Homework 08: Comparing Multiple Independent Groups (Anova) (ANSWERS)

partner names

add date

Part A

General Questions

1. **QUESTION:** Describe the purpose of a one-way between-subjects ANOVA.

ANSWER: The one-way between-subjects ANOVA is designed to compare more than 2 independent groups on a dependent measure that is either interval or ratio in scale.

2. **QUESTION:** Identifying which *t*-test uses a pooled estimate of variance. Explain why it does.

ANSWER: The independent-samples *t*-test uses a pooled estimate of variance. The variance of the two groups, which are assumed to be equal under H0 is a better estimate of population variance than small h

3. **QUESTION:** Identify one test that can be used for testing for homoegeneity of variance.

ANSWER: Levene test

3. **QUESTION:** Identify one test that can be used for testing for normality of a distribution.

ANSWER: Shapiro-Wilk

Part B

Before you begin

This homework exercise involves having you answer some questions, write some code, and create a nice HTML file with your results. When asked different questions, simply either type your coded or written responses after the ANSWER message. When asked to write code to complete sections, type your code in the empty code blocks that follow the ANSWER message (between the back ticks). After adding that code, you must make sure that it will execute. So remember to read the content and run each line of your code as you write it so that you know it executes correctly. If your code does not execute, then RMarkdown won't know what you are telling it to do and your HTML file will not be produced. Also, don't create your HTML file until you finish and know that all of your code works correctly.

1.0. Installing and using libraries in RStudio

1.1. Use the RStudio interface to install packages/libraries. Go to the Tools option and select Install Packages. Type the package name(s) correctly using the proper letter casing. Also, make sure that you *check the box to Install Dependencies*. Do not install with code.

Install the package: plyr

1.2. Key functions used for this assignment, some old, some new:

- anova() for viewing variance estimates and F-value for ANOVA; built-in stats library
- aov() for fitting an ANOVA model to data; built-in stats library
- by() for applying a function to a data frame split out by levels of a factor; built-in base library
- bartlett.test() for viewing homogeneity of variance; built-in stats library
- count() for viewing the frequency of each factor; plyr library
- densityplot() for creating density plot graphs; lattice library
- describeBy() for descriptive stats for groups; psych library
- histogram() for histograms; lattice library
- leveneTest() for viewing homogeneity of variance; car library
- pairwise.t.test() for examining pairwise comparisons with Bonferroni correction; built-in stats library
- shapiro.test() for testing normality; built-in stats library
- summary.lm() for extracting r-squared; built-in stats library
- TukeyHSD() for analyzing the differences between groups; built-in stats library

2.0. Loading libraries

If you know what libraries you will use for your code, you can load them now. Use the library() function to load the following libraries: lattice, plyr, psych, and car. A summary of the functions and the libraries is listed above.

ANSWER:

```
library(plyr)
library(car)
library(lattice)
library(psych)

#options(scipen = 999) #remove scientific notation if you want
```

Part C

1.0. Overview of a between-groups ANOVA

1.1. The statistical procedure for testing variation among the means of more than two groups is called the *analysis of variance*, abbreviated as *ANOVA*. The null hypothesis in an analysis of variance is that the several populations being compared all have the same mean. Hypothesis testing in analysis of variance is about whether the means of the samples differ more than you would expect if the null hypothesis were true. This question about means is answered, surprisingly, by analyzing variances (hence the name analysis of variance). Among other reasons, you focus on variances because when you want to know how several means differ, you are asking about the variation among those means

ANOVA is a commonly used statistical technique for investigating data by comparing more than two sample means. When there are multiple levels of one IV and the levels are based on independent groups (e.g., different demographics, random assignment, etc.), *one-way between-subjects ANOVA* is used. This is an extension of *independent-samples t-test* for instances where comparisons are made between more than two groups. There is also a *one-way within-subjects ANOVA* which corresponds to multiple repeated measures conditions for which there is one IV and the levels represent measurements of individuals more than once (e.g., measuring alterness in the morning, afternoon, and evening). This exercise is on the one-way between-subjects ANOVA only.

1.2. Data for the *one-way between-subjects ANOVA* are grouped on some classification factor, or variable (e.g., class rank) so that group means can be created based on that classification.

For example, a data frame may look like this:

- ID ClassRank Happiness
- 1 Freshman 6
- 2 Freshman 5
- 3 Sophomore 6
- 4 Sophomore 7
- 5 Junior 8
- 6 Junior 8
- 7 Senior 9
- 8 Senior 10

Using some real data, we can read in the class survey data frame. Based on some questions in the survey, we will analyze different DVs by comparing levels of different IVs.

Read in the Survey data set and assign the contents to a data frame called SURVEY:

```
#setwd("c:/users/gcook/desktop/Psyc109") #set working directory if necessary
SURVEY <- read.csv("SurveyNames.csv")

# There is a misspelling of a variable named Excercise. Let's rename it correctly.
names(SURVEY)[names(SURVEY) == 'Excercise'] <- 'Exercise'
```

2.0. Examining Data

2.1. As always, examine the str() of the data frame to see the classification of variables in the SURVEY data frame.

```
str(SURVEY)   #or View(SURVEY)
```

```
## 'data.frame':    54 obs. of  48 variables:
##  $ X          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ ID         : int  4 5 6 7 8 9 10 11 12 13 ...
##  $ Gender     : int  0 1 1 1 1 1 0 0 0 1 ...
##  $ PolParty   : int  0 0 1 0 2 2 3 3 4 1 ...
##  $ NonFiction : int  1 1 1 0 1 0 1 1 1 0 ...
##  $ Religion   : int  7 1 7 1 1 6 1 5 1 1 ...
##  $ Dress      : int  3 2 1 1 1 2 2 2 2 1 ...
##  $ Superpower : int  5 2 2 3 4 5 3 3 3 3 ...
##  $ Fight      : int  2 7 1 1 3 6 1 4 4 5 ...
##  $ GlobalIssue: int  7 6 3 2 7 1 2 2 2 2 ...
##  $ MovGenre   : int  6 3 1 2 4 1 1 1 5 2 ...
##  $ Music      : int  3 3 3 3 4 3 6 3 3 3 ...
##  $ Electronic : int  1 1 2 1 1 2 1 1 2 1 ...
##  $ Sport      : int  2 7 5 2 1 4 1 6 6 6 ...
##  $ FavTime    : int  2 1 4 4 2 4 4 1 4 3 ...
##  $ Hand       : int  3 4 4 4 4 4 4 4 4 4 ...
##  $ Quad       : int  3 3 3 3 2 3 3 2 3 3 ...
##  $ MusGenre   : int  4 7 7 2 4 4 6 5 5 4 ...
##  $ Voting     : int  3 1 2 2 1 1 3 2 2 2 ...
##  $ Partners   : int  2 2 2 1 2 2 2 3 2 1 ...
##  $ YearBorn   : int  5 4 6 6 5 5 5 6 6 7 ...
##  $ Tip        : int  15 10 10 10 15 15 20 15 15 20 ...
##  $ SocialMedia: num  3 10 2 3 2 1 2 0.5 3 1 ...
##  $ Siblings   : int  1 1 3 3 3 0 3 1 2 1 ...
##  $ MovMonth   : int  1 5 1 2 7 2 0 2 2 4 ...
##  $ Potter     : int  0 2 0 0 0 3 0 3 3 1 ...
##  $ MilesHome  : Factor w/ 41 levels "0","1","1,000",..: 12 13 20 17 40 15 14 5 20 38
...
##  $ Exercise   : num  0 12 4 6 20 10 6 20 20 20 ...
##  $ HourSleep  : num  6 7 5 5 6 7 6 7 5 7 ...
##  $ HourPhone  : num  2 100 24 5 21 0 4 1 2 1 ...
##  $ Commuting  : num  0 2 1 0.5 0 0 0 0 0 1 ...
##  $ Parents    : num  3 20 3 7 3 0.5 5 4 0 1 ...
##  $ SportsPlay : int  6 5 1 6 5 4 6 7 8 2 ...
##  $ GELeft     : int  2 6 2 1 3 3 2 4 2 0 ...
##  $ Alone      : int  2 10 4 3 2 0 0 10 0 50 ...
##  $ Countries  : int  0 4 1 10 2 2 0 7 8 9 ...
##  $ Dogs       : int  2 0 5 0 0 0 6 1 1 0 ...
##  $ Snack      : int  3 1 2 1 2 2 3 20 2 5 ...
##  $ OffCampus  : int  2 3 2 2 2 2 2 3 2 3 ...
##  $ Sleep7     : int  1 5 3 2 4 4 5 5 1 5 ...
##  $ Reusable   : int  5 5 4 3 2 5 3 5 4 5 ...
##  $ Recycle    : int  4 3 2 3 2 3 1 4 1 5 ...
##  $ JobSec     : int  3 3 2 2 2 4 1 1 5 1 ...
##  $ Enjoyment  : int  3 4 2 4 4 3 5 4 4 4 ...
##  $ Present    : int  5 4 5 4 4 4 5 5 3 4 ...
##  $ FamPhone   : int  4 5 4 5 4 3 4 4 3 4 ...
##  $ GoodNight  : int  3 5 3 3 4 4 4 4 1 5 ...
##  $ Q44        : int  1 1 1 1 1 1 1 1 1 1 ...
```

You will notice that many variables are listed as int, or integers but they should not really reflect numerical values. The integers are just placeholders for different groups (e.g., men, women). For example, check out Gender, PolyParty, MusGenre, FavTime, etc. R will read in a data file in a way that it thinks is appropriate. However, sometimes you will need to change the scaling of the variable so that you can perform certain test. Because these variables refer to categories/nominal variables, they are also known as factors. We need to modify them for the ANOVA test. Using the factor() function we will convert them to factors because they are certainly not integers.

Knowing the sample size of each of the groups is important for many reasons. For instance, you wouldn't want to run an ANOVA when you have really small sample sizes. You can count these by hand of course, but it' much easier to create a frequency table of the variable using the count() function from the plyr library so that you know the sample size for each level of a variable.

Gender…

```
# get a frequency count to see how many people are in each
count(SURVEY$Gender)
```

```
##   x freq
## 1 0   25
## 2 1   29
```

Values are either 0 (men) or 1 (women).

2.2. Now that we the levels and how many people in the different groups, we can create some variables that do not include very small samples. Let's create the factors using the factor() function. If we add two arguments to the function (e.g., *levels* and *labels*) we can replace the numbers with names that correspond to the levels of the IV. We will add ".fact" to the new variable so that we know these are our new factors. Let's practice with a simple two-level example. Remember that in order to add the variable to your existing data frame, you have to specify the data frame too.

Ex. : dataframename$newvariable <- factor(oldvariable, levels, labels)

```
SURVEY$Gender.fact <- factor(SURVEY$Gender,
                        levels = c(0, 1),
                        labels = c("Male", "Female"))
```

Political party preference…

```
# get a frequency count to see how many people are in each category
count(SURVEY$PolParty) # too few for an ANOVA with 3 groups
```

```
##   x freq
## 1 0   15
## 2 1   25
## 3 2    6
## 4 3    5
## 5 4    3
```

```
SURVEY$PolParty.fact <- factor(SURVEY$PolParty,
                               levels = c(0, 1, 2, 3, 4, 5),
                               labels = c("No Affiliation", "Democrat", "Independent",
                                          "Libertarian", "Republican", "Green"))
```

Music Genre…

```
# get a count
count(SURVEY$MusGenre)
```

```
##   x freq
## 1 1    1
## 2 2    1
## 3 3   14
## 4 4   17
## 5 5   10
## 6 6    4
## 7 7    7
```

```
# create a new factor with 3 groups with samples sized of over 10 each (we don't want really small samples, so we drop them out; they will become NA)
SURVEY$MusGenre.fact <- factor(SURVEY$MusGenre,
                            levels = c(3, 4, 5),
                            labels = c("Alternative", "Hip-Hop Rap", "Electronic"))

count(SURVEY$MusGenre.fact)
```

```
##             x freq
## 1 Alternative   14
## 2 Hip-Hop Rap   17
## 3  Electronic   10
## 4        <NA>   13
```

Favorite time of the day…

```
count(SURVEY$FavTime)
```

```
##   x freq
## 1 1   15
## 2 2   10
## 3 3   15
## 4 4   14
```

```
SURVEY$FavTime.fact <- factor(SURVEY$FavTime,
                         levels = c(1, 2, 3, 4),
                         labels = c("Morning", "Afternoon", "Night", "Late Night"))

count(SURVEY$FavTime.fact)
```

```
##            x freq
## 1    Morning   15
## 2  Afternoon   10
## 3      Night   15
## 4 Late Night   14
```

OK, now that you have some IVs with samples that aren't too small and you know what the levels of those IVs are, sized samples, proceed to comparing groups with an ANOVA.

3.0. Examing Assumptions of the between-subjects ANOVA

3.1. Assumptions of a one-way between-subjects ANOVA are the same as for a independent-samples *t*-test except that they apply to two or more groups, not just two groups.

1. DV is interval/ratio in measurement scale.
2. The populations have the same variance; homogeneity of variance.
3. The populations are distributed normally.
4. Each value is sampled independently from each other value. This assumption requires that each subject provide only one value. If an experimental unit provides two scores, then the values are not independent.

3.2. Testing the Assumptions

3.2.1. Looking at your data is always good before doing statistical testing. Examine the descriptive statisics for the groups in order to determine what the sample means and variances are. One really useful way to do this is to use the describeBy() function from the psych library. This function takes two main arguments: the DV and the factor IV. Examine amount of exercise for people who have different music genre preferences. We will describe the DV by the IV in order to obtain a bunch of statistics for the levels of MusGenre.fact.

- describeBY(DV, IV)

```
describeBy(SURVEY$Exercise, SURVEY$MusGenre.fact)
```

```
## group: Alternative
##   vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## 1    1 14 9.79 6.62      7    9.58 6.67   2  20    18 0.34    -1.62 1.77
## --------------------------------------------------------
## group: Hip-Hop Rap
##   vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## 1    1 17 8.47 7.75      6    8.27 5.93   0  20    20 0.46    -1.53 1.88
## --------------------------------------------------------
## group: Electronic
##   vars  n mean   sd median trimmed  mad min max range skew kurtosis   se
## 1    1 10 9.45 6.93    7.5    9.12 6.67 1.5  20  18.5 0.43    -1.56 2.19
```

You can see that there are 3 music groups and because we added labels, the labels are also included. If we didn’t add the labels above, the output would be more difficult to interpret, so always create labels first. The sample sizes are no greater than 17, the means are slightly different but around 8 and 9, and the standard deviations also differ slightly but not by much. All distributions also have a slight positive skew which is evidenced by the skew measure as well as the fact that the means are higher than the medians. In order to compare means appropriately for the ANOVA, you need to check assumptions.

3.2.2. Normality can be examined visually of course with a histogram or density plot. Remember that the lattice library makes creating graphs for levels of a factor very easy to do. Use those from the lattice library. The density plot provides more detail about shape. The first argument of the function tells you what to plot as a function of something else. If that argument is not used, frequencies/probabilities will be plotted, which is what we want here. Thus, we plot:

- densityplot(~ DV) # probabilities as a function of the DV

The | tells R to plot the DV separately for each level of the IV.

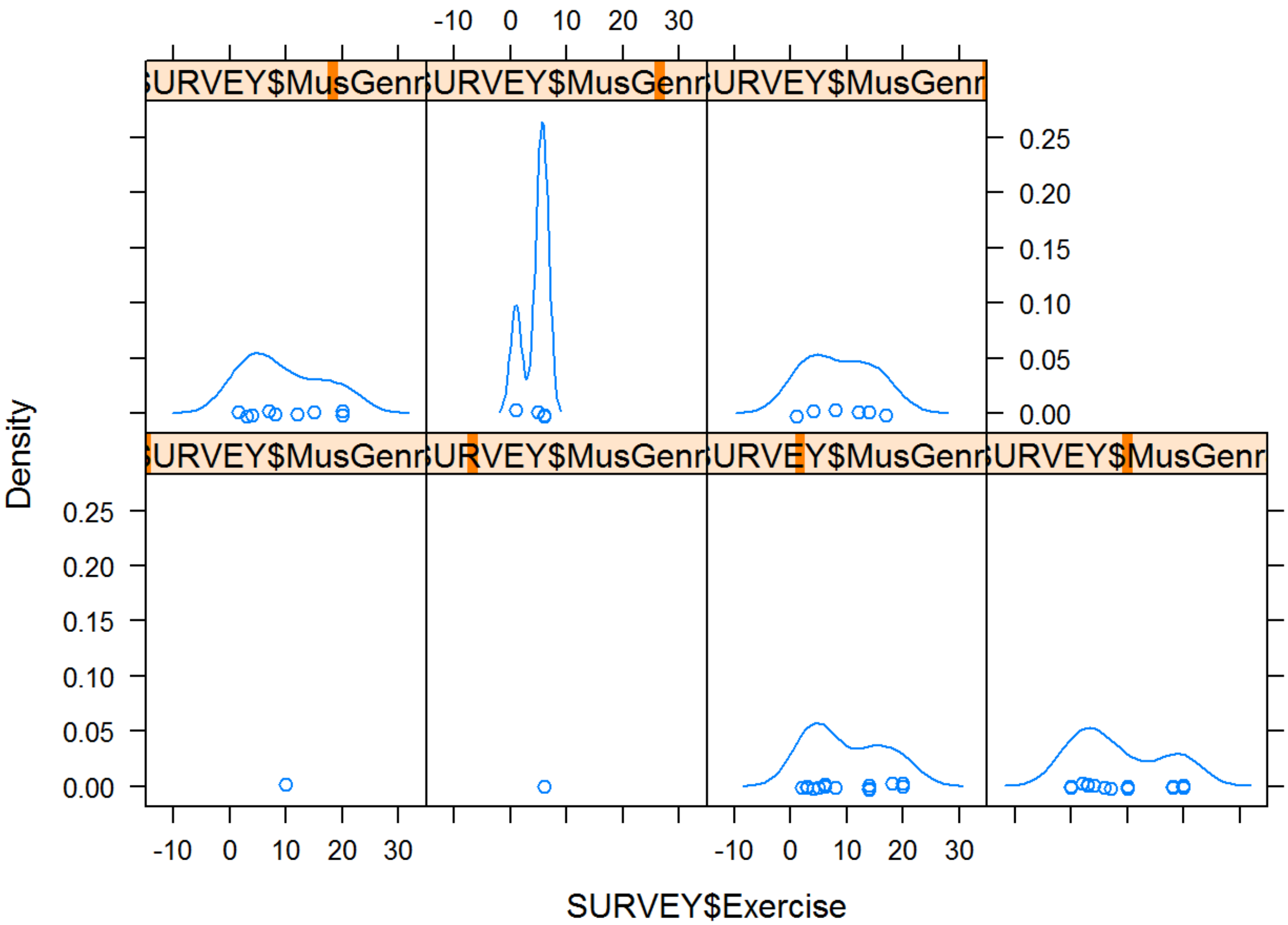
- densityplot(~ DV | IV)

To illustrate the importance of naming the *labels* for the *levels* of your IV, compare the original MusGenre vector to MusGenre.fact which is based on groups with the largest sample sizes so there will be fewer plots. However, the labelling makes reading the graphs much more easy to interpret. That’s another reason why making your categorical variables into factors matters.

```
is.factor(SURVEY$MusGenre) # see, it is NOT a factor.
```

```
## [1] FALSE
```

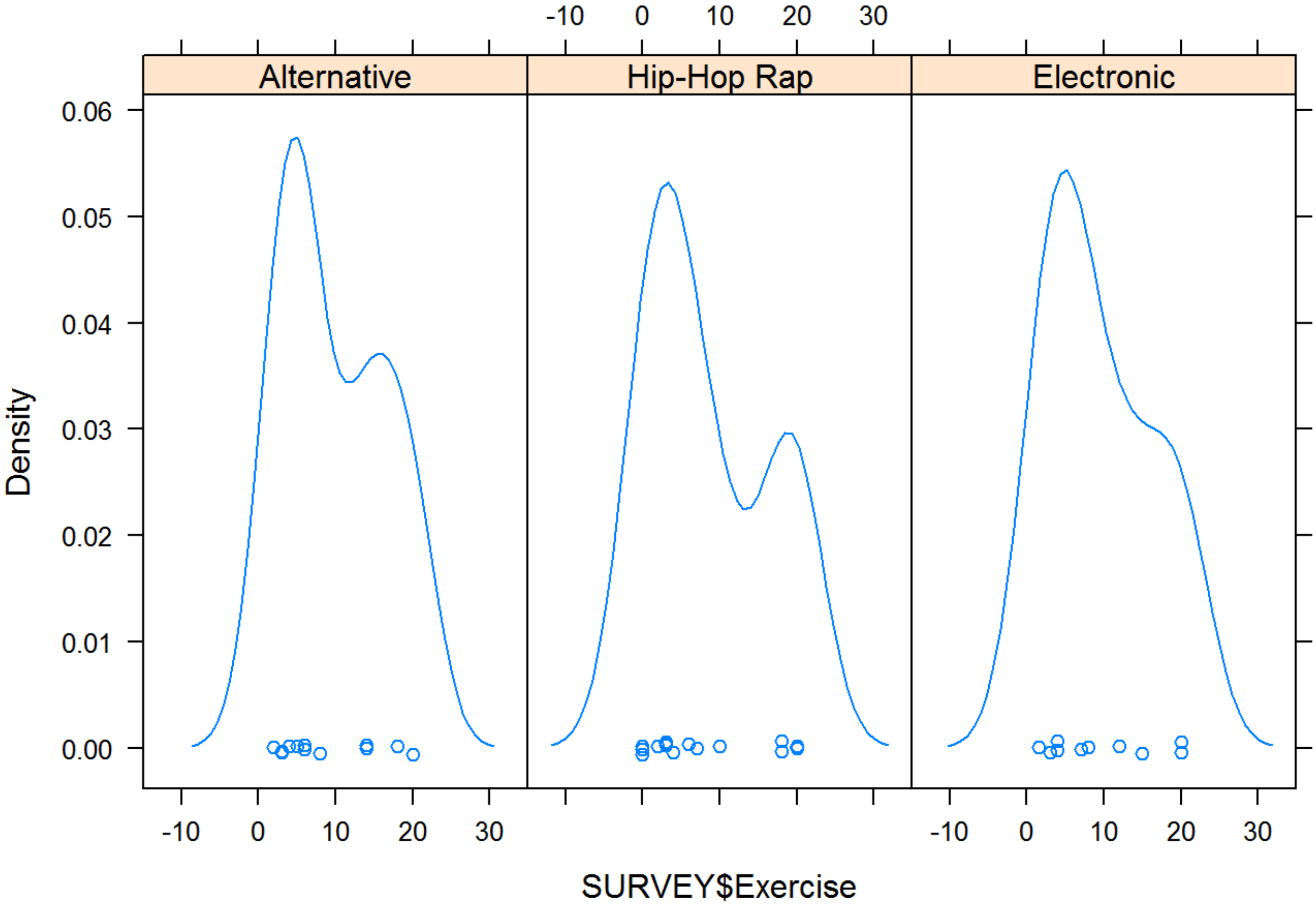
```
densityplot(~ SURVEY$Exercise | SURVEY$MusGenre)  # notice there are no level names.
```



```
is.factor(SURVEY$MusGenre.fact) # see, it IS a factor.
```

```
## [1] TRUE
```

```
densityplot(~ SURVEY$Exercise | SURVEY$MusGenre.fact)  # easy to see level names.
```

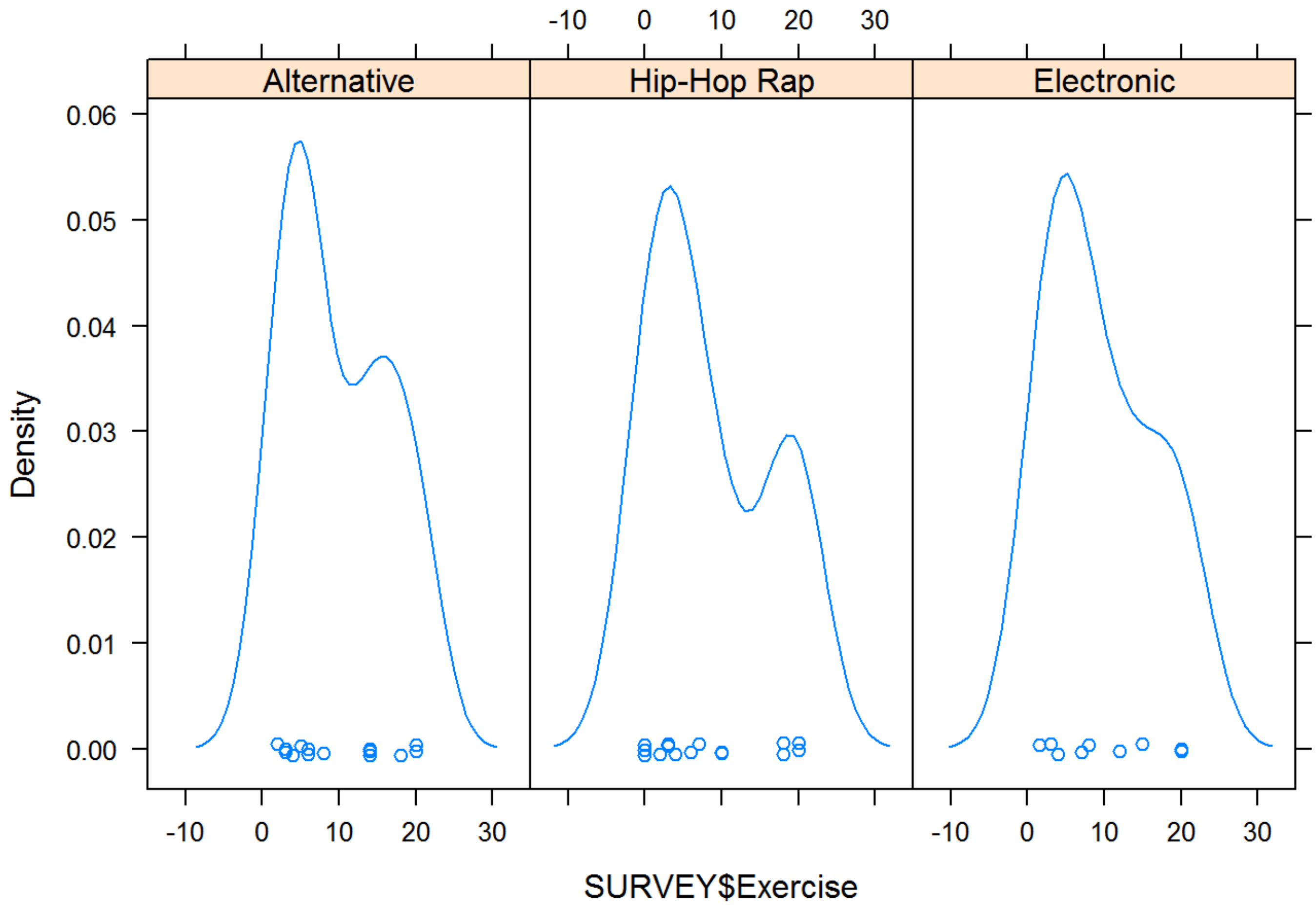


Adding the *layout* argument to the densityplot() function allows you to plot the densities on top of each other which is useful for comparing the means of the distrubutions.

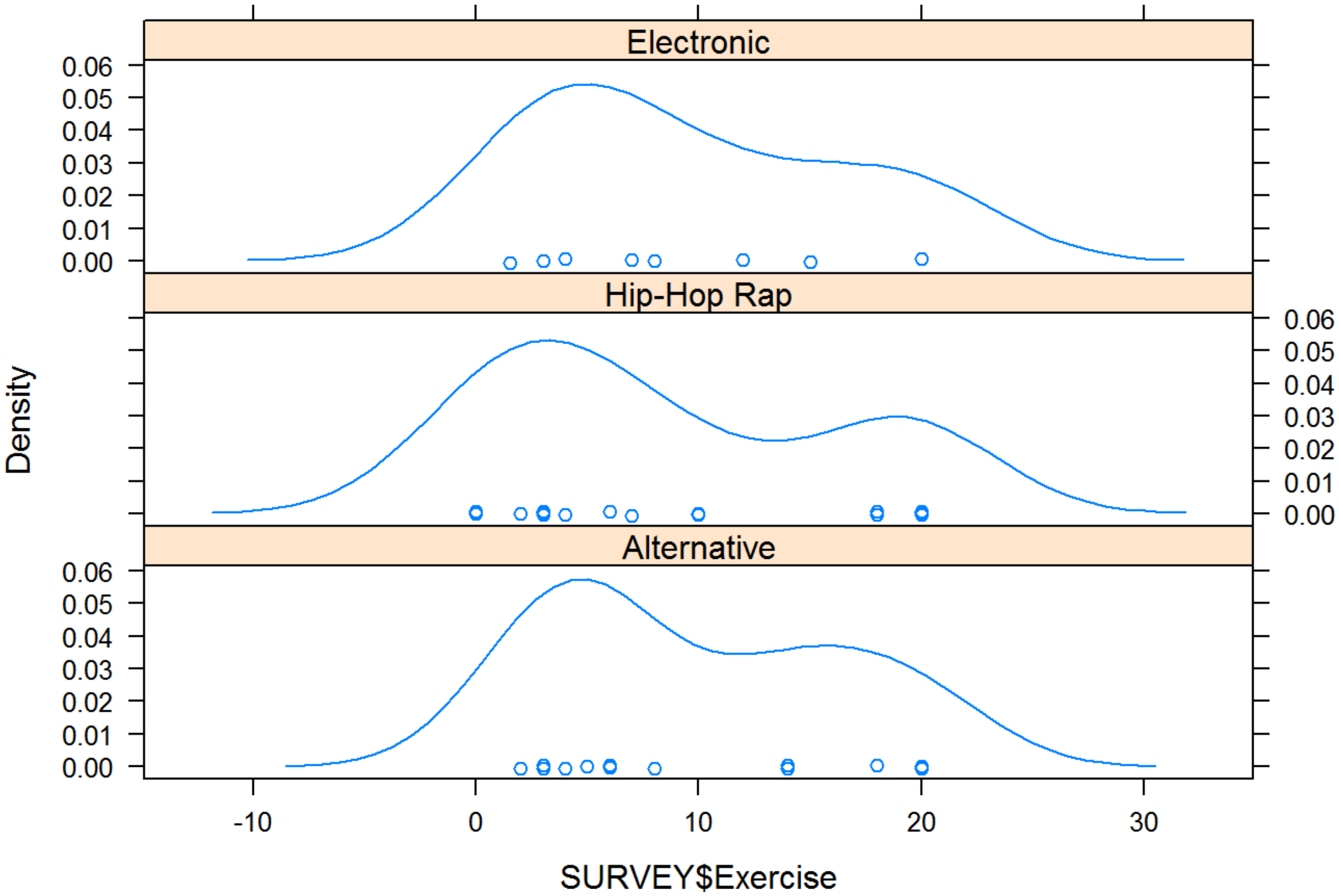
- densityplot(~ DV | IV, layout = c(cols, rows))

Layout typically relies on at least 2 elements that need to be combined with the combine or concatenate function, c(). The first is the number of COLUMNS, the second is the number of ROWS (unfortunately, this is the reverse of data frames, which are structured as ROWS, COLUMNS). There is a third element if you wanted to create the plots by not grouping them together, but that's not necessary to discuss here. If you are interested in graphing, R provides many options for you. Let's take a look at the plots.

```
densityplot(~ SURVEY$Exercise | SURVEY$MusGenre.fact, layout=c(3,1)) # 3 columns, 1 row =
too squished.
```

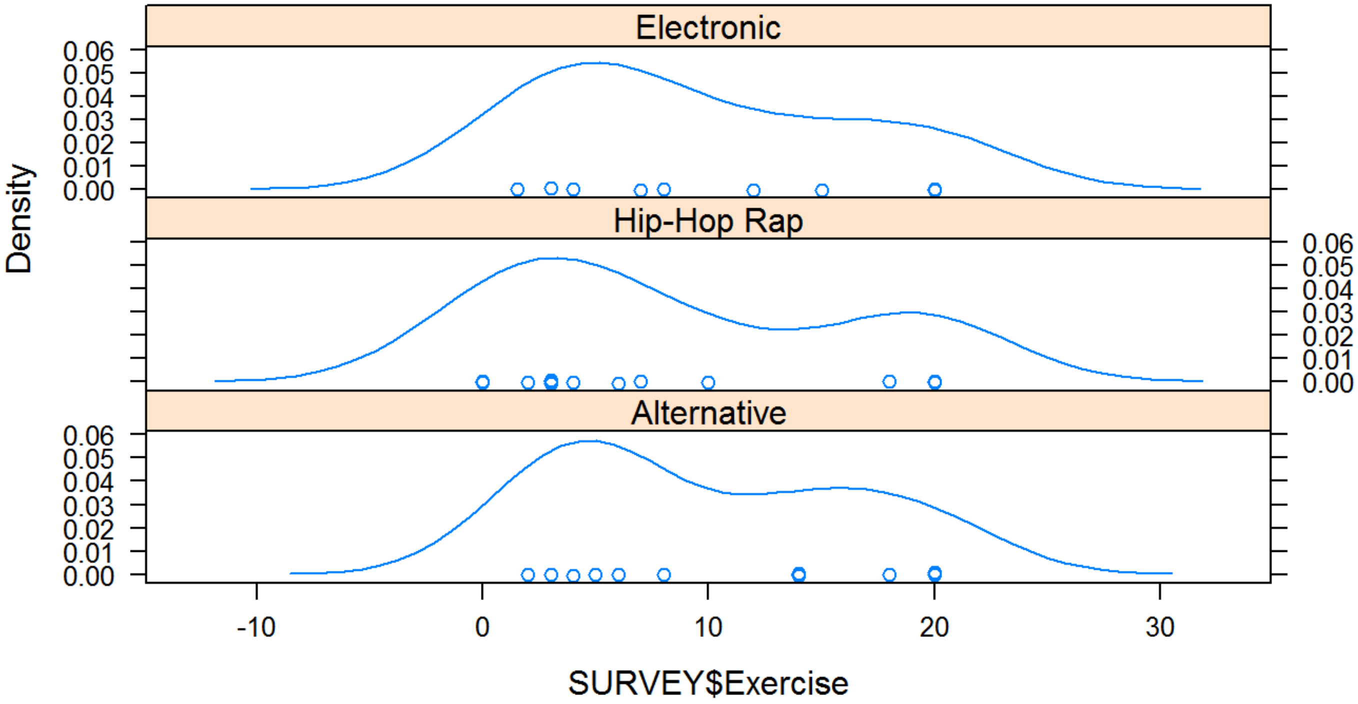


```
densityplot(~ SURVEY$Exercise | SURVEY$MusGenre.fact, layout=c(1,3)) # 1 column, 3 rows =
Much better.
```

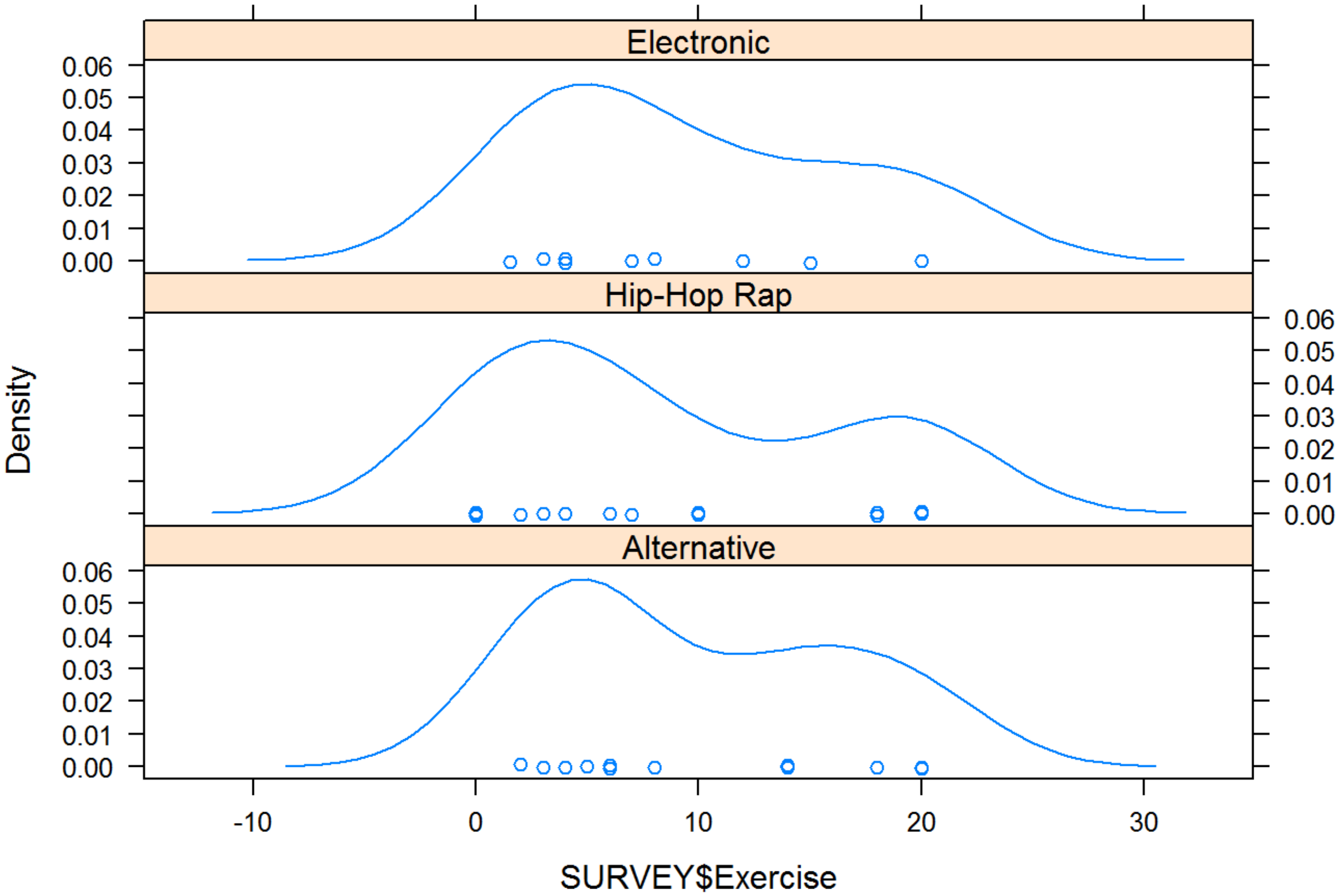


If you have 3 plots and you try to plot them by not specifying the numbers correctly, R will create an empty graph. For example, if you try to plot 4 rows instead of 3, you will get a graph that has extra white space at the top. If you don't remember how many plots there are, you can have the levels of the factor counted for you automatically using the nlevels() function. This involves an extra step, but when you are creating graphs for different factors, it also removes trying to remember which factors have different numbers of levels.

```
# wrong number of rows
densityplot(~ SURVEY$Exercise | SURVEY$MusGenre.fact, layout=c(1,4))
```



```
# using nlevels()
densityplot(~ SURVEY$Exercise | SURVEY$MusGenre.fact, layout=c(1,nlevels(SURVEY$MusGenre.
fact)))
```



Besides looking at the data visually and guessing about normality, the shapiro.test() will test for normality. However, the function does not have a built-in way to test normality for the subgroups or levels of a factor. However, the by() function will serve as a *helper function* (it helps you perform computations for another function) to allow you to conduct the test at each level of your factor. Because the levels have labels, the output will include their names; another reason why adding level labels is helpful.

- by(DV, IV, shapiro.test)

```
by(SURVEY$Exercise, SURVEY$MusGenre.fact, shapiro.test)
```

```
## SURVEY$MusGenre.fact: Alternative
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.87422, p-value = 0.04812
##
## --------------------------------------------------------
## SURVEY$MusGenre.fact: Hip-Hop Rap
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.84093, p-value = 0.007822
##
## --------------------------------------------------------
## SURVEY$MusGenre.fact: Electronic
##
##  Shapiro-Wilk normality test
##
## data:  dd[x, ]
## W = 0.88757, p-value = 0.1592
```

```
# to show you more of how the by() function works to repeat tasks, do the same for mean
by(SURVEY$Exercise, SURVEY$MusGenre.fact, mean)
```

```
## SURVEY$MusGenre.fact: Alternative
## [1] 9.785714
## --------------------------------------------------------
## SURVEY$MusGenre.fact: Hip-Hop Rap
## [1] 8.470588
## --------------------------------------------------------
## SURVEY$MusGenre.fact: Electronic
## [1] 9.45
```

Having non-normal distributions is almost preordained with small sample sizes. When you have sample sizes of about 30 or more and you still have normality issues, you may have to transform your data before conducting an ANOVA test. Or you might night be able to even do an ANOVA. Given the shapes of the distributions just examined (some being non-normal), you will consider whether the ANOVA is appropriate as a test.

Although it appears that we have violated the assumption of normality for the Exercise data for MusGenre.fact, we will continue with our analysis for illustration purposes only.

3.2.3. Homogeneity of Variance

For an *ANOVA*, one assumption is the *homogeneity of variance* (HOV) assumption. That is, in an *ANOVA* we assume that variances of the groups are equal. Much like with the *t*-test, moderate deviations from equal variances do not seriously affect the results of the *ANOVA*. In other words, the *ANOVA* is robust to small deviations from the *HOV* assumption. We only need to be concerned about large deviations from this assumption. There are several ways to test this assumption; two of those are the Levene's test and the Bartlett test. The Bartlett test is a common test for the homogeneity of variances when the data are distributed normally.

If your distributions are normal, you can use the Bartlett's test for violations in the homogeneity of variance using the bartlett.test() function. This test uses two main arguments and follows in formula format, the DV as a function of the IV:

- bartlett.test(DV ~ IV)

```
# Bartlett test
bartlett.test(SURVEY$Exercise ~ SURVEY$MusGenre.fact)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  SURVEY$Exercise by SURVEY$MusGenre.fact
## Bartlett's K-squared = 0.36332, df = 2, p-value = 0.8339
```

If you examine the *p*-value, you will see that it is larger than an alpha = .05, so there appears to homogeneity of variance. However, because some of our levels of the IV has a very small sample size and because there was some positive skew to the distributions (see earlier description of the descriptive statistics), the Levene's Test would be more appropriate than the Bartlett test because it is not as sensitive to departures from normality as is the Bartlett's test. The Levene's test takes two arguments, the DV and the factor IV, but is not in the form of a formula:

- leveneTest(DV, IV)

```
leveneTest(SURVEY$Tip, SURVEY$MusGenre.fact)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  0.6566 0.5244
##       38
```

If you notice from the title of the output, the default version of the Levene test is based on "medians" which is supposedly more robust and accurate than that based on "means". If you wanted to use means for the variance measure, you can simply use the *center* argument and set it equal to "mean". If the mean version makes more sense to you, use that one, but remember to use "center = mean" because by default "center = median".

```
leveneTest(SURVEY$Tip, SURVEY$MusGenre.fact, center = mean)
```

```
## Levene's Test for Homogeneity of Variance (center = mean)
##       Df F value Pr(>F)
## group  2  0.7532 0.4777
##       38
```

Because the Levene’s test is comparing the 3 groups, the test provides an *F*-value which ironically is an ANOVA value. One way to think about the Levene’s test when there are more than 2 groups is that it’s an ANOVA test on the variances rather than the means. The *p*-value can be used for interpretation. Both versions of the test reveal that the variances are not different (e.g., *p* > alpha), suggesting that we likely have not violated the homogeneity of variance assumption.

We can write the Levene’s test result as:

- *F*(2, 38) = .7532, *p* > .05.

4.0. Conducting the ANOVA test

4.1. The ANOVA stands for *Analysis of Variance*, which is actually a test of the ratio of variance between groups (e.g., between-groups variability = sample means deviated from a grand mean) to variance within groups (e.g., within-groups variability = how people within samples deviate from their respective sample means). In order words, when an IV does not explain the data, the differences between the sample means will be about the sames as the variability within the sample means. When the difference between groups is larger than the difference between people in groups, the *F* value will be larger and has a greater likelihood of suggesting that the sample means differ from one another.

The ANOVA test provides an *F* ratio:

- *F* = between-groups variance / within-groups variance

And because you know that the unbiased variance = SS/df…

- *F* = between groups SS/df / within groups SS/df

4.2. In order to obtain the *F* ratio value, SS, degrees of freedom, and the *p*-value, the aov() can be used for fitting ANOVA models for categorical IVs.

The aov() function follows the same format as the lm() function for regression.

- aov(DV ~ IV)

Specify the linear model by setting the DV and the IV and assign the result to an object named EXERCISE.aov. The aov is a useful reminder of the aov() function so you know which test you ran. However, you could name the object anything you wanted.

```
EXERCISE.aov <- aov(Exercise ~ MusGenre.fact, data = SURVEY)

# or if you prefer using the $ approach you can specify SURVEY twice
EXERCISE.aov <- aov(SURVEY$Exercise ~ SURVEY$MusGenre.fact)
```

Then use the anova() function on the model you created in order to examine the ouput.

```
anova(EXERCISE.aov)
```

```
## Analysis of Variance Table
## 
## Response: SURVEY$Exercise
##                      Df Sum Sq Mean Sq F value Pr(>F)
## SURVEY$MusGenre.fact  2   14.4   7.201  0.1394 0.8703
## Residuals            38 1962.8  51.653
```

BONUS: If there is heterogeneity of variance and you want to use a Welch correction for multiple groups, you can use the oneway.test() function for ANOVA. Like the aov() function, you specify the DV as a function of the IV. However, the output does not provide SS, so it's limited in this context. You can read up more on it if interested.

- oneway.test(DV ~ IV)

4.3. The output of the anova() function displays 2 rows (e.g., between-groups and within-groups/residuals information) and 5 columns of values (e.g., degrees of freedom, Sums of Squares, Mean Squares, F-value, and p-value. For illustration purposes only, the description below also describes how the values in the output are used to calculate the F-value.

Degrees of Freedom

- *Between-groups df* (#groups - 1): 3 - 1 = 2
- *Within-groups df* (n - 1):
    1. Alternative: 14 - 1
    2. Hip Hop and Rap: 17 - 1
    3. Electronic: 10 - 1
    - Total: 13 + 16 + 9 = 38

Mean Squares

- Mean squares are simply SS/df (variances). Taken from the output,

```
MSbetween <- 14.4/2
MSwithin <- 1962.8/38
```

F-value and p-value

- And the *F* ratio is simply a ratio of the two MS (variances).

```
Fval <- MSbetween/MSwithin

# examine the values
MSbetween; MSwithin; Fval
```

```
## [1] 7.2
## [1] 51.65263
```

```
## [1] 0.1393927
```

4.4. When reporting the between-subjects ANOVA, you need so specify the F value along with degrees of freedom because there is a family of F values just like there is a family of t values and the calculated test is a test of the data fitting an F distribution of a certain combination of degrees of freedom for the groups and the error:

- *F*(between-groups df, within-groups df) = F, *p*-value.
- *F*(2, 38) = .1394, *p* > .05.

5. More on Understanding the Variance

5.1 The Sum of Squares

ANOVA is a method for testing differences among means by analyzing variance. ANOVA partitions the variability among all the values into one component that is due to variability among group means (due to the treatment) and another component that is due to variability within the groups (also called residual variation). Variability within groups (within the levels of the IV) is quantified as the sum of squares of the differences between each value in a sample and that sample mean.

For Example, in our ANOVA: The between-groups SS, which examine the differences among the group means from a grand mean (mean of all means) is:

- *64.4* with a DF of *4*

and…

The within-groups SS, which examine error variation, or variation of individual scores around each group mean. This error is variation in the scores that is not due to the treatment (or IV) is:

- *437.2* with a DF of *49*.

5.2. The Mean Square

Each SS is associated with a certain number of degrees of freedom (df, computed from number of subjects and number of groups), and the mean square (MS) is computed by dividing the SS by the appropriate number of df. These can be thought of as variances, SS/df. I'm sorry that someone created a new name for them. Just think about the MS as between-groups and within-groups variance estimates.

The MS values obtained can be found in the ANOVA table that we previously ran earlier under the Mean Sq column.

5.3 The *F*-value

The F ratio is the ratio of two variances, or well, MS values. If the null hypothesis is true, you expect F to have a value close to 1.0; the variances are the same and therefore the IV does not explain the data. The larger the F ratio, the greater the variation among group means relative to the error within groups.

In order words the groups differ from one another than are the people within the groups differ from each other.

You'll see a large F ratio both when H0 of equal groups means does not really account for the data (the data are not sampled from populations with the same means) or when random sampling accidentally produced samples with means that are not equal even thoguh they should be (if the H0 did account for the data).

5.4. The P value is determined from the F ratio and the two values for degrees of freedom shown in the ANOVA table. Luckily you do not need to look up any values in a table when running the analysis in R.

6. Post-hoc tests

6.1. If you achieve significance in with your ANOVA you must run a post-hoc test to determine which levles of the factor are significantly different from other another. Post-hoc test correct for familywise error rate. Two methods for post-hoc testing include using the TukeyHSD() and the pairwise.t.test() functions.

6.2. TukeyHSD (Tukey Honest Significant Differences) is a commonly used test to show which levels are significantly different from othe another. You simply put the ANOVA model in the function.

```
TukeyHSD(EXERCISE.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = SURVEY$Exercise ~ SURVEY$MusGenre.fact)
##
## $`SURVEY$MusGenre.fact`
##                               diff       lwr      upr     p adj
## Hip-Hop Rap-Alternative -1.3151261 -7.641012 5.010760 0.8683805
## Electronic-Alternative  -0.3357143 -7.592939 6.921511 0.9930087
## Electronic-Hip-Hop Rap   0.9794118 -6.005910 7.964733 0.9376860
```

As you can see this compares each of the groups to each other and shows you the differences as well as the *p*value for each of the pairwise tests. In this case, there are no significant differences between Exercise habits for individuals who prefer different Genres of music.

6.3. Unlike, TukeyHSD, the pairwise.t.test() function requires you to specify the DV and the IV and specify the adjustment of the p value. This pairwise comparison adjusts the *p*-value due to the familywise error rate. A common correction is *Bonferroni*.

```
pairwise.t.test(SURVEY$Exercise, SURVEY$MusGenre.fact, p.adj = "bonf")
```

```
## 
##  Pairwise comparisons using t tests with pooled SD 
## 
## data:  SURVEY$Exercise and SURVEY$MusGenre.fact 
## 
##             Alternative Hip-Hop Rap
## Hip-Hop Rap 1           -          
## Electronic  1           1          
## 
## P value adjustment method: bonferroni
```

Notice that the *p*-values are all 1, or > .05.

7. Effect Size

calculate an effect size after conducting an appropriate statistical test for significance. This post will look at effect size with ANOVA, which is not the same as other tests (like a *t*-test). When using effect size with ANOVA, we use a measure of *r*-squared, which reflects the ratio of variability that the model explains out of all the variability that exists. This meausure is sometimes also referred to as *Eta squared*.

- *r*-squared is SS model / SS total. For our example we simply divide:

```
r2 <- 14.4 / (14.4 + 1962.8)
round(r2, 4) # if you don't like all the decimals
```

```
## [1] 0.0073
```

Or, because *r*-squared is based on the linear model, use summary.lm() and extract the r.squared value.

```
r2 <- summary.lm(EXERCISE.aov)$r.squared
round(r2, 4)
```

```
## [1] 0.0073
```

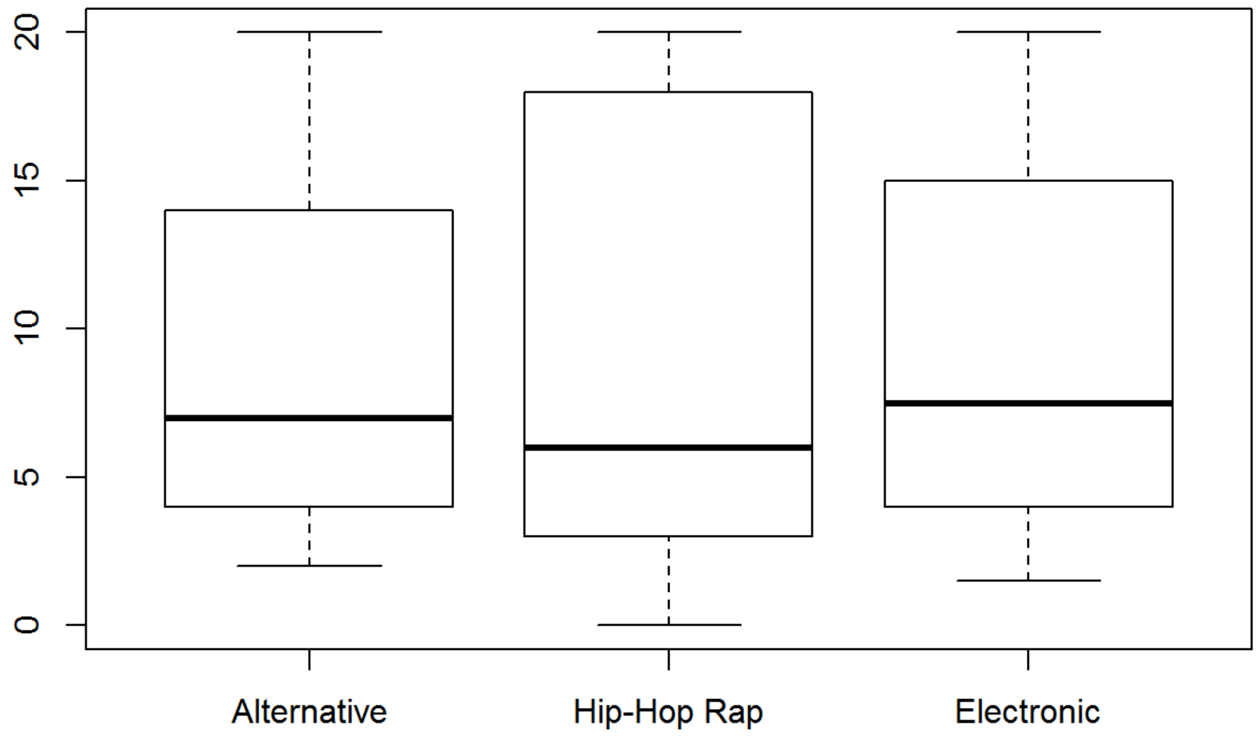
In this example, there was no difference between Genre groups and how much they exercise, so having a small effect size makes a lot of sense.

8.0. Graphing

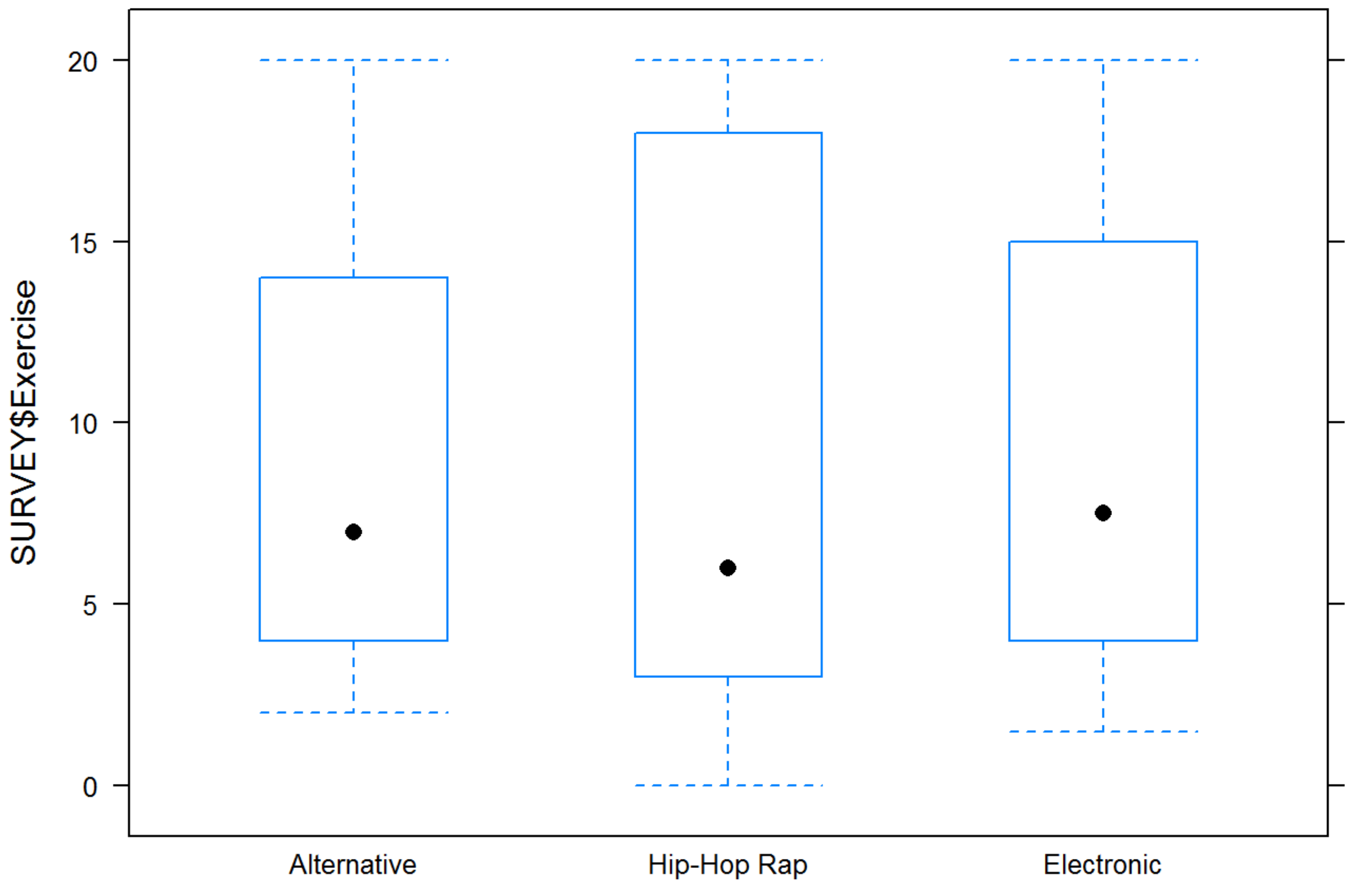
If you want to create a graph of the conditions, one way to do so is to create a simple box-and-whisker plot for the levels of the factor by using the lattice package. Plot the DV as a function of the IV. Box-and-whisker plots use the bwplot() function. They are nice because they convey a lot of information. A good summary and illustration is at http://flowingdata.com/2008/02/15/how-to-read-and-use-a-box-and-whisker-plot/ (http://flowingdata.com/2008/02/15/how-to-read-and-use-a-box-and-whisker-plot/)

- The point on the box represents the median, not the mean. However, when distributions are not skewed the means and medians are the same. When they are not, the median is more informative anyway. Remember the median is the 50th percentile score which splits the distribution into two equal parts.
- The box boundaries indicate the 25th and 75th percentiles; thus only 25% of the scores fall below or above the box boundaries.
- The whiskers provide information about the most extreme scores in the distribution (the range); thus the whiskers represent the 25% between the box and the whisker.
- If there are dots that fall beyond the whiskers, they are treated as outliers, which makes identifying values easy. There are no outliers here, however.

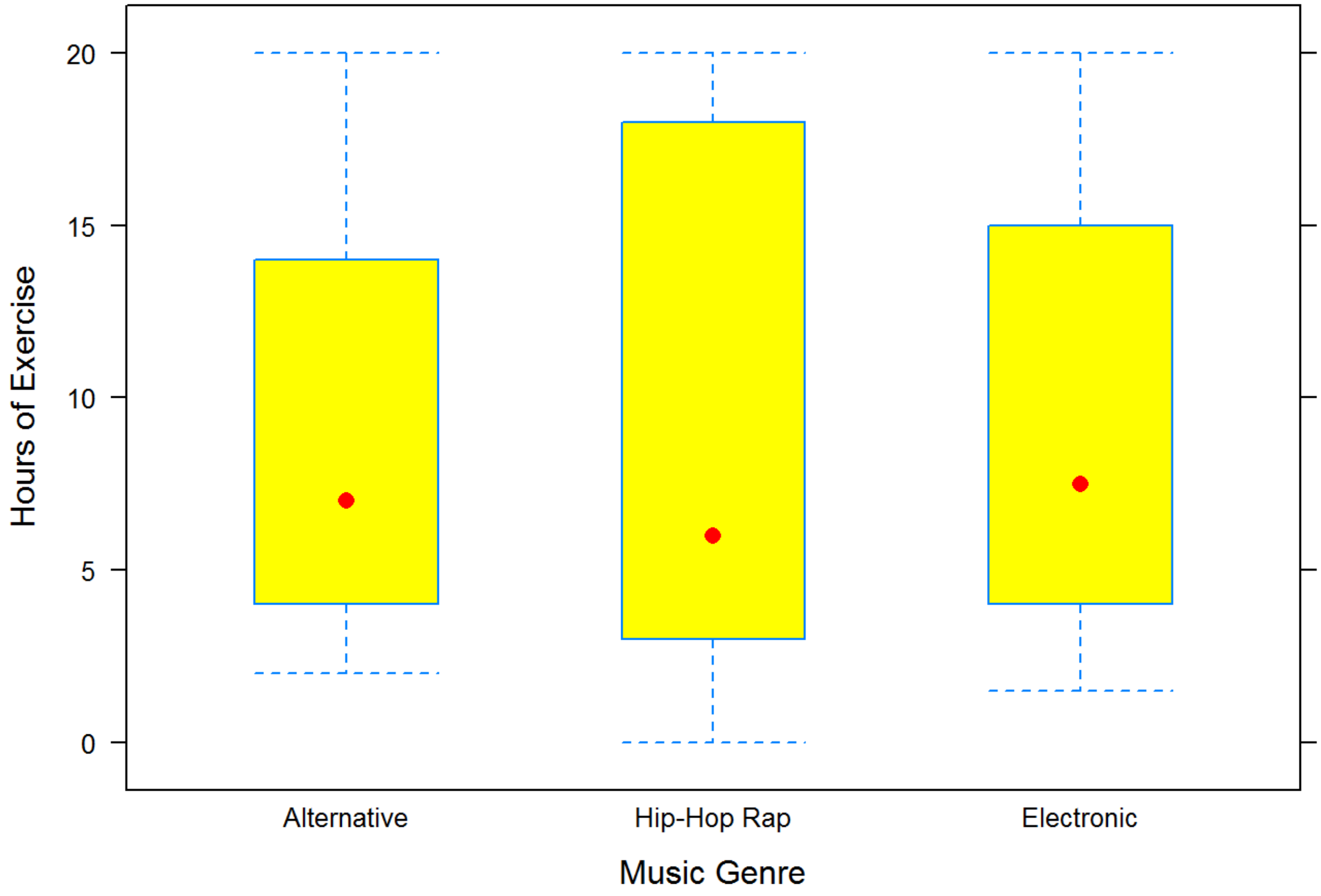
```
# basic (not really pretty)
boxplot(SURVEY$Exercise ~ SURVEY$MusGenre.fact)
```



```
# lattice version
bwplot(SURVEY$Exercise ~ SURVEY$MusGenre.fact)
```



```
# lattice more fancy
bwplot(SURVEY$Exercise ~ SURVEY$MusGenre.fact,
       ylab = "Hours of Exercise",
       xlab = "Music Genre",
       col = "red",
       fill = "yellow")
```



Part D

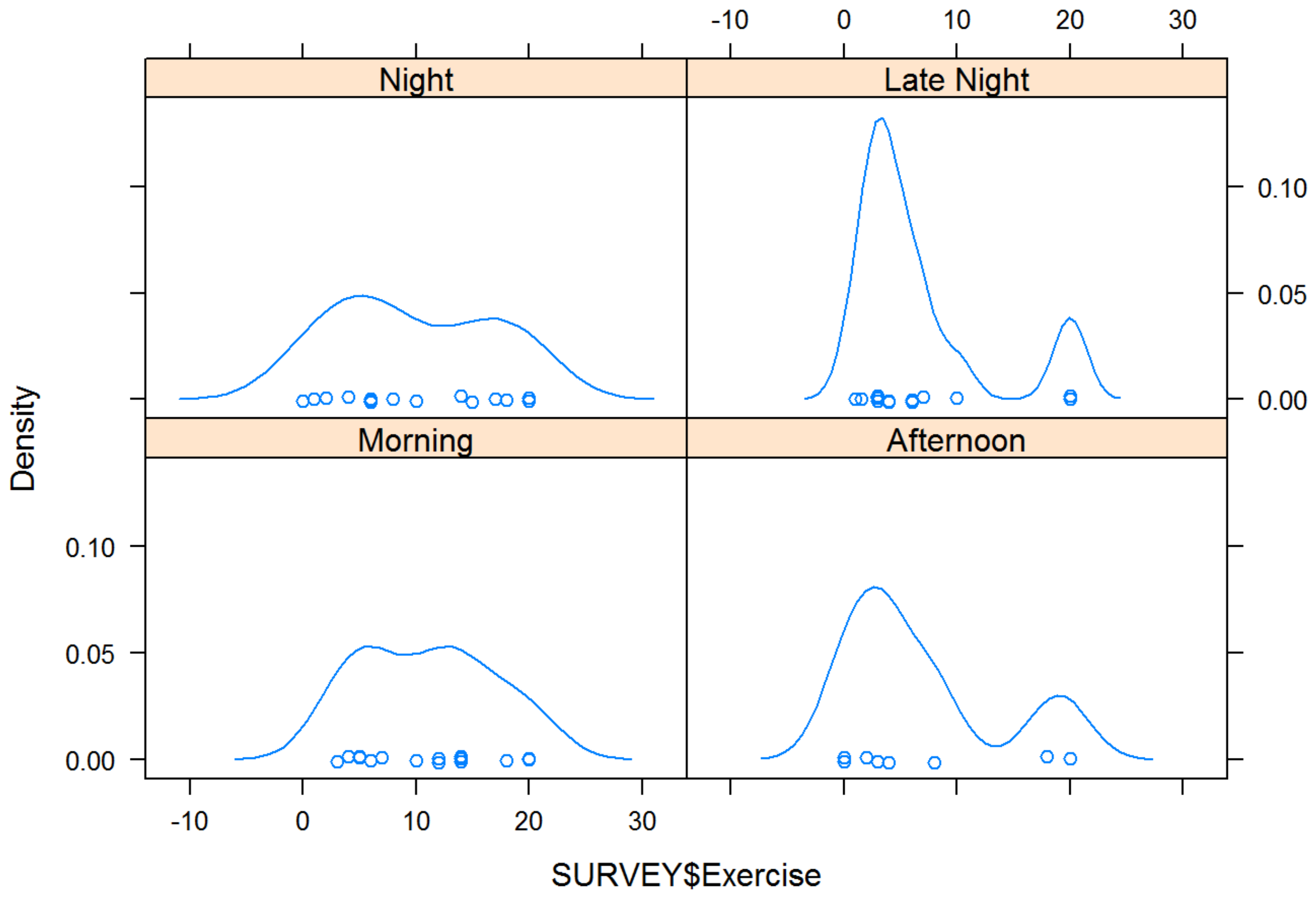
Do it yourself!

One question asked on the survey was electronic device used most and hours spent on social media. Consider the case for which you want to determine whether people who listen different types of music exercise different amounts. The variable names in the SURVEY data frame are *FavTime.fact* (we already made this a factor) and *Exercise*.

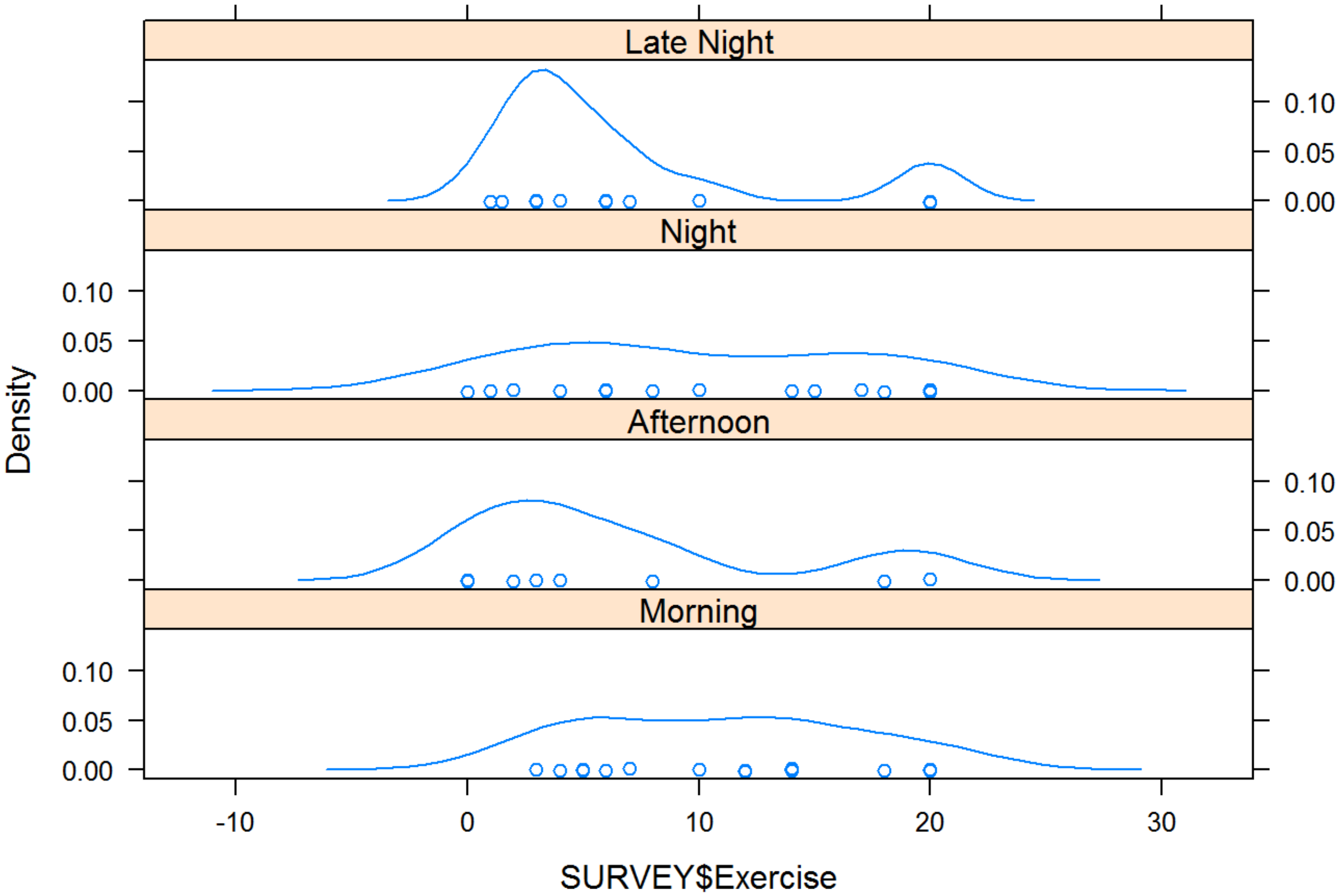
1. **QUESTION:** Produce a density plot of the DV for each level of the IV.

CODED ANSWER:

```
# this is fine.
densityplot(~ SURVEY$Exercise | SURVEY$FavTime.fact)
```



```
# but this may be more useful because the the scores are positioned on the same X-axis.
densityplot(~ SURVEY$Exercise | SURVEY$FavTime.fact, layout = c(1, nlevels(SURVEY$FavTim
e.fact)))
```



2. **QUESTION:** Based on how the plots look, you might expect there to be differences in normality for each group as well as with variance across the groups. Test for that.

CODED ANSWER:

```
by(SURVEY$Exercise, SURVEY$FavTime.fact, shapiro.test)
```

```
## SURVEY$FavTime.fact: Morning
## 
##  Shapiro-Wilk normality test
## 
## data:  dd[x, ]
## W = 0.92341, p-value = 0.2171
## 
## -------------------------------------------------------- 
## SURVEY$FavTime.fact: Afternoon
## 
##  Shapiro-Wilk normality test
## 
## data:  dd[x, ]
## W = 0.8309, p-value = 0.03431
## 
## -------------------------------------------------------- 
## SURVEY$FavTime.fact: Night
## 
##  Shapiro-Wilk normality test
## 
## data:  dd[x, ]
## W = 0.91866, p-value = 0.1838
## 
## -------------------------------------------------------- 
## SURVEY$FavTime.fact: Late Night
## 
##  Shapiro-Wilk normality test
## 
## data:  dd[x, ]
## W = 0.74194, p-value = 0.001041
```

```
# The afternoon and late-night groups look like they are not normal. Therefore, normality is not a reasonable assumption for all groups.
```

3. **QUESTION:** You might also wonder if the groups had similar variances. Test for the assumption of homogeneity of variance with center = mean.

CODED ANSWER:

```
leveneTest(SURVEY$Exercise, SURVEY$FavTime.fact, center = mean)
```

```
## Levene's Test for Homogeneity of Variance (center = mean)
##       Df F value Pr(>F)
## group  3  0.5863 0.6268
##       50
```

```
# F(3, 50) = .5862, p > .05. Therefore, equal variances is a reasonable assumption of the
data.
```

4. **QUESTION:** Create the ANOVA model using the aov() function and name the model something:

CODED ANSWER:

```
EXERCISEFAVTIME.aov <- aov(SURVEY$Exercise ~ SURVEY$FavTime.fact)
```

5. **QUESTION:** Examing the model ouput to determine whether your ANOVA test reveals differences between groups.

CODED ANSWER:

```
anova(EXERCISEFAVTIME.aov)
```

```
## Analysis of Variance Table
##
## Response: SURVEY$Exercise
##                     Df  Sum Sq Mean Sq F value Pr(>F)
## SURVEY$FavTime.fact  3  198.54  66.181   1.579 0.2061
## Residuals           50 2095.67  41.913
```

6. **QUESTION:** What is the Sum of Squares for Between-groups?

ANSWER: 198.54. BONUS: The variance is 66.181 because that’s the same as the MS.

7. **QUESTION:** What is the Sum of Squares for Within-groups?

ANSWER: 2095.67 BONUS: The variance is 41.913 because that’s the same as the MS.

8. **QUESTION:** Were the groups significantly different? Explain how you made that decision.

ANSWER: No. *F*(3, 50) = 1.579, *p* < .05. The *F*-value does not indicate much of the variability in Exercise is explained by a model based on preferred time of day. Perhaps there is another model to explain why people exercise different amounts, but time of day preferences do not seem to explain those differences (at least in this data set).

9. **QUESTION:** Run a post-hoc test to determine which factors are significantly different and the direction. For example, use the Tukey HSD test to compare all groups or a pairwise test with the Bonferroni correction.

CODED ANSWER:

```
# Of course, the ANOVA did not reveal differences between the groups, so this part would
not be completed. One way to answer this is to say "I can't do this because my ANOVA is n
ot significant." However, if you did need to conduct a post-hoc test, you would do it as
shown below.

# Tukey's HSD test (adjusting the p-value)
TukeyHSD(EXERCISEFAVTIME.aov)  # All adjusted p-values > .05. No groups are different. Ev
en the biggest difference between means (Late Night vs. Morning people does not have p <
.05)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = SURVEY$Exercise ~ SURVEY$FavTime.fact)
##
## $`SURVEY$FavTime.fact`
##                             diff        lwr       upr     p adj
## Afternoon-Morning    -4.2333333 -11.257386  2.790719 0.3869778
## Night-Morning        -1.1333333  -7.415837  5.149170 0.9633003
## Late Night-Morning   -4.3976190 -10.791326  1.996088 0.2725839
## Night-Afternoon       3.1000000  -3.924053 10.124053 0.6465366
## Late Night-Afternoon -0.1642857  -7.287975  6.959404 0.9999162
## Late Night-Night     -3.2642857  -9.657993  3.129421 0.5319014
```

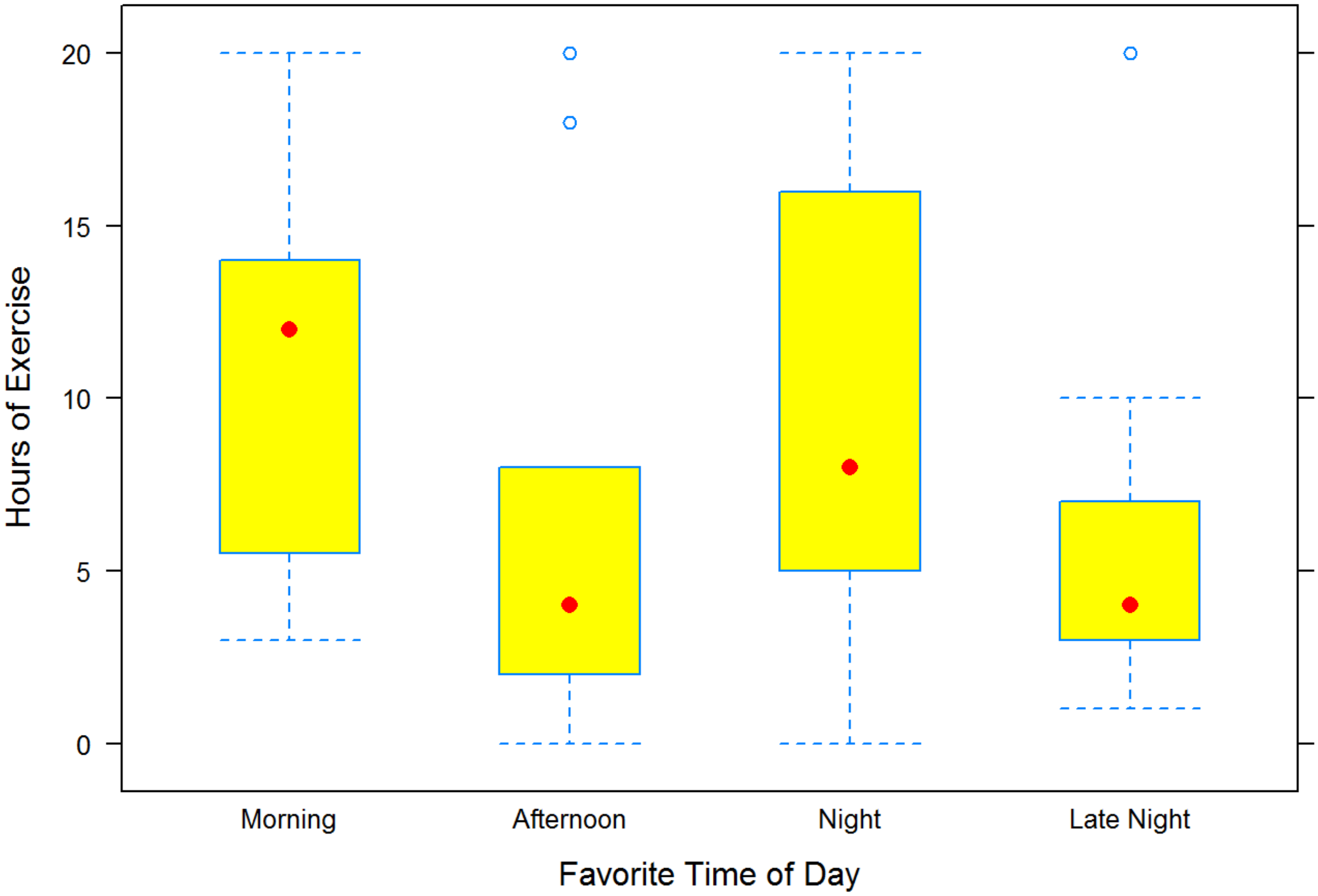
```
# Bonferroni correction (adjusting alpha)
pairwise.t.test(SURVEY$Exercise, SURVEY$FavTime.fact, p.adj = "bonf")  # All comparisons
after a Bonferroni-corrected alpha also does not reveal p < (.05/6 for all 6 comparison)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  SURVEY$Exercise and SURVEY$FavTime.fact
##
##            Morning Afternoon Night
## Afternoon  0.69    -         -
## Night      1.00    1.00      -
## Late Night 0.44    1.00      1.00
##
## P value adjustment method: bonferroni
```

10. **QUESTION:** Now that you have your data, you might want to share a graph. Create a boxplot.

CODED ANSWER:

```
bwplot(SURVEY$Exercise ~ SURVEY$FavTime.fact,
       ylab = "Hours of Exercise",
       xlab = "Favorite Time of Day",
       col = "red",
       fill = "yellow")
```



```
# Based on the box-and-whisker plot, there may be outliers to examine. Cleaning up your d
ata before proceeding to inferential statistics like the ANOVA is always important. You s
hould always look for outliers in your data even thought this was not explicitly asked in
this exercise.
```