

# Answers to Application Activities for Manipulating Variables in R

---

## Deleting Parts of a Dataset Practice Activities

### 1 Working with rows and columns

- a. `sw[1:3]` #shows first 3 columns, all 5 rows
- b. `sw[1:3,]` #shows first 3 rows, all 4 columns
- c. `sw[,1:3]` #shows first 3 columns, all 5 rows
- d. `sw[4:5, 1:3]` #shows 4th and 5th rows, columns 1thru 3
- e. `sw[[1]]` #shows the data for the first column, all rows (but no headings on columns)
- f. `sw[1, ]` #shows the data for the first row, all columns, with column headings
- g. `sw["C", ]` #shows data for the row that starts with a "C" (which is Courtelary), all columns

### 2 SW dataset

```
sw$Examination<-NULL
```

```
sw #examine the dataset again to make sure this column is gone
```

### 3 ChickWeight

```
length(ChickWeight$diet)
```

```
[1] 578 #Original number of entries
```

```
ChickDiet<-subset(ChickWeight, subset=diet!=4)
```

```
length(ChickDiet$diet)
```

[1] 460 #Number that are left after excluding those with Diet #4

## Combining or Recalculating Variables Practice Activities

### 1 Torres dataset

If you haven't already, import **torres** dataframe into R using R Commander sequence DATA >

IMPORT DATA > FROM SPSS DATA SET and name this file **torres**.

DATA > MANAGE VARIABLES IN ACTIVE DATA SET > COMPUTE NEW VARIABLE

In the box "Expression to compute" put:  $(\text{listening} + \text{speaking})/2$ .

In the box "New variable name" put "AuralPreferences."

Here is the R code for this:

```
torres$AuralPreferences <- with(torres, (listening+speaking)/2)
```

Use the "View data set" button on R Commander to verify this column was added.

```
range(torres$AuralPreferences)
```

```
[1] 1 5
```

### 2 McGuire dataset

In R Commander, click on the "Data set" button to choose the mcguireSR dataset as the active dataset (or import the data file if you have not done so yet).

Next choose DATA > MANAGE VARIABLES IN ACTIVE DATA SET > COMPUTE NEW VARIABLE.

In the box "Expression to compute" put:  $\text{Post.test} - \text{Pre.test}$ .

In the box "New variable name" put "gainscore."

Here is the R code for this:

```
mcguireSR$gainscore <- with(mcguireSR, Post.test-Pre.test)
```

```
mean(mcguireSR$gainscore)
```

```
[1] 11.35053
```

### 3 LarsonHall.Partial dataset

DATA > MANAGE VARIABLES IN ACTIVE DATA SET > COMPUTE NEW VARIABLE

In the box “Expression to compute” put:  $(r\_l\_accuracy * 100) / 7$ .

In the box “New variable name” put “PercentAccuracy.”

Here is the R code for this:

```
partial$PercentAccuracy <- with(partial, (r_l_accuracy*100)/7)
```

```
max(partial$PercentAccuracy)
```

```
[1] 68.03571
```

## Recoding group boundaries practice activities

### 1 BEQ dataset

DATA > MANAGE VARIABLES IN ACTIVE DATASET > RECODE VARIABLES; Select the

`numberoflang` variable. Put in new variable name `Glots`.

In box labeled “Enter recode directives” write:

```
"Two"="Minorglots"
```

```
"Three"="Minorglots"
```

```
"Four"="Majorglots"
```

```
"Five"="Majorglots"
```

Note that these directives must be written EXACTLY as shown here, with the same capitalization and punctuation! Make sure the quotation marks are not smart quotes, but just straight ones. Leave the “Make new variable a factor” box ticked.

R code for this is:

```
beq <- within(beq, {  
  Glots <- Recode(numberoflang,  
    "Two"="Minorglots"; "Three"="Minorglots"; "Four"="Majorglots"; "Five"="Majorglots";',  
    as.factor.result=TRUE)  
})
```

In the R console:

```
summary(beq$Glots)
```

```
Majorglots Minorglots
```

```
879    699
```

## 2 McGuire SR dataset

If you have forgotten how to get names for the dataset, the command is `names()`

```
mean(mcguireSR$Pre.test) #check what the mean is first in R console by using this
```

```
function
```

```
[1] 137.7547
```

In R Commander, DATA > MANAGE VARIABLES IN ACTIVE DATA SET > RECODE VARIABLES;

Select the `Pre.test` variable. Put in new variable name (`rate`).

In box labeled “Enter recode directives” write:

```
0:137.74="slow"
```

```
else="fast"
```

Leave the “Make new variable a factor” box ticked. The R code for this is:

```
mcguireSR <- within(mcguireSR, {
```

```
rate <- Recode(Pre.test, '0:137.74="slow"; else="fast";',
```

```
as.factor.result=TRUE)
```

```
})
```

Verify:

```
mcguireSR$rate
```

```
summary(mcguireSR$rate)
```

```
fast slow
```

```
9      10
```

### 3 Torres dataset

DATA > MANAGE VARIABLES IN ACTIVE DATA SET > RECODE VARIABLES. Select the **beginner** variable. Put in new variable name.

In box labeled “Enter recode directives” write:

```
1="strong"
```

```
2="moderate"
```

```
3="neutral"
```

```
4="moderate"
```

```
5="strong"
```

Leave “Make new variable a factor” box ticked. The R code for this is:

```
torres <- within(torres, {
```

```
  TypeOfPreference <- Recode(beginner,
```

```
    '1="strong"; 2="moderate"; 3="neutral"; 4="moderate"; 5="strong";',
```

```
    as.factor.result=TRUE)
```

```
})
```

To verify:

```
summary(torres$TypeOfPreference)
```

```
moderate neutral strong
```

```
61    28    13
```

## Getting data in the correct form for statistical tests practice activity

### 1 DeKeyser dataset

First make sure **dekeyser** is your active dataset!

In R Commander: DATA > ACTIVE DATA SET > SUBSET ACTIVE DATA SET

Click off “Include all variables” because I only want the score; select **gjtscore**

Subset expression is:

```
status=="Under 15" (has 15 entries)
```

```
status=="Over 15" (has 42 entries)
```

In R code:

```
DkCats1<-subset(dekeyser,subset=status=="Under 15", select=c(gjtscore))
```

```
DkCats2<-subset(dekeyser,subset=status=="Over 15", select=c(gjtscore))
```

### 2 Obarow.Story1.sav

In R Commander: DATA > ACTIVE DATASET > SUBSET ACTIVE DATASET

Click off “Include all variables” because I only want the gainscore; select **gnsc1.1**

In order to get the correct subset expression, first look at the levels of the Treatment variable

**levels(OSTory\$treatment)**

**[1] "No music No pics" "No music Yes pics"**

**[3] "Yes music No pics" "Yes music Yes pics"**

Subset expression is:

**treatment1=="No music Yes pics"; called this NoMusicYesPics as name of new dataset**

**(this has 17 entries)**

The R code to do this would be:

**NoMusicYesPic <- subset(OSTory, subset=treatment1=="No music Yes pics",  
select=c(gnsc1.1))**

\*Be sure to click on “Data set” on the R Commander window to get back to OSTory before  
subsetting again

**treatment1=="Yes music Yes pics"; name is YesMusicYesPics (17 entries)**

**treatment1=="Yes music No pics"; name is YesMusicNoPics (15 entries)**

**treatment1=="No music No pics"; name is NoMusicNoPics (15 entries)**



In R code:

```
YesMusicYesPics <- subset(OStory, subset= treatment1=="Yes music Yes pics",  
select=c(gnsc1.1))  
YesMusicNoPics <- subset(OStory, subset= treatment1=="Yes music No pics",  
select=c(gnsc1.1))  
NoMusicNoPics <- subset(OStory, subset= treatment1=="No music No pics",  
select=c(gnsc1.1))
```

### 3 Effort dataset

When you import this, it will have 10 rows and 4 columns.

In R Commander, choose Data > ACTIVE DATA SET > STACK VARIABLES IN ACTIVE DATA SET

Choose all 4 variables; change name for stacked dataset to **EffortLong**; change name for variable to **Score** and name for factor to **Condition**. Press OK and you will have a 2-column data frame.

`mean(EffortLong$Score)` #to get the average

```
[1] 16
```

In R code:

```
EffortLong <- stack(EffortLong[, c("MinusEffortMinusFeedback",  
"MinusEffortPlusFeedback", "PlusEffortMinusFeedback",  
"PlusEffortPlusFeedback")])
```

## 4 Arabic dataset

When you import this, it will have 10 rows and 5 columns.

In R Commander, choose DATA > ACTIVE DATA SET > STACK VARIABLES IN ACTIVE DATA SET

Choose all 4 variables; change name to **ArabicLong**; change name for variable to **Score** and name for factor to **Contrast**. Press OK and you will have a 2-column data frame.

```
mean(ArabicLong$Score)
```

```
[1] 6.35
```

In R code:

```
ArabicLong <- stack(Arabic[, c("dh_D", "GLOTTALSTOP_PHARYNGEALFRIC", "t_T",  
"th_dh")])
```

```
names(ArabicLong) <- c("Score", "Contrast")
```