BOM1 TASK 1: ESTIMATING POPULATION SIZE

Amy Nguyen

#Estimating Population Size ##Amy Nguyen ##C997 - R for Data Analysts ##November 13th, 2020

###Introduciton The United States collects and analyzes demographic data from the U.S. population. The U.S. Census Bureau provides annual estimates of the population size of each U.S. state and region. Many important decisions are made using the estimated population dynamics, including the investments in new infrastructure, such as schools and hospitals; establishing new job training centers; opening or closing schools and senior centers; and adjusting the emergency services to the size and characteristics of the demographics of metropolitan and other areas, states, or the country as a whole. The census data and estimates are publicly available on the U.S. census website. Data analysts use a variety of tools to create models for predictions, including models of population dynamics of a state of a region. For this project, you will use R to create a linear regression model of the population dynamics of your state and predict the size of its population.

###Importing the Data For this project the data can be found from the U.S. Census Bureau at WWW.census.org. For our purposes I’ve decided to use the “City and Town Population Totals: 2010-2019” for Texas. This data can be found at <https://www.census.gov/data/datasets/time-series/demo/popest/2010s-total-cities-and-towns.html#ds>. My first step of this project was to import the data into a dataframe in R.

population <- read.csv("https://www2.census.gov/programs-surveys/popest/datasets/2010-2019/cities/totals/sub-est2019\_48.csv", stringsAsFactors = FALSE)  
head(population)

## SUMLEV STATE COUNTY PLACE COUSUB CONCIT PRIMGEO\_FLAG FUNCSTAT NAME  
## 1 40 48 0 0 0 0 0 A Texas  
## 2 162 48 0 100 0 0 0 A Abbott city  
## 3 162 48 0 160 0 0 0 A Abernathy city  
## 4 162 48 0 1000 0 0 0 A Abilene city  
## 5 162 48 0 1108 0 0 0 A Ackerly city  
## 6 162 48 0 1240 0 0 0 A Addison town  
## STNAME CENSUS2010POP ESTIMATESBASE2010 POPESTIMATE2010 POPESTIMATE2011  
## 1 Texas 25145561 25146091 25241971 25645629  
## 2 Texas 356 361 362 362  
## 3 Texas 2805 2812 2818 2833  
## 4 Texas 117063 117514 117805 118762  
## 5 Texas 220 220 220 219  
## 6 Texas 13056 12537 12565 13259  
## POPESTIMATE2012 POPESTIMATE2013 POPESTIMATE2014 POPESTIMATE2015  
## 1 26084481 26480266 26964333 27470056  
## 2 361 355 354 354  
## 3 2822 2798 2738 2724  
## 4 119853 119797 120648 121707  
## 5 219 225 228 230  
## 6 14640 14871 14931 15013  
## POPESTIMATE2016 POPESTIMATE2017 POPESTIMATE2018 POPESTIMATE2019  
## 1 27914410 28295273 28628666 28995881  
## 2 356 361 365 369  
## 3 2741 2735 2711 2706  
## 4 121883 122186 122775 123420  
## 5 229 228 229 232  
## 6 14945 14924 15338 16263

### Cleaning and Prepping The Data

For this project a linear regression will be created using the total population estimate for each year from 2010 to 2019. Therefore, in order to simplify and improve the readability of the data we will be removing any unnecessary columns and/or rows.

df <- subset(population, select = -c(SUMLEV, STATE, COUNTY, PLACE, COUSUB, CONCIT, PRIMGEO\_FLAG, FUNCSTAT))  
head(df)

## NAME STNAME CENSUS2010POP ESTIMATESBASE2010 POPESTIMATE2010  
## 1 Texas Texas 25145561 25146091 25241971  
## 2 Abbott city Texas 356 361 362  
## 3 Abernathy city Texas 2805 2812 2818  
## 4 Abilene city Texas 117063 117514 117805  
## 5 Ackerly city Texas 220 220 220  
## 6 Addison town Texas 13056 12537 12565  
## POPESTIMATE2011 POPESTIMATE2012 POPESTIMATE2013 POPESTIMATE2014  
## 1 25645629 26084481 26480266 26964333  
## 2 362 361 355 354  
## 3 2833 2822 2798 2738  
## 4 118762 119853 119797 120648  
## 5 219 219 225 228  
## 6 13259 14640 14871 14931  
## POPESTIMATE2015 POPESTIMATE2016 POPESTIMATE2017 POPESTIMATE2018  
## 1 27470056 27914410 28295273 28628666  
## 2 354 356 361 365  
## 3 2724 2741 2735 2711  
## 4 121707 121883 122186 122775  
## 5 230 229 228 229  
## 6 15013 14945 14924 15338  
## POPESTIMATE2019  
## 1 28995881  
## 2 369  
## 3 2706  
## 4 123420  
## 5 232  
## 6 16263

* The cleaned data has been put into a new data frame named “df”. Specifically, a subset of the original data without the columns SUMLEV, STATE, COUNTY, PLACE, COUSUB, CONCIT, PRIMGEO\_FLAG, and FUNCSTAT has been put into a dataframe named “df”

Now that the data has each unnecessary column removed, it can be further manipulated. For the purpose of estimating the total population for the state of Texas we only need the information in the first row or the “State Total”. Thus, we can pull that data out using the head() function and put it into a new dataframe.

The issue now is that the X values and Y value are in the wrong spot. In order to build a linear regression model where the target is population and the predictor is time the rows and columns need to be reversed. Therefore, the next step is to transpose “total” and correct any other issues such as column names.

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

total <- head(df, n=1)  
total <- as.data.frame(t(total))  
total <- tibble::rownames\_to\_column(total, "Year")  
colnames(total)[2] <- "Population\_Estimates"  
head(total)

## Year Population\_Estimates  
## 1 NAME Texas  
## 2 STNAME Texas  
## 3 CENSUS2010POP 25145561  
## 4 ESTIMATESBASE2010 25146091  
## 5 POPESTIMATE2010 25241971  
## 6 POPESTIMATE2011 25645629

The next step to prepping the data for the linear regression is removing the first 3 rows, “NAME”, “STNAME”, “CENSUS2010POP” and “ESTIMATESBASE2010”. In addition I’ll be renaming the rows for column “Year”.

total <- total[-c(1,2,3,4),]  
total$Year <-c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019)  
  
head(total)

## Year Population\_Estimates  
## 5 2010 25241971  
## 6 2011 25645629  
## 7 2012 26084481  
## 8 2013 26480266  
## 9 2014 26964333  
## 10 2015 27470056

The last thing to do is change the data types for the column “Population\_Estimates”. Using the str() function it can be seen that "Population\_Estimates is a character data type.

str(total)

## 'data.frame': 10 obs. of 2 variables:  
## $ Year : num 2010 2011 2012 2013 2014 ...  
## $ Population\_Estimates: chr "25241971" "25645629" "26084481" "26480266" ...

For the current scenario it is necessary to convert “Population\_Estimates” to an integer data type.

total$Population\_Estimates <-as.integer(as.character(total$Population\_Estimates))  
str(total)

## 'data.frame': 10 obs. of 2 variables:  
## $ Year : num 2010 2011 2012 2013 2014 ...  
## $ Population\_Estimates: int 25241971 25645629 26084481 26480266 26964333 27470056 27914410 28295273 28628666 28995881

The data has now been cleaned and manipulated in order to easily create a linear regression with “Year” as the predictor or independent variable and “Population\_Estimates” as the dependent variable.

### Creating a Linear Regression

In order to predict the future population size for the state of Texas a linear regression model named “lmPop” will be created using the lm() function.

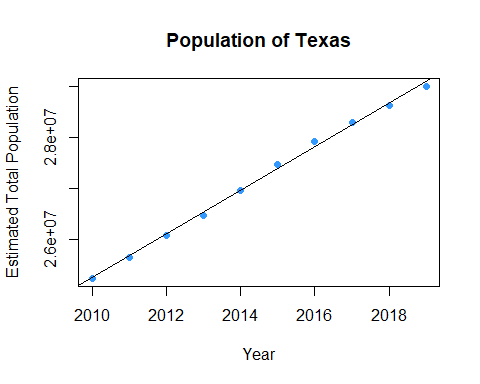
lmPop <-lm(Population\_Estimates~Year, data = total)  
lmPop

##   
## Call:  
## lm(formula = Population\_Estimates ~ Year, data = total)  
##   
## Coefficients:  
## (Intercept) Year   
## -833917553 427446

The linear regression equation for the data is Y = Year(X) + Intercept or Y = 427,446(X) + -833,917,553.

A graph of the data points and the linear regression line can be seen below:

plot(total$Year, total$Population\_Estimates,  
 main = "Population of Texas",  
 xlab = "Year",  
 ylab = "Estimated Total Population",  
 pch = 19,  
 col = "#3399FF")  
  
abline(-833917553, 427446)



### Statisitcal Description of The Model Using summary()

### Predicting Future Population

Using the linear regression function Y = 427,446(X) + -833,917,553 it can be estimated that the population for the state of Texas in 2025 will be 31,701,097.

NEED TO USE PREDICT FUNCTION(?)

(427466\*2025) + -833917553

## [1] 31701097