InitialNotebook

December 8, 2025

[181]:
```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive/')
```

```
Drive already mounted at /content/drive/; to attempt to forcibly remount, call
drive.mount("/content/drive/", force_remount=True).
```

[182]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression, LogisticRegression
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from scipy.stats import ttest_ind
from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.model_selection import train_test_split
```

[183]:
```
data = pd.read_csv('/content/drive/MyDrive/Final_Project/
 ↪shopping_behavior_updated.csv')
data[0:1]
#Looking at the data and the columns
```

[183]:

	Customer ID	Age	Gender	Item Purchased	Category	Purchase Amount (USD)	Location	Size	Color	Season	Review Rating	Subscription Status	Shipping Type	Discount Applied	Promo Code Used	Previous Purchases	Payment Method	Frequency of Purchases
0	1	55	Male	Blouse	Clothing	53	Kentucky	L	Gray	Winter	3.1	Yes	Express	Yes	Yes	14	Venmo	Fortnightly

The data has a lot of columns for us to look at. I will begin by playing with different graphs, and looking for any relationships that are important.

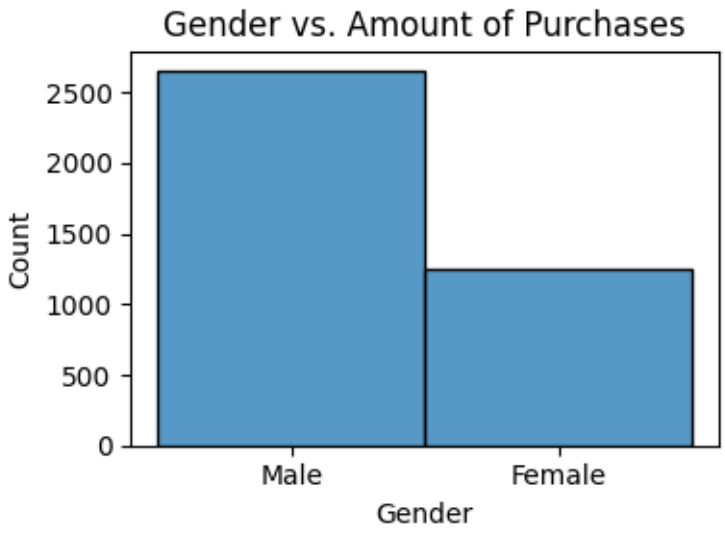
Our two goals are to look at what factors increase the **amount of money** spent in a purchase,

and what factors tend to lead to more **customers returning more frequently**.

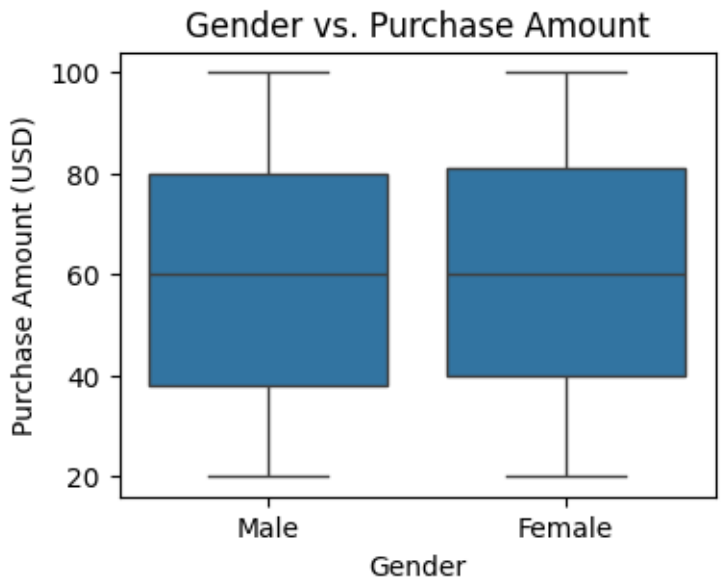
The factors I initally look at are factors that I suspect will be influential, based on my previous research and experience. This will help me get an idea of how some of the data is spread out, but we will have a better idea when we try the **greedy forward algorithm** to determine which factors are the most influential!

Exploring Data:

[184]:
```
plt.figure(figsize=(4, 3))
sns.histplot(data['Gender'])
plt.title('Gender vs. Amount of Purchases')
plt.tight_layout()
plt.show()
```



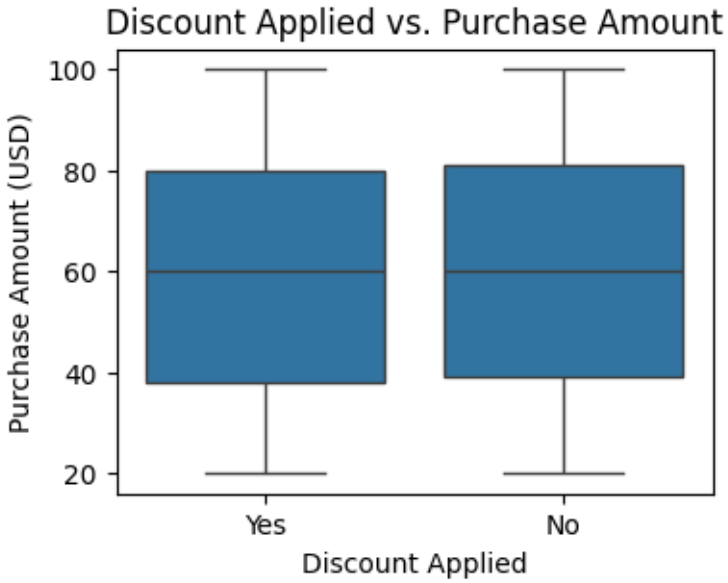
[185]:
```
#lets look at a few columns
#starting with gender and purchase amts.
plt.figure(figsize=(4, 3))
sns.boxplot(x='Gender', y='Purchase Amount (USD)', data=data)
plt.title('Gender vs. Purchase Amount')
plt.show()
```



As seen above, gender does not make seem to be a major indicator on the purchase size. We will rougly look at a few other columns before trying other methods to figure out how we can identify which are the most indicative columns.

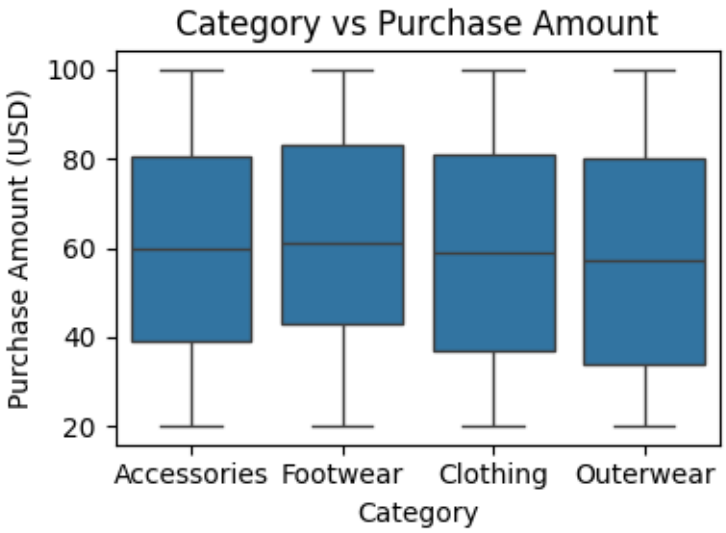
[186]:
```
plt.figure(figsize=(4, 3))
sns.boxplot(x = data['Discount Applied'], y = data['Purchase Amount (USD)'])
plt.title('Discount Applied vs. Purchase Amount')
```

[186]:
```
Text(0.5, 1.0, 'Discount Applied vs. Purchase Amount')
```



[187]:
```
#using a sample to better see the scatterplot relationships
plt.figure(figsize=(4, 3))
data_sample = data.sample(frac = .5, random_state = 1)
sns.scatterplot(x = data_sample['Age'], y = data_sample['Purchase Amount␣
 ↪(USD)'])
plt.title('Age vs Purchase Amount')
plt.tight_layout()
```

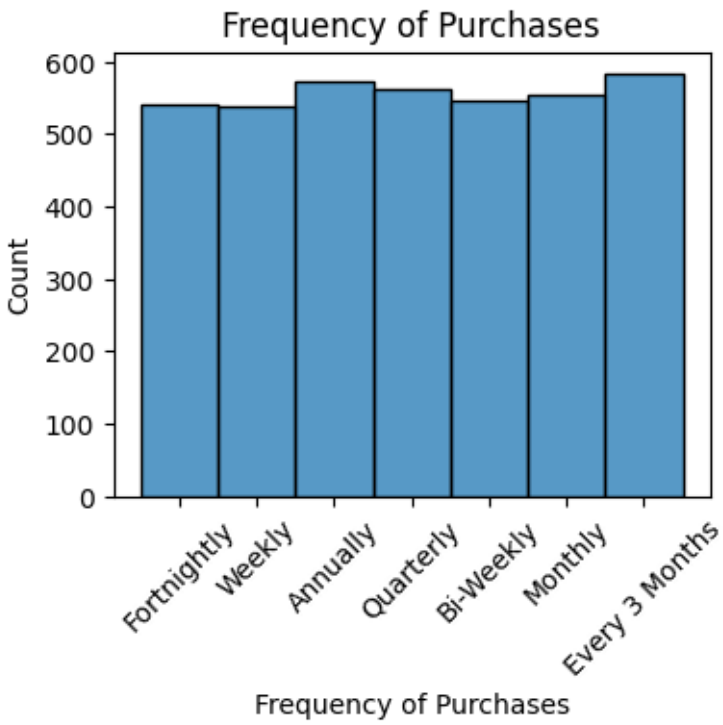
[188]:
```
plt.figure(figsize=(4, 3))
sns.boxplot(x = data_sample['Category'], y = data_sample['Purchase Amount␣
 ↪(USD)'])
plt.title('Category vs Purchase Amount')
plt.tight_layout()
```



Of the previous 3 factors, the **category** of the product seems to have more of an influence on the purchase size.

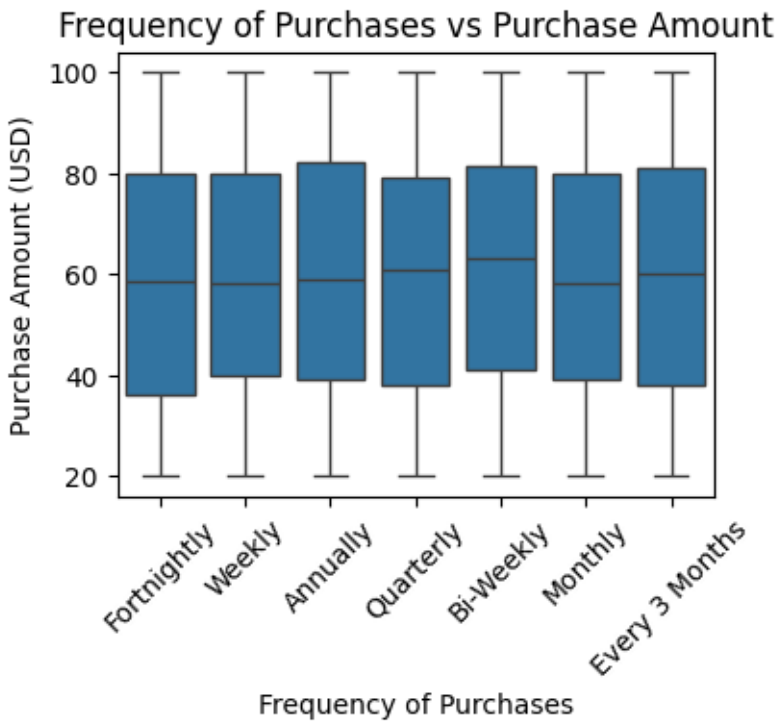
[189]:
```
plt.figure(figsize=(4, 3))
sns.histplot(data['Frequency of Purchases'])
plt.xticks(rotation=45)
plt.title('Frequency of Purchases')
```

[189]:
```
Text(0.5, 1.0, 'Frequency of Purchases')
```

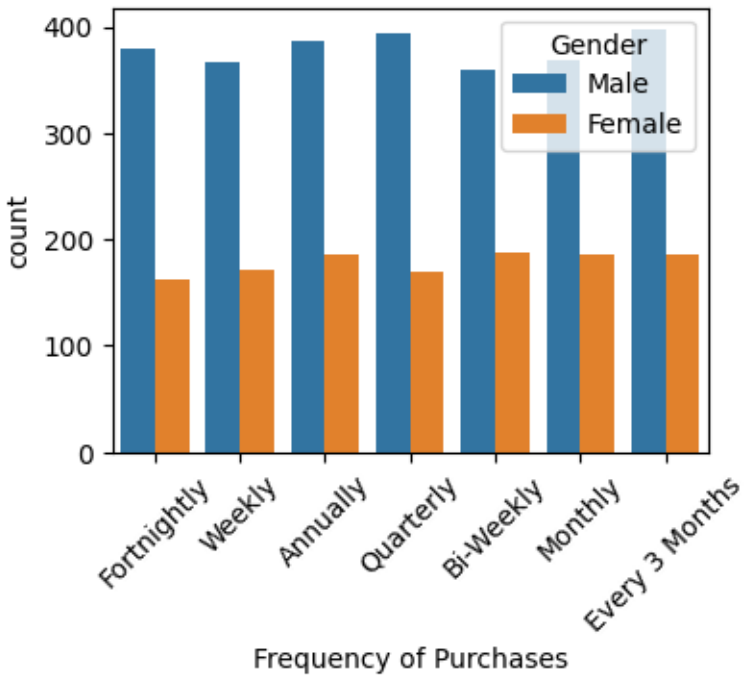


[190]:
```
plt.figure(figsize=(4, 3))
sns.boxplot(x = data['Frequency of Purchases'], y = data['Purchase Amount␣
↪(USD)'])
plt.xticks(rotation=45)
plt.title('Frequency of Purchases vs Purchase Amount')
```

[190]: Text(0.5, 1.0, 'Frequency of Purchases vs Purchase Amount')



[191]:
```
plt.figure(figsize=(4, 3))
sns.countplot(data=data, x = "Frequency of Purchases", hue="Gender")
plt.xticks(rotation = 45)
plt.show()
plt.tight_layout()
```




```
<Figure size 640x480 with 0 Axes>
```

Applying Techniques (t-test & greedy forward feature selector)

I want to get a deeper look into the difference between the genders and the frequency of purchases, so I will conduct a **t-test**.

For the **null hypothesis**: There is no significant difference between the two genders and their shopping frequency.

Alternative Hypothesis: There is a difference between the genders.

Before I begin, I need to **hot-encode** the Frequency variable, which will help me in developing future models as well!

The scale will be 1: Least Frequent (Annually) - 7: Most Frequent (Fortnightly)

[192]:
```
#hot encoding the frequency variable.
data['Frequency Encoded'] = data['Frequency of Purchases'].map({
    'Annually': 1,
    'Quarterly': 2,
    'Every 3 Months': 3,
    'Monthly': 4,
    'Weekly': 5,
    'Biweekly': 6,
    'Fortnightly': 7
})
#displaying just to make sure it encoded properly
```

```
data[['Customer ID', 'Frequency of Purchases', 'Frequency Encoded']][1:4]
```

[192]:

	Customer ID	Frequency of Purchases	Frequency Encoded
1	2	Fortnightly	7.0
2	3	Weekly	5.0
3	4	Weekly	5.0

[193]:
```
#we will inject the median into the null values, since we now are able to use the column numerically
data['Frequency Encoded'] = data['Frequency Encoded'].fillna(data['Frequency Encoded'].median())
data['Frequency Encoded'].isnull().sum()
```

[193]:
```
np.int64(0)
```

[194]:
```
#now setting up the t-test:
females = data[data['Gender'] == 'Female']['Frequency Encoded']
males = data[data['Gender'] == 'Male']['Frequency Encoded']
ttest_ind(a=males,
          b=females,
          equal_var=False, alternative='two-sided')
```

[194]:
```
TtestResult(statistic=np.float64(0.518038647587995),
pvalue=np.float64(0.6044772294810729), df=np.float64(2495.3900834759243))
```

My statistic number was about .53, meaning that females tend to buy *less* frequently than males.

The p-value was about .60, which is much higher than .05. This number being larger tells us that **there is no statistical significant difference** between the mean frequency of purchases between the two genders. We can **accept the null hypothesis**, and reject the alternative.

On the surface, it seemed like males purchased with more frequency, but using the t-test allowed us to see that there is in fact no statistical significant difference between the frequency of purchases between the genders.

We could keep searching for key features manually using tests like these, but a **greedy algorithm** could **save us some time**. It is very interesting to apply our in class techniques to a data set, and see how we can explore different relationships beyond the surface level.

[195]:
```
# I will first encode the other features I want to use that aren't previously numeric
data['Discount Encoded'] = data['Discount Applied'].map({
    'Yes': 1,
    'No': 0
})
data['Season Encoded'] = data['Season'].map({
    'Winter': 0,
    'Spring': 1,
```

```
    'Summer': 2,
    'Fall': 3
})
data[['Discount Applied', 'Discount Encoded', 'Season', 'Season Encoded']][1675:
 ↪1678]
```

[195]:

	Discount Applied	Discount Encoded	Season	Season Encoded
1675	Yes	1	Spring	1
1676	Yes	1	Winter	0
1677	No	0	Summer	2

[196]:
```
#setting up the greedy algorithm model to see what affects frequency of␣
 ↪purchases
featured_cols = data[['Age', 'Purchase Amount (USD)', 'Previous Purchases',␣
 ↪'Review Rating', 'Discount Encoded', 'Season Encoded', ]]
X = featured_cols.values
y = data[['Frequency Encoded']].values.reshape(-1, 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

# Define model
model = LinearRegression()

# Forward selection
sfs = SequentialFeatureSelector(model, n_features_to_select=3,␣
 ↪direction='forward')
sfs.fit(X_train, y_train)

# Selected features
print("Selected features:", sfs.get_support())
print(featured_cols.columns[sfs.get_support()])
```

```
Selected features: [ True False  True False False  True]
Index(['Age', 'Previous Purchases', 'Season Encoded'], dtype='object')
```

Based on the Sequential Feature Selector algorithm, from the set of columns I selected, we can see that the most influential factors for Frequency of Purchases are **Purchase Amount**, **Previous Purcases**, and **Review Rating**.

I selected the 6 columns based on eliminating some that I already looked at, and choosing ones that already had numerical values or were simple to encode.

With this information, I can now create a prediction model to predict how frequently a customer may make purchase, and I could redo this to find the most influential variables for the Purchase Amount.

Git Repo Link: https://github.com/dpineda04/HiddenGems.git