Sheet

Assignment 2

by: Mohamed Hatem El-Badry 900211356

```
!pip install pandas
import pandas as pd
!pip install numpy
import numpy as np
```

```
Requirement already satisfied: pandas in /opt/python/envs/default/lib/python3.8/site-packages (1.3.5)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/python/envs/default/lib/python3.8/site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.17.3 in /opt/python/envs/default/lib/python3.8/site-packages (from pandas) (1.21.5)
Requirement already satisfied: pytz>=2017.3 in /opt/python/envs/default/lib/python3.8/site-packages (from pandas) (2022.1)
Requirement already satisfied: six>=1.5 in /opt/python/envs/default/lib/python3.8/site-packages (from python-dateutil>=2.7.3->pandas
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is available.
You should consider upgrading via the '/opt/python/envs/default/bin/python -m pip install --upgrade pip' command.
Requirement already satisfied: numpy in /opt/python/envs/default/lib/python3.8/site-packages (1.21.5)
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is available.
You should consider upgrading via the '/opt/python/envs/default/bin/python -m pip install --upgrade pip' command.
```

Part I:

Statistical analysis of results of international football matches starting from 1872 up to 2022

The dataset in results.csv includes 43,170 results of international football matches starting from the very first official match in 1872 up to 2019. The matches range from FIFA World Cup to FIFI Wild Cup to regular friendly matches. The matches are strictly men's full internationals and the data does not include Olympic Games or matches where at least one of the teams was the nation's B-team, U-23 or a league select team.

in this part, i will analyze the following: the probability of 3 different European countries' winning chance in comparison to Egypt's, in friendly tournament in home land

```
df= pd.read_csv('results.csv' , encoding='latin-1')
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	YaoundÃ©	Cameroon	True
43185	2/3/2022	Cameroon	Egypt	0	0	African Cup of Nations	YaoundÃ©	Cameroon	False
43186	2/5/2022	Cameroon	Burkina Faso	3	3	African Cup of Nations	YaoundÃ©	Cameroon	False
43187	2/6/2022	Senegal	Egypt	0	0	African Cup of Nations	YaoundÃ©	Cameroon	True

43188 rows × 9 columns

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

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

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

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

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

```
x=df['result_home'].value_counts()
x=np.array(x)
```

```
x.sum()
```

```
43188
```

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

```
df_noneutral.shape
```

```
(32481, 10)
```

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

```
df_noneutralegy=df_noneutral[df_noneutral['country']=='Egypt'] #probability that egypt wins in it land
```

```
df_noneutralegyF=df_noneutralegy[df_noneutralegy['tournament']=='Friendly']
df_noneutralegyF  #probability of Egypt Winning in their land in friendly tournament
```

	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
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
3425	1/16/1953	Egypt	Yugoslavia	1	3	Friendly	Cairo	Egypt	False	lose
...	...	...	...	...	...	...	...	...	...	...
40925	6/13/2019	Egypt	Tanzania	1	0	Friendly	Alexandria	Egypt	False	win
40947	6/16/2019	Egypt	Guinea	3	1	Friendly	Alexandria	Egypt	False	win
41450	10/14/2019	Egypt	Botswana	1	0	Friendly	Cairo	Egypt	False	win
41514	11/7/2019	Egypt	Liberia	1	0	Friendly	Alexandria	Egypt	False	win
42758	9/30/2021	Egypt	Liberia	2	0	Friendly	Alexandria	Egypt	False	win

145 rows × 10 columns

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

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

```
x=df_noneutralegyF['result_home'].value_counts()
x=np.array(x)
x
```

```
N=x.sum()
```

```
CI_egy=proportion_confint(count=x[0], nobs=N, alpha=(1-.95))
CI_egy
```

```
(0.47077769679111225, 0.6326705790709567)
```

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

```
df_noneutralgreF=df_noneutralgre[df_noneutralgre['tournament']=='Friendly']
```

```
df_noneutralgreF.shape
```

```
(126, 10)
```

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

```
x=df_noneutralgreF['result_home'].value_counts()
x=np.array(x)
N=x.sum()
CI_gre=proportion_confint(count=x[0], nobs=N, alpha=(1-.95))
CI_gre
```

```
(0.3114005769651982, 0.4822502166855954)
```

```
df_noneutralwal=df_noneutral[df_noneutral['country']=='Wales']
df_noneutralwalF=df_noneutralwal[df_noneutralwal['tournament']=='Friendly']
df_noneutralwalF['result_home'].value_counts(normalize=True)
```

```
x=df_noneutralwalF['result_home'].value_counts()
x=np.array(x)
N=x.sum()
CI_wal=proportion_confint(count=x[0], nobs=N, alpha=(1-.95))
CI_wal
```

```
(0.2541364398001451, 0.478257926397038)
```

```
df_noneutralscot=df_noneutral[df_noneutral['country']=='Scotland']
df_noneutralscotF=df_noneutralscot[df_noneutralscot['tournament']=='Friendly']
df_noneutralscotF['result_home'].value_counts(normalize=True)
```

```
x=df_noneutralscotF['result_home'].value_counts()
x=np.array(x)
N=x.sum()
CI_scot=proportion_confint(count=x[0], nobs=N, alpha=(1-.95))
CI_scot
```

```
(0.4048157345543417, 0.5951842654456583)
```

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

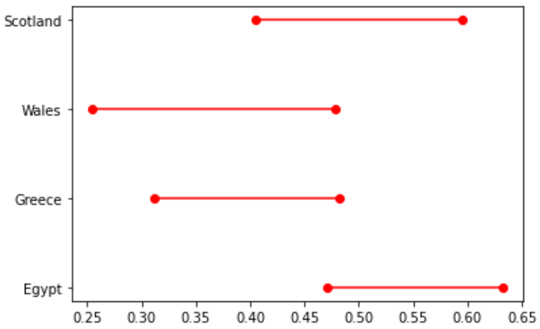
```
ci_friendly = {}
ci_friendly['country'] = ['Egypt','Greece','Wales', 'Scotland']
ci_friendly['lb'] = [CI_egy[0],CI_gre[0],CI_wal[0], CI_scot[0]]
ci_friendly['ub'] = [CI_egy[1],CI_gre[1],CI_wal[1], CI_scot[1]]
df_ci= pd.DataFrame(ci_friendly)
df_ci
```

	country	lb	ub
0	Egypt	0.470778	0.632671
1	Greece	0.311401	0.482250
2	Wales	0.254136	0.478258
3	Scotland	0.404816	0.595184

```
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['country'])) #a graph for the confidence interval of 4 different countries' winning chance in
```

```
([<matplotlib.axis.YTick at 0x7feefc655e20>,
  <matplotlib.axis.YTick at 0x7feefc655670>,
  <matplotlib.axis.YTick at 0x7feefc911670>,
  <matplotlib.axis.YTick at 0x7feefc627340>],
 [Text(0, 0, 'Egypt'),
  Text(0, 1, 'Greece'),
  Text(0, 2, 'Wales'),
  Text(0, 3, 'Scotland')])
```



this can make us determine that Egypt has the higher chance of winning a friendly tournament done on their homeland among the 4 countries; as Egypt's team members are more used to play in national matches than internati0nal ones

In this part, im going to analyze the following: the probability of losing of the same 3 European countries in comparison with Egypt, in a friendly tournament, playing as the away team.

```
x=df['home_score']-df['away_score']
conditions=[(x<0),(x>0), (x==0)]
values=['win','lose','draw']
df['result_away']=np.select(conditions,values)
df
```

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

43188 rows × 11 columns

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

```
df_noneutralegy=df_noneutral[df_noneutral['away_team']=='Egypt']
df_noneutralegyF=df_noneutralegy[df_noneutralegy['tournament']=='Friendly']
x=df_noneutralegyF['result_away'].value_counts()
x=np.array(x)
x
N=x.sum()
CI_egy=proportion_confint(count=x[1], nobs=N, alpha=(1-.95))
CI_egy
```

```
(0.24715478691048412, 0.43284521308951596)
```

```
df_noneutralgre=df_noneutral[df_noneutral['away_team']=='Greece']
df_noneutralgreF=df_noneutralgre[df_noneutralgre['tournament']=='Friendly']
x=df_noneutralgreF['result_away'].value_counts()
x=np.array(x)
x
N=x.sum()
CI_gre=proportion_confint(count=x[1], nobs=N, alpha=(1-.95))
CI_gre
```

```
(0.22828832246685596, 0.39779863405488325)
```

```
df_noneutralwal=df_noneutral[df_noneutral['away_team']=='Wales']
df_noneutralwalF=df_noneutralwal[df_noneutralwal['tournament']=='Friendly']
x=df_noneutralwalF['result_away'].value_counts()
x=np.array(x)
x
N=x.sum()
CI_wal=proportion_confint(count=x[1], nobs=N, alpha=(1-.95))
CI_wal
```

```
(0.18082080408154663, 0.37339606338833287)
```

```
df_noneutralscot=df_noneutral[df_noneutral['away_team']=='Scotland']
df_noneutralscotF=df_noneutralscot[df_noneutralscot['tournament']=='Friendly']
x=df_noneutralscotF['result_away'].value_counts()
x=np.array(x)
x
N=x.sum()
CI_scot=proportion_confint(count=x[1], nobs=N, alpha=(1-.95))
CI_scot
```

```
(0.2936725636434827, 0.466658014868914)
```

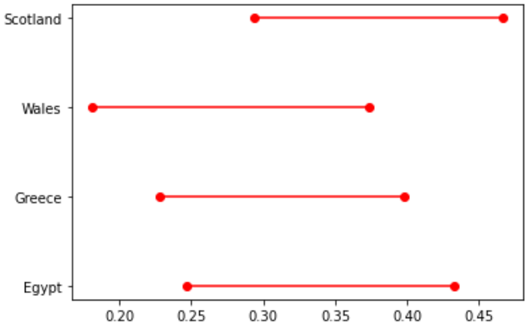
```
ci_friendlyL = {}
ci_friendlyL['country'] = ['Egypt','Greece','Wales', 'Scotland']
ci_friendlyL['lb'] = [CI_egy[0],CI_gre[0],CI_wal[0], CI_scot[0]]
ci_friendlyL['ub'] = [CI_egy[1],CI_gre[1],CI_wal[1], CI_scot[1]]
df_ci= pd.DataFrame(ci_friendlyL)
df_ci
```

	country	lb	ub
0	Egypt	0.247155	0.432845
1	Greece	0.228288	0.397799
2	Wales	0.180821	0.373396
3	Scotland	0.293673	0.466658

```
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['country']))
```

```
([<matplotlib.axis.YTick at 0x7feefc627700>,
  <matplotlib.axis.YTick at 0x7feefc299610>,
  <matplotlib.axis.YTick at 0x7feefc2997f0>,
  <matplotlib.axis.YTick at 0x7fef06e07340>],
 [Text(0, 0, 'Egypt'),
  Text(0, 1, 'Greece'),
  Text(0, 2, 'Wales'),
  Text(0, 3, 'Scotland')])
```



This shows that scotland has the higher probability of losing in a friendly tournament away from their home. this is due to the fact that scotland doesnt have a strong soccer team. it can be shown in how they have never progressed beyond the first group stage of a finals tournament.

In this part, im going to analyze the following: the probability that Egypt wins in 3 different tournaments (Friendly, FIFA world cup, and African cup of nations) as the away team

```
y=list(df['tournament'].value_counts().index)
y
```

```
['Friendly',
 'FIFA World Cup qualification',
 'UEFA Euro qualification',
 'African Cup of Nations qualification',
 'FIFA World Cup',
 'Copa AmÃ©rica',
 'African Cup of Nations',
 'AFC Asian Cup qualification',
 'CECAFA Cup',
 'CFU Caribbean Cup qualification',
 'Merdeka Tournament',
 'British Championship',
 'Gulf Cup',
 'AFC Asian Cup',
 'Gold Cup',
 'Island Games',
 'UEFA Euro',
 'COSAFA Cup',
 'UEFA Nations League',
 'AFF Championship',
 'Nordic Championship',
 'African Nations Championship',
 'CFU Caribbean Cup',
 'AmÃ\xadlcar Cabral Cup',
 "King's Cup",
 'South Pacific Games',
 'UNCAF Cup',
 'Korea Cup',
 'SAFF Cup',
 'Arab Cup',
 'Confederations Cup',
 'International Cup',
 'CCCF Championship',
 'EAFF Championship',
 'CONCACAF Nations League',
 'Windward Islands Tournament',
 'CONIFA World Football Cup',
 'Oceania Nations Cup',
 'AFC Challenge Cup',
 'WAFF Championship',
 'Baltic Cup',
 'AFC Challenge Cup qualification',
 'Nehru Cup',
 'Balkan Cup',
 'Indonesia Tournament',
 'Oceania Nations Cup qualification',
 'Cyprus International Tournament',
 'Kirin Cup',
 'CONCACAF Nations League qualification',
 'Gold Cup qualification',
 'UDEAC Cup',
 'African Nations Championship qualification',
 'Vietnam Independence Cup',
 'Palestine Cup',
 'Viva World Cup',
 'West African Cup',
 'Malta International Tournament',
 'Pacific Games',
 'CONIFA European Football Cup',
 'CONCACAF Championship',
 'Pan American Championship',
 'Brazil Independence Cup',
 'USA Cup',
 'United Arab Emirates Friendship Tournament',
 'Copa Chevallier Boutell',
 'Dynasty Cup',
 'Copa Lipton',
 'COSAFA Cup qualification',
 'Copa Newton',
 'Lunar New Year Cup',
 'Merlion Cup',
 'Arab Cup qualification',
 'Copa Paz del Chaco',
 'Copa Roca',
 "Prime Minister's Cup",
 'CONCACAF Championship qualification',
 'ABCS Tournament',
 'Inter Games Football Tournament',
 'Copa del PacÃ\xadfico',
 'Copa Rio Branco',
 'Simba Tournament',
 'Copa Carlos Dittborn',
 'Copa Juan Pinto DurÃ¡n',
 'Copa Oswaldo Cruz',
 'ELF Cup',
 'UNIFFAC Cup',
 'Millennium Cup',
 'Copa Premio Honor Uruguayo',
 'Dunhill Cup',
 'GaNEFo',
 'Nile Basin Tournament',
 'Intercontinental Cup',
 'Copa Artigas',
 'Jordan International Tournament',
 'King Hassan II Tournament',
 'Copa Premio Honor Argentino',
 'SKN Football Festival',
 'Rous Cup',
 'Atlantic Cup',
 'FIFI Wild Cup',
 "Copa Bernardo O'Higgins",
 'Tournoi de France',
 'Bolivarian Games',
 'Beijing International Friendship Tournament',
 'VFF Cup',
 'Mahinda Rajapaksa Cup',
 'Mundialito',
 'NAFU Championship',
 'Nations Cup',
 'Copa RamÃ³n Castilla',
 'Copa FÃ©lix Bogado',
 'World Unity Cup',
 'Guangzhou International Friendship Tournament',
 'Afro-Asian Games',
 'Dragon Cup',
 'Matthews Cup',
 'Dakar Tournament',
 'OSN Cup',
 'Great Wall Cup',
 'Three Nations Cup',
 'Copa AmÃ©rica qualification',
 'AFF Championship qualification',
 'Atlantic Heritage Cup',
 'Cup of Ancient Civilizations',
 'FIFA 75th Anniversary Cup',
 'TIFOCO Tournament']
```

```
df_noneutralegy=df_noneutral[df_noneutral['away_team']=='Egypt']
df_noneutralegyF=df_noneutralegy[df_noneutralegy['tournament']=='Friendly']
x=df_noneutralegyF['result_away'].value_counts()
x=np.array(x)
x
N=x.sum()
CI_egyF=proportion_confint(count=x[0], nobs=N, alpha=(1-.95))
CI_egyF
```

```
(0.28486600512143223, 0.4751339948785678)
```

```
df_noneutralegy=df_noneutral[df_noneutral['away_team']=='Egypt']
df_noneutralegyF=df_noneutralegy[df_noneutralegy['tournament']=='FIFA World Cup qualification']
x=df_noneutralegyF['result_away'].value_counts()
x=np.array(x)
x
N=x.sum()
CI_egyFIFA=proportion_confint(count=x[0], nobs=N, alpha=(1-.95))
CI_egyFIFA
```

```
(0.24249192186541954, 0.5302353508618531)
```

```
df_noneutralegy=df_noneutral[df_noneutral['away_team']=='Egypt']
df_noneutralegyF=df_noneutralegy[df_noneutralegy['tournament']=='African Cup of Nations qualification']
x=df_noneutralegyF['result_away'].value_counts()
x=np.array(x)
x
N=x.sum()
CI_egyAFRI=proportion_confint(count=x[0], nobs=N, alpha=(1-.95))
CI_egyAFRI
```

```
(0.3149304774470007, 0.6324379436056309)
```

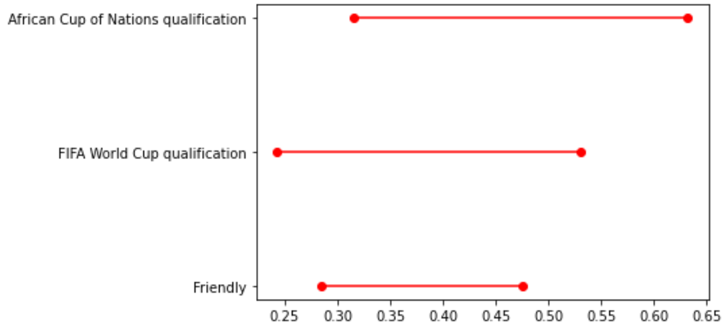
```
ci_tour = {}
ci_tour['tournament'] = ['Friendly','FIFA World Cup qualification','African Cup of Nations qualification']
ci_tour['lb'] = [CI_egyF[0],CI_egyFIFA[0],CI_egyAFRI[0]]
ci_tour['ub'] = [CI_egyF[1],CI_egyFIFA[1],CI_egyAFRI[1]]
df_ci= pd.DataFrame(ci_tour)
df_ci
```

	tournament	lb	ub
0	Friendly	0.284866	0.475134
1	FIFA World Cup qualification	0.242492	0.530235
2	African Cup of Nations qualification	0.314930	0.632438

```
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['tournament']))
```

```
([<matplotlib.axis.YTick at 0x7fef06dc5c10>,
  <matplotlib.axis.YTick at 0x7fef06dc5490>,
  <matplotlib.axis.YTick at 0x7fef06dbc3d0>],
 [Text(0, 0, 'Friendly'),
  Text(0, 1, 'FIFA World Cup qualification'),
  Text(0, 2, 'African Cup of Nations qualification')])
```



the graph shows that Egypt has a higher chance in winning in African Cup of Nations qualification as the away team rather than the other two tournaments. this is because Egypt often doesnt qualify for the FIFA world cup, and when it does, it often faces strong opponents which lead to the team losing. also, it would make sense for Egypt to have a higher probability of winning the African Cup of Nations than friendly matches just for the incentive that they would be named the best in the region.

Part II:

Statistical analysis of Coronavirus Pandemic (COVID 29) over 267 countries

The dataset in covid_data.csv includes the records of two years 2020 and 2021 in the countries affected by the COVID-19 pandemic.

In this part, im going to analyze the following: the probability of cases for each day of the week

```
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

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

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

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

```
stats
```

	dcases		
	mean	std	size
weekdays			
Fri	567.161616	428.533849	99
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

```
ci95_h = []
ci95_l = []
```

```
import scipy.stats
```

```
stats.index
```

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

```
for i in stats.index:
 m, s, n = stats.loc[i]
 x=scipy.stats.t.interval(.95, n-1, m,s/np.sqrt(n-1))
 ci95_h.append(x[1])
 ci95_l.append(x[0])
```

```
ci95_h
```

```
[653.0661477518557,
 643.8071867945697,
 630.633343760431,
 650.9457415593621,
 650.614430845941,
 643.3640186561778,
 650.3101288341297]
```

```
ci95_l
```

```
[481.2570845713766,
 473.80505810338957,
 460.4074725660996,
 472.74813599165833,
 481.6916916030385,
 479.5951650172916,
 485.0572181046457]
```

```
stats['ci95_hi'] = ci95_h
stats['ci95_lo'] = ci95_l
print(stats)
```

```
              dcases                      ci95_hi     ci95_lo
                mean         std size                        
weekdays                                                     
Fri       567.161616  428.533849   99  653.066148  481.257085
Sat       558.806122  421.803605   98  643.807187  473.805058
Sun       545.520408  422.358748   98  630.633344  460.407473
Mon       561.846939  442.137949   98  650.945742  472.748136
Tue       566.153061  419.125460   98  650.614431  481.691692
Wed       561.479592  406.337812   98  643.364019  479.595165
Thu       567.683673  410.020004   98  650.310129  485.057218
```

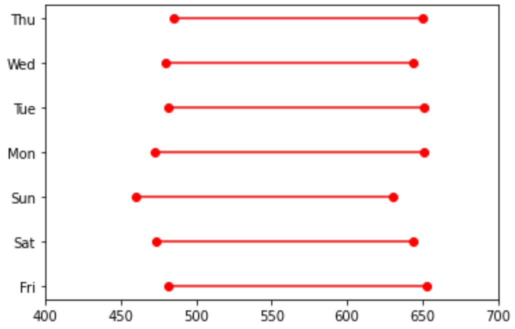
```
df_ci= pd.DataFrame(stats)
df_ci['weekdays']=df_ci.index
```

```
df_ci
```

	dcases			ci95_hi	ci95_lo	weekdays
	mean	std	size			
weekdays						
Fri	567.161616	428.533849	99	653.066148	481.257085	Fri
Sat	558.806122	421.803605	98	643.807187	473.805058	Sat
Sun	545.520408	422.358748	98	630.633344	460.407473	Sun
Mon	561.846939	442.137949	98	650.945742	472.748136	Mon
Tue	566.153061	419.125460	98	650.614431	481.691692	Tue
Wed	561.479592	406.337812	98	643.364019	479.595165	Wed
Thu	567.683673	410.020004	98	650.310129	485.057218	Thu

```
for lb,ub,y in zip(df_ci['ci95_lo'],df_ci['ci95_hi'],range(len(df_ci))):
    plt.plot((lb,ub),(y,y),'ro-')
plt.yticks(range(len(df_ci)),list(df_ci['weekdays']))
plt.xlim([400, 700])
```

```
(400.0, 700.0)
```



This shows that all the days in the week have nearly the same probability of cases, with Sunday having the least probability. that's because in all countries Saturday is a day off so it is logical that people would isolate that day and that would decrease the number of cases reported on Sunday

in this part, im going to make a new dataframe with the fatality rate (deaths/cases) for each row, a dataframe for 2020, and a dataframe for 2021

```
fatality= df['ddeaths']/df['dcases']
df['fatality'] = fatality
df['fatality'] = df['fatality'].fillna(0)
df
```

	date	iso3c	country	income	region	continent	dcases	ddeaths	population	weekdays	month	fatality
0	2020-02-24	AFG	Afghanistan	Low income	South Asia	Asia	5	0	38041754	Mon	Feb	0.000000
1	2020-02-25	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Tue	Feb	0.000000
2	2020-02-26	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Wed	Feb	0.000000
3	2020-02-27	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Thu	Feb	0.000000
4	2020-02-28	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Fri	Feb	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...
122838	2021-12-27	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	1098	17	14645468	Mon	Dec	0.015483
122839	2021-12-28	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	2099	32	14645468	Tue	Dec	0.015245
122840	2021-12-29	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	0	0	14645468	Wed	Dec	0.000000
122841	2021-12-30	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	4180	57	14645468	Thu	Dec	0.013636
122842	2021-12-31	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	1530	7	14645468	Fri	Dec	0.004575

122843 rows × 12 columns

```
mask = (df['date'] > '2019-12-31') & (df['date'] <= '2020-12-31')
df2020=df.loc[mask]
df2020
```

	date	iso3c	country	income	region	continent	dcases	ddeaths	population	weekdays	month	fatality
0	2020-02-24	AFG	Afghanistan	Low income	South Asia	Asia	5	0	38041754	Mon	Feb	0.000000
1	2020-02-25	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Tue	Feb	0.000000
2	2020-02-26	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Wed	Feb	0.000000
3	2020-02-27	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Thu	Feb	0.000000
4	2020-02-28	AFG	Afghanistan	Low income	South Asia	Asia	0	0	38041754	Fri	Feb	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...
122473	2020-12-27	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	114	8	14645468	Sun	Dec	0.070175
122474	2020-12-28	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	71	5	14645468	Mon	Dec	0.070423
122475	2020-12-29	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	177	5	14645468	Tue	Dec	0.028249
122476	2020-12-30	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	300	1	14645468	Wed	Dec	0.003333
122477	2020-12-31	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	242	3	14645468	Thu	Dec	0.012397

54958 rows × 12 columns

```
mask = (df['date'] > '2020-12-31') & (df['date'] <= '2021-12-31')
df2021=df.loc[mask]
df2021
```

	date	iso3c	country	income	region	continent	dcases	ddeaths	population	weekdays	month	fatality
312	2021-01-01	AFG	Afghanistan	Low income	South Asia	Asia	183	12	38041754	Fri	Jan	0.065574
313	2021-01-02	AFG	Afghanistan	Low income	South Asia	Asia	73	10	38041754	Sat	Jan	0.136986
314	2021-01-03	AFG	Afghanistan	Low income	South Asia	Asia	123	10	38041754	Sun	Jan	0.081301
315	2021-01-04	AFG	Afghanistan	Low income	South Asia	Asia	200	9	38041754	Mon	Jan	0.045000
316	2021-01-05	AFG	Afghanistan	Low income	South Asia	Asia	102	7	38041754	Tue	Jan	0.068627
...	...	...	...	...	...	...	...	...	...	...	...	...
122838	2021-12-27	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	1098	17	14645468	Mon	Dec	0.015483
122839	2021-12-28	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	2099	32	14645468	Tue	Dec	0.015245
122840	2021-12-29	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	0	0	14645468	Wed	Dec	0.000000
122841	2021-12-30	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	4180	57	14645468	Thu	Dec	0.013636
122842	2021-12-31	ZWE	Zimbabwe	Lower middle income	Sub-Saharan Africa	Africa	1530	7	14645468	Fri	Dec	0.004575

67885 rows × 12 columns

In this part, im going to analyze the following:the probanbility of deaths in 2020 in the 7 different regions

```
df2020['dcases'].sum()
```

```
83839670
```

```
df2020['ddeaths'].sum()
```

```
1883714
```

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

```
from pandas.api.types import CategoricalDtype
cats=['South Asia', 'Sub-Saharan Africa', 'Europe & Central Asia','Middle East & North Africa','Latin America & Caribbean','East Asia 
cat_type = CategoricalDtype(categories=cats, ordered=True)
df['region'] = df['region'].astype(cat_type)
```

```
stats=df2020.groupby("region").agg({"ddeaths": [np.mean, np.std, np.size]})
```

```
stats
```

	ddeaths		
	mean	std	size
region			
East Asia & Pacific	7.110776	27.950752	6301
Europe & Central Asia	36.913612	111.023170	15743
Latin America & Caribbean	57.772410	181.529167	9750
Middle East & North Africa	17.696025	47.867778	6415
North America(region)	534.744557	759.732535	689
South Asia	67.306966	204.340594	2541
Sub-Saharan Africa	3.070567	20.413941	13519

```
ci95_hi = []
ci95_lo = []
```

```
for i in stats.index:
    m, s, n = stats.loc[i]
    x=scipy.stats.t.interval(.95, n-1, m,s/np.sqrt(n-1))
    ci95_hi.append(x[1])
    ci95_lo.append(x[0])
```

```
stats['ci95_hi'] = ci95_hi
stats['ci95_lo'] = ci95_lo
print(stats)
```

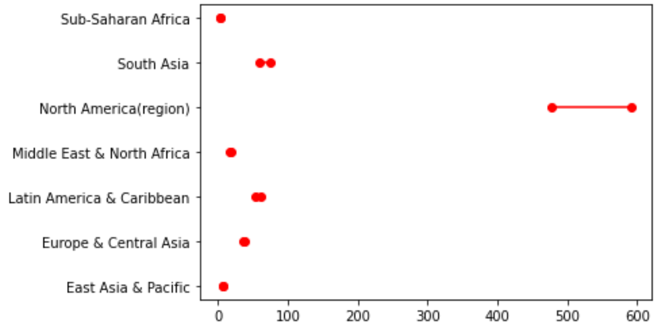
```
                               ddeaths                          ci95_hi  \
                                  mean         std   size                 
region                                                                     
East Asia & Pacific           7.110776   27.950752   6301    7.801103    
Europe & Central Asia        36.913612  111.023170  15743   38.648076    
Latin America & Caribbean    57.772410  181.529167   9750   61.376274    
Middle East & North Africa   17.696025   47.867778   6415   18.867704    
North America(region)       534.744557  759.732535    689  591.614040    
South Asia                   67.306966  204.340594   2541   75.257437    
Sub-Saharan Africa            3.070567   20.413941  13519    3.414725    

                               ci95_lo  
                                        
region                                  
East Asia & Pacific           6.420449  
Europe & Central Asia        35.179149  
Latin America & Caribbean    54.168546  
Middle East & North Africa   16.524345  
North America(region)       477.875074  
South Asia                   59.356494  
Sub-Saharan Africa            2.726410  
```

```
df_ci= pd.DataFrame(stats)
df_ci['region']=df_ci.index
```

```
for lb,ub,y in zip(df_ci['ci95_lo'],df_ci['ci95_hi'],range(len(df_ci))):
    plt.plot((lb,ub),(y,y),'ro-')
plt.yticks(range(len(df_ci)),list(df_ci['region']))
```

```
([<matplotlib.axis.YTick at 0x7fef06c879a0>,
  <matplotlib.axis.YTick at 0x7fef06c81e20>,
  <matplotlib.axis.YTick at 0x7fef06c812b0>,
  <matplotlib.axis.YTick at 0x7fef06c3e580>,
  <matplotlib.axis.YTick at 0x7fef06c3ecd0>,
  <matplotlib.axis.YTick at 0x7fef06c3e940>,
  <matplotlib.axis.YTick at 0x7fef06c46610>],
 [Text(0, 0, 'East Asia & Pacific'),
  Text(0, 1, 'Europe & Central Asia'),
  Text(0, 2, 'Latin America & Caribbean'),
  Text(0, 3, 'Middle East & North Africa'),
  Text(0, 4, 'North America(region)'),
  Text(0, 5, 'South Asia'),
  Text(0, 6, 'Sub-Saharan Africa')])
```



this shows that north America had the highest probability of number of deaths among the regions in 2020. this is probably because of the elections period in north america that occurred in 2020

In this part, im going to analyze the following: the probability of cases in 2021 regarding the income level

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

```
from pandas.api.types import CategoricalDtype
cats=['Low income', 'Lower middle income', 'Upper middle income','High income']
cat_type = CategoricalDtype(categories=cats, ordered=True)
df['income'] = df['income'].astype(cat_type)
```

```
stats=df2021.groupby("income").agg({"dcases": [np.mean, np.std, np.size]})
stats
ci95_hi = []
ci95_lo = []
```

```
ci95_hi = []
ci95_lo = []
for i in stats.index:
    m, s, n = stats.loc[i]
    x=scipy.stats.t.interval(.95, n-1, m,s/np.sqrt(n-1))
    ci95_hi.append(x[1])
    ci95_lo.append(x[0])
stats['ci95_hi'] = ci95_hi
stats['ci95_lo'] = ci95_lo
print(stats)
```

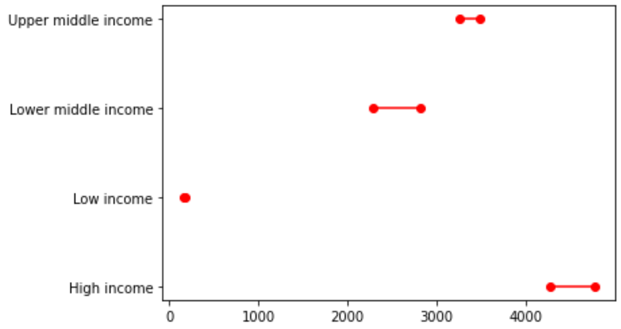
```
                          dcases                              ci95_hi  \
                            mean           std   size                  
income                                                                 
High income          4520.954339  18317.784729  20937  4769.095778   
Low income            168.681058    657.105911  10585   181.201170   
Lower middle income  2550.449589  17762.604302  16653  2820.256518   
Upper middle income  3368.106393   8585.182602  19710  3487.971286   

                         ci95_lo  
                                  
income                            
High income          4272.812901  
Low income            156.160946  
Lower middle income  2280.642659  
Upper middle income  3248.241499  
```

```
df_ci= pd.DataFrame(stats)
df_ci['income']=df_ci.index
```

```
for lb,ub,y in zip(df_ci['ci95_lo'],df_ci['ci95_hi'],range(len(df_ci))):
    plt.plot((lb,ub),(y,y),'ro-')
plt.yticks(range(len(df_ci)),list(df_ci['income']))
```

```
([<matplotlib.axis.YTick at 0x7fef06c054f0>,
  <matplotlib.axis.YTick at 0x7fef06bfdd00>,
  <matplotlib.axis.YTick at 0x7fef06bfd220>,
  <matplotlib.axis.YTick at 0x7fef06c2f790>],
 [Text(0, 0, 'High income'),
  Text(0, 1, 'Low income'),
  Text(0, 2, 'Lower middle income'),
  Text(0, 3, 'Upper middle income')])
```



it shows that the higher the income, the higher the probability of cases in 2021. this can be interpreted as the higher the income, the more often the person travels, and travelling at that time was dangerous because you could get Covid.

In this part, im going to analyze the following: the probability of deaths in different continents

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

```
from pandas.api.types import CategoricalDtype
cats=['Asia', 'Africa', 'Europe','South America(continent)', 'North America(continent)', 'Oceania']
cat_type = CategoricalDtype(categories=cats, ordered=True)
df['continent'] = df['continent'].astype(cat_type)
```

```
stats=df.groupby("continent").agg({"ddeaths": [np.mean, np.std, np.size]})
stats
```

	ddeaths		
	mean	std	size
continent			
Asia	40.516863	192.130158	31103.0
Africa	6.588286	31.043424	34677.0
Europe	52.627530	140.575791	29103.0
South America(continent)	NaN	NaN	NaN
North America(continent)	NaN	NaN	NaN
Oceania	0.758323	3.094773	4746.0

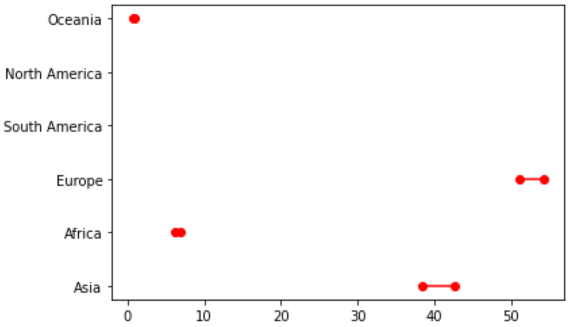
```
ci95_hi = []
ci95_lo = []
for i in stats.index:
    m, s, n = stats.loc[i]
    x=scipy.stats.t.interval(.95, n-1, m,s/np.sqrt(n-1))
    ci95_hi.append(x[1])
    ci95_lo.append(x[0])
stats['ci95_hi'] = ci95_hi
stats['ci95_lo'] = ci95_lo
print(stats)
```

```
                            ddeaths                          ci95_hi    ci95_lo
                               mean         std     size
continent
Asia                      40.516863  192.130158  31103.0  42.652200  38.381527
Africa                     6.588286   31.043424  34677.0   6.915038   6.261534
Europe                    52.627530  140.575791  29103.0  54.242689  51.012371
South America(continent)        NaN         NaN      NaN        NaN        NaN
North America(continent)        NaN         NaN      NaN        NaN        NaN
Oceania                    0.758323    3.094773   4746.0   0.846401   0.670244
```

```
df_ci= pd.DataFrame(stats)
df_ci['continent']=df_ci.index
```

```
for lb,ub,y in zip(df_ci['ci95_lo'],df_ci['ci95_hi'],range(len(df_ci))):
    plt.plot((lb,ub),(y,y),'ro-')
plt.yticks(range(len(df_ci)),list(df_ci['continent']))
```

```
([<matplotlib.axis.YTick at 0x7feefc26e730>,
  <matplotlib.axis.YTick at 0x7feefc268f70>,
  <matplotlib.axis.YTick at 0x7feefc268070>,
  <matplotlib.axis.YTick at 0x7feefc2250a0>,
  <matplotlib.axis.YTick at 0x7feefc225730>,
  <matplotlib.axis.YTick at 0x7feefc225c10>],
 [Text(0, 0, 'Asia'),
  Text(0, 1, 'Africa'),
  Text(0, 2, 'Europe'),
  Text(0, 3, 'South America'),
  Text(0, 4, 'North America'),
  Text(0, 5, 'Oceania')])
```



Europe had the highest probability of deaths, this can be because of the high number of old-aged people in Europe which affects the mortality rate of Covid 19