We now review k-fold cross-validation.

(a) Explain how k-fold cross-validation is implemented.

The data are split into k groups of about equal size. The first group is set aside to be used as a validation set. The model is trained on the remaining sets, and an MSE is computed using the validation set. This process is repeated using the next set as a validation set, producing a second MSE. The process is performed on each of the k groups the data were split into, producing k MSEs, one for each validation set. Given a good value of k, this mean squared error is a good approximation for the minimum point along the flexibility axis that allows us to judge the appropriate flexibility to model some data, i.e., to estimate the right bias-variance tradeoff for a particular model.

(b) What are the advantages and disadvantages of k-fold cross-validation relative to:

i. The validation set approach?

The validation set approach is basically k-fold with k=2. This gives a biased measure of the test error rate. However, it's very quick to compute. Of course, the MSE here will have no variance because we are only taking a single MSE from a single test set: so this gives large bias of the MSE but no variance.

ii. LOOCV?

LOOCV is on the other end of the spectrum from this. It gives a very unbiased MSE for each validation set because each validation set is a single datum and is being tested against a nearly-full training set, compared to the overall data set. However, the MSEs will be highly-correlated because the training sets are almost completely the same, and this guarantees a high variance.

K-fold cross-validation occupies the space in-between, where it has some bias and some variance of the MSE. Empirically, we know that k=5 and k=10 are two values that often work well to provide good estimates of these MSEs, because they provide balanced bias and variance.

6. We continue to consider the use of a logistic regression model to predict the probability of default using income and balance on the Default data set. In particular, we will now compute estimates for the standard errors of the income and balance logistic regression co- efficients in two different ways: (1) using the bootstrap, and (2) using the standard formula for computing the standard errors in the glm() function. Do not forget to set a random seed before beginning your analysis.

(a) Using the summary() and glm() functions, determine the estimated standard errors for the coefficients associated with income and balance in a multiple logistic regression model that uses both predictors.

10,000 entries in the table, so testing 2 ways to separate the data,

```
> train.default1 = Default[1:6001,]
> train.default2 = Default[1:5001,]
> test.default1 = Default[6002:10000,]
> test.default2 = Default[5002:10000,]

> fit.glm.default1 = glm(default ~ income + balance,train.default1, family="binomial")
> fit.glm.default2 = glm(default ~ income + balance,train.default2, family="binomial")
```

*** from summary(fit.glm.default1), we get std. error

```
    Coefficients:
                 Estimate Std. Error
(Intercept) -1.156e+01  5.622e-01
income       2.237e-05  6.491e-06
balance      5.649e-03  2.916e-04
```

*** from summary(fit.glm.default2), we get std. error

```
    Coefficients:
                 Estimate Std. Error
(Intercept) -1.198e+01  6.279e-01
income       2.885e-05  7.060e-06
balance      5.822e-03  3.232e-04
```

(b) Write a function, boot.fn() , that takes as input the\ Default data set as well as an index of the observations, and that outputs the coefficient estimates for income and balance in the multiple logistic regression model.

```
boot.fn = function(Default,index){
    model = glm(default ~ income + balance,Default,family="binomial",subset=index)
    model$coefficients[2:3]
}
```

(c) Use the boot() function together with your boot.fn() function to estimate the standard errors of the logistic regression coefficients for income and balance .

```
> set.seed(56)
> boot(Default,boot.fn,1000)

Bootstrap Statistics :
        original       bias     std. error
t1* 2.080898e-05 1.436086e-07 4.679106e-06
```

```
t2* 5.647103e-03 1.876364e-05 2.320547e-04
```

***

The std. error in t1 above is the std. error for the income coefficient, and under t2, the std. error for the balance coefficient.

(d) Comment on the estimated standard errors obtained using the glm() function and using your bootstrap function.

The bootstrap method uses random sampling to approximate sampling a real response. This gives a stronger σ^2 (variance) estimate than the formulaic approach used to calculate the std err for a linear regression model. The std err estimates from the bootstrap method are therefore more reliable than those computed earlier in this exercise.

The std err from bootstrap for the income coefficient is about 66% of that computed from the linear regression model directly. This is 72% for the coefficient of balance. These smaller values should be considered reliable, that is, we can believe the smaller errors compared to the standard meothods, because of the reason I explained above.