

## STAT 457, Fall 2010

### Lab #10: Confidence Intervals and Q-Q Plots

Topics:

- We will use R along with a plagiarised program, to see if the Central Limit Theorem is true!
- Q-Q plots are another useful tool for finding out characteristics of a distribution, namely how the target distribution compares to a known distribution. I recommend you put this plot into your “arsenal” of statistical investigative tools, along with histograms, summary stats, plots, frequency tables, etc., as you perform descriptive statistics.
- We will sample from a distribution to determine confidence intervals and determine if the Central Limit Theorem is valid!!

=====  
Below is a script which plots samples of various sizes from the uniform distribution It is saved as script clt\_demo.R .

**[1]** Run the script a few times, and comment on the following items: comparison of centers, comparisons of spread, comparisons of shape, and how these results reinforce the Central Limit Theorem.

```
# Central Limit Theorem Demonstration

# set up plot window

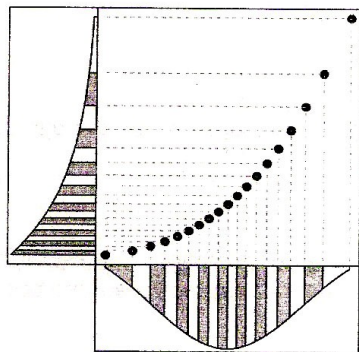
plot(0, 0, type="n", xlim=c(0,1), ylim=c(0, 15.5),
     main="CLT for n=2,10,25,100", xlab="Density estimate",
     ylab="f(x)")

# create plots
m <- 500 ; a <- 0 ; b <- 1
res <- c() ; n <- c(2, 10, 25, 100)
for(i in 1:4) {
  for(j in 1:m) {
    res[j] <- mean(runif(n[i], a, b))
  }
  lines(density(res), lwd=2)
}
```

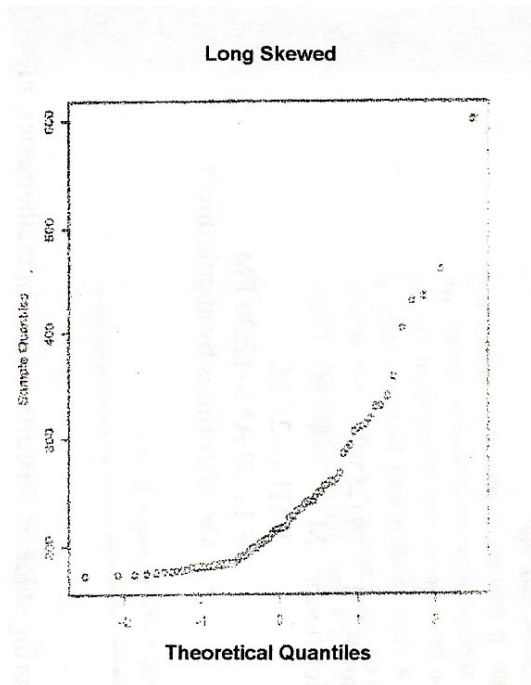
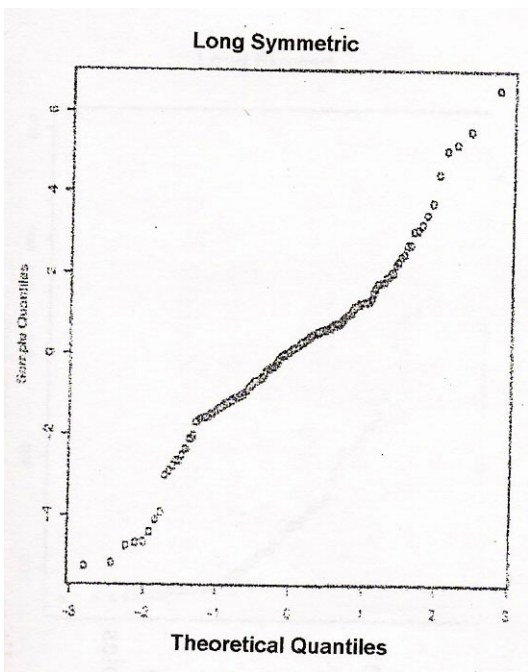
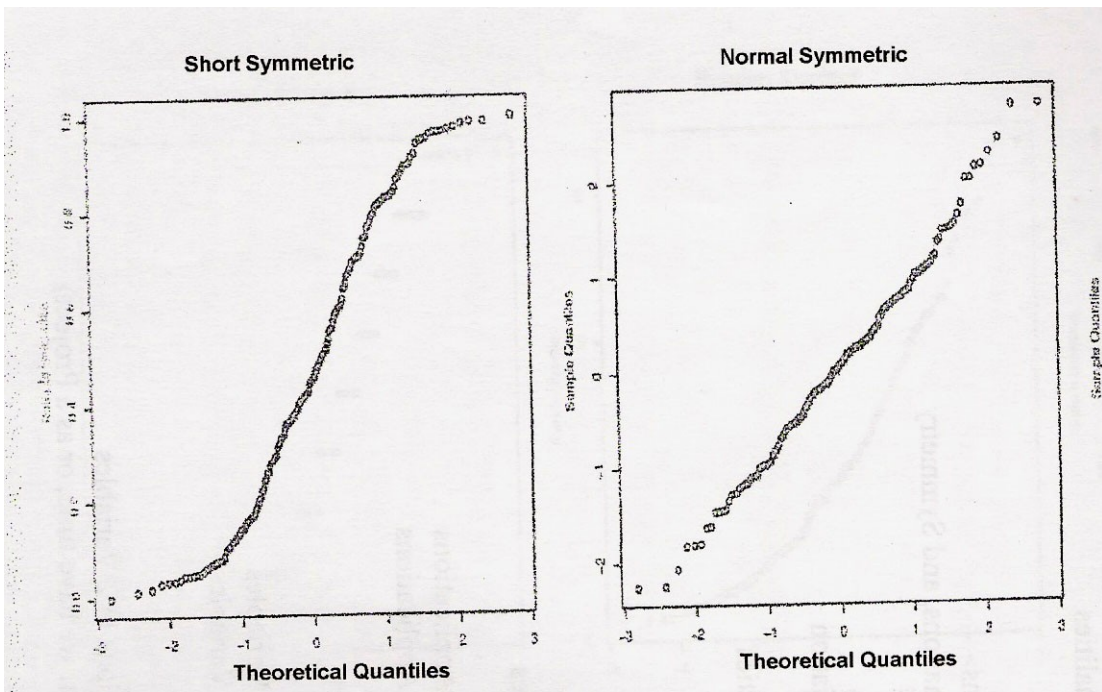
=====  
In Statistics, a Q-Q plot (“Q” stands for quantile) is a graphical method for diagnosing differences between the probability distribution of a statistical population from which a random sample has been taken and a comparison distribution. A typical example of the kind of differences that can be tested for is non-normality of the test distribution.

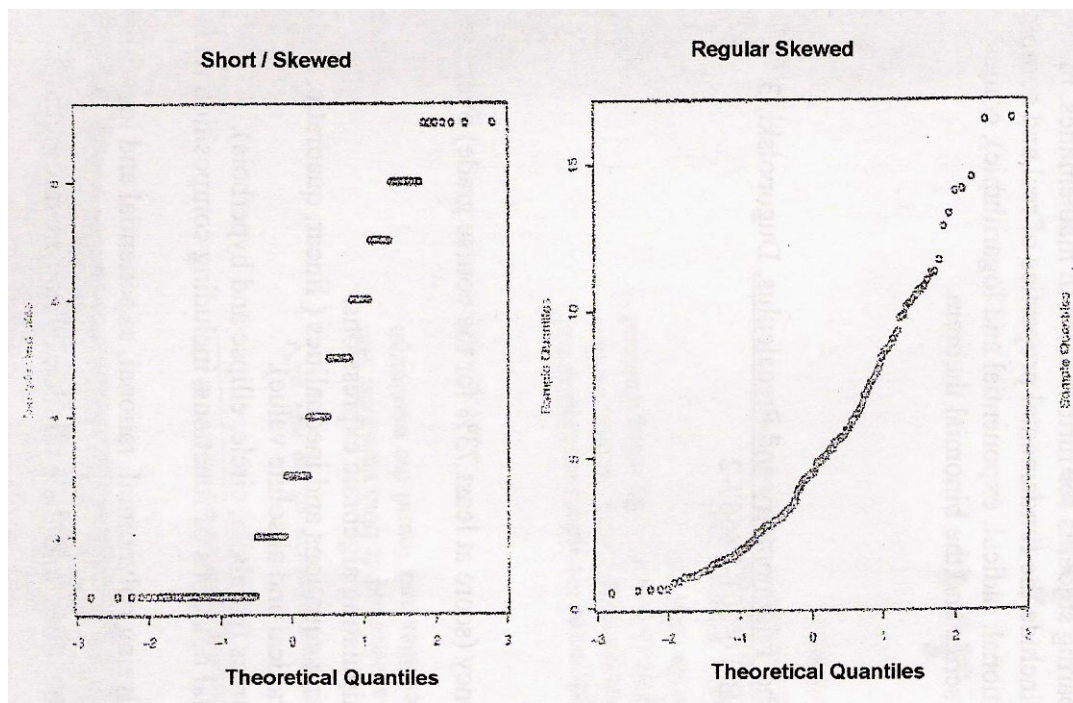
A Q-Q plot plots the quantiles of one distribution against the quantiles of another as points. If the distributions have similar shapes, the points will fall roughly along a straight line. If they are different, the points will not lie on a line, in a manner than can indicate why not.

A *normal quantile plot* plots the quantiles of a data set against the quantiles of a “benchmark distribution”, which is the normal distribution. So, the Q-Q plot is a sort of visual test to see if a distribution is approximately normally distributed. Again, the basic idea is that if the data set is similar to the benchmark one, then the graph will essentially display a straight line. If not, then the line will be “curved” in a manner that can be interpreted from the graph. The picture below shows the Q-Q plot for two theoretical distributions that are clearly not the same shape. The benchmark normal distribution lies on the x axis. Each shaded region is 5% of the total area (note that 2 areas are missing in the diagram). The difference in the shapes produces differences in the quantiles that create a curve in the Q-Q plot.



The figure below shows 6 *normal quantile plots* for data that are a combination of symmetric, skewed right, and short, normal or long tailed. The combination (normal/symmetric) gives a straight line. Were we to plot a histogram of these data, we would see the familiar bell-shaped curve. The figure (short/symmetric) shows what happens with short tails. In particular, if the right tail is short, it forces the quantile graph to curve down. In contrast the graph (long/skewed) curves up, as this data has a long right tail. Investigate the other pictures to see how the various shapes stereotypically appear in a Q-Q plot.





[2] Open R Commander (or use R-Gui if you prefer) and create `norm1`, which is a sample of 100 randomly picked values from the Normal (3,2), also called  $N(3,2)$ , distribution. This is done by clicking on **DISTRIBUTION CONTINUOUS NORMAL DIST SAMPLE FROM NORMAL**. Then, in the dialogue box, make the name of the distribution `norm1`, then mean of 3, sigma of 2, 100 rows and 1 column. Be sure to uncheck the “sample means” box. This will create a sample of 100 from the  $N(3,2)$  distribution. Now, create a histogram of `norm1` by clicking on **GRAPHS HISTOGRAM**. Finally, create a Q-Q plot by clicking on **GRAPHS QUANTILE COMPARISON PLOT** and then OK, because the box is ready to compare your distribution to a normal. If you prefer using R-Gui instead of R Commander, you can generate the sample with the command `norm1 <- as.data.frame(matrix(rnorm(100, mean=3, sd=2), ncol=1))`. You can get the histogram with `hist(norm1[,1])` and get the Q-Q plot with `qq.plot(norm1[,1], dist="norm", labels=FALSE)`. Now, comment on the shape in the Q-Q plot, and if it is close to a normal shape. Use the histogram to help you decide.

[3] Create a sample of 100 randomly picked values from the Chi Squared distribution, which (with 3 degrees of freedom) is rather skewed. In R Commander, click on **DISTRIBUTIONS CONTINUOUS CHI SQUARED SAMPLE FROM CHI SQUARED**, then enter the name as `chisq1`, then 3 degrees of freedom, 100 rows and 1 column—be sure to uncheck the “sample means” box. Next, create a histogram of this `chisq1` and a normal quantile plot (Q-Q plot) of `chisq1`. Note that we are again comparing `chisq1` to the normal distribution—and we should have a different Q-Q characteristic. If you prefer to use R Gui, the command to get a sample is

```
chisq1 <- as.data.frame(matrix(rchisq(100, df=3), ncol=1))
```

Histogram and Q-Q plot commands are similar to those shown before.

Comment on what the Q-Q plot indicates, as you did in the previous question.

=====

The Central Limit Theorem (CLT) tells us that if we keep taking constant samples of size  $n$  (big enough), from a population centered at a parameter proportion =  $p$  (centralized enough), then the distribution of resulting sample proportions ( $\hat{p}$ ) will be approximately normally distributed, with center  $p$  and standard deviation =  $\sqrt{\frac{p*(1-p)}{n}}$ . When we say that  $n$  has to be big enough and  $p$  has to be centralized enough, we sort of sum that up by stating  $np > 10$  and  $n(1-p) > 10$ .

The information from the CLT allows us to develop the formula for confidence

$$\text{intervals: } CI = \hat{p} \pm z \sqrt{\frac{\hat{p}*(1-\hat{p})}{n}}$$

Let us use this formula to sample from the distribution centered at  $p = 0.27$ . We will take 100 samples of size  $n_{10} = 10$ ,  $n_{30} = 30$ , and  $n_{100} = 100$ . Using our formula, we have the 95% CI's shown below.

$$\text{For } n_{10}, CI = .27 \pm 1.96 \sqrt{\frac{.27*.73}{10}} = (-.0052, .5452)$$

$$\text{For } n_{30}, CI = .27 \pm 1.96 \sqrt{\frac{.27*.73}{30}} = (.1111, .4289)$$

$$\text{For } n_{100}, CI = .27 \pm 1.96 \sqrt{\frac{.27*.73}{100}} = (.1830, .3570)$$

We have the script `confidence_intervals.R` listed below, to demonstrate some of these concepts.

```
# Confidence Intervals

# initiate constants
n.samples <- 100
n <- 10
p1 <- .28

# create samples and histogram of phats
distrib1 <- matrix(rbinom(n.samples*n, size=1,
  prob=p1), ncol=10)
sample1 <- rowMeans(distrib1[,1:n])
hist(sample1)

# compute CI's
summary(sample1)
sd(sample1)
```

```

diff <- 1.96 * sqrt(p1*(1-p1)/n)
lower <- p1 - diff
upper <- p1 + diff
print("95% CI is")
lower ; upper
hist(sample1)
abline(v=lower,lwd=4)
abline(v=upper,lwd=4)

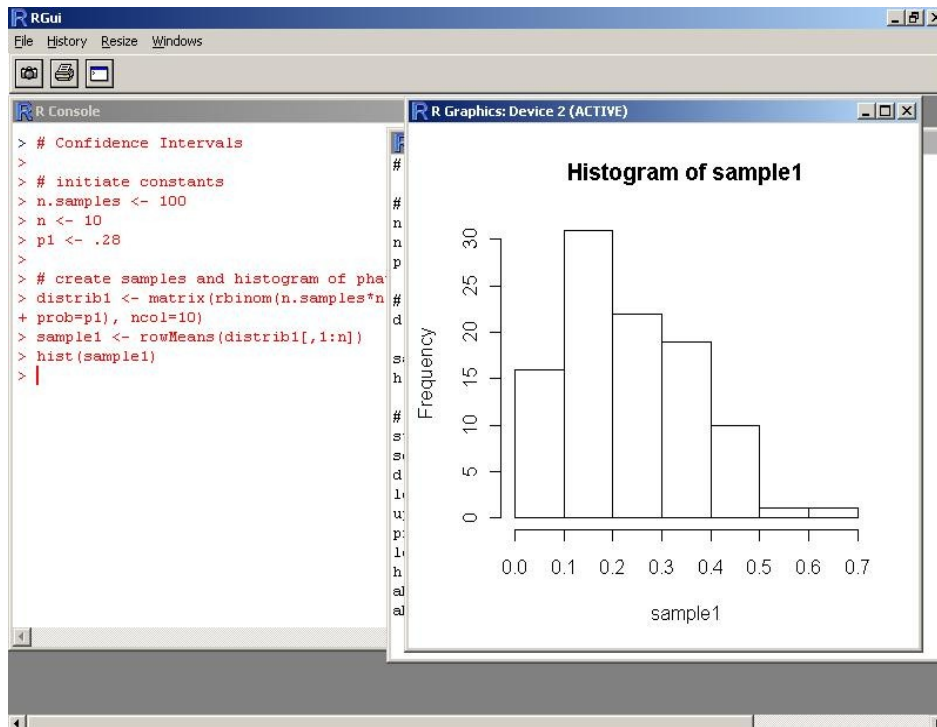
# count proportion of CI's not capturing p
test1 <- which(sample1-diff > p1)
length(test1)
cat("proportion of ci's not capturing",
    p1, "are", length(test1)/n.samples,"\n")
cat("sample interval numbers not capturing p are",
    which(sample1-diff > p1), "\n")

# plot ci's

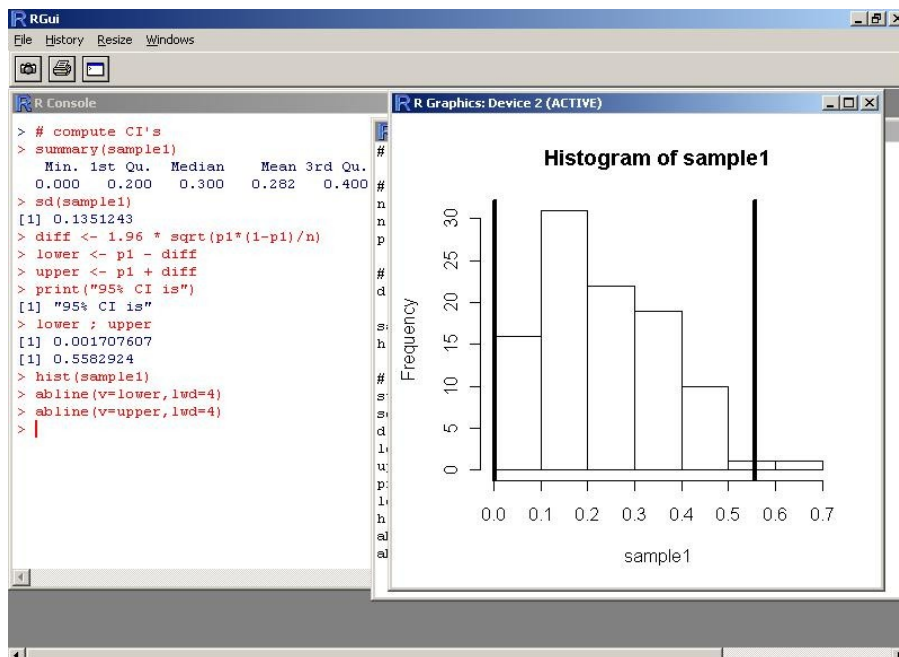
x1 <- 1:n.samples
y1 <- sample1 - diff
y2 <- sample1 + diff
x <- c(x1,x1)
y <- c(y1,y2)
plot(x,y)
abline(p1,0)
lines(x1,y1)
lines(x1,y2)

```

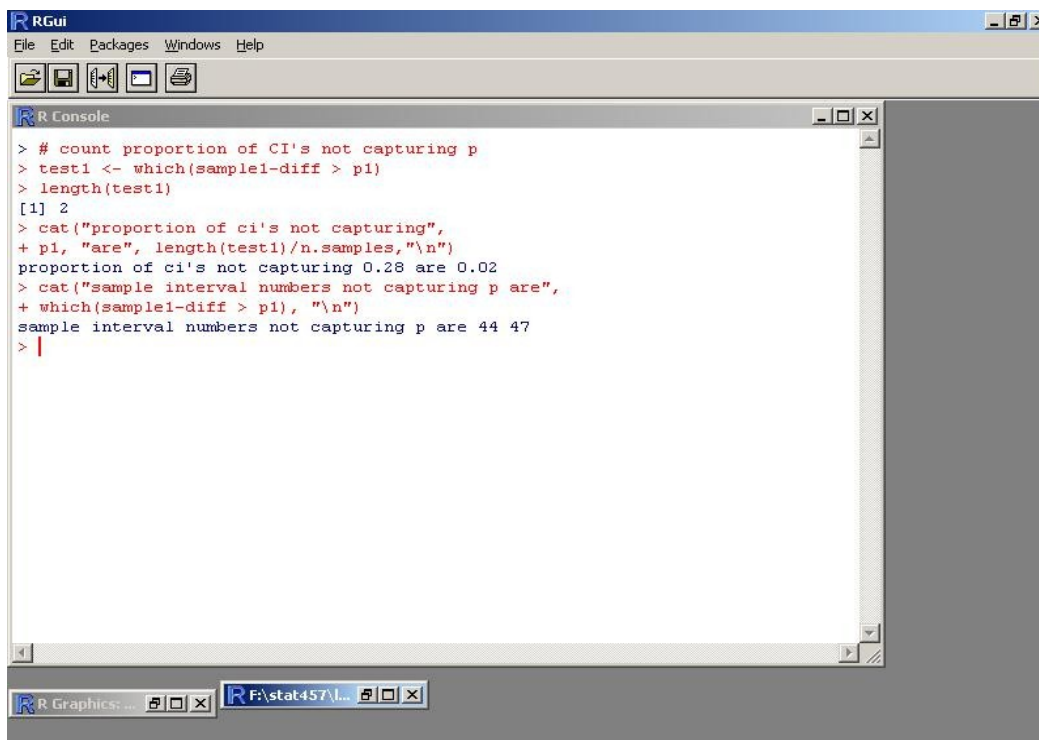
Below is a sample run of this program.



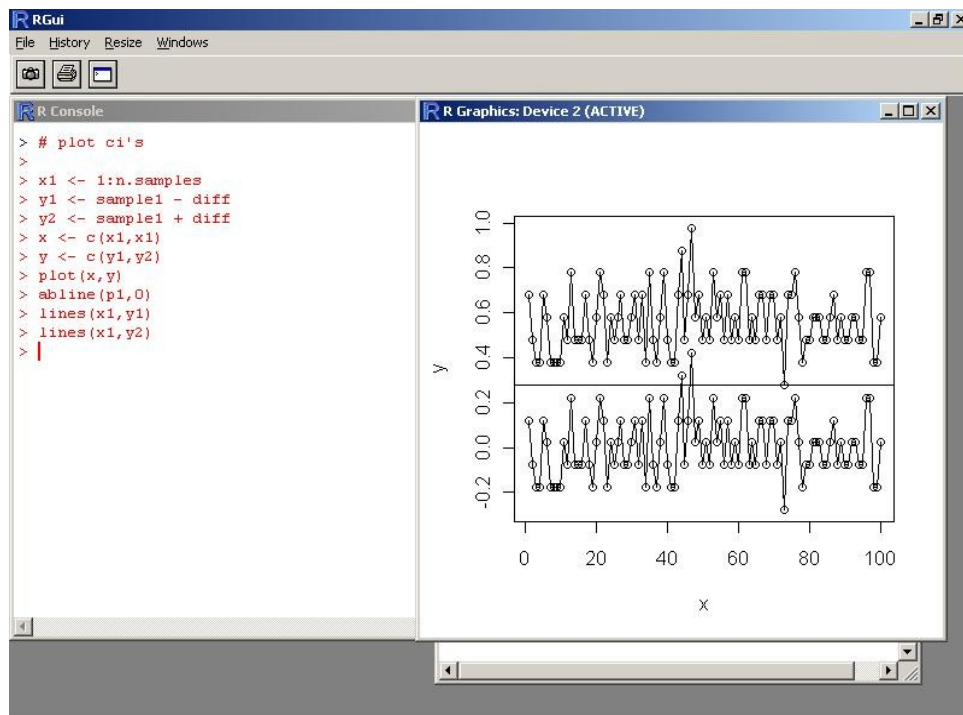
Next is the placement of the confidence interval value “fences”.



Next is the blue output saying that in this run, we failed to capture the parameter of  $p=0.28$  twice, on CI number 44 and number 47.



Next is a plot of the 100 confidence intervals, and you see how we missed on CI 44 and 47. We almost missed on CI number 73 or 74 (I can't tell exactly which).



[4] Reproduce 5 runs of this script, showing “fenced” histograms, blue output on missed CI's, and CI plots. Then produce 5 runs each (with the same output) where  $n=30$  and  $n=100$ . Compare results with each other and with your  $n=10$  runs.