Sheet

Hassan Elwakeel 900211274

Assignment 2

Table of contents

Abstract

This notebook includes a csv folder "results.csv" which is for the first part. The first part has 43188 variables and 9 columns which includes some football results of international team whether they won or lost and in which tournament such as fifa world cup qualifications and it also shows the results of winning and losing on home pitch or away pitch. On the other hand the second part includes a csv folder "covid_data.csv" which has 122843 variables and 11 columns which includes the covid-19 data. especially the daily average cases in each region, country, continent and whether income affects the number of cases or not.

Part I

```
import pandas as pd
```

```
df=pd.read_csv('results.csv')
```

```
df
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False
...	...	...	...	...	...	...	...	...	...
43183	2/1/2022	Suriname	Guyana	2	1	Friendly	Paramaribo	Suriname	False
43184	2/2/2022	Burkina Faso	Senegal	1	3	African Cup of Nations	Yaoundé	Cameroon	True
43185	2/3/2022	Cameroon	Egypt	0	0	African Cup of Nations	Yaoundé	Cameroon	False
43186	2/5/2022	Cameroon	Burkina Faso	3	3	African Cup of Nations	Yaoundé	Cameroon	False
43187	2/6/2022	Senegal	Egypt	0	0	African Cup of Nations	Yaoundé	Cameroon	True

43188 rows × 9 columns

```
x= df['home_score']-df['away_score']
```

```
x
```

```
conditions=[(x<0),(x>0),(x==0)]
```

```
values=['lose','win','draw']
```

```
import numpy as np
```

```
df['result_home']=np.select(conditions,values)
```

```
df['result_home'][0:3]
```

```
df['result_home'].value_counts()
```

```
df['result_home'].value_counts(normalize=True)
```

```
X=df['result_home'].value_counts()
```

```
X=np.array(X)
```

```
X
```

```
X.sum()
```

```
43188
```

```
df.head()
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	False	draw
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False	win
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	False	win
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False	draw
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	False	win

```
df_noneutral=df[df['neutral']==False]
```

```
df_noneutral.shape
```

```
(32481, 10)
```

```
x=df_noneutral['result_home'].value_counts()
```

```
x
```

```
df_noneutral['result_home'].value_counts(normalize=True)
```

The below codes shows the winning games at home and then comes a graph that shows winning at home

```
df_noneutralegy=df_noneutral[df_noneutral['country']=='Egypt']
```

```
df_noneutralegy
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
1463	2/19/1932	Egypt	Hungary	0	0	Friendly	Cairo	Egypt	False	draw
1661	3/16/1934	Egypt	Israel	7	1	FIFA World Cup qualification	Cairo	Egypt	False	win
1895	6/19/1936	Egypt	Greece	3	1	Friendly	Cairo	Egypt	False	win
2927	12/24/1948	Egypt	Norway	1	1	Friendly	Cairo	Egypt	False	draw
3080	2/17/1950	Egypt	Greece	2	0	Friendly	Cairo	Egypt	False	win
...	...	...	...	...	...	...	...	...	...	...
42165	3/29/2021	Egypt	Comoros	4	0	African Cup of Nations qualification	Cairo	Egypt	False	win
42590	9/1/2021	Egypt	Angola	1	0	FIFA World Cup qualification	Cairo	Egypt	False	win
42758	9/30/2021	Egypt	Liberia	2	0	Friendly	Alexandria	Egypt	False	win
42812	10/8/2021	Egypt	Libya	1	0	FIFA World Cup qualification	Alexandria	Egypt	False	win
43040	11/16/2021	Egypt	Gabon	2	1	FIFA World Cup qualification	Alexandria	Egypt	False	win

258 rows × 10 columns

```
df_noneutralegy.shape
```

```
(258, 10)
```

```
df_noneutralegy['result_home'].value_counts(normalize=True) #do this for every country
```

```
0.662791/0.166667
```

```
3.976738046523907
```

```
import statsmodels.api as sm
from statsmodels.stats.proportion import proportion_confint
```

```
df_noneutralegy['result_home'].value_counts()
```

```
df_noneutralegy.shape
```

```
(258, 10)
```

```
CI_egy=proportion_confint(count=171,nobs=258,alpha=(1-.95))
CI_egy
```

```
(0.6051039288730269, 0.7204774664758102)
```

```
df_noneutralgre=df_noneutral[df_noneutral['country']=='Greece']
```

```
df_noneutralgre
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
1268	1/26/1930	Greece	Yugoslavia	2	1	Balkan Cup	Athens	Greece	False	win
1364	12/7/1930	Greece	Bulgaria	6	1	Balkan Cup	Athens	Greece	False	win
1455	11/29/1931	Greece	Romania	2	4	Balkan Cup	Athens	Greece	False	lose
1472	3/27/1932	Greece	Bulgaria	1	2	Friendly	Athens	Greece	False	lose
1473	3/30/1932	Greece	Bulgaria	2	2	Friendly	Athens	Greece	False	draw
...	...	...	...	...	...	...	...	...	...	...
42128	3/28/2021	Greece	Honduras	2	1	Friendly	Thessaloniki	Greece	False	win
42203	3/31/2021	Greece	Georgia	1	1	FIFA World Cup qualification	Thessaloniki	Greece	False	draw
42737	9/8/2021	Greece	Sweden	2	1	FIFA World Cup qualification	Athens	Greece	False	win
42946	11/11/2021	Greece	Spain	0	1	FIFA World Cup qualification	Athens	Greece	False	lose
43005	11/14/2021	Greece	Kosovo	1	1	FIFA World Cup qualification	Athens	Greece	False	draw

270 rows × 10 columns

```
df_noneutralgre['result_home'].value_counts()
```

```
df_noneutralgre.shape
```

```
(270, 10)
```

```
CI_gre=proportion_confint(count=128,nobs=270,alpha=(1-.95))
CI_gre
```

```
(0.4145144974686389, 0.5336336506795093)
```

```
df_noneutralpor=df_noneutral[df_noneutral['country']=='Portugal']
```

```
df_noneutralpor
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
689	12/17/1922	Portugal	Spain	1	2	Friendly	Lisbon	Portugal	False	lose
896	6/18/1925	Portugal	Italy	1	0	Friendly	Lisbon	Portugal	False	win
1028	12/26/1926	Portugal	Hungary	3	3	Friendly	Porto	Portugal	False	draw
1035	3/16/1927	Portugal	France	4	0	Friendly	Lisbon	Portugal	False	win
1111	1/8/1928	Portugal	Spain	2	2	Friendly	Lisbon	Portugal	False	draw
...	...	...	...	...	...	...	...	...	...	...
42373	6/9/2021	Portugal	Israel	4	0	Friendly	Lisbon	Portugal	False	win
42594	9/1/2021	Portugal	Republic of Ireland	2	1	FIFA World Cup qualification	Faro-Loulé	Portugal	False	win
42834	10/9/2021	Portugal	Qatar	3	0	Friendly	Faro-Loulé	Portugal	False	win
42882	10/12/2021	Portugal	Luxembourg	5	0	FIFA World Cup qualification	Faro-Loulé	Portugal	False	win
43002	11/14/2021	Portugal	Serbia	1	2	FIFA World Cup qualification	Lisbon	Portugal	False	lose

296 rows × 10 columns

```
df_noneutralpor['result_home'].value_counts()
```

```
df_noneutralpor.shape
```

```
(296, 10)
```

```
CI_por=proportion_confint(count=170,nobs=296,alpha=(1-.95))
CI_por
```

```
(0.5179968526717824, 0.6306517959768663)
```

```
df_noneutralusa=df_noneutral[df_noneutral['country']=='United States']
```

```
df_noneutralusa
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
41	1885-11-28	United States	Canada	0	1	Friendly	Newark	United States	False	lose
48	1886-11-25	United States	Canada	3	2	Friendly	Newark	United States	False	win
935	11/8/1925	United States	Canada	6	1	Friendly	New York	United States	False	win
1022	11/6/1926	United States	Canada	6	2	Friendly	New York	United States	False	win
1790	5/19/1935	United States	Scotland	1	5	Friendly	New York	United States	False	lose
...	...	...	...	...	...	...	...	...	...	...
42757	9/24/2021	El Salvador	Guatemala	0	2	Friendly	Washington	United States	False	lose
42776	10/7/2021	United States	Jamaica	2	0	FIFA World Cup qualification	Austin	United States	False	win
42908	10/13/2021	United States	Costa Rica	2	1	FIFA World Cup qualification	Columbus	United States	False	win
42966	11/12/2021	United States	Mexico	2	0	FIFA World Cup qualification	Cincinnati	United States	False	win
43103	12/18/2021	United States	Bosnia and Herzegovina	1	0	Friendly	Los Angeles	United States	False	win

413 rows × 10 columns

```
df_noneutralusa['result_home'].value_counts()
```

```
df_noneutralusa.shape
```

```
(413, 10)
```

```
CI_usa=proportion_confint(count=230,nobs=413,alpha=(1-.95))
CI_usa
```

```
(0.5089922370968902, 0.6048092156876135)
```

```
df_noneutralfr=df_noneutral[df_noneutral['country']=='France']
```

```
df_noneutralfr
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
166	2/12/1905	France	Switzerland	1	0	Friendly	Paris	France	False	win
185	4/22/1906	France	Belgium	0	5	Friendly	Saint-Cloud	France	False	lose
215	4/12/1908	France	Belgium	1	2	Friendly	Colombes	France	False	lose
257	4/3/1910	France	Belgium	0	4	Friendly	Gentilly	France	False	lose
277	1/1/1911	France	Hungary	0	3	Friendly	Maisons-Alfort	France	False	lose
...	...	...	...	...	...	...	...	...	...	...
42244	6/2/2021	France	Wales	3	0	Friendly	Nice	France	False	win
42346	6/8/2021	France	Bulgaria	3	0	Friendly	Paris	France	False	win
42597	9/1/2021	France	Bosnia and Herzegovina	1	1	FIFA World Cup qualification	Strasbourg	France	False	draw
42723	9/7/2021	France	Finland	2	0	FIFA World Cup qualification	Lyon	France	False	win
42988	11/13/2021	France	Kazakhstan	8	0	FIFA World Cup qualification	Paris	France	False	win

462 rows × 10 columns

```
df_noneutralfr['result_home'].value_counts()
```

```
df_noneutralfr.shape
```

```
(462, 10)
```

```
CI_fr=proportion_confint(count=268,nobs=462,alpha=(1-.95))
CI_fr
```

```
(0.5350823444775694, 0.6250908156955908)
```

```
df_noneutralmy=df_noneutral[df_noneutral['country']=='Malaysia']
```

```
df_noneutralmy
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
5674	8/22/1964	Malaysia	Thailand	3	0	Merdeka Tournament	Kuala Lumpur	Malaysia	False	win
5677	8/24/1964	Malaysia	India	1	1	Merdeka Tournament	Kuala Lumpur	Malaysia	False	draw
5680	8/28/1964	Malaysia	Myanmar	0	3	Merdeka Tournament	Kuala Lumpur	Malaysia	False	lose
5683	8/30/1964	Malaysia	Vietnam Republic	1	2	Merdeka Tournament	Kuala Lumpur	Malaysia	False	lose
5686	9/2/1964	Malaysia	Taiwan	5	2	Merdeka Tournament	Kuala Lumpur	Malaysia	False	win
...	...	...	...	...	...	...	...	...	...	...
41300	9/10/2019	Malaysia	United Arab Emirates	1	2	FIFA World Cup qualification	Kuala Lumpur	Malaysia	False	lose
41334	10/5/2019	Malaysia	Sri Lanka	6	0	Friendly	Kuala Lumpur	Malaysia	False	win
41516	11/9/2019	Malaysia	Tajikistan	1	0	Friendly	Kuala Lumpur	Malaysia	False	win
41571	11/14/2019	Malaysia	Thailand	2	1	FIFA World Cup qualification	Kuala Lumpur	Malaysia	False	win
41697	11/19/2019	Malaysia	Indonesia	2	0	FIFA World Cup qualification	Kuala Lumpur	Malaysia	False	win

275 rows × 10 columns

```
df_noneutralmy['result_home'].value_counts()
```

```
df_noneutralmy.shape
```

```
(275, 10)
```

```
CI_my=proportion_confint(count=129,nobs=275,alpha=(1-.95))
CI_my
```

```
(0.4101087952116519, 0.5280730229701663)
```

```
df_noneutraleng=df_noneutral[df_noneutral['country']=='England']
```

```
df_noneutraleng
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	False	win
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	False	draw
6	1877-03-03	England	Scotland	1	3	Friendly	London	England	False	lose
10	1879-01-18	England	Wales	2	1	Friendly	London	England	False	win
11	1879-04-05	England	Scotland	5	4	Friendly	London	England	False	win
...	...	...	...	...	...	...	...	...	...	...
42521	7/7/2021	England	Denmark	2	1	UEFA Euro	London	England	False	win
42533	7/11/2021	England	Italy	1	1	UEFA Euro	London	England	False	draw
42680	9/5/2021	England	Andorra	4	0	FIFA World Cup qualification	London	England	False	win
42893	10/12/2021	England	Hungary	1	1	FIFA World Cup qualification	London	England	False	draw
42972	11/12/2021	England	Albania	5	0	FIFA World Cup qualification	London	England	False	win

499 rows × 10 columns

```
df_noneutraleng['result_home'].value_counts()
```

```
df_noneutraleng.shape
```

```
(499, 10)
```

```
CI_eng=proportion_confint(count=319,nobs=499,alpha=(1-.95))
CI_eng
```

```
(0.5971449281308193, 0.6814121860976375)
```

```
df_noneutralse=df_noneutral[df_noneutral['country']=='Sweden']
```

```
df_noneutralse
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
224	7/12/1908	Sweden	Norway	11	3	Friendly	Gothenburg	Sweden	False	win
296	6/18/1911	Sweden	Germany	2	4	Friendly	Solna	Sweden	False	lose
298	9/17/1911	Sweden	Norway	4	1	Friendly	Solna	Sweden	False	win
328	6/20/1912	Sweden	Hungary	2	2	Friendly	Gothenburg	Sweden	False	draw
330	6/27/1912	Sweden	Finland	7	1	Friendly	Solna	Sweden	False	win
...	...	...	...	...	...	...	...	...	...	...
42301	6/5/2021	Sweden	Armenia	3	1	Friendly	Stockholm	Sweden	False	win
42623	9/2/2021	Sweden	Spain	2	1	FIFA World Cup qualification	Stockholm	Sweden	False	win
42670	9/5/2021	Sweden	Uzbekistan	2	1	Friendly	Stockholm	Sweden	False	win
42815	10/9/2021	Sweden	Kosovo	3	0	FIFA World Cup qualification	Stockholm	Sweden	False	win
42885	10/12/2021	Sweden	Greece	2	0	FIFA World Cup qualification	Stockholm	Sweden	False	win

483 rows × 10 columns

```
df_noneutralse['result_home'].value_counts()
```

```
df_noneutralse.shape
```

```
(483, 10)
```

```
CI_se=proportion_confint(count=287,nobs=483,alpha=(1-.95))
CI_se
```

```
(0.5504107373284738, 0.6379950597729754)
```

```
df_noneutralger=df_noneutral[df_noneutral['country']=='Germany']
```

```
df_noneutralger
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
238	4/4/1909	Germany	Switzerland	1	0	Friendly	Karlsruhe	Germany	False	win
264	5/16/1910	Germany	Belgium	0	3	Friendly	Duisburg	Germany	False	lose
273	10/16/1910	Germany	Netherlands	1	2	Friendly	Kleve	Germany	False	lose
286	3/26/1911	Germany	Switzerland	6	2	Friendly	Stuttgart	Germany	False	win
301	10/9/1911	Germany	Austria	1	2	Friendly	Dresden	Germany	False	lose
...	...	...	...	...	...	...	...	...	...	...
42456	6/19/2021	Germany	Portugal	4	2	UEFA Euro	Munich	Germany	False	win
42474	6/23/2021	Germany	Hungary	2	2	UEFA Euro	Munich	Germany	False	draw
42684	9/5/2021	Germany	Armenia	6	0	FIFA World Cup qualification	Stuttgart	Germany	False	win
42805	10/8/2021	Germany	Romania	2	1	FIFA World Cup qualification	Hamburg	Germany	False	win
42950	11/11/2021	Germany	Liechtenstein	9	0	FIFA World Cup qualification	Wolfsburg	Germany	False	win

433 rows × 10 columns

```
df_noneutralger['result_home'].value_counts()
```

```
df_noneutralger.shape
```

```
(433, 10)
```

```
CI_ger=proportion_confint(count=271,nobs=433,alpha=(1-.95))
CI_ger
```

```
(0.580287721783519, 0.6714443798331091)
```

```
df_noneutralbra=df_noneutral[df_noneutral['country']=='Brazil']
```

```
df_noneutralbra
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
503	5/11/1919	Brazil	Chile	6	0	Copa América	Rio de Janeiro	Brazil	False	win
508	5/18/1919	Brazil	Argentina	3	1	Copa América	Rio de Janeiro	Brazil	False	win
512	5/26/1919	Brazil	Uruguay	2	2	Copa América	Rio de Janeiro	Brazil	False	draw
513	5/29/1919	Brazil	Uruguay	1	0	Copa América	Rio de Janeiro	Brazil	False	win
515	6/1/1919	Brazil	Argentina	3	3	Friendly	Rio de Janeiro	Brazil	False	draw
...	...	...	...	...	...	...	...	...	...	...
42513	7/5/2021	Brazil	Peru	1	0	Copa América	Rio de Janeiro	Brazil	False	win
42529	7/10/2021	Brazil	Argentina	0	1	Copa América	Rio de Janeiro	Brazil	False	lose
42756	9/9/2021	Brazil	Peru	2	0	FIFA World Cup qualification	Recife	Brazil	False	win
42916	10/14/2021	Brazil	Uruguay	4	1	FIFA World Cup qualification	Manaus	Brazil	False	win
42956	11/11/2021	Brazil	Colombia	1	0	FIFA World Cup qualification	São Paulo	Brazil	False	win

354 rows × 10 columns

```
df_noneutralbra['result_home'].value_counts()
```

```
df_noneutralbra.shape
```

```
(354, 10)
```

```
CI_bra=proportion_confint(count=260,nobs=354,alpha=(1-.95))
CI_bra
```

```
(0.6884594670837961, 0.7804670865885203)
```

```
df_noneutralqtr=df_noneutral[df_noneutral['country']=='Qatar']
```

```
df_noneutralqtr
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
8993	2/27/1974	Qatar	Morocco	0	1	Friendly	Doha	Qatar	False	lose
9744	3/26/1976	Qatar	Saudi Arabia	1	0	Gulf Cup	Doha	Qatar	False	win
9751	3/28/1976	Qatar	Kuwait	0	4	Gulf Cup	Doha	Qatar	False	lose
9757	4/1/1976	Qatar	United Arab Emirates	3	1	Gulf Cup	Doha	Qatar	False	win
9761	4/3/1976	Qatar	Iraq	0	0	Gulf Cup	Doha	Qatar	False	draw
...	...	...	...	...	...	...	...	...	...	...
43076	12/3/2021	Qatar	Oman	2	1	Arab Cup	Al Rayyan	Qatar	False	win
43085	12/6/2021	Qatar	Iraq	3	0	Arab Cup	Al Khor	Qatar	False	win
43095	12/10/2021	Qatar	United Arab Emirates	5	0	Arab Cup	Al Khor	Qatar	False	win
43100	12/15/2021	Qatar	Algeria	1	2	Arab Cup	Doha	Qatar	False	lose
43101	12/18/2021	Qatar	Egypt	0	0	Arab Cup	Doha	Qatar	False	draw

259 rows × 10 columns

```
df_noneutralqtr['result_home'].value_counts()
```

```
df_noneutralqtr.shape
```

```
(259, 10)
```

```
CI_qtr=proportion_confint(count=135,nobs=259,alpha=(1-.95))
CI_qtr
```

```
(0.46039734708330093, 0.5820736953877415)
```

```
df_noneutralsp=df_noneutral[df_noneutral['country']=='Spain']
```

```
df_noneutralsp
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
347	12/1/1912	Catalonia	France	1	0	Friendly	Barcelona	Spain	False	win
415	1/3/1915	Basque Country	Catalonia	6	1	Friendly	Bilbao	Spain	False	win
417	2/7/1915	Catalonia	Basque Country	2	2	Friendly	Barcelona	Spain	False	draw
419	5/13/1915	Basque Country	Catalonia	1	0	Friendly	Madrid	Spain	False	win
432	5/21/1916	Catalonia	Basque Country	1	3	Friendly	Barcelona	Spain	False	lose
...	...	...	...	...	...	...	...	...	...	...
42416	6/14/2021	Spain	Sweden	0	0	UEFA Euro	Seville	Spain	False	draw
42454	6/19/2021	Spain	Poland	1	1	UEFA Euro	Seville	Spain	False	draw
42472	6/23/2021	Spain	Slovakia	5	0	UEFA Euro	Seville	Spain	False	win
42675	9/5/2021	Spain	Georgia	4	0	FIFA World Cup qualification	Badajoz	Spain	False	win
43004	11/14/2021	Spain	Sweden	1	0	FIFA World Cup qualification	Seville	Spain	False	win

414 rows × 10 columns

```
df_noneutralsp['result_home'].value_counts()
```

```
df_noneutralsp.shape
```

```
(414, 10)
```

```
CI_sp=proportion_confint(count=276,nobs=414,alpha=(1-.95))
CI_sp
```

```
(0.6212576953545179, 0.7120756379788153)
```

```
df_noneutraluae=df_noneutral[df_noneutral['country']=='United Arab Emirates']
```

```
df_noneutraluae
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
11150	10/21/1979	United Arab Emirates	India	0	0	Friendly	Abu Dhabi	United Arab Emirates	False	draw
11465	8/5/1980	United Arab Emirates	Iran	0	2	Friendly	Al Ain	United Arab Emirates	False	lose
11466	8/7/1980	United Arab Emirates	Iran	0	3	Friendly	Abu Dhabi	United Arab Emirates	False	lose
12119	10/26/1981	United Arab Emirates	India	2	0	Friendly	Abu Dhabi	United Arab Emirates	False	win
12125	10/28/1981	United Arab Emirates	India	3	1	Friendly	Sharjah	United Arab Emirates	False	win
...	...	...	...	...	...	...	...	...	...	...
42395	6/11/2021	United Arab Emirates	Indonesia	5	0	FIFA World Cup qualification	Dubai	United Arab Emirates	False	win
42431	6/15/2021	United Arab Emirates	Vietnam	3	2	FIFA World Cup qualification	Dubai	United Arab Emirates	False	win
42618	9/2/2021	United Arab Emirates	Lebanon	0	0	FIFA World Cup qualification	Dubai	United Arab Emirates	False	draw
42771	10/7/2021	United Arab Emirates	Iran	0	1	FIFA World Cup qualification	Dubai	United Arab Emirates	False	lose
42878	10/12/2021	United Arab Emirates	Iraq	2	2	FIFA World Cup qualification	Dubai	United Arab Emirates	False	draw

232 rows × 10 columns

```
df_noneutraluae['result_home'].value_counts()
```

```
df_noneutraluae.shape
```

```
(232, 10)
```

```
CI_uae=proportion_confint(count=120,nobs=232,alpha=(1-.95))
CI_uae
```

```
(0.45294067232046065, 0.5815420863002291)
```

```
df_noneutralsa=df_noneutral[df_noneutral['country']=='South Africa']
```

```
df_noneutralsa
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
3131	6/24/1950	South Africa	Australia	3	2	Friendly	Durban	South Africa	False	win
3143	7/1/1950	South Africa	Australia	2	1	Friendly	Johannesburg	South Africa	False	win
3149	7/8/1950	South Africa	Australia	1	2	Friendly	Port Elizabeth	South Africa	False	lose
3159	7/23/1950	South Africa	Australia	0	2	Friendly	Cape Town	South Africa	False	lose
3635	5/1/1954	South Africa	Israel	2	1	Friendly	Johannesburg	South Africa	False	win
...	...	...	...	...	...	...	...	...	...	...
42556	7/16/2021	South Africa	Mozambique	3	0	COSAFA Cup	Port Elizabeth	South Africa	False	win
42562	7/18/2021	South Africa	Senegal	0	0	COSAFA Cup	Port Elizabeth	South Africa	False	draw
42698	9/6/2021	South Africa	Ghana	1	0	FIFA World Cup qualification	Johannesburg	South Africa	False	win
42898	10/12/2021	South Africa	Ethiopia	1	0	FIFA World Cup qualification	Johannesburg	South Africa	False	win
42932	11/11/2021	South Africa	Zimbabwe	1	0	FIFA World Cup qualification	Johannesburg	South Africa	False	win

207 rows × 10 columns

```
df_noneutralsa['result_home'].value_counts()
```

```
df_noneutralsa.shape
```

```
(207, 10)
```

```
CI_sa=proportion_confint(count=111,nobs=207,alpha=(1-.95))
CI_sa
```

```
(0.46829749427207346, 0.6041662738438687)
```

```
CI_country = {}
CI_country['result_home'] = ['Egypt','Greece','Portugal','United States','France','Malaysia','England','Sweden','Germany','Brazil','Qa
CI_country['lb'] = [CI_egy[0],CI_gre[0],CI_por[0],CI_usa[0],CI_fr[0],CI_my[0],CI_eng[0],CI_se[0],CI_ger[0],CI_bra[0],CI_qtr[0],CI_sp[0
CI_country['ub'] = [CI_egy[1],CI_gre[1],CI_por[1],CI_usa[1],CI_fr[1],CI_my[1],CI_eng[1],CI_se[1],CI_ger[1],CI_bra[1],CI_qtr[1],CI_sp[1
df_CI= pd.DataFrame(CI_country)
df_CI
```

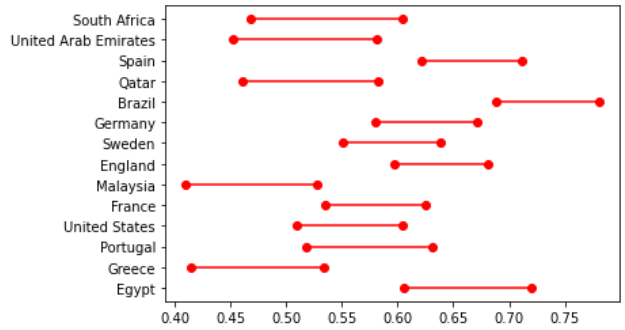
	result_home	lb	ub
0	Egypt	0.605104	0.720477
1	Greece	0.414514	0.533634
2	Portugal	0.517997	0.630652
3	United States	0.508992	0.604809
4	France	0.535082	0.625091
5	Malaysia	0.410109	0.528073
6	England	0.597145	0.681412
7	Sweden	0.550411	0.637995
8	Germany	0.580288	0.671444
9	Brazil	0.688459	0.780467
10	Qatar	0.460397	0.582074
11	Spain	0.621258	0.712076
12	United Arab Emirates	0.452941	0.581542
13	South Africa	0.468297	0.604166

Statistical Comparison Between the Winning Countries in Football based on home matches

In this part I will analyze the following grap. This graph shows the probability that the following 14 countries could win a match at home. I believe that the probability is high in this graph because usually a football team will have a higher probability to win if he is playing in stadium infront of their fans and at their home.

```
import matplotlib.pyplot as plt
for lb,ub,y in zip(df_CI['lb'],df_CI['ub'],range(len(df_CI))):
    plt.plot((lb,ub),(y,y),'ro-')
plt.yticks(range(len(df_CI)),list(df_CI['result_home']))
```

```
([<matplotlib.axis.YTick at 0x7f7a06c31370>,
  <matplotlib.axis.YTick at 0x7f7a06eebbb0>,
  <matplotlib.axis.YTick at 0x7f7a06ed0f40>,
  <matplotlib.axis.YTick at 0x7f7a06c10520>,
  <matplotlib.axis.YTick at 0x7f7a06c10c70>,
  <matplotlib.axis.YTick at 0x7f7a06c19400>,
  <matplotlib.axis.YTick at 0x7f7a06c10d60>,
  <matplotlib.axis.YTick at 0x7f7a06c19c70>,
  <matplotlib.axis.YTick at 0x7f7a06c20130>,
  <matplotlib.axis.YTick at 0x7f7a06c20880>,
  <matplotlib.axis.YTick at 0x7f7a06c210a0>,
  <matplotlib.axis.YTick at 0x7f7a06c21760>,
  <matplotlib.axis.YTick at 0x7f7a06bab040>,
  <matplotlib.axis.YTick at 0x7f7a06c21b50>],
 [Text(0, 0, 'Egypt'),
  Text(0, 1, 'Greece'),
  Text(0, 2, 'Portugal'),
  Text(0, 3, 'United States'),
  Text(0, 4, 'France'),
  Text(0, 5, 'Malaysia'),
  Text(0, 6, 'England'),
  Text(0, 7, 'Sweden'),
  Text(0, 8, 'Germany'),
  Text(0, 9, 'Brazil'),
  Text(0, 10, 'Qatar'),
  Text(0, 11, 'Spain'),
  Text(0, 12, 'United Arab Emirates'),
  Text(0, 13, 'South Africa')])
```



below are some codes of home results and losing

```
CI_egy=proportion_confint(count=43,nobs=258,alpha=(1-.95))
CI_egy
```

```
(0.12119174183927744, 0.21214159149405587)
```

```
CI_gre=proportion_confint(count=81,nobs=270,alpha=(1-.95))
CI_gre
```

```
(0.24533921602296285, 0.35466078397703715)
```

```
CI_por=proportion_confint(count=52,nobs=296,alpha=(1-.95))
CI_por
```

```
(0.1323238781720271, 0.21902747317932428)
```

```
CI_usa=proportion_confint(count=92,nobs=413,alpha=(1-.95))
CI_usa
```

```
(0.1826302643443685, 0.262890316769433)
```

```
CI_fr=proportion_confint(count=99,nobs=462,alpha=(1-.95))
CI_fr
```

```
(0.17686982795630007, 0.2517016006151285)
```

```
CI_my=proportion_confint(count=77,nobs=275,alpha=(1-.95))
CI_my
```

```
(0.22693269802318997, 0.33306730197681006)
```

```
CI_eng=proportion_confint(count=75,nobs=499,alpha=(1-.95))
CI_eng
```

```
(0.11894531042786341, 0.18165589197694623)
```

```
CI_se=proportion_confint(count=99,nobs=483,alpha=(1-.95))
CI_se
```

```
(0.1689682723602105, 0.24096961583854726)
```

```
CI_ger=proportion_confint(count=73,nobs=433,alpha=(1-.95))
CI_ger
```

```
(0.13332743886070297, 0.20385500917624855)
```

```
CI_sp=proportion_confint(count=57,nobs=414,alpha=(1-.95))
CI_sp
```

```
(0.10449025281780361, 0.1708720660227761)
```

```
CI_uae=proportion_confint(count=54,nobs=232,alpha=(1-.95))
CI_uae
```

```
(0.1783806501924808, 0.2871365911868295)
```

```
CI_sa=proportion_confint(count=39,nobs=207,alpha=(1-.95))
CI_sa
```

```
(0.13513618710244651, 0.24167540710045204)
```

```
CI_country = {}
CI_country['result_home'] = ['Egypt','Greece','Portugal','United States','France','Malaysia','England','Sweden','Germany','Brazil','Qa
CI_country['lb'] = [CI_egy[0],CI_gre[0],CI_por[0],CI_usa[0],CI_fr[0],CI_my[0],CI_eng[0],CI_se[0],CI_ger[0],CI_bra[0],CI_qtr[0],CI_sp[0
CI_country['ub'] = [CI_egy[1],CI_gre[1],CI_por[1],CI_usa[1],CI_fr[1],CI_my[1],CI_eng[1],CI_se[1],CI_ger[1],CI_bra[1],CI_qtr[1],CI_sp[1
df_CI= pd.DataFrame(CI_country)
df_CI
```

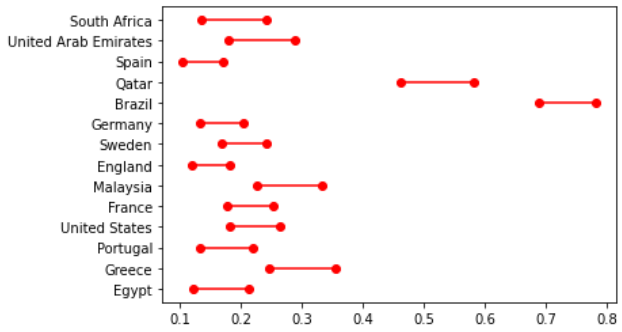
	result_home	lb	ub
0	Egypt	0.121192	0.212142
1	Greece	0.245339	0.354661
2	Portugal	0.132324	0.219027
3	United States	0.182630	0.262890
4	France	0.176870	0.251702
5	Malaysia	0.226933	0.333067
6	England	0.118945	0.181656
7	Sweden	0.168968	0.240970
8	Germany	0.133327	0.203855
9	Brazil	0.688459	0.780467
10	Qatar	0.460397	0.582074
11	Spain	0.104490	0.170872
12	United Arab Emirates	0.178381	0.287137
13	South Africa	0.135136	0.241675

Statistical Comparison Between the losing Countries in Football based on home matches

In this part I will analyze the following grap. This graph shows the probability that the following 14 countries could lose a match at home. I believe that the probability is low for most of the countries in this graph because usually a football team will have a higher probability to win if he is playing in stadium infront of their fans and at their home.

```
import matplotlib.pyplot as plt
for lb,ub,y in zip(df_CI['lb'],df_CI['ub'],range(len(df_CI))):
    plt.plot((lb,ub),(y,y),'ro-')
plt.yticks(range(len(df_CI)),list(df_CI['result_home']))
```

```
([<matplotlib.axis.YTick at 0x7f7a04b04250>,
  <matplotlib.axis.YTick at 0x7f7a04af5a90>,
  <matplotlib.axis.YTick at 0x7f7a04afc9d0>,
  <matplotlib.axis.YTick at 0x7f7a04aca430>,
  <matplotlib.axis.YTick at 0x7f7a04acab80>,
  <matplotlib.axis.YTick at 0x7f7a04ad1310>,
  <matplotlib.axis.YTick at 0x7f7a04acae50>,
  <matplotlib.axis.YTick at 0x7f7a04ad1940>,
  <matplotlib.axis.YTick at 0x7f7a04ad7040>,
  <matplotlib.axis.YTick at 0x7f7a04ad7790>,
  <matplotlib.axis.YTick at 0x7f7a04adf040>,
  <matplotlib.axis.YTick at 0x7f7a04adf670>,
  <matplotlib.axis.YTick at 0x7f7a04adfdc0>,
  <matplotlib.axis.YTick at 0x7f7a04adfa60>],
 [Text(0, 0, 'Egypt'),
  Text(0, 1, 'Greece'),
  Text(0, 2, 'Portugal'),
  Text(0, 3, 'United States'),
  Text(0, 4, 'France'),
  Text(0, 5, 'Malaysia'),
  Text(0, 6, 'England'),
  Text(0, 7, 'Sweden'),
  Text(0, 8, 'Germany'),
  Text(0, 9, 'Brazil'),
  Text(0, 10, 'Qatar'),
  Text(0, 11, 'Spain'),
  Text(0, 12, 'United Arab Emirates'),
  Text(0, 13, 'South Africa')])
```




```
import numpy as np
```

```
import statsmodels.api as sm
from statsmodels.stats.proportion import proportion_confint
```

```
dfbra=df[df['country']=='Brazil']
```

```
dfbra.head()
```

	date	home_team	away_team	home_score	away_score	tournament	city	country	neutral	result_home
503	5/11/1919	Brazil	Chile	6	0	Copa América	Rio de Janeiro	Brazil	False	win
505	5/13/1919	Argentina	Uruguay	2	3	Copa América	Rio de Janeiro	Brazil	True	lose
507	5/17/1919	Chile	Uruguay	0	2	Copa América	Rio de Janeiro	Brazil	True	lose
508	5/18/1919	Brazil	Argentina	3	1	Copa América	Rio de Janeiro	Brazil	False	win
510	5/22/1919	Argentina	Chile	4	1	Copa América	Rio de Janeiro	Brazil	True	win

```
conditions = [
    (dfbra['tournament']=='Friendly'),
    (dfbra['tournament']=='Copa América'),
    (dfbra['tournament']=='FIFA World Cup qualification')
    ]
```

```
values=['Friendly','Copa América','FIFA World Cup qualification']
```

```
dfbra['typematch'] = np.select(conditions, values)
```

```
<ipython-input-119-8c959ec913db>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
  dfbra['typematch'] = np.select(conditions, values)
```

```
dfbra['typematch'].value_counts()
```

```
x=pd.crosstab(dfbra['typematch'],dfbra['result_home'],margins=True)
x
```

result_home	draw	lose	win	All
typematch				
0	32	62	82	176
Copa América	36	41	60	137
FIFA World Cup qualification	12	0	48	60
Friendly	38	10	146	194
All	118	113	336	567

```
CI_brawin_friendly=proportion_confint(count=146,nobs=194,alpha=(1-.95))
CI_brawin_friendly
```

```
(0.6918558124106731, 0.8132988267645846)
```

```
CI_brawin_CopaAmérica=proportion_confint(count=60,nobs=137,alpha=(1-.95))
CI_brawin_CopaAmérica
```

```
(0.354877784454012, 0.521034624305112)
```

```
CI_brawin_Worldcup=proportion_confint(count=48,nobs=60,alpha=(1-.95))
CI_brawin_Worldcup
```

```
(0.6987878950494673, 0.9012121049505328)
```

```
ci_brawin = {}
ci_brawin['Typematch'] = ['Friendly','Copa América','FIFA World Cup qualification']
ci_brawin['lb'] = [CI_brawin_friendly[0],CI_brawin_CopaAmérica[0],CI_brawin_Worldcup[0]]
ci_brawin['ub'] = [CI_brawin_friendly[1],CI_brawin_CopaAmérica[1],CI_brawin_Worldcup[1]]
df_ci= pd.DataFrame(ci_brawin)
df_ci
```

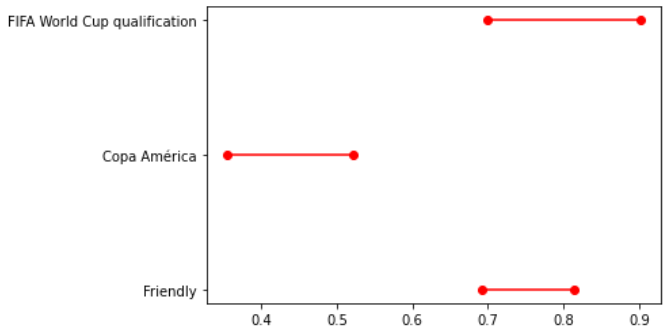
	Typematch	lb	ub
0	Friendly	0.691856	0.813299
1	Copa América	0.354878	0.521035
2	FIFA World Cup qualification	0.698788	0.901212

Statistical Analysis of Football Matches Over the World and Comparing the Winning Countries with different competitions

In this part I will analyze the following graph. This graph shows the probability that Brazil could win a match in different types of tournaments such as the following: Friendly matches, Copa América and FIFA World Cup qualification. I think the probability of winning Friendly matches is high because it is brazil with the best players in the world. As you can see their world cup qualifications wins are also very high since they are the only country that qualified to all world cups and the country with most world cup wins with 5 championships including 3 consecitive ones.

```
import matplotlib.pyplot as plt
for lb,ub,y in zip(df_ci['lb'],df_ci['ub'],range(len(df_ci))):
    plt.plot((lb,ub),(y,y),'ro-')
plt.yticks(range(len(df_ci)),list(df_ci['Typematch']))
```

```
([<matplotlib.axis.YTick at 0x7f7a049f8550>,
  <matplotlib.axis.YTick at 0x7f7a049eed90>,
  <matplotlib.axis.YTick at 0x7f7a04affd00>],
 [Text(0, 0, 'Friendly'),
  Text(0, 1, 'Copa América'),
  Text(0, 2, 'FIFA World Cup qualification')])
```



Part II

```
import numpy as np
```

```
df=pd.read_csv('covid_data.csv',encoding='latin-1')
```

```
df
```

	date	iso3c	country	income	region	continent	dcases	ddeaths	population	weekdays	month
0	2020-02-24	AFG	Afghanistan	Low income	South Asia	Asia	5	0	38041754	Mon	Feb
1	2020-02-25	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Tue	Feb
2	2020-02-26	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Wed	Feb
3	2020-02-27	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Thu	Feb
4	2020-02-28	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Fri	Feb
...	...	...	...	...	...	...	...	...	...	...	...
122838	2021-12-27	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	1098	17	14645468	Mon	Dec
122839	2021-12-28	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	2099	32	14645468	Tue	Dec
122840	2021-12-29	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	0	0	14645468	Wed	Dec
122841	2021-12-30	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	4180	57	14645468	Thu	Dec
122842	2021-12-31	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	1530	7	14645468	Fri	Dec

122843 rows × 11 columns

```
df.head()
```

	date	iso3c	country	income	region	continent	dcases	ddeaths	population	weekdays	month
0	2020-02-24	AFG	Afghanistan	Low income	South Asia	Asia	5	0	38041754	Mon	Feb
1	2020-02-25	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Tue	Feb
2	2020-02-26	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Wed	Feb
3	2020-02-27	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Thu	Feb
4	2020-02-28	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Fri	Feb

```
df['region'].unique()
```

```
df['income'].unique()
```

```
df['continent'].unique()
```

```
from pandas.api.types import CategoricalDtype
cats=['Sat', 'Sun', 'Mon','Tue','Wed','Thu','Fri']
cat_type = CategoricalDtype(categories=cats, ordered=True)
df['weekdays'] = df['weekdays'].astype(cat_type)
```

```
from pandas.api.types import CategoricalDtype
cats=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']
cat_type= CategoricalDtype(categories=cats, ordered=True)
df['month']=df['month'].astype(cat_type)
```

```
dfegy=df[df['country']=='Egypt']
```

```
stats=dfegy.groupby("weekdays").agg({"dcases": [np.mean, np.std, np.size]})
```

```
stats=pd.DataFrame(stats)
```

```
stats
```

	dcases		
	mean	std	size
weekdays			
Sat	558.806122	421.803605	98
Sun	545.520408	422.358748	98
Mon	561.846939	442.137949	98
Tue	566.153061	419.125460	98
Wed	561.479592	406.337812	98
Thu	567.683673	410.020004	98
Fri	567.161616	428.533849	99

```
import scipy.stats
```

```
from scipy.stats import t
```

```
stats.index
```

```
CategoricalIndex(['Sat', 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri'], categories=['Sat', 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri'], or
```

```
stats.shape
```

```
(7, 3)
```

```
stats.columns=['mean','std','size']
```

```
stats.columns
```

```
Index(['mean', 'std', 'size'], dtype='object')
```

```
x=dfegy['dcases']
```

```
def get_ci_lb(x, alpha=0.05):
    sample_s=np.std(x)
    sample_mean=np.mean(x)
    sample_size=len(x)
    margin_of_error = t.ppf(1 - alpha/2,sample_size-1)*sample_s/np.sqrt(sample_size-1)
    return sample_mean - margin_of_error
```

```
get_ci_lb(x)
```

```
529.7945911276134
```

```
def get_ci_ub(x, alpha=0.05):
    sample_s=np.std(x)
    sample_mean=np.mean(x)
    sample_size=len(x)
    margin_of_error = t.ppf(1 - alpha/2,sample_size-1)*sample_s/np.sqrt(sample_size-1)
    return sample_mean + margin_of_error
```

```
get_ci_ub(x)
```

```
592.6944918418188
```

```
ci_dcases=statsw=dfegy.groupby("weekdays").agg({"dcases": [np.mean, np.std, np.size,get_ci_lb,get_ci_ub]})
statsw
```

	dcases				
	mean	std	size	get_ci_lb	get_ci_ub
weekdays					
Sat	558.806122	421.803605	98	474.239849	643.372396
Sun	545.520408	422.358748	98	460.842836	630.197981
Mon	561.846939	442.137949	98	473.203887	650.489990
Tue	566.153061	419.125460	98	482.123722	650.182401
Wed	561.479592	406.337812	98	480.014014	642.945170
Thu	567.683673	410.020004	98	485.479863	649.887484
Fri	567.161616	428.533849	99	481.692047	652.631185

```
statsw.columns=['mean','std','size','lb','ub']
```

```
statsw['month']=statsw.index
```

```
statsw
```

	mean	std	size	lb	ub	month
weekdays						
Sat	558.806122	421.803605	98	474.239849	643.372396	Sat
Sun	545.520408	422.358748	98	460.842836	630.197981	Sun
Mon	561.846939	442.137949	98	473.203887	650.489990	Mon
Tue	566.153061	419.125460	98	482.123722	650.182401	Tue
Wed	561.479592	406.337812	98	480.014014	642.945170	Wed
Thu	567.683673	410.020004	98	485.479863	649.887484	Thu
Fri	567.161616	428.533849	99	481.692047	652.631185	Fri

```
df['totcases'] = df.groupby(['country'])['dcases'].cumsum()
```

```
df.loc[df['country']=='China'].head(6)
```

	date	iso3c	country	income	region	continent	dcases	ddeaths	population	weekdays	month	year	totcases
21405	2020-01-22	CHN	China	Upper middle income	East Asia & Pacific	Asia	548	17	1397715000	Wed	Jan	2020	548
21406	2020-01-23	CHN	China	Upper middle income	East Asia & Pacific	Asia	95	1	1397715000	Thu	Jan	2020	643
21407	2020-01-24	CHN	China	Upper middle income	East Asia & Pacific	Asia	277	8	1397715000	Fri	Jan	2020	920
21408	2020-01-25	CHN	China	Upper middle income	East Asia & Pacific	Asia	486	16	1397715000	Sat	Jan	2020	1406
21409	2020-01-26	CHN	China	Upper middle income	East Asia & Pacific	Asia	669	14	1397715000	Sun	Jan	2020	2075
21410	2020-01-27	CHN	China	Upper middle income	East Asia & Pacific	Asia	802	26	1397715000	Mon	Jan	2020	2877

```
df['totdeaths'] = df.groupby(['country'])['ddeaths'].cumsum()
```

```
df['totdeaths'].loc[df['country']=='China']
```

```
df['cfr']=df['totdeaths']/df['totcases']
```

```
import matplotlib.pyplot as plt
```

```
df_chi=df.loc[df['country']=='China']
```

```
df_chi['date'] = pd.to_datetime(df_chi['date'],format='%Y-%m-%d')
```

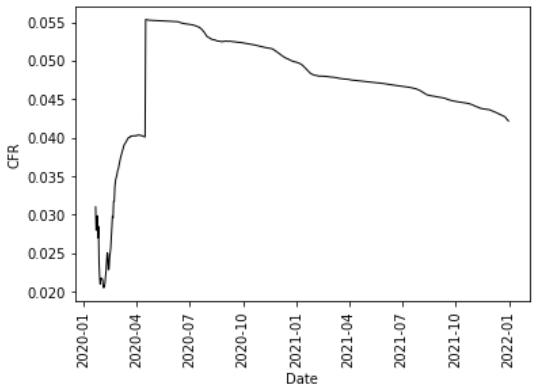
```
<ipython-input-264-f09aa949778e>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-
  df_chi['date'] = pd.to_datetime(df_chi['date'],format='%Y-%m-%d')
```

Case Fatality Rate of covid-19

This is a graph that shows the difference between deaths and confirmed cases of Covid-19. This ratio is called case fatality rate (CFR). The CFR is made on the country China, I chose China because it is the country were Covid-19 started. As you can see the cases/deaths started from the buttom but later on they started increasing faster and faster with a higher rate and we can conclude that the ratio of deaths and cases started to decrease gradually, yet it is still high.

```
plt.plot( 'date', 'cfr', data=df_chi,  color='black', markersize=4,     linewidth=1)
plt.xlabel("Date")
plt.ylabel("CFR")
plt.xticks(rotation=90)
plt.show()
```




```
import matplotlib.pyplot as plt
```

```
df['date'][0]
```

```
'2020-02-24'
```

```
df['date'] = pd. to_datetime(df['date'],format='%Y-%m-%d')
```

```
df['date'][0]
```

```
Timestamp('2020-02-24 00:00:00')
```

```
df['year'] = pd. DatetimeIndex(df['date']). year
```

```
df['year'][0]
```

```
2020
```

```
statsdcases=df.groupby(['continent','year','month']).agg({"dcases": [np.mean, np.std, np.size,get_ci_lb,get_ci_ub]})
```

```
statsdcases
```

			dcases				
			mean	std	size	get_ci_lb	get_ci_ub
continent	year	month					
Africa	2020	Jan	NaN	NaN	NaN	NaN	NaN
		Feb	0.130435	0.344350	23.0	-0.018473	0.279343
		Mar	7.053988	19.438419	815.0	5.717465	8.390511
		Apr	21.675871	47.034430	1521.0	19.310249	24.041493
		May	66.293072	174.256499	1631.0	57.829909	74.756235
...	...	...	...	...	...	...	...
South America(continent)	2021	Aug	3711.341398	8223.952633	372.0	2872.893048	4549.789748
		Sep	2457.269444	8382.716252	360.0	1588.412630	3326.126259
		Oct	1618.491935	3515.728407	372.0	1260.056422	1976.927449
		Nov	1576.597222	2788.302480	360.0	1287.593570	1865.600874
		Dec	2217.798387	5128.601112	372.0	1694.927290	2740.669485

144 rows × 5 columns

```
statsdcases=statsdcases.reset_index()
statsdcases
```

	continent	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
0	Africa	2020	Jan	NaN	NaN	NaN	NaN	NaN
1	Africa	2020	Feb	0.130435	0.344350	23.0	-0.018473	0.279343
2	Africa	2020	Mar	7.053988	19.438419	815.0	5.717465	8.390511
3	Africa	2020	Apr	21.675871	47.034430	1521.0	19.310249	24.041493
4	Africa	2020	May	66.293072	174.256499	1631.0	57.829909	74.756235
...	...	...	...	...	...	...	...	...
139	South America(continent)	2021	Aug	3711.341398	8223.952633	372.0	2872.893048	4549.789748
140	South America(continent)	2021	Sep	2457.269444	8382.716252	360.0	1588.412630	3326.126259
141	South America(continent)	2021	Oct	1618.491935	3515.728407	372.0	1260.056422	1976.927449
142	South America(continent)	2021	Nov	1576.597222	2788.302480	360.0	1287.593570	1865.600874
143	South America(continent)	2021	Dec	2217.798387	5128.601112	372.0	1694.927290	2740.669485

144 rows × 8 columns

```
statsAs20=statsdcases[(statsdcases['continent']=='Asia') & (statsdcases['year']==2020)]
```

```
statsAs20
```

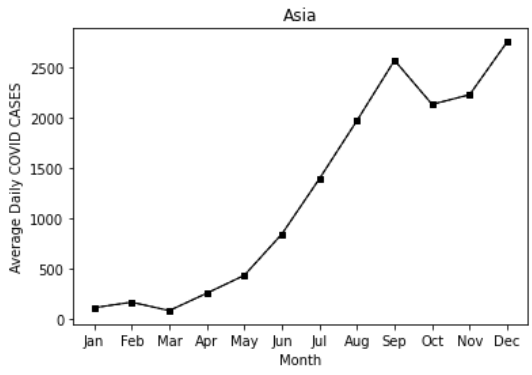
	continent	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
24	Asia	2020	Jan	111.000000	410.875728	89.0	24.448081	197.551919
25	Asia	2020	Feb	165.493274	942.082908	446.0	77.822955	253.163592
26	Asia	2020	Mar	81.610111	303.900086	1167.0	64.156129	99.064094
27	Asia	2020	Apr	255.156600	629.879960	1341.0	221.413580	288.899619
28	Asia	2020	May	433.419355	937.308182	1426.0	384.729356	482.109354
29	Asia	2020	Jun	845.336232	2187.082379	1380.0	729.843398	960.829065
30	Asia	2020	Jul	1396.580645	5509.114767	1426.0	1110.400711	1682.760580
31	Asia	2020	Aug	1976.046283	9438.275242	1426.0	1485.759712	2466.332855
32	Asia	2020	Sep	2576.080435	12741.756197	1380.0	1903.228995	3248.931874
33	Asia	2020	Oct	2137.612202	8947.155675	1426.0	1672.837636	2602.386768
34	Asia	2020	Nov	2233.760145	6712.137800	1380.0	1879.313591	2588.206699
35	Asia	2020	Dec	2760.654278	22415.623487	1426.0	1596.238227	3925.070329

```
statsAs20.columns=['country','year','month','mean','std','size','lb','ub']
```

Analysis of CoronaVirus Pandemic Over the Period Between 2020 and 2021 including several graphs (regions, income, and continents)

As you can see this graph is the average daily cases of covid-19 the continent asia throughout the year 2020 starting from january to december and as you can see it is increasing.

```
plt.plot( 'month', 'mean', data=statsAs20, marker='s', color='black', markersize=4,     linewidth=1, linestyle='--')
plt.plot( 'month', 'mean', data=statsAs20, marker='o', color='black',  markersize=4, linewidth=1,linestyle='-')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Asia")
plt.show()
```

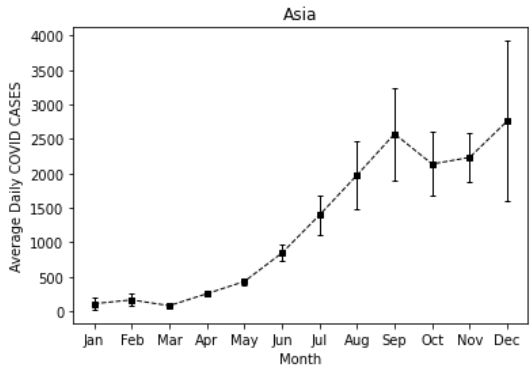



```
ci_lb_ub=[statsAs20['lb'],statsAs20['ub']]
```

```
err = np.abs(ci_lb_ub - statsAs20['mean'].to_numpy())
```

in this below graph you can see the marginal error that could happen through the calculations.

```
plt.errorbar('month', 'mean',  yerr=err, data=statsAs20,marker='s', capsize=2,
             color='black', markersize=4, linewidth=1, linestyle='--')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Asia")
plt.show()
```




```
statsAs21=statsdcases[(statsdcases['continent']=='Asia')
                      & (statsdcases['year']==2021)]
statsAs21
```

	continent	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
36	Asia	2021	Jan	1687.433380	3319.399414	1426.0	1515.001774	1859.864986
37	Asia	2021	Feb	1485.468944	2780.488524	1288.0	1333.477239	1637.460649
38	Asia	2021	Mar	2451.384993	6687.838246	1426.0	2103.974349	2798.795637
39	Asia	2021	Apr	7998.099275	37527.035782	1380.0	6016.416401	9979.782150
40	Asia	2021	May	8233.206872	44061.157010	1426.0	5944.378347	10522.035397
41	Asia	2021	Jun	3299.422464	11770.007270	1380.0	2677.885981	3920.958947
42	Asia	2021	Jul	4400.887097	9153.630152	1426.0	3925.386878	4876.387316
43	Asia	2021	Aug	5685.551192	9482.403695	1426.0	5192.972296	6178.130088
44	Asia	2021	Sep	4201.860870	7508.954554	1380.0	3805.336968	4598.384771
45	Asia	2021	Oct	2593.950912	5312.542472	1426.0	2317.982245	2869.919578
46	Asia	2021	Nov	2035.668116	4450.232884	1380.0	1800.665534	2270.670698
47	Asia	2021	Dec	1795.433380	4308.138282	1426.0	1571.640124	2019.226636

```
statsAs20.columns=['continent','year','month','mean','std','size','lb','ub']
```

```
statsAs21.columns=['continent','year','month','mean','std','size','lb','ub']
```

```
ci_lb_ub20=[statsAs20['lb'],statsAs20['ub']]
err20 = np.abs(ci_lb_ub20 - statsAs20['mean'].to_numpy())
```

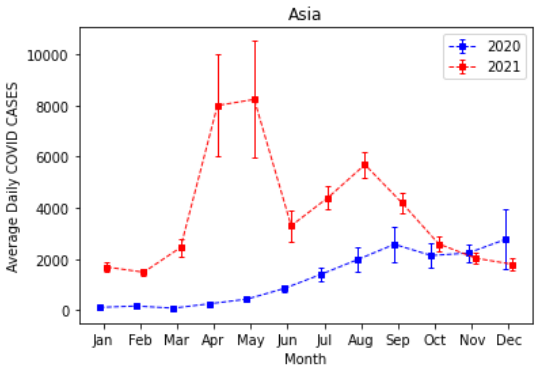
```
ci_lb_ub21=[statsAs21['lb'],statsAs21['ub']]
err21 = np.abs(ci_lb_ub21 - statsAs21['mean'].to_numpy())
```

```
from matplotlib.transforms import Affine2D
```

As you can see in the below graph we included the graph of the year 2021 to the year 2020 to be able to see whether the cases has increased in the following year or not however the cases in asia seems to have increased a lot from 2020 to 2021

```
fig, ax = plt.subplots()

trans1 = Affine2D().translate(-0.1, 0.0) + ax.transData
trans2 = Affine2D().translate(+0.1, 0.0) + ax.transData
plt.errorbar('month', 'mean',  yerr=err20, data=statsAs20,marker='s', capsize=2,
             color='blue', markersize=4, linewidth=1, linestyle='--',transform=trans1)
plt.errorbar('month', 'mean',  yerr=err21, data=statsAs21,marker='s', capsize=2,
             color='red', markersize=4, linewidth=1, linestyle='--',transform=trans2)
plt.legend(['2020','2021'])
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Asia")
plt.show()
```




```
statsEu20=statsdcases[(statsdcases['continent']=='Europe') & (statsdcases['year']==2020)]
```

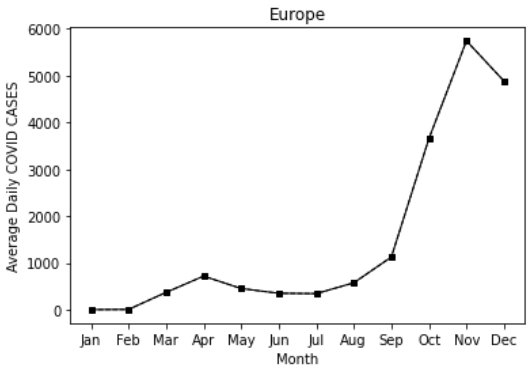
```
statsEu20
```

	continent	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
48	Europe	2020	Jan	0.894737	0.936586	19.0	0.443317	1.346157
49	Europe	2020	Feb	4.879479	25.101789	307.0	2.060416	7.698541
50	Europe	2020	Mar	369.486078	1120.091160	1257.0	307.505941	431.466215
51	Europe	2020	Apr	716.859690	2026.953638	1290.0	606.145054	827.574326
52	Europe	2020	May	451.645911	1513.905248	1333.0	370.301725	532.990098
53	Europe	2020	Jun	350.358915	1227.591513	1290.0	283.306396	417.411434
54	Europe	2020	Jul	342.822206	978.006769	1333.0	290.272573	395.371838
55	Europe	2020	Aug	574.078770	1526.829653	1333.0	492.040137	656.117402
56	Europe	2020	Sep	1124.893023	2728.262523	1290.0	975.872056	1273.913991
57	Europe	2020	Oct	3655.168792	6861.235403	1333.0	3286.505289	4023.832295
58	Europe	2020	Nov	5744.452713	9583.968860	1290.0	5220.964852	6267.940574
59	Europe	2020	Dec	4883.578395	7650.466032	1333.0	4472.508458	5294.648332

```
statsEu20.columns=['continent','year','month','mean','std','size','lb','ub']
```

As you can see this graph is the average daily cases of covid-19 the continent europe throughout the year 2020 starting from january till december and as you can see the number of daily cases increased gradually.

```
plt.plot( 'month', 'mean', data=statsEu20, marker='s', color='black', markersize=4,     linewidth=1, linestyle='--')
plt.plot( 'month', 'mean', data=statsEu20, marker='o', color='black',  markersize=4, linewidth=1,linestyle='-')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Europe")
plt.show()
```

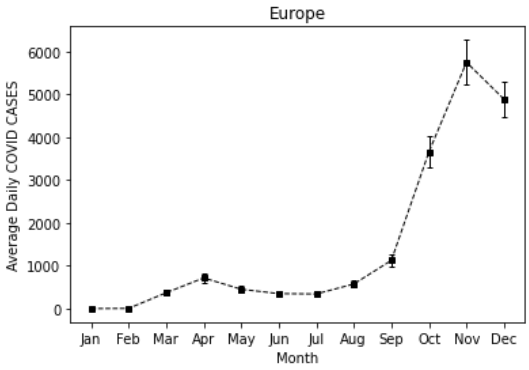



```
ci_lb_ub=[statsEu20['lb'],statsEu20['ub']]
```

```
err = np.abs(ci_lb_ub - statsEu20['mean'].to_numpy())
```

in this below graph you can see the marginal error that could happen through the calculations.

```
plt.errorbar('month', 'mean',  yerr=err, data=statsEu20,marker='s', capsize=2,
             color='black', markersize=4, linewidth=1, linestyle='--')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Europe")
plt.show()
```




```
statsEu21=statsdcases[(statsdcases['continent']=='Europe')
                      & (statsdcases['year']==2021)]
statsEu21
```

	continent	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
60	Europe	2021	Jan	4974.927982	9883.362427	1333.0	4443.881487	5505.974477
61	Europe	2021	Feb	3370.627076	5876.172663	1204.0	3038.375874	3702.878279
62	Europe	2021	Mar	4226.169542	6925.883241	1333.0	3854.032423	4598.306661
63	Europe	2021	Apr	3882.789922	7984.927128	1290.0	3446.643637	4318.936208
64	Europe	2021	May	1778.587397	3377.861190	1333.0	1597.090325	1960.084469
65	Europe	2021	Jun	1061.393798	3027.147988	1290.0	896.047349	1226.740248
66	Europe	2021	Jul	2644.664666	7673.031751	1333.0	2232.382242	3056.947090
67	Europe	2021	Aug	2896.221305	6796.809360	1333.0	2531.019501	3261.423109
68	Europe	2021	Sep	2872.321705	6136.287433	1290.0	2537.150335	3207.493076
69	Europe	2021	Oct	4167.192798	8245.780023	1333.0	3724.135831	4610.249765
70	Europe	2021	Nov	7224.170543	11493.930719	1290.0	6596.358277	7851.982808
71	Europe	2021	Dec	10521.554389	23984.334327	1333.0	9232.843510	11810.265267

```
statsEu20.columns=['continent','year','month','mean','std','size','lb','ub']
```

```
statsEu21.columns=['continent','year','month','mean','std','size','lb','ub']
```

```
ci_lb_ub20=[statsEu20['lb'],statsEu20['ub']]
err20 = np.abs(ci_lb_ub20 - statsEu20['mean'].to_numpy())
```

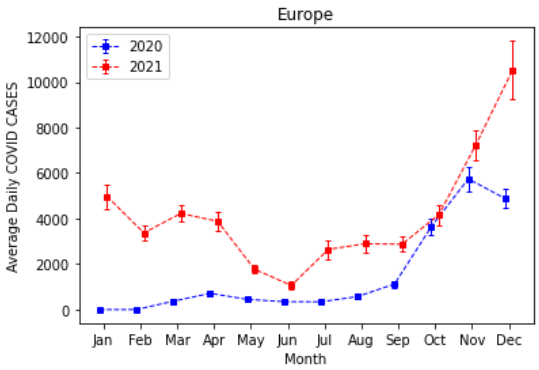
```
ci_lb_ub21=[statsEu21['lb'],statsEu21['ub']]
err21 = np.abs(ci_lb_ub21 - statsEu21['mean'].to_numpy())
```

```
from matplotlib.transforms import Affine2D
```

As you can see in the below graph we included the graph of the year 2021 to the year 2020 to be able to see whether the cases has increased in the following year or not however the cases in europe seems to have increased a lot. Now we will be able to compare not only the different years but we can also compare europe with asia and as you can see they're similar in the number of cases.

```
fig, ax = plt.subplots()

trans1 = Affine2D().translate(-0.1, 0.0) + ax.transData
trans2 = Affine2D().translate(+0.1, 0.0) + ax.transData
plt.errorbar('month', 'mean',  yerr=err20, data=statsEu20,marker='s', capsize=2,
             color='blue', markersize=4, linewidth=1, linestyle='--',transform=trans1)
plt.errorbar('month', 'mean',  yerr=err21, data=statsEu21,marker='s', capsize=2,
             color='red', markersize=4, linewidth=1, linestyle='--',transform=trans2)
plt.legend(['2020','2021'])
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Europe")
plt.show()
```




```
statsdcases=df.groupby(['region','year','month']).agg({"dcases": [np.mean, np.std, np.size,get_ci_lb,get_ci_ub]})
```

```
statsdcases
```

			dcases				
			mean	std	size	get_ci_lb	get_ci_ub
region	year	month					
East Asia & Pacific	2020	Jan	126.679487	436.945076	78.0	28.163595	225.195380
		Feb	250.304795	1155.872616	292.0	117.174576	383.435013
		Mar	55.969828	102.512114	464.0	46.617902	65.321753
		Apr	92.700000	177.828711	570.0	78.070236	107.329764
		May	86.125637	193.352744	589.0	70.478472	101.772801
...	...	...	...	...	...	...	...
Sub-Saharan Africa	2021	Aug	466.880577	1745.719642	1457.0	377.167865	556.593288
		Sep	247.950355	1021.166824	1410.0	194.603545	301.297165
		Oct	87.528483	293.259577	1457.0	72.457843	102.599123
		Nov	89.737589	588.981229	1410.0	58.968601	120.506577
		Dec	682.937543	2624.860565	1457.0	548.045689	817.829397

168 rows × 5 columns

```
statsdcases=statsdcases.reset_index()
statsdcases
```

	region	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
0	East Asia & Pacific	2020	Jan	126.679487	436.945076	78.0	28.163595	225.195380
1	East Asia & Pacific	2020	Feb	250.304795	1155.872616	292.0	117.174576	383.435013
2	East Asia & Pacific	2020	Mar	55.969828	102.512114	464.0	46.617902	65.321753
3	East Asia & Pacific	2020	Apr	92.700000	177.828711	570.0	78.070236	107.329764
4	East Asia & Pacific	2020	May	86.125637	193.352744	589.0	70.478472	101.772801
...	...	...	...	...	...	...	...	...
163	Sub-Saharan Africa	2021	Aug	466.880577	1745.719642	1457.0	377.167865	556.593288
164	Sub-Saharan Africa	2021	Sep	247.950355	1021.166824	1410.0	194.603545	301.297165
165	Sub-Saharan Africa	2021	Oct	87.528483	293.259577	1457.0	72.457843	102.599123
166	Sub-Saharan Africa	2021	Nov	89.737589	588.981229	1410.0	58.968601	120.506577
167	Sub-Saharan Africa	2021	Dec	682.937543	2624.860565	1457.0	548.045689	817.829397

168 rows × 8 columns

```
statsSA20=statsdcases[(statsdcases['region']=='South Asia') & (statsdcases['year']==2020)]
```

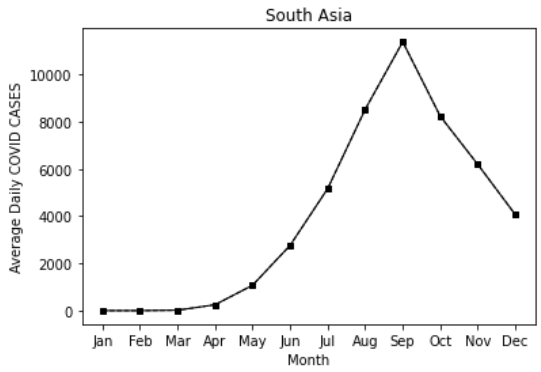
```
statsSA20
```

	region	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
120	South Asia	2020	Jan	0.214286	0.425815	14.0	-0.031573	0.460144
121	South Asia	2020	Feb	0.112245	0.589997	98.0	-0.006042	0.230532
122	South Asia	2020	Mar	16.986900	41.824978	229.0	11.540901	22.432898
123	South Asia	2020	Apr	249.016667	434.304474	240.0	193.790926	304.242407
124	South Asia	2020	May	1075.637097	1777.675480	248.0	853.302048	1297.972146
125	South Asia	2020	Jun	2764.604167	4509.400678	240.0	2191.193097	3338.015236
126	South Asia	2020	Jul	5165.862903	12359.181246	248.0	3620.091923	6711.633884
127	South Asia	2020	Aug	8523.125000	21375.481545	248.0	5849.679372	11196.570628
128	South Asia	2020	Sep	11379.983333	28894.458700	240.0	7705.791650	15054.175017
129	South Asia	2020	Oct	8227.786290	20223.802330	248.0	5698.381944	10757.190636
130	South Asia	2020	Nov	6186.725000	13958.944716	240.0	4411.718964	7961.731036
131	South Asia	2020	Dec	4063.500000	8817.022427	248.0	2960.749168	5166.250832

```
statsSA20.columns=['region','year','month','mean','std','size','lb','ub']
```

As you can see this graph is the average daily cases of covid-19 the region South Asia throughout the year 2020 starting from january till december and as you can see the average daily cases started to increase gradually then started to decline again.

```
plt.plot( 'month', 'mean', data=statsSA20, marker='s', color='black', markersize=4,     linewidth=1, linestyle='--')
plt.plot( 'month', 'mean', data=statsSA20, marker='o', color='black',  markersize=4, linewidth=1,linestyle='-')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("South Asia")
plt.show()
```

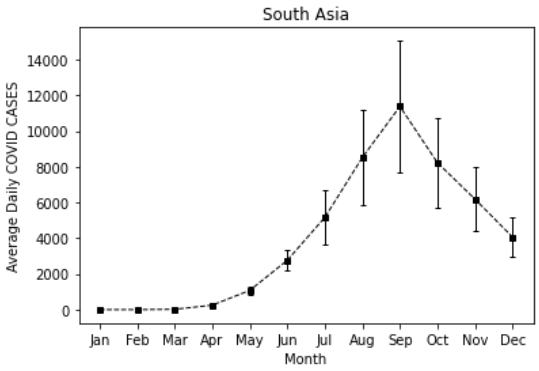



```
ci_lb_ub=[statsSA20['lb'],statsSA20['ub']]
```

```
err = np.abs(ci_lb_ub - statsSA20['mean'].to_numpy())
```

in this below graph you can see the marginal error that could happen through the calculations.

```
plt.errorbar('month', 'mean',  yerr=err, data=statsSA20,marker='s', capsize=2,
             color='black', markersize=4, linewidth=1, linestyle='--')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("South Asia")
plt.show()
```




```
statsSA21=statsdcases[(statsdcases['region']=='South Asia')
                      & (statsdcases['year']==2021)]
statsSA21
```

	region	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
132	South Asia	2021	Jan	2391.008065	5254.481990	248.0	1733.826523	3048.189606
133	South Asia	2021	Feb	1908.111607	4175.464437	224.0	1358.327410	2457.895804
134	South Asia	2021	Mar	5176.931452	13268.004868	248.0	3517.493299	6836.369604
135	South Asia	2021	Apr	30476.229167	84562.122914	240.0	19723.390638	41229.067695
136	South Asia	2021	May	38356.133065	100189.922947	248.0	25825.313073	50886.953056
137	South Asia	2021	Jun	10799.262500	26253.582576	240.0	7460.882094	14137.642906
138	South Asia	2021	Jul	7240.709677	13014.640919	248.0	5612.959922	8868.459433
139	South Asia	2021	Aug	7030.568548	12041.588531	248.0	5524.519099	8536.617998
140	South Asia	2021	Sep	5038.125000	10563.826738	240.0	3694.838914	6381.411086
141	South Asia	2021	Oct	2439.060484	5546.820540	248.0	1745.315967	3132.805001
142	South Asia	2021	Nov	1525.762500	3404.577307	240.0	1092.839712	1958.685288
143	South Asia	2021	Dec	1290.012097	3164.751858	248.0	894.194487	1685.829707

```
statsSA20.columns=['region','year','month','mean','std','size','lb','ub']
```

```
statsSA21.columns=['region','year','month','mean','std','size','lb','ub']
```

```
ci_lb_ub20=[statsSA20['lb'],statsSA20['ub']]
err20 = np.abs(ci_lb_ub20 - statsSA20['mean'].to_numpy())
```

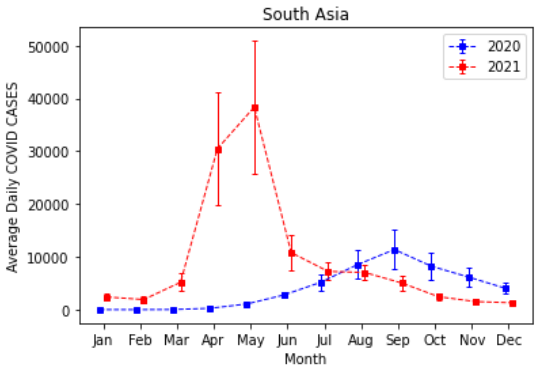
```
ci_lb_ub21=[statsSA21['lb'],statsSA21['ub']]
err21 = np.abs(ci_lb_ub21 - statsSA21['mean'].to_numpy())
```

```
from matplotlib.transforms import Affine2D
```

As you can see in the below graph we included the graph of the year 2021 to the year 2020 to be able to see whether the cases has increased in the following year or not however the cases in south asia seems to have increased a lot, however in the end of the year they started to decrease and 2020 had more cases than 2021.

```
fig, ax = plt.subplots()

trans1 = Affine2D().translate(-0.1, 0.0) + ax.transData
trans2 = Affine2D().translate(+0.1, 0.0) + ax.transData
plt.errorbar('month', 'mean',  yerr=err20, data=statsSA20,marker='s', capsize=2,
             color='blue', markersize=4, linewidth=1, linestyle='--',transform=trans1)
plt.errorbar('month', 'mean',  yerr=err21, data=statsSA21,marker='s', capsize=2,
             color='red', markersize=4, linewidth=1, linestyle='--',transform=trans2)
plt.legend(['2020','2021'])
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("South Asia")
plt.show()
```




```
statsME20=statsdcases[(statsdcases['region']=='Middle East & North Africa') & (statsdcases['year']==2020)]
```

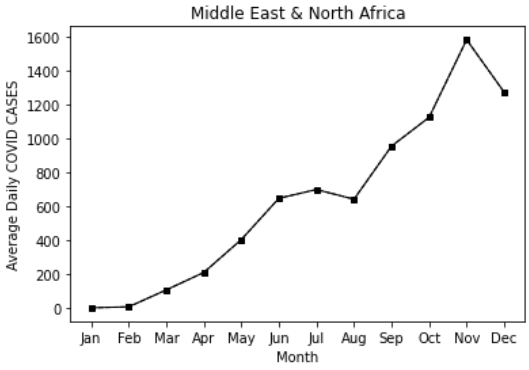
```
statsME20
```

	region	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
72	Middle East & North Africa	2020	Jan	1.333333	2.309401	3.0	-4.403537	7.070204
73	Middle East & North Africa	2020	Feb	7.000000	26.790811	104.0	1.789856	12.210144
74	Middle East & North Africa	2020	Mar	106.802583	387.634558	542.0	74.095364	139.509802
75	Middle East & North Africa	2020	Apr	209.711755	416.318100	621.0	176.904023	242.519488
76	Middle East & North Africa	2020	May	404.010753	637.407884	651.0	354.955620	453.065885
77	Middle East & North Africa	2020	Jun	647.385714	954.622947	630.0	572.698517	722.072912
78	Middle East & North Africa	2020	Jul	699.082949	889.344877	651.0	630.638658	767.527241
79	Middle East & North Africa	2020	Aug	641.867896	894.732215	651.0	573.008993	710.726798
80	Middle East & North Africa	2020	Sep	955.063492	1365.640403	630.0	848.219366	1061.907618
81	Middle East & North Africa	2020	Oct	1124.852535	1487.973491	651.0	1010.337577	1239.367493
82	Middle East & North Africa	2020	Nov	1582.941270	2690.395225	630.0	1372.451799	1793.430740
83	Middle East & North Africa	2020	Dec	1273.310292	2001.556385	651.0	1119.269816	1427.350768

```
statsME20.columns=['region','year','month','mean','std','size','lb','ub']
```

As you can see this graph is the average daily cases of covid-19 the region Middle East and north africa throughout the year 2020 starting from january to december, and as you can see the number of average dsily cases are increasing.

```
plt.plot( 'month', 'mean', data=statsME20, marker='s', color='black', markersize=4,     linewidth=1, linestyle='--')
plt.plot( 'month', 'mean', data=statsME20, marker='o', color='black',  markersize=4, linewidth=1,linestyle='-')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Middle East & North Africa")
plt.show()
```

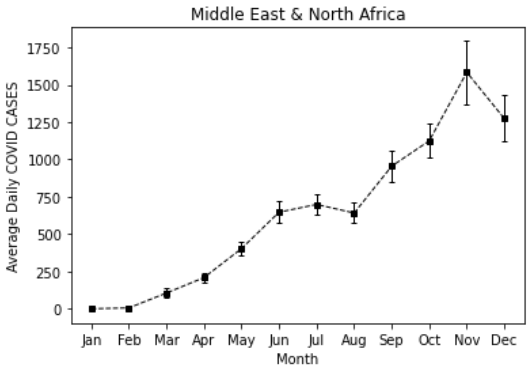



```
ci_lb_ub=[statsME20['lb'],statsME20['ub']]
```

```
err = np.abs(ci_lb_ub - statsME20['mean'].to_numpy())
```

in this below graph you can see the marginal error that could happen through the calculations.

```
plt.errorbar('month', 'mean',  yerr=err, data=statsME20,marker='s', capsize=2,
             color='black', markersize=4, linewidth=1, linestyle='--')
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Middle East & North Africa")
plt.show()
```




```
statsME21=statsdcases[(statsdcases['region']=='Middle East & North Africa')
                      & (statsdcases['year']==2021)]
statsME21
```

	region	year	month	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
84	Middle East & North Africa	2021	Jan	1402.342550	2059.496147	651.0	1243.843009	1560.842090
85	Middle East & North Africa	2021	Feb	1405.054422	1954.685370	588.0	1246.735554	1563.373290
86	Middle East & North Africa	2021	Mar	1714.384025	2338.872107	651.0	1534.383613	1894.384436
87	Middle East & North Africa	2021	Apr	2214.015873	4494.082227	630.0	1862.410659	2565.621087
88	Middle East & North Africa	2021	May	1409.675883	2994.057808	651.0	1179.252152	1640.099615
89	Middle East & North Africa	2021	Jun	1296.085714	2302.221239	630.0	1115.965959	1476.205470
90	Middle East & North Africa	2021	Jul	2359.718894	5094.053939	651.0	1967.678729	2751.759059
91	Middle East & North Africa	2021	Aug	3312.694316	7919.025055	651.0	2703.243391	3922.145242
92	Middle East & North Africa	2021	Sep	1978.706349	4665.790138	630.0	1613.667158	2343.745540
93	Middle East & North Africa	2021	Oct	964.829493	2616.431979	651.0	763.467977	1166.191009
94	Middle East & North Africa	2021	Nov	690.525397	1576.597451	630.0	567.176545	813.874249
95	Middle East & North Africa	2021	Dec	672.032258	1064.245135	651.0	590.127582	753.936934

```
statsME20.columns=['region','year','month','mean','std','size','lb','ub']
```

```
statsME21.columns=['region','year','month','mean','std','size','lb','ub']
```

```
ci_lb_ub20=[statsME20['lb'],statsME20['ub']]
err20 = np.abs(ci_lb_ub20 - statsME20['mean'].to_numpy())
```

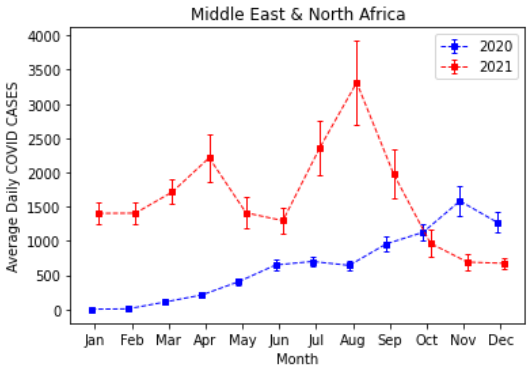
```
ci_lb_ub21=[statsME21['lb'],statsME21['ub']]
err21 = np.abs(ci_lb_ub21 - statsME21['mean'].to_numpy())
```

```
from matplotlib.transforms import Affine2D
```

As you can see in the below graph we included the graph of the year 2021 to the year 2020 to be able to see whether the cases has increased in the following year or not however the cases in Middle east and north africa seems to have increased a lot, however at the ned of the year starting from october the cases of 2020 started tp increase. Now we will be able to compare not only the different years but we can also compare south Asia with Middle east and north africa and as you can see they're similar in the number of cases.

```
fig, ax = plt.subplots()

trans1 = Affine2D().translate(-0.1, 0.0) + ax.transData
trans2 = Affine2D().translate(+0.1, 0.0) + ax.transData
plt.errorbar('month', 'mean',  yerr=err20, data=statsME20,marker='s', capsize=2,
             color='blue', markersize=4, linewidth=1, linestyle='--',transform=trans1)
plt.errorbar('month', 'mean',  yerr=err21, data=statsME21,marker='s', capsize=2,
             color='red', markersize=4, linewidth=1, linestyle='--',transform=trans2)
plt.legend(['2020','2021'])
plt.xlabel("Month")
plt.ylabel("Average Daily COVID CASES")
plt.title("Middle East & North Africa")
plt.show()
```




```
statsdcases=df.groupby(['continent','year','income']).agg({"dcases": [np.mean, np.std, np.size,get_ci_lb,get_ci_ub]})
```

```
statsdcases
```

			dcases				
			mean	std	size	get_ci_lb	get_ci_ub
continent	year	income					
Africa	2020	High income	0.787671	3.579646	292	0.375377	1.199965
		Low income	44.443453	132.969996	6614	41.238297	47.648609
		Lower middle income	189.657246	538.688967	6086	176.120740	203.193753
		Upper middle income	560.328632	1821.377564	2340	486.493181	634.164084
	2021	High income	67.295890	75.206384	365	59.554788	75.036992
		Low income	124.504110	487.424340	8395	114.075952	134.932267
		Lower middle income	348.812394	906.988518	7665	328.504617	369.120171
		Upper middle income	1111.608219	3232.130518	2920	994.327756	1228.888682
Asia	2020	High income	493.063446	805.001966	3830	467.560935	518.565958
		Low income	234.568966	592.926675	1450	204.024819	265.113112
		Lower middle income	2808.383312	11214.023994	4602	2484.304582	3132.462041
		Upper middle income	1260.403746	12681.057660	4431	886.920292	1633.887201
	2021	High income	1118.581050	2276.307266	4380	1051.149681	1186.012419
		Low income	396.871781	1160.547691	1825	343.591325	450.152236
		Lower middle income	6573.092785	30463.243191	5475	5765.991192	7380.194379
		Upper middle income	4458.967906	7567.969641	5110	4251.419252	4666.516560
Europe	2020	High income	1779.303054	5273.235892	10051	1676.199661	1882.406448
		Lower middle income	2042.810945	3495.523582	603	1763.250515	2322.371375
		Upper middle income	1741.089325	4306.900675	2754	1580.164902	1902.013747
	2021	High income	4374.541866	11242.058453	11680	4170.641729	4578.442004
		Lower middle income	4098.035616	5827.432894	730	3674.601757	4521.469476
		Upper middle income	3341.404262	6800.069089	3285	3108.780635	3574.027889
North America(continent)	2020	High income	8639.475780	31304.439194	2436	7395.730907	9883.220653
		Low income	34.839721	52.551228	287	28.734074	40.945369
		Lower middle income	199.801606	264.273075	872	182.236665	217.366546
		Upper middle income	599.063600	1714.741391	3239	539.988615	658.138584
	2021	High income	12546.505822	42428.147946	2920	11006.966250	14086.045394
		Low income	43.797260	73.758460	365	36.205195	51.389325
		Lower middle income	314.811872	572.884005	1095	280.842425	348.781319
		Upper middle income	1191.098879	2969.473694	4015	1099.219982	1282.977776
Oceania	2020	High income	47.132512	105.576757	649	38.994731	55.270292
		Lower middle income	1.938095	6.796811	420	1.286189	2.590001
		Upper middle income	0.141058	0.577571	397	0.084069	0.198046
	2021	High income	474.526682	2065.170313	862	336.468782	612.584582
		Lower middle income	26.752079	94.939477	1323	21.631574	31.872584
		Upper middle income	48.661187	182.408544	1095	37.845180	59.477194
South America(continent)	2020	High income	1034.782537	1573.531300	607	909.353703	1160.211371
		Lower middle income	540.959459	534.276573	296	479.843620	602.075299
		Upper middle income	4604.692450	9993.572248	2702	4227.710261	4981.674639
	2021	High income	2180.527397	2183.763529	730	2021.850423	2339.204372
		Lower middle income	1204.463014	1128.098194	365	1088.346232	1320.579795
		Upper middle income	7470.957686	15944.888037	3285	6925.498862	8016.416511

```
statsdcases=statsdcases.reset_index()
statsdcases
```

	continent	year	income	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
0	Africa	2020	High income	0.787671	3.579646	292	0.375377	1.199965
1	Africa	2020	Low income	44.443453	132.969996	6614	41.238297	47.648609
2	Africa	2020	Lower middle income	189.657246	538.688967	6086	176.120740	203.193753
3	Africa	2020	Upper middle income	560.328632	1821.377564	2340	486.493181	634.164084
4	Africa	2021	High income	67.295890	75.206384	365	59.554788	75.036992
5	Africa	2021	Low income	124.504110	487.424340	8395	114.075952	134.932267
6	Africa	2021	Lower middle income	348.812394	906.988518	7665	328.504617	369.120171
7	Africa	2021	Upper middle income	1111.608219	3232.130518	2920	994.327756	1228.888682
8	Asia	2020	High income	493.063446	805.001966	3830	467.560935	518.565958
9	Asia	2020	Low income	234.568966	592.926675	1450	204.024819	265.113112
10	Asia	2020	Lower middle income	2808.383312	11214.023994	4602	2484.304582	3132.462041
11	Asia	2020	Upper middle income	1260.403746	12681.057660	4431	886.920292	1633.887201
12	Asia	2021	High income	1118.581050	2276.307266	4380	1051.149681	1186.012419
13	Asia	2021	Low income	396.871781	1160.547691	1825	343.591325	450.152236
14	Asia	2021	Lower middle income	6573.092785	30463.243191	5475	5765.991192	7380.194379
15	Asia	2021	Upper middle income	4458.967906	7567.969641	5110	4251.419252	4666.516560
16	Europe	2020	High income	1779.303054	5273.235892	10051	1676.199661	1882.406448
17	Europe	2020	Lower middle income	2042.810945	3495.523582	603	1763.250515	2322.371375
18	Europe	2020	Upper middle income	1741.089325	4306.900675	2754	1580.164902	1902.013747
19	Europe	2021	High income	4374.541866	11242.058453	11680	4170.641729	4578.442004
20	Europe	2021	Lower middle income	4098.035616	5827.432894	730	3674.601757	4521.469476
21	Europe	2021	Upper middle income	3341.404262	6800.069089	3285	3108.780635	3574.027889
22	North America(continent)	2020	High income	8639.475780	31304.439194	2436	7395.730907	9883.220653
23	North America(continent)	2020	Low income	34.839721	52.551228	287	28.734074	40.945369
24	North America(continent)	2020	Lower middle income	199.801606	264.273075	872	182.236665	217.366546
25	North America(continent)	2020	Upper middle income	599.063600	1714.741391	3239	539.988615	658.138584
26	North America(continent)	2021	High income	12546.505822	42428.147946	2920	11006.966250	14086.045394
27	North America(continent)	2021	Low income	43.797260	73.758460	365	36.205195	51.389325
28	North America(continent)	2021	Lower middle income	314.811872	572.884005	1095	280.842425	348.781319
29	North America(continent)	2021	Upper middle income	1191.098879	2969.473694	4015	1099.219982	1282.977776
30	Oceania	2020	High income	47.132512	105.576757	649	38.994731	55.270292
31	Oceania	2020	Lower middle income	1.938095	6.796811	420	1.286189	2.590001
32	Oceania	2020	Upper middle income	0.141058	0.577571	397	0.084069	0.198046
33	Oceania	2021	High income	474.526682	2065.170313	862	336.468782	612.584582
34	Oceania	2021	Lower middle income	26.752079	94.939477	1323	21.631574	31.872584
35	Oceania	2021	Upper middle income	48.661187	182.408544	1095	37.845180	59.477194
36	South America(continent)	2020	High income	1034.782537	1573.531300	607	909.353703	1160.211371
37	South America(continent)	2020	Lower middle income	540.959459	534.276573	296	479.843620	602.075299
38	South America(continent)	2020	Upper middle income	4604.692450	9993.572248	2702	4227.710261	4981.674639
39	South America(continent)	2021	High income	2180.527397	2183.763529	730	2021.850423	2339.204372
40	South America(continent)	2021	Lower middle income	1204.463014	1128.098194	365	1088.346232	1320.579795
41	South America(continent)	2021	Upper middle income	7470.957686	15944.888037	3285	6925.498862	8016.416511

```
statsHI20=statsdcases[(statsdcases['income']=='High income') & (statsdcases['year']==2020)]
```

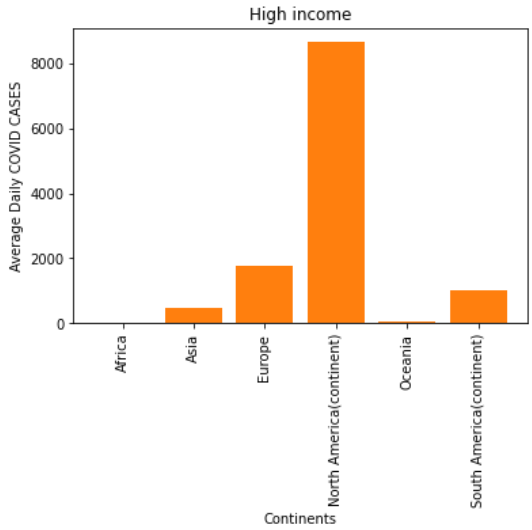
```
statsHI20
```

	continent	year	income	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
0	Africa	2020	High income	0.787671	3.579646	292	0.375377	1.199965
8	Asia	2020	High income	493.063446	805.001966	3830	467.560935	518.565958
16	Europe	2020	High income	1779.303054	5273.235892	10051	1676.199661	1882.406448
22	North America(continent)	2020	High income	8639.475780	31304.439194	2436	7395.730907	9883.220653
30	Oceania	2020	High income	47.132512	105.576757	649	38.994731	55.270292
36	South America(continent)	2020	High income	1034.782537	1573.531300	607	909.353703	1160.211371

```
statsHI20.columns=['continent','year','income','mean','std','size','lb','ub']
```

as you can see in the following graph the high income people have been infected to covid but not a lot. we can conclude from this that these people have money which means they have money to protect them selves hygenically from the virus

```
plt.bar( 'continent', 'mean', data=statsHI20)
plt.bar( 'continent', 'mean', data=statsHI20)
plt.xlabel("Continents")
plt.ylabel("Average Daily COVID CASES")
plt.xticks(rotation=90)
plt.title("High income")
plt.show()
```

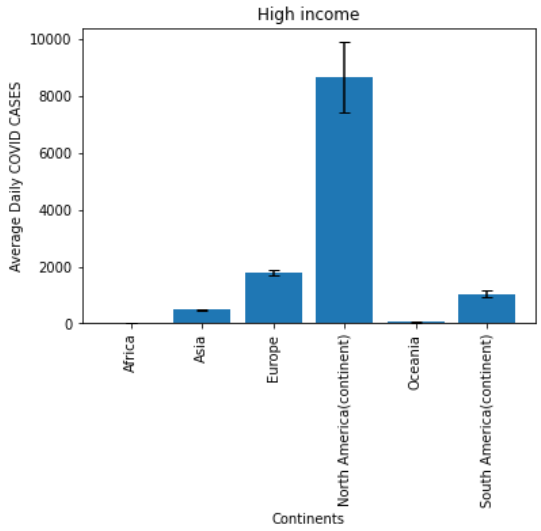



```
ci_lb_ub=[statsHI20['lb'],statsHI20['ub']]
```

```
err = np.abs(ci_lb_ub - statsHI20['mean'].to_numpy())
```

here we can see the margin of error because people could've done some mistakes while calculating

```
plt.bar('continent', 'mean',  yerr=err, data=statsHI20, capsize=4)
plt.xlabel("Continents")
plt.ylabel("Average Daily COVID CASES")
plt.xticks(rotation=90)
plt.title("High income")
plt.show()
```




```
statsHI21=statsdcases[(statsdcases['income']=='High income')
                      & (statsdcases['year']==2021)]
statsHI21
```

	continent	year	income	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
4	Africa	2021	High income	67.295890	75.206384	365	59.554788	75.036992
12	Asia	2021	High income	1118.581050	2276.307266	4380	1051.149681	1186.012419
19	Europe	2021	High income	4374.541866	11242.058453	11680	4170.641729	4578.442004
26	North America(continent)	2021	High income	12546.505822	42428.147946	2920	11006.966250	14086.045394
33	Oceania	2021	High income	474.526682	2065.170313	862	336.468782	612.584582
39	South America(continent)	2021	High income	2180.527397	2183.763529	730	2021.850423	2339.204372

```
statsHI20.columns=['continent','year','income','mean','std','size','lb','ub']
```

```
statsHI21.columns=['continent','year','income','mean','std','size','lb','ub']
```

```
ci_lb_ub20=[statsHI20['lb'],statsHI20['ub']]
err20 = np.abs(ci_lb_ub20 - statsHI20['mean'].to_numpy())
```

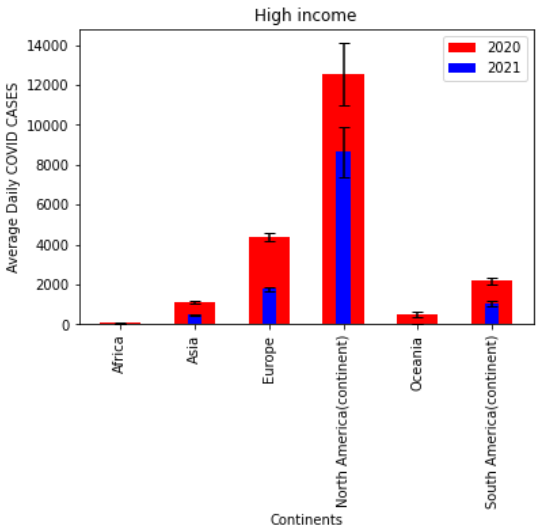
```
ci_lb_ub21=[statsHI21['lb'],statsHI21['ub']]
err21 = np.abs(ci_lb_ub21 - statsHI21['mean'].to_numpy())
```

```
from matplotlib.transforms import Affine2D
```

This graph includes 2020 and 2021 together here we can see that in 2020 cases are much more unlike the others. we can conclude from this that high income people had money to be vaccinated first and become immuned from the virus.

```
fig, ax = plt.subplots()

trans1 = Affine2D().translate(-0.1, 0.0) + ax.transData
trans2 = Affine2D().translate(+0.1, 0.0) + ax.transData
plt.bar('continent', 'mean',  yerr=err21, data=statsHI21, capsize=4,
            color='red',width=0.55)
plt.bar('continent', 'mean',  yerr=err20, data=statsHI20, capsize=4,
            color='blue',width=0.55/3)
plt.legend(['2020','2021'])
plt.xlabel('Continents')
plt.xticks(rotation=90)
plt.ylabel("Average Daily COVID CASES")
plt.title("High income")
plt.show()
```




```
statsLMI20=statsdcases[(statsdcases['income']=='Lower middle income') & (statsdcases['year']==2020)]
```

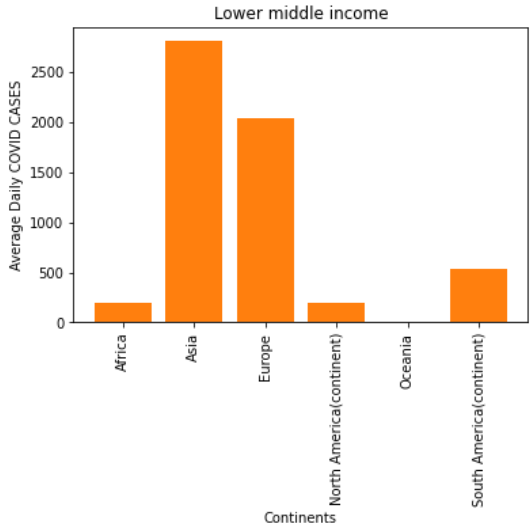
```
statsLMI20
```

	continent	year	income	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
2	Africa	2020	Lower middle income	189.657246	538.688967	6086	176.120740	203.193753
10	Asia	2020	Lower middle income	2808.383312	11214.023994	4602	2484.304582	3132.462041
17	Europe	2020	Lower middle income	2042.810945	3495.523582	603	1763.250515	2322.371375
24	North America(continent)	2020	Lower middle income	199.801606	264.273075	872	182.236665	217.366546
31	Oceania	2020	Lower middle income	1.938095	6.796811	420	1.286189	2.590001
37	South America(continent)	2020	Lower middle income	540.959459	534.276573	296	479.843620	602.075299

```
statsLMI20.columns=['continent','year','income','mean','std','size','lb','ub']
```

This graph shows that the lower middle income class have had a lot more of covid 19 cases. we can conclude that they didn't have enough of money to stay hygenically clean from the virus and they weren't able to claim the vaccine like the high income people.

```
plt.bar( 'continent', 'mean', data=statsLMI20)
plt.bar( 'continent', 'mean', data=statsLMI20)
plt.xlabel("Continents")
plt.ylabel("Average Daily COVID CASES")
plt.xticks(rotation=90)
plt.title("Lower middle income")
plt.show()
```

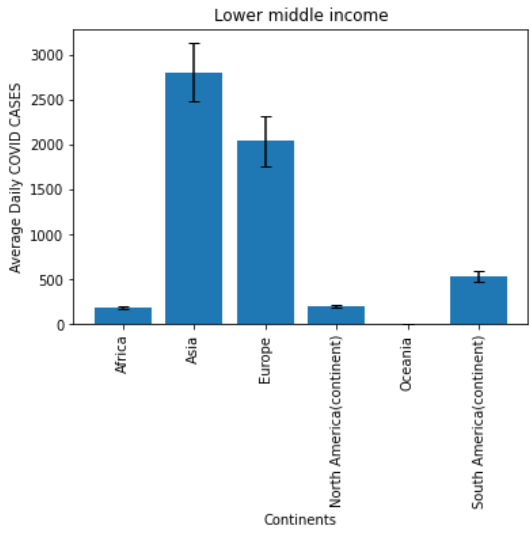



```
ci_lb_ub=[statsLMI20['lb'],statsLMI20['ub']]
```

```
err = np.abs(ci_lb_ub - statsLMI20['mean'].to_numpy())
```

here is a margin of error as people could've made a mistake while calculating the number of people

```
plt.bar('continent', 'mean',  yerr=err, data=statsLMI20, capsize=4)
plt.xlabel("Continents")
plt.ylabel("Average Daily COVID CASES")
plt.xticks(rotation=90)
plt.title("Lower middle income")
plt.show()
```



```
statsLMI21=statsdcases[(statsdcases['income']=='Lower middle income')
                       & (statsdcases['year']==2021)]
statsLMI21
```

	continent	year	income	dcases				
				mean	std	size	get_ci_lb	get_ci_ub
6	Africa	2021	Lower middle income	348.812394	906.988518	7665	328.504617	369.120171
14	Asia	2021	Lower middle income	6573.092785	30463.243191	5475	5765.991192	7380.194379
20	Europe	2021	Lower middle income	4098.035616	5827.432894	730	3674.601757	4521.469476
28	North America(continent)	2021	Lower middle income	314.811872	572.884005	1095	280.842425	348.781319
34	Oceania	2021	Lower middle income	26.752079	94.939477	1323	21.631574	31.872584
40	South America(continent)	2021	Lower middle income	1204.463014	1128.098194	365	1088.346232	1320.579795

```
statsLMI20.columns=['continent','year','income','mean','std','size','lb','ub']
```

```
statsLMI21.columns=['continent','year','income','mean','std','size','lb','ub']
```

```
ci_lb_ub20=[statsLMI20['lb'],statsLMI20['ub']]
err20 = np.abs(ci_lb_ub20 - statsHI20['mean'].to_numpy())
```

```
ci_lb_ub21=[statsLMI21['lb'],statsLMI21['ub']]
err21 = np.abs(ci_lb_ub21 - statsLMI21['mean'].to_numpy())
```

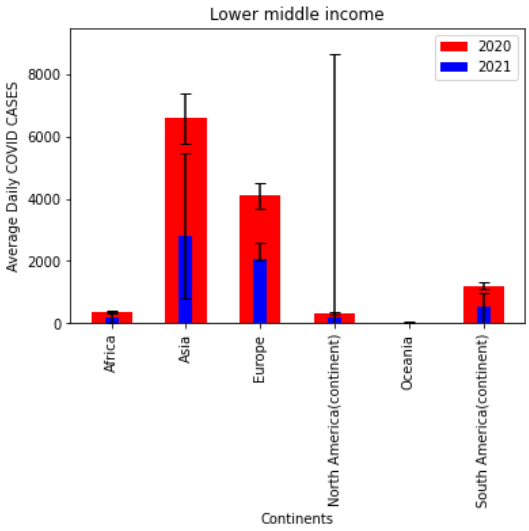
```
from matplotlib.transforms import Affine2D
```

```
import matplotlib.pyplot as plt
```

here we have the 2020 and 2021 graoh and we can conclude that 2020 had more cases probably because they had the chance to take the vaccination and become immuned. Now we will also be able to compare it with the high income and the high income have less covid-19 cases because they probably wre able to stay at hospital and have better medical conditions

```
fig, ax = plt.subplots()

trans1 = Affine2D().translate(-0.1, 0.0) + ax.transData
trans2 = Affine2D().translate(+0.1, 0.0) + ax.transData
plt.bar('continent', 'mean',  yerr=err21, data=statsLMI21, capsize=4,
            color='red',width=0.55)
plt.bar('continent', 'mean',  yerr=err20, data=statsLMI20, capsize=4,
            color='blue',width=0.55/3)
plt.ylim(ymin=0)
plt.legend(['2020','2021'])
plt.xlabel('Continents')
plt.xticks(rotation=90)
plt.ylabel("Average Daily COVID CASES")
plt.title("Lower middle income")
plt.show()
```


Lower middle income
2020
2021
Average Daily COVID CASES
0
2000
4000
6000
8000
Africa
Asia
Europe
North America(continent)
Oceania
South America(continent)
Continents