# Similarity: Regression

Valari Graham

2022-10-08

## Regression

This notebook performs regression using a linear model, kNN, and decision trees. # Data Set Link to data set in UCI Machine Learning Repository "https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset"

Load in data set and remove column 'date'

```
library(tree)
library(MASS)
library(dplyr)
```

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

## The following object is masked from 'package:MASS':
##
##     select

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

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

```
library(ggplot2)
bike1 <- read.csv("hour.csv")
bike <- select(bike1, -c(dteday, season, weathersit))
```

## Linear Regression Model

Split into train and test data

```
set.seed(1234)
i <- sample(nrow(bike), 0.8*nrow(bike), replace = FALSE)
train <- bike[i,]
test <- bike[-i,]
```

Brief Statistical Exploration of the Data

```r
str(train)
```

```
## 'data.frame':    13903 obs. of  14 variables:
##  $ instant   : int  7452 8016 7162 8086 9196 623 15241 10885 934 12688 ...
##  $ yr        : int  0 0 0 0 1 0 1 1 0 1 ...
##  $ mnth      : int  11 12 10 12 1 1 10 4 2 6 ...
##  $ hr        : int  2 15 0 13 1 4 5 16 12 20 ...
##  $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : int  6 1 1 4 2 6 2 2 5 0 ...
##  $ workingday: int  0 1 1 1 1 0 1 1 1 0 ...
##  $ temp      : num  0.24 0.46 0.26 0.3 0.32 0.16 0.56 0.62 0.22 0.62 ...
##  $ atemp     : num  0.258 0.455 0.303 0.273 0.303 ...
##  $ hum       : num  0.65 0.72 0.87 0.49 0.93 0.69 0.83 0.21 0.47 0.57 ...
##  $ windspeed : num  0.0896 0.0896 0 0.3582 0.2537 ...
##  $ casual    : int  7 16 3 9 2 1 1 145 7 101 ...
##  $ registered: int  39 132 20 115 7 2 42 340 64 201 ...
##  $ cnt       : int  46 148 23 124 9 3 43 485 71 302 ...
```

```r
summary(train)
```

```
##     instant            yr              mnth              hr       
##  Min.   :    1   Min.   :0.0000   Min.   : 1.000   Min.   : 0.00  
##  1st Qu.: 4386   1st Qu.:0.0000   1st Qu.: 4.000   1st Qu.: 6.00  
##  Median : 8768   Median :1.0000   Median : 7.000   Median :12.00  
##  Mean   : 8730   Mean   :0.5064   Mean   : 6.549   Mean   :11.55  
##  3rd Qu.:13054   3rd Qu.:1.0000   3rd Qu.:10.000   3rd Qu.:18.00  
##  Max.   :17379   Max.   :1.0000   Max.   :12.000   Max.   :23.00  
##     holiday           weekday        workingday          temp       
##  Min.   :0.00000   Min.   :0.000   Min.   :0.0000   Min.   :0.0200  
##  1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.3400  
##  Median :0.00000   Median :3.000   Median :1.0000   Median :0.5000  
##  Mean   :0.02877   Mean   :3.011   Mean   :0.6821   Mean   :0.4972  
##  3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.0000   3rd Qu.:0.6600  
##  Max.   :1.00000   Max.   :6.000   Max.   :1.0000   Max.   :1.0000  
##      atemp             hum           windspeed          casual      
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :  0.00  
##  1st Qu.:0.3333   1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.:  4.00  
##  Median :0.4848   Median :0.6300   Median :0.1940   Median : 17.00  
##  Mean   :0.4760   Mean   :0.6265   Mean   :0.1896   Mean   : 35.89  
##  3rd Qu.:0.6212   3rd Qu.:0.7800   3rd Qu.:0.2537   3rd Qu.: 49.00  
##  Max.   :0.9848   Max.   :1.0000   Max.   :0.8507   Max.   :367.00  
##    registered         cnt       
##  Min.   :  0.0   Min.   :  1.0  
##  1st Qu.: 35.0   1st Qu.: 41.0  
##  Median :117.0   Median :144.0  
##  Mean   :154.9   Mean   :190.8  
##  3rd Qu.:221.0   3rd Qu.:282.0  
##  Max.   :886.0   Max.   :977.0
```

```r
head(train)
```

```
##       instant yr mnth hr holiday weekday workingday temp  atemp  hum windspeed
```

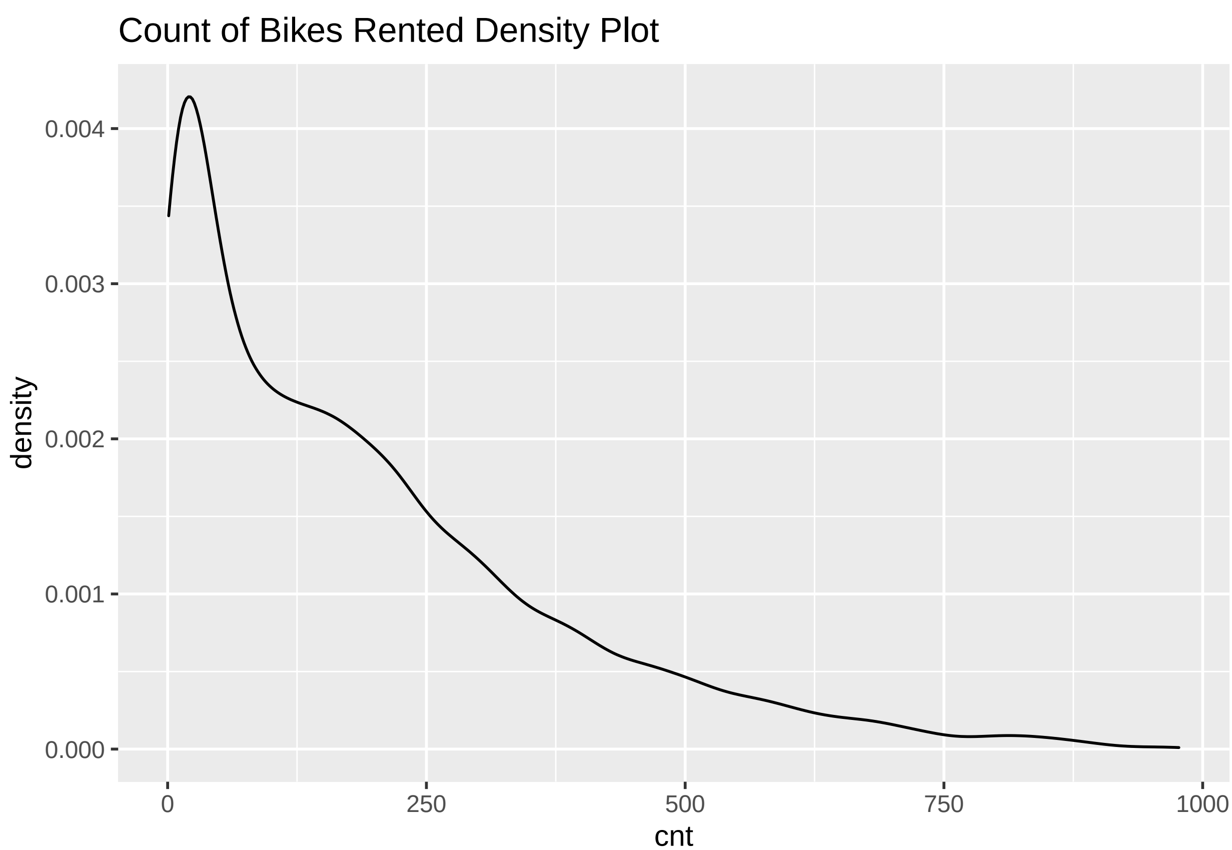
```
## 7452    7452  0    11   2        0        6           0 0.24 0.2576 0.65     0.0896
## 8016    8016  0    12  15        0        1           1 0.46 0.4545 0.72     0.0896
## 7162    7162  0    10   0        0        1           1 0.26 0.3030 0.87     0.0000
## 8086    8086  0    12  13        0        4           1 0.30 0.2727 0.49     0.3582
## 9196    9196  1     1   1        0        2           1 0.32 0.3030 0.93     0.2537
## 623      623  0     1   4        0        6           0 0.16 0.1818 0.69     0.1045
##        casual registered cnt
## 7452        7         39  46
## 8016       16        132 148
## 7162        3         20  23
## 8086        9        115 124
## 9196        2          7   9
## 623         1          2   3
```

```
sum(is.na(train))
```

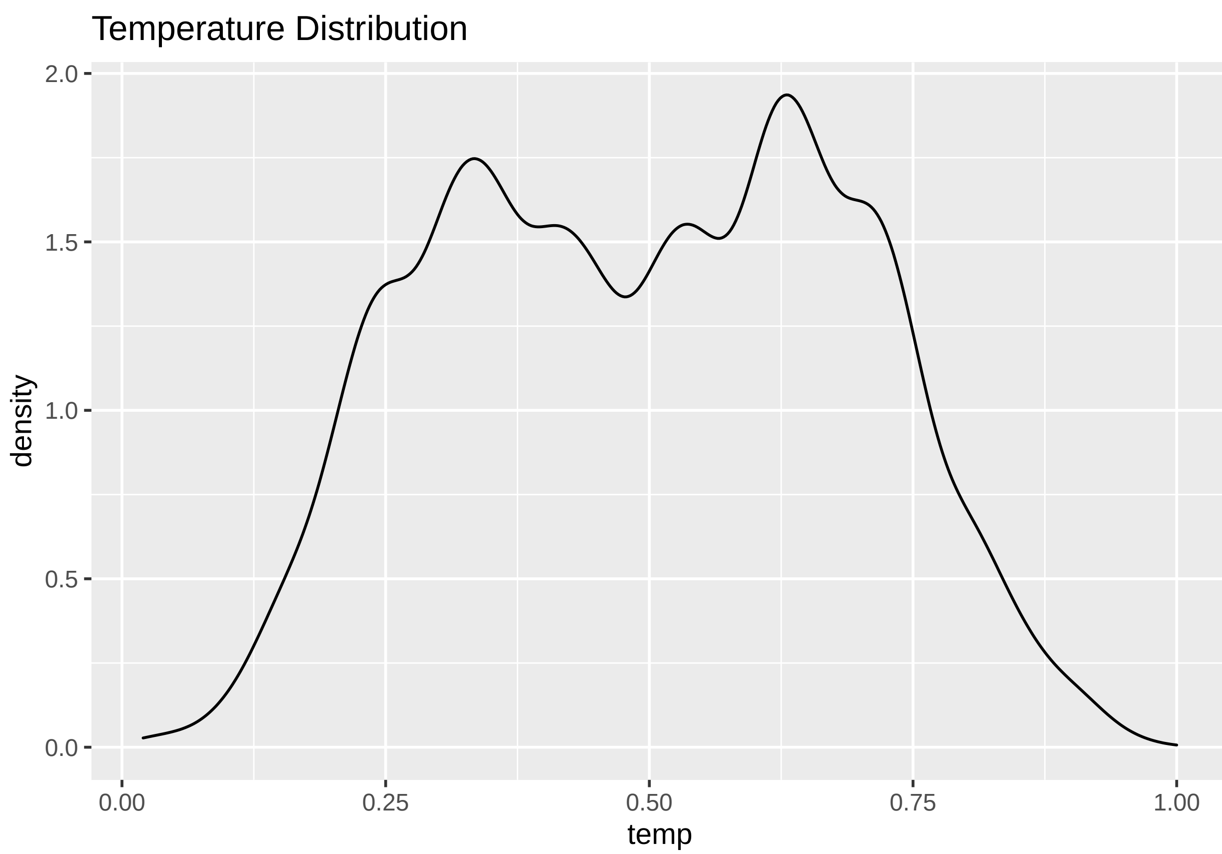
```
## [1] 0
```

Graphs on the data

```
ggplot(train, aes(x = cnt)) + geom_density() + ggtitle("Count of Bikes Rented Density Plot")
```




```
ggplot(train, aes(x = temp)) + geom_density() + ggtitle("Temperature Distribution")
```




Create base linear model

```r
lm1 <- lm(cnt~., data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = cnt ~ ., data = train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -4.641e-11 -1.300e-14  7.000e-15  2.800e-14  2.723e-11
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -2.706e-13  3.700e-14 -7.314e+00 2.73e-13 ***
## instant      3.722e-17  2.537e-17  1.467e+00 0.142424
## yr          -8.197e-13  2.222e-13 -3.689e+00 0.000226 ***
## mnth        -9.831e-14  1.851e-14 -5.310e+00 1.11e-07 ***
## hr           1.481e-14  8.603e-16  1.721e+01  < 2e-16 ***
## holiday      8.127e-14  3.316e-14  2.451e+00 0.014271 *
## weekday      1.915e-16  2.676e-15  7.200e-02 0.942948
## workingday  -4.018e-13  1.353e-14 -2.969e+01  < 2e-16 ***
## temp         1.809e-12  1.809e-13  1.000e+01  < 2e-16 ***
## atemp        9.408e-13  2.033e-13  4.628e+00 3.72e-06 ***
## hum         -6.313e-13  3.205e-14 -1.970e+01  < 2e-16 ***
```

```
## windspeed    7.446e-14  4.737e-14  1.572e+00 0.115982
## casual       1.000e+00  1.586e-16  6.304e+15  < 2e-16 ***
## registered   1.000e+00  4.646e-17  2.152e+16  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.281e-13 on 13889 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 8.99e+31 on 13 and 13889 DF,  p-value: < 2.2e-16
```

```
pred <- predict(lm1, newdata=test)
cor_lm <- cor(pred, test$cnt)
mse_lm <- mean((pred - test$cnt)^2)
rmse_lm <- sqrt(mean((pred-test$cnt)^2))
print(paste("cor=", cor_lm))
```

```
## [1] "cor= 1"
```

```
print(paste("mse=", mse_lm))
```

```
## [1] "mse= 4.07687806854585e-25"
```

The R squared value measures the relationship between your linear model and target variable. The results show the linear model has a perfect fit with an extremely low mean squared error. The independent variables overall have very low p-values as well.

## kNN Regression

Scale data and run regression

```
train_scaled <- train[,1:14]
means <- sapply(train_scaled, mean)
stdvs <- sapply(train_scaled, sd)
train_scaled <- scale(train_scaled, center=means, scale=stdvs)
test_scaled <- scale(test[,1:14], center=means, scale=stdvs)
```

Run kNN Regression

```
library(caret)
```

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

```
fit <- knnreg(train_scaled, train$cnt, k=3)
pred2 <- predict(fit, test_scaled)
cor_knn2 <- cor(pred2, test$cnt)
mse_knn2 <- mean((pred2 - test$cnt)^2)
print(paste("cor=", cor_knn2))
```

```
## [1] "cor= 0.989948580342323"
```

```r
print(paste("mse=", mse_knn2))
```

```
## [1] "mse= 642.832951748498"
```

```r
print(paste("rmse=", sqrt(mse_knn2)))
```

```
## [1] "rmse= 25.3541505822715"
```

Find the best k value for the model

```r
cor_k <- rep(0, 20)
mse_k <- rep(0, 20)
i <- 1
for (k in seq(1, 39, 2)){
  fit_k <- knnreg(train_scaled,train$cnt, k=k)
  pred_k <- predict(fit_k, test_scaled)
  cor_k[i] <- cor(pred_k, test$cnt)
  mse_k[i] <- mean((pred_k - test$cnt)^2)
  print(paste("k=", k, cor_k[i], mse_k[i]))
  i <- i + 1
}
```

```
## [1] "k= 1 0.98306775875943 1086.10241657077"
## [1] "k= 3 0.989948580342323 642.832951748498"
## [1] "k= 5 0.991008627276036 577.514972509909"
## [1] "k= 7 0.991078196656088 578.53195105358"
## [1] "k= 9 0.990837665291767 598.167699924704"
## [1] "k= 11 0.990687045758844 610.420692204237"
## [1] "k= 13 0.990536133453734 622.098251345261"
## [1] "k= 15 0.990317061453034 640.608916618719"
## [1] "k= 17 0.990101790540697 657.122606021795"
## [1] "k= 19 0.989826628747146 677.897159308069"
## [1] "k= 21 0.98953047445573 701.405957997914"
## [1] "k= 23 0.989391229072734 714.556336770811"
## [1] "k= 25 0.989123702547015 736.736215366912"
## [1] "k= 27 0.988916907999926 756.562705016046"
## [1] "k= 29 0.988689256183867 775.086251720459"
## [1] "k= 31 0.988474479915493 791.995322131275"
## [1] "k= 33 0.988231318992835 811.468623235692"
## [1] "k= 35 0.987965197510416 831.87150405248"
## [1] "k= 37 0.987713085061437 852.550849626071"
## [1] "k= 39 0.987527413132629 868.009269547041"
```

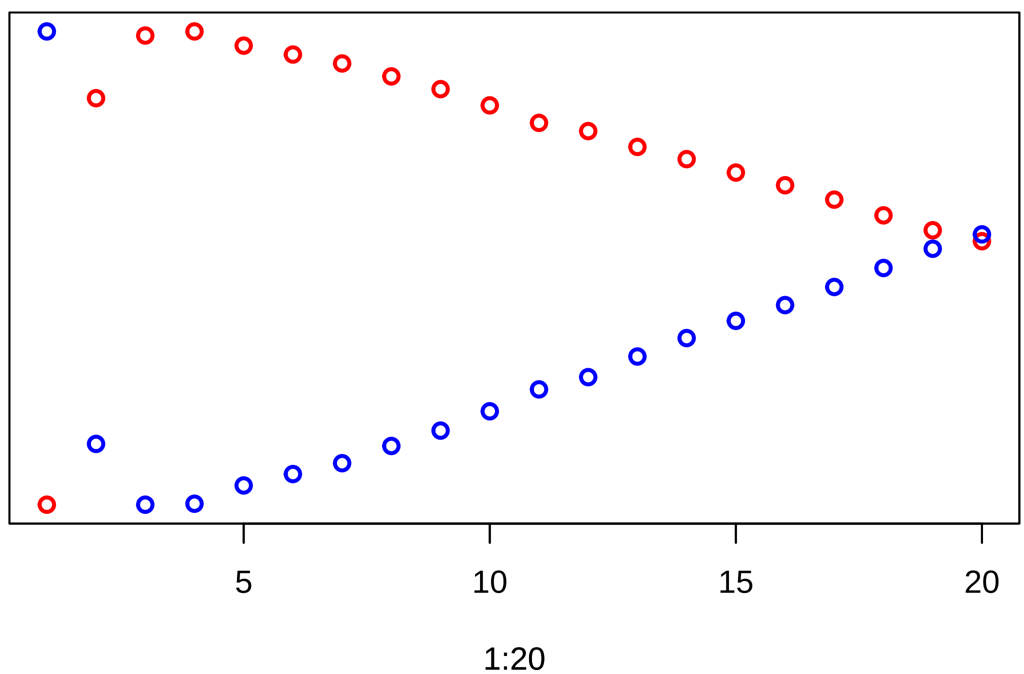
```r
plot(1:20, cor_k, lwd=2, col='red', ylab="", yaxt='n')
par(new=TRUE)
plot(1:20, mse_k, lwd=2, col='blue', labels=FALSE, ylab="", yaxt='n')
```

```
## Warning in plot.window(...): "labels" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "labels" is not a graphical parameter
```

```
## Warning in box(...): "labels" is not a graphical parameter

## Warning in title(...): "labels" is not a graphical parameter
```




Find when correlation is the highest and mse is the lowest.

```
which.min(mse_k)
```

```
## [1] 3
```

```
which.max(cor_k)
```

```
## [1] 4
```

kNN with k = 4

```
fit <- knnreg(train_scaled, train$cnt, k=4)
pred3 <- predict(fit, test_scaled)
cor_knn3 <- cor(pred3, test$cnt)
mse_knn3 <- mean((pred3 - test$cnt)^2)
print(paste("cor=", cor_knn3))
```

```
## [1] "cor= 0.990648451008266"
```

```r
print(paste("mse=", mse_knn3))
```

```
## [1] "mse= 599.040376150748"
```

The results for kNN regression has a lower correlation than the linear model. As well as, it has a much higher mean squared error than the previous linear model. The linear model out performed the bike rental data set. ## Decisions Trees Using Regression

```r
tree1 <- tree(cnt~., data=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = cnt ~ ., data = train)
## Variables actually used in tree construction:
## [1] "registered" "casual"
## Number of terminal nodes:  9
## Residual mean deviance:  1555 = 21600000 / 13890
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -149.20  -18.48   -4.03    0.00   17.37  230.80
```

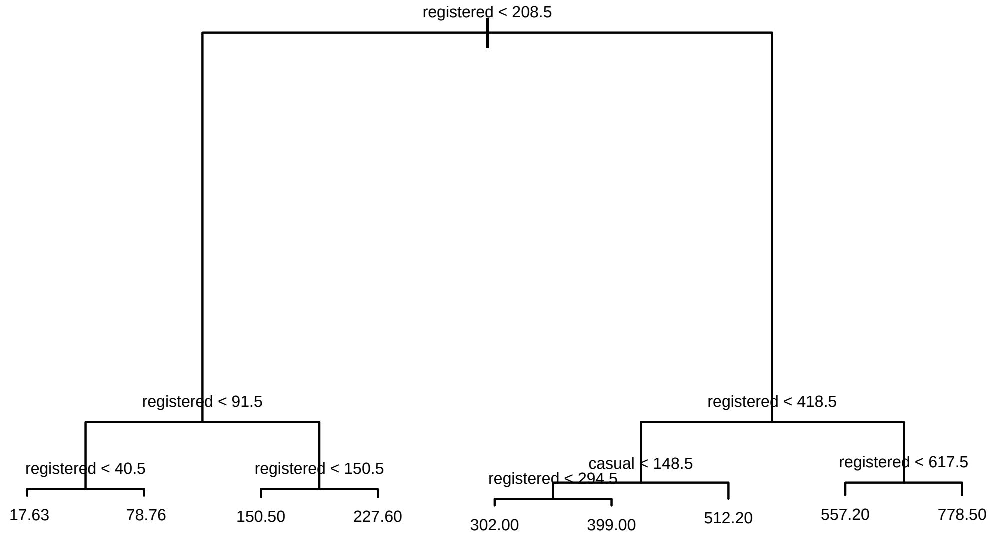
```r
pred <- predict(tree1, newdata=test)
print(paste('correlation:', cor(pred, test$cnt)))
```

```
## [1] "correlation: 0.97454610774393"
```

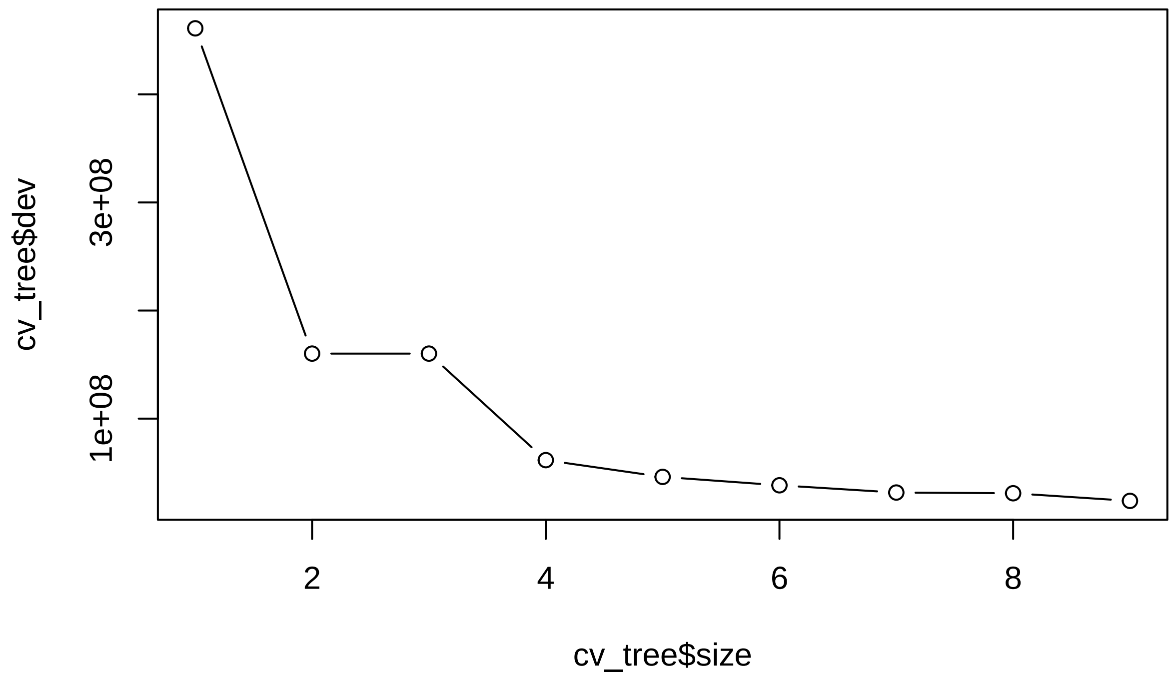
```r
rmse_tree <- sqrt(mean((pred-test$cnt)^2))
print(paste('rmse:', rmse_tree))
```

```
## [1] "rmse: 40.0014591375194"
```

```r
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```

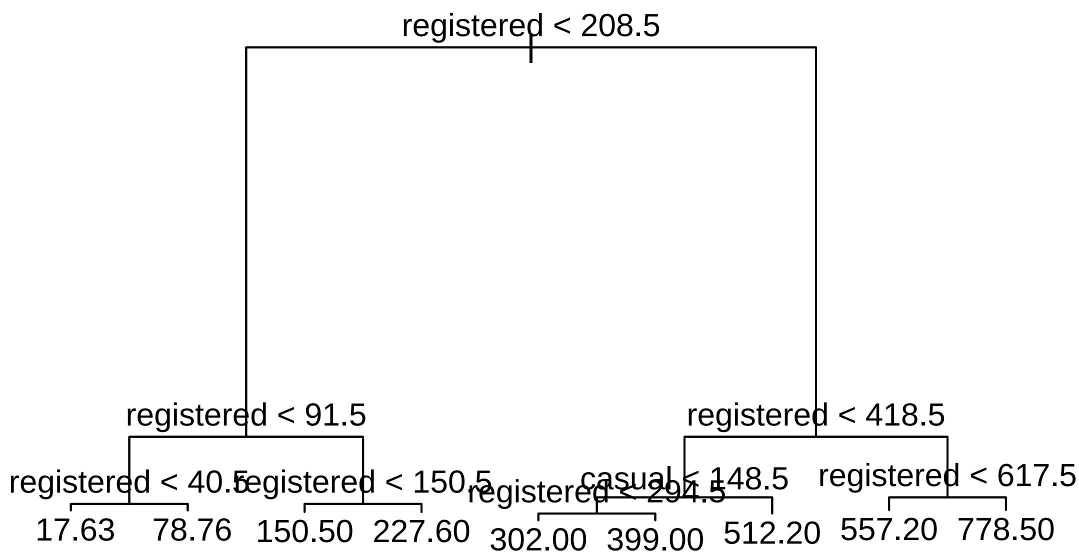



```
cv_tree <- cv.tree(tree1)
plot(cv_tree$size, cv_tree$dev, type='b')
```




The results show that that the best size of the tree is 9. Proceed with pruning.

```
tree_pruned <- prune.tree(tree1, best=9)
plot(tree_pruned)
text(tree_pruned, pretty=0)
```




Pruning Results

```
pred_pruned <- predict(tree_pruned, newdata=test)
cor_pruned <- cor(pred_pruned, test$cnt)
rmse_pruned <- rmse_pruned <- sqrt(mean((pred_pruned-test$cnt)^2))
print(paste("correlation:", cor_pruned))
```

```
## [1] "correlation: 0.97454610774393"
```

```
print(paste("rmse:", rmse_pruned))
```

```
## [1] "rmse: 40.0014591375194"
```

The results for the decision tree after pruning have a lower correlation than the linear model and kNN regression. The mean squared error for the decision tree is lower than the result of mse for the kNN regression.

Overall, the linear model out-performs both the kNN regression and decision tree. This results goes to show that sometimes more complex models are not always the right solution for data analysis and exploration. When more basic models perform well, staticians use less resources to perform analysis.