Homework_3

August 23, 2022

1 Homework 3- Robust Estimates

1.1 Question 1 Robust Esimators

- Use the data in spy_rates
- This set provides data on various asset classes.
- Use the multivariate regression model in homework 2 of SPY on the dividend-price ratio and tresury rate.

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
import matplotlib as mpl
import seaborn as sns
```

[ ]:
```
plt.style.use("seaborn")
mpl.rcParams['font.family'] = 'serif'
%matplotlib inline
```

[ ]:
```
spyrate = pd.read_excel("C:/Users/dcste/OneDrive/Uchicago_practice/
↪spy_rate_data (1).xlsx")
spyrate = spyrate.rename(columns = {"Unnamed: 0" : "Date"}).set_index("Date")
```

Calculate the regression : $r_{spy,t} = \alpha + \beta_{yield,t} + \beta_{dvd/p,t} + e$

This regression estimates the impact that the 10 year treasury yield and divident-price yield has on SPY returns.

[ ]:
```
spyreturn = spyrate["SPY US Equity"]
independ_var = sm.add_constant(spyrate.drop(columns = "SPY US Equity"))
estimation = sm.OLS(spyreturn , independ_var).fit()
estimation.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:          SPY US Equity   R-squared:                       0.036
Model:                            OLS   Adj. R-squared:                  0.028
Method:                 Least Squares   F-statistic:                     4.465
Date:                Tue, 23 Aug 2022   Prob (F-statistic):             0.0125
Time:                        14:57:51   Log-Likelihood:                 423.48
No. Observations:                 239   AIC:                            -841.0
Df Residuals:                     236   BIC:                            -830.5
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
===
                      coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
const               0.0741      0.023      3.191      0.002       0.028
0.120
10-yr Yields       -0.7707      0.280     -2.754      0.006      -1.322
-0.219
Dvd-Price Ratio    -2.2646      0.849     -2.669      0.008      -3.936
-0.593
==============================================================================
Omnibus:                        6.566   Durbin-Watson:                   1.887
Prob(Omnibus):                  0.038   Jarque-Bera (JB):                6.427
Skew:                          -0.333   Prob(JB):                       0.0402
Kurtosis:                       3.448   Cond. No.                         325.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

- Calculate the correlation between ϵ_t and ϵ_{t-1}.

[ ]:
```
residuals = estimation.resid
rho = residuals.corr(residuals.shift(1))
print(f"The correlation between the residuals at t and (t-1) is {rho:.3%}.")
```

```
The correlation between the residuals at t and (t-1) is 5.089%.
```

A correlation of 5.089% means there is a very small positive linear relationship. In theory we expected $E[cov(u_t, u_{t-1}) = 0]$.

Calculate the regression of $e_t = \alpha + \beta_{yield,t} + \beta_{dvd/p,t} + e_t$

[ ]:
```
mod1 = sm.OLS(residuals**2, independ_var).fit()
mod1.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.065
Model:                            OLS   Adj. R-squared:                  0.057
Method:                 Least Squares   F-statistic:                     8.172
Date:                Tue, 23 Aug 2022   Prob (F-statistic):           0.000370
Time:                        15:15:25   Log-Likelihood:                 1087.1
No. Observations:                 239   AIC:                            -2168.
Df Residuals:                     236   BIC:                            -2158.
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
===
                      coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
const              -0.0041      0.001     -2.840      0.005      -0.007
-0.001
10-yr Yields        0.0586      0.017      3.367      0.001       0.024
0.093
Dvd-Price Ratio     0.2053      0.053      3.887      0.000       0.101
0.309
==============================================================================
Omnibus:                      150.121   Durbin-Watson:                   1.572
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1029.503
Skew:                           2.511   Prob(JB):                    2.79e-224
Kurtosis:                      11.841   Cond. No.                         325.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

What do the previous two calculations have to do with identifying serial correlation and heteroskedasticity?

The regression of the residuals squared on the independent variables gives us information on whether or not the residual variance is always the same distribution(**homoskedastic**) or whether the residual variance varies with X (**heteroskedastic**). From the regression output above we get an F-stat $= 8.1$ meaning we reject the null-hypothesis of homoskedasticity and conclude the error terms depend on the independent variables.

1.2 Homework 3 - Robust Estimators

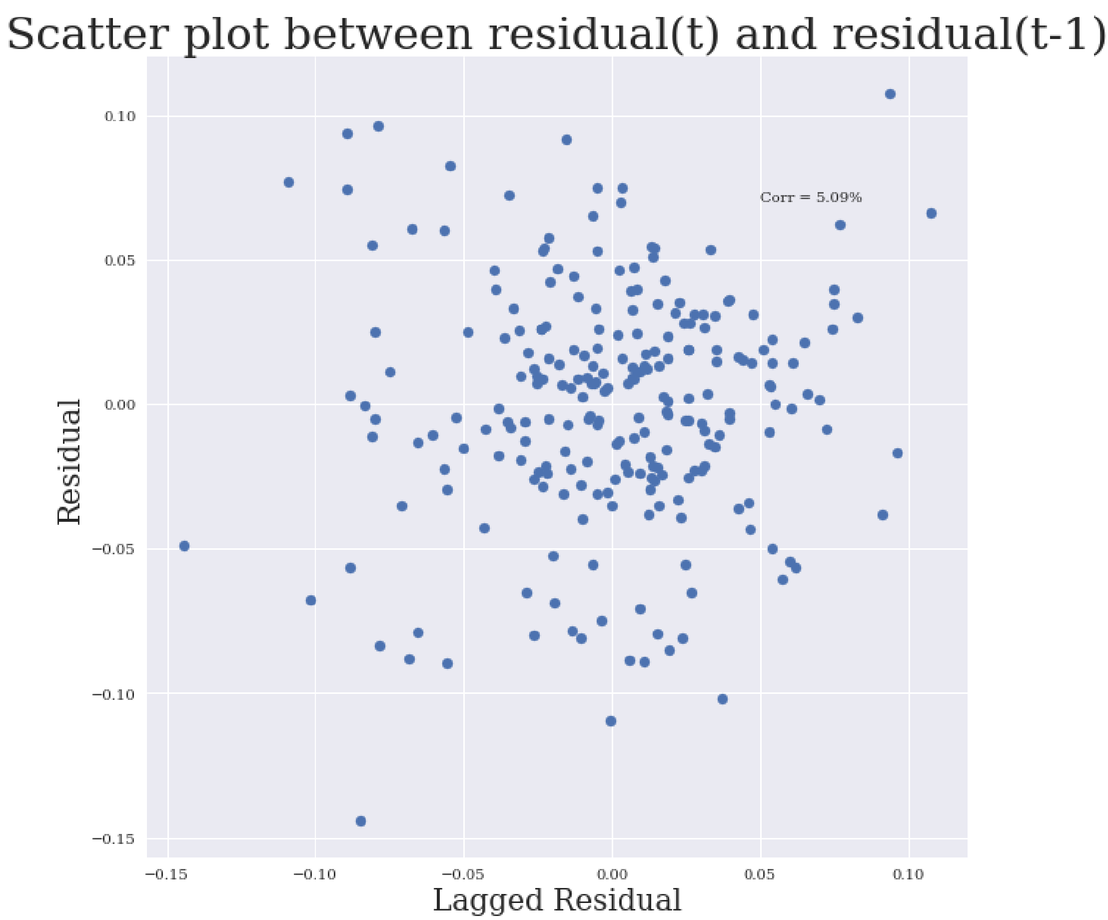
1. Calculate the residuals from the multivariate regression model. Also compute the first lag of the residuals. Plot the residuals and the first lag of residuals using a scatterplot. Calculate the correlation between the two. Interpret the results
2. Compute and Display the ACF plot for the residuals.

[ ]:
```
df_resid = pd.DataFrame(residuals , columns = ["resid"])
lag_resid = pd.DataFrame(residuals.shift(), columns = ["lag_resid"])
residual_df = pd.concat((df_resid,lag_resid), axis = 1).dropna()
residual_df.corr()
```

[ ]:

	resid	lag_resid
resid	1.000000	0.050893
lag_resid	0.050893	1.000000

[ ]:
```
# Producing a scatter plot between the residuals and the lag residuals
plt.figure(figsize = (10,10))
plt.scatter(residual_df["lag_resid"], residual_df["resid"])
plt.xlabel("Lagged Residual", fontsize = 20)
plt.ylabel("Residual", fontsize = 20)
plt.title("Scatter plot between residual(t) and residual(t-1)", fontsize = 30)
plt.annotate("Corr = {:,.2%}".format(residual_df["resid"].
 ↪corr(residual_df["lag_resid"])), (.05,.07))
plt.show()
```

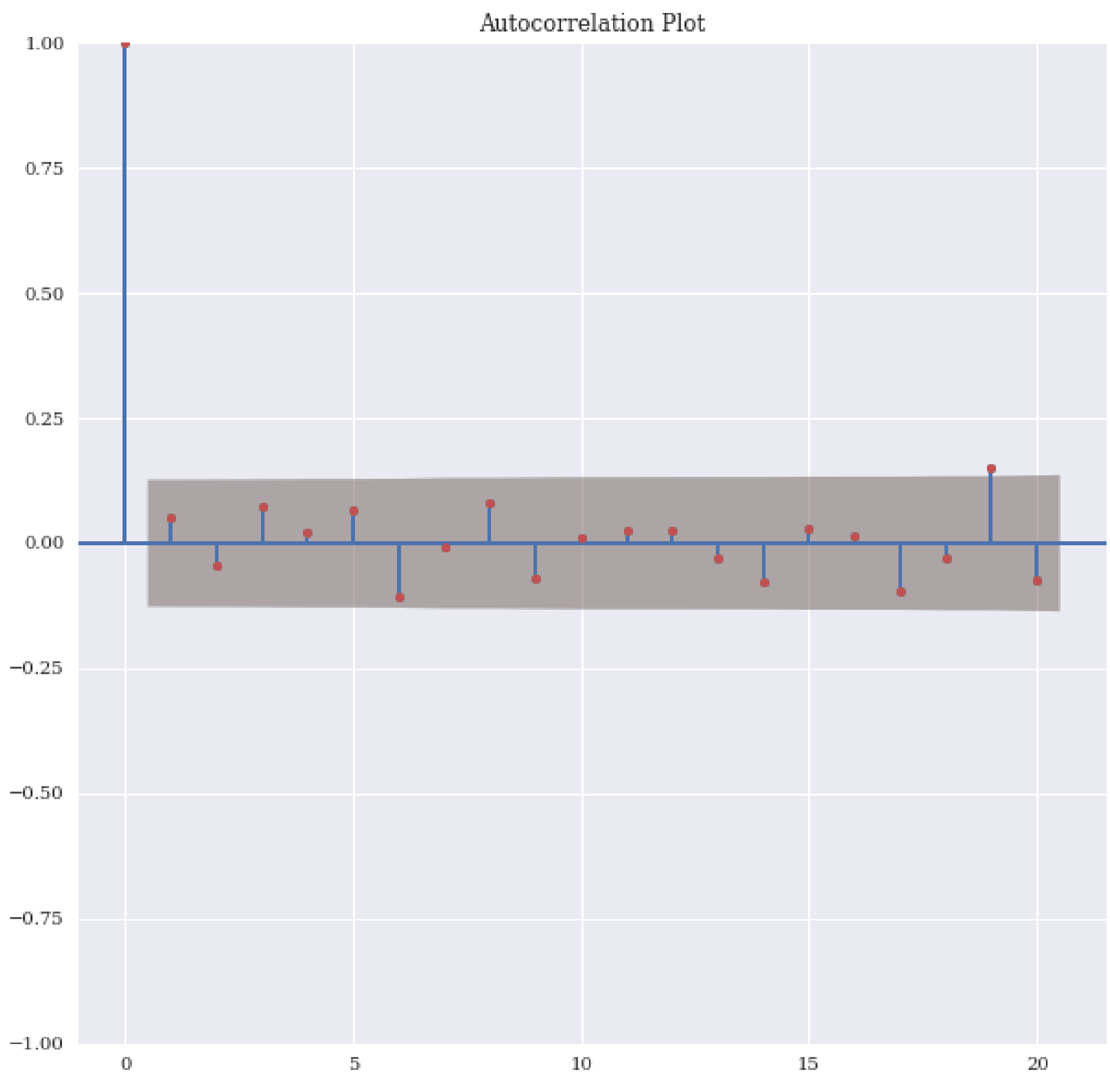


A correlation of 5% does not indicate significant serial correlation in the sample residuals. To test for correlation between higher lags we need an autocorrelation plot and the Durbin Watson Test is another method to see if serial correlation is present.

[ ]:
```
from statsmodels.graphics.tsaplots import plot_acf

plot_acf(df_resid, lags = 20, ax = ax, title = "Autocorrelation Plot")
```

[ ]:



From the autocorrelation plot we can see there is not evidence of serial correlation.

1.3 Durbin Watson Test

$DW = \frac{\sum_{t=2}^{n}(u_t - u_{t-1})^2}{\sum_{t=1}^{n} u^2}$

- Compute the Durbin-Watson Statistic

[ ]:
```
from statsmodels.stats.stattools import durbin_watson

dw = float(durbin_watson(df_resid))
print("The Durbin-Watson Statistic is equal to: {:.4f}".format(dw))
```

```
The Durbin-Watson Statistic is equal to: 1.8867
```

*The DW statistic ranges from zero to four, with a value of 2.0 indicating zero autocorrelation.

Values below 2.0 mean there is positive autocorrelation and above 2.0 indicates negative autocorrelation. In this case, since the value is close to 2, we can say that autocorrealtion is not observed.

1.4 Heteroskedasticity and White's Test

1. Create a scatterplot of the model residuals against the fitted values and the model regressors. Do you see any patterns. What can you conclude?

2. Calculate White's test for heteroskedasticity. Display the test statistic and the p-value. State the null and the alternative hypothesis for this test. Interpret your results. What can you conclude?

[ ]:
```
fitted_values = pd.DataFrame(estimation.fittedvalues, columns = ["Fitted␣
 ↪Values"])
div_price_ratio = spyrate["Dvd-Price Ratio"]
US_Tres = spyrate["10-yr Yields"]
fig, ax = plt.subplots(3,1, figsize = (16,30))

sns.regplot(x = fitted_values, y = residuals, ax = ax[0])
ax[0].set_ylabel('Residuals')
ax[0].set_xlabel('Fitted Values')
ax[0].set_title('Fig 3: Residuals vs Fitted Values')

sns.regplot(x = div_price_ratio, y = residuals, ax = ax[1])
ax[1].set_ylabel("Residual")
ax[1].set_xlabel("Dividend-Price-Ratio")
ax[1].set_title("Residuals vs Dividend-to-Price Ratio")

sns.regplot(x = US_Tres, y = residuals, ax = ax[2])
ax[2].set_ylabel("Residuals")
ax[2].set_xlabel("10-Year Yield")
ax[2].set_title("Residuals vs 10-Year Yield")

plt.show()
```

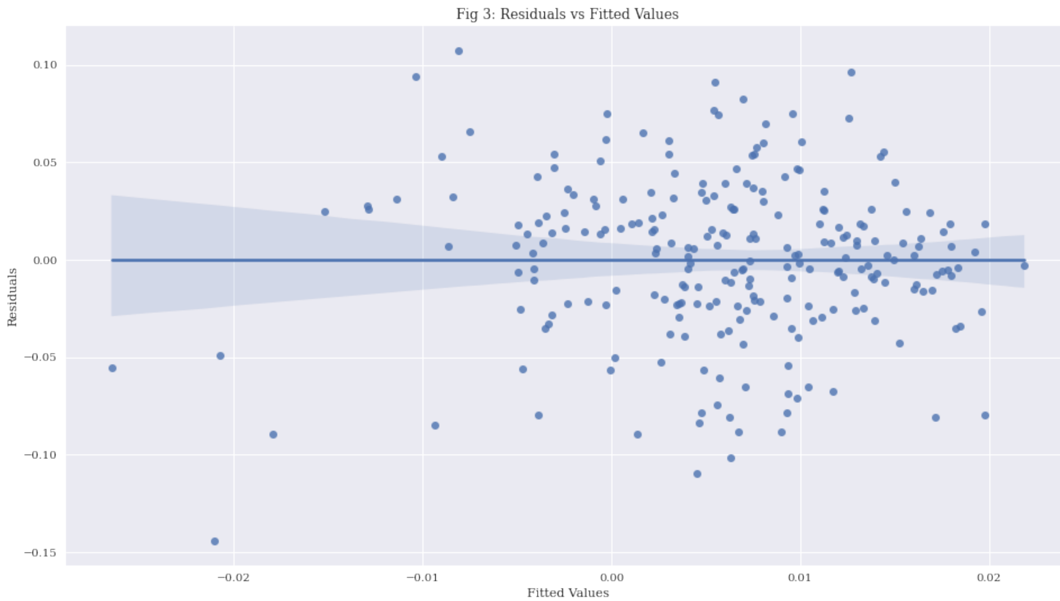
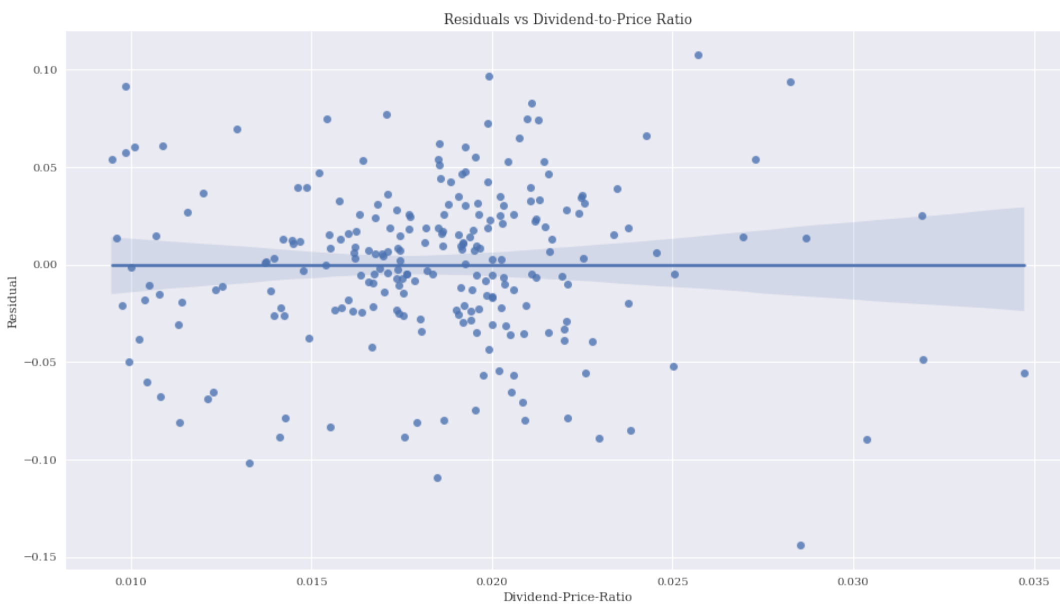
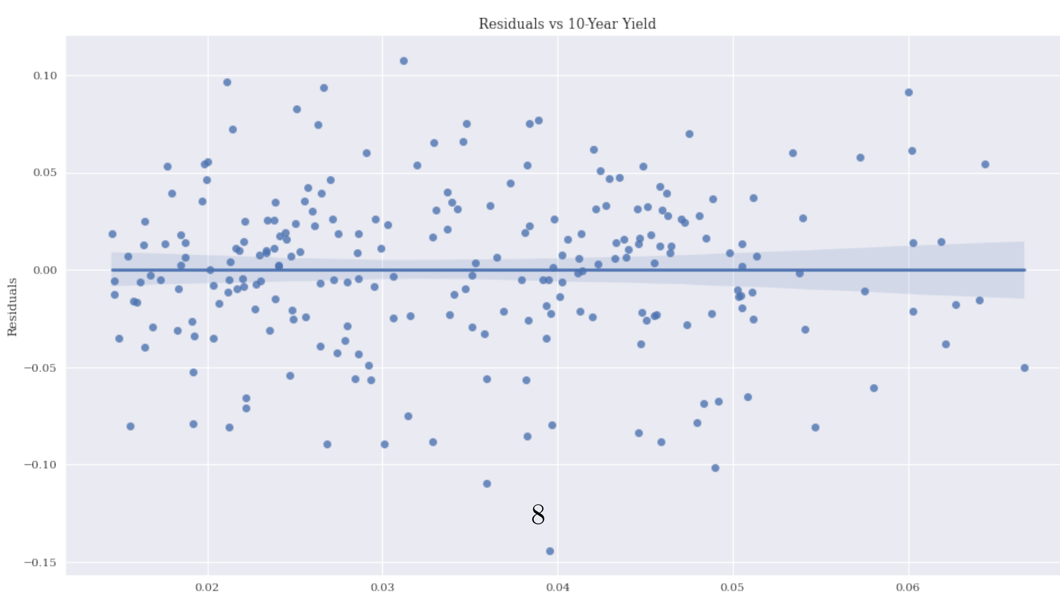


Fig 3: Residuals vs Fitted Values
Residuals
Fitted Values
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.02
−0.01
0.00
0.01
0.02


Residuals vs Dividend-to-Price Ratio
Residual
Dividend-Price-Ratio
0.10
0.05
0.00
−0.05
−0.10
−0.15
0.010
0.015
0.020
0.025
0.030
0.035


Residuals vs 10-Year Yield
Residuals
10-Year Yield
0.10
0.05
0.00
−0.05
−0.10
−0.15
0.02
0.03
0.04
0.05
0.06

[ ]:

```

```

[ ]:

```
                resid
Date
1999-10-31  0.061022
1999-11-30  0.014480
1999-12-31  0.054078
2000-01-31 -0.049956
2000-02-29 -0.015486
…                …
2019-04-30  0.024029
2019-05-31 -0.080913
2019-06-30  0.055212
2019-07-31  0.000149
2019-08-31 -0.034966

[239 rows x 1 columns]
```

[ ]:

```
estimation.predict()
```

[ ]:

```
array([ 3.05551696e-03,  2.17148586e-03,  3.01106532e-03,  1.69086786e-04,
        2.60293623e-04,  5.48331513e-03,  3.06719614e-03,  2.23562613e-03,
        5.88892111e-03,  5.54243843e-03,  7.65404902e-03,  5.70635613e-03,
        5.98846628e-03,  6.25153827e-03,  6.29369825e-03,  7.48757602e-03,
        6.30080549e-03,  1.17075844e-02,  1.00729125e-02,  9.93172197e-03,
        6.78732152e-03,  9.28159792e-03,  9.36212441e-03,  6.72938628e-03,
        9.88536118e-03,  8.16250963e-03,  4.03954203e-03,  3.87028985e-03,
        4.49743098e-03,  6.32637432e-03,  7.11101992e-03,  7.26714905e-03,
        4.78271349e-03,  4.61815564e-03,  7.30384201e-03,  4.51986264e-03,
        5.40477539e-03, -3.22943343e-04, -6.11326060e-05, -2.31802753e-03,
        7.84875390e-03,  6.95699491e-03,  9.60476480e-03,  1.50125964e-02,
        1.35925831e-02,  7.30940258e-03,  7.52513080e-03,  7.49161072e-03,
        6.59584215e-03, -3.16317760e-03, -5.95051748e-04,  1.09349275e-03,
        1.24012147e-02,  1.29240774e-02,  7.14485461e-03,  5.07143040e-03,
        6.06628025e-03,  5.79322027e-03,  4.15008586e-03,  4.35120608e-03,
        5.57269984e-03, -3.05878026e-03, -9.73435000e-04, -1.25427086e-03,
        5.29019845e-03,  3.68682469e-03,  5.19103631e-03,  6.44376474e-03,
        6.87501671e-03,  5.44279556e-03,  4.60878390e-03,  2.36212501e-03,
       -3.15109632e-04, -8.97792916e-03, -8.61292758e-03, -8.42352667e-03,
        2.32458707e-03,  4.92321921e-04, -6.21665186e-04, -4.81517631e-03,
       -5.03842178e-03, -3.63814098e-03, -2.49880865e-03, -8.56490476e-04,
        5.71110665e-04, -1.13576797e-02, -1.28450063e-02, -1.29085999e-02,
        3.47266704e-03,  3.13409073e-03,  4.79678988e-03, -2.31015484e-03,
       -4.10726004e-03, -3.18100460e-03, -4.92393354e-03, -3.92301956e-03,
       -2.46060102e-03, -3.50303430e-03, -4.95307927e-03, -4.72931410e-03,
```

```
 3.59656688e-03,  3.75504419e-03,  3.30595703e-03, -3.84363421e-04,
-3.86049192e-03, -4.11503679e-03, -3.87702777e-03, -9.33053002e-03,
-2.10069579e-02, -2.07380796e-02, -1.51635598e-02, -2.64623981e-02,
-1.79160762e-02, -1.03820545e-02, -8.11582722e-03, -7.52826669e-03,
-4.17922908e-03, -2.26651643e-04,  2.09777308e-03,  5.01184004e-03,
 3.56076845e-03,  7.56600051e-03, -3.44849018e-03, -3.31256286e-03,
-2.01701958e-03,  7.41878717e-03,  9.31445439e-03,  8.99247328e-03,
 4.88176012e-03,  8.06158461e-03,  9.37267866e-03,  6.96260432e-03,
 8.03011980e-03,  6.51018142e-03,  1.69004570e-03,  2.16850229e-03,
 3.27291203e-03,  9.54654177e-03,  1.20290540e-02,  1.33511486e-02,
 6.66163367e-03,  8.58767886e-03,  1.04426924e-02,  9.30191047e-03,
 1.26998490e-02,  1.28655603e-02,  4.04899305e-03,  7.12252179e-03,
 7.99630856e-03,  1.75799863e-02,  1.95894699e-02,  1.97916452e-02,
 1.56124804e-02,  1.75088618e-02,  1.79958054e-02,  1.24642414e-02,
 1.11199033e-02,  1.19771755e-02, -4.44475550e-03, -3.03227863e-03,
-1.41084465e-03,  1.97379424e-02,  2.18577478e-02,  1.92474410e-02,
 7.55597830e-03,  9.18646377e-03,  6.19292127e-03,  8.80140252e-03,
 1.12311084e-02,  1.10076597e-02,  2.67950728e-03,  3.88334986e-03,
 6.02508315e-03,  1.32047462e-02,  1.40367270e-02,  1.62460209e-02,
 1.12424344e-02,  1.04401941e-02,  1.37617039e-02,  1.17329049e-02,
 1.39101403e-02,  1.63581777e-02,  7.34395616e-03,  9.90637349e-03,
 9.93349340e-03,  1.84300163e-02,  1.79893010e-02,  1.78120137e-02,
 1.06319594e-02,  1.29662821e-02,  9.79891128e-03,  9.54835473e-03,
 1.25661963e-02,  1.22367468e-02,  2.77351966e-03,  2.64011564e-03,
 4.02766903e-03,  1.42031265e-02,  1.38863987e-02,  1.45600182e-02,
 1.61339684e-02,  1.79037402e-02,  1.69800269e-02,  1.64908539e-02,
 1.39102826e-02,  1.12691817e-02,  1.40285703e-03,  2.28163202e-03,
 4.77063689e-03,  1.59973744e-02,  1.72153776e-02,  1.83576938e-02,
 1.20181819e-02,  1.29836513e-02,  1.44762122e-02,  1.15975062e-02,
 1.22688759e-02,  1.33253900e-02,  9.68762696e-03,  9.82426294e-03,
 6.98769772e-03,  1.52335236e-02,  1.37750981e-02,  1.53956929e-02,
 1.04609074e-02,  1.11780567e-02,  1.31851647e-02,  9.29140948e-03,
 5.59762931e-03,  7.60308902e-03,  1.38138841e-03,  5.66112361e-03,
 6.51335104e-03,  1.60179818e-02,  1.68231557e-02,  1.71419412e-02,
 1.43807427e-02,  1.49703253e-02,  1.82222620e-02])
```