

Cheat Sheet: ANOVA

Measurement and Evaluation of HCC Systems

Scenario

Use ANOVA if you want to test the difference in continuous outcome variable `varY` between multiple levels (A, B, C, ...) of a nominal variable `varX`. In this scenario `varX` is usually an experimental manipulation with more than two conditions (but it can be any nominal variable, such as race, occupation, etc). Each participant is randomly assigned to one condition.

Power analysis

- To do a power analysis for the omnibus test, use Test family “*F* tests”, “ANOVA: Fixed effects, omnibus, one-way”.
- A power analysis has four variables: Effect size, α (usually .05), power (usually .85), and *N*. If you know three of these, G*Power will calculate the fourth. Select the correct type of power analysis, based on the information you have, and what you want to find out.
- “Number of groups” is the number of levels of your nominal variable *X*.
- By clicking on “Determine”, you can compute the effect size *f* from the mean and standard deviation in each group, or from the partial η^2 .
- Click on “Calculate” to calculate the missing parameter.
- To conduct power analyses for contrasts or post hoc tests, refer to power analysis of the independent *t* test. Use a corrected α if needed.

Plotting a line plot and a box plot

- Use the `ggplot2` package to plot a bar chart with error bars.
`ggplot(data, aes(varX, varY)) + stat_summary(fun.y=mean, geom="line") + stat_summary(fun.data=mean_cl_normal, geom="errorbar", width=0.2)`
- Plot a boxplot:
`ggplot(data, aes(varX, varY)) + geom_boxplot()`

Pre-testing assumptions

- In an ANOVA, *Y* should be independent, continuous, and unbounded. The error variance should be normally distributed, which is true if *Y* is normally distributed for each level of *X*. The ANOVA test is fairly robust against violations of normality. For a regular ANOVA,

variances are assumed to be equal across groups (homoscedasticity). Welch's ANOVA does not make this assumption.

- If your N is small:
 - Test for significant skewness, kurtosis, and Shapiro-Wilk test within each group using `stat.desc` in the `pastecs` package and the `by` function.
`by(data$varY, data$varX, stat.desc, basic=F, norm=T)`
 - Multiply `skew.2SE` and `kurt.2SE` by 2 to get the Z-scores of skewness and kurtosis. Compare these values to typical cut-off values ($Z > \pm 1.96$: $p < .05$, $Z > \pm 2.58$: $p < .01$, $Z > \pm 3.29$: $p < .001$). The significance of the Shapiro-Wilk test is listed under `normtest.p`.
 - Test the assumption of homoscedasticity, conduct Levene's test in the `car` package.
`leveneTest(data$varY, data$varX, center=median)`
 - A significant Levene's test means that the data is heteroscedastic. In this case, Welch's ANOVA should be used instead of a regular ANOVA.
- If your N is large:
 - Subset the data into multiple groups based on `varX`:
`Adata <- subset(data, data$varX == "A")`
`Bdata <- subset(data, data$varX == "B")`
etc...
 - Draw the histogram for `varY` in `Adata`, overlaid with a normal curve (using `ggplot2`), and visually inspect whether it follows the normal distribution:
`ggplot(Adata, aes(varY)) + geom_histogram(aes(y=..density..),
binwidth=1, color="black", fill="white") + stat_function(fun = dnorm,
args = list(mean = mean(Adata$varY), sd = sd(Adata$varY)))`
 - Change the `binwidth` setting based on what is suitable for your data.
 - Draw normal a Q-Q plot, and visually inspect whether the data follows the diagonal line:
`ggplot(Adata, aes(sample = varY)) + geom_qq() + geom_qq_line()`
 - Do the same for `Bdata`, etc.
 - Visually inspect the boxplot to assess the level of heteroscedasticity. You can also use the variance ratio test (Hartley's F_{\max}). This is the ratio of the variances between the group with the biggest variance and the group with the smallest variance.
`max(by(data$varY, data$varX, var)) / min(by(data$varY, data$varX, var))`
 - Compare this value against Figure 5.8 in the book.
- If your data has positive skew, and your data only has positive values, you can possibly fix this by transforming your Y variable:
 - Log transform:
`data$varYlog <- log(data$varY + 1)`
 - Or, square root transform:
`data$varYsqrt <- sqrt(data$varY)`

- In other cases of violations of assumptions (e.g. outliers), you can conduct a robust test (see below).

(optional) Preparing contrasts

- If you have specific hypotheses about where the differences between conditions exist, you can run tests as planned contrasts.
- Dummy contrasts:
 - o By default, R uses dummy contrasts, as long as the X is coded as a factor. If it is coded as numerical values (e.g. 1, 2, 3...), you can recode it as factors using the `revalue` function in the `plyr` package:


```
data$varX <- revalue(as.factor(data$varX), c("1"="A", "2"="B", "3"="C"))
```
 - o A, B, C, etc. are descriptive names of the different categories of the variable. You can revalue more categories if needed.
 - o R automatically selects a baseline category against which all other categories are compared. If it chooses this category incorrectly, you can `relevel` the variable, e.g., to make A the baseline category:


```
data$varX <- relevel(data$varX, ref="A")
```
- Other standard contrasts:
 - o The Helmert contrast compares the 2nd level against the 1st level (contrast 1), the 3rd level against the first two levels (contrast 2), etc.:


```
contrasts(data$varX) <- contr.helmert(k)
```

 where k is the number of levels of X .
 - o The Polynomial contrast tests the linear, quadratic, etc. effect of X in case X is an ordered categorical variable:


```
contrasts(data$varX) <- contr.poly(k)
```

 where k is the number of levels of X .
- Create your own contrasts:
 - o Split any chunk of variance (multiple conditions) into two chunks at most. If a condition has been singled out, you can't reuse it. Only split, don't merge. If you have a control group, your first contrast usually compares everything else against the control group (or groups).
 - o For each split, you test chunk P versus chunk Q. Conditions that belong to neither chunk get a weight of 0. Conditions in chunk P get a weight of $k_q/(k_p+k_q)$. Conditions in chunk Q get a weight of $-k_p/(k_p+k_q)$. k_p and k_q are the number of conditions in chunk P and Q.
 - o Example for two conditions (B and C) and a control group (A): First test the control against the two experimental conditions:


```
AvBC <- c(-2/3, 1/3, 1/3)
```

 Then test experimental condition B against C:

```
BvC <- c(0, -1/2, 1/2)
```

Then load the contrasts into `varX`:

```
contrasts(data$varX) <- cbind(AvBC, BvC)
```

Make sure that the levels of `varX` are ordered A, B, C (otherwise use `relevel` to fix the order).

- Note that the dummy contrast is a non-orthogonal contrast. This means that you should apply a correction to the p -values of these contrasts. See “Applying corrections” below.

Running the test

- Run the ANOVA as follows:

```
model1 <- aov(varY ~ varX, data = data)
```
- Get the model summary:

```
summary(model1)
```
- The F -statistic and its p -value tells us whether there is a significant difference between the different groups.
- (optional) if the variances are not equal across groups, use Welch’s ANOVA:

```
oneway.test(varY~varX,data=temp)
```
- You can get the effect size omega-squared using the `omega_aov` function I created:

```
omega_aov(model1)
```

(optional) Robust versions

- You can use functions in the `WRS2` package to run trimmed and/or bootstrapped ANOVAs.
- Robust ANOVA using 10% trimmed means (change the percentage if desired):

```
t1way(varY ~ varX, data = data, tr = 0.1)
```
- Robust ANOVA using bootstrapped median with 2000 samples:

```
med1way(varY ~ varX, data = data, iter = 2000)
```
- Robust ANOVA using 10% trimmed means and 2000 bootstrap samples:

```
t1waybt(varY ~ varX, data = data, tr = 0.1, nboot = 2000)
```

(optional) Evaluate contrasts

- Evaluate your planned contrasts using the `summary.lm` function.

```
summary.lm(model1)
```
- In terms of overall model fit, this output will give you the R-squared, which is the proportion of the variance of `varY` explained by the model. In ANOVA this value is called eta-squared.
- Each coefficient represents the effect of one contrast.
 - o For dummy contrasts, the coefficient represents the difference in Y between this category and the baseline category.
 - o For Helmer contrasts, the coefficient represents the difference in Y between the listed condition and all preceding conditions.

- For polynomial contrasts, the coefficient represents the linear, quadratic, or higher-level polynomial effect of the X conditions on Y .
- For custom contrasts, the coefficient represents the difference in Y between the two groups of conditions included in the contrast.
- Each coefficient has a t test and a p -value to test if the effect is significant. Divide the p -value by 2 if you were conducting a one-sided test (this is very likely for planned contrasts!). You can get the effect size r using the formula $r = \sqrt{t^2 / (t^2 + df)}$.
- You can get confidence intervals for the coefficients using the `confint` function:
`confint(model1)`
- (optional) Apply corrections for k non-orthogonal contrasts.
 - Bonferroni: use $p_{crit} = \alpha/k$.
 - Holm: order your p -values by size, smallest first. The p_{crit} for the first one is α/k . If $p < p_{crit}$, move to the next (otherwise stop). The p_{crit} for the next one is $\alpha/(k-1)$. If $p < p_{crit}$, move to the next (otherwise stop). Continue until $p > p_{crit}$. That p -value and all larger ones are non-significant.
 - Benjamini-Hochberg: order your p -values by size, largest first. The p_{crit} for the first one is $(k/k) * \alpha$. If $p > p_{crit}$, move to the next (otherwise stop). The p_{crit} for the next one is $((k-1)/k) * \alpha$. If $p < p_{crit}$, move to the next (otherwise stop). Continue until $p < p_{crit}$. That p -value and all larger ones are significant.

(optional) Post-hoc tests

- If you do not have specific hypotheses about where specific differences between conditions exist, you can test all of them using post-hoc tests.
- Bonferroni-corrected tests (use "holm" for Holm, and "BH" for Benjamini-Hochberg):
`pairwise.t.test(data$varY, data$varX, p.adjust.method = "Bonferroni")`
- Tukey post-hoc tests and confidence intervals using the package `multcomp` (can also conduct "Dunnnett" post-hoc test):
`post <- glht(model1, linfct=mcp(varX = "Tukey"))`
`summary(post)`
`confint(post)`

(optional) Robust post-hoc tests

- Post-hoc tests using 10% trimmed means (confidence intervals are corrected; p -values are not):
`lincon(varY ~ varX, data=data, tr=0.1)`
- Post-hoc tests using 10% trimmed bootstrapped means, using 2000 bootstrap samples:
`mcpb20(varY ~ varX, data=data, tr=0.1, nboot=2000, crit=.05)`

(optional) Effect sizes of specific differences

- Get means, *sds*, and *ns* from `stat.desc` in the `pastecs` package.
`desc <- by(data$varY, data$varX, stat.desc)`
- Plug the values into `mes`, e.g. to test the differences between groups A and B.
`mes(desc$`A`["mean"], desc$`B`["mean"], desc$`A`["std.dev"],
desc$`B`["std.dev"], desc$`A`["nbr.val"], desc$`B`["nbr.val"])`
- It is common to report *d* instead of *r*.

Reporting

- Use the following format to report on an ANOVA (replace the full names (not just the variable names) of A, B and varY, and replace the xx'es with the actual numbers).
- For the omnibus test: "There was a significant effect of [varX] on [varY], $F(x, x) = x.xx$, $p = .xxx$, $\omega = .xx$.
- Linear contrast: There was a significant linear trend, $t(xx) = x.xxx$, $p = .xxx$, indicating that [varY] increases proportionally with [varX].
- Planned contrasts: Planned contrasts revealed that [group Q of contrast 1] [increased / decreased] [varY], compared to [group P of contrast 1], $b = x.xx$, $t(xx) = x.xx$, $p_{\text{one-tailed}} = .xxx$."
- Post-hoc tests: "Bonferroni tests revealed significant differences between [A] and [B], $p = .xxx$, $d = x.xx$, between [A] and [C], $p = .xxx$, $d = x.xx$, and between [B] and [C], $p = .xxx$, $d = x.xx$."