An Investigation into the Stock Portfolios of the U.S. Senate

Introduction to Insider Trading

Insider Trading is the illegal practice of trading on the stock exchange to one's own advantage through having access to confidential information. On paper, this definition seems simple enough. Insider Trading laws prohibit executives, lawmakers, and others with advantages not shared by the American public from trading based on this superior knowledge. Yet in practice, defining insider trading is a much more complex task.

In 1909, the Supreme Court passed the first insider trading law after the Strong v. Repide case. Strong, the plaintiff, owned stock in a company called the Philippine Sugar Estates Development Company, a Caribbean real estate corporation. Repide, the defendant, was the administrator general at the company — he owned about 75% of the company’s stock. When the Philippine Government showed interest in purchasing the land, Repide recognized the potential growth in the company’s stock price. Bearing nonpublic information, Repide used a third party to communicate with Strong, purchasing his stock before the forthcoming land sale. After the Philippine government purchased the land, the company’s stock price increased tenfold. The Supreme Court ruled that Repide’s choice to use his nonpublic information for his own profit was wrong, resulting in a law that a company’s directors must disclose information before making trades or choose to not make trades at all.

In 1934, Congress passed the Securities Exchange Act, which better defined stock fraud. In 1942, the Securities and Exchange Commission adopted Rule 10b-5 of the Securities Exchange Act’s tenth section, declaring that such insider trading laws were applicable to both sales and purchases. A New York Times article from May of 1942 reads, “Any Person Involved in Shady Transaction Now Liable… [the law is] prohibiting fraud by any person in connection with the purchase of securities…”

In 1964, executives at the Texas Gulf Sulphur Company were aware of a newly-discovered copper ore mine in Ontario. Without sharing their information with the American public, they traded the stock. They were sued by the Securities and Exchange Commission for trading on insider information and lost the lawsuit after the court ruled that they had access to information not available to the American public. The case set precedent for similar cases to come.

During the 1980s, insider trading was rampant. In 1983, a financial analyst named Raymod Dirks discovered fraud at a financial conglomerate called Equity Funding. The company fudged their revenue numbers and insurance policies for years — when Dirks received the tip from an ex-employee, he advised his clients to sell their stock in the company. The Supreme Court ruled in favor of Dirks, arguing quite controversially that the ex-employee had not violated any legal duties to shareholders in sharing his information and therefore Dirks had no obligation to disclose the tip to the American public.

In the investment banking industry, the *New York Times* reports that almost every corporate takeover resulted in some degree of insider trading. An executive at one of the most significant American investment banks said that “It was like free sex” and while “You definitely saw the abuses growing… you also saw the absence of people getting caught.”

Insider Trading laws largely aimed to prevent people — company executives, officials, employees — with informational advantages over the American public to profit unjustly. But what about the people who regulate these companies, often determining legislation that influences their operations? Surprisingly enough, it was not until 2012, under the Obama administration, that members of Congress were prohibited from insider trading. The STOCK Act (Stop Trading on Congressional Knowledge Act) was passed with overwhelming support. In the Senate, it passed 96-3; in Congress, it passed 417-2.

Insider Trading in the U.S. Senate

The STOCK Act’s goal was simple: to foster heightened financial transparency among legislators. Insider trading laws extended to Senators, Congressman, and Federal employees, including the president, vice-president, and other roles of the executive branch. Under the STOCK Act, federal employees are required to disclose their material gains within 45 days. Such financial disclosures are accessible to the American public through the online Electronic Filing Depository — they were crucial for the research for this project. Before the passing of the STOCK Act, the database existed, but it was hard to navigate. One could only view the database by visiting the basement of the Cannon House Office Building, the office of public records. If you can make the trip, you’re not there yet. For each financial disclosure report PDF, one was required to pay 10 cents. Now, every Financial Disclosure is publicly available online, free of charge. Federal employees are also unable to participate in initial public offerings (IPOs).

In 2013, the STOCK Act was amended to loosen its requirements for legislators. In mid-April, Barack Obama signed a bill to reverse many significant aspects of the STOCK Act using a fast-track method called unanimous consent. As Tamara Keith of NPR put it, “the emailed announcement was one sentence long… Many members had already left for the weekend or were on their way out. The whole process took only 30 seconds. There was no debate.” According to members of the executive branch who supported the law, the bill was aimed to curtail risks of identity theft and address concerns for federal employees who operate businesses or work in foreign countries. According to the Congressional Bill Summary, the bill limited the financial disclosure requirements “only to members of Congress, congressional candidates, the President, the Vice President, and executive branch officers at levels I and II of the Executive Schedule who require nomination by the President and confirmation by the Senate." It also extended the deadline for such financial disclosures for certain, high-up executive branch members.

The STOCK Act was a component of Barack Obama's larger vision to create "an economy where everyone gets a fair shot, everyone does their fair share, and everyone plays by the same set of rules, including those who have been elected to serve the American people." Federal employees are now required to disclose the terms of their personal mortgages, and report their material gains. The bill was an amendment to the Ethics in Government Act, passed in 1978 in the aftermath of the Watergate scandal. If federal employees violate the STOCK Act, they must forfeit their federal pension. However, such ramifications have not kept senators and congressmen alike from using their informational advantages for personal financial gain.

In 2020, there was a Congressional Insider Trading Scandal. Four senators were accused of unloading substantial positions in the stock market after a private briefing that revealed the severity of COVID-19 and the potential of its threat to financial markets. Periodic Transaction Reports show that Kelly Loeffler (R-GA, former), Dianne Feinstein (D-CA), Richard Burr (R-NC), and Jim Inhofe (R-OK) all unloaded substantial positions in the stock market prior to the broader market crash. Allegedly, these senators had material, nonpublic information from the hearings that indicated the severity of the impact of COVID-19 on financial markets, and protected their portfolios accordingly.

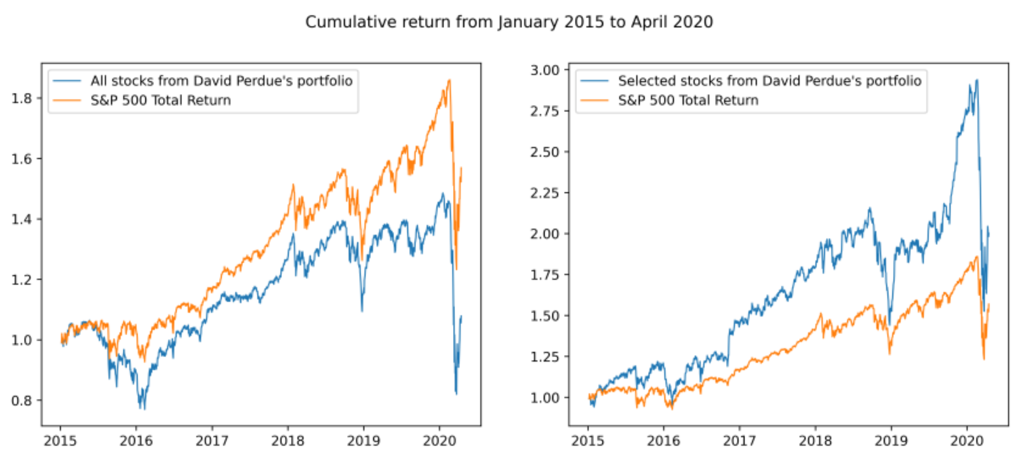
Loeffler, a former executive at Intercontinental Exchange, is married to Jeffrey Sprecher, the New York Stock Exchange's chairman. Loeffler sold between roughly $1.275 million and $3.1 million in stock. Later financial disclosures show that she sold $18.7 million worth of stock in Intercontinental Exchange. After dumping her position, the stock slid sixteen percent. Reports show that Loeffler's advisors went as far as purchasing stocks that would benefit from the COVID-19 pandemic. Loeffler and her husband Jeffrey Sprecher purchased $206,777 worth of stock in Dupont, a chemical company that makes protective gear to combat transmission of viruses. On January 24, 2020, she purchased stock in Citrix, a telecommuting company that provides platforms for online workspaces. She defended herself in a statement from her office, which contended that "Sen. Loeffler does not make investment decisions for her portfolio… Investment decisions are made by multiple third-party advisers without her or her husband's knowledge or involvement." The Senate Ethics Committee eventually dropped its investigations into Kelly Loeffler, yet some political scientists argue that her involvement in the financial scandal was a major setback in her fight to hold onto her position as Georgia senator.

Dianne Feinstein, a Democratic senator from California, was also accused of using her informational advantages regarding COVID-19 for her own financial benefit. Allegedly, Feinstein sold between $1.5 million and $5 million in stock. A spokesperson for Feinstein contested that "All of Senator Feinstein's assets are in a blund trust" and "She has no involvement in her husband's financial decisions." The Securities and Exchange Commission has since dropped their investigation into Feinstein's stock trades.

Investigations ultimately concluded into the four senators— but insider trading issues did not stop there. In December of 2020, David Perdue, the former democratic senator who served Georgia alongside Kelly Loeffler, made questionable stock trades in cybersecurity and banking companies. As an active member of the Cybersecurity Subcommittee, Perdue bought and sold stocks of a cybersecurity company known as FireEye. Similarly, Perdue bought stocks in banks including JPMorgan Chase, Bank of America, and Regions Financial, all the while passing deregulation legislation. While we cannot be certain whether these bills explicitly benefited Perdue's positions, some argue that senators trading the very companies they regulate is fundamentally corrupt. Perdue takes on a distinct responsibility as a public officer, and even having the power to use his informational advantages for his own financial benefit, they say, contradicts his duty to honorably serve the American public. Journalists at

Fortune

broke down David Perdue's portfolio using transactional data. They found that his stock positions outperformed the stock market immensely.



David Perdue's most suspicious trade, however, came earlier in 2020. Periodic transaction reports indicate that Perdue sold more than $1 million in stock in a financial company known as Cardlytics— where he was once an active member of the board. Less than fifty days later, the stock price declined dramatically as the company's founder announced that he would be stepping down as the company's CEO. The firm also released reports that indicated an unexpected decline in sales. But investigators found more— just two days before the trade, Perdue received a personal email from Cardlytics' CEO, ambiguously referencing the "upcoming changes." These trades ultimately complicated Perdue's pursuit of a second term as a Georgia senator.

Other Studies

Other studies have investigated the stock trades and returns of the U.S. Senate. One such study,

Abnormal Returns from the Common Stock Investments of the U.S. Senate

, compiled a trade-weighted Senate portfolio. , Alan J. Ziobrowski, Ping Cheng, James W. Boyd, and Brigette J. Ziobrowski, researchers from the School of Business at the University of Washington, found that the annualized return was a staggering 34.1%, demonstrating that “the Senators invested more money in the stocks that ultimately performed the best.” (667). Moreover, their investigations revealed that a portfolio that emulates the purchases of U.S. senators outperforms the market by 85 basis points per month, while one that emulates their sales loses to the market by 12 basis points. As recommended by Mitchell and Stafford (2000), the researcher used the Fama-French three-factor model and CAPM to measure abnormal performance. The researchers ran their analysis on 6,052 transactions (although many of these transactions were eliminated by screens which ignored REITs, foreign stocks, IPOs, mutual funds, and all preferred stock. In the conclusion of their study, the researchers wrote, “Political power confers many benefits. Among those benefits are privileged access to information, the power to influence legislation, and the power to influence the application of regulatory jurisdiction by administrative agencies.” (676).

FIGURE 1

Daily Cumulative Abnormal Returns for Common Stocks Bought and Sold by U.S. Senators

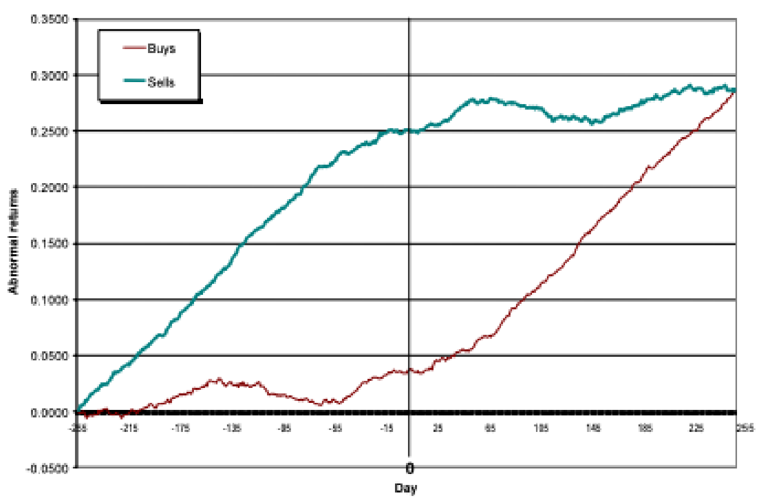


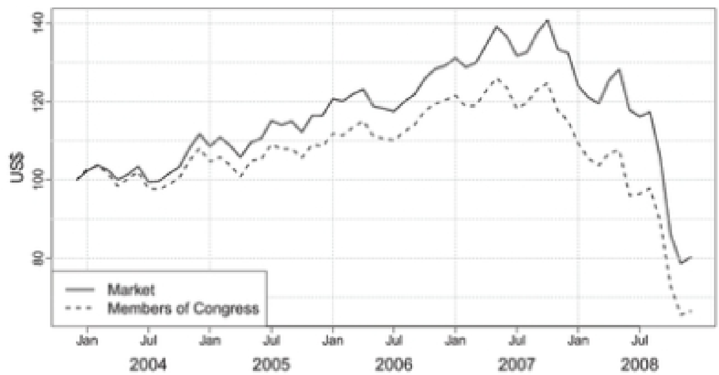
Figure 1 depicts the cumulative abnormal returns (CARs) of the buy and sell transactions of U.S. Senators during the period 255 days prior to and after the event date (day 0 on the horizontal axis). To calculate the CAR, we use the expression, $CAR_t = \sum_{T=-255}^{t} \overline{AR}_T$, where $\overline{AR}$ is the abnormal daily return on trading day t.

Another study, *Political Sentiment and Predictable Returns*, led by Jawad M. Addoum of Cornell University and Alok Kumar of the University of Miami, took a broader look at the relationship between political climate and the stock market. Though their study is not as directly related to my own, their findings were quite fascinating. Addoum and Kumar found that, as the party in power shifts, a trading strategy based on the “predictable patterns in industry returns” produced by the “systematic changes in the industry-level composition of investor portfolios” returns an annualized 6% (1). In other words, it is possible to produce a decent return by observing changes in the United States political sentiment. As public officers, senators have an acute understanding of the political climate— and thus are capable of using their informational advantages to make more informed stock market decisions. Of course, such choices are not illegal or classified in any form as insider trading.

In 2013, Andrew C. Eggers and Jens Hainmueller at the London School of Economics and Massachusetts Institute of Technology, respectively, led an investigation into Congressional Stock Portfolios. Their 2013 study, *Capital Losses: The Mediocre Performance of Congressional Stock Portfolios*, suggests that senatorial returns have decreased since the 1993-1998 survey by Ziobrowski, et. al, from which they adapted their calendar-time approach. As seen in the figure, the researchers found that despite senators’ informational advantages, they trailed the market: “$100 invested in the market index in January of 2004 would be worth about $80 by the end of 2008, whereas invested in the average congressional portfolio it would be worth only around $69 (544). In one of the study’s subsections, “Congressional Investing: Opportunities and Constraints,” the researchers addressed the unexpected nature of their findings, mentioning a 2008 study,*The Small World of Investing: Board Connections and Mutual Fund Returns.*

CAPITOL LOSSES: THE MEDIOCRE PERFORMANCE OF CONGRESSIONAL STOCK PORTFOLIOS 545

Figure 2 **Cumulative Raw Average Return of Congressional Stock Portfolios, 2004–2008 Compared to Market Benchmark**



Note: Cumulative monthly return is shown for a $100 dollar position in the CRSP market index (a value-weighted index of stocks listed on the NYSE, AMEX, and NASDAQ) and the average congressional portfolio beginning in January 2004. The average congressional portfolio return is built by averaging monthly raw returns across members for each month (see text for details).

The study's researchers, Cohen, Frazzini, and Malloy, proved that mutual fund managers perform better when they invest in companies whose executives know them personally. In an attempt to explain this discrepancy, Eggers and Hainmueller suggested legal consequences as a deterrent to insider trading by senators. The transactions of the U.S. Senators may face more intense scrutiny from the S.E.C. and government organizations alike than mutual fund managers, and therefore they are less likely to make trades based on tips from their personal network of executives.

Methodology

Because of its reputation as one of the best programming languages for data analysis, I chose to use Python for this project. To be clear, the most substantial element of the SenateTrades lies in the online dashboard. While I felt that a research paper to officially document my findings was worth writing, it should be noted that my data analysis is not as in-depth as the studies that I have discussed earlier in this paper.

My investigations rely on data from SenateStockWatcher, a political organization which is dedicated to compiling senator data from the online filing depository (where senators are legally required to disclose their trades of greater than $1000). SenateStockWatcher has been featured in many mainstream news outlets, including *CNN*, *Fortune*, *Forbes*, the *New York Times*, and *NPR*.

SenateStockWatcher is an extension of a larger project, Congress Stock Watcher, which also includes the House of Representatives. Each day, the organization runs a scraper that collects the data from the government filing depository. Each transaction is saved as an object in a JSON file. In the code snippet to the right, you can see an example transaction from Thomas H. Tuberville.


```
{
    "transaction_date": "07/14/2021",
    "owner": "Joint",
    "ticker": "BABA",
    "asset_description": "Alibaba Group Holding Limited American Depositary",
    "asset_type": "Purchase",
    "type": "Sale (Full)",
    "amount": "$15,001 - $50,000",
    "comment": "--",
    "senator": "Thomas H Tuberville",
    "ptr_link": "https://efdsearch.senate.gov/search/view/ptr/9df3fe53-21d5-43a5-a632-ea804aca39a3/",
    "disclosure_date": "08/13/2021"
  }
```


As recommended by Ziobrowski et. al (2013), I run screens on the transactional data before actually processing it. Depending on a transaction’s attributes, I add a key and value to its object (“ignored”, and the reason why the transaction was ignored). I ignore transactions that do not specify the range, date, or type (the types of transactions are Purchase, Sale (Full), Sale (Partial), and Exchange). Without any of this information, it is not possible to estimate the senator’s return on their position.

Exchanges were particularly difficult to deal with, largely because the “ticker” key in these transactions includes two tickers, not one. The first ticker is the company the senator is exchanging; the second is the one the senator is receiving. To deal with exchanges, I add two new transaction objects to the data— one sale object for the stock that was exchanged, and one purchase object, for the stock that was received in the exchange.

Often senators report in each transaction that the stock has been bought or sold not for themselves but for their spouse or children. For the sake of this analysis we will assume that a senator’s portfolio also includes the trades they make for their spouse or children.

When senators report a transaction, they are not required to specify an exact dollar amount for their purchase or sale. Instead, they provide a range. Transactions under $1,001 do not need to be reported at all. The ranges are:

1. $1,001 - $15,000
2. $15,001 - $50,000
3. $50,001 - $100,000

4. $100,001 - $250,000
5. $250,001 - $500,000
6. $500,001 - $1,000,000
7. $1,000,001 - $5,000,000
8. $5,000,001 - $25,000,000
9. Over $50,000,000

These ranges present obvious challenges for my analysis. For transactions over $50,000,000, I assume that the senator purchased $50,000,000 in stock. For all other ranges, I assume the value of their transaction was the average between the two ranges. Although this is not ideal for accuracy, it is an assumption I must make in order to produce an analysis of senatorial returns.

Formatting across transactional reports was not always consistent. Sometimes senators even submit scanned PDF files instead of the digital submission. Because I do not have the time to manually enter the data from these scanned PDFs, these senators are excluded from my analysis. The same is true for senators who invest their assets in blind trusts, a financial arrangement where the owner does not know how their assets are managed in order to prevent a conflict of interest. In these cases, senators have no control or insight into how their positions are managed and therefore are not required to report any transactions. These senators too are cut out of my analysis. After these factors are considered, seventy-one senators remain in my analysis.

Ignoring any transactions without a ticker automatically eliminates asset classes that I do not want to consider in my analysis— REITs, municipal bonds, etc. As a final data screen, I check the stock data to see if a price is available at the time of purchase. For stocks not listed on the NYSE, NASDAQ, or ASE, AlphaVantage may have no historical records of their prices. For very obscure, low-cap stocks that are listed on one of these stock exchanges, the same is sometimes true. In these cases, I have no choice but to ignore the transaction altogether.

Before I calculate returns, I first collect some basic data. How many purchases does the senator have? How many sales? As I sort through their transaction data, I also keep track of unaccounted for transactions. This occurs when a senator sells a stock that we have no record of them buying. This could be because the senator purchased the stock before they came into office and therefore never reported it, or because they mistakenly or intentionally did not report the stock. As I process their data, I collect a large dictionary of unaccounted for transactions for later review.

The most important statistics, however, are returns. In order to calculate returns for each senator, I loop through “senator_data,” which is the original transactional data from SenateStockWatcher, only it has been processed to eliminate unwanted or unprocessable transactions. The “tqdm” in the code snippet below refers to TQDM, a Fast, Extensible Progress Bar for Python. I start my analysis on January 1, 2020, although the returns that this function outputs take into consideration transactions that have occurred long before January.

```
for senator in tqdm(senator_data, "Calculating returns"):
    start_date = parse_date("Jan 1 2020")
    end_date = datetime.now()
    delta = timedelta(days=1)
    returns = {}

    while start_date <= end_date:
        portfolio = portfolio_breakdown(senator, start_date)
        if portfolio["total"] == 0:
            returns[start_date.isoformat()] = portfolio["value"] / 1
        else:
            returns[start_date.isoformat()] = portfolio["value"] / portfolio["total"]
        start_date += delta
```

I first initialized a dictionary called “returns.” The keys in this dictionary will be dates. This is important— to generate graphs, I need these dates for the x-axis. If their portfolio[“total”] — how much money they have invested into the stock market over time— on a date is zero, then this indicates that they have not yet purchased a stock or acquired one through an exchange. At this date their return is the value of their portfolio (which is almost certainly 0, considering that they have no reported positions), divided by 1. Otherwise, their return is the current value of their portfolio (considering the growth of their investments) divided by the total amount that they invested.

I also want to be able to look at the total estimated value of a senator’s portfolio. For each purchase they make, I estimate how many shares of the stock they were capable of purchasing (assuming the middle of the range). This data is added to an ongoing dictionary called “positions.” To calculate a senator’s portfolio value, I retrieve the stock price for each of their positions and simply multiply this value by their estimated number of shares. Another important facet of the online dashboard behind this project is a pie chart of each senator’s stock portfolio, displaying their top five positions. The data for this feature is achieved quite simply by sorting through the positions and extracting the five positions with the largest value. For the overall senatorial average, this process is slightly more complicated.

As you can see in the code snippet, the function “portfolio_breakdown” is doing the heavy lifting here. This function sorts through senatorial transactions given data and any date as parameters, and then returns the senator’s total invested and their portfolio value. It also keeps track of unaccounted for transactions.

```
def portfolio_breakdown(senatordata, date):
    total = 0
```

```
    cash = 0
    sales = 0
    purchases = 0
    unaccounted = []
    positions = {}

    transactions = senatordata["transactions"]

    transactions = filter(lambda k: "ignored" not in k, transactions)

    for transaction in sorted(
        transactions, key=lambda k: parse_date(k["transaction_date"])
    ):
        transaction_date = parse_date(transaction["transaction_date"])

        if transaction_date > date:
            break
        ticker = transaction["ticker"]

        if transaction["type"] == "Purchase":
            total += estimate_transaction_amount(transaction["amount"])
            purchases += 1
            if ticker in positions:
                positions[ticker] += transaction["shares"]
            else:
                positions[ticker] = transaction["shares"]
        elif transaction["type"] == "Sale (Partial)":
            if ticker in positions:
                sales += 1
                positions[ticker] -= transaction["shares"]
                cash += estimate_transaction_amount(transaction["amount"])
            else:
                # Unaccounted for sale!
                unaccounted.append(transaction)
        elif transaction["type"] == "Exchange":
            if ticker in positions:
                positions[ticker] += transaction["shares"]
                total += estimate_transaction_amount(transaction["amount"])
        elif transaction["type"] == "Sale (Full)":
            sales += 1
            if ticker in positions:
                positions[ticker] = 0
                cash += estimate_transaction_amount(transaction["amount"])
            else:
                # Unaccounted for sale!
                unaccounted.append(transaction)
        else:
            raise RuntimeError("unknown transaction type: " + transaction["type"])


    value = 0

    for (ticker, amount) in positions.items():
        if amount <= 0:
            continue
        price = stock_price(ticker, date)
        if price is None:
            continue
        value += amount * price
```

```
    return {
        "positions": positions,
        "unaccounted": unaccounted,
        "total": total,
        "value": value + cash,
        "purchases": purchases,
        "sales": sales,
        "cash": cash,
        "name": senatordata["first_name"] + " " + senatordata["last_name"],
    }
```

This function introduces a new variable, "cash." Senators' returns should consider the cash in their portfolio resulting from the sales of their stocks. This variable is updated each time there is a "Sale (Full)" meaning the entire position has been liquidated. In this case, the position is completely removed from the senators' portfolio and the cash is added to their portfolio value. Without this variable, the resulting graphs would be skewed by random decreases in returns after stock sales.

Before sorting through the senator data, I initialize a variable called transactions. Transactions has all senators' transactions, excluding the ones with the "ignored" key. Because of their ticker, type, date, these transactions are not to be processed.

Also essential in this code snippet is the stock_price() helper function. It's fairly straightforward—given a date object and a ticker symbol, the function returns a stock price. As I loop through the senator's stock positions on a given date, I can call the stock_price() function on each one, yielding the estimated value of the senator's portfolio on that day. Running this process for every day between two dates gives me a senator's return over time.

```
def stock_price(ticker, date):
    stock_data = load_alphavantage_data(ticker)

    if "Time Series (Daily)" not in stock_data:
        tqdm.write(f"{ticker} is invalid, {stock_data}")
        return None

    for j in range(4):
        time_delta = timedelta(days=j)
        date_str = (date - time_delta).strftime("%Y-%m-%d")
        if date_str in stock_data["Time Series (Daily)"]:
            return float(
                stock_data["Time Series (Daily)"][date_str]["5. adjusted close"]
            )
    return None
```

One challenge I encountered while making this stock_price() function is weekends (and select holidays), when all major stock exchanges are closed. Of course, retrieving the stock price for a senator's positions during these periods is impossible. The for loop in this function solves the problem quite elegantly. If no stock price is found altogether, then it means that the stock market was open, but there was no record of the ticker on the given day. In other words, AlphaVantage does not track the provided ticker (or it hadn't started tracking it yet). In this case, we ignore the transaction. When the stock data has records of the ticker but no stock price on the provided date, it must be a weekend or holiday. When the requested date cannot be found in the requested ticker's stock data, the function periodically checks the preceding days. If there was no stock price one day before the current date, then it goes back two days before. It repeats this process for the preceding four days.

When we finally return a stock price, we use the fifth option in the AlphaVantage Time Series (Daily) endpoint, "adjusted close." This yields the stock price at the end of the trading day, accounting for stock splits. This is very important for accuracy— if a senator owned a company that divided their shares in a 4 to 1 stock split, our graphs could show the senator's returns tumbling dramatically in a single day. Using the "adjusted close" prevents this from occurring.

Senatorial returns make much more sense when compared to market returns. Identifying trends in a senator's portfolio gains and losses is much easier next to the movement of the general market. The get_index() function returns a dictionary of the returns of the S&P 500 over time.

```
def get_index():
    index_returns = {}

    spy_start_price = stock_price("SPY", parse_date("Jan 1 2020"))
    spy_data = load_alphavantage_data("SPY")
    start_date = parse_date("Jan 1 2020")
    end_date = datetime.now()
    delta = timedelta(days=1)

    while start_date <= end_date:
        if start_date in spy_data:
            index_returns[start_date] = stock_price["SPY", start_date] / spy_start_price
        else:
            for j in range(4):
                time_delta = timedelta(days=j)
                date_str = (start_date - time_delta).strftime("%Y-%m-%d")
                if date_str in spy_data["Time Series (Daily)"]:
                    index_returns[start_date.isoformat()] = (
                        float(
                            spy_data["Time Series (Daily)"][date_str][
                                "5. adjusted close"
                            ]
                        )
                        / spy_start_price
                    )
        start_date += delta
    return index_returns
```

You'll notice that the get_index() function relies on the same implementation to deal with weekends and holidays as the stock_price() function. It also uses "5. adjusted close," though it's not likely that SPY would undergo a stock split.

There are a few other important functions in my analysis. One such function is get_top_positions(), which sorts through a senator's positions and returns their largest five positions and how much they have invested in each position.

```
def get_top_positions():
    overall_positions = {}
    overall_top_positions = {}

    for senator in processed_data.values():
        for (ticker, amount) in senator["positions"].items():
            if ticker in overall_positions:
                overall_positions[ticker] += amount
            else:
                overall_positions[ticker] = amount
    for (ticker, amount) in overall_positions.items():
        if stock_price(ticker, datetime.now()) is not None:
            overall_top_positions[ticker] = amount * stock_price(ticker, datetime.now())
        else:
            continue

    top_five_stocks = {
        stock: value
        for (stock, value) in sorted(
            overall_top_positions.items(), key=operator.itemgetter(1), reverse=True
        )[:5]
    }

    for (ticker, amount) in top_five_stocks.items():
        if amount < 0:
            top_five_stocks[ticker] = 0
```

Get_top_positions() operates similarly to portfolio_breakdown(). It uses the stock_price() helper function to retrieve the value of each position as it sorts through the holdings for each senator.

The function is designed such that I can change the number of top positions from five to a higher or lower number by changing a single line of code. However, for the portfolio breakdown pie chart feature on the online dashboard, it is my opinion that more than five positions causes the chart legend to be over-crowded and less comprehensible— five positions is enough.

The final part of my analysis is where I take averages of the data to find what the senatorial average overtime. The get_average_returns() function achieves this quite simply.

```
def get_average_returns():
    average_daily_returns = {}

    start_date = parse_date("Jan 1 2020")
```

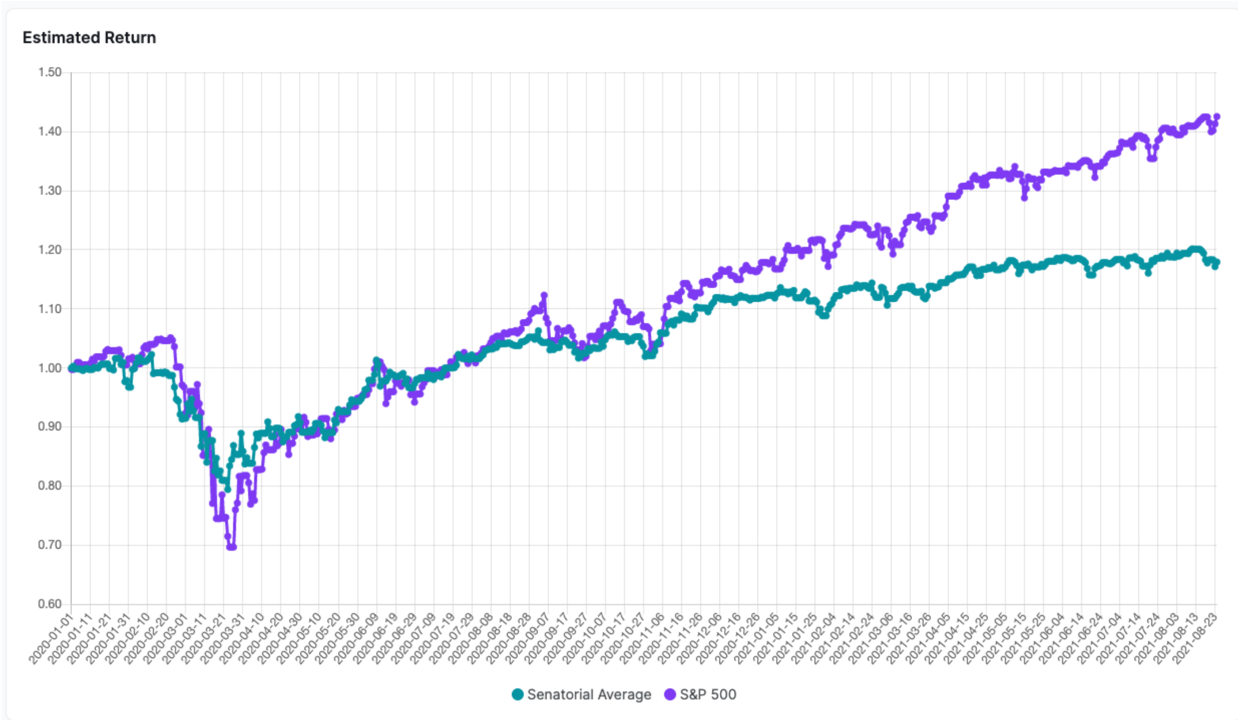
```
    end_date = datetime.now()
    delta = timedelta(days=1)

    while start_date <= end_date:
        temporary_average = []
        for senator in processed_data.values():
            if senator["returns"][start_date.isoformat()] == 0.0:
                continue
            else:
                temporary_average.append(senator["returns"][start_date.isoformat()])
        average_daily_returns[start_date.isoformat()] = sum(temporary_average) / len(
            temporary_average
        )
        start_date += delta
    return average_daily_returns
```

The function loops through each senator in the processed data and extracts their returns. If their returns are 0.0, it means they haven't reported any positions or portfolio growth at that time. For these senators, we omit their return at that time so we do not skew the average. When the function completes, the dictionary average_daily_returns contains dates as keys and the senatorial average return as values. We then project this dictionary onto a chart against the S&P 500, with the keys as the x-axis and the senatorial returns on the y-axis.

Results

Senators, on average, underperformed the market in 2020. As of August 27, 2021, their average return is 17.92%. In other words, if you put $1,000 in a portfolio that mimics the stock trades of senators, you'd have $1179 today. The same investment into the S&P 500 would be worth $1,420. While their impressive return doesn't compare the the S&P 500's record-breaking run, their portfolio maintained more of its value during March losses, where the average senatorial portfolio lost 20% of its value compared to a mere 25% loss for the general market index. In general, the senatorial average seems to mimic SPY quite closely.



The combined portfolio value of the senators tracked by my analysis is $219,987,148. Senators made 3243 combined purchases with 2615 sales, totalling to 5858 stock trades. This number excludes a fairly significant amount of non-stock related transactions for reasons provided in the Methods section.

Conclusion