# Python:
# Linear Regression with Statsmodels

November 2021

THE MARQUEE GROUP

TheMarqueeGroup.com
info@themarqueegroup.com

# The Marquee Group



## Leaders in Financial Modeling Since 2002

- We believe that spreadsheet-based financial models are the most important decision-making tools in modern finance
- We have developed a framework and discipline for model design, structure and development that leads to best-in-class, user-friendly financial models
- We help finance professionals use this framework to turn their models into powerful communication tools that lead to better, more effective decisions

### The Marquee Group Offering

| TRAINING | CONSULTING | ACCREDITATION |
|---|---|---|
| ✓ Instructors have real-world experience and a passion for teaching | ✓ Services include:<br>– Model Development<br>– Model Re-builds<br>– Model Reviews<br>– Model Audits | ✓ Offered by the Financial Modeling Institute (FMI) |
| ✓ Topics include: Modeling, Valuation, Excel, Python | ✓ Clients include a wide range of companies in various industries | ✓ The Marquee Group was one of the founders of the FMI |
| ✓ Courses are interactive | | ✓ FMI administers official exams for three levels of financial modeling certifications |
| ✓ Clients include banks, corporations, business schools and societies | | |

# Presenter Bio

## BOGDAN TUDOSE – Principal, The Marquee Group

As a Principal and Instructor, Bogdan teaches more than ten different courses throughout the year. He also leads Marquee's Data Sciences practice and has helped create many new courses, including three day-long sessions on Python for Finance Professionals and new technical content for various bank training programs.

Bogdan has worked in the capital markets for over ten years. Prior to Marquee, he started his career as an Analyst/Associate in investment banking in the Mergers & Acquisitions Group at BMO Capital Markets before joining Anson Funds, a long-short hedge fund in Toronto as an Investment Analyst.

At BMO Capital Markets, Bogdan was actively involved in 14 live transactions spanning numerous industries which included Infrastructure, Metals & Mining, Financial Institutions, Media & Communications, Retail and Oil & Gas.

At Anson Funds, Bogdan developed and maintained in-depth financial models for the fund's largest positions and conducted extensive due diligence, including meeting with management teams, industry experts, and buy-side and sell-side analysts. He was responsible for sourcing and developing investment ideas and recommending trade and risk mitigation strategies for the portfolio.

# Using *statsmodels*

# Using *statsmodels*

- Statsmodels is a python package that provides functions to perform linear and time series analysis
  - Similar to R for linear regression and time series
- This lesson will focus on
  - Linear Regression (CAPM)
  - Time Series (ARMA)
  - Visualizing Analysis

```
# %% Packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import statsmodels.api as sm
import statsmodels.tsa as tsa
```

# Using *statsmodels*

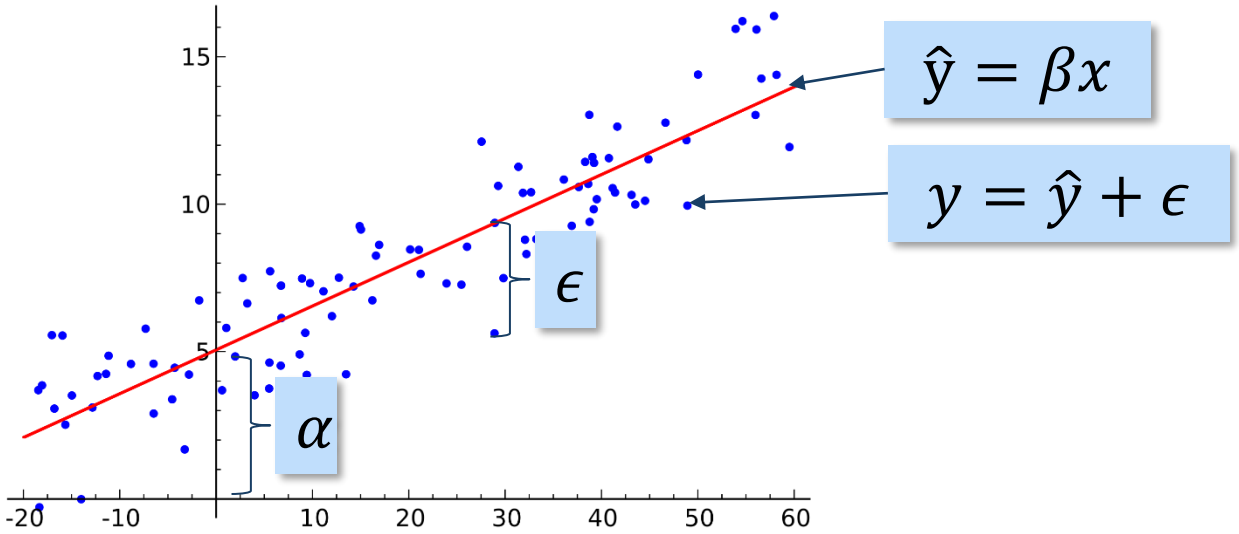
- Organizing data
  - It is best to organize the data into a single DataFrame
  - Verify the index is aligned, then handle missing data accordingly
- After the data is cleaned, copy the DataFrame into an array and if necessary scale the data and check for outliers
  - For numerical stability, try to keep numbers close to one
  - There are several techniques to handle outliers
    - Normalization: Transform the data to be zero mean and unit variance
    - Winsorizing: replace the outliers with the value contained at 90th, 95th or 99th percentile
    - Log Transform: take the log of the data

```
>>> sp500 = pd.read_csv('StockData/SP500.csv', header=0, index_col=0, parse_dates=True)
>>> aapl = pd.read_csv('StockData/aapl.csv', header=0, index_col=0, parse_dates=True)
# Need to cast the series as a DataFrame to merge
>>> data = pd.DataFrame(sp500['Adj Close']).merge(aapl['Adj Close’],
                                  left_index=True, right_index=True)
>>> data.columns = ['sp500', 'aapl']
>>> data[['sp500_rtns', 'aapl_rtns']] = np.log(data[['sp500’, 'aapl']]).diff()
>>> data[['sp500_rtns', 'aapl_rtns']] *= 100
>>> data.dropna(inplace=True)
```

# Linear Regression with *statsmodels*

# Linear Regression

- The simplest form of linear regression is finding the line of “best fit” between two variables, the dependent and independent variable
  - Traditionally the dependent variable is called endogenous and labelled as $y$
  - The independent variables are called exogenous and labelled as $x$
  - The error term is what is left over, not explained by the exogenous variable, and is labelled as $\epsilon$
  - The general form equation is $y = \alpha + \beta x + \epsilon$
  - The slope, beta, is the sensitivity of the dependent variable to the independent variable
    - Typically thought of as a unit change in x results in a beta change of y
  - Best fit is found by minimizing the least squares objective function $\min_{\beta} \sum(y_i - \beta x_i)^2$



# Linear Regression

- This simple two variable model can be expanded into a multivariate model
  - The equation doesn't change, $y = X\beta + \epsilon$, but now $X$ is a matrix with multiple variables used to explain $y$ including a column of ones for the intercept
- The following assumptions are made for the model
  - **The model is correctly specified:** a linear model is specified, thus the relationship in reality should be linear (not exponential, square etc.)
  - **Exogeneity**: $E[\epsilon|X] = 0$ the conditional mean of the error term is zero based on the exogenous variables. Be aware for this assumption that the objective function will make this true but its not necessarily true
  - **Linearly independent**: Full rank system, no perfect multicollinearity between variables
  - **Errors are uncorrelated and homoscedastic** : $E\left[\epsilon_i \epsilon_j \middle| X\right] = 0 \; for \; i \neq j$ and $E\left[\epsilon_i^2 \middle| X\right] = \sigma^2$, meaning the error terms are not changing with $x$ and the variance is constant throughout the data (there is no value of x that has a smaller error variance)
  - **Error terms are normally distributed**: $\epsilon|X \sim N(0, \sigma^2)$, used to impose the asymptotic properties to determine the standard errors and confidence intervals on the betas

# Linear Regression

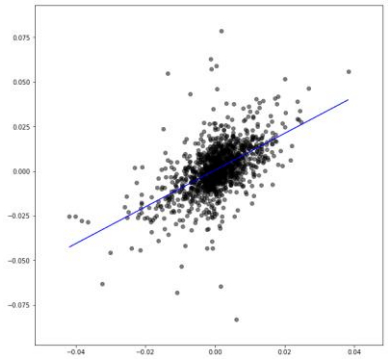
- The CAPM can be estimated using linear regression
  - $E(r_i) = r_f + \beta(E(r_m) - r_f)$
  - Rearrange to make it easier to estimate: $E(r_i) - r_f = \beta(E(r_m) - r_f)$
- Syntax: sm.OLS(endo, exog)
  - **Endo**: the dependent variable, y or left hand side
  - **Exog**: the independent variable(s), X or right hand side
- An intercept is not included in this model
  - To add one use the sm.add_constant() function

```
# Performing a simple/modified CAPM, not going to subtract rf
>>> capm = sm.OLS(data['aapl_rtns'], data['sp500_rtns'])
# y, X -> apple, sp500 (endogenous, exogenous)
>>> results = capm.fit()

>>> print(results.params)
[1.03475198]
# This means that Apple has a beta of 1.0347 to the market (sp500)
```

# Linear Regression

- To plot the linear regression, generate predicted values for Y, which in this example are the returns for Apple
  - Calculating $\hat{y} = \beta x$, or more specifically $\widehat{r_{aapl}} = \beta r_{mrkt}$



```
>>> data['aapl_hat'] = results.predict(data['sp500_rtns'])
# Plot the scatter plot and the linear regression
>>> plt.subplots(figsize=(10, 10))
>>> plt.scatter(data['sp500_rtns’],
                  data['aapl_rtns'], color='black', alpha=0.5)
>>> plt.plot(data['sp500_rtns’],
                  data['aapl_hat'], color='blue', linewidth=1)
>>> plt.show()
```

# Linear Regression

- When a linear regression is performed the coefficients are a point estimate; the confidence interval must be reviewed to learn if they are significant
  - The statsmodels package will automatically provide summary statistics, using the .summary() method for the results object

```
>>> print(results.summary())
OLS Regression Results
==============================================================================
Dep. Variable:              aapl_rtns   R-squared:                       0.322
Model:                            OLS   Adj. R-squared:                  0.322
Method:                 Least Squares   F-statistic:                     598.1
Date:                Thu, 02 May 2019   Prob (F-statistic):          2.28e-108
Time:                        17:47:34   Log-Likelihood:                -1979.4
No. Observations:                1258   AIC:                             3961.
Df Residuals:                    1257   BIC:                             3966.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
sp500_rtns     1.0348      0.042     24.456      0.000       0.952       1.118
==============================================================================
Omnibus:                      209.489   Durbin-Watson:                   1.933
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3680.503
Skew:                           0.080   Prob(JB):                         0.00
Kurtosis:                      11.378   Cond. No.                         1.00
==============================================================================
```

For more information on
The Marquee Group:

56 Temperance Street, Suite 801
Toronto, ON  M5H 3V5

TheMarqueeGroup.com
info@themarqueegroup.com

+1 416 583 1802

THE MARQUEE GROUP