Analyzing data from a memory study

William Revelle
Psychology Research Methods: Fall, 2007

Abstract

A brief discussion of the process of data analysis of the Roediger and McDermott study. This will be elaborated as the analyses proceeds. Although the analyses are done in R, the important concepts are the results.

Data analysis is a straight forward process of telling a truthful and coherent story about your data. You do not need to report every single data point, but you need to report the important components of the data (Bem, 2003). Here we do a number of analyses on the data from the class replication of the (Roediger & McDermott, 1995) using the R language (R Development Core Team, 2007) and the **psych** package (Revelle, 2007).

Although the analyses are done in R, it is much more important to understand the meaning of the results rather than the specific program used. In order to show the process of data analysis, we show more analyses than would normally be reported.

Preliminaries

1. Start R
2. Activate the **psych** package

```
> library(psych)
```

3. Get the data

There are several ways of doing this. In all cases, we assume that the file you are reading has "header" information on the first line. That is, it is assumed that the first line of the file has labels for each variable. Each label should be unique, and should not have spaces in it.

1. If you have loaded the **psych** package, you can also copy the data from another program (in this an Excel file) into your computer clipboard and then copy them into R using the `read.clipboard` function.

```
>my.data <- read.clipboard()
```

2. If the data are in a file on your computer they can be read directly. You can use the `file.choose` to locate the file and then read that file using either the `read.csv` (if it is a comma delimited file) or the `read.table` function.

```
>  file.name <- file.choose()
>  my.data <- read.csv(file.name)
```

3. If the data are on a web based server (as they are for this example), they can also be read directly specifying the *url* for the file. We get the data and then echo the first two and last two lines to see what they look like using `headtail`.

```
> file.name <- "http://personality-project.org/revelle/syllabi/205/rm.07.recall.csv"
> my.data <- read.csv(file.name)
> headtail(my.data, n = 2)
   cond w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 w11 w12 w13 w14 w15 total1 total2 total3
1     0  7  8  5  6  6  8  5  4  6   7   6   7   7   8   8      0      9     13
2     0  6  7  7  5  4  4  2  3  4   4   4   2   5   3   3      0      6      7
22    0  8  8  4  3  7  6  4  4  7   7   6   7   6   7   7      0     10     11
23    0  8  6  8  6  7  7  4  6  4   4   7   4   4   5   7      0      8     11
   total4 total5 total6 total7 total8 total9 total10 total11 total12 total13
1       0      0     12     13      0     12       0       0      13      13
2       0      0     11      9      0     11       0       0       5       8
22      0      0     12     12      0     11       0       0      12      12
23      0      0     12     11      0     12       0       0      11      10
   total14 total15 total16 tpr
1        0       0      13   0
2        0       0       6   3
22       0       0      11   1
23       0       0      12   2
```

Descriptive statistics

Basic descriptive statistics may be found by using the `describe` function. First we describe the number of people in each condition(cond), the words recalled by position (w 1 – w 15) and the number recalled on each list (total 1 – total 16), and then the total number of false words recalled (tpr).

```
> describe(my.data)

        var  n mean   sd median   mad min max range  skew kurtosis   se
cond      1 23 0.48 0.51      0  0.00   0   1     1  0.08    -2.08 0.11
w1        2 23 7.00 0.80      7  1.48   6   8     2  0.00    -1.50 0.17
w2        3 23 6.65 1.23      7  1.48   4   8     4 -0.76    -0.43 0.26
```

```
w3         4 23 6.09 1.41      6  1.48   4   8     4 -0.15    -1.45 0.29
w4         5 23 6.22 1.44      7  1.48   3   8     5 -0.62    -0.73 0.30
w5         6 23 5.57 1.44      6  1.48   2   7     5 -0.74    -0.40 0.30
w6         7 23 5.52 1.53      6  1.48   3   8     5 -0.15    -1.13 0.32
w7         8 23 4.78 2.00      5  2.97   2   8     6  0.12    -1.24 0.42
w8         9 23 5.09 1.95      5  1.48   1   8     7 -0.19    -0.93 0.41
w9        10 23 5.04 1.89      6  1.48   1   8     7 -0.44    -0.91 0.40
w10       11 23 4.87 2.22      5  1.48   1   8     7 -0.46    -1.05 0.46
w11       12 23 5.17 1.72      5  1.48   1   7     6 -0.66    -0.45 0.36
w12       13 23 5.04 1.43      5  1.48   2   7     5 -0.34    -0.87 0.30
w13       14 23 5.48 1.68      5  1.48   2   8     6 -0.41    -0.65 0.35
w14       15 23 6.22 1.51      6  1.48   3   8     5 -0.51    -0.57 0.31
w15       16 23 6.04 1.77      7  1.48   2   8     6 -0.72    -0.61 0.37
total1    17 23 4.35 4.72      0  0.00   0  11    11  0.17    -1.94 0.98
total2    18 23 4.52 4.55      6  7.41   0  11    11  0.06    -1.94 0.95
total3    19 23 6.26 5.79      8  7.41   0  13    13 -0.09    -1.93 1.21
total4    20 23 4.83 5.48      0  0.00   0  14    14  0.38    -1.66 1.14
total5    21 23 4.87 5.32      0  0.00   0  12    12  0.20    -1.92 1.11
total6    22 23 5.57 5.96      1  1.48   0  14    14  0.17    -1.95 1.24
total7    23 23 5.39 5.54      5  7.41   0  14    14  0.15    -1.84 1.16
total8    24 23 5.61 6.21      0  0.00   0  15    15  0.27    -1.81 1.30
total9    25 23 6.04 6.00      9  7.41   0  14    14 -0.01    -2.00 1.25
total10   26 23 4.70 5.57      0  0.00   0  13    13  0.43    -1.74 1.16
total11   27 23 6.09 6.65      0  0.00   0  15    15  0.19    -1.94 1.39
total12   28 23 5.13 5.39      5  7.41   0  14    14  0.24    -1.74 1.12
total13   29 23 5.65 5.78      7 10.38   0  15    15  0.15    -1.81 1.21
total14   30 23 4.83 5.97      0  0.00   0  15    15  0.54    -1.51 1.25
total15   31 23 4.57 5.32      0  0.00   0  15    15  0.52    -1.38 1.11
total16   32 23 6.00 5.81      6  8.90   0  15    15  0.09    -1.80 1.21
tpr       33 23 2.26 2.14      2  2.97   0   6     6  0.40    -1.39 0.45
```

Graphical presentations

A more useful way to treat these numbers is to describe them graphically. We do this in two figures, the first for the serial position effects of words 1 - 15 (Figure 1), and the second in the number recalled by list (Figure 2). In order to show the amount of variability, we plot the means and their 95% confidence intervals using the `error.bars` function.

The plot of recall by list (Figure 2) shows a great deal of variation, but this is because only half the participants responded on any one trial. We can rearrange the data slightly, choosing participants by condition, and showing the number recalled for the appropriate trials. We do this by using the `subset` function and selectively choosing the right lists.

```
> error.bars(my.data[, 2:16]/8, ylim = c(0, 1), ylab = "percent recalled",
+     xlab = "serial position", typ = "b", main = "Percent recalled as a function of serial
> abline(h = mean(my.data[, 33])/8)
> text(8, 0.2, "false recall")
```

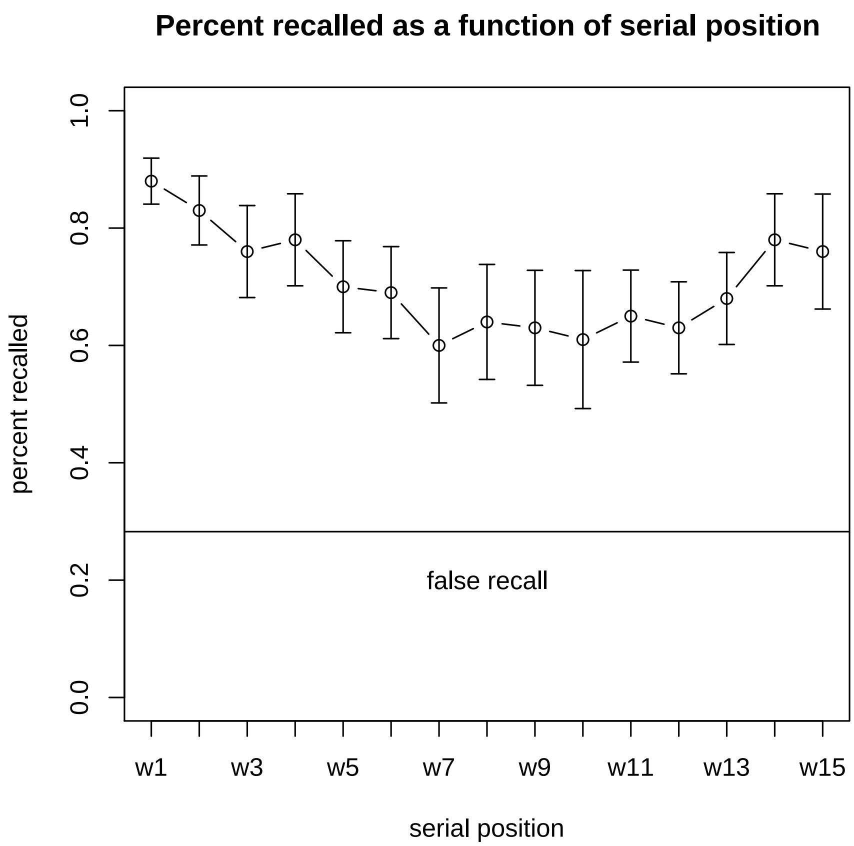

Figure 1. Recall as a function of serial position. The numbers are expressed as percentages by dividing by 8 (since only 1/2 of the trials did participants have an opportunity to recall). The existence of a serial position effect (a U shaped function higher at the beginning and end) suggests that the participants were following instructions, and that the memory performance was typical.

```
> error.bars(my.data[, 17:32], ylab = "numer recalled by list",
+      xlab = "list", typ = "b", main = "Number recalled as a function of list")
```

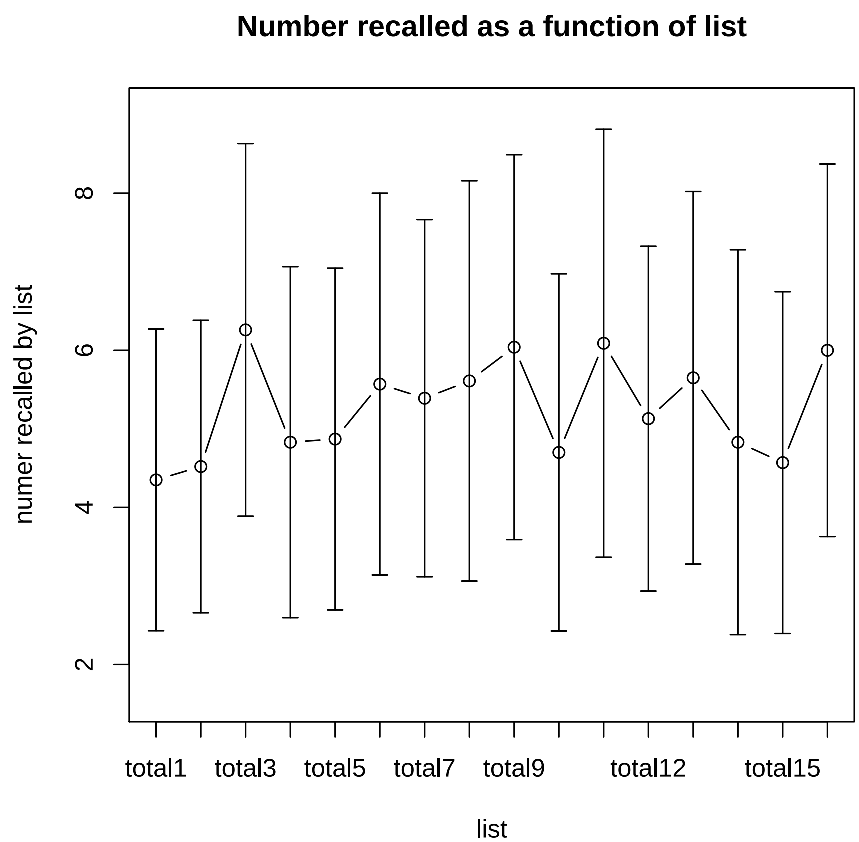

Figure 2. Number recalled by list. The extreme variability within each list is probably because 50% of the people were instructed not to recall on any particular list. Compare this to the next figure.

```
> condA <- subset(my.data[, c(1, 17, 20, 21, 24, 26, 27, 30, 31)],
+     my.data$cond == 1)
> condB <- subset(my.data[, c(1, 18, 19, 22, 23, 25, 28, 29, 32)],
+     my.data$cond == 0)
> describe(condA)

        var  n  mean   sd median  mad min max range  skew kurtosis   se
cond      1 11  1.00 0.00      1 0.00   1   1     0   NaN      NaN 0.00
total1    2 11  9.09 1.22      9 1.48   8  11     3  0.45    -1.57 0.37
total4    3 11 10.09 2.77     10 2.97   5  14     9 -0.19    -1.28 0.84
total5    4 11 10.18 1.66     10 1.48   7  12     5 -0.50    -1.09 0.50
total8    5 11 11.73 2.45     12 2.97   8  15     7 -0.16    -1.36 0.74
total10   6 11  9.82 3.60     11 1.48   2  13    11 -1.00    -0.50 1.09
total11   7 11 12.73 2.05     13 1.48   9  15     6 -0.80    -0.73 0.62
total14   8 11 10.09 4.48     11 4.45   0  15    15 -0.80    -0.33 1.35
total15   9 11  9.55 3.17      9 4.45   6  15     9  0.19    -1.61 0.96

> describe(condB)

        var  n  mean   sd median  mad min max range  skew kurtosis   se
cond      1 12  0.00 0.00    0.0 0.00   0   0     0   NaN      NaN 0.00
total2    2 12  8.67 1.50    9.0 1.48   6  11     5 -0.52    -0.72 0.43
total3    3 12 11.33 1.78   11.5 2.22   7  13     6 -1.00     0.27 0.51
total6    4 12 10.67 3.42   11.5 1.48   1  14    13 -1.78     2.47 0.99
total7    5 12 10.33 2.39   10.5 2.22   5  14     9 -0.51    -0.26 0.69
total9    6 12 11.58 1.38   11.5 0.74   9  14     5 -0.06    -0.84 0.40
total12   7 12  9.83 2.76   10.5 2.22   5  14     9 -0.29    -1.19 0.80
total13   8 12 10.83 2.37   11.0 2.97   7  15     8  0.03    -1.26 0.68
total16   9 12 11.08 2.54   11.5 2.22   6  15     9 -0.39    -0.79 0.73

> colnames(condA) <- colnames(condB) <- c("cond", "V1", "A1", "A2",
+     "V2", "A3", "V3", "V4", "A4")
> recall <- rbind(condA, condB)
> describe(recall)

     var  n  mean   sd median  mad min max range  skew kurtosis   se
cond   1 23  0.48 0.51      0 0.00   0   1     1  0.08    -2.08 0.11
V1     2 23  8.87 1.36      9 1.48   6  11     5 -0.30    -0.35 0.28
A1     3 23 10.74 2.34     11 2.97   5  14     9 -0.68    -0.48 0.49
A2     4 23 10.43 2.68     11 1.48   1  14    13 -1.83     4.13 0.56
V2     5 23 11.00 2.47     11 2.97   5  15    10 -0.30    -0.38 0.51
A3     6 23 10.74 2.77     11 1.48   2  14    12 -1.69     2.46 0.58
```

```
V3      7 23 11.22 2.81     12 2.97   5  15    10 -0.59    -0.71 0.59
V4      8 23 10.48 3.48     11 2.97   0  15    15 -0.99     1.17 0.72
A4      9 23 10.35 2.90     11 2.97   6  15     9 -0.17    -1.26 0.61
```

We can show the combined data graphically using the `error.bars.by` function. It does not appear that the lists differ across the experiment, although it is possible that the first list in each condition led to less recall (Figure 3).

```
> error.bars.by(recall[, 2:9]/15, recall$cond, ylim = c(0, 1),
+     ylab = "Percent recalled", main = "Recall by order")
```

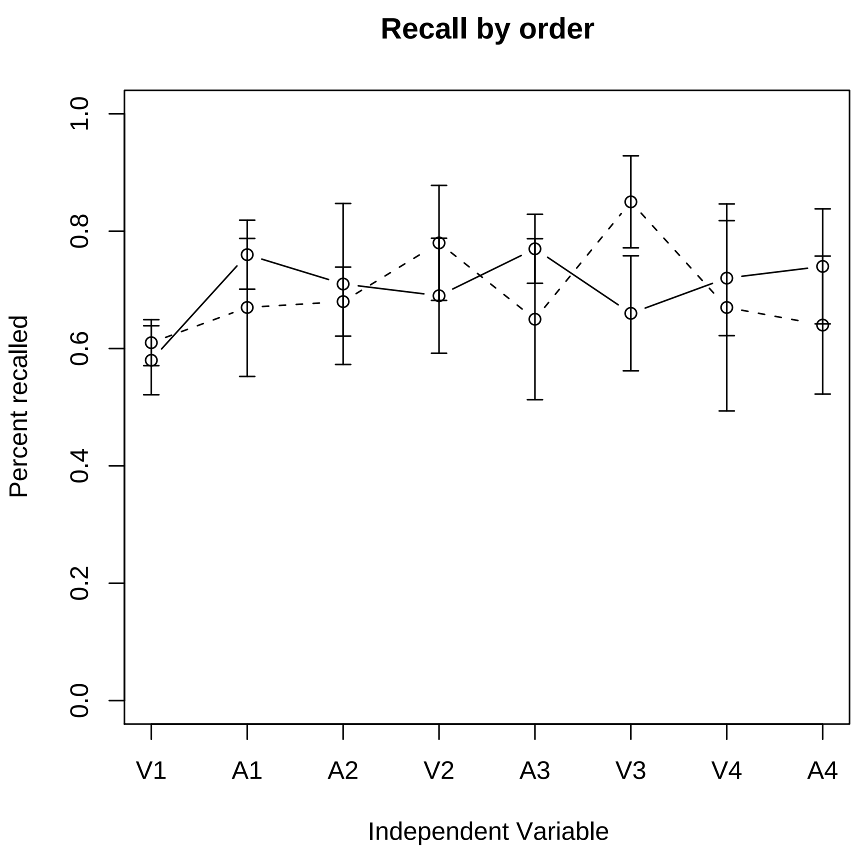

Figure 3. Recall by list number for two orders: A or B. Graphically it appears that the first trial for each condition is less than the others, but that the others do not differ.

Inferential Tests: Recall by modality

Did people recall more words as a function of the modality of presentation?
This first requires our examining how the stimuli were presented:

Table 1: The basic experimental design had two different Independent Variables: the modality of presentation and the instructions to recall or do math.

List	Modality (within)	A/B (between)
1	Visual	Recall
2	Visual	Math
3	Aural	Math
4	Aural	Recall
5	Aural	Recall
6	Aural	Math
7	Visual	Math
8	Visual	Recall
9	Aural	Math
10	Aural	Recall
11	Visual	Recall
12	Visual	Math
13	Visual	Math
14	Visual	Recall
15	Aural	Recall
16	Aural	Math

This information then entered into R and we can first show that the two experimental conditions are unrelated to each other, and unrelated to order. This merely means our design did not confound the two conditions.

```
> modality <- c(1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0)
> AB <- c(1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0)
> order <- 1:16
> design.df <- data.frame(order, modality, AB)
```

We now need to combine the data from the two conditions into a way that allows us to examine recall as a function of modality.

```
> visual <- recall$V1 + recall$V2 + recall$V3 + recall$V4
> aural <- recall$A1 + recall$A2 + recall$A3 + recall$A4
> t.test(visual, aural, paired = TRUE)
```

```
> pairs.panels(design.df)
```

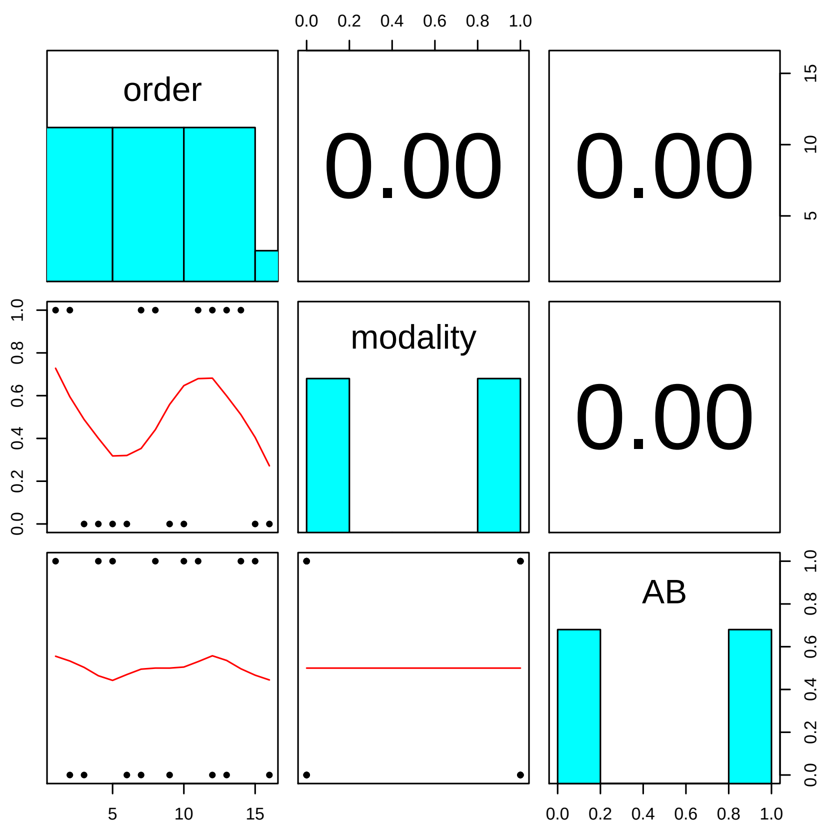

Figure 4. There is no correlation between presentation order or the conditions.

```
        Paired t-test

data:  visual and aural
t = -0.4864, df = 22, p-value = 0.6315
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.661506  2.270202
sample estimates:
mean of the differences
             -0.6956522
```

```
> mean(visual)

[1] 41.56522

> mean(aural)

[1] 42.26087
```

The t-test suggests that there was no difference between these two groups (t= .49, df=22). We can see this graphically by examining the `boxplot` of the data (Figure 5) as well as showing individual data points using the `stripchart` function.

We can also show these results in terms of the means and the 95% confidence regions (the mean +/- 1.96* s.e. where the standard error (s.e.) (Figure 6).

$$s.e. = \sigma/\sqrt{n} \tag{1}$$

Some data manipulation

In this section we show how it is possible to combine different variables to form new variables. What is missing from this graph is the number of false recalls. We can figure these out from the complete data set. Note that the names we use in this large file were chosen for convenience, rather than readability. In general, names that only you will see can be abbreviations, but if you are going to show a graph or a table, the data names should be meaningful.

```
>all.data.file <- file.choose() #find the file or, if it is on the sever
>all.data.file <- "http://personality-project.org/revelle/syllabi/205/205.rm.corrected.csv"
```

```
> all.data <- read.csv(all.data.file)
> names(all.data)
```

```
> all.df <- data.frame(visual, aural)
> boxplot(all.df/60, ylim = c(0, 1), ylab = "Number recalled",
+     main = "Recall by mode of presentation")
> stripchart(all.df/60, method = "jitter", vertical = TRUE, add = TRUE,
+     jitter = 0.05)
```

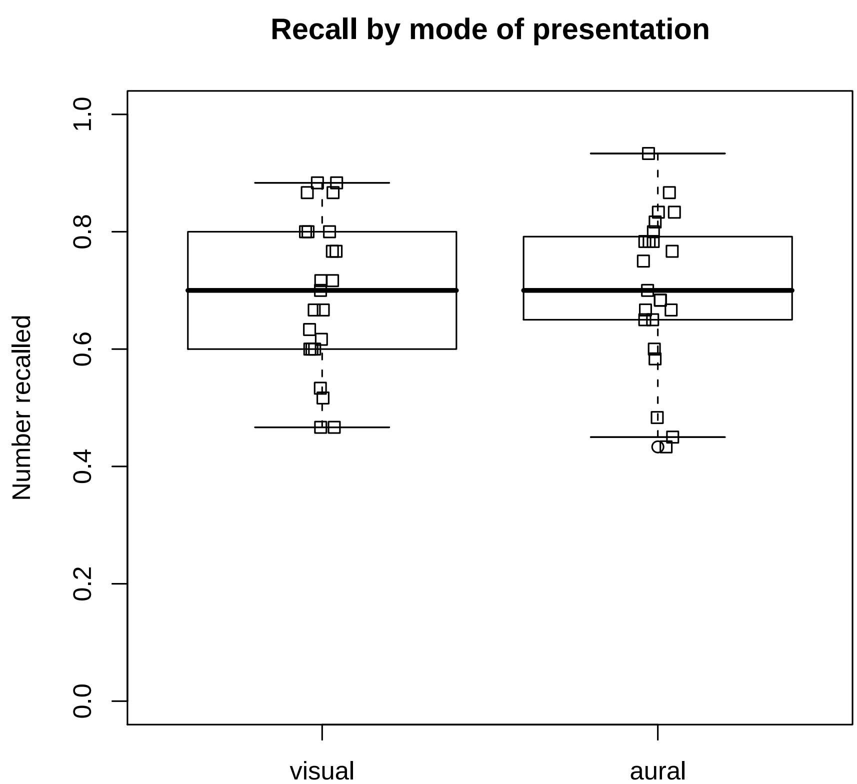

Figure 5. A boxplot of the mean percent recalled as a function of condition. We show individual data points by using the stripchart function.

```
> error.bars(all.df/60, ylim = c(0, 1), ylab = "Percent recalled",
+     xlab = "Presentation Condition", main = "Recall by mode of presentation")
```

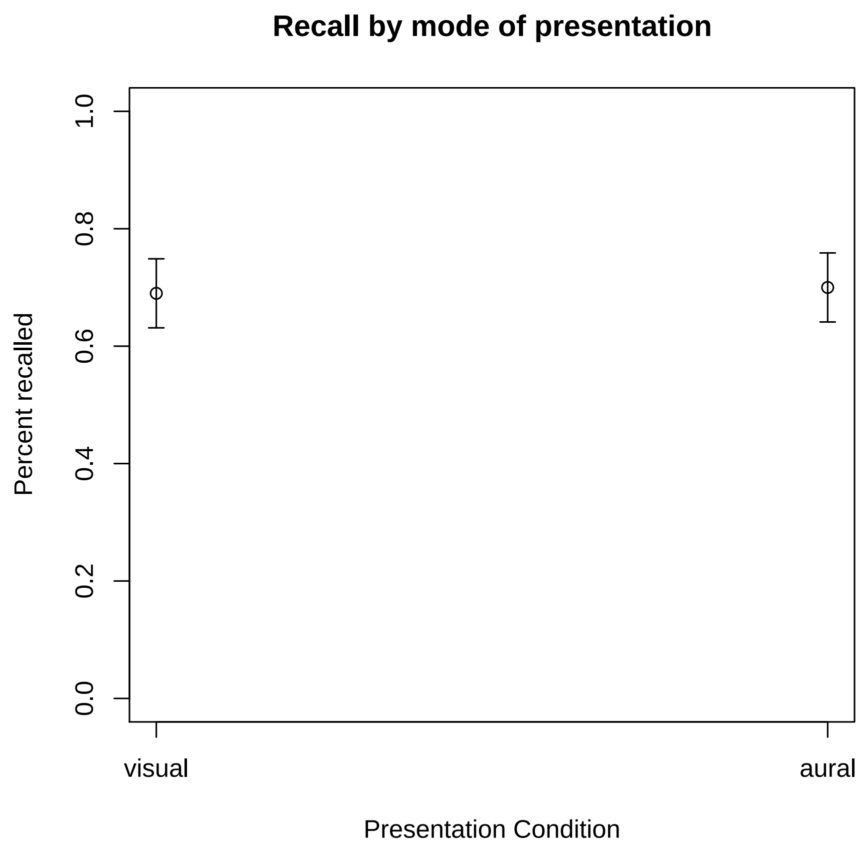

Figure 6. Yet another way of showing the (non) effect of mode of presentation upon recall. Mean percent recalled is shown for each condition, with 95% confidence regions (1.96 * s.e.). Note that some prefer to show confidence regions of 1.0 s.e.

```
  [1] "PID"     "cond"    "w1"      "w2"      "w3"      "w4"      "w5"
  [8] "w6"      "w7"      "w8"      "w9"      "w10"     "w11"     "w12"
 [15] "w13"     "w14"     "w15"     "total1"  "total2"  "total3"  "total4"
 [22] "total5"  "total6"  "total7"  "total8"  "total9"  "total10" "total11"
 [29] "total12" "total13" "total14" "total15" "total16" "tpr"     "al1_r1"
 [36] "al1_r2"  "al1_r3"  "al1_r4"  "al2_r1"  "al2_r2"  "al2_r3"  "al2_r4"
 [43] "al3_r1"  "al3_r2"  "al3_r3"  "al3_r4"  "al4_r1"  "al4_r2"  "al4_r3"
 [50] "al4_r4"  "al5_r1"  "al5_r2"  "al5_r3"  "al5_r4"  "al6_r1"  "al6_r2"
 [57] "al6_r3"  "al6_r4"  "al7_r1"  "al7_r2"  "al7_r3"  "al7_r4"  "al8_r1"
 [64] "al8_r2"  "al8_r3"  "al8_r4"  "al9_r1"  "al9_r2"  "al9_r3"  "al9_r4"
 [71] "al10_r1" "al10_r2" "al10_r3" "al10_r4" "al11_r1" "al11_r2" "al11_r3"
 [78] "al11_r4" "al12_r1" "al12_r2" "al12_r3" "al12_r4" "al13_r1" "al13_r2"
 [85] "al13_r3" "al13_r4" "al14_r1" "al14_r2" "al14_r3" "al14_r4" "al15_r1"
 [92] "al15_r2" "al15_r3" "al15_r4" "al16_r1" "al16_r2" "al16_r3" "al16_r4"
 [99] "pl1_r1"  "pl1_r2"  "pl1_r3"  "pl1_r4"  "pl2_r1"  "pl2_r2"  "pl2_r3"
[106] "pl2_r4"  "pl3_r1"  "pl3_r2"  "pl3_r3"  "pl3_r4"  "pl4_r1"  "pl4_r2"
[113] "pl4_r3"  "pl4_r4"  "pl5_r1"  "pl5_r2"  "pl5_r3"  "pl5_r4"  "pl6_r1"
[120] "pl6_r2"  "pl6_r3"  "pl6_r4"  "pl7_r1"  "pl7_r2"  "pl7_r3"  "pl7_r4"
[127] "pl8_r1"  "pl8_r2"  "pl8_r3"  "pl8_r4"  "pl9_r1"  "pl9_r2"  "pl9_r3"
[134] "pl9_r4"  "pl10_r1" "pl10_r2" "pl10_r3" "pl10_r4" "pl11_r1" "pl11_r2"
[141] "pl11_r3" "pl11_r4" "pl12_r1" "pl12_r2" "pl12_r3" "pl12_r4" "pl13_r1"
[148] "pl13_r2" "pl13_r3" "pl13_r4" "pl14_r1" "pl14_r2" "pl14_r3" "pl14_r4"
[155] "pl15_r1" "pl15_r2" "pl15_r3" "pl15_r4" "pl16_r1" "pl16_r2" "pl16_r3"
[162] "pl16_r4" "ta_r1"   "ta_r2"   "ta_r3"   "ta_r4"   "tp_r1"   "tp_r2"
[169] "tp_r3"   "tp_r4"   "Foils"
```

```
> falserecall <- matrix(NA, 23, 17)
> falserecall[, 1] <- all.data$cond
> for (i in 1:16) {
+     falserecall[, i + 1] <- all.data[, (94 + i * 4 + 1)] + all.data[,
+         (94 + i * 4 + 2)]
+ }
> realrecall <- matrix(NA, 23, 16)
> for (i in 1:16) {
+     realrecall[, i] <- all.data[, (30 + i * 4 + 1)] + all.data[,
+         (30 + i * 4 + 2)]
+ }
> falseA <- subset(falserecall[, c(1, 2, 5, 6, 9, 11, 12, 15, 16)],
+     all.data$cond == 1)
> falseB <- subset(falserecall[, c(1, 3, 4, 7, 8, 10, 13, 14, 17)],
+     all.data$cond == 0)
```

```
> colnames(falseA) <- colnames(falseB) <- c("cond1", "FV1", "FA1",
+     "FA2", "FV2", "FA3", "FV3", "FV4", "FA4")
> false.recall <- rbind(falseA, falseB)
> all.recall <- data.frame(recall[, 2:9]/15, false.recall[, 2:9])
> falseAB <- matrix(NA, 23, 8)
> describe(all.recall)
```

	var	n	mean	sd	median	mad	min	max	range	skew	kurtosis	se
V1	1	23	0.59	0.09	0.60	0.1	0.40	0.73	0.33	-0.30	-0.35	0.02
A1	2	23	0.72	0.16	0.73	0.2	0.33	0.93	0.60	-0.68	-0.48	0.03
A2	3	23	0.70	0.18	0.73	0.1	0.07	0.93	0.87	-1.83	4.13	0.04
V2	4	23	0.73	0.16	0.73	0.2	0.33	1.00	0.67	-0.30	-0.38	0.03
A3	5	23	0.72	0.18	0.73	0.1	0.13	0.93	0.80	-1.69	2.46	0.04
V3	6	23	0.75	0.19	0.80	0.2	0.33	1.00	0.67	-0.59	-0.71	0.04
V4	7	23	0.70	0.23	0.73	0.2	0.00	1.00	1.00	-0.99	1.17	0.05
A4	8	23	0.69	0.19	0.73	0.2	0.40	1.00	0.60	-0.17	-1.26	0.04
FV1	9	23	0.26	0.45	0.00	0.0	0.00	1.00	1.00	1.02	-1.00	0.09
FA1	10	23	0.22	0.42	0.00	0.0	0.00	1.00	1.00	1.28	-0.37	0.09
FA2	11	23	0.26	0.45	0.00	0.0	0.00	1.00	1.00	1.02	-1.00	0.09
FV2	12	23	0.17	0.39	0.00	0.0	0.00	1.00	1.00	1.61	0.62	0.08
FA3	13	23	0.13	0.34	0.00	0.0	0.00	1.00	1.00	2.05	2.32	0.07
FV3	14	23	0.04	0.21	0.00	0.0	0.00	1.00	1.00	4.19	16.26	0.04
FV4	15	23	0.22	0.42	0.00	0.0	0.00	1.00	1.00	1.28	-0.37	0.09
FA4	16	23	0.26	0.45	0.00	0.0	0.00	1.00	1.00	1.02	-1.00	0.09

True and False Recall as a function of experimental condition

Does the likelihood of recalling (either true recall or false recall) depend upon the way the stimuli were presented? That is, is one more likely to recall if the word lists were seen or heard? Does it make a difference for false recall? Do the differences differ? All of these questions may be addressed by an analysis of variance (ANOVA) with repeated measures (the two conditions and the two types of scores).

Unfortunately, doing a repeated measures ANOVA is less than transparent (at least at first). We need to reorganize the data so that we have the Dependent Variable (DV) in one column and the Independent Variables (IVs) in different columns. We do this by first "stacking" the data using the `stack` function, and then specifying the various values of the IVs by creating some labeling variables. Once we do this, we can show what the rearranged data look like using `headtail` and then proceed with the ANOVA.

```
> stacked.recall <- stack(all.recall)
> subj <- rep(paste("S", 1:23), 8)
> RF <- c(rep("R", 23 * 8), rep("F", 23 * 8))
```

```
> modality <- rep(c(rep("V", 23), rep("A", 46), rep("V", 23), rep("A",
+     23), rep("V", 46), rep("A", 23)), 2)
> recall.df <- data.frame(recall = stacked.recall$values, subj,
+     RF, modality)
> headtail(recall.df)

       recall subj RF modality
1   0.6000000  S 1  R        V
2   0.7333333  S 2  R        V
3   0.5333333  S 3  R        V
4   0.5333333  S 4  R        V
5   0.5333333  S 5  R        V
6   0.6666667  S 6  R        V
363 0.0000000 S 18  F        A
364 0.0000000 S 19  F        A
365 1.0000000 S 20  F        A
366 0.0000000 S 21  F        A
367 0.0000000 S 22  F        A
368 1.0000000 S 23  F        A

> aov.recall <- aov(recall ~ RF * modality + Error(subj/(RF * modality)),
+     data = recall.df)
> print(model.tables(aov.recall, "means"), digits = 3)

Tables of means
Grand mean

0.4471014

 RF
RF
    F     R
0.196 0.699

 modality
modality
    A     V
0.461 0.433

 RF:modality
   modality
RF  A     V
```

```
  F 0.217 0.174
  R 0.704 0.693

> summary(aov.recall)

Error: subj
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 22 3.5291  0.1604

Error: subj:RF
          Df  Sum Sq Mean Sq F value    Pr(>F)
RF         1 23.2674 23.2674  80.298 8.544e-09 ***
Residuals 22  6.3748  0.2898
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: subj:modality
          Df  Sum Sq Mean Sq F value Pr(>F)
modality   1 0.06976 0.06976  1.0036 0.3273
Residuals 22 1.52913 0.06951

Error: subj:RF:modality
            Df  Sum Sq Mean Sq F value Pr(>F)
RF:modality  1 0.02338 0.02338  0.2092 0.6519
Residuals   22 2.45884 0.11177

Error: Within
           Df  Sum Sq Mean Sq F value Pr(>F)
Residuals 276 20.8733  0.0756
```

How do we read the above output? What should we look for? First examine the means for the various conditions. The Grand Mean (.45) is the percentage of all words that were recalled. But this mean combines two very different types of recall, True recall of words that were presented (mean = .70) and False recall of words that were not presented (mean = .20). In addition, there were two ways of perceiving the words, aural or visual. The modality of presentation could have interacted (it did not) with the type of word. This is seen in the four means of modality by type of word table.

Once we examine the means, then we can ask whether the observed difference is likely to have happened by chance. That is, if there really were no difference between the two types of words or two ways of presenting them, then what is the likelihood of observing a difference as big as we do (or bigger) by chance? Here we compare the variability between the conditions (the Between conditions Sums of Squares and the associated variance estimate or Mean Squares) to the estimate of variance found within the conditions (the Residual or error Sums of Squares and associated Mean Squares.). If the hypothesis of no difference is correct, these two estimates of variance should be roughly equally. To the extent that they are not, then we reject the hypothesis of no difference.

The difference between the two types of words (the RF comparison) shows that the Real words were recalled much more (.70) than were false words (.20). This difference was very reliable (statistically significant) with an F (1,22) of 80.3 with a $p < .001$. When reporting F (the ratio of between to within variances), it is necessary to report the degrees of freedom between the conditions (in this case, 1) and the degrees of freedom within conditions (in this case, 22). The associated probability value is usually given by the computer program (in this case, it is $8.5 * 10^{-9}$) .

Recall did not differ as a function of the two modes of presentation (visual = .43, auditory = .46), F (1,22) = 1.00 $p < .33$. Nor was there a reliable interaction between mode of presentation and type of word (F <1).

Are there individual differences in recall ability?

An interesting analysis, not done by Roediger and McDermott (1995) is to examine the correlation of the recall scores across conditions. This examines whether there are reliable individual differences in recalling the word lists. We can do the same for the false recall. By finding the average correlation within each set, we can also find an estimate of the reliability of the true recall as well as the false recall. Finally, we can find whether true and false recall are correlated.

```
> round(cor(recall), 2)
```

```
      cond   V1    A1    A2   V2    A3   V3    V4    A4
cond  1.00 0.16 -0.27 -0.09 0.29 -0.33 0.53 -0.11 -0.27
V1    0.16 1.00  0.27  0.54 0.49  0.24 0.42  0.37  0.40
A1   -0.27 0.27  1.00  0.14 0.25  0.68 0.44  0.40  0.68
A2   -0.09 0.54  0.14  1.00 0.31  0.25 0.14  0.15  0.14
V2    0.29 0.49  0.25  0.31 1.00  0.45 0.58  0.67  0.34
A3   -0.33 0.24  0.68  0.25 0.45  1.00 0.21  0.44  0.48
V3    0.53 0.42  0.44  0.14 0.58  0.21 1.00  0.20  0.34
V4   -0.11 0.37  0.40  0.15 0.67  0.44 0.20  1.00  0.67
A4   -0.27 0.40  0.68  0.14 0.34  0.48 0.34  0.67  1.00
```

```
> total.recall <- rowSums(recall[, 2:9])
> alpha.recall <- alpha.scale(total.recall, recall[, 2:9])
> round(alpha.recall, 2)
```

```
[1] 0.82
```

```
> round(cor(false.recall), 2)

      cond1   FV1   FA1   FA2   FV2   FA3   FV3   FV4   FA4
cond1  1.00  0.42  0.34  0.42 -0.21 -0.11 -0.20 -0.08  0.22
FV1    0.42  1.00  0.41  0.77  0.25 -0.23 -0.13  0.17  0.55
FA1    0.34  0.41  1.00  0.41 -0.24  0.11  0.40 -0.02  0.65
FA2    0.42  0.77  0.41  1.00  0.25 -0.23 -0.13 -0.07  0.55
FV2   -0.21  0.25 -0.24  0.25  1.00 -0.18 -0.10  0.04 -0.01
FA3   -0.11 -0.23  0.11 -0.23 -0.18  1.00  0.55 -0.20  0.06
FV3   -0.20 -0.13  0.40 -0.13 -0.10  0.55  1.00 -0.11  0.36
FV4   -0.08  0.17 -0.02 -0.07  0.04 -0.20 -0.11  1.00  0.17
FA4    0.22  0.55  0.65  0.55 -0.01  0.06  0.36  0.17  1.00

> total.false <- rowSums(false.recall[, 2:9])
> alpha.false <- alpha.scale(total.false, false.recall[, 2:9])
> round(alpha.false, 2)

[1] 0.53

> round(cor(total.false, total.recall), 2)

[1] -0.33
```

What does this mean? Individual differences in recall are very reliable (α = .82) but false recall is less so (α = .53). Interestingly enough, the ability to correctly recall is negatively correlated with the probability of false recall although this correlation is not reliably different from 0 (r -.33, t = -1.62, p = .12).

Discussion of Recall data

Although it is not surprising that participants recalled more words that were presented than were not presented (70% versus 20%, F(1,22)= 80.3, $p < .001$), that words were falsely recalled at all indicates that memory is not just a tape recording, but that memory is a reconstructive process. Unfortunately, although there were hints that the Auditory Modality led to slightly more recall than did the visual modality (46 % versus 43%, this difference is quite likely just a chance event ($p > .3$).

Recognition

What are the effect of the conditions on true versus false recognition? Rather than asking for recognition for all of the words that were presented, the recognition list included just words from the 1st, 8th, and 10th position on each list, as well as the (unpresented) cue word. Thus, we can examine the likelihood of being recognized if the word were presented or not presented as a function of the mode of presentation (Aural versus Visual) and whether the words had a chance to be recalled or not. Later analyses will also consider the effect of actually recalling a word on subsequent recognition.

It is clear that although false words were recognized less than real words, that false words were recognized considerably more than would be expected (Figure 7).

The probability of falsely recognizing "foils" or words that were not presented and were not associated with the list is very small (mean = .10).

```
> realrecog <- matrix(NA, 23, 16)
> for (i in 1:16) {
+     realrecog[, i] <- all.data[, (30 + i * 4 + 1)] + all.data[,
+         (30 + i * 4 + 3)]
+ }
> falserecog <- matrix(NA, 23, 16)
> for (i in 1:16) {
+     falserecog[, i] <- all.data[, (94 + i * 4 + 1)] + all.data[,
+         (94 + i * 4 + 3)]
+ }
> realrecog <- realrecog/3
> realrecall <- realrecall/3
> foils <- all.data[, 171]/32
```

We can test for the likelihood of this difference happening by chance by using the t-test.

```
> describe(recognize.df)

      var  n mean   sd median  mad  min  max range  skew kurtosis   se
real    1 23 0.80 0.11   0.81 0.09 0.56 0.96  0.40 -0.58    -0.61 0.02
false   2 23 0.42 0.24   0.44 0.19 0.00 0.94  0.94  0.13    -0.67 0.05
foils   3 19 0.10 0.14   0.06 0.09 0.00 0.47  0.47  1.48     1.05 0.03

> t.test(recognize.df$real, recognize.df$false, paired = TRUE)

        Paired t-test

data:  recognize.df$real and recognize.df$false
t = 6.4305, df = 22, p-value = 1.806e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2577420 0.5031275
sample estimates:
mean of the differences
              0.3804348
```

```
> recognize.df <- data.frame(real = rowSums(realrecog/16), false = rowSums(falserecog/16),
+     foils)
> boxplot(recognize.df, ylim = c(0, 1), ylab = "Probability of Recognition",
+     main = "Probability of real and false recognition")
> stripchart(recognize.df, method = "jitter", jitter = 0.05, vertical = TRUE,
+     add = TRUE)
```

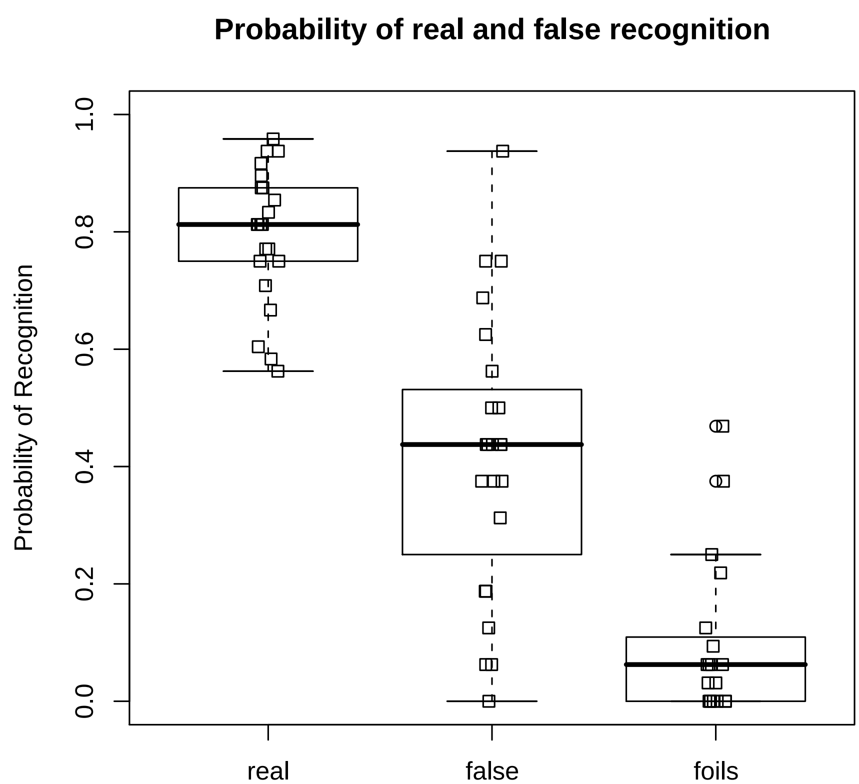

Figure 7. The probability of real and false recogntion. There are considerably larger individual differences in the probability of false rather than real recognition. Also included is the probability of recognizing the foils.

```
> t.test(recognize.df$false, recognize.df$foils, paired = TRUE)

        Paired t-test

data:  recognize.df$false and recognize.df$foils
t = 5.5849, df = 18, p-value = 2.666e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1939185 0.4277921
sample estimates:
mean of the differences
              0.3108553
```

Real words were recognized much more (mean=.79, sd=.12) than were false words (mean=.42, sd = .24), t (22) = 6.19, p < .001. Note that there is much more variability in the likelihood of recognition of the false rather than the true words (Figure 7). False words were recalled much more than were "foils" or words unassociated with the presented words (mean = .10, sd = .14), t(18) = 5.58, p <.001.

Analyzing the data as a function of trials shows this variation as well (Figure 8).

The effect of modality of presentation and of recall on recognition

There seems to be a great deal of variation in the recognition probabilities, particularly for the false words. We can examine the effects by condition by using an Analysis of Variance. Once again, this requires some reorganization of the data.

We first combine the two and false results into one data frame, then stack this dataframe so that recognition is the Dependent Variable and the conditions are the Independent Variables. To help see what we are doing, we use the `headtail` to show the first and last few rows of the original and then reorganized data.

```
> headtail(realrecog, n = 2)

          [,1]      [,2]      [,3]      [,4]      [,5] [,6]      [,7]
     0.6666667 0.3333333 1.0000000 0.6666667 0.6666667    1 0.6666667
     0.6666667 0.3333333 0.6666667 0.3333333 1.0000000    1 0.6666667
[22,] 1.0000000 1.0000000 0.6666667 1.0000000 0.6666667    1 0.6666667
[23,] 0.0000000 0.3333333 0.6666667 1.0000000 1.0000000    1 0.6666667
          [,8] [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
     1.0000000    1 0.3333333 0.6666667 1.0000000 0.6666667 0.6666667
     0.3333333    1 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667
[22,] 0.6666667    1 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667
[23,] 1.0000000    1 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667
```

```
> error.bars(realrecog, ylim = c(0, 1), typ = "b", ylab = "Percent recognized",
+     xlab = "trial #", main = "Correct and incorrect recognition")
> error.bars(falserecog, add = TRUE, type = "b", lty = "dashed")
```

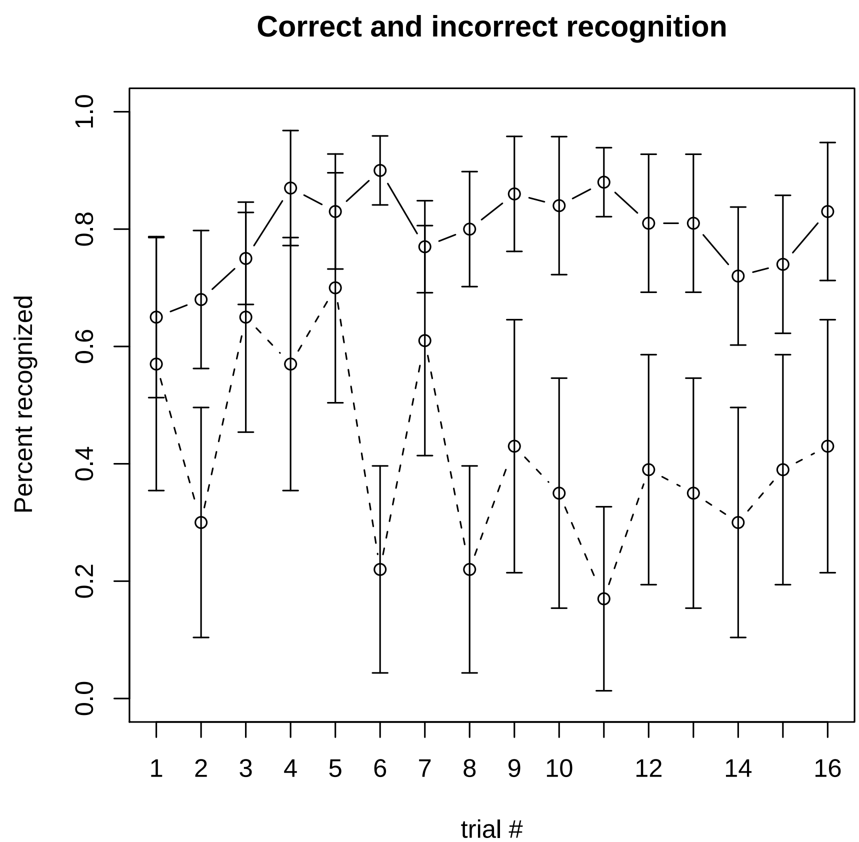

Figure 8. Probability of correct and incorrect recognition with 95% confidence bars. Because of the repeated measures, these confidence bars are overly broad.

```
          [,15]     [,16]
      0.6666667 1.0000000
      0.6666667 0.6666667
[22,] 0.6666667 1.0000000
[23,] 0.6666667 1.0000000
```

```
> headtail(falserecog, n = 2)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
         0    0    0    0    1    0    0    0    0     0     0     0     0
         1    0    0    1    1    0    1    0    0     0     0     1     1
[22,]    1    1    0    0    1    0    1    0    0     0     0     1     0
[23,]    0    0    1    0    1    0    1    1    1     0     0     0     1
      [,14] [,15] [,16]
          0     0     0
          0     1     0
[22,]     0     0     0
[23,]     1     0     1
```

```
> recog.df <- data.frame(realrecog, falserecog)
> headtail(recog.df, n = 2)
```

```
          X1        X2        X3        X4        X5 X6        X7        X8 X9
1  0.6666667 0.3333333 1.0000000 0.6666667 0.6666667  1 0.6666667 1.0000000  1
2  0.6666667 0.3333333 0.6666667 0.3333333 1.0000000  1 0.6666667 0.3333333  1
22 1.0000000 1.0000000 0.6666667 1.0000000 0.6666667  1 0.6666667 0.6666667  1
23 0.0000000 0.3333333 0.6666667 1.0000000 1.0000000  1 0.6666667 1.0000000  1
         X10       X11       X12       X13       X14       X15       X16 X1.1
1  0.3333333 0.6666667 1.0000000 0.6666667 0.6666667 0.6666667 1.0000000    0
2  0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667 0.6666667    1
22 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667 0.6666667 1.0000000    1
23 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667 0.6666667 1.0000000    0
   X2.1 X3.1 X4.1 X5.1 X6.1 X7.1 X8.1 X9.1 X10.1 X11.1 X12.1 X13.1 X14.1 X15.1
1     0    0    0    1    0    0    0    0     0     0     0     0     0     0
2     0    0    1    1    0    1    0    0     0     0     1     1     0     1
22    1    0    0    1    0    1    0    0     0     0     1     0     0     0
23    0    1    0    1    0    1    1    1     0     0     0     1     1     0
   X16.1
1      0
2      0
22     0
23     1
```

```
> stacked.recog <- stack(recog.df)
> subj <- rep(paste("S", 1:23), 16)
> RF <- c(rep("R", 23 * 16), rep("F", 23 * 16))
> modality <- rep(c(rep("V", 23), rep("A", 46), rep("V", 23), rep("A",
+     23), rep("V", 46), rep("A", 23)), 4)
> cond <- all.data[, 2]
> condition <- rep(cond, 32)
> recognition.df <- data.frame(recog = stacked.recog$values, subj,
+     RF, modality, condition)
> headtail(recognition.df)
```

```
        recog subj RF modality condition
1   0.6666667  S 1  R        V         0
2   0.6666667  S 2  R        V         0
3   0.0000000  S 3  R        V         1
4   0.3333333  S 4  R        V         0
5   1.0000000  S 5  R        V         1
6   0.6666667  S 6  R        V         0
731 1.0000000 S 18  F        A         1
732 0.0000000 S 19  F        A         0
733 1.0000000 S 20  F        A         0
734 1.0000000 S 21  F        A         0
735 0.0000000 S 22  F        A         0
736 1.0000000 S 23  F        A         0
```

```
> aov.recog <- aov(recog ~ RF * modality * condition + Error(subj/(RF *
+     modality)), data = recognition.df)
> print(model.tables(aov.recog, "means"), digits = 3)
```

```
Tables of means
Grand mean

0.6059783


 RF
RF
    F     R
0.416 0.796


 modality
modality
    A     V
```

```
0.600 0.612

 condition
condition
    0     1
0.605 0.607

 RF:modality
   modality
RF   A     V
  F 0.397 0.435
  R 0.803 0.790

 RF:condition
   condition
RF   0     1
  F 0.406 0.426
  R 0.804 0.788

 modality:condition
        condition
modality 0     1
       A 0.608 0.591
       V 0.602 0.623

 RF:modality:condition
, , condition = 0

   modality
RF   A     V
  F 0.406 0.406
  R 0.809 0.799

, , condition = 1

   modality
RF   A     V
  F 0.386 0.466
  R 0.795 0.780

> summary(aov.recog)
```

```
Error: subj
          Df  Sum Sq Mean Sq F value Pr(>F)
condition  1  0.0007  0.0007  0.0014 0.9705
Residuals 21 10.6983  0.5094

Error: subj:RF
             Df  Sum Sq Mean Sq F value    Pr(>F)
RF            1 26.6304 26.6304 39.6364 3.041e-06 ***
RF:condition  1  0.0589  0.0589  0.0877      0.77
Residuals    21 14.1092  0.6719
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Error: subj:modality
                   Df Sum Sq Mean Sq F value Pr(>F)
modality            1 0.0296  0.0296  0.1628 0.6906
modality:condition  1 0.0642  0.0642  0.3535 0.5585
Residuals          21 3.8159  0.1817

Error: subj:RF:modality
                      Df  Sum Sq Mean Sq F value Pr(>F)
RF:modality            1 0.11836 0.11836  0.9172 0.3491
RF:modality:condition  1 0.08153 0.08153  0.6318 0.4356
Residuals             21 2.70983 0.12904

Error: Within
           Df Sum Sq Mean Sq F value Pr(>F)
Residuals 644 81.861   0.127
```

Individual differences in recognition

Just as we examined the reliability and intercorrelation of the recall scores, so can we examine the reliability of recognition.

```
> total.true.recog <- rowSums(realrecog)
> total.false.recog <- rowSums(falserecog)
> alpha.true <- alpha.scale(total.true.recog, realrecog)
> alpha.false <- alpha.scale(total.false.recog, falserecog)
> round(alpha.true, 2)
```

```
[1] 0.68
```

```
> round(alpha.false, 2)
```

```
[1] 0.76

> cor.test(total.true.recog, total.false.recog)

        Pearson's product-moment correlation

data:  total.true.recog and total.false.recog
t = -0.8507, df = 21, p-value = 0.4045
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5531130  0.2483627
sample estimates:
       cor
-0.1825256
```

Discussion of recognition results

Participants were very able to recognize words that had been presented before (mean = 79%) but were also willing to report recognizing words that had not been presented, but were rather highly associated with the words that had been presented (mean = .42%). There were no reliable effects of mode of presentation (oral versus visual) or of having a chance to recall the words earlier. Nor did these two conditions interact with each other.

There were reliable individual differences in the ability recognize both true ($\alpha = .68$) and false ($\alpha = .76$) words, but these two were themselves not correlated (r = -.18, df = 21, p =.40).

A further analysis will examine the effect of prior recall upon subsequent recognition, as well as the strength of recognition of words in the middle of the list versus the beginning of the list.

Individual differences in recall and recognition

Does the ability to recall relate to success in recognition? We examine this by correlating total recall scores with true and false recognition scores (Figure 9).

```
> total.recall <- rowSums(all.data[18:33])
> recall.recog.df <- data.frame(true.recall = total.recall, false.recall = all.data[,
+     34], true.recog = total.true.recog, false.recog = total.false.recog)
```

Separating the effect of ability to recall from the chance to recall upon subsequent recognition

We have seen that people who are able to recall well seem to have very good recognition. But what about for those trials where they were not able to recall, did they still have good recognition? In order to answer this question, we need to consider the conditional probabilities of recognizing as a function of recall.

```
> pairs.panels(recall.recog.df)
```

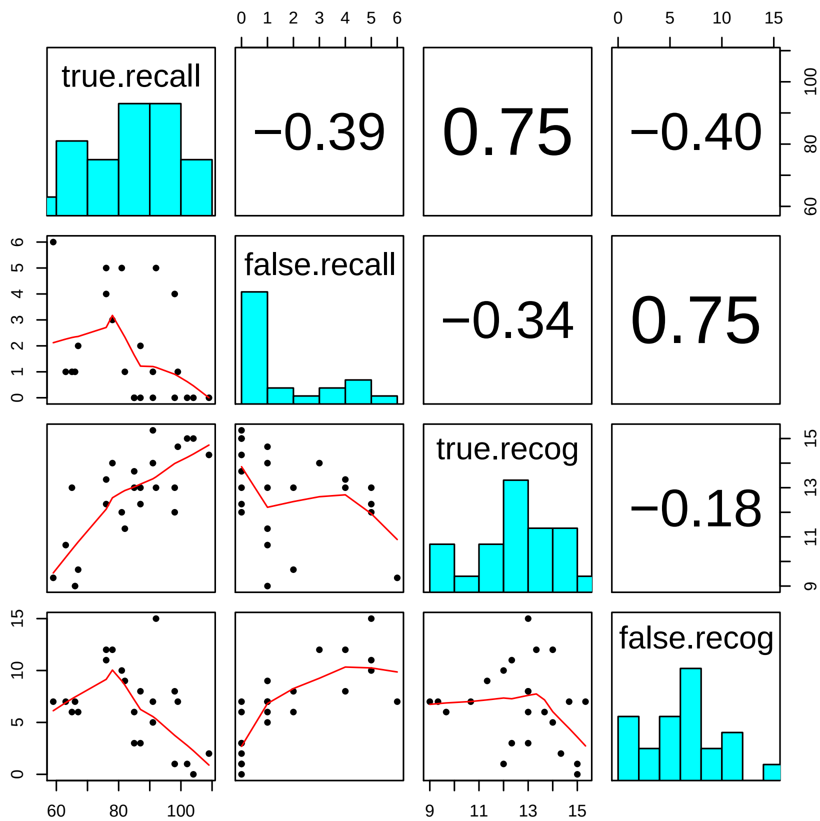

Figure 9. The relationship between true and false recall and true and false recognition.

Finding conditional probabilities

The conditional probability of recognizing given prior recall is found by considering the four cells in the recall and recognition data (Table 2):

Table 2: Estimating conditional probabilities

	Recognized	Did not Recognize	Total
Recall	A	B	A +B
Did not Recall	C	D	C+D
Total	A + C	B+D	A+B+C+D

The probability of recall in Table 2 is (A+B)/((A+B+C+D) and the probability of recognition is (A+C)/(A+B+C+D). But the conditional probability of recognition given prior recall is the ratio of A/(A+B). We can find this through some basic data manipulation:

```
> recog.recall.array <- array(NA, c(16, 4, 23))
> recog.recall.false <- array(NA, c(16, 4, 23))
> for (i in 1:16) {
+     for (j in 1:4) {
+         recog.recall.array[i, j, ] <- all.data[, 30 + i * 4 +
+             j]
+         recog.recall.false[i, j, ] <- all.data[, 94 + 4 * i +
+             j]
+     }
+ }
> recog.recall <- colSums(recog.recall.array, dims = 1)
> recog.recall.f <- colSums(recog.recall.false, dims = 1)
> condit.recog <- recog.recall[1, ]/(recog.recall[1, ] + recog.recall[2,
+     ])
> condit.recog.n <- recog.recall[3, ]/(recog.recall[3, ] + recog.recall[4,
+     ])
> condit.frecog <- recog.recall.f[1, ]/(recog.recall.f[1, ] + recog.recall.f[2,
+     ])
> condit.frecog.n <- recog.recall.f[3, ]/(recog.recall.f[3, ] +
+     recog.recall.f[4, ])
> conditional.df <- data.frame(real.rec = condit.recog, real.notrec = condit.recog.n,
+     false.rec = condit.frecog, false.notrec = condit.frecog.n)
> headtail(round(conditional.df, 2))
```

```
   real.rec real.notrec false.rec false.notrec
1      1.00        0.60       NaN         0.06
```

```
2       1.00        0.54      1.00          0.40
3       1.00        0.69      1.00          0.91
4       0.93        0.40      0.33          0.85
5       0.94        0.67       NaN          0.19
6       0.81        0.81       NaN          0.19
18      0.85        0.46      1.00          0.40
19      1.00        0.93       NaN          0.44
20      1.00        0.73      1.00          0.67
21      0.83        0.63      1.00          0.53
22      1.00        0.79      1.00          0.27
23      0.94        0.74      1.00          0.43
```

```
> describe(conditional.df)
```

```
             var  n mean   sd median  mad  min  max range  skew kurtosis   se
real.rec       1 23 0.93 0.08   0.96 0.06 0.75 1.00  0.25 -0.83    -0.67 0.02
real.notrec    2 23 0.69 0.16   0.69 0.18 0.36 0.93  0.56 -0.41    -0.96 0.03
false.rec      3 15 0.92 0.19   1.00 0.00 0.33 1.00  0.67 -2.04     2.91 0.05
false.notrec   4 23 0.37 0.24   0.38 0.23 0.00 0.91  0.91  0.56    -0.42 0.05
```

Does recall lead to recognition?

In the previous section (we found the conditional probabilities for recognition given prior recall. Doing this for both the true and false words, we find that words that were previously recalled were almost certain (mean = .93) to be recognized later. This is true for true words (mean = .93) as well as false words (mean = .92). There are a number of missing conditional probabilities for the false words, because eight participants recalled no false words, and thus the conditional probability of recognition given recall could not be calculated.

Recognition does depend upon prior recall, particularly for false (primed) words. If a true word was not recalled, it was recognized with a probability of .69, but a false word that was not recalled was recognized with a probability of .37 (Figure 10).

Discussion

The data analyses reported in this document are far more extensive than would be appropriate for a normal paper. The task of the writer of a scientific report is to know far more about the data than is needed to be reported. You should "own" the data (Bem, 2003). You should have analyzed it in many different ways so that you can understand what is going on and what are potential problems of interpretation.

Although these analyses were done using the R computer language, similar analyses could have been done using other common statistical packages (e.g., SPSS or SAS) or even,

```
> error.bars(conditional.df, ylim = c(0, 1), ylab = "conditional probability",
+     xlab = "Prior recall (or not)", main = "Conditional probability of recognition given
```

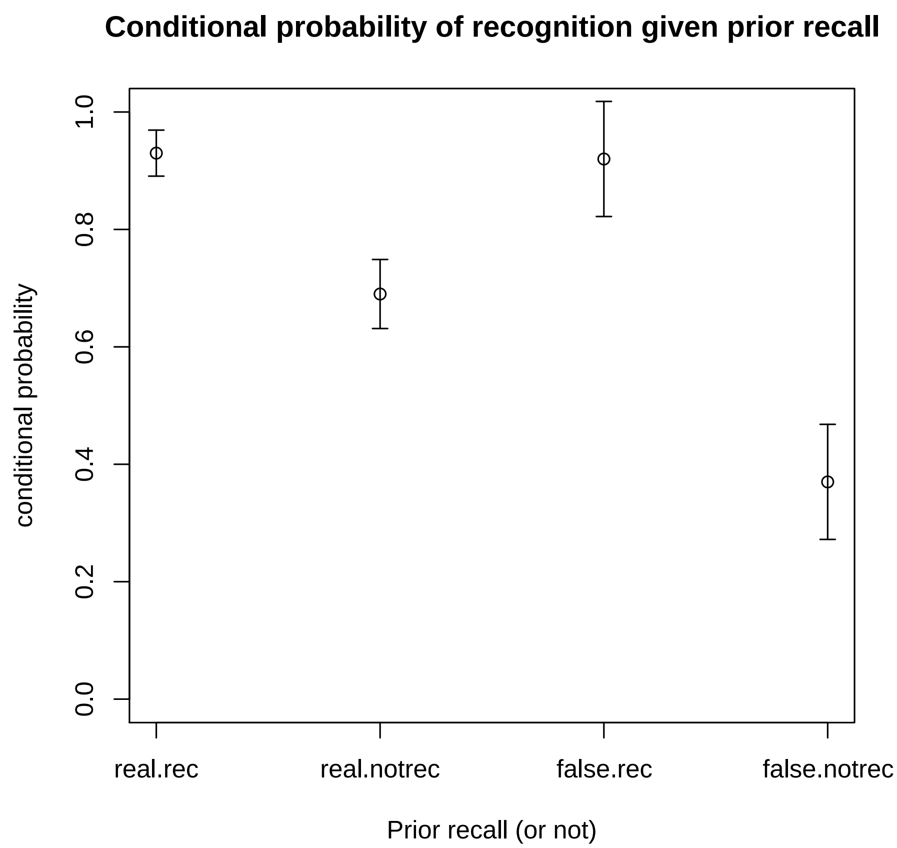

Figure 10. The conditional probability of recognition given prior recall shows that recalling a word leads to higher subsequent recognition for both real words and false (primed) words.

with some more work, a spreadsheet program such as EXCEL. The advantage of using R is that it is (relatively) easy to ask questions of the data that are not the commonly asked questions.

References

Bem, D. J. (2003). Writing the emperical journal article. In J. M. Darley, M. P. Zanna, & H. L. Roediger (Eds.), *The compleat academic: A practical guide for the beginning social scientist* (2nd ed.). Washington, D.C.: American Psychological Association.

R Development Core Team. (2007). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing. (ISBN 3-900051-07-0)

Revelle, W. (2007). *psych: Procedures for personality and psychological research.* (R package version 1.0-34)

Roediger, H. L., & McDermott, K. B. (1995). Creating false memories: Remembering words not presented in lists. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*(4), 803-814.