# Data Analysis with R — C997

Carl Andrew Perkins

Western Governors University

Data Analysis with R — C997

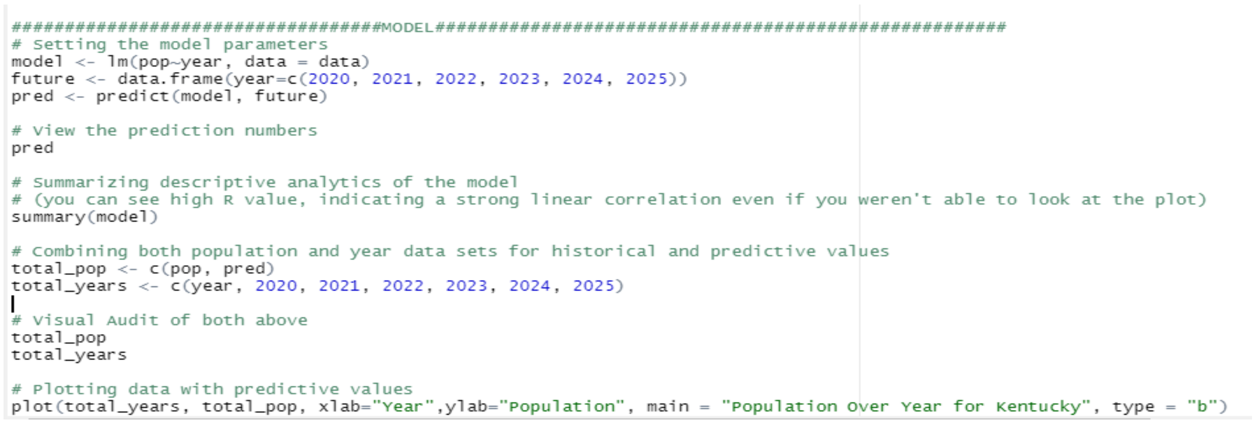
Below is the process and deliverables by which I imported, scrubbed and created the model in R according to the task's request.

**A Create a linear regression analysis with R to predict the size of the population for the state you live in based on the "Current Estimates Data" dataset (see weblink below). Provide a screenshot of your results.**

nst-est2019-alldata.xlsx

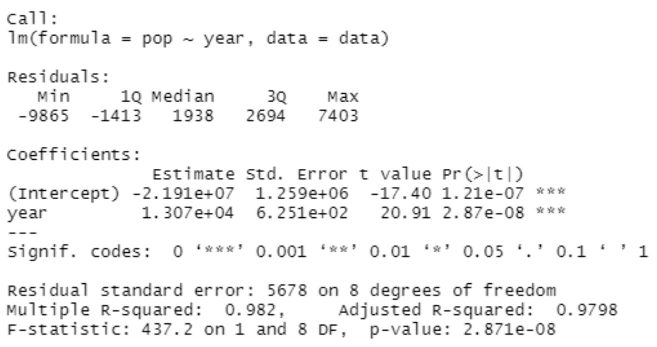
Current Estimates

I have given screenshots below of the linear regression analysis. Note, this was the actual end deliverable and not the beginning of work.



```
###################################MODEL######################################################
# Setting the model parameters
model <- lm(pop~year, data = data)
future <- data.frame(year=c(2020, 2021, 2022, 2023, 2024, 2025))
pred <- predict(model, future)

# View the prediction numbers
pred

# Summarizing descriptive analytics of the model
# (you can see high R value, indicating a strong linear correlation even if you weren't able to look at the plot)
summary(model)

# Combining both population and year data sets for historical and predictive values
total_pop <- c(pop, pred)
total_years <- c(year, 2020, 2021, 2022, 2023, 2024, 2025)

# Visual Audit of both above
total_pop
total_years

# Plotting data with predictive values
plot(total_years, total_pop, xlab="Year",ylab="Population", main = "Population Over Year for Kentucky", type = "b")
```

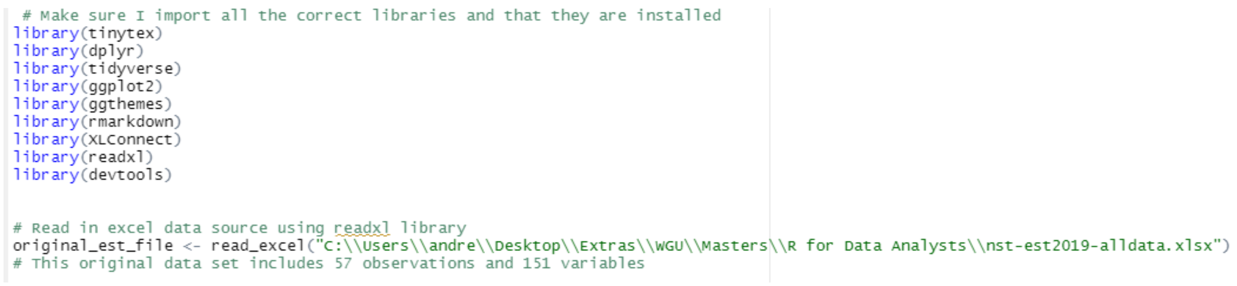
**Console Results**



```
Call:
lm(formula = pop ~ year, data = data)

Residuals:
   Min     1Q Median     3Q    Max 
 -9865  -1413   1938   2694   7403 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.191e+07  1.259e+06  -17.40 1.21e-07 ***
year         1.307e+04  6.251e+02   20.91 2.87e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5678 on 8 degrees of freedom
Multiple R-squared:  0.982,	Adjusted R-squared:  0.9798 
F-statistic: 437.2 on 1 and 8 DF,  p-value: 2.871e-08
```

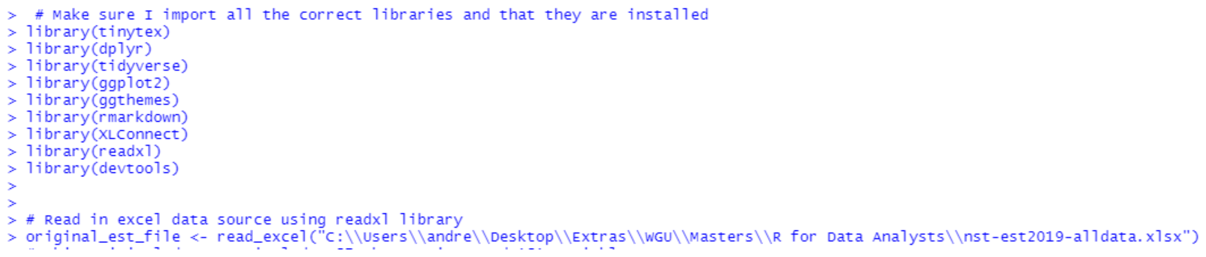
**B. Explain how you prepared the data from part A and how the dataset was imported into R, including screenshots of your results.**

The first step was to import libraries like with any R scripts. After the libraries were imported, I preceded to import the excel data via the readxl library



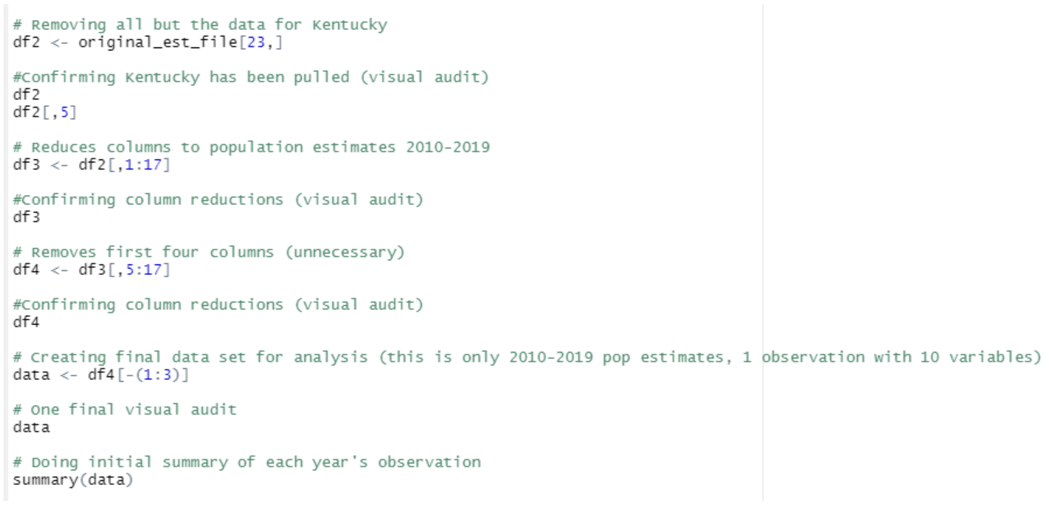
```
 # Make sure I import all the correct libraries and that they are installed
library(tinytex)
library(dplyr)
library(tidyverse)
library(ggplot2)
library(ggthemes)
library(rmarkdown)
library(XLConnect)
library(readxl)
library(devtools)


# Read in excel data source using readxl library
original_est_file <- read_excel("C:\\Users\\andre\\Desktop\\Extras\\WGU\\Masters\\R for Data Analysts\\nst-est2019-alldata.xlsx")
# This original data set includes 57 observations and 151 variables
```

**Console Results**



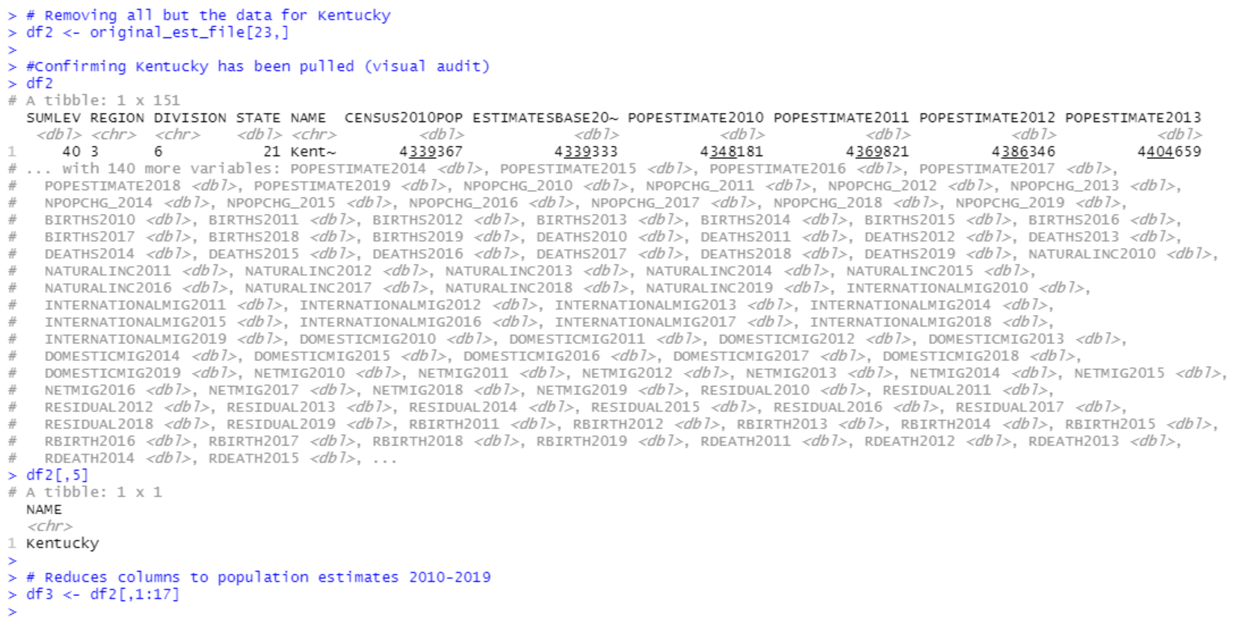
```
>  # Make sure I import all the correct libraries and that they are installed
> library(tinytex)
> library(dplyr)
> library(tidyverse)
> library(ggplot2)
> library(ggthemes)
> library(rmarkdown)
> library(XLConnect)
> library(readxl)
> library(devtools)
>
>
> # Read in excel data source using readxl library
> original_est_file <- read_excel("C:\\Users\\andre\\Desktop\\Extras\\WGU\\Masters\\R for Data Analysts\\nst-est2019-alldata.xlsx")
```

I then proceeded to wrangle the data by first removing all data but KY and then getting down to only the necessary columns. Please see below

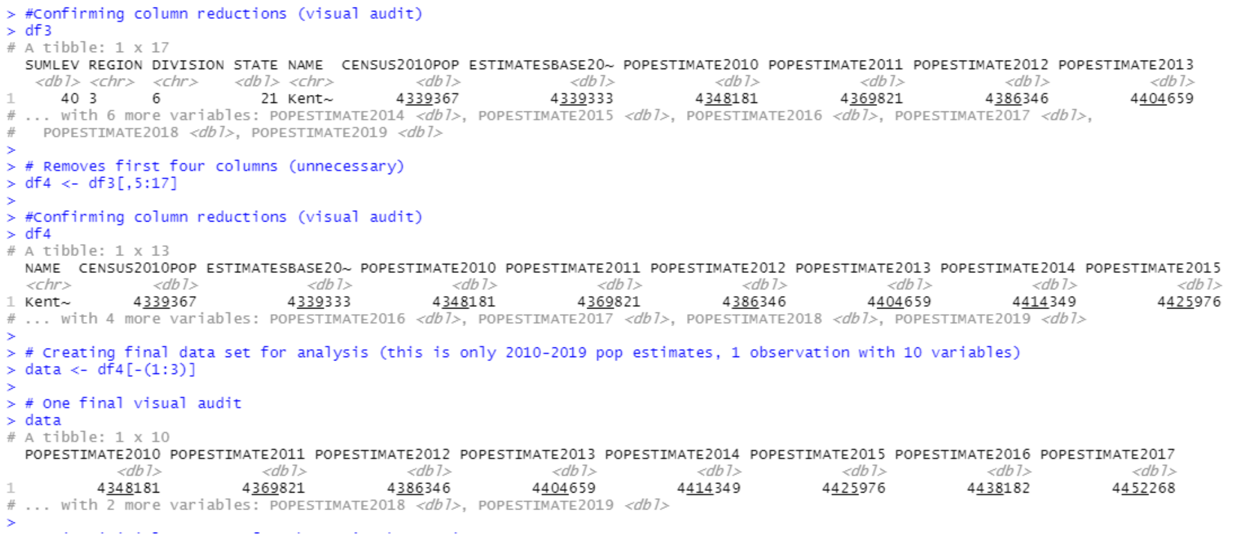


```
# Removing all but the data for Kentucky
df2 <- original_est_file[23,]

#Confirming Kentucky has been pulled (visual audit)
df2
df2[,5]

# Reduces columns to population estimates 2010-2019
df3 <- df2[,1:17]

#Confirming column reductions (visual audit)
df3

# Removes first four columns (unnecessary)
df4 <- df3[,5:17]

#Confirming column reductions (visual audit)
df4

# Creating final data set for analysis (this is only 2010-2019 pop estimates, 1 observation with 10 variables)
data <- df4[-(1:3)]

# One final visual audit
data

# Doing initial summary of each year's observation
summary(data)
```

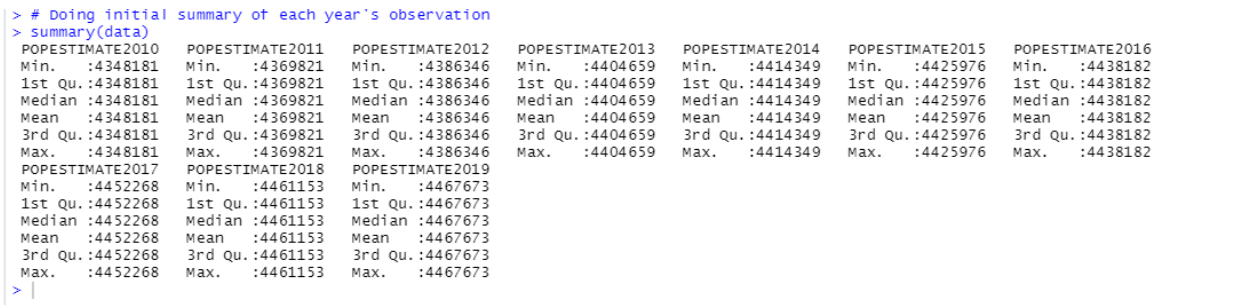
## Console Results



```
> # Removing all but the data for Kentucky
> df2 <- original_est_file[23,]
>
> #Confirming Kentucky has been pulled (visual audit)
> df2
# A tibble: 1 x 151
  SUMLEV REGION DIVISION STATE NAME  CENSUS2010POP ESTIMATESBASE20~ POPESTIMATE2010 POPESTIMATE2011 POPESTIMATE2012 POPESTIMATE2013
   <dbl> <chr>  <chr>    <dbl> <chr>         <dbl>            <dbl>           <dbl>           <dbl>           <dbl>           <dbl>
1     40 3      6           21 Kent~       4339367          4339333         4348181         4369821         4386346         4404659
# ... with 140 more variables: POPESTIMATE2014 <dbl>, POPESTIMATE2015 <dbl>, POPESTIMATE2016 <dbl>, POPESTIMATE2017 <dbl>,
#   POPESTIMATE2018 <dbl>, POPESTIMATE2019 <dbl>, NPOPCHG_2010 <dbl>, NPOPCHG_2011 <dbl>, NPOPCHG_2012 <dbl>, NPOPCHG_2013 <dbl>,
#   NPOPCHG_2014 <dbl>, NPOPCHG_2015 <dbl>, NPOPCHG_2016 <dbl>, NPOPCHG_2017 <dbl>, NPOPCHG_2018 <dbl>, NPOPCHG_2019 <dbl>,
#   BIRTHS2010 <dbl>, BIRTHS2011 <dbl>, BIRTHS2012 <dbl>, BIRTHS2013 <dbl>, BIRTHS2014 <dbl>, BIRTHS2015 <dbl>, BIRTHS2016 <dbl>,
#   BIRTHS2017 <dbl>, BIRTHS2018 <dbl>, BIRTHS2019 <dbl>, DEATHS2010 <dbl>, DEATHS2011 <dbl>, DEATHS2012 <dbl>, DEATHS2013 <dbl>,
#   DEATHS2014 <dbl>, DEATHS2015 <dbl>, DEATHS2016 <dbl>, DEATHS2017 <dbl>, DEATHS2018 <dbl>, DEATHS2019 <dbl>, NATURALINC2010 <dbl>,
#   NATURALINC2011 <dbl>, NATURALINC2012 <dbl>, NATURALINC2013 <dbl>, NATURALINC2014 <dbl>, NATURALINC2015 <dbl>,
#   NATURALINC2016 <dbl>, NATURALINC2017 <dbl>, NATURALINC2018 <dbl>, NATURALINC2019 <dbl>, INTERNATIONALMIG2010 <dbl>,
#   INTERNATIONALMIG2011 <dbl>, INTERNATIONALMIG2012 <dbl>, INTERNATIONALMIG2013 <dbl>, INTERNATIONALMIG2014 <dbl>,
#   INTERNATIONALMIG2015 <dbl>, INTERNATIONALMIG2016 <dbl>, INTERNATIONALMIG2017 <dbl>, INTERNATIONALMIG2018 <dbl>,
#   INTERNATIONALMIG2019 <dbl>, DOMESTICMIG2010 <dbl>, DOMESTICMIG2011 <dbl>, DOMESTICMIG2012 <dbl>, DOMESTICMIG2013 <dbl>,
#   DOMESTICMIG2014 <dbl>, DOMESTICMIG2015 <dbl>, DOMESTICMIG2016 <dbl>, DOMESTICMIG2017 <dbl>, DOMESTICMIG2018 <dbl>,
#   DOMESTICMIG2019 <dbl>, NETMIG2010 <dbl>, NETMIG2011 <dbl>, NETMIG2012 <dbl>, NETMIG2013 <dbl>, NETMIG2014 <dbl>, NETMIG2015 <dbl>,
#   NETMIG2016 <dbl>, NETMIG2017 <dbl>, NETMIG2018 <dbl>, NETMIG2019 <dbl>, RESIDUAL2010 <dbl>, RESIDUAL2011 <dbl>,
#   RESIDUAL2012 <dbl>, RESIDUAL2013 <dbl>, RESIDUAL2014 <dbl>, RESIDUAL2015 <dbl>, RESIDUAL2016 <dbl>, RESIDUAL2017 <dbl>,
#   RESIDUAL2018 <dbl>, RESIDUAL2019 <dbl>, RBIRTH2011 <dbl>, RBIRTH2012 <dbl>, RBIRTH2013 <dbl>, RBIRTH2014 <dbl>, RBIRTH2015 <dbl>,
#   RBIRTH2016 <dbl>, RBIRTH2017 <dbl>, RBIRTH2018 <dbl>, RBIRTH2019 <dbl>, RDEATH2011 <dbl>, RDEATH2012 <dbl>, RDEATH2013 <dbl>,
#   RDEATH2014 <dbl>, RDEATH2015 <dbl>, ...
> df2[,5]
# A tibble: 1 x 1
  NAME
  <chr>
1 Kentucky
>
> # Reduces columns to population estimates 2010-2019
> df3 <- df2[,1:17]
>
```



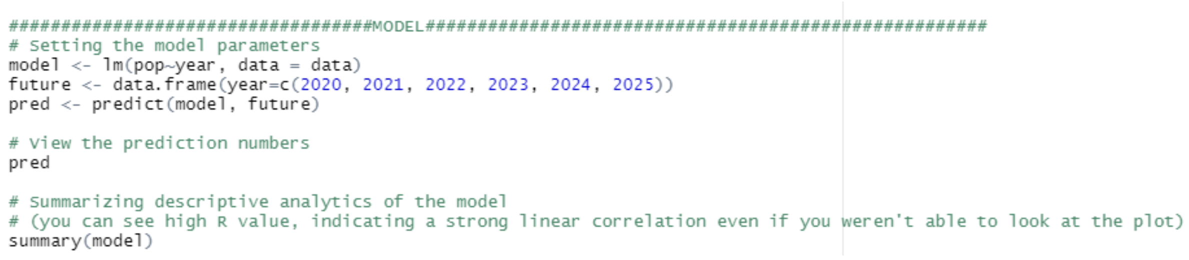
```
> #Confirming column reductions (visual audit)
> df3
# A tibble: 1 x 17
  SUMLEV REGION DIVISION STATE NAME  CENSUS2010POP ESTIMATESBASE20~ POPESTIMATE2010 POPESTIMATE2011 POPESTIMATE2012 POPESTIMATE2013
   <dbl> <chr>  <chr>    <dbl> <chr>         <dbl>            <dbl>           <dbl>           <dbl>           <dbl>           <dbl>
1     40 3      6           21 Kent~       4339367          4339333         4348181         4369821         4386346         4404659
# ... with 6 more variables: POPESTIMATE2014 <dbl>, POPESTIMATE2015 <dbl>, POPESTIMATE2016 <dbl>, POPESTIMATE2017 <dbl>,
#   POPESTIMATE2018 <dbl>, POPESTIMATE2019 <dbl>
>
> # Removes first four columns (unnecessary)
> df4 <- df3[,5:17]
>
> #Confirming column reductions (visual audit)
> df4
# A tibble: 1 x 13
  NAME  CENSUS2010POP ESTIMATESBASE20~ POPESTIMATE2010 POPESTIMATE2011 POPESTIMATE2012 POPESTIMATE2013 POPESTIMATE2014 POPESTIMATE2015
  <chr>         <dbl>            <dbl>           <dbl>           <dbl>           <dbl>           <dbl>           <dbl>           <dbl>
1 Kent~       4339367          4339333         4348181         4369821         4386346         4404659         4414349         4425976
# ... with 4 more variables: POPESTIMATE2016 <dbl>, POPESTIMATE2017 <dbl>, POPESTIMATE2018 <dbl>, POPESTIMATE2019 <dbl>
>
> # Creating final data set for analysis (this is only 2010-2019 pop estimates, 1 observation with 10 variables)
> data <- df4[-(1:3)]
>
> # One final visual audit
> data
# A tibble: 1 x 10
  POPESTIMATE2010 POPESTIMATE2011 POPESTIMATE2012 POPESTIMATE2013 POPESTIMATE2014 POPESTIMATE2015 POPESTIMATE2016 POPESTIMATE2017
            <dbl>           <dbl>           <dbl>           <dbl>           <dbl>           <dbl>           <dbl>           <dbl>
1         4348181         4369821         4386346         4404659         4414349         4425976         4438182         4452268
# ... with 2 more variables: POPESTIMATE2018 <dbl>, POPESTIMATE2019 <dbl>
>
```



```
> # Doing initial summary of each year's observation
> summary(data)
 POPESTIMATE2010   POPESTIMATE2011   POPESTIMATE2012   POPESTIMATE2013   POPESTIMATE2014   POPESTIMATE2015   POPESTIMATE2016
 Min.   :4348181   Min.   :4369821   Min.   :4386346   Min.   :4404659   Min.   :4414349   Min.   :4425976   Min.   :4438182
 1st Qu.:4348181   1st Qu.:4369821   1st Qu.:4386346   1st Qu.:4404659   1st Qu.:4414349   1st Qu.:4425976   1st Qu.:4438182
 Median :4348181   Median :4369821   Median :4386346   Median :4404659   Median :4414349   Median :4425976   Median :4438182
 Mean   :4348181   Mean   :4369821   Mean   :4386346   Mean   :4404659   Mean   :4414349   Mean   :4425976   Mean   :4438182
 3rd Qu.:4348181   3rd Qu.:4369821   3rd Qu.:4386346   3rd Qu.:4404659   3rd Qu.:4414349   3rd Qu.:4425976   3rd Qu.:4438182
 Max.   :4348181   Max.   :4369821   Max.   :4386346   Max.   :4404659   Max.   :4414349   Max.   :4425976   Max.   :4438182
 POPESTIMATE2017   POPESTIMATE2018   POPESTIMATE2019
 Min.   :4452268   Min.   :4461153   Min.   :4467673
 1st Qu.:4452268   1st Qu.:4461153   1st Qu.:4467673
 Median :4452268   Median :4461153   Median :4467673
 Mean   :4452268   Mean   :4461153   Mean   :4467673
 3rd Qu.:4452268   3rd Qu.:4461153   3rd Qu.:4467673
 Max.   :4452268   Max.   :4461153   Max.   :4467673
> |
```

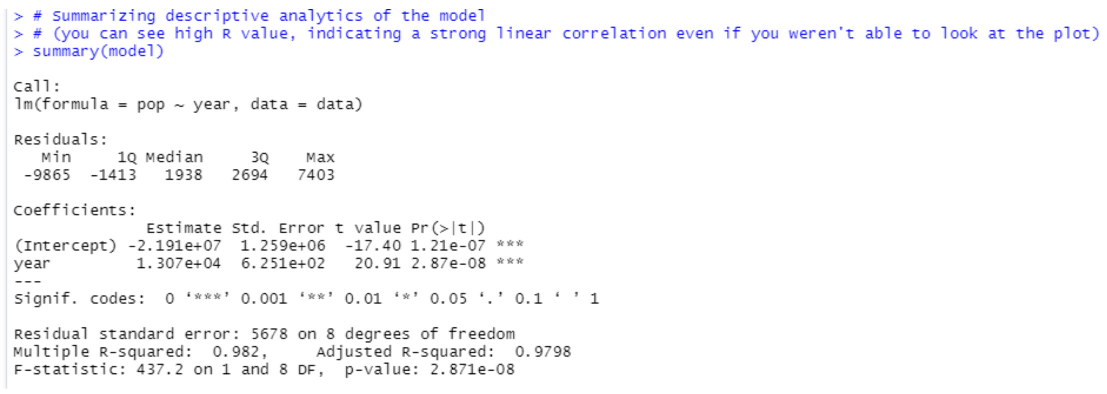
**C. Create an R script that will tabulate a statistical description of the model using R’s summary() function and provide a screenshot of your results.**

I ran an initial summary and you can see those statistics on the original data set above. However, I also ran a summary on the model itself, please see below.



```
##################################MODEL####################################################
# Setting the model parameters
model <- lm(pop~year, data = data)
future <- data.frame(year=c(2020, 2021, 2022, 2023, 2024, 2025))
pred <- predict(model, future)

# View the prediction numbers
pred

# Summarizing descriptive analytics of the model
# (you can see high R value, indicating a strong linear correlation even if you weren't able to look at the plot)
summary(model)
```

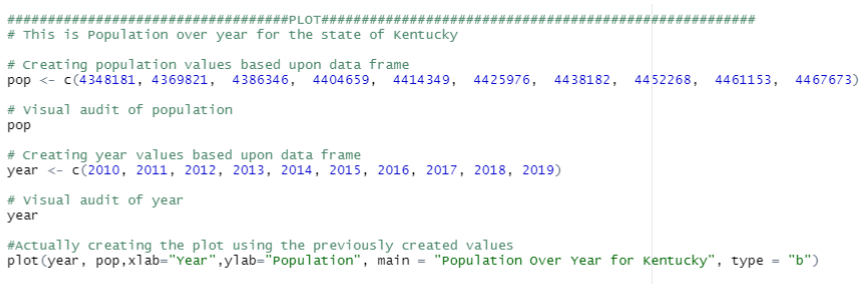
**Console Results**



```
> # Summarizing descriptive analytics of the model
> # (you can see high R value, indicating a strong linear correlation even if you weren't able to look at the plot)
> summary(model)

Call:
lm(formula = pop ~ year, data = data)

Residuals:
   Min     1Q Median     3Q    Max 
 -9865  -1413   1938   2694   7403 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2.191e+07  1.259e+06  -17.40 1.21e-07 ***
year         1.307e+04  6.251e+02   20.91 2.87e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5678 on 8 degrees of freedom
Multiple R-squared:  0.982,	Adjusted R-squared:  0.9798 
F-statistic: 437.2 on 1 and 8 DF,  p-value: 2.871e-08
```

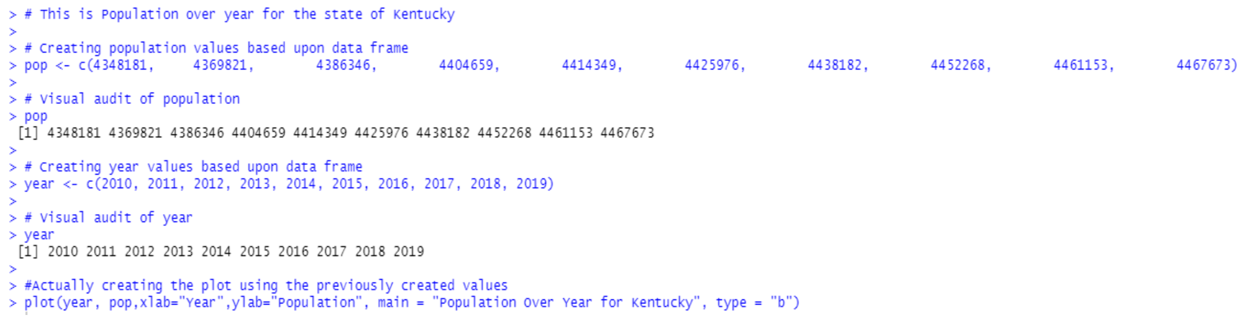
**D. Predict the population size of your state in five years using a linear regression from part A and provide a screenshot of your results.**

This follow is the original plotting of the data all the way through the modeling and plotting of the regression prediction.

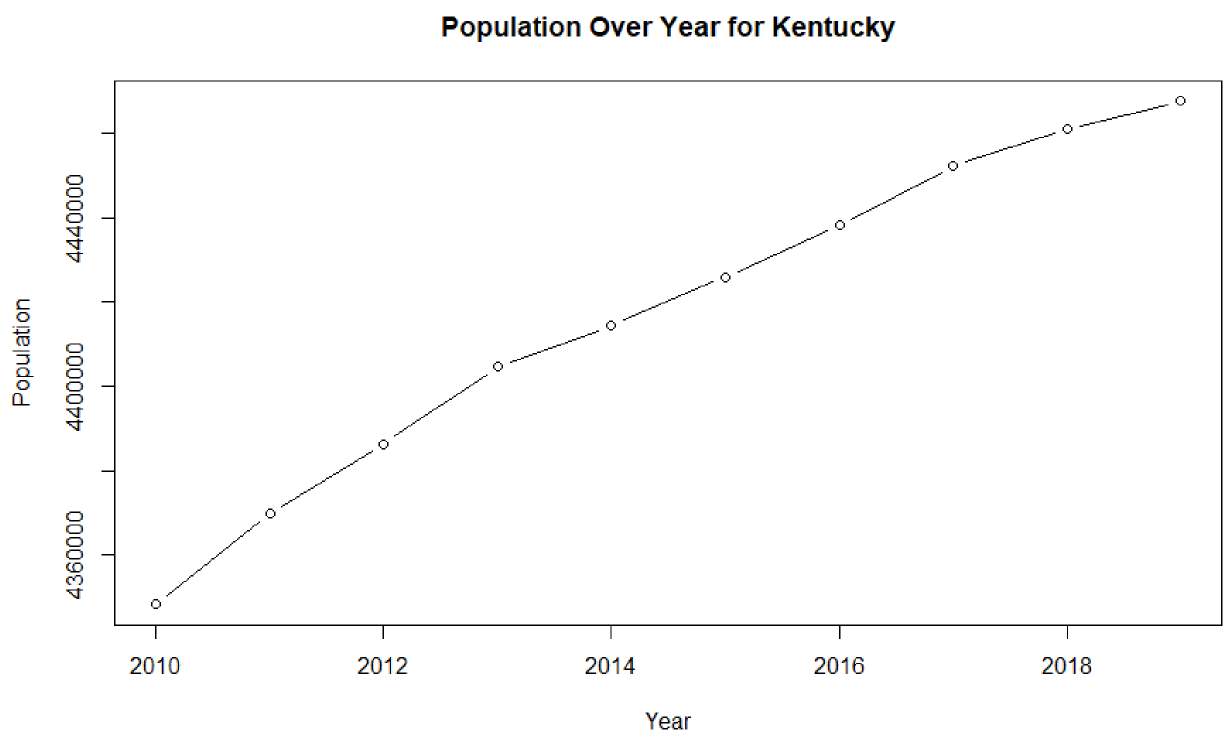


```
##################################PLOT#####################################################
# This is Population over year for the state of Kentucky

# Creating population values based upon data frame
pop <- c(4348181, 4369821,  4386346,  4404659,  4414349,  4425976,  4438182,  4452268,  4461153,  4467673)

# Visual audit of population
pop

# Creating year values based upon data frame
year <- c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019)

# Visual audit of year
year

#Actually creating the plot using the previously created values
plot(year, pop,xlab="Year",ylab="Population", main = "Population Over Year for Kentucky", type = "b")
```

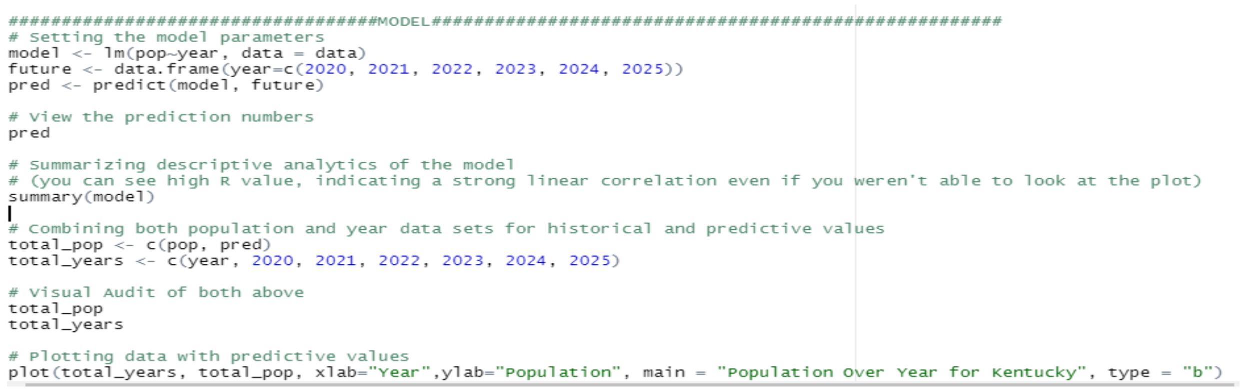
## Console and Plot Results for Historical



```
> # This is Population over year for the state of Kentucky
>
> # Creating population values based upon data frame
> pop <- c(4348181,     4369821,       4386346,       4404659,       4414349,       4425976,       4438182,       4452268,       4461153,       4467673)
>
> # Visual audit of population
> pop
 [1] 4348181 4369821 4386346 4404659 4414349 4425976 4438182 4452268 4461153 4467673
>
> # Creating year values based upon data frame
> year <- c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019)
>
> # Visual audit of year
> year
 [1] 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
>
> #Actually creating the plot using the previously created values
> plot(year, pop,xlab="Year",ylab="Population", main = "Population Over Year for Kentucky", type = "b")
```

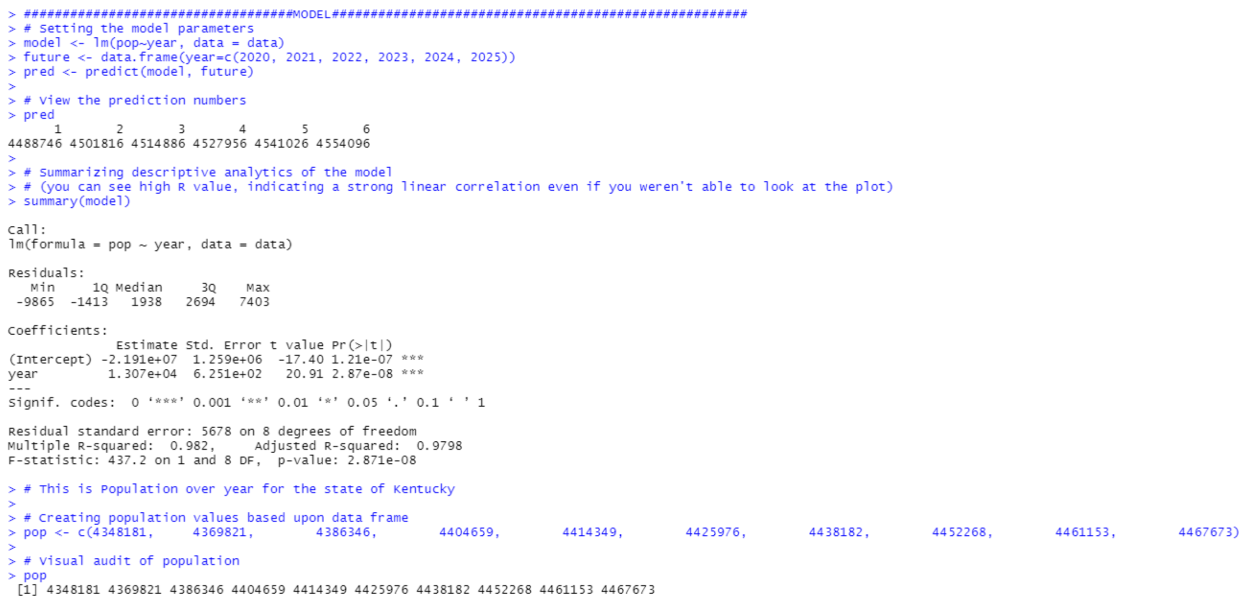


## R code for predictive model

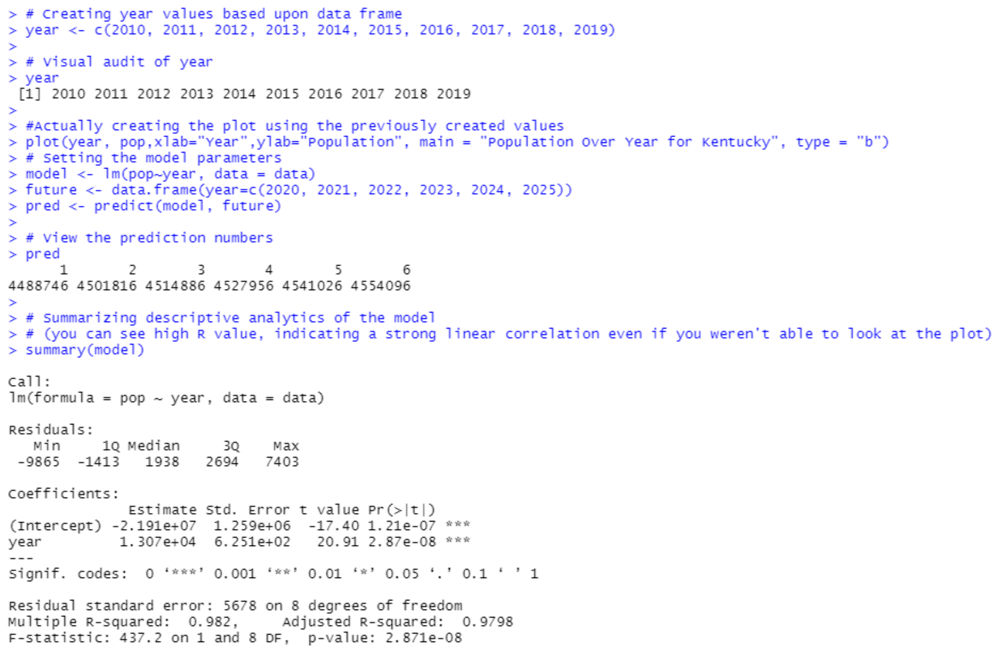


```
####################################MODEL########################################################
# Setting the model parameters
model <- lm(pop~year, data = data)
future <- data.frame(year=c(2020, 2021, 2022, 2023, 2024, 2025))
pred <- predict(model, future)

# View the prediction numbers
pred

# Summarizing descriptive analytics of the model
# (you can see high R value, indicating a strong linear correlation even if you weren't able to look at the plot)
summary(model)

# Combining both population and year data sets for historical and predictive values
total_pop <- c(pop, pred)
total_years <- c(year, 2020, 2021, 2022, 2023, 2024, 2025)

# Visual Audit of both above
total_pop
total_years

# Plotting data with predictive values
plot(total_years, total_pop, xlab="Year",ylab="Population", main = "Population Over Year for Kentucky", type = "b")
```

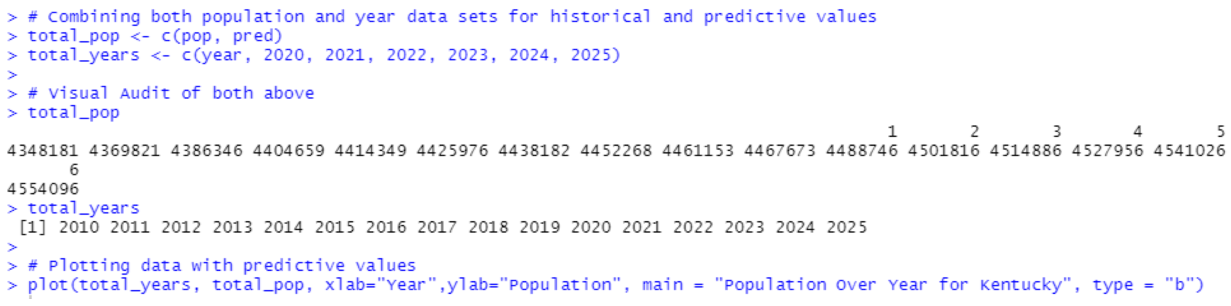
## Console and Plot Results in conjunction with predictive model



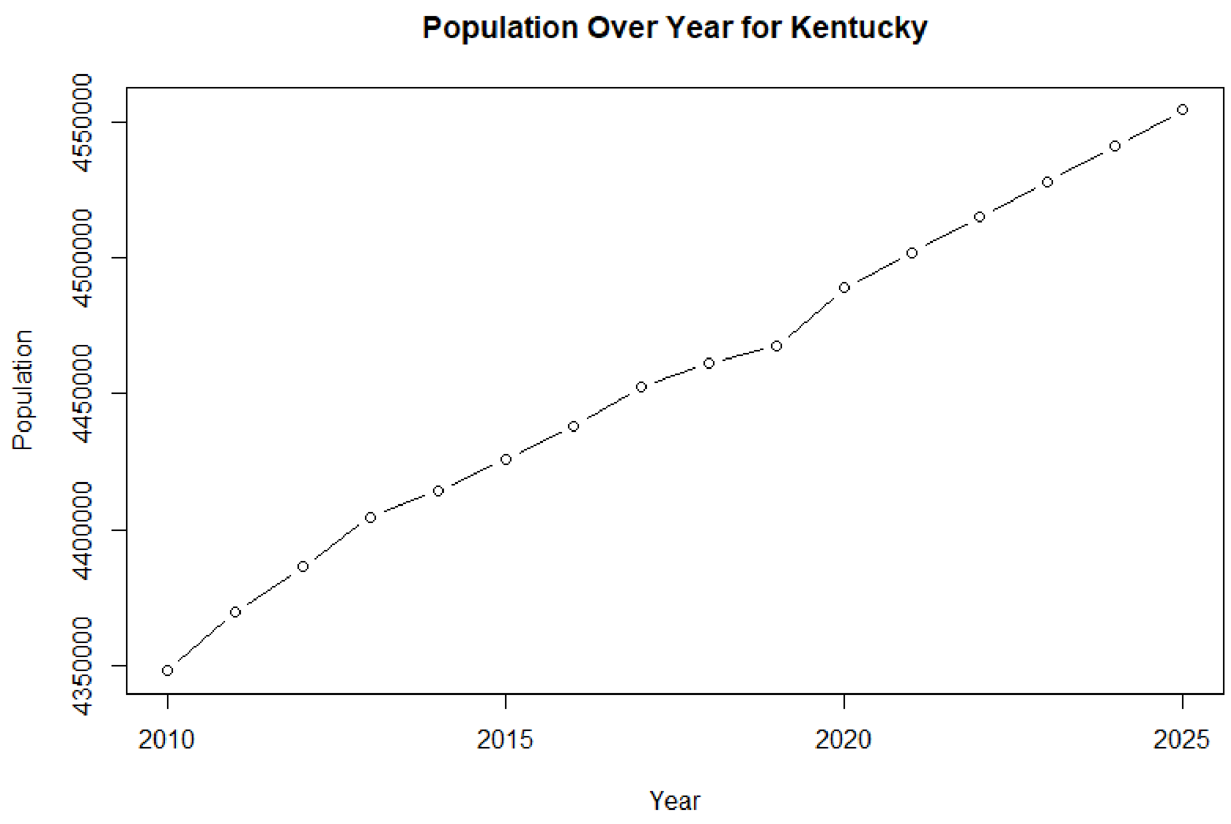
```
> ##################################MODEL###################################################
> # Setting the model parameters
> model <- lm(pop~year, data = data)
> future <- data.frame(year=c(2020, 2021, 2022, 2023, 2024, 2025))
> pred <- predict(model, future)
>
> # View the prediction numbers
> pred
      1       2       3       4       5       6
4488746 4501816 4514886 4527956 4541026 4554096
>
> # Summarizing descriptive analytics of the model
> # (you can see high R value, indicating a strong linear correlation even if you weren't able to look at the plot)
> summary(model)

Call:
lm(formula = pop ~ year, data = data)

Residuals:
   Min     1Q Median     3Q    Max
 -9865  -1413   1938   2694   7403

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.191e+07  1.259e+06  -17.40 1.21e-07 ***
year         1.307e+04  6.251e+02   20.91 2.87e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5678 on 8 degrees of freedom
Multiple R-squared:  0.982,	Adjusted R-squared:  0.9798
F-statistic: 437.2 on 1 and 8 DF,  p-value: 2.871e-08

> # This is Population over year for the state of Kentucky
>
> # Creating population values based upon data frame
> pop <- c(4348181,     4369821,       4386346,       4404659,       4414349,       4425976,       4438182,       4452268,       4461153,       4467673)
>
> # Visual audit of population
> pop
 [1] 4348181 4369821 4386346 4404659 4414349 4425976 4438182 4452268 4461153 4467673
```



```
> # Creating year values based upon data frame
> year <- c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019)
>
> # Visual audit of year
> year
 [1] 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
>
> #Actually creating the plot using the previously created values
> plot(year, pop,xlab="Year",ylab="Population", main = "Population Over Year for Kentucky", type = "b")
> # Setting the model parameters
> model <- lm(pop~year, data = data)
> future <- data.frame(year=c(2020, 2021, 2022, 2023, 2024, 2025))
> pred <- predict(model, future)
>
> # View the prediction numbers
> pred
      1       2       3       4       5       6
4488746 4501816 4514886 4527956 4541026 4554096
>
> # Summarizing descriptive analytics of the model
> # (you can see high R value, indicating a strong linear correlation even if you weren't able to look at the plot)
> summary(model)

Call:
lm(formula = pop ~ year, data = data)

Residuals:
   Min     1Q Median     3Q    Max
 -9865  -1413   1938   2694   7403

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.191e+07  1.259e+06  -17.40 1.21e-07 ***
year         1.307e+04  6.251e+02   20.91 2.87e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5678 on 8 degrees of freedom
Multiple R-squared:  0.982,	Adjusted R-squared:  0.9798
F-statistic: 437.2 on 1 and 8 DF,  p-value: 2.871e-08
```



```
> # Combining both population and year data sets for historical and predictive values
> total_pop <- c(pop, pred)
> total_years <- c(year, 2020, 2021, 2022, 2023, 2024, 2025)
>
> # Visual Audit of both above
> total_pop
                                                                                      1       2       3       4       5
4348181 4369821 4386346 4404659 4414349 4425976 4438182 4452268 4461153 4467673 4488746 4501816 4514886 4527956 4541026
      6
4554096
> total_years
 [1] 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025
>
> # Plotting data with predictive values
> plot(total_years, total_pop, xlab="Year",ylab="Population", main = "Population Over Year for Kentucky", type = "b")
```


Population Over Year for Kentucky
Population
4350000
4400000
4450000
4500000
4550000
2010
2015
2020
2025
Year

## References

How to Create a Data Frame from Scratch in R. (2019, November 21). Dummies. Retrieved from: https://www.dummies.com/programming/r/how-to-create-a-data-frame-from-scratch-in-r/