1. Faraway (1st edition) problem 6.1

2. Faraway (1st edition) problem 6.5 (you can find the function names for LAD, Huber, etc regression in the robust regression code example from lecture 10)

3. Multiple testing simulation. First generate data:

   - Generate $X \in \mathbb{R}^{n \times p}$ with i.i.d. $N(0,1)$ entries, then normalize the columns to have unit norm.
   - Let the true coefficient vector be $\beta = (\underbrace{5,\dots,5}_{10\text{ times}}, \underbrace{0,\dots,0}_{p-10\text{ times}})$.
   - Generate $Y = X\beta +$ (noise) where the noise values are i.i.d. $N(0,1)$.

   Run this simulation with $n = 400$ and try each value $p = 200, 400, 600, 800$. You may want to run both parts of the simulation a few times to get a clear picture of what the results typically look like, since these plots may be fairly noisy, especially for the second part.

   (a) First, run forward stepwise starting with a model of size 0 (intercept only) and up to a model of size 30. (You can either implement forward selection "by hand", or use pre-existing R code as long as you?re certain it's doing the same thing.) Evaluate the BIC at each model size, and plot BIC against model size. Repeat this for each choice of $p$, the total number of available covariates. Compare the resulting plots you see. How does BIC perform when $p$ is smaller—does it do a good job of picking an appropriate model? What goes wrong as $p$ increases?

   (b) Next, we'll do this again but we'll use a held-out validation set to test the model. Split your data at random into 200 training points and 200 validation points. Run forward stepwise on the 200 training points to obtain a model of size 0, a model of size 1, ... , a model of size 30. Then evaluate the prediction error RSS of each model on the validation set (i.e. using the selected subset & fitted coefficients $\hat{\beta}$ from the training set). Plot prediction error against model size. What do you see? How do these results compare to BIC?

4. Consider the following iteratively reweighted least squares (IRLS) algorithm:

   - Solve least squares (weights $w_i = 1$) to get $\hat{\beta}$.
   - Update the weights by setting $w_i = 1/\sqrt{|Y_i - X_i^\top \beta|}$
   - Solve WLS to get a new $\hat{\beta}$
   - Iterate the last two steps until convergence

   (a) What is the M-estimator that this IRLS procedure is trying to solve? That is, the procedure above is designed to minimize

   $$\sum_{i=1}^{n} \ell(Y_i - X_i^\top \beta)$$

   for what loss function $\ell$?

   (b) Using your answer from above, how would this loss $\ell$ compared to existing options, specifically least squares & LAD & Huber, in terms of robustness to outliers?