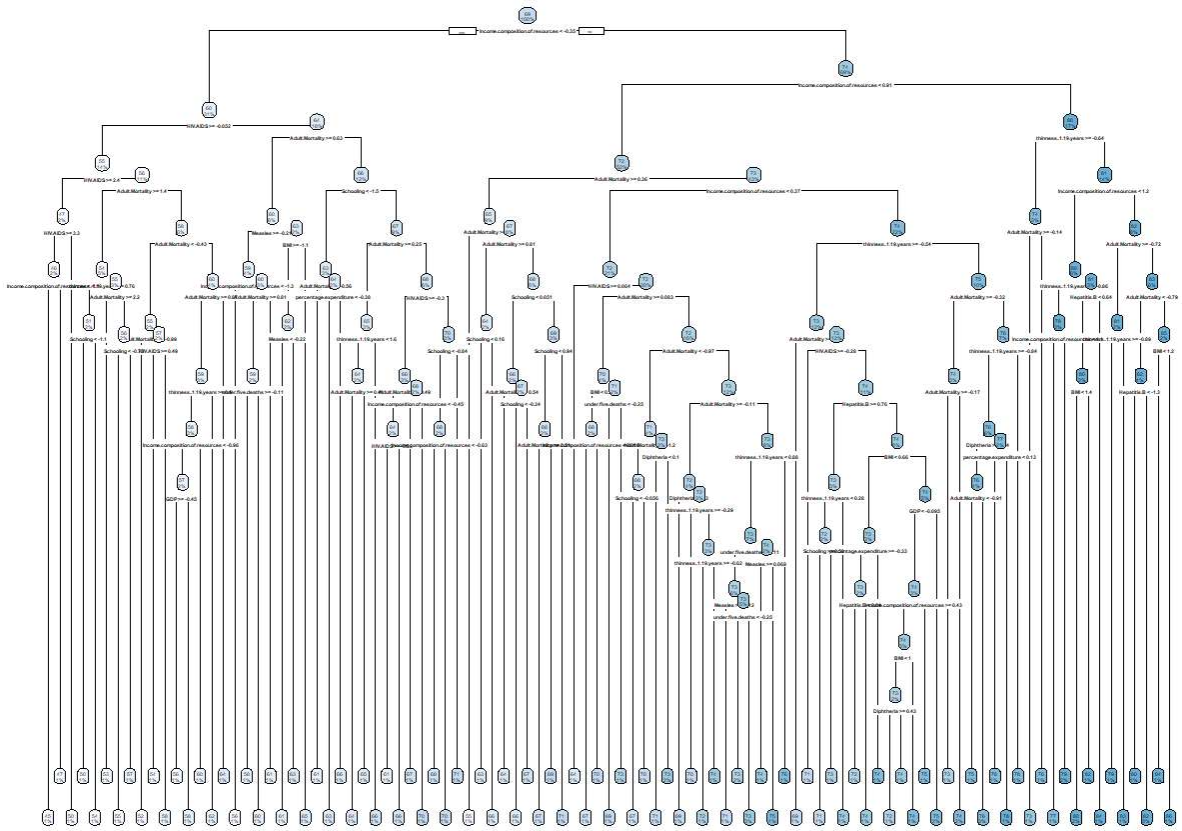
Hide

```
best_complexity <- select_best(tune_res_tree, metric = "rsq")

reg_tree_final <- finalize_workflow(reg_tree_wkflow, best_complexity)

reg_tree_final_fit <- fit(reg_tree_final, data = Life_Train)

reg_tree_final_fit %>%
  extract_fit_engine() %>%
  rpart.plot()
```



Hide

```
decision_tree_acc<-augment(reg_tree_final_fit, new_data = Life_Test) %>%
  rsq(truth = Life.expectancy, estimate = .pred)
decision_tree_acc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.932
```

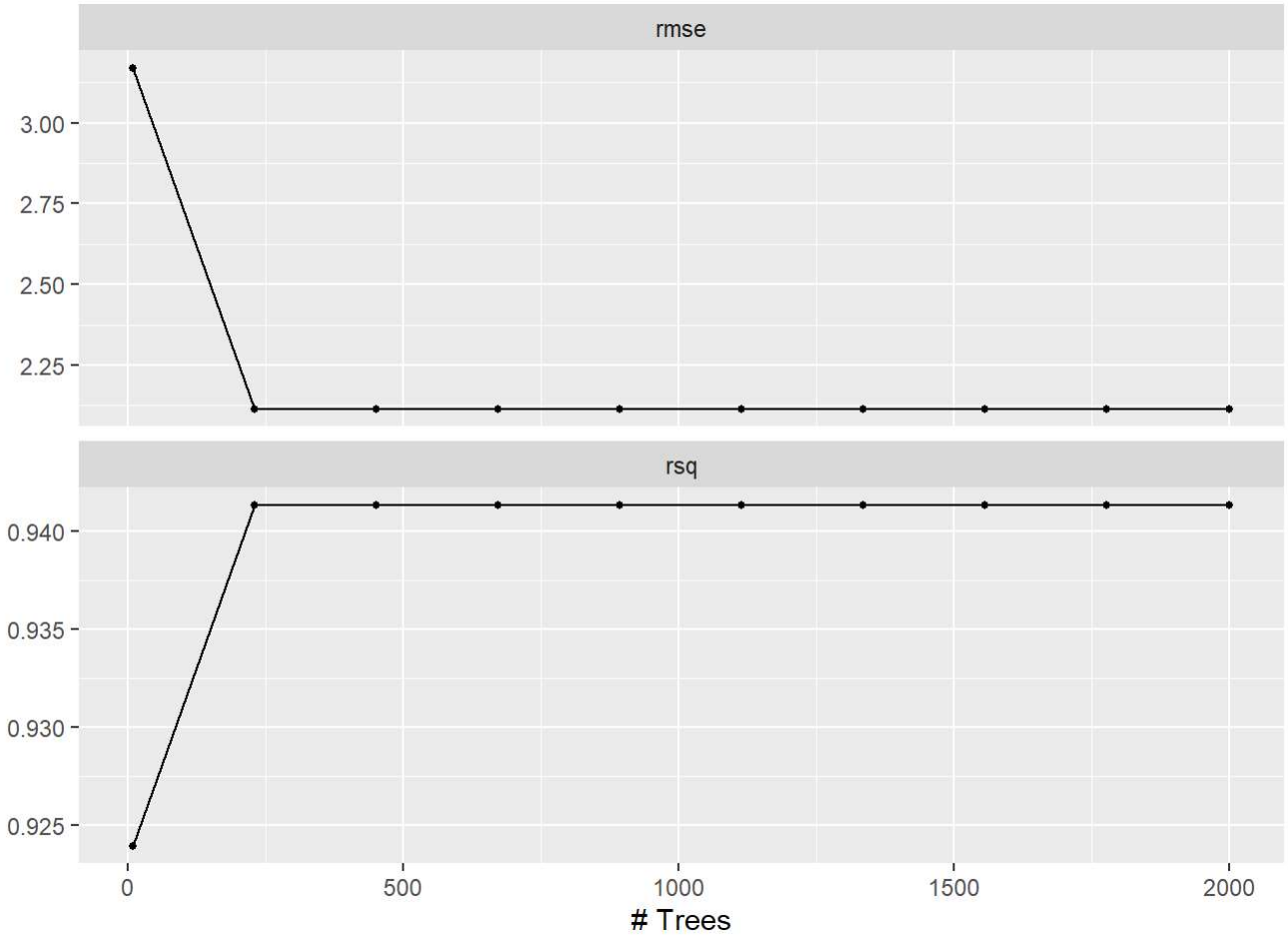
Hence the rsq of decision tree is 0.932291 #### Boosted Tree Model For the fourth model presented in the project, we used Boosted Tree Model. The advantage of using this model is that within each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule. However, boosted model is not perfect, no statistical method is. However, the major downside to this model is that it could over-fit the data when the trees are too deep with noisy data. Again, a trade-off.

Hide

```
boost_spec <- boost_tree() %>%
  set_engine("xgboost") %>%
  set_mode("regression")

boost_wkflow <- workflow() %>%
  add_model(boost_spec %>%
  set_args(trees = tune())) %>%
  add_recipe(life_recipe)
boost_grid <- grid_regular(trees(range = c(10,2000)), levels = 10)

boost_tune_res <- tune_grid(
  boost_wkflow,
  resamples = life_folds,
  grid = boost_grid
)
autoplot(boost_tune_res)
```



Hide

```
best_penalty <- select_best(boost_tune_res, metric = "rsq")

boost_final <- finalize_workflow(boost_wkflow, best_penalty)

boost_final_fit <- fit(boost_final, data = Life_Train)

boost_acc<-augment(boost_final_fit, new_data = Life_Test) %>%
  rsq(truth = Life.expectancy, estimate = .pred)
boost_acc
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.964
```

From the results we could observe that the model roc_auc reaches its maximum at around 10 trees. The maximum rsq is 0.9636825

Best Model Analyzing

In this section, our focus is to discover the best performing model when it comes to Life_Expectancy_World and predict make a prediction using it. We will mainly be comparing the Rsquare of different models.R-Squared is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable(s) in a regression model. In general, the higher the R-squared, the better the model fits the data. ### R Squared Comparison

Hide

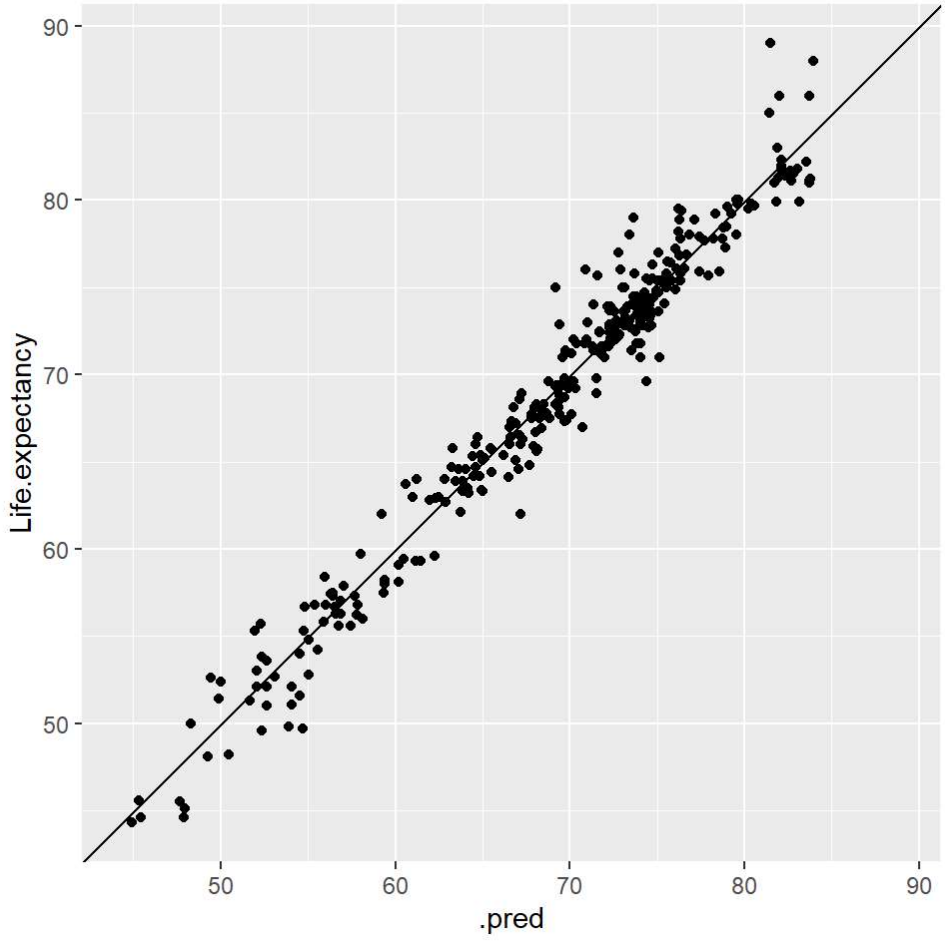
```
compare <- bind_rows(rsq_acc, ls_acc, decision_tree_acc,boost_acc)%>%
  arrange(.estimate)
compare
```

```
## # A tibble: 4 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.849
## 2 rsq     standard       0.850
## 3 rsq     standard       0.932
## 4 rsq     standard       0.964
```

Based on R Squared data, it seems that the boosted tree model performed the best, with a RSQ of 0.9636825. Congratulations Boosted Model!Ridge model performed the worst, yet with a rsq of 0.8485144, that is still significant. The reason why boost model became the best model might be it's feature that each model tries to compensate for the weaknesses of its predecessor.

Hide

```
Boost_Prediction <- predict(boost_final_fit, new_data = Life_Test %>% dplyr::select(-`Life.expectancy`))
Boost_Prediction <- bind_cols(Boost_Prediction, Life_Test %>% dplyr::select(`Life.expectancy`))
Boost_Graph <- Boost_Prediction %>%
  ggplot(aes(x=.pred, y=`Life.expectancy`)) + geom_point(alpha=1) + geom_abline() + coord_obs_pred()
Boost_Graph
```



Hide

```
Boost_Prediction_actual <- Boost_Prediction %>%
  bind_cols(Life_Test)
```

```
## New names:
## * `Life.expectancy` -> `Life.expectancy...2`
## * `Life.expectancy` -> `Life.expectancy...6`
```

Hide

```
Boost_Prediction_actual
```

```
## # A tibble: 331 x 24
##    .pred Life.exp~1 Country  Year Status Life.~2 Adult~3 infan~4 Alcohol perce~5
##    <dbl>      <dbl> <chr>   <int> <chr>    <dbl>   <int>   <int>   <dbl>   <dbl>
##  1  57.7       57.3 Afghan~  2005 Devel~    57.3     291      85    0.02    1.39
##  2  56.9       57   Afghan~  2004 Devel~    57       293      87    0.02   15.3
##  3  76.3       77.8 Albania  2015 Devel~    77.8      74       0    4.6   365.
##  4  76.5       76.1 Albania  2009 Devel~    76.1      91       1    5.79  348.
##  5  72.7       72.9 Algeria  2005 Devel~    72.9     136      19    0.5     2.55
##  6  54.1       51.1 Angola   2013 Devel~    51.1     355      69    8.1    36.0
##  7  50.5       48.2 Angola   2007 Devel~    48.2     375      87    6.35  185.
##  8  78.6       75.9 Argent~  2012 Devel~    75.9      12       9    8.35 1134.
##  9  78.0       75.7 Argent~  2011 Devel~    75.7      12       9    8.11 1504.
## 10  75.0       75.4 Argent~  2008 Devel~    75.4     126      10    8.41 1414.
## # ... with 321 more rows, 14 more variables: Hepatitis.B <int>, Measles <int>,
## #   BMI <dbl>, under.five.deaths <int>, Polio <int>, Total.expenditure <dbl>,
## #   Diphtheria <int>, HIV.AIDS <dbl>, GDP <dbl>, Population <dbl>,
## #   thinness..1.19.years <dbl>, thinness.5.9.years <dbl>,
## #   Income.composition.of.resources <dbl>, Schooling <dbl>, and abbreviated
## #   variable names 1: Life.expectancy...2, 2: Life.expectancy...6,
## #   3: Adult.Mortality, 4: infant.deaths, 5: percentage.expenditure
```

Hide

```
Boost_augmented <- augment(boost_final_fit, new_data = Life_Test)
Boost_augmented
```

```
## # A tibble: 331 x 23
##    Country   Year Status Life.~1 Adult~2 infan~3 Alcohol perce~4 Hepat~5 Measles
##    <chr>    <int> <chr>    <dbl>   <int>   <int>   <dbl>   <dbl>   <int>   <int>
##  1 Afghani~  2005 Devel~    57.3     291      85    0.02    1.39      66    1296
##  2 Afghani~  2004 Devel~    57       293      87    0.02   15.3       67     466
##  3 Albania   2015 Devel~    77.8      74       0    4.6   365.        99       0
##  4 Albania   2009 Devel~    76.1      91       1    5.79  348.        98       0
##  5 Algeria   2005 Devel~    72.9     136      19    0.5     2.55      83    2302
##  6 Angola    2013 Devel~    51.1     355      69    8.1    36.0       77    8523
##  7 Angola    2007 Devel~    48.2     375      87    6.35  185.        73    1014
##  8 Argenti~  2012 Devel~    75.9      12       9    8.35 1134.        91       2
##  9 Argenti~  2011 Devel~    75.7      12       9    8.11 1504.        91       3
## 10 Argenti~  2008 Devel~    75.4     126      10    8.41 1414.         9       0
## # ... with 321 more rows, 13 more variables: BMI <dbl>,
## #   under.five.deaths <int>, Polio <int>, Total.expenditure <dbl>,
## #   Diphtheria <int>, HIV.AIDS <dbl>, GDP <dbl>, Population <dbl>,
## #   thinness..1.19.years <dbl>, thinness.5.9.years <dbl>,
## #   Income.composition.of.resources <dbl>, Schooling <dbl>, .pred <dbl>, and
## #   abbreviated variable names 1: Life.expectancy, 2: Adult.Mortality,
## #   3: infant.deaths, 4: percentage.expenditure, 5: Hepatitis.B
```

Hide

```
Boost_results <- augment(boost_final_fit, new_data = Life_Test) %>%
  rsq(Life.expectancy, estimate = .pred) %>%
  select(.estimate)
Boost_results
```

```
## # A tibble: 1 x 1
##   .estimate
##       <dbl>
## 1     0.964
```

In the world of statistics, any rsq greater than 0.7 is considered to be great in terms of measuring the model, so having a rsq of 0.9636825 indicates high accuracy in predicting the outcome.

Conclusion

Throughout the entire research, testing, and analysis, the best model to predict the life expectancy and other factors is the boosted model. Such result may have occured due to the nature/logic of boosted model, where a random sample of data is selected, fitted with a model and then trained sequentially.Each model tries to compensate for the weaknesses of its predecessor. With each iteration, the weak rules from each individual classifier are combined to form one, strong prediction rule. However, boosted model is not perfect, no statistical method is. The problem with Boosted model is that it could over-fit the data when the trees are too deep with noisy data. Nonetheless, the model yielded great result with a significant R square. As for improvements, perhaps the best way is to find more data regarding recent(less than five years till now) life expectancy and other factors as such data is lacking. In other words, the result and model would be more accurate if the original data was prepared better. One of the main struggles of writing this project was that certain correlations were perplexing: Take the Alcohol component as the example, it showed positive correlation with life expectancy, which means that the more alcohol one consumes, the longer they are expected to live which goes against common sense. Eventually Alcohol was deleted from being included in the final recipe to diminish possible errors caused. Further research avenues may be including data of more recent dates while substituting bad data, for example, alcohol and include longitudinal data analysis. Overall, the life expectancy data set provided various challenges for the author to utilze Machine Learning tools and provided a chance to apply what is taught in class to “real life scenarios”

Acknowledgements

This project could not be completed without the help of Professor Coburn, the help from TAs, and the help from various lab notes. We sincerely thank you for your time and patience. *The original dataset came from World Health Organization, available at https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-life-expectancy-and-healthy-life-expectancy (https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-life-expectancy-and-healthy-life-expectancy)* The Integrated dataset came from Non-profit Website Kaggle, available at https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who (https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who)