* **Categorical data** represents data characteristics or qualitative descriptions.
* **Dichotomous data** is collected from either one of two categories. (an online survey might collect member/non-member or demographic information. )
* **Ordinal data** has a ranked order. Data that is collected on a value scale (movie rankings, survey results).
* **Nominal data** is data used as labels or names for other measures.
* **Numerical data** is obtained by taking a measurement from an instrument or by counting.
  + **Continuous data** can be subdivided infinitely.
  + **Interval data** is spaced out evenly on a scale. Also known as integer data, interval data does not use decimal places and can't be subdivided, multiplied or divided.
* Normal distribution

The **qualitative test for normality** is a visual assessment of the distribution of data.

Chart, histogram

Description automatically generated

geom\_density(), plot the distribution.

To test the distribution of vehicle weights from the built in mtcars dataset:

ggplot(mtcars,aes(x=wt)) + geom\_density() #visualize distribution using density plot

The **quantitative test for normality** uses a statistical test to quantify the probability of whether or not the test data came from a normally distributed dataset.

shapiro.test() to perform quantitative test.

shapiro.test(mtcars$wt)

If the p-value is greater than 0.05, the data is considered normally distributed.

**left skewed,** or negative skewed

Chart

Description automatically generated

There is a higher probability that extreme negative values exist within our dataset. We would use the median to describe the central tendency of the data.

**right skewed**, or positive skewed

Chart

Description automatically generated

There is a higher probability that extreme positive values exist within our dataset. We would use the median to describe the central tendency of the data.

* **Hypothesis**

**null hypothesis** ( H0 ) is can be explained by random chance.

**alternate hypothesis** ( Ha ) is influenced by non-random events.

1. Generate a null hypothesis, alternate hypothesis, and the significance level.
2. Identify a statistical analysis to assess the truth of the null hypothesis.
3. Compute p-value using statistical analysis.
4. Compare p-value to the significance level.
5. Reject (or fail to reject) the null hypothesis and generate the conclusion.

A **significance level** (alpha or ɑ) is a predetermined cutoff for our hypothesis test.

Table

Description automatically generated

**one-tailed hypothesis** describes one side of the distribution,

such as "x is greater than y" or "x is less than or equal to y."

**two-tailed hypotheses** describe both sides of the distribution,

such as "equal to" or "not equal to."

Chart

Description automatically generated

If **hypotheses** and **statistical test** are both **two-tailed**, use the statistical test p-value as is.

If **hypotheses** are **one-tailed**, but **statistical test** is **two-tailed**, divide the statistical test p-value by 2.

If calculated **p-value** is smaller than significance level, the null hypothesis is not true, so reject null hypothesis.

If calculated **p-value** is larger than significance level, not sufficient evidence to reject null hypothesis, and therefore fail to reject null hypothesis.

**Incorrect hypothesis**:

* **Type I error** (**false positive** error)—an error in which we reject the null hypothesis when it is actually true.
* **Type I error** can be limited by making your significance level smaller. A smaller significance level makes it harder to accidentally reject the null hypothesis when the data was truly random.
* **Type II error** (**false negative** error)—an error in which we fail to reject the null hypothesis when it is actually false.
* **Type II error** can be limited by increasing the power of the analysis.

sample\_n()requires two arguments:

* **tbl** is the name of the input table, the name of a data frame. We can use a dplyr pipe (%>%) to provide the data frame object.
* **size** is the number of rows to return.

sample\_table <- population\_table %>% sample\_n(50) #randomly sample 50 data points

The **Student's t-test** (**t-test**) is one of the most basic and popular statistical tests. In statistics, we use a t-test to compare the mean of one dataset to another under a few assumptions.

**One-sample t-test** is to determine whether there is a statistical difference between the means of a sample dataset and a hypothesized, potential population dataset.

* H0 : **no statistical difference** between the observed sample mean and its presumed population mean.
* Ha : **statistical difference** between the observed sample mean and its presumed population mean.

1. The input data is numerical and continuous. This is because we are testing the distribution of two datasets.
2. The sample data was selected randomly from its population data.
3. The input data is considered to be normally distributed.
4. The sample size is reasonably large.
5. The variance of the input data should be very similar.

t.test() implement a one-sample t-test.

Graphical user interface, text, application, email

Description automatically generated

To test if the miles driven from previous sample dataset is statistically different from the miles driven in population data,

t.test(log10(sample\_table$Miles\_Driven),mu=mean(log10(population\_table$Miles\_Driven)))

t.test(log10(sample\_table$Miles\_Driven),log10(sample\_table2$Miles\_Driven)) #compare means of two samples

**Two-sample t-Test** determines whether the means of two samples are statistically different.

* H0 : **no statistical difference** between the two observed sample means.
* Ha : **statistical difference** between the two observed sample means.

1. The input data is numerical and continuous.
2. Each sample data was selected randomly from the population data.
3. The input data is considered to be normally distributed.
4. Each sample size is reasonably large.
5. The variance of the input data should be very similar.

**Pair t-test**, we pair observations in one dataset with observations in another. We use the pair t-test when:

* Comparing measurements on the same subjects across a single span of time (fuel efficiency before and after an oil change)
* Comparing different methods of measurement (testing tire pressure using two different tire pressure gauges)

In an unpaired t-test, the means are calculated by adding up all observations in a dataset, and dividing by the number of data points.

In a paired t-test, the means are determined from the difference between each paired observation.

* H0 : The **difference** between our paired observations (the true mean difference, or "μd") is **equal to** **zero**.
* Ha : The **difference** between our paired observations (the true mean difference, or "μd") is **not equal to zero**.

The required arguments are slightly different from the unpaired two-sample t-test:

* **x** is the first numeric vector of sample data.
* **y** is the second numeric vector of sample data.
* **paired** tells the t.test() function to perform a paired t-test. This value must be set to TRUE.
* **alternative** tells the t.test() function if the hypothesis is one-sided (one-tailed) or two-sided (two-tailed). The options for the alternative argument are "two.sided," "less," or "greater." By default, the t.test() function assumes a two-sided t-test.

**Analysis of variance (ANOVA) test,** to compare the means of a continuous numerical variable across a number of groups.

* **one-way ANOVA** is used to test the means of a single dependent variable across a single independent variable with multiple groups. (fuel efficiency of different cars based on vehicle class).
* **two-way ANOVA** does the same thing, but for two different independent variables (vehicle braking distance based on weather conditions and transmission type).
* H0 : The means of all groups are equal, or µ1 = µ2 = … = µn.
* Ha : At least one of the means is different from all other groups.

1. The dependent variable is numerical and continuous, and the independent variables are categorical.
2. The dependent variable is considered to be normally distributed.
3. The variance among each group should be very similar.

Graphical user interface, text, application, email

Description automatically generated

mtcars\_filt <- mtcars[,c("hp","cyl")] #filter columns from mtcars dataset

mtcars\_filt$cyl <- factor(mtcars\_filt$cyl) #convert numeric column to factor

aov(hp ~ cyl,data=mtcars\_filt) #compare means across multiple levels

summary(aov(hp ~ cyl,data=mtcars\_filt)) #get the p-value

* **Correlation**

**ideal positive correlation** where r = 1

Chart, scatter chart

Description automatically generated

**ideal negative correlation** where r = -1

Chart, scatter chart

Description automatically generated

**no correlation** where r ≈ 0

Table

Description automatically generated

plt <- ggplot(mtcars,aes(x=hp,y=qsec)) #import dataset into ggplot2

plt + geom\_point() #create scatter plot

cor(mtcars$hp,mtcars$qsec) #calculate correlation coefficient

#cor(x-axis, y-axis)

**Correlation matrix** is a lookup table where the variable names of a data frame are stored as rows and columns, and the intersection of each variable is the corresponding Pearson correlation coefficient.

used\_matrix <- as.matrix(used\_cars[,c(“Selling\_Price”, “Present\_Price”, “Miles\_Driver”)]) #Convert data frame into numeric matrix

cor(used\_matrix)

A screenshot of a computer

Description automatically generated with medium confidence

* **Linear Regression**

**Simple linear regression** builds a linear regression model with one independent variable.

**Multiple linear regression** builds a linear regression model with two or more independent variables.

1. The input data is numerical and continuous.
2. The input data should follow a linear pattern.
3. There is variability in the independent *x* variable*.* This means that there must be more than one observation in the x-axis and they must be different values.
4. The residual error (the distance from each data point to the line) should be normally distributed.

lm(qsec ~ hp,mtcars) #create linear model

summary(lm(qsec~hp,mtcars)) #summarize linear model

Text

Description automatically generated

model <- lm(qsec ~ hp,mtcars) #create linear model

yvals <- model $ coefficients['hp'] \* mtcars $ hp +

model $ coefficients['(Intercept)'] #determine y-axis values from linear model (y=mx+b)

plt <- ggplot(mtcars,aes(x=hp,y=qsec)) #import dataset into ggplot2

plt + geom\_point() + geom\_line(aes(y=yvals), color = "red") #plot scatter and linear model

Multiple linear regression becomes *y = m1x1 + m2x2 + … + mnxn + b*, for all independent *x* variables and their *m* coefficients.

lm(qsec ~ mpg + disp + drat + wt + hp, data=mtcars) #generate multiple linear regression model

A screenshot of a computer

Description automatically generated with medium confidence

**Chi-squared test** is to compare the distribution of frequencies across two groups.

H0 : There **is no difference** in frequency distribution between both groups.

Ha : There **is a difference** in frequency distribution between both groups

1. Each subject within a group contributes to only one frequency. In other words, the sum of all frequencies equals the total number of subjects in a dataset.
2. Each unique value has an equal probability of being observed.
3. There is a minimum of five observed instances for every unique value for a 2x2 chi-squared table.
4. For a larger chi-squared table, there is at least one observation for every unique value and at least 80% of all unique values have five or more observations.

The most straightforward implementation of chisq.test() function is passing the function to a contingency table.

A **contingency table** is a frequency table produced using R's table() function.

To test whether there is a statistical difference in the distributions of vehicle class across 1999 and 2008 from mpg dataset:

table(mpg$class, mpg$year) #generate contingency table

tbl <- table(mpg$class, mpg$year) #generate contingency table

chisq.test(tbl) #compare categorical distributions

**A/B testing** is a randomized controlled experiment that uses a control (unchanged) and experimental (changed) group to test potential changes using a success metric.

* If success metric is **numerical** and **sample size is small**, a **z-score summary** **statistic** can be sufficient to compare the mean and variability of both groups.
* If success metric is **numerical** and **sample size is large**, a **two-sample t-test** should be used to compare the distribution of both groups.
* If success metric is **categorical**, use a **chi-squared test** to compare the distribution of categorical values between both groups.

In data science, researchers use **retrospective analysis** to analyze and interpret a previously generated dataset where the outcome is already known.

Depending on the dataset and input variables, there is a limitless number of statistical questions that can be asked from the data:

* Are two groups statistically different? Use a t-test with one dichotomous independent variable and one continuous dependent variable.
* Can one continuous dependent variable be predicted using another independent variable? What about multiple independent variables and one dependent variable? Use regression analysis.
* Are there multiple categorical variables tightly linked in a dataset? Are the distributions of the different categorical variables equal? We can test with chi-squared.