Chao and Albert Applied Project

Albert Mandizha, Chao Yang

2022-11-30

# Multiple Linear Regression

# 1. Clean Environment & Load Libraries

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(caTools)  
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

# 2. Source, Collect and Load Data

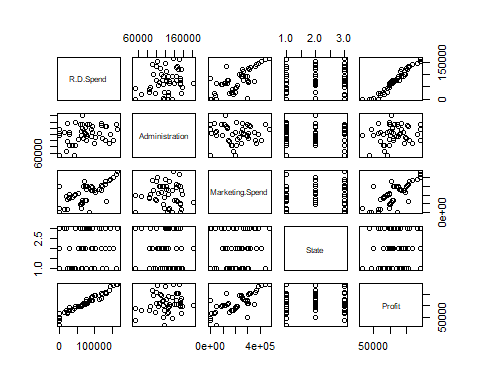
## 2.1 Source and Collect Data

Our data is collected online based on existed numbers.

## 2.2 Load Data

Importing the dataset

dataset = read.csv("50\_Startups.csv")  
plot(dataset)



# 3. Data preprocessing and Data cleaning

## 3.1 Explore missing value patterns

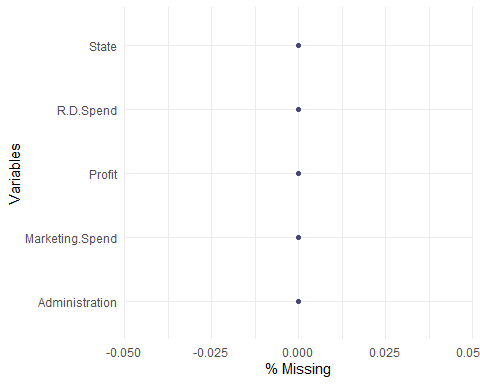
## 3.2 Check if there are NA values

NANumbers <- sum(is.na(dataset))   
paste("\*\*\*\* The number of NA values in this data set =", NANumbers)

## [1] "\*\*\*\* The number of NA values in this data set = 0"

# Plotting percentage of missing values per feature  
library(naniar)  
gg\_miss\_var(dataset, show\_pct = TRUE)

## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please  
## use `guide = "none"` instead.



We don’t have missing values but we drop na values just to check

# 3.3 Data parsing (Set dummy variables)

We don’t have dummy variables to set in our dataset

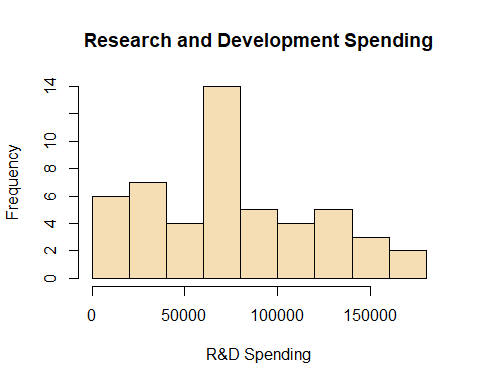
## 3.3.1 Encoding categorical data

dataset$State = factor(dataset$State,  
 levels = c('New York', 'California', 'Florida'),  
 labels = c(1, 2, 3))  
dataset$State=as.integer(dataset$State)  
# Display new data frame  
head(dataset)

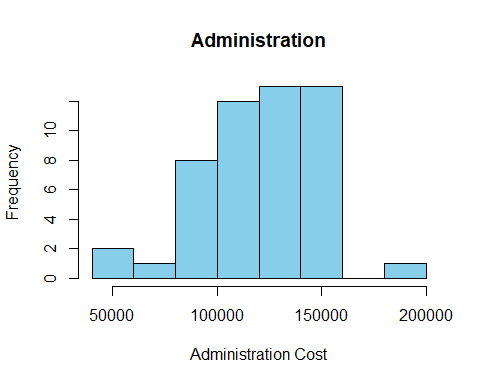
## R.D.Spend Administration Marketing.Spend State Profit  
## 1 165349.2 136897.80 471784.1 1 192261.8  
## 2 162597.7 151377.59 443898.5 2 191792.1  
## 3 153441.5 101145.55 407934.5 3 191050.4  
## 4 144372.4 118671.85 383199.6 1 182902.0  
## 5 142107.3 91391.77 366168.4 3 166187.9  
## 6 131876.9 99814.71 362861.4 1 156991.1

## 3.4 Data histograms

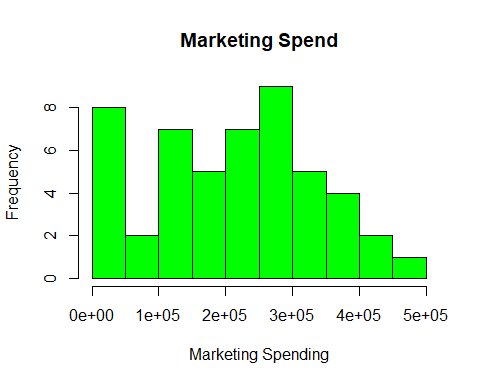
par(mfrow=c(1,1))  
hist(dataset$R.D.Spend, col = "wheat",main = "Research and Development Spending",xlab = "R&D Spending")



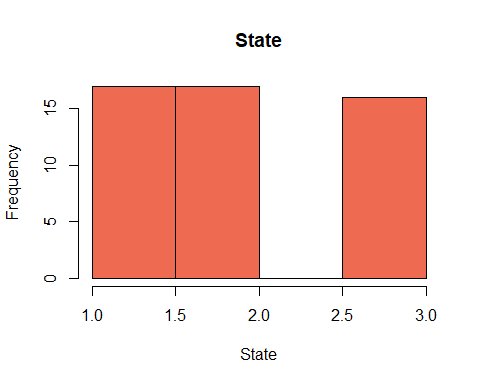
hist(dataset$Administration, col = "skyblue",main = "Administration",xlab = "Administration Cost")



hist(dataset$Marketing.Spend, col = "green", main = "Marketing Spend",xlab = "Marketing Spending")

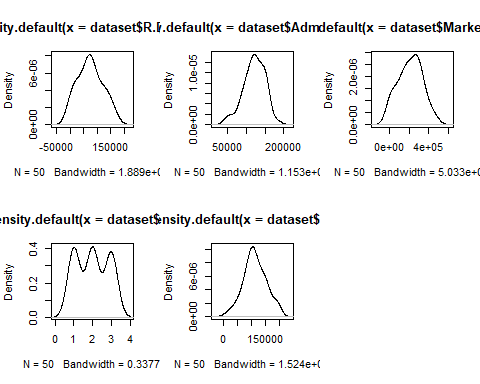


hist(dataset$State, col = "coral2",main = "State",xlab = "State")

 ## 3.4.1 Kernel Density Plot

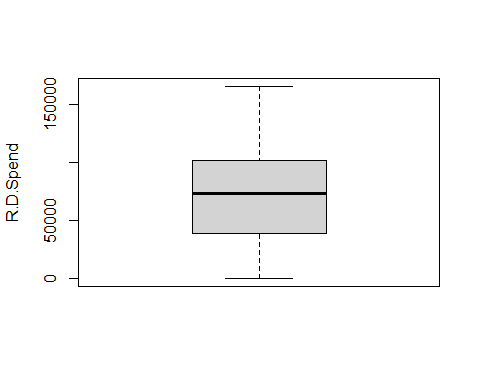
Density curves allow us to quickly see whether or not a graph is left skewed, right skewed, or has no skew. We see there are no features that their density have no skew.

library(ggplot2)  
par(mfrow=c(2,3))  
plot(density(dataset$R.D.Spend))  
plot(density(dataset$Administration))  
plot(density(dataset$Marketing.Spend))  
plot(density(dataset$State))  
plot(density(dataset$Profit))

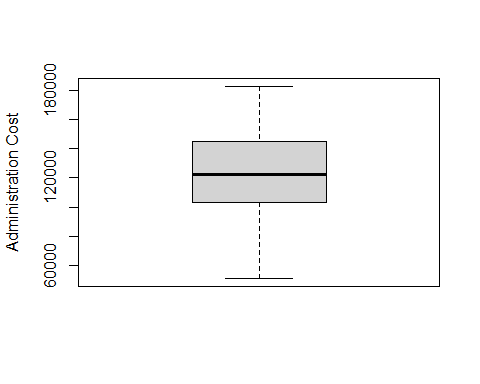


## 3.5 Checking & Treating Outliers

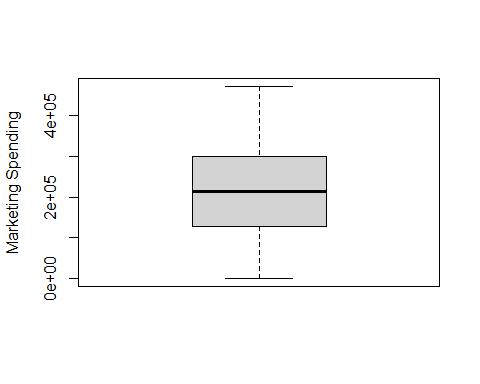
boxplot(dataset$R.D.Spend,  
 ylab = "R.D.Spend")



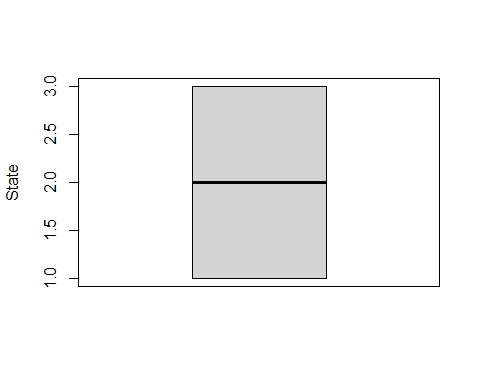
boxplot(dataset$Administration,  
 ylab = "Administration Cost")



boxplot(dataset$Marketing.Spend,  
 ylab = "Marketing Spending")



boxplot(dataset$State,  
 ylab = "State")



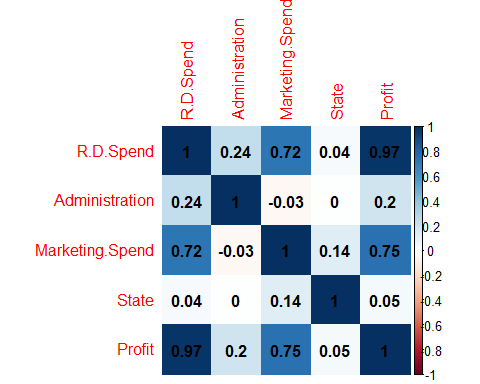
It can be noted that from the boxplots there are no outliers from the dataset.

## 3.6 Data Correlation Analysis

library(corrplot)

## corrplot 0.92 loaded

corrplot(cor(dataset), addCoef.col = 'black', method="color")



#4.0 Data Analysis Stage

## 4.1 Splitting the dataset into the Training set and Test set

library(caret)  
set.seed(123)  
training.samples=dataset$Profit%>%  
 createDataPartition(p=.8, list =FALSE)  
training\_set=dataset[training.samples,]  
test\_set=dataset[-training.samples,]

## 4.2 Model Fitting Multiple Linear Regression to the Training set

library(Metrics)

##   
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':  
##   
## precision, recall

regressor = lm(formula = Profit ~ .,  
 data = training\_set)  
summary(regressor)

##   
## Call:  
## lm(formula = Profit ~ ., data = training\_set)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30896.3 -4638.4 758.2 6770.3 16049.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.797e+04 8.090e+03 5.930 7.82e-07 \*\*\*  
## R.D.Spend 8.292e-01 4.887e-02 16.967 < 2e-16 \*\*\*  
## Administration -4.863e-02 5.667e-02 -0.858 0.396   
## Marketing.Spend 3.115e-02 1.860e-02 1.675 0.102   
## State 9.455e+02 1.853e+03 0.510 0.613   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9500 on 37 degrees of freedom  
## Multiple R-squared: 0.9504, Adjusted R-squared: 0.945   
## F-statistic: 177.1 on 4 and 37 DF, p-value: < 2.2e-16

predy = predict(regressor, newdata = test\_set)  
mse <- mse(training\_set$Profit,predy)

## Warning in actual - predicted: longer object length is not a multiple of shorter  
## object length

cat("MSE" , mse, "\n")

## MSE 3592993407

coef(regressor)

## (Intercept) R.D.Spend Administration Marketing.Spend State   
## 4.796872e+04 8.291682e-01 -4.863310e-02 3.115408e-02 9.454625e+02

confint(regressor)

## 2.5 % 97.5 %  
## (Intercept) 3.157735e+04 6.436008e+04  
## R.D.Spend 7.301466e-01 9.281898e-01  
## Administration -1.634631e-01 6.619686e-02  
## Marketing.Spend -6.527932e-03 6.883608e-02  
## State -2.808957e+03 4.699882e+03

## 4.3 Hypothesis test on coefficients of alpha value is setting as 0.05

#H0: Bi = 0, i =0,1,2,3,4   
#Ha: Bi != 0  
#Reject Ha fs F0 > F(0.05,4,37)  
F\_critical = qf(p=.05, df1=4, df=37, lower.tail=FALSE)  
F\_statistic = 177.1  
F\_critical < F\_statistic

## [1] TRUE

#Hence we reject H0, which means there are relationships between independent variables and dependent variables.

## 4.4 Preparing X and Y vectors for lasso regression

x\_train <- model.matrix(training\_set$Profit~., data = training\_set)[, -1]  
y\_train <- training\_set$Profit

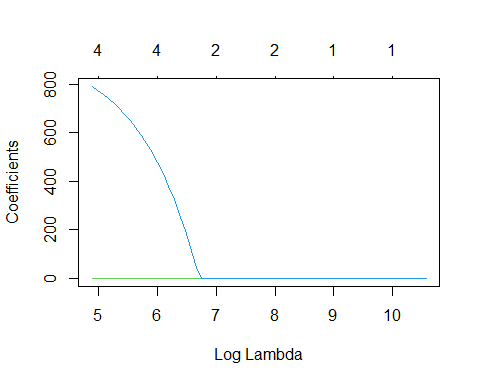
## 4.5 Displaying how the coefficients vary with lambda

library(glmnet)

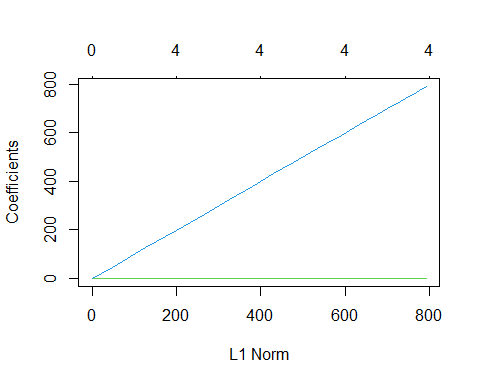
## Loading required package: Matrix

## Loaded glmnet 4.1-6

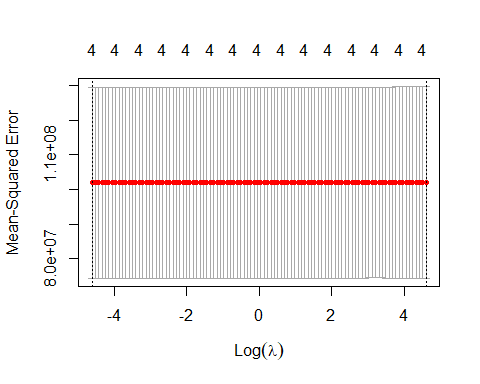
lasso\_model <- glmnet(x\_train, y\_train, alpha = 1)  
plot(lasso\_model, "lambda")



plot(lasso\_model, "norm")



set.seed(4)  
grid <- 10^seq(2, -2, length = 100)  
lasso\_cv\_model <- cv.glmnet(x\_train, y\_train, alpha = 1, lambda = grid)  
plot(lasso\_cv\_model)



best\_lambda <- lasso\_cv\_model$lambda.min  
cat("Best Lambda: ", best\_lambda, "\n")

## Best Lambda: 0.01

coef(lasso\_cv\_model)

## 5 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 4.783428e+04  
## R.D.Spend 8.259521e-01  
## Administration -4.375055e-02  
## Marketing.Spend 3.122857e-02  
## State 8.307380e+02

## 4.6 Performance on test data

library(Metrics)  
x\_test <- model.matrix(test\_set$Profit~. , data = test\_set)[, -1]  
lasso\_pred <- predict(lasso\_cv\_model, s = best\_lambda, newx = x\_test)  
test\_mse <- mse(test\_set$Profit, lasso\_pred)  
cat("Test MSE" , test\_mse, "\n")

## Test MSE 90907605

The resulting model have coefficients for all predictor variables.All the coefficients for other predictors got smaller with the except Intercept. This was expected as lasso performs feature selection through shrinking irrelevant coefficients to 0

# 5.0 Model Accuracy

## 5.1. Make predictions

predictions <- regressor %>% predict(test\_set)

## 5.2 Model performance

### 5.2.1 (a) Compute the prediction error, RMSE

RMSE(predictions, test\_set$Profit)

## [1] 9534.568

### 5.2.2 (b) Compute R-square

R2(predictions, test\_set$Profit)

## [1] 0.9801895

## 5.3 Conclusion

From the output above, the R2 is 0.98, meaning that the observed and the predicted outcome values are highly correlated, which is very good.

Error\_rate= 9534.568/mean(test\_set$Profit)  
Error\_rate

## [1] 0.08379864

The prediction error RMSE is 9534.568, representing an error rate of 9534.568/mean(test\_set$Profit) = 8.3%, which is good.