Linear Regression (Lab-11)

Code

Bibek Sapkota

Linear Regression

task 1:Loading the dataset

Hide

```
dataset = read.csv("data-marketing-budget-12mo.csv")
```

```
Warning message:
In file(con, "rb") :
  cannot open file 'C:/Users/sapko/AppData/Local/RStudio/notebooks/5119004B-LinearRegression/1/
s/chunks.json': No such file or directory
```

Hide

```
dataset
```

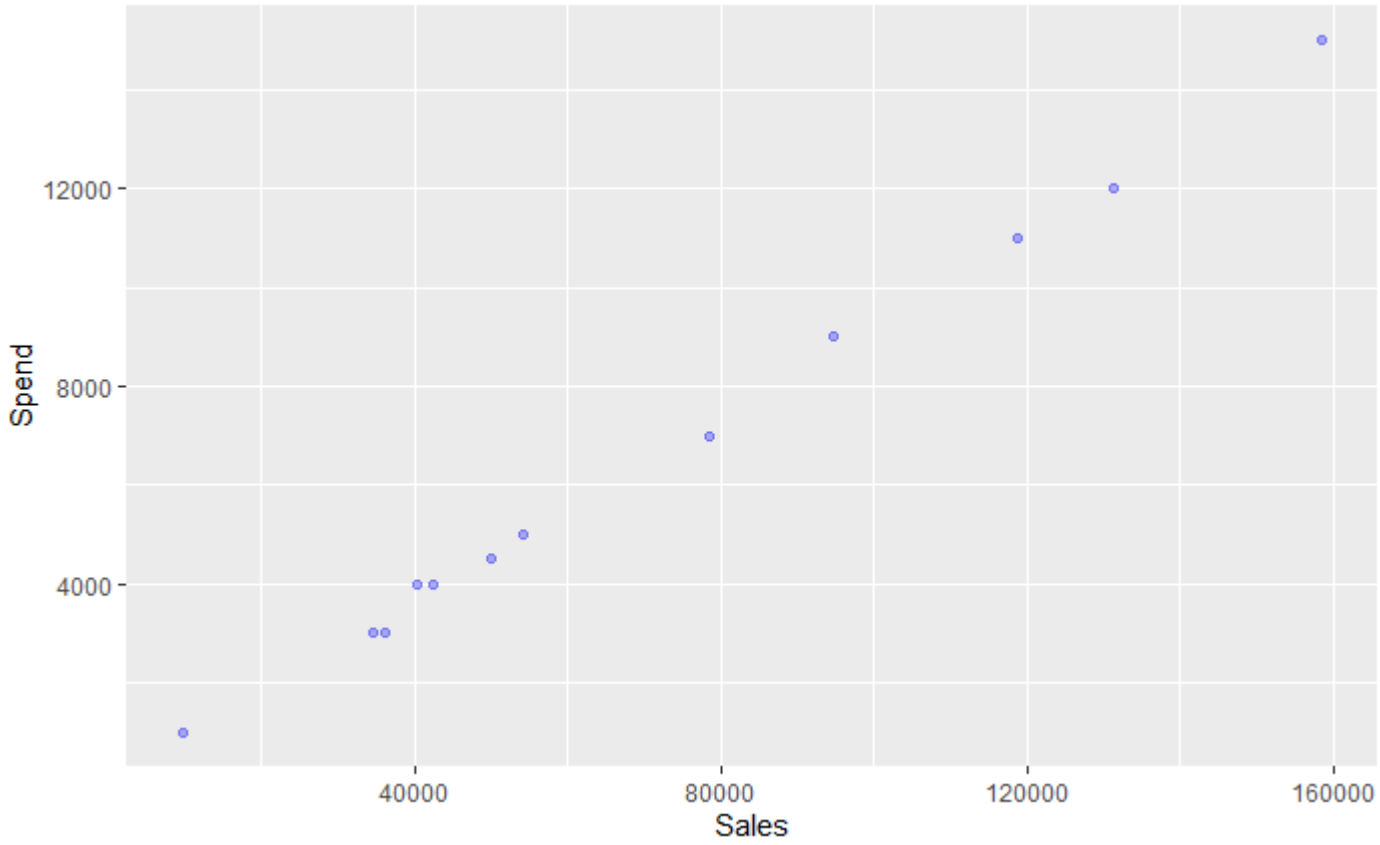
Month <int>	Spend <int>	Sales <int>
1	1000	9914
2	4000	40487
3	5000	54324
4	4500	50044
5	3000	34719
6	4000	42551
7	9000	94871
8	11000	118914
9	15000	158484
10	12000	131348

1-10 of 12 rows Previous **1** 2 Next

Use ggplot to plot a scatter plot between variables

Hide

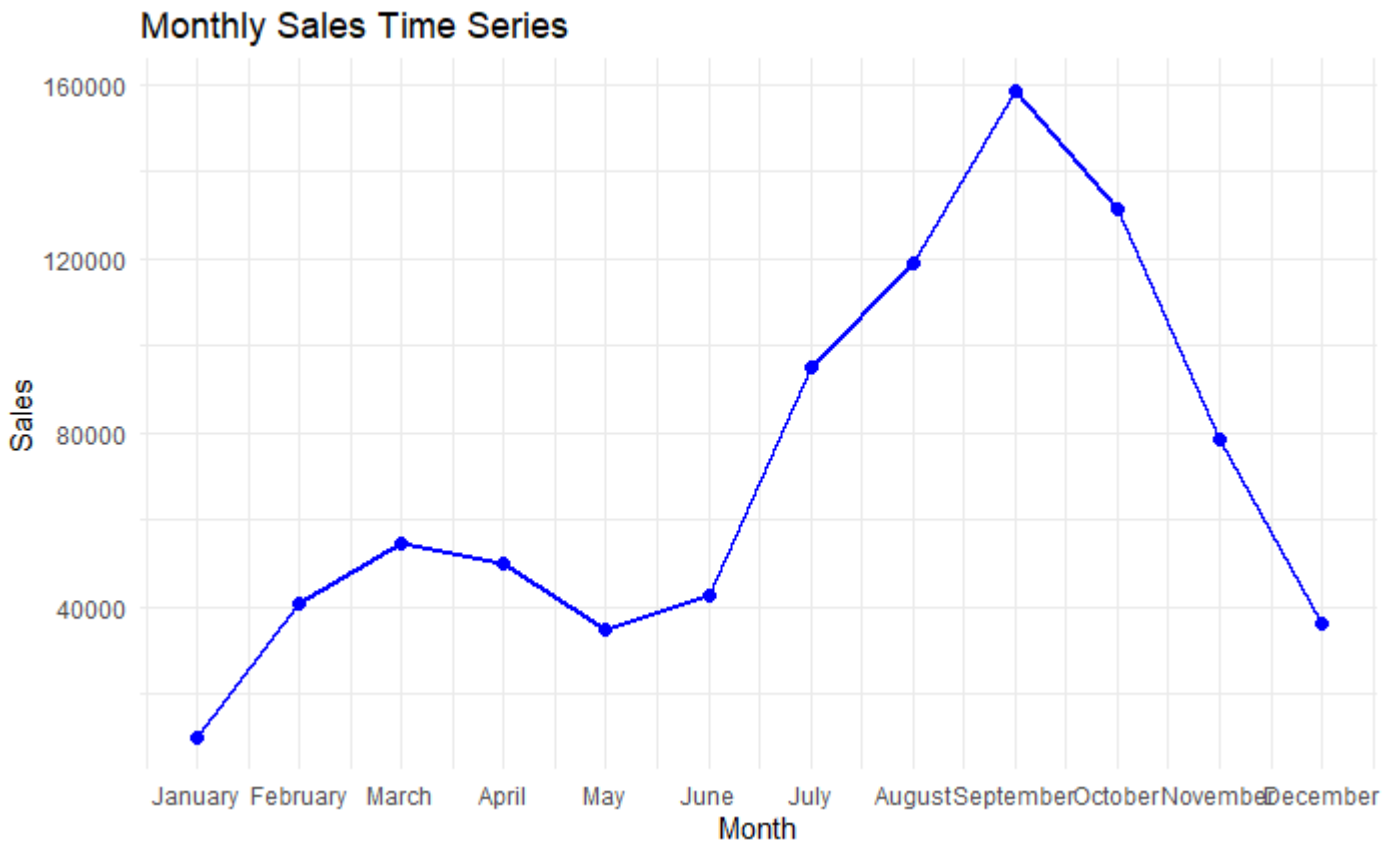
```
library(ggplot2)

ggplot(data = dataset, aes(x = Sales, y = Spend)) + geom_point(alpha= 0.3, color= "blue")
```



Q1:Write a command to plot sales for each month?

Hide

```
ggplot(data = dataset, aes(x = Month, y = Sales)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "blue", size = 2) +
  scale_x_continuous(breaks = 1:12, labels = month.name) +
  labs(title = "Monthly Sales Time Series", x = "Month", y = "Sales") +
  theme_minimal()
```



Simple (One Variable) and Multiple Linear Regression

Using lm()

One variable:

Hide

```
simple.fit = lm(Sales~Spend, data=dataset)
summary(simple.fit)
```

```
Call:
lm(formula = Sales ~ Spend, data = dataset)

Residuals:
   Min     1Q Median     3Q    Max
 -3385  -2097    258   1726   3034

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1383.4714  1255.2404   1.102    0.296
Spend         10.6222     0.1625  65.378 1.71e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2313 on 10 degrees of freedom
Multiple R-squared:  0.9977,    Adjusted R-squared:  0.9974
F-statistic:  4274 on 1 and 10 DF,  p-value: 1.707e-14
```

Multiple variables:

Hide

```
multi.fit = lm(Sales~Spend+Month, data=dataset)
summary(multi.fit)
```

```
Call:
lm(formula = Sales ~ Spend + Month, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-1793.73 -1558.33    -1.73  1374.19  1911.58

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -567.6098  1041.8836  -0.545  0.59913
Spend         10.3825     0.1328  78.159 4.65e-14 ***
Month        541.3736   158.1660   3.423  0.00759 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1607 on 9 degrees of freedom
Multiple R-squared:  0.999, Adjusted R-squared:  0.9988
F-statistic:  4433 on 2 and 9 DF,  p-value: 3.368e-14
```

Interpreting R’s Regression Output

task 1: Display each: # capture model summary as an object

Hide

```
modelSummary <- summary(simple.fit)
```

task 2: model coefficients

Hide

```
modelCoeffs <- modelSummary$coefficients
```

task 3: get beta estimate for Spend - 10.6222

Hide

```
beta.estimate <- modelCoeffs["Spend", "Estimate"]
```

task 4: get std.error for Spend - 0.1624745

Hide

```
std.error <- modelCoeffs["Spend", "Std. Error"]
```

task 6: get t value for Spend - 65.37761

Hide

```
t_value <- modelCoeffs["Spend", "t value"]
```

task 7: get model F-statistic - 4274 1 10

Hide

```
f <- modelSummary$fstatistic
f_statistic <- modelSummary$fstatistic[1]
```

task 8: get model p-value - 1.707e-14

Hide

```
model_p <- pf(f[1], f[2], f[3], lower=FALSE)
```

task 9:get model R-squared - 0.9976659

Hide

```
r_2 <- modelSummary$r.squared
```

Q2: Based on residual, which model is better? Why?

Ans- Based on residual Multiple Regression Output is better beacuse it has less error.

Q3: Try to write the multiple regression equation based on the numbers in the output (round to 1 decimal place).

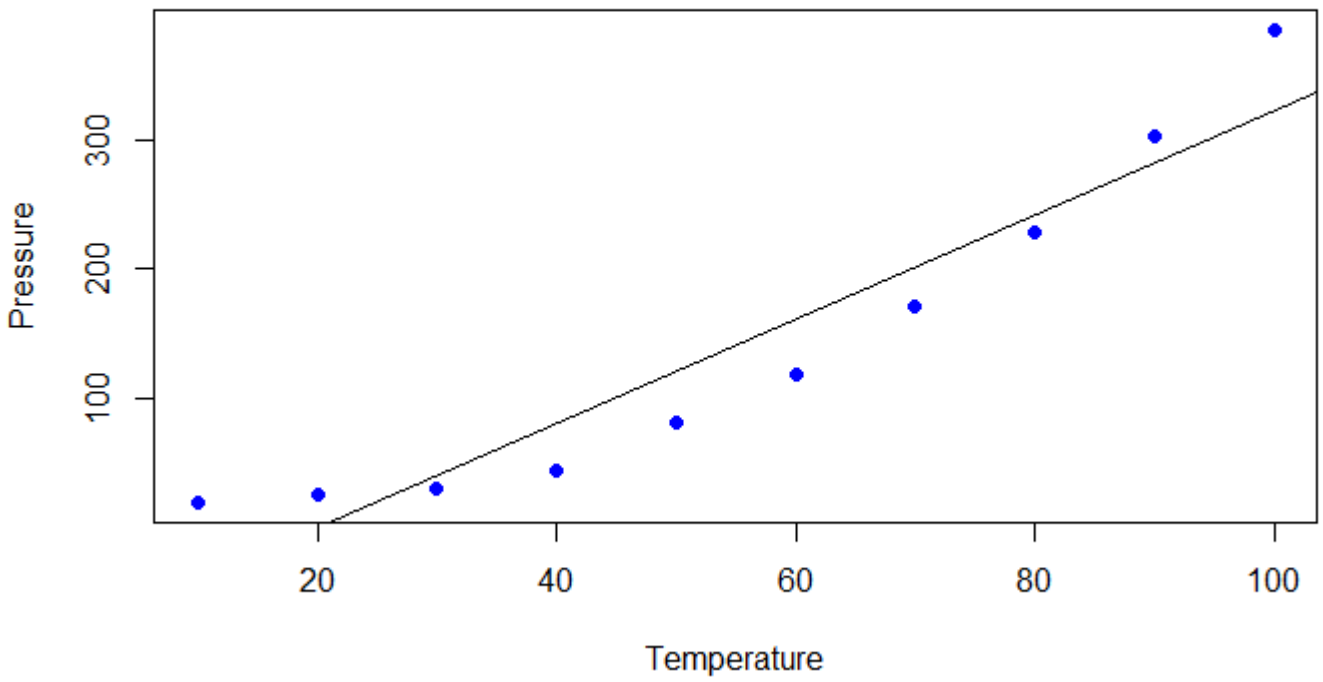
Ans- Sales= 10.4 · Spend + 541.4 · Month − 567.6

R2 abd residual

task 1:Loading data and creating linear regression and ploting the result

Hide

```
library(readxl)

pressure <- read_excel("pressure.xlsx") #Upload the data
lmTemp = lm(Pressure~Temperature, data = pressure) #Create the linear regression
plot(pressure, pch = 16, col = "blue") #Plot the results
abline(lmTemp) #Add a regression line
```



task 2: Summarizing the lmTemp

Hide

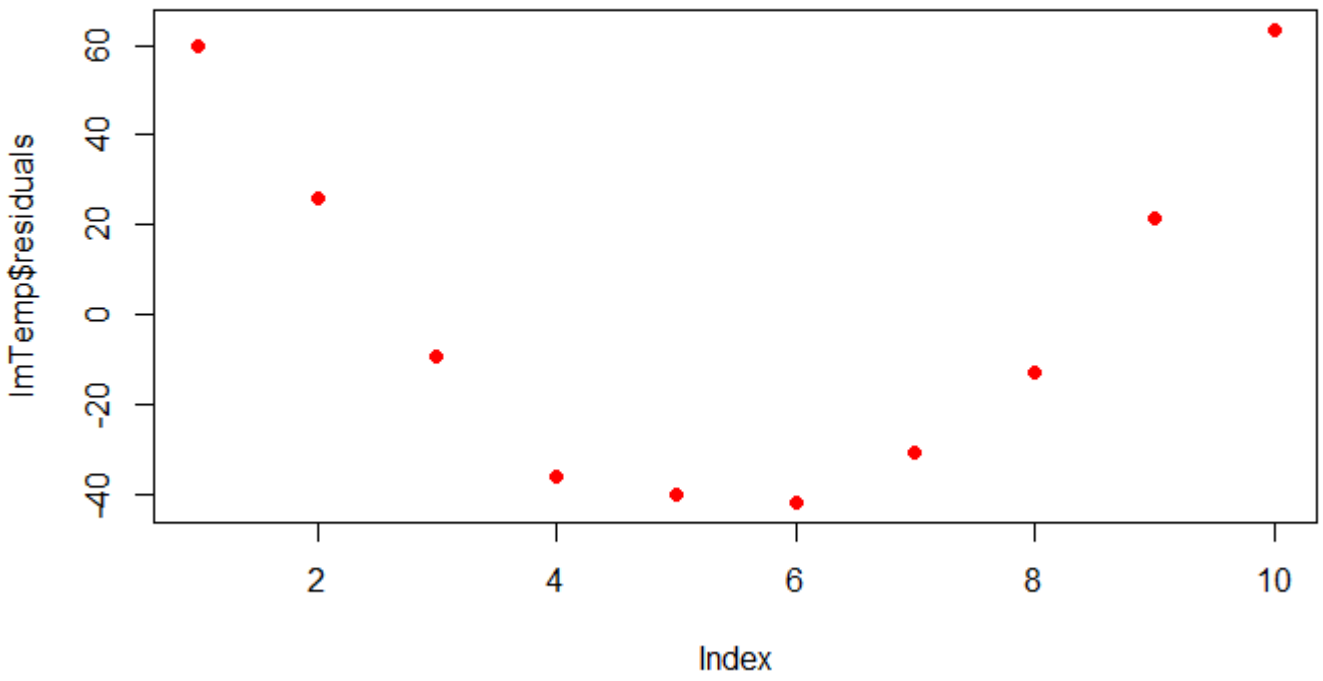
```
summary(lmTemp)
```

```
Call:
lm(formula = Pressure ~ Temperature, data = pressure)

Residuals:
   Min     1Q Median     3Q    Max
-41.85 -34.72 -10.90  24.69  63.51

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -81.5000    29.1395  -2.797   0.0233 *
Temperature   4.0309     0.4696   8.583 2.62e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 42.66 on 8 degrees of freedom
Multiple R-squared:  0.902, Adjusted R-squared:  0.8898
F-statistic: 73.67 on 1 and 8 DF,  p-value: 2.622e-05
```

task 3: ploting the residuals, use the command plot(lmTemp$residuals).

Hide

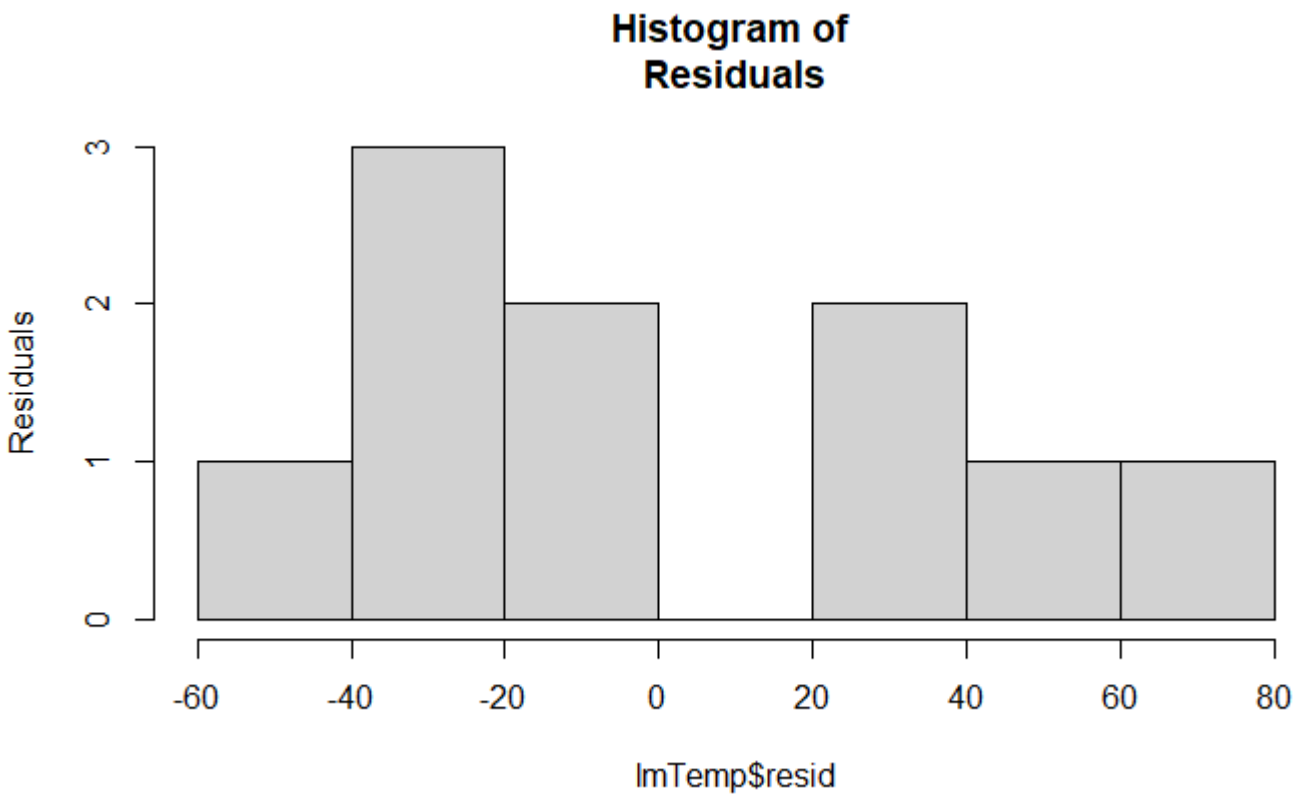
```
plot(lmTemp$residuals, pch = 16, col = "red")
```



task 4: Printing the residuals in histogram

Hide

```
hist(lmTemp$resid, main="Histogram of
Residuals", ylab="Residuals")
```



Use linear regression to predict:

Hide

```
a <- data.frame(Temperature = 170)
result <- predict(lmTemp,a)
print(result)
```

```
       1
603.7545
```

Q4: Use the linear regression to predict the pressure for temperature 40. Write your result.

Hide

```
a <- data.frame(Temperature = 40)
result <- predict(lmTemp,a)
print(result)
```

```
       1
79.73636
```

Ans- The prediction of the pressure for temperature 40 is 79.73636

Linear regression to impute missing values:

task 1: Giving the value of x,y,z and w

Hide

```
x <- 1:10
y <- c(11,12,18,14,17, NA,NA,19,NA,27)
z <- sample(1:20, 10)
w <- c(seq(1,10,3), 3,5,7,6,6,9)
```

task 2: Loading data into dataset

Hide

```
data <- data.frame(x,y,z,w)
data
```

x <int>	**y** <dbl>	**z** <int>	**w** <dbl>
1	11	2	1
2	12	6	4
3	18	15	7
4	14	10	10
5	17	13	3
6	*NA*	16	5
7	*NA*	18	7
8	19	3	6
9	*NA*	7	6
10	27	12	9

1-10 of 10 rows

task 3: Summarizing the data

Hide

```
summary(data)
```

```
       x                y               z               w         
 Min.   : 1.00   Min.   :11.00   Min.   : 2.00   Min.   : 1.00  
 1st Qu.: 3.25   1st Qu.:13.00   1st Qu.: 6.25   1st Qu.: 4.25  
 Median : 5.50   Median :17.00   Median :11.00   Median : 6.00  
 Mean   : 5.50   Mean   :16.86   Mean   :10.20   Mean   : 5.80  
 3rd Qu.: 7.75   3rd Qu.:18.50   3rd Qu.:14.50   3rd Qu.: 7.00  
 Max.   :10.00   Max.   :27.00   Max.   :18.00   Max.   :10.00  
                 NA's   :3                                      
```

Creating a dummy variable that will indicate missing data:

Hide

```
missDummy <- function(t)
{
x <- dim(length(t))
x[which(!is.na(t))] = 1
x[which(is.na(t))] = 0
return(x)
}
data$dummy <- missDummy(data$y)
data
```

x <int>	y <dbl>	z <int>	w <dbl>	dummy <dbl>
1	11	2	1	1
2	12	6	4	1
3	18	15	7	1
4	14	10	10	1
5	17	13	3	1
6	*NA*	16	5	0
7	*NA*	18	7	0
8	19	3	6	1
9	*NA*	7	6	0
10	27	12	9	1

1-10 of 10 rows

task 2: Next let us split data to 2sets (train and test):

Hide

```
TrainData<- data[data['dummy']==1,]
TestData<- data[data['dummy']==0,]

TrainData<- TrainData[,-5]
TestData<- TestData[,-5]
```

task 3: Let’s then fit a linear model with y as dependent variable and x as independent variable.

Hide

```
model<- lm(y~x, TrainData)
```

task 4: Predict missing values based on the model:

Hide

```
pred<- predict(model, TestData)
pred
```

```
       6        7        9
18.79730 20.30631 23.32432
```

task 5:Insert it back in the original

Hide

```
# Where are NAs?
data$y[is.na(y)]
```

```
[1] NA NA NA
```

Hide

```

# Replace with predicted
data$y[is.na(y)]<- pred
```