In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
match = pd.read_csv('matches.csv')
delivery = pd.read_csv('deliveries.csv')
```

In [3]:

```
match.head()
```

Out[3]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	result	dl_appl
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	normal	
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	normal	
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	normal	
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	normal	
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	normal	

In [4]:

```
match.shape
```

Out[4]:

```
(756, 18)
```

In [5]:

```
delivery.head(6)
```

Out[5]:

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_supe
0	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	1	DA Warner	S Dhawan	TS Mills	
1	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	2	DA Warner	S Dhawan	TS Mills	
2	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	3	DA Warner	S Dhawan	TS Mills	
3	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	4	DA Warner	S Dhawan	TS Mills	

	match_id	inning	batting_team	bowling_team	over	ball	batsman	non_striker	bowler	is_supe
4	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	5	DA Warner	S Dhawan	TS Mills	
5	1	1	Sunrisers Hyderabad	Royal Challengers Bangalore	1	6	S Dhawan	DA Warner	TS Mills	

6 rows × 21 columns

In [6]:

```
delivery.groupby(['match_id','inning']).sum()['total_runs']
```

Out[6]:

```
match_id  inning
1         1         207
          2         172
2         1         184
          2         187
3         1         183
                   ...
11413     2         170
11414     1         155
          2         162
11415     1         152
          2         157
Name: total_runs, Length: 1528, dtype: int64
```

In [7]:

```
total_score_df = delivery.groupby(['match_id','inning']).sum()['total_runs'].reset_i
total_score_df = total_score_df[total_score_df['inning'] == 1]
total_score_df
```

Out[7]:

	match_id	inning	total_runs
0	1	1	207
2	2	1	184
4	3	1	183
6	4	1	163
8	5	1	157
...	...	...	...
1518	11347	1	143
1520	11412	1	136
1522	11413	1	171
1524	11414	1	155
1526	11415	1	152

756 rows × 3 columns

In [8]:

```
match_df = match.merge(total_score_df[['match_id','total_runs']],left_on='id',right_
match_df
```

Out[8]:

	id	Season	city	date	team1	team2	toss_winner	toss_decision	resul
0	1	IPL-2017	Hyderabad	05-04-2017	Sunrisers Hyderabad	Royal Challengers Bangalore	Royal Challengers Bangalore	field	norma
1	2	IPL-2017	Pune	06-04-2017	Mumbai Indians	Rising Pune Supergiant	Rising Pune Supergiant	field	norma
2	3	IPL-2017	Rajkot	07-04-2017	Gujarat Lions	Kolkata Knight Riders	Kolkata Knight Riders	field	norma
3	4	IPL-2017	Indore	08-04-2017	Rising Pune Supergiant	Kings XI Punjab	Kings XI Punjab	field	norma
4	5	IPL-2017	Bangalore	08-04-2017	Royal Challengers Bangalore	Delhi Daredevils	Royal Challengers Bangalore	bat	norma
...	...	...	...	...	...	...	...	...	.
751	11347	IPL-2019	Mumbai	05-05-2019	Kolkata Knight Riders	Mumbai Indians	Mumbai Indians	field	norma
752	11412	IPL-2019	Chennai	07-05-2019	Chennai Super Kings	Mumbai Indians	Chennai Super Kings	bat	norma
753	11413	IPL-2019	Visakhapatnam	08-05-2019	Sunrisers Hyderabad	Delhi Capitals	Delhi Capitals	field	norma
754	11414	IPL-2019	Visakhapatnam	10-05-2019	Delhi Capitals	Chennai Super Kings	Chennai Super Kings	field	norma
755	11415	IPL-2019	Hyderabad	12-05-2019	Mumbai Indians	Chennai Super Kings	Mumbai Indians	bat	norma

756 rows × 20 columns

In [9]:

```
match_df['team1'].unique()
```

Out[9]:

```
array(['Sunrisers Hyderabad', 'Mumbai Indians', 'Gujarat Lions',
       'Rising Pune Supergiant', 'Royal Challengers Bangalore',
       'Kolkata Knight Riders', 'Delhi Daredevils', 'Kings XI Punjab',
       'Chennai Super Kings', 'Rajasthan Royals', 'Deccan Chargers',
       'Kochi Tuskers Kerala', 'Pune Warriors', 'Rising Pune Supergiants',
       'Delhi Capitals'], dtype=object)
```

In [10]:

```
teams = [
    'Sunrisers Hyderabad',
    'Mumbai Indians',
    'Royal Challengers Bangalore',
    'Kolkata Knight Riders',
    'Kings XI Punjab',
    'Chennai Super Kings',
    'Rajasthan Royals',
    'Delhi Capitals',
    'Gujrat Titans',
    'Lucknow Super Giants'
]
```

In [11]:

```
match_df['team1'] = match_df['team1'].str.replace('Delhi Daredevils','Delhi Capitals
match_df['team2'] = match_df['team2'].str.replace('Delhi Daredevils','Delhi Capitals

match_df['team1'] = match_df['team1'].str.replace('Deccan Chargers','Sunrisers Hyder
match_df['team2'] = match_df['team2'].str.replace('Deccan Chargers','Sunrisers Hyder

match_df['team1'] = match_df['team1'].str.replace('Gujrat Lions','Gujrat Titans')
match_df['team2'] = match_df['team2'].str.replace('Gujrat Lions','Gujrat Titans')

match_df['team1'] = match_df['team1'].str.replace('Pune Warriors','Lucknow Super Gia
match_df['team2'] = match_df['team2'].str.replace('Pune Warriors','Lucknow Super Gia

match_df = match_df[match_df['team1'].isin(teams)]
match_df = match_df[match_df['team2'].isin(teams)]
match_df = match_df[match_df['team2'].isin(teams)]
match_df = match_df[match_df['team2'].isin(teams)]
match_df.shape
```

Out[11]:

```
(686, 20)
```

In [12]:

```
match_df = match_df[match_df['dl_applied'] == 0]
match_df = match_df[['match_id','city','winner','total_runs']]
delivery_df = match_df.merge(delivery,on='match_id')
delivery_df = delivery_df[delivery_df['inning'] == 2]
delivery_df.head(20)
```

Out[12]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5
130	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	6

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball
131	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	2	1
132	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	2	2
133	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	2	3
134	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	2	4
135	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	2	5
136	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	2	6
137	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	3	1
138	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	3	2
139	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	3	3
140	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	3	4
141	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	3	5
142	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	3	6
143	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	4	1
144	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	4	2

20 rows × 24 columns

In [13]:

```
delivery_df['current_score'] = delivery_df.groupby('match_id').cumsum()['total_runs_
delivery_df['runs_left'] = delivery_df['total_runs_x'] - delivery_df['current_score'
delivery_df['balls_left'] = 120 - (delivery_df['over']*6 + delivery_df['ball'])
delivery_df
```

Out[13]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1
...	...	...	...	...	...	...	...	...
160227	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20
160228	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20
160229	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20
160230	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20
160231	11415	Hyderabad	Mumbai Indians	152	2	Chennai Super Kings	Mumbai Indians	20

77596 rows × 27 columns

In [14]:

```
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].fillna("0")
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].apply(lambda x:x i
delivery_df['player_dismissed'] = delivery_df['player_dismissed'].astype('int')
wickets = delivery_df.groupby('match_id').cumsum()['player_dismissed'].values
delivery_df['wickets'] = 10 - wickets
delivery_df.head()
```

Out[14]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5

5 rows × 28 columns

In [15]:

```
delivery_df.head()
```

Out[15]:

	match_id	city	winner	total_runs_x	inning	batting_team	bowling_team	over	ball
125	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	1
126	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	2
127	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	3
128	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	4
129	1	Hyderabad	Sunrisers Hyderabad	207	2	Royal Challengers Bangalore	Sunrisers Hyderabad	1	5

5 rows × 28 columns

In [16]:

```
# crr = runs/overs
delivery_df['crr'] = (delivery_df['current_score']*6)/(120 - delivery_df['balls_left
delivery_df['rrr'] = (delivery_df['runs_left']*6)/delivery_df['balls_left']
def result(row):
    return 1 if row['batting_team'] == row['winner'] else 0
delivery_df['result'] = delivery_df.apply(result,axis=1)
final_df = delivery_df[['batting_team','bowling_team','city','runs_left','balls_left
final_df = final_df.sample(final_df.shape[0])
final_df.sample()
```

Out[16]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	crr	
22067	Kings XI Punjab	Royal Challengers Bangalore	Durban	85	50	9	168	7.114286	1(

In [17]:

```
final_df.dropna(inplace=True)
final_df = final_df[final_df['balls_left'] != 0]
```

In [18]:

```
pip install sklearn
```

```
Requirement already satisfied: sklearn in c:\users\asus\anaconda3\lib\site-packages
(0.0)
Requirement already satisfied: scikit-learn in c:\users\asus\anaconda3\lib\site-pack
ages (from sklearn) (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\users\asus\anaconda3\lib\site-pack
ages (from scikit-learn->sklearn) (1.1.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\asus\anaconda3\lib\site-pac
kages (from scikit-learn->sklearn) (1.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\asus\anaconda3\lib\s
ite-packages (from scikit-learn->sklearn) (2.2.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\asus\anaconda3\lib\site-pac
kages (from scikit-learn->sklearn) (1.20.3)
Note: you may need to restart the kernel to use updated packages.
```

In [19]:

```
X = final_df.iloc[:,:-1]
y = final_df.iloc[:,-1]
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=1)
X_train
```

Out[19]:

	batting_team	bowling_team	city	runs_left	balls_left	wickets	total_runs_x	c
121990	Rajasthan Royals	Chennai Super Kings	Chennai	111	73	9	157	5.87234
136725	Rajasthan Royals	Mumbai Indians	Jaipur	39	8	4	172	7.12500
9300	Mumbai Indians	Kings XI Punjab	Chandigarh	173	106	10	182	3.85714
20372	Chennai Super Kings	Mumbai Indians	Cape Town	147	95	8	165	4.32000
155583	Delhi Capitals	Rajasthan Royals	Jaipur	39	17	8	196	9.14563
...	...	...	...	...	...	...	...	
80259	Chennai Super Kings	Mumbai Indians	Chennai	40	14	4	148	6.11320
46307	Chennai Super Kings	Kings XI Punjab	Dharamsala	160	85	8	192	5.48571
147546	Royal Challengers Bangalore	Mumbai Indians	Bengaluru	100	47	8	197	7.97260
134972	Delhi Daredevils	Kolkata Knight Riders	Kolkata	145	79	8	206	8.92682
86686	Mumbai Indians	Kolkata Knight Riders	Kolkata	144	94	10	159	3.46153

61056 rows × 9 columns

In [21]:

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
```

In [22]:

```
trf = ColumnTransformer([('trf',OneHotEncoder(sparse=False,drop='first'),['batting_t
,remainder='passthrough')
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
pipe = Pipeline(steps=[
    ('step1',trf),
    ('step2',LogisticRegression(solver='liblinear'))
])
pipe.fit(X_train,y_train)
```

Out[22]:

```
Pipeline(steps=[('step1',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('trf',
                                                  OneHotEncoder(drop='first',
                                                                sparse=False),
                                                  ['batting_team',
                                                   'bowling_team', 'city'])])),
                ('step2', LogisticRegression(solver='liblinear'))])
```

In [23]:

```
y_pred = pipe.predict(X_test)
```

In [24]:

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)

pipe.predict_proba(X_test)[10]
```

Out[24]:

```
array([0.50774037, 0.49225963])
```

In [25]:

```
def match_summary(row):
    print("Batting Team-" + row['batting_team'] + " | Bowling Team-" + row['bowling_
```

In [26]:

```
def match_progression(x_df,match_id,pipe):
    match = x_df[x_df['match_id'] == match_id]
    match = match[(match['ball'] == 6)]
    temp_df = match[['batting_team','bowling_team','city','runs_left','balls_left','
    temp_df = temp_df[temp_df['balls_left'] != 0]
    result = pipe.predict_proba(temp_df)
    temp_df['lose'] = np.round(result.T[0]*100,1)
    temp_df['win'] = np.round(result.T[1]*100,1)
    temp_df['end_of_over'] = range(1,temp_df.shape[0]+1)

    target = temp_df['total_runs_x'].values[0]
    runs = list(temp_df['runs_left'].values)
    new_runs = runs[:]
    runs.insert(0,target)
    temp_df['runs_after_over'] = np.array(runs)[:-1] - np.array(new_runs)
    wickets = list(temp_df['wickets'].values)
    new_wickets = wickets[:]
    new_wickets.insert(0,10)
    wickets.append(0)
    w = np.array(wickets)
    nw = np.array(new_wickets)
    temp_df['wickets_in_over'] = (nw - w)[0:temp_df.shape[0]]

    print("Target-",target)
    temp_df = temp_df[['end_of_over','runs_after_over','wickets_in_over','lose','win
    return temp_df,target
```

In [27]:

```
temp_df,target = match_progression(delivery_df,74,pipe)
temp_df
```

```
Target- 178
```

Out[27]:

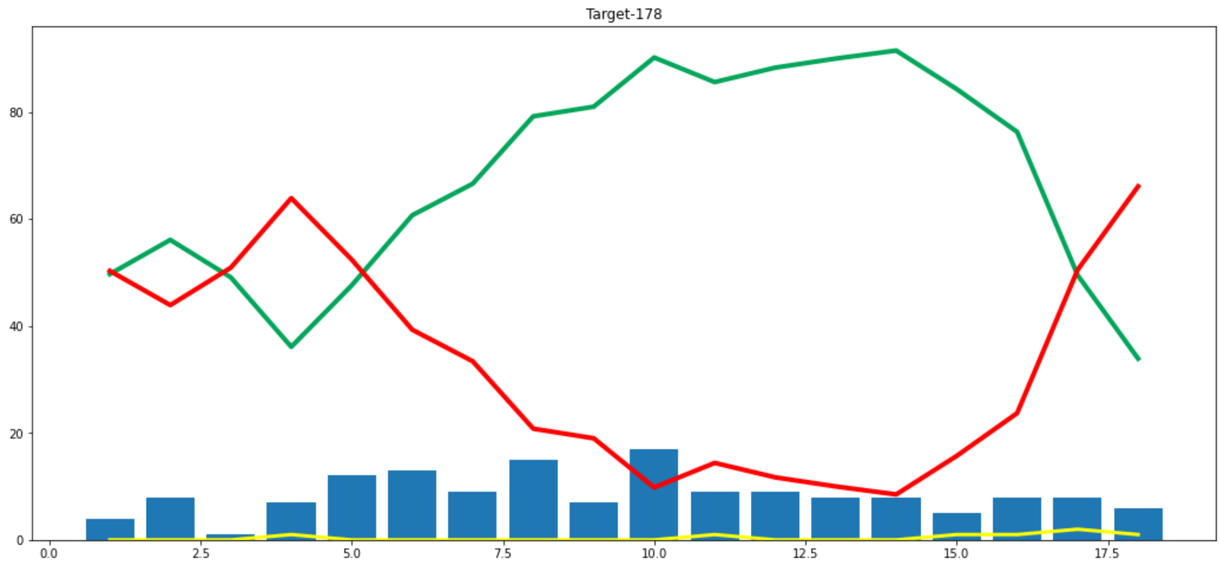
	end_of_over	runs_after_over	wickets_in_over	lose	win
10459	1	4	0	50.3	49.7
10467	2	8	0	43.9	56.1
10473	3	1	0	50.9	49.1
10479	4	7	1	63.9	36.1
10485	5	12	0	52.4	47.6
10491	6	13	0	39.3	60.7
10497	7	9	0	33.4	66.6
10505	8	15	0	20.8	79.2
10511	9	7	0	19.0	81.0
10518	10	17	0	9.8	90.2
10524	11	9	1	14.4	85.6
10530	12	9	0	11.7	88.3
10536	13	8	0	10.0	90.0
10542	14	8	0	8.5	91.5
10548	15	5	1	15.7	84.3
10555	16	8	1	23.7	76.3
10561	17	8	2	50.5	49.5
10567	18	6	1	66.1	33.9

In [28]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(temp_df['end_of_over'],temp_df['wickets_in_over'],color='yellow',linewidth=
plt.plot(temp_df['end_of_over'],temp_df['win'],color='#00a65a',linewidth=4)
plt.plot(temp_df['end_of_over'],temp_df['lose'],color='red',linewidth=4)
plt.bar(temp_df['end_of_over'],temp_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[28]:

```
Text(0.5, 1.0, 'Target-178')
```



In [29]:

```
temp_df,target = match_progression(delivery_df,14,pipe)
temp_df
```

```
Target- 172
```

Out[29]:

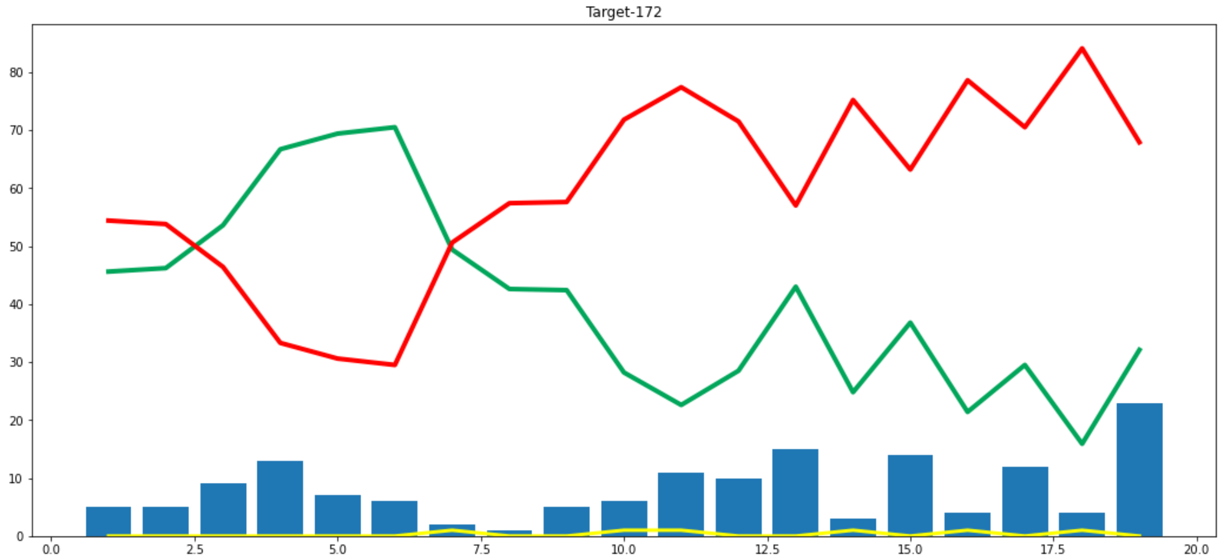
	end_of_over	runs_after_over	wickets_in_over	lose	win
1809	1	5	0	54.4	45.6
1816	2	5	0	53.8	46.2
1822	3	9	0	46.4	53.6
1828	4	13	0	33.3	66.7
1834	5	7	0	30.6	69.4
1840	6	6	0	29.5	70.5
1846	7	2	1	50.6	49.4
1852	8	1	0	57.4	42.6
1858	9	5	0	57.6	42.4
1864	10	6	1	71.8	28.2
1870	11	11	1	77.4	22.6
1876	12	10	0	71.5	28.5
1882	13	15	0	57.0	43.0
1888	14	3	1	75.2	24.8
1894	15	14	0	63.2	36.8
1902	16	4	1	78.6	21.4
1908	17	12	0	70.5	29.5
1915	18	4	1	84.1	15.9
1927	19	23	0	67.9	32.1

In [30]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(temp_df['end_of_over'],temp_df['wickets_in_over'],color='yellow',linewidth=
plt.plot(temp_df['end_of_over'],temp_df['win'],color='#00a65a',linewidth=4)
plt.plot(temp_df['end_of_over'],temp_df['lose'],color='red',linewidth=4)
plt.bar(temp_df['end_of_over'],temp_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[30]:

```
Text(0.5, 1.0, 'Target-172')
```



In [32]:

```
temp_df,target = match_progression(delivery_df,104,pipe)
temp_df
```

```
Target- 154
```

Out[32]:

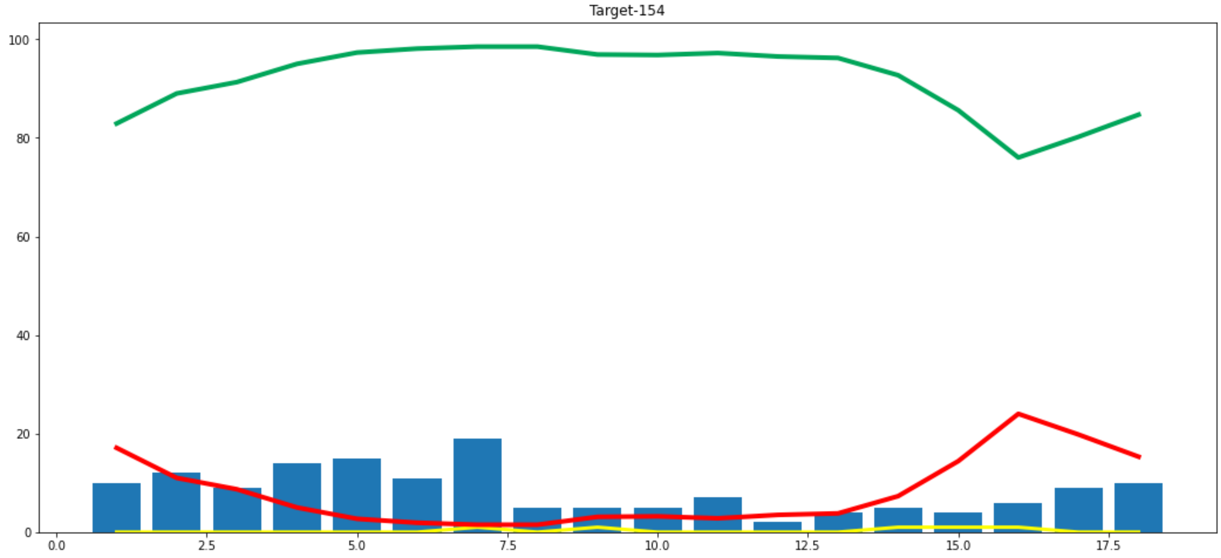
	end_of_over	runs_after_over	wickets_in_over	lose	win
16986	1	10	0	17.1	82.9
16992	2	12	0	11.0	89.0
16999	3	9	0	8.7	91.3
17005	4	14	0	5.0	95.0
17011	5	15	0	2.7	97.3
17017	6	11	0	1.9	98.1
17023	7	19	1	1.5	98.5
17029	8	5	0	1.5	98.5
17035	9	5	1	3.1	96.9
17041	10	5	0	3.2	96.8
17047	11	7	0	2.8	97.2
17053	12	2	0	3.5	96.5
17059	13	4	0	3.8	96.2
17065	14	5	1	7.3	92.7
17071	15	4	1	14.4	85.6
17077	16	6	1	24.0	76.0
17083	17	9	0	19.8	80.2
17089	18	10	0	15.3	84.7

In [33]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(temp_df['end_of_over'],temp_df['wickets_in_over'],color='yellow',linewidth=
plt.plot(temp_df['end_of_over'],temp_df['win'],color='#00a65a',linewidth=4)
plt.plot(temp_df['end_of_over'],temp_df['lose'],color='red',linewidth=4)
plt.bar(temp_df['end_of_over'],temp_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[33]:

```
Text(0.5, 1.0, 'Target-154')
```



In [36]:

```
temp_df,target = match_progression(delivery_df,24,pipe)
temp_df
```

```
Target- 142
```

Out[36]:

	end_of_over	runs_after_over	wickets_in_over	lose	win
3272	1	9	1	44.8	55.2
3278	2	5	1	61.6	38.4
3284	3	5	0	61.7	38.3
3290	4	2	1	79.7	20.3
3297	5	0	1	91.6	8.4
3303	6	1	1	96.6	3.4
3310	7	6	1	98.1	1.9
3316	8	6	0	98.0	2.0
3322	9	9	0	97.4	2.6
3329	10	14	0	95.5	4.5
3335	11	10	0	93.9	6.1
3341	12	2	0	94.9	5.1
3347	13	9	0	93.6	6.4
3354	14	5	0	93.6	6.4
3360	15	8	0	92.4	7.6
3366	16	6	0	92.1	7.9
3372	17	4	0	92.6	7.4

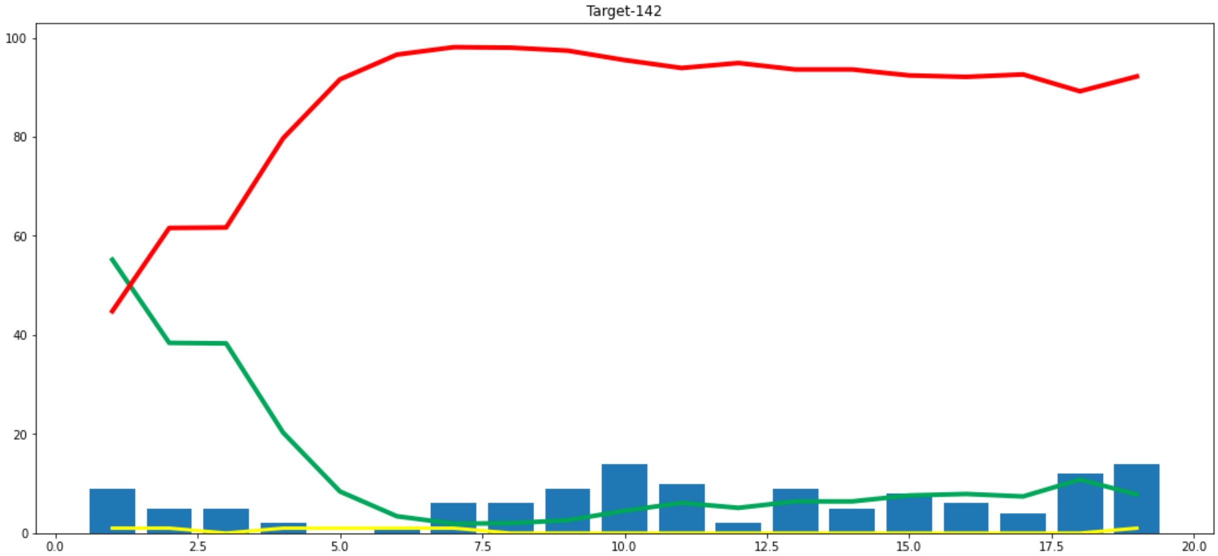
	end_of_over	runs_after_over	wickets_in_over	lose	win
3378	18	12	0	89.2	10.8
3390	19	14	1	92.2	7.8

In [37]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(temp_df['end_of_over'],temp_df['wickets_in_over'],color='yellow',linewidth=
plt.plot(temp_df['end_of_over'],temp_df['win'],color='#00a65a',linewidth=4)
plt.plot(temp_df['end_of_over'],temp_df['lose'],color='red',linewidth=4)
plt.bar(temp_df['end_of_over'],temp_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[37]:

```
Text(0.5, 1.0, 'Target-142')
```



In [40]:

```
temp_df,target = match_progression(delivery_df,12,pipe)
temp_df
```

```
Target- 142
```

Out[40]:

	end_of_over	runs_after_over	wickets_in_over	lose	win
1564	1	3	0	8.4	91.6
1571	2	4	1	15.9	84.1
1577	3	1	3	65.5	34.5
1583	4	2	0	69.7	30.3
1589	5	8	0	64.9	35.1
1595	6	3	0	67.8	32.2
1601	7	10	0	60.1	39.9
1607	8	2	1	78.4	21.6
1613	9	3	0	80.6	19.4
1620	10	12	0	72.5	27.5
1626	11	6	0	71.2	28.8
1632	12	9	0	65.8	34.2

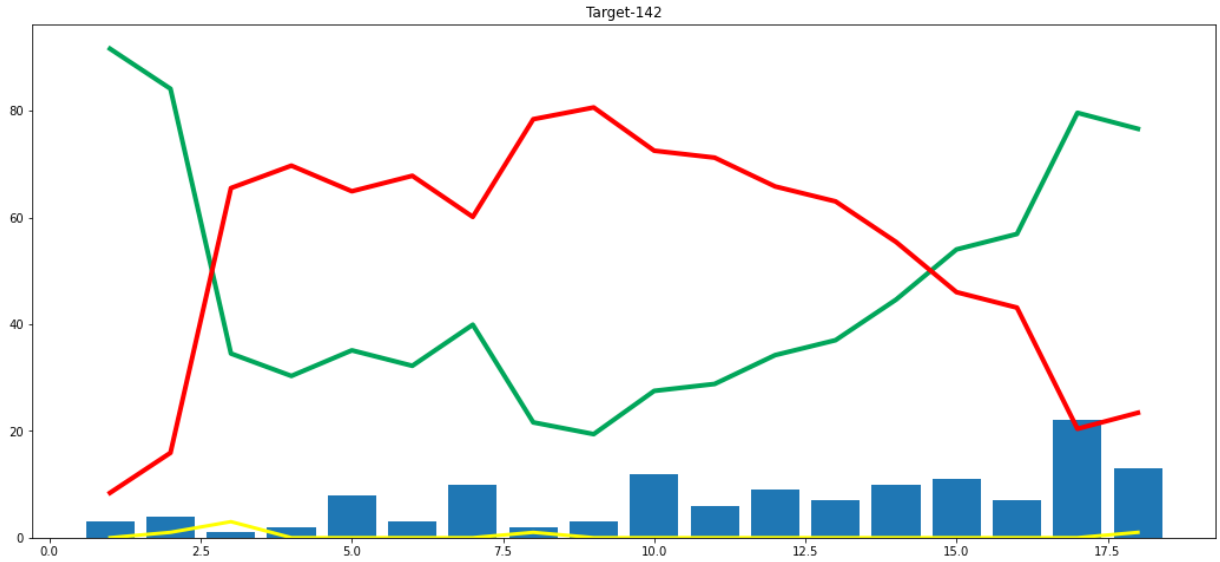
	end_of_over	runs_after_over	wickets_in_over	lose	win
1639	13	7	0	63.0	37.0
1645	14	10	0	55.4	44.6
1651	15	11	0	46.0	54.0
1657	16	7	0	43.1	56.9
1665	17	22	0	20.4	79.6
1671	18	13	1	23.4	76.6

In [41]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(18,8))
plt.plot(temp_df['end_of_over'],temp_df['wickets_in_over'],color='yellow',linewidth=
plt.plot(temp_df['end_of_over'],temp_df['win'],color='#00a65a',linewidth=4)
plt.plot(temp_df['end_of_over'],temp_df['lose'],color='red',linewidth=4)
plt.bar(temp_df['end_of_over'],temp_df['runs_after_over'])
plt.title('Target-' + str(target))
```

Out[41]:

```
Text(0.5, 1.0, 'Target-142')
```



In [ ]: