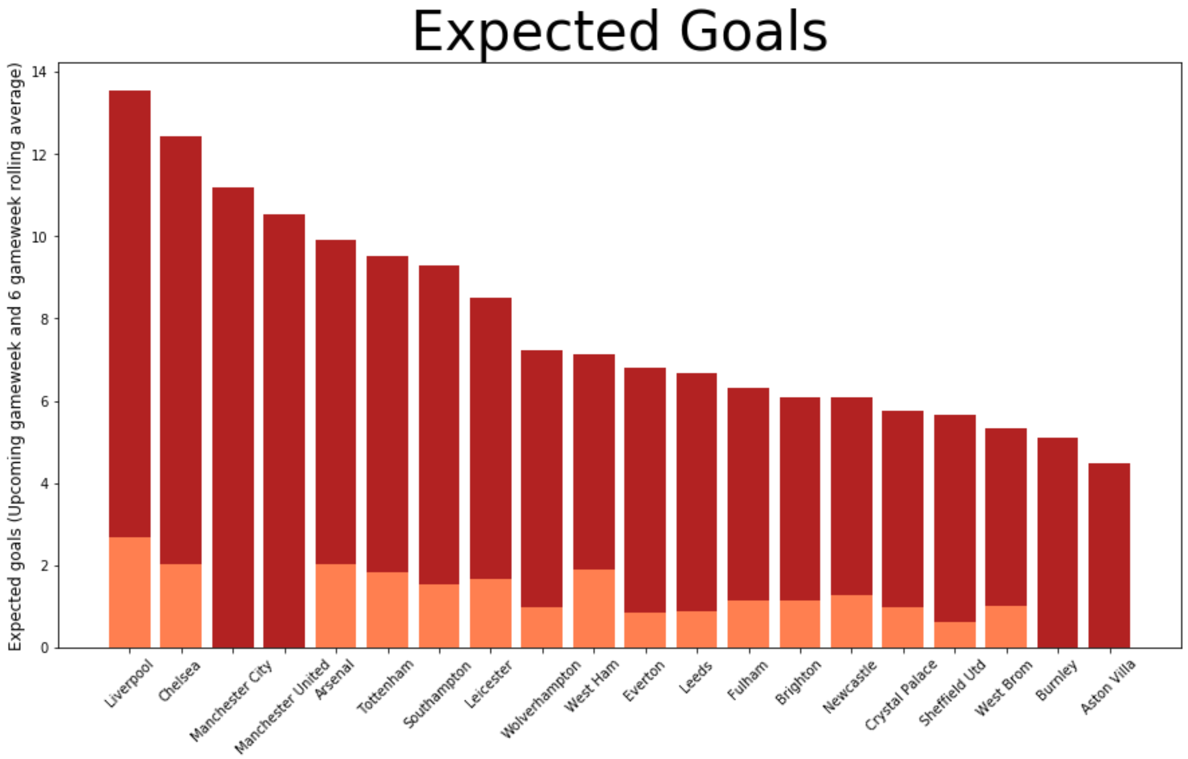
Transfer and selection Policy

This script looks at transfer policies. For the first part of the season, we will just focus on fixtures (expected goals and clean sheets) until the dust settles and we can see which players are performing well this season and calculate fresh ROI's. I feel that anything under 6 points will not have asymptoted enough towards its true value.

Expected goals

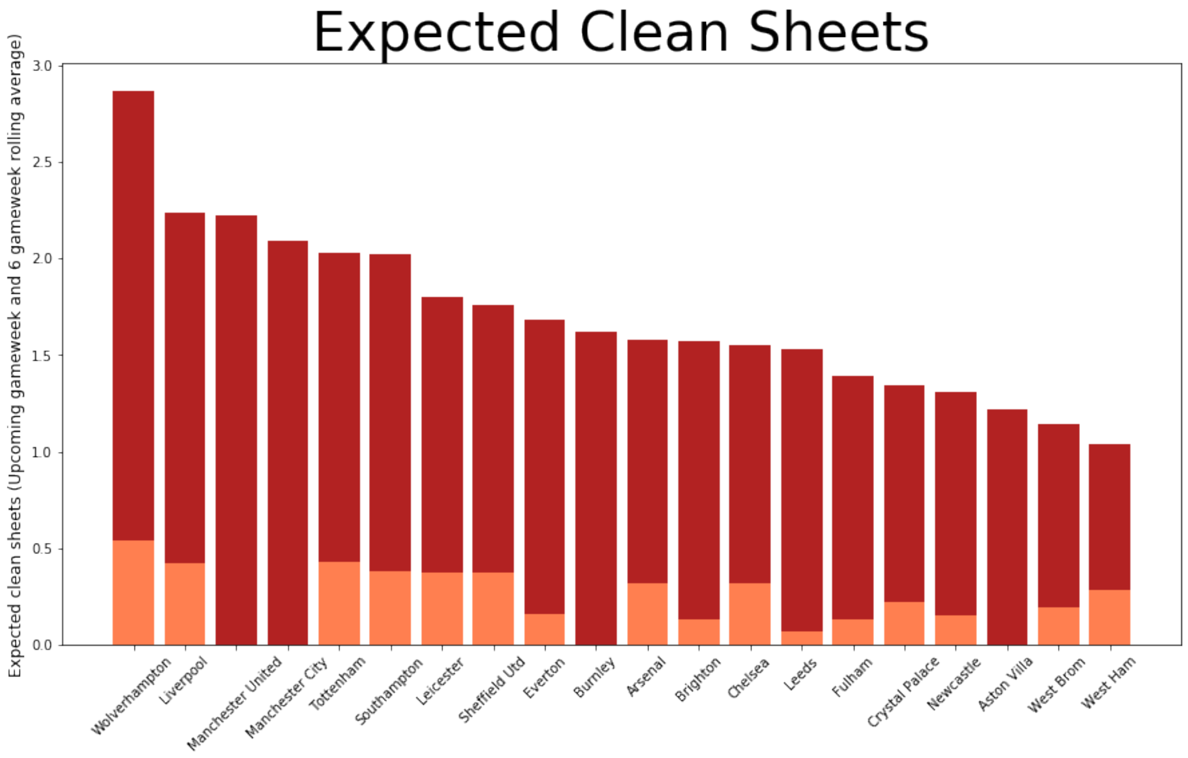
In [6]:

```
import requests
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

url = "https://playerdatabase247.com/include_premier_league_fixture
_tracker_uusi.php?listtype=expgoals"
r = requests.get(url)
#print(r.status_code)

soup = BeautifulSoup(r.text, 'html.parser')
table = soup.find('table')
cells = table.find_all("td")

def scrape(cells):
    lizt = []
    for cell in cells:
        text = cell.text.strip()
        lizt.append(text)
    return(lizt)

output = scrape(cells)

output = np.array(output)
output = output.reshape(21,8)
output = pd.DataFrame(output)
header_row = 0
output.columns = output.iloc[header_row]
output = output.drop(header_row)
output = output.reset_index(drop = True)
output.columns = ["team", "gw1", "gw2", "gw3", "gw4", "gw5", "gw6",
"total"]

def remove_char(string):
    import re
    string = re.sub("[A-Za-z]", "", string);
    return string

output.gw1 = output.gw1.apply(remove_char)
output.gw1 = output.gw1.apply(float)
output.gw2 = output.gw2.apply(remove_char)
output.gw2 = output.gw2.apply(float)
output.gw3 = output.gw3.apply(remove_char)
output.gw3 = output.gw3.apply(float)
output.gw4 = output.gw4.apply(remove_char)
output.gw4 = output.gw4.apply(float)
output.gw5 = output.gw5.apply(remove_char)
output.gw5 = output.gw5.apply(float)
output.gw6 = output.gw6.apply(remove_char)
output.gw6 = output.gw6.apply(float)

fig, ax = plt.subplots(figsize = (15,8))
output.total = output.total.apply(float)
ax.bar(output.team, output.total, color = "firebrick")
ax.bar(output.team, output.gw1, color = "coral")
ax.set_ylabel("Expected goals (Upcoming gameweek and 6 gameweek rol
ling average)", size = 12)
plt.xticks(rotation = 45)
plt.title("Expected Goals", size = 40)



plt.show()
```



In [5]:

```
url = "https://playerdatabase247.com/include_premier_league_fixture_tracker_uusi.php?listtype=cs"

r = requests.get(url)
#print(r.status_code)

soup = BeautifulSoup(r.text, 'html.parser')
table = soup.find('table')
cells = table.find_all("td")

output = scrape(cells)

output = np.array(output)
output = output.reshape(21,8)
output = pd.DataFrame(output)
header_row = 0
output.columns = output.iloc[header_row]
output = output.drop(header_row)
output = output.reset_index(drop = True)
output.columns = ["team", "gw1", "gw2", "gw3", "gw4", "gw5", "gw6", "total"]

output.gw1 = output.gw1.apply(remove_char)
output.gw1 = output.gw1.apply(float)
output.gw2 = output.gw2.apply(remove_char)
output.gw2 = output.gw2.apply(float)
output.gw3 = output.gw3.apply(remove_char)
output.gw3 = output.gw3.apply(float)
output.gw4 = output.gw4.apply(remove_char)
output.gw4 = output.gw4.apply(float)
output.gw5 = output.gw5.apply(remove_char)
output.gw5 = output.gw5.apply(float)
output.gw6 = output.gw6.apply(remove_char)
output.gw6 = output.gw6.apply(float)

fig, ax = plt.subplots(figsize = (15,8))
output.total = output.total.apply(float)
ax.bar(output.team, output.total, color = "firebrick")
ax.bar(output.team, output.gw1, color = "coral")
ax.set_ylabel("Expected clean sheets (Upcoming gameweek and 6 gameweek rolling average)", size = 12)
plt.xticks(rotation = 45)
plt.title("Expected Clean Sheets", size = 40)

plt.show()
```


Expected Clean Sheets
Expected clean sheets (Upcoming gameweek and 6 gameweek rolling average)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Wolverhampton
Liverpool
Manchester United
Manchester City
Tottenham
Southampton
Leicester
Sheffield Utd
Everton
Burnley
Arsenal
Brighton
Chelsea
Leeds
Fulham
Crystal Palace
Newcastle
Aston Villa
West Brom
West Ham

```
In [ ]:
```