Codecademy NBA Trends Project

Analyze National Basketball Association (NBA) data to look at associations between teams, win-rates, playoff appearances, and more.

In this project, you'll analyze data from the NBA (National Basketball Association) and explore possible associations.

This data was originally sourced from 538's Analysis of the Complete History Of The NBA and contains the original, unmodified data from Basketball Reference as well as several additional variables 538 added to perform their own analysis.

You can read more about the data and how it's being used by 538 here. For this project we've limited the data to just 5 teams and 10 columns (plus one constructed column, `point_diff`, the difference between `pts` and `opp_pts`).

You will create several charts and tables in this project, so you'll need to use `plt.clf()` between plots in your code so that the plots don't layer on top of one another.

In [1]:

```
import pandas as pd
import numpy as np
from scipy.stats import pearsonr, chi2_contingency
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#to make the output look nicer
np.set_printoptions(suppress=True, precision = 2)
```

In [3]:

```
nba = pd.read_csv('nba_games.csv')
nba.head()
```

Out[3]:

	game_id	year_id	fran_id	opp_fran	game_location	is_playoffs	pts	opp_pts	game_result	forecast	point_diff
0	194611010TRH	1947	Knicks	Huskies	A	0	68	66	W	0.359935	2
1	194611020CHS	1947	Knicks	Stags	A	0	47	63	L	0.368899	-16
2	194611020PRO	1947	Celtics	Steamrollers	A	0	53	59	L	0.359935	-6
3	194611050BOS	1947	Celtics	Stags	H	0	55	57	L	0.620204	-2
4	194611070STB	1947	Knicks	Bombers	A	0	68	63	W	0.339290	5

In [4]:

```
# Subset Data to 2010 Season, 2014 Season
nba_2010 = nba[nba.year_id == 2010]
nba_2014 = nba[nba.year_id == 2014]
```

Task 1

The data has been subset for you into two smaller datasets: games from 2010 (named nba_2010) and games from 2014 (named nba_2014). To start, let’s focus on the 2010 data.

Suppose you want to compare the knicks to the nets with respect to points earned per game. Using the pts column from the nba_2010 DataFrame, create two series named knicks_pts (fran_id = "Knicks") and

nets_pts(fran_id = "Nets") that represent the points each team has scored in their games.

In [5]:

```
knicks_pts = nba_2010[nba_2010.fran_id == 'Knicks']['pts']
nets_pts = nba_2010[nba_2010.fran_id == 'Nets']['pts']
```

Task 2

Calculate the difference between the two teams’ average points scored and save the result as diff_means_2010. Based on this value, do you think fran_id and pts are associated? Why or why not?

In [6]:

```
diff_means_2010 = knicks_pts.mean() - nets_pts.mean()
diff_means_2010
```

Out[6]:

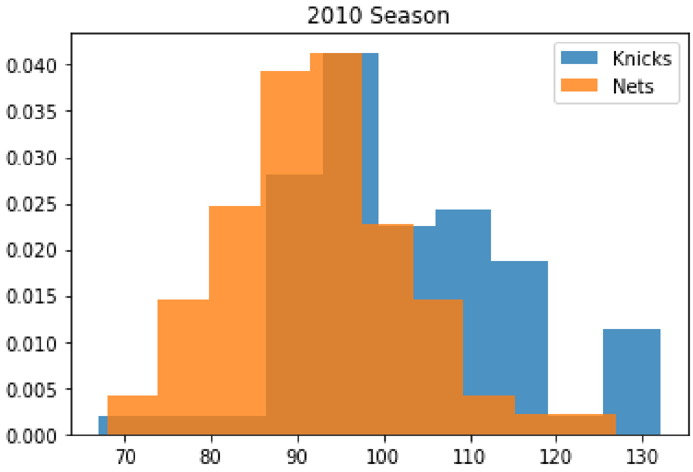
```
9.731707317073173
```

Task 3

Rather than comparing means, it’s useful look at the full distribution of values to understand whether a difference in means is meaningful. Create a set of overlapping histograms that can be used to compare the points scored for the Knicks compared to the Nets. Use the series you created in the previous step (1) and the code below to create the plot. Do the distributions appear to be the same?

In [7]:

```
plt.hist(knicks_pts, alpha = .8, density = True, label = 'Knicks')
plt.hist(nets_pts, alpha = .8, density = True, label = 'Nets')
#note that density is used for newer version of matplotlib
plt.legend()
plt.title("2010 Season")
plt.show()
```



Task 4

Now, let’s compare the 2010 games to 2014. Replicate the steps from Tasks 2 and 3 using `nba_2014`. First, calculate the mean difference between the two teams points scored. Save and print the value as `diff_means_2014`. Did the difference in points get larger or smaller in 2014? Then, plot the overlapping histograms. Does the mean difference you calculated make sense?

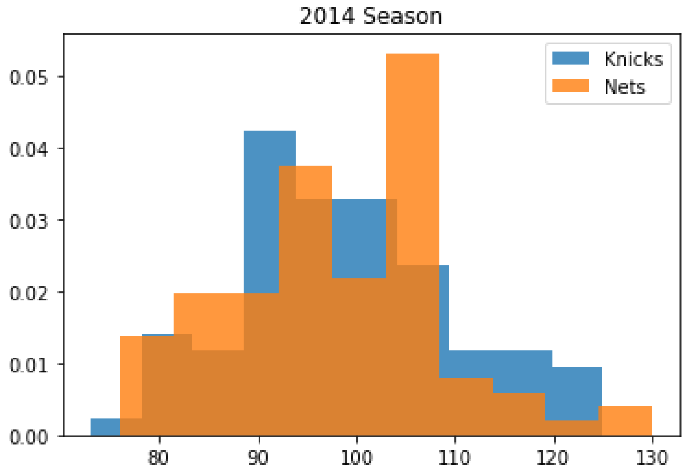
In [8]:

```
knicks_pts_14 = nba_2014[nba_2014.fran_id == 'Knicks']['pts']
nets_pts_14 = nba_2014[nba_2014.fran_id == 'Nets']['pts']
```

```
diff_means_2014 = knicks_pts_14.mean() - nets_pts_14.mean()
print(diff_means_2014)

plt.hist(knicks_pts_14, alpha = .8, density = True, label = 'Knicks')
plt.hist(nets_pts_14, alpha = .8, density = True, label = 'Nets')
plt.legend()
plt.title("2014 Season")
plt.show()
```

```
9.731707317073173
```



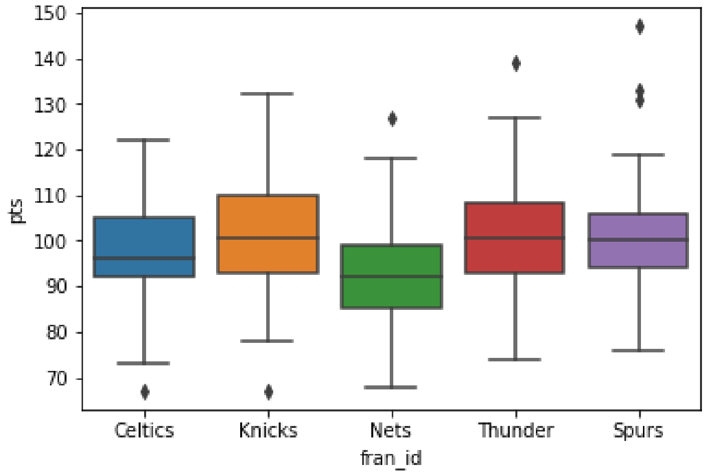
Task 5

For the remainder of this project, we'll focus on data from 2010. Let's now include all teams in the dataset and investigate the relationship between franchise and points scored per game.

Using nba_2010, generate side-by-side boxplots with points scored (pts) on the y-axis and team (fran_id) on the x-axis. Is there any overlap between the boxes? Does this chart suggest that fran_id and pts are associated? Which pairs of teams, if any, earn different average scores per game?

In [9]:

```
sns.boxplot(data = nba_2010, x = 'fran_id', y = 'pts')
plt.show()
```



Task 6

We'd like to know if teams tend to win more games at home compared to away.

The variable, `game_result`, indicates whether a team won a particular game ('W' stands for "win" and 'L' stands for "loss"). The variable, `game_location`, indicates whether a team was playing at home or away ('H' stands for "home" and 'A' stands for "away").

Data scientists will often calculate a contingency table of frequencies to help them determine if categorical variables are associated. Calculate a table of frequencies that shows the counts of game_result and

game_location.

Save your result as `location_result_freq` and print your result. Based on this table, do you think the variables are associated?`

In [10]:

```
location_result_freq = pd.crosstab(nba_2010.game_result, nba_2010.game_location)
location_result_freq
```

Out[10]:

game_location	A	H
game_result		
L	133	105
W	92	120

Task 7

Convert this table of frequencies to a table of proportions and save the result as `location_result_proportions`.

In [11]:

```
location_result_proportions = location_result_freq/len(nba_2010)
location_result_proportions
```

Out[11]:

game_location	A	H
game_result		
L	0.295556	0.233333
W	0.204444	0.266667

Task 8

Using the contingency table created in the previous exercise (Ex. 7), calculate the expected contingency table (if there were no association) and the Chi-Square statistic.

Does the actual contingency table look similar to the expected table — or different? Based on this output, do you think there is an association between these variables?

In [12]:

```
chi2, pval, dof, expected = chi2_contingency(location_result_freq)
print(expected)
print(chi2)
```

```
[[119. 119.]
 [106. 106.]]
6.501704455367053
```

For a 2x2 table, Chi-squared greater than about 4 indicates an association. We've exceeded that!

Task 9

For each game, 538 has calculated the probability that each team will win the game. We want to know if teams with a higher probability of winning (according to 538) also tend to win games by more points.

In the data, 538's prediction is saved as `forecast`. The `point_diff` column gives the margin of

victory/defeat for each team (positive values mean that the team won; negative values mean that they lost).

Using `nba_2010`, calculate the covariance between `forecast` (538's projected win probability) and `point_diff` (the margin of victory/defeat) in the dataset. Save and print your result. Looking at the matrix, what is the covariance between these two variables?

In [13]:

```
point_diff_forecast_cov = np.cov(nba_2010.point_diff, nba_2010.forecast)
point_diff_forecast_cov
```

Out[13]:

```
array([[186.56,   1.37],
       [  1.37,   0.05]])
```

Task 10

Because 538's forecast variable is reported as a probability (not a binary), we can calculate the strength of the correlation.

Using nba_2010, calculate the correlation between `forecast` and `point_diff`. Call this `point_diff_forecast_corr`. Save and print your result. Does this value suggest an association between the two variables?

In [14]:

```
point_diff_forecast_corr = pearsonr(nba_2010.forecast, nba_2010.point_diff)
point_diff_forecast_corr
```

Out[14]:

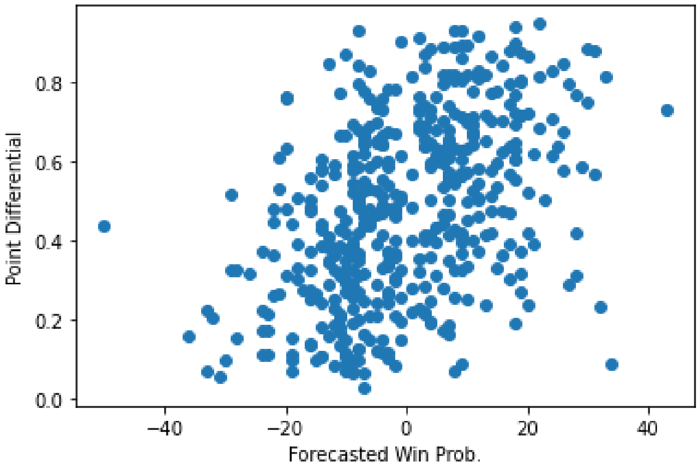
```
(0.44020887084680815, 9.410391573138826e-23)
```

Task 11

Generate a scatter plot of `forecast` (on the x-axis) and `point_diff` (on the y-axis). Does the correlation value make sense?

In [15]:

```
plt.clf() #to clear the previous plot
plt.scatter('forecast', 'point_diff', data=nba_2010)
plt.xlabel('Forecasted Win Prob.')
plt.ylabel('Point Differential')
plt.show()
```



In [ ]:

```

```