In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [15]:
```
#import csv from sheet 1
df1 = pd.read_csv("C:\\Users\\taylo\\Desktop\\GA\\DSI_Assignments_NEW\\SEMRush
\\SQL1.csv")
```

In [16]:
```
#checks out dataframe
df1
```

Out[16]:

	name	country	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	Karen	Armenia	NaN	NaN	NaN
1	Alexandra	Australia	NaN	NaN	NaN
2	Aman	Australia	NaN	NaN	NaN
3	Bill	Australia	NaN	NaN	NaN
4	Brandon	Australia	NaN	NaN	NaN
5	Emily	Australia	NaN	NaN	NaN
6	Jay	Australia	NaN	NaN	NaN
7	Kate	Australia	NaN	NaN	NaN
8	Rebecca	Australia	NaN	NaN	NaN
9	Rick	Australia	NaN	NaN	NaN
10	Sara	Australia	NaN	NaN	NaN
11	Sumit	Australia	NaN	NaN	NaN
12	Christian	Austria	NaN	NaN	NaN
13	Nicole	Austria	NaN	NaN	NaN
14	Olga	Austria	NaN	NaN	NaN
15	Oliver	Austria	NaN	NaN	NaN
16	Victor	Austria	NaN	NaN	NaN
17	Maria	Belarus	NaN	NaN	NaN
18	Alessandro	Belgium	NaN	NaN	NaN
19	Barbara	Belgium	NaN	NaN	NaN
20	Kim	Belgium	NaN	NaN	NaN
21	Manuel	Belgium	NaN	NaN	NaN
22	Matt	Belgium	NaN	NaN	NaN
23	Peter	Belgium	NaN	NaN	NaN
24	Tim	Belgium	NaN	NaN	NaN
25	Daniela	Brazil	NaN	NaN	NaN
26	Denis	Brazil	NaN	NaN	NaN
27	Lucas	Brazil	NaN	NaN	NaN
28	Luiz	Brazil	NaN	NaN	NaN
29	Thiago	Brazil	NaN	NaN	NaN
...	...	...	...	...	...
962	NaN	NaN	NaN	NaN	NaN
963	NaN	NaN	NaN	NaN	NaN
964	NaN	NaN	NaN	NaN	NaN
965	NaN	NaN	NaN	NaN	NaN

	name	country	Unnamed: 2	Unnamed: 3	Unnamed: 4
966	NaN	NaN	NaN	NaN	NaN
967	NaN	NaN	NaN	NaN	NaN
968	NaN	NaN	NaN	NaN	NaN
969	NaN	NaN	NaN	NaN	NaN
970	NaN	NaN	NaN	NaN	NaN
971	NaN	NaN	NaN	NaN	NaN
972	NaN	NaN	NaN	NaN	NaN
973	NaN	NaN	NaN	NaN	NaN
974	NaN	NaN	NaN	NaN	NaN
975	NaN	NaN	NaN	NaN	NaN
976	NaN	NaN	NaN	NaN	NaN
977	NaN	NaN	NaN	NaN	NaN
978	NaN	NaN	NaN	NaN	NaN
979	NaN	NaN	NaN	NaN	NaN
980	NaN	NaN	NaN	NaN	NaN
981	NaN	NaN	NaN	NaN	NaN
982	NaN	NaN	NaN	NaN	NaN
983	NaN	NaN	NaN	NaN	NaN
984	NaN	NaN	NaN	NaN	NaN
985	NaN	NaN	NaN	NaN	NaN
986	NaN	NaN	NaN	NaN	NaN
987	NaN	NaN	NaN	NaN	NaN
988	NaN	NaN	NaN	NaN	NaN
989	NaN	NaN	NaN	NaN	NaN
990	NaN	NaN	NaN	NaN	NaN
991	NaN	NaN	NaN	NaN	NaN

992 rows × 5 columns

In [12]:
```
#checks nulls in dataframe
df1.isna
```

Out[12]:
```
<bound method DataFrame.isna of            name    country  Unnamed: 2  Unnam
ed: 3  Unnamed: 4
0         Karen    Armenia         NaN         NaN         NaN
1     Alexandra  Australia         NaN         NaN         NaN
2          Aman  Australia         NaN         NaN         NaN
3          Bill  Australia         NaN         NaN         NaN
4       Brandon  Australia         NaN         NaN         NaN
5         Emily  Australia         NaN         NaN         NaN
6           Jay  Australia         NaN         NaN         NaN
7          Kate  Australia         NaN         NaN         NaN
8       Rebecca  Australia         NaN         NaN         NaN
9          Rick  Australia         NaN         NaN         NaN
10         Sara  Australia         NaN         NaN         NaN
11        Sumit  Australia         NaN         NaN         NaN
12    Christian    Austria         NaN         NaN         NaN
13       Nicole    Austria         NaN         NaN         NaN
14         Olga    Austria         NaN         NaN         NaN
15       Oliver    Austria         NaN         NaN         NaN
16       Victor    Austria         NaN         NaN         NaN
17        Maria    Belarus         NaN         NaN         NaN
18   Alessandro    Belgium         NaN         NaN         NaN
19      Barbara    Belgium         NaN         NaN         NaN
20          Kim    Belgium         NaN         NaN         NaN
21       Manuel    Belgium         NaN         NaN         NaN
22         Matt    Belgium         NaN         NaN         NaN
23        Peter    Belgium         NaN         NaN         NaN
24          Tim    Belgium         NaN         NaN         NaN
25      Daniela     Brazil         NaN         NaN         NaN
26        Denis     Brazil         NaN         NaN         NaN
27        Lucas     Brazil         NaN         NaN         NaN
28         Luiz     Brazil         NaN         NaN         NaN
29       Thiago     Brazil         NaN         NaN         NaN
..          ...        ...         ...         ...         ...
962         NaN        NaN         NaN         NaN         NaN
963         NaN        NaN         NaN         NaN         NaN
964         NaN        NaN         NaN         NaN         NaN
965         NaN        NaN         NaN         NaN         NaN
966         NaN        NaN         NaN         NaN         NaN
967         NaN        NaN         NaN         NaN         NaN
968         NaN        NaN         NaN         NaN         NaN
969         NaN        NaN         NaN         NaN         NaN
970         NaN        NaN         NaN         NaN         NaN
971         NaN        NaN         NaN         NaN         NaN
972         NaN        NaN         NaN         NaN         NaN
973         NaN        NaN         NaN         NaN         NaN
974         NaN        NaN         NaN         NaN         NaN
975         NaN        NaN         NaN         NaN         NaN
976         NaN        NaN         NaN         NaN         NaN
977         NaN        NaN         NaN         NaN         NaN
978         NaN        NaN         NaN         NaN         NaN
979         NaN        NaN         NaN         NaN         NaN
980         NaN        NaN         NaN         NaN         NaN
981         NaN        NaN         NaN         NaN         NaN
982         NaN        NaN         NaN         NaN         NaN
983         NaN        NaN         NaN         NaN         NaN
984         NaN        NaN         NaN         NaN         NaN
985         NaN        NaN         NaN         NaN         NaN
```

```
986         NaN         NaN         NaN         NaN         NaN
987         NaN         NaN         NaN         NaN         NaN
988         NaN         NaN         NaN         NaN         NaN
989         NaN         NaN         NaN         NaN         NaN
990         NaN         NaN         NaN         NaN         NaN
991         NaN         NaN         NaN         NaN         NaN

[992 rows x 5 columns]>
```

In [17]:
```
#checks columns of df
df1.columns
```

Out[17]:
```
Index(['name', 'country', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='o
bject')
```

In [25]:
```
#drops unnamed columns
df1 = df1.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1)
```

In [26]:
```
#drops unused rows
df1[:267]
```

Out[26]:

	name	country
0	Karen	Armenia
1	Alexandra	Australia
2	Aman	Australia
3	Bill	Australia
4	Brandon	Australia
5	Emily	Australia
6	Jay	Australia
7	Kate	Australia
8	Rebecca	Australia
9	Rick	Australia
10	Sara	Australia
11	Sumit	Australia
12	Christian	Austria
13	Nicole	Austria
14	Olga	Austria
15	Oliver	Austria
16	Victor	Austria
17	Maria	Belarus
18	Alessandro	Belgium
19	Barbara	Belgium
20	Kim	Belgium
21	Manuel	Belgium
22	Matt	Belgium
23	Peter	Belgium
24	Tim	Belgium
25	Daniela	Brazil
26	Denis	Brazil
27	Lucas	Brazil
28	Luiz	Brazil
29	Thiago	Brazil
...	...	...
237	Samuel	United Kingdom
238	Vladimir	United Kingdom
239	Aaron	United States
240	Aditya	United States

	name	country
241	Akash	United States
242	Alexander	United States
243	Amanda	United States
244	Andy	United States
245	Anthony	United States
246	Danielle	United States
247	Deepak	United States
248	Elena	United States
249	Francisco	United States
250	Guilherme	United States
251	Jim	United States
252	John	United States
253	Julie	United States
254	Justin	United States
255	Md	United States
256	Mohit	United States
257	Monica	United States
258	Nathan	United States
259	Neha	United States
260	Robert	United States
261	Roman	United States
262	Sachin	United States
263	Vanessa	United States
264	Vincent	United States
265	Leonardo	Uruguay
266	Nguyen	Viet Nam

267 rows × 2 columns

In [10]:
```
#imports csv from sheet 2 and creates second df
df2 = pd.read_csv("C:\\Users\\taylo\\Desktop\\GA\\DSI_Assignments_NEW\\SEMRush
\\SQL2.csv")
```

In [11]:
```
#checks out 2nd df
df2
```

Out[11]:

	name	amount
0	Aaron	$17,200
1	Abdul	$300
2	Abhishek	$4,200
3	Adam	$30,800
4	Aditya	$500
5	Adrian	$4,100
6	Ahmed	$1,600
7	Ajay	$300
8	Akash	$700
9	Akshay	$300
10	Alan	$4,700
11	Alberto	$6,700
12	Alejandro	$1,700
13	Alessandro	$2,700
14	Alex	$17,200
15	Alexander	$8,300
16	Alexandra	$700
17	Alexandre	$4,400
18	Ali	$4,500
19	Aman	$300
20	Amanda	$1,300
21	Amit	$1,100
22	Amy	$17,100
23	Ana	$2,100
24	Anand	$300
25	Andre	$1,600
26	Andrea	$12,100
27	Andreas	$6,400
28	Andres	$700
29	Andrew	$21,500
...	...	...
969	NaN	NaN
970	NaN	NaN
971	NaN	NaN
972	NaN	NaN

	name	amount
973	NaN	NaN
974	NaN	NaN
975	NaN	NaN
976	NaN	NaN
977	NaN	NaN
978	NaN	NaN
979	NaN	NaN
980	NaN	NaN
981	NaN	NaN
982	NaN	NaN
983	NaN	NaN
984	NaN	NaN
985	NaN	NaN
986	NaN	NaN
987	NaN	NaN
988	NaN	NaN
989	NaN	NaN
990	NaN	NaN
991	NaN	NaN
992	NaN	NaN
993	NaN	NaN
994	NaN	NaN
995	NaN	NaN
996	NaN	NaN
997	NaN	NaN
998	NaN	NaN

999 rows × 2 columns

In [28]:
```
#gets ride of nan rows in df2
df2[:267]
```

Out[28]:

	name	amount
0	Aaron	$17,200
1	Abdul	$300
2	Abhishek	$4,200
3	Adam	$30,800
4	Aditya	$500
5	Adrian	$4,100
6	Ahmed	$1,600
7	Ajay	$300
8	Akash	$700
9	Akshay	$300
10	Alan	$4,700
11	Alberto	$6,700
12	Alejandro	$1,700
13	Alessandro	$2,700
14	Alex	$17,200
15	Alexander	$8,300
16	Alexandra	$700
17	Alexandre	$4,400
18	Ali	$4,500
19	Aman	$300
20	Amanda	$1,300
21	Amit	$1,100
22	Amy	$17,100
23	Ana	$2,100
24	Anand	$300
25	Andre	$1,600
26	Andrea	$12,100
27	Andreas	$6,400
28	Andres	$700
29	Andrew	$21,500
...	...	...
237	Sergey	$500
238	Sergio	$2,600
239	Shubham	$400
240	Silvia	$200

	name	amount
241	Simon	$14,900
242	Stefan	$2,400
243	Stephanie	$7,200
244	Stephen	$13,700
245	Steve	$10,900
246	Steven	$8,500
247	Sumit	$400
248	Sunil	$400
249	Susan	$13,200
250	Syed	$1,700
251	Team	$2,200
252	Thiago	$5,400
253	Thomas	$9,500
254	Tim	$5,500
255	Tom	$11,900
256	Tony	$18,300
257	Vanessa	$1,400
258	Victor	$3,500
259	Victoria	$1,900
260	Vijay	$1,000
261	Vikas	$600
262	Vincent	$3,800
263	Vishal	$200
264	Vivek	$200
265	Vladimir	$500
266	William	$15,000

267 rows × 2 columns

In [29]:
```
#imports csv from sheet 3 and creates 3rd df
df3 = pd.read_csv("C:\\Users\\taylo\\Desktop\\GA\\DSI_Assignments_NEW\\SEMRush
\\SQL3.csv")
```

In [31]:
```
#checks out 3rd df
df3.head()
```

Out[31]:

	userId	time	Action type	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unr
0	17693.0	1.580680e+09	open page B	NaN	NaN	NaN	NaN	NaN	
1	21048.0	1.580693e+09	open page D	NaN	NaN	NaN	NaN	NaN	
2	21048.0	1.580693e+09	open page C	NaN	NaN	NaN	NaN	NaN	
3	21048.0	1.580693e+09	click on banner	NaN	NaN	NaN	NaN	NaN	
4	326102.0	1.580689e+09	search custom info	NaN	NaN	NaN	NaN	NaN	

5 rows × 26 columns

In [32]:
```
#checks out columns of 3rd df
df3.columns
```

Out[32]:
```
Index(['userId', 'time', 'Action type', 'Unnamed: 3', 'Unnamed: 4',
       'Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9',
       'Unnamed: 10', 'Unnamed: 11', 'Unnamed: 12', 'Unnamed: 13',
       'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 16', 'Unnamed: 17',
       'Unnamed: 18', 'Unnamed: 19', 'Unnamed: 20', 'Unnamed: 21',
       'Unnamed: 22', 'Unnamed: 23', 'Unnamed: 24', 'Unnamed: 25'],
      dtype='object')
```

In [34]:
```
#drops unnamed columns from 3rd df
df3 = df3.drop(['Unnamed: 3', 'Unnamed: 4', 'Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7','Unnamed: 8','Unnamed: 9','Unnamed: 10','Unnamed: 11','Unnamed: 12','Unnamed: 13','Unnamed: 14','Unnamed: 15','Unnamed: 16','Unnamed: 17','Unnamed: 18','Unnamed: 19','Unnamed: 20','Unnamed: 21','Unnamed: 22','Unnamed: 23','Unnamed: 24','Unnamed: 25', ], axis=1)
```

In [36]:
```
#confirms unnamed columns deleted
df3.columns
```

Out[36]:
```
Index(['userId', 'time', 'Action type'], dtype='object')
```

In [38]:
```
#drops nan rows from df
df3[:769]
```

Out[38]:

	userId	time	Action type
0	17693.0	1.580680e+09	open page B
1	21048.0	1.580693e+09	open page D
2	21048.0	1.580693e+09	open page C
3	21048.0	1.580693e+09	click on banner
4	326102.0	1.580689e+09	search custom info
5	415404.0	1.580635e+09	make a report
6	415404.0	1.580635e+09	open page C
7	415404.0	1.580635e+09	make a report
8	415404.0	1.580635e+09	log out
9	415404.0	1.580635e+09	make a report
10	415404.0	1.580635e+09	open email
11	807375.0	1.580645e+09	open email
12	1060722.0	1.580638e+09	open email
13	1060722.0	1.580638e+09	open page C
14	1060722.0	1.580643e+09	open page C
15	1060722.0	1.580638e+09	open page D
16	1060722.0	1.580696e+09	sign in
17	1060722.0	1.580638e+09	sign in
18	1060722.0	1.580696e+09	make a report
19	1060722.0	1.580696e+09	make a report
20	1060722.0	1.580696e+09	open email
21	1060722.0	1.580696e+09	open page B
22	1060722.0	1.580696e+09	make a report
23	1060722.0	1.580696e+09	sign in
24	1060722.0	1.580696e+09	open page B
25	1060722.0	1.580696e+09	open email
26	1060722.0	1.580696e+09	click on banner
27	1060722.0	1.580696e+09	open page D
28	1060722.0	1.580696e+09	open page B
29	1060722.0	1.580696e+09	click on banner
...	...	...	...
739	8301584.0	1.580654e+09	click on banner
740	8301584.0	1.580654e+09	open page A
741	8301584.0	1.580654e+09	open page C
742	8301584.0	1.580654e+09	open page A

	userId	time	Action type
743	8301584.0	1.580654e+09	open email
744	8301584.0	1.580654e+09	make a report
745	8301584.0	1.580654e+09	open page B
746	8301584.0	1.580654e+09	log out
747	8301584.0	1.580654e+09	search custom info
748	8301584.0	1.580654e+09	open page B
749	8301584.0	1.580656e+09	open page A
750	8301584.0	1.580656e+09	open page C
751	8301584.0	1.580667e+09	open page C
752	8301584.0	1.580656e+09	open email
753	8301584.0	1.580667e+09	open page C
754	8301584.0	1.580667e+09	make a report
755	8301584.0	1.580654e+09	click on banner
756	8301584.0	1.580654e+09	open page A
757	8301584.0	1.580654e+09	open page A
758	8301584.0	1.580667e+09	open page C
759	8301584.0	1.580667e+09	open page A
760	8301584.0	1.580667e+09	open page C
761	8301584.0	1.580656e+09	search custom info
762	8301584.0	1.580667e+09	sign in
763	8301584.0	1.580667e+09	click on banner
764	8301584.0	1.580667e+09	click on banner
765	8301584.0	1.580655e+09	open page A
766	NaN	1.580655e+09	open page D
767	8301584.0	1.580654e+09	open page D
768	8301584.0	1.580667e+09	open page C

769 rows × 3 columns

In [88]:
```
#creates merged df for task 1
dftask1 = df1.merge(df2, left_on='name', right_on='name')
```

In [89]:
```
#confirms merged df was created
dftask1
```

Out[89]:

	name	country	amount
0	Karen	Armenia	$9,900
1	Alexandra	Australia	$700
2	Aman	Australia	$300
3	Bill	Australia	$4,700
4	Brandon	Australia	$6,000
5	Emily	Australia	$3,800
6	Jay	Australia	$3,000
7	Kate	Australia	$1,100
8	Rebecca	Australia	$6,400
9	Rick	Australia	$3,200
10	Sara	Australia	$1,100
11	Sumit	Australia	$400
12	Christian	Austria	$8,900
13	Nicole	Austria	$7,500
14	Olga	Austria	$2,500
15	Oliver	Austria	$2,600
16	Victor	Austria	$3,500
17	Maria	Belarus	$2,200
18	Alessandro	Belgium	$2,700
19	Barbara	Belgium	$700
20	Kim	Belgium	$6,600
21	Manuel	Belgium	$1,400
22	Matt	Belgium	$8,900
23	Peter	Belgium	$20,300
24	Tim	Belgium	$5,500
25	Daniela	Brazil	$3,100
26	Denis	Brazil	$1,400
27	Lucas	Brazil	$2,900
28	Luiz	Brazil	$500
29	Thiago	Brazil	$5,400
...	...	...	...
530937	NaN	NaN	NaN
530938	NaN	NaN	NaN
530939	NaN	NaN	NaN
530940	NaN	NaN	NaN

	name	country	amount
530941	NaN	NaN	NaN
530942	NaN	NaN	NaN
530943	NaN	NaN	NaN
530944	NaN	NaN	NaN
530945	NaN	NaN	NaN
530946	NaN	NaN	NaN
530947	NaN	NaN	NaN
530948	NaN	NaN	NaN
530949	NaN	NaN	NaN
530950	NaN	NaN	NaN
530951	NaN	NaN	NaN
530952	NaN	NaN	NaN
530953	NaN	NaN	NaN
530954	NaN	NaN	NaN
530955	NaN	NaN	NaN
530956	NaN	NaN	NaN
530957	NaN	NaN	NaN
530958	NaN	NaN	NaN
530959	NaN	NaN	NaN
530960	NaN	NaN	NaN
530961	NaN	NaN	NaN
530962	NaN	NaN	NaN
530963	NaN	NaN	NaN
530964	NaN	NaN	NaN
530965	NaN	NaN	NaN
530966	NaN	NaN	NaN

530967 rows × 3 columns

In [58]:
```
#checks out data types in merged df
dftask1.dtypes
```

Out[58]:
```
name       object
country    object
amount     object
dtype: object
```

In [90]:
```
#gets rid of nan rows in task1df
dftask1[:267]
```

Out[90]:

	name	country	amount
0	Karen	Armenia	$9,900
1	Alexandra	Australia	$700
2	Aman	Australia	$300
3	Bill	Australia	$4,700
4	Brandon	Australia	$6,000
5	Emily	Australia	$3,800
6	Jay	Australia	$3,000
7	Kate	Australia	$1,100
8	Rebecca	Australia	$6,400
9	Rick	Australia	$3,200
10	Sara	Australia	$1,100
11	Sumit	Australia	$400
12	Christian	Austria	$8,900
13	Nicole	Austria	$7,500
14	Olga	Austria	$2,500
15	Oliver	Austria	$2,600
16	Victor	Austria	$3,500
17	Maria	Belarus	$2,200
18	Alessandro	Belgium	$2,700
19	Barbara	Belgium	$700
20	Kim	Belgium	$6,600
21	Manuel	Belgium	$1,400
22	Matt	Belgium	$8,900
23	Peter	Belgium	$20,300
24	Tim	Belgium	$5,500
25	Daniela	Brazil	$3,100
26	Denis	Brazil	$1,400
27	Lucas	Brazil	$2,900
28	Luiz	Brazil	$500
29	Thiago	Brazil	$5,400
...	...	...	...
237	Samuel	United Kingdom	$2,300
238	Vladimir	United Kingdom	$500
239	Aaron	United States	$17,200
240	Aditya	United States	$500

	name	country	amount
241	Akash	United States	$700
242	Alexander	United States	$8,300
243	Amanda	United States	$1,300
244	Andy	United States	$4,200
245	Anthony	United States	$8,200
246	Danielle	United States	$1,500
247	Deepak	United States	$700
248	Elena	United States	$16,900
249	Francisco	United States	$3,400
250	Guilherme	United States	$500
251	Jim	United States	$11,500
252	John	United States	$46,100
253	Julie	United States	$3,500
254	Justin	United States	$12,900
255	Md	United States	$100
256	Mohit	United States	$1,300
257	Monica	United States	$4,300
258	Nathan	United States	$3,400
259	Neha	United States	$1,000
260	Robert	United States	$24,500
261	Roman	United States	$2,100
262	Sachin	United States	$100
263	Vanessa	United States	$1,400
264	Vincent	United States	$3,800
265	Leonardo	Uruguay	$400
266	Nguyen	Viet Nam	$100

267 rows × 3 columns

In [91]:
```
#eliminates $ in task1df
dftask1['amount'] = dftask1['amount'].astype(str).str.replace('$', '')
```

In [92]:
```
#eliminates , in task1df
dftask1['amount'] = dftask1['amount'].astype(str).str.replace(',', '')
```

In [93]:
```
#eliminates nan in task1df
dftask1['amount'] = dftask1['amount'].astype(str).str.replace('nan', '-999')
```

In [94]:
```
#gets rid of unnamed rows in task1df
dftask1[:267]
```

Out[94]:

	name	country	amount
0	Karen	Armenia	9900
1	Alexandra	Australia	700
2	Aman	Australia	300
3	Bill	Australia	4700
4	Brandon	Australia	6000
5	Emily	Australia	3800
6	Jay	Australia	3000
7	Kate	Australia	1100
8	Rebecca	Australia	6400
9	Rick	Australia	3200
10	Sara	Australia	1100
11	Sumit	Australia	400
12	Christian	Austria	8900
13	Nicole	Austria	7500
14	Olga	Austria	2500
15	Oliver	Austria	2600
16	Victor	Austria	3500
17	Maria	Belarus	2200
18	Alessandro	Belgium	2700
19	Barbara	Belgium	700
20	Kim	Belgium	6600
21	Manuel	Belgium	1400
22	Matt	Belgium	8900
23	Peter	Belgium	20300
24	Tim	Belgium	5500
25	Daniela	Brazil	3100
26	Denis	Brazil	1400
27	Lucas	Brazil	2900
28	Luiz	Brazil	500
29	Thiago	Brazil	5400
...	...	...	...
237	Samuel	United Kingdom	2300
238	Vladimir	United Kingdom	500
239	Aaron	United States	17200
240	Aditya	United States	500

	name	country	amount
241	Akash	United States	700
242	Alexander	United States	8300
243	Amanda	United States	1300
244	Andy	United States	4200
245	Anthony	United States	8200
246	Danielle	United States	1500
247	Deepak	United States	700
248	Elena	United States	16900
249	Francisco	United States	3400
250	Guilherme	United States	500
251	Jim	United States	11500
252	John	United States	46100
253	Julie	United States	3500
254	Justin	United States	12900
255	Md	United States	100
256	Mohit	United States	1300
257	Monica	United States	4300
258	Nathan	United States	3400
259	Neha	United States	1000
260	Robert	United States	24500
261	Roman	United States	2100
262	Sachin	United States	100
263	Vanessa	United States	1400
264	Vincent	United States	3800
265	Leonardo	Uruguay	400
266	Nguyen	Viet Nam	100

267 rows × 3 columns

In [96]:
```
#changes amount in dftask1 into integer
dftask1["amount"] = dftask1["amount"].astype(int)
print(dftask1.dtypes)
```

```
name       object
country    object
amount      int32
dtype: object
```

In [98]:
```
#selects name, country and amount in dftask1 for amounts over 100,000 - there
 are NONE
select_100k = dftask1.loc[dftask1['amount'] >= 100000, ]
print (select_100k)
```

```
Empty DataFrame
Columns: [name, country, amount]
Index: []
```

In [122]:
```
#selects name, country and amount in dftask1 for amounts over 10,000 - just to
make sure it worked, it does
select_100k = dftask1.loc[dftask1['amount'] >= 10000, ]
print (select_100k)
```

	name	country	amount
23	Peter	Belgium	20300
30	William	Brazil	15000
32	Joe	Cambodia	11600
38	Eric	Canada	30500
41	Joseph	Canada	11400
44	Kyle	Canada	10500
47	Melissa	Canada	18300
50	Scott	Canada	26100
62	Daniel	Denmark	30500
65	Martin	Denmark	12200
66	Michelle	Denmark	13200
67	Steve	Denmark	10900
69	Tom	Denmark	11900
81	George	Greece	11300
83	Adam	Hong Kong	30800
84	Amy	Hong Kong	17100
85	Christopher	Hong Kong	13100
86	Joshua	Hong Kong	10400
87	Tony	Hong Kong	18300
125	Mark	Ireland	38700
126	Neil	Ireland	14900
127	Paul	Ireland	14900
130	Sean	Ireland	14700
131	Stephen	Ireland	13700
132	Susan	Ireland	13200
142	Ian	Jamaica	12100
143	James	Japan	31700
144	Jonathan	Japan	12800
145	Kevin	Malaysia	10100
148	Matthew	Malta	14800
151	Jason	New Zealand	48400
152	Jeff	New Zealand	14100
154	Michael	New Zealand	88400
157	Richard	New Zealand	20000
173	Simon	Portugal	14900
174	Juan	Puerto Rico	10700
180	Alex	South Africa	17200
181	Andrew	South Africa	21500
182	Ryan	South Africa	38300
184	Carlos	Spain	17500
211	Andrea	Switzerland	12100
213	David	Switzerland	50900
218	Mike	Switzerland	21800
221	Brian	Thailand	25200
230	Chris	United Kingdom	14500
231	Craig	United Kingdom	11000
236	Patrick	United Kingdom	34200
239	Aaron	United States	17200
248	Elena	United States	16900
251	Jim	United States	11500
252	John	United States	46100
254	Justin	United States	12900
260	Robert	United States	24500

In [ ]:
```
#Task 2
```

In [100]:
```
#creates df for task 3
df3 = pd.read_csv("C:\\Users\\taylo\\Desktop\\GA\\DSI_Assignments_NEW\\SEMRush
\\SQL3.csv")
```

In [101]:
```
#confirms df created
df3
```

Out[101]:

	userId	time	Action type	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
0	17693.0	1.580680e+09	open page B	NaN	NaN	NaN	NaN	NaN
1	21048.0	1.580693e+09	open page D	NaN	NaN	NaN	NaN	NaN
2	21048.0	1.580693e+09	open page C	NaN	NaN	NaN	NaN	NaN
3	21048.0	1.580693e+09	click on banner	NaN	NaN	NaN	NaN	NaN
4	326102.0	1.580689e+09	search custom info	NaN	NaN	NaN	NaN	NaN
5	415404.0	1.580635e+09	make a report	NaN	NaN	NaN	NaN	NaN
6	415404.0	1.580635e+09	open page C	NaN	NaN	NaN	NaN	NaN
7	415404.0	1.580635e+09	make a report	NaN	NaN	NaN	NaN	NaN
8	415404.0	1.580635e+09	log out	NaN	NaN	NaN	NaN	NaN
9	415404.0	1.580635e+09	make a report	NaN	NaN	NaN	NaN	NaN
10	415404.0	1.580635e+09	open email	NaN	NaN	NaN	NaN	NaN
11	807375.0	1.580645e+09	open email	NaN	NaN	NaN	NaN	NaN
12	1060722.0	1.580638e+09	open email	NaN	NaN	NaN	NaN	NaN
13	1060722.0	1.580638e+09	open page C	NaN	NaN	NaN	NaN	NaN
14	1060722.0	1.580643e+09	open page C	NaN	NaN	NaN	NaN	NaN
15	1060722.0	1.580638e+09	open page D	NaN	NaN	NaN	NaN	NaN
16	1060722.0	1.580696e+09	sign in	NaN	NaN	NaN	NaN	NaN
17	1060722.0	1.580638e+09	sign in	NaN	NaN	NaN	NaN	NaN
18	1060722.0	1.580696e+09	make a report	NaN	NaN	NaN	NaN	NaN
19	1060722.0	1.580696e+09	make a report	NaN	NaN	NaN	NaN	NaN
20	1060722.0	1.580696e+09	open email	NaN	NaN	NaN	NaN	NaN

	userId	time	Action type	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
21	1060722.0	1.580696e+09	open page B	NaN	NaN	NaN	NaN	NaN
22	1060722.0	1.580696e+09	make a report	NaN	NaN	NaN	NaN	NaN
23	1060722.0	1.580696e+09	sign in	NaN	NaN	NaN	NaN	NaN
24	1060722.0	1.580696e+09	open page B	NaN	NaN	NaN	NaN	NaN
25	1060722.0	1.580696e+09	open email	NaN	NaN	NaN	NaN	NaN
26	1060722.0	1.580696e+09	click on banner	NaN	NaN	NaN	NaN	NaN
27	1060722.0	1.580696e+09	open page D	NaN	NaN	NaN	NaN	NaN
28	1060722.0	1.580696e+09	open page B	NaN	NaN	NaN	NaN	NaN
29	1060722.0	1.580696e+09	click on banner	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
751	8301584.0	1.580667e+09	open page C	NaN	NaN	NaN	NaN	NaN
752	8301584.0	1.580656e+09	open email	NaN	NaN	NaN	NaN	NaN
753	8301584.0	1.580667e+09	open page C	NaN	NaN	NaN	NaN	NaN
754	8301584.0	1.580667e+09	make a report	NaN	NaN	NaN	NaN	NaN
755	8301584.0	1.580654e+09	click on banner	NaN	NaN	NaN	NaN	NaN
756	8301584.0	1.580654e+09	open page A	NaN	NaN	NaN	NaN	NaN
757	8301584.0	1.580654e+09	open page A	NaN	NaN	NaN	NaN	NaN
758	8301584.0	1.580667e+09	open page C	NaN	NaN	NaN	NaN	NaN
759	8301584.0	1.580667e+09	open page A	NaN	NaN	NaN	NaN	NaN
760	8301584.0	1.580667e+09	open page C	NaN	NaN	NaN	NaN	NaN
761	8301584.0	1.580656e+09	search custom info	NaN	NaN	NaN	NaN	NaN
762	8301584.0	1.580667e+09	sign in	NaN	NaN	NaN	NaN	NaN

	userId	time	Action type	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
763	8301584.0	1.580667e+09	click on banner	NaN	NaN	NaN	NaN	NaN
764	8301584.0	1.580667e+09	click on banner	NaN	NaN	NaN	NaN	NaN
765	8301584.0	1.580655e+09	open page A	NaN	NaN	NaN	NaN	NaN
766	NaN	1.580655e+09	open page D	NaN	NaN	NaN	NaN	NaN
767	8301584.0	1.580654e+09	open page D	NaN	NaN	NaN	NaN	NaN
768	8301584.0	1.580667e+09	open page C	NaN	NaN	NaN	NaN	NaN
769	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
770	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
771	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
772	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
773	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
774	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
775	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
776	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
777	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
778	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
779	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
780	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

781 rows × 26 columns

In [102]:
```
#checks out columns in df3
df3.columns
```

Out[102]:
```
Index(['userId', 'time', 'Action type', 'Unnamed: 3', 'Unnamed: 4',
       'Unnamed: 5', 'Unnamed: 6', 'Unnamed: 7', 'Unnamed: 8', 'Unnamed: 9',
       'Unnamed: 10', 'Unnamed: 11', 'Unnamed: 12', 'Unnamed: 13',
       'Unnamed: 14', 'Unnamed: 15', 'Unnamed: 16', 'Unnamed: 17',
       'Unnamed: 18', 'Unnamed: 19', 'Unnamed: 20', 'Unnamed: 21',
       'Unnamed: 22', 'Unnamed: 23', 'Unnamed: 24', 'Unnamed: 25'],
      dtype='object')
```

In [103]:
```
#drops unnamed columns from df3
df3 = df3.drop(['Unnamed: 3', 'Unnamed: 4', 'Unnamed: 5', 'Unnamed: 6', 'Unnam
ed: 7','Unnamed: 8','Unnamed: 9','Unnamed: 10','Unnamed: 11','Unnamed: 12','Un
named: 13','Unnamed: 14','Unnamed: 15','Unnamed: 16','Unnamed: 17','Unnamed: 1
8','Unnamed: 19','Unnamed: 20','Unnamed: 21','Unnamed: 22','Unnamed: 23','Unna
med: 24','Unnamed: 25', ], axis=1)
```

In [123]:
```
#confirms unnamed columns dropped
df3.columns
```

Out[123]:
```
Index(['userId', 'time', 'Action type'], dtype='object')
```

In [105]:
```
#drops unnamed rows from df3
df3[:769]
```

Out[105]:

	userId	time	Action type
0	17693.0	1.580680e+09	open page B
1	21048.0	1.580693e+09	open page D
2	21048.0	1.580693e+09	open page C
3	21048.0	1.580693e+09	click on banner
4	326102.0	1.580689e+09	search custom info
5	415404.0	1.580635e+09	make a report
6	415404.0	1.580635e+09	open page C
7	415404.0	1.580635e+09	make a report
8	415404.0	1.580635e+09	log out
9	415404.0	1.580635e+09	make a report
10	415404.0	1.580635e+09	open email
11	807375.0	1.580645e+09	open email
12	1060722.0	1.580638e+09	open email
13	1060722.0	1.580638e+09	open page C
14	1060722.0	1.580643e+09	open page C
15	1060722.0	1.580638e+09	open page D
16	1060722.0	1.580696e+09	sign in
17	1060722.0	1.580638e+09	sign in
18	1060722.0	1.580696e+09	make a report
19	1060722.0	1.580696e+09	make a report
20	1060722.0	1.580696e+09	open email
21	1060722.0	1.580696e+09	open page B
22	1060722.0	1.580696e+09	make a report
23	1060722.0	1.580696e+09	sign in
24	1060722.0	1.580696e+09	open page B
25	1060722.0	1.580696e+09	open email
26	1060722.0	1.580696e+09	click on banner
27	1060722.0	1.580696e+09	open page D
28	1060722.0	1.580696e+09	open page B
29	1060722.0	1.580696e+09	click on banner
...	...	...	...
739	8301584.0	1.580654e+09	click on banner
740	8301584.0	1.580654e+09	open page A
741	8301584.0	1.580654e+09	open page C
742	8301584.0	1.580654e+09	open page A

	userId	time	Action type
743	8301584.0	1.580654e+09	open email
744	8301584.0	1.580654e+09	make a report
745	8301584.0	1.580654e+09	open page B
746	8301584.0	1.580654e+09	log out
747	8301584.0	1.580654e+09	search custom info
748	8301584.0	1.580654e+09	open page B
749	8301584.0	1.580656e+09	open page A
750	8301584.0	1.580656e+09	open page C
751	8301584.0	1.580667e+09	open page C
752	8301584.0	1.580656e+09	open email
753	8301584.0	1.580667e+09	open page C
754	8301584.0	1.580667e+09	make a report
755	8301584.0	1.580654e+09	click on banner
756	8301584.0	1.580654e+09	open page A
757	8301584.0	1.580654e+09	open page A
758	8301584.0	1.580667e+09	open page C
759	8301584.0	1.580667e+09	open page A
760	8301584.0	1.580667e+09	open page C
761	8301584.0	1.580656e+09	search custom info
762	8301584.0	1.580667e+09	sign in
763	8301584.0	1.580667e+09	click on banner
764	8301584.0	1.580667e+09	click on banner
765	8301584.0	1.580655e+09	open page A
766	NaN	1.580655e+09	open page D
767	8301584.0	1.580654e+09	open page D
768	8301584.0	1.580667e+09	open page C

769 rows × 3 columns

Task 2 question This table T3 shows user's activity on a web page. People can do some actions on this page, also they can log in their profile. So please find: 1) Count of actions that user do being logged in. After 'sign in' action and before 'log out action'. If there is no log out - count everything untill end of the day. Please note, that user can open site on several devices/tabs etc, so you can be confuced a bit with action sequences. Time between actions - doesnt' matter. usreId SELECT COUNT(*) FROM t3 WHERE action types NOT LIKE '%%sign in%%' '%%log out%%'

In [106]:
```
#checks out df3
df3.head()
```

Out[106]:

	userId	time	Action type
0	17693.0	1.580680e+09	open page B
1	21048.0	1.580693e+09	open page D
2	21048.0	1.580693e+09	open page C
3	21048.0	1.580693e+09	click on banner
4	326102.0	1.580689e+09	search custom info

In [110]:
```
#finds counts in df3 that DO NOT contain sign in or log out - gives me idea of
df3['Action type'].str.contains('sign in|log out').value_counts()[False]
```

Out[110]: 628

Task 2 - part 2

2) Pages(!) that user opens after openning an email

In [111]:
```
df3.head()
```

Out[111]:

	userId	time	Action type
0	17693.0	1.580680e+09	open page B
1	21048.0	1.580693e+09	open page D
2	21048.0	1.580693e+09	open page C
3	21048.0	1.580693e+09	click on banner
4	326102.0	1.580689e+09	search custom info

In [112]:
```
df3.dtypes
```

Out[112]:
```
userId         float64
time           float64
Action type     object
dtype: object
```

In [113]:
```
df3
```

Out[113]:

	userId	time	Action type
0	17693.0	1.580680e+09	open page B
1	21048.0	1.580693e+09	open page D
2	21048.0	1.580693e+09	open page C
3	21048.0	1.580693e+09	click on banner
4	326102.0	1.580689e+09	search custom info
5	415404.0	1.580635e+09	make a report
6	415404.0	1.580635e+09	open page C
7	415404.0	1.580635e+09	make a report
8	415404.0	1.580635e+09	log out
9	415404.0	1.580635e+09	make a report
10	415404.0	1.580635e+09	open email
11	807375.0	1.580645e+09	open email
12	1060722.0	1.580638e+09	open email
13	1060722.0	1.580638e+09	open page C
14	1060722.0	1.580643e+09	open page C
15	1060722.0	1.580638e+09	open page D
16	1060722.0	1.580696e+09	sign in
17	1060722.0	1.580638e+09	sign in
18	1060722.0	1.580696e+09	make a report
19	1060722.0	1.580696e+09	make a report
20	1060722.0	1.580696e+09	open email
21	1060722.0	1.580696e+09	open page B
22	1060722.0	1.580696e+09	make a report
23	1060722.0	1.580696e+09	sign in
24	1060722.0	1.580696e+09	open page B
25	1060722.0	1.580696e+09	open email
26	1060722.0	1.580696e+09	click on banner
27	1060722.0	1.580696e+09	open page D
28	1060722.0	1.580696e+09	open page B
29	1060722.0	1.580696e+09	click on banner
...	...	...	...
751	8301584.0	1.580667e+09	open page C
752	8301584.0	1.580656e+09	open email
753	8301584.0	1.580667e+09	open page C
754	8301584.0	1.580667e+09	make a report

	userId	time	Action type
755	8301584.0	1.580654e+09	click on banner
756	8301584.0	1.580654e+09	open page A
757	8301584.0	1.580654e+09	open page A
758	8301584.0	1.580667e+09	open page C
759	8301584.0	1.580667e+09	open page A
760	8301584.0	1.580667e+09	open page C
761	8301584.0	1.580656e+09	search custom info
762	8301584.0	1.580667e+09	sign in
763	8301584.0	1.580667e+09	click on banner
764	8301584.0	1.580667e+09	click on banner
765	8301584.0	1.580655e+09	open page A
766	NaN	1.580655e+09	open page D
767	8301584.0	1.580654e+09	open page D
768	8301584.0	1.580667e+09	open page C
769	NaN	NaN	NaN
770	NaN	NaN	NaN
771	NaN	NaN	NaN
772	NaN	NaN	NaN
773	NaN	NaN	NaN
774	NaN	NaN	NaN
775	NaN	NaN	NaN
776	NaN	NaN	NaN
777	NaN	NaN	NaN
778	NaN	NaN	NaN
779	NaN	NaN	NaN
780	NaN	NaN	NaN

781 rows × 3 columns

In [ ]:
```
#Big picture how to code this in Python
sort by time and userid, find row nunmber for log in, log out, IF no log out,
stop when not that day print dataframe from there
```

In [125]:
```
df3[:769]
```

Out[125]:

	userId	time	Action type
0	17693.0	1.580680e+09	open page B
1	21048.0	1.580693e+09	open page D
2	21048.0	1.580693e+09	open page C
3	21048.0	1.580693e+09	click on banner
4	326102.0	1.580689e+09	search custom info
5	415404.0	1.580635e+09	make a report
6	415404.0	1.580635e+09	open page C
7	415404.0	1.580635e+09	make a report
8	415404.0	1.580635e+09	log out
9	415404.0	1.580635e+09	make a report
10	415404.0	1.580635e+09	open email
11	807375.0	1.580645e+09	open email
12	1060722.0	1.580638e+09	open email
13	1060722.0	1.580638e+09	open page C
14	1060722.0	1.580643e+09	open page C
15	1060722.0	1.580638e+09	open page D
16	1060722.0	1.580696e+09	sign in
17	1060722.0	1.580638e+09	sign in
18	1060722.0	1.580696e+09	make a report
19	1060722.0	1.580696e+09	make a report
20	1060722.0	1.580696e+09	open email
21	1060722.0	1.580696e+09	open page B
22	1060722.0	1.580696e+09	make a report
23	1060722.0	1.580696e+09	sign in
24	1060722.0	1.580696e+09	open page B
25	1060722.0	1.580696e+09	open email
26	1060722.0	1.580696e+09	click on banner
27	1060722.0	1.580696e+09	open page D
28	1060722.0	1.580696e+09	open page B
29	1060722.0	1.580696e+09	click on banner
...	...	...	...
739	8301584.0	1.580654e+09	click on banner
740	8301584.0	1.580654e+09	open page A
741	8301584.0	1.580654e+09	open page C
742	8301584.0	1.580654e+09	open page A

	userId	time	Action type
743	8301584.0	1.580654e+09	open email
744	8301584.0	1.580654e+09	make a report
745	8301584.0	1.580654e+09	open page B
746	8301584.0	1.580654e+09	log out
747	8301584.0	1.580654e+09	search custom info
748	8301584.0	1.580654e+09	open page B
749	8301584.0	1.580656e+09	open page A
750	8301584.0	1.580656e+09	open page C
751	8301584.0	1.580667e+09	open page C
752	8301584.0	1.580656e+09	open email
753	8301584.0	1.580667e+09	open page C
754	8301584.0	1.580667e+09	make a report
755	8301584.0	1.580654e+09	click on banner
756	8301584.0	1.580654e+09	open page A
757	8301584.0	1.580654e+09	open page A
758	8301584.0	1.580667e+09	open page C
759	8301584.0	1.580667e+09	open page A
760	8301584.0	1.580667e+09	open page C
761	8301584.0	1.580656e+09	search custom info
762	8301584.0	1.580667e+09	sign in
763	8301584.0	1.580667e+09	click on banner
764	8301584.0	1.580667e+09	click on banner
765	8301584.0	1.580655e+09	open page A
766	NaN	1.580655e+09	open page D
767	8301584.0	1.580654e+09	open page D
768	8301584.0	1.580667e+09	open page C

769 rows × 3 columns

In [ ]:
```
PROBABILITY TASKS
There is 0.9 chance of meeting a passing by car on a desert road in 1 hour. Wh
at is the probability of seeing a car in 15 minutes?
```

In [129]:
```
cars_per_minute = 11/50
minutes = 15

# Expected events
lam = events_per_minute * minutes

k = 1
p_k = np.exp(-lam) * np.power(lam, k) / factorial(k)
print(f'The probability of {k} cars in {minutes} minutes is {100*p_k:.2f}%.')
```

```
The probability of 1 cars in 15 minutes is 28.75%.
```

In [114]:
```
from scipy.special import factorial
```

In [116]:
```
import numpy as np
```

In [118]:
```
emails_per_hour = 1/8
hours = 24

# Expected events
lam = events_per_minute * minutes

k = 3
p_k = np.exp(-lam) * np.power(lam, k) / factorial(k)
print(f'The probability of {k} emails in {minutes} hours is {100*p_k:.2f}%.')
```

```
The probability of 3 emails in 24 hours is 22.40%.
```

In [121]:
```
emails_per_hour = 1/4
hours = 24

# Expected events
lam = events_per_minute * minutes

k = 6
p_k = np.exp(-lam) * np.power(lam, k) / factorial(k)
print(f'The probability of {k} emails in {minutes} hours is {100*p_k:.2f}%.')
```

```
The probability of 6 emails in 24 hours is 5.04%.
```