Psychology 252
Thomas & Monin

Homework #2 (10/1/14, due 10/8/14; submit **only #1, 6, 7, 9, 10 & 12** for grading.)

Readings: *Howell,* Chapters 3 & 7. Also, the lecture .ppt slides.
Data: 'hw2data.csv'. Most of the variables are the same as in the Class Project on memory bias: *Type* = 1, 2, 3 refers to 'free', 'biased' and 'varied' recall, respectively; *Complain* = 1, if you seriously considered complaining when you missed your plane/train, and *Complain* = 0, otherwise.
R scripts: Use the functions (leaving the parentheses empty) `getwd(), dir(), ls(),` and `search(),` to familiarize yourself with the R environment. Make sure you have a sense of the snapshot provided by each command. A relevant script is 'shw2sols.r'.

1. Here is a sample: 1, 1, 0. **Calculate by hand** the mean, $\bar{x}$; sum of squares, *SS*; variance, s_x^2; and standard deviation, s_x. **Calculate** also the standard error (s.e.) of $\bar{x}$.

2. **Importing '*.sav' files into R**.

If SPSS is installed on your computer, you will often wish to use the SPSS-formatted data files. One way is to save the '*.sav' file as a '*.csv' file, and use `read.csv()` to read it into R. Another way (especially if you don't own SPSS yourself to open the .sav file) is to use `read.spss()` from the `foreign` package. This means you need to download the `foreign` package, and then load it by using `library(foreign).` Then try `?read.spss` to learn about the function. One difficulty is going to be how to specify the path to get to the file once you've downloaded it from CourseWork. Note that `getwd()` tells you what the working directory is, `setwd(...)` enables you to reset it to wherever you want, and `dir()` tells you what's in the working directory. If you set the directory to the folder where you downloaded the file (or if you move the file to wherever your working directory is) then you can just use the name of the file without specifying the path – see examples in the script. Finally, don't forget to assign the data you read to an R object, using, for example,

`data0 = read...`, or
`data0<-read...`, or
`read...->data0`.

Here is one way to import the SPSS file (you can of course work from the CSV file instead if you prefer):

```
library(foreign)
?read.spss
# Relative reference to the file, in working directory
read.spss("hw2data.sav",use.value.labels=T,to.data.frame=T)->data0
# Absolute reference to the file, specifying the full path (obviously use your own path)
read.spss("C:/Users/Benoit/Desktop/OB 652/Week 2/hw2data.sav",
      use.value.labels=T, to.data.frame=T)->data0
```

If you plan to use this data in the future and don't want to go through this again, you can save your workspace using `save.image("filename.RData").` Again this will be put in the working directory, whatever it is. Make sure you do this at the end of your session. Once you have created your data frame `data0`, use the commands `str(…)` (for "structure"), `names(…)`, and `summary(…)` to examine it.

Important : If you ever get stuck with a command, make sure you try ?commandname and/or example(commandname) to let R help you. If you are at a loss what to use, but have a vague idea try ??vagueidea for a broader ("fuzzy") search.

3. **Plot and examine** the histograms of the variables in 'hw2data.csv' and, as appropriate, **recode** these variables into a small number of categories (e.g., 2-8 values or categories). Use the recoded variables in the analyses below.

Hints: In HO-1, Sec. 6.0.2, the `plot()` and `hist()` functions are used in the 'smem1a.r' script used in Week 1 and downloadable on CourseWork. To recode variables, use `findInterval()`, as in HW-1, #8(b). Also, `summary()` gives helpful summaries of the variables.

4. **Explain** any relationship (or lack thereof) between *Complain* and *Memory Group*.

Hints: An 'explanation' should go **beyond** mere description. You are expected to **recognize** that these variables are categorical and that, therefore, the chi-square contingency test is appropriate for deciding whether or not there is a significant relationship. `table()` and `chisq.test()` are relevant functions.

5. **Simple Correlation**. Ignoring *Type*, **calculate** the simple ('zero-order') **correlation** between *Past Happiness* and *Future Happiness*. Is this correlation **significant**? Plot *Future Happiness* (on the y-axis, because it is the dependent variable) against *Past Happiness*; is the relationship approximately **linear**, or is there a **non-linear** component?

Hints: The correlation coefficient, *r*, between 2 quantitative variables, *X* and *Y*, lies between -1 and +1, and is a measure of the strength of the linear relation between *X* and *Y*. To calculate the correlation in R, use, e.g., `cor(x, y)`. To test whether *r* is significantly different from 0, use, e.g., `cor.test(x, y)`. One can include more arguments in `cor.test(x, y)` (not needed here but good to know about): e.g., `cor.test(x, y, use = "pairwise.complete.obs", method = 'p')`, where 'p' = 'Pearson' and 's' = Spearman. The plotting function has already been introduced.

6. You may wonder if the relationship between *Past Happiness* and *Future Happiness* differs across memory group (*Type*). To examine this, **look at the correlation, and plot** *Future Happiness* **against** *Past Happiness*, **within each of the memory groups**. (See 'shw2sols.r' for a relevant script.)

Hints: A simple, brute force way to analyze the data for each level of *Type* would be to create 3 subsets of data, one for each level, and then analyze each subset. E.g.,

```
d0.f = d0[d0$Type==1,]
d0.b = d0[d0$Type==2,]
d0.v = d0[d0$Type==3,]

rs0 = lm(d0$Futurehapp ~ d0$Pasthapp, data=d0)
rs.f = lm(d0$Futurehapp ~ d0$Pasthapp, data=d0.f), etc.
```

A more elegant way to get the correlation, but not the p-value, is to use the function, `by()`:

```
rs2 = by(d0[, c(2,4)], Type, cor)
print(rs2)
```

However, replacing `cor` by `cor.test` leads to an error message; i.e., we can't get p-values this way. Yet another alternative would be to use a `for()` loop. We illustrate the `with()` function, although it isn't necessary here. E.g.,

```
pdf('hw2.pastfutplots2.pdf')
par(mfrow=c(2,2))

rs1 = list(length = 3)    # a placeholder for the output of the for() loop
for (i in levels(d0$Type)) {
     rs1[[i]] = cor.test(~ Pasthapp + Futurehapp, d0, Type==i, na.action=na.omit)
     with(d0[Type==i,], plot(Pasthapp, Futurehapp, main = i))
```

```
      plot(d0[Type==i, c(2,4)], type = 'p', main = i)
      lines(lowess(d0[Type==i, c(2,4)]), lty = 2)
      }
print(rs1)
```

7. Perform **a formal test of whether *Future Happiness* depends on *Memory Group***. [Use `lm(…)`, `summary(…)`, and compare the results to those obtained with `oneway.test(…)`] . Should you transform the data? [Consider using `boxplot(Futurehapp~Type)` to look at distributions.]

8. Use the **General Linear Model** to examine the dependence of *Future Happiness* on *Past Happiness,* Memory Group *(Type), Responsible* and *FTP*. Describe and explain the data.

9. Use **logistic regression** to examine the relation between *Responsible* and *Complain*. **Explain** any relationship (or lack thereof).

Hints: The general linear model, `lm()`, is used when the dependent variable, *Y*, is quantitative and Normally distributed, and is **linear**ly dependent on the predictor or independent variables. The predictor variables can be quantitative **or** categorical – hence the '**general**' in GLM. In this example, the dependent variable is Complain, *C [Why is C 'dependent'?]*, which is dichotomous with values 0 and 1. *C* has the Binomial distribution, not the familiar Normal distribution. The Binomial distribution is indexed by the parameter, Prob($C = 1$), which *is* quantitative and lies between 0 and 1. We model the relationship between the dependent variable, Prob($C = 1$), and the predictor variables by assuming that there is a specific, **non-linear transformation** of the dependent variable that is linearly dependent on the predictor or independent variables. This more general model is referred to as the **generalized linear model** (`glm()` in R). The details of `glm()` are deferred until much later in the quarter. For the present, we use `glm()` only to decide if the effect of each predictor variable on the **categorical** dependent variable is significant. **Because we are using the binomial distribution, we need to specify in the command `glm(…, family = binomial, …)`**. The relevant R script is contained in 'shw2sols.r'.

10. A population is Normal with s.d., $\sigma = 2$, and unknown mean, μ. We wish to test the null hypothesis, H_0: $\mu = 4$, against the alternative hypothesis, H_1: $\mu = 4.4$, using a sample size of 96, and $\alpha = 0.05$. Calculate the power of this test. [Ans. 0.6235.]

Hints: To test H_0, the relevant statistic is the sample mean, $\bar{X}$. The mean of $\bar{X}$ is the same as that of *X*, i.e., μ. The sd of $\bar{X}$, i.e., the standard error of $\bar{X}$, is σ/sqrt(n), where n is the sample size. In this simplified decision problem, we have only 2 alternatives, $\mu = 4$ and $\mu = 4.4$. Therefore, a sensible decision procedure would be to calculate a cutoff, *c*, such that:

(i) when $\bar{X} > c$, we would reject H_0, and when $\bar{X} < c$, we would retain H_0 as being consistent with the data; and
(ii) *c* is chosen such that the Type I error rate is $\alpha = 0.05$, i.e., such that P($\bar{X} > c$ | H_0 is true) = 0.05.

The **power** of the test is defined as 1 minus the Type II error rate, i.e., as the probability of rejecting H_0 given that H_0 is false, which equals the probability of rejecting H_0 given that H_1 is true: P($\bar{X} > c$ | H_1 is true).

To calculate *c*, we use the facts that we are given the **upper-tail percentile** or p-value, 0.05, and we wish to find the **quantile** (or 'q-value'), when the relevant distribution is Normal with **mean** = **4.0** and **sd** = 2/sqrt(96). This can be obtained with:

```
> c1 = qnorm(.05, mean=4.0, sd=2/sqrt(96), lower.tail=F)
```

To calculate the power of the test, we use the facts that we are given the **upper-tail quantile** or q-value, c1, and we wish to find the **upper-tail percentile** or p-value, when the relevant distribution is Normal with **mean** = **4.4** and **sd** = 2/sqrt(96). This can be obtained with:

```
> pwr = pnorm(c1, mean=4.4, sd=2/sqrt(96), lower.tail=F)
```

The results are:

```
> c1
[1] 4.335754
> pwr
[1] 0.6235198
```

11. Nine participants in a clinical study were given a depressive symptoms questionnaire before and after a 6-week intervention. *Before* the intervention the scores were {45,88,65,42,53,11,26,31,59}. *After* the intervention the same participants had these scores (in the same order): {61,87,68,44,52,14,33,44,66}. Test whether the intervention was successful. What would you conclude?

12. A different clinician is comparing two techniques, *cognitive behavioral therapy (CBT)* and *dialectical behavioral therapy (DBT)*. She only reports the Before→After change scores for each group. The changes scores for the CBT group are {13,21,9,12,9,2,3,6,4,7}. The change scores for DBT are {5,1,2,2,6,0,0,4}. Test whether either intervention was more successful. What would you conclude?

13. Calculate $P(1 < \chi_3^2 < 6.25)$. [Ans. 0.7.]

Hints: $P(1 < \chi_3^2 < 6.25) = P(\chi_3^2 < 6.25) - P(\chi_3^2 < 1.00)$.
To get $P(\chi_3^2 < 6.25)$, note that we are given the lower-tail **quantile**, 6.25, and we wish the lower-tail **p-value**. Use, e.g.,

```
> p1 = pchisq(6.25, df = 3, lower.tail=T)
> p1
[1] 0.899939
```

etc.