STAT 604: Mu-Fen Hsieh
Intro. to Statistical Computing Sep. 14, 2006

HW 4

This homework consists of writing a function to perform a simulation study regarding simple linear regression.

Recall that the simple linear regression model (in LaTeX's math notation) is:

$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ for $i = 1, \ldots, n$,

where the error terms $\epsilon_1, \ldots, \epsilon_n$ are mean zero, independent, and identically distributed random variables.

Let $\hat{\beta}_1$ denote the parameter estimate of β_1. The bias $\hat{\beta}_1$ in estimating β_1 is defined as the expected value of $\hat{\beta}_1$ minus β_1. A $100(1 - \alpha 1)\%$ confidence interval for the slope β_1 (based on normally distributed errors) is obtained as $\hat{\beta}_1 + / - t_{\alpha 1/2, df} \hat{\sigma}_{\hat{\beta}_1}$, where df is the residual (a.k.a., error) degrees of freedom, $\hat{\sigma}_{\hat{\beta}_1}$ is the standard error of the slope estimate, and $t_{\alpha 1/2, df}$ is the $\alpha 1/2$ upper quantile of the t-distribution with df degrees of freedom. The coverage of a confidence interval estimator is the long-run relative frequency that the interval contains the true value.

The regression coefficients β_0 and β_1 can be estimated using the method of least-squares. In R, if `x` and `y` are numeric vectors of the same length, the least-squares parameter estimates, standard errors, degrees of freedom, etc. are obtained using:

```
fm <- lm( y ~ x )
fm
summary(fm)
```

While theory provides results on the bias and coverage in simple linear regression models, the goal for this assignment to empirically investigate:

- The bias in estimating the slope, and
- The coverage of the $100(1 - \alpha 1)\%$ confidence interval estimator for the slope based on the usual normality assumption.

Make a text file named exactly "slope.R" whose content is the following:

```
## You should not modify the function definition.
## Also, you should not add any code before or after this function.
slope.simulation <- function(regressor,intercept,slope,
                      error.distribution=c("normal","chisq")[1],
                      nreps,alpha1,alpha2) {
  ## Your code goes here
}
```

For your convenience, you can download this skeleton file `"slope.R"`. In the body of the function, replace the comment with your code.

The arguments to the function are:

- `regressor`: Independent variable passed as a numeric vector of arbitrary length.
- `intercept`: True intercept passed as a numeric vector of length one.
- `slope`: True slope passed as a numeric vector of length one.
- `error.distribution`: Distribution of the error term passed as a character vector of length one, equaling either:
  - `"normal"`: Error term is standard normally distributed.
  - `"chisq"`: Error term has a chi-squared distribution with 0.5 degrees of freedom and shifted to the left by 0.5 (so that the mean of the error term is zero).
- `nreps`: Number of replications.
- `alpha1`: Equals 1.0 minus the confidence coefficient for the confidence interval estimator of the slope. For example, for 95% confidence intervals on the slope, `alpha1 = 0.05`.
- `alpha2`: Equals 1.0 minus the confidence coefficient for the confidence intervals of the bias and coverage probabilities. For example, for a 90% confidence interval on the bias, `alpha2 = 0.10`.

For each iteration of the nreps iterations of the simulation, randomly generate the response (i.e., dependent) vector using the supplied intercept, slope, regressor, and error term distribution. Fit the simple linear regression model and compute a $100(1 - \alpha 1)\%$ confidence interval on the slope parameter. Record whether it contains the true slope. Also, record the difference between the slope estimate and its true value.

The proportion of times that the confidence interval contains the true slope is a point estimate of the its coverage. Theory says that the coverage should be $1 - \alpha 1$ when the error distribution is normal. In addition to providing a point estimate of the coverage, you will provide a $100(1-\alpha 2)\%$ confidence interval on the coverage. (Use the normal approximation to the binomial, which is justified by the Central Limit Theorem since `nreps` is large.)

The average difference between the slope estimate and its true value is a point estimate of the bias. In addition to providing a point estimate of the bias, you will provide a $100(1 - \alpha 2)\%$ confidence interval on the bias. (Again, the Central Limit Theorem is applicable.)

Your function should return a numeric vector of length six whose elements, in order, are:

1. A point estimate of the bias in estimating the slope
2. The lower bound of a $100(1 - \alpha 2)\%$ confidence interval for the bias
3. The upper bound of a $100(1 - \alpha 2)\%$ confidence interval for the bias
4. A point estimate of the coverage of the $100(1 - \alpha 1)\%$ confidence interval estimator of the slope
5. The lower bound of a $100(1 - \alpha 2)\%$ confidence interval for the coverage
6. The upper bound of a $100(1 - \alpha 2)\%$ confidence interval for the coverage

In writing your code, you will likely find the following R functions helpful: `lm, qt, summary, str`. If `fm` is an object of class `lm, str(fm)` and `str(summary(fm))` will reveal how you can pick out parameter estimates, degrees of freedom, standard errors, etc.

To help you test your code and see the effects of various parameter values, you may want to download `"verify.R"`. Once you have your code running, see if the results make sense. Experiment with different parameter settings. Think about the following questions: Is the estimator of the slope biased under either error terms? Does the confidence interval estimator have the right coverage under the normally distributed error terms? How about the chi-squared distributed error terms? If the coverage is off, under what situation is it noticeable. Further, when does this coverage problem go away? What phenomenon makes it go away?

Submit your `"slope.R"` file to hw04@dahlgrapevine.org and bring a hard copy to class.