hw_2_3_v2

Grigoreva Elizaveta

6/7/2020

```r
library(mlbench)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(boot)
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':
##
##     melanoma
```

```r
library(tree)
library(class)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
library(ggfortify)
library(RColorBrewer)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

Load data, check NA, duplicates

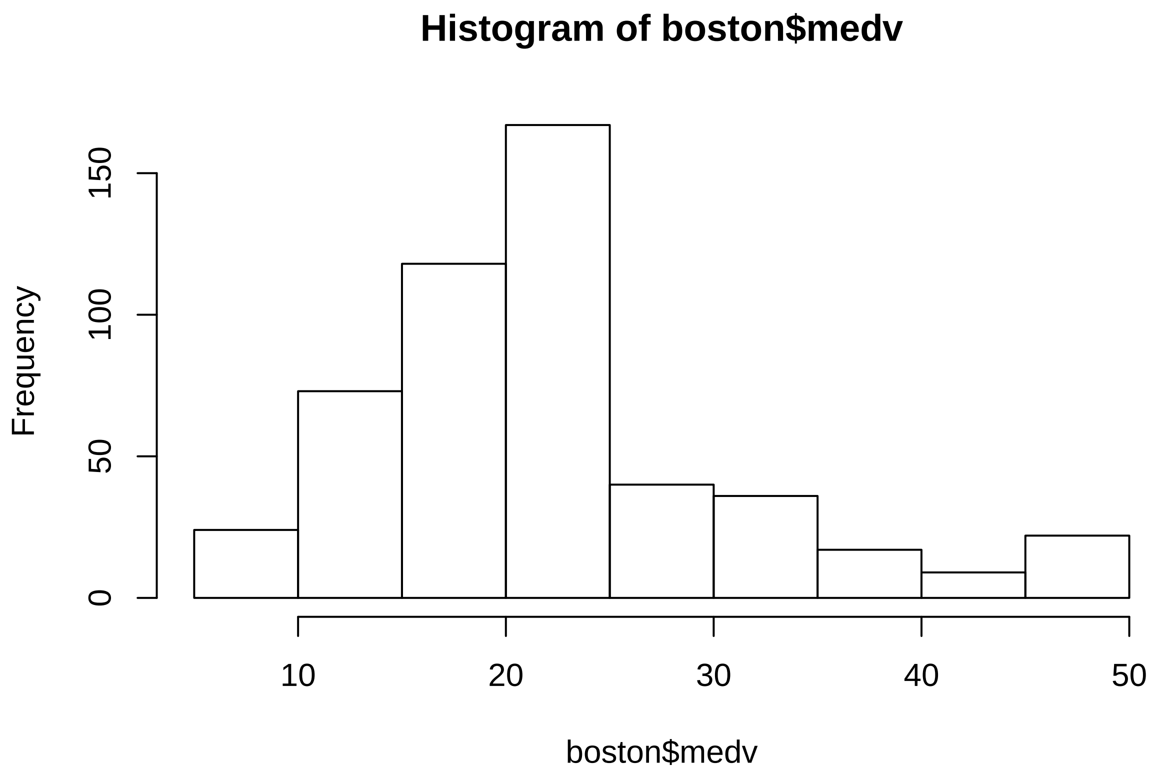
```r
data("BostonHousing")
boston <-  BostonHousing
str(boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : num  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ b      : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```r
summary(boston)
```

```
##       crim                zn             indus       chas        nox        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   0:471   Min.   :0.3850  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1: 35   1st Qu.:0.4490  
##  Median : 0.25651   Median :  0.00   Median : 9.69           Median :0.5380  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14           Mean   :0.5547  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10           3rd Qu.:0.6240  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74           Max.   :0.8710  
##        rm             age              dis              rad        
##  Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000  
##  1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000  
##  Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000  
##  Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549  
##  3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000  
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000  
##       tax           ptratio             b              lstat      
##  Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73  
##  1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95  
##  Median :330.0   Median :19.05   Median :391.44   Median :11.36  
##  Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65  
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95  
##  Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97  
##       medv      
##  Min.   : 5.00  
##  1st Qu.:17.02  
##  Median :21.20  
##  Mean   :22.53  
##  3rd Qu.:25.00  
##  Max.   :50.00
```

```r
hist(boston$medv)
```




```r
summary(is.na(boston))
```

```
##     crim              zn            indus            chas        
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical  
##  FALSE:506       FALSE:506       FALSE:506       FALSE:506      
##     nox              rm             age              dis         
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical  
##  FALSE:506       FALSE:506       FALSE:506       FALSE:506      
##     rad             tax           ptratio            b          
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical  
##  FALSE:506       FALSE:506       FALSE:506       FALSE:506      
##    lstat            medv        
##  Mode :logical   Mode :logical  
##  FALSE:506       FALSE:506
```

```r
summary(duplicated(boston))
```

```
##    Mode   FALSE 
## logical     506
```

Create subset w/o Charles River dummy variable and paerform correlation plot

```r
summary(is.na(boston))
```

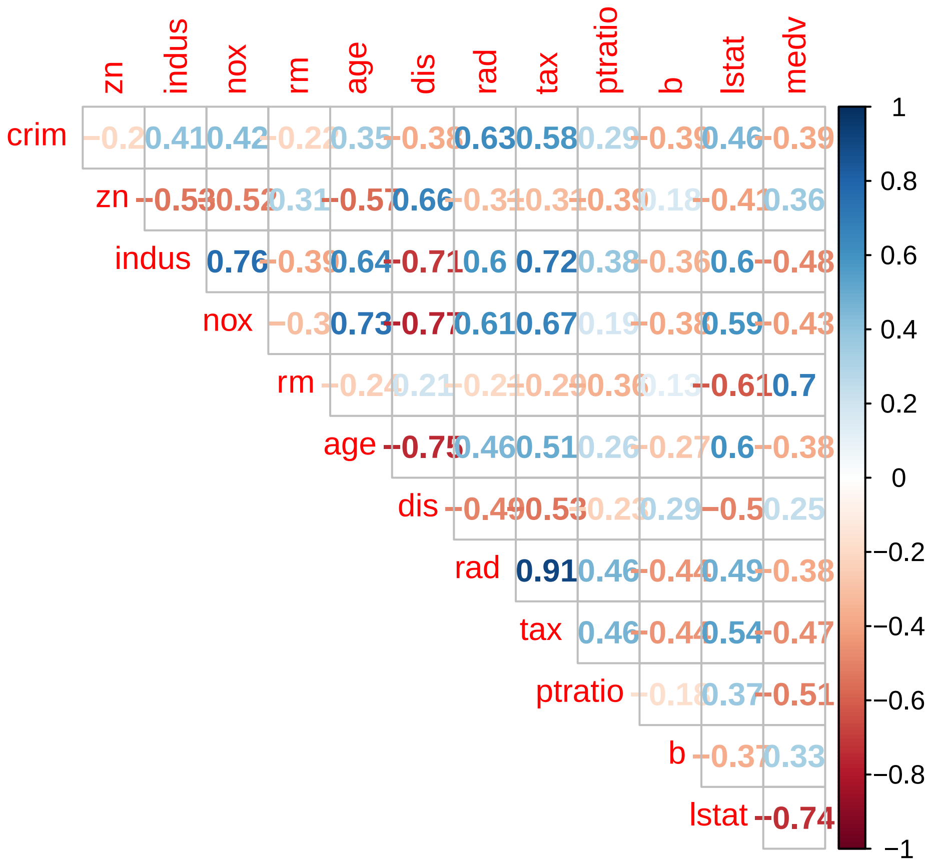
```
##     crim              zn            indus            chas        
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical  
##  FALSE:506       FALSE:506       FALSE:506       FALSE:506      
##     nox              rm             age              dis         
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical  
##  FALSE:506       FALSE:506       FALSE:506       FALSE:506      
##     rad             tax           ptratio            b          
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
```

```
##  FALSE:506       FALSE:506       FALSE:506       FALSE:506
##    lstat            medv
##  Mode :logical   Mode :logical
##  FALSE:506       FALSE:506
```

```r
summary(duplicated(boston))
```

```
##    Mode   FALSE
## logical     506
```

```r
#Remove  Charles River variable
boston_sub<-  boston[,-4]
corrplot(cor(boston_sub), method="number", type="upper", diag=F)
```



Create categorical variable randomly

```r
boston_sub1 <- boston_sub
boston_sub1 <- data.table(boston_sub1)

boston_sub1[,medv_Cat:= cut(medv,c(0,quantile(boston_sub1$medv, 0.25),quantile(boston_sub1$medv, 0.50),

#Table it
boston_sub1[,table(medv_Cat)]
```

```
## medv_Cat
##   1   2   3   4
## 127 129 126 124
```

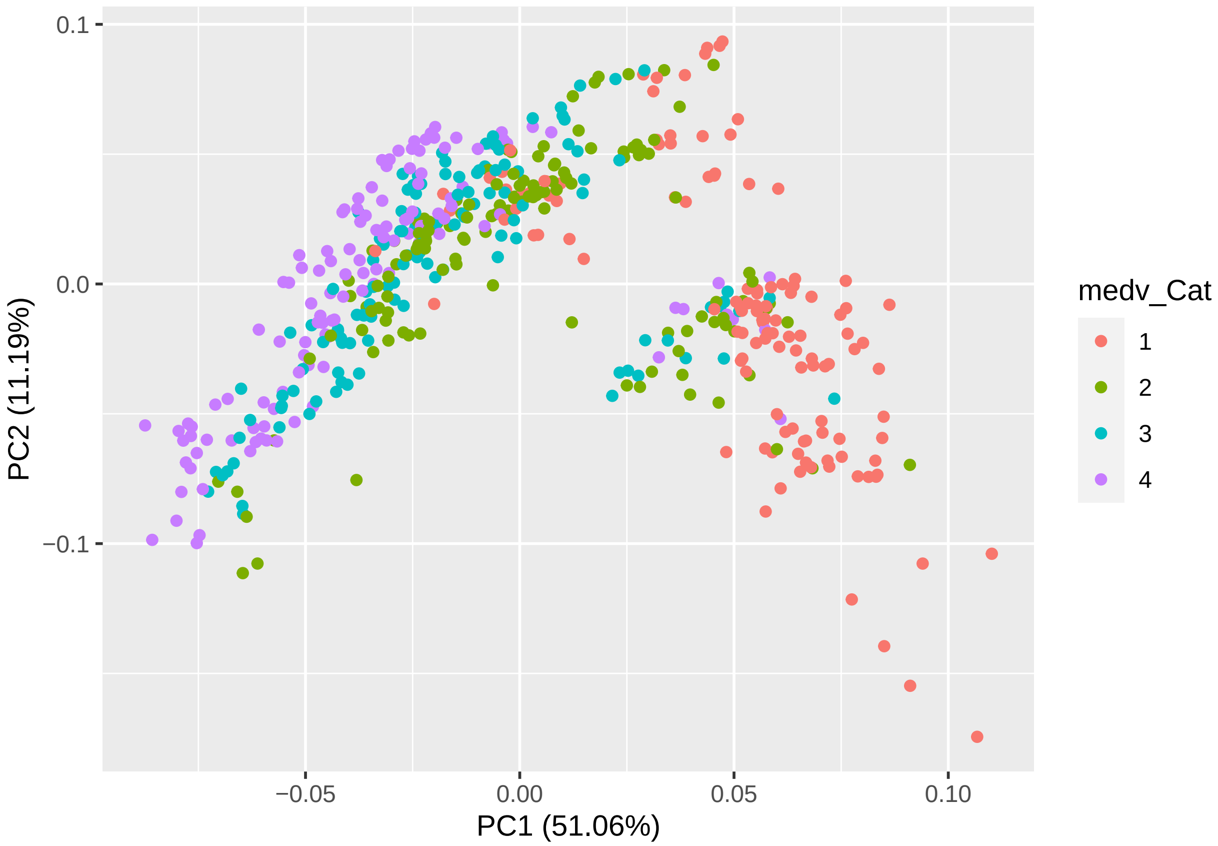
```r
summary(boston_sub1)
```

```
##       crim                zn             indus            nox        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.3850  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.4490  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.5380  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.5547  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.6240  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :0.8710  
##        rm             age              dis              rad        
##  Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000  
##  1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000  
##  Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000  
##  Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549  
##  3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000  
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000  
##       tax           ptratio            b               lstat      
##  Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73  
##  1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95  
##  Median :330.0   Median :19.05   Median :391.44   Median :11.36  
##  Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65  
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95  
##  Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97  
##       medv       medv_Cat
##  Min.   : 5.00   1:127   
##  1st Qu.:17.02   2:129   
##  Median :21.20   3:126   
##  Mean   :22.53   4:124   
##  3rd Qu.:25.00           
##  Max.   :50.00           
```

```r
boston_df1 <- boston_sub1[,-13]
#Perform PCA based on this classification
```

PCA

```r
boston_pca <- boston_df1[, -13]
pca <- prcomp(boston_pca, center = TRUE,
              scale. = TRUE)
autoplot(pca, data = boston_df1, colour = "medv_Cat")
```




```
#So we have 5 categories but 2 clear clusters, let's make 2 clusters
```

```
#Try to set up 2 clusters
boston_sub <- data.table(boston_sub)
boston_sub[,medv_Cat:= cut(medv,c(0,quantile(boston_sub$medv, 0.35), max(boston_sub$medv)),labels=c("yes

boston_sub[,table(medv_Cat)]
```

```
## medv_Cat
## yes  no
## 179 327
```

```
boston_df <- boston_sub[,-13]
boston_pca <- boston_df[, -13]
pca <- prcomp(boston_pca, center = TRUE,
              scale. = TRUE)
autoplot(pca, data = boston_df, colour = "medv_Cat")
```

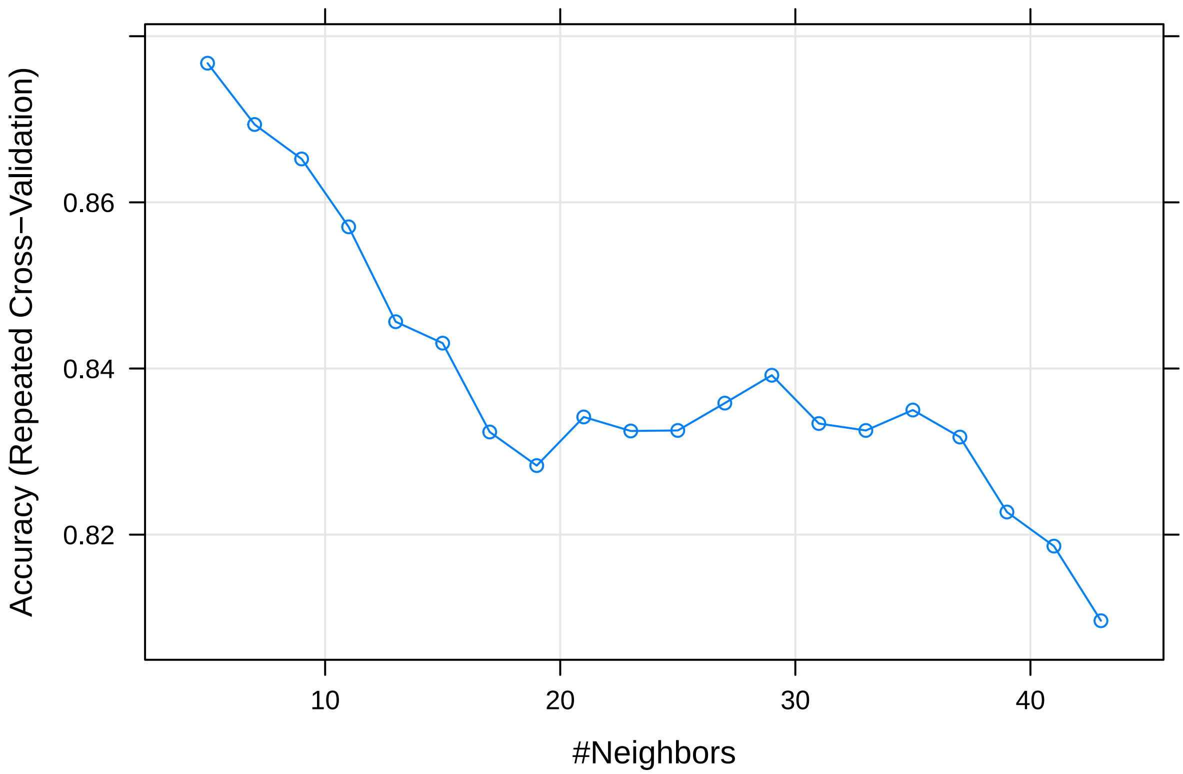
KNN for predict to which category some observation belongs to

```
set.seed(42)

for_train <- createDataPartition(y = boston_sub$medv_Cat, p= 0.8, list = FALSE)
training <- boston_sub[for_train,-13]
testing <- boston_sub[-for_train,-13]



knn <- train(medv_Cat ~ ., data = training, method = "knn",
 trControl=trainControl(method = "repeatedcv", number = 10, repeats = 3),
 preProcess = c("center", "scale"),
 tuneLength = 20) #Range of k?

knn
```

```
## k-Nearest Neighbors
##
## 406 samples
##  12 predictor
##   2 classes: 'yes', 'no'
##
## Pre-processing: centered (12), scaled (12)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 365, 366, 366, 365, 365, 365, ...
## Resampling results across tuning parameters:
##
```

```
##    k  Accuracy   Kappa    
##     5  0.8767480  0.7241543
##     7  0.8693699  0.7097523
##     9  0.8652236  0.7025707
##    11  0.8570528  0.6839796
##    13  0.8456301  0.6573275
##    15  0.8430488  0.6521292
##    17  0.8323577  0.6273424
##    19  0.8283130  0.6198062
##    21  0.8341667  0.6307289
##    23  0.8324797  0.6239412
##    25  0.8325407  0.6227029
##    27  0.8358333  0.6314846
##    29  0.8391870  0.6382910
##    31  0.8333740  0.6251189
##    33  0.8325407  0.6235014
##    35  0.8350000  0.6295236
##    37  0.8317480  0.6216412
##    39  0.8227236  0.6018037
##    41  0.8186179  0.5934927
##    43  0.8096341  0.5736192
## 
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```r
plot(knn)
```




```r
print(paste0("best k to minimize MSE: ", knn$bestTune))
```

```
## [1] "best k to minimize MSE: 5"
```

```
#Test this model
test_pred <- predict(knn, newdata =testing)
table(test_pred, Real = testing$medv_Cat)
```

```
##          Real
## test_pred yes no
##       yes  30  3
##       no    5 62
```

```
knn_fit <- knn3Train(train = training[,-13], test = testing[,-13], k=7, cl = training$medv_Cat)
table(knn_fit, Real = testing$medv_Cat)
```

```
##        Real
## knn_fit yes no
##     no    9 59
##     yes  26  6
```

```
#accuracy is a ratio of correctly predicted observation to the total observations. (TP+TN)/(FP+FN+TP+TN)
accuracy = sum(knn_fit == testing$medv_Cat)/length(testing$medv_Cat)
paste0("Accuracy:", accuracy)
```

```
## [1] "Accuracy:0.85"
```

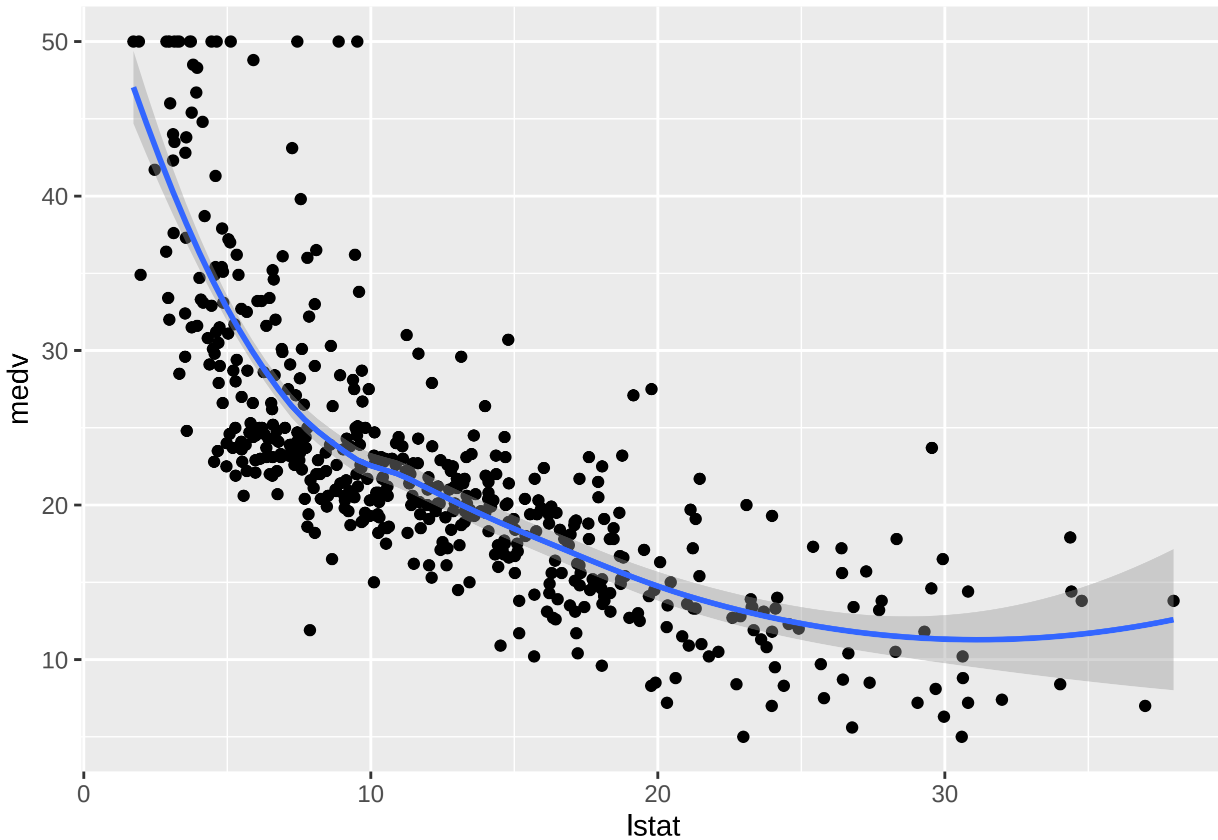
Logistic regression

```
#Check multicol
mean(boston$medv)
```

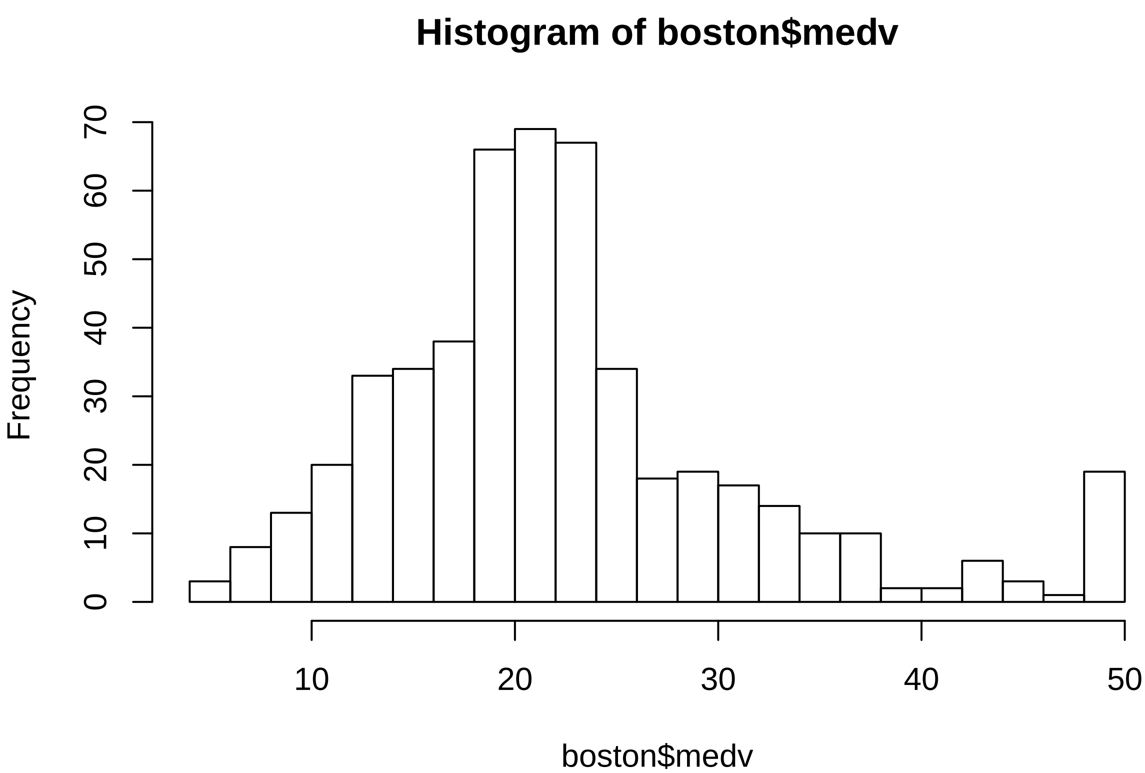
```
## [1] 22.53281
```

```
#Create binomial variable and separate to the hight and low level of criminal based on median

ggplot(boston,aes(lstat,medv))+geom_point()+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
cv.err <- 1:5
hist(boston$medv, median(boston$medv))
```



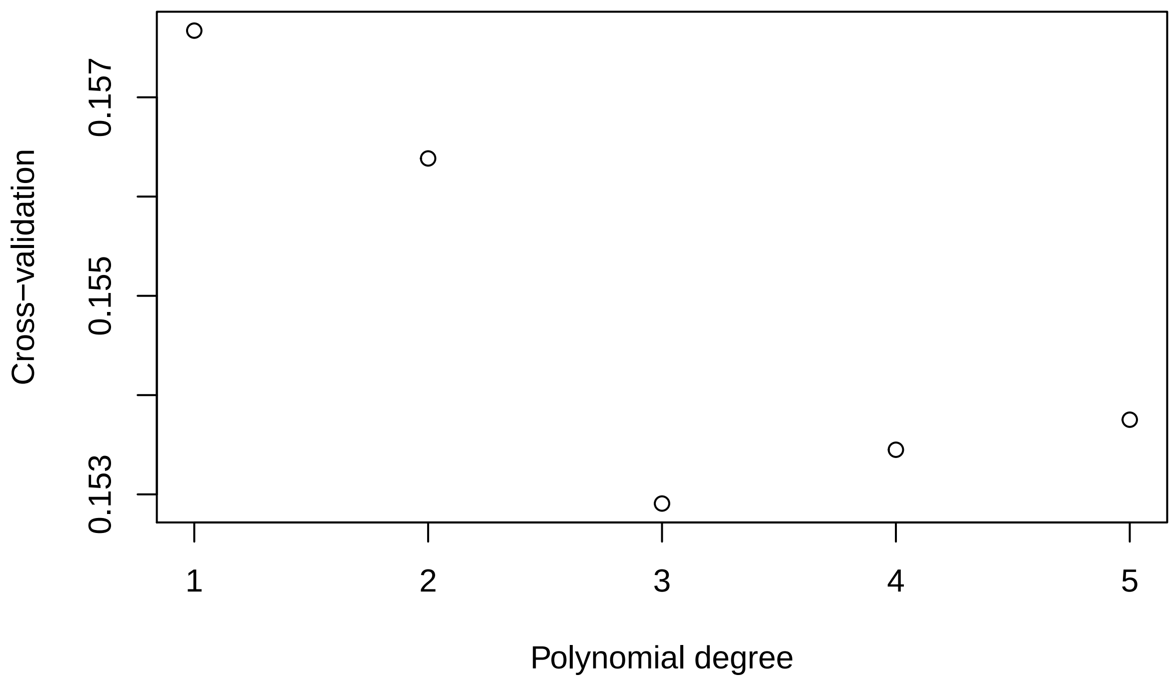
```r
median(boston$medv)
```

```
## [1] 21.2
```

```r
boston_bi <- boston %>% mutate(medv_bi = ifelse(medv >= 22.5, "high", "low")) %>%
  mutate(medv_bi = factor(medv_bi, levels = c("high", "low")))
boston_bi <- boston_bi[,-c(13:14)]
for (i in 1:5){
  gl <- glm(medv_bi ~ poly(rm, i), family = "binomial",
            data = boston_bi)
  cv.err[i] <- cv.glm(boston_bi, gl)$delta[1]
}
cv.err
```

```
## [1] 0.1576716 0.1563841 0.1529078 0.1534497 0.1537524
```

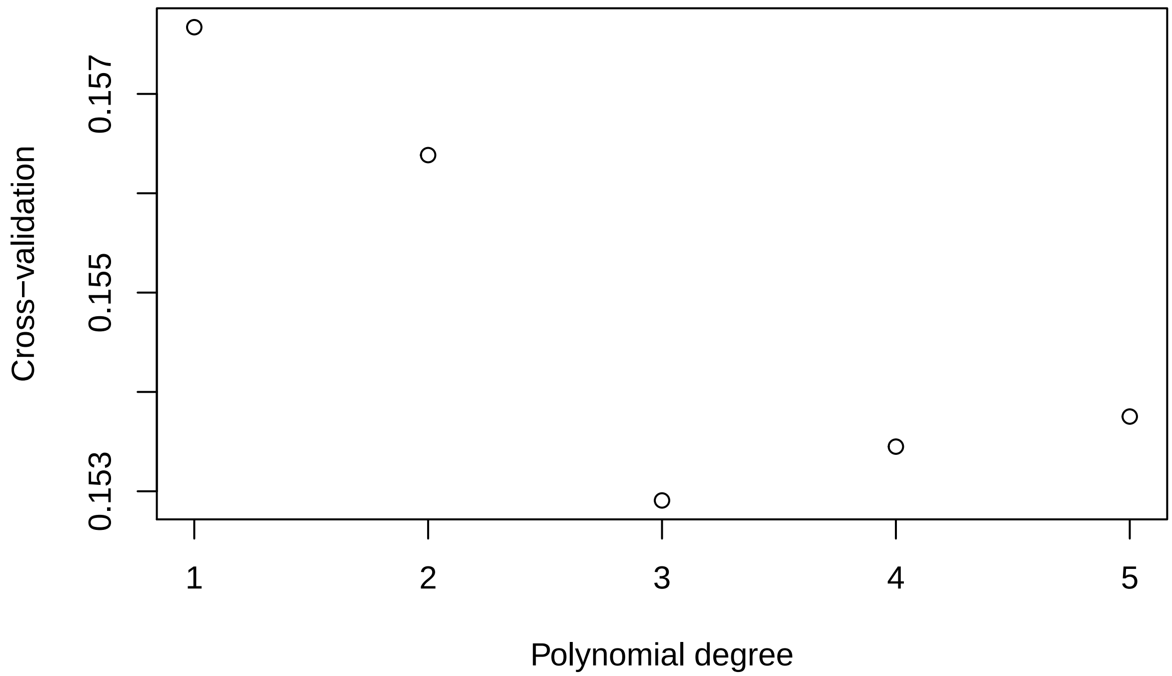
```r
plot(x = 1:5, y = cv.err,
     xlab = 'Polynomial degree', ylab='Cross-validation')
```




```r
# on each iteration, create the model for the given power and test MSE
for (i in 1:5){
  gl <- glm(medv_bi ~ poly(rm, i), family = "binomial",
            data = boston_bi)
  cv.err[i] <- cv.glm(boston_bi, gl)$delta[1]
}
cv.err
```

```
## [1] 0.1576716 0.1563841 0.1529078 0.1534497 0.1537524
```

```r
plot(x = 1:5, y = cv.err,
     xlab = 'Polynomial degree', ylab='Cross-validation')
```




```r
# use 3rd degree
glm <- glm(medv_bi ~ poly(rm, 3), data = boston_bi, family = "binomial")
summary(glm)
```

```
##
## Call:
## glm(formula = medv_bi ~ poly(rm, 3), family = "binomial", data = boston_bi)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5703  -0.6721   0.3852   0.7137   2.4747
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.3929     0.1198   3.279  0.00104 **
## poly(rm, 3)1 -34.8454     3.5521  -9.810  < 2e-16 ***
## poly(rm, 3)2  -5.0733     3.1524  -1.609  0.10754
## poly(rm, 3)3  18.5221     3.0243   6.124  9.1e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 688.12  on 505  degrees of freedom
## Residual deviance: 469.84  on 502  degrees of freedom
## AIC: 477.84
##
## Number of Fisher Scoring iterations: 5
```

```r
pred_glm <- predict(glm, type = "response") > 0.5
table(pred_glm, Real = boston_bi[, 13])
```

```
##         Real
```

```
## pred_glm high low
##    FALSE  146  41
##    TRUE    66 253
```

```r
boston_df <- boston_sub[,-13]
test <- sample(nrow(boston_df), 0.8*nrow(boston_df))
glm2  <- glm(medv_bi ~ rm, data = boston_bi[-test, ],
             family = "binomial")
```

#Summary

```r
summary(glm2)
```

```
##
## Call:
## glm(formula = medv_bi ~ rm, family = "binomial", data = boston_bi[-test,
##     ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2001  -1.0995   0.6616   0.8873   2.4534
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   9.2227     2.6394   3.494 0.000475 ***
## rm           -1.3874     0.4201  -3.303 0.000958 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 133.62  on 101  degrees of freedom
## Residual deviance: 118.13  on 100  degrees of freedom
## AIC: 122.13
##
## Number of Fisher Scoring iterations: 4
```

```r
pred_glm2 <- predict(glm2, type = "response", newdata = boston_bi[test, ]) > 0.5
table(pred_glm2, Real = boston_bi[test, 13])
```

```
##          Real
## pred_glm2 high low
##     FALSE   88  11
##     TRUE    87 218
```