Pitch Tempo Analysis

December 4, 2024

[3]:
```
pip install gspread pandas gspread-dataframe oauth2client
```

```
Requirement already satisfied: gspread in /opt/anaconda3/lib/python3.12/site-
packages (6.1.4)
Requirement already satisfied: pandas in /opt/anaconda3/lib/python3.12/site-
packages (2.2.2)
Requirement already satisfied: gspread-dataframe in
/opt/anaconda3/lib/python3.12/site-packages (4.0.0)
Requirement already satisfied: oauth2client in
/opt/anaconda3/lib/python3.12/site-packages (4.1.3)
Requirement already satisfied: google-auth>=1.12.0 in
/opt/anaconda3/lib/python3.12/site-packages (from gspread) (2.36.0)
Requirement already satisfied: google-auth-oauthlib>=0.4.1 in
/opt/anaconda3/lib/python3.12/site-packages (from gspread) (1.2.1)
Requirement already satisfied: numpy>=1.26.0 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/opt/anaconda3/lib/python3.12/site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.12.0 in
/opt/anaconda3/lib/python3.12/site-packages (from gspread-dataframe) (1.16.0)
Requirement already satisfied: httplib2>=0.9.1 in
/opt/anaconda3/lib/python3.12/site-packages (from oauth2client) (0.22.0)
Requirement already satisfied: pyasn1>=0.1.7 in
/opt/anaconda3/lib/python3.12/site-packages (from oauth2client) (0.4.8)
Requirement already satisfied: pyasn1-modules>=0.0.5 in
/opt/anaconda3/lib/python3.12/site-packages (from oauth2client) (0.2.8)
Requirement already satisfied: rsa>=3.1.4 in /opt/anaconda3/lib/python3.12/site-
packages (from oauth2client) (4.9)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from google-auth>=1.12.0->gspread)
(5.3.3)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/anaconda3/lib/python3.12/site-packages (from google-auth-
oauthlib>=0.4.1->gspread) (2.0.0)
```

```
Requirement already satisfied:
pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in
/opt/anaconda3/lib/python3.12/site-packages (from httplib2>=0.9.1->oauth2client)
(3.0.9)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib>=0.4.1->gspread) (3.2.2)
Requirement already satisfied: requests>=2.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib>=0.4.1->gspread) (2.32.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/anaconda3/lib/python3.12/site-packages (from requests>=2.0.0->requests-
oauthlib>=0.7.0->google-auth-oauthlib>=0.4.1->gspread) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
/opt/anaconda3/lib/python3.12/site-packages (from requests>=2.0.0->requests-
oauthlib>=0.7.0->google-auth-oauthlib>=0.4.1->gspread) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/opt/anaconda3/lib/python3.12/site-packages (from requests>=2.0.0->requests-
oauthlib>=0.7.0->google-auth-oauthlib>=0.4.1->gspread) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/anaconda3/lib/python3.12/site-packages (from requests>=2.0.0->requests-
oauthlib>=0.7.0->google-auth-oauthlib>=0.4.1->gspread) (2024.7.4)
Note: you may need to restart the kernel to use updated packages.
```

[3]:
```python
import gspread
import pandas as pd
from gspread_dataframe import get_as_dataframe
from oauth2client.service_account import ServiceAccountCredentials

# Set up the credentials
scope = ["https://spreadsheets.google.com/feeds", "https://www.googleapis.com/
 ↪auth/drive"]
credentials = ServiceAccountCredentials.from_json_keyfile_name("/Users/
 ↪joshuagoldberg/Desktop/BAClass/chosen-metric-3205800a614d.json", scope)

# Authorize and open the sheet
gc = gspread.authorize(credentials)
spreadsheet_url = "https://docs.google.com/spreadsheets/d/
 ↪130RPUzMh841eevpbZFV3w3aU8_HhB_Gj8ihJEfwOFGQ/edit"
spreadsheet = gc.open_by_url(spreadsheet_url)

# Load data from the first tab ("pitch_tempo")
pitch_tempo_sheet = spreadsheet.worksheet("pitch_tempo")
pitch_tempo_data = get_as_dataframe(pitch_tempo_sheet)

# Load data from the second tab ("woba")
woba_sheet = spreadsheet.worksheet("woba")
```

```
woba_data = get_as_dataframe(woba_sheet)

# Drop empty rows/columns (optional)
pitch_tempo_data = pitch_tempo_data.dropna(how="all").dropna(axis=1, how="all")
woba_data = woba_data.dropna(how="all").dropna(axis=1, how="all")

# Display the data (optional)
print("Pitch Tempo Data:")
print(pitch_tempo_data.head())

print("\nWOBA Data:")
print(woba_data.head())
```

```
Pitch Tempo Data:
   entity_id          entity_name entity_code  team_id  total_pitches  \
0   643396.0  Kiner-Falefa, Isiah         Bat    134.0          808.0
1   545341.0      Grichuk, Randal         Bat    109.0          493.0
2   623912.0      Ramírez, Harold         Bat    120.0          327.0
3   657656.0      Laureano, Ramón         Bat    144.0          477.0
4   621028.0        Newman, Kevin         Bat    109.0          442.0

   total_pitches.1  total_pitches_empty  median_seconds_empty  \
0            808.0                556.0                14.304
1            493.0                266.0                14.475
2            327.0                183.0                14.533
3            477.0                305.0                14.615
4            442.0                265.0                14.633

   total_pitches_onbase  median_seconds_empty.1  freq_hot  freq_warm  \
0                 252.0                  14.304  0.456683   0.542079
1                 227.0                  14.475  0.342799   0.655172
2                 144.0                  14.533  0.342508   0.657492
3                 172.0                  14.615  0.396226   0.603774
4                 177.0                  14.633  0.375566   0.622172

   freq_cold
0   0.001238
1   0.002028
2   0.000000
3   0.000000
4   0.002262

WOBA Data:
  last_name, first_name  player_id    year     pa  k_percent  bb_percent  \
0         DeLuca, Jonny   676356.0  2024.0  362.0       21.3         6.6
1         Buxton, Byron   621439.0  2024.0  388.0       25.5         5.2
2       Marte, Starling   516782.0  2024.0  370.0       21.9         7.3
3         Duran, Jarren   680776.0  2024.0  735.0       21.8         7.3
```

```
4          Rosario, Amed   642708.0  2024.0  346.0       21.1          2.6

    woba  xwoba  sweet_spot_percent  barrel_batted_rate  hard_hit_percent  \
0  0.269  0.272                32.7                 3.9              28.0
1  0.366  0.353                39.5                13.2              48.1
2  0.313  0.337                37.1                 6.9              42.9
3  0.357  0.342                33.8                 9.3              43.9
4  0.299  0.280                33.5                 4.6              36.2

   avg_best_speed  avg_hyper_speed  whiff_percent  swing_percent
0       96.241116        92.149059           24.0           53.8
1      102.451275        95.815195           32.1           54.0
2      101.202056        94.944189           28.4           50.2
3      101.928156        95.341113           24.8           48.3
4       98.559932        93.290414           24.6           55.2
```

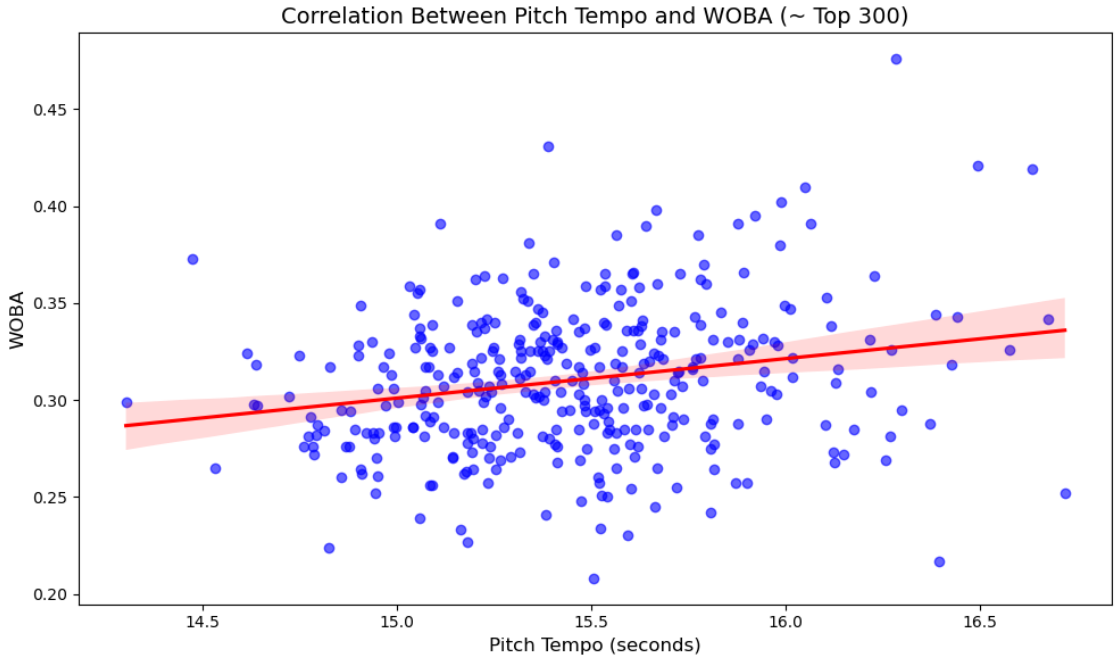
[5]:
```python
import pandas as pd

# Assuming you have already loaded the datasets as DataFrames
# pitch_tempo_data = pd.read_csv("path_to_pitch_tempo_data.csv")
# woba_data = pd.read_csv("path_to_woba_data.csv")

# Rename the player name columns to a common name (e.g., 'player_name')
pitch_tempo_data.rename(columns={'entity_name': 'player_name',␣
 ↪'median_seconds_empty': 'pitch_tempo_empty'}, inplace=True)
woba_data.rename(columns={'last_name, first_name': 'player_name'}, inplace=True)

# Merge the datasets on 'player_name'
merged_data = pd.merge(pitch_tempo_data, woba_data, on="player_name",␣
 ↪how="inner")

# Select the desired columns and create a copy to avoid SettingWithCopyWarning
final_data = merged_data[['player_name', 'pitch_tempo_empty',␣
 ↪'total_pitches_empty', 'woba']].copy()

# Add a percentile column for total_pitches_empty
final_data['total_pitches_empty_percentile'] =␣
 ↪final_data['total_pitches_empty'].rank(pct=True)

# Sort by pitch_tempo_empty in descending order
final_data_sorted = final_data.sort_values(by='total_pitches_empty',␣
 ↪ascending=False)

# Display the resulting dataframe
print(final_data_sorted)
```

```
          player_name  pitch_tempo_empty  total_pitches_empty   woba  \
311    Schwarber, Kyle             15.892                967.0  0.366
```

```
56     India, Jonathan             15.058                953.0  0.333
354         Soto, Juan             16.495                947.0  0.421
145  Henderson, Gunnar             15.339                918.0  0.381
143    Torres, Gleyber             15.330                884.0  0.313
..                 …                …                  …    …
328  Soderstrom, Tyler             16.018                204.0  0.322
191        Ahmed, Nick             15.476                202.0  0.248
90        Carter, Evan             15.183                200.0  0.278
262    Carpenter, Matt             15.663                196.0  0.303
2      Ramírez, Harold             14.533                183.0  0.265

     total_pitches_empty_percentile
311                        1.000000
56                         0.997214
354                        0.994429
145                        0.991643
143                        0.988858
..                              …
328                        0.013928
191                        0.011142
90                         0.008357
262                        0.005571
2                          0.002786

[359 rows x 5 columns]
```

[7]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming you already have the `final_data_sorted` DataFrame

# Plot settings
plt.figure(figsize=(10, 6))  # Set the figure size

# Create the scatterplot with regression line
sns.regplot(
    data=final_data_sorted,
    x='pitch_tempo_empty',
    y='woba',
    scatter_kws={'color': 'blue', 'alpha': 0.6},  # Customize scatter points
    line_kws={'color': 'red'},  # Customize regression line
)

# Add titles and labels
plt.title('Correlation Between Pitch Tempo and WOBA (~ Top 300)', fontsize=14)
plt.xlabel('Pitch Tempo (seconds)', fontsize=12)
```

```
plt.ylabel('WOBA', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```



[9]:
```
import numpy as np
from scipy.stats import pearsonr

# Calculate correlation coefficient (Pearson's r) and p-value
correlation_coefficient, p_value =␣
 ↪pearsonr(final_data_sorted['pitch_tempo_empty'], final_data_sorted['woba'])

# Print key metrics
print(f"Correlation Coefficient (r): {correlation_coefficient:.4f}")
print(f"P-value: {p_value:.4e}")

# Additional statistics
x_mean = final_data_sorted['pitch_tempo_empty'].mean()
y_mean = final_data_sorted['woba'].mean()
x_std = final_data_sorted['pitch_tempo_empty'].std()
y_std = final_data_sorted['woba'].std()

print("\nDescriptive Statistics:")
print(f"Mean of Pitch Tempo: {x_mean:.2f} seconds")
```

```
print(f"Mean of WOBA: {y_mean:.4f}")
print(f"Standard Deviation of Pitch Tempo: {x_std:.2f} seconds")
print(f"Standard Deviation of WOBA: {y_std:.4f}")
```

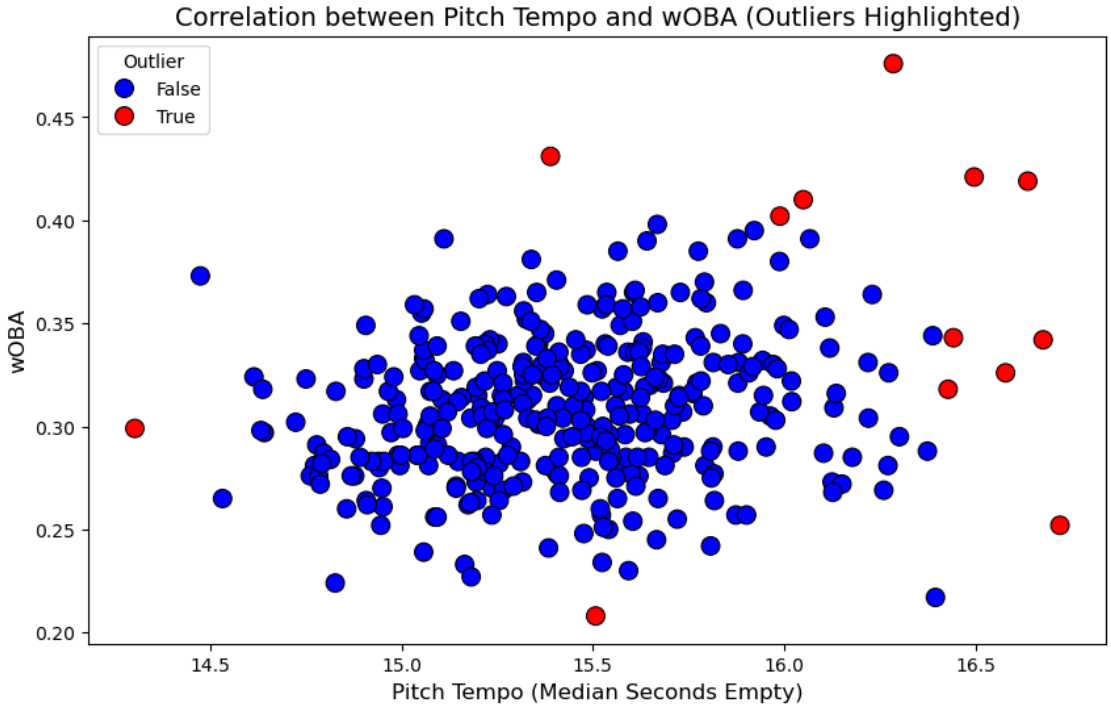
```
Correlation Coefficient (r): 0.2218
P-value: 2.2260e-05

Descriptive Statistics:
Mean of Pitch Tempo: 15.45 seconds
Mean of WOBA: 0.3101
Standard Deviation of Pitch Tempo: 0.41 seconds
Standard Deviation of WOBA: 0.0373
```

[11]:
```
# Function to identify outliers using the IQR method
def detect_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data[column] < lower_bound) | (data[column] > upper_bound)]
    return outliers, lower_bound, upper_bound

# Detect outliers for 'pitch_tempo_empty' and 'woba'
outliers_pitch_tempo, lower_pitch, upper_pitch =␣
 ↪detect_outliers_iqr(final_data_sorted, 'pitch_tempo_empty')
outliers_woba, lower_woba, upper_woba = detect_outliers_iqr(final_data_sorted,␣
 ↪'woba')

# Print outlier details
print("Outliers in 'pitch_tempo_empty':")
print(outliers_pitch_tempo[['player_name', 'pitch_tempo_empty']])
print(f"Lower Bound: {lower_pitch:.2f}, Upper Bound: {upper_pitch:.2f}\n")

print("Outliers in 'woba':")
print(outliers_woba[['player_name', 'woba']])
print(f"Lower Bound: {lower_woba:.4f}, Upper Bound: {upper_woba:.4f}")
```

```
Outliers in 'pitch_tempo_empty':
            player_name  pitch_tempo_empty
354           Soto, Juan             16.495
352       Martinez, J.D.             16.427
353    Walker, Christian             16.441
0    Kiner-Falefa, Isiah             14.304
356         Tucker, Kyle             16.635
355       Realmuto, J.T.             16.577
358         Duvall, Adam             16.719
357       Casas, Triston             16.675
```

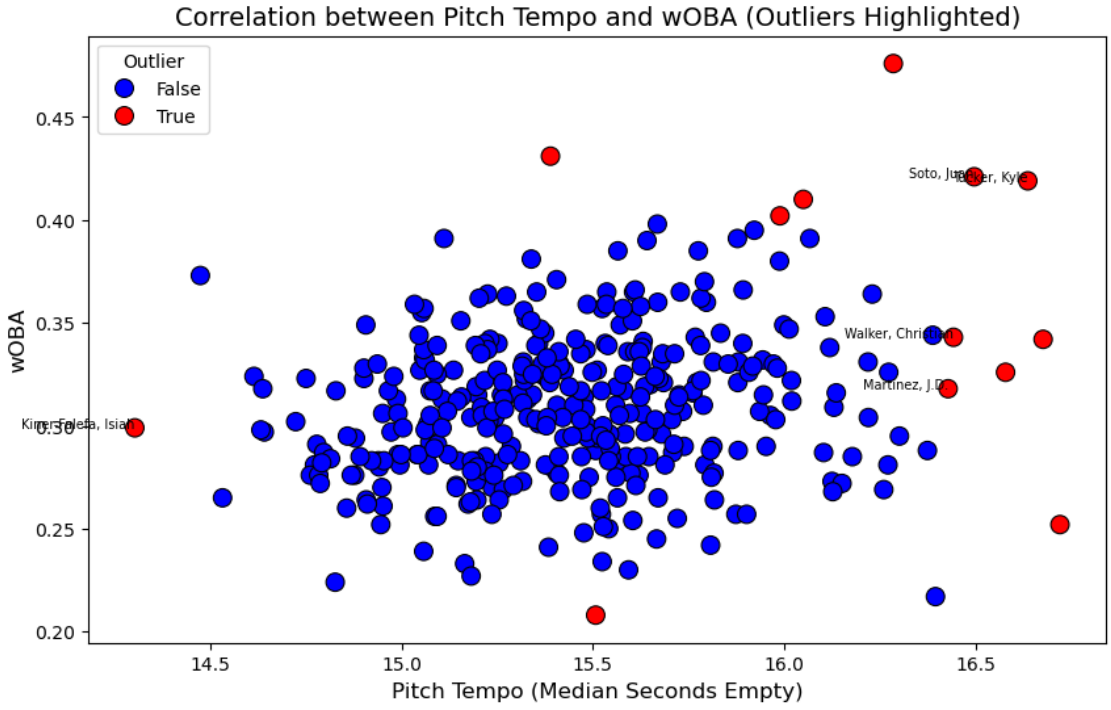
```
Lower Bound: 14.44, Upper Bound: 16.42

Outliers in 'woba':
         player_name   woba
354       Soto, Juan  0.421
166   Ohtani, Shohei  0.431
347     Judge, Aaron  0.476
330  Witt Jr., Bobby  0.410
325  Alvarez, Yordan  0.402
356     Tucker, Kyle  0.419
201    Anderson, Tim  0.208
Lower Bound: 0.2147, Upper Bound: 0.4008
```

[13]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Detect outliers using IQR method (you can choose other methods too)
outliers_pitch_tempo, lower_pitch, upper_pitch =␣
 ↪detect_outliers_iqr(final_data_sorted, 'pitch_tempo_empty')
outliers_woba, lower_woba, upper_woba = detect_outliers_iqr(final_data_sorted,␣
 ↪'woba')

# Merge the outliers back into the dataset for easy visualization
final_data_sorted['is_outlier'] = final_data_sorted['player_name'].
 ↪isin(outliers_pitch_tempo['player_name']) | final_data_sorted['player_name'].
 ↪isin(outliers_woba['player_name'])

# Set up the correlation plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=final_data_sorted, x='pitch_tempo_empty', y='woba',␣
 ↪hue='is_outlier', palette={True: 'red', False: 'blue'}, marker='o', s=100,␣
 ↪edgecolor='black')

# Add labels and title
plt.title("Correlation between Pitch Tempo and wOBA (Outliers Highlighted)",␣
 ↪fontsize=14)
plt.xlabel("Pitch Tempo (Median Seconds Empty)", fontsize=12)
plt.ylabel("wOBA", fontsize=12)
plt.legend(title='Outlier', loc='upper left')

# Display the plot
plt.show()
```

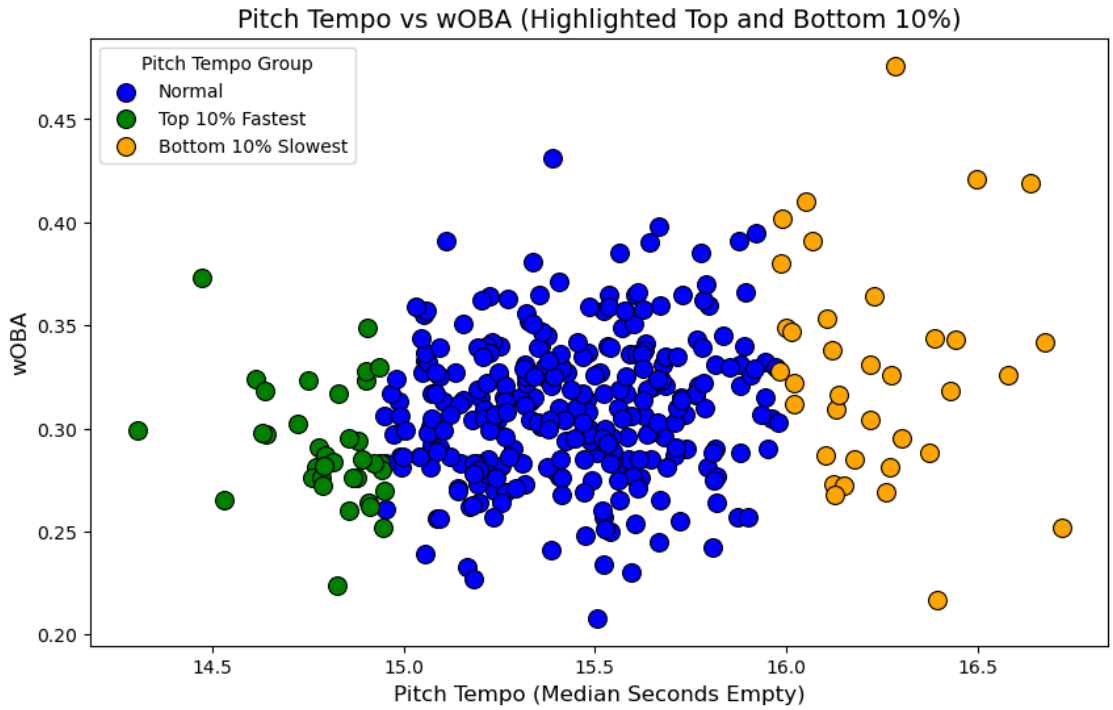


[65]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Function to detect outliers using the IQR method
def detect_outliers_iqr(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)]
    return outliers, lower_bound, upper_bound

# Detect outliers for both pitch_tempo_empty and woba
outliers_pitch_tempo, lower_pitch, upper_pitch =␣
 ↪detect_outliers_iqr(final_data_sorted, 'pitch_tempo_empty')
outliers_woba, lower_woba, upper_woba = detect_outliers_iqr(final_data_sorted,␣
 ↪'woba')

# Merge the outliers back into the dataset for easy visualization
final_data_sorted['is_outlier'] = final_data_sorted['player_name'].
 ↪isin(outliers_pitch_tempo['player_name']) | final_data_sorted['player_name'].
 ↪isin(outliers_woba['player_name'])
```

```
# Set up the correlation plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=final_data_sorted, x='pitch_tempo_empty', y='woba',␣
 ↪hue='is_outlier', palette={True: 'red', False: 'blue'}, marker='o', s=100,␣
 ↪edgecolor='black')

# Annotate extreme outliers (top 5 most extreme outliers based on both␣
 ↪variables)
outliers_combined = pd.concat([outliers_pitch_tempo, outliers_woba]).
 ↪drop_duplicates()
top_outliers = outliers_combined.head(5)  # Get the top 5 extreme outliers, you␣
 ↪can adjust this as needed

for _, row in top_outliers.iterrows():
    plt.text(row['pitch_tempo_empty'], row['woba'], row['player_name'],␣
 ↪fontsize=7, ha='right', color='black')

# Add labels and title
plt.title("Correlation between Pitch Tempo and wOBA (Outliers Highlighted)",␣
 ↪fontsize=14)
plt.xlabel("Pitch Tempo (Median Seconds Empty)", fontsize=12)
plt.ylabel("wOBA", fontsize=12)
plt.legend(title='Outlier', loc='upper left')

# Display the plot
plt.show()
```

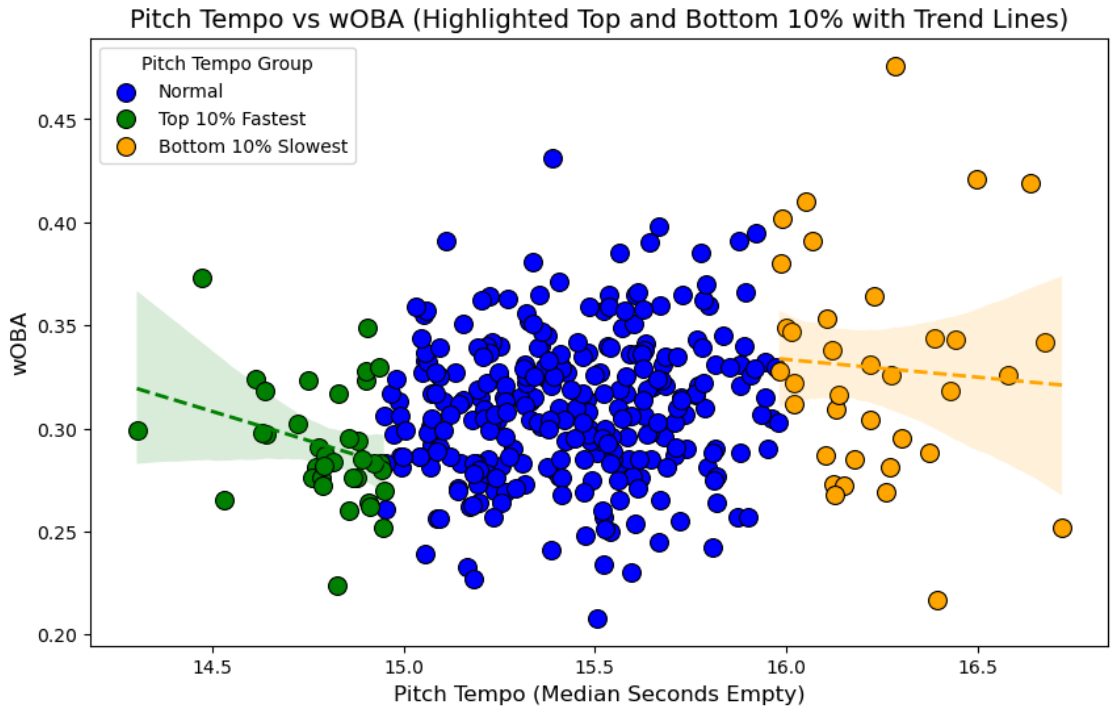


[19]:
```
# Create a new column 'highlight' with corrected labels
final_data_sorted['highlight'] = final_data_sorted['pitch_tempo_empty'].apply(
    lambda x: 'bottom_10%' if x >= top_10_percentile else ('top_10%' if x <=␣
 ↪bottom_10_percentile else 'normal')
)

# Set up the correlation plot
plt.figure(figsize=(10, 6))

# Plot the normal points in blue
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'normal'],
                x='pitch_tempo_empty', y='woba', color='blue', label='Normal',␣
 ↪marker='o', s=100, edgecolor='black')

# Plot the corrected bottom 10% (now "Top 10% Fastest") in green
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'top_10%'],
                x='pitch_tempo_empty', y='woba', color='green', label='Top 10%␣
 ↪Fastest', marker='o', s=100, edgecolor='black')

# Plot the corrected top 10% (now "Bottom 10% Slowest") in orange
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'bottom_10%'],
                x='pitch_tempo_empty', y='woba', color='orange', label='Bottom␣
 ↪10% Slowest', marker='o', s=100, edgecolor='black')

# Add labels and title
plt.title("Pitch Tempo vs wOBA (Highlighted Top and Bottom 10%)", fontsize=14)
plt.xlabel("Pitch Tempo (Median Seconds Empty)", fontsize=12)
plt.ylabel("wOBA", fontsize=12)

# Display the legend and plot
plt.legend(title='Pitch Tempo Group', loc='upper left')
plt.show()
```



[23]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the 10th and 90th percentiles of pitch_tempo_empty
top_10_percentile = final_data_sorted['pitch_tempo_empty'].quantile(0.90)
bottom_10_percentile = final_data_sorted['pitch_tempo_empty'].quantile(0.10)

# Create a new column 'highlight' with corrected labels
final_data_sorted['highlight'] = final_data_sorted['pitch_tempo_empty'].apply(
    lambda x: 'bottom_10%' if x >= top_10_percentile else ('top_10%' if x <=␣
 ↪bottom_10_percentile else 'normal')
)

# Set up the plot
plt.figure(figsize=(10, 6))

# Plot the normal points in blue
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'normal'],
                x='pitch_tempo_empty', y='woba', color='blue', label='Normal',␣
 ↪marker='o', s=100, edgecolor='black')

# Plot the corrected bottom 10% (now "Top 10% Fastest") in green
```

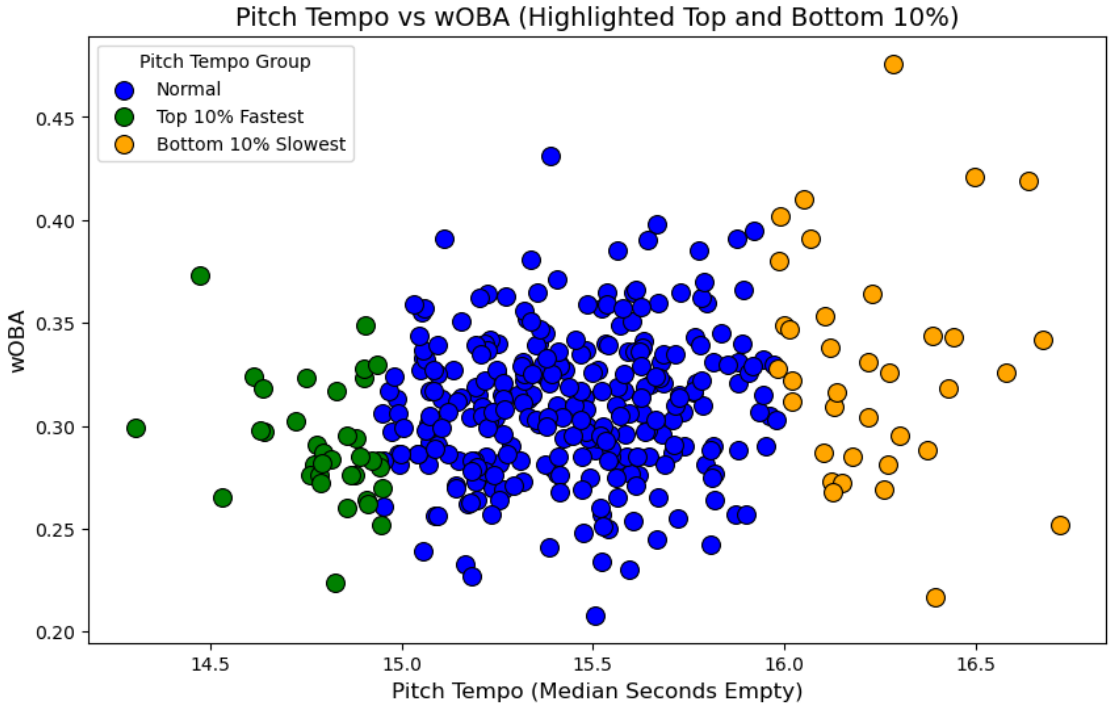
```
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'top_10%'],
                x='pitch_tempo_empty', y='woba', color='green', label='Top 10%␣
 ↪Fastest', marker='o', s=100, edgecolor='black')

# Plot the corrected top 10% (now "Bottom 10% Slowest") in orange
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'bottom_10%'],
                x='pitch_tempo_empty', y='woba', color='orange', label='Bottom␣
 ↪10% Slowest', marker='o', s=100, edgecolor='black')

# Add trend line for top 10% (fastest pitch tempos)
sns.regplot(data=final_data_sorted[final_data_sorted['highlight'] == 'top_10%'],
            x='pitch_tempo_empty', y='woba', scatter=False, color='green',␣
 ↪line_kws={'linewidth': 2, 'linestyle': '--'})

# Add trend line for bottom 10% (slowest pitch tempos)
sns.regplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'bottom_10%'],
            x='pitch_tempo_empty', y='woba', scatter=False, color='orange',␣
 ↪line_kws={'linewidth': 2, 'linestyle': '--'})

# Add labels and title
plt.title("Pitch Tempo vs wOBA (Highlighted Top and Bottom 10% with Trend␣
 ↪Lines)", fontsize=14)
plt.xlabel("Pitch Tempo (Median Seconds Empty)", fontsize=12)
plt.ylabel("wOBA", fontsize=12)

# Display the legend and plot
plt.legend(title='Pitch Tempo Group', loc='upper left')
plt.show()
```



[27]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the 10th and 90th percentiles of pitch_tempo_empty
top_10_percentile = final_data_sorted['pitch_tempo_empty'].quantile(0.90)
bottom_10_percentile = final_data_sorted['pitch_tempo_empty'].quantile(0.10)

# Create a new column 'highlight' with corrected labels
final_data_sorted['highlight'] = final_data_sorted['pitch_tempo_empty'].apply(
    lambda x: 'bottom_10%' if x >= top_10_percentile else ('top_10%' if x <=␣
 ↪bottom_10_percentile else 'normal')
)

# Set up the correlation plot
plt.figure(figsize=(10, 6))

# Plot the normal points in blue
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'normal'],
                x='pitch_tempo_empty', y='woba', color='blue', label='Normal',␣
 ↪marker='o', s=100, edgecolor='black')

# Plot the corrected bottom 10% (now "Top 10% Fastest") in green
```

```
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'top_10%'],
                x='pitch_tempo_empty', y='woba', color='green', label='Top 10%␣
 ↪Fastest', marker='o', s=100, edgecolor='black')

# Plot the corrected top 10% (now "Bottom 10% Slowest") in orange
sns.scatterplot(data=final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'bottom_10%'],
                x='pitch_tempo_empty', y='woba', color='orange', label='Bottom␣
 ↪10% Slowest', marker='o', s=100, edgecolor='black')

# Add labels and title
plt.title("Pitch Tempo vs wOBA (Highlighted Top and Bottom 10%)", fontsize=14)
plt.xlabel("Pitch Tempo (Median Seconds Empty)", fontsize=12)
plt.ylabel("wOBA", fontsize=12)

# Display the legend and plot
plt.legend(title='Pitch Tempo Group', loc='upper left')
plt.show()

# Print statistics for overall dataset, top 10%, and bottom 10%

# Overall statistics
overall_stats = final_data_sorted['pitch_tempo_empty'].describe().round(2)

# Top 10% statistics (now referring to fastest pitch tempos)
top_10_stats = final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'top_10%']['pitch_tempo_empty'].describe().round(2)

# Bottom 10% statistics (now referring to slowest pitch tempos)
bottom_10_stats = final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'bottom_10%']['pitch_tempo_empty'].describe().round(2)

# Calculate variability: standard deviation and IQR
overall_std = round(final_data_sorted['pitch_tempo_empty'].std(), 2)
top_10_std = round(final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'top_10%']['pitch_tempo_empty'].std(), 2)
bottom_10_std = round(final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'bottom_10%']['pitch_tempo_empty'].std(), 2)

overall_iqr = final_data_sorted['pitch_tempo_empty'].quantile(0.75) -␣
 ↪final_data_sorted['pitch_tempo_empty'].quantile(0.25)
top_10_iqr = final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'top_10%']['pitch_tempo_empty'].quantile(0.75) -␣
 ↪final_data_sorted[final_data_sorted['highlight'] ==␣
 ↪'top_10%']['pitch_tempo_empty'].quantile(0.25)
```

```
bottom_10_iqr = final_data_sorted[final_data_sorted['highlight'] ==␣
  ↪'bottom_10%']['pitch_tempo_empty'].quantile(0.75) -␣
  ↪final_data_sorted[final_data_sorted['highlight'] ==␣
  ↪'bottom_10%']['pitch_tempo_empty'].quantile(0.25)

# Print rounded statistics and variability
print("\nOverall Statistics:")
print(overall_stats)
print(f"Overall Standard Deviation: {overall_std}")
print(f"Overall IQR: {overall_iqr:.2f}")

print("\nTop 10% Fastest Pitch Tempos Statistics:")
print(top_10_stats)
print(f"Top 10% Standard Deviation: {top_10_std}")
print(f"Top 10% IQR: {top_10_iqr:.2f}")

print("\nBottom 10% Slowest Pitch Tempos Statistics:")
print(bottom_10_stats)
print(f"Bottom 10% Standard Deviation: {bottom_10_std}")
print(f"Bottom 10% IQR: {bottom_10_iqr:.2f}")
```




```
Overall Statistics:
count    359.00
```

```
mean      15.45
std        0.41
min       14.30
25%       15.18
50%       15.42
75%       15.68
max       16.72
Name: pitch_tempo_empty, dtype: float64
Overall Standard Deviation: 0.41
Overall IQR: 0.50

Top 10% Fastest Pitch Tempos Statistics:
count    36.00
mean     14.80
std       0.15
min      14.30
25%      14.76
50%      14.83
75%      14.90
max      14.95
Name: pitch_tempo_empty, dtype: float64
Top 10% Standard Deviation: 0.15
Top 10% IQR: 0.14

Bottom 10% Slowest Pitch Tempos Statistics:
count    36.00
mean     16.23
std       0.21
min      15.98
25%      16.09
50%      16.20
75%      16.38
max      16.72
Name: pitch_tempo_empty, dtype: float64
Bottom 10% Standard Deviation: 0.21
Bottom 10% IQR: 0.28
```

[5]:
```
pip install -U notebook-as-pdf
```

```
Collecting notebook-as-pdf
  Downloading notebook_as_pdf-0.5.0-py3-none-any.whl.metadata (2.4 kB)
Requirement already satisfied: nbconvert in /opt/anaconda3/lib/python3.12/site-
packages (from notebook-as-pdf) (7.10.0)
Collecting pyppeteer (from notebook-as-pdf)
  Downloading pyppeteer-2.0.0-py3-none-any.whl.metadata (7.1 kB)
Collecting PyPDF2 (from notebook-as-pdf)
  Downloading pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: beautifulsoup4 in
```

```
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(4.12.3)
Requirement already satisfied: bleach!=5.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(4.1.0)
Requirement already satisfied: defusedxml in /opt/anaconda3/lib/python3.12/site-
packages (from nbconvert->notebook-as-pdf) (0.7.1)
Requirement already satisfied: jinja2>=3.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(3.1.4)
Requirement already satisfied: jupyter-core>=4.7 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(5.7.2)
Requirement already satisfied: jupyterlab-pygments in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(0.1.2)
Requirement already satisfied: markupsafe>=2.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(2.1.3)
Requirement already satisfied: mistune<4,>=2.0.3 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(2.0.4)
Requirement already satisfied: nbclient>=0.5.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(0.8.0)
Requirement already satisfied: nbformat>=5.7 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(5.9.2)
Requirement already satisfied: packaging in /opt/anaconda3/lib/python3.12/site-
packages (from nbconvert->notebook-as-pdf) (23.2)
Requirement already satisfied: pandocfilters>=1.4.1 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(1.5.0)
Requirement already satisfied: pygments>=2.4.1 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(2.15.1)
Requirement already satisfied: tinycss2 in /opt/anaconda3/lib/python3.12/site-
packages (from nbconvert->notebook-as-pdf) (1.2.1)
Requirement already satisfied: traitlets>=5.1 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert->notebook-as-pdf)
(5.14.3)
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer->notebook-as-pdf)
(1.4.4)
Requirement already satisfied: certifi>=2023 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer->notebook-as-pdf)
(2024.7.4)
Requirement already satisfied: importlib-metadata>=1.4 in
```

```
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer->notebook-as-pdf)
(7.0.1)
Collecting pyee<12.0.0,>=11.0.0 (from pyppeteer->notebook-as-pdf)
  Downloading pyee-11.1.1-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer->notebook-as-pdf)
(4.66.4)
Collecting urllib3<2.0.0,>=1.25.8 (from pyppeteer->notebook-as-pdf)
  Downloading urllib3-1.26.20-py2.py3-none-any.whl.metadata (50 kB)
                           50.1/50.1 kB
882.0 kB/s eta 0:00:00 0:00:01
Collecting websockets<11.0,>=10.0 (from pyppeteer->notebook-as-pdf)
  Downloading websockets-10.4.tar.gz (84 kB)
                           84.9/84.9 kB
2.0 MB/s eta 0:00:00a 0:00:01
  Preparing metadata (setup.py) … done
Requirement already satisfied: six>=1.9.0 in
/opt/anaconda3/lib/python3.12/site-packages (from
bleach!=5.0.0->nbconvert->notebook-as-pdf) (1.16.0)
Requirement already satisfied: webencodings in
/opt/anaconda3/lib/python3.12/site-packages (from
bleach!=5.0.0->nbconvert->notebook-as-pdf) (0.5.1)
Requirement already satisfied: zipp>=0.5 in /opt/anaconda3/lib/python3.12/site-
packages (from importlib-metadata>=1.4->pyppeteer->notebook-as-pdf) (3.17.0)
Requirement already satisfied: platformdirs>=2.5 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-
core>=4.7->nbconvert->notebook-as-pdf) (3.10.0)
Requirement already satisfied: jupyter-client>=6.1.12 in
/opt/anaconda3/lib/python3.12/site-packages (from
nbclient>=0.5.0->nbconvert->notebook-as-pdf) (8.6.0)
Requirement already satisfied: fastjsonschema in
/opt/anaconda3/lib/python3.12/site-packages (from
nbformat>=5.7->nbconvert->notebook-as-pdf) (2.16.2)
Requirement already satisfied: jsonschema>=2.6 in
/opt/anaconda3/lib/python3.12/site-packages (from
nbformat>=5.7->nbconvert->notebook-as-pdf) (4.19.2)
Requirement already satisfied: typing-extensions in
/opt/anaconda3/lib/python3.12/site-packages (from
pyee<12.0.0,>=11.0.0->pyppeteer->notebook-as-pdf) (4.11.0)
Requirement already satisfied: soupsieve>1.2 in
/opt/anaconda3/lib/python3.12/site-packages (from
beautifulsoup4->nbconvert->notebook-as-pdf) (2.5)
Requirement already satisfied: attrs>=22.2.0 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (2023.7.1)
```

```
Requirement already satisfied: referencing>=0.28.4 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert->notebook-as-pdf) (0.10.6)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert->notebook-as-pdf) (2.9.0.post0)
Requirement already satisfied: pyzmq>=23.0 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert->notebook-as-pdf) (25.1.2)
Requirement already satisfied: tornado>=6.2 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert->notebook-as-pdf) (6.4.1)
Downloading notebook_as_pdf-0.5.0-py3-none-any.whl (6.5 kB)
Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
                           232.6/232.6 kB
5.4 MB/s eta 0:00:00ta 0:00:01
Downloading pyppeteer-2.0.0-py3-none-any.whl (82 kB)
                           82.9/82.9 kB
3.1 MB/s eta 0:00:00
Downloading pyee-11.1.1-py3-none-any.whl (15 kB)
Downloading urllib3-1.26.20-py2.py3-none-any.whl (144 kB)
                           144.2/144.2 kB
5.4 MB/s eta 0:00:00
Building wheels for collected packages: websockets
  Building wheel for websockets (setup.py) … done
  Created wheel for websockets:
filename=websockets-10.4-cp312-cp312-macosx_11_0_arm64.whl size=97696
sha256=4363f12145bea595cb49335d93d74fdc62e7ef20277d7649ce2e324687316080
  Stored in directory: /Users/joshuagoldberg/Library/Caches/pip/wheels/80/cf/6d/
5d7e4c920cb41925a178b2d2621889c520d648bab487b1d7fd
Successfully built websockets
Installing collected packages: websockets, urllib3, PyPDF2, pyee, pyppeteer,
notebook-as-pdf
  Attempting uninstall: urllib3
    Found existing installation: urllib3 2.2.2
    Uninstalling urllib3-2.2.2:
      Successfully uninstalled urllib3-2.2.2
Successfully installed PyPDF2-3.0.1 notebook-as-pdf-0.5.0 pyee-11.1.1
pyppeteer-2.0.0 urllib3-1.26.20 websockets-10.4
Note: you may need to restart the kernel to use updated packages.
```

[7]:
```
!pip install pyppeteer
```

```
Requirement already satisfied: pyppeteer in /opt/anaconda3/lib/python3.12/site-
packages (2.0.0)
```

```
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer) (1.4.4)
Requirement already satisfied: certifi>=2023 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer) (2024.7.4)
Requirement already satisfied: importlib-metadata>=1.4 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer) (7.0.1)
Requirement already satisfied: pyee<12.0.0,>=11.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer) (11.1.1)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer) (4.66.4)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer) (1.26.20)
Requirement already satisfied: websockets<11.0,>=10.0 in
/opt/anaconda3/lib/python3.12/site-packages (from pyppeteer) (10.4)
Requirement already satisfied: zipp>=0.5 in /opt/anaconda3/lib/python3.12/site-
packages (from importlib-metadata>=1.4->pyppeteer) (3.17.0)
Requirement already satisfied: typing-extensions in
/opt/anaconda3/lib/python3.12/site-packages (from
pyee<12.0.0,>=11.0.0->pyppeteer) (4.11.0)
```

[9]:
```
!pip install nbconvert
```

```
Requirement already satisfied: nbconvert in /opt/anaconda3/lib/python3.12/site-
packages (7.10.0)
Requirement already satisfied: beautifulsoup4 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (4.12.3)
Requirement already satisfied: bleach!=5.0.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (4.1.0)
Requirement already satisfied: defusedxml in /opt/anaconda3/lib/python3.12/site-
packages (from nbconvert) (0.7.1)
Requirement already satisfied: jinja2>=3.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (3.1.4)
Requirement already satisfied: jupyter-core>=4.7 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (5.7.2)
Requirement already satisfied: jupyterlab-pygments in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: markupsafe>=2.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (2.1.3)
Requirement already satisfied: mistune<4,>=2.0.3 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (2.0.4)
Requirement already satisfied: nbclient>=0.5.0 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (0.8.0)
Requirement already satisfied: nbformat>=5.7 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (5.9.2)
Requirement already satisfied: packaging in /opt/anaconda3/lib/python3.12/site-
packages (from nbconvert) (23.2)
Requirement already satisfied: pandocfilters>=1.4.1 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (1.5.0)
```

```
Requirement already satisfied: pygments>=2.4.1 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (2.15.1)
Requirement already satisfied: tinycss2 in /opt/anaconda3/lib/python3.12/site-
packages (from nbconvert) (1.2.1)
Requirement already satisfied: traitlets>=5.1 in
/opt/anaconda3/lib/python3.12/site-packages (from nbconvert) (5.14.3)
Requirement already satisfied: six>=1.9.0 in /opt/anaconda3/lib/python3.12/site-
packages (from bleach!=5.0.0->nbconvert) (1.16.0)
Requirement already satisfied: webencodings in
/opt/anaconda3/lib/python3.12/site-packages (from bleach!=5.0.0->nbconvert)
(0.5.1)
Requirement already satisfied: platformdirs>=2.5 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-core>=4.7->nbconvert)
(3.10.0)
Requirement already satisfied: jupyter-client>=6.1.12 in
/opt/anaconda3/lib/python3.12/site-packages (from nbclient>=0.5.0->nbconvert)
(8.6.0)
Requirement already satisfied: fastjsonschema in
/opt/anaconda3/lib/python3.12/site-packages (from nbformat>=5.7->nbconvert)
(2.16.2)
Requirement already satisfied: jsonschema>=2.6 in
/opt/anaconda3/lib/python3.12/site-packages (from nbformat>=5.7->nbconvert)
(4.19.2)
Requirement already satisfied: soupsieve>1.2 in
/opt/anaconda3/lib/python3.12/site-packages (from beautifulsoup4->nbconvert)
(2.5)
Requirement already satisfied: attrs>=22.2.0 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in
/opt/anaconda3/lib/python3.12/site-packages (from
jsonschema>=2.6->nbformat>=5.7->nbconvert) (0.10.6)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (2.9.0.post0)
Requirement already satisfied: pyzmq>=23.0 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (25.1.2)
Requirement already satisfied: tornado>=6.2 in
/opt/anaconda3/lib/python3.12/site-packages (from jupyter-
client>=6.1.12->nbclient>=0.5.0->nbconvert) (6.4.1)
```

[21]:
```
pip install pandoc
```

```
Collecting pandoc
  Downloading pandoc-2.4.tar.gz (34 kB)
  Preparing metadata (setup.py) … done
Collecting plumbum (from pandoc)
  Downloading plumbum-1.9.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: ply in /opt/anaconda3/lib/python3.12/site-
packages (from pandoc) (3.11)
Downloading plumbum-1.9.0-py3-none-any.whl (127 kB)
                         128.0/128.0 kB
3.5 MB/s eta 0:00:00
Building wheels for collected packages: pandoc
  Building wheel for pandoc (setup.py) … done
  Created wheel for pandoc: filename=pandoc-2.4-py3-none-any.whl
size=34794
sha256=5efed8b5649d45f76d55b7761b35c71f350ed20604c7eb066b83fb056c427719
  Stored in directory: /Users/joshuagoldberg/Library/Caches/pip/wheels/9c/2f/9f/
b1aac8c3e74b4ee327dc8c6eac5128996f9eadf586e2c0ba67
Successfully built pandoc
Installing collected packages: plumbum, pandoc
Successfully installed pandoc-2.4 plumbum-1.9.0
Note: you may need to restart the kernel to use updated packages.
```

[1]:
```
!jupyter nbconvert --to pdf chosen_metric.ipynb
```

```
[NbConvertApp] Converting notebook chosen_metric.ipynb to pdf
[NbConvertApp] Support files will be in chosen_metric_files/
[NbConvertApp] Making directory ./chosen_metric_files
[NbConvertApp] Writing 105780 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 593595 bytes to chosen_metric.pdf
```

[ ]:
```

```