Regression

Rikita Patangay

2022-09-25

Intro to Linear Regression

In linear regression the data will consist of predictor values (x) and target values (y). To find the relationship between x and y we will use linear regression. This relationship can be defined using parameters such as w and b. Regression models a target prediction value for the provided data, based on independent variables.

Strengths: - It is a relatively simple algorithm. - Efficient when the data follows a linear pattern. - Has low variance.

Weaknesses: - High bias because it assumes a linear shape to the data.

Read in Dataset: Weather Data

Using read.csv to read in the file and put the data set in the variable (df).

```
df <- read.csv("WeatherData.csv")
str(df)
```

```
## 'data.frame':    8784 obs. of  9 variables:
##  $ Date.Time      : chr  "1/1/2012 0:00" "1/1/2012 1:00" "1/1/2012 2:00" "1/1/2012 3:00" ...
##  $ Temp_C         : num  -1.8 -1.8 -1.8 -1.5 -1.5 -1.4 -1.5 -1.4 -1.4 -1.3 ...
##  $ Dew.Point      : num  -3.9 -3.7 -3.4 -3.2 -3.3 -3.3 -3.1 -3.6 -3.6 -3.1 ...
##  $ Temp_C.1       : int  86 87 89 88 88 87 89 85 85 88 ...
##  $ Rel.Hum_.      : int  4 4 7 6 7 9 7 7 9 15 ...
##  $ Wind.Speed_km.h: num  8 8 4 4 4.8 6.4 6.4 8 8 4 ...
##  $ Visibility_km  : num  101 101 101 101 101 ...
##  $ Press_kPa      : chr  "Fog" "Fog" "Freezing Drizzle,Fog" "Freezing Drizzle,Fog" ...
##  $ Weather        : logi  NA NA NA NA NA NA ...
```

a.) Split data 80/20

Here I am dividing the data in training and test sets. This works by randomly sampling the data using the sample() function. This is an 80/20 split.

```
set.seed(1234)
i <- sample(1:nrow(df), .80*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

b.) 5 Functions

These are 5 pretty simple functions used for data exploration.

head() - gives some of the start of the data sample

tail() - gives some of the end of the data sample

nrow() - gives number of rows in the sample

ncol() - gives number of columns in the sample

summary() - gives a brief summary of values associated with the data.

```
head(train)
```

```
##             Date.Time Temp_C Dew.Point Temp_C.1 Rel.Hum_. Wind.Speed_km.h
## 7452  11/6/2012 11:00    1.9      -4.5       62         9            48.3
## 8016 11/29/2012 23:00   -9.4     -15.3       62        20            25.0
## 7162  10/25/2012 9:00    6.4      -0.3       62        17            48.3
## 8086  12/2/2012 21:00    6.7       6.6       99        11             8.0
## 7269 10/29/2012 20:00   14.3       9.7       74        52            25.0
## 623   1/26/2012 22:00   -5.3     -10.2       68        22            16.1
##      Visibility_km     Press_kPa Weather
## 7452        101.94 Mostly Cloudy      NA
## 8016        102.42  Mainly Clear      NA
## 7162        101.97 Mostly Cloudy      NA
## 8086        100.70           Fog      NA
## 7269         99.55        Cloudy      NA
## 623         100.92          Snow      NA
```

```
tail(train)
```

```
##             Date.Time Temp_C Dew.Point Temp_C.1 Rel.Hum_. Wind.Speed_km.h
## 941     2/9/2012 4:00   -3.4      -8.0       70        26            25.0
## 2071   3/27/2012 6:00   -7.5     -16.4       49         9            48.3
## 6782  10/9/2012 13:00   15.0       4.4       49         7            48.3
## 94     1/4/2012 21:00   -7.6     -11.6       73         7            11.3
## 3226   5/14/2012 9:00   19.2       8.6       50         6            48.3
## 7764 11/19/2012 11:00    2.4      -3.4       65         6            24.1
##      Visibility_km     Press_kPa Weather
## 941         101.10         Clear      NA
## 2071        102.31         Clear      NA
## 6782        101.71 Mostly Cloudy      NA
## 94          100.54          Snow      NA
## 3226        101.54 Mostly Cloudy      NA
## 7764        102.83  Mainly Clear      NA
```

```
nrow(train)
```

```
## [1] 7027
```

```
ncol(train)
```

```
## [1] 9
```

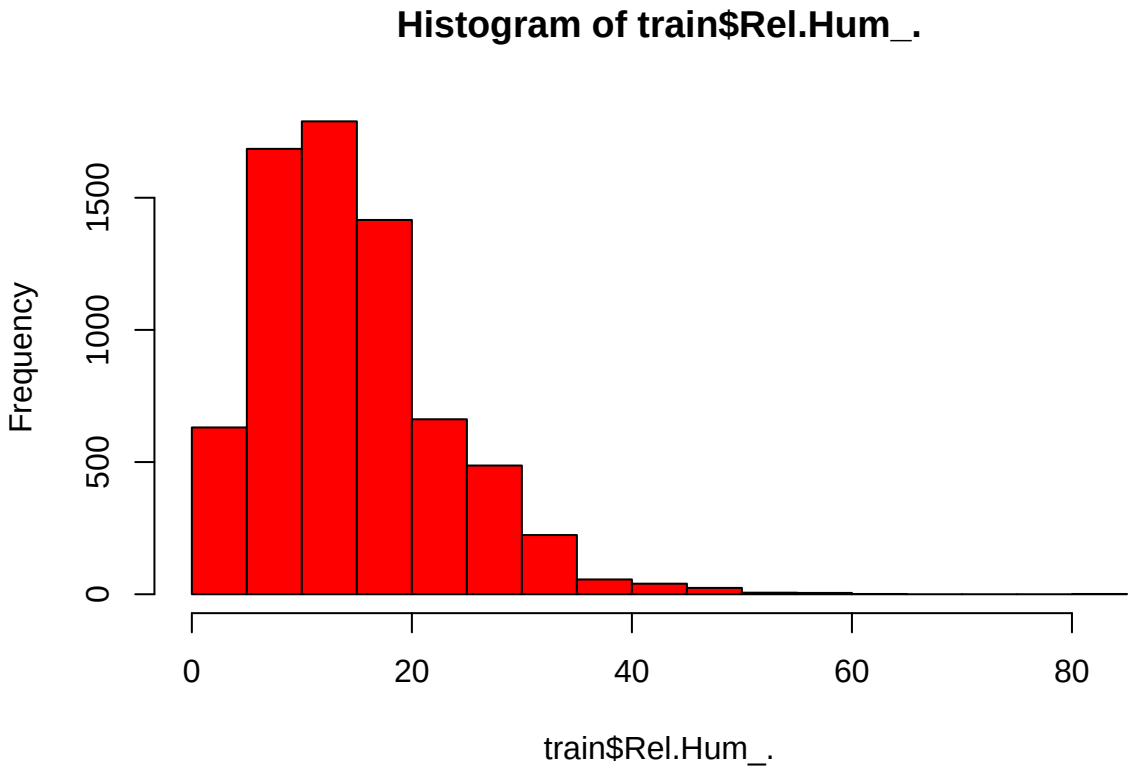
```
summary(train)
```

```
##   Date.Time             Temp_C          Dew.Point          Temp_C.1
##  Length:7027        Min.   :-23.300   Min.   :-28.500   Min.   : 18.00
##  Class :character   1st Qu.:  0.300   1st Qu.: -5.800   1st Qu.: 56.00
##  Mode  :character   Median :  9.400   Median :  3.600   Median : 68.00
##                     Mean   :  8.837   Mean   :  2.612   Mean   : 67.51
##                     3rd Qu.: 18.700   3rd Qu.: 11.800   3rd Qu.: 81.00
##                     Max.   : 33.000   Max.   : 24.400   Max.   :100.00
##    Rel.Hum_.     Wind.Speed_km.h Visibility_km     Press_kPa
##  Min.   : 0.00   Min.   : 0.2    Min.   : 97.52   Length:7027
##  1st Qu.: 9.00   1st Qu.:24.1    1st Qu.:100.56   Class :character
##  Median :13.00   Median :25.0    Median :101.07   Mode  :character
##  Mean   :14.97   Mean   :27.7    Mean   :101.05
##  3rd Qu.:20.00   3rd Qu.:25.0    3rd Qu.:101.58
##  Max.   :83.00   Max.   :48.3    Max.   :103.65
##  Weather
##  Mode:logical
##  NA's:7027
##
##
##
##
```

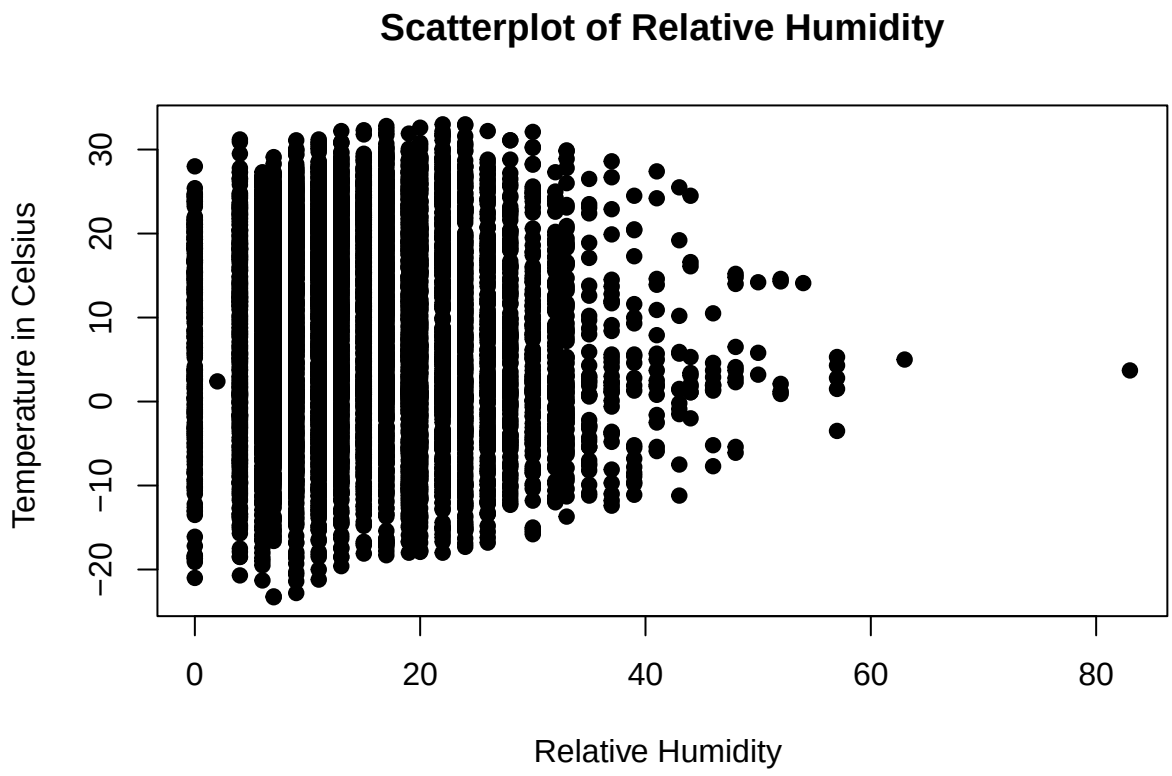
c.) 2 Informative Graphs

These are two infromative graphs. The first one is a histogram and the second is scatterplot.

```
hist(train$Rel.Hum_., breaks=12, col="red")
```




```
attach(train)
plot(Rel.Hum_., Temp_C, main="Scatterplot of Relative Humidity",
   xlab="Relative Humidity ", ylab="Temperature in Celsius ", pch=19)
```



d.) Build a Simple Regression Model

Here we are building a simple regression model. What we are doing in the model is using the predictor which would be the Rel.Hum. Which is the relative humidity, and our target (y) is Visibility.

Call: lm(formula = Rel.Hum_. ~ Visibility_km, data = train)

Residuals: Min:-48.806 1Q: -8.806 Median: 0.427 3Q: 9.194 Max: 50.039

Coefficients: (Estimate Std., Error, t value, Pr(>|t|)<– corresponds with numbers below)
(Intercept) 91.09954, 0.37749, 241.33, <2e-16 ***Visibility_km -0.85173, 0.01241, -68.65, <2e-16***

Signif. codes: 0 ‘***’ ***0.001*** ‘**’ *0.01* ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.09 on 7025 degrees of freedom

Multiple R-squared: 0.4015, Adjusted R-squared: 0.4014

F-statistic: 4712 on 1 and 7025 DF

p-value: < 2.2e-16

e.)

Could not figure out residual plots :(

f.) Multiple Regression Model

Here we are building a multiple regression model. What we are doing in the model is using the predictor which would be the Rel.Hum. Which is the relative humidity, and our target (y) is Temp_C which is temperature in Celsius. We can use the data to compare the levels of Temp/Humidity and how they correlate.

Call: lm(formula = Rel.Hum_. ~ Temp_C, data = train)

Residuals: Min: -49.966 1Q: -12.004 Median: 0.735 3Q: 13.177 Max: 34.096

Coefficients:

(Estimate Std., Error, t value, Pr(>|t|) <– corresponds with numbers below)

(Intercept) 70.20387, 0.24781, 283.30, <2e-16 ***

Temp_C -0.30494, 0.01695, -17.99, <2e-16 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.55 on 7025 degrees of freedom

Multiple R-squared: 0.04405, Adjusted R-squared: 0.04391

F-statistic: 323.7 on 1 and 7025 DF p-value: < 2.2e-16

g.) Third Regression Model

Here we are building another regression model. What we are doing in the model is using the predictor which would be the Wind.Speed_km.h. Which is the Wind Speed in kilometers/hr, and our target (y) is Visibility_km which is the visibility in km We can use the data to compare the levels of Visibility/Wind Speed and how they correlate.

Call: lm(formula = Wind.Speed_km.h ~ Visibility_km, data = train)

Residuals:

Min: -15.022 1Q: -6.022 Median: -1.926 3Q: 5.036 Max: 68.036 68.036

Coefficients:

(Estimate Std., Error, t value, Pr(>|t|)<– corresponds with numbers below)

(Intercept) 14.901389, 0.251386, 59.277, <2e-16 ***

Visibility_km 0.002507, 0.008263, 0.303, 0.762

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

h.) Comparing the results

I think that the multiple regression is the most accuate because it shows the correlation of multiple variables, and the correlation between them.

i.) Predict and evaluate