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

Patrick Lee

```
import matplotlib.pyplot as plt
import numpy as np
import csv
import random
from math import ceil

import scipy as sp
from scipy import stats
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from tqdm import tqdm

from BSE import market_session

from os import listdir
from os.path import isfile, join

# HELPER FUNCTIONS

# FROM W8 LABSHEET
# Use this to run an experiment n times and plot all trades.
def n_runs_plot_trades(n, trial_id, start_time, end_time,
traders_spec, order_sched, dump_flags, verbose, plot=False):
    x = np.empty(0)
    y = np.empty(0)

    for i in tqdm(range(n)):
        trialId = trial_id + '_' + str(i)

        market_session(trialId, start_time, end_time, traders_spec,
order_sched, dump_flags, verbose)

        if plot:
            with open(trialId + '_tape.csv', newline='') as csvfile:
                reader = csv.reader(csvfile)
                for row in reader:
                    time = float(row[1])
                    price = float(row[2])
                    x = np.append(x,time)
                    y = np.append(y,price)

    if plot: plt.plot(x, y, 'x', color='black')

def plot_best_bid_ask(N, trial_id):
```

```python
    x = np.empty(0)
    y = np.empty(0)

    for i in range(N):
        trialId = trial_id + '_' + str(i)
        with open(trialId + '_tape.csv', newline='') as csvfile:
            reader = csv.reader(csvfile)
            for row in reader:
                time = float(row[1])
                bid = float(row[2])
                x = np.append(x,time)
                y = np.append(y,bid)

    plt.plot(x, y, 'x', color='black')


# FROM W8 LABSHEET
# !!! Don't use on it's own
def getorderprice(i, sched, n, mode):
    pmin = min(sched[0][0], sched[0][1])
    pmax = max(sched[0][0], sched[0][1])
    prange = pmax - pmin
    stepsize = prange / (n - 1)
    halfstep = round(stepsize / 2.0)

    if mode == 'fixed':
        orderprice = pmin + int(i * stepsize)
    elif mode == 'jittered':
        orderprice = pmin + int(i * stepsize) + random.randint(-
halfstep, halfstep)
    elif mode == 'random':
        if len(sched) > 1:
            # more than one schedule: choose one equiprobably
            s = random.randint(0, len(sched) - 1)
            pmin = min(sched[s][0], sched[s][1])
            pmax = max(sched[s][0], sched[s][1])
        orderprice = random.randint(pmin, pmax)
    return orderprice


# FROM W8 LABSHEET
# !!! Don't use on it's own
def make_supply_demand_plot(bids, asks):
    # total volume up to current order
    volS = 0
    volB = 0

    fig, ax = plt.subplots()
    plt.ylabel('Price')
    plt.xlabel('Quantity')
```

```
    pr = 0
    for b in bids:
        if pr != 0:
            # vertical line
            ax.plot([volB,volB], [pr,b], 'r-')
        # horizontal lines
        line, = ax.plot([volB,volB+1], [b,b], 'r-')
        volB += 1
        pr = b
    if bids:
        line.set_label('Demand')

    pr = 0
    for s in asks:
        if pr != 0:
            # vertical line
            ax.plot([volS,volS], [pr,s], 'b-')
        # horizontal lines
        line, = ax.plot([volS,volS+1], [s,s], 'b-')
        volS += 1
        pr = s
    if asks:
        line.set_label('Supply')

    if bids or asks:
        plt.legend()
    plt.show()


# FROM W8 LABSHEET
# Use this to plot supply and demand curves from supply and demand
ranges and stepmode
def plot_sup_dem(seller_num, sup_ranges, buyer_num, dem_ranges,
stepmode):
    asks = []
    for s in range(seller_num):
        asks.append(getorderprice(s, sup_ranges, seller_num,
stepmode))
    asks.sort()
    bids = []
    for b in range(buyer_num):
        bids.append(getorderprice(b, dem_ranges, buyer_num, stepmode))
    bids.sort()
    bids.reverse()

    make_supply_demand_plot(bids, asks)


# Forms two dataframes, one containing the final profitability for
```

```
each agent for each session (N rows, no. of agents columns)),
# and one containing the number of wins for each agent (1 row, no. of
agents columns) across each session
def get_final_profitabilities_wins_df(N, R, agents_types, file_app=4):

    csv_files = ['out_data/test_N=' + str(N) + "_R=" + str(R) + '_' +
str(i) + '_avg_balance.csv' for i in range(0, N)]

    wins = {}
    profitabilities = {}
    for agent in agents_types:
        profitabilities[agent] = []
        wins[agent] = 0

    dfs_rows = []
    dfs_cols = {}
    for file in csv_files:
        df_cols = pd.read_csv(file, usecols=range(4 *
len(agents_types) + file_app))
        df_rows = df_cols.copy()

        new_columns = ['testId', 'time', 'bb', 'bo']
        for agent in agents_types:
            new_columns.append(agent)
            new_columns.append(agent + '-tp')
            new_columns.append(agent + '-ta')
            new_columns.append(agent + '-ap')

        df_rows.columns = new_columns
        dfs_rows.append(df_rows)
        dfs_cols[file] = df_cols

        #print(df_rows.columns)
        #print(df_rows.head(5))

        max_profitability = 0
        max_profitability_agent = ''

        for agent in agents_types:
            profitability = df_rows[agent + '-ap'].iloc[-1]
            #print("Agent " + agent + " profitability: " +
str(profitability))
            profitabilities[agent].append(profitability)
            if profitability > max_profitability:
                max_profitability = profitability
                max_profitability_agent = agent

        wins[max_profitability_agent] += 1

    profitability_df = pd.DataFrame(profitabilities)
```

```python
    return profitability_df, wins

## Get useful dataframes from strat csvs
##
## \return    df_strat_comp   a dataframe containing all strat data
across all trials, ordered by time
## \return    df_half_prof    a dataframe containing the profitability
of the active strat at a given time, first and second half of the
strat data
## \return    df_sample_prof  a dataframe containing the profitability
of the active strat at a given time for 4 random trials

def load_strat_data(trial_id, N=100, dir='', remove_outliers=False):
    csvs = []
    if len(dir) > 0:
        csvs  = [dir + f for f in listdir(dir) if isfile(join(dir,
f))]
    else:
        for i in range(0, N):
            csvs.append(trial_id + '_' + str(i) + '_strats.csv')

    dfs_strat = []
    for csv in csvs:
        df_strat = pd.read_csv(csv, header=None,
usecols=[1,3,7,9,11,13,15,17,19,21,23,26,28,30,32,34,36])
        new_columns = ['t', 'id', 'actv_mBuy', 'actv_mSel', 'actv_b',
'actv_m', 'actv_ca', 'actv_cr', 'actvprof', 'best_B_id',
'best_B_prof', 'bstr_mBuy', 'bstr_mSel', 'bstr_b' ,'bstr_m',
'bstr_ca', 'bstr_cr']
        df_strat.columns = new_columns
        dfs_strat.append(df_strat)

    # Create paired half dataframe
    df_half_paired = pd.DataFrame({'First half': [], 'Second half':
[], 'Difference': []})
    for df in dfs_strat:
        first = df['actvprof'].iloc[0:ceil(len(df) / 2)].values
        second = df['actvprof'].iloc[(len(df) // 2):len(df)].values
        for i in range(0, ceil(len(df) / 2)):
            new_row = {'First half': first[i], 'Second half':
second[i], 'Difference': second[i] - first[i]}
            df_half_paired= pd.concat([df_half_paired,
pd.DataFrame(new_row, index=[0])], ignore_index=True)

    # Create combinatory dataframe
    df_strat_comp = pd.concat(dfs_strat)
    df_strat_comp = df_strat_comp.sort_values(by=['t'])

    #Create quart dataframe
```

```
    q_size = ceil(len(df_strat_comp) / 4)
    def lb(i):
        return (i * len(df_strat_comp) // 4)
    df_quart_prof = pd.DataFrame({
        'Q1':
df_strat_comp['actvprof'].iloc[lb(0):lb(0)+q_size].values,
        'Q2':
df_strat_comp['actvprof'].iloc[lb(1):lb(1)+q_size].values,
        'Q3':
df_strat_comp['actvprof'].iloc[lb(2):lb(2)+q_size].values,
        'Q4':
df_strat_comp['actvprof'].iloc[lb(3):lb(3)+q_size].values,
    })

    if remove_outliers:
        df_quart_prof =
df_quart_prof[(np.abs(stats.zscore(df_quart_prof)) < 2).all(axis=1)]



    sample_indicies = random.sample(range(0, N), 4*3)

    df_sample_hyp_prof_arr = []
    for i in range(0, 3):
        df_sample_prof = pd.DataFrame({'Sample': [], 't': [],
'actvprof': []})
        for j in range(i*4, (i+1)*4):
            for index, row in
dfs_strat[sample_indicies[j]].iterrows():
                new_row = {
                    'Sample': sample_indicies[j],
                    't': row['t'],
                    'actv_b': row['actv_b'],
                    'actv_m': row['actv_m'],
                    'actv_ca': row['actv_ca'],
                    'actv_cr': row['actv_cr'],
                    'actvprof': row['actvprof']
                    }
                df_sample_prof = pd.concat([df_sample_prof ,
pd.DataFrame(new_row, index=[0])], ignore_index=True)
        df_sample_hyp_prof_arr.append(df_sample_prof)


    return df_strat_comp, df_half_paired, df_quart_prof,
df_sample_hyp_prof_arr


# FROM W5 LABSHEET
# Wrapper function for performing Shapiro-Wilk test on dataframe.
def shapiro_wilk(df):
```

```
    # Shapiro-Wilk test tests the null hypothesis that
    # the data was drawn from a normal distribution

    print("Using Shapiro-Wilk test to test the null hypothesis " +
        "that the data was drawn from a normal distribution:")

    for col in df.columns:
        _, pvalue = stats.shapiro(df[col])
        if pvalue < 0.05:
            print("Condition " + "{:}".format(col) +
                ". We can reject the null hypothesis (p=" +
                "{:.2f}".format(pvalue) +
                "). Therefore, profitability for " + col + " is not
normally distributed.")
        else:
            print("Condition " + "{:}".format(col) +
                ". We cannot reject the null hypothesis (p=" +
                "{:.2f}".format(pvalue) +
                "). Therefore, profitability for " + col + " is
normally distributed.")


# FROM W5 LABSHEET
# Wrapper function for performing Kolmogorov-Smirnov test on
dataframe.
def kolmogorov_smirnov(df):

    print("Using Kolmogorov-Smirnov test to test the null hypothesis "
+
        "that the data was drawn from a normal distribution:")

    for col in df.columns:
        # Normalise data
        norm_col = (df[col] - df[col].mean())/df[col].std()
        _, pvalue = stats.kstest(norm_col, 'norm')
        if pvalue < 0.05:
            print("Condition " + "{:}".format(col) +
                ". We can reject the null hypothesis (p=" +
                "{:.2f}".format(pvalue) +
                "). Therefore, profitability for " + col + " is not
normally distributed.")
        else:
            print("Condition " + "{:}".format(col) +
                ". We cannot reject the null hypothesis (p=" +
                "{:.2f}".format(pvalue) +
                "). Therefore, profitability for " + col + " is
normally distributed.")


# FROM W5 LABSHEET
```

```
# Wrapper function for performing t-test on dataframe.
def t_test(df, col1, col2):
    # T-test to test the hypothesis that the two sets of profitability
sample are from the same distribution
    # i.e. SHVR and ZIC have the same profitability

    print("Using t-test to test the null hypothesis " +
        "that the two profitability samples are from the same
distribution:")

    result = stats.ttest_ind(df[col1], df[col2])
    if result.pvalue < 0.05:
        print("Condition " + "{:}".format(col1) +
                ". We can reject the null hypothesis (p=" +
                "{:.2f}".format(result.pvalue) +
                "). Therefore, SHVR and ZIC have statistically
different profitabilities")
    else:
        print("Condition " + "{:}".format(col2) +
                ". We cannot reject the null hypothesis (p=" +
                "{:.2f}".format(result.pvalue) +
                "). Therefore, profitability of SHVR and ZIC is
statistically indistinguishable")


# FROM W5 LABSHEET
# Wrapper function for performing Mann-Whitney-U test on dataframe.
def mann_whitney_u_test(df, col1, col2):
    # T-test to test the hypothesis that the two sets of profitability
sample are from the same distribution
    # i.e. SHVR and ZIC have the same profitability

    print("Using Mann-Whitney-U test to test the null hypothesis " +
        "that the two profitability samples are from the same
distribution:")

    result = stats.mannwhitneyu(df[col1], df[col2])
    if result.pvalue < 0.05:
        print("Condition " + "{:}".format(col1) +
                ". We can reject the null hypothesis (p=" +
                "{:.8f}".format(result.pvalue) +
                "). Therefore, " + col1  + " and " + col2 + " have
statistically different profitabilities")
    else:
        print("Condition " + "{:}".format(col2) +
                ". We cannot reject the null hypothesis (p=" +
                "{:.8f}".format(result.pvalue) +
                "). Therefore, profitability of "  + col1  + " and " +
col2 + " is statistically indistinguishable")
```

```
    print("\n")


# FROM W5 LABSHEET
# Wrapper function for performing Wilcoxon Signed Rank test given the
differences between the two sets
def wilcoxon_signed_rank_test(ranks, col1, col2):

    print("Using the Wilcoxon signed rank test")

    result = stats.wilcoxon(ranks, alternative='greater')
    if result.pvalue < 0.05:
        print("Condition " + "{:}".format(col1) +
                ". We can reject the null hypothesis (p=" +
                "{:.8f}".format(result.pvalue) +
                "). Therefore, " + col1  + " and " + col2 + " have
statistically different profitabilities")
    else:
        print("Condition " + "{:}".format(col2) +
                ". We cannot reject the null hypothesis (p=" +
                "{:.8f}".format(result.pvalue) +
                "). Therefore, profitability of "  + col1  + " and " +
col2 + " is statistically indistinguishable")

    print("\n")

# FROM W5 LABSHEET
# Wrapper function for performing Kruskal-Wallis on dataframe.
def kruskal_wallis_test(df, cols):

    print("Using Kruskal-Wallis test to test the null hypothesis " +
        "that the profitability samples are from the same
distribution:")

    samples = (df[col] for col in cols)

    result = stats.kruskal(*samples)
    if result.pvalue < 0.05:
        print("Condition " + "{:}".format(','.join(cols)) +
                ". We can reject the null hypothesis (p=" +
                "{:.8f}".format(result.pvalue) +
                "). Therefore, " + ",".join(cols) + " have
statistically different profitabilities")
    else:
        print("Condition " + "{:}".format(','.join(cols)) +
                ". We cannot reject the null hypothesis (p=" +
                "{:.8f}".format(result.pvalue) +
                "). Therefore, profitability of "  + ",".join(cols) +
" is statistically indistinguishable")
    print("\n")
```

PART A

Here we aim to compare the profitability of ZIC and SHVR traders in a market consiting only of these two trader types.

First we set up the experiment with supply-demand as described in [1]

```
# Reusable setup function for A-C
def setup(start_time, end_time, buyers_spec, sellers_spec):

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

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

    demand_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[dem_range], 'stepmode': 'fixed'}]
    supply_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[sup_range], 'stepmode': 'fixed'}]

    order_interval = 60
    order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
                'interval': order_interval, 'timemode': 'periodic'}

    return traders_spec, order_sched
```

Now setting up the traders evenly split between SHVR and ZIC for 50 IIDs.

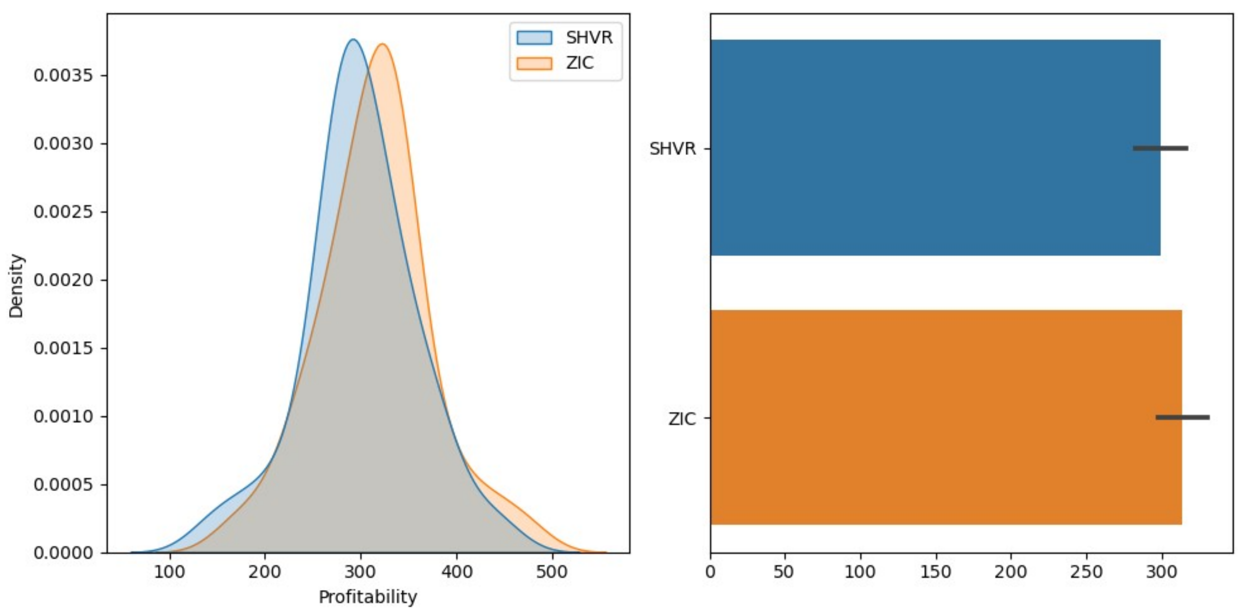
```
N = 50
R = 50
start_time = 0
end_time = 60 * 10

num_buyers = 20
num_sellers = 20

buyers_spec = [('SHVR', int(num_buyers * R/100)), ('ZIC',
int(num_buyers * (100-R)/100))]
sellers_spec = [('SHVR', int(num_sellers * R/100)), ('ZIC',
int(num_sellers * (100-R)/100))]

trial_id = 'test_N=' + str(N) + "_R=" + str(R)
traders_spec, order_sched = setup(start_time, end_time, buyers_spec,
sellers_spec)

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

verbose = False
```

```
n_runs_plot_trades(N, trial_id, start_time, end_time, traders_spec,
order_sched, dump_flags, verbose)
```

```
e

100%|██████████| 50/50 [00:07<00:00,  6.38it/s]
```

Now we have the avg. profitabilities per trader for SHVR and ZIC at the end of each trading session, we can plot their respective distributions

```
profitability_df, wins = get_final_profitabilities_wins_df(N=50, R=50,
agents_types=['SHVR', 'ZIC'])

plt.rcParams["figure.figsize"] = [10.0, 5.0]
plt.rcParams["figure.autolayout"] = True
f, axes = plt.subplots(1, 2)
sns.kdeplot(data=profitability_df, fill=True, ax=axes[0])
sns.barplot(profitability_df, ax=axes[1], orient='h')

axes[0].set_xlabel("Profitability")

plt.show()
```



The difference in location between the two distributions suggests that the ZIC agents maintain a higher profitability on average and both distributions resemble Normal distributions. We can verifiy this normality, which in turn will influence our choice of overall hypothesis test, by using the Shapiro-Wilk or the Kolmogorov-Smirnov test. It is generally recommended to use the former for sample sizes of (n < 50), due to it's higher power of detecting non-normality [2]. As we are only just at this threshold with 50 trading runs, we shall use the Shapiro-Wilk test.

```
shapiro_wilk(profitability_df)
```

```
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.36).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.37).
Therefore, profitability for ZIC is normally distributed.
```

Now we've established normality, we can use a parametric test to determine if the two profitability samples are from the same distribution; if we reject this hypothesis, we know that one of the agents is more profitable. We can use a t-test due to its power with smaller sample sizes.

```
t_test(profitability_df, 'SHVR', 'ZIC')
```

```
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.22).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable
```

Now repeating the experiment with N=500

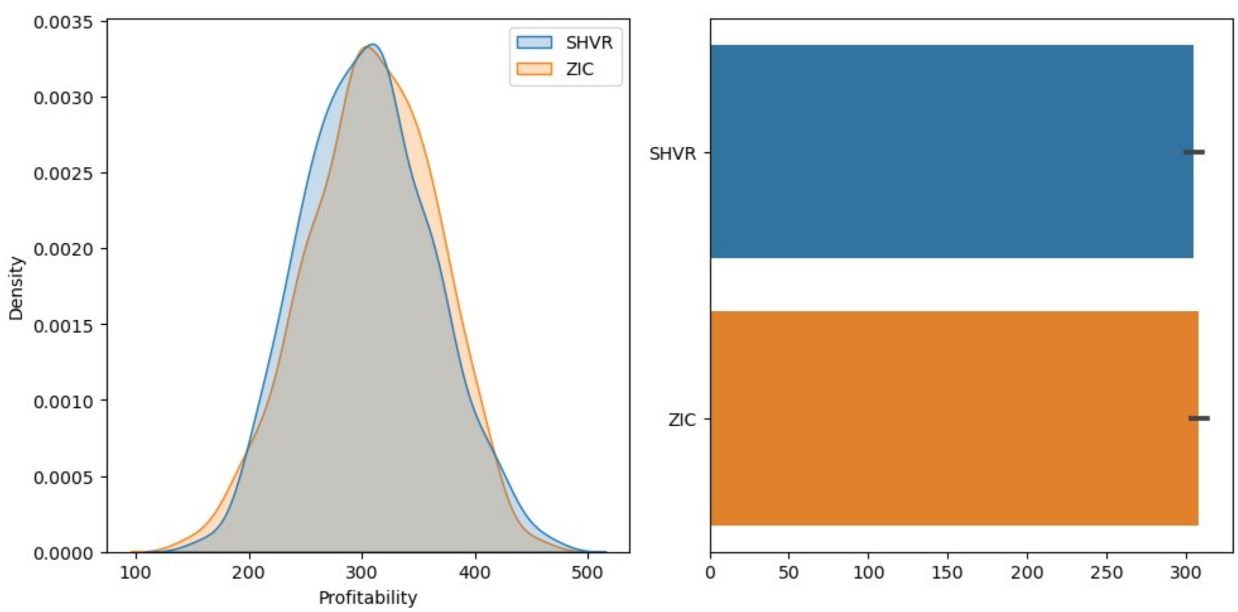
```
N = 500
R = 50
start_time = 0
end_time = 60 * 10

num_buyers = 20
num_sellers = 20

buyers_spec = [('SHVR', int(num_buyers * R/100)), ('ZIC',
int(num_buyers * (100-R)/100))]
sellers_spec = [('SHVR', int(num_sellers * R/100)), ('ZIC',
int(num_sellers * (100-R)/100))]

trial_id = 'test_N=500_R=50'
traders_spec, order_sched = setup(start_time, end_time, buyers_spec,
sellers_spec)

# Configure output file settings. This is new for BSE version
28/10/2023.
# For each data file type, set True to write file data, false to not
write file.
# In this case, we set all to True so all output files will be
created. Set to False those that you do not want to create.
dump_flags = {'dump_blotters': True, 'dump_lobs': False,
'dump_strats': True,
```

```
                 'dump_avgbals': True, 'dump_tape': True}

verbose = False

n_runs_plot_trades(N, trial_id, start_time, end_time, traders_spec,
order_sched, dump_flags, verbose)
```

```
100%|██████████| 500/500 [03:03<00:00,  2.72it/s]
```

And plotting the profitabilities as before

```
profitability_df, wins_df = get_final_profitabilities_wins_df(N=500,
R=50, agents_types=['SHVR', 'ZIC'])

plt.rcParams["figure.figsize"] = [10.0, 5.0]
plt.rcParams["figure.autolayout"] = True
f, axes = plt.subplots(1, 2)
sns.kdeplot(data=profitability_df, fill=True, ax=axes[0])
sns.barplot(profitability_df, ax=axes[1], orient='h')

axes[0].set_xlabel("Profitability")

plt.show()
```



We see that the distributions have converged to be both more similiar, supporting our previous conclusion. We will confirm with another t-test.

```
t_test(profitability_df, 'SHVR', 'ZIC')
```

```
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
```

```
Condition ZIC. We cannot reject the null hypothesis (p=0.35).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable
```

From the t-test we can see that there is no significant difference between the profitabilitiy of SHVR and ZIC in these experiments.

PART B

We now repeat the experiment, varying the values of R (proportion of SHVR taders), using both small and large samples (50, 500).

```
for N in [50, 500]:
    for R in [10, 20, 30, 40, 60, 70, 80, 90]:
        print('N=' + str(N) + ", " + 'R=' + str(R))
        start_time = 0
        end_time = 60 * 10

        num_buyers = 20
        num_sellers = 20

        buyers_spec = [('SHVR', int(num_buyers * R/100)), ('ZIC',
int(num_buyers * (100-R)/100))]
        sellers_spec = [('SHVR', int(num_sellers * R/100)), ('ZIC',
int(num_sellers * (100-R)/100))]

        trial_id = 'test_N=' + str(N) + "_R=" + str(R)
        traders_spec, order_sched = setup(start_time, end_time,
buyers_spec, sellers_spec)

        # Configure output file settings. This is new for BSE version
28/10/2023.
        # For each data file type, set True to write file data, false
to not write file.
        # In this case, we set all to True so all output files will be
created. Set to False those that you do not want to create.
        dump_flags = {'dump_blotters': True, 'dump_lobs': False,
'dump_strats': True,
                      'dump_avgbals': True, 'dump_tape': True}

        verbose = False

        n_runs_plot_trades(N, trial_id, start_time, end_time,
traders_spec, order_sched, dump_flags, verbose)
```

Now we have the data we will proceed in the following order

1. For each pair (N, R), collate the final avg. profitability per trader values in a two-column dataframe ('SHVR', 'ZIC')

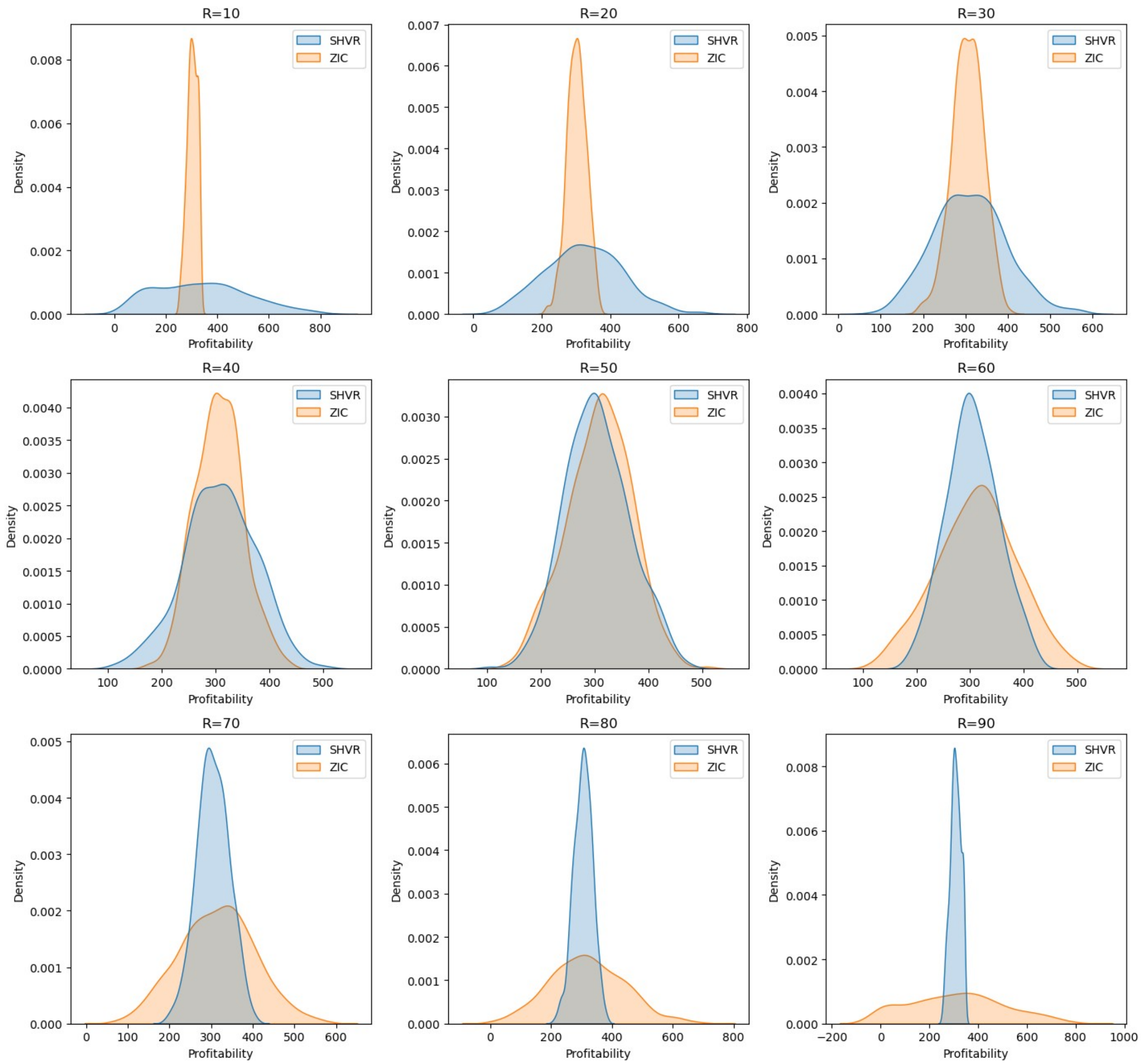
2. We will use these datasets to plot KDE graphs (for N=500 only), visualising the distributions of profitabilities for each R
3. Comment on location, spread and normality of the distributions
4. Compare the profitabilities for each R side-by-side on a bar plot (for N = 500 only)

Then for each N we will:

1. then run the Shapiro-Wilk test to confirm normality for each value of R
2. Decide which hypothesis test to use for each value of R.
3. Peform the tests to see if there is a significant difference in profitability between SHVR and ZIC for each value of R

We'll start with steps 1 and 2 to visualise profitability distrbutions for each R

```
fig, ax = plt.subplots(nrows=3, ncols=3, figsize=(14, 13))
fig.tight_layout(pad=15)

profitability_dfs = {50: {}, 500: {}}

for i, N in enumerate([50, 500]):
    for j, R in enumerate([10, 20, 30, 40, 50, 60, 70, 80, 90]):
        trial_id = 'test_N=' + str(N) + "_R=" + str(R)
        df,_ = get_final_profitabilities_wins_df(N,R,['SHVR','ZIC'])

        if N == 500:
            sns.kdeplot(df, fill=True, ax=ax[int(j/3), j % 3])
            ax[int(j/3), j % 3].set_title('R=' + str(R))
            ax[int(j/3), j % 3].set_xlabel('Profitability')

        profitability_dfs[N][R] = df

plt.show()
```

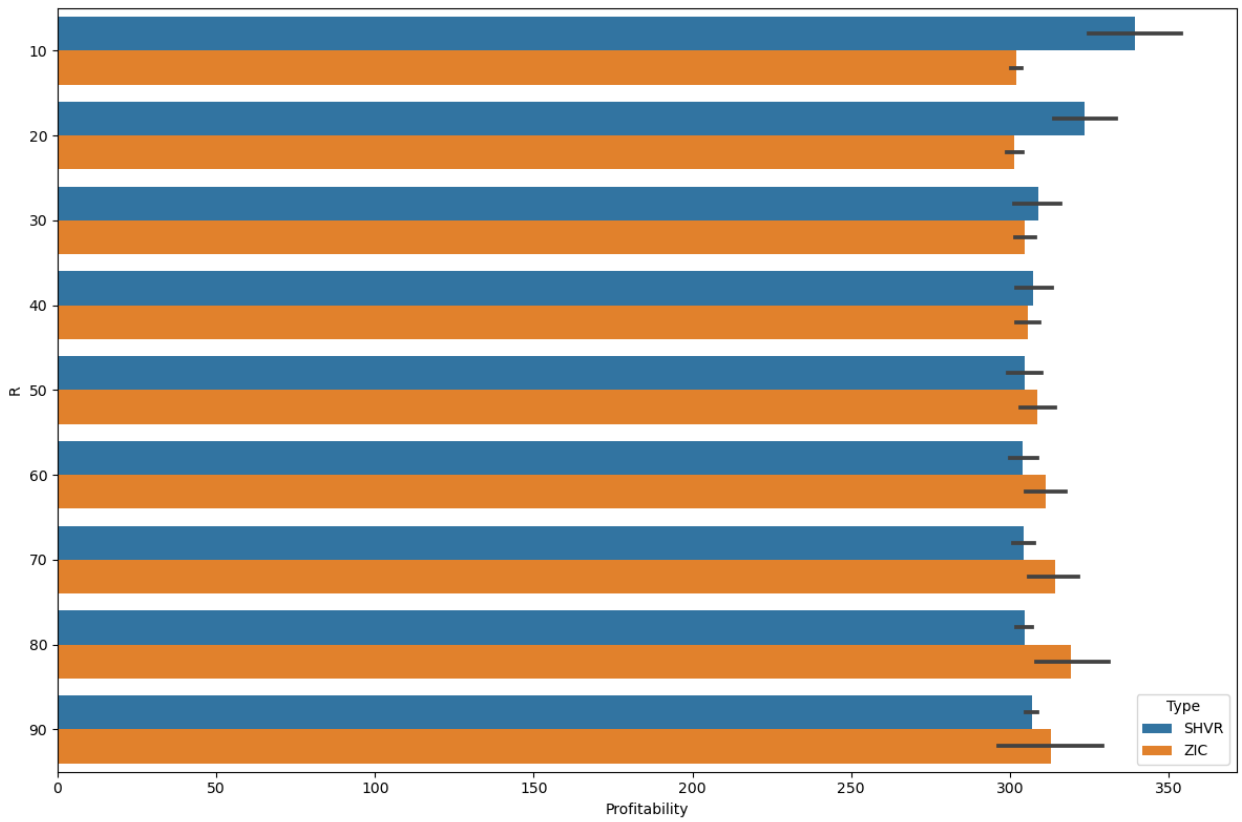


From these, we can see that the central tendencies for SHVR and ZIC are generally similar throughout, suggesting that, as in part A, there may be no significant difference in profitability. The spread of the proftiabilities seems to correlate with trader type's proportion of the market i.e. a lower variance for SHVR when R is low, and a higher variance when R is high - and vice-versa for ZIC. [Suggests?]

We will also show the results for each R value on a barplot (N=500):

```
N = 500

df_r = []
df_t = []
df_p = []

for j, R in enumerate([10, 20, 30, 40, 50, 60, 70, 80, 90]):
```

```
    profitability_df = profitability_dfs[N][R]
    for c in profitability_df:
        for r in profitability_df[c]:
            df_t.append(c)
            df_r.append(R)
            df_p.append(r)

df_barplot = pd.DataFrame({'R': df_r, 'Type': df_t, 'Profitability':
df_p})

plt.figure(figsize=(12,8))
sns.barplot(df_barplot, x='Profitability', y='R', hue='Type',
orient='h')

plt.show()
```



In a similar fashion to the decreae in variance of SHVR profitability as R increases, we see the same trend with the mean profitability itself. This graph would suggest that the agent type whose proportional presence is the smallest, enjoys a higher profitability.

We will now verify the normality for each value of R using Shapiro-Wilk as we did in part A

```
N = 50

for R in [10, 20, 30, 40, 60, 70, 80, 90]:
```

```
    print('\nN = ' + str(N) + ', ' + str(R))
    shapiro_wilk(profitability_dfs[N][R])
```

```
N = 50, 10
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We can reject the null hypothesis (p=0.01). Therefore,
profitability for SHVR is not normally distributed.
Condition ZIC. We can reject the null hypothesis (p=0.02). Therefore,
profitability for ZIC is not normally distributed.

N = 50, 20
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.11).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.10).
Therefore, profitability for ZIC is normally distributed.

N = 50, 30
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.94).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.94).
Therefore, profitability for ZIC is normally distributed.

N = 50, 40
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.55).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.46).
Therefore, profitability for ZIC is normally distributed.

N = 50, 60
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.41).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.42).
Therefore, profitability for ZIC is normally distributed.

N = 50, 70
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.60).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.58).
```

```
Therefore, profitability for ZIC is normally distributed.

N = 50, 80
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.07).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.07).
Therefore, profitability for ZIC is normally distributed.

N = 50, 90
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We can reject the null hypothesis (p=0.01). Therefore,
profitability for SHVR is not normally distributed.
Condition ZIC. We can reject the null hypothesis (p=0.01). Therefore,
profitability for ZIC is not normally distributed.
```

From this, we can see that the proftiabilities are not normally distributed only for values R=10,90

```
for R in [20, 30, 40, 60, 70, 80]:
    print('\nN = ' + str(N) + ', ' + str(R))
    t_test(profitability_dfs[N][R], 'SHVR', 'ZIC')


N = 50, 20
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.11).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 50, 30
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.75).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 50, 40
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.15).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 50, 60
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.58).
```

```
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 50, 70
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.39).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 50, 80
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.62).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable
```

From this we can see that there is no significant difference in profitability between SHVR and ZIC for values 20<=R<=80.

For values R=10,90, we assume a non-normal distribution, and therefore use the Mann Whitney-U test [further justification?]

```
for R in [10, 90]:
    print('\nN = ' + str(N) + ', ' + str(R))
    mann_whitney_u_test(profitability_dfs[N][R], 'SHVR', 'ZIC')
```

```
N = 50, 10
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.45).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 50, 90
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.81).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable
```

We will now repeat this process, starting with the normality testing for N = 500

```
N = 500

for R in [10, 20, 30, 40, 60, 70, 80, 90]:
    print('\nN = ' + str(N) + ', ' + str(R))
    shapiro_wilk(profitability_dfs[N][R])
```

```
N = 500, 10
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00). Therefore,
profitability for SHVR is not normally distributed.
Condition ZIC. We can reject the null hypothesis (p=0.00). Therefore,
profitability for ZIC is not normally distributed.

N = 500, 20
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.07).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.08).
Therefore, profitability for ZIC is normally distributed.

N = 500, 30
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.11).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.16).
Therefore, profitability for ZIC is normally distributed.

N = 500, 40
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.56).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.53).
Therefore, profitability for ZIC is normally distributed.

N = 500, 60
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.24).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.26).
Therefore, profitability for ZIC is normally distributed.

N = 500, 70
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.71).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.74).
Therefore, profitability for ZIC is normally distributed.

N = 500, 80
```

```
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.25).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.25).
Therefore, profitability for ZIC is normally distributed.

N = 500, 90
Using Shapiro-Wilk test to test the null hypothesis that the data was
drawn from a normal distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00). Therefore,
profitability for SHVR is not normally distributed.
Condition ZIC. We can reject the null hypothesis (p=0.00). Therefore,
profitability for ZIC is not normally distributed.
```

```python
for R in [20, 30, 40, 60, 70, 80]:
    print('\nN = ' + str(N) + ', ' + str(R))
    t_test(profitability_dfs[N][R], 'SHVR', 'ZIC')
```

```
N = 500, 20
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00). Therefore,
SHVR and ZIC have statistically different profitabilities

N = 500, 30
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.29).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 500, 40
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.66).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 500, 60
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.08).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable

N = 500, 70
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
```

```
Condition SHVR. We can reject the null hypothesis (p=0.02). Therefore,
SHVR and ZIC have statistically different profitabilities

N = 500, 80
Using t-test to test the null hypothesis that the two profitability
samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.01). Therefore,
SHVR and ZIC have statistically different profitabilities
```

```
for R in [10, 90]:
    print('\nN = ' + str(N) + ', ' + str(R))
    mann_whitney_u_test(profitability_dfs[N][R], 'SHVR', 'ZIC')
```

```
N = 500, 10
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00). Therefore,
SHVR and ZIC have statistically different profitabilities

N = 500, 90
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.43).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable
```

With the larger number of samples, we can see there is a significant difference in profitability between SHVR and ZIC when R=10,20,70,80. Based on our previous bar plot we can conclude that SHVR outperforms ZIC when R=10,20, and conversely, ZIC is more profitable when R=70,80

PART C

For this section, we will compare the profitability SHVR, GVWY, ZIC and ZIP with various ratios of these agents in the market. In the previous section, larger sample sizes of N >= 500 were show to yield significant results. Due to the increased number of trader types, we will increase this number to 1000.

We will collect data for each of the specified trader ratios and proceed as follows:

1. Form a a table of profitabilities and wins for each trader at each ratio
2. Display these 'wins' in a table

And then for each ratio:

1. Plot the profitabilities at each ratio in a violin plot in order to compare both profitabilities and their underlying distributions
2. Perform tests of normarlity in order to decide on hypothesis test
3. Perform appropriate tests to compare profitabilities

```
ratios = [
        [25, 25, 25, 25],
        [40, 20, 20, 20], [20, 40, 20, 20], [20, 20, 40, 20], [20, 20,
20, 40],
        [10, 30, 30, 30], [30, 10, 30, 30], [30, 30, 10, 30], [30, 30,
30, 10],
        [70, 10, 10, 10], [10, 70, 10, 10], [10, 10, 70, 10], [10, 10,
10, 70]
    ]

N = 1000

for R in ratios:
        print('N=' + str(N) + ", " + 'R=' + ":".join(map(str, R)))
        start_time = 0
        end_time = 60 * 10

        num_buyers = 20
        num_sellers = 20

        buyers_spec = [('SHVR', int(num_buyers * R[0]/100)), ('GVWY',
int(num_buyers * R[1]/100)), ('ZIC', int(num_buyers * R[2]/100)),
('ZIP', int(num_buyers * R[3]/100))]
        sellers_spec = [('SHVR', int(num_sellers * R[0]/100)),
('GVWY', int(num_sellers * R[1]/100)), ('ZIC', int(num_sellers *
R[2]/100)), ('ZIP', int(num_sellers * R[3]/100))]


        trial_id = 'test_N=' + str(N) + ", " + 'R=' +
":".join(map(str, R))
        traders_spec, order_sched = setup(start_time, end_time,
buyers_spec, sellers_spec)

        # Configure output file settings. This is new for BSE version
28/10/2023.
        # For each data file type, set True to write file data, false
to not write file.
        # In this case, we set all to True so all output files will be
created. Set to False those that you do not want to create.
        dump_flags = {'dump_blotters': True, 'dump_lobs': True,
'dump_strats': True,
                    'dump_avgbals': True, 'dump_tape': True}

        verbose = False

        #n_runs_plot_trades(N, trial_id, start_time, end_time,
traders_spec, order_sched, dump_flags, verbose)
```

```
N=1000, R=25:25:25:25
N=1000, R=40:20:20:20
```

```
N=1000, R=20:40:20:20
N=1000, R=20:20:40:20
N=1000, R=20:20:20:40
N=1000, R=10:30:30:30
N=1000, R=30:10:30:30
N=1000, R=30:30:10:30
N=1000, R=30:30:30:10
N=1000, R=70:10:10:10
N=1000, R=10:70:10:10
N=1000, R=10:10:70:10
N=1000, R=10:10:10:70
```

Now we have our data for each ratio, we can use the final avg. profitability for each trader in each session and create a table showing the number of 'wins' for each trader - a win being a trader type having the highest avg. profitability per trader at the end of a session.

At the same time, we can create a list of profitbility dataframes for each ratio to use for analysing the distributions and for hypothesis testing

```
ratios = [
        [25, 25, 25, 25],
        [40, 20, 20, 20], [20, 40, 20, 20], [20, 20, 40, 20], [20, 20,
20, 40],
        [10, 30, 30, 30], [30, 10, 30, 30], [30, 30, 10, 30], [30, 30,
30, 10],
        [70, 10, 10, 10], [10, 70, 10, 10], [10, 10, 70, 10], [10, 10,
10, 70]
    ]

N = 1000

wins_df = pd.DataFrame(columns=['Ratio', 'SHVR', 'GVWY', 'ZIC',
'ZIP'])
win_total_row = {'Ratio': 'Total', 'SHVR': 0, 'GVWY': 0, 'ZIC': 0,
'ZIP': 0}
profitability_dfs = {}

for R in ratios:
    ratio = ":".join(map(str, R))
    profitability_df, wins = get_final_profitabilities_wins_df(N,
':'.join(map(str, R)), ['GVWY', 'SHVR', 'ZIC', 'ZIP'])
    win_row = {'Ratio': ratio, 'SHVR': wins['SHVR'], 'GVWY':
wins['GVWY'], 'ZIC': wins['ZIC'], 'ZIP': wins['ZIP']}
    win_total_row = {'Ratio': 'Total','SHVR': win_total_row['SHVR'] +
wins['SHVR'], 'GVWY': win_total_row['GVWY'] + wins['GVWY'], 'ZIC':
win_total_row['ZIC'] + wins['ZIC'], 'ZIP': win_total_row['ZIP'] +
wins['ZIP']}

    wins_df = pd.concat([wins_df, pd.DataFrame(win_row, index=[0])],
```

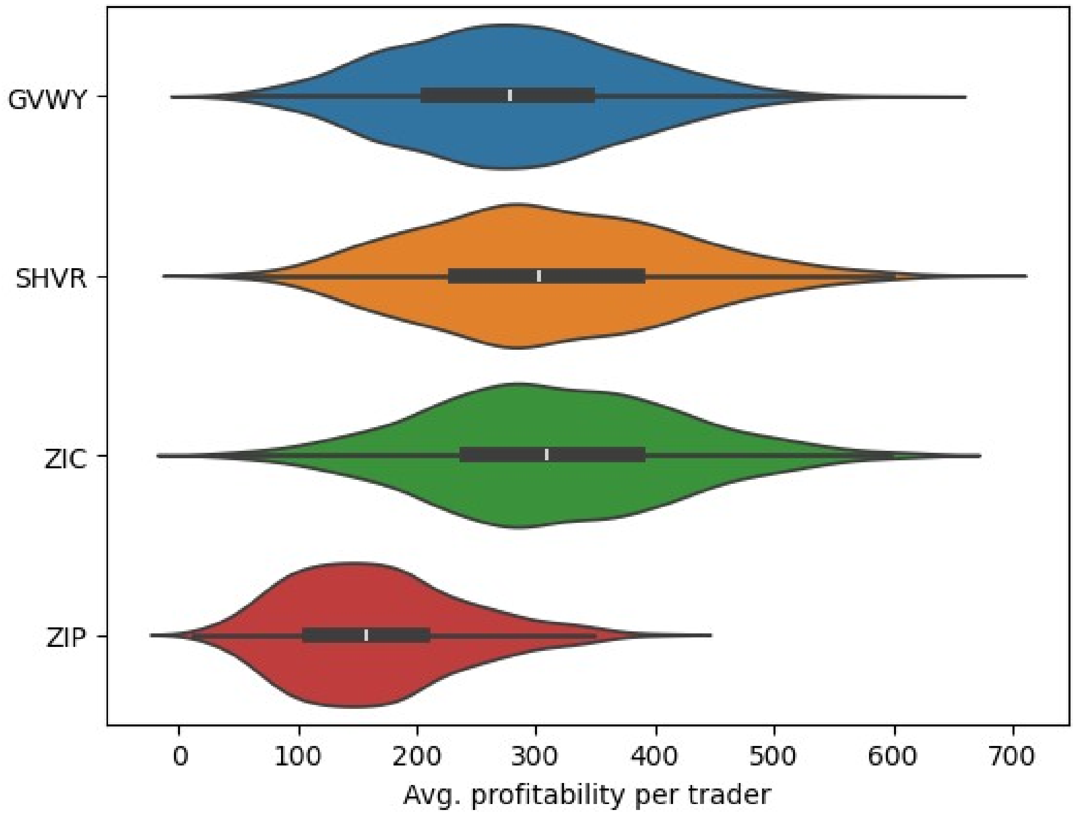
```
ignore_index=True)

    profitability_dfs[ratio] = profitability_df

wins_df = pd.concat([wins_df, pd.DataFrame(win_total_row, index=[0])],
ignore_index=True)


print(wins_df)
```

```
          Ratio  SHVR  GVWY   ZIC  ZIP
0   25:25:25:25   364   229   375   32
1   40:20:20:20   311   280   368   41
2   20:40:20:20   339   227   412   22
3   20:20:40:20   353   219   333   95
4   20:20:20:40   360   271   363    6
5   10:30:30:30   392   243   357    8
6   30:10:30:30   329   295   350   26
7   30:30:10:30   313   232   444   11
8   30:30:30:10   290   208   397  105
9   70:10:10:10   174   265   431  130
10  10:70:10:10   309   208   450   33
11  10:10:70:10   335   188   186  291
12  10:10:10:70   329   292   369   10
13        Total  4198  3157  4835  810
```

From this table, we see that ZIC overall has the most wins, whilst ZIP performs significantly worse than the other three.

We will now create violin plots for profitbility at each of the ratios, starting with 25:25:25:25

```
plt.xlabel('Avg. profitability per trader')
sns.violinplot(data=profitability_dfs['25:25:25:25'], orient='h')
plt.show()
```



The central tendancies of thi splot reflect the #wins results we saw in the table, but also show the smaller variance of ZIP's profitability. The distributions also seem to follow that of a Normal one, although ZIP's is slightly skewed, and we will confirm this with the Kolmogorov-Smirnov test due to the large sample size

```
kolmogorov_smirnov(profitability_dfs['25:25:25:25'])
```

```
Using Kolmogorov-Smirnov test to test the null hypothesis that the
data was drawn from a normal distribution:
Condition GVWY. We cannot reject the null hypothesis (p=0.69).
Therefore, profitability for GVWY is normally distributed.
Condition SHVR. We cannot reject the null hypothesis (p=0.18).
Therefore, profitability for SHVR is normally distributed.
Condition ZIC. We cannot reject the null hypothesis (p=0.32).
Therefore, profitability for ZIC is normally distributed.
Condition ZIP. We can reject the null hypothesis (p=0.02). Therefore,
profitability for ZIP is not normally distributed.
```

From this we can see that profitability for ZIP only is not normally distributed. Despite the having slightly less power, non-parametric tests are valid when the data is normal or otherwise. We could use the Krusakal-Wallis test on all 4 distrbutions at once, but the plot already shows a large discrenpency between the medians of ZIP and the others. Instead we will do a pair-wise Mann-Whitney-U test

```python
print(profitability_dfs['25:25:25:25'])

#ruskal_wallis_test(profitability_dfs['25:25:25:25'], ['GVWY', 'SHVR',
'ZIC', 'ZIP'])

mann_whitney_u_test(profitability_dfs['25:25:25:25'], 'GVWY', 'SHVR')
mann_whitney_u_test(profitability_dfs['25:25:25:25'], 'GVWY', 'ZIC')
mann_whitney_u_test(profitability_dfs['25:25:25:25'], 'GVWY', 'ZIP')
mann_whitney_u_test(profitability_dfs['25:25:25:25'], 'SHVR', 'ZIC')
mann_whitney_u_test(profitability_dfs['25:25:25:25'], 'SHVR', 'ZIP')
mann_whitney_u_test(profitability_dfs['25:25:25:25'], 'ZIC', 'ZIP')
```

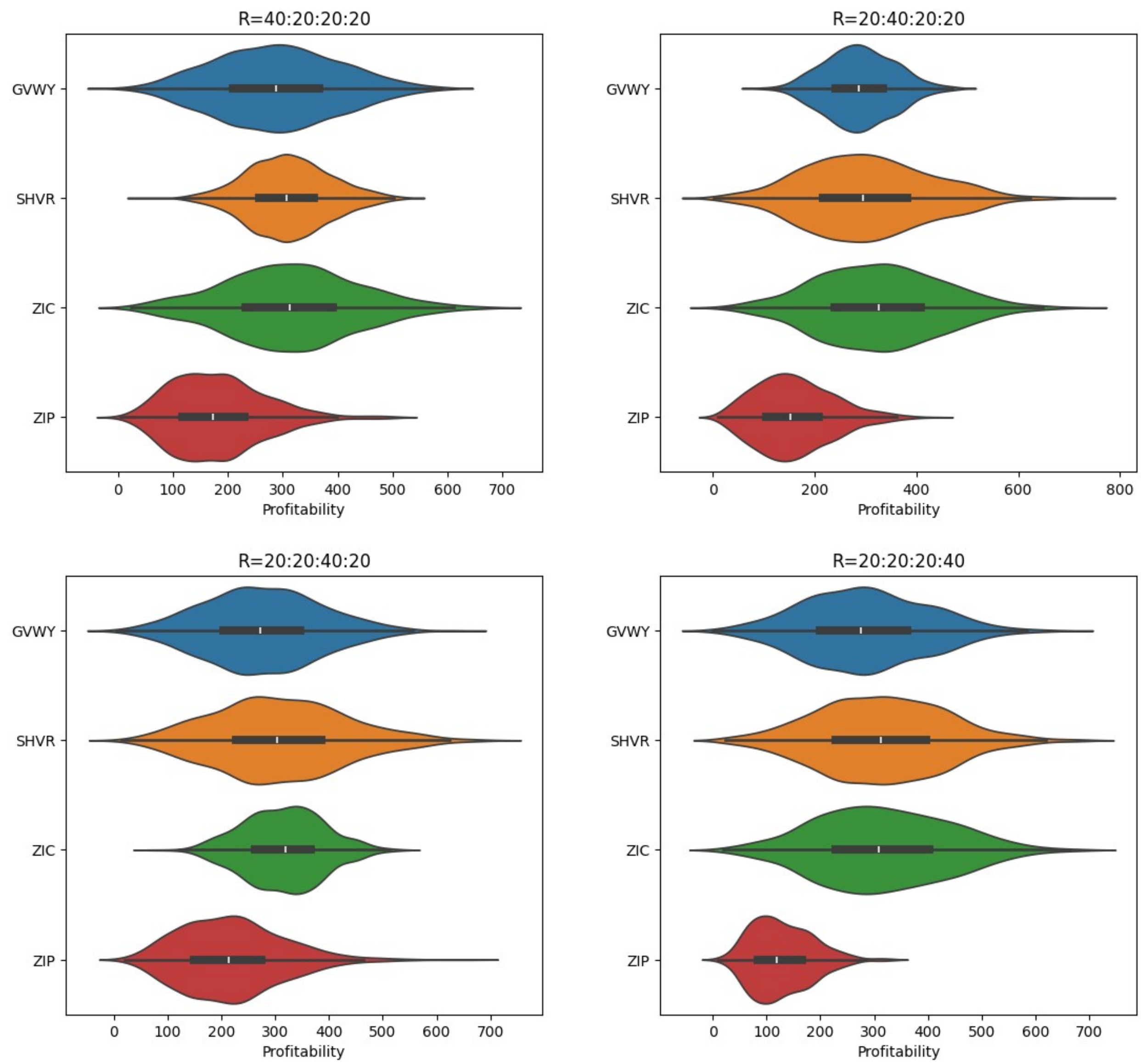
```
      GVWY   SHVR    ZIC    ZIP
0    310.2  321.7  253.0  124.3
1    254.4  225.5  369.4  208.0
2    516.7  145.5  353.0   90.5
3    240.3  267.4  366.0  173.5
4    367.8  330.9  310.3  139.9
..     ...    ...    ...    ...
995  234.8  455.0  229.1  203.5
996  141.5  272.8  349.2  232.4
997  241.8  176.7  370.2  133.2
998   75.1  372.6  323.3  239.7
999  288.3  336.7  172.2  213.2

[1000 rows x 4 columns]
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.24276243).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable
```

```
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities
```

From this, the table and the plot we can see that SHVR and ZIC have indistinguisable performance and both beat GVWY, which in turn beats ZIP.

We continue with the ratio 40:20:20:20 and its permutations. N.B. ratio is SHVR:GVWYY:ZIC:ZIP

```
fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(12, 11))
fig.tight_layout(pad=5)

for i, R in enumerate(['40:20:20:20', '20:40:20:20', '20:20:40:20',
'20:20:20:40']):
       sns.violinplot(data=profitability_dfs[R], orient='h',
ax=ax[int(i/2), i % 2])
       ax[int(i/2), i % 2].set_title('R=' + R)
       ax[int(i/2), i % 2].set_xlabel('Profitability')

plt.show()
```



Here we can see similar results to 25:25:25:25, except in a similar fashion to part A, the variance of each agent's profitability decreases as its prevalance in the market increases - and its highest and lowest performing traders do not deviate from the mean as much.

For the remainder of these we will assume non-normality and proceed with the pairwise Mann-Whitney-U test

```
for i, R in enumerate(['40:20:20:20', '20:40:20:20', '20:20:40:20',
'20:20:20:40']):
    print('\nR = ' + R)
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'SHVR')
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'ZIC')
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'ZIP')
    mann_whitney_u_test(profitability_dfs[R], 'SHVR', 'ZIC')
```

```
    mann_whitney_u_test(profitability_dfs[R], 'SHVR', 'ZIP')
    mann_whitney_u_test(profitability_dfs[R], 'ZIC', 'ZIP')
```

```
R = 40:20:20:20
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000172).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000233).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.35871167).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 20:40:20:20
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.04468563).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
```

```
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000327).
Therefore, SHVR and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 20:20:40:20
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000002).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
```

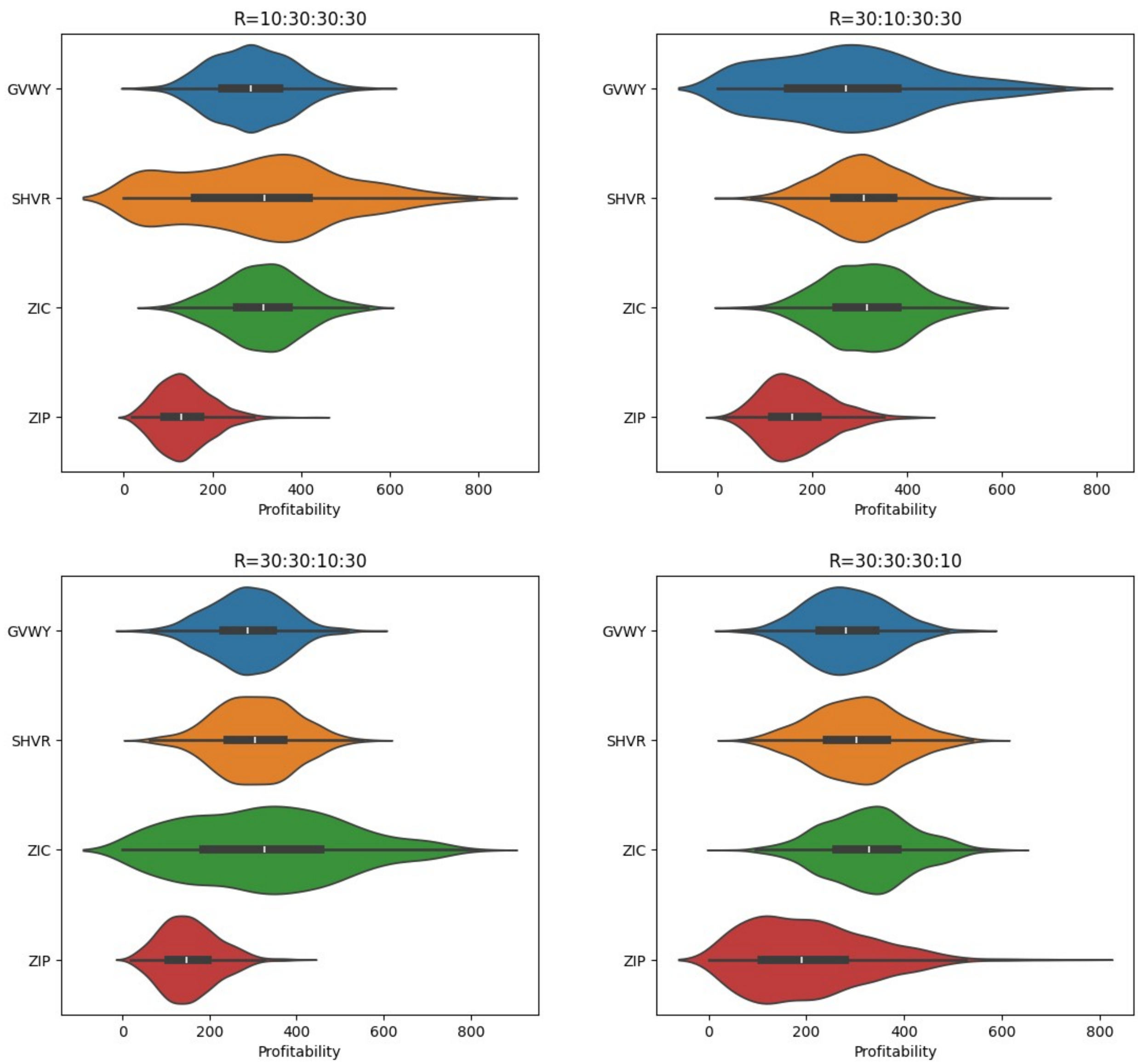
```
Condition SHVR. We can reject the null hypothesis (p=0.02097156).
Therefore, SHVR and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 20:20:20:40
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000001).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.77219061).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
```

```
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities
```

Here we see that again, all agent pairings except SHVR and ZIC are statistcally indisinguishable, except when either GVWY or ZIC is the dominant agent type, in which case they too have different profitabilities.

Now permutations of 10:30:30:30

```
fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(12, 11))
fig.tight_layout(pad=5)

for i, R in enumerate(['10:30:30:30', '30:10:30:30', '30:30:10:30',
'30:30:30:10']):
       sns.violinplot(data=profitability_dfs[R], orient='h',
ax=ax[int(i/2), i % 2])
       ax[int(i/2), i % 2].set_title('R=' + R)
       ax[int(i/2), i % 2].set_xlabel('Profitability')

plt.show()
```



Here we see that the minority trading type has a much larger variance of profitability, slightly skewed to the lower side. The higher profitabilities of each minority agent now reach much higher, notably in the case of ZIP.

```
for i, R in enumerate(['10:30:30:30', '30:10:30:30', '30:30:10:30',
'30:30:30:10']):
    print('\nR = ' + R)
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'SHVR')
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'ZIC')
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'ZIP')
    mann_whitney_u_test(profitability_dfs[R], 'SHVR', 'ZIC')
    mann_whitney_u_test(profitability_dfs[R], 'SHVR', 'ZIP')
    mann_whitney_u_test(profitability_dfs[R], 'ZIC', 'ZIP')
```

```
R = 10:30:30:30
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00538055).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.26911098).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 30:10:30:30
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities
```

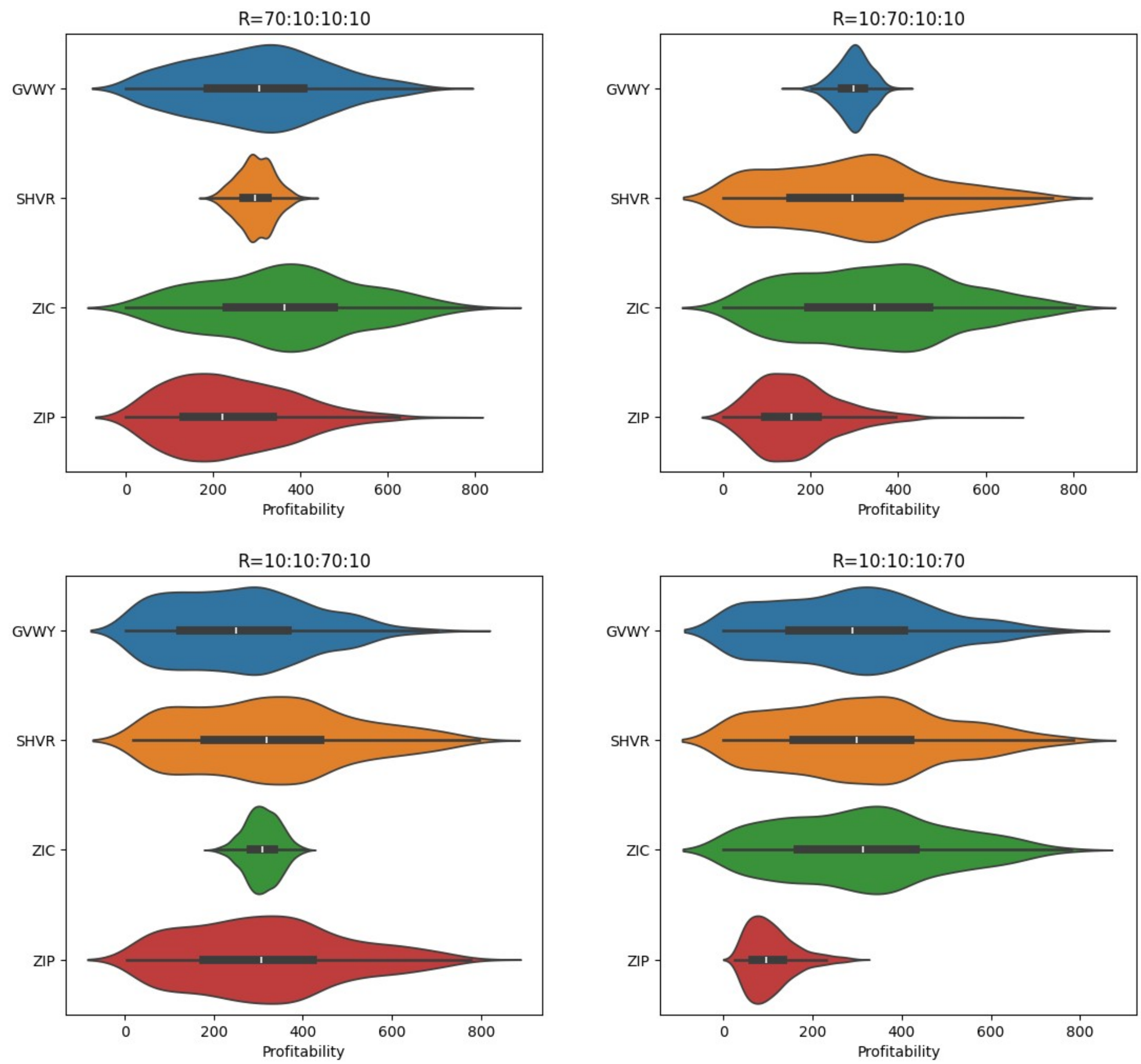
```
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.09831459).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 30:30:10:30
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00003444).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000005).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00336905).
Therefore, SHVR and ZIC have statistically different profitabilities
```

```
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 30:30:30:10
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000062).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000001).
Therefore, SHVR and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities
```

Now SHVR and GVWY are only distinguishable when neither ZIC nor ZIP is the minority agent. The rest of the time, all have different profitabilities.

Finally for permutations of 70:10:10:10

```
fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(12, 11))
fig.tight_layout(pad=5)

for i, R in enumerate(['70:10:10:10', '10:70:10:10', '10:10:70:10',
'10:10:10:70']):
       sns.violinplot(data=profitability_dfs[R], orient='h',
ax=ax[int(i/2), i % 2])
       ax[int(i/2), i % 2].set_title('R=' + R)
       ax[int(i/2), i % 2].set_xlabel('Profitability')

plt.show()
```



Similar to permutations of R=40:10:10:10, we can see the most prevelant agent type having shrunken variance. This makes particular sense in the case of SHVR as it is parasitic and relies on other traders to set the price [3]. Interestingly when ZIC, is the majority agent, ZIP's profitbality distribution becomes closer to that of SHVR and GVWY.

```
for i, R in enumerate(['70:10:10:10', '10:70:10:10', '10:10:70:10',
'10:10:10:70']):
    print('\nR = ' + R)
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'SHVR')
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'ZIC')
    mann_whitney_u_test(profitability_dfs[R], 'GVWY', 'ZIP')
    mann_whitney_u_test(profitability_dfs[R], 'SHVR', 'ZIC')
    mann_whitney_u_test(profitability_dfs[R], 'SHVR', 'ZIP')
    mann_whitney_u_test(profitability_dfs[R], 'ZIC', 'ZIP')
```

```
R = 70:10:10:10
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.25346494).
Therefore, profitability of GVWY and SHVR is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 10:70:10:10
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We cannot reject the null hypothesis (p=0.79444102).
Therefore, profitability of GVWY and SHVR is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
```

```
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities



R = 10:10:70:10
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.08616202).
Therefore, profitability of SHVR and ZIC is statistically
```

```
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIP. We cannot reject the null hypothesis (p=0.25282160).
Therefore, profitability of SHVR and ZIP is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIP. We cannot reject the null hypothesis (p=0.77100585).
Therefore, profitability of ZIC and ZIP is statistically
indistinguishable



R = 10:10:10:70
Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.03633357).
Therefore, GVWY and SHVR have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00406196).
Therefore, GVWY and ZIC have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition GVWY. We can reject the null hypothesis (p=0.00000000).
Therefore, GVWY and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition ZIC. We cannot reject the null hypothesis (p=0.52327703).
Therefore, profitability of SHVR and ZIC is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition SHVR. We can reject the null hypothesis (p=0.00000000).
Therefore, SHVR and ZIP have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
```

```
profitability samples are from the same distribution:
Condition ZIC. We can reject the null hypothesis (p=0.00000000).
Therefore, ZIC and ZIP have statistically different profitabilities
```

Here we can see that when SHVR or GVWY is the majority agent, their profitabilities are indistinguishable. However when it is ZIC, all pairings [SHVR, ZIC, ZIP] become indistinguishable. And for ZIP, only ZIC and SHVR's become indistinguishable

PART D

We will first aim to replicate the outlined described experiment and determine if a single ZIPSH buyer (Stochastic hillclimber ZIP variant with k=4 candidate strategies per evaluation cycle, as described in [1]) makes a reliable improvement in profitability in an otherwise all-ZIC market (ratio of 1:19). Although longer market sessions would have allowed for further adaptation, they were restricted to 30 days to allow for a larger number of independent trials (100 here). Strategy evaluation time was kept uniformly random between 2-3 hours as in default BSE and to prevent synchronised updates between agents, although is nullified by there only being one adaptive ZIPSH in the market.

The S-D curve was set to ensure all traders were intramarginal - all ZICs would be expected to trade and the ZIPSH would get useful PPS feedback.

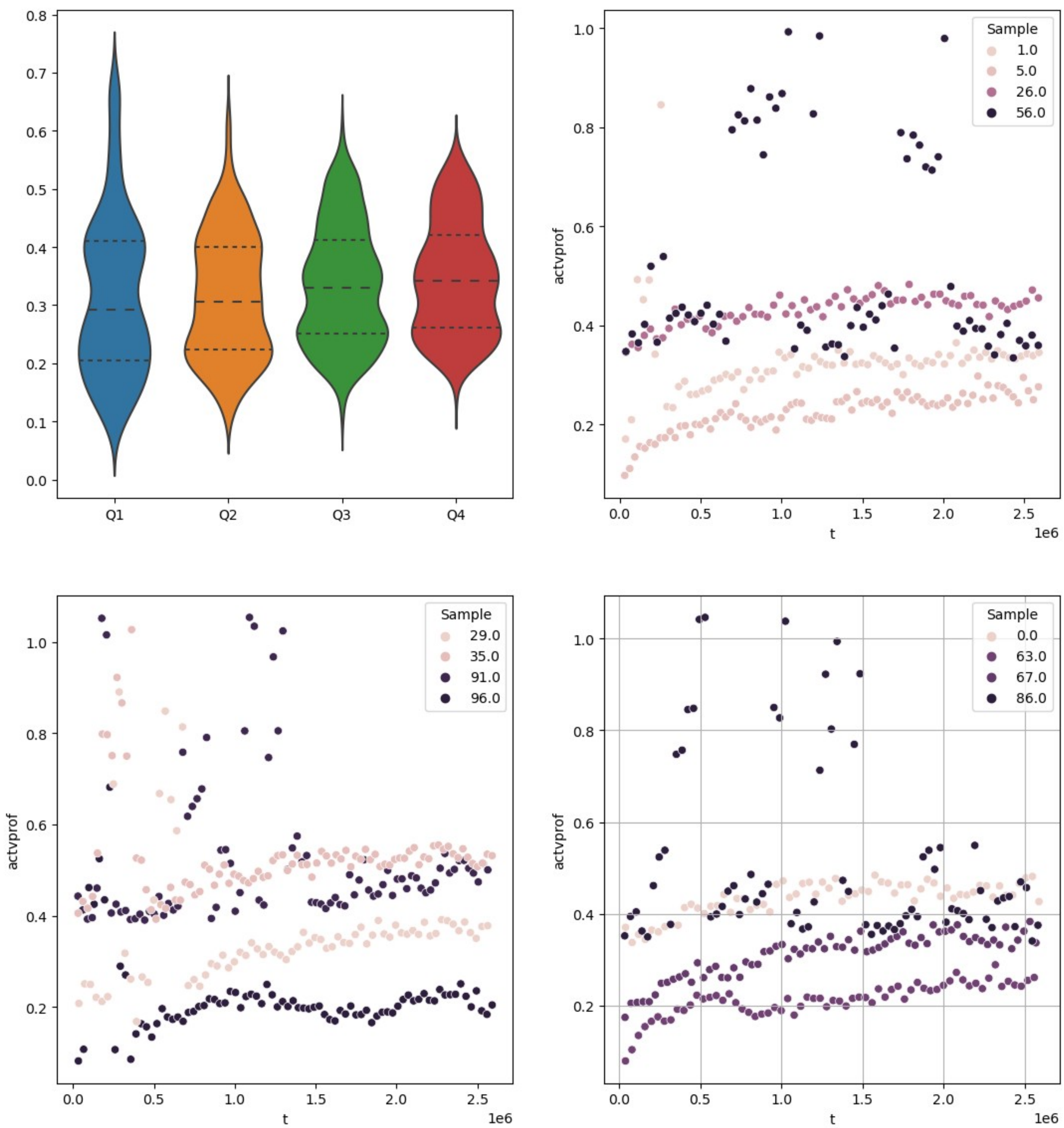
```
N = 50

start_time = 0
end_time = 60 * 60 * 24 * 30

sellers_spec = [('ZIC', 10)]
buyers_spec = [('ZIPSH', 1, {'optimizer': 'ZIPSH', 'k': 4}), ('ZIC',
9)]
traders_spec = {'sellers':sellers_spec, 'buyers':buyers_spec}

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

demand_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[dem_range], 'stepmode': 'fixed'}]
supply_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[sup_range], 'stepmode': 'fixed'}]

order_interval = 60
order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
        'interval': order_interval, 'timemode': 'periodic'}

trial_id = 'test_N=1_R=1:9'

dump_flags = {'dump_blotters': True, 'dump_lobs': True, 'dump_strats':
True,
```

```
                     'dump_avgbals': True, 'dump_tape': True}

verbose = False

n_runs_plot_trades(N, trial_id, start_time, end_time, traders_spec,
order_sched, dump_flags, verbose)
```

Across the 100 trials, we can gather the pps values into a dataframe with a time quater for each period within the experiment . We can then compare their distributions side-by-side. We also can pick 3 random samples of 4 sessions, showing how PPS varies with time

```
df_strat_comp, df_half_paired, df_quart_prof, df_sample_prof_arr =
load_strat_data('./data-d1/test_d1_N=50_R=1_9', remove_outliers=True)

print(df_half_paired.head(5))

f, axes = plt.subplots(2, 2, figsize=(13, 14))
#sns.boxplot(data=df_quart_prof, orient='h', ax=axes[0])
sns.violinplot(data=df_quart_prof,  split=True, gap=-.25,
inner="quart", orient='x', ax=axes[0,0])
plt.grid()
#sns.violinplot(data=df_quart_prof, split=True, gap=-.25,
inner="quart", orient='x', ax=axes[1])
sns.scatterplot(data=df_sample_prof_arr[0], x='t', y='actvprof',
hue='Sample', ax=axes[0,1])
plt.grid()
sns.scatterplot(data=df_sample_prof_arr[1], x='t', y='actvprof',
hue='Sample', ax=axes[1,0])
plt.grid()
sns.scatterplot(data=df_sample_prof_arr[2], x='t', y='actvprof',
hue='Sample', ax=axes[1,1])
plt.grid()

plt.grid()
```

	First half	Second half	Difference
0	0.370690	0.477101	0.106411
1	0.337541	0.427491	0.089950
2	0.355101	0.440185	0.085084
3	0.344130	0.443649	0.099519
4	0.364698	0.451655	0.086957



We can see the upwards shift in the quartile and median pps between the 4 quarters of the trading sessions, with an even greater shift between the first and last quater. The large outliers and extreme variance in PPS towards the start of many sessions reflects the large initial hyperparameter adjustments; most of the agents converge to a steady climb in PPS.

A non-normal distribution is also apparent - we will use Kolmogorov-Smirnov to confirm non-normality.

```
kolmogorov_smirnov(df_quart_prof)
```

```
Using Kolmogorov-Smirnov test to test the null hypothesis that the
data was drawn from a normal distribution:
Condition First half. We can reject the null hypothesis (p=0.00).
Therefore, profitability for First half is not normally distributed.
Condition Second half. We can reject the null hypothesis (p=0.00).
Therefore, profitability for Second half is not normally distributed.
```

And now a Mann-Whitney-U test for significance

```
mann_whitney_u_test(df_quart_prof, 'Q1', 'Q2')
mann_whitney_u_test(df_quart_prof, 'Q1', 'Q3')
mann_whitney_u_test(df_quart_prof, 'Q1', 'Q4')
mann_whitney_u_test(df_quart_prof, 'Q2', 'Q3')
mann_whitney_u_test(df_quart_prof, 'Q3', 'Q4')

Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition Q2. We cannot reject the null hypothesis (p=0.14155991).
Therefore, profitability of Q1 and Q2 is statistically
indistinguishable


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition Q1. We can reject the null hypothesis (p=0.00000000).
Therefore, Q1 and Q3 have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition Q1. We can reject the null hypothesis (p=0.00000000).
Therefore, Q1 and Q4 have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition Q2. We can reject the null hypothesis (p=0.00000005).
Therefore, Q2 and Q3 have statistically different profitabilities


Using Mann-Whitney-U test to test the null hypothesis that the two
profitability samples are from the same distribution:
Condition Q3. We can reject the null hypothesis (p=0.00050983).
Therefore, Q3 and Q4 have statistically different profitabilities
```

We also have a dataframe of PPS at t in the first of a session, and t's counterpart in the second half of the form as well as the difference between the two. This gives us paired ranks of the difference between the PPS at time t and time t+(T/2) where T is the total experiment time. We can then use a Wilcoxon-Signed-Rank test to check for a significant difference in PPS between the two halves of the experiment

```
wilcoxon_signed_rank_test(df_half_paired['Difference'], 'First Half', 
'Second Half')
```

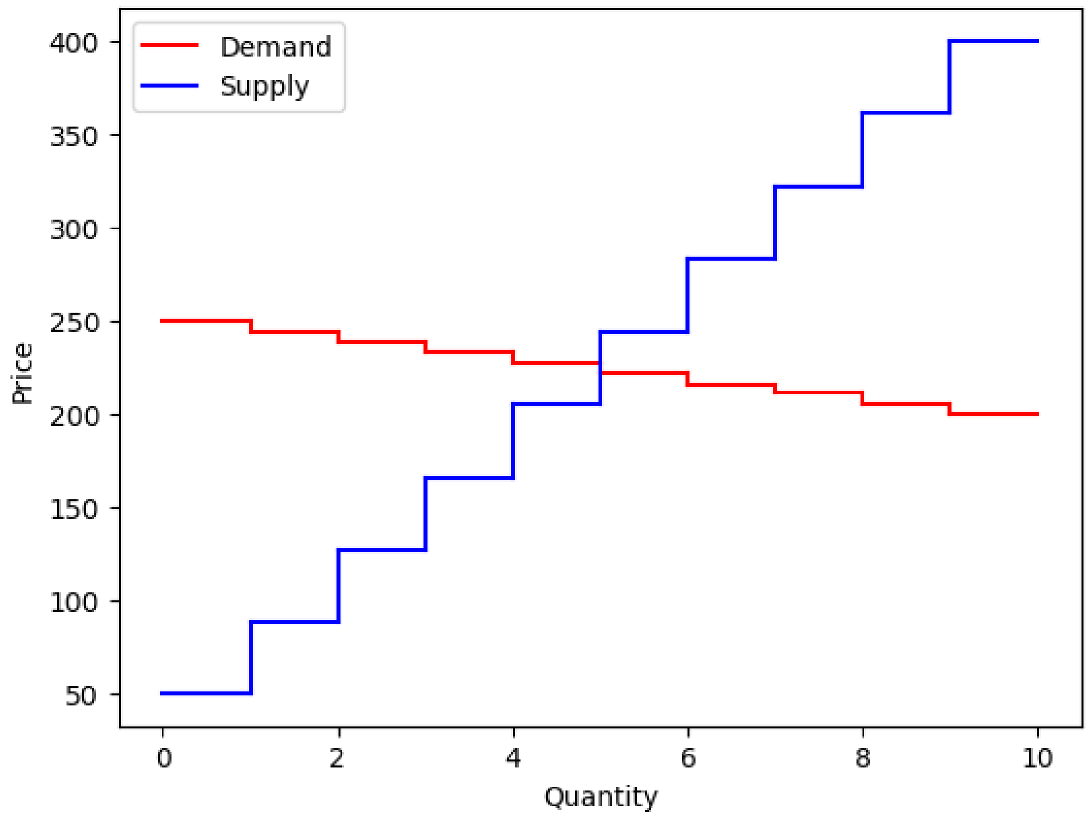
```
Using the Wilcoxon signed rank test
Condition First Half. We can reject the null hypothesis 
(p=0.00000000). Therefore, First Half and Second Half have 
statistically different profitabilities
```

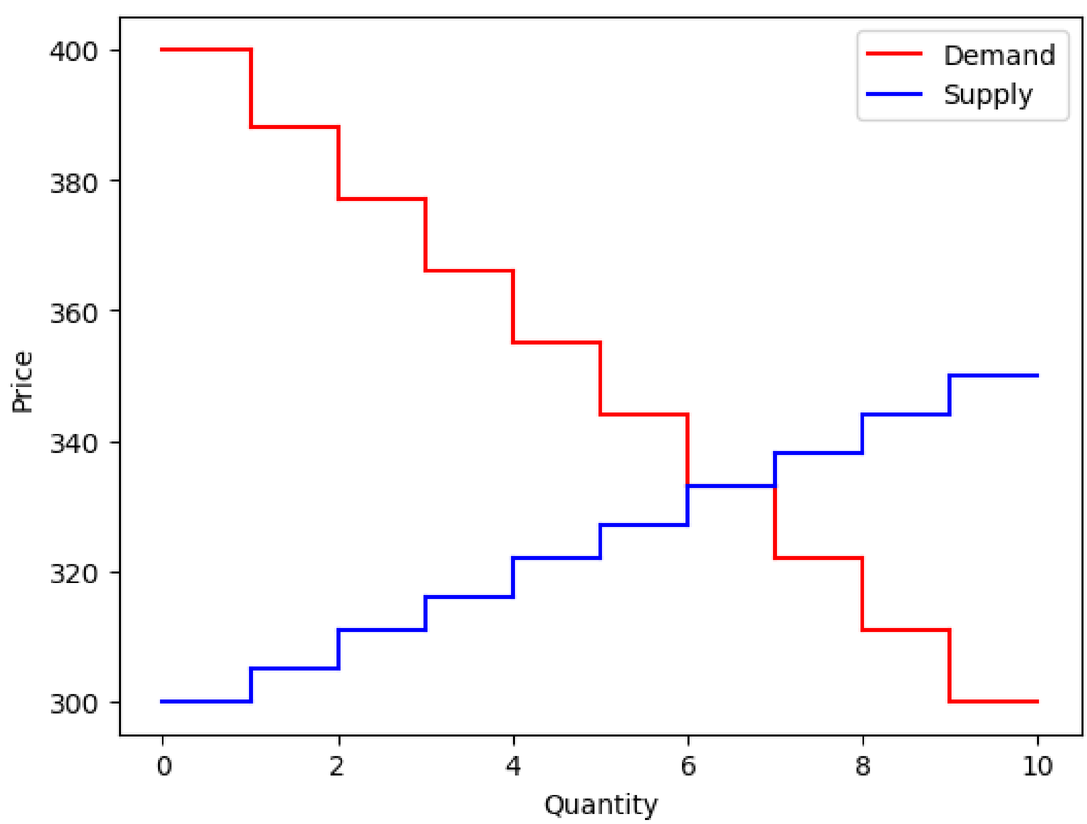
Next, we aim to see how ZIPSH adapts to more realistic market conditions by:

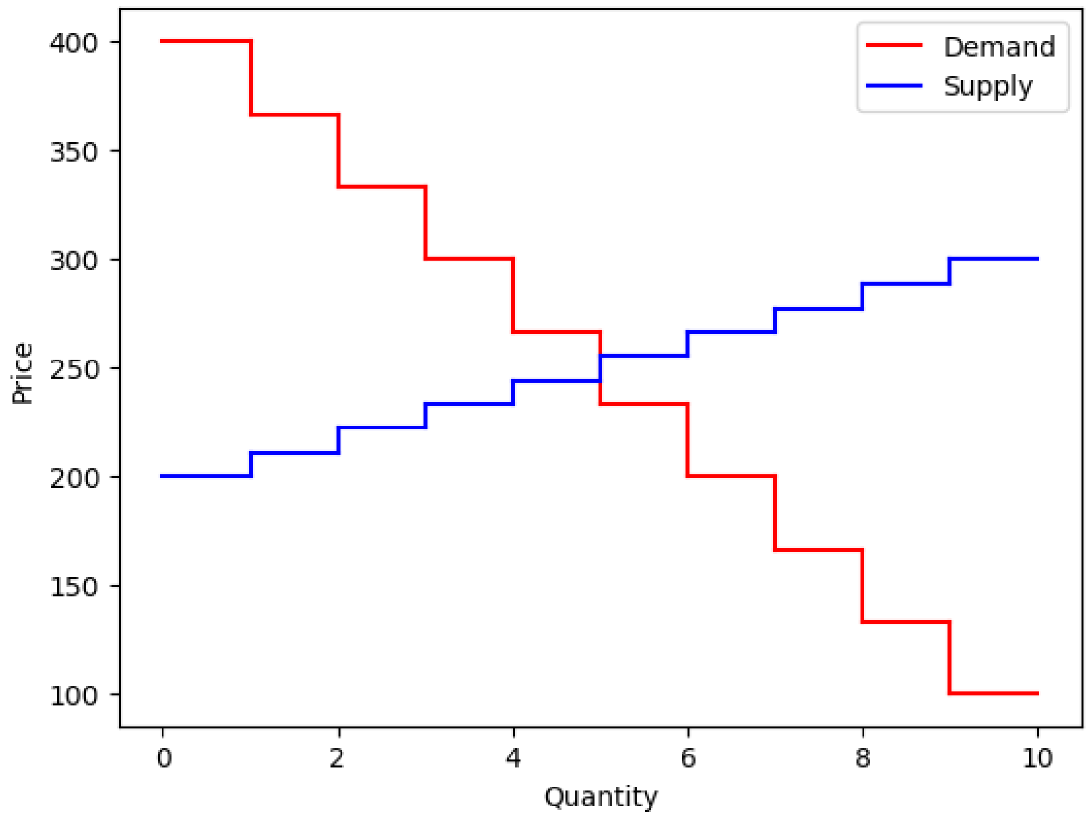
1. Introducing market shocks to alter the supply-demand curve
2. Model the replenshing of orders as a Poisson process - similar to the way trader entry and exit is often modelled in real life.

```
# SETUP

N = 100

start_time = 0
end_time = 60 * 60 * 24 * 30

sellers_spec = [('ZIC', 10)]
buyers_spec = [('ZIPSH', 1, {'k': 4}), ('ZIC', 9)]
traders_spec = {'sellers':sellers_spec, 'buyers':buyers_spec}

dem_range = (250, 200)
sup_range = (50, 400)
dem_range_shocked = (400, 300)
sup_range_shocked = (300, 350)
dem_range_retracted = (400, 100)
sup_range_retracted = (200, 300)

shock_time = end_time / 3
retract_time = (end_time / 3) * 2

demand_schedule = [{'from': start_time, 'to': shock_time, 'ranges': 
[dem_range], 'stepmode': 'random'},
                       {'from': shock_time, 'to': retract_time, 
'ranges': [dem_range_shocked], 'stepmode': 'random'},
                       {'from': retract_time, 'to': end_time, 
'ranges': [dem_range_retracted], 'stepmode': 'random'},
                       ]

supply_schedule = [{'from': start_time, 'to': shock_time, 'ranges': 
[sup_range], 'stepmode': 'random'},
                       {'from': shock_time, 'to': retract_time,
```

```
'ranges': [sup_range_shocked], 'stepmode': 'random'},
                       {'from': retract_time, 'to': end_time,
'ranges': [sup_range_retracted], 'stepmode': 'random'},
]

order_interval = 120
order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
            'interval': order_interval, 'timemode': 'drip-poisson'}

trial_id = 'test_tri_shock_N=50_R=1:9'

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

verbose = False

#n_runs_plot_trades(N, trial_id, start_time, end_time, traders_spec,
order_sched, dump_flags, verbose)

# Plotting the supply-demand curves
plot_sup_dem(10, [sup_range], 10, [dem_range], 'fixed')
plot_sup_dem(10, [sup_range_shocked], 10, [dem_range_shocked],
'fixed')
plot_sup_dem(10, [sup_range_retracted], 10, [dem_range_retracted],
'fixed')
```


Demand
Supply
Price
Quantity
400
350
300
250
200
150
100
50
0
2
4
6
8
10


Demand
Supply
Price
Quantity
400
380
360
340
320
300
0
2
4
6
8
10



Each shock changes the equilibrium price, possibly requiring ZIC to alter its strategy. The SD curves have been chosen such that each shock also changes the number of intramarginal traders in the market. The Poisson model of tader replenishment also gives further realistic stochascisity in trader entry/exits.

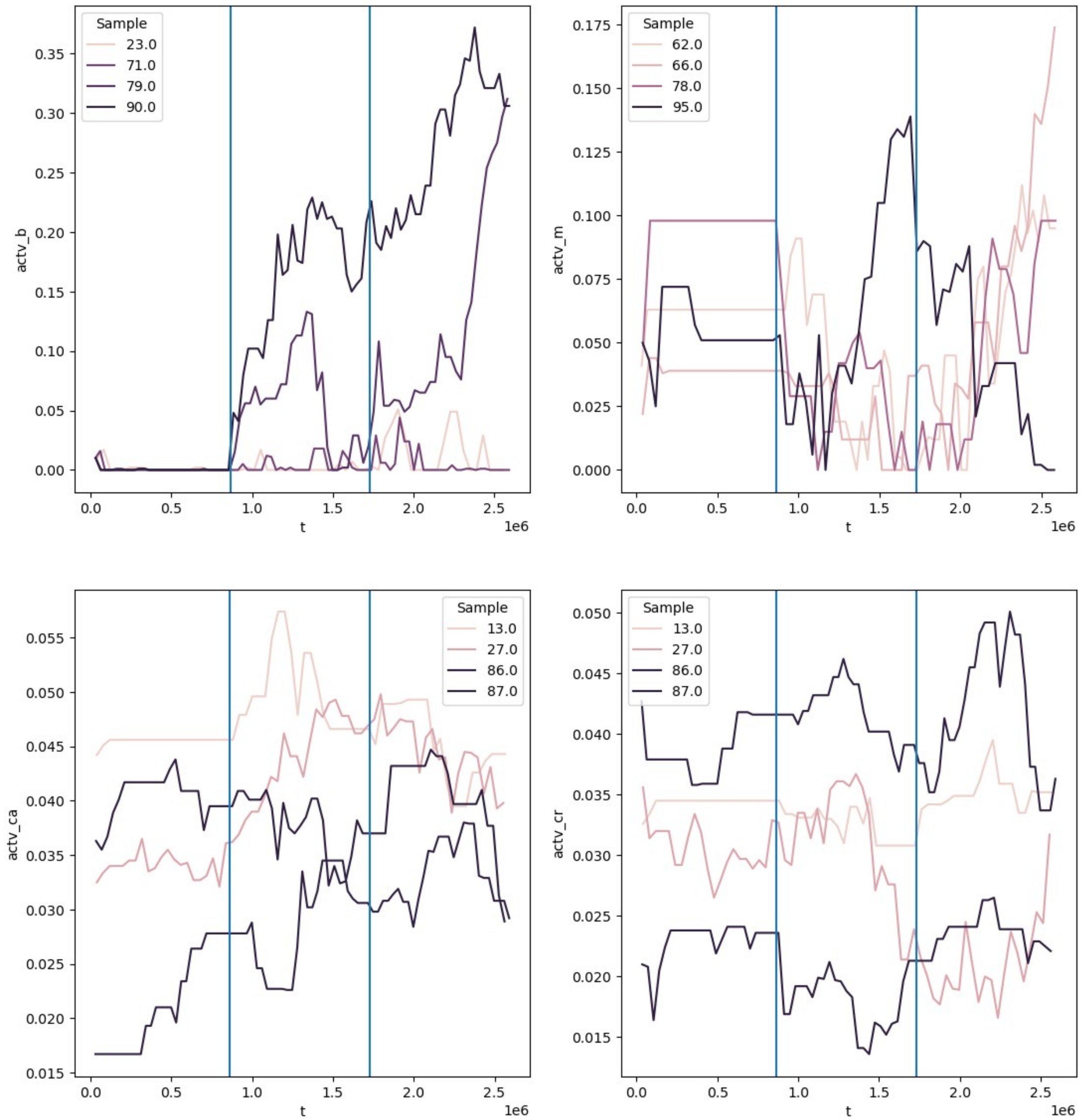
Once we have our results, we now take a sample and see how the hyperparameters vary, particularly at the marked shock points

```
df_strat_comp, df_half_paired, df_quart_prof, df_sample_hyp_prof_arr =
load_strat_data('./tri_shock/test_tri_shock_N=50_R=1_9',
remove_outliers=True)

print(df_sample_hyp_prof_arr[0].head(5))

f, axes = plt.subplots(2, 2, figsize=(13, 14))
#sns.boxplot(data=df_quart_prof, orient='h', ax=axes[0])
#sns.violinplot(data=df_quart_prof,  split=True, gap=-.25,
inner="quart", orient='x', ax=axes[0,0])
#sns.violinplot(data=df_quart_prof, split=True, gap=-.25,
inner="quart", orient='x', ax=axes[1])
ax = sns.lineplot(df_sample_hyp_prof_arr[0], x='t', y='actv_b',
hue='Sample', ax=axes[0,0])
ax.axvline(x = (60*60*24*30)/3, ymin = 0, ymax = 1)
ax.axvline(x = (60*60*24*30*2)/3, ymin = 0, ymax = 1)
```

```
ax = sns.lineplot(data=df_sample_hyp_prof_arr[1], x='t', y='actv_m', hue='Sample', ax=axes[0,1])
ax.axvline(x = (60*60*24*30)/3, ymin = 0, ymax = 1)
ax.axvline(x = (60*60*24*30*2)/3, ymin = 0, ymax = 1)
ax = sns.lineplot(data=df_sample_hyp_prof_arr[2], x='t', y='actv_ca', hue='Sample', ax=axes[1,0])
ax.axvline(x = (60*60*24*30)/3, ymin = 0, ymax = 1)
ax.axvline(x = (60*60*24*30*2)/3, ymin = 0, ymax = 1)
ax = sns.lineplot(data=df_sample_hyp_prof_arr[2], x='t', y='actv_cr', hue='Sample', ax=axes[1,1])
ax.axvline(x = (60*60*24*30)/3, ymin = 0, ymax = 1)
ax.axvline(x = (60*60*24*30*2)/3, ymin = 0, ymax = 1)

#plt.grid()
```

	Sample	t	actvprof	actv_b	actv_m	actv_ca	actv_cr
0	23.0	40504.0	0.018070	0.010	0.060	0.0330	0.0367
1	23.0	81008.0	0.021592	0.017	0.007	0.0326	0.0383
2	23.0	121513.0	0.136956	0.000	0.051	0.0303	0.0393
3	23.0	162017.0	0.190964	0.000	0.051	0.0303	0.0393
4	23.0	202522.0	0.191111	0.000	0.051	0.0303	0.0393

```
<matplotlib.lines.Line2D at 0x7f73fa80c850>
```



We can see that an increase in the beta learning rate parameter weakly correlates with the shock points. Another valuable assessment could be analysis of the difference of profit dispersion ZIV vs ZIPSH about the shock points, as well average profitability during the rise and fall of the equilibrium or whether ZIPSH shows any anticipation of price movement like ASAD in [4]

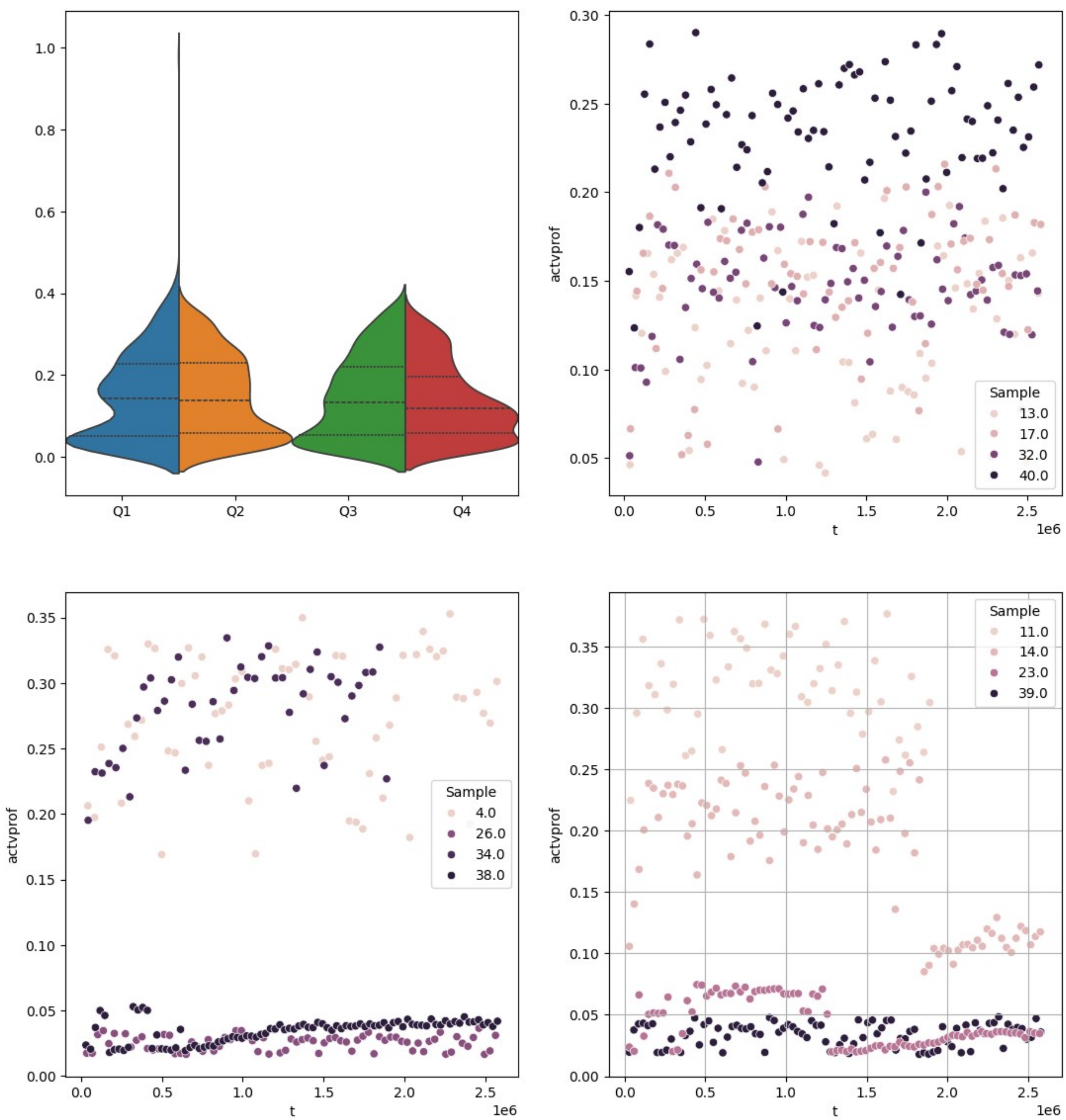
Now we will test the impact of both an increased number of traders and an increased number of candidate strategies for ZIPSH. The strategy evaluation time was halved to maintain the previous number of total strategy evaluations.

```
N = 50
start_time = 0
```

```
end_time = 60 * 60 * 24 * 30

sellers_spec = [('ZIC', 25)]
buyers_spec = [('ZIPSH', 1, {'optimizer': 'ZIPSH', 'k': 8}), ('ZIC',
24)]
traders_spec = {'sellers':sellers_spec, 'buyers':buyers_spec}

dem_range = (150, 125)
sup_range = (120, 125)

#plot_sup_dem(10, [sup_range], 10, [dem_range], 'fixed')

demand_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[dem_range], 'stepmode': 'fixed'}]
supply_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[sup_range], 'stepmode': 'fixed'}]

order_interval = 60
order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
            'interval': order_interval, 'timemode': 'periodic'}
trial_id = 'test_2k_more_zic_N=60_R=1:49'

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

verbose = False

n_runs_plot_trades(N, trial_id, start_time, end_time, traders_spec,
order_sched, dump_flags, verbose)


df_strat_comp, df_half_paired, df_quart_prof, df_sample_prof_arr =
load_strat_data('./data-d1/test_d1_N=50_R=1_9', N=50, dir='./2k-xzic-
strat/', remove_outliers=True)

print(df_half_paired.head(5))

f, axes = plt.subplots(2, 2, figsize=(13, 14))
#sns.boxplot(data=df_quart_prof, orient='h', ax=axes[0])
sns.violinplot(data=df_quart_prof,  split=True, gap=-.25,
inner="quart", orient='x', ax=axes[0,0])
plt.grid()
#sns.violinplot(data=df_quart_prof, split=True, gap=-.25,
inner="quart", orient='x', ax=axes[1])
sns.scatterplot(data=df_sample_prof_arr[0], x='t', y='actvprof',
hue='Sample', ax=axes[0,1])
plt.grid()
sns.scatterplot(data=df_sample_prof_arr[1], x='t', y='actvprof',
hue='Sample', ax=axes[1,0])
plt.grid()
```

```
sns.scatterplot(data=df_sample_prof_arr[2], x='t', y='actvprof',
hue='Sample', ax=axes[1,1])
plt.grid()

plt.grid()
```

	First half	Second half	Difference
0	0.262667	0.373288	0.110621
1	0.314656	0.336727	0.022071
2	0.324959	0.359062	0.034103
3	0.387517	0.337968	-0.049549
4	0.429745	0.351880	-0.077865



We can see a much less reliable pps increase and this is confirmed in the subsequent test.

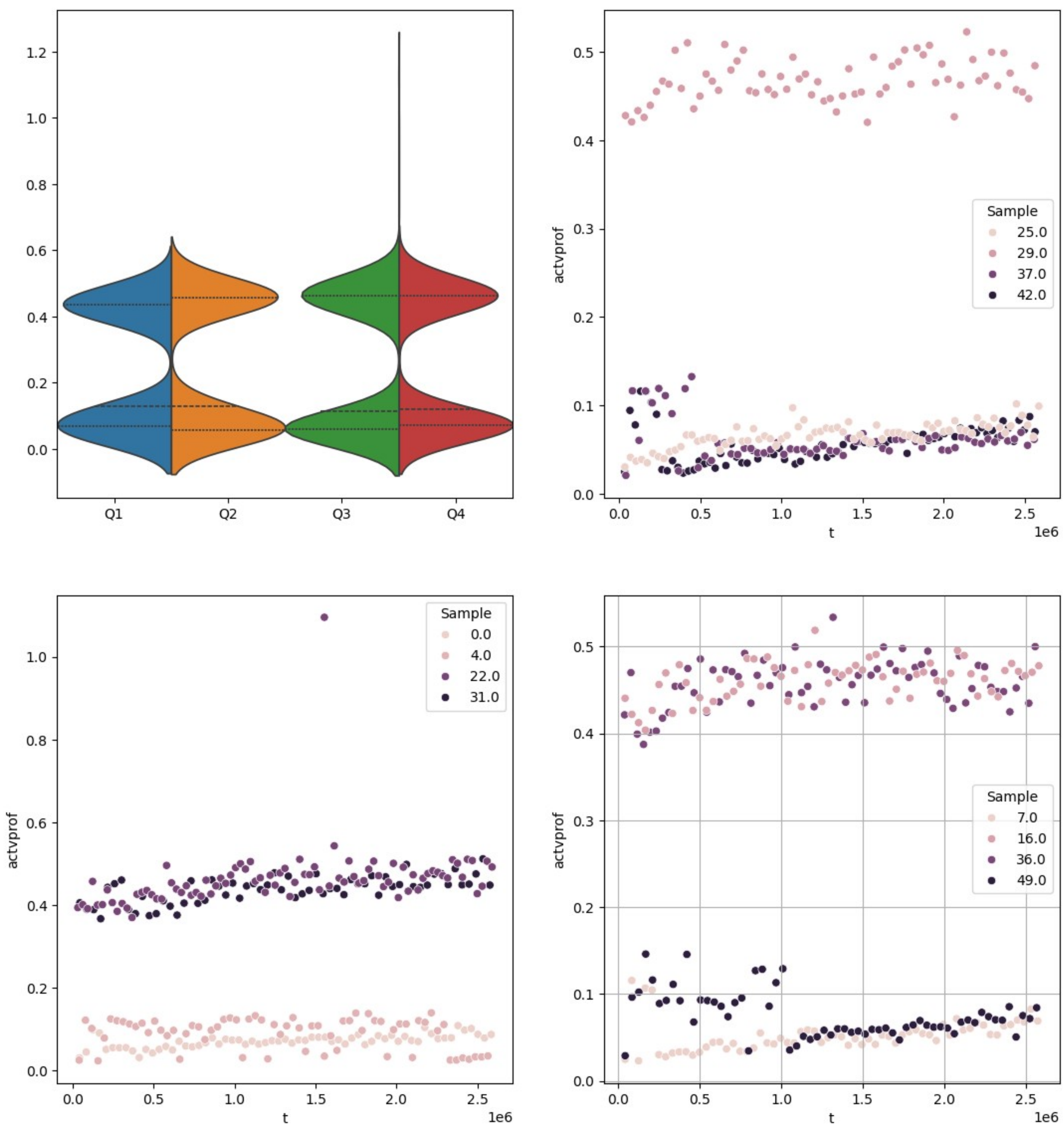
```
wilcoxon_signed_rank_test(df_half_paired['Difference'], 'First Half',
'Second Half')

Using the Wilcoxon signed rank test
Condition Second Half. We cannot reject the null hypothesis
(p=0.99877806). Therefore, profitability of First Half and Second Half
is statistically indistinguishable
```

Conversely, running again with ZIPSH:ZIC 1:3

```
N = 50
start_time = 0
end_time = 60 * 60 * 24 * 30

sellers_spec = [('ZIC', 25)]
buyers_spec = [('ZIPSH', 1, {'optimizer': 'ZIPSH', 'k': 8}), ('ZIC',
24)]
traders_spec = {'sellers':sellers_spec, 'buyers':buyers_spec}

dem_range = (150, 125)
sup_range = (120, 125)

#plot_sup_dem(10, [sup_range], 10, [dem_range], 'fixed')

demand_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[dem_range], 'stepmode': 'fixed'}]
supply_schedule = [{'from': start_time, 'to': end_time, 'ranges':
[sup_range], 'stepmode': 'fixed'}]

order_interval = 60
order_sched = {'sup': supply_schedule, 'dem': demand_schedule,
            'interval': order_interval, 'timemode': 'periodic'}
trial_id = 'test_2k_more_zic_N=60_R=1:49'

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

verbose = False

n_runs_plot_trades(N, trial_id, start_time, end_time, traders_spec,
order_sched, dump_flags, verbose)

df_strat_comp, df_half_paired, df_quart_prof, df_sample_prof_arr =
load_strat_data('./2k-lzic/test_2k_less_zic_N=50_R=1:3', N=50,
remove_outliers=True)

print(df_half_paired.head(5))

f, axes = plt.subplots(2, 2, figsize=(13, 14))
#sns.boxplot(data=df_quart_prof, orient='h', ax=axes[0])
sns.violinplot(data=df_quart_prof,  split=True, gap=-.25,
inner="quart", orient='x', ax=axes[0,0])
plt.grid()
#sns.violinplot(data=df_quart_prof, split=True, gap=-.25,
inner="quart", orient='x', ax=axes[1])
sns.scatterplot(data=df_sample_prof_arr[0], x='t', y='actvprof',
hue='Sample', ax=axes[0,1])
plt.grid()
```

```python
sns.scatterplot(data=df_sample_prof_arr[1], x='t', y='actvprof',
hue='Sample', ax=axes[1,0])
plt.grid()
sns.scatterplot(data=df_sample_prof_arr[2], x='t', y='actvprof',
hue='Sample', ax=axes[1,1])
plt.grid()

plt.grid()
```

	First half	Second half	Difference
0	0.030675	0.075538	0.044863
1	0.044762	0.066326	0.021564
2	0.098612	0.074179	-0.024433
3	0.090757	0.072977	-0.017780
4	0.051098	0.080703	0.029605




```
wilcoxon_signed_rank_test(df_half_paired['Difference'], 'First Half',
'Second Half')
```

```
Using the Wilcoxon signed rank test
Condition First Half. We can reject the null hypothesis
(p=0.00000000). Therefore, First Half and Second Half have
statistically different profitabilities
```

We can conclude that, a moderate (10-20) ratio of ZICs to ZIPSH is a more suitable range for adaptation - many more ZICS produce too much noise and many less means there is less price action for ZIPSH to adapt to.


```
#References

#[1] COMSM0140: Internet Economics and Financial Technology (IEFT)
Coursework Specification

#[2] Mishra P, Pandey CM, Singh U, Gupta A, Sahu C, Keshri A.
Descriptive statistics and normality tests for statistical data. Ann
Card Anaesth. 2019 Jan-Mar;22(1):67-72. doi: 10.4103/aca.ACA_157_18.
PMID: 30648682; PMCID: PMC6350423.

#[3] D. Cliff and M. Rollins, "Methods Matter: A Trading Agent with No
Intelligence Routinely Outperforms AI-Based Traders," 2020 IEEE
Symposium Series on Computational Intelligence (SSCI), Canberra, ACT,
Australia, 2020, pp. 392-399, doi: 10.1109/SSCI47803.2020.9308172.

#[4] Stotter, S., Cartlidge, J., & Cliff, D. (2013, February).
Exploring Assignment-Adaptive (ASAD) Trading Agents in Financial
Market Experiments. In ICAART (1) (pp. 77-88).
```


END OF REPORT. ONLY WORD COUNT BELOW THIS POINT.

```
# 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().split(' '))
    print("Word count: " + str(word_count) + ". Limit is 2000 words.")

# 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.ipynb" # Enter name of this file here
printWordCount(this_file_name)

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