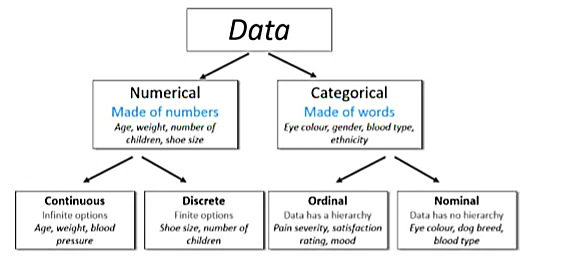
DATA TYPES

1. Useless
2. Nominal
3. Binary
4. Ordinal
5. Count
6. Time
7. Interval

### Useless

Useless data is unique, discrete data with no potential relationship with the outcome variable. A useless feature has high cardinality. If a feature consists of unique values with no order and no meaning, that feature is useless and need not be included when fitting a model. An example would be bank account numbers that were generated randomly.

### Nominal

Nominal data is made of discrete values with no numerical relationship between the different categories — mean and median are meaningless. Animal species is one example. For example, pig is not higher than bird and lower than fish.

You can one-hot-encode or hash nominal features. Do not ordinal encode them because the relationship between the groups cannot be reduced to a monotonic function. The assigning of values would be random.

### Ordinal

Ordinal data is the data which is placed into some kind of order or scale. (Again, this is easy to remember because ordinal sounds like order). An example of ordinal data is rating happiness on a scale of 1-10.

In scale data there is no standardised value for the difference from one score to the next. This can be explained in terms of positions in a race (1st, 2nd, 3rd etc). This is ordinal data because the runners are placed in order of who completed the race in the fastest time to the slowest time, but there is no standardised difference in time between the scores. For example the difference in time between the runners in first place and second place is by no means the same as the difference in time between the runners in second and third place.

Broadly speaking, ordinal data can be encoded in one of three ways.

* It can be assumed to be close enough to interval data, with relatively equal magnitudes between the values, to treat it as such. Social scientists make this assumption all the time with Likert scales.
* A second option is to treat ordinal data as nominal data, where each category has no relationship to any other. One-hot encoding it or a similar scheme can be used in that case.
* A third option that will be explored in more detail in a future article is something like reverse Helmert coding, which can be used to encode various potential magnitudes between the values.

### Binary

Binary data is discrete data that can be in only one of two categories — either yes or no, 1 or 0, off or on, etc. Binary can be thought of as a special case of ordinal, nominal, count, or interval data.

Binary data is a very common outcome variable in machine learning classification problems. For example, we may want to create a supervised learning model to predict whether a tumor is malignant or benign.

### Count

Count data is discrete whole number data — no negative numbers here. Count data often has many small values, such as zero and one.Count data often follows a Poisson distribution.

### Time

Time data is a cyclical, repeating continuous form of data. The relevant time features can be any period— daily, weekly, monthly, annual, etc. Time series data often takes some wrangling and manipulation to create features with the periods that might be meaningful for your model. The Pandas python library was designed with time data in mind. Financial and marketing data often has a time component that is very important to capture in a model.

### Interval

Interval data has equal spaces between the numbers and does not represent a temporal pattern. Examples include percentages, temperatures, and income.

STATISTICS

## →Statistics is a form of mathematical analysis that uses quantified models, representations and synopses for a given set of experimental data or real-life studies.

## →Statistics studies methodologies to gather, review, analyze and draw conclusions from data

**Types**

* Descriptive Statistics
* Inferential Statistics

DESCRIPTIVE STATISTICS

Descriptive statistics uses the data to provide descriptions of the population, either through numerical calculations or graphs or tables. Inferential statistics makes inferences and predictions about a population based on a sample of data taken from the population in question.

MEASURE OF CENTRAL TENDENCY

Once we talk about measuring central tendency of a variable then 3 M’s come into the picture.

1. **Mode**
2. **Median**
3. **Mean**

Mode

If your variable of interest is measured in nominal or ordinal (Categorical) level, then Mode is the most often used technique to measure the central tendency of your data.

Finding the mode is easy. Basically, it is the value that occurs most frequently. In other words, mode is the most common outcome.

Median

The second measure of central tendency is the median. The median is nothing more than the middle value of your observations when they are order from the smallest to the largest.

It involves two steps:

1. Oder your cases from smallest to largest
2. Find the middle Value

* If you have an odd number of cases then finding middle value is easy. Let’s think you have 5 cases. So, after ordering always 3rd position is the middle value.
* If you have even number number. In this case there is no single middle value. Then how do we calculate median Well, we just take the average of the two middle values.

Mean

The third measure of central tendency is the most often used one, and also the one you most probably already know quite well: the mean. The mean is the sum of all the values divided by the number of observations. It is nothing but the average value.

When to use what measurement of central tendency

If data is Categorical (Nominal or Ordinal) it is impossible to calculate mean or median. So, go for mode.

If your data is quantitative then go for mean or median. Basically, if your data is having some influential outliers or data is highly skewed then median is the best measurement for finding central tendency. Otherwise go for Mean.

MEASURES OF VARIABILITY

* **Range**: Difference between the highest and lowest values in a set.

Range =max(xi)-min(xi)

* **Quartiles**: It splits the data into 4 equal parts. Q1, Q2, Q3, Q4

Q1, is the 25th percentile. The second quartile, Q2, is the 50th percentile. The third quartile, Q3, is the 75th percentile. It's important to note that the median is both the 50th percentile and the second quartile, Q2.

* **Deciles**: It splits the data into 10 equal parts

Quartiles sort data into four quarters, deciles sort data into ten equal parts: The 10th, 20th, 30th, 40th, 50th, 60th, 70th, 80th, 90th and 100th percentiles.

* **Percentile**: It splits the data into 100 equal parts

A percentile is simply a measure that tells us what percent of the total frequency of a data set was at or below that measure.

* **Inter Quartile Range**: It is the measure of variability, based on dividing a dataset into quartiles. IQR=Q3-Q1
* **Variance**
* **Standard** **division**

Variance

Variance is the measure of dispersion in a data set. In other words, it measures how spread out a data set is. It is calculated by first finding the deviation of each element in the data set from the mean, and then by squaring it.

**Variance is the average of all squared deviations.**

### Steps to Calculate Variance:

1. **List elements of data set.**The following are ages of students pursuing a Master’s degree:  
   Data set 1: 28,25,26,27,31,32,24
2. **Calculate the mean.**(28 + 25 +26 +27 +31 +32 + 24) / 7 = 27.57
3. **Find the deviation from the mean for each data point.**

28 --> 28- 27.57

25 --> 25- 27.57

26 --> 26- 27.57 etc.

1. **Square it. (x-X)^2**
2. **The average of all squared differences is the variance. To find it, add all squared variances and divide the sum by a number of elements in data set (n).**(0.1849 + 6.6049 + 2.4649 + .3249 + 11.76 + 19.6249 + 12. 4609) / 7  **= 7.632 (Ans)**

Mean Absolute Deviation (MAD)

The Mean Absolute Deviation (MAD) of a set of data is the average distance between each data value and the mean. The mean absolute deviation is the "average" of the "positive distances" of each point from the mean. The larger the MAD, the greater variability there is in the data (the data is more spread out). The MAD helps determine whether the set's mean is a useful indicator of the values within the set. The larger the MAD, the less relevant is the mean as an indicator of the values within the set.

**MAD=(∑|x- |) / N**

Standard Deviation

Standard deviation tells about the concentration of the data around the mean of the data set. Standard deviation is inversely proportional to the concentration of the data around the mean i.e with high concentration, the standard deviation will be low, and vice versa. It cannot be negative. The value of standard deviation can be easily impacted by outliers as a single outlier (abnormal value) distorts the overall mean, and thereby, deviation from the mean of all elements.

|  |  |
| --- | --- |
|  |  |

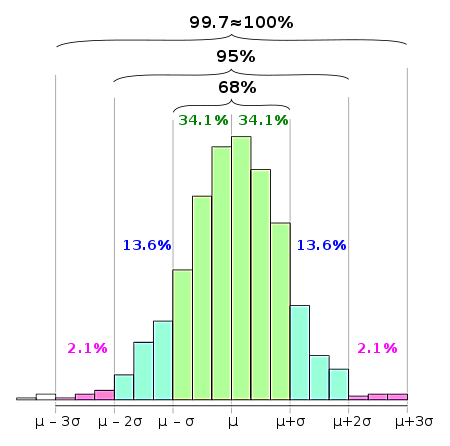
Coefficient of variation

The coefficient of variation (CV) is a statistical measure of the dispersion of data points in a data series around the mean. The coefficient of variation represents the ratio of the standard deviation to the mean, and it is a useful statistic for comparing the degree of variation from one data series to another, even if the means are drastically different from one another.

For example, it will help scenarios where 85/100 is good or 55/75?

|  |  |
| --- | --- |
| C:\Users\Vinsmon TP\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture1.png | Where: σ is the standard deviation and μ is the mean. |

68–95–99.7 rule

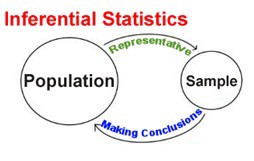


For an approximately normal data set, the values within one standard deviation of the mean account for about 68% of the set; while within two standard deviations account for about 95%; and within three standard deviations account for about 99.7%. Shown percentages are rounded theoretical probabilities intended only to approximate the empirical data derived from a normal population. This rule is generally taught within common core mathematics.

INFERENTIAL STATISTICS

Inferential statistics helps us answer the following questions:

* Making inferences about a population from a sample
* Concluding whether a sample is significantly different from the population. Let’s look at the previous example where I pointed out that the sample is different from the population as the children are more interested in sports rather than watching television.
* If adding or removing a feature from a model will help in improving it.
* If one model is significantly different from the other.
* Hypothesis Testing.



* **Statistic**– A Single measure of some attribute of a sample. For eg: Mean/Median/Mode of a sample of Data Scientists in Bangalore.
* **Population Statistic**– The statistic of the entire population in context. For eg: Population mean for the salary of the entire population of Data Scientists across India.
* **Sample Statistic** – The statistic of a group taken from a population. For eg: Mean of salaries of all Data Scientists in Bangalore.
* **Standard Deviation**– It is the amount of variation in the population data. It is given by σ.
* **Standard Error**– It is the amount of variation in the sample data. It is related to Standard Deviation as σ/√n, where, n is the sample size.

**Population & Sample**

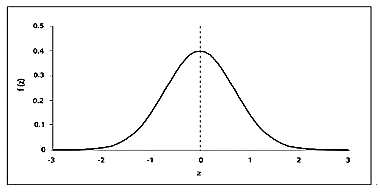
A sample is a representative subset of a population. Conducting a census on population is an ideal but impractical approach in most of the cases. Sampling is much more practical, however it is prone to sampling error. A sample non-representative of population is called bias, method chosen for such sampling is called sampling bias. Convenience bias, judgement bias, size bias, response bias are main types of sampling bias. The best technique for reducing bias in sampling is randomization. Simple random sampling is the simplest of randomization techniques, cluster sampling & stratified sampling are other systematic sampling techniques.

**Sampling Distributions**

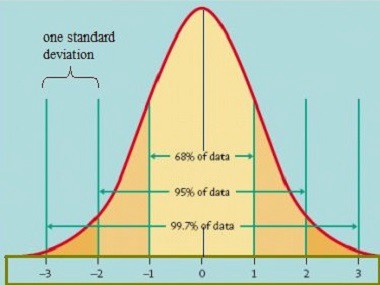
Sample means become more and more normally distributed around the true mean (the population parameter) as we increase our sample size. The variability of the sample means decreases as sample size increases.

Normal distribution and z — statistic

The normal distribution is also known as the bell curve having the following properties:



1. mean = median = mode.
2. The curve is symmetric with half of the values on the left and half of the values on the right.
3. The area under the curve is 1.



In a normal distribution:

* 68% of the data falls within one standard deviation of the mean
* 95% of the data falls within two standard deviations of the mean
* 99.7 % of the data falls within three standard deviations of the mean.

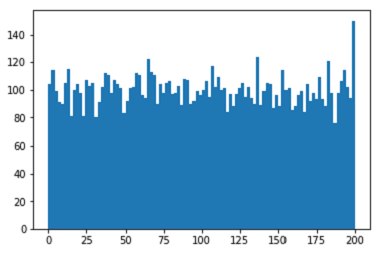
For calculating the probability of occurrence of an event we need the z — statistic. The formula for calculating the z — statistic is

https://miro.medium.com/max/141/1*twdboHPSJiaMJU_M68Odxg.png

where x is the value for which we want to calculate the z — value. μ and σ are the population mean and standard deviation respectively. Basically what we are doing here is standardizing the normal curve by moving the mean to 0 and converting the standard deviation to 1. The z — statistic is essentially the distance of the value from the mean calculated in standard deviation terms. So a z — value of 1.67 means that the value is 1.67 standard deviations away from the mean in the positive direction. We then find the probability by looking up the corresponding z — value from the z table.

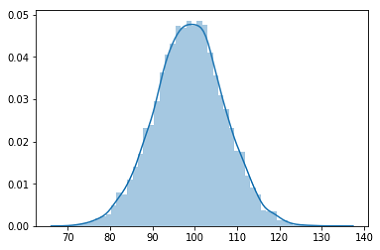
Central Limit Theorem

The main importance of the normal distribution comes from the central limit theorem. The central limit theorem states that the sampling distribution of the mean of sample means approaches the normal distribution as the sample size gets larger no matter the shape of the population distribution. Let’s look at this through an example. Consider the case where we look at the number of tweets a person makes in a week (randomly generated data between 0 and 200). The frequency distribution of the data looks like this:



This is not similar to any kind of distribution we know.

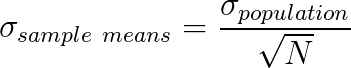
Now lets take 1000 random samples of size 50 from this data and calculate the mean of each sample. When we plot these means we get a normal distribution curve also known as the **sampling curve**or the **sampling distribution**.



**Mean = 98.78 (population mean = 98.87)**

The central limit theorem has some important properties:

1. The mean of the population is approximately equal to the mean of the sampling distribution. We can see this in the example above where population mean( 98.87 ) is approximately equal to the mean of sampling distribution (98.78).
2. The standard deviation of the sampling distribution also known as the **standard error**is equal to the population standard deviation divided by the square root of the sample size. As a result the greater the sample size, the lower the standard deviation and greater accuracy in determining the sample mean from the population mean.



1. The distribution of sample means is normal regardless of the shape of the population distribution. It means that even if our original distribution is skewed or bimodal or some other distribution the mean of sample means is always a normal distribution. This is what makes the central limit theorem so powerful.

For the central limit theorem to hold the sample size should be sufficiently large (generally > 30)

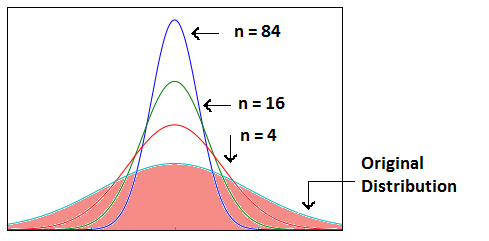
It states that when plotting a sampling distribution of means, the mean of sample means will be equal to the population mean. And the sampling distribution will approach a normal distribution with variance equal to σ/√n where σ is the standard deviation of population and n is the sample size.

Points to note:

1. Central Limit Theorem holds true irrespective of the type of distribution of the population.
2. Now, we have a way to estimate the population mean by just making repeated observations of samples of a fixed size.
3. Greater the sample size, lower the standard error and greater the accuracy in determining the population mean from the sample mean.

This seemed too technical isn’t it? Let’s break this down to understand this point by point.

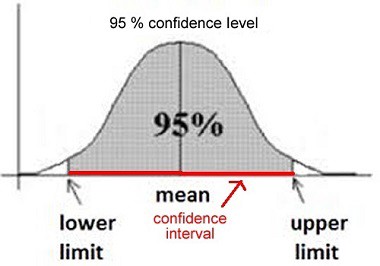
1. This means – No matter the shape of the population distribution, be it bi-modal, right skewed etc. The shape of the Sampling Distribution will remain the same (remember the normal curve- bell shaped). This gives us a mathematical advantage to estimate the population statistic – no matter the shape of the population.
2. The number of samples have to be sufficient (generally more than 50) to satisfactorily achieve a normal curve distribution. Also, care has to be taken to keep the sample size fixed since any change in sample size will change the shape of the sampling distribution and it will no longer be bell shaped.
3. As we increase the sample size, the sampling distribution squeezes from both sides giving us a better estimate of the population statistic since it lies somewhere in the middle of the sampling distribution (generally). The below image will help you visualize the effect of sample size on the shape of distribution.



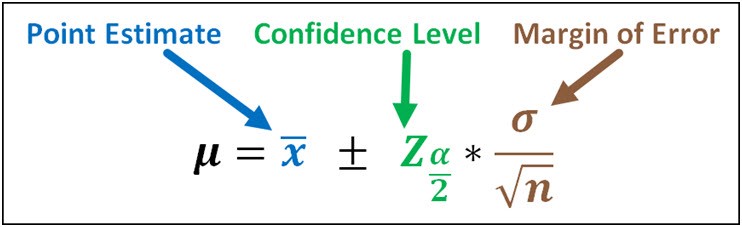
Now, since we have collected the samples and plotted their means, it is important to know where the population mean lies with respect to a particular sample mean and how confident can we be about it. This brings us to our next topic – **Confidence Interval.**

Confidence Interval

Like I have previously said we find the sample mean and would like to estimate the population mean. How well the sample statistics estimates the underlying population is always an issue. A confidence interval addresses this issue as it provides a range of values which may likely contain the population parameter.



There are one sided or two sided confidence intervals. In two sided confidence interval, if we are choosing a 95% confidence interval, we choose 2.5% on each side of the tail and then calculate the range. In a one sided confidence interval we calculate the confidence interval by taking the entire 5% either to the left or to the right of the distribution. The above image displays a two sided confidence interval. The formula we use to calculate the confidence interval is:



where the symbols stand for

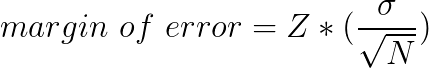
https://miro.medium.com/max/26/1*rn6YmWEE5SVqNJYM0Eo7wg.png **Sample Mean**

https://miro.medium.com/max/59/1*Yiv3RJ96b9MhajzxgCg4Rg.png **Z value for the desired confidence level**

https://miro.medium.com/max/19/1*qYL4tFpfNgTDEqqHAcS44A.png **confidence level**

https://miro.medium.com/max/18/1*753-QA-tQjJSqCa14hGNZw.png **the standard deviation of the population**

Another important term in the confidence interval concept is the **margin of error**. It is half the size of a confidence interval. It is known as the sampling error and it means that if the sample mean is within the margin of error then its actual value is the population mean and the difference only occurs due to chance. Otherwise the results are considered to be statistically significant. We will revisit this concept in a later tutorial.



When we take a 95% confidence interval it does not mean that the population means is within the range we derive with a 95% chance. The confidence interval represents the frequency (i.e. proportion) of possible confidence intervals that contain the true value of the unknown population parameter. So if we take infinitely many samples and find the confidence interval range for each of these samples then the number of intervals that contain the population parameter is equal to the confidence interval. If we take a 95% confidence interval then it means that the population parameter is present in 95% of all possible confidence interval ranges.

Interesting points to note about Confidence Intervals:

1. Confidence Intervals can be built with difference degrees of confidence suitable to a user’s needs like 70 %, 90% etc.
2. Greater the sample size, smaller the Confidence Interval, i.e more accurate determination of population mean from the sample means.
3. There are different confidence intervals for different sample means. For example, a sample mean of 40 will have a difference confidence interval from a sample mean of 45.
4. By 95% Confidence Interval, we do not mean that – The probability of a population mean to lie in an interval is 95%. Instead, 95% C.I means that 95% of the Interval estimates will contain the population statistic.

Many people do not have right knowledge about confidence interval and often interpret it incorrectly. So, I would like you to take your time visualizing the 4th argument and let it sink in.

Practical example

Calculate the 95% confidence interval for a sample mean of 40 and sample standard deviation of 40 with sample size equal to 100.

**Solution:**

We know, z-value for 95% C.I is 1.96. Hence, Confidence Interval (C.I) is calculated as:

C.I= [{x(bar) – (z\*s/√n)},{x(bar) – (z\*s/√n)}]

C.I = [{40-(1.96\*40/10},{ 40+(1.96\*40/10)}]

C.I = [32.16, 47.84]

HYPOTHESIS TEST

The goal of hypothesis testing is to determine the likelihood that a population parameter, such as the mean, is likely to be true.

1. Step 1: State the hypotheses.
2. Step 2: Set the criteria for a decision.
3. Step 3: Compute the test statistic.
4. Step 4: Make a decision.

Step 1: State the hypotheses

We begin by stating the value of a population mean in a null hypothesis, which we presume is true. This is a starting point so that we can decide whether this is likely to be true, similar to the presumption of innocence in a courtroom. When a defendant is on trial, the jury starts by assuming that the defendant is innocent. The basis of the decision is to determine whether this assumption is true. Likewise, in hypothesis testing, we start by assuming that the hypothesis or claim we are testing is true. This is stated in the null hypothesis. The basis of the decision is to determine whether this assumption is likely to be true.

The null hypothesis (H0), stated as the null, is a statement about a population parameter, such as the population mean, that is assumed to be true. The null hypothesis is a starting point. We will test whether the value stated in the null hypothesis is likely to be true.

Keep in mind that the only reason we are testing the null hypothesis is because we think it is wrong. We state what we think is wrong about the null hypothesis in an alternative hypothesis. In a courtroom, since the defendant is assumed to be innocent (this is the null hypothesis so to speak), the burden is on a prosecutor to conduct a trial to show evidence that the defendant is not innocent. In a similar way, we assume the null hypothesis is true, placing the burden on the researcher to conduct a study to show evidence that the null hypothesis is unlikely to be true. Regardless, we always make a decision about the null hypothesis (that it is likely or unlikely to be true). The alternative hypothesis is needed for Step 2.

An alternative hypothesis (H1) is a statement that directly contradicts a null hypothesis by stating that that the actual value of a population parameter is less than, greater than, or not equal to the value stated in the null hypothesis. The alternative hypothesis states what we think is wrong about the null hypothesis, which is needed for Step 2.

Step 2: Set the criteria for a decision

To set the criteria for a decision, we state the level of significance for a test. This is similar to the criterion that jurors use in a criminal trial. Jurors decide whether the evidence presented shows guilt beyond a reasonable doubt (this is the criterion). Likewise, in hypothesis testing, we collect data to show that the null hypothesis is not true, based on the likelihood of selecting a sample mean from a population (the likelihood is the criterion). The likelihood or level of significance is typically set at 5% in behavioral research studies. When the probability of obtaining a sample mean is less than 5% if the null hypothesis were true, then we conclude that the sample we selected is too unlikely and so we reject the null hypothesis.

**Level of significance, or significance level,** refers to a criterion of judgment upon which a decision is made regarding the value stated in a null hypothesis. The criterion is based on the probability of obtaining a statistic measured in a sample if the value stated in the null hypothesis were true. In behavioral science, the criterion or level of significance is typically set at 5%. When the probability of obtaining a sample mean is less than 5% if the null hypothesis were true, then we reject the value stated in the null hypothesis.

The alternative hypothesis establishes where to place the level of significance. Remember that we know the sample mean will equal the population mean on average if the null hypothesis is true. All other possible values of the sample mean are normally distributed (central limit theorem). The empirical rule tells us that at least 95% of all sample means fall within about 2 standard deviations (SD) of the population mean, meaning that there is less than a 5% probability of obtaining a sample mean that is beyond 2 SD from the population mean.

Step 3: Compute the test statistic

Specifically, a test statistic tells us how far, or how many standard deviations, a sample mean is from the population mean. The larger the value of the test statistic, the further the distance, or number of standard deviations, a sample mean is from the population mean stated in the null hypothesis. The value of the test statistic is used to make a decision in Step 4.

The test statistic is a mathematical formula that allows researchers to determine the likelihood of obtaining sample outcomes if the null hypothesis were true. The value of the test statistic is used to make a decision regarding the null hypothesis.

Step 4: Make a decision

We use the value of the test statistic to make a decision about the null hypothesis. The decision is based on the probability of obtaining a sample mean, given that the value stated in the null hypothesis is true. If the probability of obtaining a sample mean is less than 5% when the null hypothesis is true, then the decision is to reject the null hypothesis. If the probability of obtaining a sample mean is greater than 5% when the null hypothesis is true, then the decision is to retain the null hypothesis. In sum, there are two decisions a researcher can make:

1. Reject the null hypothesis. The sample mean is associated with a low probability of occurrence when the null hypothesis is true.

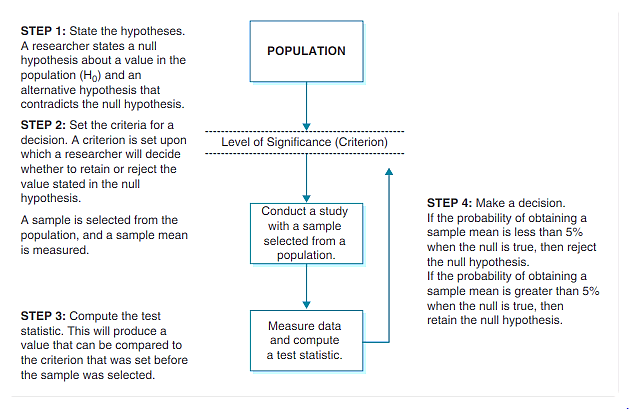
2. Retain the null hypothesis. The sample mean is associated with a high probability of occurrence when the null hypothesis is true.

The probability of obtaining a sample mean, given that the value stated in the null hypothesis is true, is stated by the p value. The p value is a probability: It varies between 0 and 1 and can never be negative. In Step 2, we stated the criterion or probability of obtaining a sample mean at which point we will decide to reject the value stated in the null hypothesis, which is typically set at 5% in behavioral research. To make a decision, we compare the p value to the criterion we set in Step 2.

A **p value** is the probability of obtaining a sample outcome, given that the value stated in the null hypothesis is true. The p value for obtaining a sample outcome is compared to the level of significance.

Significance, or statistical significance, describes a decision made concerning a value stated in the null hypothesis. When the null hypothesis is rejected, we reach significance. When the null hypothesis is retained, we fail to reach significance.

When the p value is less than 5% (p < .05), we reject the null hypothesis. We will refer to p < .05 as the criterion for deciding to reject the null hypothesis, although note that when p = .05, the decision is also to reject the null hypothesis. When the p value is greater than 5% (p > .05), we retain the null hypothesis. The decision to reject or retain the null hypothesis is called significance. When the p value is less than .05, we reach significance; the decision is to reject the null hypothesis. When the p value is greater than .05, we fail to reach significance; the decision is to retain the null hypothesis.

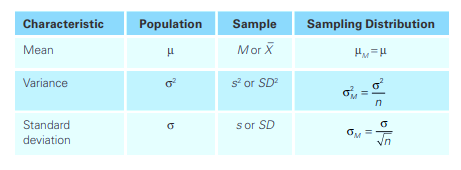


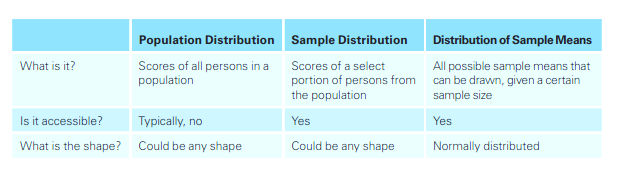
HYPOTHESIS TESTING AND SAMPLING DISTRIBUTIONS

The logic of hypothesis testing is rooted in an understanding of the sampling distribution of the mean.

1. The sample mean is an unbiased estimator of the population mean. On average, a randomly selected sample will have a mean equal to that in the population. In hypothesis testing, we begin by stating the null hypothesis. We expect that, if the null hypothesis is true, then a random sample selected from a given population will have a sample mean equal to the value stated in the null hypothesis.
2. Regardless of the distribution in the population, the sampling distribution of the sample mean is normally distributed. Hence, the probabilities of all other possible sample means we could select are normally distributed. Using this distribution, we can therefore state an alternative hypothesis to locate the probability of obtaining sample means with less than a 5% chance of being selected if the value stated in the null hypothesis is true.

To locate the probability of obtaining a sample mean in a sampling distribution, we must know (1) the population mean and (2) the standard error of the mean. Each value is entered in the test statistic formula computed in Step 3, thereby allowing us to make a decision in Step 4.





MAKING A DECISION: TYPES OF ERROR

DECISION: RETAIN THE NULL HYPOTHESIS

When we decide to retain the null hypothesis, we can be correct or incorrect. The correct decision is to retain a true null hypothesis. This decision is called a null result or null finding. This is usually an uninteresting decision because the decision is to retain what we already assumed: that the value stated in the null hypothesis is correct. For this reason, null results alone are rarely published in behavioral research.

The incorrect decision is to retain a false null hypothesis. This decision is an example of a Type II error, or b error. With each test we make, there is always some probability that the decision could be a Type II error. In this decision, we decide to retain previous notions of truth that are in fact false. While it’s an error, we still did nothing; we retained the null hypothesis. We can always go back and conduct more studies.

Type II error, or beta (b) error, is the probability of retaining a null hypothesis that is actually false.

DECISION: REJECT THE NULL HYPOTHESIS

When we decide to reject the null hypothesis, we can be correct or incorrect. The incorrect decision is to reject a true null hypothesis. This decision is an example of a Type I error. With each test we make, there is always some probability that our decision is a Type I error. A researcher who makes this error decides to reject previous notions of truth that are in fact true. Making this type of error is analogous to finding an innocent person guilty. To minimize this error, we assume a defendant is innocent when beginning a trial. Similarly, to minimize making a Type I error, we assume the null hypothesis is true when beginning a hypothesis test.

Type I error is the probability of rejecting a null hypothesis that is actually true. Researchers directly control for the probability of committing this type of error.

An alpha (a) level is the level of significance or criterion for a hypothesis test. It is the largest probability of committing a Type I error that we will allow and still decide to reject the null hypothesis.

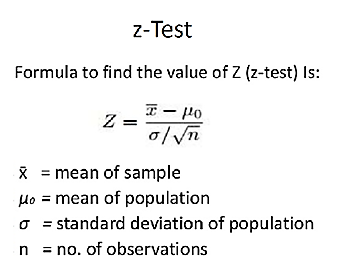
Since we assume the null hypothesis is true, we control for Type I error by stating a level of significance. The level we set, called the alpha level (symbolized as a), is the largest probability of committing a Type I error that we will allow and still decide to reject the null hypothesis. This criterion is usually set at .05 (a = .05), and we compare the alpha level to the p value. When the probability of a Type I error is less than 5% (p < .05), we decide to reject the null hypothesis; otherwise, we retain the null hypothesis.

The correct decision is to reject a false null hypothesis. There is always some probability that we decide that the null hypothesis is false when it is indeed false. This decision is called the power of the decision-making process. It is called power because it is the decision we aim for. Remember that we are only testing the null hypothesis because we think it is wrong. Deciding to reject a false null hypothesis, then, is the power, inasmuch as we learn the most about populations when we accurately reject false notions of truth. This decision is the most published result in behavioral research.

The power in hypothesis testing is the probability of rejecting a false null hypothesis. Specifically, it is the probability that a randomly selected sample will show that the null hypothesis is false when the null hypothesis is indeed false.

**HYPOTHESIS TEST ON ONE SAMPLE MEAN WHEN THE POPULATION PARAMETERS ARE KNOWN**

find the z-statistic of our sample mean in the sampling distribution and determine if that z-score falls within the critical(rejection) region or not. This test is only appropriate when you know the true mean and standard deviation of the population.



**NONDIRECTIONAL - TWO-TAILED HYPOTHESIS TESTS (H1: ≠)**

we will use the z test for a nondirectional, or two-tailed test, where the alternative hypothesis is stated as not equal to (≠) the null hypothesis. For this test, we will place the level of significance in both tails of the sampling distribution. We are therefore interested in any alternative from the null hypothesis. This is the most common alternative hypothesis tested in behavioral science.

**Example**: Templer and Tomeo (2002) reported that the population mean score on the quantitative portion of the Graduate Record Examination (GRE) General Test for students taking the exam between 1994 and 1997 was 558 ± 139 (m ± s). Suppose we select a sample of 100 participants (n = 100). We record a sample mean equal to 585 (M = 585). Compute the one–independent sample z test for whether or not we will retain the null hypothesis (µ = 558) at a .05 level of significance (α = .05).

**Solution**

**Step 1**: State the hypotheses. The population mean is 558, and we are testing whether the null hypothesis is (=) or is not (≠) correct:

H0: µ = 558 Mean test scores are equal to 558 in the population.

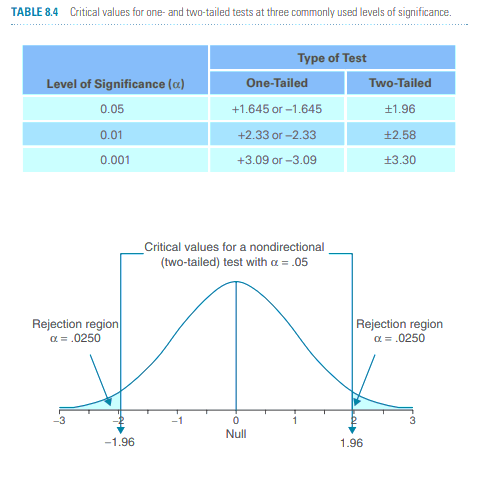
H1: µ ≠ 558 Mean test scores are not equal to 558 in the population.

**Step 2:** Set the criteria for a decision. The level of significance is .05, which makes the alpha level α = .05. To locate the probability of obtaining a sample mean from a given population, we use the standard normal distribution. We will locate the z scores in a standard normal distribution that are the cutoffs, or critical values, for sample mean values with less than a 5% probability of occurrence if the value stated in the null (µ = 558) is true.

In a nondirectional two­tailed test, we divide the alpha value in half so that an equal proportion of area is placed in the upper and lower tail. Table 8.4 gives the critical values for one­ and two­tailed tests at a .05, .01, and .001 level of significance. Figure 8.4 displays a graph with the critical values for Example 8.1 shown. In this example α = .05, so we split this probability in half.

Splitting in half = = .025 in each tail

.0250 is listed from the z table for a z-score equal to z = 1.96. This is the critical value for the upper tail of the standard normal distribution. Since the normal distribution is symmetrical, the critical value in the bottom tail will be the same distance below the mean, or z = –1.96. The regions beyond the critical values, displayed in Figure 8.4, are called the rejection regions. If the value of the test statistic falls in these regions, then the decision is to reject the null hypothesis; otherwise, we retain the null hypothesis.



**Step 3**: Compute the test statistic. Step 2 sets the stage for making a decision because the criterion is set. The probability is less than 5% that we will obtain a sample mean that is at least 1.96 standard deviations above or below the value of the population mean stated in the null hypothesis. In this step, we will compute a test statistic to determine whether the sample mean we selected is beyond or within the critical values we stated in Step 2. The test statistic for a one–independent sample z test is called the z statistic. The z statistic converts any sampling distribution into a standard normal distribution. The z statistic is therefore a z transformation. The solution of the formula gives the number of standard deviations, or z-scores, that a sample mean falls above or below the population mean stated in the null hypothesis. We can then compare the value of the z statistic, called the obtained value, to the critical values we determined in Step 2. The z statistic formula is the sample mean minus the population mean stated in the null hypothesis, divided by the standard error of the mean:



The z statistic is an inferential statistic used to determine the number of standard deviations in a standard normal distribution that a sample mean deviates from the population mean stated in the null hypothesis.

The obtained value is the value of a test statistic. This value is compared to the critical value(s) of a hypothesis test to make a decision. When the obtained value exceeds a critical value, we decide to reject the null hypothesis; otherwise, we retain the null hypothesis.

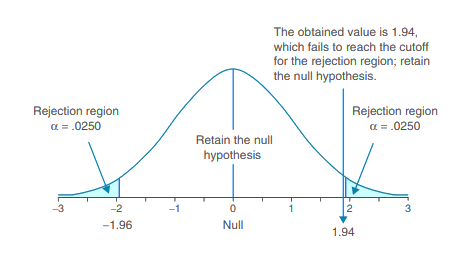
To calculate the z statistic, first compute the standard error (), which is the denominator for the z statistic:



Then compute the z statistic by substituting the values of the sample mean, M = 585; the population mean stated by the null hypothesis, µ = 558; and the standard error we just calculated, = 13.9

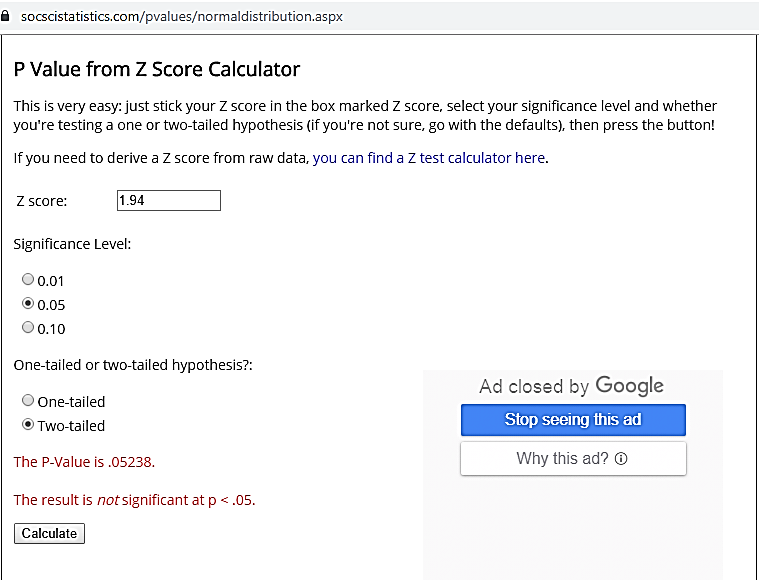


Step 4: Make a decision. To make a decision, we compare the obtained value to the critical values. We reject the null hypothesis if the obtained value exceeds a critical value. Figure below shows that the obtained value (Zobt = 1.94) is less than the critical value; it does not fall in the rejection region. **The decision is to retain the null hypothesis.**



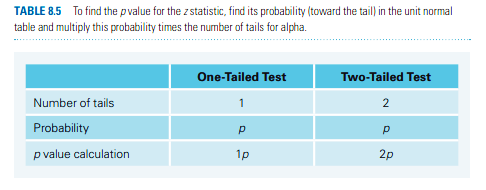
**Using P-Values**

***Software***



**From Table**

To locate the p value or probability of obtaining the z statistic, we refer to the unit normal table in Table B1 in Appendix B. Look for a z score equal to 1.94 in column A, then locate the probability toward the tail in column C. The value is .0262.



Finally, multiply the value given in column C times the number of tails for alpha. Since this is a two tailed test, we multiply .0262 times 2: p = (.0262) × 2 tails = .0524.

**Since p-value is > .05 (.0524 > .05) we cannot reject null hypothesis.**

DIRECTIONAL, UPPER-TAIL CRITICAL HYPOTHESIS TESTS (H1: >)

In previous Example, we will use the z test for a directional, or one-tailed test, where the alternative hypothesis is stated as greater than (>) the null hypothesis. A directional test can also be stated as less than (<) the null hypothesis (an example for this alternative is given in Example 8.3). For an upper­tail critical test, or a greater than statement, we place the level of significance in the upper tail of the sampling distribution. So we are interested in any alternative greater than the value stated in the null hypothesis. This test is appropriate when it is not possible or highly unlikely that a sample mean will fall below the population mean stated in the null hypothesis.

Directional tests, or one-tailed tests, are hypothesis tests where the alternative hypothesis is stated as greater than (>) or less than (<) a value stated in the null hypothesis. Hence, the researcher is interested in a specific alternative from the null hypothesis.

Using the same study from Example, Templer and Tomeo (2002) reported that the population mean on the quantitative portion of the GRE General Test for students taking the exam between 1994 and 1997 was 558 ± 139 (m ± s). Suppose we select a sample of 100 students enrolled in an elite private school (n = 100). We hypothesize that students at this elite school will score higher than the general population. We record a sample mean equal to 585 (M = 585), same as measured in previous Example. Compute the one–independent sample z test at a .05 level of significance.

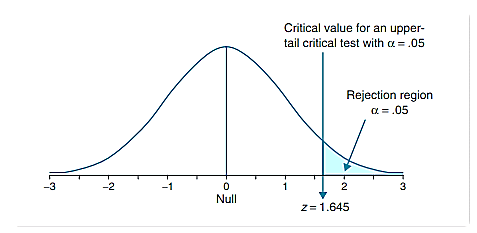
**Step 1**: State the hypotheses. The population mean is 558, and we are testing whether the alternative is greater than (>) this value:

H0: m = 558 Mean test scores are equal to 558 in the population of students at the elite school.

H1: m > 558 Mean test scores are greater than 558 in the population of students at the elite school.

**Step 2**: Set the criteria for a decision. The level of significance is .05, which makes the alpha level a = .05. To determine the critical value for an upper-tail critical test, we locate the probability .0500 toward the tail in column C in the unit normal table. The z-score associated with this probability is between z = 1.64 and z = 1.65. The average of these z-scores is z = 1.645. This is the critical value or cutoff for the rejection region.

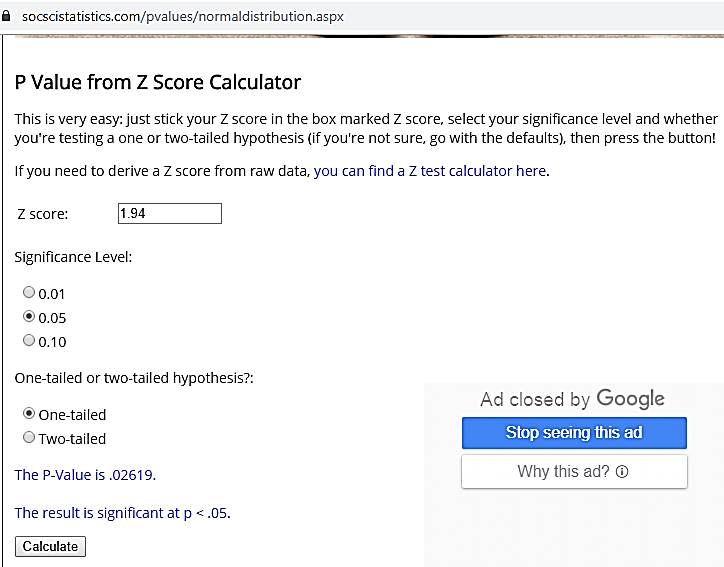
**Step 3**: Compute the test statistic. Step 2 sets the stage for making a decision because the criterion is set. The probability is less than 5% that we will obtain a sample mean that is at least 1.645 standard deviations above the value of the population mean stated in the null hypothesis. In this step, we will compute a test statistic to determine whether or not the sample mean we selected is beyond the critical value we stated in Step 2.



The test statistic does not change from that in Example 8.1. We are testing the same population, and we measured the same value of the sample mean. We changed only the location of the rejection region in Step 2.



Step 4: Make a decision. To make a decision, we compare the obtained value to the critical value. We reject the null hypothesis if the obtained value exceeds the critical value. Figure 8.7 shows that the obtained value (Zobt = 1.94) is greater than the critical value; it falls in the rejection region. The decision is to reject the null hypothesis. The p value for this test is .0262 (p = .0262). We do not double the p value for one-tailed tests. We found in Example that if the null hypothesis were true, then p = .0262 that we could have selected this sample mean from this population. The criteria we set in Step 2 was that the probability must be less than 5% that we obtain a sample mean, if the null hypothesis were true. Since p is less than 5%, we decide to reject the null hypothesis. We decide that the mean score on the GRE General Test in this population is not 558, which was the value stated in the null hypothesis.



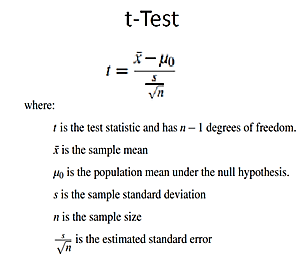
HYPOTHESIS TEST WHEN DON’T KNOW YOUR POPULATION PARAMETERS

The Student’s t-distribution is similar to the normal distribution, except it is more spread out and wider in appearance, and has thicker tails. The differences between the t-distribution and the normal distribution are more exaggerated when there are fewer data points, and therefore fewer degrees of freedom.

**Degrees of freedom** refers to the values in a study that has the freedom to vary and are essential for assessing the importance and the validity of the null hypothesis. Computation of these values usually depends upon the number of data records available in the sample set.

**ONE SAMPLE T-TEST**

The One Sample t Test determines whether the sample mean is statistically different from a known or hypothesized population mean. The One Sample t Test is a parametric test.



1 Sample T Test in following steps

* State Hypothesis:
* Compute T Test Statistics
* Compute Critical value using T table (For Two Tailed Test We need to find the critical cut-off value)
* Evaluate null hypothesis

**TWO TAILED ONE SAMPLE T TEST**

**Example Problem**

A coffee shop relocates to Italy and wants to make sure that all lattes are consistent. They believe that each latte has an average of 4 Oz of espresso. If this is not the case, they must increase or decrease the amount. A random sample of 25 lattes shows a mean of 4.6 Oz of espresso and a standard deviation of .22 Oz. Use alpha = .05 and run a one sample t-test to compare with the known population mean.

**1. Let’s State Hypothesis:**

Null Hypothesis H0: There is no significant difference between sample Mean (M )of espresso in latte and population means μ

M= μ=4 oz.

Alternative Hypothesis HA: There is significant difference between sample Mean, M and population means μ. So the sample mean of espresso in latte is not equal to 4 oz.

The purpose of the one sample t-test is to determine if the null hypothesis should be rejected, given the sample data. Here we will perform two tailed (or two sided) one sample t test.

Remember! We are performing a two tailed t test here because we are not trying to find whether the mean value is less than or greater than any given value.

**2. Compute T test Statistics:**

T test = (sample mean — population mean)/(stddev/sqrt(n))

Here are the known values given in the example:

Sample Mean M = 4.6 oz

Population Mean μ = 4 oz.

Sample standard deviation = 0.22 oz.

Sample size n = 25

So we can calculate,

Degree Of Freedom, df = Sample size -1 = 25–1 = 24.

So if we replace all the known values in t-test formula

T test = (4.6–4)/(0.22/sqrt(25)) = 13.6

So our t test value comes out to be 13.6.

**3. Compute Critical Value for Two Tailed One Sample Test:**

Know that we know the t test value = 13.6

Degree of freedom df = 24,

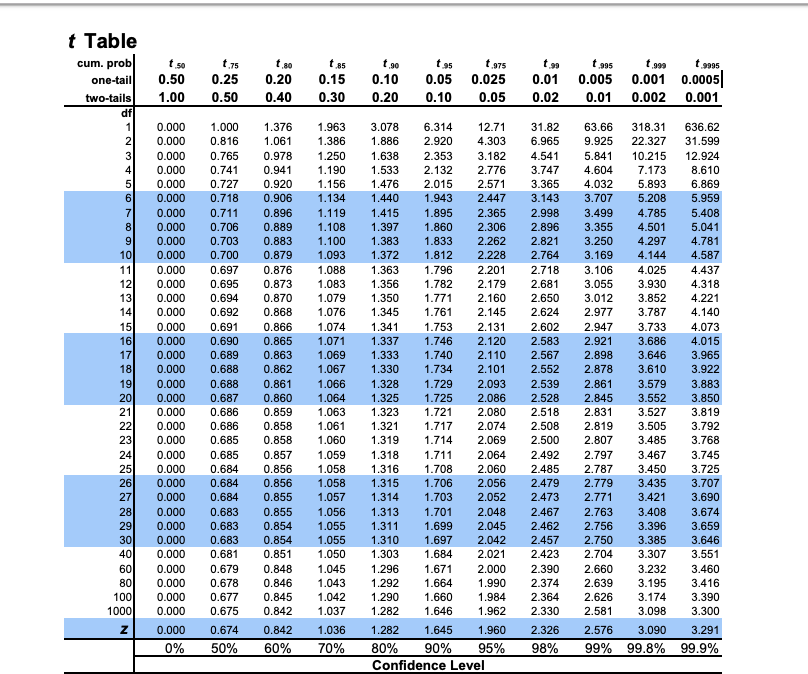
given, alpha value which is .05,

We can refer to T table to find the critical values(tc) for the two tailed test.

In below given t table, go df row no 24 (as df =24), navigate to t.975 column where we have alpha value of 0.05 for 2 tailed test, you will get the cut off value to be

tc = +/- 2.064,

Which is a cutt-off point for rejecting the null hypothesis.



**4. Accepting / Rejecting our Null Hypothesis: Using The region of acceptance method.**

Since our t-test result is 13.6 which is either bigger than 2.064 or less than -2.064, comes out to be in the rejection region being > 2.064, so we can conclude that there is a significant difference between our sample mean of the amount of espresso in the coffee in Italy and the expected population amount.

Hence we dis-agree with our Null hypothesis which stated that there is no significant difference between sample Mean (M )of espresso in latte and population means μ

So we reject Ho here & accept HA, our alternative hypothesis.

Therefore, we can easily say that there is too much espresso being placed in the coffee in Italy and it should be reduced to meet the normal (population) mean.

TWO-SAMPLE T-TEST

**Tests weather the means of two populations are significantly different from one another.**

**PAIRED**

Each value of one group corresponds directly to a value in the other group, before and after values in an experiment. Subtract two values and perform a one-sample t-test with null mean set to 0.

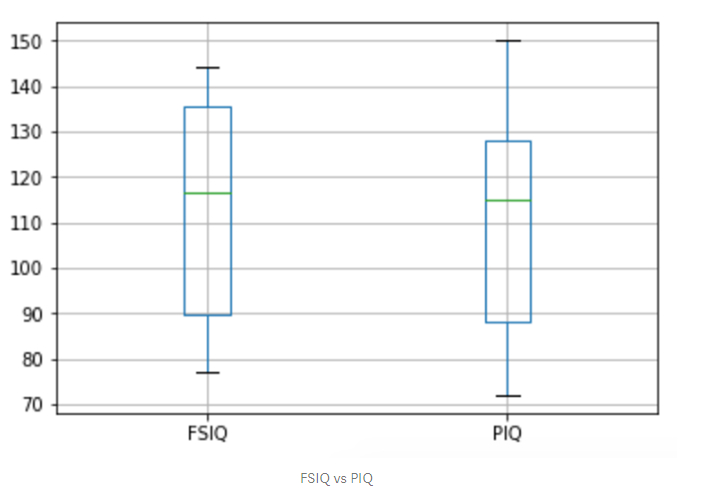
The paired sample t-test is also called dependent sample t-test. It’s an uni variate test that tests for a significant difference between 2 related variables. An example of this is if you where to collect the blood pressure for an individual before and after some treatment, condition, or time point.

**Wilcoxon signed-rank test** can also be used when the **population cannot be assumed normal**.

**Example :1**

import pandas as pd  
from scipy import stats  
df = pd.read\_csv("blood\_pressure.csv")df[['bp\_before','bp\_after']].describe()ttest,pval = stats.ttest\_rel(df['bp\_before'], df['bp\_after'])  
print(pval)if pval<0.05:  
 print("reject null hypothesis")  
else:  
 print("accept null hypothesis")

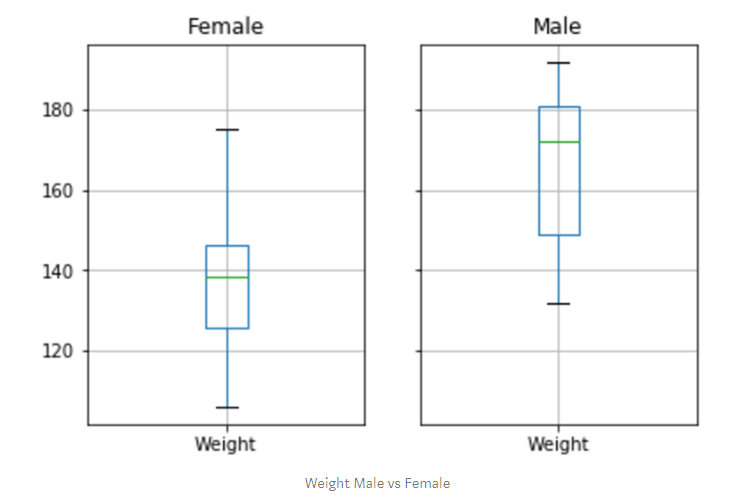
**Example :2**



|  |
| --- |
|  |
|  | data.boxplot(column=['FSIQ','PIQ']) |
|  | stats.ttest\_rel(data['FSIQ'], data['PIQ']) |
|  | # paired, pvalue > 0.05, accept null hypothesis |
|  | stats.wilcoxon(data['FSIQ'], data['PIQ']) |
|  | # paired no gaussian assumption, pvalue > 0.05, accept null hypothesis |
|  | stats.ttest\_1samp(data['FSIQ'] - data['PIQ'], 0) |
|  | # 1 sample t-test with difference, same result as stats.ttest\_rel |
|  |  |
|  | # one-tailed, greater than 'PIQ' |
|  | results = stats.ttest\_rel(data['FSIQ'], data['PIQ']) |
|  | alpha = 0.05 |
|  | if (results[0] > 0) & (results[1]/2 < alpha): |
|  | print "reject null hypothesis, mean of {} is greater than mean of {}".format('FSIQ','PIQ') |
|  | else: |
|  | print "accept null hypothesis" |
|  |  |
|  | # one-tailed, less than 'FSIQ |
|  | results = stats.ttest\_rel(data['PIQ'], data['FSIQ']) |
|  | alpha = 0.05 |
|  | if (results[0] < 0) & (results[1]/2 < alpha): |
|  | print "reject null hypothesis, mean of {} is less than mean of {}".format('PIQ','FSIQ') |
|  | else: |
|  | print "accept null hypothesis" |

**UNPAIRED**

Null hypothesis accepts the mean of two populations are equal. Alternative hypothesis can be one-tailed or two-tailed for means being unequal. The default case is to assume equal variance between two groups. We can also assume unequal variance between two groups.



One-tailed test can also be performed by making sure the order of sample. First mean is greater than second mean and etc.

|  |
| --- |
|  |
|  | |  | | --- | | data\_gender = data.groupby("Gender") | |  |   data\_gender.boxplot(column=['Weight']) |
|  |  |
|  | weight\_male = data[data['Gender'] == 'Male']['Weight'].dropna() |
|  | weight\_female = data[data['Gender'] == 'Female']['Weight'].dropna() |
|  |  |
|  | # paired 2 sample t-test |
|  | stats.ttest\_ind(weight\_male, weight\_female, equal\_var=False) |
|  | # pvalue <<< 0.05 reject null hypothesis, weight mean between male and female are not equal |
|  |  |
|  | # one-tailed, greater than female |
|  | results = stats.ttest\_ind(weight\_male, weight\_female, equal\_var=False) |
|  | alpha = 0.05 |
|  | if (results[0] > 0) & (results[1]/2 < alpha): |
|  | print "reject null hypothesis, mean of {} is greater than mean of {}".format('Male','Female') |
|  | else: |
|  | print "accept null hypothesis" |
|  | # reject null hypothesis, mean of Male is greater than mean of Female |
|  |  |
|  | # one-tailed, less than female |
|  | results = stats.ttest\_ind(weight\_male, weight\_female, equal\_var=False) |
|  | alpha = 0.05 |
|  | if (results[0] < 0) & (results[1]/2 < alpha): |
|  | print "reject null hypothesis, mean of {} is less than mean of {}".format('Male','Female') |
|  | else: |
|  | print "accept null hypothesis" |
|  | # accept null hypothesis |
|  |  |
|  | # one-tailed, less than male |
|  | results = stats.ttest\_ind(weight\_female, weight\_male, equal\_var=False) |
|  | alpha = 0.05 |
|  | if (results[0] < 0) & (results[1]/2 < alpha): |
|  | print "reject null hypothesis, mean of {} is less than mean of {}".format('Female','Male') |
|  | else: |
|  | print "accept null hypothesis" |
|  | # reject null hypothesis, mean of Female is less than mean of Male |

CHI-SQUARE TEST

It is an inferential statistical test which works on categorical data.

The Chi-Squared test is a statistical hypothesis test that assumes (the null hypothesis) that the observed frequencies for a categorical variable match the expected frequencies for the categorical variable. The test calculates a statistic that has a chi-squared distribution, named for the Greek capital letter Chi (X) pronounced “ki” as in kite.

**DEGREE OF FREEDOM IN CHI SQUARED DISTRIBUTION**

The degrees of freedom in Chi Squared distribution is equal to the number of standard normal deviates being summed. The mean of a Chi-square distribution is its degrees of freedom. A chi-square distribution constructed by squaring a single standard normal distribution is said to have 1 degree of freedom

The **degrees of freedom** ( **df** or **d**) tell you how many numbers in your grid are actually independent. For a Chi-square grid, the degrees of freedom can be said to be the number of cells you need to fill in before, given the totals in the margins, you can fill in the rest of the grid using a formula.

The degrees of freedom for a Chi-square grid are equal to the number of rows minus one times the number of columns minus one: that is, (R-1)\*(C-1).

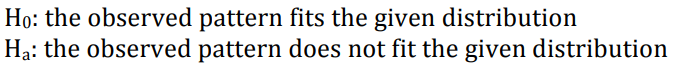
As the degree of freedom (df), increases the Chi-square distribution approaches a normal distribution

Following important steps used for Chi square test

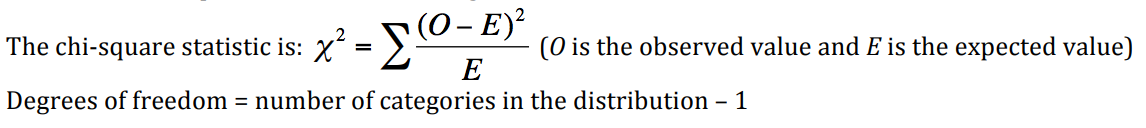
1. State The Hypothesis
2. Formulate Data Analysis Plan
3. Analyze The Smaple Data
4. Interpret The Outcome

**CHI-SQUARE TEST FOR GOODNESS OF FIT**

**Checks weather or not an observed pattern of data fits some given distribution**



We want to know if the pattern from our data follows a given distribution.



The main data using for chi-square test is the frequency count (crosstab) for categories. The default setting is to expect equal distribution among different categories. Degree of freedom (dof) is number of categories minus 1.

If t value obtained is greater than corresponding t value from table corresponds to (dof-1, α = .05) then reject null hypothesis else accept it.

**Use pandas.crosstab** to get the frequency count from different categorical features. This will be your data fed into your test

import scipy**,** scipy.stats.chisquare

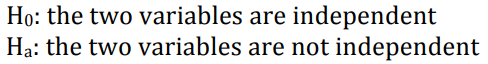
observed\_values**=**scipy**.**array([18,21,16,7,15])

expected\_values**=**scipy**.**array([22,19,44,8,16])

scipy**.**stats**.**chisquare(observed\_values, f\_exp**=**expected\_values)

**CHI-SQUARE TEST FOR INDEPENDENCE**

**Checks weather two categorical variables are related or not (independence)**



The key to the test is computing chi-square statistic and p-value for the hypothesis test of independence of the observed frequencies in the contingency table. The expected frequencies are computed based on the marginal sums under the assumption of independence.

**Use pandas.crosstab** to get the frequency count from different categorical features. This will be your data fed into your test

dfct = pd.crosstab(pd.Series(['Nighthawks', 'Scouts', 'Dragoons']) \

.sample(100, random\_state=1, replace=True) \

.reset\_index(drop=True),

pd.Series(['1st', '2nd', '3rd']) \

.sample(100, random\_state=2, replace=True) \

.reset\_index(drop=True))

# dfct

chi2, p, dof, exp = stats.chi2\_contingency(dfct)

# p > 0.05, accept null hypothesis, two variables are independent

# p < 0.05, reject null hypothesis, two variables are NOT independent

[ANALYSIS OF VARIANCE (ANOVA)](http://hamelg.blogspot.com/2015/11/python-for-data-analysis-part-16_23.html)

The analysis of variance (ANOVA) can be thought of as an extension to the t-test. The [independent t-test](https://pythonfordatascience.org/independent-t-test-python/) is used to compare the means of a condition between 2 groups. ANOVA is used when one wants to compare the means of a condition between 2+ groups. ANOVA is an omnibus test, meaning it tests the data as a whole. Another way to say that is this, ANOVA tests if there is a difference in the mean somewhere in the model (testing if there was an overall effect), but it does not tell one where the difference is if the there is one. To find out where the difference is between the groups, one has to conduct post-hoc tests.

ONE-WAY ANOVA

The one-way ANOVA tests whether the mean of some numeric variable differs across the levels of one categorical variable. It essentially answers the question: do any of the group means differ from one another? We won't get into the details of carrying out an ANOVA by hand as it involves more calculations than the t-test, but the process is similar: you go through several calculations to arrive at a test statistic and then you compare the test statistic to a critical value based on a probability distribution. In the case of the ANOVA, you use the "[f-distribution](https://en.wikipedia.org/wiki/F-distribution)".

The scipy library has a function for carrying out one-way ANOVA tests called scipy.stats.f\_oneway(). Let's generate some fake voter age and demographic data and use the ANOVA to compare average ages across the groups:

In [2]:

import **numpy** **as** **np**

import **pandas** **as** **pd**

import **matplotlib.pyplot** **as** **plt**

import **scipy.stats** **as** **stats**

In [3]:

np.random.seed(12)

races = ["asian","black","hispanic","other","white"]

*# Generate random data*

voter\_race = np.random.choice(a= races,

p = [0.05, 0.15 ,0.25, 0.05, 0.5],

size=1000)

voter\_age = stats.poisson.rvs(loc=18,

mu=30,

size=1000)

*# Group age data by race*

voter\_frame = pd.DataFrame({"race":voter\_race,"age":voter\_age})

groups = voter\_frame.groupby("race").groups

*# Etract individual groups*

asian = voter\_age[groups["asian"]]

black = voter\_age[groups["black"]]

hispanic = voter\_age[groups["hispanic"]]

other = voter\_age[groups["other"]]

white = voter\_age[groups["white"]]

*# Perform the ANOVA*

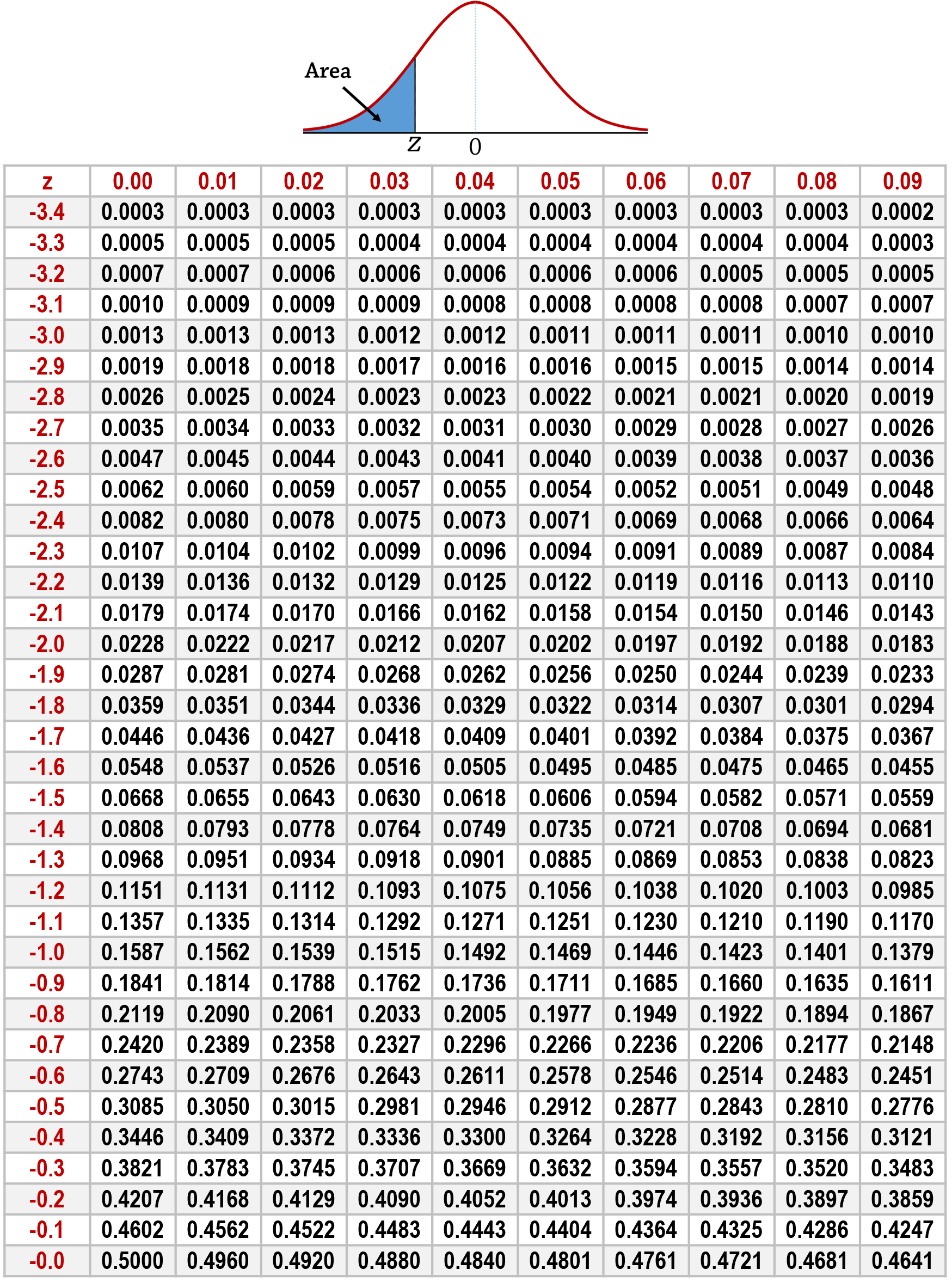
stats.f\_oneway(asian, black, hispanic, other, white)

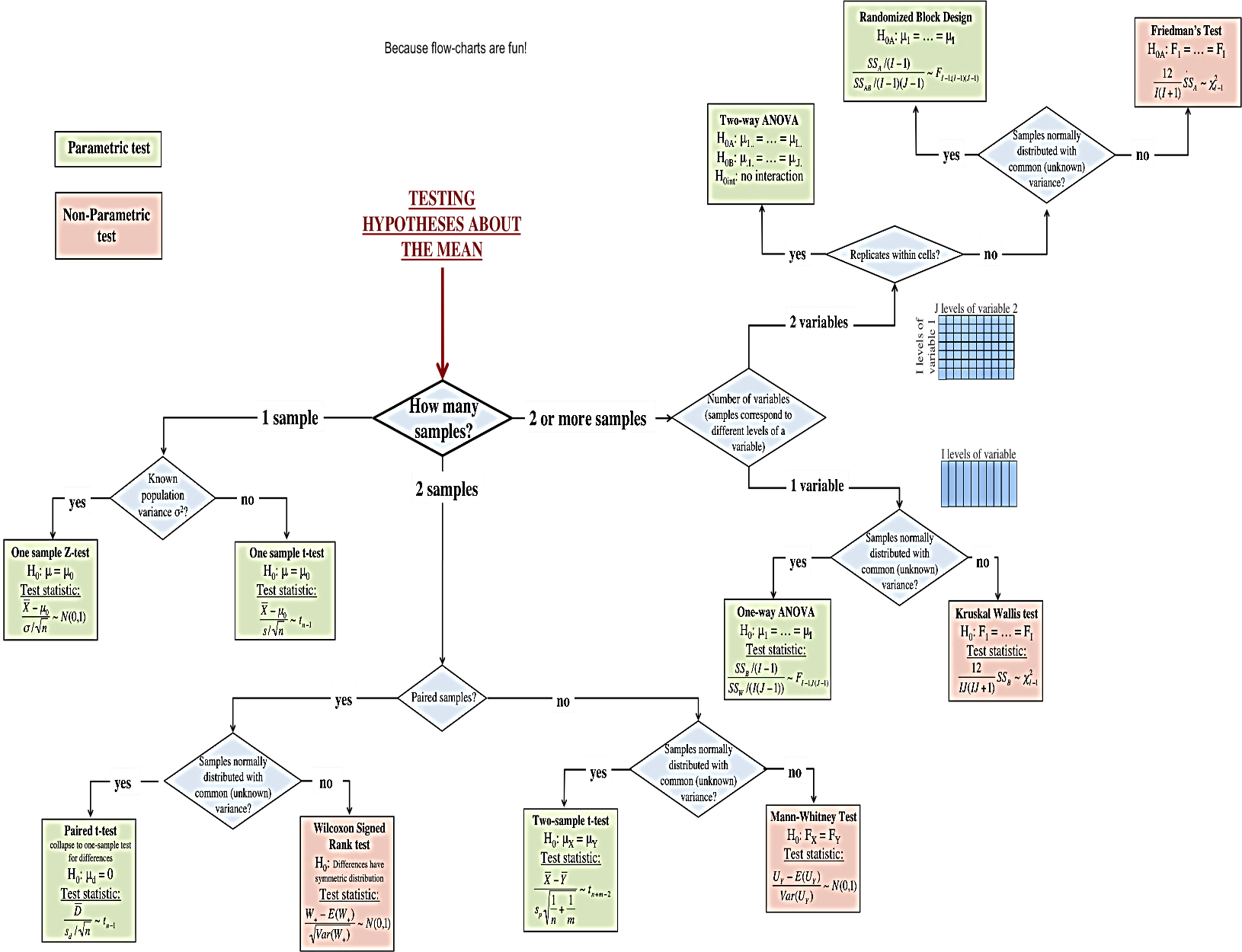
Out[3]:

F\_onewayResult(statistic=1.7744689357289216, pvalue=0.13173183202014213)

The test output yields an F-statistic of 1.774 and a p-value of 0.1317, indicating that there is no significant difference between the means of each group.

**Z-Table**





DATA PREPROCESSING

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.

1. Feature Scaling
2. Reduce Data Skewness
3. Handling Missing Data
4. Handling Categorical Data
5. Outlier Detection & Treatment
6. Correlation

FEATURE SCALING

Most of the times, your dataset will contain features highly varying in magnitudes, units and range. But since, most of the machine learning algorithms use Euclidian distance between two data points in their computations, this is a problem. To suppress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling.

Rule of thumb I follow here is any algorithm that computes distance or assumes normality, **scale your features!!!**

Some examples of algorithms where feature scaling matters are:

* **k-nearest neighbors** with a Euclidean distance measure is sensitive to magnitudes and hence should be scaled for all features to weigh in equally.
* Scaling is critical, while performing**Principal Component Analysis(PCA)**. PCA tries to get the features with maximum variance and the variance is high for high magnitude features. This skews the PCA towards high magnitude features.
* We can speed up**gradient descent** by scaling. This is because θ will descend quickly on small ranges and slowly on large ranges, and so will oscillate inefficiently down to the optimum when the variables are very uneven.
* **Tree based models** are not distance based models and can handle varying ranges of features. Hence, Scaling is not required while modelling trees.
* Algorithms like **Linear Discriminant Analysis(LDA), Naive Bayes**are by design equipped to handle this and gives weights to the features accordingly. Performing a features scaling in these algorithms may not have much effect.

****How to Scale Features****

There are four common methods to perform Feature Scaling.

* Standardization

Standardization replaces the values by their Z scores. **Standardization** of datasets is a **common requirement for many machine learning estimators** implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with **zero mean and unit variance**.

In practice we often ignore the shape of the distribution and just transform the data to center it by removing the mean value of each feature, then scale it by dividing non-constant features by their standard deviation.

https://miro.medium.com/max/120/1*LysCPCvg0AzQenGoarL_hQ.png

This redistributes the features with their mean μ = 0 and standard deviation σ = 1.

* Mean Normalization

This distribution will have values between -1 and 1with μ=0.

X new = (X – X min) / (X max – X min)

**Standardization** and **Mean Normalization** can be used for algorithms that assumes zero centric data like **Principal Component Analysis(PCA).**

* ****Min-Max Scaling****

https://miro.medium.com/max/169/1*19hq_t_NFQ6YVxMxsT0Cqg.png

This scaling brings the value between 0 and 1.

* ****Unit Vector****

https://miro.medium.com/max/99/1*u2Up0eaer56dpmaElU3Zxw.png

Scaling is done considering the whole feature vector to be of unit length.

Min-Max Scaling and Unit Vector techniques produces values of range [0,1]. When dealing with features with hard boundaries this is quite useful. For example, when dealing with image data, the colors can range from only 0 to 255.

* MaxAbsScaler

Scale each feature by its maximum absolute value. This estimator scales and translates each feature individually such that the maximal absolute value of each feature in the training set will be 1.0. It does not shift/center the data, and thus does not destroy any sparsity.

The MaxAbsScaler works very similarly to the MinMaxScaler but automatically scales the data to a [-1,1] range based on the absolute maximum. This scaler is meant for data that is already centered at zero or sparse data. It does not shift/center the data, and thus does not destroy any sparsity.

* RobustScaler

Scale features using statistics that are robust to outliers. This Scaler removes the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).

If your data contains many **outliers**, scaling using the mean and standard deviation of the data is likely to not work very well. In these cases, you can use the RobustScaler. **It removes the median and scales the data according to the quantile range**. The exact formula of the RobustScaler is not specified by the documentation. If you want full details you can always check [the source code](https://github.com/scikit-learn/scikit-learn/blob/55bf5d9/sklearn/preprocessing/data.py#L1035).

By default, the scaler uses the Inter Quartile Range (IQR), which is the range between the 1st quartile and the 3rd quartile. The quantile range can be manually set by specifying the quantile\_range parameter when initiating a new instance of the RobustScaler. Here, we transform feature 3 using an quantile range from 10% till 90%.

* QuantileTransformer

Transform features using quantiles information. This method transforms the features to follow a uniform or a normal distribution. Therefore, for a given feature, this transformation tends to spread out the most frequent values. It also reduces the impact of (marginal) outliers: this is therefore a robust preprocessing scheme.

An alternative standardization is scaling features to lie between a given minimum and maximum value, often between zero and one, or so that the maximum absolute value of each feature is scaled to unit size. This can be achieved using MinMaxScaler or MaxAbsScaler, respectively.

MaxAbsScaler works in a very similar fashion, but scales in a way that the training data lies within the range [-1, 1] by dividing through the largest maximum value in each feature. It is meant for data that is already centered at zero or sparse data.

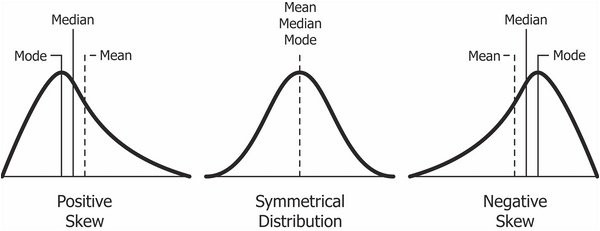
Centering sparse data would destroy the sparseness structure in the data, and thus rarely is a sensible thing to do. However, it can make sense to scale sparse inputs, especially if features are on different scales.

MaxAbsScaler and maxabs\_scale was specifically designed for scaling sparse data, and are the recommended way to go about this. However, scale and StandardScaler can accept scipy.sparse matrices as input, as long as with\_mean=False is explicitly passed to the constructor. Otherwise a ValueError will be raised as silently centering would break the sparsity and would often crash the execution by allocating excessive amounts of memory unintentionally. RobustScaler cannot be fitted to sparse inputs, but you can use the transform method on sparse inputs.

DATA SKEWNESS

It is the *degree of distortion* from the symmetrical bell curve or the normal distribution. It measures the lack of symmetry in data distribution. It differentiates extreme values in one versus the other tail. A symmetrical distribution will have a skewness of 0.

There are two types of Skewness: Positive and Negative



A **left-skewed distribution** has a long left tail. Left-skewed distributions are also called negatively-skewed distributions. That’s because there is a long tail in the negative direction on the number line. The mean is also to the left of the peak.

A **right-skewed distribution** has a long right tail. Right-skewed distributions are also called positive-skew distributions. That’s because there is a long tail in the positive direction on the number line. The mean is also to the right of the peak.

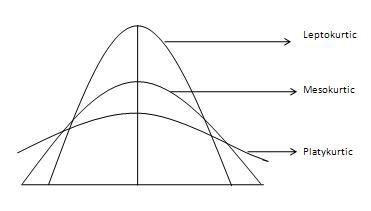
The rule of thumb seems to be:

* If the skewness is between -0.5 and 0.5, the data are fairly symmetrical.
* If the skewness is between -1 and -0.5(negatively skewed) or between 0.5 and 1(positively skewed), the data are moderately skewed.
* If the skewness is less than -1(negatively skewed) or greater than 1(positively skewed), the data are highly skewed.

Kurtosis

Kurtosis is all about the tails of the distribution — not the peakedness or flatness. It is used to describe the extreme values in one versus the other tail. It is actually the measure of outliers present in the distribution.

**High kurtosis** in a data set is an indicator that data has heavy tails or outliers. If there is a high kurtosis, then, we need to investigate why do we have so many outliers. It indicates a lot of things, maybe wrong data entry or other things. Investigate!  
**Low kurtosis** in a data set is an indicator that data has light tails or lack of outliers. If we get low kurtosis(too good to be true), then also we need to investigate and trim the dataset of unwanted results.



**Mesokurtic**: This distribution has kurtosis statistic similar to that of the normal distribution. It means that the extreme values of the distribution are similar to that of a normal distribution characteristic. This definition is used so that the standard normal distribution has a kurtosis of three.  
**Leptokurtic (**Kurtosis > 3**):**Distribution is longer, tails are fatter. Peak is higher and sharper than Mesokurtic, which means that data are heavy-tailed or profusion of outliers.  
Outliers stretch the horizontal axis of the histogram graph, which makes the bulk of the data appear in a narrow (“skinny”) vertical range, thereby giving the “skinniness” of a leptokurtic distribution.  
**Platykurtic: (**Kurtosis < 3**):**Distribution is shorter, tails are thinner than the normal distribution. The peak is lower and broader than Mesokurtic, which means that data are light-tailed or lack of outliers.  
The reason for this is because the extreme values are less than that of the normal distribution.

Identify Skewness

|  |  |  |
| --- | --- | --- |
| **METHOD** | **RIGHT SKEWNESS** | **LEFT SKEWNESS** |
| scipy stats.skew() | skewness > 0.5 : more weight in the left tail of the distribution | skewness < -0.5 : more weight in the right tail of the distribution. |
| Boxplot | Image: Seton Hall University  If a box plot is skewed to the right, the box shifts to the left and the right whisker gets longer. As a result, the mean is greater than the median. | left skewed boxplot  If the bulk of observations are on the high end of the scale, a boxplot is left skewed. Consequently, the left whisker is longer than the right whisker. |
| Histogram | right skewed histogram  A histogram is right skewed if the peak of the histogram veers to the left. Therefore, the histogram’s tail has a positive skew to the right. | left skewed histogram  Left skewed histograms are Histograms with long tails on the left. |

Shapiro-Wilks test

We can objectively determine if the variable is skewed using the Shapiro-Wilks test. The null hypothesis for this test is that the data is a sample from a normal distribution, so a p-value less than 0.05 indicates significant skewness. We’ll apply the test to the response variable Sale Price above labeled “resp” using Scipy.stats in Python.

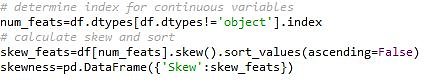
**[https://opendatascience.com/wp-content/uploads/2019/05/Code-1-3.jpg](https://opendatascience.com/wp-content/uploads/2019/05/Code-1-3.jpg)**

**[https://opendatascience.com/wp-content/uploads/2019/05/Code-2-3.jpg](https://opendatascience.com/wp-content/uploads/2019/05/Code-2-3.jpg)**

The p-value is not surprisingly less than 0.05, so we can conclude that the variable is skewed.

Pearson Test Pandas

A more convenient way of evaluating skewness is with pandas’ “. skew” method. It calculates the Fisher–Pearson standardized moment coefficient for all columns in a dataframe. We can calculate it for all the features in Kaggle’s Home Value dataset (labeled “df”) simultaneously with the following code.

**[](https://opendatascience.com/wp-content/uploads/2019/05/Code-3-3.jpg)**

A few of the variables like Pool Area are highly right skewed due to lots of zeros, this is okay. Some models like decision trees are fairly robust to skewed features.

Treatment for Skewness

Cube Root Transformation

The cube root transformation involves converting x to x^(1/3). This is a fairly strong transformation with a substantial effect on distribution shape: but is weaker than the logarithm. It can be applied to negative and zero values too.

Square Root Transformation

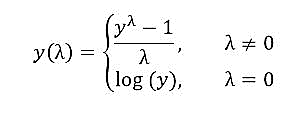
The square, x to x², has a moderate effect on distribution shape and it could be used to reduce left skewness. Applied to positive values only. Hence, observe the values of column before applying.

Logarithm Transformation

The logarithm, x to log base 10 of x, or x to log base e of x (ln x), or x to log base 2 of x, is a strong transformation and can be used to reduce right skewness.

Box Cox Transformation

An alternative to manually trying a variety of transformations is the Box Cox transformation. For each variable, a Box Cox transformation estimates the value lambda from -5 to 5 that maximizes the normality of the data using the equation below.

**[](https://opendatascience.com/wp-content/uploads/2019/05/BoxCox.jpg)**

For negative values of lambda, the transformation performs a variant of the reciprocal of the variable. At a lambda of zero, the variable is log transformed, and for positive lambda values, the variable is transformed the power of lambda. We can apply “boxcox” to all the skewed variables in the dataframe “df” using Scipy.stats.

**[https://opendatascience.com/wp-content/uploads/2019/05/Code-8-1.jpg](https://opendatascience.com/wp-content/uploads/2019/05/Code-8-1.jpg)**

HANDLING MISSING DATA

Data in real world are rarely clean and homogeneous. Typically, they tend to be incomplete, noisy, and inconsistent and it is an important task of a Data scientist to prepossess the data by filling missing values. It is important to be handled as they could lead to wrong prediction or classification for any given model being used.

Find Missing Values

Missing values could be: NaN, empty string, -1, -99, -999 and so on. In order to understand if -1 is a missing value or not we could draw a histogram. If this variable has a uniform distribution between 0 and 1 and it has a small peak at -1 then -1 is actually a missing value.

Missing values can be hidden from us and by hidden mean replaced by some other value beside NaN. Therefore, it is always beneficial to plot a **histogram** in order to identify those values.

Standard missing values: Taking a look at the column, we can see that Pandas filled in the blank space with “NA”. Using the isnull() method, we can confirm that both the missing value and “NA” were recognized as missing values. Both Boolean responses are True.

Non-Standard Missing Values: Sometimes it might be the case where there’s missing values that have different formats, examples n/a, NA, na etc.

Unexpected Missing Values: For example, if our feature is expected to be a string, but there’s a numeric type, then technically this is also a missing value.

Resolve Missing Values

* Ignore the data row

This is a quick solution and typically is preferred in cases where the percentage of missing values is relatively low (<5%). It is a dirty approach as you lose data. Imagine you drop one whole observation just because one of the features had a missing value, even if the rest of the features are perfectly filled and informative!

* Will drop all rows that have any missing values.

*dataframe.dropna(inplace=True)*

* You can also select to drop the rows only if all of the values in the row are missing.

*dataframe.dropna(how='all',inplace=True)*

* Sometimes, you may just want to drop a column (variable) that has some missing values.

*dataframe.dropna(axis=1,inplace=True)*

* Finally, you may want to keep only the rows with at least 4 non-na values:

*dataframe.dropna(thresh=4,inplace=True)*

Even if it’s a small percentage of the dataset doesn’t mean you should drop it.

* Back-fill or forward-fill to propagate next or previous values respectively

for back fill

*dataframe.fillna(method='bfill',inplace=True)*

for forward-fill

*dataframe.fillna(method='ffill',inplace=True)*

Note that the NaN value will remain even after forward filling or back filling if a next or previous value isn’t available or it is also a NaN value.

* Replace with some constant value outside fixed value

This method is useful as it gives the possibility to group missing values as a separate category represented by a constant value. It is a preferred option when it doesn’t make sense to try and predict a missing value. The downside is that performance of linear models can suffer. Use a global constant to fill in for missing values.

For example, in the titanic dataset filling in the missing value of the Embarked feature with the most common Port of Embarkation might not really makes sense as opposed to using something like “N/A”.

*dataframe.Column\_Name.fillna(-99,inplace=True)*

* Replace with mean, median value

This simple imputation method is based on treating every variable individually, ignoring any interrelationships with other variables. This method is beneficial for simple linear models and NN. But for tree based methods it can be harder for the algorithm to understand that there was a missing value.

**MEAN: Suitable for continuous data without outliers**

dataframe.Column\_Name.fillna(dataframe.Column\_Name.mean(),inplace=True)

**MEDIAN: Suitable for continuous data with outliers**

dataframe.Column\_Name.fillna(dataframe.Column\_Name.median(),inplace=True)

Another option would be to randomly fill them with values close to the mean value but within one standard deviation.

Column\_Name\_avg = dataframe['Column\_Name'].mean()  
Column\_Name\_std = dataframe['Column\_Name'].std()  
Column\_Name\_null\_count = dataframe['Column\_Name'].isnull().sum()  
Column\_Name\_null\_random\_list = np.random.randint(Column\_Name\_avg - Column\_Name\_std, Column\_Name\_avg + Column\_Name\_std, size=Column\_Name\_null\_count)  
dataframe['Column\_Name'][np.isnan(dataframe['Column\_Name'])] = Column\_Name\_null\_random\_list  
dataframe['Column\_Name'] = dataframe['Column\_Name'].astype(int)

For categorical feature you can select to fill in the missing values with the most common value as illustrated below.

dataframe.Column\_Name.fillna(dataframe.Column\_Name.mode()[0], inplace=True)

Note that Mean, Median and Mode imputation diminishes any correlations involving the variable(s) that are imputed. This is because we assume that there is no relationship between the imputed variable and any other measured variables. Thus, those imputations have some attractive properties for univariate analysis but become problematic for multivariate analysis.

* Mean, Median in combination with group by

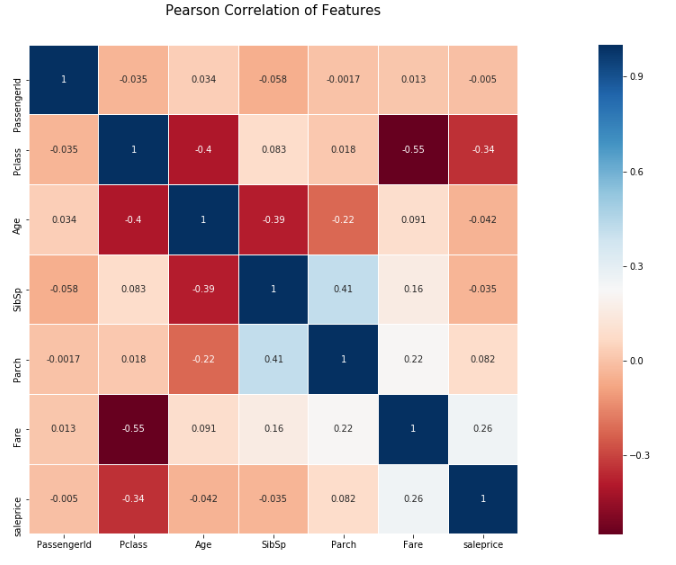
In this case, we impute the missing values by not considering the global mean/median value of the observations but by investigating some dependency with other variables.

## Fill missing values in Age feature with each sex’s mean value of ## Age   
train['Age'].fillna(train.groupby('Sex')['Age'].transform("mean"), inplace=True)## Fill missing values in Age feature with each sex’s median value ## of Age   
train['Age'].fillna(train.groupby('Sex')['Age'].transform("median"), inplace=True)

* Linear Regression

To begin, several predictors of the variable with missing values are identified using a correlation matrix. The best predictors are selected and used as independent variables in a regression equation. The variable with missing data is used as the target variable.

colormap = plt.cm.RdBu  
plt.figure(figsize=(32,10))  
plt.title('Pearson Correlation of Features', y=1.05, size=15)  
sns.heatmap(train.corr(),linewidths=0.1,vmax=1.0,   
square=True, cmap=colormap, linecolor='white', annot=True)



Observe that target variable ‘Age’ is correlated with the features ‘Pclass’,’SibSp’,’Parch’,’Fare’ and they don’t contain missing values.

Thus, we are going to fit a linear model and we will try to predict the missing values of the ‘Age’ feature. We used a Linear regression model to replace the nulls in the feature ‘Age’. In general, one can experiment with different algorithms and check which gives the best accuracy instead of sticking to a single algorithm.

from sklearn.linear\_model import LinearRegression

linreg = LinearRegression()

data = train[['Pclass','SibSp','Parch','Fare','Age']]

**#Step-1: Split the dataset that contains the missing values and no missing values are test and train respectively.**

x\_train = data[data['Age'].notnull()].drop(columns='Age')

y\_train = data[data['Age'].notnull()]['Age']

x\_test = data[data['Age'].isnull()].drop(columns='Age')

y\_test = data[data['Age'].isnull()]['Age']

**#Step-2: Train the machine learning algorithm**

linreg.fit(x\_train, y\_train)

**#Step-3: Predict the missing values in the attribute of the test data.**

predicted = linreg.predict(x\_test)

**#Step-4: Let’s obtain the complete dataset by combining with the target attribute.**

train.Age[train.Age.isnull()] = predicted

train.info()

Estimating the missing values by using a weighted least squares or generalized least squares model leads to better results (Lasso & Ridge).

The limitations of the model tend to outweigh its advantages.: They lead to an underestimation of standard errors and, thus, overestimation of test statistics. The main reason is that the replaced values are completely determined by a model applied to other variables and they tend to fit together “too well”, in other words, they contain no error. Also, one must also assume that there is a linear relationship between the variables used in the regression equation when there may not be one.

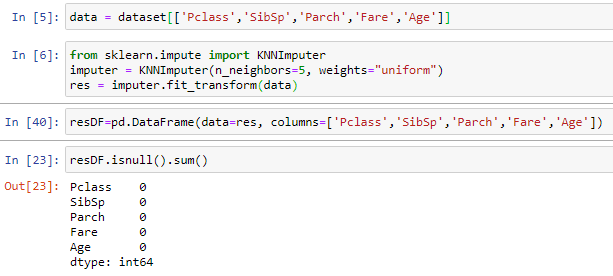
* ****k-Neareast Neighbor (kNN) Imputation****

Even though XGBoost and Random Forest could also be used for data imputation we will be discuss KNN as it is widely used. For k-Nearest Neighbor imputation, the missing values are based on a kNN algorithm. In this method, k neighbors are chosen based on some distance measure and their average is used as an imputation estimate.

The method requires the selection of the number of nearest neighbors, and a distance metric. KNN can predict both discrete (most frequent value among the k nearest neighbors) and continuous attributes (mean among the k nearest neighbors). The distance metric varies according to the type of data:

1. Continuous Data: The commonly used distance metrics for continuous data are Euclidean, Manhattan and Cosine
2. Categorical Data: Hamming distance is generally used in this case. It takes all the categorical attributes and for each, count one if the value is not the same between two points. The Hamming distance is then equal to the number of attributes for which the value was different.

One of the most attractive features of the KNN is that it is simple to understand and easy to implement. The main disadvantage of using kNN imputation is that it becomes time-consuming when analyzing large datasets. Also, the accuracy of KNN can be severely degraded with high-dimensional data because there is little difference between the nearest and farthest neighbor. Finally, the number of neighbors (k) has to be carefully selected when using kNN imputation.



* ****Multiple Imputation using MICE (Multiple Imputation by Chained Equations)****

Unarguably one of the most advanced methodology for performing missing data imputation is Multivariate imputation by chained equations (MICE).

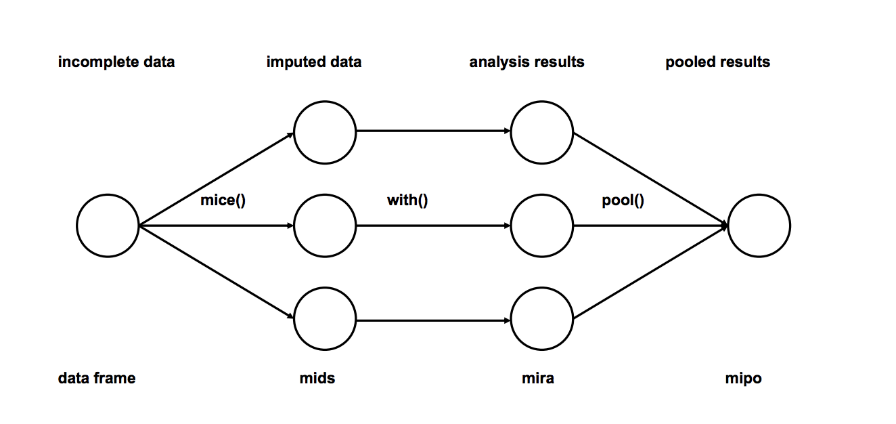
Creating multiple imputations, as opposed to single imputations to “complete” datasets, accounts for the statistical uncertainty in the imputations. In general, the limitation with single imputation is that because these techniques find maximally likely values, they do not generate entries which accurately reflect the distribution of the underlying data. Moreover, the chained equations approach is very flexible and can handle variables of varying types (e.g., continuous or binary).

Note that all of the techniques discussed so far are what one might call “single imputation”: each value in the dataset is filled in exactly once.

The MICE algorithm works: by running multiple regression models and each missing value is modeled conditionally depending on the observed (non-missing) values.

For example, let’s consider the extreme case when we impute the missing values with the mean value. In reality, we would expect to see some variability in it: extreme values, outliers, and records which do not completely fit the “pattern” of the data. All dataset contain noise in some extent and mean value replacement makes no attempt to represent it in its result. This leads to bias in any models, which are exposed to a trend (the presence of the mean value in the dataset) which does not exist in the underlying data. This will ultimate decreases accuracy during both the train and test phases. The algorithm consists of three stages.

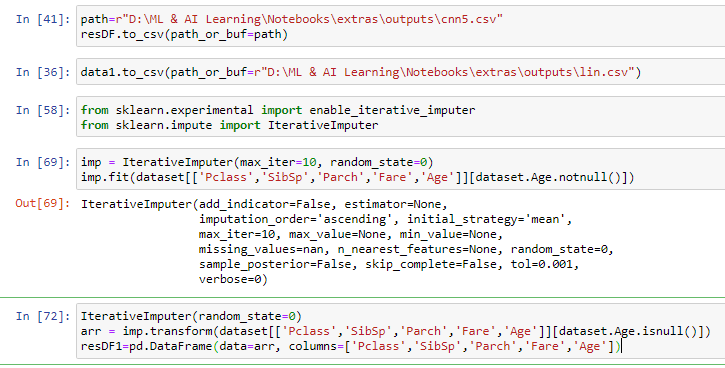
1. **Imputation**: Impute the missing entries of the incomplete data sets *m*times (*m*=3 in the figure). Note that imputed values are drawn from a distribution. Simulating random draws doesn’t include uncertainty in model parameters. Better approach is to use Markov Chain Monte Carlo (MCMC) simulation. This step results in m complete data sets.
2. **Analysis**: Analyze each of the *m* completed data sets.
3. **Pooling**: Integrate the *m* analysis results into a final result



from fancyimpute import MICE#We use the train dataframe from Titanic dataset*#fancy impute removes column names.*  
train\_cols = list(train)# Use MICE to fill in each row's missing featurestrain = pd.DataFrame(MICE(verbose=False).complete(train))  
train.columns = train\_cols

By default, *fancyimpute* uses its own Bayesian ridge regression implementation.

In scikit lean we can flow the below



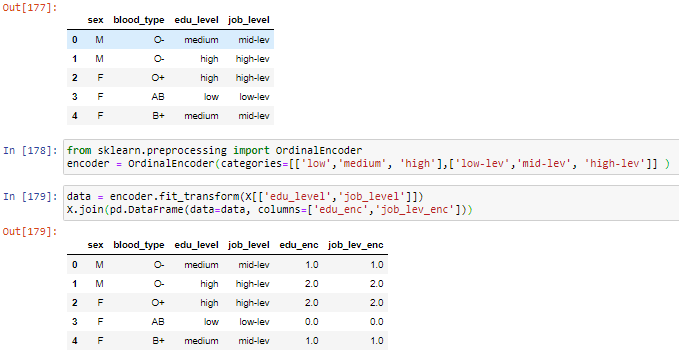
HANDLING CATEGORICAL DATA

* Ordinal Feature

Ordinal attributes are categorical attributes with a sense of order amongst the values. Hence they have a sense of order amongst them. In general, there is no generic module or function to map and transform these features into numeric representations based on order automatically.

Looking at the dataframe, you should notice education & job levels are the only ordinal feature (it can be ordered and the distance between the categories is not known). We’ll start with encoding this feature with the OrdinalEncoder class. Import the class and create a new instance. Then update the education& job level features by fitting and transforming the feature to the encoder.

**Note that input is sensitive to missing values & the order of our data can be respected by explicitly mention each category in a list with respective order.**



* Nominal Features

We can’t replace these features by a number since this would imply the features have an order, which is untrue in case of sex or blood type.

The most popular way to encode nominal features is one-hot-encoding. Essentially, each categorical feature with n categories is transformed into n binary features. This can be done with ColumTransformer and OneHotEncoder, a missing value can easily be handled as an extra feature. Note that to do this, you need to replace the missing value by an arbitrary value first (e.g. ‘missing’) If you, on the other hand, want to ignore the missing value and create an instance with all zeros (False), you can just set the handle\_unkown parameter of the OneHotEncoder to ignore.



Numerical features

Just like categorical data can be encoded, numerical features can be ‘decoded’ into categorical features. The two most common ways to do this are **discretization**and **binarization**.

Discretization

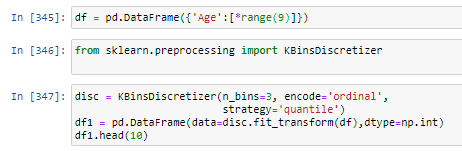
Discretization, also known as quantization or binning, **divides a continuous feature into a pre-specified number of categories** (bins), and thus makes the data discrete.

One of the main goals of a discretization is to significantly **reduce the number of discrete intervals of a continuous attribute**. Hence, why this transformation can increase the performance of tree based models.

Sklearn provides a KBinsDiscretizer class that can take care of this. The only thing you have to specify are the number of bins (n\_bins) for each feature and how to encode these bins (ordinal, onehot or onehot-dense). The optional strategy parameter can be set to three values:

* uniform, where all bins in each feature have identical widths.
* quantile (default), where all bins in each feature have the same number of points.
* kmeans, where all values in each bin have the same nearest center of a 1D k-means cluster.

It is important to pick the strategy parameter with care. Using the uniform strategy for example, is very sensitive for outliers and can make you end up with bins with just a few data points, i.e. the outliers.



|  |  |
| --- | --- |
|  |  |

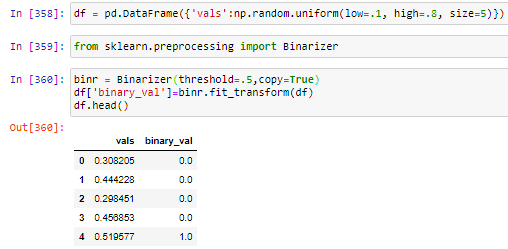
* Binarization

Feature binarization is **the process of tresholding numerical features to get boolean values**. Or in other words, assign a boolean value (True or False) to each sample based on a threshold. Note that binarization is an extreme form of two-bin discretization.

In general binarization is useful as a **feature engineering technique for creating new features that indicate something meaningful**. Just like the above-mentioned MissingIndicator is used to mark meaningful missing values.

The Binarizer class in sklearn implements binarization in a very intuitive way. The only parameters you need to specify are the threshold and copy. All values below or equal to the threshold are replaced by 0, above it by 1. If copy is set to False, inplace binarization is performed, otherwise a copy is made.

Consider feature 3 (f3) of our example and let’s create an extra binary feature with True for positive values and False for negative values. Import the Binarizer class, create a new instance with the threshold set to zero and copy to True. Then, fit and transform the binarizer to feature 3. The output is a new array with boolean values.



OUTLIER DETECTION & TREATMENT

An outlier is a data point that is distant from other similar points. Further simplifying an outlier is an observation that lies on abnormal observation amongst the normal observations in a sample set of population. In statistics, an **outlier** is an observation point that is distant from other observations.

There are many modeling techniques which are resistant to outliers or reduce the impact of them, but still detecting outliers and understanding them can lead to interesting findings. We generally define outliers as samples that are exceptionally far from the mainstream of data.There is no rigid mathematical definition of what constitutes an outlier; determining whether or not an observation is an outlier is ultimately a subjective exercise.

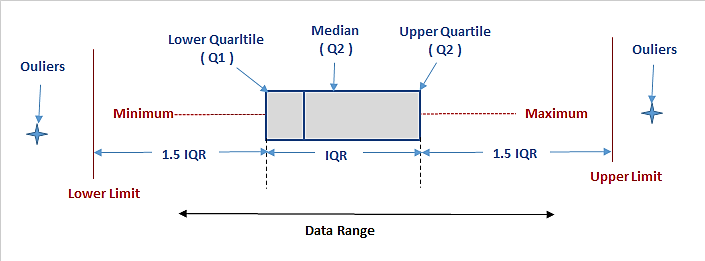
**Univariate Outlier:**A univariate outlier is a data point that consists of an extreme value on one variable.

**Multivariate Outlier**: A multivariate outlier is a combination of unusual scores on at least two variables.

****Detect Outliers****

Various visualization methods, like **Box-plot**, **Histogram** **and Scatter Plot** can be used. Various visualization methods, like **Box-plot**, **Histogram** **and Scatter Plot** can be used.

Box Plot Diagram



Box plot diagram also termed as Whisker’s plot is a graphical method typically depicted by quartiles and inter quartiles that helps in defining the upper limit and lower limit beyond which any data lying will be considered as outliers. The very purpose of this diagram is to identify outliers and discard it from the data series before making any further observation so that the conclusion made from the study gives more accurate results not influenced by any extremes or abnormal values.

IQR rule

In descriptive statistics, the interquartile range, also called the midspread or middle 50%, or technically H-spread, is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, IQR = Q₃ − Q₁.

Steps to perform Outlier Detection by identifying the lower bound and upper bound of the data:

1. Arrange your data in ascending order
2. Calculate Q1 (the first Quarter)
3. Calculate Q3 (the third Quartile)
4. Find IQR = (Q3 - Q1)
5. Find the lower Range = Q1 -(1.5 \* IQR)
6. Find the upper Range = Q3 + (1.5 \* IQR)

q1 = df[['column']].quantile(0.25)

q3 = df[['column']].quantile(0.75)

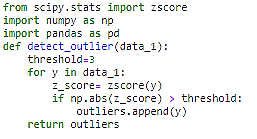
lower\_bound = q1 -(1.5 \* iqr)

upper\_bound = q3 +(1.5 \* iqr)

df[(df.column < lower\_range) | (df.column > upper\_range)]

Extreme Value Analysis

This is the most basic form of outlier detection and only good for 1-dimension data. In these types of analysis, it is assumed that values which are too large or too small are outliers. Z-test and Student’s t-test are examples of these statistical methods. These are good heuristics for initial analysis of data but they don’t have much value in multivariate settings. They can be used as final steps for interpreting outputs of other outlier detection methods.



****Linear Models****

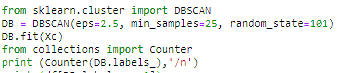
These methods model the data into a lower dimensional sub-spaces with the use of linear correlations. Then the distance of each data point to plane that fits the sub-space is being calculated. This distance is used to find outliers. PCA (Principal Component Analysis) is an example of linear models for anomaly detection.

Principal Component Analysis

Principal component analysis can completely restructure the data, removing redundancies and ordering newly obtained components according to the amount of the original variance that they express. This type of analysis offers a synthetic and complete view over data distribution, making multivariate outliers particularly evident.

cluster analysis

Outliers are isolated points in the space of variables, and DBScan is a clustering algorithm that links dense data parts together and marks the too-sparse parts. DBScan is therefore an ideal tool for an automated exploration of your data for possible outliers to verify.



However, DBSCAN requires two parameters, eps and min\_samples. These two parameters require multiple tries to locate the right values, making using the parameters a little tricky.

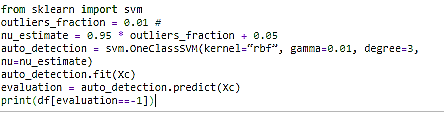
Start with a low value of min\_samples and try growing the values of eps from 0.1 upward. After every trial with modified parameters, check the situation by counting the number of observations in the class –1 inside the attribute labels, and stop when the number of outliers seems reasonable for a visual inspection.

Automating Outliers Detection with SVM

Support Vector Machines (SVM) is a powerful machine learning technique. OneClassSVM is an algorithm that specializes in learning the expected distributions in a dataset. OneClassSVM is especially useful as a novelty detector method if you can first provide data cleaned from outliers; otherwise, it’s effective as a detector of multivariate outliers. In order to have OneClassSVM work properly, you have two key parameters to fix:

* gamma, telling the algorithm whether to follow or approximate the dataset distributions. For novelty detection, it is better to have a value of 0 or superior (follow the distribution); for outlier detection values, smaller than 0 values are preferred (approximate the distribution).
* nu, which can be calculated by the following formula: nu\_estimate = 0.95 \* f + 0.05, where f is the percentage of expected outliers (a number from 1 to 0). If your purpose is novelty detection, f will be 0.

Executing the following script, you will get a OneClassSVM working as an outlier detection system:



Outliers should be investigated carefully. Often they contain valuable information about the process under investigation or the data gathering and recording process. Before considering the possible elimination of these points from the data, one should try to understand why they appeared and whether it is likely similar values will continue to appear. Of course, outliers are often bad data points.

Methods of Dealing Outliers

The common techniques used to deal with outliers are:

Deleting observations

We delete outlier values if it is due to data entry error, data processing error or outlier observations are very small in numbers. We can also use trimming at both ends to remove outliers.

Transforming and binning values

Transforming variables can also eliminate outliers. Natural log of a value reduces the variation caused by extreme values. Binning is also a form of variable transformation. Decision Tree algorithm allows to deal with outliers well due to binning of variable. We can also use the process of assigning weights to different observations.

Imputing

We can also impute outliers. We can use mean, median, mode imputation methods. Before imputing values, we should analyse if it is natural outlier or artificial. If it is artificial, we can go with imputing values. We can also use statistical model to predict values of outlier observation and after that we can impute it with predicted values.

Treat Outliers separately

If there are significant number of outliers, we should treat them separately in the statistical model. One of the approach is to treat both groups as two different groups and build individual model for both groups and then combine the output.

CORRELATION

Variables within a dataset can be related for lots of reasons. For example:

* One variable could cause or depend on the values of another variable.
* One variable could be lightly associated with another variable.
* Two variables could depend on a third unknown variable.

It can be useful in data analysis and modeling to better understand the relationships between variables. The statistical relationship between two variables is referred to as their correlation.

A correlation could be positive, meaning both variables move in the same direction, or negative, meaning that when one variable’s value increases, the other variables’ values decrease. Correlation can also be neural or zero, meaning that the variables are unrelated.

* Positive Correlation: both variables change in the same direction.
* Neutral Correlation: No relationship in the change of the variables.
* Negative Correlation: variables change in opposite directions.

The performance of some algorithms can deteriorate if two or more variables are tightly related, called multicollinearity. An example is linear regression, where one of the offending correlated variables should be removed in order to improve the skill of the model.

We may also be interested in the correlation between input variables with the output variable in order provide insight into which variables may or may not be relevant as input for developing a model.

The structure of the relationship may be known, e.g. it may be linear, or we may have no idea whether a relationship exists between two variables or what structure it may take. Depending what is known about the relationship and the distribution of the variables, different correlation scores can be calculated.

Covariance

Variables can be related by a linear relationship. This is a relationship that is consistently additive across the two data samples.

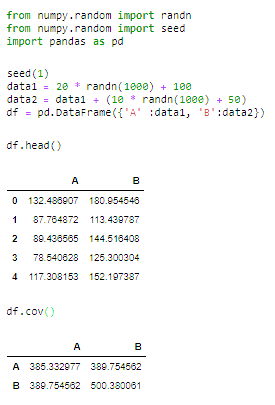
This relationship can be summarized between two variables, called the covariance. It is calculated as the average of the product between the values from each sample, where the values haven been centered (had their mean subtracted).

The calculation of the sample covariance is as follows:

cov(X, Y) = (sum (x - mean(X)) \* (y - mean(Y)) ) \* 1/(n-1)

The use of the mean in the calculation suggests the need for each data sample to have a Gaussian or Gaussian-like distribution.

The sign of the covariance can be interpreted as whether the two variables change in the same direction (positive) or change in different directions (negative). The magnitude of the covariance is not easily interpreted. A covariance value of zero indicates that both variables are completely independent.



The diagonal of the matrix contains the covariance between each variable and itself. The other values in the matrix represent the covariance between the two variables; in this case, the remaining two values are the same given that we are calculating the covariance for only two variables.

The covariance and covariance matrix are used widely within statistics and multivariate analysis to characterize the relationships between two or more variables.

The covariance between the two variables is 389.75. We can see that it is positive, suggesting the variables change in the same direction.

Pearson’s Correlation

The Pearson correlation coefficient (named for Karl Pearson) can be used to summarize the strength of the linear relationship between two data samples.

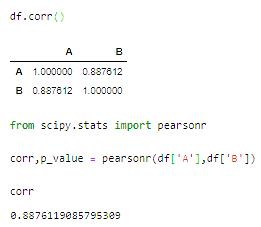
The Pearson’s correlation coefficient is calculated as the covariance of the two variables divided by the product of the standard deviation of each data sample. It is the normalization of the covariance between the two variables to give an interpretable score.

Pearson's correlation coefficient = covariance(X, Y) / (stdv(X) \* stdv(Y))

The use of mean and standard deviation in the calculation suggests the need for the two data samples to have a Gaussian or Gaussian-like distribution.

The result of the calculation, the correlation coefficient can be interpreted to understand the relationship.

The coefficient returns a value between -1 and 1 that represents the limits of correlation from a full negative correlation to a full positive correlation. A value of 0 means no correlation. The value must be interpreted, where often a value below -0.5 or above 0.5 indicates a notable correlation, and values below those values suggests a less notable correlation.



The pearsonr() SciPy function can be used to calculate the Pearson’s correlation coefficient between two data samples with the same length & DataFrame.corr Compute pairwise correlation of columns, excluding NA/null values.

The Pearson’s correlation coefficient can be used to evaluate the relationship between more than two variables.

This can be done by calculating a matrix of the relationships between each pair of variables in the dataset. The result is a symmetric matrix called a correlation matrix with a value of 1.0 along the diagonal as each column always perfectly correlates with itself.

Spearman’s Correlation

Two variables may be related by a nonlinear relationship, such that the relationship is stronger or weaker across the distribution of the variables.

Further, the two variables being considered may have a non-Gaussian distribution.

In this case, the Spearman’s correlation coefficient (named for Charles Spearman) can be used to summarize the strength between the two data samples. This test of relationship can also be used if there is a linear relationship between the variables, but will have slightly less power (e.g. may result in lower coefficient scores).

As with the Pearson correlation coefficient, the scores are between -1 and 1 for perfectly negatively correlated variables and perfectly positively correlated respectively.

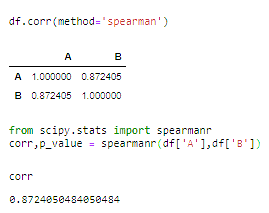
Instead of calculating the coefficient using covariance and standard deviations on the samples themselves, these statistics are calculated from the relative rank of values on each sample. This is a common approach used in non-parametric statistics, e.g. statistical methods where we do not assume a distribution of the data such as Gaussian.

Spearman's correlation coefficient = covariance(rank(X), rank(Y)) / (stdv(rank(X)) \* stdv(rank(Y)))

A linear relationship between the variables is not assumed, although a monotonic relationship is assumed. This is a mathematical name for an increasing or decreasing relationship between the two variables.

If you are unsure of the distribution and possible relationships between two variables, Spearman correlation coefficient is a good tool to use.

The spearmanr() SciPy function can be used to calculate the Spearman’s correlation coefficient between two data samples with the same length.



Variance Inflation Factor (VIF) Test

Colinearity is the state where two variables are highly correlated and contain similiar information about the variance within a given dataset. To detect colinearity among variables, simply create a correlation matrix and find variables with large absolute values. In R use the corr function and in python this can by accomplished by using numpy's corrcoef function.

Multicolinearity on the other hand is more troublesome to detect because it emerges when three or more variables, which are highly correlated, are included within a model. To make matters worst multicolinearity can emerge even when isolated pairs of variables are not colinear.

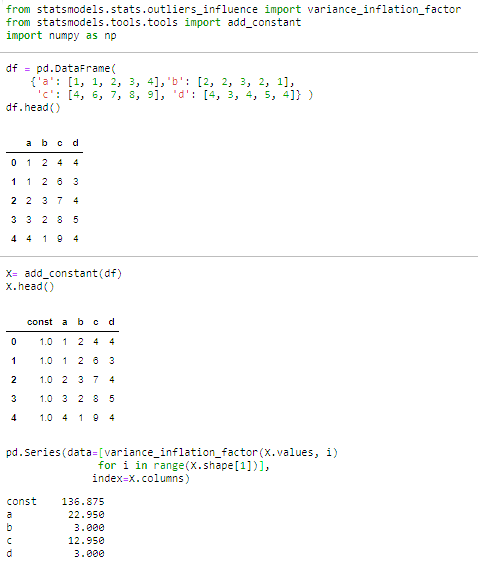
A common R function used for testing regression assumptions and specifically multicolinearity is "VIF()" and unlike many statistical concepts, its formula is straightforward:

V.I.F = 1 / (1 - R^2)

The Variance Inflation Factor (VIF) is a measure of colinearity among predictor variables within a multiple regression. It is calculated by taking the the ratio of the variance of all a given model's betas divide by the variane of a single beta if it were fit alone.

Steps for Implementing VIF

1. Run a multiple regression.
2. Calculate the VIF factors.
3. Inspect the factors for each predictor variable, if the VIF is between 5-10, multicolinearity is likely present and you should consider dropping the variable.



Below a & c has high VIF