Waiter Tips (Case Study)

The food server of a restaurant recorded data about the tips given to the waiters for serving the food. The data recorded by the food server is as follows:

- **total_bill**: Total bill in dollars including taxes
- **tip**: Tip given to waiters in dollars
- **sex**: Gender of the person paying the bill
- **smoker**: Whether the person smoked or not
- **day**: Day of the week
- **time**: Lunch or dinner
- **size**: Number of people at the table

Task

Based on this data, our task is to find the factors affecting waiter tips and train a machine learning model to predict the waiter’s tipping.

In [1]:
```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
```

dataset: https://www.kaggle.com/datasets/aminizahra/tips-dataset (https://www.kaggle.com/datasets/aminizahra/tips-dataset)

In [2]:
```
data = pd.read_csv("tips.csv")
print(data.head())
```

```
   total_bill   tip     sex smoker  day    time  size  price_per_pe
rson  \
0       16.99  1.01  Female     No  Sun  Dinner     2
8.49
1       10.34  1.66    Male     No  Sun  Dinner     3
3.45
2       21.01  3.50    Male     No  Sun  Dinner     3
7.00
3       23.68  3.31    Male     No  Sun  Dinner     2             1
1.84
4       24.59  3.61  Female     No  Sun  Dinner     4
6.15

           Payer Name         CC Number Payment ID
0  Christy Cunningham  3560325168603410    Sun2959
1      Douglas Tucker  4478071379779230    Sun4608
2      Travis Walters  6011812112971322    Sun4458
3    Nathaniel Harris  4676137647685994    Sun5260
4        Tonya Carter  4832732618637221    Sun2251
```

last 3 column are not important

```
In [7]: data = pd.read_csv("tips.csv")
        data = data.iloc[:, :7]
        print(data.head())
```

```
   total_bill   tip     sex smoker  day    time  size
0       16.99  1.01  Female     No  Sun  Dinner     2
1       10.34  1.66    Male     No  Sun  Dinner     3
2       21.01  3.50    Male     No  Sun  Dinner     3
3       23.68  3.31    Male     No  Sun  Dinner     2
4       24.59  3.61  Female     No  Sun  Dinner     4
```

EDA

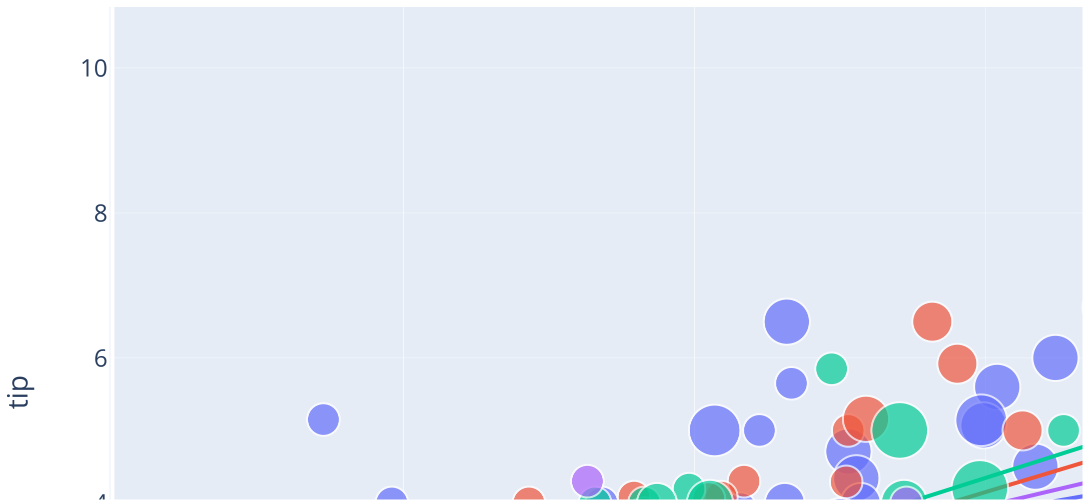
Tips Analysis

Let's analyze the tips given to the waiters based on the following factors:

1. Explore the relationship between the total bill paid and the corresponding tips.
2. Investigate how the number of people at a table correlates with the tips provided to the waiters.
3. Examine the variation in tips based on the day of the week.

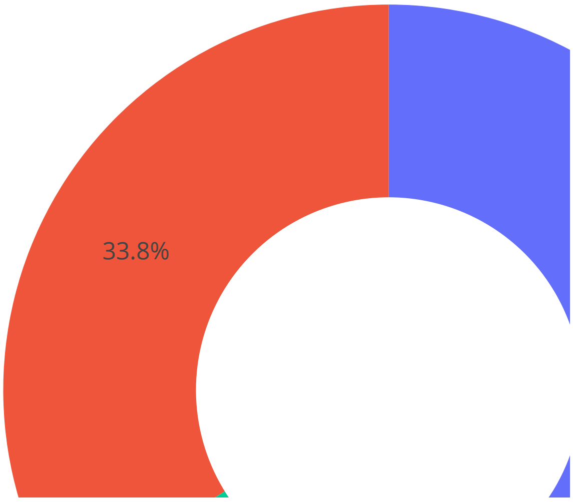
In [8]:

```
figure = px.scatter(data_frame = data, x="total_bill",
                    y="tip", size="size", color= "day", trendline="o
figure.show()
```



In [11]:

```
figure = px.pie(data,
                values='tip',
                names='day',hole = 0.5)
figure.show()
```

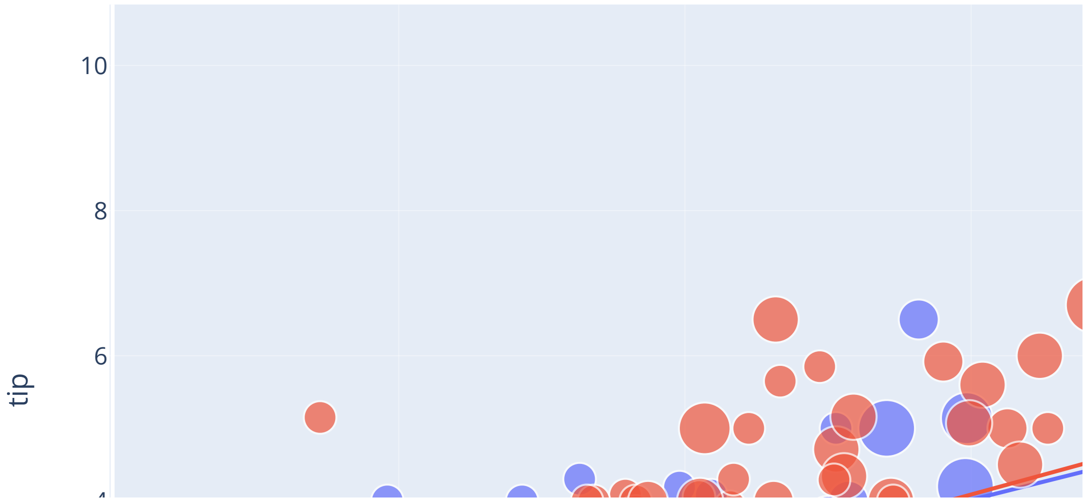


on Saturdays, most tips are given to the waiters. Now let’s look at the number of tips given to waiters by gender of the person paying the bill to see who tips waiters the most:

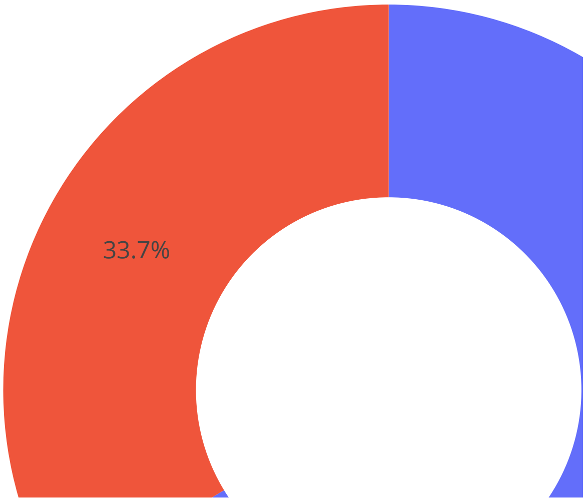
Now let’s have a look at the tips given to the waiters according to:

1. the total bill paid
2. the number of people at a table
3. and the gender of the person paying the bill:

In [9]:
```
figure = px.scatter(data_frame = data, x="total_bill",
                    y="tip", size="size", color= "sex", trendline="o
figure.show()
```



In [14]:
```
figure = px.pie(data,
             values='tip',
             names='sex',hole = 0.5)
figure.show()
```



According to the visualization above, most tips are given by men. Now let’s see if a smoker tips more or a non-smoker:

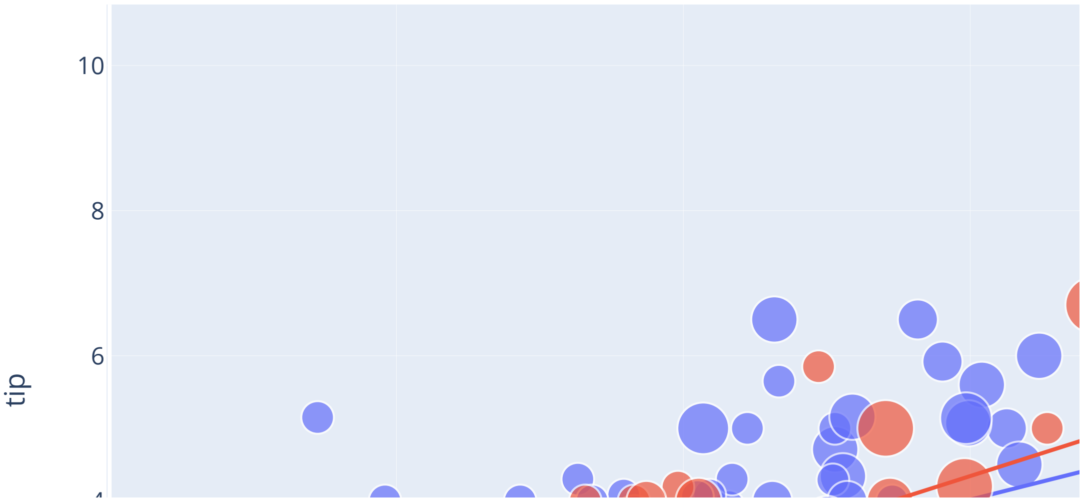
According to the visualization above, most tips are given by men. Now let’s see if a smoker tips more or a non-smoker:

Now let’s have a look at the tips given to the waiters according to:

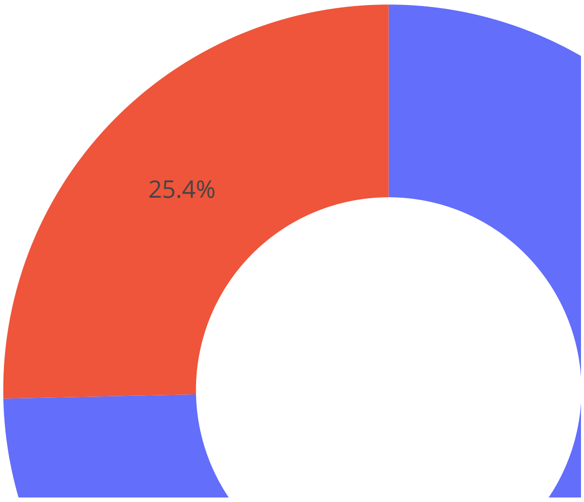
1. the total bill paid
2. the number of people at a table
3. and the time of the meal:

In [10]:

```
figure = px.scatter(data_frame = data, x="total_bill",
                    y="tip", size="size", color= "time", trendline="
figure.show()
```



In [16]:
```
figure = px.pie(data,
             values='tip',
             names='time',hole = 0.5)
figure.show()
```



So this is how we can analyze all the factors affecting waiter tips. Now in the section below, I will take you through how to train a machine learning model for the task of waiter tips prediction.

Waiter Tips Prediction Model

Before training a waiter tips prediction model, I will do some data transformation by transforming the categorical values into numerical values:

In [17]:

```
data["sex"] = data["sex"].map({"Female": 0, "Male": 1})
data["smoker"] = data["smoker"].map({"No": 0, "Yes": 1})
data["day"] = data["day"].map({"Thur": 0, "Fri": 1, "Sat": 2, "Sun":
data["time"] = data["time"].map({"Lunch": 0, "Dinner": 1})
data.head()
```

Out[17]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	0	0	3	1	2
1	10.34	1.66	1	0	3	1	3
2	21.01	3.50	1	0	3	1	3
3	23.68	3.31	1	0	3	1	2
4	24.59	3.61	0	0	3	1	4

Now I will split the data into training and test sets:

In [27]:

```
x = np.array(data[["total_bill", "sex", "smoker", "day",
                   "time", "size"]])
y = np.array(data["tip"])

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.2,
                                                    random_state=47)
```

Now below is how we can train a machine learning model for the task of waiter tips prediction using Python:

In [28]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(train, ytrain)
```

Out[28]:

```
▾ LinearRegression
LinearRegression()
```

Now let’s test the performance of this model by giving inputs to this model according to the features that we have used to train this model:

In [29]:

```
# features = [[total_bill, "sex", "smoker", "day", "time", "size"]]
features = np.array([[24.50, 1, 0, 0, 1, 4]])
model.predict(features)
```

Out[29]:

```
array([3.76813381])
```

In [51]:

```
w_init = model.coef_
b_init = model.intercept_
print("Coefficients:", w_init)
print("Intercept:", b_init)
```

```
Coefficients: [ 0.1   -0.14 -0.1   -0.03   0.08   0.19]
Intercept: 0.7012536263964435
```

In [52]:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print(f"Mean Absolute Error: {mae:.2f}")
print(f"Mean Squared Error: {mse:.2f}")
print(f"Root Mean Squared Error: {rmse:.2f}")
```

```
Mean Absolute Error: 0.89
Mean Squared Error: 1.23
Root Mean Squared Error: 1.11
```