UCL

School of

Management

MSIN0143 Programming for Business Analytics

Group Submission:

An Exploratory Data Analysis of English Premier League (EPL) Match Data

Student : Student Number

Teddy Favre-Gilly : 16010338

John Oikonomou : 16052919

Marwan Farha : 16042374

Siyu Liu : 15040876

Red Jalleh : 16005480

EPL Data - Exploratory Analysis

Contents:

---

1 Business context and management

1.1 Frame the Business Problem:

The objective of this exploratory data analysis is to improve our prediction of outright match winners in football games in the English Premier League (EPL).

We wish to identify which match events and stats (e.g. Yellow Cards, Shot Accuracy) most affect football results, in order to place bets during the live games.

Betting odds are calculated according to the public's opinion; if one team has received more bets than the other, their odds will be lower. We believe that by looking at statistical reasoning, we can outperform the betting odds.

1.2 Project Management

The team adopted an agile methodology in creating this analysis: Each team member was assigned 4 variables to analyse:

- The first week, the team presented their 4 variables' findings to the rest of the team
- The second week, each member iterated on their analysis, and plotted their findings graphically
- In the third week, the members wrote about what their analysis represented and what about it was interesting and relevant

This task breakdown was visually represented on a Jira Kanban board, where team members could place their work packages into 'To-do', 'started' and 'complete' stages.

...(more about project management in Jira)

1.3 Solving the Problem Manually

Solving the problem manually, one would not be able to consider the 13 years of match data presented in csv form. Instead, one would rely on expert knowledge. Mark Lawrenson, the BBC's expert football pundit, gets the Full_Time Winner correct 38% of the time (Lawrenson, 2016) whereas betting companies calculate the correct winner 53% of the time (Hosseini, 2018).

Manually one would certainly concentrate on recent form to predict game outcome, but that would fail to consider in-game statistics.

---

2. Data ETL:

First we imported the libraries we needed and the dataset 'as epl_data':

2.1 Data Import

In [97]:

```
#import libraries
import pandas as pd
import numpy as np
import itertools
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns
from sklearn.linear_model import LinearRegression #
#import dataset
epl_data = pd.read_csv('/Users/teddyFG/Documents/UCL/ML/EPL/epl-training.csv')
epl_data.head()
```

Out[97]:

	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	Referee	...	HST
0	13-Aug-05	Aston Villa	Bolton	2	2	D	2	2	D	M Riley	...	
1	13-Aug-05	Everton	Man United	0	2	A	0	1	A	G Poll	...	
2	13-Aug-05	Fulham	Birmingham	0	0	D	0	0	D	R Styles	...	
3	13-Aug-05	Man City	West Brom	0	0	D	0	0	D	C Foy	...	
4	13-Aug-05	Middlesbrough	Liverpool	0	0	D	0	0	D	HalseyM	...	

5 rows × 22 columns

2.2 Data Cleaning:

Here we cleaned up the data.

1. I replaced the columns with more intuitive variable names
2. I made the FTResult (Full Time Result) column either:
   - 1 for home victory
   - 0 for draw
   - 1 for away victory

...so that we could plot it on the correlation matrix

In [98]:

```
#first declare the dictionary of values di = {'H':
1, 'D': 0, 'A': -1} #replace "FTR" with dictionary
value clean_epl_data = epl_data.replace({"FTR": di})
clean_epl_data = clean_epl_data.replace({"HTR": di})
#replace column names in new(clean) dataset
clean_epl_data.columns = ['Date','HomeTeam','AwayTeam', 'FT_HomeGls','FT_AwayGl
s','FTResult','HT_HomeGls','HT_AwayGls','HT_Res','Ref','HomeShots','AwayShots',
'HomeShotsTarget','AwayShotsTarget','HomeFouls','AwayFouls','HomeCorners','AwayC
orners','HomeYellow','AwayYellow','HomeRed','AwayRed'] clean_epl_data.head()
```

Out[98]:

	Date	HomeTeam	AwayTeam	FT_HomeGls	FT_AwayGls	FTResult	HT_HomeGls	HT_Aw
0	13-Aug-05	Aston Villa	Bolton	2	2	0	2	
1	13-Aug-05	Everton	Man United	0	2	-1	0	
2	13-Aug-05	Fulham	Birmingham	0	0	0	0	
3	13-Aug-05	Man City	West Brom	0	0	0	0	
4	13-Aug-05	Middlesbrough	Liverpool	0	0	0	0	

5 rows × 22 columns

Replacing referee names with numbers

I then want to replace Referee names with unique ID numbers from 1 to 44

In [99]:

```
#create array of referee names as the key (to be replaced)
refKeys=clean_epl_data.Ref.unique()
#create numbered array of referees as the values (to replace)
refValues=np.arange(1,len(refKeys)+1)
```

```
#create a dictionary assigning a number for each referee  refDict=
dict(zip(refKeys, refValues))
#replace referee names using the key,value pair dictionary
clean_epl_data = clean_epl_data.replace({'Ref': refDict})
clean_epl_data.head()  Out[99]:
```

	Date	HomeTeam	AwayTeam	FT_HomeGls	FT_AwayGls	FTResult	HT_HomeGls	HT_Aw
0	13-Aug-05	Aston Villa	Bolton	2	2	0	2	
1	13-Aug-05	Everton	Man United	0	2	-1	0	
2	13-Aug-05	Fulham	Birmingham	0	0	0	0	
3	13-Aug-05	Man City	West Brom	0	0	0	0	
4	13-Aug-05	Middlesbrough	Liverpool	0	0	0	0	

5 rows × 22 columns

3. Exploratory Data Anlysis

3.1.1. Correlation Matrix

The first exploratory analysis to conduct will be to plot the correlation matrix of all our variables

The matrix reveals that FT_Results (full time result) is most correlated to:

- Goals (Full Time, Half Time) (0.61, 0.43)
- Shots (Shots, Shots on target) (0.2, 0.29)
- Yellow cards (Home, Away) (0.11, 0.02)
- Red cards (Home, Away) (0.13, 0.09)

In [18]:

```
CORR=clean_epl_data.corr()
CORR
```

Out[18]:

	FT_HomeGls	FT_AwayGls	FTResult	HT_HomeGls	HT_AwayGls	HT_Res

FT_HomeGls	1.000000	-0.069588	0.619898	0.685735	-0.045099	0.469714
FT_AwayGls	-0.069588	1.000000	-0.635873	-0.062301	0.677013	-0.452340
FTResult	0.619898	-0.635873	1.000000	0.436912	-0.431214	0.607989
HT_HomeGls	0.685735	-0.062301	0.436912	1.000000	-0.048048	0.675412
HT_AwayGls	-0.045099	0.677013	-0.431214	-0.048048	1.000000	-0.636888
HT_Res	0.469714	-0.452340	0.607989	0.675412	-0.636888	1.000000
Ref	0.011137	0.016059	-0.002418	-0.006382	0.015189	-0.011044
HomeShots	0.273781	-0.116926	0.209724	0.105284	-0.039928	0.065600
AwayShots	-0.132696	0.317108	-0.249396	-0.035559	0.153713	-0.101376
HomeShotsTarget	0.410706	-0.104575	0.294095	0.234008	-0.057396	0.170363
AwayShotsTarget	-0.096834	0.430870	-0.299453	-0.040841	0.264184	-0.175840
HomeFouls	-0.074602	0.022984	-0.044934	-0.008293	0.005755	0.004295
AwayFouls	-0.025172	-0.037895	0.039496	-0.012858	-0.000644	-0.007233
HomeCorners	0.031122	-0.070180	0.044409	-0.061773	-0.025216	-0.046449
AwayCorners	-0.065836	0.039028	-0.037685	0.012137	-0.031819	0.055028
HomeYellow	-0.103312	0.123430	-0.117888	-0.078629	0.106767	-0.116270
AwayYellow	0.005265	-0.015149	0.023018	0.000279	-0.000069	-0.001662
HomeRed	-0.079627	0.113487	-0.137097	-0.036020	0.079623	-0.067243
AwayRed	0.083077	-0.070977	0.098309	0.036290	-0.022510	0.037759

In order to see which values are highly correlated with FTResult, we set a threshold of 0.1 (10%) In [19]:

```
CORR[abs(CORR)>0.1]
```

Out[19]:

	FT_HomeGls	FT_AwayGls	FTResult	HT_HomeGls	HT_AwayGls	HT_Res
FT_HomeGls	1.000000	NaN	0.619898	0.685735	NaN	0.469714
FT_AwayGls	NaN	1.000000	-0.635873	NaN	0.677013	-0.452340
FTResult	0.619898	-0.635873	1.000000	0.436912	-0.431214	0.607989
HT_HomeGls	0.685735	NaN	0.436912	1.000000	NaN	0.675412
HT_AwayGls	NaN	0.677013	-0.431214	NaN	1.000000	-0.636888
HT_Res	0.469714	-0.452340	0.607989	0.675412	-0.636888	1.000000
Ref	NaN	NaN	NaN	NaN	NaN	NaN
HomeShots	0.273781	-0.116926	0.209724	0.105284	NaN	NaN
AwayShots	-0.132696	0.317108	-0.249396	NaN	0.153713	-0.101376
HomeShotsTarget	0.410706	-0.104575	0.294095	0.234008	NaN	0.170363
AwayShotsTarget	NaN	0.430870	-0.299453	NaN	0.264184	-0.175840
HomeFouls	NaN	NaN	NaN	NaN	NaN	NaN
AwayFouls	NaN	NaN	NaN	NaN	NaN	NaN
HomeCorners	NaN	NaN	NaN	NaN	NaN	NaN

AwayCorners	NaN	NaN	NaN	NaN	NaN	NaN
HomeYellow	-0.103312	0.123430	-0.117888	NaN	0.106767	-0.116270
AwayYellow	NaN	NaN	NaN	NaN	NaN	NaN
HomeRed	NaN	0.113487	-0.137097	NaN	NaN	NaN
AwayRed	NaN	NaN	NaN	NaN	NaN	NaN

3.1.2. Feature Histograms & Descriptive Statistics

The feature historgrams reveal the variables' normal distributions

- Corners (Away, Home)

```
Mean (4.8, 6.1)
Std (2.74, 3.12)
```

- Fouls (Away, Home)

```
 Mean (11.6, 11.1)
Std (3.78, 3.65)
```

- Shots (Away, Home)

```
 Mean (10.4, 13.8)
Std (4.61, 5.34)
```

- ShotsTarget (Away, Home)

```
 Mean (5.0, 6.38)
Std (2.90, 3.47)
```

- Yellows (Away, Home)

```
Mean (1.79,1.43)
Std (1.29, 1.19)
```

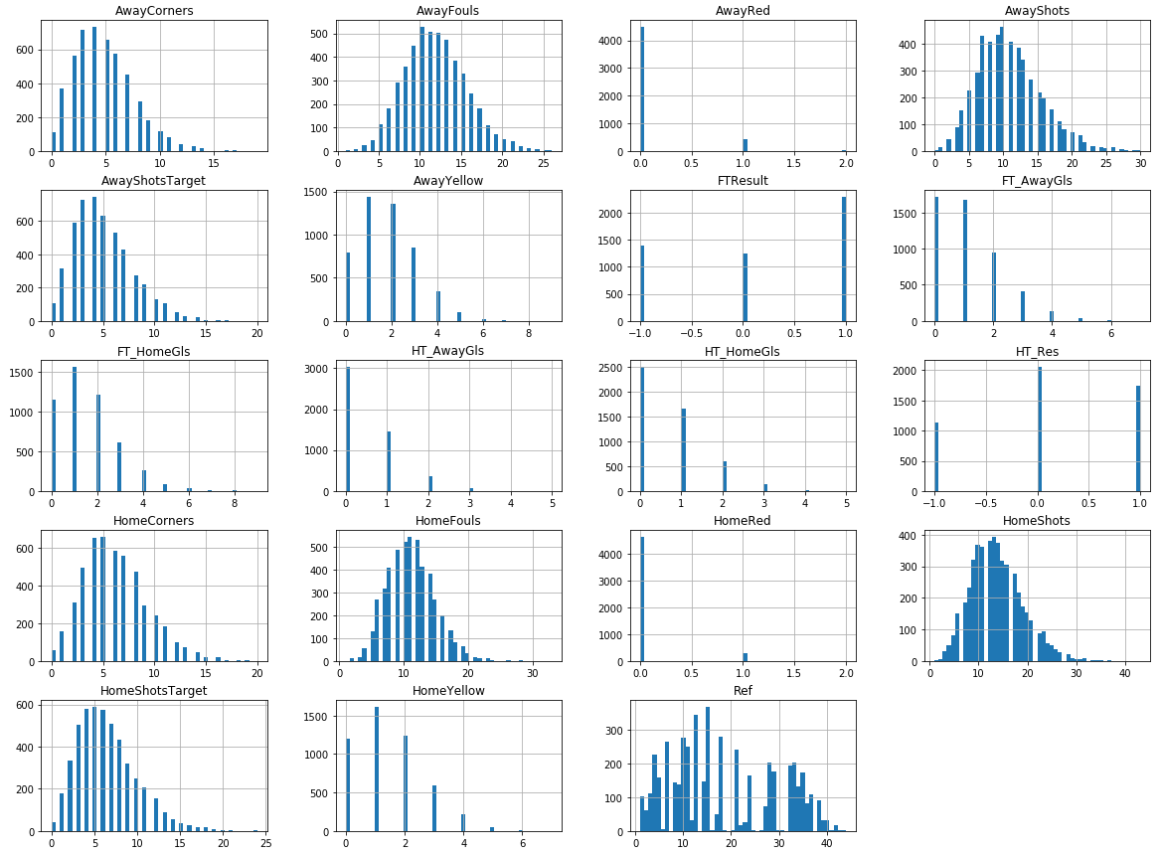
- FullTimeGoals (Away, Home)

```
 Mean (1.13, 1.53)
Std (1.13, 1.31)
```

There emerges a trend that the home team have an advantage and have favorable statistics:

In [8]:

```
clean_epl_data.hist(bins=50, figsize=(20,15))
plt.show()
```



Out[8]:

```
4.804048582995952
```

3.1.3. Percentage Calculation for Total Shots and Shots on Target

The calculation below shows the shot accuracy rate of the home (46.2%) and away team (45.7%) respectively.

Teams had a (.5%) percentage point increase in shooting accuracy when playing at home.

In [40]:

```
TotalHomeShots = clean_epl_data['HomeShots'].sum()
TotalHomeShotsTarget=clean_epl_data['HomeShotsTarget'].sum()
print('HomeShotAccuracy: ' + str((TotalHomeShotsTarget/TotalHomeShots).round(3
)))
```

```
HomeShotAccuracy: 0.462
```

In [41]:

```
TotalAwayShots = clean_epl_data['AwayShots'].sum()
TotalAwayShotsTarget=clean_epl_data['AwayShotsTarget'].sum()
print('AwayShotAccuracy: ' + str((TotalAwayShotsTarget/TotalAwayShots).round(3
)))
```

```
AwayShotAccuracy: 0.457
```

3.2. Goal Analysis

3.2.1. Half-Time Goal Difference (HTGD)

We wish to find the relationship between the goal difference at half time and the FT Result (winning team).

To calculate HTGD, we take HTAwayGls away from HTHomeGls.

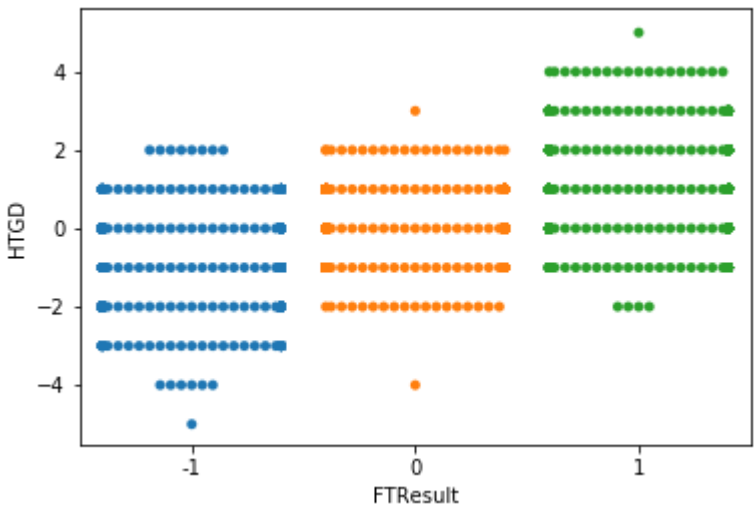
In [44]:

```
clean_epl_data['HTGD'] = clean_epl_data['HT_HomeGls'] - clean_epl_data['HT_AwayG
ls']
#HTGD is Half Time Goal Difference, HTHG - HTAG.
```

In [45]:

```
htgd = clean_epl_data['HTGD'] ftr =
clean_epl_data['FTResult'] sns.swarmplot(x=ftr, y=
htgd , data=clean_epl_data) Out[45]:
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a19a342b0>
```

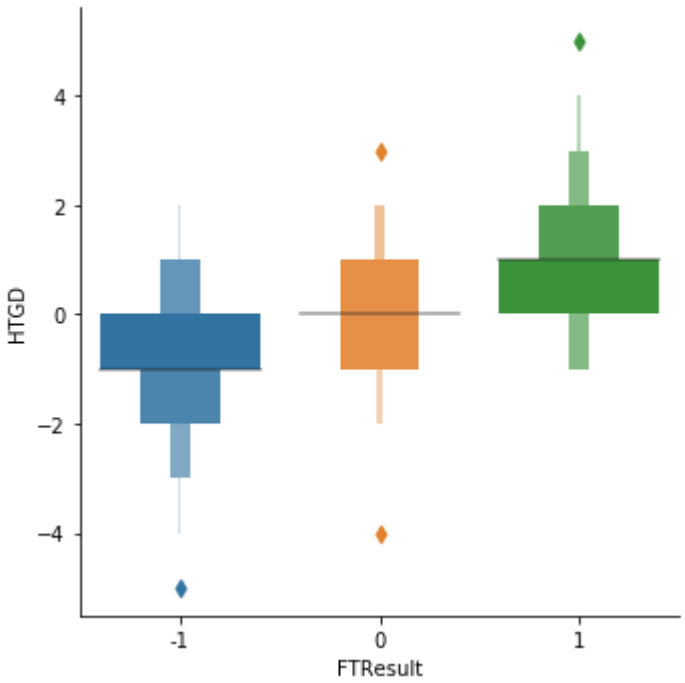


The Distribution of HTGD against the FTR is not clearly represented on the scatter graph, as it is discrete data.

A boxplot was deemed a better way of representing this data:

In [47]:
```
sns.catplot(x="FTResult", y="HTGD", kind="boxen",
data=clean_epl_data)
```

Out[47]: <seaborn.axisgrid.FacetGrid at 0x1a19a40e10>



As seen on the boxplot above:

When FTResult is 0 (Draw), the HTGD has median 0, however the interquartile range (IQR) is 2.

When FTResult is -1 (AwayWin), the HTGD has median -1 and IQR 2, meaning the Away team is already winning at half in 75% of cases.

When FTResult is 1 (HomeWin), the HTGD has median 1 and IQR 2, meaning the Home team is already winning at half in 75% of cases.

The team that is winning at half time (Away, Home) has a (70%, 82%) high chance of winning the game. ~calculation below

The more goals the team is leading by at half time, the more likely they are to win the game.

This boxplot chosen is ideal for representing this information as it also shows the density of values within the quartile; for example, in the graph above, with FTResult = -1, it can be seen that there is a higher number of values between 0 and -1 HTGD than between 0 and 1 HTGD.

In [63]:

```
awayHTWinning = clean_epl_data[clean_epl_data['HT_Res']==-1]
awayHTconversion = len(awayHTWinning[awayHTWinning['FTResult']==-1])/len(awayHTW
inning)

print(awayHTconversion)

homeHTWinning = clean_epl_data[clean_epl_data['HT_Res']==1]
homeHTconversion = len(homeHTWinning[homeHTWinning['FTResult']==1])/len(homeHTWi
nning)

print(homeHTconversion)
```

```
0.7028218694885362
0.8180252583237658
```

3.2.2. Goals & Win Rate

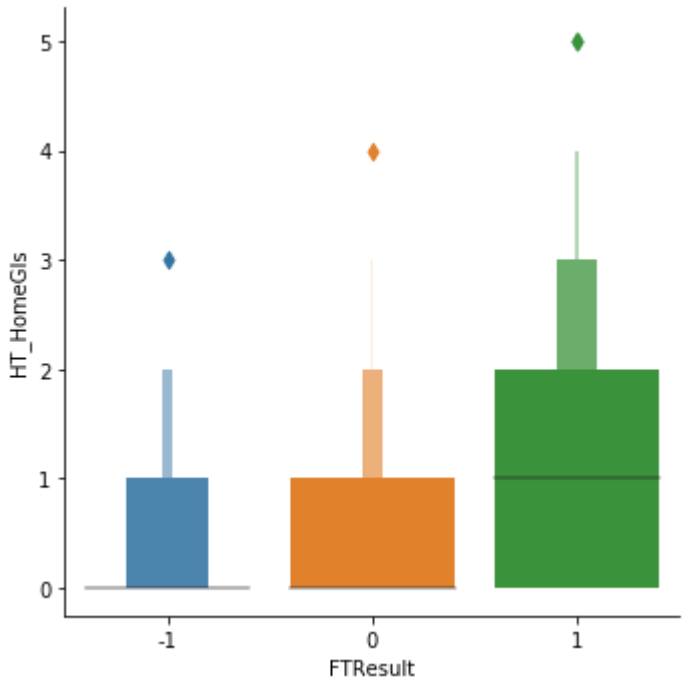
Below the box plot is plottedfor the HTHomeGls and HTAwayGls, each against FTResult. This is to gain a clearer understanding of the influence of the half time scoreline on the home and away team respectively.

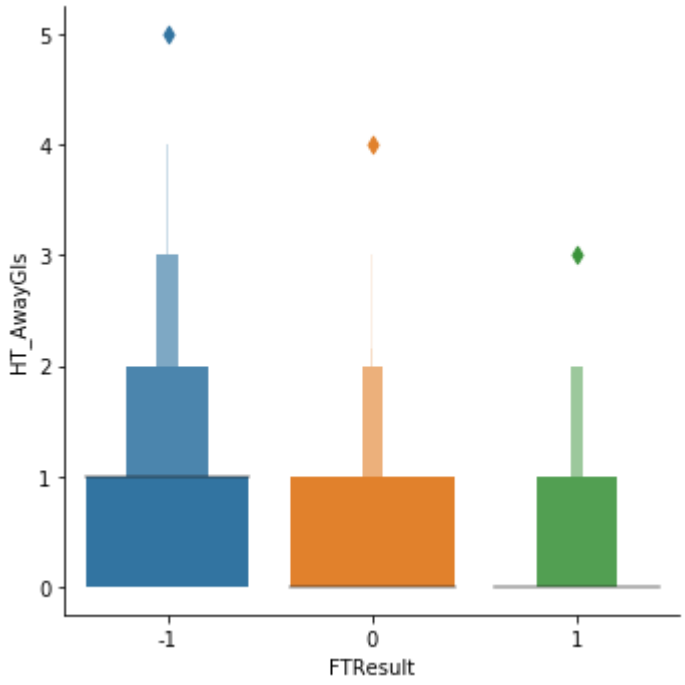
There is a strong positive correlation between the number of goals a team scores and their chance of winning.

In [28]:

```
sns.catplot(x="FTResult", y="HT_HomeGls", kind="boxen", data=clean_epl_data)
sns.catplot(x="FTResult", y="HT_AwayGls", kind="boxen", data=clean_epl_data)
```

Out[28]: <seaborn.axisgrid.FacetGrid at 0x1a19486390>





3.2.3. ShotAccuracy & Win-Rate

To find a stronger correlation between shots and Win-Rate, the box plot below represents ShotAccuracy (Shots on Target/Total Shots) versus Win-Rate

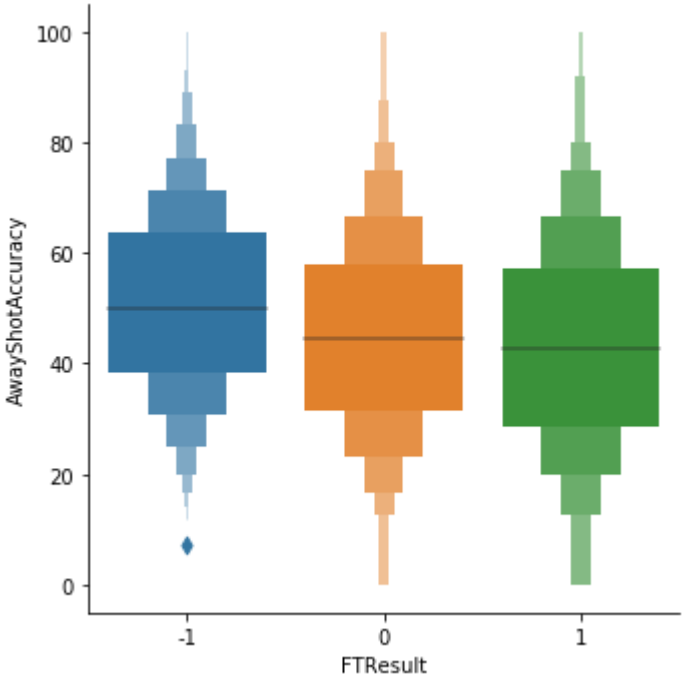
There is a strong positive correlation between a team's ShotAccuracy and Win-Rate

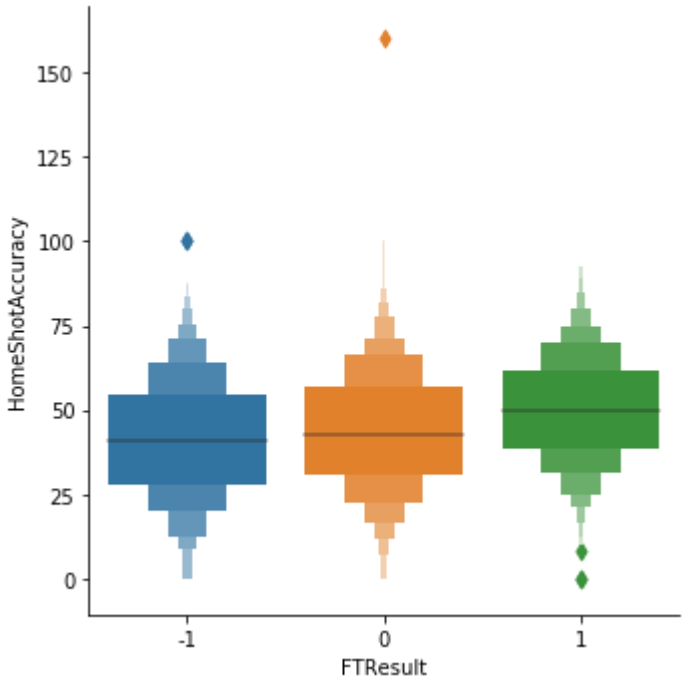
In [35]:

```
clean_epl_data['HomeShotAccuracy'] = 100*(clean_epl_data['HomeShotsTarget']/clea
n_epl_data['HomeShots'])
#HSA is home shot accuracy (as a percentage)
clean_epl_data['AwayShotAccuracy'] = 100*(clean_epl_data['AwayShotsTarget']/clea
n_epl_data['AwayShots'])
#ASA is away shot accuracy (as a percentage)
sns.catplot(x='FTResult', y='AwayShotAccuracy',kind='boxen', data=clean_epl_data
)
sns.catplot(x='FTResult', y='HomeShotAccuracy',kind='boxen', data=clean_epl_data
)
```

Out[35]:

```
<seaborn.axisgrid.FacetGrid at 0x1a19bb40b8>
```





3.3. Shot Analysis

3.2.2. Shots & Win-Rate

3.2.3. Shots & Cards

3.2.4.1. Home (Shots & Cards)

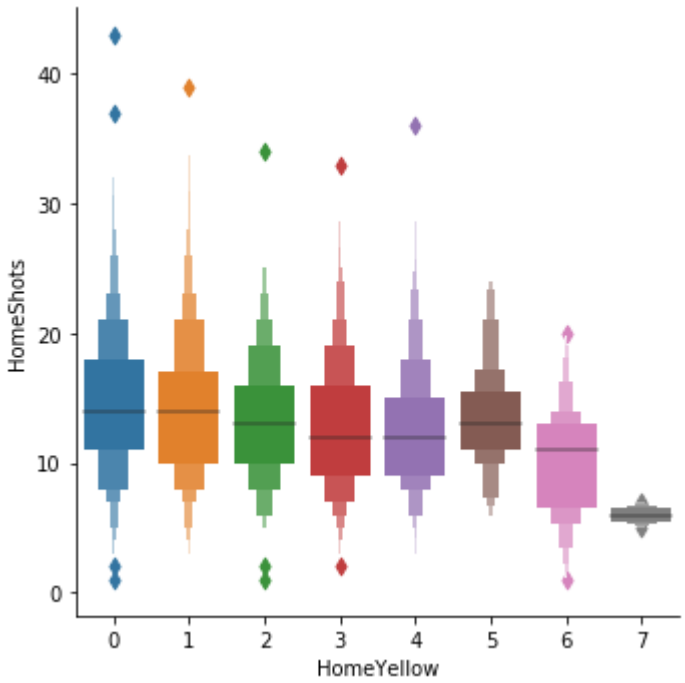
The box plot below shows the relationship between HomeYellow and the number of HomeShots.

There is a wek negative correlation between the HomeTeam's number of Yellow cards and Win-Rate. Teams who receive more yellow cards decrease their probability of winning. This could be because yellow cards accumulate and become Red Cards (2 yellow cards for one player) and the player gets sent off.

```
In [66]: sns.catplot(x="HomeYellow", y="HomeShots", kind="boxen",
data=clean_epl_data)
```

Out[66]:

```
<seaborn.axisgrid.FacetGrid at 0x1a19cf8a90>
```



This also could be because the teams become more cautious when they accumulate cards (Lawrenson, 2018). We sought to prove this qualitative data.

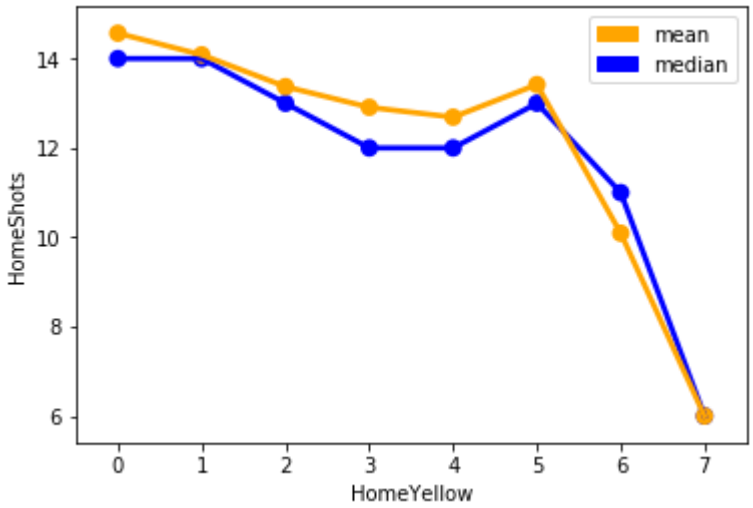
Below, the linegraph shows how the median number of HomeShots per HomeYellow card decreases, meaning the teams shoot less when they receive yellow cards. This supports Lawrenson's claim that teams play more conservatively when they receive more yellow cards.

In [32]:

```
f, ax = plt.subplots(1, 1)

sns.pointplot(x='HomeYellow', y='HomeShots', data=clean_epl_data.groupby('HomeYellow', as_index=False).median(), color='blue', label='median')
sns.pointplot(x='HomeYellow', y='HomeShots', data=clean_epl_data.groupby('HomeYellow', as_index=False).mean(), color='orange', label='mean')

mean = mpatches.Patch(color='orange', label='mean') median
= mpatches.Patch(color='blue', label='median')
plt.legend(handles=[mean, median])

#mean is orange and median is blue.
```

Out[32]:

```
<matplotlib.legend.Legend at 0x1a19acc7b8>
```



It is important to note that there were only two datapoints for 7 HomeYellow cards

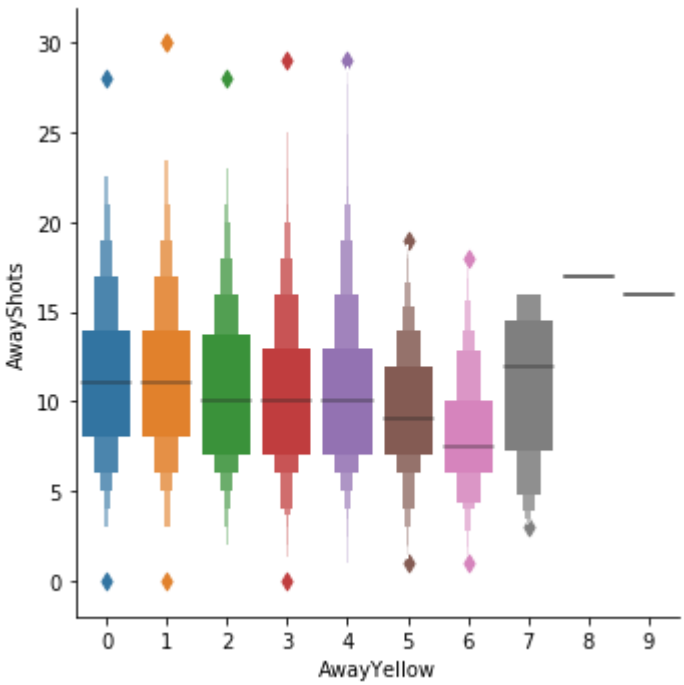
3.2.4.2. Away (Shots & Cards)

Below I will perform the same analysis but on the Away Teams stats.

```
In [33]: sns.catplot(x="AwayYellow", y="AwayShots", kind="boxen",
data=clean_epl_data)
```

Out[33]:

```
<seaborn.axisgrid.FacetGrid at 0x1a19a5aef0>
```

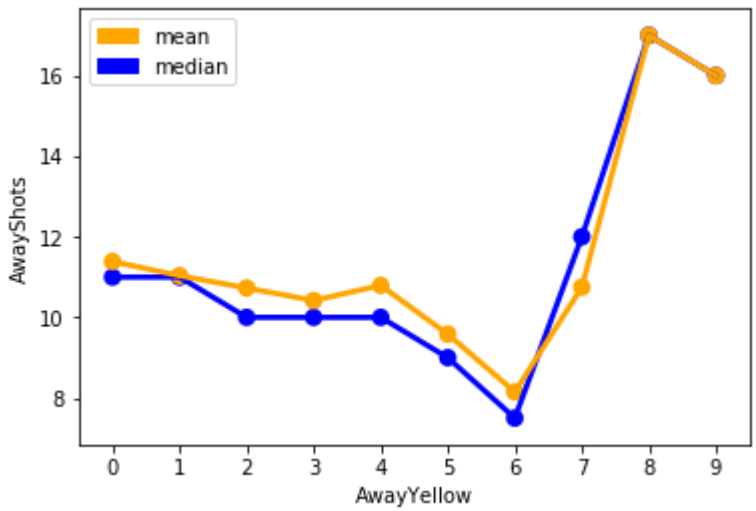


In [34]:

```
f, ax = plt.subplots(1, 1)

sns.pointplot(x='AwayYellow', y='AwayShots', data=clean_epl_data.groupby('AwayYellow', as_index=False).median(), color='blue', label='median')
sns.pointplot(x='AwayYellow', y='AwayShots', data=clean_epl_data.groupby('AwayYellow', as_index=False).mean(), color='orange', label='mean')

'''below is a line plot of the medians (and means in second graph below)
of each box plot in the graph above.'''

mean = mpatches.Patch(color='orange', label='mean') median
= mpatches.Patch(color='blue', label='median')
plt.legend(handles=[mean, median])



#mean is orange and median is blue.
```

Out[34]:

```
<matplotlib.legend.Legend at 0x1a19a90080>
```



Whereas with the HomeTeam, the mean and median number of shots significantly decreases at 6 HomeYellow cards, the number of AwayShots increases rapidly when the AwayTeam surpasses 6 AwayYellow cards.

This, combined with our data on number of AwayYellows and likelihood of each FTResult, can provide insight into the probability of the FT game Result, given the occurance of HomeYellow and AwayYellow cards.

3.4. Card Analysis

3.4.1. Cards & Win Rate

Given our analysis of how card affect shot-rates, we sought to evaluate how cards affected Win-Rate.

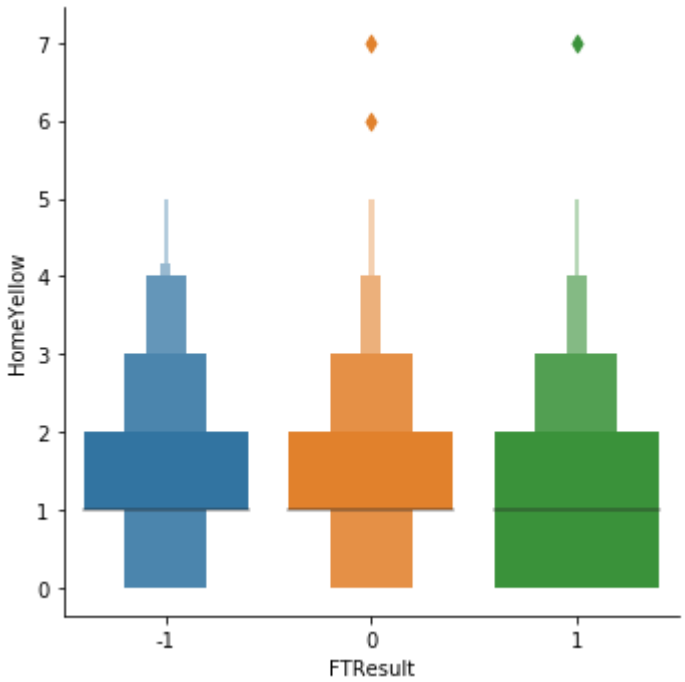
The two below box plots revealed weak negative correlation between a team's yellow cards number and Win-Rate.

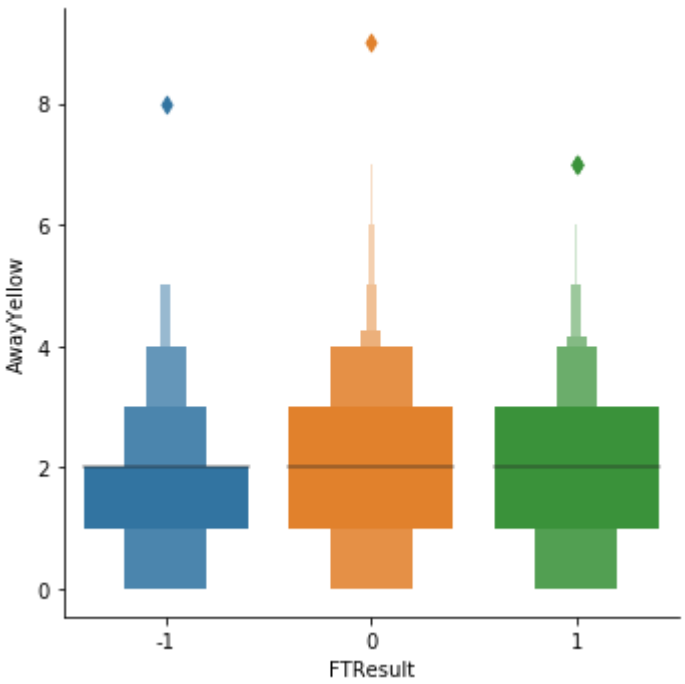
In [30]:

```
sns.catplot(x="FTResult", y="HomeYellow", kind="boxen", data=clean_epl_data)
sns.catplot(x="FTResult", y="AwayYellow", kind="boxen", data=clean_epl_data)
```

Out[30]: <seaborn.axisgrid.FacetGrid at

0x1a19695d68>





3.4.2. Cards Away & Home

Please run the following cell before the analysis can begin:

In [84]:

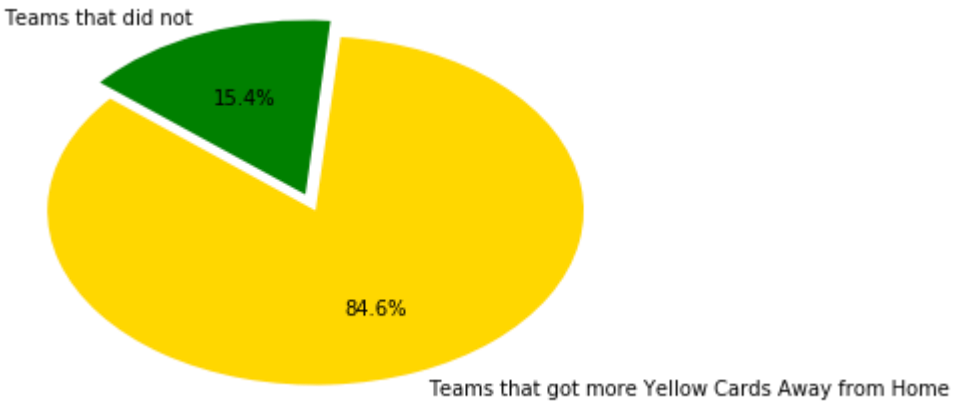
```
# Finding out how many yellow cards each team received when they're home team.
home_yellows_total = {}
home_teams = clean_epl_data['HomeTeam'].unique()

for team in home_teams:
    home_yellows_total[team] = clean_epl_data[clean_epl_data['HomeTeam'] == team
]['HomeYellow'].sum()

# Finding out how many yellow cards each team received when they're away team.
away_yellows_total = {}
away_teams = clean_epl_data['AwayTeam'].unique()

for team in away_teams:
    away_yellows_total[team] = clean_epl_data[clean_epl_data['AwayTeam'] == team
]['AwayYellow'].sum()

# The following section will compare each team to see whether they get more yell
ows home or away.
# We could assume that if they get more yellows away then home, the team likes t
o play aggresive to get the win. team_agression_yellow = {}

for team in home_teams:    if (home_yellows_total[team] >
 away_yellows_total[team]):
 team_agression_yellow[team] = False
            if (home_yellows_total[team] <
 away_yellows_total[team]):
 team_agression_yellow[team] = True
            else:
 team_agression_yellow[team] = 0

# Finding out how many red cards each team received when they're home team.
home_reds_total = {}

for team in home_teams:    home_reds_total[team] =
 clean_epl_data[clean_epl_data['HomeTeam'] == team][
'HomeRed'].sum()
# Finding out how many red cards each team received when they're away team.
away_reds_total = {}

for team in away_teams:    away_reds_total[team] =
 clean_epl_data[clean_epl_data['AwayTeam'] == team][
'AwayRed'].sum()
    team_agression_red
= {}

for team in home_teams:    if (home_reds_total[team] >
 away_reds_total[team]):
 team_agression_red[team] = False
            if (home_reds_total[team] <
 away_reds_total[team]):
 team_agression_red[team] = True
            else:
 team_agression_red[team] = 0
```

The below calculation shows whether teams recieved more yellow cards away from home than at home. Out of the 39 teams that have played in since 2005, 33 (84.6%) have received more cards away than home.

This could either be because the teams play more aggressively away from home, or could be because the referees are influenced by home fans.

In [85]:

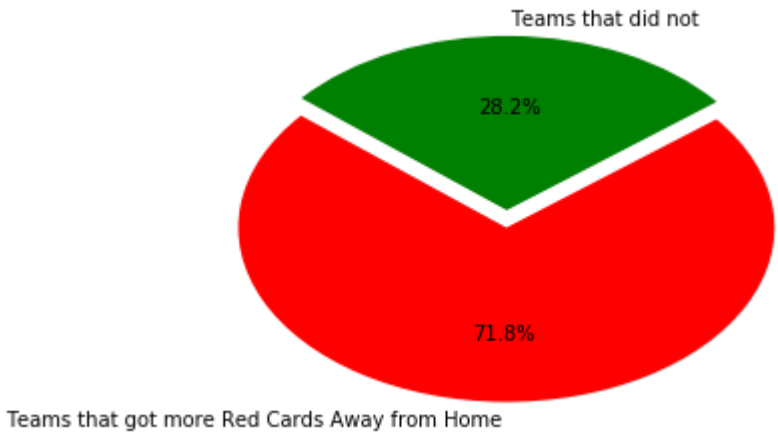
```
teams = len(clean_epl_data['HomeTeam'].unique())
aggyteams = sum(team_agression_yellow.values())

labels = "Teams that got more Yellow Cards Away from Home", "Teams that did not"
sizes = [aggyteams, (teams-aggyteams)]
colors = ['gold', 'green']
explode = (0,0.1)

plt.pie(sizes, labels = labels, colors = colors, autopct= '%1.1f%%', startangle
= 140, explode = explode)
plt.show()
```



The same is then shown for Red cards: 28 out of 39 (71.8%) unique teams received more red cards away from home than at home.

In [86]:

```
aggyred= sum(team_agression_red.values())

labels = "Teams that got more Red Cards Away from Home", "Teams that did not"
sizes = [aggyred, (teams-aggyred)]
colors = ['red', 'green']
explode = (0,0.1)

plt.pie(sizes, labels = labels, colors = colors, autopct= '%1.1f%%', startangle
= 140, explode = explode)
plt.show()
```



Once it was discovered that teams get more cards away than home, we wanted to discover whether cards affected full time result. This would show us how much the referee's Away team bias, would disadvantage teams playing away The boxplot below revealed there was a weak correlation between Yellow Card rate a win rate. The more the Away Team received yellow cards, the higher the probability of the Home team winning.

When the home team won, the away team had on average 1.78 Yellow Cards, when the away team won, they had lower on average of 1.68 Yellow Cards.

3.4.2. Cards Away & Home

Once it was discovered that teams get more cards away than home, we wanted to discover whether cards affected full time result.

This would show us how much the referee's Away team bias, would disadvantage the AwayTeam.

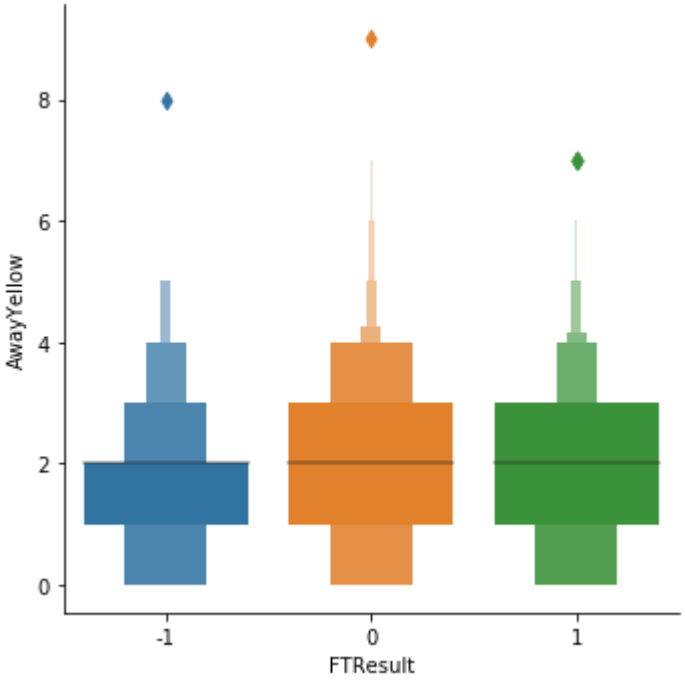
The boxplot below revealed there was a weak correlation between Yellow Card rate a win rate. The more the Away Team received yellow cards, the higher the probability of them losing.

In [87]:

```
sns.catplot(x="FTResult", y="AwayYellow", kind="boxen",
data=clean_epl_data)
```

Out[87]:

```
<seaborn.axisgrid.FacetGrid at 0x1a1b407c18>
```



When the home team won, the away team had on average 1.78 Yellow Cards, when the away team won, they had lower on average of 1.68 Yellow Cards. ~calculation below

In [88]:

```
homewins = clean_epl_data[clean_epl_data['FTResult']==1]
print(homewins['AwayYellow'].mean())

awaywins = clean_epl_data[clean_epl_data['FTResult']==-1]
print(awaywins['AwayYellow'].mean())
```

```
1.7826086956521738
1.688348820586133
```

The cards analysis has shown that the Away team tends to receive more cards - which consequently impacts their win rate. This supports our original hypothesis that the away team is at a disadvantage.

3.5. Corner Analysis

3.5.1. Corners & Result

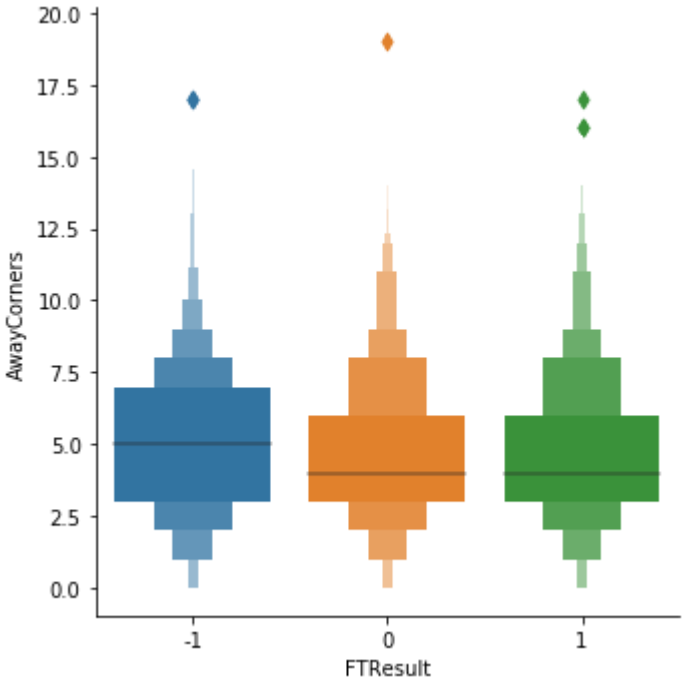
The boxplots below shows there is a weak correlation between corners and win rate:

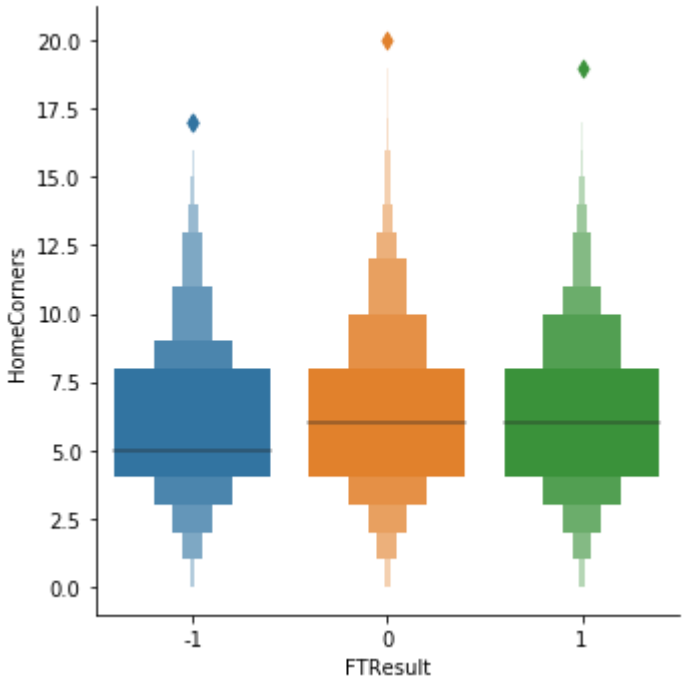
When the away team won Away team had a mean of 5.1 corners Home team had a mean of 6.3 corners
When the home team won Away team had a mean of 5.0 corners Home team had a mean of 6.2 corners
Teams that score a higher number of corners, have a higher win rate.

In [89]:

```
sns.catplot(x='FTResult', y='AwayCorners',kind='boxen', data=clean_epl_data)
sns.catplot(x='FTResult', y='HomeCorners',kind='boxen', data=clean_epl_data)
```

```
Out[89]: <seaborn.axisgrid.FacetGrid at 0x1a1b21e5f8>
```





4. EDA Conclusions

4.1. Insights

The exploratory analysis revealed a few relevant and interesting insights:

1. The AwayTeam is at significant disadvantage, not just in Win-Rate (28% compared to 46.5% at home)* but across all in-game event statistics
2. The most important in-game events to predict FTResult are:

- HalfTime Goals
- Shot Accuracy (& shots on target)
- Home Yellow Cards
- Red Cards

*calculation below

In [95]:

```
AwayWinTotal = len(clean_epl_data[clean_epl_data['FTResult']==-1])
print ('Away win rate is: '+ str(AwayWinTotal/len(clean_epl_data)))
HomeWinTotal = len(clean_epl_data[clean_epl_data['FTResult']==1]) print
('Home win rate is: '+ str(HomeWinTotal/len(clean_epl_data)))
```

```
Away win rate is: 0.28319838056680163
Home win rate is: 0.46558704453441296
```

4.2. Conclusion

Our data has shown insight on which are the most pertinent in-game event variables to predict game results. However, given the very stochastic nature of football, very few of the variables have strong correlation with the result of the game.

This unpredictability of football is perhaps what makes it so exciting!