Week 5 Cheat Sheet

Statistics and Data Analysis with R

Course Link: https://www.coursera.org/learn/statistics-and-data-analysis-with-r/

Charlie Nuttelman

Here, I provide the functions in R required to perform various calculations in Week 5 of the course. The headings represent the screencasts in which you will find those concepts and examples.

Testing the Assumption of Normality

As I included in the Week 4 Cheat Sheet, this is not a screencast in itself, but you need to make sure that you are verifying the assumption of normality when you use many of the tests in Week 5 of the course. To test for the assumption of normality, we can do the following (note that the **ad.test** function requires a sample vector of at least size 8):

1. Load the **nortest** library: **library(nortest**) (if this library is not installed, install it using **install.packages("nortest")**.
2. Calculate the P-value of the Anderson-Darling (AD) statistic of the data in vector **x** using **ad.test(x)**.
3. To store the P-value of the AD statistic as a variable: `P <- ad.test(x)$p.value`
4. If the P-value of the AD statistic is *greater* than 0.05, then we can assume that the data in vector **x** come from a normally distributed population; if the P-value of the AD statistic is *not greater* than 0.05, then we can assume that the data in vector **x** do not come from a normally distributed population.

Comparison of Means, Variance Known

In R, we can compare the means of two normally distributed populations. This can be a lower-tailed, two-tailed, or upper-tailed test. If you already have the experimental data, it's more common to use a one-tailed (lower- or upper-tailed) test. For example, we can perform the following upper-tailed hypothesis test:

$$H_0: \mu_A = \mu_B$$

$$H_0: \mu_A > \mu_B$$

When variance is known, we base this off a "z test". In R, we can perform a z test using the following options:

- **z.test** (BSDA)
    - Make sure that if you've loaded the **TeachingDemos** library that you detach (unload) that library prior to using the **z.test** function of the **BSDA** library. You can do this by

going over to the **Packages** tab in the lower right pane and deselecting the check mark next to the **TeachingDemos** library. Alternatively, you can type:

```
detach("package:TeachingDemos", unload = TRUE)
```

- To perform a z test to compare the data in vectors **A** and **B** when the variances (**sigma.x** and **sigma.y**, respectively) are known and equal, we can use (showing here an upper-tailed test but you can also perform a lower-tailed test by using **alternative="less"**):

```
z.test(A, B, alternative="greater", sigma.x=15,
sigma.y=15)
```

- To perform a z test to compare the data in vectors **A** and **B** when the variances (**sigma.x** and **sigma.y**, respectively) are known but unequal, we can use (showing here an upper-tailed test but you can also perform a lower-tailed test by using **alternative="less"**):

```
z.test(A, B, alternative="greater", sigma.x=10,
sigma.y=15)
```

- To store the P-value of the **z.test** function in a single variable, we can use the following:

```
P <- z.test(A, B, alternative="greater", sigma.x=10,
sigma.y=15)$p.value
```

- User-defined function:
  - We can also use a user-defined function, an example of which is shown here:

```
z.comp.test <- function(x1,x2,sigma1,sigma2){
  xbar1 <- mean(x1)
  xbar2 <- mean(x2)
  n1 <- length(x1)
  n2 <- length(x2)
  z0 <- (xbar1-xbar2)/sqrt(sigma1^2/n1+sigma2^2/n2)
  print(z0)
  # P-value
  if (z0>0){
    P <- pnorm(z0, lower.tail=FALSE)
  } else {
    P <- pnorm(z0, lower.tail=TRUE)
  }
  print(P)
}
```

  - This function will work for both the variance equal case and variance unequal case. To use the function, we can simply type (just one example shown here):

```
z.comp.test(A,B,15,15)
```

- Note that, in general, it's more common to have the "variance unknown" case in real statistical analysis, unless the sample size is big.

Comparison of Variances

In order to perform a comparison of means of two populations when the variance is unknown (see next section), we need to know if the assumption of equal variance or unequal variance holds. To determine whether or not variances can be assumed to be equal, we can use the following method in R to perform a comparison of variances:

- **var.test** (stats)
    - For a lower-tailed hypothesis test (H_0: $\sigma_A^2 = \sigma_B^2$; H_1: $\sigma_A^2 < \sigma_B^2$) on the comparison of variances, we can use the **var.test** function in R, where **A** and **B** are two vectors containing our data:

```
var.test(A, B, alternative="less")
```

    - For an upper-tailed hypothesis test (H_0: $\sigma_A^2 = \sigma_B^2$; H_1: $\sigma_A^2 > \sigma_B^2$) on the comparison of variances, we just need to use **alternative="greater"**:

```
var.test(A, B, alternative="greater")
```

    - We can use a different confidence level (the default is 95%), for example 90% here:

```
var.test(A, B, alternative="greater",conf.level=0.9)
```

    - To store the P-value of the **var.test** function in a single variable, we can use:

```
P <- var.test(A, B,
alternative="greater",conf.level=0.9)$p.value
```

Comparison of Means, Variance Unknown

In R, we can compare the means of two normally distributed populations. This can be a lower-tailed, two-tailed, or upper-tailed test. If you already have the experimental data, it's more common to use a one-tailed (lower- or upper-tailed) test. For example, we can perform the following upper-tailed hypothesis test:

$$H_0: \mu_A = \mu_B$$

$$H_0: \mu_A > \mu_B$$

When variance is unknown (more common than when the variance is known), we base this off a "t test". In R, we can perform a t test using the following options:

- **t.test** (stats)
    - To perform a t test to compare the data in vectors **A** and **B** when the variances are unknown and equal (to determine if variances can be assumed to be equal or unequal, see above in "Comparison of Variances" section), we can use (showing here an upper-tailed test but you can also perform a lower-tailed test by using **alternative="less"**):

```
t.test(A, B, alternative="greater", var.equal=TRUE)
```

    - To perform a t test to compare the data in vectors **A** and **B** when the variances are known and unequal (to determine if variances can be assumed to be equal or unequal, see above in "Comparison of Variances" section), we can use (showing here an upper-tailed test but you can also perform a lower-tailed test by using **alternative="less"**):

```
t.test(A, B, alternative="greater", var.equal=FALSE)
```

    - To store the P-value of the **t.test** function as a single variable, we can use:

```
P <- t.test(A, B, alternative="greater",
var.equal=FALSE)$p.value
```

- User-defined function
    - We can also use a user-defined function, similar to that shown above in "Comparison of Means, Variance Known". I did not show how to do this in any of the course videos, but you could easily adapt the code.

Paired T Tests

A paired t-test is when it makes more sense to pair increases or decreases in a parameter rather than to keep data independent. For example, if you were studying the increase in effectiveness of a skin cream to alleviate rashes, every person responds differently so you would look at the left arm (control) vs. right arm (treatment). It makes more sense to look at the average of the differences for individuals rather than differences in the averages. Or we have two temperature measuring devices and we record the daily high temperature on different days for each device – it's the difference in measurements that we are interested in.

The hypothesis test that we set up is a lower-tailed, upper-tailed, or two-tailed test. Here is an example of an upper-tailed test (μ_D is the mean of the differences):

$$H_0: \mu_D = 0$$

$$H_1: \mu_D > 0$$

To perform a paired t test in R, we can use the following methods:

- Mathematical formulas
    - Given two vectors (**x** and **y**) that contain our paired data, we can use the following mathematical formulas in R to calculate the P-value (P):

```
d <- x-y
n <- length(d)
dbar <- mean(d)
s <- sd(d)
t0 <- dbar/s*sqrt(n)
P <- pt(t0, n-1, lower.tail=FALSE)
```

    - Note that this code performs the upper-tailed test ($H_1: \mu_D > 0$), as shown above.
    - To perform a lower-tailed hypothesis test ($H_1: \mu_D < 0$), we can simply use `lower.tail=TRUE` in the last line.
- **t.test** (stats)
    - We can use the **t.test** function with `paired=TRUE`:

```
t.test(x,y,alternative="greater",paired=TRUE)
```

    - Note that the **var.equal** optional argument is overridden in the **t.test** function when `paired=TRUE`. Note also that an error will occur if the lengths of **x** and **y** are not the same.
    - To store the P-value of the **t.test** function for a paired test as a single variable, we can use:

```
P <- t.test(x,y,alternative="greater",
paired=TRUE)$p.value
```

Comparison of Binomial Proportions

For comparing binomial proportions from two different populations, there is no direct binomial approach as there was with hypothesis tests on a binomial proportion so we must base our approach on the normal approximation to the binomial distribution. We aim to determine whether the proportion of population A (p_A) having some characteristic is the same as or different from that of population B (p_B).

The sample size of population A is denoted as n_A and that of population B is denoted as n_B. The number of items in population A that have the characteristic of interest is x_A and that of population B is x_B.

Thus, we can calculate the estimators for p_A and p_B as:

$$\hat{p}_A = x_A/n_A$$

$$\hat{p}_B = x_B/n_B$$

The hypothesis test we set up can be a lower-tailed, upper-tailed, or two-tailed test. Shown here is an example of an upper-tailed test:

$$H_0: p_A = p_B$$

$$H_1: p_A > p_B$$

In R, we can use the following methods to perform a comparison test on binomial proportions from two populations:

- Mathematical formulas
    - Given the data for two samples that we wish to compare, we can perform the hypothesis test on the comparison of binomial proportions to calculate a P-value (see the screencast on "Comparison of Binomial Proportions" for more of the theory):

```
pAhat <- xA/nA
pBhat <- xB/nB
phat <- (xA+xB)/(nA+nB)
z0 <- (pAhat-pBhat)/sqrt(phat*(1-phat)*(1/nA+1/nB))
P <- pnorm(z0, lower.tail=FALSE)
P
```

    - Note that this code performs the upper-tailed test ($H_1: p_A > p_B$), as shown above.
    - To perform a lower-tailed hypothesis test ($H_1: p_A > p_B$), we can simply use `lower.tail=TRUE` in the last line.
- **prop.test** (stats)
    - For a lower-tailed test ($H_1: p_A < p_B$), we can perform the test using the following line of code in R:

```
prop.test(x=c(xA, xB), n=c(nA,nB), alternative="less",
correct=FALSE)
```

    - For a lower-tailed test ($H_1: p_A < p_B$), we can perform the test using the following line of code in R:

```
prop.test(x=c(xA, xB), n=c(nA,nB),
alternative="greater", correct=FALSE)
```

    - In both cases, note the use of "`correct=FALSE`".

- As always, if **P** (the P-value) for either of the above is less than our significance level (α), then we can accept the alternate hypothesis.
- To store the P-value of the prop.test function as a single variable (shown for the upper-tailed test), we can use:

```
P <- prop.test(x=c(xA, xB), n=c(nA,nB),
alternative="greater", correct=FALSE)$p.value
```