##Loading packages

```r
#install.packages("remotes")
#remotes::install_github("zeeva85/gapminderplus")
library(gapminderplus)
library(ggplot2)
library(tidyr)
library(corrplot)
library(car)
library(boot)
library(class)
library(caret)
set.seed(42)
```

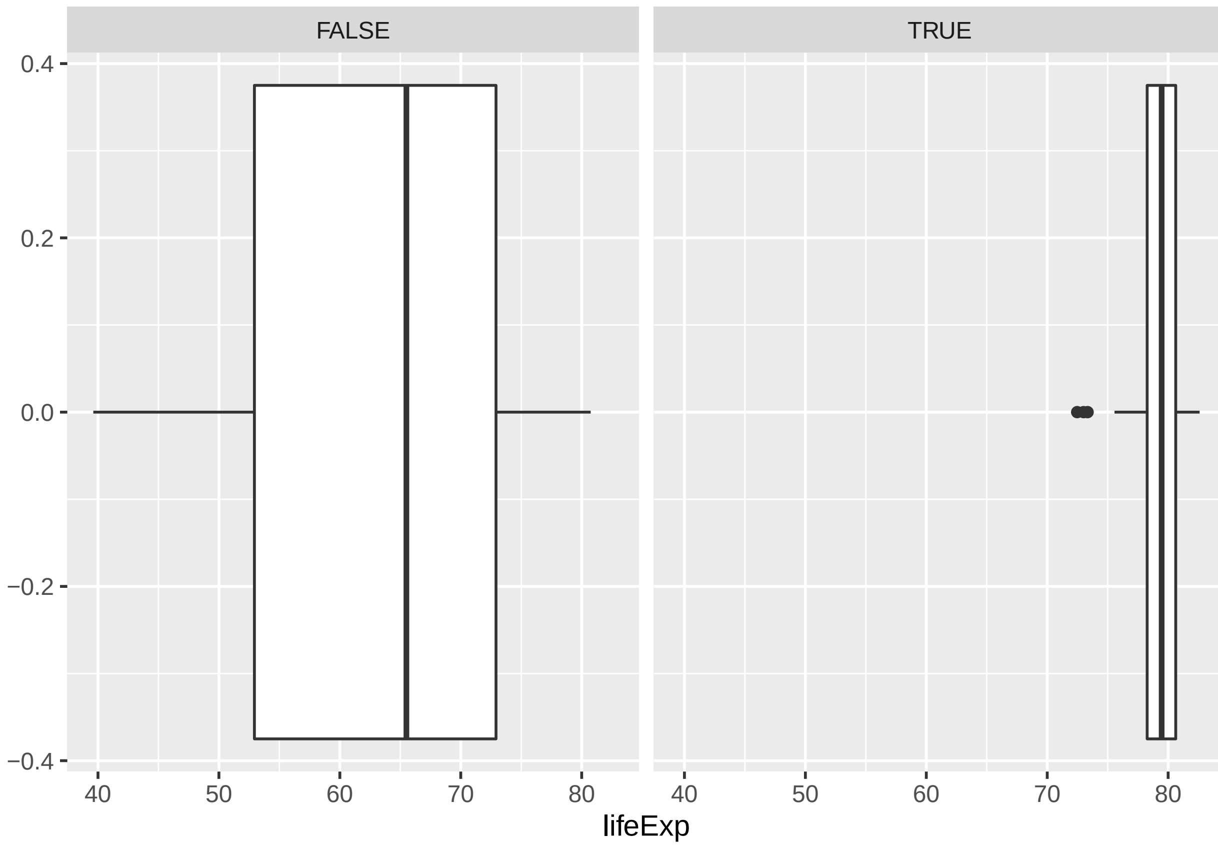
##Loading and processing dataframe

```r
raw_df <- gapminder3
df2007 <- raw_df[raw_df$year==2007,]

#List of develeped countries from 2008 economic report
developed_countries <- c("United States", "Canada", "Japan", "Australia", "New Zealand", "Austria", "Bel

df2007$developed <-F
isDeveloped <- df2007$country %in% developed_countries
df2007[isDeveloped,]$developed <- T
df2007 <- df2007[complete.cases(df2007),]
sum(df2007$developed)
```

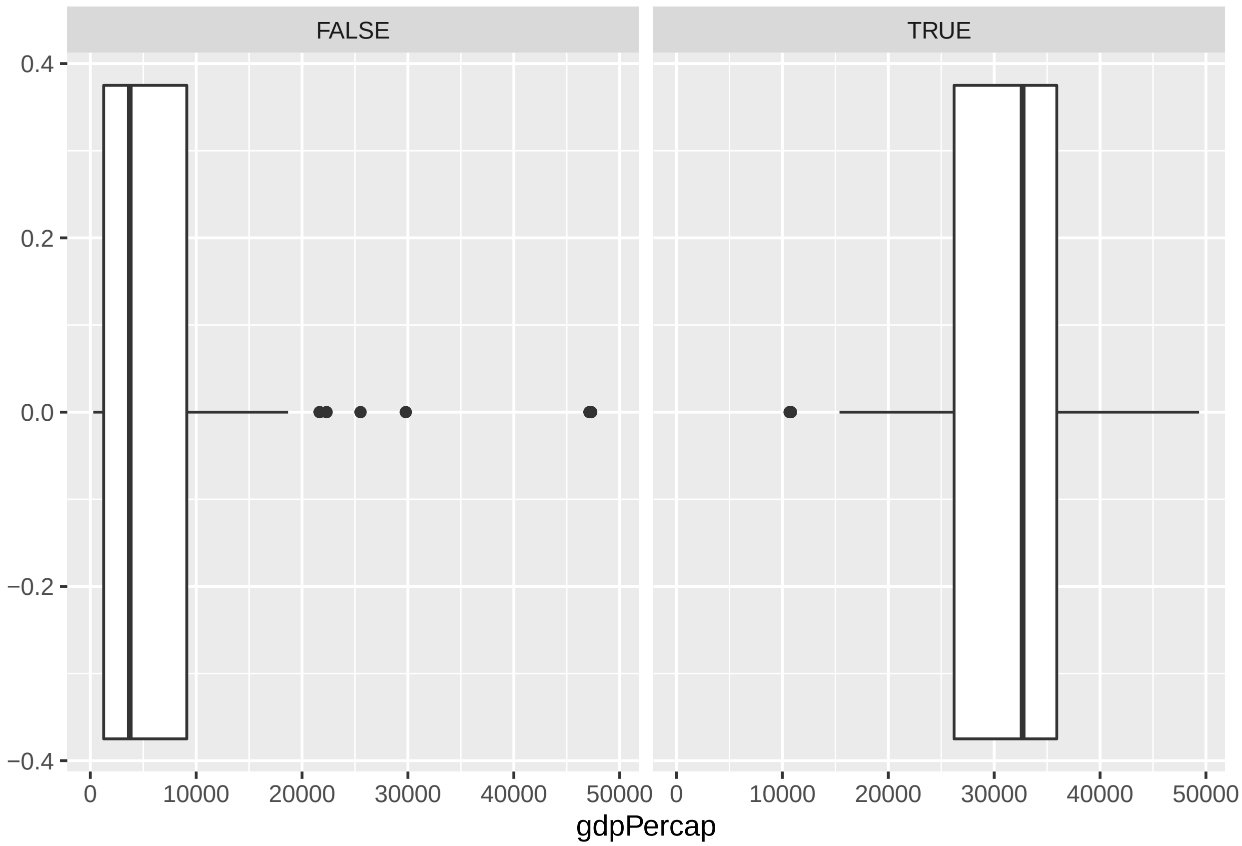
```
## [1] 26
```

###Basic plots

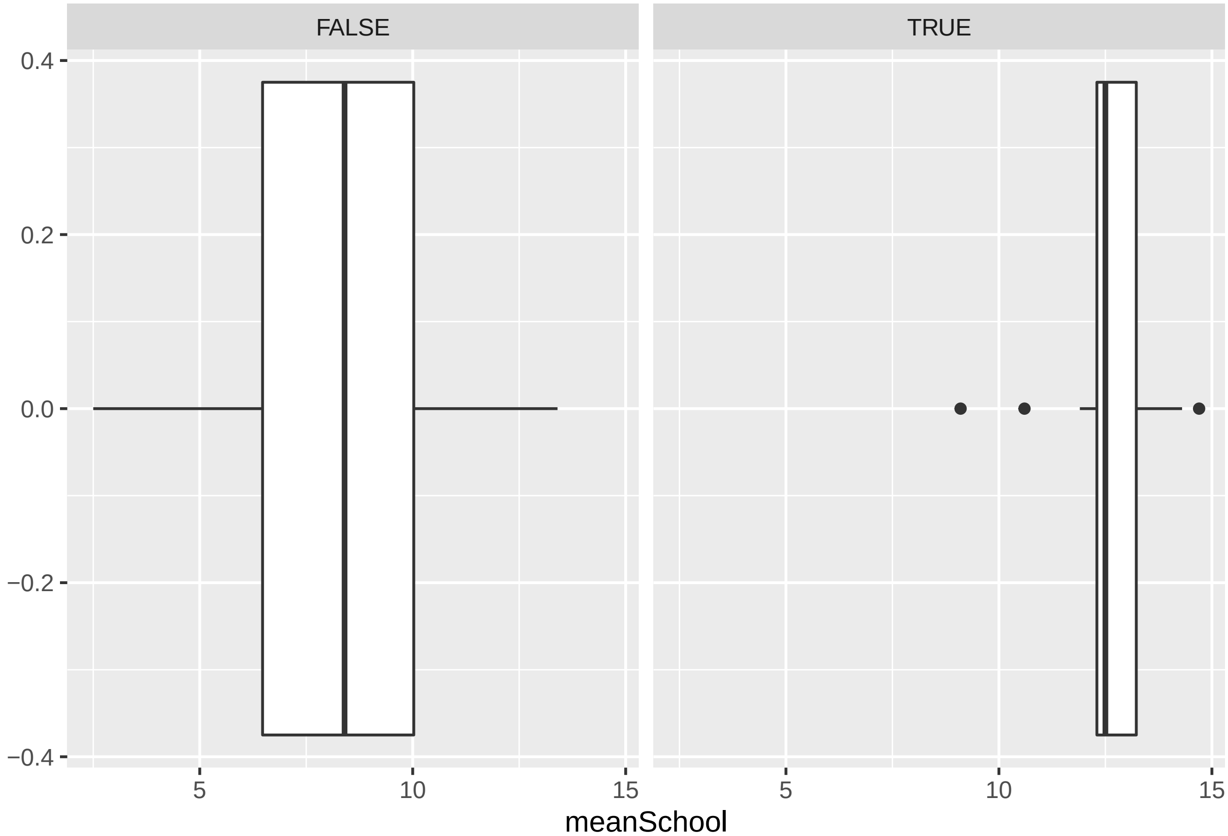
```r
ggplot(df2007, aes(lifeExp)) + geom_boxplot() + facet_wrap(~developed)
```



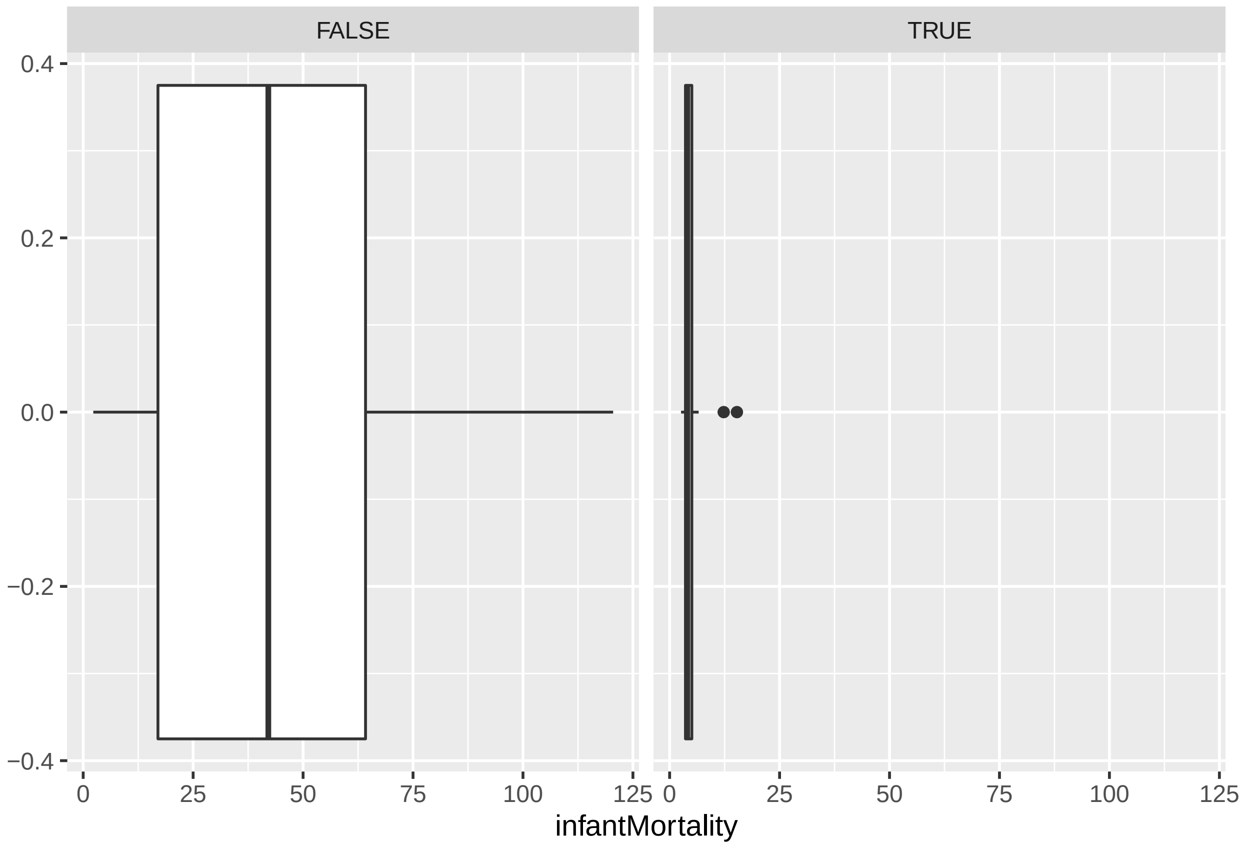

```
ggplot(df2007, aes(gdpPercap)) + geom_boxplot() + facet_wrap(~developed)
```




```
ggplot(df2007, aes(meanSchool)) + geom_boxplot() + facet_wrap(~developed)
```

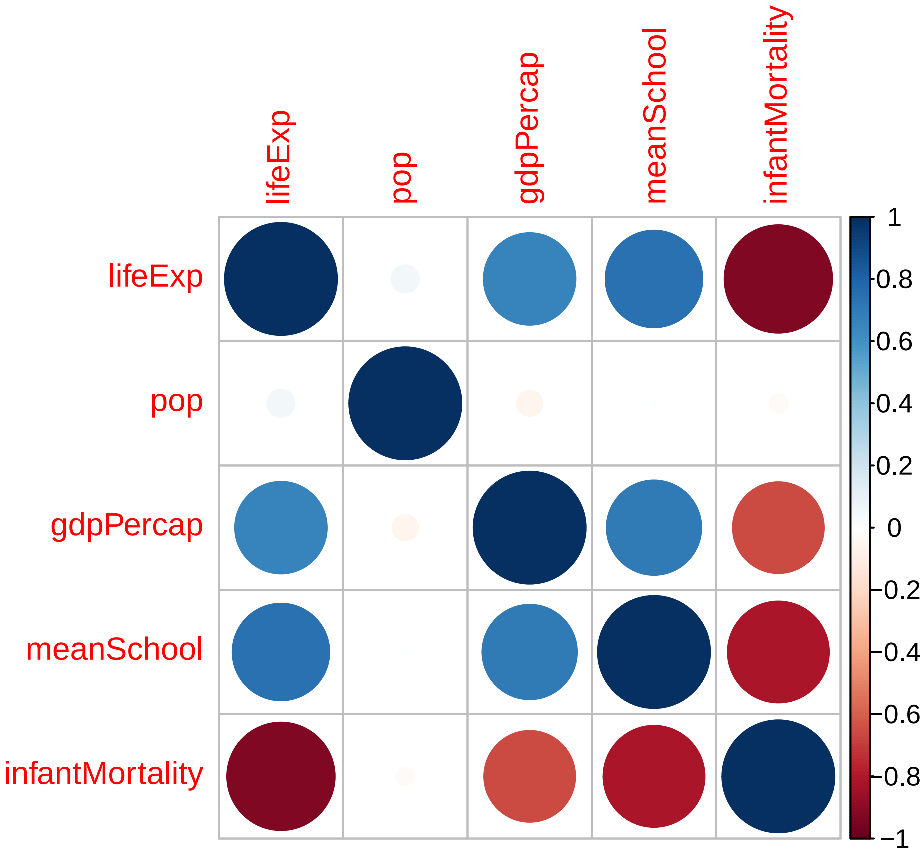



```
ggplot(df2007, aes(infantMortality)) + geom_boxplot() + facet_wrap(~developed)
```



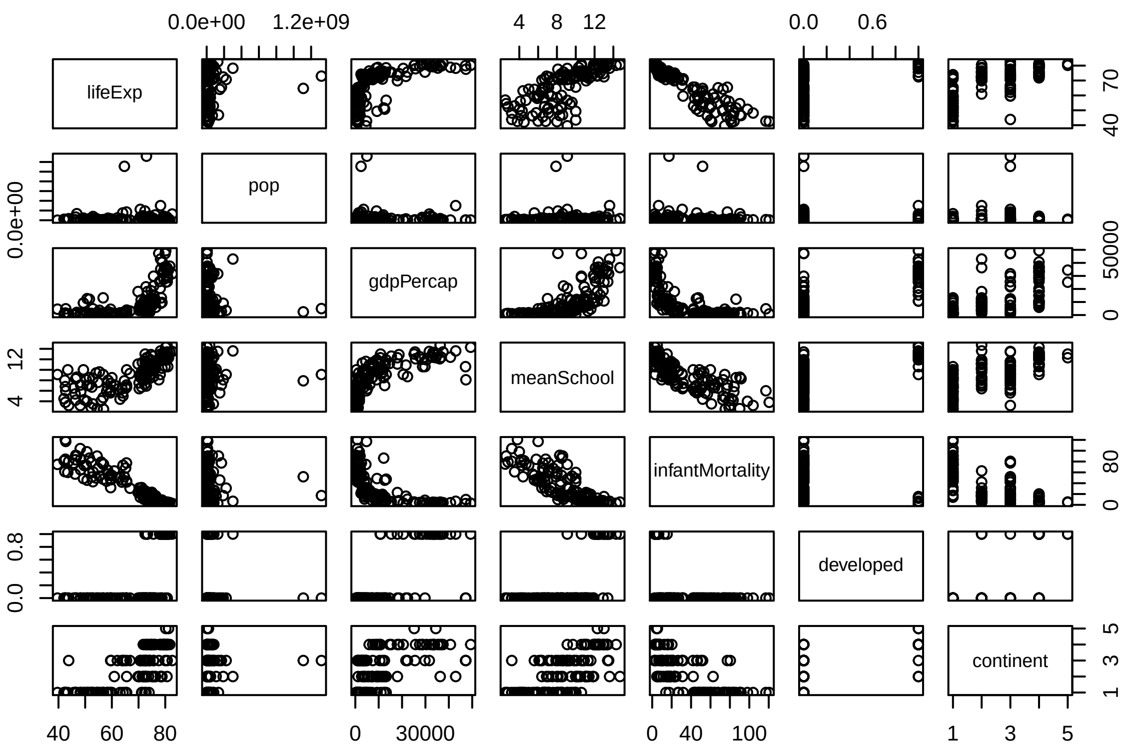
###Correlation plot

```
analysis_df2007 <-df2007[,c("lifeExp", "pop", "gdpPercap", "meanSchool", "infantMortality", "developed"
corrplot(cor(df2007[,c("lifeExp", "pop", "gdpPercap", "meanSchool","infantMortality")]))
```



###Pairwise scatterplot

```
pairs(df2007[,c("lifeExp", "pop", "gdpPercap", "meanSchool","infantMortality","developed",  "continent"
```



###First model

```r
log_reg_all <- glm(developed ~ ., data = analysis_df2007, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(log_reg_all)
```

```
##
## Call:
## glm(formula = developed ~ ., family = "binomial", data = analysis_df2007)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2898  -0.0017   0.0000   0.0000   1.4439
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -7.932e+01  4.199e+03  -0.019   0.9849
## lifeExp             5.905e-01  4.699e-01   1.257   0.2089
## pop                 9.989e-09  6.857e-09   1.457   0.1452
## gdpPercap           2.605e-04  1.693e-04   1.538   0.1240
## meanSchool          1.225e+00  6.155e-01   1.990   0.0466 *
## infantMortality     1.389e-01  2.270e-01   0.612   0.5408
## continentAmericas   8.230e+00  4.199e+03   0.002   0.9984
## continentAsia       5.021e+00  4.199e+03   0.001   0.9990
## continentEurope     1.597e+01  4.199e+03   0.004   0.9970
## continentOceania    3.060e+01  3.315e+04   0.001   0.9993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 130.105  on 129  degrees of freedom
## Residual deviance:  14.881  on 120  degrees of freedom
## AIC: 34.881
##
## Number of Fisher Scoring iterations: 21
```

testing collinearity

```r
vif(log_reg_all)
```

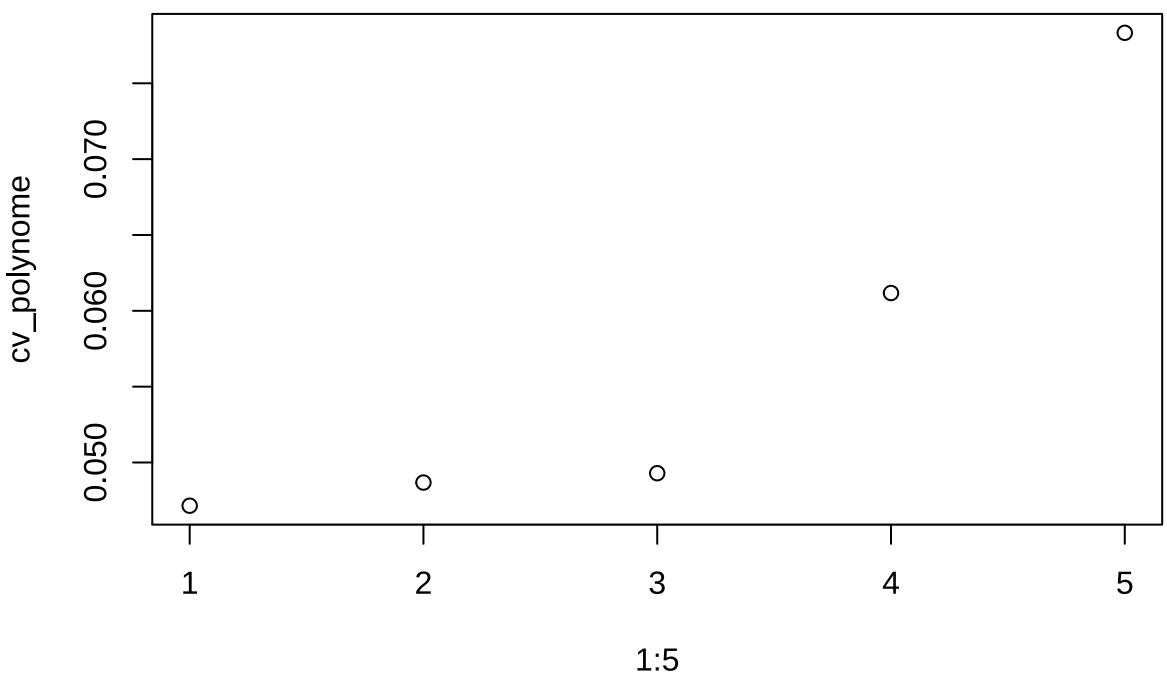
```
##                      GVIF Df GVIF^(1/(2*Df))
## lifeExp          4.940097  1        2.222633
## pop              3.061405  1        1.749687
## gdpPercap        7.885455  1        2.808105
## meanSchool       1.600270  1        1.265018
## infantMortality  3.163629  1        1.778659
## continent       12.249127  4        1.367770
```

```r
analysis_df2007 <-df2007[,c("meanSchool","gdpPercap", "developed")]
log_reg1 <- glm(developed ~ ., data = analysis_df2007, family = "binomial")

summary(log_reg1)
```

```
##
```

```
## Call:
## glm(formula = developed ~ ., family = "binomial", data = analysis_df2007)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.32624  -0.15328  -0.02499  -0.00187   2.76825
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.863e+01  4.745e+00  -3.927 8.62e-05 ***
## meanSchool   1.360e+00  3.973e-01   3.423 0.000618 ***
## gdpPercap    1.192e-04  3.825e-05   3.116 0.001833 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 130.105  on 129  degrees of freedom
## Residual deviance:  37.086  on 127  degrees of freedom
## AIC: 43.086
##
## Number of Fisher Scoring iterations: 8
```

```
cv_polynome <- 1:5
for (i in 1:5) {
  model <- glm(developed ~ poly(gdpPercap,i) + poly(meanSchool,i) , data = analysis_df2007, family = "bi
  cv_polynome[i] <- cv.glm(analysis_df2007, model)$delta[1]
}

plot(x=1:5, cv_polynome)
```



Probabilities of correct result

```r
summary(predict(log_reg1, type = "response"))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000003 0.0001203 0.0063407 0.2000000 0.1948820 0.9987726
```

Since data is not even I chose a threshold of 0.2

```r
pred_glm <- predict(log_reg1, type = "response") > 0.2
table(pred_glm, Real = analysis_df2007[,3])
```

```
##          Real
## pred_glm FALSE TRUE
##    FALSE    97    1
##    TRUE      7   25
```

```r
Result <- ifelse(pred_glm, T, F)
mean(Result == df2007[, "developed"])
```

```
## [1] 0.9384615
```

##KKN

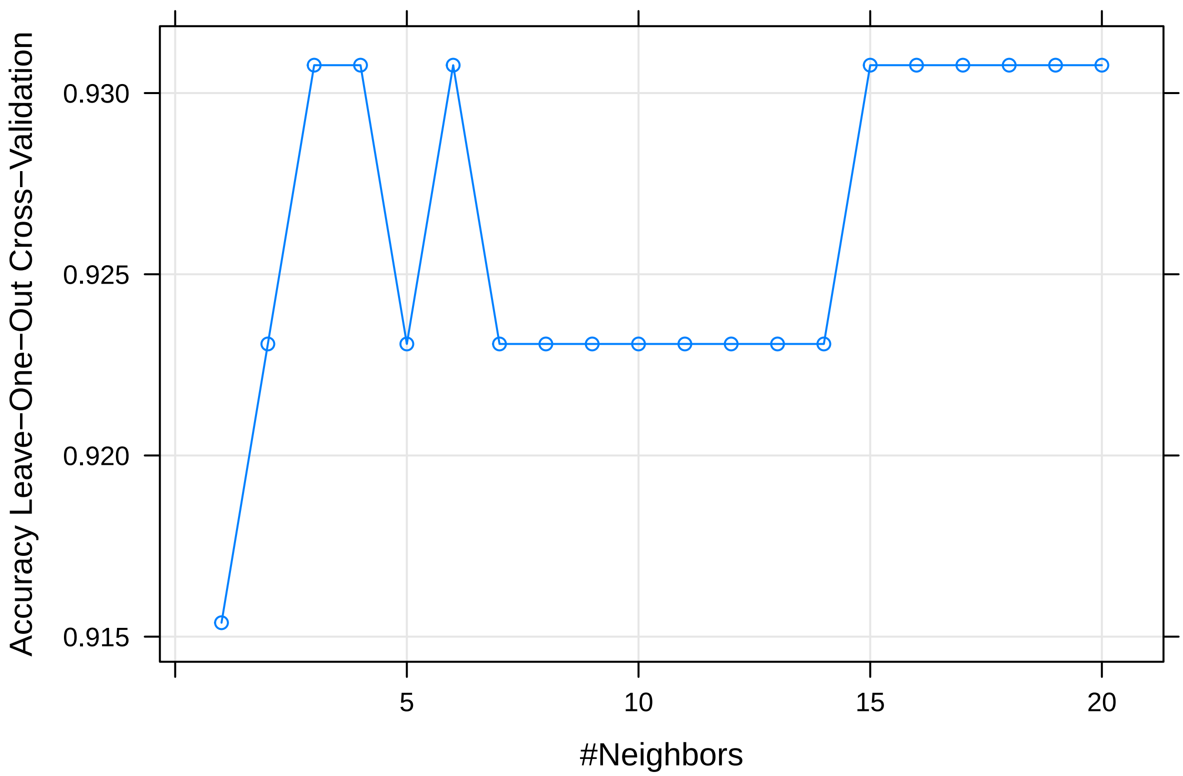
```r
df2007$developed <-0
isDeveloped <- df2007$country %in% developed_countries
df2007[isDeveloped,]$developed <- 1
df2007 <- df2007[complete.cases(df2007),]
df2007$developed <- factor(df2007$developed)
analysis_df2007 <-df2007[,c("lifeExp", "gdpPercap", "meanSchool", "infantMortality", "developed")]


test <- sample(nrow(analysis_df2007), floor(0.40*nrow(analysis_df2007)))

test_df <- analysis_df2007[test,]
train_df <- analysis_df2007[-test,]
```

```r
grid_knn <- expand.grid(k=1:20)

kkn_model <- train(developed ~ ., data =analysis_df2007, method = "knn", tuneGrid = grid_knn,
                   trControl = trainControl(method = "LOOCV"), preProcess = c("center", "scale"), metric

kkn_model
```

```
## k-Nearest Neighbors
##
## 130 samples
##   4 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (4), scaled (4)
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 129, 129, 129, 129, 129, 129, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.9153846  0.7393365
##    2  0.9230769  0.7663551
##    3  0.9307692  0.7867299
```

```
##     4  0.9307692  0.7804878
##     5  0.9230769  0.7596154
##     6  0.9307692  0.7867299
##     7  0.9230769  0.7596154
##     8  0.9230769  0.7596154
##     9  0.9230769  0.7596154
##    10  0.9230769  0.7596154
##    11  0.9230769  0.7596154
##    12  0.9230769  0.7596154
##    13  0.9230769  0.7596154
##    14  0.9230769  0.7596154
##    15  0.9307692  0.7804878
##    16  0.9307692  0.7804878
##    17  0.9307692  0.7804878
##    18  0.9307692  0.7804878
##    19  0.9307692  0.7804878
##    20  0.9307692  0.7804878
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 20.
```

```r
plot(kkn_model)
```



Testing

```r
knn_model_test <- knn(train = train_df, test = test_df, cl = train_df[,"developed"], k = 2)

table(knn_model_test, Real = test_df[,"developed"])
```

```
##                 Real
## knn_model_test  0  1
```

```
##                  0 38  2
##                  1  5  7
```