Recitation - Week 4

------ 1. Warm up ---------

Exercise 1.

1. Download "Lizards.csv" data from Trunk.
2. Read in csv file to R
3. Calculate mean, standard deviation, variance, range and standard error of horn length of live vs killed horned lizards.
4. Plot a histogram of horn length of live vs killed horned lizards.

* **Histogram binning**: Rules of Thumb (**always** use this guideline in this class!)

|  |  |
| --- | --- |
| Sample Size | Number of Bins |
| Less than 16 | Not Enough Data |
| 16--31 | 5 |
| 32--63 | 6 |
| 64--127 | 7 |
| 128--255 | 8 |

lizard<-read.csv("Lizards.csv")  
head(lizard)

## HornLength Survival  
## 1 25.2 living  
## 2 26.9 living  
## 3 26.6 living  
## 4 25.6 living  
## 5 25.7 living  
## 6 25.9 living

table(lizard$Survival)

##   
## killed living   
## 30 155

library(dplyr)

##   
## Attaching package: 'dplyr'

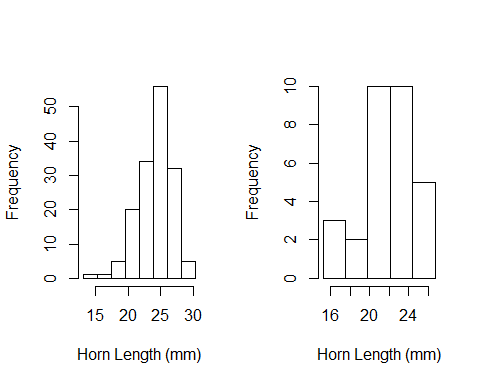
## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

lizard %>% filter(!is.na(HornLength)) %>% group\_by(Survival) %>% summarise(mean=mean(HornLength),  
 sd = sd(HornLength),  
 var = var(HornLength),  
 range = max(HornLength) - min(HornLength),  
 se = sd(HornLength)/sqrt(n()))

## # A tibble: 2 x 6  
## Survival mean sd var range se  
## <fctr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 killed 21.98667 2.709464 7.341195 11.5 0.4946782  
## 2 living 24.28117 2.630782 6.921016 17.2 0.2119945

bins <- function(x, bin.num){  
 seq(min(x), max(x), (max(x) - min(x)) / bin.num)  
}  
  
living=subset(lizard, Survival=="living" & !is.na(HornLength))  
killed=subset(lizard, Survival=="killed" & !is.na(HornLength))  
  
par(mfrow=c(1,2))  
hist(living$HornLength, breaks=bins(living$HornLength, 8),  
 xlab="Horn Length (mm)", main="")  
hist(killed$HornLength, breaks=bins(killed$HornLength, 5),  
 xlab="Horn Length (mm)", main="")



----- 2. Introduce t.test function ----

*x* = for one sample t.test

*y* = optional (for two sample t.test)

*mu* = 0 for default (true mean for 1 sample, or difference if two sample) <- NULl HYPOTHESIS

*alternative* = default is two.sided/non-directional (is the pop mean different from mu0), pop mean "greater" than sample, or pop mean "less" than sample mean.

QUESTION: What if you were conducting a two sample t.test and you left the default where mu = 0 - what would your null hypothesis be? ANSWER: Null hypothesis is that your two samples have no differences in their means.

*paired* = default is false (meaning samples are independent). TRUE means that each sample of each group is paired together, e.g. twin studies.

*var.equal* = default is false, and therefore different variances for each group are calculated. Welsh

bee<-read.csv("BeeNectar.csv")  
head(bee)

## Colony BeeID Pollen ITS Nectar  
## 1 57 B91 0.00000 4.03 0.00027  
## 2 41 838E52010B000000 0.00153 2.87 0.00033  
## 3 41 798C52010B000000 0.02089 3.19 0.00035  
## 4 57 B58 0.00034 3.07 0.00063  
## 5 58 G49 0.00000 3.42 0.00065  
## 6 32 R24 0.02855 3.86 0.00114

# Example 1  
bee.test<-with(bee, t.test(Pollen), mu=0)   
bee.test # explain all the output

##   
## One Sample t-test  
##   
## data: Pollen  
## t = 8.5888, df = 524, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 0.001538212 0.002450550  
## sample estimates:  
## mean of x   
## 0.001994381

names(bee.test) # outputs of model

## [1] "statistic" "parameter" "p.value" "conf.int" "estimate"   
## [6] "null.value" "alternative" "method" "data.name"

bee.test$parameter

## df   
## 524

bee.test$estimate

## mean of x   
## 0.001994381

bee.test$statistic # t statistic ... or the standard error of the mean, e.g. how well known is the mean based on the sample

## t   
## 8.588843

# Example 2  
  
lizard <- read.csv("Lizards.csv")  
head(lizard)

## HornLength Survival  
## 1 25.2 living  
## 2 26.9 living  
## 3 26.6 living  
## 4 25.6 living  
## 5 25.7 living  
## 6 25.9 living

with(lizard, t.test(HornLength ~ Survival))

##   
## Welch Two Sample t-test  
##   
## data: HornLength by Survival  
## t = -4.2634, df = 40.372, p-value = 0.0001178  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.381912 -1.207092  
## sample estimates:  
## mean in group killed mean in group living   
## 21.98667 24.28117

----- 3. Introduce x-t functions ------

We can use other functions, qt and pt to ask questions about the sampling distribution for T tests.

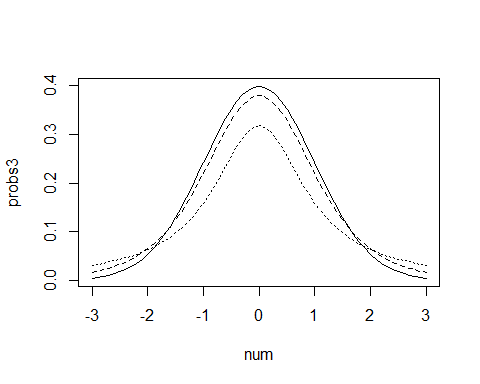
# What is the probability density of critical values given degrees of freedom....   
  
dt(0, df = 10) # probability T distribution function - returns the area under the T distribution

## [1] 0.3891084

pt(0, df = 10) # returns the cumulative area under the T distribution

## [1] 0.5

# If we wanted to plot the probability density of critical values of the T distribution, given we know the degrees of freedom.  
num<-seq(-3, 3, 0.1)  
probs1<-dt(num, df=1)  
probs2<-dt(num, df=5)  
probs3<-dt(num, df=300)  
  
plot(num, probs3, type="l", lty=1)  
points(num, probs2, type="l", lty=2)  
points(num, probs1, type="l", lty=3)



# What is the critical value, given we know the degrees of freedom and the significance level. This is the T table!  
qt(0.85, df = 1) # T value in the 97.5% percentile is less than 1.96 if df = 524

## [1] 1.962611

qt(0.85, df = 1)

## [1] 1.962611

-------- Exercise 1. ---------

Problem: biologists were interested in whether worker size varies significantly among colonies across the landscape.

1. If the population mean (mu) for bumble bee forager size (ITS) is 3.81, do you think your sample is representative of the whole population? In other words, are the population and sample means similar? If not, is your sample greater or less than the population mean?
2. Are the sample means for foraging worker size for colony 41 and colony 43 different from each other? If not, which colony has greater bumble bee worker size (report p-value)? Repeat question 2 for colony 32 and colony 2.
3. Calculate the mean, standard deviation, standard error, and confidence intervals for forager size for each colony, except Colony 22. [Hint: dplyr might be good here]
4. Create three barplots for mean body size (ITS) for each colony (excluding colony 22) with error bars representing (i) standard deviation, (ii) standard error and (iii) 95% confidence intervals. [HINT: use arrows() function for creating error bars]

bee<-read.csv("BeeNectar.csv")  
head(bee)

## Colony BeeID Pollen ITS Nectar  
## 1 57 B91 0.00000 4.03 0.00027  
## 2 41 838E52010B000000 0.00153 2.87 0.00033  
## 3 41 798C52010B000000 0.02089 3.19 0.00035  
## 4 57 B58 0.00034 3.07 0.00063  
## 5 58 G49 0.00000 3.42 0.00065  
## 6 32 R24 0.02855 3.86 0.00114

m1<-with(bee, t.test(ITS, mu=3.81))  
m1

##   
## One Sample t-test  
##   
## data: ITS  
## t = 1.7083, df = 523, p-value = 0.08817  
## alternative hypothesis: true mean is not equal to 3.81  
## 95 percent confidence interval:  
## 3.805473 3.874909  
## sample estimates:  
## mean of x   
## 3.840191

m2<-with(bee, t.test(ITS, mu=3.81, alternative="greater")) # sample mean is greater than population mean.   
m2

##   
## One Sample t-test  
##   
## data: ITS  
## t = 1.7083, df = 523, p-value = 0.04408  
## alternative hypothesis: true mean is greater than 3.81  
## 95 percent confidence interval:  
## 3.81107 Inf  
## sample estimates:  
## mean of x   
## 3.840191

m3<-with(bee, t.test(ITS, mu=3.81, alternative="less"))  
m3

##   
## One Sample t-test  
##   
## data: ITS  
## t = 1.7083, df = 523, p-value = 0.9559  
## alternative hypothesis: true mean is less than 3.81  
## 95 percent confidence interval:  
## -Inf 3.869311  
## sample estimates:  
## mean of x   
## 3.840191

m4<-t.test(subset(bee, Colony=="41")$ITS, subset(bee, Colony=="43")$ITS)  
m4

##   
## Welch Two Sample t-test  
##   
## data: subset(bee, Colony == "41")$ITS and subset(bee, Colony == "43")$ITS  
## t = -0.84498, df = 146.99, p-value = 0.3995  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.22351919 0.08962667  
## sample estimates:  
## mean of x mean of y   
## 3.708642 3.775588

m5<-t.test(subset(bee, Colony=="32")$ITS, subset(bee, Colony=="2")$ITS, alternative="less")  
m5

##   
## Welch Two Sample t-test  
##   
## data: subset(bee, Colony == "32")$ITS and subset(bee, Colony == "2")$ITS  
## t = 2.9602, df = 96.502, p-value = 0.9981  
## alternative hypothesis: true difference in means is less than 0  
## 95 percent confidence interval:  
## -Inf 0.2844462  
## sample estimates:  
## mean of x mean of y   
## 4.078667 3.896452

library(dplyr)  
  
table(bee$Colony) # number of observations within each colony. Colony 22 does not have enough

##   
## 2 22 32 35 41 43 57 58   
## 63 1 45 62 81 68 169 36

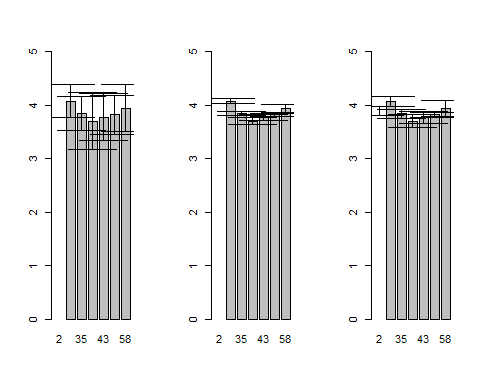
# Solution 1  
bee.m22<-subset(bee, Colony!="22")  
bee.m22 %>% group\_by(Colony) %>% summarise(mean=mean(ITS, na.rm=T),  
 sd=sd(ITS, na.rm=T),  
 se=sd(ITS, na.rm=T)/sqrt(n()),  
 CI.upper=t.test(ITS)$conf.int[1],  
 CI.lower=t.test(ITS)$conf.int[2])

## # A tibble: 7 x 6  
## Colony mean sd se CI.upper CI.lower  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 3.896452 0.3195032 0.04025362 3.815313 3.977590  
## 2 32 4.078667 0.3105098 0.04628807 3.985379 4.171954  
## 3 35 3.846290 0.3239320 0.04113940 3.764027 3.928554  
## 4 41 3.708642 0.5277186 0.05863540 3.591954 3.825330  
## 5 43 3.775588 0.4393733 0.05328184 3.669237 3.881939  
## 6 57 3.821243 0.3598201 0.02767847 3.766600 3.875885  
## 7 58 3.950278 0.4382269 0.07303781 3.802003 4.098552

#Solution 2  
sums<-bee %>% filter(Colony!="22") %>% group\_by(Colony) %>% summarise(mean=mean(ITS),  
 sd=sd(ITS),  
 se=sd(ITS)/sqrt(n()),  
 CI.upper=t.test(ITS)$conf.int[1],  
 CI.lower=t.test(ITS)$conf.int[2])  
sums

## # A tibble: 7 x 6  
## Colony mean sd se CI.upper CI.lower  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 NA NaN NaN 3.815313 3.977590  
## 2 32 4.078667 0.3105098 0.04628807 3.985379 4.171954  
## 3 35 3.846290 0.3239320 0.04113940 3.764027 3.928554  
## 4 41 3.708642 0.5277186 0.05863540 3.591954 3.825330  
## 5 43 3.775588 0.4393733 0.05328184 3.669237 3.881939  
## 6 57 3.821243 0.3598201 0.02767847 3.766600 3.875885  
## 7 58 3.950278 0.4382269 0.07303781 3.802003 4.098552

par(mfrow=c(1,3))  
barCenters<-with(sums, barplot(mean, names.arg=Colony, ylim=c(0,5)))  
arrows(barCenters, sums$mean-sums$sd, barCenters, sums$mean+sums$sd, angle=90, code=3)  
  
barCenters<-with(sums, barplot(mean, names.arg=Colony, ylim=c(0,5)))  
arrows(barCenters, sums$mean-sums$se, barCenters, sums$mean+sums$se, angle=90, code=3)  
  
barCenters<-with(sums, barplot(mean, names.arg=Colony, ylim=c(0,5)))  
arrows(barCenters, sums$CI.upper, barCenters, sums$CI.lower, angle=90, code=3)



-------- Exercise 2. ---------

Problem: biologists were interested in whether introduced brook trout to affects the survival of chinook salmon.

1. Does the presence or absence of introduced trout affect the proportion of chinook salmon surviving?
2. Create three barplot for mean proportion surviving for both treatments with error bars representing (i) standard deviation, (ii) standard error and (iii) 95% confidence intervals.

salmon<-read.csv("BrookTrout.csv")  
head(salmon)

## troutTreatment nReleased nSurvivors proportionSurvived  
## 1 present 820 166 0.202  
## 2 absent 467 180 0.385  
## 3 present 960 136 0.142  
## 4 present 700 153 0.219  
## 5 absent 959 178 0.186  
## 6 present 545 103 0.189

salmon.t<-with(salmon, t.test(proportionSurvived ~ troutTreatment))  
salmon.t

##   
## Welch Two Sample t-test  
##   
## data: proportionSurvived by troutTreatment  
## t = 0.93148, df = 5.8165, p-value = 0.3886  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.06753173 0.14953173  
## sample estimates:  
## mean in group absent mean in group present   
## 0.2351667 0.1941667

salmon.stats<-salmon %>% group\_by(troutTreatment) %>% summarise(mean=mean(proportionSurvived),  
 sd=sd(proportionSurvived),  
 se=sd(proportionSurvived)/sqrt(n()),  
 CI.upper=t.test(proportionSurvived)$conf.int[1],  
 CI.lower=t.test(proportionSurvived)$conf.int[2])  
salmon.stats

## # A tibble: 2 x 6  
## troutTreatment mean sd se CI.upper CI.lower  
## <fctr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 absent 0.2351667 0.10364056 0.04231108 0.1264026 0.3439308  
## 2 present 0.1941667 0.02971476 0.01213100 0.1629829 0.2253504

par(mfrow=c(1,3))  
barCenters<-with(salmon.stats, barplot(mean, names.arg=troutTreatment, ylim=c(0,1), ylab="Proportion surviving"))  
arrows(barCenters, salmon.stats$mean-salmon.stats$sd, barCenters, salmon.stats$mean+salmon.stats$sd, angle=90, code=3)  
  
barCenters<-with(salmon.stats, barplot(mean, names.arg=troutTreatment, ylim=c(0,1),ylab="Proportion surviving"))  
arrows(barCenters, salmon.stats$mean-salmon.stats$se, barCenters, salmon.stats$mean+salmon.stats$se, angle=90, code=3)  
  
barCenters<-with(salmon.stats, barplot(mean, names.arg=troutTreatment, ylim=c(0,1), ylab="Proportion surviving"))  
arrows(barCenters, salmon.stats$CI.upper, barCenters, salmon.stats$CI.lower, angle=90, code=3)

