R Notebook

Sai Gonuguntla

09/14/2022

###in linear regression the target variable y is quantitative and x is the predictor which can be quantitative or qualitative. It displays the relationship between x and y. Strengths are,it works well when there is a linear pattern in the data,it has low variance,and is a pretty simple algorithm. Weaknesses are, it has high bias because its looking for a linear relation in the data, so it doesn't perform well when there are non-linear relationships.

#read in the data set

```
options(stringsAsFactors = FALSE)
df <- read.csv("weatherHistory.csv", header=TRUE)
str(df)
```

```
## 'data.frame':    11759 obs. of  12 variables:
##  $ Formatted.Date          : chr  "2006-04-01 00:00:00.000 +0200" "2006-04-01 01:0
0:00.000 +0200" "2006-04-01 02:00:00.000 +0200" "2006-04-01 03:00:00.000 +0200" ...
##  $ Summary                 : chr  "Partly Cloudy" "Partly Cloudy" "Mostly Cloudy"
"Partly Cloudy" ...
##  $ Precip.Type             : chr  "rain" "rain" "rain" "rain" ...
##  $ Temperature..C.         : num  9.47 9.36 9.38 8.29 8.76 ...
##  $ Apparent.Temperature..C.: num  7.39 7.23 9.38 5.94 6.98 ...
##  $ Humidity                : num  0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.82 0.7
2 ...
##  $ Wind.Speed..km.h.       : num  14.12 14.26 3.93 14.1 11.04 ...
##  $ Wind.Bearing..degrees.  : num  251 259 204 269 259 258 259 260 259 279 ...
##  $ Visibility..km.         : num  15.8 15.8 15 15.8 15.8 ...
##  $ Loud.Cover              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Pressure..millibars.    : num  1015 1016 1016 1016 1017 ...
##  $ Daily.Summary           : chr  "Partly cloudy throughout the day." "Partly clou
dy throughout the day." "Partly cloudy throughout the day." "Partly cloudy throughout
the day." ...
```

a. Divide into train and test

```
#data cleaning
whist <- df[,c(5,6,7,9)]

set.seed(1234)
i <- sample(1:nrow(df), 0.80*nrow(df), replace=FALSE)
train <- whist[i,]
test <- whist[-i,]
```

b. 5 R functions for data exploration using the training data.

```
attach(whist)
tail(train,n=2)
```

	Apparent.Temperature..C. <dbl>	Humidity <dbl>	Wind.Speed..km.h. <dbl>	Visibility..km. <dbl>
10616	-4.983333	0.93	2.0769	2.8014
10609	-5.027778	0.92	3.1073	2.9624

2 rows

```
dim(train)
```

```
## [1] 9407    4
```

```
str(train)
```

```
## 'data.frame':    9407 obs. of  4 variables:
##  $ Apparent.Temperature..C.: num  17.71 11.06 -3.16 12.89 3.52 ...
##  $ Humidity                : num  0.6 0.84 0.57 0.9 0.89 0.86 0.69 0.69 0.79 0.87
...
##  $ Wind.Speed..km.h.       : num  10.37 6.75 24.05 1.61 6.44 ...
##  $ Visibility..km.         : num  9.76 14.17 11.03 9.98 5.92 ...
```

```
summary(train)
```

```
##  Apparent.Temperature..C.    Humidity      Wind.Speed..km.h. Visibility..km.
##  Min.   :-22.0944         Min.   :0.1500   Min.   : 0.000    Min.   : 0.000
##  1st Qu.:  0.4167         1st Qu.:0.6300   1st Qu.: 5.015    1st Qu.: 7.905
##  Median : 10.8611         Median :0.8000   Median : 9.322    Median : 9.982
##  Mean   :  9.7363         Mean   :0.7502   Mean   :10.210    Mean   : 9.760
##  3rd Qu.: 17.9917         3rd Qu.:0.9000   3rd Qu.:13.701    3rd Qu.:11.270
##  Max.   : 38.3778         Max.   :1.0000   Max.   :47.527    Max.   :16.100
```

```
head(train,n=5)
```

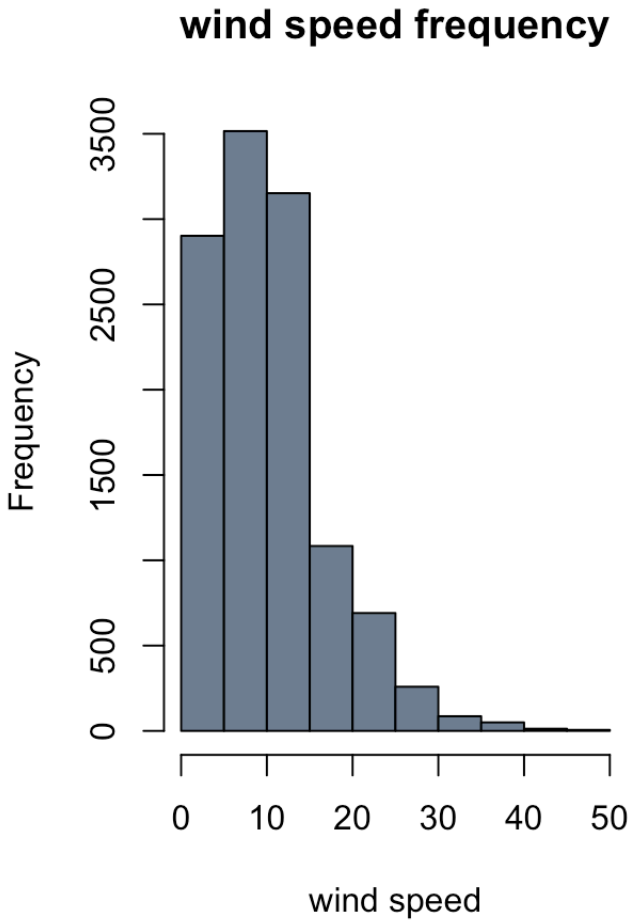
	Apparent.Temperature..C. <dbl>	Humidity <dbl>	Wind.Speed..km.h. <dbl>	Visibility..km. <dbl>
7452	17.705556	0.60	10.3684	9.7566
8016	11.055556	0.84	6.7459	14.1680
7162	-3.155556	0.57	24.0534	11.0285
8086	12.888889	0.90	1.6100	9.9820
7269	3.522222	0.89	6.4400	5.9248

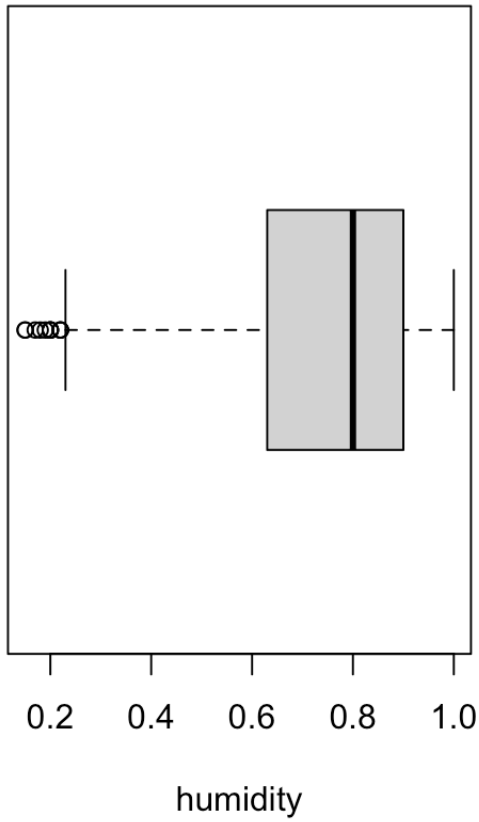
5 rows

c. 2 informative graphs, using the training data.

Displays a histogram that shows windspeed frequency and a box plot which displays the humidity.

```
par(mfrow=c(1,2))
hist(Wind.Speed..km.h., col="slategray", main="wind speed frequency",xlab="wind speed
")
boxplot(Humidity,horizontal=TRUE, xlab="humidity")
```

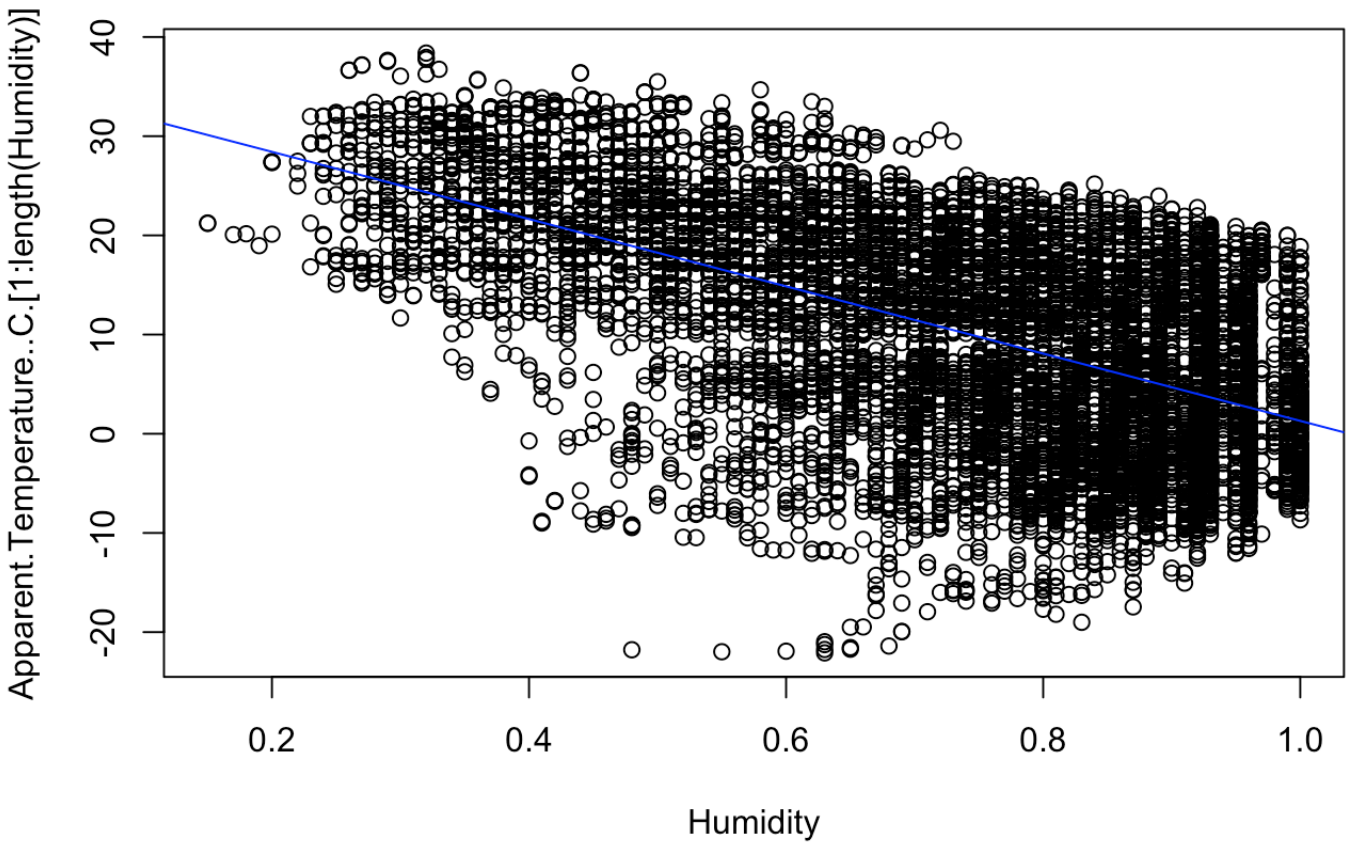




d. Build a simple linear regression model (one predictor) and output the summary.

The predictor is humidity and target is Apparent Temperature. Residuals show the distance that a point is from the regression line and range from -40.708 to 19.316. Standard error is the estimate of variation and is used to predict confidence interval. The p value is very close to 0 which is good because we want a value less than 0.5 so the model shows a good goodness of fit. On the other hand R squared is 0.3325 which is not ideal because it should be close to 1. The RSE is in units of y so it shows that the model is off by about 8.968 degrees celsius.

```
plot(Humidity, Apparent.Temperature..C.[1:length(Humidity)])
abline(lm(Apparent.Temperature..C.~Humidity), col="blue")
```




```
lm1 <- lm(Apparent.Temperature..C.~Humidity, data=train)
lm1
```

```
##
## Call:
## lm(formula = Apparent.Temperature..C. ~ Humidity, data = train)
##
## Coefficients:
## (Intercept)     Humidity
##       35.27       -34.03
```

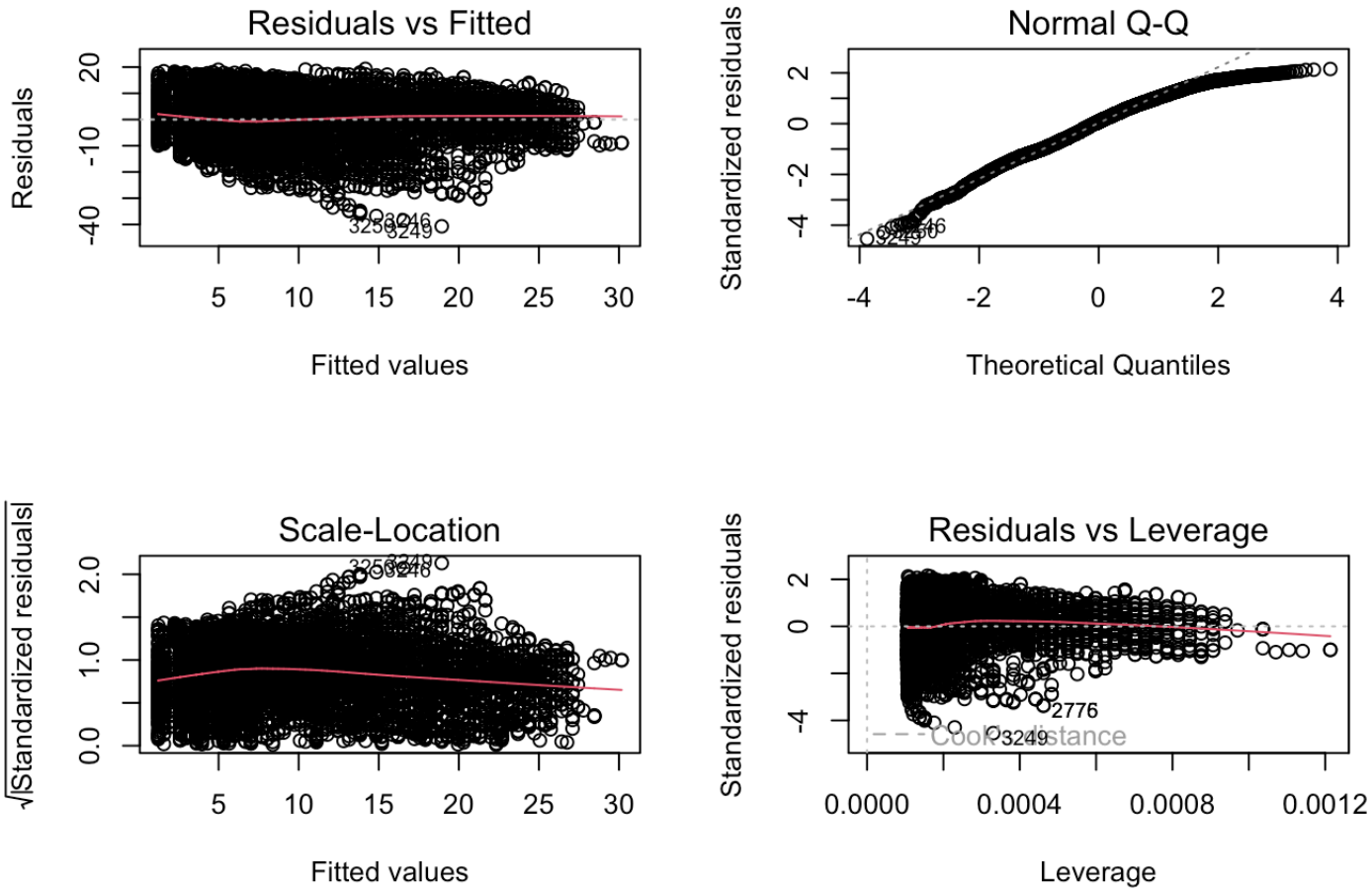
```
summary(lm1)
```

```
##
## Call:
## lm(formula = Apparent.Temperature..C. ~ Humidity, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.709  -6.401   0.625   6.897  19.316
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.2660     0.3843   91.78   <2e-16 ***
## Humidity    -34.0304     0.4972  -68.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.968 on 9405 degrees of freedom
## Multiple R-squared:  0.3325, Adjusted R-squared:  0.3325
## F-statistic:  4685 on 1 and 9405 DF,  p-value: < 2.2e-16
```

e. simple linear regression residual

The 1st plot shows whether the residuals have a non-linear pattern. There are equally spread residuals around a horizontal line so it shows that we don't have non-linear relationships The 2nd plot shows that the residuals are normally distributed since the residuals are lined well with the dashed line. The 3rd plot shows whether he residuals are spread equally around the predictors. The plot shows that there are equally spread points along the horizontal line which is good. The 4h plot shows if there are any influential cases that may affect the regression line. In this case 3249 seems to be a problem point as it lies far away from cook's distance

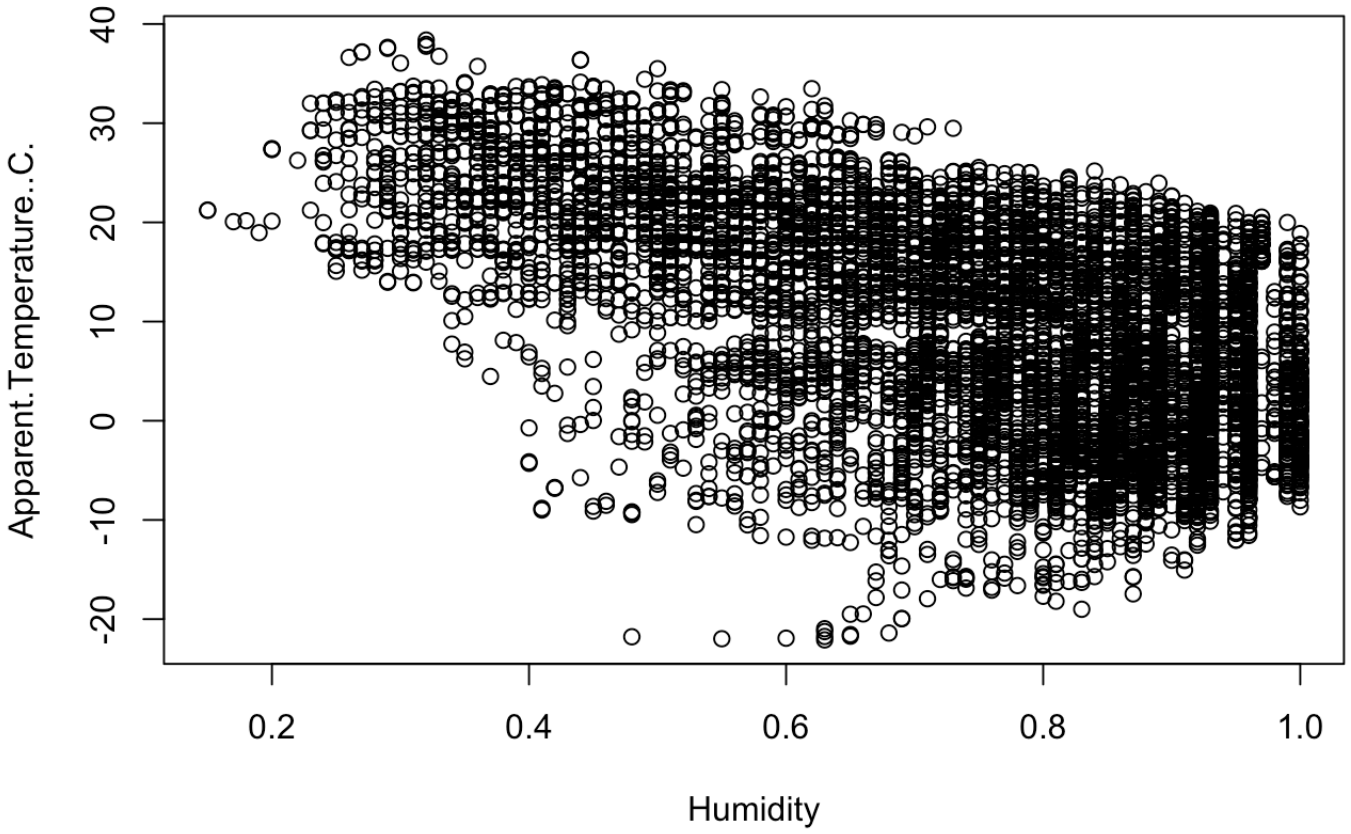
```
par(mfrow=c(2,2))
plot(lm1)
```



f. multiple linear regression model (multiple predictors), output the summary and residual plots.

The predictors are humidity and wind speed, target is Apparent Temperature. residuals show the distance that a point is from the regression line and range from -37.479 to 21.921. The p value is very close to 0 which is good because we want a value less than 0.5 so the model shows a good goodness of fit. On the other hand R squared is 0.3696 which is not ideal because it should be close to 1. The RSE shows that the model is off by about 8.716 degrees celsius.

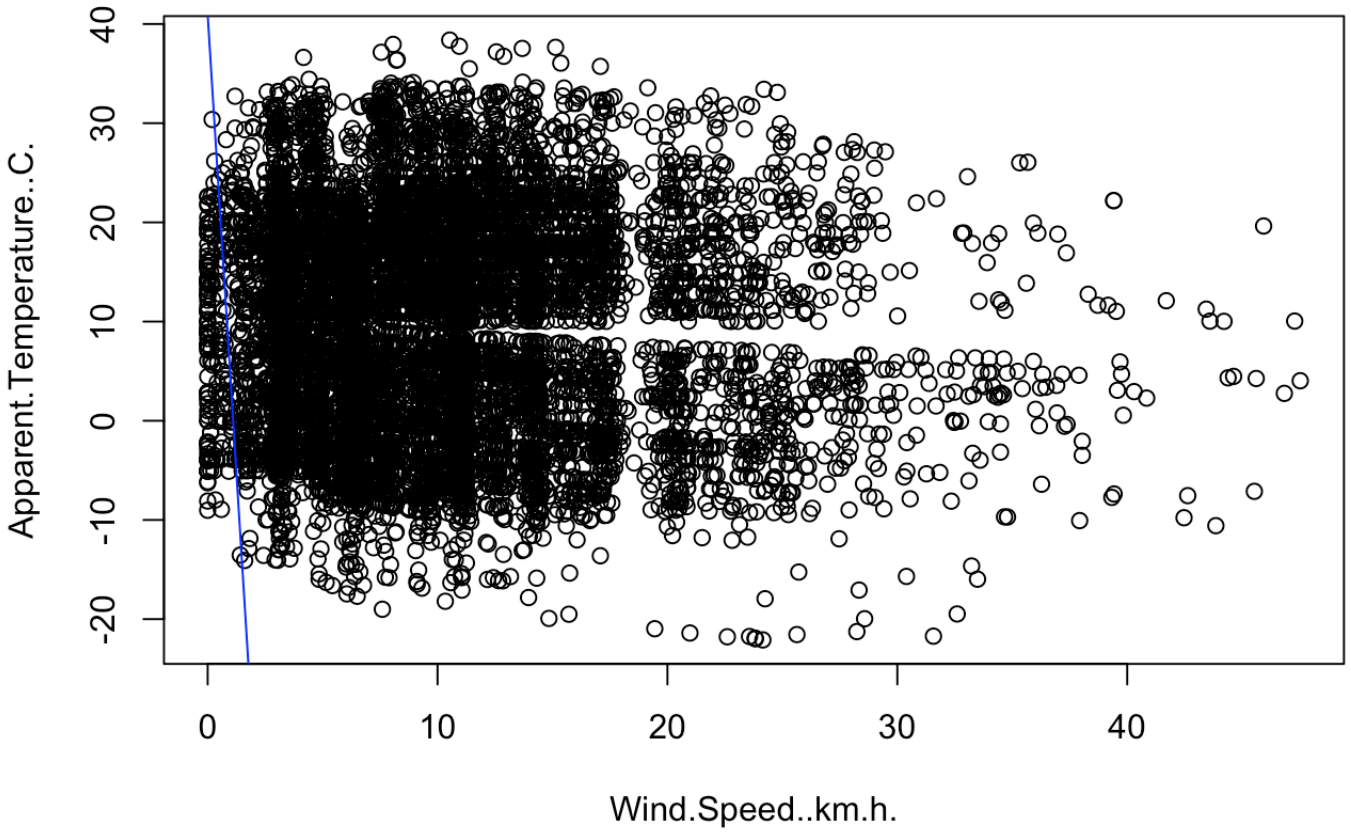
```
plot(Apparent.Temperature..C.~Humidity+Wind.Speed..km.h., data=train)
```




```
#plot(Apparent.Temperature..C.[1:length(Humidity+wind.Speed)])

abline(lm(Apparent.Temperature..C.~Humidity+Wind.Speed..km.h.), col="blue")
```

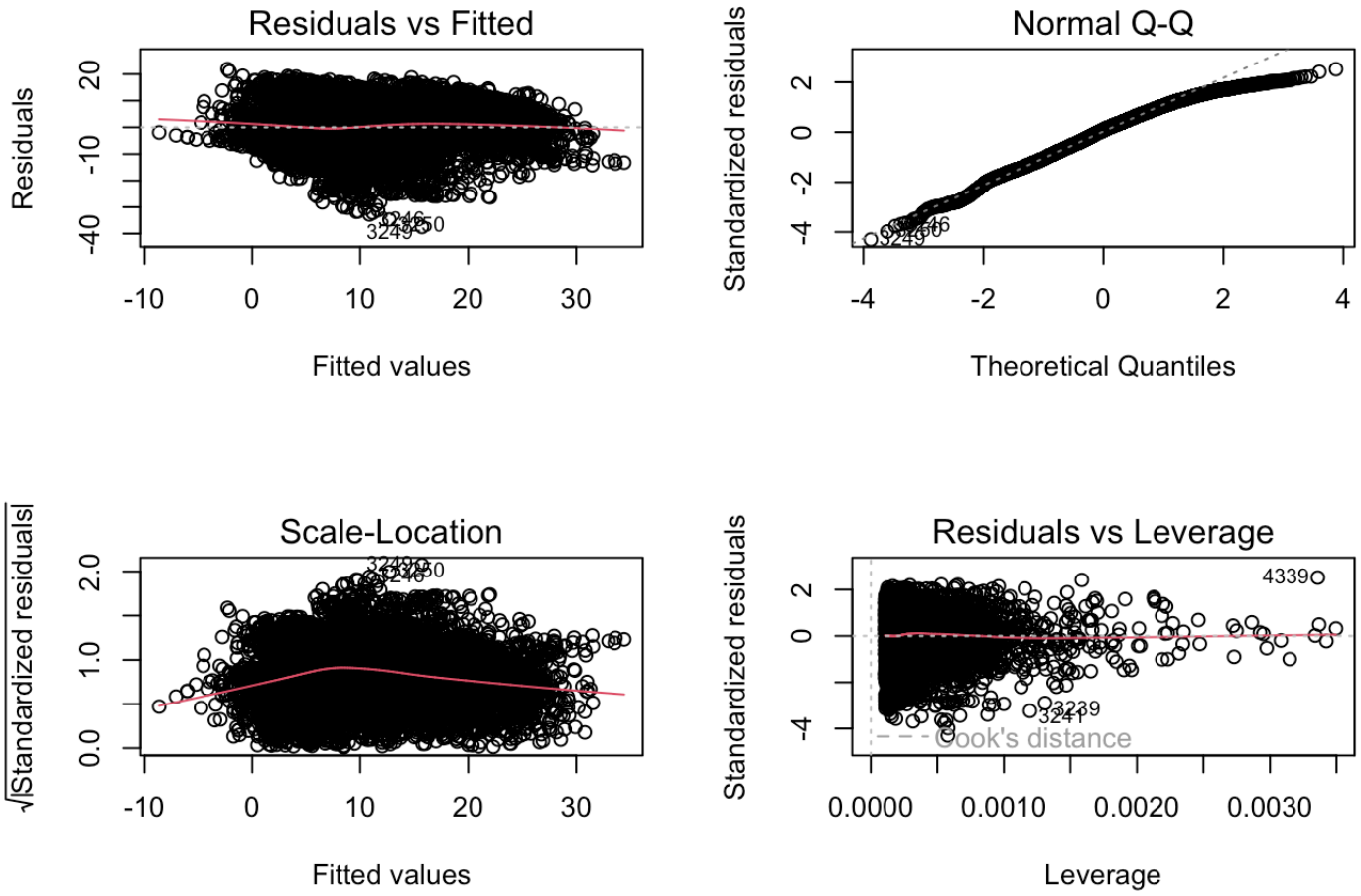
```
## Warning in abline(lm(Apparent.Temperature..C. ~ Humidity + Wind.Speed..km.h.), :
## only using the first two of 3 regression coefficients
```




```
lm2 <- lm(Apparent.Temperature..C.~Humidity+Wind.Speed..km.h., data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Apparent.Temperature..C. ~ Humidity + Wind.Speed..km.h.,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.479  -6.059   0.737   6.554  21.921
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         40.86159    0.44283   92.27   <2e-16 ***
## Humidity           -37.04687    0.49992  -74.11   <2e-16 ***
## Wind.Speed..km.h.   -0.32641    0.01388  -23.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.716 on 9404 degrees of freedom
## Multiple R-squared:  0.3696, Adjusted R-squared:  0.3695
## F-statistic:  2757 on 2 and 9404 DF,  p-value: < 2.2e-16
```

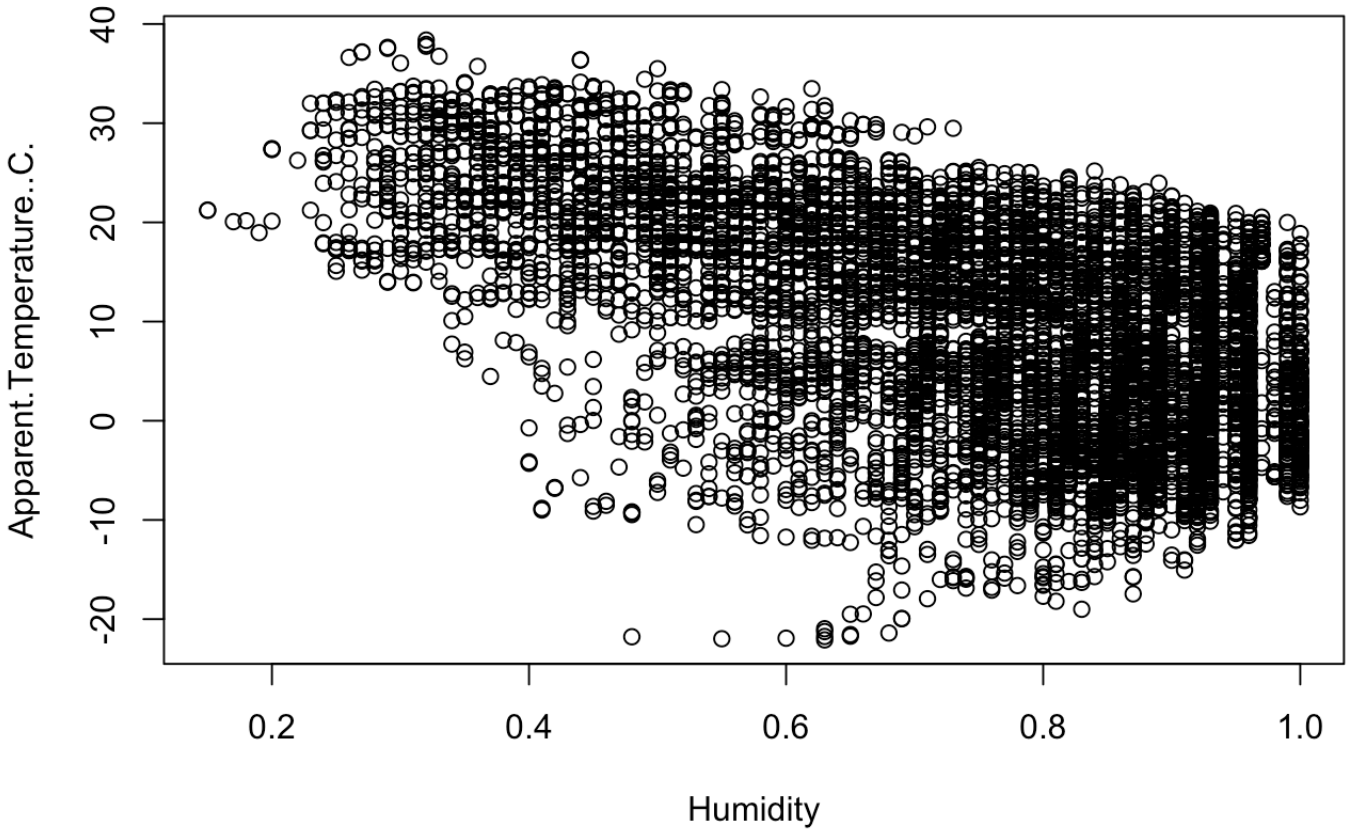
```
par(mfrow=c(2,2));
plot(lm2)
```



g. Third Linear Regression Model

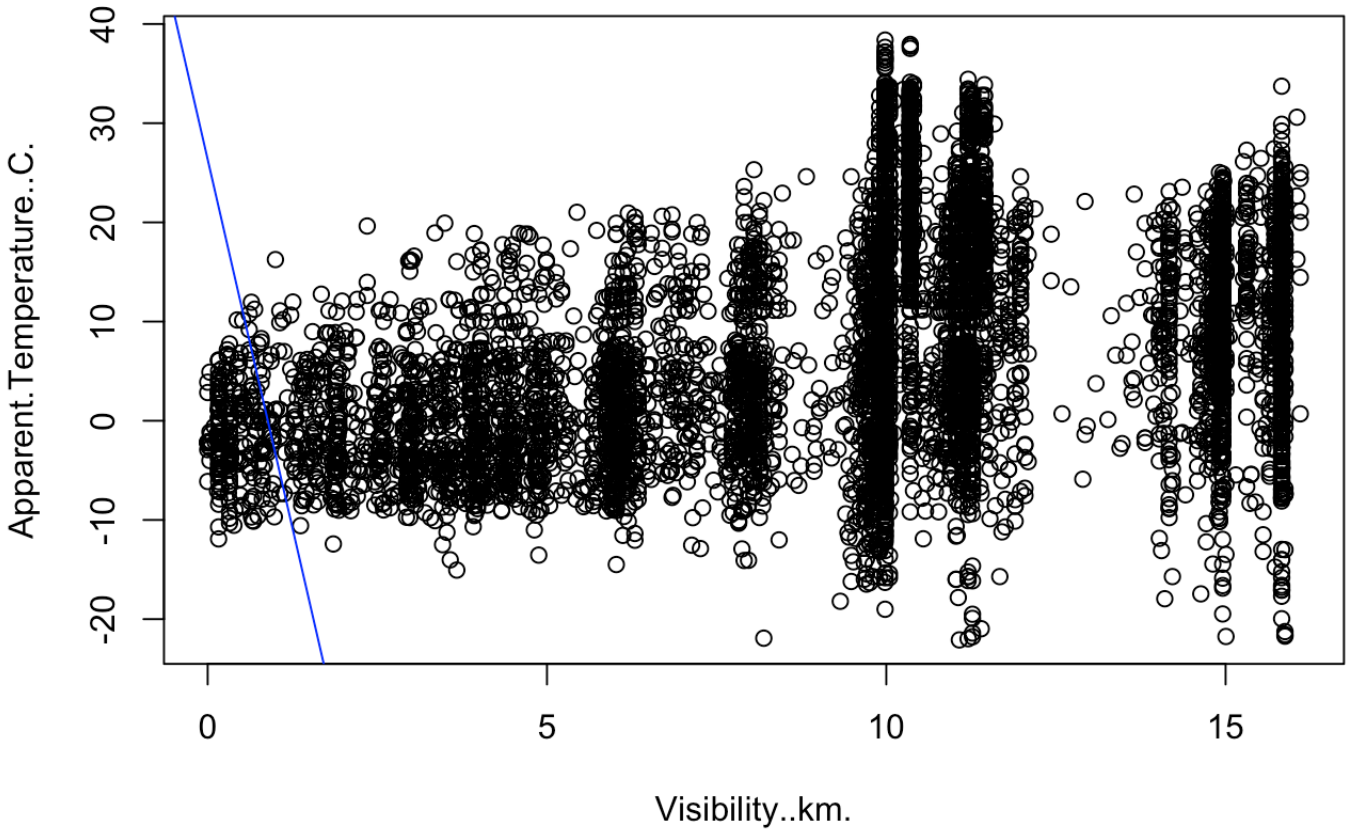
The predictors are humidity and visibility, target is Apparent Temperature. Residuals show the distance that a point is from the regression line and range from -40.460 to 19.983. The p value is very close to 0 which is good because we want a value less than 0.5 so the model shows a good goodness of fit. On the other hand R squared is 0.3711 which is not ideal because it should be close to 1. The RSE is in units of y so it shows that the model is off by about 8.706 degrees celsius.

```
plot(Apparent.Temperature..C.~Humidity+Visibility..km., data=train)
```




```
abline(lm(Apparent.Temperature..C.~Humidity+Visibility..km.), col="blue")
```

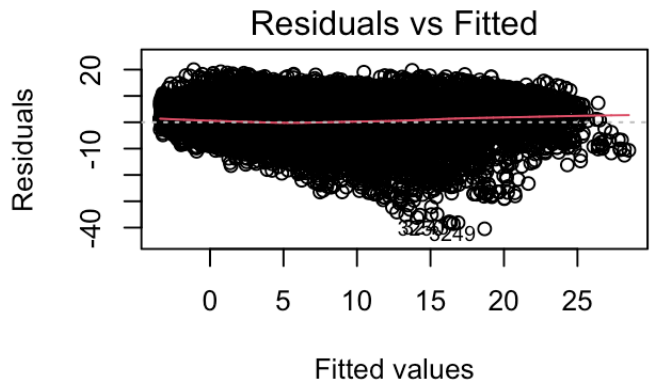
```
## Warning in abline(lm(Apparent.Temperature..C. ~ Humidity + Visibility..km.), :
## only using the first two of 3 regression coefficients
```

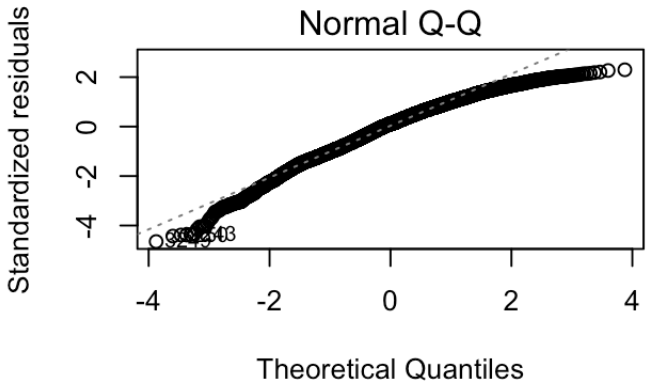


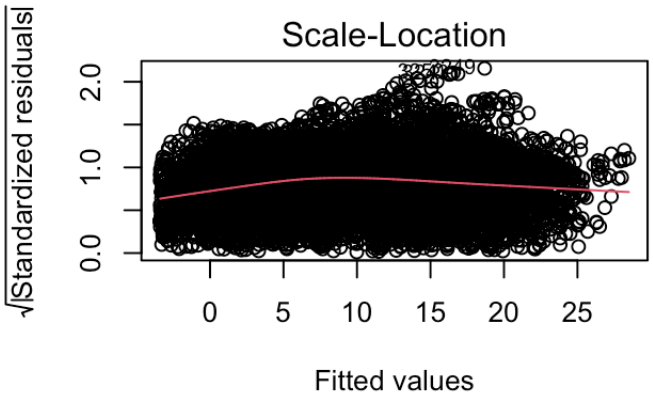

```
lm3 <- lm(Apparent.Temperature..C.~Humidity+Visibility..km., data=train)
summary(lm3)
```

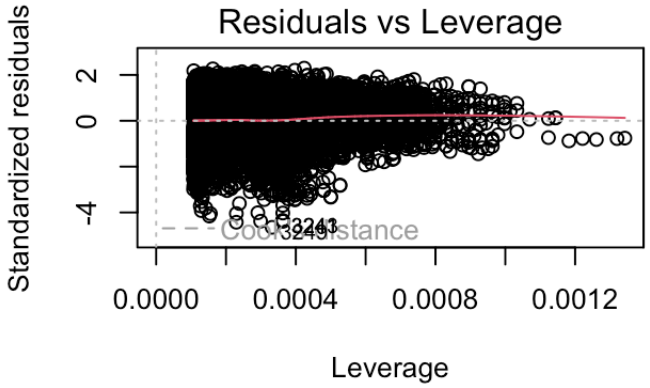
```
##
## Call:
## lm(formula = Apparent.Temperature..C. ~ Humidity + Visibility..km.,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.460  -5.847   0.969   6.452  19.983
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      26.4673     0.5229   50.62   <2e-16 ***
## Humidity        -29.8605     0.5129  -58.22   <2e-16 ***
## Visibility..km.   0.5810     0.0242   24.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.706 on 9404 degrees of freedom
## Multiple R-squared:  0.3711, Adjusted R-squared:  0.3709
## F-statistic:  2774 on 2 and 9404 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2));
plot(lm3)
```









h. comparing the results

The third plot gave the best results as although not ideal, its r squared was the highest at 3.711, whereas the first plot's was 0.3325 and second plot's was 0.3696. Also the RSE was the lowest at 8.706 compared to the first plot which was off by 8.968, and the second plot which was off by 8.716. The third plot could have performed the best because both humidity and visibility could have influenced apparent temperature more than humidity alone, or both humidity and wind speed together.

i.predict and evaluate on the test data. The higher the correlation the better the lower the mse the better.

The correlation for plot 3 is the highest at 0.6247 compared to plot 1 at 0.5865 and plot 2 at .6218, the higher the correlation better so plot 3 is the best. The mse for plot 3 is the lowest at 72.0802 compared to plot 1 at 77.5607 and plot 2 at 72.5182 which is good because the lower the mse the better. Overall plot 3 performed the best. These results happened because the data values using predictors humidity and visibility for plot 3 are closer to the mean and shows that the data values are more centralized and less skewed.

```
#simple linear regression
pred1 <- predict(lm1, newdata=test)
cor1 <- cor(pred1, test$Apparent.Temperature..C.)
mse1 <- mean((pred1-test$Apparent.Temperature..C.)^2)
rmse1 <- sqrt(mse1)
print(paste('correlation:', cor1))
```

```
## [1] "correlation: 0.586480474819058"
```

```
print(paste('mse:', mse1))
```

```
## [1] "mse: 77.5607588790075"
```

```
print(paste('rmse:', rmse1))
```

```
## [1] "rmse: 8.80685862717277"
```

```
#multiple linear regression
pred2 <- predict(lm2, newdata=test)
cor2 <- cor(pred2, test$Apparent.Temperature..C.)
mse2 <- mean((pred2-test$Apparent.Temperature..C.)^2)
rmse2 <- sqrt(mse2)
print(paste('correlation:', cor2))
```

```
## [1] "correlation: 0.621766081473664"
```

```
print(paste('mse:', mse2))
```

```
## [1] "mse: 72.5181634154257"
```

```
print(paste('rmse:', rmse2))
```

```
## [1] "rmse: 8.51575970864759"
```

```
#multiple linear regression
pred3 <- predict(lm3, newdata=test)
cor3 <- cor(pred3, test$Apparent.Temperature..C.)
mse3 <- mean((pred3-test$Apparent.Temperature..C.)^2)
rmse3 <- sqrt(mse3)
print(paste('correlation:', cor3))
```

```
## [1] "correlation: 0.624674165251754"
```

```
print(paste('mse:', mse3))
```

```
## [1] "mse: 72.0802442342089"
```

```
print(paste('rmse:', rmse3))
```

```
## [1] "rmse: 8.49000849435434"
```