Bayesian Modeling for Predicting Temperature: Identifying Key Weather Variables and their Influence

S/18/836 – W.M.C.C.M.Wijesingha

1. Introduction

Weather consists of multiple parameters, including air temperature, atmospheric (barometric) pressure, humidity, precipitation, solar radiation, and wind, all of which interact to define typical weather patterns and influence local atmospheric conditions. These parameters collectively shape environmental conditions, which can affect the surrounding ecosystem. Weather elements are interconnected, creating a chain reaction beyond just atmospheric changes. For example, interactions between temperature, pressure, and humidity can lead to cloud formation. In this study, we focus specifically on apparent temperature a measure of the temperature perceived by humans, which is influenced by a combination of air temperature, relative humidity, and wind speed. Apparent temperature is most commonly used to describe outdoor conditions. The aim of this study is to predict apparent temperature using Bayesian analysis, which will help identify the key factors influencing it. Based on this, we will develop a predictive model for apparent temperature.

2. Methodology

2.1. Data Set

The dataset includes weather data, containing variables such as date, temperature, apparent temperature, humidity, wind speed, visibility, pressure, and daily summary. Only numerical data were used in the analysis, with categorical variables removed.

2.2. Statistical Methods

Bayesian statistical techniques were applied, which involve updating probabilities as new data is collected. The following methods were used:

2.2.1. **Bayesian Simple Linear Regression**: A Bayesian approach to simple linear regression, using a reference prior on coefficients to provide a baseline for comparison with frequentist results.

2.2.2. **Bayesian Multiple Linear Regression**: Bayesian inference applied to multiple linear regression, with reference priors used to connect Bayesian and frequentist approaches.

2.2.3. **Bayesian Model Selection**: Model selection was based on the Bayesian Information Criterion (BIC), which helps in identifying the most suitable variables for the model.

2.3. Analysis Process

After determining correlations between dependent and independent variables, the best model was selected based on the results. The analysis was conducted in three steps:

2.3.1. **Bayesian Simple Linear Regression**: The model was fitted, and residuals were analyzed for normality and independence. Credible intervals for slope and intercept were calculated, showing equivalence to classical frequentist confidence intervals.

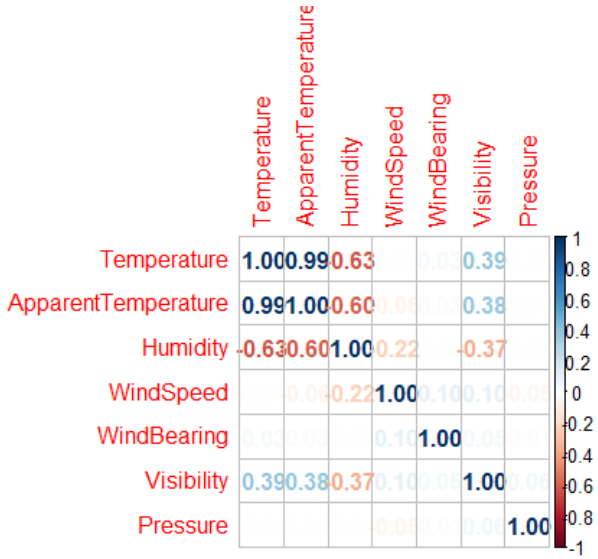
2.3.2. **Bayesian Multiple Linear Regression**: A multiple regression model was fitted, with posterior means, standard deviations, and credible intervals calculated for the coefficients.

2.3.3. **Bayesian Model Selection**: Variables were iteratively removed based on BIC, and the model with the lowest BIC was chosen. Posterior probabilities were then calculated.

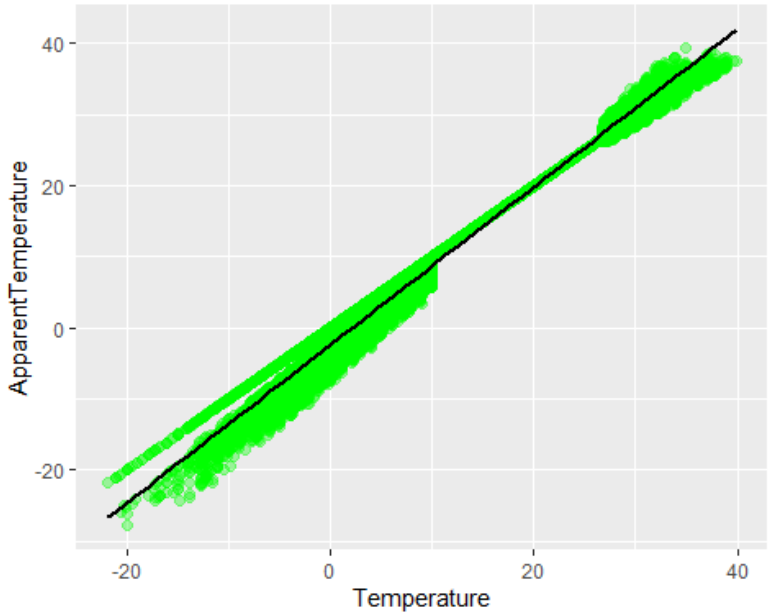
3. Results and Discussion

3.1. Correlation Coefficient

This figure illustrates the correlation between the numerical variables. From the graph, we observe the strongest positive correlation between temperature and apparent temperature. Additionally, there is a negative correlation between humidity and temperature, as well as a weak negative correlation between humidity, wind speed, and visibility. Based on these observations, we will focus on the relationship between temperature and apparent temperature, where temperature is the independent variable and apparent temperature is the dependent variable.



3.2. Bayesian Simple Linear Regression

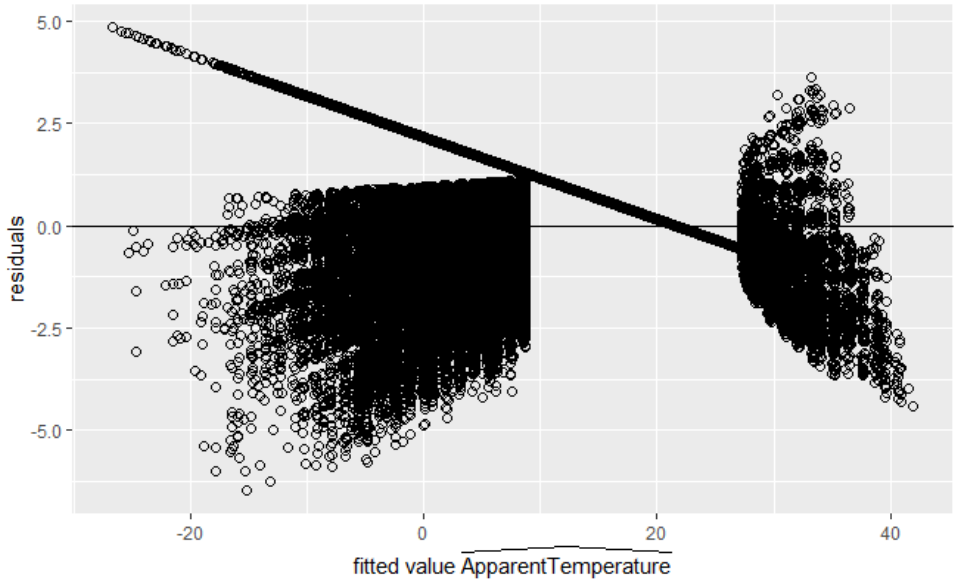


This dataset contains 96,453 observations of various weather parameters. In this analysis, we will develop a Bayesian simple linear regression model to predict apparent temperature based on temperature. Let yi(for i=1,2,...,96,453) represent the observed values of the response variable, apparent temperature, and let xi represent the corresponding temperature measurements.

The figure presents the relationship between apparent temperature and temperature for 96,453 observations. The model estimates a slope (β^) of 1.111652 and a y-intercept (α^) of approximately -2.409958, resulting in the following prediction equation:

$$\text{Apparent Temperature}\hat{} = -2.409958 + 1.111652 \times \text{Temperature}$$

This means that for every additional degree Celsius, the apparent temperature is expected to rise by 1.111652. Although the negative y-intercept does not align with physical expectations, particularly at a temperature of zero degrees Celsius, the linear regression model provides a reasonable approximation for predictions within the observed data range. Thus, it is effective for prediction purposes within this specific dataset.



Aside from one observation with the largest fitted value, the residual plot indicates that the linear regression model offers a reasonable approximation. The case number corresponding to the observation with the highest fitted value can be identified. Upon further investigation, this case is also found to have the highest apparent temperature, suggesting it could be a potential outlier.

3.3. **Bayesian Multiple linear Regression**

Let,

$Y_{at,i}$ = The i the apparent temperature

$X_{T,i}$ = The temperature of the i_{th} observation

$X_{H,i}$ = The Humidity of the i_{th} observation

$X_{WS,i}$=The wind speed of the i_{th} observation

$X_{WV,i}$=The wind bearing of i_{th} observation

$X_{v,i}$=The visibility of i_{th} observation

$X_{P,i}$=The pressure of i_{th} observation

n=The number of observations in this dataset

We set up the model as follows,

$Y_{at,I} = \alpha + \beta_1 X_{T,i} + \beta_2 X_{H,i} + \beta_3 X_{WS,i} + \beta_4 X_{WV,i} + \beta_5 X_{v,i} + \beta_6 X_{P,i} + \varepsilon_i$,i=1,2,…, n ---->(1)

For better analyses, one usually centers the variable, which ends up getting the following form.

$y_{at,I} = \beta_0 + \beta_1(X_{T,i} - \bar{\bar{x}}) + \beta_2(X_{H,i} - \bar{\bar{H}}) + \beta_3(X_{WS,i} - \bar{X}) + \beta_4(X_{WV,i} - \bar{WV}) + \beta_5(X_{v,i} - \bar{\bar{v}}) + \beta_6(X_{P,i} - \bar{\bar{p}}) + \varepsilon_i$

Under this transformation, the coefficients Under β1, β2, β3, β4, β5, β6 that are in front of the variables, are unchanged compared to the ones in (1). However, the constant coefficient β0 is no longer the constant coefficient α in (1). Instead, under the assumption that εi is independently, identically normal, β0 is the sample mean of the response variable Yat. This provides more meaning to β0 as this is the mean of Y when each of the predictors is equal to their respective means. Moreover, it is more convenient to use this "centered" model to derive analyses.

```
Marginal Posterior Summaries of Coefficients:

          Using  BMA

          Based on the top  1 models
                      post mean   post SD     post p(B != 0)
Intercept            1.086e+01   3.471e-03   1.000e+00
Temperature          1.126e+00   4.903e-04   1.000e+00
Humidity             1.032e+00   2.420e-02   1.000e+00
WindSpeed           -9.566e-02   5.286e-04   1.000e+00
WindBearing          5.315e-04   3.259e-05   1.000e+00
Visibility          -8.152e-04   9.183e-04   1.000e+00
Pressure             2.001e-04   2.979e-05   1.000e+00
```

In the final column of this summary, we observe that the probability of the coefficients being non-zero is consistently 1. This occurs because we used the argument include.always = ~ ., which forces the model to retain all variables. The posterior mean of β0 is 10.86, which differs significantly from the original y-intercept obtained under the frequentist OLS regression. Here, we are considering the "centered" model within the Bayesian framework. In this "centered" model with a reference prior, the posterior mean of the intercept β0 corresponds to the sample mean of the response variable Yat.

```
             posterior mean posterior std  2.5% 97.5%
Intercept             10.86          0.00 10.85 10.86
Temperature            1.13          0.00  1.12  1.13
Humidity               1.03          0.02  0.98  1.08
WindSpeed             -0.10          0.00 -0.10 -0.09
WindBearing            0.00          0.00  0.00  0.00
Visibility             0.00          0.00  0.00  0.00
Pressure               0.00          0.00  0.00  0.00
```

Temperature has highest credible interval. The credible intervals of the predictors WindBearing, Pressure and Visibility include 0, which implies that we may improve this model so that the model will accomplish a desired level of explanation or prediction with fewer predictors.

Bayesian Model Selection

This method mainly use Backward Elimination with BIC.

The Bayesian information criterion, BIC, is defined to be

BIC = -2ln ($likelihood$) + (p+1) ln(n)

n = number of observations in the model.

p = number of predictors

That is, p+1 is the number of total parameters (also the total number of coefficients, including the intercept) in the model. The model with the smallest BIC is preferrable. According to this table we can see that Full model-visibility has the minimum BIC value.

model	BIC
Full model	14554.45
Full model-Temperature	402207.5
Full model-Humidity	16343.49
Full model-WindSpeed	42738.03
Full model-WindBearing	14808.6
Full model-Visibility	14543.76
Full model-Pressure	14588.1

4. Conclusion and Recommendation

In this analysis, it can be concluded that apparent temperature is primarily influenced by temperature. Although the other variables have some degree of impact, visibility appears to be the least significant factor. In 2019, Mukul conducted a study on predicting apparent temperature using various weather parameters. His research mainly focused on identifying the key factors that affect apparent temperature.

5. References

1. Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis* (3rd ed.). Chapman and Hall/CRC.
2. Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences* (3rd ed.). Academic Press..
3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). Springer.
4. Lunn, D., Jackson, C., Best, N., Thomas, A., & Spiegelhalter, D. (2012). *The BUGS book: A practical introduction to Bayesian analysis*. CRC Press..
5. Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4), 583-639. https://doi.org/10.1111/1467-9868.00353

6. Appendices

6.1. Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	FormattedDate	Summary	PrecipType	Temperature	ApparentTem	Humidity	WindSpeed	WindBearir	Visibility	LoudCove	Pressure	DailySummary			
2	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222222	7.388888889	0.89	14.1197	251	15.8263	0	1015.13	Partly cloudy throughout the day.			
3	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355555556	7.227777778	0.86	14.2646	259	15.8263	0	1015.63	Partly cloudy throughout the day.			
4	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377777778	9.377777778	0.89	3.9284	204	14.9569	0	1015.94	Partly cloudy throughout the day.			
5	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288888889	5.944444444	0.83	14.1036	269	15.8263	0	1016.41	Partly cloudy throughout the day.			
6	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755555556	6.977777778	0.83	11.0446	259	15.8263	0	1016.51	Partly cloudy throughout the day.			
7	2006-04-01 05:00:00.000 +0200	Partly Cloudy	rain	9.222222222	7.111111111	0.85	13.9587	258	14.9569	0	1016.66	Partly cloudy throughout the day.			
8	2006-04-01 06:00:00.000 +0200	Partly Cloudy	rain	7.733333333	5.522222222	0.95	12.3648	259	9.982	0	1016.72	Partly cloudy throughout the day.			
9	2006-04-01 07:00:00.000 +0200	Partly Cloudy	rain	8.772222222	6.527777778	0.89	14.1519	260	9.982	0	1016.84	Partly cloudy throughout the day.			
10	2006-04-01 08:00:00.000 +0200	Partly Cloudy	rain	10.82222222	10.82222222	0.82	11.3183	259	9.982	0	1017.37	Partly cloudy throughout the day.			
11	2006-04-01 09:00:00.000 +0200	Partly Cloudy	rain	13.77222222	13.77222222	0.72	12.5258	279	9.982	0	1017.22	Partly cloudy throughout the day.			
12	2006-04-01 10:00:00.000 +0200	Partly Cloudy	rain	16.01666667	16.01666667	0.67	17.5651	290	11.2056	0	1017.42	Partly cloudy throughout the day.			
13	2006-04-01 11:00:00.000 +0200	Partly Cloudy	rain	17.14444444	17.14444444	0.54	19.7869	316	11.4471	0	1017.74	Partly cloudy throughout the day.			
14	2006-04-01 12:00:00.000 +0200	Partly Cloudy	rain	17.8	17.8	0.55	21.9443	281	11.27	0	1017.59	Partly cloudy throughout the day.			
15	2006-04-01 13:00:00.000 +0200	Partly Cloudy	rain	17.33333333	17.33333333	0.51	20.6885	289	11.27	0	1017.48	Partly cloudy throughout the day.			
16	2006-04-01 14:00:00.000 +0200	Partly Cloudy	rain	18.87777778	18.87777778	0.47	15.3755	262	11.4471	0	1017.17	Partly cloudy throughout the day.			
17	2006-04-01 15:00:00.000 +0200	Partly Cloudy	rain	18.91111111	18.91111111	0.46	10.4006	288	11.27	0	1016.47	Partly cloudy throughout the day.			
18	2006-04-01 16:00:00.000 +0200	Partly Cloudy	rain	15.38888889	15.38888889	0.6	14.4095	251	11.27	0	1016.15	Partly cloudy throughout the day.			

6.2. Codes

ST406

W.M.C.C.M.Wijesingha S/18/836

2024-10-13

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.3

## Warning: package 'ggplot2' was built under R version 4.1.3

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'purrr' was built under R version 4.1.3

## Warning: package 'dplyr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## Warning: package 'lubridate' was built under R version 4.1.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tinytex)
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.1.3

##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test

library(latex2exp)

## Warning: package 'latex2exp' was built under R version 4.1.3

library(skimr)

## Warning: package 'skimr' was built under R version 4.1.3

#install.packages("BAS")
library(BAS)

## Warning: package 'BAS' was built under R version 4.1.3

weather_data<-read_csv("../data/weatherHistory.csv")

## Rows: 96453 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): FormattedDate, Summary, PrecipType, DailySummary
## dbl (8): Temperature, ApparentTemperature, Humidity, WindSpeed, WindBearing,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

head(weather_data)

## # A tibble: 6 x 12
##   FormattedDate      Summary PrecipType Temperature ApparentTemperature Humidity
##   <chr>              <chr>   <chr>           <dbl>               <dbl>    <dbl>
## 1 2006-04-01 00:00:~ Partly~ rain             9.47                7.39     0.89
## 2 2006-04-01 01:00:~ Partly~ rain             9.36                7.23     0.86
## 3 2006-04-01 02:00:~ Mostly~ rain             9.38                9.38     0.89
## 4 2006-04-01 03:00:~ Partly~ rain             8.29                5.94     0.83
## 5 2006-04-01 04:00:~ Mostly~ rain             8.76                6.98     0.83
## 6 2006-04-01 05:00:~ Partly~ rain             9.22                7.11     0.85
## # i 6 more variables: WindSpeed <dbl>, WindBearing <dbl>, Visibility <dbl>,
## #   LoudCover <dbl>, Pressure <dbl>, DailySummary <chr>

colnames(weather_data)

##  [1] "FormattedDate"       "Summary"             "PrecipType"
##  [4] "Temperature"         "ApparentTemperature" "Humidity"
##  [7] "WindSpeed"           "WindBearing"         "Visibility"
## [10] "LoudCover"           "Pressure"            "DailySummary"

str(weather_data)

## spc_tbl_ [96,453 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ FormattedDate      : chr [1:96453] "2006-04-01 00:00:00.000 +0200" "2006-04-01 01:00:00.000 +0200" "
2006-04-01 02:00:00.000 +0200" "2006-04-01 03:00:00.000 +0200" ...
##  $ Summary            : chr [1:96453] "Partly Cloudy" "Partly Cloudy" "Mostly Cloudy" "Partly Cloudy" .
..
##  $ PrecipType         : chr [1:96453] "rain" "rain" "rain" "rain" ...
##  $ Temperature        : num [1:96453] 9.47 9.36 9.38 8.29 8.76 ...
##  $ ApparentTemperature: num [1:96453] 7.39 7.23 9.38 5.94 6.98 ...
##  $ Humidity           : num [1:96453] 0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.82 0.72 ...
##  $ WindSpeed          : num [1:96453] 14.12 14.26 3.93 14.1 11.04 ...
##  $ WindBearing        : num [1:96453] 251 259 204 269 259 258 259 260 259 279 ...
##  $ Visibility         : num [1:96453] 15.8 15.8 15 15.8 15.8 ...
```

```
##  $ LoudCover          : num [1:96453] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Pressure           : num [1:96453] 1015 1016 1016 1016 1017 ...
##  $ DailySummary       : chr [1:96453] "Partly cloudy throughout the day." "Partly cloudy throughout the
day." "Partly cloudy throughout the day." "Partly cloudy throughout the day." ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   FormattedDate = col_character(),
##   ..   Summary = col_character(),
##   ..   PrecipType = col_character(),
##   ..   Temperature = col_double(),
##   ..   ApparentTemperature = col_double(),
##   ..   Humidity = col_double(),
##   ..   WindSpeed = col_double(),
##   ..   WindBearing = col_double(),
##   ..   Visibility = col_double(),
##   ..   LoudCover = col_double(),
##   ..   Pressure = col_double(),
##   ..   DailySummary = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
numarical_data <- weather_data[, -c(1,2,3,10,12)]
colnames(numarical_data)
```

```
## [1] "Temperature"         "ApparentTemperature" "Humidity"
## [4] "WindSpeed"           "WindBearing"         "Visibility"
## [7] "Pressure"
```

```
head(numarical_data)
```

```
## # A tibble: 6 x 7
##   Temperature ApparentTemperature Humidity WindSpeed WindBearing Visibility
##         <dbl>               <dbl>    <dbl>     <dbl>       <dbl>      <dbl>
## 1        9.47                7.39     0.89     14.1          251       15.8
## 2        9.36                7.23     0.86     14.3          259       15.8
## 3        9.38                9.38     0.89      3.93         204       15.0
## 4        8.29                5.94     0.83     14.1          269       15.8
## 5        8.76                6.98     0.83     11.0          259       15.8
## 6        9.22                7.11     0.85     14.0          258       15.0
## # i 1 more variable: Pressure <dbl>
```

```
summary(numarical_data)
```

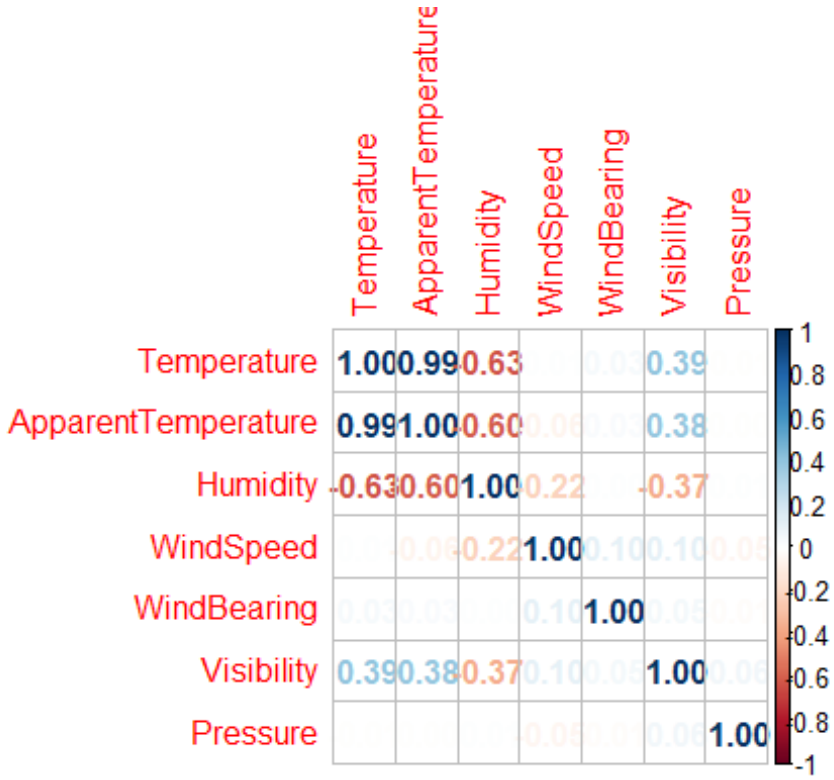
```
##   Temperature      ApparentTemperature    Humidity        WindSpeed
##  Min.   :-21.822   Min.   :-27.717     Min.   :0.0000   Min.   : 0.000
##  1st Qu.:  4.689   1st Qu.:  2.311     1st Qu.:0.6000   1st Qu.: 5.828
##  Median : 12.000   Median : 12.000     Median :0.7800   Median : 9.966
##  Mean   : 11.933   Mean   : 10.855     Mean   :0.7349   Mean   :10.811
##  3rd Qu.: 18.839   3rd Qu.: 18.839     3rd Qu.:0.8900   3rd Qu.:14.136
##  Max.   : 39.906   Max.   : 39.344     Max.   :1.0000   Max.   :63.853
##   WindBearing      Visibility       Pressure
##  Min.   :  0.0   Min.   : 0.00   Min.   :   0
##  1st Qu.:116.0   1st Qu.: 8.34   1st Qu.:1012
##  Median :180.0   Median :10.05   Median :1016
##  Mean   :187.5   Mean   :10.35   Mean   :1003
##  3rd Qu.:290.0   3rd Qu.:14.81   3rd Qu.:1021
##  Max.   :359.0   Max.   :16.10   Max.   :1046
```

```
library(corrplot)
```

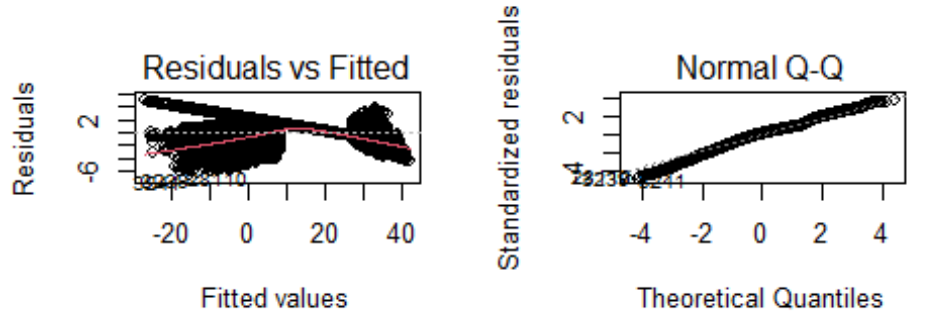
```
## Warning: package 'corrplot' was built under R version 4.1.3
```

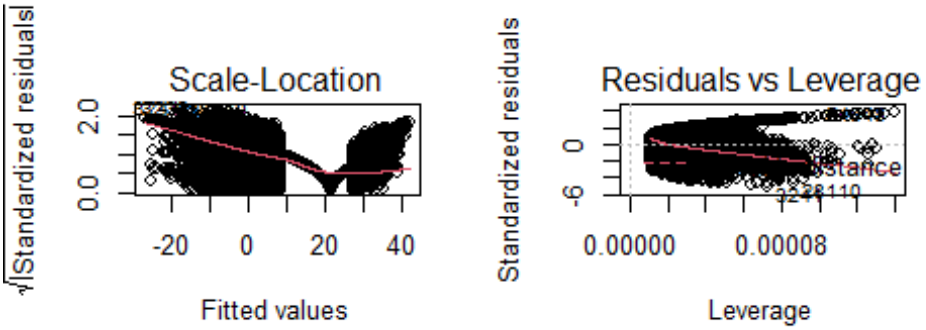
```
## corrplot 0.92 loaded
```

```
m <- cor(numarical_data)
m_round <- round(m, digits = 4)
corrplot(corr = m_round, method = "number")
```

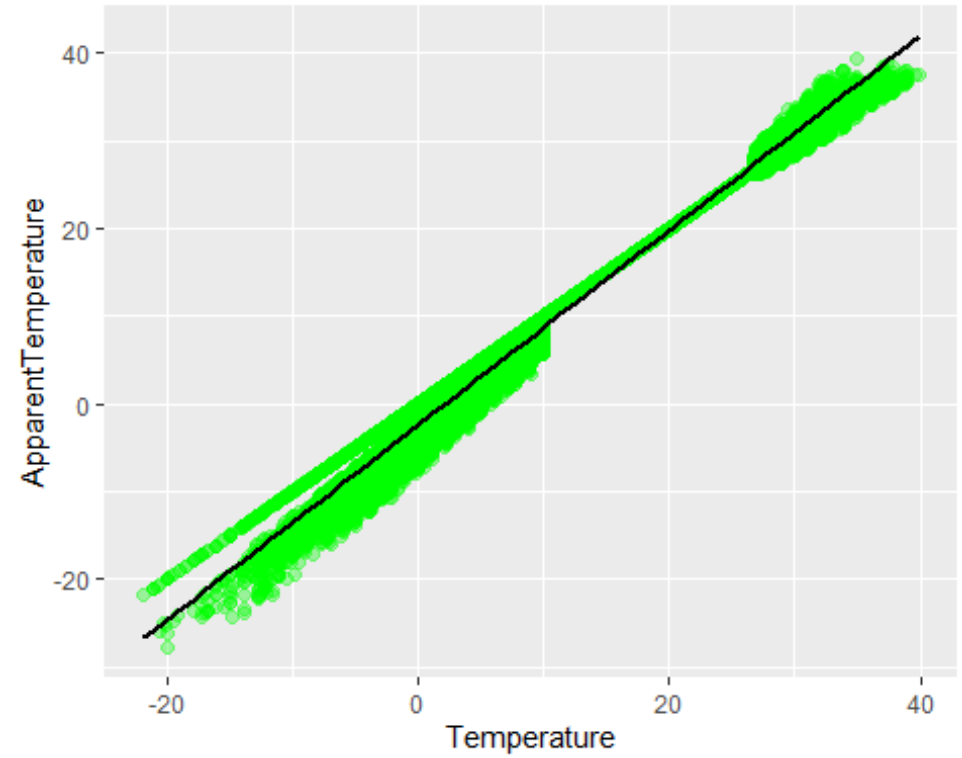



```
fit <- lm(formula = ApparentTemperature ~ Temperature, data = numarical_data)
summary(fit)

##
## Call:
## lm(formula = ApparentTemperature ~ Temperature, data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4857 -0.7137  0.1769  0.8357  4.8465
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.409958   0.006680  -360.8   <2e-16 ***
## Temperature  1.111652   0.000437  2543.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.296 on 96451 degrees of freedom
## Multiple R-squared:  0.9853, Adjusted R-squared:  0.9853
## F-statistic: 6.47e+06 on 1 and 96451 DF,  p-value: < 2.2e-16

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





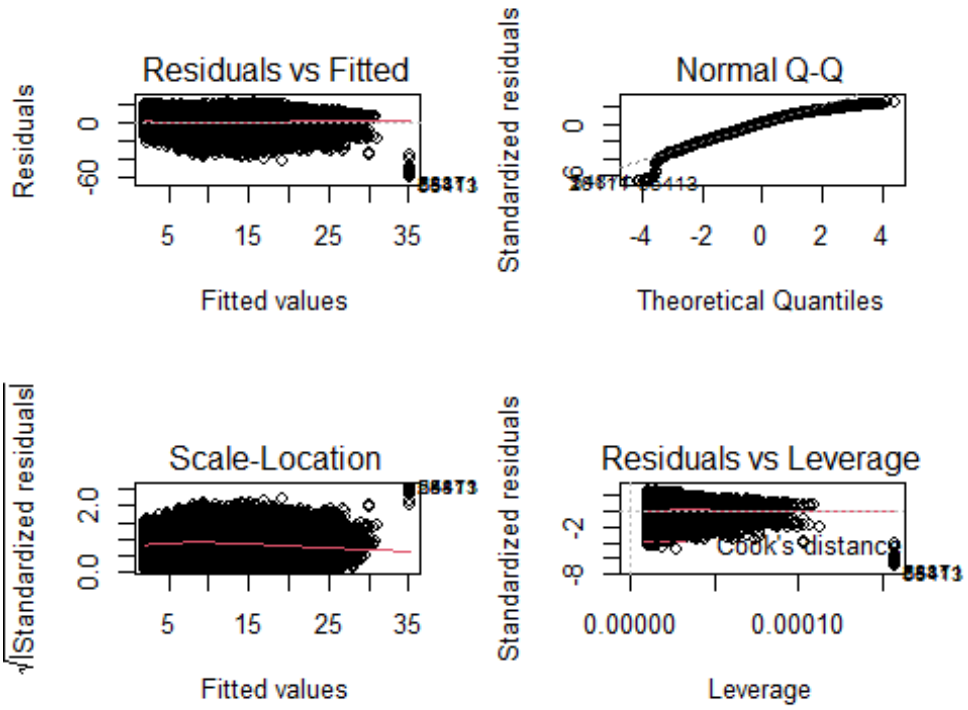

```
ggplot(data = numarical_data, aes(x = Temperature, y =ApparentTemperature )) + geom_point(color="green", a
lpha=0.35, size=2) + geom_smooth(method=lm, color="black")

## `geom_smooth()` using formula = 'y ~ x'
```



```
fit <- lm(formula = ApparentTemperature ~ Humidity, data = numarical_data)
summary(fit)
```

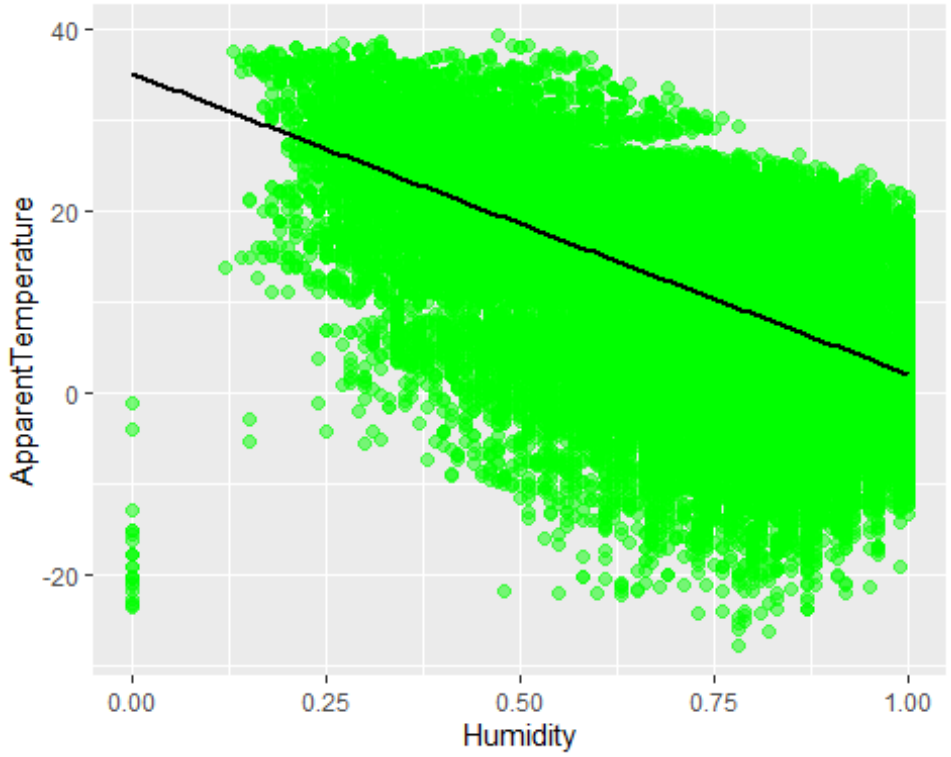
```
##
## Call:
## lm(formula = ApparentTemperature ~ Humidity, data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.705  -5.929   0.831   6.540  21.226
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.0879     0.1069   328.1   <2e-16 ***
## Humidity    -32.9745     0.1406  -234.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.537 on 96451 degrees of freedom
## Multiple R-squared:  0.3631, Adjusted R-squared:  0.3631
## F-statistic: 5.499e+04 on 1 and 96451 DF,  p-value: < 2.2e-16
```

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

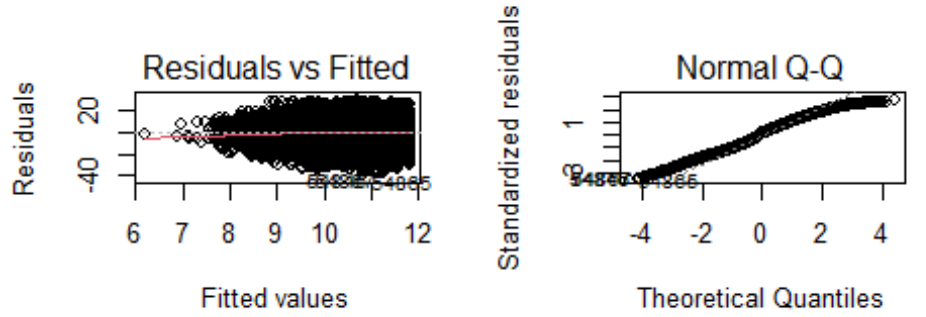


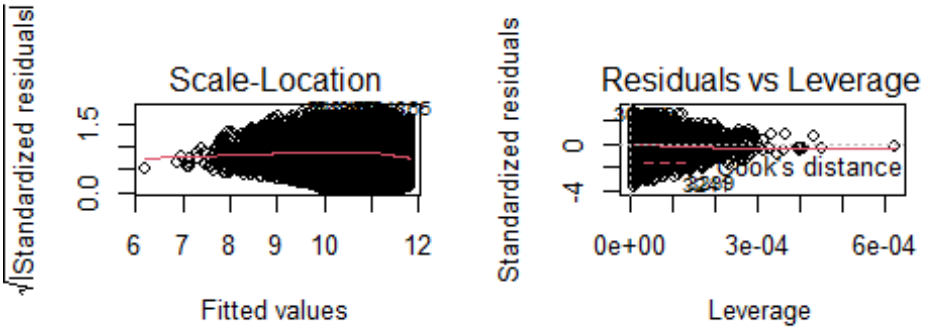

```
ggplot(data = numarical_data, aes(x = Humidity, y = ApparentTemperature)) + geom_point(color="green", alpha=0.5, size=2) + geom_smooth(method=lm, color="black")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
fit <- lm(formula = ApparentTemperature ~ WindSpeed, data = numarical_data)
summary(fit)

##
## Call:
## lm(formula = ApparentTemperature ~ WindSpeed, data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.531  -8.530   0.995   7.898  28.628
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.802579   0.063827  184.91   <2e-16 ***
## WindSpeed   -0.087650   0.004974  -17.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.68 on 96451 degrees of freedom
## Multiple R-squared:  0.003209,   Adjusted R-squared:  0.003199
## F-statistic: 310.5 on 1 and 96451 DF,  p-value: < 2.2e-16

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



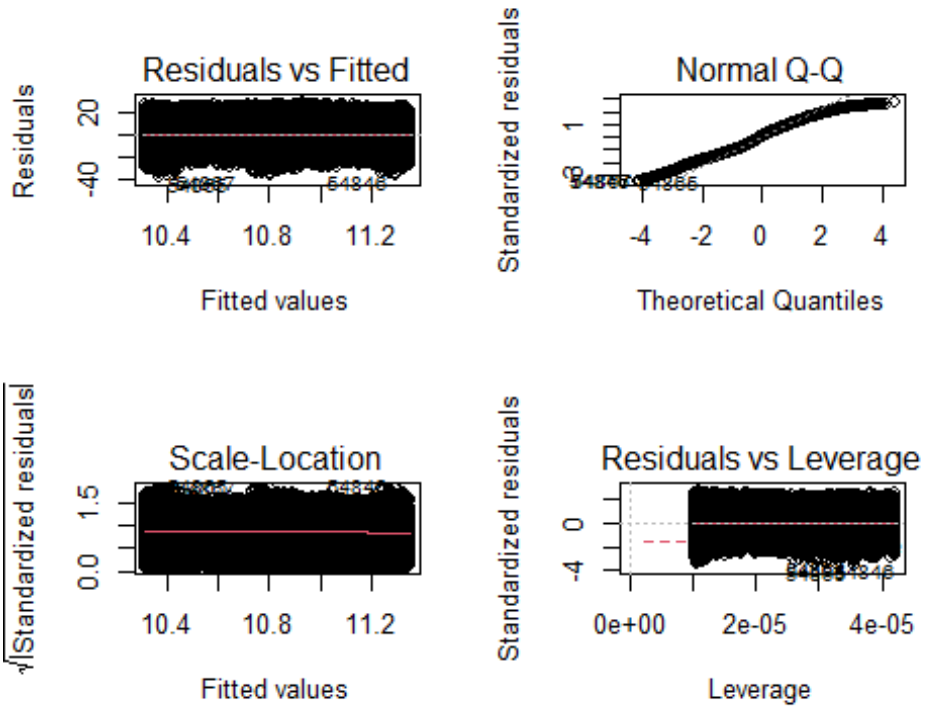



```
ggplot(data = numarical_data, aes(x = WindSpeed, y = ApparentTemperature)) + geom_point(color="green", alp
ha=0.5, size=2) + geom_smooth(method=lm, color="black")

## `geom_smooth()` using formula = 'y ~ x'
```

```
fit <- lm(formula = ApparentTemperature ~ WindBearing, data = numarical_data)
summary(fit)
```

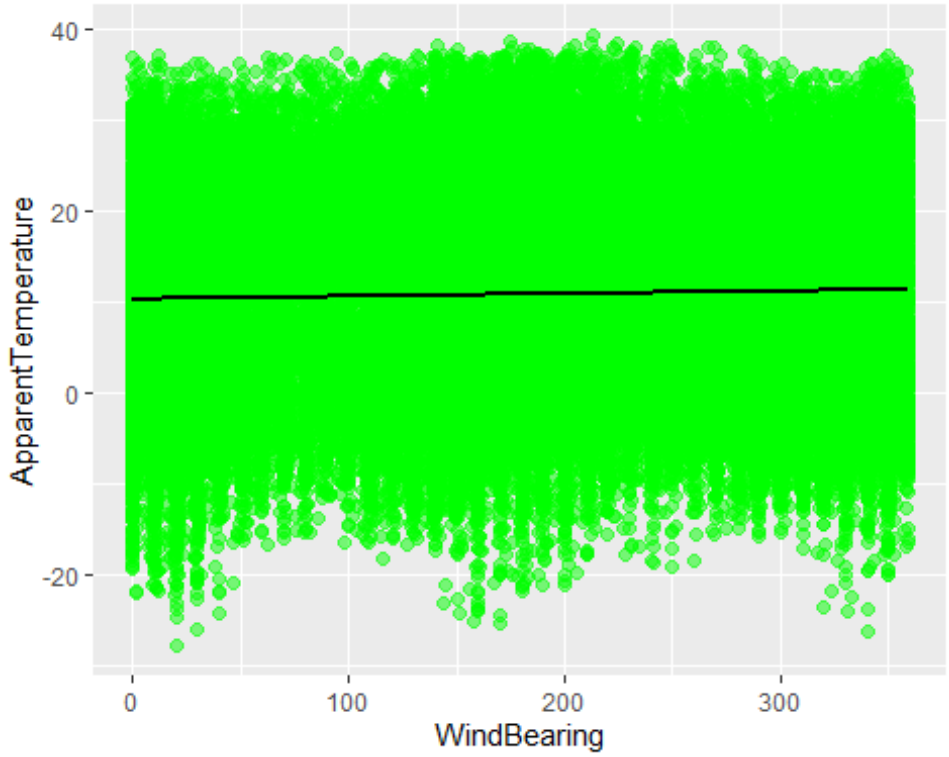
```
##
## Call:
## lm(formula = ApparentTemperature ~ WindBearing, data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.087  -8.524   0.964   7.935  28.416
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.031e+01  6.928e-02  148.86   <2e-16 ***
## WindBearing 2.892e-03  3.206e-04    9.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.69 on 96451 degrees of freedom
## Multiple R-squared:  0.0008428,  Adjusted R-squared:  0.0008324
## F-statistic: 81.35 on 1 and 96451 DF,  p-value: < 2.2e-16
```

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

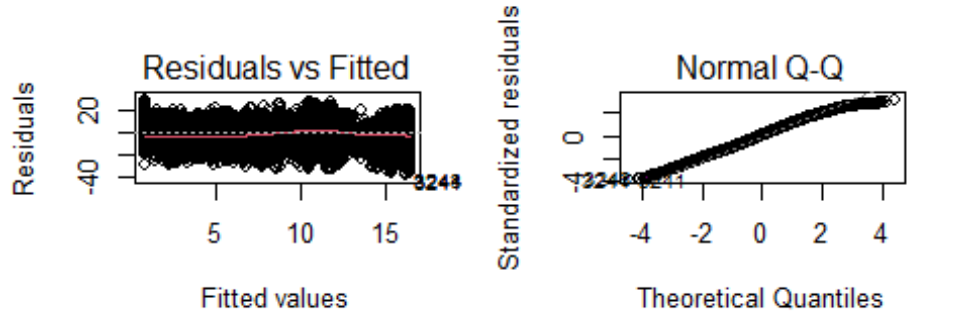


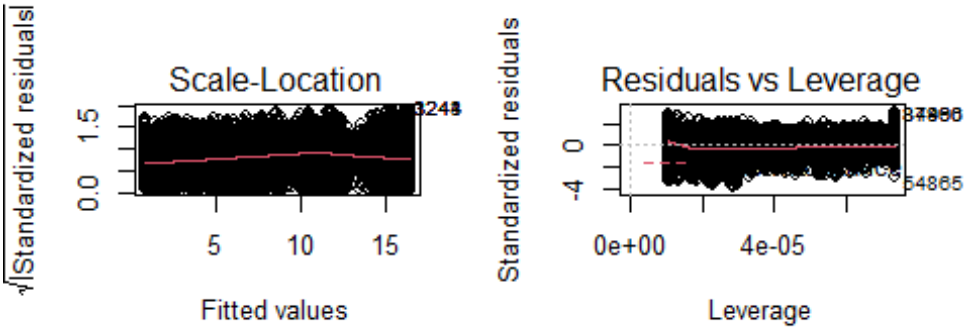

```
ggplot(data = numarical_data, aes(x = WindBearing, y = ApparentTemperature)) + geom_point(color="green", a
lpha=0.5, size=2) + geom_smooth(method=lm, color="black")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



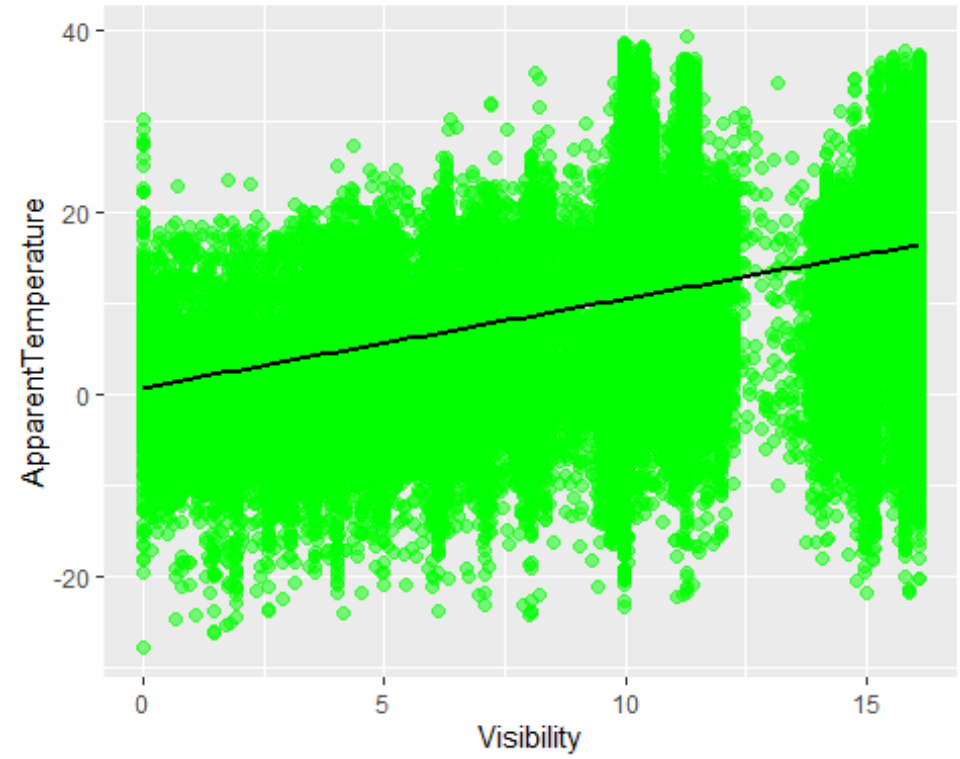

```
fit <- lm(formula = ApparentTemperature ~ Visibility, data = numarical_data)
summary(fit)

##
## Call:
## lm(formula = ApparentTemperature ~ Visibility, data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.955  -7.198   0.314   6.981  29.446
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.776596   0.084782    9.16   <2e-16 ***
## Visibility  0.974013   0.007594  128.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.887 on 96451 degrees of freedom
## Multiple R-squared:  0.1457, Adjusted R-squared:  0.1457
## F-statistic: 1.645e+04 on 1 and 96451 DF,  p-value: < 2.2e-16

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






```
ggplot(data = numarical_data, aes(x = Visibility, y = ApparentTemperature)) + geom_point(color="green", al
pha=0.5, size=2) + geom_smooth(method=lm, color="black")
```

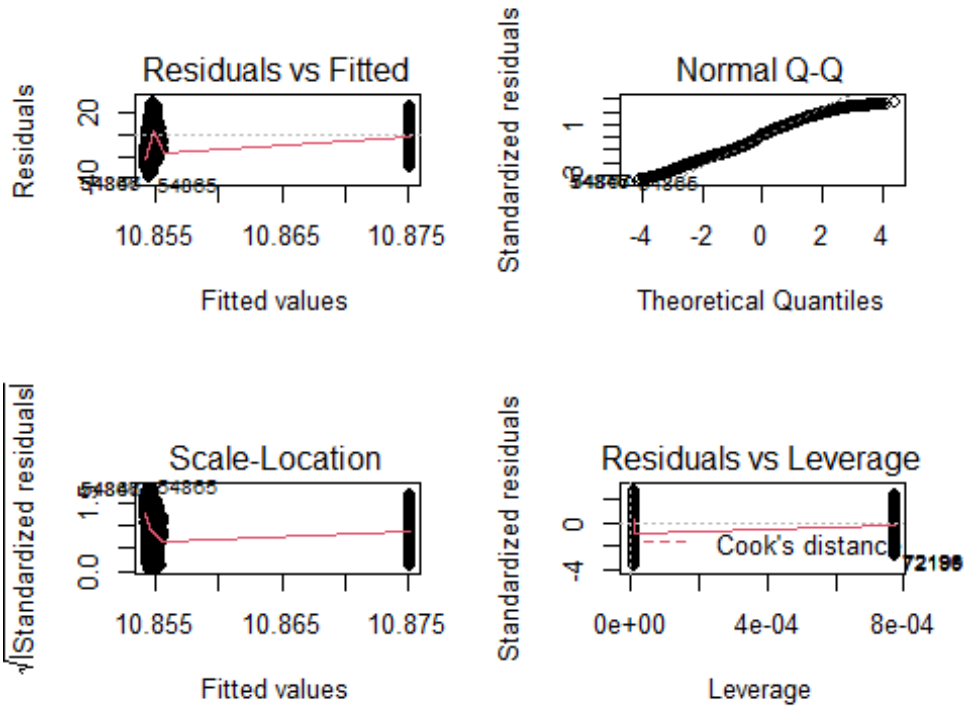
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
fit <- lm(formula = ApparentTemperature ~ Pressure, data = numarical_data)
summary(fit)
```

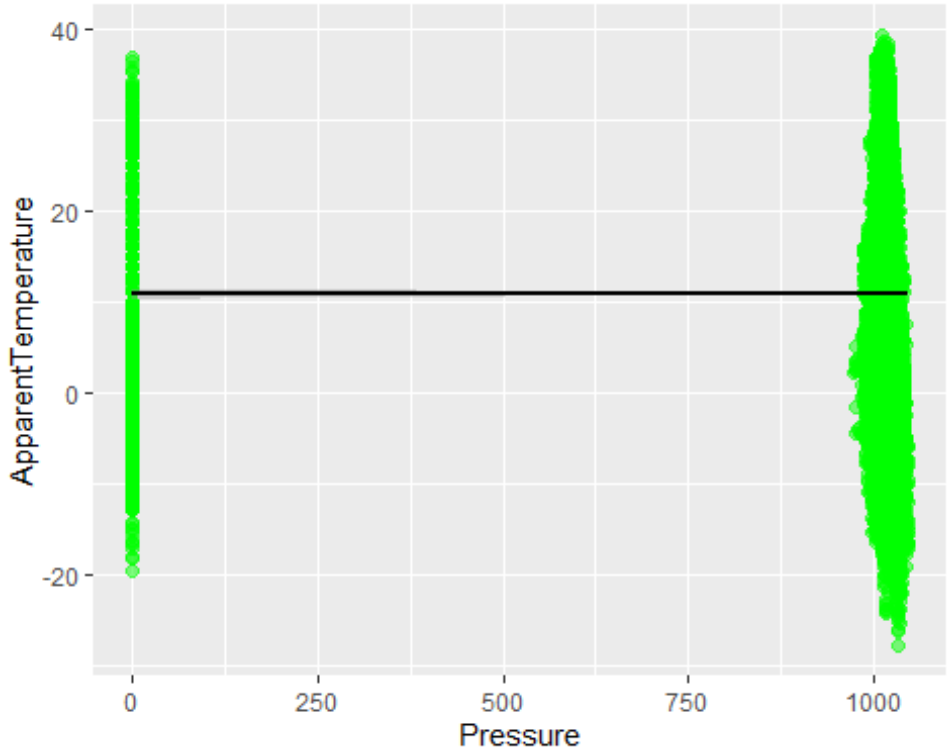
```
##
## Call:
## lm(formula = ApparentTemperature ~ Pressure, data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.571  -8.544   1.145   7.984  28.490
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.088e+01  2.974e-01  36.565   <2e-16 ***
## Pressure    -2.003e-05  2.945e-04  -0.068    0.946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.7 on 96451 degrees of freedom
## Multiple R-squared:  4.796e-08,  Adjusted R-squared:  -1.032e-05
## F-statistic: 0.004626 on 1 and 96451 DF,  p-value: 0.9458
```

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

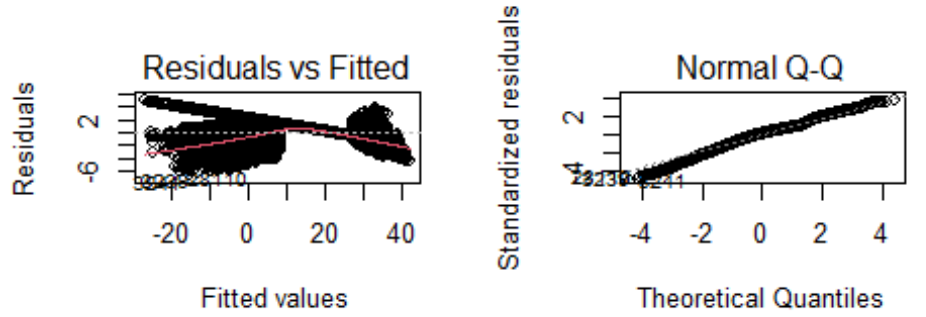


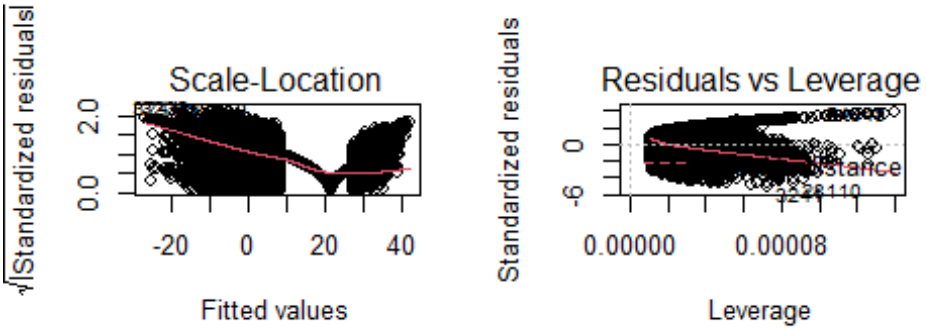

```
ggplot(data = numarical_data, aes(x = Pressure, y = ApparentTemperature)) + geom_point(color="green", alph
a=0.5, size=2) + geom_smooth(method=lm, color="black")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

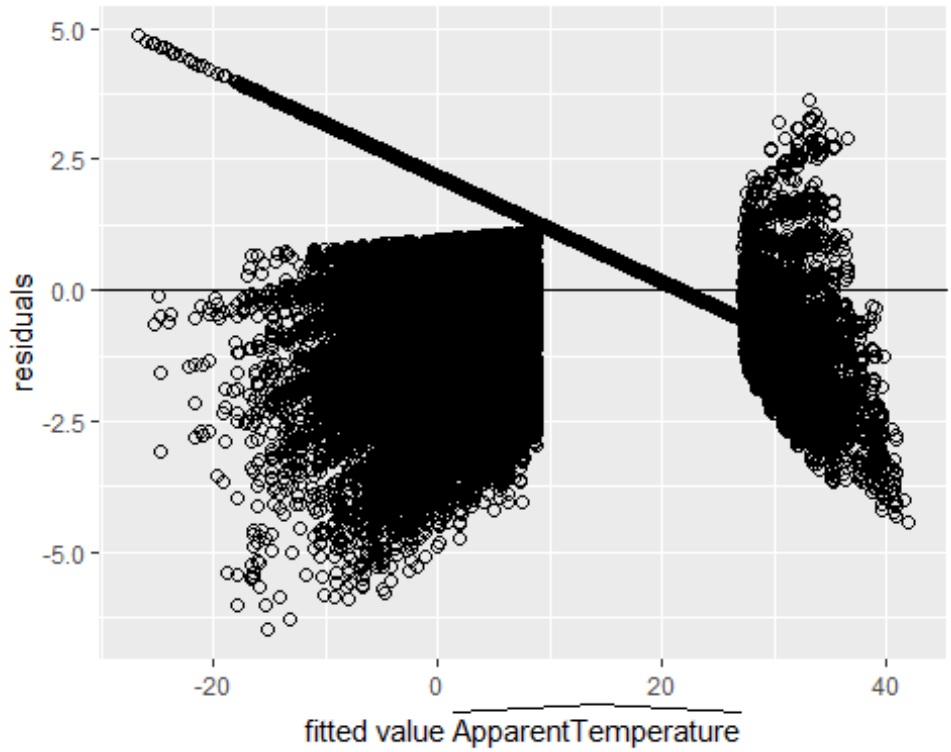



```
fit <- lm(formula = ApparentTemperature ~ Temperature, data = numarical_data)
summary(fit)

##
## Call:
## lm(formula = ApparentTemperature ~ Temperature, data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4857 -0.7137  0.1769  0.8357  4.8465
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.409958   0.006680  -360.8   <2e-16 ***
## Temperature  1.111652   0.000437  2543.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.296 on 96451 degrees of freedom
## Multiple R-squared:  0.9853, Adjusted R-squared:  0.9853
## F-statistic: 6.47e+06 on 1 and 96451 DF,  p-value: < 2.2e-16

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






```r
data_frame <- data.frame(fitted_values = fitted.values(fit), residuals = residuals(fit))

ggplot(data = data_frame, aes(x = fitted_values, y = residuals)) + geom_point(pch = 1, size = 2) + geom_ab
line(intercept = 0, slope = 0) +
xlab(expression(paste("fitted value ", widehat(ApparentTemperature)))) + ylab("residuals")
```




```r
fit <- lm(formula = ApparentTemperature ~ Temperature, data = numarical_data)

#Obtain residuals and n
resid = residuals(fit)
n = length(resid)

#Calculate MSE
MSE = 1/ (n-2) * sum((resid ^ 2))
MSE
```

```
## [1] 1.680717
```

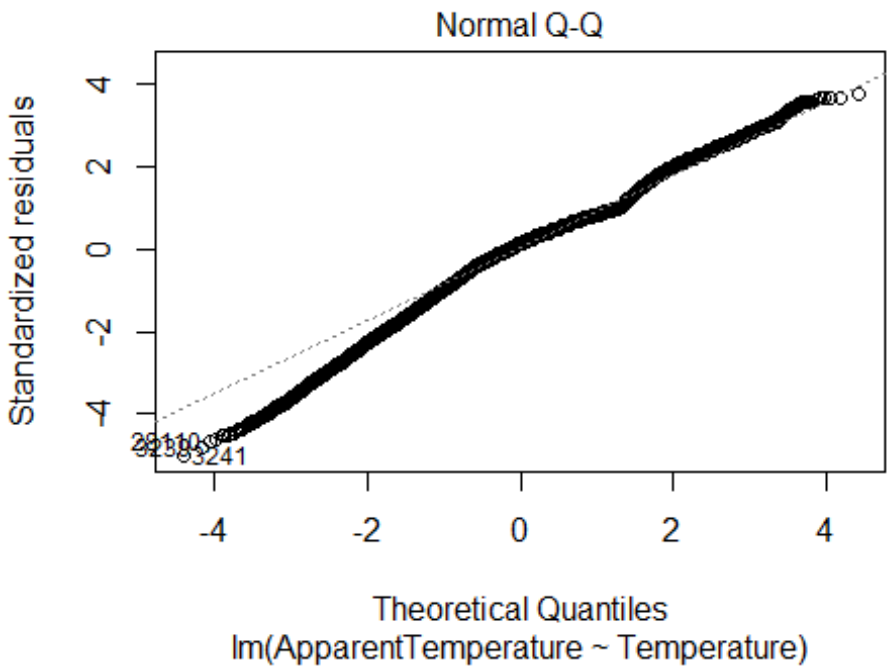
```r
#Combine residuals and fitted values into a data frame
result = data.frame(fitted_values = fitted.values(fit),residuals = residuals(fit))

#Find the observation with the largest fitted value
which.max(as.vector(fitted.values(fit)))
```

```
## [1] 12760
```

```r
#Shows this observation has the largest Temperature
which.max(numarical_data$Temperature)
```

```
## [1] 12760
```

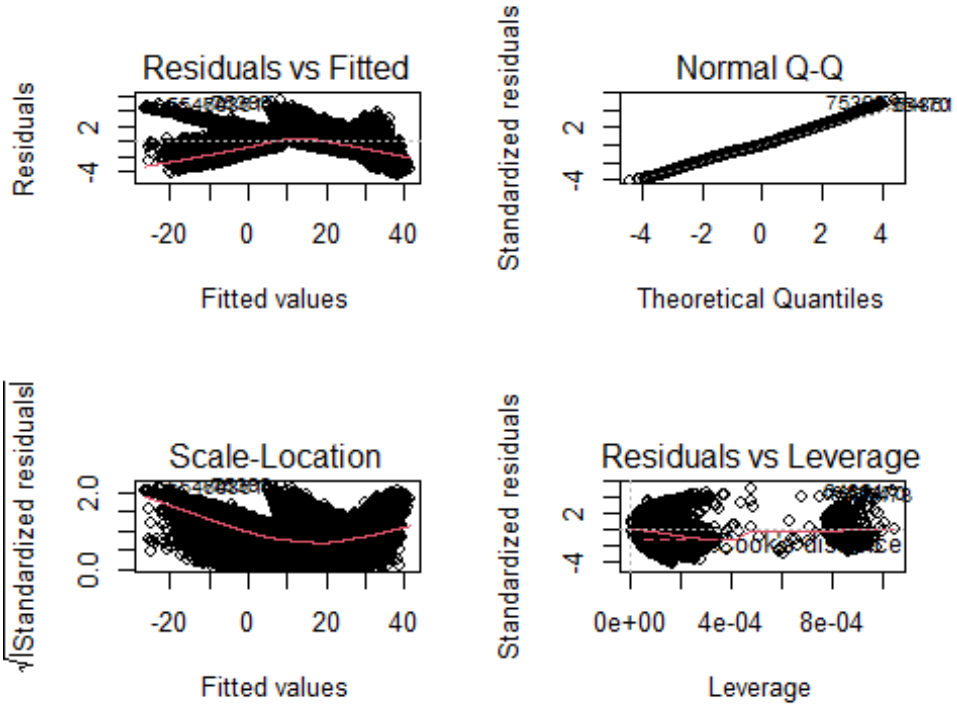
```r
#normal probability plot of the residuals
plot(fit, which = 2)
```




```
##Credible Intervals for Slope Beta and y-Intercept alpha
output = summary(fit)$coef[, 1:2]
output

##              Estimate   Std. Error
## (Intercept) -2.409958 0.0066799450
## Temperature  1.111652 0.0004370368

out <- cbind(output, confint(fit))
colnames(out) <- c("Posterior Mean", "Posterior Std", "2.5", "97.5")
round(out, 3)

##             Posterior Mean Posterior Std    2.5   97.5
## (Intercept)         -2.410         0.007 -2.423 -2.397
## Temperature          1.112         0.000  1.111  1.113

fit <- lm(formula = ApparentTemperature ~ ., data = numarical_data)
summary(fit)

##
## Call:
## lm(formula = ApparentTemperature ~ ., data = numarical_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3768 -0.7138 -0.1080  0.6832  5.3322
##
## Coefficients:
##                Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -2.593e+00  3.984e-02  -65.075  < 2e-16 ***
## Temperature  1.126e+00  4.903e-04 2295.962  < 2e-16 ***
## Humidity     1.032e+00  2.420e-02   42.630  < 2e-16 ***
## WindSpeed   -9.566e-02  5.286e-04 -180.960  < 2e-16 ***
## WindBearing  5.315e-04  3.259e-05   16.309  < 2e-16 ***
## Visibility  -8.152e-04  9.183e-04   -0.888    0.375
## Pressure     2.001e-04  2.979e-05    6.718 1.85e-11 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.078 on 96446 degrees of freedom
## Multiple R-squared:  0.9898, Adjusted R-squared:  0.9898
## F-statistic: 1.567e+06 on 6 and 96446 DF,  p-value: < 2.2e-16
```

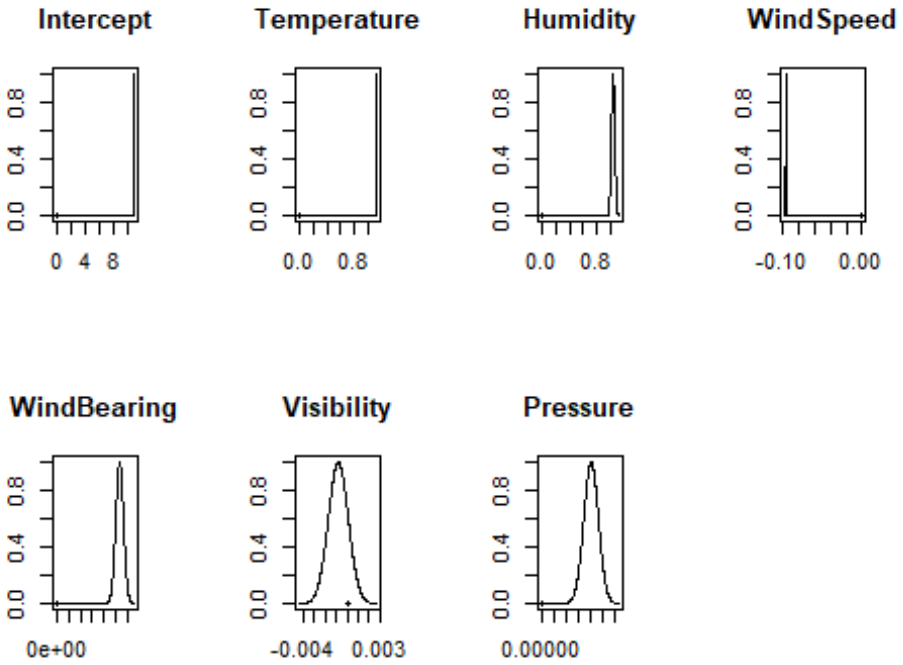
```
par(mfrow = c(2,2))
plot(fit)
```




```
bas.weather = bas.lm(ApparentTemperature ~ ., data = numarical_data, prior = "BIC", modelprior = Bernoulli
(1), include.always = ~ ., n.models = 1)
weather.coef <- coef(bas.weather)
weather.coef
```

```
##
##  Marginal Posterior Summaries of Coefficients:
##
##  Using  BMA
##
##  Based on the top  1 models
##              post mean   post SD     post p(B != 0)
## Intercept     1.086e+01   3.471e-03   1.000e+00
## Temperature   1.126e+00   4.903e-04   1.000e+00
## Humidity      1.032e+00   2.420e-02   1.000e+00
## WindSpeed    -9.566e-02   5.286e-04   1.000e+00
## WindBearing   5.315e-04   3.259e-05   1.000e+00
## Visibility   -8.152e-04   9.183e-04   1.000e+00
## Pressure      2.001e-04   2.979e-05   1.000e+00
```

```
par(mfrow = c(2, 4))
plot(weather.coef, ask = F)
```




```
#Summary table
out = confint(weather.coef)[, 1:2]
# Extract the upper and lower bounds of the credible intervals
names = c("posterior mean", "posterior std", colnames(out))
out = cbind(weather.coef$postmean, weather.coef$postsd, out)
colnames(out) = names
round(out, 2)

##             posterior mean posterior std  2.5% 97.5%
## Intercept            10.86          0.00 10.85 10.86
## Temperature           1.13          0.00  1.12  1.13
## Humidity              1.03          0.02  0.98  1.08
## WindSpeed            -0.10          0.00 -0.10 -0.09
## WindBearing           0.00          0.00  0.00  0.00
## Visibility            0.00          0.00  0.00  0.00
## Pressure              0.00          0.00  0.00  0.00

# Total number of observation
n <- nrow(numarical_data)

# Full Model
wea.lm1 <- lm(ApparentTemperature ~ ., data = numarical_data)
wea.step <- step(wea.lm1, k = log(n))

## Start:  AIC=14554.45
## ApparentTemperature ~ Temperature + Humidity + WindSpeed + WindBearing +
##     Visibility + Pressure
##
##               Df Sum of Sq     RSS    AIC
## - Visibility   1         1  112070  14544
## <none>                     112070  14554
## - Pressure     1        52  112122  14588
## - WindBearing  1       309  112379  14809
## - Humidity     1      2112  114181  16343
## - WindSpeed    1     38051  150121  42738
## - Temperature  1   6125376 6237445 402207
```

```
##
## Step:  AIC=14543.76
## ApparentTemperature ~ Temperature + Humidity + WindSpeed + WindBearing +
##     Pressure
##
##               Df Sum of Sq     RSS    AIC
## <none>                      112070  14544
## - Pressure     1        52  112122  14577
## - WindBearing  1       308  112379  14797
## - Humidity     1      2172  114243  16384
## - WindSpeed    1     38212  150282  42830
## - Temperature  1   6458673 6570744 407217

#Full model-Visibility
wea.lmt <- lm(ApparentTemperature ~ Temperature+Humidity+WindSpeed+WindBearing+Pressure,data = numarical_d
ata)
wea.step <- step(wea.lmt, k = log(n))

## Start:  AIC=14543.76
## ApparentTemperature ~ Temperature + Humidity + WindSpeed + WindBearing +
##     Pressure
##
##               Df Sum of Sq     RSS    AIC
## <none>                      112070  14544
## - Pressure     1        52  112122  14577
## - WindBearing  1       308  112379  14797
## - Humidity     1      2172  114243  16384
## - WindSpeed    1     38212  150282  42830
## - Temperature  1   6458673 6570744 407217

#Full model-WindBearing
wea.lmt <- lm(ApparentTemperature ~ Temperature+Humidity+WindSpeed+Pressure+Visibility,data = numarical_da
ta)
wea.step <- step(wea.lmt, k = log(n))

## Start:  AIC=14808.6
## ApparentTemperature ~ Temperature + Humidity + WindSpeed + Pressure +
##     Visibility
##
##               Df Sum of Sq     RSS    AIC
## - Visibility   1         0  112379  14797
## <none>                      112379  14809
## - Pressure     1        50  112429  14840
## - Humidity     1      2222  114601  16686
## - WindSpeed    1     37761  150140  42739
## - Temperature  1   6144511 6256890 402496
##
## Step:  AIC=14797.2
## ApparentTemperature ~ Temperature + Humidity + WindSpeed + Pressure
##
##               Df Sum of Sq     RSS    AIC
## <none>                      112379  14797
## - Pressure     1        50  112429  14829
## - Humidity     1      2275  114653  16718
## - WindSpeed    1     37928  150307  42835
## - Temperature  1   6486816 6599194 407622

#Full model-WindSpeed
wea.lmt <- lm(ApparentTemperature ~ Temperature+Humidity+Pressure+WindBearing+Visibility,data = numarical_
data)
wea.step <- step(wea.lmt, k = log(n))

## Start:  AIC=42738.03
## ApparentTemperature ~ Temperature + Humidity + Pressure + WindBearing +
##     Visibility
##
##               Df Sum of Sq     RSS    AIC
## <none>                      150121  42738
```

```
## - WindBearing  1        19  150140  42739
## - Visibility   1       161  150282  42830
## - Pressure     1       315  150435  42929
## - Humidity     1     10729  160850  49385
## - Temperature  1   6549433 6699553 409090

#Full model-Humidity
wea.lmt <- lm(ApparentTemperature ~ Temperature+Pressure+WindSpeed+WindBearing+Visibility,data = numarical
_data)
wea.step <- step(wea.lmt, k = log(n))

## Start:  AIC=16343.49
## ApparentTemperature ~ Temperature + Pressure + WindSpeed + WindBearing +
##     Visibility
##
##               Df Sum of Sq     RSS    AIC
## <none>                      114181  16343
## - Pressure     1        52  114233  16376
## - Visibility   1        61  114243  16384
## - WindBearing  1       419  114601  16686
## - WindSpeed    1     46668  160850  49385
## - Temperature  1   9199701 9313883 440867

#Full model-Temperature
wea.lmt <- lm(ApparentTemperature ~ Pressure+Humidity+WindSpeed+WindBearing+Visibility,data = numarical_da
ta)
wea.step <- step(wea.lmt, k = log(n))

## Start:  AIC=402207.5
## ApparentTemperature ~ Pressure + Humidity + WindSpeed + WindBearing +
##     Visibility
##
##               Df Sum of Sq     RSS    AIC
## <none>                     6237445 402207
## - Pressure     1      3632 6241077 402252
## - WindBearing  1     19444 6256890 402496
## - Visibility   1    333298 6570744 407217
## - WindSpeed    1    462108 6699553 409090
## - Humidity     1   3076437 9313883 440867

# Model
bas_model <- bas.lm(ApparentTemperature ~ ., data = numarical_data, prior = "BIC", modelprior = uniform())
#bas_model
bas_coef <- coef(bas_model)
bas_coef

##
##  Marginal Posterior Summaries of Coefficients:
##
##  Using  BMA
##
##  Based on the top  64 models
##              post mean   post SD     post p(B != 0)
## Intercept     1.086e+01   3.471e-03   1.000e+00
## Temperature   1.125e+00   4.775e-04   1.000e+00
## Humidity      1.035e+00   2.394e-02   1.000e+00
## WindSpeed    -9.568e-02   5.277e-04   1.000e+00
## WindBearing   5.304e-04   3.257e-05   1.000e+00
## Visibility   -3.874e-06   8.456e-05   4.752e-03
## Pressure      1.982e-04   2.971e-05   1.000e+00

# Best model
best <- which.max(bas_model$logmarg)
bestmodel <- bas_model$which[[best]]
bestmodel

## [1] 0 1 2 3 4 6
```

```r
bestgamma <- rep(0, bas_model$n.vars)
bestgamma[bestmodel + 1] <- 1
bestgamma
```

```
## [1] 1 1 1 1 1 0 1
```

```r
bas_bestmodel <- bas.lm(ApparentTemperature ~ Temperature+Humidity+WindSpeed+WindBearing+Pressure, data =
numarical_data, prior = "BIC", n.models = 1, bestmodel = bestgamma, modelprior = uniform())

## Coefficients
bas_coef <- coef(bas_bestmodel)
## credible intervals
out <- confint(bas_coef)[, 1:2]

## Summary table
bas_summary <- cbind(bas_coef$postmean, bas_coef$postsd, out)
names <- c("Posterior Mean", "Posterior SD", colnames(out))
colnames(bas_summary) <- names
bas_summary
```

```
##             Posterior Mean Posterior SD          2.5%         97.5%
## Intercept    10.8550288742 0.0034716929 10.8482243956 10.8618333527
## Temperature   1.1254628259 0.0004774694  1.1245269914  1.1263986604
## Humidity      1.0331213485 0.0239408986  0.9861974607  1.0800452363
## WindSpeed    -0.0958594092 0.0005271231 -0.0968925645 -0.0948262538
## WindBearing   0.0005291495 0.0000325733  0.0004653062  0.0005929928
## Pressure      0.0000000000 0.0000000000  0.0000000000  0.0000000000
```

```r
## Get posterior probability
bas_model <- bas.lm(ApparentTemperature ~ Temperature+Humidity+WindSpeed+WindBearing+Pressure, data = numa
rical_data, prior = "BIC", modelprior = uniform())
round(summary(bas_model), 3)
```

```
##             P(B != 0 | Y)    model 1    model 2    model 3    model 4   model 5
## Intercept               1       1.00       1.00       1.00       1.00       1.0
## Temperature             1       1.00       1.00       1.00       1.00       0.0
## Humidity                1       1.00       1.00       1.00       1.00       0.0
## WindSpeed               1       1.00       1.00       1.00       1.00       0.0
## WindBearing             1       1.00       1.00       0.00       0.00       0.0
## Pressure                1       1.00       0.00       1.00       0.00       0.0
## BF                     NA       1.00       0.00       0.00       0.00       0.0
## PostProbs              NA       1.00       0.00       0.00       0.00       0.0
## R2                     NA       0.99       0.99       0.99       0.99       0.0
## dim                    NA       6.00       5.00       5.00       4.00       1.0
## logmarg                NA -560758.31 -560774.82 -560885.03 -560900.86 -782080.4
```

```r
## Marginal posterior inclusion probability
print(bas_model)
```

```
##
## Call:
## bas.lm(formula = ApparentTemperature ~ Temperature + Humidity +
##     WindSpeed + WindBearing + Pressure, data = numarical_data,
##     prior = "BIC", modelprior = uniform())
##
##
##  Marginal Posterior Inclusion Probabilities:
##   Intercept  Temperature     Humidity    WindSpeed  WindBearing     Pressure
##           1            1            1            1            1            1
```