Stat 107: Introduction to Business and Financial Statistics

Homework 3 Sketch Solutions

1) Develop a simulation for the following problem. The management of Madeira Manufacturing Company is considering the introduction of a new product. The fixed cost to begin the production of the product is $30,000. The variable cost for the product is uniformly distributed between $16 and $24 per unit. The product will sell for $50 per unit. Demand for the product is best described by a normal probability distribution with a mean of 1200 units and a standard deviation of 300 units. Use 500 simulation trials to answer the following questions:

   a) What is the mean profit for the simulation?
   b) What is the probability that the project will result in a loss?
   c) What is your recommendation concerning the introduction of the product?

```
n.sims=500
profits=rep(NA,n.sims)
fixed.cost=30000
price=50
for (i in 1:n.sims)
{
variable.cost=runif(1,16,24)
demand=rnorm(1,1200,300)
profits[i]=(price-variable.cost)*demand-fixed.cost
}
```

```
> mean(profits)
[1] 6272.024
```

Answers will vary. Should be roughly between 5,000 and 7,000

```
> n.loss=length(profits[profits<0])
> p.loss=n.loss/n.sims
> p.loss
[1] 0.238
```

Answers will vary. Should be roughly between 20% and 30%

Recommendation: If the management team is risk-averse, then I would NOT recommend going ahead with the production due to the ~25% chance of negative profit. However, if the management team is comprised of risk seekers, then I would suggest they go for it.

2) Read about the Volume Weighted Moving Average here (http://www.financialwisdomforum.org/gummy-stuff/VMA.htm). Modify the 200day moving average code from class to instead use VMA [the R code is for example `VWMA(n=200,Ad(AAPL),Vo(AAPL))`] For AAPL, ANF, GE, C and LCI compare the returns from a 200 day moving average to a 200 day volume weighted moving average. Use data staring from 2010-01-01.

Really awful looking code but it gets the job done. Uses 10 months (equivalent to 200 days) for the moving averages [working with monthly data]

Note that sometimes the weighted routine is better, and other times its not.

```
> my200("AAPL",weighted=FALSE)
[1] "MA CAGR = 0.2664"
[1] "BH CAGR = 0.2239"
> my200("AAPL",weighted=TRUE)
[1] "MA CAGR = 0.2664"
[1] "BH CAGR = 0.2239"
> my200("C",weighted=FALSE)
[1] "MA CAGR = -0.0067"
[1] "BH CAGR = 0.059"
> my200("C",weighted=TRUE)
[1] "MA CAGR = 0.0078"
[1] "BH CAGR = 0.059"
```

```
my200=function(ticker,weighted=FALSE) {
startdate="2009-12-01"
enddate = "2016-08-30"
malength=10
##ticker="AAPL"

stockdata=getSymbols(ticker,from=startdate,to=enddate,auto.assign=FALSE)
which=endpoints(stockdata,"months")
stockdata=stockdata[which,]
ostockdata=stockdata

##silliness
thetimes=time(stockdata)
goback=thetimes[1]-malength*30
morestockdata=getSymbols(ticker,from=goback,to=enddate,auto.assign=FALSE)
which=endpoints(morestockdata,"months")
morestockdata=morestockdata[which,]

stockdata=morestockdata
sp=as.numeric(Ad(stockdata))
ndays=length(sp)
if (weighted) ma=VWMA(n=malength, Ad(stockdata), Vo(stockdata))
if (!weighted) ma=SMA(sp,malength)

signal="inStock"
buyprice=sp[malength]
sellprice=0
mawealth=1

for(d in (malength+1):ndays) {
if((sp[d]>ma[d]) && (signal=="inCash")) {
   buyprice=sp[d]
   signal = "inStock"
  ## print(paste("Buy Price = ",buyprice))
}

if(((sp[d]<ma[d]) || (d==ndays)) && (signal=="inStock")) {
   sellprice=sp[d]
   signal = "inCash"
   mawealth=mawealth*(sellprice/buyprice)
##print(paste("Sell Price = ",sellprice))
}
}


bhwealth=sp[ndays]/sp[malength]
ndays=nrow(ostockdata)
print(paste("MA CAGR = ",mawealth^(1/((ndays-1)/12))-1))
print(paste("BH CAGR = ",bhwealth^(1/((ndays-1)/12))-1))
##print(paste(mawealth,bhwealth))
}
```

3) A student of mine was asked this question in an interview. “Consider the following game: A cup is filled with 100 pennies. The cup is shaken, and the pennies are poured onto a table. If at least 60 of the pennies are Heads, you win $20. Otherwise, you lose $1. Is this a good game to play?” (that is does it have a positive expected value?). Answer this using simulation in R. That is, simulate playing this game many times and determine how much you would win or lose. Does it appear to be a fair game?

   P(X>=60) can be found from the fact that X is a binomial random variable with n=100 and p=.5.

```
> 1-pbinom(59,100,.5)
[1] 0.02844397
```

   So P(X>=60)=0.02844
   E(Winnings) = 20*P(X>=60)-1P(X<60)=-0.403

   Let’s now simulate 1000 plays of this game:

```
> nsims=1000
> coinmatrix=matrix(nrow=nsims,ncol=100,sample(c(0,1),100*nsims,replace=TRUE))
> indgames=apply(coinmatrix,1,sum)
> over60=sum(indgames>=60)/nsims
> winnings=20*over60-1*(1-over60)
> print(paste("Average winnings over many simulations is ",winnings))
[1] "Average winnings over many simulations is  -0.454"
>
```

4) For this problem we are going to build our own normality test using the bootstrap. To do so, we first need to know how to extract the bootstrap confidence interval from the `boot` function.

   The following *example* code extracts the bootstrap confidence interval:

```
mymean = function(x,i) return(mean(x[i])
myboots = boot(mydata,mymean,R=1000)
mybootsci=boot.ci(myboots)$normal
lowerci=mybootsci[2]
upperci=mybootsci[3]
```

   If data is truly normal, it should have a skewness value of 0 and a kurtosis value of 3. Write an R function that conducts a normality test as follows: it takes as input a data set, calculates a bootstrap confidence interval for the skewness, calculates a bootstrap confidence interval for the kurtosis, then sees if 0 is in the skewness interval and 3 is in the kurtosis interval. If so, your routine prints that the data is normally distributed, otherwise your routine should print that the data is not normally distributed. Test your routine on normal (rnorm), uniform (runif), exponential (rexp) , AAPL daily and AAPL monthly returns.

```
install.packages("moments")
library(moments)
normcheck = function(my.data)
{
my.kurtosis = function(x,i) return(kurtosis(x[i]))
my.skewness = function(x,i) return(skewness(x[i]))
my.kurt.boot = boot(my.data,my.kurtosis,R=1000)
my.skew.boot = boot(my.data,my.skewness,R=1000)
my.kurt.ci = boot.ci(my.kurt.boot)$normal
my.skew.ci = boot.ci(my.skew.boot)$normal
if (my.kurt.ci[3] > 3 && my.kurt.ci[2] < 3 && my.skew.ci[3] > 0 &&
my.skew.ci[2] < 0){
return("Data is normally distributed!")
} else{
return("Data is not normal.")
}
}

library(quantmod)
getSymbols("AAPL", from="2013-01-01")
dr.AAPL=as.numeric(dailyReturn(Ad(AAPL)))
mr.AAPL=as.numeric(monthlyReturn(Ad(AAPL)))
```

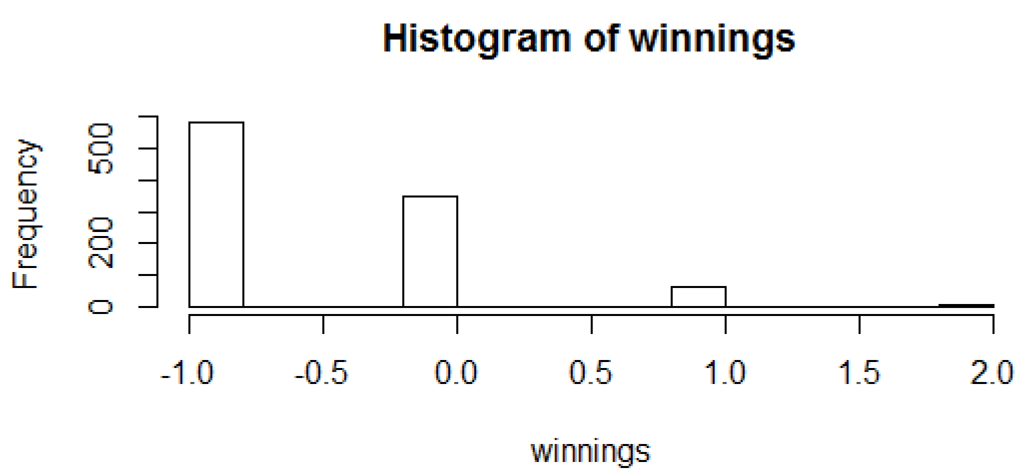
```
> normcheck(rnorm(100))
[1] "Data is normally distributed!"

> normcheck(runif(100))
[1] "Data is not normal."

> normcheck(rexp(100))
[1] "Data is not normal."

> normcheck(dr.AAPL)
[1] "Data is not normal."

> normcheck(mr.AAPL)
[1] "Data is not normal."  OR “Data is normal” ** Depending on date range of getSymbols
```

5) In the game of Chuck-A-Luck a player tosses three dice and earns a dollar for each 6 that appears. It costs one dollar for each roll of the three dice. Let X be the random variable representing the dollar amount won in a single roll of the three dice minus the dollar spent to play. Example: if the player rolls (6, 2, 6), then X = 1.

Write R code to simulate the game of Chuck-A-Luck in the following steps.

a) Write a function to simulate a roll of Chuck-A-Luck. The function should return the amount won minus the dollar spent (i.e., the random variable X from above.)
b) Run your function 1000 times, storing the values in a vector. Compute the sample mean and variance for this simulation. Plot a histogram of the X variable. Is the histogram symmetric about the mean?

```
onechuck=function(numdice) {
rolls=sample(1:6,numdice,replace=TRUE)
return(sum(rolls==6)-1)
}

winnings=1:1000
for(i in 1:1000) winnings[i]=onechuck(3)

> mean(winnings)
[1] -0.51
> sd(winnings)
[1] 0.6374216
```



Histogram is not symmetric around the mean.

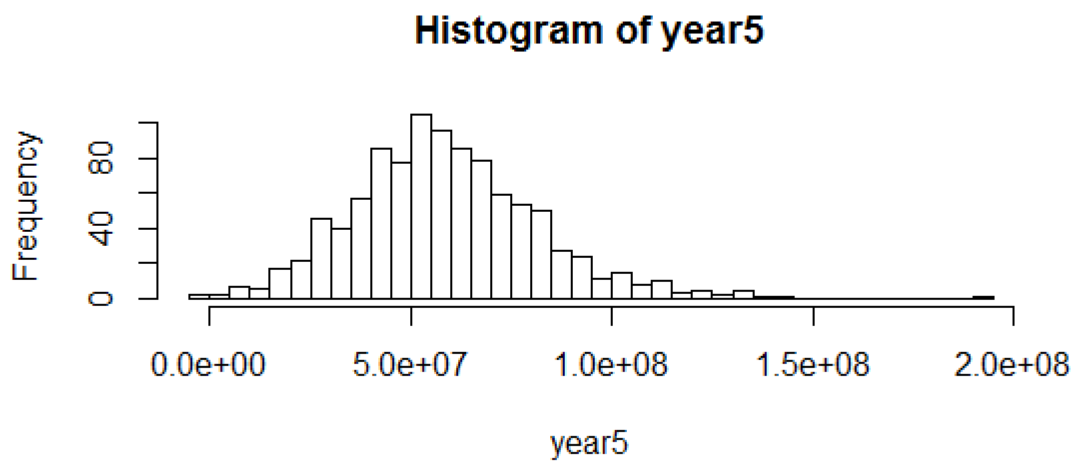
6) Introducing the Leverage Brothers (this problem is shorter than it looks).

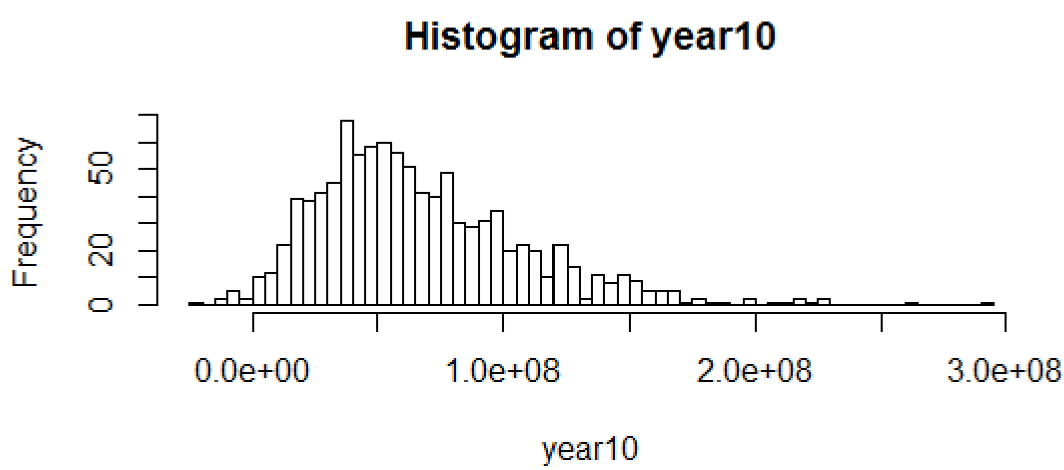
Everyone will have slightly different output based on their assumptions. The code below assumes the following:

1) The management fee is paid at the end of the year based on total assets under management.
2) No Taxes are paid if the returns are negative for the year
3) We made the code into a function-no reason to do so but makes it slicker.

```
fname="http://people.fas.harvard.edu/~mparzen/stat107/histrets_sb.csv"
mydata = read.csv(fname)
returns = mydata$SP500
fit = logspline(returns)

LBros = function(num.years,lawsuit) {
values = 1:1000
for (j in 1:1000) {
money = 50000000
expenses = 1500000
for (i in 1:num.years) {
ret = rlogspline(1,fit)
taxes = 0
lawsuit = 0
if (ret > 0)
taxes = .3*money*ret
if (lawsuit == "yes" && i == 5)
lawsuit = 10000000
fee = .005*money*(1+ret)
money = money*(1+ret) - fee - expenses - taxes - lawsuit
expenses = expenses * 1.05
}
values[j] = money
}
return(values)
}
year5 = LBros(5,"no")
year10 = LBros(10,"yes")

summary(year5)
summary(year10)
sd(year5)
sd(year10)
hist(year5,breaks=50)
hist(year10,breaks=50)
```

```
> summary(year5)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
 -3532000  43220000  56940000  58510000  72100000 193000000

> summary(year10)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-22920000  38020000  58420000  66510000  89670000 293900000

> sd(year5)
[1] 23107353

> sd(year10)
[1] 41099416
```





```
> sum(year5 < 50000000)/1000
[1] 0.37
> sum(year10 < 50000000)/1000
[1] 0.364
```

Is there a chance that they lose the fortune? Well, it depends on what you define as "lose the fortune." Above, I defined losing the fortune just as losing money over the course of the years. (You can define it however you'd like – it's arbitrary and you will get credit if the logic is there.) Basically, we see a 37% chance of making no money/losing money over the course of both simulations. And for standard deviation? It isn't the best measure of risk here. Our distribution is skewed (meaning our large, large values have a lot of impact on the value), and, realistically, we care about downward variability more than upward. It's a fair assumption that the brothers would care much more about losing all of their money than the slim chance of their money growing to over 100M.