Cheat Sheet: Correlation

Measurement and Evaluation of HCC Systems

Scenario

Use correlation if you want to test the linear association between two continuous variables variables `var1` and `var2` in your dataset `data`.

Power analysis for correlation

- Use Test family “Exact”, “Bivariate normal model”.
- 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.
- “Correlation ρ H1” is the effect size. It is the same as *r*. “Correlation ρ H0” is the null hypothesis, which is typically zero.
- Make sure you select a one- or two-tailed test based on your hypothesis. If you hypothesize a particular direction (positive or negative correlation), use a one-tailed test. If you hypothesize any correlation, use a two-tailed test.
- Click on “Calculate” to calculate the missing parameter.

Plotting a scatterplot with linear trend line

- Use the `ggplot2` package to plot a scatterplot with a linear trend line.
```
ggplot(data, aes(var1, var2)) + geom_point() + geom_smooth(method=”lm”,
color=”red”, se=F)
```
- (optional) To add a mean line, add:
```
+ geom_line(aes(y = mean(data$var2)), color=“blue”
```
- Visually inspect if the relationship is indeed a linear one.

Pre-testing assumptions

- Correlation is valid for independent interval data; the significance test requires that both variables are normally distributed.
- If your *N* is small:
  - Test for significant skewness, kurtosis, and Shapiro-Wilk test using `stat.desc` in the `pastecs` package.
```
stat.desc(data$var1, desc=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`. Repeat the procedure for `var2`.

- If your *N* is large:
  - Draw the histograms for the two variables, overlaid with normal curves (using `ggplot2`), and visually inspect whether they follow the normal distribution:
    ```
    ggplot(data, aes(var1)) + geom_histogram(aes(y=..density..), binwidth=1, color="black", fill="white") + stat_function(fun = dnorm, args = list(mean = mean(data$var1), sd = sd(data$var1)))
    ```
  - Change the `binwidth` setting based on what is suitable for your data. Repeat the inspection for `var2`.
  - Draw normal Q-Q plots, and visually inspect whether the data follows the diagonal line:
    ```
    qplot(sample = data$var1, stat="qq")
    ```
  - Repeat this inspection for `var2`.
- If your data has positive skew, and your data only has positive values, you can possibly fix this by transforming your data, using a transform:
  - Log transform:
    ```
    data$var1log <- log(data$var1 + 1)
    ```
  - Or, square root transform:
    ```
    data$var1sqrt <- sqrt(data$var1)
    ```
  - Repeat the normality tests for the transformed variables.
- In other cases of violations of assumptions, you can conduct a robust test (see below).

Running the test

- If you want to run a single correlation, with *p*-values and confidence intervals, use `cor.test`:
  ```
  cor.test(data$var1, data$var2)
  ```
- This test gives you the correlation, the *t* statistic, *p*-value, and a 95% confidence interval. Divide the *p*-value by 2 if you were conducting a one-sided test (i.e. if you had a directional hypothesis).
- (optional) If you want to run several correlations at once, with p-values, use `rcorr` in the `Hmisc` package:
  ```
  rcorr(as.matrix(data[ , c("var1", "var2", "var3")]))
  ```
- One variable that is not reported, is R^2. This variable can be interpreted as the proportion of shared variation between `var1` and `var2`. You can calculate it by squaring the correlation coefficient ($r * r$).

(optional) Robust correlation

- For ordinal or non-normal data, use Kendall's Tau. The interpretation is the same as for a regular correlation:
  `cor.test(data$var1, data$var2, method="kendall")`
- You can also use a bootstrapped correlation. This works for both the regular (Pearson) correlation and Kendall's Tau.
  - First create a function for running the bootstrap sample:
    `bootFun <- function(sample,i) cor(sample$var1[i],sample$var2[i], method="kendall")`
  - Then run the bootstrap sample over the function 2000 times:
    `bootResult <- boot(data, bootFun, 2000)`
  - Get the output; the `original` column shows the correlation in the original sample, the `bias` column shows the difference between this and the correlation in the bootstrap sample, and the `std. error` column shows the bootstrapped standard error:
    `bootResult`
  - Get the confidence interval; the `BCa` version is the most robust variant:
    `boot.ci(bootResult)`

(optional) Controlling for other variables (partial correlation)

- With "partial correlation" you can get the correlation between `var1` and `var2`, controlling for the variability that is explained by `var3` (and more variables, if required). You can run partial correlation using the `pcor` function in the `ggm` package:
  `pc <- pcor(c("var1", "var2", "var3", var(data))`
- Get the output:
  `pc`
- Do a *t* test on the partial correlation using `pcor.test`; `q` is the number of variables you are controlling for, `N` is the sample size:
  `pcor.test(pc,q,N)`

Reporting

- Use one of the following phrasings to report on a correlation (replace the full names (not just the variable names) of var1 and var2, and replace the xx'es with the actual numbers:
  - "[var1] was significantly correlated with [var2], *r* = .xx, *p* = .xxx"
  - "There was a significant relationship between [var1] and [var2], *r* = .xx, *p* = .xxx"
  - "[var1] was significantly related to [var2], *r* = .xx, *p* = .xxx"