Section III - Visualizing Data

In [129]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [130]:
```
names = pd.read_csv("names_id_age.csv")
sales = pd.read_csv("lead_sale_stats.csv")
```

In [131]:
```
# dropping the NaN values
sales = sales.dropna()
```

In [132]:
```
# spliting the "lead_id" column into "lead" & "id"
sales[['lead','id']] = sales['lead_id'].str.split('_',expand=True)
```

In [133]:
```
del sales['lead_id']
```

In [134]:
```
sales.head()
```

Out[134]:

	name	bought_policy	policy_amount	lead	id
0	AN4UFZ08R	0	0	1197608	b
1	M0XPQP	1	403	c	1116417
2	LUH4V4F9	1	367	a	1125118
3	KVC2IK	0	0	a	449886
4	3ClXG65M6W	0	0	b	668018

In [135]:
```
# cleaning the "lead" and "id" column to get leads and ids in order
sales['lead_type'] = np.where(sales['lead'].str.isalpha(), sales['lead'
], sales['id'])

sales['lead_id'] = np.where(sales['id'].str.isnumeric(), sales['id'], sa
les['lead'])
```

In [136]:
```
sales.head()
```

Out[136]:

	name	bought_policy	policy_amount	lead	id	lead_type	lead_id
0	AN4UFZ08R	0	0	1197608	b	b	1197608
1	M0XPQP	1	403	c	1116417	c	1116417
2	LUH4V4F9	1	367	a	1125118	a	1125118
3	KVC2IK	0	0	a	449886	a	449886
4	3CIXG65M6W	0	0	b	668018	b	668018

In [137]:
```
del sales['lead']
```

In [138]:
```
del sales['id']
```

In [139]:
```
# Capitalizing the "lead_type" column
sales['lead_type'] = sales['lead_type'].str.upper()
```

In [140]:
```
sales.head()
```

Out[140]:

	name	bought_policy	policy_amount	lead_type	lead_id
0	AN4UFZ08R	0	0	B	1197608
1	M0XPQP	1	403	C	1116417
2	LUH4V4F9	1	367	A	1125118
3	KVC2IK	0	0	A	449886
4	3CIXG65M6W	0	0	B	668018

In [141]:
```
# converting the "lead_id" column to integer
sales['lead_id'] = sales['lead_id'].astype(str).astype(int)
```

In [142]:
```
sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 996 entries, 0 to 999
Data columns (total 5 columns):
name              996 non-null object
bought_policy     996 non-null int64
policy_amount     996 non-null int64
lead_type         996 non-null object
lead_id           996 non-null int64
dtypes: int64(3), object(2)
memory usage: 46.7+ KB
```

In [143]:
```
# joining the "sales" and "names" dataframes
df = pd.merge(names, sales, on='lead_id', how='left')
```

In [145]:
```
del df['name_y']
```

In [146]:
```
del df['lead_type_y']
```

In [147]:
```
df = df.rename({'name_x': 'name', 'lead_type_x': 'lead_type'}, axis=1)
```

In [148]:
```
df = df.dropna()
```

In [149]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 996 entries, 0 to 999
Data columns (total 7 columns):
id                996 non-null int64
name              996 non-null object
age               996 non-null int64
lead_id           996 non-null int64
lead_type         996 non-null object
bought_policy     996 non-null float64
policy_amount     996 non-null float64
dtypes: float64(2), int64(3), object(2)
memory usage: 62.2+ KB
```

In [150]:
```
# converting the "bought_policy" column to integer
df['bought_policy'] = df['bought_policy'].astype(int)
```

In [151]:
```
# converting the "policy_amount" column to integer
df['policy_amount'] = df['policy_amount'].astype(int)
```

In [152]:
```
df.head()
```

Out[152]:

	id	name	age	lead_id	lead_type	bought_policy	policy_amount
0	0	AN4UFZ08R	40	1197608	B	0	0
1	1	M0XPQP	41	1116417	C	1	403
2	2	LUH4V4F9	45	1125118	A	1	367
3	3	KVC2IK	47	449886	A	0	0
4	4	3CIXG65M6W	53	668018	B	0	0

Come up with a single figure that uses the data to help us determine how we can grow as a business. Produce a single figure (with a line or two description if you would like) to help our executive team grow the business

For this I decided to go with a Stacked Bar Chart, with the lead type on the X-axis and where the policy was bought or not on the Y -axis.

In [153]:
```
df2 = pd.DataFrame(df.groupby(['bought_policy','lead_type'])['id'].count
())
```

In [154]:
```
df2.reset_index(level=0, inplace=True)
df2
```

Out[154]:

lead_type	bought_policy	id
A	0	152
B	0	157
C	0	130
A	1	192
B	1	173
C	1	192

In [155]:
```
df2 = df2.pivot(index=df2.index,columns='bought_policy')['id']
```

In [156]:
```
df2
```

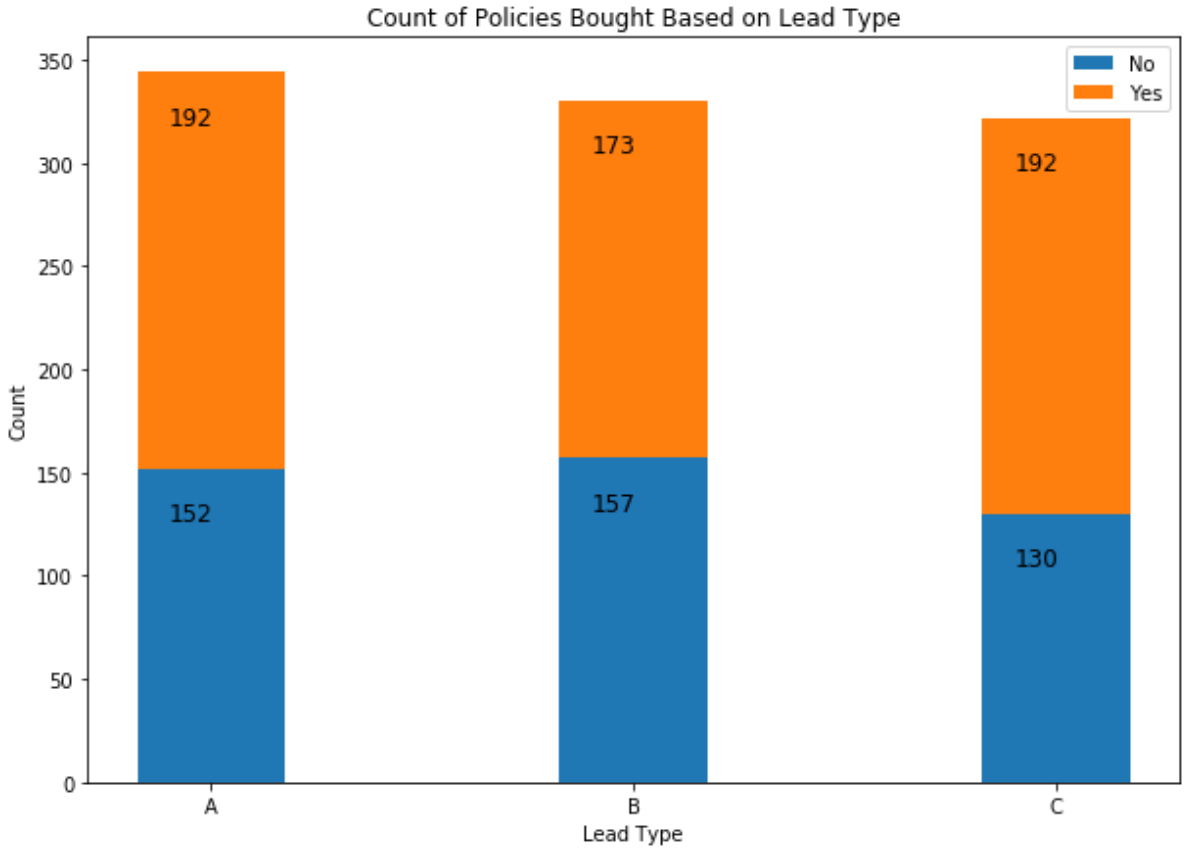
Out[156]:

bought_policy	0	1
lead_type		
A	152	192
B	157	173
C	130	192

In [157]:
```
df2.reset_index(level=0, inplace=True)
```

In [158]:
```
# code for the stacked bar chart
leg = ['No','Yes']
lead = df2.iloc[:,0]
zero = df2.iloc[:,1]
one = df2.iloc[:,2]
index = np.arange(len(df2))
plt.figure(figsize=(10,7))
plt.bar(x=index, height=zero, width=0.35)
plt.bar(x=index, height=one, width=0.35, bottom=zero)
plt.xlabel('Lead Type')
plt.ylabel('Count')
plt.legend(leg)
plt.xticks(index, lead)
nl = list(df2[0])
yl = list(df2[1])
for i in range(len(nl)):
    plt.text(x = index[i]-0.1 , y = zero[i]-25, s = nl[i], size = 12)
for j in range(len(yl)):
    plt.text(x = index[j]-0.1 , y = zero[j]+yl[j]-25, s = yl[j], size =
12)
plt.title("Count of Policies Bought Based on Lead Type")

plt.show()
```



Following insights can be drawn from the above figure:

1. Assuming Insurify gets some commission from the parterns for customers going from Insurify to the partner website and buying a policy, so this graph can help the business team build new strategy for their Type A customers
2. The lead type metric (A, B, C) is not a very reliable source to predict the customer type since Type A and C has the same conversion for the policy being bought (i.e. 192)

In [ ]: