I. 1-1 processing missing values.ipynb

The methods used are as follows:

1. Delete the row that has the missing value
2. Delete the variable that has the missing value
3. Use acceptable previous/next data
4. Replace the missing values with the mean value of the column
5. Insert missing values with the middle value of the column
6. Using the k nearest neighbor method

I open the data first. The names of the columns are in the third line and the first two lines are data source information and have nothing to do with the variables. I will correct the naming of the columns (variables). And we have a dataframe ready to follow the work.

From the obtained information, we have 4 missing values. 2 related to Population growth column and the other two each belong to Area (sq. km) and Total population columns.

1. Delete the row that has the missing value

   From the initial 15 rows, 4 were deleted and we have 11 usable and complete data. That is, we deleted about 25% of the data, while there might be valuable information inside. We try to use better methods.

2. Deleting a variable that has a missing value:

   Among the variables we have, (of course, we have not yet entered the modeling phase), but according to the text of the question, we target Coronavirus Cases as a variable (in the future) and according to the information we have, we have 4 useful variables to use in the model.

   If we delete all 3 columns, only 1 variable is left to continue and we have thrown away a lot of information, which is not logical at all. For example, if our mode is regression, we have multiple regression at the beginning, and at the end of removing the columns, only one variable remains to estimate the target value. So this method is not suitable at all. It may be acceptable to use this method only when we have a column that has missing values in most of the rows. In that case, removing it can be suggested as an option (if it is not a better option).

3. Use acceptable previous/next data

   In each column, we fill the missing value of the desired row with the first recorded value before it. Or I use the first value after that. (Does not matter)

That is, in total, we put 4 duplicate values instead of 4 missing values. Compared to the previous options, it is relatively better, but not necessarily desirable. The information obtained may be invalid.

4. Replace the missing values with the mean value of the column

Put the average size of each column in place of the missing value of that column. The advantage of the method is that the average value does not change in the end. I used impleImputer(strategy='mean') because I had trouble doing this with data4.mean() in pandas. which is the same result. If one of the missing columns was qualitative, this method would not work.

5. Insert missing values with the middle value of the column

Like the previous case, just use the median instead of the mean. It can also be used for non-quantitative data. If we draw a box plot of the data, it will not change in this method and the points will remain in the previous place.

6. Using the k nearest neighbor method

In this method, I use the KNN algorithm to replace it with a reasonable value that is closest to the possible missing answer based on the total value of the variables of each row that has a missing value. This is a classification method and because it uses all the data in all the other rows and columns to find the missing value, I think it is preferable. And I continue with the same result for the next steps.

Finally, I save the results in OUT 1-1 data (processed missing values).csv for the next step.

Note: Of course, another method was proposed in the class, which is more suitable for categorical data. To fill in the missing value with the highest frequency of each column, which was not useful here at all.

II. 1-2 encoding categorical values.ipynb

For the regression model, we only deal with the numerical value, but it does not mean that the categorical qualitative variable cannot be used with the regression model. It only needs pre-processing to be converted into a numerical value that can be used in equations. Because it is available in the same way, it cannot be added together with the other variables in the optimization equations and the error can be optimized to obtain the regression equation. For each group or category of bags in this International Visitors column, we must create a similar code (for example, if we have 5 groups, numbers 1 to 5) and then add a column to the variables according to each of these numbers, with the concept that "Does this line have the characteristic of a certain number (for example, 1 for the first column... to 5 for the fifth) or not" and put the number of the similar value as 0 or 1. If the desired variable is binary, in the first step we have 0 to 1 and that is enough. But here our variable is 4 states ABCD.

A: 0

B: 1

C: 2

D: 3

The encoded_international_visitors dataframe has 4 columns as a result of this step. These should be placed in the main dataframe instead of the International Visitors column.

Of course, in the end, one of these should be removed because if we have anything for ABC, the value of column D will be obtained depending on these and it is not needed. I save the result in OUT 1-2 data (categorial values encoded).csv for the next step.

III. 1-3 feature scaling.ipynb

Because the model we are running is supposed to be a regression model, this is not very mandatory. While, for example, for a method like k nearest neighbor, the effect of the magnitude of the vector of variables can completely change the result. But in the regression model, since a coefficient such as each of the variables (xi's) is obtained, which shows its relationship with the target, this effect is then reduced in the model itself, and for example, a variable that has a value of 106 is equal to the linear relationship. It takes the coefficient of importance with the target variable and not proportional to the fact that its value is greater than about 0.1 of a variable. Therefore, we do not have to standardize in regression, and all kinds of standardization methods are not very useful, except in special situations where the logarithm of the size of the variables is very different from each other, maybe we use these methods for ease of work or to prevent possible overflow . In the code file, the MinMaxScaler and Normalizer functions are used, which firstly maps all quantitative data columns to the distance between 0 and 1 (based on the minimum-maximum standardization method) and secondly calculates the data normalization based on the 2-vector software and The data does not affect our future regression model and it is unnecessary to use them. But data standardization for regression can be useful. It means to make the mean of the columns equal to zero and their variance equal to 1. In order for the regression to give a better model, it is necessary that the distribution of the data is close to the normal distribution. In the lesson, we learned that if we deal with averages, this normal distribution exists, but here the results of our columns are not averages (the average of several samples is meant), so this distribution can be brought closer to the normal distribution by using transformation. So that the answer obtained from the multiple regression that I get in the following is more accurate and correct.

So, first, I prepare and separate the columns using numpy. (Because of the errors that occurred, I did not do this on the entire data frame at once, and on the other hand, since there were not more than 3 columns, I did it one by one) and then I applied scale2=StandardScaler() on each of these presentations and in Finally, I rewrote the amount of columns like these presentations on the main data frame. Now the data distribution on each of these 3 columns is close to normal. with mean 0 and unit variance.

z = (x - u) / s

I save the answer of this conversion on the quantitative columns Area (sq. km), Total population and Population growth in OUT 1-3 data(standard-transformed).csv to continue working.

Note: Without this transformation, there would be no special problem for the regression model.

Note 2: Another reason for using this transformation is explained in the next question, which is to reduce the impact of possible outliers.

IV. 1-4 processing outliers.ipynb

The first thing I do is to check whether there are outliers in the data and if so, in which columns and how is their status. I check each of the columns by drawing their boxplot. Yes, we have outliers.

In population growth, we have three outliers. And in Total population and Area (sq. km) one and two respectively. We have two main methods to find and remove these, one is IQR and the other is Z-Score. Since the IQR method matches the graphs I drew and is a better method for detecting outliers, I use it. For each column, we just need to find the value of the first and third quartile. I get the difference of these. According to this method, we reject all the data that are in the distance of one and a half times of this number compared to the first and third quartiles as outliers.

The result of (column123<Q1-1.5*IQR) | (column123>Q3+1.5*IQR) actually found the same outlier data in the initial box plots for me. Now I will remove these to make the point of the question clear.

After deleting 10 lines of data, I was left with 5 of the initial data, that is, about 30% of the data. The question is, was this method appropriate?

In general, since regression is sensitive to outlier data, I think removing outlier data can be a suitable method, if the effect of outlier data is not important for the correctness of the model. The bank transaction example that was mentioned in the class is one of the examples that cannot be deleted. (Of course, regression itself is usually not a very good method in such examples)

In my opinion, it is not appropriate to remove it here for the following reasons:

1. Is 30% of the data really outliers? Such amount is too much to be passed over and deleted without checking.

2. Considering the type of data I am dealing with, it is very unlikely that these reported numbers are errors. For example, about India, the size of the area and the number of population are clearly reported correctly. Or the population growth of Kuwait and Qatar is relevant to the situation of these countries.
3. So, we probably don't have wrong information here, and the outliers of these values make more sense than their balance with other figures

The main question is, how much should these strange values have an effect on our model? If it is to be deleted, I mean not to affect them at all, which I think is throwing away information that can be useful. Outlier data can be replaced with values such as the average or, for example, with clustering methods based on similar data or even missing data with the nearest neighbor algorithm, so that, for example, if we have an outlier data field, the information of other fields can be used. and the effect of discarding data is less noticeable. But in my opinion, for the reasons I said, these numbers are significant and it is not appropriate to do these things here. What can be done is to use transformations, for example, the same transformation that was used to bring the data closer to the normal distribution in the previous question, so that these numbers, even though they have a meaning, are not completely removed, but their effect on the answer that is finally It becomes less. Similarly, RobustScaler can be used, which sets the median to zero and divides the values by the distance between the quartiles that we calculated (above), which reduces the impact of the values that are outside the game by 1.5 times. And this happens as a result of reducing their value and making them positive and negative on both sides of the middle. I did both (the first one in the previous question), but finally I preferred to use the same result of the previous question, i.e. the result of StandardScaler() to continue the work, which has two advantages: it reduces the impact of outlier data to some extent. It is slow and it makes it easier for the regression model to converge to a more accurate model. Apart from this, the RobustScaler method in this particular example does not have much effect on the whole story because all the outliers are located on one side of the boxplot and their value is small.

The result of this part is the same file as the previous step, which, of course, I saved again under the name OUT 1-4 data(outlier_passed).csv for the convenience of continuing work.

Note: In order to make sure that it was better not to remove the outliers, in the last step, I did the regression model once with these data and once after removing the outliers. The result of not deleting was better, so I continued the same method. Of course, I did not report this test because the number of data was very small, reporting the results and graphs were completely meaningless. But in addition to the reasons mentioned above, this method can also be used to answer the question whether it is appropriate to remove the outer layer or not.

V. 1-5 multiple regression.ipynb

I use lrm=LinearRegression() to generate the model. This model estimates the amount of Coronavirus Cases based on the value of 6 input variables. To get model information, including coefficients, etc., I use the table I get from statsmodels.formula.api. And finally, I count the number of errors (it was not in the table).

Information as follows:

The first line shows the value of the width from the origin and the next lines each variable coefficient written like it in the multiple linear regression equation.

Intercept	*50366.943585*
Population growth	-7295.630057
Total population	-2979.059785
Area (sq. km)	11846.317535
A	130108.758225
B	-28201.448627
C	-41830.434172

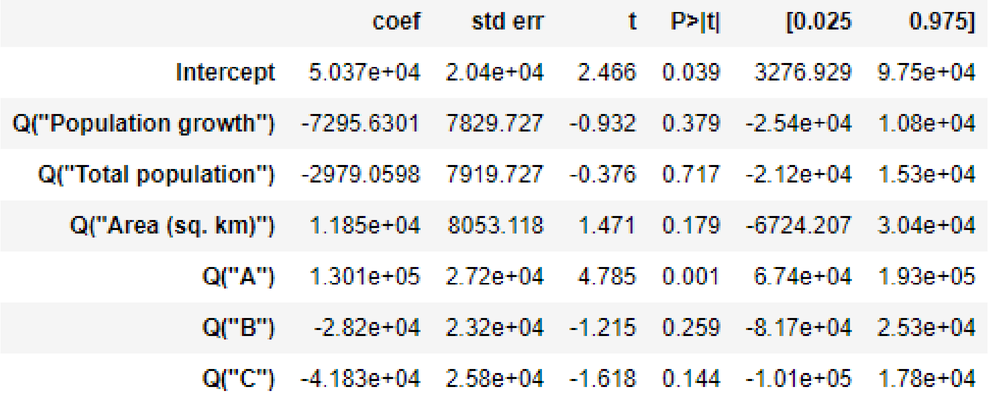
Errors are as following:

MSE	*405381115.26471084*
RMSE	*20134.078455809962*

Model evaluation criteria:

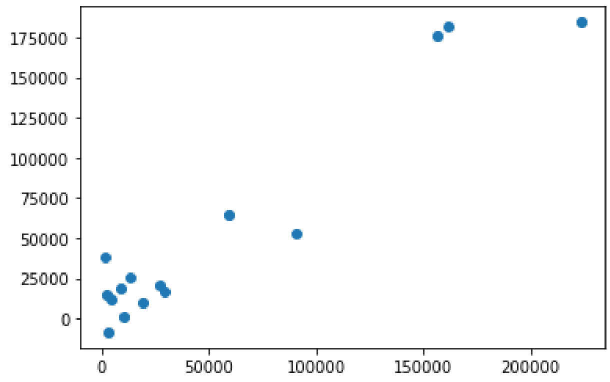
First, for the pi value test:

We said that the smaller the value of pi is than alpha, H_0 can be assumed (H_0 : the equality of the coefficient of the variable with zero, which means there is no linear relationship between y and xi) in the table below, the value of pi for each line except B and the Bias Term are greater than alpha (if we take alpha 0.05 as usual), which shows that it is not a good model (the relationship between the variables and the target variable are not linear).



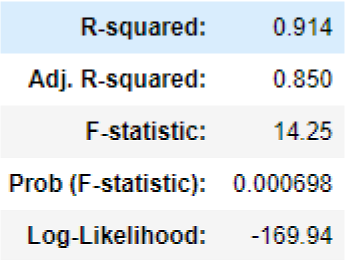
	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.037e+04	2.04e+04	2.466	0.039	3276.929	9.75e+04
Q("Population growth")	-7295.6301	7829.727	-0.932	0.379	-2.54e+04	1.08e+04
Q("Total population")	-2979.0598	7919.727	-0.376	0.717	-2.12e+04	1.53e+04
Q("Area (sq. km)")	1.185e+04	8053.118	1.471	0.179	-6724.207	3.04e+04
Q("A")	1.301e+05	2.72e+04	4.785	0.001	6.74e+04	1.93e+05
Q("B")	-2.82e+04	2.32e+04	-1.215	0.259	-8.17e+04	2.53e+04
Q("C")	-4.183e+04	2.58e+04	-1.618	0.144	-1.01e+05	1.78e+04

Of course, it is true that all these values are close to zero, but they are not small enough, so our model is not very reliable. (It can be seen in the diagram that the obtained shape is not very similar to a line). In addition, it is true that we do not have much to do with the obtained confidence interval, but it is not bad to notice that in some of these, zero is inside the interval, and this makes the assumption of H0 unrejectable.



So far, of course, it is enough to reject the model, but to complete all the parts:

Another evaluation criterion that can be used is the coefficient of determination test or the regression coefficient, or the second line of the table below, which is equal to the correlation coefficient to the power of 2 that we use for multiple regression. If this value is close to 1, we have a good model. Here we got 0.8.



R-squared:	0.914
Adj. R-squared:	0.850
F-statistic:	14.25
Prob (F-statistic):	0.000698
Log-Likelihood:	-169.94

Of course, in the event of a conflict between these two criteria, we said that the validity of the test is higher. Therefore, the model is still rejected. With the rest of the numbers in this table, we have nothing to do with linear regression.

Note: Of course, the results may be due to the fact that the number of data is very small.

VI. 1-6 non linear regression.ipynb

From the data we prepared, we create a new data frame with the Total population column, the first column of which is the value of Total population and the second column of the same values to the power of 2. And I will create a linear regression model for two variables (columns of the new data frame) for the target variable of Coronavirus Cases, the working method is the same as the previous question, only we do not need analyzes and errors, etc. This model is equivalent to the model asked in the question. The coefficients obtained from this method are interpreted as follows:

The coefficient obtained for a variable such as Total population → coefficient x

The coefficient obtained for a variable such as Total population**2 → x2 coefficient

Bias term instead of itself

So the obtained numbers for the unknowns:

a	-15440.070964
b	44642.402653
c	69393.4042975207