Skill estimation of the top 50 PGU Smash Ultimate Professionals using Stan, a simple continuous variable model

In [1]:

```
from platform import python_version
print(python_version())

import pystan
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
3.9.6
```

In [2]:

```
# needed to run multiple chains
import multiprocessing
multiprocessing.set_start_method("fork")
```

We'll use "stan" to define the model; this will compile an inference engine which can then perform Monte Carlo (and some other) approximate inference procedures to give estimates.

We'll describe the (prior) distribution of skill levels as Gaussian, and then model the probability of Player A winning over Player B as a logistic function of the two players' skill differences, with a scaling coefficient (which we can set or try to learn):

Parse data from SQLite database: https://github.com/smashdata/ThePlayerDatabase

In [3]:

```
import sqlite3 as sq
import ast
import parse as data

con = sq.connect("ultimate_player_database/ultimate_player_database.db")
cur = con.cursor()
```

In [4]:

```
# ************************ DATA DICTIONARIES ************************ #

# playerids = set( int(player_id) )
# playerid_to_placement = dict( int(player_id), int(placement) )
# placement_to_playerid = dict( int(placement), int(player_id) )
# playerid_to_name = dict( int(player_id), str(player_tag) )
# name_to_playerid = dict( str(player_tag), int(player_id) )
# matches = [ int(player_1_id), int(player_2_id), int(winner_id) ]
# n = len(playerids)
```

player_ids and player_id_to_placement

In [5]:

```
playerid_to_placement = dict()
placement_to_playerid = dict()

request = cur.execute("select by_id from ranking_seasons limit 1;")
playerid_to_placement_str = ast.literal_eval(request.fetchall()[0][0])

inconsistent_data_playerid = {6905,270293,160103,24835,58557}

change = 0;
for pl_id,place in playerid_to_placement_str.items() :
    if int(pl_id) not in inconsistent_data_playerid :
        playerid_to_placement[int(pl_id)] = place - change
        placement_to_playerid[place - change] = int(pl_id)
    else :
        # hard coded due to inconsistencies in data
        if place != 51:
            change += 1

playerids = {player_id for player_id in playerid_to_placement}
n = len(playerids)
all_player_id_str = ' or '.join(f'player_id = {p_id}' for p_id in playerids)
```

adjust player_ids to correlate to 1-n

In [6]:

```
playerid_to_adjustedid = dict()
adjustedid_to_playerid = dict()

for i,player_id in enumerate(playerids) :
    playerid_to_adjustedid[player_id] = i + 1
    adjustedid_to_playerid[i + 1] = player_id
```

player_id_to_name

In [7]:

```
playerid_to_name = dict()
name_to_playerid = dict()

request = cur.execute(f"select player_id,tag from players where {all_player_ic
players = request.fetchall()
for player_id,tag in players :
    playerid_to_name[int(player_id)] = tag
    name_to_playerid[tag] = int(player_id)
```

matches: every match played between any 2 players from the top 50 players

In [8]:

```
all_p1_ids_str = ' or '.join(f'p1_id = {p_id}' for p_id in playerids)
all_p2_ids_str = ' or '.join(f'p2_id = {p_id}' for p_id in playerids)

request = cur.execute(f'select p1_id, p2_id,winner_id from sets where ({all_p1
matches = request.fetchall()
matches = [ (int(p1), int(p2), int(win)) for p1,p2,win in matches]
print(f'Total number of matches: {len(matches)}')
```

```
Total number of matches: 2976
```

StanModel Template

In [9]:

```
skill_model = """
data {
  int<lower=1> N;             # Total number of players
  int<lower=1> E;             # number of games
  real<lower=0> scale;        # scale value for probability computation
  int<lower=0,upper=1> win[E]; # PA wins vs PB
  int PA[E];                  # player info between each game
  int PB[E];                  #
}
parameters {
  vector[N] skill;           # skill values for each player
}

model{
  for (i in 1:N){ skill[i]~normal(0,3); }
  for (i in 1:E){
    win[i] ~ bernoulli_logit( (scale)*(skill[PA[i]]-skill[PB[i]]) );
  }   # win probability is a logit function of skill difference
}
"""
```

Compile model

In [10]:

```python
import pickle
try:
    sm = pickle.load(open('skill_model.pkl', 'rb'))
except:
    sm = pystan.StanModel(model_code = skill_model)
    with open('skill_model.pkl', 'wb') as f: pickle.dump(sm, f)
```

Evaluate matches and set win, pA, pB lists

In [11]:

```python
win =  [1*(w == p1) for p1,_,w in matches]
pA =  [playerid_to_adjustedid[p1] for p1,_,_ in matches]
pB =  [playerid_to_adjustedid[p2] for _,p2,_ in matches]
```

Perform MCMC on model and take the empirical average over the samples to get the mean estimate for every player's skill level

In [12]:

```python
scales = [i/10 for i in range(1,10)]
skill_levels = []

for scale in scales :
    skill_data = {
        'N': n,
        'E': len(matches),
        'scale': scale,
        'win': win,
        'PA': pA,
        'PB': pB
    }

    fit = sm.sampling(data=skill_data, iter=10000, chains=4)
    samples = fit.extract()

    skill_level = [(adjustedid_to_playerid[i+1], skill) for i,skill in enumera
    skill_level.sort(key = lambda x : -x[1])

    skill_levels.append(skill_level)
```

```
Gradient evaluation took 0.000775 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.75 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000907 seconds
1000 transitions using 10 leapfrog steps per transition would take 9.07 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.001082 seconds
1000 transitions using 10 leapfrog steps per transition would take 10.82 secon
ds.
Adjust your expectations accordingly!



Gradient evaluation took 0.001003 seconds
1000 transitions using 10 leapfrog steps per transition would take 10.03 secon
ds.
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
```

```
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 31.8279 seconds (Warm-up)
               17.1893 seconds (Sampling)
               49.0172 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 34.3123 seconds (Warm-up)
               16.6417 seconds (Sampling)
               50.954 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 33.3824 seconds (Warm-up)
               17.7854 seconds (Sampling)
               51.1678 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 34.601 seconds (Warm-up)
               16.7082 seconds (Sampling)
               51.3091 seconds (Total)


Gradient evaluation took 0.000858 seconds
1000 transitions using 10 leapfrog steps per transition would take 8.58 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.001078 seconds
1000 transitions using 10 leapfrog steps per transition would take 10.78 secon
ds.
Adjust your expectations accordingly!



Gradient evaluation took 0.000918 seconds
1000 transitions using 10 leapfrog steps per transition would take 9.18 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000978 seconds
1000 transitions using 10 leapfrog steps per transition would take 9.78 second
s.
```

```
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 27.7927 seconds (Warm-up)
               17.2116 seconds (Sampling)
               45.0043 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 27.6679 seconds (Warm-up)
               18.2089 seconds (Sampling)
               45.8769 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)
```

```
 Elapsed Time: 29.0292 seconds (Warm-up)
               17.5669 seconds (Sampling)
               46.596 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 29.7424 seconds (Warm-up)
               17.233 seconds (Sampling)
               46.9755 seconds (Total)


Gradient evaluation took 0.00098 seconds
1000 transitions using 10 leapfrog steps per transition would take 9.8 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000721 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.21 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000789 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.89 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.001034 seconds
1000 transitions using 10 leapfrog steps per transition would take 10.34 secon
ds.
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
```

```
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 25.9709 seconds (Warm-up)
               17.1792 seconds (Sampling)
               43.1501 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.9088 seconds (Warm-up)
               16.9476 seconds (Sampling)
               43.8564 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 27.2117 seconds (Warm-up)
               17.8223 seconds (Sampling)
               45.0339 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 28.5106 seconds (Warm-up)
               17.0358 seconds (Sampling)
               45.5464 seconds (Total)


Gradient evaluation took 0.001303 seconds
1000 transitions using 10 leapfrog steps per transition would take 13.03 secon
ds.
Adjust your expectations accordingly!



Gradient evaluation took 0.001191 seconds

1000 transitions using 10 leapfrog steps per transition would take 11.91 secon
ds.
Adjust your expectations accordingly!
```

```
Gradient evaluation took 0.001162 seconds
1000 transitions using 10 leapfrog steps per transition would take 11.62 secon
ds.
Adjust your expectations accordingly!



Gradient evaluation took 0.001369 seconds
1000 transitions using 10 leapfrog steps per transition would take 13.69 secon
ds.
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.3109 seconds (Warm-up)
```

```
               18.6294 seconds (Sampling)
               44.9403 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.1861 seconds (Warm-up)
               18.8925 seconds (Sampling)
               45.0785 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 25.4391 seconds (Warm-up)
               20.5429 seconds (Sampling)
               45.982 seconds (Total)

Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 25.9069 seconds (Warm-up)
               32.0121 seconds (Sampling)
               57.919 seconds (Total)



Gradient evaluation took 0.000765 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.65 second
s.
Adjust your expectations accordingly!


Gradient evaluation took 0.000754 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.54 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000722 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.22 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000722 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.22 second
s.
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
```

```
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 25.1107 seconds (Warm-up)
               17.5761 seconds (Sampling)
               42.6868 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 25.2722 seconds (Warm-up)
               17.4536 seconds (Sampling)
               42.7258 seconds (Total)

Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 25.0348 seconds (Warm-up)
               18.8501 seconds (Sampling)
               43.8849 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.7775 seconds (Warm-up)
               20.7999 seconds (Sampling)
               47.5774 seconds (Total)
```

```
Gradient evaluation took 0.000755 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.55 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000821 seconds
1000 transitions using 10 leapfrog steps per transition would take 8.21 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000737 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.37 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000861 seconds
1000 transitions using 10 leapfrog steps per transition would take 8.61 second
s.
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
```

```
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.3455 seconds (Warm-up)
               18.1933 seconds (Sampling)
               44.5388 seconds (Total)

Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.5279 seconds (Warm-up)
               26.955 seconds (Sampling)
               53.483 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.1434 seconds (Warm-up)
               29.5855 seconds (Sampling)
               55.7288 seconds (Total)

Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 26.705 seconds (Warm-up)
               36.1391 seconds (Sampling)
               62.8441 seconds (Total)


Gradient evaluation took 0.000747 seconds

1000 transitions using 10 leapfrog steps per transition would take 7.47 second
s.
Adjust your expectations accordingly!


Gradient evaluation took 0.000733 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.33 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000734 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.34 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000734 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.34 second
s.
```

```
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 28.4305 seconds (Warm-up)
               20.9756 seconds (Sampling)
               49.4061 seconds (Total)

Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 27.9645 seconds (Warm-up)
               27.0826 seconds (Sampling)
               55.047 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)
```

```
 Elapsed Time: 29.4557 seconds (Warm-up)
               29.6565 seconds (Sampling)
               59.1123 seconds (Total)

Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 27.6739 seconds (Warm-up)
               42.1461 seconds (Sampling)
               69.8201 seconds (Total)


Gradient evaluation took 0.000953 seconds
1000 transitions using 10 leapfrog steps per transition would take 9.53 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000889 seconds
1000 transitions using 10 leapfrog steps per transition would take 8.89 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000752 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.52 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.0014 seconds
1000 transitions using 10 leapfrog steps per transition would take 14 seconds.
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
```

```
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 31.5781 seconds (Warm-up)
               31.5897 seconds (Sampling)
               63.1678 seconds (Total)

Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 31.9472 seconds (Warm-up)
               35.175 seconds (Sampling)
               67.1222 seconds (Total)

Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 31.6051 seconds (Warm-up)
               42.0778 seconds (Sampling)
               73.683 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 32.0281 seconds (Warm-up)
               50.0325 seconds (Sampling)
               82.0606 seconds (Total)



Gradient evaluation took 0.000771 seconds
1000 transitions using 10 leapfrog steps per transition would take 7.71 second
s.
Adjust your expectations accordingly!



Gradient evaluation took 0.000834 seconds
1000 transitions using 10 leapfrog steps per transition would take 8.34 second
s.
Adjust your expectations accordingly!
```

```
Gradient evaluation took 0.001154 seconds
1000 transitions using 10 leapfrog steps per transition would take 11.54 secon
ds.
Adjust your expectations accordingly!



Gradient evaluation took 0.001196 seconds
1000 transitions using 10 leapfrog steps per transition would take 11.96 secon
ds.
Adjust your expectations accordingly!


Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration:    1 / 10000 [  0%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 1000 / 10000 [ 10%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 2000 / 10000 [ 20%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 3000 / 10000 [ 30%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 4000 / 10000 [ 40%]  (Warmup)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 5000 / 10000 [ 50%]  (Warmup)
Iteration: 5001 / 10000 [ 50%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 6000 / 10000 [ 60%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 7000 / 10000 [ 70%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 38.7165 seconds (Warm-up)
               32.8707 seconds (Sampling)
               71.5872 seconds (Total)
```

```
Iteration: 8000 / 10000 [ 80%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 36.9745 seconds (Warm-up)
               38.4573 seconds (Sampling)
               75.4318 seconds (Total)

Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 9000 / 10000 [ 90%]  (Sampling)
Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 38.2327 seconds (Warm-up)
               48.5357 seconds (Sampling)
               86.7684 seconds (Total)

Iteration: 10000 / 10000 [100%]  (Sampling)

 Elapsed Time: 36.1955 seconds (Warm-up)
               61.0011 seconds (Sampling)
               97.1967 seconds (Total)
```

Helper Functions for printing

In [17]:

```
import math

def printResults(skill_levels) :
    mse = 0
    for i in range(n) :
        playerid,_ = skill_levels[i]
        mse += (i+1 - playerid_to_placement[playerid])**2
    mse /= n
    print(f"\nMEAN SQUARED ERROR: {mse}\n")

    print('{:3} {:<12}  {:<16}  {:<15}\n'.format(" ", "ACTUAL", "PREDICTED", "
    for i,(playerid,skill) in enumerate(skill_levels) :
        acutal = playerid_to_name[placement_to_playerid[i+1]]
        predicted = playerid_to_name[playerid]
        print('{:3} {:<12}  {:<16}  {:<15}'.format(i+1, acutal, predicted + f'

    return mse

def printErrorBiasForHigherPlacements(skill_levels) :
    mse = 0
    alpha = 1
    for i in range(n) :
        playerid,_ = skill_levels[i]
        mse += (i+1 - playerid_to_placement[playerid])**2 * math.exp(alpha*(n+
    mse /= n
    print(f"\nMEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: {mse}

    return mse

def printMSEByScale(mse, mse_bias, scales) :

    print('{:3} {:<12}  {:<16}  {:<15}\n'.format(" ", "scale", "MSE", "MSE_Bia
    for mse,mse_b,scale in zip(mse, mse_bias, scales) :
        print('{:3} {:<12}  {:<16}  {:<15}'.format(" ", scale, mse, mse_b))
```

Results

In [14]:

```
mse = []
mse_bias = []

for scale, skill_level in zip(scales, skill_levels) :
    print(f'MCMC Sampling with 10,000 iterations with 4 chains on a StanModel

    mse.append(printResults(skill_level))
    mse_bias.append(printErrorBiasForHigherPlacements(skill_level))

    print('-------------------------------------------------------------------
MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.1

MEAN SQUARED ERROR: 113.76

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         10.979739409030808
  2 Tweek         Tweek (2)         8.820777237980005
  3 Marss         Shuton (5)        7.224228786438038
  4 Samsora       zackray (12)      6.261066748523263
  5 Shuton        ProtoBanham (23)  6.249699229614205
  6 Ally          Glutonny (14)     6.217147782940904
  7 Dabuz         Marss (3)         5.735728106279453
  8 Nairo         Nairo (8)         5.7143598008812875
  9 VoiD          Samsora (4)       5.392393794476569
 10 Light         Light (10)        5.140584921084673
 11 Cosmos        Tea (15)          3.845478185373747
 12 zackray       Dabuz (7)         3.219915948812649
 13 Myran         Salem (22)        2.159916004407288
 14 Glutonny      Nietono (44)      1.9554359011249922
 15 Tea           ESAM (16)         1.899785474449577
 16 ESAM          Ally (6)          1.8536821584584344
 17 MVD           Kameme (19)       1.7479262378894214
 18 Rivers        Cosmos (11)       1.4595485455244295
 19 Kameme        Rivers (18)       1.2178593443746368
 20 Raito         LeoN (34)         0.7680140898415182
 21 justy         VoiD (9)          0.4044067886316225
 22 Salem         Lea (30)          0.3114757996243297
 23 ProtoBanham   Dark Wizzy (31)   -0.030839847147495318
 24 WaDi          Kurama (36)       -0.24719365996420675
 25 Sinji         Myran (13)        -0.8449449087359239
 26 NAKAT         Abadango (39)     -1.0593167221041837
 27 MuteAce       Tsu (43)          -1.0597414288510965
 28 Puppeh        Ryuga (33)        -1.1692000842765824
 29 Stroder Ame   WaDi (24)         -1.5125866004716848
 30 Lea           T (46)            -1.5457653736815387
 31 Dark Wizzy    Secretary (32)    -1.6841121957543608
 32 Secretary     Marcus (42)       -1.960333457758064
 33 Ryuga         Sinji (25)        -2.1843884368225424
 34 LeoN          justy (21)        -2.285595061972718
 35 Mr.R          Stroder Ame (29)  -2.628612899010608
 36 Kurama        MVD (17)          -2.681468967557393
 37 Mr. E         Mr. E (37)        -2.6866702588530575
 38 Goblin        Puppeh (28)       -3.0813425004840074
 39 Abadango      Captain L (49)    -3.5368231075886505
 40 ScAtt         Fatality (47)     -3.8687346462485817
 41 Umeki         MuteAce (27)      -3.948724922729132
 42 Marcus        ScAtt (40)        -4.151260193373242
 43 Tsu           Umeki (41)        -4.187723686029087
 44 Nietono       Mr.R (35)         -4.188714096139262
 45 Frozen        NAKAT (26)        -4.234479038290618
 46 T             ZD (50)           -4.335142527369656
 47 Fatality      Goblin (38)       -5.33042612273482
 48 Suarez        Raito (20)        -6.547301106719237
 49 Captain L     Frozen (45)       -8.624953491959129
 50 ZD            Suarez (48)       -8.87899940659451

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 180.65289712674385
```

```
------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.2

MEAN SQUARED ERROR: 109.36

    ACTUAL        PREDICTED          SKILL

  1 MkLeo         MkLeo (1)          7.152761364250594
  2 Tweek         Tweek (2)          5.744949793593025
  3 Marss         Shuton (5)         5.0442512606050105
  4 Samsora       ProtoBanham (23)   4.76039760894299
  5 Shuton        zackray (12)       4.61368511057346
  6 Ally          Glutonny (14)      4.474759727080356
  7 Dabuz         Nairo (8)          4.212414651039047
  8 Nairo         Marss (3)          4.046456598957886
  9 VoiD          Samsora (4)        3.972917275209029
 10 Light         Light (10)         3.562965842139682
 11 Cosmos        Tea (15)           3.085067875587834
 12 zackray       Dabuz (7)          2.2596031349024575
 13 Myran         Nietono (44)       1.9831838973968239
 14 Glutonny      Ally (6)           1.7003310387497705
 15 Tea           Kameme (19)        1.6488839752775333
 16 ESAM          Salem (22)         1.4530174924189936
 17 MVD           ESAM (16)          1.4156015395835544
 18 Rivers        Cosmos (11)        1.2463247670866275
 19 Kameme        Rivers (18)        0.8526691730019058
 20 Raito         VoiD (9)           0.7533113673092295
 21 justy         Lea (30)           0.7213706019535717
 22 Salem         LeoN (34)          0.396432927931427
 23 ProtoBanham   Dark Wizzy (31)    0.21039827456927132
 24 WaDi          Kurama (36)        0.10551397496987854
 25 Sinji         Abadango (39)      -0.23975320363826022
 26 NAKAT         Tsu (43)           -0.27087470897022553
 27 MuteAce       Myran (13)         -0.37946204810514583
 28 Puppeh        T (46)             -0.5549855886530025
 29 Stroder Ame   WaDi (24)          -0.9388184290485401
 30 Lea           Ryuga (33)         -1.445458779357731
 31 Dark Wizzy    MVD (17)           -1.5148678929580988
 32 Secretary     justy (21)         -1.5342435597463961
 33 Ryuga         Marcus (42)        -1.5537968557788457
 34 LeoN          Secretary (32)     -1.7163620771475923
 35 Mr.R          Sinji (25)         -1.820946957303965
 36 Kurama        Stroder Ame (29)   -1.9152310748907855
 37 Mr. E         Mr. E (37)         -1.9793957465046712
 38 Goblin        Puppeh (28)        -2.3406825377350473
 39 Abadango      MuteAce (27)       -2.368321265904996
 40 ScAtt         Umeki (41)         -2.4176071031326525
 41 Umeki         Fatality (47)      -2.9749894254385865
 42 Marcus        ZD (50)            -3.1920305434083374
 43 Tsu           Mr.R (35)          -3.2109834688267065
 44 Nietono       Goblin (38)        -3.423071105262377
 45 Frozen        ScAtt (40)         -3.6400783893201742
 46 T             NAKAT (26)         -3.64024116999197
 47 Fatality      Raito (20)         -3.9232026942065845
 48 Suarez        Captain L (49)     -3.9795190144864887
 49 Captain L     Suarez (48)        -6.1090906552703625
 50 ZD            Frozen (45)        -7.996403286224477
```

```
MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 171.1471931049876

-------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.3

MEAN SQUARED ERROR: 109.56

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         5.106565778662765
  2 Tweek         Tweek (2)         4.101689616482888
  3 Marss         Shuton (5)        3.7033681245687213
  4 Samsora       ProtoBanham (23)  3.580347110525753
  5 Shuton        zackray (12)      3.4359536943859785
  6 Ally          Glutonny (14)     3.2796865913007336
  7 Dabuz         Nairo (8)         3.1040072908078846
  8 Nairo         Marss (3)         2.951956528082662
  9 VoiD          Samsora (4)       2.931947483123424
 10 Light         Light (10)        2.5863157698254486
 11 Cosmos        Tea (15)          2.339141765698015
 12 zackray       Dabuz (7)         1.6494729156447978
 13 Myran         Nietono (44)      1.6346467796540598
 14 Glutonny      Kameme (19)       1.3243701089692945
 15 Tea           Ally (6)          1.3199899744609596
 16 ESAM          ESAM (16)         1.0570591432430594
 17 MVD           Salem (22)        1.0471284589062786
 18 Rivers        Cosmos (11)       0.9592905169087675
 19 Kameme        Lea (30)          0.7075764119681116
 20 Raito         VoiD (9)          0.6575136892232316
 21 justy         Rivers (18)       0.5970877213974327
 22 Salem         LeoN (34)         0.2435575849338195
 23 ProtoBanham   Dark Wizzy (31)   0.2008967212134101
 24 WaDi          Kurama (36)       0.15604411161167958
 25 Sinji         Abadango (39)     -0.011909279015859637
 26 NAKAT         Tsu (43)          -0.035357792238156015
 27 MuteAce       Myran (13)        -0.22598739700800494
 28 Puppeh        T (46)            -0.23259996608758174
 29 Stroder Ame   WaDi (24)         -0.6747453462982815
 30 Lea           MVD (17)          -1.030089768699379
 31 Dark Wizzy    justy (21)        -1.1269868630038735
 32 Secretary     Marcus (42)       -1.1976589513304954
 33 Ryuga         Stroder Ame (29)  -1.3807055468812224
 34 LeoN          Ryuga (33)        -1.3856946980639266
 35 Mr.R          Sinji (25)        -1.4231659869729207
 36 Kurama        Secretary (32)    -1.4455507797288105
 37 Mr. E         Mr. E (37)        -1.4874291550516572
 38 Goblin        Umeki (41)        -1.5930030863207645
 39 Abadango      MuteAce (27)      -1.6298467728905384
 40 ScAtt         Puppeh (28)       -1.756866661591838
 41 Umeki         Fatality (47)     -2.2412421868339547
 42 Marcus        ZD (50)           -2.364293903732143
 43 Tsu           Goblin (38)       -2.427799597736325
 44 Nietono       Mr.R (35)         -2.435066746512953
 45 Frozen        Raito (20)        -2.660565809856121
 46 T             ScAtt (40)        -2.8479589541535106
 47 Fatality      NAKAT (26)        -2.873892679189704
 48 Suarez        Captain L (49)    -3.5081838972052415
 49 Captain L     Suarez (48)       -4.451964147983954
 50 ZD            Frozen (45)       -6.630454382981157

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 170.67117115204704

------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.4

MEAN SQUARED ERROR: 113.2

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         3.962979026616765
  2 Tweek         Tweek (2)         3.1875505902825005
  3 Marss         Shuton (5)        2.913647223975868
  4 Samsora       ProtoBanham (23)  2.8430601615432085
  5 Shuton        zackray (12)      2.7206559395132857
  6 Ally          Glutonny (14)     2.584530955175557
  7 Dabuz         Nairo (8)         2.446648788188236
  8 Nairo         Marss (3)         2.3183143845642533
  9 VoiD          Samsora (4)       2.3107121863197317
 10 Light         Light (10)        2.0327745657720158
 11 Cosmos        Tea (15)          1.8718757168864155
 12 zackray       Nietono (44)      1.3698983452362943
 13 Myran         Dabuz (7)         1.3048535717268162
 14 Glutonny      Kameme (19)       1.1059001691076447
 15 Tea           Ally (6)          1.086736881864433
 16 ESAM          ESAM (16)         0.8540793810106189
 17 MVD           Salem (22)        0.8275024384870587
 18 Rivers        Cosmos (11)       0.7810320543225966
 19 Kameme        Lea (30)          0.643443350149947
 20 Raito         VoiD (9)          0.5675058107951976
 21 justy         Rivers (18)       0.4768568990079621
 22 Salem         LeoN (34)         0.19930669772420664
 23 ProtoBanham   Dark Wizzy (31)   0.19504366359154893
 24 WaDi          Kurama (36)       0.168689373058287
 25 Sinji         Tsu (43)          0.06060456559845363
 26 NAKAT         Abadango (39)     0.06037790436387337
 27 MuteAce       T (46)            -0.09879402832497543
 28 Puppeh        Myran (13)        -0.14161559648265437
 29 Stroder Ame   WaDi (24)         -0.48880182003946604
 30 Lea           MVD (17)          -0.7505823620120075
 31 Dark Wizzy    justy (21)        -0.8491102441635727
 32 Secretary     Marcus (42)       -0.9209607005011301
 33 Ryuga         Stroder Ame (29)  -1.0381837487418275
 34 LeoN          Sinji (25)        -1.1156936251024265
 35 Mr.R          Mr. E (37)        -1.1488427798681315
 36 Kurama        Umeki (41)        -1.1511119822593725
 37 Mr. E         Secretary (32)    -1.1624508582485524
 38 Goblin        Ryuga (33)        -1.179996499404393
 39 Abadango      MuteAce (27)      -1.206042978493296
 40 ScAtt         Puppeh (28)       -1.3480011993437204
 41 Umeki         Fatality (47)     -1.742667245355798
 42 Marcus        ZD (50)           -1.8127042018940196
 43 Tsu           Goblin (38)       -1.8428955134021394
 44 Nietono       Mr.R (35)         -1.9065551638779419
 45 Frozen        Raito (20)        -1.9639057655066408
 46 T             ScAtt (40)        -2.2587025186587217
 47 Fatality      NAKAT (26)        -2.296035945706626
 48 Suarez        Captain L (49)    -2.9723491383275404
 49 Captain L     Suarez (48)       -3.431039777483951
 50 ZD            Frozen (45)       -5.478180649903033

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 175.65266193560356

------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.5

MEAN SQUARED ERROR: 113.48

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         3.2201291697964884
  2 Tweek         Tweek (2)         2.594243999859888
  3 Marss         Shuton (5)        2.3908936856110694
  4 Samsora       ProtoBanham (23)  2.3485603864167532
  5 Shuton        zackray (12)      2.2411621253415555
  6 Ally          Glutonny (14)     2.115474636865826
  7 Dabuz         Nairo (8)         2.003743352934233
  8 Nairo         Marss (3)         1.8997275287545181
  9 VoiD          Samsora (4)       1.8936245874521613
 10 Light         Light (10)        1.6662043132503916
 11 Cosmos        Tea (15)          1.5516371169646908
 12 zackray       Nietono (44)      1.1644305480502974
 13 Myran         Dabuz (7)         1.075991951949985
 14 Glutonny      Kameme (19)       0.9302073784388558
 15 Tea           Ally (6)          0.9062647725749297
 16 ESAM          ESAM (16)         0.7059516208979129
 17 MVD           Salem (22)        0.6862290616985751
 18 Rivers        Cosmos (11)       0.6525746286061626
 19 Kameme        Lea (30)          0.569688542827844
 20 Raito         VoiD (9)          0.49193692266101136
 21 justy         Rivers (18)       0.40069537878505973
 22 Salem         Dark Wizzy (31)   0.1773985785435086
 23 ProtoBanham   LeoN (34)         0.16800313699748695
 24 WaDi          Kurama (36)       0.16385733683014547
 25 Sinji         Abadango (39)     0.09332436904062955
 26 NAKAT         Tsu (43)          0.0886187065710175
 27 MuteAce       T (46)            -0.03920383159837061
 28 Puppeh        Myran (13)        -0.10122910290848582
 29 Stroder Ame   WaDi (24)         -0.38105905789730204
 30 Lea           MVD (17)          -0.5933360798160604
 31 Dark Wizzy    justy (21)        -0.6794489188666304
 32 Secretary     Marcus (42)       -0.7502527576969438
 33 Ryuga         Stroder Ame (29)  -0.8333533839884627
 34 LeoN          Umeki (41)        -0.8957726364219605
 35 Mr.R          Sinji (25)        -0.9018479934079129
 36 Kurama        Mr. E (37)        -0.9289280746286275
 37 Mr. E         Secretary (32)    -0.9524372210089008
 38 Goblin        MuteAce (27)      -0.9655450862753842
 39 Abadango      Ryuga (33)        -1.0284225901165978
 40 ScAtt         Puppeh (28)       -1.0919322661670152
 41 Umeki         Fatality (47)     -1.4234543291995676
 42 Marcus        ZD (50)           -1.4684599648137353
 43 Tsu           Goblin (38)       -1.4848340496277572
 44 Nietono       Mr.R (35)         -1.5430443617960594
 45 Frozen        Raito (20)        -1.5581680496801975
 46 T             ScAtt (40)        -1.8495472886806612
 47 Fatality      NAKAT (26)        -1.898533984614107
 48 Suarez        Captain L (49)    -2.5031064199451554
 49 Captain L     Suarez (48)       -2.777600672490494
 50 ZD            Frozen (45)       -4.643978182140836

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 176.0966176608784

------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.6

MEAN SQUARED ERROR: 112.68

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         2.7093471827623317
  2 Tweek         Tweek (2)         2.1793122053826517
  3 Marss         Shuton (5)        2.015495110574019
  4 Samsora       ProtoBanham (23)  1.9807168047041785
  5 Shuton        zackray (12)      1.8920083478439607
  6 Ally          Glutonny (14)     1.7849298271797833
  7 Dabuz         Nairo (8)         1.6887341296630836
  8 Nairo         Marss (3)         1.5990685998590703
  9 VoiD          Samsora (4)       1.5943248267345693
 10 Light         Light (10)        1.3995881619618191
 11 Cosmos        Tea (15)          1.3135253381419634
 12 zackray       Nietono (44)      0.9907667678892506
 13 Myran         Dabuz (7)         0.9036846910376795
 14 Glutonny      Kameme (19)       0.7971562025760005
 15 Tea           Ally (6)          0.7634762855267481
 16 ESAM          ESAM (16)         0.597108690943673
 17 MVD           Salem (22)        0.5803550696157151
 18 Rivers        Cosmos (11)       0.5527767897348634
 19 Kameme        Lea (30)          0.4974343909981988
 20 Raito         VoiD (9)          0.4221238736478186
 21 justy         Rivers (18)       0.3367499941298221
 22 Salem         Dark Wizzy (31)   0.15053476203619215
 23 ProtoBanham   LeoN (34)         0.140759484245712
 24 WaDi          Kurama (36)       0.13311651356028034
 25 Sinji         Abadango (39)     0.08951238806027761
 26 NAKAT         Tsu (43)          0.08876753418130802
 27 MuteAce       T (46)            -0.015165744075557347
 28 Puppeh        Myran (13)        -0.08451027849892229
 29 Stroder Ame   WaDi (24)         -0.320216564218067
 30 Lea           MVD (17)          -0.49290769577611593
 31 Dark Wizzy    justy (21)        -0.5714006166742674
 32 Secretary     Marcus (42)       -0.6309010490917052
 33 Ryuga         Stroder Ame (29)  -0.6980333912163975
 34 LeoN          Umeki (41)        -0.7363589558603784
 35 Mr.R          Sinji (25)        -0.7665253797571397
 36 Kurama        Mr. E (37)        -0.7842848163115314
 37 Mr. E         MuteAce (27)      -0.8043750892708826
 38 Goblin        Secretary (32)    -0.8230138043773081
 39 Abadango      Ryuga (33)        -0.877889681693354
 40 ScAtt         Puppeh (28)       -0.9237653240075713
 41 Umeki         Fatality (47)     -1.2034299233027603
 42 Marcus        ZD (50)           -1.234052913529722
 43 Tsu           Goblin (38)       -1.2453904621735983
 44 Nietono       Raito (20)        -1.2881928279526924
 45 Frozen        Mr.R (35)         -1.3098182164976788
 46 T             ScAtt (40)        -1.5744167480783535
 47 Fatality      NAKAT (26)        -1.6133536145446563
 48 Suarez        Captain L (49)    -2.1724031689837004
 49 Captain L     Suarez (48)       -2.335253394725979
 50 ZD            Frozen (45)       -4.002910150250495

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 174.4411280910711

------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.7

MEAN SQUARED ERROR: 112.76

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         2.3133921126873886
  2 Tweek         Tweek (2)         1.860671372543586
  3 Marss         Shuton (5)        1.7238832373812831
  4 Samsora       ProtoBanham (23)  1.693745935631141
  5 Shuton        zackray (12)      1.6191388925399581
  6 Ally          Glutonny (14)     1.5222487162568157
  7 Dabuz         Nairo (8)         1.437618057388941
  8 Nairo         Marss (3)         1.3625864926286964
  9 VoiD          Samsora (4)       1.357230686752678
 10 Light         Light (10)        1.1919053407372655
 11 Cosmos        Tea (15)          1.117845369951226
 12 zackray       Nietono (44)      0.8391837002804756
 13 Myran         Dabuz (7)         0.7638454131233033
 14 Glutonny      Kameme (19)       0.6761622131630924
 15 Tea           Ally (6)          0.6502669300287186
 16 ESAM          ESAM (16)         0.4990672634900424
 17 MVD           Salem (22)        0.4751664674619442
 18 Rivers        Cosmos (11)       0.46023710376450006
 19 Kameme        Lea (30)          0.4179940879151505
 20 Raito         VoiD (9)          0.3500902705902501
 21 justy         Rivers (18)       0.2671595520951485
 22 Salem         Dark Wizzy (31)   0.11572398575296795
 23 ProtoBanham   Kurama (36)       0.1022482937966432
 24 WaDi          LeoN (34)         0.1017226687981238
 25 Sinji         Abadango (39)     0.06751077130534693
 26 NAKAT         Tsu (43)          0.06688851726136744
 27 MuteAce       T (46)            -0.024793152290094444
 28 Puppeh        Myran (13)        -0.08926709955355079
 29 Stroder Ame   WaDi (24)         -0.291867387164128
 30 Lea           MVD (17)          -0.4413943279681102
 31 Dark Wizzy    justy (21)        -0.5122543804822322
 32 Secretary     Marcus (42)       -0.566940553461282
 33 Ryuga         Stroder Ame (29)  -0.6213153258616165
 34 LeoN          Umeki (41)        -0.6520140668893362
 35 Mr.R          Sinji (25)        -0.6892940203274367
 36 Kurama        Mr. E (37)        -0.6984870179084641
 37 Mr. E         MuteAce (27)      -0.7041522406694835
 38 Goblin        Secretary (32)    -0.7379455650317482
 39 Abadango      Ryuga (33)        -0.7953123296620712
 40 ScAtt         Puppeh (28)       -0.8143322771964394
 41 Umeki         Fatality (47)     -1.0544601109829574
 42 Marcus        ZD (50)           -1.079797765821234
 43 Tsu           Goblin (38)       -1.0870693788129968
 44 Nietono       Raito (20)        -1.1234368556967493
 45 Frozen        Mr.R (35)         -1.1487874618667755
 46 T             ScAtt (40)        -1.3783500777721893
 47 Fatality      NAKAT (26)        -1.4127864431095674
 48 Suarez        Captain L (49)    -1.9119900627696464
 49 Captain L     Suarez (48)       -2.0379231472907495
 50 ZD            Frozen (45)       -3.536355105773172

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 174.5259795068224

------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.8

MEAN SQUARED ERROR: 112.56

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         2.0560405835222806
  2 Tweek         Tweek (2)         1.6548532013982953
  3 Marss         Shuton (5)        1.5389181699964622
  4 Samsora       ProtoBanham (23)  1.516307724557869
  5 Shuton        zackray (12)      1.448811283853132
  6 Ally          Glutonny (14)     1.360352052798821
  7 Dabuz         Nairo (8)         1.2860337298968554
  8 Nairo         Marss (3)         1.2209315773021778
  9 VoiD          Samsora (4)       1.2172004355788282
 10 Light         Light (10)        1.0708201433845457
 11 Cosmos        Tea (15)          1.007815397995126
 12 zackray       Nietono (44)      0.766840185318614
 13 Myran         Dabuz (7)         0.6930291178512453
 14 Glutonny      Kameme (19)       0.6199080738499168
 15 Tea           Ally (6)          0.595582163146622
 16 ESAM          ESAM (16)         0.46181568103339676
 17 MVD           Salem (22)        0.4434269147425176
 18 Rivers        Cosmos (11)       0.4284466088006434
 19 Kameme        Lea (30)          0.3983923285616227
 20 Raito         VoiD (9)          0.3352810867376731
 21 justy         Rivers (18)       0.2590892693867286
 22 Salem         Dark Wizzy (31)   0.12375262578321568
 23 ProtoBanham   Kurama (36)       0.11723316500735363
 24 WaDi          LeoN (34)         0.11263031817532655
 25 Sinji         Abadango (39)     0.08859450975869734
 26 NAKAT         Tsu (43)          0.08688663265635177
 27 MuteAce       T (46)            0.009562727404560354
 28 Puppeh        Myran (13)        -0.05331344864486149
 29 Stroder Ame   WaDi (24)         -0.23126161202501275
 30 Lea           MVD (17)          -0.3630450561504546
 31 Dark Wizzy    justy (21)        -0.42398589893750166
 32 Secretary     Marcus (42)       -0.4741289661914474
 33 Ryuga         Stroder Ame (29)  -0.518013845615441
 34 LeoN          Umeki (41)        -0.5437907510522526
 35 Mr.R          Sinji (25)        -0.5808109115934088
 36 Kurama        Mr. E (37)        -0.5900457103541985
 37 Mr. E         MuteAce (27)      -0.5946396781727944
 38 Goblin        Secretary (32)    -0.6237298798154239
 39 Abadango      Puppeh (28)       -0.6912905537548005
 40 ScAtt         Ryuga (33)        -0.6936380245157377
 41 Umeki         Fatality (47)     -0.9048103739592949
 42 Marcus        ZD (50)           -0.9270144521178015
 43 Tsu           Goblin (38)       -0.9312319301682433
 44 Nietono       Raito (20)        -0.9584566191341822
 45 Frozen        Mr.R (35)         -0.9857126524520027
 46 T             ScAtt (40)        -1.1912733158423394
 47 Fatality      NAKAT (26)        -1.2270031309546707
 48 Suarez        Captain L (49)    -1.6854721479040446
 49 Captain L     Suarez (48)       -1.7642038657853292
 50 ZD            Frozen (45)       -3.115868324234071

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 174.16997112151066

-------------------------------------------------------------------------------
--------

MCMC Sampling with 10,000 iterations with 4 chains on a StanModel with scale l
evel: 0.9

MEAN SQUARED ERROR: 112.56

    ACTUAL        PREDICTED         SKILL

  1 MkLeo         MkLeo (1)         1.7938646329402186
  2 Tweek         Tweek (2)         1.437362817643228
  3 Marss         Shuton (5)        1.3329605995806175
  4 Samsora       ProtoBanham (23)  1.313611659061951
  5 Shuton        zackray (12)      1.2515754267548902
  6 Ally          Glutonny (14)     1.1739917883767363
  7 Dabuz         Nairo (8)         1.1117059958110191
  8 Nairo         Marss (3)         1.0494829426913987
  9 VoiD          Samsora (4)       1.0475229632216896
 10 Light         Light (10)        0.9171197375116339
 11 Cosmos        Tea (15)          0.8592275068655377
 12 zackray       Nietono (44)      0.6481049750826796
 13 Myran         Dabuz (7)         0.5830036653172651
 14 Glutonny      Kameme (19)       0.5142162040823848
 15 Tea           Ally (6)          0.4972957357197236
 16 ESAM          ESAM (16)         0.37616486311353875
 17 MVD           Salem (22)        0.3603439845813855
 18 Rivers        Cosmos (11)       0.34789894933381144
 19 Kameme        Lea (30)          0.3157457087822692
 20 Raito         VoiD (9)          0.2606004855605129
 21 justy         Rivers (18)       0.19464097428555857
 22 Salem         Dark Wizzy (31)   0.07619568237488457
 23 ProtoBanham   Kurama (36)       0.06846845402525165
 24 WaDi          LeoN (34)         0.06362089646343001
 25 Sinji         Abadango (39)     0.04253150794611341
 26 NAKAT         Tsu (43)          0.041100661562689024
 27 MuteAce       T (46)            -0.02938749920981263
 28 Puppeh        Myran (13)        -0.08222071173849872
 29 Stroder Ame   WaDi (24)         -0.24396499856319398
 30 Lea            MVD (17)          -0.3569390118323179
 31 Dark Wizzy     justy (21)        -0.41628995773204286
 32 Secretary      Marcus (42)       -0.45985699703395605
 33 Ryuga          Stroder Ame (29)  -0.5021626945268053
 34 LeoN           Umeki (41)        -0.5191115334240114
 35 Mr.R           Sinji (25)        -0.5571071178087226
 36 Kurama         Mr. E (37)        -0.5627589026928002
 37 Mr. E          MuteAce (27)      -0.5662134865522609
 38 Goblin         Secretary (32)    -0.6025361541280349
 39 Abadango       Puppeh (28)       -0.6528768844870569
 40 ScAtt          Ryuga (33)        -0.6564930399237657
 41 Umeki          Fatality (47)     -0.8405727029273806
 42 Marcus         ZD (50)           -0.8641433569535735
 43 Tsu            Goblin (38)       -0.8658456830319671
 44 Nietono        Raito (20)        -0.8911974701807636
 45 Frozen         Mr.R (35)         -0.9157327685936784
 46 T              ScAtt (40)        -1.0958204552312563
 47 Fatality       NAKAT (26)        -1.1322847341614668
 48 Suarez         Captain L (49)    -1.5404455789802571
 49 Captain L      Suarez (48)       -1.6113862454484196
 50 ZD             Frozen (45)       -2.826196499157645

MEAN SQUARED ERROR WITH BIAS FOR CORRECT HIGHER PLACEMENT: 174.16997112151066

-------------------------------------------------------------------------------
--------
```

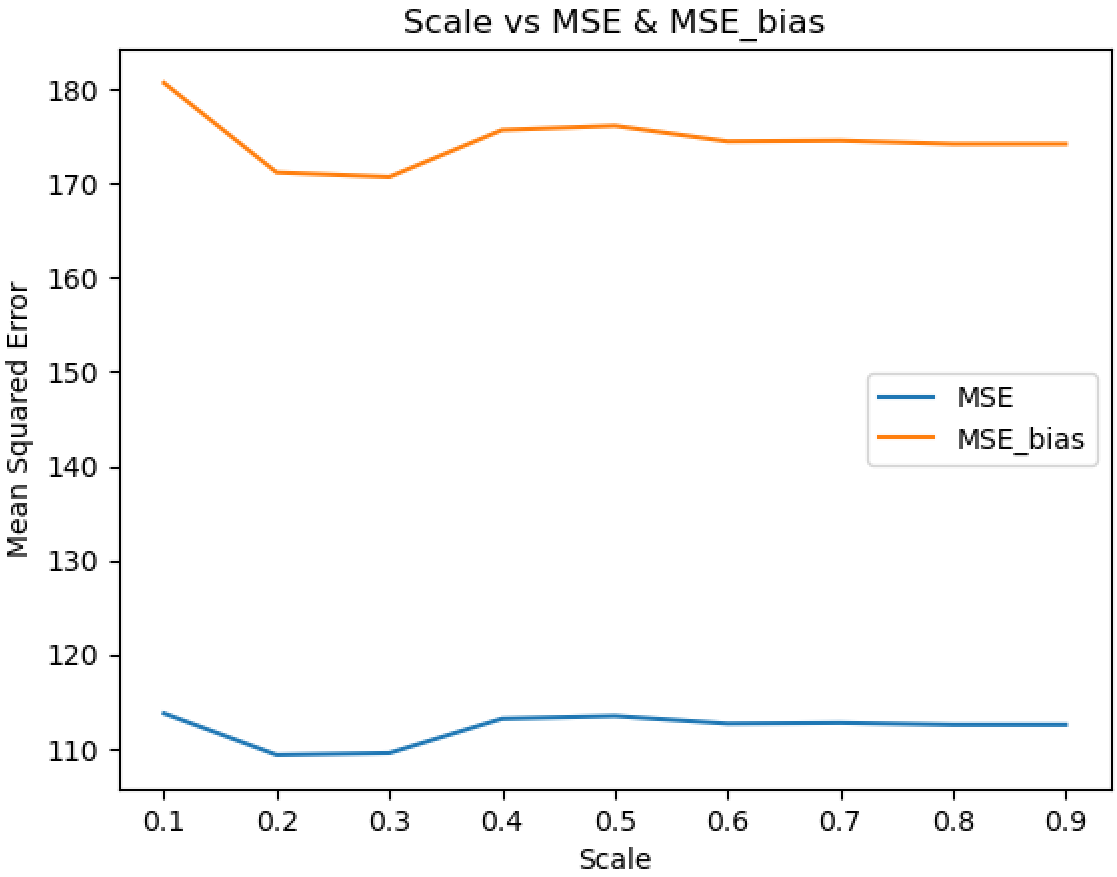
In [20]:

```
printMSEByScale(mse, mse_bias, scales)

plt.plot(scales, mse, label = 'MSE')
plt.plot(scales, mse_bias, label = 'MSE_bias')
plt.xlabel('Scale')
plt.ylabel('Mean Squared Error')
plt.title('Scale vs MSE & MSE_bias')
plt.legend()
```

```
    scale          MSE                MSE_Bias

    0.1            113.76             180.65289712674385
    0.2            109.36             171.1471931049876
    0.3            109.56             170.67117115204704
    0.4            113.2              175.65266193560356
    0.5            113.48             176.0966176608784
    0.6            112.68             174.4411280910711
    0.7            112.76             174.5259795068224
    0.8            112.56             174.16997112151066
    0.9            112.56             174.16997112151066
```

Out[20]:

```
<matplotlib.legend.Legend at 0x7fba14745a60>
```



In [16]:

```
print(f'Best scale: {scales[np.argmin(mse)]} with MSE of {min(mse)}')
print(f'Best scale for more correct higher placements: {scales[np.argmin(mse_b
```

```
Best scale: 0.2 with MSE of 109.36
Best scale for more correct higher placements: 0.3 with MSE of 170.67117115204
704
```