Question 1

The variables that contain missing values are as follows:

|  |  |  |
| --- | --- | --- |
| Variables | Description | Missing Value Expression |
| Actual | Actual date of claim settlement | Blanks |
| Terms | Internal terms and condition code | “???” and “Unkn” |
| Claim\_ID | Unique identifier of claim | Blanks |

Table 1:

**Reading the dataset as a Pandas dataframe:**

#Import necessary libraries

import pandas as pd

# Read CSV file and specify empty strings as missing values

df = pd.read\_csv("C:/Users/shann/Documents/SUSS Modules/ECA.csv", na\_values=['', 'Unkn', '???'])

# Get the names of columns with missing values

cols\_with\_missing = df.columns[df.isna().any()].tolist()

# Print the names of columns with missing values

print("Columns with missing values:", cols\_with\_missing)

Question 2 As part of data preparation, treat the missing data, and explain your rationale of the treatments. (15 marks)

For data analysis, the data must first be prepped in order for accurate analysis of the data. There are mainly 3 methods to treat the missing data from the CSV file, this includes:

**Deletion:** This approach is to delete any rows or columns that contain missing values. This approach is straightforward and can be easy to do. However, by doing so can lead to a loss of valuable information and reduce the sample size. Deletion is usually recommended when the missing data is small or when it does not have a significant impact on the analysis.

**Imputation:** Imputation involves replacing missing values with a reasonable estimate. This approach preserves the sample size and ensures that all available data is used. Common imputation methods include mean imputation, median imputation, and mode imputation. However, it is essential to note that imputation can introduce bias and affect the accuracy of the analysis if not done appropriately.

**Modelling:** Modelling involves using statistical models to predict missing values. For example, multiple imputation can be used to estimate missing values using a regression model. This approach produces multiple imputed datasets, and the results are combined to produce more accurate estimates. However, modelling requires more advanced statistical techniques and may not always be feasible.

Therefore after careful consideration, I have decided on the following to treat the missing data found in the file:

**Actual Variable (Blanks):**

To treat the blanks in the “Actual” variable column, I would use the method of Modelling. By using a linear regression model, i can accurately estimate the “Actual” date of payment for the claims.

Some of the claims take a unreasonable duration for payment therefore, we cannot use the Modelling method

As the blank data consists of a significant amount of data (1677 datapoints), Deletion can also no longer be an option as it will affect the accuracy of the data due to the huge reduction in sample size.

Therefore, the most suitable method to estimate the values of the blanks is through Modelling. Modelling will allow a more accurate estimate using a linear regression line as it shows a best fit line to estimate when payment can be made based on the Planned date of claim settlement.

**Python code to be the same as Question 5**

**Terms Variable (“???” and “Unkn”):**

The values “???” and “Unkn” will not be accepted by python as a value. Therefore, we need to prepare this data for analysis. I will use the Deletion Method to clean up this data as the Term Variable is a unique code that cannot be estimated via its mean, median or mode which means that the method of Imputation cannot be used.

Additionally, these values are uniquely set to each Claim\_ID which means that they cannot be estimated using a Model. Therefore, the Modelling method could not be used in this case as well.

In conclusion, only the method of Deletion would best suit this anomaly as it is not only unique to each Claimant but also consists of a small number of data points affected which does not affect the overall analysis if it would be deleted.

# Import required library

import numpy as np

import pandas as pd

# Load the CSV file

df = pd.read\_csv("C:/Users/shann/Documents/SUSS Modules/ECA.csv")

# Remove missing values in the "Terms" column

df = df[~df['Terms'].isin(['???', 'Unkn'])]

# Print Updated CSV file with remaining values

print (df)

**Claim\_ID Variable (Blanks) :**

To treat the blanks in this variable, I would use the Deletion method.

The unique identifier of the claim is unique to each claimant and cannot be estimated or use to a model to accurately predict the ID for each claimant. Therefore, both Imputation and Modelling methods cannot be used to clean this data. The only logical method to use is to delete the datapoints that has the blanks. As there are only 5 data points that are affected by the blanks in the Claim\_ID, it would not have a significant impact on the overall analysis of the data. This makes the Deletion Method to be the most suitable to clean this data.

# Import required library

import numpy as np

import pandas as pd

# Load the CSV file

df = pd.read\_csv("C:/Users/shann/Documents/SUSS Modules/ECA.csv")

# Remove missing values in the "Claim\_ID" column

df.dropna(subset=['Claim\_ID'], inplace=True)

# Print Updated CSV file with remaining values

print (df)

So in order to clean the data, the Python code should be as follows:

# Import required library

import numpy as np

import pandas as pd

# Load the CSV file

df = pd.read\_csv("C:/Users/shann/Documents/SUSS Modules/ECA.csv")

# Remove missing values in the "Claim\_ID" column

df.dropna(subset=['Claim\_ID'], inplace=True)

# Remove missing values and invalid values in the "Terms" column

df = df[~df['Terms'].isin(['???', 'Unkn'])]

# Convert the 'Planned' and 'Actual' columns to datetime format using the specified format string

df['Planned'] = pd.to\_datetime(df['Planned'], format='%d/%m/%Y')

df['Actual'] = pd.to\_datetime(df['Actual'], format='%d/%m/%Y %H:%M')

# Calculate the duration between 'Planned' and 'Actual' columns using the absolute value to avoid negative durations

df['Duration'] = (df['Actual'] - df['Planned']).abs()

# Calculate the mean duration between 'Planned' and 'Actual' columns

mean\_duration = df['Duration'].mean()

# Estimate the missing 'Actual' dates using the mean duration

df['Actual'] = np.where(df['Actual'].notnull(), df['Actual'], df['Planned'] + pd.Timedelta(mean\_duration))

# Print the updated dataframe with the absolute value of the durations in days

print(df)

Question 3

**Handling Duplicates:**

Duplicate records in a dataset can skew the analysis and create inaccurate results. To handle duplicates, we can use the ‘drop\_duplicates’ method in Pandas. This method removes duplicate rows based on the specified columns.

# Import the necessary libraries

import pandas as pd

# Load the dataset

df = pd.read\_csv(‘ECA.csv’)

# drop duplicate rows based on 'Claim\_ID'

df.drop\_duplicates(subset=['Claim\_ID'], keep='first', inplace=True)

# Print updated data without blanks in ‘Claim\_ID’

print (df)

In the above python code, the ‘subset’ specifies the columns to consider for duplicates, and the ‘keep’ specifies on which duplicate to keep (first, last, or None).

**Data Transformation:**

Data transformation involves converting data from one format to another, such as changing data types, scaling data, or encoding categorical variables. Pandas provides several methods for data transformation, such as astype, apply, and replace.

For the ECA.csv file, we can utilise the “to\_datetime” to convert the 'Actual' column to datetime format using the specified format string to ensure that the format for data points would be kept the same as the ‘Planned’ column.

# Import the necessary libraries

import pandas as pd

# Load the dataset

df = pd.read\_csv(‘ECA.csv’)

# convert 'Actual' column to datetime format

df['Actual'] = pd.to\_datetime(df['Actual'], format='%d/%m/%Y %H:%M')

# Print updated data after the date format change

print (df)

**Handling Outliers:**

Outliers are data points that are significantly different from the rest of the data and can impact the accuracy of the analysis. To handle outliers, we can use statistical techniques such as Z-score or Interquartile Range (IQR). In this Particular case, we can find the duration between the “Planned” and “Actual” columns and calculate the IQR for the total duration of the payment.

# Import necessary libraries

import pandas as pd

import numpy as np

# Read the CSV file into a Pandas dataframe

df = pd.read\_csv(‘ECA.csv’)

# Remove missing values in the "Claim\_ID" column

df.dropna(subset=['Claim\_ID'], inplace=True)

# Remove missing values and invalid values in the "Terms" column

df = df[~df['Terms'].isin(['???', 'Unkn'])]

# Convert the 'Planned' and 'Actual' columns to datetime format using the specified format string

df['Planned'] = pd.to\_datetime(df['Planned'], format='%d/%m/%Y')

df['Actual'] = pd.to\_datetime(df['Actual'], format='%d/%m/%Y %H:%M')

# Calculate the duration between 'Planned' and 'Actual' columns using the absolute value to avoid negative durations

df['Duration'] = (df['Actual'] - df['Planned']).abs()

# Calculate the mean duration between 'Planned' and 'Actual' columns

mean\_duration = df['Duration'].mean()

# Estimate the missing 'Actual' dates using the mean duration

df['Actual'] = np.where(df['Actual'].notnull(), df['Actual'], df['Planned'] + pd.Timedelta(mean\_duration))

# Calculate the estimated 'Actual' dates by adding the duration to the 'Planned' dates

df['Estimated Actual'] = df['Planned'] + df['Duration']

# Convert the durations to days

df['Duration'] = df['Duration'].dt.days

# Calculate the IQR for the duration column

Q1 = df['Duration'].quantile(0.25)

Q3 = df['Duration'].quantile(0.75)

IQR = Q3 - Q1

# Calculate the lower and upper bounds for the data

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Remove any outliers from the dataset

df = df[(df['Duration'] > lower\_bound) & (df['Duration'] < upper\_bound)]

# Print the updated dataset without outliers from the ‘Actual’ column

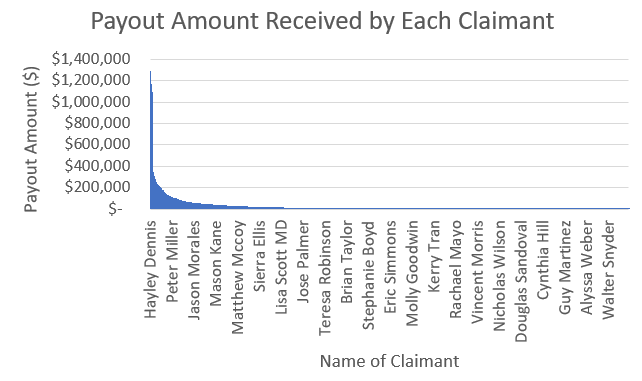
print(df)

In the Python code above, I calculated the IQR for the duration between ‘Planned’ and ‘Actual’ columns and remove any data points that fall outside of 1.5 times the IQR range.

There are many more data preparation tasks that can be performed depending on the specific needs of the analysis. Pandas provides a wide range of methods and functions to help with data preparation, and other libraries such as NumPy, SciPy, and Scikit-Learn can also be used for more advanced data preparation tasks.

Question 4

**Insight 1**

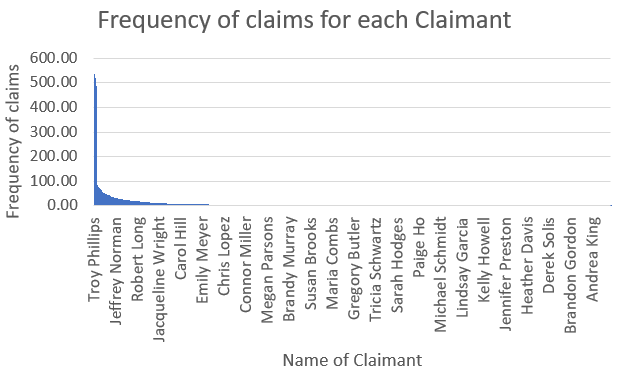


Model 1: Bar chart showing payout amount received by each claimant

With respect to Model 1, the insurance company paid out a maximum of $1,295,137 on their corporate insurance claims to an individual claimant named Hayley Dennis. The company also paid out more than 1 million dollars to 10 other claimants. This suggests that the company might be overpaying on insurance claims and could use this insight to make amendments to their insurance policies to ensure that policy holders receive reasonable payouts based on the terms and conditions, as well as policy premium. This could help to identify over-claims or overly frequent claims that amounts to a large payout being made by claimants.

**Insight 2**

Their biggest client bought insurance from them over 500 times.

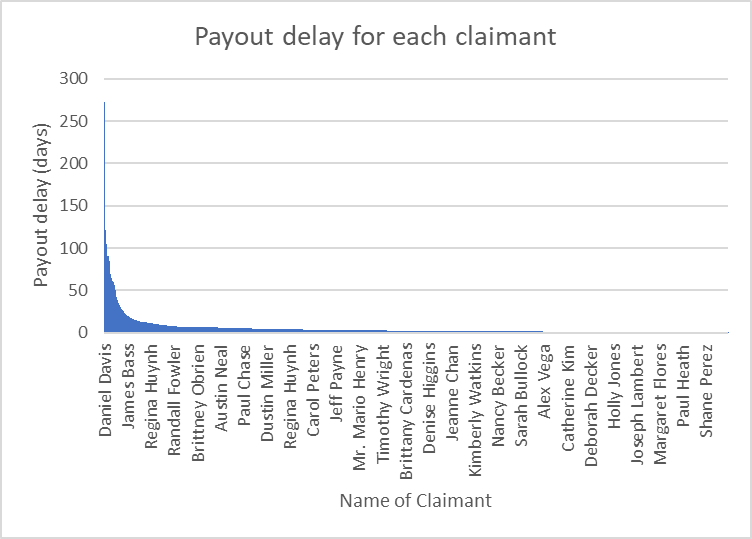


Model 2: Bar chart which shows the frequency of claims of each claimant.

Model 2 shows the frequency of the claims made by each claimant. The individual ‘Troy Philips’ made the greatest number of claims at 535 times over the course of this study. This could be administratively tedious on the company which has to review and process each claim. This could allow the insurance company to identify who are their high-risk customers to devise their business strategy and impose regulations such as policy premiums to deter policy holders from making regular minor claims.

**Insight 3**

Some of the claimants have waited for a long time to receive their payout.

  
Model 3: Bar Graph showing the number of delayed payments per claimant

Model 3 shows the payment delays of the different claimants of the insurance company. From the model, you could tell that the maximum delay is about 272 days from a claimant named “Danial Davis”. The insurance company could use this information above to ensure that their clients’ insurance claims are paid out on time. If there are delays in their payouts, it could affect customers’ confidence in the company and damage their reputation in the long term.

Question 5

**Step 1:**

To perform the linear regression model, we must first clean the data as we did in question 1 and question 2 where the data is prepped for analysis. This will help to clear any anomalies that occur in the dataset such as “blanks, duplicates and unknown values”. This will ensure that the analysis will be accurate in its estimation. By eliminating the blanks in the dataset. We can use its mean to determine the actual date that the claim was paid out.

The Python code to clean the csv dataset is as follows:

# Import necessary libraries

import pandas as pd

import numpy as np

# Read the CSV file into a Pandas dataframe

df = pd.read\_csv(‘ECA.csv’)

# Remove missing values in the "Claim\_ID" column

df.dropna(subset=['Claim\_ID'], inplace=True)

# Remove missing values and invalid values in the "Terms" column

df = df[~df['Terms'].isin(['???', 'Unkn'])]

# Convert the 'Planned' and 'Actual' columns to datetime format using the specified format string

df['Planned'] = pd.to\_datetime(df['Planned'], format='%d/%m/%Y')

df['Actual'] = pd.to\_datetime(df['Actual'], format='%d/%m/%Y %H:%M')

# Calculate the duration between 'Planned' and 'Actual' columns using the absolute value to avoid negative durations

df['Duration'] = (df['Actual'] - df['Planned']).abs()

# Calculate the mean duration between 'Planned' and 'Actual' columns

mean\_duration = df['Duration'].mean()

# Estimate the missing 'Actual' dates using the mean duration

df['Actual'] = np.where(df['Actual'].notnull(), df['Actual'], df['Planned'] + pd.Timedelta(mean\_duration))

# Calculate the estimated 'Actual' dates by adding the duration to the 'Planned' dates

df['Estimated Actual'] = df['Planned'] + df['Duration']

# Convert the durations to days

df['Duration'] = df['Duration'].dt.days

Calculate the IQR for the duration column

Q1 = df['Duration'].quantile(0.25)

Q3 = df['Duration'].quantile(0.75)

IQR = Q3 - Q1

# Calculate the lower and upper bounds for the data

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Remove any outliers from the dataset

df = df[(df['Duration'] > lower\_bound) & (df['Duration'] < upper\_bound)]

**Step 2:**

After data preparation, we can calculate the mode of the claims prior to the recent blank values in order to accurately estimate the delay in the payment. To instruct python to make that estimation, we have to first find out the prior claims of the claimants that have blanks in their values.

The Python code to solve this:

This is a continuation from the above code

from sklearn.linear\_model import LinearRegression

# Sort the data by 'Name' and 'Planned'

df = df.sort\_values(['Name', 'Planned'])

# Calculate the mode of 'Actual' for each claimant based on prior claims

df['Mode\_Actual'] = df.groupby('Name')['Actual'].apply(lambda x: x.fillna(method='ffill').mode().iloc[0])

# Fill in the blank values in 'Actual' with the mode of 'Actual' for each claimant

df['Actual'] = df['Actual'].fillna(df['Mode\_Actual'])

**Step 3:**

Creating the linear regression model based on the values that Python estimated for us in Step 2.

The Python code is as follows:

#import the necessary libraries

import matplotlib.dates as mdates

# Convert the datetime object to matplotlib date format

mdates.date2num(dt)

# Create scatter plot with linear regression line

plt.scatter(df['Duration'], df['Mode\_Actual'])

plt.plot(df['Duration'], LinearRegression().fit(df[['Duration']], df['Mode\_Actual']).predict(df[['Duration']]), color='red')

plt.xlabel('Duration')

plt.ylabel('Mode Actual')

plt.show()

Question 6

The linear regression above depicts the estimation of the delays based on past payout delays of previous claims they encountered with the insurance company. The graph shows a positive relationship between past delays and its frequency of they delay. There is a high possibility of a delay in future claims if the claimant experienced multiple delays regarding their payout in prior claims.

The linear regression equation is as follows:

Y = mX + C

Where Y is the frequency of the delay and X refers to the duration of the delay.

Y =