BOM1 TASK 1: ESTIMATING POPULATION SIZE

Amy Nguyen

###Packages Directly copy and paste the following to ensure that the necessary packages are present to load the code:

install.packages(“dplyr”) install.packages(“tibble”) install.packages(“tidyverse”) install.packages(“ggpubr”)

###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 I 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 of Texas only the first row or the “State Total” is need. Thus, we can pull that data out using the head() function and put it into a new dataframe.From there the data needs to be transposed to invert the X and Y columns so that a column for all of the years and a separate column for the population estimates can be created.

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 4 rows, “NAME”, “STNAME”, “CENSUS2010POP” and “ESTIMATESBASE2010” as they are unnecessary. In addition each row in the column “Year” will be renamed in order to convert the data type to a numeric type and remove the “POPESTIMATE” text before the 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 type 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.

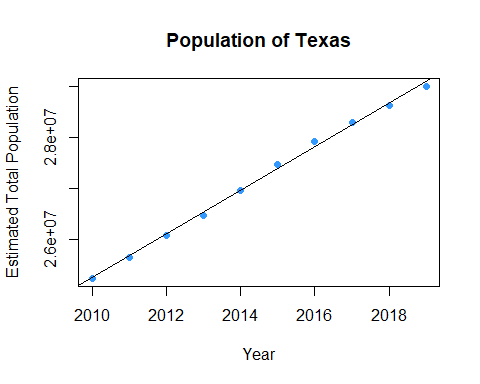
lmPop <-lm(Population\_Estimates~Year, data = total)  
lmPop

##   
## 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(lmPop)



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

The summary() function can be used to provide statistical information about the linear model.

summary(lmPop)

##   
## Call:  
## lm(formula = Population\_Estimates ~ Year, data = total)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -99722 -37220 -12810 42411 101145   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -833917553 14823968 -56.26 1.11e-11 \*\*\*  
## Year 427446 7359 58.09 8.57e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 66840 on 8 degrees of freedom  
## Multiple R-squared: 0.9976, Adjusted R-squared: 0.9973   
## F-statistic: 3374 on 1 and 8 DF, p-value: 8.567e-12

Looking at the summary we can see that the model is a good fit to the data. The multiple R-squared value is very close to 1 and the 3-stars for the variable “Year” indicate a low p-value close to 0 which then indicates a high significance level.

### Predicting Future Population

In order to predict the future population for the state of Texas a data frame needs to be created with one column for the years to be predicted and another column to hold the future predicted values.

future\_df <- data.frame("Year" = 2020:2030, "Population\_Estimates" = 0)  
future\_df

## Year Population\_Estimates  
## 1 2020 0  
## 2 2021 0  
## 3 2022 0  
## 4 2023 0  
## 5 2024 0  
## 6 2025 0  
## 7 2026 0  
## 8 2027 0  
## 9 2028 0  
## 10 2029 0  
## 11 2030 0

Now that the data frame holding future year values has been created it is now possible to use the predict() function to predict the population for future years.

predictions <- predict(lmPop, newdata = future\_df)  
  
future\_df$Population\_Estimates <- predictions  
future\_df

## Year Population\_Estimates  
## 1 2020 29523049  
## 2 2021 29950495  
## 3 2022 30377940  
## 4 2023 30805386  
## 5 2024 31232832  
## 6 2025 31660278  
## 7 2026 32087724  
## 8 2027 32515170  
## 9 2028 32942615  
## 10 2029 33370061  
## 11 2030 33797507

Looking at the table above we can see that the total population of Texas in 2025 is predicted to be 31,660,278.