

```
1 import numpy as np
2 import pandas as pd
```

```
1 df = pd.read_excel("Prices3.xlsx")
```

```
1 df.head()
```

	Date	Contract Month	Price
0	2021-01-01	2021-08-01	1.01
1	2021-01-01	2021-09-01	1.02
2	2021-01-01	2021-10-01	1.03
3	2021-01-01	2021-11-01	1.04
4	2021-01-01	2022-01-01	1.06

```
1 da = df.Date.unique()
```

```
1 new_dic={}
2 flag=True
3 for i in da:
4     df1 = df[df.Date==i]
5     a=[i.month for i in df1["Contract Month"]]
6     b=[i.year for i in df1["Contract Month"]]
7     c=len(a)
8     hh=[]
9     for w in range(0,c):
10         g=(a[w],b[w])
11         hh.append(g)
12     df1["Month"]=hh
13     df2=pd.DataFrame({"Month":hh})
14     a=pd.merge(df2,df1.loc[:,["Month","Price"]],how="outer")
15     ddf=pd.DataFrame({"Month":a.Month,"Price":a.Price})
16     new_dic[i]=ddf
```

⚠ /usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:12: SettingWithCopyWarning  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/user>  
if sys.path[0] == '':

```
1 final_df=pd.DataFrame({"Month":[]})
2 for a in da:
```

```
3 final_df=pd.merge(final_df,new_dic[a],on="Month",how="outer")

1 da=list(da)
2 da.insert(0,"Month")

1 final_df.columns=da

1 vs=[]
2 for a in final_df["Month"]:
3     vs.append(pd.to_datetime(str(a[1])+"-" + str(a[0])))

1 final_df["Month1"]=vs
```

	Month	2021-01-01T00:00:00.000000000	2021-01-02T00:00:00.000000000	2021-01-03T00:00:00.000000000	2021-01-04T00:00:00.000000000
0	(8, 2021)	1.01	2.01	3.01	
1	(9, 2021)	1.02	2.02	3.02	
2	(10, 2021)	1.03	2.03	3.03	
3	(11, 2021)	1.04	2.04	3.04	
26	(12, 2021)	NaN	2.05	3.05	

```
1 final_df=final_df.T
2 final_df=final_df.sort_values(by="Month1")
```

	0	1	2	3	26	4	5
Month	(8, 2021)	(9, 2021)	(10, 2021)	(11, 2021)	(12, 2021)	(1, 2022)	(2, 2022)
2021-01-01T00:00:00.000000000	1.01	1.02	1.03	1.04	NaN	1.06	1.07
2021-01-02T00:00:00.000000000	2.01	2.02	2.03	2.04	2.05	2.06	2.07
2021-01-03T00:00:00.000000000	3.01	3.02	3.03	3.04	3.05	3.06	3.07
2021-01-04T00:00:00.000000000	4.01	4.02	4.03	4.04	4.05	4.06	4.07
2021-01-05T00:00:00.000000000	5.01	5.02	5.03	5.04	5.05	5.06	5.07
2021-01-06T00:00:00.000000000	NaN	6.01	6.02	6.03	6.04	6.05	6.06
Month1	2021-08-01 00:00:00	2021-09-01 00:00:00	2021-10-01 00:00:00	2021-11-01 00:00:00	2021-12-01 00:00:00	2022-01-01 00:00:00	2022-02-01 00:00:00

```
1 final_df=final_df.T
2 final_df.drop(labels=["Month", "Month1"], axis=0, inplace=True)
3 final_df
```

	0	1	2	3	26	4	5	6	7	8	9	10
2021-01-01T00:00:00.000000000	1.01	1.02	1.03	1.04	NaN	1.06	1.07	1.08	1.09	1.1	1.11	1.12
2021-01-02T00:00:00.000000000	2.01	2.02	2.03	2.04	2.05	2.06	2.07	2.08	2.09	2.1	2.11	2.12
2021-01-03T00:00:00.000000000	3.01	3.02	3.03	3.04	3.05	3.06	3.07	3.08	3.09	3.1	3.11	3.12
2021-01-04T00:00:00.000000000	4.01	4.02	4.03	4.04	4.05	4.06	4.07	4.08	4.09	4.1	4.11	4.12
2021-01-05T00:00:00.000000000	5.01	5.02	5.03	5.04	5.05	5.06	5.07	5.08	5.09	5.1	5.11	5.12

```
1 final_df.fillna(method="ffill",axis=0,inplace=True)
```

```
1 final_df
```

	Month	2021-01-01T00:00:00.000000000	2021-01-02T00:00:00.000000000	2021-01-03T00:00:00.000000000	2021-01-04T00:00:00.000000000
0	(8, 2021)	1.01	2.01	3.01	
1	(9, 2021)	1.02	2.02	3.02	
2	(10, 2021)	1.03	2.03	3.03	
3	(11, 2021)	1.04	2.04	3.04	
26	(12, 2021)	NaN	2.05	3.05	
4	(1, 2022)	1.06	2.06	3.06	
5	(2, 2022)	1.07	2.07	3.07	
	(3, 2022)				

```
1 final_df.to_excel("Final.xlsx")
```

7	(4, 2022)	1.09	2.09	3.09	
8	(5, 2022)	1.10	2.10	3.10	
9	(6, 2022)	1.11	2.11	3.11	
10	(7, 2022)	1.12	2.12	3.12	
11	(8, 2022)	1.13	2.13	3.13	
12	(9, 2022)	1.14	NaN	3.14	
13	(10, 2022)	1.15	2.15	3.15	
14	(11, 2022)	1.16	2.16	3.16	
15	(12, 2022)	1.17	2.17	3.17	

16	(1, 2023)	1.18	2.18	3.18	
----	-----------	------	------	------	--