Analysis of US Population using Data Analytics and Data Science Tools

Sharan Kumar Paratala Rajagopal

Senior Manager, Capgemini America Inc. Dallas, TX, USA

Abstract: ***The research document explains the various Data Science tools and techniques used to analyze US population data in order to make better decisions by using data analysis of the population data for investments in schools, hospitals, establishing new job training 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. There are various benefits by analyzing the data which gives holistic view for government to invest in improvements which can be made in order to serve the people better.***

Keywords: Data Analytics, Data Science tools, Python, SQL

1. Introduction

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. Various decisions will be made from this data by seeing where the best fit is to invest and helps in serving the community better.

There are multiple data science tools and techniques which can be used to clean the data in the required format, analyze it, represent in the required format and visually shown to make decisions. Will introduce to some of them in the next subsequent sections and will extract the raw data from the US census website.

2. Data Science Tools

This will be a multistep process from extraction to final representation of the data. The next following sections gives a detailed description of what tools can be used to perform the specific activities. In this research article we are going to use three main tools which helps in extraction, analyzing and representing the data.

- Python
- R
- SQL

3. Extract data using python

Python is the most powerful tool which can quickly extract data from the website using web scrapping. The advantages of python is due to the inbuilt libraries which can be used to perform the action quickly and effectively.

Let's see how we can extract the web links from the HTML code of the "Current Estimates" web link. "Figure 1" shows the python code which can be used to extract the data from the US government website [1]. "href" provides the details of the link locator to another HTML page.

```
import requests
from bs4 import BeautifulSoup
import urllib.request
import urllib.parse

# website link for the assignment
url = "https://www.census.gov/programs-surveys/popest.html"

r = requests.get(url)

soup = BeautifulSoup(r.content,"html.parser")

# find_all with 'a' parameter helps in locating the links of the html pages
links = soup.find_all("a")

# for all the external web pages from the website
for link in links:
    link.get("href")
```

Figure 1: Webscrapping using Python

"Figure 2" provides details of relative links saved as absolute URL's in the output file.

```
# function to get the unique links and modify the url to absolute path
# and set() function gets all the unique links

def unique_links(tags,url):
    cleaned_links = set()

    for link in links:
        link = link.get("href")

        if link is None:
            continue
        if link.endswith('/') or link.endswith('#'):
            link = link [-1]

        actual_url = urllib.parse.urljoin(url,link)
        cleaned_links.add(actual_url)

    return cleaned_links

cleaned_links = unique_links(links,url)
```

Figure 2: Convert relative to absolute links

"Figure 3" shows that there are only unique links in the output file. And any duplicates will be removed. set() function provides the unique URL from the website.

```
def unique_links(tags,url):
    cleaned_links = set()
```

Figure 3: Unique links extraction

To create the csv output file
filename = "output_unique_weblink.csv"
f = open(filename,"w")
header = "Output_Weblinks\n"

f.write(header)

Write into the csv file all unique links from the website
for link in cleaned_links:
f.write(str(link) + '\n')

4. Linear regression analysis using R

Predicting the size of the population for the state which you currently live in for 2020 based on the current estimates.

The data extracted from the US census [1] website can be cleaned up and imported into the R Integrated development environment and run the commands to interpret the slope and intercept parameters accordingly. “Figure 4” provides the R command to verify the slope and intercept parameters.

```
> lm

Call:
lm(formula = population ~ year, data = input_r)

Coefficients:
(Intercept)         year
 -857358730       439088
```

Figure 4: R command for intercepting parameters

“Figure 5” provides the details of the linear regression model summary.

```
> summary(lm)

Call:
lm(formula = population ~ year, data = input_r)

Residuals:
       1        2        3        4        5        6        7
 36877.9   -130.7 -13952.4 -51170.0 -19031.6  26768.7  20638.1

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -857358730   12819393  -66.88 1.42e-08 ***
year            439088       6368   68.95 1.22e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33700 on 5 degrees of freedom
Multiple R-squared:  0.9989,    Adjusted R-squared:  0.9987
F-statistic:  4754 on 1 and 5 DF,  p-value: 1.215e-08
```

Figure 5: Summary of the linear regression.

“Figure 6” plot for the linear regression.

```
> plot(lm)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

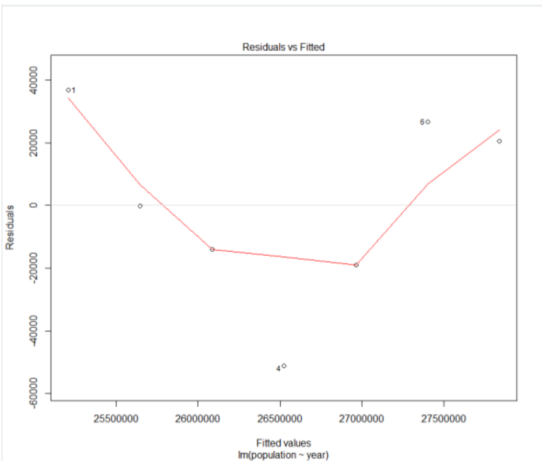

Figure 6: Plot for linear regression.

“Figure 7” provides the details of linear regression for year vs population.

```
> plot(input_r$year,input_r$population)
> abline(lm)
```

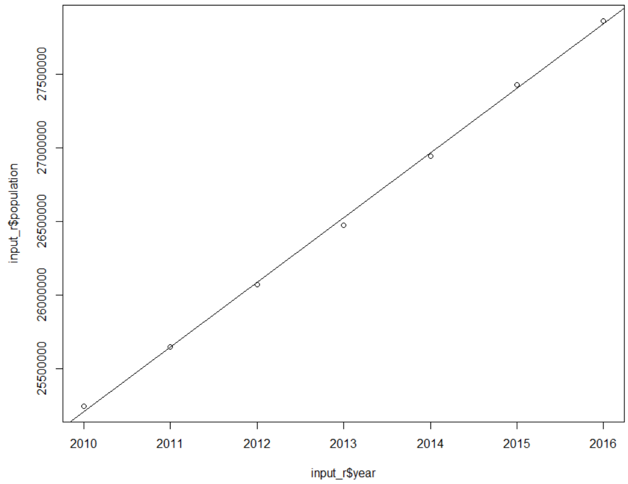

Figure 7: Year vs Population linear line

Use the predict command to predict the population size for the year 2020 for state Texas based on the input value and linear regression model.

Finding the coefficient parameters

```
> coef(lm)
 (Intercept)         year
-857358730.1     439087.6
```

```
> lm$coef[1]
(Intercept)
 -857358730
> lm$coef[2]
    year
439087.6
```

Calculate using the coefficients

```
> coef(lm)
 (Intercept)         year
-857358730.1     439087.6
> lm$coef[1]
(Intercept)
 -857358730
> lm$coef[2]
    year
439087.6
> lm$coef[1] + lm$coef[2] * 2020
(Intercept)
   29598308
```

Using predict command

```
> predict(lm,list(year =2020))
       1
29598308
```

Predict for list of values from 2017 to 2020

```
> predict(lm,data.frame(year=c(2017,2018,2019,2020)))
       1        2        3        4
28281046 28720133 29159221 29598308
```

Plot the linear regression line

```
> plot(predicted_output_r$year,predicted_output_r$population, main ="Predicted values for State Texas 2020", xlab = "Year",y
lab ="Population",ylim=c(25000000,30000000))
> abline(lm)
```

"Figure 8" provides the details of predicted values for state of Texas 2020 and the linear line is plotted against year vs population.

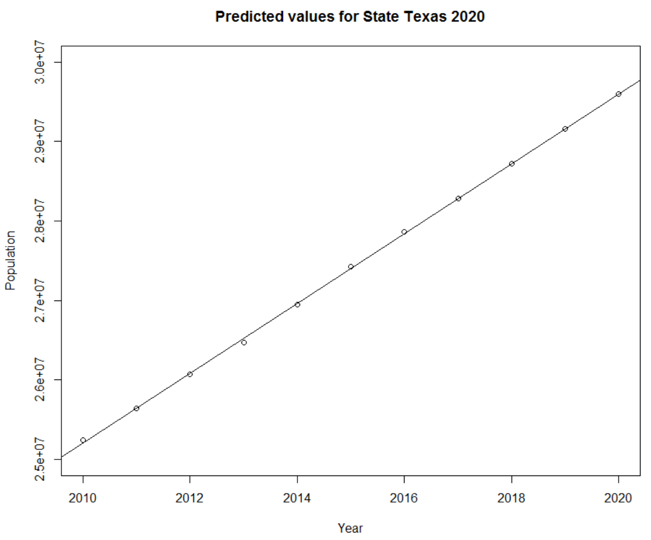

Figure 8: Predicted values for state Texas 2020

"Figure 9" shows the histogram of the predicted population for state of Texas in 2020.

```
> hist(predicted_output_r$year, main = "Histogram for predicted population", xlab = "Year", ylab = "Population", xlim=c(2010
,2020))
```

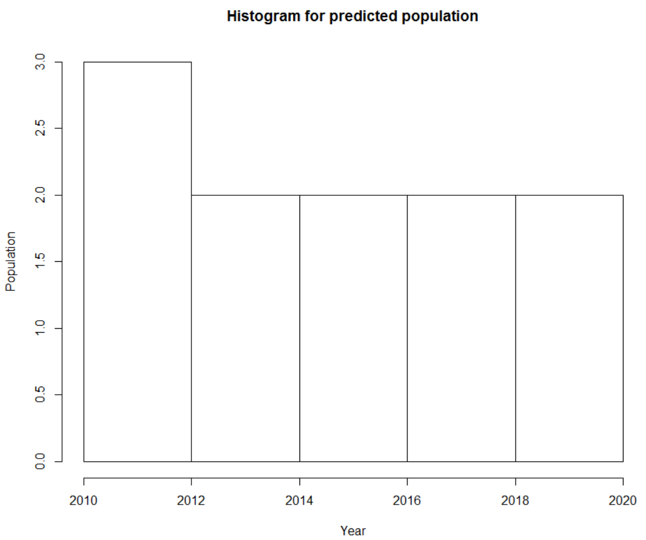

Figure 9: Histogram for predicted population

hist(predicted_output_r$population, breaks=seq(20000000, 30000000, 1000000), main='Histogram of the population estimates for Texas (2016 Dataset)', xlab='Population (Bin Size One Million)')

"Figure 10" shows the histogram for the population estimate for Texas state based on 2016 dataset chosen for the analysis.

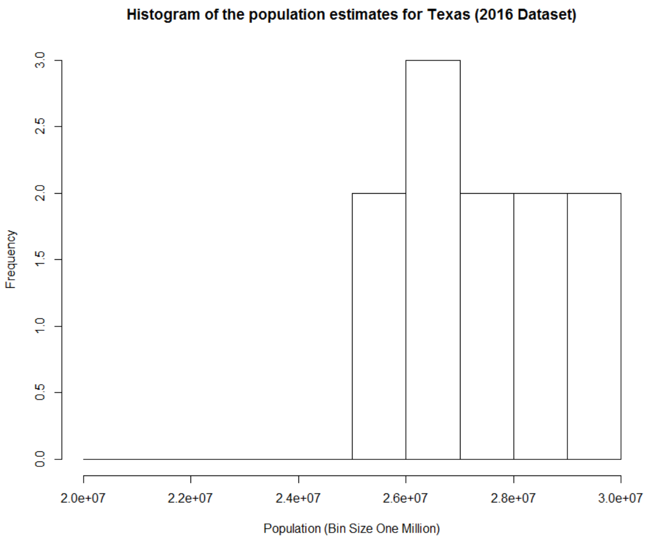

Figure 10: Histogram for predicted population of Texas 2020

Get the data from the current estimates [2] and most recent vintage estimates [3] data. After getting the data in the format load them into the SQL tables for analysis.

```
SELECT nst_est2015.area,
       nst_est2015.year,
       (nst_est2015.population - nst_est2016.population) AS population_diff
  FROM nst_est2015
       INNER JOIN
       nst_est2016
          ON     nst_est2015.area = nst_est2016.area
             AND nst_est2015.year = nst_est2016.year
ORDER BY nst_est2015.year, nst_est2015.area;
```

"Figure 11" provides the detailed output based on the data loaded into the tables.

AREA	YEAR	POPULATION_DIFF
Alabama	2010	-331
Alaska	2010	-10
Arizona	2010	-104
Arkansas	2010	399
California	2010	1394
Colorado	2010	-390
Connecticut	2010	-182
Delaware	2010	-25
District of Columbia	2010	-57
Florida	2010	792
Georgia	2010	-67
Hawaii	2010	35
Idaho	2010	-24

Figure 11: Difference in population

We can create multiple SQL queries to join the data set to get the output in required format. One of the query where we can find useful information is to find the differences between the states that exceed 10,000 individuals.

```
SELECT nst_est2015.area,
       nst_est2015.year,
       ROUND (nst_est2016.population - nst_est2015.population, -2)
  FROM nst_est2015
       INNER JOIN
       nst_est2016
          ON     nst_est2015.area = nst_est2016.area
             AND nst_est2015.year = nst_est2016.year
 WHERE ABS (nst_est2016.population - nst_est2015.population) > 10000;
```

"Figure 12" shows the result of the sql query and shows the states where there is individual > 10000

AREA	YEAR	POPULATION
California	2011	-23200
California	2012	-45000
Texas	2012	-18100
California	2013	-78900
Florida	2013	-12400
Illinois	2013	-10100
New York	2013	-17500
Texas	2013	-27100
California	2014	-111500
Florida	2014	-16800
Illinois	2014	-14600

Figure 12: Difference > 10000 population

Now we can see one more variation in the query to provide the data in a readable data format which helps for analysis.

```
  SELECT x.area,
         MAX (DECODE (x.year, 2010, x.population)) year_2010,
         MAX (DECODE (x.year, 2011, x.population)) year_2011,
         MAX (DECODE (x.year, 2012, x.population)) year_2012,
         MAX (DECODE (x.year, 2013, x.population)) year_2013,
         MAX (DECODE (x.year, 2014, x.population)) year_2014,
         MAX (DECODE (x.year, 2015, x.population)) year_2015
    FROM (SELECT r.area, r.year, r.population
            FROM c742_section_i r) x
GROUP BY x.area
order by x.area
```

"Figure 13" shows tabular format of data representation using SQL query. Year is transposed to column and state as rows which shows the population accordingly.

AREA	YEAR_2010	YEAR_2011	YEAR_2012	YEAR_2013	YEAR_2014	YEAR_2015
Arizona						-10500
California		-23200	-45000	-78900	-111500	-150900
Florida				-12400	-16800	-26400
Georgia						-15500
Illinois				-10100	-14600	-20900
Maryland						-11400
Massachusetts						-10200
New Jersey					-13800	-22600
New York				-17500	-30300	-48600
Pennsylvania						-10600
Texas			-18100	-27100	-34300	-39500
Virginia					-10700	-15400
Washington						-10100

Figure 13: Transpose output of population

5. Result of Analysis

Data representation and data analysis can be performed using various tools and techniques [4]. The research paper shows that the most effective tools to use for data analysis includes and not limited to R, Python and SQL. By using various commands and inbuilt libraries plots can be constructed for easy readability of the huge data set and gives the picture in a faster and better way to identify the most influential factors in a data set.

References

[1] US population data available: https://www.census.gov/programs-surveys/popest.html
[2] Current estimates of the census details obtained from https://www2.census.gov/programs-surveys/popest/tables/2010-2016/state/totals/nst-est2016-01.xlsx
[3] Most recent vintage estimates obtained from https://www2.census.gov/programs-surveys/popest/tables/2010-2015/state/totals/nst-est2015-01.xlsx
[4] Sharan Kumar Paratala Rajagopal, 2020, Data Visualization for Vehicle Selection Process, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 07 (July 2020),

Author Profile

Sharan Kumar Paratala Rajagopal is a Senior Manager with Capgemini America, Inc. having 14+ years of design, development and architecture experience. He is specialized in Java/J2EE, Integration methodologies, Guidewire Product, Data Analytics, AI and Cloud technologies. He has vast domain experience in Public Services, Hospitality and Property & Casualty Insurance. Also contributed multiple technical articles to major Dev communities.