Homework #1

Katie Daisey

Tuesday, March 10, 2015

library(knitr)  
library(lattice)

## Question 1 - T-tests

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:

xrf<-c(85.1, 81.4, 77.1, 84.5, 87.9, 83.2, 86.6, 83.9, 81.1, 80.8)  
icp<-c(87.4, 90.1, 86.2, 89.2, 88.4, 82.9, 81.9, 87.4, 82.1, 80.6)  
data.1<-cbind(xrf,icp)  
data.2<-rbind(xrf,icp)  
rownames(data.2)<-c("XRF","ICP")  
colnames(data.2)<-c(1:10)  
kable(data.2,row.names=T, caption="Na concentrations (\*$\\mu$\*g/g)")

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

Na concentrations (g/g)

#### a) Spots as Replicates

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

where MSE equals

#pooled variance  
tstat1<-t.test(xrf,icp,var.equal=TRUE)  
tstat1p<-tstat1[[3]]

giving a t statistic of -1.666 and a p-value of 0.113.

The critical statistics, c, are calculated at 2.262 and 1.833 for 95% and 90% confidence respectively or alternatively have an of 0.05 and 0.10 .

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) Different Spots Tested by 2 Methods

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 is calculated as:

where and is the mean of the differences between the pairs.

#paired  
tstat2<-t.test(xrf,icp,paired=TRUE)  
tstat2p<-tstat2[[3]]

Since the calculated t statistic, -1.839 and the p-value is 0.099, is above the c for 90% but below the c for 95%, we would reject the null hypothesis (that the results are distinguishable) at 90% confidence, but cannot at 95% confidence. Alternatively, the p-value, 0.099 is above for 95% confidence, but below the for 90% confidence. Thus, the results are not distinguishable with 95% confidence but distinguishable with 90% confidence.

#### c) P-values

The p value calculated by a t-test is the probability that the null hypothesis has been falsely rejected. So, for instance, a p-value of 0.09 would indicate there is a 0.09 probability that the null hypothesis is true, but we have mistakenly rejected it.

#### d) Confidence Intervals

95% confidence states that 95% of the ranges we calculate will contain the true statistic. We must then accept a broader range than we would need to contain the true statistic 90% of the time. A higher confidence necessarily requires a smaller probability of falsely rejecting the null hypothesis ie. a smaller p-value. At times, a test result will fall into this zone where it would be accepted in the larger range (95% confidence) but not in the smaller range (90% confidence).

#### e) Statistical Power

Statistical power, the probability of not falsely accepting the null hypothesis (), depends on the the variance of the sample; a higher variance producing a less powerful test.

(var.pooled<-var(xrf)/length(xrf) + var(icp)/length(icp))

## [1] 2.179556

n.pooled<-10 #n in each group  
(var.paired<-var(xrf)+var(icp)-2\*cov(xrf,icp)) #var of differences between pairs

## [1] 17.89822

n.paired<-10 #number of pairs

Checking the variance of the two samples(pooled vs paired), originally you might suspect that the paired t-test has a much smaller power, but actually computing the power shows that both tests have the same power.

(power.pooled<-power.t.test(n=n.pooled, delta=0, sd=sqrt(var.pooled), sig.level=0.05, power=NULL, type="two.sample", alternative="two.sided"))

##   
## Two-sample t test power calculation   
##   
## n = 10  
## delta = 0  
## sd = 1.476332  
## sig.level = 0.05  
## power = 0.025  
## alternative = two.sided  
##   
## NOTE: n is number in \*each\* group

(power.paired<-power.t.test(n=n.paired, delta=0, sd=sqrt(var.paired), sig.level=0.05, power=NULL, type="paired", alternative="two.sided"))

##   
## Paired t test power calculation   
##   
## n = 10  
## delta = 0  
## sd = 4.230629  
## sig.level = 0.05  
## power = 0.025  
## alternative = two.sided  
##   
## NOTE: n is number of \*pairs\*, sd is std.dev. of \*differences\* within pairs

Remember, the t-tests are testing different situations, but with the same dataset.

## Question 2 - Poisson Distributions

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:

#poisson counting  
num<-c(0:14)  
freq<-c(57,203,383,525,532,408,273,139,45,27,10,4,0,1,1)  
Freq<-matrix(c("Freq",freq),nrow=1)  
gold<-cbind(num,freq)  
colnames(Freq)<-c("Num",0:14)  
kable(Freq,row.names=F,caption="Frequency of alpha particles")

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Num | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Freq | 57 | 203 | 383 | 525 | 532 | 408 | 273 | 139 | 45 | 27 | 10 | 4 | 0 | 1 | 1 |

Frequency of alpha particles

#### a) Weighted Mean

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:

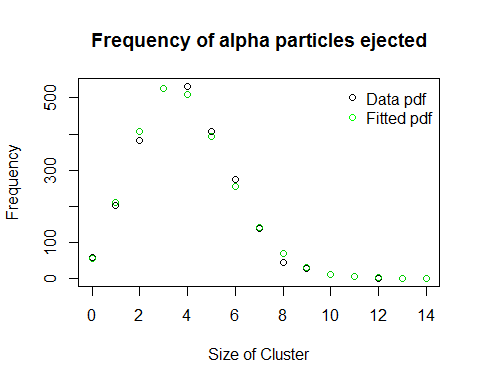
particles<-num\*freq  
total\_particles<-sum(particles)  
total\_obs<-sum(freq)  
weighted\_mean<-signif(total\_particles/total\_obs,3)

giving a mean number of 3.87 particles per interval.

#### b) Calculated vs Actual Distribution

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

plotpois<-cbind(num,dpois(num,weighted\_mean))  
weight<-freq[[1]]/plotpois[1,2]  
plotpois[,2]<-total\_obs\*plotpois[,2]  
  
plot(gold,main="Frequency of alpha particles ejected",ylab="Frequency",  
 xlab="Size of Cluster")  
 points(plotpois,col=3)  
 legend("topright",pch=1,col=c("black","green"),  
 legend=c("Data pdf" ,"Fitted pdf"),bty="n")



## Question 3 - Randomness

#### a) Law of Large Numbers

Several sets of random numbers (mean = 0, standard deviation = 1) were generated in R using the rnorm function.

set.seed(292015)  
  
#generate random numbers  
  
 r.10<-rnorm(10)  
 r.100<-rnorm(100)  
 r.1000<-rnorm(1000)  
 mean.10<-mean(r.10)  
 mean.100<-mean(r.100)  
 mean.1000<-mean(r.1000)  
#join in matrix  
 all\_means<-c(mean.10,mean.100,mean.1000)  
 all\_var<-c(var(r.10),var(r.100),var(r.1000))  
 table.3<-rbind(all\_means,all\_var)  
 colnames(table.3)<-c(10,100,1000)  
  
kable(table.3)

|  |  |  |  |
| --- | --- | --- | --- |
|  | 10 | 100 | 1000 |
| all\_means | -0.3845336 | -0.0251075 | 0.026363 |
| all\_var | 1.7077779 | 0.8646761 | 1.002967 |

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 size of the samples we generate increases, 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) Central Limit Theorem

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,sqrt(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)  
  
#create population  
dataset<-rnorm(1000,mean=0,sd=sqrt(10))  
sample\_means<-data.frame()  
  
#sample population  
 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)  
 sample\_means<-rbind(sample\_means,s.all)  
 }

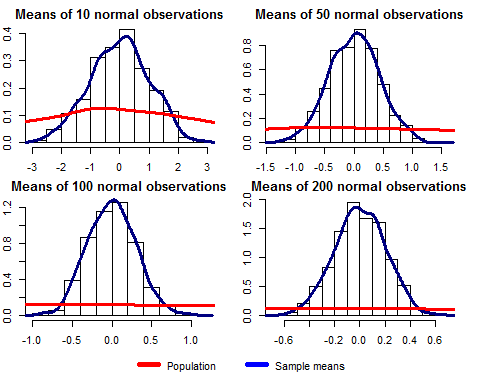
colnames(sample\_means)<-c("10","50","100","200")  
a<-apply(sample\_means,2,FUN=mean)  
b<-apply(sample\_means,2,FUN=var)  
var\_sx<-rbind(a,b)  
colnames(var\_sx)<-c("10","50","100","200")  
rownames(var\_sx)<-c("Mean","Var")  
  
kable(var\_sx,row.names=T,caption="Various-sized Samples of 1000 random Numbers")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10 | 50 | 100 | 200 |
| Mean | -0.0051807 | 0.0065885 | -0.0002508 | 0.0019662 |
| Var | 1.0423460 | 0.1850702 | 0.0920712 | 0.0421384 |

Various-sized Samples of 1000 random Numbers

Comparing the histogram and pdf for each set of sample means to the pdf for the original dataset shows interesting differences.

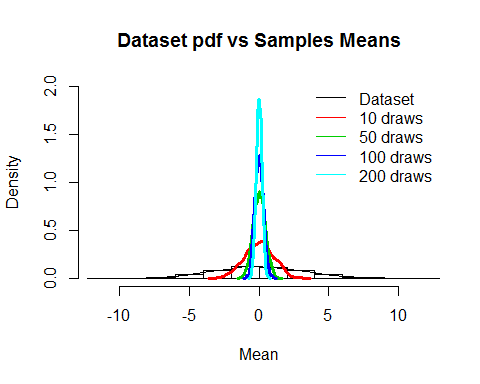
#plots  
par(mfrow=c(1,1),mar=c(2,2,2,2))  
  
m<-matrix(c(1,2,3,4,5,5),3,2,byrow=TRUE)  
layout(m,heights=c(.45,.45,.1))  
  
#10 observations  
 hist(sample\_means[,1],freq=F,main="Means of 10 normal observations",xlab="mean")  
 lines(density(sample\_means[,1]),col="navy",lwd=3)  
 lines(density(dataset),col="red",lwd=3)  
  
#50 observations  
 hist(sample\_means[,2],freq=F,main="Means of 50 normal observations",xlab="mean")  
 lines(density(sample\_means[,2]),col="navy",lwd=3)  
 lines(density(dataset),col="red",lwd=3)  
  
#100 observations  
 hist(sample\_means[,3],freq=F,main="Means of 100 normal observations",xlab="mean")  
 lines(density(sample\_means[,3]),col="navy",lwd=3)  
 lines(density(dataset),col="red",lwd=3)  
  
#200 observations  
 hist(sample\_means[,4],freq=F,main="Means of 200 normal observations",xlab="mean")  
 lines(density(sample\_means[,4]),col="navy",lwd=3)  
 lines(density(dataset),col="red",lwd=3)  
  
par(mar=c(0,0,0,0))  
plot(5, type = "n", axes=FALSE, xlab="", ylab="")  
legend("center",inset=0,legend=c("Population","Sample means"),col=c("red","blue"), lty=1, lwd=5,cex=1,horiz=T,bty="n")



In each of these plots, the red line, the pdf for the dataset itself, is constant. As the number of observations made increases, ie the more numbers we sample to calculate the mean increases, the curve not only becomes more normal-like, but the probability of the mean being around 0 increases. In fact, since we are averaging the numbers (and therefore selecting more than one), the probability of finding a mean equal to 0 has surpassed the probability of finding a number equal to 0.

It becomes much clear by plotting the various mean pdfs with the original dataset.

par(mfrow=c(1,1))  
hist(dataset,freq=F,ylim=(c(0,2)),main="Dataset pdf vs Samples Means", xlab="Mean")  
lines(density(dataset),col=1,type="l")  
lines(density(sample\_means[,1]),col=2,lwd=3)  
lines(density(sample\_means[,2]),col=3,lwd=3)  
lines(density(sample\_means[,3]),col=4,lwd=3)  
lines(density(sample\_means[,4]),col=5,lwd=3)  
legend("topright",bty="n",lty=1, col=c(1:5),   
 legend = c("Dataset","10 draws","50 draws","100 draws","200 draws"),cex=1)



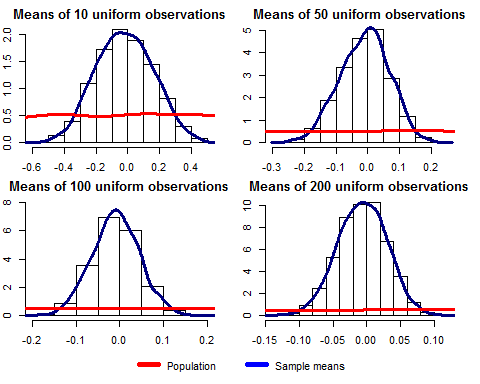
With more draws being averaged (ie 200 vs 10), the pdf is more symmetric, more "Gaussian-like", closer to being centered at the mean, and has a smaller variance, demonstrating the Central Limit Theorem.

#### c) CLT with non-Gaussian data

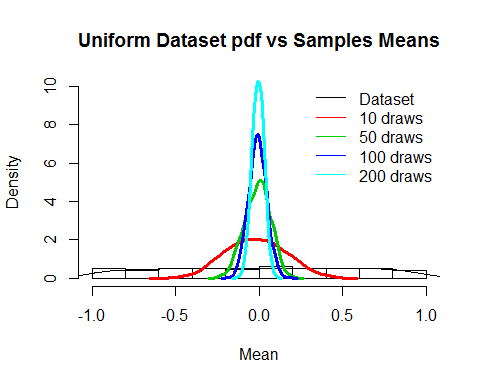
Suppose we now compare that with a uniformly-distributed dataset of 1000 numbers from -1 to 1.

set.seed(2802015)  
dataset2<-runif(1000,-1,1)  
sample\_meansu<-data.frame()  
for (i in 1:1000){  
 s.10<-mean(sample(dataset2,10))   
 s.50<-mean(sample(dataset2,50))  
 s.100<-mean(sample(dataset2,100))  
 s.200<-mean(sample(dataset2,200))  
 s.all<-c(s.10,s.50,s.100,s.200)  
 sample\_meansu<-rbind(sample\_meansu,s.all)  
}  
colnames(sample\_meansu)<-c("10","50","100","200")

#plots  
par(mfrow=c(1,1),mar=c(2,2,2,2))  
m<-matrix(c(1,2,3,4,5,5),3,2,byrow=TRUE)  
layout(m,heights=c(.45,.45,.1))  
  
#10 observations  
 hist(sample\_meansu[,1],freq=F,main="Means of 10 uniform observations",xlab="mean")  
 lines(density(sample\_meansu[,1]),col="navy",lwd=3)  
 lines(density(dataset2),col="red",lwd=3)  
  
#50 observations  
 hist(sample\_meansu[,2],freq=F,main="Means of 50 uniform observations",xlab="mean")  
 lines(density(sample\_meansu[,2]),col="navy",lwd=3)  
 lines(density(dataset2),col="red",lwd=3)  
  
#100 observations  
 hist(sample\_meansu[,3],freq=F,main="Means of 100 uniform observations",xlab="mean",ylim=(c(0,8)))  
 lines(density(sample\_meansu[,3]),col="navy",lwd=3)  
 lines(density(dataset2),col="red",lwd=3)  
  
#200 observations  
 hist(sample\_meansu[,4],freq=F,main="Means of 200 uniform observations",xlab="mean")  
 lines(density(sample\_meansu[,4]),col="navy",lwd=3)  
 lines(density(dataset2),col="red",lwd=3)  
  
par(mar=c(0,0,0,0))  
plot(5, type = "n", axes=FALSE, xlab="", ylab="")  
legend("center",inset=0,legend=c("Population","Sample means"),col=c("red","blue"), lty=1, lwd=5,cex=1,horiz=T,bty="n")



par(mfrow=c(1,1))  
hist(dataset2,freq=F,ylim=(c(0,10)),main="Uniform Dataset pdf vs Samples Means", xlab="Mean")  
lines(density(dataset2),col=1,type="l")  
lines(density(sample\_meansu[,1]),col=2,lwd=3)  
lines(density(sample\_meansu[,2]),col=3,lwd=3)  
lines(density(sample\_meansu[,3]),col=4,lwd=3)  
lines(density(sample\_meansu[,4]),col=5,lwd=3)  
legend("topright",bty="n",lty=1, col=c(1:5),   
 legend = c("Dataset","10 draws","50 draws","100 draws","200 draws"),cex=1)



Here, even though the underlying dataset is clearly not normal, the distribution of the means of replicate samples is indeed normal, an important distinction of the Central Limit Theorem.

## Question 4 - Extra Credit

## Question 5 - Probability

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) PDF for errors

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)

## [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)

## [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)

## [1] 0.181629

P(2) is calculated similarly.

(P.2<-.07\*.07\*.93\*3)

## [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)

## [1] 1

#### b) Additional Test Probability

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

(P.both<-(1-P.0)\*(1-P.0))

## [1] 0.03827618

Whether 0.038 is a small enough probability is not a statistical question.