PES University, Bangalore Established under the Karnataka Act No. 16 of 2013

UE21CS342AA2 - Data Analytics

Worksheet 1b : Correlation Analysis

Richa Shahi - shahiricha2412@gmail.com , Abhay K Iyengar – abzee2002@gmail.com

Correlation

Correlation is a measure of the strength and direction of linear relationship between two random variables in other words it is a measure of the association between two variables. This association is measured using the correlation coefficients. Correlation coefficients help us choose features in model building and leverage the association relationship between two variables.

There are different types of correlation coefficients, based on the nature of the data being compared:

- Between two continuous (interval, ratio) random variables - Pearson's Product Moment Correlation Coefficient
- Between two ordinal random variables - Spearman-Rank Correlation Coefficient
- Between a continuous RV and a dichotomous RV - Point Bi-Serial Correlation Coefficient
- Between two binary random variables - Phi Coefficient

Road Accidents

India is the world's second-most populous country with a population of around 1.2 billion people (as of July 2022). Roads are a very important mode of transport in India, spanning over 6.2 million kilometers of length, making it the country with the second-largest road network, after the United States of America. (Source: Wikipedia).

With India trying to modernize its road infrastructure, there is still the problem of frequent road accidents. Road accidents in India is a major cause of death and injury. The NCRB (National Crime Records Bureau) of India collects detailed data on traffic accidents and collisions annually. Please use the dataset provided for analysis that contains road accident data in India from 2016. You can download the dataset from here.

Data Dictionary

S. No.: Serial number State/ UT: name of state/union terrirory in India

Fine/Clear - Total Accidents: total accidents per state/UT in Fine/Clear weather conditions

Fine/Clear - Persons Killed: total fatalities per state/UT in Fine/Clear weather conditions

Fine/Clear - Persons Injured: total injured people per state/UT in Fine/Clear weather conditions

Mist/Foggy - Total Accidents: total accidents per state/UT in Mist/Foggy weather conditions

Mist/ Foggy - Persons Killed: total fatalities perstate/UT in Mist/Foggy weather conditions

Mist/ Foggy - Persons Injured:total injured people per state/UT in Mist/Foggy weather conditions

Cloudy - Total Accidents: total accidents per state/UT in Cloudy weather conditions

Cloudy - Persons Killed: total fatalities per state/UT inCloudy weather conditions

Cloudy - Persons Injured: total injured people per state/UT in Cloudy weather conditions

Rainy - Total Accidents: total accidents per state/UT in Rainy weather conditions

Rainy - Persons Killed: total fatalities per state/UT in Rainy weather conditions

Rainy - Persons Injured: total injured people per state/UT in Rainy weather conditions

Snowfall - Total Accidents: total accidents per state/UT in Snowfall weather conditions

Snowfall - Persons Killed: totalfatalities per state/UT in Snowfall weather conditions

Snowfall - Persons Injured: total injured people per state/UT in Snowfall weather conditions

Hail/Sleet - Total Accidents: total accidents per state/UT in Hail/Sleet weather conditions

Hail/Sleet - Persons Killed: total fatalities per state/UT in Hail/Sleet weather conditions

Hail/Sleet - Persons Injured: total injured people per state/UT in Hail/Sleet weather conditions

Dust Storm - Total Accidents: total accidents per state/UT in Dust Storm weather conditions

Dust Storm - Persons Killed: total fatalities per state/UT in Dust Storm weather conditions

Dust Storm - Persons Injured: total injured people per state/UT in Dust Storm weather conditions

Others - Total Accidents: total accidents per state/UT in Other weather conditions

Others - Persons Killed: total fatalities perstate/UT in Other weather conditions

Others - Persons Injured: total injured people per state/UT in Other weather conditions

Problems

Problem 1

Find the total number of accidents in each state for the year 2016 and display your results. Make sure to display all rows while printing the dataframe. Print only the necessary columns. (Hint: use the grep command to help filter out column names).

```
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df <- read.csv('road_accidents_india_2016.csv', row.names=1)
```

```
acc_cols <- grep("Total.Accidents$", colnames(df), ignore.case=T, value=TRUE)
total_accidents <- data.frame(state..ut=df$State..UT,
total_acc=rowSums(df[ , c(acc_cols)], na.rm=TRUE))
print.data.frame(total_accidents)
```

```
##            state..ut total_acc
## 0     Andhra Pradesh     24888
## 1  Arunachal Pradesh       249
## 2              Assam      7435
## 3              Bihar      8222
## 4       Chhattisgarh     13580
## 5                Goa      4304
## 6            Gujarat     21859
## 7            Haryana     11234
## 8   Himachal Pradesh      3168
## 9    Jammu & Kashmir      5501
## 10         Jharkhand      4932
## 11         Karnataka     44403
## 12            Kerala     39420
## 13    Madhya Pradesh     53972
## 14       Maharashtra     39878
## 15           Manipur       538
## 16         Meghalaya       620
## 17           Mizoram        83
## 18          Nagaland        75
## 19            Orissa     10532
## 20            Punjab      6952
## 21         Rajasthan     23066
## 22            Sikkim       210
## 23        Tamil Nadu     71431
## 24         Telangana     22811
## 25           Tripura       557
## 26       Uttarakhand      1591
## 27     Uttar Pradesh     35612
## 28       West Bengal     13580
## 29     A & N Islands       238
## 30        Chandigarh       428
## 31      D & N Haveli        70
## 32       Daman & Diu        71
```

```
## 33             Delhi      7375
## 34       Lakshadweep         1
## 35        Puducherry      1766
```

Problem 2

Find the (fatality rate = $\frac{\text{total number of deaths}}{\text{total number of accidents}}$) in each state. Find out if there is a significant linear correlation at a significance of $\alpha = 0.05$ between the fatality rate of a state and the mist/foggy rate (fraction of total accidents that happen in mist/foggy conditions).

Plot the fatality rate against the mist/foggy rate. (Hint: use the ggscatter library to plot a scatterplot with the confidence interval of the correlation coefficient).

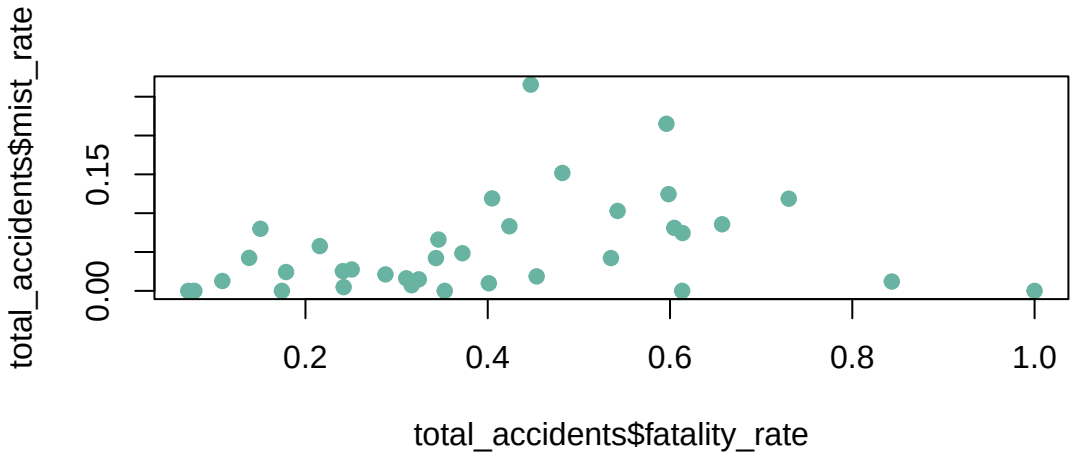
```
death_cols <- grep("Persons.Killed$", colnames(df), ignore.case=T, value=TRUE)
total_accidents$total_deaths <- rowSums(df[ , c(death_cols)])
total_accidents$fatality_rate <- total_accidents$total_deaths/total_accidents$total_acc
total_accidents$mist_rate <- df$Mist..Foggy...Total.Accidents/total_accidents$total_acc
print.data.frame(total_accidents)
```

```
##            state..ut total_acc total_deaths fatality_rate   mist_rate
## 0     Andhra Pradesh     24888         8541    0.34317743 0.042229187
## 1  Arunachal Pradesh       249          149    0.59839357 0.124497992
## 2              Assam      7435         2572    0.34593141 0.066039005
## 3              Bihar      8222         4901    0.59608368 0.215154464
## 4       Chhattisgarh     13580         3908    0.28777614 0.021207658
## 5                Goa      4304          336    0.07806691 0.000000000
## 6            Gujarat     21859         8136    0.37220367 0.048446864
## 7            Haryana     11234         5024    0.44721382 0.265533203
## 8   Himachal Pradesh      3168         1271    0.40119949 0.009785354
## 9    Jammu & Kashmir      5501          958    0.17415015 0.000000000
## 10         Jharkhand      4932         3027    0.61374696 0.074412003
## 11         Karnataka     44403        11133    0.25072630 0.027520663
## 12            Kerala     39420         4287    0.10875190 0.012683917
## 13    Madhya Pradesh     53972         9646    0.17872230 0.024216260
## 14       Maharashtra     39878        12935    0.32436431 0.014820202
## 15           Manipur       538           81    0.15055762 0.079925651
## 16         Meghalaya       620          150    0.24193548 0.004838710
## 17           Mizoram        83           70    0.84337349 0.012048193
## 18          Nagaland        75           46    0.61333333 0.000000000
## 19            Orissa     10532         4463    0.42375617 0.083175085
## 20            Punjab      6952         5077    0.73029344 0.118670886
## 21         Rajasthan     23066        10465    0.45369808 0.018642157
## 22            Sikkim       210           85    0.40476190 0.119047619
## 23        Tamil Nadu     71431        17218    0.24104380 0.025353138
## 24         Telangana     22811         7219    0.31647012 0.007233352
## 25           Tripura       557          173    0.31059246 0.016157989
## 26       Uttarakhand      1591          962    0.60465116 0.081081081
## 27     Uttar Pradesh     35612        19320    0.54251376 0.102886667
## 28       West Bengal     13580         6544    0.48188513 0.151767305
## 29     A & N Islands       238           17    0.07142857 0.000000000
## 30        Chandigarh       428          151    0.35280374 0.000000000
## 31      D & N Haveli        70           46    0.65714286 0.085714286
## 32       Daman & Diu        71           38    0.53521127 0.042253521
```

```
## 33             Delhi      7375         1591    0.21572881 0.057627119
## 34       Lakshadweep         1            1    1.00000000 0.000000000
## 35        Puducherry      1766          244    0.13816535 0.042468856
```

Plot the fatality rate and mist/foggy rate (see this and this for R plot customization).

```
plot(x=total_accidents$fatality_rate, y=total_accidents$mist_rate,
col='#69b3a2', pch=19)
```



Correlation between two continuous RVs: Pearson's correlation coefficient. Pearson's correlation coefficient between two RVs x and y is given by:

$$\rho = \frac{\text{Covariance}(x, y)}{\sigma_x \cdot \sigma_y}$$

where:

ρ represents the Pearson's correlation coefficient

Covariance(x, y) is the covariance between x and y

σ_x is the standard deviation of x

σ_y is the standard deviation of y.

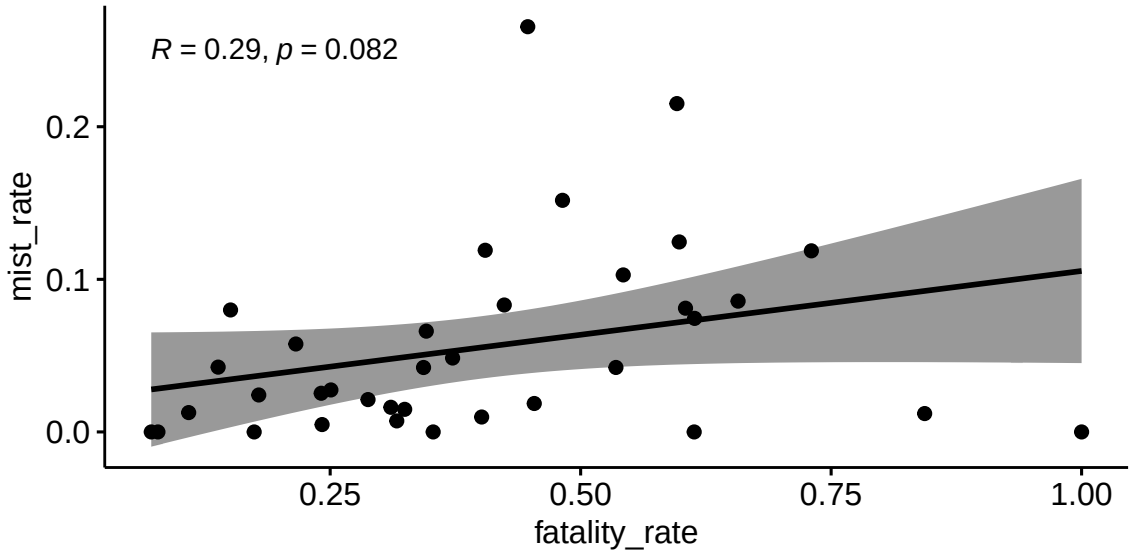
```
rho <- cor(total_accidents$fatality_rate,
total_accidents$mist_rate, method='pearson')
rho
```

```
## [1] 0.2935159
```

```
corr_test = cor.test(total_accidents$fatality_rate,
total_accidents$mist_rate, method='pearson')
print(corr_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  total_accidents$fatality_rate and total_accidents$mist_rate
## t = 1.7903, df = 34, p-value = 0.08231
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03875722  0.56734253
## sample estimates:
##       cor
## 0.2935159
```

```
ggscatter(total_accidents, x='fatality_rate', y='mist_rate',
add='reg.line', conf.int=TRUE,
cor.coef=TRUE, cor.method = 'pearson')
```



Since the p-value of $0.07693 > 0.05$ (the correlation coefficient lies within the 95% confidence interval), there is no statistically significant correlation between the fatality rate and the mist rate.

Problem 3

Rank the states based on total accidents and total fatalities (give a rank of 1 to the state that has the highest value of a property). You are free to use any tie-breaking method for assigning ranks.

Find the Spearman-Rank correlation coefficient between the two rank columns and determine if there is any statistical significance at a significance level of $\alpha = 0.05$. Also test the hypothesis that the correlation coefficient is at least 0.2.

```
total_accidents$acc_ranks <- rank(desc(total_accidents$total_acc),
ties.method='random')
total_accidents$death_ranks <- rank(desc(total_accidents$total_deaths),
ties.method='random')
```

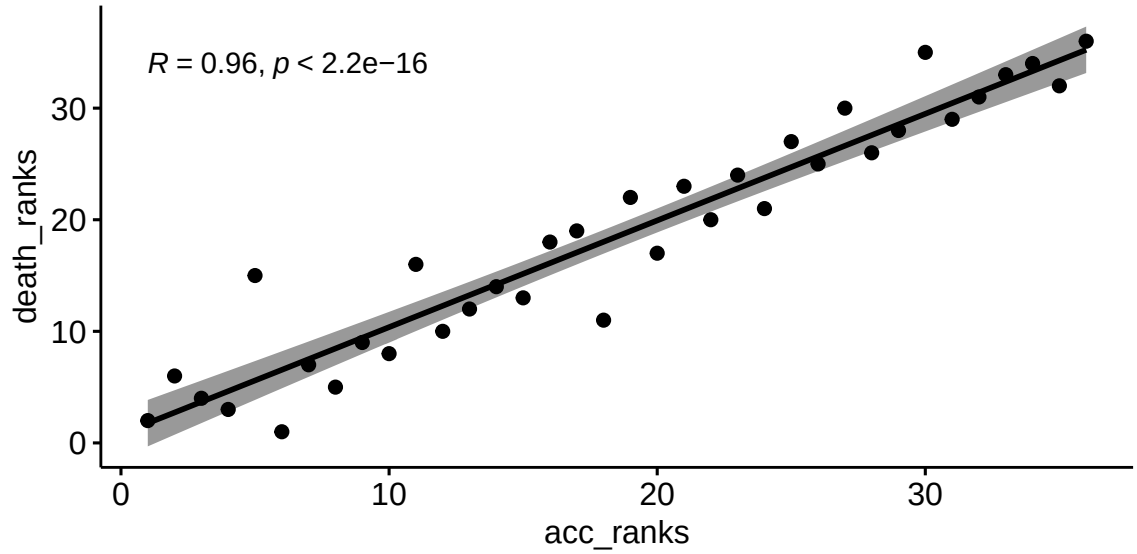
```
rs <- cor(total_accidents$acc_ranks, total_accidents$death_ranks,
method='spearman')
rs
```

```
## [1] 0.9559846
```

```
print(cor.test(total_accidents$acc_ranks,
total_accidents$death_ranks, method='spearman'))
```

```
##
##  Spearman's rank correlation rho
##
## data:  total_accidents$acc_ranks and total_accidents$death_ranks
## S = 342, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.9559846
```

```
ggscatter(total_accidents, x='acc_ranks', y='death_ranks',
add='reg.line', conf.int=TRUE,
cor.coef=TRUE, cor.method = 'spearman')
```



Check if correlation coefficient is at least 0.2. The t statistic is given by

$$t = \frac{r_s - \rho_s}{\sqrt{\frac{1-r_s^2}{n-2}}}$$

where:

- t represents the t-statistic,
- r_s is the Spearman-Rank correlation coefficient,

- ρ_s value of the population correlation coefficient being tested against,
- n is the number of data points in the sample.

```
degrees <- nrow(total_accidents) - 2
t_stat <- (rs - 0.2)/sqrt((1 - rs*rs)/(nrow(total_accidents) - 2))
t_stat
```

```
## [1] 15.02336
```

Two-tailed test - p-value

```
2 * pt(q=t_stat, df=degrees, lower.tail=FALSE)
```

```
## [1] 1.415e-16
```

Problem 4

Convert the column Hail.Sleet... Total.Accidents to a binary column as follows. If a hail/sleet accident has occurred in a state, give that state a value of 1. Otherwise, give it a value of 0. Once converted, find out if there is a significant correlation between the hail_accident_occcur binary column created and the number of rainy total accidents for every state.

Calculate the point bi-serial correlation coefficient between the two columns. (Hint: it is equivalent to calculating the Pearson correlation between a continuous and a dichotomous variable. You could also use the ltm package's biserial.cor function).

```
total_accidents$hail_binary <- ifelse(df$Hail.Sleet...Total.Accidents > 0, 1, 0)
total_accidents$rain_acc <- df$Rainy...Total.Accidents
print.data.frame(total_accidents[, c('state..ut', 'hail_binary', 'rain_acc')])
```

```
##            state..ut hail_binary rain_acc
## 0     Andhra Pradesh           1     1456
## 1  Arunachal Pradesh           1       30
## 2              Assam           1      528
## 3              Bihar           0      939
## 4       Chhattisgarh           0     1279
## 5                Goa           0      529
## 6            Gujarat           1      759
## 7            Haryana           1     1656
## 8   Himachal Pradesh           0      136
## 9    Jammu & Kashmir           0       77
## 10         Jharkhand           1      859
## 11         Karnataka           1     3475
## 12            Kerala           0     6902
## 13    Madhya Pradesh           1     3931
## 14       Maharashtra           0     1958
## 15           Manipur           1       81
## 16         Meghalaya           0       64
## 17           Mizoram           0        0
## 18          Nagaland           0        0
## 19            Orissa           1     1637
```

```
## 20            Punjab           0      402
## 21         Rajasthan           1      475
## 22            Sikkim           1       19
## 23        Tamil Nadu           0     2893
## 24         Telangana           0      237
## 25           Tripura           0       30
## 26       Uttarakhand           1       75
## 27     Uttar Pradesh           1     3168
## 28       West Bengal           1     2267
## 29     A & N Islands           0       63
## 30        Chandigarh           0        0
## 31      D & N Haveli           0       15
## 32       Daman & Diu           0        7
## 33             Delhi           1      449
## 34       Lakshadweep           0        0
## 35        Puducherry           1      198
```

```
cor.test(total_accidents$rain_acc,
total_accidents$hail_binary, method='pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  total_accidents$rain_acc and total_accidents$hail_binary
## t = 0.84232, df = 34, p-value = 0.4055
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1947090  0.4503544
## sample estimates:
##       cor
## 0.1429725
```

There is no significant correlation.

Problem 5

Similar to in Problem 4, create a binary column to represent whether a dust storm accident has occurred in a state (1 = occurred, 0 = not occurred). Convert the two columns into a contingency table. Calculate the phi coefficient of the two tables. (Hint: use the psych package).

```
total_accidents$dust_binary <- ifelse(df$Dust.Storm...Total.Accidents > 0, 1, 0)
contingency_table <- table(total_accidents[, c('dust_binary', 'hail_binary')])
contingency_table
```

```
##            hail_binary
## dust_binary  0  1
##           0 14  2
##           1  5 15
```

```
# install.packages("psych") # Install package if not already installed
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
phi(contingency_table)
```

```
## [1] 0.62
```

Problem 6

Read about correlation on this website and analyze the effect of sample size on correlation coefficients and spurious correlation. Are correlation coefficients affected by outliers ?

SOLUTION:

Increasing the sample size decreases the chance of spurious correlation.

When you have a large sample size you will be more confident that your observed correlation was not a spurious correlation. This increases the confidence of the correlation coefficient.

When we increase the sample-size, sampling error is reduced, making it less possible for "correlations" to occur just by chance alone.

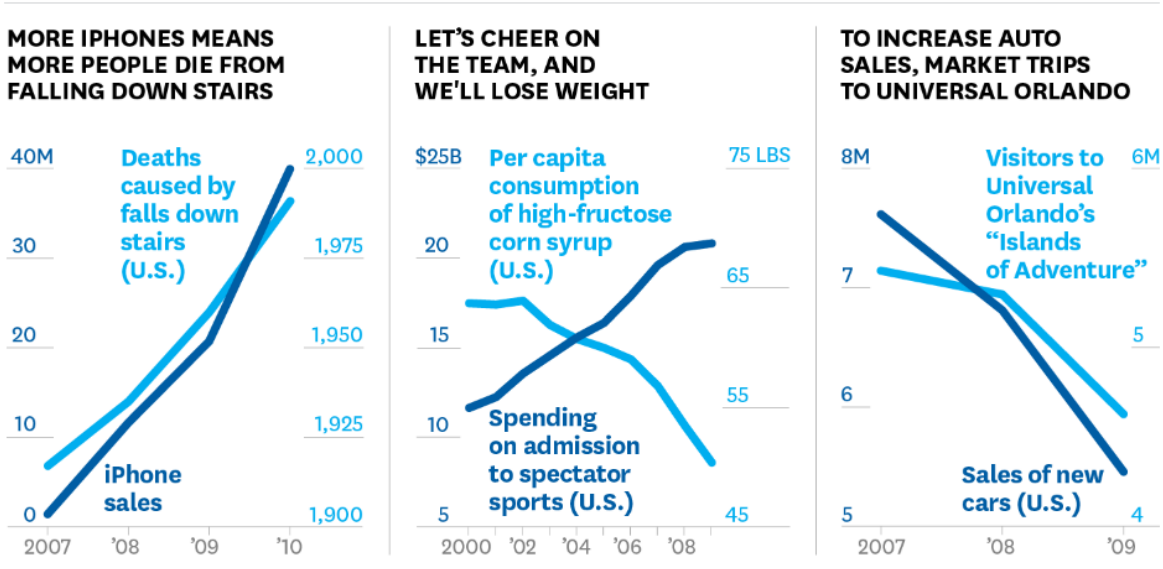
When the sample size is small, even a small change in the data can cause the correlation coefficients to drift with a huge margin whereas for relatively larger sample sizes, the correlation coefficient is fairly stable and it does not depend so much on the particular sample.

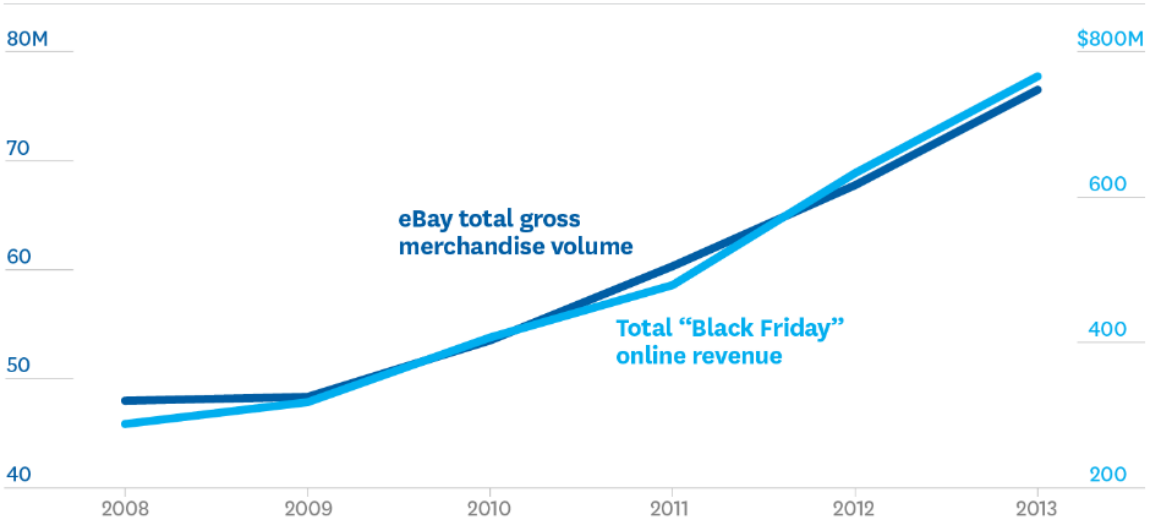
Yes correlation coefficients such as Pearson's correlation coefficient that use continuous variables are affected by the presence of outliers.

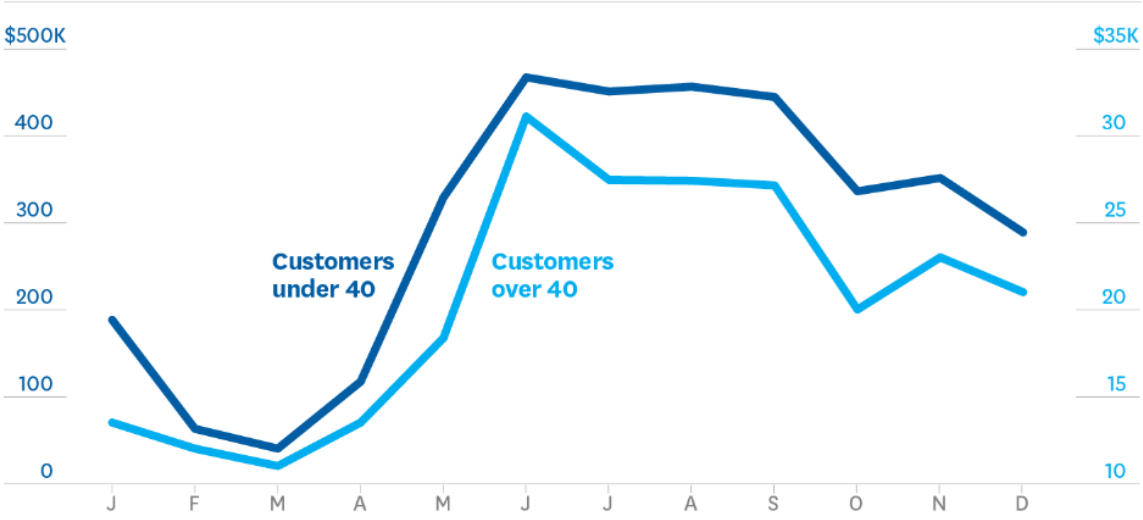
Problem 7

Look at these plots and answer What problems do they have? How do they affect correlation analysis?

1.



2. 

3. 

The y-axis is the monthly revenue of a company and x-axis represents month.

SOLUTION:

1. Spurious Correlation

   Spurious correlation refers to a correlation between two variables that is not meaningful or genuine. It occurs when two variables appear to be correlated, but there is no causal relationship between them. Instead, the correlation arises due to chance or because both variables are influenced by a third, lurking variable.

2. Y axis do not measure the same category (Comparing Dissimilar Variables)

   When the Y-axis in a correlation analysis does not measure the same category or the same type of variables, it can lead to misleading or invalid correlation results. This situation is often referred to as "comparing dissimilar variables" or "apples and oranges" comparison.

3. Skewed Scales for Manipulating Ranges to Align Data

   Maniuplated Ranges and Skewed scales hide the true relationship between variables and paints a deceptive picture of correlation.

For elaborate explanation look here