

Answers to Application Activities in Chapter 3

3.2.2 Application Activities for Numerical Summaries in SPSS

1 DeKeyser (2000)

To split files, choose DATA > SPLIT FILE. Click radio button for “Compare Groups” and move STATUS into the box. Click OK. To get the numerical summaries, choose ANALYZE > DESCRIPTIVE STATISTICS > DESCRIPTIVES, and move GJTSORE to the “Variable(s)” box. Open the OPTIONS button and make sure all the required statistics are ticked. Open the “Bootstrap” button and tick the “Perform bootstrapping” box. Change the radio button under “Confidence Intervals” to BCa. Press “Continue,” then press OK. You could also use the ANALYZE > DESCRIPTIVE STATISTICS > EXPLORE choice and would not need to split the file. There are 15 participants in the group with Status 1 (Under 15), and 42 in the group with Status 2 (Over 15). Their mean scores and confidence intervals look quite different (Under 15 mean = 191.67, 95% BCa bootstrapped CI [187.1, 195.5]; Over 15 mean = 145.14, CI [138.4, 152.1]), given that 200 was the maximum score. Also, the Under 15 group had a much higher minimum score (170) than the Over 15 group (76). The standard deviation also looks quite different (Group 1 $s = 8.45$, Group 2 $s = 24.3$).

2 Obarow (2004)

To split files, choose DATA > SPLIT FILES. Click the radio button for “Compare Groups” and move TRTMNT1 into the box. Press OK. To get the numerical summaries, choose ANALYZE > DESCRIPTIVE STATISTICS > DESCRIPTIVES, and move GNSC1.1 to the “Variable(s)” box. Open the

OPTIONS button and make sure all the required statistics are ticked. Open the “Bootstrap” button and tick the “Perform bootstrapping” box. Change radio button under “Confidence Intervals” to BCa. Press “Continue,” then press OK. You could also use the ANALYZE > DESCRIPTIVE STATISTICS > EXPLORE choice and would not need to split the file. There are 20 participants in each group except the “Yes music No pics” group, which has 18. Note that because the gain score is the posttest score minus the pretest score, gain scores can be negative, as seen by negative numbers in the minimum statistic. Mean scores for the 4 groups are all very small, and range from a low of .45 to a high of 1.25. Confidence intervals: 1) No music no pics, 95% BCa bootstrapped CI [-.18, 1.15]; 2) No music yes pics, CI [.72, 1.76]; 3) Yes music no pics, CI [.19, 1.7]; 4) Yes music yes pics, CI [.19, 1.7]. Therefore, it does not appear that mean scores or confidence intervals are all that different. Standard deviations range from a low of 1.12 to a high of 1.82, which also does not seem very different.

3 Larson-Hall & Connell (2005)

To split files, choose DATA > SPLIT FILES. Click radio button for “Compare Groups” and move STATUS into the box. Press OK. To get the numerical summaries, choose ANALYZE > DESCRIPTIVE STATISTICS > DESCRIPTIVES, and move R/L/W TEST (Score on the R/L/W listening test) to the “Variable(s)” box. Open the “Bootstrap” button and tick the “Perform bootstrapping” box. Change the radio button under “Confidence Intervals” to BCa. Press “Continue,” then press OK. You could also use the ANALYZE > DESCRIPTIVE STATISTICS > EXPLORE choice and would not need to split the file. There are 15 participants in the Non group, mean = 67.73, 95% BCa bootstrapped CI [61.0, 74.2], SD = 13.76, min = 47, max = 91. There are 15 participants in the Late group, mean = 69.20, CI [61.1, 77.4], SD = 15.37, min = 36, max = 90. There are 14

participants in the Early group, mean = 80.86, CI [73.3, 88.0], SD = 14.27, min = 47, max = 95.

The Non and Late groups are fairly similar to each other in mean score and standard deviation.

The Early group scores the highest, but still has wide standard deviation, similar to the other two groups.

3.3.2 Application Activities for Numerical Summaries in R

1 DeKeyser (2000)

In R commander, first make sure the imported **dekeyser** file is the active data set. Then go to STATISTICS > SUMMARIES > NUMERICAL SUMMARIES.

Choose the variable **GJTScore** and in the “Summarize by groups” button choose **Status**.

By default you’ll get the mean, standard deviation, quantiles (including min and max at the end of the quantiles) and N.

Here is the R code for this command:

```
numSummary(dekeyser[, "gjtscore"], groups=dekeyser$status,  
statistics=c("mean", "sd", "IQR", "quantiles"), quantiles=c(0,.25,.5,.75,1))
```

```
      .      .      .      .      .      .      .      .  
      mean      sd  0%   25%  50%   75% 100%   n  
Under 15 191.6667  8.448725 170 188.0 195 197.0 199 15  
Over 15  145.1429 24.270399  76 127.5 143 163.5 190 42
```

To get confidence intervals, first look at which rows form which groups. Open the “View data set” button and you’ll see the “Under 15” group goes from rows 1–15, while the “Over 15” groups goes from rows 16–57 (or you might use the R code **summary(dekeyser\$status)**

because there are only two groups and we expect they will all group together in the rows). Now run the syntax for the confidence interval for the “Under 15” group:

```
library(boot)

samplemean<-function(x,d){
  return(mean(x[d]))
}

b=boot(dekeyser$gjtscore[1:15], samplemean, R=1000)

plot(b)
```

The histogram from the `plot()` command looks distinctly non-normal so I try it again with 2000 replicates:

```
b=boot(dekeyser$gjtscore[1:15], samplemean, R=2000)

plot(b)
```

The histogram still looks somewhat bad, so I will try 3000 replicates:

```
b=boot(dekeyser$gjtscore[1:15], samplemean, R=3000)

plot(b)
```

This looks OK, so I will now call for confidence intervals:

`boot.ci(b)`

The BCa 95% CI for the Under 15 group is [186.2, 194.8] (remember, because it's a bootstrap and uses random groupings your numbers will be slightly different!). This means that although we have a point estimate for the mean for the Under 15 group of 191.7, we can believe this interval contains the true population mean, with 95% confidence. Repeating the process for rows 16:57 (the plot looks fine with 1000 replicates), I find the BCa 95% CI for the Over 15 group is [137.5, 151.7].

It does not look like the groups have similar mean scores or standard deviations. They look quite different.

2 Obarow (2004)

In R commander, go to STATISTICS > SUMMARIES > NUMERICAL SUMMARIES.

Choose the variable `gnsc1.1` and in the "Summarize by groups" button choose `trtmnt1`.

This gives mean, standard deviation, quantiles and N.

Here is the R code for this command:

```
numSummary(obarow[, "gnsc1.1"], groups=obarow$trtmnt1, statistics=c("mean",  
"sd", "IQR", "quantiles"), quantiles=c(0,.25,.5,.75,1))
```

		mean	sd	0%	25%	50%	75%	100%	n	NA
No music	No pics	0.4500000	1.538112	-2	-1.00	0.5	1	5	20	1
No music	Yes pics	1.2500000	1.118034	-1	1.00	1.0	2	3	20	0
Yes music	No pics	0.7222222	1.708303	-2	-0.75	0.5	2	4	18	2
Yes music	Yes pics	0.9500000	1.820208	-2	0.00	1.0	2	6	20	0

As we see, there are NAs in the data but they appear to be missing at random, so let's impute the data:

```
library(mice)
```

```
imp<-mice(obarow, 5)
```

```
impObarow<-complete(imp)
```

```
sapply(impObarow, function(x)(sum(is.na(x)))) #just to check there are no NAs now
```

To find the CIs, you'll use the `boot` library but if you already opened it and wrote the `samplemean` function in Exercise #1, you won't need to do those steps again.

Checking out the obarow file, we see the groups go from row 1–20 (no music yes pics), 21–40 (yes music no pics), 41–60 (yes music yes pics) and 61–81 (no music no pics). So the code will be:

```
b=boot(obarow$gnsc1.1[1:20], samplemean, R=1000) #no music yes pics
```

```
plot(b)
```

```
boot.ci(b)
```

Repeat this process with the other rows, and you should get something close to the following results:

```
b=boot(obarow$gnsc1.1[21:40], samplemean, R=1000) #yes music no pics
```

So this one doesn't work! I get the following error message:

```
> plot(b)
Error in if (t0 < rg[1L]) rg[1L] <- t0 else if (t0 > rg[2L]) rg[2L] <- t0 :
missing value where TRUE/FALSE needed
> boot.ci(b)
Error in bca.ci(boot.out, conf, index[1L], L = L, t = t.o, t0 = t0.o, :
estimated adjustment 'w' is infinite
```

I don't know enough about bootstrapping to know how to fix this.

```
b=boot(obarow$gnsc1.1[41:60], samplemean, R=1000) #yes music yes pics
```

```
b=boot(obarow$gnsc1.1[61:81], samplemean, R=1000) #no music yes pics
```

The third one works (yes music yes pics) but the last one gives the same warning messages.

Again, I don't know how to take care of this. So I can say the following only:

no music yes pics 95% BCa bootstrapped CI: [0.70, 1.70] (M = 1.25)

yes music yes pics 95% BCa bootstrapped CI: [0.25, 1.80] (M = 0.95)

3 Larson-Hall & Connell (2005)

In R commander, with data set **forget** set as the active data set, go to STATISTICS > SUMMARIES > NUMERICAL SUMMARIES.

Choose the variable **rlwtest** and in the “Summarize by groups” button choose **status**.

This gives mean, standard deviation, quantiles and N.

```
numSummary(forget[, "rlwtest"], groups=forget$status, statistics=c("mean", "sd", "IQR",
"quantiles"), quantiles=c(0,.25,.5,.75,1))
```

	mean	sd	IQR	0%	25%	50%	75%	100%	data:n
Non	67.73333	13.76054	22.50	47	56.5	67	79.00	91	15
Late	69.20000	15.37716	20.00	36	60.5	66	80.50	90	15
Early	80.85714	14.27670	19.25	47	71.5	88	90.75	95	14

Checking out the **forget** file, we see the Non group goes from rows 1–15, Late from 16–30, and

Early from 31–44.

To find the CIs, you'll use the `boot` library but if you've already opened it and wrote the `samplemean` function in Exercise #1, you won't need to do those steps again.

Repeat this process with the other rows, and you should get the following results:

```
b=boot(forget$rlwtest[1:15], samplemean, R=1000) #Non  
plot(b)  
boot.ci(b)
```

Repeat this process with the other rows, and you should get something close to the following results:

Non: 95% BCa bootstrapped CI: [61.1, 74.9] (M = 67.7)

Late: 95% BCa bootstrapped CI: [60.5, 75.4] (M = 69.2)

Early: 95% BCa bootstrapped CI: [72.1, 86.9] (M = 80.9) (this bootstrap looked skewed no matter how I increased the replicates . . .)

The Non and Late groups are very similar in their mean scores but all of the groups are fairly similar in their standard deviations. The confidence intervals show the Non and Late performed very similarly too (the confidence intervals are nearly the same). The Early group performed the best.

3.6.1 Application Activity: Looking at Normality Assumptions

1 Flege, Yeni-Komshian, and Liu (1999)

If you use the EXPLORE option, put the pronunciation rating in the “Dependent” list and the group in the “Factor” list. Looking at histograms with a normal curve imposed, groups 4, 10, and 11 seem to be clearly different from a normal distribution, with group 11 having positive skewness and groups 4 and 10 having some outliers (too much data in the tails of the distributions for a normal distribution). The Shapiro-Wilk test of normality is under $p = .05$ for groups 10 and 11. Stem and leaf plots show two modes for both groups 10 and 11.

2 DeKeyser (2000)

First, split the file by using DATA > SPLIT FILE then putting the STATUS variable in the box and using the “Compare Groups” button. Now create a histogram with normal curve imposed by using ANALYZE > DESCRIPTIVE STATISTICS > FREQUENCIES. Put the GJTSCORE in the “Variables” box. Click the CHARTS button and pick “Histograms” and also tick the “With normal curve” box. Click CONTINUE and take the tick off “Display frequency tables” in the main dialogue box (they give a lot of data and are not necessary here). Click OK. The data for the Under 15 group are clearly negatively skewed (with the most frequent scores being near the maximum of 200 points, and the tail of the distribution going to the left). The data for the Over 15 group seem much more normally distributed, with no clear skewing and the majority of observations in the middle area.

3 Lyster (2004)

I used the Explore option for this. I put the COMPGAIN1 and COMPGAIN2 in the “Dependent List” box and the COND in the “Factor List” box. I left the STATISTICS button alone but opened the

PLOTS button and asked for “None” for boxplots and added “Histogram” and “Normality plots with tests,” then continued. Note that the groups are fairly large in this data set, with almost all near or above 40 per group. For COMPGAIN1, the histograms look fairly normal except for the “Comparison” condition. For COMPGAIN2, the histogram for the “FFIonly” condition is definitely positively skewed and again for the “Comparison” condition there is a lot of data across the whole spectrum of scores. The same trends can generally be observed in the Q-Q plots. The formal Shapiro-Wilks test of normality finds a problem only for COMPGAIN2 for the “FFIonly” group.

4 Leow (1997)

Import the Excel file into SPSS by going to FILE > OPEN > DATA, then changing the type of file to an Excel file and navigating to the place you have stored the file. It should open with only one additional question about which sheet of the worksheet to import (but you will have to name it yourself). I used the Explore option (to avoid splitting groups) and put PRODUCTION into the “Dependent List” box and CATEGORY in the “Factor List” box. In the PLOTS button I asked for “None” for boxplots and added “Histogram” and “Normality plots with tests”, then continued. The formal test of normality find a problem for CATEGORY A but not for CATEGORY B. Neither one of the histograms for Category A or B seems especially normal but the N is not so large (15 in A, 13 in B). The Q-Q plots don’t look great but don’t look too terribly skewed either.

3.7.5 Application Activity: Looking at Normality Assumptions with R

1 DeKeyser (2000)

Import the SPSS file DeKeyser2000.sav file (open R Commander from R console by typing `library(Rcmdr)`; in R Commander, choose DATA > IMPORT DATA > FROM SPSS DATA SET). Get histograms in R Commander by choosing GRAPHS > HISTOGRAM, and then choosing the `gjtscore` variable. Open the “Plot by groups” button and choose `status`. The data for the Under 15 group are clearly negatively skewed (with the most frequent scores being near the maximum of 200 points, and the tail of the distribution going to the left). The data for the Over 15 group seem much more normally distributed, with no clear skewing and the majority of observations in the middle area. The R code for the histogram is:

```
with(dekeyser, Hist(gjtscore, groups=status, scale="frequency",  
breaks="Sturges", col="darkgray"))
```

For a Q-Q plot, in R Commander choose GRAPHS > QUANTILE-COMPARISON PLOT. Choose the `gjtscore` as the variable and press OK. Take the R code in the Output window and add row numbers for the two groups:

```
qqPlot(dekeyser$gjtscore[1:15], dist="norm", id.method="y", id.n=2,  
labels=rownames(dekeyser))  
  
qqPlot(dekeyser$gjtscore[16:57], dist="norm", id.method="y", id.n=2,  
labels=rownames(dekeyser))
```

For the Under 15 group, there is one point outside the normal distribution area, and one point right on the border. For the Over 15 group, all of the points are contained in the normal distribution area.

2 Flege, Yeni-Komshian, and Liu (1999)

Import the SPSS file FlegeYKLiu.sav (in R Commander, choose DATA > IMPORT DATA > FROM SPSS DATA SET). It wouldn't be easy to look up the row numbers of all the 11 groups, so let's use the Q-Q plot found in the lattice library instead, which can automatically split the file into different groups. We'll have to use the R console for this.

```
library(lattice)

qqmath(~pronunciation|group, aspect="xy", data=FYL, xlab="Q-Q plot",

prepanel = prepanel.qqmathline,

panel = function(x, ...) {

panel.qqmathline(x, ...)

panel.qqmath(x, ...)

})
```

This code will get us the basic Q-Q plots split by groups but they will not be labeled nicely. This is because in the original data file the variable 'group' is not a factor.

```
class(FYL$group)

[1] "numeric"
```

```
category=factor(FYL$group) #make the group names into a factor  
summary(category) #see the factor names
```

If you run the `lattice` library command again but substitute `category` instead of `group`, you will end up with the numbers of the groups on the graphs (which is not as informative as one might wish, but better than the previous condition).

Looking at the 8 groups, the data in groups 1 and 3 seem to follow the reference line quite nicely, meaning they are fairly normally distributed, and there are some deviations from the reference line in the other groups but most likely nothing too far outside a 95% confidence interval—in other words, none of the points looks too far away from the line, except perhaps in the very last group (Group 11). The data seem to be fairly normally distributed.

3 Lyster (2004)

Import the SPSS file `Lyster.written.sav` (in R Commander, choose DATA > IMPORT DATA > FROM SPSS DATA SET). For Q-Q plots, since there are 4 different groups in the `cond` variable (verify this by typing `summary(lyster$cond)` to see that this is a factor with 4 named conditions), let's use the `lattice` package Q-Q plot command.

```
library(lattice)  
qqmath(~compgain1|cond,aspect="xy", data=lyster, xlab="Q-Q plot",  
prepanel = prepanel.qqmathline,  
panel = function(x, ...) {
```

```
panel.qqmathline(x, ...)
```

```
panel.qqmath(x, ...)
```

```
})
```

Repeat with `compgain2`. None of the Q-Q plots for `compgain1` look too far from normal. For the data for the `FFIonly` group and the `FFIprompt` group look as though they may deviate from normality.

Looking now at histograms, we can use R Commander, choosing `GRAPHS > HISTOGRAM`, and then the two different variables (`compgain1` and `compgain2`). Click the “Plot by Groups” button and choose `cond`. The R code for the first variable is:

```
with(lyster, Hist(compgain1, groups=cond, scale="frequency",  
breaks="Sturges", col="darkgray"))
```

For `COMPAIN1`, the histograms look fairly normal except for the “`FFIrecast`” condition, which is slightly positively skewed. For `COMPAIN2`, the histogram for the “`FFIonly`” condition is definitely positively skewed and the “`FFIrecast`” and “`FFIprompt`” groups both have some slight skewness.

For the Shapiro-Wilk test we’ll need to know row numbers to most easily divide up the data by groups. Using R Commander, I open the “View Data Set” button (making sure `lyster` is the active

data set) and find that FFIrecast uses rows 1–38, FFIprompt uses rows 39–87, FFIonly has rows 88–129, and the Comparison group has rows 130–180. Using the R console, I type in:

```
shapiro.test(lyster$compgain1[1:38])#FFIrecast
```

and continue with both variables and all 4 conditions. The formal Shapiro-Wilks test of normality finds a problem (a p -value less than .05) only for COMPGAIN2 for the “FFIonly” group.

4 Leow (1997)

Import the Leow1997 Excel file into R using R Commander (DATA > IMPORT DATA > FROM TEXT FILE, CLIPBOARD, OR URL, and choose “Tabs” as the Field Separator. This will work if you have saved the file as a .txt file, which you need to do from inside Excel itself first). The file has 28 rows and 3 columns.

This exercise does not ask you to use any certain test, just asks whether the data are normally distributed. Looking at histograms (in R Commander, GRAPHS > HISTOGRAM), neither one of the histograms for Category A or B seems especially normal but the N is small (15 in A, 13 in B). The Shapiro-Wilk test of normality finds a problem for CATEGORY A but not for CATEGORY B.

```
shapiro.test(leow$Written.Production[1:15])#CategoryA
```

```
shapiro.test(leow$ Written.Production[16:28])#CategoryB
```

I would conclude that it would be hard to say that either group was normally distributed, but because the group size is so small, it is possible that it is normally distributed.

3.8.1 Application Activities for Checking Homogeneity of Variance

(Answers for both SPSS and R)

1 Flege, Yeni-Komshian, and Liu (1999)

SPSS Instructions: If you do not have your output from the previous application activity, the most compact way to call for standard deviations for all groups is to split groups (DATA > SPLIT FILE) then use ANALYZE > DESCRIPTIVE STATISTICS > DESCRIPTIVES. Using this method you can then open the OPTIONS button and call only for the standard deviation.

R Instructions: Using the imported FYL file, in R Commander get standard deviations by choosing STATISTICS > SUMMARIES > NUMERICAL SUMMARIES. Choose the **pronunciation** variable, and split by the variable **group** (when you first import this file, group will not be a factor, so you'll have to make it a factor. This is most easily done in R console:

```
summary(FYL$group) # you'll see it's treated as a normal variable and summarized
```

```
FYL$group=as.factor(FYL$group)
```

```
summary(FYL$group) # now you should see a summary of the number of cases for  
#each group)
```

To now run the NUMERICAL SUMMARIES command with the group as a factor in R Commander, you'll have to detach the FYL by choosing a different active dataset, then going back to FYL.

Now you'll see that group is a factor. In the dialogue box, click on the "Statistics" tab and click off everything except for standard deviation.

Evaluation of SD: The standard deviations range from a low of .32 for Group 1 to a high of 1.07 for Group 8. Since the total number of points is only 9, this seems like rather a large difference and we would conclude that the variances are not homogeneous.

Levene's test (for R):

Choose STATISTICS > VARIANCES > LEVENE'S TEST in R Commander for the pronunciation variable. The probability is $p = .0006243$, meaning that this test says variances are unequal (the probability of the null hypothesis that variances are equal is very small). In this case, the results from looking at the standard deviation and Levene's test concur. The R code for this is:

```
with(FYL, tapply(pronunciation, group, var, na.rm=TRUE))
```

```
leveneTest(pronunciation ~ group, data=FYL, center="median")
```

2 DeKeyser (2000)

SPSS Instructions: See information in Exercise #1. Split file by STATUS and use the GJTSCORE variable.

R Instructions: See information in Exercise #1.

Evaluation of SD: In a test of 200 points, the standard deviation for the Under 15 group is 8.5 and for the Over 15 group it is 24.3. This is a large difference and we conclude the groups are not homogeneous.

Levene's test (for R):

See information for Exercise #1. The probability is $p = .0008479$. Again, the results from looking at the standard deviation and Levene's test concur.

3 Lyster (2004)

Evaluation of SD: Keeping in mind that the maximum anyone gained was 21 points, the standard deviations range from 4.6 to 6.8, which is not that large a range. The groups appear to be fairly homogeneous.

Levene's test (for R):

The probability is $p = .109$. The interpretation of this test is that the variances are homogeneous, and this matches with the conclusion we made by looking at the SDs.