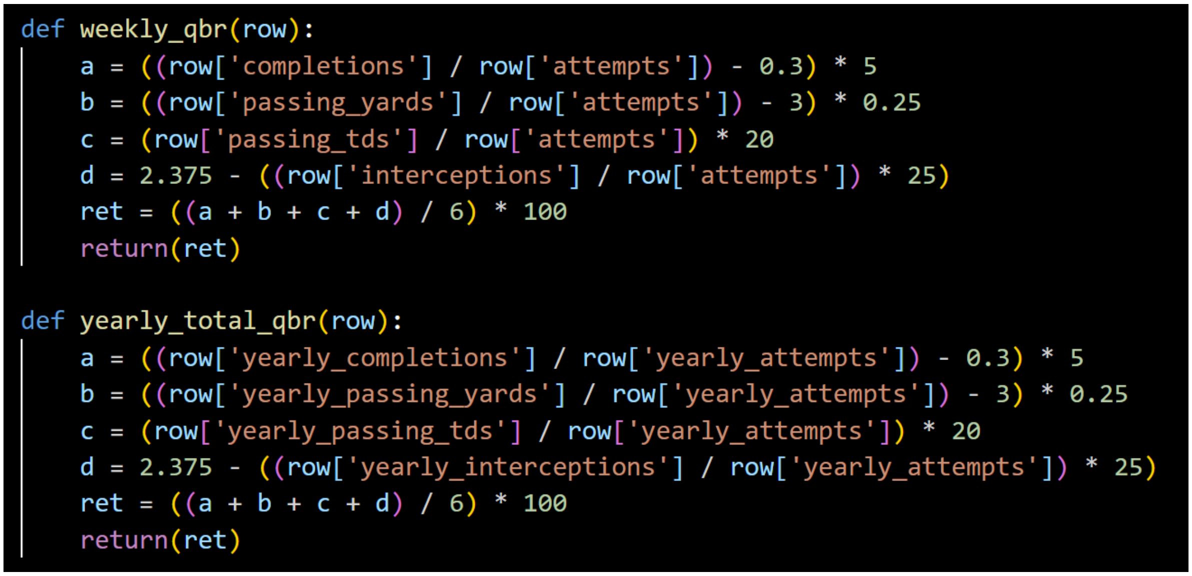
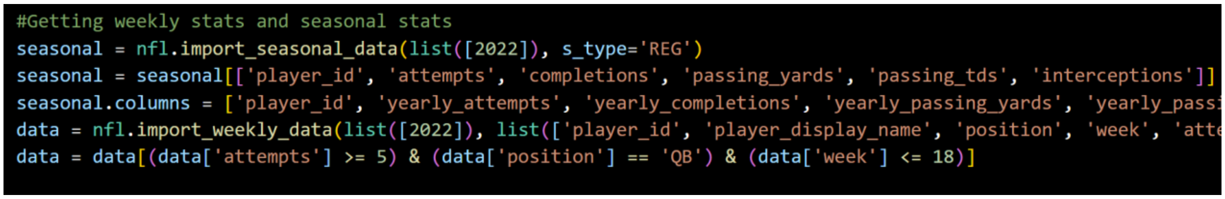


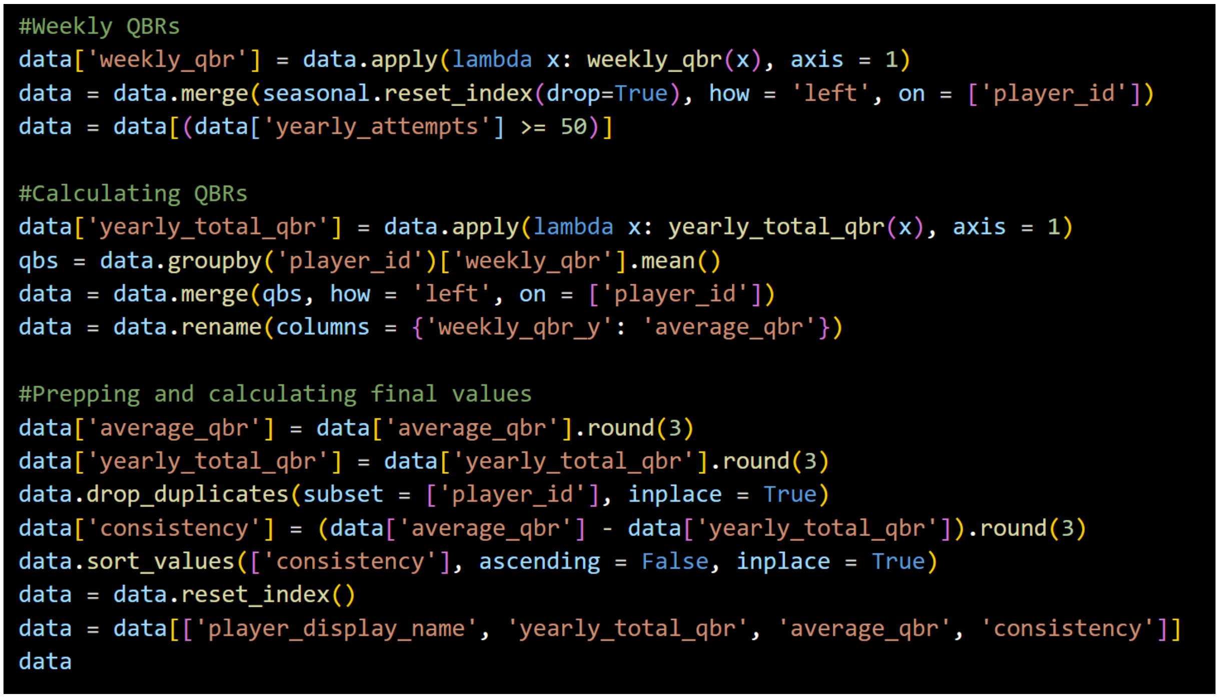
```
def weekly_qbr(row):
    a = ((row['completions'] / row['attempts']) - 0.3) * 5
    b = ((row['passing_yards'] / row['attempts']) - 3) * 0.25
    c = (row['passing_tds'] / row['attempts']) * 20
    d = 2.375 - ((row['interceptions'] / row['attempts']) * 25)
    ret = ((a + b + c + d) / 6) * 100
    return(ret)

def yearly_total_qbr(row):
    a = ((row['yearly_completions'] / row['yearly_attempts']) - 0.3) * 5
    b = ((row['yearly_passing_yards'] / row['yearly_attempts']) - 3) * 0.25
    c = (row['yearly_passing_tds'] / row['yearly_attempts']) * 20
    d = 2.375 - ((row['yearly_interceptions'] / row['yearly_attempts']) * 25)
    ret = ((a + b + c + d) / 6) * 100
    return(ret)
```

Define the QBR calculations as callable functions. Weekly_qbr calculates the QBR for just one week, while yearly_total_qbr calculates the QBR based on year totals. Weekly_qbr will need to be calculated for each week, then averaged.



```
#Getting weekly stats and seasonal stats
seasonal = nfl.import_seasonal_data(list([2022]), s_type='REG')
seasonal = seasonal[['player_id', 'attempts', 'completions', 'passing_yards', 'passing_tds', 'interceptions']]
seasonal.columns = ['player_id', 'yearly_attempts', 'yearly_completions', 'yearly_passing_yards', 'yearly_passi
data = nfl.import_weekly_data(list([2022]), list(['player_id', 'player_display_name', 'position', 'week', 'atte
data = data[(data['attempts'] >= 5) & (data['position'] == 'QB') & (data['week'] <= 18)]
```

Collect the weekly and seasonal stats. NFL Py is used for the data. Just regular season games where a QB attempts 5 or more passes are considered.



```
#Weekly QBRs
data['weekly_qbr'] = data.apply(lambda x: weekly_qbr(x), axis = 1)
data = data.merge(seasonal.reset_index(drop=True), how = 'left', on = ['player_id'])
data = data[(data['yearly_attempts'] >= 50)]

#Calculating QBRs
data['yearly_total_qbr'] = data.apply(lambda x: yearly_total_qbr(x), axis = 1)
qbs = data.groupby('player_id')['weekly_qbr'].mean()
data = data.merge(qbs, how = 'left', on = ['player_id'])
data = data.rename(columns = {'weekly_qbr_y': 'average_qbr'})

#Prepping and calculating final values
data['average_qbr'] = data['average_qbr'].round(3)
data['yearly_total_qbr'] = data['yearly_total_qbr'].round(3)
data.drop_duplicates(subset = ['player_id'], inplace = True)
data['consistency'] = (data['average_qbr'] - data['yearly_total_qbr']).round(3)
data.sort_values(['consistency'], ascending = False, inplace = True)
data = data.reset_index()
data = data[['player_display_name', 'yearly_total_qbr', 'average_qbr', 'consistency']]
data
```

The first section of code calls the weekly_qbr function on each record and then merges the weekly data with the seasonal data for easier display. The second section of code calculates the yearly QBRs. It then groups the database by player name, averaging the weekly QBRs, then remerges the databases. Finally, the third section of code establishes the display table by rounding figures, dropping duplicates, and calculating and sorting by consistency. The columns are renamed for readability and the table is displayed.