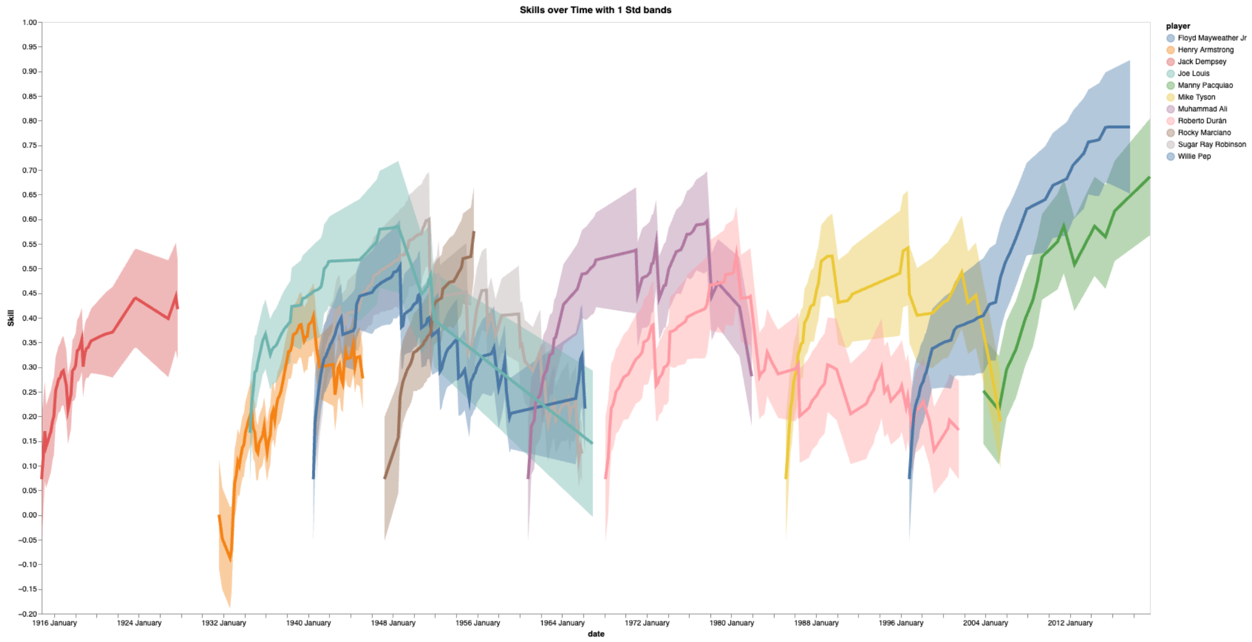
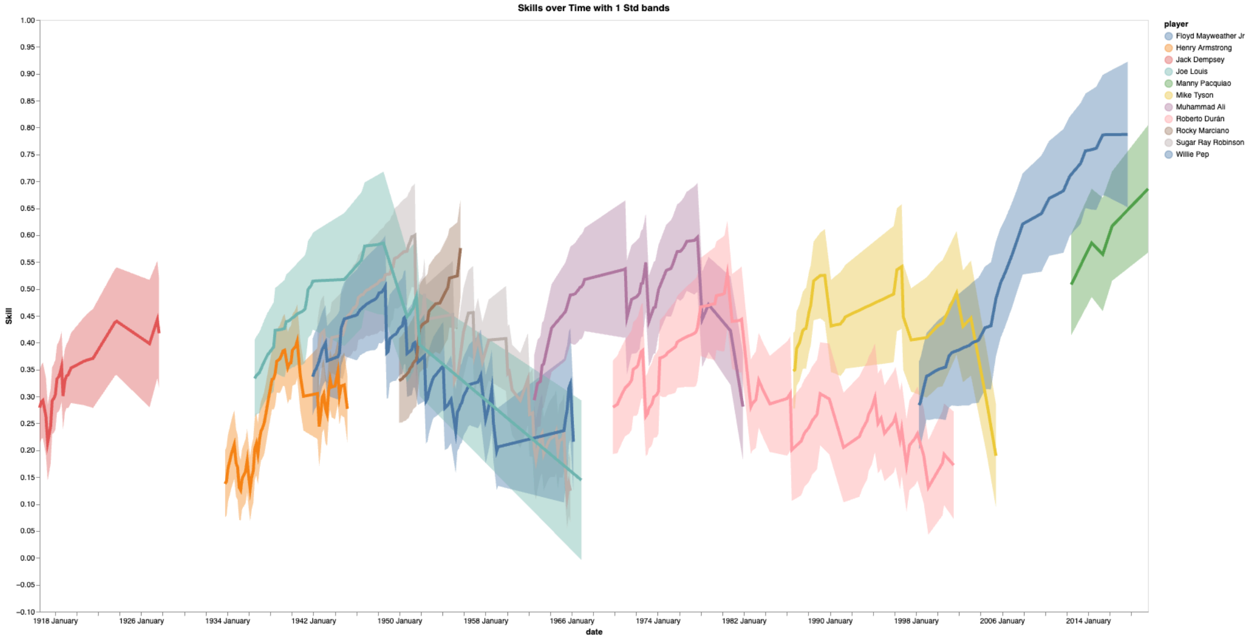
Apply TrueSkillThroughTime algo on the full dataset:

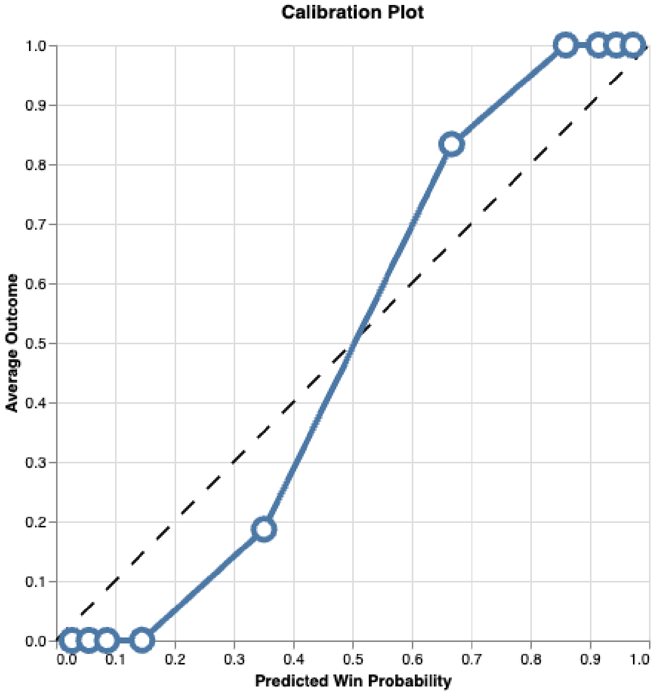
```
top_boxers_online = ['Muhammad Ali', 'Joe Louis', 'Sugar Ray Robinson', 'Rocky
Marciano', 'Floyd Mayweather Jr', 'Manny Pacquiao', 'Jack Dempsey', 'Roberto Durán',
'Henry Armstrong', 'Willie Pep', 'Mike Tyson']

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()
```



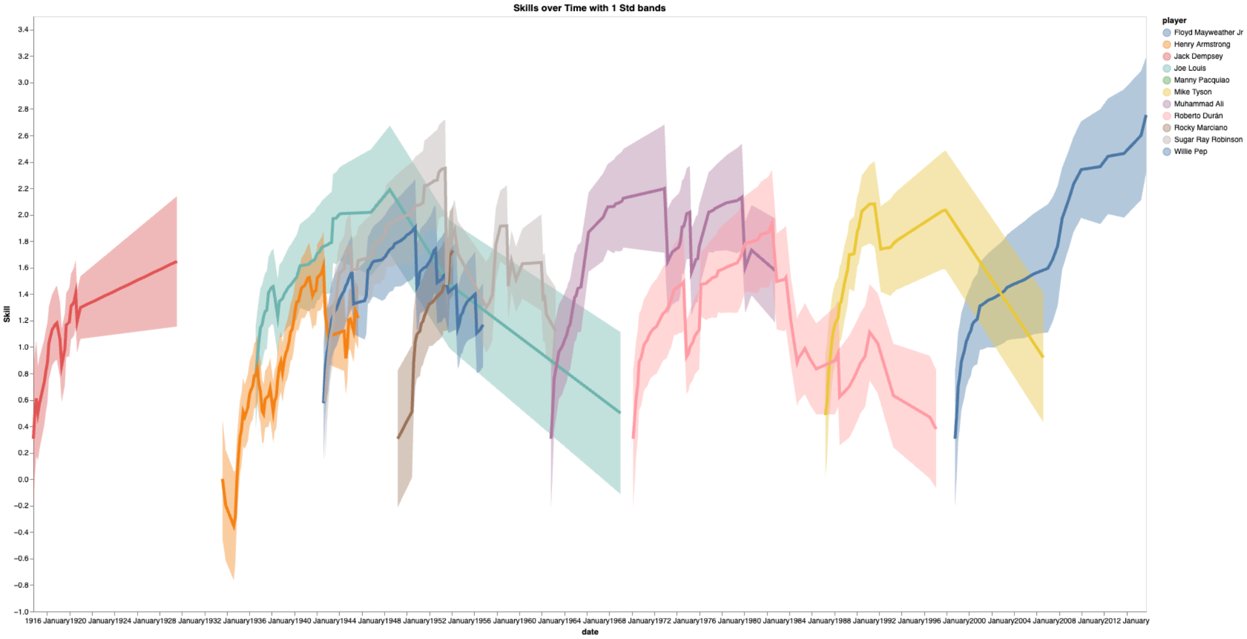
Out-of-sample evaluation:

- 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:

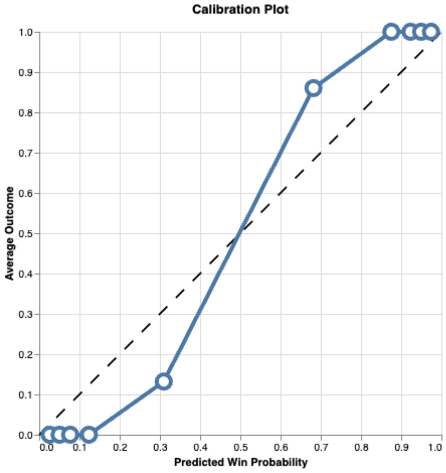
Filter only those players who have played at least 40 matches, then for each player, do a 80:20 split by timestamp and apply the algo on the training set(of course the data needs to be converted back to the winner-loser-timestamp format).

```
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.

```
self.plot_calibration()
```


Calibration Plot
Average Outcome
Predicted Win Probability