IPS9 in R: One-way analysis of variance (Chapter 12)

Shukry Zablah (szablah20@amherst.edu) and Nicholas Horton (nhorton@amherst.edu)

July 19, 2018

Introduction and background

These documents are intended to help describe how to undertake analyses introduced as examples in the Ninth Edition of *Introduction to the Practice of Statistics* (2017) by Moore, McCabe, and Craig.

More information about the book can be found here. The data used in these documents can be found under Data Sets in the Student Site. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at https://nhorton.people.amherst.edu/ips9/.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (http://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

Chapter 12: One-way analysis of variance

This file replicates the analyses from Chapter 12: One-way analysis of variance.

First, load the packages that will be needed for this document:

```
library(mosaic)
library(readr)
```

> Complicated computations do not guarantee a valid statistical analysis. (648)

Section 12.1: Inference for one-way analysis of variance

We begin with Example 12.3 in page 648. Let's read in our data

```
Friends <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter12/EG12-03FRIENDS.csv")
head(Friends)
```

```
## # A tibble: 6 x 3
##   Friends Participant Score
##     <int>       <int> <dbl>
## 1     102           1   3.8
## 2     102           2   3.6
## 3     102           3   3.2
## 4     102           4   2.4
## 5     102           5   4.8
## 6     102           6   3
```

We want to get a nice summary table like the one in example 12.3. To do this we have to group the dataset by the number of friends. Then we can use some aggregating functions to get the desired table structure.

```
SummaryFriends <- Friends %>%
  group_by(Friends) %>%
  summarize(n = n(), x_hat = mean(Score), s = sd(Score))
SummaryFriends
```

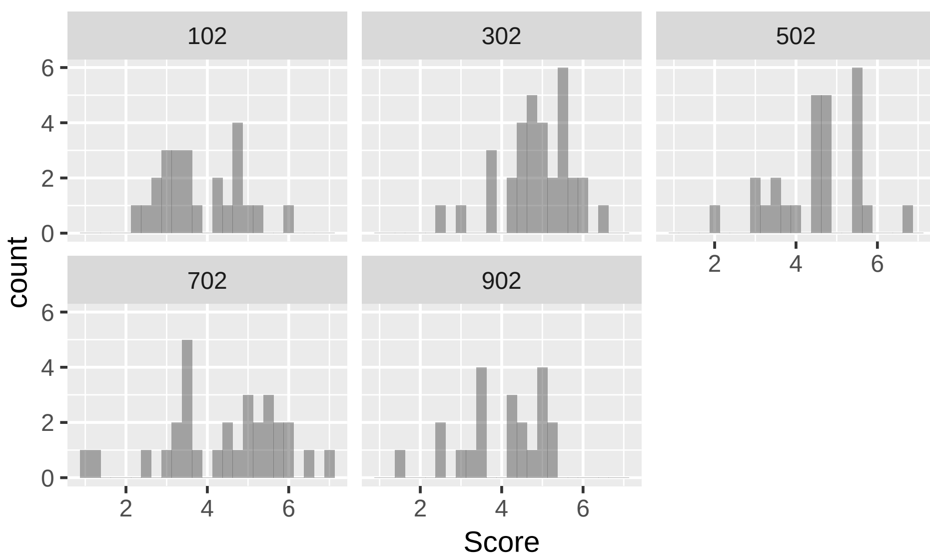
```
## # A tibble: 5 x 4
##   Friends     n x_hat     s
##     <int> <int> <dbl> <dbl>
## 1     102    24  3.82 0.999
## 2     302    33  4.88 0.851
## 3     502    26  4.56 1.07
## 4     702    30  4.41 1.43
## 5     902    21  3.99 1.02
```

In R the combination of `group_by()` and `summary()` is a powerful one. With these two data verbs we can create summary outputs like the one above with very little code.

There are some helpful visualizations starting in page 649 that will help us in our analysis of our Friends data.

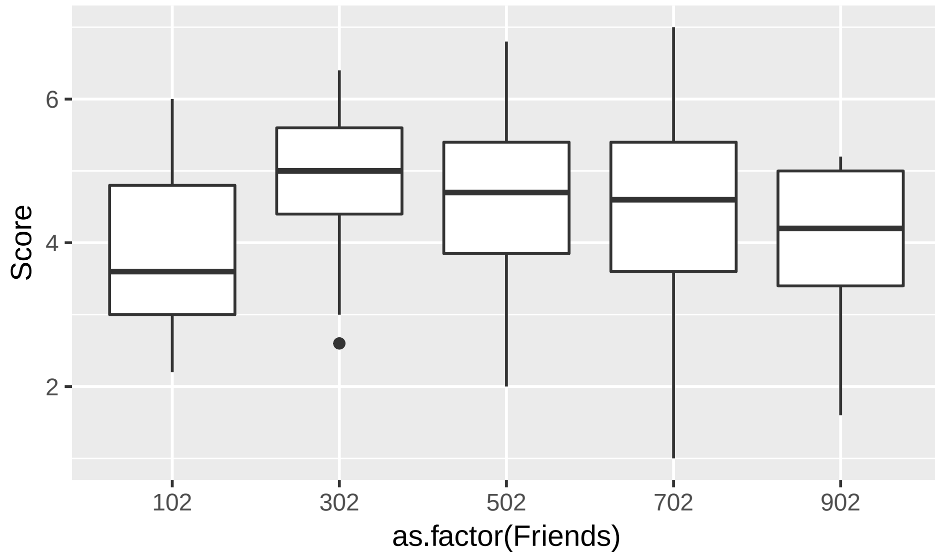
To recreate the histogram (note the `|` used to facet based on the number of friends):

```
gf_histogram(~ Score | Friends, data = Friends) #XX need to make this match the figure
```



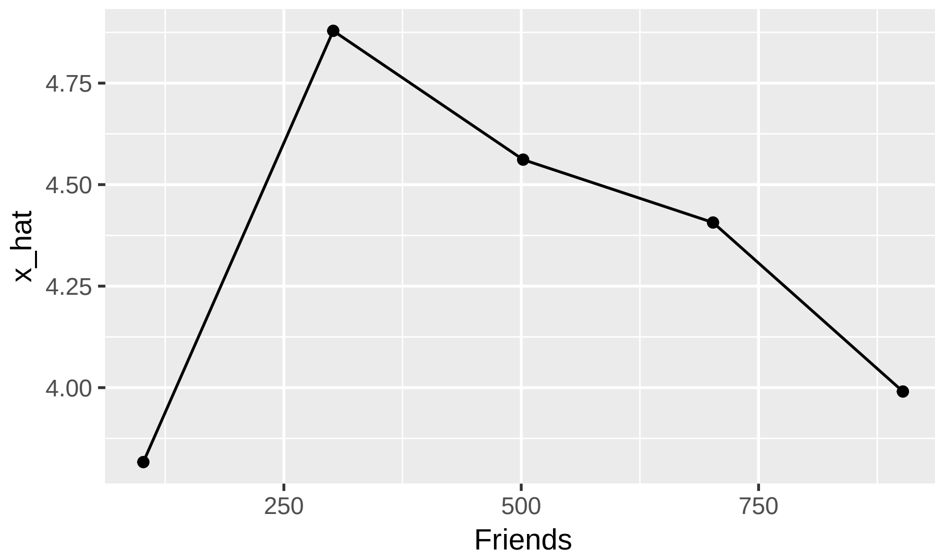
To recreate the boxplot (note: we have to convert Friends to a factor for the boxplots to work):

```
gf_boxplot(Score ~ as.factor(Friends), data = Friends)
```



And to recreate the linegraph (note the piping through gf_point to add the points to the line):

```
gf_line(x_hat ~ Friends, data = SummaryFriends) %>% gf_point()
```



These visualizations are a quick way to know what is going on with your data. Just a few lines of codes can save much time in your analysis.

```
Friends %>%
  group_by(Friends) %>%
  summarize(N = n(),
            Mean = mean(Score),
            Std.Dev = sd(Score),
            Minimum = min(Score),
            Maximum = max(Score))
```

```
## # A tibble: 5 x 6
##   Friends     N  Mean Std.Dev Minimum Maximum
##     <int> <int> <dbl>   <dbl>   <dbl>   <dbl>
```

```
## 1      102    24  3.82   0.999     2.2     6
## 2      302    33  4.88   0.851     2.6     6.4
## 3      502    26  4.56   1.07      2       6.8
## 4      702    30  4.41   1.43      1       7
## 5      902    21  3.99   1.02      1.6     5.2
```

XX Missing confidence intervals in table?

The ANOVA analysis can be done in a straightforward manner in R. We will create a linear model out of the Scores given their Friends group and pipe it into the anova function. (Note: the anova function takes an lm object. This is useful to us since we would maybe want to use the lm object for other purposes too.)

```
modFriends <- lm(Score ~ Friends, data = Friends)
modFriends %>%
  anova()
```

```
## Analysis of Variance Table
##
## Response: Score
##            Df  Sum Sq Mean Sq F value Pr(>F)
## Friends     1   0.102 0.10154  0.0767 0.7822
## Residuals 132 174.655 1.32315
```

Let's turn to page 663 and look at Example 12.15.

```
Eyes <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter12/EG12-15EYES.csv")
```

We will recreate the ouput from the Excel spreadsheet. With the help of the `group_by()` and `summarize()` idiom we can use the aggregating functions to summarize our dataset. To recreate the anova part of the output we use again the `anova()` function that takes an `lm` object.

```
Eyes %>%
  group_by(Group) %>%
  summarize(n = n(),
            Sum = sum(Score),
            Average = Sum / n,
            Variance = sd(Score)^2)
```

```
## # A tibble: 4 x 5
##   Group     n   Sum Average Variance
##   <chr> <int> <dbl>   <dbl>    <dbl>
## 1 Blue     67  214     3.19     3.08
## 2 Brown    37  138.    3.72     2.94
## 3 Down     41  127.    3.11     2.33
## 4 Green    77  297.    3.86     2.78
```

```
modEyes <- lm(Score ~ Group, data = Eyes)
modEyes %>%
  anova()
```

```
## Analysis of Variance Table
##
## Response: Score
##            Df Sum Sq Mean Sq F value  Pr(>F)
## Group       3  24.42  8.1399  2.8941 0.03618 *
## Residuals 218 613.14  2.8126
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

XXExample 12.16 recreate table equal variance test

Section 12.2: Comparing the means

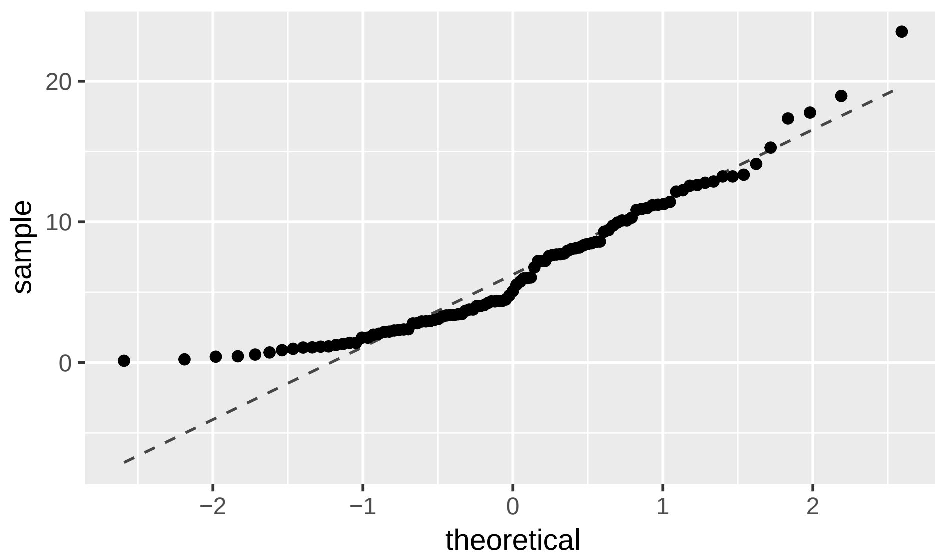
Anova gives us the answer to the question "are the differences between the means of the groups statistically significant?" However, it gives no information on what these differences are. This section covers those PostHocTests.

Let's read in the dataset for the times for times people spend on Facebook.

```
Facetym <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter12/EG12-17FACETYM.csv")
```

The analysis starts by checking the data. We will check the distribution of our data with a qqplot.

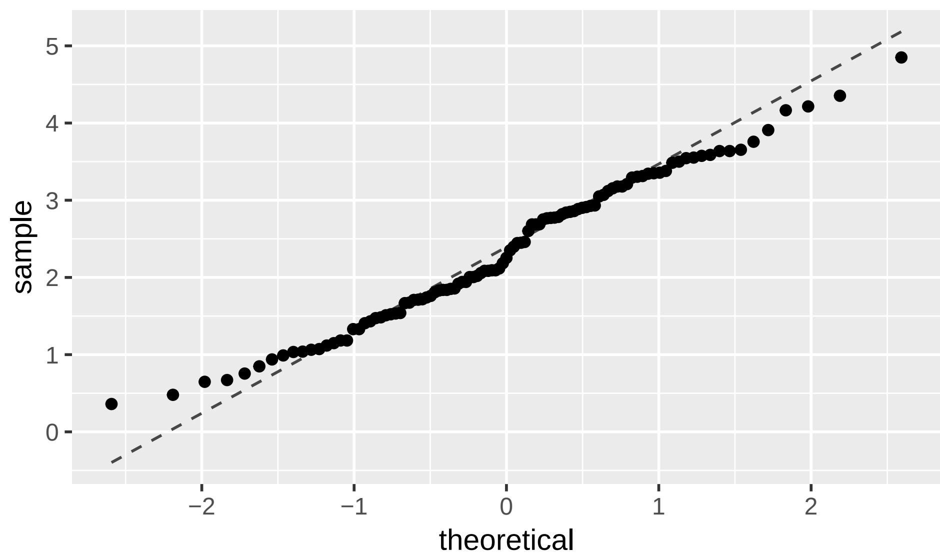
```
gf_qq(~ Time, data = Facetym) %>%
  gf_qqline()
```



The `gf_qq()` is piped to `gf_qqline()` in order to get the reference line through the middle of the plot. Note the skewdness of the data.

We now visualize the transformed data.

```
gf_qq(~ SqrtTime, data = Facetym) %>%
  gf_qqline()
```



Much better. Next step in our analysis is to get some descriptive statistics about the data. After that we perform the Anova test with `anova()`.

```
Facetym %>%
  group_by(Grp) %>%
  summarize(n = n(),
            Sum = sum(SqrtTime),
            Average = mean(SqrtTime),
            SD = sd(SqrtTime),
            min = min(SqrtTime),
            max = max(SqrtTime)
            )
```

```
## # A tibble: 5 x 7
##     Grp     n   Sum Average    SD   min   max
##   <int> <int> <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1     1    21  52.9    2.52 0.850 0.849  3.76
## 2     2    21  54.3    2.58 0.892 1.04   4.35
## 3     3    21  50.5    2.40 0.921 1.06   4.17
## 4     4    21  54.9    2.61 1.04  0.480  4.85
## 5     5    21  33.6    1.60 0.834 0.361  3.65
```

```
modFacetym <- lm(SqrtTime ~ factor(Grp), data = Facetym)
modFacetym %>%
  anova()
```

```
## Analysis of Variance Table
##
## Response: SqrtTime
##              Df Sum Sq Mean Sq F value   Pr(>F)
## factor(Grp)   4  15.08  3.7701   4.545 0.002051 **
## Residuals   100  82.95  0.8295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the result is statistically significant, there is enough evidence to reject the null hypothesis that the mean differences are all 0. However, we now need PostHocTests in order to find out what those differences are.

We will use a package called `DescTools` which has many multiple comparisons tests for after you reject the null hypothesis based on your `anova()` output.

To use the `PostHocTest()` function we have to pass it an object returned by the `aov()` function, which does the ANOVA test for a formula (and not an `lm` object). After that we only have to specify the `method` parameter, which is equal to the name of the multiple comparisons test that you want to perform.

```
library(DescTools)
```

```
##
## Attaching package: 'DescTools'

## The following object is masked from 'package:mosaic':
##
##     MAD
```

```
PostHocTest(aov(SqrtTime ~ factor(Grp), data = Facetym), method = "bonferroni")
```

```
##
##   Posthoc multiple comparisons of means : Bonferroni
##     95% family-wise confidence level
##
## $`factor(Grp)`
##            diff     lwr.ci       upr.ci   pval
## 2-1  0.06723591 -0.7396166  0.874088419 1.0000
## 3-1 -0.11275242 -0.9196049  0.694100090 1.0000
## 4-1  0.09736089 -0.7094916  0.904213396 1.0000
## 5-1 -0.91714318 -1.7239957 -0.110290666 0.0151 *
## 3-2 -0.17998833 -0.9868408  0.626864181 1.0000
## 4-2  0.03012498 -0.7767275  0.836977488 1.0000
## 5-2 -0.98437909 -1.7912316 -0.177526574 0.0069 **
## 4-3  0.21011331 -0.5967392  1.016965818 1.0000
## 5-3 -0.80439076 -1.6112433  0.002461756 0.0513 .
## 5-4 -1.01450406 -1.8213566 -0.207651551 0.0048 **
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
PostHocTest(aov(SqrtTime ~ factor(Grp), data = Facetym), method = "lsd")
```

```
##
##   Posthoc multiple comparisons of means : Fisher LSD
##     95% family-wise confidence level
##
## $`factor(Grp)`
##            diff     lwr.ci     upr.ci    pval
## 2-1  0.06723591 -0.4903979  0.6248697 0.81143
## 3-1 -0.11275242 -0.6703863  0.4448814 0.68916
## 4-1  0.09736089 -0.4602729  0.6549947 0.72977
## 5-1 -0.91714318 -1.4747770 -0.3595093 0.00151 **
## 3-2 -0.17998833 -0.7376222  0.3776455 0.52340
## 4-2  0.03012498 -0.5275089  0.5877588 0.91486
## 5-2 -0.98437909 -1.5420129 -0.4267453 0.00069 ***
## 4-3  0.21011331 -0.3475205  0.7677471 0.45649
## 5-3 -0.80439076 -1.3620246 -0.2467569 0.00513 **
## 5-4 -1.01450406 -1.5721379 -0.4568702 0.00048 ***
##
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

XX Anova power?