# Data Cleaning and Analysis of food claims dataset

Taha Nasir

2023-03-21

## Task 1

Based on the table description and the actual structure of the 'data' dataset, here is an analysis of whether each column values match the description and what code can be used in R to match them to the description:

*1.claim_id:* The column values match the description of being nominal with no missing values, as the values are integers and there are no missing values. No code is needed to match the description.

```
library(readxl)
data<-read.csv("food_claims_2212.csv")
str(data)
```

```
## 'data.frame':    2000 obs. of  8 variables:
##  $ claim_id            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ time_to_close       : int  317 195 183 186 138 183 190 183 149 149 ...
##  $ claim_amount        : chr  "R$ 74474.55" "R$ 52137.83" "R$ 24447.2" "R$
29006.28" ...
##  $ amount_paid         : num  51231 42111 23986 27943 16251 ...
##  $ location            : chr  "RECIFE" "FORTALEZA" "SAO LUIS" "FORTALEZA"
...
##  $ individuals_on_claim: int  15 12 10 11 11 11 12 8 9 6 ...
##  $ linked_cases        : logi  FALSE TRUE TRUE FALSE FALSE FALSE ...
##  $ cause               : chr  "unknown" "unknown" "meat" "meat" ...
```

```
#number of missing values=0
sum(is.na(data$claim_id))
```

```
## [1] 0
```

*2.time_to_close:* The column values doesnot match the description of being discrete with positive values, as some of the values doesnot match the description after checking with the 'is.integer' and 'all' functions.In order to make it right, the values are made discrete and positive and then missing values are replaced with the median of time_to_close by using a code in RStudio.

```
#The number of missing values=0
sum(is.na(data$time_to_close))
```

```
## [1] 0
```

```
#Replace missing values of time to close
is_discrete_positive <- is.integer(data$time_to_close) & all(data$time_to_clo
se > 0)
is_discrete_positive

## [1] TRUE

if(!is_discrete_positive){
  # Convert the values to discrete and positive
  data$time_to_close  <- as.integer(round(abs(data$time_to_close )))
}
#Check again and convert to median
is_discrete_positive2 <- is.integer(data$time_to_close) & all(data$time_to_cl
ose > 0)
is_discrete_positive2

## [1] TRUE

data$time_to_close[is.na(data$time_to_close)] <- median(data$time_to_close, n
a.rm = TRUE)
```

*3.claim_amount:* The column values do not match the description of being continuous, as the values are stored as character strings representing currency. To match the description, the values would need to be converted to numeric values. Additionally, missing values are not addressed in the actual structure of the data set. To replace missing values with the overall median claim amount and convert the column to a continuous variable, coding In RStuio has been done.

```
sum(is.na(data$claim_amount))

## [1] 0

data$claim_amount <- as.numeric(gsub("R\\$ ", "", data$claim_amount)) # conve
rt to numeric
data$claim_amount[is.na(data$claim_amount)] <- median(data$claim_amount, na.r
m = TRUE) # replace missing values with median
```

*4.amount_paid:* The column values partly match the description of being continuous, as the values are stored as numeric values representing currency and are rounded to 2 decimal places. However, there are 36 missing values as checked by the is.na function. To replace missing values with the overall median amount paid, you can use the following code:

```
sum(is.na(data$amount_paid))#Missing values=36

## [1] 36

data$amount_paid[is.na(data$amount_paid)] <- median(data$amount_paid, na.rm =
TRUE)
```

*5.location:*The column values match the description of being nominal with no missing values, as the values are character strings representing locations and there are no missing

values. No code is needed to match the description.If there were missing values, 'na.omit' would have been used.

```
sum(is.na(data$location))
```

*6.individuals_on_claim:* The column values match the description of being discrete with no missing values, as the values are integers representing the number of individuals on a claim and there are no missing values. If there were missing values then we would have used the following code to replace missing values with 0:

```
sum(is.na(data$individuals_on_claim))

## [1] 0

data$individuals_on_claim[is.na(data$individuals_on_claim)] <- 0
```

*7.linked_cases:* The column values partly matches the description as the values are nominal but the column has 26 missing values as checked in R. The values are logical values representing whether a claim is linked to other cases. However, missing values are not addressed in the actual structure of the dataset. To replace missing values with FALSE, R code has been used in RStudio

```
sum(is.na(data$linked_cases))#Missing values=26

## [1] 26

subset(data,is.na(data$linked_cases))

##      claim_id time_to_close claim_amount amount_paid location
## 130       130           108      5122.45     3864.18   RECIFE
## 249       249           193     21638.91    15486.70 SAO LUIS
## 264       264           193     30726.17    24351.11 SAO LUIS
## 283       283           277     32041.49    30510.69 SAO LUIS
## 290       290           219     34513.08    25671.63 SAO LUIS
## 334       334           134      9645.07     8324.35    NATAL
## 372       372           190     31776.64    30708.88   RECIFE
## 599       599           132      5366.81     4589.93   RECIFE
## 661       661           271     49627.81    42277.28    NATAL
## 675       675           162     21200.69    19195.59   RECIFE
## 769       769           162     25937.29    21086.13   RECIFE
## 830       830           184     38979.78    27706.51 SAO LUIS
## 860       860           191     33956.63    29073.06   RECIFE
## 920       920           176     10211.68     9210.47 SAO LUIS
## 921       921           318     53742.41    49632.63   RECIFE
## 1119     1119           147     29733.89    20694.71 SAO LUIS
## 1124     1124           190     25387.40    17218.52    NATAL
## 1151     1151           153     28211.95    24154.78   RECIFE
## 1229     1229           232     48431.25    34641.08   RECIFE
## 1270     1270           153      7101.10     5668.75   RECIFE
## 1476     1476           135     24961.92    20247.56   RECIFE
## 1623     1623           168     24182.11    19174.03   RECIFE
```

```
## 1691     1691           188     35479.82    34584.48   RECIPE
## 1789     1789           202     25434.76    19631.02    NATAL
## 1834     1834           172      7509.08     7416.49 SAO LUIS
## 1979     1979           217     31008.59    22398.82 SAO LUIS
##      individuals_on_claim linked_cases     cause
## 130                     2           NA      meat
## 249                    10           NA vegetable
## 264                     7           NA   unknown
## 283                    12           NA      meat
## 290                    10           NA      meat
## 334                     2           NA   unknown
## 372                    12           NA      meat
## 599                     1           NA   unknown
## 661                    12           NA   unknown
## 675                    13           NA vegetable
## 769                     6           NA   unknown
## 830                    11           NA      meat
## 860                     8           NA   unknown
## 920                     4           NA      meat
## 921                    14           NA   unknown
## 1119                    6           NA   unknown
## 1124                    7           NA      meat
## 1151                   10           NA      meat
## 1229                   14           NA      meat
## 1270                    2           NA   unknown
## 1476                    6           NA   unknown
## 1623                   13           NA vegetable
## 1691                   14           NA      meat
## 1789                   13           NA vegetable
## 1834                    2           NA   unknown
## 1979                    6           NA   unknown

data$linked_cases[is.na(data$linked_cases)] <- FALSE
```

*8.cause:*The column values match the description of being nominal with no missing values, as the values are character strings representing the cause of the food poisoning and there are no missing values.There is no use of replacing the missing values with 'unknown' since there are no missing values in this column.

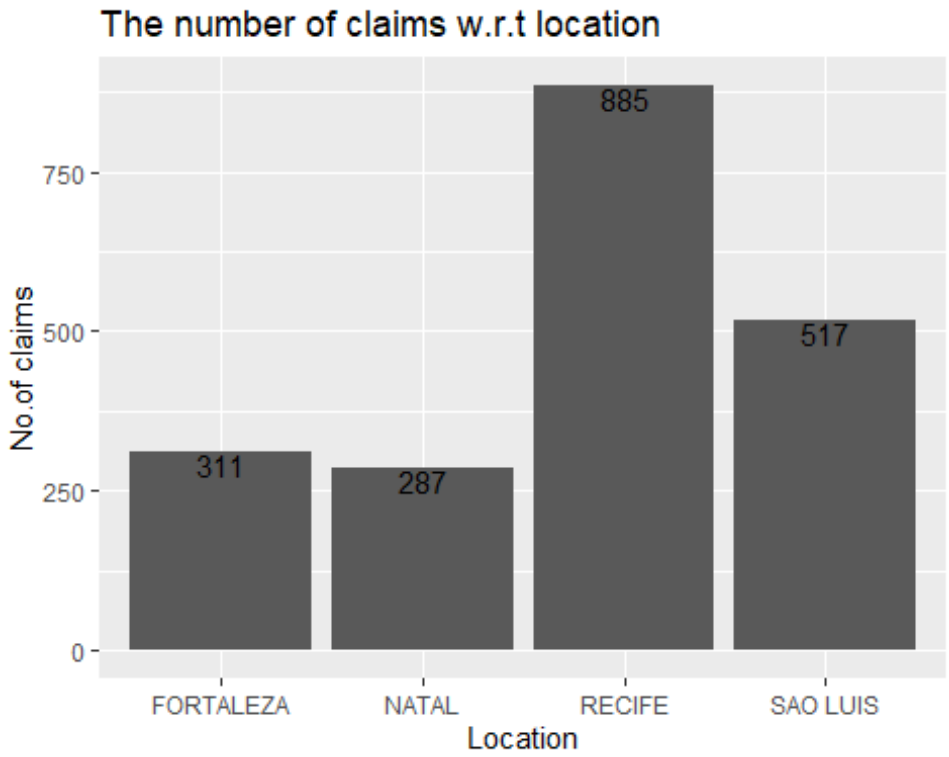
```
sum(is.na(data$cause)) #No missing value

## [1] 0
```

#Task 2: Create a visualization that shows the number of claims in each location. Use the visualization to: a. State which category of the variable location has the most observations b. Explain whether the observations are balanced across categories of the variable location

```
library(ggplot2)
ggplot(data, aes(x=location))+
  geom_bar()+
  geom_text(stat = "count", aes(label = after_stat(count), vjust = 1)) +
```

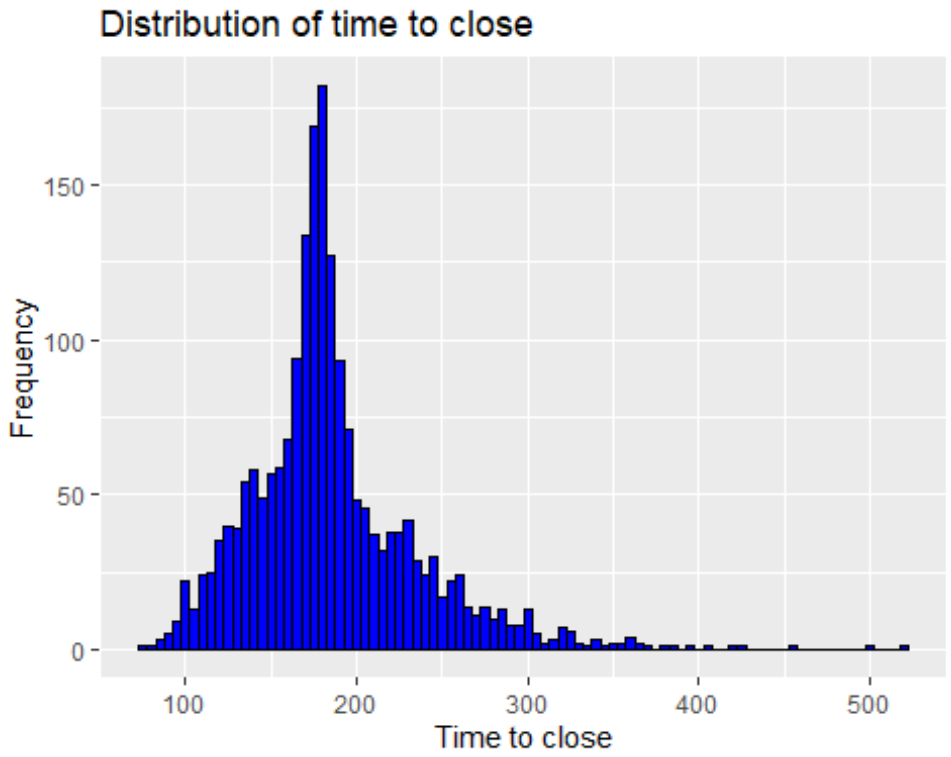
```
  labs(title="The number of claims w.r.t location", x='Location', y='No.of cl
aims')
```




The location with the most no.of claims turn out to be RECIFE with 885 claims. The one with the second most no.of claims is SAO LUIS with 517 claims and the location with the lowest no.of claims is NATAL with 287 claims. Since the heights of the bars are different, hence the observations are not balanced.

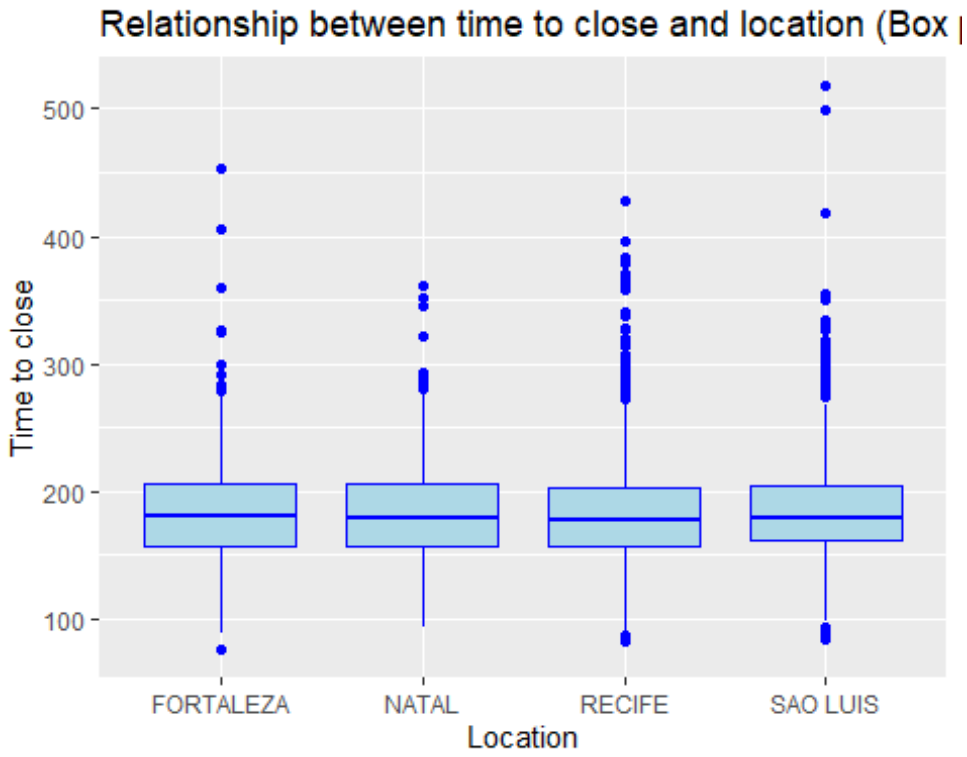
Describe the distribution of time to close for all claims. Your answer must include a visualization that shows the distribution.

```
# Create a histogram showing the distribution of time to close
ggplot(data, aes(x = time_to_close)) +
  geom_histogram(binwidth = 5, color = "black", fill = "blue") +
  labs(x = "Time to close", y = "Frequency", title = "Distribution of time to
close")
```

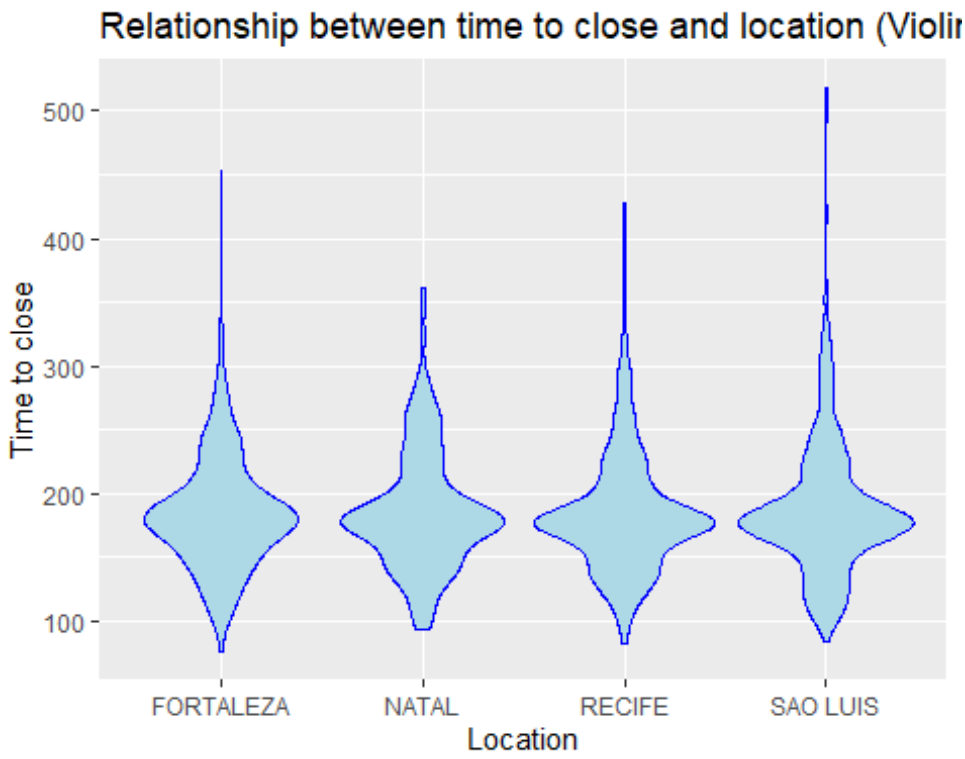



Describe the relationship between time to close and location. Your answer must include a visualization to demonstrate the relationship.

```
# Create a box plot showing the relationship between time to close and locati
on
ggplot(data, aes(x = location, y = time_to_close)) +
  geom_boxplot(color = "blue", fill = "lightblue") +
  labs(x = "Location", y = "Time to close", title = "Relationship between tim
e to close and location (Box plot)")
```




```
# Create a violin plot showing the relationship between time to close and loc
ation
ggplot(data, aes(x = location, y = time_to_close)) +
  geom_violin(color = "blue", fill = "lightblue") +
  labs(x = "Location", y = "Time to close", title = "Relationship between tim
e to close and location (Violin plot)")
```


Relationship between time to close and location (Violi
Time to close
500
400
300
200
100
FORTALEZA
NATAL
RECIFE
SAO LUIS
Location