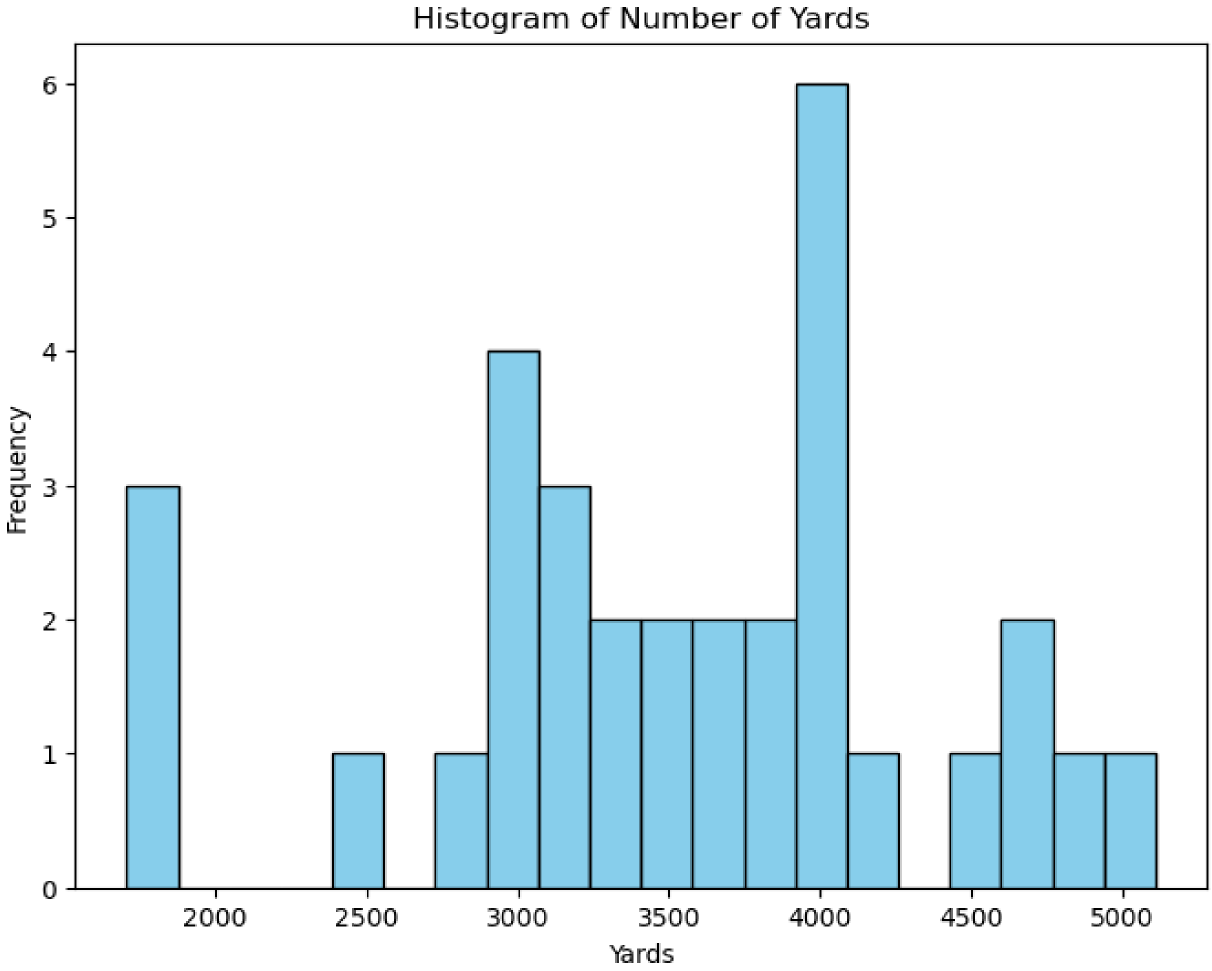
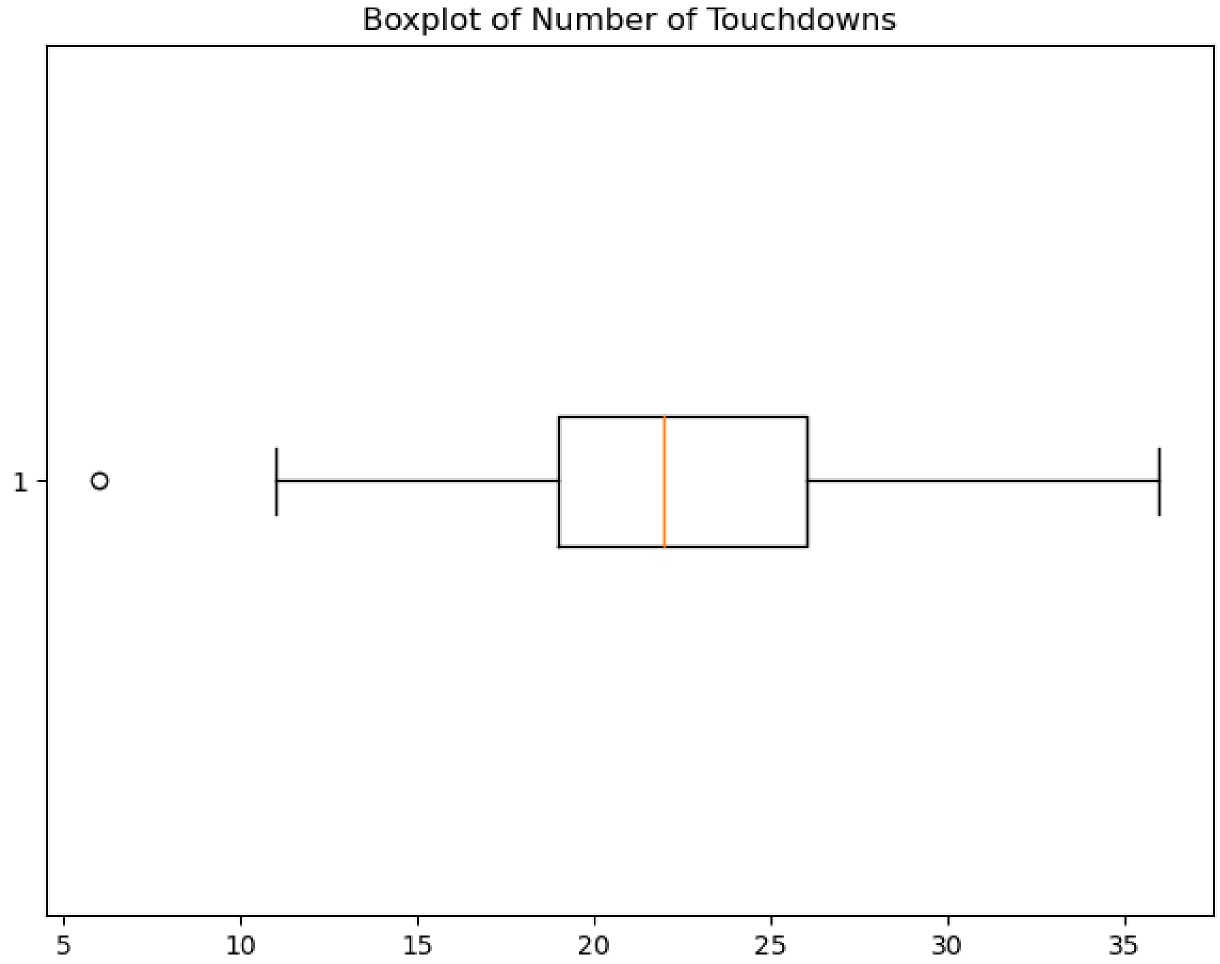
In this exercise, you will be working with the file qb stats.csv. This file contains stats on NFL quar□terbacks in the 2019 season. The data was obtained from https://www.footballdb.com/statistics/nfl/player-stats/passing/2019/regular-season. The data includes the following. • player's name • team • number of games played • pass attempts • pass completions • completion percentage • number of yards • yards per attempt • touchdowns • interceptions • sacks • quarterback rating Use Python code to answer the following questions. (a) Find the mean of each numerical column of data. (b) Find the standard deviation of each numerical column of data. (c) Create a histogram of the number of yards; label it appropriately. (d) Create a boxplot of the number of touchdowns. Identify any outliers.

In [3]:

```python
#load pd, plt
import pandas as pd
import matplotlib.pyplot as plt

#qb_stats.csv
data = pd.read_csv('qb_stats.csv')

#(a) Find the mean of each numerical column of data
means = data.mean()
print("mean of each numerical column of data:")
print(means)

#(b) Find the standard deviation of each numerical column of data
std_devs = data.std()
print("\nstandard deviation of each numerical column:")
print(std_devs)

#(c) Create a histogram of the number of yards
#yds
plt.figure(figsize=(8, 6))
plt.hist(data['yds'], bins=20, color='skyblue', edgecolor='black')
plt.title('Histogram of Number of Yards')
plt.xlabel('Yards')
plt.ylabel('Frequency')
plt.show()

#(d) Create a boxplot of the number of touchdowns
plt.figure(figsize=(8, 6))
plt.boxplot(data['td'], vert=False)
plt.title('Boxplot of Number of Touchdowns')
plt.show()

#identify outliers
#td
outliers = data[(data['td'] > data['td'].mean() + 1.5 * data['td'].std()) | (data['td'
print("\noutliers:")
print(outliers)
```

```
C:\Users\lexiw\AppData\Local\Temp\ipykernel_13404\1233245125.py:9: FutureWarning: The
default value of numeric_only in DataFrame.mean is deprecated. In a future version, i
t will default to False. In addition, specifying 'numeric_only=None' is deprecated. S
elect only valid columns or specify the value of numeric_only to silence this warnin
g.
  means = data.mean()
C:\Users\lexiw\AppData\Local\Temp\ipykernel_13404\1233245125.py:14: FutureWarning: Th
e default value of numeric_only in DataFrame.std is deprecated. In a future version,
it will default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this warnin
g.
  std_devs = data.std()
mean of each numerical column of data:
games         14.125000
att          478.406250
cmp          306.562500
comp_pct      64.296875
yds         3515.062500
ypa            7.375000
td            22.375000
int           10.343750
sack          32.312500
loss         217.093750
rating        93.387500
dtype: float64

standard deviation of each numerical column:
games         2.406309
att         110.360366
cmp          68.929575
comp_pct      3.627537
yds         852.437489
ypa           0.796768
td            6.256609
int           5.839849
sack         10.514008
loss         70.786075
rating       11.054725
dtype: float64
```


Histogram of Number of Yards
Frequency
0
1
2
3
4
5
6
2000
2500
3000
3500
4000
4500
5000
Yards



```
outliers:
              name team  games    att    cmp  comp_pct     yds  ypa    td  \
2    Lamar Jackson  BAL   15.0  401.0  265.0      66.1  3127.0  7.8  36.0   
14     Case Keenum  WAS   10.0  247.0  160.0      64.8  1707.0  6.9  11.0   
24      Joe Flacco  DEN    8.0  262.0  171.0      65.3  1822.0  7.0   6.0   
26  Jameis Winston   TB   16.0  626.0  380.0      60.7  5109.0  8.2  33.0   

     int  sack   loss  rating  
2    6.0  23.0  106.0   113.3  
14   5.0  15.0  145.0    91.3  
24   5.0  26.0  194.0    85.1  
26  30.0  47.0  282.0    84.3  
```

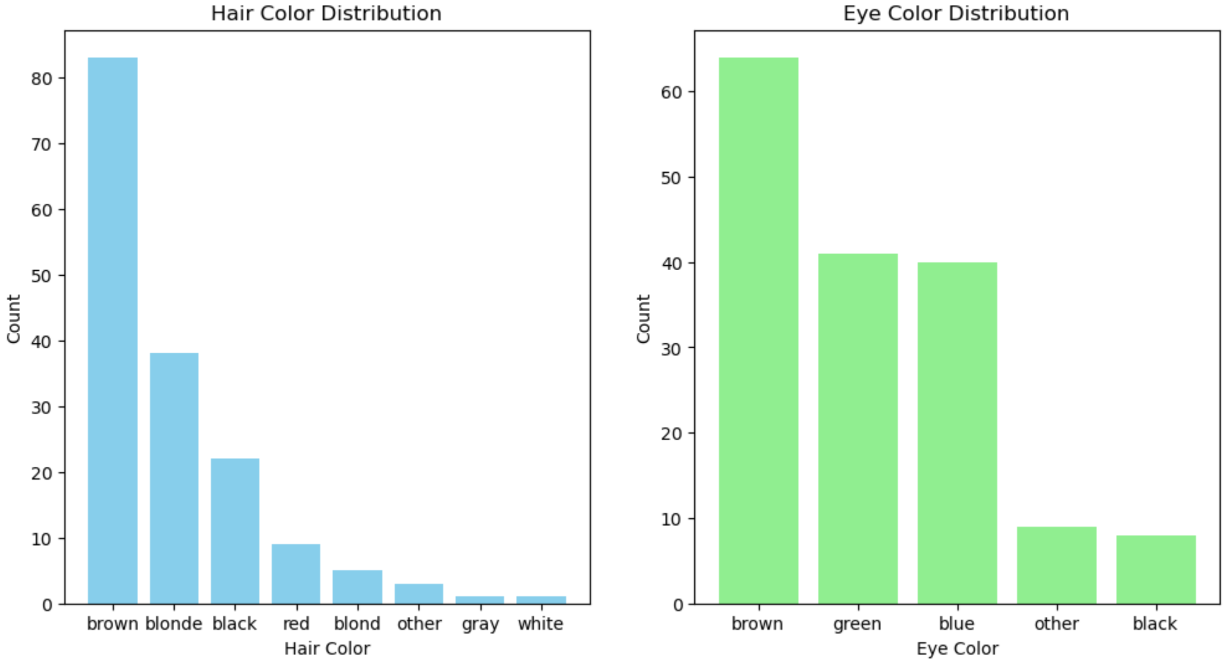
In this exercise, you will be working with the file survey data.csv. This data has a list of college students who were asked their hair color and eye color. Use Python code to answer the following questions. (a) Based off of this data, what is the probability a college student has brown hair? (b) Based off of this data, what is the probability a college student has blue eyes? (c) Based off of this data, what is the probability a college student has blue eyes given that they have brown hair? (d) Based off of this data, what is the probability a college student has brown hair given that they have blue eyes? (e) Do your results above indicate that college students having brown hair and blue eyes are inde□pendent of one another? Explain. (f) Create a bar graph of the hair color and eye color of this group of students. Label the graphs appropriatel

In [8]:

```
#load pd, plt
import pandas as pd
import matplotlib.pyplot as plt
```

```
#survey_data.csv
data = pd.read_csv('survey_data.csv')

#uniform 2 columns
data['eye_color'] = data['eye_color'].str.lower()
data['hair_color'] = data['hair_color'].str.lower()

#probabilities
#use caps
total_students = len(data)
brown_hair_count = len(data[data['hair_color'] == 'brown'])
prob_brown_hair = brown_hair_count / total_students
print(f"Probability a college student has brown hair: {prob_brown_hair:.2f}")

blue_eyes_count = len(data[data['eye_color'] == 'blue'])
prob_blue_eyes = blue_eyes_count / total_students
print(f"Probability a college student has blue eyes: {prob_blue_eyes:.2f}")

brown_blue_count = len(data[(data['hair_color'] == 'brown') & (data['eye_color'] == 'b
prob_blue_given_brown = brown_blue_count / brown_hair_count
print(f"Probability a college student has blue eyes given brown hair: {prob_blue_giver

blue_brown_count = len(data[(data['eye_color'] == 'blue') & (data['hair_color'] == 'br
if blue_eyes_count == 0:
    print("No students have blue eyes in the dataset.")
else:
    prob_brown_given_blue = blue_brown_count / blue_eyes_count
    print(f"Probability a college student has brown hair given blue eyes: {prob_brown_

#if/else for prob
if prob_blue_given_brown == prob_blue_eyes and prob_brown_given_blue == prob_brown_hai
    print("The results indicate that hair color and eye color are independent.")
else:
    print("The results indicate that hair color and eye color are not independent.")

#bar graph of hair color and eye color
hair_color_counts = data['hair_color'].value_counts()
eye_color_counts = data['eye_color'].value_counts()

fig, ax = plt.subplots(1, 2, figsize=(12, 6))
ax[0].bar(hair_color_counts.index, hair_color_counts.values, color='skyblue')
ax[0].set_title('Hair Color Distribution')
ax[0].set_xlabel('Hair Color')
ax[0].set_ylabel('Count')

ax[1].bar(eye_color_counts.index, eye_color_counts.values, color='lightgreen')
ax[1].set_title('Eye Color Distribution')
ax[1].set_xlabel('Eye Color')
ax[1].set_ylabel('Count')

plt.show()
```

```
Probability a college student has brown hair: 0.51
Probability a college student has blue eyes: 0.25
Probability a college student has blue eyes given brown hair: 0.22
Probability a college student has brown hair given blue eyes: 0.45
The results indicate that hair color and eye color are not independent.
```



Hair Color Distribution

Eye Color Distribution

In [ ]: