Lab Sheet-10

Linear Regression

Simple linear regression is a statistical method you can use to understand the relationship between two variables, x and y.

One variable, x, is known as the predictor variable.

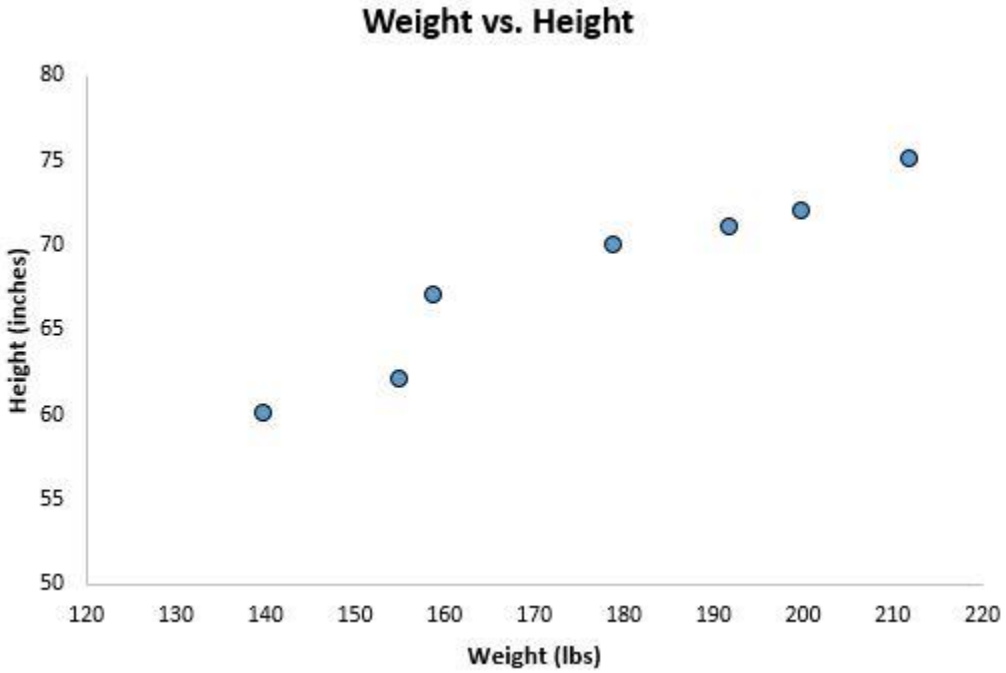
The other variable, y, is known as the response variable.

For example, suppose we have the following dataset with the weight and height of seven individuals:

Weight (lbs)	Height (inches)
140	60
155	62
159	67
179	70
192	71
200	72
212	75

Let weight be the predictor variable and let height be the response variable.

If we graph these two variables using a scatterplot, with weight on the x-axis and height on the y-axis, here's what it would look like:



Suppose we're interested in understanding the relationship between weight and height. From the scatterplot we can clearly see that as weight increases, height tends to increase as well, but to actually quantify this relationship between weight and height, we need to use linear regression.

Using linear regression, we can find the line that best "fits" our data. This line is known as the least squares regression line and it can be used to help us understand the relationships between weight and height.

The formula for the line of best fit is written as:

$$\hat{y} = b0 + b1x$$

y=mx+c

where $\hat{y}$ is the predicted value of the response variable, b0 is the y-intercept, b1 is the regression coefficient, and x is the value of the predictor variable.

How to Interpret a Least Squares Regression Line

Here is how to interpret this least squares regression line: $\hat{y} = 32.7830 + 0.2001x$

b0 = 32.7830. This means when the predictor variable weight is zero pounds, the predicted height is 32.7830 inches. Sometimes the value for b0 can be useful to know, but in this specific example it doesn't actually make sense to interpret b0 since a person can't weight zero pounds.

b1 = 0.2001. This means that a one unit increase in x is associated with a 0.2001 unit increase in y. In this case, a one pound increase in weight is associated with a 0.2001 inch increase in height.

How to Use the Least Squares Regression Line

Using this least squares regression line, we can answer questions like:

For a person who weighs 170 pounds, how tall would we expect them to be?

To answer this, we can simply plug in 170 into our regression line for x and solve for y:

$\hat{y} = 32.7830 + 0.2001(170) = 66.8$ inches

For a person who weighs 150 pounds, how tall would we expect them to be?

To answer this, we can plug in 150 into our regression line for x and solve for y:

$\hat{y} = 32.7830 + 0.2001(150) = 62.798$ inches

The Coefficient of Determination(R-squared)

One way to measure how well the least squares regression line "fits" the data is using the coefficient of determination, denoted as R2.

The coefficient of determination is the proportion of the variance in the response variable that can be explained by the predictor variable.

The coefficient of determination can range from 0 to 1. A value of 0 indicates that the response variable cannot be explained by the predictor variable at all. A value of 1 indicates that the response variable can be perfectly explained without error by the predictor variable.

An R2 between 0 and 1 indicates just how well the response variable can be explained by the predictor variable. For example, an R2 of 0.2 indicates that 20% of the variance in the response variable can be explained by the predictor variable; an R2 of 0.77 indicates that 77% of the variance in the response variable can be explained by the predictor variable.

Notice in our output from earlier we got an R2 of 0.9311, which indicates that 93.11% of the variability in height can be explained by the predictor variable of weight:

Coefficient of determination in linear regression

This tells us that weight is a very good predictor of height.

Assumptions of Linear Regression

For the results of a linear regression model to be valid and reliable, we need to check that the following four assumptions are met:

1. Linear relationship: There exists a linear relationship between the independent variable, x, and the dependent variable, y.

2. Independence: The residuals are independent. In particular, there is no correlation between consecutive residuals in time series data.

3. Homoscedasticity: The residuals have constant variance at every level of x.

4. Normality: The residuals of the model are normally distributed.

If one or more of these assumptions are violated, then the results of our linear regression may be unreliable or even misleading.

Y=mx+c

```
# x represents the weight of the people
x <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

#y represents the height of the people
y<-c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
```

```
#Apply the lm() function to find the relation between
the height and weight of the people
relation <- lm(y~x)  # output variable ~ input variable
print(relation)
#Get the summary of the Relationship
print(summary(relation))
```

```
#predict the height of new person with weight 70.
```

```
a <- data.frame(x = 70)
result <- predict(relation, a)
print(result)      # y=1.414*x+61.3    # y=160.4
```

```
#visualize the Regression Graphically
png(file = "linearregression.png")
plot(y,x,col="blue",main="Height and Weight
Regression")
abline(lm(y ~ x),cex = 1.3, pch = 16, xlab = "Weight in
Kg", ylab = "Height in cm")

#Save the file.
dev.off()
```

Example: Predict the age of a person with height 78.5

Age	height
18	76.1
19	77.7
20	78.1
21	78.2
22	78.8
23	79.7

Logistic Regression

When we want to understand the relationship between one or more predictor variables and a continuous response variable, we often use linear regression.

However, when the response variable is categorical we can instead use logistic regression.

Logistic regression is a type of classification algorithm because it attempts to "classify" observations from a dataset into distinct categories.

Here are a few examples of when we might use logistic regression:

We want to use credit score and bank balance to predict whether or not a given customer will default on a loan. (Response variable = "Default" or "No default")
We want to use average rebounds per game and average points per game to predict whether or not a given basketball player will get drafted into the NBA (Response variable = "Drafted" or "Not Drafted")
We want to use square footage and number of bathrooms to predict whether or not a house in a certain city will be listed at a selling price of $200k or more. (Response variable = "Yes" or "No")
Notice that the response variable in each of these examples can only take on one of two values. Contrast this with linear regression in which the response variable takes on some continuous value.

Assumptions of Logistic Regression
Logistic regression uses the following assumptions:

1. The response variable is binary. It is assumed that the response variable can only take on two possible outcomes.

2. The observations are independent. It is assumed that the observations in the dataset are independent of each other. That is, the observations should not come

from repeated measurements of the same individual or be related to each other in any way.

3. There is no severe multicollinearity among predictor variables. It is assumed that none of the predictor variables are highly correlated with each other.

4. There are no extreme outliers. It is assumed that there are no extreme outliers or influential observations in the dataset.

5. There is a linear relationship between the predictor variables and the logit of the response variable. This assumption can be tested using a Box-Tidwell test.

The Logistic Regression Equation
Logistic regression uses a method known as maximum likelihood estimation to find an equation of the following form:

$$\log[p(X) / (1-p(X))] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

where:

X_j: The jth predictor variable
β_j: The coefficient estimate for the jth predictor variable
The formula on the right side of the equation predicts the log odds of the response variable taking on a value of 1.

Thus, when we fit a logistic regression model we can use the following equation to calculate the probability that a given observation takes on a value of 1:

$$\mathbf{p(X) = e(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p) / (1 + e(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p))}$$

We then use some probability threshold to classify the observation as either 1 or 0.

For example, we might say that observations with a probability greater than or equal to 0.5 will be classified as “1” and all other observations will be classified as “0.”

How to Interpret Logistic Regression Output

Suppose we use a logistic regression model to predict whether or not a given basketball player will get drafted into the NBA based on their average rebounds per game and average points per game.

Here is the output for the logistic regression model:

Interpret logistic regression output

Using the coefficients, we can compute the probability that any given player will get drafted into the NBA based on their average rebounds and points per game using the following formula:

P(Drafted) = e-2.8690 + 0.0698*(rebs) + 0.1694*(points) / (1+e-2.8690 + 0.0698*(rebs) + 0.1694*(points))

For example, suppose a given player averages 8 rebounds per game and 15 points per game. According to the model, the probability that this player will get drafted into the NBA is 0.557.

P(Drafted) = e-2.8690 + 0.0698*(8) + 0.1694*(15) / (1+e-2.8690 + 0.0698*(8) + 0.1694*(15)) = 0.557

Since this probability is greater than 0.5, we would predict that this player will get drafted.

Contrast this with a player who only averages 3 rebounds and 7 points per game. The probability that this player will get drafted into the NBA is 0.186.

P(Drafted) = e-2.8690 + 0.0698*(3) + 0.1694*(7) / (1+e-2.8690 + 0.0698*(3) + 0.1694*(7)) = 0.186

Since this probability is less than 0.5, we would predict that this player will not get drafted.

6. The sample size is sufficiently large.

The function used to create the regression model is the **glm()** function.

Syntax

The basic syntax for **glm()** function in logistic regression is −

glm(formula,data,family)

Following is the description of the parameters used −

- **formula** is the symbol presenting the relationship between the variables.
- **data** is the data set giving the values of these variables.
- **family** is R object to specify the details of the model. It's value is binomial for logistic regression.

Example

The in-built data set "mtcars" describes different models of a car with their various engine specifications. In "mtcars" data set, the transmission mode (automatic or manual) is described by the column am which is a binary value (0 or 1). We can create a logistic regression model between the columns "am" and 3 other columns - hp, wt and cyl.

```
# Select some columns form mtcars.
input <- mtcars[,c("am","cyl","hp","wt")]

print(head(input))
```

Create Regression Model

We use the **glm()** function to create the regression model and get its summary for analysis.

```
input <- mtcars[,c("am","cyl","hp","wt")]

amdata = glm(formula = am ~ cyl + hp + wt, data = input, family = binomial)

print(summary(amdata))
```

Conclusion

In the summary as the p-value in the last column is more than 0.05 for the variables "cyl" and "hp", we consider them to be insignificant in contributing to the value of the variable "am". Only weight (wt) impacts the "am" value in this regression model.

Example 2:

```
# Installing the package
install.packages("caTools")    # For Logistic regression
install.packages("ROCR")       # For ROC curve to evaluate model

# Loading package
library(caTools)
library(ROCR)

# Splitting dataset
split <- sample.split(mtcars, SplitRatio = 0.8)
split

train_reg <- subset(mtcars, split == "TRUE")
test_reg <- subset(mtcars, split == "FALSE")

# Training model
logistic_model <- glm(vs ~ wt + disp,
                      data = train_reg,
                      family = "binomial")
logistic_model

# Summary
summary(logistic_model)

# Predict test data based on model
predict_reg <- predict(logistic_model,
                       test_reg, type = "response")
predict_reg

# Changing probabilities
predict_reg <- ifelse(predict_reg >0.5, 1, 0)
```

```
# Evaluating model accuracy
# using confusion matrix
table(test_reg$vs, predict_reg)
err <- mean(predict_reg != test_reg$vs)
print(paste('Accuracy =', 1 - err))
 # ROC-AUC Curve
ROCPred <- prediction(predict_reg, test_reg$vs)
ROCPer <- performance(ROCPred, measure = "tpr",
                              x.measure = "fpr")
auc <- performance(ROCPred, measure = "auc")
auc <- auc@y.values[[1]]
auc


# Plotting curve


plot(ROCPer)
```

wt influences dependent variables positively and one unit increase in wt increases the log of odds for vs =1 by 1.44. **disp** influences dependent variables negatively and one unit increase in disp decreases the log of odds for vs =1 by 0.0344. Null deviance is 31.755(fit dependent variable with intercept) and Residual deviance is 14.457(fit dependent variable with all independent variable). AIC(Alkaline Information criteria) value is 20.457 i.e the lesser the better for the model. Accuracy comes out to be 0.75 i.e 75%.

Simulations:

Simulations are a powerful statistical tool. Simulation techniques allow us to carry out statistical inference in complex models, estimate quantities that we can cannot calculate analytically.

Standard Probability Distributions

Sometimes you want to generate data from a distribution (such as normal), or want to see where a value falls in a known distribution. `R` has these distributions built in:

- Normal
- Binomial
- Beta
- Exponential
- Gamma

- Hypergeometric

Each family of functions for a distribution has 4 options:

- `r` for random number generation [e.g. `rnorm()`]
- `d` for density [e.g. `dnorm()`]
- `p` for probability [e.g. `pnorm()`]
- `q` for quantile [e.g. `qnorm()`]

For example, we can simulate a random sample of size 5 from a standard normal distribution by using `rnorm`.

```
rnorm(5)
[1]  0.81199910  1.32466134 -0.05470965  0.38102517 -1.83418402
```

rnorm(5, mean=10,sd=5)

To find the probability of being less than 4 in a Normal distribution with mean 5 and standard deviation 2, we would use `pnorm`.

```
pnorm(4, mean = 5, sd = 2)
[1] 0.3085375
```

To find the 97.5 percentile in a standard normal distribution (i.e the number z such that 97.5% of the probability is to the left of z in a standard normal distribution), we would use `qnorm`.

```
qnorm(0.975)
[1] 1.959964
```

Acceptance-Rejection Technique to Generate Random Variate

- Example: use following steps to generate uniformly distributed random numbers between 1/4 and 1.

  Step 1.
  Generate a random number R
  Step 2a.
  If $R \geq 1/4$, accept $X = R$, goto Step 3

Step 2b.

If $R < 1/4$, reject *R*, return to Step 1

Step 3.

If another uniform random variate on [1/4, 1] is needed, repeat the procedure begining at Step 1. Otherwise stop.

- Do we know if the random variate generated using above methods is indeed uniformly distributed over [1/4, 1]? The answer is Yes.