Part. One

1.. I selected two companies, the 1st being 'Continental Illinois' (CONTIL) from the banking industry which at one time was one of the 7th largest commercial bank in the history of the United States. In 1984, Continental Illinois became the largest ever bank failure in U.S. history. The second company I chose was 'BOISE'(BOISE) which is part of the forestry industry. All the uploaded data 'Data_For_Analysis' is in (IN[2]), the index of the newly uploaded data frame to the date column, using the command 'set index' is in (IN[3]), Cleaned data by removing Nan in the selected series(IN[6]).

2. Data is plotted on (In [11]) for both of my selected stocks 'CONTIL' and 'BOISE'. Scatter graph for 'BOISE' is on (IN [24]). Scatter graph for 'CONTIL' is on [IN [43]). From the plotted data graph of both stocks i.e. (IN [11]) we can see during the month of October 1981 both stocks had their lowest return on their stocks within the selected time period. A negative return can be interpreted to mean many things, but it often occurs when a company or business has a financial loss or significant lacklustre returns on an investment. In addition to this a rate of return can be negative when an investor puts money into a company that, due to poor management or factors beyond its control, struggles during the period of investment. It acts as a signal for outside investors that it is not a very wise to invest in the 'BOISE' and 'CONTIL' stocks during that month. From the plotted graph we can work out from both stocks 'BOISE' and 'CONTIL' there is no correlation between the Market returns and the returns on both the selected individual stocks. The trends differ no matter the selected time period. Comparing 'BOISE' stock with the forestry company Weyerhaeuser 'we can see that both companies follow very similar trends in their stock returns, the is exemplified in (IN[109]) where during the selected time period they both experienced one of their highest return in stock at 1982-08-02(0.379 – BOISE) and Weyerhaeuser(0.221 – WEYER). Returns are arguably similar between these two companies because they are in the same industry i.e. Forestry. Comparing 'BOISE' stock with the banking company Citicorp we can see similar trends in their sock returns as shown in (IN [108]). However, there is fluctuations in 1984 stock returns for Continental Illinois. CONTIL experienced a fall in expected return due to its insolvency due to its participation in bad loans purchased from failed banks and risky loans for oil and gas producers.

3. OLS estimate on (IN [47]) For 'CONTIL', estimate on (IN [48]) for 'BOISE'.
P value for 'CONTIL' is 0.302 which is greater than the level of significance 0.05. So, we accept the null hypothesis when testing beta=0 with a 95% confidence level. We also reject the null hypothesis when beta is greater than 1. We can Reject the hypothesis if 0 is not included in the confidence interval for beta. With the CAPM, alpha is the rate of return that exceeds the OLS model's prediction. Investors generally prefer investments with high alpha. For example, if the CAPM analysis indicates that the return should have earned 7%, based on risk, but instead the return earned just 4 % it is considered as discouraging for investors.

When alpha is = 0 we accept the null hypothesis, when alpha is not equal to zero, we reject the null hypothesis. (IN [122]). Comparing ‘CONTIL’ stock with the stock of ‘Citicorp’ which is within the same industry it is very similar to ‘CONTIL’ hypotheses. We accept the null hypothesis when testing for beta=0 and reject it when beta is larger than 1. (IN [128-130]).

4.The standard deviation of the ‘CONTIL’ stock (IN [52]). The standard deviation on the ‘Boise’ stock (IN [53]). The standard deviation is often used by investors to measure the risk of a stock. The basic idea is that the standard deviation is a measure of volatility: the more a stock's returns vary from the stock's average return, the more volatile the stock. To illustrate we can check the mean and variance of one of the stocks such as ‘CONTIL’. In finance, the coefficient of variation allows investors to determine how much volatility, or risk, is assumed in comparison to the amount of return expected from investments. The lower the ratio of standard deviation to mean return, the better risk-return trade-off. ([IN (90)]. Therefore, we can say the better risk return trade of is the ‘BOISE’ stock compared to ‘CONTIL’ stock.

5.Economists believe that R-squared cannot determine whether estimates and predictions are biased, which is why it is imperative to assess the residual plots by the proportion of the risk attributable to the market. R-squared does not indicate whether the OLS regression model is adequate which is why economist believe in risk attributed to the market rather than individual factors. You can have a low R-squared value for a good model, or a high R-squared value for a model that does not fit the data. A high R-squared does not necessarily indicate that the model has a good fit. Economists believe that R squared is the percentage of total risk explained by systematic risk. The selected stock I have chosen shows than values show Boise stock is more correlated the ‘CONTIL’ stock shown in (IN[46]-),(IN[47]) i.e. values of R squared vales for ‘BOISE’ stock is much higher than the Squared value of ‘CONTIL ’stock. Moreover, the more variance that is accounted for by the regression model the closer the data points will fall to the fitted regression line. If a model could explain 100% of the variance, the fitted values would always equal the observed values and, therefore, all the data points would fall on the fitted regression line and r squared will equal to one.

6. Chow test for both stocks (BOISE and CONTIL) is (IN128-130]). P Value for both stocks are greater than the level of significance 0.05, therefore so we accept the null hypothesis. ‘CONTIL’ data contained a structural break, due to a change in policy or sudden shock to the economy. According to the New York Times ‘market *perceptions of Continental's condition deteriorated abruptly in July 1982 with the failure of Penn Square, a relatively small bank in Oklahoma. The late 1970s saw a sharp upward spike in oil prices, which led to rapid growth in opportunities to lend in the oil-producing states, including Oklahoma’. ‘*The is exemplified by the chow test illustrating the structural break during this specific time period of 1984. As a result, continental bank lost more than any other bank having participated in careless oil and gas loan and had to declare $1.3 billion in bankruptcy. The company then had to borrow $3.6 billion from the government, however this was still not enough to make it solvent. However, the Chow test may suggest splitting the data. This may mean fewer degrees of freedom. But

there is the potential for structural instability across the whole data range. It is possible to test every observation for a structural break for both the ‘BOISE’ and ‘CONTIL’ stocks to make sure its stable of over the full time period.

7. Computed data with Na dropped. (IN [76]).

Wald test /F Test– [IN (104}.

AIC and BIC describe goodness of fit of a model, the lower the value the better the goodness of fit. For example, using the Continental Illinois ‘CONTIL; stock, if the CAPM AIC of. -122.9 [IN (142)], and APM has AIC of -117.8 [IN (103)], the CAPM has greater goodness of fit. CAPM was better at modelling the returns on this stock. CAPM is criticized because of the difficulties in selecting a substitution for the market returns and having an appropriate benchmark and target. Using models like the Arbitrage Pricing Model seeks to resolve these issues and is less restrictive than CAPM. The APM is based on assumptions such markets are perfectly competitive, which from the previous analysis shows that companies like ‘Boise’ and ‘Weyerhaeuser’ are in a competitive forestry industry. Another assumption APM make is that investors always prefer more wealth to less wealth. CAPM uses the expected market return whereas, APM uses the expected rate of return and the risk of a number of (macroeconomic) factors. The CAPM lets investors calculate the expected return on investment given the risk, risk-free rate of return, expected market return. The arbitrage pricing model is paired to the CAPM that uses fewer expectations and can be harder to implement than the CAPM. It can be concluded that many investors prefer to use the CAPM, a one-factor model, over the more convoluted APM, which requires users to quantify numerous factors of a model. Evidence suggests APM is not resilient for all stocks. It can be argued that the APM results may not be favourable on stock such as ‘Gerber’ and ‘General Mills’ in the food industry. APM uses the expected stock return as the dependent variable in the same way as CAPM, but a variety of independent variables and can be included, including the market return. Macroeconomic variables such as inflation and exchange rates can also be included. The Wald test is one approach to perform hypothesis testing. It is a way to find out if variables in a model are significant. An advantage of the Wald test over the others is that it only requires the estimation of the OLS model as shown through the results on the table. Wald test /F Test– [IN (104}]. For the ‘Contil’ stock the F test was more than 0.05 so it means all are jointly insignificant and assumes the intercept model is better at explaining variations stock returns. This was also the case for the ‘Boise’ stock.

8. A CAPM approach is useful as it allows us to determine the required return on a risky asset. In equilibrium, all assets should generate the same risk-to-reward ratio. Assets with poor risk-to-reward ratio would be avoided by investors. Without a risk-free rate, we have no tangency that is best for all investors, so portfolio choice will be based on the tangency of risky assets in relation to their degree of risk aversion

PART 2

1.In my sample I decided to conduct a test of normality, the Jarque– Bera test shows a large J-B value indicates that errors are not normally distributed. This is the case ‘CONTIL’, so it hints towards abnormal distribution. The p value for the Contil stock is less than the 5% significance level therefore we can reject the null hypothesis and conclude that the stock is not normally distributed. The Kolmogorov-Smirnov test generally can’t be used for discrete distributions and normality. So, I reject the null hypothesis if $p < 0.05$. So, if $p < 0.05$, we don't believe that the residuals follows a normal distribution. With the ‘Contil’ stock we have a very small deviation and a very large P value greater than 0.05% hinting normal distribution. I used the Anderson-Darling statistic to determine whether the residuals are significant are not. The P value is greater than 0.05. This means that the data might originate from normal distribution at a 5% significance level using the Anderson-Darling test. The shapiro- wilk test shows that if the p-value is less than 0.05, then the null hypothesis which states whether the data is normally distributed is rejected. If the p-value is greater than 0.05, then the null hypothesis is not rejected. The p value is greater than the level of significance, so we accept the shapiro wilk test for the Contil stock. For the Boise stock we have a very small deviation and a large P value hinting normal distribution. Overall, we should be worried if they do not result in normal distribution due to the fact that normality tests are only needed for small sample sizes. Most of the residuals tested tend to be selected with a smaller sample size. It can be argued that small sample sizes results in low significance power for normality tests. But it also means that significant deviations from normality will not result in significance. Tests can say there's no deviation from normality while it could theoretically be huge.

2. For naive forecasts, we set all forecasts to be the value of the last observation. This method works well for many economic and financial models. Normally the higher the number, the worse it affects a company because a widely inaccurate forecast makes it impossible to plan for firms and companies. This was not the case for my selected stocks. The problem with this method is that it expects the future to be like the last. This is not a good approach as it is unsophisticated and an ingenious way of plotting data. The simple moving average is calculated as the arithmetic average of a stock returns price over a specific period. It is thereby simpler to calculate; however, it relies on more historic data. Moving Average can be an effective method of forecasting in some instances, you can get better accuracy by combining forecasting method, not taking all the data, and taking specific observations. Simple Exponential Smoothing - the exponential smoothing method is going to use historical data to forecast. One of the benefits of this model is that it takes the most current observations into account and weights them appropriately. Another benefit is that spikes in the data aren’t quite as unfavourable to the forecast as previous methods such as the simple average method or the moving average method. The most recent forecast has the highest weight and therefore should be the most precise in calculating requirements, contrasting to the moving averages method where the weight for each period is stable. However, the exponential smoothing method limits our ability to forecast demand using seasonality. The

Holt’s Linear Trend method extends the simple exponential smoothing method to allow the forecasting of data with a trend. This method implicates a forecast equation and two more equations. However, the difficulty with this method is that, sometimes different accuracy measures will suggest different forecasting methods, and then a decision will them have to be required as to which forecasting method to use. Holt’s forecasting method can vary by many proportions. ARIMA forecasts are usually more accurate and reliable. ARIMA is a univariate model (working with one variable only). However, if other variables are important, then a multivariate model a better choice. A distinct disadvantage of The ARIMA model is that it tends to be unstable, both with respect to changes in observations and changes in model requirements. Relating to the MSE - lower the value the better is the forecasting (as it is closer to the actual mean value observed). shows that the moving average forecast was a better forecasting method than the simple average forecasting method.

3. The Breusch–Godfrey Test shows us that the null hypothesis for a test means the error variances are all equal. The alternate hypothesis is that the error variances are not equal. The test is important because it shows heteroskedasticity could still be present at any given time in my regression model. It is more appropriate for a general regression model such as the OLS regression model which I have been using for my selected stock ‘BOISE’ and ‘CONTIL’, so this is a strong econometric technique to use. The Durbin-Watson statistic will always have a value between 0 and 4. A value of 2.0 means that there is no autocorrelation detected in the sample. Values from 0 to less than 2 indicate positive autocorrelation and values from 2 to 4 indicate negative autocorrelation. This is a good econometric technique because the Autocorrelation can be useful in technical analysis, which is most concerned with the certain trends of stocks and in certain instances a company's financial performance. [IN66] In my coursework a Durbin Watson statistical test was done the Continental Illinois’ stock and it showed no auto correlation. The Watson test gave us may give conclusive results compared to other techniques such as the Jarque Bera test. The White test is explicitly intended to test for forms of heteroskedasticity. One of the main benefits of using the white test for heteroskedasticity is it does not rely on the normality assumptions and it is also easy to implement. This test is not perceptive to normality disruptions, which means it is ideal to use for my selected stocks.

4. My stocks showed me how useful it is testing certain econometric theories in finance and stock returns. It also taught the importance of testing hypotheses regarding the relationships between variables. I saw the effect on stock markets of changes in economic conditions. Furthermore, I learned that Econometric forecasters have a history of neglecting future crises for example the in 1984 Continental Illinois became the largest ever bank failure in U.S. history. Consequently, I saw the effect it had on the stock returns. Moreover, the most important thing I learned is that the better the forecast the more you are able to oversee the potential of returns and understandably how that could potentially generate more money and success for a company.

In [1]:

```
import numpy as np, pandas as pd
import statistics as st
import statsmodels.api as sm
import RamiFunctions as RF, statsmodels.formula.api as smf
# from yahoofinancials import YahooFinancials
import statsmodels.tsa.api as smt

import scipy.stats as ss
import scipy.stats.mstats as st1
from scipy import stats
import pandas_datareader.data as web
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import iqr
from statsmodels.graphics.gofplots import qqplot
from matplotlib import pyplot
import statsmodels.stats._adnorm
import statsmodels.stats.api as sms
from statsmodels.compat import lzip
from statsmodels.stats.diagnostic import het_breuschpagan
from statsmodels.stats.diagnostic import het_white

from statsmodels.stats.diagnostic import acorr_breusch_godfrey

import warnings
warnings.filterwarnings('ignore')



# Applying Chow test for Market vs Stock
def get_rss(y, x):
    '''
    Inputs:
    formula_model: Is the regression equation that you are run
ning
    data: The DataFrame of the independent and dependent varia
ble

    Outputs:
    rss: the sum of residuals
    N: the observations of inputs
    K: total number of parameters
    '''
    x = sm.add_constant(x)
```

```
    x = sm.add_constant(x)
    results=sm.OLS(y, x).fit()
    rss= (results.resid**2).sum()
    N=results.nobs
    K=results.df_model
    return rss, N, K, results


def Chow_Test(df, y, x, special_date, level_of_sig=0.05):

    from scipy.stats import f
    date=special_date
    x1=df[x][:date]
    y1=df[y][:date]
    x2=df[x][date:]
    y2=df[y][date:]

    RSS_total, N_total, K_total, results=get_rss(df[y], df[x])
    RSS_1, N_1, K_1, results1=get_rss(y1, x1)
    RSS_2, N_2, K_2, results2=get_rss(y2, x2)
    num=(RSS_total-RSS_1-RSS_2)/K_total
    den=(RSS_1+RSS_2)/(N_1+N_2-2*K_total)

    p_val = f.sf(num/den, 2, N_1+N_2-2*K_total)


    df['Before_Special'] = np.where(df.index<special_date , 'Before', 'After')
    g = sns.lmplot(x=x, y=y, hue="Before_Special", truncate=True, height=5, markers=["o", "x"], data=df)
    # return df

    if p_val<level_of_sig:
        print('The P vale {:3.5f} is lower than the level of significance {}. Therefore, reject the null that the coefficients are the same in the two periods are equal'.format(p_val,level_of_sig))
    else:
        print('The P vale {:3.5f} is higher than the level of significance {}. Therefore, accept the null that the coefficients are the same in the two periods are equal'.format(p_val,level_of_sig))

    return num/den, p_val

def tsplot(y, lags=None, figsize=(10, 8), style='bmh'):
    if not isinstance(y, pd.Series):
        y = pd.Series(y)
    with plt.style.context(style):
```

```
        fig = plt.figure(figsize=figsize)
        #mpl.rcParams['font.family'] = 'Ubuntu Mono'
        layout = (3, 2)
        ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
        acf_ax = plt.subplot2grid(layout, (1, 0))
        pacf_ax = plt.subplot2grid(layout, (1, 1))
        qq_ax = plt.subplot2grid(layout, (2, 0))
        pp_ax = plt.subplot2grid(layout, (2, 1))

        y.plot(ax=ts_ax)
        ts_ax.set_title('Time Series Analysis Plots')
        smt.graphics.plot_acf(y, lags=lags, ax=acf_ax, alpha=0
.05)
        smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax, alpha
=0.05)
        sm.qqplot(y, line='s', ax=qq_ax)
        qq_ax.set_title('QQ Plot')
        ss.probplot(y, sparams=(y.mean(), y.std()), plot=pp_ax
)

        plt.tight_layout()
    return


def Naive_Forecast(df_in_sample, df_out_sample, column):
    a=df_in_sample.tail(1).values
    df_naive_method_forecast=pd.DataFrame(np.repeat(a, df_out_
sample.shape[0], axis=0))
    df_naive_method_forecast.columns=df_out_sample.columns
    # df_naive_method_forecast.head(3)

    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='
Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label
='Actual Data')
    plt.plot(df_out_sample.index, df_naive_method_forecast[col
umn], label='Naive Forecast Data')
    plt.legend(loc='best')
    plt.title("Naive Forecast")
    plt.show()
    df_naive_method_forecast.index=df_out_sample.index

    return df_naive_method_forecast


def Average_Forecast(df_in_sample, df_out_sample, column):
    a=df_in_sample.mean().values
```

```
    a=a.repeat(df_out_sample.shape[0]).reshape(a.shape[0], df_out_sample.shape[0]).transpose()
    df_Average_Forecast=pd.DataFrame(a)
    df_Average_Forecast.columns=df_out_sample.columns
    df_Average_Forecast.index=df_out_sample.index

    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label='Actual Data')
    plt.plot(df_out_sample.index, df_Average_Forecast[column], label='Simple Average Forecast Data')
    plt.legend(loc='best')
    plt.title("Simple Average Forecast")
    plt.show()
    df_Average_Forecast.index=df_out_sample.index

    return df_Average_Forecast


def Moving_Average_Forecast(df_in_sample, df_out_sample, column, window_leng):
    a=df_in_sample.rolling(window_leng).mean().iloc[-1].values
    a=a.repeat(df_out_sample.shape[0]).reshape(a.shape[0], df_out_sample.shape[0]).transpose()
    df_Moving_Average_Forecast=pd.DataFrame(a)
    df_Moving_Average_Forecast.columns=df_out_sample.columns
    df_Moving_Average_Forecast.index=df_out_sample.index

    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label='Actual Data')
    plt.plot(df_out_sample.index, df_Moving_Average_Forecast[column], label='Moving Average Forecast Data')
    plt.legend(loc='best')
    plt.title("Simple Moving Average Forecast")
    plt.show()

    df_Moving_Average_Forecast.index=df_out_sample.index

    return df_Moving_Average_Forecast


def Simple_Exponential_Smoothing_Forecast(df_in_sample, df_out_sample, column, level):
```

```
    a=[]

    for col in df_in_sample.columns:
        fit2 =smt.SimpleExpSmoothing(np.asarray(df_in_sample[col])).fit(smoothing_level=level, optimized=False)
        a.append(fit2.forecast())
    a=np.array(a)
    a=a.repeat(df_out_sample.shape[0]).reshape(a.shape[0], df_out_sample.shape[0]).transpose()
    df_Simple_Exponential_Smoothing_Forecast=pd.DataFrame(a)
    df_Simple_Exponential_Smoothing_Forecast.columns=df_out_sample.columns
    df_Simple_Exponential_Smoothing_Forecast.index=df_out_sample.index
    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label='Actual Data')
    plt.plot(df_out_sample.index, df_Simple_Exponential_Smoothing_Forecast[column], label='Simple Exponential Smoothing Forecast Data')
    plt.legend(loc='best')
    plt.title("Simple Exponential Smoothing Forecast")
    plt.show()
    df_Simple_Exponential_Smoothing_Forecast.index=df_out_sample.index

    return df_Simple_Exponential_Smoothing_Forecast


# Here we are conducting a one tail test by speecifing if the alternative is "two-sided", "larger", or "smaller"

# def ttest_OLS(res, numberofbeta, X, value=0, alternative='two-sided', level_of_sig = 0.05):
#     results=np.zeros([2])
#     # numberofbeta represent the coeffiecent you would like to test 0 standts for interecept
#     results[0]=res.tvalues[numberofbeta]
#     results[1]=res.pvalues[numberofbeta]
#     if isinstance(X, pd.DataFrame):
#         column=X.columns[numberofbeta]
#     else:
#         column=numberofbeta
#     if alternative == 'two-sided':
#         if results[1]<level_of_sig:
#             print("We reject the null hypothesis that the Selected Coefficient: {} is equal to {} with a {} % significance
```

```
level".format(column, value, level_of_sig*100))
#         else: print("We accept the null hypothesis that the Selected Coefficient: {} is equal to {} with a {} % significance level".format(column, value, level_of_sig*100))
#     elif alternative == 'larger':
#         if (results[0] > 0) & (results[1]/2 < level_of_sig):
#             print("We reject the null hypothesis that the Selected Coefficient: {} is less than {} with a {} % significance level".format(column, value, level_of_sig*100))
#         else: print("We accept the null hypothesis that the Selected Coefficient: {} is less than {} with a {} % significance level".format(column, value, level_of_sig*100))

#     elif alternative == 'smaller':
#         if (results[0] < 0) & (results[1]/2 < level_of_sig):
#             print("We reject the null hypothesis that the Selected Coefficient: {} is more than {} with a {} % significance level".format(column, value, level_of_sig*100))
#         else: print("We accept the null hypothesis that the Selected Coefficient: {} is more than {} with a {} % significance level".format(column, value, level_of_sig*100))

def Simple_ttest_Ols(results, hypothesis, alternative='two-sided', level_of_sig = 0.05):
    results1=np.zeros([2])
    t_test = results.t_test(hypothesis)
    results1[0]=t_test.tvalue
    results1[1]=t_test.pvalue
    if alternative == 'two-sided':
        if results1[1]<level_of_sig:
            print("We reject the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
        else: print("We accept the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
    elif alternative == 'larger':
        if (results1[0] > 0) & (results1[1]/2 < level_of_sig):
            print("We reject the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
        else: print("We accept the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))

    elif alternative == 'smaller':
        if (results1[0] < 0) & (results1[1]/2 < level_of_sig):
            print("We reject the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
        else: print("We accept the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
```

```
    pass




def Joining_columns(df1, x, y=None,Name_of_new_column=None):
    # Find all columns except x
    f=df1.columns.to_list()
    f.remove(x)
    if y!=None:
        df2=pd.melt(df1, id_vars=[x], value_vars=y)
    else:
        df2=pd.melt(df1, id_vars=[x], value_vars=f)
    if Name_of_new_column!=None:
        df2=df2.rename(columns={"value": Name_of_new_column})
    return df2




def get_betas_SLR(df1, x, column=None):
    if column==None:
    # Choose all the columns except the x
        f=df1.columns.to_list()
        f.remove(x)
        column=f
    A=np.zeros([len(column)])
    j=0
    for i in column:
        formula ='Q("'+ i+'"'+') ~ Q("Excess Market Returns")'
        results = smf.ols(formula, df1).fit()
        A[j]=results.params[1]
        j=j+1

    A=pd.DataFrame(data=A,columns=['Beta'], index=column)
    return A


def Get_indicators(BB, indicat):
    # BB represents the BB=yahoo_financials.get_key_statistics
_data()
    V=np.zeros([len(BB)])
    j=0
    for i in BB.keys():
        V[j]=BB[i][indicat]
        j=j+1
    return V

# Examples V=Get_indicators(BB, 'priceToBook')
```

```
# Get Data from a dictionary downloaded from yahoo finance
def Get_Dataframe_of_tickes(tickers):
    yahoo_financials = YahooFinancials(tickers)
    BB=yahoo_financials.get_key_statistics_data()
    dict_of_df = {k: pd.DataFrame.from_dict(v, orient='index')
for k,v in BB.items()}
    df = pd.concat(dict_of_df, axis=1)
    return df

# Examples
# tickers = ['AAPL', 'WFC', 'F', 'FB', 'DELL', 'SNE','NOK', 'M
SFT', 'JPM', 'GE', 'BAC']
# Name=['Apple', 'Wells_Fargo_Company', 'Ford Motor Company',
'Facebook', 'Dell Technologies', 'Sony', 'Nokia', 'Microsoft',
'JPMorgan Chase & Co', 'General Electric', 'Bank of America']
# df=RF.Get_Dataframe_of_tickes(tickers)


def Get_Yahoo_stats(tickers):
    yahoo_financials = YahooFinancials(tickers)
    f=['get_interest_expense()', 'get_operating_income()', 'ge
t_total_operating_expense()', 'get_total_revenue()', 'get_cost
_of_revenue()', 'get_income_before_tax()', 'get_income_tax_exp
ense()', 'get_gross_profit()', 'get_net_income_from_continuing
_ops()', 'get_research_and_development()', 'get_current_price(
)', 'get_current_change()', 'get_current_percent_change()', 'g
et_current_volume()', 'get_prev_close_price()', 'get_open_pric
e()', 'get_ten_day_avg_daily_volume()', 'get_three_month_avg_d
aily_volume()', 'get_stock_exchange()', 'get_market_cap()', 'g
et_daily_low()', 'get_daily_high()', 'get_currency()', 'get_ye
arly_high()', 'get_yearly_low()', 'get_dividend_yield()', 'get
_annual_avg_div_yield()', 'get_five_yr_avg_div_yield()', 'get_
dividend_rate()', 'get_annual_avg_div_rate()', 'get_50day_movi
ng_avg()', 'get_200day_moving_avg()', 'get_beta()', 'get_payou
t_ratio()', 'get_pe_ratio()', 'get_price_to_sales()', 'get_exd
ividend_date()', 'get_book_value()', 'get_ebit()', 'get_net_in
come()', 'get_earnings_per_share()', 'get_key_statistics_data(
)']
    i=0
    exec('d=yahoo_financials.'+f[i], locals(), globals())
    col=f[i].replace("get_","").replace("()","")
    A=pd.DataFrame.from_dict(d, orient='index', columns=[col])
    for i in range(1,3):
        exec('d=yahoo_financials.'+f[i], locals(), globals())
        col=f[i].replace("get_","").replace("()","").replace("
_"," ")
        B=pd.DataFrame.from_dict(d, orient='index', columns=[c
```

```
ol])
        A= pd.concat([A, B], axis=1, sort=False)
    return A

# from yahoofinancials import YahooFinancials
# tickers = ['AAPL', 'WFC', 'F', 'FB', 'DELL', 'SNE']
# Get_Yahoo_stats(tickers)

def forward_selected(data, response):
    """Linear model designed by forward selection.

    Parameters:
    -----------
    data : pandas DataFrame with all possible predictors and r
esponse

    response: string, name of response column in data

    Returns:
    --------
    model: an "optimal" fitted statsmodels linear model
           with an intercept
           selected by forward selection
           evaluated by adjusted R-squared
    """
    remaining = set(data.columns)
    remaining.remove(response)
    selected = []
    current_score, best_new_score = 0.0, 0.0
    while remaining and current_score == best_new_score:
        scores_with_candidates = []
        for candidate in remaining:
            formula = "{} ~ {} + 1".format(response, ' + '.joi
n(selected + [candidate]))

            score = smf.ols(formula, data).fit().rsquared_adj
            scores_with_candidates.append((score, candidate))
        scores_with_candidates.sort()

        best_new_score, best_candidate = scores_with_candidate
s.pop()
        if current_score < best_new_score:
            remaining.remove(best_candidate)
            selected.append(best_candidate)
            current_score = best_new_score
    formula = "{} ~ {} + 1".format(response,
                                   ' + '.join(selected))
    model = smf.ols(formula, data).fit()
```

```
    return model


# Examples

# data = sm.datasets.longley.load_pandas()
# df1=data.data
# formula = 'GNP ~ YEAR + UNEMP + POP + GNPDEFL'
# results = smf.ols(formula, df1).fit()
# print(results.summary())
# res = RF.forward_selected(df1, 'GNP')
# print(res.model.formula)
# print(res.rsquared_adj)
# print(res.summary())


def GQTest(lm2, level_of_sig=0.05, sp=None):
    name = ['F statistic', 'p-value']
    test = sms.het_goldfeldquandt(lm2.resid, lm2.model.exog, s
plit=sp)
    R=lzip(name, test)

    if test[1]>level_of_sig:
        print('The P vale of this test is {:3.5f}, which is gr
eater than the level of significance {} therefore, we accept t
he null that the error terms are homoscedastic'.format(test[1]
, level_of_sig))
    else:
        print('The P vale of this test is {:3.5f}, which is sm
aller than the level of significance {} therefore, we reject t
he null, hence the error terms are hetroscedastic'.format(test
[1], level_of_sig))
    return R


def WhiteTest(statecrime_model, level_of_sig=0.05):
    white_test = het_white(statecrime_model.resid,  statecrime
_model.model.exog)
    labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic'
, 'F-Test p-value']
    R=dict(zip(labels, white_test))

    if white_test[3]>level_of_sig:
        print('The P vale of this test is {:3.5f}, which is gr
eater than the level of significance {} therefore, we accept t
he null that the error terms are homoscedastic'.format(white_t
est[3], level_of_sig))
    else:
        print('The P vale of this test is {:3.5f}, which is sm
```

```
aller than the level of significance {} therefore, we reject t
he null, hence the error terms are hetroscedastic'.format(whit
e_test[3], level_of_sig))
    return R


def Plot_resi_corr(results):
    res_min_1=results.resid[:-1]
    res_plus_1=results.resid[1:]
    data1=pd.DataFrame(np.column_stack((res_min_1.T,res_plus_1
.T)), columns=['u_t-1','u_t'])
    sns.set()
    plt.figure(figsize=(5,5))
    ax = sns.scatterplot(x='u_t-1', y='u_t', data=data1)
    pass

def Plot_resi_corr_time(results,df):
    C=pd.DataFrame(results.resid, index=df.index, columns=['Re
siduals'])
    C.plot(figsize=(10,5), linewidth=1.5, fontsize=10)
    plt.xlabel('Date', fontsize=10);
    return C


def Breusch_Godfrey(results, level_of_sig=0.05, lags=None):
    A=acorr_breusch_godfrey(results, nlags=lags)
    labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic'
, 'F-Test p-value']
    R=dict(zip(labels, A))

    if A[3]>level_of_sig:
        print('The P vale of this test is {:3.5f}, which is gr
eater than the level of significance {} therefore, we accept t
he null that the error terms are not Auto-corrolated'.format(A
[3], level_of_sig))
    else:
        print('The P vale of this test is {:3.5f}, which is sm
aller than the level of significance {} therefore, we reject t
he null, hence the error terms are Auto-corrolated'.format(A[3
], level_of_sig))
    return R

def Create_lags_of_variable(MainDF_first_period, lags, column)
:
    # Crete a new dataframe based on the lag variables
    x=column
    if type(lags) == int:
        j=lags
```

```
        values=MainDF_first_period[x]
        dataframe = pd.concat([values.shift(j), values], axis=
1)
        dataframe.columns = [x+' at time t-'+str(j), x+' at ti
me t']
        dataframe=dataframe.dropna()
    else:
        values=MainDF_first_period[x]
        dataframe=values
        for j in lags:
            dataframe = pd.concat([values.shift(j), dataframe]
, axis=1)
        c=[x+' for time t-'+str(j) for j in range(len(lags),-1
,-1)]
        dataframe.columns=c
        dataframe=dataframe.dropna()
    return dataframe

# # Things students shouldn't know
# MainDF1=MainDF.reset_index(drop=False)
# df2=pd.melt(MainDF1, id_vars=MainDF.index.name)
# palette = dict(zip(df2['variable'].unique(), sns.color_palet
te("rocket_r", len(df2['variable'].unique()))))
# sns.relplot(x=MainDF.index.name, y="value",
#             hue="variable", palette=palette,
#             height=5, aspect=3, facet_kws=dict(sharex=False)
, kind="line", data=df2)import numpy as np, pandas as pd
import statistics as st
import statsmodels.api as sm
import RamiFunctions as RF, statsmodels.formula.api as smf
# from yahoofinancials import YahooFinancials
import statsmodels.tsa.api as smt

import scipy.stats as ss
import scipy.stats.mstats as st1
from scipy import stats
import pandas_datareader.data as web
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import iqr
from statsmodels.graphics.gofplots import qqplot
from matplotlib import pyplot
import statsmodels.stats._adnorm
import statsmodels.stats.api as sms
from statsmodels.compat import lzip
from statsmodels.stats.diagnostic import het_breuschpagan
from statsmodels.stats.diagnostic import het_white
```

```python
from statsmodels.stats.diagnostic import acorr_breusch_godfrey

import warnings
warnings.filterwarnings('ignore')



# Applying Chow test for Market vs Stock
def get_rss(y, x):
    '''
    Inputs:
    formula_model: Is the regression equation that you are running
    data: The DataFrame of the independent and dependent variable

    Outputs:
    rss: the sum of residuals
    N: the observations of inputs
    K: total number of parameters
    '''
    x = sm.add_constant(x)
    results=sm.OLS(y, x).fit()
    rss= (results.resid**2).sum()
    N=results.nobs
    K=results.df_model
    return rss, N, K, results


def Chow_Test(df, y, x, special_date, level_of_sig=0.05):

    from scipy.stats import f
    date=special_date
    x1=df[x][:date]
    y1=df[y][:date]
    x2=df[x][date:]
    y2=df[y][date:]

    RSS_total, N_total, K_total, results=get_rss(df[y], df[x])
    RSS_1, N_1, K_1, results1=get_rss(y1, x1)
    RSS_2, N_2, K_2, results2=get_rss(y2, x2)
    num=(RSS_total-RSS_1-RSS_2)/K_total
    den=(RSS_1+RSS_2)/(N_1+N_2-2*K_total)

    p_val = f.sf(num/den, 2, N_1+N_2-2*K_total)
```

```
    df['Before_Special'] = np.where(df.index<special_date , 'Before', 'After')
    g = sns.lmplot(x=x, y=y, hue="Before_Special", truncate=True, height=5, markers=["o", "x"], data=df)
    # return df

    if p_val<level_of_sig:
        print('The P vale {:3.5f} is lower than the level of significance {}. Therefore, reject the null that the coefficients are the same in the two periods are equal'.format(p_val,level_of_sig))
    else:
        print('The P vale {:3.5f} is higher than the level of significance {}. Therefore, accept the null that the coefficients are the same in the two periods are equal'.format(p_val,level_of_sig))

    return num/den, p_val

def tsplot(y, lags=None, figsize=(10, 8), style='bmh'):
    if not isinstance(y, pd.Series):
        y = pd.Series(y)
    with plt.style.context(style):
        fig = plt.figure(figsize=figsize)
        #mpl.rcParams['font.family'] = 'Ubuntu Mono'
        layout = (3, 2)
        ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
        acf_ax = plt.subplot2grid(layout, (1, 0))
        pacf_ax = plt.subplot2grid(layout, (1, 1))
        qq_ax = plt.subplot2grid(layout, (2, 0))
        pp_ax = plt.subplot2grid(layout, (2, 1))

        y.plot(ax=ts_ax)
        ts_ax.set_title('Time Series Analysis Plots')
        smt.graphics.plot_acf(y, lags=lags, ax=acf_ax, alpha=0.05)
        smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax, alpha=0.05)
        sm.qqplot(y, line='s', ax=qq_ax)
        qq_ax.set_title('QQ Plot')
        ss.probplot(y, sparams=(y.mean(), y.std()), plot=pp_ax)

        plt.tight_layout()
    return


def Naive_Forecast(df_in_sample, df_out_sample, column):
```

```
    a=df_in_sample.tail(1).values

    df_naive_method_forecast=pd.DataFrame(np.repeat(a, df_out_sample.shape[0], axis=0))
    df_naive_method_forecast.columns=df_out_sample.columns
    # df_naive_method_forecast.head(3)

    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label='Actual Data')
    plt.plot(df_out_sample.index, df_naive_method_forecast[column], label='Naive Forecast Data')
    plt.legend(loc='best')
    plt.title("Naive Forecast")
    plt.show()
    df_naive_method_forecast.index=df_out_sample.index

    return df_naive_method_forecast


def Average_Forecast(df_in_sample, df_out_sample, column):
    a=df_in_sample.mean().values
    a=a.repeat(df_out_sample.shape[0]).reshape(a.shape[0], df_out_sample.shape[0]).transpose()
    df_Average_Forecast=pd.DataFrame(a)
    df_Average_Forecast.columns=df_out_sample.columns
    df_Average_Forecast.index=df_out_sample.index

    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label='Actual Data')
    plt.plot(df_out_sample.index, df_Average_Forecast[column], label='Simple Average Forecast Data')
    plt.legend(loc='best')
    plt.title("Simple Average Forecast")
    plt.show()
    df_Average_Forecast.index=df_out_sample.index

    return df_Average_Forecast


def Moving_Average_Forecast(df_in_sample, df_out_sample, column, window_leng):
    a=df_in_sample.rolling(window_leng).mean().iloc[-1].values
    a=a.repeat(df_out_sample.shape[0]).reshape(a.shape[0], df_
```

```
out_sample.shape[0]).transpose()

    df_Moving_Average_Forecast=pd.DataFrame(a)
    df_Moving_Average_Forecast.columns=df_out_sample.columns
    df_Moving_Average_Forecast.index=df_out_sample.index

    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label='Actual Data')
    plt.plot(df_out_sample.index, df_Moving_Average_Forecast[column], label='Moving Average Forecast Data')
    plt.legend(loc='best')
    plt.title("Simple Moving Average Forecast")
    plt.show()

    df_Moving_Average_Forecast.index=df_out_sample.index

    return df_Moving_Average_Forecast


def Simple_Exponential_Smoothing_Forecast(df_in_sample, df_out_sample, column, level):
    a=[]
    for col in df_in_sample.columns:
        fit2 =smt.SimpleExpSmoothing(np.asarray(df_in_sample[col])).fit(smoothing_level=level, optimized=False)
        a.append(fit2.forecast())
    a=np.array(a)
    a=a.repeat(df_out_sample.shape[0]).reshape(a.shape[0], df_out_sample.shape[0]).transpose()
    df_Simple_Exponential_Smoothing_Forecast=pd.DataFrame(a)
    df_Simple_Exponential_Smoothing_Forecast.columns=df_out_sample.columns
    df_Simple_Exponential_Smoothing_Forecast.index=df_out_sample.index
    plt.figure(figsize=(7,4))
    plt.plot(df_in_sample.index, df_in_sample[column], label='Training Data')
    plt.plot(df_out_sample.index, df_out_sample[column], label='Actual Data')
    plt.plot(df_out_sample.index, df_Simple_Exponential_Smoothing_Forecast[column], label='Simple Exponential Smoothing Forecast Data')
    plt.legend(loc='best')
    plt.title("Simple Exponential Smoothing Forecast")
    plt.show()
    df_Simple_Exponential_Smoothing_Forecast.index=df_out_samp
```

```
le.index

    return df_Simple_Exponential_Smoothing_Forecast


# Here we are conducting a one tail test by speecifing if the alternative is "two-sided", "larger", or "smaller"

# def ttest_OLS(res, numberofbeta, X, value=0, alternative='two-sided', level_of_sig = 0.05):
#     results=np.zeros([2])
#     # numberofbeta represent the coeffiecent you would like to test 0 standts for interecept
#     results[0]=res.tvalues[numberofbeta]
#     results[1]=res.pvalues[numberofbeta]
#     if isinstance(X, pd.DataFrame):
#         column=X.columns[numberofbeta]
#     else:
#         column=numberofbeta
#     if alternative == 'two-sided':
#         if results[1]<level_of_sig:
#             print("We reject the null hypothesis that the Selected Coefficient: {} is equal to {} with a {} % significance level".format(column, value, level_of_sig*100))
#         else: print("We accept the null hypothesis that the Selected Coefficient: {} is equal to {} with a {} % significance level".format(column, value, level_of_sig*100))
#     elif alternative == 'larger':
#         if (results[0] > 0) & (results[1]/2 < level_of_sig):
#             print("We reject the null hypothesis that the Selected Coefficient: {} is less than {} with a {} % significance level".format(column, value, level_of_sig*100))
#         else: print("We accept the null hypothesis that the Selected Coefficient: {} is less than {} with a {} % significance level".format(column, value, level_of_sig*100))

#     elif alternative == 'smaller':
#         if (results[0] < 0) & (results[1]/2 < level_of_sig):
#             print("We reject the null hypothesis that the Selected Coefficient: {} is more than {} with a {} % significance level".format(column, value, level_of_sig*100))
#         else: print("We accept the null hypothesis that the Selected Coefficient: {} is more than {} with a {} % significance level".format(column, value, level_of_sig*100))

def Simple_ttest_Ols(results, hypothesis, alternative='two-sided', level_of_sig = 0.05):
    results1=np.zeros([2])
```

```
    t_test = results.t_test(hypothesis)

    results1[0]=t_test.tvalue
    results1[1]=t_test.pvalue
    if alternative == 'two-sided':
        if results1[1]<level_of_sig:
            print("We reject the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
        else: print("We accept the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
    elif alternative == 'larger':
        if (results1[0] > 0) & (results1[1]/2 < level_of_sig):
            print("We reject the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
        else: print("We accept the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))

    elif alternative == 'smaller':
        if (results1[0] < 0) & (results1[1]/2 < level_of_sig):
            print("We reject the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))
        else: print("We accept the null hypothesis: {} with a {} % significance level".format(hypothesis, level_of_sig*100))

    pass



def Joining_columns(df1, x, y=None,Name_of_new_column=None):
    # Find all columns except x
    f=df1.columns.to_list()
    f.remove(x)
    if y!=None:
        df2=pd.melt(df1, id_vars=[x], value_vars=y)
    else:
        df2=pd.melt(df1, id_vars=[x], value_vars=f)
    if Name_of_new_column!=None:
        df2=df2.rename(columns={"value": Name_of_new_column})
    return df2



def get_betas_SLR(df1, x, column=None):
    if column==None:
    # Choose all the columns except the x
        f=df1.columns.to_list()
        f.remove(x)
        column=f
    A=np.zeros([len(column)])
```

```
    j=0

    for i in column:
        formula ='Q("'+ i+'"'+') ~ Q("Excess Market Returns")'
        results = smf.ols(formula, df1).fit()
        A[j]=results.params[1]
        j=j+1

    A=pd.DataFrame(data=A,columns=['Beta'], index=column)
    return A


def Get_indicators(BB, indicat):
    # BB represents the BB=yahoo_financials.get_key_statistics
_data()
    V=np.zeros([len(BB)])
    j=0
    for i in BB.keys():
        V[j]=BB[i][indicat]
        j=j+1
    return V

# Examples V=Get_indicators(BB, 'priceToBook')


# Get Data from a dictionary downloaded from yahoo finance
def Get_Dataframe_of_tickes(tickers):
    yahoo_financials = YahooFinancials(tickers)
    BB=yahoo_financials.get_key_statistics_data()
    dict_of_df = {k: pd.DataFrame.from_dict(v, orient='index')
for k,v in BB.items()}
    df = pd.concat(dict_of_df, axis=1)
    return df

# Examples
# tickers = ['AAPL', 'WFC', 'F', 'FB', 'DELL', 'SNE','NOK', 'M
SFT', 'JPM', 'GE', 'BAC']
# Name=['Apple', 'Wells_Fargo_Company', 'Ford Motor Company',
'Facebook', 'Dell Technologies', 'Sony', 'Nokia', 'Microsoft',
'JPMorgan Chase & Co', 'General Electric', 'Bank of America']
# df=RF.Get_Dataframe_of_tickes(tickers)


def Get_Yahoo_stats(tickers):
    yahoo_financials = YahooFinancials(tickers)
    f=['get_interest_expense()', 'get_operating_income()', 'ge
t_total_operating_expense()', 'get_total_revenue()', 'get_cost
_of_revenue()', 'get_income_before_tax()', 'get_income_tax_exp
ense()', 'get_gross_profit()', 'get_net_income_from_continuing
```

```
_ops()', 'get_research_and_development()', 'get_current_price(
)', 'get_current_change()', 'get_current_percent_change()', 'g
et_current_volume()', 'get_prev_close_price()', 'get_open_pric
e()', 'get_ten_day_avg_daily_volume()', 'get_three_month_avg_d
aily_volume()', 'get_stock_exchange()', 'get_market_cap()', 'g
et_daily_low()', 'get_daily_high()', 'get_currency()', 'get_ye
arly_high()', 'get_yearly_low()', 'get_dividend_yield()', 'get
_annual_avg_div_yield()', 'get_five_yr_avg_div_yield()', 'get_
dividend_rate()', 'get_annual_avg_div_rate()', 'get_50day_movi
ng_avg()', 'get_200day_moving_avg()', 'get_beta()', 'get_payou
t_ratio()', 'get_pe_ratio()', 'get_price_to_sales()', 'get_exd
ividend_date()', 'get_book_value()', 'get_ebit()', 'get_net_in
come()', 'get_earnings_per_share()', 'get_key_statistics_data(
)']
    i=0
    exec('d=yahoo_financials.'+f[i], locals(), globals())
    col=f[i].replace("get_","").replace("()","")
    A=pd.DataFrame.from_dict(d, orient='index', columns=[col])
    for i in range(1,3):
        exec('d=yahoo_financials.'+f[i], locals(), globals())
        col=f[i].replace("get_","").replace("()","").replace("
_"," ")
        B=pd.DataFrame.from_dict(d, orient='index', columns=[c
ol])
        A= pd.concat([A, B], axis=1, sort=False)
    return A

# from yahoofinancials import YahooFinancials
# tickers = ['AAPL', 'WFC', 'F', 'FB', 'DELL', 'SNE']
# Get_Yahoo_stats(tickers)

def forward_selected(data, response):
    """Linear model designed by forward selection.

    Parameters:
    -----------
    data : pandas DataFrame with all possible predictors and r
esponse

    response: string, name of response column in data

    Returns:
    --------
    model: an "optimal" fitted statsmodels linear model
           with an intercept
           selected by forward selection
           evaluated by adjusted R-squared
    """
```

```
    remaining = set(data.columns)

    remaining.remove(response)
    selected = []
    current_score, best_new_score = 0.0, 0.0
    while remaining and current_score == best_new_score:
        scores_with_candidates = []
        for candidate in remaining:
            formula = "{} ~ {} + 1".format(response, ' + '.joi
n(selected + [candidate]))

            score = smf.ols(formula, data).fit().rsquared_adj
            scores_with_candidates.append((score, candidate))
        scores_with_candidates.sort()

        best_new_score, best_candidate = scores_with_candidate
s.pop()
        if current_score < best_new_score:
            remaining.remove(best_candidate)
            selected.append(best_candidate)
            current_score = best_new_score
    formula = "{} ~ {} + 1".format(response,
                                   ' + '.join(selected))
    model = smf.ols(formula, data).fit()
    return model

# Examples

# data = sm.datasets.longley.load_pandas()
# df1=data.data
# formula = 'GNP ~ YEAR + UNEMP + POP + GNPDEFL'
# results = smf.ols(formula, df1).fit()
# print(results.summary())
# res = RF.forward_selected(df1, 'GNP')
# print(res.model.formula)
# print(res.rsquared_adj)
# print(res.summary())


def GQTest(lm2, level_of_sig=0.05, sp=None):
    name = ['F statistic', 'p-value']
    test = sms.het_goldfeldquandt(lm2.resid, lm2.model.exog, s
plit=sp)
    R=lzip(name, test)

    if test[1]>level_of_sig:
        print('The P vale of this test is {:3.5f}, which is gr
eater than the level of significance {} therefore, we accept t
he null that the error terms are homoscedastic'.format(test[1]
```

```
, level_of_sig))
    else:
        print('The P vale of this test is {:3.5f}, which is smaller than the level of significance {} therefore, we reject the null, hence the error terms are hetroscedastic'.format(test[1], level_of_sig))
    return R


def WhiteTest(statecrime_model, level_of_sig=0.05):
    white_test = het_white(statecrime_model.resid,  statecrime_model.model.exog)
    labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic', 'F-Test p-value']
    R=dict(zip(labels, white_test))

    if white_test[3]>level_of_sig:
        print('The P vale of this test is {:3.5f}, which is greater than the level of significance {} therefore, we accept the null that the error terms are homoscedastic'.format(white_test[3], level_of_sig))
    else:
        print('The P vale of this test is {:3.5f}, which is smaller than the level of significance {} therefore, we reject the null, hence the error terms are hetroscedastic'.format(white_test[3], level_of_sig))
    return R


def Plot_resi_corr(results):
    res_min_1=results.resid[:-1]
    res_plus_1=results.resid[1:]
    data1=pd.DataFrame(np.column_stack((res_min_1.T,res_plus_1.T)), columns=['u_t-1','u_t'])
    sns.set()
    plt.figure(figsize=(5,5))
    ax = sns.scatterplot(x='u_t-1', y='u_t', data=data1)
    pass

def Plot_resi_corr_time(results,df):
    C=pd.DataFrame(results.resid, index=df.index, columns=['Residuals'])
    C.plot(figsize=(10,5), linewidth=1.5, fontsize=10)
    plt.xlabel('Date', fontsize=10);
    return C


def Breusch_Godfrey(results, level_of_sig=0.05, lags=None):
```

```
    A=acorr_breusch_godfrey(results, nlags=lags)

    labels = ['LM Statistic', 'LM-Test p-value', 'F-Statistic'
, 'F-Test p-value']
    R=dict(zip(labels, A))

    if A[3]>level_of_sig:
        print('The P vale of this test is {:3.5f}, which is gr
eater than the level of significance {} therefore, we accept t
he null that the error terms are not Auto-corrolated'.format(A
[3], level_of_sig))
    else:
        print('The P vale of this test is {:3.5f}, which is sm
aller than the level of significance {} therefore, we reject t
he null, hence the error terms are Auto-corrolated'.format(A[3
], level_of_sig))
    return R

def Create_lags_of_variable(MainDF_first_period, lags, column)
:
    # Crete a new dataframe based on the lag variables
    x=column
    if type(lags) == int:
        j=lags
        values=MainDF_first_period[x]
        dataframe = pd.concat([values.shift(j), values], axis=
1)
        dataframe.columns = [x+' at time t-'+str(j), x+' at ti
me t']
        dataframe=dataframe.dropna()
    else:
        values=MainDF_first_period[x]
        dataframe=values
        for j in lags:
            dataframe = pd.concat([values.shift(j), dataframe]
, axis=1)
        c=[x+' for time t-'+str(j) for j in range(len(lags),-1
,-1)]
        dataframe.columns=c
        dataframe=dataframe.dropna()
    return dataframe

# # Things students shouldn't know
# MainDF1=MainDF.reset_index(drop=False)
# df2=pd.melt(MainDF1, id_vars=MainDF.index.name)
# palette = dict(zip(df2['variable'].unique(), sns.color_palet
te("rocket_r", len(df2['variable'].unique()))))
# sns.relplot(x=MainDF.index.name, y="value",
#               hue="variable", palette=palette,
```

```
#              height=5, aspect=3, facet_kws=dict(sharex=False)
, kind="line", data=df2)

# Call the important packages I want to use
import numpy as np, pandas as pd
import statistics as st
import statsmodels.api as sm
import RamiFunctions as RF, statsmodels.formula.api as smf
import statsmodels.tsa.api as smt
import statsmodels.stats.api as sms
import statsmodels.graphics.tsaplots as smgtsplot

import scipy.stats as ss
import scipy.stats.mstats as st1
from scipy import stats
import pandas_datareader.data as web
import datetime as dt
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import iqr
from statsmodels.graphics.gofplots import qqplot
from matplotlib import pyplot
import statsmodels.stats._adnorm

import warnings
warnings.filterwarnings('ignore')
```

In [ ]:

```

```

In [2]:

```
df =pd.read_excel('Data_For_Analysis.xlsx')
df.head(80)
```

Out[2]:

	Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DI
0	1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	
1	1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	
2	1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	

3	1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN
4	1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN
5	1976-06-01	NaN	NaN	NaN	NaN	NaN	NaN
6	1976-07-01	NaN	NaN	NaN	NaN	NaN	NaN
7	1976-08-01	NaN	NaN	NaN	NaN	NaN	NaN
8	1976-09-01	NaN	NaN	NaN	NaN	NaN	NaN
9	1976-10-01	NaN	NaN	NaN	NaN	NaN	NaN
10	1976-11-01	NaN	NaN	NaN	NaN	NaN	NaN
11	1976-12-01	NaN	NaN	NaN	NaN	NaN	NaN
12	1977-01-01	NaN	NaN	NaN	NaN	NaN	NaN
13	1977-02-01	NaN	NaN	NaN	NaN	NaN	NaN
14	1977-03-01	NaN	NaN	NaN	NaN	NaN	NaN
15	1977-04-01	NaN	NaN	NaN	NaN	NaN	NaN
16	1977-05-01	NaN	NaN	NaN	NaN	NaN	NaN
17	1977-06-01	NaN	NaN	NaN	NaN	NaN	NaN
18	1977-07-01	NaN	NaN	NaN	NaN	NaN	NaN
19	1977-08-01	NaN	NaN	NaN	NaN	NaN	NaN
20	1977-09-01	NaN	NaN	NaN	NaN	NaN	NaN
21	1977-10-01	NaN	NaN	NaN	NaN	NaN	NaN
22	1977-11-01	NaN	NaN	NaN	NaN	NaN	NaN
23	1977-	NaN	NaN	NaN	NaN	NaN	NaN

	12-01							
24	1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0
25	1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0
26	1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0
27	1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0
28	1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0
29	1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0
...	...	...	...	...	...	...	...	
50	1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0
51	1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0
52	1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0
53	1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0
54	1980-07-01	0.073	-0.023	-0.027	-0.034	0.212	0.183	0
55	1980-08-01	-0.045	0.029	-0.005	-0.018	0.058	0.081	-0
56	1980-09-01	0.019	-0.068	-0.010	0.034	-0.136	0.045	-0
57	1980-10-01	-0.054	-0.049	-0.021	0.035	0.007	-0.028	0
58	1980-11-01	0.028	0.123	-0.035	-0.017	0.000	0.056	0
59	1980-12-01	-0.047	0.131	0.131	0.103	-0.098	0.035	0
60	1981-01-01	0.011	-0.062	-0.015	0.040	-0.231	-0.089	0
61	1981-02-01	0.152	-0.005	-0.021	0.069	-0.072	0.006	0
62	1981-03-01	0.056	0.045	0.151	0.024	0.184	0.075	0

63	1981-04-01	0.045	0.086	0.061	-0.025	0.088	0.075	C
64	1981-05-01	0.032	0.099	0.017	0.117	0.112	0.107	C
65	1981-06-01	-0.037	-0.013	0.022	0.077	-0.178	-0.112	-C
66	1981-07-01	-0.065	-0.019	0.026	-0.092	0.007	-0.014	-C
67	1981-08-01	-0.125	-0.108	0.021	-0.030	-0.191	-0.065	-C
68	1981-09-01	-0.062	0.032	-0.013	0.003	0.089	-0.019	-C
69	1981-10-01	0.016	0.052	0.112	0.049	0.094	0.102	-C
70	1981-11-01	0.092	0.045	0.038	0.010	0.093	-0.065	-C
71	1981-12-01	-0.029	-0.028	-0.008	-0.106	-0.083	-0.060	-C
72	1982-01-01	-0.084	0.035	0.042	0.102	-0.002	0.027	C
73	1982-02-01	-0.159	0.000	0.036	-0.175	-0.152	-0.049	C
74	1982-03-01	0.108	0.007	0.022	-0.017	-0.302	-0.104	C
75	1982-04-01	-0.009	0.101	0.050	-0.013	0.047	0.054	-C
76	1982-05-01	-0.189	-0.101	0.016	-0.091	-0.180	-0.056	C
77	1982-06-01	-0.044	-0.003	-0.024	-0.096	-0.060	-0.073	C
78	1982-07-01	0.006	-0.025	-0.032	-0.303	-0.054	-0.055	-C
79	1982-08-01	0.379	0.077	0.133	0.070	0.216	0.273	C

In [3]:

```
df.set_index('Date', inplace=True)
```

In [4]:

```
df.head(300)
```

Out[4]:

Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-06-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-07-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-08-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-09-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-11-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-12-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-02-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-03-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-							

04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-06-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-07-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-08-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-09-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-11-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-12-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
...	...	...	...	...	...	...	...
1985-06-01	0.086	0.043	0.046	-0.050	0.060	-0.101	0.021
1985-07-01	-0.026	-0.030	-0.084	0.018	0.043	0.080	0.008
1985-08-01	0.011	-0.063	0.043	-0.052	-0.006	0.032	-0.066
1985-09-01	-0.095	-0.085	-0.032	0.036	0.000	0.036	-0.112

1985-10-01	-0.035	0.090	0.066	0.105	0.032	0.040	-0.083
1985-11-01	0.088	0.062	0.032	0.048	0.109	0.073	0.020
1985-12-01	0.064	0.065	0.082	0.197	0.023	0.095	0.030
1986-01-01	0.032	0.005	0.022	0.000	-0.055	0.162	0.122
1986-02-01	0.093	0.101	0.048	-0.051	-0.044	0.093	-0.055
1986-03-01	0.066	0.153	0.021	-0.040	-0.043	-0.063	0.076
1986-04-01	-0.013	-0.042	-0.006	-0.097	0.061	0.119	0.059
1986-05-01	0.072	0.038	0.042	-0.046	-0.015	0.037	-0.043
1986-06-01	-0.013	-0.036	0.017	-0.161	-0.155	-0.063	-0.070
1986-07-01	-0.060	-0.117	0.125	-0.038	-0.072	0.066	0.018
1986-08-01	0.115	0.082	0.061	-0.040	0.167	0.105	0.018
1986-09-01	-0.052	-0.111	-0.139	0.021	-0.240	-0.110	0.026
1986-10-01	0.059	0.040	0.045	0.000	0.105	0.103	0.134
1986-11-01	0.023	0.010	0.070	-0.143	0.020	0.048	-0.018
1986-12-01	-0.027	0.019	-0.046	0.028	-0.078	0.008	-0.010
1987-01-01	0.276	0.087	0.040	0.093	0.135	0.385	0.161
1987-02-01	-0.008	-0.066	-0.067	-0.064	0.045	0.056	0.133
1987-03-01	0.071	-0.052	-0.050	-0.087	-0.096	0.061	-0.129
1987-04-01	-0.037	0.070	0.020	-0.025	-0.020	0.055	-0.121
1987-	-0.111	0.052	-0.012	0.000	0.161	-0.082	0.151

05-01							
1987-06-01	0.063	0.051	0.059	0.081	-0.145	0.041	0.014
1987-07-01	0.064	0.041	-0.039	0.071	0.057	0.000	0.043
1987-08-01	0.061	0.033	0.043	-0.044	-0.008	0.157	-0.037
1987-09-01	-0.029	-0.086	-0.006	0.004	0.015	0.001	-0.067
1987-10-01	-0.274	-0.282	-0.017	-0.372	-0.342	-0.281	-0.260
1987-11-01	0.043	-0.136	-0.012	-0.148	-0.075	-0.127	-0.137

143 rows × 27 columns

In [5]:

```
df=df.dropna()
```

In [6]:

```
df[['BOISE','CONTIL','MARKET']].head(200)
```

Out[6]:

	BOISE	CONTIL	MARKET
Date			
1978-01-01	-0.079	-0.129	-0.045
1978-02-01	0.013	0.037	0.010
1978-03-01	0.070	0.003	0.050
1978-04-01	0.120	0.180	0.063
1978-05-01	0.071	0.061	0.067
1978-06-01	-0.098	-0.059	0.007
1978-07-01	0.140	0.066	0.071
1978-08-01	0.078	0.033	0.079
1978-09-01	-0.059	-0.013	0.002

1978-10-01	-0.118	-0.123	-0.189
1978-11-01	-0.060	-0.038	0.084
1978-12-01	0.067	0.047	0.015
1979-01-01	0.168	-0.024	0.058
1979-02-01	-0.032	-0.020	0.011
1979-03-01	0.178	0.043	0.123
1979-04-01	-0.043	0.064	0.026
1979-05-01	-0.026	0.005	0.014
1979-06-01	0.057	0.092	0.075
1979-07-01	0.047	-0.034	-0.013
1979-08-01	0.038	0.058	0.095
1979-09-01	0.050	-0.033	0.039
1979-10-01	-0.151	-0.136	-0.097
1979-11-01	-0.004	0.081	0.116
1979-12-01	0.042	0.104	0.086
1980-01-01	0.107	-0.103	0.124
1980-02-01	-0.070	-0.087	0.112
1980-03-01	-0.138	0.085	-0.243
1980-04-01	0.042	0.074	0.080
1980-05-01	0.109	0.023	0.062
1980-06-01	0.068	0.064	0.086
1980-07-01	0.073	-0.034	0.065
1980-08-01	-0.045	-0.018	0.025
1980-09-01	0.019	0.034	0.015
1980-10-01	-0.054	0.035	0.006
1980-11-01	0.028	-0.017	0.092
1980-12-01	-0.047	0.103	-0.056
1981-01-01	0.011	0.040	-0.014
1981-02-01	0.152	0.069	-0.009
1981-03-01	0.056	0.024	0.067
1981-04-01	0.045	-0.025	-0.008

1981-05-01	0.032	0.117	0.064
1981-06-01	-0.037	0.077	-0.003
1981-07-01	-0.065	-0.092	-0.033
1981-08-01	-0.125	-0.030	-0.031
1981-09-01	-0.062	0.003	-0.164

In [7]:

```
BOISE_prices=df['BOISE'].head(100)
BOISE_prices
```

Out[7]:

```
Date
1978-01-01   -0.079
1978-02-01    0.013
1978-03-01    0.070
1978-04-01    0.120
1978-05-01    0.071
1978-06-01   -0.098
1978-07-01    0.140
1978-08-01    0.078
1978-09-01   -0.059
1978-10-01   -0.118
1978-11-01   -0.060
1978-12-01    0.067
1979-01-01    0.168
1979-02-01   -0.032
1979-03-01    0.178
1979-04-01   -0.043
1979-05-01   -0.026
1979-06-01    0.057
1979-07-01    0.047
1979-08-01    0.038
1979-09-01    0.050
1979-10-01   -0.151
1979-11-01   -0.004
1979-12-01    0.042
1980-01-01    0.107
1980-02-01   -0.070
1980-03-01   -0.138
1980-04-01    0.042
1980-05-01    0.109
1980-06-01    0.068
1980-07-01    0.073
1980-08-01   -0.045
```

```
1980-09-01    0.019
1980-10-01   -0.054
1980-11-01    0.028
1980-12-01   -0.047
1981-01-01    0.011
1981-02-01    0.152
1981-03-01    0.056
1981-04-01    0.045
1981-05-01    0.032
1981-06-01   -0.037
1981-07-01   -0.065
1981-08-01   -0.125
1981-09-01   -0.062
Name: BOISE, dtype: float64
```

In [ ]:

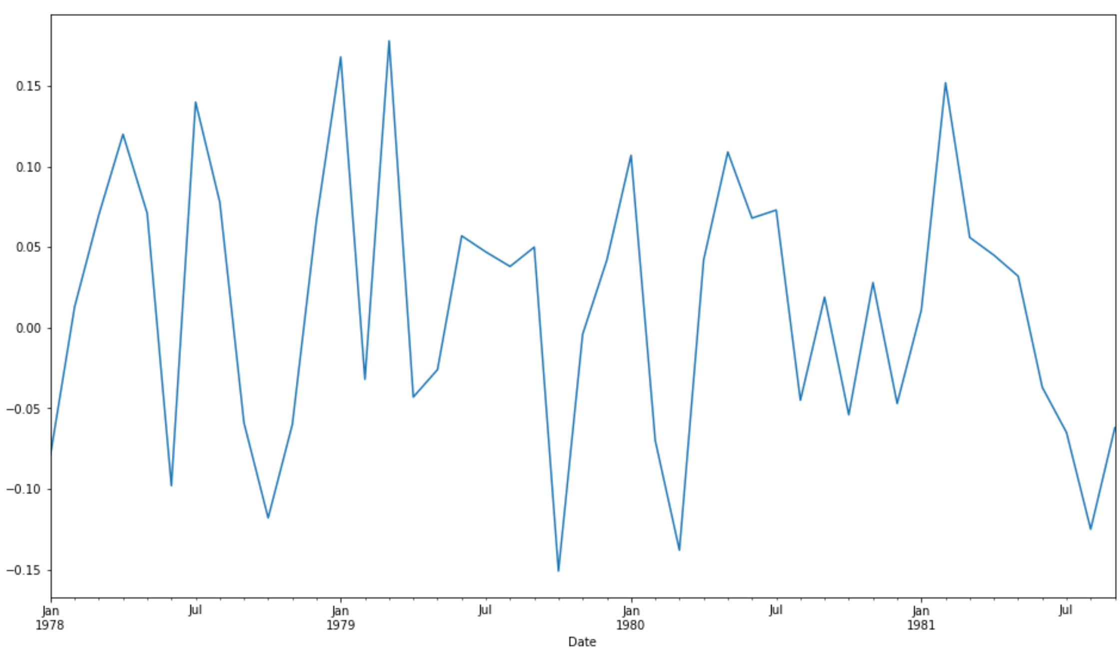
```

```

In [8]:

```
plt.rcParams["figure.figsize"]=[16,9]
BOISE_prices.plot()
plt.show
```

Out[8]:

```
<function matplotlib.pyplot.show(*args, **kw)>
```



In [9]:

```
CONTIL_prices=df['CONTIL']
CONTIL_prices
```

Out[9]:

```
Date
1978-01-01   -0.129
1978-02-01    0.037
1978-03-01    0.003
1978-04-01    0.180
1978-05-01    0.061
1978-06-01   -0.059
1978-07-01    0.066
1978-08-01    0.033
1978-09-01   -0.013
1978-10-01   -0.123
1978-11-01   -0.038
1978-12-01    0.047
1979-01-01   -0.024
1979-02-01   -0.020
1979-03-01    0.043
1979-04-01    0.064
1979-05-01    0.005
1979-06-01    0.092
1979-07-01   -0.034
1979-08-01    0.058
1979-09-01   -0.033
1979-10-01   -0.136
1979-11-01    0.081
1979-12-01    0.104
1980-01-01   -0.103
1980-02-01   -0.087
1980-03-01    0.085
1980-04-01    0.074
1980-05-01    0.023
1980-06-01    0.064
1980-07-01   -0.034
1980-08-01   -0.018
1980-09-01    0.034
1980-10-01    0.035
1980-11-01   -0.017
1980-12-01    0.103
1981-01-01    0.040
1981-02-01    0.069
1981-03-01    0.024
1981-04-01   -0.025
```

```
1981-05-01     0.117
1981-06-01     0.077
1981-07-01    -0.092
1981-08-01    -0.030
1981-09-01     0.003
Name: CONTIL, dtype: float64
```

In [ ]:

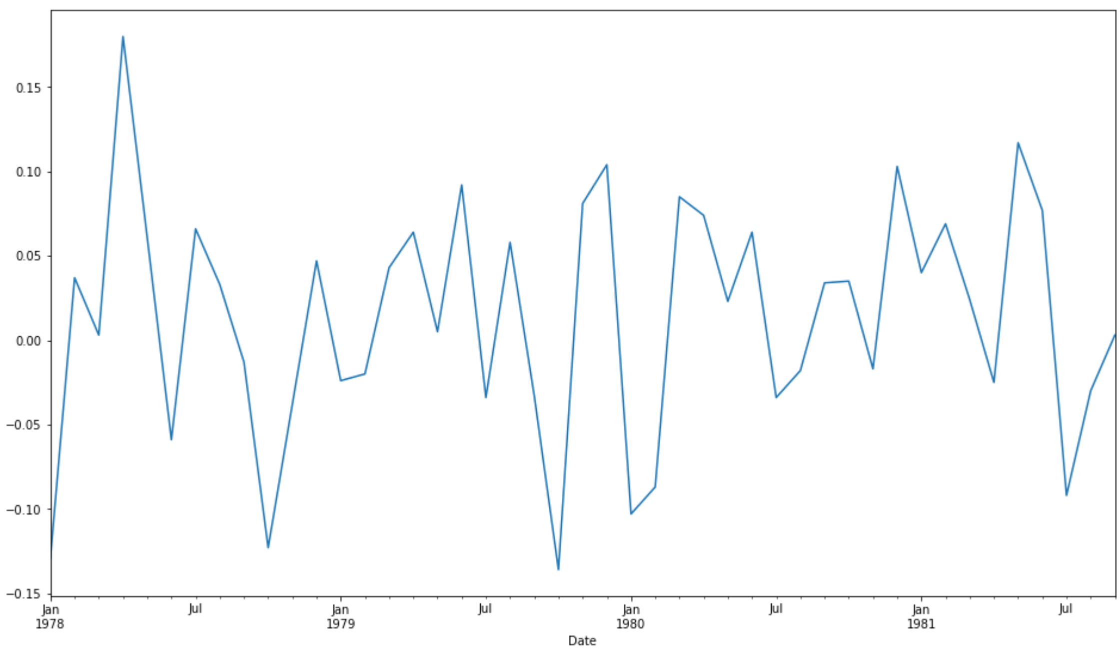
```

```

In [10]:

```
plt.rcParams["figure.figsize"]=[16,9]
CONTIL_prices.plot()
plt.show
```

Out[10]:

```
<function matplotlib.pyplot.show(*args, **kw)>
```



In [11]:

```
df.columns
```

Out[11]:

```
Index(['BOISE', 'CITCRP', 'CONED', 'CONTIL', 'DATG
EN', 'DEC', 'DELTA',
       'GENMIL', 'GERBER', 'IBM', 'MARKET', 'MOBIL
', 'MOTOR', 'PANAM', 'PSNH',
       'RKFREE', 'TANDY', 'TEXACO', 'WEYER', 'POIL
', 'FRBIND', 'CPI', 'GPU',
       'DOW', 'DUPONT', 'GOLD', 'CONOCO'],
      dtype='object')
```

In [12]:

```
df1=df[['BOISE','CONTIL','MARKET']].copy()
```

In [13]:

```
df1.head(200)
```

Out[13]:

	BOISE	CONTIL	MARKET
Date			
1978-01-01	-0.079	-0.129	-0.045
1978-02-01	0.013	0.037	0.010
1978-03-01	0.070	0.003	0.050
1978-04-01	0.120	0.180	0.063
1978-05-01	0.071	0.061	0.067
1978-06-01	-0.098	-0.059	0.007
1978-07-01	0.140	0.066	0.071
1978-08-01	0.078	0.033	0.079
1978-09-01	-0.059	-0.013	0.002
1978-10-01	-0.118	-0.123	-0.189
1978-11-01	-0.060	-0.038	0.084
1978-12-01	0.067	0.047	0.015
1979-01-01	0.168	-0.024	0.058

1979-01-01	[illegible]	0.024	[illegible]
1979-02-01	-0.032	-0.020	0.011
1979-03-01	0.178	0.043	0.123
1979-04-01	-0.043	0.064	0.026
1979-05-01	-0.026	0.005	0.014
1979-06-01	0.057	0.092	0.075
1979-07-01	0.047	-0.034	-0.013
1979-08-01	0.038	0.058	0.095
1979-09-01	0.050	-0.033	0.039
1979-10-01	-0.151	-0.136	-0.097
1979-11-01	-0.004	0.081	0.116
1979-12-01	0.042	0.104	0.086
1980-01-01	0.107	-0.103	0.124
1980-02-01	-0.070	-0.087	0.112
1980-03-01	-0.138	0.085	-0.243
1980-04-01	0.042	0.074	0.080
1980-05-01	0.109	0.023	0.062
1980-06-01	0.068	0.064	0.086
1980-07-01	0.073	-0.034	0.065
1980-08-01	-0.045	-0.018	0.025
1980-09-01	0.019	0.034	0.015
1980-10-01	-0.054	0.035	0.006
1980-11-01	0.028	-0.017	0.092
1980-12-01	-0.047	0.103	-0.056
1981-01-01	0.011	0.040	-0.014
1981-02-01	0.152	0.069	-0.009
1981-03-01	0.056	0.024	0.067
1981-04-01	0.045	-0.025	-0.008
1981-05-01	0.032	0.117	0.064
1981-06-01	-0.037	0.077	-0.003
1981-07-01	-0.065	-0.092	-0.033

1981-08-01	-0.125	-0.030	-0.031
1981-09-01	-0.062	0.003	-0.164

In [ ]:

```

```

In [14]:

```
df1.head(150)
```

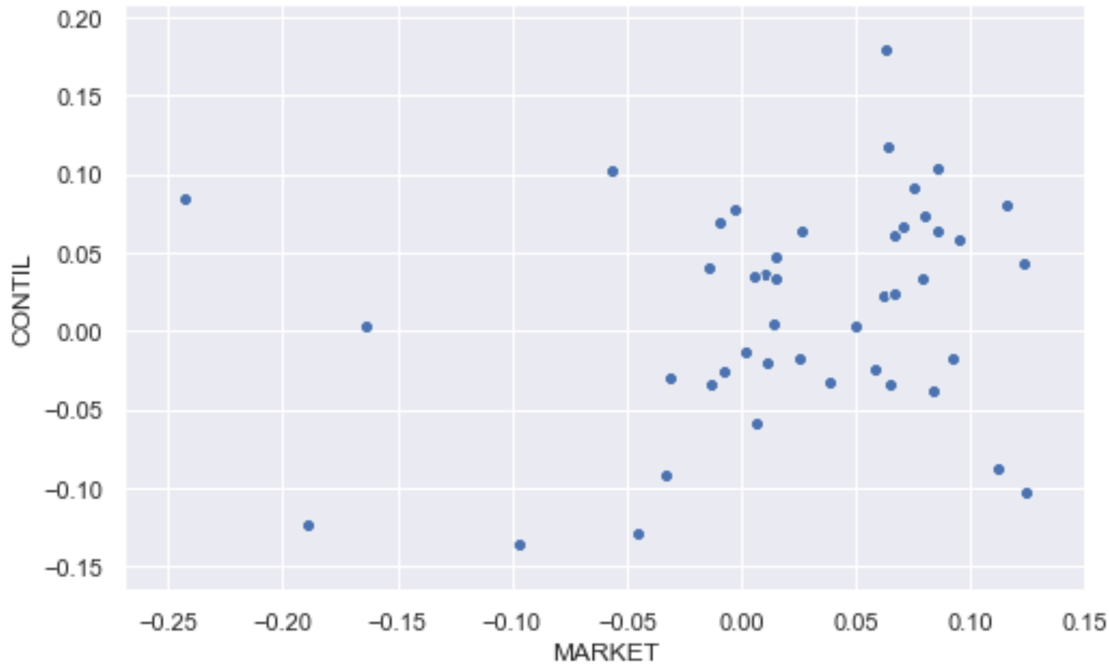
Out[14]:

	BOISE	CONTIL	MARKET
Date			
1978-01-01	-0.079	-0.129	-0.045
1978-02-01	0.013	0.037	0.010
1978-03-01	0.070	0.003	0.050
1978-04-01	0.120	0.180	0.063
1978-05-01	0.071	0.061	0.067
1978-06-01	-0.098	-0.059	0.007
1978-07-01	0.140	0.066	0.071
1978-08-01	0.078	0.033	0.079
1978-09-01	-0.059	-0.013	0.002
1978-10-01	-0.118	-0.123	-0.189
1978-11-01	-0.060	-0.038	0.084
1978-12-01	0.067	0.047	0.015
1979-01-01	0.168	-0.024	0.058
1979-02-01	-0.032	-0.020	0.011
1979-03-01	0.178	0.043	0.123
1979-04-01	-0.043	0.064	0.026
1979-05-01	-0.026	0.005	0.014
1979-06-01	0.057	0.092	0.075
1979-07-01	0.047	-0.034	-0.013

1979-08-01	0.038	0.058	0.095
1979-09-01	0.050	-0.033	0.039
1979-10-01	-0.151	-0.136	-0.097
1979-11-01	-0.004	0.081	0.116
1979-12-01	0.042	0.104	0.086
1980-01-01	0.107	-0.103	0.124
1980-02-01	-0.070	-0.087	0.112
1980-03-01	-0.138	0.085	-0.243
1980-04-01	0.042	0.074	0.080
1980-05-01	0.109	0.023	0.062
1980-06-01	0.068	0.064	0.086
1980-07-01	0.073	-0.034	0.065
1980-08-01	-0.045	-0.018	0.025
1980-09-01	0.019	0.034	0.015
1980-10-01	-0.054	0.035	0.006
1980-11-01	0.028	-0.017	0.092
1980-12-01	-0.047	0.103	-0.056
1981-01-01	0.011	0.040	-0.014
1981-02-01	0.152	0.069	-0.009
1981-03-01	0.056	0.024	0.067
1981-04-01	0.045	-0.025	-0.008
1981-05-01	0.032	0.117	0.064
1981-06-01	-0.037	0.077	-0.003
1981-07-01	-0.065	-0.092	-0.033
1981-08-01	-0.125	-0.030	-0.031
1981-09-01	-0.062	0.003	-0.164

In [15]:

```
sns.set()
plt.figure(figsize=(8,5))
ax = sns.scatterplot(x='MARKET', y='CONTIL', data=df)
```



In [16]:

```
df1.head(100)
```

Out[16]:

	BOISE	CONTIL	MARKET
Date			
1978-01-01	-0.079	-0.129	-0.045
1978-02-01	0.013	0.037	0.010
1978-03-01	0.070	0.003	0.050
1978-04-01	0.120	0.180	0.063
1978-05-01	0.071	0.061	0.067
1978-06-01	-0.098	-0.059	0.007
1978-07-01	0.140	0.066	0.071
1978-08-01	0.078	0.033	0.079
1978-09-01	-0.059	-0.013	0.002
1978-10-01	-0.118	-0.123	-0.189
1978-11-01	-0.060	-0.038	0.084

1978-11-01	-0.080	-0.038	0.084
1978-12-01	0.067	0.047	0.015
1979-01-01	0.168	-0.024	0.058
1979-02-01	-0.032	-0.020	0.011
1979-03-01	0.178	0.043	0.123
1979-04-01	-0.043	0.064	0.026
1979-05-01	-0.026	0.005	0.014
1979-06-01	0.057	0.092	0.075
1979-07-01	0.047	-0.034	-0.013
1979-08-01	0.038	0.058	0.095
1979-09-01	0.050	-0.033	0.039
1979-10-01	-0.151	-0.136	-0.097
1979-11-01	-0.004	0.081	0.116
1979-12-01	0.042	0.104	0.086
1980-01-01	0.107	-0.103	0.124
1980-02-01	-0.070	-0.087	0.112
1980-03-01	-0.138	0.085	-0.243
1980-04-01	0.042	0.074	0.080
1980-05-01	0.109	0.023	0.062
1980-06-01	0.068	0.064	0.086
1980-07-01	0.073	-0.034	0.065
1980-08-01	-0.045	-0.018	0.025
1980-09-01	0.019	0.034	0.015
1980-10-01	-0.054	0.035	0.006
1980-11-01	0.028	-0.017	0.092
1980-12-01	-0.047	0.103	-0.056
1981-01-01	0.011	0.040	-0.014
1981-02-01	0.152	0.069	-0.009
1981-03-01	0.056	0.024	0.067
1981-04-01	0.045	-0.025	-0.008
1981-05-01	0.032	0.117	0.064

1981-06-01	-0.037	0.077	-0.003
1981-07-01	-0.065	-0.092	-0.033
1981-08-01	-0.125	-0.030	-0.031
1981-09-01	-0.062	0.003	-0.164

In [17]:

```
df2=df1.copy()
df2=df2.sub(df.ix[:,-1],axis=0)
df2=df2.ix[:,:-1]
```

In [18]:

```
df2
```

Out[18]:

	BOISE	**CONTIL**
Date		
1978-01-01	0.03350	-0.01650
1978-02-01	-0.00015	0.02385
1978-03-01	0.06531	-0.00169
1978-04-01	0.06393	0.12393
1978-05-01	0.02321	0.01321
1978-06-01	0.00029	0.03929
1978-07-01	0.15422	0.08022
1978-08-01	-0.01719	-0.06219
1978-09-01	-0.10789	-0.06189
1978-10-01	0.01336	0.00836
1978-11-01	-0.08927	-0.06727
1978-12-01	-0.01473	-0.03473
1979-01-01	0.10133	-0.09067
1979-02-01	-0.05700	-0.04500
1979-03-01	0.05043	-0.08457
1979-04-01	-0.04300	0.06400

1979-05-01	-0.03111	-0.00011
1979-06-01	-0.05697	-0.02197
1979-07-01	0.02720	-0.05380
1979-08-01	-0.00860	0.01140
1979-09-01	-0.04375	-0.12675
1979-10-01	-0.11671	-0.10171
1979-11-01	-0.07441	0.01059
1979-12-01	-0.01387	0.04813
1980-01-01	-0.01734	-0.22734
1980-02-01	-0.08600	-0.10300
1980-03-01	0.02789	0.25089
1980-04-01	-0.01402	0.01798
1980-05-01	0.05754	-0.02846
1980-06-01	-0.00598	-0.00998
1980-07-01	0.05162	-0.05538
1980-08-01	-0.02267	0.00433
1980-09-01	-0.03388	-0.01888
1980-10-01	-0.13482	-0.04582
1980-11-01	-0.21081	-0.25581
1980-12-01	0.05283	0.20283
1981-01-01	0.09322	0.12222
1981-02-01	0.15992	0.07692
1981-03-01	0.09422	0.06222
1981-04-01	0.17083	0.10083
1981-05-01	-0.02406	0.06094
1981-06-01	-0.30577	-0.19177
1981-07-01	-0.45813	-0.48513
1981-08-01	-0.03897	0.05603
1981-09-01	0.16156	0.22656

In [19]:

```
f=['BOISE','CONTIL']
```

In [20]:

```
f
```

Out[20]:

```
['BOISE', 'CONTIL']
```

In [21]:

```
df2.columns=f
```

In [22]:

```
df2.head(100)
```

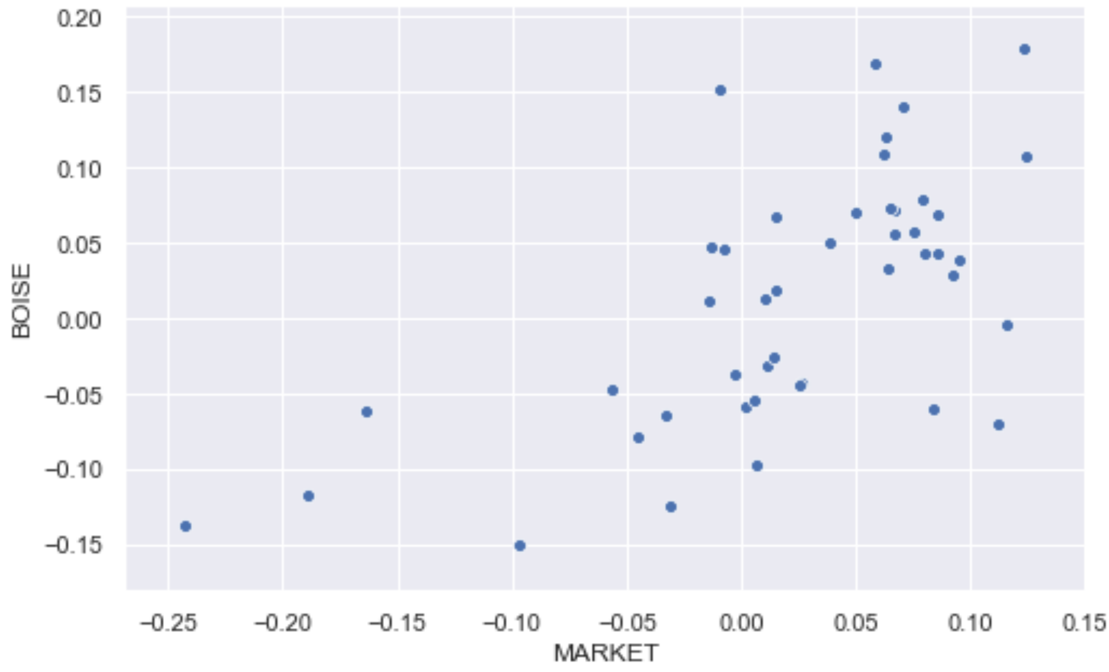
Out[22]:

	BOISE	CONTIL
Date		
1978-01-01	0.03350	-0.01650
1978-02-01	-0.00015	0.02385
1978-03-01	0.06531	-0.00169
1978-04-01	0.06393	0.12393
1978-05-01	0.02321	0.01321
1978-06-01	0.00029	0.03929
1978-07-01	0.15422	0.08022
1978-08-01	-0.01719	-0.06219
1978-09-01	-0.10789	-0.06189
1978-10-01	0.01336	0.00836
1978-11-01	-0.08927	-0.06727
1978-12-01	-0.01473	-0.03473
1979-01-01	0.10133	-0.09067
1979-02-01	-0.05700	-0.04500

1979-03-01	0.05043	-0.08457
1979-04-01	-0.04300	0.06400
1979-05-01	-0.03111	-0.00011
1979-06-01	-0.05697	-0.02197
1979-07-01	0.02720	-0.05380
1979-08-01	-0.00860	0.01140
1979-09-01	-0.04375	-0.12675
1979-10-01	-0.11671	-0.10171
1979-11-01	-0.07441	0.01059
1979-12-01	-0.01387	0.04813
1980-01-01	-0.01734	-0.22734
1980-02-01	-0.08600	-0.10300
1980-03-01	0.02789	0.25089
1980-04-01	-0.01402	0.01798
1980-05-01	0.05754	-0.02846
1980-06-01	-0.00598	-0.00998
1980-07-01	0.05162	-0.05538
1980-08-01	-0.02267	0.00433
1980-09-01	-0.03388	-0.01888
1980-10-01	-0.13482	-0.04582
1980-11-01	-0.21081	-0.25581
1980-12-01	0.05283	0.20283
1981-01-01	0.09322	0.12222
1981-02-01	0.15992	0.07692
1981-03-01	0.09422	0.06222
1981-04-01	0.17083	0.10083
1981-05-01	-0.02406	0.06094
1981-06-01	-0.30577	-0.19177
1981-07-01	-0.45813	-0.48513
1981-08-01	-0.03897	0.05603
1981-09-01	0.16156	0.22656

In [23]:

```
sns.set()
plt.figure(figsize=(8,5))
ax = sns.scatterplot(x='MARKET', y='BOISE', data=df)
```



In [24]:

```
df1.index
```

Out[24]:

```
DatetimeIndex(['1978-01-01', '1978-02-01', '1978-0
3-01', '1978-04-01',
               '1978-05-01', '1978-06-01', '1978-0
7-01', '1978-08-01',
               '1978-09-01', '1978-10-01', '1978-1
1-01', '1978-12-01',
               '1979-01-01', '1979-02-01', '1979-0
3-01', '1979-04-01',
               '1979-05-01', '1979-06-01', '1979-0
7-01', '1979-08-01',
               '1979-09-01', '1979-10-01', '1979-1
1-01', '1979-12-01',
               '1980-01-01', '1980-02-01', '1980-0
3-01', '1980-04-01',
               '1980-05-01', '1980-06-01', '1980-0
7-01', '1980-08-01',
               '1980-09-01', '1980-10-01', '1980-1
1-01', '1980-12-01',
               '1981-01-01', '1981-02-01', '1981-0
3-01', '1981-04-01',
               '1981-05-01', '1981-06-01', '1981-0
7-01', '1981-08-01',
               '1981-09-01'],
              dtype='datetime64[ns]', name='Date',
freq=None)
```

In [25]:

```
len(df1)
```

Out[25]:

```
45
```

In [26]:

```
df3=df1.groupby(np.arange(len(df1))//(3*3))
```

In [27]:

```
np.arange(len(df1))//(3*3)
```

Out[27]:

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4
, 4, 4, 4, 4, 4, 4, 4,
       4])
```

In [28]:

```
df1['Group column']=np.arange(len(df1))//(3*3)
```

In [29]:

```
df1.head(200)
```

Out[29]:

	BOISE	CONTIL	MARKET	Group column
Date				
1978-01-01	-0.079	-0.129	-0.045	0
1978-02-01	0.013	0.037	0.010	0
1978-03-01	0.070	0.003	0.050	0
1978-04-01	0.120	0.180	0.063	0
1978-05-01	0.071	0.061	0.067	0
1978-06-01	-0.098	-0.059	0.007	0
1978-07-01	0.140	0.066	0.071	0
1978-08-01	0.078	0.033	0.079	0
1978-09-01	-0.059	-0.013	0.002	0
1978-10-01	-0.118	-0.123	-0.189	1
1978-11-01	-0.060	-0.038	0.084	1
1978-12-01	0.067	0.047	0.015	1
1979-01-01	0.168	-0.024	0.058	1
1979-02-01	-0.032	-0.020	0.011	1

1979-03-01	0.178	0.043	0.123	1
1979-04-01	-0.043	0.064	0.026	1
1979-05-01	-0.026	0.005	0.014	1
1979-06-01	0.057	0.092	0.075	1
1979-07-01	0.047	-0.034	-0.013	2
1979-08-01	0.038	0.058	0.095	2
1979-09-01	0.050	-0.033	0.039	2
1979-10-01	-0.151	-0.136	-0.097	2
1979-11-01	-0.004	0.081	0.116	2
1979-12-01	0.042	0.104	0.086	2
1980-01-01	0.107	-0.103	0.124	2
1980-02-01	-0.070	-0.087	0.112	2
1980-03-01	-0.138	0.085	-0.243	2
1980-04-01	0.042	0.074	0.080	3
1980-05-01	0.109	0.023	0.062	3
1980-06-01	0.068	0.064	0.086	3
1980-07-01	0.073	-0.034	0.065	3
1980-08-01	-0.045	-0.018	0.025	3
1980-09-01	0.019	0.034	0.015	3
1980-10-01	-0.054	0.035	0.006	3
1980-11-01	0.028	-0.017	0.092	3
1980-12-01	-0.047	0.103	-0.056	3
1981-01-01	0.011	0.040	-0.014	4
1981-02-01	0.152	0.069	-0.009	4
1981-03-01	0.056	0.024	0.067	4
1981-04-01	0.045	-0.025	-0.008	4
1981-05-01	0.032	0.117	0.064	4
1981-06-01	-0.037	0.077	-0.003	4
1981-07-01	-0.065	-0.092	-0.033	4
1981-08-01	-0.125	-0.030	-0.031	4
1981-09-01	-0.062	0.003	-0.164	4

In [30]:

```
1986-1978
```

Out[30]:

8

In [31]:

```
df_group_0=df3.get_group(0)
df_group_1=df3.get_group(1)
df_group_2=df3.get_group(2)
df_group_3=df3.get_group(3)
```

In [32]:

```
df_group_0
df.dropna()
```

Out[32]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
Date							
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
1978-07-01	0.140	0.032	0.011	0.066	0.143	0.107	0.185
1978-08-01	0.078	0.088	0.024	0.033	0.026	-0.017	-0.021
1978-09-01	-0.059	0.011	0.048	-0.013	-0.031	-0.037	-0.081

1978-10-01	-0.118	-0.071	-0.067	-0.123	-0.085	-0.077	-0.153
1978-11-01	-0.060	-0.005	0.035	-0.038	0.044	0.064	0.055
1978-12-01	0.067	-0.019	0.005	0.047	0.034	0.117	-0.023
1979-01-01	0.168	0.043	0.076	-0.024	-0.008	-0.012	-0.054
1979-02-01	-0.032	-0.082	-0.011	-0.020	-0.015	-0.066	-0.060
1979-03-01	0.178	0.026	0.000	0.043	0.171	0.088	0.098
1979-04-01	-0.043	0.000	-0.057	0.064	0.009	0.005	-0.056
1979-05-01	-0.026	0.022	0.032	0.005	-0.045	-0.028	0.063
1979-06-01	0.057	0.095	0.066	0.092	0.019	0.059	-0.006
1979-07-01	0.047	-0.075	0.015	-0.034	-0.059	0.009	0.075
1979-08-01	0.038	0.065	-0.021	0.058	0.078	0.140	0.021
1979-09-01	0.050	-0.017	0.000	-0.033	-0.031	-0.027	-0.026
1979-10-01	-0.151	-0.125	-0.049	-0.136	-0.246	-0.010	-0.147
1979-11-01	-0.004	0.030	0.109	0.081	0.062	0.095	0.063
1979-12-01	0.042	0.113	0.005	0.104	0.021	0.018	0.020
1980-01-01	0.107	-0.079	-0.039	-0.103	0.157	0.058	0.022
1980-02-01	-0.070	-0.080	-0.061	-0.087	0.043	0.034	-0.093
1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0.031
1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0.018
1980-							

05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0.144
1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0.010
1980-07-01	0.073	-0.023	-0.027	-0.034	0.212	0.183	0.283
1980-08-01	-0.045	0.029	-0.005	-0.018	0.058	0.081	-0.056
1980-09-01	0.019	-0.068	-0.010	0.034	-0.136	0.045	-0.053
1980-10-01	-0.054	-0.049	-0.021	0.035	0.007	-0.028	0.046
1980-11-01	0.028	0.123	-0.035	-0.017	0.000	0.056	0.220
1980-12-01	-0.047	0.131	0.131	0.103	-0.098	0.035	0.040
1981-01-01	0.011	-0.062	-0.015	0.040	-0.231	-0.089	0.112
1981-02-01	0.152	-0.005	-0.021	0.069	-0.072	0.006	0.031
1981-03-01	0.056	0.045	0.151	0.024	0.184	0.075	0.024
1981-04-01	0.045	0.086	0.061	-0.025	0.088	0.075	0.062
1981-05-01	0.032	0.099	0.017	0.117	0.112	0.107	0.105
1981-06-01	-0.037	-0.013	0.022	0.077	-0.178	-0.112	-0.114
1981-07-01	-0.065	-0.019	0.026	-0.092	0.007	-0.014	-0.094
1981-08-01	-0.125	-0.108	0.021	-0.030	-0.191	-0.065	-0.072
1981-09-01	-0.062	0.032	-0.013	0.003	0.089	-0.019	-0.013

In [33]:

```
df2=df1.copy()
df2=df2.sub(df.ix[:,-1],axis=0)
df2=df2.ix[:,:-1]
df2.dropna()
```

Out[33]:

	BOISE	CONTIL	MARKET
Date			
1978-01-01	0.03350	-0.01650	0.06750
1978-02-01	-0.00015	0.02385	-0.00315
1978-03-01	0.06531	-0.00169	0.04531
1978-04-01	0.06393	0.12393	0.00693
1978-05-01	0.02321	0.01321	0.01921
1978-06-01	0.00029	0.03929	0.10529
1978-07-01	0.15422	0.08022	0.08522
1978-08-01	-0.01719	-0.06219	-0.01619
1978-09-01	-0.10789	-0.06189	-0.04689
1978-10-01	0.01336	0.00836	-0.05764
1978-11-01	-0.08927	-0.06727	0.05473
1978-12-01	-0.01473	-0.03473	-0.06673
1979-01-01	0.10133	-0.09067	-0.00867
1979-02-01	-0.05700	-0.04500	-0.01400
1979-03-01	0.05043	-0.08457	-0.00457
1979-04-01	-0.04300	0.06400	0.02600
1979-05-01	-0.03111	-0.00011	0.00889
1979-06-01	-0.05697	-0.02197	-0.03897
1979-07-01	0.02720	-0.05380	-0.03280
1979-08-01	-0.00860	0.01140	0.04840
1979-09-01	-0.04375	-0.12675	-0.05475
1979-10-01	-0.11671	-0.10171	-0.06271

1979-11-01	-0.07441	0.01059	0.04559
1979-12-01	-0.01387	0.04813	0.03013
1980-01-01	-0.01734	-0.22734	-0.00034
1980-02-01	-0.08600	-0.10300	0.09600
1980-03-01	0.02789	0.25089	-0.07711
1980-04-01	-0.01402	0.01798	0.02398
1980-05-01	0.05754	-0.02846	0.01054
1980-06-01	-0.00598	-0.00998	0.01202
1980-07-01	0.05162	-0.05538	0.04362
1980-08-01	-0.02267	0.00433	0.04733
1980-09-01	-0.03388	-0.01888	-0.03788
1980-10-01	-0.13482	-0.04582	-0.07482
1980-11-01	-0.21081	-0.25581	-0.14681
1980-12-01	0.05283	0.20283	0.04383
1981-01-01	0.09322	0.12222	0.06822
1981-02-01	0.15992	0.07692	-0.00108
1981-03-01	0.09422	0.06222	0.10522
1981-04-01	0.17083	0.10083	0.11783
1981-05-01	-0.02406	0.06094	0.00794
1981-06-01	-0.30577	-0.19177	-0.27177
1981-07-01	-0.45813	-0.48513	-0.42613
1981-08-01	-0.03897	0.05603	0.05503
1981-09-01	0.16156	0.22656	0.05956

In [34]:

```
f=['BOISE','CONTIL','MARKET']
```

In [35]:

```
f
```

Out[35]:

```
['BOISE', 'CONTIL', 'MARKET']
```

In [36]:

```
df2.columns=f
```

In [37]:

```
df2.head(100)
```

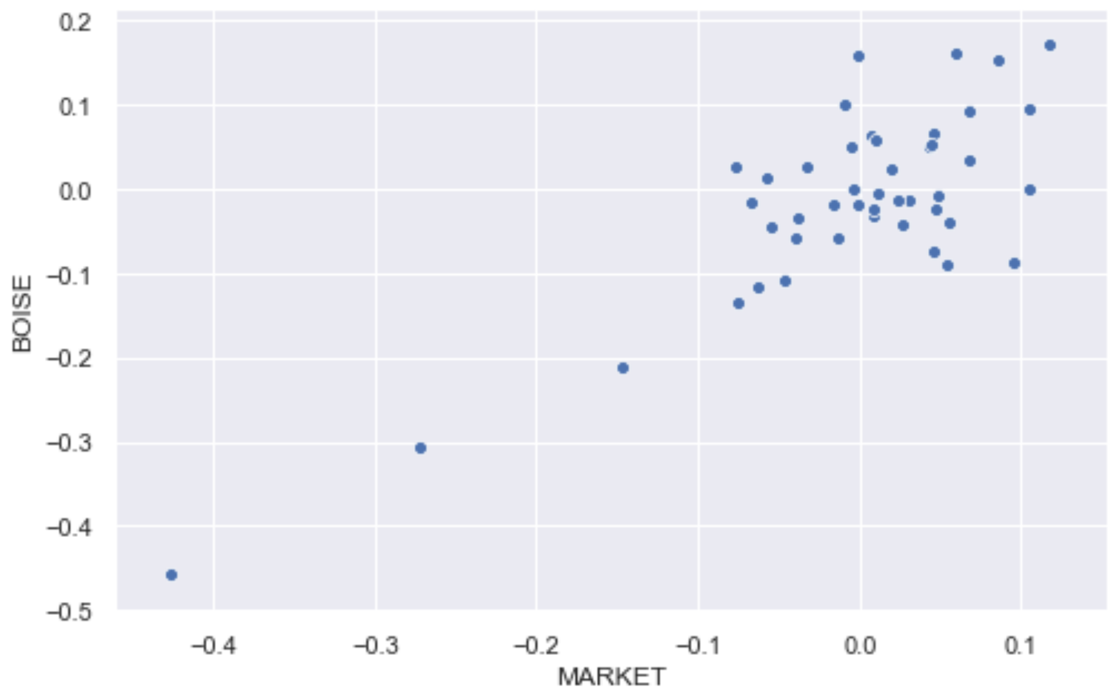
Out[37]:

Date	BOISE	CONTIL	MARKET
1978-01-01	0.03350	-0.01650	0.06750
1978-02-01	-0.00015	0.02385	-0.00315
1978-03-01	0.06531	-0.00169	0.04531
1978-04-01	0.06393	0.12393	0.00693
1978-05-01	0.02321	0.01321	0.01921
1978-06-01	0.00029	0.03929	0.10529
1978-07-01	0.15422	0.08022	0.08522
1978-08-01	-0.01719	-0.06219	-0.01619
1978-09-01	-0.10789	-0.06189	-0.04689
1978-10-01	0.01336	0.00836	-0.05764
1978-11-01	-0.08927	-0.06727	0.05473
1978-12-01	-0.01473	-0.03473	-0.06673
1979-01-01	0.10133	-0.09067	-0.00867
1979-02-01	-0.05700	-0.04500	-0.01400
1979-03-01	0.05043	-0.08457	-0.00457
1979-04-01	-0.04300	0.06400	0.02600
1979-05-01	-0.03111	-0.00011	0.00889

1979-06-01	-0.05697	-0.02197	-0.03897
1979-07-01	0.02720	-0.05380	-0.03280
1979-08-01	-0.00860	0.01140	0.04840
1979-09-01	-0.04375	-0.12675	-0.05475
1979-10-01	-0.11671	-0.10171	-0.06271
1979-11-01	-0.07441	0.01059	0.04559
1979-12-01	-0.01387	0.04813	0.03013
1980-01-01	-0.01734	-0.22734	-0.00034
1980-02-01	-0.08600	-0.10300	0.09600
1980-03-01	0.02789	0.25089	-0.07711
1980-04-01	-0.01402	0.01798	0.02398
1980-05-01	0.05754	-0.02846	0.01054
1980-06-01	-0.00598	-0.00998	0.01202
1980-07-01	0.05162	-0.05538	0.04362
1980-08-01	-0.02267	0.00433	0.04733
1980-09-01	-0.03388	-0.01888	-0.03788
1980-10-01	-0.13482	-0.04582	-0.07482
1980-11-01	-0.21081	-0.25581	-0.14681
1980-12-01	0.05283	0.20283	0.04383
1981-01-01	0.09322	0.12222	0.06822
1981-02-01	0.15992	0.07692	-0.00108
1981-03-01	0.09422	0.06222	0.10522
1981-04-01	0.17083	0.10083	0.11783
1981-05-01	-0.02406	0.06094	0.00794
1981-06-01	-0.30577	-0.19177	-0.27177
1981-07-01	-0.45813	-0.48513	-0.42613
1981-08-01	-0.03897	0.05603	0.05503
1981-09-01	0.16156	0.22656	0.05956

In [38]:

```
sns.set()
plt.figure(figsize=(8,5))
ax = sns.scatterplot(x='MARKET', y='BOISE', data=df2)
```



In [39]:

```
start_date = dt.datetime(1979,3,1)
end_date = dt.datetime(1987,2,1)
#greater than the start date and smaller than the end date
select = (df2.index>=start_date)*(df2.index<=end_date)

# Copy the selected dataframe into df2
df3=df2[select].copy()
```

In [40]:

```
df3.head(200)
```

Out[40]:

	BOISE	CONTIL	MARKET
Date			
1979-03-01	0.05043	-0.08457	-0.00457
1979-04-01	-0.04300	0.06400	0.02600
1979-05-01	-0.03111	-0.00011	0.00889

1979-06-01	-0.05697	-0.02197	-0.03897
1979-07-01	0.02720	-0.05380	-0.03280
1979-08-01	-0.00860	0.01140	0.04840
1979-09-01	-0.04375	-0.12675	-0.05475
1979-10-01	-0.11671	-0.10171	-0.06271
1979-11-01	-0.07441	0.01059	0.04559
1979-12-01	-0.01387	0.04813	0.03013
1980-01-01	-0.01734	-0.22734	-0.00034
1980-02-01	-0.08600	-0.10300	0.09600
1980-03-01	0.02789	0.25089	-0.07711
1980-04-01	-0.01402	0.01798	0.02398
1980-05-01	0.05754	-0.02846	0.01054
1980-06-01	-0.00598	-0.00998	0.01202
1980-07-01	0.05162	-0.05538	0.04362
1980-08-01	-0.02267	0.00433	0.04733
1980-09-01	-0.03388	-0.01888	-0.03788
1980-10-01	-0.13482	-0.04582	-0.07482
1980-11-01	-0.21081	-0.25581	-0.14681
1980-12-01	0.05283	0.20283	0.04383
1981-01-01	0.09322	0.12222	0.06822
1981-02-01	0.15992	0.07692	-0.00108
1981-03-01	0.09422	0.06222	0.10522
1981-04-01	0.17083	0.10083	0.11783
1981-05-01	-0.02406	0.06094	0.00794
1981-06-01	-0.30577	-0.19177	-0.27177
1981-07-01	-0.45813	-0.48513	-0.42613
1981-08-01	-0.03897	0.05603	0.05503
1981-09-01	0.16156	0.22656	0.05956

In [41]:

```
df4=df3[['CONTIL','MARKET']].copy()
df4.head(100)
```

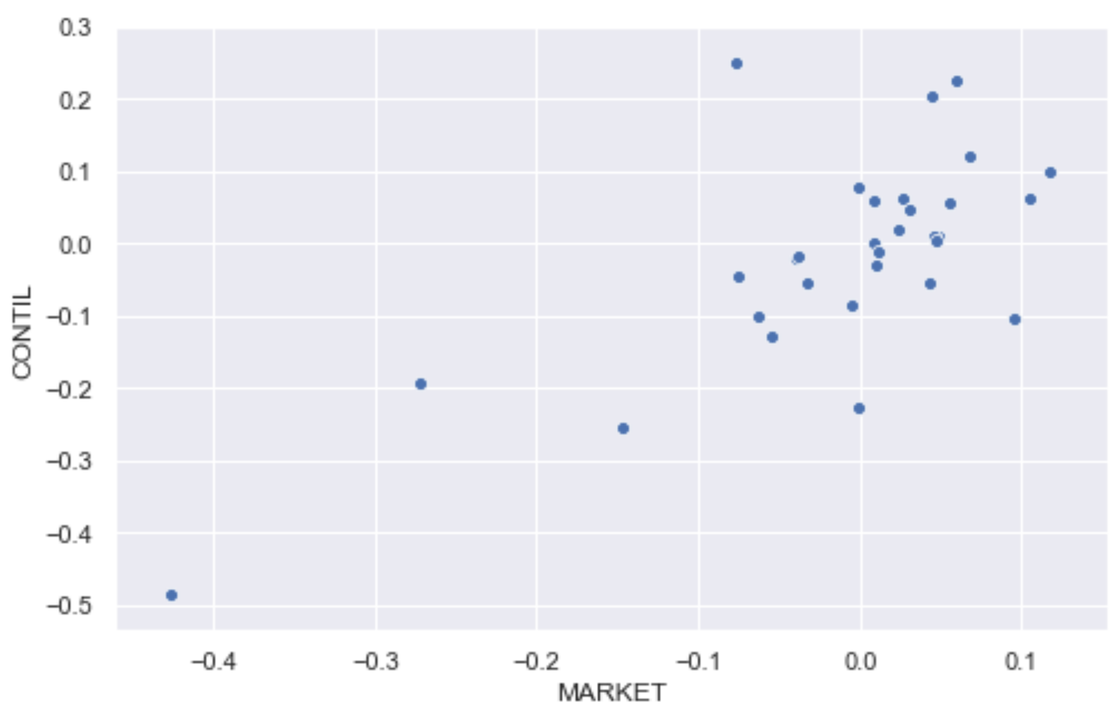
Out[41]:

	CONTIL	MARKET
Date		
1979-03-01	-0.08457	-0.00457
1979-04-01	0.06400	0.02600
1979-05-01	-0.00011	0.00889
1979-06-01	-0.02197	-0.03897
1979-07-01	-0.05380	-0.03280
1979-08-01	0.01140	0.04840
1979-09-01	-0.12675	-0.05475
1979-10-01	-0.10171	-0.06271
1979-11-01	0.01059	0.04559
1979-12-01	0.04813	0.03013
1980-01-01	-0.22734	-0.00034
1980-02-01	-0.10300	0.09600
1980-03-01	0.25089	-0.07711
1980-04-01	0.01798	0.02398
1980-05-01	-0.02846	0.01054
1980-06-01	-0.00998	0.01202
1980-07-01	-0.05538	0.04362
1980-08-01	0.00433	0.04733
1980-09-01	-0.01888	-0.03788
1980-10-01	-0.04582	-0.07482
1980-11-01	-0.25581	-0.14681
1980-12-01	0.20283	0.04383
1981-01-01	0.12222	0.06822
1981-02-01	0.07692	-0.00108

1981-03-01	0.06222	0.10522
1981-04-01	0.10083	0.11783
1981-05-01	0.06094	0.00794
1981-06-01	-0.19177	-0.27177
1981-07-01	-0.48513	-0.42613
1981-08-01	0.05603	0.05503
1981-09-01	0.22656	0.05956

In [42]:

```
sns.set()
plt.figure(figsize=(8,5))
ax = sns.scatterplot(x='MARKET', y='CONTIL', data=df3)
```



In [43]:

```
choice_of_columns=['CONTIL', 'BOISE']
df6=Joining_columns(df3, 'MARKET', choice_of_columns, Name_of_
new_column='Stock Return')
df6.head(100)
```

Out[43]:

	MARKET	variable	Stock Return
0	-0.00457	CONTIL	-0.08457
1	0.02600	CONTIL	0.06400

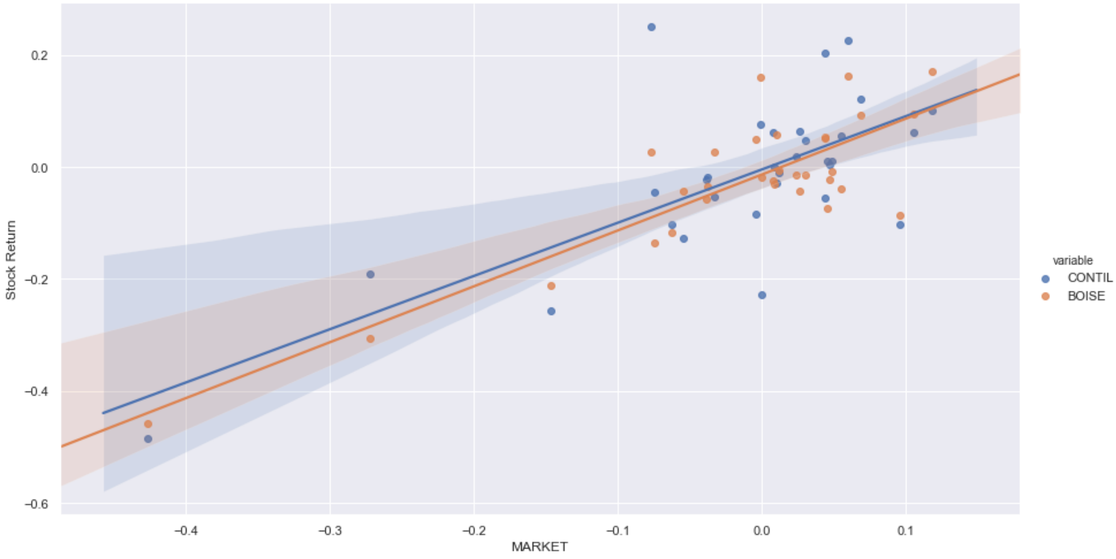
2	0.00889	CONTIL	-0.00011
3	-0.03897	CONTIL	-0.02197
4	-0.03280	CONTIL	-0.05380
5	0.04840	CONTIL	0.01140
6	-0.05475	CONTIL	-0.12675
7	-0.06271	CONTIL	-0.10171
8	0.04559	CONTIL	0.01059
9	0.03013	CONTIL	0.04813
10	-0.00034	CONTIL	-0.22734
11	0.09600	CONTIL	-0.10300
12	-0.07711	CONTIL	0.25089
13	0.02398	CONTIL	0.01798
14	0.01054	CONTIL	-0.02846
15	0.01202	CONTIL	-0.00998
16	0.04362	CONTIL	-0.05538
17	0.04733	CONTIL	0.00433
18	-0.03788	CONTIL	-0.01888
19	-0.07482	CONTIL	-0.04582
20	-0.14681	CONTIL	-0.25581
21	0.04383	CONTIL	0.20283
22	0.06822	CONTIL	0.12222
23	-0.00108	CONTIL	0.07692
24	0.10522	CONTIL	0.06222
25	0.11783	CONTIL	0.10083
26	0.00794	CONTIL	0.06094
27	-0.27177	CONTIL	-0.19177
28	-0.42613	CONTIL	-0.48513
29	0.05503	CONTIL	0.05603
...	...	...	...
32	0.02600	BOISE	-0.04300

33	0.00889	BOISE	-0.03111
34	-0.03897	BOISE	-0.05697
35	-0.03280	BOISE	0.02720
36	0.04840	BOISE	-0.00860
37	-0.05475	BOISE	-0.04375
38	-0.06271	BOISE	-0.11671
39	0.04559	BOISE	-0.07441
40	0.03013	BOISE	-0.01387
41	-0.00034	BOISE	-0.01734
42	0.09600	BOISE	-0.08600
43	-0.07711	BOISE	0.02789
44	0.02398	BOISE	-0.01402
45	0.01054	BOISE	0.05754
46	0.01202	BOISE	-0.00598
47	0.04362	BOISE	0.05162
48	0.04733	BOISE	-0.02267
49	-0.03788	BOISE	-0.03388
50	-0.07482	BOISE	-0.13482
51	-0.14681	BOISE	-0.21081
52	0.04383	BOISE	0.05283
53	0.06822	BOISE	0.09322
54	-0.00108	BOISE	0.15992
55	0.10522	BOISE	0.09422
56	0.11783	BOISE	0.17083
57	0.00794	BOISE	-0.02406
58	-0.27177	BOISE	-0.30577
59	-0.42613	BOISE	-0.45813
60	0.05503	BOISE	-0.03897
61	0.05956	BOISE	0.16156

62 rows × 3 columns

In [44]:

```
ig=sns.lmplot(x="MARKET", y="Stock Return", hue="variable", da
ta=df6, height=7, aspect=1.8/1)
```



In [45]:

```
import statsmodels.formula.api as smf
```

In [46]:

```
formula = 'Q("CONTIL") ~ Q("MARKET")'
results = smf.ols(formula, df3).fit()
print(results.summary())
```

```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:          Q("CONTIL")   R-square
d:                        0.500
Model:                          OLS   Adj. R-s
quared:                   0.483
Method:               Least Squares   F-statis
tic:                      29.04
Date:              Wed, 18 Dec 2019   Prob (F-
statistic):            8.60e-06
Time:                      14:38:32   Log-Like
lihood:                  26.925
No. Observations:                31   AIC:
-49.85
Df Residuals:                    29   BIC:
-46.98
Df Model:                         1
Covariance Type:          nonrobust
================================================
===============================
                  coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
-------------------------------
Intercept      -0.0043      0.019     -0.227
0.822      -0.043       0.035
Q("MARKET")     0.9517      0.177      5.389
0.000       0.590       1.313
================================================
==============================
Omnibus:                      9.455   Durbin-W
atson:                    2.064
Prob(Omnibus):                0.009   Jarque-B
era (JB):                10.072
Skew:                         0.777   Prob(JB)
:                       0.00650
Kurtosis:                     5.320   Cond. No
.                          9.37
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
atrix of the errors is correctly specified.
```

In [47]:

```
formula = 'Q("BOISE") ~ Q("MARKET")'
results = smf.ols(formula, df3).fit()
print(results.summary())
```

```
                            OLS Regression Results
==================================================
============================
Dep. Variable:              Q("BOISE")   R-squared:
0.711
Model:                             OLS   Adj. R-squ
ared:                    0.701
Method:                  Least Squares   F-statisti
c:                       71.25
Date:                 Wed, 18 Dec 2019   Prob (F-st
atistic):             2.66e-09
Time:                         14:38:32   Log-Likeli
hood:                   39.332
No. Observations:                   31   AIC:
-74.66
Df Residuals:                       29   BIC:
-71.80
Df Model:                            1
Covariance Type:             nonrobust
==================================================
=============================
                  coef    std err          t
P>|t|      [0.025      0.975]
--------------------------------------------------
-----------------------------
Intercept      -0.0134      0.013     -1.051
0.302      -0.039       0.013
Q("MARKET")     0.9989      0.118      8.441
0.000       0.757       1.241
==================================================
============================
Omnibus:                         1.931   Durbin-Wat
son:                     2.237
Prob(Omnibus):                   0.381   Jarque-Ber
a (JB):                  0.823
Skew:                            0.254   Prob(JB):
0.662
Kurtosis:                        3.616   Cond. No.
9.37
==================================================
============================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
```

In [48]:

```
df = pd.read_excel('Data_For_Analysis.xlsx')
df.set_index('Date', inplace=True)
df.tail(100)
```

Out[48]:

Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
1979-08-01	0.038	0.065	-0.021	0.058	0.078	0.140	0.021
1979-09-01	0.050	-0.017	0.000	-0.033	-0.031	-0.027	-0.026
1979-10-01	-0.151	-0.125	-0.049	-0.136	-0.246	-0.010	-0.147
1979-11-01	-0.004	0.030	0.109	0.081	0.062	0.095	0.063
1979-12-01	0.042	0.113	0.005	0.104	0.021	0.018	0.020
1980-01-01	0.107	-0.079	-0.039	-0.103	0.157	0.058	0.022
1980-02-01	-0.070	-0.080	-0.061	-0.087	0.043	0.034	-0.093
1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0.031
1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0.018
1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0.144
1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0.010
1980-07-01	0.073	-0.023	-0.027	-0.034	0.212	0.183	0.283
1980-08-01	-0.045	0.029	-0.005	-0.018	0.058	0.081	-0.056
1980-09-01	0.019	-0.068	-0.010	0.034	-0.136	0.045	-0.053
1980-10-01	-0.054	-0.049	-0.021	0.035	0.007	-0.028	0.046

1980-11-01	0.028	0.123	-0.035	-0.017	0.000	0.056	0.220
1980-12-01	-0.047	0.131	0.131	0.103	-0.098	0.035	0.040
1981-01-01	0.011	-0.062	-0.015	0.040	-0.231	-0.089	0.112
1981-02-01	0.152	-0.005	-0.021	0.069	-0.072	0.006	0.031
1981-03-01	0.056	0.045	0.151	0.024	0.184	0.075	0.024
1981-04-01	0.045	0.086	0.061	-0.025	0.088	0.075	0.062
1981-05-01	0.032	0.099	0.017	0.117	0.112	0.107	0.105
1981-06-01	-0.037	-0.013	0.022	0.077	-0.178	-0.112	-0.114
1981-07-01	-0.065	-0.019	0.026	-0.092	0.007	-0.014	-0.094
1981-08-01	-0.125	-0.108	0.021	-0.030	-0.191	-0.065	-0.072
1981-09-01	-0.062	0.032	-0.013	0.003	0.089	-0.019	-0.013
1981-10-01	0.016	0.052	0.112	0.049	0.094	0.102	-0.072
1981-11-01	0.092	0.045	0.038	0.010	0.093	-0.065	-0.032
1981-12-01	-0.029	-0.028	-0.008	-0.106	-0.083	-0.060	-0.062
1982-01-01	-0.084	0.035	0.042	0.102	-0.002	0.027	0.056
...	...	...	...	...	...	...	...
1985-06-01	0.086	0.043	0.046	-0.050	0.060	-0.101	0.021
1985-07-01	-0.026	-0.030	-0.084	0.018	0.043	0.080	0.008
1985-08-01	0.011	-0.063	0.043	-0.052	-0.006	0.032	-0.066
1985-09-01	-0.095	-0.085	-0.032	0.036	0.000	0.036	-0.112

1985-10-01	-0.035	0.090	0.066	0.105	0.032	0.040	-0.083
1985-11-01	0.088	0.062	0.032	0.048	0.109	0.073	0.020
1985-12-01	0.064	0.065	0.082	0.197	0.023	0.095	0.030
1986-01-01	0.032	0.005	0.022	0.000	-0.055	0.162	0.122
1986-02-01	0.093	0.101	0.048	-0.051	-0.044	0.093	-0.055
1986-03-01	0.066	0.153	0.021	-0.040	-0.043	-0.063	0.076
1986-04-01	-0.013	-0.042	-0.006	-0.097	0.061	0.119	0.059
1986-05-01	0.072	0.038	0.042	-0.046	-0.015	0.037	-0.043
1986-06-01	-0.013	-0.036	0.017	-0.161	-0.155	-0.063	-0.070
1986-07-01	-0.060	-0.117	0.125	-0.038	-0.072	0.066	0.018
1986-08-01	0.115	0.082	0.061	-0.040	0.167	0.105	0.018
1986-09-01	-0.052	-0.111	-0.139	0.021	-0.240	-0.110	0.026
1986-10-01	0.059	0.040	0.045	0.000	0.105	0.103	0.134
1986-11-01	0.023	0.010	0.070	-0.143	0.020	0.048	-0.018
1986-12-01	-0.027	0.019	-0.046	0.028	-0.078	0.008	-0.010
1987-01-01	0.276	0.087	0.040	0.093	0.135	0.385	0.161
1987-02-01	-0.008	-0.066	-0.067	-0.064	0.045	0.056	0.133
1987-03-01	0.071	-0.052	-0.050	-0.087	-0.096	0.061	-0.129
1987-04-01	-0.037	0.070	0.020	-0.025	-0.020	0.055	-0.121
1987-							

05-01	-0.111	0.052	-0.012	0.000	0.161	-0.082	0.151
1987-06-01	0.063	0.051	0.059	0.081	-0.145	0.041	0.014
1987-07-01	0.064	0.041	-0.039	0.071	0.057	0.000	0.043
1987-08-01	0.061	0.033	0.043	-0.044	-0.008	0.157	-0.037
1987-09-01	-0.029	-0.086	-0.006	0.004	0.015	0.001	-0.067
1987-10-01	-0.274	-0.282	-0.017	-0.372	-0.342	-0.281	-0.260
1987-11-01	0.043	-0.136	-0.012	-0.148	-0.075	-0.127	-0.137

100 rows × 27 columns

In [49]:

```
hypotheses = 'Q("MARKET")=0'
Simple_ttest_Ols(results, hypotheses, alternative='larger', le
vel_of_sig = 0.05)
```

```
We reject the null hypothesis: Q("MARKET")=0 with
a 5.0 % significance level
```

In [50]:

```
hypotheses = 'Q("MARKET")=1'
Simple_ttest_Ols(results, hypotheses, alternative='larger', le
vel_of_sig = 0.05)
```

```
We accept the null hypothesis: Q("MARKET")=1 with
a 5.0 % significance level
```

In [51]:

```
hypotheses = 'Q("MARKET")=1'
Simple_ttest_Ols(results, hypotheses, alternative='larger', le
vel_of_sig = 0.05)
```

```
We accept the null hypothesis: Q("MARKET")=1 with
a 5.0 % significance level
```

In [52]:

```
formula = 'CONTIL ~ MARKET'
results = smf.ols(formula, df1).fit()
std_residuals_CONTIL=results.resid
print(std_residuals_CONTIL.std())
```

```
0.06839421589721249
```

In [53]:

```
formula = 'BOISE ~ MARKET'
results = smf.ols(formula, df1).fit()
std_residuals_BOISE=results.resid
print(std_residuals_BOISE.std())
```

```
0.06435324835690541
```

In [54]:

```
df1.head(200)
```

Out[54]:

	BOISE	CONTIL	MARKET	Group column
Date				
1978-01-01	-0.079	-0.129	-0.045	0
1978-02-01	0.013	0.037	0.010	0
1978-03-01	0.070	0.003	0.050	0
1978-04-01	0.120	0.180	0.063	0
1978-05-01	0.071	0.061	0.067	0
1978-06-01	-0.098	-0.059	0.007	0
1978-07-01	0.140	0.066	0.071	0
1978-08-01	0.078	0.033	0.079	0
1978-09-01	-0.059	-0.013	0.002	0
1978-10-01	-0.118	-0.123	-0.189	1
1978-11-01	-0.060	-0.038	0.084	1
1978-12-01	0.067	0.047	0.015	1
1979-01-01	0.168	-0.024	0.058	1

1979-02-01	-0.032	-0.020	0.011	1
1979-03-01	0.178	0.043	0.123	1
1979-04-01	-0.043	0.064	0.026	1
1979-05-01	-0.026	0.005	0.014	1
1979-06-01	0.057	0.092	0.075	1
1979-07-01	0.047	-0.034	-0.013	2
1979-08-01	0.038	0.058	0.095	2
1979-09-01	0.050	-0.033	0.039	2
1979-10-01	-0.151	-0.136	-0.097	2
1979-11-01	-0.004	0.081	0.116	2
1979-12-01	0.042	0.104	0.086	2
1980-01-01	0.107	-0.103	0.124	2
1980-02-01	-0.070	-0.087	0.112	2
1980-03-01	-0.138	0.085	-0.243	2
1980-04-01	0.042	0.074	0.080	3
1980-05-01	0.109	0.023	0.062	3
1980-06-01	0.068	0.064	0.086	3
1980-07-01	0.073	-0.034	0.065	3
1980-08-01	-0.045	-0.018	0.025	3
1980-09-01	0.019	0.034	0.015	3
1980-10-01	-0.054	0.035	0.006	3
1980-11-01	0.028	-0.017	0.092	3
1980-12-01	-0.047	0.103	-0.056	3
1981-01-01	0.011	0.040	-0.014	4
1981-02-01	0.152	0.069	-0.009	4
1981-03-01	0.056	0.024	0.067	4
1981-04-01	0.045	-0.025	-0.008	4
1981-05-01	0.032	0.117	0.064	4
1981-06-01	-0.037	0.077	-0.003	4
1981-07-01	-0.065	-0.092	-0.033	4
1981-08-01	-0.125	-0.030	-0.031	4

1981-09-01	-0.062	0.003	-0.164	4

In [55]:

```
df1.head(200)
df.dropna()
```

Out[55]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
Date							
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
1978-07-01	0.140	0.032	0.011	0.066	0.143	0.107	0.185
1978-08-01	0.078	0.088	0.024	0.033	0.026	-0.017	-0.021
1978-09-01	-0.059	0.011	0.048	-0.013	-0.031	-0.037	-0.081
1978-10-01	-0.118	-0.071	-0.067	-0.123	-0.085	-0.077	-0.153
1978-11-01	-0.060	-0.005	0.035	-0.038	0.044	0.064	0.055
1978-12-01	0.067	-0.019	0.005	0.047	0.034	0.117	-0.023
1979-01-01	0.168	0.043	0.076	-0.024	-0.008	-0.012	-0.054
1979-02-01	-0.032	-0.082	-0.011	-0.020	-0.015	-0.066	-0.060

1979-03-01	0.178	0.026	0.000	0.043	0.171	0.088	0.098
1979-04-01	-0.043	0.000	-0.057	0.064	0.009	0.005	-0.056
1979-05-01	-0.026	0.022	0.032	0.005	-0.045	-0.028	0.063
1979-06-01	0.057	0.095	0.066	0.092	0.019	0.059	-0.006
1979-07-01	0.047	-0.075	0.015	-0.034	-0.059	0.009	0.075
1979-08-01	0.038	0.065	-0.021	0.058	0.078	0.140	0.021
1979-09-01	0.050	-0.017	0.000	-0.033	-0.031	-0.027	-0.026
1979-10-01	-0.151	-0.125	-0.049	-0.136	-0.246	-0.010	-0.147
1979-11-01	-0.004	0.030	0.109	0.081	0.062	0.095	0.063
1979-12-01	0.042	0.113	0.005	0.104	0.021	0.018	0.020
1980-01-01	0.107	-0.079	-0.039	-0.103	0.157	0.058	0.022
1980-02-01	-0.070	-0.080	-0.061	-0.087	0.043	0.034	-0.093
1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0.031
1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0.018
1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0.144
1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0.010
1980-07-01	0.073	-0.023	-0.027	-0.034	0.212	0.183	0.283
1980-08-01	-0.045	0.029	-0.005	-0.018	0.058	0.081	-0.056
1980-09-01	0.019	-0.068	-0.010	0.034	-0.136	0.045	-0.053
1980-							

10-01	-0.054	-0.049	-0.021	0.035	0.007	-0.028	0.046
1980-11-01	0.028	0.123	-0.035	-0.017	0.000	0.056	0.220
1980-12-01	-0.047	0.131	0.131	0.103	-0.098	0.035	0.040
1981-01-01	0.011	-0.062	-0.015	0.040	-0.231	-0.089	0.112
1981-02-01	0.152	-0.005	-0.021	0.069	-0.072	0.006	0.031
1981-03-01	0.056	0.045	0.151	0.024	0.184	0.075	0.024
1981-04-01	0.045	0.086	0.061	-0.025	0.088	0.075	0.062
1981-05-01	0.032	0.099	0.017	0.117	0.112	0.107	0.105
1981-06-01	-0.037	-0.013	0.022	0.077	-0.178	-0.112	-0.114
1981-07-01	-0.065	-0.019	0.026	-0.092	0.007	-0.014	-0.094
1981-08-01	-0.125	-0.108	0.021	-0.030	-0.191	-0.065	-0.072
1981-09-01	-0.062	0.032	-0.013	0.003	0.089	-0.019	-0.013

In [56]:

```
formula_with_intercept = 'MARKET ~ BOISE + CONTIL'

results_with_intercept = smf.ols(formula_with_intercept, df1).
fit()
print(results_with_intercept.summary())
```

```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:                 MARKET   R-square
d:                       0.400
Model:                            OLS   Adj. R-s
quared:                  0.372
Method:                 Least Squares   F-statis
tic:                     14.03
Date:                Wed, 18 Dec 2019   Prob (F-
statistic):           2.16e-05
Time:                        14:38:45   Log-Like
lihood:                 62.669
No. Observations:                  45   AIC:
-119.3
Df Residuals:                      42   BIC:
-113.9
Df Model:                           2
Covariance Type:            nonrobust
================================================
==============================
                 coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
------------------------------
Intercept      0.0146      0.009      1.543
0.130      -0.005       0.034
BOISE          0.6088      0.124      4.927
0.000       0.359       0.858
CONTIL        -0.0327      0.146     -0.224
0.824      -0.327       0.262
================================================
==============================
Omnibus:                        6.784   Durbin-W
atson:                   2.511
Prob(Omnibus):                  0.034   Jarque-B
era (JB):                5.819
Skew:                          -0.662   Prob(JB)
:                       0.0545
Kurtosis:                       4.161   Cond. No
.                         17.5
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
```

In [57]:

```
R=GQTest(results_with_intercept)
```

The P vale of this test is 0.01935, which is small
er than the level of significance 0.05 therefore,
we reject the null, hence the error terms are hetr
oscedastic

In [58]:

```
formula_without_intercept = 'MARKET ~ BOISE + CONTIL '

results_without_intercept = smf.ols(formula_without_intercept,
df1).fit()
print(results_without_intercept.summary())
```

```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:                 MARKET   R-square
d:                       0.400
Model:                            OLS   Adj. R-s
quared:                  0.372
Method:                 Least Squares   F-statis
tic:                     14.03
Date:                Wed, 18 Dec 2019   Prob (F-
statistic):           2.16e-05
Time:                        14:38:45   Log-Like
lihood:                 62.669
No. Observations:                  45   AIC:
-119.3
Df Residuals:                      42   BIC:
-113.9
Df Model:                           2
Covariance Type:            nonrobust
================================================
==============================
                 coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
------------------------------
Intercept      0.0146      0.009      1.543
0.130      -0.005       0.034
BOISE          0.6088      0.124      4.927
0.000       0.359       0.858
CONTIL        -0.0327      0.146     -0.224
0.824      -0.327       0.262
================================================
==============================
Omnibus:                        6.784   Durbin-W
atson:                   2.511
Prob(Omnibus):                  0.034   Jarque-B
era (JB):                5.819
Skew:                          -0.662   Prob(JB)
:                       0.0545
Kurtosis:                       4.161   Cond. No
.                         17.5
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
atrix of the errors is correctly specified.
```

In [59]:

```
R=GQTest(results_without_intercept)
```

```
The P vale of this test is 0.01935, which is small
er than the level of significance 0.05 therefore,
we reject the null, hence the error terms are hetr
oscedastic
```

In [60]:

```
R=WhiteTest(results_with_intercept)
```

```
The P vale of this test is 0.02118, which is small
er than the level of significance 0.05 therefore,
we reject the null, hence the error terms are hetr
oscedastic
```

In [61]:

```
resid_model1=results_with_intercept.resid
resid_model1
```

Out[61]:

```
Date
1978-01-01   -0.015748
1978-02-01   -0.011327
1978-03-01   -0.007143
1978-04-01   -0.018791
1978-05-01    0.011147
1978-06-01    0.050110
1978-07-01   -0.026699
1978-08-01    0.017968
1978-09-01    0.022872
1978-10-01   -0.135808
1978-11-01    0.104662
1978-12-01   -0.038876
1979-01-01   -0.059692
1979-02-01    0.015204
1979-03-01    0.001413
1979-04-01    0.039651
1979-05-01    0.015370
1979-06-01    0.028685
1979-07-01   -0.057351
1979-08-01    0.059140
1979-09-01   -0.007145
1979-10-01   -0.024142
1979-11-01    0.106463
```

```
1979-11-01    0.106463
1979-12-01    0.049210
1980-01-01    0.040861
1980-02-01    0.137147
1980-03-01   -0.170823
1980-04-01    0.042228
1980-05-01   -0.018233
1980-06-01    0.032071
1980-07-01    0.004819
1980-08-01    0.037185
1980-09-01   -0.010078
1980-10-01    0.025399
1980-11-01    0.059773
1980-12-01   -0.038637
1981-01-01   -0.034011
1981-02-01   -0.113907
1981-03-01    0.019068
1981-04-01   -0.050839
1981-05-01    0.033724
1981-06-01    0.007423
1981-07-01   -0.011061
1981-08-01    0.029498
1981-09-01   -0.140778
dtype: float64
```

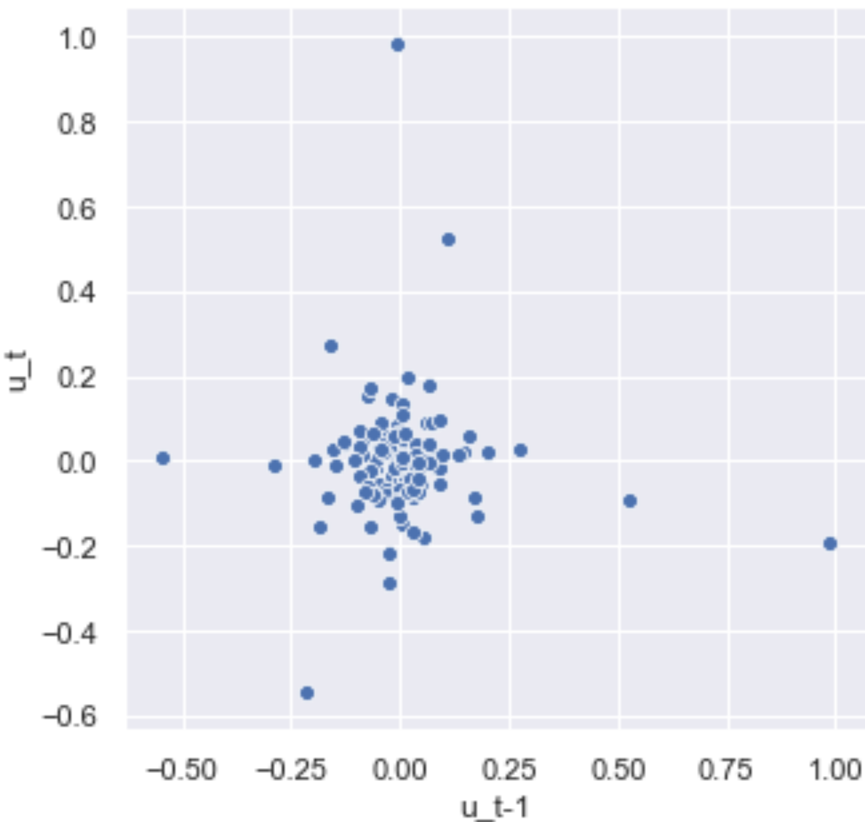
In [62]:

```
formula = 'Q("CONTIL") ~ Q("MARKET")'

results = smf.ols(formula, df).fit()
print(results.summary())
```

```
                            OLS Regression Results
==================================================
============================
Dep. Variable:          Q("CONTIL")   R-squared:
0.111
Model:                          OLS   Adj. R-squ
ared:                  0.104
Method:               Least Squares   F-statisti
c:                     14.65
Date:              Wed, 18 Dec 2019   Prob (F-st
atistic):           0.000209
Time:                      14:38:46   Log-Likeli
hood:                 63.430
No. Observations:               119   AIC:
-122.9
Df Residuals:                   117   BIC:
-117.3
Df Model:                         1
Covariance Type:          nonrobust
==================================================
=============================
                  coef    std err          t
P>|t|      [0.025      0.975]
--------------------------------------------------
-----------------------------
Intercept      -0.0115      0.013     -0.858
0.393      -0.038       0.015
Q("MARKET")     0.7375      0.193      3.828
0.000       0.356       1.119
==================================================
============================
Omnibus:                    106.122   Durbin-Wat
son:                   2.068
Prob(Omnibus):                0.000   Jarque-Ber
a (JB):             2224.184
Skew:                         2.697   Prob(JB):
0.00
Kurtosis:                    23.481   Cond. No.
14.7
==================================================
============================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
```

In [63]:

```
Plot_resi_corr(results)
```



In [64]:

```
Plot_resi_corr_time(results,df)
```

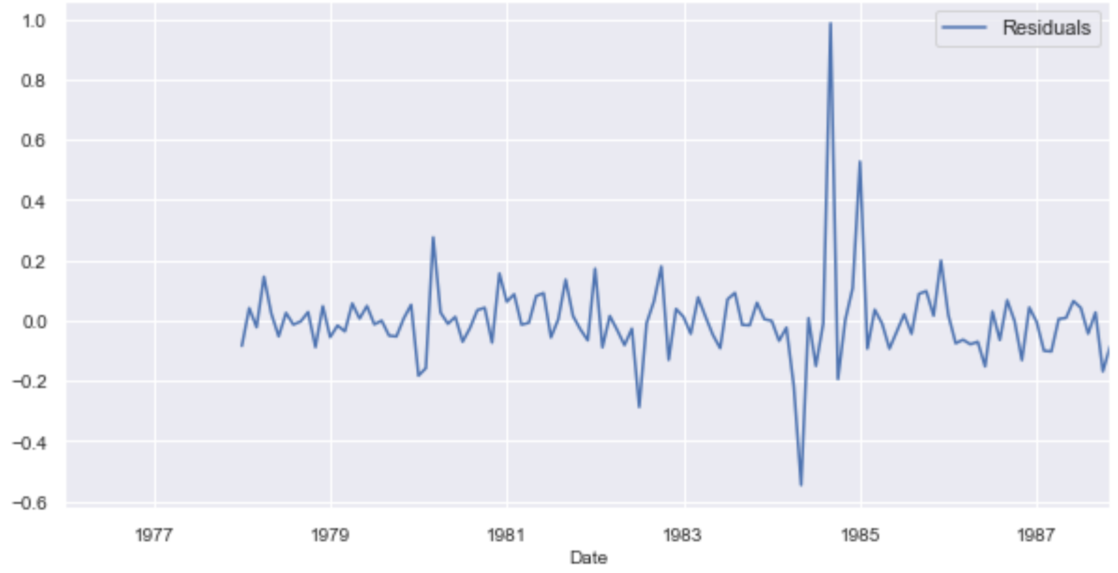
Out[64]:

	Residuals
Date	
1976-01-01	NaN
1976-02-01	NaN
1976-03-01	NaN
1976-04-01	NaN
1976-05-01	NaN
1976-06-01	NaN
1976-07-01	NaN
1976-08-01	NaN
1976-09-01	NaN
1976-10-01	NaN

1976-10-01	NaN
1976-11-01	NaN
1976-12-01	NaN
1977-01-01	NaN
1977-02-01	NaN
1977-03-01	NaN
1977-04-01	NaN
1977-05-01	NaN
1977-06-01	NaN
1977-07-01	NaN
1977-08-01	NaN
1977-09-01	NaN
1977-10-01	NaN
1977-11-01	NaN
1977-12-01	NaN
1978-01-01	-0.084327
1978-02-01	0.041108
1978-03-01	-0.022394
1978-04-01	0.145018
1978-05-01	0.023068
1978-06-01	-0.052680
...	...
1985-06-01	-0.036304
1985-07-01	0.020633
1985-08-01	-0.044205
1985-09-01	0.088048
1985-10-01	0.097307
1985-11-01	0.015968
1985-12-01	0.198895
1986-01-01	0.018121
1986-02-01	-0.075657

1986-03-01	-0.063919
1986-04-01	-0.078879
1986-05-01	-0.070657
1986-06-01	-0.152467
1986-07-01	0.029537
1986-08-01	-0.064657
1986-09-01	0.067148
1986-10-01	-0.001793
1986-11-01	-0.131517
1986-12-01	0.043171
1987-01-01	-0.004674
1987-02-01	-0.100457
1987-03-01	-0.102806
1987-04-01	0.004922
1987-05-01	0.008533
1987-06-01	0.064456
1987-07-01	0.041918
1987-08-01	-0.043580
1987-09-01	0.026546
1987-10-01	-0.168755
1987-11-01	-0.084889

143 rows × 1 columns



In [65]:

```
import statsmodels.stats.stattools as tools
tools.durbin_watson(results.resid)
```

Out[65]:

```
2.06843613615565
```

In [66]:

```
Breusch_Godfrey(results, lags=6, level_of_sig=0.1)
```

```
The P vale of this test is 0.81024, which is great
er than the level of significance 0.1 therefore, w
e accept the null that the error terms are not Aut
o-corrolated
```

Out[66]:

```
{'LM Statistic': 3.10673167028625,
 'LM-Test p-value': 0.7953366307235248,
 'F-Statistic': 0.4959264392887972,
 'F-Test p-value': 0.8102445238729414}
```

In [67]:

```
# Break the data into two sets
start_date = dt.datetime(1978,1,1)
end_date = dt.datetime(1987,12,1)

select = (df.index>=start_date)*(df.index<=end_date)

# Copy the selected dataframe into df1
df1=df[select].copy()
df1.head(100)
```

Out[67]:

Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
1978-07-01	0.140	0.032	0.011	0.066	0.143	0.107	0.185
1978-08-01	0.078	0.088	0.024	0.033	0.026	-0.017	-0.021
1978-09-01	-0.059	0.011	0.048	-0.013	-0.031	-0.037	-0.081
1978-10-01	-0.118	-0.071	-0.067	-0.123	-0.085	-0.077	-0.153
1978-11-01	-0.060	-0.005	0.035	-0.038	0.044	0.064	0.055
1978-12-01	0.067	-0.019	0.005	0.047	0.034	0.117	-0.023

1979-01-01	0.168	0.043	0.076	-0.024	-0.008	-0.012	-0.054
1979-02-01	-0.032	-0.082	-0.011	-0.020	-0.015	-0.066	-0.060
1979-03-01	0.178	0.026	0.000	0.043	0.171	0.088	0.098
1979-04-01	-0.043	0.000	-0.057	0.064	0.009	0.005	-0.056
1979-05-01	-0.026	0.022	0.032	0.005	-0.045	-0.028	0.063
1979-06-01	0.057	0.095	0.066	0.092	0.019	0.059	-0.006
1979-07-01	0.047	-0.075	0.015	-0.034	-0.059	0.009	0.075
1979-08-01	0.038	0.065	-0.021	0.058	0.078	0.140	0.021
1979-09-01	0.050	-0.017	0.000	-0.033	-0.031	-0.027	-0.026
1979-10-01	-0.151	-0.125	-0.049	-0.136	-0.246	-0.010	-0.147
1979-11-01	-0.004	0.030	0.109	0.081	0.062	0.095	0.063
1979-12-01	0.042	0.113	0.005	0.104	0.021	0.018	0.020
1980-01-01	0.107	-0.079	-0.039	-0.103	0.157	0.058	0.022
1980-02-01	-0.070	-0.080	-0.061	-0.087	0.043	0.034	-0.093
1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0.031
1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0.018
1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0.144
1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0.010
...	...	...	...	...	...	...	...
1983-11-01	0.147	0.162	-0.025	0.096	-0.014	0.065	0.120

1983-12-01	-0.012	0.023	0.005	-0.016	0.068	0.034	-0.028
1984-01-01	-0.054	0.024	0.005	-0.034	0.117	0.208	-0.013
1984-02-01	-0.088	-0.039	-0.069	-0.101	0.027	-0.024	-0.117
1984-03-01	0.079	-0.054	0.055	-0.033	0.056	0.057	0.065
1984-04-01	0.012	-0.004	0.031	-0.231	0.089	0.053	-0.085
1984-05-01	-0.172	-0.148	0.021	-0.600	-0.094	-0.071	-0.070
1984-06-01	0.025	0.078	0.020	0.000	0.056	-0.043	-0.012
1984-07-01	0.015	-0.029	0.054	-0.205	-0.061	-0.009	0.045
1984-08-01	0.177	0.164	0.029	0.086	0.312	0.159	0.040
1984-09-01	-0.056	0.076	0.051	0.974	-0.132	-0.025	0.008
1984-10-01	0.053	-0.027	0.019	-0.232	0.047	0.093	0.161
1984-11-01	-0.038	0.000	0.004	-0.023	0.019	0.006	-0.026
1984-12-01	0.068	0.098	0.084	0.095	0.096	0.070	0.156
1985-01-01	0.046	0.097	-0.021	0.587	0.215	0.084	-0.010
1985-02-01	-0.059	-0.015	0.034	-0.096	-0.210	-0.067	0.087
1985-03-01	-0.029	0.046	0.057	0.030	-0.195	-0.071	-0.003
1985-04-01	0.010	0.012	0.019	-0.029	-0.157	-0.050	-0.123
1985-05-01	0.158	0.094	0.098	-0.091	-0.078	0.057	0.179
1985-06-01	0.086	0.043	0.046	-0.050	0.060	-0.101	0.021
1985-							

07-01	-0.026	-0.030	-0.084	0.018	0.043	0.080	0.008
1985-08-01	0.011	-0.063	0.043	-0.052	-0.006	0.032	-0.066
1985-09-01	-0.095	-0.085	-0.032	0.036	0.000	0.036	-0.112
1985-10-01	-0.035	0.090	0.066	0.105	0.032	0.040	-0.083
1985-11-01	0.088	0.062	0.032	0.048	0.109	0.073	0.020
1985-12-01	0.064	0.065	0.082	0.197	0.023	0.095	0.030
1986-01-01	0.032	0.005	0.022	0.000	-0.055	0.162	0.122
1986-02-01	0.093	0.101	0.048	-0.051	-0.044	0.093	-0.055
1986-03-01	0.066	0.153	0.021	-0.040	-0.043	-0.063	0.076
1986-04-01	-0.013	-0.042	-0.006	-0.097	0.061	0.119	0.059

100 rows × 27 columns

In [68]:

```
select1 = (df.index>end_date)
df2=df[select1].copy()
df2.head(100)
```

Out[68]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
Date							

0 rows × 27 columns

In [69]:

```
formula = 'Q("CONTIL") ~ Q("MARKET")'

results = smf.ols(formula, df1).fit()
print(results.summary())
```

```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:             Q("CONTIL")   R-square
d:                         0.111
Model:                             OLS   Adj. R-s
quared:                    0.104
Method:              Least Squares   F-statis
tic:                       14.65
Date:             Wed, 18 Dec 2019   Prob (F-
statistic):             0.000209
Time:                     14:39:02   Log-Like
lihood:                   63.430
No. Observations:              119   AIC:
-122.9
Df Residuals:                  117   BIC:
-117.3
Df Model:                        1
Covariance Type:         nonrobust
================================================
===============================
                  coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
-------------------------------
Intercept      -0.0115      0.013     -0.858
0.393      -0.038       0.015
Q("MARKET")     0.7375      0.193      3.828
0.000       0.356       1.119
================================================
==============================
Omnibus:                   106.122   Durbin-W
atson:                     2.068
Prob(Omnibus):               0.000   Jarque-B
era (JB):               2224.184
Skew:                        2.697   Prob(JB)
:                           0.00
Kurtosis:                   23.481   Cond. No
.                           14.7
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
```

In [70]:

```
formula = 'Q("BOISE") ~ Q("MARKET")'

results = smf.ols(formula, df1).fit()
print(results.summary())
```

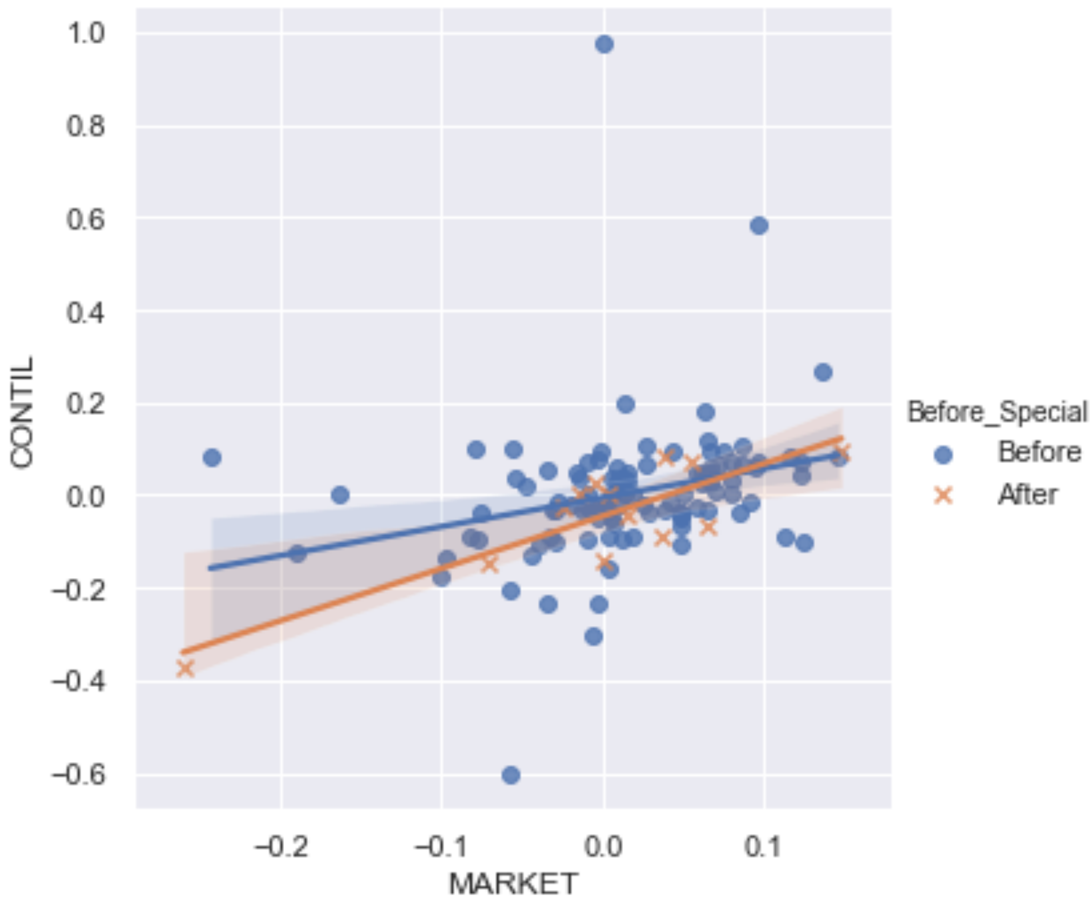
```
                            OLS Regression Results
==================================================
============================
Dep. Variable:             Q("BOISE")   R-squared:
0.422
Model:                            OLS   Adj. R-squ
ared:                      0.417
Method:                 Least Squares   F-statisti
c:                         85.45
Date:                Wed, 18 Dec 2019   Prob (F-st
atistic):               1.31e-15
Time:                        14:39:02   Log-Likeli
hood:                     141.64
No. Observations:                 119   AIC:
-279.3
Df Residuals:                     117   BIC:
-273.7
Df Model:                           1
Covariance Type:            nonrobust
==================================================
=============================
                  coef    std err          t
P>|t|      [0.025      0.975]
--------------------------------------------------
-----------------------------
Intercept       0.0032      0.007      0.456
0.649      -0.011       0.017
Q("MARKET")     0.9230      0.100      9.244
0.000       0.725       1.121
==================================================
============================
Omnibus:                        4.937   Durbin-Wat
son:                       2.183
Prob(Omnibus):                  0.085   Jarque-Ber
a (JB):                    5.734
Skew:                           0.215   Prob(JB):
0.0569
Kurtosis:                       3.986   Cond. No.
14.7
==================================================
============================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
```

In [71]:

```
RF.Chow_Test(df1, y='CONTIL', x='MARKET', special_date='1986-1
1-01')
```

Out[71]:

```
(1.0162280527443794, 0.3651031333275064 4)
```



In [72]:

```
df.head(100)
```

Out[72]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
Date							
1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-							

04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-06-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-07-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-08-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-09-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-11-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-12-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-02-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-03-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-06-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-07-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-08-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-09-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-11-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1977-12-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
...	...	...	...	...	...	...	...
1981-11-01	0.092	0.045	0.038	0.010	0.093	-0.065	-0.032
1981-12-01	-0.029	-0.028	-0.008	-0.106	-0.083	-0.060	-0.062
1982-01-01	-0.084	0.035	0.042	0.102	-0.002	0.027	0.056
1982-02-01	-0.159	0.000	0.036	-0.175	-0.152	-0.049	0.145
1982-03-01	0.108	0.007	0.022	-0.017	-0.302	-0.104	0.038
1982-04-01	-0.009	0.101	0.050	-0.013	0.047	0.054	-0.025
1982-05-01	-0.189	-0.101	0.016	-0.091	-0.180	-0.056	0.042
1982-06-01	-0.044	-0.003	-0.024	-0.096	-0.060	-0.073	0.106
1982-07-01	0.006	-0.025	-0.032	-0.303	-0.054	-0.055	-0.118
1982-08-01	0.379	0.077	0.133	0.070	0.216	0.273	0.055
1982-09-01	-0.109	0.059	0.039	0.058	-0.165	-0.061	-0.139
1982-10-01	0.314	0.318	-0.050	0.268	0.528	0.133	0.171

1982-11-01	0.145	0.007	-0.011	-0.106	0.003	0.175	0.289
1982-12-01	-0.001	-0.098	0.123	0.037	0.053	-0.052	0.093
1983-01-01	-0.045	0.085	-0.012	0.049	0.208	0.225	0.040
1983-02-01	0.037	0.039	0.060	-0.035	0.237	-0.010	0.027
1983-03-01	0.113	0.132	0.048	0.097	0.040	0.034	-0.016
1983-04-01	0.082	0.104	0.045	0.073	0.079	-0.060	-0.043
1983-05-01	-0.014	-0.102	-0.012	0.000	-0.114	-0.052	-0.045
1983-06-01	-0.130	-0.016	0.000	-0.068	-0.042	0.075	0.012
1983-07-01	-0.087	-0.079	0.017	0.046	0.173	-0.142	-0.259
1983-08-01	0.060	-0.007	-0.023	0.055	0.053	0.007	0.080
1983-09-01	0.102	0.006	0.087	-0.026	0.090	-0.005	0.041
1983-10-01	-0.052	-0.118	0.101	-0.088	-0.069	-0.364	0.039
1983-11-01	0.147	0.162	-0.025	0.096	-0.014	0.065	0.120
1983-12-01	-0.012	0.023	0.005	-0.016	0.068	0.034	-0.028
1984-01-01	-0.054	0.024	0.005	-0.034	0.117	0.208	-0.013
1984-02-01	-0.088	-0.039	-0.069	-0.101	0.027	-0.024	-0.117
1984-03-01	0.079	-0.054	0.055	-0.033	0.056	0.057	0.065
1984-04-01	0.012	-0.004	0.031	-0.231	0.089	0.053	-0.085

100 rows × 27 columns

In [73]:

```
df1.head(100)
```

Out[73]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
Date							
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
1978-07-01	0.140	0.032	0.011	0.066	0.143	0.107	0.185
1978-08-01	0.078	0.088	0.024	0.033	0.026	-0.017	-0.021
1978-09-01	-0.059	0.011	0.048	-0.013	-0.031	-0.037	-0.081
1978-10-01	-0.118	-0.071	-0.067	-0.123	-0.085	-0.077	-0.153
1978-11-01	-0.060	-0.005	0.035	-0.038	0.044	0.064	0.055
1978-12-01	0.067	-0.019	0.005	0.047	0.034	0.117	-0.023
1979-01-01	0.168	0.043	0.076	-0.024	-0.008	-0.012	-0.054
1979-02-01	-0.032	-0.082	-0.011	-0.020	-0.015	-0.066	-0.060
1979-03-01	0.178	0.026	0.000	0.043	0.171	0.088	0.098
1979-	-0.043	0.000	-0.057	0.064	0.009	0.005	-0.056

04-01							
1979-05-01	-0.026	0.022	0.032	0.005	-0.045	-0.028	0.063
1979-06-01	0.057	0.095	0.066	0.092	0.019	0.059	-0.006
1979-07-01	0.047	-0.075	0.015	-0.034	-0.059	0.009	0.075
1979-08-01	0.038	0.065	-0.021	0.058	0.078	0.140	0.021
1979-09-01	0.050	-0.017	0.000	-0.033	-0.031	-0.027	-0.026
1979-10-01	-0.151	-0.125	-0.049	-0.136	-0.246	-0.010	-0.147
1979-11-01	-0.004	0.030	0.109	0.081	0.062	0.095	0.063
1979-12-01	0.042	0.113	0.005	0.104	0.021	0.018	0.020
1980-01-01	0.107	-0.079	-0.039	-0.103	0.157	0.058	0.022
1980-02-01	-0.070	-0.080	-0.061	-0.087	0.043	0.034	-0.093
1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0.031
1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0.018
1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0.144
1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0.010
...	...	...	...	...	...	...	...
1983-11-01	0.147	0.162	-0.025	0.096	-0.014	0.065	0.120
1983-12-01	-0.012	0.023	0.005	-0.016	0.068	0.034	-0.028
1984-01-01	-0.054	0.024	0.005	-0.034	0.117	0.208	-0.013
1984-02-01	-0.088	-0.039	-0.069	-0.101	0.027	-0.024	-0.117
1984-							

03-01	0.079	-0.054	0.055	-0.033	0.056	0.057	0.065
1984-04-01	0.012	-0.004	0.031	-0.231	0.089	0.053	-0.085
1984-05-01	-0.172	-0.148	0.021	-0.600	-0.094	-0.071	-0.070
1984-06-01	0.025	0.078	0.020	0.000	0.056	-0.043	-0.012
1984-07-01	0.015	-0.029	0.054	-0.205	-0.061	-0.009	0.045
1984-08-01	0.177	0.164	0.029	0.086	0.312	0.159	0.040
1984-09-01	-0.056	0.076	0.051	0.974	-0.132	-0.025	0.008
1984-10-01	0.053	-0.027	0.019	-0.232	0.047	0.093	0.161
1984-11-01	-0.038	0.000	0.004	-0.023	0.019	0.006	-0.026
1984-12-01	0.068	0.098	0.084	0.095	0.096	0.070	0.156
1985-01-01	0.046	0.097	-0.021	0.587	0.215	0.084	-0.010
1985-02-01	-0.059	-0.015	0.034	-0.096	-0.210	-0.067	0.087
1985-03-01	-0.029	0.046	0.057	0.030	-0.195	-0.071	-0.003
1985-04-01	0.010	0.012	0.019	-0.029	-0.157	-0.050	-0.123
1985-05-01	0.158	0.094	0.098	-0.091	-0.078	0.057	0.179
1985-06-01	0.086	0.043	0.046	-0.050	0.060	-0.101	0.021
1985-07-01	-0.026	-0.030	-0.084	0.018	0.043	0.080	0.008
1985-08-01	0.011	-0.063	0.043	-0.052	-0.006	0.032	-0.066
1985-09-01	-0.095	-0.085	-0.032	0.036	0.000	0.036	-0.112
1985-10-01	-0.035	0.090	0.066	0.105	0.032	0.040	-0.083

1985-11-01	0.088	0.062	0.032	0.048	0.109	0.073	0.020
1985-12-01	0.064	0.065	0.082	0.197	0.023	0.095	0.030
1986-01-01	0.032	0.005	0.022	0.000	-0.055	0.162	0.122
1986-02-01	0.093	0.101	0.048	-0.051	-0.044	0.093	-0.055
1986-03-01	0.066	0.153	0.021	-0.040	-0.043	-0.063	0.076
1986-04-01	-0.013	-0.042	-0.006	-0.097	0.061	0.119	0.059

100 rows × 28 columns

In [74]:

```
formula_with_intercept = 'BOISE ~ MARKET + CPI + FRBIND'

results_with_intercept = smf.ols(formula_with_intercept, df1).
fit()
print(results_with_intercept.summary())
```

```
                            OLS Regression Results
==================================================
==============================
Dep. Variable:                  BOISE   R-squared:
0.430
Model:                            OLS   Adj. R-squ
ared:                       0.415
Method:                 Least Squares   F-statisti
c:                          28.91
Date:                Wed, 18 Dec 2019   Prob (F-st
atistic):                5.27e-14
Time:                        14:39:12   Log-Likeli
hood:                      142.46
No. Observations:                 119   AIC:
-276.9
Df Residuals:                     115   BIC:
-265.8
Df Model:                           3
Covariance Type:            nonrobust
==================================================
```

```
==============================
                  coef    std err          t       P
>|t|      [0.025      0.975]
------------------------------------------------------
------------------------------
Intercept      -0.0426      0.096     -0.445       0
.657      -0.232       0.147
MARKET          0.9416      0.102      9.274       0
.000       0.740       1.143
CPI             0.0002      0.000      1.016       0
.312      -0.000       0.001
FRBIND      -6.588e-05      0.001     -0.085       0
.933      -0.002       0.001
======================================================
==============================
Omnibus:                          4.443   Durbin-Wat
son:                       2.215
Prob(Omnibus):                    0.108   Jarque-Ber
a (JB):                    4.956
Skew:                             0.196   Prob(JB):
0.0839
Kurtosis:                         3.920   Cond. No.
5.07e+03
======================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
[2] The condition number is large, 5.07e+03. This
might indicate that there are
strong multicollinearity or other numerical proble
ms.
```

In [75]:

```
df.head(100)
```

Out[75]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
Date							
1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-06-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-07-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-08-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-09-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-11-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1976-12-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-02-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-03-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-06-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-07-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-08-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-09-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-							

10-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-11-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1977-12-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
...	...	...	...	...	...	...	...
1981-11-01	0.092	0.045	0.038	0.010	0.093	-0.065	-0.032
1981-12-01	-0.029	-0.028	-0.008	-0.106	-0.083	-0.060	-0.062
1982-01-01	-0.084	0.035	0.042	0.102	-0.002	0.027	0.056
1982-02-01	-0.159	0.000	0.036	-0.175	-0.152	-0.049	0.145
1982-03-01	0.108	0.007	0.022	-0.017	-0.302	-0.104	0.038
1982-04-01	-0.009	0.101	0.050	-0.013	0.047	0.054	-0.025
1982-05-01	-0.189	-0.101	0.016	-0.091	-0.180	-0.056	0.042
1982-06-01	-0.044	-0.003	-0.024	-0.096	-0.060	-0.073	0.106
1982-07-01	0.006	-0.025	-0.032	-0.303	-0.054	-0.055	-0.118
1982-08-01	0.379	0.077	0.133	0.070	0.216	0.273	0.055

1982-09-01	-0.109	0.059	0.039	0.058	-0.165	-0.061	-0.139
1982-10-01	0.314	0.318	-0.050	0.268	0.528	0.133	0.171
1982-11-01	0.145	0.007	-0.011	-0.106	0.003	0.175	0.289
1982-12-01	-0.001	-0.098	0.123	0.037	0.053	-0.052	0.093
1983-01-01	-0.045	0.085	-0.012	0.049	0.208	0.225	0.040
1983-02-01	0.037	0.039	0.060	-0.035	0.237	-0.010	0.027
1983-03-01	0.113	0.132	0.048	0.097	0.040	0.034	-0.016
1983-04-01	0.082	0.104	0.045	0.073	0.079	-0.060	-0.043
1983-05-01	-0.014	-0.102	-0.012	0.000	-0.114	-0.052	-0.045
1983-06-01	-0.130	-0.016	0.000	-0.068	-0.042	0.075	0.012
1983-07-01	-0.087	-0.079	0.017	0.046	0.173	-0.142	-0.259
1983-08-01	0.060	-0.007	-0.023	0.055	0.053	0.007	0.080
1983-09-01	0.102	0.006	0.087	-0.026	0.090	-0.005	0.041
1983-10-01	-0.052	-0.118	0.101	-0.088	-0.069	-0.364	0.039
1983-11-01	0.147	0.162	-0.025	0.096	-0.014	0.065	0.120
1983-12-01	-0.012	0.023	0.005	-0.016	0.068	0.034	-0.028
1984-01-01	-0.054	0.024	0.005	-0.034	0.117	0.208	-0.013
1984-02-01	-0.088	-0.039	-0.069	-0.101	0.027	-0.024	-0.117
1984-03-01	0.079	-0.054	0.055	-0.033	0.056	0.057	0.065
1984-							

04-01	0.012	-0.004	0.031	-0.231	0.089	0.053	-0.085

100 rows × 27 columns

In [76]:

```
df4=df1[['BOISE','CONTIL','MARKET', 'CPI', 'POIL', 'FRBIND']].
dropna()
df4['RINF']=df4['CPI'].pct_change(1)
df4['GIND']=df4['FRBIND'].pct_change(1)
df4['real_POIL']=df4['POIL']/df4['CPI']
df4['ROIL']=df4['real_POIL'].pct_change(1)
df4.head(100)
```

Out[76]:

	BOISE	CONTIL	MARKET	CPI	POIL	FRBIND	RINF
Date							
1978-01-01	-0.079	-0.129	-0.045	187.2	8.68	138.8	NaN
1978-02-01	0.013	0.037	0.010	188.4	8.84	139.2	0.006410
1978-03-01	0.070	0.003	0.050	189.8	8.80	140.9	0.007431
1978-04-01	0.120	0.180	0.063	191.5	8.82	143.2	0.008957
1978-05-01	0.071	0.061	0.067	193.3	8.81	143.9	0.009399
1978-06-01	-0.098	-0.059	0.007	195.3	9.05	144.9	0.010347
1978-07-01	0.140	0.066	0.071	196.7	8.96	146.1	0.007168
1978-08-01	0.078	0.033	0.079	197.8	8.05	147.1	0.005592
1978-09-01	-0.059	-0.013	0.002	199.3	9.15	147.8	0.007583
1978-10-01	-0.118	-0.123	-0.189	200.9	9.17	148.6	0.008028
1978-11-01	-0.060	-0.038	0.084	202.0	9.20	149.5	0.005475

1978-12-01	0.067	0.047	0.015	203.3	9.47	150.4	0.006436
1979-01-01	0.168	-0.024	0.058	204.7	9.46	152.0	0.006886
1979-02-01	-0.032	-0.020	0.011	207.1	9.69	152.5	0.011724
1979-03-01	0.178	0.043	0.123	209.1	9.83	153.5	0.009657
1979-04-01	-0.043	0.064	0.026	211.5	10.33	151.1	0.011478
1979-05-01	-0.026	0.005	0.014	214.1	10.71	152.7	0.012293
1979-06-01	0.057	0.092	0.075	216.6	11.70	153.0	0.011677
1979-07-01	0.047	-0.034	-0.013	218.9	13.39	153.0	0.010619
1979-08-01	0.038	0.058	0.095	221.1	14.00	152.1	0.010050
1979-09-01	0.050	-0.033	0.039	223.4	14.57	152.7	0.010403
1979-10-01	-0.151	-0.136	-0.097	225.4	15.11	152.7	0.008953
1979-11-01	-0.004	0.081	0.116	227.5	15.52	152.3	0.009317
1979-12-01	0.042	0.104	0.086	229.9	17.03	152.5	0.010549
1980-01-01	0.107	-0.103	0.124	233.2	17.86	152.7	0.014354
1980-02-01	-0.070	-0.087	0.112	236.4	18.81	152.6	0.013722
1980-03-01	-0.138	0.085	-0.243	239.8	19.34	152.1	0.014382
1980-04-01	0.042	0.074	0.080	242.5	20.29	148.3	0.011259
1980-05-01	0.109	0.023	0.062	244.9	21.01	144.0	0.009897
1980-06-01	0.068	0.064	0.086	247.6	21.53	141.5	0.011025
...	...	...	...	...	...	...	...

1983-11-01	0.147	0.096	0.066	303.1	26.09	155.3	0.001652
1983-12-01	-0.012	-0.016	-0.012	303.5	25.88	156.2	0.001320
1984-01-01	-0.054	-0.034	-0.029	305.4	25.93	158.5	0.006260
1984-02-01	-0.088	-0.101	-0.030	306.6	26.06	160.0	0.003929
1984-03-01	0.079	-0.033	0.003	307.3	26.05	160.8	0.002283
1984-04-01	0.012	-0.231	-0.003	308.8	25.93	162.1	0.004881
1984-05-01	-0.172	-0.600	-0.058	309.7	26.00	162.8	0.002915
1984-06-01	0.025	0.000	0.005	310.7	26.09	164.4	0.003229
1984-07-01	0.015	-0.205	-0.058	311.7	26.11	165.9	0.003219
1984-08-01	0.177	0.086	0.146	313.0	26.02	166.0	0.004171
1984-09-01	-0.056	0.974	0.000	314.5	25.97	165.0	0.004792
1984-10-01	0.053	-0.232	-0.035	315.3	25.92	164.5	0.002544
1984-11-01	-0.038	-0.023	-0.019	315.3	25.44	165.2	0.000000
1984-12-01	0.068	0.095	-0.001	315.5	25.05	166.2	0.000634
1985-01-01	0.046	0.587	0.097	316.1	24.28	165.6	0.001902
1985-02-01	-0.059	-0.096	0.012	317.4	23.63	165.7	0.004113
1985-03-01	-0.029	0.030	0.008	318.8	23.88	166.1	0.004411
1985-04-01	0.010	-0.029	-0.010	320.1	24.15	166.2	0.004078
1985-05-01	0.158	-0.091	0.019	321.3	24.18	166.2	0.003749
1985-							

06-01	0.086	-0.050	-0.003	322.3	24.03	166.5	0.003112
1985-07-01	-0.026	0.018	0.012	322.8	24.00	166.2	0.001551
1985-08-01	0.011	-0.052	0.005	323.5	23.92	167.7	0.00216
1985-09-01	-0.095	0.036	-0.055	324.5	23.93	167.6	0.003091
1985-10-01	-0.035	0.105	0.026	325.5	24.06	166.6	0.003082
1985-11-01	0.088	0.048	0.059	326.6	24.31	167.6	0.00337
1985-12-01	0.064	0.197	0.013	327.4	24.53	168.8	0.00244
1986-01-01	0.032	0.000	-0.009	328.4	23.12	169.6	0.003054
1986-02-01	0.093	-0.051	0.049	327.5	17.65	168.4	-0.002741
1986-03-01	0.066	-0.040	0.048	326.0	12.62	166.1	-0.00458
1986-04-01	-0.013	-0.097	-0.009	325.3	10.68	167.6	-0.002147

In [77]:

```
formula_without_intercept = 'CONTIL ~ MARKET + RINF + GIND + R
OIL'

results_without_intercept = smf.ols(formula_without_intercept,
df4).fit()
print(results_without_intercept.summary())
```

```
                            OLS Regression Results
==============================================================================
==============================
Dep. Variable:                 CONTIL   R-squared:
0.124
Model:                            OLS   Adj. R-squ
ared:                     0.093
Method:                 Least Squares   F-statisti
c:                        3.988
Date:                Wed, 18 Dec 2019   Prob (F-st
atistic):               0.00460
```

```
Time:                            14:39:21   Log-Likeli
hood:                      63.586
No. Observations:                     118   AIC:
-117.2
Df Residuals:                         113   BIC:
-103.3
Df Model:                               4
Covariance Type:            nonrobust
======================================================
============================
                 coef    std err          t      P
>|t|      [0.025      0.975]
------------------------------------------------------
----------------------------
Intercept     -0.0096      0.024     -0.399      0
.691      -0.057       0.038
MARKET         0.6962      0.196      3.550      0
.001       0.308       1.085
RINF           0.5381      3.769      0.143      0
.887      -6.929       8.005
GIND          -1.6486      1.440     -1.145      0
.255      -4.502       1.205
ROIL           0.1740      0.265      0.657      0
.512      -0.351       0.699
======================================================
============================
Omnibus:                          103.711   Durbin-Wat
son:                        2.058
Prob(Omnibus):                      0.000   Jarque-Ber
a (JB):                  2129.891
Skew:                               2.634   Prob(JB):
0.00
Kurtosis:                          23.136   Cond. No.
284.
======================================================
============================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
```

In [78]:

```
formula_without_intercept = 'BOISE ~ MARKET + RINF + GIND + RO
IL'

results_without_intercept = smf.ols(formula_without_intercept,
df4).fit()
print(results_without_intercept.summary())
```

```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:                  BOISE   R-square
d:                       0.459
Model:                            OLS   Adj. R-s
quared:                  0.440
Method:                 Least Squares   F-statis
tic:                     23.97
Date:                Wed, 18 Dec 2019   Prob (F-
statistic):           2.26e-14
Time:                        14:39:22   Log-Like
lihood:                 144.34
No. Observations:                 118   AIC:
-278.7
Df Residuals:                     113   BIC:
-264.8
Df Model:                           4
Covariance Type:            nonrobust
================================================
==============================
                 coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
------------------------------
Intercept      0.0320      0.012      2.630
0.010       0.008       0.056
MARKET         0.9047      0.099      9.145
0.000       0.709       1.101
RINF          -5.1257      1.901     -2.696
0.008      -8.892      -1.359
GIND          -0.7855      0.727     -1.081
0.282      -2.225       0.654
ROIL           0.2110      0.134      1.580
0.117      -0.054       0.476
================================================
==============================
Omnibus:                        1.808   Durbin-W
```

```
atson:                    2.325
Prob(Omnibus):                   0.405   Jarque-B
era (JB):                 1.343
Skew:                            0.110   Prob(JB)
:                         0.511
Kurtosis:                        3.474   Cond. No
.                          284.
=================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
```

In [79]:

```
df3.head(100)
df.dropna()
```

Out[79]:

Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
1978-07-01	0.140	0.032	0.011	0.066	0.143	0.107	0.185
1978-08-01	0.078	0.088	0.024	0.033	0.026	-0.017	-0.021
1978-09-01	-0.059	0.011	0.048	-0.013	-0.031	-0.037	-0.081
1978-10-01	-0.118	-0.071	-0.067	-0.123	-0.085	-0.077	-0.153

1978-11-01	-0.060	-0.005	0.035	-0.038	0.044	0.064	0.055
1978-12-01	0.067	-0.019	0.005	0.047	0.034	0.117	-0.023
1979-01-01	0.168	0.043	0.076	-0.024	-0.008	-0.012	-0.054
1979-02-01	-0.032	-0.082	-0.011	-0.020	-0.015	-0.066	-0.060
1979-03-01	0.178	0.026	0.000	0.043	0.171	0.088	0.098
1979-04-01	-0.043	0.000	-0.057	0.064	0.009	0.005	-0.056
1979-05-01	-0.026	0.022	0.032	0.005	-0.045	-0.028	0.063
1979-06-01	0.057	0.095	0.066	0.092	0.019	0.059	-0.006
1979-07-01	0.047	-0.075	0.015	-0.034	-0.059	0.009	0.075
1979-08-01	0.038	0.065	-0.021	0.058	0.078	0.140	0.021
1979-09-01	0.050	-0.017	0.000	-0.033	-0.031	-0.027	-0.026
1979-10-01	-0.151	-0.125	-0.049	-0.136	-0.246	-0.010	-0.147
1979-11-01	-0.004	0.030	0.109	0.081	0.062	0.095	0.063
1979-12-01	0.042	0.113	0.005	0.104	0.021	0.018	0.020
1980-01-01	0.107	-0.079	-0.039	-0.103	0.157	0.058	0.022
1980-02-01	-0.070	-0.080	-0.061	-0.087	0.043	0.034	-0.093
1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0.031
1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0.018
1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0.144
1980-	0.068	0.058	0.040	0.064	0.108	0.021	0.010

06-01							
1980-07-01	0.073	-0.023	-0.027	-0.034	0.212	0.183	0.283
1980-08-01	-0.045	0.029	-0.005	-0.018	0.058	0.081	-0.056
1980-09-01	0.019	-0.068	-0.010	0.034	-0.136	0.045	-0.053
1980-10-01	-0.054	-0.049	-0.021	0.035	0.007	-0.028	0.046
1980-11-01	0.028	0.123	-0.035	-0.017	0.000	0.056	0.220
1980-12-01	-0.047	0.131	0.131	0.103	-0.098	0.035	0.040
1981-01-01	0.011	-0.062	-0.015	0.040	-0.231	-0.089	0.112
1981-02-01	0.152	-0.005	-0.021	0.069	-0.072	0.006	0.031
1981-03-01	0.056	0.045	0.151	0.024	0.184	0.075	0.024
1981-04-01	0.045	0.086	0.061	-0.025	0.088	0.075	0.062
1981-05-01	0.032	0.099	0.017	0.117	0.112	0.107	0.105
1981-06-01	-0.037	-0.013	0.022	0.077	-0.178	-0.112	-0.114
1981-07-01	-0.065	-0.019	0.026	-0.092	0.007	-0.014	-0.094
1981-08-01	-0.125	-0.108	0.021	-0.030	-0.191	-0.065	-0.072
1981-09-01	-0.062	0.032	-0.013	0.003	0.089	-0.019	-0.013

In [80]:

```
np.percentile(df3['CONTIL'],25)
```

Out[80]:

```
-0.069975
```

In [ ]:

```

```

In [81]:

```
np.percentile(df3['BOISE'],25)
```

Out[81]:

```
-0.05036
```

In [82]:

```
print("25th percentile of CONTIL stock ", np.percentile(df3['C
ONTIL'],25))
```

```
25th percentile of CONTIL stock  -0.069975
```

In [83]:

```
print("25th percentile of CONTIL stock ", np.percentile(df3['C
ONTIL'],25))
print("75th percentile of CONTIL stock ", np.percentile(df3['C
ONTIL'],75))
```

```
25th percentile of CONTIL stock  -0.069975
75th percentile of CONTIL stock  0.06158
```

In [84]:

```
IQR=np.percentile(df3['CONTIL'],75)-np.percentile(df3['CONTIL'
],25)
IQR
```

Out[84]:

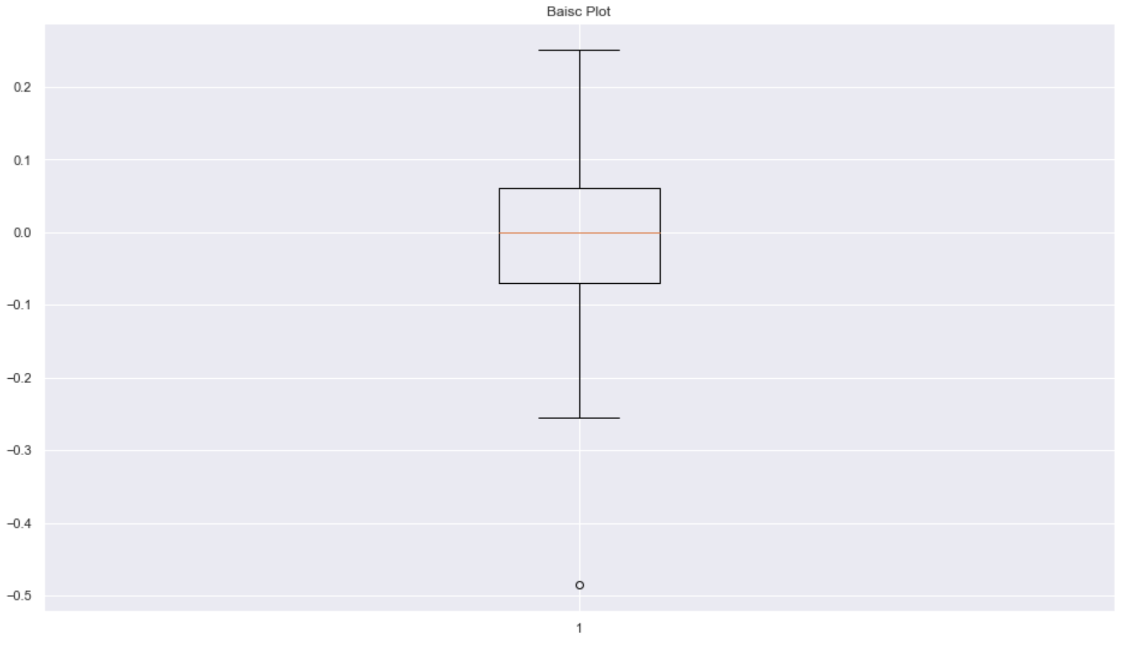
```
0.131555
```

In [85]:

```
fig1, ax1 = plt.subplots()
ax1.set_title('Baisc Plot')
ax1.boxplot(df3['CONTIL'])
```

Out[85]:

```
{'whiskers': [<matplotlib.lines.Line2D at 0x1c1fa1
3550>,
  <matplotlib.lines.Line2D at 0x1c1fa13668>],
 'caps': [<matplotlib.lines.Line2D at 0x1c1fb60400
>,
  <matplotlib.lines.Line2D at 0x1c1fb60780>],
 'boxes': [<matplotlib.lines.Line2D at 0x1c1fa130f
0>],
 'medians': [<matplotlib.lines.Line2D at 0x1c1fb60
b00>],
 'fliers': [<matplotlib.lines.Line2D at 0x1c1fb60e
80>],
 'means': []}
```



In [86]:

```
df =pd.read_excel('Data_For_Analysis.xlsx')
df.head(80)
```

Out[86]:

	Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DI
0	1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	

1	1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN
2	1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN
3	1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN
4	1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN
5	1976-06-01	NaN	NaN	NaN	NaN	NaN	NaN
6	1976-07-01	NaN	NaN	NaN	NaN	NaN	NaN
7	1976-08-01	NaN	NaN	NaN	NaN	NaN	NaN
8	1976-09-01	NaN	NaN	NaN	NaN	NaN	NaN
9	1976-10-01	NaN	NaN	NaN	NaN	NaN	NaN
10	1976-11-01	NaN	NaN	NaN	NaN	NaN	NaN
11	1976-12-01	NaN	NaN	NaN	NaN	NaN	NaN
12	1977-01-01	NaN	NaN	NaN	NaN	NaN	NaN
13	1977-02-01	NaN	NaN	NaN	NaN	NaN	NaN
14	1977-03-01	NaN	NaN	NaN	NaN	NaN	NaN
15	1977-04-01	NaN	NaN	NaN	NaN	NaN	NaN
16	1977-05-01	NaN	NaN	NaN	NaN	NaN	NaN
17	1977-06-01	NaN	NaN	NaN	NaN	NaN	NaN
18	1977-07-01	NaN	NaN	NaN	NaN	NaN	NaN
19	1977-08-01	NaN	NaN	NaN	NaN	NaN	NaN
20	1977-09-01	NaN	NaN	NaN	NaN	NaN	NaN
21	1977-	NaN	NaN	NaN	NaN	NaN	NaN

	10-01							
22	1977-11-01	NaN	NaN	NaN	NaN	NaN	NaN	
23	1977-12-01	NaN	NaN	NaN	NaN	NaN	NaN	
24	1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0
25	1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0
26	1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0
27	1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0
28	1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0
29	1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0
...	...	...	...	...	...	...	...	
50	1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0
51	1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0
52	1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0
53	1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0
54	1980-07-01	0.073	-0.023	-0.027	-0.034	0.212	0.183	0
55	1980-08-01	-0.045	0.029	-0.005	-0.018	0.058	0.081	-0
56	1980-09-01	0.019	-0.068	-0.010	0.034	-0.136	0.045	-0
57	1980-10-01	-0.054	-0.049	-0.021	0.035	0.007	-0.028	0
58	1980-11-01	0.028	0.123	-0.035	-0.017	0.000	0.056	0
59	1980-12-01	-0.047	0.131	0.131	0.103	-0.098	0.035	0
60	1981-01-01	0.011	-0.062	-0.015	0.040	-0.231	-0.089	0

61	1981-02-01	0.152	-0.005	-0.021	0.069	-0.072	0.006
62	1981-03-01	0.056	0.045	0.151	0.024	0.184	0.075
63	1981-04-01	0.045	0.086	0.061	-0.025	0.088	0.075
64	1981-05-01	0.032	0.099	0.017	0.117	0.112	0.107
65	1981-06-01	-0.037	-0.013	0.022	0.077	-0.178	-0.112
66	1981-07-01	-0.065	-0.019	0.026	-0.092	0.007	-0.014
67	1981-08-01	-0.125	-0.108	0.021	-0.030	-0.191	-0.065
68	1981-09-01	-0.062	0.032	-0.013	0.003	0.089	-0.019
69	1981-10-01	0.016	0.052	0.112	0.049	0.094	0.102
70	1981-11-01	0.092	0.045	0.038	0.010	0.093	-0.065
71	1981-12-01	-0.029	-0.028	-0.008	-0.106	-0.083	-0.060
72	1982-01-01	-0.084	0.035	0.042	0.102	-0.002	0.027
73	1982-02-01	-0.159	0.000	0.036	-0.175	-0.152	-0.049
74	1982-03-01	0.108	0.007	0.022	-0.017	-0.302	-0.104
75	1982-04-01	-0.009	0.101	0.050	-0.013	0.047	0.054
76	1982-05-01	-0.189	-0.101	0.016	-0.091	-0.180	-0.056
77	1982-06-01	-0.044	-0.003	-0.024	-0.096	-0.060	-0.073
78	1982-07-01	0.006	-0.025	-0.032	-0.303	-0.054	-0.055
79	1982-08-01	0.379	0.077	0.133	0.070	0.216	0.273

In [87]:

```
df =pd.read_excel('Data_For_Analysis.xlsx')
df.head(80)
```

Out[87]:

	Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DI
0	1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	
1	1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	
2	1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	
3	1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN	
4	1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN	
5	1976-06-01	NaN	NaN	NaN	NaN	NaN	NaN	
6	1976-07-01	NaN	NaN	NaN	NaN	NaN	NaN	
7	1976-08-01	NaN	NaN	NaN	NaN	NaN	NaN	
8	1976-09-01	NaN	NaN	NaN	NaN	NaN	NaN	
9	1976-10-01	NaN	NaN	NaN	NaN	NaN	NaN	
10	1976-11-01	NaN	NaN	NaN	NaN	NaN	NaN	
11	1976-12-01	NaN	NaN	NaN	NaN	NaN	NaN	
12	1977-01-01	NaN	NaN	NaN	NaN	NaN	NaN	
13	1977-02-01	NaN	NaN	NaN	NaN	NaN	NaN	
14	1977-03-01	NaN	NaN	NaN	NaN	NaN	NaN	
15	1977-04-01	NaN	NaN	NaN	NaN	NaN	NaN	
16	1977-	NaN	NaN	NaN	NaN	NaN	NaN	

	05-01							
17	1977-06-01	NaN	NaN	NaN	NaN	NaN	NaN	
18	1977-07-01	NaN	NaN	NaN	NaN	NaN	NaN	
19	1977-08-01	NaN	NaN	NaN	NaN	NaN	NaN	
20	1977-09-01	NaN	NaN	NaN	NaN	NaN	NaN	
21	1977-10-01	NaN	NaN	NaN	NaN	NaN	NaN	
22	1977-11-01	NaN	NaN	NaN	NaN	NaN	NaN	
23	1977-12-01	NaN	NaN	NaN	NaN	NaN	NaN	
24	1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0
25	1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0
26	1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0
27	1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0
28	1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0
29	1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0
...	...	...	...	...	...	...	...	
50	1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0
51	1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0
52	1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0
53	1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0
54	1980-07-01	0.073	-0.023	-0.027	-0.034	0.212	0.183	0
55	1980-08-01	-0.045	0.029	-0.005	-0.018	0.058	0.081	-0

56	1980-09-01	0.019	-0.068	-0.010	0.034	-0.136	0.045	-0
57	1980-10-01	-0.054	-0.049	-0.021	0.035	0.007	-0.028	0
58	1980-11-01	0.028	0.123	-0.035	-0.017	0.000	0.056	0
59	1980-12-01	-0.047	0.131	0.131	0.103	-0.098	0.035	0
60	1981-01-01	0.011	-0.062	-0.015	0.040	-0.231	-0.089	0
61	1981-02-01	0.152	-0.005	-0.021	0.069	-0.072	0.006	0
62	1981-03-01	0.056	0.045	0.151	0.024	0.184	0.075	0
63	1981-04-01	0.045	0.086	0.061	-0.025	0.088	0.075	0
64	1981-05-01	0.032	0.099	0.017	0.117	0.112	0.107	0
65	1981-06-01	-0.037	-0.013	0.022	0.077	-0.178	-0.112	-0
66	1981-07-01	-0.065	-0.019	0.026	-0.092	0.007	-0.014	-0
67	1981-08-01	-0.125	-0.108	0.021	-0.030	-0.191	-0.065	-0
68	1981-09-01	-0.062	0.032	-0.013	0.003	0.089	-0.019	-0
69	1981-10-01	0.016	0.052	0.112	0.049	0.094	0.102	-0
70	1981-11-01	0.092	0.045	0.038	0.010	0.093	-0.065	-0
71	1981-12-01	-0.029	-0.028	-0.008	-0.106	-0.083	-0.060	-0
72	1982-01-01	-0.084	0.035	0.042	0.102	-0.002	0.027	0
73	1982-02-01	-0.159	0.000	0.036	-0.175	-0.152	-0.049	0
74	1982-03-01	0.108	0.007	0.022	-0.017	-0.302	-0.104	0
75	1982-04-01	-0.009	0.101	0.050	-0.013	0.047	0.054	-0
	1982-							

76	05-01	-0.189	-0.101	0.016	-0.091	-0.180	-0.056	0
77	1982-06-01	-0.044	-0.003	-0.024	-0.096	-0.060	-0.073	0
78	1982-07-01	0.006	-0.025	-0.032	-0.303	-0.054	-0.055	-0
79	1982-08-01	0.379	0.077	0.133	0.070	0.216	0.273	0

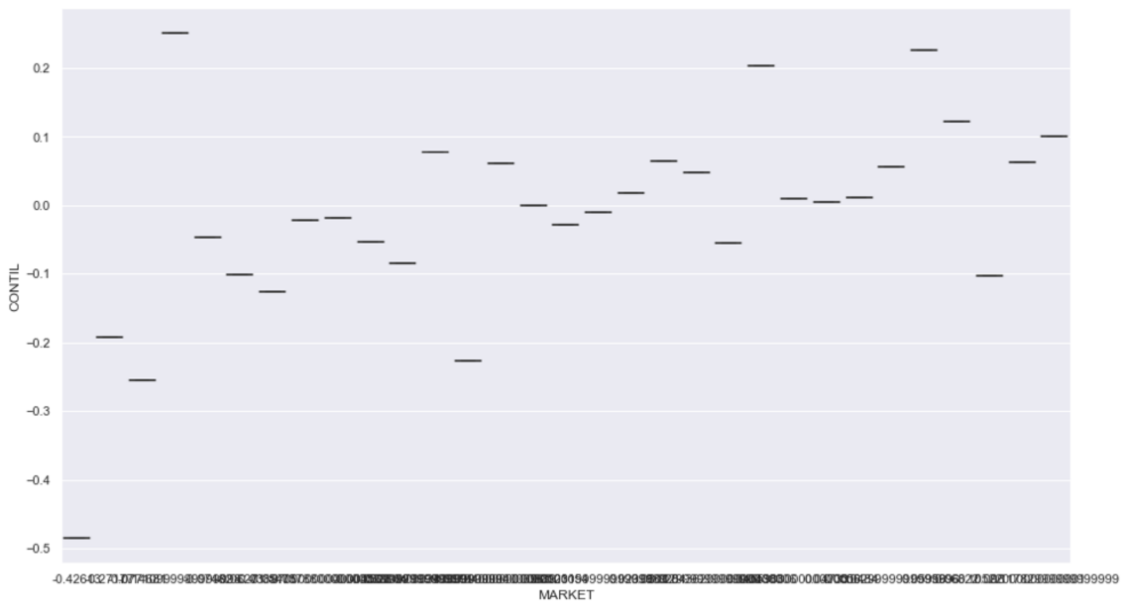
80 rows × 28 columns

In [88]:

```
sns.boxplot(x="MARKET", y='CONTIL', data=df3, palette="Set1")
```

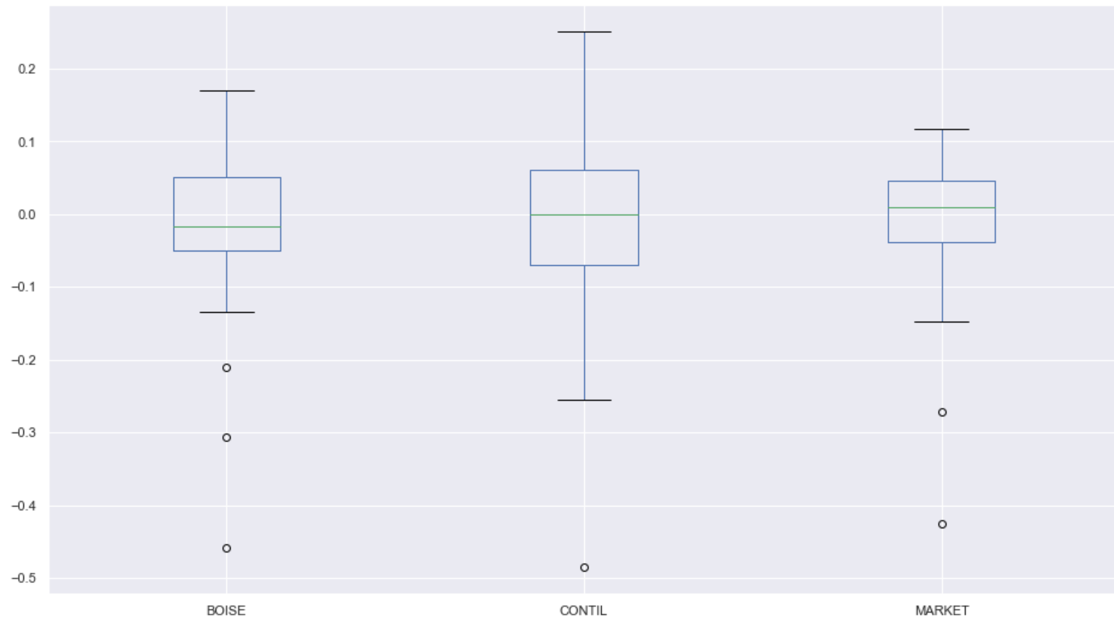
Out[88]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c1fb7
fa90>
```



In [89]:

```
boxplot = df3.boxplot()
```



In [90]:

```
df3.var()
```

Out[90]:

```
BOISE     0.016534
CONTIL    0.021317
MARKET    0.011777
dtype: float64
```

In [91]:

```
df3.std
df3.dropna()
```

Out[91]:

	BOISE	CONTIL	MARKET
Date			
1979-03-01	0.05043	-0.08457	-0.00457
1979-04-01	-0.04300	0.06400	0.02600
1979-05-01	-0.03111	-0.00011	0.00889
1979-06-01	-0.05697	-0.02197	-0.03897
1979-07-01	0.02720	-0.05380	-0.03280

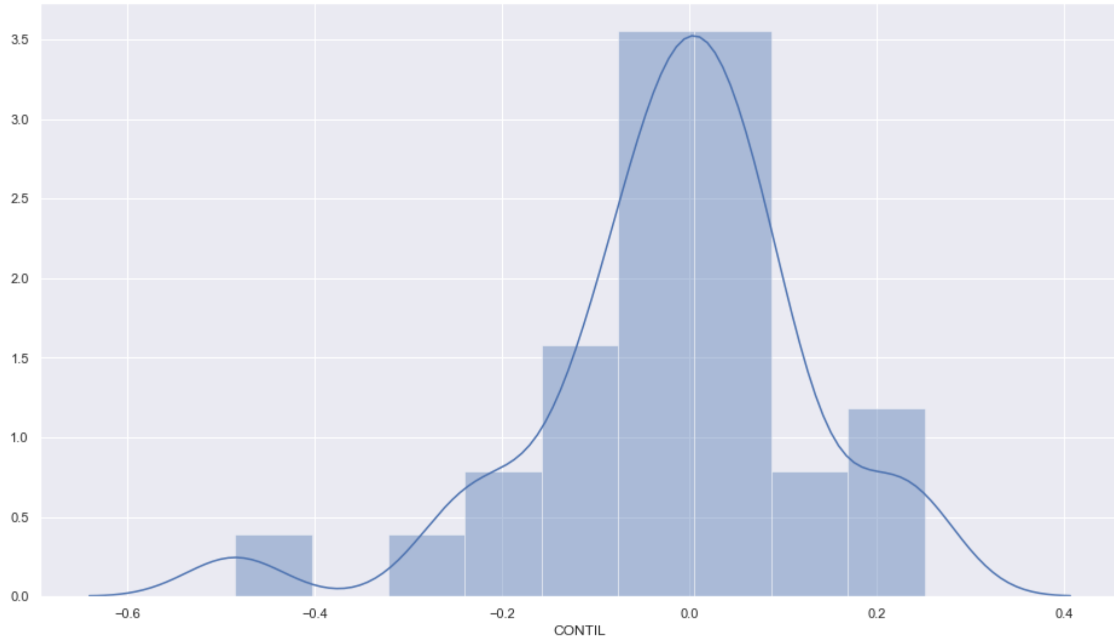
1979-08-01	-0.00860	0.01140	0.04840
1979-09-01	-0.04375	-0.12675	-0.05475
1979-10-01	-0.11671	-0.10171	-0.06271
1979-11-01	-0.07441	0.01059	0.04559
1979-12-01	-0.01387	0.04813	0.03013
1980-01-01	-0.01734	-0.22734	-0.00034
1980-02-01	-0.08600	-0.10300	0.09600
1980-03-01	0.02789	0.25089	-0.07711
1980-04-01	-0.01402	0.01798	0.02398
1980-05-01	0.05754	-0.02846	0.01054
1980-06-01	-0.00598	-0.00998	0.01202
1980-07-01	0.05162	-0.05538	0.04362
1980-08-01	-0.02267	0.00433	0.04733
1980-09-01	-0.03388	-0.01888	-0.03788
1980-10-01	-0.13482	-0.04582	-0.07482
1980-11-01	-0.21081	-0.25581	-0.14681
1980-12-01	0.05283	0.20283	0.04383
1981-01-01	0.09322	0.12222	0.06822
1981-02-01	0.15992	0.07692	-0.00108
1981-03-01	0.09422	0.06222	0.10522
1981-04-01	0.17083	0.10083	0.11783
1981-05-01	-0.02406	0.06094	0.00794
1981-06-01	-0.30577	-0.19177	-0.27177
1981-07-01	-0.45813	-0.48513	-0.42613
1981-08-01	-0.03897	0.05603	0.05503
1981-09-01	0.16156	0.22656	0.05956

In [92]:

```
sns.distplot(df3['CONTIL'])
```

Out[92]:

```
&lt;matplotlib.axes._subplots.AxesSubplot at 0x1c1f16
2ba8&gt;
```



In [93]:

```
m, s = stats.norm.fit(df3['CONTIL'])
```

In [94]:

```
m
```

Out[94]:

```
-0.015955161290322577
```

In [95]:

```
s
```

Out[95]:

```
0.14362765395219923
```

In [96]:

```
m, s = stats.norm.fit(df3['BOISE'])
```

In [97]:

```
m
```

Out[97]:

```
-0.025600322580645162
```

In [98]:

```
s
```

Out[98]:

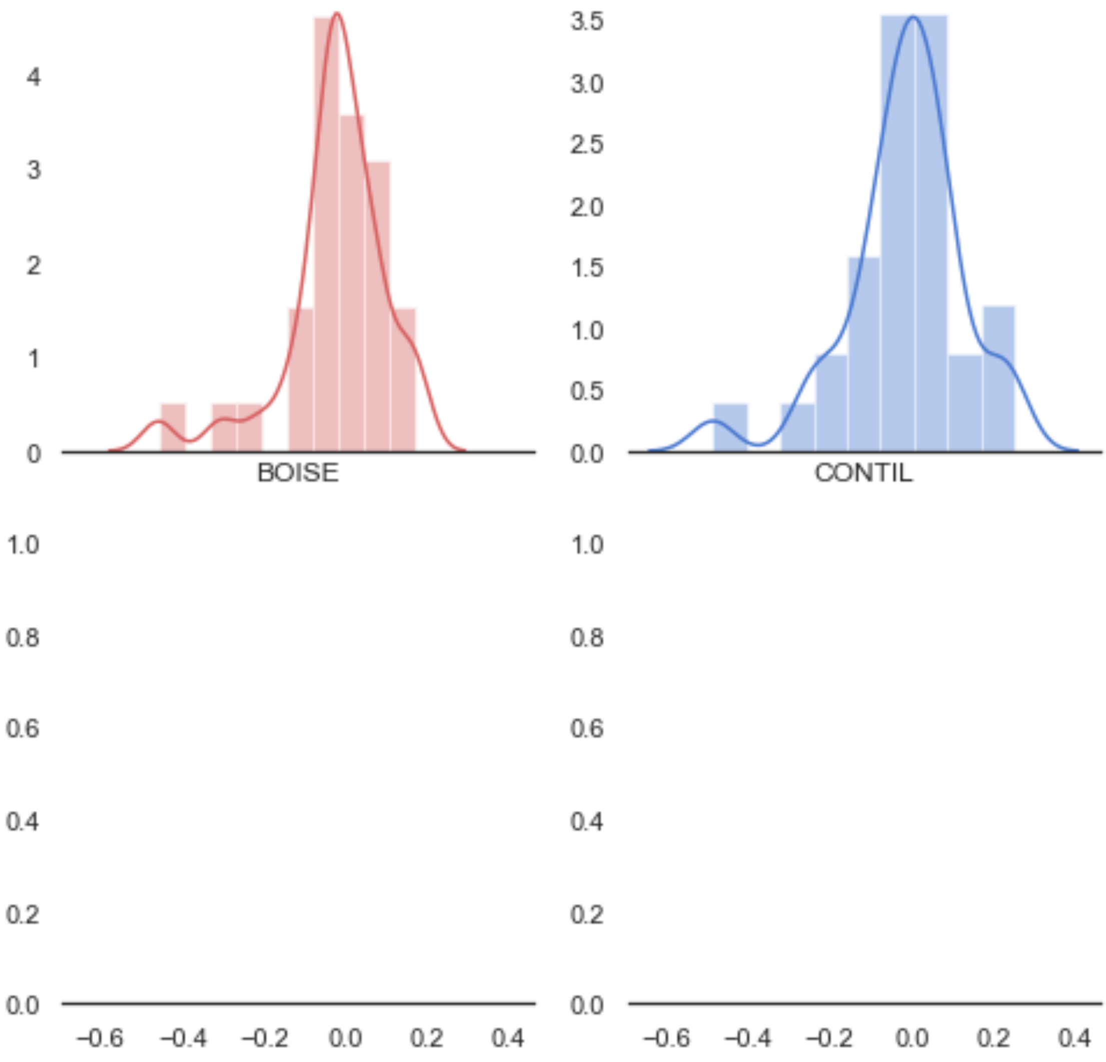
```
0.12649389666479696
```

In [99]:

```
sns.set(style="white", palette="muted", color_codes=True)

f, axes = plt.subplots(2, 2, figsize=(8, 8), sharex=True)
sns.despine(left=True)

sns.distplot(df3['BOISE'], color="r", ax=axes[0,0])
sns.distplot(df3['CONTIL'], color="b", ax=axes[0,1])
```

Out[99]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c1f76
b128>
```



In [100]:

```
df1.var()/df1.mean()
```

Out[100]:

```
BOISE       0.604939
CITCRP      0.576942
CONED       0.135915
CONTIL    -14.960264
DATGEN      2.679905
DEC         0.521622
DELTA       0.852008
GENMIL      0.265865
GERBER      0.485193
IBM         0.375298
MARKET      0.346923
MOBIL       0.422214
MOTOR       0.442681
PANAM       2.697942
PSNH       -9.911186
RKFREE      0.000681
TANDY       0.659466
TEXACO      0.585124
WEYER       0.816287
POIL        2.842325
FRBIND      0.873288
CPI         7.680525
GPU         1.956240
DOW         0.453628
DUPONT      0.317565
GOLD        0.660820
CONOCO      0.455283
dtype: float64
```

In [101]:

```
formula ='CONTIL ~ MARKET + RINF + GIND + ROIL'
results = smf.ols(formula, df4).fit()
print(results.summary())
```

```
                            OLS Regression Results
==================================================
============================
Dep. Variable:                 CONTIL   R-squared:
0.124
Model:                            OLS   Adj. R-squ
ared:                   0.093
Method:                 Least Squares   F-statisti
c:                      3.988
Date:                Wed, 18 Dec 2019   Prob (F-st
```

```
atistic):                  0.00460
Time:                           14:39:43   Log-Likeli
hood:                       63.586
No. Observations:                    118   AIC:
-117.2
Df Residuals:                        113   BIC:
-103.3
Df Model:                              4
Covariance Type:               nonrobust
==================================================
============================
                 coef    std err          t      P
>|t|      [0.025      0.975]
--------------------------------------------------
----------------------------
Intercept     -0.0096      0.024     -0.399      0
.691      -0.057       0.038
MARKET         0.6962      0.196      3.550      0
.001       0.308       1.085
RINF           0.5381      3.769      0.143      0
.887      -6.929       8.005
GIND          -1.6486      1.440     -1.145      0
.255      -4.502       1.205
ROIL           0.1740      0.265      0.657      0
.512      -0.351       0.699
==================================================
============================
Omnibus:                         103.711   Durbin-Wat
son:                         2.058
Prob(Omnibus):                     0.000   Jarque-Ber
a (JB):                   2129.891
Skew:                              2.634   Prob(JB):
0.00
Kurtosis:                         23.136   Cond. No.
284.
==================================================
============================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
```

In [102]:

```
hypotheses = 'GIND=0, ROIL=0'
f_test=results.f_test(hypotheses)
print(f_test)
```

```
<F test: F=array([[0.84148198]]), p=0.433754222009
85014, df_denom=113, df_num=2>
```

In [103]:

```
formula ='CONTIL ~ MARKET + RINF + ROIL'
results = smf.ols(formula, df4).fit()
print(results.summary())
```

```
                            OLS Regression Results
==================================================
============================
Dep. Variable:                 CONTIL   R-squared:
0.114
Model:                            OLS   Adj. R-squ
ared:                     0.090
Method:                 Least Squares   F-statisti
c:                        4.867
Date:                Wed, 18 Dec 2019   Prob (F-st
atistic):               0.00319
Time:                        14:39:43   Log-Likeli
hood:                    62.906
No. Observations:                 118   AIC:
-117.8
Df Residuals:                     114   BIC:
-106.7
Df Model:                           3
Covariance Type:            nonrobust
==================================================
============================
                 coef    std err          t      P
>|t|      [0.025      0.975]
--------------------------------------------------
----------------------------
Intercept     -0.0142      0.024     -0.595      0
.553      -0.061       0.033
MARKET         0.7163      0.196      3.662      0
.000       0.329       1.104
RINF           0.6614      3.772      0.175      0
.861      -6.812       8.135
ROIL           0.1616      0.265      0.610      0
.543      -0.363       0.686
```

```
==================================================
============================
Omnibus:                      106.538   Durbin-Wat
son:                       2.069
Prob(Omnibus):                  0.000   Jarque-Ber
a (JB):                 2279.411
Skew:                           2.729   Prob(JB):
0.00
Kurtosis:                      23.828   Cond. No.
284.
==================================================
============================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
```

In [104]:

```
formula ='CONTIL ~ MARKET + RINF + ROIL + GIND'
results = smf.ols(formula, df4).fit()
hypotheses = 'GIND'
wald_0 = results.wald_test(hypotheses)
print('H0:', hypotheses)
print(wald_0)
```

```
H0: GIND
<F test: F=array([[1.31001863]]), p=0.254809857978
58324, df_denom=113, df_num=1>
```

In [105]:

```
df1.head(100)
```

Out[105]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA
Date							
1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0.028
1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0.033
1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0.070
1978-							

04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0.150
1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0.031
1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0.023
1978-07-01	0.140	0.032	0.011	0.066	0.143	0.107	0.185
1978-08-01	0.078	0.088	0.024	0.033	0.026	-0.017	-0.021
1978-09-01	-0.059	0.011	0.048	-0.013	-0.031	-0.037	-0.081
1978-10-01	-0.118	-0.071	-0.067	-0.123	-0.085	-0.077	-0.153
1978-11-01	-0.060	-0.005	0.035	-0.038	0.044	0.064	0.055
1978-12-01	0.067	-0.019	0.005	0.047	0.034	0.117	-0.023
1979-01-01	0.168	0.043	0.076	-0.024	-0.008	-0.012	-0.054
1979-02-01	-0.032	-0.082	-0.011	-0.020	-0.015	-0.066	-0.060
1979-03-01	0.178	0.026	0.000	0.043	0.171	0.088	0.098
1979-04-01	-0.043	0.000	-0.057	0.064	0.009	0.005	-0.056
1979-05-01	-0.026	0.022	0.032	0.005	-0.045	-0.028	0.063
1979-06-01	0.057	0.095	0.066	0.092	0.019	0.059	-0.006
1979-07-01	0.047	-0.075	0.015	-0.034	-0.059	0.009	0.075
1979-08-01	0.038	0.065	-0.021	0.058	0.078	0.140	0.021
1979-09-01	0.050	-0.017	0.000	-0.033	-0.031	-0.027	-0.026
1979-10-01	-0.151	-0.125	-0.049	-0.136	-0.246	-0.010	-0.147
1979-11-01	-0.004	0.030	0.109	0.081	0.062	0.095	0.063

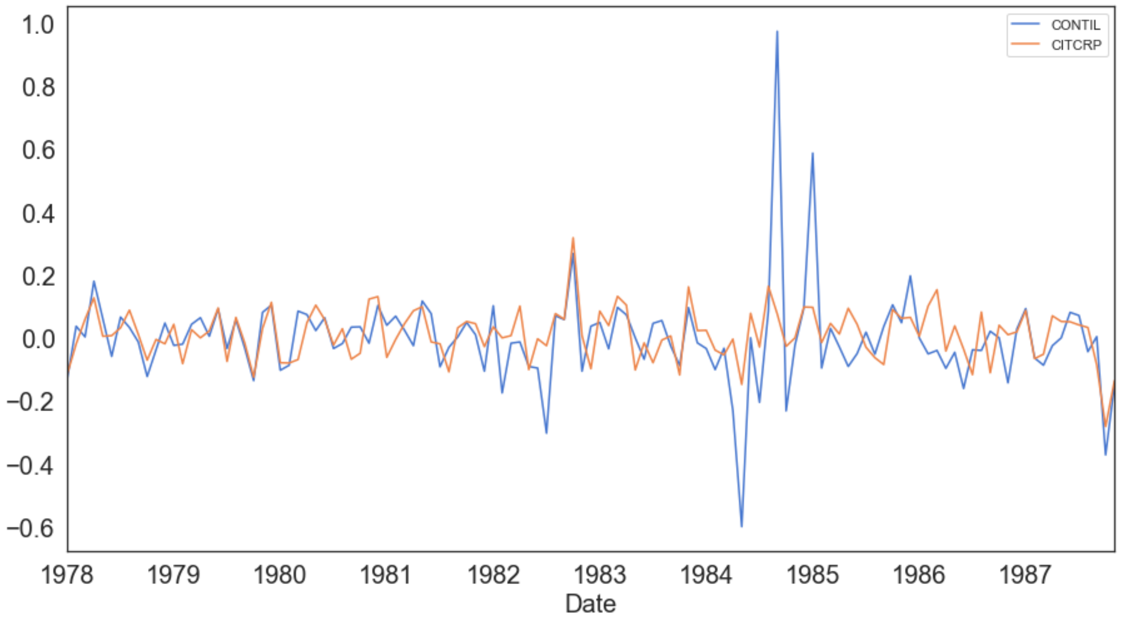
1979-12-01	0.042	0.113	0.005	0.104	0.021	0.018	0.020
1980-01-01	0.107	-0.079	-0.039	-0.103	0.157	0.058	0.022
1980-02-01	-0.070	-0.080	-0.061	-0.087	0.043	0.034	-0.093
1980-03-01	-0.138	-0.069	0.006	0.085	-0.094	-0.182	-0.031
1980-04-01	0.042	0.048	0.140	0.074	0.027	0.047	-0.018
1980-05-01	0.109	0.104	0.043	0.023	-0.043	0.016	0.144
1980-06-01	0.068	0.058	0.040	0.064	0.108	0.021	0.010
...	...	...	...	...	...	...	...
1983-11-01	0.147	0.162	-0.025	0.096	-0.014	0.065	0.120
1983-12-01	-0.012	0.023	0.005	-0.016	0.068	0.034	-0.028
1984-01-01	-0.054	0.024	0.005	-0.034	0.117	0.208	-0.013
1984-02-01	-0.088	-0.039	-0.069	-0.101	0.027	-0.024	-0.117
1984-03-01	0.079	-0.054	0.055	-0.033	0.056	0.057	0.065
1984-04-01	0.012	-0.004	0.031	-0.231	0.089	0.053	-0.085
1984-05-01	-0.172	-0.148	0.021	-0.600	-0.094	-0.071	-0.070
1984-06-01	0.025	0.078	0.020	0.000	0.056	-0.043	-0.012
1984-07-01	0.015	-0.029	0.054	-0.205	-0.061	-0.009	0.045
1984-08-01	0.177	0.164	0.029	0.086	0.312	0.159	0.040
1984-09-01	-0.056	0.076	0.051	0.974	-0.132	-0.025	0.008
1984-10-01	0.053	-0.027	0.019	-0.232	0.047	0.093	0.161

1984-11-01	-0.038	0.000	0.004	-0.023	0.019	0.006	-0.026
1984-12-01	0.068	0.098	0.084	0.095	0.096	0.070	0.156
1985-01-01	0.046	0.097	-0.021	0.587	0.215	0.084	-0.010
1985-02-01	-0.059	-0.015	0.034	-0.096	-0.210	-0.067	0.087
1985-03-01	-0.029	0.046	0.057	0.030	-0.195	-0.071	-0.003
1985-04-01	0.010	0.012	0.019	-0.029	-0.157	-0.050	-0.123
1985-05-01	0.158	0.094	0.098	-0.091	-0.078	0.057	0.179
1985-06-01	0.086	0.043	0.046	-0.050	0.060	-0.101	0.021
1985-07-01	-0.026	-0.030	-0.084	0.018	0.043	0.080	0.008
1985-08-01	0.011	-0.063	0.043	-0.052	-0.006	0.032	-0.066
1985-09-01	-0.095	-0.085	-0.032	0.036	0.000	0.036	-0.112
1985-10-01	-0.035	0.090	0.066	0.105	0.032	0.040	-0.083
1985-11-01	0.088	0.062	0.032	0.048	0.109	0.073	0.020
1985-12-01	0.064	0.065	0.082	0.197	0.023	0.095	0.030
1986-01-01	0.032	0.005	0.022	0.000	-0.055	0.162	0.122
1986-02-01	0.093	0.101	0.048	-0.051	-0.044	0.093	-0.055
1986-03-01	0.066	0.153	0.021	-0.040	-0.043	-0.063	0.076
1986-04-01	-0.013	-0.042	-0.006	-0.097	0.061	0.119	0.059

100 rows × 28 columns

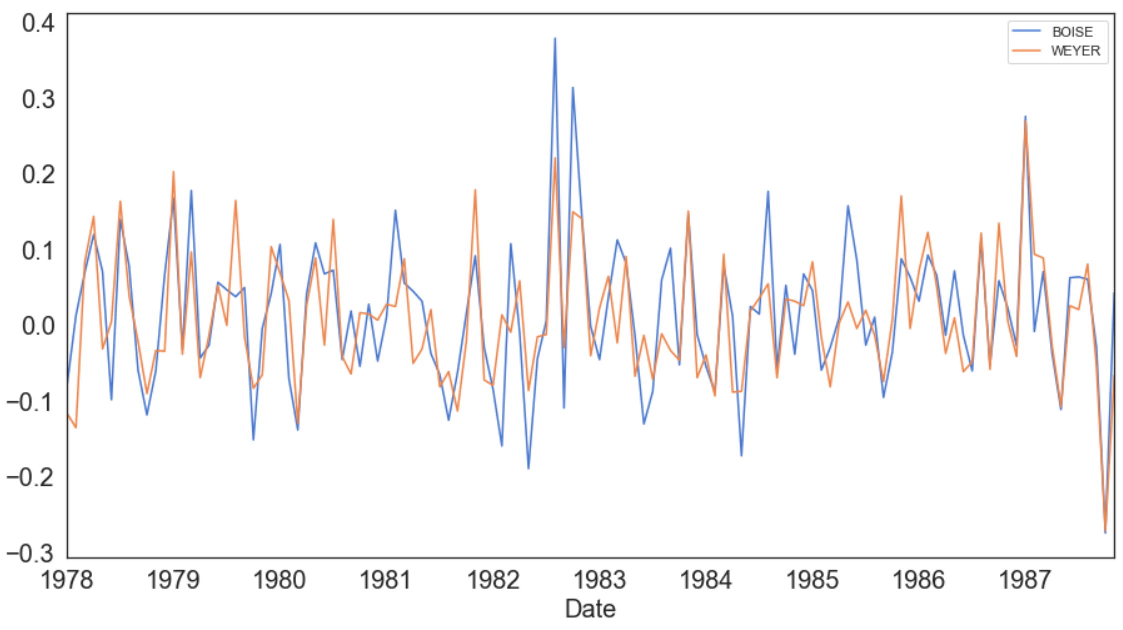
In [106]:

```
df1[['CONTIL','CITCRP']].plot(figsize=(15,8),linewidth=1.5,fon
tsize=20)
plt.xlabel('Date',fontsize=20);
```



In [107]:

```
df1[['BOISE','WEYER']].plot(figsize=(15,8),linewidth=1.5,fonts
ize=20)
plt.xlabel('Date',fontsize=20);
```



In [108]:

```
formula = 'Q("CONTIL") ~ Q("MARKET")'
results = smf.ols(formula, df3).fit()
print(results.summary())
```

```
                           OLS Regression Resul
ts
================================================
===============================
Dep. Variable:             Q("CONTIL")   R-square
d:                         0.500
Model:                             OLS   Adj. R-s
quared:                    0.483
Method:                  Least Squares   F-statis
tic:                       29.04
Date:                 Wed, 18 Dec 2019   Prob (F-
statistic):             8.60e-06
Time:                         14:39:49   Log-Like
lihood:                   26.925
No. Observations:                   31   AIC:
-49.85
Df Residuals:                       29   BIC:
-46.98
Df Model:                            1
Covariance Type:             nonrobust
================================================
===============================
                  coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
-------------------------------
Intercept      -0.0043      0.019     -0.227
0.822      -0.043       0.035
Q("MARKET")     0.9517      0.177      5.389
0.000       0.590       1.313
================================================
===============================
Omnibus:                         9.455   Durbin-W
atson:                     2.064
Prob(Omnibus):                   0.009   Jarque-B
era (JB):                 10.072
Skew:                            0.777   Prob(JB)
:                        0.00650
Kurtosis:                        5.320   Cond. No
.                           9.37
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
```

In [109]:

```
hypotheses = 'Intercept=0'
Simple_ttest_Ols(results, hypotheses, alternative='larger', le
vel_of_sig = 0.05)
```

```
We accept the null hypothesis: Intercept=0 with a
5.0 % significance level
```

In [110]:

```
hypotheses = 'Q("MARKET")=0'
Simple_ttest_Ols(results, hypotheses, alternative='smaller', l
evel_of_sig = 0.05)
```

```
We accept the null hypothesis: Q("MARKET")=0 with
a 5.0 % significance level
```

In [111]:

```
formula = 'Q("CITCRP") ~ Q("MARKET")'
results = smf.ols(formula, df).fit()
print(results.summary())
```

```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:            Q("CITCRP")   R-square
d:                        0.316
Model:                            OLS   Adj. R-s
quared:                   0.310
Method:                 Least Squares   F-statis
tic:                      53.95
Date:                Wed, 18 Dec 2019   Prob (F-
statistic):            2.99e-11
Time:                        14:39:49   Log-Like
lihood:                  153.04
No. Observations:                 119   AIC:
-302.1
Df Residuals:                     117   BIC:
-296.5
Df Model:                           1
Covariance Type:            nonrobust
================================================
===============================
                 coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
-------------------------------
Intercept       0.0024      0.006      0.385
0.701      -0.010       0.015
Q("MARKET")     0.6664      0.091      7.345
0.000       0.487       0.846
================================================
==============================
Omnibus:                        1.688   Durbin-W
atson:                    1.820
Prob(Omnibus):                  0.430   Jarque-B
era (JB):                 1.213
Skew:                           0.110   Prob(JB)
:                         0.545
Kurtosis:                       3.443   Cond. No
.                          14.7
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
atrix of the errors is correctly specified.
```

In [112]:

```
hypotheses = 'Q("MARKET")=0'
Simple_ttest_Ols(results, hypotheses, alternative='smaller', l
evel_of_sig = 0.05)
```

```
We accept the null hypothesis: Q("MARKET")=0 with
a 5.0 % significance level
```

In [113]:

```
hypotheses = 'Intercept=0'
Simple_ttest_Ols(results, hypotheses, alternative='larger', le
vel_of_sig = 0.05)
```

```
We accept the null hypothesis: Intercept=0 with a
5.0 % significance level
```

In [114]:

```
formula = 'Q("CONTIL") ~ Q("MARKET")'

results = smf.ols(formula, df1).fit()
print(results.summary())
```

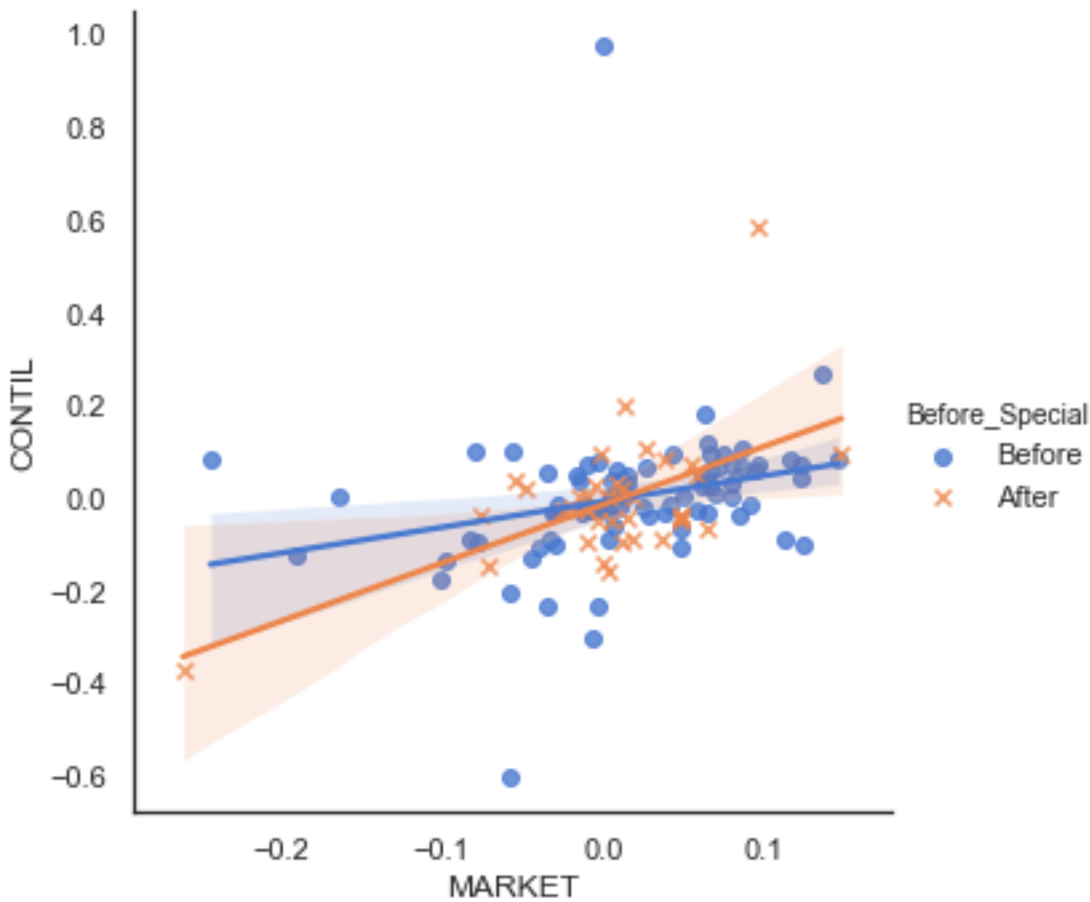
```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:           Q("CONTIL")   R-square
d:                       0.111
Model:                           OLS   Adj. R-s
quared:                  0.104
Method:                Least Squares   F-statis
tic:                     14.65
Date:               Wed, 18 Dec 2019   Prob (F-
statistic):            0.000209
Time:                       14:39:49   Log-Like
lihood:                 63.430
No. Observations:                119   AIC:
-122.9
Df Residuals:                    117   BIC:
-117.3
Df Model:                          1
Covariance Type:           nonrobust
================================================
===============================
                  coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
-------------------------------
Intercept      -0.0115      0.013     -0.858
0.393      -0.038       0.015
Q("MARKET")     0.7375      0.193      3.828
0.000       0.356       1.119
================================================
==============================
Omnibus:                     106.122   Durbin-W
atson:                   2.068
Prob(Omnibus):                 0.000   Jarque-B
era (JB):             2224.184
Skew:                          2.697   Prob(JB)
:                         0.00
Kurtosis:                     23.481   Cond. No
.                         14.7
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
atrix of the errors is correctly specified.
```

In [115]:

```
RF.Chow_Test(df1, y='CONTIL', x='MARKET', special_date='1984-1
1-01')
```

Out[115]:

```
(2.5869412758378107, 0.07951319460600349)
```



In [116]:

```
formula = 'Q("BOISE") ~ Q("MARKET")'

results = smf.ols(formula, df1).fit()
print(results.summary())
```

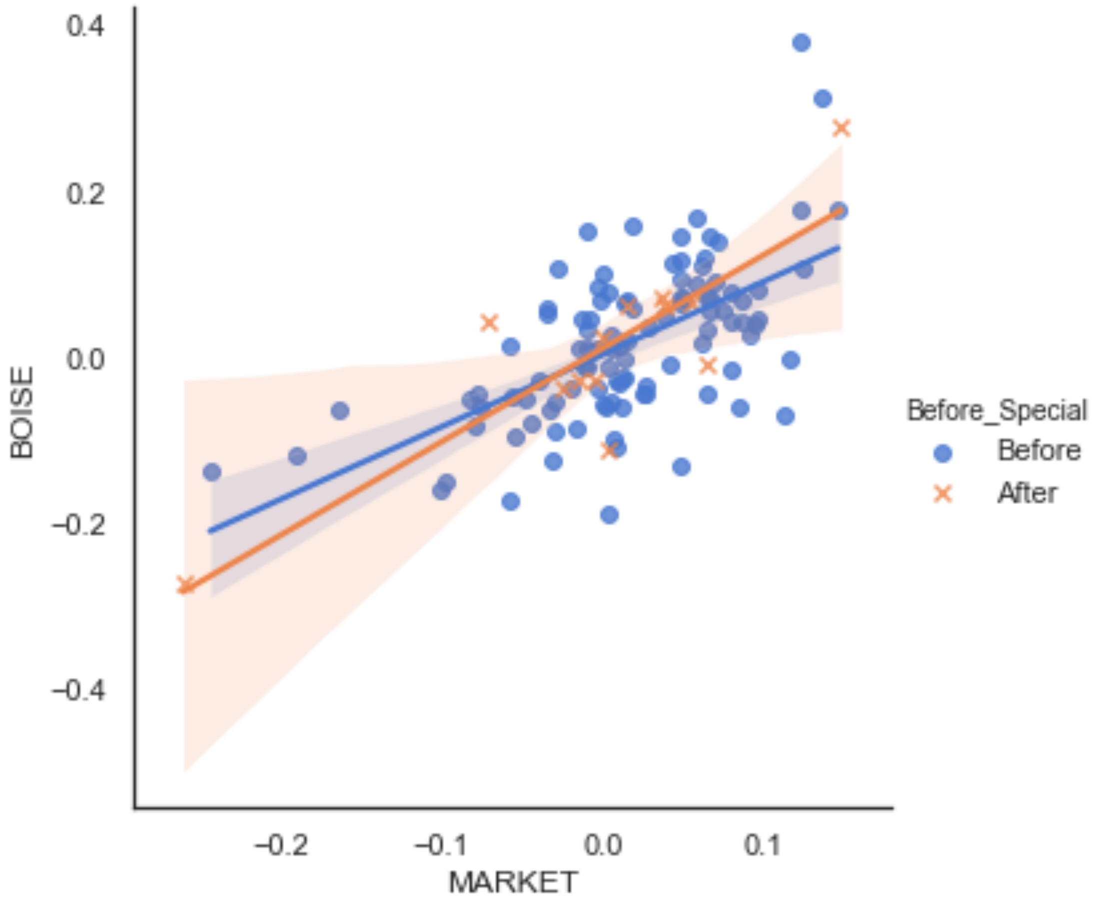
```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:              Q("BOISE")   R-square
d:                        0.422
Model:                             OLS   Adj. R-s
quared:                   0.417
Method:                  Least Squares   F-statis
tic:                      85.45
Date:                 Wed, 18 Dec 2019   Prob (F-
statistic):            1.31e-15
Time:                         14:39:50   Log-Like
lihood:                  141.64
No. Observations:                  119   AIC:
-279.3
Df Residuals:                      117   BIC:
-273.7
Df Model:                            1
Covariance Type:             nonrobust
================================================
===============================
                  coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
-------------------------------
Intercept       0.0032      0.007      0.456
0.649      -0.011       0.017
Q("MARKET")     0.9230      0.100      9.244
0.000       0.725       1.121
================================================
==============================
Omnibus:                         4.937   Durbin-W
atson:                    2.183
Prob(Omnibus):                   0.085   Jarque-B
era (JB):                 5.734
Skew:                            0.215   Prob(JB)
:                        0.0569
Kurtosis:                        3.986   Cond. No
.                          14.7
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
atrix of the errors is correctly specified.
```

In [117]:

```
RF.Chow_Test(df1, y='BOISE', x='MARKET', special_date='1986-11
-01')
```

Out[117]:

```
(1.0439225070169136, 0.35529933592969426)
```



In [118]:

```
formula = 'Q("CONTIL") ~ Q("MARKET")'
results = smf.ols(formula, df1).fit()
print(results.summary())
```

```
                              OLS Regression Resul
ts
================================================
==============================
Dep. Variable:            Q("CONTIL")   R-square
d:                         0.111
Model:                            OLS   Adj. R-s
quared:                    0.104
Method:                 Least Squares   F-statis
tic:                       14.65
Date:                Wed, 18 Dec 2019   Prob (F-
statistic):             0.000209
Time:                        14:39:51   Log-Like
lihood:                   63.430
No. Observations:                 119   AIC:
-122.9
Df Residuals:                     117   BIC:
-117.3
Df Model:                           1
Covariance Type:            nonrobust
================================================
===============================
                  coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
-------------------------------
Intercept      -0.0115      0.013     -0.858
0.393      -0.038       0.015
Q("MARKET")     0.7375      0.193      3.828
0.000       0.356       1.119
================================================
==============================
Omnibus:                      106.122   Durbin-W
atson:                     2.068
Prob(Omnibus):                  0.000   Jarque-B
era (JB):               2224.184
Skew:                           2.697   Prob(JB)
:                           0.00
Kurtosis:                      23.481   Cond. No
.                           14.7
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
atrix of the errors is correctly specified.
```

In [119]:

```
formula = 'Q("BOISE") ~ Q("MARKET")'
results = smf.ols(formula, df1).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             Q("BOISE")   R-squared:
0.422
Model:                            OLS   Adj. R-squ
ared:                   0.417
Method:                 Least Squares   F-statisti
c:                      85.45
Date:                Wed, 18 Dec 2019   Prob (F-st
atistic):            1.31e-15
Time:                        14:39:51   Log-Likeli
hood:                  141.64
No. Observations:                 119   AIC:
-279.3
Df Residuals:                     117   BIC:
-273.7
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                  coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept       0.0032      0.007      0.456
0.649      -0.011       0.017
Q("MARKET")     0.9230      0.100      9.244
0.000       0.725       1.121
==============================================================================
Omnibus:                        4.937   Durbin-Wat
son:                    2.183
Prob(Omnibus):                  0.085   Jarque-Ber
a (JB):                 5.734
Skew:                           0.215   Prob(JB):
0.0569
Kurtosis:                       3.986   Cond. No.
14.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance mat
rix of the errors is correctly specified.
```

In [120]:

```
df.head(100)
```

Out[120]:

	Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DI
0	1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	
1	1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	
2	1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	
3	1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN	
4	1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN	
5	1976-06-01	NaN	NaN	NaN	NaN	NaN	NaN	
6	1976-07-01	NaN	NaN	NaN	NaN	NaN	NaN	
7	1976-08-01	NaN	NaN	NaN	NaN	NaN	NaN	
8	1976-09-01	NaN	NaN	NaN	NaN	NaN	NaN	
9	1976-10-01	NaN	NaN	NaN	NaN	NaN	NaN	
10	1976-11-01	NaN	NaN	NaN	NaN	NaN	NaN	
11	1976-12-01	NaN	NaN	NaN	NaN	NaN	NaN	
12	1977-01-01	NaN	NaN	NaN	NaN	NaN	NaN	
13	1977-02-01	NaN	NaN	NaN	NaN	NaN	NaN	
14	1977-03-01	NaN	NaN	NaN	NaN	NaN	NaN	
15	1977-04-01	NaN	NaN	NaN	NaN	NaN	NaN	
16	1977-05-01	NaN	NaN	NaN	NaN	NaN	NaN	

17	1977-06-01	NaN	NaN	NaN	NaN	NaN	NaN	
18	1977-07-01	NaN	NaN	NaN	NaN	NaN	NaN	
19	1977-08-01	NaN	NaN	NaN	NaN	NaN	NaN	
20	1977-09-01	NaN	NaN	NaN	NaN	NaN	NaN	
21	1977-10-01	NaN	NaN	NaN	NaN	NaN	NaN	
22	1977-11-01	NaN	NaN	NaN	NaN	NaN	NaN	
23	1977-12-01	NaN	NaN	NaN	NaN	NaN	NaN	
24	1978-01-01	-0.079	-0.115	-0.079	-0.129	-0.084	-0.100	-0
25	1978-02-01	0.013	-0.019	-0.003	0.037	-0.097	-0.063	-0
26	1978-03-01	0.070	0.059	0.022	0.003	0.063	0.010	0
27	1978-04-01	0.120	0.127	-0.005	0.180	0.179	0.165	0
28	1978-05-01	0.071	0.005	-0.014	0.061	0.052	0.038	-0
29	1978-06-01	-0.098	0.007	0.034	-0.059	-0.023	-0.021	0
...	...	...	...	...	...	...	...	
70	1981-11-01	0.092	0.045	0.038	0.010	0.093	-0.065	-0
71	1981-12-01	-0.029	-0.028	-0.008	-0.106	-0.083	-0.060	-0
72	1982-01-01	-0.084	0.035	0.042	0.102	-0.002	0.027	0
73	1982-02-01	-0.159	0.000	0.036	-0.175	-0.152	-0.049	0
74	1982-03-01	0.108	0.007	0.022	-0.017	-0.302	-0.104	0
75	1982-04-01	-0.009	0.101	0.050	-0.013	0.047	0.054	-0
76	1982-05-01	-0.189	-0.101	0.016	-0.091	-0.180	-0.056	0

77	1982-06-01	-0.044	-0.003	-0.024	-0.096	-0.060	-0.073	0
78	1982-07-01	0.006	-0.025	-0.032	-0.303	-0.054	-0.055	-0
79	1982-08-01	0.379	0.077	0.133	0.070	0.216	0.273	0
80	1982-09-01	-0.109	0.059	0.039	0.058	-0.165	-0.061	-0
81	1982-10-01	0.314	0.318	-0.050	0.268	0.528	0.133	0
82	1982-11-01	0.145	0.007	-0.011	-0.106	0.003	0.175	0
83	1982-12-01	-0.001	-0.098	0.123	0.037	0.053	-0.052	0
84	1983-01-01	-0.045	0.085	-0.012	0.049	0.208	0.225	0
85	1983-02-01	0.037	0.039	0.060	-0.035	0.237	-0.010	0
86	1983-03-01	0.113	0.132	0.048	0.097	0.040	0.034	-0
87	1983-04-01	0.082	0.104	0.045	0.073	0.079	-0.060	-0
88	1983-05-01	-0.014	-0.102	-0.012	0.000	-0.114	-0.052	-0
89	1983-06-01	-0.130	-0.016	0.000	-0.068	-0.042	0.075	0
90	1983-07-01	-0.087	-0.079	0.017	0.046	0.173	-0.142	-0
91	1983-08-01	0.060	-0.007	-0.023	0.055	0.053	0.007	0
92	1983-09-01	0.102	0.006	0.087	-0.026	0.090	-0.005	0
93	1983-10-01	-0.052	-0.118	0.101	-0.088	-0.069	-0.364	0
94	1983-11-01	0.147	0.162	-0.025	0.096	-0.014	0.065	0
95	1983-12-01	-0.012	0.023	0.005	-0.016	0.068	0.034	-0
96	1984-01-01	-0.054	0.024	0.005	-0.034	0.117	0.208	-0
	1984-							

97	02-01	-0.088	-0.039	-0.069	-0.101	0.027	-0.024	-0
98	1984-03-01	0.079	-0.054	0.055	-0.033	0.056	0.057	0
99	1984-04-01	0.012	-0.004	0.031	-0.231	0.089	0.053	-0

100 rows × 28 columns

In [121]:

```
static,pvalue=ss.jarque_bera(df3['CONTIL'])
```

In [122]:

```
static
```

Out[122]:

```
9.75995360623815
```

In [123]:

```
pvalue
```

Out[123]:

```
0.007597190256651287
```

In [124]:

```
print('The test statistic is given by {} and the P-value is gi
ven by {}'.format(static, pvalue))
```

```
The test statistic is given by 9.75995360623815 an
d the P-value is given by 0.007597190256651287
```

In [125]:

```
static,pvalue=ss.kstest(df3['CONTIL'], 'norm')
print('The test statistic is given by {} and the P-value is gi
ven by {}'.format(static, pvalue))
```

```
The test statistic is given by 0.400949578020683З
and the P-value is given by 5.191736905658404e-05
```

In [126]:

```
static,pvalue=statsmodels.stats._adnorm.normal_ad(df3['CONTIL'
])
print('The test statistic is given by {} and the P-value is gi
ven by {}'.format(static, pvalue))
```

```
The test statistic is given by 0.6431851643635831
and the P-value is given by 0.08479062672413287
```

In [127]:

```
static,pvalue=ss.shapiro(df3['CONTIL'])
print('The test statistic is given by {} and the P-value is gi
ven by {}'.format(static, pvalue))
```

```
The test statistic is given by 0.9337706565856934
and the P-value is given by 0.0555806644260883
```

In [128]:

```
static,pvalue=ss.jarque_bera(df3['BOISE'])
```

In [129]:

```
static
```

Out[129]:

```
20.667853044913137
```

In [130]:

```
pvalue
```

Out[130]:

```
3.251118017322252e-05
```

In [131]:

```
static,pvalue=ss.kstest(df3['BOISE'], 'norm')
print('The test statistic is given by {} and the P-value is gi
ven by {}'.format(static, pvalue))
```

```
The test statistic is given by 0.43217871960473353
and the P-value is given by 8.857278685089354e-06
```

In [132]:

```
static,pvalue=statsmodels.stats._adnorm.normal_ad(df3['BOISE']
)
print('The test statistic is given by {} and the P-value is gi
ven by {}'.format(static, pvalue))
```

```
The test statistic is given by 1.1523255888499122
and the P-value is given by 0.004367941102311427
```

In [133]:

```
static,pvalue=ss.shapiro(df3['BOISE'])
print('The test statistic is given by {} and the P-value is gi
ven by {}'.format(static, pvalue))
```

```
The test statistic is given by 0.8807262182235718
and the P-value is given by 0.0024624823126941 92
```

In [134]:

```
df.head()
```

Out[134]:

	Date	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELT
0	1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN	Na

5 rows × 28 columns

In [135]:

```
df = pd.read_excel('Data_For_Analysis.xlsx')
df.set_index('Date', inplace=True)

df.columns
# df=df.dropna()
```

Out[135]:

```
Index(['BOISE', 'CITCRP', 'CONED', 'CONTIL', 'DATG
EN', 'DEC', 'DELTA',
       'GENMIL', 'GERBER', 'IBM', 'MARKET', 'MOBIL
', 'MOTOR', 'PANAM', 'PSNH',
       'RKFREE', 'TANDY', 'TEXACO', 'WEYER', 'POIL
', 'FRBIND', 'CPI', 'GPU',
       'DOW', 'DUPONT', 'GOLD', 'CONOCO'],
      dtype='object')
```

In [202]:

```
MainDF=df[['CONTIL','IBM','MARKET','RKFREE','CPI','POIL','FRBI
ND']]
MainDF=MainDF.dropna()
MainDF.head(100)
```

Out[202]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND
Date							
1978-01-01	-0.129	-0.029	-0.045	0.00487	187.2	8.68	138.8
1978-02-01	0.037	-0.043	0.010	0.00494	188.4	8.84	139.2
1978-03-01	0.003	-0.063	0.050	0.00526	189.8	8.80	140.9
1978-04-01	0.180	0.130	0.063	0.00491	191.5	8.82	143.2
1978-05-01	0.061	-0.018	0.067	0.00513	193.3	8.81	143.9
1978-06-01	-0.059	-0.004	0.007	0.00527	195.3	9.05	144.9
1978-07-01	0.066	0.092	0.071	0.00528	196.7	8.96	146.1

1978-08-01	0.033	0.049	0.079	0.00607	197.8	8.05	147.1
1978-09-01	-0.013	-0.051	0.002	0.00645	199.3	9.15	147.8
1978-10-01	-0.123	-0.046	-0.189	0.00685	200.9	9.17	148.6
1978-11-01	-0.038	0.031	0.084	0.00719	202.0	9.20	149.5
1978-12-01	0.047	0.108	0.015	0.00690	203.3	9.47	150.4
1979-01-01	-0.024	0.034	0.058	0.00761	204.7	9.46	152.0
1979-02-01	-0.020	-0.017	0.011	0.00761	207.1	9.69	152.5
1979-03-01	0.043	0.052	0.123	0.00769	209.1	9.83	153.5
1979-04-01	0.064	-0.004	0.026	0.00764	211.5	10.33	151.1
1979-05-01	0.005	-0.022	0.014	0.00772	214.1	10.71	152.7
1979-06-01	0.092	-0.035	0.075	0.00715	216.6	11.70	153.0
1979-07-01	-0.034	-0.049	-0.013	0.00728	218.9	13.39	153.0
1979-08-01	0.058	0.016	0.095	0.00789	221.1	14.00	152.1
1979-09-01	-0.033	-0.032	0.039	0.00802	223.4	14.57	152.7
1979-10-01	-0.136	-0.079	-0.097	0.00913	225.4	15.11	152.7
1979-11-01	0.081	0.060	0.116	0.00819	227.5	15.52	152.3
1979-12-01	0.104	-0.013	0.086	0.00747	229.9	17.03	152.5
1980-01-01	-0.103	0.066	0.124	0.00883	233.2	17.86	152.7
1980-02-01	-0.087	-0.062	0.112	0.01073	236.4	18.81	152.6
1980-							

03-01	0.085	-0.122	-0.243	0.01181	239.8	19.34	152.1
1980-04-01	0.074	-0.016	0.080	0.00753	242.5	20.29	148.3
1980-05-01	0.023	0.025	0.062	0.00630	244.9	21.01	144.0
1980-06-01	0.064	0.061	0.086	0.00503	247.6	21.53	141.5
...	...	...	...	...	...	...	...
1983-11-01	0.096	-0.066	0.066	0.00683	303.1	26.09	155.3
1983-12-01	-0.016	0.039	-0.012	0.00693	303.5	25.88	156.2
1984-01-01	-0.034	-0.065	-0.029	0.00712	305.4	25.93	158.5
1984-02-01	-0.101	-0.026	-0.030	0.00672	306.6	26.06	160.0
1984-03-01	-0.033	0.034	0.003	0.00763	307.3	26.05	160.8
1984-04-01	-0.231	-0.002	-0.003	0.00741	308.8	25.93	162.1
1984-05-01	-0.600	-0.044	-0.058	0.00627	309.7	26.00	162.8
1984-06-01	0.000	-0.019	0.005	0.00748	310.7	26.09	164.4
1984-07-01	-0.205	0.047	-0.058	0.00771	311.7	26.11	165.9
1984-08-01	0.086	0.127	0.146	0.00852	313.0	26.02	166.0
1984-09-01	0.974	0.004	0.000	0.00830	314.5	25.97	165.0
1984-10-01	-0.232	0.012	-0.035	0.00688	315.3	25.92	164.5
1984-11-01	-0.023	-0.023	-0.019	0.00602	315.3	25.44	165.2
1984-12-01	0.095	0.011	-0.001	0.00612	315.5	25.05	166.2
1985-01-01	0.587	0.108	0.097	0.00606	316.1	24.28	165.6

1985-02-01	-0.096	-0.009	0.012	0.00586	317.4	23.63	165.7
1985-03-01	0.030	-0.052	0.008	0.00650	318.8	23.88	166.1
1985-04-01	-0.029	-0.004	-0.010	0.00601	320.1	24.15	166.2
1985-05-01	-0.091	0.025	0.019	0.00512	321.3	24.18	166.2
1985-06-01	-0.050	-0.038	-0.003	0.00536	322.3	24.03	166.5
1985-07-01	0.018	0.062	0.012	0.00562	322.8	24.00	166.2
1985-08-01	-0.052	-0.028	0.005	0.00545	323.5	23.92	167.7
1985-09-01	0.036	-0.022	-0.055	0.00571	324.5	23.93	167.6
1985-10-01	0.105	0.048	0.026	0.00577	325.5	24.06	166.6
1985-11-01	0.048	0.085	0.059	0.00540	326.6	24.31	167.6
1985-12-01	0.197	0.113	0.013	0.00479	327.4	24.53	168.8
1986-01-01	0.000	-0.026	-0.009	0.00548	328.4	23.12	169.6
1986-02-01	-0.051	0.003	0.049	0.00523	327.5	17.65	168.4
1986-03-01	-0.040	0.004	0.048	0.00508	326.0	12.62	166.1
1986-04-01	-0.097	0.031	-0.009	0.00444	325.3	10.68	167.6

100 rows × 7 columns

In [137]:

```
MainDF.index
```

Out[137]:

```
DatetimeIndex(['1978-01-01', '1978-02-01', '1978-0
3-01', '1978-04-01',
               '1978-05-01', '1978-06-01', '1978-0
7-01', '1978-08-01',
               '1978-09-01', '1978-10-01',
               ...
               '1987-02-01', '1987-03-01', '1987-0
4-01', '1987-05-01',
               '1987-06-01', '1987-07-01', '1987-0
8-01', '1987-09-01',
               '1987-10-01', '1987-11-01'],
              dtype='datetime64[ns]', name='Date',
length=119, freq=None)
```

In [138]:

```
MainDF['RINF']=MainDF['CPI'].pct_change(1)
MainDF['GIND']=MainDF['FRBIND'].pct_change(1)
MainDF['real_POIL']=MainDF['POIL']/MainDF['CPI']
MainDF['ROIL']=MainDF['real_POIL'].pct_change(1)

MainDF.head()
```

Out[138]:

	MOBIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND
Date							
1978-01-01	-0.046	-0.029	-0.045	0.00487	187.2	8.68	138.8
1978-02-01	-0.017	-0.043	0.010	0.00494	188.4	8.84	139.2
1978-03-01	0.049	-0.063	0.050	0.00526	189.8	8.80	140.9
1978-04-01	0.077	0.130	0.063	0.00491	191.5	8.82	143.2
1978-05-01	-0.011	-0.018	0.067	0.00513	193.3	8.81	143.9

In [139]:

```
MainDF=MainDF.dropna()
MainDF.head()
```

Out[139]:

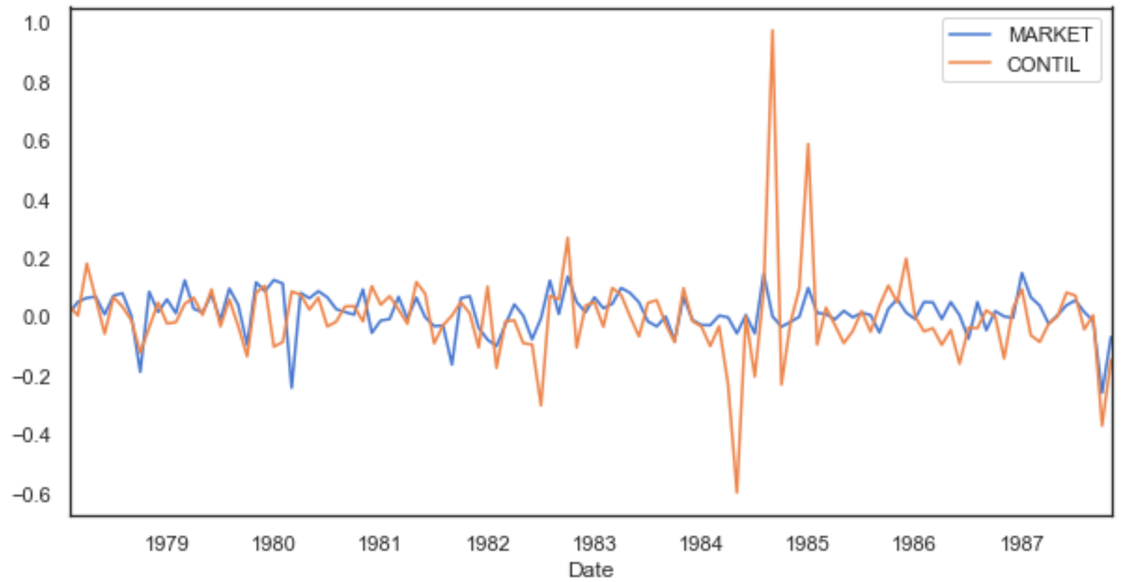
	MOBIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
Date								
1978-02-01	-0.017	-0.043	0.010	0.00494	188.4	8.84	139.2	0.0
1978-03-01	0.049	-0.063	0.050	0.00526	189.8	8.80	140.9	0.0
1978-04-01	0.077	0.130	0.063	0.00491	191.5	8.82	143.2	0.0
1978-05-01	-0.011	-0.018	0.067	0.00513	193.3	8.81	143.9	0.0
1978-06-01	-0.043	-0.004	0.007	0.00527	195.3	9.05	144.9	0.0

In [212]:

```
plt.rcParams["figure.figsize"] = [10,5]
MainDF[['MARKET', 'CONTIL']].plot()
```

Out[212]:

<matplotlib.axes._subplots.AxesSubplot at 0x1c20d7
e320>



In [217]:

```
    MainDF[MainDF['CONTIL']== MainDF['CONTIL'].max()]
```

Out[217]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
Date								
1984-09-01	0.974	0.004	0.0	0.0083	314.5	25.97	165.0	0.(

In [218]:

```
start_date = dt.datetime(1978,1,1)
end_date = dt.datetime(1980,1,1)

start_date2 = dt.datetime(1980,1,1)
end_date2 = dt.datetime(1985,1,1)

select = (MainDF.index>=start_date)*(MainDF.index<end_date)

select2 = (MainDF.index>=start_date2)*(MainDF.index<end_date2)

MainDF_first_period=MainDF[select]
MainDF_second_period=MainDF[select2]
```

In [219]:

```
MainDF_first_period.head()
```

Out[219]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
Date								
1978-02-01	0.037	-0.043	0.010	0.00494	188.4	8.84	139.2	0.(
1978-03-01	0.003	-0.063	0.050	0.00526	189.8	8.80	140.9	0.(
1978-04-01	0.180	0.130	0.063	0.00491	191.5	8.82	143.2	0.(
1978-05-01	0.061	-0.018	0.067	0.00513	193.3	8.81	143.9	0.(
1978-06-01	-0.059	-0.004	0.007	0.00527	195.3	9.05	144.9	0.(

In [220]:

```
MainDF_first_period.tail(2)
```

Out[220]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND
Date							
1979-11-01	0.081	0.060	0.116	0.00819	227.5	15.52	152.3
1979-12-01	0.104	-0.013	0.086	0.00747	229.9	17.03	152.5

In [145]:

```
MainDF_second_period.head(2)
```

Out[145]:

	MOBIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
Date								
1980-01-01	0.075	0.066	0.124	0.00883	233.2	17.86	152.7	0.0
1980-02-01	0.366	-0.062	0.112	0.01073	236.4	18.81	152.6	0.0

In [146]:

```
MainDF_second_period.tail(2)
```

Out[146]:

	MOBIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND
Date							
1984-11-01	-0.042	-0.023	-0.019	0.00602	315.3	25.44	165.2
1984-12-01	-0.052	0.011	-0.001	0.00612	315.5	25.05	166.2

In [221]:

```
formula = 'CONTIL ~ MARKET'
results_Mobil_period1 = smf.ols(formula, MainDF_first_period).
fit()
print(results_Mobil_period1.summary())
```

```
                            OLS Regression Resul
ts
================================================
==============================
Dep. Variable:                 CONTIL   R-square
d:                        0.512
Model:                            OLS   Adj. R-s
quared:                   0.489
Method:                 Least Squares   F-statis
tic:                      22.03
Date:                Wed, 18 Dec 2019   Prob (F-
statistic):             0.000124
Time:                        16:20:38   Log-Like
lihood:                  36.492
No. Observations:                  23   AIC:
-68.98
Df Residuals:                      21   BIC:
-66.71
Df Model:                           1
Covariance Type:            nonrobust
================================================
==============================
                 coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------
------------------------------
Intercept     -0.0088      0.012     -0.726
0.476      -0.034       0.016
MARKET         0.7533      0.161      4.693
0.000       0.420       1.087
================================================
==============================
Omnibus:                        4.224   Durbin-W
atson:                    2.335
Prob(Omnibus):                  0.121   Jarque-B
era (JB):                 2.320
Skew:                           0.636   Prob(JB)
:                         0.313
Kurtosis:                       3.895   Cond. No
.                          14.9
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
atrix of the errors is correctly specified.
```

In [222]:

```
formula = 'CONTIL ~ MARKET + RINF + GIND + ROIL'
results_Mobil_Secondmodel_period1 = smf.ols(formula, MainDF_fi
rst_period).fit()
print(results_Mobil_Secondmodel_period1.summary())
```

```
                            OLS Regression Resul
ts
===============================================================
==============================
Dep. Variable:                 CONTIL   R-square
d:                       0.526
Model:                            OLS   Adj. R-s
quared:                  0.420
Method:                 Least Squares   F-statis
tic:                     4.988
Date:                Wed, 18 Dec 2019   Prob (F-
statistic):            0.00696
Time:                        16:20:49   Log-Like
lihood:                 36.821
No. Observations:                  23   AIC:
-63.64
Df Residuals:                      18   BIC:
-57.97
Df Model:                           4
Covariance Type:            nonrobust
===============================================================
==============================
                 coef    std err          t
P>|t|      [0.025      0.975]
---------------------------------------------------------------
------------------------------
Intercept     -0.0475      0.062     -0.769
0.452      -0.177       0.082
MARKET         0.7556      0.173      4.374
0.000       0.393       1.119
RINF           3.9987      6.696      0.597
0.558     -10.069      18.066
GIND           0.4681      1.999      0.234
0.817      -3.732       4.668
ROIL           0.0382      0.276      0.138
0.891      -0.542       0.618
===============================================================
==============================
Omnibus:                        3.598   Durbin-W
atson:                   2.368
Prob(Omnibus):                  0.165   Jarque-B
```

```
era (JB):                    1.973
Skew:                               0.664   Prob(JB)
:                            0.373
Kurtosis:                           3.543   Cond. No
.                             584.
================================================
==============================

Warnings:
[1] Standard Errors assume that the covariance m
```

In [223]:

```
hypotheses = 'RINF=0, GIND=0, ROIL=0'
f_test=results_Mobil_Secondmodel_period1.f_test(hypotheses)
print(f_test)
```

```
<F test: F=array([[0.17434189]]), p=0.912356228553
3326, df_denom=18, df_num=3>
```

In [224]:

```
resi_Model_1=results_Mobil_period1.resid
resi_Model_2=results_Mobil_Secondmodel_period1.resid
```

In [225]:

```
resi_Model_1.std()
```

Out[225]:

```
0.05062466879452886
```

In [226]:

```
resi_Model_2.std()
```

Out[226]:

```
0.04990481881444332
```

In [227]:

```
wald_0 = results_Mobil_Secondmodel_period1.wald_test(hypothese
s)
print('H0:', hypotheses)
print(wald_0)
```

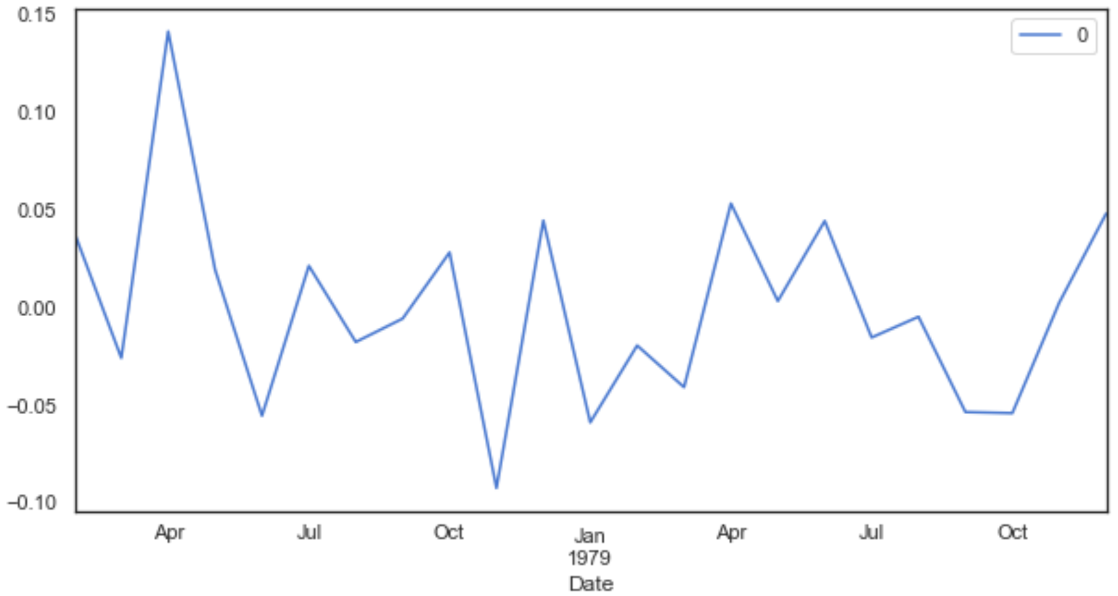
```
H0: RINF=0, GIND=0, ROIL=0
<F test: F=array([[0.17434189]]), p=0.912356228553
3326, df_denom=18, df_num=3>
```

In [228]:

```
residuals=pd.DataFrame(resi_Model_1)
residuals.plot()
```

Out[228]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c2170
08d0>
```



In [ ]:

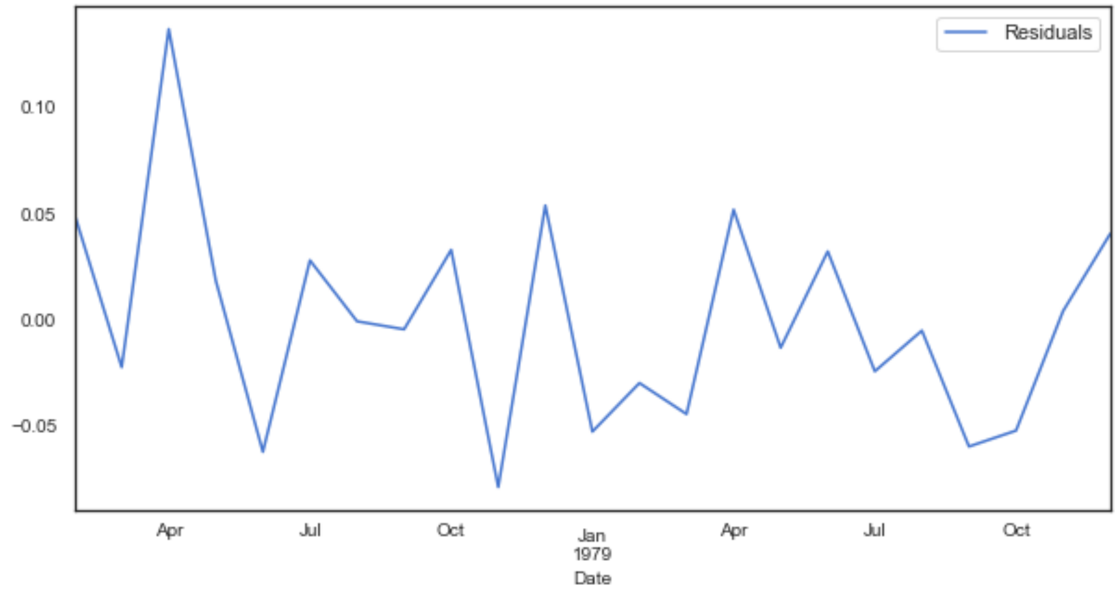
```

```

In [229]:

```
Plot_resi_corr_time(results_Mobil_Secondmodel_period1,MainDF_f
irst_period)
```

Out[229]:

	Residuals
Date	
1978-02-01	0.049545
1978-03-01	-0.022220
1978-04-01	0.136730
1978-05-01	0.018437
1978-06-01	-0.062015
1978-07-01	0.027997
1978-08-01	-0.000652
1978-09-01	-0.004416
1978-10-01	0.032938
1978-11-01	-0.078579
1978-12-01	0.053783
1979-01-01	-0.052503
1979-02-01	-0.029670
1979-03-01	-0.044270
1979-04-01	0.051829
1979-05-01	-0.013077
1979-06-01	0.032207
1979-07-01	-0.024156
1979-08-01	-0.005022
1979-09-01	-0.059519
1979-10-01	-0.052028
1979-11-01	0.004184
1979-12-01	0.040478



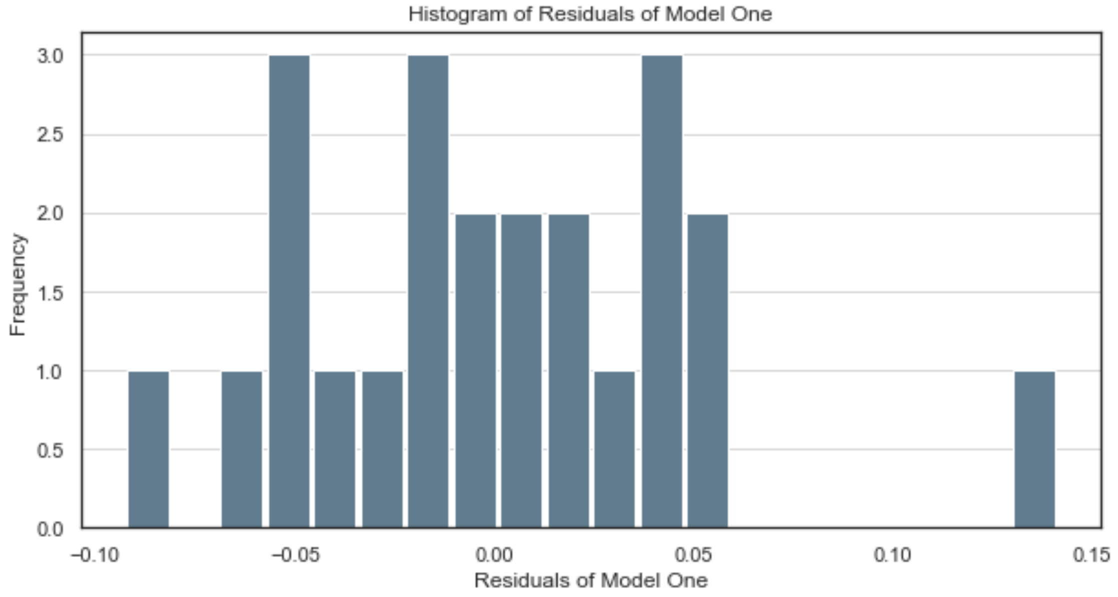
In [230]:

```
print(resi_Model_1.describe())
```

```
count    2.300000e+01
mean     3.921984e-18
std      5.062467e-02
min     -9.246818e-02
25%     -3.335145e-02
50%     -4.754541e-03
75%      3.323031e-02
max      1.413512e-01
dtype: float64
```

In [231]:

```
Figure=resi_Model_1.plot.hist(grid=False, bins=20, rwidth=0.9,
                                    color='#607c8e')
plt.title('Histogram of Residuals of Model One')
plt.xlabel('Residuals of Model One')
plt.ylabel('Frequency')
plt.grid(axis='y', alpha=0.75)
```



In [234]:

```
Adj_df= Create_lags_of_variable(MainDF_first_period, lags=[1,2
], column='CONTIL')
Adj_df.head()
```

Out[234]:

Date	CONTIL for time t-2	CONTIL for time t-1	CONTIL for time t-0
1978-04-01	0.037	0.003	0.180
1978-05-01	0.003	0.180	0.061
1978-06-01	0.180	0.061	-0.059
1978-07-01	0.061	-0.059	0.066
1978-08-01	-0.059	0.066	0.033

In [235]:

```
Adj_df1=Create_lags_of_variable(MainDF_first_period, lags=3, c
olumn='CONTIL')
Adj_df1.head()
```

Out[235]:

Date	CONTIL at time t-3	CONTIL at time t
1978-05-01	0.037	0.061
1978-06-01	0.003	-0.059
1978-07-01	0.180	0.066
1978-08-01	0.061	0.033
1978-09-01	-0.059	-0.013

In [236]:

```
Adj_df= Create_lags_of_variable(MainDF_first_period, lags=[1,2
,3], column='CONTIL')
Adj_df.head()
```

Out[236]:

Date	CONTIL for time t-3	CONTIL for time t-2	CONTIL for time t-1	CONTIL for time t-0
1978-05-01	0.037	0.003	0.180	0.061
1978-06-01	0.003	0.180	0.061	-0.059
1978-07-01	0.180	0.061	-0.059	0.066
1978-08-01	0.061	-0.059	0.066	0.033
1978-09-01	-0.059	0.066	0.033	-0.013

In [237]:

```
Adj_df.cov()
```

Out[237]:

	CONTIL for time t-3	CONTIL for time t-2	CONTIL for time t-1	CONTIL for time t-0
CONTIL for time t-3	0.004234	0.000363	-0.000880	0.000672
CONTIL for time t-2	0.000363	0.005371	-0.000179	-0.001827
CONTIL for time t-1	-0.000880	-0.000179	0.005629	0.000234
CONTIL for time t-0	0.000672	-0.001827	0.000234	0.004578

In [238]:

```
Adj_df.corr()
```

Out[238]:

	CONTIL for time t-3	CONTIL for time t-2	CONTIL for time t-1	CONTIL for time t-0
CONTIL for time t-3	1.000000	0.076213	-0.180310	0.152744
CONTIL for time t-2	0.076213	1.000000	-0.032594	-0.368462
CONTIL for time t-1	-0.180310	-0.032594	1.000000	0.046181
CONTIL for time t-0	0.152744	-0.368462	0.046181	1.000000

In [239]:

```
smt.stattools.acovf(MainDF_first_period['CONTIL'])[:5]
```

Out[239]:

```
array([ 5.02254820e-03,  1.15328183e-04, -1.305997
70e-03,  5.54623490e-04,
        7.23278540e-05])
```

In [240]:

```
smt.stattools.acf(MainDF_first_period['CONTIL'])[:5]
```

Out[240]:

```
array([ 1.        ,  0.02296209, -0.26002691,  0.1
1042671,  0.01440063])
```

In [241]:

```
LujungStatitic, Pvalue=sms.diagnostic.acorr_ljungbox(MainDF_fi
rst_period['CONTIL'], lags=15)
```

In [242]:

```
LujungStatitic
```

Out[242]:

```
array([ 0.01378059,  1.86511616,  2.21569536,  2.2
2197128,  2.51331336,
        5.79369477,  6.16667323,  6.67289896,  6.6
9039293,  7.83214627,
        9.62068974, 11.26534231, 11.30084601, 11.8
8387249, 12.80880527])
```

In [243]:

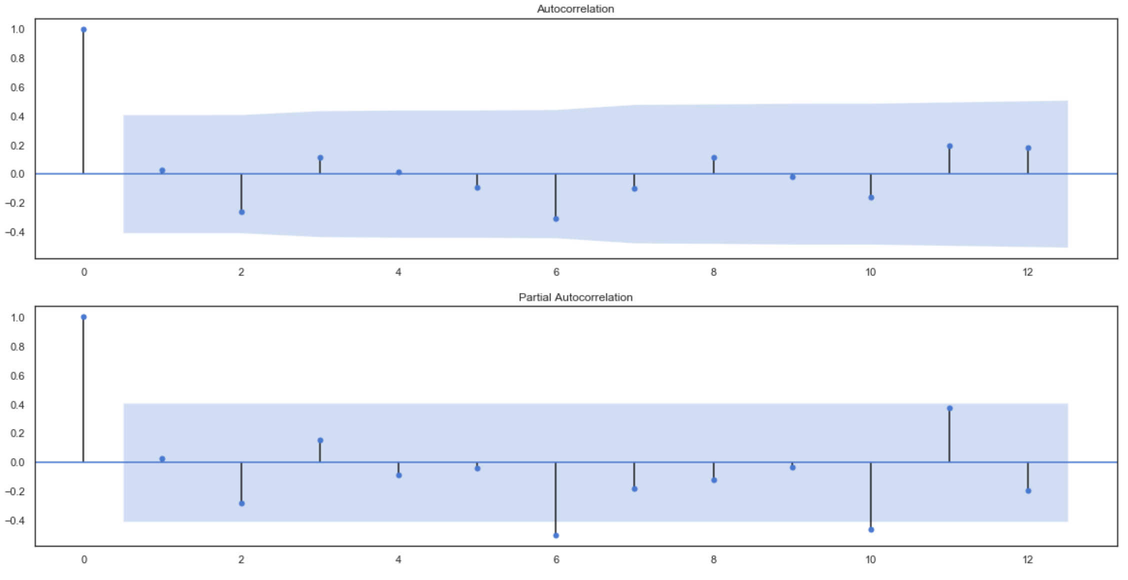
```
Pvalue
```

Out[243]:

```
array([0.90655041, 0.3935457 , 0.5288635 , 0.69500
887, 0.77448873,
       0.44669298, 0.52042835, 0.57229989, 0.66931
804, 0.6452286 ,
       0.56479404, 0.50632614, 0.58563063, 0.61562
829, 0.61706293])
```

In [244]:

```
plt.rcParams["figure.figsize"] = [20,10]
pyplot.figure()
pyplot.subplot(211)
smgtsplot.plot_acf(MainDF_first_period['CONTIL'], lags=12,  ax
=pyplot.gca())
pyplot.subplot(212)

smgtsplot.plot_pacf(MainDF_first_period['CONTIL'], lags=12, ax
=pyplot.gca())
pyplot.show()
```



In [245]:

```
acf,q,pval = smt.acf(MainDF_first_period['CONTIL'],nlags=12,qs
tat=True)
pacf = smt.pacf(MainDF_first_period['CONTIL'],nlags=12)

correlogram = pd.DataFrame({'acf':acf[1:],
                            'pacf':pacf[1:],
                            'Q':q,
                            'p-val':pval})
correlogram
```

Out[245]:

	acf	pacf	Q	p-val
0	0.022962	0.024006	0.013781	0.906550
1	-0.260027	-0.285532	1.865116	0.393546
2	0.110427	0.155384	2.215695	0.528863
3	0.014401	-0.087671	2.221971	0.695009
4	-0.095500	-0.039645	2.513313	0.774489
5	-0.311424	-0.504251	5.793695	0.446693
6	-0.101875	-0.184599	6.166673	0.520428
7	0.114917	-0.122394	6.672899	0.572300
8	-0.020638	-0.034259	6.690393	0.669318
9	-0.160666	-0.460632	7.832146	0.645229
10	0.193200	0.373956	9.620690	0.564794
11	0.177378	-0.195072	11.265342	0.506326

In [246]:

```
x=MainDF_first_period['CONTIL']
a= smt.stattools.arma_order_select_ic(x, max_ar=5, max_ma=3, i
c=['aic', 'bic', 'hqic'])
```

In [247]:

```
a
```

Out[247]:

```
{'aic':             0           1           2
3
 0 -52.486638 -50.514004 -51.856820 -49.609435
 1 -50.499138 -50.846145 -50.516722 -47.802699
 2 -50.157470 -49.327535 -49.710367 -46.841670
 3 -48.996520 -47.649060 -47.799412 -44.899620
 4 -47.044130 -49.105173        NaN -44.650059
 5 -45.180742        NaN        NaN        NaN,
 'bic':             0           1           2
3
 0 -50.215650 -47.107521 -47.314843 -43.931964
 1 -47.092655 -46.304168 -44.839251 -40.989734
 2 -45.615493 -43.650064 -42.897402 -38.893211
 3 -43.319049 -40.836095 -39.850952 -35.815666
 4 -40.231164 -41.156713        NaN -34.430611
 5 -37.232283        NaN        NaN        NaN,
 'hqic':             0           1           2
3
 0 -51.915491 -49.657283 -50.714525 -48.181567
 1 -49.642417 -49.703851 -49.088854 -46.089257
 2 -49.015176 -47.899667 -47.996926 -44.842655
 3 -47.568652 -45.935619 -45.800396 -42.615031
 4 -45.330688 -47.106157        NaN -42.079896
 5 -43.181727        NaN        NaN        NaN,
 'aic_min_order': (0, 0),
 'bic_min_order': (0, 0),
 'hqic_min_order': (0, 0)}
```

In [248]:

```
res=smt.ARIMA(MainDF_first_period['CONTIL'], order=(0,0,1)).fi
t()
print(res.summary())
```

```
                              ARMA Model Results
=================================================
==============================
Dep. Variable:                     CONTIL   No. Obse
rvations:                  23
Model:                         ARMA(0, 1)   Log Like
lihood                  28.257
Method:                           css-mle   S.D. of
innovations              0.071
Date:                    Wed, 18 Dec 2019   AIC
-50.514
Time:                            16:56:04   BIC
-47.108
Sample:                        02-01-1978   HQIC
-49.657
                             - 12-01-1979
=================================================
================================
                   coef    std err          z
P>|z|      [0.025      0.975]
-------------------------------------------------
--------------------------------
const            0.0174      0.016      1.116
0.277      -0.013       0.048
ma.L1.CONTIL     0.0521      0.316      0.165
0.871      -0.568       0.672
                                    Roots
=================================================
=============================
                  Real          Imaginary
Modulus         Frequency
-------------------------------------------------
-----------------------------
MA.1          -19.1907          +0.0000j
19.1907           0.5000
-------------------------------------------------
-----------------------------
```

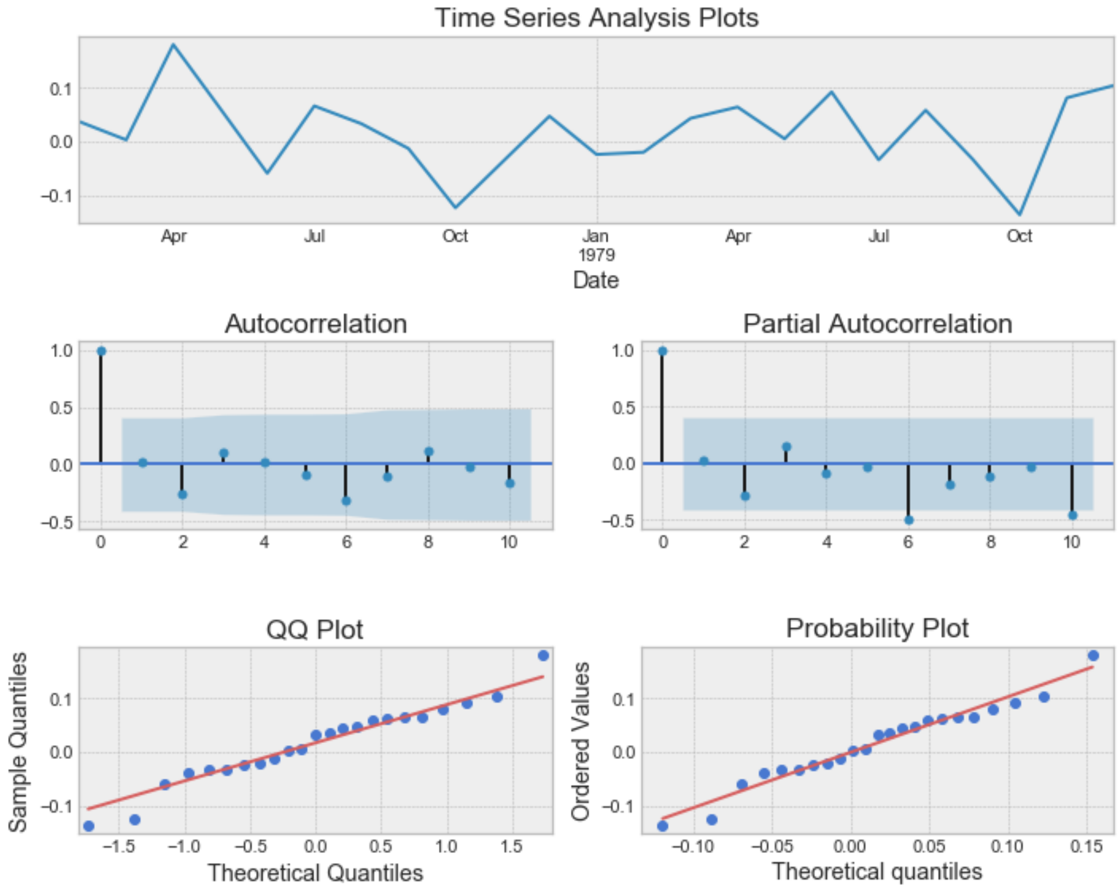
In [250]:

```
res=smt.ARIMA(MainDF_first_period['CONTIL'], order=(2,0,1)).fi
t()
print(res.summary())
```

```
                              ARMA Model Results
==================================================
============================
Dep. Variable:                   CONTIL   No. Observ
ations:                    23
Model:                       ARMA(2, 1)   Log Likeli
hood                   29.664
Method:                         css-mle   S.D. of in
novations               0.064
Date:                  Wed, 18 Dec 2019   AIC
-49.328
Time:                          16:56:24   BIC
-43.650
Sample:                      02-01-1978   HQIC
-47.900
                           - 12-01-1979
==================================================
==============================
                   coef    std err          z
P>|z|      [0.025      0.975]
--------------------------------------------------
------------------------------
const             0.0176      0.014      1.266
0.221      -0.010       0.045
ar.L1.CONTIL     -0.7729      0.213     -3.631
0.002      -1.190      -0.356
ar.L2.CONTIL     -0.1487      0.211     -0.705
0.489      -0.562       0.265
ma.L1.CONTIL      1.0000        nan        nan
nan          nan          nan
                                    Roots
==================================================
===========================
                  Real          Imaginary
Modulus         Frequency
--------------------------------------------------
---------------------------
AR.1           -2.4229           +0.0000j
2.4229            0.5000
AR.2           -2.7765           +0.0000j
2.7765            0.5000
MA.1           -1.0000           +0.0000j
1.0000            0.5000
--------------------------------------------------
---------------------------
```

In [251]:

```
RF.tsplot(MainDF_first_period['CONTIL'], lags=10)
```



In [252]:

```
MainDF.index.get_loc('1980-02-01')
```

Out[252]:

```
24
```

In [253]:

```
MainDF.index.get_loc('1983-02-01')
```

Out[253]:

```
60
```

In [254]:

```
df.head()
```

Out[254]:

	BOISE	CITCRP	CONED	CONTIL	DATGEN	DEC	DELTA	GI
Date								
1976-01-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1976-02-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1976-03-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1976-04-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1976-05-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

5 rows × 27 columns

In [255]:

```
train=MainDF[0:25]
test=MainDF[25:61]
```

In [256]:

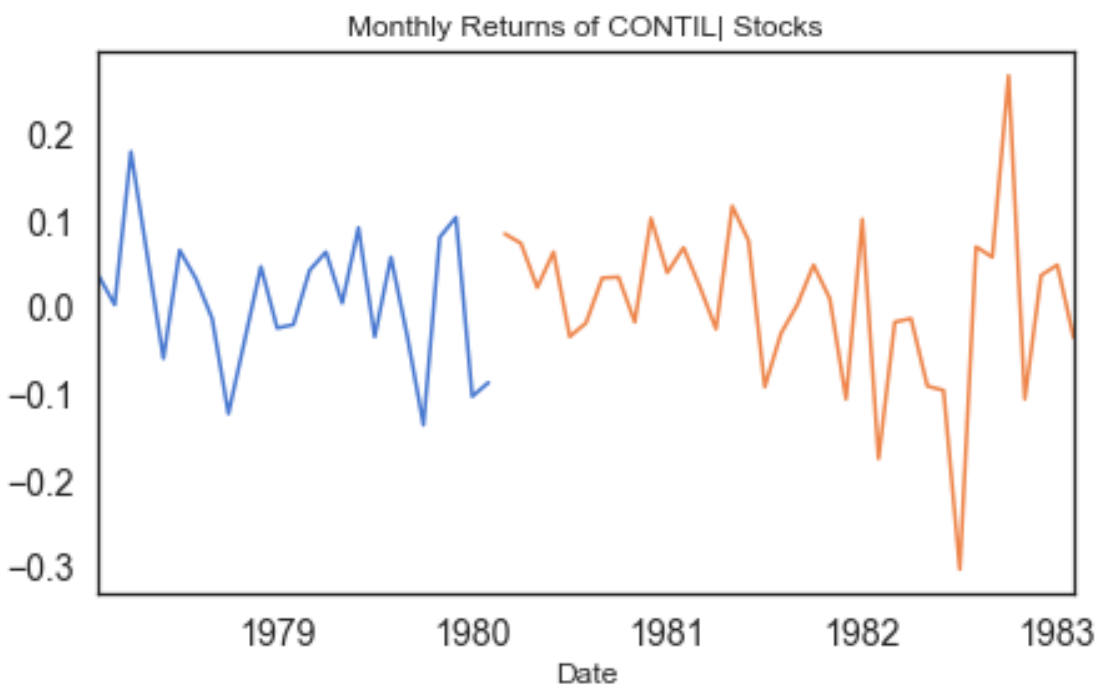
```
train.head()
```

Out[256]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
Date								
1978-02-01	0.037	-0.043	0.010	0.00494	188.4	8.84	139.2	0.(
1978-03-01	0.003	-0.063	0.050	0.00526	189.8	8.80	140.9	0.(
1978-04-01	0.180	0.130	0.063	0.00491	191.5	8.82	143.2	0.(
1978-05-01	0.061	-0.018	0.067	0.00513	193.3	8.81	143.9	0.(
1978-06-01	-0.059	-0.004	0.007	0.00527	195.3	9.05	144.9	0.(

In [259]:

```
train['CONTIL'].plot(figsize=(7,4), title= 'Monthly Returns of CONTIL Stocks', fontsize=14)
test['CONTIL'].plot(figsize=(7,4), title= 'Monthly Returns of CONTIL| Stocks', fontsize=14)
plt.show()
```



In [260]:

```
df1 = MainDF.resample('Y').mean()
df1
```

Out[260]:

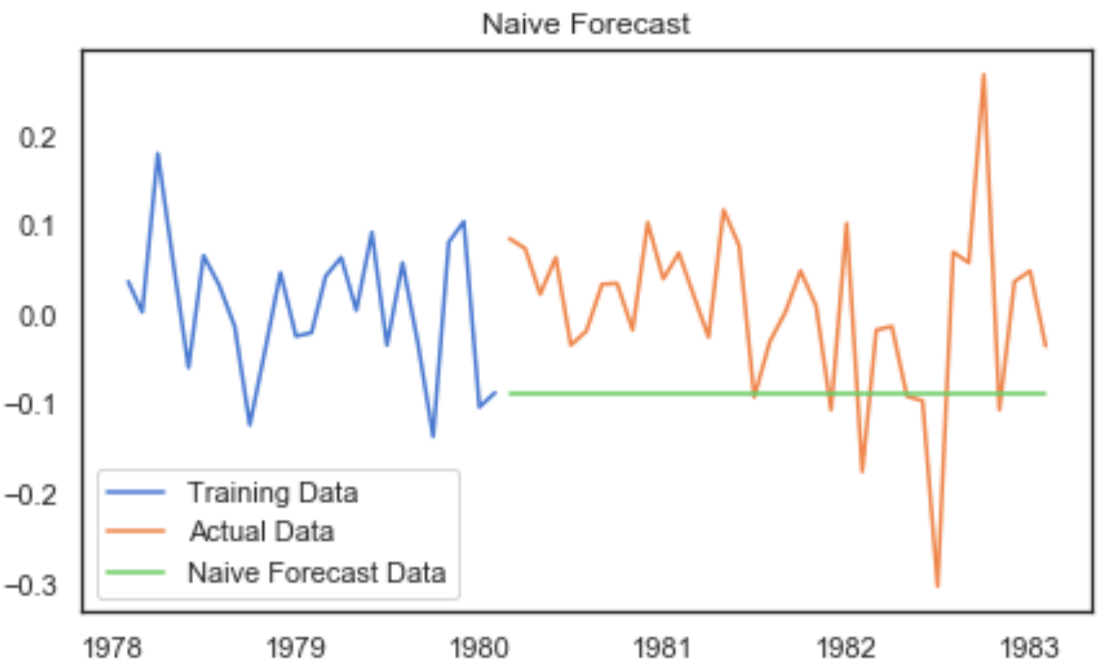
	CONTIL	IBM	MARKET	RKFREE	CPI	PC
Date						
1978-12-31	0.017636	0.016818	0.023545	0.005841	196.209091	8.9381
1979-12-31	0.016667	-0.007417	0.044417	0.007783	217.441667	12.6116
1980-12-31	0.013250	0.010750	0.030667	0.008521	246.816667	21.6058
1981-12-31	0.011333	-0.008917	-0.003250	0.010513	272.350000	31.8508
1982-12-31	-0.022167	0.050750	0.006750	0.007848	289.150000	28.5358
1983-12-31	0.015250	0.024250	0.023500	0.006703	298.416667	26.1333
1984-12-31	-0.025333	0.004667	-0.006583	0.007182	311.150000	25.8808
1985-12-31	0.058583	0.024000	0.015250	0.005637	322.191667	24.0750
1986-12-31	-0.047250	-0.018083	0.005917	0.004506	328.383333	12.4508
1987-12-31	-0.044636	-0.000455	-0.000727	0.004038	339.900000	15.5227

In [261]:

```
df_out_sample= test.copy()
df_in_sample=train.copy()
```

In [262]:

```
f_Method_NF=RF.Naive_Forecast(df_in_sample, df_out_sample, 'CO
NTIL')
```



In [263]:

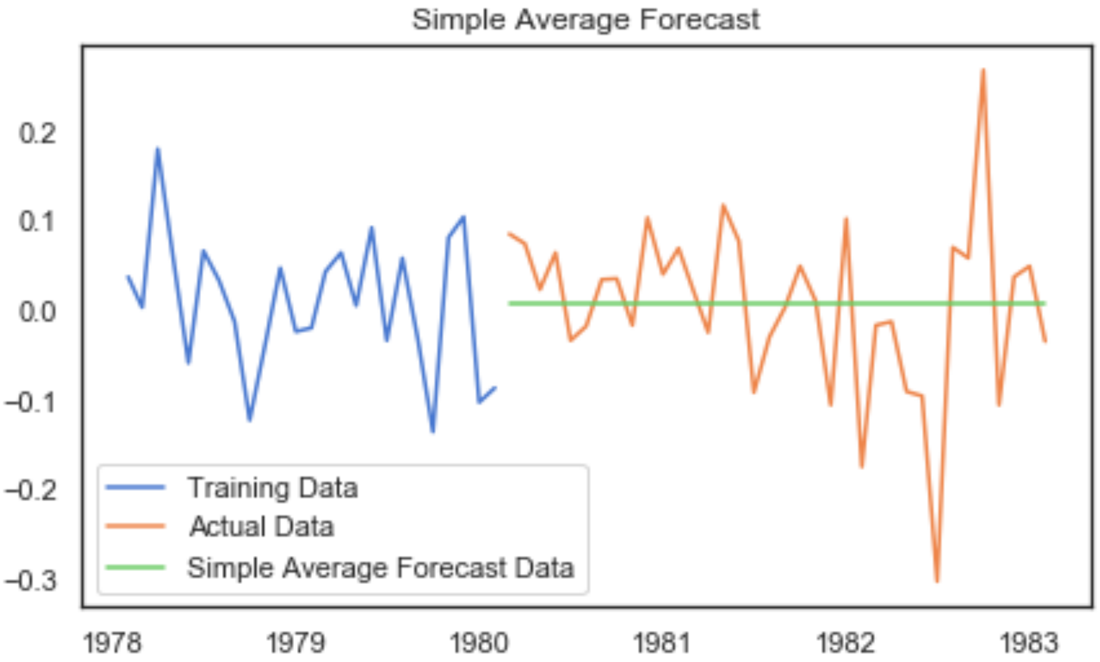
```
f_Method_NF.head()
```

Out[263]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
0	-0.087	-0.062	0.112	0.01073	236.4	18.81	152.6	0.0
1	-0.087	-0.062	0.112	0.01073	236.4	18.81	152.6	0.0
2	-0.087	-0.062	0.112	0.01073	236.4	18.81	152.6	0.0
3	-0.087	-0.062	0.112	0.01073	236.4	18.81	152.6	0.0
4	-0.087	-0.062	0.112	0.01073	236.4	18.81	152.6	0.0

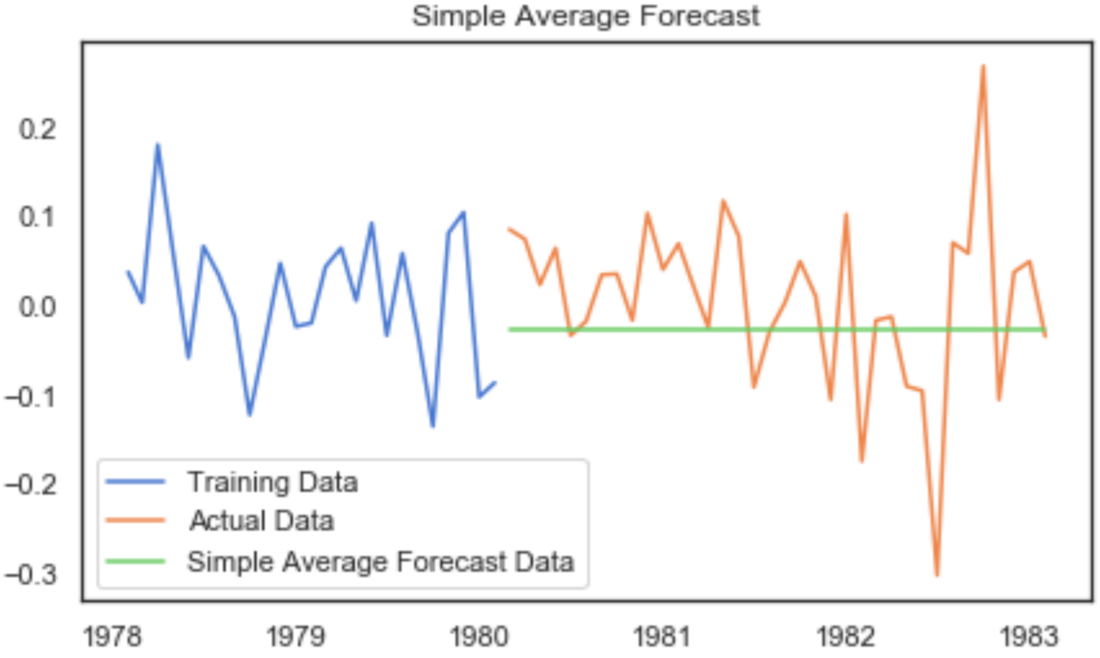
In [264]:

```
f_Method_AF=RF.Average_Forecast(df_in_sample, df_out_sample, 'CONTIL')
```



In [266]:

```
f_Method_MAF=RF.Moving_Average_Forecast(df_in_sample, df_out_sample, 'CONTIL', 3)
```



In [191]:

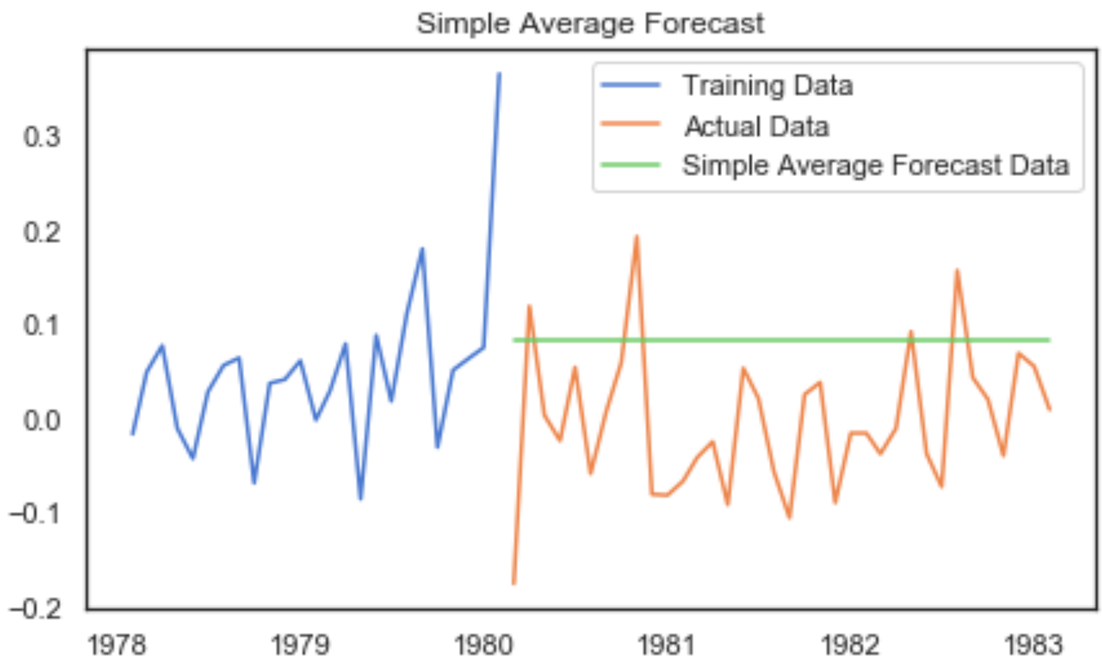
```
f_Method_MAF.head()
```

Out[191]:

	MOBIL	IBM	MARKET	RKFREE	CPI	POIL	FRBINI
Date							
1980-03-01	0.168	-0.003	0.107333	0.00901	233.166667	17.9	152.(
1980-04-01	0.168	-0.003	0.107333	0.00901	233.166667	17.9	152.(
1980-05-01	0.168	-0.003	0.107333	0.00901	233.166667	17.9	152.(
1980-06-01	0.168	-0.003	0.107333	0.00901	233.166667	17.9	152.(
1980-07-01	0.168	-0.003	0.107333	0.00901	233.166667	17.9	152.(

In [192]:

```
f_Method_MAF_1=RF.Moving_Average_Forecast(df_in_sample, df_out
_sample, 'MOBIL', 10)
```



In [193]:

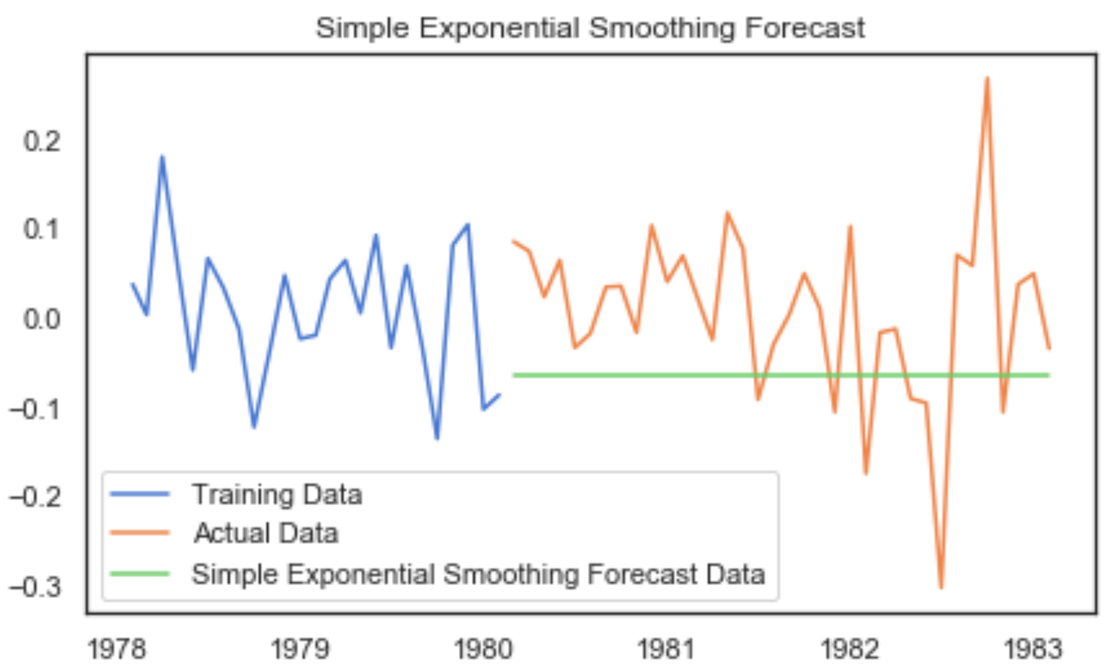
```
f_Method_MAF_1.head()
```

Out[193]:

Date	MOBIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
1980-03-01	0.0835	-0.015	0.0551	0.008241	224.65	14.87	152.63	0
1980-04-01	0.0835	-0.015	0.0551	0.008241	224.65	14.87	152.63	0
1980-05-01	0.0835	-0.015	0.0551	0.008241	224.65	14.87	152.63	0
1980-06-01	0.0835	-0.015	0.0551	0.008241	224.65	14.87	152.63	0
1980-07-01	0.0835	-0.015	0.0551	0.008241	224.65	14.87	152.63	0

In [267]:

```
f_Method_SES=RF.Simple_Exponential_Smoothing_Forecast(df_in_sa
mple, df_out_sample, 'CONTIL', 0.6)
```



In [268]:

```
f_Method_SES.head()
```

Out[268]:

Date	CONTIL	IBM	MARKET	RKFREE	CPI	PO
1980-03-01	-0.065966	-0.021739	0.108678	0.00981	234.348543	18.1797
1980-04-01	-0.065966	-0.021739	0.108678	0.00981	234.348543	18.1797
1980-05-01	-0.065966	-0.021739	0.108678	0.00981	234.348543	18.1797
1980-06-01	-0.065966	-0.021739	0.108678	0.00981	234.348543	18.1797
1980-07-01	-0.065966	-0.021739	0.108678	0.00981	234.348543	18.1797

In [269]:

```
(df_out_sample-f_Method_SES).head()
```

Out[269]:

Date	CONTIL	IBM	MARKET	RKFREE	CPI	PO
1980-03-01	0.150966	-0.100261	-0.351678	0.00200	5.451457	1.1602
1980-04-01	0.139966	0.005739	-0.028678	-0.00228	8.151457	2.1102
1980-05-01	0.088966	0.046739	-0.046678	-0.00351	10.551457	2.8302
1980-06-01	0.129966	0.082739	-0.022678	-0.00478	13.251457	3.3502
1980-07-01	0.031966	0.132739	-0.043678	-0.00379	13.451457	4.0802

In [270]:

```
((df_out_sample-f_Method_SES)**2).head()
```

Out[270]:

Date	CONTIL	IBM	MARKET	RKFREE	CPI	POIL
1980-03-01	0.022791	0.010052	0.123677	0.000004	29.718383	1.346138
1980-04-01	0.019591	0.000033	0.000822	0.000005	66.446251	4.453079
1980-05-01	0.007915	0.002185	0.002179	0.000012	111.333245	8.010212
1980-06-01	0.016891	0.006846	0.000514	0.000023	175.601112	11.224054
1980-07-01	0.001022	0.017620	0.001908	0.000014	180.941695	16.648292

In [271]:

```
np.mean((df_out_sample-f_Method_SES)**2)
```

Out[271]:

```
CONTIL            0.014358
IBM               0.004682
MARKET            0.015962
RKFREE            0.000006
CPI            1768.163085
POIL            108.838698
FRBIND          100.875058
RINF              0.000073
GIND              0.000157
real_POIL         0.000753
ROIL              0.003114
dtype: float64
```

In [272]:

```
np.std((df_out_sample-f_Method_SES)**2)
```

Out[272]:

```
CONTIL            0.020123
IBM               0.005847
MARKET            0.023549
RKFREE            0.000007
CPI            1235.213421
POIL             73.032603
FRBIND           99.424877
RINF              0.000073
GIND              0.000189
real_POIL         0.000748
ROIL              0.003148
dtype: float64
```

In [280]:

```
np.mean((df_out_sample-f_Method_MAF_1)**2)
```

Out[280]:

```
CONTIL                 NaN
CPI            2605.754167
FRBIND          101.286233
GIND              0.000163
IBM               0.004186
MARKET            0.007982
MOBIL                  NaN
POIL            183.487608
RINF              0.000046
RKFREE            0.000005
ROIL              0.003888
real_POIL         0.001451
dtype: float64
```

In [274]:

```
co_of_variation_Method_SES=np.std((df_out_sample-f_Method_SES)**2)/np.mean((df_out_sample-f_Method_SES)**2)
co_of_variation_Method_SES
```

Out[274]:

```
CONTIL       1.401542
IBM          1.248733
MARKET       1.475277
RKFREE       1.116010
CPI          0.698586
POIL         0.671017
FRBIND       0.985624
RINF         1.011821
GIND         1.206223
real_POIL    0.994007
ROIL         1.011014
dtype: float64
```

In [275]:

```
MainDF_first_period.tail()
```

Out[275]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND
Date							
1979-08-01	0.058	0.016	0.095	0.00789	221.1	14.00	152.1
1979-09-01	-0.033	-0.032	0.039	0.00802	223.4	14.57	152.7
1979-10-01	-0.136	-0.079	-0.097	0.00913	225.4	15.11	152.7
1979-11-01	0.081	0.060	0.116	0.00819	227.5	15.52	152.3
1979-12-01	0.104	-0.013	0.086	0.00747	229.9	17.03	152.5

In [276]:

```
MainDF_second_period.head()
```

Out[276]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
Date								
1980-01-01	-0.103	0.066	0.124	0.00883	233.2	17.86	152.7	0.
1980-02-01	-0.087	-0.062	0.112	0.01073	236.4	18.81	152.6	0.
1980-03-01	0.085	-0.122	-0.243	0.01181	239.8	19.34	152.1	0.
1980-04-01	0.074	-0.016	0.080	0.00753	242.5	20.29	148.3	0.
1980-05-01	0.023	0.025	0.062	0.00630	244.9	21.01	144.0	0.

In [277]:

```
MainDF_1=MainDF_first_period.append(MainDF_second_period)
```

In [278]:

```
MainDF_1.head()
```

Out[278]:

	CONTIL	IBM	MARKET	RKFREE	CPI	POIL	FRBIND	
Date								
1978-02-01	0.037	-0.043	0.010	0.00494	188.4	8.84	139.2	0.(
1978-03-01	0.003	-0.063	0.050	0.00526	189.8	8.80	140.9	0.(
1978-04-01	0.180	0.130	0.063	0.00491	191.5	8.82	143.2	0.(
1978-05-01	0.061	-0.018	0.067	0.00513	193.3	8.81	143.9	0.(
1978-06-01	-0.059	-0.004	0.007	0.00527	195.3	9.05	144.9	0.(

In [281]:

```
np.mean((df_out_sample-f_Method_MAF_1)**2)
```

Out[281]:

```
CONTIL                 NaN
CPI            2605.754167
FRBIND          101.286233
GIND              0.000163
IBM               0.004186
MARKET            0.007982
MOBIL                  NaN
POIL            183.487608
RINF              0.000046
RKFREE            0.000005
ROIL              0.003888
real_POIL         0.001451
dtype: float64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: