STATS 415 Project Appendix

Ailun Huang, Judiah Lin, Raymond Pang

4/23/2019

Appendix I: External Requirements

Reading in the dataset and appropriate packages to analyze data:

```
suicide <- read.csv("~/Downloads/STATS 415/Project/suicide.csv")
library(tidyverse)
library(ggplot2)
library(knitr)
library(SignifReg)
library(leaps)
library(boot)
library(knitr)
library(splines)
library(gam)
```

Overview

This is the appendix for the STATS 415 Final Project. Our goal is to look at various methods and concepts taught from class to answer a question that is of interest to us. The following is the data set and the link of where we found it:

Data set: Suicide Rates Overview 1985 to 2016
Link: https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016

We are particularly interested in suicide data because we believe it is a prominent issue all over the world. Because the data set contains 27,400 rows, we will only look at a subsection of the data. We are particularly interested in the United States, so we will only be using rows of data pertaining to the U.S. We are hoping to predict the percentages of suicide with various factors such as year, sex, age group, number of suicides, population, and GDP per capita. The goal of this project is to see how accurately we can predict suicide rates using a data method taught in class. If we are able to predict this accurately, we can make conclusions for certain sex and/or age group. We can advise the American Suicide Prevention to create programs specific to those sex and/or age group in hopes of lowering suicide rates.

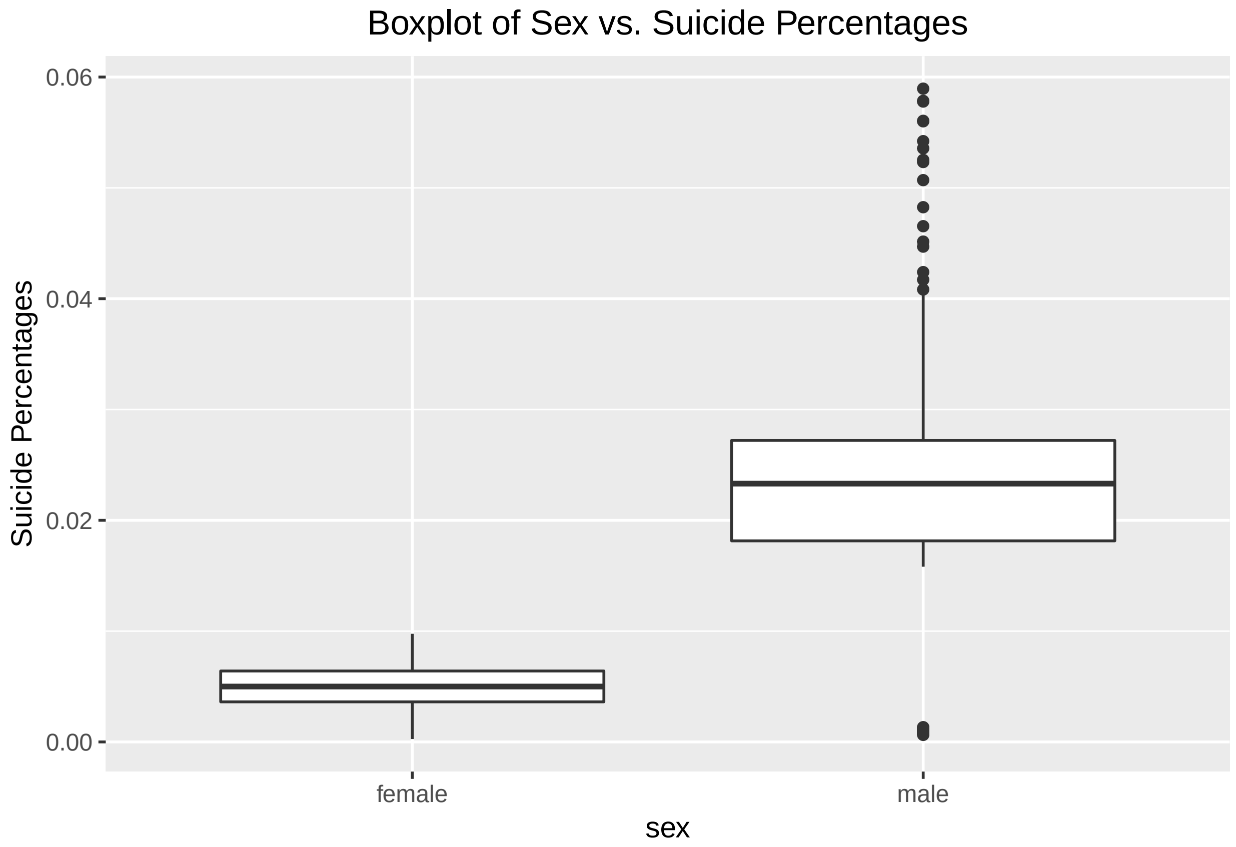
Appendix II: Exploratory Data

Modifying the data set to fit the needs of our research question:

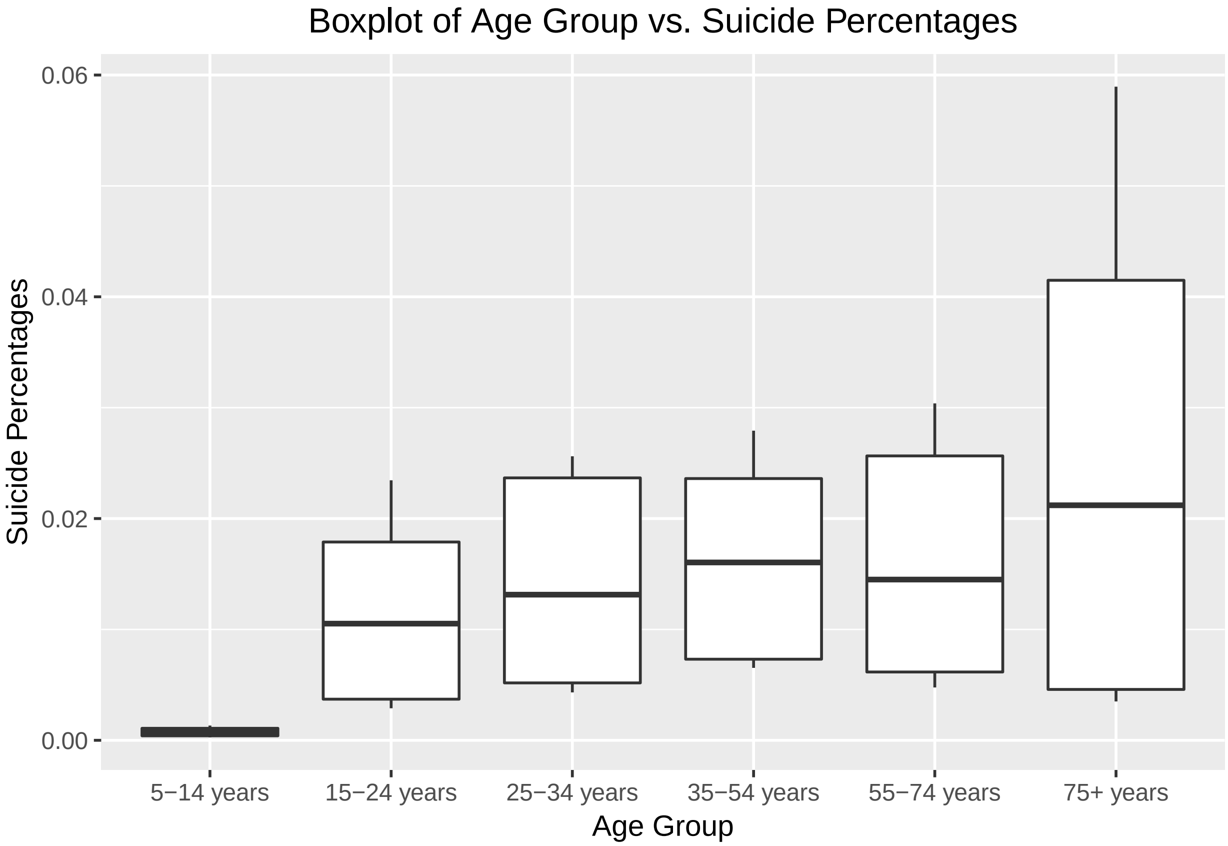
```
# Filtering to only the United States and selecting variables of interest.
# Also created a new variable called suicide_perc.
onlyUS = suicide %>% filter(country == "United States") %>%
  select(year, sex, age, suicides_no,
         population, gdp_per_capita....) %>%
  mutate(suicide_perc = (suicides_no / population) * 100)

# Ordering the age group
onlyUS$age <- factor(onlyUS$age,
  levels =  c('5-14 years', '15-24 years', '25-34 years',
              '35-54 years', '55-74 years', '75+ years'),
  ordered = TRUE)

# Boxplot of Sex vs. Suicide Percentages
ggplot(onlyUS, aes(x = sex, y = suicide_perc)) +
  geom_boxplot() +
  ylab("Suicide Percentages") +
  ggtitle("Boxplot of Sex vs. Suicide Percentages") +
  theme(plot.title = element_text(hjust = 0.5))
```

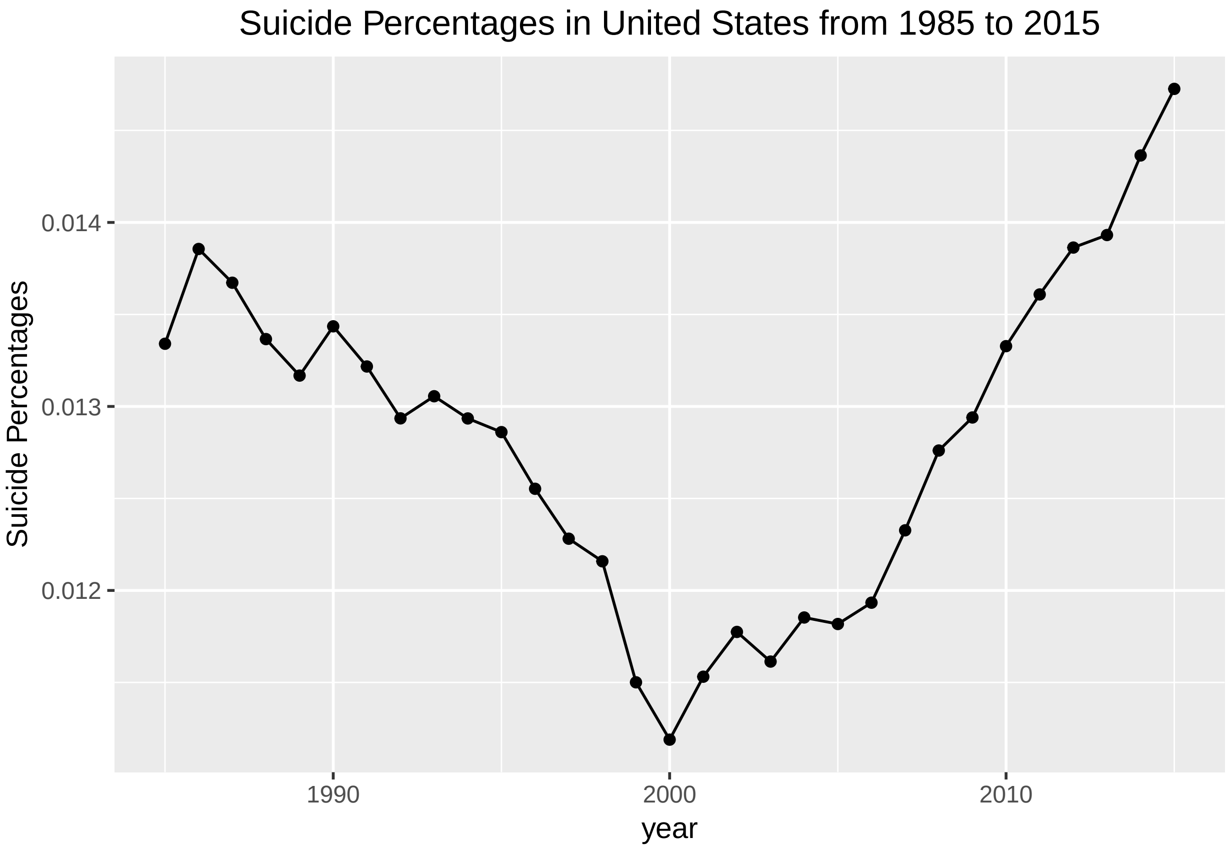


```
# Boxplot of Age Group vs. Suicide Percentages
ggplot(onlyUS, aes(x = age, y = suicide_perc)) +
  geom_boxplot() +
  xlab("Age Group") +
  ylab("Suicide Percentages") +
  ggtitle("Boxplot of Age Group vs. Suicide Percentages") +
  theme(plot.title = element_text(hjust = 0.5))
```



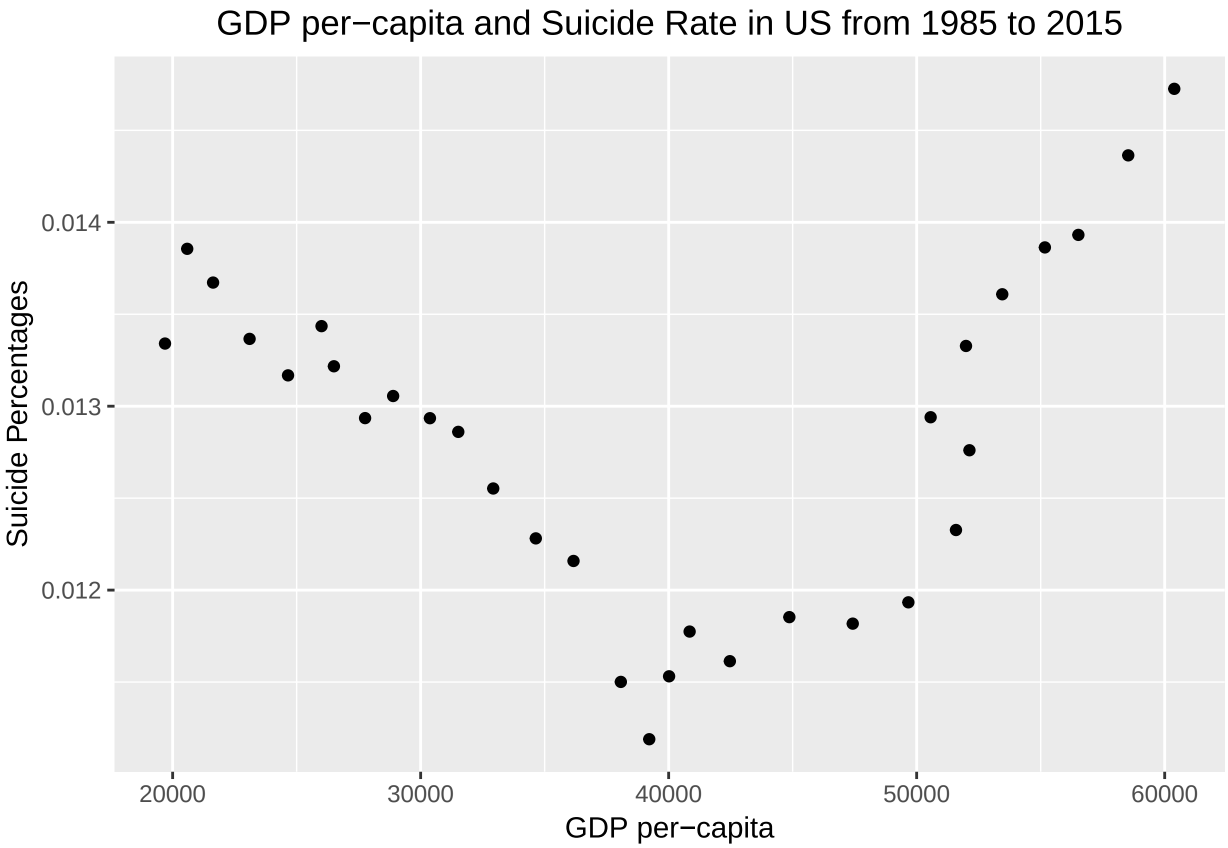
```
# Manipulating onlyUS with functions to create time series graph
suicide_no_sum_by_year <- aggregate(onlyUS$suicides_no,
                                    by = list(Category = onlyUS$year),
                                    FUN = sum)
population_sum_by_year <- aggregate(onlyUS$population,
                                    by = list(Category = onlyUS$year),
                                    FUN = sum)
year_suicide_perc <- suicide_no_sum_by_year
year_suicide_perc$suicide_perc <- (suicide_no_sum_by_year$x
                                   / population_sum_by_year$x) * 100
year_suicide_perc$year <- suicide_no_sum_by_year$Category

# Time series graph between the year and suicide rate
ggplot(year_suicide_perc, aes(x = year, y = suicide_perc, group = 1)) +
  geom_line() +
  geom_point() +
  ggtitle("Suicide Percentages in United States from 1985 to 2015") +
  ylab("Suicide Percentages") +
  theme(plot.title = element_text(hjust = 0.5))
```

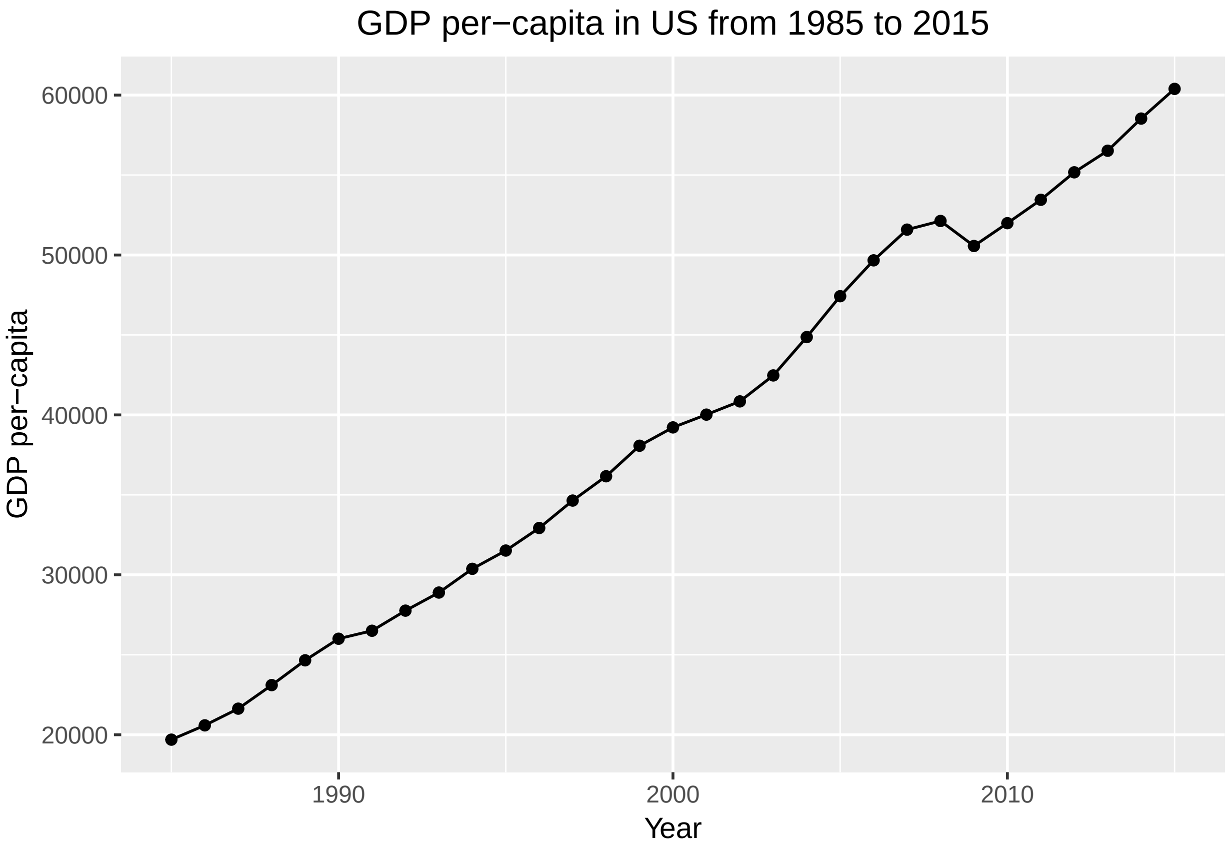


```r
year_gdp_per_capita <- aggregate(onlyUS$gdp_per_capita....,
                                 by = list(Category = onlyUS$year),
                                 FUN = sum)
year_gdp_per_capita$gdp_per_capita <- year_gdp_per_capita$x / 12
year_gdp_per_capita$suicide_perc <- year_suicide_perc$suicide_perc

# Scatterplot between GDP per-capita and suicide percentage
ggplot(data = year_gdp_per_capita,
       aes(x = gdp_per_capita, y = suicide_perc, group = 1)) +
  geom_point() +
  labs(title = 'GDP per-capita and Suicide Rate in US from 1985 to 2015') +
  xlab("GDP per-capita") +
  ylab("Suicide Percentages") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
# Line graph between year and GDP per-capita
ggplot(data = year_gdp_per_capita, aes(x = Category, y = gdp_per_capita, group = 1)) +
  geom_line() +
  geom_point() +
  labs(title='GDP per-capita in US from 1985 to 2015') +
  xlab("Year") +
  ylab("GDP per-capita") +
  theme(plot.title = element_text(hjust = 0.5))
```



```
head(onlyUS)
```

```
##   year    sex         age suicides_no population gdp_per_capita....
## 1 1985   male   75+ years        2177    4064000              19693
## 2 1985   male 55-74 years        5302   17971000              19693
## 3 1985   male 25-34 years        5134   20986000              19693
## 4 1985   male 35-54 years        6053   26589000              19693
## 5 1985   male 15-24 years        4267   19962000              19693
## 6 1985 female 35-54 years        2105   27763000              19693
##   suicide_perc
## 1  0.053567913
## 2  0.029503088
## 3  0.024463928
## 4  0.022765053
## 5  0.021375614
## 6  0.007582034
```

Appendix III: Splitting Data into Training and Testing

We will split our data set of 372 rows into training and test data. We will have approximately 80% of the data as “training” and 20% of the data as “test”. We will set a seed of 415.

```r
set.seed(415)

test_size = floor(nrow(onlyUS) * 0.2)
test_id = sample(1:nrow(onlyUS), test_size)

test_suicide = onlyUS[test_id, ]
train_suicide = onlyUS[-test_id, ]
```

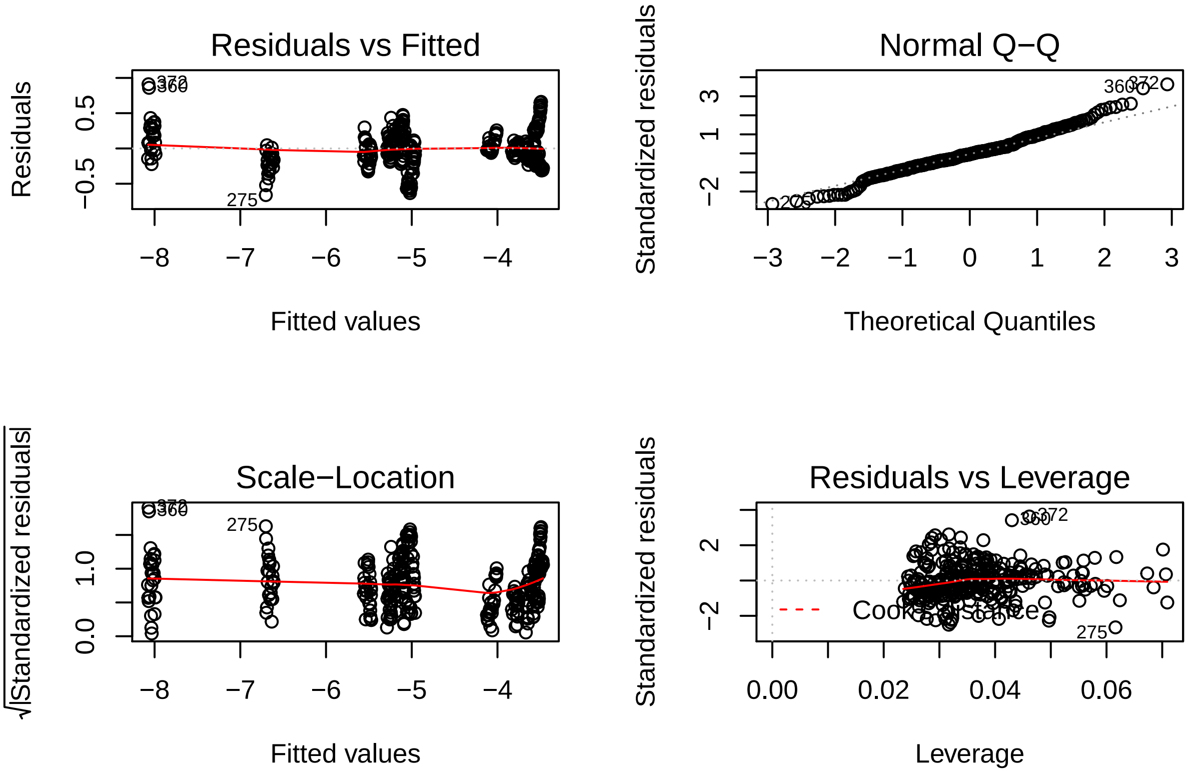
Appendix IV: Best Subset Selection Regression

Fitting a linear model on the training set (full model):

```
full = lm(log(suicide_perc) ~ ., data = train_suicide)
summary(full)
```

```
##
## Call:
## lm(formula = log(suicide_perc) ~ ., data = train_suicide)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66099 -0.15023 -0.01178  0.13373  0.91046
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -2.918e+01  4.141e+01  -0.705 0.481526
## year                1.205e-02  2.100e-02   0.574 0.566552
## sexmale             1.382e+00  5.523e-02  25.022  < 2e-16 ***
## age.L               1.931e+00  5.155e-02  37.470  < 2e-16 ***
## age.Q              -1.552e+00  1.048e-01 -14.816  < 2e-16 ***
## age.C               7.801e-01  8.204e-02   9.509  < 2e-16 ***
## age^4              -3.107e-01  4.081e-02  -7.613 3.89e-13 ***
## age^5               2.608e-01  6.914e-02   3.772 0.000197 ***
## suicides_no         2.542e-05  1.328e-05   1.913 0.056683 .
## population         -1.049e-08  6.674e-09  -1.571 0.117175
## gdp_per_capita.... -9.892e-06  1.516e-05  -0.652 0.514656
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2571 on 287 degrees of freedom
## Multiple R-squared:  0.965,  Adjusted R-squared:  0.9638
## F-statistic: 792.3 on 10 and 287 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(full)
```



```
train_mse_full = mean(full$residuals^2)
test_mse_full = mean((log(test_suicide$suicide_perc) -
                        predict(full, test_suicide))^2)
```

Forward Selection and Backward Selection

```
# Forward
train_suicide_2 = train_suicide
train_suicide_2$suicide_perc = log(train_suicide_2$suicide_perc)
forward = SignifReg(suicide_perc ~ ., train_suicide_2, alpha = 0.05,
                    direction = 'forward', correction = 'None', trace = FALSE)
```

```
##
## Call:
## lm(formula = reg, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74526 -0.14337 -0.00328  0.13121  0.92248
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.466e+00  1.057e-01 -51.690  < 2e-16 ***
## age.L        1.977e+00  3.992e-02  49.534  < 2e-16 ***
## age.Q       -1.627e+00  8.077e-02 -20.143  < 2e-16 ***
## age.C        7.628e-01  6.556e-02  11.635  < 2e-16 ***
## age^4       -3.036e-01  3.917e-02  -7.751 1.55e-13 ***
## age^5        2.931e-01  5.727e-02   5.119 5.62e-07 ***
## sexmale      1.468e+00  3.028e-02  48.472  < 2e-16 ***
## population  -9.236e-09  4.663e-09  -1.981   0.0486 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2578 on 290 degrees of freedom
## Multiple R-squared:  0.9645, Adjusted R-squared:  0.9636
## F-statistic:  1125 on 7 and 290 DF,  p-value: < 2.2e-16
```

```
train_mse_fwd = mean(forward$residuals^2)
test_mse_fwd = mean((log(test_suicide$suicide_perc) -
                      predict(forward, test_suicide))^2)

# Backward
backward = SignifReg(log(suicide_perc) ~ ., train_suicide, alpha = 0.05,
                    direction = 'backward', correction = 'None', trace = FALSE)
```

```
##
## Call:
## lm(formula = reg, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74526 -0.14337 -0.00328  0.13121  0.92248
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.466e+00  1.057e-01 -51.690  < 2e-16 ***
## sexmale      1.468e+00  3.028e-02  48.472  < 2e-16 ***
## age.L        1.977e+00  3.992e-02  49.534  < 2e-16 ***
## age.Q       -1.627e+00  8.077e-02 -20.143  < 2e-16 ***
## age.C        7.628e-01  6.556e-02  11.635  < 2e-16 ***
## age^4       -3.036e-01  3.917e-02  -7.751 1.55e-13 ***
## age^5        2.931e-01  5.727e-02   5.119 5.62e-07 ***
## population  -9.236e-09  4.663e-09  -1.981   0.0486 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2578 on 290 degrees of freedom
## Multiple R-squared:  0.9645, Adjusted R-squared:  0.9636
## F-statistic:  1125 on 7 and 290 DF,  p-value: < 2.2e-16
```

```
train_mse_bwd = mean(backward$residuals^2)
test_mse_bwd = mean((log(test_suicide$suicide_perc) -
                      predict(backward, test_suicide))^2)
```

AIC, BIC, Adjusted R-squared

```
regfit_full = regsubsets(log(suicide_perc) ~ ., data = train_suicide,
                        nvmax = NULL)
regfit_summary = summary(regfit_full)

# AIC
coef(regfit_full, which.min(regfit_summary$cp))
```

```
##   (Intercept)       sexmale         age.L         age.Q         age.C
## -5.402615e+00  1.378340e+00  1.920467e+00 -1.596810e+00  7.463289e-01
##         age^4         age^5   suicides_no    population
## -3.027955e-01  2.871239e-01  2.581215e-05 -1.342287e-08
```

```
AIC_fit = lm(log(suicide_perc) ~ sex + age + suicides_no + population,
             data = train_suicide)
summary(AIC_fit)
```

```
##
## Call:
## lm(formula = log(suicide_perc) ~ sex + age + suicides_no + population,
##     data = train_suicide)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69282 -0.14632 -0.00975  0.13421  0.88340
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.403e+00  1.101e-01 -49.064  < 2e-16 ***
## sexmale      1.378e+00  5.495e-02  25.083  < 2e-16 ***
## age.L        1.920e+00  4.931e-02  38.950  < 2e-16 ***
## age.Q       -1.597e+00  8.185e-02 -19.510  < 2e-16 ***
## age.C        7.463e-01  6.579e-02  11.344  < 2e-16 ***
## age^4       -3.028e-01  3.898e-02  -7.767 1.41e-13 ***
## age^5        2.871e-01  5.708e-02   5.031 8.60e-07 ***
## suicides_no  2.581e-05  1.325e-05   1.948  0.05241 .
## population  -1.342e-08  5.114e-09  -2.625  0.00914 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2566 on 289 degrees of freedom
## Multiple R-squared:  0.9649, Adjusted R-squared:  0.964
## F-statistic: 994.1 on 8 and 289 DF,  p-value: < 2.2e-16
```

```
train_mse_AIC = mean((AIC_fit$residuals)^2)
test_mse_AIC = mean((log(test_suicide$suicide_perc) -
                      predict(AIC_fit, test_suicide))^2)

# BIC
coef(regfit_full, which.min(regfit_summary$bic))
```

```
## (Intercept)     sexmale       age.L       age.Q       age.C       age^4
##  -5.6710673   1.4771966   2.0073343  -1.4841348   0.8700820  -0.3298728
##       age^5
##   0.2051714
```

```
BIC_fit = lm(log(suicide_perc) ~ sex + age, data = train_suicide)
summary(BIC_fit)
```

```
##
## Call:
## lm(formula = log(suicide_perc) ~ sex + age, data = train_suicide)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76063 -0.15115 -0.01369  0.13311  0.91939
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -5.67107    0.02097 -270.383  < 2e-16 ***
## sexmale      1.47720    0.03006   49.139  < 2e-16 ***
## age.L        2.00733    0.03712   54.074  < 2e-16 ***
## age.Q       -1.48413    0.03666  -40.483  < 2e-16 ***
## age.C        0.87008    0.03711   23.447  < 2e-16 ***
## age^4       -0.32987    0.03704   -8.906  < 2e-16 ***
## age^5        0.20517    0.03635    5.645 3.92e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2591 on 291 degrees of freedom
## Multiple R-squared:  0.964,  Adjusted R-squared:  0.9633
## F-statistic:  1299 on 6 and 291 DF,  p-value: < 2.2e-16
```

```r
train_mse_BIC = mean((BIC_fit$residuals)^2)
test_mse_BIC = mean((log(test_suicide$suicide_perc) -
                       predict(BIC_fit, test_suicide))^2)

# Adjusted R-squared
coef(regfit_full, which.max(regfit_summary$adjr2))
```

```
##   (Intercept)       sexmale         age.L         age.Q         age.C
## -5.402615e+00  1.378340e+00  1.920467e+00 -1.596810e+00  7.463289e-01
##         age^4         age^5   suicides_no    population
## -3.027955e-01  2.871239e-01  2.581215e-05 -1.342287e-08
```

```r
adjr2_fit = lm(log(suicide_perc) ~ sex + age + suicides_no + population,
               data = train_suicide)
summary(adjr2_fit)
```

```
##
## Call:
## lm(formula = log(suicide_perc) ~ sex + age + suicides_no + population,
##     data = train_suicide)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69282 -0.14632 -0.00975  0.13421  0.88340
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.403e+00  1.101e-01 -49.064  < 2e-16 ***
## sexmale      1.378e+00  5.495e-02  25.083  < 2e-16 ***
## age.L        1.920e+00  4.931e-02  38.950  < 2e-16 ***
## age.Q       -1.597e+00  8.185e-02 -19.510  < 2e-16 ***
## age.C        7.463e-01  6.579e-02  11.344  < 2e-16 ***
## age^4       -3.028e-01  3.898e-02  -7.767 1.41e-13 ***
## age^5        2.871e-01  5.708e-02   5.031 8.60e-07 ***
## suicides_no  2.581e-05  1.325e-05   1.948  0.05241 .
## population  -1.342e-08  5.114e-09  -2.625  0.00914 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2566 on 289 degrees of freedom
## Multiple R-squared:  0.9649, Adjusted R-squared:  0.964
## F-statistic: 994.1 on 8 and 289 DF,  p-value: < 2.2e-16
```

```
train_mse_adjr2 = mean((adjr2_fit$residuals)^2)
test_mse_adjr2 = mean((log(test_suicide$suicide_perc) -
                         predict(adjr2_fit, test_suicide))^2)
```

Cross-Validation Error

```
glm_full = glm(full)
cv_mse_full = cv.glm(train_suicide, glm_full, K = 5)$delta[1]

glm_forward = glm(forward)
cv_mse_fwd = cv.glm(train_suicide_2, glm_forward, K = 5)$delta[1]

glm_backward = glm(backward)
cv_mse_bwd = cv.glm(train_suicide, glm_backward, K = 5)$delta[1]

glm_AIC = glm(AIC_fit)
cv_mse_aic = cv.glm(train_suicide, glm_AIC, K = 5)$delta[1]

glm_BIC = glm(BIC_fit)
cv_mse_bic = cv.glm(train_suicide, glm_BIC, K = 5)$delta[1]

glm_adjr2 = glm(adjr2_fit)
cv_mse_adjr2 = cv.glm(train_suicide, glm_adjr2, K = 5)$delta[1]
```

Table of Errors (Linear Methods)

```
models_linear = c("Full", "Forward", "Backward", "AIC", "BIC", "Adjusted-R^2")

train_err_linear = c(
  train_mse_full,
  train_mse_fwd,
  train_mse_bwd,
  train_mse_AIC,
  train_mse_BIC,
  train_mse_adjr2
)

test_err_linear = c(
  test_mse_full,
  test_mse_fwd,
  test_mse_bwd,
  test_mse_AIC,
  test_mse_BIC,
  test_mse_adjr2
)

cv_err_linear = c(
  cv_mse_full,
  cv_mse_fwd,
  cv_mse_bwd,
  cv_mse_aic,
  cv_mse_bic,
  cv_mse_adjr2
)

results_linear = data.frame(
  models_linear,
  train_err_linear,
  test_err_linear,
  cv_err_linear
)

colnames(results_linear) = c("Methods", "Train Error", "Test Error", "CV Error")
knitr::kable(results_linear)
```

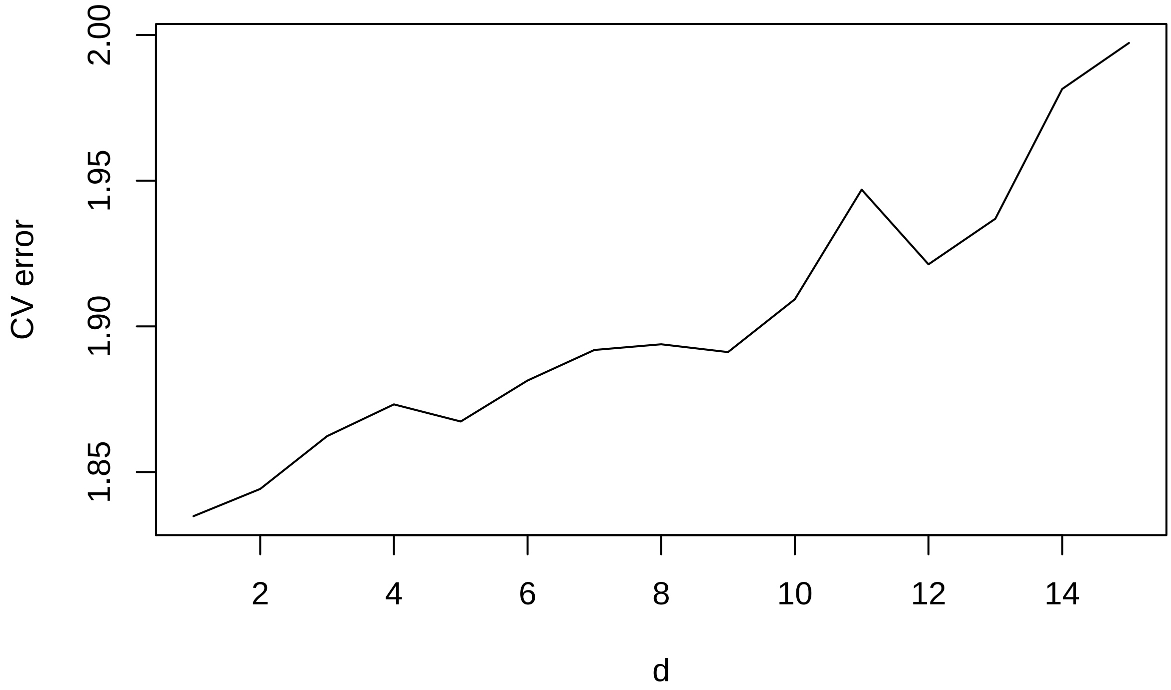
Methods	Train Error	Test Error	CV Error
Full	0.0636563	0.0568892	0.0678483
Forward	0.0646916	0.0605790	0.0702350
Backward	0.0646916	0.0605790	0.0700189
AIC	0.0638534	0.0572245	0.0666711
BIC	0.0655668	0.0623151	0.0687811
Adjusted-R^2	0.0638534	0.0572245	0.0662563

Appendix V: Non-Linear Methods

Polynomial regression

```
cv.error_poly = rep(0, 15)
for(i in 1:15) {
  fitpoly = glm(log(suicide_perc) ~ poly(gdp_per_capita...., i), data = train_suicide)
  cv.error_poly[i] = cv.glm(train_suicide, fitpoly, K = 10)$delta[1]
}

plot(1:15, cv.error_poly, xlab = "d", ylab = "CV error", type = "l")
```



```
which.min(cv.error_poly)
```

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

```
fit_poly = lm(log(suicide_perc) ~ poly(gdp_per_capita...., 1), data = train_suicide)
summary(fit_poly)
```

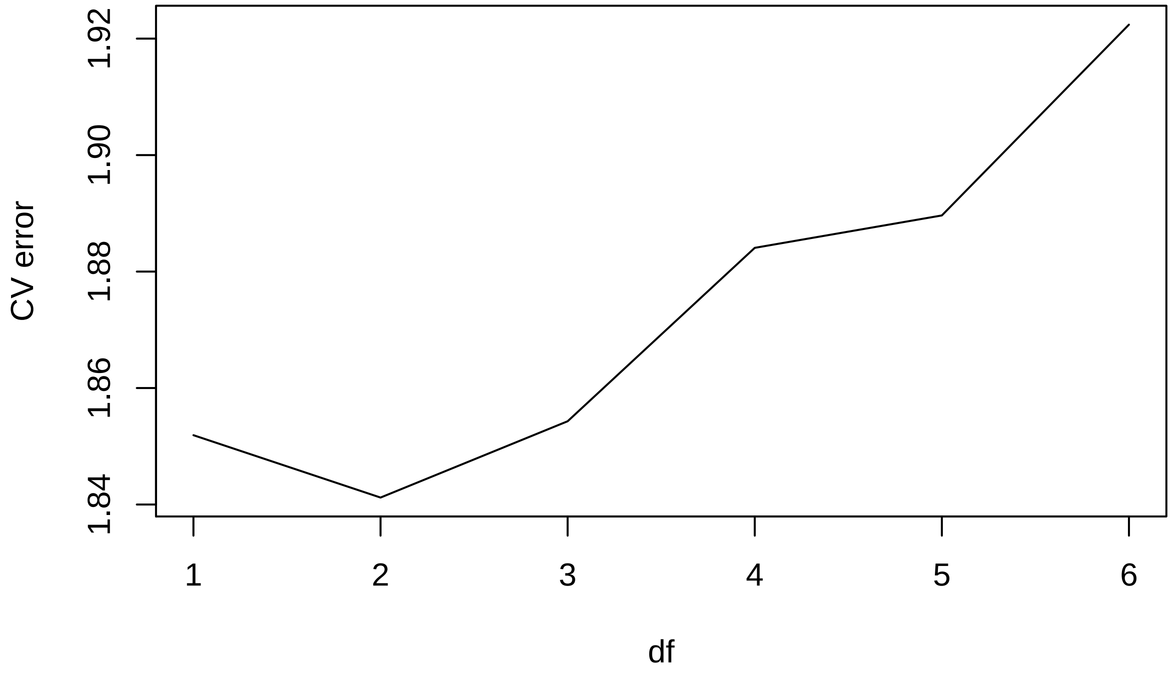
```
##
## Call:
## lm(formula = log(suicide_perc) ~ poly(gdp_per_capita...., 1),
##     data = train_suicide)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3185 -0.5784 -0.0610  1.1565  2.0438
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -4.94521    0.07839 -63.087   <2e-16 ***
## poly(gdp_per_capita...., 1) -0.80968    1.35318  -0.598     0.55
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.353 on 296 degrees of freedom
## Multiple R-squared:  0.001208,   Adjusted R-squared:  -0.002166
## F-statistic: 0.358 on 1 and 296 DF,  p-value: 0.5501
```

Splines

Natural Splines

```
cv.error_ns = rep(0, 6)
for (i in 1:6) {
  fitns = glm(log(suicide_perc) ~ ns(gdp_per_capita...., df = i), data = train_suicide)
  cv.error_ns[i] = cv.glm(train_suicide, fitns, K = 10)$delta[1]
}

plot(1:6, cv.error_ns, xlab = "df", ylab = "CV error", type = "l")
```



```
which.min(cv.error_ns)
```

```
## [1] 2
```

```
fit_ns = lm(log(suicide_perc) ~ ns(gdp_per_capita...., df = 2), data = train_suicide)
summary(fit_ns)
```

```
##
## Call:
## lm(formula = log(suicide_perc) ~ ns(gdp_per_capita...., df = 2),
##     data = train_suicide)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max 
## -3.3626 -0.6053 -0.0983  1.1954  1.9916 
## 
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                      -4.6487     0.2108 -22.048   <2e-16 ***
## ns(gdp_per_capita...., df = 2)1  -0.7331     0.4785  -1.532    0.127    
## ns(gdp_per_capita...., df = 2)2   0.1376     0.2705   0.509    0.611    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.35 on 295 degrees of freedom
## Multiple R-squared:  0.008695,   Adjusted R-squared:  0.001975 
## F-statistic: 1.294 on 2 and 295 DF,  p-value: 0.2758
```

Smoothing Spline

```
fit_ss = smooth.spline(x = train_suicide$gdp_per_capita....,
                      y = log(train_suicide$suicide_perc), cv = T)
fit_ss
```

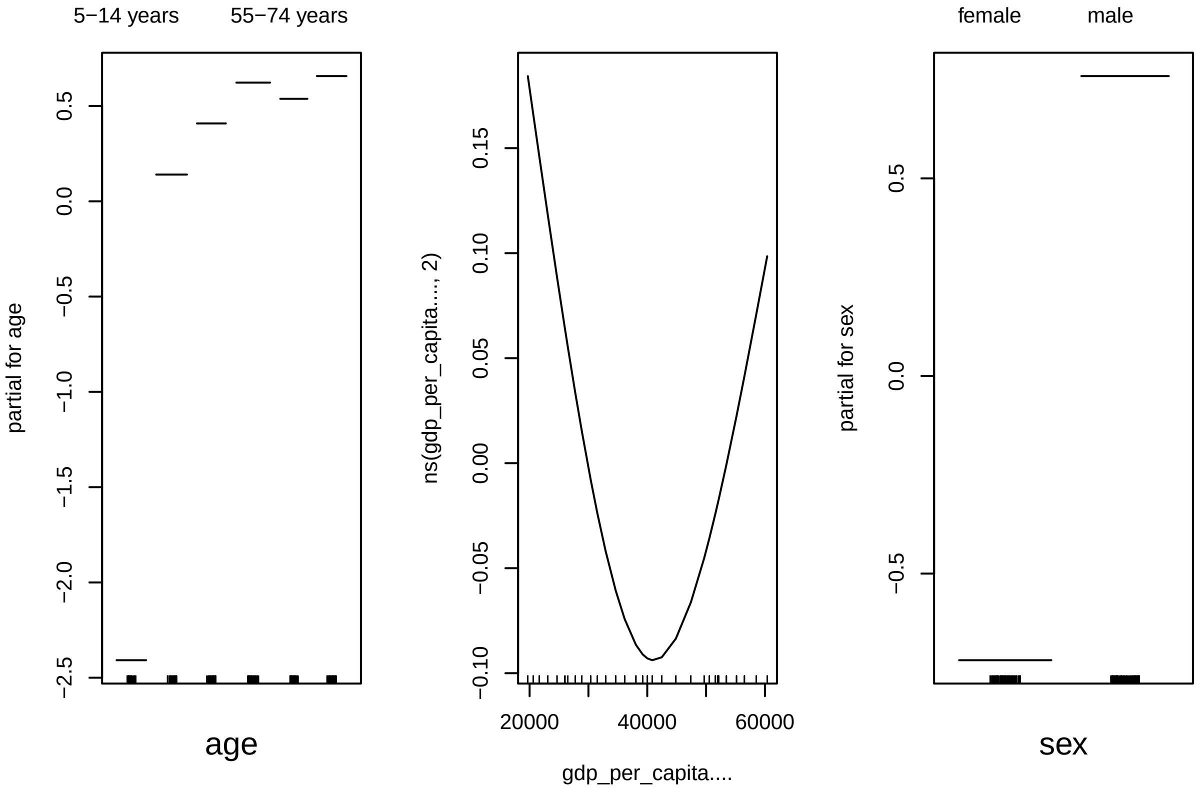
```
## Call:
## smooth.spline(x = train_suicide$gdp_per_capita...., y = log(train_suicide$suicide_perc), 
##     cv = T)
## 
## Smoothing Parameter  spar= 0.8847244  lambda= 0.2101054 (13 iterations)
## Equivalent Degrees of Freedom (Df): 3.200752
## Penalized Criterion (RSS): 10.79945
## PRESS(l.o.o. CV): 1.843534
```

GAM

```
fit_gam_1 <- lm(log(suicide_perc) ~ age + ns(gdp_per_capita...., 2) + sex, data = train_suicide)
summary(fit_gam_1)
```

```
## 
## Call:
## lm(formula = log(suicide_perc) ~ age + ns(gdp_per_capita...., 
##     2) + sex, data = train_suicide)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.74491 -0.10412 -0.01632  0.12074  0.81333 
## 
## Coefficients:
##                            Estimate Std. Error  t value Pr(>|t|)    
## (Intercept)                -5.48709    0.04121 -133.152  < 2e-16 ***
## age.L                       1.99978    0.03550   56.334  < 2e-16 ***
## age.Q                      -1.47920    0.03503  -42.223  < 2e-16 ***
## age.C                       0.87098    0.03545   24.569  < 2e-16 ***
## age^4                      -0.32546    0.03539   -9.196  < 2e-16 ***
## age^5                       0.20272    0.03474    5.835 1.44e-08 ***
## ns(gdp_per_capita...., 2)1 -0.45589    0.08780   -5.193 3.92e-07 ***
## ns(gdp_per_capita...., 2)2  0.08828    0.04968    1.777   0.0766 .
## sexmale                     1.47791    0.02873   51.438  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2475 on 289 degrees of freedom
## Multiple R-squared:  0.9674, Adjusted R-squared:  0.9665
## F-statistic:  1071 on 8 and 289 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(1, 3))
plot.Gam(fit_gam_1)
```



```
fit_gam_2 <- lm(log(suicide_perc) ~ ns(gdp_per_capita...., 2) + age + sex + age * sex, data = train_sui
summary(fit_gam_2)
```

```
##
## Call:
## lm(formula = log(suicide_perc) ~ ns(gdp_per_capita...., 2) +
##     age + sex + age * sex, data = train_suicide)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47640 -0.07891  0.00396  0.08581  0.55166
##
## Coefficients:
##                            Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                -5.48472    0.02258 -242.945  < 2e-16 ***
## ns(gdp_per_capita...., 2)1 -0.46985    0.04812   -9.764  < 2e-16 ***
## ns(gdp_per_capita...., 2)2  0.11250    0.02715    4.143 4.52e-05 ***
## age.L                       1.67257    0.02697   62.010  < 2e-16 ***
## age.Q                      -1.62770    0.02661  -61.169  < 2e-16 ***
## age.C                       0.54529    0.02706   20.148  < 2e-16 ***
## age^4                      -0.26567    0.02713   -9.792  < 2e-16 ***
## age^5                       0.23472    0.02652    8.850  < 2e-16 ***
## sexmale                     1.48401    0.01573   94.355  < 2e-16 ***
## age.L:sexmale               0.68043    0.03879   17.542  < 2e-16 ***
## age.Q:sexmale               0.28381    0.03831    7.408 1.47e-12 ***
## age.C:sexmale               0.66943    0.03877   17.266  < 2e-16 ***
## age^4:sexmale              -0.12306    0.03880   -3.171  0.00168 **
## age^5:sexmale              -0.05228    0.03801   -1.375  0.17007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1352 on 284 degrees of freedom
## Multiple R-squared:  0.9904, Adjusted R-squared:   0.99
## F-statistic:  2262 on 13 and 284 DF,  p-value: < 2.2e-16
```

Table of Errors (Non-Linear Methods)

```r
fit.poly = lm(log(suicide_perc) ~ poly(gdp_per_capita...., 1), data = train_suicide)
train_mse_poly = mean((predict(fit.poly, train_suicide) - log(train_suicide$suicide_perc))^2)
test_mse_poly = mean((predict(fit.poly, test_suicide) - log(test_suicide$suicide_perc))^2)

fit.ns = glm(log(suicide_perc) ~ ns(gdp_per_capita...., df = 2), data = train_suicide)
train_mse_ns = mean((predict(fit.ns, train_suicide) - log(train_suicide$suicide_perc))^2)
test_mse_ns = mean((predict(fit.ns, test_suicide) - log(test_suicide$suicide_perc))^2)

fit.ss = smooth.spline(x = train_suicide$gdp_per_capita....,
                       y = log(train_suicide$suicide_perc), df = 3.2)
train_mse_ss = mean((predict(fit.ss, x = train_suicide$gdp_per_capita....)$y
                     - log(train_suicide$suicide_perc))^2)
test_mse_ss = mean((predict(fit.ss, x = test_suicide$gdp_per_capita....)$y
                    - log(test_suicide$suicide_perc))^2)

train_mse_gam_1 = mean((predict(fit_gam_1, train_suicide) - log(train_suicide$suicide_perc))^2)
test_mse_gam_1 = mean((predict(fit_gam_1, test_suicide) - log(test_suicide$suicide_perc))^2)

train_mse_gam_2 = mean((predict(fit_gam_2, train_suicide) - log(train_suicide$suicide_perc))^2)
test_mse_gam_2 = mean((predict(fit_gam_2, test_suicide) - log(test_suicide$suicide_perc))^2)

models_nonlinear = c("Polynomial", "Natural Spline", "Smoothing Spline",
                     "GAM (without interaction)", "GAM (with interaction)")

train_err_nonlinear = c(
  train_mse_poly,
  train_mse_ns,
  train_mse_ss,
  train_mse_gam_1,
  train_mse_gam_2
)

test_err_nonlinear = c(
  test_mse_poly,
  test_mse_ns,
  test_mse_ss,
  test_mse_gam_1,
  test_mse_gam_2
)

cv_err_nonlinear = c(
  round(cv.error_poly[1], 7),
  round(cv.error_ns[2], 7),
  round(fit_ss$cv.crit, 7),
  "-",
  "-"
)

results_nonlinear = data.frame(
  models_nonlinear,
  train_err_nonlinear,
  test_err_nonlinear,
  cv_err_nonlinear
)

colnames(results_nonlinear) = c("Methods", "Train Error", "Test Error", "CV Error")
knitr::kable(results_nonlinear)
```

Methods	Train Error	Test Error	CV Error
Polynomial	1.8188110	1.8053070	1.8348424
Natural Spline	1.8051766	1.8355548	1.8411967
Smoothing Spline	1.8057784	1.8277272	1.8435343
GAM (without interaction)	0.0594228	0.0511489	-
GAM (with interaction)	0.0174200	0.0171541	-