COMSM0140: Internet Economics and Financial Technology 2023. Main coursework.

Vasudev Menon - 2060697

This report compares trading strategies ZIC, SHVR, GVWY, ZIP, ZIPSH against each other in various market conditions. It is detailed with multiple visualisations and statistical comparisions to support the arguments made.

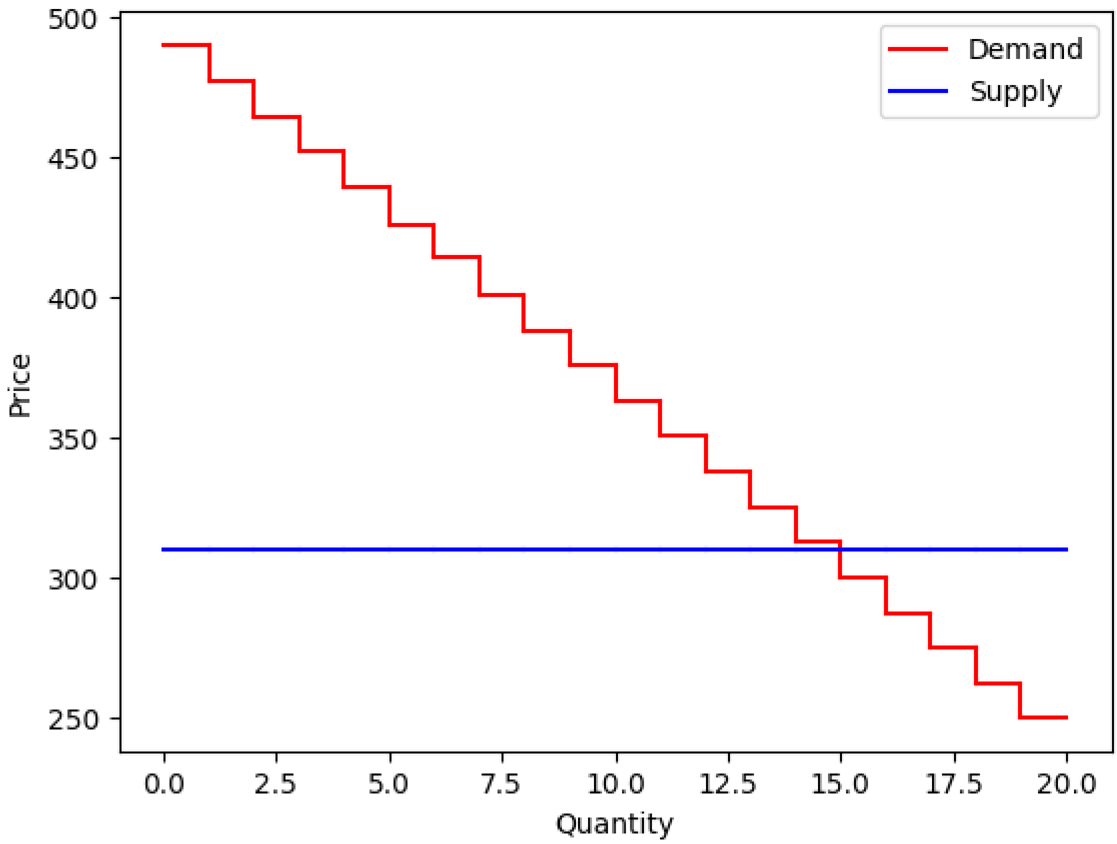
In [ ]:
```
#PLEASE RUN
from BSE import *
from helper import *
```

In [ ]:
```
#Market conditions

start_time = 0
end_time = 60 * 60

#PART A - B
#Code changed to output data for 50 trials at R = 10, 20, 30, 40, 50, 60,
R = 70
num = 20
num_SHVR = int(R/100 * num)
num_ZIC = num - num_SHVR

sellers_spec = [('ZIC', num_ZIC ),('SHVR', num_SHVR)]
buyers_spec = sellers_spec
traders_spec = {'sellers':sellers_spec, 'buyers':buyers_spec}

sup_range = (310,310)
dem_range = (250,490)

dump_all = True
verbose = False

supply_schedule = [{'from': start_time, 'to': end_time, 'ranges': [sup_ra
demand_schedule = [{'from': start_time, 'to': end_time, 'ranges': [dem_ra

order_interval = 15
order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
               'interval': order_interval, 'timemode': 'periodic'}

dump_flags = {'dump_blotters': True, 'dump_lobs': True, 'dump_strats': Tr
              'dump_avgbals': True, 'dump_tape': True}
```

PART A

Here we are comparing SHVR vs ZIC at equal ratios. Below is the supply-demand curve for the market conditions defined by the coursework brief. The supply in this market is perfectly elastic. So, there is a fixed ceiling on the supply of the commodity with respect to the demand, while demand varies between (250,490).

In [ ]:
```
#All necessary functions for Part A code chunks in the helper.py file

plot_sup_dem(20, [sup_range], 20, [dem_range], 'fixed')
```



In [ ]:
```
#***DO NOT RUN*** - this simulates the market
#PART A - B
#Code changed to output data for 50 trials at R = 10, 20, 30, 40, 50, 60,
output_folder = 'part_b_50_R70'
os.makedirs(output_folder, exist_ok=True)

# Loop to run the trial 50 times
for i in range(50):
    trial_id = "part_b_50_R70/" + "trial_"+ str(i)
    # Call your function with appropriate parameters
    output = market_session(trial_id, start_time, end_time, traders_spec,
```

In [ ]:
```
##***DO NOT RUN*** - this simulates the market
#PART A - B
#Code changed to output data for 500 trials at R = 10, 20, 30, 40, 50, 60
output_folder = 'part_b_500_R90'
os.makedirs(output_folder, exist_ok=True)

# Loop to run the trial 50 times
for i in range(500):
    trial_id = "part_b_500_R90/" + "trial_"+ str(i)
    # Call your function with appropriate parameters
    output = market_session(trial_id, start_time, end_time, traders_spec,
```

We first conduct an empirical test to see which strategy is more profitable at R=50. This is done for 50 and 500 trials.

In [ ]:
```
# let n be the number of trials
# For part_b_50 where R = 50
n1 = 50
file_path_n50 = f"part_b_50_R50/trial_*_avg_balance.csv"
n2 = 500
file_path_n500 = f"part_b_500_R50/trial_*_avg_balance.csv"

profit_by_shvr_50, profit_by_zic_50 , winner_50= read_csv(file_path_n50)
profit_by_shvr_500, profit_by_zic_500, winner_500= read_csv(file_path_n50

num_shvr_winners50 = winner_50['SHVR']
num_zic_winners50 = winner_50['ZIC']

num_shvr_winners500 = winner_500['SHVR']
num_zic_winners500 = winner_500['ZIC']

if num_shvr_winners50 > num_zic_winners50:
    print(f"SHVR won more trials ({num_shvr_winners50}) than ZIC ({num_zi
elif num_shvr_winners500 < num_zic_winners500:
    print(f"ZIC won more trials ({num_zic_winners50}) than SHVR ({num_shv

if num_shvr_winners500 > num_zic_winners500:
    print(f"SHVR won more trials ({num_shvr_winners500}) than ZIC ({num_z
elif num_shvr_winners500 < num_zic_winners500:
    print(f"ZIC won more trials ({num_zic_winners500}) than SHVR ({num_sh
```

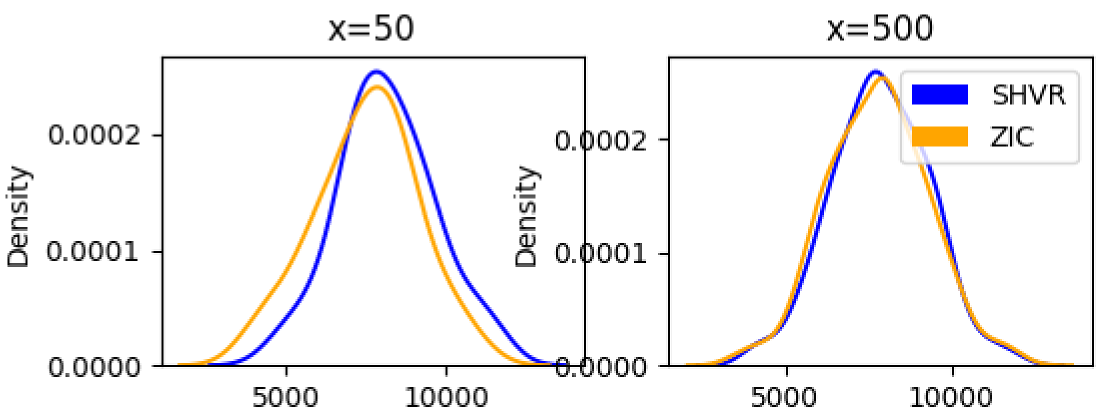
```
SHVR won more trials (28) than ZIC (22)for 50 trials
ZIC won more trials (253) than SHVR (247)for 500 trials
```

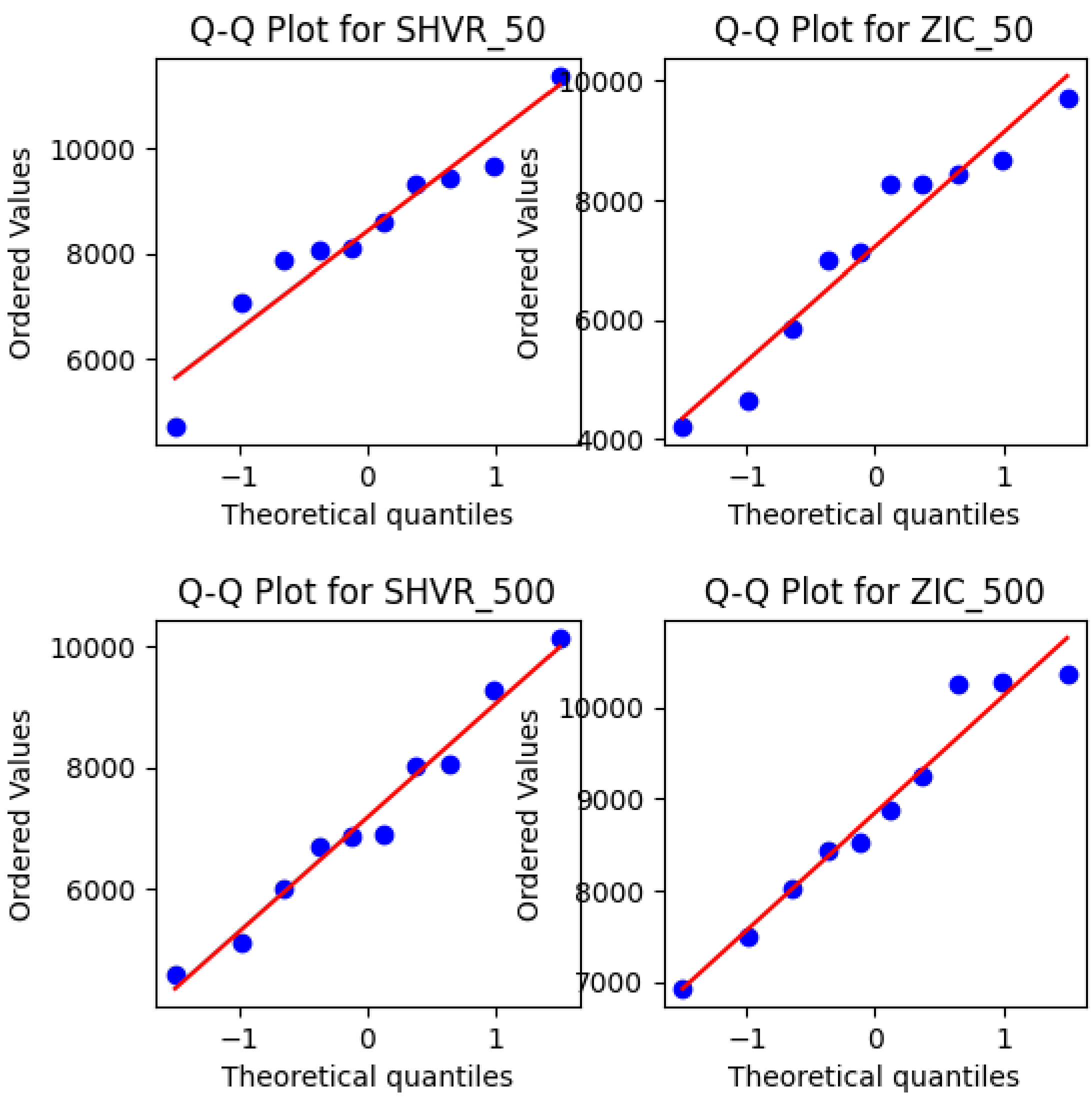
In [ ]:
```
fig, axs = plt.subplots(1, 2, figsize=(6, 2))

plt.legend(handles=[Patch(facecolor='blue', label='SHVR'), Patch(facecolo


perform_shapirowilktest(profit_by_shvr_50)
perform_shapirowilktest(profit_by_zic_50)
ttest_50_two_sided = perform_ttest(profit_by_zic_50, profit_by_shvr_50, a
ttest_50_greater = perform_ttest(profit_by_zic_50, profit_by_shvr_50, alt
ttest_50_less = perform_ttest(profit_by_zic_50, profit_by_shvr_50, altern
plot_kde(profit_by_shvr_50, profit_by_zic_50, n1, axs[0])
fig_qq_50, axs_qq_50 = plt.subplots(1, 2, figsize=(6, 2.5))
create_qq_plots(profit_by_shvr_50, profit_by_zic_50, 50, axs_qq_50)

# For part_b_500 where R = 50
perform_shapirowilktest(profit_by_shvr_50)
perform_shapirowilktest(profit_by_zic_50)
ttest_500_two_sided = perform_ttest(profit_by_zic_500, profit_by_shvr_500
ttest_500_greater = perform_ttest(profit_by_zic_500, profit_by_shvr_500,
ttest_500_less = perform_ttest(profit_by_zic_500, profit_by_shvr_500, alt

plot_kde(profit_by_shvr_500, profit_by_zic_500, n2, axs[1])
fig_qq_500, axs_qq_500 = plt.subplots(1, 2, figsize=(6,2.5))
create_qq_plots(profit_by_shvr_500, profit_by_zic_500, 500, axs_qq_500)

df_ttest_results = pd.DataFrame({
    'Metric': ['t-statistic', 'p-value', 'Null Hypothesis Finding', 'Alte
    '50 Trials Two-Sided SHVR = ZIC': ttest_50_two_sided,
    '50 Trials Greater SHVR > ZIC': ttest_50_greater,
    '50 Trials Less SHVR < ZIC': ttest_50_less,
    '500 Trials Two-Sided SHVR = ZIC': ttest_500_two_sided,
```

```
    '500 Trials GreaterSHVR > ZIC': ttest_500_greater,
    '500 Trials Less SHVR < ZIC': ttest_500_less
})

df_ttest_results = df_ttest_results.set_index('Metric').T

df_ttest_results
```

Out[ ]:

Metric	t-statistic	p-value	Null Hypothesis Finding	Alternative Hypothesis Finding
50 Trials Two-Sided SHVR = ZIC	-1.502066	0.139498	Failed to reject H_0. No significant difference...	Not enough evidence to support a significant d...
50 Trials Greater SHVR > ZIC	-1.502066	0.930251	Failed to reject H_0. No significant difference...	Not enough evidence to support a significant d...
50 Trials Less SHVR < ZIC	-1.502066	0.069749	Failed to reject H_0. No significant difference...	Not enough evidence to support a significant d...
500 Trials Two-Sided SHVR = ZIC	-0.646527	0.518235	Failed to reject H_0. No significant difference...	Not enough evidence to support a significant d...
500 Trials GreaterSHVR > ZIC	-0.646527	0.740882	Failed to reject H_0. No significant difference...	Not enough evidence to support a significant d...
500 Trials Less SHVR < ZIC	-0.646527	0.259118	Failed to reject H_0. No significant difference...	Not enough evidence to support a significant d...





From the KDE/QQ plots and Shapiro test we can assume that both distributions of profits_by_shvr and profit_by_zic are normal for both n=50,500, permitting us to perform the t-test.

The t-test was carried out to check if one strategy was better than the other, but the results are inconclusive as none of the null hypotheses could be rejected at 95% confidence.

Conclusion

Given the current evidence, we cannot conclude that either strategy is better since they are statistically indistinguishable.

PART B

Let us now test SHVR and ZIC at different ratios. Here the number of SHVR is given by $int(R/100 * num)$ for R=10,20,30,40,50,60,70,80,90.

In [ ]:
```
#All necessary functions for Part B code chunks in the helper.py file
from BSE import *
```

```
from helper import *
```

In [ ]:
```
# In 50 trials
x_values = [10, 20, 30, 40, 50, 60, 70, 80, 90]

file_pattern = "part_b_50_R{x}/trial_*_avg_balance.csv"

profit_by_shvr50, profit_by_zic50, winner50 = read_csv_multi(x_values, fi

num_shvr_winners50 = [winner50[x]['SHVR'] for x in x_values]
num_zic_winners50 = [winner50[x]['ZIC'] for x in x_values]

if sum(num_shvr_winners50) > sum(num_zic_winners50):
    print(f"SHVR won more trials ({sum(num_shvr_winners50)}) than ZIC ({s
```

```
Results for x = 10: SHVR wins: 27, ZIC wins: 23, Ties: 0
Results for x = 20: SHVR wins: 29, ZIC wins: 21, Ties: 0
Results for x = 30: SHVR wins: 29, ZIC wins: 21, Ties: 0
Results for x = 40: SHVR wins: 25, ZIC wins: 25, Ties: 0
Results for x = 50: SHVR wins: 28, ZIC wins: 22, Ties: 0
Results for x = 60: SHVR wins: 22, ZIC wins: 28, Ties: 0
Results for x = 70: SHVR wins: 26, ZIC wins: 24, Ties: 0
Results for x = 80: SHVR wins: 27, ZIC wins: 23, Ties: 0
Results for x = 90: SHVR wins: 21, ZIC wins: 29, Ties: 0
SHVR won more trials (234) than ZIC (216)
```

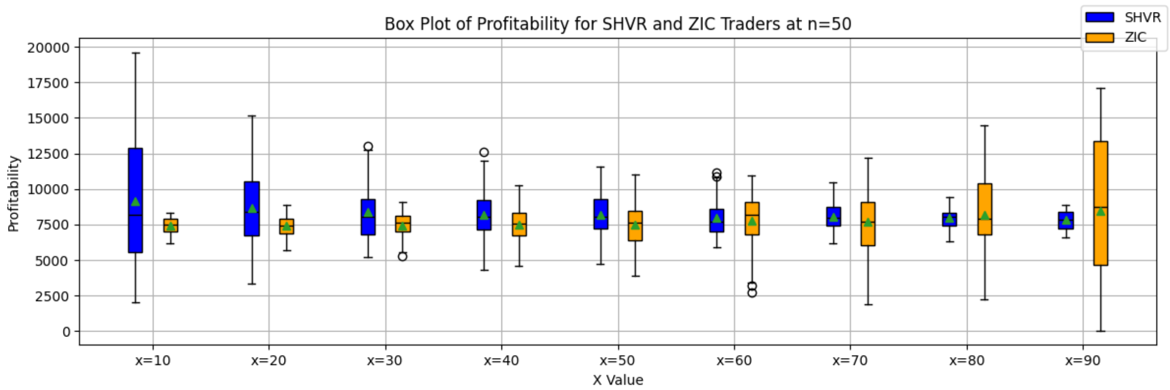
In [ ]:
```
# In 500 trials
x_values = [10, 20, 30, 40, 50, 60, 70, 80, 90]

file_pattern = "part_b_500_R{x}/trial_*_avg_balance.csv"

profit_by_shvr500, profit_by_zic500, winner500 = read_csv_multi(x_values,

num_shvr_winners500 = [winner500[x]['SHVR'] for x in x_values]
num_zic_winners500 = [winner500[x]['ZIC'] for x in x_values]

if sum(num_shvr_winners500) > sum(num_zic_winners500):
    print(f"SHVR won more trials ({sum(num_shvr_winners500)}) than ZIC ({
```

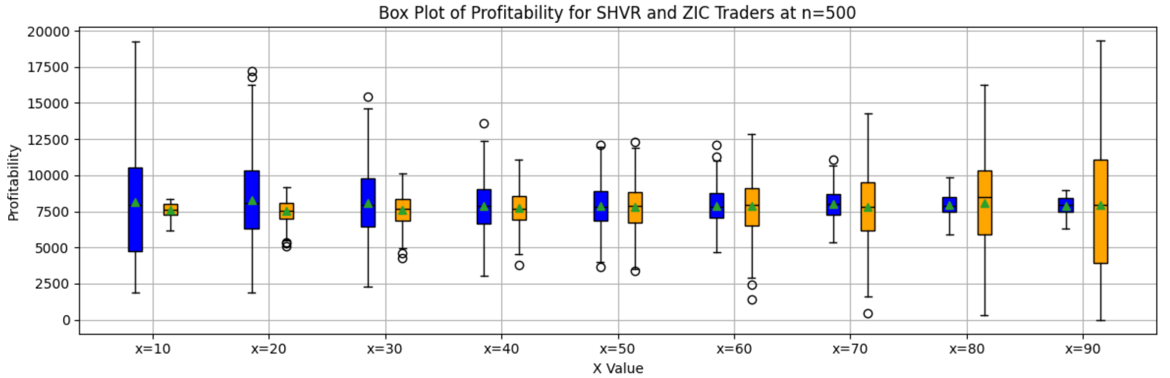
```
Results for x = 10: SHVR wins: 272, ZIC wins: 228, Ties: 0
Results for x = 20: SHVR wins: 279, ZIC wins: 221, Ties: 0
Results for x = 30: SHVR wins: 267, ZIC wins: 233, Ties: 0
Results for x = 40: SHVR wins: 264, ZIC wins: 236, Ties: 0
Results for x = 50: SHVR wins: 247, ZIC wins: 253, Ties: 0
Results for x = 60: SHVR wins: 241, ZIC wins: 259, Ties: 0
Results for x = 70: SHVR wins: 258, ZIC wins: 242, Ties: 0
Results for x = 80: SHVR wins: 216, ZIC wins: 284, Ties: 0
Results for x = 90: SHVR wins: 246, ZIC wins: 254, Ties: 0
SHVR won more trials (2290) than ZIC (2210)
```

At both 50 and 500 trials SHVR is more profitable, specially at lower ratios. This is because it has more opportunities to shave profit from other traders.

In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.patches import Patch

fig, axs = plt.subplots(2, 1, figsize=(12, 8))

boxprops = dict(facecolor='blue', edgecolor='black')
```

```
medianprops = dict(color='black')
meanprops = dict(linestyle='--', color='black')

# Plot for n=50
positions_shvr_50 = [x - 1.5 for x in x_values]
positions_zic_50 = [x + 1.5 for x in x_values]
axs[0].boxplot([profit_by_shvr50[x] for x in x_values], positions=positio
               patch_artist=True, boxprops=boxprops, medianprops=medianpr
boxprops['facecolor'] = 'orange'
axs[0].boxplot([profit_by_zic50[x] for x in x_values], positions=position
               patch_artist=True, boxprops=boxprops, medianprops=medianpr

# Plot for n=500
positions_shvr_500 = [x - 1.5 for x in x_values]
positions_zic_500 = [x + 1.5 for x in x_values]
boxprops['facecolor'] = 'blue'
axs[1].boxplot([profit_by_shvr500[x] for x in x_values], positions=positi
               patch_artist=True, boxprops=boxprops, medianprops=medianpr
boxprops['facecolor'] = 'orange'
axs[1].boxplot([profit_by_zic500[x] for x in x_values], positions=positio
               patch_artist=True, boxprops=boxprops, medianprops=medianpr

axs[0].set_title('Box Plot of Profitability for SHVR and ZIC Traders at n
axs[1].set_title('Box Plot of Profitability for SHVR and ZIC Traders at n
for ax in axs:
    ax.set_xlabel('X Value')
    ax.set_ylabel('Profitability')
    ax.set_xticks(x_values)
    ax.set_xticklabels([f'x={x}' for x in x_values])
    ax.grid(True)

legend_elements = [Patch(facecolor='blue', edgecolor='black', label='SHVR
                   Patch(facecolor='orange', edgecolor='black', label='ZI
fig.legend(handles=legend_elements, loc='upper right')

plt.tight_layout()
plt.show()
```





The IQR of SHVR from the box plot shows large variance at lower ratios. This could be because, at lower ratios, SHVR has a lot of ZIC's to "steal" profit by shaving profit by a pence. But as the number of SHVR increases, it's IQR drops significantly. This could suggest that they are competing with each other.

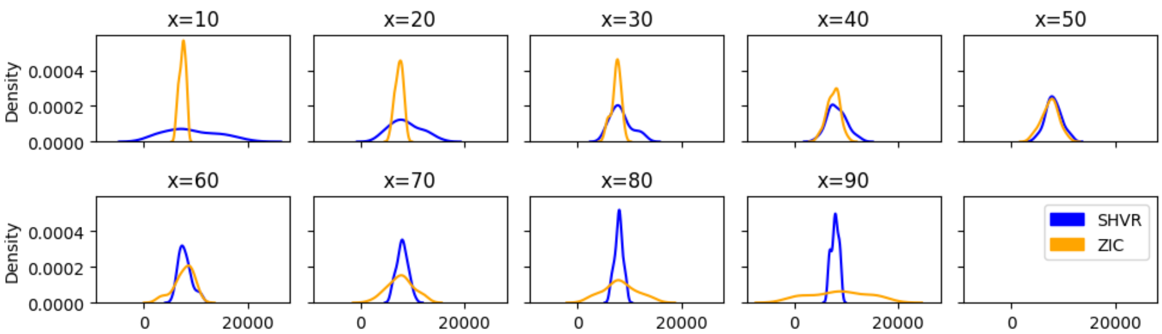
The plot also shows that lower SHVR ratios are associated with higher means. When the ratio increases, ZIC appears to have a higher mean at 50 trials, but this is likely caused by the smaller sample size. In the 500 trials data, starting from x=40, the means for both are similar. So, we can make an assumption that SHVR is better at lower ratios.

Following are the statistical tests to support these observations.

In [ ]:

```
x_values = [10, 20, 30, 40, 50, 60, 70, 80, 90]
normality_results_df = perform_normality_tests(x_values, profit_by_shvr50

normality_shvr = []
normality_zic = []
test_statistic = []
test_p_value = []
test_type = []
test_findings = []

for x in x_values:
    norm_shvr_tuple = normality_results_df[normality_results_df['X Value'
    norm_zic_tuple = normality_results_df[normality_results_df['X Value']

    norm_shvr = norm_shvr_tuple[0] if isinstance(norm_shvr_tuple, tuple)
    norm_zic = norm_zic_tuple[0] if isinstance(norm_zic_tuple, tuple) els

    normality_shvr.append(norm_shvr > 0.05)
    normality_zic.append(norm_zic > 0.05)
```

```
    if norm_shvr > 0.05 and norm_zic > 0.05:
        t_stat, p_val, null_hypothesis_finding, alternative_hypothesis_fi
        test = 't-test'
        findings = null_hypothesis_finding if p_val > 0.05 else alternati
    else:
        t_stat, p_val, null_hypothesis_finding, alternative_hypothesis_fi
        test = 'Wilcoxon'
        findings = null_hypothesis_finding if p_val > 0.05 else alternati

    test_statistic.append(t_stat)
    test_p_value.append(p_val)
    test_type.append(test)
    test_findings.append(findings)

results_df = pd.DataFrame({
    'X Value': x_values,
    'Normality SHVR': normality_shvr,
    'Normality ZIC': normality_zic,
    'Test Type': test_type,
    'P-Value': test_p_value,
    'Findings': test_findings
})

results_df
```



Out[ ]:

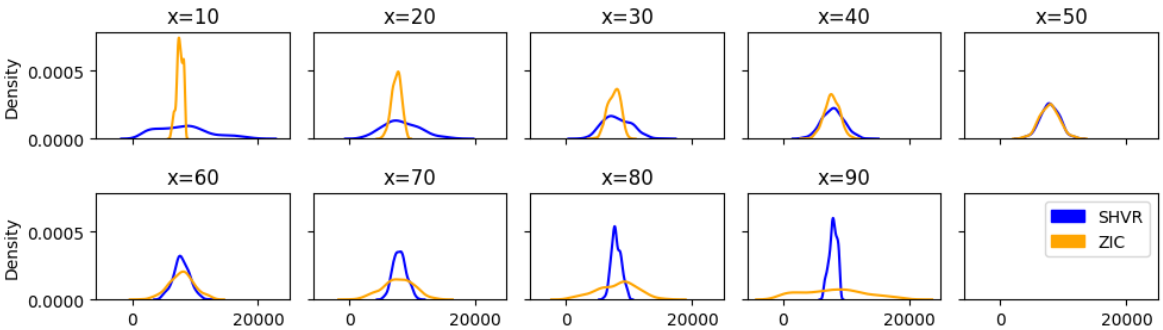
	X Value	Normality SHVR	Normality ZIC	Test Type	P-Value	Findings
0	10	False	False	Wilcoxon	0.048547	Evidence supports the alternative hypothesis.
1	20	True	True	t-test	0.010522	Evidence supports the alternative hypothesis.
2	30	False	False	Wilcoxon	0.054765	Failed to reject H_0. No significant difference.
3	40	True	True	t-test	0.051320	Failed to reject H_0. No significant difference...
4	50	True	True	t-test	0.069749	Failed to reject H_0. No significant difference...
5	60	False	False	Wilcoxon	0.557057	Failed to reject H_0. No significant difference.
6	70	True	True	t-test	0.246053	Failed to reject H_0. No significant difference...
7	80	True	True	t-test	0.672379	Failed to reject H_0. No significant difference...
8	90	False	False	Wilcoxon	0.818878	Failed to reject H_0. No significant difference.

For n=50, we have done a t-test for normal distributions and Wilcoxon Signed Rank test for non-normal distributions. We do it one tailed in the favour of SHVR>ZIC based on our previous assumption. Given our results, we reject the null hypothesis at x=10,20, meaning SHVR performs statistically better than ZIC in these conditions. The results are very close to the confidence interval at x=30,40 but not statistically strong enough to reject the null hypothesis with 95% confidence.

In [ ]:

```
x_values = [10, 20, 30, 40, 50, 60, 70, 80, 90]
normality_results_df = perform_normality_tests(x_values, profit_by_shvr50

normality_shvr = []
normality_zic = []
test_statistic = []
test_p_value = []
test_type = []
test_findings = []

for x in x_values:
    norm_shvr_tuple = normality_results_df[normality_results_df['X Value'
    norm_zic_tuple = normality_results_df[normality_results_df['X Value']

    # Extract the actual p-value from the tuple
    norm_shvr = norm_shvr_tuple[0] if isinstance(norm_shvr_tuple, tuple)
    norm_zic = norm_zic_tuple[0] if isinstance(norm_zic_tuple, tuple) els

    normality_shvr.append(norm_shvr > 0.05)
    normality_zic.append(norm_zic > 0.05)

    if norm_shvr > 0.05 and norm_zic > 0.05:
        t_stat, p_val, null_hypothesis_finding, alternative_hypothesis_fi
```

```
        test = 't-test'
        findings = null_hypothesis_finding if p_val > 0.05 else alternati
    else:
        t_stat, p_val, null_hypothesis_finding, alternative_hypothesis_fi
        test = 'Wilcoxon'
        findings = null_hypothesis_finding if p_val > 0.05 else alternati

    test_statistic.append(t_stat)
    test_p_value.append(p_val)
    test_type.append(test)
    test_findings.append(findings)

results_df = pd.DataFrame({
    'X Value': x_values,
    'Normality SHVR': normality_shvr,
    'Normality ZIC': normality_zic,
    'Test Type': test_type,
    'P-Value': test_p_value,
    'Findings': test_findings
})

results_df
```



Out[ ]:

	X Value	Normality SHVR	Normality ZIC	Test Type	P-Value	Findings
0	10	False	False	Wilcoxon	0.064572	Failed to reject H_0. No significant difference.
1	20	False	False	Wilcoxon	0.000025	Evidence supports the alternative hypothesis.
2	30	False	False	Wilcoxon	0.002072	Evidence supports the alternative hypothesis.
3	40	True	True	t-test	0.112493	Failed to reject H_0. No significant difference...
4	50	True	True	t-test	0.259118	Failed to reject H_0. No significant difference...
5	60	True	True	t-test	0.541352	Failed to reject H_0. No significant difference...
6	70	True	True	t-test	0.078504	Failed to reject H_0. No significant difference...
7	80	False	False	Wilcoxon	0.899271	Failed to reject H_0. No significant difference.
8	90	False	False	Wilcoxon	0.461798	Failed to reject H_0. No significant difference.

At n=500, we do the same tests as before. Here, with the same null hypothesis, SHVR outperforms ZIC at x=20,30. At x=10 SHVR might be more profitable, but there is insufficient evidence to reject the null hypothesis.

Conclusions

At a lower ratio (x=10,20,30) of SHVR, it is statistically more profitable than ZIC. Whereas at the higher ratios (x=40,50,60,70,80,90) the evidence is inconclusive, hence their profitability is statistically indistinguishable. This might be due to the fact that SHVR will undercut each other to get their profits at higher ratios. As ZIC is essentially a random 'noise' trader, its performance isn't affected by the number of SHVR traders, hence there isn't a significant causal impact on its profits relative to SHVR at these higher ratios.

PART C

In this section we will test ZIC, SHVR, GVWY and ZIP at different ratios and its permutations. All permutations are in the order ZIC:SHVR:ZIP:GVWY.

In [ ]:

```
#All necessary functions for Part C code chunks in the helper.py file
from itertools import permutations
from BSE import *
from helper import *
from itertools import permutations
import itertools
```

```
#Z=ZIC, S=SHVR, P=ZIP, G=GVWY
#ZIC:SHVR::ZIP:GVWY

# The below use Sup {310,310} Dem {250,490}
#base_path1 = "part_c"
#base_path2 = "part_c_500"

# The below use Sup {250,490} Dem {250,490}
#base_path3 = "part_c_10_SDC"
#base_path4 = "part_c_10_24"
#num_trials1 = 10
#num_trials2 = 500
```

In [ ]:
```
start_time = 0

#set 4 end time 24 hours
end_time = 60 * 60

total_num = 20


#ZIC:SHVR:ZIO:GVWY

# Section 1 - 25:25:25:25
sec1_ratios = [0.25, 0.25, 0.25, 0.25]
sec1_permutations = list(permutations(sec1_ratios))

# Section 2 - 40:20:20:20 (all permutations)
sec2_ratios = [0.4, 0.2, 0.2, 0.2]
sec2_permutations = list(permutations(sec2_ratios))

# Section 3 - 10:30:30:30 (all permutations)
sec3_ratios = [0.1, 0.3, 0.3, 0.3]
sec3_permutations = list(permutations(sec3_ratios))

# Section 4 - 70:10:10:10 (all permutations)
sec4_ratios = [0.7, 0.1, 0.1, 0.1]
sec4_permutations = list(permutations(sec4_ratios))

sup_range = (310,310)
dem_range = (250,490)

dump_all = True
verbose = False

supply_schedule = [{'from': start_time, 'to': end_time, 'ranges': [sup_ra
demand_schedule = [{'from': start_time, 'to': end_time, 'ranges': [dem_ra

order_interval = 15
order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
               'interval': order_interval, 'timemode': 'periodic'}

dump_flags = {'dump_blotters': True, 'dump_lobs': True, 'dump_strats': Tr
              'dump_avgbals': True, 'dump_tape': True}
```

For this part we have run 4 sets of experiments. Set 1 and 2 use the same supply and demand range as previous parts supply (310,310),demand (250,490) run over n=10, 500 respectively. Set 3 and set 4 use a different range supply (250,490),demand

(250,490) with set 4 running for 24 hours instead of 1 hour like every other experiment.

In [ ]:
```
# ***DO NOT RUN*** - this simulates the market

run_section('10_S1', sec1_permutations, total_num)

run_section('10_S2', sec2_permutations, total_num)

run_section('10_S3', sec3_permutations, total_num)

run_section('10_S4', sec4_permutations, total_num)
```

In [ ]:
```
base_path1 = "part_c"
num_trials1 = 10
sections = {
    '10_S1': itertools.product([25], repeat=4),
    '10_S2': itertools.permutations([40, 20, 20, 20]),
    '10_S3': itertools.permutations([10, 30, 30, 30]),
    '10_S4': itertools.permutations([70, 10, 10, 10])
}
section_profitabilities1 = read_and_process_files(sections, num_trials1,
title1 = "Set 1 - 10 Trials - Sup (310,310) Dem (250,490)"
```

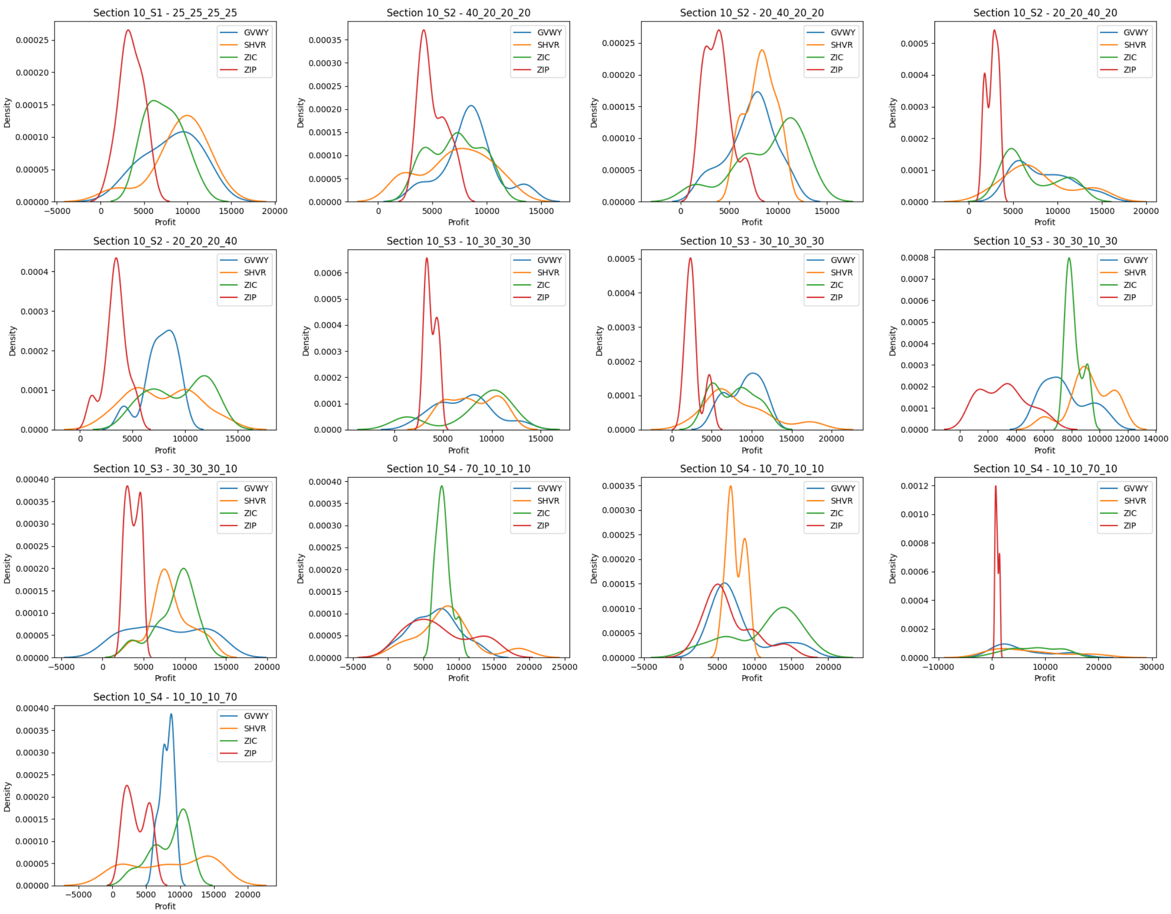
In [ ]:
```
base_path2 = "part_c_500"
num_trials2 = 500
sections = {
    '10_S1': itertools.product([25], repeat=4),
    '10_S2': itertools.permutations([40, 20, 20, 20]),
    '10_S3': itertools.permutations([10, 30, 30, 30]),
    '10_S4': itertools.permutations([70, 10, 10, 10])
}
section_profitabilities2 = read_and_process_files(sections, num_trials2,
title2 = "Set 2 - 500 Trials - Sup (310,310) Dem (250,490)"
```

In [ ]:
```
base_path3 = "part_c_10_SDC"
num_trials1 = 10
sections = {
    '10_S1': itertools.product([25], repeat=4),
    '10_S2': itertools.permutations([40, 20, 20, 20]),
    '10_S3': itertools.permutations([10, 30, 30, 30]),
    '10_S4': itertools.permutations([70, 10, 10, 10])
}
section_profitabilities3 = read_and_process_files(sections, num_trials1,
title3 = "Set 3 - 10 Trials - Sup (250,490) Dem (250,490)"
```

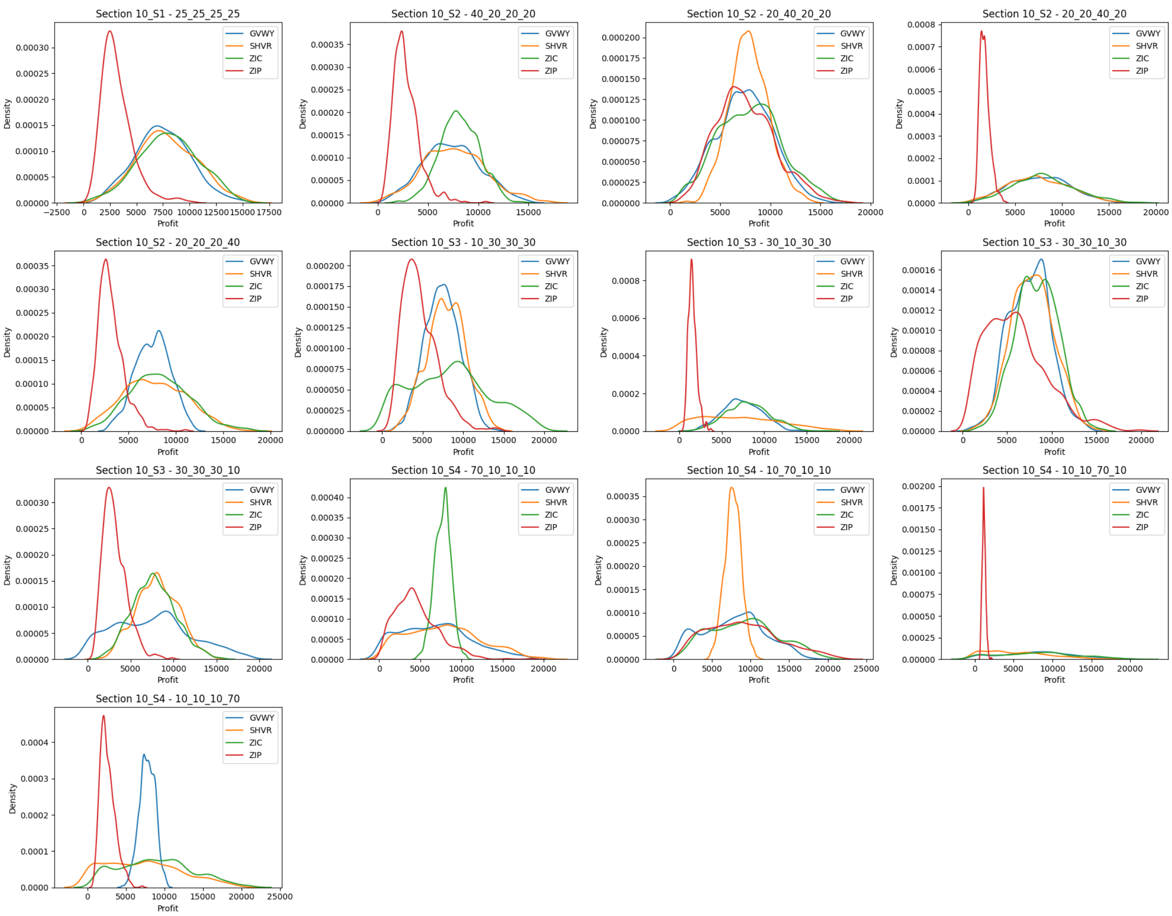
In [ ]:
```
base_path4 = "part_c_10_24"
num_trials1 = 10
sections = {
    '10_S1': itertools.product([25], repeat=4),
    '10_S2': itertools.permutations([40, 20, 20, 20]),
    '10_S3': itertools.permutations([10, 30, 30, 30]),
    '10_S4': itertools.permutations([70, 10, 10, 10])
}
section_profitabilities4 = read_and_process_files(sections, num_trials1,
title4 = "Set 4 - 10 Trials - Sup (250,490) Dem (250,490), 24 hours"
```

In [ ]:
```
plot_all_kdes(section_profitabilities1,"Set 1 - 10 Trials - Sup (310,310)

plot_all_kdes(section_profitabilities2, "Set 2 - 500 Trial - Sup (310,310

plot_all_kdes(section_profitabilities3, "Set 3 - 10 Trials - Sup (250,490

plot_all_kdes(section_profitabilities4, "Set 4 - 10 Trials - Sup (250,490
```

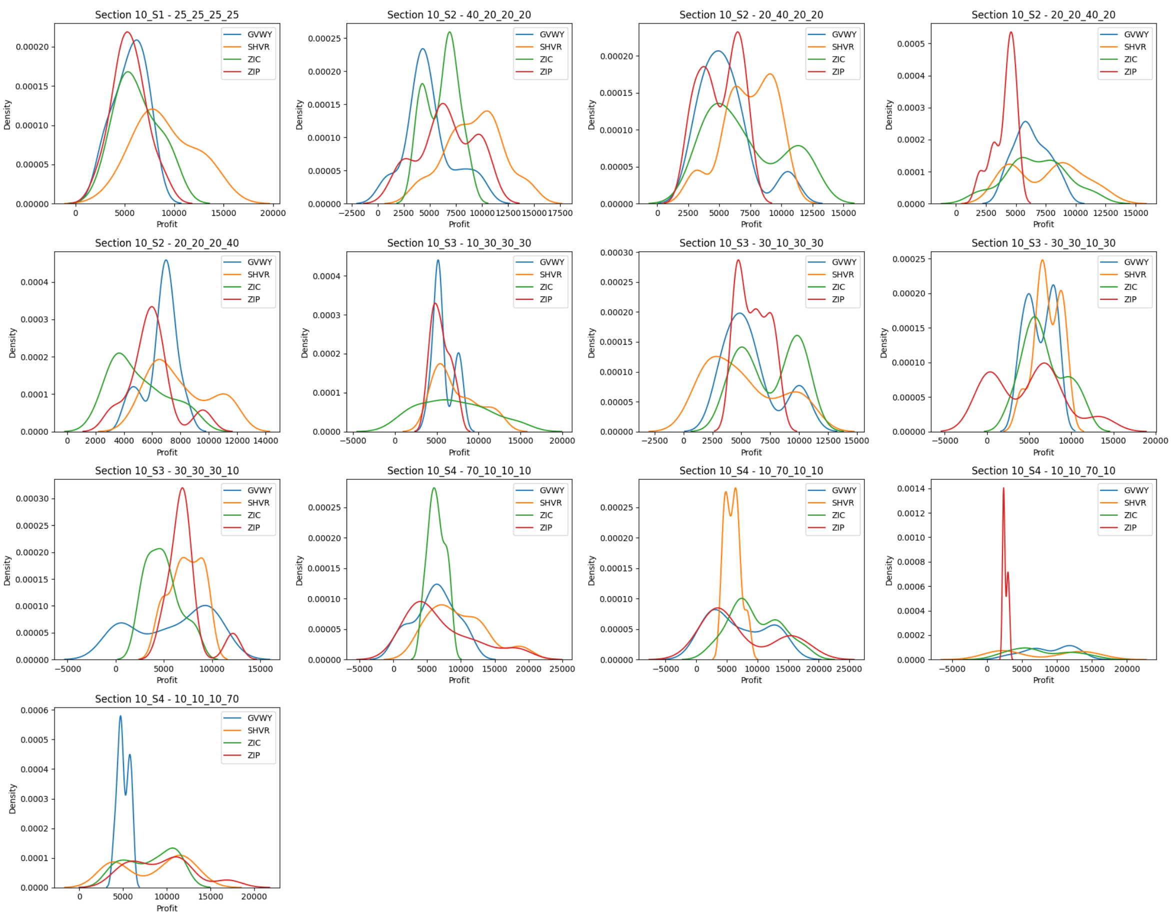
Set 1 - 10 Trials - Sup (310,310) Dem (250,490)



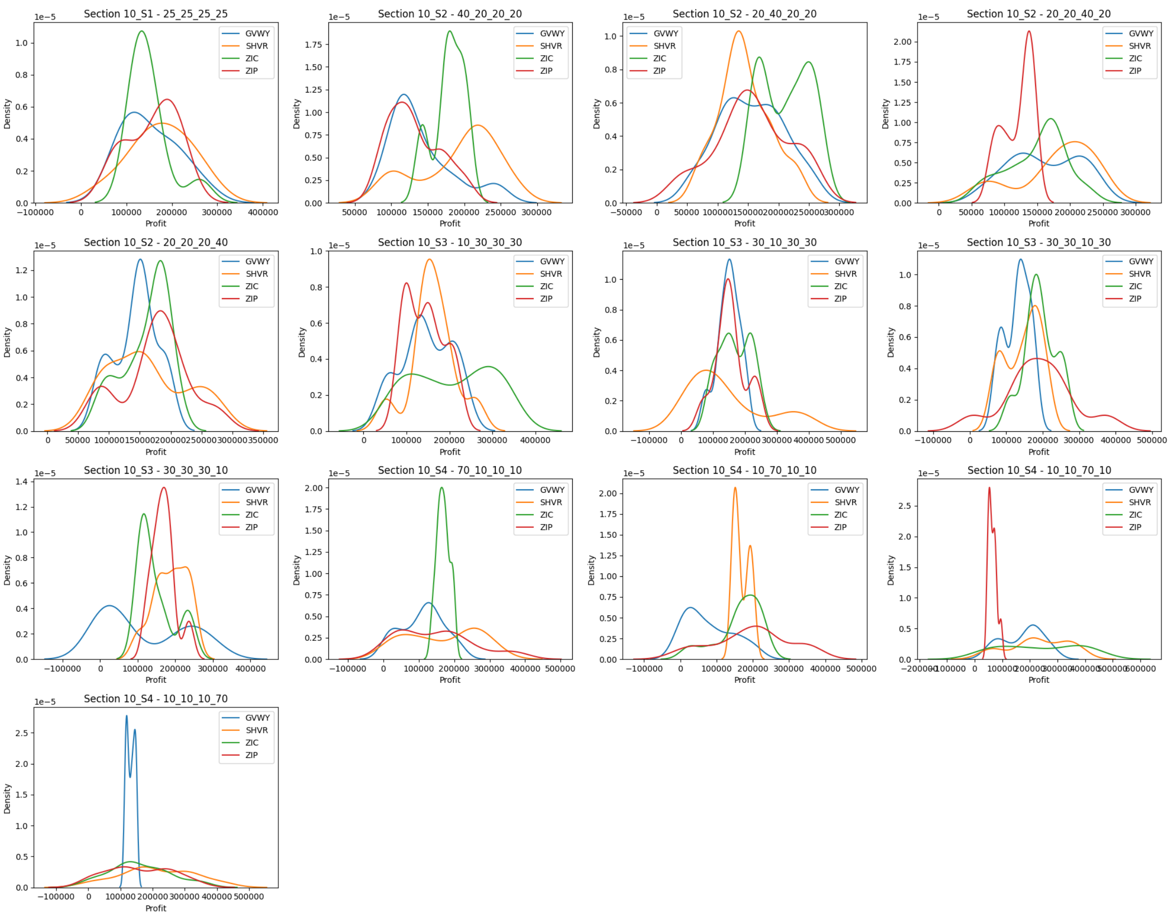
Set 2 - 500 Trial - Sup (310,310) Dem (250,490)



Set 3 - 10 Trials - Sup (250,490) Dem (250,490)



Set 4 - 10 Trials - Sup (250,490) Dem (250,490), 24 hours



These plots show a large sum of data across the different market conditions but it is hard to conclude that there is a statistical difference between them. Some key points to note

- GVWY tends to perform consistently across the board, possibly because it operates at the market prices.
- SHVR has the highest volatility, as it is constantly adapting to the market orders.
- ZIC is performs moderately, since it is just noise constrained in a region, its profits appear to be in the middle.
- ZIP performs very well or very bad depending on market conditions, possibly because it is the only adaptive strategy in this mix.

To statistically analyse this we will do a 1 vs All comparison to test each trader against the other at every section and permutation.

We do the ANOVA test if the distributions are normal else we do the Friedman test. In most cases the distributions are not normal, except when they are in equal ratios of each other.

In [ ]:

```
test_results1 = perform_1vALL(section_profitabilities1, title1)
test_results1.to_csv('part_c_10.csv', index=False)
test_results1
```

```
Set 1 - 10 Trials - Sup (310,310) Dem (250,490)
```

Out[ ]:

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
0	10_S1	25_25_25_25	GVWY	ANOVA	1.985634e-01	False	1.000000e+00
1	10_S1	25_25_25_25	SHVR	ANOVA	1.581985e-02	True	8.226324e-01
2	10_S1	25_25_25_25	ZIC	ANOVA	7.651014e-01	False	1.000000e+00
3	10_S1	25_25_25_25	ZIP	ANOVA	1.050060e-05	True	5.460315e-04
4	10_S2	40_20_20_20	GVWY	Friedman	1.508967e-07	True	7.846629e-06
5	10_S2	40_20_20_20	SHVR	Friedman	1.508967e-07	True	7.846629e-06
6	10_S2	40_20_20_20	ZIC	Friedman	1.508967e-07	True	7.846629e-06
7	10_S2	40_20_20_20	ZIP	Friedman	1.508967e-07	True	7.846629e-06
8	10_S2	20_40_20_20	GVWY	Friedman	1.591900e-15	True	8.277882e-14
9	10_S2	20_40_20_20	SHVR	Friedman	1.591900e-15	True	8.277882e-14
10	10_S2	20_40_20_20	ZIC	Friedman	1.591900e-15	True	8.277882e-14
11	10_S2	20_40_20_20	ZIP	Friedman	1.591900e-15	True	8.277882e-14
12	10_S2	20_20_40_20	GVWY	Friedman	7.322247e-22	True	3.807569e-20
13	10_S2	20_20_40_20	SHVR	Friedman	7.322247e-22	True	3.807569e-20
14	10_S2	20_20_40_20	ZIC	Friedman	7.322247e-22	True	3.807569e-20
15	10_S2	20_20_40_20	ZIP	Friedman	7.322247e-22	True	3.807569e-20
16	10_S2	20_20_20_40	GVWY	Friedman	5.484381e-16	True	2.851878e-14
17	10_S2	20_20_20_40	SHVR	Friedman	5.484381e-16	True	2.851878e-14
18	10_S2	20_20_20_40	ZIC	Friedman	5.484381e-16	True	2.851878e-14
19	10_S2	20_20_20_40	ZIP	Friedman	5.484381e-16	True	2.851878e-14
20	10_S3	10_30_30_30	GVWY	Friedman	5.484381e-16	True	2.851878e-14

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
21	10_S3	10_30_30_30	SHVR	Friedman	5.484381e-16	True	2.851878e-14
22	10_S3	10_30_30_30	ZIC	Friedman	5.484381e-16	True	2.851878e-14
23	10_S3	10_30_30_30	ZIP	Friedman	5.484381e-16	True	2.851878e-14
24	10_S3	30_10_30_30	GVWY	Friedman	3.047197e-21	True	1.584542e-19
25	10_S3	30_10_30_30	SHVR	Friedman	3.047197e-21	True	1.584542e-19
26	10_S3	30_10_30_30	ZIC	Friedman	3.047197e-21	True	1.584542e-19
27	10_S3	30_10_30_30	ZIP	Friedman	3.047197e-21	True	1.584542e-19
28	10_S3	30_30_10_30	GVWY	Friedman	3.475415e-24	True	1.807216e-22
29	10_S3	30_30_10_30	SHVR	Friedman	3.475415e-24	True	1.807216e-22
30	10_S3	30_30_10_30	ZIC	Friedman	3.475415e-24	True	1.807216e-22
31	10_S3	30_30_10_30	ZIP	Friedman	3.475415e-24	True	1.807216e-22
32	10_S3	30_30_30_10	GVWY	Friedman	5.484381e-16	True	2.851878e-14
33	10_S3	30_30_30_10	SHVR	Friedman	5.484381e-16	True	2.851878e-14
34	10_S3	30_30_30_10	ZIC	Friedman	5.484381e-16	True	2.851878e-14
35	10_S3	30_30_30_10	ZIP	Friedman	5.484381e-16	True	2.851878e-14
36	10_S4	70_10_10_10	GVWY	Friedman	6.578905e-02	False	1.000000e+00
37	10_S4	70_10_10_10	SHVR	Friedman	6.578905e-02	False	1.000000e+00
38	10_S4	70_10_10_10	ZIC	Friedman	6.578905e-02	False	1.000000e+00
39	10_S4	70_10_10_10	ZIP	Friedman	6.578905e-02	False	1.000000e+00
40	10_S4	10_70_10_10	GVWY	Friedman	4.309971e-07	True	2.241185e-05
41	10_S4	10_70_10_10	SHVR	Friedman	4.309971e-07	True	2.241185e-05

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
42	10_S4	10_70_10_10	ZIC	Friedman	4.309971e-07	True	2.241185e-05
43	10_S4	10_70_10_10	ZIP	Friedman	4.309971e-07	True	2.241185e-05
44	10_S4	10_10_70_10	GVWY	Friedman	1.909919e-14	True	9.931580e-13
45	10_S4	10_10_70_10	SHVR	Friedman	1.909919e-14	True	9.931580e-13
46	10_S4	10_10_70_10	ZIC	Friedman	1.909919e-14	True	9.931580e-13
47	10_S4	10_10_70_10	ZIP	Friedman	1.909919e-14	True	9.931580e-13
48	10_S4	10_10_10_70	GVWY	Friedman	6.614840e-13	True	3.439717e-11
49	10_S4	10_10_10_70	SHVR	Friedman	6.614840e-13	True	3.439717e-11
50	10_S4	10_10_10_70	ZIC	Friedman	6.614840e-13	True	3.439717e-11
51	10_S4	10_10_10_70	ZIP	Friedman	6.614840e-13	True	3.439717e-11

In [ ]:

```
test_results2 = perform_1vALL(section_profitabilities2, title2)
test_results2.to_csv('part_c_500.csv', index=False)
test_results2
```

```
Set 2 - 500 Trials - Sup (310,310) Dem (250,490)
```

Out[ ]:

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted Va
0	10_S1	25_25_25_25	GVWY	Friedman	1.215918e-135	True	6.32277
1	10_S1	25_25_25_25	SHVR	Friedman	1.215918e-135	True	6.32277
2	10_S1	25_25_25_25	ZIC	Friedman	1.215918e-135	True	6.32277
3	10_S1	25_25_25_25	ZIP	Friedman	1.215918e-135	True	6.32277
4	10_S2	40_20_20_20	GVWY	Friedman	0.000000e+00	True	0.000000e
5	10_S2	40_20_20_20	SHVR	Friedman	0.000000e+00	True	0.000000e
6	10_S2	40_20_20_20	ZIC	Friedman	0.000000e+00	True	0.000000e
7	10_S2	40_20_20_20	ZIP	Friedman	0.000000e+00	True	0.000000e
8	10_S2	20_40_20_20	GVWY	Friedman	9.123214e-10	True	4.744071e
9	10_S2	20_40_20_20	SHVR	Friedman	9.123214e-10	True	4.744071e
10	10_S2	20_40_20_20	ZIC	Friedman	9.123214e-10	True	4.744071e
11	10_S2	20_40_20_20	ZIP	Friedman	9.123214e-10	True	4.744071e
12	10_S2	20_20_40_20	GVWY	Friedman	0.000000e+00	True	0.000000e
13	10_S2	20_20_40_20	SHVR	Friedman	0.000000e+00	True	0.000000e
14	10_S2	20_20_40_20	ZIC	Friedman	0.000000e+00	True	0.000000e
15	10_S2	20_20_40_20	ZIP	Friedman	0.000000e+00	True	0.000000e
16	10_S2	20_20_20_40	GVWY	Friedman	0.000000e+00	True	0.000000e
17	10_S2	20_20_20_40	SHVR	Friedman	0.000000e+00	True	0.000000e
18	10_S2	20_20_20_40	ZIC	Friedman	0.000000e+00	True	0.000000e
19	10_S2	20_20_20_40	ZIP	Friedman	0.000000e+00	True	0.000000e
20	10_S3	10_30_30_30	GVWY	Friedman	0.000000e+00	True	0.000000e
21	10_S3	10_30_30_30	SHVR	Friedman	0.000000e+00	True	0.000000e
22	10_S3	10_30_30_30	ZIC	Friedman	0.000000e+00	True	0.000000e
23	10_S3	10_30_30_30	ZIP	Friedman	0.000000e+00	True	0.000000e
24	10_S3	30_10_30_30	GVWY	Friedman	0.000000e+00	True	0.000000e
25	10_S3	30_10_30_30	SHVR	Friedman	0.000000e+00	True	0.000000e
26	10_S3	30_10_30_30	ZIC	Friedman	0.000000e+00	True	0.000000e
27	10_S3	30_10_30_30	ZIP	Friedman	0.000000e+00	True	0.000000e
28	10_S3	30_30_10_30	GVWY	Friedman	1.185507e-170	True	6.16463
29	10_S3	30_30_10_30	SHVR	Friedman	1.185507e-170	True	6.16463

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjuste Va
30	10_S3	30_30_10_30	ZIC	Friedman	1.185507e-170	True	6.16463
31	10_S3	30_30_10_30	ZIP	Friedman	1.185507e-170	True	6.16463
32	10_S3	30_30_30_10	GVWY	Friedman	0.000000e+00	True	0.000000e
33	10_S3	30_30_30_10	SHVR	Friedman	0.000000e+00	True	0.000000e
34	10_S3	30_30_30_10	ZIC	Friedman	0.000000e+00	True	0.000000e
35	10_S3	30_30_30_10	ZIP	Friedman	0.000000e+00	True	0.000000e
36	10_S4	70_10_10_10	GVWY	Friedman	4.113853e-283	True	2.13920
37	10_S4	70_10_10_10	SHVR	Friedman	4.113853e-283	True	2.13920
38	10_S4	70_10_10_10	ZIC	Friedman	4.113853e-283	True	2.13920
39	10_S4	70_10_10_10	ZIP	Friedman	4.113853e-283	True	2.13920
40	10_S4	10_70_10_10	GVWY	Friedman	4.684044e-45	True	2.435703e
41	10_S4	10_70_10_10	SHVR	Friedman	4.684044e-45	True	2.435703e
42	10_S4	10_70_10_10	ZIC	Friedman	4.684044e-45	True	2.435703e
43	10_S4	10_70_10_10	ZIP	Friedman	4.684044e-45	True	2.435703e
44	10_S4	10_10_70_10	GVWY	Friedman	0.000000e+00	True	0.000000e
45	10_S4	10_10_70_10	SHVR	Friedman	0.000000e+00	True	0.000000e
46	10_S4	10_10_70_10	ZIC	Friedman	0.000000e+00	True	0.000000e
47	10_S4	10_10_70_10	ZIP	Friedman	0.000000e+00	True	0.000000e
48	10_S4	10_10_10_70	GVWY	Friedman	0.000000e+00	True	0.000000e
49	10_S4	10_10_10_70	SHVR	Friedman	0.000000e+00	True	0.000000e
50	10_S4	10_10_10_70	ZIC	Friedman	0.000000e+00	True	0.000000e
51	10_S4	10_10_10_70	ZIP	Friedman	0.000000e+00	True	0.000000e

In [ ]:
```
test_results3 = perform_1vALL(section_profitabilities3, title3)
test_results3.to_csv('part_c_10_SDC.csv', index=False)
test_results3
```

```
Set 3 - 10 Trials - Sup (250,490) Dem (250,490)
```

Out[ ]:

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
0	10_S1	25_25_25_25	GVWY	ANOVA	1.284938e-01	False	1.000000e+0(
1	10_S1	25_25_25_25	SHVR	ANOVA	1.134124e-04	True	5.897444e-03
2	10_S1	25_25_25_25	ZIC	Friedman	8.580109e-02	False	1.000000e+0(
3	10_S1	25_25_25_25	ZIP	ANOVA	1.114217e-01	False	1.000000e+0(
4	10_S2	40_20_20_20	GVWY	Friedman	2.275348e-11	True	1.183181e-09
5	10_S2	40_20_20_20	SHVR	Friedman	2.275348e-11	True	1.183181e-09
6	10_S2	40_20_20_20	ZIC	Friedman	2.275348e-11	True	1.183181e-09
7	10_S2	40_20_20_20	ZIP	Friedman	2.275348e-11	True	1.183181e-09
8	10_S2	20_40_20_20	GVWY	Friedman	7.758074e-10	True	4.034198e-08
9	10_S2	20_40_20_20	SHVR	Friedman	7.758074e-10	True	4.034198e-08
10	10_S2	20_40_20_20	ZIC	Friedman	7.758074e-10	True	4.034198e-08
11	10_S2	20_40_20_20	ZIP	Friedman	7.758074e-10	True	4.034198e-08
12	10_S2	20_20_40_20	GVWY	Friedman	1.330287e-10	True	6.917490e-09
13	10_S2	20_20_40_20	SHVR	Friedman	1.330287e-10	True	6.917490e-09
14	10_S2	20_20_40_20	ZIC	Friedman	1.330287e-10	True	6.917490e-09
15	10_S2	20_20_40_20	ZIP	Friedman	1.330287e-10	True	6.917490e-09
16	10_S2	20_20_20_40	GVWY	Friedman	5.273789e-08	True	2.742370e-06
17	10_S2	20_20_20_40	SHVR	Friedman	5.273789e-08	True	2.742370e-06
18	10_S2	20_20_20_40	ZIC	Friedman	5.273789e-08	True	2.742370e-06
19	10_S2	20_20_20_40	ZIP	Friedman	5.273789e-08	True	2.742370e-06
20	10_S3	10_30_30_30	GVWY	Friedman	4.398497e-04	True	2.287218e-02

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
21	10_S3	10_30_30_30	SHVR	Friedman	4.398497e-04	True	2.287218e-02
22	10_S3	10_30_30_30	ZIC	Friedman	4.398497e-04	True	2.287218e-02
23	10_S3	10_30_30_30	ZIP	Friedman	4.398497e-04	True	2.287218e-02
24	10_S3	30_10_30_30	GVWY	Friedman	6.604651e-03	True	3.434419e-01
25	10_S3	30_10_30_30	SHVR	Friedman	6.604651e-03	True	3.434419e-01
26	10_S3	30_10_30_30	ZIC	Friedman	6.604651e-03	True	3.434419e-01
27	10_S3	30_10_30_30	ZIP	Friedman	6.604651e-03	True	3.434419e-01
28	10_S3	30_30_10_30	GVWY	Friedman	4.398497e-04	True	2.287218e-02
29	10_S3	30_30_10_30	SHVR	Friedman	4.398497e-04	True	2.287218e-02
30	10_S3	30_30_10_30	ZIC	Friedman	4.398497e-04	True	2.287218e-02
31	10_S3	30_30_10_30	ZIP	Friedman	4.398497e-04	True	2.287218e-02
32	10_S3	30_30_30_10	GVWY	Friedman	3.494282e-06	True	1.817026e-04
33	10_S3	30_30_30_10	SHVR	Friedman	3.494282e-06	True	1.817026e-04
34	10_S3	30_30_30_10	ZIC	Friedman	3.494282e-06	True	1.817026e-04
35	10_S3	30_30_30_10	ZIP	Friedman	3.494282e-06	True	1.817026e-04
36	10_S4	70_10_10_10	GVWY	Friedman	1.983098e-05	True	1.031211e-03
37	10_S4	70_10_10_10	SHVR	Friedman	1.983098e-05	True	1.031211e-03
38	10_S4	70_10_10_10	ZIC	Friedman	1.983098e-05	True	1.031211e-03
39	10_S4	70_10_10_10	ZIP	Friedman	1.983098e-05	True	1.031211e-03
40	10_S4	10_70_10_10	GVWY	Friedman	6.604651e-03	True	3.434419e-01
41	10_S4	10_70_10_10	SHVR	Friedman	6.604651e-03	True	3.434419e-01

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
42	10_S4	10_70_10_10	ZIC	Friedman	6.604651e-03	True	3.434419e-01
43	10_S4	10_70_10_10	ZIP	Friedman	6.604651e-03	True	3.434419e-01
44	10_S4	10_10_70_10	GVWY	Friedman	2.783931e-07	True	1.447644e-05
45	10_S4	10_10_70_10	SHVR	Friedman	2.783931e-07	True	1.447644e-05
46	10_S4	10_10_70_10	ZIC	Friedman	2.783931e-07	True	1.447644e-05
47	10_S4	10_10_70_10	ZIP	Friedman	2.783931e-07	True	1.447644e-05
48	10_S4	10_10_10_70	GVWY	Friedman	7.004462e-06	True	3.642320e-04
49	10_S4	10_10_10_70	SHVR	Friedman	7.004462e-06	True	3.642320e-04
50	10_S4	10_10_10_70	ZIC	Friedman	7.004462e-06	True	3.642320e-04
51	10_S4	10_10_10_70	ZIP	Friedman	7.004462e-06	True	3.642320e-04

In [ ]:
```
test_results4 = perform_1vALL(section_profitabilities4, title4)
test_results4.to_csv('part_c_10_24.csv', index=False)
test_results4
```

```
Set 4 - 10 Trials - Sup (250,490) Dem (250,490), 24 hours
```

Out[ ]:

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P- Valu
0	10_S1	25_25_25_25	GVWY	ANOVA	7.143077e-01	False	1.000000e+0
1	10_S1	25_25_25_25	SHVR	ANOVA	2.854049e-01	False	1.000000e+0
2	10_S1	25_25_25_25	ZIC	Friedman	6.684932e-01	False	1.000000e+0
3	10_S1	25_25_25_25	ZIP	ANOVA	9.057086e-01	False	1.000000e+0
4	10_S2	40_20_20_20	GVWY	Friedman	3.713491e-08	True	1.931015e-0
5	10_S2	40_20_20_20	SHVR	Friedman	3.713491e-08	True	1.931015e-0
6	10_S2	40_20_20_20	ZIC	Friedman	3.713491e-08	True	1.931015e-0
7	10_S2	40_20_20_20	ZIP	Friedman	3.713491e-08	True	1.931015e-0
8	10_S2	20_40_20_20	GVWY	Friedman	4.612409e-11	True	2.398453e 0
9	10_S2	20_40_20_20	SHVR	Friedman	4.612409e-11	True	2.398453e 0
10	10_S2	20_40_20_20	ZIC	Friedman	4.612409e-11	True	2.398453e 0
11	10_S2	20_40_20_20	ZIP	Friedman	4.612409e-11	True	2.398453e 0
12	10_S2	20_20_40_20	GVWY	Friedman	8.666935e-07	True	4.506806e 0
13	10_S2	20_20_40_20	SHVR	Friedman	8.666935e-07	True	4.506806e 0
14	10_S2	20_20_40_20	ZIC	Friedman	8.666935e-07	True	4.506806e 0
15	10_S2	20_20_40_20	ZIP	Friedman	8.666935e-07	True	4.506806e 0
16	10_S2	20_20_20_40	GVWY	Friedman	6.604651e-03	True	3.434419e-0
17	10_S2	20_20_20_40	SHVR	Friedman	6.604651e-03	True	3.434419e-0
18	10_S2	20_20_20_40	ZIC	Friedman	6.604651e-03	True	3.434419e-0
19	10_S2	20_20_20_40	ZIP	Friedman	6.604651e-03	True	3.434419e-0
20	10_S3	10_30_30_30	GVWY	Friedman	3.374727e-03	True	1.754858e-0

	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
21	10_S3	10_30_30_30	SHVR	Friedman	3.374727e-03	True	1.754858e-0
22	10_S3	10_30_30_30	ZIC	Friedman	3.374727e-03	True	1.754858e-0
23	10_S3	10_30_30_30	ZIP	Friedman	3.374727e-03	True	1.754858e-0
24	10_S3	30_10_30_30	GVWY	Friedman	1.572629e-04	True	8.177670e-0
25	10_S3	30_10_30_30	SHVR	Friedman	1.572629e-04	True	8.177670e-0
26	10_S3	30_10_30_30	ZIC	Friedman	1.572629e-04	True	8.177670e-0
27	10_S3	30_10_30_30	ZIP	Friedman	1.572629e-04	True	8.177670e-0
28	10_S3	30_30_10_30	GVWY	Friedman	1.741165e-06	True	9.054056e-0
29	10_S3	30_30_10_30	SHVR	Friedman	1.741165e-06	True	9.054056e-0
30	10_S3	30_30_10_30	ZIC	Friedman	1.741165e-06	True	9.054056e-0
31	10_S3	30_30_10_30	ZIP	Friedman	1.741165e-06	True	9.054056e-0
32	10_S3	30_30_30_10	GVWY	Friedman	5.595679e-05	True	2.909753e-0
33	10_S3	30_30_30_10	SHVR	Friedman	5.595679e-05	True	2.909753e-0
34	10_S3	30_30_30_10	ZIC	Friedman	5.595679e-05	True	2.909753e-0
35	10_S3	30_30_30_10	ZIP	Friedman	5.595679e-05	True	2.909753e-0
36	10_S4	70_10_10_10	GVWY	Friedman	2.486833e-02	True	1.000000e+0
37	10_S4	70_10_10_10	SHVR	Friedman	2.486833e-02	True	1.000000e+0
38	10_S4	70_10_10_10	ZIC	Friedman	2.486833e-02	True	1.000000e+0
39	10_S4	70_10_10_10	ZIP	Friedman	2.486833e-02	True	1.000000e+0
40	10_S4	10_70_10_10	GVWY	Friedman	1.122060e-11	True	5.834712e-1
41	10_S4	10_70_10_10	SHVR	Friedman	1.122060e-11	True	5.834712e-1

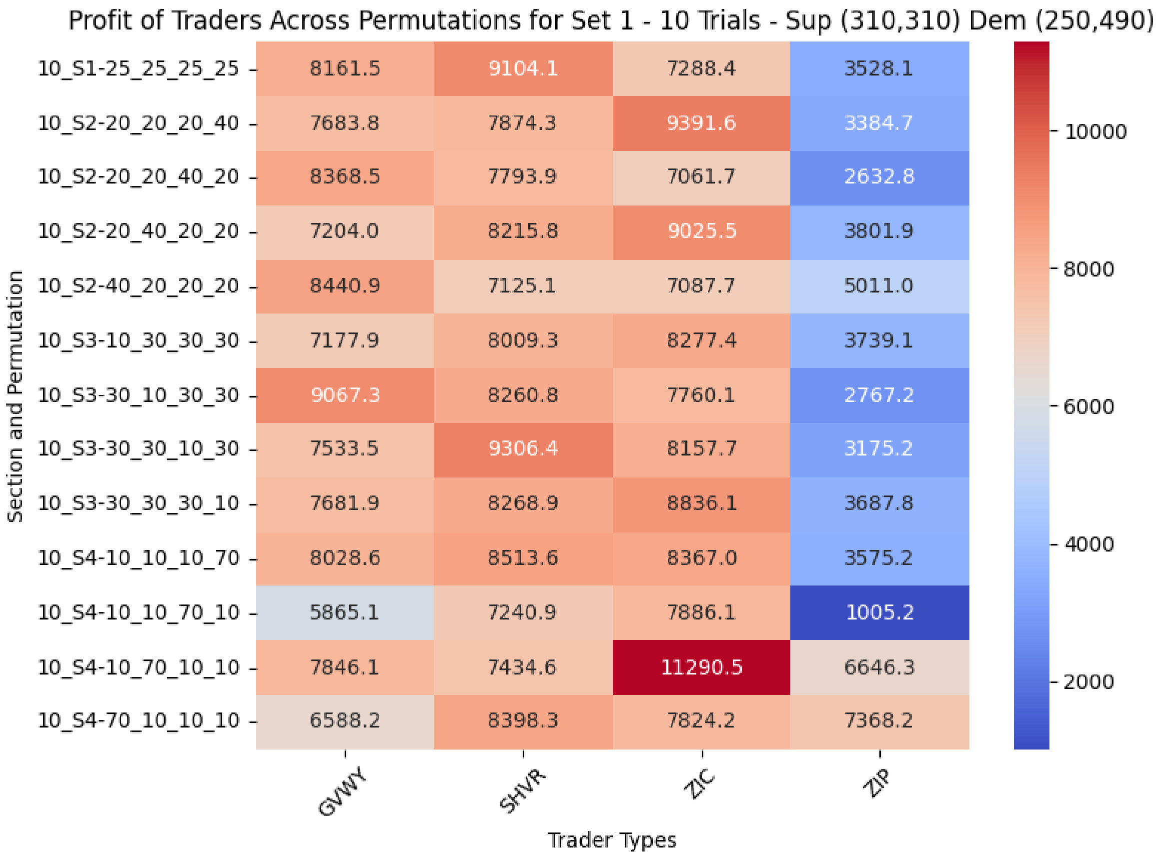
	Section	Permutation	Trader Type	Test Type	P-Value	Is Significant Before	Adjusted P-Value
42	10_S4	10_70_10_10	ZIC	Friedman	1.122060e-11	True	5.834712e-1(
43	10_S4	10_70_10_10	ZIP	Friedman	1.122060e-11	True	5.834712e-1(
44	10_S4	10_10_70_10	GVWY	Friedman	3.882601e-14	True	2.018952e-12
45	10_S4	10_10_70_10	SHVR	Friedman	3.882601e-14	True	2.018952e-12
46	10_S4	10_10_70_10	ZIC	Friedman	3.882601e-14	True	2.018952e-12
47	10_S4	10_10_70_10	ZIP	Friedman	3.882601e-14	True	2.018952e-12
48	10_S4	10_10_10_70	GVWY	Friedman	2.466940e-06	True	1.282809e-04
49	10_S4	10_10_10_70	SHVR	Friedman	2.466940e-06	True	1.282809e-04
50	10_S4	10_10_10_70	ZIC	Friedman	2.466940e-06	True	1.282809e-04
51	10_S4	10_10_10_70	ZIP	Friedman	2.466940e-06	True	1.282809e-04

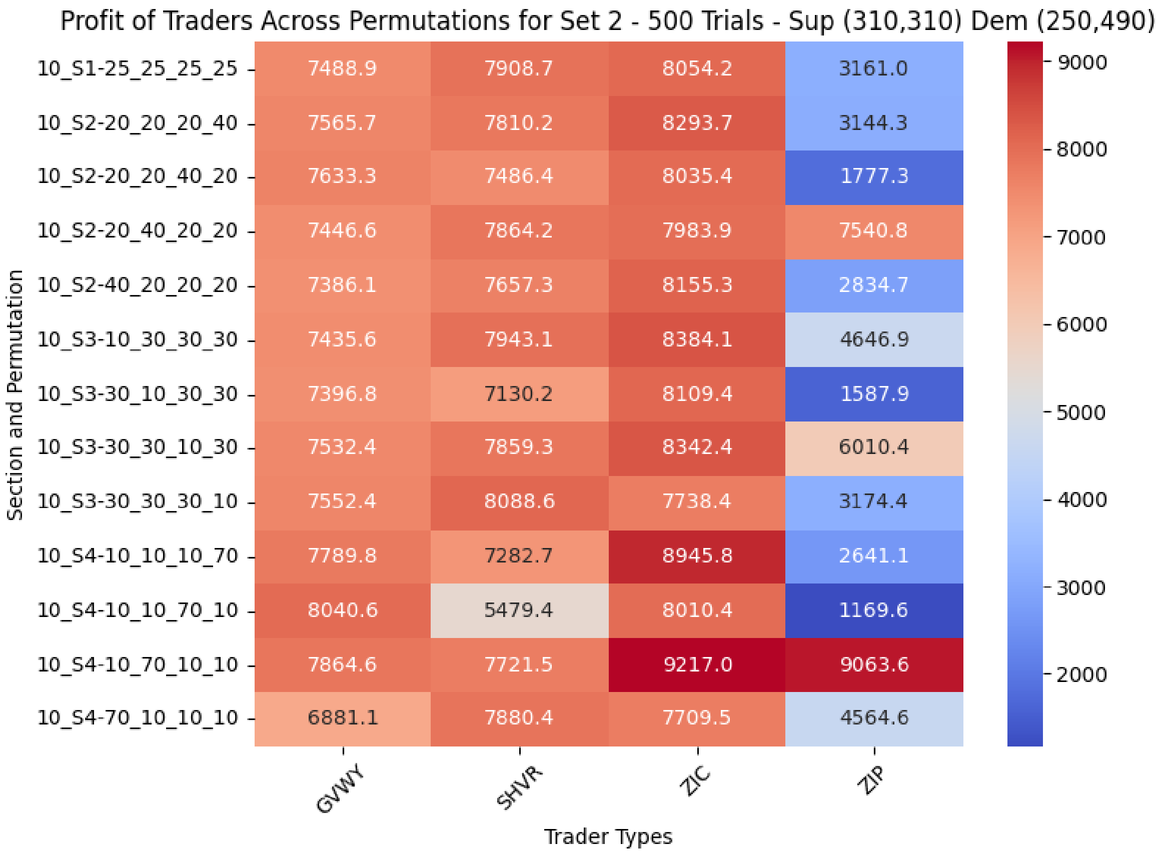
All the above data and tables have used Bonferroni correction due to the large number of comparisons being made. The summary from these results are as follows:

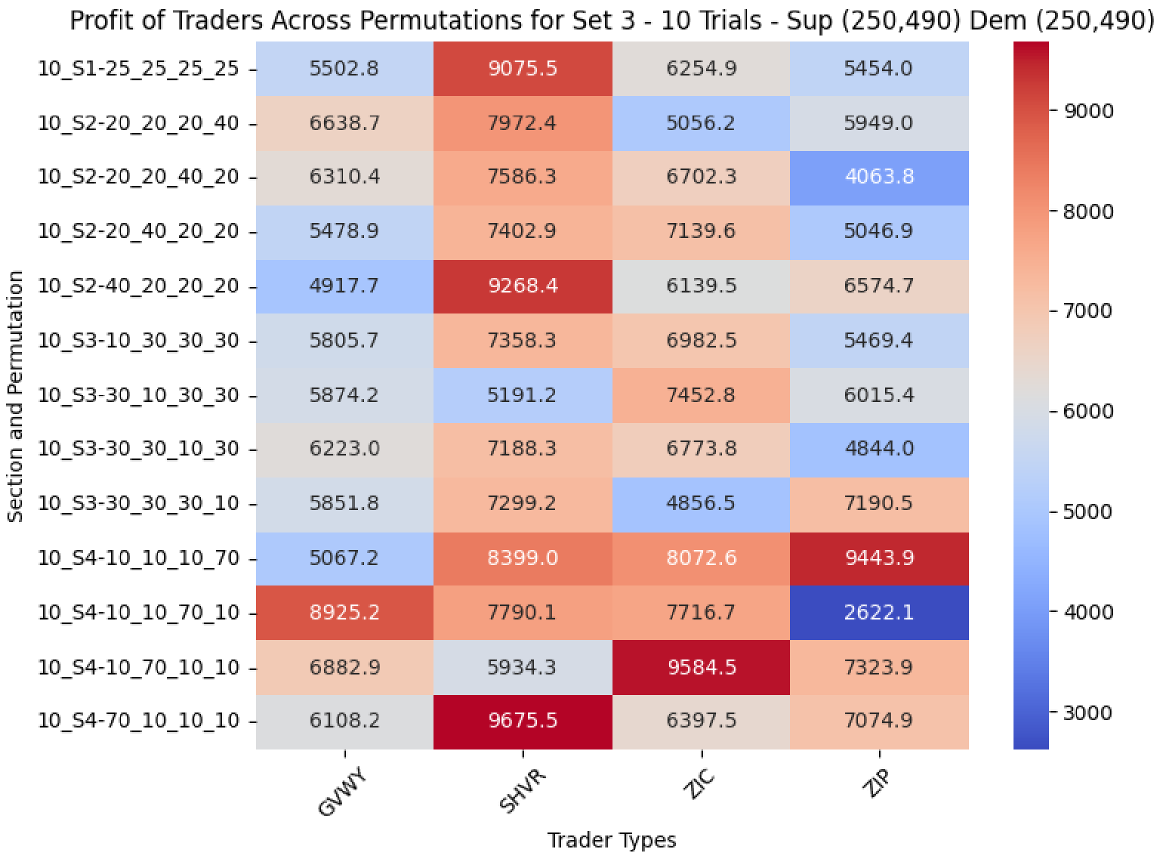
- Ratio (25:25:25:25): No single trading strategy demonstrated a clear, statistically significant advantage over others, suggesting a level playing field amongst them.
- Ratio (40:20:20:20): In permutations with a dominant trader type, the market is impacted in terms of their behaviour.
- Ratio (10:30:30:30): Certain strategies do well/worse depending on the market, where they are statistically advantaged or disadvantaged.
- Ratio (70:10:10:10): The dominant trading strategy is either statistically very profitable or vice versa.

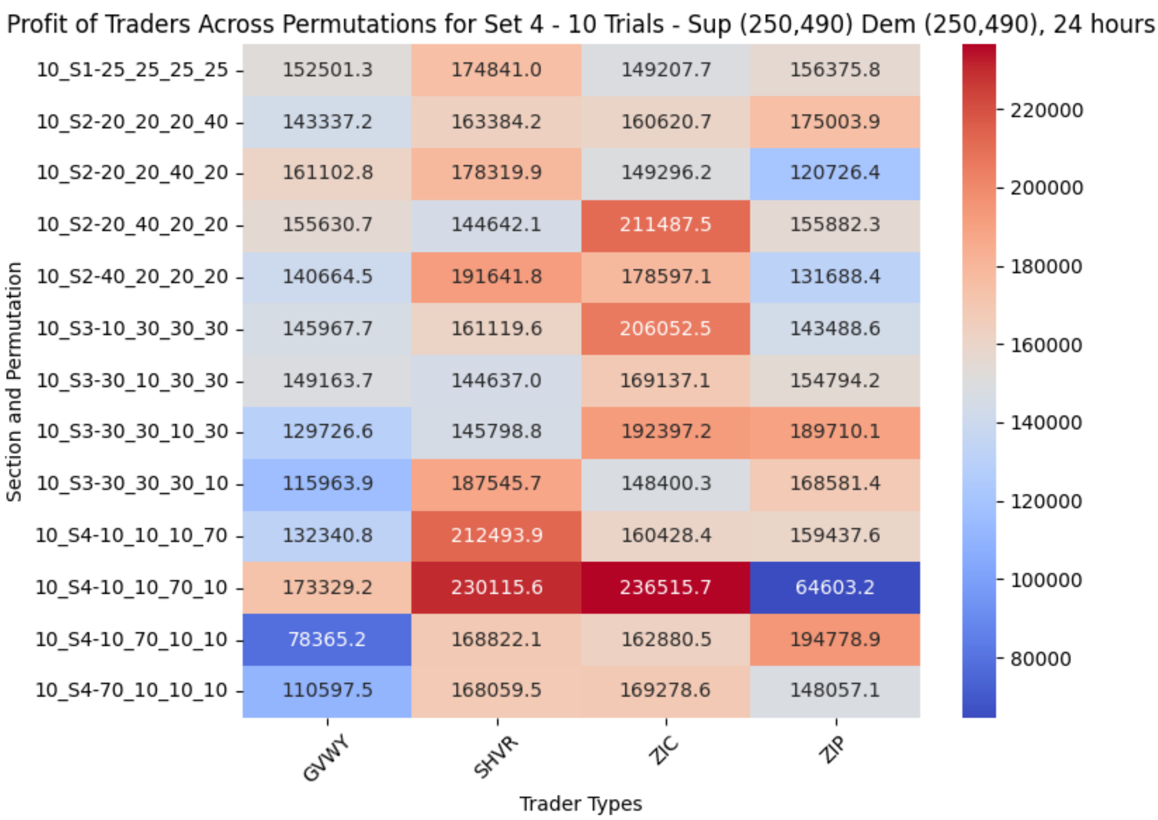
In [ ]:

```
heat = plot_section_profitability_heatmap(section_profitabilities1 , titl
heat = plot_section_profitability_heatmap(section_profitabilities2 , titl
heat = plot_section_profitability_heatmap(section_profitabilities3 , titl
heat = plot_section_profitability_heatmap(section_profitabilities4 , titl
```


Profit of Traders Across Permutations for Set 1 - 10 Trials - Sup (310,310) Dem (250,490)
10_S1-25_25_25_25
8161.5
9104.1
7288.4
3528.1
10_S2-20_20_20_40
7683.8
7874.3
9391.6
3384.7
10_S2-20_20_40_20
8368.5
7793.9
7061.7
2632.8
10_S2-20_40_20_20
7204.0
8215.8
9025.5
3801.9
10_S2-40_20_20_20
8440.9
7125.1
7087.7
5011.0
10_S3-10_30_30_30
7177.9
8009.3
8277.4
3739.1
10_S3-30_10_30_30
9067.3
8260.8
7760.1
2767.2
10_S3-30_30_10_30
7533.5
9306.4
8157.7
3175.2
10_S3-30_30_30_10
7681.9
8268.9
8836.1
3687.8
10_S4-10_10_10_70
8028.6
8513.6
8367.0
3575.2
10_S4-10_10_70_10
5865.1
7240.9
7886.1
1005.2
10_S4-10_70_10_10
7846.1
7434.6
11290.5
6646.3
10_S4-70_10_10_10
6588.2
8398.3
7824.2
7368.2
Section and Permutation
GVWY
SHVR
ZIC
ZIP
Trader Types
10000
8000
6000
4000
2000


Profit of Traders Across Permutations for Set 2 - 500 Trials - Sup (310,310) Dem (250,490)
10_S1-25_25_25_25
7488.9
7908.7
8054.2
3161.0
10_S2-20_20_20_40
7565.7
7810.2
8293.7
3144.3
10_S2-20_20_40_20
7633.3
7486.4
8035.4
1777.3
10_S2-20_40_20_20
7446.6
7864.2
7983.9
7540.8
10_S2-40_20_20_20
7386.1
7657.3
8155.3
2834.7
10_S3-10_30_30_30
7435.6
7943.1
8384.1
4646.9
10_S3-30_10_30_30
7396.8
7130.2
8109.4
1587.9
10_S3-30_30_10_30
7532.4
7859.3
8342.4
6010.4
10_S3-30_30_30_10
7552.4
8088.6
7738.4
3174.4
10_S4-10_10_10_70
7789.8
7282.7
8945.8
2641.1
10_S4-10_10_70_10
8040.6
5479.4
8010.4
1169.6
10_S4-10_70_10_10
7864.6
7721.5
9217.0
9063.6
10_S4-70_10_10_10
6881.1
7880.4
7709.5
4564.6
Section and Permutation
GVWY
SHVR
ZIC
ZIP
Trader Types
9000
8000
7000
6000
5000
4000
3000
2000



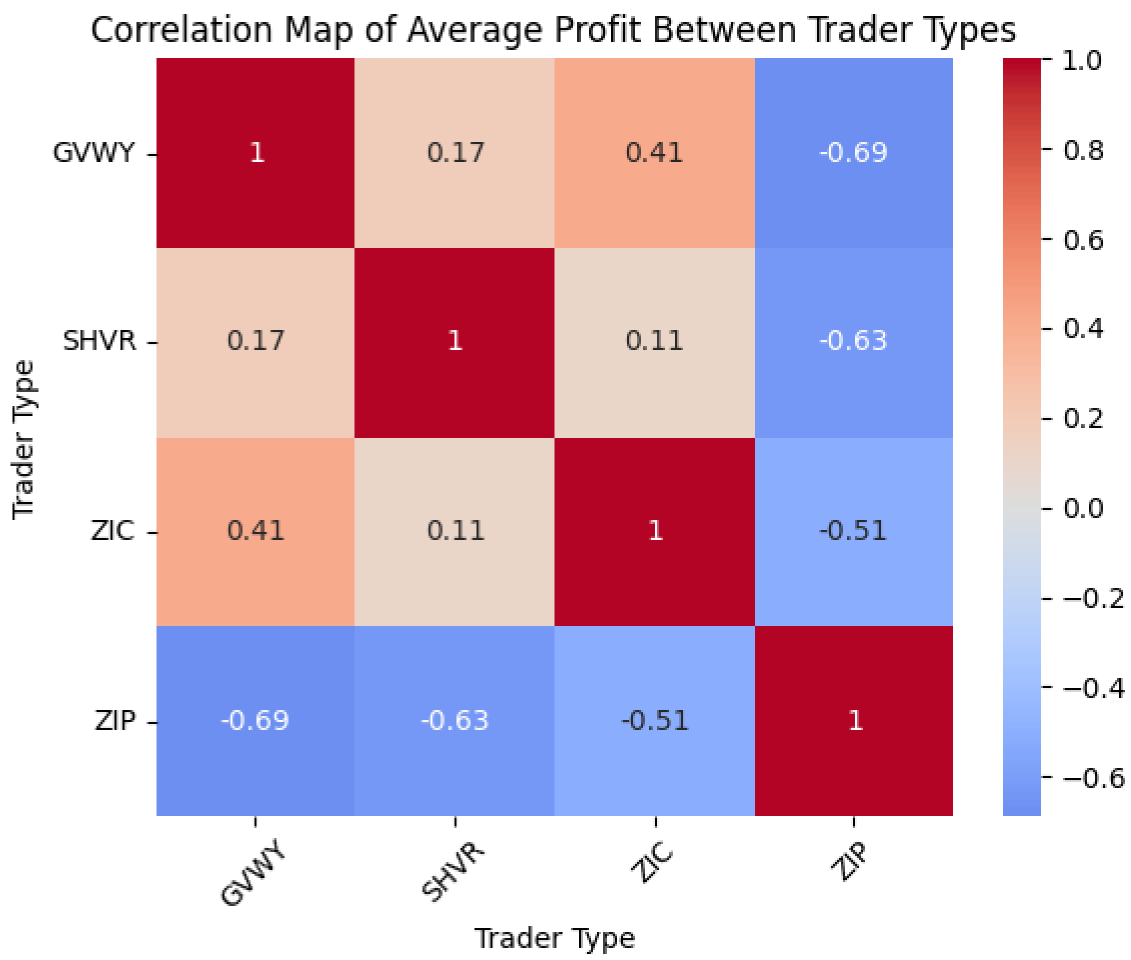


Looking at the heat maps generated from the data we can learn a lot more about the data.

Clearly in Set 1,2 the zero intelligence traders GVWY, SHVR and ZIC tend to perform very well as compared to ZIP. This is possible because they are taking advantage of market imperfections. In comparison, ZIP either performs very well or very bad.

Whereas in Set 3,4 the heat maps are a lot more balanced in terms of general profitability of traders compared to Set 1,2. At the ratio 70:10:10:10, profits are extremely variable due to the impact of the dominant trader. GVWY and ZIP both tend to do worse in these conditions, because they tend to trade around the equilibrium, whereas the others don't. SHVR shows the highest profit overall in these cases as shown by the deep red in the plots. ZIP tends to perform the worst when it is the most dominant trader in the market.

In [ ]:
```
pivot_df = heat.pivot_table(index=['Section', 'Permutation'], columns='Tr

correlation_matrix = pivot_df.corr()

plt.figure(figsize=(6, 5))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Map of Average Profit Between Trader Types')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```



The correlation matrix of these competing strategies give us the following insights:

- GVWY is mostly unaffected by all other traders, with slight negative to ZIC, and positive to SHVR .
- SHVR is again mostly unaffeted by other traders, with slight negative to ZIC and positive to GVWY.

- ZIC shows negative correlation with GVWY, SHVR but positive correlation with ZIP.
- ZIP shows positive correlation ZIC, and is unaffected by others

This does not indicate strong correlations between the profits of any two trader types. The most notable correlation being between ZIP and ZIC where their profits follow each other.

Conclusion

The statistical significance of differences in profitability are heavily dependent on the market conditions that these traders are in. From the four tests, the biggest takeaway being that, the presence of a dominant trading strategy in the market exerts a significant influence on overall market behaviour.

PART D

D.1

In this section, we replicate the experiment in the coursework brief:

- 20 traders - 10 ZIC sellers, 9 ZIC and one ZIPSH buyer
- Sup(50,75),Dem(125,150)
- 30 Days across 50 trials

The initial hyperaparameters of ZIPSH were set very low to give us a clear indication of its improvement.

In [ ]:
```
#All necessary functions for Part C code chunks in the helper.py file
from itertools import permutations
from BSE import *
from helper import *
```

In [ ]:
```
start_time = 0
end_time = 60 * 60 * 24 * 30

num = 20
num_ZIPSH = 1
num_ZIC = num - num_ZIPSH

sellers_spec = [('ZIC', num)]
buyers_spec = [('ZIC', num_ZIC ),('ZIPSH', num_ZIPSH, {'k': 4})]

traders_spec = {'sellers':sellers_spec, 'buyers':buyers_spec}

sup_range = (50,75)
dem_range = (125,150)

dump_all = True
verbose = False
```

```
supply_schedule = [{'from': start_time, 'to': end_time, 'ranges': [sup_ra
demand_schedule = [{'from': start_time, 'to': end_time, 'ranges': [dem_ra

order_interval = 15
order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
               'interval': order_interval, 'timemode': 'periodic'}

dump_flags = {'dump_blotters': True, 'dump_lobs': True, 'dump_strats': Tr
              'dump_avgbals': True, 'dump_tape': True}
```

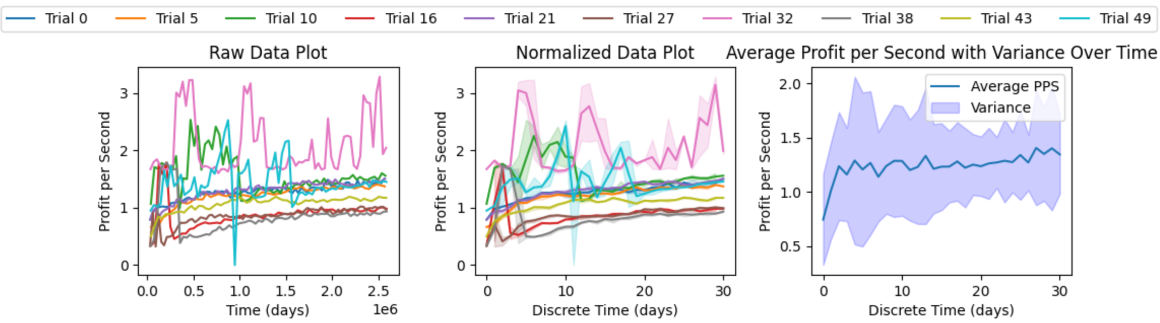
In [ ]:
```
#***DO NOT RUN*** - this simulates the market
output_folder = 'part_d1_50_30D'
os.makedirs(output_folder, exist_ok=True)

for i in range(50):
    trial_id = "part_d1_50_30D/" + "trial_"+ str(i)
    output = market_session(trial_id, start_time, end_time, traders_spec,
```

In [ ]:
```
def adjust_for_resets(data):
    adjusted_data = []
    last_non_reset_value = 0
    total_offset = 0

    for value in data:
        if value < last_non_reset_value:  # Reset detected
            total_offset += last_non_reset_value
        adjusted_value = value + total_offset
        adjusted_data.append(adjusted_value)
        last_non_reset_value = value

    return adjusted_data

# Assuming profit_by_zipsh and profit_by_zic are populated correctly

# Convert the lists to DataFrames for easy manipulation
df_zipsh = pd.DataFrame(profit_by_zipsh).transpose()
df_zipsh.columns = [f'Trial {i+1} ZIPSH' for i in range(5)]

df_zic = pd.DataFrame(profit_by_zic).transpose()
df_zic.columns = [f'Trial {i+1} ZIC' for i in range(5)]

# Apply the adjustment function to each DataFrame
for column in df_zipsh.columns:
    df_zipsh[column] = adjust_for_resets(df_zipsh[column])

for column in df_zic.columns:
    df_zic[column] = adjust_for_resets(df_zic[column])

# Combine both DataFrames
combined_df = pd.concat([df_zipsh, df_zic], axis=1)

# Save to CSV
combined_df.to_csv('profit_results.csv', index=False)

# Plotting
plt.figure(figsize=(15, 6))

# Plot for ZIPSH
for column in df_zipsh.columns:
```

```
    plt.plot(df_zipsh[column], marker='', label=column, linestyle=':')

# Plot for ZIC
for column in df_zic.columns:
    plt.plot(df_zic[column], marker='', label=column)

plt.title('Profit Comparison by Trader Type and Trial')
plt.xlabel('Data Points')
plt.ylabel('Profit')
plt.legend()
plt.show()
```

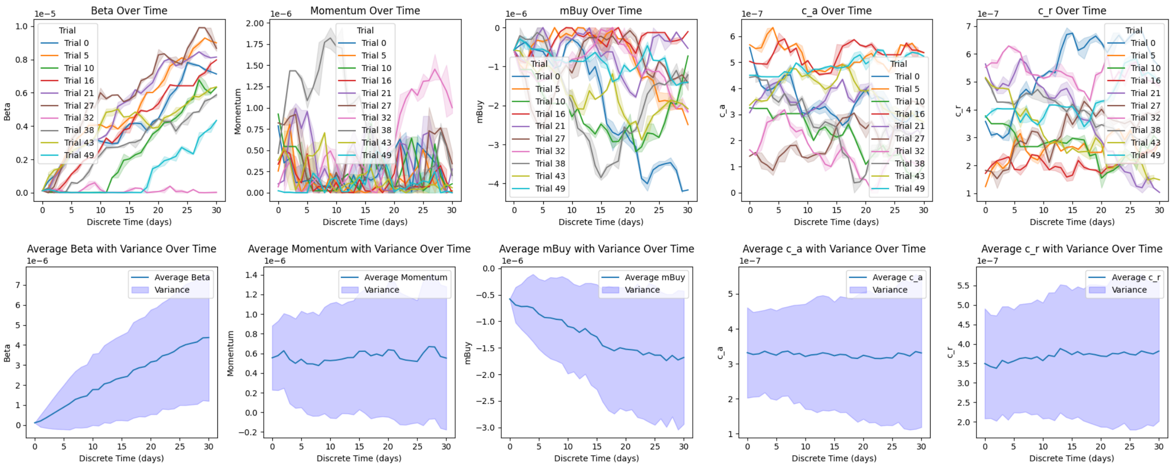
From the above graph, we notice that ZIC tends to outperform ZIPSH overall across the trials. The ZIPSH lines have slight deviations where the strategy is getting mutated. By plotting its profit per second we understand this phenomenon better.

In [ ]:
```
file_pattern = "part_d1_50_30D_1/trial_{trial}_strats.csv"
trials = range(50)
number_of_days = 30
cmap = plt.get_cmap('viridis')
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)


normalize_time(data, number_of_days)
plot_data(data, colors)
```



The above plots show us the improvement of the ZIPSH trader over time. The raw data plot is noisy and the points aren't on the same timescale because ZIPSH might mutate it strategy at different times in the day. To better understand what is going on, we normalise these points into profit per second (PPS) at each day. We can then combine the values across the 50 trials to see the general upward trend for the ZIPSH trader. It is clear from the last plot that the trader is improving over time, moving from around 0.8 to 1.5 from Day 1 to Day 30. That is a 87.5% increase. Note that the variance is quite high, so in reality for a specific trial this number could vary significantly.

In [ ]:
```
param_data = read_data_param(trials, file_pattern)
normalize_time(param_data, number_of_days)
plot_params_data(param_data)
plot_params_data_avg(param_data)
```



There is also a clear change in the hyperparameters as the PPS increases. We see that as ZIPSH does better beta, momentum and c_r tend to increase while the initial buyer margin tends to decrease. The c_a tends to stay around the same. It is also evident that the variance is very large in these cases as the trader is trying to improve.

Now let us test this statistically, comparing Day 1 to Day 30 across the 50 trials.

In [ ]:
```
trials = data['Trial'].unique()

trial_dict = {}

for trial in trials:
    trial_data = data[data['Trial'] == trial]

    first_row = trial_data.iloc[0]
    last_row = trial_data.iloc[-1]


    trial_dict[trial] = {'first': first_row.to_dict(), 'last': last_row.t


first_values = [trial_dict[trial]['first']['Profit per Second'] for trial
last_values = [trial_dict[trial]['last']['Profit per Second'] for trial i

t_stat, p_val, null_hypothesis_finding, alternative_hypothesis_finding =

p_val_full = "{:.2e}".format(p_val)
```

```
ttest_results_df = pd.DataFrame({
    'Metric': ['t-statistic', 'p-value', 'Null Hypothesis', 'Alternative
    'Value': [t_stat, p_val_full, null_hypothesis_finding, alternative_hy
})

ttest_results_df
```

Out[ ]:

	Metric	Value
0	t-statistic	17.581582
1	p-value	4.17e-23
2	Null Hypothesis	Reject H_0. Significant difference in means.
3	Alternative Hypothesis	Evidence supports the alternative hypothesis.

The above paired one tailed t-test very strongly suggests that there is a significant difference in the means of the last values (Day 30) and the first value (Day 1) across the 50 trials. So, we can conclude that the ZIPSH trader has a statistically significant improvement in performance from Day 1 to Day 30.

D.2

Below we test new market sessions where the changes from the previous experiment are as follows-

- Set 1 - Run for 5 trials of 1, 2, 7, 14 days
- Set 2 - Run for 10 trials of 60 days
- Set 3 - Run with the same hyperparameters as the original Cliff 1997 paper
- Set 4 - Run with 20 ZIPSH traders amongst each other

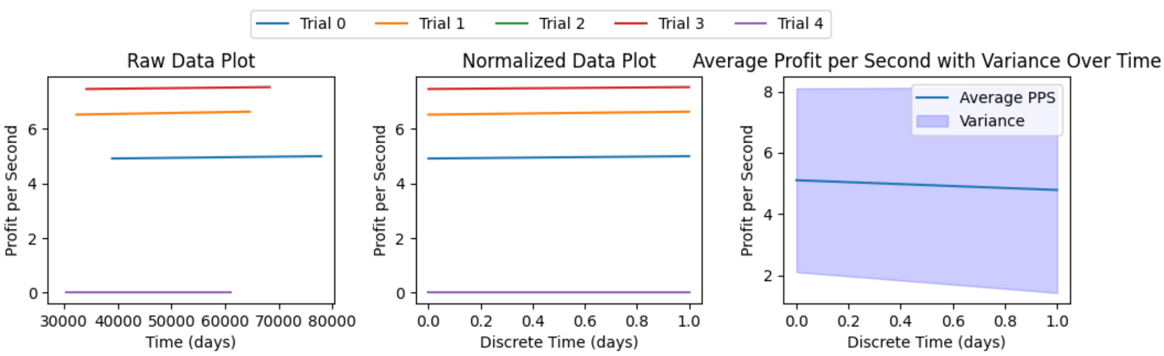
We will only compare PPS(ProfitPerSecond) for consistency.

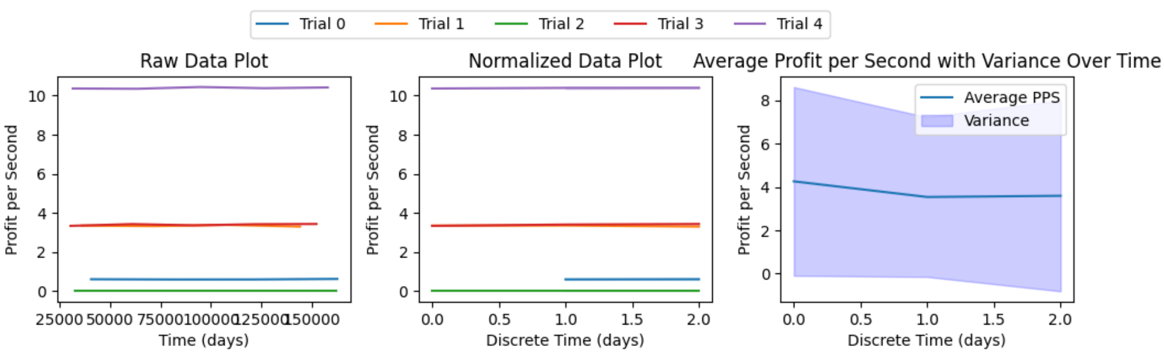
Set 1

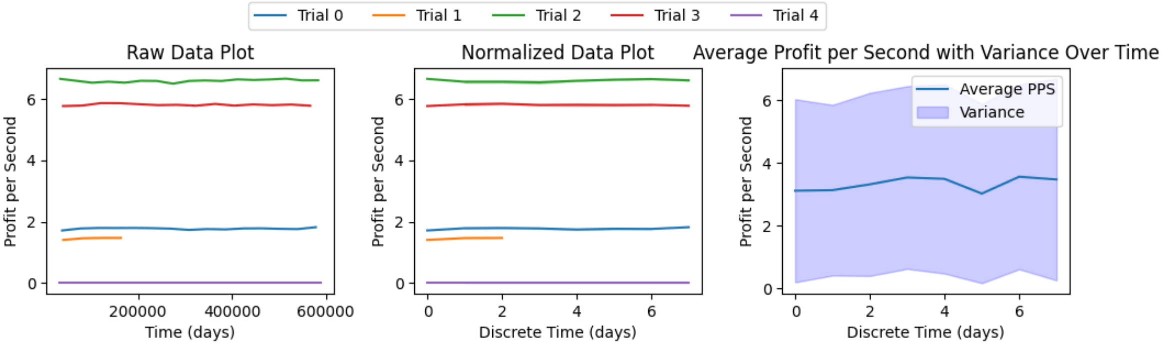
In [ ]:

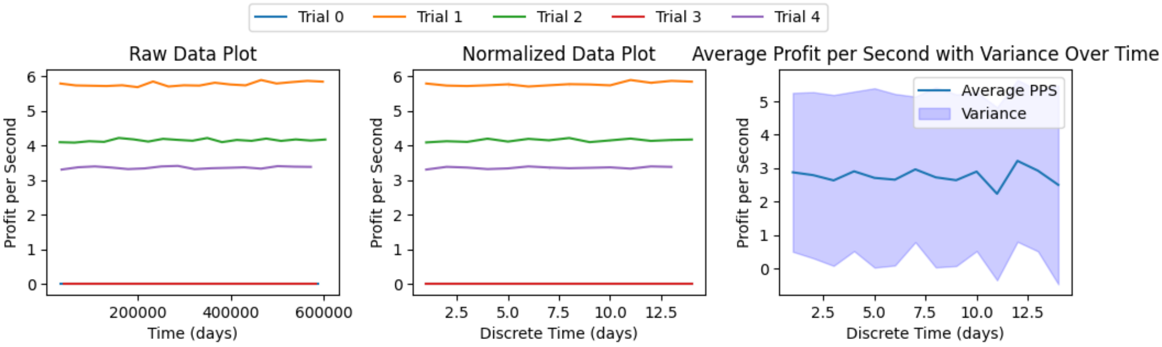
```
#Set 1 - 1 Day

file_pattern = "part_d1_5_1D/trial_{trial}_strats.csv"
trials = range(5)
number_of_days = 1
cmap = plt.get_cmap('viridis')
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)

normalize_time(data, number_of_days)
plot_data(data, colors)

#Set 1 - 2 days

file_pattern = "part_d1_5_2D/trial_{trial}_strats.csv"
trials = range(5)
number_of_days = 2
cmap = plt.get_cmap('viridis')
```

```
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)

normalize_time(data, number_of_days)
plot_data(data, colors)

#Set 1 - 7 days

file_pattern = "part_d1_5_7D/trial_{trial}_strats.csv"
trials = range(5)
number_of_days = 7
cmap = plt.get_cmap('viridis')
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)

normalize_time(data, number_of_days)
plot_data(data, colors)

#Set 1 - 14 days

file_pattern = "part_d1_5_14D/trial_{trial}_strats.csv"
trials = range(5)
number_of_days = 14
cmap = plt.get_cmap('viridis')
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)

normalize_time(data, number_of_days)
plot_data(data, colors)
```





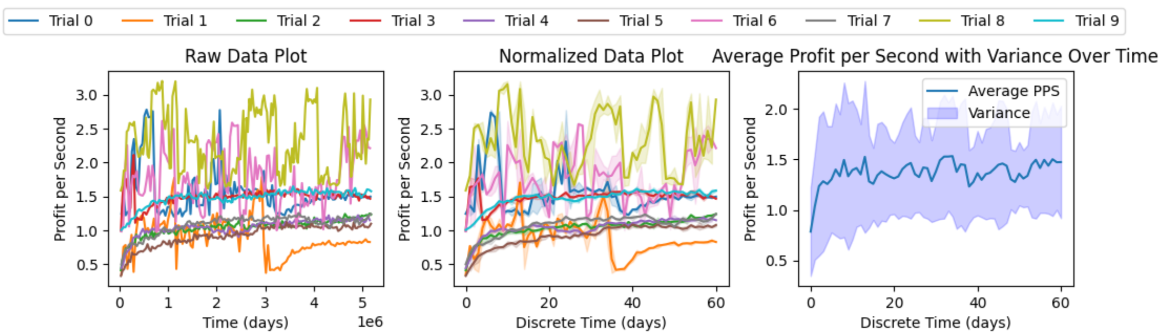




The above plots for Set 1 tell us that 1, 2, 7 and 14 days are just not enough data to see any meaningful difference in the PPS of the ZIPSH trader. So, we will not be looking at Set 1 from here.

Set 2

In [ ]:
```
file_pattern = "part_d1_10_60D_1/trial_{trial}_strats.csv"
trials = range(10)
number_of_days = 60
cmap = plt.get_cmap('viridis')
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)

normalize_time(data, number_of_days)
plot_data(data, colors)

day_ranges = [(10, 15), (55, 60)]
ttest_results_df = perform_ttests_for_day_ranges(data, day_ranges)
ttest_results_df
```



Out[ ]:

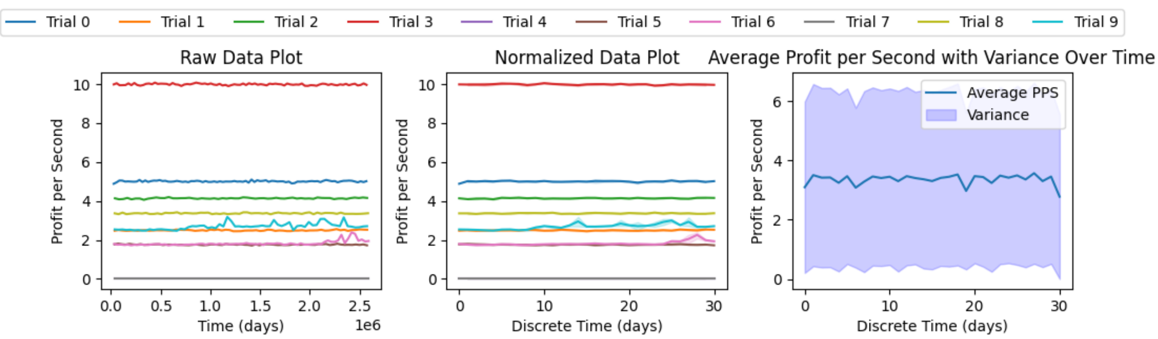
	Comparison	Test Used	Statistic	p-value	Null Hypothesis	Alternative Hypothesis
0	Range of Days (55, 60) vs Range of Days(10, 15)	t-test	1.567511	7.57e-02	Failed to reject H_0	Not enough evidence

There is a clear increase in the PPS of the trader, moving from around 0.75 to 1.5, which is twice its original value. It is important to note here that the PPS doesn't improve much after around Day 10. This might mean that once the ZIPSH trader has optimised to a certain extent, no matter the updates it makes, its rate of progress declines. This is supported by the statistical test. We have taken an average of the PPS across 5 days from Day 10-15 and Day 55-60 and their isn't enough statistical evidence to suggest that the trader does improve in performance.

Set 3

In [ ]:

```
#Sup(310,310) Dem(250,490)

file_pattern = "part_d1_50_30D/trial_{trial}_strats.csv"
trials = range(10)
number_of_days = 30
cmap = plt.get_cmap('viridis')
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)

normalize_time(data, number_of_days)
plot_data(data, colors)

day_ranges = [(1, 1), (30, 30)]
ttest_results_df = perform_ttests_for_day_ranges(data, day_ranges)
ttest_results_df
```



Out[ ]:

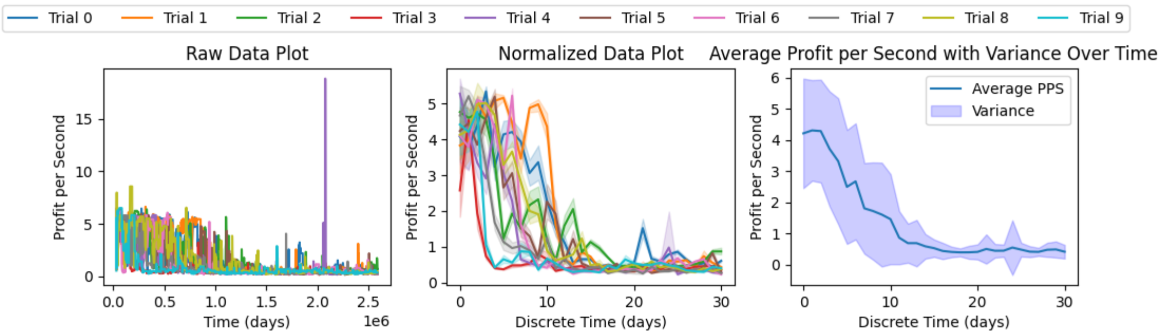
	Comparison	Test Used	Statistic	p-value	Null Hypothesis	Alternative Hypothesis
0	Range of Days (30, 30) vs Range of Days(1, 1)	t-test	1.403635	9.70e-02	Failed to reject H_0	Not enough evidence

In this market the ZIPSH trader doesn't seem to make an improvement at all. As Cliff 1997 notes, these are likely already effective hyperparameters. Since it is stochastic in nature the PPS does fluctuate quite a lot, but it is pointless at the end of the 30 days as there is no overall improvement. This is also statistically supported by a one tailed t-test to check if it performed better at Day 30.

Set 4

In [ ]:

```
file_pattern = "part_d1_10_30D_2_ZIPSHONLY/trial_{trial}_strats.csv"
trials = range(10)
number_of_days = 30
cmap = plt.get_cmap('viridis')
colors = cmap(np.linspace(0, 1, len(trials)))

data = read_data_prof(trials, file_pattern)

normalize_time(data, number_of_days)
plot_data(data, colors)

day_ranges = [(1, 1), (30, 30)]
ttest_results_df = perform_ttests_for_day_ranges(data, day_ranges, altern
ttest_results_df
```



Out[ ]:

	Comparison	Test Used	Statistic	p-value	Null Hypothesis	Alternative Hypothesis
0	Range of Days (30, 30) vs Range of Days(1, 1)	Wilcoxon	0.0	9.77e-04	Reject H_0	Evidence supports alternative

The above plots tell a very interesting story. Clearly from the figures we notice that the PPS of the ZIPSH trader tends to go down over the 30 day period. Supported by the statistical test, this reinforces the notion stated in previous sections that the ZIP traders tend to do very poorly when it is the dominant/only trader in the market. This is likely because all the ZIP traders are being impacted by the behaviours of each other.

Conclusion

The performance of Zero/Minimal Intelligence trading strategies are heavily dependent on the market conditions and competitors they are against. Therefore, there is insufficient evidence to claim one trading strategy is most profitable across all market conditions and regardless of competitors.

END OF REPORT. ONLY WORD COUNT BELOW THIS POINT.

In [ ]:
```
# Do not edit this code. It will print the word count of your notebook.
import io
from nbformat import current

def printWordCount(filepath):

    with io.open(filepath, 'r', encoding='utf-8') as f:
        nb = current.read(f, 'json')

    word_count = 0
    for cell in nb.worksheets[0].cells:
        if cell.cell_type == "markdown":
            word_count += len(cell['source'].replace('#', '').lstrip().sp
    print("Word count: " + str(word_count) + ". Limit is 2000 words.")
```

```
/var/folders/51/y8m0pgmd0txbsy5hlltxny180000gn/T/ipykernel_54921/190210797
9.py:3: DeprecationWarning: nbformat.current is deprecated since before nb
format 3.0

- use nbformat for read/write/validate public API
- use nbformat.vX directly to composing notebooks of a particular version

  from nbformat import current
```

In [ ]:
```
# This should be the final output of your notebook.
# Edit filename to be the same as this filename and then run.
# Save your file before running this code.

this_file_name = "CW-IEFT-Vasudev.ipynb" # Enter name of this file here
printWordCount(this_file_name)
```

```
Word count: 1993. Limit is 2000 words.
```