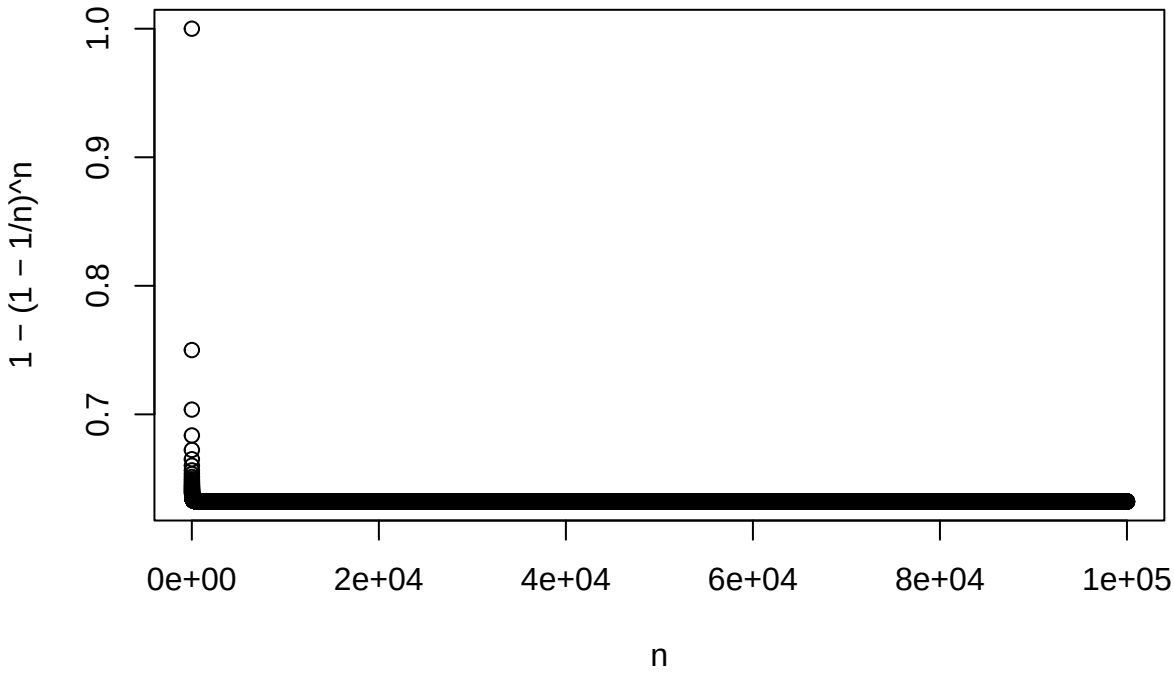
# Exercise 5.4

Section 23 -
Group 6 Project Groups (Bosan Hsu, Fan Liu, Jimeng Yin, Michael Liu, Richard Wang, Zhuoqian Zhang

Problem 2: We will now derive the probability that a given observation is part of a bootstrap sample. Suppose that we obtain a bootstrap sample from a set of n observations.

(g) Create a plot that displays, for each integer value of n from 1 to 100, 000, the probability that the jth observation is in the bootstrap sample. Comment on what you observe.

```
n = 1:100000
plot(n,1-(1-1/n)^n)
```




When n gets bigger, the probability that a observation is part of the bootstrap sample tends towards a constant value near 0.632.

(h) We will now investigate numerically the probability that a bootstrap sample of size n=100 contains the jth observation. Here j=4. We repeatedly create bootstrap samples, and each time we record whether or not the fourth observation is contained in the bootstrap sample.

```
store=rep(NA, 10000)
for(i in 1:10000) {store[i]=sum(sample (1:100, rep=TRUE)==4) >0
}
mean(store)
```

```
## [1] 0.6267
```

Comment on the results obtained. The value is close to 0.632 as we get in the individual text book quiz.

Exercise 5.4: Problem 6 Problem 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 coefficients in two different ways:

```
library(ISLR)
attach(Default)
set.seed(1)
Default = na.omit(Default)
```

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.

```
glm.fit = glm(default~income+balance, family = "binomial")
summary(glm.fit)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = "binomial")
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
## income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
## balance      5.647e-03  2.274e-04  24.836  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1579.0  on 9997  degrees of freedom
## AIC: 1585
##
## Number of Fisher Scoring iterations: 8
```

(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.

```r
boot.fn = function(df,index){
  glm.fit = glm(default~income+balance, data = df, subset=index, family = "binomial")
  return(coef(glm.fit))
}
```

(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.

```r
library(boot)
set.seed(1)
boot(Default,boot.fn,1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Default, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##          original        bias     std. error
## t1* -1.154047e+01 -3.945460e-02 4.344722e-01
## t2*  2.080898e-05  1.680317e-07 4.866284e-06
## t3*  5.647103e-03  1.855765e-05 2.298949e-04
```

(d) Comment on the estimated standard errors obtained using the glm() function and using your bootstrap function. There are no obviously difference from the calculated standard errors.