In [1]:
```python
# MEMORY CLEAN
# import gc
# gc.collect()
from IPython import get_ipython
get_ipython().run_line_magic('reset',  '-sf')
```

In [2]:
```python
# IMPORT
import os as os
import pandas as pd
import numpy as np
import matplotlib as plt
```

In [3]:
```python
# DATA WRANGLER
path1 = os.path.join("openbiomechanics", "baseball_pitching", "data", "metadata.csv")
meta_data = pd.read_csv(path1)
path2 = os.path.join("openbiomechanics", "baseball_pitching", "data", "full_sig", "forces_moments.csv")
data = pd.read_csv(path2)
path3 = os.path.join("openbiomechanics", "baseball_pitching", "data", "full_sig", "force_plate.csv")
force_data = pd.read_csv(path3)
path4 = os.path.join("openbiomechanics", "baseball_pitching", "data", "poi", "poi_metrics.csv")
poi = pd.read_csv(path4)

# data.groupby('session_pitch').count()
# force_data.groupby('session_pitch').count()
# meta_data.groupby('session_pitch').count()

meta_data = meta_data[meta_data.pitch_speed_mph >= 89].sort_values(by=['session_pitch'], ascending=True)
# Filter by pitch speed, sort by ID num
filt_meta_data = meta_data[meta_data['session_pitch'].str.endswith('1')]
# Filter session ID by first trial (..._1)
constr_data = pd.merge(filt_meta_data[['session_pitch']], data, on='session_pitch', how='left')
poi = poi[poi.pitch_speed_mph >= 89].sort_values(by=['session_pitch'], ascending=True)
filt_poi = poi[poi['session_pitch'].str.endswith('1')]
variable_names = [poi.columns]
```

In [4]:
```python
# STATISTICS
mean_age = np.mean(filt_meta_data.age_yrs)
std_age = np.std(filt_meta_data.age_yrs)
mean_height = np.mean(filt_meta_data.session_height_m)
std_height = np.std(filt_meta_data.session_height_m)
mean_mass = np.mean(filt_meta_data.session_mass_kg)
std_mass = np.std(filt_meta_data.session_mass_kg)
# filt_meta_data['playing_level_num'] = filt_meta_data['playing_level'].replace({'college': 1, 'independent': 2, 'milb': 3})
val_counts = filt_meta_data['playing_level'].value_counts()
rel_dist = filt_meta_data['playing_level'].value_counts(normalize=True)
playing_levels = pd.DataFrame({'Playing Level': val_counts.index, 'Amount': val_counts.values, 'Relative Distribution (%)': rel_dist.values * 100})

descriptives_table = {
        'Descriptive': ['Age (years)', 'Height (m)', 'Mass (kg)'],
        'Mean': [mean_age, mean_height, mean_mass],
        'Std Dev': [std_age, std_height, std_mass]
}
```