Bentley University MA255 in R

Content extracted from the How to Data Website

This PDF generated on 03 August 2023

Contents

MA255 is an undergraduate statistics course at Bentley University on the Design of Experiments. The description from the course catalog can be found here.

The course covers various experimental designs including factorial and fractional factorial designs, interaction among factors, and applications in management (including cost savings and policy making) as well as in marketing.

The sequence of topics below is not necessarily the final version; this topic page is under construction.

Summarizing data and exploratory analysis

- How to summarize a column
- How to compute summary statistics
- How to summarize and compare data by groups
- How to create bivariate plots to compare groups

Experiments with one treatment factor

- How to check the assumptions of a linear model
- How to compute the power of a test comparing two population means
- How to perform an analysis of covariance (ANCOVA)
- How to perform pairwise comparisons
- How to perform post-hoc analysis with Tukey's HSD test
- How to test for a treatment effect in a single factor design

Analyzing data from a larger design

- How to plot interaction effects of treatments
- How to analyze the sample means of different treatment conditions
- How to compare two nested linear models
- How to conduct a mixed designs ANOVA
- How to conduct a repeated measures ANOVA
- How to perform a planned comparison test

Content last modified on 03 August 2023.

How to summarize a column

Description

When provided with a dataset in which you want to focus on one column, how would you compute descriptive statistics for that column?

Related task:

- How to compute summary statistics
- How to summarize and compare data by groups

Solution in pure R

The solution below uses an example dataset about the teeth of 10 guinea pigs at three Vitamin C dosage levels (in mg) with two delivery methods (orange juice vs. ascorbic acid). (See how to quickly load some sample data (on website).)

```
df <- ToothGrowth
```

Let us consider qualitative and quantitative variables separately.

Consider the qualitative column “supp” in the dataset (which type of supplement the animal received). To count the distribution of each categorical value, use `table()`:

```
table(df$supp) # OR summary(df$supp)
```

```
OJ VC 
30 30 
```

The output says that there are 30 observations under each of the two levels, Orange Juice and Ascorbic Acid.

If you wish to jointly summarize two categorical columns, provide both to `table()`:

```
table(df$supp, df$dose)
```

```
     0.5  1  2
  OJ  10 10 10
  VC  10 10 10
```

This informs us that there are 10 observations for each of the combinations.

Note: If there are more than 2 categorical variables of interest, you can use `ftable()` instead.

Now consider the quantitative column `len` in the dataset (the length of the animal’s tooth). We can compute summary statistics for it just as we can for a whole dataframe (as we cover in how to compute summary statistics).

```
summary(df$len)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   4.20   13.07   19.25   18.81   25.27   33.90 
```

The individual functions for mean, standard deviation, etc. covered under “how to compute summary statistics” apply to individual columns as well. For example, we can compute quantiles:

```
quantile(df$len) # quantiles
```

```
    0%    25%    50%    75%   100% 
 4.200 13.075 19.250 25.275 33.900 
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute summary statistics

Description

The phrase "summary statistics" usually refers to a common set of simple computations that can be done about any dataset, including mean, median, variance, and some of the others shown below.

Related tasks:

- How to summarize a column
- How to summarize and compare data by groups

Solution in pure R

We first load a famous dataset, Fisher's irises, just to have some example data to use in the code that follows. (See how to quickly load some sample data (on website).)

```r
library(datasets)
data(iris)
```

How big is the dataset? The output shows number of rows then number of columns.

```r
dim(iris)  # Short for "dimensions."
```

```
[1] 150   5
```

What are the columns and their data types? Can I see a sample of each column?

```r
str(iris)  # Short for "structure."
```

```
'data.frame':	150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

What do the first few rows look like?

```r
head(iris) # Gives 5 rows by default.  You can do head(iris,10), etc.
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1          3.5         1.4          0.2         setosa
2 4.9          3.0         1.4          0.2         setosa
3 4.7          3.2         1.3          0.2         setosa
4 4.6          3.1         1.5          0.2         setosa
5 5.0          3.6         1.4          0.2         setosa
6 5.4          3.9         1.7          0.4         setosa
```

The easiest way to get summary statistics for an R `data.frame` is with the `summary` function.

```
summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

The columns from the original dataset are the column headings in the summary output, and the statistics computed for each are listed below those headings.

We can also compute these statistics (and others) one at a time for any given set of data points. Here, we let `xs` be one column from the above `data.frame` but you could use any vector or list.

```
xs <- iris$Sepal.Length

mean( xs )           # mean, or average, or center of mass
median( xs )         # 50th percentile
quantile( xs, 0.25 ) # compute any percentile, such as the 25th
var( xs )            # variance
sd( xs )             # standard deviation, the square root of the variance
sort( xs )           # data in increasing order
sum( xs )            # sum, or total
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to summarize and compare data by groups

Description

When given a set of data that has different treatment conditions and an outcome variable, we need to perform some exploratory data analysis. How would you quantitatively compare the treatment conditions with regards to the outcome variable?

Related tasks:

- How to compute summary statistics

Solution in pure R

The solution below uses an example dataset about the teeth of 10 guinea pigs at three Vitamin C dosage levels (in mg) with two delivery methods (orange juice vs. ascorbic acid). (See how to quickly load some sample data (on website).)

```
df <- ToothGrowth
```

To obtain the descriptive statistics of the quantitative column (`len` for length of teeth) based on the treatment levels (`supp`), we can use either the `tapply` or `favstats` functions.

```
attach(df)
tapply(len, supp, summary)
```

```
$OJ
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   8.20   15.53   22.70   20.66   25.73   30.90 

$VC
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   4.20   11.20   16.50   16.96   23.10   33.90 
```

You can replace `summary` in the call to `tapply` with `mean`, `median`, `max`, `min`, or `quantile` to get just one value. An example is shown below for quantiles.

```
tapply(len, supp, quantile, prob = 0.25, data=df) # 1st quartile
```

```
    OJ     VC 
15.525 11.200 
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to create bivariate plots to compare groups

Description

Suppose we have a dataset with different treatment conditions and an outcome variable, and we want to perform exploratory data analysis. How would we visually compare the treatment conditions with regards to the outcome variable?

Related tasks:

- How to create basic plots (on website)
- How to add details to a plot (on website)
- How to create a histogram (on website)
- How to create a box (and whisker) plot (on website)
- How to change axes, ticks, and scale in a plot (on website)
- How to plot interaction effects of treatments

Solution in R using lattice and gplots

We use a built-in dataset called `ToothGrowth` that discusses the length of the teeth (`len`) in each of 10 guinea pigs at three Vitamin C dosage levels (0.5, 1, and 2 mg) with two delivery methods - orange juice or ascorbic acid (`supp`).

```r
# You can replace this example data frame with your own data
df <- ToothGrowth
```

If you wish to understand the distribution of the length of the tooth based on the delivery methods, you can construct a bivariate histogram plot.

```r
# install.packages( "lattice" ) # if you have not already done this
library(lattice)
histogram( ~ len | supp, data = df)
```

To visualize the summary statistics of the length of the tooth based on the delivery methods, you can construct a bivariate box plot.

```r
bwplot(df$len ~ df$supp)
# Or the following code produces a similar figure, using the mosaic package:
# boxplot(len ~ supp, data = df)
```

To plot the means for both treatment levels of `supp` for the `len` column, we load the `gplots` package and use the `plotmeans` function.

```r
# install.packages( "gplots" ) # if you have not already done this
library(gplots)
plotmeans(df$len ~ df$supp)
```

```
Attaching package: ‘gplots’


The following object is masked from ‘package:stats’:

    lowess
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to check the assumptions of a linear model

Description

If you plan to use a linear model to describe some data, it's important to check if it satisfies the assumptions for linear regression. How can we do that?

Solution in pure R

When performing a linear regression, the following assumptions should be checked.

1. We have two or more columns of numerical data of the same length.

The solution below uses an example dataset about car design and fuel consumption from a 1974 Motor Trend magazine. (See how to quickly load some sample data (on website).) We can see that our columns all have the same length.

```
df <- mtcars
head(df)
```

```
                  mpg  cyl disp hp  drat wt    qsec  vs am gear carb
Mazda RX4         21.0 6   160  110 3.90 2.620 16.46 0  1  4    4
Mazda RX4 Wag     21.0 6   160  110 3.90 2.875 17.02 0  1  4    4
Datsun 710        22.8 4   108   93 3.85 2.320 18.61 1  1  4    1
Hornet 4 Drive    21.4 6   258  110 3.08 3.215 19.44 1  0  3    1
Hornet Sportabout 18.7 8   360  175 3.15 3.440 17.02 0  0  3    2
Valiant           18.1 6   225  105 2.76 3.460 20.22 1  0  3    1
```

2. Scatter plots we've made suggest a linear relationship.

Scatterplots are covererd in how to create basic plots (on website), but after making the model, we can also examine the residuals.

So let's make the model. Our predictors will be the number of cylinders and the weight of the car and the response will be miles per gallon. (See also how to fit a linear model to two columns of data (on website).)

```
model = lm(mpg~ cyl + wt, data=df)
```

We test for linearity with residual plots. We show just one residual plot here; you should make one for each predictor. R's plot function knows how to create residual plots. (See also how to compute the residuals of a linear model (on website).)

```
plot(model, which = 1)
```

3. After making the model, the residuals seem normally distributed.

We can check this by constructing a QQ-plot, which compares the distribution of the residuals to a normal distribution. Here we use SciPy, but there are other methods; see how to create a QQ-plot (on website).

```
plot(model, which = 2)
```

4. After making the model, the residuals seem homoscedastic.

This assumption is sometimes called “equal variance,” and can be checked by the `regplot` function in Seaborn. We must first standardize the residuals, which we can do with NumPy. We want to see a plot with no clear pattern; a cone shape to the data would indicate heteroscedasticity, the opposite of homoscedasticity.

```
plot(model, which = 3) # assumption of equal variance
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compute the power of a test comparing two population means

Description

When creating a factorial design, it is important that it has adequate power to detect significant main effects and interaction effects of interest. How can we calculate the power of a two-sample t test that we aim to perform in such a situation?

Related tasks:

- How to choose the sample size in a study with two population means (on website)

Solution in pure R

We use the `power.t.test` function in R. It embodies a relationship among five variables; you provide any four of them and it will compute the fifth to be consistent with the first four, regarding the two-sample t-test you plan

For this example, let's say that:

- You plan to create a balanced 4×2 factorial experiment with 32 subjects.
- You wish to be able to detect a difference
- You want to know the expected power for the test of a main effect of factor A.
- Your significance level is $\alpha = 0.05$.

We proceed as follows.

```
# install.packages('pwr') # if you have not already installed it
library(pwr)

obs <- 32       # number of subjects (or observations)
effect <- 0.25  # effect size
alpha <- 0.05   # significance level
ratio <- 1      # ratio of the number of observations in one sample to the other

# We leave power unspecified, so that power.t2n.test will compute it for us:
pwr.t2n.test(n1=obs, n2=obs, d=effect, sig.level=alpha, power=NULL)
```

```
     t test power calculation 

             n1 = 32
             n2 = 32
              d = 0.25
      sig.level = 0.05
          power = 0.1662985
    alternative = two.sided
```

The power is 0.1663, which means that the probability of rejecting the null hypothesis when in fact it is false OR the probability of avoiding a Type II error is 0.1663.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to perform an analysis of covariance (ANCOVA)

Description

Recall that covariates are variables that may be related to the outcome but are unaffected by treatment assignment. In a randomized experiment with one or more observed covariates, an analysis of covariance (ANCOVA) addresses this question: How would the mean outcome in each treatment group change if all groups were equal with respect to the covariate? The goal is to remove any variability in the outcome associated with the covariate from the unexplained variability used to determine statistical significance.

Related tasks:

- How to do a one-way analysis of variance (ANOVA) (on website)
- How to compare two nested linear models
- How to conduct a mixed designs ANOVA
- How to conduct a repeated measures ANOVA

Solution in pure R

The solution below uses an example dataset about car design and fuel consumption from a 1974 Motor Trend magazine. (See how to quickly load some sample data (on website).)

```r
df <- mtcars
df$vs <- as.factor(df$vs)
```

Let's use ANCOVA to check the effect of the engine type (0 = V-shaped, 1 = straight, in the variable `vs`) on the miles per gallon when considering the weight of the car as a covariate. We will use the `ancova` function from the `pingouin` package to conduct the test.

```r
cov.model <- lm(mpg ~ wt + vs, data = df)
anova(cov.model)
```

```
          Df Sum Sq    Mean Sq     F value    Pr(>F)
wt         1 847.72525 847.725250 109.704168 2.284396e-11
vs         1  54.22806  54.228061   7.017656 1.292580e-02
Residuals 29 224.09388   7.727375         NA           NA
```

The *p*-value for each variable can be found in the final column of the output, called `Pr(>F)`.

The *p*-value for the `wt` variable tests the null hypothesis, "The quantities `wt` and `mpg` are not related." Since it is below 0.05, we reject the null hypothesis, and conclude that `wt` is significant in predicting `mpg`.

The *p*-value for the `vs` variable tests the null hypothesis, "The quantities `vs` and `mpg` are not related if we hold `wt` constant." Since it is below 0.05, we reject the null hypothesis, and conclude that `vs` is significant in predicting `mpg` even among cars with equal weight (`wt`).

If we wish to create a 2-factor ANCOVA model, we can test to see if the engine type (0 = V-shaped, 1 = straight) and transmission type (0 = automatic, 1 = manual) have an effect on the Miles/gallon per car when considering the weight of the car as a covariate.

```r
cov.model.2 <- lm(mpg ~ wt + vs + am, data = df)
anova(cov.model.2)
```

```
          Df Sum Sq     Mean Sq    F value    Pr(>F)
wt         1 847.725250 847.725250 109.729918 3.420018e-11
vs         1  54.228061  54.228061   7.019303 1.310627e-02
am         1   7.778149   7.778149   1.006807 3.242621e-01
Residuals 28 216.315728   7.725562         NA           NA
```

The *p*-values are again in the final column of output. They show that at the 5% significance level, we would conclude that engine type (`vs`) significantly impacts the Miles/gallon per car while accounting for the weight of the car (`wt`) but the transmission type (`am`) does not.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to perform pairwise comparisons

Description

When analyzing data from a completely randomized single-factor design, suppose that you have performed an ANOVA and noticed that there's a significant difference between at least one pair of treatment levels. How can pairwise comparisons help us explore which pairs of treatment levels are different?

Related tasks:

- How to do a one-way analysis of variance (ANOVA) (on website)
- How to perform post-hoc analysis with Tukey's HSD test

Solution in pure R

The solution below uses an example dataset that details the counts of insects in an agricultural experiment with six types of insecticides, labeled A through F. (This is one of the datasets built into R for use in examples like this one.)

```
df <- InsectSprays
head( df, 10 )
```

```
   count spray
1  10    A
2   7    A
3  20    A
4  14    A
5  14    A
6  12    A
7  10    A
8  23    A
9  17    A
10 20    A
```

Before we perform any post hoc analysis, we need to see if the count of insects depends on the type of insecticide given by conducting a one way ANOVA. (See also how to do a one-way analysis of variance (ANOVA) (on website).)

```
aov1 = aov(count ~ spray, data = df)
summary(aov1)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)    
spray        5   2669   533.8    34.7 <2e-16 ***
Residuals   66   1015    15.4                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

At the 5% significance level, we see that the count differs according to the type of insecticide used. We assume that the model assumptions are met, but do not verify that here.

If we would like to compare the pairs without any corrections, we can use the `pairwise.t.test` function built into R.

```r
pairwise.t.test(df$count, df$spray, p.adj="none")
```

```
    Pairwise comparisons using t tests with pooled SD

data:  df$count and df$spray

  A       B       C       D       E
B 0.604   -       -       -       -
C 7.3e-11 8.5e-12 -       -       -
D 9.8e-08 1.2e-08 0.081   -       -
E 2.8e-09 3.3e-10 0.379   0.379   -
F 0.181   0.408   2.8e-13 4.0e-10 1.1e-11

P value adjustment method: none
```

Techniques to adjust the above table for multiple comparisons include the Bonferroni correction, Fisher's Least Significant Difference (LSD) method, Dunnett's procedure, and Scheffe's method. These can be used in place of "none" for the `p.adj` argument; see details here.

You can also determine the magnitude of these differences; see how to perform post-hoc analysis with Tukey's HSD test.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to perform post-hoc analysis with Tukey's HSD test

Description

If we run a one-way ANOVA test and find that there is a significant difference between population means, we might want to know which means are actually different from each other. One way to do so is with Tukey's Honestly Significant Differences (HSD) method. It creates confidence intervals for each pair of samples, while controlling for Type I error rate across all pairs. Thus the resulting intervals are a little wider than those produced using Fisher's LSD method. How do we make these confidence intervals, with an appropriate visualization?

Solution in pure R

We load here the same data that appears in the solution for how to perform pairwise comparisons. That solution used ANOVA to determine which pairs of groups have significant differences in their means; follow its link for more details.

```r
# Load an inbuilt data set called InsectSprays and assign it to the variable df
df <- InsectSprays
head( df, 10 )
```

```
   count spray
1  10    A
2   7    A
3  20    A
4  14    A
5  14    A
6  12    A
7  10    A
8  23    A
9  17    A
10 20    A
```

We now want to perform an unplanned comparison test on the data to determine the magnitudes of the differences between pairs of groups. We do this by applying Tukey's HSD approach to perform pairwise comparisons and generate confidence intervals that maintain a specified experiment-wide error rate. We use R's built-in `TukeyHSD` function, and we give it the same ANOVA results that we computed in the solution for how to perform pairwise comparisons.

```r
aov1 <- aov(count ~ spray, data = df)
TukeyHSD(aov1, "spray", ordered=TRUE, conf.level = 0.95)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = count ~ spray, data = df)

$spray
          diff       lwr       upr     p adj
E-C  1.4166667 -3.282742  6.116075 0.9488669
D-C  2.8333333 -1.866075  7.532742 0.4920707
A-C 12.4166667  7.717258 17.116075 0.0000000
B-C 13.2500000  8.550591 17.949409 0.0000000
F-C 14.5833333  9.883925 19.282742 0.0000000
D-E  1.4166667 -3.282742  6.116075 0.9488669
A-E 11.0000000  6.300591 15.699409 0.0000000
B-E 11.8333333  7.133925 16.532742 0.0000000
F-E 13.1666667  8.467258 17.866075 0.0000000
A-D  9.5833333  4.883925 14.282742 0.0000014
B-D 10.4166667  5.717258 15.116075 0.0000002
F-D 11.7500000  7.050591 16.449409 0.0000000
B-A  0.8333333 -3.866075  5.532742 0.9951810
F-A  2.1666667 -2.532742  6.866075 0.7542147
F-B  1.3333333 -3.366075  6.032742 0.9603075
```

Because the above table contains a lot of information, it's often helpful to visualize these intervals. R lets us do so by simply calling `plot` on the above table. We add a few plotting parameters to improve its appearance.

```
plot( TukeyHSD(aov1, "spray", ordered=TRUE, conf.level = 0.95),
      las=1, cex.axis=0.9 )
```

Confidence intervals that cross the vertical, dashed line at $x = 0$ are those in which the means across those groups may be equal. Other intervals have mean differences whose 95% confidence intervals do not include zero.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to test for a treatment effect in a single factor design

Description

Suppose you are given a dataset that has more than one treatment level and you wish to see if there is a unit-level treatment effect. How would you check that?

Solution in R using perm

The solution below uses an example dataset about the teeth of 10 guinea pigs at three Vitamin C dosage levels (in mg) with two delivery methods (orange juice vs. ascorbic acid). (See how to quickly load some sample data (on website).)

```
df <- ToothGrowth
```

In this dataset, there are only two treatments (orange juice and ascorbic acid, in the variable `supp`). We can therefore perrform a two-sample t test. But first we must filter the outcome variable `len` (tooth length) based on `supp`.

```
t.test(len ~ supp, data=df)
```

```
    Welch Two Sample t-test

data:  len by supp
t = 1.9153, df = 55.309, p-value = 0.06063
alternative hypothesis: true difference in means between group OJ and group VC is not equal to 0
95 percent confidence interval:
 -0.1710156  7.5710156
sample estimates:
mean in group OJ mean in group VC 
        20.66333         16.96333 
```

The p-value is reported in the first row of numerical output as 0.06063. Because this is greater than 0.05, at a 5% significance level, we see that the length of the tooth does not differ between the two delivery methods.

Since the `t.test` makes some assumptions, we can use the `permTS` function instead. It can conduct a permutation or randomization test, but it requires us to load the `perm` package first.

```
# install.packages("perm") # If you have not already installed it
library(perm)
permTS(len ~ supp, data=df)
```

```
    Permutation Test using Asymptotic Approximation

data:  len by supp
Z = 1.8734, p-value = 0.06102
alternative hypothesis: true mean supp=OJ - mean supp=VC is not equal to 0
sample estimates:
mean supp=OJ - mean supp=VC 
                        3.7 
```

The *p*-value is reported in the first row of numerical output as 0.06102. Because this is greater than 0.05, at a 5% significance level, we see that the length of the tooth does not differ between the two delivery methods. We assume that the model assumptions are met but not shown in this task.

If there are multiple levels (2 or more), you can apply the parametric ANOVA test which in this case will provide a similar *p*-value.

```
aov1 <- aov(len ~ supp, data = df)
summary(aov1)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
supp         1    205  205.35   3.668 0.0604 .
Residuals   58   3247   55.98
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The *p*-value for `supp` is shown at the end of the `supp` row, in the `Pr(>F)` column. Because it is 0.0604, which is greater than 0.05, at a 5% significance level, we see that the length of the tooth does not differ between the delivery methods.

However, if the assumptions of ANOVA are not met, we can utilize the non parametric approach via the Kruskal-Wallis Test.

```
kruskal.test(len ~ supp, data = df)
```

```

    Kruskal-Wallis rank sum test

data:  len by supp
Kruskal-Wallis chi-squared = 3.4454, df = 1, p-value = 0.06343
```

The *p*-value is the last part of the output, and is 0.06343. Because it is greater than 0.05, at a 5% significance level, we see that the length of the tooth does not differ between the delivery methods.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to plot interaction effects of treatments

Description

When there are multiple treatment conditions with multiple levels and you wish to undertsand the interaction effects of each of them, a plot can be useful. How can we create the right kind of plot for that situation?

- How to create basic plots (on website)
- How to add details to a plot (on website)
- How to create a histogram (on website)
- How to create a box (and whisker) plot (on website)
- How to change axes, ticks, and scale in a plot (on website)
- How to create bivariate plots to compare groups

Solution in R using ggpubr

The solution below uses an example dataset about the teeth of 10 guinea pigs at three Vitamin C dosage levels (in mg) with two delivery methods (orange juice vs. ascorbic acid). (See how to quickly load some sample data (on website).)

```
df <- ToothGrowth
```

To plot the interaction effects among tooth length, supplement, and dosage, we can use the `ggline` function in the `ggpubr` package. You can change the `x` and `color` inputs below depending on your goals, but the `y` input should always be the dependent variable.

```
# install.packages("ggpubr") # If you have not already installed it
library(ggpubr)
ggline(df, x="dose", y="len", color="supp", add=c("mean"))
```

```
Loading required package: ggplot2
```

Looking at the output, we first see that there is an interaction effect because the two supp lines intersect. We also see that there is a difference in length when giving 0.5mg and 1mg dosage of either of the two delivery methods. However, there is barely any difference between the delivery methods when the dosage level is 2mg.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to analyze the sample means of different treatment conditions

Description

In a single-factor experiment with three or more treatment levels, how can we compare them to see which one impacts the outcome variable the most?

Solution in R using gplots and emmeans

The solution below uses an example dataset about the teeth of 10 guinea pigs at three Vitamin C dosage levels (in mg) with two delivery methods (orange juice vs. ascorbic acid). (See how to quickly load some sample data (on website).)

```
df <- ToothGrowth
```

To visually plot the means of the length of the tooth based on the Vitamin C dosage levels we can create a pointplot. We will use the `gplots` package. In the code below, `bars=TRUE` gives 95% confidence intervals for the means.

```
# install.packages("gplots") # If you have not yet installed it
library(gplots)
plotmeans(len~dose, data=df, bars=TRUE)
```

```
Attaching package: ‘gplots’


The following object is masked from ‘package:stats’:

    lowess
```

The point plot informs us that as the dosage levels increase, the tooth length also increases.

To obtain the actual numbers, we can use the code below. The first line converts the numerical dosage values to a categorical variable, which may not be necessary if your data was already categorical.

```
df$dose.factor = as.factor(df$dose)
aov1 = aov(len~dose.factor, data=df)
model.tables(aov1, type='means')
```

```
Tables of means
Grand mean

18.81333

 dose.factor
dose.factor
   0.5      1      2
10.605 19.735 26.100
```

If you wish to display the difference between the overall mean and the group means, you can simply omit the `type='means'` parameter.

```
model.tables(aov1)
```

```
Tables of effects

 dose.factor
dose.factor
   0.5      1      2
-8.208  0.922  7.287
```

To also see the specific values for the confidence intervals plotted earlier, we can use the `emmeans` package (Estimated Marginal Means or Least-Squares Means).

```
# install.packages("emmeans") # If you have not yet installed it
library(emmeans)
emmeans(aov1,'dose.factor')
```

```
 dose.factor emmean    SE df lower.CL upper.CL
 0.5           10.6 0.949 57     8.71     12.5
 1             19.7 0.949 57    17.84     21.6
 2             26.1 0.949 57    24.20     28.0

Confidence level used: 0.95
```

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to compare two nested linear models

Description

Model A is said to be "nested" in model B if the predictors included in A are a subset of those included in B. In such a situation, how can we determine if the larger model (in this case B) is significantly better than the smaller (reduced) model? We can use an Extra Sums of Squares test, also called a partial F-test, to compare two nested linear.

This technique will also help us with another question. If we have a multivarate linear model,

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k,$$

how can we test the influence of only some of the coefficients? If we remove some of the coefficients, we have a smaller model nested in the larger one, so the question is the same.

Related tasks:

- How to do a one-way analysis of variance (ANOVA) (on website)
- How to conduct a mixed designs ANOVA
- How to conduct a repeated measures ANOVA
- How to perform an analysis of covariance (ANCOVA)

Solution in pure R

The solution below uses an example dataset about car design and fuel consumption from a 1974 Motor Trend magazine. (See how to quickly load some sample data (on website).)

We will create two models, one nested inside the other, in a natural way in this example. But this is not the only way to create nested models; it is just an example.

```
# install.packages("datasets") # if you have not done so already
library(datasets)
data(mtcars)
df <- mtcars
```

Consider a model using number of cylinders (cyl) and weight of car (wt) to predict its fuel efficiency (mpg). We create this model and perform an ANOVA to see if the predictors are significant.

```
# Build the model
add_model <- lm(mpg ~ cyl + wt, data = df)
# Perform an ANOVA
anova(add_model)
```

```
          Df Sum Sq   Mean Sq    F value   Pr(>F)
cyl        1 817.7130 817.712952 124.04369 5.424327e-12
wt         1 117.1623 117.162269  17.77303 2.220200e-04
Residuals 29 191.1720   6.592137        NA           NA
```

The final column of output suggests that both predictors are significant. A natural question to ask is whether the two predictors have an interaction effect. Let's create a model containing the interaction term.

```r
# Build the model with interaction
int_model <- lm(mpg ~ cyl * wt, data = df)
# Perform an ANOVA
anova(int_model)
```

```
          Df Sum Sq    Mean Sq    F value    Pr(>F)
cyl        1 817.71295 817.712952 145.856269 1.280635e-12
wt         1 117.16227 117.162269  20.898350 8.942713e-05
cyl:wt     1  34.19577  34.195767   6.099533 1.988242e-02
Residuals 28 156.97620   5.606293         NA           NA
```

As seen in the final column of output, there is a significant interaction between the two predictors.

We now have one model (`add_model`) nested inside a larger model (`int_model`). To check which model is better, we can conduct an ANOVA comparing the two models.

```r
# Use ANOVA to compare the models
anova(add_model, int_model)
```

```
  Res.Df RSS      Df Sum of Sq F        Pr(>F)
1 29     191.1720 NA       NA        NA         NA
2 28     156.9762  1 34.19577  6.099533 0.01988242
```

We have just performed this hypothesis test:

H_0 = the two models are equally useful for predicting the outcome

H_a = the larger model is significantly better than the smaller model

In the final column of the output, called `Pr(>F)`, the only number in that column is our test statistic, 0.01988. Since is below our chosen threshold of 0.05, we reject the null hypothesis, and prefer to use the second model.

This method can be used to check if covariates should be included in the model, or if additional variables should be added as well.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to conduct a mixed designs ANOVA

Description

When you have a dataset that includes the responses of a mixed design test, where one factor is a within-subjects factor and the other is a between-subjects factor, and you wish check if there is a significant difference for both factors, this requires a Mixed Design ANOVA. How can we conduct one?

Related tasks:

- How to do a one-way analysis of variance (ANOVA) (on website)
- How to do a two-way ANOVA test with interaction (on website)
- How to do a two-way ANOVA test without interaction (on website)
- How to compare two nested linear models using ANOVA
- How to conduct a repeated measures ANOVA
- How to perform an analysis of covariance (ANCOVA)

Solution in pure R

We create the data for a hypothetical 2×2 mixed design with the following attributes.

- Between-subjects treatment factor: Type of music played (classical vs. rock)
- Within-subjects treatment factor: Type of room (light vs. no light)
- Outcome variable: Heart rate of subject

```
subject    <- as.factor(c(1,2,3,4,5,6,7,8,9,10,1,2,3,4,5,6,7,8,9,10))
music      <- c('Classical','Rock','Classical','Rock','Classical','Rock','Classical',
                'Rock','Classical','Rock','Classical','Rock','Classical','Rock','Classical',
                'Rock','Classical','Rock','Classical','Rock')
room.type  <- c('Light','Light','Light','Light','Light','Light','Light','Light','Light',
                'Light','No Light','No Light','No Light','No Light','No Light','No Light',
                'No Light','No Light','No Light', 'No Light')
heart.rate <- c(78,60,85,75,99,94,75,84,100,76,90,109,99,94,113,92,91,88,89,90)
df <- data.frame(subject,music,room.type,heart.rate)
head(df)
```

```
  subject music     room.type heart.rate
1 1       Classical Light     78
2 2       Rock      Light     60
3 3       Classical Light     85
4 4       Rock      Light     75
5 5       Classical Light     99
6 6       Rock      Light     94
```

We conduct a two-way mixed-design ANOVA as shown below. The specific parameters have these meanings:

- The dependent variable is `heart.rate`.
- The within-group factor is `room.type`.
- The between-group factor is `music`.
- The `Error()` term is critical in differentiating between a between subjects and within subjects model. It tells R that there is one observation per `subject` for each level of `room.type`.

```
aov_mixed <- aov(heart.rate ~ room.type*music + Error(subject/room.type), data=df)
summary(aov_mixed)
```

```
Error: subject
          Df Sum Sq Mean Sq F value Pr(>F)
music      1  162.4   162.4   1.587  0.243
Residuals  8  819.0   102.4

Error: subject:room.type
                Df Sum Sq Mean Sq F value Pr(>F)
room.type        1  832.1   832.1   6.416 0.0351 *
room.type:music  1   76.0    76.0   0.586 0.4658
Residuals        8 1037.4   129.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The output informs us that, on average, the subjects that listened to classical music did not significantly differ ($p = 0.243 > 0.05$) from those that listened to rock music. However, there is, on average, a significant difference ($p = 0.0351 < 0.05$) between each of the subject's heart rate when put in a room with or without light. Additionally, since the interaction term is not significant ($p = 0.4658 > 0.05$), we can use the additive (no interaction) model.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to conduct a repeated measures ANOVA

Description

In a repeated measures test, the same subject receives multiple treatments. When you have a dataset that includes the responses of a repeated measures test where the measurements are dependent (within subjects design), you may wish to check if there is a difference in the treatment effects. How would you conduct a repeated measures ANOVA to answer that question?

Related tasks:

- How to do a one-way analysis of variance (ANOVA) (on website)
- How to do a two-way ANOVA test with interaction (on website)
- How to do a two-way ANOVA test without interaction (on website)
- How to compare two nested linear models using ANOVA
- How to conduct a mixed designs ANOVA
- How to perform an analysis of covariance (ANCOVA)

Solution in R using rstatix and tidyr and car

We create a hypothetical repeated measures dataset where the 5 subjects undergo all 4 skin treatments and their rating of the treatment is measured.

```
subject <- as.factor(c(1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5))
skin.treatment <- c('W','X','Y','Z','W','X','Y','Z','W','X',
                    'Y','Z','W','X','Y','Z','W','X','Y','Z')
rating <- c(7,5,8,4,8,10,7,5,7,6,5,4,7,7,4,5,8,8,6,6)
df <- data.frame(subject,skin.treatment,rating)
head(df)
```

```
  subject skin.treatment rating
1 1       W               7
2 1       X               5
3 1       Y               8
4 1       Z               4
5 2       W               8
6 2       X              10
```

Before we conduct a repeated measures ANOVA, we need to decide which approach to use - Univariate or Multivariate. We decide this using Mauchly's test of sphericity. If we fail to reject the null hypothesis then we use the univariate approach.

- H_0 = the sphericity assumption holds
- H_A = the sphericity assumption is violated

We use the `rstatix` package to conduct the test.

- The dependent variable is `rating`.
- The within-group factor is `skin.treatment`.
- The `Error()` term is critical in differentiating between a between subjects and within subjects model. It tells R that there is one observation per `subject` for each level of `skin.treatment`.

```
# install.packages("rstatix") # If you have not already installed it
library(rstatix)
anova_test(rating ~ skin.treatment + Error(subject/skin.treatment), data=df)
```

```
Attaching package: ‘rstatix’


The following object is masked from ‘package:stats’:

    filter





ANOVA Table (type III tests)

$ANOVA
          Effect DFn DFd     F     p p<.05  ges
1 skin.treatment   3  12 5.118 0.017     * 0.43

$`Mauchly's Test for Sphericity`
          Effect     W     p p<.05
1 skin.treatment 0.062 0.207

$`Sphericity Corrections`
          Effect   GGe     DF[GG] p[GG] p[GG]<.05   HFe     DF[HF] p[HF]
1 skin.treatment 0.541 1.62, 6.49 0.051           0.858 2.57, 10.3 0.023
  p[HF]<.05
1         *
```

The p-value we care about in the output is under “Macuhly’s test for sphericity,” for the variable `skin.treatment`. Because the p-value is 0.207, we fail to reject the sphericity assumption at a 5% significance level and use the univariate approach. to conduct the repeated measures ANOVA.

Repeated measures ANOVA - univariate

```
aov1 <- aov(rating ~ skin.treatment + Error(subject/skin.treatment), data=df)
summary(aov1)
```

```
Error: subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4   11.8    2.95

Error: subject:skin.treatment
               Df Sum Sq Mean Sq F value Pr(>F)
skin.treatment  3  21.75   7.250   5.118 0.0165 *
Residuals      12  17.00   1.417
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

You can find the *p*-value at the end of the row of output marked for `skin.treatment`; it is 0.0165. This is less than 0.05, so we conclude that there is significant evidence of a treatment effect.

Repeated measures ANOVA - multivariate

If instead the first test had rejected the sphericity assumption, we would have used a multivariate approach for the repeated measures ANOVA. We show here how to do such a test, even though it does not apply to this situation. We must first reorganize the data into a matrix where each row represents a single subject, and columns represent levels of the treatment factor. This is possible using the `tidyr` package.

```r
# install.packages("tidyr") # If you have not already installed it
library(tidyr)
multi.data <- spread(df, skin.treatment, rating)
multi.data <- as.matrix(multi.data[,-c(1)])
multi.data
```

```
     W X  Y Z
[1,] 7  5 8 4
[2,] 8 10 7 5
[3,] 7  6 5 4
[4,] 7  7 4 5
[5,] 8  8 6 6
```

We then create a multivariate model and also set up a variable that defines the design of the study.

```r
# In this model there are no between-subjects factors, so we write ~ 1:
multi.ml <- lm(multi.data ~ 1)
# The design of the study is a single factor with four levels:
rfactor <- factor(c("f1", "f2", "f3", "f4"))
```

Conduct the repeated measures ANOVA using a multivariate approach. This requires creating a new model using the `Anova()` function that calculates ANOVA tables. The `car` package provides the `Anova()` function. The parameters have the following meanings.

- `idata` includes information about the number of levels, in this case four.
- `idesign` states that `rfactor` describes a repeated-measures variable.
- `type` tells `Anova()` to calculate the “Type-III” sums of squares when forming the ANOVA table.
- `multivariate` suppresses output about multivariate statistical tests, which are relevant only when the experimental design includes multiple *dependent* variables.

```r
# install.packages("car") # If you have not already installed it
library(car)
multi.ml <- Anova(multi.ml, idata=data.frame(rfactor), idesign = ~rfactor, type="III")
summary(multi.ml, multivariate=FALSE)
```

```
Loading required package: carData






Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

            Sum Sq num Df Error SS den Df  F value    Pr(>F)
(Intercept) 806.45      1     11.8      4 273.3729 7.837e-05 ***
rfactor      21.75      3     17.0     12   5.1176    0.0165 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Mauchly Tests for Sphericity

        Test statistic p-value
rfactor       0.062101 0.20708


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

        GG eps Pr(>F[GG])
rfactor 0.5412    0.05068 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

          HF eps Pr(>F[HF])
rfactor 0.858156 0.02319302
```

Although this test was run just as an example, and does not actually apply in this dataset, the output shows a p-value of 0.0165, at the end of the first `rfactor` row. That p-value could be compared to a chosen α.

(We also see that Mauchly’s test was performed, which is not significant, and is the reason this data actually demands a univariate approach.)

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.

How to perform a planned comparison test

Description

Suppose that ANOVA reveals a significant difference between treatment levels, and you wish to explore further through post hoc analysis by comparing two specific treatment levels. How can we perform perform planned comparisons, also called a contrast test?

Solution in R using gmodels

Usually, you have data you wish to compare, but we will use example data here. We load the "oats" dataset from R's `MASS` package, about the yield of oats from a split-plot field trial using three varieties (V) and four levels of manurial treatment (N). The experiment was laid out in 6 blocks (B) of 3 main plots, each split into 4 sub-plots. The varieties were applied to the main plots and the manurial treatments to the sub-plots.

```r
# install.package('MASS')  # if you have not already done so, and want this data
library(MASS)
df <- oats
```

Before we perform the contrast test, let's verify that the yield of oats `Y` depends on the nitrogen manurial treatment given to it `N`.

```r
aov1 <- aov(Y ~ N, data = df)
summary(aov1)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
N            3  20020    6673    14.2 2.78e-07 ***
Residuals   68  31965     470                     
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p-value in the `Pr(>F)` column is below $\alpha = 0.05$. So at the 5% significance level, the yield differs according to the nitrogen manurial treatment. We assume that the model assumptions are met but do not verify them here.

We now want to perform a planned comparison test (or contrast test) on the data to see whether there is a difference between the $N < 0.5$ levels and the $N > 0.5$ levels. We will use the `fit.contrast` function in the `gmodels` package. Since the order of the levels is 0, 0.2, 0.4 and 0.6, the contrast coefficients will be -0.5, -0.5, 0.5, 0.5, respectively.

```r
# install.package('gmodels')  # if you have not already done so
library(gmodels)
fit.contrast(aov1, "N", coeff=c(-1/2,-1/2,1/2,1/2))
```

```
                          Estimate Std. Error  t value     Pr(>|t|)
N c=( -0.5 -0.5 0.5 0.5 ) 29.66667   5.110338 5.805225 1.855598e-07
attr(,"class")
[1] "fit_contrast"
```

The p-value in the `Pr(>|t|)` column is below $\alpha = 0.05$. This tells us that there is a significant difference between the average yields of the $N < 0.5$ and $N > 0.5$ levels.

Content last modified on 24 July 2023.

See a problem? Tell us or edit the source.