DIRECTIONS: Write your solutions in a single **.tex** file, including R code. Your **.tar** package will consist of that file, its output **.pdf** file, and a separate file for each problem requiring R code, each file named in the form **x.R**, where x is the problem number. Name your **.tar** file as you did in the homework, but with your own address only. Your submission must be in the 256quiz2 directory in handin, **timestamped on or before 3:00 p.m. NO LATE SUBMISSIONS; keep submitting the work you have, as you go along, so that you at least have something turned in.** You are not necessarily expected to solve all the problems.

1. (35) Consider the "Wharton experiment," p.427 of our book. Write a function with "declaration"

```
wharton <- function(indata,sg,nnoise)
```

that performs this experiment on any data frame **indata**. Here **nnoise** is the number of $N(0, \sigma^2)$ noise variables to be added, where σ is **sg**.

It is assumed that the response variable is the one in the final, i.e. rightmost, column of **indata**.

There will be no return value. Instead, your function will make a call **print(summary(lm()))** to run the regression and print out the results, replete with asterisks. (You may not get a lot of them.)

You may wish to convert **indata** to a matrix within your code.

Place your code in **1.R**, and a listing in your **.tex** file.

2. (35) Suppose

$$m_{Y;X}(t) = 2t \quad \text{(as in (22.1))} \tag{1}$$

$$Var(Y \mid X = t) = t \tag{2}$$

and that X has a uniform distribution on (0,1). Find $Var(Y)$.

For full credit, have no explicit integrals.

3. (30) Suppose X has support (0,1), with density

$$f_X(t) = \begin{cases} c, & \text{if } t \in (0, q) \\ 2c, & \text{if } t \in (q, 1) \end{cases} \tag{3}$$

This is a 1-parameter density family, with the parameter q. Note that c is a constant to make the function integrate to 1; it is not a second parameter.

Write a function with "declaration"

```
gmmq <- function(x,initq)
```

which uses the **gmm** package (this is required) to estimate q from the data vector **x** and an initial guess **initq**. The return value of the function will be the object that **gmm()** returns. You'll need to use the recipe given to Nick by the author of **gmm()**.

Note: You must include your mathematical derivation in your **.tex** file, and your R code in both that file and **3.R**.

Solutions:

1.

```
wh <- function(xy,sg,nnoise) {
   n <- nrow(xy)
   p <- ncol(xy)
   xy <- cbind(xy,matrix(sg*rnorm(n*nnoise),ncol=nnoise))
   xy <- as.matrix(xy)
   print(summary(lm(xy[,p] ~ xy[,-p])))
}
```

2.

$$\begin{aligned} Var(Y) &= E[Var(Y|X)] + Var[E(Y|X)] \quad (4) \\ &= EX + Var(2X) \quad (5) \\ &= 0.5 + 4(1/12) \quad (6) \end{aligned}$$

3.

```
gmmq <- function(x,initq) {
   if (is.vector(x)) x <- matrix(x,ncol=1)
   g <- function(th,x) {
      q <- th[1]
      c <- 1/(2-q)
      c*q^2/2 + c * (1-q^2) - x
   }
   gmm(g,x,c(q = initq),lower=0,upper=1, method='Brent')
}
```