CHIGO'S UBEREATS DATA

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
import seaborn as sns
sns.set()
```

In [3]:

```
data = pd.read_csv('trips_data.csv')
data.head()
```

Out[3]:

	City	Product Type	Trip or Order Status	Request Time	Begin Trip Time	Begin Trip Lat	Begin Trip Lng	Begin Trip Address	Dropoff Time	Dropoff Lat
0	25	UberEATS Marketplace	COMPLETED	2020-11-26 00:52:28 +0000 UTC	2020-11-26 01:39:58 +0000 UTC	32.782505	-96.799784	NaN	2020-11-26 01:39:58 +0000 UTC	32.782505
1	25	UberEATS Marketplace	COMPLETED	2020-11-20 18:50:01 +0000 UTC	2020-11-20 19:10:04 +0000 UTC	32.782505	-96.799784	NaN	2020-11-20 19:10:04 +0000 UTC	32.782505
2	25	UberEATS Marketplace	COMPLETED	2020-11-19 03:07:54 +0000 UTC	2020-11-19 03:22:48 +0000 UTC	32.782505	-96.799784	NaN	2020-11-19 03:22:48 +0000 UTC	32.782505
3	25	UberEATS Marketplace	COMPLETED	2020-11-12 16:02:54 +0000 UTC	2020-11-12 16:18:50 +0000 UTC	32.782505	-96.799784	NaN	2020-11-12 16:18:50 +0000 UTC	32.782505
4	25	UberEATS Marketplace	COMPLETED	2020-11-03 01:35:12 +0000 UTC	2020-11-03 02:14:22 +0000 UTC	32.782505	-96.799784	NaN	2020-11-03 02:14:22 +0000 UTC	32.782505

In [4]:

```
data.describe()
```

Out[4]:

	City	Begin Trip Lat	Begin Trip Lng	Dropoff Lat	Dropoff Lng	Distance (miles)	Fare Amount
count	334.000000	312.000000	312.000000	312.000000	312.000000	334.000000	334.000000
mean	39.685629	35.650023	-85.100756	35.650668	-85.100366	4.343982	18.862246
std	112.069610	6.518907	30.179615	6.519503	30.176603	7.645819	16.359621
min	3.000000	18.370229	-149.768854	18.301156	-149.768854	0.000000	0.000000
25%	25.000000	32.780028	-96.802784	32.780029	-96.801336	0.780000	7.015000

	City	Begin Trip Lat	Begin Trip Lng	Dropoff Lat	Dropoff Lng	Distance (miles)	Fare Amount
50%	25.000000	32.782505	-96.799784	32.782505	-96.799784	2.120000	15.145000
75%	25.000000	32.989847	-95.373899	32.980071	-95.373859	4.310000	27.417500
max	1459.000000	61.201350	9.188496	61.201350	9.699618	89.800000	132.650000

In [5]:

```
x = data ['Distance (miles)']
y = data ['Fare Amount']
```

In [6]:

```
x.shape
```

Out[6]:

```
(334,)
```

In [7]:

```
y.shape
```

Out[7]:

```
(334,)
```

In [8]:

```
x_matrix = x.values.reshape(-1,1)
x_matrix.shape
```

Out[8]:

```
(334, 1)
```

In [9]:

```
reg = LinearRegression()
```

In [10]:

```
reg.fit(x_matrix,y)
```

Out[10]:

```
LinearRegression()
```

In [11]:

```
reg.score(x_matrix,y)
```

Out[11]:

```
0.15904289526522675
```

In [12]:

```
reg.coef_
```

Out[12]:

```
array([0.85330903])
```

In [13]:

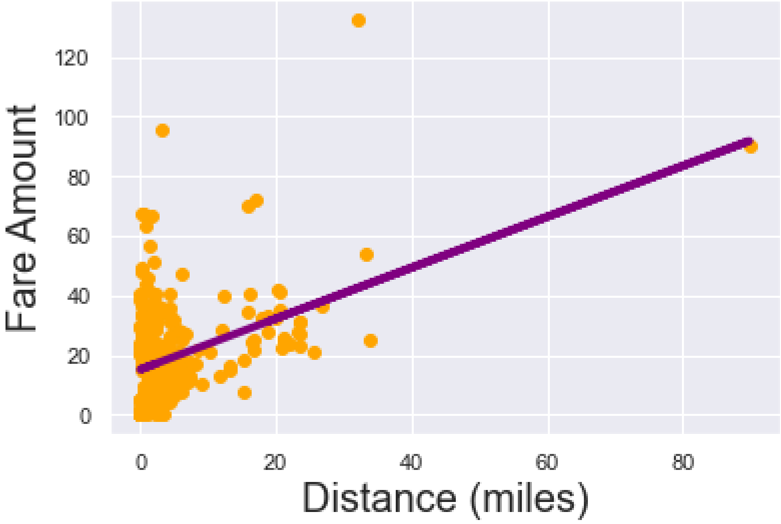
```
reg.intercept_
```

Out[13]:

```
15.155486415774194
```

In [59]:

```
plt.scatter(x,y,c = 'orange')
yhat = reg.coef_*x_matrix + reg.intercept_
yhat = 0.8533*x + 15.155
fig = plt.plot(x,yhat, lw=4, c='purple', label ='regression line')
plt.xlabel('Distance (miles)', fontsize = 20)
plt.ylabel('Fare Amount', fontsize = 20)
plt.show()
```



Hey Chigo, thank you once again for buying my gig. So I used linear regression to analyse your ubereats data. Linear regression seeks to understand the relationship between two or more variables. Simply put, how a variable called the independent variable eg. size, year a house was built etc. can affect another variable eg rent(dependent variable) . In this case, I used two variables, distance in miles as the independent variable and fare amount as the dependent variable and I used a scatter plot (shown above) to visualize this data. As you can see, there's a cluster of points around the bottom of the line(that line is a regression line). This implies that you order mostly from the same place or restaurants within the same area but you can also some some points stretched out far from the cluster(those are called outliers), it's possible you ordered something from a restaurant thats some distance from where you live. What I have done on the whole is that I've built a model that can predict a fare amount based on the distance.

Hey, I just realized not all of the data is from UberEats ;), anyways the theory still holds.

In [87]:

```
new_data = pd.DataFrame(data=[500,3000,2,15,0.5,7.32],columns=['Random_distance(Miles)'])
new_data
```

Out[87]:

	Random_distance(Miles)
0	500.00
1	3000.00
2	2.00
3	15.00
4	0.50
5	7.32

In [89]:

```
reg.predict(new_data)
```

Out[89]:

```
array([ 441.81000093, 2575.08257351,   16.86210447,   27.95512185,
         15.58214093,   21.40170851])
```

In [102…

```
new_data['Predicted_Prices(USD)'] = reg.predict(new_data)
new_data
```

Out[102…

	Random_distance(Miles)	Predicted_Prices(USD)
0	500.00	441.810001

	Random_distance(Miles)	Predicted_Prices(USD)
1	3000.00	2575.082574
2	2.00	16.862104
3	15.00	27.955122
4	0.50	15.582141
5	7.32	21.401709

Hey Chigo, I built a machine learning model that predicted the prices (in USD) of your trips depending on the miles accrued(i used random figures) as you can see in the table directly above this comment. Lets say you took an out-of-town trip that was about 500.00 miles, your fare would cost $441.8. In a situation where there were other variables like traffic congestion, trip duration, etc I would build a model that would predict the price based on a combination of these variables. Thank you once again, I really enjoyed working on this datasets and I appreciate you buying my gig.