Statistics Lab #2

PSYCH 710

2 Lab 2

2.1 Initialize R

Create the folder Rlab2 inside the PSY710 folder located in your home directory. Make sure Rlab2 is empty. Then launch R and enter the following commands:

```r
system.file("/PSY710/Rlab2") # set working directory
options(contrasts=c("contr.sum","contr.poly")) # sum-to-zero effects
```

2.2 Graphics

Graphics are an essential component of statistical analysis. Fortunately, R has a strong set of graphics commands. The following sections illustrate several useful and common types of graphs. However, before examining the graphs we will use R’s built-in random number generators to create several synthetic sets of data.

```r
set.seed(7212241) # set random number generator
# create sets of random variables
N <- 100
d0 <- rnorm(n=N) # normal
d1 <- rchisq(n=N,df=1) # chisq
d2 <- rt(n=N,df=2) # t
```

2.2.1 Boxplots

Please watch the Kahn Academy’s nice, brief video that illustrates how to interpret a boxplot.

Boxplots are a very useful graphical summary of major features of a distribution, including central tendency, variability, skewness, and the presence of outliers. The following commands create Figure 1.

```r
dev.new()
par(mfrow=c(1,3),cex=1.15) # divide window into 1 x 3 units; scale fonts
boxplot(d0,ylab='value',xlab='normal')
boxplot(d1,ylab='value',xlab='chisq (df=1)'
boxplot(d2,ylab='value',xlab='t (df=2)'
```

Boxplots show a number of characteristics of the data. The thick dark line represents the median. The lower and upper parts of the box are the so-called hinges, which correspond closely to the first and third quartiles. The vertical height of the box – the difference between the upper and lower hinges – corresponds closely to the Interquartile Range (IQR), which is a measure of the spread of the middle 50% of the data. The whiskers stretch out to the farthest points that are not greater than \( d \) units away from the upper and lower hinges, where \( d = 1.5 \times IQR \). Finally, the points represent extreme scores that are more than 1.5 times IQR away from the
hinges. Compare the box plot for the normally distributed data with the numerical summaries provided by the
following commands:

\begin{verbatim}
fivenum(d0) # Tukey's five-number summary
## [1] -2.34295656 -0.54241462 0.03559709 0.76753927 2.63002787
median(d0)
## [1] 0.03559709
range(d0)
## [1] -2.342957 2.630028
quantile(d0,c(.25,.5,.75)) # 1st, 2nd, & 3rd quartiles
## 25% 50% 75%
## -0.52851811 0.03559709 0.76465582
IQR(d0) # inter-quartile range
## [1] 1.293174
\end{verbatim}

![Boxplots of three sets of data.](image)

Figure 1: Boxplots of three sets of data.
2.2.2 Histograms

Histograms can provide detailed information about the shape of a distribution, although the information value of a histogram depends critically on bin width. R has several ways of calculating bin widths which you can read on the help page for `hist`.

```r
dev.new(height=8)
par(mfrow=c(3,1))  # divide window in 3 x 1 units
hist(d0,breaks=15,ylab='value',xlab='X',main='normal')
hist(d1,breaks=15,ylab='value',xlab='X',main='chisq (df=1)')
hist(d2,breaks=15,ylab='value',xlab='X',main='t (df=2)')
```

2.2.3 Scatterplots

The following commands create two sets of correlated numbers, `x` and `y`, plots them in a scatterplot and then draws the best-fitting line through the points.

```r
set.seed(451)  # set random number generator
x <- rnorm(n=20,0,1)
y <- 0.5*x + 2 + rnorm(n=20,0,.75)
cor(x,y)  # calculate correlation
dev.new()  # new graphics window
plot(x,y,'p',xlab='x',ylab='y')  # create plot [there are lots of options]
reg.line <- lm(y~x)  # estimate and save linear regression parameters
abline(reg.line)  # abline draws best-fitting line in current graphics window
title(line=2,main='A Scatter Plot & Regression Line')  # add title
r.val <- round(cor(x,y),digits=2)  # compute r and round to 2 digits
text(x=1,y=0.5,pos=4,labels=bquote(r==.(r.val)))  # add text to plot
```

2.2.4 Always look at your data

You may be tempted to forgo graphical inspection of your data and instead rely entirely on numerical summaries and analyses. Don’t do it! Good graphs can tell you a lot about your data. To illustrate this point, let’s examine the `anscombe` data set. The following commands load it into R and then rearrange it into a more useful format:

```r
anscombe  # contains 4 pairs of x,y data sets
## contains 4 pairs of x,y data sets
##
x1 x2 x3 x4 y1 y2 y3 y4
## 1  10 10 10  8  8.04  9.14  7.46  6.58
## 2  8  8  8  8  6.95  8.14  6.77  5.76
## 3  13 13 13  8  7.58  8.74 12.74  7.71
## 4  9  9  9  8  8.81  8.77  7.11  8.84
## 5 11 11 11  8  8.33  9.26  7.81  6.47
## 6 14 14 14  8  9.96  8.10  8.84  7.04
## 7  6  6  6  8  7.24  6.13  6.08  5.25
## 8  4  4  4 19  4.26  3.10  5.39 12.50
## 9 12 12 12  8 10.84  9.13  8.15  5.66
## 10 7  7  7  8  4.82  7.26  6.42  7.91
## 11 5  5  5  8  6.58  4.74  5.73  6.89
```
Figure 2: Histograms of three sets of data.
# separate data into 4 distinct x,y pairs:
dat1 <- anscombe[,c('x1','y1')]  # set 1
dat2 <- anscombe[,c('x2','y2')]  # set 2
dat3 <- anscombe[,c('x3','y3')]  # set 3
dat4 <- anscombe[,c('x4','y4')]  # set 4

# rename variables from (x1,y1), (x2,y2), etc to (x,y)
names(dat1) <- c('x','y')
names(dat2) <- c('x','y')
names(dat3) <- c('x','y')
names(dat4) <- c('x','y')

Next, I’m going to compute the mean and standard deviations of the x and y variables in data set 1, and compute the (x,y) correlation:

```r
apply(dat1,MARGIN=2,FUN=mean) # column means
## x y
## 9.000000 7.500909
apply(dat1,2,sd) # column sd
## x y
## 3.316625 2.031568
cor(dat1) # x,y correlation
```

Figure 3: Scatterplot and regression line.
Exercise: Repeat these calculations for `dat2`, `dat3`, and `dat4`. How do the results compare to those obtained with `dat1`? These numerical summaries should suggest that the data sets are similar. To test this idea, plot each data set in four separate scatter plots. Do they look the same?

2.3 Confidence Intervals

A confidence interval is a range of values, calculated from the sample observations, that are believed, with a particular probability, to contain the true population parameter. A 95% confidence interval, for example, implies that were the estimation process repeated again and again, then 95% of the calculated intervals would be expected to contain the true parameter value. Note that the stated probability level refers to the properties of the interval and not to the parameter itself which is not considered a random variable. – B.S. Everitt, Dictionary of Statistics

Watch the Kahn Academy video that illustrates how to use a \( t \) distribution to calculate a 95% confidence interval. The following code calculates the values from the Kahn Academy example:

```r
# kahn academy example data:
the.sample <- c(15.6,16.2,22.5,20.5,16.4,19.4,16.6,17.9,12.7,13.9)
n <- length(the.sample)
y.bar <- mean(the.sample) # sample mean
stdev <- sd(the.sample) # sample sd
alpha <- .05 # want the 100(1-alpha)% confidence interval
t.val <- qt(1-(alpha/2),df=n-1) # note we divide alpha by 2 (because it is 2-sided)
SEM <- stdev/sqrt(n) # standard error of mean

ci.L <- y.bar - t.val*SEM
# CI lower bound

ci.U <- y.bar + t.val*SEM
# CI upper bound

print(sprintf('%i%% Confidence Interval = [%4.2f, %4.2f]',round(100*(1-alpha)),ci.L,ci.U))
```

Note that the bounds of the confidence interval are calculated using a critical \( t \) value that is calculated in R with the `qt` command, which calculates the \( t \) value that corresponds to given values of \( p \) (probability) and \( df \) (degrees-of-freedom). The command `qt()` works in the opposite direction than `pt()`, which returns the \( p \) value that corresponds to a given \( t \) and \( df \).

Exercise: Calculate the 90% and 95% confidence intervals for `my.scores`:

```r
set.seed(710902)

my.scores <- rnorm(n=25,100,sd=5)
```

2.3.1 One-sided Confidence Intervals

Two-sided confidence intervals divide the number line into three ranges (Figure 4). The interval proper (i.e., the range between the upper- and lower-bounds) covers the true mean with a probability of \( 1 - \alpha \). The upper bound may lie below the true mean with \( p = \alpha/2 \), and the lower bound may lie above the true mean with \( p = \alpha/2 \). Now, suppose we ignore the upper bound: in effect, we would combine the green and right-hand
Figure 4: A two-sided confidence interval is an estimate of the range of estimates of the population mean that are consistent with the data. The population mean, $\mu$, is a constant, but the confidence interval varies randomly across samples. If the assumptions underlying the calculation of the confidence interval are valid, then the confidence interval will include, or cover, $\mu$, with a probability $p = 1 - \alpha$. 

Probability of interval containing true population mean = 1-alpha
Probability of lower-bound being greater than true mean = alpha/2
Probability of upper-bound being less than true mean = alpha/2
red regions in Figure 4. The result would be the one-sided confidence interval shown in the top of Figure 5: there still is a $p = \alpha/2$ probability that the lower-bound (now the only bound) will lie above the true mean, and a $p = (1 - \alpha) + \alpha/2$ probability that lower bound lies below the true mean. Similarly, dropping the lower bound of our two-sided confidence yields the one-sided interval shown in the bottom of Figure 5: now there is a $p = (1 - \alpha) + \alpha/2$ probability that the one-sided interval contains the true mean, and a $p = \alpha/2$ probability that the (upper) bound falls below the true mean. Notice that the coverage of 2- and 1-sided intervals change even when $\alpha$ is the same. For example, $\alpha = 0.05$ produces 95% 2-sided confidence intervals, but produces 97.5% 1-sided confidence intervals. Using the equations shown above, setting $\alpha = 0.1$ produces 95% 1-sided confidence intervals.

Figure 5: One-sided confidence intervals which mark the estimated upper (top) and lower (bottom) bounds for the population mean.

I’ve written a short R function, `confint.demo`, to illustrate the random nature of confidence intervals. Load and run the command with the following commands:
The command draws a figure with the mean and 95% confidence intervals for 100 samples of 20 scores drawn randomly from a normal distribution with \( \mu = 0 \) and \( \sigma = 1 \). Red points and intervals do not contain the true mean (i.e., 0). You can explore the effects of sample size (\( n \)) and alpha by changing the values passed to `confint.demo`.

**Exercise:** How should the coverage (width) of the confidence interval change by increasing sample size from 20 to 50? How should coverage change by changing alpha from 0.05 to 0.1 or 0.01? Use `confint.demo` to evaluate your intuition.

### 2.4 \( t \) tests

In class we discussed how \( t \) tests can be used to evaluate hypotheses about group means. In this section you will perform several \( t \) tests with the R command `t.test`. Before beginning, you should review the help page for `t.test`.

The following code illustrates how to conduct a one-sample `t.test` that evaluates the null hypothesis that the scores were drawn from a population with a mean \( \mu = 15 \). Notice that the output of `t.test` includes the sample mean and 95% confidence interval.

```r
the.sample <- c(15.6,16.2,22.5,20.5,16.4,19.4,16.6,17.9,12.7,13.9)
t.test(the.sample,mu=15)
```

#### Output

```
# # One Sample t-test # # data: the.sample # t = 2.3016, df = 9, p-value = 0.04688 # alternative hypothesis: true mean is not equal to 15 # 95 percent confidence interval: # 15.0372 19.3028 # sample estimates: # mean of x # 17.17
```

**Exercise:** Setting \( \alpha = 0.05 \), what is the conclusion regarding the null and alternative hypotheses? Does your conclusion change if \( \alpha = 0.01 \)? Explain. Use `t.test` to calculate the 99% confidence interval of the mean. Finally, use `t.test` to evaluate the null hypothesis that the scores were selected randomly from a population with a mean \( \mu = 17 \).

Next, we will conduct a two-sample \( t \) test:

```r
set.seed(812)
x0 <- rnorm(n=12,100,5)
x1 <- 120 + 4*rnorm(n=12,0,5)
t.test(x0,x1,var.equal=T)
```

#### Output

```
# # Two Sample t-test # # data: x0 and x1
```
Exercise: What is the null hypothesis that is being evaluated by this \( t \) test? What is your conclusion regarding the null hypothesis? Is the assumption of equal group variance reasonable here? (N.B. Looking at boxplots of the data will be helpful.) How can we alter our \( t \) test so that we do not assume that the variances in the two groups are equal? What affect do you think that would have on the \( p \) value? Re-calculate the \( t \) test without assuming equal variances.

2.4.1 one-tailed tests

The logic of one-tailed tests is illustrated in Figure 6. Consider a situation where we want to determine if our scores are drawn from a population with a mean that is less than or equal to 15 (\( \mu \leq 15 \)). Assuming that this hypothesis is correct, it would be unusual to obtain a sample with mean that was much greater than 15. Indeed, if we assume that \( \mu = 15 \), and that the means are distributed normally, we can use the \( t \) distribution to calculate the probability of observing a sample mean (for a our sample size \( n \)) that is at least as large as \( \bar{Y} \). For example, when \( \mu = 15 \) and \( n = 10 \), the probability of obtain a sample mean greater than 16.83 is only \( p = 0.05 \). Of course, \( \mu \) might be less than 15. However, notice that the probability of obtaining a sample mean greater than 16.83 decreases if \( \mu < 15 \). Therefore, setting the critical value of the sample mean to 16.83 ensures that our Type I error rate is no greater than \( \alpha = 0.05 \) (assuming, of course, that all of the assumptions in our analysis are valid).

![Figure 6: Illustration of the logic of a one-tailed t test.](image-url)

Exercise: We believe that that the scores in \( x_0 \) and \( x_1 \) may come from populations with different means: specifically, we think that \( \mu_1 > \mu_0 \). Use a \( t \) test to evaluate that hypothesis. What are the null and alternative hypotheses? What is the confidence interval? What is your conclusion regarding the null hypothesis?
2.4.2 Paired-sample $t$ tests

Finally, we conduct a paired-sample $t$ test. This test is appropriate when the scores in one sample are paired, or correspond, to scores in another sample. The most common case is when the two sets of scores are measured on the same subjects in different conditions (or different times). Paired $t$ tests are performed by setting the `paired` parameter in the `t.test` command to TRUE.

\[ t.test(samp1, samp2, paired=TRUE) \]

**Exercise:** Conduct a paired $t$ to evaluate the null hypothesis that the samples created by the following commands were drawn from populations with different means. Does the $t$ test depend on the order in which the samples appear in `t.test`. Do the samples appear to have different variances? Does is matter if we set `var.equal` to TRUE or FALSE? What are the degrees-of-freedom? Finally, conduct a $t$ test to determine if the mean of `samp2` is significantly greater than the mean of `samp1`.

```
set.seed(41098)
x <- rnorm(20, 0, 10)
samp2 <- 100 + 0.5*x + rnorm(20, 0, 20)
samp1 <- 100 + x
```