Task 1 – Example Answer with Python Code

**import** os

cwd **=** os**.**getcwd()

print("Current working directory: {0}"**.**format(cwd))

print ("os.getcwd() returns an object of type {0}"**.**format(type(cwd)))

*# copy the filepath*

os**.**chdir ("\_\_\_\_\_\_\_\_")

*# let's jump into task 1*

**import** pandas **as** pd

**import** numpy **as** np

**from** matplotlib **import** pyplot **as** plt

**from** datetime **import** date, timedelta

date\_time **=** ["10-2020", "11-2020", "12-2020"]

date\_time **=** pd**.**to\_datetime(date\_time)

data **=** [1, 2, 3]

df **=** pd**.**read\_csv('natgas\_R.csv', parse\_dates**=**['Dates'])

prices **=** df['Prices']**.**values

dates **=** df['Dates']**.**values

*# plot prices against dates*

fig, ax **=** plt**.**subplots()

ax**.**plot\_date(dates, prices, '-')

ax**.**set\_xlabel('Date')

ax**.**set\_ylabel('Price')

ax**.**set\_title('Natural Gas Prices')

ax**.**tick\_params(axis**=**'x', rotation**=**45)

plt**.**show()

*# From the plot - we can see the prices have a natural frequency of around a year, but trend upwards.*

*# We can do a linear regression to get the trend, and then fit a sin function to the variation in each year.*

*# First we need the dates in terms of days from the start, to make it easier to interpolate later.*

start\_date **=** date(2020,10,31)

end\_date **=** date(2024,9,30)

months **=** []

year **=** start\_date**.**year

month **=** start\_date**.**month **+** 1

**while** **True**:

current **=** date(year, month, 1) **+** timedelta(days**=-**1)

months**.**append(current)

**if** current**.**month **==** end\_date**.**month **and** current**.**year **==** end\_date**.**year:

**break**

**else**:

month **=** ((month **+** 1) **%** 12) **or** 12

**if** month **==** 1:

year **+=** 1

days\_from\_start **=** [(day **-** start\_date )**.**days **for** day **in** months]

*# Simple regression for the trend will fit to a model y = Ax + B. The estimator for the slope is given by \hat{A} = \frac{\sum (x\_i - \bar{x})(y\_i - \bar{y})}{\sum (x\_i - \bar{x})^2},*

*# and that for the intercept by \hat{B} = \bar{y} - hat{A} \* \xbar*

**def** simple\_regression(x, y):

xbar **=** np**.**mean(x)

ybar **=** np**.**mean(y)

slope **=** np**.**sum((x **-** xbar) **\*** (y **-** ybar))**/** np**.**sum((x **-** xbar)**\*\***2)

intercept **=** ybar **-** slope**\***xbar

**return** slope, intercept

time **=** np**.**array(days\_from\_start)

slope, intercept **=** simple\_regression(time, prices)

*# Plot linear trend*

plt**.**plot(time, prices)

plt**.**plot(time, time **\*** slope **+** intercept)

plt**.**xlabel('Days from start date')

plt**.**ylabel('Price')

plt**.**title('Linear Trend of Monthly Input Prices')

plt**.**show()

print(slope, intercept)

*# From this plot we see the linear trend has been captured. Now to fit the intra-year variation.*

*# Given that natural gas is used more in winter, and less in summer, we can guess the frequency of the price movements to be about a year, or 12 months.*

*# Therefore we have a model y = Asin( kt + z ) with a known frequency.Rewriting y = Acos(z)sin(kt) + Asin(z)cos(kt),*

*# we can use bilinear regression, with no intercept, to solve for u = Acos(z), w = Asin(z)*

sin\_prices **=** prices **-** (time **\*** slope **+** intercept)

sin\_time **=** np**.**sin(time **\*** 2 **\*** np**.**pi **/** (365))

cos\_time **=** np**.**cos(time **\*** 2 **\*** np**.**pi **/** (365))

**def** bilinear\_regression(y, x1, x2):

*# Bilinear regression without an intercept amounts to projection onto the x-vectors*

slope1 **=** np**.**sum(y **\*** x1) **/** np**.**sum(x1 **\*\*** 2)

slope2 **=** np**.**sum(y **\*** x2) **/** np**.**sum(x2 **\*\*** 2)

**return**(slope1, slope2)

slope1, slope2 **=** bilinear\_regression(sin\_prices, sin\_time, cos\_time)

*# We now recover the original amplitude and phase shift as A = slope1 \*\* 2 + slope2 \*\* 2, z = tan^{-1}(slope2/slope1)*

amplitude **=** np**.**sqrt(slope1 **\*\*** 2 **+** slope2 **\*\*** 2)

shift **=** np**.**arctan2(slope2, slope1)

*# Plot smoothed estimate of full dataset*

plt**.**plot(time, amplitude **\*** np**.**sin(time **\*** 2 **\*** np**.**pi **/** 365 **+** shift))

plt**.**plot(time, sin\_prices)

plt**.**title('Smoothed Estimate of Monthly Input Prices')

*# Define the interpolation/extrapolation function*

**def** interpolate(date):

days **=** (date **-** pd**.**Timestamp(start\_date))**.**days

**if** days **in** days\_from\_start:

*# Exact match found in the data*

**return** prices[days\_from\_start**.**index(days)]

**else**:

*# Interpolate/extrapolate using the sin/cos model*

**return** amplitude **\*** np**.**sin(days **\*** 2 **\*** np**.**pi **/** 365 **+** shift) **+** days **\*** slope **+** intercept

*# Create a range of continuous dates from start date to end date*

continuous\_dates **=** pd**.**date\_range(start**=**pd**.**Timestamp(start\_date), end**=**pd**.**Timestamp(end\_date), freq**=**'D')

*# Plot the smoothed estimate of the full dataset using interpolation*

plt**.**plot(continuous\_dates, [interpolate(date) **for** date **in** continuous\_dates], label**=**'Smoothed Estimate')

*# Fit the monthly input prices to the sine curve*

x **=** np**.**array(days\_from\_start)

y **=** np**.**array(prices)

fit\_amplitude **=** np**.**sqrt(slope1 **\*\*** 2 **+** slope2 **\*\*** 2)

fit\_shift **=** np**.**arctan2(slope2, slope1)

fit\_slope, fit\_intercept **=** simple\_regression(x, y **-** fit\_amplitude **\*** np**.**sin(x **\*** 2 **\*** np**.**pi **/** 365 **+** fit\_shift))

plt**.**plot(dates, y, 'o', label**=**'Monthly Input Prices')

plt**.**plot(continuous\_dates, fit\_amplitude **\*** np**.**sin((continuous\_dates **-** pd**.**Timestamp(start\_date))**.**days **\*** 2 **\*** np**.**pi **/** 365 **+** fit\_shift) **+** (continuous\_dates **-** pd**.**Timestamp(start\_date))**.**days **\*** fit\_slope **+** fit\_intercept, label**=**'Fit to Sine Curve')

plt**.**xlabel('Date')

plt**.**ylabel('Price')

plt**.**title('Natural Gas Prices')

plt**.**legend()

plt**.**show()