stocks

November 11, 2022

1 Stock Trades by Members of the US House of Representatives

This project uses public data about the stock trades made by members of the US House of Representatives. This data is collected and maintained by Timothy Carambat as part of the House Stock Watcher project. The project describes itself as follows:

> With recent and ongoing investigations of incumbent congressional members being investigated for potentially violating the STOCK act. This website compiles this publicly available information in a format that is easier to digest then the original PDF source.
>
> Members of Congress must report periodic reports of their asset transactions. This website is purely for an informative purpose and aid in transparency.
>
> This site does not manipluate or censor any of the information from the original source. All data is transcribed by our community of contributors, which you can join for free by going to our transcription tool. Our moderation team takes great care in ensuring the accuracy of the information.
>
> This site is built and maintained by Timothy Carambat and supported with our contributors.

Some interesting questions to consider for this data set include:

- Is there a difference in stock trading behavior between political parties? For example:
  - does one party trade more often?
  - does one party make larger trades?
  - do the two parties invest in different stocks or sectors? For instance, do Democrats invest in Tesla more than Republicans?
- What congresspeople have made the most trades?
- What companies are most traded by congresspeople?
- Is there evidence of insider trading? For example, Boeing stock dropped sharply in February 2020. Were there a suspiciously-high number of sales of Boeing before the drop?
- When are stocks bought and sold? Is there a day of the week that is most common? Or a month of the year?

1.0.1 Getting the Data

The full data set of stock trade disclosures is available as a CSV or as JSON at https://housestockwatcher.com/api.

This data set does not, however, contain the political affiliation of the congresspeople. If you wish to investigate a question that relies on having this information, you'll need to find another dataset

that contains it and perform a merge. *Hint*: Kaggle is a useful source of data sets.

[3]:
```
stocks_df=pd.read_csv('all_transactions.csv')
stocks_df
```

[3]:

	disclosure_year	disclosure_date	transaction_date	owner	ticker	\
0	2021	10/04/2021	2021-09-27	joint	BP	
1	2021	10/04/2021	2021-09-13	joint	XOM	
2	2021	10/04/2021	2021-09-10	joint	ILPT	
3	2021	10/04/2021	2021-09-28	joint	PM	
4	2021	10/04/2021	2021-09-17	self	BLK	
…	…	…	…	…	…	
15694	2020	06/10/2020	2020-04-09	--	SWK	
15695	2020	06/10/2020	2020-04-09	--	USB	
15696	2020	06/10/2020	2020-03-13	NaN	BMY	
15697	2020	06/10/2020	2020-03-13	NaN	LLY	
15698	2020	06/10/2020	2020-03-13	NaN	DIS	

	asset_description	type	\
0	BP plc	purchase	
1	Exxon Mobil Corporation	purchase	
2	Industrial Logistics Properties Trust - Common…	purchase	
3	Phillip Morris International Inc	purchase	
4	BlackRock Inc	sale_partial	
…	…	…	
15694	Stanley Black & Decker, Inc.	sale_partial	
15695	U.S. Bancorp	sale_partial	
15696	Bristol-Myers Squibb Company	sale_full	
15697	Eli Lilly and Company	sale_full	
15698	Walt Disney Company	sale_full	

	amount	representative	district	\
0	$1,001 - $15,000	Hon. Virginia Foxx	NC05	
1	$1,001 - $15,000	Hon. Virginia Foxx	NC05	
2	$15,001 - $50,000	Hon. Virginia Foxx	NC05	
3	$15,001 - $50,000	Hon. Virginia Foxx	NC05	
4	$1,001 - $15,000	Hon. Alan S. Lowenthal	CA47	
…	…	…	…	
15694	$1,001 - $15,000	Hon. Ed Perlmutter	CO07	
15695	$1,001 - $15,000	Hon. Ed Perlmutter	CO07	
15696	$100,001 - $250,000	Hon. Nicholas Van Taylor	TX03	
15697	$500,001 - $1,000,000	Hon. Nicholas Van Taylor	TX03	
15698	$250,001 - $500,000	Hon. Nicholas Van Taylor	TX03	

	ptr_link	\
0	https://disclosures-clerk.house.gov/public_dis…	
1	https://disclosures-clerk.house.gov/public_dis…	

```
2      https://disclosures-clerk.house.gov/public_dis…
3      https://disclosures-clerk.house.gov/public_dis…
4      https://disclosures-clerk.house.gov/public_dis…
…                                                    …
15694  https://disclosures-clerk.house.gov/public_dis…
15695  https://disclosures-clerk.house.gov/public_dis…
15696  https://disclosures-clerk.house.gov/public_dis…
15697  https://disclosures-clerk.house.gov/public_dis…
15698  https://disclosures-clerk.house.gov/public_dis…

       cap_gains_over_200_usd
0                       False
1                       False
2                       False
3                       False
4                       False
…                       …
15694                   False
15695                   False
15696                   False
15697                   False
15698                   False

[15699 rows x 12 columns]
```

1.0.2 Cleaning and EDA

- Clean the data.
  - Certain fields have "missing" data that isn't labeled as missing. For example, there are fields with the value "–." Do some exploration to find those values and convert them to null values.
  - You may also want to clean up the date columns to enable time-series exploration.
- Understand the data in ways relevant to your question using univariate and bivariate analysis of the data as well as aggregations.

1.0.3 Assessment of Missingness

- Assess the missingness per the requirements in `project03.ipynb`

1.0.4 Hypothesis Test / Permutation Test

Find a hypothesis test or permutation test to perform. You can use the questions at the top of the notebook for inspiration.

2 Summary of Findings

2.0.1 Introduction

Stocks Data

Dataset Name: Stock Trades by Members of the US House of Representatives

Link to Dataset: https://housestockwatcher.com/api

Number of Observations: 15699

The recent COVID-19 pandemic has resulted in substantial changes to the economy including the stock market. For example, according to the AICPA, "In comparison to its Q4 2019 record high, the market index was down 21 percent (20.9 points) during the pandemic, wiping out all its gains from the past three years" (1). In this project, we further analyze how the pandemic affected the stock market and aim to answer the question "How has the COVID-19 pandemic affected the stock market through the capital gains generated from the difference between selling price and purchasing price of the stock, i.e., the proportion of stocks which achieved capital gains over 200 usd in the years ranging from 2018-2022?" In order to answer this question, we chose to analyze the "Stock Trades by Members of the US House of Representatives" data set which provides various information about stock transactions made by members of the US House of Representatives. More specifically, some of the important information it provides includes information about the year a US House of Representatives member bought/sold/exchanged a stock (the transaction date), their ownership type (self, joint, or dependent), the company they bought the stock from, the type of transaction they made (whether they purchased, sold, or exchanged a stock), the amount of shares they bought, the district they represent, and whether or not they made a capital gain over 200 usd. This information helps to address our research question because with the transaction_date column ranging from pre-pandemic to post-pandemic years and capital_gains_over_200_usd column, the effect of the pandemic on the stock market during pre-pandemic, pandemic, and post-pandemic periods can be analyzed. In addition, analysis of the other columns can potentially reveal more trends which can help in answering our research question but the most important columns of the dataframe for our analysis will be the transaction_date and cap_gains_over_200_usd columns. A complete description of all 12 columns in the data frame are listed below.

Note: In this project, we are considering dates from 2018-2019 pre-pandemic, 2020-2021 during the pandemic, and 2022 post-pandemic.

Column Descriptions: - disclosure_year: The year in which a company released important information about itself which may influence stock buyer's decisions. - disclosure_date: The date on which a company released important information about itself which may influence stock buyer's decisions. - transaction_date: The date that a stock was purchased/sold/exchanged. - owner: The type of ownership the US House of Representative had on a stock. Joint means that the stock is owned by multiple people, self means that the stock is only owned by one person - ticker: The symbol for the company - asset_description: The full name of the ticker (the company name) - type: The type of transaction the US House of Representative made (either a purchase, sale, or exchange) - amount: The amount of shares purchased/sold/exchanged - representative: The congress member who bought the stock - district: The district the congress member is representing - ptr_link: A link to where the data came from - cap_gains_over_200_usd: Whether or not someone made capital gains over 200 usd on their stock. (True indicates that they did make a gain over 200 usd, False indicates that they did not).

(1) https://www.aicpa.org/news/article/coronavirus-causes-largest-drop-in-americans-financial-satisfaction#:~:text=Stock%20Market%20Decline%20Drives%20Drop&text=In%20comparison%20to%20its

2.0.2 Cleaning and EDA

Cleaning:

These are the following steps we took in order to clean our data:

1) Dropping the missing values We did this because there were various quantitative and qualitative values for which we couldn't assume any replacement value.

2) Dropping the disclosure_data and ptr_link columns We did this because these columns did not contribute to addressing our research question.

3) Adding a transaction_year column (which just contained the years from the transaction_date column) We did this because if we used the transaction_date values themselves, there wasn't enough data to make any concrete claims but by doing our analysis using only the year, we were able to garner sufficient unique data to conduct analyze and produce valid results to address our research question.

4) Dropping the transaction_date column We did this because as mentioned, we only used the transaction years so after we created the transaction_year column, we dropped the transaction_date column.

5) Restrict the years from 2018-2022 (inclusive) We did this because there was too much pre-pandemic data (2016 and 2017) and not proportionally enough pandemic and post-pandemic data so leaving in those years would have skewed the analyses/results. Also, we wanted to try and keep the data relatively recent so that the results are more accurate for current times.

6) Removing any duplicate values We did this because the duplicates would lead to a problem of double counting (i.e. considering the same data more than once) which could lead to a faulty conclusion.

These steps follow the data generating process and ensure data quality because they assure that any irrelevant data is not included in our analysis and more importantly, there is no faulty data that could potentially incorrectly affect our results.

EDA:

Specific details on our EDA (including univariate analyses, bivariate analyses, and different aggregates) are included in markdown cells in the EDA code section below.

2.0.3 Assessment of Missingness

Describing the setup/results for the assessment of missingness:

In order to assess the missingness, we ran a standard permutation test.

Type of missingness/results:

From our tests, we got the following p-values:

- The p-value between the cap_gains_over_200_usd and owner columns was 0

Since the p-value was less than our alpha value of 0.05, we failed to reject the null hypothesis indicating that there is a dependency between the cap_gains_over_200_usd and owner columns.

- The p-value between the type and owner columns was 0

Since the p-value was less than our alpha value of 0.05, we failed to reject the null hypothesis indicating that there is a dependency between the type and owner columns.

- The p-value between the owner and district columns was 0.998

Since the p-value was greater than our alpha value of 0.05, we rejected the null hypothesis indicating that there is no dependency between the owner and district columns.

Overall, because of the dependency between the cap_gains_over_200_usd and owner columns, and type and owner columns, we can say that the data is not missing at random (NMAR).

How the missingness affects our ability to answer questions about the dataset:

The missingness results may affect our ability to answer questions about the dataset because since there is a dependency between the cap_gains_over_200_usd and owner columns, and type and owner columns, any altering of one of those column can inadvertently affect the other column potentially affecting our analysis.

2.0.4 Hypothesis Test

Describing the setup/results of the hypothesis test:

In our project, we hypothesized that there were more gains over 200 usd during the pandemic in comparison to normal times than after the pandemic in comparison to normal times (before pandemic). This is because stocks were cheaper during the pandemic. For example, according to Regions, a wealth and investments site, "As COVID-19 spread across the world with unprecedented speed, consumer and investor behavior dramatically shifted. Triggered by massive selloffs, the major stock indexes plummeted" (2). Our null and alternate hypothesis for pandemic and post-pandemic periods are stated below:

HYPOTHESIS FOR DURING THE PANDEMIC:

Our null/alternative hypothesis:

Ho: p = 0.0635 Ha: p > 0.0635

Our test-statistic:

-0.3744 ###### Our significance level: alpha = 0.05 (5% significance level) ###### Our p-value: 0.7081 ###### The conclusions we drew from the results: Since the p-value is greater than the significance level of 5%, we fail to reject our null hypothesis that there was no difference between the proportion of stocks with capital gains over 200 usd during the pre-pandemic period and normal times (before the pandemic).

HYPOTHESIS FOR POST-PANDEMIC:

Our null/alternative hypothesis:

Ho: p = 0.0635 Ha: p < 0.0635

Our test-statistic:

-57.2607 ###### Our significance level: alpha = 0.05 (5% significance level) ###### Our p-value: approximately 0 ###### The conclusions we drew from the results: Since the p-value is less than the significance level of 5%, we reject our null hypothesis that there was no difference between the proportion of stocks with capital gains over 200 usd during the post-pandemic period and normal times (before the pandemic). ###### Reasoning: The steps we described above helped to answer our research question because we were able to conduct a robust hypothesis test using a one-proportion z-test and were able to determine if there were any statistically significant effects of the pandemic on the capital gains from stocks.

Limitations:

One limitation we faced in answering our research question is that our data set did not adequately represent the population we were looking to analyze. This is because our research question was aiming to look at the effect of the pandemic on stocks for the general public - not just US House of Representative members. Another limitation we faced is that within each year, there was some discrepancy in terms of the number of stocks present so the years with fewer stocks would output values which weren't as representative of the population as would be observed in other years where the sample was large enough. Thus, applying the results of the data frame onto the overall market may not be feasible.

(2) https://www.regions.com/insights/wealth/investments-and-markets/developing-and-adjusting-your-strategy/covid-19-and-your-investment-portfolio

3 Code

[8]:
```
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
import seaborn as sns
%matplotlib inline
%config InlineBackend.figure_format = 'retina'  # Higher resolution figures
```

3.0.1 Cleaning and EDA

[168]:
```
#Cleaning:
#Drop NA Values:
stocks_df=pd.read_csv('all_transactions.csv')#load dataframe
stocks_dropped=stocks_df.dropna()#dropping na values from the dataframe
stocks_cols=stocks_dropped.drop(['ptr_link','disclosure_date'],axis=1)#dropping␣
 ↪the unnecessary columns in dataframe
stock2=stocks_cols[stocks_cols['owner']!='--']#removing missing values from␣
 ↪owner column
stock=stock2[stock2['ticker']!='--']#removing missing values from ticker column
stock.head()
```

[168]:
```
   disclosure_year transaction_date  owner ticker  \
0             2021       2021-09-27  joint     BP
```

```
1             2021       2021-09-13  joint    XOM
2             2021       2021-09-10  joint   ILPT
3             2021       2021-09-28  joint     PM
4             2021       2021-09-17   self    BLK

                                asset_description          type  \
0                                           BP plc      purchase
1                             Exxon Mobil Corporation      purchase
2  Industrial Logistics Properties Trust - Common…      purchase
3                    Phillip Morris International Inc      purchase
4                                       BlackRock Inc  sale_partial

              amount          representative district  cap_gains_over_200_usd
0   $1,001 - $15,000      Hon. Virginia Foxx     NC05                   False
1   $1,001 - $15,000      Hon. Virginia Foxx     NC05                   False
2  $15,001 - $50,000      Hon. Virginia Foxx     NC05                   False
3  $15,001 - $50,000      Hon. Virginia Foxx     NC05                   False
4   $1,001 - $15,000  Hon. Alan S. Lowenthal     CA47                   False
```

[167]:
```
#Create Transaction Year:
yearst=stock['transaction_date'].str[0:4]#producing just years from date strings
stock_copy=stock.copy()
stock_copy['transaction_year']=yearst #create transaction_year column
int_year=stock_copy.copy()
int_year['transaction_year']=stock_copy['transaction_year'].astype(int)#convert␣
 ↪transaction_year column to integer values
stock_df=int_year.drop(['transaction_date'],axis=1)#dropping transaction_date␣
 ↪column
stock_df.head()
```

[167]:
```
   disclosure_year  owner ticker  \
0             2021  joint     BP
1             2021  joint    XOM
2             2021  joint   ILPT
3             2021  joint     PM
4             2021   self    BLK

                                asset_description          type  \
0                                           BP plc      purchase
1                             Exxon Mobil Corporation      purchase
2  Industrial Logistics Properties Trust - Common…      purchase
3                    Phillip Morris International Inc      purchase
4                                       BlackRock Inc  sale_partial

              amount          representative district  cap_gains_over_200_usd  \
0   $1,001 - $15,000      Hon. Virginia Foxx     NC05                   False
1   $1,001 - $15,000      Hon. Virginia Foxx     NC05                   False
```

```
2  $15,001 - $50,000      Hon. Virginia Foxx     NC05                    False
3  $15,001 - $50,000      Hon. Virginia Foxx     NC05                    False
4   $1,001 - $15,000  Hon. Alan S. Lowenthal     CA47                    False

   transaction_year
0              2021
1              2021
2              2021
3              2021
4              2021
```

[169]:
```
#Restrict Years in df:
stock_rest=stock_df[(stock_df['transaction_year']>=2018)&(stock_df['transaction_year']<=2022)]
 ↪the years of transaction_year
stock_rest.head()
```

[169]:
```
   disclosure_year  owner ticker  \
0             2021  joint     BP
1             2021  joint    XOM
2             2021  joint   ILPT
3             2021  joint     PM
4             2021   self    BLK

                                asset_description          type  \
0                                         BP plc      purchase
1                           Exxon Mobil Corporation      purchase
2  Industrial Logistics Properties Trust - Common…      purchase
3                  Phillip Morris International Inc      purchase
4                                     BlackRock Inc  sale_partial

              amount          representative district  cap_gains_over_200_usd  \
0   $1,001 - $15,000      Hon. Virginia Foxx     NC05                    False
1   $1,001 - $15,000      Hon. Virginia Foxx     NC05                    False
2  $15,001 - $50,000      Hon. Virginia Foxx     NC05                    False
3  $15,001 - $50,000      Hon. Virginia Foxx     NC05                    False
4   $1,001 - $15,000  Hon. Alan S. Lowenthal     CA47                    False

   transaction_year
0              2021
1              2021
2              2021
3              2021
4              2021
```

[170]:
```
#Remove Duplicates From the Dataframe:
stocks=stock_rest.drop_duplicates()#dropping duplicates
stocks.head()
```

[170]:
```
   disclosure_year  owner ticker  \
0             2021  joint     BP
1             2021  joint    XOM
2             2021  joint   ILPT
3             2021  joint     PM
4             2021   self    BLK

                                asset_description          type  \
0                                           BP plc      purchase
1                              Exxon Mobil Corporation      purchase
2  Industrial Logistics Properties Trust - Common…      purchase
3                    Phillip Morris International Inc      purchase
4                                        BlackRock Inc  sale_partial

              amount          representative district  cap_gains_over_200_usd  \
0   $1,001 - $15,000      Hon. Virginia Foxx     NC05                   False
1   $1,001 - $15,000      Hon. Virginia Foxx     NC05                   False
2  $15,001 - $50,000      Hon. Virginia Foxx     NC05                   False
3  $15,001 - $50,000      Hon. Virginia Foxx     NC05                   False
4   $1,001 - $15,000  Hon. Alan S. Lowenthal     CA47                   False

   transaction_year
0              2021
1              2021
2              2021
3              2021
4              2021
```

[171]:
```
#Final Cleaned Dataframe:
stocks.head()
```

[171]:
```
   disclosure_year  owner ticker  \
0             2021  joint     BP
1             2021  joint    XOM
2             2021  joint   ILPT
3             2021  joint     PM
4             2021   self    BLK

                                asset_description          type  \
0                                           BP plc      purchase
1                              Exxon Mobil Corporation      purchase
2  Industrial Logistics Properties Trust - Common…      purchase
3                    Phillip Morris International Inc      purchase
4                                        BlackRock Inc  sale_partial

              amount          representative district  cap_gains_over_200_usd  \
0   $1,001 - $15,000      Hon. Virginia Foxx     NC05                   False
```

```
1   $1,001 - $15,000      Hon. Virginia Foxx     NC05                    False
2  $15,001 - $50,000      Hon. Virginia Foxx     NC05                    False
3  $15,001 - $50,000      Hon. Virginia Foxx     NC05                    False
4   $1,001 - $15,000  Hon. Alan S. Lowenthal     CA47                    False

   transaction_year
0              2021
1              2021
2              2021
3              2021
4              2021
```

[163]:
```
#EDA:
stocks.describe()
```

[163]:
```
       disclosure_year  transaction_year
count      5611.000000       5611.000000
mean       2020.781857       2020.683657
std           0.749687          0.781868
min        2020.000000       2018.000000
25%        2020.000000       2020.000000
50%        2021.000000       2021.000000
75%        2021.000000       2021.000000
max        2022.000000       2022.000000
```

The above code is used to produce the descriptive statistics for quantitative columns of the dataframe

[164]:
```
stocks.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5611 entries, 0 to 15693
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   disclosure_year         5611 non-null   int64
 1   owner                   5611 non-null   object
 2   ticker                  5611 non-null   object
 3   asset_description       5611 non-null   object
 4   type                    5611 non-null   object
 5   amount                  5611 non-null   object
 6   representative          5611 non-null   object
 7   district                5611 non-null   object
 8   cap_gains_over_200_usd  5611 non-null   bool
 9   transaction_year        5611 non-null   int64
dtypes: bool(1), int64(2), object(7)
memory usage: 603.8+ KB
```

The above code is used to print information about the cleaned dataframe

[165]:
```
pre_pand=stocks[stocks['transaction_year']<=2019]
pre_pand.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 189 entries, 912 to 15325
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   disclosure_year        189 non-null    int64 
 1   owner                  189 non-null    object
 2   ticker                 189 non-null    object
 3   asset_description      189 non-null    object
 4   type                   189 non-null    object
 5   amount                 189 non-null    object
 6   representative         189 non-null    object
 7   district               189 non-null    object
 8   cap_gains_over_200_usd 189 non-null    bool  
 9   transaction_year       189 non-null    int64 
dtypes: bool(1), int64(2), object(7)
memory usage: 15.0+ KB
```

The above code is used to filter the dataframe to suit pre-pandemic stock data and then print out the information about that dataframe

[114]:
```
pre_pand.describe()
```

[114]:

	disclosure_year	transaction_year
count	189.000000	189.000000
mean	2020.148148	2018.994709
std	0.356190	0.072739
min	2020.000000	2018.000000
25%	2020.000000	2019.000000
50%	2020.000000	2019.000000
75%	2020.000000	2019.000000
max	2021.000000	2019.000000

The above code is used to produce descriptive statistics about pre-pandemic stocks data descriptive statistics

[111]:
```
dur_pand=stocks[(stocks['transaction_year']>=2020)&(stocks['transaction_year']<=2021)]
dur_pand.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4505 entries, 0 to 15693
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
```

```
 0   disclosure_year         4505 non-null   int64
 1   owner                   4505 non-null   object
 2   ticker                  4505 non-null   object
 3   asset_description       4505 non-null   object
 4   type                    4505 non-null   object
 5   amount                  4505 non-null   object
 6   representative          4505 non-null   object
 7   district                4505 non-null   object
 8   cap_gains_over_200_usd  4505 non-null   bool
 9   transaction_year        4505 non-null   int64
dtypes: bool(1), int64(2), object(7)
memory usage: 356.4+ KB
```

The above code is used to filter the dataframe to suit during pandemic stock data and then print out the information about that dataframe

[115]:
```
dur_pand.describe()
```

[115]:

	disclosure_year	transaction_year
count	4505.000000	4505.000000
mean	2020.560488	2020.486570
std	0.571637	0.499875
min	2020.000000	2020.000000
25%	2020.000000	2020.000000
50%	2021.000000	2020.000000
75%	2021.000000	2021.000000
max	2022.000000	2021.000000

The above code is used to produce descriptive statistics about in pandemic stocks data descriptive statistics

[119]:
```
post_pand=stocks[stocks['transaction_year']==2022]
post_pand.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 917 entries, 453 to 15574
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   disclosure_year         917 non-null    int64
 1   owner                   917 non-null    object
 2   ticker                  917 non-null    object
 3   asset_description       917 non-null    object
 4   type                    917 non-null    object
 5   amount                  917 non-null    object
 6   representative          917 non-null    object
 7   district                917 non-null    object
 8   cap_gains_over_200_usd  917 non-null    bool
 9   transaction_year        917 non-null    int64
```

```
dtypes: bool(1), int64(2), object(7)
memory usage: 72.5+ KB
```

The above code is used to filter the dataframe to suit post-pandemic stock data and then print out the information about that dataframe

[116]:
```
post_pand.describe()
```

[116]:

	disclosure_year	transaction_year
count	917.0	917.0
mean	2022.0	2022.0
std	0.0	0.0
min	2022.0	2022.0
25%	2022.0	2022.0
50%	2022.0	2022.0
75%	2022.0	2022.0
max	2022.0	2022.0

The above code is used to produce descriptive statistics about post-pandemic stocks data descriptive statistics

[87]:
```
grp_owner=stocks.groupby(['owner']).aggregate({'cap_gains_over_200_usd':
 ↪'sum','transaction_year':'count'})
grp_owner.rename(columns={'transaction_year':'counts'})
```

[87]:

owner	cap_gains_over_200_usd	counts
dependent	1.0	272
joint	162.0	3395
self	130.0	1944

Through the above code data is generated about the number of stocks which had more than $200 capital gains in the various categories of stock owners, and the total number of stocks present in each category of stock owners. Most of the stocks with capital gains over 200 dollars are in joint ownership, but most of the stocks are also present in joint ownership. Overall there are more stocks with greater than 200 dollars capital gains in self ownership by proportion than any other type of ownership.

[88]:
```
grp_type=stocks.groupby(['type']).aggregate({'cap_gains_over_200_usd':
 ↪'sum','transaction_year':'count'})
grp_type.rename(columns={'transaction_year':'counts'})
```

[88]:

type	cap_gains_over_200_usd	counts
exchange	0.0	40
purchase	3.0	2980
sale	0.0	1
sale_full	138.0	1676

```
sale_partial                    152.0      914
```

Through the above code data is generated about the number of stocks which had more than $200 capital gains in the various categories of stock types, and the total number of stocks present in each category of stock types. Most of the stocks with capital gains over 200 dollars are in sale_partial type, and most of the stocks are present in purchase type. Overall there are more stocks with greater than 200 dollars capital gains in sale_partial type by proportion than any other type of stock.

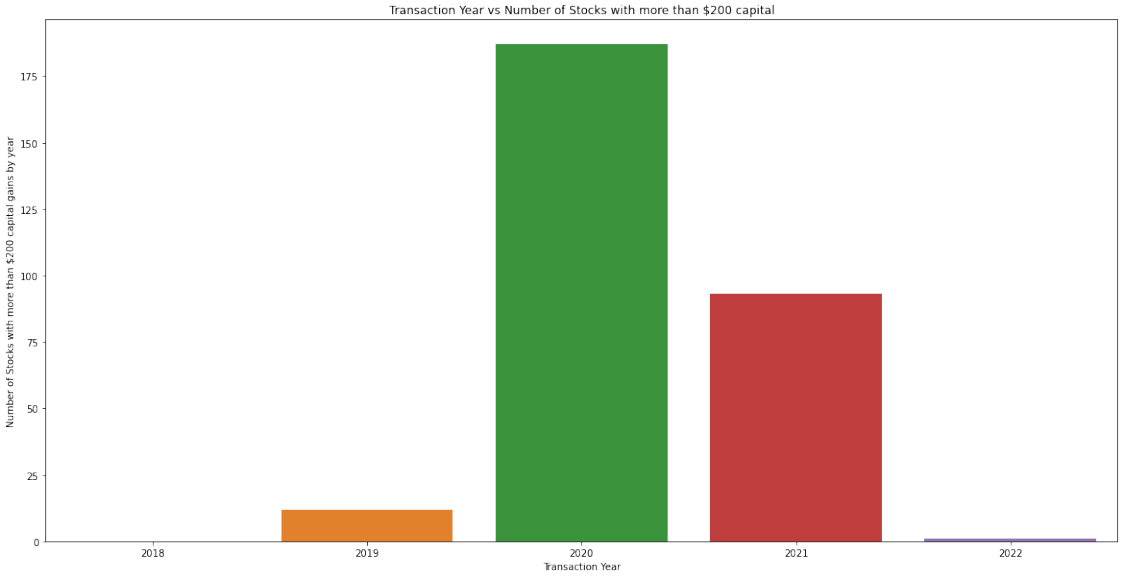
[89]:
```
grp_tnsyr=stocks.groupby(['transaction_year']).
 ↪aggregate({'cap_gains_over_200_usd':'sum'}).reset_index()
grp_tnsyr
```

[89]:

	transaction_year	cap_gains_over_200_usd
0	2018	0.0
1	2019	12.0
2	2020	187.0
3	2021	93.0
4	2022	1.0

Using the above code we group the dataframe based on the various transaction years present in our cleaned dataframe, and then calculate the number of stocks which resulted in capital gains over 200 dollars. It is observed that 2020 had the most stocks with capital gains over 200 dollars.

[174]:
```
plt.figure(figsize=(16,8))
sns.barplot(data=grp_tnsyr,x='transaction_year',y='cap_gains_over_200_usd');
plt.xlabel('Transaction Year')
plt.ylabel('Number of Stocks with more than $200 capital gains by year')
plt.title('Transaction Year vs Number of Stocks with more than $200 capital')
fig1 = plt.gcf()
fig1.set_size_inches(20, 10)
```



The above barplot depicts the distribution of number of stocks with capital gains over 200 dollars by the various years present in transaction_year column of the cleaned dataframe with 2020 having an abnormally large number of such stocks in comparison to other years.

[166]:
```
#Replacing the intervals in amount column to individual integers by considering␣
 ↪the midpoint of each of the intervals
ser=stocks['amount'].str.split(' - ')
lst=[]
for i in ser:
    if (len(i)==2):
        a=i[0].replace('$','')
        a1=a.replace(',','')
        b=i[1].replace('$','')
        b1=b.replace(',','')
        mid=(int(a1)+int(b1))/2
        lst.append(mid)
    elif (len(i)==1):
        c=i[0].replace('$','')
        c1=c.replace(',','')
        c2=c1.replace(' -','')
        lst.append(int(c2))
stocks_amts=stocks.copy()
stocks_amts['amount']=lst
```

[149]:
```
stocks_amts.groupby('owner').aggregate({'amount':'mean'})
```

[149]:
```
                    amount
owner
dependent    21210.058824
joint        55665.154934
self        107249.488169
```

The above code produces the mean amount of stocks purchased by each category of stock owners. It is observed that self-ownership stocks are purchased the most.

[150]:
```
stocks_amts.groupby('type').aggregate({'amount':'mean'})
```

[150]:
```
                     amount
type
exchange       101775.512500
purchase        74281.713255
sale             8000.500000
sale_full       68110.885442
sale_partial    69642.200766
```

The above code produces the mean amount of stocks purchased by each category of stock type. It is observed that exchange stock type are purchased the most.

[172]:
```
stocks.groupby(['transaction_year', 'amount']).
↪aggregate({'cap_gains_over_200_usd':'sum','owner':'count','type':'count'}).
↪head()
```

[172]:

		cap_gains_over_200_usd	owner	type
transaction_year	amount			
2018	$15,001 - $50,000	0.0	1	1
2019	$1,000,001 - $5,000,000	0.0	1	1
	$1,001 -	0.0	1	1
	$1,001 - $15,000	7.0	163	163
	$100,001 - $250,000	0.0	2	2

The above code is to conduct a multiple groupby of transaction year and then within each transaction year in the cleaned dataframe the various amount intervals present. Then the code counts the number of stocks with more than 200 dollars capital gains in each amount interval in each transaction_year as well as the number of stocks in each amount interval in each transaction year.

[157]:
```
stocks.groupby('district').aggregate({'cap_gains_over_200_usd':'sum'})
```

[157]:
```
          cap_gains_over_200_usd
district
AL05                         1.0
AR02                         1.0
AZ01                         0.0
AZ03                         0.0
```

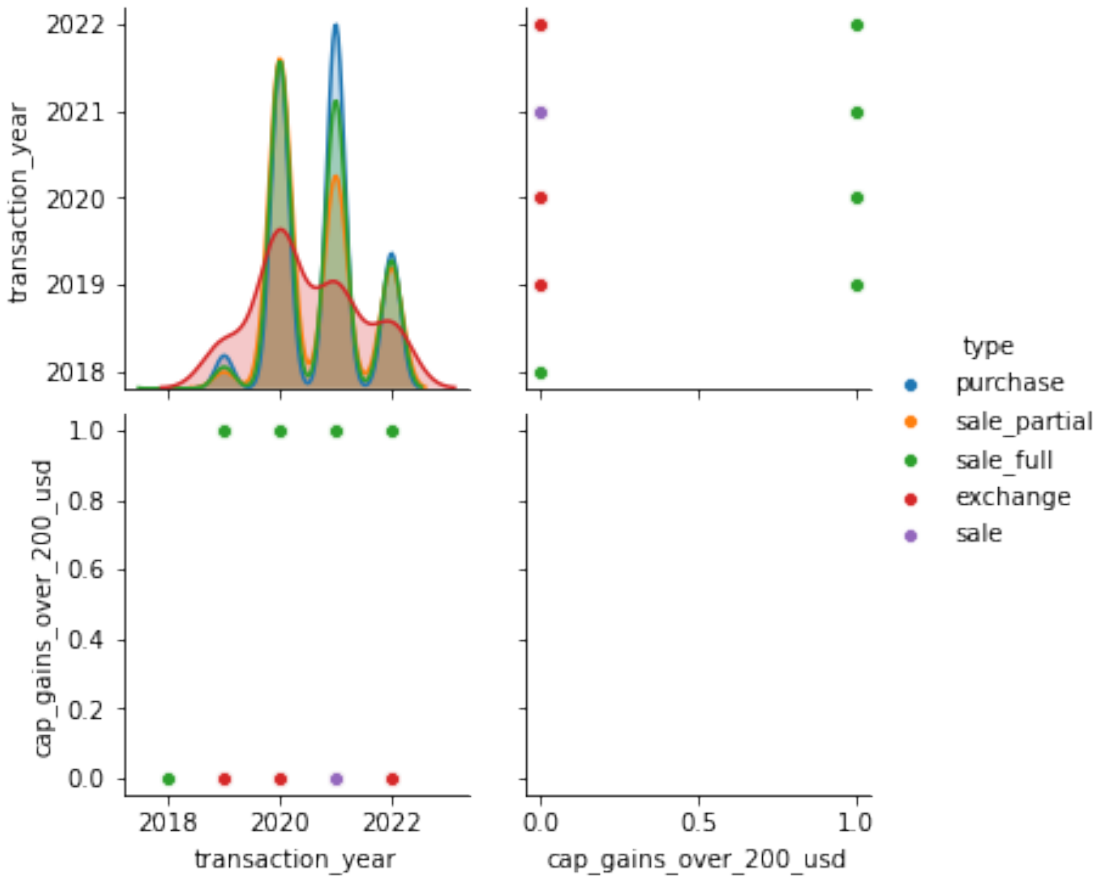
```
CA03                          0.0
…                             …
WA01                          1.0
WA08                          0.0
WI08                          3.0
WV01                          8.0
WV03                          6.0

[90 rows x 1 columns]
```

The above code counts the number of stocks which have more than 200 dollars capital gains in each of the districts present in the cleaned dataframe.

[159]:
```
sns.
↪pairplot(stocks[['transaction_year','type','cap_gains_over_200_usd']],hue='type')
```

```
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:283:
UserWarning: Data must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:369:
UserWarning: Default bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:369:
UserWarning: Default bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:369:
UserWarning: Default bandwidth for data is 0; skipping density estimation.
  warnings.warn(msg, UserWarning)
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:283:
UserWarning: Data must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
/opt/conda/lib/python3.8/site-packages/seaborn/distributions.py:283:
UserWarning: Data must have variance to compute a kernel density estimate.
  warnings.warn(msg, UserWarning)
```

[159]:
```
<seaborn.axisgrid.PairGrid at 0x7f195f223670>
```



[162]:
```
num_cap_true=np.count_nonzero(stocks['cap_gains_over_200_usd'])
num_cap_false=stocks.shape[0]-num_cap_true
```

The above variables are such num_cap_true counts the overall number of stocks with more than 200 dollars in capital gains in the cleaned dataframe, and num_cap_false counts the overall number of stocks with less than 200 dollars in capital gains in the cleaned dataframe.

3.0.2 Assessment of Missingness

[252]:
```
#column chosen is owner in stocks_df
stocks_miss=stocks_df.copy()
stocks_miss['owner_missing']=stocks_miss['owner'].isna()
n_repetitions = 500
tvds = []
for _ in range(n_repetitions):
    # Shuffling cap_gains_over_200_usd and assigning back to the DataFrame
    stocks_miss['cap_gains_over_200_usd'] = np.random.
↪permutation(stocks_miss['cap_gains_over_200_usd'])
```

```
    # Computing and storing TVD
    pivoted = (
        stocks_miss
        .pivot_table(index='owner_missing', columns='cap_gains_over_200_usd',␣
↪aggfunc='size')
        .apply(lambda x: x / x.sum(), axis=1)
    )

    tvd = pivoted.diff().iloc[-1].abs().sum() / 2
    tvds.append(tvd)
```

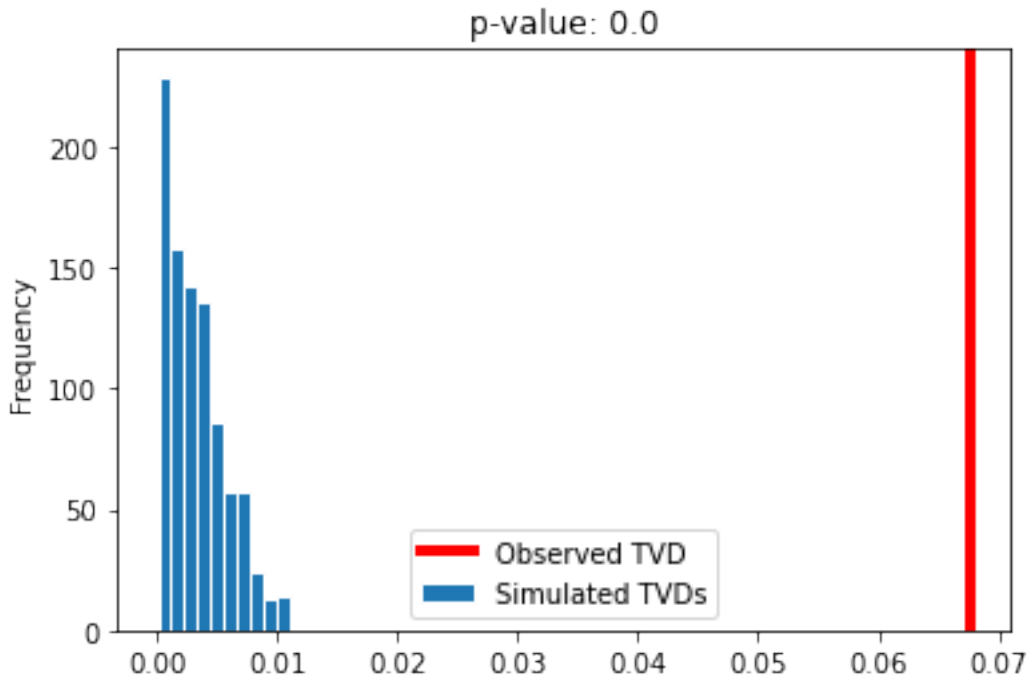
[253]:
```
cap_dist = (
    stocks_df
    .assign(owner_missing=stocks_df['owner'].isna())
    .pivot_table(index='cap_gains_over_200_usd', columns='owner_missing',␣
↪aggfunc='size')
)

cap_dist = cap_dist / cap_dist.sum()
cap_dist
```

[253]:
```
owner_missing              False     True
cap_gains_over_200_usd
False                   0.964518  0.896883
True                    0.035482  0.103117
```

[254]:
```
obs_tvd = cap_dist.diff(axis=1).iloc[:, -1].abs().sum() / 2
obs_tvd
```

[254]:
```
0.06763517575585296
```

[255]:
```
pval = np.mean(tvds >= obs_tvd)
pd.Series(tvds).plot(kind='hist', density=True, ec='w', bins=10,␣
↪title=f'p-value: {pval}', label='Simulated TVDs')
plt.axvline(x=obs_tvd, color='red', linewidth=4, label='Observed TVD')
plt.legend();
```



We reject the null hypothesis which stated that the distribution of district when owner was missing was the same as the distribution of district when owner was not missing. Thus, missingness in owner column is dependent on cap_gains_over_200_usd.

[258]:
```
#column chosen is owner in stocks_df
stocks_miss=stocks_df.copy()
stocks_miss['owner_missing']=stocks_miss['owner'].isna()
n_repetitions = 500
tvds = []
for _ in range(n_repetitions):
    # Shuffling cap_gains_over_200_usd and assigning back to the DataFrame
    stocks_miss['type'] = np.random.permutation(stocks_miss['type'])
    # Computing and storing TVD
    pivoted = (
        stocks_miss
        .pivot_table(index='owner_missing', columns='type', aggfunc='size')
        .apply(lambda x: x / x.sum(), axis=1)
    )

    tvd = pivoted.diff().iloc[-1].abs().sum() / 2
    tvds.append(tvd)
```

[259]:
```
typ_dist = (
    stocks_df
    .assign(owner_missing=stocks_df['owner'].isna())
```

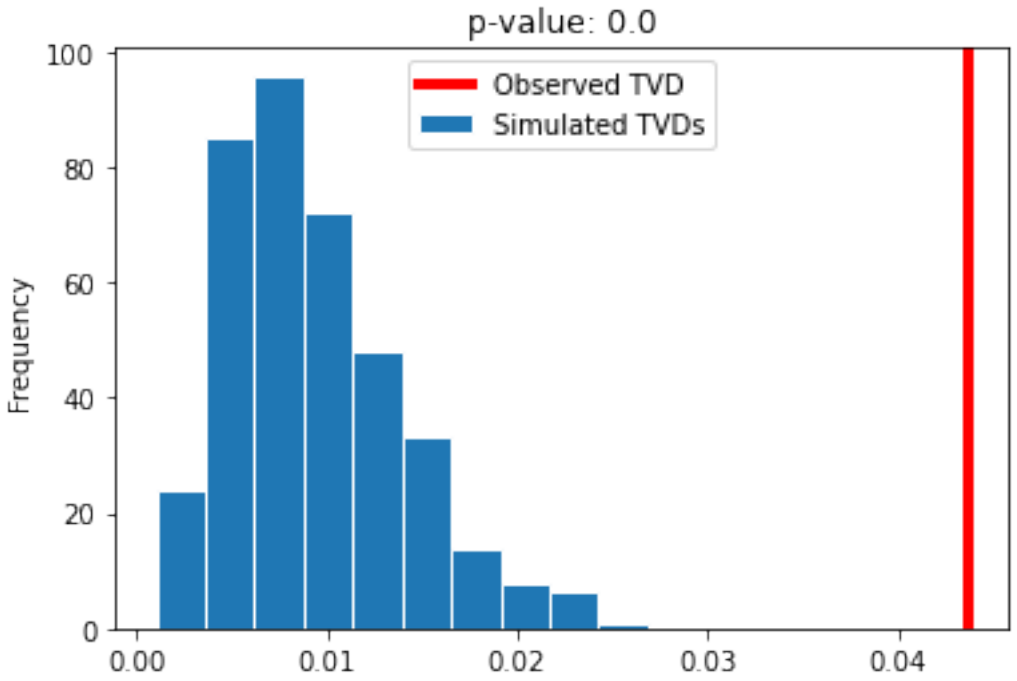
```
    .pivot_table(index='type', columns='owner_missing', aggfunc='size')
)

typ_dist = typ_dist / typ_dist.sum()
typ_dist
```

[259]:
```
owner_missing     False     True
type
exchange       0.007034  0.011771
purchase       0.537499  0.506797
sale           0.000103       NaN
sale_full      0.300197  0.339191
sale_partial   0.155167  0.142241
```

[260]:
```
obs_tvd = typ_dist.diff(axis=1).iloc[:, -1].abs().sum() / 2
obs_tvd
```

[260]:
```
0.043679030961797094
```

[261]:
```
pval = np.mean(tvds >= obs_tvd)
pd.Series(tvds).plot(kind='hist', density=True, ec='w', bins=10,␣
 ↪title=f'p-value: {pval}', label='Simulated TVDs')
plt.axvline(x=obs_tvd, color='red', linewidth=4, label='Observed TVD')
plt.legend();
```



We reject the null hypothesis which stated that the distribution of district when owner was missing was the same as the distribution of district when owner was not missing. Thus, missingness in owner column is dependent on type.

[263]:
```
#column chosen is owner in stocks_df
stocks_miss=stocks_df.copy()
stocks_miss['owner_missing']=stocks_miss['owner'].isna()
n_repetitions = 500
tvds = []
for _ in range(n_repetitions):
    # Shuffling cap_gains_over_200_usd and assigning back to the DataFrame
    stocks_miss['district'] = np.random.permutation(stocks_miss['district'])
    # Computing and storing TVD
    pivoted = (
        stocks_miss
        .pivot_table(index='owner_missing', columns='district', aggfunc='size')
        .apply(lambda x: x / x.sum(), axis=1)
    )

    tvd = pivoted.diff().iloc[-1].abs().sum() / 2
    tvds.append(tvd)
```

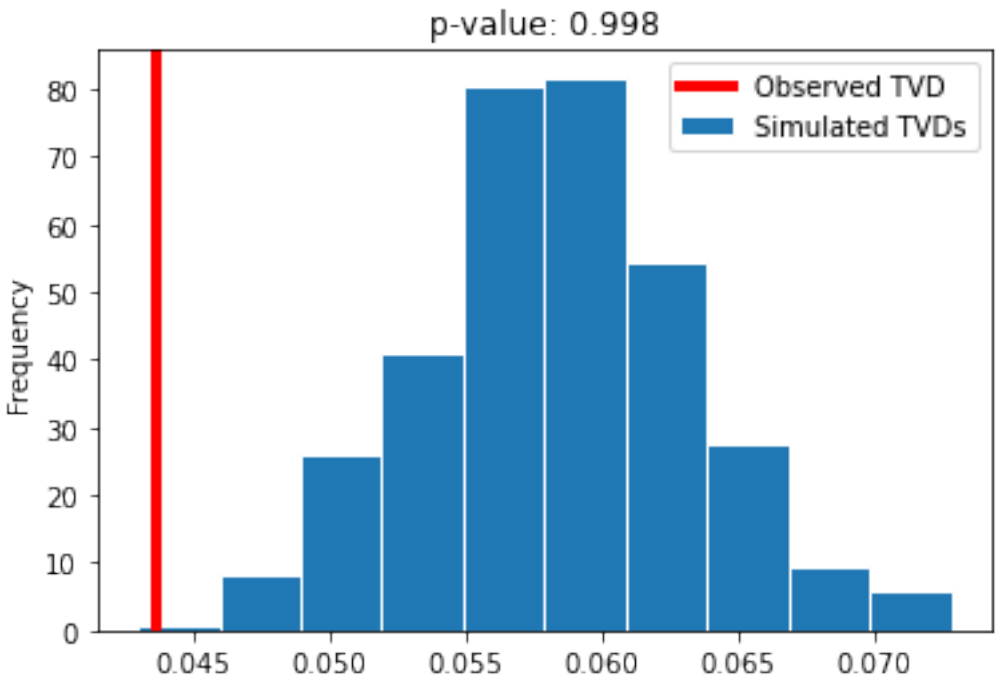
[264]:
```
dis_dist = (
    stocks_df
    .assign(owner_missing=stocks_df['owner'].isna())
    .pivot_table(index='district', columns='owner_missing', aggfunc='size')
)

dis_dist = dis_dist / dis_dist.sum()
dis_dist
```

[264]:
```
owner_missing     False     True
district
AL02                NaN  0.000995
AL04                NaN  0.000166
AL05           0.004552  0.000829
AR02           0.000724  0.000332
AZ01           0.000207       NaN
…                   …        …
WA04                NaN  0.000166
WA08           0.003931       NaN
WI08           0.003827  0.000166
WV01           0.007655  0.014423
WV03           0.006517  0.000497

[166 rows x 2 columns]
```

[265]:
```
obs_tvd = typ_dist.diff(axis=1).iloc[:, -1].abs().sum() / 2
obs_tvd
```

[265]:
```
0.043679030961797094
```

[266]:
```
pval = np.mean(tvds >= obs_tvd)
pd.Series(tvds).plot(kind='hist', density=True, ec='w', bins=10,␣
 ↪title=f'p-value: {pval}', label='Simulated TVDs')
plt.axvline(x=obs_tvd, color='red', linewidth=4, label='Observed TVD')
plt.legend();
```



We fail to reject the null hypothesis which stated that the distribution of district when owner was missing was the same as the distribution of district when owner was not missing. Thus, missingness in owner column is not dependent on district.

3.0.3 Hypothesis Test / Permutation Test

[195]:
```
pre_obs=np.count_nonzero(pre_pand['cap_gains_over_200_usd'])/pre_pand.shape[0]␣
 ↪#proportion of stocks with capital gains more than 200 dollars during the␣
 ↪pre-pandemic period
in_obs=np.count_nonzero(dur_pand['cap_gains_over_200_usd'])/dur_pand.shape[0]␣
 ↪#proportion of stocks with capital gains more than 200 dollars during the␣
 ↪pandemic period
```

```
post_obs=np.count_nonzero(post_pand['cap_gains_over_200_usd'])/post_pand.
 ↪shape[0] #proportion of stocks with capital gains more than 200 dollars␣
 ↪during the post-pandemic period
sample_dur=dur_pand.shape[0] #number of stocks in the dataframe for the␣
 ↪pandemic period
sample_post=post_pand.shape[0] #number of stocks in the dataframe for the␣
 ↪post-pandemic period
```

[196]:
```
from statsmodels.stats.proportion import proportions_ztest
```

[197]:
```
proportions_ztest(count=np.count_nonzero(dur_pand['cap_gains_over_200_usd']),␣
 ↪nobs=sample_dur, value=0.0635)
#one proportion z-test in python to check whether pandemic period had more␣
 ↪stocks by proportion with capital gains over 200 dollars than normal times
```

[197]:
```
(-0.37442497576094425, 0.708088164127187)
```

[198]:
```
proportions_ztest(count=np.count_nonzero(post_pand['cap_gains_over_200_usd']),␣
 ↪nobs=sample_post, value=0.0635)
#one proportion z-test in python to check whether post-pandemic period had␣
 ↪fewer stocks by proportion with capital gains over 200 dollars than normal␣
 ↪times
```

[198]:
```
(-57.26073028881826, 0.0)
```