Stat 251 RA5 reference

Your `R` tools:

`t.test()`

For 1-sample test/CI for mean when σ is not known (you will use s) and for independent and dependent 2-sample tests/CI.

`t.test()`: `t.test(x,y,alternative="",mu,conf.level=0.95,paired=F,var.equal=F,...)`
x: data vector (variable)
y: another data vector (only used in 2-sample methods)
alternative: alternative hypothesis ("g" (or "greater") for $Ha :>$, "l" ("less") for $Ha :<$, "two.sided" (default) for $Ha :\neq$)
mu: hypothesized mean
conf.level: confidence level; 0.95 (default)
paired: for paried t-test; F is default
... : other options available
x,y can also be input as a formula of `y~x` when x is numeric and y is categorical

Example for the difference of 2 means

Estimate the true difference in mean tail lengths for female and male possums with 99% confidence. Then perform hypothesis test to see if male possum tail lengths are longer than female possums (let $\alpha = 0.05$)

$$\bar{x}_1 - \bar{x}_2 \pm t^\star \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

$$H_0 : \mu_1 = \mu_2 \text{ vs. } H_a : \mu_1 > \mu_2$$

```r
library(openintro); data(possum); attach(possum)
head(possum)
```

```
  site pop sex age headL skullW totalL tailL
1    1 Vic   m   8  94.1   60.4   89.0  36.0
2    1 Vic   f   6  92.5   57.6   91.5  36.5
3    1 Vic   f   6  94.0   60.0   95.5  39.0
4    1 Vic   f   6  93.2   57.1   92.0  38.0
5    1 Vic   f   2  91.5   56.3   85.5  36.0
6    1 Vic   f   1  93.1   54.8   90.5  35.5
```

```r
# 99% CI
t.test(tailL~sex,conf.level=0.99)
```

```

    Welch Two Sample t-test

data:  tailL by sex
t = 0.42208, df = 96.526, p-value = 0.6739
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 -0.8467008  1.1707572
sample estimates:
mean in group f mean in group m 
       37.10465        36.94262 
```

```r
# hypothesis test
t.test(tailL~sex,alternative='g')
```

```

    Welch Two Sample t-test

data:  tailL by sex
t = 0.42208, df = 96.526, p-value = 0.337
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -0.4755155        Inf
sample estimates:
mean in group f mean in group m 
       37.10465        36.94262 
```

Example of paired (dependent) samples t-test/CI

A school athletics has taken a new instructor, and want to test the effectiveness of the new type of training proposed by comparing the average times of 10 runners in the 100 meters. Are below the time in seconds before and after training for each athlete. Doing a 98% CI on mean difference and hypothesis test (with $\alpha = 0.02$) to test if mean difference is greater than 0.

$$\bar{x}_d \pm t^{\star}(s_d/\sqrt{n})$$

$$H_0 : \mu_D = 0 \text{ vs. } H_a : \mu_D > 0$$

```r
a = c(12.9, 13.5, 12.8, 15.6, 17.2, 19.2, 12.6, 15.3, 14.4, 11.3)
b = c(12.7, 13.6, 12.0, 15.2, 16.8, 20.0, 12.0, 15.9, 16.0, 11.1)
# 98% CI on mean difference
t.test(a,b,paired=T,conf.level=.98)
```

```
    Paired t-test

data:  a and b
t = -0.21331, df = 9, p-value = 0.8358
alternative hypothesis: true difference in means is not equal to 0
98 percent confidence interval:
 -0.7113516  0.6113516
sample estimates:
mean of the differences
                  -0.05
```

```r
# H0: mu_d=0 Ha: mu_d>0
t.test(a,b,paired=T,conf.level=.98,alternative='g')
```

```
    Paired t-test

data:  a and b
t = -0.21331, df = 9, p-value = 0.5821
alternative hypothesis: true difference in means is greater than 0
98 percent confidence interval:
 -0.6122001        Inf
sample estimates:
mean of the differences
                  -0.05
```

Example of 2 independent proportions

Who plays online or video games? A survey in 2006 found that 69% of 223 boys aged 12-14 said they "played computer, console, or online games." of 248 boys aged 15-17, only 62% played these games. Is there evidence of an age-based difference? 95% CI/Test
Note: the se for the test is different than the one for the CI. This example gives percents as the responses from each age group rather than the count from the samples. Your $\hat{p}$ for the hypothesis test *se*

```r
# CI
phat1=0.69
n1=223
qhat1=1-phat1
phat2=0.62
n2=248
qhat2=1-phat2
alpha=0.05
zstar=qnorm(1-alpha/2)
se=sqrt(phat1*qhat1/n1+phat2*qhat2/n2)
bound=zstar*se
lower=phat1-phat2-bound
upper=phat1-phat2+bound
rbind(lower,upper)
```

```
             [,1]
lower -0.01563928
upper  0.15563928
```

```
# Test
# H0: p1=p2 Ha: p1 not= p2
x1=ceiling(phat1*n1) # ceiling() rounds up to nearest whole number
x2=ceiling(phat2*n2)
# if counts are given instead of percents, just input the counts for x1 and x2
# x1=154; x2=154
phat=(x1+x2)/(n1+n2)
qhat=1-phat
se=sqrt(phat*qhat*(1/n1+1/n2))
zcalc=(phat1-phat2)/se
pvalue=2*(1-pnorm(abs(zcalc)))
rbind(se,zcalc,pvalue)
```

```
             [,1]
se     0.04390161
zcalc  1.59447460
pvalue 0.11082978
```

Chi-square Goodness-of-Fit test (GoF)

`chisq.test(x,y,p= ,rescale.p=F,...)`
* x,y: x,y are vectors (variables); **can also input a table (dataset) name**
* p= : vector of probabilities for GoF test; use `p=rep(1/sum(datasetname),length(datasetname))`
* rescale.p=F: F is the default; T rescales p so it sums to 1 (gives error if it does not)

Example of GoF test

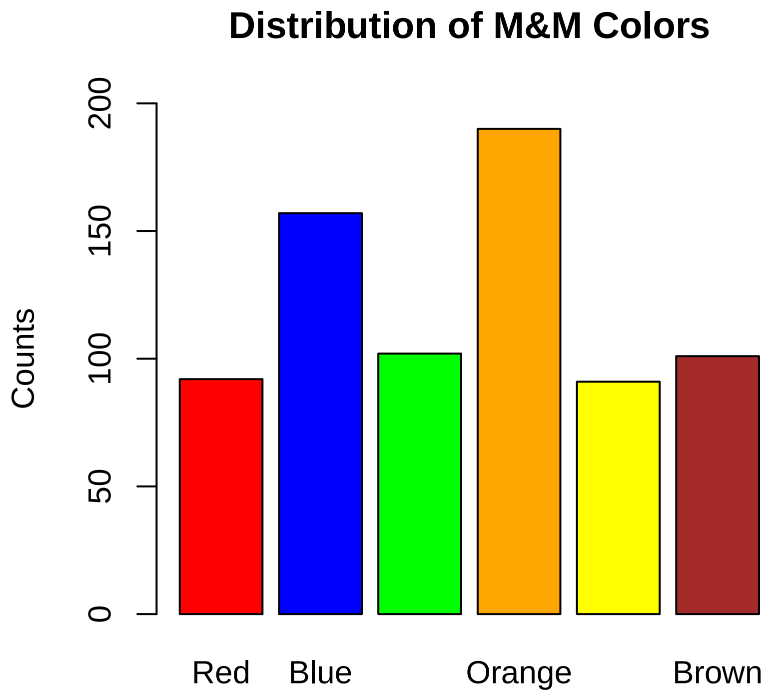
M&M's from Stat 251-04 Spring 2016

H_0: Distribution of colors for M&Ms as Mars indicates (13% red, 24% blue, 16% green, 20% orange, 14% yellow, 13% brown)
H_a: Distribution of colors for M&Ms are not as stated by Mars

```
colors=c('Red','Blue','Green','Orange','Yellow','Brown')
observed=c(92,157,102,190,91,101)
mars=c(.13,.24,.16,.2,.14,.13)
M.M=data.frame(colors,observed,mars)
M.M
```

```
  colors observed mars
1    Red       92 0.13
2   Blue      157 0.24
3  Green      102 0.16
4 Orange      190 0.20
5 Yellow       91 0.14
6  Brown      101 0.13
```

```
pi=mars*sum(observed)
# graph not necessary but fun!
gcolors=colors
barplot(observed,names.arg=colors,col=gcolors,ylim=c(0,200),ylab='Counts')
title("Distribution of M&M Colors")
```




```r
# actual test
chisq.test(observed,p=mars)
```

```
	Chi-squared test for given probabilities

data:  observed
X-squared = 18.645, df = 5, p-value = 0.002237
```

```r
# to see expected values:
chisq.test(observed,p=mars)$expected
```

```
[1]  95.29 175.92 117.28 146.60 102.62  95.29
```

Example of Independence test

How is the hatching of water python eggs influenced by the temperature of the snake's nest? Researchers assigned newly laid eggs to one of three temperatures: hot, neutral, or cold. Hot duplicates the extra warmth provided by the mother python, and cold duplicates the absence of the mother. Given the provided data, is there sufficient evidence that the success of water python eggs hatching is related to the temperature?

```r
# each `c()` are the values from a row
python=as.table(rbind(c(27,16),c(56,38),c(104,75)))
# `dimnames()` below is not completely necessary but helps to keep information organized
# each list has a name (row title first then column titles)
dimnames(python) <- list(Temperature=c("Cold","Neutral","Hot"),Eggs=c("Laid","Hatched"))
python
```

```
           Eggs
Temperature Laid Hatched
    Cold      27      16
    Neutral   56      38
    Hot      104      75
```

```r
chisq.test(python)
```

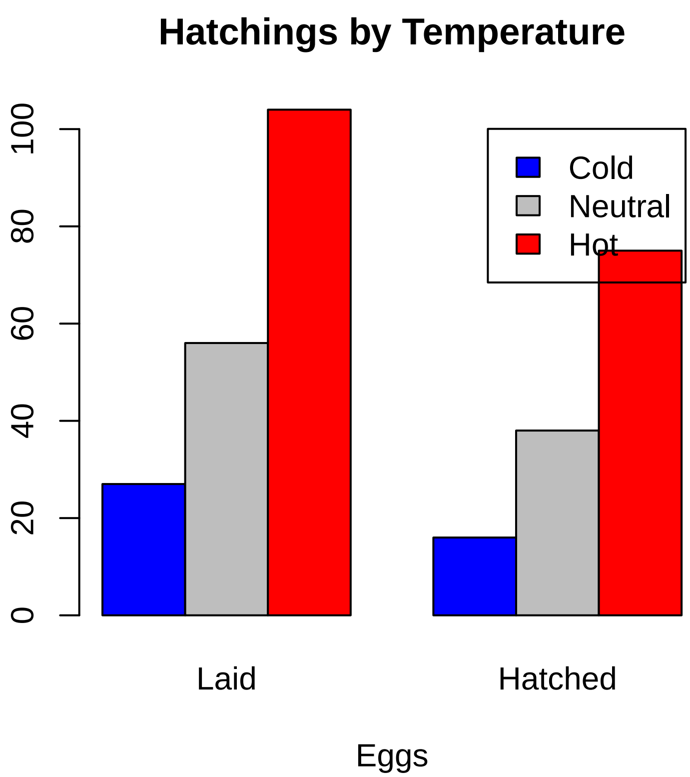
```
	Pearson's Chi-squared test

data:  python
X-squared = 0.32445, df = 2, p-value = 0.8503
```

```
# to see expected values:
chisq.test(python)$expected
```

```
           Eggs
Temperature      Laid  Hatched
    Cold     25.44620 17.55380
    Neutral  55.62658 38.37342
    Hot     105.92722 73.07278
```

```
# graph; again not necessary but fun!
data=as.data.frame(python)
counts <- xtabs(Freq~Temperature+Eggs,data=data)
barplot(counts,main="Hatchings by Temperature",
  xlab="Eggs",col=c("blue","grey","red"),
  legend=rownames(counts),beside=TRUE)
```



Regression (SLR: simple linear regression)

`fit=lm(y~x,data= ,...)` with `summary(fit)`:
`lm()` is the linear model function
`y~x`: the "formula" for the linear model, x and y must be numeric (variable names) '

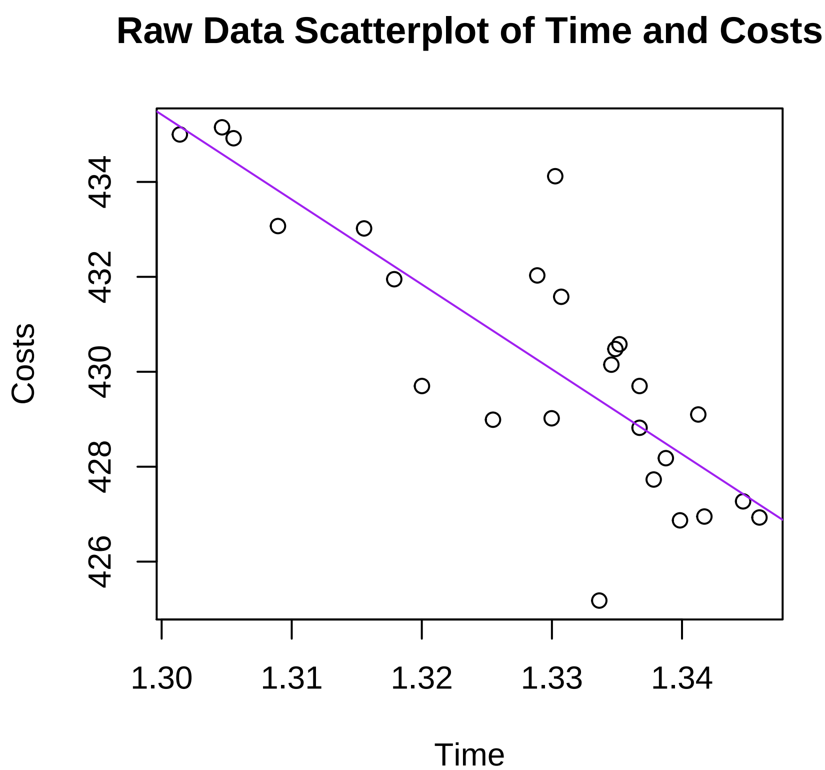
Looking at the time and cost associated with the production of one product. Of interest is using time to model costs. Follow the checklist from class.

(1) Scatterplot
(2) Hypothesis test for slope
(3) R^2 (coefficient of determination)
(4) r (correlation)

```r
decagon # you will paste your data here from assignment
```

```
       Time  Costs
1  1.337824 427.73
2  1.301401 435.00
3  1.341250 429.10
4  1.330715 431.58
5  1.344695 427.27
6  1.333642 425.18
7  1.305530 434.92
8  1.328873 432.03
9  1.308942 433.07
10 1.335188 430.58
11 1.341719 426.95
12 1.345952 426.93
13 1.334569 430.15
14 1.338757 428.18
15 1.336737 428.82
16 1.334878 430.48
17 1.304643 435.15
18 1.330254 434.12
19 1.336737 429.70
20 1.339846 426.87
21 1.315560 433.02
22 1.317880 431.95
23 1.320010 429.70
24 1.325470 428.99
25 1.329980 429.02
```

```r
# create linear model y~x
# y=costs, x=time
fit=lm(Costs~Time)
# plot x and y
plot(Time,Costs,main='Raw Data Scatterplot of Time and Costs')
# with regression line
abline(fit,col='purple')
```




```
# regression analysis results
# including hypothesis test for slope
# H0: Beta1=0 Ha: Beta1 not= 0
summary(fit)
```

```
Call:
lm(formula = Costs ~ Time)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.2211 -0.9232 -0.1543  0.9147  4.1129 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   667.92      34.03  19.626 7.29e-16 ***
Time         -178.85      25.61  -6.984 4.06e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.638 on 23 degrees of freedom
Multiple R-squared:  0.6795,    Adjusted R-squared:  0.6656 
F-statistic: 48.77 on 1 and 23 DF,  p-value: 4.062e-07
```

```
# extracting coefficients for equation
a=coefficients(fit)[[1]]; a
```

```
[1] 667.9208
```

```
b=coefficients(fit)[[2]]; b
```

```
[1] -178.8483
```

```
# yhat=a+bx
# calculate yhat with x=1.3
x1=1.3
```

```r
yhat1=a+b*x1
yhat1
```

```
[1] 435.418
```

```r
# another one: x=1.34
x2=1.34
yhat2=a+b*x2
yhat2
```

```
[1] 428.264
```

```r
# residuals e=y-yhat
# observed y values
y1=decagon[12,2]
y2=decagon[17,2]
e1=y1-yhat1; e1
```

```
[1] -8.487958
```

```r
e2=y2-yhat2; e2
```

```
[1] 6.885976
```

```r
# R-squared (called 'Multiple R-sq' on output)
R2=summary(fit)$r.squared
R2
```

```
[1] 0.6795443
```

```r
# slope is negative here (remove '-' if yours is +)
r=-sqrt(R2)
r
```

```
[1] -0.8243448
```

```r
# there is a function for correlation
cor(Time,Costs)
```

```
[1] -0.8243448
```