Logistic Regression

In linear regression the Y variable is always a continuous variable. If suppose, the Y variable was categorical, you cannot use linear regression model it.

So what would you do when the Y is a categorical variable with 2 classes?

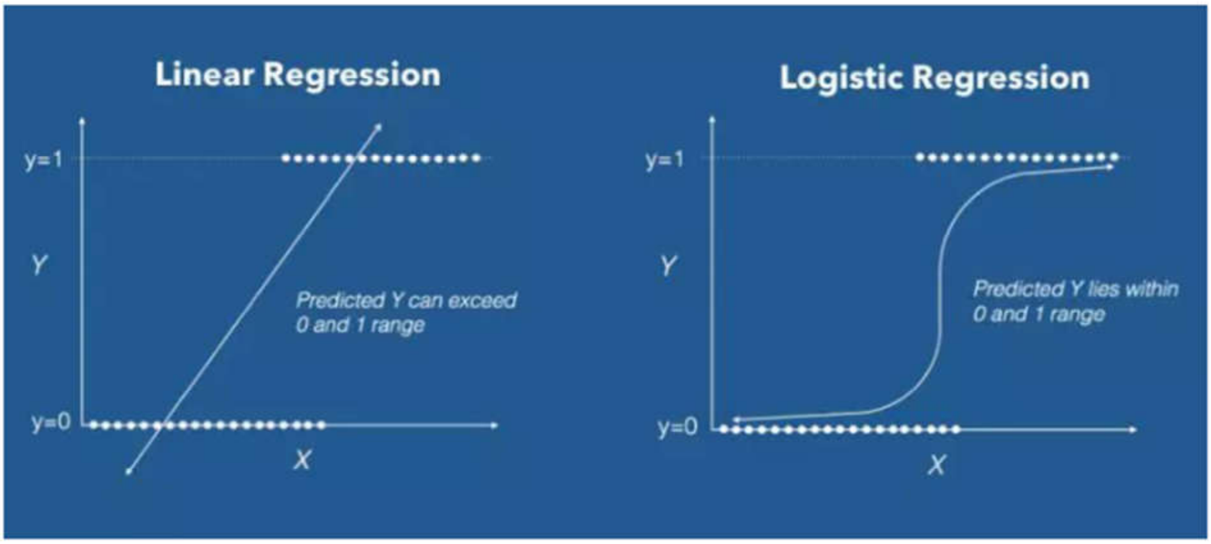
Logistic regression can be used to model and solve such problems, also called as binary classification problems.

A key point to note here is that Y can have 2 classes only and not more than that. If Y has more than 2 classes, it would become a multi class classification and you can no longer use the logistic regression for that.

Here are some examples of binary classification problems:

- **Spam Detection** : Predicting if an email is Spam or not
- **Credit Card Fraud** : Predicting if a given credit card transaction is fraud or not
- **Health** : Predicting if a given mass of tissue is benign or malignant
- **Marketing** : Predicting if a given user will buy an insurance product or not
- **Banking** : Predicting if a customer will default on a loan.

When the response variable has only 2 possible values, it is desirable to have a model that predicts the value either as 0 or 1 or as a probability score that ranges between 0 and 1.



About the storms data

This data is a subset of the NOAA Atlantic hurricane database best track data.The data includes the positions and attributes of 198 tropical storms, measured every six hours during the lifetime of a storm.

A tibble with 10,010 observations and 13 variables:

Name : Storm Name
year,month,day: Date of report
hour: Hour of report (in UTC)
lat,long: Location of storm center(longuitude and latitude)
status: Storm classification (Tropical Depression, Tropical Storm, or Hurricane)
category: Saffir-Simpson storm category (estimated from wind speed. -1 = Tropical Depression, 0 = Tropical Storm)
wind: storm's maximum sustained wind speed (in knots)
pressure: Air pressure at the storm's center (in millibars)
ts_diameter: Diameter of the area experiencing tropical storm strength winds (34 knots or above)
hu_diameter: Diameter of the area experiencing hurricane strength winds (64 knots or above)

```
> str(storms)
tibble [10,010 x 13] (S3: tbl_df/tbl/data.frame)
 $ name       : chr [1:10010] "Amy" "Amy" "Amy" "Amy" ...
 $ year       : num [1:10010] 1975 1975 1975 1975 1975 ...
 $ month      : num [1:10010] 6 6 6 6 6 6 6 6 6 6 ...
 $ day        : int [1:10010] 27 27 27 27 28 28 28 28 29 29 ...
 $ hour       : num [1:10010] 0 6 12 18 0 6 12 18 0 6 ...
 $ lat        : num [1:10010] 27.5 28.5 29.5 30.5 31.5 32.4 33.3 34 34.4 34 ...
 $ long       : num [1:10010] -79 -79 -79 -79 -78.8 -78.7 -78 -77 -75.8 -74.8 ...
 $ status     : chr [1:10010] "tropical depression" "tropical depression" "tropical depression" "tropical
 depression" ...
 $ category   : Ord.factor w/ 7 levels "-1"<"0"<"1"<"2"<..: 1 1 1 1 1 1 1 1 2 2 ...
 $ wind       : int [1:10010] 25 25 25 25 25 25 25 30 35 40 ...
 $ pressure   : int [1:10010] 1013 1013 1013 1013 1012 1012 1011 1006 1004 1002 ...
 $ ts_diameter: num [1:10010] NA NA NA NA NA NA NA NA NA NA ...
 $ hu_diameter: num [1:10010] NA NA NA NA NA NA NA NA NA NA ...
> 
```

```
> print(dat)
# A tibble: 10,010 x 13
   name   year month   day  hour   lat  long status    category  wind pressure ts_diameter hu_diameter
   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>     <ord>    <int>    <int>       <dbl>       <dbl>
 1 Amy    1975     6    27     0  27.5 -79   tropical~ -1          25     1013          NA          NA
 2 Amy    1975     6    27     6  28.5 -79   tropical~ -1          25     1013          NA          NA
 3 Amy    1975     6    27    12  29.5 -79   tropical~ -1          25     1013          NA          NA
 4 Amy    1975     6    27    18  30.5 -79   tropical~ -1          25     1013          NA          NA
 5 Amy    1975     6    28     0  31.5 -78.8 tropical~ -1          25     1012          NA          NA
 6 Amy    1975     6    28     6  32.4 -78.7 tropical~ -1          25     1012          NA          NA
 7 Amy    1975     6    28    12  33.3 -78   tropical~ -1          25     1011          NA          NA
 8 Amy    1975     6    28    18  34   -77   tropical~ -1          30     1006          NA          NA
 9 Amy    1975     6    29     0  34.4 -75.8 tropical~ 0           35     1004          NA          NA
10 Amy    1975     6    29     6  34   -74.8 tropical~ 0           40     1002          NA          NA
# ... with 10,000 more rows
```

Now lest apply logistic resgression for **storms: Amy&Bob~long**. So here the independent variable(X) is "long" and the dependent variable(Y) which must be catagorical is the "name" of the storm. In this case the independent variable can have only two possible values (Amy and Bob). Sowe can apply logistic regression without any problem.

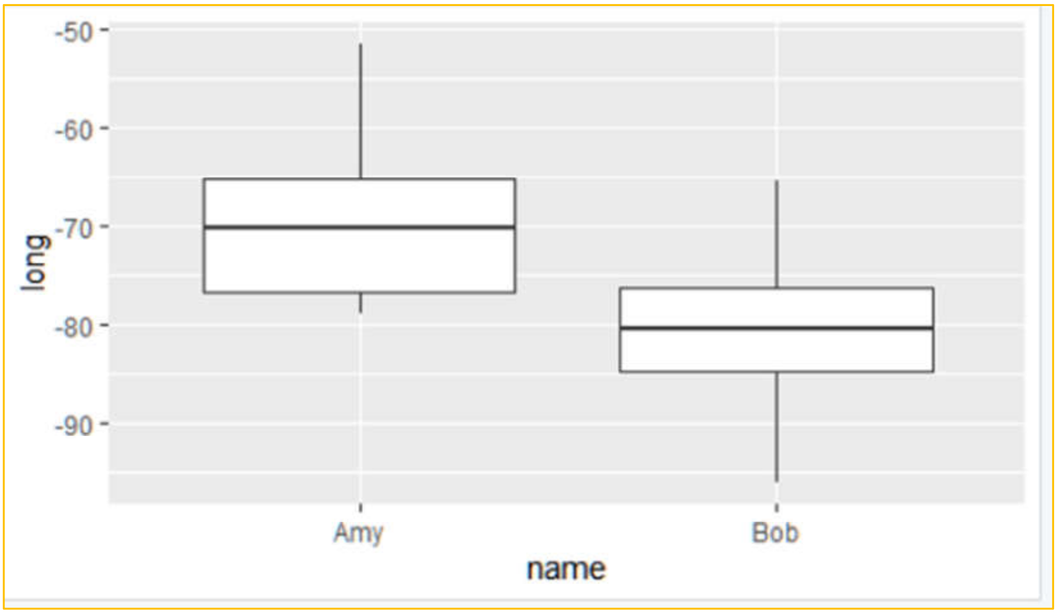
Step 1: # Limit the dataset to the two columns of interest.

```
> df <-data.frame(sqldf("select name,long  from dat  where name='Amy' OR name='Bob'"))
> str(df)
'data.frame':	101 obs. of  2 variables:
 $ name: chr  "Amy" "Amy" "Amy" "Amy" ...
 $ long: num  -79 -79 -79 -79 -78.8 -78.7 -78 -77 -75.8 -74.8 ...
>
```

Step 2: Plot the graph to see how the name of a storm related to longuitude (long), Moreover to see if storm longuitude could predict whether a storm name is Amy or Bob. Visually, this relationship would look like:

Using t-test

```
> #1 t-test statistics
> ggplot(df, aes(name, long)) +
+   geom_boxplot()
>
```



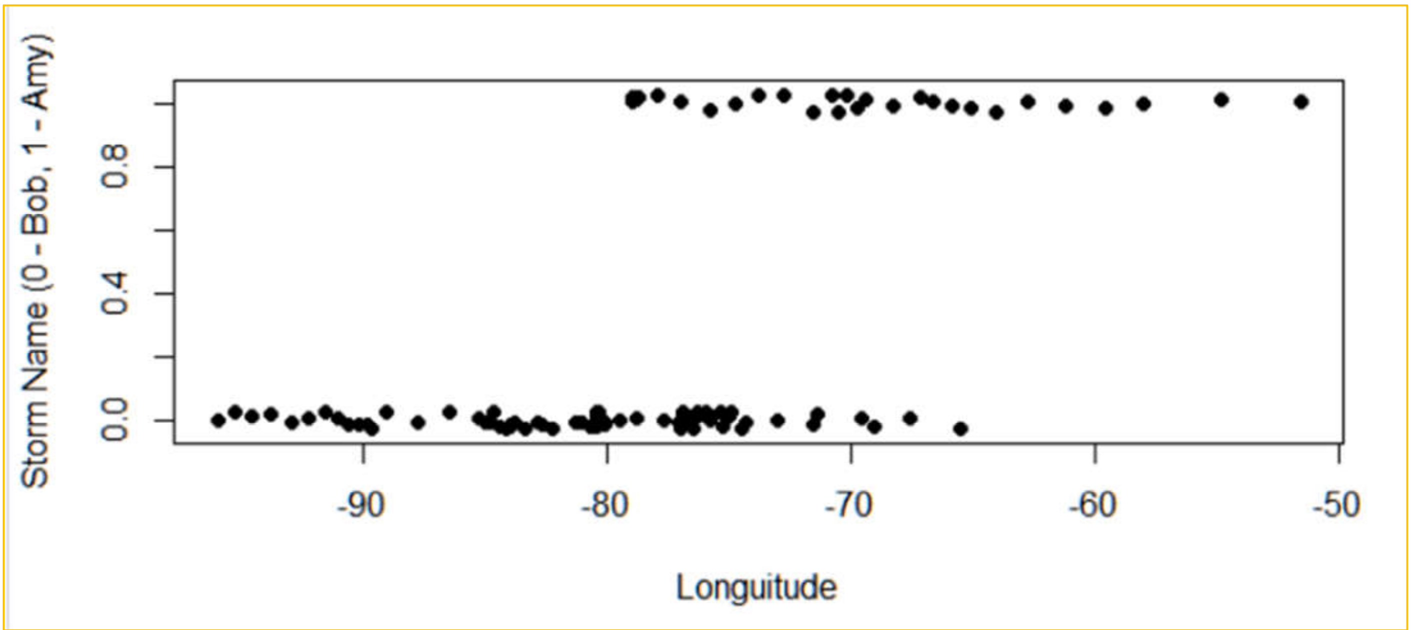
Since logistic regression involves fitting models in which we have 0's and 1's so we need to translate the two names (Amy and Bob) to 0(Bob) and 1(Amy). Here is the resulting structure.

```
> df$name <-ifelse(df$name=="Amy",1,0)
> str(df)
'data.frame':	101 obs. of  2 variables:
 $ name: num  1 1 1 1 1 1 1 1 1 1 ...
 $ long: num  -79 -79 -79 -79 -78.8 -78.7 -78 -77 -75.8 -74.8 ...
> 
```

Store the name and long in variables for readability as shown below.

```
> name_code <-df$name
> Longitude <-df$long
> 
```

```
> #2 plotting graph
> plot(Longitude,jitter(name_code,0.15),pch=19,xlab="Longuitude",ylab="Storm Name (0 - Bob, 1 - Amy)")
> 
```



Clearly as we can see from the graph as the longitude increases then there is high probability that the name of the storm is Amy and otherwise Bob.

Step 3: Building the logistic regression model. *What is the probability that the storm name is Amy or Bob given the storm's longuitude?*

```
> #3 Building the model
> model <-glm(name_code~Longitude,data=df,family=binomial)
> model
```

```
Call:  glm(formula = name_code ~ Longitude, family = binomial, data = df)

Coefficients:
(Intercept)    Longitude
    17.5228       0.2432

Degrees of Freedom: 100 Total (i.e. Null);  99 Residual
Null Deviance:      122.9
Residual Deviance: 80.22        AIC: 84.22
>
```

Formula mathematically represented as

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

$$p_{name} = \frac{1}{1 + e^{-(17.5 + 0.2432 long)}}$$

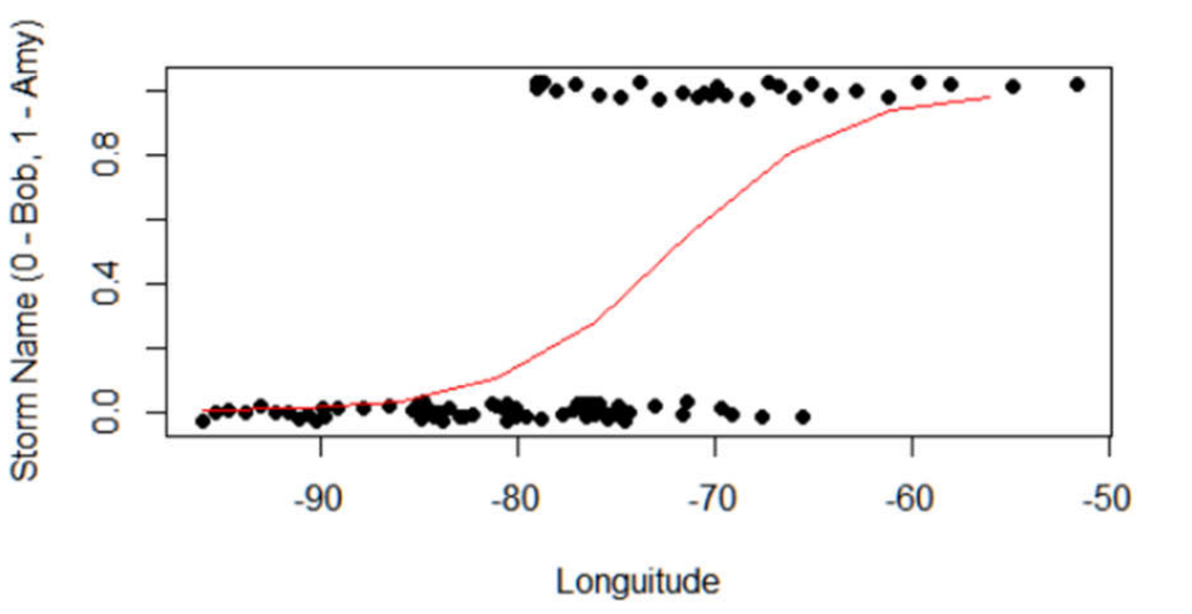
Step 4: Making predictions, first we create a sequence of test data set ranging from minimum to maximum longitude with a difference of 0.01.

```
> #4 making predictions
> #create a sequence of test data set
> sprintf("Min Longuitude:%f Maximum Loguitude:%f",min(Longitude),max(Longitude))
[1] "Min Longuitude:-96.000000 Maximum Loguitude:-51.600000"
```

Then make prediction using the model and test data set.

Using increase x axis by 5

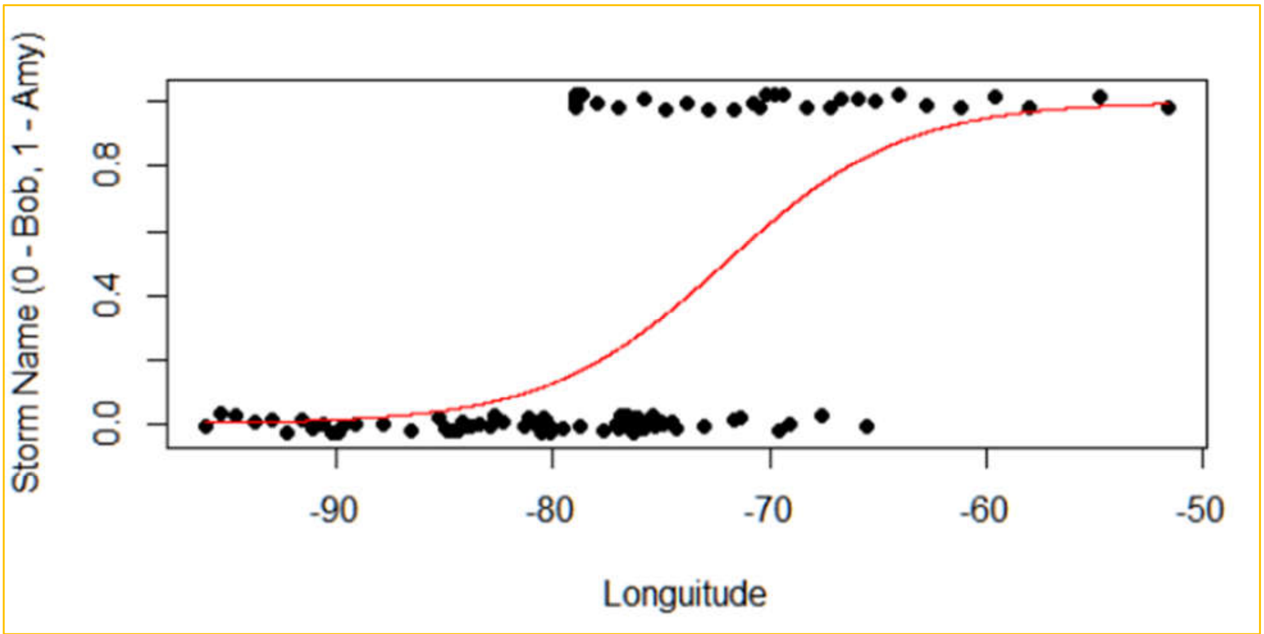
```
xv <-seq(min(Longitude),max(Longitude),5)
yv <-predict(model,list(Longitude=xv),type="response")
plot(Longitude,jitter(name_code,0.15),pch=19,xlab="Longuitude",ylab="Storm Name (0 - Bob, 1 - Amy)")
lines(xv,yv,col="red")
```



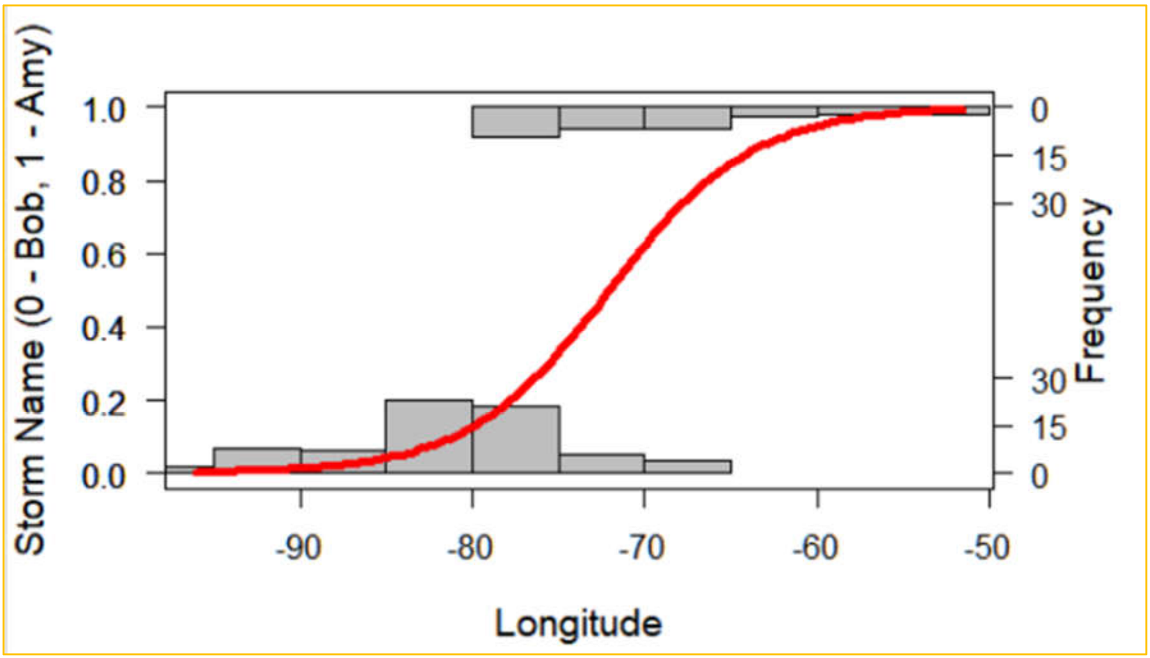
The logistic curve does not follow the complete sigmoid shape when limited to the original longitude range. To see the full shape, we can increase the x-axis range.

Increase x axis by 0.01

```
xv <-seq(min(Longitude),max(Longitude),0.01)
yv <-predict(model,list(Longitude=xv),type="response")
plot(Longitude,jitter(name_code,0.15),pch=19,xlab="Longuitude",ylab="Storm Name (0 - Bob, 1 - Amy)")
lines(xv,yv,col="red")
```



```
> logi.hist.plot(Longitude,name_code,boxp=FALSE,type="count",col="gray"
+                ,xlabel="Longitude",ylabel="Storm Name (0 - Bob, 1 - Amy)")
> |
```



Assessment of linear regression model

1: Assessing the fit with a pseudo R^2

To assess how well a logistic model fits the data, we make use of the **log-likelihood** method ((aka *null* model)

$$p_{null} = \frac{1}{1 + e^{-(b_0)}}$$

$$p_{null} = \frac{1}{1 + e^{-(17.5)}}$$

The log-likelihood statistic (often labeled as **-2LL** in some statistical packages) for the null model is,

```
> #Assessment of logistic regression
> model$null.deviance
[1] 122.882
```

We want -2LL for the full model (i.e. the model with the Longitude predictor variable) to be smaller than that of the null model. To extract -2LL from the model, type:

```
> model$deviance
[1] 80.22148
```

It is good that the value is smaller than that of the null model.

The difference between both log-likelihood values is referred to as the **model Chi-square**.

```
> modelChi <- model$null.deviance - model$deviance
> pseudo.R2 <- modelChi / model$null.deviance
> pseudo.R2
[1] 0.3471663
```

So according to the result the model can explain for 34.7% of the variability in the name variable.

Alternative pseudo R2

```
> #Alternative pseudo R2
> lrm(name ~ long, df)
Logistic Regression Model

 lrm(formula = name ~ long, data = df)

                      Model Likelihood     Discrimination    Rank Discrim.
                         Ratio Test           Indexes           Indexes
 Obs           101    LR chi2      42.66    R2       0.490    C       0.859
  0             71    d.f.             1    g        2.380    Dxy     0.719
  1             30    Pr(> chi2) <0.0001    gr      10.810    gamma   0.720
 max |deriv| 2e-05                          gp       0.310    tau-a   0.303
                                            Brier    0.130

           Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept 17.5228 4.0040 4.38   <0.0001
 long       0.2432 0.0534 4.56   <0.0001
```

Note how this value of 0.49 differs from that of the *Hosmer and Lemeshow* R2 whose value is 0.34.

Assessing the significance

```
> #Assessing model significance
> Chidf <- model$df.null - model$df.residual
> chisq.prob <- 1 - pchisq(modelChi, Chidf)
> chisq.prob
[1] 6.511469e-11
```

Since the p-value is small then we can reject the null hypothesis that the current model does not improve on the base model. Here, the p-value is almost 0.

Parameter significance

```
Call:
glm(formula = name_code ~ Longitude, family = binomial, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8873  -0.7335  -0.3193   0.3710   1.9267

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 17.52277    4.00346   4.377 1.20e-05 ***
Longitude    0.24315    0.05337   4.556 5.22e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 122.882  on 100  degrees of freedom
Residual deviance:  80.221  on  99  degrees of freedom
AIC: 84.221

Number of Fisher Scoring iterations: 5
```

The name coefficient p-value is almost 0 so we reject the null hypothesis.

Multi-variable model

So far, we've worked with a single variable model. We can augment the model by adding more variables. For example, we will add the fraction of the population that has attained a bachelor's degree to the model (we'll ignore the possibility of co-dependence between variables for pedagogical sake).

The entire workflow follows:

Grab variables of interest

```
> #Multi-variable model adding new variable pressure
> # Grab variables of interest
> df2 <-data.frame(sqldf("select name,long,lat  from dat  where name='Amy' OR name='Bob'"))
> df2$name <-ifelse(df$name=="Amy",1,0)
> str(df2)
'data.frame':   101 obs. of  3 variables:
 $ name: num  1 1 1 1 1 1 1 1 1 1 ...
 $ long: num  -79 -79 -79 -79 -78.8 -78.7 -78 -77 -75.8 -74.8 ...
 $ lat : num  27.5 28.5 29.5 30.5 31.5 32.4 33.3 34 34.4 34 ...
```

Run regression model
Compute pseudo R-square,
Compute the pseudo p-value

```
> # Run regression model
> model2 <-glm(name ~ long + lat, df2,family=binomial,control = list(maxit = 50))
> # Compute pseudo R-square
> modelChi <- model2$null.deviance - model2$deviance
> pseudo.R2 <- modelChi / model2$null.deviance
> pseudo.R2
[1] 0.3918381
> # Compute the pseudo p-value
> Chidf <- model2$df.null - model2$df.residual
> modelChi <-model2$null.deviance - model2$deviance
> modelChi
[1] 48.14984
> 1 - pchisq(modelChi, Chidf)
[1] 3.502643e-11
```

Assess each parameter's significance

```
> # Assess each parameter's significance
> summary(model2)

Call:
glm(formula = name ~ long + lat, family = binomial, data = df2,
    control = list(maxit = 50))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5560  -0.6462  -0.2998   0.2534   1.9866

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 31.40922    8.16335   3.848 0.000119 ***
long         0.35441    0.08312   4.264 2.01e-05 ***
lat         -0.16484    0.07429  -2.219 0.026493 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 122.882  on 100  degrees of freedom
Residual deviance:  74.732  on  98  degrees of freedom
AIC: 80.732

Number of Fisher Scoring iterations: 6
```

Note the change in the long coefficient p-value when adding another variable that may well be explaining the same variability in name (i.e. long and lat are very likely correlated). In fact longitude and latitude are highly correlated.

R CODE

```
#install neccessary packages
install.packages("dplyr")
install.packages("ggplot2")
install.packages("rms",dependencies = TRUE)

#import neccessary packages
library(dplyr)
library(ggplot2)
library(rms)
require(sqldf)
library(popbio)
#data(storms)
str(storms) #to view structure of data
dat <-storms

#1 Limit data to two variables
df <-data.frame(sqldf("select name,long  from dat  where name='Amy' OR
name='Bob'"))
str(df)

#1 t-test statistics
ggplot(df, aes(name, long)) +
  geom_boxplot()

#Assign numeric values to classes(Amy,Bob)
df$name <-ifelse(df$name=="Amy",1,0)
str(df)
name_code <-df$name
Longitude <-df$long

#2 plotting graph
plot(df$long,jitter(df$name,0.15),pch=19,xlab="long",ylab="name")

#3 Building the model
model <-glm(name~long,data=df,family=binomial)
model

#4 making predictions
#create a sequence of test data set
sprintf("Min Longuitude:%f Maximum Loguitude:%f",min(Longitude),max(Longitude))

xv <-seq(min(Longitude),max(Longitude),0.01)
yv <-predict(model,list(long=xv),type="response")

plot(Longitude,jitter(name_code,0.15),pch=19,xlab="Longuitude",ylab="Storm Name
(0 - Bob, 1 - Amy)")
lines(xv,yv,col="red")

logi.hist.plot(Longitude,name_code,boxp=FALSE,type="count",col="gray"
               ,xlabel="Longitude",ylabel="Storm Name (0 - Bob, 1 - Amy)")
```

```
#Assessment of logistic regression
model$null.deviance
model$deviance
modelChi <- model$null.deviance - model$deviance
pseudo.R2 <- modelChi / model$null.deviance
pseudo.R2

#Alternative pseudo R2
lrm(name ~ long, df)

#Assessing model significance
Chidf <- model$df.null - model$df.residual
chisq.prob <- 1 - pchisq(modelChi, Chidf)
chisq.prob

#Assessing parameter significance
summary(model)

#Multi-variable model adding new variable pressure
#Grab variables of interest
df2 <-data.frame(sqldf("select name,long,lat  from dat  where name='Amy' OR
name='Bob'"))
df2$name <-ifelse(df$name=="Amy",1,0)
str(df2)

#Run regression model
model2 <-glm(name ~ long + lat, df2,family=binomial,control = list(maxit = 50))


#Compute pseudo R-square
modelChi <- model2$null.deviance - model2$deviance
pseudo.R2 <- modelChi / model2$null.deviance
pseudo.R2

#Compute the pseudo p-value
Chidf <- model2$df.null - model2$df.residual
modelChi <-model2$null.deviance - model2$deviance
modelChi
1 - pchisq(modelChi, Chidf)


#Assess each parameter's significance
summary(model2)
```