

## Census Analysis

### Part I – Python

A.

The python script utilizes the beautiful soup library to parse the HTML from the link provided. After extracting the HTML from the webpage into memory, the script will use the beautiful soup library to find the 'A' elements that contain 'HREF' links. The list returned is specifically the links from the webpage

B.

To differentiate between internal and external links, the following code segment was implemented:

```
tf=a['href'].startswith['http']
```

This sets the variable tf to a Boolean value based on whether the link provided starts with HTTP or not. If the links starts with http it is a good indicator that it is an external link.

C.

The following logic will manipulate the internal links to be outputted in a working format to the CSV file.

```
If tf == True:
```

```
link=a['href']
```

```
else:
```

```
link=baselink + a['href']
```

If the Boolean logic from 1.B returns True, then the link will export as is, with no changes. If the Boolean logic from 1.B returns False, then the link will be manipulated to add the base URL to the beginning.

D.

The links found are added into a set, linkSet. The set type will automatically check the existing set for duplicates before adding a new record. This is the way the program reports on each link only one time.

E.

See scraper.py

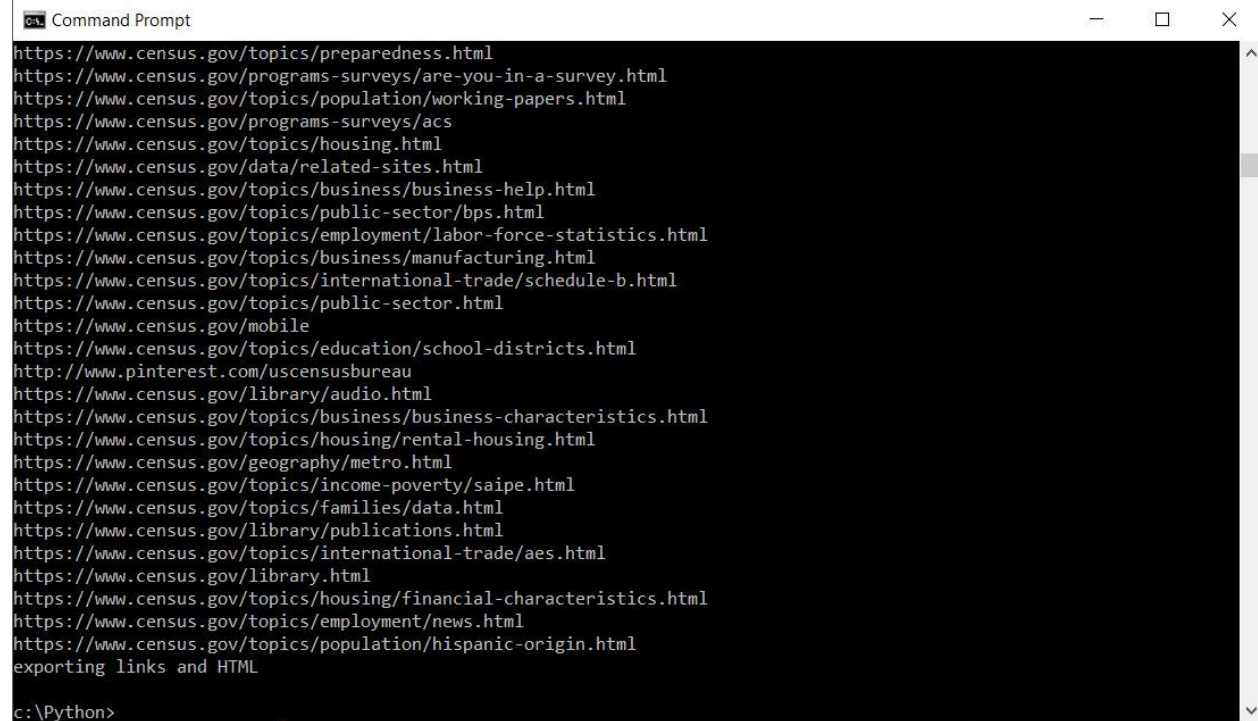
F.

See htmlFile.txt

G.

See links.csv

H.



```
Command Prompt
https://www.census.gov/topics/preparedness.html
https://www.census.gov/programs-surveys/are-you-in-a-survey.html
https://www.census.gov/topics/population/working-papers.html
https://www.census.gov/programs-surveys/acs
https://www.census.gov/topics/housing.html
https://www.census.gov/data/related-sites.html
https://www.census.gov/topics/business/business-help.html
https://www.census.gov/topics/public-sector/bps.html
https://www.census.gov/topics/employment/labor-force-statistics.html
https://www.census.gov/topics/business/manufacturing.html
https://www.census.gov/topics/international-trade/schedule-b.html
https://www.census.gov/topics/public-sector.html
https://www.census.gov/mobile
https://www.census.gov/topics/education/school-districts.html
http://www.pinterest.com/uscensusbureau
https://www.census.gov/library/audio.html
https://www.census.gov/topics/business/business-characteristics.html
https://www.census.gov/topics/housing/rental-housing.html
https://www.census.gov/geography/metro.html
https://www.census.gov/topics/income-poverty/saipe.html
https://www.census.gov/topics/families/data.html
https://www.census.gov/library/publications.html
https://www.census.gov/topics/international-trade/aes.html
https://www.census.gov/library.html
https://www.census.gov/topics/housing/financial-characteristics.html
https://www.census.gov/topics/employment/news.html
https://www.census.gov/topics/population/hispanic-origin.html
exporting links and HTML
c:\Python>
```

Fig1.1 Output of web scraper coded using the python library BeautifulSoup

## Part II – SQL

I.

See popDifferences.csv

J.

See popDifferencesPartJ.csv

140	•	select c.state,
141		c.pop15 as 'Population 2015',
142		c.pop16 as 'Population 2016',
143		case
144		when c.diff > 10000 then c.diff*-1
145		else c.diff
146		end as diff
147		from
148		(
149		select b.state as State,
150		b.pop15 as pop15,
151		b.pop16 as pop16,
152		round(abs(b.diff),-2) as diff
153		from (
154		select (a.pop16-a.pop15) as diff, a.state, a.pop15, a.pop16
155		from (
156		select
157		pop2015.state,
158		pop2015.population as pop15,
159		pop2016.population as pop16
160		from pop2015
161		inner join pop2016
162		on pop2015.state=pop2016.state
163		and pop2016.year=pop2016.year
164		) a
165		) b
166		where b.diff > 10000
167		) c ;
168		

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
state	Population 2015	Population 2016	diff
Arizona	6817565	6931071	-113500
Arkansas	2977853	2988248	-10400
California	38993940	39250017	-256100
Colorado	5448819	5540545	-91700
District of Columbia	670377	681170	-10800
Florida	20244914	20612439	-367500
Georgia	10199398	10310371	-111000
Idaho	1652828	1683140	-30300
Indiana	6612768	6633053	-20300
Iowa	3121997	3134693	-12700
Kentucky	4424611	4436974	-12400
Louisiana	4668960	4681666	-12700
Marvland	5994983	6016447	-21500
Massachusetts	6784240	6811779	-27500
Michigan	9917715	9928300	-10600
Minnesota	5482435	5519952	-37500
Missouri	6076204	6093000	-16800
Montana	1032073	1042520	-10400
Nebraska	1893765	1907116	-13400

Fig 2.1 Query used to combine the 2015 and 2016 population values ("nst-est2016-01", 2017) and query results below

K.

```

insert into pop2015(year, state, population) Values(2015,'Alabama',4853875);
insert into pop2015(year, state, population) Values(2015,'Alaska',737709);
insert into pop2015(year, state, population) Values(2015,'Arizona',6817565);
insert into pop2015(year, state, population) Values(2015,'Arkansas',2977853);
insert into pop2015(year, state, population) Values(2015,'California',38993940);
insert into pop2015(year, state, population) Values(2015,'Colorado',5448819);
insert into pop2015(year, state, population) Values(2015,'Connecticut',3584730);
insert into pop2015(year, state, population) Values(2015,'Delaware',944076);
insert into pop2015(year, state, population) Values(2015,'District of Columbia',670377);
insert into pop2015(year, state, population) Values(2015,'Florida',20244914);
insert into pop2015(year, state, population) Values(2015,'Georgia',10199398);
insert into pop2015(year, state, population) Values(2015,'Hawaii',1425157);
insert into pop2015(year, state, population) Values(2015,'Idaho',1652828);
insert into pop2015(year, state, population) Values(2015,'Illinois',12839047);
insert into pop2015(year, state, population) Values(2015,'Indiana',6612760);
insert into pop2015(year, state, population) Values(2015,'Iowa',3121997);
insert into pop2015(year, state, population) Values(2015,'Kansas',2906721);
insert into pop2015(year, state, population) Values(2015,'Kentucky',4424611);
insert into pop2015(year, state, population) Values(2015,'Louisiana',4668960);
insert into pop2015(year, state, population) Values(2015,'Maine',1329453);
insert into pop2015(year, state, population) Values(2015,'Maryland',5994983);
insert into pop2015(year, state, population) Values(2015,'Massachusetts',6784240);
insert into pop2015(year, state, population) Values(2015,'Michigan',9917715);
insert into pop2015(year, state, population) Values(2015,'Minnesota',5482435);
insert into pop2015(year, state, population) Values(2015,'Mississippi',2989390);
insert into pop2015(year, state, population) Values(2015,'Missouri',6076204);
insert into pop2015(year, state, population) Values(2015,'Montana',1032073);
insert into pop2015(year, state, population) Values(2015,'Nebraska',1893765);
insert into pop2015(year, state, population) Values(2015,'Nevada',2883758);
insert into pop2015(year, state, population) Values(2015,'New Hampshire',1330111);
insert into pop2015(year, state, population) Values(2015,'New Jersey',8935421);
insert into pop2015(year, state, population) Values(2015,'New Mexico',2080328);
insert into pop2015(year, state, population) Values(2015,'New York',19747183);
insert into pop2015(year, state, population) Values(2015,'North Carolina',10035186);
insert into pop2015(year, state, population) Values(2015,'North Dakota',756835);
insert into pop2015(year, state, population) Values(2015,'Ohio',11605090);
insert into pop2015(year, state, population) Values(2015,'Oklahoma',3907414);
insert into pop2015(year, state, population) Values(2015,'Oregon',4024634);
insert into pop2015(year, state, population) Values(2015,'Pennsylvania',12791904);
insert into pop2015(year, state, population) Values(2015,'Rhode Island',1055607);
insert into pop2015(year, state, population) Values(2015,'South Carolina',4894834);
insert into pop2015(year, state, population) Values(2015,'South Dakota',857919);

```

Fig 2.2 Sample of the insert statements used to create the tables needed for analysis in Fig 2.1



L.

	state	Population 2015	Population 2016	diff
	Arizona	6817565	6931071	-113500
	Arkansas	2977853	2988248	-10400
	California	38993940	39250017	-256100
	Colorado	5448819	5540545	-91700
	District of Columbia	670377	681170	-10800
	Florida	20244914	20612439	-367500
	Georgia	10199398	10310371	-111000
	Idaho	1652828	1683140	-30300
	Indiana	6612768	6633053	-20300
	Iowa	3121997	3134693	-12700
	Kentucky	4424611	4436974	-12400
	Louisiana	4668960	4681666	-12700
	Maryland	5994983	6016447	-21500
	Massachusetts	6784240	6811779	-27500
	Michigan	9917715	9928300	-10600
	Minnesota	5482435	5519952	-37500
	Missouri	6076204	6093000	-16800
	Montana	1032073	1042520	-10400
	Nebraska	1893765	1907116	-13400
	Nevada	2883758	2940058	-56300
	North Carolina	10035186	10146788	-111600
	Oklahoma	3907414	3923561	-16100
	Oregon	4024634	4093465	-68800
	South Carolina	4894834	4961119	-66300
	Tennessee	6595056	6651194	-56100
	Texas	27429639	27862596	-433000
	Utah	2990632	3051217	-60600
	Virginia	8367587	8411808	-44200
	Washington	7160290	7288000	-127700
	Wisconsin	5767891	5778708	-10800

Fig 2.3 Query Results showing the rounded difference between the population in 2016 and 2015 in states where the difference is greater than 10000

See sqlInsert.csv

The dataset was prepared using the current population estimates spreadsheet from the census website. There were periods leading all of the state names that were scrubbed from the data. Afterwards the cells containing state names and the cells containing their population were related using an excel formula. The excel formula would relate the cells and format them in a SQL insert statement. First the formula was applied to the 2015 data and then the 2016 data to create two sets of inserts for two tables. Here is one insert from each data set.

```
= "insert into pop2015(year, state, population) Values(2015,'"&M10&"', "&N10&");"
```

```
= "insert into pop2016(year, state, population) Values(2016,'"&M10&"', "&N10&");"
```

Here M was the column that containing the name of each state and N was the column containing population values for that year. First the N column contained 2015 population data. After the inserts were created, the N column was replaced with 2016 data and a new set of inserts were created. The formula needed to be slightly changed between sets to account for the new year and table name.

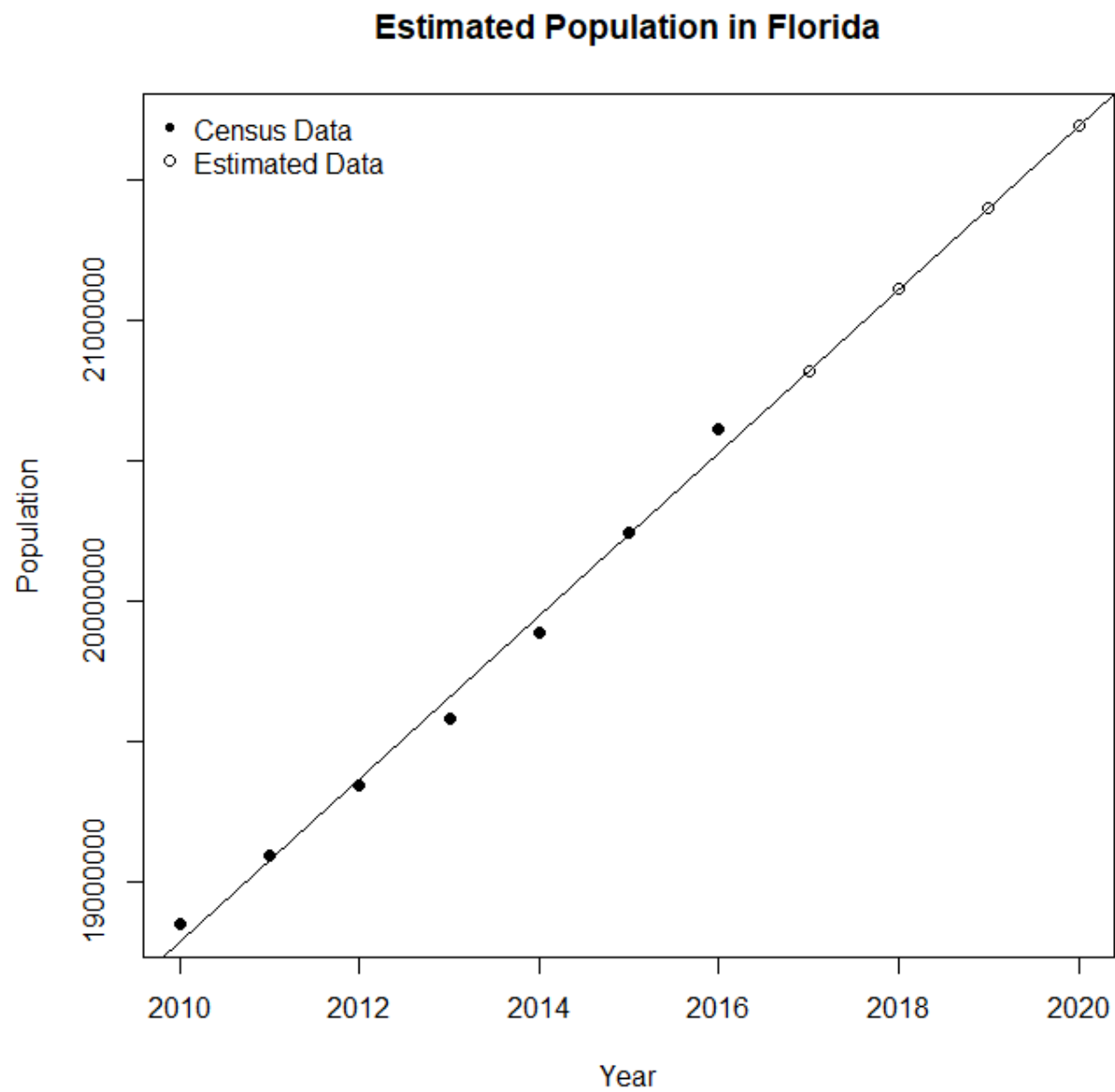
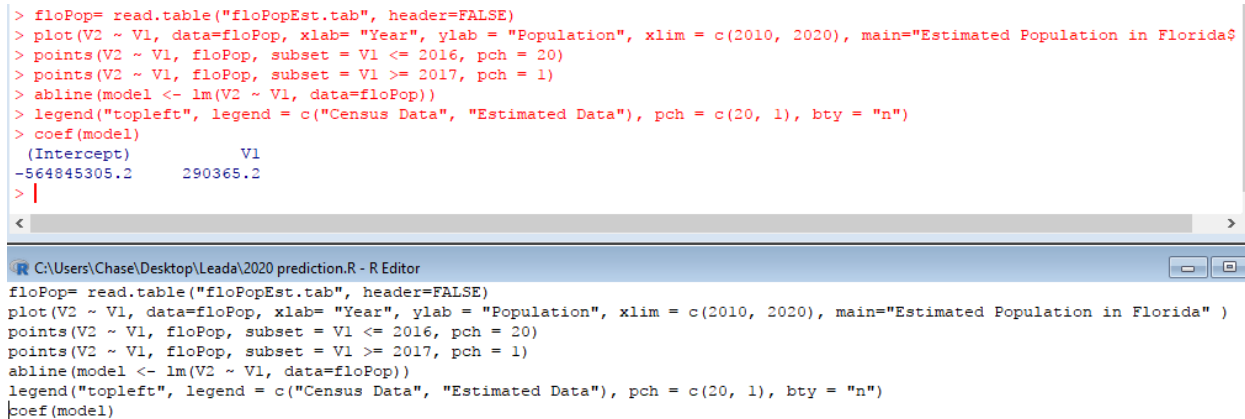


Fig 3.1 Estimated population in Florida for 2017-2020 based on the census data gathered from 2010-2016 ("nst-est2016-01", 2017)

```

> floPop= read.table("floPopEst.tab", header=FALSE)
> plot(V2 ~ V1, data=floPop, xlab= "Year", ylab = "Population", xlim = c(2010, 2020), main="Estimated Population in Florida$
> points(V2 ~ V1, floPop, subset = V1 <= 2016, pch = 20)
> points(V2 ~ V1, floPop, subset = V1 >= 2017, pch = 1)
> abline(model <- lm(V2 ~ V1, data=floPop))
> legend("topleft", legend = c("Census Data", "Estimated Data"), pch = c(20, 1), bty = "n")
> coef(model)
      (Intercept)           V1
-564845305.2      290365.2
> |

```



```

C:\Users\Chase\Desktop\Leada\2020 prediction.R - R Editor
floPop= read.table("floPopEst.tab", header=FALSE)
plot(V2 ~ V1, data=floPop, xlab= "Year", ylab = "Population", xlim = c(2010, 2020), main="Estimated Population in Florida" )
points(V2 ~ V1, floPop, subset = V1 <= 2016, pch = 20)
points(V2 ~ V1, floPop, subset = V1 >= 2017, pch = 1)
abline(model <- lm(V2 ~ V1, data=floPop))
legend("topleft", legend = c("Census Data", "Estimated Data"), pch = c(20, 1), bty = "n")
coef(model)

```

Fig 3.2 R script and console output for the plot in fig 3.1

Based on the linear regression in fig 3.1, one can infer the population of Florida would increase by 290,376 residents per year. By this logic, it is inferred that the population for Florida will be 21,692,307 in the year 2020.

N.

The data was prepared by selecting the population data for Florida 2010-2016 and adding it into a .tab file. All of the commas from the population values were purged with find/replace. Each population value was entered on a new row and the corresponding year was added in tandem on the same row, one tab apart.

See floPop.tab

O.

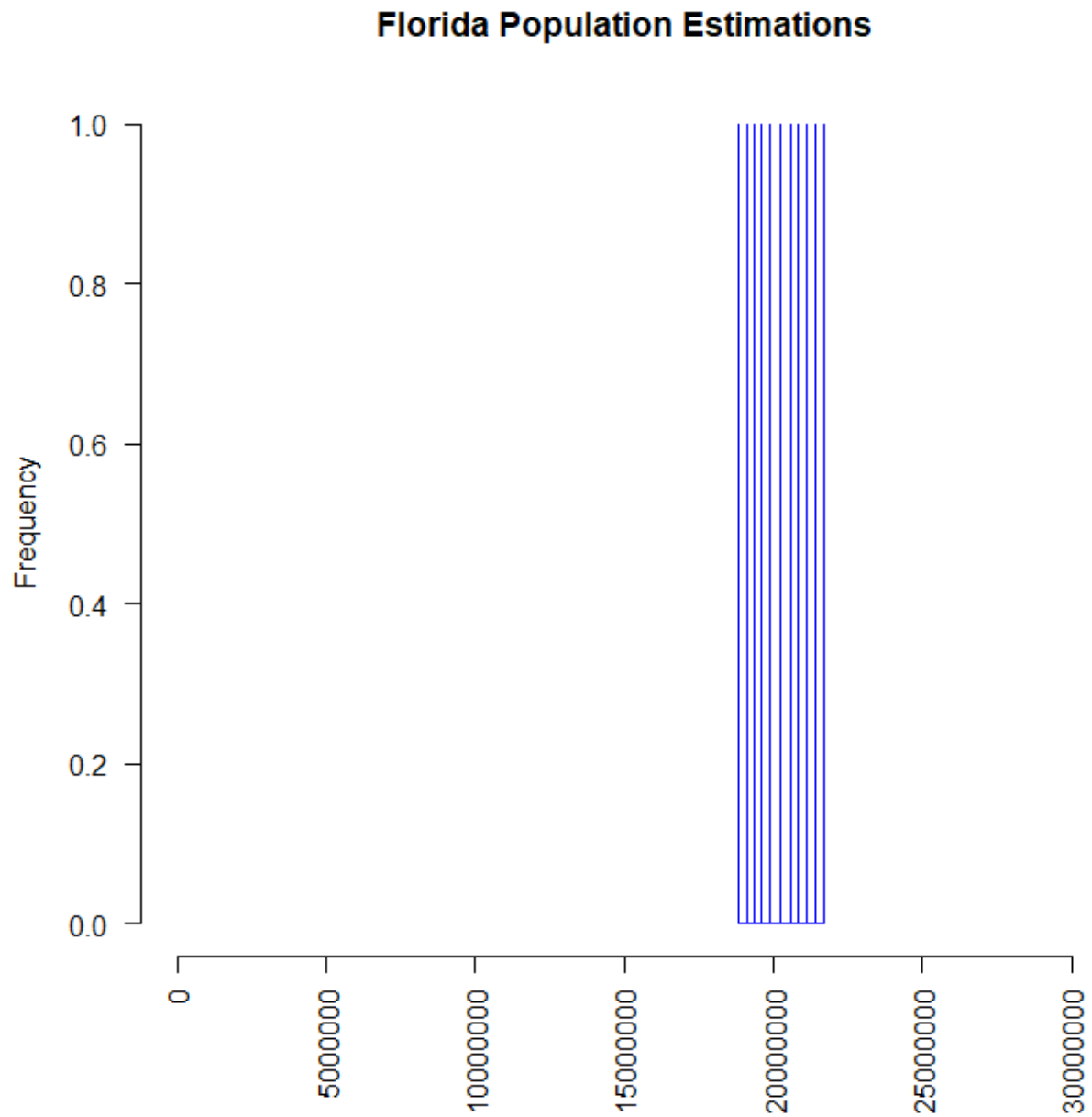
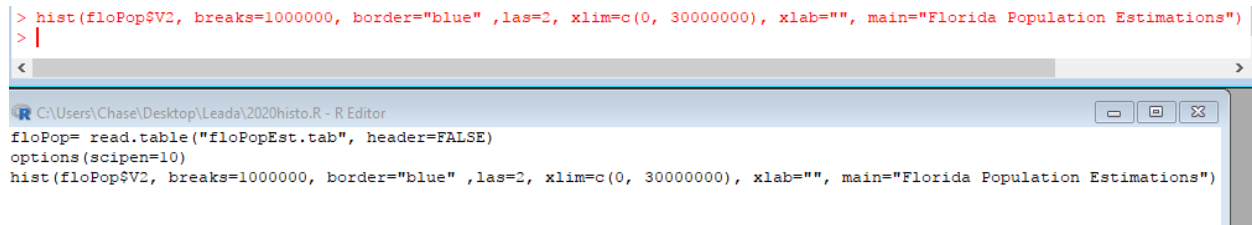


Fig 3.3 Histogram displaying dataset of census population information ("nst-est2016-01", 2017) as well as the estimated data.



```
> hist(floPop$V2, breaks=1000000, border="blue" ,las=2, xlim=c(0, 30000000), xlab="", main="Florida Population Estimations")
> |
```



The screenshot shows an R console window with a histogram command. Below it, an R Editor window is open, displaying the same script that was executed in the console. The script includes reading a table, setting options, and plotting a histogram.

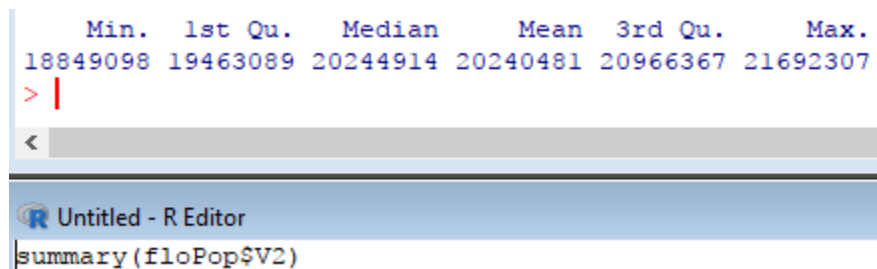
Fig 3.4 R script and console output for the histogram in fig 3.3

P.

```
      Min.   1st Qu.   Median     Mean  3rd Qu.     Max.
18849098 19463089 20244914 20240481 20966367 21692307
```

Fig 3.5 Summary statistics of the estimated data set

```
      Min.   1st Qu.   Median     Mean  3rd Qu.     Max.
18849098 19463089 20244914 20240481 20966367 21692307
> |
```



The screenshot shows an R console window with a summary command. Below it, an R Editor window is open, displaying the script that was executed in the console. The script includes a summary command for the data set.

Fig 3.6 R script and console output for the table in fig 3.5

Q.

Based on the linear regression in fig 3.1, one can infer the population of Florida would increase by 290,376 residents per year. By this logic, it is inferred that the population for Florida will be 20,821,178 in the year 2017.

R.

## Sources

United States Census Bureau. (2017). [Annual Estimates of the Resident Population for the United States, Regions, States, and Puerto Rico: April 1, 2010 to July 1, 2016, 2017](https://www.census.gov/data/tables/2016/demo/popest/state-total.html) [Table]. Retrieved from <https://www.census.gov/data/tables/2016/demo/popest/state-total.html>

