PythonTest

February 2, 2022

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
# Import dataset
data = pd.read_csv("https://raw.githubusercontent.com/lecriste/LDC/main/commodities.csv?token=GI
```

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19019 entries, 0 to 19018
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   date         19019 non-null  object 
 1   instrument   19019 non-null  object 
 2   maturity     19019 non-null  object 
 3   observation  19019 non-null  object 
 4   value        19019 non-null  float64
 5   currency     19019 non-null  object 
dtypes: float64(1), object(5)
memory usage: 891.6+ KB
```

In [4]:
```
# Check for single-value features
for feature in data:
    if data[feature].nunique() == 1:
        print(feature)
```

```
currency
```

In [5]:
```
currency = data['currency'][0]
```

In [6]:
```
data_orig = data
# Strip single-value feature
data = data_orig.drop('currency', axis=1)
```

In [7]:
```
# Convert date values to "datetime" format
data['date'] = pd.to_datetime(data['date'])
```

In [8]:
```
# List commodities
commodities = data.groupby(['instrument']).nunique()
commodities
```

Out[8]:

instrument	date	maturity	observation	value
CBOT.ZC	231	17	3	1047
CBOT.ZS	231	30	3	1931

```
In [9]: corn = [commodities.index[0], 'Corn']
        soy = [commodities.index[1], 'Soybeans']

In [10]: # Split dataset per commodity
         dataCorn = data[data['instrument'] == corn[0]].drop('instrument', axis=1)
         dataSoy = data[data['instrument'] == soy[0]].drop('instrument', axis=1)

In [11]: def sort_maturities(df):
             #maturities = df.groupby(['maturity']).nunique()
             #maturities.sort_index(key = lambda m: m[1:]+m[-1])
             maturities = df.groupby(['maturity']).nunique().index
             maturities_sorted = list(maturities)
             maturities_sorted.sort(key=lambda m: m[-1])
             return maturities_sorted

In [12]: def plotAll(comm, df, maturities_sorted):
             plt.figure(figsize=(16*2, 9*2))
             ax = df[df['maturity'] == 0].plot(x='date')
             ax.clear()
             plt.title(comm+" price by maturity")
             plt.ylabel(currency)

             df_dateIdx = df.set_index('date')
             linear = {}
             for m in maturities_sorted:
                 df[df['maturity'] == m].plot(x='date', ax=ax, legend=False)
                 #linear[m] = df['value'][df['maturity'] == m].interpolate()
                 #method='time': time-weighted interpolation only works on Series or DataFrames with a
                 linear[m] = df_dateIdx['value'][df_dateIdx['maturity'] == m].interpolate(method='time'
             plt.legend(maturities_sorted, bbox_to_anchor=(1., 1.1))

             return linear

In [13]: def minmax(comm, metric, data, error=False):
             minIdx = min(data, key=data.get)
             maxIdx = max(data, key=data.get)
             metrics = {'minimum': minIdx, 'maximum': maxIdx}
             for m in metrics:
                 err = ""
                 if error:
                     err = "+/- {:.0f} ".format(data[metrics[m]][1])
                 print("In 2021, the %s maturity-%s of %s price was %.0f %s%s for %s." \
                       % (m, metric, comm, data[metrics[m]][0], err, currency, metrics[m]))

In [14]: def getMean(comm, df, maturities_sorted):
             means = {}
             for m in maturities_sorted:
                 means[m] = (df['value'][df.maturity == m].mean(), \
                             df['value'][df.maturity == m].std())
             minmax(comm, "average", means, True)

             return means

In [15]: def getSpread(comm, df, linear, whole_year, days, verbose=False):
             spreads = {}
```

```
            for m in linear:
                obs = df['date'][df.maturity == m].count()
                spread = df['value'][df.maturity == m].max() - df['value'][df.maturity == m].min()
                if (df['date'][df['value'][df.maturity == m].idxmax()] <  df['date'][df['value'][df.ma
                    spread *= -1
                slope = (linear[m].iloc[-1] - linear[m].iloc[0]) / days
                spreads[m] = (spread, slope, obs)

                if verbose:
                    if obs < whole_year:
                        print("\nIn only part of 2021:")
                    else:
                        print("\nIn the whole 2021:")
                    print("the maximum spread of %s %s is %.2f %s," % (corn[1], m, spread, currency))
                    print("with an average trend of %+.2f %s per day." % (slope, currency))
            minmax(comm, "spread", spreads)

            return spreads
```

In [16]:
```
def plotByYear(comm, df, maturities_sorted):
    maturities_year = sorted(set([m[1:] for m in maturities_sorted]))
    for y in maturities_year:
        plt.figure(figsize=(16*2, 9*2))
        ax = df[df['maturity'] == 0].plot(x='date', ylabel=currency, legend=False)
        ax.clear()
        plt.title(comm+" price with maturity in "+y)
        plt.ylabel(currency)
        legend = []

        for m in maturities_sorted:
            if y in m:
                df[df['maturity'] == m].plot(x='date', y='value', ax=ax, legend=False)
                legend.append(m)
        plt.legend(legend, bbox_to_anchor=(1., 1.))
```

In [17]:
```
def inspectDataset(df, comm):
    print("Inspecting "+comm+" dataset:")
    print(df)
    print("\nGroup by maturity:")
    print(df.groupby(['maturity']).nunique())
    whole_year = df.groupby(['maturity']).nunique()['date'].max()
    days = (df['date'].max() - df['date'].min()).days

    df_Settle = df[df.observation == 'Settle']
    maturities_sorted = sort_maturities(df_Settle)

    # General overview
    linear = plotAll(comm, df_Settle, maturities_sorted)

    # Means
    print("\nCalculate average price per maturity:")
    means = getMean(comm, df_Settle, maturities_sorted)

    # Group by maturity year
    print("\nCalculate price spread per maturity:")
```

```
            plotByYear(comm, df_Settle, maturities_sorted)

            # Slopes
            spreads = getSpread(comm, df_Settle, linear, whole_year, days)

            # Summary
            metrics = pd.DataFrame(index=range(0, len(maturities_sorted))\
                                   , columns=("Maturity","Observations [%]","Spread","Slope (/day)","M
                                   , dtype=float)
            for m in metrics.index:
                mat = maturities_sorted[m]
                metrics.loc[m] = [mat, spreads[mat][2]/whole_year *100, spreads[mat][0], spreads[mat][
                                   , means[mat][0], means[mat][1]]

            return metrics
```

```
In [18]: metrics = inspectDataset(dataCorn, corn[1])
         #metrics.style.format(precision=0)
         metrics.round({"Observations [%]": 0,"Spread": 2, "Slope (/day)": 2, "Mean": 0, "Std": 0})
```

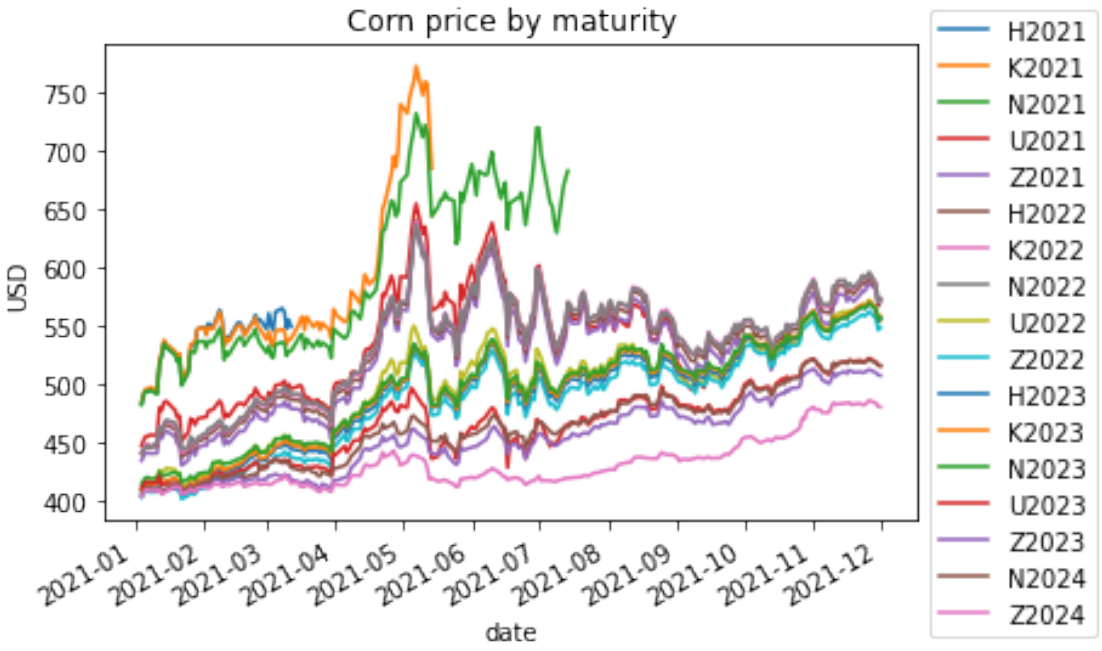
```
Inspecting Corn dataset:
                          date maturity observation   value
10215 2021-01-04 00:00:00+00:00    H2021        High  497.75
10216 2021-01-04 00:00:00+00:00    H2021         Low  479.50
10217 2021-01-04 00:00:00+00:00    H2021      Settle  483.75
10218 2021-01-04 00:00:00+00:00    K2021        High  497.25
10219 2021-01-04 00:00:00+00:00    K2021         Low  480.25
...                          ...      ...         ...     ...
19014 2021-12-01 00:00:00+00:00    Z2023        High  507.75
19015 2021-12-01 00:00:00+00:00    Z2023         Low  505.75
19016 2021-12-01 00:00:00+00:00    Z2023      Settle  507.50
19017 2021-12-01 00:00:00+00:00    N2024      Settle  515.75
19018 2021-12-01 00:00:00+00:00    Z2024      Settle  480.50

[8804 rows x 4 columns]

Group by maturity:
          date  observation  value
maturity
H2021       48            3    109
H2022      231            3    410
H2023      231            3    306
K2021       92            3    192
K2022      231            3    403
K2023      231            3    244
N2021      133            3    280
N2022      231            3    419
N2023      231            3    260
N2024      231            3    178
U2021      176            3    343
U2022      231            3    362
U2023      231            3    202
Z2021      231            3    402
Z2022      231            3    374
Z2023      231            3    299
```

```
Z2024      231            3     184

Calculate average price per maturity:
In 2021, the minimum maturity-average of Corn price was 432 +/- 22 USD for Z2024.
In 2021, the maximum maturity-average of Corn price was 594 +/- 72 USD for N2021.

Calculate price spread per maturity:
In 2021, the minimum maturity-spread of Corn price was 81 USD for H2021.
In 2021, the maximum maturity-spread of Corn price was 288 USD for K2021.
```

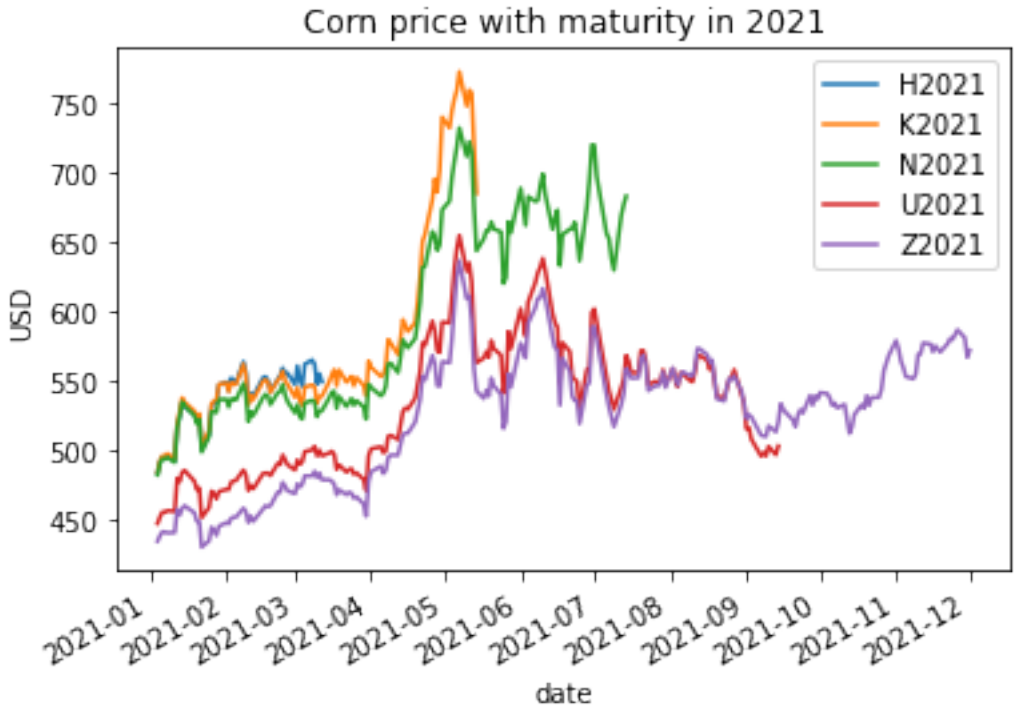
Out[18]:

	Maturity	Observations [%]	Spread	Slope (/day)	Mean	Std
0	H2021	21.0	81.25	0.20	537.0	22.0
1	K2021	40.0	288.50	0.61	578.0	74.0
2	N2021	58.0	250.00	0.61	594.0	72.0
3	U2021	76.0	207.50	0.17	533.0	49.0
4	Z2021	100.0	206.25	0.42	526.0	47.0
5	H2022	100.0	202.50	0.40	533.0	47.0
6	K2022	100.0	198.50	0.40	537.0	47.0
7	N2022	100.0	195.50	0.40	537.0	46.0
8	U2022	100.0	161.50	0.44	497.0	44.0
9	Z2022	100.0	161.00	0.44	485.0	44.0
10	H2023	100.0	161.00	0.44	492.0	44.0
11	K2023	100.0	161.75	0.45	495.0	44.0
12	N2023	100.0	155.00	0.43	497.0	42.0
13	U2023	100.0	116.00	0.32	465.0	32.0
14	Z2023	100.0	109.00	0.31	454.0	31.0
15	N2024	100.0	116.00	0.33	463.0	32.0
16	Z2024	100.0	82.25	0.23	432.0	22.0

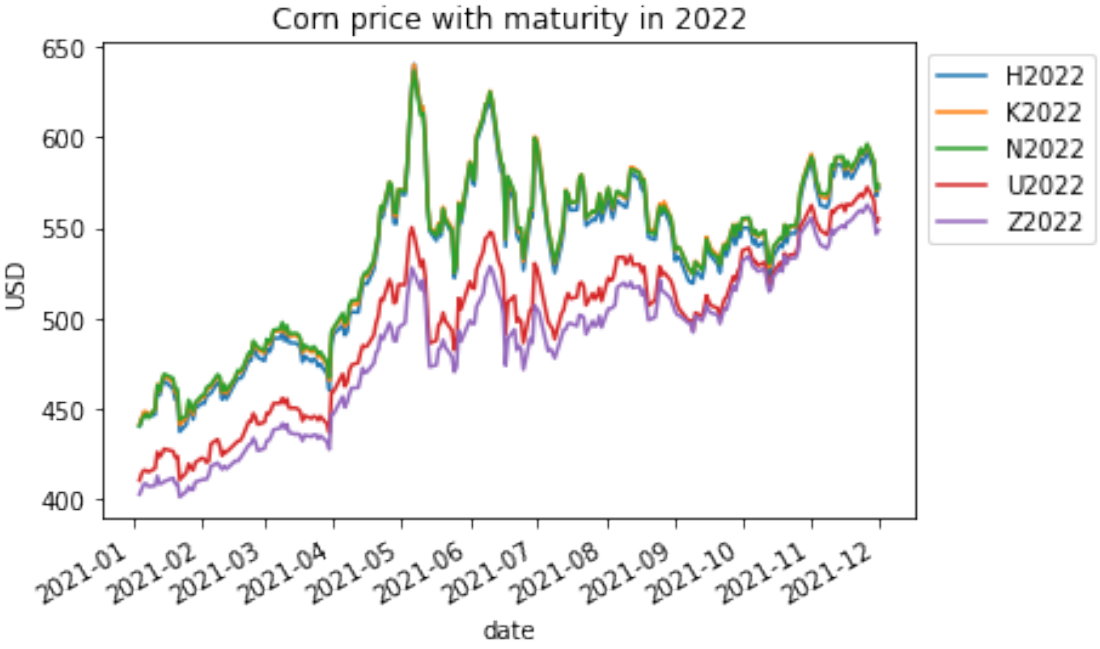
```
<Figure size 2304x1296 with 0 Axes>
```



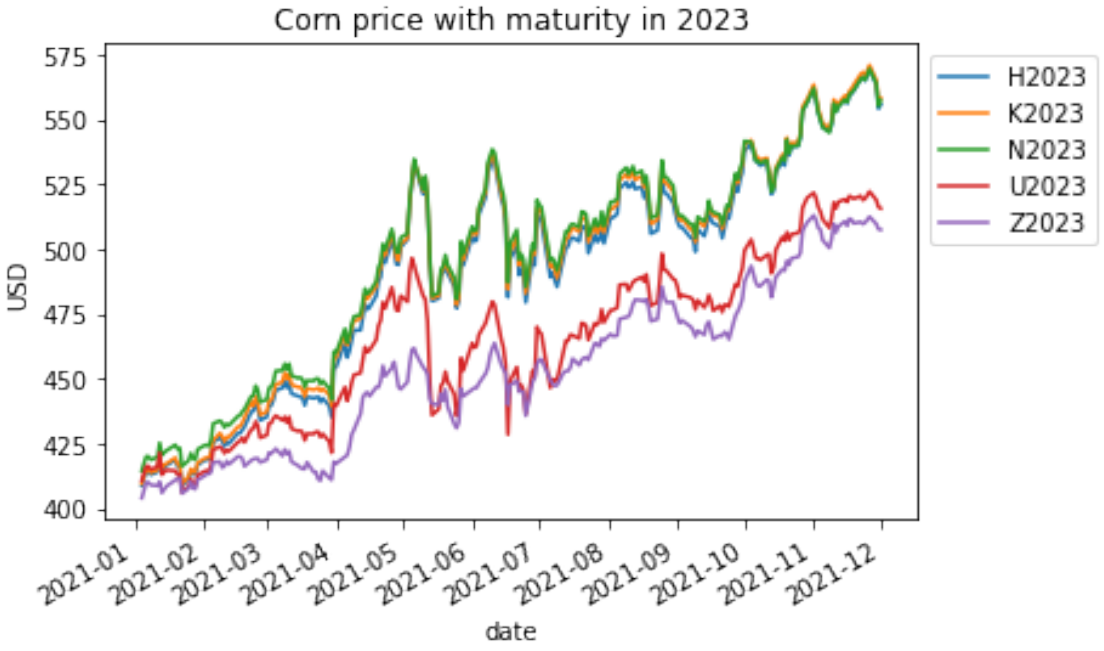
<Figure size 2304x1296 with 0 Axes>



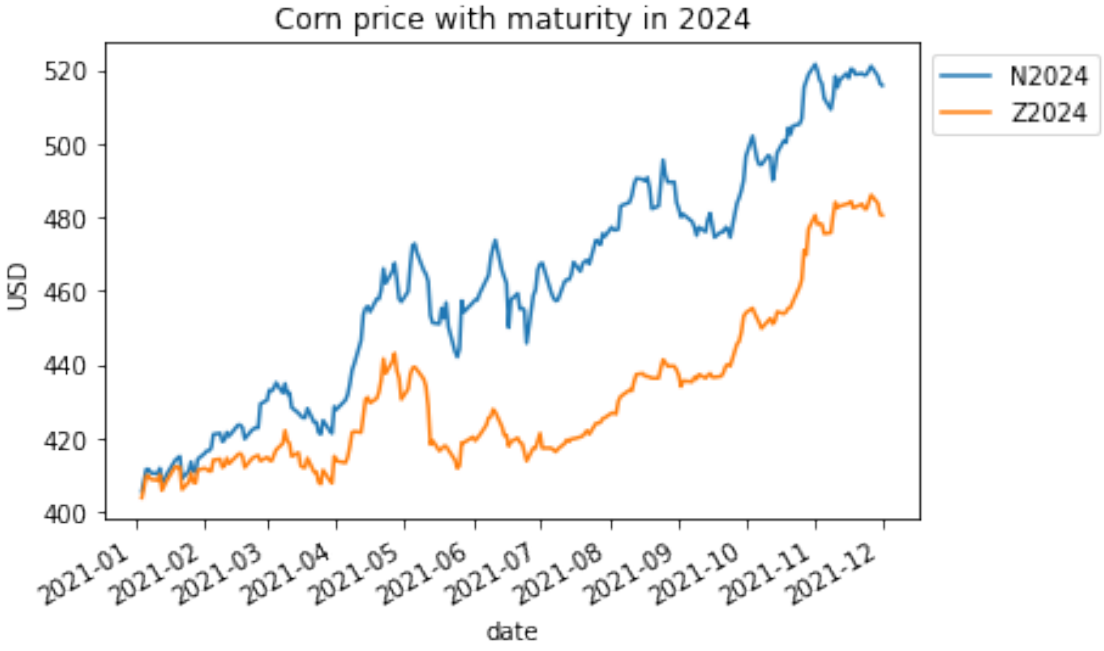
<Figure size 2304x1296 with 0 Axes>



<Figure size 2304x1296 with 0 Axes>



<Figure size 2304x1296 with 0 Axes>



```
In [19]: metrics = inspectDataset(dataSoy, soy[1])
         metrics.round({"Observations [%]": 0, "Spread": 2, "Slope (/day)": 2, "Mean": 0, "Std": 0})
```

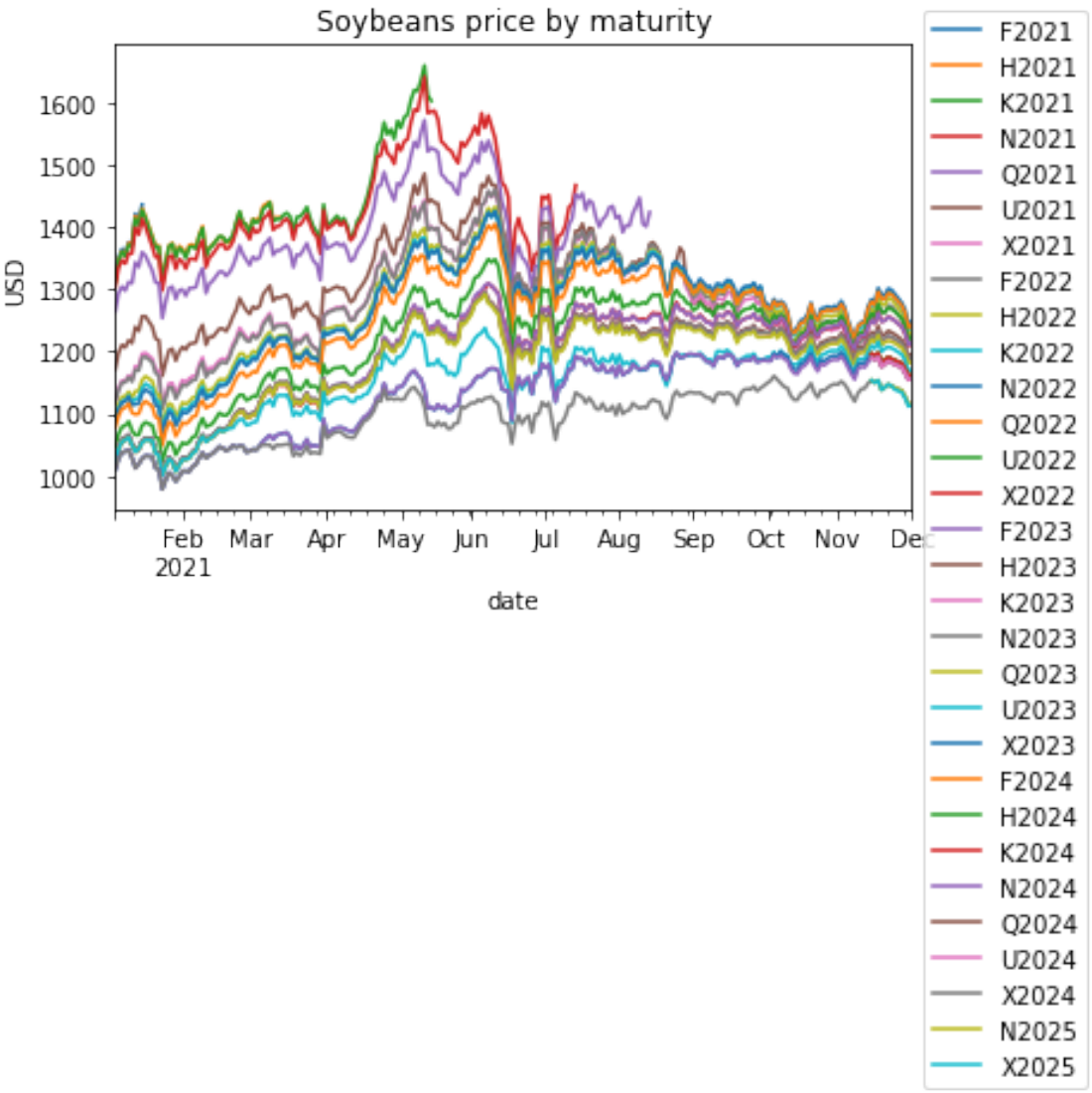
```
Inspecting Soybeans dataset:
                          date maturity observation    value
0     2021-01-04 00:00:00+00:00    F2021        High  1349.75
1     2021-01-04 00:00:00+00:00    F2021         Low  1306.25
2     2021-01-04 00:00:00+00:00    F2021      Settle  1316.50
3     2021-01-04 00:00:00+00:00    H2021        High  1349.50
4     2021-01-04 00:00:00+00:00    H2021         Low  1301.00
...                         ...      ...         ...      ...
10210 2021-12-01 00:00:00+00:00    X2024        High  1116.50
10211 2021-12-01 00:00:00+00:00    X2024         Low  1116.00
10212 2021-12-01 00:00:00+00:00    X2024      Settle  1116.50
10213 2021-12-01 00:00:00+00:00    N2025      Settle  1116.50
10214 2021-12-01 00:00:00+00:00    X2025      Settle  1113.50

[10215 rows x 4 columns]

Group by maturity:
          date  observation  value
maturity
F2021        9            3     26
F2022      231            3    485
F2023      231            3    275
F2024       12            1     12
H2021       48            3    125
H2022      231            3    488
H2023      231            3    246
H2024       12            1     12
K2021       92            3    212
K2022      231            3    454
K2023      231            3    226
K2024       12            1     12
N2021      133            3    311
N2022      231            3    448
N2023      231            3    231
N2024      231            3    189
N2025       12            1     11
Q2021      155            3    361
Q2022      231            3    295
Q2023      231            3    197
Q2024       12            1     12
U2021      176            3    385
U2022      231            3    287
U2023      231            3    190
U2024       12            1     12
X2021      219            3    501
X2022      231            3    440
X2023      231            3    306
X2024      231            3    204
X2025       12            1     11
```

```
Calculate average price per maturity:
In 2021, the minimum maturity-average of Soybeans price was 1095 +/- 44 USD for X2024.
In 2021, the maximum maturity-average of Soybeans price was 1432 +/- 80 USD for N2021.

Calculate price spread per maturity:
In 2021, the minimum maturity-spread of Soybeans price was -42 USD for X2025.
In 2021, the maximum maturity-spread of Soybeans price was 358 USD for N2022.
```

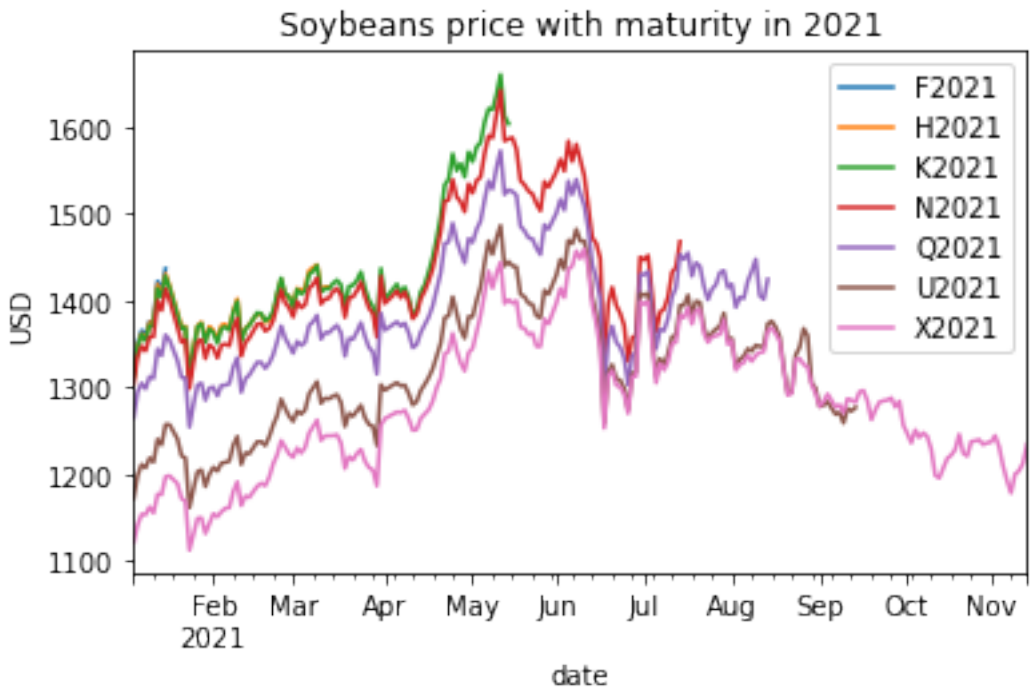
Out[19]:

	Maturity	Observations [%]	Spread	Slope (/day)	Mean	Std
0	F2021	4.0	120.00	0.36	1379.0	38.0
1	H2021	21.0	129.50	0.31	1385.0	30.0
2	K2021	40.0	349.25	0.88	1431.0	81.0
3	N2021	58.0	344.00	0.50	1432.0	80.0
4	Q2021	67.0	318.50	0.48	1394.0	72.0
5	U2021	76.0	325.25	0.32	1320.0	76.0
6	X2021	95.0	347.50	0.34	1281.0	80.0
7	F2022	100.0	354.25	0.33	1282.0	80.0
8	H2022	100.0	351.50	0.42	1269.0	80.0
9	K2022	100.0	353.25	0.46	1268.0	81.0
10	N2022	100.0	357.75	0.49	1268.0	83.0
11	Q2022	100.0	356.25	0.50	1254.0	82.0
12	U2022	100.0	326.50	0.54	1219.0	79.0
13	X2022	100.0	307.00	0.58	1196.0	79.0
14	F2023	100.0	306.00	0.57	1197.0	78.0
15	H2023	100.0	295.00	0.51	1186.0	72.0
16	K2023	100.0	295.00	0.49	1184.0	71.0
17	N2023	100.0	292.00	0.50	1185.0	71.0
18	Q2023	100.0	292.00	0.49	1181.0	70.0
19	U2023	100.0	237.25	0.45	1154.0	57.0
20	X2023	100.0	221.25	0.46	1128.0	63.0
21	F2024	5.0	-36.25	-0.10	1183.0	12.0
22	H2024	5.0	-36.25	-0.10	1183.0	12.0
23	K2024	5.0	-36.25	-0.10	1183.0	12.0
24	N2024	100.0	219.50	0.44	1128.0	63.0
25	Q2024	5.0	-36.25	-0.10	1177.0	12.0
26	U2024	5.0	-36.25	-0.10	1177.0	12.0
27	X2024	100.0	182.25	0.32	1095.0	44.0
28	N2025	5.0	-38.50	-0.11	1139.0	13.0
29	X2025	5.0	-41.50	-0.12	1138.0	14.0

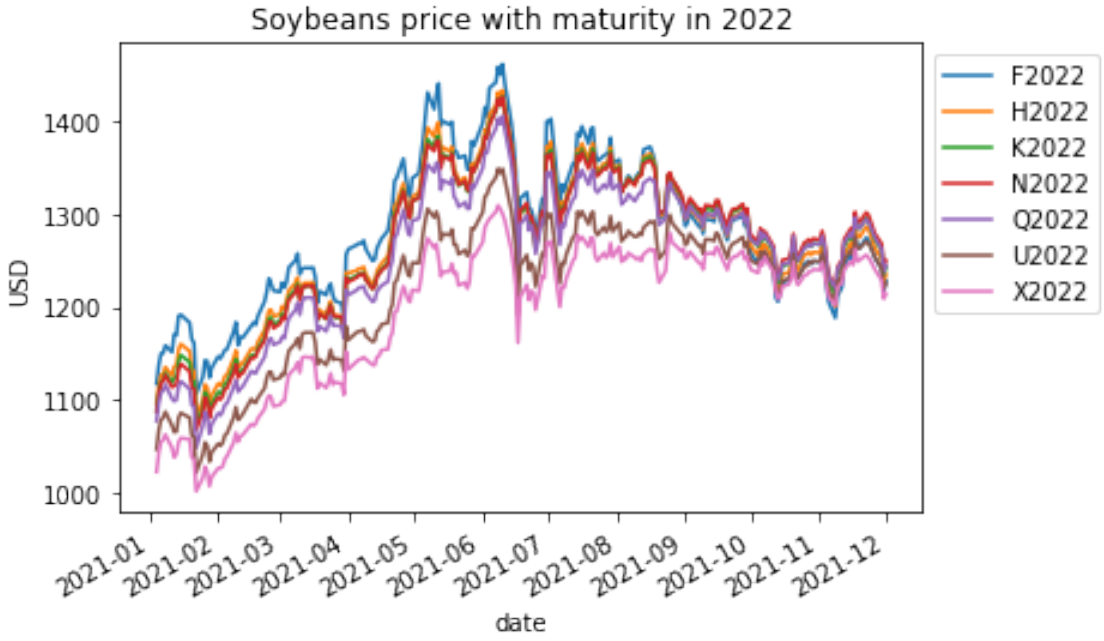
```
<Figure size 2304x1296 with 0 Axes>
```



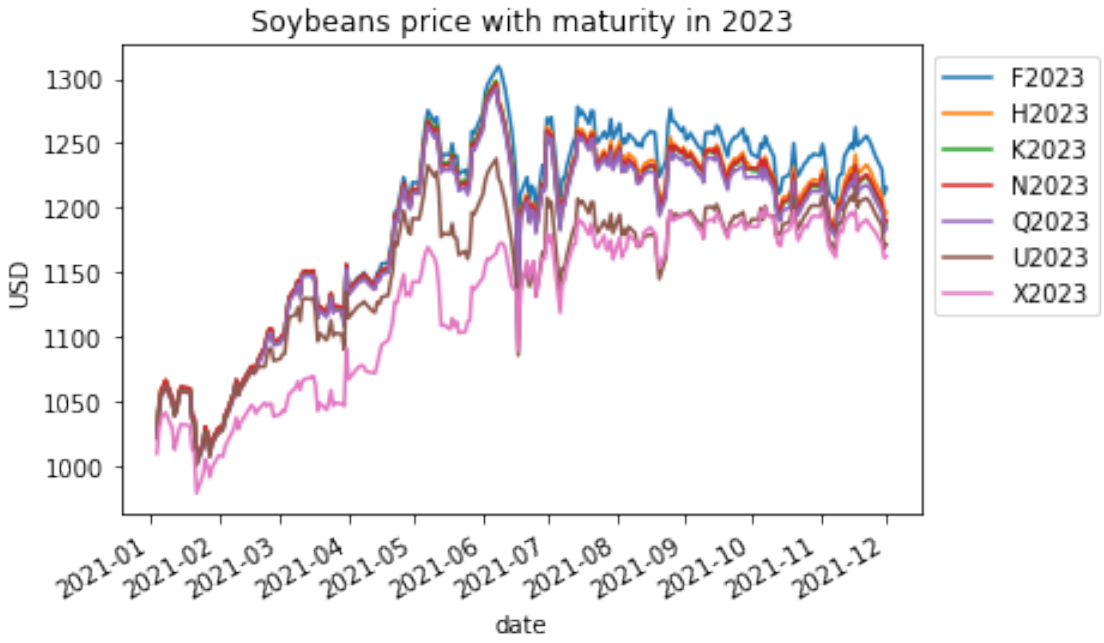
<Figure size 2304x1296 with 0 Axes>



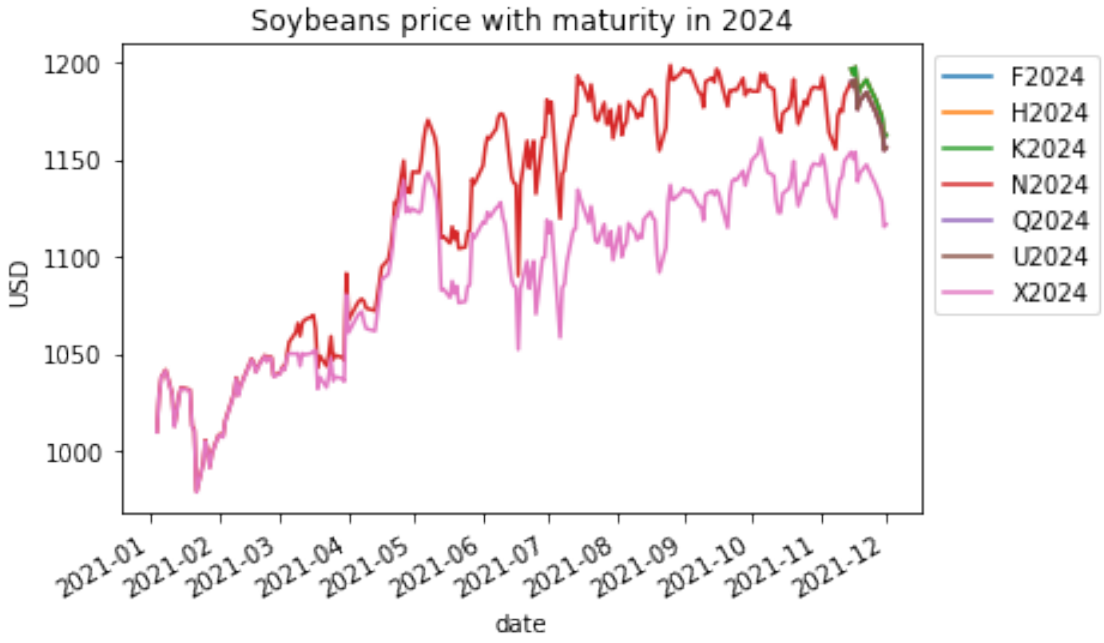
<Figure size 2304x1296 with 0 Axes>



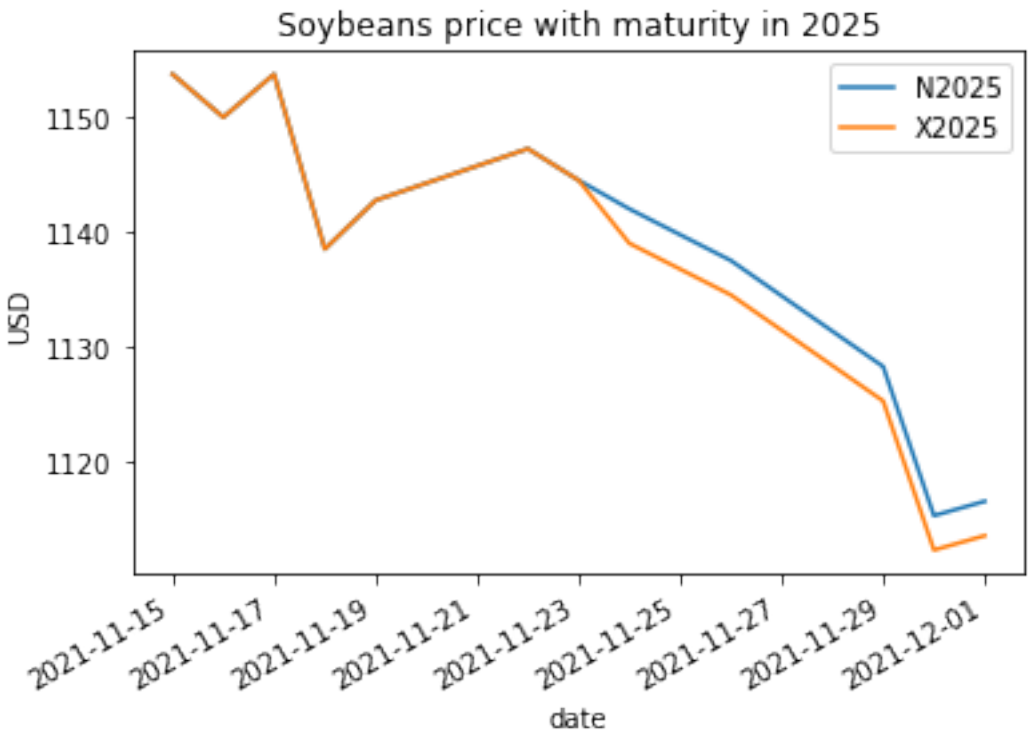
<Figure size 2304x1296 with 0 Axes>



<Figure size 2304x1296 with 0 Axes>



<Figure size 2304x1296 with 0 Axes>


Soybeans price with maturity in 2025
N2025
X2025
1150
1140
1130
1120
USD
2021-11-15
2021-11-17
2021-11-19
2021-11-21
2021-11-23
2021-11-25
2021-11-27
2021-11-29
2021-12-01
date

In [ ]: