In [1]:

```
import pybaseball as pyb
from pybaseball import statcast, pitching_stats, playerid_lookup, stat
import numpy as np
import math
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import glob
import os
import re
import unicodedata
from datetime import datetime
from itertools import groupby
from operator import itemgetter
from fuzzywuzzy import process
```

```
C:\Users\johns\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:6
0: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck'
(version '1.3.5' currently installed).
  from pandas.core import (
```

In [2]:
```
hist_tj_df = pd.read_csv('~/Documents/Flatiron/Project_5_/data/hist_tj
hist_tj_df
```

Out[2]:

	Name	Age	Year	Throws	IP	G	GS	CG	SHO	sDR	Career Start	Career End	Inact Ye
240	steve blass	31	1973	1	88.2	23	18	1	0	1	1972	1973	
285	eddie fisher	36	1973	1	117.2	32	16	2	0	9	1972	1973	
385	milt pappas	34	1973	1	162.0	30	29	1	1	8	1972	1973	
463	steve arlin	28	1974	1	107.2	27	22	2	0	9	1972	1974	
589	ernie mcanally	27	1974	1	128.2	25	21	5	2	6	1972	1974	
...	...	...	...	...	...	...	...	...	...	...	...	...	
15334	zack wheeler	33	2023	1	192.0	32	32	0	0	0	2013	2023	[20 20
15339	trevor williams	31	2023	1	144.1	30	30	0	0	0	2016	2023	
15346	alex wood	32	2023	0	97.2	29	12	0	0	1	2013	2023	
15348	brandon woodruff	30	2023	1	67.0	11	11	1	1	0	2017	2023	
15350	ryan yarbrough	31	2023	0	89.2	25	9	0	0	0	2018	2023	

1244 rows × 20 columns

In [3]:
```
1 hist_tj_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1244 entries, 240 to 15350
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Name             1244 non-null   object
 1   Age              1244 non-null   int64
 2   Year             1244 non-null   int64
 3   Throws           1244 non-null   int64
 4   IP               1244 non-null   float64
 5   G                1244 non-null   int64
 6   GS               1244 non-null   int64
 7   CG               1244 non-null   int64
 8   SHO              1244 non-null   int64
 9   sDR              1244 non-null   int64
 10  Career Start     1244 non-null   int64
 11  Career End       1244 non-null   int64
 12  Inactive Years   1244 non-null   object
 13  Surgery          1244 non-null   float64
 14  TJ Surgery Date  1244 non-null   object
 15  Surgeon(s)       197 non-null    object
 16  Country          268 non-null    object
 17  Level            268 non-null    object
 18  Total_IP         1244 non-null   float64
 19  TJ Surgery Year  1244 non-null   object
dtypes: float64(3), int64(10), object(7)
memory usage: 204.1+ KB
```

In [4]:
```
pitcher_data_df = hist_tj_df
pitcher_data_df
```

Out[4]:

	Name	Age	Year	Throws	IP	G	GS	CG	SHO	sDR	Career Start	Career End	Inact Ye
240	steve blass	31	1973	1	88.2	23	18	1	0	1	1972	1973	
285	eddie fisher	36	1973	1	117.2	32	16	2	0	9	1972	1973	
385	milt pappas	34	1973	1	162.0	30	29	1	1	8	1972	1973	
463	steve arlin	28	1974	1	107.2	27	22	2	0	9	1972	1974	
589	ernie mcanally	27	1974	1	128.2	25	21	5	2	6	1972	1974	
...	...	...	...	...	...	...	...	...	...	...	...	...	
15334	zack wheeler	33	2023	1	192.0	32	32	0	0	0	2013	2023	[20 20
15339	trevor williams	31	2023	1	144.1	30	30	0	0	0	2016	2023	
15346	alex wood	32	2023	0	97.2	29	12	0	0	1	2013	2023	
15348	brandon woodruff	30	2023	1	67.0	11	11	1	1	0	2017	2023	
15350	ryan yarbrough	31	2023	0	89.2	25	9	0	0	0	2018	2023	

1244 rows × 20 columns

Goal: Take previous DF and attach identification ('key_mlbam') back to the players.

In [5]:

```
# Function to fetch key_mlbam
def fetch_key_mlbam(row):
    try:
        # Splitting the name into first and last name
        first_name, last_name = row['Name'].split(' ')[0], ' '.join(ro
        # Fetching player ID
        player_id_df = playerid_lookup(last_name, first_name)
        # Assuming the first result is the correct one, adjust as nece
        key_mlbam = player_id_df.iloc[0]['key_mlbam']
        return key_mlbam
    except Exception as e:
        print(f"Error fetching key_mlbam for {row['Name']}: {e}")
        return pd.NA

# Apply the function to each row and create a new column 'key_mlbam'
pitcher_data_df['key_mlbam'] = pitcher_data_df.apply(fetch_key_mlbam,

# Display the updated DataFrame
pitcher_data_df.head()
```

```
Gathering player lookup table. This may take a moment.
Error fetching key_mlbam for blue moon odom: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jr richard: single positional indexer is out
-of-bounds
Error fetching key_mlbam for silvio martinez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for willie hernandez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for john henry johnson: single positional indexe
r is out-of-bounds
Error fetching key_mlbam for alejandro pena: single positional indexer is
out-of-bounds
Error fetching key_mlbam for joaquin andujar: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for oil can boyd: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for pascual perez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jose deleon: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for jose mesa: single positional indexer is out-
of-bounds
Error fetching key_mlbam for jose de jesus: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jose guzman: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for jose bautista: single positional indexer is
out-of-bounds
Error fetching key_mlbam for angel miranda: single positional indexer is
out-of-bounds
Error fetching key_mlbam for dennis martinez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for francisco cordova: single positional indexer
is out-of-bounds
Error fetching key_mlbam for juan guzman: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for carlos perez: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for hipolito pichardo: single positional indexer
is out-of-bounds
Error fetching key_mlbam for jose rosado: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for jose silva: single positional indexer is out
-of-bounds
Error fetching key_mlbam for osvaldo fernandez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for ramon martinez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jose mercedes: single positional indexer is
out-of-bounds
Error fetching key_mlbam for vladimir nunez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jose rijo: single positional indexer is out-
of-bounds
Error fetching key_mlbam for ruben quevedo: single positional indexer is
out-of-bounds
```

```
Error fetching key_mlbam for salomon torres: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jesus sanchez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for wilson alvarez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for joaquin benoit: single positional indexer is
out-of-bounds
Error fetching key_mlbam for geremi gonzalez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for sunwoo kim: single positional indexer is out
-of-bounds
Error fetching key_mlbam for jose lima: single positional indexer is out-
of-bounds
Error fetching key_mlbam for gustavo chacin: single positional indexer is
out-of-bounds
Error fetching key_mlbam for orlando hernandez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for byunghyun kim: single positional indexer is
out-of-bounds
Error fetching key_mlbam for victor santos: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jae weong seo: single positional indexer is
out-of-bounds
Error fetching key_mlbam for julian tavarez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for victor zambrano: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for shawn chacon: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for runelvys hernandez: single positional indexe
r is out-of-bounds
Error fetching key_mlbam for hungchih kuo: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for odalis perez: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for jose contreras: single positional indexer is
out-of-bounds
Error fetching key_mlbam for chan ho park: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for horacio ramirez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for oliver perez: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for ryan rowlandsmith: single positional indexer
is out-of-bounds
Error fetching key_mlbam for dj carrasco: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for livan hernandez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for rodrigo lopez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for joel pineiro: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for jojo reyes: single positional indexer is out
-of-bounds
Error fetching key_mlbam for javier vazquez: single positional indexer is
```

```
out-of-bounds
Error fetching key_mlbam for freddy garcia: single positional indexer is
out-of-bounds
Error fetching key_mlbam for ramon ortiz: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for jonathan sanchez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for chienming wang: single positional indexer is
out-of-bounds
Error fetching key_mlbam for erik bedard: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for aj burnett: single positional indexer is out
-of-bounds
Error fetching key_mlbam for felix doubront: single positional indexer is
out-of-bounds
Error fetching key_mlbam for wandy rodriguez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for cj wilson: single positional indexer is out-
of-bounds
Error fetching key_mlbam for jose fernandez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for roberto hernandez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for jon niese: single positional indexer is out-
of-bounds
Error fetching key_mlbam for vidal nuno iii: single positional indexer is
out-of-bounds
Error fetching key_mlbam for alfredo simon: single positional indexer is
out-of-bounds
Error fetching key_mlbam for henderson alvarez iii: single positional ind
exer is out-of-bounds
Error fetching key_mlbam for ra dickey: single positional indexer is out-
of-bounds
Error fetching key_mlbam for aj griffin: single positional indexer is out
-of-bounds
Error fetching key_mlbam for ubaldo jimenez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for weiyin chen: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for bartolo colon: single positional indexer is
out-of-bounds
Error fetching key_mlbam for roenis elias: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for jaime garcia: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for miguel gonzalez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for felix hernandez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for hector noesi: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for edinson volquez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for ivan nova: single positional indexer is out-
of-bounds
Error fetching key_mlbam for jose alvarez: single positional indexer is o
ut-of-bounds
```

```
Error fetching key_mlbam for jhoulys chacin: single positional indexer is
out-of-bounds
Error fetching key_mlbam for ja happ: single positional indexer is out-of
-bounds
Error fetching key_mlbam for jorge lopez: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for carlos martinez: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for hector santiago: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for reynaldo lopez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for lance mccullers jr: single positional indexe
r is out-of-bounds
Error fetching key_mlbam for anibal sanchez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for sandy alcantara: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for jaime barria: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for jose berrios: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for matthew boyd: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for nestor cortes: single positional indexer is
out-of-bounds
```

```
Error fetching key_mlbam for domingo german: single positional indexer is
out-of-bounds
Error fetching key_mlbam for carlos hernandez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for pablo lopez: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for jesus luzardo: single positional indexer is
out-of-bounds
Error fetching key_mlbam for german marquez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for nick martinez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for martin perez: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for jose quintana: single positional indexer is
out-of-bounds
Error fetching key_mlbam for carlos rodon: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for eduardo rodriguez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for hyun jin ryu: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for jose suarez: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for ranger suarez: single positional indexer is
out-of-bounds
Error fetching key_mlbam for julio teheran: single positional indexer is
out-of-bounds
Error fetching key_mlbam for jose urena: single positional indexer is out
-of-bounds
Error fetching key_mlbam for julio urias: single positional indexer is ou
t-of-bounds
Error fetching key_mlbam for jose urquidy: single positional indexer is o
ut-of-bounds
```

Out[5]:

	Name	Age	Year	Throws	IP	G	GS	CG	SHO	sDR	...	Career End	Inactive Years	Sur
240	steve blass	31	1973	1	88.2	23	18	1	0	1	...	1973	[]	
285	eddie fisher	36	1973	1	117.2	32	16	2	0	9	...	1973	[]	
385	milt pappas	34	1973	1	162.0	30	29	1	1	8	...	1973	[]	
463	steve arlin	28	1974	1	107.2	27	22	2	0	9	...	1974	[]	
589	ernie mcanally	27	1974	1	128.2	25	21	5	2	6	...	1974	[]	

5 rows × 21 columns

In [6]:

```
names_list = ['blue moon odom', 'jr richard', 'silvio martinez', 'will
'john henry johnson', 'alejandro pena', 'joaquin andujar', 'oil can bo
'pascual perez', 'jose deleon', 'jose mesa', 'jose de jesus',
'jose guzman', 'jose bautista', 'angel miranda', 'dennis martinez',
'francisco cordova', 'juan guzman', 'carlos perez', 'hipolito pichardo
'jose rosado', 'jose silva', 'osvaldo fernandez', 'ramon martinez',
'jose mercedes', 'vladimir nunez', 'jose rijo', 'ruben quevedo',
'salomon torres', 'jesus sanchez', 'wilson alvarez', 'joaquin benoit',
'geremi gonzalez', 'sunwoo kim', 'jose lima', 'gustavo chacin',
'orlando hernandez', 'byunghyun kim', 'victor santos', 'jae weong seo'
'julian tavarez', 'victor zambrano', 'shawn chacon', 'runelvys hernand
'hungchih kuo', 'odalis perez', 'jose contreras', 'chan ho park',
'horacio ramirez', 'oliver perez', 'ryan rowlandsmith', 'dj carrasco',
'livan hernandez', 'rodrigo lopez', 'joel pineiro', 'jojo reyes', 'jav
'freddy garcia', 'ramon ortiz', 'jonathan sanchez', 'chienming wang',
'erik bedard', 'aj burnett', 'felix doubront', 'wandy rodriguez',
'cj wilson', 'jose fernandez', 'roberto hernandez', 'jon niese',
'vidal nuno iii', 'alfredo simon', 'henderson alvarez iii', 'ra dickey
'aj griffin', 'ubaldo jimenez', 'weiyin chen', 'bartolo colon', 'roeni
'jaime garcia', 'miguel gonzalez', 'felix hernandez', 'hector noesi',
'edinson volquez', 'ivan nova', 'jose alvarez', 'jhoulys chacin', 'ja
'jorge lopez', 'carlos martinez', 'hector santiago', 'reynaldo lopez',
'lance mccullers jr', 'anibal sanchez', 'sandy alcantara', 'jaime barr
'matthew boyd', 'nestor cortes', 'domingo german', 'carlos hernandez',
'jesus luzardo', 'german marquez', 'nick martinez', 'martin perez', 'j
'carlos rodon', 'eduardo rodriguez', 'hyun jin ryu', 'jose suarez', 'r
'julio teheran', 'jose urena', 'julio urias', 'jose urquidy']

# Target name you are trying to match
all_player_names = ['Blue Moon Odom', 'J. R. Richard', 'Silvio Martíne
                    'Willie Hernández', 'John Henry Johnson', 'Alejand
                    'Joaquín Andújar', 'Oil Can Boyd', 'Pascual Pérez'
                    'José Mesa', 'José DeJesús', 'José Guzmán', 'José
                    'Ángel Miranda', 'Dennis Martínez', 'Francisco Cór
                    'Juan Guzmán', 'Carlos Pérez', 'Hipólito Pichardo'
                    'José Silva', 'Osvaldo Fernández', 'Ramón Martínez
                    'Vladimir Núñez', 'José Rijo', 'Rubén Quevedo', 'S
                    'Jesús Sánchez', 'Wilson Álvarez', 'Joaquín Benoit
                    'Sun-woo Kim', 'José Lima', 'Gustavo Chacín', 'Orl
                    'Byung-hyun Kim', 'Víctor Santos', 'Jae Weong Seo'
                    'Julián Tavárez', 'Víctor Zambrano', 'Shawn Chacón
                    'Runelvys Hernández', 'Hong-Chih Kuo', 'Odalis Pér
                   'José Contreras', 'Chan Ho Park', 'Horacio Ramírez'
                    'Ryan Rowland-Smith', 'D. J. Carrasco', 'Liván Her
                    'Rodrigo López', 'Joel Piñeiro', 'Jo-Jo Reyes', 'J
                    'Freddy García', 'Ramón Ortiz', 'Jonathan Sánchez'
                    'Érik Bédard', 'A. J. Burnett', 'Félix Doubront',
                    'C. J. Wilson', 'José Fernández', 'Roberto Hernánd
                    'Jon Niese', 'Vidal Nuño', 'Alfredo Simón', 'Hende
                    'R. A. Dickey', 'A. J. Griffin', 'Ubaldo Jiménez',
                    'Bartolo Colón', 'Roenis Elías', 'Jaime García', '
                    'Félix Hernández', 'Héctor Noesí', 'Edinson Vólque
                   'José Álvarez', 'Jhoulys Chacín', 'J.A. Happ', 'Jor
                    'Carlos Martínez', 'Hector Santiago', 'Reynaldo Ló
                    'Lance McCullers Jr.', 'Aníbal Sánchez', 'Sandy Al
                    'José Berríos', 'Matthew Boyd', 'Nestor Cortes', '
                    'Carlos Hernández', 'Pablo López', 'Jesús Luzardo'
```

```
                  'Germán Márquez', 'Martín Pérez', 'Carlos Rodón', '
                   'Hong-Chih Kuo', 'Edinson Vólquez', 'Jose Alvarez'

# Function to find the best match for each name in names_list
def find_best_matches(names_list, all_player_names):
    best_matches = {}
    for name in names_list:
        match = process.extractOne(name, all_player_names)
        best_matches[name] = match
    return best_matches

# Get best matches
best_matches = find_best_matches(names_list, all_player_names)

# Print best matches
for name, match in best_matches.items():
    print(f"Original: {name}, Best Match: {match[0]}, Score: {match[1]
```

```
Original: blue moon odom, Best Match: Blue Moon Odom, Score: 100
Original: jr richard, Best Match: J. R. Richard, Score: 90
Original: silvio martinez, Best Match: Silvio Martínez, Score: 97
Original: willie hernandez, Best Match: Willie Hernández, Score: 97
Original: john henry johnson, Best Match: John Henry Johnson, Score: 100
Original: alejandro pena, Best Match: Alejandro Peña, Score: 96
Original: joaquin andujar, Best Match: Joaquín Andújar, Score: 93
Original: oil can boyd, Best Match: Oil Can Boyd, Score: 100
Original: pascual perez, Best Match: Pascual Pérez, Score: 96
Original: jose deleon, Best Match: José DeLeón, Score: 90
Original: jose mesa, Best Match: José Mesa, Score: 94
Original: jose de jesus, Best Match: José DeJesús, Score: 87
Original: jose guzman, Best Match: José Guzmán, Score: 90
Original: jose bautista, Best Match: José Bautista, Score: 96
Original: angel miranda, Best Match: Ángel Miranda, Score: 96
Original: dennis martinez, Best Match: Dennis Martínez, Score: 97
Original: francisco cordova, Best Match: Francisco Córdova, Score: 97
Original: juan guzman, Best Match: Juan Guzmán, Score: 95
Original: carlos perez, Best Match: Carlos Pérez, Score: 96
Original: hipolito pichardo, Best Match: Hipólito Pichardo, Score: 97
Original: jose rosado, Best Match: José Rosado, Score: 95
Original: jose silva, Best Match: José Silva, Score: 95
Original: osvaldo fernandez, Best Match: Osvaldo Fernández, Score: 97
Original: ramon martinez, Best Match: Ramón Martínez, Score: 92
Original: jose mercedes, Best Match: José Mercedes, Score: 96
Original: vladimir nunez, Best Match: Vladimir Núñez, Score: 92
Original: jose rijo, Best Match: José Rijo, Score: 94
Original: ruben quevedo, Best Match: Rubén Quevedo, Score: 96
Original: salomon torres, Best Match: Salomón Torres, Score: 96
Original: jesus sanchez, Best Match: Jesús Sánchez, Score: 92
Original: wilson alvarez, Best Match: Wilson Álvarez, Score: 96
Original: joaquin benoit, Best Match: Joaquín Benoit, Score: 96
Original: geremi gonzalez, Best Match: Geremi González, Score: 97
Original: sunwoo kim, Best Match: Sun-woo Kim, Score: 95
Original: jose lima, Best Match: José Lima, Score: 94
Original: gustavo chacin, Best Match: Gustavo Chacín, Score: 96
Original: orlando hernandez, Best Match: Orlando Hernández, Score: 97
Original: byunghyun kim, Best Match: Byung-hyun Kim, Score: 96
Original: victor santos, Best Match: Víctor Santos, Score: 96
Original: jae weong seo, Best Match: Jae Weong Seo, Score: 100
Original: julian tavarez, Best Match: Julián Tavárez, Score: 92
Original: victor zambrano, Best Match: Víctor Zambrano, Score: 97
Original: shawn chacon, Best Match: Shawn Chacón, Score: 96
Original: runelvys hernandez, Best Match: Runelvys Hernández, Score: 97
Original: hungchih kuo, Best Match: Hong-Chih Kuo, Score: 88
Original: odalis perez, Best Match: Odalis Pérez, Score: 96
Original: jose contreras, Best Match: José Contreras, Score: 96
Original: chan ho park, Best Match: Chan Ho Park, Score: 100
Original: horacio ramirez, Best Match: Horacio Ramírez, Score: 97
Original: oliver perez, Best Match: Óliver Pérez, Score: 91
Original: ryan rowlandsmith, Best Match: Ryan Rowland-Smith, Score: 97
Original: dj carrasco, Best Match: D. J. Carrasco, Score: 91
Original: livan hernandez, Best Match: Liván Hernández, Score: 93
Original: rodrigo lopez, Best Match: Rodrigo López, Score: 96
Original: joel pineiro, Best Match: Joel Piñeiro, Score: 96
Original: jojo reyes, Best Match: Jo-Jo Reyes, Score: 95
Original: javier vazquez, Best Match: Javier Vázquez, Score: 96
```

```
Original: freddy garcia, Best Match: Freddy García, Score: 96
Original: ramon ortiz, Best Match: Ramón Ortiz, Score: 95
Original: jonathan sanchez, Best Match: Jonathan Sánchez, Score: 97
Original: chienming wang, Best Match: Chien-Ming Wang, Score: 97
Original: erik bedard, Best Match: Érik Bédard, Score: 90
Original: aj burnett, Best Match: A. J. Burnett, Score: 90
Original: felix doubront, Best Match: Félix Doubront, Score: 96
Original: wandy rodriguez, Best Match: Wandy Rodríguez, Score: 97
Original: cj wilson, Best Match: C. J. Wilson, Score: 90
Original: jose fernandez, Best Match: José Fernández, Score: 92
Original: roberto hernandez, Best Match: Roberto Hernández, Score: 97
Original: jon niese, Best Match: Jon Niese, Score: 100
Original: vidal nuno iii, Best Match: Vidal Nuño, Score: 86
Original: alfredo simon, Best Match: Alfredo Simón, Score: 96
Original: henderson alvarez iii, Best Match: Henderson Álvarez, Score: 86
Original: ra dickey, Best Match: R. A. Dickey, Score: 86
Original: aj griffin, Best Match: A. J. Griffin, Score: 90
Original: ubaldo jimenez, Best Match: Ubaldo Jiménez, Score: 96
Original: weiyin chen, Best Match: Wei-Yin Chen, Score: 96
Original: bartolo colon, Best Match: Bartolo Colón, Score: 96
Original: roenis elias, Best Match: Roenis Elías, Score: 96
Original: jaime garcia, Best Match: Jaime García, Score: 96
Original: miguel gonzalez, Best Match: Miguel González, Score: 97
Original: felix hernandez, Best Match: Félix Hernández, Score: 93
Original: hector noesi, Best Match: Héctor Noesí, Score: 91
Original: edinson volquez, Best Match: Edinson Vólquez, Score: 97
Original: ivan nova, Best Match: Iván Nova, Score: 94
Original: jose alvarez, Best Match: Jose Alvarez, Score: 100
Original: jhoulys chacin, Best Match: Jhoulys Chacín, Score: 96
Original: ja happ, Best Match: J.A. Happ, Score: 88
Original: jorge lopez, Best Match: Jorge López, Score: 95
Original: carlos martinez, Best Match: Carlos Martínez, Score: 97
Original: hector santiago, Best Match: Hector Santiago, Score: 100
Original: reynaldo lopez, Best Match: Reynaldo López, Score: 96
Original: lance mccullers jr, Best Match: Lance McCullers Jr., Score: 100
Original: anibal sanchez, Best Match: Aníbal Sánchez, Score: 92
Original: sandy alcantara, Best Match: Sandy Alcántara, Score: 97
Original: jaime barria, Best Match: Jaime Barría, Score: 96
Original: jose berrios, Best Match: José Berríos, Score: 91
Original: matthew boyd, Best Match: Matthew Boyd, Score: 100
Original: nestor cortes, Best Match: Nestor Cortes, Score: 100
Original: domingo german, Best Match: Domingo Germán, Score: 96
Original: carlos hernandez, Best Match: Carlos Hernández, Score: 97
Original: pablo lopez, Best Match: Pablo López, Score: 95
Original: jesus luzardo, Best Match: Jesús Luzardo, Score: 96
Original: german marquez, Best Match: Germán Márquez, Score: 92
Original: nick martinez, Best Match: Dennis Martínez, Score: 74
Original: martin perez, Best Match: Martín Pérez, Score: 91
Original: jose quintana, Best Match: José Bautista, Score: 64
Original: carlos rodon, Best Match: Carlos Rodón, Score: 96
Original: eduardo rodriguez, Best Match: Wandy Rodríguez, Score: 71
Original: hyun jin ryu, Best Match: Byung-hyun Kim, Score: 51
Original: jose suarez, Best Match: Jose Alvarez, Score: 78
Original: ranger suarez, Best Match: Ramón Martínez, Score: 64
Original: julio teheran, Best Match: Julián Tavárez, Score: 56
Original: jose urena, Best Match: José Mesa, Score: 67
```

```
Original: julio urias, Best Match: José Urquidy, Score: 55
Original: jose urquidy, Best Match: José Urquidy, Score: 96
```

In [7]:

```
updated_names = {
    "blue moon odom": "Blue Moon Odom",
    "jr richard": "J. R. Richard",
    "silvio martinez": "Silvio Martínez",
    "willie hernandez": "Willie Hernández",
    "john henry johnson": "John Henry Johnson",
    "alejandro pena": "Alejandro Peña",
    "joaquin andujar": "Joaquín Andújar",
    "oil can boyd": "Oil Can Boyd",
    "pascual perez": "Pascual Pérez",
    "jose deleon": "José DeLeón",
    "jose mesa": "José Mesa",
    "jose de jesus": "José DeJesús",
    "jose guzman": "José Guzmán",
    "jose bautista": "José Bautista",
    "angel miranda": "Ángel Miranda",
    "dennis martinez": "Dennis Martínez",
    "francisco cordova": "Francisco Córdova",
    "juan guzman": "Juan Guzmán",
    "carlos perez": "Carlos Pérez",
    "hipolito pichardo": "Hipólito Pichardo",
    "jose rosado": "José Rosado",
    "jose silva": "José Silva",
    "osvaldo fernandez": "Osvaldo Fernández",
    "ramon martinez": "Ramón Martínez",
    "jose mercedes": "José Mercedes",
    "vladimir nunez": "Vladimir Núñez",
    "jose rijo": "José Rijo",
    "ruben quevedo": "Rubén Quevedo",
    "salomon torres": "Salomón Torres",
    "jesus sanchez": "Jesús Sánchez",
    "wilson alvarez": "Wilson Álvarez",
    "joaquin benoit": "Joaquín Benoit",
    "geremi gonzalez": "Geremi González",
    "sunwoo kim": "Sun-woo Kim",
    "jose lima": "José Lima",
    "gustavo chacin": "Gustavo Chacín",
    "orlando hernandez": "Orlando Hernández",
    "byunghyun kim": "Byung-hyun Kim",
    "victor santos": "Víctor Santos",
    "jae weong seo": "Jae Weong Seo",
    "julian tavarez": "Julián Tavárez",
    "victor zambrano": "Víctor Zambrano",
    "shawn chacon": "Shawn Chacón",
    "runelvys hernandez": "Runelvys Hernández",
    "hungchih kuo": "Hong-Chih Kuo",
    "odalis perez": "Odalis Pérez",
    "jose contreras": "José Contreras",
    "chan ho park": "Chan Ho Park",
    "horacio ramirez": "Horacio Ramírez",
    "oliver perez": "Óliver Pérez",
    "ryan rowlandsmith": "Ryan Rowland-Smith",
    "dj carrasco": "D. J. Carrasco",
    "livan hernandez": "Liván Hernández",
    "rodrigo lopez": "Rodrigo López",
    "joel pineiro": "Joel Piñeiro",
    "jojo reyes": "Jo-Jo Reyes",
```

```
    "javier vazquez": "Javier Vázquez",
    "freddy garcia": "Freddy García",
    "ramon ortiz": "Ramón Ortiz",
    "jonathan sanchez": "Jonathan Sánchez",
    "chienming wang": "Chien-Ming Wang",
    "erik bedard": "Érik Bédard",
    "aj burnett": "A. J. Burnett",
    "felix doubront": "Félix Doubront",
    "wandy rodriguez": "Wandy Rodríguez",
    "cj wilson": "C. J. Wilson",
    "jose fernandez": "José Fernández",
    "roberto hernandez": "Roberto Hernández",
    "jon niese": "Jon Niese",
    "vidal nuno iii": "Vidal Nuño",
    "alfredo simon": "Alfredo Simón",
    "henderson alvarez iii": "Henderson Álvarez",
    "ra dickey": "R. A. Dickey",
    "aj griffin": "A. J. Griffin",
    "ubaldo jimenez": "Ubaldo Jiménez",
    "weiyin chen": "Wei-Yin Chen",
    "bartolo colon": "Bartolo Colón",
    "roenis elias": "Roenis Elías",
    "jaime garcia": "Jaime García",
    "miguel gonzalez": "Miguel González",
    "felix hernandez": "Félix Hernández",
    "hector noesi": "Héctor Noesí",
    "edinson volquez": "Edinson Vólquez",
    "ivan nova": "Iván Nova",
    "jose alvarez": "Jose Alvarez",
    "jhoulys chacin": "Jhoulys Chacín",
    "ja happ": "J.A. Happ",
    "jorge lopez": "Jorge López",
    "carlos martinez": "Carlos Martínez",
    "hector santiago": "Hector Santiago",
    "reynaldo lopez": "Reynaldo López",
    "lance mccullers jr": "Lance McCullers Jr.",
    "anibal sanchez": "Aníbal Sánchez",
    "sandy alcantara": "Sandy Alcántara",
    "jaime barria": "Jaime Barría",
    "jose berrios": "José Berríos",
    "matthew boyd": "Matthew Boyd",
    "nestor cortes": "Nestor Cortes",
    "domingo german": "Domingo Germán",
    "carlos hernandez": "Carlos Hernández",
    "pablo lopez": "Pablo López",
    "jesus luzardo": "Jesús Luzardo",
    "german marquez": "Germán Márquez",
    "nick martinez": "Dennis Martínez",
    "martin perez": "Martín Pérez",
    "jose quintana": "José Quintana",
    "carlos rodon": "Carlos Rodón",
    "eduardo rodriguez": "Eduardo Rodriguez",
    "hyun jin ryu": "Hyun-jin Ryu",
    "jose suarez": "José Suarez",
    "ranger suarez": "Ranger Suárez",
    "julio teheran": "Julio Teheran",
    "jose urena": "José Ureña",
```

```
    "julio urias": "Julio Urías",
    "jose urquidy": "José Urquidy",
}

pitcher_data_df['Name'] = pitcher_data_df['Name'].map(updated_names).f
pitcher_data_df
```

Out[7]:

	Name	Age	Year	Throws	IP	G	GS	CG	SHO	sDR	...	Career End	Inactive Years
240	steve blass	31	1973	1	88.2	23	18	1	0	1	...	1973	[]
285	eddie fisher	36	1973	1	117.2	32	16	2	0	9	...	1973	[]
385	milt pappas	34	1973	1	162.0	30	29	1	1	8	...	1973	[]
463	steve arlin	28	1974	1	107.2	27	22	2	0	9	...	1974	[]
589	ernie mcanally	27	1974	1	128.2	25	21	5	2	6	...	1974	[]
...	...	...	...	...	...	...	...	...	...	...	...	...	...
15334	zack wheeler	33	2023	1	192.0	32	32	0	0	0	...	2023	[2015, 2016]
15339	trevor williams	31	2023	1	144.1	30	30	0	0	0	...	2023	[]
15346	alex wood	32	2023	0	97.2	29	12	0	0	1	...	2023	[]
15348	brandon woodruff	30	2023	1	67.0	11	11	1	1	0	...	2023	[]
15350	ryan yarbrough	31	2023	0	89.2	25	9	0	0	0	...	2023	[]

1244 rows × 21 columns

In [8]:

```
# Function to fetch key_mlbam
def fetch_key_mlbam(row):
    try:
        # Splitting the name into first and last name
        first_name, last_name = row['Name'].split(' ')[0], ' '.join(ro
        # Fetching player ID
        player_id_df = playerid_lookup(last_name, first_name)
        # Assuming the first result is the correct one, adjust as nece
        key_mlbam = player_id_df.iloc[0]['key_mlbam']
        return key_mlbam
    except Exception as e:
        print(f"Error fetching key_mlbam for {row['Name']}: {e}")
        return pd.NA

# Apply the function to each row and create a new column 'key_mlbam'
pitcher_data_df['key_mlbam'] = pitcher_data_df.apply(fetch_key_mlbam,

# Display the updated DataFrame
pitcher_data_df.head()
```

```
Error fetching key_mlbam for Blue Moon Odom: single positional indexer is
out-of-bounds
Error fetching key_mlbam for J. R. Richard: single positional indexer is
out-of-bounds
Error fetching key_mlbam for John Henry Johnson: single positional indexe
r is out-of-bounds
Error fetching key_mlbam for Oil Can Boyd: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for José DeJesús: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for Jae Weong Seo: single positional indexer is
out-of-bounds
Error fetching key_mlbam for Hong-Chih Kuo: single positional indexer is
out-of-bounds
Error fetching key_mlbam for Chan Ho Park: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for D. J. Carrasco: single positional indexer is
out-of-bounds
Error fetching key_mlbam for A. J. Burnett: single positional indexer is
out-of-bounds
Error fetching key_mlbam for C. J. Wilson: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for Jon Niese: single positional indexer is out-
of-bounds
Error fetching key_mlbam for R. A. Dickey: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for A. J. Griffin: single positional indexer is
out-of-bounds
Error fetching key_mlbam for Jose Alvarez: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for J.A. Happ: single positional indexer is out-
of-bounds
Error fetching key_mlbam for Hector Santiago: single positional indexer i
s out-of-bounds
Error fetching key_mlbam for Lance McCullers Jr.: single positional index
er is out-of-bounds
Error fetching key_mlbam for Matthew Boyd: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for Nestor Cortes: single positional indexer is
out-of-bounds
Error fetching key_mlbam for Eduardo Rodriguez: single positional indexer
is out-of-bounds
Error fetching key_mlbam for Hyun-jin Ryu: single positional indexer is o
ut-of-bounds
Error fetching key_mlbam for Julio Teheran: single positional indexer is
out-of-bounds
```

Out[8]:

	Name	Age	Year	Throws	IP	G	GS	CG	SHO	sDR	...	Career End	Inactive Years	Sur
240	steve blass	31	1973	1	88.2	23	18	1	0	1	...	1973	[]	
285	eddie fisher	36	1973	1	117.2	32	16	2	0	9	...	1973	[]	
385	milt pappas	34	1973	1	162.0	30	29	1	1	8	...	1973	[]	
463	steve arlin	28	1974	1	107.2	27	22	2	0	9	...	1974	[]	
589	ernie mcanally	27	1974	1	128.2	25	21	5	2	6	...	1974	[]	

5 rows × 21 columns

From here, looked up player names, referenced back to baseball-reference.com to confirm the player was correct, and manually entered player keys.

In [9]:
```
1 playerid_lookup('teheran', fuzzy=True)
```

```
No identically matched names found! Returning the 5 most similar names.
```

Out[9]:

	name_last	name_first	key_mlbam	key_retro	key_bbref	key_fangraphs	mlb_played_first
0	sherman	joe	122138	sherj101	shermjo01	1011880	1915.0
1	torian	lonnie	-1		torialo01	-1	1920.0
2	teherán	julio	527054	tehej001	teherju01	6797	2011.0
3	stephens	john	-1		stephfr01	-1	1921.0
4	stephens	john	407856	stepj001	stephjo03	1569	2002.0
5	stephens	john	-1		stephfr01	-1	1921.0
6	stephens	john	407856	stepj001	stephjo03	1569	2002.0

In [10]:
```
1 teheran_key = '527054'
2 pitcher_data_df.loc[pitcher_data_df['Name'] == "Julio Teheran", 'key_m
```

In [11]:
```
1 odom_key = '119935'
2 pitcher_data_df.loc[pitcher_data_df['Name'] == "Blue Moon Odom", 'key_
```

In [12]:
```
1 richard_key = '121145'
2 pitcher_data_df.loc[pitcher_data_df['Name'] == "J. R. Richard", 'key_m
```

In [13]:
```
1 oil_key = '111312'
2 pitcher_data_df.loc[pitcher_data_df['Name'] == "Oil Can Boyd", 'key_ml
```

In [14]:
```
seo_key = '150242'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Jae Weong Seo", 'key_m
```

In [15]:
```
kuo_key = '425539'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Hong-Chih Kuo", 'key_m
```

In [16]:
```
park_key = '120221'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Chan Ho Park", 'key_ml
```

In [17]:
```
djcarrasco_key = '425647'
pitcher_data_df.loc[pitcher_data_df['Name'] == "D. J. Carrasco", 'key_
```

In [18]:
```
burnett_key = '150359'
pitcher_data_df.loc[pitcher_data_df['Name'] == "A. J. Burnett", 'key_m
```

In [19]:
```
wilson_key = '450351'
pitcher_data_df.loc[pitcher_data_df['Name'] == "C. J. Wilson", 'key_ml
```

In [20]:
```
niese_key = '477003'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Jon Niese", 'key_mlbam
```

In [21]:
```
dickey_key = '285079'
pitcher_data_df.loc[pitcher_data_df['Name'] == "R. A. Dickey", 'key_ml
```

In [22]:
```
griffin_key = '456167'
pitcher_data_df.loc[pitcher_data_df['Name'] == "A. J. Griffin", 'key_m
```

In [23]:
```
alvarez_key = '571439'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Jose Alvarez", 'key_ml
```

In [24]:
```
happ_key = '457918'
pitcher_data_df.loc[pitcher_data_df['Name'] == "J.A. Happ", 'key_mlbam
```

In [25]:
```
santiago_key = '502327'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Hector Santiago", 'key
```

In [26]:
```
mccullers_key = '621121'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Lance McCullers Jr.",
```

In [27]:
```
mattboyd_key = '571510'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Matthew Boyd", 'key_ml
```

Nestor Cortes, José DeJesús

Can't find. Drop.

In [28]:
```
pitcher_data_df = pitcher_data_df[pitcher_data_df['Name'] != 'Nestor C
```

In [29]:
```
pitcher_data_df = pitcher_data_df[pitcher_data_df['Name'] != 'José DeJ
```

In [30]:
```
pitcher_data_df['key_mlbam'].isna().value_counts()
```

Out[30]:
```
key_mlbam
False    1239
True        3
Name: count, dtype: int64
```

(Within the saved CSV, here are no True value counts. All ID's are accounted for.)

In [31]:
```
pitcher_data_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1242 entries, 240 to 15350
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Name             1242 non-null   object 
 1   Age              1242 non-null   int64  
 2   Year             1242 non-null   int64  
 3   Throws           1242 non-null   int64  
 4   IP               1242 non-null   float64
 5   G                1242 non-null   int64  
 6   GS               1242 non-null   int64  
 7   CG               1242 non-null   int64  
 8   SHO              1242 non-null   int64  
 9   sDR              1242 non-null   int64  
 10  Career Start     1242 non-null   int64  
 11  Career End       1242 non-null   int64  
 12  Inactive Years   1242 non-null   object 
 13  Surgery          1242 non-null   float64
 14  TJ Surgery Date  1242 non-null   object 
 15  Surgeon(s)       197 non-null    object 
 16  Country          268 non-null    object 
 17  Level            268 non-null    object 
 18  Total_IP         1242 non-null   float64
 19  TJ Surgery Year  1242 non-null   object 
 20  key_mlbam        1239 non-null   object 
dtypes: float64(3), int64(10), object(8)
memory usage: 213.5+ KB
```

In [32]:
```
"""
pitcher_data_df.to_csv('data/pitcher_key_df.csv')
"""
```

Out[32]:
```
"\npitcher_data_df.to_csv('pitcher_key_df.csv')\n"
```

In [33]:
```
playerid_lookup('ohtani', 'shohei')
```

Out[33]:

	name_last	name_first	key_mlbam	key_retro	key_bbref	key_fangraphs	mlb_played_first
0	ohtani	shohei	660271	ohtas001	ohtansh01	19755	2018.0

In [34]:
```
ohtani_stats = statcast_pitcher('2018-01-01', '2023-10-01', 660271)
ohtani_stats
```

```
Gathering Player Data
```

Out[34]:

	pitch_type	game_date	release_speed	release_pos_x	release_pos_z	player_name	b
0	FF	2023-08-23	94.2	-1.98	5.78	Ohtani, Shohei	68
1	ST	2023-08-23	76.1	-2.01	5.74	Ohtani, Shohei	68
2	ST	2023-08-23	77.2	-2.05	5.60	Ohtani, Shohei	68
3	FS	2023-08-23	90.1	-1.94	5.69	Ohtani, Shohei	68
4	FS	2023-08-23	90.0	-1.83	5.76	Ohtani, Shohei	68
...	...	...	...	...	...	...	
7948	NaN	2018-02-24	NaN	NaN	NaN	Ohtani, Shohei	54
7949	NaN	2018-02-24	NaN	NaN	NaN	Ohtani, Shohei	54
7950	NaN	2018-02-24	NaN	NaN	NaN	Ohtani, Shohei	54
7951	NaN	2018-02-24	NaN	NaN	NaN	Ohtani, Shohei	54
7952	NaN	2018-02-24	NaN	NaN	NaN	Ohtani, Shohei	54

7953 rows × 92 columns

In [35]:
```
1 ohtani_stats.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7953 entries, 0 to 7952
Data columns (total 92 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   pitch_type                      7745 non-null   object 
 1   game_date                       7953 non-null   object 
 2   release_speed                   7746 non-null   float64
 3   release_pos_x                   7746 non-null   float64
 4   release_pos_z                   7746 non-null   float64
 5   player_name                     7953 non-null   object 
 6   batter                          7953 non-null   int64  
 7   pitcher                         7953 non-null   int64  
 8   events                          2063 non-null   object 
 9   description                     7953 non-null   object 
 10  spin_dir                        0 non-null      float64
 11  spin_rate_deprecated            0 non-null      float64
 12  break_angle_deprecated          0 non-null      float64
 13  break_length_deprecated         0 non-null      float64
 14  zone                            7746 non-null   float64
 15  des                             7953 non-null   object 
 16  game_type                       7953 non-null   object 
 17  stand                           7953 non-null   object 
 18  p_throws                        7953 non-null   object 
 19  home_team                       7953 non-null   object 
 20  away_team                       7953 non-null   object 
 21  type                            7953 non-null   object 
 22  hit_location                    1779 non-null   float64
 23  bb_type                         1203 non-null   object 
 24  balls                           7953 non-null   int64  
 25  strikes                         7953 non-null   int64  
 26  game_year                       7953 non-null   int64  
 27  pfx_x                           7746 non-null   float64
 28  pfx_z                           7746 non-null   float64
 29  plate_x                         7746 non-null   float64
 30  plate_z                         7746 non-null   float64
 31  on_3b                           522 non-null    float64
 32  on_2b                           1283 non-null   float64
 33  on_1b                           2263 non-null   float64
 34  outs_when_up                    7953 non-null   int64  
 35  inning                          7953 non-null   int64  
 36  inning_topbot                   7953 non-null   object 
 37  hc_x                            1191 non-null   float64
 38  hc_y                            1191 non-null   float64
 39  tfs_deprecated                  0 non-null      float64
 40  tfs_zulu_deprecated             0 non-null      float64
 41  fielder_2                       7953 non-null   int64  
 42  umpire                          0 non-null      float64
 43  sv_id                           0 non-null      float64
 44  vx0                             7746 non-null   float64
 45  vy0                             7746 non-null   float64
 46  vz0                             7746 non-null   float64
 47  ax                              7746 non-null   float64
 48  ay                              7746 non-null   float64
 49  az                              7746 non-null   float64
 50  sz_top                          7746 non-null   float64
 51  sz_bot                          7746 non-null   float64
```

```
 52  hit_distance_sc                  2307 non-null   float64
 53  launch_speed                     2299 non-null   float64
 54  launch_angle                     2305 non-null   float64
 55  effective_speed                  7744 non-null   float64
 56  release_spin_rate                7708 non-null   float64
 57  release_extension                7745 non-null   float64
 58  game_pk                          7953 non-null   int64
 59  pitcher.1                        7953 non-null   int64
 60  fielder_2.1                      7953 non-null   int64
 61  fielder_3                        7953 non-null   int64
 62  fielder_4                        7953 non-null   int64
 63  fielder_5                        7953 non-null   int64
 64  fielder_6                        7953 non-null   int64
 65  fielder_7                        7953 non-null   int64
 66  fielder_8                        7953 non-null   int64
 67  fielder_9                        7953 non-null   int64
 68  release_pos_y                    7746 non-null   float64
 69  estimated_ba_using_speedangle    1150 non-null   float64
 70  estimated_woba_using_speedangle  1150 non-null   float64
 71  woba_value                       2063 non-null   float64
 72  woba_denom                       1950 non-null   float64
 73  babip_value                      2063 non-null   float64
 74  iso_value                        2063 non-null   float64
 75  launch_speed_angle               1150 non-null   float64
 76  at_bat_number                    7953 non-null   int64
 77  pitch_number                     7953 non-null   int64
 78  pitch_name                       7745 non-null   object
 79  home_score                       7953 non-null   int64
 80  away_score                       7953 non-null   int64
 81  bat_score                        7953 non-null   int64
 82  fld_score                        7953 non-null   int64
 83  post_away_score                  7953 non-null   int64
 84  post_home_score                  7953 non-null   int64
 85  post_bat_score                   7953 non-null   int64
 86  post_fld_score                   7953 non-null   int64
 87  if_fielding_alignment            7735 non-null   object
 88  of_fielding_alignment            7735 non-null   object
 89  spin_axis                        7715 non-null   float64
 90  delta_home_win_exp               7953 non-null   float64
 91  delta_run_exp                    7683 non-null   float64
dtypes: float64(47), int64(28), object(17)
memory usage: 5.6+ MB
```

In [36]:
```
1 ohtani_stats['game_date'].value_counts()
```

Out[36]:
```
game_date
2021-09-03    117
2021-09-26    112
2023-07-27    111
2023-04-05    111
2022-09-03    111
              ...
2021-03-05     26
2021-03-13     26
2022-03-21     22
2018-02-24     21
2023-02-28     18
Name: count, Length: 95, dtype: int64
```

In [37]:
```
1 ohtani_stats['game_date'].unique
```

Out[37]:
```
<bound method Series.unique of 0       2023-08-23
1       2023-08-23
2       2023-08-23
3       2023-08-23
4       2023-08-23
           ...
7948    2018-02-24
7949    2018-02-24
7950    2018-02-24
7951    2018-02-24
7952    2018-02-24
Name: game_date, Length: 7953, dtype: object>
```

In [38]:
```
ohtani_start = ohtani_stats[ohtani_stats['game_date'] == '2021-09-03']
ohtani_start
```

Out[38]:

	pitch_type	game_date	release_speed	release_pos_x	release_pos_z	player_name	b
5093	FF	2021-09-03	99.4	-2.14	5.94	Ohtani, Shohei	64
5094	FF	2021-09-03	99.5	-2.12	5.98	Ohtani, Shohei	64
5095	FF	2021-09-03	99.1	-2.15	5.95	Ohtani, Shohei	64
5096	ST	2021-09-03	83.8	-2.35	5.75	Ohtani, Shohei	64
5097	FF	2021-09-03	98.4	-2.06	5.94	Ohtani, Shohei	66
...	...	...	...	...	...	...	
5205	FF	2021-09-03	94.0	-1.94	6.11	Ohtani, Shohei	64
5206	ST	2021-09-03	80.9	-2.04	6.09	Ohtani, Shohei	64
5207	FC	2021-09-03	86.4	-1.99	6.17	Ohtani, Shohei	64
5208	FF	2021-09-03	94.1	-1.74	6.16	Ohtani, Shohei	64
5209	FF	2021-09-03	93.8	-1.76	6.18	Ohtani, Shohei	66

117 rows × 92 columns

In [39]:
```
1 ohtani_start.columns
```

Out[39]:
```
Index(['pitch_type', 'game_date', 'release_speed', 'release_pos_x',
       'release_pos_z', 'player_name', 'batter', 'pitcher', 'events',
       'description', 'spin_dir', 'spin_rate_deprecated',
       'break_angle_deprecated', 'break_length_deprecated', 'zone', 'de
s',
       'game_type', 'stand', 'p_throws', 'home_team', 'away_team', 'typ
e',
       'hit_location', 'bb_type', 'balls', 'strikes', 'game_year', 'pfx_
x',
       'pfx_z', 'plate_x', 'plate_z', 'on_3b', 'on_2b', 'on_1b',
       'outs_when_up', 'inning', 'inning_topbot', 'hc_x', 'hc_y',
       'tfs_deprecated', 'tfs_zulu_deprecated', 'fielder_2', 'umpire', 's
v_id',
       'vx0', 'vy0', 'vz0', 'ax', 'ay', 'az', 'sz_top', 'sz_bot',
       'hit_distance_sc', 'launch_speed', 'launch_angle', 'effective_spee
d',
       'release_spin_rate', 'release_extension', 'game_pk', 'pitcher.1',
       'fielder_2.1', 'fielder_3', 'fielder_4', 'fielder_5', 'fielder_6',
       'fielder_7', 'fielder_8', 'fielder_9', 'release_pos_y',
       'estimated_ba_using_speedangle', 'estimated_woba_using_speedangl
e',
       'woba_value', 'woba_denom', 'babip_value', 'iso_value',
       'launch_speed_angle', 'at_bat_number', 'pitch_number', 'pitch_nam
e',
       'home_score', 'away_score', 'bat_score', 'fld_score', 'post_away_s
core',
       'post_home_score', 'post_bat_score', 'post_fld_score',
       'if_fielding_alignment', 'of_fielding_alignment', 'spin_axis',
       'delta_home_win_exp', 'delta_run_exp'],
      dtype='object')
```

In [40]:
```
1 ohtani_start.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 117 entries, 5093 to 5209
Data columns (total 92 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   pitch_type                      117 non-null    object 
 1   game_date                       117 non-null    object 
 2   release_speed                   117 non-null    float64
 3   release_pos_x                   117 non-null    float64
 4   release_pos_z                   117 non-null    float64
 5   player_name                     117 non-null    object 
 6   batter                          117 non-null    int64  
 7   pitcher                         117 non-null    int64  
 8   events                          29 non-null     object 
 9   description                     117 non-null    object 
 10  spin_dir                        0 non-null      float64
 11  spin_rate_deprecated            0 non-null      float64
 12  break_angle_deprecated          0 non-null      float64
 13  break_length_deprecated         0 non-null      float64
 14  zone                            117 non-null    float64
 15  des                             117 non-null    object 
 16  game_type                       117 non-null    object 
 17  stand                           117 non-null    object 
 18  p_throws                        117 non-null    object 
 19  home_team                       117 non-null    object 
 20  away_team                       117 non-null    object 
 21  type                            117 non-null    object 
 22  hit_location                    26 non-null     float64
 23  bb_type                         19 non-null     object 
 24  balls                           117 non-null    int64  
 25  strikes                         117 non-null    int64  
 26  game_year                       117 non-null    int64  
 27  pfx_x                           117 non-null    float64
 28  pfx_z                           117 non-null    float64
 29  plate_x                         117 non-null    float64
 30  plate_z                         117 non-null    float64
 31  on_3b                           6 non-null      float64
 32  on_2b                           15 non-null     float64
 33  on_1b                           41 non-null     float64
 34  outs_when_up                    117 non-null    int64  
 35  inning                          117 non-null    int64  
 36  inning_topbot                   117 non-null    object 
 37  hc_x                            19 non-null     float64
 38  hc_y                            19 non-null     float64
 39  tfs_deprecated                  0 non-null      float64
 40  tfs_zulu_deprecated             0 non-null      float64
 41  fielder_2                       117 non-null    int64  
 42  umpire                          0 non-null      float64
 43  sv_id                           0 non-null      float64
 44  vx0                             117 non-null    float64
 45  vy0                             117 non-null    float64
 46  vz0                             117 non-null    float64
 47  ax                              117 non-null    float64
 48  ay                              117 non-null    float64
 49  az                              117 non-null    float64
 50  sz_top                          117 non-null    float64
 51  sz_bot                          117 non-null    float64
```

```
 52  hit_distance_sc                  48 non-null     float64
 53  launch_speed                     48 non-null     float64
 54  launch_angle                     48 non-null     float64
 55  effective_speed                  117 non-null    float64
 56  release_spin_rate                117 non-null    float64
 57  release_extension                117 non-null    float64
 58  game_pk                          117 non-null    int64
 59  pitcher.1                        117 non-null    int64
 60  fielder_2.1                      117 non-null    int64
 61  fielder_3                        117 non-null    int64
 62  fielder_4                        117 non-null    int64
 63  fielder_5                        117 non-null    int64
 64  fielder_6                        117 non-null    int64
 65  fielder_7                        117 non-null    int64
 66  fielder_8                        117 non-null    int64
 67  fielder_9                        117 non-null    int64
 68  release_pos_y                    117 non-null    float64
 69  estimated_ba_using_speedangle    19 non-null     float64
 70  estimated_woba_using_speedangle  19 non-null     float64
 71  woba_value                       29 non-null     float64
 72  woba_denom                       29 non-null     float64
 73  babip_value                      29 non-null     float64
 74  iso_value                        29 non-null     float64
 75  launch_speed_angle               19 non-null     float64
 76  at_bat_number                    117 non-null    int64
 77  pitch_number                     117 non-null    int64
 78  pitch_name                       117 non-null    object
 79  home_score                       117 non-null    int64
 80  away_score                       117 non-null    int64
 81  bat_score                        117 non-null    int64
 82  fld_score                        117 non-null    int64
 83  post_away_score                  117 non-null    int64
 84  post_home_score                  117 non-null    int64
 85  post_bat_score                   117 non-null    int64
 86  post_fld_score                   117 non-null    int64
 87  if_fielding_alignment            117 non-null    object
 88  of_fielding_alignment            117 non-null    object
 89  spin_axis                        117 non-null    float64
 90  delta_home_win_exp               117 non-null    float64
 91  delta_run_exp                    117 non-null    float64
dtypes: float64(47), int64(28), object(17)
memory usage: 85.0+ KB
```

This kind of info is great!

Will filter down columns to information only applicable to testing for TJ surgery.

Need to filter years to only include regular season schedule.

This took some manual input.

In [41]:
```
"""
season_dates
    2008: ('2008-03-25', '2008-09-30'),
    2009: ('2009-04-05', '2009-10-06'),
    2010: ('2010-04-04', '2010-10-03'),
    2011: ('2011-03-31', '2011-09-28'),
    2012: ('2012-03-28', '2012-10-03'),
    2013: ('2013-03-31', '2013-09-30'),
    2014: ('2014-03-22', '2014-09-28'),
    2015: ('2015-04-05', '2015-10-04'),
    2016: ('2016-04-03', '2016-10-02'),
    2017: ('2017-04-02', '2017-10-01'),
    2018: ('2018-03-29', '2018-10-01'),
    2019: ('2019-03-20', '2019-09-29'),
    2020: ('2020-07-23', '2020-09-27'),
    2021: ('2021-04-01', '2021-10-03'),
    2022: ('2022-04-07', '2022-10-05'),
    2023: ('2023-03-30', '2023-10-01')
"""
```

Out[41]:
```
"\nseason_dates\n    2008: ('2008-03-25', '2008-09-30'),\n    2009: ('200
9-04-05', '2009-10-06'),\n    2010: ('2010-04-04', '2010-10-03'),\n    20
11: ('2011-03-31', '2011-09-28'),\n    2012: ('2012-03-28', '2012-10-0
3'),\n    2013: ('2013-03-31', '2013-09-30'),\n    2014: ('2014-03-22',
'2014-09-28'),\n    2015: ('2015-04-05', '2015-10-04'),\n    2016: ('2016
-04-03', '2016-10-02'),\n    2017: ('2017-04-02', '2017-10-01'),\n    201
8: ('2018-03-29', '2018-10-01'),\n    2019: ('2019-03-20', '2019-09-2
9'),\n    2020: ('2020-07-23', '2020-09-27'),\n    2021: ('2021-04-01',
'2021-10-03'),\n    2022: ('2022-04-07', '2022-10-05'),\n    2023: ('2023
-03-30', '2023-10-01')\n"
```

In [42]:
```
"""
all_player_stats = []

# Loop through each MLBAM ID in the pitcher_data_df
for key_mlbam in pitcher_data_df['key_mlbam']:
    # Fetch pitching stats from statcast
    player_stats = statcast_pitcher('2018-03-29', '2018-10-01', key_ml
    # Append the fetched stats to the list
    all_player_stats.append(player_stats)

# Concatenate all DataFrames into a single DataFrame
all_2018_stats_df = pd.concat(all_player_stats, ignore_index=True)

all_2018_stats_df.head()
"""
```

Out[42]:
```
"\nall_player_stats = []\n\n# Loop through each MLBAM ID in the pitcher_d
ata_df\nfor key_mlbam in pitcher_data_df['key_mlbam']:\n    # Fetch pitch
ing stats from statcast\n    player_stats = statcast_pitcher('2018-03-2
9', '2018-10-01', key_mlbam)\n    # Append the fetched stats to the list
\n    all_player_stats.append(player_stats)\n\n# Concatenate all DataFram
es into a single DataFrame\nall_2018_stats_df = pd.concat(all_player_stat
s, ignore_index=True)\n\nall_2018_stats_df.head()\n"
```

Start grouping data by game date and pitcher.

Later, further condense data to season and pitcher, while retaining important information about the number of different types of pitches, the averages of those pitches velocity, etc.

Was not done here due to sheer size of each season's file.

In [44]:

```
"""
# Group by 'game_date' and 'pitcher' to calculate the total pitches
total_pitches = all_2018_stats_df.groupby(['game_date', 'pitcher']).si

# Group by 'game_date', 'pitcher', and 'pitch_type' to calculate the s
total_pitches_by_type = all_2018_stats_df.groupby(['game_date', 'pitch

# Calculate averages of the specified metrics for each pitch type, gro
avg_metrics = all_2018_stats_df.groupby(['game_date', 'pitcher', 'pitc
    'release_speed': 'mean',
    'release_pos_x': 'mean',
    'release_pos_z': 'mean',
    'spin_dir': 'mean',
    'vx0': 'mean',
    'vy0': 'mean',
    'vz0': 'mean',
    'ax': 'mean',
    'ay': 'mean',
    'az': 'mean',
    'effective_speed': 'mean',
    'release_spin_rate': 'mean',
    'release_extension': 'mean',
    'release_pos_y': 'mean',
    'spin_axis': 'mean'
}).reset_index()

# Merging total pitches and total pitches by type back
grouped_2018_df = total_pitches.merge(total_pitches_by_type, on=['game
grouped_2018_df = final_df.merge(avg_metrics, on=['game_date', 'pitche
"""
```

Out[44]:

```
"\n# Group by 'game_date' and 'pitcher' to calculate the total pitches\nt
otal_pitches = all_2018_stats_df.groupby(['game_date', 'pitcher']).size
().reset_index(name='total_pitches')\n\n# Group by 'game_date', 'pitche
r', and 'pitch_type' to calculate the sum total of each pitch_type\ntotal
_pitches_by_type = all_2018_stats_df.groupby(['game_date', 'pitcher', 'pi
tch_type']).size().reset_index(name='count_by_pitch_type')\n\n# Calculate
averages of the specified metrics for each pitch type, grouped by game_da
te and pitcher\navg_metrics = all_2018_stats_df.groupby(['game_date', 'pi
tcher', 'pitch_type']).agg({\n    'release_speed': 'mean',\n    'release_
pos_x': 'mean',\n    'release_pos_z': 'mean',\n    'spin_dir': 'mean',\n
'vx0': 'mean',\n    'vy0': 'mean',\n    'vz0': 'mean',\n    'ax': 'mea
n',\n    'ay': 'mean',\n    'az': 'mean',\n    'effective_speed': 'mea
n',\n    'release_spin_rate': 'mean',\n    'release_extension': 'mean',\n
'release_pos_y': 'mean',\n    'spin_axis': 'mean'\n}).reset_index()\n\n#
Merging total pitches and total pitches by type back\ngrouped_2018_df = t
otal_pitches.merge(total_pitches_by_type, on=['game_date', 'pitcher'], ho
w='left')\ngrouped_2018_df = final_df.merge(avg_metrics, on=['game_date',
'pitcher', 'pitch_type'], how='left')\n"
```

This information was saved to individual CSV's in order to save time and processing power.

In [45]:
```
"""
all_2018_stats_df.to_csv('data/all_2018_stats_df.csv')
"""
```

Out[45]:
```
"\nall_2018_stats_df.to_csv('all_2018_stats_df.csv')\n"
```

This process was repeated for each season from 2008 - 2023

In [ ]:
```

```