Learning Outcomes

1. Use R Studio to conduct one-sample hypothesis tests using the Z distribution
2. Use R Studio to conduct one-sample hypothesis tests using the t distribution
3. Use R Studio to conduct one-sample hypothesis tests for proportions
4. Use R Studio to conduct chi square tests.

**One-sample Z-tests**

Base R does not contain an inbuilt function to perform one-sample Z-tests. However, it is not overly difficult to do using code. As you recall from lectures, for a one-sample Z test we need a null hypothesis, a significance level, and a test statistic, which we compare to the Z-distribution.

# we can extract some of these quantities from a vector like so:

Sample <-rnorm(100,0,1) #simulate some data

n <- length(Sample) #size of the sample

x\_bar <- mean(Sample) #mean of the sample

sd <- sd(Sample) #this is the sample SD, in this case we know the population SD is 1 since we simulated our data

# lets define mu (the mean of the population under the null hypothesis

mu <- 0

#calculate the test statistic for a Z-distribution

test\_stat <- (x\_bar-mu)/(sd/sqrt(n))

#get the p-value for this test statistic

p\_val <- 2\*pnorm(abs(test\_stat), lower.tail=FALSE) # for two tailed

p\_val <- pnorm(test\_stat , lower.tail=TRUE) # for left-tailed

p\_val <- pnorm(test\_stat , lower.tail=FALSE) # for right tailed

**One-Sample Z-test for proportions**

The only change needed to use the code above for a proportion is in your calculation of standard error.

Propdat <- rbinom(100, 1, 0.3) # simulate some data

n <- length(Propdat) #size of the sample

prop <- sum( ( Propdat == 1 ) / n ) #proportion of ones in the sample

prop <- mean(Propdat) #could also do it this way since as results are either 0 or 1, the mean will be the proportion of values that are 1

#calculate sd for a z-test of proportions

sd = sqrt((prop)\*(1-prop)/n)

**One-Sample t-tests**

Similar code can be used for a one-sample t-test, the difference being that the df must be specified, and the t-distribution is used rather than the Z-distribution.

df <- n-1 #define df

p\_val <- 2\*pt(abs(test\_stat), df, lower.tail=FALSE) #two-tailed

p\_val <- pt(test\_stat , df, lower.tail=TRUE) #left-tailed

p\_val <- pt(test\_stat , df, lower.tail=FALSE) # right-tailed

We could also use the t.test function in the package {stats}

t.test(Sample) #two-tailed

t.test(Sample, alternative = "less") #left-tailed

t.test(Sample, alternative = "greater") #right-tailed

***Can you think about how you might write code to perform two-sample tests?***

**Chi Square tests**

The following code will make a new dataset “dat” which contains the standard iris flower data and a new variable size, which records if the sepal length of each observation is above or below the median.

dat <- iris

dat$size <- ifelse(dat$Sepal.Length < median(dat$Sepal.Length), "small", "big")

Next, we can create a contingency table from size and species

Tbl <- table(dat$size, dat$Species)

Tbl #look at the table

Finally, we can conduct a chi square test.

chisq.test(Tbl)

**Making a function**

***The below code is just an example for your own interest. You will not be asked to work with the below code in your Lab Task***

We can make a function where we enter the values needed and get a printout of our results for our desired test.

##code for a one-sample hypothesis test function

OneSampTest <-function(type=NULL, tails=NULL, alpha, mu, n, x\_bar, sd)

{

#calculate the test statistic

se = (sd/sqrt(n))

test\_stat <- (x\_bar-mu)/(sd/sqrt(n))

if (type=="z") {

#get the p-value for this test statistic

if (tails =="two") { p\_val <- 2\*pnorm(abs(test\_stat), lower.tail=FALSE)

} else if (tails=="left") {p\_val <- pnorm(test\_stat , lower.tail=TRUE)

} else if (tails=="right") {p\_val <- pnorm(test\_stat , lower.tail=FALSE)

} else {stop("please choose tails as two, left, or right")}

}

else if (type =="t") {

#define df

df <- n-1

#get the p-value for this test statistic

if (tails =="two") { p\_val <- pt(abs(test\_stat), df, lower.tail=FALSE)

} else if (tails=="left") {p\_val <- pt(test\_stat , df,lower.tail=TRUE)

} else if (tails=="right") {p\_val <- pt(test\_stat , df, lower.tail=FALSE)

} else {stop("please choose tails as two, left, or right")}

}

else {stop("please choose z or t")}

#check if significant

if (p\_val <alpha) {sig <-"significant"

} else {sig <-"not significant"}

ret <- list(type=paste("One Sample", type, "test.", tails, "tailed"),

input=paste("alpha=", alpha, "; mu=", mu, "; n=", n, "; x-bar=", x\_bar, "; sd=", sd),

calculations=paste("se=", se, "; test statistic", test\_stat),

conclusion=paste("At a significance level of", alpha, "the p-value of", p\_val, "is", sig))

#return the list

return( ret )

}

We can use the function like so:

