reinforce_visualisation

May 14, 2021

[1]:

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

RESULTS_DIR = '..\\..\\results\\REINFORCE-evaluation\\'
CSV_FILENAME = 'REINFORCE-evaluation.csv'

def load_data(path, file):
    csv_path = os.path.join(path, file)
    return pd.read_csv(csv_path, sep=',')
```

[2]:

```
rl_eval = load_data(RESULTS_DIR, CSV_FILENAME)
rl_eval
```

[2]:

```
    eps  illegal_count  east_win_rate  south_win_rate  west_win_rate  \
0    10              0       0.184000        0.234000       0.156000
1    20              4       0.487903        0.504032       0.497984
2    30              4       0.560484        0.514113       0.546371
3    40              3       0.599598        0.531187       0.549296
4    50              2       0.636546        0.580321       0.588353
5    60              6       0.635628        0.607287       0.613360
6    70              3       0.629779        0.621730       0.623742
7    80              6       0.680162        0.623482       0.621457
8    90              3       0.694165        0.639839       0.617706
9   100              4       0.707661        0.639113       0.625000
10  110              4       0.655242        0.590726       0.614919
11  120              2       0.624498        0.564257       0.590361
12  130              4       0.679435        0.643145       0.598790
13  140              4       0.653226        0.641129       0.631048
14  150              6       0.625506        0.637652       0.631579
15  160              6       0.647773        0.651822       0.641700
16  170              5       0.650505        0.658586       0.668687
17  180              2       0.738956        0.664659       0.710843
18  190              4       0.713710        0.693548       0.733871
19  200              5       0.634343        0.628283       0.595960
20  210              3       0.607646        0.627767       0.601610
21  220              6       0.670040        0.690283       0.645749
22  230              5       0.612121        0.632323       0.644444
```

```
23  240                0        0.658000        0.662000        0.664000
24  250                3        0.668008        0.659960        0.682093
25  260                4        0.693548        0.655242        0.651210
26  270                4        0.697581        0.689516        0.677419
27  280                2        0.720884        0.666667        0.692771
28  290                3        0.750503        0.684105        0.714286
29  300                3        0.746479        0.714286        0.744467
30  310                3        0.768612        0.710262        0.754527
31  320                2        0.744980        0.706827        0.722892
32  330                1        0.729459        0.705411        0.713427
33  340                1        0.733467        0.709419        0.703407
34  350                4        0.752016        0.701613        0.733871
35  360                5        0.767677        0.733333        0.753535
36  370                6        0.751012        0.720648        0.759109
37  380                4        0.653226        0.606855        0.657258
38  390                6        0.619433        0.647773        0.657895
39  400                5        0.688889        0.658586        0.666667

    total_win_rate  east_loss_rate  south_loss_rate  west_loss_rate  \
0         0.191333        0.002000         0.006000        0.006000
1         0.496640        0.040323         0.030242        0.016129
2         0.540323        0.044355         0.044355        0.022177
3         0.560027        0.052314         0.038229        0.024145
4         0.601740        0.044177         0.040161        0.026104
5         0.618758        0.042510         0.040486        0.024291
6         0.625084        0.052314         0.054326        0.038229
7         0.641700        0.040486         0.046559        0.052632
8         0.650570        0.046278         0.054326        0.058350
9         0.657258        0.046371         0.058468        0.060484
10        0.620296        0.042339         0.044355        0.036290
11        0.593039        0.024096         0.034137        0.026104
12        0.640457        0.048387         0.056452        0.054435
13        0.641801        0.042339         0.052419        0.052419
14        0.631579        0.050607         0.044534        0.046559
15        0.647099        0.052632         0.048583        0.036437
16        0.659259        0.040404         0.038384        0.032323
17        0.704819        0.038153         0.032129        0.020080
18        0.713710        0.034274         0.032258        0.024194
19        0.619529        0.022222         0.024242        0.024242
20        0.612341        0.022133         0.026157        0.032193
21        0.668691        0.012146         0.010121        0.032389
22        0.629630        0.024242         0.022222        0.028283
23        0.661333        0.022000         0.028000        0.022000
24        0.670020        0.032193         0.034205        0.030181
25        0.666667        0.022177         0.034274        0.038306
26        0.688172        0.012097         0.026210        0.028226
27        0.693440        0.018072         0.042169        0.022088
```

```
28        0.716298         0.028169          0.042254         0.040241
29        0.735077         0.034205          0.038229         0.026157
30        0.744467         0.026157          0.040241         0.026157
31        0.724900         0.022088          0.042169         0.028112
32        0.716099         0.030060          0.044088         0.028056
33        0.715431         0.026052          0.044088         0.034068
34        0.729167         0.028226          0.036290         0.028226
35        0.751515         0.032323          0.036364         0.022222
36        0.743590         0.030364          0.032389         0.036437
37        0.639113         0.028226          0.030242         0.036290
38        0.641700         0.016194          0.038462         0.024291
39        0.671380         0.018182          0.034343         0.036364

    total_loss_rate  east_avg_score  south_avg_score  west_avg_score  \
0          0.004667    10446.236559      7377.500000     6495.061728
1          0.028898     9362.977099      6906.037736     7170.588235
2          0.036962     9937.000000      6400.361011     6829.078014
3          0.038229     9899.382716      6356.890459     7008.771930
4          0.036814    10244.247788      6621.035599     6980.718954
5          0.035762     9874.626866      6633.437500     6780.000000
6          0.048290     9797.640118      6609.523810     6793.009119
7          0.046559    10238.483146      6725.679758     6617.117117
8          0.052985    10263.315217      6731.884058     6563.690476
9          0.055108     9990.106952      6488.728324     6443.235294
10         0.040995    10292.774566      6586.666667     6641.486068
11         0.028112    10547.678019      6420.805369     6544.299674
12         0.053091    10361.218837      6642.651297     6491.975309
13         0.049059    10101.739130      6415.406977     6596.165192
14         0.047233     9904.191617      6560.830861     6512.537313
15         0.045884     9566.184971      6802.890173     6537.313433
16         0.037037     9665.204678      6911.304348     6611.239193
17         0.030120     9932.558140      6791.354467     6544.230769
18         0.030242    10085.444744      6563.888889     6536.170213
19         0.023569    10146.153846      6831.269350     6696.742671
20         0.026828    10530.670927      7330.153846     6433.015873
21         0.018219    10175.074184      7375.144509     6471.940299
22         0.024916    10244.444444      7216.358025     6569.069069
23         0.024000    10148.529412      7066.666667     6632.653061
24         0.032193    10122.413793      6703.768116     6580.790960
25         0.031586    10454.084507      6652.046784     6587.426901
26         0.022177    10286.363636      6761.408451     6361.714286
27         0.027443    10273.097826      6327.195467     6682.865169
28         0.036888    10367.700258      6615.235457     6399.200000
29         0.032864     9969.587629      6755.882353     6650.130548
30         0.030852    10794.936709      6789.276139     6561.855670
31         0.030790    10404.450262      6816.085791     6664.438503
32         0.034068    10421.899736      6944.117647     6560.810811
```

```
33        0.034736   10511.609499     6783.776596     6491.576087
34        0.030914   10263.307494     6589.617486     6503.968254
35        0.030303   10424.747475     6609.973753     6729.947917
36        0.033063   10200.518135     6711.290323     6762.849873
37        0.031586   10069.230769     6961.075949     6558.430233
38        0.026316   10545.222930     6883.480826     6556.379822
39        0.029630   10498.285714     6744.897959     6501.149425

    total_avg_score
0       8105.102041
1       7815.473146
2       7776.251455
3       7851.905830
4       8023.899371
5       7800.412371
6       7746.115538
7       7916.274510
8       7916.873213
9       7709.528302
10      7907.825203
11      7898.060345
12      7896.124031
13      7712.159533
14      7654.771372
15      7647.224927
16      7721.470019
17      7816.575592
18      7734.688347
19      7916.125654
20      8084.784890
21      8004.813360
22      7975.925926
23      7941.147860
24      7798.471824
25      7929.836381
26      7802.932829
27      7793.036212
28      7836.242208
29      7809.519651
30      8081.660900
31      7979.982285
32      7991.540516
33      7946.126447
34      7818.037135
35      7950.818260
36      7899.044309
37      7874.949900
```

```
38      7933.535354
39      7925.360231
```

[3]:
```
max(rl_eval['total_win_rate'])
```

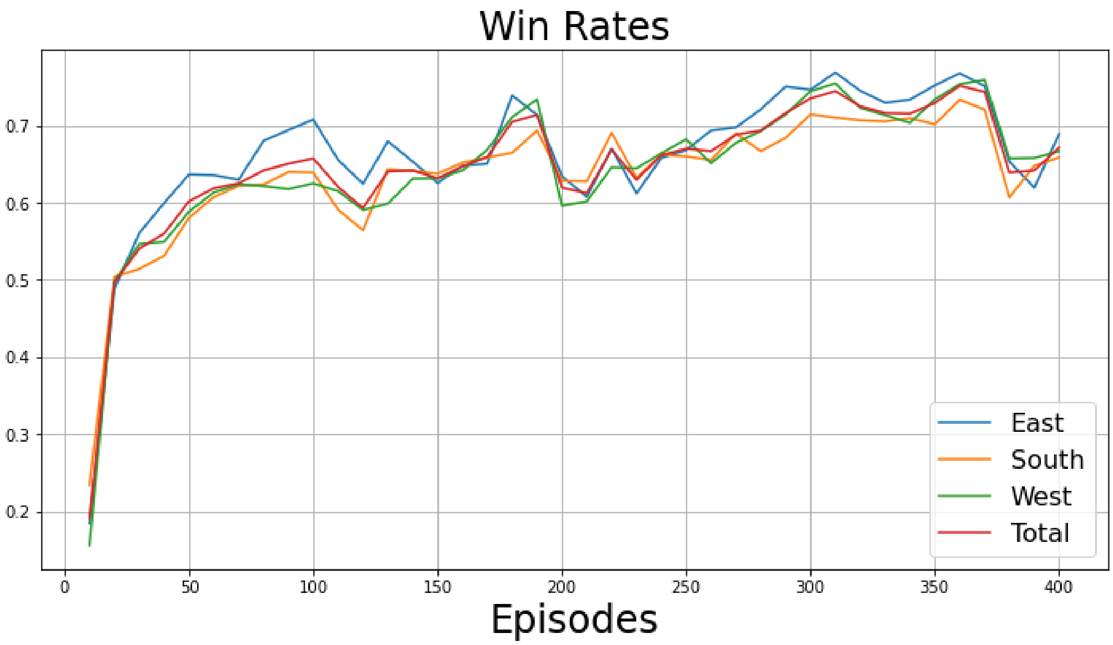
[3]:
```
0.7515151515151515
```

[4]:
```
plt.figure(figsize=(12, 6))
plt.grid()

plt.plot(rl_eval['eps'], rl_eval['east_win_rate'], label='East')
plt.plot(rl_eval['eps'], rl_eval['south_win_rate'], label='South')
plt.plot(rl_eval['eps'], rl_eval['west_win_rate'], label='West')
plt.plot(rl_eval['eps'], rl_eval['total_win_rate'], label='Total')

plt.title('Win Rates', fontsize=24)
plt.xlabel('Episodes', fontsize=24)
plt.legend(fontsize=16, loc='lower right')

plt.savefig("../../../../Dissertation/figs/rl-win-rates.png",␣
 ↪bbox_inches='tight')
plt.show()
```



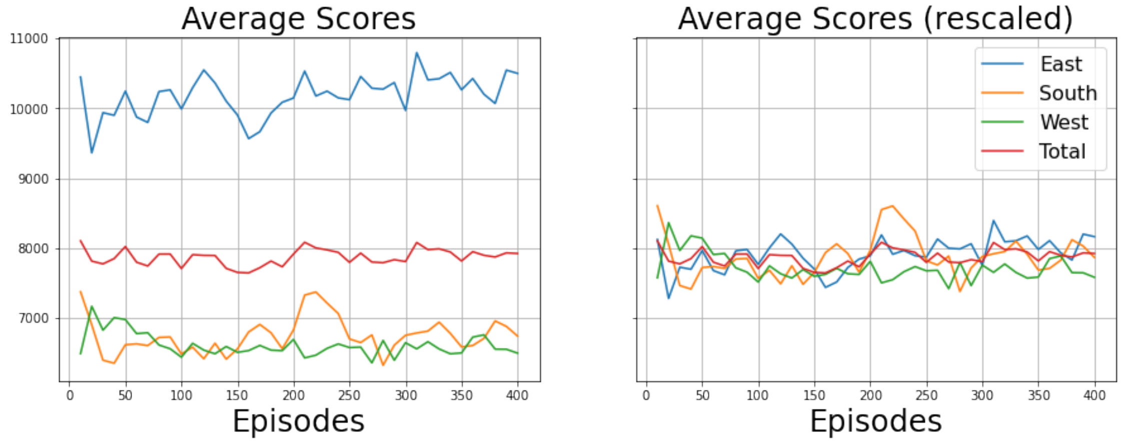
[5]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 5), sharey='row')

ax1.grid()
```

```
ax1.plot(rl_eval['eps'], rl_eval['east_avg_score'], label='East')
ax1.plot(rl_eval['eps'], rl_eval['south_avg_score'], label='South')
ax1.plot(rl_eval['eps'], rl_eval['west_avg_score'], label='West')
ax1.plot(rl_eval['eps'], rl_eval['total_avg_score'], label='Total')
ax1.set_title('Average Scores', fontsize=24)
ax1.set_xlabel('Episodes', fontsize=24)

ax2.grid()
ax2.plot(rl_eval['eps'], rl_eval['east_avg_score'] * (3.5 / 3) / 1.5,␣
 ↪label='East')
ax2.plot(rl_eval['eps'], rl_eval['south_avg_score'] * (3.5 / 3), label='South')
ax2.plot(rl_eval['eps'], rl_eval['west_avg_score'] * (3.5 / 3), label='West')
ax2.plot(rl_eval['eps'], rl_eval['total_avg_score'], label='Total')
ax2.set_title('Average Scores (rescaled)', fontsize=24)
ax2.set_xlabel('Episodes', fontsize=24)
ax2.legend(fontsize=16)

plt.savefig("../../../../Dissertation/figs/rl-avg-scores.png",␣
 ↪bbox_inches='tight')
plt.show()
```

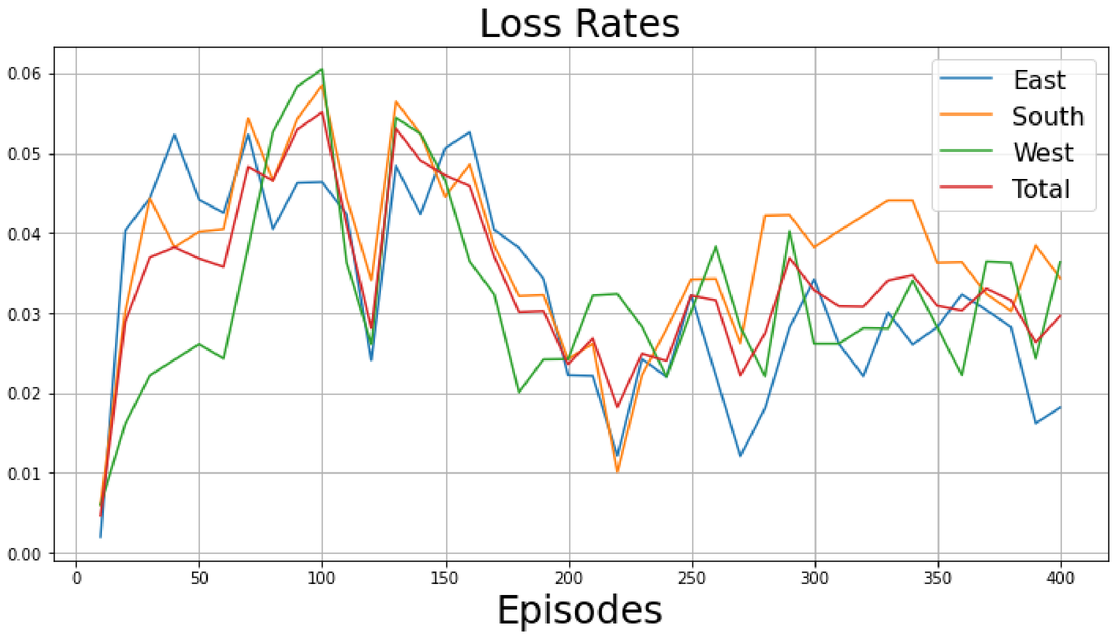


[6]:
```
plt.figure(figsize=(12, 6))
plt.grid()

plt.plot(rl_eval['eps'], rl_eval['east_loss_rate'], label='East')
plt.plot(rl_eval['eps'], rl_eval['south_loss_rate'], label='South')
plt.plot(rl_eval['eps'], rl_eval['west_loss_rate'], label='West')
plt.plot(rl_eval['eps'], rl_eval['total_loss_rate'], label='Total')

plt.title('Loss Rates', fontsize=24)
plt.xlabel('Episodes', fontsize=24)
plt.legend(fontsize=16)

plt.savefig("../../../../Dissertation/figs/rl-loss-rates.png",␣
 ↪bbox_inches='tight')
plt.show()
```

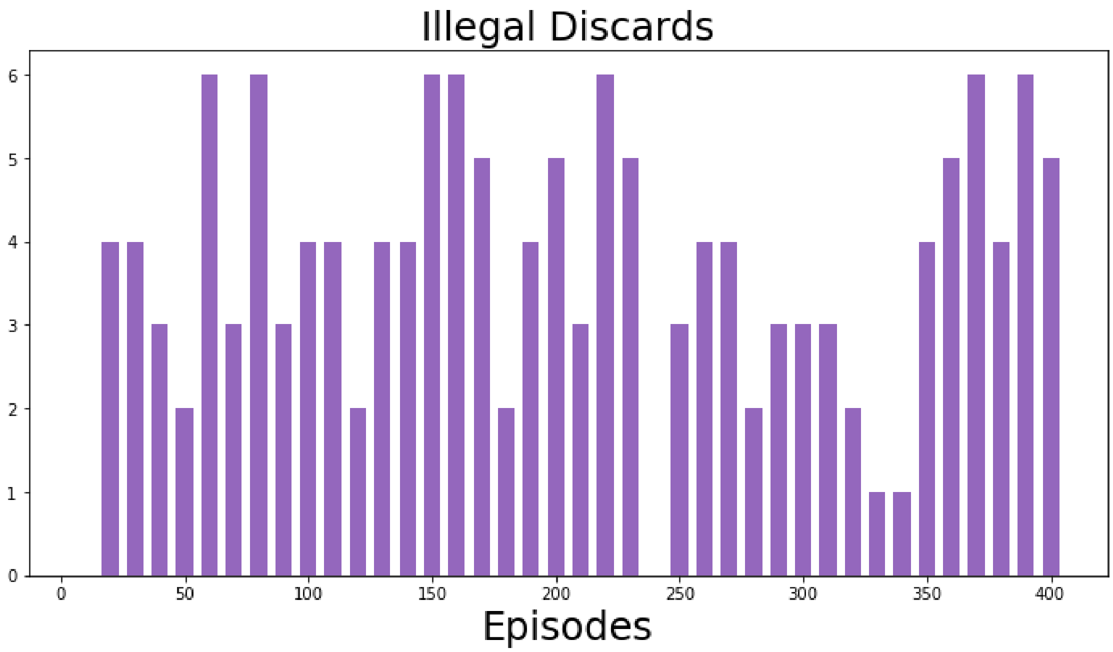


[7]:
```
plt.figure(figsize=(12, 6))

plt.bar(rl_eval['eps'], rl_eval['illegal_count'], width=6.6667, color='tab:
 ↪purple')

plt.title('Illegal Discards', fontsize=24)
plt.xlabel('Episodes', fontsize=24)

plt.savefig("../../../../Dissertation/figs/rl-illegal-discards.png",␣
 ↪bbox_inches='tight')
plt.show()
```


Illegal Discards
6
5
4
3
2
1
0
0
50
100
150
200
250
300
350
400
Episodes