World Bank

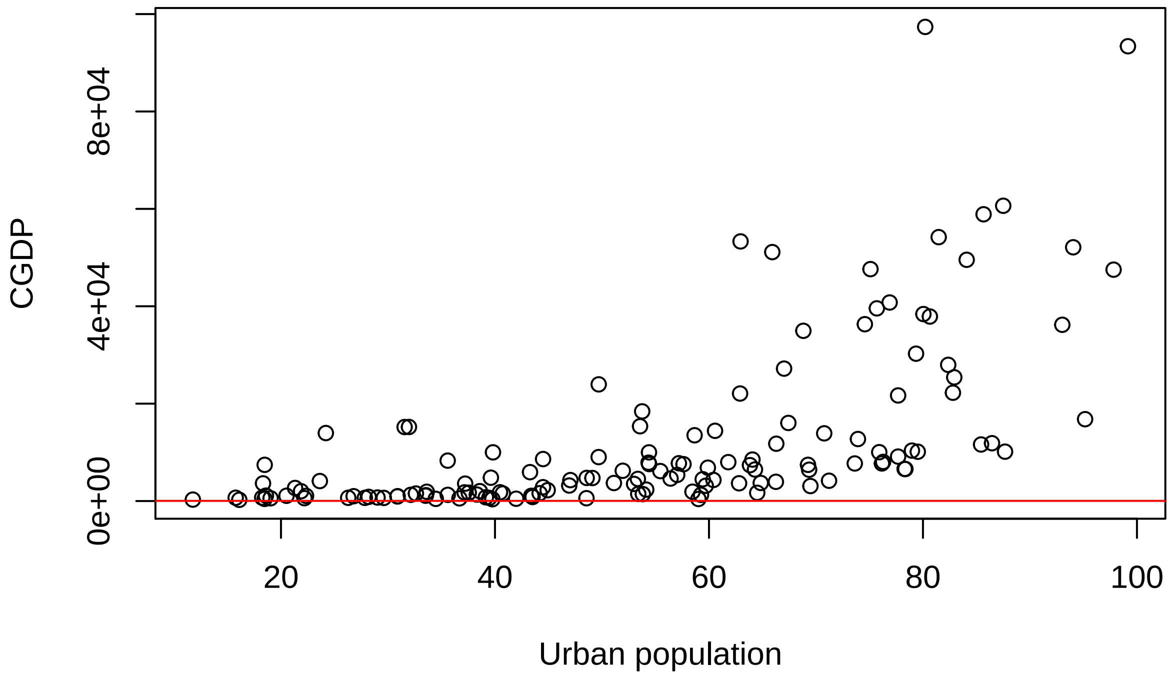
Introduction

The objective of this project was analyzing the data of the World Bank to predict the urban population in the data set using three different regression models and analyzing which one would predict the urban population accurately.

Linear Model

My first analysis was plotting urban population as a function of cgdp. I then constucted a linear model but the results were barely acceptable.

```
# Plot urb_pop as function of cgdp
plot(world_bank_train$urb_pop , world_bank_train$cgdp , xlab = "Urban population" , ylab = "CGDP")

# Set up a linear model between the two variables
lm_wb = lm(urb_pop ~ cgdp , data= world_bank_train)

# Add a red regression line to your scatter plot
abline(lm_wb$coefficient , col = "red")
```



```
# Summarize lm_wb and select R-squared
summary(lm_wb)$r.squared
```

```
## [1] 0.3822067
```

In the previous model, the scatter plot didn’t show a strong linear relationship. This was confirmed with the regression line and R^2

To improve the linear model we need to study the nature of the data. The predictor variable is numerical, while the response variable is expressed in percentiles. It would make more sense if there were a linear relationship between the percentile changes of the GDP / capita and the changes in the response.

To obtain an estimation of percentile changes, we need to take the natural logarithm of the GDP / capita and use this as your new predictor variable. world_bank_test set is used to check if the model generalizes well.

Log-Linear Model

```r
# Build the log-linear model
lm_wb_log <- lm(urb_pop ~ log(cgdp), data = world_bank_train)

# Calculate rmse_train
rmse_train <- sqrt(mean(lm_wb_log$residuals ^ 2))

# Summarize lm_wb and select R-squared
summary(lm_wb_log)$r.squared
```

```
## [1] 0.5787588
```

```r
# The real percentage of urban population in the test set
world_bank_test_truth <- world_bank_test$urb_pop

# The predictions of the percentage of urban population in the test set
world_bank_test_input <- data.frame(cgdp = world_bank_test$cgdp)
world_bank_test_output <- predict(lm_wb_log, world_bank_test_input)

# The residuals: the difference between the ground truth and the predictions
res_test <- world_bank_test_output - world_bank_test_truth


# Use res_test to calculate rmse_test
rmse_test = sqrt(sum((res_test)^2)/nrow(world_bank_test))

# Print the ratio of the test RMSE over the training RMSE
rmse_test/rmse_train
```

```
## [1] 1.08308
```

Linear model v/s Log-Linear model

Prediction differed substantially between these two models. R^2 value for Log-Linear model is greater than the R^2 value of Linear model. Therefore Log-Linear model gives better prediction as its explained variance is the highest.

k-NN regression model

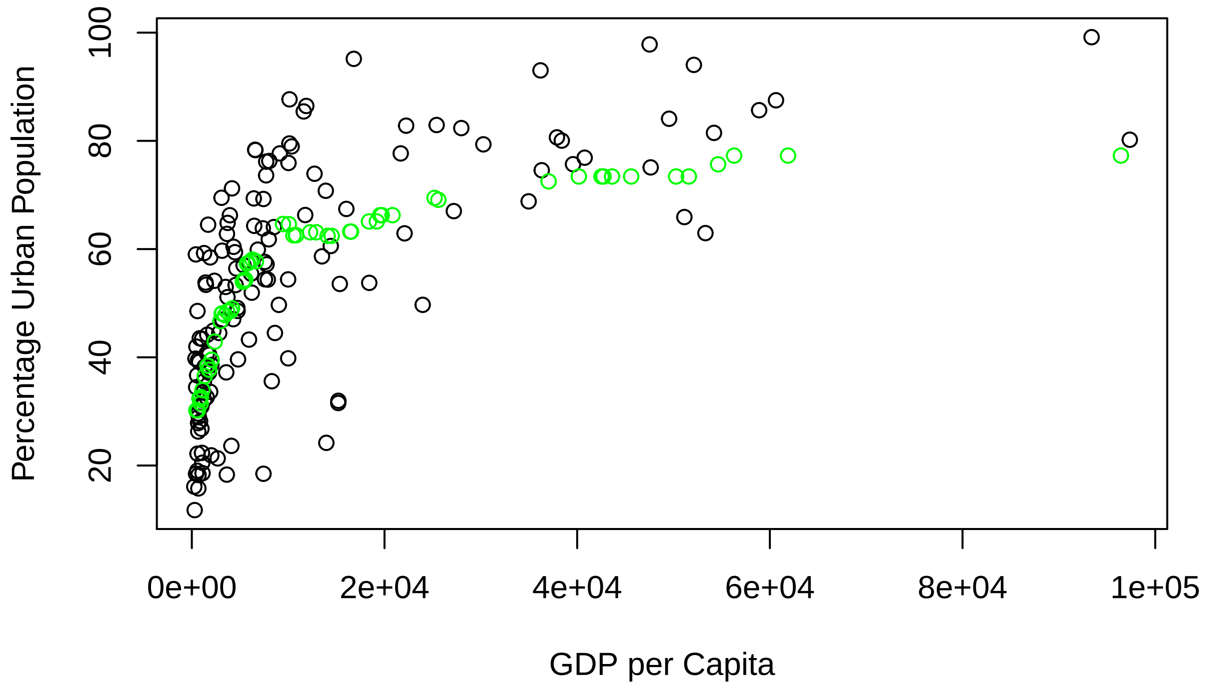
The function my_knn contains a k-NN algorithm.

It’s arguments are:

```
                x_pred : predictor values of the new observations
                x : predictor values of the training set
                y: corresponding response values of the training set
                k: the number of neighbors
```

The function returns the predicted values for your new observations (predict_knn).

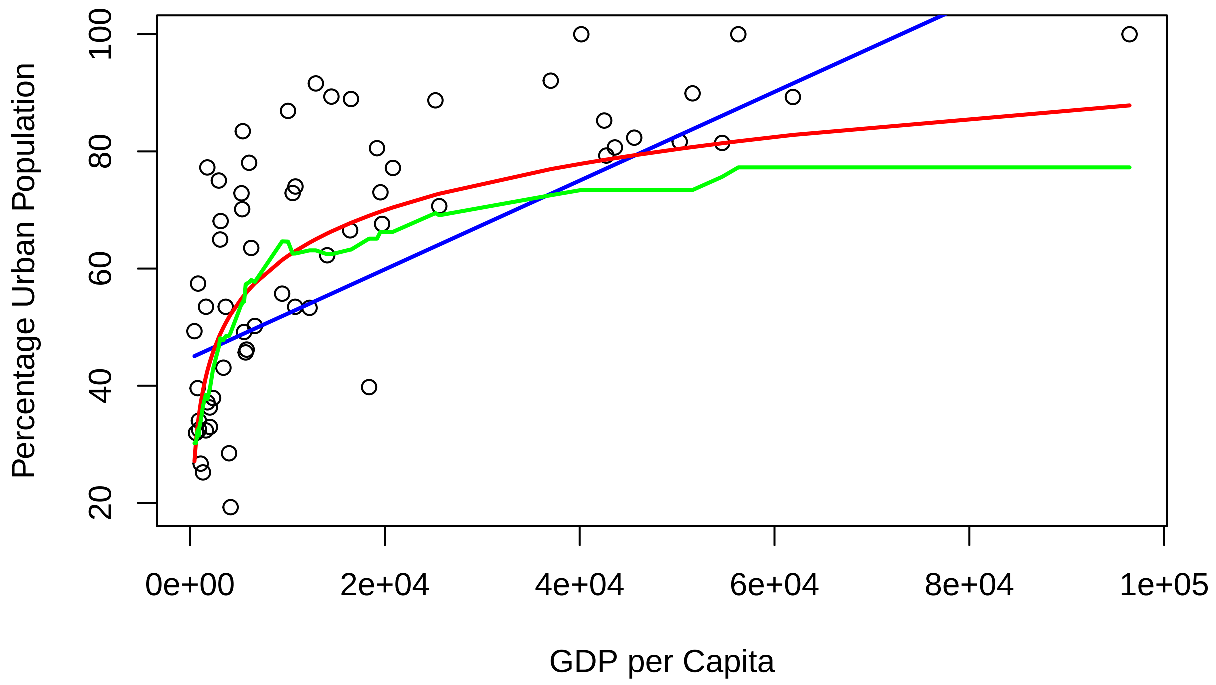
```r
my_knn <- function(x_pred, x, y, k){
  m <- length(x_pred)
  predict_knn <- rep(0, m)
  for (i in 1:m) {

    # Calculate the absolute distance between x_pred[i] and x
    dist <- abs(x_pred[i] - x)

    # Apply order() to dist, sort_index will contain
    # the indices of elements in the dist vector, in
    # ascending order. This means sort_index[1:k] will
    # return the indices of the k-nearest neighbors.
    sort_index <- order(dist)

    # Apply mean() to the responses of the k-nearest neighbors
    predict_knn[i] <- mean(y[sort_index[1:k]])

  }
  return(predict_knn)
}
###

# Applied alogrithm on the test set
test_output = my_knn(world_bank_test$cgdp ,world_bank_train$cgdp , world_bank_train$urb_pop , 30)

# Plot of the output
plot(world_bank_train,
     xlab = "GDP per Capita",
     ylab = "Percentage Urban Population")
points(world_bank_test$cgdp, test_output, col = "green")
```



Linear model v/s Log-Linear model v/s k-NN Regression model

Compare the RMSE of the three models to see which will give the best predictions.

```r
# Define ranks to order the predictor variables in the test set
ranks <- order(world_bank_test$cgdp)

# Scatter plot of test set
plot(world_bank_test,
     xlab = "GDP per Capita", ylab = "Percentage Urban Population")

# Predict with simple linear model and add line
test_output_lm <- predict(lm_wb, data.frame(cgdp = world_bank_test$cgdp))
lines(world_bank_test$cgdp[ranks], test_output_lm[ranks], lwd = 2, col = "blue")

# Predict with log-linear model and add line
test_output_lm_log <- predict(lm_wb_log, data.frame(cgdp = world_bank_test$cgdp))
lines(world_bank_test$cgdp[ranks], test_output_lm_log[ranks], lwd = 2, col = "red")

# Predict with k-NN and add line
test_output_knn <- my_knn(world_bank_test$cgdp, world_bank_train$cgdp, world_bank_train$urb_pop, 30)
lines(world_bank_test$cgdp[ranks], test_output_knn[ranks], lwd = 2, col = "green")
```



```
# Calculate RMSE for simple linear model
sqrt(mean( (test_output_lm - world_bank_test$urb_pop) ^ 2))
```

```
## [1] 17.41897
```

```
# Calculate RMSE for log-linear model
sqrt(mean( (test_output_lm_log - world_bank_test$urb_pop) ^ 2))
```

```
## [1] 15.01911
```

```
# Calculate RMSE for k-NN technique
sqrt(mean( (test_output_knn - world_bank_test$urb_pop) ^ 2))
```

```
## [1] 16.10026
```

In conclusion the Log-Linear model gives the best RMSE value out of all the three models and thereby giving the most accurate prediction.