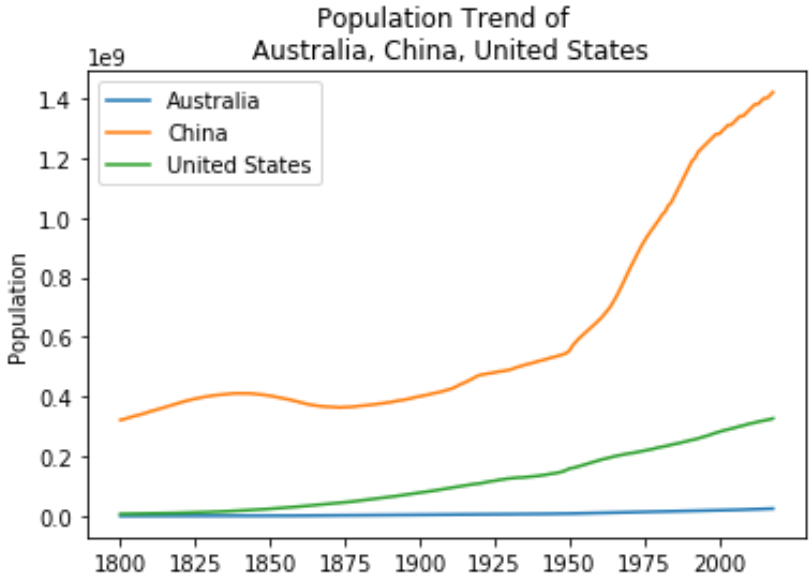
FIT5145 – Assignment 1

YUHAN ZHANG
29400988

Contents

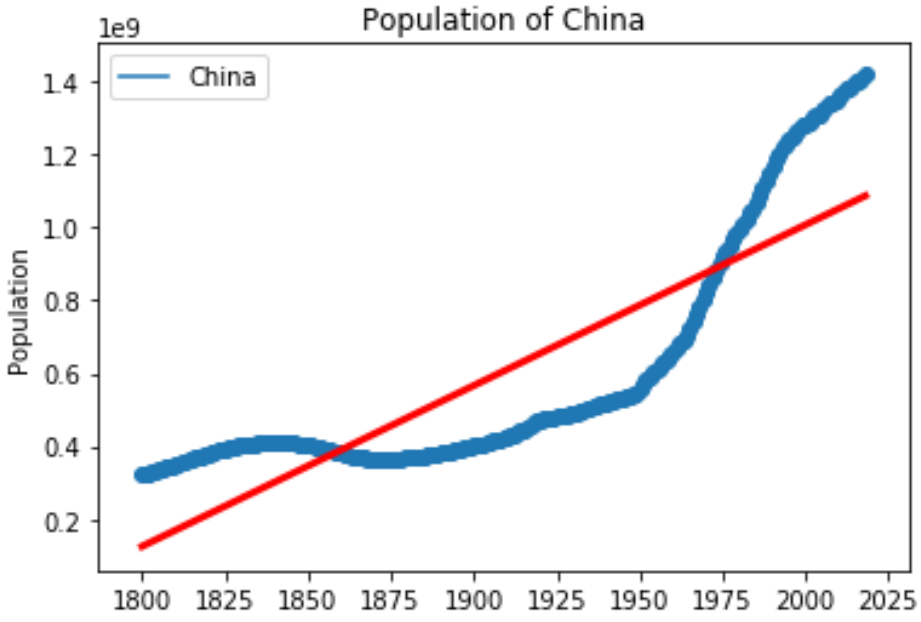
A1

1.



Although sometimes it declined, the Chinese population continues to grow over time. The people of the United States and Australia keep increasing almost no change.

2.



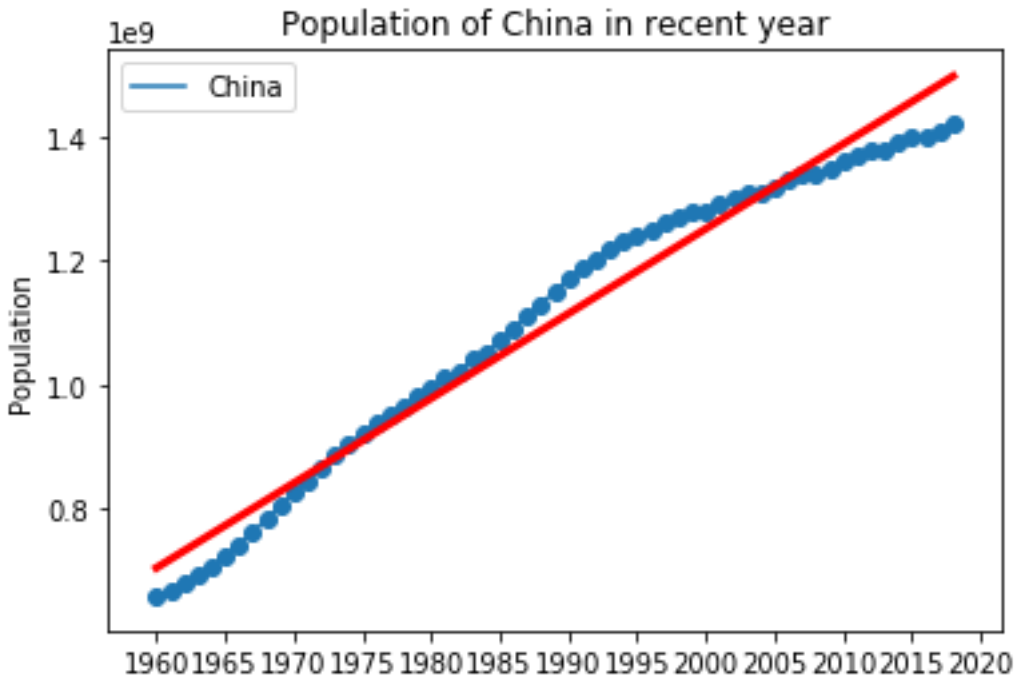
1. The linear fit not good might because r_value too large.

2.

```
In [4]: x2020 = slope*2020+intercept
        x2020

Out[4]: 1095698498.7489567

In [5]: x2100 = slope*2100+intercept
        x2100

Out[5]: 1448594862.1568203
```

2020: 1095698498.7489567
2100: 1448594862.1568203

3.

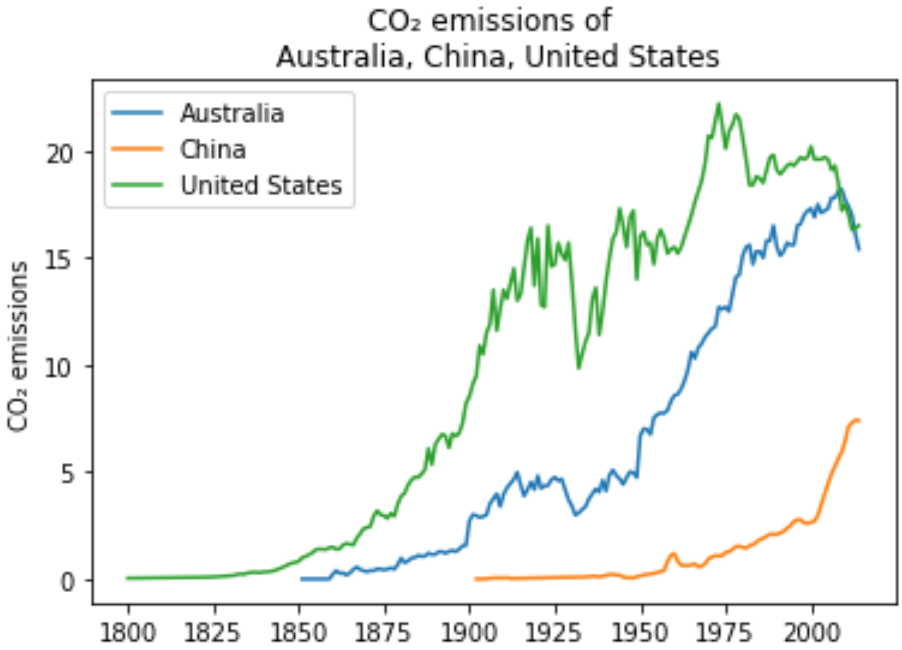


The line is fit.
The recent year model is better than the predicted line over time

A2

1.

2. Show the CO_2 emissions vs year.

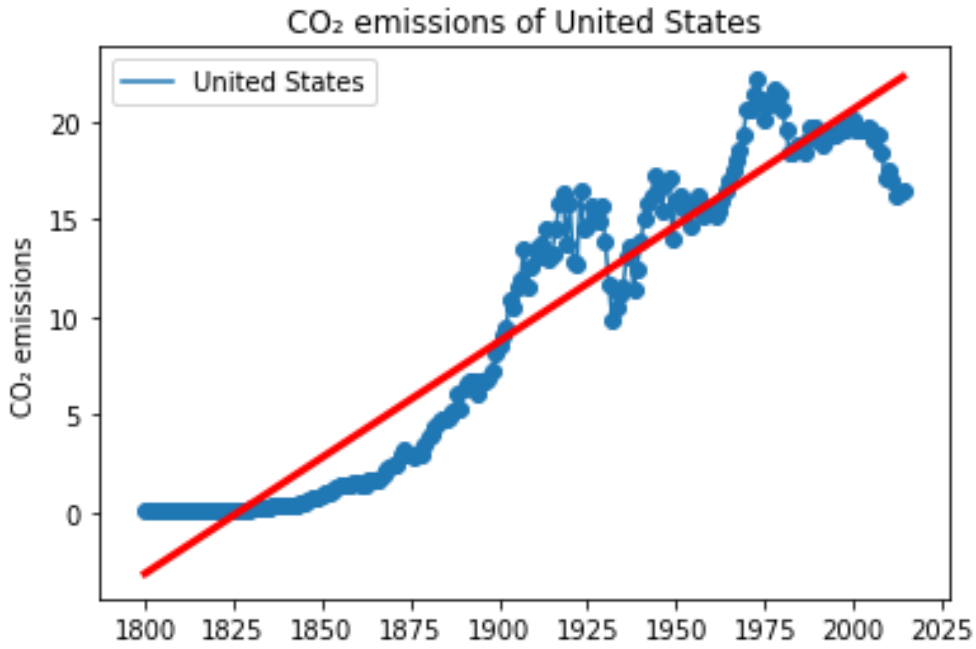


3. The trend of Australia and the United States are similar, but China's decline is rare.

4. Maximum values for CO 2 emissions (tonners per person) for Australia, China and the United States.

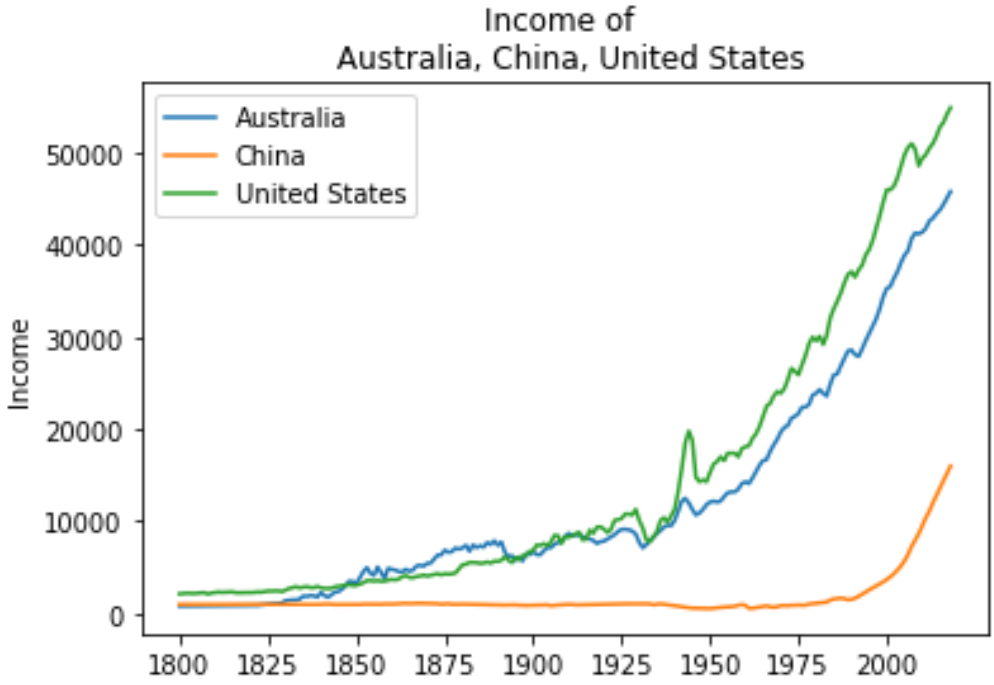
```
In [26]: co2_ACU['Australia'].max()
Out[26]: 18.2

In [27]: co2_ACU['China'].max()
Out[27]: 7.42

In [28]: co2_ACU['United States'].max()
Out[28]: 22.2
```

2.



This prediction line is not good. And the linear model is not suited to predict with long term and changeable trend data.

A3



```
#find the minimum income of year in China and match the income of year in Australia
income_MinChina = income_report_ACU.loc[[income_report['China'].idxmin()],['Australia','China']]
income_MinChina
```

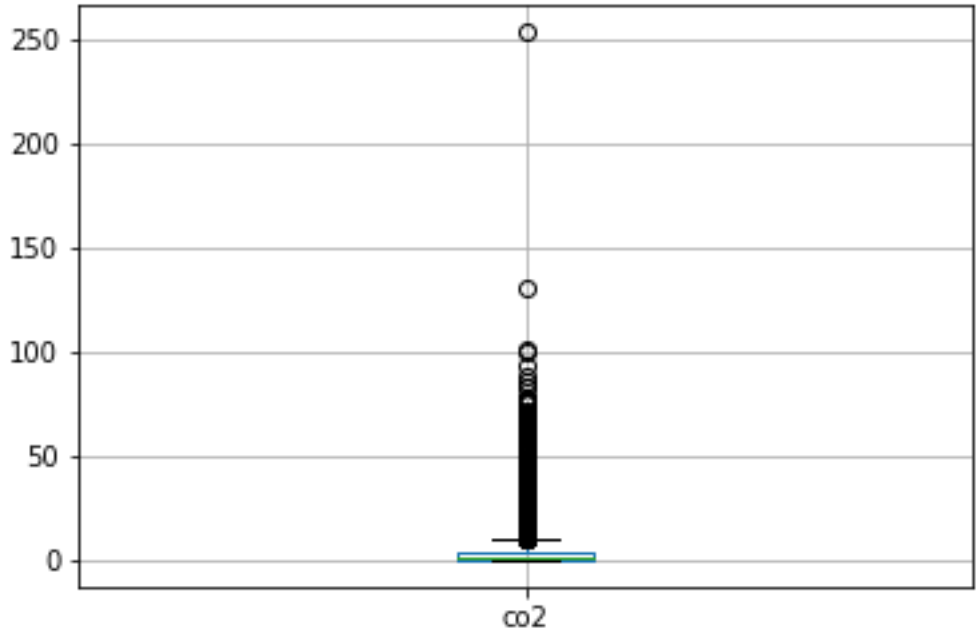
	Australia	China
1949	11800	530

A4

In order to make the statistical results more accurate, I made a box plot of carbon dioxide data for each year in each country and filtered extreme outliers. A dataframe called relationship_report without extreme outliers was created.

```
relation_column = np.asarray(sorted(list(set(population_reports.columns) | set(co2_reports.columns) | set(income_report.columns))))
relation_index = np.asarray(sorted(list(set(population_reports.index) | set(co2_reports.index) | set(income_report.index))))
relationship_report = pd.DataFrame(columns = pd.MultiIndex.from_product([relation_column,['population','co2','income']]),
                                   index = relation_index) #creat new multiple level df with every country and every year

#isnert data to cell with matching location
#for CO2 emissions, we only leave the value under 100 as reasonable data
for i in relation_column:
    for j in relation_index:

        if ((j in population_reports.index) & (i in population_reports.columns)):
            relationship_report.at[j, pd.IndexSlice[ i , 'population']] = population_reports.at[j,i]
        else:
            relationship_report.at[j, pd.IndexSlice[ i , 'population']] = np.nan

        if ((j in co2_reports.index) & (i in co2_reports.columns)):
            if co2_reports.at[j,i] < 100:
                relationship_report.at[j, pd.IndexSlice[ i , 'co2']] = co2_reports.at[j,i]
            else:
                relationship_report.at[j, pd.IndexSlice[ i , 'co2']] = np.nan
```

```
        else:
            relationship_report.at[j, pd.IndexSlice[ i , 'co2']] = np.nan


        if ((j in income_report.index) & (i in income_report.columns)):
            relationship_report.at[j, pd.IndexSlice[ i , 'income']] = income_report.at[j,i]
        else:
            relationship_report.at[j, pd.IndexSlice[ i , 'income']] = np.nan

relationship_report
```



1.

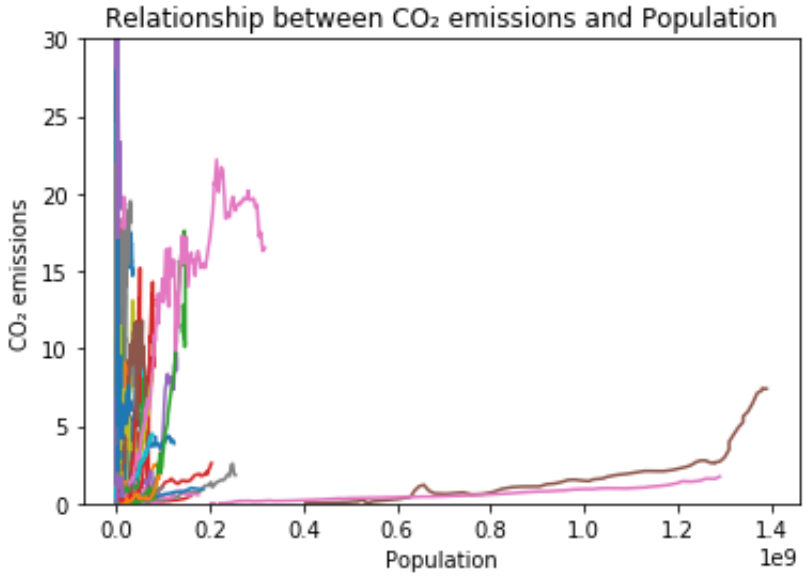
The first year is 1800. The last year in dataframe is 2018 but the last have CO 2 emissions record is 2014.

Out[33]:

	Afghanistan			Albania			Algeria			Andorra	...	Vietnam	Yemen			Zambia		
	population	co2	income	population	co2	income	population	co2	income	population	...	income	population	co2	income	population	co2	inco
1800	3280000	NaN	603	410000	NaN	667	2500000	NaN	715	2650	...	861	2590000	NaN	877	747000	NaN	
1801	3280000	NaN	603	412000	NaN	667	2510000	NaN	716	2650	...	861	2590000	NaN	879	747000	NaN	
1802	3280000	NaN	603	413000	NaN	667	2520000	NaN	717	2650	...	861	2590000	NaN	882	747000	NaN	
1803	3280000	NaN	603	414000	NaN	667	2530000	NaN	718	2650	...	861	2590000	NaN	884	747000	NaN	
1804	3280000	NaN	603	416000	NaN	667	2540000	NaN	719	2650	...	861	2590000	NaN	887	747000	NaN	

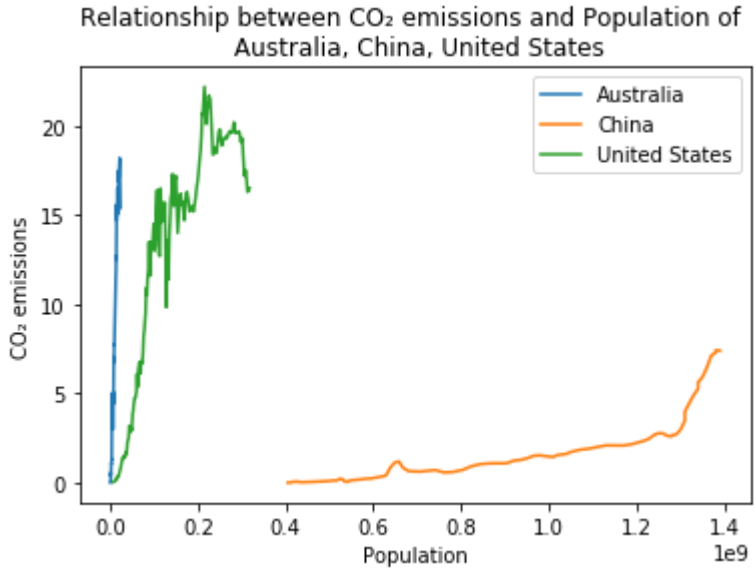
	Afghanistan			Albania			Algeria			Andorra	...	Vietnam	Yemen			Zambia		
	population	co2	income	population	co2	income	population	co2	income	population	...	income	population	co2	income	population	co2	i
2014	32800000	0.299	1780	2920000	1.96	10700	39100000	3.72	13500	79200	...	5370	26200000	0.865	3770	15600000	0.288	
2015	33700000	NaN	1750	2920000	NaN	11000	39900000	NaN	13700	78000	...	5670	26900000	NaN	2640	16100000	NaN	
2016	34700000	NaN	1740	2930000	NaN	11400	40600000	NaN	14000	77300	...	5960	27600000	NaN	2330	16600000	NaN	
2017	35500000	NaN	1800	2930000	NaN	11900	41300000	NaN	13800	77000	...	6250	28300000	NaN	2380	17100000	NaN	
2018	36400000	NaN	1870	2930000	NaN	12400	42000000	NaN	13700	77000	...	6550	28900000	NaN	2430	17600000	NaN	

2.



Carbon dioxide emissions may be positively correlated with population. In most countries, the more people there are, the greater the emissions. But the relationship may be misunderstood because the data is too dense and makes it difficult to compare. Some data differences are too large.

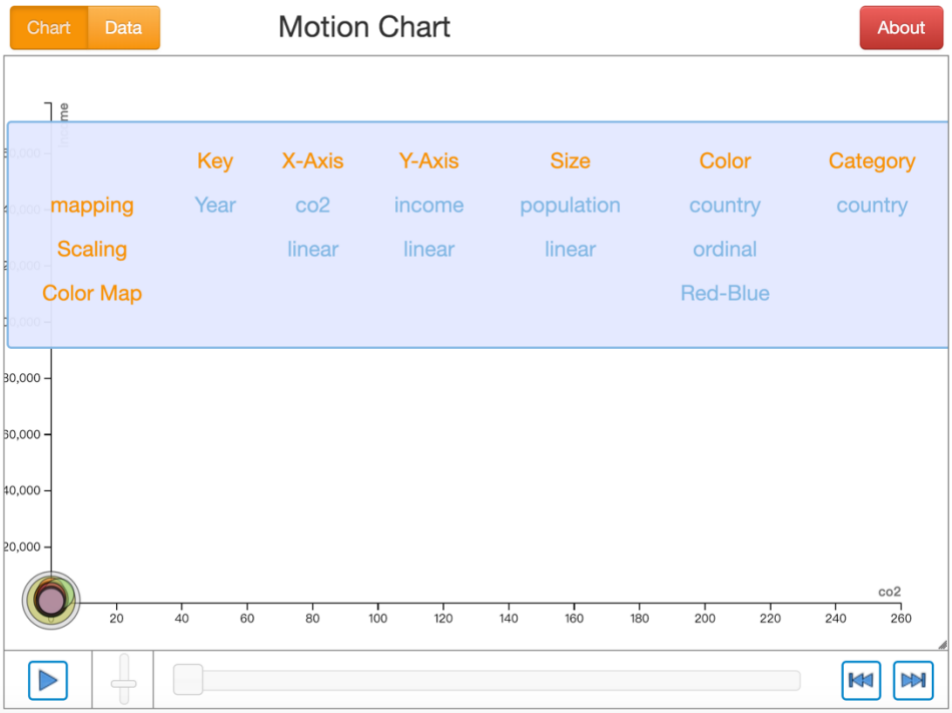
3.



Carbon dioxide emissions are positively correlated with the population, and in three countries, the higher the population, the greater the emissions. However, emissions in Australia and the United States began to decrease after the population had grown to a certain extent.

A5

1.



	Year	country	population	co2	income
42700	2014	Zimbabwe	15400000	0.78	1910.0
42701	2015	Zimbabwe	15800000	NaN	1890.0
42702	2016	Zimbabwe	16200000	NaN	1860.0
42703	2017	Zimbabwe	16500000	NaN	1910.0
42704	2018	Zimbabwe	16900000	NaN	1950.0

2.

1. The relationship is existing between CO 2 emissions and income. Most countries' incomes increase with the increase of CO 2 emissions, but in some countries, although incomes have risen in recent years, carbon emissions have decreased.

2. Income and carbon dioxide emissions have occasionally declined, but the entire period has grown.

3. Burundi have the minimum average CO2 emissions.

```
In [41]: des = co2_reports.describe()
         des.head()
```

Out[41]:

	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	United Kingdom	United States	Urugu
count	66.000000	82.000000	105.000000	25.000000	65.000000	58.000000	128.000000	163.000000	164.000000	197.000000	...	215.000000	215.000000	83.00
mean	0.126237	1.241449	1.336551	6.889600	0.520222	4.600655	2.138937	0.701549	6.766415	3.588878	...	8.044372	9.586467	1.36
std	0.096251	0.805333	1.401746	0.688655	0.362969	3.112242	1.417567	0.718731	6.058693	2.825096	...	3.070226	7.754755	0.76
min	0.001910	0.007070	0.000000	5.830000	0.041100	0.409000	0.266000	0.000060	0.000000	0.051700	...	2.130000	0.037200	0.02
25%	0.049475	0.572250	0.016900	6.430000	0.201000	3.647500	0.915750	0.059550	1.255000	1.170000	...	5.430000	1.180000	1.12

5 rows × 192 columns

```
In [42]: des.loc['mean',].idxmin()
Out[42]: 'Burundi'
```

4. Qatar has the maximum average CO2 emissions.

```
In [45]: des.loc['mean',].idxmax()
Out[45]: 'Qatar'
```

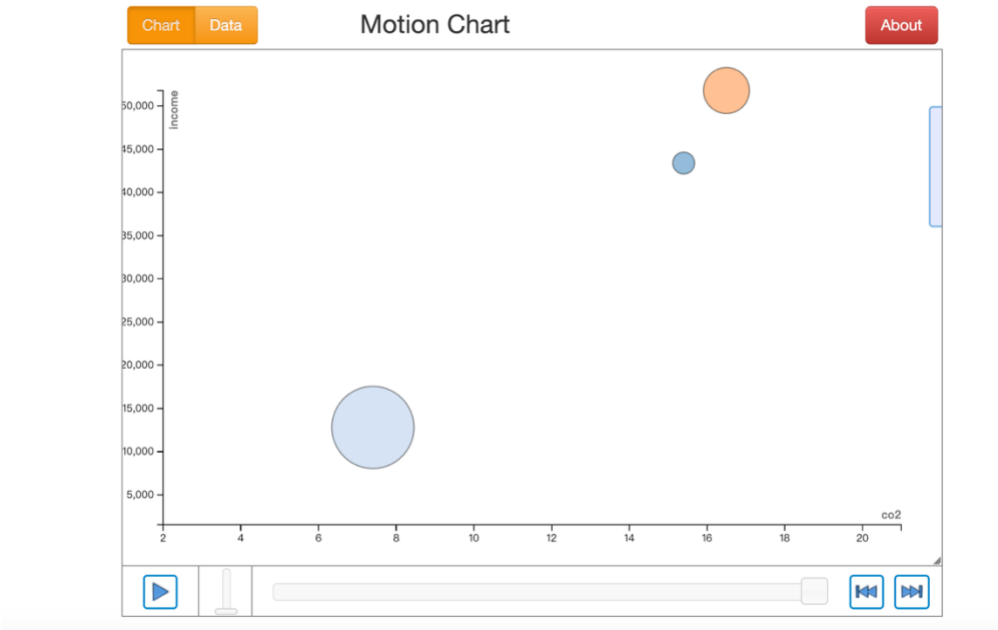
3.

The country missing the start in the recent year will stay at the origin point. And when the data appear, the target will move to location point dramatically.

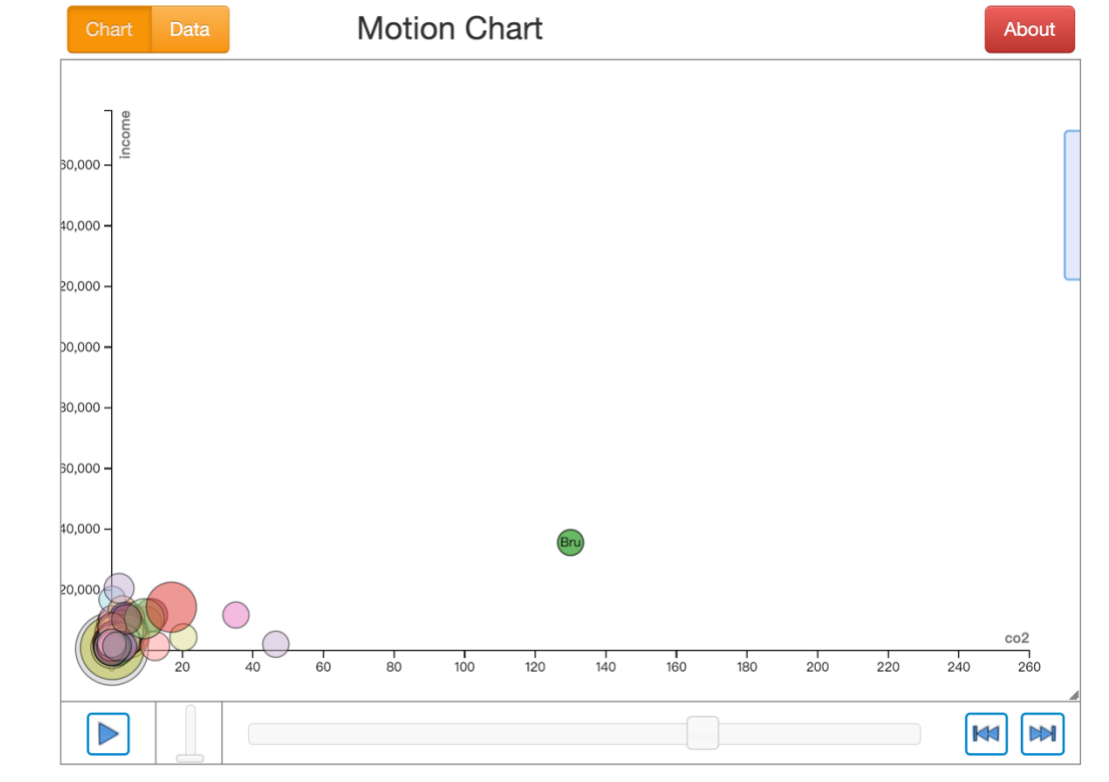
4.

```
filt = ((final_df['Year'] >= 1990) & (final_df['Year'] <= 2014)) & ((final_df['country'] == 'Australia') | (final_df['country'] == 'China') | (final_df['country'] == 'United States'))
```

At first, China had no data. Revenues in the US and Australia increase as carbon dioxide emissions increase. Then China also showed the same trend. At the end, China maintains this trend, but incomes in the US and Australia increase, and carbon dioxide emissions are decreasing.



5.



Brunei's data may have errors, so jumping in the picture is very irregular.

Task B

0. Note

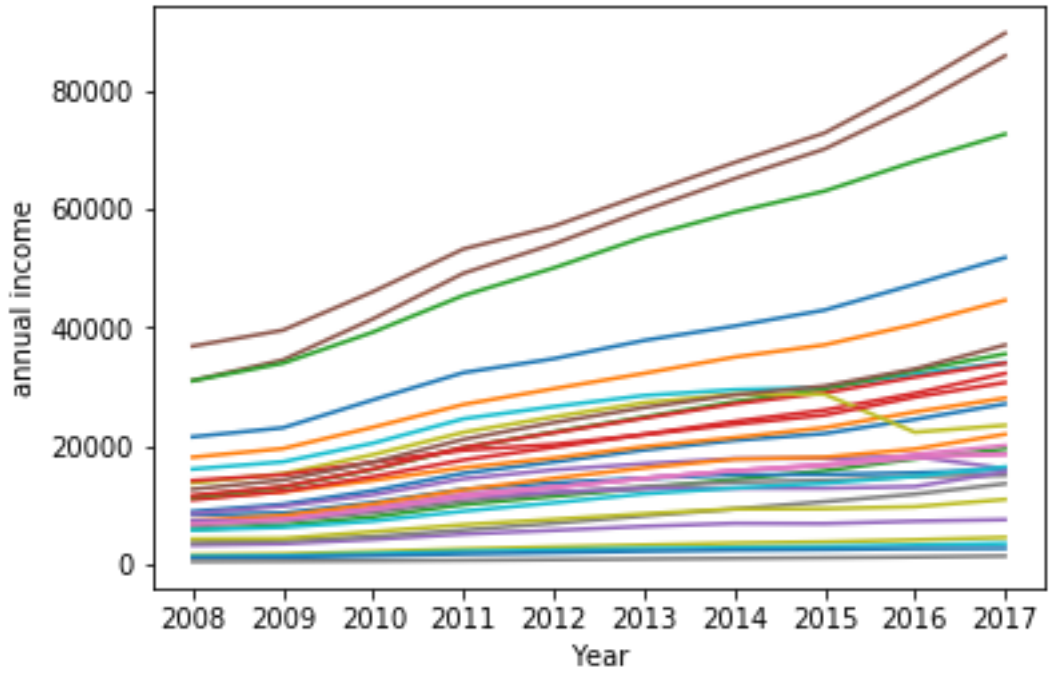
In task B, I will process data of China's higher education, population growth rate and annual income. The data from National Data of China (http://data.stats.gov.cn/english/
). There are three .csv files be used:

higher_education_population.csv
growth_rate.csv
annual_income.csv

The unit for data:

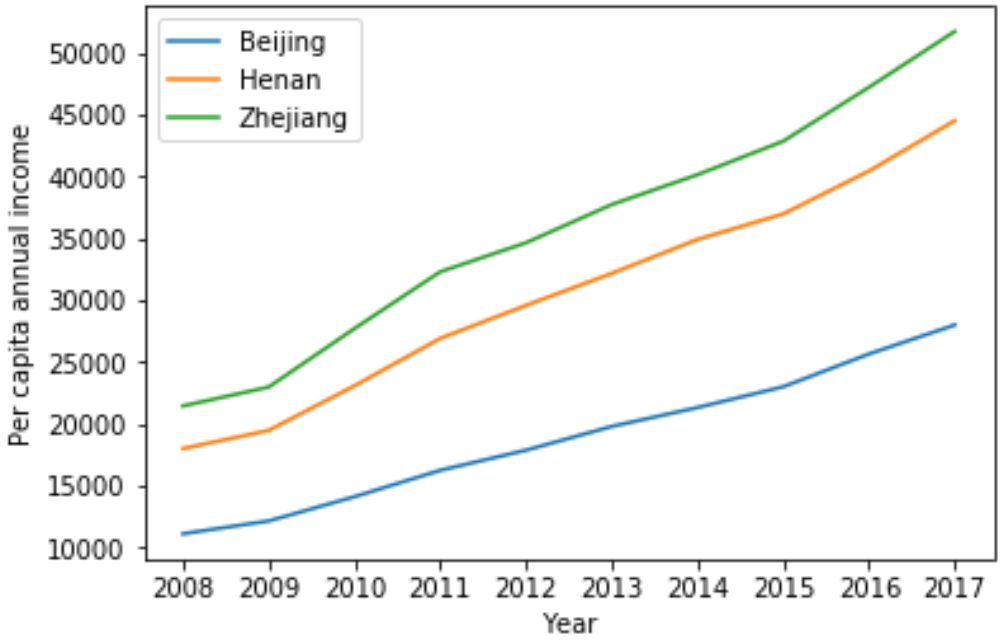
Higher education population: Number of higher education population per 100,000 people
Population growth rate: %
Annual income: $ per year

1. Analysis of annual income of various provinces in China



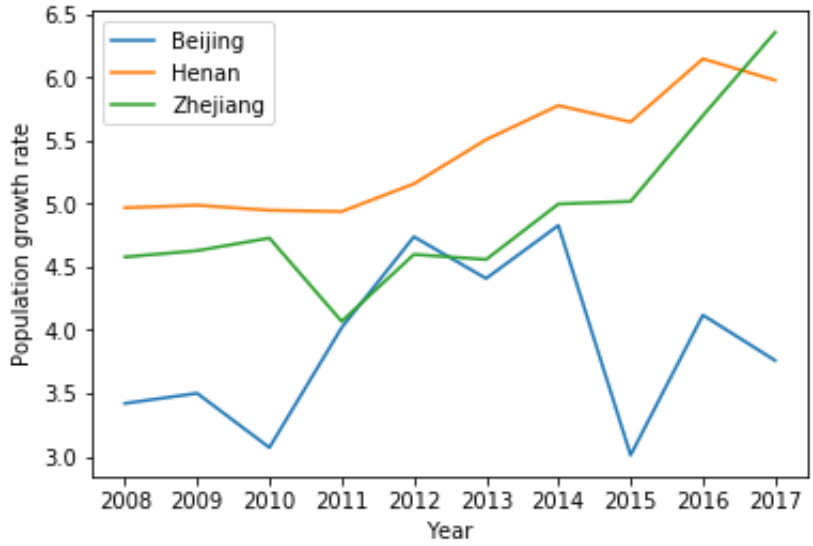
Most of province's resident annual income continues to grow over time.

2. Analysis of annual income in Beijing, Henan and Zhejiang



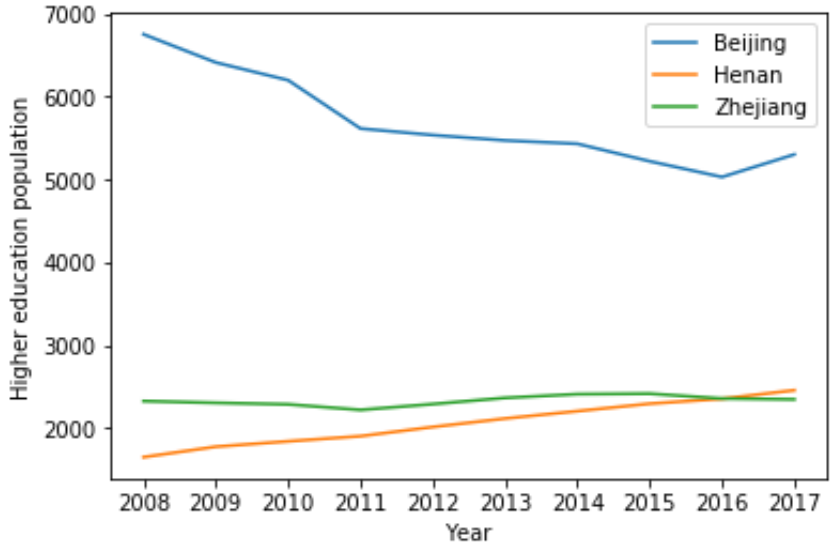
Beijing, Henan and Zhejiang province's resident annual income continue to grow over time.

3. Analysis of population growth rates in Beijing, Henan and Zhejiang



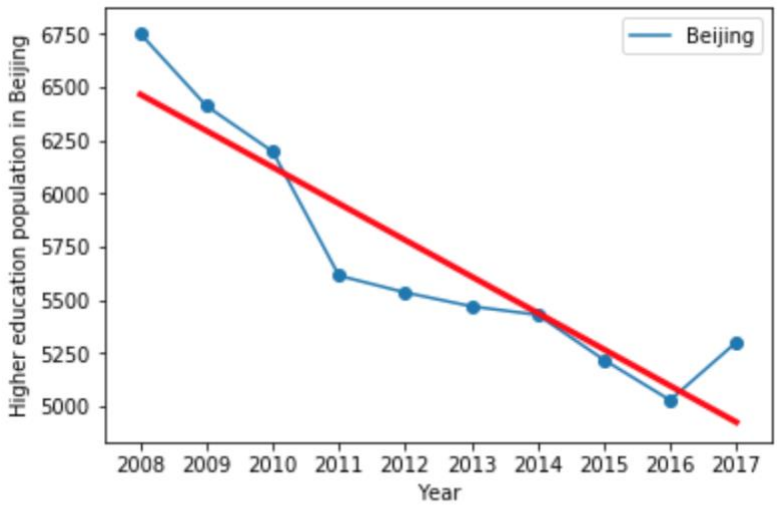
Beijing's population growth rate has increased or decreased over time, and the final population. The population growth rate in Henan and Zhejiang keeps growing, although there will be fluctuations during the period.

4. Analysis of the population of higher education in Beijing, Henan and Zhejiang



The background has the largest population of higher education, Henan has the fastest growth in data, and the data at the end is similar to that in Zhejiang Province.

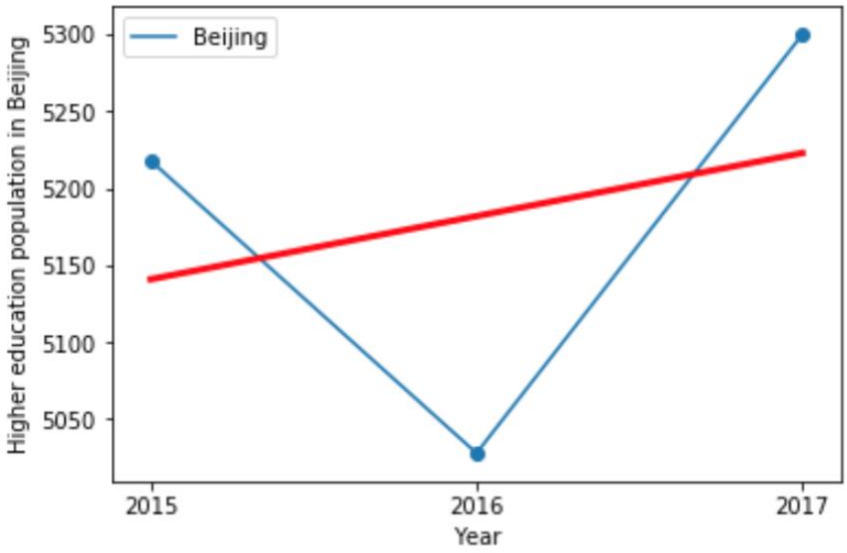
5. Prediction of the population of higher education in Beijing




```
In [9]: x2025 = slope*2025+intercept
        int(x2025)
```

```
Out[9]: 3555
```

Beijing's higher education population is showing a negative growth trend. According to this trend, the population of higher education in Beijing will be 3,555 in 2025.

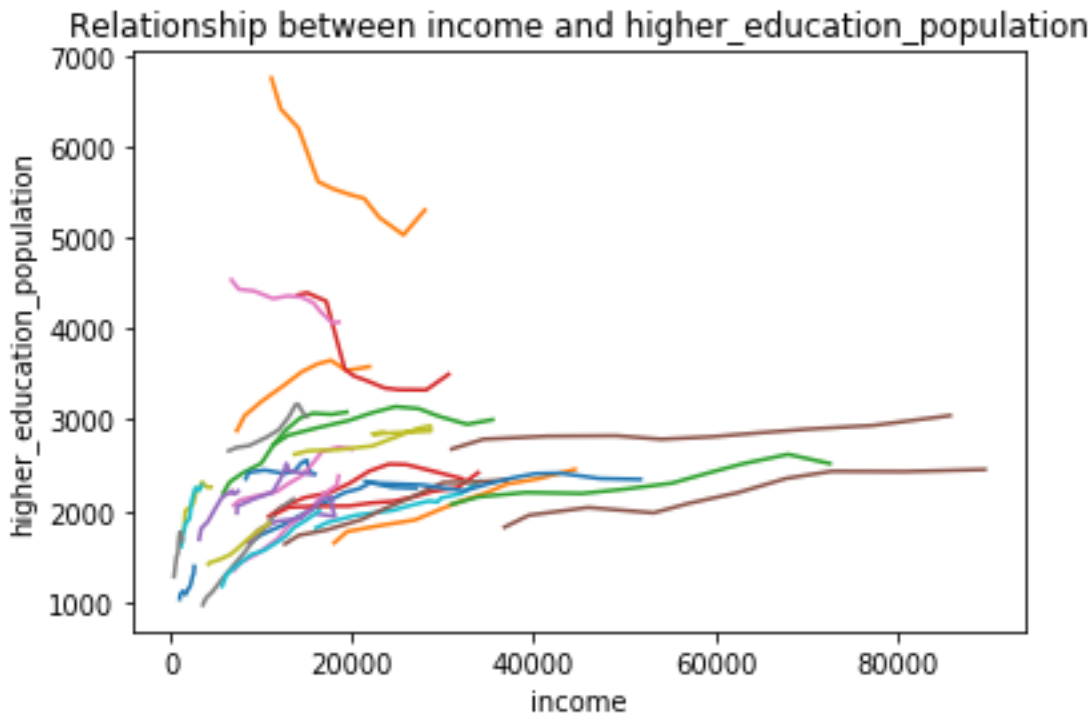



```
11]: x2025 = slope*2025+intercept
     int(x2025)
```

```
11]: 5551
```

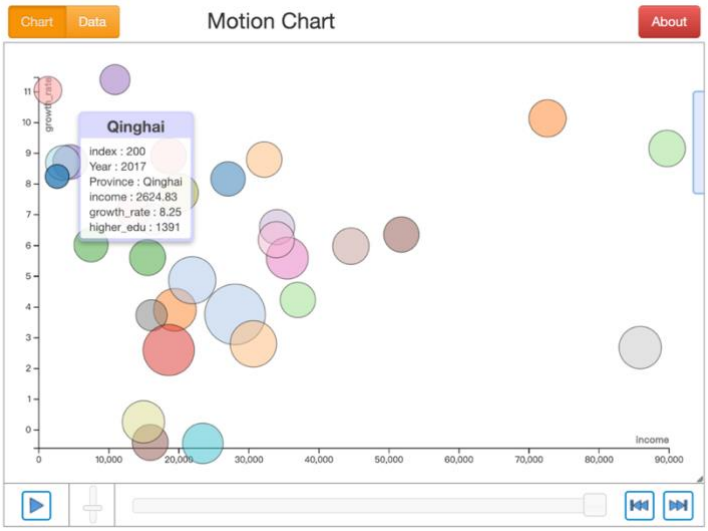
The population of higher education in Beijing is slowly increasing between 2015 and 2017. According to this trend, the population of higher education in Beijing will be 5,551 in 2025.

6. Analysis of the relationship between higher education population and annual income in Chinese provinces



Overall, with China's annual income increases, the number of higher education populations in the provinces approaches a similar range.

7. **Analysis of the relationship between population growth rate and income of higher education population in China's provinces**

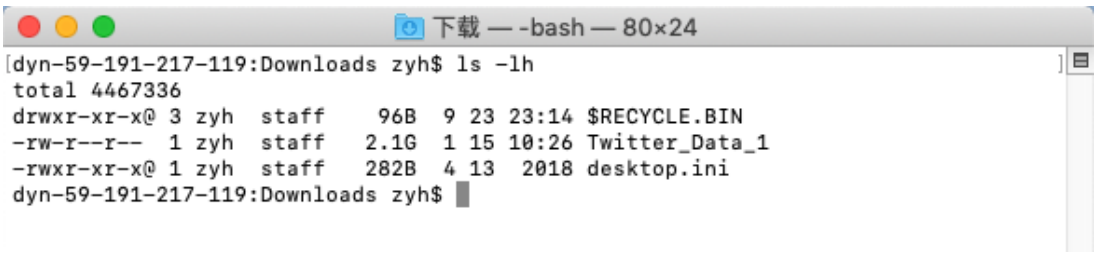


Through the analysis of the motion chart, the income growth rate is higher in areas with lower income growth.

Task C

C1

1.




```
1. gunzip Twitter_Data_1.gz
2. ls -lh
```

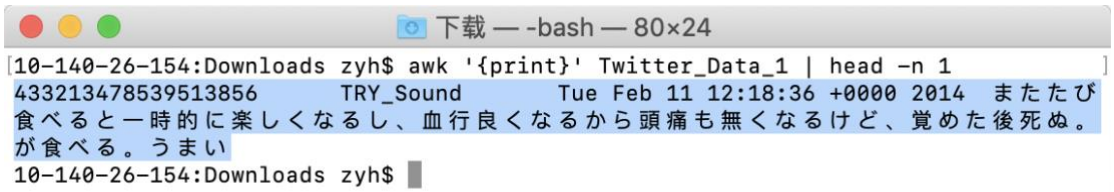
After decompressed the file the size of **Twitter_Data_1** is 2.1G

2.

1. awk -F' ' 'NR==1 {print NF}' Twitter_Data_1
2. awk -F'\t' 'NR==1 {print NF}' Twitter_Data_1

The delimiter could be ‘ ‘ or ‘\t. If separate the columns by ‘ ‘ , the number is 9 columns. If separate the columns by ‘\t’ , there are 4 columns. For **Twitter_Data_1**, separate to 4 columns by ‘\t’ is better choice for processing data.

3.



1. awk '{print}' Twitter_Data_1 | head -n 1

$2: User name of tweets $3: Post time of tweets
$4: Twitter content

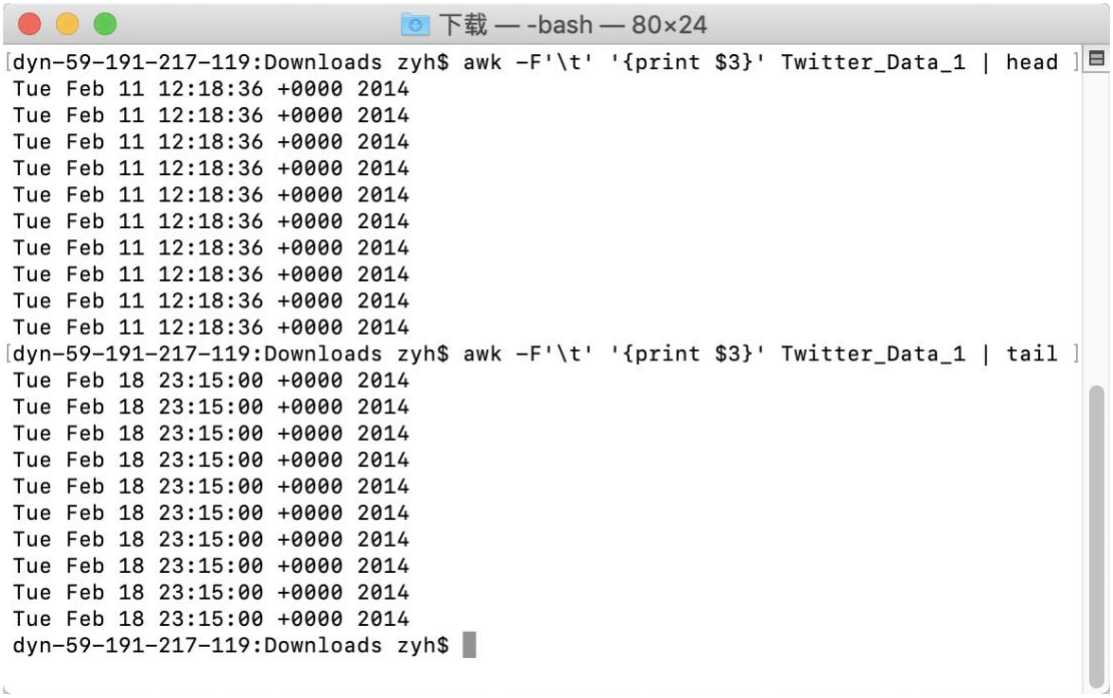
4.

```
dyn-59-191-217-119:Downloads zyh$ awk -F'\t' '{print}' Twitter_Data_1 | wc -l
  15089920
dyn-59-191-217-119:Downloads zyh$
```

1. awk -F'\t' '{print}' Twitter_Data_1 | wc -l

There are **15089920** tweets

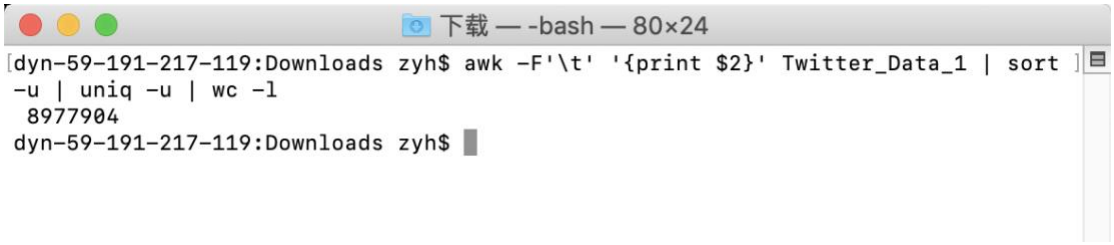
5.



1. awk -F'\t' '{print $3}' Twitter_Data_1 | head
2. awk -F'\t' '{print $3}' Twitter_Data_1 | tail

The data range is **Tue Feb 11 12:18:36 +0000 2014** to T**ue Feb 18 23:15:00 +0000 2014**.

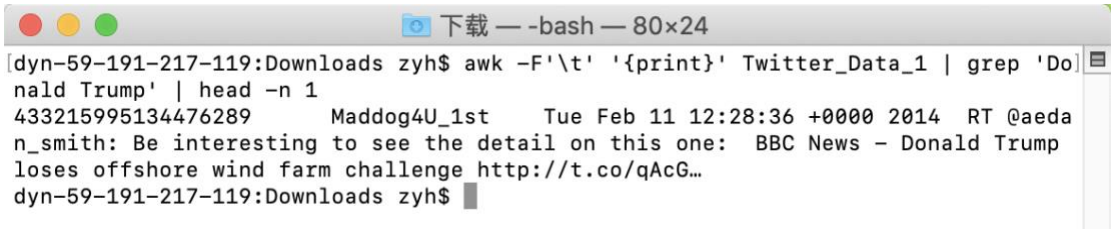
6.



1. awk -F'\t' '{print $2}' Twitter_Data_1 | sort | uniq | wc -l

There are **8977904** unique users

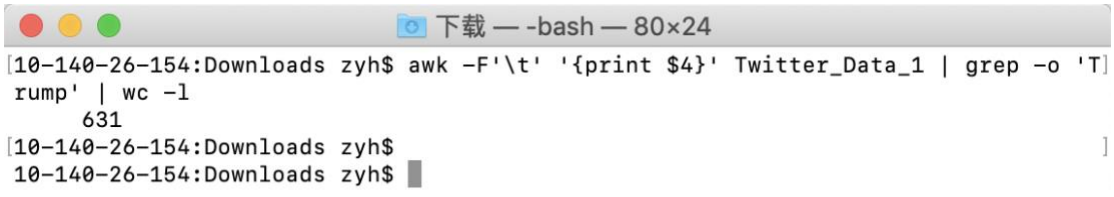
7.



1. awk -F'\t' '{print}' Twitter_Data_1 | grep 'Donald Trump' | head -n 1

The first time ‘Donald Trump’ appearing in **Twitter_Data_1** is **Tue Feb 11 12:28:36 +0000 2014**

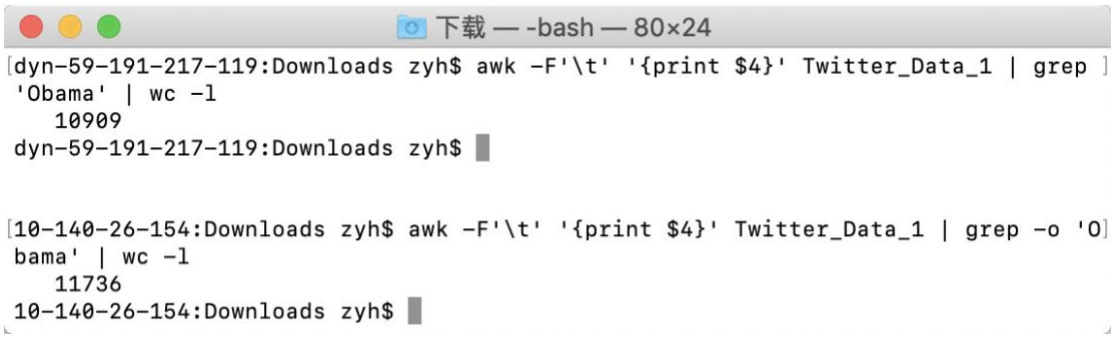
8.



1. awk -F'\t' '{print $4}' Twitter_Data_1 | grep 'Trump' | wc -l
2. awk -F'\t' '{print $4}' Twitter_Data_1 | grep -o 'Trump' | wc -l

Key word ‘Trump’ appeared in **Twitter_Data_1 631** times in **562** tweets.

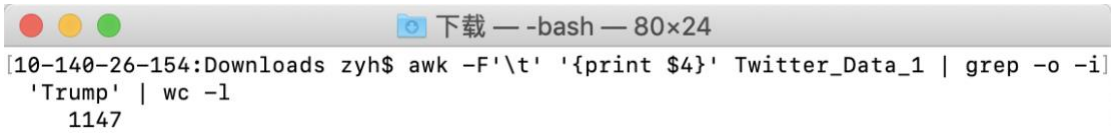
9.



1. awk -F'\t' '{print $4}' Twitter_Data_1 | grep 'Obama' | wc -l
2. awk -F'\t' '{print $4}' Twitter_Data_1 | grep -o 'Obama' | wc -l

Key word ‘Obama’ appeared in **Twitter_Data_1 11736** times in **10909** tweets.

10.



1. awk -F'\t' '{print $4}' Twitter_Data_1 | grep -o -i 'Trump' | wc -l

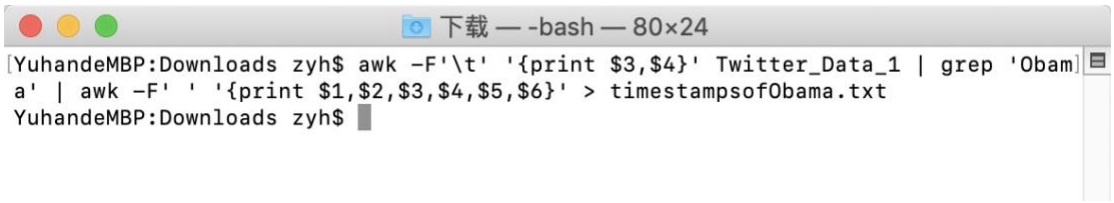
Till now, not all of references be captured, since the key word ‘Trump’ and ‘Obama’ cannot ignore the same word in upper case or lower case. Another reason is that these key words not note the first name, it means the result will lower than the real amount.

The key word needs to try: ‘Donald Trump’ ‘Barack Obama’
Note: ignore the upper case and lower case!!!

If we search by grep ignore the upper case and lower case. The result is **1147**.

C2

1.



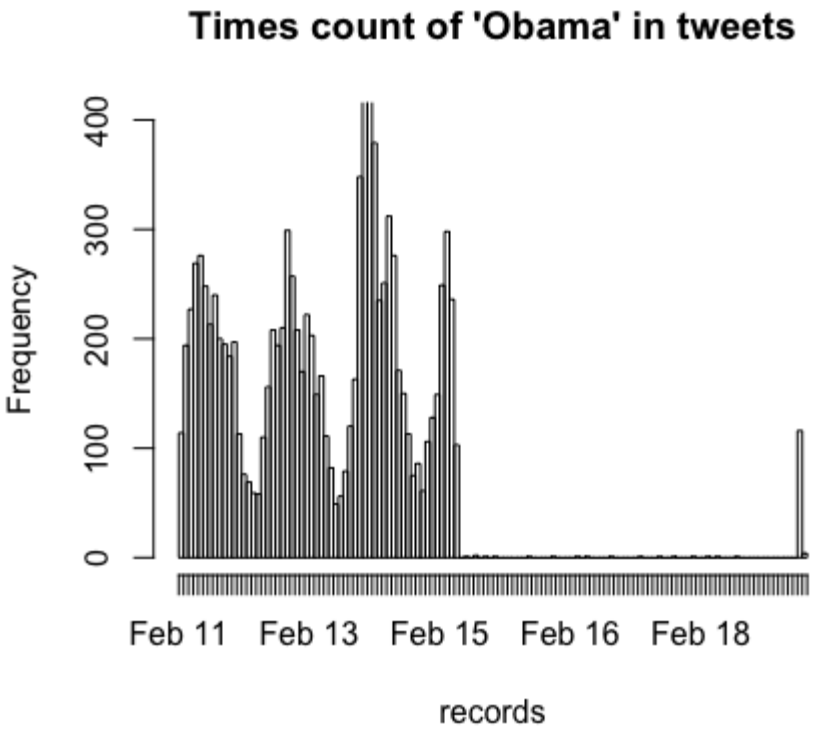

```
awk -F'\t' '{print $3,$4}' Twitter_Data_1 | grep 'Obama' | awk -
F' ' '{print $1,$2,$3,$4,$5,$6}' > timestampsofObama.txt
```

Output the time record related to ‘Obama’ to timestampsofObama.txt

```
> getwd()
> dir(
> setwd("/Users/zyh/Downloads")
> timesofObama<-read.table("timestampsofObama.txt", header = FALSE, sep = ",")
> records<-strptime(timesofObama$V1, format = "%a %b %d %H:%M:%S %z %Y")
```

Format string:
%a: Abbreviated weekday name
%b: Abbreviated month name
%d: Day of the month as decimal number (01–31)
%H: Hours as decimal number (00–23)
%M: Minute as decimal number (00–59).
%S: Second as integer (00–61)
%z: Signed offset in hours and minutes from UTC
%Y: Year with century

2.

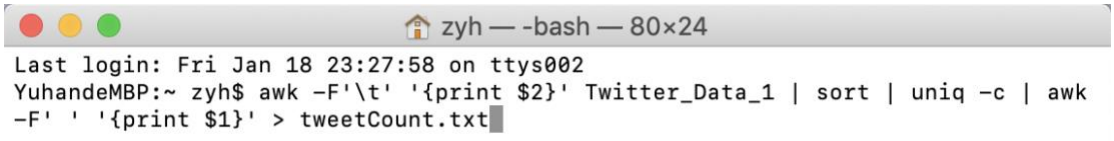



```
1.  >
    hist(records, breaks = 100, freq = TRUE, main="Times count of 'Obama' in tweets", yl
    im = c(0, 400))
```

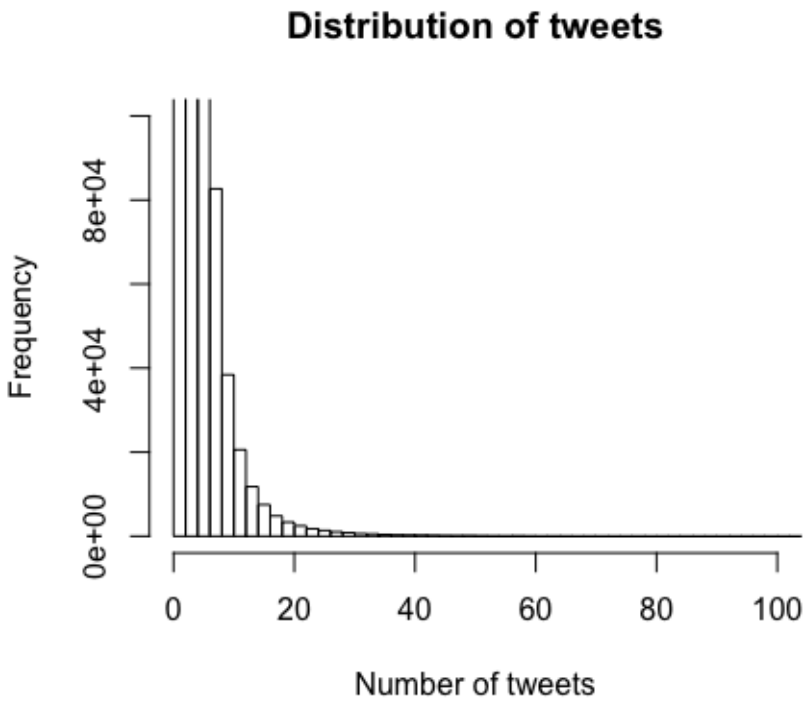
3.

The number of keywords has increased slightly between February 11 and February 13. Then it increased dramatically. Declined to the same number as on the 13 on February 15. The data disappeared and appeared again at the end of February 18.

4.



1. awk -F'\t' '{print $2}' Twitter_Data_1 | sort | uniq -c | awk -F' ' '{print $1}' > tweetCount.txt



5. Code:

```
#4.1
> getwd()
> dir(
> setwd("/Users/zyh/Downloads")
> timesofObama<-read.table("timestampsofObama.txt", header = FALSE, sep = ",")
> records<-strptime(timesofObama$V1, format = "%a %b %d %H:%M:%S %z %Y")

#4.2
hist(records, breaks = 100, freq = TRUE, main="Times count of 'Obama' in tweets", ylim = c(0, 400))

#4.4
> tweetCounts<-read.table("tweetCounts.txt", header = FALSE, sep = ",")
> hist(tweetCounts$V1, breaks = 100, freq = TRUE, plot = TRUE, xlim=c(0,100), ylim = c(0,100000),
xlab = "Number of tweets", main = "Distribution of tweets")
```