# Linear Regression

Code

## Import Library

Hide

```
library(Metrics)
library(DAAG)
```

## Get the data set

Hide

```
data_set <- read.csv("Advertising.csv", header=TRUE)
```

## Summary of the data set

Hide

```
summary(data_set)
```

```
       X                TV              Radio          Newspaper          Sales      
 Min.   :  1.00   Min.   :  0.70   Min.   : 0.000   Min.   :  0.30   Min.   : 1.60  
 1st Qu.: 50.75   1st Qu.: 74.38   1st Qu.: 9.975   1st Qu.: 12.75   1st Qu.:10.38  
 Median :100.50   Median :149.75   Median :22.900   Median : 25.75   Median :12.90  
 Mean   :100.50   Mean   :147.04   Mean   :23.264   Mean   : 30.55   Mean   :14.02  
 3rd Qu.:150.25   3rd Qu.:218.82   3rd Qu.:36.525   3rd Qu.: 45.10   3rd Qu.:17.40  
 Max.   :200.00   Max.   :296.40   Max.   :49.600   Max.   :114.00   Max.   :27.00  
```

Hide

```
print(nrow(data_set))
```

```
[1] 200
```

Hide

```
print(ncol(data_set))
```

```
[1] 5
```

## Drop index column

Hide

```
data_set <- data_set[c(2:5)]

colnames(data_set)
```

```
[1] "TV"        "Radio"     "Newspaper" "Sales"
```

# SPLIT DATA SET INTO TRAINING AND TESTING

## 80% of the sample size

Hide

```
ind_split <- floor(0.80 * nrow(data_set))
```

## Set seed to make your partition reproducible

Hide

```
set.seed(123)
train_ind <- sample(seq_len(nrow(data_set)), size = ind_split)

train <- data_set[train_ind, ]
test <- data_set[-train_ind, ]

View(train)
View(test)
```

# LINEAR REGRESSION

Hide

```
lr_age <- lm(Sales~TV, data=train)
summary(lr_age)
```

```
Call:
lm(formula = Sales ~ TV, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-8.3396 -1.9922  0.0219  2.0201  7.2355

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.069052   0.507788   13.92   <2e-16 ***
TV          0.047237   0.002983   15.83   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.257 on 158 degrees of freedom
Multiple R-squared:  0.6134,    Adjusted R-squared:  0.611
F-statistic: 250.7 on 1 and 158 DF,  p-value: < 2.2e-16
```

Hide

```
lr_dist <- lm(Sales~Radio, data=train)
summary(lr_dist)
```

```
Call:
lm(formula = Sales ~ Radio, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-15.5706  -2.1136   0.8746   2.9348   8.3634

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.41474    0.64372  14.626  < 2e-16 ***
Radio        0.19586    0.02325   8.426 2.08e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.352 on 158 degrees of freedom
Multiple R-squared:   0.31, Adjusted R-squared:  0.3056
F-statistic: 70.99 on 1 and 158 DF,  p-value: 2.083e-14
```

Hide

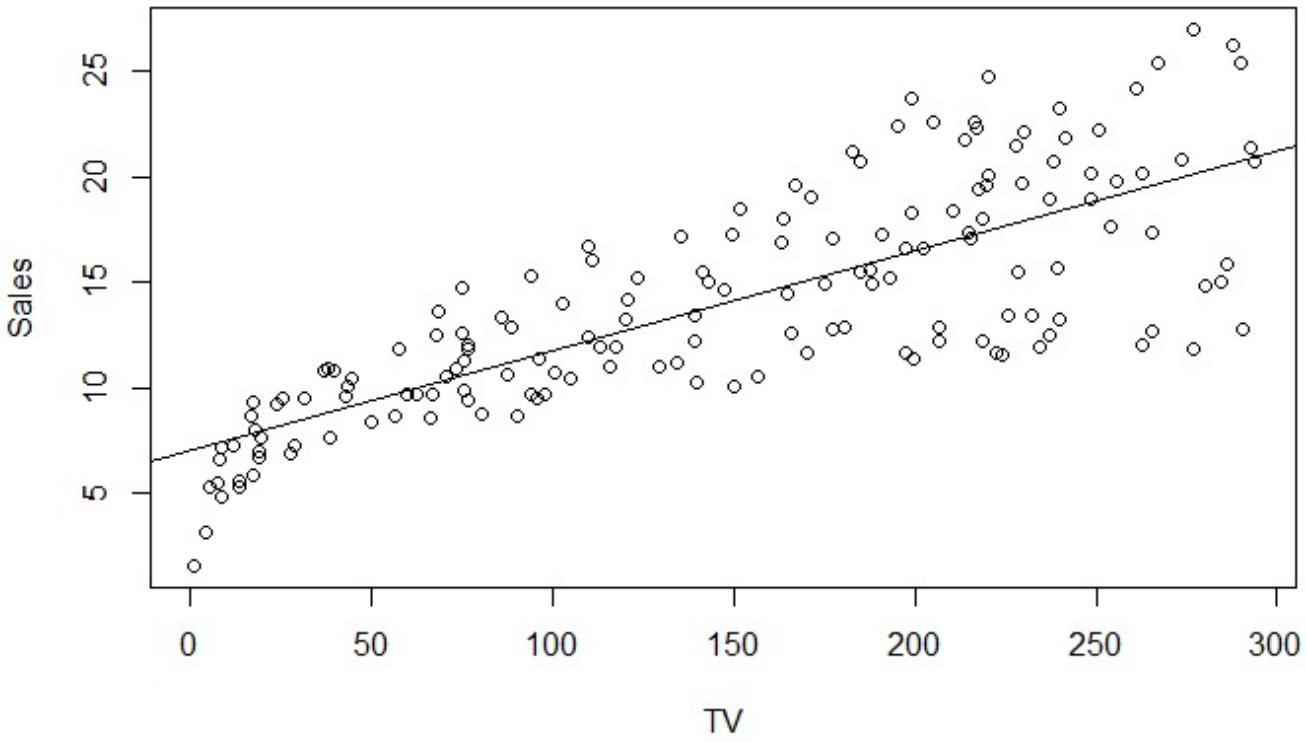
```
lr_conv <- lm(Sales~Newspaper, data=train)
summary(lr_conv)
```

```
Call:
lm(formula = Sales ~ Newspaper, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-11.728  -3.717  -1.147   3.767  12.643

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.05783    0.70700  18.469   <2e-16 ***
Newspaper    0.03109    0.01901   1.635    0.104
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.195 on 158 degrees of freedom
Multiple R-squared:  0.01664,   Adjusted R-squared:  0.01042
F-statistic: 2.674 on 1 and 158 DF,  p-value: 0.104
```

# PLOT GRAPH FOR RELATIONSHIP BETWEEN FEATURES AND TARGET

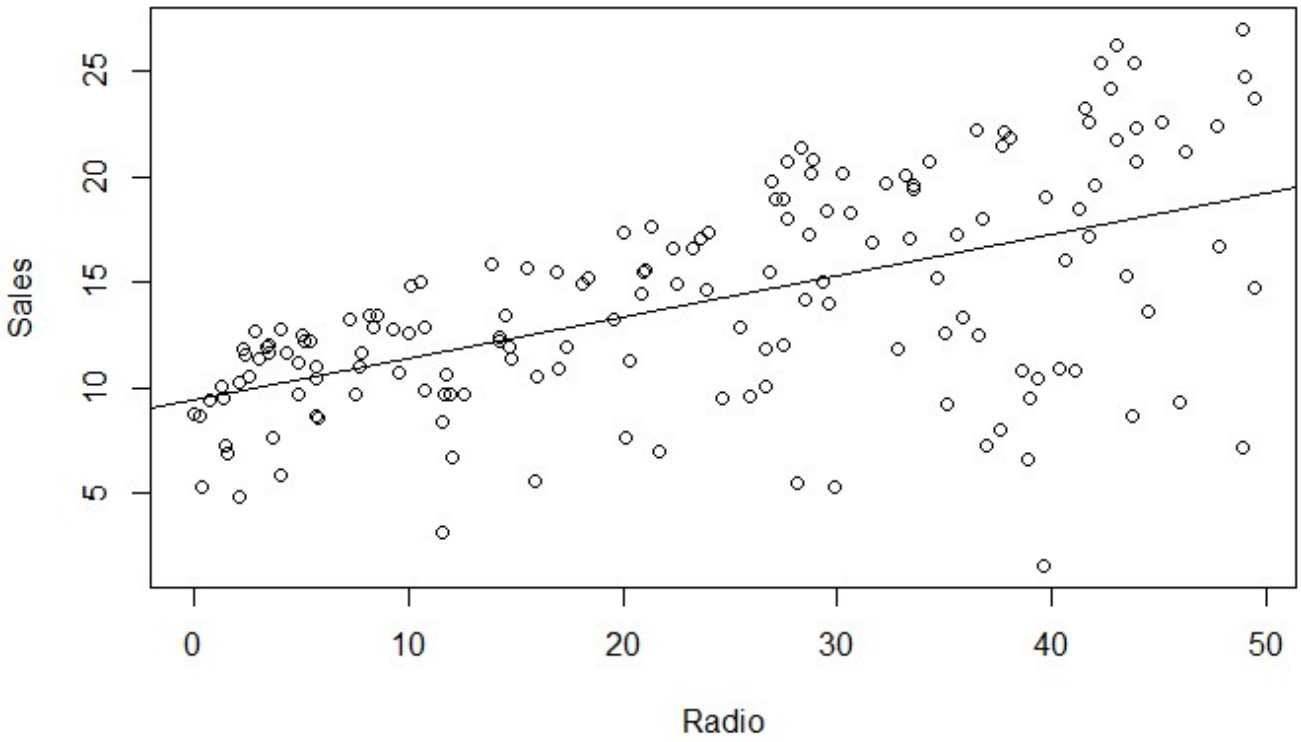
Hide

```
plot(train$Sales~train$TV, xlab="TV", ylab ="Sales")
abline(lr_age)
```



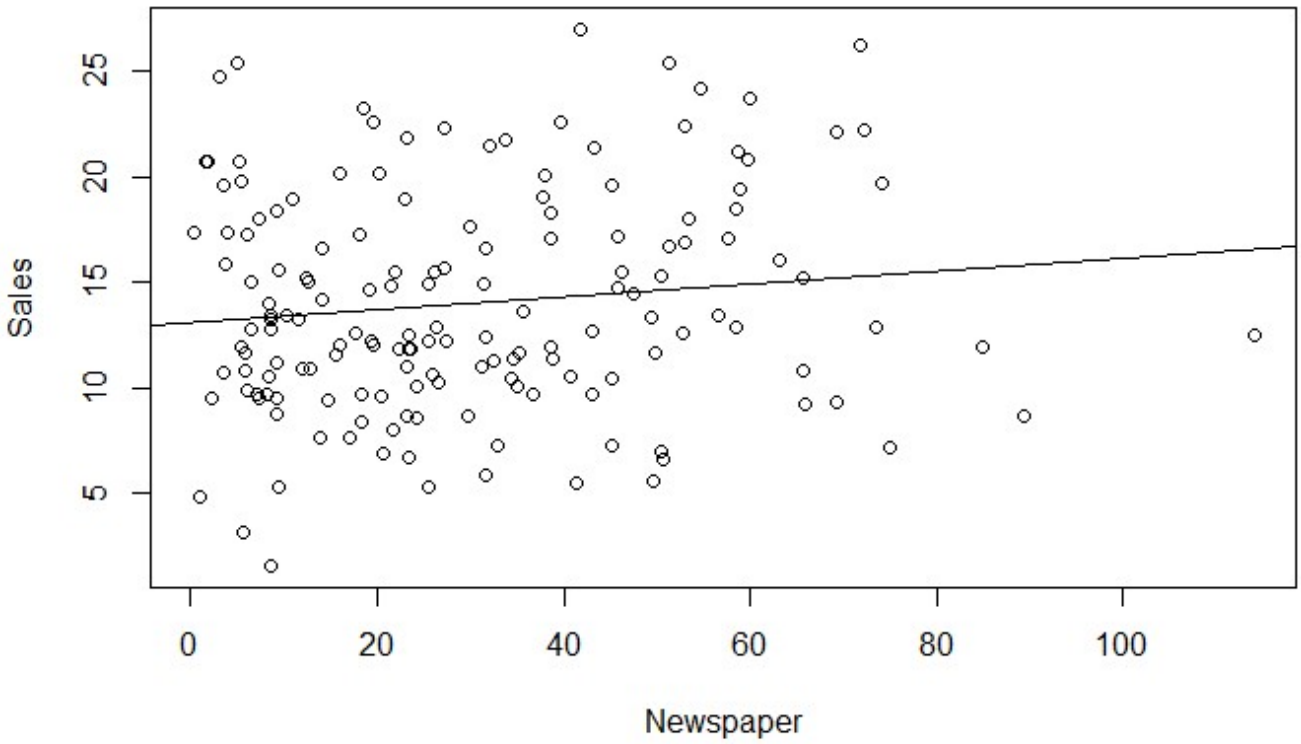

Hide

```
plot(train$Sales~train$Radio, xlab="Radio", ylab ="Sales")
abline(lr_dist)
```




Hide

```
plot(train$Sales~train$Newspaper, xlab="Newspaper", ylab ="Sales")
abline(lr_conv)
```




# Calculate MSE for distance

Hide

```
lr_dist_1 <- lm(Sales~log(TV), data=train)
View(lr_dist_1)
```

# Prediction on train

Hide

```
lr_train_1 <- predict(lr_dist_1, train)
lr_train_1
```

```
      159       179        14       195       170        50       118        43
198       194       153
 5.643334 17.687556 13.716094 15.348648 17.790723 12.282012 12.787577 17.913278 15.98
6456 15.760469 16.405635
       90        91       188       185        92       137        99        72
26         7       196
14.168448 14.935324 16.278285 17.358643  9.046485  8.624567 17.862363 14.168448 17.49
2768 11.705513 10.148466
      184       164        78        81       193       103       117        76
143        32       109
17.834663 15.684387 14.522498 12.787577  7.110416 17.735415 15.071766  7.043421 16.82
3131 14.274454  6.073663
      180       178        74        23       155        53       135       162
163        34        69
15.732978 15.837298 14.793811  6.102617 16.211962 16.751669 10.016637 13.224937 16.22
4107 17.531671 17.104306
      182       171        63       141        97       187        38        21
41       189        60
16.788439 11.173379 17.134657 12.635056 16.405635 15.079963 12.701900 16.786696 16.49
8899 17.813422 16.650037
       16       116        94         6        86       190        39       191
197       158         4
16.363007 12.722233 17.314887  4.515322 16.319896  7.428771 10.607975 10.275882 13.58
4996 15.351191 15.394156
       13       157       127        52        22        89       169       110
25        87        35
 8.346979 13.572851  4.099553 13.827689 17.104306 13.338731 16.734034 17.382570 12.01
0779 12.782590 13.645146
       40       112        30        12        31       132       121        64
183       192        93
16.950482 17.172652 12.486970 16.721641 17.904189 17.525932 15.128777 13.913927 11.61
8444 12.742458 16.774474
       96        71        67       177        79        85        37         8
51       165       166
15.679726 16.434429  9.414210 17.276759  2.699465 16.700300 17.550261 14.513007 16.44
7792 14.416774 17.057509
       98       173       140       200        84        46        17        62
122        54       124
16.152709  7.607743 16.152709 17.018341 12.366437 15.945364 12.332891 17.469525  7.44
9077 16.105051 14.603777
      108        24       125       167       147       168       181       128
88        27        42
13.428221 16.955489 16.975449  7.262299 17.147364 16.578902 15.520217 12.972393 14.19
9529 15.171648 15.986456
        5        70       145       129        83       149        55        36
139       186         9
16.067332 16.758700 13.664987 16.819676 12.732359 10.128479 17.489870 17.875483 10.59
9131 16.545616  4.471305
       48        56       111         1       123       172       154       131
126        77       101
```

```
17.144191 16.430602 16.913565 16.985390 16.883092 15.707603 15.861826 -5.079401 13.29
1002  8.897155 16.855799
      176        11        68        20       144         3        29       156
44       130        80
17.690307 12.236207 15.074500 15.287113 13.983723  7.110416 17.282885  1.650853 16.58
0742 11.842088 14.377589
       57       138       105       100         2       134
 3.847313 17.646051 17.117115 14.960754 10.729684 16.811025
```

# Prediction on test

Hide

```
lr_pred_1 <- predict(lr_dist_1, test)
lr_pred_1
```

```
       10        15        18        19        28        33        45        47
49        58        59
16.447792 16.528864 17.751686 12.410710 17.147364 13.704361  8.549467 13.398624 16.93
7099 14.988812 16.651843
       61        65        66        73        75        82        95       102
104       106       107
11.430985 14.843505 12.399690  8.798983 16.698517 17.142604 14.084302 17.949416 16.21
3989 15.036041  8.534268
      113       114       115       119       120       133       136       142
146       148       150
15.958389 16.630107 12.876240 14.683356  7.568692  4.381714 11.041674 16.329737 15.10
1735 17.196208 10.746758
      151       152       160       161       174       175       199
17.742203 14.538264 14.860891 15.888405 15.796817 16.855799 17.781337
```

# Calculate MSE

Hide

```
train_mse_1 = mse(train$Sales, lr_train_1)
print(train_mse_1)
```

```
[1] 11.48834
```

Hide

```
test_mse_1 = mse(test$Sales, lr_pred_1)
print(test_mse_1)
```

```
[1] 13.00607
```

# Graph of test vs train mse

Hide

```
plotter <- c(train_mse_1, test_mse_1)
barplot(plotter, width = 0.02, xlab="data", names.arg = c("Train MSE","Test MSE"), yl
ab="error", main="Error (TV Sales)")
```


# Correlation (Subset Selection Method)

Hide

```
res <- cor(train)
print(res)
```

```
                  TV      Radio   Newspaper    Sales
TV         1.00000000 0.04039595 -0.03283919 0.783218
Radio      0.04039595 1.00000000  0.33173584 0.556789
Newspaper -0.03283919 0.33173584  1.00000000 0.128995
Sales      0.78321799 0.55678901  0.12899496 1.000000
```

Hide

```
lin_gen <- lm(Sales~TV+Radio+Newspaper, data=train)
print(lin_gen)
```

```
Call:
lm(formula = Sales ~ TV + Radio + Newspaper, data = train)

Coefficients:
(Intercept)           TV        Radio    Newspaper
    3.04081      0.04589      0.18774     -0.00556
```

# Prediction on train

Hide

```
lin_train <- predict(lin_gen, train)
lin_train
```

```
      159       179        14       195       170        50       118        43
198       194       153
10.253887 16.038942  8.901963 16.560754 18.042100  8.102846  6.614804 21.704741 12.87
3914 18.560345 16.404208
       90        91       188       185        92       137        99        72
26         7       196
16.767666 10.072199 17.097449 18.519984  4.451435 11.485607 23.992065 10.588046 15.65
4275 11.706631  5.411751
      184       164        78        81       193       103       117        76
143        32       109
23.912583 17.411552 13.842219 11.435460  4.424171 17.676685 11.971161 11.523394 19.18
1939 11.273929  3.574752
      180       178        74        23       155        53       135       162
163        34        69
12.419903 12.120137  9.875212  6.355815 15.567585 20.580087 11.616083 13.420535 14.94
2405 18.982561 19.036959
      182       171        63       141        97       187        38        21
41       189        60
13.929482  7.410806 16.780705  9.529010 12.733187  9.688995 15.488950 17.966850 16.34
4604 18.754653 18.196585
       16       116        94         6        86       190        39       191
197       158         4
20.668843 12.764998 21.005313 12.203368 14.996068  6.040480  9.836120 12.537198  8.23
8632 10.024273 17.421577
       13       157       127        52        22        89       169       110
25        87        35
10.356165 15.235736 10.420361  9.430533 14.762188 11.472192 17.036110 19.780937  8.16
3559 11.616084  7.654286
       40       112        30        12        31       132       121        64
183       192        93
20.403735 21.137689  9.057669 17.377070 21.555095 15.516535 14.299702 13.264109  6.52
4869  8.499781 18.992459
       96        71        67       177        79        85        37         8
51       165       166
16.173191 17.707307  9.092445 19.996945  8.849654 20.723304 23.484198 12.172060 12.59
9496 11.148950 13.969155
       98       173       140       200        84        46        17        62
122        54       124
15.346233  7.619248 19.758237 15.258319 14.336082 15.125287 12.389571 22.744390  7.69
7225 19.767587 15.116731
      108        24       125       167       147       168       181       128
88        27        42
 7.116713 16.544847 19.224188 10.801032 15.381393 13.399484 10.669342  6.670135 15.39
1670 15.029266 17.218770
        5        70       145       129        83       149        55        36
139       186         9
13.040810 21.080400 10.017765 22.331899 10.126764 12.284259 20.414822 17.103847  9.76
2511 20.806418  3.824156
       48        56       111         1       123       172       154       131
126        77       101
```

```
21.738289 21.109140 14.628366 20.312067 13.684279 14.250067 18.145484 10.458892  9.11
3810  4.488110 13.777416
      176        11        68        20       144         3        29       156
44       130        80
24.695972  7.028543 12.098914 14.181279  8.719875 12.061925 19.418867  5.375004 13.96
5910  7.789136  9.681328
       57       138       105       100         2       134
 8.421019 20.694893 20.381963 16.818694 12.210241 19.166108
```

# Prediction on test

Hide

```
lin_pred <- predict(lin_gen, test)
lin_pred
```

```
       10        15        18        19        28        33        45        47
49        58        59
12.580125 18.327971 23.078708  9.963320 17.067163  7.616252  8.776754  8.817358 16.15
6097 12.803428 21.816770
       61        65        66        73        75        82        95       102
104       106       107
 5.752485 16.931561  7.948238 10.358667 17.379461 14.610094 10.537224 22.896816 14.79
3306 17.752128  6.088061
      113       114       115       119       120       133       136       142
146       148       150
13.981681 16.467475 15.223724 15.296465  6.810888  8.521026 14.033670 18.155479  9.78
6008 23.154312  9.821197
      151       152       160       161       174       175       199
18.326298  9.899876 12.346657 14.184381 12.030651 13.812488 23.572416
```

# Calculate MSE on subset

Hide

```
train_mse = mse(train$Sales, lin_train)
print(train_mse)
```

```
[1] 2.978008
```

Hide

```
test_mse = mse(test$Sales, lin_pred)
print(test_mse)
```

```
[1] 2.068075
```

# Graph of test vs train mse for subset

Hide

```
plotter <- c(train_mse, test_mse)
barplot(plotter, width = 0.02, xlab="data", names.arg = c("Train MSE","Test MSE"), yl
ab="error", main="Error (Subset)")
```


# K fold cross validation

Hide

```
model = cv.lm(df, (Sales~TV+Radio+Newspaper), m=5)
```