CS-E-106: Data Modeling

Assignment 10

Instructor: Hakan Gogtas
Submitted by: *Saurabh Kulkarni*

Due Date: *12/12/2019*

Question 1 Refer to the Prostate Cancer data set in Appendix C.5 and Homework 9. Select a random sample of 65 observations to use as the model-building data set (use set.seed(1023)). Use the remaining observations for the test data. (10 pts)

(a) Develop a neural network model for predicting PSA. Justify your choice of number of hidden nodes and interpret your model. Test the model performance on the test data.

```
prostate_data = read.csv("Prostate Cancer.csv")
summary(prostate_data)
```

```
##    PSA.level       Cancer.volume         Weight            Age
##  Min.   :  0.651   Min.   : 0.2592   Min.   : 10.70   Min.   :41.00
##  1st Qu.:  5.641   1st Qu.: 1.6653   1st Qu.: 29.37   1st Qu.:60.00
##  Median : 13.330   Median : 4.2631   Median : 37.34   Median :65.00
##  Mean   : 23.730   Mean   : 6.9987   Mean   : 45.49   Mean   :63.87
##  3rd Qu.: 21.328   3rd Qu.: 8.4149   3rd Qu.: 48.42   3rd Qu.:68.00
##  Max.   :265.072   Max.   :45.6042   Max.   :450.34   Max.   :79.00
##  Benign.prostatic.hyperplasia Seminal.vesicle.invasion
##  Min.   : 0.000               Min.   :0.0000
##  1st Qu.: 0.000               1st Qu.:0.0000
##  Median : 1.350               Median :0.0000
##  Mean   : 2.535               Mean   :0.2165
##  3rd Qu.: 4.759               3rd Qu.:0.0000
##  Max.   :10.278               Max.   :1.0000
##  Capsular.penetration Gleason.score
##  Min.   : 0.0000      Min.   :6.000
##  1st Qu.: 0.0000      1st Qu.:6.000
##  Median : 0.4493      Median :7.000
##  Mean   : 2.2454      Mean   :6.876
##  3rd Qu.: 3.2544      3rd Qu.:7.000
##  Max.   :18.1741      Max.   :8.000
```

```
max = apply(prostate_data, 2, max)
min = apply(prostate_data, 2, min)
scaled_df = as.data.frame(scale(prostate_data, center=min, scale=max-min))
```

```
set.seed(1023)
train_ind = sample(1:nrow(scaled_df), 65)
test_ind = setdiff(1:nrow(scaled_df), train_ind)
train_df = scaled_df[train_ind,]
test_df = scaled_df[test_ind,]
```

```
NN = neuralnet(PSA.level ~ ., data=train_df, hidden=7 , linear.output= T, stepmax=1e6)
plot(NN)
```

```
maxY= max(prostate_data$PSA.level)
minY = min(prostate_data$PSA.level)
```

```
yHat_NN_te = predict(NN, test_df)*(maxY-minY)+minY
yAct_te = test_df$PSA.level*(maxY-minY)+minY
SSE_NN = sum((yHat_NN_te-yAct_te)^2)
SSE_NN
```

```
## [1] 70171.34
```

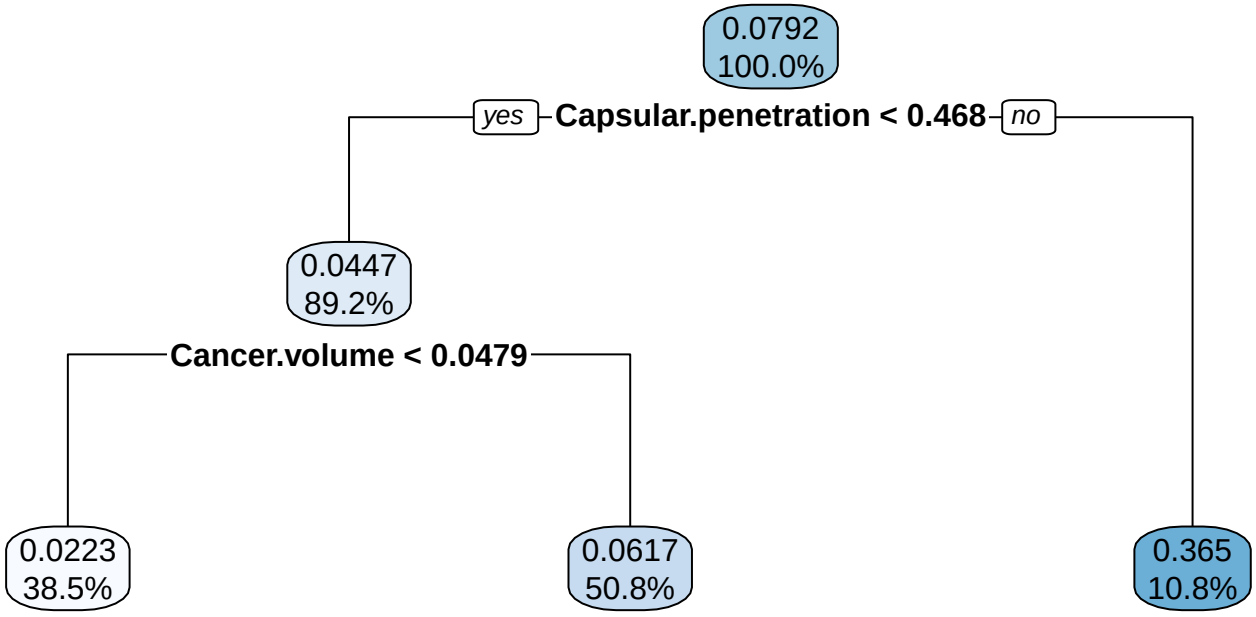
Interpretation:

We select 7 hidden nodes, one for each variable. SSE on test data = 70171.34

(b) Compare the performance of your neuron network model with regression tree model obtained in HW9. Which model is more easily interpreted and why? (5pts)

Tree Model - HW 9

```
library(rpart.plot)
tree_prostate = rpart(PSA.level~., data=train_df)
rpart.plot(tree_prostate, digits = 3)
```




```
yHat_tree_te = predict(tree_prostate, test_df)*(maxY-minY)+minY
SSE_tree = sum((yHat_tree_te-yAct_te)^2)
SSE_tree
```

```
## [1] 69339.32
```

Interpretation:

We see that the tree model performs slightly better than the neural network with the selected architecture above.

The tree model is more interpretable, since it gives a clear decision flow for branching into each of the regions.

(c) Compare the performance of your neural network model with that of the best regression model obtained in homework 8. Which model is more easily interpreted and why?

Best Model - HW 8

```
lm_prostate_best = lm(PSA.level~Cancer.volume+Capsular.penetration, data=train_df)
summary(lm_prostate_best)
```

```
##
## Call:
## lm(formula = PSA.level ~ Cancer.volume + Capsular.penetration,
##     data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26203 -0.02961  0.00500  0.02080  0.51969
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.007e-05  1.651e-02   0.005  0.99615
## Cancer.volume        3.122e-01  1.031e-01   3.028  0.00359 **
## Capsular.penetration 2.608e-01  8.371e-02   3.115  0.00279 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.105 on 62 degrees of freedom
## Multiple R-squared:  0.5266, Adjusted R-squared:  0.5113
## F-statistic: 34.48 on 2 and 62 DF,  p-value: 8.572e-11
```

```
yHat_lm_te = predict(lm_prostate_best, test_df)*(maxY-minY)+minY
SSE_best_lm = sum((yHat_lm_te-yAct_te)^2)
SSE_best_lm
```

```
## [1] 47256.33
```

Interpretation:

The best subset model selected in homework 8 is the best amongst the three model above based on the SSE on test data.

Again, the neural network model is comparitively less interpetable as it does not give us any inferencing measures like p-values for different variables or any other statistical inferencing.

Question 2 Refer to the Disease outbreak data set in Appendix C.10. Savings account status is the response variable and age, socioeconomic status, and city sector are the predictor variables.

(a) Fit logistic regression model to predict the saving account status on the predictor variables in first-order terms and interaction terms for. all pairs of predictor variables. State the fitted response function.

```
disease_data = read.csv("Disease Outbreak.csv")
disease_data$Socioeconomic.status = as.factor(disease_data$Socioeconomic.status)
disease_data$Sector = as.factor(disease_data$Sector)
disease_data$Disease.status = as.factor(disease_data$Disease.status)
summary(disease_data)
```

```
##       Age        Socioeconomic.status Sector  Disease.status
##  Min.   : 1.00   1:77                 1:117   0:139
##  1st Qu.:10.75   2:49                 2: 79   1: 57
##  Median :21.00   3:70
##  Mean   :25.18
##  3rd Qu.:35.00
##  Max.   :85.00
##  Savings.account.status
```

```
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.5459
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```
Y = disease_data$Savings.account.status
X1 = disease_data$Age
X2 = disease_data$Socioeconomic.status
X3 = disease_data$Sector
X4 = disease_data$Disease.status
X2_2 = ifelse(X2==2, 1, 0)
X2_3 = ifelse(X2==3, 1, 0)
X3_2 = ifelse(X3==2, 1, 0)
X4_1 = ifelse(X4==1, 1, 0)

df = as.data.frame(cbind(Y,X1,X2_2,X2_3,X3_2,X4_1))

df$X1.X2_2 = X1*X2_2
df$X1.X2_3 = X1*X2_3
df$X1.X3_2 = X1*X3_2
df$X1.X4_1 = X1*X4_1
df$X2_3.X3_2 = X2_3*X3_2
df$X2_3.X4_1 = X2_3*X4_1
df$X2_2.X3_2 = X2_2*X3_2
df$X2_2.X4_1 = X2_2*X4_1
df$X3_2.X4_1 = X3_2*X4_1

summary(df)
```

```
##        Y                X1              X2_2            X2_3
##  Min.   :0.0000   Min.   : 1.00   Min.   :0.00   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:10.75   1st Qu.:0.00   1st Qu.:0.0000
##  Median :1.0000   Median :21.00   Median :0.00   Median :0.0000
##  Mean   :0.5459   Mean   :25.18   Mean   :0.25   Mean   :0.3571
##  3rd Qu.:1.0000   3rd Qu.:35.00   3rd Qu.:0.25   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :85.00   Max.   :1.00   Max.   :1.0000
##       X3_2             X4_1           X1.X2_2          X1.X2_3
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 0.00
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 0.00
##  Median :0.0000   Median :0.0000   Median : 0.000   Median : 0.00
##  Mean   :0.4031   Mean   :0.2908   Mean   : 5.755   Mean   : 8.51
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 0.500   3rd Qu.:10.25
##  Max.   :1.0000   Max.   :1.0000   Max.   :68.000   Max.   :85.00
##     X1.X3_2          X1.X4_1         X2_3.X3_2         X2_3.X4_1
##  Min.   : 0.00   Min.   : 0.00   Min.   :0.00000   Min.   :0.00000
##  1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.:0.00000   1st Qu.:0.00000
##  Median : 0.00   Median : 0.00   Median :0.00000   Median :0.00000
##  Mean   :11.16   Mean   : 9.48   Mean   :0.08673   Mean   :0.09694
##  3rd Qu.:15.00   3rd Qu.:13.25   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :79.00   Max.   :74.00   Max.   :1.00000   Max.   :1.00000
##    X2_2.X3_2        X2_2.X4_1         X3_2.X4_1
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
```

```
##  Median :0.0000   Median :0.00000   Median :0.0000
##  Mean   :0.1173   Mean   :0.07143   Mean   :0.1786
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
```

```
logreg_full = glm(Y~., data=df, family=binomial)
summary(logreg_full)
```

```
##
## Call:
## glm(formula = Y ~ ., family = binomial, data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3855  -0.8886   0.4118   0.7943   2.0273
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.155908   0.587157   0.266  0.79060
## X1           0.035838   0.021966   1.632  0.10277
## X2_2        -1.306280   0.817101  -1.599  0.10989
## X2_3        -2.151271   0.758426  -2.836  0.00456 **
## X3_2         0.916937   0.781780   1.173  0.24084
## X4_1        -0.946814   1.062247  -0.891  0.37275
## X1.X2_2      0.008166   0.029619   0.276  0.78278
## X1.X2_3      0.002890   0.024113   0.120  0.90461
## X1.X3_2     -0.021077   0.022438  -0.939  0.34755
## X1.X4_1      0.021247   0.025814   0.823  0.41045
## X2_3.X3_2    0.388653   0.867955   0.448  0.65431
## X2_3.X4_1   -0.137603   0.958732  -0.144  0.88587
## X2_2.X3_2   -0.131848   0.880545  -0.150  0.88097
## X2_2.X4_1   -0.111640   1.044638  -0.107  0.91489
## X3_2.X4_1    0.930980   0.835249   1.115  0.26502
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 212.84  on 181  degrees of freedom
## AIC: 242.84
##
## Number of Fisher Scoring iterations: 5
```

Estimate Response function: $\log_e(\frac{Y}{1-Y}) = 0.155908 + 0.035838 * X1 - 1.306280 * X2_2 - 2.151271 * X2_3 + 0.916937 * X3_2 - 0.946814 * X4_1 + 0.008166 * X1.X2_2 + 0.002890 * X1.X2_3 - 0.021077 * X1.X3_2 + 0.021247 * X1.X4_1 + 0.388653 * X2_3.X3_2 - 0.137603 * X2_3.X4_1 - 0.131848 * X2_2.X3_2 - 0.111640 * X2_2.X4_1 + 0.930980 * X3_2.X4_1$

(b) Use the likelihood ratio test to determine whether all interaction terms can be dropped from the regression model; use = .01. State the alternatives, full and reduced models, decision rule, and conclusion. What is the approximate P-value of the test?

$H_0 : E(Y_i j) = [1 + \exp(-X_j' \beta)^{-1}]$ $H_1 : E(Y_i j) \neq [1 + \exp(-X_j' \beta)^{-1}]$

```r
logreg_red = glm(Y~X1+X2_2+X2_3+X3_2+X4_1, data=df, family=binomial)
summary(logreg_red)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X2_2 + X2_3 + X3_2 + X4_1, family = binomial,
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2845  -0.8649   0.3885   0.8206   1.9874
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.05550    0.38189   0.145 0.884449
## X1           0.03563    0.01003   3.552 0.000382 ***
## X2_2        -1.17468    0.41776  -2.812 0.004926 **
## X2_3        -1.95235    0.40287  -4.846 1.26e-06 ***
## X3_2         0.79651    0.36120   2.205 0.027441 *
## X4_1        -0.02908    0.39303  -0.074 0.941026
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 270.06  on 195  degrees of freedom
## Residual deviance: 215.36  on 190  degrees of freedom
## AIC: 227.36
##
## Number of Fisher Scoring iterations: 4
```

```r
anova(logreg_red, logreg_full, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X2_2 + X2_3 + X3_2 + X4_1
## Model 2: Y ~ X1 + X2_2 + X2_3 + X3_2 + X4_1 + X1.X2_2 + X1.X2_3 + X1.X3_2 +
##     X1.X4_1 + X2_3.X3_2 + X2_3.X4_1 + X2_2.X3_2 + X2_2.X4_1 +
##     X3_2.X4_1
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       190     215.36
## 2       181     212.84  9   2.5213   0.9803
```

Interpretation:

The p-value is 0.9803.

With such high p-value, we can drop all the interaction terms at $\alpha = 0.01$ as it suggests there is no significant difference in the deviance measure of the two models.

(c) Conduct the Hosmer-Lemeshow goodness of fit test for the appropriateness of the logistic regression function by forming five groups of approximately 20 cases each; use = .05.

$H_0 : E(Y_i j) = [1 + \exp(-X_j' \beta)^{-1}]$ $H_1 : E(Y_i j) \neq [1 + \exp(-X_j' \beta)^{-1}]$

```r
?hoslem.test
hoslem.test(logreg_full$y,fitted(logreg_full),g=5)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  logreg_full$y, fitted(logreg_full)
## X-squared = 0.85353, df = 3, p-value = 0.8366
```

```
qchisq(1-0.05,3)
```

```
## [1] 7.814728
```

Decision Rule:

$DEV(X_0, X_1, ...X_{p-1}) \leq \chi^2(1-\alpha; c-p)$ conclude H_0

$DEV(X_0, X_1, ...X_{p-1}) > \chi^2(1-\alpha; c-p)$ conclude H_1

Result:

The p-value is 0.8366.

Also, 0.85353<=7.814728 i.e. $DEV(X_0, X_1, ...X_{p-1}) \leq \chi^2(1-\alpha; c-p)$. Thus, we conclude H_0, the fit is good.

Question 3 Refer to the Geriatric study. A researcher in geriatrics designed a prospective study to investigate the effects of two interventions on the frequency of falls. One hundred subjects were randomly assigned to one of the two interventions: education only (X1 = 0) and education plus aerobic exercise training (X1 = 1). Subjects were at least 65 years of age and in reasonably good health. Three variables considered to be important as control variables were gender (X2:0=female;1=male), a balance index (X3). and a strength index (X4). The higher balance index, the more stable is the subject and the higher the strength index, the stronger is the subject. Each subject kept a diary recording the number of falls (Y) during the six months of the study.

(a) Fit the regression model. State the estimated regression coefficients, their estimated standard deviations. and the estimated response function.

```
geriatric_data = read.csv("Geriatric Study.csv")
geriatric_data$X1 = as.factor(geriatric_data$X1)
geriatric_data$X2 = as.factor(geriatric_data$X2)
summary(geriatric_data)
```

```
##        Y         X1     X2           X3              X4
##  Min.   : 0.00   0:50   0:47   Min.   :13.00   Min.   :18.00
##  1st Qu.: 1.00   1:50   1:53   1st Qu.:39.00   1st Qu.:52.00
##  Median : 3.00                 Median :51.50   Median :60.00
##  Mean   : 3.04                 Mean   :52.83   Mean   :60.78
##  3rd Qu.: 4.00                 3rd Qu.:66.25   3rd Qu.:70.25
##  Max.   :11.00                 Max.   :98.00   Max.   :90.00
```

```
pmod_geriatric = glm(Y~X1+X2+X3+X4, data=geriatric_data, family=poisson)
summary(pmod_geriatric)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X2 + X3 + X4, family = poisson, data = geriatric_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1854  -0.7819  -0.2564   0.5449   2.3626
##
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.489467   0.336869   1.453  0.14623
## X11         -1.069403   0.133154  -8.031 9.64e-16 ***
## X21         -0.046606   0.119970  -0.388  0.69766
## X3           0.009470   0.002953   3.207  0.00134 **
## X4           0.008566   0.004312   1.986  0.04698 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 199.19  on 99  degrees of freedom
## Residual deviance: 108.79  on 95  degrees of freedom
## AIC: 377.29
##
## Number of Fisher Scoring iterations: 5
```

```
confint(pmod_geriatric)
```

```
## Waiting for profiling to be done...
```

```
##                      2.5 %      97.5 %
## (Intercept) -0.1836076944  1.13605432
## X11         -1.3360219299 -0.81332114
## X21         -0.2823288477  0.18838553
## X3           0.0036833502  0.01526299
## X4           0.0001457923  0.01704817
```

Interpretation:

We can see the estimated coefficients and their estimated standard deviations in the summary print out of the model. We can also see the 95% confidence intervals on the coefficients.

Estimated Response Function:

$\log_e(\frac{Y}{1-Y}) = 0.489467 - 1.069403 * X1_1 - 0.046606 * X2_1 + 0.009470 * X3 + 0.008566 * X4$

(b) Assuming that the fitted model is appropriate, use the likelihood ratio test to determine whether gender (X2) can be dropped from the model: State the full and reduced models. decision rule. and conclusion. What is the P-value of the test

$H_0 : \beta_2 = 0$ $H_1 : \beta_2 \neq 0$

```
pmod_geriatric_red = glm(Y~X1+X3+X4, data=geriatric_data, family=poisson)
summary(pmod_geriatric_red)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X3 + X4, family = poisson, data = geriatric_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2152  -0.7512  -0.2594   0.5830   2.2893
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.443890   0.317289   1.399  0.16181
## X11         -1.077770   0.131415  -8.201 2.38e-16 ***
## X3           0.009471   0.002957   3.203  0.00136 **
```

```
## X4           0.008979   0.004190   2.143  0.03209 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 199.19  on 99  degrees of freedom
## Residual deviance: 108.94  on 96  degrees of freedom
## AIC: 375.44
##
## Number of Fisher Scoring iterations: 5
```

```
anova(pmod_geriatric_red, pmod_geriatric, test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Y ~ X1 + X3 + X4
## Model 2: Y ~ X1 + X2 + X3 + X4
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        96     108.94
## 2        95     108.79  1    0.151   0.6976
```

Interpretation:

The p-value is 0.6976.

With such high p-value, we can drop X2 as it suggests there is no significant difference in the deviance measure of the two models.

(c) Predict the number of falls for X1=1, X2=0, X3=45, X4=70.

```
Xh = data.frame(X1=as.factor(c(1)), X2=as.factor(c(0)), X3=c(45), X4=c(70))
predict(pmod_geriatric, Xh, type="response", se.fit=TRUE)
```

```
## $fit
##        1
## 1.561773
##
## $se.fit
##         1
## 0.2146503
##
## $residual.scale
## [1] 1
```