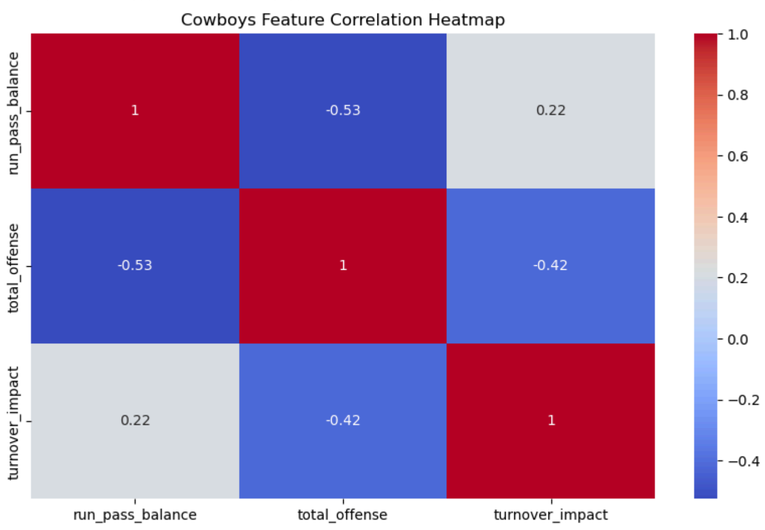
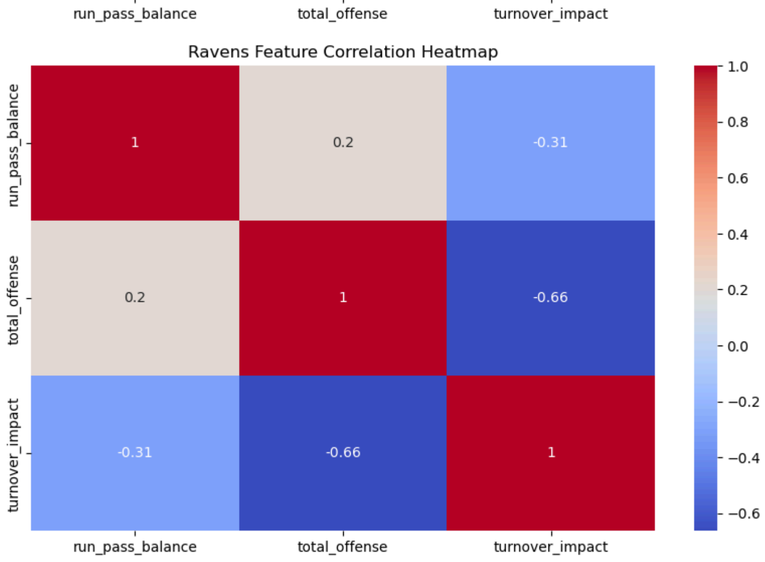
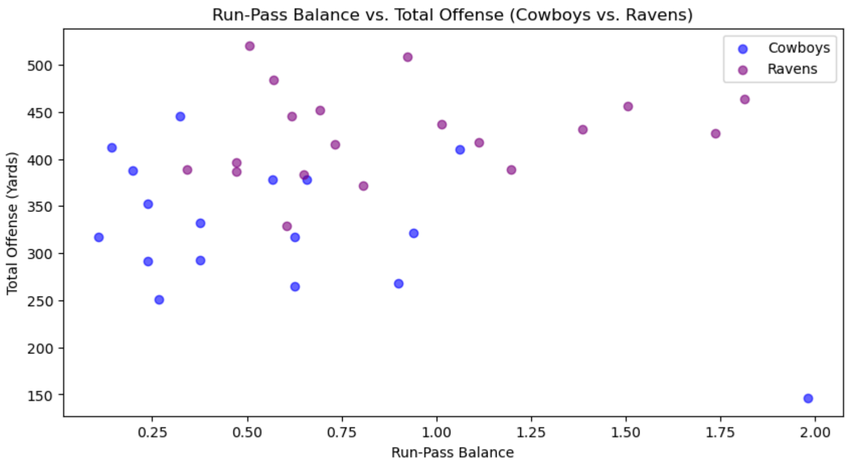
In [175…

```python
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load datasets
cowboys_raw = pd.read_excel('/Users/jwsue/Desktop/Bellevue/DSC680 - Applied Data Science/Milestone 2 draft/24-25_cowboys_stats_EX.xlsx', header=None)
ravens_raw = pd.read_excel('/Users/jwsue/Desktop/Bellevue/DSC680 - Applied Data Science/Milestone 2 draft/24-25_ravens_stats_EX.xlsx', header=None)

# Extract headers from second row
cowboys_headers = cowboys_raw.iloc[1]
ravens_headers = ravens_raw.iloc[1]

# Assign headers and drop the first two rows
cowboys_stats = cowboys_raw[2:].reset_index(drop=True)
ravens_stats = ravens_raw[2:].reset_index(drop=True)

cowboys_stats.columns = cowboys_headers
ravens_stats.columns = ravens_headers

# Standardize column names
cowboys_stats.columns = cowboys_stats.columns.astype(str).str.strip()
ravens_stats.columns = ravens_stats.columns.astype(str).str.strip()

# Add team identifier
cowboys_stats['team'] = 'Cowboys'
ravens_stats['team'] = 'Ravens'

# Rename duplicate columns to avoid errors
def rename_duplicate_columns(df):
    counts = {}
    new_columns = []
    for col in df.columns:
        if col in counts:
            counts[col] += 1
            new_columns.append(f"{col}_{counts[col]}")
        else:
            counts[col] = 1
            new_columns.append(col)
    df.columns = new_columns
    return df

cowboys_stats = rename_duplicate_columns(cowboys_stats)
ravens_stats = rename_duplicate_columns(ravens_stats)

# Identify the correct columns for PassY and RushY
cowboys_pass_col = [col for col in cowboys_stats.columns if 'PassY' in col][0]
cowboys_rush_col = [col for col in cowboys_stats.columns if 'RushY' in col][0]

ravens_pass_col = [col for col in ravens_stats.columns if 'PassY' in col][0]
ravens_rush_col = [col for col in ravens_stats.columns if 'RushY' in col][0]

# Compute key metrics
cowboys_stats['run_pass_balance'] = pd.to_numeric(cowboys_stats[cowboys_rush_col], errors='coerce') / \
                                    (pd.to_numeric(cowboys_stats[cowboys_pass_col], errors='coerce') + 1e-5)

cowboys_stats['total_offense'] = pd.to_numeric(cowboys_stats[cowboys_pass_col], errors='coerce') + \
                                 pd.to_numeric(cowboys_stats[cowboys_rush_col], errors='coerce')

ravens_stats['run_pass_balance'] = pd.to_numeric(ravens_stats[ravens_rush_col], errors='coerce') / \
                                   (pd.to_numeric(ravens_stats[ravens_pass_col], errors='coerce') + 1e-5)

ravens_stats['total_offense'] = pd.to_numeric(ravens_stats[ravens_pass_col], errors='coerce') + \
                                pd.to_numeric(ravens_stats[ravens_rush_col], errors='coerce')

#Enhancement: Calculate Turnover Impact
cowboys_stats['turnover_impact'] = pd.to_numeric(cowboys_stats['TO'], errors='coerce') - pd.to_numeric(cowboys_stats['TO_2'], errors='coerce')
ravens_stats['turnover_impact'] = pd.to_numeric(ravens_stats['TO'], errors='coerce') - pd.to_numeric(ravens_stats['TO_2'], errors='coerce')

print("Cowboys and Ravens metrics computed successfully!")

# Display sample results
print("\nCowboys Sample Data with New Metrics:")
print(cowboys_stats[['Week', 'run_pass_balance', 'total_offense', 'turnover_impact']].head())

print("\nRavens Sample Data with New Metrics:")
print(ravens_stats[['Week', 'run_pass_balance', 'total_offense', 'turnover_impact']].head())

# Save cleaned datasets
cowboys_stats.to_csv("Processed_Cowboys_Data.csv", index=False)
ravens_stats.to_csv("Processed_Ravens_Data.csv", index=False)

#Enhancement: Correlation Heatmap for Feature Relationships
plt.figure(figsize=(10,6))
sns.heatmap(cowboys_stats[['run_pass_balance', 'total_offense', 'turnover_impact']].corr(), annot=True, cmap='coolwarm')
plt.title("Cowboys Feature Correlation Heatmap")
plt.show()

plt.figure(figsize=(10,6))
sns.heatmap(ravens_stats[['run_pass_balance', 'total_offense', 'turnover_impact']].corr(), annot=True, cmap='coolwarm')
plt.title("Ravens Feature Correlation Heatmap")
plt.show()

#Enhancement: Visualization - Run-Pass Balance vs. Total Offense
plt.figure(figsize=(10, 5))
plt.scatter(cowboys_stats['run_pass_balance'], cowboys_stats['total_offense'], alpha=0.6, label="Cowboys", color='blue')
plt.scatter(ravens_stats['run_pass_balance'], ravens_stats['total_offense'], alpha=0.6, label="Ravens", color='purple')
plt.xlabel("Run-Pass Balance")
plt.ylabel("Total Offense (Yards)")
plt.title("Run-Pass Balance vs. Total Offense (Cowboys vs. Ravens)")
plt.legend()
plt.show()
```

```
Cowboys and Ravens metrics computed successfully!

Cowboys Sample Data with New Metrics:
  Week  run_pass_balance  total_offense  turnover_impact
0    1          0.625767          265.0              NaN
1    2          0.238596          353.0              1.0
2    3          0.141274          412.0              NaN
3    4          0.375587          293.0              NaN
4    5          0.324405          445.0              2.0

Ravens Sample Data with New Metrics:
  Week  run_pass_balance  total_offense  turnover_impact
0    1          0.692884          452.0              0.0
1    2          0.650862          383.0              0.0
2    3          1.505494          456.0              NaN
3    4          1.737179          427.0              0.0
4    5          0.507246          520.0              0.0
```






Run-Pass Balance vs. Total Offense (Cowboys vs. Ravens)
Cowboys
Ravens
Total Offense (Yards)
Run-Pass Balance
500
450
400
350
300
250
200
150
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00