ETFs and Historical Returns Copy

Hao Qi, Chen-Yun Yang, Xiaoyi Hu, Yu-Ting Yeh, and Po-Hsi Lin

Department of Statistics

Rice University

STAT 686

Group 14

Market Model Homework 1

(hq15, cy60, xh57, yy152, pl55)@rice.edu

Due Date: February 5, 2025

Contents

1 Introduction

This study analyzes the historical return characteristics of various Exchange-Traded Funds (ETFs) across different investment strategies and market segments. Specifically, we examine the returns of 11 Global Industry Classification Standard (GICS) sector ETFs, along with small-cap, large-cap, growth, value, momentum, carry trade, and volatility ETFs. Our analysis includes the calculation of the average and median returns over multiple timeframes, including daily, weekly, monthly, yearly, and 10-year rolling windows. Additionally, we compute the Compound Annual Growth Rate (CAGR) for all available decade-long periods and an arbitrary N-year period. Beyond return metrics, this study summarizes the methodologies behind these ETFs, visualizes their price series, and examines their risk-adjusted performance using Sharpe ratios. We further assess Value at Risk (VaR) and Conditional Value at Risk (CVaR) at significance levels $\alpha = 1\%$ and $\alpha = 5\%$ for the GICS sector ETFs. To enhance the robustness of our findings, we compare mean versus median returns and explore whether investment conclusions would differ based on the chosen measure. A statistical analysis of a selected ETF is also performed to assess its distributional properties. Finally, we compute the maximum drawdown for an equally weighted portfolio of the 11 GICS sector ETFs to evaluate potential downside risk.

This study aims to provide a comprehensive understanding of the risk-return characteristics of different market strategies while highlighting the importance of selecting appropriate performance metrics when evaluating historical returns.

2 Data

Our dataset is sourced from the yfinance library in Python, which retrieves daily financial and market data from Yahoo Finance. For this project, we collected the closing prices of available stocks from 1993 through the end of January 2025 to perform our computations. A detailed explanation of the methodology is provided in each section.

3 Assumptions

This assignment relies on several key assumptions. Returns are presumed to be stationary, allowing past trends to inform future performance, though financial markets often experience structural shifts. Risk measures such as Value at Risk (VaR) and Conditional Value at Risk (CVaR) rely on historical distributions, potentially underestimating extreme market events. The Sharpe ratio assumes standard deviation is a valid risk measure, despite financial returns being non-normally distributed. Additionally, the equally weighted portfolio used in maximum drawdown analysis simplifies risk assessment but does not reflect real-world portfolio optimization. While these assumptions support the studys analytical framework, they introduce limitations in practical applications.

4 Methodology and Results

4.1 ETFs Performance Analysis Summary Table

For this section, we explore the average and median return at different time bases: daily, weekly, monthly, yearly, and 10-year measurements. The equation is as follows:

$$\text{Average Return} = \frac{1}{n}\sum_{i=1}^{n} r_i$$

$$\text{Median Return} = \begin{cases} r_{\frac{n+1}{2}}, & \text{if } n \text{ is odd} \\ \frac{r_{\frac{n}{2}} + r_{\frac{n}{2}+1}}{2}, & \text{if } n \text{ is even} \end{cases}$$

Where:

- r_i = The value at the end of the period.
- n = The number of years,or time periods, between the two values.

The result is as follows

Table 1: Average and Median Returns for Selected ETFs

Window	**Ticker**	**Average Return**	**Median Return**
Daily	DBV	0.00%	0.00%
Daily	IWM	0.04%	0.10%
Daily	MTUM	0.06%	0.11%
Daily	SPY	0.05%	0.07%
Daily	VTV	0.04%	0.07%
⋮	⋮	⋮	⋮
10_Year	XLP	140.14%	156.03%
10_Year	XLRE	–	–
10_Year	XLU	128.31%	132.09%
10_Year	XLV	174.61%	176.79%
10_Year	XLY	210.50%	174.95%

Table 1 summarizes the differences in the short-term and long-term gains of ETFs. The full table is included in the appendix. In the short term, represented by daily, weekly, and monthly returns, most ETFs exhibit modest average returns, with some variance in their median returns. ETFs like MTUM, show relatively strong short-term performance with consistently higher average and median returns. Meanwhile, DBV exhibits near-zero return in the short term, reflecting its potential volatility or weaker alignment with short-term investment horizons.

In contrast, long-term performance, represented by yearly and 10-year returns, highlights these ETFs' stability and growth potential. Notably, ETFs like DBV show a significant improvement in long-term returns, resulting an increment of both average and median returns, showing long-term investments mitigate short-term instability. SPY and VTV maintain their steady performance over the long term, reflecting their alignment with market indices and value investing principles, respectively. MTUM, although strong in the short term, shows only a marginal decline in long-term returns.

One thing that's worth keeping track of is the trade-off between stability and growth. ETFs such as DBV or MTUM tend to display greater short-term variability but can yield significant benefits over the long term. Broad-market ETFs like SPY and VTV offer consistent performance over time, making them ideal for investors seeking balanced returns with reduced volatility across short and long horizons. This comparison underscores the importance of aligning ETF selection with investment goals, as some ETFs thrive in short-term tactical opportunities while others excel in long-term wealth building.

4.2 Compound Annual Growth Rate (CAGR) for N years

CAGR, as metric used to calculate the average annual growth rate of an investment, business metric, or financial performance over a specified period, assuming the profits are reinvested at the end of each period. It provides a **smoothed rate of growth** that eliminates the effects of volatility and fluctuations over the analyzed period.

Formula for CAGR:

$$\text{CAGR} = \left(\frac{\text{Ending Value}}{\text{Beginning Value}}\right)^{\frac{1}{n}} - 1$$

Where

- Ending Value = The value at the end of the period.
- Beginning Value = The value at the start of the period.
- n = The number of years (or time periods) between the two values.

In this problem, we set N to be 1,3,5,10 respectively and use rolling windows to get different periods of CAGR. For simplicity, we assume there is 252 trading days in a year.

Due to the large size of the CAGR table, we are unable to include in this section. In Canvas, we will include our results, which incorporate 1 year, 3 years, 5 years, and 10 years.

4.3 Introduction of ETFs

The GIC sectors represent a range of industries and investment strategies, primarily using the S&P 500. The GIC ETFs include 11 distinct sectors: Financials, Technology, Energy, Consumer Discretionary, Health Care, Consumer Staples, Industrials, Utilities, Materials, Real Estate, and Communication Services. The ETFs within their respective sectors aim to effectively represent how well the industry goes regularly with the GICS, known as the Global Industry Classification Standard. Each sector includes well-known companies within their selection. For example, companies within Financial (XLF) include JPMorgan Chase, Bank of America, etc. Apple, Microsoft, Oracle for Technology. Besides the GIC sectors, the SPY, or the S&P 500 ETF, represents the largest or top 500 scale companies across all industries.

Besides using the S&P 500 index, the below ETFs represent a variety of investment strategies and market exposures. Small Cap ETF (IWM), which captures the small-scale companies' growth rate, tracks the Russell 2000 Index. Using the CRSP US Large Cap Growth Index, the Vanguard Growth ETF (VUG), and the Vanguard Value ETF (VTV), respectively, target firms with the potential to grow in revenue and firms believed to be undervalued. The Momentum ETF eye on mid and large-scale capitalization companies with strong recent performance. The Invesco DB G10 Currency Harvest Fund (DBV) tracks the DB G10 Currency Future Harvest Index, aiming to reflect the profitability of interest rates and currency fluctuations over time. Unlike the other ETFs, the Volatility ETF(VIX) seeks to track changes in market volatility rather than the price of stocks or other traditional assets using the Cboe Volatility Index (VIX).

4.4 Historical Price Movement

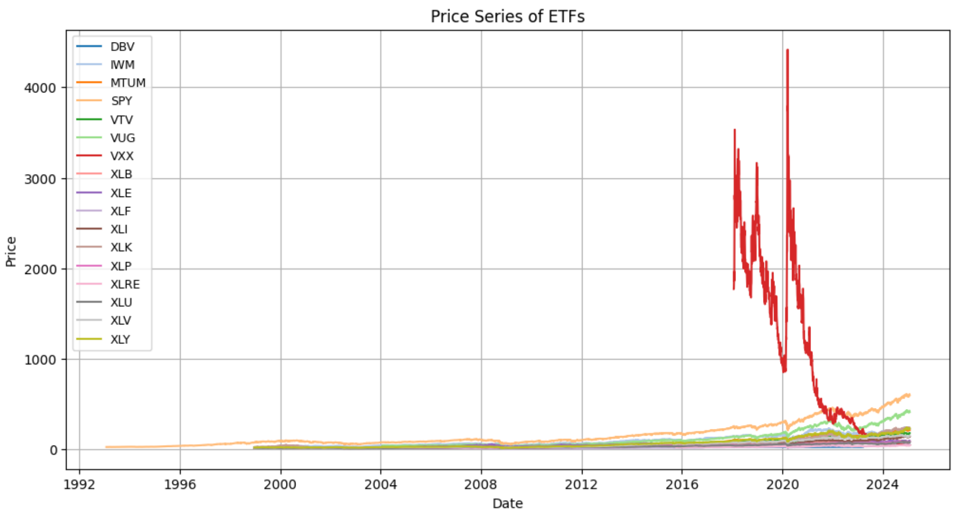

Figure 1: Price Series of Selected ETFs

Figure 1 illustrates the historical price trends of selected ETFs, covering broad market indices, sector-based ETFs, and alternative strategies like volatility and carry trade. The volatility ETF (VXX) exhibits extreme fluctuations, highlighting its short-term trading nature, while broad equity ETFs such as SPY and IWM show steady long-term growth. Sector ETFs display varied trends, reflecting economic cycles and market rotations. The carry trade ETF (DBV) remains relatively stable, as its performance is driven by currency interest rate differentials. This analysis underscores the diverse risk-return profiles of different ETF categories, emphasizing the importance of strategy selection in portfolio management.

4.5 Daily Return and Daily Volatility (Mean Measure)

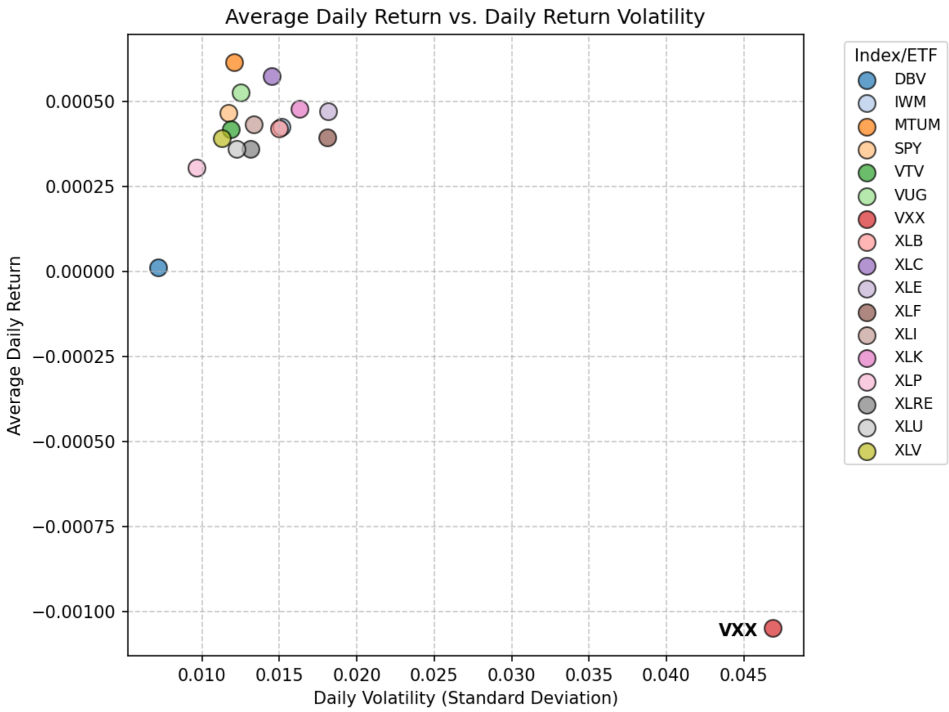

Figure 2: scatterplot of average daily return vs daily return volatility

Figure 2 illustrates the relationship between average daily returns (mean daily return) and daily return volatility (standard deviation of daily return) for selected ETFs. In general, most ETFs cluster around volatility levels below 0.02 with modest positive daily returns, ranging between 0.00025 and 0.0005. Broad market ETFs like SPY show strong daily performance with low volatility. ETFs focusing on particular industries, such as XLK in Technology and XLY Consumer Discretionary, demonstrate slightly higher returns with comparable risk. At the same time, sectors like XLU in Utilities and XLV in Health Care maintain low volatility and low returns, emphasizing their stability. Notably, VXX stands out as an outlier, exhibiting high volatility (above 0.04) and only negative mean returns.

4.6 Comparison to Median Measure: Which is Preferred?

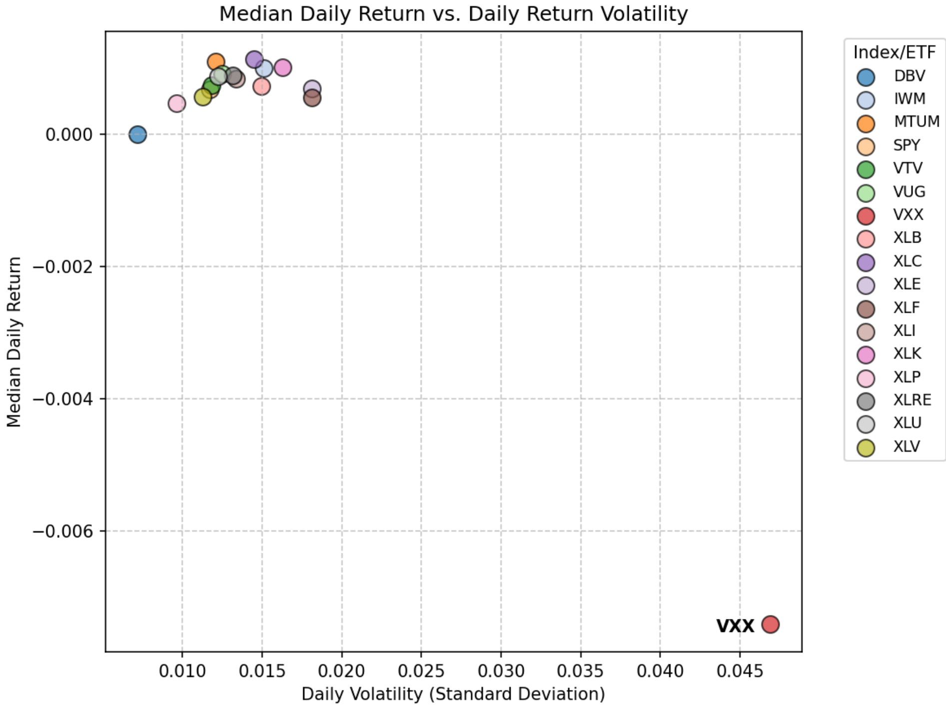

Figure 3: scatterplot of median daily return vs daily return volatility

The two graphs compare average daily returns (Figure 2) and median daily returns (Figure 3) against daily return volatility for various ETFs and indices. The analysis highlights how the choice of metricmean or mediancan influence the interpretation of profitability and risk-adjusted returns. In Figure 2, the average daily returns are generally positive from 0.025% to 0.05% for most ETFs. Only for VXX, the volatility-focused ETF, stands out with high volatility and significantly negative average returns. In contrast, Figure 3 shows median returns and loss for all ETFs are closer to zero for most ETFs, as the median reduces the influence of outliers. The choice between mean and median returns can lead to different conclusions about profitability. When using average returns, ETFs with high volatility or infrequent extreme gains appear less favorable due to the exaggerated influence of outliers. Conversely, median returns paint a more stable and realistic picture of the typical performance of an ETF. For example, stable ETFs like SPY or VUG exhibit consistent results across both metrics, while the volatile nature of VXX results in significant differences between its average and median returns.

Generally speaking, the median should be a more favorable selection, especially for most ETF buyers who aim for return in the short-run, as the median return should reflect more accurate results in their investment strategies. In the long run, the two strategies should become identical with the diminishing power of outlier return.

4.7 VaR and CVaR Table for ETFs

Value at Risk (VaR) is a measure of downside risk, the maximum potential loss that a position could suffer given a certain level of confidence *alpha*.

We use historical simulation to obtain our Var and CVaR here. We sort historical returns and find the quantile corresponding to the confidence level 1% and 5% respectively.

The risk values and the conditional risk values for alpha = 1% and 5% for the GIC sector ETFs are listed in Table 2.

Table 2: VaR and CVaR Metrics for Selected ETFs

Ticker	**VaR 1%**	**CVaR 1%**	**VaR 5%**	**CVaR 5%**
DBV	-2.19%	-3.27%	-1.01%	-1.80%
IWM	-3.84%	-5.64%	-2.32%	-3.44%
MTUM	-3.50%	-4.79%	-1.82%	-2.94%
SPY	-3.22%	-4.66%	-1.81%	-2.79%
VTV	-3.33%	-5.22%	-1.71%	-2.90%
VUG	-3.69%	-5.00%	-1.92%	-3.04%
VXX	-8.80%	-11.50%	-5.91%	-7.96%
XLB	-4.06%	-5.53%	-2.31%	-3.44%
XLC	-3.95%	-5.42%	-2.20%	-3.46%
XLE	-4.68%	-7.02%	-2.75%	-4.16%
XLF	-5.06%	-7.62%	-2.50%	-4.20%
XLI	-3.72%	-5.16%	-2.05%	-3.15%
XLK	-4.59%	-5.69%	-2.59%	-3.80%
XLP	-2.70%	-3.68%	-1.45%	-2.24%
XLRE	-3.69%	-5.51%	-1.85%	-3.06%
XLU	-3.33%	-4.67%	-1.82%	-2.83%
XLV	-3.04%	-4.31%	-1.72%	-2.61%
XLY	-3.87%	-5.37%	-2.21%	-3.34%

4.8 Statistical Analysis on SPY Return Series

The statistical summary of SPY returns provides key insights into its historical performance. The mean daily return is 0.0004, indicating a small positive drift, while the standard deviation (1.17%) highlights typical daily price fluctuations. The median return (0.00063 or 0.063%) is slightly higher than the mean, suggesting a slightly right-tailed distribution, but the skewness (-0.061) indicates a minor leftward bias, meaning that extreme negative returns are marginally more frequent. The kurtosis (11.31) is significantly higher than that of a normal distribution (which has a kurtosis of 3), confirming the presence of heavy tails, meaning extreme market movements occur more often than expected under normality.

Table 3: Statistics of SPY Returns

Stat	Value
Count	8058
Mean	0.0004
Std	0.0117
Min	-0.1094
25%	-0.0046
50%	0.0006
75%	0.0059
Max	0.1452
Kurtosis	11.3142
Skewness	-0.0610

The histogram of SPY returns further supports this observation, as the return distribution appears highly peaked with fat tails, deviating from the normal distribution fit. This suggests that risk measures relying solely on standard deviation (such as the Sharpe ratio) may underestimate the probability of extreme losses. Given the frequent occurrence of outlier returns, risk-conscious investors should consider alternative risk measures such as Value at Risk (VaR) and Conditional Value at Risk (CVaR) for a more robust risk assessment of SPYs return distribution.

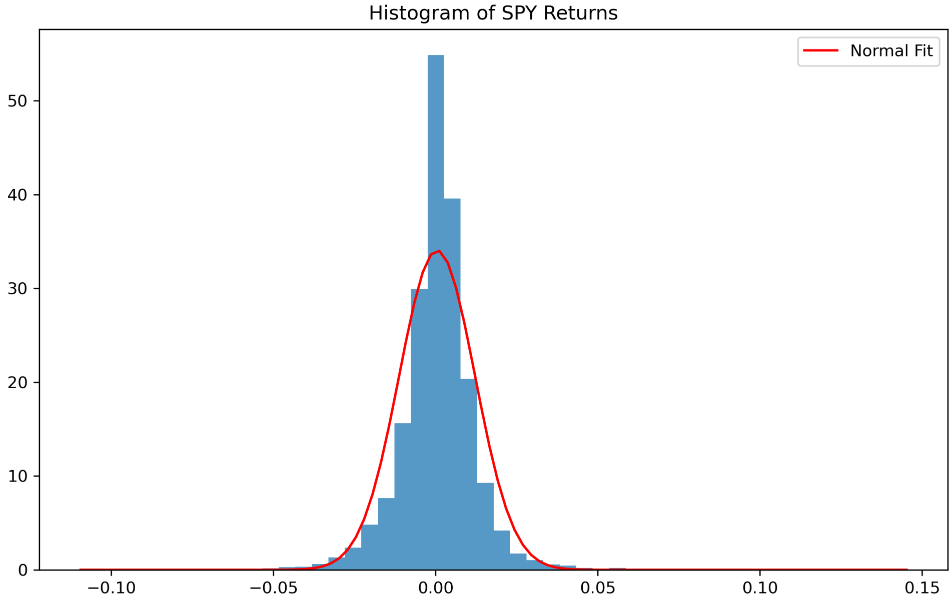

Figure 4: Histogram of SPY Returns

4.9 Sharpe Ratios for ETFs

The Sharpe ratios for GICS sector ETFs were calculated over different time periods using the US T-Bill rate as the risk-free rate. The Sharpe ratio measures risk-adjusted returns:

$$S = \frac{E[R_p] - R_f}{\sigma_p}$$

where $E[R_p]$ is the expected return of the ETF, R_f is the risk-free rate, and σ_p is the standard deviation of returns. The results show that momentum (MTUM) and growth (VUG) ETFs had the highest Sharpe ratios, especially over longer periods, indicating strong risk-adjusted performance. The volatility ETF (VXX) consistently had negative Sharpe ratios, reflecting its high decay and poor long-term returns. Defensive ETFs like Consumer Staples (XLP) and Health Care (XLV) maintained stable Sharpe ratios, suggesting lower risk and steady returns. Meanwhile, cyclical sectors such as Energy (XLE) and Financials (XLF) had more variable Sharpe ratios, fluctuating with market cycles.

Table 4: Ticker Values

Ticker	Value
DBV	-0.96%
IWM	2.28%
MTUM	4.39%
SPY	3.3%
VTV	2.84%
VUG	3.57%
VXX	-2.44%
XLB	2.28%
XLC	3.39%
XLE	2.17%
XLF	1.73%
XLI	2.63%
XLK	2.44%
XLP	2.35%
XLRE	2.09%
XLU	2.26%
XLV	2.75%
XLY	2.83%

4.10 Maximum Drawdown Risk Measure

In this problem, we calculated the Maximum Drawdown from an equally weighted portfolio comprised of the 11 GIC sector ETFs. We will explain the Maximum Drawdown and portfolio return first.

Maximum Drawdown (MDD) is a key risk metric that measures the largest observed loss from a portfolios peak to its trough before it recovers. It is mathematically defined as:

$$MDD = \min\left(\frac{C_t - R_t}{R_t}\right)$$

where: - C_t represents the cumulative return of the portfolio at time t, - R_t is the maximum cumulative return observed up to time t:

$$R_t = \max_{i \leq t} C_i$$

The Maximum Drawdown for this equally weighted portfolio of 11 GIC sector ETFs is calculated as:

$$MDD = -0.265766$$

indicating a maximum loss of approximately 26.58% from the highest portfolio value before a significant recovery.

Portfolio return is the weighted average return of the assets in the portfolio:

$$R_p = \sum_{i=1}^{n} w_i R_i$$

Where:

- R_p is the portfolio return.
- w_i is the weight of asset i (the amount invested in the asset divided by the total portfolio value).
- R_i is the return of asset i.
- n is the number of assets in the portfolio.

The accompanying figure presents the cumulative return of the portfolio over time. The portfolio experiences multiple periods of drawdown, with the most severe decline occurring during early 2020, likely due to market-wide disruptions. Despite this downturn, the portfolio demonstrates resilience and recovers to new highs. The drawdown periods highlight temporary losses, but the long-term trend suggests overall portfolio growth.

Figure 5 shows the cumulative returns of an equally weighted portfolio of 11 GIC sector ETFs over time. The portfolio follows an upward trend but experiences noticeable drawdowns, particularly a sharp decline in early 2020, where the **maximum drawdown (MDD) of -26.58%** occurs. This represents the largest drop from a peak before recovery. Despite short-term losses, the portfolio rebounds and reaches new highs, highlighting long-term growth potential and resilience against market fluctuations.

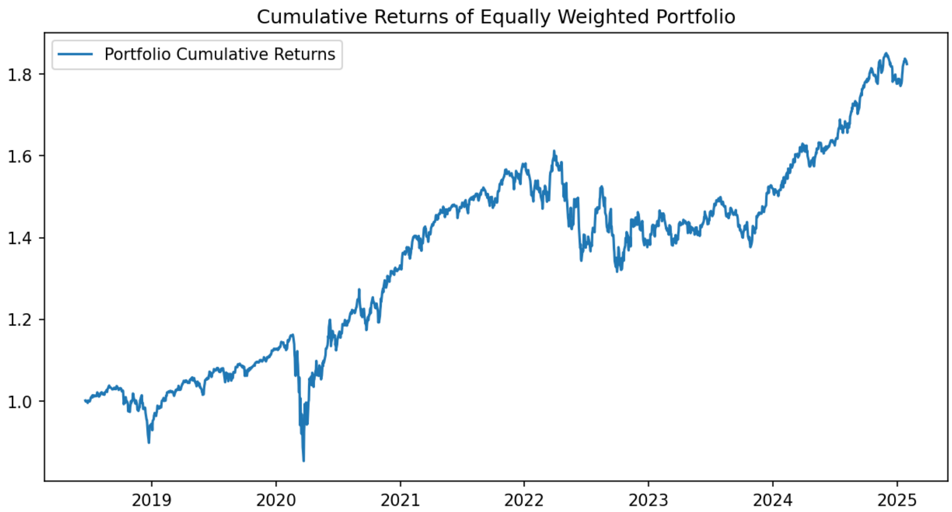

Figure 5: Histogram of SPY Returns

5 Conclusions

This assignment provides a comprehensive analysis of the risk-return characteristics of various ETFs, highlighting the differences in performance across investment strategies and market segments. Through statistical measures such as CAGR, Sharpe ratios, VaR, and CVaR, we evaluated the trade-offs between stability and growth, emphasizing the importance of selecting appropriate risk-adjusted performance metrics. The findings reveal that while broad-market ETFs like SPY offer consistent long-term growth, sector-specific and strategy-driven ETFs exhibit varied risk-return profiles. Understanding these dynamics is crucial for investors in constructing portfolios that align with their risk tolerance and investment objectives. Future research could further explore dynamic allocation strategies and the impact of macroeconomic factors on ETF performance.

References

State Street Global Advisors. The Technology Select Sector SPDR Fund. Accessed February 3, 2025. https://www.ssga.com/us/en/intermediary/etfs/the-technology-select-sector-spdr-fund-xlk

Vanguard. VUG Vanguard Growth ETF. Accessed February 3, 2025.
https://investor.vanguard.com/investment-products/etfs/profile/vug
Yfinance. All close price of indexed and ETFs data is from Yfinance, imported from Python package.

A Appendix: Code

```python
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm

# Define the tickers for the 11 GIC sectors and other indices
etf_tickers = {
    "GIC_Sector_1": "XLF",  # Example: Financials
    "GIC_Sector_2": "XLK",  # Example: Technology
    "GIC_Sector_3": "XLE",  # Example: Energy
    "GIC_Sector_4": "XLY",  # Example: Consumer Discretionary
    "GIC_Sector_5": "XLV",  # Example: Health Care
    "GIC_Sector_6": "XLP",  # Example: Consumer Staples
    "GIC_Sector_7": "XLI",  # Example: Industrials
    "GIC_Sector_8": "XLU",  # Example: Utilities
    "GIC_Sector_9": "XLB",  # Example: Materials
    "GIC_Sector_10": "XLRE", # Example: Real Estate
    "GIC_Sector_11": "XLC", # Example: Communication Services
    "Small_Cap": "IWM",      # Example: Small Cap ETF
    "Large_Cap": "SPY",      # Example: S&P 500 ETF
    "Growth": "VUG",         # Example: Growth ETF
    "Value": "VTV",          # Example: Value ETF
    "Momentum": "MTUM",      # Example: Momentum ETF
    "Carry": "DBV",          # Example: Carry ETF
    "Volatility": "VXX"      # Example: Volatility ETF
}

data = yf.download(list(etf_tickers.values()), period="max", progress=
                                    False)["Close"]
data

data.to_csv('raw_data.csv')

data = pd.read_csv('raw_data.csv')
```

```
data.iloc[:, 1:] = data.iloc[:, 1:].apply(pd.to_numeric, errors='coerce
                                          ')

rolling_windows = {
    "Daily": 1,
    "Weekly": 5,
    "Monthly": 21,
    "Yearly": 252,
    "10_Year": 2520
}

results = {}


for window_name, window_size in rolling_windows.items():
    avg_return = {}
    median_return = {}

    for ticker in data.columns[1:]:
        if len(data[ticker]) >= window_size:
            returns = data[ticker] / data[ticker].shift(window_size) -
                                          1
            avg_return[ticker] = returns.mean()
            median_return[ticker] = returns.median()
        else:
            avg_return[ticker] = np.nan
            median_return[ticker] = np.nan

    results[window_name] = {
        "Average": avg_return,
        "Median": median_return
    }

final_results = {
    "Window": [],
    "Ticker": [],
    "Average Return": [],
    "Median Return": []
}
```

```
for window_name, stats in results.items():
    for ticker in stats["Average"].keys():
        final_results["Window"].append(window_name)
        final_results["Ticker"].append(ticker)
        final_results["Average Return"].append(stats["Average"][ticker]
                                                )
        final_results["Median Return"].append(stats["Median"][ticker])

final_results_df = pd.DataFrame(final_results)
final_results_df

final_results_df.to_csv("returns_analysis_q1.csv", index=False)

def calculate_cagr(data, years):
    """
    Calculate CAGRs for specified time windows.

    Parameters:
    - data: DataFrame containing price data.
    - years: List of years (e.g., [1, 3, 5, 10]) to calculate CAGRs.

    Returns:
    - cagr_results: A dictionary of CAGRs for each year window.
    """
    trading_days_per_year = 252
    cagr_results = {year: pd.DataFrame(index=data.index) for year in
                                        years}

    for ticker in data.columns:
        for year in years:
            window_size = year * trading_days_per_year
            if len(data[ticker].dropna()) >= window_size:
                # Calculate the rolling CAGR
                cagr_results[year][ticker] = (
                    data[ticker]
                    .rolling(window=window_size)
                    .apply(lambda x: (x[-1] / x[0]) ** (1 / year) - 1,
                                                raw=False)
                )
```

```
            else:
                cagr_results[year][ticker] = pd.Series(index=data.index
                                                       , dtype="float64")

    return cagr_results


years = [1, 3, 5, 10]
cagr_results = calculate_cagr(data, years)

for year, df in cagr_results.items():
    file_name = f"{year}_year_CAGR.csv"
    df.to_csv(file_name)

## q4 Plot all ETF price series on the same graph
colors = ['#1f77b4', '#aec7e8', '#ff7f0e', '#ffbb78', '#2ca02c',  '#
                                    98df8a', '#d62728', '#ff9896', '#
                                    9467bd', '#c5b0d5',  '#8c564b', '#
                                    c49c94', '#e377c2', '#f7b6d2', '#
                                    7f7f7f',  '#c7c7c7', '#bcbd22']
plt.figure(figsize=(12, 6))
for ticker, color in zip(data.columns, colors):
    plt.plot(data.index, data[ticker], label=ticker, color=color)

plt.title("Price Series of ETFs")
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend(loc="upper left", fontsize=9)
plt.grid()
plt.show()

##q5 Scatter plot: Average Daily Return vs. Daily Return Volatility

avg_daily_return = daily_returns.mean()
median_daily_return = daily_returns.median()
daily_volatility = daily_returns.std()

scatter_df = pd.DataFrame({
    "ETF": avg_daily_return.index,
```

```
    "Average Daily Return": avg_daily_return.values,
    "Median Daily Return" : median_daily_return.values,
    "Daily Volatility": daily_volatility.values
})

colors = ['#1f77b4', '#aec7e8', '#ff7f0e', '#ffbb78', '#2ca02c',  '#
                                    98df8a', '#d62728', '#ff9896',
          '#9467bd', '#c5b0d5', '#8c564b', '#c49c94', '#e377c2', '#
                                              f7b6d2', '#7f7f7f', '#
                                              c7c7c7', '#bcbd22']

plt.figure(figsize=(8, 6), dpi=150)

for (ticker, color) in zip(scatter_df["ETF"], colors):
    x, y = scatter_df.loc[scatter_df["ETF"] == ticker, "Daily
                                        Volatility"].values[0], \
           scatter_df.loc[scatter_df["ETF"] == ticker, "Average Daily
                                              Return"].values[0]

    plt.scatter(x, y, label=ticker, color=color, s=100, edgecolors='
                                        black', alpha=0.7)

    if ticker == "VXX":
        plt.annotate(ticker, (x, y), textcoords="offset points", xytext
                                             =(-20, -5),
                     ha='center', fontsize=10, fontweight="bold", color
                                                         ="black")

plt.legend(title="Index/ETF", fontsize=9, bbox_to_anchor=(1.05, 1), loc
                                    ="upper left")
plt.title("Average Daily Return vs. Daily Return Volatility")
plt.xlabel("Daily Volatility (Standard Deviation)")
plt.ylabel("Average Daily Return")
plt.grid(True, linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()



plt.figure(figsize=(8, 6), dpi=150)
```

```
for (ticker, color) in zip(scatter_df["ETF"], colors):
    x, y = scatter_df.loc[scatter_df["ETF"] == ticker, "Daily
                                        Volatility"].values[0], \
           scatter_df.loc[scatter_df["ETF"] == ticker, "Median Daily
                                               Return"].values[0]

    plt.scatter(x, y, label=ticker, color=color, s=100, edgecolors='
                                        black', alpha=0.7)

    if ticker == "VXX":
        plt.annotate(ticker, (x, y), textcoords="offset points", xytext
                                            =(-20, -5),
                     ha='center', fontsize=10, fontweight="bold", color
                                                        ="black")

plt.legend(title="Index/ETF", fontsize=9, bbox_to_anchor=(1.05, 1), loc
                                    ="upper left")
plt.title("Median Daily Return vs. Daily Return Volatility")
plt.xlabel("Daily Volatility (Standard Deviation)")
plt.ylabel("Median Daily Return")
plt.grid(True, linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()

# q6
mean_returns = daily_returns.mean()
median_returns = daily_returns.median()
diff = mean_returns - median_returns

print("Comparison of Mean vs Median Returns:")
print(pd.DataFrame({'Mean': mean_returns, 'Median': median_returns, '
                                    Difference': diff}))


# q7
confidence_levels = [0.01, 0.05]
var_cvar_results = {}

for confidence in confidence_levels:
```

```
    var = daily_returns.quantile(confidence)
    cvar = daily_returns[daily_returns <= var].mean()
    var_cvar_results[f"VaR {int(confidence * 100)}%"] = var
    var_cvar_results[f"CVaR {int(confidence * 100)}%"] = cvar

var_cvar_df = pd.DataFrame(var_cvar_results)
var_cvar_df.to_csv("var_cvar_results.csv", index=True)
print(var_cvar_df)

# q8
selected_etf = 'SPY'
etf_returns = daily_returns[selected_etf]

summary_stats = etf_returns.describe()
kurtosis_value = etf_returns.kurtosis()
skewness_value = etf_returns.skew()

robust_analysis_results = summary_stats.to_frame()
robust_analysis_results.loc['Kurtosis'] = kurtosis_value
robust_analysis_results.loc['Skewness'] = skewness_value
robust_analysis_results.to_csv("robust_analysis_results.csv", index=
                                    True)

plt.figure(figsize=(10,6), dpi=300)
plt.hist(etf_returns, bins=50, alpha=0.75, density=True)
x = np.linspace(etf_returns.min(), etf_returns.max(), 100)
plt.plot(x, norm.pdf(x, etf_returns.mean(), etf_returns.std()), 'r-',
                                    label='Normal Fit')
plt.title(f'Histogram of {selected_etf} Returns')
plt.legend()
plt.show()

# q9
risk_free_rate = 0.02 / 252  # Assuming 2% annualized RF rate converted
                                    to daily
sharpe_ratios = (daily_returns.mean() - risk_free_rate) / daily_returns
                                    .std()
sharpe_ratios.to_csv("sharpe_ratios.csv", index=True)
print(sharpe_ratios)
```

```
# q10
num_etfs = len(daily_returns.columns)
weights = np.ones(num_etfs) / num_etfs
portfolio_returns = daily_returns @ weights
cumulative_returns = (1 + portfolio_returns).cumprod()
rolling_max = cumulative_returns.cummax()
drawdown = (cumulative_returns - rolling_max) / rolling_max
max_drawdown = drawdown.min()

max_drawdown_df = pd.DataFrame({'Max Drawdown': [max_drawdown]})
max_drawdown_df.to_csv("max_drawdown.csv", index=False)

print(max_drawdown_df)

plt.figure(figsize=(10,5), dpi=150)
plt.plot(cumulative_returns, label='Portfolio Cumulative Returns')
plt.title('Cumulative Returns of Equally Weighted Portfolio')
plt.legend()
plt.savefig("cumulative_returns.png", dpi=150)
plt.show()
```

B Returns Analysis Table

Table 5: Returns Analysis

Window	Ticker	Average Return	Median Return
Daily	DBV	0.000011	0.000000
Daily	IWM	0.000424	0.001001
Daily	MTUM	0.000614	0.001105
Daily	SPY	0.000466	0.000678
Daily	VTV	0.000417	0.000746
Daily	VUG	0.000525	0.000918
Daily	VXX	-0.001047	-0.007400
Daily	XLB	0.000420	0.000730
Daily	XLC	0.000573	0.001140
Daily	XLE	0.000469	0.000694
Daily	XLF	0.000394	0.000560
Daily	XLI	0.000431	0.000844
Daily	XLK	0.000477	0.001022
Daily	XLP	0.000305	0.000473
Daily	XLRE	0.000359	0.000895
Daily	XLU	0.000359	0.000880
Daily	XLV	0.000391	0.000574
Daily	XLY	0.000483	0.000915
Weekly	DBV	0.000013	0.000000
Weekly	IWM	0.000417	0.000983
Weekly	MTUM	0.000609	0.001055
Weekly	SPY	0.000465	0.000674
Weekly	VTV	0.000417	0.000794
Weekly	VUG	0.000525	0.000905
Weekly	VXX	-0.001090	-0.007578
Weekly	XLB	0.000418	0.000755
Weekly	XLC	0.000568	0.001170
Weekly	XLE	0.000470	0.000646
Weekly	XLF	0.000392	0.000518
Weekly	XLI	0.000427	0.000822
Weekly	XLK	0.000475	0.001032
Weekly	XLP	0.000303	0.000467
Weekly	XLRE	0.000360	0.000798
Weekly	XLU	0.000356	0.000880
Weekly	XLV	0.000388	0.000576
Weekly	XLY	0.000481	0.000889
Monthly	DBV	0.000013	0.000000
Monthly	IWM	0.000407	0.000983
Monthly	MTUM	0.000601	0.001154
Monthly	SPY	0.000465	0.000674
Monthly	VTV	0.000412	0.000766
Monthly	VUG	0.000523	0.000943
Monthly	VXX	-0.001291	-0.007405
Monthly	XLB	0.000410	0.000752
Monthly	XLC	0.000550	0.001075
Monthly	XLE	0.000474	0.000800
Monthly	XLF	0.000386	0.000455
Monthly	XLI	0.000423	0.000844
Monthly	XLK	0.000471	0.000982
Monthly	XLP	0.000303	0.000447
Monthly	XLRE	0.000346	0.000838
Monthly	XLU	0.000357	0.000898
Monthly	XLV	0.000378	0.000628
Monthly	XLY	0.000476	0.000908