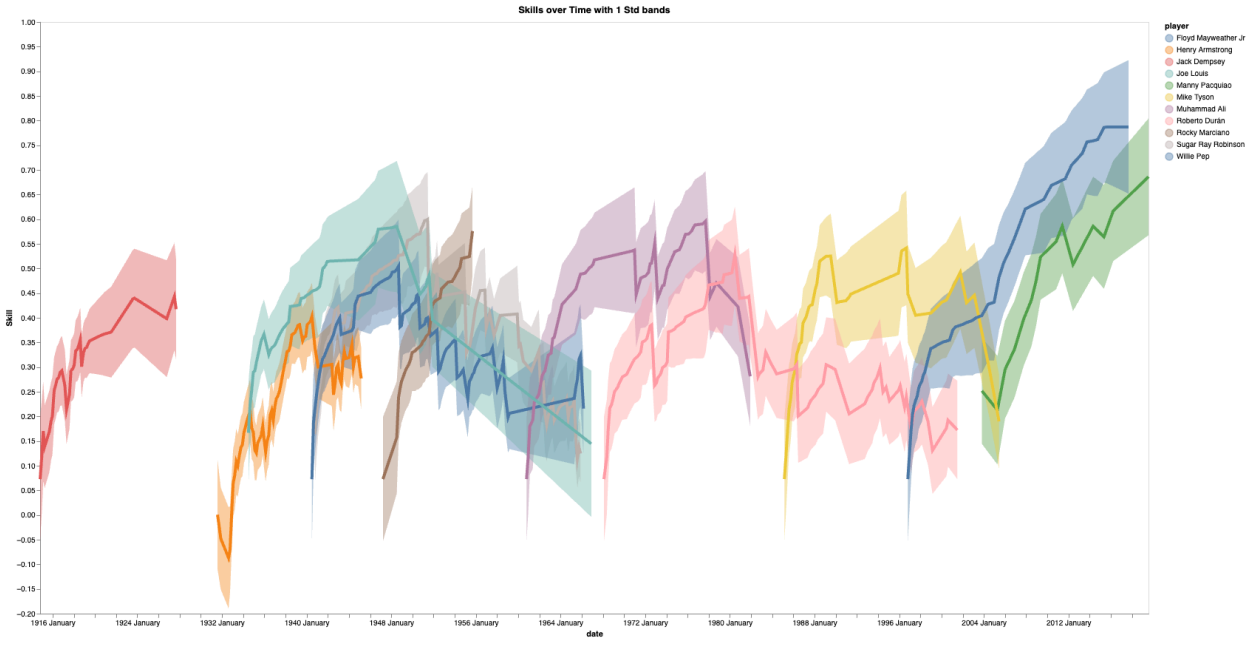
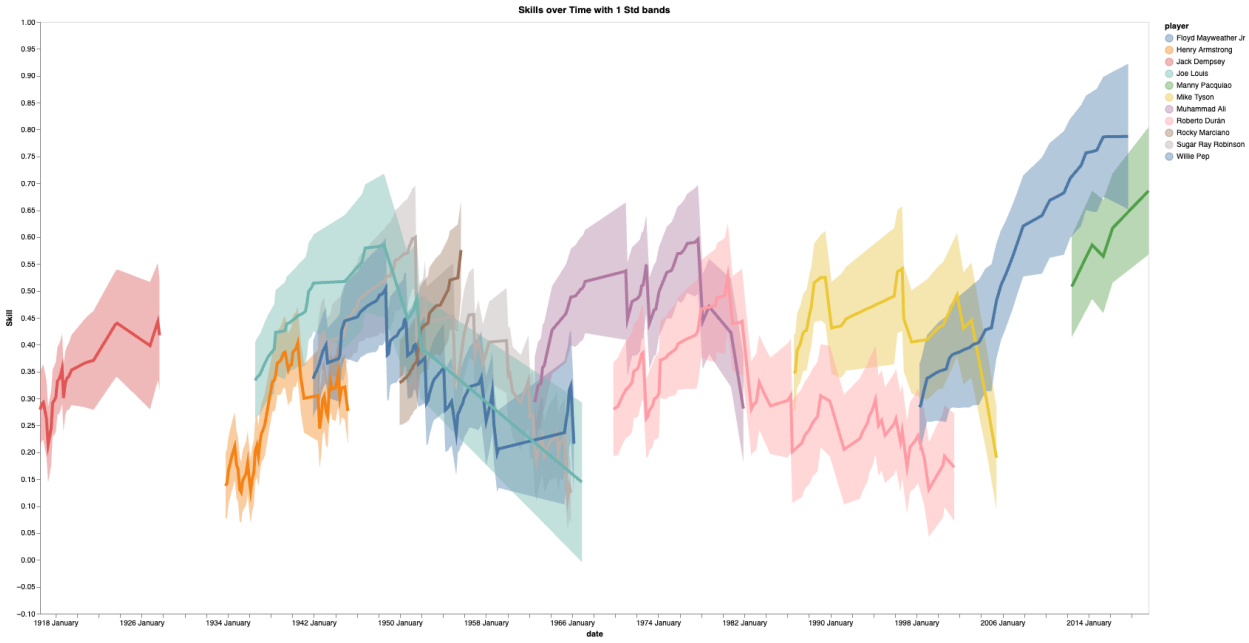
Apply TrueSkillThroughTime algo on the full dataset(25651 matches):

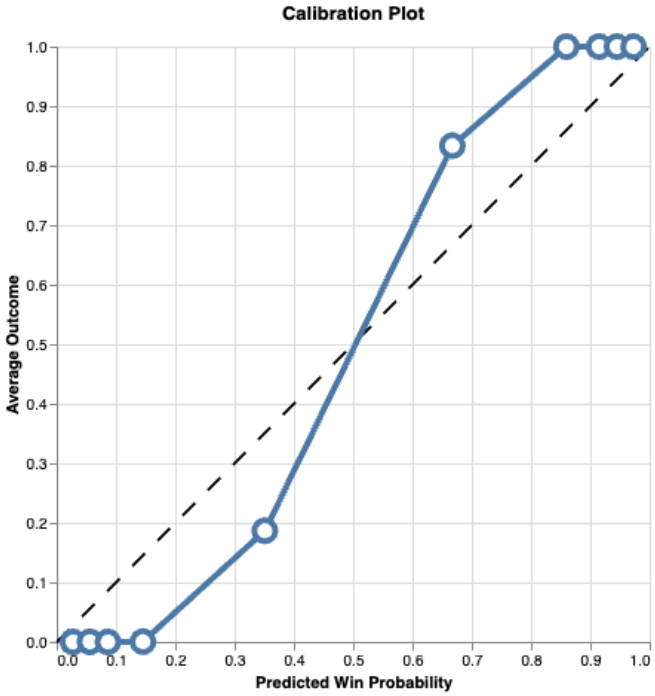
```
top_boxers_online = ['Muhammad Ali', 'Joe Louis', 'Sugar Ray Robinson', 'Rocky Marciano', 'Floyd Mayweather Jr', 'Manny Pacquiao', 'Jack Dempsey', 'Roberto Durán', 'Henry Armstrong', 'Willie Pep', 'Mike Tyson']

self.plot_player_skills(players = top_boxers_online, width=1500, burnin=0)
```



```
self.plot_player_skills(players = top_boxers_online, width=1500, burnin=10)
```




```
self.plot_calibration()
```

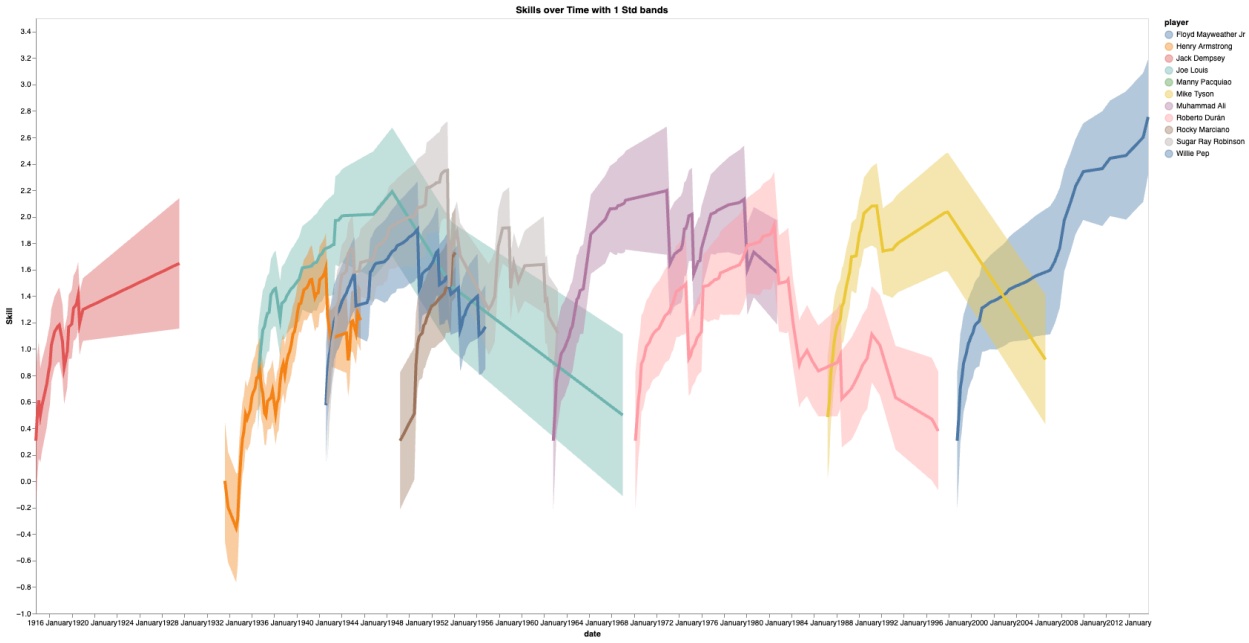


Out-of-sample evaluation:

- We do a 80:20 train test split based on time for each one of 303 players who has played at least 40 matches in the career.
    - data needs to be converted back to the winner-loser-timestamp format in order to apply TrueSkillThroughTime algo.
- We use the first 80% of a player's match to train the model and evaluate his last 20% of matches in the data.
    - (see posts/trueskill/data/oos_eval/games_ge_40_train.csv and posts/trueskill/data/oos_eval/games_ge_40_test.csv)
- To evaluate the algo, for each player in the training set, we take his last point in the skill_curve and use the mu at that point to measure the boxer's performance.
    - It turns out that the accuracy is **73.21%**(i.e. there are 73.21% of matches in the test say whose winner have a higher mu than the loser)
    - We predict the boxer would win the match if his last mu is greater than his opponent's last mu.
    - We make sure there is no overlap data between train and test set. For example game_index 13416, it's Aaron Pryor,1982-11-12,13416,1 in train.csv, however it's Alexis Argüello,1982-11-12,13416,0 in test.csv; It is Aaron Pryor's first 80% of matches in his career however it's Alexis Argüello last 20% of his career.
    - There are only 5% of such overlapping match in the test set. (258 out of 4387 matches)
    - And player not on the 303 player list will have a mu of zero as assumption.

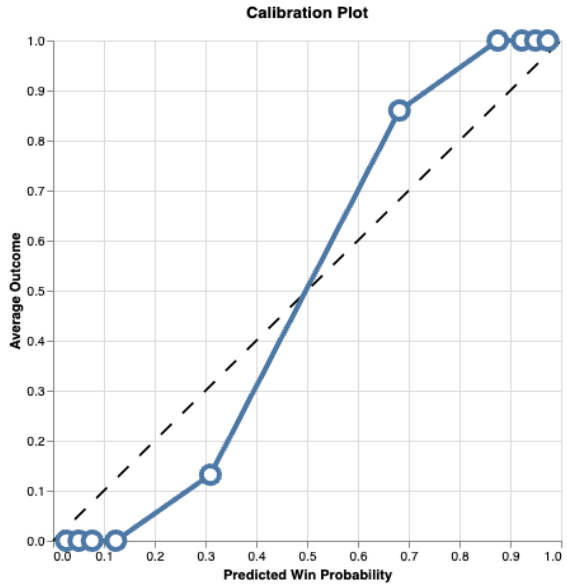
Do a train-test split of data:
Filter only those players who have played at least 40 matches, then for each player, do a 80:20 split by timestamp and apply the algo on the training set(of course the data needs to be converted back to the winner-loser-timestamp format).

```
self.plot_player_skills(players = top_boxers_online[:]+['Mike Tyson'], width=1500,
burnin=0)
```

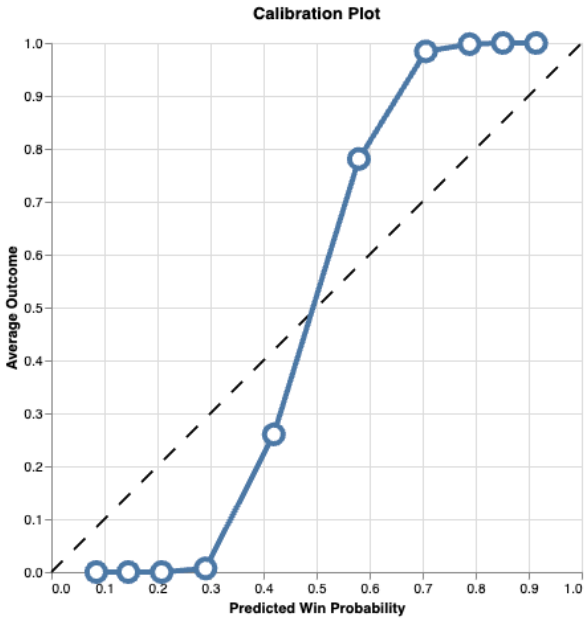


The overall trend for the train set is similar to the result of the full sample, however players on the train set have higher skill levels in absolute value.

Calibration plot on training set - 16946 matches:



Calibration plot on test set - 4129 matches:
There are 258 matches removed from the test set to ensure no overlapping between train and test set



For the entire dataset, here is the statistic that a boxer has played.

```
total_matches_df = games.winner.value_counts().add(games.loser.value_counts(), fill_value=0).sort_values()
```

```
total_matches_df.describe()
```

```
count    16033.000000
mean         3.199775
std         11.276392
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max        280.000000
Name: count, dtype: float64
```

Even after we focus on boxers who have played over 40 matches, we still have to include many boxers who have played few because they are the opponents of those frequently played boxers.

There are 303 players who have played at least 40 matches in their career. The win rate for those players are quite high, which makes boxing a very winner-dominant game.