Relevant R commands

Building a raw data set with `rbind()` and `data.frame()`. Sometimes we are given summarized data in a contingency table, and it would be convenient to have the data as a raw data set. Consider the information form Table 2.11, p. 57. The combination of `do()` with `data.frame()`

```
do(8) * data.frame(smokingStatus = "smoker", pregnant = "yes")
```

```
  smokingStatus pregnant .row .index
1        smoker      yes    1      1
2        smoker      yes    1      2
3        smoker      yes    1      3
4        smoker      yes    1      4
5        smoker      yes    1      5
6        smoker      yes    1      6
7        smoker      yes    1      7
8        smoker      yes    1      8
```

generates a data frame with 2 columns called `smokingStatus"` and `pregnant`, and 8 rows as shown. We can generate a raw data set corresponding to Table 2.11 by using similar commands for the four cells of the table and binding them together using `rbind()`:

```
rawDataForTable2.11 <- rbind(
   do(38) * data.frame(smokingStatus = "smoker", pregnant = "yes"),
   do(206) * data.frame(smokingStatus = "nonsmoker", pregnant = "yes"),
   do(97) * data.frame(smokingStatus = "smoker", pregnant = "no"),
   do(337) * data.frame(smokingStatus = "nonsmoker", pregnant = "no")
  )
```

A contingency table built from this raw data should match Table 2.11.

```
tally(~pregnant | smokingStatus, data=rawDataForTable2.11)
```

```
        smokingStatus
pregnant smoker nonsmoker
     yes     38       206
     no      97       337
```

You may have the column totals, too, if you add `margins`:

```
tally(~pregnant | smokingStatus, data=rawDataForTable2.11, margins=TRUE)
```

```
        smokingStatus
pregnant smoker nonsmoker
   yes       38       206
   no        97       337
   Total    135       543
```

Getting `prop()` command to focus on the right value. Now that we have a raw data frame corresponding to Table 2.11, we might want to apply the `prop()` command to it.

```
prop(~pregnant | smokingStatus, data=rawDataForTable2.11)

   prop_yes.smoker prop_yes.nonsmoker
         0.2814815          0.3793738
```

This command appears to have chosen for itself what value of the variable `pregnant` to count. The 0.2814815 is the proportion, among all smokers in our sample, who *are* pregnant (have the value "yes" in the `pregnant` column). If we want to focus on those who have the value "no", we can use the `success` switch:

```
prop(~pregnant | smokingStatus, data=rawDataForTable2.11, success="no")

   prop_no.smoker prop_no.nonsmoker
        0.7185185         0.6206262
```

Caution: You should take note of what difference it makes to place the variables in the other order:

```
tally(~smokingStatus | pregnant, data=rawDataForTable2.11)
prop(~smokingStatus | pregnant, data=rawDataForTable2.11)
```

Computing differences with `diff()`. As we saw above, the command

```
prop(~pregnant | smokingStatus, data=rawDataForTable2.11)

   prop_yes.smoker prop_yes.nonsmoker
         0.2814815          0.3793738
```

computes, separately for the two values of smokingStatus, the sample proportion of women who are pregnant. Assuming these are the proportions we want, we may want to subtract them to obtain a point estimate. We can use the `diff()` command on the results from above:

```
diff(prop(~pregnant | smokingStatus, data=rawDataForTable2.11))

prop_yes.nonsmoker
        0.09789237
```

Notice that the result is obtained by subtracting the proportion on the left from the proportion on the right. That may, indeed, be precisely what we want. If, however, that is $\hat{p}_2 - \hat{p}_1$, and we wanted $\hat{p}_1 - \hat{p}_2$, we only need to multiply the result by (-1), which amounts to adding a minus sign:

```
-diff(prop(~pregnant | smokingStatus, data=rawDataForTable2.11))

prop_yes.nonsmoker
       -0.09789237
```

Using `resample()` with a grouping variable. Take note of the effect of the extra `gropus=smokingStatus` on the results for these commands:

```
tally(~pregnant | smokingStatus, data=resample(rawDataForTable2.11), margins=T)
tally(~pregnant | smokingStatus, data=resample(rawDataForTable2.11, groups=smokingStatus), margins=T)
```

An example bootstrapping on $\hat{p}_1 - \hat{p}_2$

Do take the time to check that

$$n_1 p_1 \geq 10, \quad n_1(1-p_1) \geq 10, \quad n_2 p_2 \geq 10, \quad n_2(1-p_2) \geq 10.$$

The raw data for Table 2.9, "Smoking Habits by Gender", can be built using rbind() and data.frame() commands as described above. But it already exists as two columns, `Smoke` and `Sex` in the Lock 5 data frame **StudentSurvey**.

```
tally(~Smoke | Sex, data=StudentSurvey)

     Sex
Smoke Female Male
  No     153  166
  Yes     16   27
```

Our point estimate for the difference $p_M - p_F$ of proportion of smokers amongst male and female populations is

```
diff(prop(~Smoke | Sex, data=StudentSurvey, success="Yes"))

prop_Yes.Male
   0.04522182
```

We generate many bootstrap statistics using

```
manyDiffsOfProps <- do(5000) * diff(prop(~Smoke | Sex,
  data=resample(StudentSurvey, groups=Sex), success="Yes"))
head(manyDiffsOfProps)

  prop_Yes.Male
1    0.03335684
2    0.09626882
3    0.05040316
4    0.03191587
5    0.08517031
6    0.03338750
```

and use it to find the approximate standard error

```
se <- sd(~prop_Yes.Male, data=manyDiffsOfProps)
se

[1] 0.03340241
```

so a 95% bootstrap confidence interval is

$$0.0452 \pm 2(0.0334), \qquad \text{or} \qquad [-0.022, 0.112].$$

An example bootstrapping on $\bar{x}_1 - \bar{x}_2$

The book, as well as notes found here `https://rstudio.calvin.edu:3939/content/54/#section-bootstrapping-a-difference-across-groups`, carry out an example of using bootstrapping to generate a confidence interval for $\mu_1 - \mu_2$, where Populations 1 and 2 are females and males, and the quantitative variable is number of hours spent exercising per week. The relevant data comes from columns `Exercise` (response variable) and `Gender` (explanatory variable) in the Lock 5 data frame **ExerciseHours**.

Here, my aim is to explain very little but give the basic commands that lead to the result.

We obatin the point estimate $\bar{x}_M - \bar{x}_F$:

```
ptEst <- diff( mean( ~Exercise | Gender, data=ExerciseHours ) )
ptEst

M
3
```

Next we generate a bootstrap distribution:

```
manyDiffsSampleMean <- do(5000) * diff( mean( ~Exercise | Gender,
                          data=resample(ExerciseHours, groups=Gender) ) )
head(manyDiffsSampleMean)

         M
1 8.133333
2 3.683333
3 3.533333
4 3.683333
5 1.050000
6 3.783333
```

We obtain the approximate standard error from this

```
stdEr <- sd(~M, data=manyDiffsSampleMean)
stdEr

[1] 2.340221
```

Then a 95% confidence interval for the difference $\mu_M - \mu_F$ is

$$3 \pm 2(2.340), \qquad \text{or} \qquad [-1.68, 7.68]$$