Problem 1

a) Run regression Model 2. Format the output for legibility. According to Model 2, do free samples and detailing visits have a statistically significant impact on new prescriptions?

Ans: We begin the analysis by examining the 'Original Variables' sheet of data for the new prescriptions, samples, and details. We keep the New Prescriptions as the y-variable, and Samples and Details as the x-variables. We run an 'ordinary least squares' regression (Regression Model 2) on a pandas data frame created for the sheet of variables, and attempt to minimize the square of errors.

In [220]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
import seaborn as sns
```

In [147]:
```python
df= pd.read_excel('DetailingData.xlsx', sheet_name='Original Variables')
df
```

Out[147]:

	Physician	Date	ID_Date	NewPrescription	Samples	Details	CompetitorNew	Psych	Medical CPI	DRG
0	1	1998-12-01	1_36130	10	0	0	11	1	245.2	389.85
1	2	1998-12-01	2_36130	11	0	2	2	0	245.2	389.85
2	3	1998-12-01	3_36130	11	0	1	38	0	245.2	389.85
3	4	1998-12-01	4_36130	10	0	0	15	0	245.2	389.85
4	5	1998-12-01	5_36130	11	0	2	2	0	245.2	389.85
...	...	...	...	...	...	...	...	...	...	...
4387	179	2000-11-01	179_36831	12	0	0	36	0	264.1	441.84
4388	180	2000-11-01	180_36831	13	0	1	43	0	264.1	441.84
4389	181	2000-11-01	181_36831	15	0	4	42	0	264.1	441.84
4390	182	2000-11-01	182_36831	14	2	4	19	0	264.1	441.84
4391	183	2000-11-01	183_36831	13	0	0	7	0	264.1	441.84

4392 rows × 10 columns

In [169]:
```python
regression_model_2 = smf.ols('NewPrescription~Samples+Details',data=df).fit()
regression_model_2.summary()
```

Out[169]:

OLS Regression Results

Dep. Variable:	NewPrescription	**R-squared:**	0.199
Model:	OLS	**Adj. R-squared:**	0.198
Method:	Least Squares	**F-statistic:**	543.8
Date:	Mon, 10 Oct 2022	**Prob (F-statistic):**	1.02e-211
Time:	19:54:24	**Log-Likelihood:**	-7532.7
No. Observations:	4392	**AIC:**	1.507e+04
Df Residuals:	4389	**BIC:**	1.509e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	11.1611	0.029	383.162	0.000	11.104	11.218
Samples	0.0596	0.004	13.363	0.000	0.051	0.068
Details	0.3844	0.016	24.610	0.000	0.354	0.415

Omnibus:	591.727	**Durbin-Watson:**	1.517
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	859.241
Skew:	1.015	**Prob(JB):**	2.62e-187
Kurtosis:	3.759	**Cond. No.**	8.16

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	11.1611	0.029	383.162	0.000	11.104	11.218
Samples	0.0596	0.004	13.363	0.000	0.051	0.068
Details	0.3844	0.016	24.610	0.000	0.354	0.415

On checking the summary we see that the coefficients of both Samples and Details are *positive* which means that they positively affect the number of new prescriptions. Increasing the number of free samples distributed to a physician in a particular month, or increasing the number of detailing visits would have a positive impact on the number of new prescriptions made for Xuris by the physician. We can also see that the 'p' values for both are *zero*. We know that if the 'p' values are less than 0.05, we can reject the null hypothesis, which means that both have a statistically significant impact on the new prescriptions.

b) According to Model 2, what is the expected number of new prescriptions generated by a detailing visit? Report a 95% confidence interval. What is the expected $ margin generated by a detailing visit and what is the 95% confidence interval for the expected margin? Assume each new prescription generates 2.5 additional refills.

Ans: If we focus just on the effect of the number of detailing visits on the number of new prescriptions generated, we extract the sum of the 'Details' parameter from the results of our regression analysis using model 2.

If we assume 2.5 additional refills per new prescription, the total number of prescriptions=1+2.5=3.5.

Each prescription is valued at 115$. So each new prescription leads to revenue of 115*3.5=402.5$.

It is given that the confidence interval is 95%. We know that, for any random variable with a standard Normal distribution, Z=N(0,1), we know that P(−1.96<Z<1.96)=0.95. Hence, we can calculate the expected margin.

By multiplying +1.96 and -1.96 with the std error value for Details, we can get the expected number of prescriptions with a 95% confidence interval.

```
In [187]: total=regression_model_2.params['Details'].sum()
          total

Out[187]: 0.38443112827453263

In [185]: upperlimit=regression_model_2.params['Details'].sum()+(1.96*0.016)
          upperlimit

Out[185]: 0.41579112827453263

In [186]: lowerlimit=regression_model_2.params['Details'].sum()-(1.96*0.016)
          lowerlimit

Out[186]: 0.35307112827453263

In [172]: upperlimit*115*3.5  # 1 sale+2.5 additional refills=3.5 * 115(sale value for each)

Out[172]: 167.3559291304994

In [173]: lowerlimit*115*3.5

Out[173]: 142.1111291304994
```

We get this range as: **0.35 to 0.41.**
Similarly, based on the sales, the expected margin generated by each detailing visit would be 0.35*402.5$ to 0.41*402.5$. This gives us the expected range for sales: **142.11$ to 167.35$.**

c) Answer (b) for Model 7 and compare the answers to those you found for Model 2. Why are the results different? Which model do you consider more informative? The case states that physicians draw on their own knowledge and experience in deciding which drugs to prescribe. How does this bear on the comparison between Models 2 and 7?
Ans: As we see in Table 2, we now begin the analysis by examining the 'Differenced Variables' sheet of data for the new prescriptions, samples, and details(difference between Jan 1999 and Dec 1998). We keep the NewPrescDiff as the y-variable, and SamplesDiff and DetailsDiff as the x-variables.

```
In [157]: diff_df=pd.read_excel('DetailingData.xlsx', sheet_name='Differenced Variables')
          diff_df
```

Out[157]:

	Physician	Date	ID_Date	NewPrescDiff	SamplesDiff	DetailsDiff	CompetitorNewDiff	Psych	MedicalCPIDiff	DRGDiff
0	1	1999-01-01	1_36161	0	0	0	-2	1	1.4	1.01
1	2	1999-01-01	2_36161	0	0	1	-1	0	1.4	1.01
2	3	1999-01-01	3_36161	0	0	2	-16	0	1.4	1.01
3	4	1999-01-01	4_36161	0	0	0	5	0	1.4	1.01
4	5	1999-01-01	5_36161	0	0	0	0	0	1.4	1.01
...	...	...	...	...	...	...	...	...	...	...
4204	179	2000-11-01	179_36831	0	0	0	9	0	0.4	20.61
4205	180	2000-11-01	180_36831	0	0	-2	6	0	0.4	20.61
4206	181	2000-11-01	181_36831	-1	-26	-2	11	0	0.4	20.61
4207	182	2000-11-01	182_36831	1	2	2	-6	0	0.4	20.61
4208	183	2000-11-01	183_36831	0	0	-1	-14	0	0.4	20.61

4209 rows × 10 columns

```
In [158]: regression_model_7=smf.ols('NewPrescDiff ~ SamplesDiff + DetailsDiff', data = diff_df).fit()
          regression_model_7.summary()
```

Out[158]:

OLS Regression Results

Dep. Variable:	NewPrescDiff	**R-squared:**	0.613
Model:	OLS	**Adj. R-squared:**	0.613
Method:	Least Squares	**F-statistic:**	3335.
Date:	Mon, 10 Oct 2022	**Prob (F-statistic):**	0.00
Time:	19:54:00	**Log-Likelihood:**	202.92
No. Observations:	4209	**AIC:**	-399.8
Df Residuals:	4206	**BIC:**	-380.8
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0799	0.004	22.473	0.000	0.073	0.087
SamplesDiff	0.0354	0.001	58.217	0.000	0.034	0.037
DetailsDiff	0.1115	0.003	41.334	0.000	0.106	0.117

Omnibus:	398.475	**Durbin-Watson:**	2.004
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	517.794
Skew:	0.825	**Prob(JB):**	3.65e-113
Kurtosis:	3.477	**Cond. No.**	6.05

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0799	0.004	22.473	0.000	0.073	0.087
SamplesDiff	0.0354	0.001	58.217	0.000	0.034	0.037
DetailsDiff	0.1115	0.003	41.334	0.000	0.106	0.117

On performing calculations similar to the previous questions, we see that the standard error for DetailsDiff is now 0.03. By multiplying +1.96 and -1.96 with the std error value for Details, we can get the expected number of prescriptions with a 95% confidence interval.

```
In [159]: new_total=regression_model_7.params['DetailsDiff'].sum()

In [188]: new_upperlimit=regression_model_7.params['DetailsDiff'].sum()+(1.96*0.003)
          new_upperlimit

Out[188]: 0.11735133274439843

In [189]: new_lowerlimit=regression_model_7.params['DetailsDiff'].sum()-(1.96*0.003)
          new_lowerlimit

Out[189]: 0.10559133274439844

In [179]: new_upperlimit*115*3.5

Out[179]: 47.23391142962037

In [180]: new_lowerlimit*115*3.5

Out[180]: 42.500511429620374
```

We get this range as: **0.10 to 0.11.** Then, based on the sales, the expected margin generated by each detailing visit would be 0.10*402.5$ to 0.11*402.5$. This gives us the expected range for

sales: **42.5$ to 47.2$.** The difference between the two models occurs due to the differenced variables in model 7, as compared to the original variables in model 2.
If we consider the fact that physicians draw on their own knowledge and experience in deciding which drugs to prescribe and not just based on the free samples or number of detailing visits, which is not being taken into account in model 7 as we use only differenced variables. Also, the Standard Error for Details is less when we consider DetailsDiff as compared to when we use Details, which makes model 7 more accurate.

d) Overall, of Models 1-9, which one do you consider most reliable? Why? What does your preferred model say about the cost-effectiveness of detailing visits?
Ans: Models 1 to 5 use undifferenced variables so they might include effects constant in time. We can't say for sure whether the increase in new prescriptions was only because of the increased number of detailing visits or some other reasons(behavioral, etc). Alternatively, in models 6-9 the differenced variables give a more clear idea about how only the detailing visits or samples affect the new prescriptions, by removing the behavioral bias altogether. Moreover, from Table 2, if we compare the DetailsDiff, R-sq, and Standard Error values for Models 7,8,9, are the same.

TABLE 2.

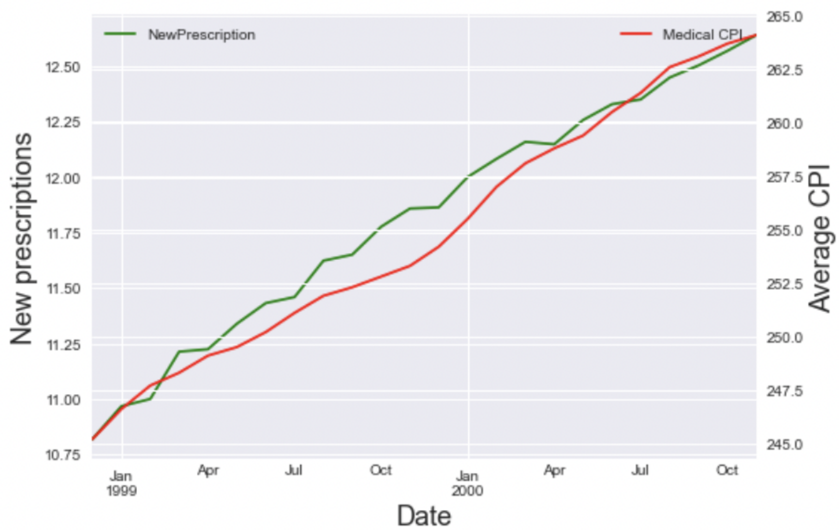
	Model 6	Model 7	Model 8	Model 9
Intercept	0.080 (0.004)	0.080 (0.004)	0.080 (0.004)	0.081 (0.010)
SamplesDiff	0.041 (0.001)	0.035 (0.001)	0.035 (0.001)	0.035 (0.001)
DetailsDiff		0.111 (0.003)	0.111 (0.003)	0.111 (0.003)
CompetitorNewDiff			0.000 (0.000)	0.000 (0.000)
MedicalCPIDiff				-0.002 (0.011)
DRG Diff				0.000 (0.000)
F test (p-value)	0.000	0.000	0.000	0.000
R-sq	45.6%	61.3%	61.3%	61.3%
se	0.273	0.231	0.231	0.231

The highest R-sq value and lowest Std Error values make these models the most preferred ones. And since the P-Values are zero and Coefficients are positive, we can say that the effect of increased detailing visits is statistically significant and has a positive impact on the number of new prescriptions.

e) Does MedicalCPI appear to have a statistically significant effect? Why might this be? Look at the data before trying to answer. Compare Model 9 with Models 3-5.
Ans: If we only look at Models 3-5, we can see that Medical CPI does have an effect on the number of new prescriptions, and since the coefficient is positive, we can assume that as the level of the consumer price index for medical expenses in that month increases, the number of new prescriptions increases.

We can also plot a twin-axis graph using matplotlib to see how the new prescriptions and medical CPI vary with time.

```
In [289]: df.Date = pd.to_datetime(df.Date)

          fig, ax1 = plt.subplots()

          ax2 = ax1.twinx()

          df.groupby('Date').NewPrescription.mean().plot(ax=ax1,color='green')
          df.groupby('Date')[['Medical CPI']].mean().plot(ax=ax2, color='red')

          ax1.set_xlabel('Date', fontsize=18)
          ax1.set_ylabel('New prescriptions', fontsize=18)
          ax2.set_ylabel('Average CPI', fontsize=18)

          ax1.legend(loc=0)
          ax2.legend(loc=1)

Out[289]: <matplotlib.legend.Legend at 0x7f9b43ea5490>
```



But as we run Regression Model 9, we consider the data using differenced variables.

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0825	0.010	8.266	0.000	0.063	0.102
SamplesDiff	0.0354	0.001	58.162	0.000	0.034	0.037
DetailsDiff	0.1114	0.003	41.214	0.000	0.106	0.117
CompetitorNewDiff	0.0003	0.000	0.650	0.516	-0.001	0.001
Psych	-0.0699	0.028	-2.496	0.013	-0.125	-0.015
MedicalCPIDiff	-0.0020	0.011	-0.177	0.860	-0.024	0.020
DRGDiff	5.915e-05	0.000	0.384	0.701	-0.000	0.000

We see that the effect of Medical CPI Diff p-value=0.860>0.05). Hence we cannot reject the null hypothesis, and the effect of Medical CPI is not statistically significant. We can plot the graph again and see that there is not a direct correlation between the two variables.

f) Interpret the coefficients for CompetitorNew and Psych in Model 5.

Ans:

			Model 5
CompetitorNew			0.004 (0.001)
Psych		-0.862 (0.143)	-0.834 (0.144)

We can see that in Model 5, for Competitor New, the value is positive and statistically significant, which means that as the number of prescriptions for Xuris's competitor product prescribed by a particular physician in a month increases, the number of new prescriptions for Xuris also increases. Alternatively, the value of the coefficient for Pysch is negative and statistically significant, which means that psychiatrists are less likely to prescribe Xuris.

Problem 2

a) Run a regression of number of eggs on feed and interpret the result. Does it align with your intuition?

Ans:

OLS Regression Results

Dep. Variable:	eggs	**R-squared:**	0.176
Model:	OLS	**Adj. R-squared:**	0.175
Method:	Least Squares	**F-statistic:**	331.1
Date:	Sun, 20 Feb 2022	**Prob (F-statistic):**	3.38e-67
Time:	21:22:06	**Log-Likelihood:**	-1190.7
No. Observations:	1552	**AIC:**	2385.
Df Residuals:	1550	**BIC:**	2396.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	3.8328	0.114	33.635	0.000	3.609	4.056
feed	-0.0891	0.005	-18.195	0.000	-0.099	-0.080

Omnibus:	0.504	**Durbin-Watson:**	2.111
Prob(Omnibus):	0.777	**Jarque-Bera (JB):**	0.484
Skew:	0.043	**Prob(JB):**	0.785
Kurtosis:	3.005	**Cond. No.**	201.

We need to analyze the effect of the amount fed to each chicken a day before they laid the eggs on the number of eggs laid. We run an 'ordinary least squares' regression on a pandas data frame created for the egg_production.csv file, and attempt to minimize the square of errors.
From the results, we can interpret that since the coefficient is negative and the p-value is zero, 'feed' has a negative yet statistically significant impact on the number of eggs laid. But this would be against our intuition that if we feed the chickens more, they would lay more eggs. We can see that the R-squared value is 0.176 which means that the model is not very accurate in its prediction.

b) Now run a regression using both variables. Interpret the result. Does this make sense to you?
Ans:

OLS Regression Results

Dep. Variable:	eggs	**R-squared:**	0.176
Model:	OLS	**Adj. R-squared:**	0.175
Method:	Least Squares	**F-statistic:**	165.6
Date:	Tue, 11 Oct 2022	**Prob (F-statistic):**	6.63e-66
Time:	19:27:25	**Log-Likelihood:**	-1190.5
No. Observations:	1552	**AIC:**	2387.
Df Residuals:	1549	**BIC:**	2403.
Df Model:	2		
Covariance Type:	nonrobust		

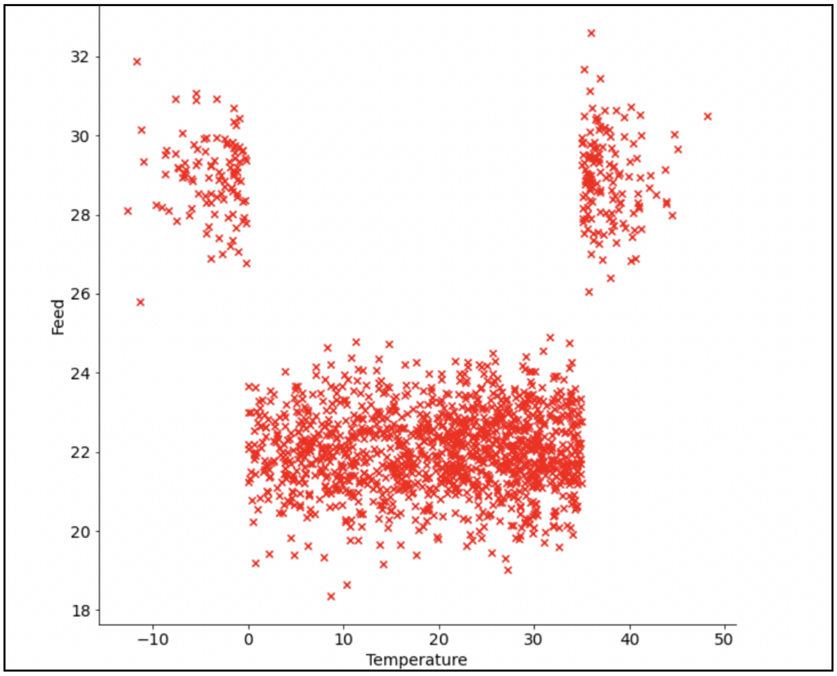
	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	3.8449	0.116	33.137	0.000	3.617	4.072
feed	-0.0891	0.005	-18.190	0.000	-0.099	-0.079
temperature	-0.0006	0.001	-0.557	0.577	-0.003	0.002

Omnibus:	0.513	**Durbin-Watson:**	2.111
Prob(Omnibus):	0.774	**Jarque-Bera (JB):**	0.489
Skew:	0.043	**Prob(JB):**	0.783
Kurtosis:	3.007	**Cond. No.**	278.

We then include temperature as one of the x-variables and try to analyze the effects of both feed (amount fed to each chicken before the day they lay eggs) and the daily temperature that day, on the number of eggs laid. As the coefficient is negative, we can infer a negative relationship between the temperature on the number of eggs laid. But the p-value is 0.577>0.05. So we cannot reject the null hypothesis, and thus conclude that the effect of temperature is not statistically significant. But this does not make sense, as a warmer environment would be more conducive to the laying of eggs.

c) You suspect that something fishy is going on, and that the amount of feed given to each chicken depends on the temperature. Investigate this hypothesis and create a new binary/discrete/categorical variable that captures this phenomenon.

Ans) Considering the hypothesis that the amount of feed given to each chicken depends on the temperature, we try to plot the data in the form of a scatter plot.



From the scatter plot we can infer that as the temperature drops below 0° or goes above 35°, the feed increases. So we can use this interpretation to create a new binary variable 'X' which takes the value 1, every time the temperature is below 0° or goes above 35°, otherwise, it takes the value 0.

In [258]:
```
egg_prod_df['X']=""
egg_prod_df['X'] = np.where(np.logical_or(egg_prod_df['temperature']<0, egg_prod_df['temperature']>35),1,0)
egg_prod_df
```

Out[258]:

	eggs	feed	temperature	X
0	1.944645	28.521682	-3.920247	1
1	2.367084	20.810192	7.489837	0
2	1.361380	29.259575	-5.425451	1
3	1.763221	22.245235	1.486627	0
4	2.003410	23.331641	9.976938	0
...	...	...	...	...
1547	1.641620	22.939631	11.102256	0
1548	2.660458	22.726205	18.844973	0
1549	1.367134	21.987339	3.645734	0
1550	1.724994	22.862650	12.987750	0
1551	2.305316	22.943871	20.767244	0

1552 rows × 4 columns

```
In [274]: group=egg_prod_df.groupby('X').count()
          group
```

Out[274]:

X	eggs	feed	temperature
0	1310	1310	1310
1	242	242	242

d) Regress number of eggs on feed, temperature, and the new variable you created. Interpret the results.

Ans)

OLS Regression Results

Dep. Variable:	eggs	**R-squared:**	0.236
Model:	OLS	**Adj. R-squared:**	0.234
Method:	Least Squares	**F-statistic:**	159.0
Date:	Tue, 11 Oct 2022	**Prob (F-statistic):**	7.96e-90
Time:	20:18:55	**Log-Likelihood:**	-1132.5
No. Observations:	1552	**AIC:**	2273.
Df Residuals:	1548	**BIC:**	2294.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.0529	0.278	3.786	0.000	0.507	1.598
feed	0.0388	0.013	3.081	0.002	0.014	0.063
temperature	-0.0007	0.001	-0.695	0.487	-0.003	0.001
X	-1.0276	0.094	-10.966	0.000	-1.211	-0.844

Omnibus:	0.478	**Durbin-Watson:**	2.108
Prob(Omnibus):	0.787	**Jarque-Bera (JB):**	0.390
Skew:	0.023	**Prob(JB):**	0.823
Kurtosis:	3.062	**Cond. No.**	724.

We then run regression on the data frame with all the variable columns including the newly created 'X' variable. As the R-squared value is now more(0.236>0.176), we can assume that this regression model is more predictive than when we only considered feed and temperature.

Now we also see that the coefficient for feed is now positive and the p-value is 0.002(which is <0.05) which means that, feed has a statistically significant and positive effect on the number of eggs laid. This result is now in agreement with our original hypothesis that more feed should result in more eggs being laid. But, the p-value for temperature is still high(0.487>0.05), therefore, we can ignore it altogether as it has no significant effect.

e) For this model, what is a 90% confidence interval for the prediction of the number of eggs that were produced if the feed was 25 and the temperature was −1. Interpret the results.

Ans)

```
In [285]: prediction_data = pd.DataFrame({'feed':[25], 'temperature':[-1], 'X':[1]})

In [286]: regression_egg_4.predict(prediction_data)

Out[286]: 0    0.994741
          dtype: float64

In [287]: regression_egg_4.get_prediction(prediction_data).summary_frame(alpha=0.1)

Out[287]:
```

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
0	0.994741	0.062953	0.891131	1.098351	0.16108	1.828403

If we assume a 90% confidence interval, we can assume alpha to be (1-0.90)=0.1.
We put the values in the built-in function get_prediction to make a prediction based on the data given in the dataframe and for values of feed=25 and temperature as -1. We can interpret that the mean number of eggs laid based on assumed conditions would be 0.99 with a standard error is 0.06.

Appendix:

Python Notebook:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
import seaborn as sns

df= pd.read_excel('DetailingData.xlsx', sheet_name='Original
Variables')
df
```

```
      Physician       Date    ID_Date  NewPrescription  Samples
Details  \
0             1 1998-12-01    1_36130               10        0
0
1             2 1998-12-01    2_36130               11        0
2
2             3 1998-12-01    3_36130               11        0
1
3             4 1998-12-01    4_36130               10        0
0
4             5 1998-12-01    5_36130               11        0
2
...         ...        ...        ...              ...      ...      .
..
4387        179 2000-11-01  179_36831               12        0
0
4388        180 2000-11-01  180_36831               13        0
1
4389        181 2000-11-01  181_36831               15        0
4
4390        182 2000-11-01  182_36831               14        2
4
4391        183 2000-11-01  183_36831               13        0
0

      CompetitorNew  Psych  Medical CPI     DRG
0                11      1       245.2  389.85
1                 2      0       245.2  389.85
2                38      0       245.2  389.85
3                15      0       245.2  389.85
4                 2      0       245.2  389.85
...             ...    ...         ...     ...
4387             36      0       264.1  441.84
4388             43      0       264.1  441.84
4389             42      0       264.1  441.84
4390             19      0       264.1  441.84
4391              7      0       264.1  441.84

[4392 rows x 10 columns]
```

```
regression_model_2 =
smf.ols('NewPrescription~Samples+Details',data=df).fit()
regression_model_2.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

==============================================================================
========
Dep. Variable:        NewPrescription   R-squared:
0.199
Model:                            OLS   Adj. R-squared:
0.198
Method:                 Least Squares   F-statistic:
543.8
Date:                Mon, 10 Oct 2022   Prob (F-statistic):
1.02e-211
Time:                        19:54:24   Log-Likelihood:
-7532.7
No. Observations:                4392   AIC:
1.507e+04
Df Residuals:                    4389   BIC:
1.509e+04
Df Model:                           2

Covariance Type:            nonrobust

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
Intercept     11.1611      0.029    383.162      0.000      11.104
11.218
Samples        0.0596      0.004     13.363      0.000       0.051
0.068
Details        0.3844      0.016     24.610      0.000       0.354
0.415
==============================================================================
========
Omnibus:                      591.727   Durbin-Watson:
1.517
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
859.241
Skew:                           1.015   Prob(JB):
2.62e-187
Kurtosis:                       3.759   Cond. No.
8.16
```

```
==============================================================================
========

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

#both Samples and Details have p=0; therefore statistically signifcant

total=regression_model_2.params['Details'].sum()
total

0.38443112827453263

upperlimit=regression_model_2.params['Details'].sum()+(1.96*0.016)
upperlimit

0.41579112827453263

lowerlimit=regression_model_2.params['Details'].sum()-(1.96*0.016)
lowerlimit

0.35307112827453263

upperlimit*115*3.5  # 1 sale+2.5 additional refills=3.5 * 115(sale
value for each)

167.3559291304994

lowerlimit*115*3.5

142.1111291304994

total*115*3.5

154.7335291304994

#so margin range= 142.74$-167.30$

diff_df=pd.read_excel('DetailingData.xlsx', sheet_name='Differenced
Variables')
diff_df

      Physician       Date    ID_Date  NewPrescDiff  SamplesDiff
DetailsDiff  \
0              1 1999-01-01    1_36161             0            0
0
1              2 1999-01-01    2_36161             0            0
1
2              3 1999-01-01    3_36161             0            0
2
3              4 1999-01-01    4_36161             0            0
0
```

```
4             5 1999-01-01    5_36161             0            0
0
...         ...        ...        ...           ...          ...
...
4204        179 2000-11-01  179_36831             0            0
0
4205        180 2000-11-01  180_36831             0            0
-2
4206        181 2000-11-01  181_36831            -1          -26
-2
4207        182 2000-11-01  182_36831             1            2
2
4208        183 2000-11-01  183_36831             0            0
-1

      CompetitorNewDiff  Psych  MedicalCPIDiff  DRGDiff
0                    -2      1             1.4     1.01
1                    -1      0             1.4     1.01
2                   -16      0             1.4     1.01
3                     5      0             1.4     1.01
4                     0      0             1.4     1.01
...                 ...    ...             ...      ...
4204                  9      0             0.4    20.61
4205                  6      0             0.4    20.61
4206                 11      0             0.4    20.61
4207                 -6      0             0.4    20.61
4208                -14      0             0.4    20.61

[4209 rows x 10 columns]
```

```
regression_model_7=smf.ols('NewPrescDiff ~ SamplesDiff + DetailsDiff',
data = diff_df).fit()
regression_model_7.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

======================================================================
========
Dep. Variable:           NewPrescDiff   R-squared:
0.613
Model:                            OLS   Adj. R-squared:
0.613
Method:                 Least Squares   F-statistic:
3335.
Date:                Mon, 10 Oct 2022   Prob (F-statistic):
0.00
Time:                        19:54:00   Log-Likelihood:
202.92
```

```
No. Observations:                  4209   AIC:
-399.8
Df Residuals:                      4206   BIC:
-380.8
Df Model:                             2

Covariance Type:            nonrobust

======================================================================
=========
                  coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------
---------
Intercept       0.0799      0.004     22.473      0.000       0.073
0.087
SamplesDiff     0.0354      0.001     58.217      0.000       0.034
0.037
DetailsDiff     0.1115      0.003     41.334      0.000       0.106
0.117
======================================================================
========
Omnibus:                    398.475   Durbin-Watson:
2.004
Prob(Omnibus):                0.000   Jarque-Bera (JB):
517.794
Skew:                         0.825   Prob(JB):
3.65e-113
Kurtosis:                     3.477   Cond. No.
6.05
======================================================================
========

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

```
new_total=regression_model_7.params['DetailsDiff'].sum()

new_upperlimit=regression_model_7.params['DetailsDiff'].sum()
+(1.96*0.003)
new_upperlimit
```

```
0.11735133274439843
```

```
new_lowerlimit=regression_model_7.params['DetailsDiff'].sum()-
(1.96*0.003)
new_lowerlimit
```

```
0.10559133274439844
```

```
new_upperlimit*115*3.5
```

```
47.23391142962037
```

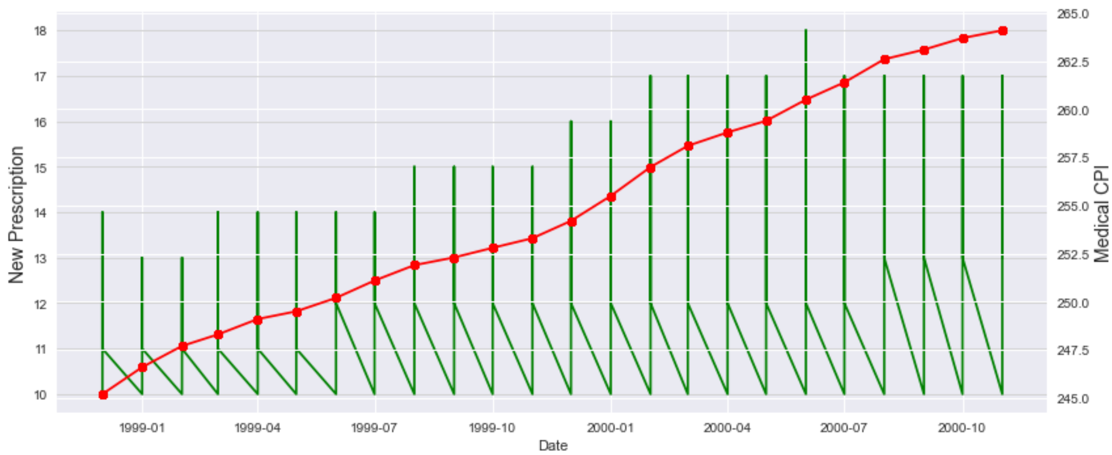
```
new_lowerlimit*115*3.5
```

```
42.500511429620374
```

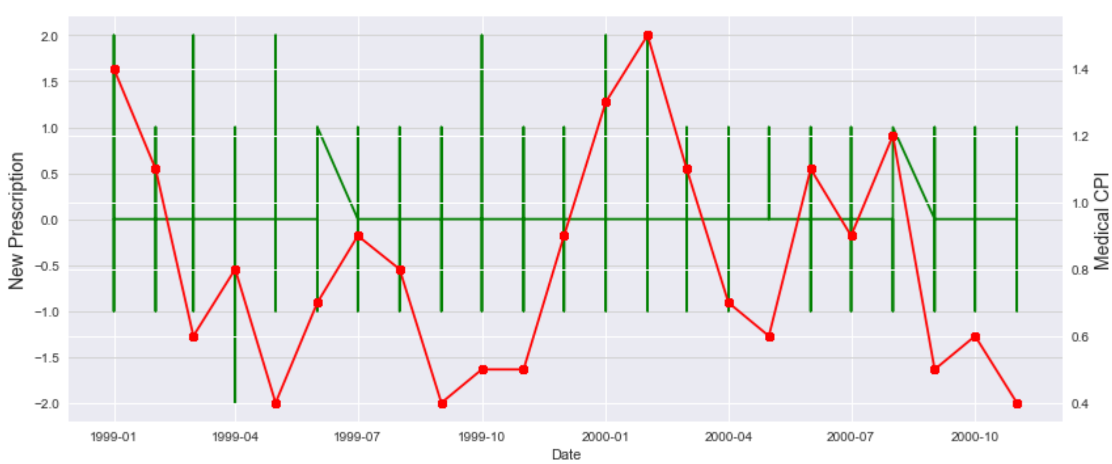
```
regression_model_9= smf.ols('NewPrescDiff ~ SamplesDiff + DetailsDiff
+ CompetitorNewDiff + Psych + MedicalCPIDiff + DRGDiff', data =
diff_df).fit()
regression_model_9.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

======================================================================
========
Dep. Variable:           NewPrescDiff   R-squared:
0.614
Model:                            OLS   Adj. R-squared:
0.613
Method:                 Least Squares   F-statistic:
1114.
Date:                Mon, 10 Oct 2022   Prob (F-statistic):
0.00
Time:                        20:29:36   Log-Likelihood:
206.36
No. Observations:                4209   AIC:
-398.7
Df Residuals:                    4202   BIC:
-354.3
Df Model:                           6

Covariance Type:            nonrobust

======================================================================
===============
                        coef    std err          t      P>|t|
[0.025      0.975]
----------------------------------------------------------------------
---------------
Intercept             0.0825      0.010      8.266      0.000
0.063       0.102
SamplesDiff           0.0354      0.001     58.162      0.000
0.034       0.037
DetailsDiff           0.1114      0.003     41.214      0.000
0.106       0.117
CompetitorNewDiff     0.0003      0.000      0.650      0.516      -
0.001       0.001
```

```
Psych                -0.0699      0.028     -2.496      0.013      -
0.125      -0.015
MedicalCPIDiff       -0.0020      0.011     -0.177      0.860      -
0.024       0.020
DRGDiff            5.915e-05      0.000      0.384      0.701      -
0.000       0.000
======================================================================
========
Omnibus:                     391.998   Durbin-Watson:
2.007
Prob(Omnibus):                 0.000   Jarque-Bera (JB):
507.407
Skew:                          0.816   Prob(JB):
6.58e-111
Kurtosis:                      3.477   Cond. No.
188.
======================================================================
========

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

%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn')

# group = df.groupby(['Medical CPI','NewPrescription'])
# group.size().unstack().plot(kind='bar')

# fig, ax = plt.subplots(figsize=(12,5))
# ax2 = ax.twinx()
# ax.set_xlabel('Date')

# ax.plot(df['Date'],df['NewPrescription'], color='green', marker='x')
# ax2.plot(df['Date'], df['Medical CPI'], color='red', marker='o')
# ax.set_ylabel('New Prescription',fontsize=14)
# ax2.set_ylabel('Medical CPI',fontsize=14)
# ax.yaxis.grid(color='lightgray')

# plt.tight_layout()
# plt.show()
```




```
# fig, ax = plt.subplots(figsize=(12,5))
# ax2 = ax.twinx()
# ax.set_xlabel('Date')

# ax.plot(diff_df['Date'],diff_df['NewPrescDiff'], color='green',
marker='x')
# ax2.plot(diff_df['Date'], diff_df['MedicalCPIDiff'], color='red',
marker='o')
# ax.set_ylabel('New Prescription',fontsize=14)
# ax2.set_ylabel('Medical CPI',fontsize=14)
# ax.yaxis.grid(color='lightgray')

# plt.tight_layout()
# plt.show()
```



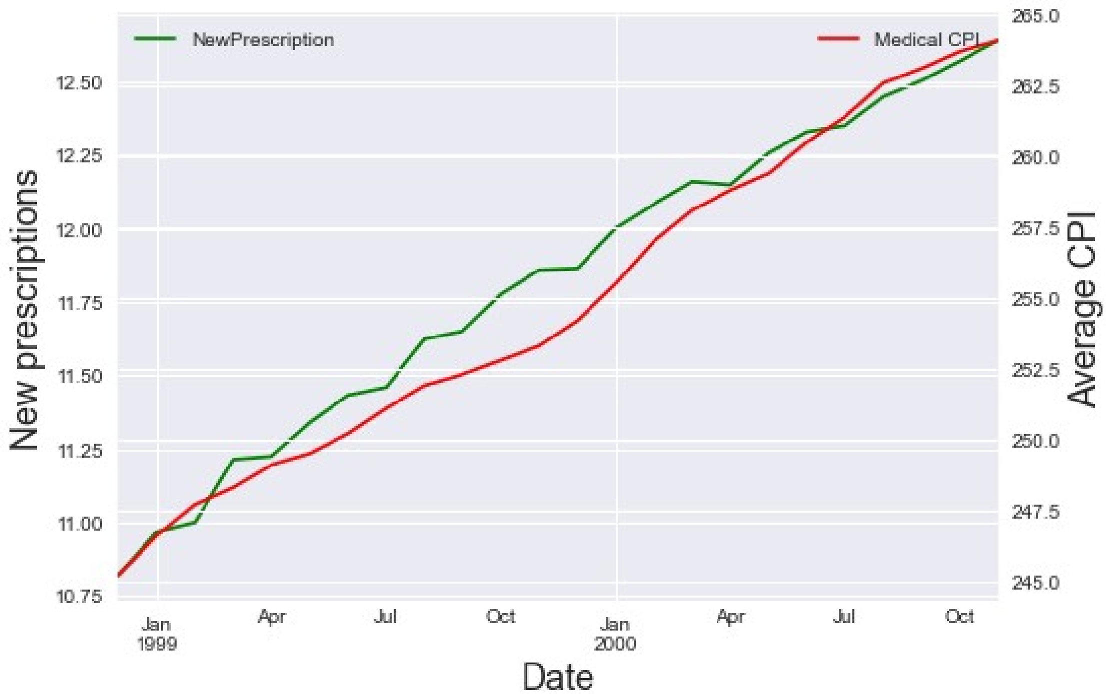

```
# df.rename(columns={"Medical CPI": "MedicalCPI"})
# df
# regression_model_3 = smf.ols('NewPrescription~
Samples+Details+MedicalCPI',data=df).fit()
# regression_model_3.summary()
```

```
      Physician       Date    ID_Date  NewPrescription  Samples  
Details  \
0             1 1998-12-01    1_36130               10        0        
0
1             2 1998-12-01    2_36130               11        0        
2
2             3 1998-12-01    3_36130               11        0        
1
3             4 1998-12-01    4_36130               10        0        
0
4             5 1998-12-01    5_36130               11        0        
2
...         ...        ...        ...              ...      ...      .
..
4387        179 2000-11-01  179_36831               12        0        
0
4388        180 2000-11-01  180_36831               13        0        
1
4389        181 2000-11-01  181_36831               15        0        
4
4390        182 2000-11-01  182_36831               14        2        
4
4391        183 2000-11-01  183_36831               13        0        
0

      CompetitorNew  Psych  Medical CPI     DRG  
0                11      1        245.2  389.85  
1                 2      0        245.2  389.85  
2                38      0        245.2  389.85  
3                15      0        245.2  389.85  
4                 2      0        245.2  389.85  
...             ...    ...          ...     ...  
4387             36      0        264.1  441.84  
4388             43      0        264.1  441.84  
4389             42      0        264.1  441.84  
4390             19      0        264.1  441.84  
4391              7      0        264.1  441.84  

[4392 rows x 10 columns]
```

```python
df.Date = pd.to_datetime(df.Date)

fig, ax1 = plt.subplots()

ax2 = ax1.twinx()

df.groupby('Date').NewPrescription.mean().plot(ax=ax1,color='green')
df.groupby('Date')[['Medical CPI']].mean().plot(ax=ax2, color='red')

ax1.set_xlabel('Date', fontsize=18)
```

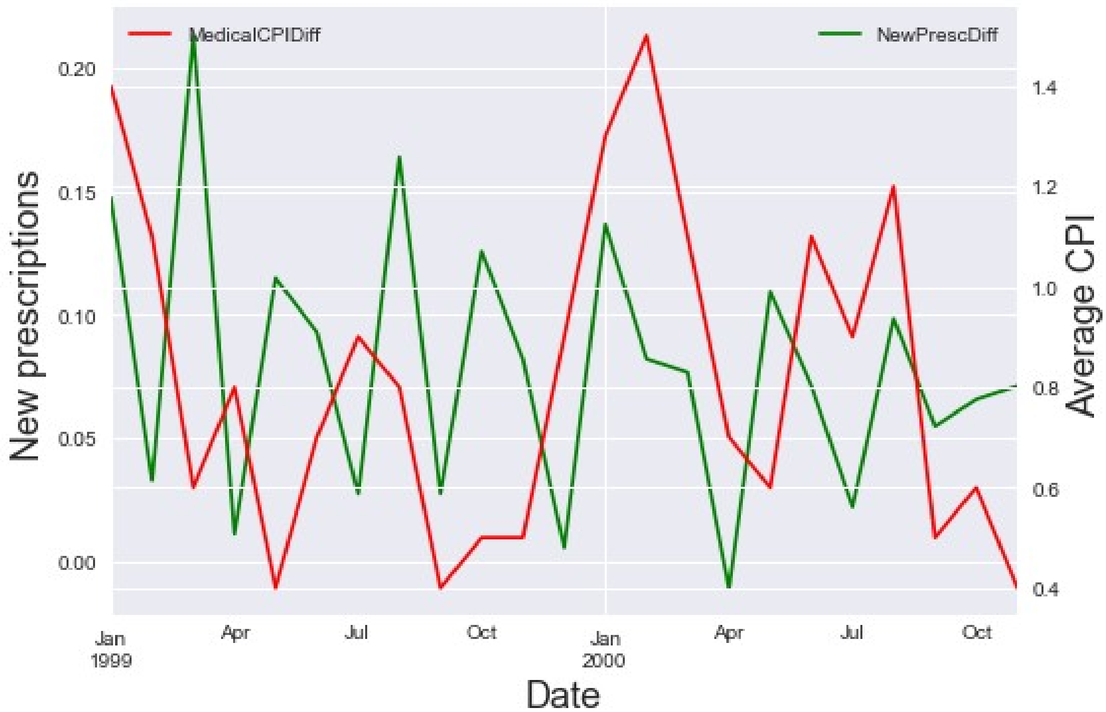
```
ax1.set_ylabel('New prescriptions', fontsize=18)
ax2.set_ylabel('Average CPI', fontsize=18)

ax1.legend(loc=0)
ax2.legend(loc=1)
```

```
<matplotlib.legend.Legend at 0x7f9b43ea5490>
```




```
diff_df.Date = pd.to_datetime(diff_df.Date)

fig, ax1 = plt.subplots()

ax2 = ax1.twinx()

diff_df.groupby('Date').NewPrescDiff.mean().plot(ax=ax1,color='green')
diff_df.groupby('Date')[['MedicalCPIDiff']].mean().plot(ax=ax2,
color='red')

ax1.set_xlabel('Date', fontsize=18)
ax1.set_ylabel('New prescriptions', fontsize=18)
ax2.set_ylabel('Average CPI', fontsize=18)

ax1.legend(loc=1)
ax2.legend(loc=2)
```

```
<matplotlib.legend.Legend at 0x7f9b3417f400>
```



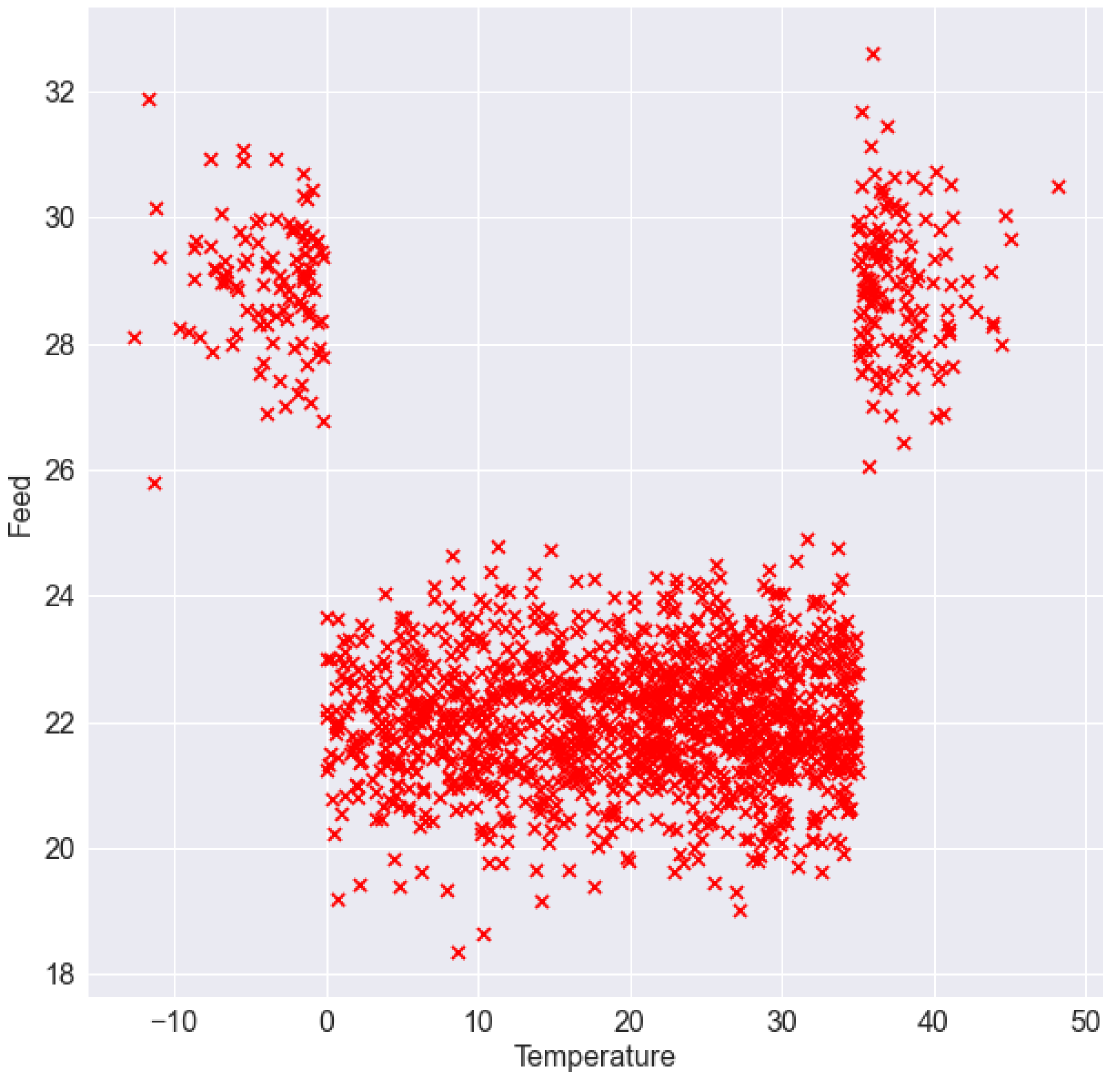

```
egg_prod_df= pd.read_csv('egg_production.csv')
egg_prod_df

          eggs       feed  temperature
0     1.944645  28.521682    -3.920247
1     2.367084  20.810192     7.489837
2     1.361380  29.259575    -5.425451
3     1.763221  22.245235     1.486627
4     2.003410  23.331641     9.976938
...        ...        ...          ...
1547  1.641620  22.939631    11.102256
1548  2.660458  22.726205    18.844973
1549  1.367134  21.987339     3.645734
1550  1.724994  22.862650    12.987750
1551  2.305316  22.943871    20.767244

[1552 rows x 3 columns]

regression_egg=smf.ols('eggs ~ feed', data = egg_prod_df).fit()
regression_egg.summary()

<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
```

```
==============================================================================
========
Dep. Variable:                   eggs   R-squared:
0.176
Model:                            OLS   Adj. R-squared:
0.175
Method:                 Least Squares   F-statistic:
331.1
Date:                Tue, 11 Oct 2022   Prob (F-statistic):
3.38e-67
Time:                        19:26:11   Log-Likelihood:
-1190.7
No. Observations:                1552   AIC:
2385.
Df Residuals:                    1550   BIC:
2396.
Df Model:                           1

Covariance Type:            nonrobust

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
Intercept      3.8328      0.114     33.635      0.000       3.609
4.056
feed          -0.0891      0.005    -18.195      0.000      -0.099
-0.080
==============================================================================
========
Omnibus:                        0.504   Durbin-Watson:
2.111
Prob(Omnibus):                  0.777   Jarque-Bera (JB):
0.484
Skew:                           0.043   Prob(JB):
0.785
Kurtosis:                       3.005   Cond. No.
201.
==============================================================================
========

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

```
regression_egg_2 = smf.ols('eggs ~ feed + temperature', data =
egg_prod_df).fit()
regression_egg_2.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

==============================================================================
========
Dep. Variable:                   eggs   R-squared:
0.176
Model:                            OLS   Adj. R-squared:
0.175
Method:                 Least Squares   F-statistic:
165.6
Date:                Tue, 11 Oct 2022   Prob (F-statistic):
6.63e-66
Time:                        19:27:25   Log-Likelihood:
-1190.5
No. Observations:                1552   AIC:
2387.
Df Residuals:                    1549   BIC:
2403.
Df Model:                           2

Covariance Type:            nonrobust

==============================================================================
=========
                  coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
---------
Intercept       3.8449      0.116     33.137      0.000       3.617
4.072
feed           -0.0891      0.005    -18.190      0.000      -0.099
-0.079
temperature    -0.0006      0.001     -0.557      0.577      -0.003
0.002
==============================================================================
========
Omnibus:                        0.513   Durbin-Watson:
2.111
Prob(Omnibus):                  0.774   Jarque-Bera (JB):
0.489
Skew:                           0.043   Prob(JB):
0.783
Kurtosis:                       3.007   Cond. No.
278.
```

```
==============================================================================
========

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

# regression_egg_3 = smf.ols('temperature ~ feed', data =
egg_prod_df).fit()
# regression_egg_3.summary()

plt.figure(figsize=(10,10))

plt.scatter(egg_prod_df.temperature, egg_prod_df.feed, marker='x',
color='red')

plt.xlabel('Temperature', fontsize=16)
plt.ylabel('Feed', fontsize=16)

plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

sns.despine()
```




```
egg_prod_df['X']=""
egg_prod_df['X'] =
np.where(np.logical_or(egg_prod_df['temperature']<0,
egg_prod_df['temperature']>35),1,0)
egg_prod_df
```

	eggs	feed	temperature	X
0	1.944645	28.521682	-3.920247	1
1	2.367084	20.810192	7.489837	0
2	1.361380	29.259575	-5.425451	1
3	1.763221	22.245235	1.486627	0
4	2.003410	23.331641	9.976938	0
...	...	...	...	..
1547	1.641620	22.939631	11.102256	0
1548	2.660458	22.726205	18.844973	0
1549	1.367134	21.987339	3.645734	0

```
1550  1.724994  22.862650    12.987750  0
1551  2.305316  22.943871    20.767244  0

[1552 rows x 4 columns]
```

```
group=egg_prod_df.groupby('X').count()
group
```

```
   eggs  feed  temperature
X
0  1310  1310         1310
1   242   242          242
```

```
regression_egg_4= smf.ols('eggs ~ feed + temperature + X', data =
egg_prod_df).fit()
regression_egg_4.summary()
```

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results

======================================================================
========
Dep. Variable:                   eggs   R-squared:
0.236
Model:                            OLS   Adj. R-squared:
0.234
Method:                 Least Squares   F-statistic:
159.0
Date:                Tue, 11 Oct 2022   Prob (F-statistic):
7.96e-90
Time:                        20:18:55   Log-Likelihood:
-1132.5
No. Observations:                1552   AIC:
2273.
Df Residuals:                    1548   BIC:
2294.
Df Model:                           3

Covariance Type:            nonrobust

======================================================================
=========
                  coef    std err          t      P>|t|      [0.025
0.975]
----------------------------------------------------------------------
---------
Intercept       1.0529      0.278      3.786      0.000       0.507
1.598
feed            0.0388      0.013      3.081      0.002       0.014
0.063
```

```
temperature    -0.0007      0.001     -0.695      0.487      -0.003
0.001
X              -1.0276      0.094    -10.966      0.000      -1.211
-0.844
======================================================================
========
Omnibus:                        0.478   Durbin-Watson:
2.108
Prob(Omnibus):                  0.787   Jarque-Bera (JB):
0.390
Skew:                           0.023   Prob(JB):
0.823
Kurtosis:                       3.062   Cond. No.
724.
======================================================================
========

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

```python
prediction_data = pd.DataFrame({'feed':[25], 'temperature':[-1], 'X':
[1]})

regression_egg_4.predict(prediction_data)
```

```
0    0.994741
dtype: float64
```

```python
regression_egg_4.get_prediction(prediction_data).summary_frame(alpha=0
.1)
```

```
       mean   mean_se  mean_ci_lower  mean_ci_upper  obs_ci_lower  \
0  0.994741  0.062953       0.891131       1.098351       0.16108

   obs_ci_upper
0      1.828403
```

```python
# regression_egg_4.get_prediction(prediction_data).summary_frame()
```