Part I - Using R To Analyze Data

Download the following file (https://www.dropbox.com/s/zjf0lm830wzmkwz/snmp-delta.csv?dl=0 (https://www.dropbox.com/s/zjf0lm830wzmkwz/snmp-delta.csv?dl=0)), which contains the following as table as comma-separated values.

here are a total of 8 columns of data. You should be able to read the data into R using the provided commands below.

These are measurements in packets per interval and bytes per interval for specific interfaces on the router from the prior assignment, ideally structured as a notebook. For this assignment, you're going to hand in an "R Program" that shows how you analyzed the data.

You should hand in a single PDF document with your answers to the questions, including any graphics needed. R can produce plots of graphs (described below) -- if you don't or can't use the notebook form (which will include the plots directly), you should be able to suck the plots into OpenOffice or other word processing tools.

In [664]:
```
temporaryFile <- tempfile()
snmpData = read.csv("snmp-delta.csv")
```

Problem #1

Data Manipulation: Show how to compute a table that contain the sum of all packets (input and outputt, unicast and non-unicast) for each interface and the sum of bytes (input and outputt, unicast and non-unicast) for each interface for each output link. The table should have a total of 3 columns (interface, packets, bytes) and a row for each interface.

Refer to "slices" on page 5 and 6 of the SimpleR manual and the **data.frame** function to construct a data frame from vectors. In my solution, I defined functions that sliced the data for a specific interface and then used **sapply** to compute the data for all interfaces.

In [665]:
```
library(plyr)
df=data.frame(snmpData)
out=ddply(df, .(if.), numcolwise(sum) )
d1=data.frame(interface=out$if. , TotalOctets=(out$inOctets+out$outOctets), TotalPackets=(out$outUCastPkts+ out$inUcastPkts+ out
d1
```

```
Error in eval(expr, envir, enclos): '2776493L' is not a function, character or symbol
```

Out[665]:

	interface	TotalOctets	TotalPackets
1	1	0	0
2	2	3946320	39054
3	3	2493713	17027
4	6	2186780	15260
5	7	3340889	37306
6	9	0	0
7	10	1833765	19793
8	11	1833765	19793

Problem #2

Summary Statistics: Using the data for the inUcastPkts packets on interface #2, show how you would use R to compute the mean, variance, standard deviation and COV "by hand". For example, if you had read the data into data frame x, you could compute the sum of the first column using a **sum** over the slice of the inUcastPkts for interface 2, and then use this to compute the mean. You should compute these terms using e.g. sum(), length() and other functions following the computations in the textbook. Your solution should show both the formulas and their output.

It will be easiest to first extract the data you are working on to a vector that you use in the remainder of your calculations.

In [666]:
```
sum=sum(df$inOctets[df$if.==2])
length=length(df$inOctets[df$if.==2])

d2=data.frame(Interface="2",Parameter="inUcastPkts",
sum=sum(df$inUcastPkts[df$if.==2]),
mean=mean(df$inUcastPkts[df$if.==2]),
variance=var(df$inUcastPkts[df$if.==2]),
std_dev=sd(df$inUcastPkts[df$if.==2]),
COV=sd(df$inUcastPkts[df$if.==2])/mean(df$inUcastPkts[df$if.==2]))

d2
```

Out[666]:

	Interface	Parameter	sum	mean	variance	std_dev	COV
1	2	inUcastPkts	29480	1551.579	151828.9	389.6523	0.2511328

Problem #3

R Bulit-in: Use the built-in R commands (see the SimpleR manual) to compute the mean, variance, sd & COV for the other columns (I don't think there is a function for COV in R, so you will need to compute that by hand, or better yet, define an R function).

In [667]:
```
d3=data.frame("Interface"= character(), "parameter"= character(), "Mean"= character(), "Variance"= character(), "Std-dev"= chara
for (i in c("outUCastPkts","inOctets", "outOctets", "inUcastPkts", "outNUCastPkts", "inNUcastPkts")) {
    d3[nrow(d3)+1,]=c("2",i,mean(df[if.==2,i]),
    var(df[if.==2,i]),
    sd(df[if.==2,i]),
    sd(df[if.==2,i])/mean(df[if.==2,i]))
}

d3
```

Out[667]:

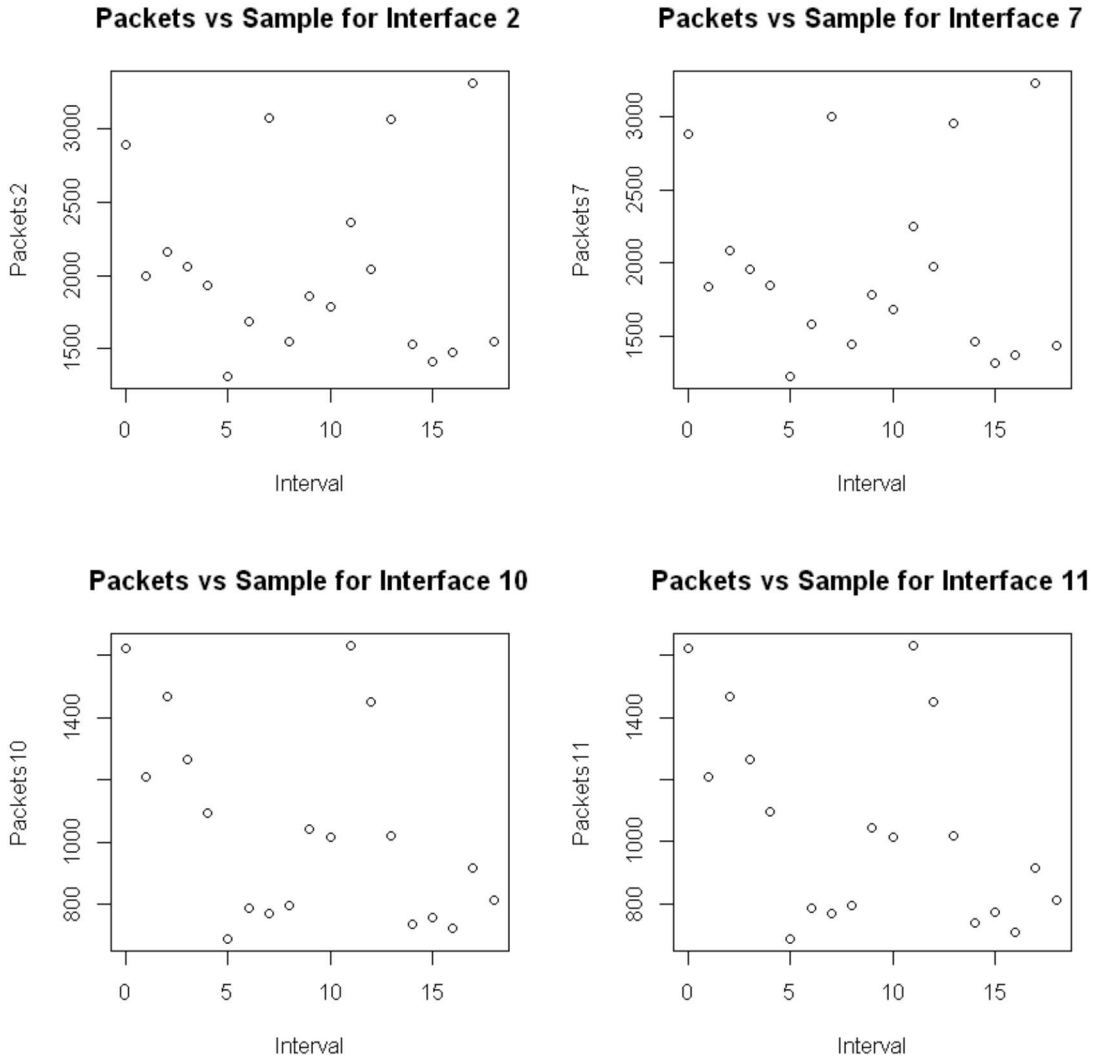
	Interface	parameter	Mean	Variance	Std.dev	COV
1	2	outUCastPkts	503.894736842105	84325.0994152047	290.387843091278	0.576286715973917
2	2	inOctets	146131.210526316	1011585204.06433	31805.4272737268	0.217649789933131
3	2	outOctets	61569.8421052632	782747504.473684	27977.6250685022	0.454404690865865
4	2	inUcastPkts	1551.57894736842	151828.923976608	389.652311653105	0.251132765312381
5	2	outNUCastPkts	0	0	0	NaN
6	2	inNUcastPkts	0	0	0	NaN

Problem #4

Plot histograms for the number of packets (input and output of all types) per interval on interfaces 2, 7, 10 and 11. You can arrange plots into a grid using the following commands:

```
par(mfrow=c(2,2))
plot(....)
etc
plot(...)
```

In [668]:
```
Interval=df[if.==2,"sample"]
Packets2=df[if.==2,"outUCastPkts"]+df[if.==2,"inUcastPkts"]
Packets7=df[if.==7,"outUCastPkts"]+df[if.==7,"inUcastPkts"]
Packets10=df[if.==10,"outUCastPkts"]+df[if.==10,"inUcastPkts"]
Packets11=df[if.==11,"outUCastPkts"]+df[if.==11,"inUcastPkts"]
par(mfrow=c(2,2))
plot(Interval,Packets2,main="Packets vs Sample for Interface 2")
plot(Interval,Packets7,main="Packets vs Sample for Interface 7")
plot(Interval,Packets10,main="Packets vs Sample for Interface 10")
plot(Interval,Packets11,main="Packets vs Sample for Interface 11")
```



Problem #5

Is the behavior across any of interfaces 2, 7, 10 and 11 correlated? In other words, are the number of packets sent on one link similar to the number of packets sent on another link? A similar property might be seen for the number of bytes. Use the "correlation" function (corr) across interfaces 2, 7, 10 and 11 for packets (as above) to argue what links have the most correlated behavior.

In [669]:
```
cor(Packets2,Packets7)
cor(Packets2,Packets10)
cor(Packets2,Packets11)
cor(Packets7,Packets10)
cor(Packets7,Packets11)
cor(Packets10,Packets11)
print ("correlation is high between interfaces 10 & 11")
```

Out[669]: 0.99910454763274

Out[669]: 0.390749159036285

Out[669]: 0.390500964864765

Out[669]: 0.396481388012108

Out[669]: 0.396253619377111

Out[669]: 0.999888610646897

```
[1] "correlation is high between interfaces 10 & 11"
```

Problem #6

Using the equations from Baudec (section 2.2.3), compute the confidence interval for the mean for the number of inUcastPkts packets per interval on interface #2. You can use the built in commands in R to compute the mean and variance (or sd). Compute the 90 and 95% confidence intervals "by hand" (i.e. follow the equations in the text and don't use the built-in tests).

In [670]:
```
val=df[if.==2,"inUcastPkts"]
z_90=qnorm(1-0.1/2)
z_95=qnorm(1-0.05/2)
n=length(val)

ci_90u=mean(val)+((z_90)*(sd(val)/sqrt(n)))
ci_90l=mean(val)-((z_90)*(sd(val)/sqrt(n)))
ci_90=c("CI for 90%",ci_90l,ci_90u)
ci_95u=mean(val)+((z_95)*(sd(val)/sqrt(n)))
ci_95l=mean(val)-((z_95)*(sd(val)/sqrt(n)))


ci_95=c("CI for 95%",ci_95l,ci_95u)
ci_90
ci_95
```

Out[670]: "CI for 90%" "1404.54158165422" "1698.61631308262"

Out[670]: "CI for 95%" "1376.37311965255" "1726.78477508429"

Problem #7

Using paired confidence intervals (you can use the t.test function -- see SimpleR, page 69) to argue whether links 3 and 6 have a statistically identical number of total packets (input + output, all types) per sampling period at the 90, 95 and 99% confidence level. Show the calculations and justify your answer.

In [671]:
```
Packets3=df[if.==3,"outUCastPkts"]+df[if.==3,"inUcastPkts"]
Packets6=df[if.==6,"outUCastPkts"]+df[if.==6,"inUcastPkts"]
t.test(Packets3,conf.level=0.90) #854.5338 937.7820
t.test(Packets6,conf.level=0.90) #762.4874 843.8284
t.test(Packets3) #845.7278 946.5880
t.test(Packets6) #753.8832 852.4326
t.test(Packets3,conf.level=0.99) #827.0644 965.2514
t.test(Packets6,conf.level=0.99) #735.6474 870.6684
print ("From the CI calculations, Links don't have identical number of packets")
```

Out[671]:
```
        One Sample t-test

data:  Packets3
t = 37.334, df = 18, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 854.5338 937.7820
sample estimates:
mean of x
 896.1579
```

Out[671]:
```
        One Sample t-test

data:  Packets6
t = 34.2442, df = 18, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
90 percent confidence interval:
 762.4874 843.8284
sample estimates:
mean of x
 803.1579
```

Out[671]:
```
        One Sample t-test

data:  Packets3
t = 37.334, df = 18, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 845.7278 946.5880
sample estimates:
mean of x
 896.1579
```

Out[671]:
```
        One Sample t-test

data:  Packets6
t = 34.2442, df = 18, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 753.8832 852.4326
sample estimates:
mean of x
 803.1579
```

Out[671]:
```
        One Sample t-test

data:  Packets3
t = 37.334, df = 18, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
99 percent confidence interval:
 827.0644 965.2514
sample estimates:
mean of x
 896.1579
```

Out[671]:
```
        One Sample t-test

data:  Packets6
t = 34.2442, df = 18, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
99 percent confidence interval:
 735.6474 870.6684
sample estimates:
mean of x
 803.1579
```

```
[1] "From the CI calculations, Links don't have identical number of packets"
```

Problem #8

Use the R "bootstrap" method (see Boudec Algorithm 2.1 and http://www.statmethods.net/advstats/bootstrapping.html (http://www.statmethods.net/advstats/bootstrapping.html)) to compute the confidence interval for the MEDIAN number of total packets (input + output, all types) per sampling period for link 3 at the 90 and 95 level.

In [672]:
```
library(boot)
Packets3=df[if.==3,"outUCastPkts"]+df[if.==3,"inUcastPkts"]
#wilcox.test(Packets3,conf.int=TRUE)

rsq <- function( data, id) {
  val <- data[id]
  c(median(val)-sd(val)/mean(val), median(val)+sd(val)/mean(val))
}

bootobj=boot(data=Packets3,statistic=rsq,R=1000)
boot.ci(bootobj, type="basic",conf=c(0.90,0.95))
```

Out[672]:
```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = bootobj, conf = c(0.9, 0.95), type = "basic")

Intervals :
Level      Basic
90%   (851.8, 945.9 )
95%   (846.9, 945.9 )
Calculations and Intervals on Original Scale
```

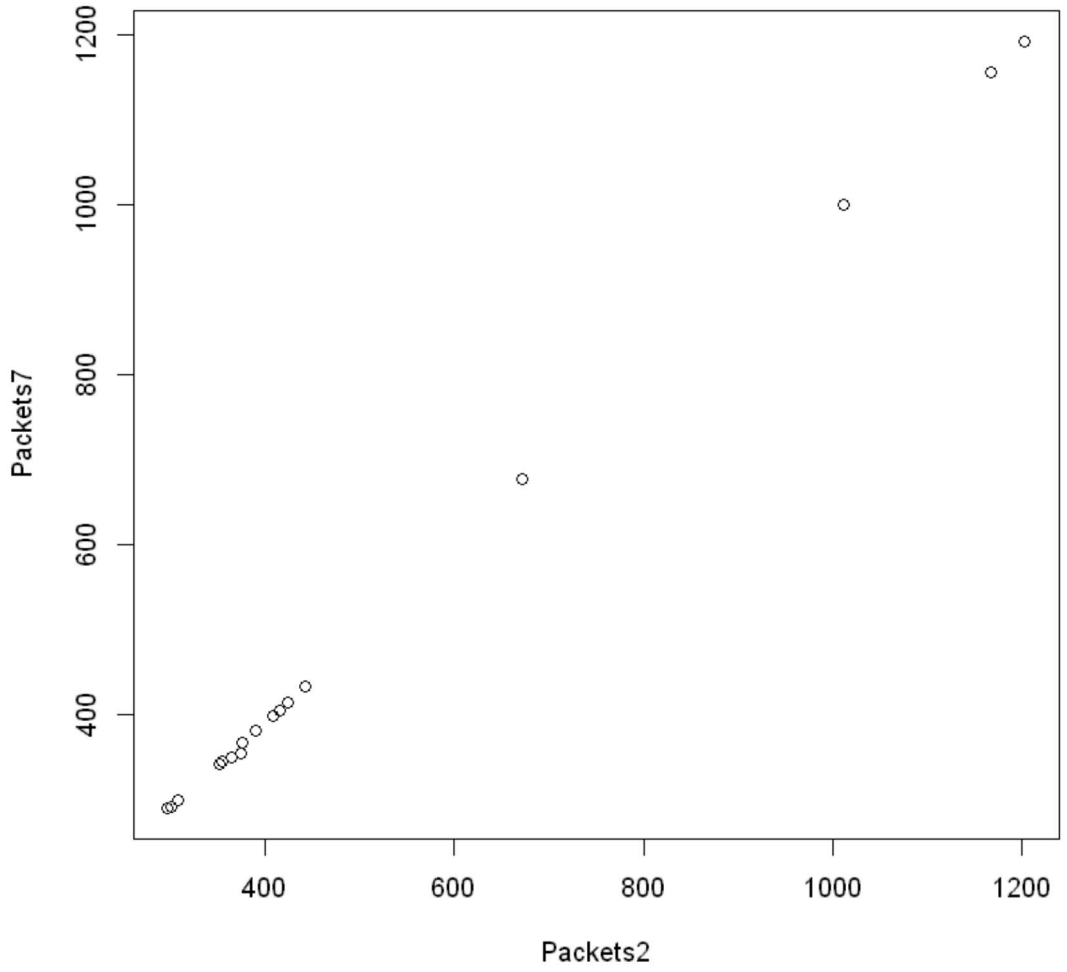
Problem #9

Do the output packets of links 2 and 7 have a similar distribution of packets per samping period? One way to determine this is to use the correlation function. Another is to produce a qqplot showing the relationship of one link to another. Produce such a plot and include it in your report.

In [673]:
```
Packets2=df[if.==2,"outUCastPkts"]
Packets7=df[if.==7,"outUCastPkts"]
cor(Packets2,Packets7)
require(ggplot2)
qqplot(Packets2,Packets7)
print("since slope is 1, both are higly correlated")
```

Out[673]:
```
0.999831800299426

[1] "since slope is 1, both are higly correlated"
```



Problem #10

This question does not use the data, but you need to produce an R program. Rather than record the full vector of data when computing a mean, you can use a pair of variables -- you simply record the sum of the terms in the vector (Sx) and the number of items (n) and then report Sx/n. Rearrange the equations equation for variance ($\sum_i^n (x_i - \bar{x})^2$) to demonstrate how you can compute the variance of an arbitrary number of data samples using a small, fixed number of variables, much as I demonstrated for computing the mean. Your solution will need to record three values from the data. You should show your derivation step-by-step. You can either do this using code (i.e. define a series of MyVar() functions that simplify expression into terms that don't involve the mean) or using math (notebooks support MathJax and LaTeX).

In [674]:
```
variance1=function(data) {
    n=0;Sx=0;v=0
    for (i in data) {
        n=n+1
        Sx=Sx+i
    }

    Mean=Sx/n

    for (i in data) {
        v=v+((i-Mean)^2)
    }
    Variance=v/n
    return (c(n,Sx,Variance))
}

v=variance1(c(1,2,4,5,6))
v
```

Out[674]:
```
5  18  3.44
```