Statistical Computing (140.776)
Fall 2004

Homework Assignment #2
(Due Wednesday, October 6, 2004)

To get full credit for your solution of the homework problems, please:

- Hand in a hard copy of your code, results, and plots.
- Send an electronic version of all of your code to Benilton (bcarvalh@jhsph.edu). Make sure the code is not platform dependent, and can be run at any other machine in any subdirectory.
- Send Benilton a file called `YOURNAME.functions.R` that countains the functions you wrote for problems 1(a), 2(a) and 3(d). We will need to be able to `source` that file and run your functions on some data.

1. (a) Write a function to create a contingency table of adjacent k-tuples in a string of characters from the set {`A,C,G,T`}. For example, with k=3 and with the string '`CAGACAAAAC`', you would want to produce the following table:

AAA	AAC	ACA	AGA	CAA	CAG	GAC
2	1	1	1	1	1	1

   (b) To check your function, run it on a simulated string of 10,000 characters, drawn uniformly and independently from the set {`A,C,G,T`}.

2. Linear models have the form $\mathbf{Y} = \mathbf{X}\beta + \varepsilon$, where $\mathbf{Y}$ is the resonse vector of length n, β is the vector containing the p parameters, $\mathbf{X}$ is the design matrix (i.e. the matrix with the predictors as columns) of dimension $n \times p$, and ε represents gaussian noise with mean zero and variance σ^2. The parameter estimates are $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$, ($\mathbf{X}'\mathbf{X}$ stands for the matrix multiplication of the transpose of $\mathbf{X}$ with $\mathbf{X}$, and $(\mathbf{X}'\mathbf{X})^{-1}$ is the inverse of that product), an estimate for the variance is $\hat{\sigma}^2 = (\mathbf{Y} - \mathbf{X}\hat{\beta})'(\mathbf{Y} - \mathbf{X}\hat{\beta})/(n-p)$, an estimate for the covariance matrix of the parameter estimates is $\widehat{\text{cov}}(\hat{\beta}) = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$, the projection ("hat") matrix is $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, the fitted values are $\hat{\mathbf{Y}} = \mathbf{P}\mathbf{Y}$, and the residuals are $\hat{\mathbf{Y}} - \mathbf{Y}$.

   (a) Write a function that takes the response vector $\mathbf{Y}$ and the matrix of covariates $\mathbf{X}$ as input, and returns a list of the following:
   - **beta**, the vector of least squares estimates,
   - **sigma**, the residual standard error,
   - **varbeta**, the covariance matrix of the least squares estimates,
   - **fitted**, the vector of fitted values,
   - **residuals**, the vector of residuals.

   Further, put in an option to return **hat**, the projection matrix, upon request. The default should be to not return it. To fit an intercept, the elements in the first column of $\mathbf{X}$ have to be equal to one, so your function should also have an option to add a vector of ones to the matrix with the predictors. Further, your function should check whether or not $\mathbf{X}'\mathbf{X}$ is invertible, and stop if it is not.

   (b) To check your function, install the package `VR`, use `library(MASS)` to make the data in the `MASS` package available, and load the data set `hills`. Use your linear model function you wrote to fit `time` as a function of `dist` and `climb`. Compare the results to the results from the `lm()` function.

3. An easy way to perform power calculations is by simulation. Suppose you are testing a drug that reduces blood pressure. You have 50 people in each of treatment and control groups, and expect the systolic blood pressure to have a mean of 150mmHg and standard devation of 15mmHg in the control group, and to be 10mmHg lower in the treatment group.

   (a) Assuming the distributions to be approximately Normal, simulate one set of data and perform a t-test using the `t.test` function.

   (b) Using the `names` function, look at the components of the object returned by `t.test`. The p-value is `t.test(x,y)$p.value`.

   (c) Write a loop to generate data and perform a t-test 10,000 times, storing the values in a vector. What is the power of the study? Compare the results with those given by `power.t.test`.

   (d) Suppose in the treated group the standard deviation were increased to 20mmHg. The `power.t.test` function can't handle this, so write your own function to compute the power.

4. Create a 1000×5000 matrix of random numbers. Then determine how long it takes to 'demean' each column (substracting the column mean from each element in the respective column) using:

   (a) a `for` loop,

   (b) `apply` twice,

   (c) `sweep` and `apply`,

   (d) `sweep` and `rowsum`.

5. `data(airquality)` contains measurements of ozone concentration in New York from May to September 1983, together with other relevant variables.

   (a) Plot the ozone concentration over time. Look at how different plotting types (`"l"`, `"h"`) affect the appearance. The EPA standard for ozone until recently was 120ppb, 140ppb was moderate nonattainment and 160ppb serious nonattainment. Indicate these with horizontal lines in appropriate colors. Use the `text()` function to annotate the severe ozone days with their dates.

   (b) Ozone is produced by chemical reactions in the air that require sunlight. Draw a scatterplot of ozone and solar radiation.

   (c) Perhaps wind or temperature are responsible for the shape of the plot. Use the `coplot` function to draw scatterplots of ozone and solar radiation for different levels of wind speed, of temperature, and of both at once. Do the same thing with the trellis command `xyplot`.