**# Setting Up Data**

Below we will use the code setwd() to set our working directory so that R can find our file. Then we will input our data set "t4data.csv" and call it GroupData.

|  |
| --- |
| *```{r setup, include=TRUE, tidy=TRUE, tidy.opts=list(width.cutoff=60)}*  *knitr::opts\_chunk$set(echo = TRUE)*  *setwd("/Users/lauranstoner/Downloads")*  *GroupData <- read.csv("t4data.csv", header = TRUE, sep = ",")*  *```* |

**## Sampling Designs and Considerations #1**

Taking a sample from a larger population can enable us to further examine data. The most useful function for sampling is *sample().* For this function, we must input the vector we want to sample from, the size of the sample, and we have the option of sampling with or without replacement. Below, we see an example of a sample of 15 different weight values using replacement.

|  |
| --- |
| *```{r}*  *# Example:*  *sample(GroupData$weight, 15, replace = TRUE)*  *```* |

**## Sampling Variability #2**

Under sampling variability we briefly discuss the types of variables. Broadly, variables are either categorical or quantitative. We can use functions to determine the variable types in a data set, or even change them if need be. Using is.\_\_\_() can help us determine the type of variable. Using as.\_\_\_() can help us change the type of variable. We can either fill the blank with numeric, character, factor, integer, or others depending on what type we want our variable to be.

|  |
| --- |
| *```{r}*  *# Determine variable type in our example data set.*  *is.numeric(GroupData$day)*  *# We are returned with a "true" meaning the variable "day" is numeric.*  *#Change data type of "day" to character.*  *DayC<- as.character(GroupData$day, length = 2)*  *is.character(DayC)*  *# We have now reformed the variable "day" into one of character type with a field length of 2.*  *```* |

**## Summarizing and Exploring Data #2**

**### Numerical Summaries**

|  |
| --- |
| *```{r}*  *# Frequency Distribution: We can use functions such as length() or table() to examine relative frequencies. For example...*  *table(GroupData$day)*  *length(GroupData$day)*  *# With the relative frequency of each value and the length of the variable itself, we can calculate the frequency of each value.*  *```* |

|  |
| --- |
| *```{r}*  *# Measures of centre: This describes the typical value of a distribution.*  *mean(GroupData$weight)*  *median(GroupData$weight)*  *```* |

|  |
| --- |
| *```{r}*  *# Percentile: Percentiles are a value below which a particular percentage of the distribution lies.*  *quantile(GroupData$weight, 0.75)*  *```* |

|  |
| --- |
| *```{r}*  *# Measure of Spread: Values such as range, 5 number summaries, interquartile range, variance, and standard deviations can be used to describe spread.*  *sd(GroupData$weight)*  *var(GroupData$weight)*  *IQR(GroupData$weight)*  *fivenum(GroupData$weight)*  *range(GroupData$weight)*  *```* |

**### Graphical Summaries**

|  |
| --- |
| *```{r}*  *# Barplot: Visual representation of the frequency distribution of a variable.*  *table(GroupData$day)*  *barplot(table(GroupData$day))*  *```* |

|  |
| --- |
| *```{r}*  *# Stacked Barplot: Stacked barplots are a visual representation of a two-way table. Our example below shows the distribution of the types of crop across the various days of measurement.*  *StackedPlot <- table(GroupData$crop, GroupData$day)*  *barplot(StackedPlot)*  *```* |

|  |
| --- |
| *```{r}*  *# Mosaic Plot: Area of rectangles reflect relative frequencies for two or more variables.*  *MosaicExample <- table(GroupData$crop, GroupData$day)*  *mosaicplot(MosaicExample)*  *```* |

|  |
| --- |
| *```{r}*  *# Histogram: Visual representation of frequency distribution for a single quantitative variable.*  *hist(GroupData$weight, xlab = "Weight", main = "Histogram of Weight")*  *```* |

|  |
| --- |
| *```{r}*  *# Evaluating Symmetry: We can describe distributions by their degree of symmetry. In our example below, we can see that the distribution of height for our data is approximately symmetrical. If our data fell more to the right, it would be right-skewed. If the data fell more to the left, it would be left-skewed.*  *# Evaluating Modality: We can also describe the distributions by modality. Modality is the number of prominent peaks in the distribution. In our example below, we only see one peak.*  *hist(GroupData$height, xlab = "Height", main = "Histogram of Height")*  *```* |

|  |
| --- |
| *```{r}*  *# Boxplot: Boxplots are an incredibly informative summary of data. Boxplots incorporate the min, first quartile, mean, third quartile, and maximum of the data.*  *boxplot(GroupData$weight, ylab = "Weight", main = "Boxplot of Weight")*  *```* |

|  |
| --- |
| *```{r}*  *# Stripchart: Stripcharts plot data points of a quantitative variable across categorical groups.*  *plot(GroupData$height~GroupData$day)*  *```* |

|  |
| --- |
| *```{r}*  *# Scatterplot: Scatterplots plot quantitative variables as x,y coordinates.*  *plot(GroupData$height, GroupData$weight, xlab = "Height", ylab = "Weight", main = "Height vs. Weight in Crops")*  *```* |

**## Probability Models: Binomial #3**

Binomial models can be described by their shape, center, and spread. This means we can use curves, mean functions, and SD functions as shown above. We can also use separate functions curated just for binomial models to work with our data.

|  |
| --- |
| *```{r}*  *# Using dbinom to find the probability at a certain value.*  *```* |

|  |
| --- |
| *```{r}*  *# Using qbinom to find the value given a probability.*  *```* |

|  |
| --- |
| *```{r}*  *# Using pbinom to find the cumulative probability given a certain value.*  *```* |

**## Probability Models: Normal #4**

Normal distribution are defined by their mean and SD, so we can use these functions as presented in previous sections. We can also use separately curated function to work with the data.

|  |
| --- |
| *```{r}*  *# Using dnorm to find density at a certain value.*  *```* |

|  |
| --- |
| *```{r}*  *# Using pnorm to find the cumulative probability given a certain value.*  *```* |

|  |
| --- |
| *```{r}*  *# Using qnorm to find the value given a certain probability.*  *```* |

**## Sampling Distributions #5**

**## T Confidence Intervals for the Mean #5.1**

T distributions are used when we are relying on a sample s to estime the population standard deviation.

|  |
| --- |
| *```{r}*  *# Using pt for cumulative probability.*  *```* |

|  |
| --- |
| *```{r}*  *# Using qt to find value of t given a cumulative probability.*  *```* |

|  |
| --- |
| *```{r}*  *# Using t.test to compute a confidence interval.*  *```* |

**## Large Sample Confidence Intervals for Proportion #5.2**

Large sample confidence intervals are computed given 4 conditions: data is from a SRS, count of successes can be approximated by a binomial model, normal approximation of binomial model is reasonable, and np and nq must be at least 15. Due to this being a normal model, we are also able to use pnorm and qnorm functions as well.

|  |
| --- |
| *```{r}*  *# The qqnorm function can be used to determine if a normal approximation is reasonable. We can also use nrow to determine if our sample size is large enough to apply the central limit theorem.*  *qqnorm(GroupData$height)*  *nrow(GroupData)*  *```* |

|  |
| --- |
| *```{r}*  *# Prop.test enables us to compute a confidence interval around the data. We use x (5) as a count of our successes, n (20) as our size, and choose a confidence level (0.90) for our interval. For this example we will input randomly selected numbers to demonstrate the use. We will also apply correct = FALSE to ensure that R does not compute with the continuity correction.*  *prop.test(5, 20, conf.level = 0.90, correct = FALSE)*  *```* |

**## Large Sample Test for Proportion** **#5.3**

The large sample Test for Proportion has 3 conditions that must be fulfilled: the sample data is from a SRS, count of successes can be described by a binomial model, and lastly, the normal approximation of the binomial model must be reasonable. However, in this case, we must check that np0 and nq0 are greater or equal to 10 (where q0 is the compliment of p0). It is important to emphasize that these values are based on the hypothesis values and not the sample estimates.

|  |
| --- |
| *```{r}*  *#In this case, our x value is the count of "successes" in our sample while n will represent the size of the sample. the p value is the value of our null hypothesis value. The alternative value is what we are assigning to our alternative hypothesis. In this case, since we have put "less", we are indicating that the true p value will be less than our null value (0.0425). On other words, we have assigned a left-tailed test to this given function. Alternatively, we would assign greater if we wanted a a right tailed test, and two.sided if we wanted a two tailed test. The "correct" argument indicates whether or not continuity correction will be used in the function. For our purposes, we will not be using continuity correction, and as such, we have indicated it as FALSE.*  *prop.test(x=4, n=200, p=0.0425, alternative="less", correct=FALSE)*  *```* |

**## T Test for the Mean #5.4**

Since we are dealing with another hypothesis test, we will need to define our null and alternative hypothesis. In this case, our null hypothesis value will equal the parameter (u=u0) and our alternative hypothesis will show a difference in u and u0 (Can be left tailed, right tailed, or two tailed). There are 2 conditions that must be fulfilled to use the T test for the mean. The first is that the sample must be from a SRS. Additionally, the sample must come from a population that follows a normal distribution as we will need a t distribution for this test. However, if we have a large enough sample, in this case n=50, we can consider this condition fulfilled. From the t distribution with degrees of freedom equal to n-1, we can calculate t test, that is, (Sample statistic - NULL value)/SE\_mean. This test statistic is appropriate for pairwise difference in data as well as testing against a null value for a given single quantitative variable.

|  |
| --- |
| *```{r}*  *#the x argument will compute what data values to take from. the mu argument will indicate the tru value of the mean, or the difference in 2 means if you are dealing with 2 sample tests. the alternative argument will indicate what the alternative hypothesis states: this can be left-tailed, right-tailed, or two-tailed.*  *t.test(x=diff,mu=0,alternative="less")*  *```* |

**## T Test for Differences in Means #5.5**

The T test for difference in means is used when comparing 2 population means. Again, we will need to define our null and alternative hypothesis. In this case, the null hypothesis will indicate no difference between our two population means. The alternative hypothesis can once again be left-tailed, right-tailed, and two-tailed indicating that there is a difference between population means. There are 2 conditions for T tst for difference in means: Both samples must be SRS, and the samples must be independent from each other and are normally distributed.

*##(I HAVENT GONE OVER STANDARD ERROR YET)##*

|  |
| --- |
| *```{r}*  *#the x represents the first data set you will take from while the y argument represents the second data set. This is necessary as we are comparing the two population means. mu represents the difference in the 2 means and finally, alternative will indicate whether your alternative hypothesis is left-tailed, right-tailed, or two-tailed.*  *t.test(x=injection, y=IUD, mu=0, alternative="two.sided")*  *```* |

**## Large Sample Test for Difference in Proportion #5.6**

**### Use R to run t procedures to conduct inference on two population means**

The independent two sample t-test and confidence interval are parametric methods appropriate for examining the difference in means for two populations. They can also be thought of as a way of examining the relationship between a numeric outcome or Y variable and a categorical explanatory or X variable.

|  |
| --- |
| *```{r t.test}*  *# Compare the Weight separated on days.*  *#The null hypothesis is that the mean difference is 0 and the alt is two-sided*  *#The confidence of 95 percent for the confidence interval*  *# The variances are not equal and these two groups are not paired*  *t.test(GroupData$weight~ GroupData$crop, mu=0, alt="two.sided", conf=0.95, var.eq=F, paired=F)*  *```* |

We can see the output returned by R the test statistic of negative 30.379, the p-value is 2.2 x10-16,

The 95 percent confidence interval for the difference in means running from negative 1.938746 to negative 1.703589; as well the sample group means of 7.113237 to 8.934404

**###User R to run large sample (approximate Z) procedures to conduct inference on two proportions**

**###Large sample procedures for p1-p2**

|  |
| --- |
| *```{r barplot}*  *# Create a table with two rows: crop type “A” and crop type “B”*  *#And two columns: Height >=10 and Height <10*  *#Display the barplot*  *#Total number of A items*  *total\_A <- nrow(GroupData[(GroupData$crop=="A"),])*  *#Total number of B items*  *total\_B <- nrow(GroupData[(GroupData$crop=="B"),])*  *#Total number of A items with height >= 10*  *gl10\_A <- nrow(GroupData[(GroupData$crop=="A" & GroupData$height >=10),])*  *#Total number of B items with height >= 10*  *gl10\_B <- nrow(GroupData[(GroupData$crop=="B" & GroupData$height >=10),])*  *matrixGL10 <- matrix(c(gl10\_A, total\_A-gl10\_A, gl10\_B, total\_B-gl10\_B), ncol=2)*  *rownames(matrixGL10) <- c("A", "B")*  *colnames(matrixGL10) <- c("Height>=10", "Height<10" )*  *matrixGL10*  *barplot(prop.table(matrixGL10, margin = 2), beside=FALSE, ylim=c(0,1), legend = rownames(matrixGL10),*  *main="Height proportions >=10 and <10", ylab="proportion", cex.names = 0.9 )*  *```* |

|  |
| --- |
| *```{r prop.test}*  *prop.test(c(gl10\_A, gl10\_B ), c(total\_A-gl10\_A, total\_B-gl10\_B))*  *```* |

**###R Output for the prop.test() function**

Hypothesis test of H0 : p1=p2 vs HA: p1 ≠p2

A 95% confidence interval for p1-p2 is (,)

The proportion estimates based on these data are 0.381 (Height>=10) and 0.33 (Height <10)

**## Simple Linear Regressions #6**

Simple linear regression is useful for examining or modelling the relationship between two numeric variables.

**# Fit simple linear regression models**

|  |
| --- |
| *```{r cor}*  *# Calculate the Pearson's correlation between Day and Weight*  *cor(GroupData$day, GroupData$weight)*  *```* |

|  |
| --- |
| *```{r lm}*  *# Fit a linear regression and save it in the object: fit*  *fit <- lm( GroupData$weight, GroupData$day)*  *```* |

*Summary* function returns a summary for the residuals, the estimate of the intercept, its standard error as well as the test statistic, and p-value for a hypothesis test that the intercept is zero.

|  |
| --- |
| *```{r fit summary}*  *# Summary of the model fit*  *summary(fit)*  *```* |

|  |
| --- |
| *```{r lm to fit}*  *# Fit a linear regression and save it in the object: fit*  *fit <- lm(GroupData$weight, GroupData$day)*  *```* |

**# Produce regression diagnostics**

|  |
| --- |
| *```{r}*  *# Add fitted line to scatterplot*  *plot(GroupData$day, GroupData$weight)*  *abline(fit, col=2, lwd=3)*  *```* |

|  |
| --- |
| *```{r}*  *# Produce residual plot*  *par(mfrow=c(2,1))*  *plot(GroupData$day, fit$residuals)*  *plot(fit$fitted.values, fit$residuals)*  *```* |

|  |
| --- |
| *```{r}*  *# produce QQPlot of residuals*  *qqnorm(fit$residuals)*  *qqline(fit$residuals)*  *```* |

Instead of starting from scratch for all plots, we can plot fit to display all four graphs

|  |
| --- |
| *```{r}*  *# Include group as a predictor in the model*  *par(mfrow=c(2,2))*  *plot(fit)*  *```* |

### Adding a predictor

|  |
| --- |
| *```{r 3rd variable plot}*  *# Adding the third variable crop using different colors*  *plot(GroupData$day[GroupData$crop == 'A'], GroupData$weight[GroupData$crop == 'A'], pch=1, col="red", ylim=c(0,10), xlim=c(3,10))*  *points(GroupData$day[GroupData$crop =='B'], GroupData$weight[GroupData$crop == 'B'], pch=3, col="blue")*  *legend("topright", legend=c("A", "B"), pch=c(1,3), col=c("red", "blue"), cex=0.8)*  *```* |

Allow each line to have a different weight depending on crop group

|  |
| --- |
| *```{r fit2}*  *# Having a different height depending on the crop group*  *fit2 <- lm(GroupData$height~GroupData$day+factor(GroupData$crop))*  *summary(fit2)*  *```* |

Allow each line to have different heights & slopes depending on crop group

|  |
| --- |
| *```{r fit3 }*  *# Having a different height & slopes depending on the crop group*  *fit2 <- lm(GroupData$height~GroupData$day\*factor(GroupData$crop))*  *summary(fit2)*  *```* |

**## One-Factor ANOVA #7**

ANOVA is a parametric method appropriate for comparing the Means for two or more independent populations

We can conduct an analysis of variance using the "aov" command/function in R

|  |
| --- |
| *```{r aov}*  *# Compare Weight separated by Crop type*  *#Store aov results in R object*  *anova1 <- aov(GroupData$weight~GroupData$crop)*  *anova1*  *```* |

Call summary function to return the Sum of Squares, the Mean Squares between groups [‘crop’ row], and errors [‘residuals’ row] of computing F test statistic.

F test statistic and P-value for test of:

H0: all population means are the same

HA: at least one population mean differs

|  |
| --- |
| *```{r aov summary}*  *# Use summary command to display ANOVA table*  *summary(anova1)*  *```* |

Assessing conditions

|  |
| --- |
| *```{r QQ plots of residuals}*  *# Produce Normal Quantile (QQ) plots of residuals:*  *qqnorm(anova1$residuals)*  *qqline(anova1$residuals)*  *```* |

|  |
| --- |
| *```{r QQ plots of response for each group}*  *# Produce Normal Quantile (QQ) plots of response for each group:*  *par(mfrow=c(1,2))*  *qqnorm(GroupData$weight[GroupData$crop=="A"])*  *qqnorm(GroupData$weight[GroupData$crop=="B"])*  *```* |

|  |
| --- |
| *```{r stripchart of residuals}*  *# Produce stripchart of residuals*  *stripchart(anova1$residuals~GroupData$crop, vertical=TRUE)*  *```* |

Instead of starting from scratch for all plots, we can plot anova to display all four graphs

|  |
| --- |
| *```{r plot anova}*  *# Use plot(anoval1) to display plots for diagnostics*  *par(mfrow=c(2,2))*  *plot(anova1)*  *```* |

**# Two sample t-tests for pairwise comparisons:**

Paired t-test is a parametric approach (or large sample approach) used to compare means of two paired groups (dependent groups or matched groups)

|  |
| --- |
| *```{r pairwise t test}*  *# Data vectors: response=weight; factor = crop*  *pairwise.t.test(GroupData$weight, GroupData$crop)*  *```* |

**# Tukey's HSD tests and confidence intervals:**

TukeyHSD function generates P values and confidence intervals for all possible pairwise populations when combining means of more than two populations.

|  |
| --- |
| *```{r pairwise t test}*  *# Data vectors: response=weight; factor = crop*  *TukeyHSD (anova1)*  *# set confidence intervals to 0.90*  *TukeyHSD(anova1, conf.level = 0.90)*  *```* |

We can use multiple comparisons to help us decide which Means or Crops may differ from the others.

Here we use “TukeyHSD” function to conduct all possible pair-wise comparisons for this analysis of variance fit.

We are returned overall 95% confidence intervals for the difference in Means of pairs B-A; we’re also returned an adjusted p-value

|  |
| --- |
| *```{r TukeyHSD}*  *# Summary of anova1 object*  *TukeyHSD(anova1)*  *```* |

We can see a visual display by using “plot”; using “las=1” to rotate the labels on the y-axis

|  |
| --- |
| *```{r TukeyHSD plot}*  *# Summary of anova1 object*  *plot(TukeyHSD(anova1), las=1)*  *```* |

Producing the Kruskal-Wallis one-way analysis of variance using ranks.

Kruskal Wallis One-way Analysis of Variance is a non-parametric equivalent to the one-way Analysis of Variance. We can conduct this test in R using the “kruskal.test” command

|  |
| --- |
| *```{r Kruskal-Wallis}*  *# Kruskal-Wallis rank sum test*  *kruskal.test(GroupData$weight~GroupData$crop)*  *```* |

