training_overview_1643243086.040512

February 1, 2022

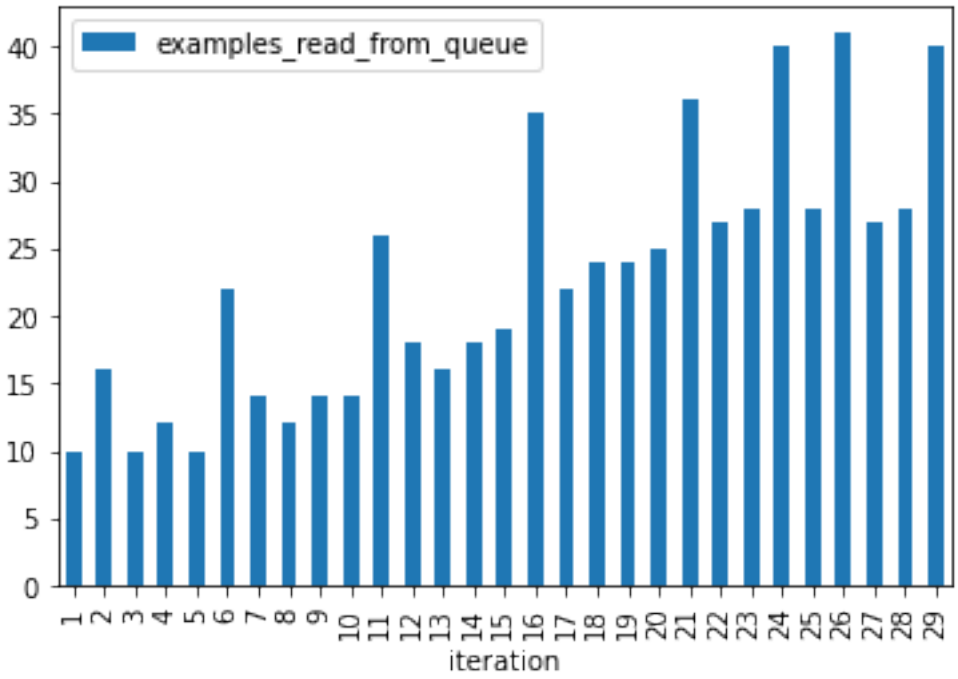
[ ]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt

TIMESTAMP = '1643243086.040512'
DATA_DIRECTORY = '/run/media/ture/Backup Plus/data/2022-01-28_local_training/'

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	training_duration	examples_read_from_queue	length_experience_buffer
0	1	1970-01-01 00:00:01.643245028	1299.558588	48.728450	10	24070
1	2	1970-01-01 00:00:01.643245841	813.311811	127.607025	16	62582
2	3	1970-01-01 00:00:01.643246735	894.111533	174.741285	10	86652
3	4	1970-01-01 00:00:01.643247670	934.921427	233.072750	12	115536
4	5	1970-01-01 00:00:01.643249419	1749.018278	289.855121	10	139606

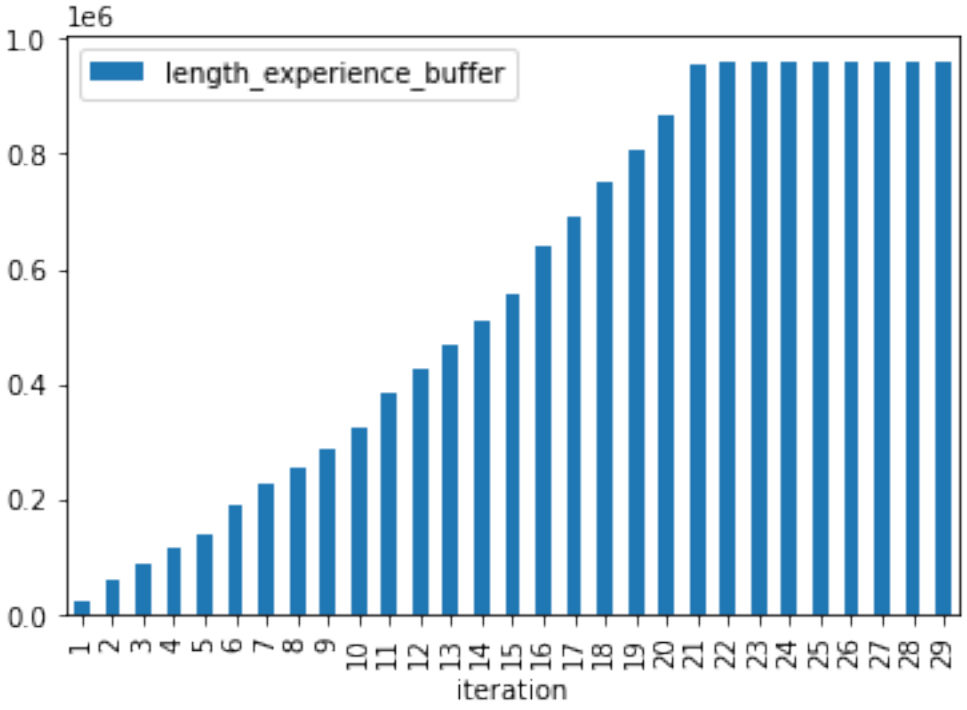
[ ]:
```
perf_data.plot.bar(x='iteration', y='examples_read_from_queue')
```

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



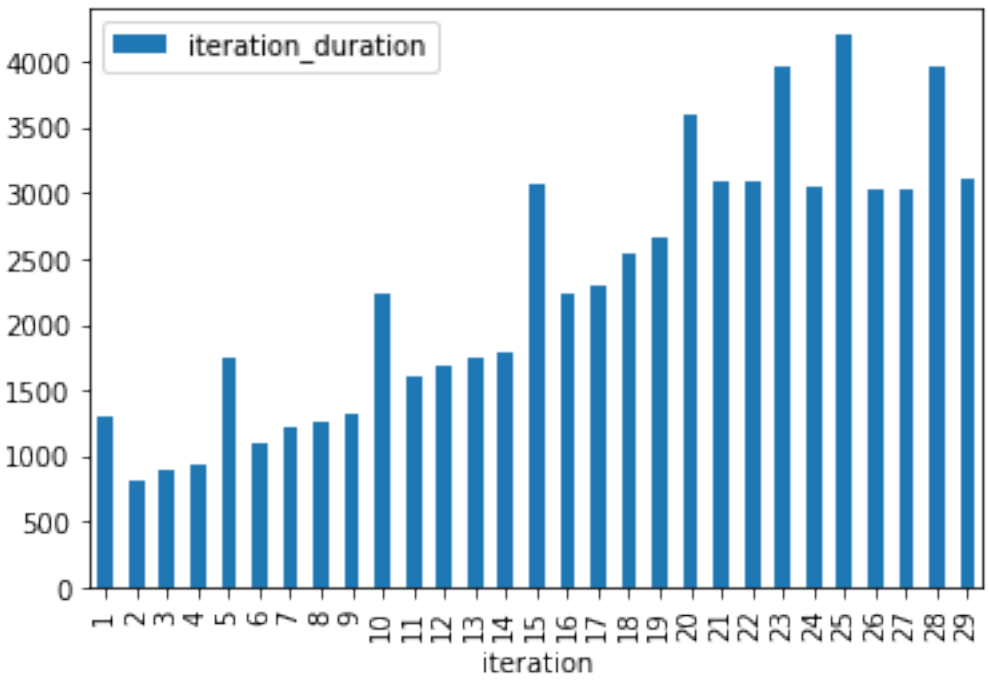
[ ]:
```
perf_data.plot.bar(x='iteration', y='length_experience_buffer')
```

[ ]:
```
<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()
```

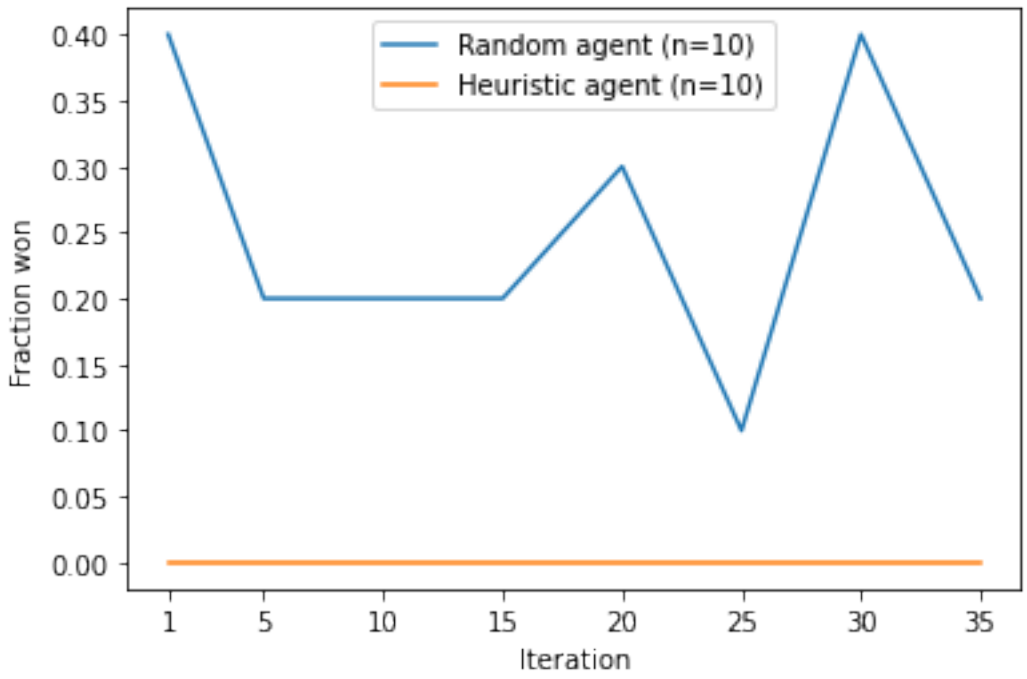
[ ]:
```
   iteration                     timestamp  wins  losses  draws  \
0          1 1970-01-01 00:00:01.643245028   4.0     2.0    4.0
1          5 1970-01-01 00:00:01.643245841   2.0     1.0    7.0
2         10 1970-01-01 00:00:01.643246735   2.0     1.0    7.0
3         15 1970-01-01 00:00:01.643247670   2.0     1.0    7.0
4         20 1970-01-01 00:00:01.643249419   3.0     1.0    6.0

   nnet_cumul_rewards  random_cumul_rewards
0            0.333333             -0.333333
1            0.166667             -0.166667
2            0.333333             -0.333333
3            0.166667             -0.166667
```

```
4            0.833333             -0.833333
```

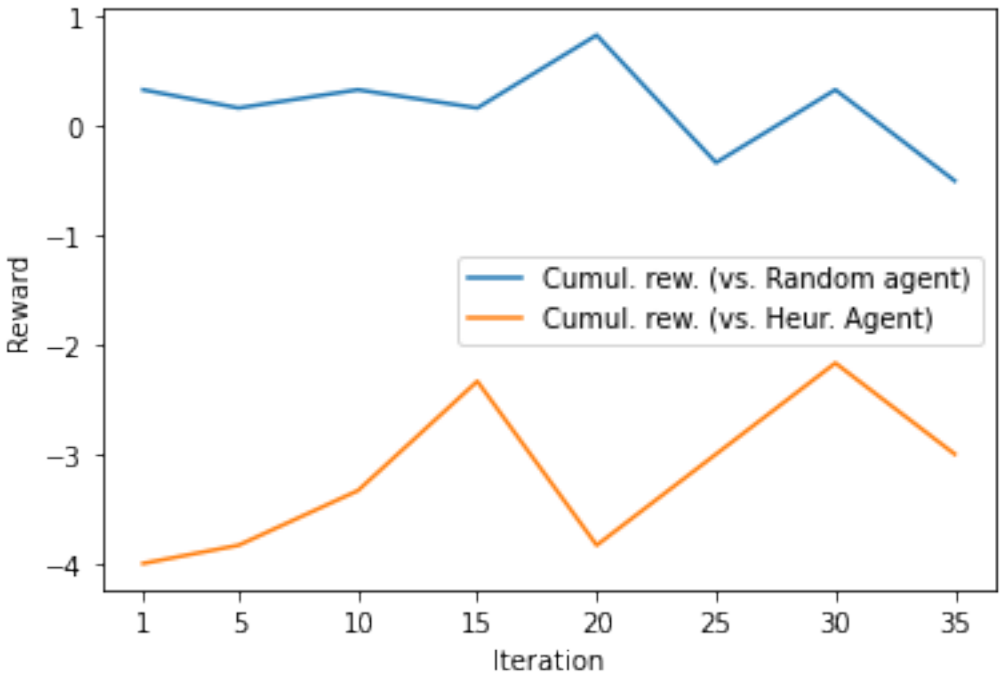
[ ]:
```
hrstc_data = pd.read_csv(os.path.join(DATA_DIRECTORY,␣
↪f'{TIMESTAMP}_heuristic_player_game_stats.csv'))
hrstc_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
hrstc_data.set_index('iteration')
hrstc_data.head()
```

[ ]:
```
   iteration                     timestamp  wins  losses  draws  \
0          1 1970-01-01 00:00:01.643245028   0.0     4.0    0.0
1          5 1970-01-01 00:00:01.643245841   0.0     4.0    0.0
2         10 1970-01-01 00:00:01.643246735   0.0     4.0    0.0
3         15 1970-01-01 00:00:01.643247670   0.0     3.0    1.0
4         20 1970-01-01 00:00:01.643249419   0.0     4.0    0.0

   nnet_cumul_rewards  random_cumul_rewards
0           -4.000000              4.000000
1           -3.833333              3.833333
2           -3.333333              3.333333
3           -2.333333              2.333333
4           -3.833333              3.833333
```

[ ]:
```
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'Heuristic 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()
```


1
0
−1
−2
−3
−4
Reward
1
5
10
15
20
25
30
35
Iteration
Cumul. rew. (vs. Random agent)
Cumul. rew. (vs. Heur. Agent)