PSYC 6780: Lab Tutorial 5

Chris M. Fiacconi

Common Inferential Tests

R is a great tool for conducting statistical analyses. In this tutorial we will learn how to implement various statisitical tests using the *R* software platform. In particular, we will use the following functions to illustrate how you can use *R* to run statistical tests:

```
t.test()
pairwise.t.test()
aov()
ezANOVA() # from 'ez' package - install.packages("ez")
lm()
cor()
cor.test()
binom.test()
```

One-Sample *t*-test

Let's start with a one-sample *t*-test. Such tests are useful when the goal is to evaluate whether the mean from a sample (M) you collected comes from population with a mean that is different from some pre-specified value (μ).

```
set.seed(8000)
one_sample_data<-rnorm(n=40,mean=0,sd=5)
t.test(one_sample_data) # default is a test against mu = 0
```

```
##
##  One Sample t-test
##
## data:  one_sample_data
## t = -1.1969, df = 39, p-value = 0.2386
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -2.2646030  0.5808779
## sample estimates:
##  mean of x
## -0.8418626
```

```
one_sample_data2<-rnorm(n=40,mean=.7,sd=.4)
t.test(one_sample_data2,mu=.5) # Can specify value of mu (e.g., chance = 0.5)
```

```
##
##  One Sample t-test
##
## data:  one_sample_data2
## t = 1.5154, df = 39, p-value = 0.1377
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.4659403 0.7375541
## sample estimates:
## mean of x
## 0.6017472
```

Independent-Groups *t*-test

What if we wanted to know whether Guelph students drank more or less than Western students? How could we evaluate the null hypothesis that U of G and Western students consume an equal number of drinks per week? We need to take two independent samples - one from Guelph and one from Western and compare them. Let's set the population parameters for Western and Guelph students.

```
set.seed(41273)
UWOpop<-rnorm(n=30000,mean=18,sd=6) # Create population - 30,000 students, mean = 18, sd=6
UGpop<-rnorm(n=20000,mean=16,sd=6) # Create population - 20,000 students, mean = 16, sd=6
```

Now, let's draw a sample of 200 students from each university

```
set.seed(92742)

UGsample<-sample(x=UGpop,size=200)
UWOsample<-sample(x=UWOpop,size=200)

# Combine into dataframe
scores<-c(UGsample,UWOsample)
school<-rep(c("UG","UWO"),times=1,each=200)
drinkdata<-data.frame(school,scores)

head(drinkdata) # take a peek at the new data frame with a column for school and scores
```

```
##   school   scores
## 1     UG 20.41871
## 2     UG 14.12249
## 3     UG 12.84645
## 4     UG 17.64575
## 5     UG 15.32508
## 6     UG 11.38887
```

```
# Create boxplot
library(tidyverse)

ggplot(drinkdata,aes(x=school,y=scores)) +
  geom_boxplot() +
  theme_classic(base_size=14) +
  labs(x="School",y="Drinks/Week")
```

To evaluate the null hypothesis that Guelph and Western students consume an equal number of drinks per week, we need to run an independent-samples *t*-test.

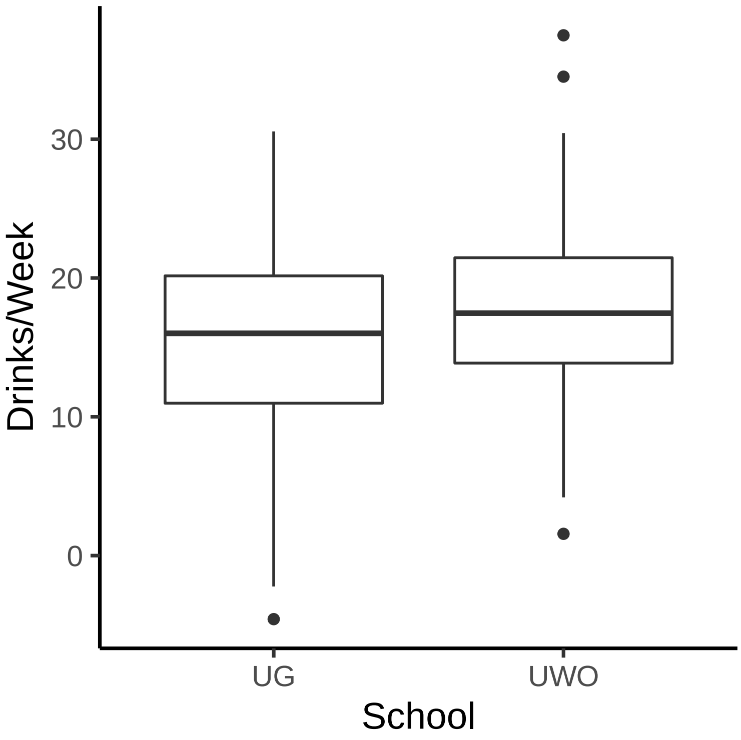


Figure 1: Boxplot depicting number of drinks consumed by students at two Ontario universities

```
t.test(scores~school,paired=F,var.equal=T,data=drinkdata)
```

```
##
##  Two Sample t-test
##
## data:  scores by school
## t = -2.5328, df = 398, p-value = 0.0117
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.8284519 -0.3563822
## sample estimates:
##  mean in group UG mean in group UWO
##          15.89340          17.48582
```

Our p-value is $< .05$, so we reject the null hypothesis that UG and UWO students consume an equal number of drinks per week.

Effect Size

What is our effect size? For an independent groups design Cohen's d is calculated as follows:

$$d = \frac{\bar{Y_2} - \bar{Y_1}}{SD_{pooled}}$$

Again, let's let the software do the work. We'll install and load the *effsize* package and use the function **cohen.d**.

Calculate Cohen's *d*:

```
install.packages("effsize") # install package - make sure you're online!!
```

```
library(effsize) # load in the effsize package
chn_d<-cohen.d(scores~school,data=drinkdata,noncentral=TRUE)
print(chn_d)
```

```
##
## Cohen's d
##
## d estimate: -0.2532777 (small)
## 95 percent confidence interval:
##       lower       upper
## -0.44990409 -0.05633554
```

Although the estimate of Cohen's *d* is -.27, the negative sign is unimportant and simply reflects the order in which the two means were subtracted. What's important is the absolute magnitude, so for our purposes, let's interpret d = -.27 as d = .27. This would be considered a small effect according to Cohen's criteria (.2 to .5 = small; .5 to .8 = medium; > .8 = large).

You'll also notice that a 95% CI for Cohen's *d* is provided by this function. Similar to 95% CI's around sample means, this CI represents a range of plausible values within which the true *population* effect size is likely to fall.

Paired-samples *t*-test

Although some research questions necessitate the use of an independent-groups design, repeated-measure designs (or *paired designs* in the case of a study with two conditions), are generally a more powerful design. This increase in statistical power stems from the fact that each individual is tested in both experimental conditions, and so variability in our dependent measure owing to differences *between* individuals can be ignored. This can drastically reduce the amount of error variability in our design, and consequently makes it easier to detect a true difference between population means.

Let's go through an example: A researcher wants to know if people can improve their math skills with practice. The researcher takes a random sample of 20 individuals, and tests the speed with which they complete 40 multiplication questions on two consecutive days. Because each participant is tested twice (once on each day), this constitutes an example of a paired design - that is, each participant's score on day one is *paired* with his/her score on day two.

First, let's load in the data file:

```
math_data<-read.table(file="mathdata.txt",header=T)
```

Let's take a peek at the data! Note that it is arranged in long format:

```
head(math_data,n=10)
```

```
##    Subject  Day CompleteTime
## 1        1 Day1     4.829410
## 2        2 Day1     4.480651
## 3        3 Day1     3.696141
## 4         4 Day1     4.600279
## 5         5 Day1     6.779562
## 6         6 Day1     4.802778
## 7         7 Day1     5.658276
## 8         8 Day1     5.267206
## 9         9 Day1     3.200148
## 10       10 Day1     4.969112
```

Notice how the data are organized - separate columns for Subject, Day, and our dependent variable (CompleteTime) - **This is important**

Now let's get the means for each condition, and then plot the data to see how it looks:

```
math_means<-math_data %>% group_by(Day) %>% summarize(Mean=mean(CompleteTime))
print(math_means)
```

```
## # A tibble: 2 x 2
##   Day    Mean
##   <fct> <dbl>
## 1 Day1   5.51
## 2 Day2   4.60
```

```
ggplot(math_means,aes(x=Day,y=Mean)) +
  geom_bar(stat="identity") +
  geom_line(data=math_data,aes(x=Day,y=CompleteTime,group=Subject)) +
  theme_classic(base_size=14) +
  labs(y="Time to Completion (sec)")
```

In this plot, the gray bars represent the mean completion times (in seconds) for Day 1 and Day 2. The filled circles represent individual data points, and the lines connect data points from the same subject.

A few things should be noted here:

- 1) Not all subjects show a reduction in completion time from Day 1 to Day 2.
- 2) There are individual differences in completion time within each Day condition, but it doesn't matter! We only care about how a given individual *changes* from Day 1 to Day 2

How do we evaluate the null hypothesis that the completion times in the population do NOT differ by Day?

We need our *t*-test to account for the fact that we no longer need to worry about differences between individuals in completion times, but rather we care only about how *consistent* the reduction in completion times are from Day 1 to 2 **within** individuals.

```
t.test(CompleteTime~Day,paired=T,data=math_data)
```

```
##
##  Paired t-test
##
## data:  CompleteTime by Day
## t = 3.655, df = 19, p-value = 0.001684
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

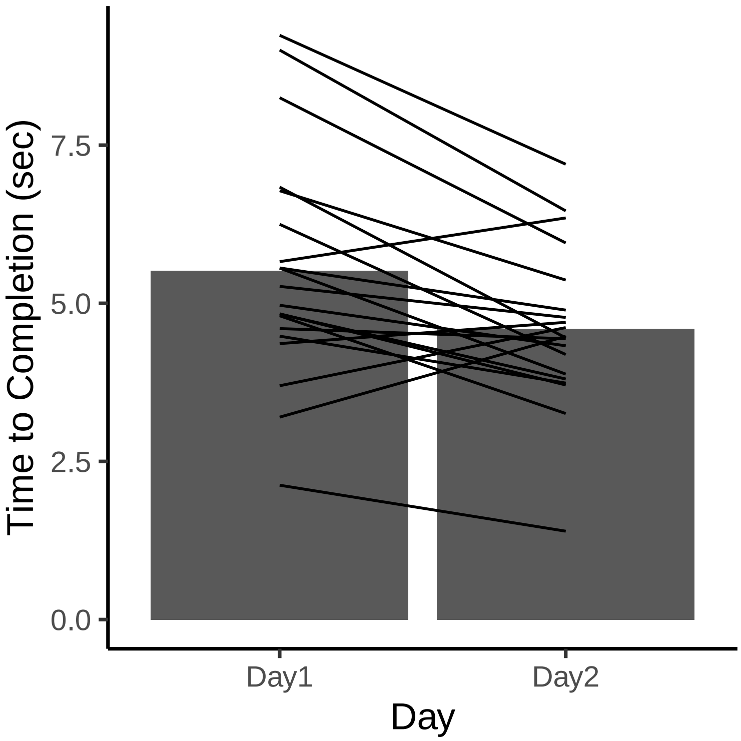


Figure 2: Bar graph showing math performance as a function of Day. Lines represent performance of individual subjects on days 1 and 2.

```
##  0.3911375 1.4393710
## sample estimates:
## mean of the differences
##               0.9152543
```

Note how I change the *paired* argument in the t.test function to **TRUE.** Our p-value is $< .05$, so we reject the null hypothesis that the time to solve math problems does not change with practice.

Effect Size

What is our effect size? For a paired design, Cohen's d can be calculated as follows:

$$d = \frac{\bar{Y_2} - \bar{Y_1}}{SD_{avg}}$$

We'll use the **cohensd_rm** function in the *itns* package built by Geoff Cumming & Robert Calin-Jageman.

```
install.packages("remotes") # make sure you're online!!
remotes::install_github("gitrman/itns") # use install_github function from remotes package
```

```
library(itns) # load itns package into R

# Store Completion Times for each Day in seperate variables
d1<-filter(math_data,Day=="Day1")[,3] # Get completion times for Day 1
d2<-filter(math_data,Day=="Day2")[,3] # Get completion times for Day 2
```

```
# Calculate Cohen's d
chn_d_rm<-cohensd_rm(x=d1,y=d2,ci=95) # Get Cohen's d for paired design
print(chn_d_rm)
```

```
##       est        ll        ul
## 0.5835008 0.2146888 0.9409143
```

The *est* column contains our estimate of the population effect size (d = .58), and the *ll* and *ul* columns represent the lower and upper limits of the 95% CI for the population effect size.

Analysis of Variance (ANOVA)

To this point, we have considered analysis of designs with no more than two levels of an independent variable (factor). However, many designs consist of at least three levels of a single factor, and others consist of multiple levels of multiple factors (i.e., factorial designs). Let's start with a one-way ANOVA to analyze a design with a single factor with three levels manipulated between subjects.

One-Way ANOVA (between)

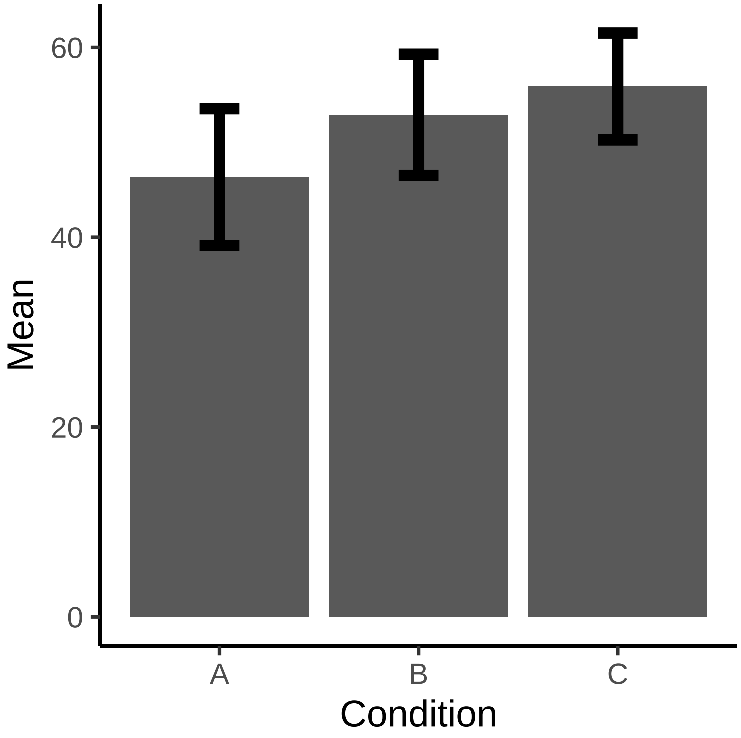
```
b_one_way_data<-read.table(file="L3data1.txt",header=TRUE)
print(b_one_way_data)
```

```
##    Subj Condition       DV
## 1     1         A 48.48704
## 2     2         A 43.80733
## 3     3         A 48.70364
## 4     4         A 52.96105
## 5     5         A 37.71348
## 6     6         B 47.38295
## 7     7         B 61.25589
## 8     8         B 53.34239
## 9     9         B 51.30100
## 10   10         B 51.22754
## 11   11         C 58.19738
## 12   12         C 48.25902
## 13   13         C 57.14311
## 14   14         C 60.03119
## 15   15         C 55.80871
```

```
# Get mean of each condition
b_one_way_means<-b_one_way_data %>% group_by(Condition) %>%
  summarize(Mean=mean(DV),SD=sd(DV),nObs=n(),SEM=sd(DV)/sqrt(n()))

# Get critical t-value to compute MoE for 95% CIs
t_crit<-qt(p=.975,df=4) # p=.975 for two-tailed bounds

# Add MoE column to b_one_way_means
b_one_way_means<-b_one_way_means %>% mutate(CI=t_crit*SEM)

print(b_one_way_means)
```

```
## # A tibble: 3 x 6
##   Condition  Mean    SD  nObs   SEM    CI
##   <fct>     <dbl> <dbl> <int> <dbl> <dbl>
## 1 A          46.3  5.81     5  2.60  7.21
## 2 B          52.9  5.14     5  2.30  6.39
## 3 C          55.9  4.54     5  2.03  5.63
```

```
# Plot means with 95% CIs
ggplot(b_one_way_means,aes(x=Condition,y=Mean)) +
  geom_bar(stat="identity") +
  geom_errorbar(aes(ymin=Mean-CI,ymax=Mean+CI),size=2,width=.2) +
  theme_classic(base_size=14)
```



```
# Do one-way ANOVA
b_one_way_aov<-aov(DV~Condition,data=b_one_way_data)
print(summary(b_one_way_aov))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)  
## Condition    2  238.9  119.43   4.438 0.0361 *
## Residuals   12  323.0   26.91                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(ez) # Load ez package to use ezANOVA

b_one_way_ezaov<-ezANOVA(data=b_one_way_data,dv=DV,wid=Subj,between=Condition,detailed=TRUE)
print(b_one_way_ezaov)
```

```
## $ANOVA
##      Effect DFn DFd      SSn      SSd        F          p p<.05       ges
## 1 Condition   2  12 238.8568 322.9565 4.437566 0.03608378     * 0.4251534
##
## $`Levene's Test for Homogeneity of Variance`
##   DFn DFd     SSn      SSd        F         p p<.05
## 1   2  12 3.76051 194.4338 0.116045 0.8914223
```

If the omnibus ANOVA is significant, then you need to perform pairwise comparisons to determine which conditions differ. These pairwise comparisons are best perfored using Tukey's HSD procedure to control for the inflation of Type I Error rates with multiple comparisons (Note: If you have *a priori* contrasts of interest, this correction is unnecessary and overly conservative - use *t*-tests instead, but focus on those comparisons that are of primary theoretical interest).

```
# Only works with aov object - will not work with output from ezANOVA
b_one_way_phoc<-TukeyHSD(b_one_way_aov)
print(b_one_way_phoc)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = DV ~ Condition, data = b_one_way_data)
##
## $Condition
##         diff        lwr      upr     p adj
## B-A 6.567448 -2.1859157 15.32081 0.1541764
## C-A 9.553374  0.8000096 18.30674 0.0324591
## C-B 2.985925 -5.7674389 11.73929 0.6445035
```

One-Way ANOVA (within)

Here is an example of how to perform a repeated-measures ANOVA with a single factor with four levels:

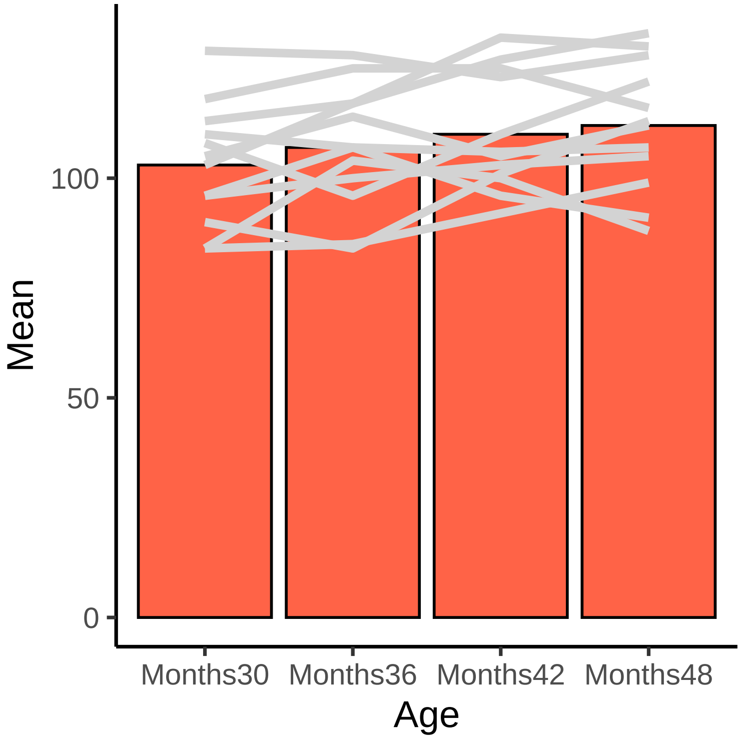
```
age_data<-read.table(file="Ch11T5.txt",header=T)
head(age_data,n=20)
```

```
##         Age Score Subject
## 1  Months30   108       1
## 2  Months30   103       2
## 3  Months30    96       3
## 4  Months30    84       4
## 5  Months30   118       5
## 6  Months30   110       6
## 7  Months30   129       7
## 8  Months30    90       8
## 9  Months30    84       9
## 10 Months30    96      10
## 11 Months30   105      11
## 12 Months30   113      12
## 13 Months36    96       1
## 14 Months36   117       2
## 15 Months36   107       3
## 16 Months36    85       4
## 17 Months36   125       5
## 18 Months36   107       6
## 19 Months36   128       7
## 20 Months36    84       8
```

```
# Always set Subject variable as a factor in within designs!!
age_data$Subject<-as.factor(age_data$Subject)

# Get descriptives for each Age group
age_descriptives<-age_data %>% group_by(Age) %>% summarize(Mean=mean(Score),S.Dev=sd(Score),nObs=n())
print(age_descriptives)
```

```
## # A tibble: 4 x 4
##   Age       Mean S.Dev  nObs
##   <fct>    <dbl> <dbl> <int>
## 1 Months30   103  13.7    12
## 2 Months36   107  14.2    12
## 3 Months42   110  13.3    12
## 4 Months48   112  14.8    12
```

```
# Create barplot to visualize the data
ggplot(age_descriptives,aes(x=Age,y=Mean)) +
  geom_bar(stat="identity",fill="tomato",color="black") +
  geom_line(data=age_data,aes(x=Age,y=Score,group=Subject),size=1.5,color="lightgray") +
  theme_classic(base_size=14)
```



```r
# Perform one-way repeated-measures ANOVA
age_aov<-ezANOVA(data=age_data,dv=Score,wid=Subject,within=Age,detailed=TRUE)
print(age_aov)
```

```
## $ANOVA
##        Effect DFn DFd    SSn  SSd          F            p p<.05        ges
## 1 (Intercept)   1  11 559872 6624 929.739130 5.586316e-12     * 0.98481975
## 2         Age   3  33    552 2006   3.026919 4.321863e-02     * 0.06011762
##
## $`Mauchly's Test for Sphericity`
##   Effect         W          p p<.05
## 2    Age 0.2426472 0.01771762     *
##
## $`Sphericity Corrections`
##   Effect       GGe     p[GG] p[GG]<.05       HFe      p[HF] p[HF]<.05
## 2    Age 0.6095445 0.0747874           0.7248502 0.06353773
```

```r
# Follow-up tests using pairwise.t.test - Bonferroni correction
age_pairwise_bonf<-with(age_data,pairwise.t.test(x=Score,g=Age,paired=TRUE,p.adjust.method = "bonf"))
print(age_pairwise_bonf)
```

```
##
##  Pairwise comparisons using paired t tests
##
## data:  Score and Age
##
##          Months30 Months36 Months42
## Months36 0.90     -        -
## Months42 0.25     1.00     -
## Months48 0.20     1.00     1.00
##
## P value adjustment method: bonferroni
```

```r
# Follow-up tests using pairwise.t.test - NO correction
age_pairwise<-with(age_data,pairwise.t.test(x=Score,g=Age,paired=TRUE,p.adjust.method = "none"))
print(age_pairwise)
```

```
##
##  Pairwise comparisons using paired t tests
##
## data:  Score and Age
##
##          Months30 Months36 Months42
## Months36 0.149    -        -
## Months42 0.042    0.300    -
## Months48 0.033    0.271    0.390
##
## P value adjustment method: none
```

Factorial ANOVA

A common design in experimental psychology is the factorial design in which two or more independent variables are crossed. Each variable can be manipulated between- or within-subjects meaning that a factorial design could consist of all between-subject factors, all within-subject factors, or a mix of between- and within-subject factors. We will now learn how to analyze each of these designs.

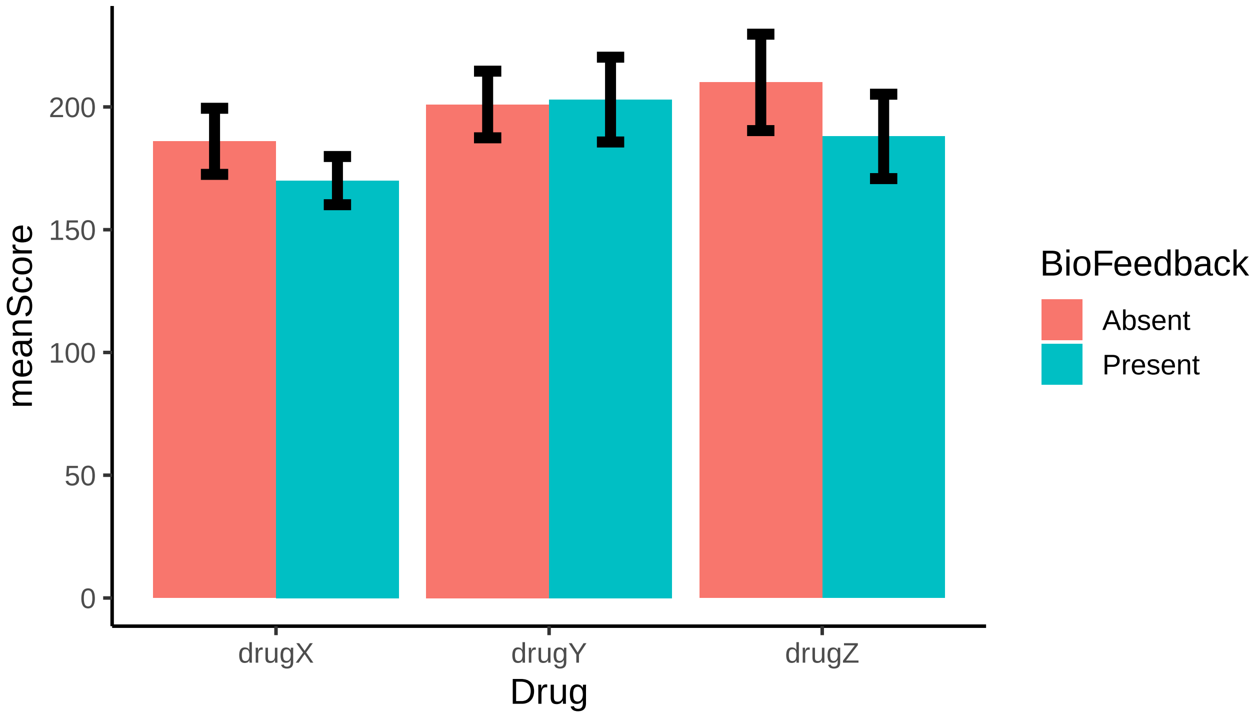
```
# 2 Between-subject factors
drug_bf_data<-read.table(file="Ch7T5.txt",header=TRUE)
str(drug_bf_data)
```

```
## 'data.frame':    30 obs. of  4 variables:
##  $ Score     : int  170 175 165 180 160 186 194 201 215 219 ...
##  $ BioFeedback: Factor w/ 2 levels "Absent","Present": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Drug      : Factor w/ 3 levels "drugX","drugY",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Subj      : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
# Get means for each condition and plot them
drug_bf_means<-drug_bf_data %>% group_by(Drug,BioFeedback) %>%
  summarize(meanScore=mean(Score),SD=sd(Score),nObs=n(),SEM=SD/sqrt(n()))

t_crit<-qt(p=.975,df=4)

drug_bf_means<-drug_bf_means %>% mutate(CI=t_crit*SEM)
print(drug_bf_means)
```

```
## # A tibble: 6 x 7
## # Groups:   Drug [3]
##   Drug  BioFeedback meanScore    SD  nObs   SEM    CI
##   <fct> <fct>           <dbl> <dbl> <int> <dbl> <dbl>
## 1 drugX Absent            186 10.8      5  4.85 13.5
## 2 drugX Present           170  7.91     5  3.54  9.82
## 3 drugY Absent            201 10.9      5  4.89 13.6
## 4 drugY Present           203 13.9      5  6.22 17.3
## 5 drugZ Absent            210 15.8      5  7.07 19.6
## 6 drugZ Present           188 13.8      5  6.19 17.2
```

```
ggplot(drug_bf_means,aes(x=Drug,y=meanScore,fill=BioFeedback)) +
  geom_bar(stat="identity",position="dodge") +
  geom_errorbar(aes(ymin=meanScore-CI,ymax=meanScore+CI),size=2,width=.2,position=position_dodge(.9)) +
  theme_classic(base_size=14)
```



```
# Do ANOVA
drug_bf_aov<-aov(Score~Drug*BioFeedback,data=drug_bf_data)
print(summary(drug_bf_aov))
```

```
##                  Df Sum Sq Mean Sq F value   Pr(>F)    
## Drug              2   3420  1710.0  10.979 0.000411 ***
## BioFeedback       1   1080  1080.0   6.934 0.014563 *  
## Drug:BioFeedback  2    780   390.0   2.504 0.102877    
## Residuals        24   3738   155.8                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Do post-hoc comparisons of marginal means for Drug
TukeyHSD(drug_bf_aov,which="Drug")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = Score ~ Drug * BioFeedback, data = drug_bf_data)
## 
## $Drug
##             diff        lwr      upr     p adj
## drugY-drugX   24  10.062095 37.93791 0.0006958
## drugZ-drugX   21   7.062095 34.93791 0.0026568
## drugZ-drugY   -3 -16.937905 10.93791 0.8537283
```

Here is an exmple of how you would perform an ANOVA for a fully within-subjects factorial design, and for a split-plot design with one between- and one-within-subject factor

```
# Repeated-measures factorial design (2 factors)
ezANOVA(data=my_rm_data,dv=DV,wid=Subject,between=NULL,within=.(IV1,IV2),detailed=TRUE)
# Note period (.) before within arg

# Split-plot design (2 factors - IV1 (between), IV2 (within))
ezANOVA(data=my_sp_data,dv=DV,wid=Subject,between=IV1,within=IV2,detailed=TRUE)
```

Follow up tests for significant main effects and/or interactions can be performed using **pairwise.t.test** or by simply using **t.test** where appropriate. In cases where there is a significant interaction, you may wish to do an ANOVA if your simple main effect analyses involve comparing three or more levels.

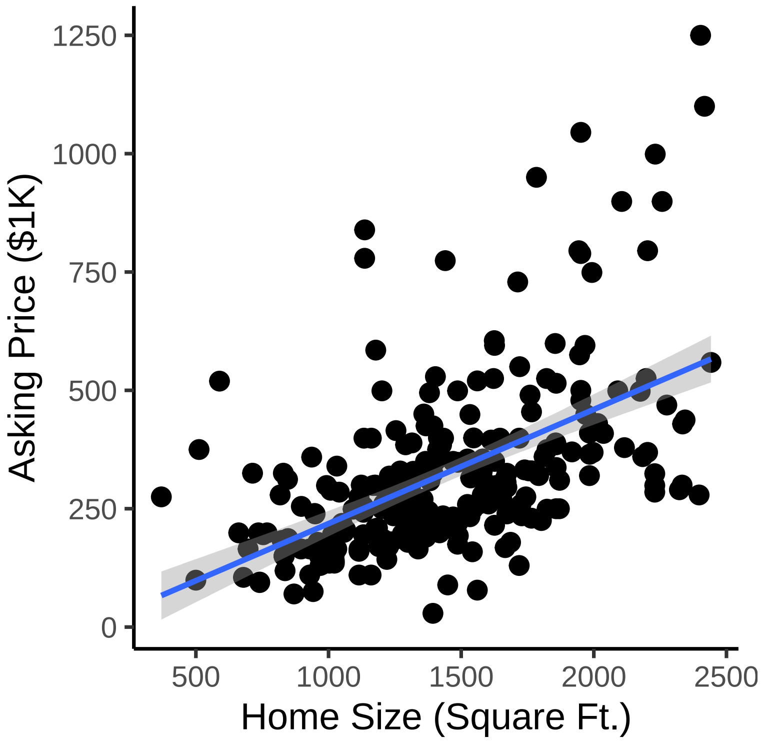
Correlational Designs

You can also use *R* to analyze correlational data. Let's go through an example with the real estate data set from Lab Tutorial 3.

```
listings<-read.table("realestate.txt",header=T)
head(listings,n=10)
```

```
##       mls           location price bedrooms bathrooms     size
## 1  132842      Arroyo Grande 795.0        3         3 2202.730
## 5  136282 Santa Maria-Orcutt 109.9        3         1 1160.358
## 6  136431             Oceano 324.9        3         3 1672.254
## 7  137036 Santa Maria-Orcutt 192.9        4         2 1489.235
## 8  137090 Santa Maria-Orcutt 215.0        3         2 1347.093
## 10 137570         Atascadero 319.0        3         2 1229.107
## 11 138053 Santa Maria-Orcutt 350.0        3         2 1625.803
## 12 138730 Santa Maria-Orcutt 249.0        3         2 1300.642
## 13 139291      Arroyo Grande 299.0        2         2 1167.791
## 14 139427 Santa Maria-Orcutt 235.9        3         2 1300.642
```

```
# Make scatterplot
ggplot(listings,aes(x=size,y=price)) +
  geom_point(size=3) +
  geom_smooth(method="lm") +
  theme_classic(base_size=14) +
  labs(x ="Home Size (Square Ft.)",y="Asking Price ($1K)")
```



```
# Calculate correlation between sq. ft and listing price
with(listings,cor(size,price))
```

```
## [1] 0.5514618
```

```
# Test whether this correlation is signicantly different than zero
with(listings,cor.test(x=size,y=price,alternative="two.sided",method="pearson"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  size and price
## t = 10.618, df = 258, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4607163 0.6307969
## sample estimates:
##       cor
## 0.5514618
```

```
# Fit a linear regression model to the data
listings_lm<-lm(price~size,data=listings)
print(summary(listings_lm))
```

```
##
## Call:
## lm(formula = price ~ size, data = listings)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -284.44  -87.34  -36.08   43.51  693.25
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -22.62048   33.72916  -0.671    0.503
## size          0.24116    0.02271  10.618   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 156.2 on 258 degrees of freedom
## Multiple R-squared:  0.3041, Adjusted R-squared:  0.3014
## F-statistic: 112.7 on 1 and 258 DF,  p-value: < 2.2e-16
```

Dichotomous Outcome Data - Binomial Tests

Although not especially common in experimental psychology, there certainly are cases where the primary DV is a binary outcome (e.g., coin toss), and the question of interest is whether the proportion of times a particular event occured is larger (or less than) chance. In such cases, a binomial test is usually appropriate.

```r
number_of_trials<-50
number_of_successes<-35 # number of heads

binom.test(x=number_of_successes,n=number_of_trials,p=0.5,alternative="greater")
```

```
##
##  Exact binomial test
##
## data:  number_of_successes and number_of_trials
## number of successes = 35, number of trials = 50, p-value = 0.0033
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.576267 1.000000
## sample estimates:
## probability of success
##                    0.7
```