DS3001_HW1_Q1

February 11, 2021

[1]:
```
import pandas as pd
import numpy as np
import warnings
import matplotlib.pyplot as plt

warnings.filterwarnings('ignore')
```

[2]:
```
df = pd.read_csv('soccer17.csv')
df = df.loc[(df.Y == 17)]
flds = ['Div','Date','Y','HomeTeam','AwayTeam','FTHG','FTAG']
df = df.loc[:,flds].copy()
df = df.rename({'HomeTeam':'Team_Home','AwayTeam':'Team_Away','FTHG':
 ↪'G_Home','FTAG':'G_Away'}, axis = 1)
df.loc[:,'GameID'] = df.index
df['pd_Home'] = df.G_Home - df.G_Away
df['pd_Away'] = df.G_Away - df.G_Home
```

[3]:
```
df_melt = pd.melt(df, id_vars = ['GameID', 'Div'], value_vars =␣
 ↪['Team_Home','Team_Away'],
            var_name = 'isHome', value_name = 'Team')
df_melt.loc[:,'isHome'] = 1*(df_melt.isHome == 'Team_Home')
```

[4]:
```
df_melt2 = pd.melt(df, id_vars=['GameID', 'Div'], value_vars=['pd_Home',␣
 ↪'pd_Away'], var_name = 'isHome', value_name = 'pd')
df_melt2.loc[:,'isHome'] = 1*(df_melt2.isHome == 'pd_Home')
```

[5]:
```
df_merge = df_melt.merge(df_melt2, on=['GameID', 'Div', 'isHome'])
df_merge['Win'] = 1*(df_merge.pd > 0)
df_merge['Draw'] = 1*(df_merge.pd == 0)
df_merge['Loss'] = 1*(df_merge.pd < 0)
```

[6]:
```
avg_pd = df_merge.groupby(['Team']).mean()['pd']
gp = df_merge.groupby(['Team']).size()
wins = df_merge.groupby(['Team']).agg({'Win': sum})['Win']
draws = df_merge.groupby(['Team']).agg({'Draw': sum})['Draw']
losses = df_merge.groupby(['Team']).agg({'Loss': sum})['Loss']
team = avg_pd.index
```

```
div = df_merge[['Div', 'Team']].drop_duplicates().sort_values('Team')['Div']

output = pd.DataFrame(list(zip(div, team, avg_pd, gp, wins, losses, draws)),
                      columns=['Div', 'Team', 'Avg PD', 'Games Played', 'Wins',␣
↪'Losses', 'Draws'])
```

1 1a

[7]:
```
df1 = output.loc[output.Div == 'EPL'].drop('Div', 1)
df1.sort_values(by='Avg PD', ascending=False).reset_index().drop('index', 1)
```

[7]:

	Team	Avg PD	Games Played	Wins	Losses	Draws
0	Man City	2.078947	38	32	2	4
1	Liverpool	1.210526	38	21	5	12
2	Man United	1.052632	38	25	7	6
3	Tottenham	1.000000	38	23	7	8
4	Chelsea	0.631579	38	21	10	7
5	Arsenal	0.605263	38	19	13	6
6	Burnley	-0.078947	38	14	12	12
7	Leicester	-0.105263	38	12	15	11
8	Newcastle	-0.210526	38	12	18	8
9	Crystal Palace	-0.263158	38	11	16	11
10	Everton	-0.368421	38	13	15	10
11	Bournemouth	-0.421053	38	11	16	11
12	Southampton	-0.500000	38	7	16	15
13	Brighton	-0.526316	38	9	16	13
14	Watford	-0.526316	38	11	19	8
15	West Ham	-0.526316	38	10	16	12
16	West Brom	-0.657895	38	6	19	13
17	Swansea	-0.736842	38	8	21	9
18	Huddersfield	-0.789474	38	9	19	10
19	Stoke	-0.868421	38	7	19	12

2 1b

[8]:
```
df1['Points'] = df1.Wins*3 + df1.Draws
df1.sort_values(by='Points', ascending=False).reset_index().drop('index', 1)
```

[8]:

	Team	Avg PD	Games Played	Wins	Losses	Draws	Points
0	Man City	2.078947	38	32	2	4	100
1	Man United	1.052632	38	25	7	6	81
2	Tottenham	1.000000	38	23	7	8	77
3	Liverpool	1.210526	38	21	5	12	75

4	Chelsea	0.631579	38	21	10	7	70
5	Arsenal	0.605263	38	19	13	6	63
6	Burnley	-0.078947	38	14	12	12	54
7	Everton	-0.368421	38	13	15	10	49
8	Leicester	-0.105263	38	12	15	11	47
9	Crystal Palace	-0.263158	38	11	16	11	44
10	Bournemouth	-0.421053	38	11	16	11	44
11	Newcastle	-0.210526	38	12	18	8	44
12	West Ham	-0.526316	38	10	16	12	42
13	Watford	-0.526316	38	11	19	8	41
14	Brighton	-0.526316	38	9	16	13	40
15	Huddersfield	-0.789474	38	9	19	10	37
16	Southampton	-0.500000	38	7	16	15	36
17	Stoke	-0.868421	38	7	19	12	33
18	Swansea	-0.736842	38	8	21	9	33
19	West Brom	-0.657895	38	6	19	13	31

3 1c

[9]:
```
df2 = output.sort_values(['Div', 'Avg PD'], ascending=[True, False])
df2.groupby('Div').head(3).reset_index(drop=1)
```

[9]:

	Div	Team	Avg PD	Games Played	Wins	Losses	Draws
0	Bundesliga	Bayern Munich	1.882353	34	27	4	3
1	Bundesliga	Hoffenheim	0.529412	34	15	9	10
2	Bundesliga	Dortmund	0.500000	34	15	9	10
3	EPL	Man City	2.078947	38	32	2	4
4	EPL	Liverpool	1.210526	38	21	5	12
5	EPL	Man United	1.052632	38	25	7	6
6	La_Liga	Barcelona	1.842105	38	28	1	9
7	La_Liga	Real Madrid	1.315789	38	22	6	10
8	La_Liga	Ath Madrid	0.947368	38	23	5	10
9	Ligue_1	Paris SG	2.078947	38	29	3	6
10	Ligue_1	Lyon	1.157895	38	23	6	9
11	Ligue_1	Monaco	1.052632	38	24	6	8
12	Serie_A	Juventus	1.631579	38	30	3	5
13	Serie_A	Napoli	1.263158	38	28	3	7
14	Serie_A	Lazio	1.052632	38	21	8	9

DS3001_HW1_Q2

February 11, 2021

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import math
```

[2]:
```
df = pd.read_csv('soccer17.csv')
```

[3]:
```
df['All_Goals'] = df.FTHG + df.FTAG
df['H1_Goals'] = df.HTHG + df.HTAG
df['H2_Goals'] = df['All_Goals'] - df['H1_Goals']
X = np.linspace(0, 10)
```

1 2a

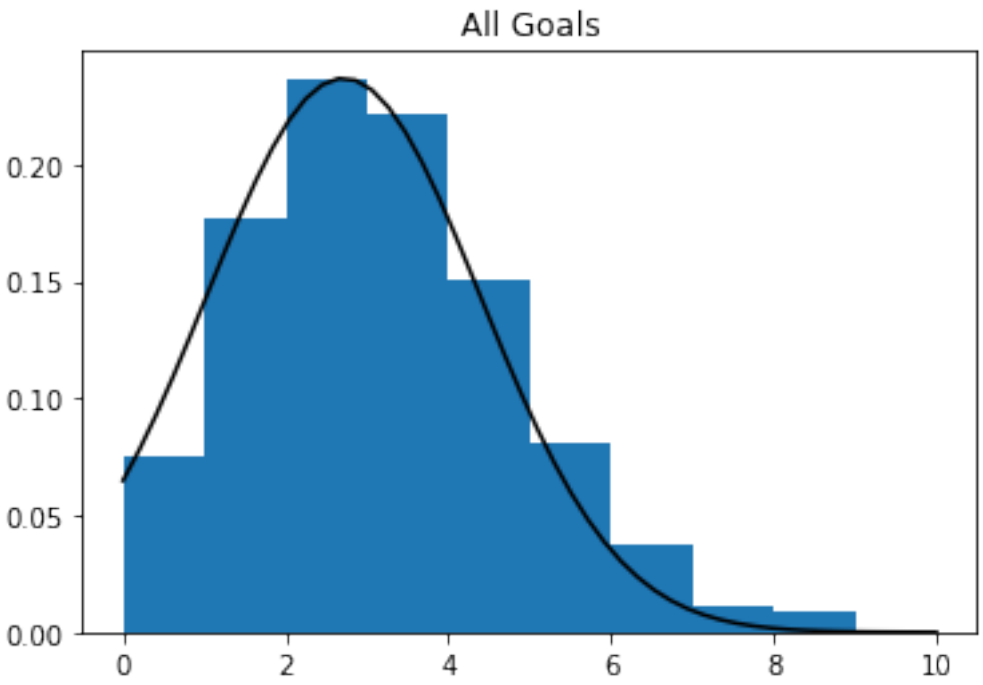
1.1 i. Model Selection

$A_1, ..., A_{7304} \overset{iid}{\sim} \mathcal{N}(\mu_A, \sigma_A^2)$ where A_i represents all goals scored in a single game i

1.2 ii. Histogram

[4]:
```
mu, sigma = norm.fit(df['All_Goals'])
plt.hist(df['All_Goals'], bins=np.arange(0, 10), density=True)
plt.plot(X, norm.pdf(X, mu, sigma), color='black')
plt.title('All Goals')
```

[4]:
```
Text(0.5, 1.0, 'All Goals')
```



2 2b

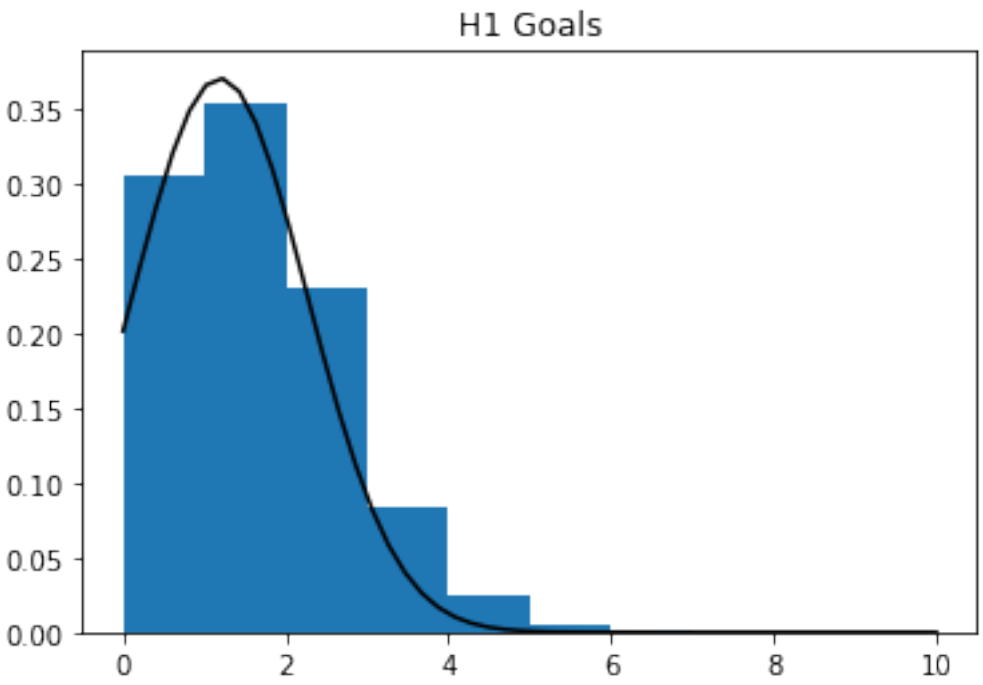
2.1 i. Model Selection

$H_1, ..., H_{7304} \overset{iid}{\sim} \mathcal{N}(\mu_H, \sigma_H^2)$ where H_i represents all first half goals scored in a single game i

2.2 ii. Histogram

[5]:
```
mu, sigma = norm.fit(df['H1_Goals'])
plt.hist(df['H1_Goals'], bins=np.arange(0, 10), density=True)
plt.plot(X, norm.pdf(X, mu, sigma), color='black' )
plt.title('H1 Goals')
```

[5]:
```
Text(0.5, 1.0, 'H1 Goals')
```



3 2c

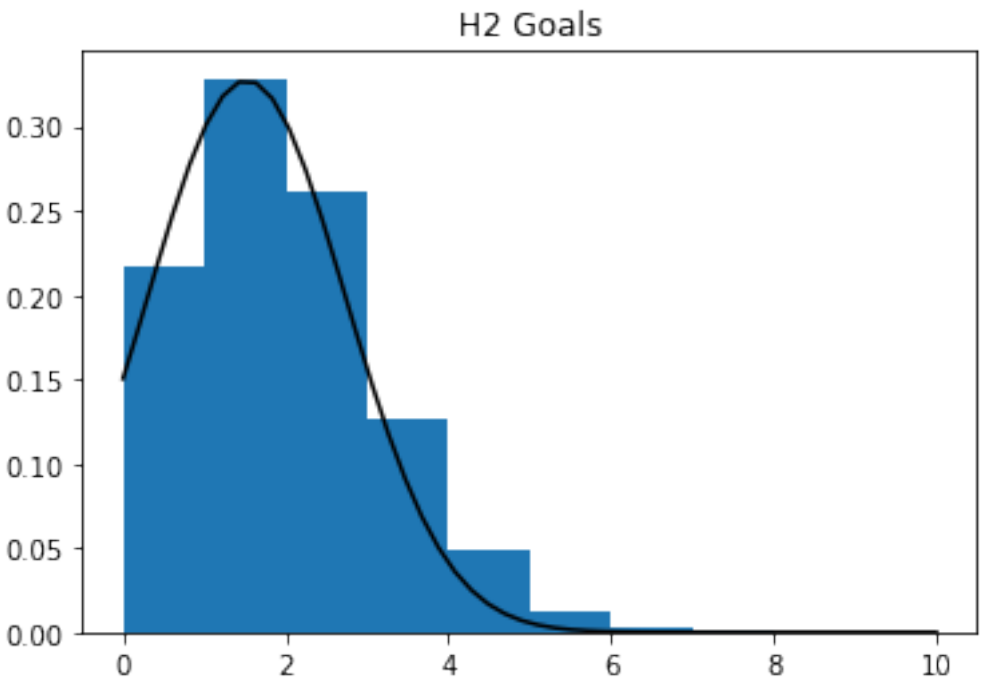
3.1 i. Model Selection

$I_1, ..., I_{7304} \overset{iid}{\sim} \mathcal{N}(\mu_I, \sigma_I^2)$ where I_i represents all second half goals scored in a single game i

3.2 ii. Histogram

[6]:
```
mu, sigma = norm.fit(df['H2_Goals'])
plt.hist(df.H2_Goals, bins=np.arange(0, 10), density=True)
plt.plot(X, norm.pdf(X, mu, sigma), color='black')
plt.title('H2 Goals')
```

[6]:
```
Text(0.5, 1.0, 'H2 Goals')
```



4 2d

[7]:
```
df2 = pd.DataFrame()
df2['Num_Games'] = df.groupby('Div').count()['All_Goals']
df2['Avg_Goals'] = df.groupby('Div').mean()['All_Goals']

df2.reset_index().sort_values(by='Avg_Goals', ascending=False)
```

[7]:

	Div	Num_Games	Avg_Goals
0	Bundesliga	1224	2.811275
2	La_Liga	1520	2.759211
4	Serie_A	1520	2.725658
1	EPL	1520	2.686184
3	Ligue_1	1520	2.588158

5 2e

[8]:
```
epsilon = 0.04
df4 = df.loc[df.All_Goals == 4]
fgg = df4.loc[(np.abs(df4.pH - df4.pA) < epsilon)]
```

[9]:
```
prob_draw =  math.factorial(4)/(math.factorial(2) * math.factorial(2)) * (0.
↪5)**2 * (0.5)**2
exp_draws = len(fgg) * prob_draw
draws = len(fgg.loc[fgg.FTHG == fgg.FTAG])
sigma = np.sqrt(0.375*0.625*len(fgg))

t = (draws-exp_draws)/sigma
p = 1 - norm.cdf(t)
print('p-value = {}'.format(p))
```

```
p-value = 0.007348710885011767
```

Our null hypothesis states that there is no "comeback tendency" within the dataset provided

Using an epsilon of $\epsilon = 0.04$, we were able to obtain a statistically significant p-value meaning that we reject the null hypothesis. This means that there is empirical evidence that suggests that a "comeback tendency" exists.