Exercise 2

(a) 1 - 1/n

(b) 1 - 1/n

(c) In bootstrap, we sample with replacement so each observation in the bootstrap sample has the same 1/n (independent) chance of equaling the jth observation. Applying the product rule for a total of n observations gives us (1 - 1/n)^n.

(d) Pr(in) = 1 - Pr(out) = 1 - (1 - 1/5)^5 = 1 - (4/5)^5 = 67.2%

(e) Pr(in) = 1 - Pr(out) = 1 - (1 - 1/100)^{10} = 1 - (99/100)^{100} = 63.4%

(f) 1 - (1 - 1/10000)^{10000} = 63.2%

(g)

```
pr = function(n) return(1 - (1 - 1/n)^n)
x = 1:100000
plot(x, pr(x))
```

The plot quickly reaches an asymptote of about 63.2%.

(h)

```
set.seed(1)
store = rep(NA, 1e4)
for (i in 1:1e4) {
  store[i] = sum(sample(1:100, rep=T) == 4) > 0
}
mean(store)
```

The numerical results show an approximate mean probability of 64.1%, close to our theoretically derived result.

Exercise 3

(a) k-fold cross-validation is implemented by taking the set of n observations and randomly splitting into k non-overlapping groups. Each of these groups acts as a validation set and the remainder as a training set. The test error is estimated by averaging the k resulting MSE estimates.

(b)

i. The validation set approach is conceptually simple and easily implemented as you are simply partitioning the existing training data into two sets. However, there are two drawbacks: (1) the estimate of the test error rate can be highly variable depending on which observations are included in the training and validation sets; (2) the validation set error rate may tend to overestimate the test error rate for the model fit on the entire data set.

ii. LOOCV is a special case of k-fold cross-validation with k = n. Thus, LOOCV is the most computationally intense method since the model must be fit n times. Also, LOOCV has higher variance, but lower bias, than k-fold CV.

Exercise 5

(a)

```
library(ISLR)
summary(Default)
attach(Default)

set.seed(1)
glm.fit = glm(default~income+balance, data=Default, family=binomial)
```

(b)

```
FiveB = function() {
# i.
train = sample(dim(Default)[1], dim(Default)[1]/2)
# ii.
glm.fit = glm(default~income+balance, data=Default, family=binomial,
              subset=train)
# iii.
glm.pred = rep("No", dim(Default)[1]/2)
glm.probs = predict(glm.fit, Default[-train,], type="response")
glm.pred[glm.probs>.5] = "Yes"
# iv.
return(mean(glm.pred != Default[-train,]$default))
}
FiveB()
```

2.86% test error rate from validation set approach.

(c)

```
FiveB()
FiveB()
FiveB()
```

It seems to average around 2.6% test error rate.

(d)

```
train = sample(dim(Default)[1], dim(Default)[1]/2)
glm.fit = glm(default~income+balance+student, data=Default, family=binomial,
              subset=train)
glm.pred = rep("No", dim(Default)[1]/2)
glm.probs = predict(glm.fit, Default[-train,], type="response")
glm.pred[glm.probs>.5] = "Yes"
mean(glm.pred != Default[-train,]$default)
```

2.64% test error rate, with student dummy variable. Using the validation set approach, it doesn't appear adding the student dummy variable leads to a reduction in the test error rate.

Exercise 8

(a)

```
set.seed(1)
y = rnorm(100)
x = rnorm(100)
y = x - 2*x^2 + rnorm(100)
```

n = 100, p = 2.

Y = X - 2 X^2 + ϵ

(b)

```
plot(x, y)
```

Quadratic plot. X from about -2 to 2. Y from about -8 to 2.

(c)

```
library(boot)
Data = data.frame(x,y)
set.seed(1)
# i.
glm.fit = glm(y~x)
cv.glm(Data, glm.fit)$delta
# ii.
glm.fit = glm(y~poly(x,2))
cv.glm(Data, glm.fit)$delta
# iii.
glm.fit = glm(y~poly(x,3))
cv.glm(Data, glm.fit)$delta
# iv.
glm.fit = glm(y~poly(x,4))
cv.glm(Data, glm.fit)$delta
```

(d)

```
set.seed(10)
# i.
glm.fit = glm(y~x)
cv.glm(Data, glm.fit)$delta
# ii.
glm.fit = glm(y~poly(x,2))
cv.glm(Data, glm.fit)$delta
# iii.
glm.fit = glm(y~poly(x,3))
cv.glm(Data, glm.fit)$delta
# iv.
glm.fit = glm(y~poly(x,4))
cv.glm(Data, glm.fit)$delta
```

Exact same, because LOOCV will be the same since it evaluates n folds of a single observation.

(e) The quadratic polynomial had the lowest LOOCV test error rate. This was expected because it matches the true form of Y.

(f)

```
summary(glm.fit)
```

p-values show statistical significance of linear and quadratic terms, which agrees with the CV results.