Data Analyst Associate Practical Exam (Data Camp)

Given a CSV file from a restaurant chain containing food poisoning claims, perform these tasks to help the company's lawyers draw insights on the claims and the time it takes to close each claim by location.

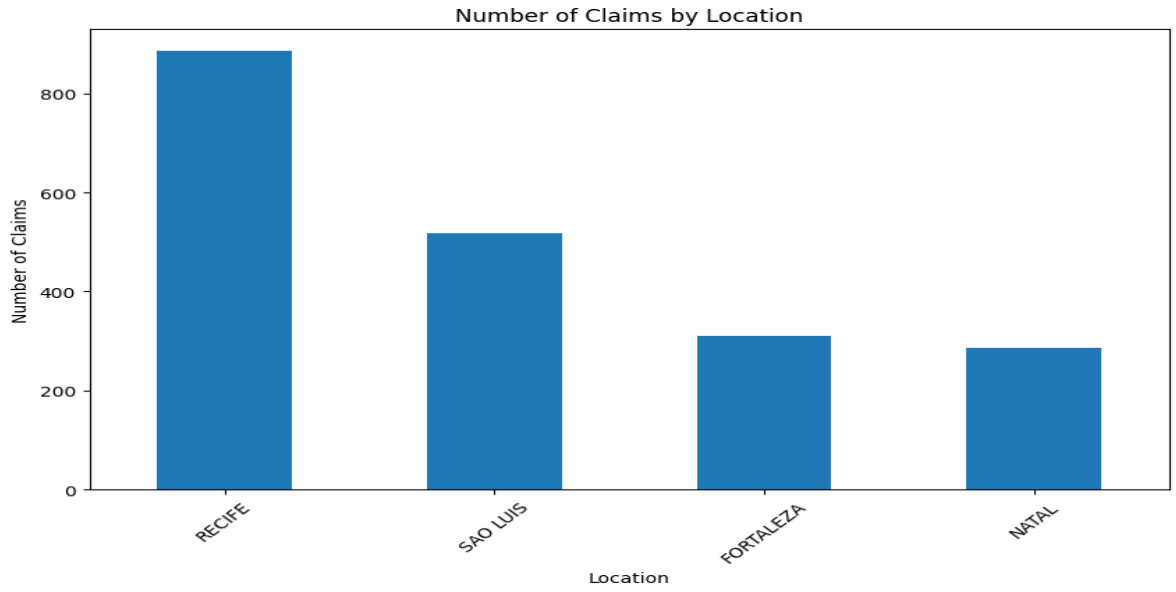
Task 1: clean/verify data

- Values for **claim_id** match the description with 2000 distinct entries, there are no missing values.
- Values for **time_to_close** match the description. There are no negative values. There are no null values.
- Values for **claim_amount** are in the currency of Brazil. The 'R$' has been removed and values were converted from data type 'string' to data type 'float' then rounded to two decimal places to match the description. There are no null values.
- Values for **amount_paid** have been rounded to 2 decimal places, there were 36 counts where the amount_paid is missing. These values have been replaced with the median amount_paid and now match the description.
- Values for **location** match the description. There are no typos. There are no null values.
- Values for **individuals_on_claim** match the description. There are no missing values.
- Values for **linked_cases**** contained 26 counts of missing data. These missing values have been replaced by 'FALSE' and now match the description.
- Values for **cause** must be one of "vegetable", "meat", or "unknown". There were 14 cases where the values were "Meat" and 16 cases where the values were "VEGETABLES". They have been replaced with the appropriate corresponding values and now match the description. There are no null values.

Task 2: are claims balanced across locations?

The number of claims are not balanced across locations. 'RECIFE' has the most claims at 885. 'SAO LUIS' has the second most claims at 517. 'FORTALEZA' has the third most claims at 311. 'NATAL' has the least claims at 287.

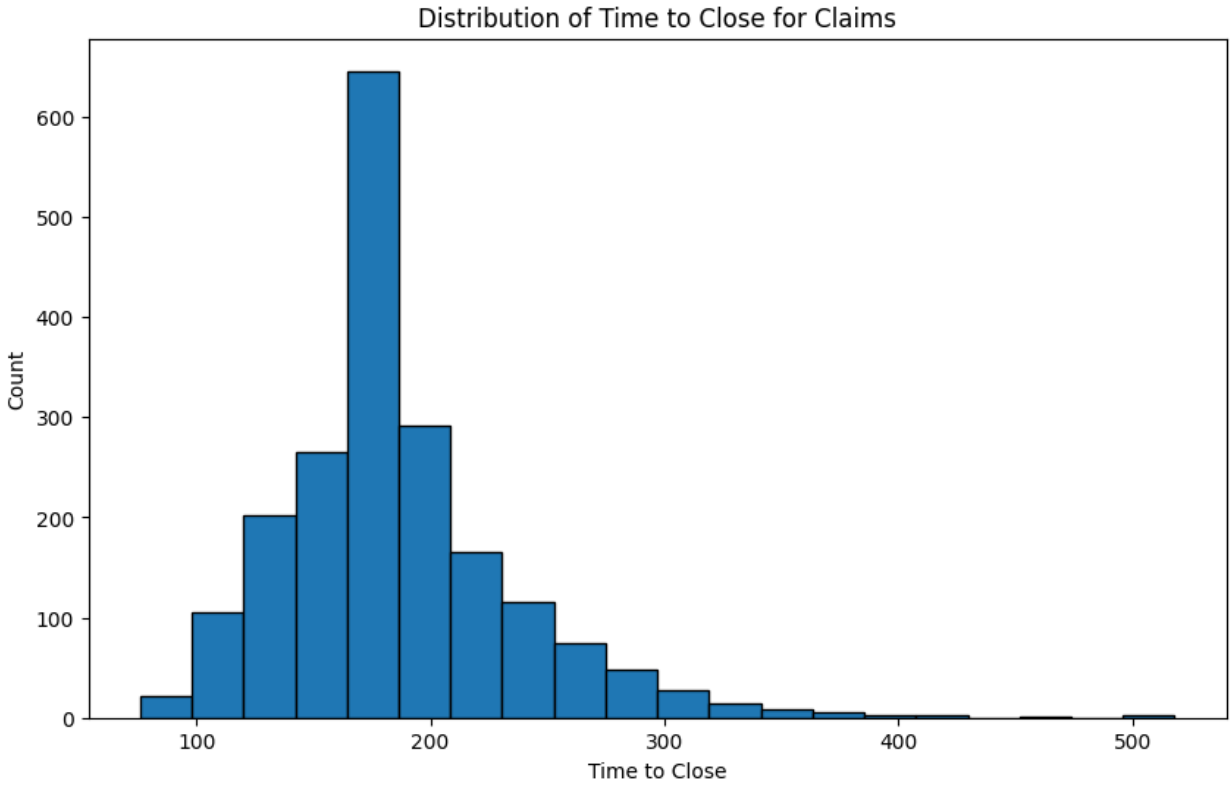
Of the 2000 claims, 47.85% of claims are from meat (957) , 16.5% of claims are from Vegetables (330), and 0.3565% of claims are unknown (713).



Task 3: Explain the distribution of time-to-close for claims

The distribution of time-to-close for all claims maintains a positively skewed distribution with a mean of 185.568, a median of 179.0, and a standard deviation of 49.163.

The majority of all claims get resolved within about 6 months.



Task 4: Describe relationship between time to close and each location.

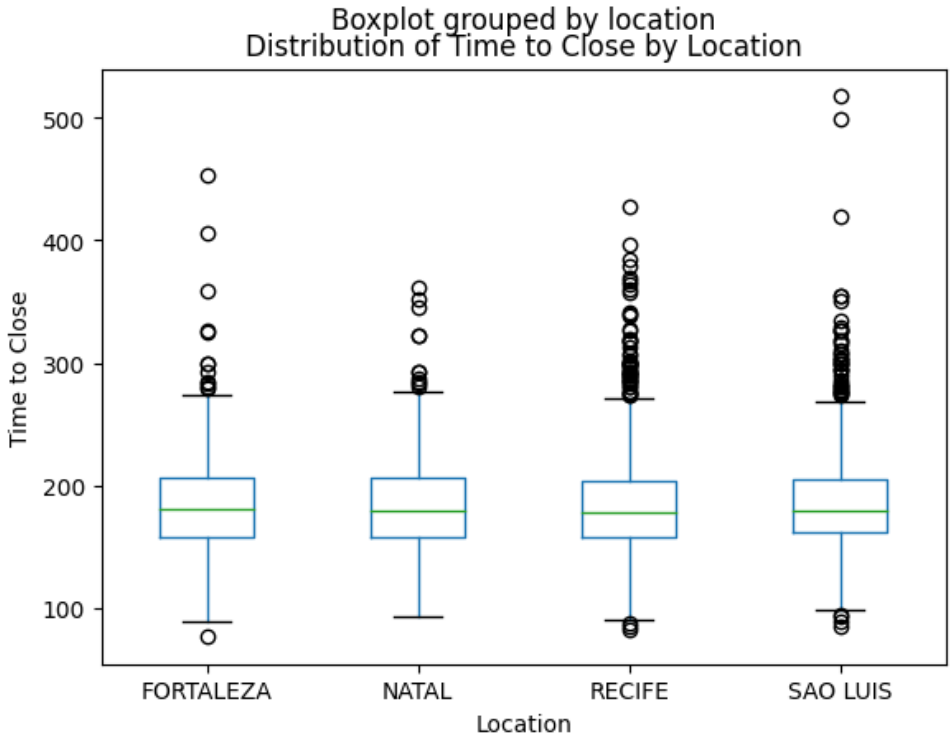
The time-to-close between locations are relatively balanced despite varying number of claims. The overall average time-to-close is 185.56 days, and the average is constant between locations.

For 'FORTALEZA' which holds 15.55% of total claims (311), about 43% of these claims take longer than the overall average time-to-close. It has a minimum time-to-close of 76 days and a maximum of 453 days. The first quartile, median, and third quartiles are 157.0, 180.0 and 205.5 days, respectively.

For 'NATAL' which holds 14.35% of total claims (287), about 38% of these claims take longer than the overall average time-to-close. It has a minimum time-to-close of 93 days and a maximum of 361 days. Natal has the lowest maximum time-to-close. The first quartile, median, and third quartiles are 157.0, 179.0 and 205.5 days, respectively.

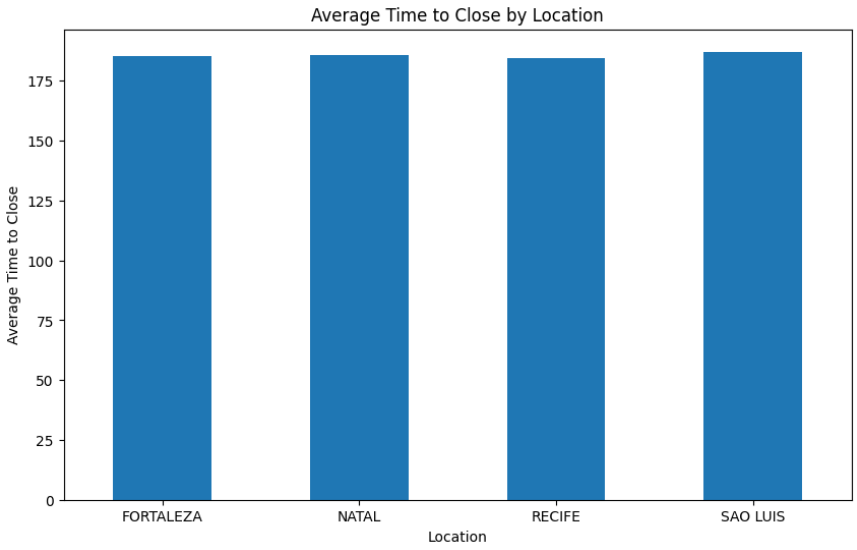
For 'RECIFE' which holds 44.25% of total claims (885), about 38% of these claims take longer than the overall average time-to-close. It has a minimum time-to-close of 82 days and a maximum of 427 days. The first quartile, median, and third quartiles are 157.0, 178.0 and 203.0 days, respectively.

For 'SAO LUIS' which holds 25.85% of total claims (517), about 39% of these claims take longer than the overall average time-to-close. It has a minimum time-to-close of 84 days and a maximum of 518 days. Sao Luis has the highest maximum time-to-close. The first quartile, median, and third quartiles are 161.0, 179.0 and 205.0 days, respectively.



The average time-to-close is roughly the same for each location:

RECIFE = 184.61 days, FORTALEZA = 185.31 days, NATAL = 185.93 days, SAO LUIS = 187.17 days



Python Code:

```
import pandas as pd
import matplotlib.pyplot as plt
def replace_missing_with_median(column):
    median = column.median()
    column.fillna(median, inplace=True)
    return column
def remove_missing_values(column):
    column.dropna(inplace=True)
    return column
def replace_missing_with_zero(column):
    column.fillna(0, inplace=True)
    return column
def replace_missing_with_false(column):
    column.fillna(False, inplace=True)
    return column
def replace_missing_with_unknown(column):
    column.fillna('unknown', inplace=True)
    return column
def data_quality_check(df):
    columns = []
    missing_values = []
    for column in df.columns:
      columns.append(column)
      missing_values.append(df[column].isnull().sum())
    dicta = {'Columns':columns,'Missing_Values':missing_values}
    missing_df = pd.DataFrame(dicta)
    print(missing_df)
```

```
    df['time_to_close'] = replace_missing_with_median(df['time_to_close'])
    #df['claim_amount'] = replace_missing_with_median(df['claim_amount'])
    df['amount_paid'] = replace_missing_with_median(df['amount_paid'])
    df['location'] = remove_missing_values(df['location'])
    df['individuals_on_claim'] = replace_missing_with_zero(df['individuals_on_claim'])
    df['linked_cases'] = replace_missing_with_false(df['linked_cases'])
       #desired_lovs = ['unknown', 'meat', 'vegetable']
    df.loc[df['cause'] == 'VEGETABLE', 'cause'] = 'vegetable'
    df.loc[df['cause'] == 'MEAT', 'cause'] = 'meat'
    # Filter the DataFrame to include only the desired LOVs
    df['cause'] = replace_missing_with_unknown(df['cause'])
    #df = df[df['cause'].isin(desired_lovs)]
    return df
def visualize_claims_by_location(df):
    # Read the CSV file
    df = pd.read_csv(file_path)
    # Count the number of claims in each location
    claims_by_location = df['location'].value_counts()
    # Create the bar chart
    plt.figure(figsize=(10, 6))
    claims_by_location.plot(kind='bar')
    plt.title('Number of Claims by Location')
    plt.xlabel('Location')
    plt.ylabel('Number of Claims')
    plt.xticks(rotation=45)
    plt.show()
def visualize_time_to_complete(df):
  print(df.columns)
  time_to_close = df['time_to_close']
```

```
  # Calculate statistical measures
  mean = time_to_close.mean()
  median = time_to_close.median()
  std_dev = time_to_close.std()
  # Print the statistical measures
  print("Statistical Measures for Time to Close:")
  print(f"Mean: {mean}")
  print(f"Median: {median}")
  print(f"Standard Deviation: {std_dev}")
  # Create a histogram of the time to close
  plt.figure(figsize=(10, 6))
  plt.hist(time_to_close, bins=20, edgecolor='black')
  plt.title('Distribution of Time to Close for Claims')
  plt.xlabel('Time to Close')
  plt.ylabel('Count')
  plt.show()
def analyze_time_to_close_by_location(df):
    # Read the CSV file
    # Group the data by location
    grouped_data = df.groupby('location')
    # Calculate statistics for each location
    statistics_by_location = grouped_data['time_to_close'].describe()
    # Print the statistics for each location
    print("Statistics for Time to Close by Location:")
    print(statistics_by_location)
    # Create boxplots to visualize the distribution of time to close by location
    plt.figure(figsize=(10, 6))
    df.boxplot(column='time_to_close', by='location', grid=False)
    plt.title('Distribution of Time to Close by Location')
```

```
    plt.xlabel('Location')
    plt.ylabel('Time to Close')
    plt.show()
file_path = "food_claims_2212.csv"
df = pd.read_csv(file_path)
new_df = data_quality_check(df)
visualize_claims_by_location(new_df)
visualize_time_to_complete(new_df)
analyze_time_to_close_by_location(new_df)
avg_time_to_close = df.groupby('location')['time_to_close'].mean()
# Create a bar chart to visualize the average time to close by location
plt.figure(figsize=(10, 6))
avg_time_to_close.plot(kind='bar')
plt.title('Average Time to Close by Location')
plt.xlabel('Location')
plt.ylabel('Average Time to Close')
plt.xticks(rotation=0)
plt.show()
```

SQL Code:

```
select location, (percentile_cont(0.25) within group (order by time_to_close)) as q1,
(percentile_cont(0.50) within group (order by time_to_close)) as median,
(percentile_cont(0.75) within group (order by time_to_close)) as q3
from food_claims_2212.csv
group by location;
select DISTINCT cause, count(cause)
from food_claims_2212.csv
group by 1 order by 2;
```

By Vincent Forca

July 2023

- End of Submission -