logistic_regression.R

Fiona

Fri Oct 13 17:47:06 2017

```
#Logistic Regression Example

#Load state data which is available in R base package. The dataset includes data on 50 US
#states such as Population, Illiteracy, Murder Rate, Income
data <- state.x77

#View the first few records of the data
head(data)
```

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20
## Alaska            365   6315        1.5    69.31   11.3    66.7   152
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65
## California      21198   5114        1.1    71.71   10.3    62.6    20
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166
##              Area
## Alabama     50708
## Alaska     566432
## Arizona    113417
## Arkansas    51945
## California 156361
## Colorado   103766
```

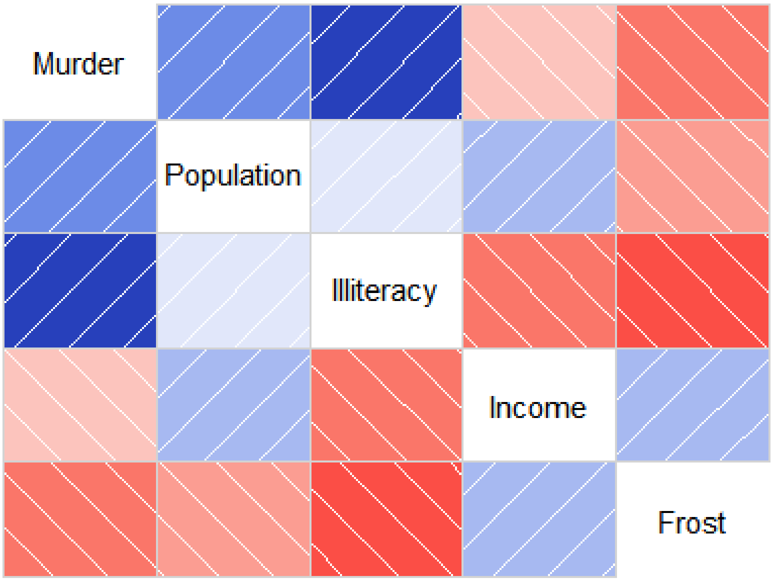
```
#As an example we will use Population, Illiteracy, Income and Frost to predict the murder
#variable. We first view how the variables correlate to each other
cor(data[,c("Murder", "Population","Illiteracy", "Income", "Frost")])
```

```
##                Murder Population Illiteracy     Income      Frost
## Murder      1.0000000  0.3436428  0.7029752 -0.2300776 -0.5388834
## Population  0.3436428  1.0000000  0.1076224  0.2082276 -0.3321525
## Illiteracy  0.7029752  0.1076224  1.0000000 -0.4370752 -0.6719470
## Income     -0.2300776  0.2082276 -0.4370752  1.0000000  0.2262822
## Frost      -0.5388834 -0.3321525 -0.6719470  0.2262822  1.0000000
```

```
#Install corrgram package which is used to visualize correlations between variables
#install.packages("corrgram")
library(corrgram)
```

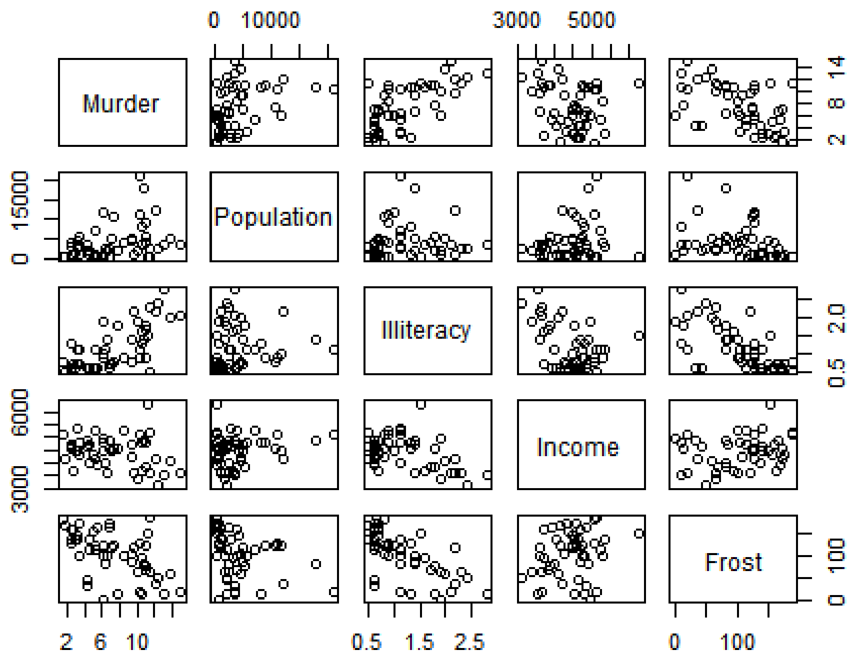
```
## Warning: package 'corrgram' was built under R version 3.3.3
```

```r
corrgram(data[,c("Murder", "Population","Illiteracy", "Income", "Frost")])
```




```r
#Use also pairs() to visualize correlations

pairs(data[,c("Murder", "Population","Illiteracy", "Income", "Frost")])
```




```
#Conduct a Multiple Linear Regression Using the following code
#the data set is stored as a matrix, we therefore use as.data.frame() to
convert it to a
#data frame because lm() only handles data frames

fit <- lm(Murder ~ Population + Illiteracy + Income + Frost,
data=as.data.frame(data))

summary(fit)

##
## Call:
## lm(formula = Murder ~ Population + Illiteracy + Income + Frost,
##     data = as.data.frame(data))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7960 -1.6495 -0.0811  1.4815  7.6210
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.235e+00  3.866e+00   0.319   0.7510
## Population  2.237e-04  9.052e-05   2.471   0.0173 *
## Illiteracy  4.143e+00  8.744e-01   4.738 2.19e-05 ***
## Income      6.442e-05  6.837e-04   0.094   0.9253
## Frost       5.813e-04  1.005e-02   0.058   0.9541
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.535 on 45 degrees of freedom
## Multiple R-squared:  0.567,  Adjusted R-squared:  0.5285
## F-statistic: 14.73 on 4 and 45 DF,  p-value: 9.133e-08

#From these results we see that income and frost are not significant

fit2 <- lm(Murder ~ Population + Illiteracy, data=as.data.frame(data))

summary(fit2)

##
## Call:
## lm(formula = Murder ~ Population + Illiteracy, data = as.data.frame(data))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7652 -1.6561 -0.0898  1.4570  7.6758
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.652e+00  8.101e-01   2.039  0.04713 *
## Population  2.242e-04  7.984e-05   2.808  0.00724 **
## Illiteracy  4.081e+00  5.848e-01   6.978 8.83e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.481 on 47 degrees of freedom
## Multiple R-squared:  0.5668, Adjusted R-squared:  0.5484
## F-statistic: 30.75 on 2 and 47 DF,  p-value: 2.893e-09

anova(fit,fit2)

## Analysis of Variance Table
##
## Model 1: Murder ~ Population + Illiteracy + Income + Frost
## Model 2: Murder ~ Population + Illiteracy
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     45 289.17
## 2     47 289.25 -2 -0.078505 0.0061 0.9939

predict(fit2,list(Population=10000,Illiteracy=1.1))

##        1
## 8.382221

#We can use stepwise regression to select significant predictor variables. We
start by
#creating a model using all the variables then remove the least significant
variables
```

```r
fit3 <- lm(Murder ~., data=as.data.frame(data))

#Use the step() function to conduct a stepwise regression.

fit4 <- step(fit3, direction="backward")
```

```
## Start:  AIC=63.01
## Murder ~ Population + Income + Illiteracy + `Life Exp` + `HS Grad` +
##     Frost + Area
##
##              Df Sum of Sq    RSS    AIC
## - Income      1     0.236 128.27 61.105
## - `HS Grad`   1     0.973 129.01 61.392
## <none>                    128.03 63.013
## - Area        1     7.514 135.55 63.865
## - Illiteracy  1     8.299 136.33 64.154
## - Frost       1     9.260 137.29 64.505
## - Population  1    25.719 153.75 70.166
## - `Life Exp`  1   127.175 255.21 95.503
##
## Step:  AIC=61.11
## Murder ~ Population + Illiteracy + `Life Exp` + `HS Grad` + Frost +
##     Area
##
##              Df Sum of Sq    RSS    AIC
## - `HS Grad`   1     0.763 129.03 59.402
## <none>                    128.27 61.105
## - Area        1     7.310 135.58 61.877
## - Illiteracy  1     8.715 136.98 62.392
## - Frost       1     9.345 137.61 62.621
## - Population  1    27.142 155.41 68.702
## - `Life Exp`  1   127.500 255.77 93.613
##
## Step:  AIC=59.4
## Murder ~ Population + Illiteracy + `Life Exp` + Frost + Area
##
##              Df Sum of Sq    RSS    AIC
## <none>                    129.03 59.402
## - Illiteracy  1     8.723 137.75 60.672
## - Frost       1    11.030 140.06 61.503
## - Area        1    15.937 144.97 63.225
## - Population  1    26.415 155.45 66.714
## - `Life Exp`  1   140.391 269.42 94.213
```

```r
summary(fit4)
```

```
##
## Call:
## lm(formula = Murder ~ Population + Illiteracy + `Life Exp` +
##     Frost + Area, data = as.data.frame(data))
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2976 -1.0711 -0.1123  1.1092  3.4671
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.202e+02  1.718e+01   6.994 1.17e-08 ***
## Population   1.780e-04  5.930e-05   3.001  0.00442 **
## Illiteracy   1.173e+00  6.801e-01   1.725  0.09161 .
## `Life Exp`  -1.608e+00  2.324e-01  -6.919 1.50e-08 ***
## Frost       -1.373e-02  7.080e-03  -1.939  0.05888 .
## Area         6.804e-06  2.919e-06   2.331  0.02439 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.712 on 44 degrees of freedom
## Multiple R-squared:  0.8068, Adjusted R-squared:  0.7848
## F-statistic: 36.74 on 5 and 44 DF,  p-value: 1.221e-14
```

```
#Logistic Regression

#Logistic regression is used when the forecast variable is categorical and
the predictor
#variables either or both categorical and continous

Titanic <- read.csv("http://www.hodgett.co.uk/titanic.csv", header=TRUE)


#We are going to use survived as the forecast variable and sex, age and fare
as the
#predictor variables, so first we will remove all of the rows where survived
is NA using

Titanic <- Titanic[which(!is.na(Titanic$survived)),]

train <- Titanic[1:1000,]

test <- Titanic[1001:1309,]

#use logistic regression on training data set

lfit <- glm(survived ~ sex + age + fare, family=binomial, data=train)

summary(lfit)
```

```
##
## Call:
## glm(formula = survived ~ sex + age + fare, family = binomial,
```

```
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4110  -0.6709  -0.5796   0.7103   2.0984
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.433870   0.232716   6.161 7.21e-10 ***
## sexmale     -2.580053   0.177306 -14.551  < 2e-16 ***
## age         -0.015309   0.006122  -2.501   0.0124 *
## fare         0.009548   0.001923   4.966 6.85e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.19  on 860  degrees of freedom
## Residual deviance:  847.86  on 857  degrees of freedom
##   (139 observations deleted due to missingness)
## AIC: 855.86
##
## Number of Fisher Scoring iterations: 5
```

```r
#sex and fare are statistically significant with sex having the lowest p-
value suggesting a
#strong association of the sex of the passenger with the probability of
having survived

#We can assess the model using the test set. We extract the sex, age and fare
columns from
#the data set

test2 <- subset(test,select=c(4,5,9))

results <- predict(lfit, newdata=test2,type='response')

#The returned results are between 0 and 1, but we need 0 and 1. Any value
below 0.5
#is 0 while above is assigned 1

results <- ifelse(results > 0.5,1,0)

#Calculate the accuracy of the model

mean(results == test$survived)
```

```
## [1] NA
```