Homework #3.1

In [ ]:
```
%matplotlib inline
import math
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm

from sklearn.metrics import r2_score
```

In [ ]:
```
# read data
securities = pd.read_excel('data/hw_3_1_data.xlsx', sheet_name='security return
portfolio = pd.read_excel('data/hw_3_1_data.xlsx', sheet_name='portfolio return
data = pd.merge(securities, portfolio, on='Date')
```

1. Regression

1. Estimation of regression of the portfolio return on SPY

In [ ]:
```
# Define and train model
y = data.portfolio
X = data.SPY
model = sm.OLS(y, sm.add_constant(X))
results = model.fit()
results.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	portfolio	**R-squared:**	0.745
Model:	OLS	**Adj. R-squared:**	0.743
Method:	Least Squares	**F-statistic:**	455.5
Date:	Sat, 18 Jun 2022	**Prob (F-statistic):**	3.93e-48
Time:	22:42:52	**Log-Likelihood:**	443.21
No. Observations:	158	**AIC:**	-882.4
Df Residuals:	156	**BIC:**	-876.3
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-0.0006	0.001	-0.479	0.633	-0.003	0.002
SPY	0.6142	0.029	21.342	0.000	0.557	0.671

Omnibus:	21.658	**Durbin-Watson:**	1.946
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	65.437
Skew:	0.448	**Prob(JB):**	6.17e-15
Kurtosis:	6.023	**Cond. No.**	24.6

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

From the summary we have: </br> $\alpha = -6 * 10^{-4}$ </br> $\beta = 0.6142$ </br> $R^2 = 0.745$ </br>

2. Estimation of regression of portfolio on SPY and HYG

In [ ]:

```
# Define and train model
X =  data[['SPY', 'HYG']]
model = sm.OLS(y, sm.add_constant(X))
results = model.fit()
results.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	portfolio	**R-squared:**	0.832
Model:	OLS	**Adj. R-squared:**	0.830
Method:	Least Squares	**F-statistic:**	384.2
Date:	Sat, 18 Jun 2022	**Prob (F-statistic):**	8.57e-61
Time:	22:42:52	**Log-Likelihood:**	476.28
No. Observations:	158	**AIC:**	-946.6
Df Residuals:	155	**BIC:**	-937.4
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.0009	0.001	-0.915	0.362	-0.003	0.001
SPY	0.3843	0.035	11.072	0.000	0.316	0.453
HYG	0.5166	0.058	8.977	0.000	0.403	0.630

Omnibus:	1.294	**Durbin-Watson:**	2.246
Prob(Omnibus):	0.524	**Jarque-Bera (JB):**	1.296
Skew:	0.214	**Prob(JB):**	0.523
Kurtosis:	2.886	**Cond. No.**	66.9

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

From the summary we have: </br> $\alpha = -9 * 10^{-4}$ </br> $\tilde{\beta}^{spy} = 0.3843$ </br> $\tilde{\beta}^{hyg} = 0.5166$ </br> $R^2 = 0.832$ </br>

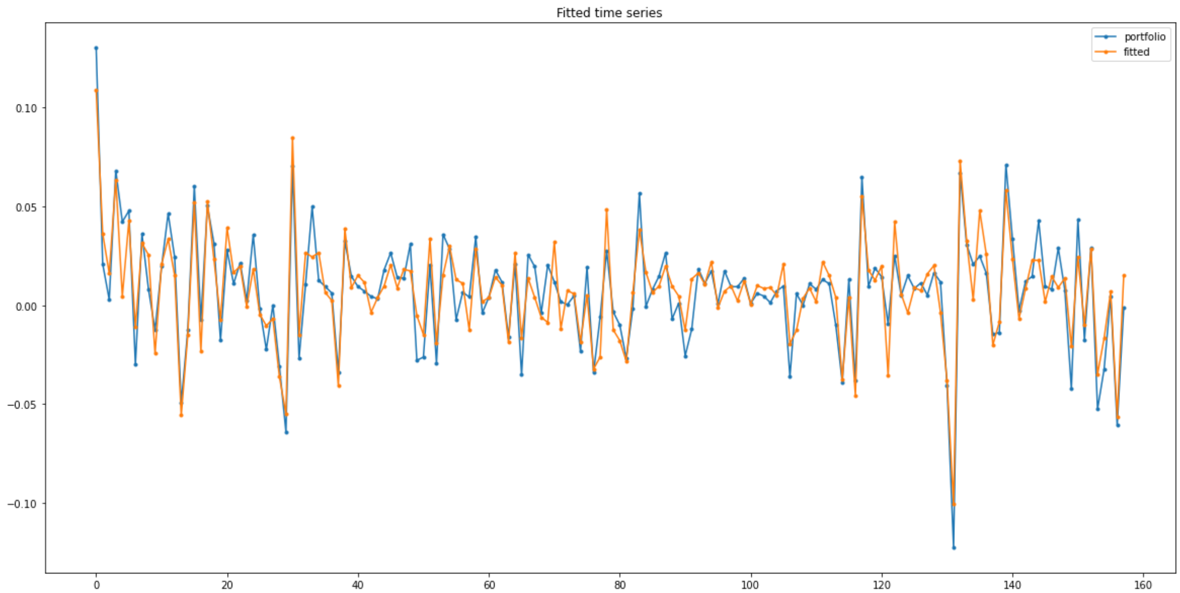
3. Time series of fitted regression

In [ ]:

```
fitted = pd.concat([y, pd.DataFrame(data=results.predict(sm.add_constant(X)), i
fitted.plot(title='Fitted time series', figsize=(20, 10), marker='.')
```

Out[ ]:

```
<AxesSubplot:title={'center':'Fitted time series'}>
```



Correlation between $\hat{r}^p_t$ and r^p_t

In [ ]:
```
fitted.corr()['fitted'].iloc[0]
```

Out[ ]:
```
0.9122188135325914
```

The r-squared of the regression in equation (2) is **0.832**, which is squared correlation.
$R^2 = corr(\hat{r}^p_t, r^p_t)^2$
$0.912^2 = 0.832$

4.

In [ ]:
```
data[['SPY', 'HYG']].corr()['SPY'].iloc[1]
```

Out[ ]:
```
0.7380170112481026
```

$\beta^{spy} = 0.6142$ in (1) and $\tilde{\beta}^{spy} = 0.3843$ in (2). Hence, β^{spy} is greater than $\tilde{\beta}^{spy}$ </br> The correlation between SPY and HYG is 0.738, which is fairly high. This means that the addition of HYG as a regressor "dilutes" the contribution of SPY given the high correlation between both regressors.

5.

ϵ_t should have higher correlation with r^{hyg}_t because the contribution of HYG isn't accounted for in equation (1)

2. Decomposing and Replicating

1.

Let's first fit our model with all the assets.

In [ ]:
```
# Define and train model
X = data.drop(columns=['Date', 'portfolio'])
y = data.portfolio
model = sm.OLS(y, sm.add_constant(X))
results = model.fit()
results.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	portfolio	**R-squared:**	1.000
Model:	OLS	**Adj. R-squared:**	1.000
Method:	Least Squares	**F-statistic:**	7.609e+28
Date:	Sat, 18 Jun 2022	**Prob (F-statistic):**	0.00
Time:	22:42:53	**Log-Likelihood:**	5392.1
No. Observations:	158	**AIC:**	-1.076e+04
Df Residuals:	145	**BIC:**	-1.072e+04
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.526e-16	3.94e-17	8.960	0.000	2.75e-16	4.3e-16
SPY	-5.274e-16	2.12e-15	-0.248	0.804	-4.72e-15	3.67e-15
EFA	-8.361e-16	2.04e-15	-0.410	0.683	-4.87e-15	3.2e-15
EEM	-5.794e-16	1.23e-15	-0.471	0.638	-3.01e-15	1.85e-15
PSP	0.2500	1.53e-15	1.64e+14	0.000	0.250	0.250
QAI	0.2500	5.48e-15	4.56e+13	0.000	0.250	0.250
HYG	-3.539e-16	2.46e-15	-0.144	0.886	-5.22e-15	4.52e-15
DBC	-6.713e-16	8.6e-16	-0.780	0.436	-2.37e-15	1.03e-15
IYR	0.2500	1.03e-15	2.42e+14	0.000	0.250	0.250
IEF	0.2500	3.97e-15	6.3e+13	0.000	0.250	0.250
BWX	-1.527e-16	2.44e-15	-0.062	0.950	-4.98e-15	4.68e-15
TIP	-1.388e-16	4.2e-15	-0.033	0.974	-8.44e-15	8.16e-15
SHV	-2.019e-15	4.31e-14	-0.047	0.963	-8.72e-14	8.32e-14

Omnibus:	13.115	**Durbin-Watson:**	0.533
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	30.155
Skew:	-0.262	**Prob(JB):**	2.83e-07
Kurtosis:	5.075	**Cond. No.**	1.42e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.42e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

We get $R^2 = 1$ but by looking at the t-stats PSP, QAI, IYR and IEF seem to be the most important regressors. </br> Also, by taking a look at the correlation among the dependent variable we can see that these securities have strong correlations with other assets such as SPY, EFA and EEM for instance. </br> Let's run the regression by only keeping PSP, QAI, IYR, IEF and see if we can replicate the R^2

In [ ]:
```
# Correlation accross dependant variables
X.corr()
```

Out[ ]:

	SPY	EFA	EEM	PSP	QAI	HYG	DBC	IYR
SPY	1.000000	0.870442	0.745782	0.898231	0.824195	0.738017	0.494877	0.729380
EFA	0.870442	1.000000	0.851419	0.904901	0.830989	0.754130	0.578556	0.669529
EEM	0.745782	0.851419	1.000000	0.796636	0.800203	0.745496	0.558936	0.602920
PSP	0.898231	0.904901	0.796636	1.000000	0.816837	0.811428	0.485034	0.735933
QAI	0.824195	0.830989	0.800203	0.816837	1.000000	0.747440	0.528369	0.609705
HYG	0.738017	0.754130	0.745496	0.811428	0.747440	1.000000	0.460296	0.736794
DBC	0.494877	0.578556	0.558936	0.485034	0.528369	0.460296	1.000000	0.282182
IYR	0.729380	0.669529	0.602920	0.735933	0.609705	0.736794	0.282182	1.000000
IEF	-0.328991	-0.311078	-0.253117	-0.304326	-0.076174	-0.154525	-0.414037	-0.060927
BWX	0.396519	0.555328	0.605680	0.479873	0.627331	0.506193	0.325511	0.384560
TIP	0.133462	0.150721	0.227128	0.166647	0.362915	0.228485	0.064877	0.284087
SHV	-0.188703	-0.164626	-0.108666	-0.197424	-0.111890	-0.127904	-0.186239	-0.136566

In [ ]:
```
# Fitting with only 'PSP', 'QAI', 'IYR', 'IEF'
model = sm.OLS(y, sm.add_constant(X[['PSP', 'QAI', 'IYR', 'IEF']]))
results = model.fit()
results.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	portfolio	**R-squared:**	1.000
Model:	OLS	**Adj. R-squared:**	1.000
Method:	Least Squares	**F-statistic:**	2.195e+31
Date:	Sat, 18 Jun 2022	**Prob (F-statistic):**	0.00
Time:	22:42:53	**Log-Likelihood:**	5748.5
No. Observations:	158	**AIC:**	-1.149e+04
Df Residuals:	153	**BIC:**	-1.147e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-7.806e-18	3.24e-18	-2.411	0.017	-1.42e-17	-1.41e-18
PSP	0.2500	1.15e-16	2.18e+15	0.000	0.250	0.250
QAI	0.2500	4.09e-16	6.11e+14	0.000	0.250	0.250
IYR	0.2500	9.01e-17	2.77e+15	0.000	0.250	0.250
IEF	0.2500	2.07e-16	1.21e+15	0.000	0.250	0.250

Omnibus:	42.870	**Durbin-Watson:**	2.049
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	246.405
Skew:	0.787	**Prob(JB):**	3.12e-54
Kurtosis:	8.912	**Cond. No.**	138.

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

With only the four assets PSP, QAI, IYR and IEF we are able to perfectly capture the variations in the portfolio with $R^2 = 1$ and very small p-values. </br> The weights are: </br> $\beta^{PSP} = 0.25$ </br> $\beta^{QAI} = 0.25$ </br> $\beta^{IYR} = 0.25$ </br> $\beta^{IEF} = 0.25$ </br>

2. Regression on TIP using 2018 data

In [ ]:

```
data_2018 = data.loc[data.Date.dt.year < 2019]
X = data_2018.drop(columns=['Date', 'TIP', 'portfolio'])
y = data_2018.TIP

model = sm.OLS(y, sm.add_constant(X))
results = model.fit()
results.summary()
```

Out[ ]:

OLS Regression Results

Dep. Variable:	TIP	**R-squared:**	0.699
Model:	OLS	**Adj. R-squared:**	0.667
Method:	Least Squares	**F-statistic:**	22.16
Date:	Sat, 18 Jun 2022	**Prob (F-statistic):**	1.17e-22
Time:	22:42:54	**Log-Likelihood:**	410.59
No. Observations:	117	**AIC:**	-797.2
Df Residuals:	105	**BIC:**	-764.0
Df Model:	11		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	0.0012	0.001	1.199	0.233	-0.001	0.003
SPY	-0.0385	0.057	-0.670	0.505	-0.152	0.075
EFA	-0.0213	0.048	-0.442	0.660	-0.117	0.074
EEM	0.0676	0.028	2.424	0.017	0.012	0.123
PSP	0.0291	0.038	0.769	0.444	-0.046	0.104
QAI	0.0880	0.127	0.695	0.489	-0.163	0.339
HYG	-0.0805	0.060	-1.347	0.181	-0.199	0.038
DBC	0.0759	0.021	3.600	0.000	0.034	0.118
IYR	0.0250	0.026	0.950	0.344	-0.027	0.077
IEF	0.6629	0.072	9.201	0.000	0.520	0.806
BWX	-4.003e-05	0.056	-0.001	0.999	-0.111	0.111
SHV	-1.8066	1.448	-1.248	0.215	-4.677	1.064

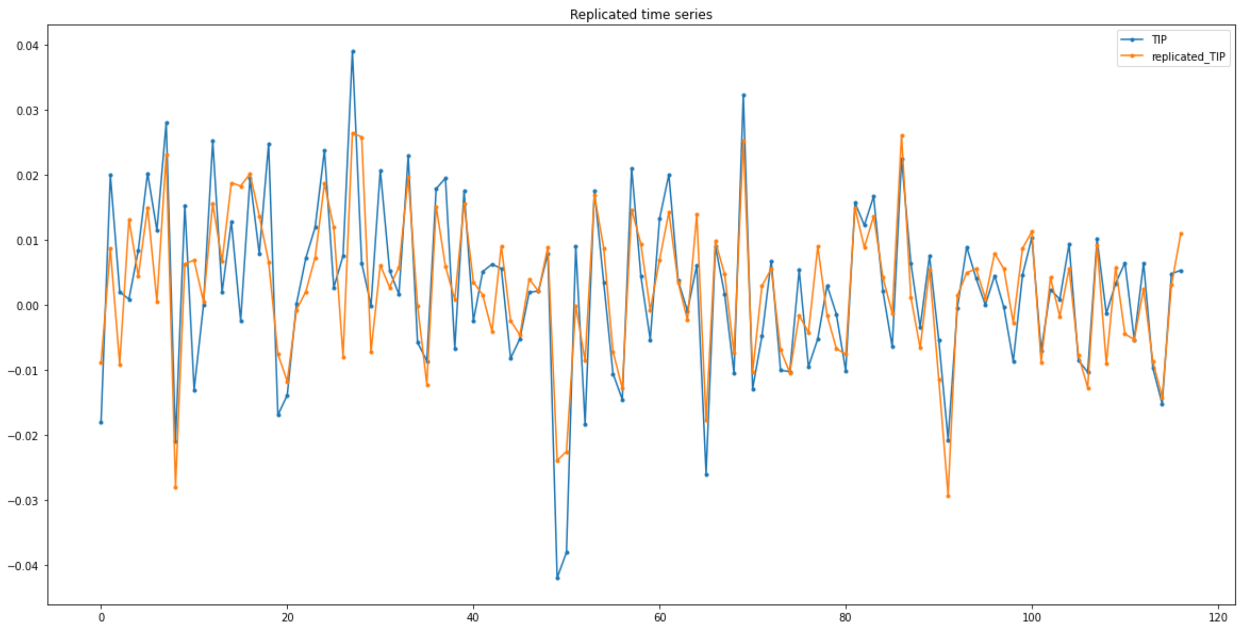
Omnibus:	4.571	**Durbin-Watson:**	2.100
Prob(Omnibus):	0.102	**Jarque-Bera (JB):**	4.120
Skew:	-0.342	**Prob(JB):**	0.127
Kurtosis:	3.614	**Cond. No.**	2.05e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.05e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

(a) $R^2 = 0.669$ </br> β is the **coef** column (except const) in the OLS summary above. </br> (b) t-stats can be seen in the t column in the summary. The ones with an absolute value greater than 2 are: EEM, DBC and IEF. </br> (c)

In [ ]:

```
replication = pd.concat([y, pd.DataFrame(data=results.predict(sm.add_constant(X
replication.plot(title='Replicated time series', figsize=(20, 10), marker='.')
plt.show()
```



3. out-of-sample replication on 2019-2020

In [ ]:

```
data_2019_2022 = data.loc[data.Date.dt.year >= 2019]
X = data_2019_2022.drop(columns=['Date', 'TIP', 'portfolio'])
y = data_2019_2022.TIP
oos_replicated_TIP = pd.concat([y, pd.DataFrame(data=results.predict(sm.add_cor
```

(a) Correlation between $\hat{r}_t^{TIPoos}$ and r_t^{oos}

In [ ]:

```
corr = math.sqrt(r2_score(y, oos_replicated_TIP.oos_replicated_TIP)); corr
```

Out[ ]:

```
0.7791907101334139
```

(b) From the R^2 from the 2018 regression we can derive the correlation which is: </br> $corr_{2018} = \sqrt{(}R^2) = 0.818$ </br> This is slightly above the 0.780 correlation obtained on the out-of-sample replication. This is reasonable and makes a lot of sense since the out-of-sample is data not observed during fitting.