**Association Analysis:**

The design team at ABCarz Inc. is trying ­to come up with a new design for their upcoming automobile. Before the design begins, they collected data about various automobiles to understand what parameters in a car are likely to influence its turning radius, mileage and type like sports, van.

Data about price, mileage, drive-train-type, engine-size, horsepower, fuel-tank-capacity, length, width, wheel-base etc. of about 90 cars is collected for this analysis.

This data is available in the Cars93 dataset. You can download the same [here](https://lex.infosysapps.com/content-store/Infosys/Infosys_Ltd/Public/lex_auth_0126044372703477761032/web-hosted/assets/Cars93.csv).

1. import pandas as pd
2. cars93 = pd.read\_csv("Cars93.csv")

Let us explore the dataset to see what features exist.

1. cars93.drop("Unnamed: 0", axis=1).columns

Cars were selected at random from among 1993 passenger car models that were listed in both the "Consumer Reports" issue and the "PACE Buying Guide". Pickup trucks and Sport/Utility vehicles were eliminated due to incomplete information in the Consumer Reports source. Duplicate models (e.g., Dodge Shadow and Plymouth Sundance) were listed at most once.

The Cars93 data frame has 93 rows and 27 columns. The details of the 27 columns are as follows:

1. Manufacturer: Manufacturer of the car such as Audi, BMW, Cadillac, Chevrolet etc.
2. Model: Model of the car such as Corvette (Chevrolet), 535i (BMW) etc.
3. Type: Indicates the type of the car. Is a factor with levels "Small", "Sporty", "Compact", "Midsize", "Large" and "Van"
4. Min.Price: Minimum Price (in $1,000): price for a basic version of the car.
5. Price: Midrange Price (in $1,000): average of Min.Price and Max.Price.
6. Max.Price: Maximum Price (in $1,000): price for "a premium version" of the car.
7. MPG.city: City MPG (miles per US gallon by EPA rating).
8. MPG.highway: Highway MPG (miles per US gallon by EPA rating).
9. AirBags: Is a factor with levels "none", "driver only", and "driver & passenger".
10. DriveTrain: Indicates the drive train type of the car. Is a factor with levels: rear wheel, front wheel and 4WD.
11. Cylinders: Number of cylinders in the engine.
12. EngineSize: Size of the engine (litres).
13. Horsepower: Maximum horsepower of the car.
14. RPM: Revolutions per minute (RPM) at maximum horsepower.
15. Rev.per.mile: Engine revolutions per mile (in highest gear).
16. Man.trans.avail: Indicates if a manual transmission version is available. Is a factor with level yes or no.
17. Fuel.tank.capacity: Fuel tank capacity (US gallons).
18. Passengers: Passenger capacity (persons)
19. Length: Length of the car (inches).
20. Wheelbase: Wheelbase (inches).
21. Width: Width of the car (inches).
22. Turn.circle: U-turn space (feet).
23. Rear.seat.room: Rear seat room (inches)
24. Luggage.room: Luggage capacity (cubic feet).
25. Weight: Weight of the car (pounds).
26. Origin: Indicates the origin of the manufacturer. Is a factor with levels non-USA or USA.
27. Make: Combination of Manufacturer and Model.

This is a multivariate data as it has multiple parameters such as price, mileage, drive train type, engine size, horsepower, fuel tank capacity, length, width, and wheelbase.

The given dataset contains both categorical and numeric parameters. Type, DriveTrain, AirBags, Cylinders, Man.trans.avail and Origin are some of the categorical parameters whereas Length, Width, Wheelbase, Turn.cirlce, EngineSize, Horsepower, RPM, and MPG.city are some of the numerical parameters available.

Recall that parameters in a data set can be considered as variables which can be dependent or independent in nature. A variable is said to be dependent if its value changes based on another variable.

The design team has a high interest in analyzing parameters that are likely to influence the mileage of a car, its turn circle and its type. Considering the above viewpoint let us eliminate the following parameters.

1. Manufacturer, Model, Make: Analysis is being done independently of the manufacturer, model and make.
2. Min.Price, Max.Price, Price: Analysis is being done independently of the price of a car
3. MPG.Highway: MPG.city and MPG.Highway are seen to follow a similar trend for a car. Hence only one of these are being considered i.e. MPG.city.
4. No. of passengers, rear seat room, luggage room, origin, manual transmission availability.

After eliminating the above said 12 parameters from the data set, the remaining 15 parameters are being considered for further study.

1. cars93reduced = cars93[["Type","MPG.city","AirBags","DriveTrain","Cylinders",
2. "EngineSize","Horsepower","RPM","Rev.per.mile",
3. "Fuel.tank.capacity","Length","Wheelbase","Width",
4. "Turn.circle","Weight"]]

Of these 15 parameters, 12 are numeric and 3 are categorical. Few rows of this data are shown below.

1. cars93reduced.head()

The analysis of the given dataset is multivariate analysis since it takes multiple variables into consideration.

**Association Analysis**

Association analysis or Association rule learning is a method for discovering interesting associations between variables in datasets. Association analysis deals with finding the degree and direction of the relationship between two variables. The degree/strength indicates the magnitude with which two variables may be associated. The direction indicates if the association between them is directly or inversely proportional.

Consider an outcome variable *y* and predictor variable *x*, the relationship between *y* and *x* can be represented as *y=f(x)*. If *y* and *x* have a linear relationship, then *f(x)* may be represented as *ax+b* and thus *y=ax+b*.

For a variable *y*which can be predicted based on *x*, it can be said that a change in *x*influences *y*, whereas it cannot be guaranteed that a change in *y*indicates a change in *x*. For example, if rainfall is high there may be a greater sale of umbrellas, but having a greater sale of umbrellas need not indicate higher rainfall.

In the given sample dataset, if the turn-circle of a car can be influenced by its length, then turn-circle is said to be the outcome whereas length is said to be the predictor variable.

To find out whether length influences the turn-circle of a car, we study the association between these variables. Such study is termed *Association analysis*. There are two types of analysis multivariate and bivariate.

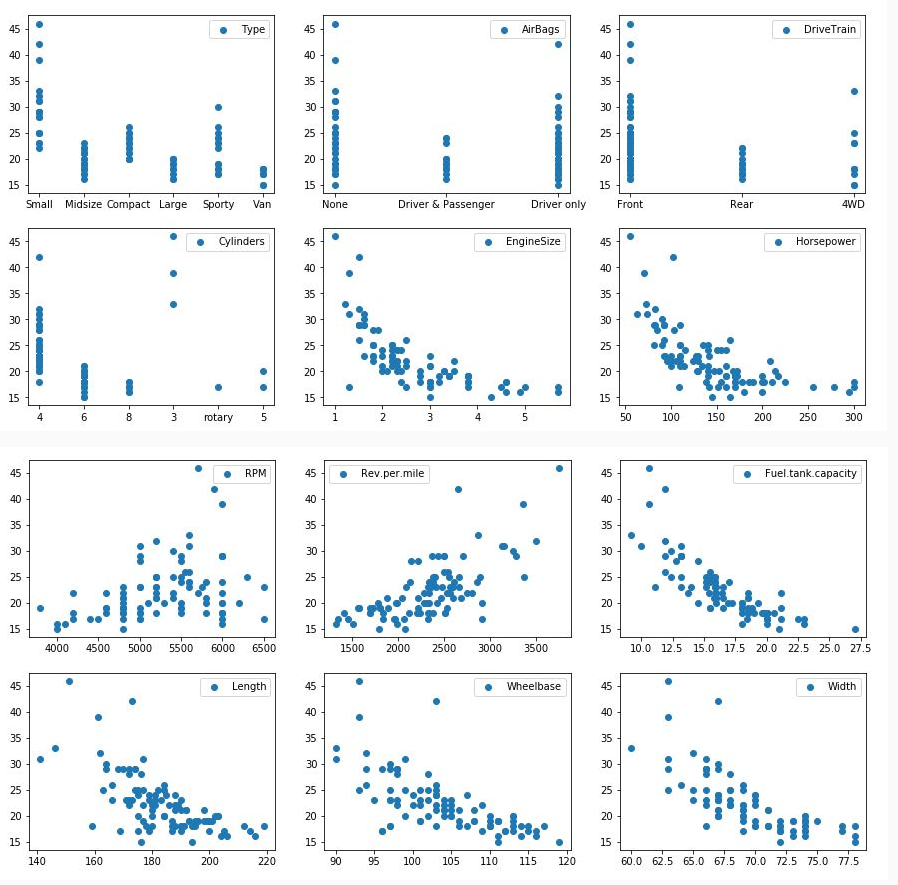
Multivariate analysis is the statistical process of simultaneously analyzing multiple variables. When only two variables are being analyzed then such analysis is termed as bivariate analysis.

The variables being analyzed may either be dependent or independent in nature. A dependent variable can be called outcome or criterion, while the variable(s) on which it depends can be called predictor(s). In bivariate analysis, there is one predictor and one outcome variable.

# Scatter Plot

A scatter plot can be used to get an insight into the nature of the relationship between two numeric variables. Using a scatter plot, one can visually determine whether there exists a linear association between the two variables. The scatter plot matrix of the chosen numeric variables of the given data set is shown below. Let us see an example of MPG.city plotting with other columns except MPG.city in our cars93reduced.

1. import matplotlib.pyplot as plt
2. selcols = ["Type","AirBags","DriveTrain","Cylinders",
3. "EngineSize","Horsepower","RPM","Rev.per.mile",
4. "Fuel.tank.capacity","Length","Wheelbase","Width",
5. "Turn.circle","Weight"]
6. fig, ax = plt.subplots(4, 3,squeeze=False,figsize=(15,15))
7. count = 0
8. for i in range(4):
9. for j in range(3):
10. ax[i, j].scatter(cars93reduced[selcols[count]],cars93["MPG.city"],label=selcols[count])
11. ax[i, j].legend()
12. count+=1



While performing association analysis, one might consider a subset of all possible associations between variables in a dataset because all the associations may not give insight into which feature may be important in that dataset.

Using the previous scatter plot matrix, some of the relationships can be eliminated from further analysis. For example, there might be relationships between RPM and MPG.city, RPM and Wheelbase, RPM and Horsepower, Horsepower and Wheelbase, etc. as shown below.

1. sns.pairplot(cars93reduced, vars=["Horsepower","RPM","Wheelbase","MPG.city"])

However, the association between RPM and MPG.City, Horsepower and Wheelbase looks interesting as compared to others as it suggests a linear relationship. Therefore, one might choose these associations (RPM and MPG.city, Horsepower and Wheelbase) for further analysis.

Some scenarios where association analysis is being used are:

1. Credit card purchases can provide insight into the type of products a customer is likely to purchase. Using the credit card statements, one can determine if the customer spends more on household items/ jewellery etc.
2. Supermarkets can rearrange their shelves by understanding the combinations of frequently bought items. For example, if the supermarket observes that bread and butter are frequently purchased together, then the supermarket may choose to place these items close to each other.
3. Telecommunication agencies can structure product bundles based on commonly associated options (internet packs, SMS services and other value added services) to maximize revenue.
4. Click stream analysis on websites is used to observe patterns in user’s browsing behavior in order to deliver content accordingly. For example, the click stream analysis may suggest that visitors who land on a webpage X, clicked on links A, B and C more often than on links D,E and F. Such observations provide an insight on how to personalize and recommend the content to website visitors.

**Finding associations**

The design team wants to figure out which parameters of the data may be related. They speculate that the following associations are possible.

1. MPG.City and EngineSize
2. MPG.City and Horsepower
3. MPG.City and RPM
4. EngineSize and Horsepower
5. RPM and Horsepower
6. Length and Wheelbase
7. Length and Turn.Circle
8. Wheelbase and Turn.Circle
9. DriveTrain and Type
10. Type and AirBags
11. DriveTrain and MPG.City
12. Horsepower and Type

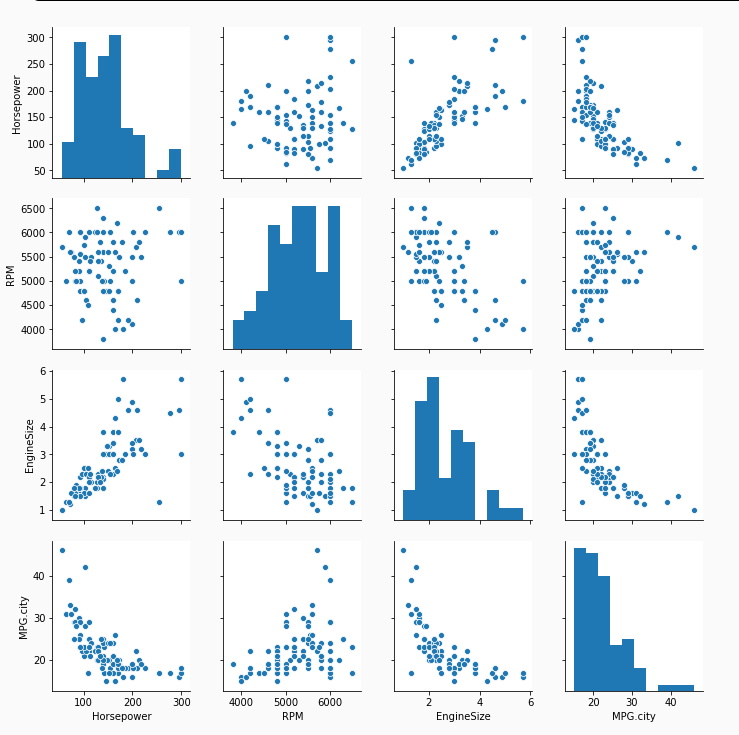
In the given data set, DriveTrain, Type and AirBags are categorical variables, whereas MPG.City, Horsepower, RPM, Turn.circle, Length, Wheelbase and EngineSize are numeric variables as shown below.

1. columns = ["MPG.city","EngineSize","Horsepower","RPM","Length","Wheelbase","Turn.circle","DriveTrain","Type","AirBags"]
2. cars93reduced[columns].head(10)

Associations 1 to 8 suggests the association between two numeric variables, 9 and 10 suggest an association between two categorical variables whereas 11 and 12 suggest an association between a numeric and a categorical variable.

Recall that to find out if the association between two numeric variables may be linear, a scatter plot is drawn. The scatter plot matrix for associations 1 to 5 comprising variables MPG.city, EngineSize, Horsepower and RPM is shown below.

1. import seaborn as sns
2. sns.pairplot(cars93reduced, vars=["Horsepower","RPM","EngineSize","MPG.city"])

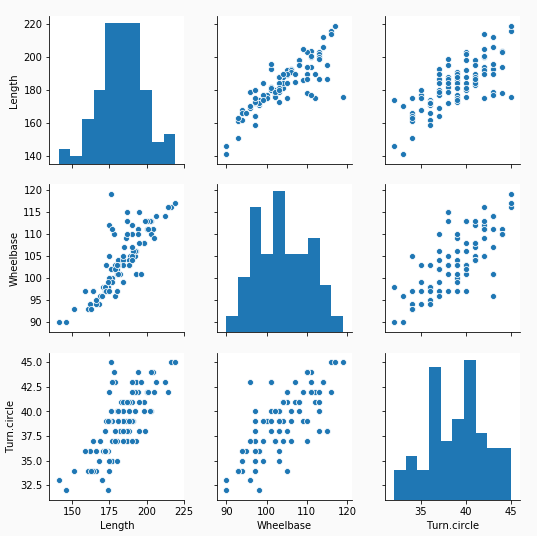


The following associations seem to be linear based on the scatter plot:

1. MPG.city and EngineSize
2. MPG.city and Horsepower
3. MPG.city and RPM
4. EngineSize and Horsepower

Similarly, the scatter plot matrix for associations 6 to 8 comprising variables Length, Wheelbase and Turn.cirlce shown below indicates possible linear associations between

1. Length and Wheelbase
2. Length and Turn.circle
3. Wheelbase and Turn.circle
4. import seaborn as sns
5. sns.pairplot(cars93reduced, vars=["Length","Wheelbase","Turn.circle"])



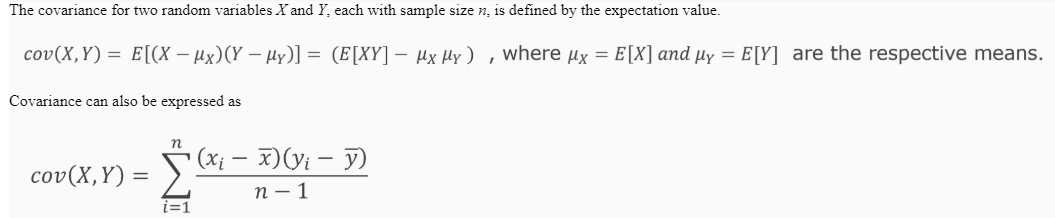
Based on the observations from the scatter plots, the following associations are selected for further analysis:

1. MPG.city and EngineSize
2. MPG.city and Horsepower
3. MPG.city and RPM
4. EngineSize and Horsepower
5. Length and Wheelbase
6. Length and Turn.circle
7. Wheelbase and Turn.circle

# Covariance

Given that there are linear associations between MPG.city and Horsepower, MPG.city and RPM, and MPG.city and EngineSize, the design team at ABCarz wants to understand the degree and direction of these associations.

Covariance is a measure of how much two random variables change together. It helps measure the direction and strength (degree) of the linear association between a pair of numeric variables.



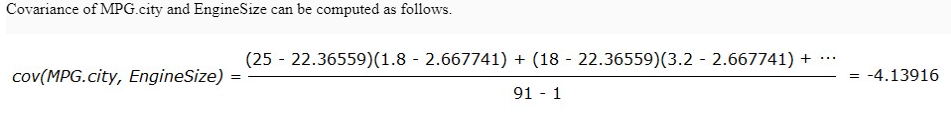
Covariance helps to understand the association between the variables X and Y. It helps to notice the effect of the change in the value of X on Y, i.e. if the value of X increases, does the value of Y increase or decrease.

If the covariance value is positive, it indicates that while observing an increase in the value of X, the value of Y is likely to increase. Similarly, if the covariance value is negative, it indicates that while observing a decrease in the value of X, the value of Y is likely to decrease.

While covariance helps determine the direction of the association, its magnitude/strength is not easy to interpret.

In Python, the covariance of X and Y can be calculated using X.cov(Y)

Here X and Y are Data frame objects.



Here, mean(MPG.city) = 22.36559 and mean(EngineSize) = 2.667741

1. sum((cars93["MPG.city"] - cars93["MPG.city"].mean()) \* (cars93["EngineSize"] - (cars93["EngineSize"].mean()))/(cars93.shape[0]-1))
2. *#prints -4.139165497896213*
3. cars93["EngineSize"].cov(cars93["MPG.city"])
4. *#prints -4.139165497896213*

The obtained covariance value (-4.13916) indicates that the direction of covariance is negative and its degree/magnitude is 4.13916.

The covariance of all the speculated relationships is as shown below:

1. cars93["EngineSize"].cov(cars93["MPG.city"])
2. *#prints -4.139165497896213*
3. cars93["Horsepower"].cov(cars93["MPG.city"])
4. *#prints -197.9798971482001*
5. cars93["RPM"].cov(cars93["MPG.city"])
6. *#prints 1217.4789621318369*
7. cars93["EngineSize"].cov(cars93["Horsepower"])
8. *#prints 39.776998597475455*
9. cars93["Length"].cov(cars93["Wheelbase"])
10. *#prints 82.02197288452543*
11. cars93["Length"].cov(cars93["Turn.circle"])
12. *#prints 34.780621785881266*
13. cars93["Wheelbase"].cov(cars93["Turn.circle"])
14. *#prints 15.899836372136514*

Here it is observed that:

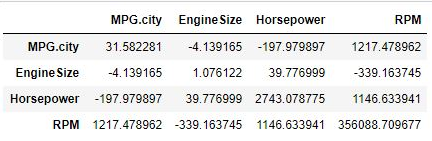
1. There is a negative association between:
   1. MPG.city and EngineSize
   2. MPG.city and Horsepower
2. There is a positive association between:
   1. MPG.city and RPM
   2. EngineSize and Horsepower
   3. Length and Wheelbase
   4. Length and Turn.circle
   5. Wheelbase and Turn.circle

This indicates that MPG.city is inversely proportional to EngineSize, inversely proportional to Horsepower, and directly proportional to RPM in the given sample dataset.

Similarly, Turn.circle is directly proportional to Wheelbase and directly proportional to Length in this sample dataset.

The covariance matrix captures the values of covariance between selected parameters of the data set. The covariance matrix of parameters - MPG.city, EngineSize, Horsepower and RPM in the given dataset is as shown below.

1. cars93[["MPG.city","EngineSize","Horsepower","RPM"]].cov()



# Correlation

Having understood that there could be a negative association between MPG.city and EngineSize, and MPG.city and Horsepower, the design team ponders on the following question:

* Is it likely that decreasing the EngineSize yields a better mileage than decreasing the Horsepower?

Given that cov(Horsepower, MPG.city) = -186.09072 and cov (EngineSize, MPG.city) = -4.1297314, one might assume that decreasing the Horsepower is more likely to yield a higher mileage as compared to decreasing the EngineSize. Is this a valid assumption?

Similarly, having understood that Length and Turn.circle, Wheelbase and Turn.circle are positively associated

* Is it likely that decreasing the Wheelbase contributes more to the decrease in Turn.circle in comparison to the decrease in the Length?

Given that cov(Length, Turn.circle) = 35.43822 and cov(Wheelbase, Turn.circle) = 16.44139, one might assume that decreasing the Length is more likely to reduce the Turn.circle as compared to decreasing the Wheelbase. Is this a valid assumption?

The above assumptions cannot be made because two covariance values cannot be compared directly. This is because covariance involves the use of the means of participating variables, as shown below.



To overcome this limitation, **correlation**is used.

Correlation is a measure of the strength and direction of the linear association between two quantitative variables.

Correlation is normalized covariance and is generally obtained by dividing the covariance by the product of standard deviation of the variables



Owing to normalization, correlation always lies in the range of [-1, +1].

The correlation of a sample is represented using *r* and that of a population is represented using *ρ*(rho).

The sign of correlation indicates the direction of the association, as expressed below

* positive association: *r*or *ρ* > 0
* negative association: *r*or *ρ* < 0
* no linear association: *r*or *ρ* ≈ 0

The closer r (or ρ) is to ±1, the stronger the linear association.

If the correlation is closer to 0, then the variables might be statistically independent.

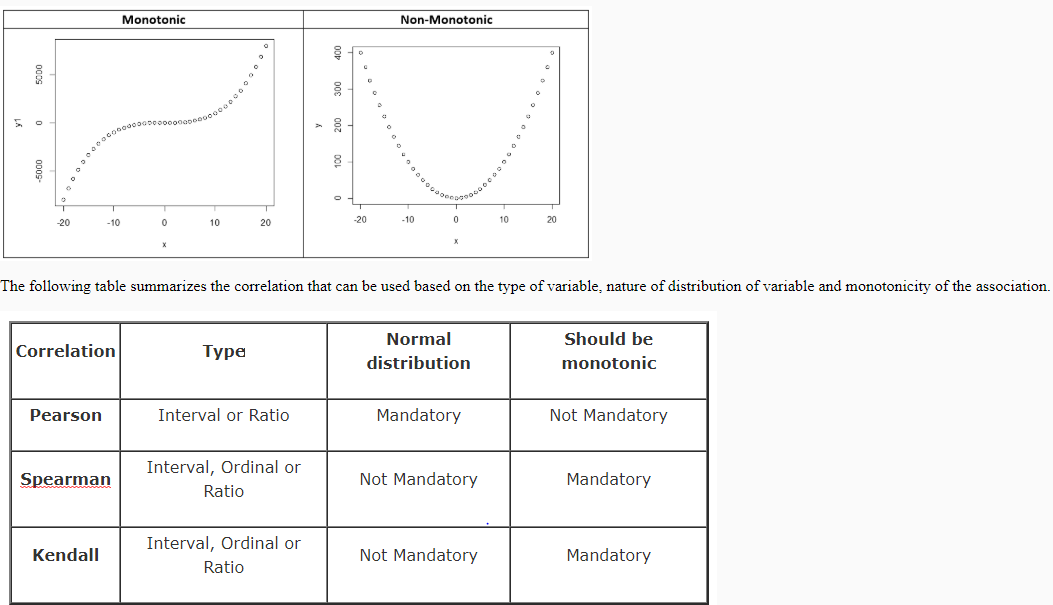
Correlation is computed using corr method of the pandas library.

# Types of correlation

Based on the type of variable, the nature of its distribution and the monotonicity of the association, the appropriate correlation has to be used. Commonly used correlations are **Pearson**, **Spearman** and **Kendall**.

An association between two variables y and x where y=f(x) is said to be monotonic if and only if the value of y is either entirely increasing or decreasing as x increases.

The following graphs distinguish between monotonic and non-monotonic associations.



# Pearson correlation

Pearson correlation coefficient is used on numeric variables pulled out of a population that is normally distributed. These numeric variables have to be of the interval or ratio scale.

It is defined as

The Shapiro-Wilk test on Turn.circle, Length and Wheelbase in the given data set for a significance level of 0.05 is shown below.

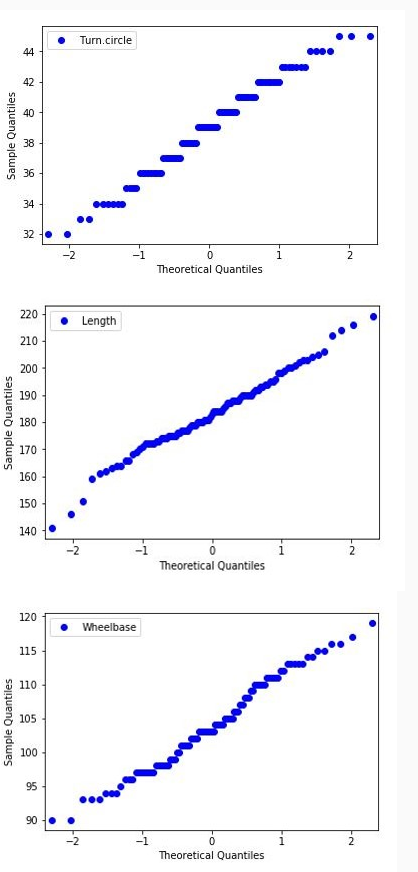
1. from scipy.stats import shapiro
2. shapiro(cars93["Turn.circle"])
3. *#prints (0.9762530326843262, 0.08784385770559311)*
4. shapiro(cars93["Length"])
5. *#prints (0.9909828305244446, 0.7838887572288513)*
6. shapiro(cars93["Wheelbase"])
7. *#prints (0.9785544276237488, 0.12946628034114838)*

The first value is the test statistic and the second value is the p-value.

The p-value obtained from the Shapiro-Wilk test suggests that the underlying population could be normally distributed (p-value > chosen significance level).

QQplot can also be used to determine if the underlying data is normally distributed. The QQ plot of Turn.circle, Length and Wheelbase is as shown below.

1. import numpy as np
2. import matplotlib.pyplot as plt
3. import statsmodels.api as sm
4. sm.qqplot(cars93["Turn.circle"])
5. plt.legend(["Turn.circle"])
6. plt.show()
7. sm.qqplot(cars93["Length"])
8. plt.legend(["Length"])
9. plt.show()
10. sm.qqplot(cars93["Wheelbase"])
11. plt.legend(["Wheelbase"])
12. plt.show()



The covariance and Pearson correlation of Length and Turn.circle, Wheelbase and Turn.circle are shown below.

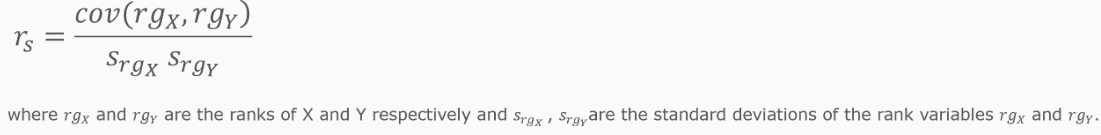
1. cars93["Length"].cov(cars93["Turn.circle"])
2. *#prints 34.780621785881266*
3. cars93["Length"].corr(cars93["Turn.circle"],method="pearson")
4. *#prints 0.7389545018604418*
5. cars93["Wheelbase"].cov(cars93["Turn.circle"])
6. *#prints 15.899836372136514*
7. cars93["Wheelbase"].corr(cars93["Turn.circle"],method="pearson")
8. *#prints 0.7233244020299519*

The correlation values of Length & Turn.circle and Wheelbase & Turn.circle, suggest that the association between these variables is strong (correlation is close to 1).

# Spearman Correlation

The Spearman Rho correlation coefficient is a rank based coefficient, where the variables need not be normally distributed. However, it is mandatory for these variables to be either in interval, ordinal or ratio scale. In addition, it is expected that the association between the variables is monotonic.

It is defined as



The Shapiro-Wilk test with a significance level of 0.05 for Horsepower, RPM, EngineSize and MPG.city is as shown below.

1. shapiro(cars93["Horsepower"])
2. *#prints (0.9358058571815491, 0.00019156414782628417)*
3. shapiro(cars93["RPM"])
4. *#prints (0.9739203453063965, 0.059240180999040604)*
5. shapiro(cars93["EngineSize"])
6. *#prints (0.9360973238945007, 0.00019901638734154403)*
7. shapiro(cars93["MPG.city"])
8. *#prints (0.8583050966262817, 5.762488086702433e-08)*

The p-value here suggests that the data in the underlying population may not be normally distributed (p-value smaller than chosen significance value).

Therefore the association between these variables can be measured using Spearman correlation coefficient.

Rank is assigned by picking a random variable and comparing against other values of that variable in the dataset.

The below code illustrates the assignment of rank for the first 10 values of Horsepower using rank() method in Pandas.

1. first10hp = cars93["Horsepower"].head(10)
2. first10hp = first10hp.sort\_values(ascending=False)
3. rankfirst10hp = first10hp.rank()
4. print(list(rankfirst10hp)) *#prints [10.0, 8.5, 8.5, 7.0, 5.5, 5.5, 3.5, 3.5, 2.0, 1.0]*

It can be observed that the highest value gets the highest rank. Here 208 is the highest value, among 10 values, hence it is assigned the highest rank (10.0). If a value is found to be repeating, then it is assigned the average rank. Here 200 occurs at 9th and 8th position and therefore it receives the rank 8.5 which is the average of 9 and 8. Similarly, 172 and 170 receive 5.5 and 3.5 as rank respectively.

The Spearman correlation of Horsepower and MPG.city, RPM and MPG.city, and EngineSize and MPG.city are shown below

1. cars93["Horsepower"].corr(cars93["MPG.city"],method="spearman")
2. *#prints -0.7893070882303793*
3. cars93["RPM"].corr(cars93["MPG.city"],method="spearman")
4. *#prints 0.38964509885135185*
5. cars93["EngineSize"].corr(cars93["MPG.city"],method="spearman")
6. *#prints -0.8212079921743175*

The correlation values suggest that the association between EngineSize and MPG.city is stronger than the association between Horsepower and MPG.city. Here it can be noticed that the strength of the association is attributed to the magnitude of the correlation value and not the sign.

The Pearson and Spearman correlation of RPM & MPG.city is as shown below.

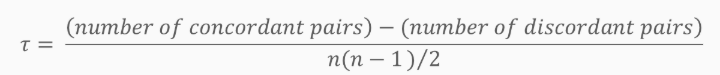
1. cars93["RPM"].corr(cars93["MPG.city"],method="pearson")
2. *#prints 0.36304512864824134*
3. cars93["RPM"].corr(cars93["MPG.city"],method="spearman")
4. *#prints 0.38964509885135185*

It can be noticed here that the correlation coefficients obtained using Pearson and Spearman for RPM and MPG.city do not differ significantly.

# Kendall Tau Correlation

The Kendall Tau correlation coefficient is a rank based coefficient similar to Spearman correlation, where the variables need not be normally distributed. However, it is mandatory for these variables to be either in interval, ordinal or ratio scale. In addition, it is expected that the association between the variables is monotonic.

It is defined as



Let (x1, y1), (x2, y2), …, (xn, yn) be a set of observations derived from the two joint random variables X and Y.

In a way that all the values of (xi) and (yi) are to be different.

If both the elements of a particular pair agree(both the elements of the lesser or greater compared to the other pair) then they are called concordant pairs. Similarly if the elements of the pair disagree then they are called discordant. And they are neither concordant nor discordant if both elements are equal to the other observation.

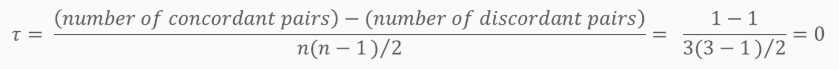
Computing the number of concordant and discordant pairs for a given data set is as illustrated below.

Ranks of three entries of RPM and MPG.city from the sample dataset is given

1. first25rpm = np.array(cars93["RPM"].head(25).rank())
2. first25mpg = np.array(cars93["MPG.city"].head(25).rank())
3. np.array([first25rpm,first25mpg])[:,19:22]
4. *#prints array([[18. , 12.5, 8.5],*
5. [14. , 20.5, 14. ]])

Here, the pair of (x1=18.0, y1=14.0) and (x2=12.5, y2=20.5) is discordant because x1 > x2 and y1 < y2. The pair of (x2=12.5, y2=20.5) and (x3=8.5, y3=14.0) is concordant because x2 > x3 and y2 > y3. The pair of (x1=18.0, y1=14.0) and (x3=8.5, y3=14.0) is neither concordant nor discordant because y2 = y3.

The Kendall Tau correlation for these pairs can thus be computed as



The Kendall Tau correlation for these pairs is observed to be 0 as shown below.

1. rpm\_values = pd.DataFrame({"rpm\_ranks" : [18.0 , 12.5,  8.5]})
2. mpg\_values = pd.DataFrame({"mpg\_ranks" :[14.0 , 20.5, 14.0 ]})
3. mpg\_values["mpg\_ranks"].corr(rpm\_values["rpm\_ranks"], method = "kendall")
4. *#prints 0.0*

The Kendall Tau correlation coefficient of MPG.city and Horsepower, MPG.city and RPM, MPG.city and EngineSize are shown below.

1. cars93["Horsepower"].corr(cars93["MPG.city"],method="kendall")
2. *#prints -0.6232924179871033*
3. cars93["RPM"].corr(cars93["MPG.city"],method="kendall")
4. *#prints 0.29452246826301876*
5. cars93["EngineSize"].corr(cars93["MPG.city"],method="kendall")
6. *#prints -0.6820807909116785*

Kendall coefficient value for a dataset is usually lower than the Spearman coefficient value for the same data set. The relationship between the Kendall Tau coefficient and Spearman Rho coefficient is given by



# Correlation Matrix

For multivariate data, the correlation matrix captures the values of correlation between the selected parameters of the data set.

The Pearson correlation matrix of MPG.city, EngineSize, Horsepower and RPM is shown below.

1. cars93[["MPG.city","EngineSize","Horsepower","RPM"]].corr()



# Cautions with correlation

While computing correlation to determine the nature of association between two variables, the following cautions have to be kept in mind:

1. Correlation is not the right measure for finding association for non-linear relationships. To determine whether two variables may be linearly related, a scatter plot can be used.
2. Pearson correlation can be affected by outliers.  A box plot can be used to identify the presence of outliers. The effect of outliers is minimal for Spearman correlation therefore, if outliers cannot be manipulated or eliminated from the analysis with proper justification, Spearman correlation is preferred.
3. A correlation value close to 0 indicates that the variables are not linearly associated, however these variables may still be related. Thus it is advised to plot the data.
4. Correlation does not imply causation i.e. based on the value of correlation, it cannot be asserted that one variable causes the other.

**Association between categorical variables**

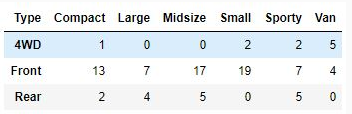
# Chi-Squared Test

The design team at ABCarz speculate that DriveTrain may influence the Type of the car.

Since these variables are categorical, covariance and correlation may be inappropriate metrics to determine the existence of an association between them.

Two-way tables are used to examine the relationship between two categorical variables. The two-way table between DriveTrain and Type is shown below.

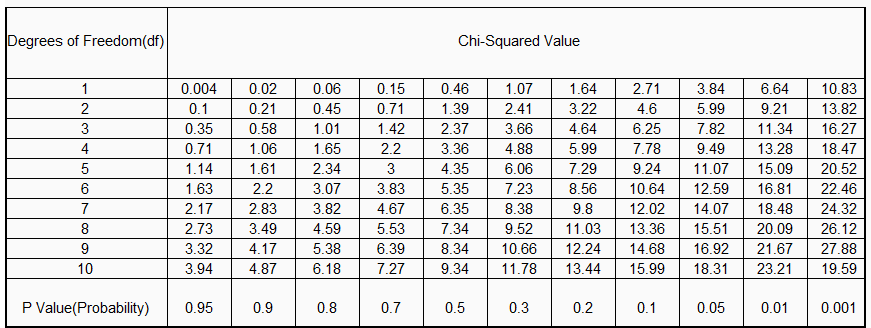
1. two\_way\_table = pd.crosstab(index=cars93["DriveTrain"],
2. columns=cars93["Type"])
3. two\_way\_table.index = ["4WD","Front","Rear"]
4. two\_way\_table

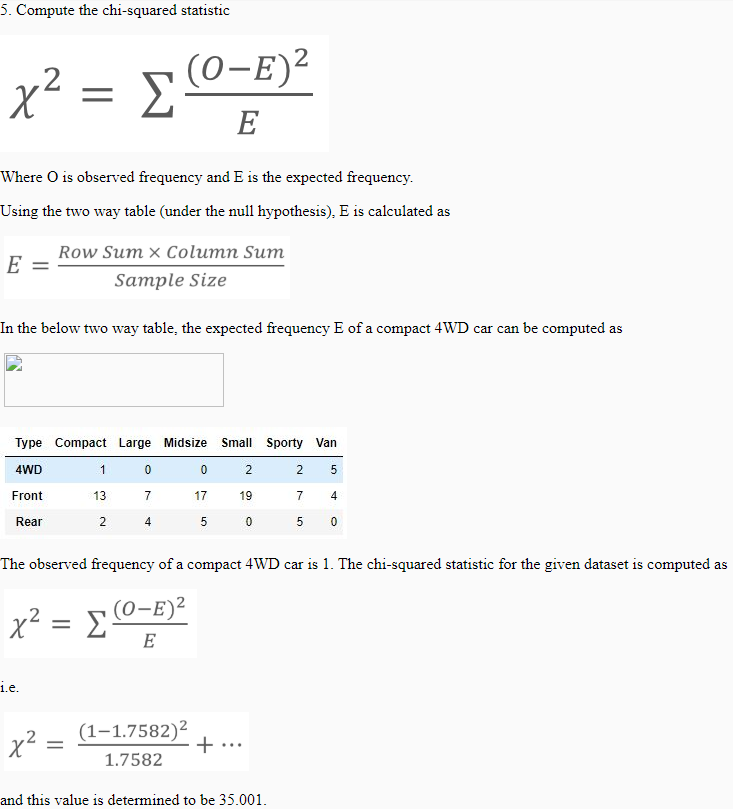
The existence of an association can be determined by performing a **Chi-squared** **test** on the two-way table.

 Chi-squared test for independence is used to test if evidence pertaining to the relationship of variables in the sample is strong enough to generalize this relationship for the population.

For the given dataset, the existence of a relationship between DriveTrain and Type of car can be tested using the Chi-squared test as follows.

1. Define the null and the alternate hypothesis  
   H0 (Null hypothesis): The variables are independent.  
   H1 (Alternate hypothesis): The variables are dependent.
2. Select a level of significance – usually 0.05 or 0.1. For the given dataset, the level of significance is selected as 0.05.
3. Identify the degrees of freedom. This is usually computed as, (no. of rows – 1) × (no. of columns – 1).
4. For the relationship between DriveTrain and Type of car, the degrees of freedom is (3 – 1) × (6 – 1) = 10. Find the critical value for the given significance level and degrees of freedom from the table below



For the significance level of 0.05 and the degrees of freedom as 10, the critical value is traced as 18.31.

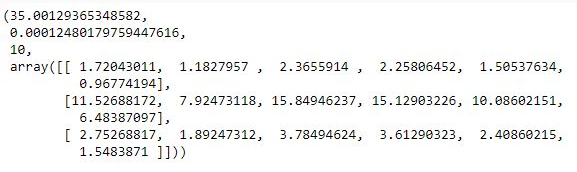
6.    In order to reject the null hypothesis, the computed chi-squared value should be greater than the identified critical value.

       Here, the computed value is 35.001 which is larger than the critical value 18.31.

       Therefore, the null hypothesis is rejected, i.e. the DriveTrain and Type of a car are not independent.

Chi-squared test can be performed using chi2\_contingency,chi2 in scipy.stats. The chi-squared test on the given data set for the association between drive-train and type is shown below.

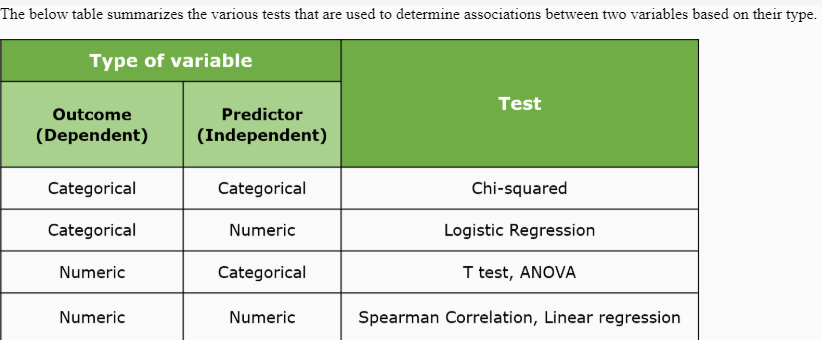
1. from scipy.stats import chi2\_contingency,chi2
2. chi2.ppf(1-0.05,10)
3. chi2\_contingency(two\_way\_table)

The output of the above code is as follows.

* The first line indicates the chi-squared value.
* Second line indicates the p-value.
* Third indicates the degrees of freedom

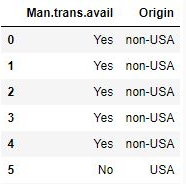
It can be observed that the computed chi-squared statistic is greater than the critical value, hence the null hypothesis is rejected.

Also, the p-value obtained is lesser than the chosen significance level. This is another indicator which helps determine whether to reject or accept the null hypothesis. If the obtained p-value is lower than the significance level, then the null hypothesis is rejected.



**Association between two binary variables**

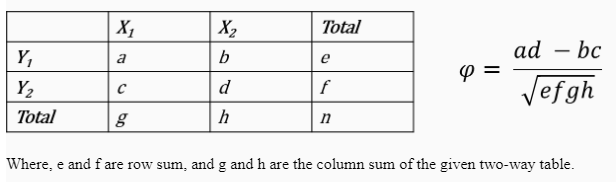
# Phi coefficient

The design team is interested to know if there is any association between the availability of manual transmission in the car and the origin of the car.

The snap of the data indicating the presence of manual transmission and the origin of the car is shown below.

1. cars93[["Man.trans.avail","Origin"]].head(10)

Here, Man.trans.avail and Origin are binary variables and the existence of the association between binary variables can be determined using the Phi coefficient.

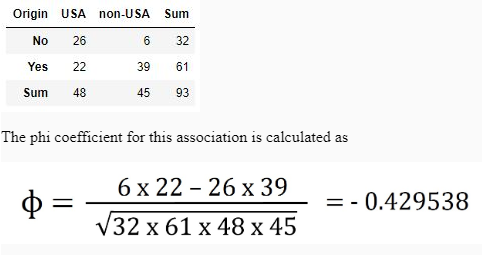
The Phi coefficient is determined using the two-way table between the binary variables. The two binary variables are said to be positively associated if most of the data fall along the diagonal. The Phi value for a given two-way table is as expressed below.

The two-way table between Man.trans.avail and Origin is as shown below.

We create a Sum column with the sum of values in columns using axis parameter.

1. origin\_trans = pd.crosstab(index=cars93["Man.trans.avail"],
2. columns=cars93["Origin"])
3. origin\_trans.index = ["No","Yes"]
4. origin\_trans["Sum"] = origin\_trans.sum(axis=1)

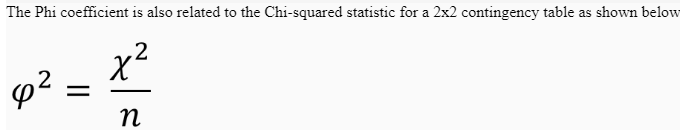
We create an index Sum with the sum of values in rows. Transpose and append it to our existing data frame.

1. other = origin\_trans.sum(axis=0)
2. other = pd.DataFrame({"Sum":list(other)})
3. other.index=["USA","non-USA","Sum"]
4. transposed\_other = other.T
5. origin\_trans.append(transposed\_other,sort=False)

The Phi coefficient is also the Pearson correlation coefficient for two binary variables. This can be determined using the corr method.

1. man\_vec = cars93["Man.trans.avail"].apply(lambda x: 1 if x=="Yes" else 0)
2. origin\_vec = cars93["Origin"].apply(lambda x: 1 if x=="USA" else 0)
3. man\_vec.corr(origin\_vec, method="pearson")
4. *#prints -0.4295382099074451*

The value of Phi also lies in the range [-1,+1] and a value closer to 1 indicates a stronger association.



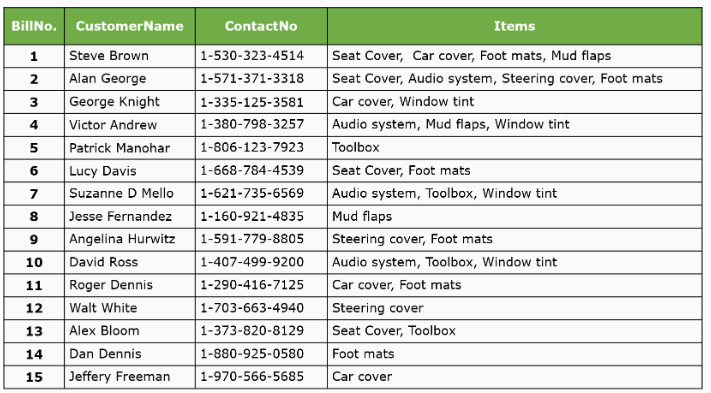
# Market Basket Analysis

ABCarz have various showrooms across the city where they sell their cars and various other automobile accessories to customers. The marketing managers of ABCarz are looking to market their products better by analyzing customer buying patterns. Data about the various accessories that are purchased is collected for this analysis.

The data presented here is hypothetical and comprises some attributes as described below.

* BillNumber: This is the Bill number for each customer purchase.
* CustomerName: Name of the Customer.
* ContactNo: Contact number of the customer.
* Items: List of items purchased by the customer. This is a factor with various levels

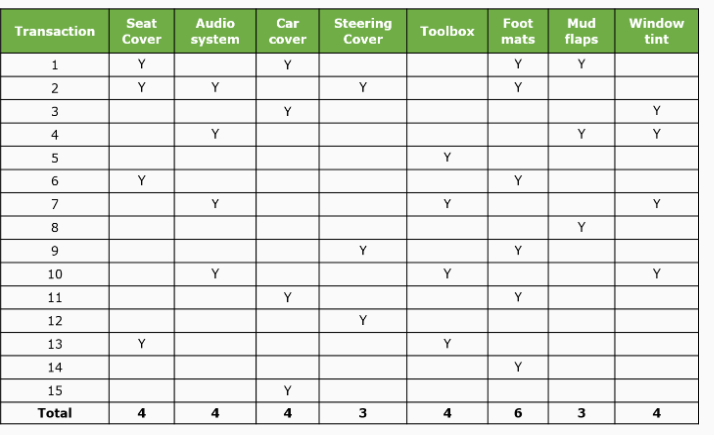
A snap of the data is shown below.



Of the various attributes in the data set, the analysis that leads to determining purchase behavior of customers arises from the items attribute. The marketing team seeks to study the items attribute more closely to determine associations between various items.

From the given data, they wish to find the items that were purchased most frequently. They also wish to determine the item(s) which encouraged the customer to purchase additional item(s). Such analysis is commonly termed as **Market Basket Analysis**, where the interesting associations between various items are determined.

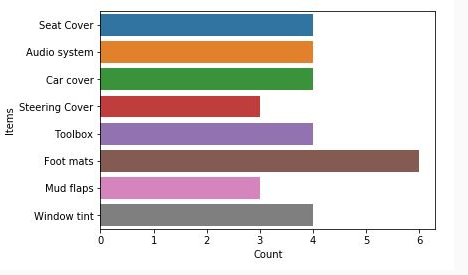
From the given sample data set, the most frequently purchased item can be determined using a frequency table, as shown below



This table reveals that foot mats are the most frequently purchased item.

The transactions can be plotted on a bar plot to determine the most frequent item. The code to convert the transactions to the sparse matrix and display the bar plot is as shown below.

1. import pandas as pd
2. import seaborn as sns
3. import matplotlib.pyplot as plt
4. df = pd.read\_excel("market basket - car accessories.xlsx")
5. df = df.replace(float("nan"), 0)
6. df = df.replace("y", 1)
7. df = pd.DataFrame({"Items":list(df.columns),
8. "Count":list(df.sum(axis=0))
9. })
10. bar = sns.barplot(data = df,y="Items",x="Count") *#plot the items x count*



The team seeks to determine the purchase behavior wherein the purchase of foot mat was accompanied by the purchase of which other items. For example, the team seeks to determine the no. of transactions wherein the purchase of foot mats is accompanied by the purchase of seat cover. The team also seeks to determine the no. of transactions where the purchase of foot mats is accompanied by the purchase of mud flaps.

From the frequency table, it can be observed that seat covers and foot mats were purchased together in 3 transactions. Also, mud flaps and foot mats were purchased together in 1 transaction alone.

These associations can be represented as association rules which are specified as follows:

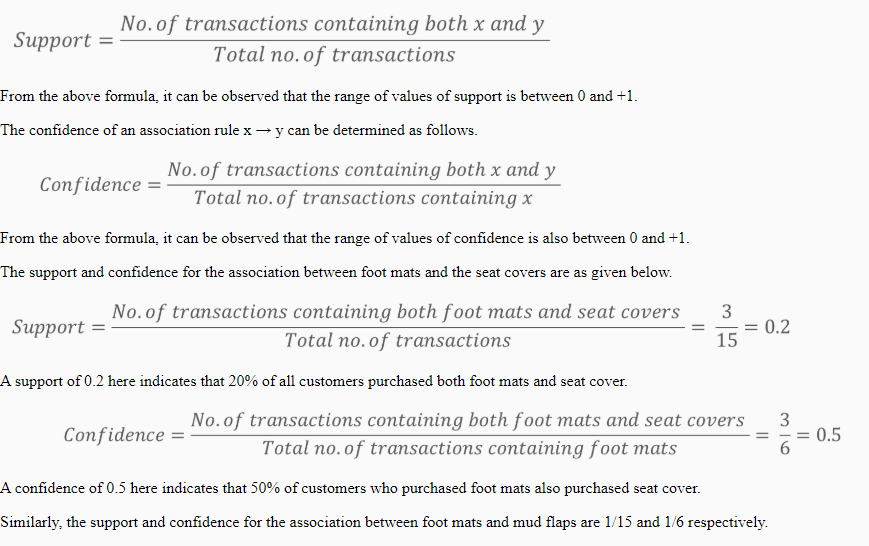
1. Foot mats → Seat cover: Customers who bought Foot mats also bought seat cover.
2. Foot mats → Mud flaps: Customers who bought Foot mats also bought mud flaps.

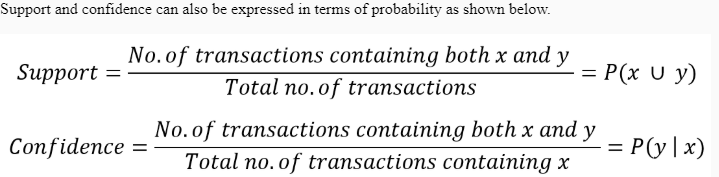
In order to determine if these association rules are interesting, measures such as support & confidence can be used.

# Support and Confidence

Support and confidence indicate how interesting an association rule is. Support indicates the usefulness of the rule, while confidence indicates the certainty of the rule. For a given association rule x → y, support indicates the fraction of transactions that contain both x and y. The confidence indicates the fraction of transactions that contain both x and y among the transactions that contain x.

The support of an association rule x → y can be determined as follows.





An association rule can be said to be sufficiently useful by interpreting its support and confidence. Typically, the obtained value of support is compared against a chosen threshold to signify that the association rule must be observed in at least a significant number of transactions.

For example, if the chosen support threshold is 0.1 then the association rule is expected to hold true for at least 10% of the transactions. The choice of this threshold is based on domain and the total number of transactions being studied.

Typically an association rule is said to be interesting if:

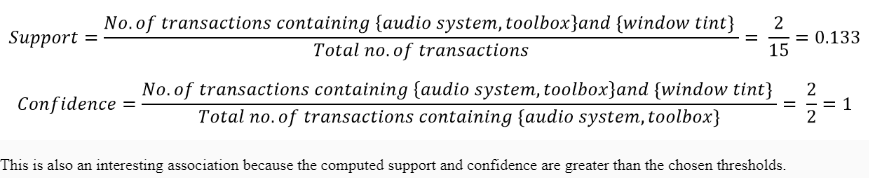
1. *Support  ≥ min\_support threshold*
2. *Confidence ≥ min\_confidence threshold*

The choice of min\_support threshold and min\_confidence threshold is purely based on domain knowledge.

If the min\_support threshold and min\_confidence threshold for this sample study are considered as 0.1 and 0.4 respectively, then the association between foot mats and seat cover can be said to be interesting because the support and confidence for this association are 0.2 and 0.5 respectively.

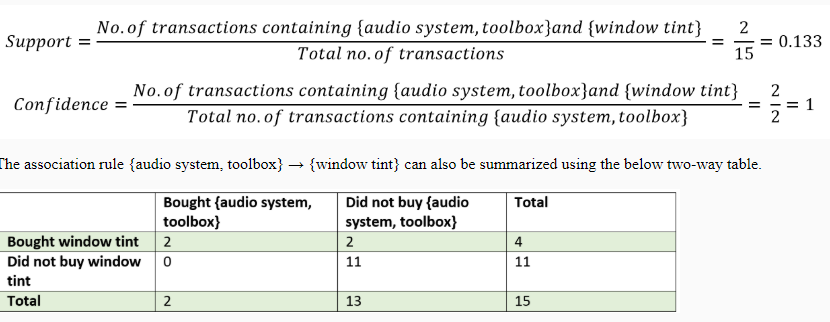
It is also possible that customers buy a set of items together. For example, in the given sample dataset, customers who purchased both audio system and toolbox also purchased window tint.

The support-confidence for the association between {audio system, toolbox} and {window tint} expressed as {audio system, toolbox} → {window tint} is as follows.



# Lift

Consider the below scenario for the given sample data where the usefulness of the association rule {audio system, toolbox} → {window tint} is being determined using support and confidence.

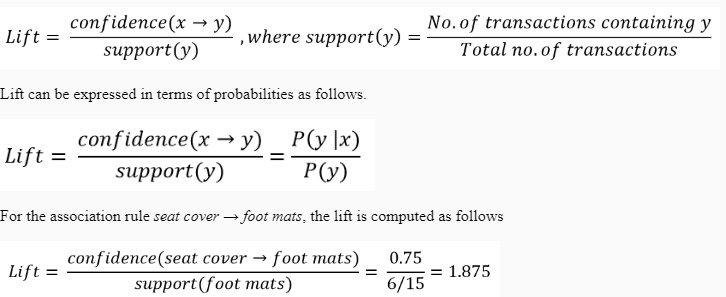


Based on the above two-way table, it can be observed that

This is because for an association between x and y, represented as association rule x → y, confidence considers only x and the co-occurrence of x.

To determine whether the association rules discovered are sufficiently interesting enough, additional metrics such as lift, conviction and leverage can be used.

For an association rule represented as x → y, Lift is calculated by the ratio of the confidence of the association and the number of transactions containing the rule consequent. Where x is the antecedent and y is the consequent of the rule. This measure indicates the relationship between x and y by measuring their strength of the association which can be computed as shown below.



Larger values of lift indicate stronger association because lift indicates the strength of association. The nature of association based on the value of lift can be summarized as below:

1. If lift = 1, then x and y are said to be disassociated/statistically independent. The presence of x may have no influence on the presence of y.
2. If lift > 1, then x and y are said to be positively associated i.e. the presence of x tends to indicate the presence of y.
3. If lift < 1, then x and y are said to be negatively associated.

# Apriori

When the data set has a large number of transactions with a considerably large number of items, discovering interesting associations among items will be time-consuming. The technique illustrated earlier may not facilitate the purpose of determining interesting associations in an efficient manner. In order to analyze interesting associations among a larger number of items for a substantially higher number of transactions, sophisticated techniques are adopted. One such technique is the Apriori technique.

Consider that the Apriori algorithm resulted {I1, I2, I3} and {I1, I2, I5} as the frequent item sets. Valid association rules can be generated based on the obtained frequent item sets as follows:

1. List out all non-empty subset for each of the obtained frequent item sets. Here for the frequent item set {I1, I2, I3}, the non-empty proper subsets are {I1, I2}, {I1, I3}, {I2, I3}, {I1}, {I2}, {I3}.
2. Frame the association rules using the subsets identified. The association rule as {elements of a subset} *→* {elements in frequent item set that is/are not in the chosen subset}. For example, if {I1, I2} is chosen as the subset from the {I1, I2, I3} frequent item set, then the association rule is {I1, I2} *→* {I3}
3. Compute the confidence for each of the association rules framed in the previous step.
4. Select the association rules whose computed confidence is greater than or equal to the min\_confidence threshold.

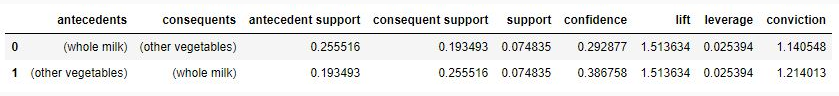
The mlxtend module in Python provides the apriori() method which determines the association rules for a given data set based on the values of min\_support and min\_confidence.

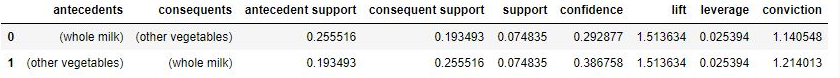
The following code illustrates the usage of the apriori() method of mlxtend to mine association rules.

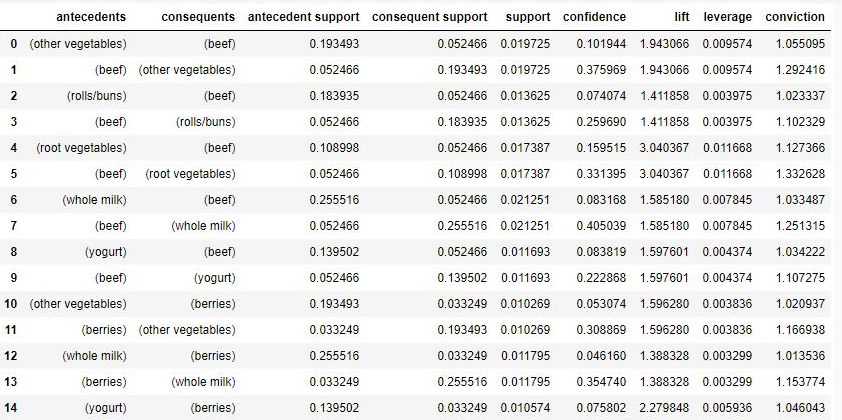
1. from mlxtend.frequent\_patterns import apriori
2. from mlxtend.preprocessing import TransactionEncoder
3. import pandas as pd
4. file = open("Groceries.csv",'r')
5. strings = []
6. for i in file.readlines():
7. strings.append(i.strip().split("\n")) *#get strings as a list of strings*
8. *#print(strings)*
9. dataset = []
10. for i in strings:
11. dataset.append(i[0].split(",")) *# split strings to the list of strings.*
12. *#print(dataset)*
13. te = TransactionEncoder()
14. te\_ary = te.fit(dataset).transform(dataset)
15. df = pd.DataFrame(te\_ary, columns=te.columns\_)
16. frequent\_itemsets = apriori(df, min\_support=0.07, use\_colnames=True)
17. *#print(frequent\_itemsets)*

Uncomment to see the transformation of the data.

1. association\_rules(frequent\_itemsets, metric="confidence", min\_threshold=0.1).head() *#confidence metric*



association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1).head() *#lift metric with higher threshold*



The rules discovered using the Apriori algorithm can be classified as interesting based on domain knowledge. For example, a rule comprising supplementary goods such as pen and paper may not be interesting as the association might seem to be obvious. On the other hand, a rule containing bread and shampoo might be interesting as this association is not so obvious and may reveal an unexpected association.