training_overview_1638355223.9429553

December 4, 2021

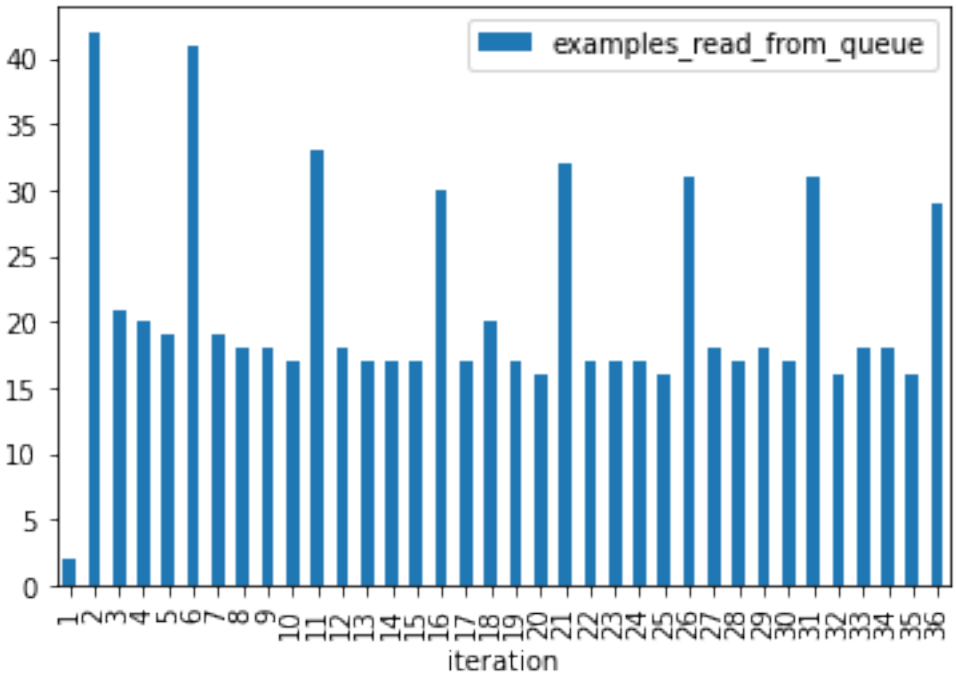
[ ]:
```
import pandas as pd
import os
import matplotlib.pyplot as plt

TIMESTAMP = 1638355223.9429553
DATA_DIRECTORY = '../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	training_duration	examples_read_from_queue	length_train_examples
0	1	1970-01-01 00:00:01.638359832	3766.003021	3.169783	2	402
1	2	1970-01-01 00:00:01.638362029	2197.623581	21.752734	42	2402
2	3	1970-01-01 00:00:01.638364174	2144.967788	36.442115	21	4402
3	4	1970-01-01 00:00:01.638366413	2238.904411	55.739818	20	6402
4	5	1970-01-01 00:00:01.638370947	4533.327736	73.305841	19	8402

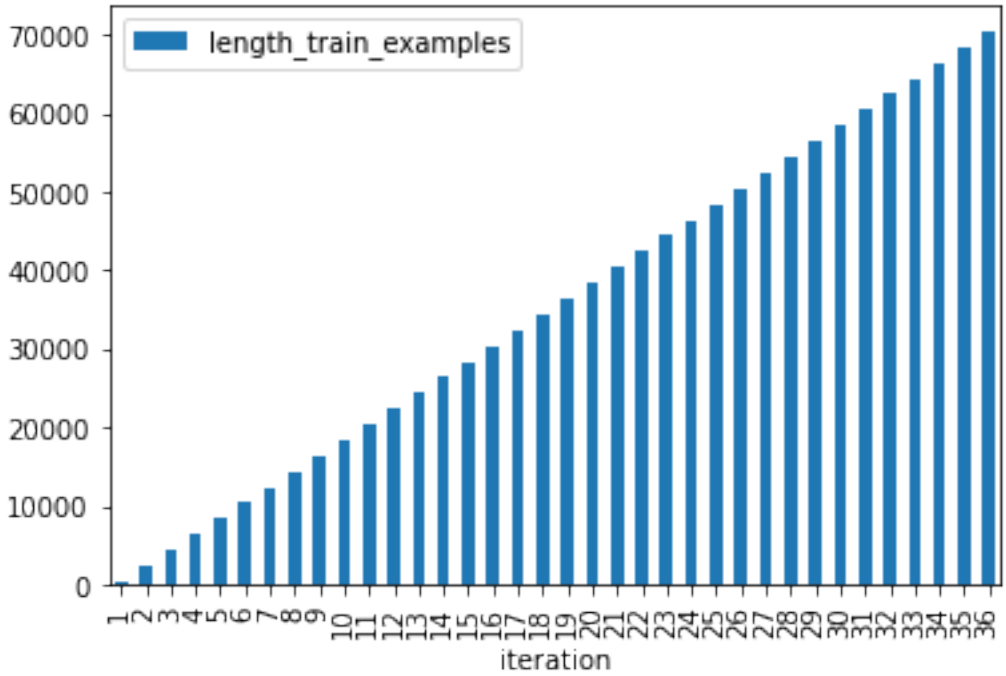
[ ]:
```
perf_data.plot.bar(x='iteration', y='examples_read_from_queue')
```

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



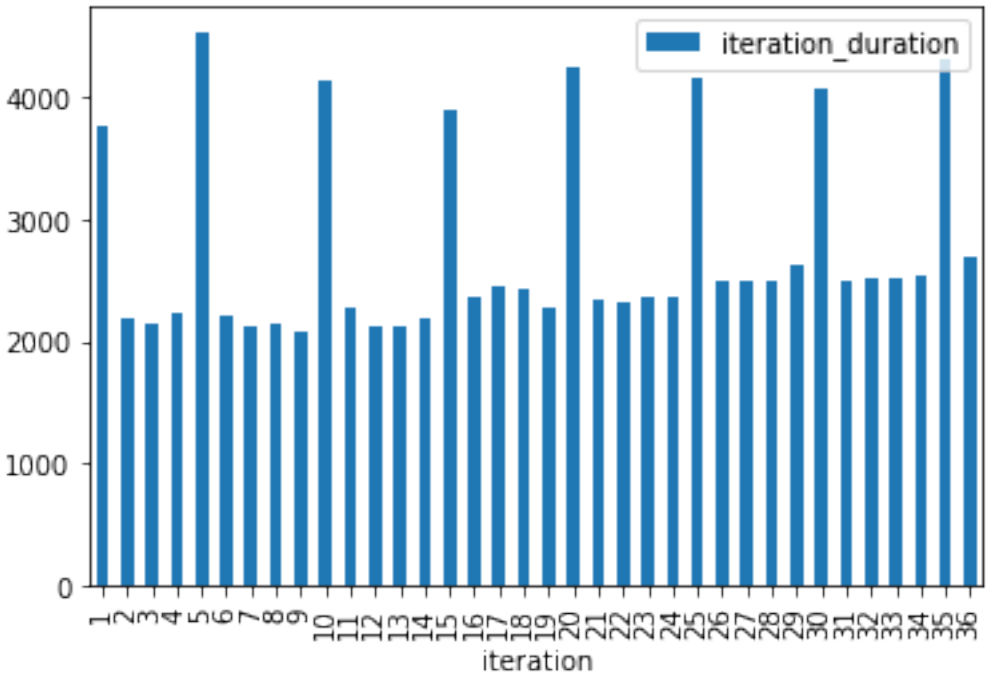
[ ]:
```
perf_data.plot.bar(x='iteration', y='length_train_examples')
```

[ ]:
```
<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 00:00:01.638359832	3.0	3.0	2.0	
1	5	1970-01-01 00:00:01.638362029	2.0	2.0	4.0	
2	10	1970-01-01 00:00:01.638364174	3.0	1.0	4.0	
3	15	1970-01-01 00:00:01.638366413	3.0	0.0	5.0	
4	20	1970-01-01 00:00:01.638370947	4.0	1.0	3.0	

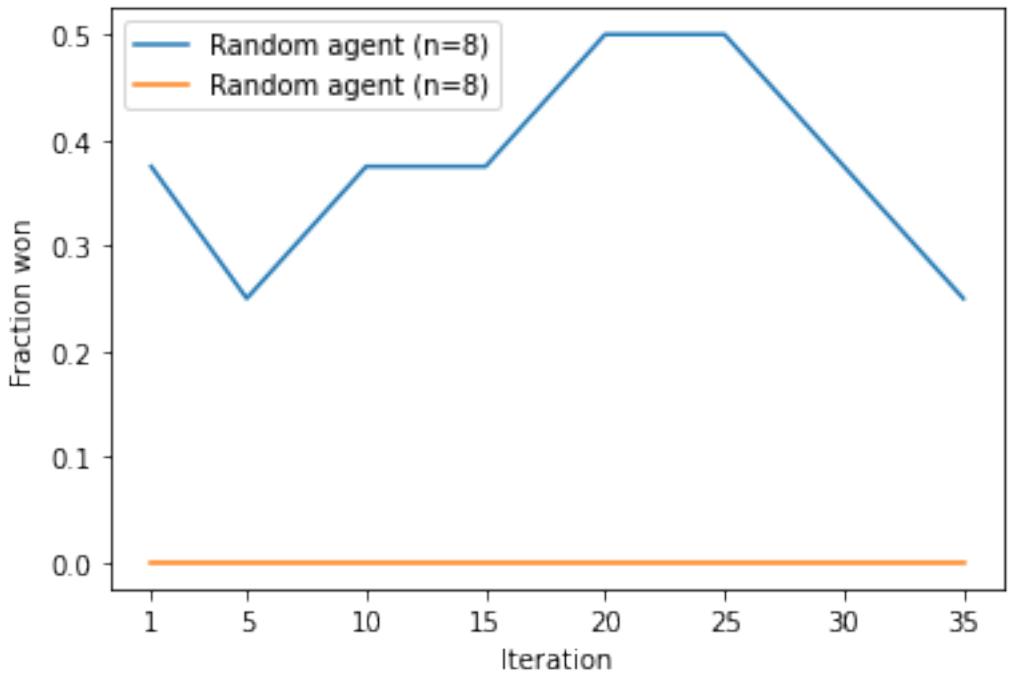
	nnet_cumul_rewards	random_cumul_rewards
0	0.166667	-0.166667
1	0.000000	0.000000
2	0.500000	-0.500000
3	0.666667	-0.666667
4	0.666667	-0.666667

[ ]:
```
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()
```

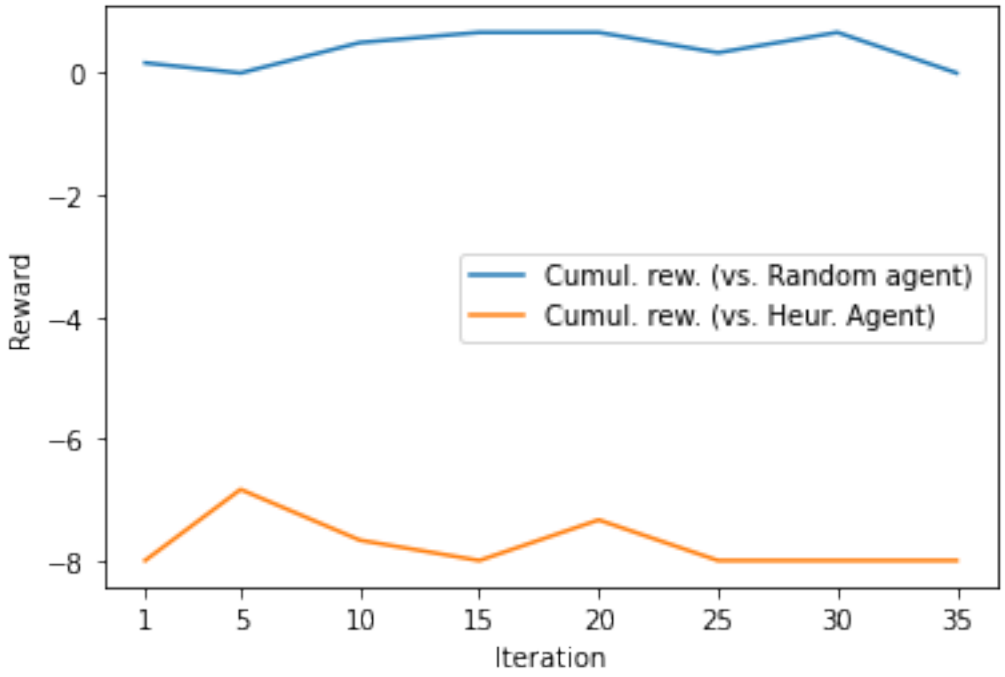
[ ]:

	iteration	timestamp	wins	losses	draws	nnet_cumul_rewards	random_cumul_rewards
0	1	1970-01-01 00:00:01.638359832	0.0	8.0	0.0	-8.000000	8.000000
1	5	1970-01-01 00:00:01.638362029	0.0	7.0	1.0	-6.833333	6.833333
2	10	1970-01-01 00:00:01.638364174	0.0	8.0	0.0	-7.666667	7.666667
3	15	1970-01-01 00:00:01.638366413	0.0	8.0	0.0	-8.000000	8.000000
4	20	1970-01-01 00:00:01.638370947	0.0	8.0	0.0	-7.333333	7.333333

[ ]:
```
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)