training_overview_1639102435.334922

February 1, 2022

[ ]:

```
import pandas as pd
import os
import matplotlib.pyplot as plt

TIMESTAMP = 1639102435.334922
DATA_DIRECTORY = '/run/media/ture/Backup Plus/data/'

perf_data = pd.read_csv(os.path.join(DATA_DIRECTORY,␣
 ↪f'{TIMESTAMP}_performance_stats.csv'))
perf_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
perf_data.head()
```

[ ]:

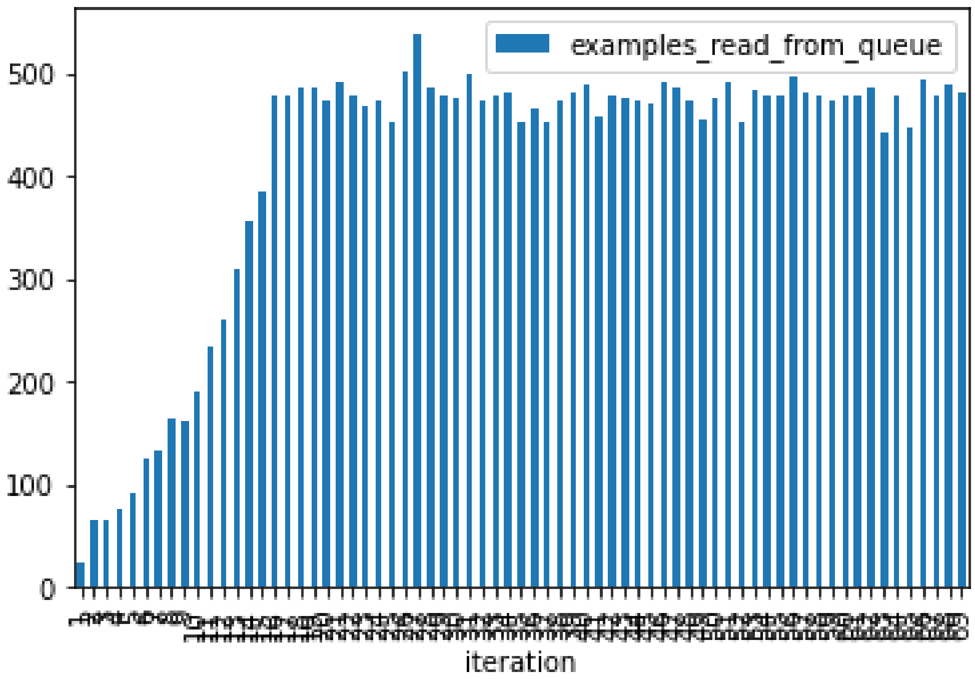
```
   iteration                     timestamp  iteration_duration  \
0          1 1970-01-01 00:00:01.639102511           54.457768
1          2 1970-01-01 00:00:01.639102566           55.257671
2          3 1970-01-01 00:00:01.639102623           57.373703
3          4 1970-01-01 00:00:01.639102687           63.553503
4          5 1970-01-01 00:00:01.639102781           94.168959

   training_duration  examples_read_from_queue  length_experience_buffer
0           5.455627                        23                      1200
1          11.203310                        65                      4792
2          17.952741                        64                      8328
3          27.208010                        76                     12568
4          38.719191                        90                     17488
```

[ ]:

```
perf_data.plot.bar(x='iteration', y='examples_read_from_queue')
```

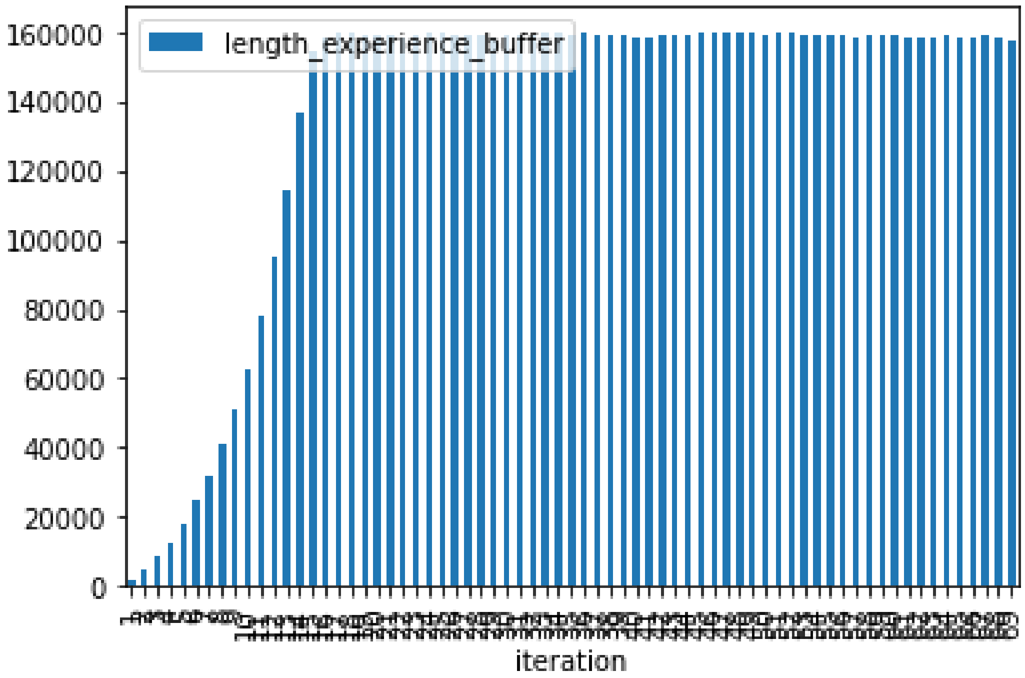
[ ]:

```
<AxesSubplot:xlabel='iteration'>
```



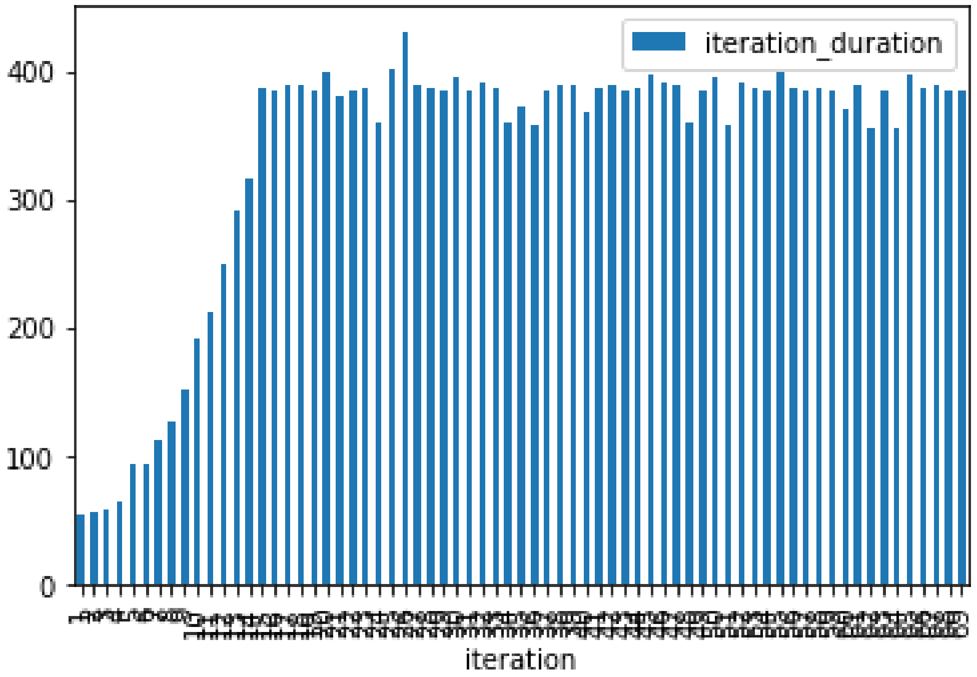
```
[ ]: perf_data.plot.bar(x='iteration', y='length_experience_buffer')
```

```
[ ]: <AxesSubplot:xlabel='iteration'>
```



```
[ ]: perf_data.plot.bar(x='iteration', y='iteration_duration')
```

```
[ ]: <AxesSubplot:xlabel='iteration'>
```



```
[ ]: rndm_data = pd.read_csv(os.path.join(DATA_DIRECTORY,␣
      ↪f'{TIMESTAMP}_random_player_game_stats.csv'))
     rndm_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
     rndm_data.set_index('iteration')
     rndm_data.head()
```

```
[ ]:    iteration                     timestamp  losses  wins  draws  \
     0          1 1970-01-01 00:00:01.639102511       4    32      4
     1          5 1970-01-01 00:00:01.639102566       0    37      3
     2         10 1970-01-01 00:00:01.639102623       0    37      3
     3         15 1970-01-01 00:00:01.639102687       0    32      8
     4         20 1970-01-01 00:00:01.639102781       0    39      1

        nnet_cumul_rewards  random_cumul_rewards
     0                 NaN                   NaN
     1                 NaN                   NaN
     2                 NaN                   NaN
     3                 NaN                   NaN
     4                 NaN                   NaN
```

[ ]:

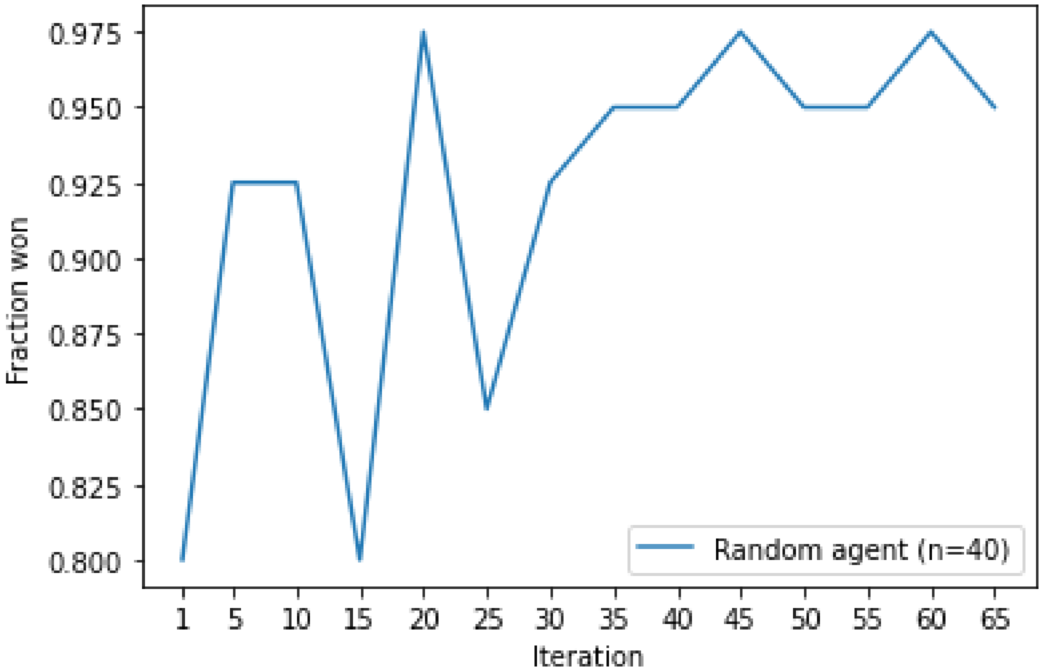
```
hrstc_data = pd.read_csv(os.path.join(DATA_DIRECTORY,␣
 ↪f'{TIMESTAMP}_heuristic_player_game_stats.csv'))
hrstc_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
hrstc_data.set_index('iteration')
hrstc_data.head()
```

[ ]:

```
   iteration                     timestamp  wins  losses  draws  \
0        NaN 1970-01-01 00:00:01.639102511   NaN     NaN    NaN
1        NaN 1970-01-01 00:00:01.639102566   NaN     NaN    NaN
2        NaN 1970-01-01 00:00:01.639102623   NaN     NaN    NaN
3        NaN 1970-01-01 00:00:01.639102687   NaN     NaN    NaN
4        NaN 1970-01-01 00:00:01.639102781   NaN     NaN    NaN

   nnet_cumul_rewards  random_cumul_rewards
0                 NaN                   NaN
1                 NaN                   NaN
2                 NaN                   NaN
3                 NaN                   NaN
4                 NaN                   NaN
```

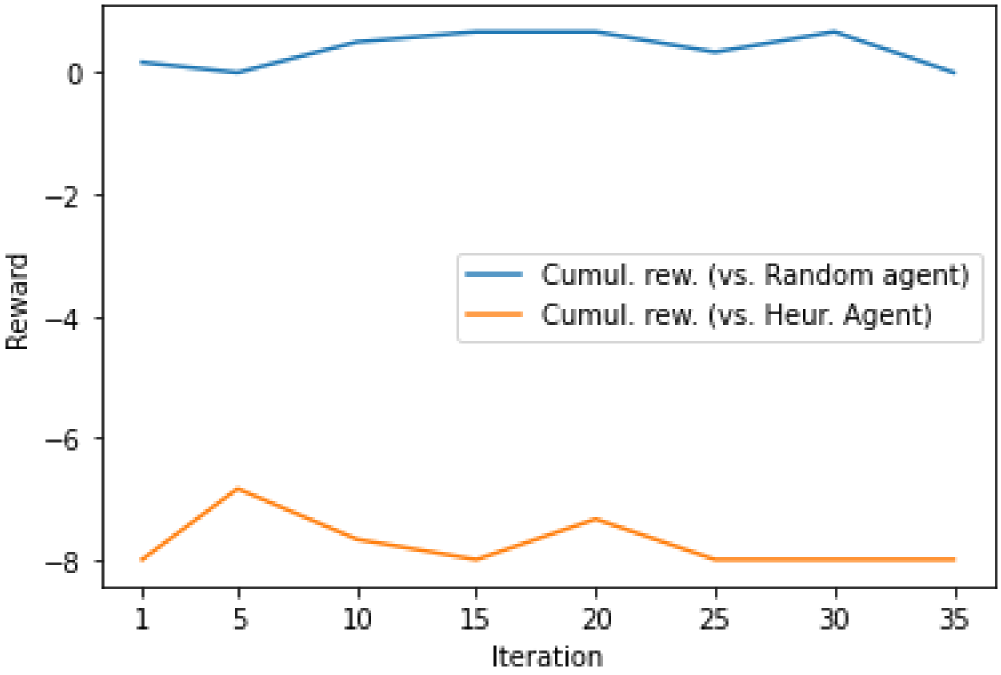
[ ]:

```
rndm_n_games = int(rndm_data['wins'][0] + rndm_data['losses'][0] +␣
 ↪rndm_data['draws'][0])
#hrstc_n_games = int(rndm_data['wins'][0] + rndm_data['losses'][0] +␣
 ↪rndm_data['draws'][0])

rndm_fraction_won = rndm_data.apply(lambda row: row['wins'] / rndm_n_games,␣
 ↪axis=1).to_list()
#hrstc_fraction_won = hrstc_data.apply(lambda row: row['wins'] / hrstc_n_games,␣
 ↪axis=1).to_list()

plt.plot(rndm_data['iteration'], rndm_fraction_won,  label=f'Random agent␣
 ↪(n={rndm_n_games})')
#plt.plot(rndm_data['iteration'], hrstc_fraction_won,  label=f'Random agent␣
 ↪(n={hrstc_n_games})')
plt.xticks(rndm_data['iteration'])
plt.xlabel('Iteration')
plt.ylabel(f'Fraction won')
plt.legend()
plt.show()
```



[ ]:

```
plt.plot(rndm_data['iteration'], rndm_data['nnet_cumul_rewards'], ␣
 ↪label=f'Cumul. rew. (vs. Random agent)')
plt.plot(rndm_data['iteration'], hrstc_data['nnet_cumul_rewards'], ␣
 ↪label=f'Cumul. rew. (vs. Heur. Agent)')
plt.xticks(rndm_data['iteration'])
plt.xlabel('Iteration')
plt.ylabel(f'Reward')
plt.legend()
plt.show()
```


0
−2
−4
−6
−8
Reward
1
5
10
15
20
25
30
35
Iteration
Cumul. rew. (vs. Random agent)
Cumul. rew. (vs. Heur. Agent)