Stat 135, Fall 2006 A. Adhikari

HOMEWORK 5 SOLUTIONS

1. $\bar{X}$ has the normal distribution with mean 0 and standard error $10/\sqrt{25} = 2$. So the values of $\bar{X}$ are, with overwhelming probability, going to be in the range -8 to 8. To plot the density:

> x ← seq(-8, 8, by=0.1)

> plot(x, dnorm(x, 0, 2), type="l")

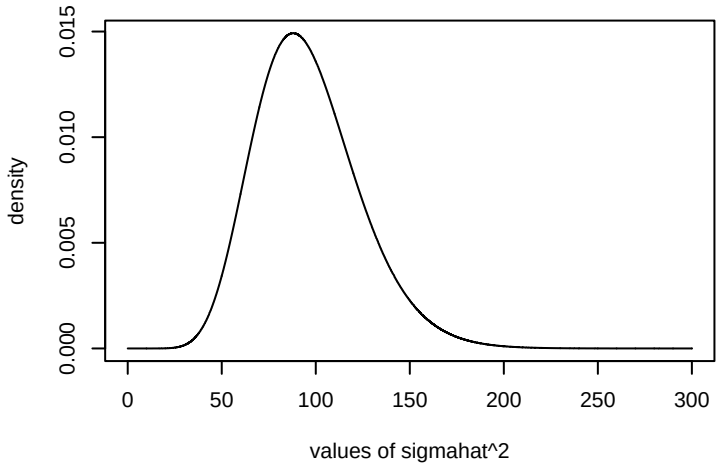
You know that $n\hat{\sigma}^2/\sigma^2$ has the chi-squared density with 24 degrees of freedom. Plug in the values of n and σ^2 to see that $\hat{sigma}^2 = 4X$ where X has chi-squared density with 24 degrees of freedom. By the change of variable formula, the density f of $\hat{\sigma}^2$ is calculated as

$$f(y) \quad = \quad \frac{1}{4} f_X(y/4), \qquad y > 0$$

where f_X is the chi-squared density of X. Since $E(X) = 24$ (the expectation of a chi-squared variable is its degrees of freedom), you know that $E(\hat{\sigma}^2) = 96$, which gives you some sense of the values to use for the horizontal axis of your plot:

> y ← seq(0, 300, by=0.1)

> plot(y, dchisq(y/4, 24)/4, type="l")



2. We have a population size of N (which is the parameter we're trying to estimate), $G = 100$ "good elements" (i.e. tagged fish) in the population, and a simple random sample of size $n = 50$. We have observed X, the number of good elements in the sample.

So X has hypergeometric distribution with $E(X) = n\frac{G}{N}$. So $N = n\frac{G}{E(X)}$ and the MOM estimate is $\hat{N} = n\frac{G}{X}$, whose observed value is 250.

Example I of Sec 1.4.2 shows (after some translation of the notation) that $\hat{N}$ above is also the MLE of N.

3a. You have observed two exponential variables X_1 and X_2, and an indicator: $I(X_3 > 10)$. The likelihood function is

$$lik(\lambda) \quad = \quad (\lambda e^{-\lambda X_1}) \cdot (\lambda e^{-\lambda X_2}) \cdot e^{-\lambda 10}$$

where the third factor is $P(X_3 > 10)$.

b. The log-likelihood is

$$l(\lambda) \;=\; 2\log\lambda \;-\; \lambda(X_1 + X_2 + 10)$$

which is maximized by (the usual differentiation etc)

$$\hat{\lambda} \;=\; \frac{2}{X_1 + X_2 + 10}$$

The observed value of the estimate is 1/9.

4a. According to *R*, the mean of the sample is 3.61, roughly, so that's your guess for μ. The variance of the sample is $S^2 = 3.418$, roughly. That's your estimate for σ^2. I have no problem with you using $\hat{sigma}^2$ here, provided you fix things when you make the confidence interval below.

b. This is a confidence interval based on the t distribution with 15 degrees of freedom: $3.61 \pm 1.753 \times 1.8488/sqrt16$. The interval is $(2.8, 4.42)$, roughly.

c. You know how to get a confidence interval for σ^2, so do that first. You need the upper and lower 5% points of the $\chi^2_1 5$ distribution: these are 24.996 and 7.261 respectively. You also need $n\hat{\sigma}^2$ which is equal to $(n-1)S^2$ which is $15 \times 3.418 = 51.27$. The endpoints of the confidence interval for σ^2 are therefore $51.27/24.996 = 2.05$ and $51.27/7.261 = 7.061$.

Since the square root is a one-to-one function, the confidence interval for σ is obtained just by taking square roots of the endpoints above. The interval is $(1.432, 2.657)$. No, the negative square root is not a problem. **You** figure out why.

d. The sample size has to be multiplied by $2^2 = 4$, i.e. sample size around 64, in order to halve the length of the interval. That is because the sample size appears as $\sqrt{n}$ in the denominator of the standard error of $\bar{X}$. You can do a bit better because the value of t gets smaller as n increases, but the sample size is still going to be around 60 or so.

5. Set up notation first. The variable with known expectation and variance is Y/n. This is what we were calling X when we developed the method in class. Y/n has expectation $\mu = p_0 = e^{-\lambda}$ and variance $p_0(1-p_0)/n = e^{-\lambda}(1-e^{-\lambda})/n$.

The function to use is $g(u) = -\log(u)$. So $g'(u) = -1/u$ and $g"(u) = 1/u^2$.

By the δ method, the expectation of our estimate is approximately

$$g(\mu) \;+\; \frac{e^{-\lambda}(1-e^{-\lambda})}{2n} g"(\mu) \;=\; -\log(e^{-\lambda}) \;+\; \frac{e^{-\lambda}(1-e^{-\lambda})}{2n} \cdot \frac{1}{e^{-2\lambda}} \;=\; \lambda \;+\; \frac{e^{\lambda}-1}{2n}$$

So the bias is approximately $(e^{\lambda}-1)/2n$.

By the δ method again, the variance of our estimate is approximately

$$\frac{e^{-\lambda}(1-e^{-\lambda})}{n} \cdot \frac{1}{e^{-2\lambda}} \;=\; \frac{e^{\lambda}-1}{n} \;=\; \frac{1}{n}[\lambda + \lambda^2/2! + \lambda^3/3! + \cdots]$$

noindent The maximum likelihood estimate of λ is $\bar{X}$, our old friend the sample mean. This is easy to guess, and the text derives it formally in Example A of Section 8.5. Its variance is λ/n because the Poisson variance is λ. It's clear that $\bar{X}$ is more efficient, by a factor of

$$1 + \lambda/2! + \lambda^2/3! + \cdots$$

6. Freebie.

7. The commands below will clean up the data as required.

> x ← read.table("http://www.stat.berkeley.users/ani/s135f06/HW5data1.txt", header=TRUE)

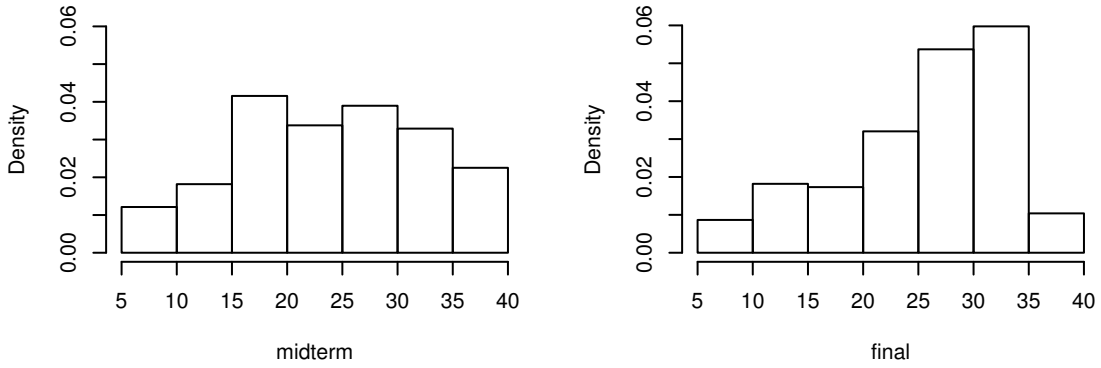
> scores ← cbind(x[,2], x[,1])

> scores ← scores[scores[,1]>0 & scores[,2]>0,]

> scores[,1] ← 2*scores[,1]

a. The commands below will draw the histogram of the midterm scores with the data grouped into fives. Replace 1 by 2, and change the labels appropriately, to get the histogram of final scores.

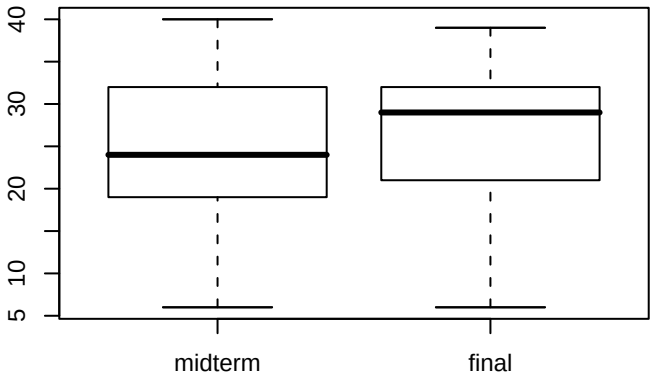
> hist(scores[,1], breaks=6, prob=TRUE, xlab="midterm", main="")



b. The boxplots are obtained by

> boxplot(data.frame(scores), names=c("midterm", "final"))

The boxplots are quite consistent with the histograms: the final has a higher median and has a left-hand tail.

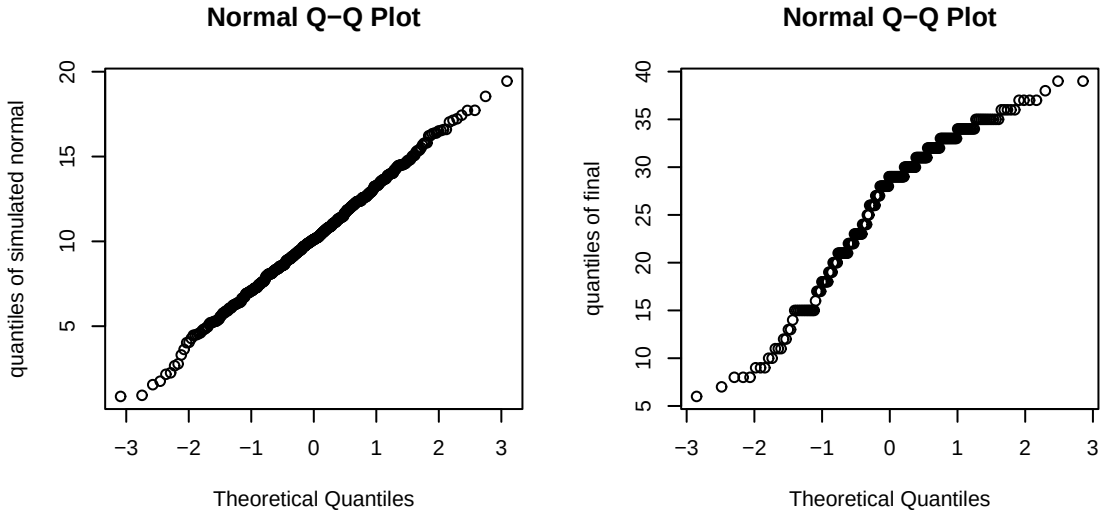


8. The normal qq-plot is linear when the data are the simulated normals, and clearly non-linear when the data are the final exam scores. That's no big surprise, since the distribution of final exam scores is clearly non-normal.

> simnorm ← rnorm(500, 10, 3)

> qqnorm(simnorm, ylab="quantiles of simulated normal")

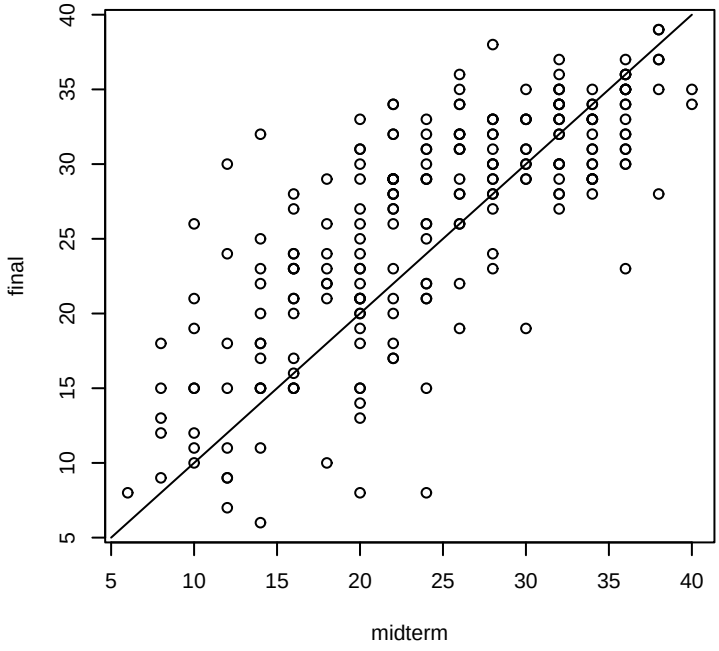
> qqnorm(scores[,2], ylab="quantiles of final")



9. The stemleafs are easy; I won't draw them all. The one with `scale=2` is too long and too detailed. It's a toss-up between `scale=0.5` and `scale=1`. I prefer the former because I can recover all the data from it. The command used for that one was:

> stem(scores[,2], scale=0.5)

10. By now you don't need me to tell you how to do the plot.



a) It's clear that more than half gained from the grading scheme, because most of the points

are above the line.

b) 137 students out of 231, more than half.

> sum(scores[,2] > scores[,1])

c) The line is too steep. E.g. if you look at the points corresponding to a midterm score of about 35, the line is clearly at the high end of the corresponding final scores. And look at midterm scores around 10 or so. The line is too low there. So I'd flatten the line a bit.

By how much? Wait till we've done regression.