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('hist_tj.csv', index_col=0)
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()
```

...

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]
```

...

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
```

...

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()
```

...

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]:

```
playerid_lookup('teheran', fuzzy=True)
```

...

In [10]:

```
teheran_key = '527054'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Julio Teheran", 'key_m
```

In [11]:

```
odom_key = '119935'
pitcher_data_df.loc[pitcher_data_df['Name'] == "Blue Moon Odom", 'key_
```

In [12]:

```
richard_key = '121145'
pitcher_data_df.loc[pitcher_data_df['Name'] == "J. R. Richard", 'key_m
```

In [13]:

```
oil_key = '111312'
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()
```

...

In [32]:
```
"""
pitcher_data_df.to_csv('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
```

...

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

...

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

...

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

...

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

...

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

...

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

. . .

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_mlb
    # 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('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 [ ]:
```
1
```