

Creating and Comparing Pitching Release Metrics

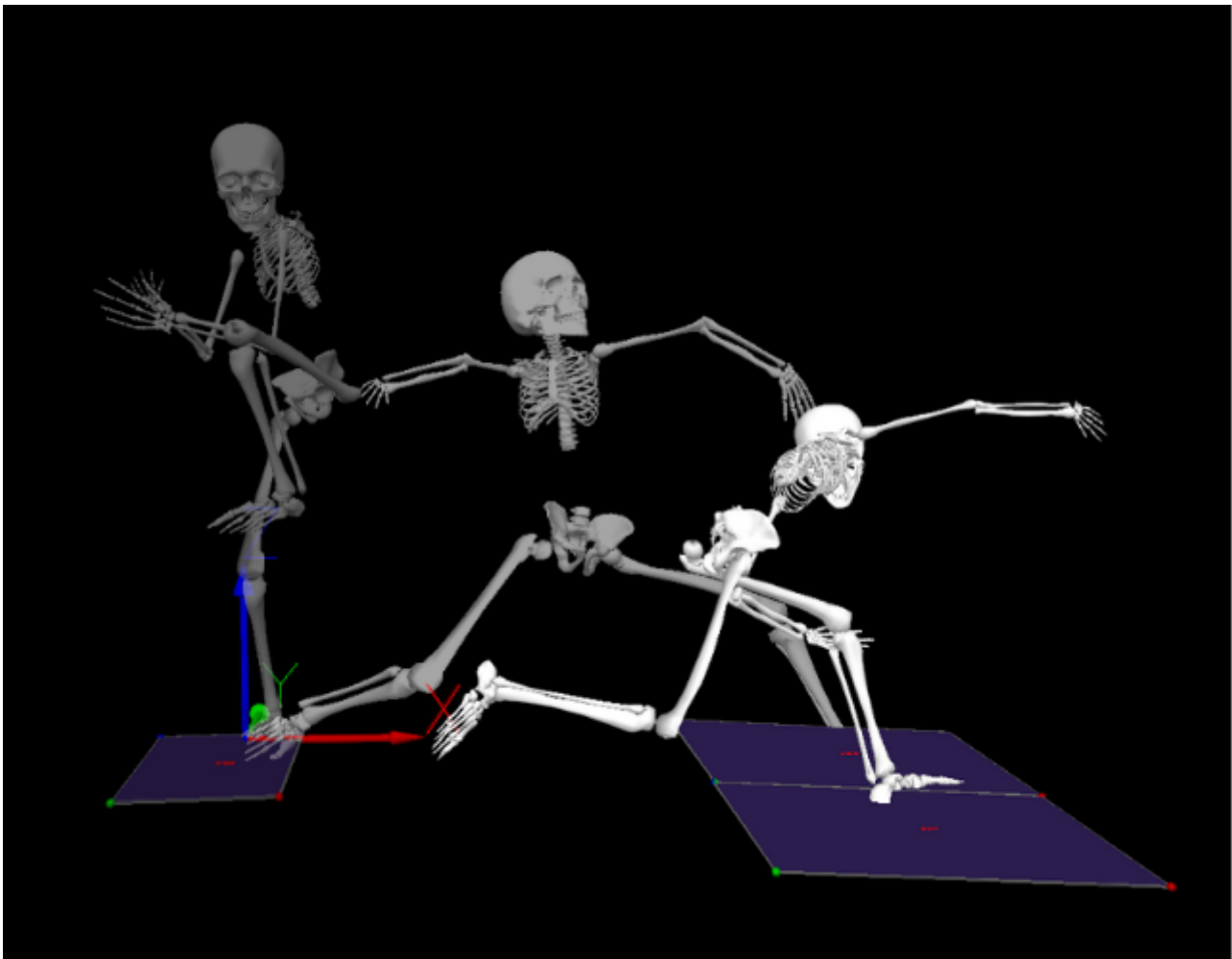
Abstract

One of the most highly documented and discussed areas of biomechanics in sports is a pitcher's throwing motion from the mound. The pitching motion is different from most others in sports in that it occurs from the same origin every time, and several repetitions with highly similar intent and mechanics occur during each outing. As such, the positions that a pitcher achieves at the release point of their throw are the subject of intense scrutiny by scouts, players, and coaches. The positions that a pitcher achieves for each of their pitch types can be crucial in determining the velocity, movement, and effectiveness of each pitch. These positions are often the product of a combination of the pitcher's unique musculoskeletal structure and coaching they have received [1].

This tutorial will describe how to build a pipeline in Visual3D to identify key release metrics (extension, stride length, release height, etc.) and compare them across different groups of pitchers in Sift.

Key Definitions

Extension	Horizontal distance (towards home plate) between the front of the rubber and the throwing hand/ball at the time of release. A pitch that is released at the same velocity but with more extension will appear faster to the batter (known as perceived velocity) as it has to travel a shorter distance.
Stride Length	The distance between the front of the rubber and the left heel marker (for a right handed pitcher) at lead leg plant.
Release Height	The vertical position of the ball relative to the ground at the time of release.
Arm Slot	The angle between the throwing hand/ball and the shoulder joint center, relative to the ground, at the time of release.



Data

This tutorial will uses a public dataset made available by [Driveline Baseball](#). The data set includes motion capture data for 101 different male college and high school pitchers throwing 2-5 pitches each. The dataset is available through Driveline Baseball's [Github repository](#). The OpenBiomechanics Project is an initiative started by Driveline Baseball Research & Development to provide raw (in the form of cleaned C3D files) and processed (full signal + point of interest) sports biomechanics data to the general public.

The pitching data used in this tutorial can be downloaded [here](#).

All of the C3Ds provided by Driveline follow a common naming convention:

Naming Convention	USERid_SESSIONid_HEIGHT_WEIGHT_PITCHNUMBER_PITCHTYPE_PITCHSPEED.c3d
USERid	Unique athlete identifier
SESSIONid	Unique session identifier
HEIGHT	Body height in inches (body height in meters is also provided in the metadata CSV)
WEIGHT	Bodyweight in pounds (body mass in kilograms is also provided in the metadata CSV)
PITCHNUMBER	Pitch number from the athlete’s assessment

PITCHTYPE	Type of pitch thrown
FF	Fastball
PITCHSPEED	Speed of the pitch thrown in miles per hour to one decimal place (with the decimal place removed ex. ~_905.c3d would be a pitch speed of 90.5 mph; ~_950.c3d would be a pitch speed of 95.0 mph)

The marker set, force plate setup, and lab coordinate system can be viewed [Here](#).

Visual3D for Data Processing

Follow the steps below to build and execute the pipeline needed to process the raw dataset. Alternatively, you may download the [completed pipeline](#). The pipeline serves to automate specific processing commands like sorting, event marking, and computing metrics across all files in the dataset. For a more detailed explanation of Visual3D Pipelines visit [this tutorial](#).

Folder structure

In the **baseball_pitching** folder you downloaded, create a new folder named **pitching_processed**, this folder will hold the processed .cmz files produced by the final pipeline.



To start building your pipeline, navigate to **Pipeline** in **Visual3D**.

Creating the main Pipeline

First, we'll call the baseball pitching folder that was just created using 'Set_Pipeline_Parameter_To_Folder_Path'. Change the **PARAMETER_NAME** to 'CALL_SCRIPT_SCRIPT_PATH', and **PARAMETER_VALUE** to the 'c3d' folder within the 'baseball_pitching' folder you just downloaded.

```
Set_Pipeline_Parameter_To_Folder_Path
/PARAMETER_NAME=CALL_SCRIPT_SCRIPT_PATH
/PARAMETER_VALUE=C:\Users\brook\Downloads\baseball
tutorial(s)\baseball_pitching\data\c3d
;
```

Next, we'll use 'Set_Pipeline_Parameter_To_List_Of_Files' to select the cmz files within the folder. Set **FILE_MASK** to *.cmz to apply to the selected files.

```
Set_Pipeline_Parameter_To_List_Of_Files
/PARAMETER_NAME= FILES
/FOLDER=C:\Users\brook\Downloads\baseball
tutorial(s)\baseball_pitching\data\c3d
/FILE_MASK=*.cmz
;
```

With all the files loaded into the pipeline, the next step is to run the for-loop that creates .cmz files.

Add 'For_Each' followed by 'End_For_Each', which closes the loop. The **ITERATION_PARAMETER_NAME** should be set to 'INDEX' for both. This for loop will be used to add the events and calculate the metrics needed to analyze each pitch.

```
For_Each
/ITERATION_PARAMETER_NAME= INDEX
/ITERATION_PARAMETER_COUNT_NAME=COUNT
/ITEMS= ::FILES
;

End_For_Each
/ITERATION_PARAMETER_NAME= INDEX
;
```

Inside the loop, we will create a new file, open it, then save the file. These commands should go between 'For_Each' and 'End_For_Each'. Ensure the **FOLDER** in the "File_Save_As" command is the correct file path to the folder we created.

```
File_New
;

File_Open
/FILE_NAME= ::INDEX
;

File_Save_As
/FILE_NAME=TRIAL&::COUNT
/FOLDER=C:\Users\brook\Downloads\baseball
tutorial(s)\baseball_pitching\data\pitching_processed
;
```

The following commands in the pipeline will be situated between the 'File_Open' and "File_Save_As" commands. The final order will be shown at the end of this section.

Create Release Event

Several events, like start, end, and leg plant time came pre-made by Driveline. An event for the release time of the pitch does not yet exist and is necessary to identify the release metrics in the next steps. As there is no ball tracking data available for these trials we must create a release event based off of the biomechanical signals provided. Most research suggests that the release of a pitch and the peak throwing hand velocity towards the target occur almost simultaneously [1].

To define the release time event we will use the **Event Global Maximum** command to record when the pitchers throwing hand reaches its maximum velocity towards home plate:

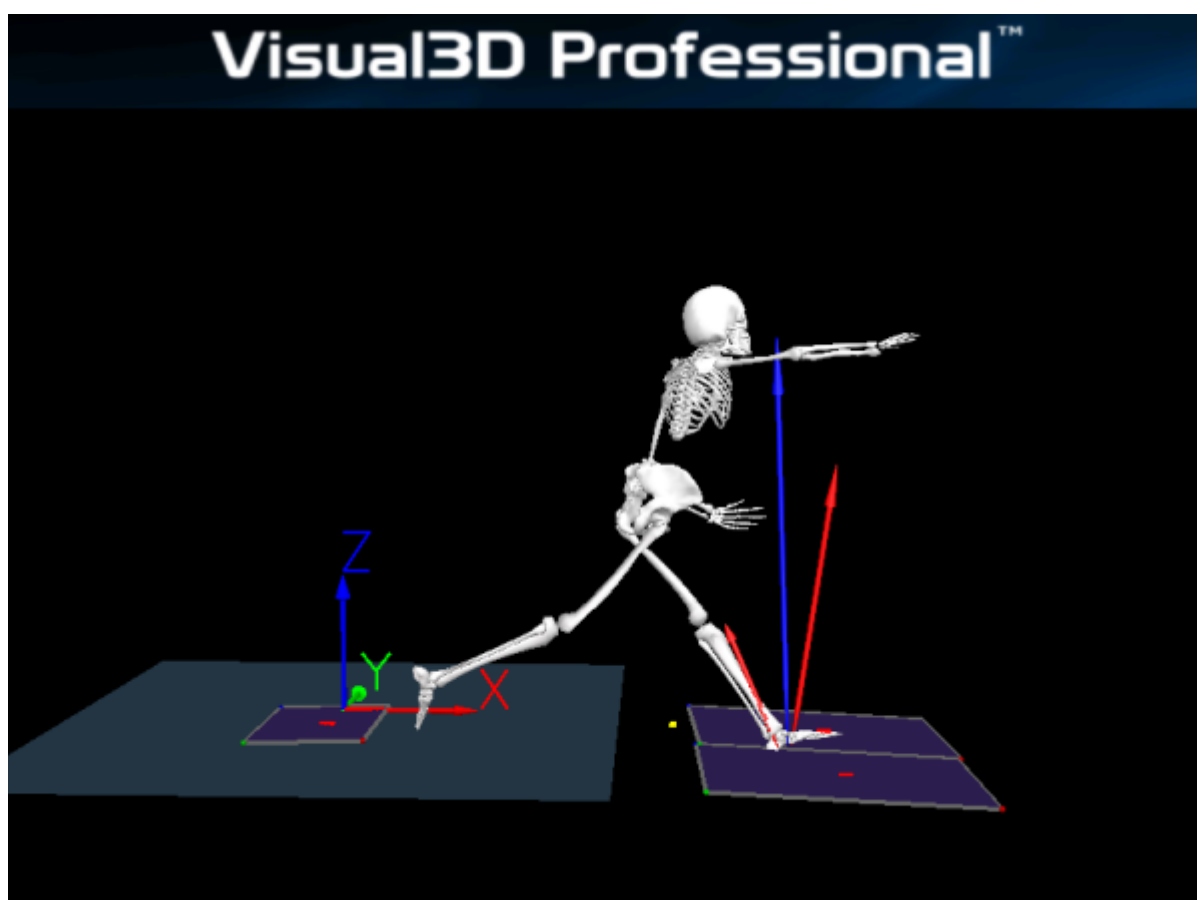
```
Event_Global_Maximum
/RESULT_EVENT_NAME=Release_Time
/SIGNAL_TYPES=Kinetic_Kinematic
```

```

/SIGNAL_FOLDER=RHA
/SIGNAL_NAMES=DistEndVel
/SIGNAL_COMPONENTS=X
! /FRAME_OFFSET=0
! /TIME_OFFSET=
! /EVENT_SEQUENCE=
! /EXCLUDE_EVENTS=
! /EVENT_SEQUENCE_INSTANCE=0
! /EVENT_SUBSEQUENCE=
! /SUBSEQUENCE_EXCLUDE_EVENTS=
! /EVENT_SUBSEQUENCE_INSTANCE=0
! /THRESHOLD=
;

```

To approximate position of the pitcher at the time of release looks like so:



Computing Release Metrics

Now that we have defined the release time for each pitch we can use metric commands to identify several key metrics at the time of release that are frequently used to compare different elite pitching prospects [3].

First, we will use **Metric Signal Value at Event** 4 times to identify the release height, release extension, release depth, and stride length for each pitch.

```
Metric_Signal_Value_At_Event
```

```
/SIGNAL_TYPES=KINETIC_KINEMATIC
/SIGNAL_FOLDER=RHA
/SIGNAL_NAMES=DistEndPos
! /RESULT_METRIC_FOLDER=PROCESSED
/RESULT_METRIC_NAME=Release_Height
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=FALSE
/SIGNAL_COMPONENTS=Z
!/COMPONENT_SEQUENCE=
/EVENT_NAME=Release_Time
! /EVENT_INSTANCE=0
!/SCALE_FACTORS=
/GENERATE_GLOBAL_MEAN_AND_STDDEV=TRUE
/GENERATE_LOCAL_MEAN_AND_STDDEV=FALSE
! /APPEND_TO_EXISTING_VALUES=FALSE
! /GENERATE_VECTOR_LENGTH_METRIC=FALSE
! /RETAIN_NO_DATA_VALUES=FALSE
;
```

```
Metric_Signal_Value_At_Event
/SIGNAL_TYPES=KINETIC_KINEMATIC
/SIGNAL_FOLDER=RHA
/SIGNAL_NAMES=DistEndPos
! /RESULT_METRIC_FOLDER=PROCESSED
/RESULT_METRIC_NAME=Release_Extension
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=FALSE
/SIGNAL_COMPONENTS=X
!/COMPONENT_SEQUENCE=
/EVENT_NAME=Release_Time
! /EVENT_INSTANCE=0
!/SCALE_FACTORS=
/GENERATE_GLOBAL_MEAN_AND_STDDEV=TRUE
/GENERATE_LOCAL_MEAN_AND_STDDEV=FALSE
! /APPEND_TO_EXISTING_VALUES=FALSE
! /GENERATE_VECTOR_LENGTH_METRIC=FALSE
! /RETAIN_NO_DATA_VALUES=FALSE
;
```

```
Metric_Signal_Value_At_Event
/SIGNAL_TYPES=KINETIC_KINEMATIC+KINETIC_KINEMATIC+KINETIC_KINEMATIC
/SIGNAL_FOLDER=RHA+RHA+RHA
/SIGNAL_NAMES=DistEndPos+DistEndPos+DistEndPos
! /RESULT_METRIC_FOLDER=PROCESSED
/RESULT_METRIC_NAME=Release_Point
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=FALSE
/SIGNAL_COMPONENTS=X+Y+Z
!/COMPONENT_SEQUENCE=
/EVENT_NAME=Release_Time
! /EVENT_INSTANCE=0
!/SCALE_FACTORS='1'
```

```
/GENERATE_GLOBAL_MEAN_AND_STDDEV=TRUE
/GENERATE_LOCAL_MEAN_AND_STDDEV=FALSE
! /APPEND_TO_EXISTING_VALUES=FALSE
! /GENERATE_VECTOR_LENGTH_METRIC=FALSE
! /RETAIN_NO_DATA_VALUES=FALSE
;

Metric_Signal_Value_At_Event
/SIGNAL_TYPES=TARGET
/SIGNAL_FOLDER=PROCESSED
/SIGNAL_NAMES=LHEE
! /RESULT_METRIC_FOLDER=PROCESSED
/RESULT_METRIC_NAME=Stride_Length
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=FALSE
/SIGNAL_COMPONENTS=X
!/COMPONENT_SEQUENCE=
/EVENT_NAME=LON
! /EVENT_INSTANCE=0
!/SCALE_FACTORS='1'
/GENERATE_GLOBAL_MEAN_AND_STDDEV=TRUE
/GENERATE_LOCAL_MEAN_AND_STDDEV=FALSE
! /APPEND_TO_EXISTING_VALUES=FALSE
! /GENERATE_VECTOR_LENGTH_METRIC=FALSE
! /RETAIN_NO_DATA_VALUES=FALSE
;
```

Next, we will identify the arm slot (angle between the hand and shoulder joint centers relative to the ground) and record a metric value of the arm slot at release for each pitch:

```
Compute_Planar_Angle
/SIGNAL_TYPES=LANDMARK
! /SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=RSJC+RHJC+VirtualLabY+VirtualLabOrigin
/RESULT_FOLDER=PROCESSED
/RESULT_NAME=Arm_Slot
/COMPUTE_3PT_ANGLE=FALSE
! /NORMALX=
! /NORMALY=
! /NORMALZ=
! /REFERENCE_SEGMENT=LAB
! /PROJECTION_PLANE=XY
! /USE_RIGHT_HAND_RULE=TRUE
/USE_0_TO_360_DEGREES=FALSE
! /MAX_ALLOWABLE_NORMALIZED_DISTANCE_TO_PLANE=0.1
;

Metric_Signal_Value_At_Event
/SIGNAL_TYPES=DERIVED
/SIGNAL_FOLDER=PROCESSED
/SIGNAL_NAMES=Arm_Slot
/RESULT_METRIC_FOLDER=PROCESSED
```

```
/RESULT_METRIC_NAME=Arm_Slot_Angle
! /APPLY_AS_SUFFIX_TO_SIGNAL_NAME=FALSE
/EVENT_NAME=Release_Time
! /EVENT_INSTANCE=0
!/SCALE_FACTORS='1'
/GENERATE_GLOBAL_MEAN_AND_STDDEV=TRUE
/GENERATE_LOCAL_MEAN_AND_STDDEV=FALSE
! /APPEND_TO_EXISTING_VALUES=FALSE
! /GENERATE_VECTOR_LENGTH_METRIC=FALSE
! /RETAIN_NO_DATA_VALUES=FALSE
;
```

Final Pipeline

Now that all of the commands necessary to build the pipeline are complete, ensure that they are in the correct order (shown below).

```
....Set_Pipeline_Parameter_To_Folder_Path
....Set_Pipeline_Parameter_To_List_Of_Files
....For_Each
....File_New
....File_Open
....Event_Global_Maximum
....Metric_Signal_Value_At_Event
....Metric_Signal_Value_At_Event
....Metric_Signal_Value_At_Event
....Metric_Signal_Value_At_Event
....Compute_Planar_Angle
....Metric_Signal_Value_At_Event
....File_Save_As
....End_For_Each
```

You can now execute the pipeline to process each .cmz file and continue to analyzing the data in sift.

Using Sift for Statistical Analysis and Data Visualization

Sift will be used to perform important statistical analysis on our processed pitching files and produce the corresponding visualizations.

Loading Data



To load the processed files into sift, select **Load Library**. Select the 'pitching_processed' folder with the containing the results of our pipeline.

Building Query Definitions

In order to properly identify all data groups for this tutorial a total of 8 queries will be needed, comparing pitches above and below 90 mph for each of the metrics we defined. To complete the tutorial manually follow the steps below. Alternatively, the completed queries can be downloaded here: [Completed Queries](#).

1. Navigate to the Explore tab on the left and select the Query Builder . Click the plus sign beside "Queries" to add a new query definition and name it 'Extension_90+', click save.
2. In the conditions list below your new query add a new condition using the +.
 1. Under the "Signals" tab select "METRIC, PROCESSED, Release_Extension, and X". This command will be used to record the extension towards home plate (in meters) for each pitch over 90 mph.
 2. Under the "Refinement" tab select "Refine using signal". For more information on refinements click [here](#).
 3. Add a new refinement named "over_90" using the "pitch_speed_mph" signal within the metric signals "meta" folder. Set value must be greater than 90 and save the refinement. For queries looking at pitches under 90 we will use the same signals and "value must be less than" 90 to create an "under_90" refinement. Your "over_90" refinement should look as shown to the right.
 4. Click Save at the bottom of the dialog box to save the query.
3. Repeat these steps for the remaining metrics, creating a query for pitches above and below 90 mph for each. For your arm slot queries you will need to select only right handed pitchers using "Refine using tag" and "Use AND Logic" in addition to the "over_90" refinement. Your final list of queries should include the following:

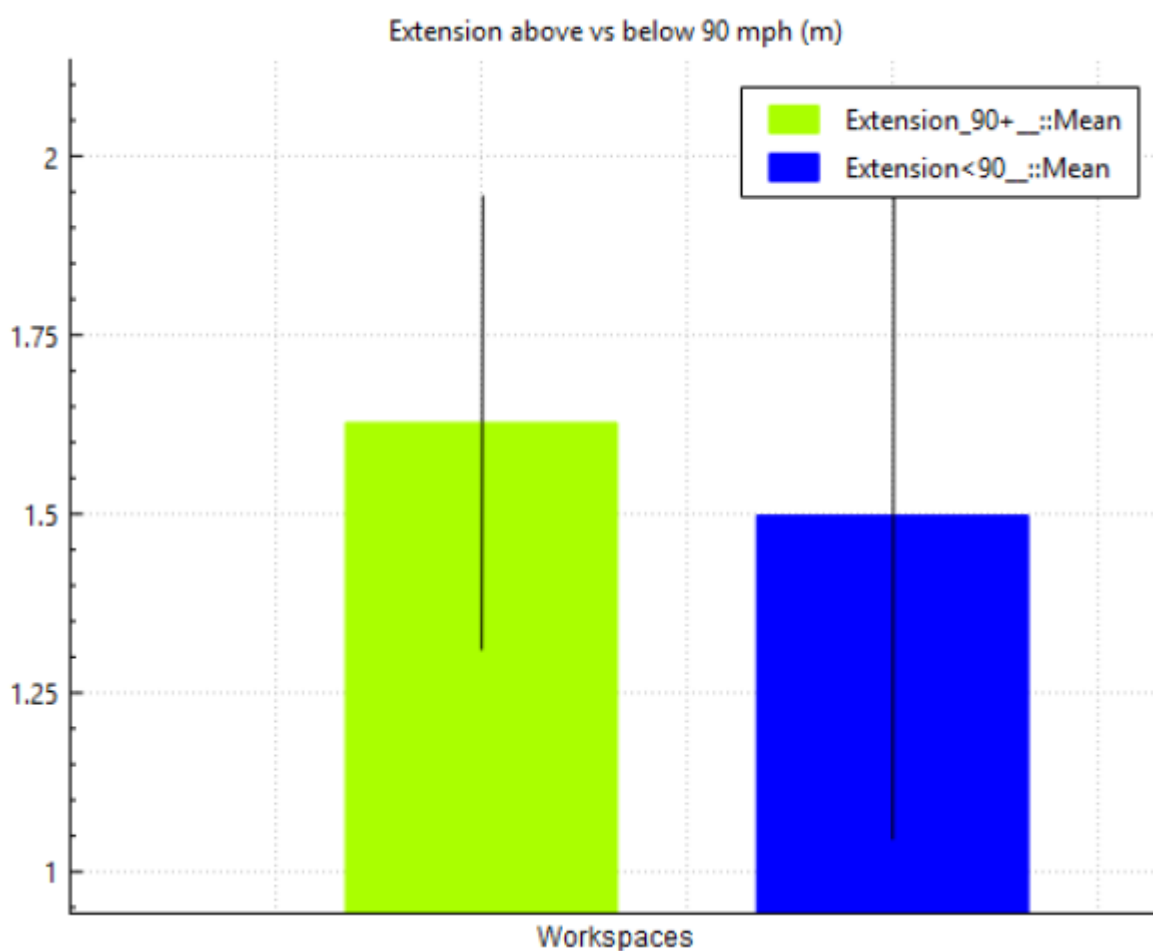
```
Extension_Over_90
Extension_Under_90
Stride_Length_Over_90
Stride_Length_Under_90
Release_Height_Over_90
Release_Height_Under_90
Arm_Slot_Over_90
Arm_Slot_Under_90
```

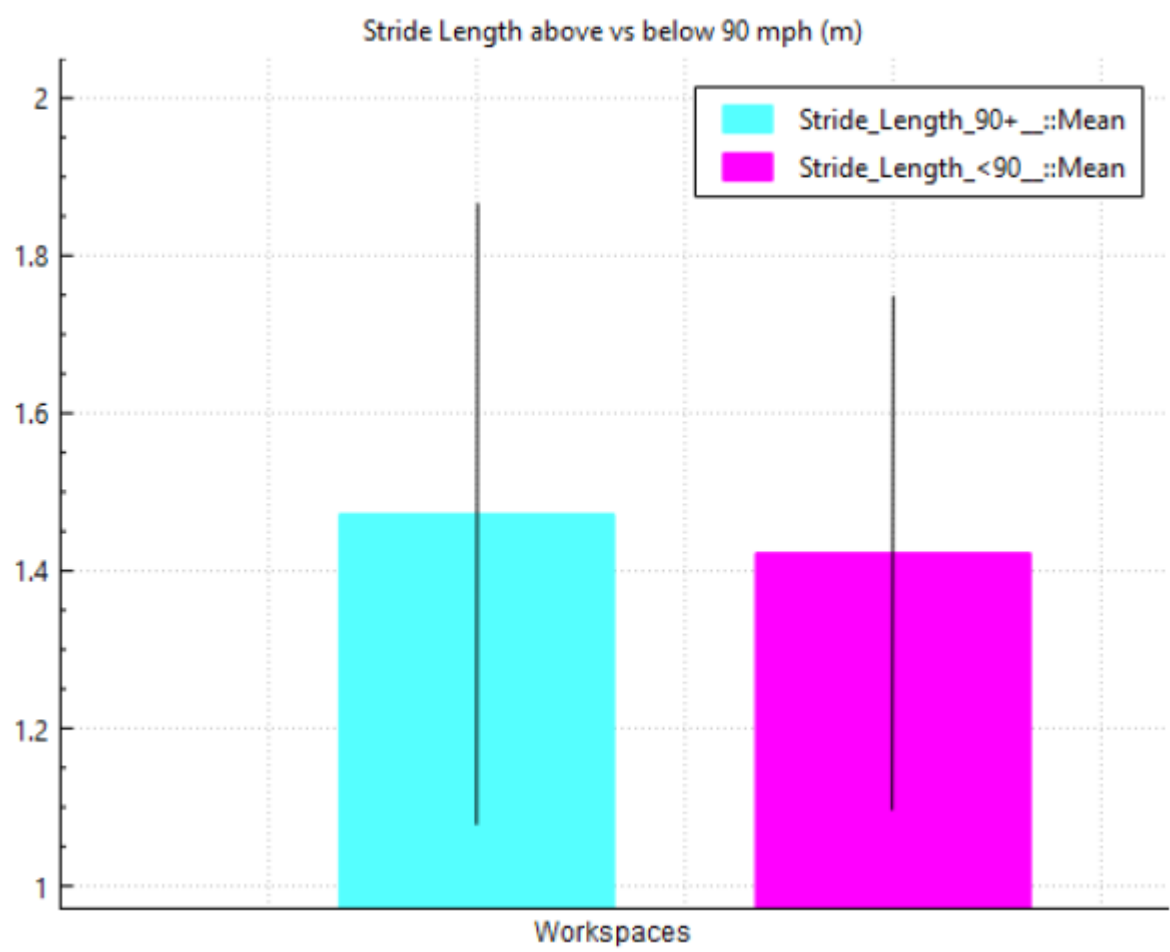
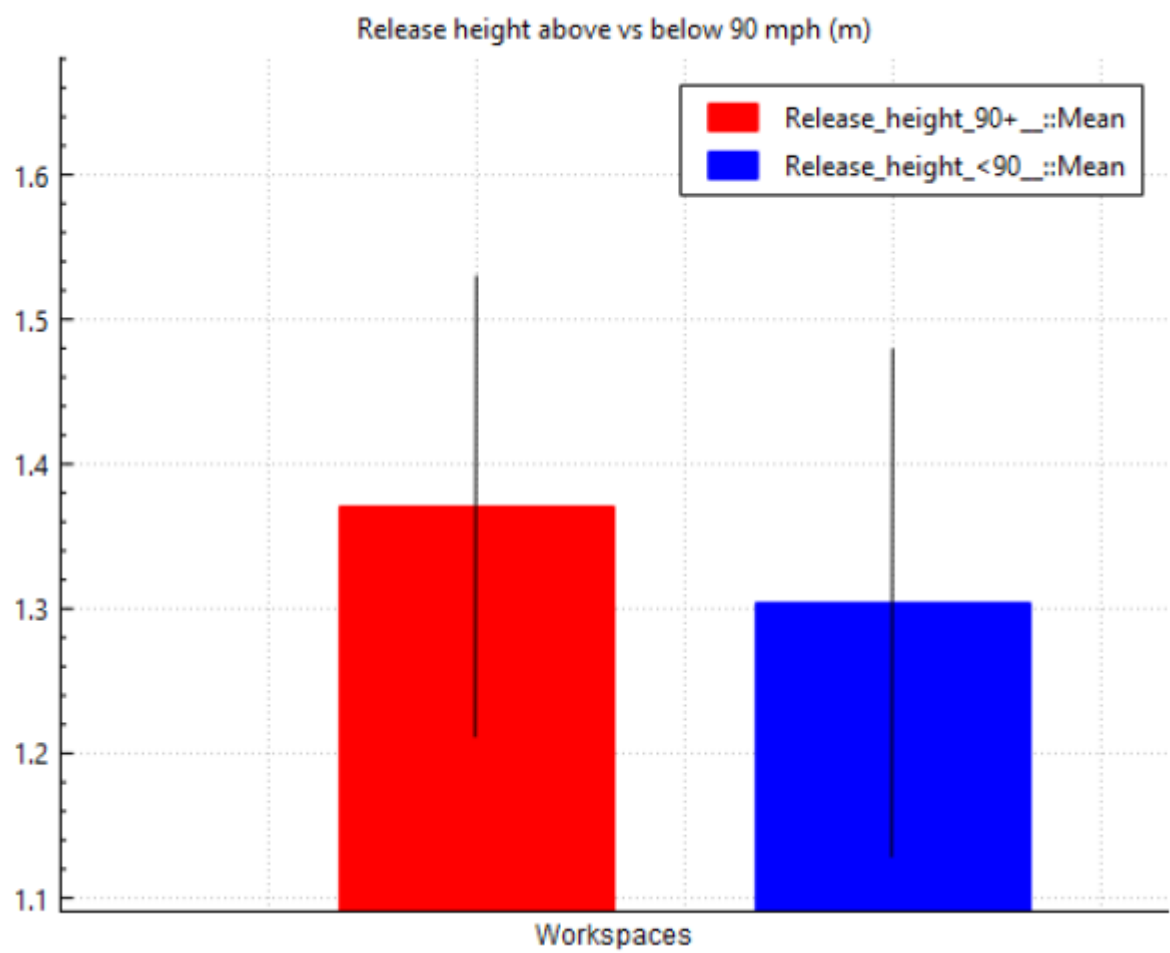
4. Click “Calculate All Queries” at the bottom of the dialog box to load all data.

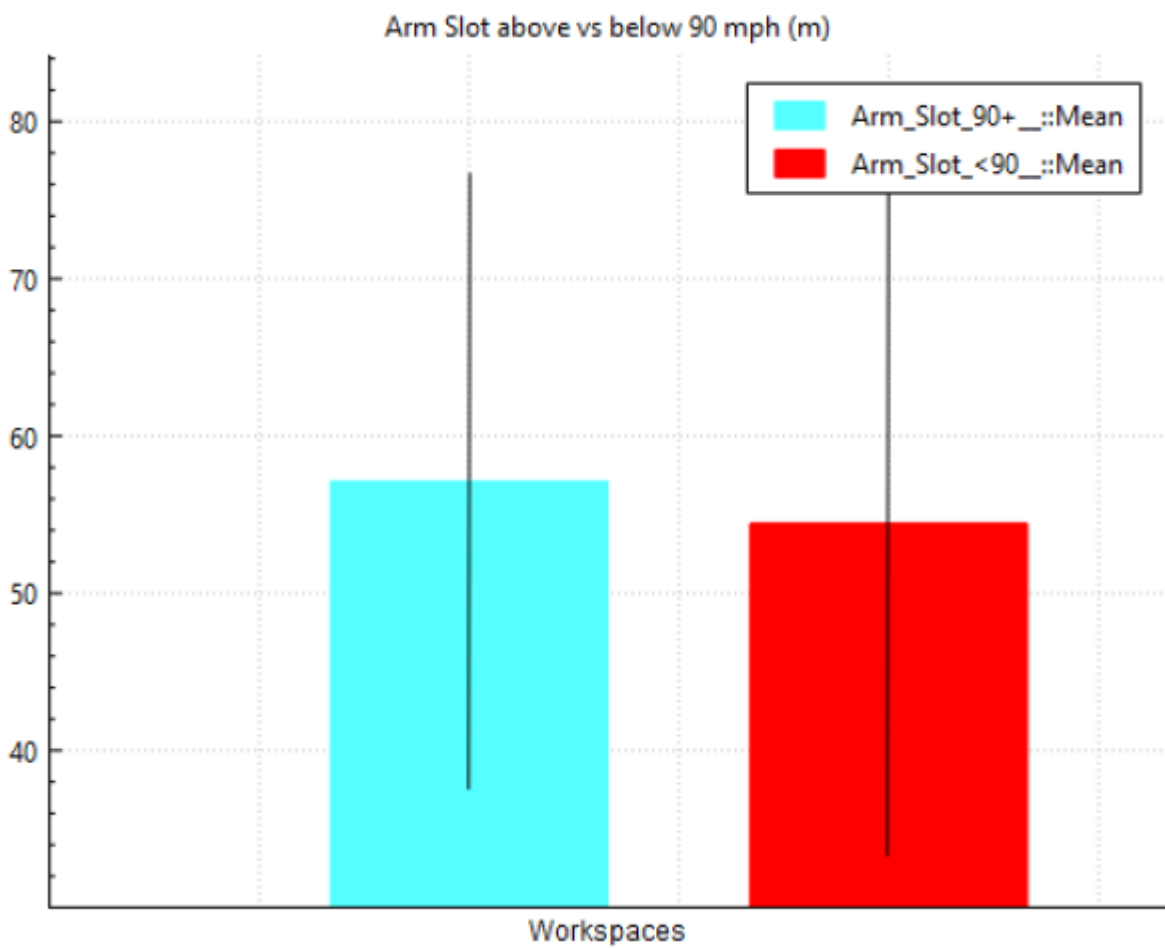
Visualizing Release Metrics

The queries that we built allows us to view bar charts for each of the 8 groups we defined. It makes the most sense to compare just the two groups for each metric at a time.

1. In the explore tab select the 2 groups that you wish to compare. For this tutorial we'll start with 'Extension_Over_90' and 'Extension_Under_90'.
2. Select Metric Plot as the plot type and check Plot Group Mean, Plot Group Dispersion, and Select All Workspaces. Your plots should look similar to below.



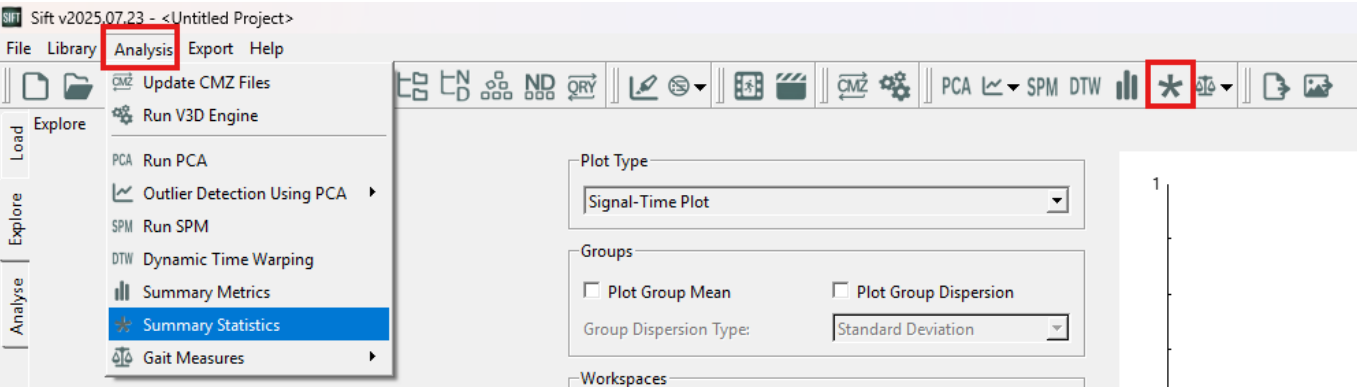




Analyzing Statistical Significance

Now that we have successfully created all of our queries and had a chance to visually compare the metrics in sift, we can use **Summary Statistics** to perform two sample t-tests on each of the metrics we computed and queried.

The Summary Statistics dialog is found on the toolbar and under the Analysis menu.



For each metric we'll use a two sample t test, a significance level of 0.05, and select all workspaces. Set up your t-test for arm slot as shown below, repeat this for release height, stride length, and extension.

The screenshot shows the Sift software interface with the following settings:

- Test Information:**
 - Test Name: Arm_Slot
 - Statistical Test: Two Sample T-Test
 - Significance Level: 0.05
- Select Data:**
 - Group:** Stride_Length_90+, Stride_Length_<90, Release_height_90+, Release_height_<90, Arm_Slot_90+, Arm_Slot_<90
 - Workspaces:** Arm_Slot_90+:TRIAL1.cmz, Arm_Slot_90+:TRIAL10.cmz, Arm_Slot_90+:TRIAL100.cmz, Arm_Slot_90+:TRIAL101.cmz, Arm_Slot_90+:TRIAL11.cmz, Arm_Slot_90+:TRIAL12.cmz
 - ☐ Use Excluded Data
 - ☒ Use All Workspaces
- Result Information:**
 - Result Name: Arm_Slot
 - Test Information:**
 - Statistical Test: Two Sample T Test
 - Significance Level: 0.05
 - Group1: Arm_Slot_90+
 - Group2: Arm_Slot_<90
 - Group Results:**
 - Overall
 - Arm_Slot_90+
 - Arm_Slot_<90
 - Test Results:**
 - Degrees of Freedom: 466
 - Pooled Standard Deviation: 22.9333
 - T Stat: 1.77686
 - Reject Null Hypothesis

Buttons at the bottom: Cancel, Compute Stats Test

The test results that are produced can be used to identify if there are statistically significant differences between the 90+ and less than 90 groups. A higher t stat, with the same degrees of freedom will result in a greater likelihood of the null hypothesis being rejected.

Bonferroni Correction

When testing multiple null hypotheses in the same family the likelihood of a false positive increases with each additional test. To control for the probability of this error a **Bonferroni Correction** can be applied [4]. This correction is done by dividing the original significance threshold (alpha value) by the number of tests being performed to determine a corrected significance threshold.

Our original significance threshold was set at 0.05. In this tutorial we performed 4 t-tests, thus our now significance threshold is 0.0125. The python script shown below uses [this function](#) to take the degrees of freedom calculated by our t-tests and the new alpha value to determine the t score range necessary to reject the null hypothesis for each of our tests.

```
pip install scipy

import scipy.stats

alpha = 0.0125
```

```
df = 412 # degrees of freedom (e.g., sample size - 1)
```

```
# T critical value for a two-tailed test
```

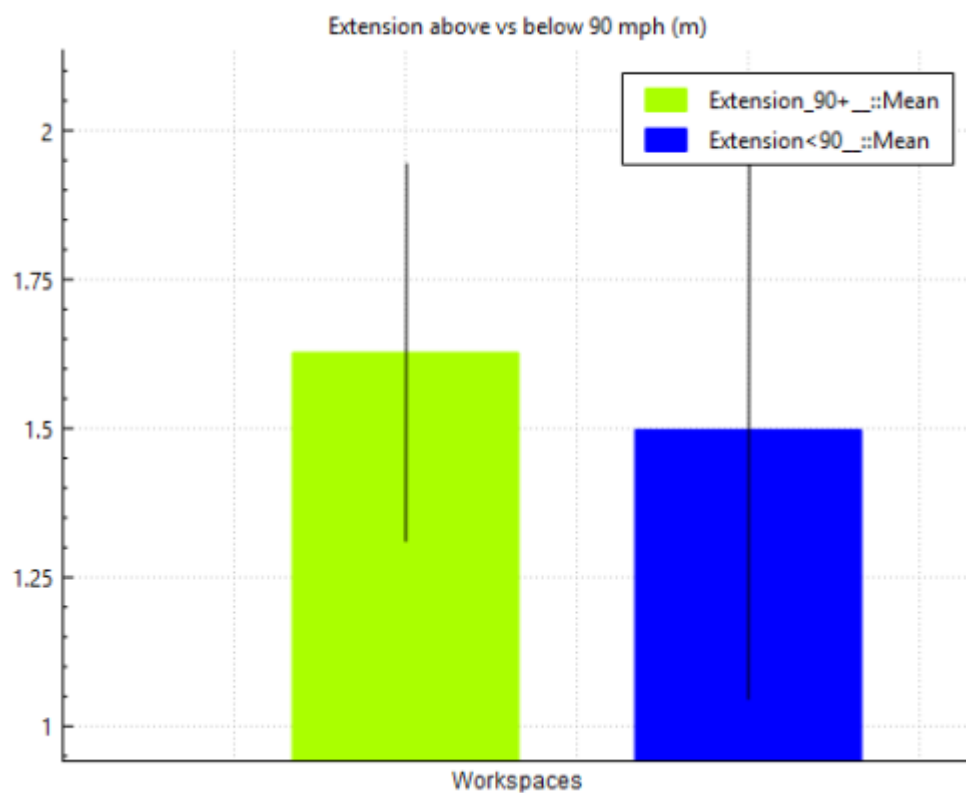
```
t_critical_two = scipy.stats.t.ppf(q=1-alpha/2, df)
```

```
print(f"Two-tailed critical t-value: +/- {t_critical_two:.3f}")
```

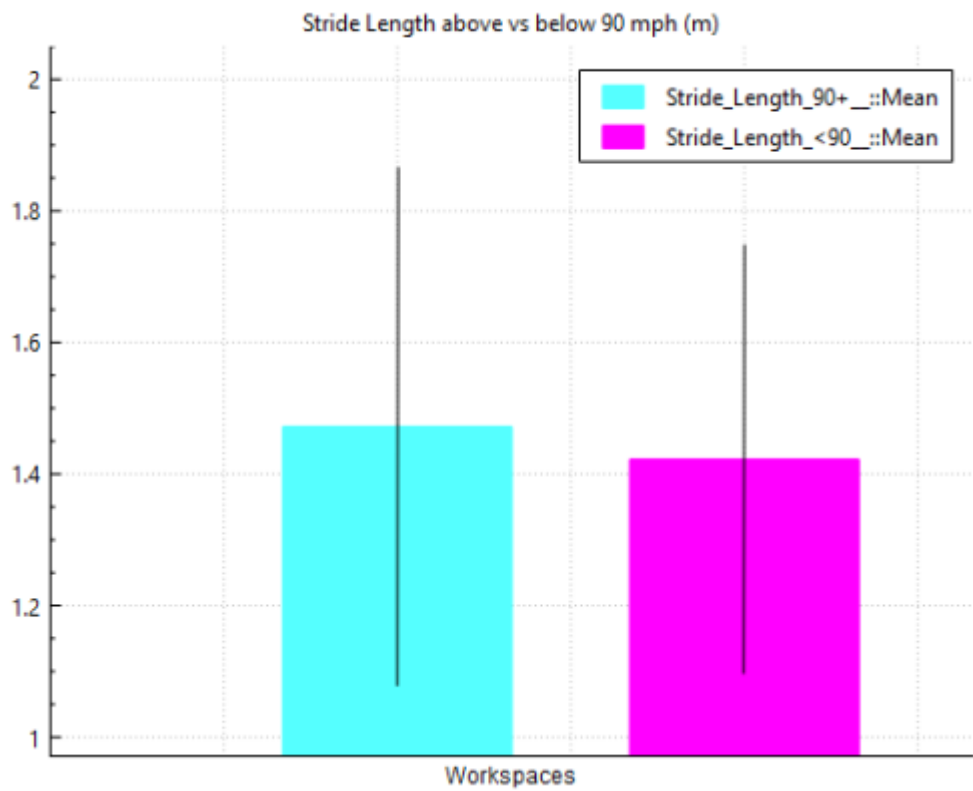
Results

The resulting plots and t-test values that we calculated in the steps above provide some valuable insights into the different patterns exhibited by the fastest pitchers compared to those throwing under 90 mph.

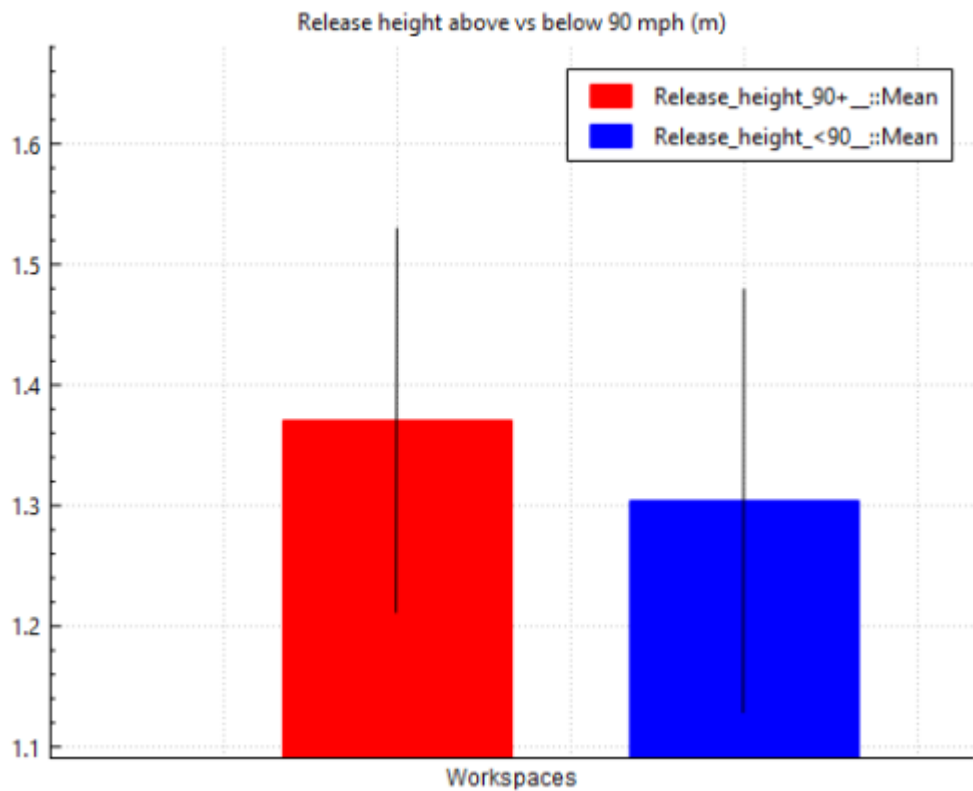
Extension



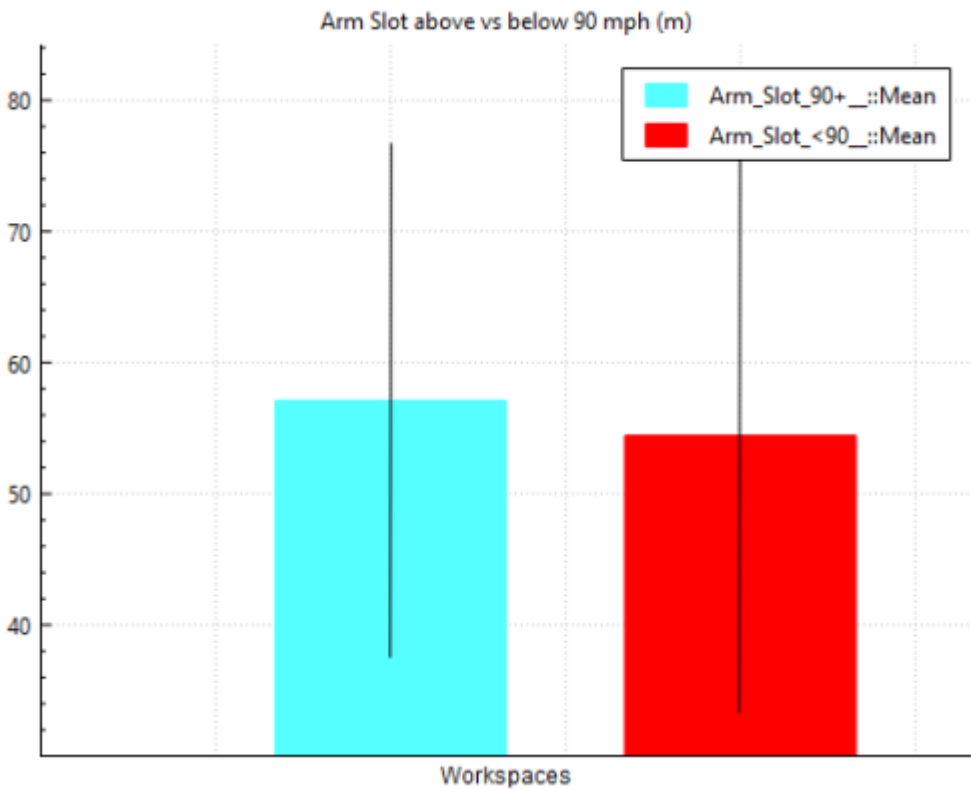
Stride_Length



Release Height



Arm Slot



T-Test Results:

The following table presents the t-test results for each metric comparison we performed. The table includes the t-stat and degrees of freedom reported by Sift, and whether or not the null hypothesis was rejected before and after the Bonferroni Correction. In this case, the null hypothesis is that the mean value of the metric in question (release height, arm slot, stride length, or extension) is the same for pitches thrown above 90 mph and pitches thrown below 90 mph.

Metric	T Stat	DOF	Null Hypothesis State (Significance Level = 0.05)	Corrected Null Hypothesis State (Significance Level = 0.0125)
Release Height	1.88	412	Rejected	Accepted
Arm Slot	0.29	328	Accepted	Accepted
Stride Length	2.23	308	Rejected	Accepted
Extension	1.86	412	Rejected	Accepted

Interpreting Results

Through visual inspection of our plots noticeably higher values in all 4 metrics for faster pitchers were apparent. However, the results from our 4 t-tests told a different story. After applying the Bonferroni correction to our significance threshold to account for false positives, the null hypothesis was accepted for all 4 t-tests. These results tell us that there is likely not a statistically significant trend between any of these mechanical patterns and the resulting fastball velocity.

One key observation that can be made is substantially higher t-stats and lower p-values associated with release height, extension, and stride length in comparison to the results for arm slot. All 3 of these metrics are correlated to pitcher height, suggesting that there may be a link between subject height and average fastball velocity. Some existing research shows that taller pitchers are more likely to throw harder [5]. Further analysis of the dataset could involve separating subjects into groups by

height and including this as part of the queries, or creating metrics like extension divided by height to further isolate these variables and investigate this relationship.

References

- [1]K. A. Giordano, A. Schmitt, A. Nebel, Y. Yanagita, and G. D. Oliver, "Normative In-Game Data for Collegiate Baseball Pitchers Using Markerless Tracking Technology," *Orthopaedic Journal of Sports Medicine*, vol. 12, no. 10, Oct. 2024, doi: <https://doi.org/10.1177/23259671241274137>.
- [2]BAID, "The Science of the Perfect Arm Slot - Motor Preferences in Baseball," *Motor Preferences in Baseball* - <https://www.baseballactionid.com/the-science-of-the-perfect-arm-slot/>!, Sep. 05, 2023.
- [3]Y. Hashimoto, Tomoyuki Nagami, S. Yoshitake, and H. Nakata, "The relationship between pitching parameters and release points of different pitch types in major league baseball players," *Frontiers in sports and active living*, vol. 5, Apr. 2023, doi: <https://doi.org/10.3389/fspor.2023.1113069>.
- [4]Zach, "The Bonferroni Correction: Definition & Example," *Statology*, Feb. 16, 2021. <https://www.statology.org/bonferroni-correction/>
- [5]J. H. Huang, S.-H. Chen, and C. H. Chiu, "Correlation of pitching velocity with anthropometric measurements for adult male baseball pitchers in tryout settings," *PLOS ONE*, vol. 17, no. 3, p. e0265525, Mar. 2022, doi: <https://doi.org/10.1371/journal.pone.0265525>.

From:
<https://has-motion.com/wiki/> - **Wiki Documentation Site**

Permanent link:
https://has-motion.com/wiki/doku.php?id=creating_pitching_release_metrics&rev=1765303544

Last update: **2025/12/09 18:05**

