training_overview_1643275233.5268605

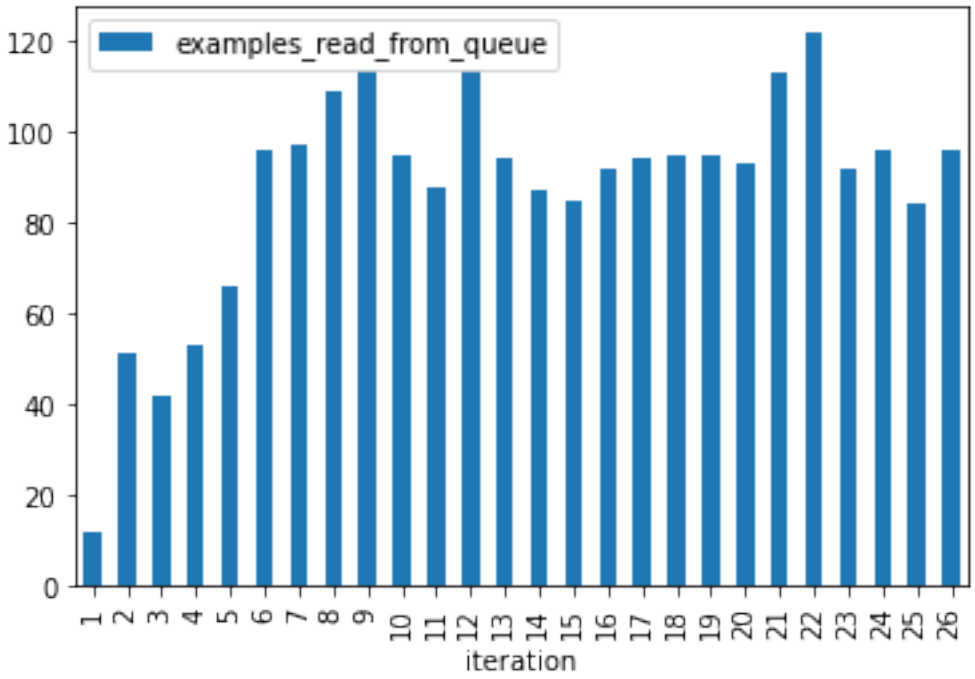
February 1, 2022

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

TIMESTAMP = '1643275233.5268605'
DATA_DIRECTORY = '/run/media/ture/Backup Plus/data/2022-01-28_server_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  \
0          1 1970-01-01 00:00:01.643277282         1515.136067
1          2 1970-01-01 00:00:01.643278455         1173.027582
2          3 1970-01-01 00:00:01.643280007         1552.738886
3          4 1970-01-01 00:00:01.643281921         1913.236352
4          5 1970-01-01 00:00:01.643285460         3539.642873

   training_duration  examples_read_from_queue  length_experience_buffer
0          89.194729                        12                     28884
1         470.165958                        51                    151641
2         781.792642                        42                    252735
3        1174.927254                        53                    380306
4        1665.939276                        66                    539168
```

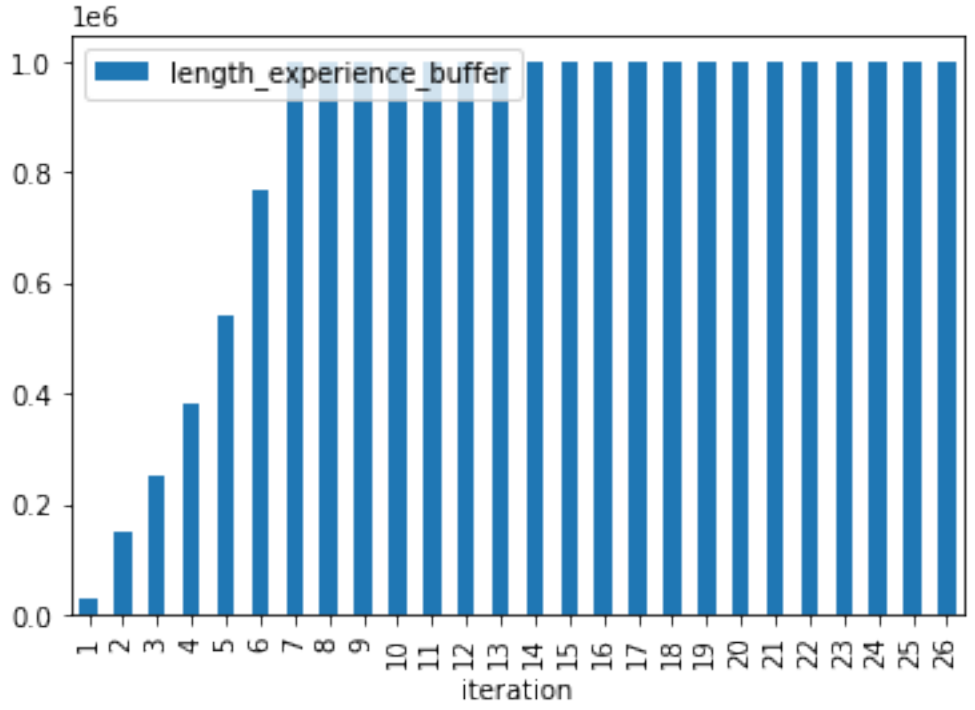
[ ]:
```
perf_data.plot.bar(x='iteration', y='examples_read_from_queue')
```

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



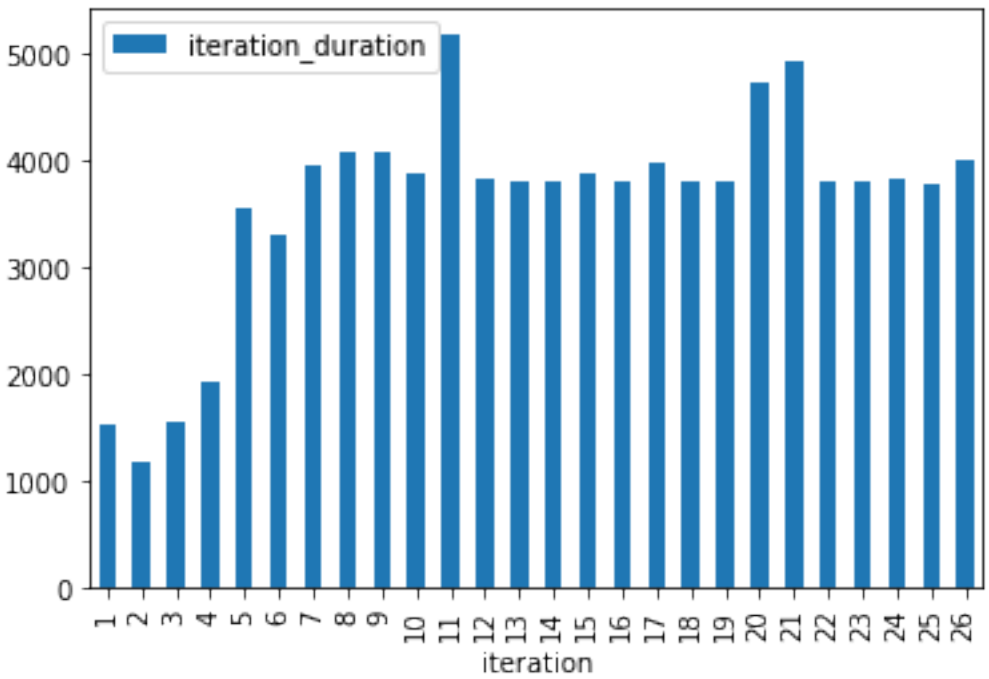
[ ]:
```
perf_data.plot.bar(x='iteration', y='length_experience_buffer')
```

[ ]:
```
<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 00:00:01.643277282	4.0	1.0	11.0	
1	5	1970-01-01 00:00:01.643278455	4.0	1.0	11.0	
2	10	1970-01-01 00:00:01.643280007	3.0	1.0	12.0	
3	15	1970-01-01 00:00:01.643281921	3.0	2.0	11.0	
4	20	1970-01-01 00:00:01.643285460	1.0	2.0	13.0	

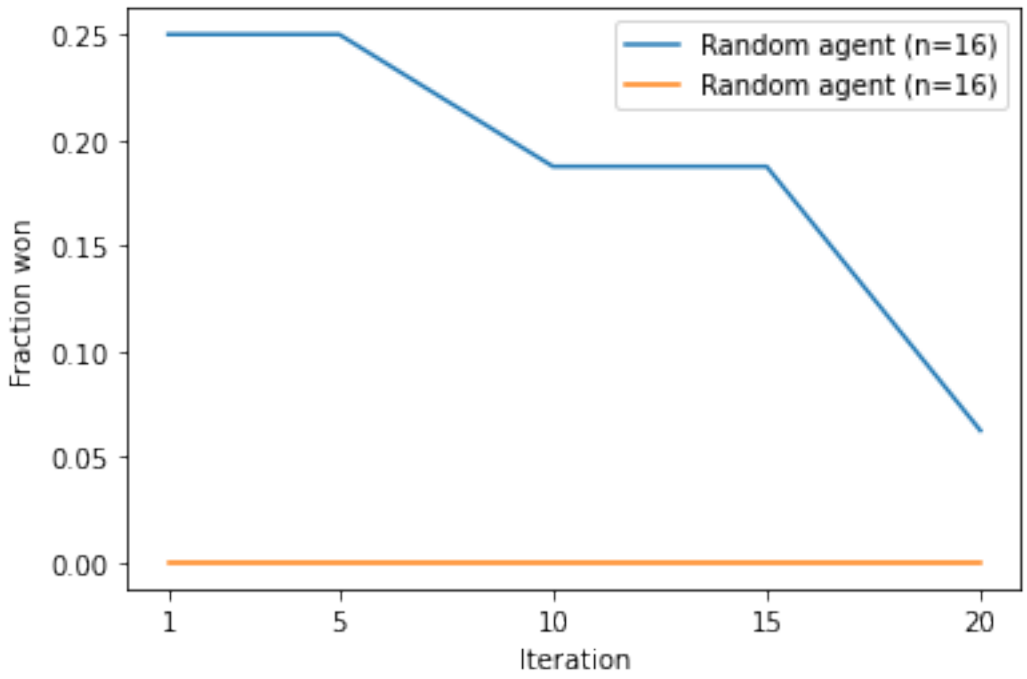
	nnet_cumul_rewards	random_cumul_rewards
0	1.666667	-1.666667
1	0.666667	-0.666667
2	0.333333	-0.333333
3	0.166667	-0.166667

```
4           -0.166667              0.166667
```

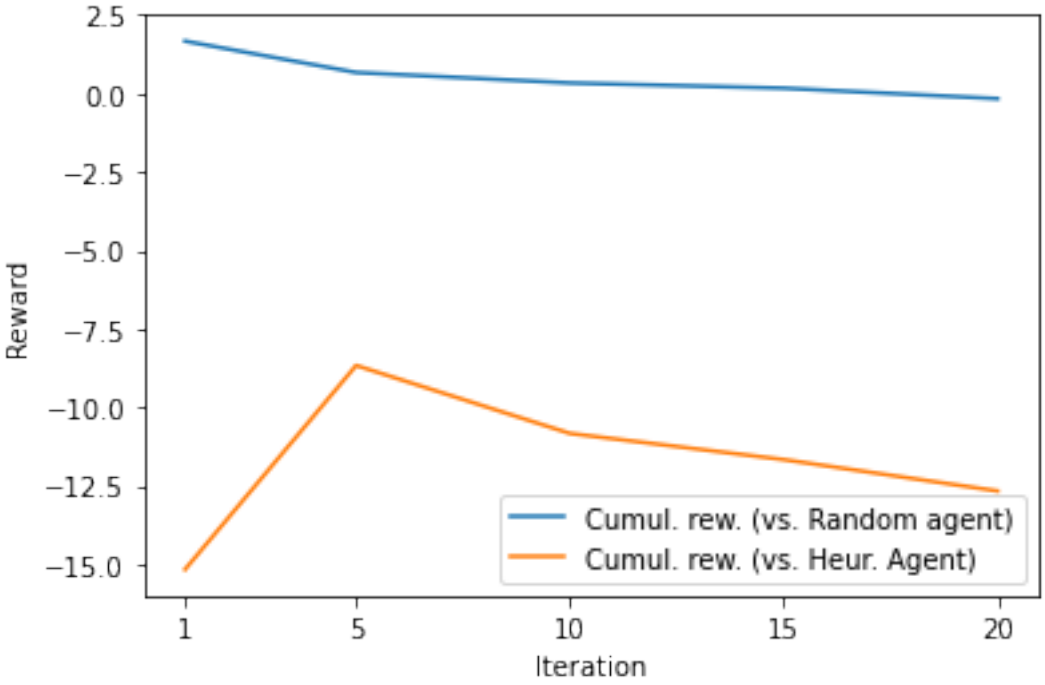
[ ]:
```
hrstc_data = pd.read_csv(os.path.join(DATA_DIRECTORY,␣
↪f'{TIMESTAMP}_heuristic_player_game_stats.csv'))
hrstc_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
hrstc_data.set_index('iteration')
hrstc_data.head()
```

[ ]:
```
   iteration                     timestamp  wins  losses  draws  \
0          1 1970-01-01 00:00:01.643277282   0.0    16.0    0.0
1          5 1970-01-01 00:00:01.643278455   0.0    10.0    6.0
2         10 1970-01-01 00:00:01.643280007   0.0    11.0    5.0
3         15 1970-01-01 00:00:01.643281921   0.0    13.0    3.0
4         20 1970-01-01 00:00:01.643285460   0.0    14.0    2.0

   nnet_cumul_rewards  random_cumul_rewards
0          -15.166667             15.166667
1           -8.666667              8.666667
2          -10.833333             10.833333
3          -11.666667             11.666667
4          -12.666667             12.666667
```

[ ]:
```
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()
```


2.5
0.0
−2.5
−5.0
−7.5
−10.0
−12.5
−15.0
Reward
1
5
10
15
20
Iteration
Cumul. rew. (vs. Random agent)
Cumul. rew. (vs. Heur. Agent)