Project_DirectedStudies

Sarah Chopra

01/12/2022

Census Data: For this project I have used data from https://archive.ics.uci.edu/ (https://archive.ics.uci.edu/).

The data set is called census income data. The dataset contains continuous and categorical data. For this project, I have chosen the predicted column to be income and predcitor as number of hours people work.

The columm income is a categorical column, it has values >=50K or <50K, depicting whether the income is greater than 50K or not. Here, the function raw_data() is from my user defined package called "UserDefinedPackage" which reads the data and returns a dataframe with proper datatype.

```
library(ggplot2)
library(UserDefinedPackage)
data <- raw_data()
head(data)
```

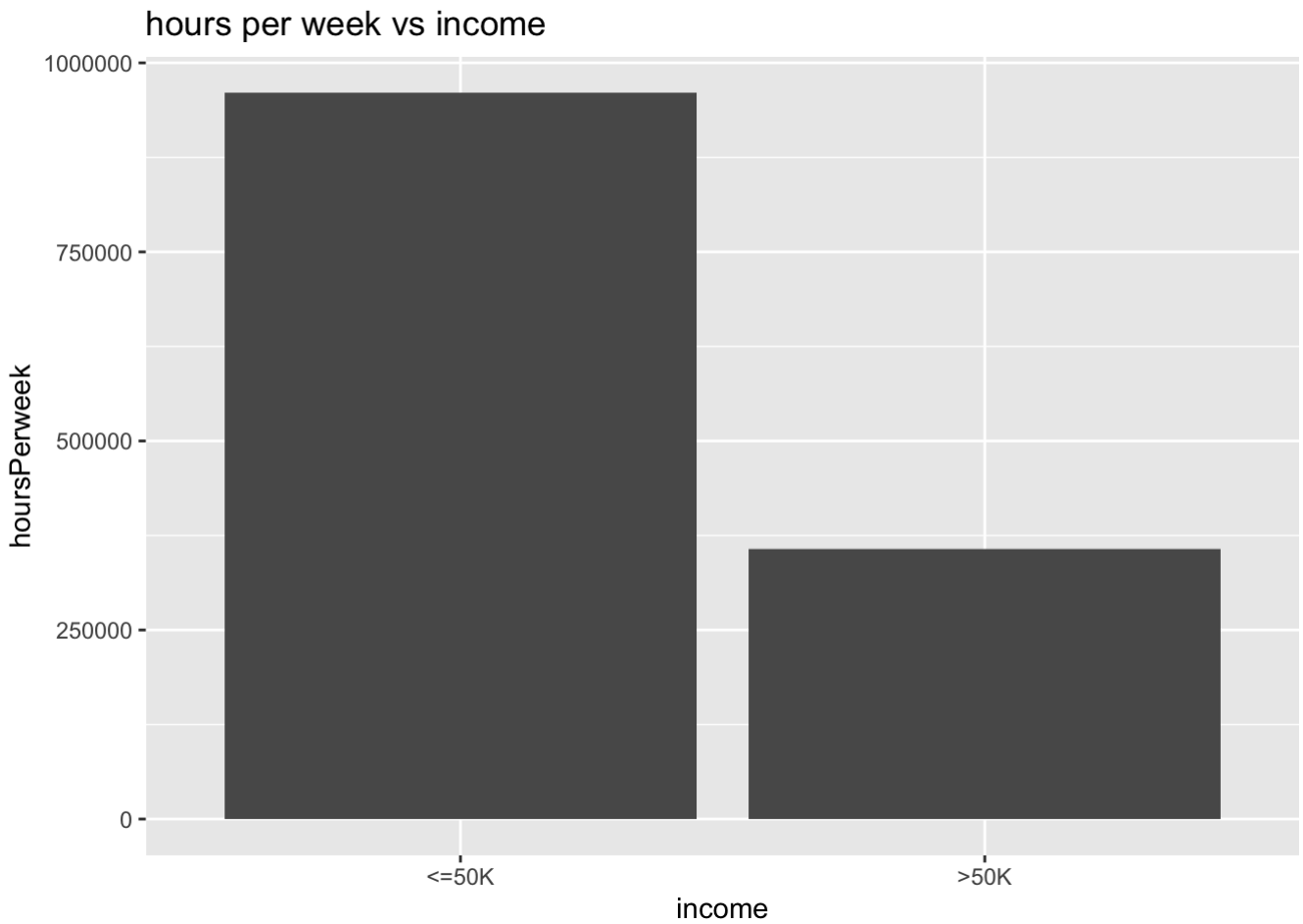
```
##   age         workclass  fnlwgt  education educationNum       maritalStatus
## 1 39,        State-gov,  77516, Bachelors,          13,      Never-married,
## 2 50, Self-emp-not-inc,  83311, Bachelors,          13, Married-civ-spouse,
## 3 38,          Private, 215646,   HS-grad,           9,           Divorced,
## 4 53,          Private, 234721,      11th,           7, Married-civ-spouse,
## 5 28,          Private, 338409, Bachelors,          13, Married-civ-spouse,
## 6 37,          Private, 284582,   Masters,          14, Married-civ-spouse,
##           occupation   relationship   race     sex capitalGain capitalLoss
## 1      Adm-clerical, Not-in-family, White,   Male,       2174,          0,
## 2   Exec-managerial,       Husband, White,   Male,          0,          0,
## 3 Handlers-cleaners, Not-in-family, White,   Male,          0,          0,
## 4 Handlers-cleaners,       Husband, Black,   Male,          0,          0,
## 5    Prof-specialty,          Wife, Black, Female,          0,          0,
## 6   Exec-managerial,          Wife, White, Female,          0,          0,
##   hoursPerweek  nativeCountry income
## 1           40 United-States,  <=50K
## 2           13 United-States,  <=50K
## 3           40 United-States,  <=50K
## 4           40 United-States,  <=50K
## 5           40          Cuba,  <=50K
## 6           40 United-States,  <=50K
```

```
#ggplot(data$hoursPerweek~data$income,order = as.numeric(data$income))

ggplot(data = data, aes(y=hoursPerweek,x=income, order = as.numeric(income)), color =
"lightgrey") + geom_bar(stat="identity")+ggtitle("hours per week vs income")
```

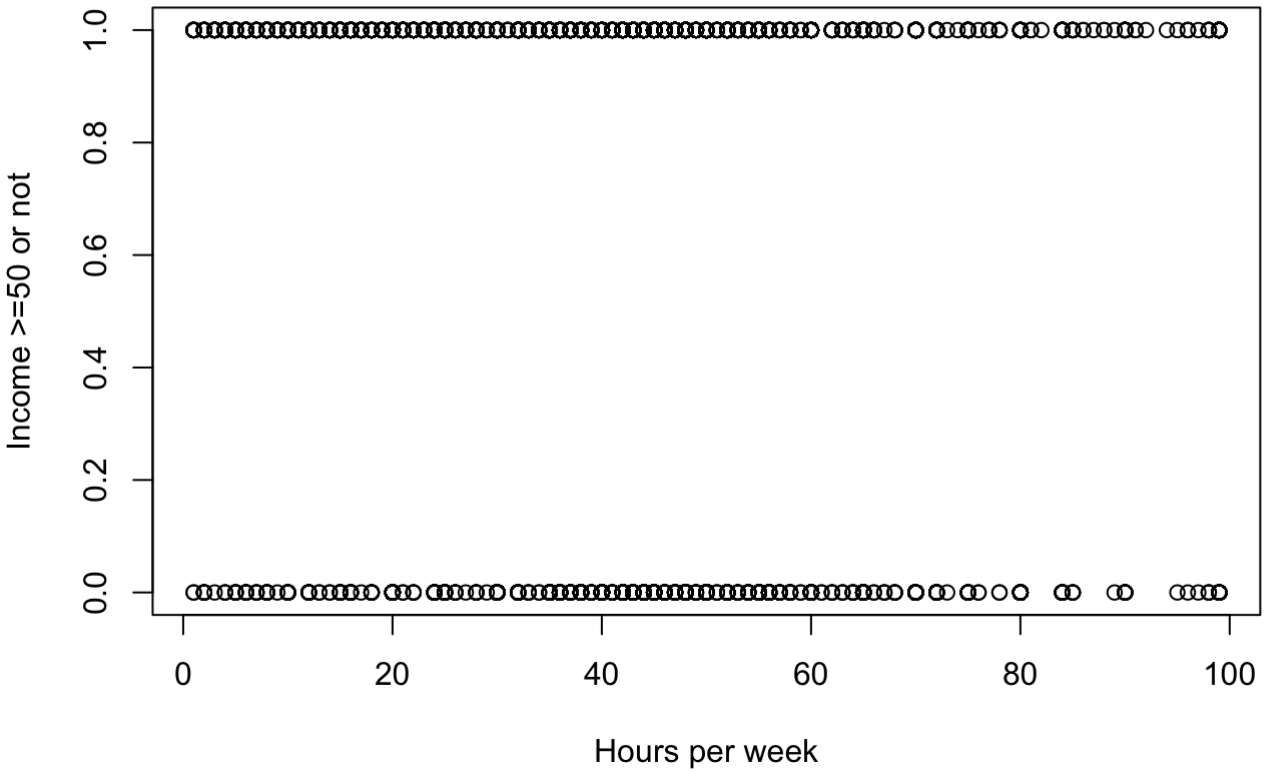
```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion

## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```



Now, we will read the data again and fetch only the specific columns that we need for our logistic regression. We will make use of another function called read_mydata() to do this.

```
all_data <- read_mydata()
plot(all_data$data.x,all_data$data.y, xlab = "Hours per week", ylab = "Income >=50 or
not")
```




```
#plot(all_data$data.y)
```

The data that I have picked, does not contain an missing values, hence, Im creating those values using function includeNas, randomly some NAs are generated.

```
all_data_withNA <- includeNas(all_data)
```

Once, NAs are generated, I'm using Mice package and have built a function ImputationFix to fix these imputations.

```
all_data_withoutNA <- ImputationFix(all_data_withNA)
```

```
##
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
##
##  iter imp variable
##   1   1  data.y  data.x
##   1   2  data.y  data.x
##   1   3  data.y  data.x
##   1   4  data.y  data.x
##   1   5  data.y  data.x
##   1   6  data.y  data.x
##   1   7  data.y  data.x
##   2   1  data.y  data.x
##   2   2  data.y  data.x
##   2   3  data.y  data.x
##   2   4  data.y  data.x
##   2   5  data.y  data.x
##   2   6  data.y  data.x
##   2   7  data.y  data.x
##   3   1  data.y  data.x
##   3   2  data.y  data.x
##   3   3  data.y  data.x
##   3   4  data.y  data.x
##   3   5  data.y  data.x
##   3   6  data.y  data.x
##   3   7  data.y  data.x
##   4   1  data.y  data.x
##   4   2  data.y  data.x
##   4   3  data.y  data.x
##   4   4  data.y  data.x
##   4   5  data.y  data.x
##   4   6  data.y  data.x
##   4   7  data.y  data.x
##   5   1  data.y  data.x
##   5   2  data.y  data.x
##   5   3  data.y  data.x
##   5   4  data.y  data.x
##   5   5  data.y  data.x
##   5   6  data.y  data.x
##   5   7  data.y  data.x
## Class: mids
## Number of multiple imputations:  7
## Imputation methods:
## data.y data.x
## "mean" "mean"
## PredictorMatrix:
##        data.y data.x
## data.y      0      1
## data.x      1      0
```

Once, we have the NAs removed, we will fit a logisitic regression using three methods.

Method 1: Using Optim, the function OptimUserDefined is a function in my package UserDefinedPackage, which in turns calls Optim function.

```
OptimCoef <- OptimUserDefined()
```

```
## [1]  3.10084423 -0.04646653
```

```
print(OptimCoef)
```

```
## [1]  3.10084423 -0.04646653
```

Method 2: Using In build GLM function. The function, InbuildGlmUserDefined is a user defined function in my package UserDefinedPackage which called GLM.

```
InBuildGLM <- InbuildGlmUserDefined(all_data_withoutNA)
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
##
## Call:  glm(formula = y ~ x1, family = binomial(link = "logit"))
##
## Coefficients:
## (Intercept)           x1
##      3.0977      -0.0464
##
## Degrees of Freedom: 32560 Total (i.e. Null);  32559 Residual
## Null Deviance:       35940
## Residual Deviance: 34190     AIC: 34200
## [1] "Coefficients are:  3.09773731594719"
## [2] "Coefficients are:  -0.0464026421030457"
```

```
print(InBuildGLM)
```

```
## [1] "Coefficients are:  3.09773731594719"
## [2] "Coefficients are:  -0.0464026421030457"
```

Method 3: Using User defined Newton Raphson. The function NewtonRaphsonUserDefined defined newton raphson method.

```
NewtonRaphson <- NewtonRaphsonUserDefined(all_data_withoutNA)
print(NewtonRaphson)
```

```
##           b0          b1
## 151 3.036054 -0.04499915
```