Data Cleaning & Linear Regression

2023-02-09

Part 1: Data Wrangling

Call Libraries

```
library(tidyverse)
library(broom)
library(dplyr)
```

Importing the Data

```
life_df <- read.csv("life_data.csv")
str(life_df)
```

```
## 'data.frame':    2938 obs. of  22 variables:
##  $ Country                        : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ Year                           : int  2015 2014 2013 2012 2011 2010 2009 2008 2007 2006 ...
##  $ Status                         : chr  "Developing" "Developing" "Developing" "Developing" ...
##  $ Life.expectancy                : num  65 59.9 59.9 59.5 59.2 58.8 58.6 58.1 57.5 57.3 ...
##  $ Adult.Mortality                : int  263 271 268 272 275 279 281 287 295 295 ...
##  $ infant.deaths                  : int  62 64 66 69 71 74 77 80 82 84 ...
##  $ Alcohol                        : num  0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.03 0.02 0.03 ...
##  $ percentage.expenditure         : num  71.3 73.5 73.2 78.2 7.1 ...
##  $ Hepatitis.B                    : int  65 62 64 67 68 66 63 64 63 64 ...
##  $ Measles                        : int  1154 492 430 2787 3013 1989 2861 1599 1141 1990 ...
##  $ BMI                            : num  19.1 18.6 18.1 17.6 17.2 16.7 16.2 15.7 15.2 14.7 ...
##  $ under.five.deaths              : int  83 86 89 93 97 102 106 110 113 116 ...
##  $ Polio                          : int  6 58 62 67 68 66 63 64 63 58 ...
##  $ Total.expenditure              : num  8.16 8.18 8.13 8.52 7.87 9.2 9.42 8.33 6.73 7.43 ...
##  $ Diphtheria                     : int  65 62 64 67 68 66 63 64 63 58 ...
##  $ HIV.AIDS                       : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
##  $ GDP                            : num  584.3 612.7 631.7 670 63.5 ...
##  $ Population                     : num  33736494 327582 31731688 3696958 2978599 ...
##  $ thinness..1.19.years           : num  17.2 17.5 17.7 17.9 18.2 18.4 18.6 18.8 19 19.2 ...
##  $ thinness.5.9.years             : num  17.3 17.5 17.7 18 18.2 18.4 18.7 18.9 19.1 19.3 ...
##  $ Income.composition.of.resources: num  0.479 0.476 0.47 0.463 0.454 0.448 0.434 0.433 0.415 0.405
##  $ Schooling                      : num  10.1 10 9.9 9.8 9.5 9.2 8.9 8.7 8.4 8.1 ...
```

Rename colnames to names easier to reference in future code

```
life_df <- life_df %>%
  rename("country" = "Country",
         "year" = "Year",
         "status" = "Status",
         "life_exp_yrs" = "Life.expectancy",
```

```
        "adult_mortality" = "Adult.Mortality",
        "infant_deaths" = "infant.deaths",
        "alcohol" = "Alcohol",
        "perc_expend" = "percentage.expenditure",
        "hep_b" = "Hepatitis.B",
        "measles" = "Measles",
        "bmi" = "BMI",
        "5yr_deaths" = "under.five.deaths",
        "polio" = "Polio",
        "tot_expend" = "Total.expenditure",
        "diphtheria" = "Diphtheria",
        "hiv_aids" = "HIV.AIDS",
        "gdp" = "GDP",
        "population" = "Population",
        "thin_1to19" = "thinness..1.19.years",
        "thin_5to9" = "thinness.5.9.years",
        "inc_comp_resources" = "Income.composition.of.resources",
        "schooling" = "Schooling"
        )
```

```
colnames(life_df)
```

```
##  [1] "country"            "year"               "status"
##  [4] "life_exp_yrs"       "adult_mortality"    "infant_deaths"
##  [7] "alcohol"            "perc_expend"        "hep_b"
## [10] "measles"            "bmi"                "5yr_deaths"
## [13] "polio"              "tot_expend"         "diphtheria"
## [16] "hiv_aids"           "gdp"                "population"
## [19] "thin_1to19"         "thin_5to9"          "inc_comp_resources"
## [22] "schooling"
```

The column names are now much easier for our group to call for future code. Other than that the data is tidy, the only wrangling left to do is perhaps create a few more categorical variables since or data is light on categorical variables.

Creating another categorical variable from the hiv_aids column
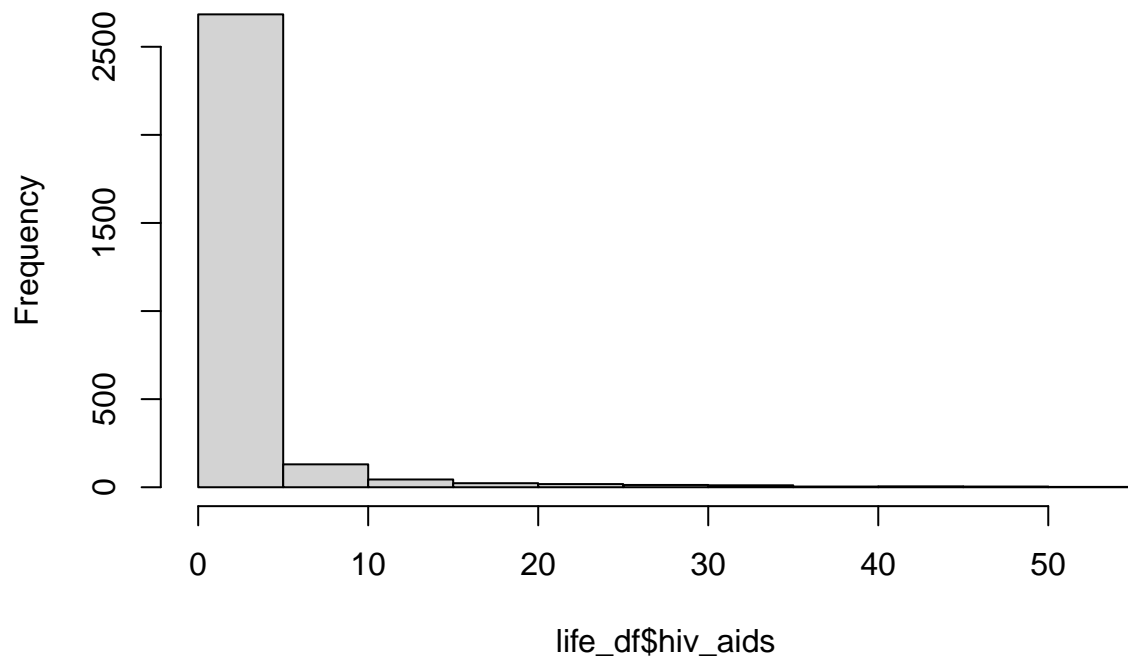
The hiv_aids column looks like it is able to be split into two groups: significantly low and high hiv_aids deaths per 1000 people.
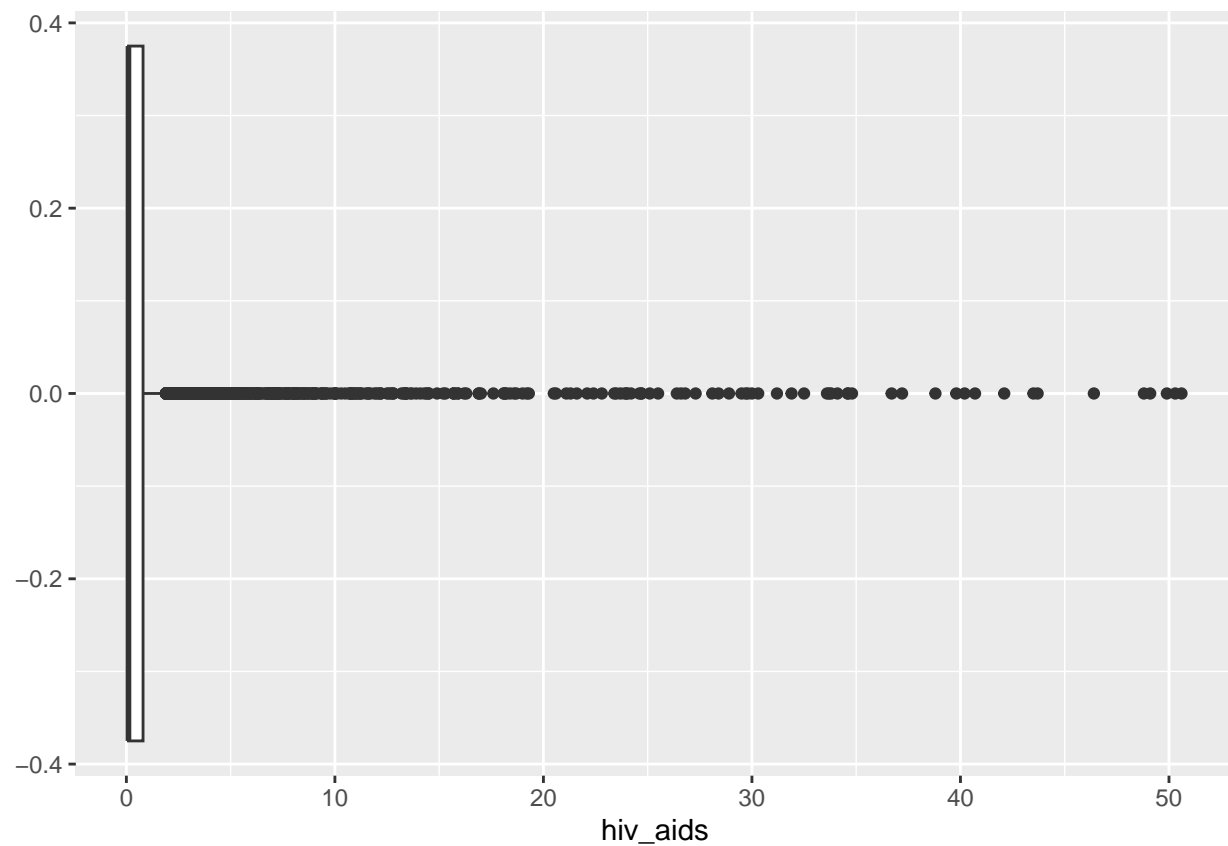
```
hist(life_df$hiv_aids,
     main = "Distribution of hiv_aids values in life_df")
```

Distribution of hiv_aids values in life_df


```
ggplot(data = life_df, aes(hiv_aids))+
  geom_boxplot()
```

Domain Information

After some research, I was able to find reasonable numbers to create our bins from. In 2019, our world in data reports average levels across North America and Europe as approximately 1 deaths per 100,000 people (0.1 in our data). The study reports that higher levels of HIV deaths occur at 100 deaths per 100,000 which means for my data, high levels of hiv deaths would occur at 1 death per 1,000 people so 1 will be the cutoff for these bins.

```
life_df <- life_df %>%
  mutate(
    hiv_deaths_cat = case_when(
      hiv_aids < 1 ~ 'Under 1 Death/1000',
      TRUE ~ 'Over 1 Death/1000'
    )
  )
```

Histogram of new column

```
ggplot(data = life_df, aes(x = hiv_deaths_cat))+
  geom_bar()
```
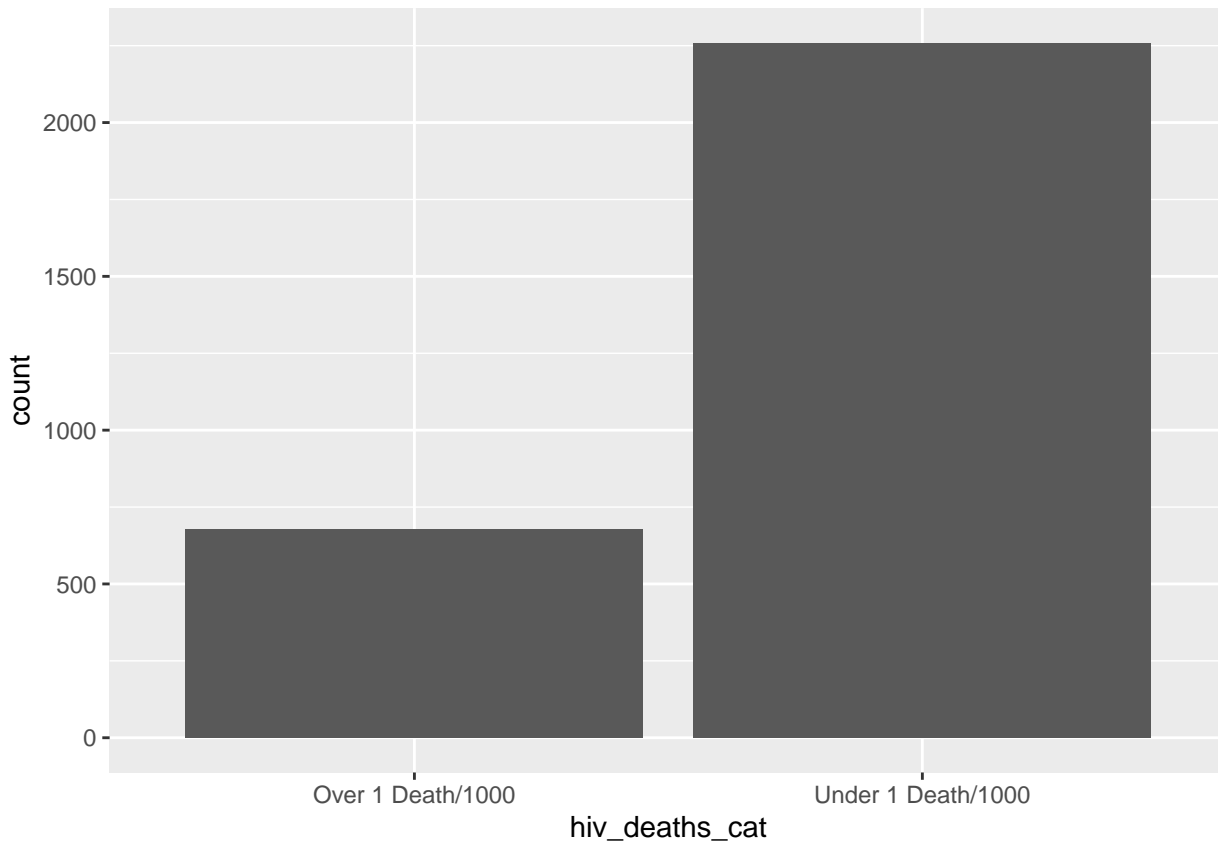

Creating a BMI categorical variable

I wanted to have one more categorical variable ready if we needed it in the future. Here are the official body mass index categories obtained from the CDC:

```
life_df <- life_df %>%
  mutate(bmi_cat = case_when(
```

```
    bmi <= 18.5 ~ "underweight",
    bmi <= 25 ~ "healthy",
    bmi <= 30 ~ "overweight",
    bmi <= 100 ~ "obese")
         )
```
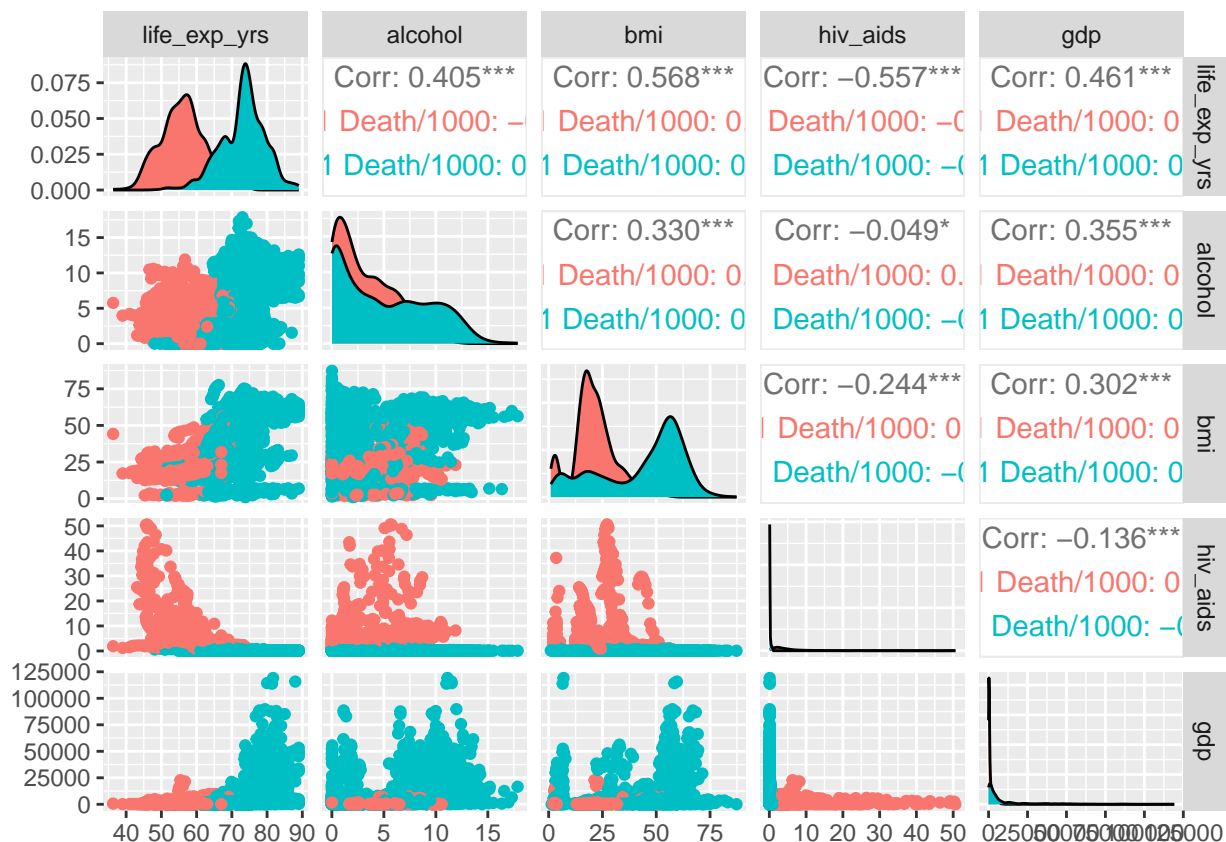
Part 2: Linear Models

Step 0 - Pairs Plot

```
library(GGally)
life_df %>%
ggpairs(columns=c(4, 7, 11, 16, 17),
        ggplot2::aes(color = hiv_deaths_cat))
```


Step 1 - Identifying Variables

Response:

'life_exp_yrs' - I am attempting to create a model that can accurately predict life expectancy given a selection of explanatory variables. The unit of this variable is years which is stored as a decimal accurate to 1 tenth of a year. Each record in the data is one country for a specific year, holding that country's average population life expectation.

Explanantory Numeric:

'bmi' - This is my first explanatory feature which is a record of a given country population's average BMI for that year. BMI is a calculation of body composition given height and weight.

Explanatory Categorical:

'hiv_deaths_cat' - This is my derived categorical variable which the creation of was explained in our data wrangling section. The levels of this variable are 'Under 1 Death/1000' and 'Over 1 Death/1000'.

Step 2 - Training and Testing Sets

We are using a 70:30 split for our testing and training sets.

```
dim(life_df) # 2938
```

```
## [1] 2938   24
```

```
set.seed(123)

trainInd<-sample(1:2938, 2057)

life_df_train<-life_df[trainInd, ]
life_df_test<-life_df[-trainInd, ]
```

```
dim(life_df_train)
```

```
## [1] 2057   24
```

```
dim(life_df_test)
```

```
## [1] 881   24
```

Step 3 - Simple Linear Model

```
life_mod1 <- lm(life_exp_yrs ~ bmi, data = life_df_train)
summary(life_mod1)
```

```
##
## Call:
## lm(formula = life_exp_yrs ~ bmi, data = life_df_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.676  -4.816   0.368   4.530  27.891
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 59.284322   0.377176  157.18   <2e-16 ***
## bmi          0.264514   0.008775   30.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.942 on 2029 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.3093, Adjusted R-squared:  0.309
## F-statistic: 908.7 on 1 and 2029 DF,  p-value: < 2.2e-16
```
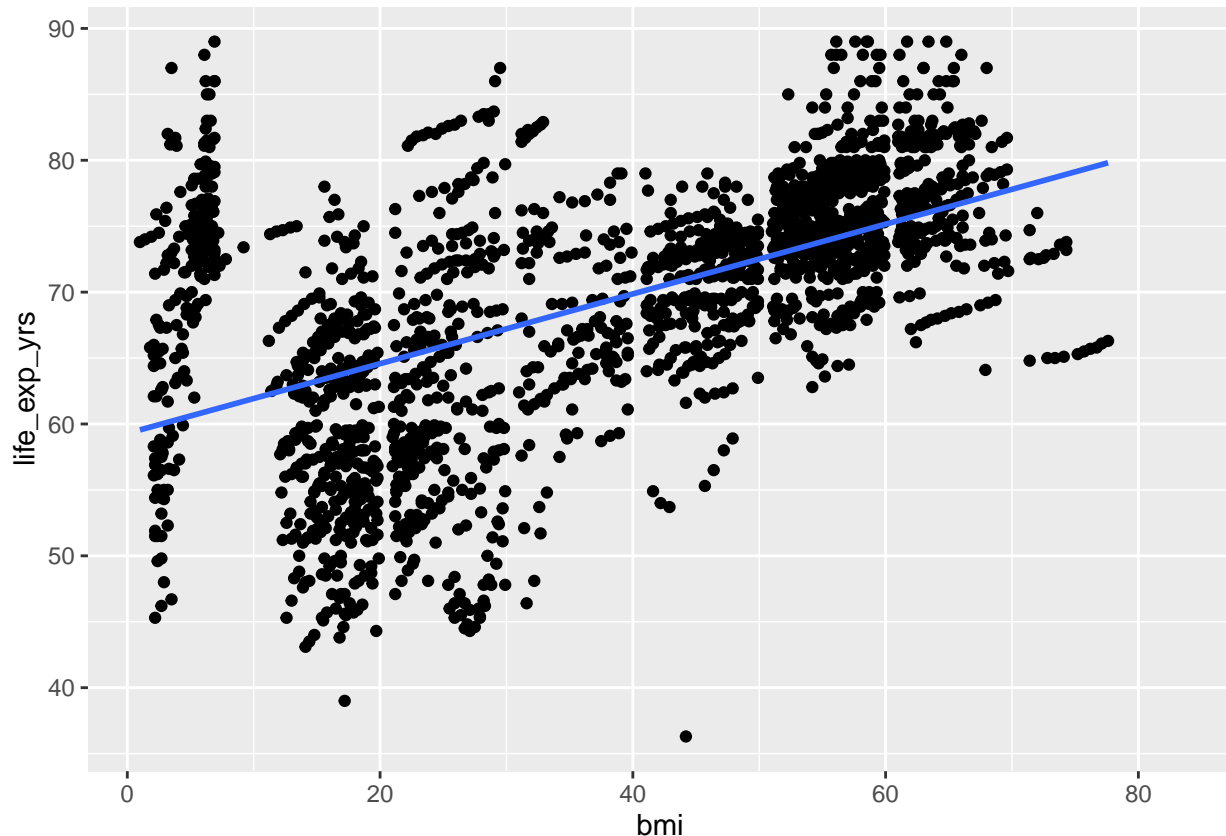
Model: $Y = 59.284322 + 0.264514x$

The relationship does appear to be significant; the p-value for bmi is significant at <2e-16. However the r-squared value shows the model does not account for much of the variability in the dataset.

Graphic

```
ggplot(data = life_df_train, aes(x=bmi,y=life_exp_yrs))+
  geom_point()+
  geom_smooth(method = 'lm', se = FALSE)
```


Step 4 - Parallel Slopes MLR Model

```
life_mod2 <- lm(life_exp_yrs ~ bmi + hiv_deaths_cat, data = life_df_train)
summary(life_mod2)
```

```
##
## Call:
## lm(formula = life_exp_yrs ~ bmi + hiv_deaths_cat, data = life_df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.0213  -3.5461  -0.0837   3.7482  20.5330
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     53.447460   0.314187  170.11   <2e-16 ***
```

```
## bmi                                  0.132892    0.007263   18.30    <2e-16 ***
## hiv_deaths_catUnder 1 Death/1000 14.102622    0.345896   40.77    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.889 on 2028 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.6204, Adjusted R-squared:  0.6201
## F-statistic:  1658 on 2 and 2028 DF,  p-value: < 2.2e-16
```

Model for reference group (hiv_deaths_cat Over 1 Death/1000): $Y = 53.447460 + 0.132892x$

Model for alt group (hiv_deaths_cat Under 1 Death/1000): $Y = (53.447460 + 14.102622) + 0.132892x$

Graphic

```
ggplot(data = life_df_train, aes(x = bmi, y = life_exp_yrs, color = hiv_deaths_cat))+
  geom_point()+
  geom_abline(intercept=life_mod2$coefficients[1],
              slope = life_mod2$coefficients[2], color = "red")+
  geom_abline(intercept=life_mod2$coefficients[1] + life_mod2$coefficients[3],
              slope = life_mod2$coefficients[2], color = "blue")
```
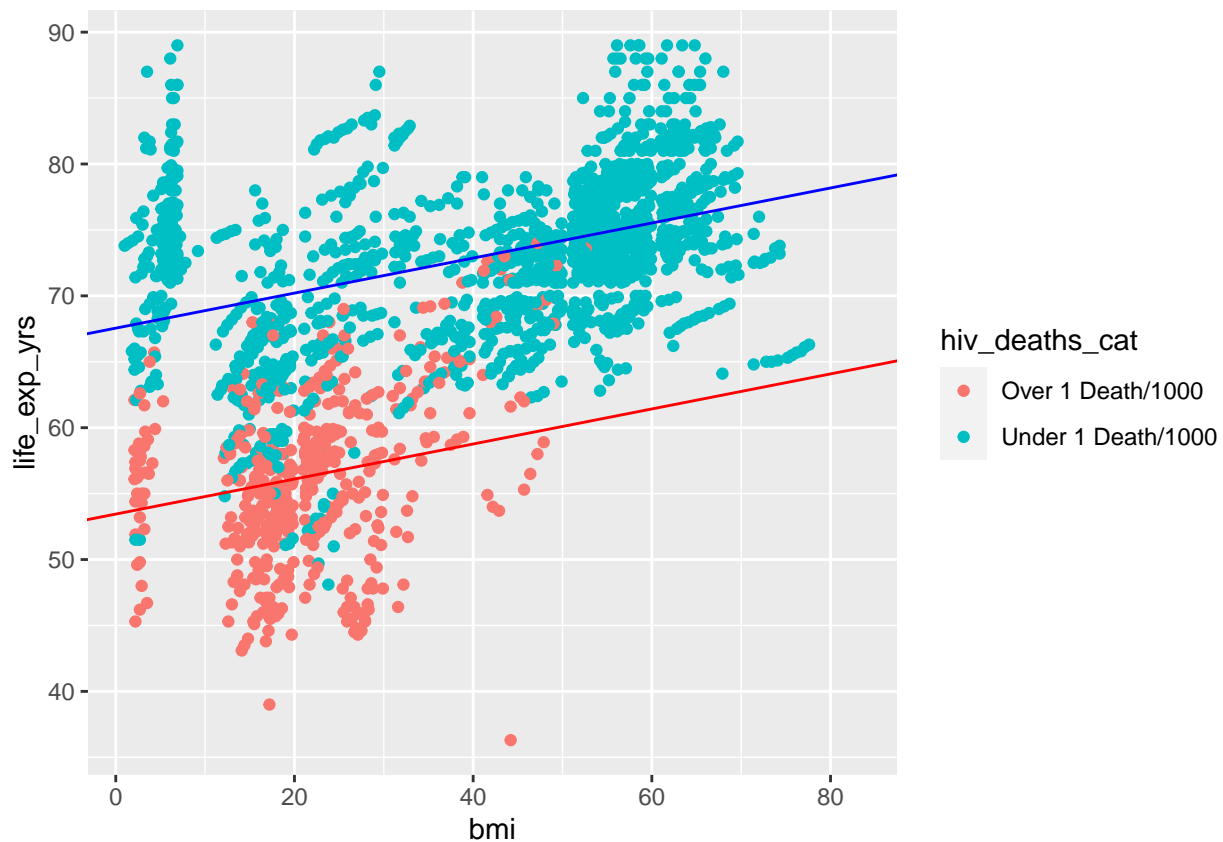

Step 5 - MLR with Interaction

```r
life_mod3 <- lm(life_exp_yrs ~ bmi * hiv_deaths_cat, data = life_df_train)
summary(life_mod3)
```

```
##
## Call:
## lm(formula = life_exp_yrs ~ bmi * hiv_deaths_cat, data = life_df_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.402  -3.456  -0.016   3.814  20.466
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          51.14432    0.62851  81.374  < 2e-16 ***
## bmi                                   0.23885    0.02610   9.151  < 2e-16 ***
## hiv_deaths_catUnder 1 Death/1000     16.78394    0.72206  23.244  < 2e-16 ***
## bmi:hiv_deaths_catUnder 1 Death/1000 -0.11478    0.02716  -4.225 2.49e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.865 on 2027 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.6238, Adjusted R-squared:  0.6232
## F-statistic:  1120 on 3 and 2027 DF,  p-value: < 2.2e-16
```

Model for reference group (hiv_deaths_cat Over 1 Death/1000): Y = 51.14432 + 0.23885x

Model for alt group (hiv_deaths_cat Under 1 Death/1000): Y = (51.14432 + 16.78394) + (0.23885 - 0.11478)x

Graphic

```r
life_mod3$coefficients
```

```
##                          (Intercept)                                  bmi
##                           51.1443245                            0.2388523
##     hiv_deaths_catUnder 1 Death/1000 bmi:hiv_deaths_catUnder 1 Death/1000
##                           16.7839429                           -0.1147757
## Reference
mod3_yint_0<-life_mod3$coefficients[1]
mod3_slope_0<-life_mod3$coefficients[2]

## Alternative
mod3_yint_1<-mod3_yint_0 + life_mod3$coefficients[3]
mod3_slope_1<-mod3_slope_0 + life_mod3$coefficients[4]
```
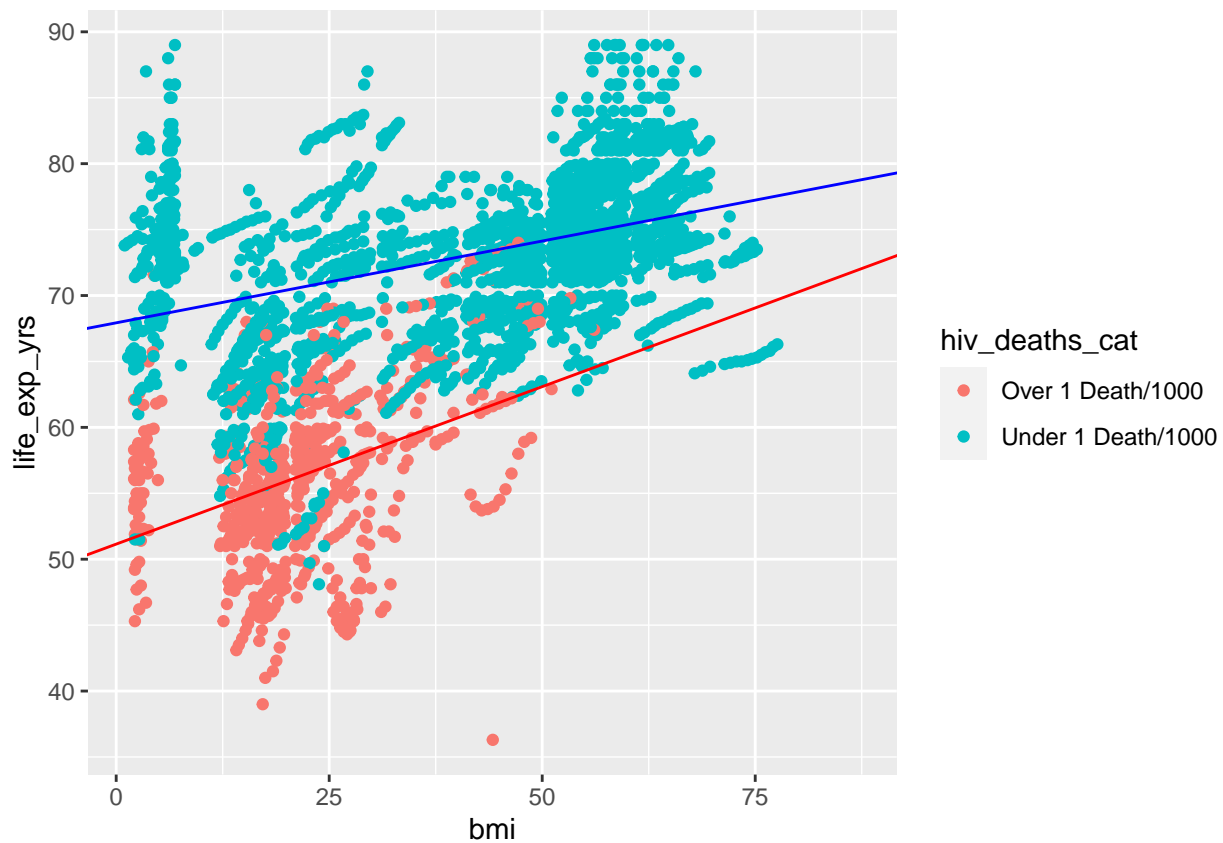
```r
ggplot(data=life_df, aes(x=bmi, y=life_exp_yrs, color=hiv_deaths_cat))+
  geom_point()+
  geom_abline(intercept=mod3_yint_0,
            slope=mod3_slope_0, color="red")+
  geom_abline(intercept=mod3_yint_1,
            slope=mod3_slope_1, color="blue")
```

Step 6 Prediction Tests

Mod 1

```
library(caret)
life_testPred1<-predict(life_mod1, life_df_test)

RMSE(life_testPred1, life_df_test$life_exp_yrs, na.rm = TRUE)

## [1] 7.501672
```

Mod 2

```
life_testPred2<-predict(life_mod2, life_df_test)
RMSE(life_testPred2, life_df_test$life_exp_yrs, na.rm = TRUE)

## [1] 5.476938
```

Mod 3

```
life_testPred3<-predict(life_mod3, life_df_test)
RMSE(life_testPred3, life_df_test$life_exp_yrs, na.rm = TRUE)

## [1] 5.425755
```

The model with the lowest RMSE is Model 3 (model with interaction between explanatory variables of 'bmi' and 'hiv_deaths_cat' columns).

```r
write.csv(life_df, "life_df.csv", row.names=FALSE)
```