Homework Set 1

Katie Daisey

February 18, 2015

1-c,d,e;,3-b,c;4 1-pooled variences t, paired t test 2-b check for normalization by the total, adjust axis to include top point 3-b,c check pdfs - think it might be okay, variance

t.test uses Welch t-test

Question 1

Two analytical methods (XRF and ICP) were applied to random areas on the same semiconductor material with a trace sodium contaminant. The following data was obtained:

```
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
xrf 85.1 81.4 77.1 84.5 87.9 83.2 86.6 83.9 81.1  80.8
icp 87.4 90.1 86.2 89.2 88.4 82.9 81.9 87.4 82.1  80.6
```

a) Treating these methods as separate, but producing replicated results(ie. the variation between samples arising from chance), we calculate the t statistic for pooled variences t-Test.

$$t = \frac{\mu_{xrf} - \mu_{icp}}{\sqrt{MSE}\sqrt{\frac{1}{n} + \frac{1}{n}}} \tag{1}$$

where MSE equals

$$MSE = \frac{(n_{xrf} - 1)\sigma^2_{xrf} + (n_{icp} - 1)\sigma^2_{icp}}{n_{xrf} + n_{icp} - 2} \tag{2}$$

```
[1] -1.666292
```

```
> crit95<-qt(.975,9)
> crit90<-qt(.95,9)
```

The critical statistics,c, are calculated at 2.262 and 1.833 for 95% and 90% confidence respectively.

We cannot reject the null hypothesis (that the means of the two sets are equal) at either confidence level. Thus, we say we cannot distinguish between the two sets with neither 95% nor 90% confidence.

b) If we instead treat the data as pairs of non-replicated samples, treating each spot as separate from the other locations and variation between methods arising from chance) we must use the paired t-Test. The t statistic for paired t-Test is calculated by:

$$t = \frac{\bar{x}_D}{\sigma_D \sqrt{n}} \tag{3}$$

where $\bar{x}_D$ and σ_D is the mean of the differences between the pairs.The value is:

```
[1] -1.838782
```

Since the calculated t statistic is above the c for 90% but below the c for 95%, we would reject the null hypothesis (that the results are not distinguishable) at 90% confidence, but cannot at 95% confidence. Thus, the results are not distinguishable with 95% confidence but distinguishable with 90% confidence.

c) what p values mean

d) 95% vs 90% confidence

e) more statistical power

Question 2

Rutherford and Geiger (Phil. Mag. (1910)20, 698-707) counted alpha particles emitted by polonium using scintillation. With N as the number of particles and f as the frequency N particles were observed during fixed time intervals, the following data was reported:

```
> N<-c(0:14)
> f<-c(57,203,383,525,532,408,273,139,45,27,10,4,0,1,1)
> gold<-cbind(N,f)
> gold2<-rbind(N,f)
> gold2
```

```
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
N    0    1    2    3    4    5    6    7    8     9    10    11    12    13
f   57  203  383  525  532  408  273  139   45    27    10     4     0     1
```

```
  [,15]
N    14
f     1
```

a) The mean number of alpha particles emitted in the fixed time interval can be calculated by finding the weighted mean, ie multiplying each N value by the corresponding f value and dividing by the total number of observations:

```
> particles<-N*f
> totalparticles<-sum(particles)
> totalobs<-sum(f)
> weightedmean<-signif(totalparticles/totalobs,3)
```

giving a mean number of Sexprweightedmean particles per interval.

b)A Poisson distribution is easily calculated in R using the 'dpois' function with a specified lambda of 'weightedmean'. The vertical axis (frequency) is normalized via the first element (number of no clusters emitted).

```
> plotpois<-cbind(N,dpois(N,weightedmean))
> weight<-f[[1]]/plotpois[1,2]
> plotpois[,2]<-weight*plotpois[,2]
> plot(gold,main="Frequency of alpha particles ejected",ylab="frequency",xlab="size of
> points(plotpois,col=3)
```

Question 3

Question 3 Looking a little closer at randomness

a) Several sets of random numbers (mean = 0, standard deviation = 1) were generated in R using the 'rnorm' function (for replicability, the seed was set to 292015).

```
> set.seed(292015)
> r.10<-rnorm(10)
> r.100<-rnorm(100)
> r.1000<-rnorm(1000)
> obs<-c(10,100,1000)
> mean.10<-mean(r.10)
> mean.100<-mean(r.100)
> mean.1000<-mean(r.1000)
> mean<-c(mean.10,mean.100,mean.1000)
> std<-c(sd(r.10),sd(r.100),sd(r.1000))
> table.3<-rbind(mean,std)
> colnames(table.3)<-c(10,100,1000)
> table.3
```

```
             10          100       1000
mean -0.3845336 -0.02510749 0.02636297
std   1.3068197  0.92987961 1.00148238
```

As the data was generated using the normal distribution, we expect the mean to be 0 and the standard deviation to be 1, but they are not. These parameters do however become closer to expected as the number of samples increase. The expected parameters belong to the population. We hope that the sample reflects the population, but because the numbers are generated at random, they only have a probability of exactly mirroring the sample. As the number of samples we generate increase, the probability that the sample parameters equal the population parameters also increases. To put simply, the more observations we make, the more likely the random noise cancels itself out. This is the Law of Large Numbers.

b) The Central Limit Theorem, a related but separate theorem, states that the means of samples generated independently and randomly, *regardless of the probablity distribution used to generate them*, will approximate a normal distribution. For instance, say we have a normally-generated dataset of 1000 integers with a mean of 0 and a variance of 10 (‘rnorm(1000,0,10)‘). We then sample (without replacement) 10 observations from dataset 1000 times. We do similarly for 50, 100, and 200 observations, calculating the mean for each sample.

```
> set.seed(2102015)
> dataset<-rnorm(1000,0,10)
> samplemeans<-data.frame()
> for (i in 1:1000){
+     s.10<-mean(sample(dataset,10))
+     s.50<-mean(sample(dataset,50))
+     s.100<-mean(sample(dataset,100))
+     s.200<-mean(sample(dataset,200))
+     s.all<-c(s.10,s.50,s.100,s.200)
+     samplemeans<-rbind(samplemeans,s.all)
+ }
> colnames(samplemeans)<-c("10","50","100","200")
> varsx<-matrix(c(var(samplemeans[,1]),var(samplemeans[,2]),var(samplemeans[,3]),var(s
> colnames(varsx)<-c("10","50","100","200")
> varsx

          10       50       100      200
[1,] 10.42346 1.850702 0.9207121 0.421384

> #plots
> #10 observations
> hist(samplemeans[,1],freq=F,main="Means of 10 normal observations",xlab="mean")
> lines(density(samplemeans[,1]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
```

```
> #50 observations
> hist(samplemeans[,2],freq=F,main="Means of 50 normal observations",xlab="mean")
> lines(density(samplemeans[,2]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
> #100 observations
> hist(samplemeans[,3],freq=F,main="Means of 100 normal observations",xlab="mean")
> lines(density(samplemeans[,3]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
> #200 observations
> hist(samplemeans[,4],freq=F,main="Means of 200 normal observations",xlab="mean")
> lines(density(samplemeans[,4]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
```

c) Suppose we now compare that with a uniformly-distributed dataset.

```
> set.seed(2102015)
> dataset<-runif(1000,0,10)
> samplemeans<-data.frame()
> for (i in 1:1000){
+     s.10<-mean(sample(dataset,10))
+     s.50<-mean(sample(dataset,50))
+     s.100<-mean(sample(dataset,100))
+     s.200<-mean(sample(dataset,200))
+     s.all<-c(s.10,s.50,s.100,s.200)
+     samplemeans<-rbind(samplemeans,s.all)
+ }
> colnames(samplemeans)<-c("10","50","100","200")
> var

function (x, y = NULL, na.rm = FALSE, use)
{
    if (missing(use))
        use <- if (na.rm)
            "na.or.complete"
        else "everything"
    na.method <- pmatch(use, c("all.obs", "complete.obs", "pairwise.complete.obs",
        "everything", "na.or.complete"))
    if (is.na(na.method))
        stop("invalid 'use' argument")
    if (is.data.frame(x))
        x <- as.matrix(x)
    else stopifnot(is.atomic(x))
    if (is.data.frame(y))
        y <- as.matrix(y)
```

```
    else stopifnot(is.atomic(y))
    .Call(C_cov, x, y, na.method, FALSE)
}
<bytecode: 0x0000000007d9ad90>
<environment: namespace:stats>

> #10 observations
> hist(samplemeans[,1],main="Means of 10 uniform observations",xlab="mean")
> lines(density(samplemeans[,1]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
> #50 observations
> hist(samplemeans[,2],main="Means of 50 uniform observations",xlab="mean")
> lines(density(samplemeans[,2]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
> #100 observations
> hist(samplemeans[,3],main="Means of 100 uniform observations",xlab="mean")
> lines(density(samplemeans[,3]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
> #200 observations
> hist(samplemeans[,4],main="Means of 200 uniform observations",xlab="mean")
> lines(density(samplemeans[,4]),col="navy",lwd=3)
> lines(density(dataset),col="red",lwd=3)
```

Question 4

4.

Question 5

5. A certain chemical analysis is performed with a known probability of error of 0.07. The chemical analysis can be performed qualitatively, producing either a "positive" or a "negative" outcome. The analysis is independently run in triplicate to produce one result, therefore even a single erroneous analysis will produce an erroneous result.

a) Probability density functions can be calculated for errors by calculating independently the probability that 0, 1, 2, and 3 analyses will be in error. As the analyses are independent, the probabilities for each can be multiplied. P(3), the probability that all 3 analyses will be in error is easily calculated as the cube of the error, 0.07.

```
> P.3<-0.07*0.07*0.07
> P.3

[1] 0.000343
```

The probability of none of the analyses being erroneous would be the probability of all the analyses being correct or (1-0.07) cubed.

```
> P.0<-.93*.93*.93
> P.0

[1] 0.804357
```

We then must consider P(1) and P(2), the probability of having only 1 and 2 erroneous analyses respectively. This is a bit more difficult as we must consider the ways in which we can get *exactly* one erroneous test, but it is easily seen that there are only three ways to do so (TTE, TET, ETT). We can then calculated P(1) as the independent probabilities multiplied by the number of ways we can get those analyses.

```
> P.1<-.07*.93*.93*3
> P.1

[1] 0.181629
```

P(2) is calculated similarly.

```
> P.2<-.07*.07*.93*3
> P.2

[1] 0.013671
```

As a check, we can see that the sum of all possible outcomes equals 1.

```
> P.total<-P.0+P.1+P.2+P.3
> P.total

[1] 1
```

b) Now, knowing this, we decide to perform 3 additional analyses on those samples that test "positive" for an analyte (regardless of if it is erroneous or not), generating an additional result. Since the outcome of the analysis is independent from the reliability of the result, we can consider the two tests independent. The probability that a single analysis will be erroneous is 1-P(0), thus the probability that both analyses will be erroneous is simply (1-P(0))*(1-P(0))

```
> P.both<-(1-P.0)*(1-P.0)
> P.both

[1] 0.03827618
```