**Report**

**Introduction**

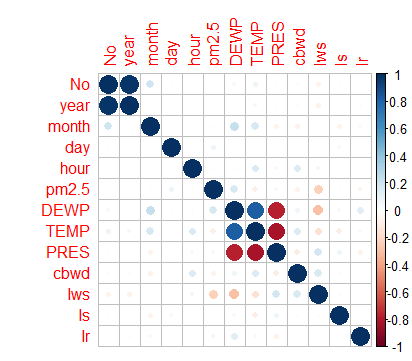
This report was prepared to analyze the year wise temperature related data in an area. The dataset contained 13 attributes which are No (like row number), year, month, day, hour, pm2.5 (ug/m^3), DEWP, TEMP, PRES (hpa), cbwd (Combined Wind Direction), Iws (Cumulated Wind Speed m/s), Is (Cumulated hours of snow), Ir (Cumulated hours of rain). The following packages are installed and loaded for this assignment:

* shiny
* corrplot
* ggplot2
* gridExtra

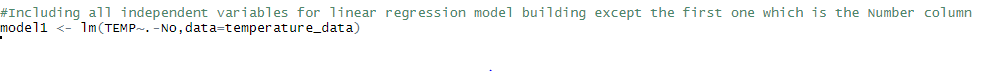
We observed that cbwd (Combined Wind direction) was a categorical variable, so before doing the corrplot, it has been encoded as the numerical variable.

**1:**

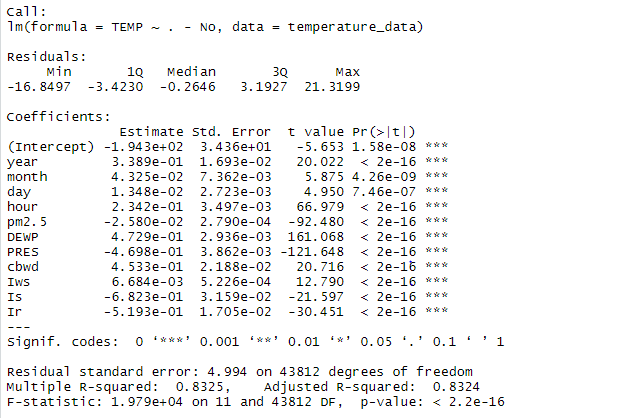
Firstly, the correlation between the variables was found plotting the correlation using the corrplot package.



Then, Linear Regression models have been built using lm() function in R between TEMP (Temperature) as y variable and many combinations of other independent variables. Four linear models has been built which are as follows:

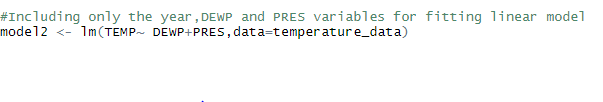


We are getting an R-squared value of 0.8325 with all x variables and intercept pretty significant and the p-value (2.2e-16) is also very less than 0.05. But the complexity of the model is pretty high as we have 11 independent variables.

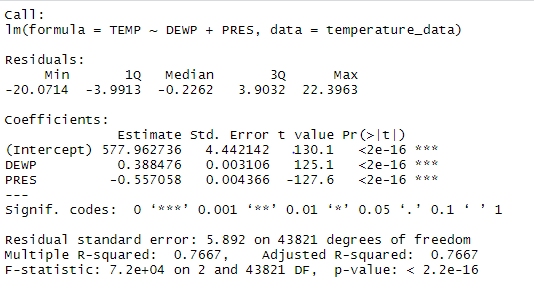


Now, we are building the model for TEMP as y variable and Dew Point and Pressure as x variables

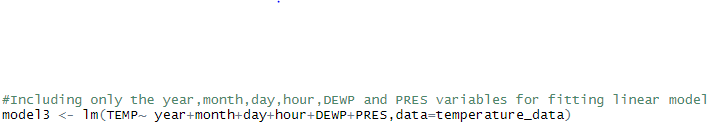
Because we had found strong positive correlation and strong negative correlation for them from the corrplot.



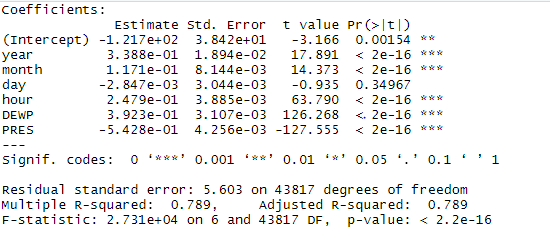
We are getting an R-squared value of 0.7667 with all x variables and intercept pretty significant. But now the complexity of the model is very less as we have only independent variables involved. The p-value (2.2e-16) is also less than 0.05 indicating the model significance. There is a trade-off between R-squared value and model complexity. In this case, even though the R-squared value is less compared to our previous model, the complexity also greatly decreases.



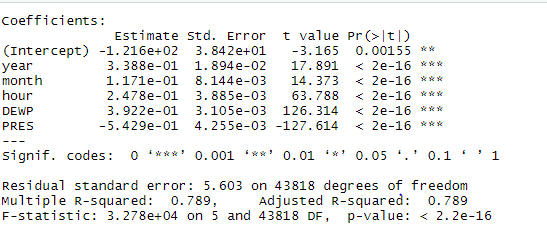
Then, we went upon with fitting a linear model between TEMP with time data (year, month, day, hour) and Dew Point and Pressure to check whether the R-squared and model significance increases with this combination.



Here, it has been found that the R-squared value (0.789) was increased compared to the previous model but the day variable was insignificant. So, we tried to remove the day variable in our next model and check for improvement in R-squared and model significance.







Even after removing the day variable, the R-squared value remains the same with rest of the variables being significant.

Iws, Is and Ir have not been fitted in the linear model with TEMP as it has been observed from the

Corrplot that they were not strongly correlated with Temperature.

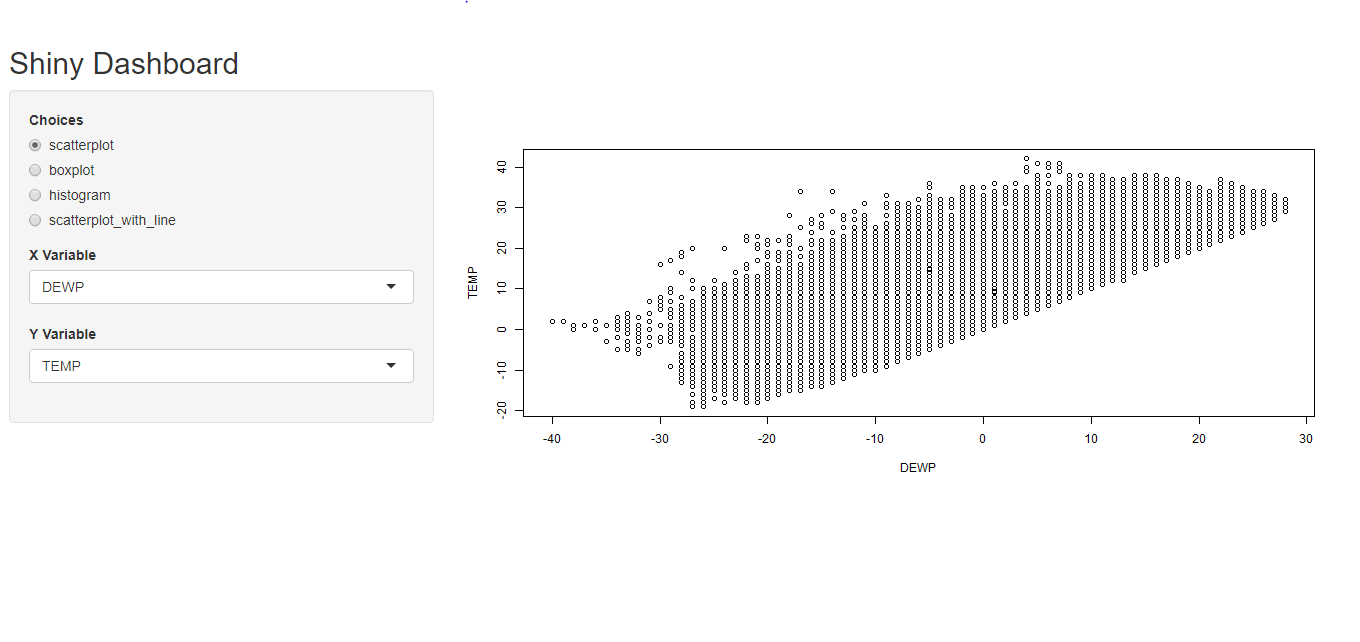
Therefore after building all the above models, it has been found that model 1 was our best model in terms of R-squared value.

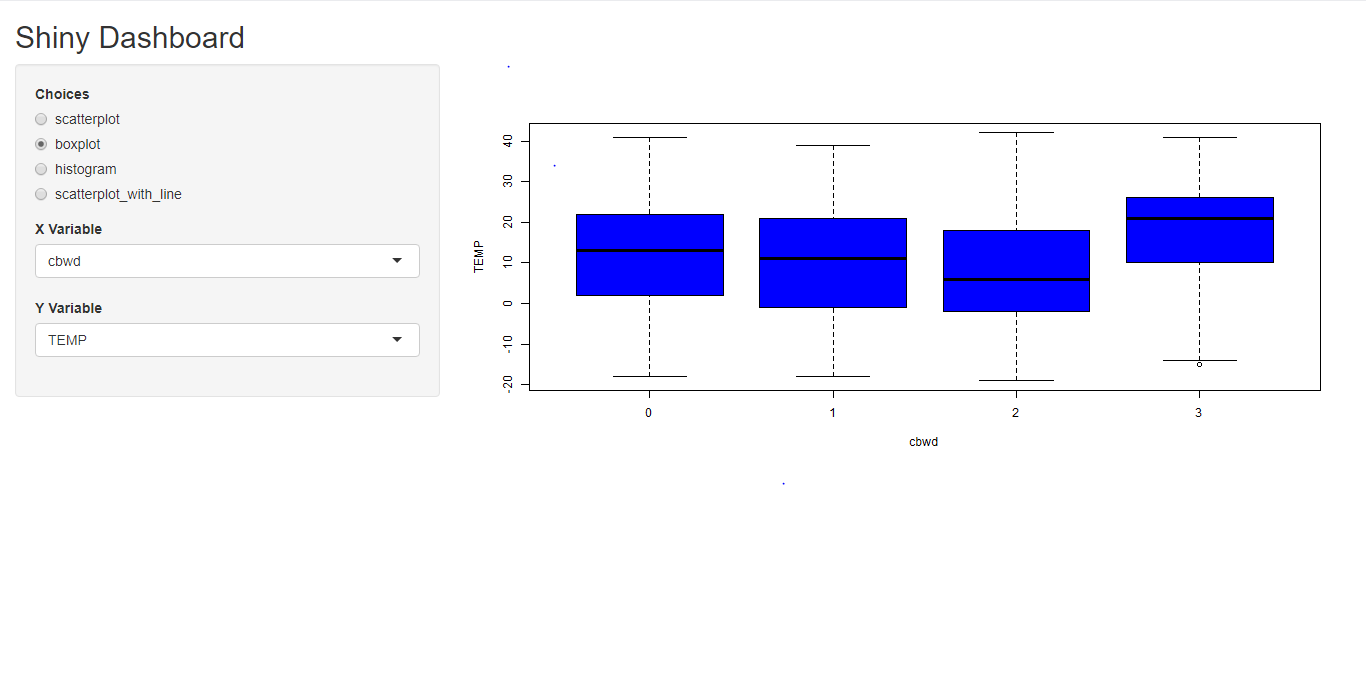
**2:**

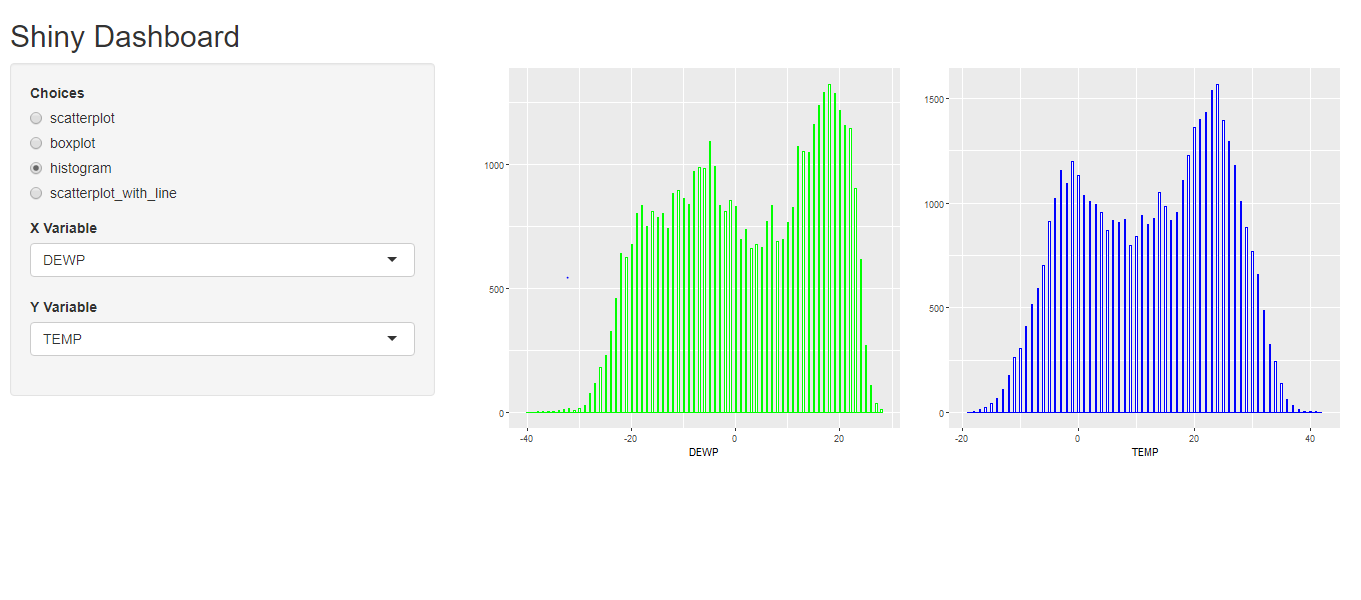
A shiny app has been developed to demonstrate various plots of our data in a dashboard using the shiny package in R. The application is structured with 2 parts – ui and server. Then, need to call the shiny app passing ui and server components to it.

Ui part contains one title panel containing the name of the app, one sidebarLayout with a sidebarPanel inside it containing various elements like dropdown, radio buttons etc.. and one mainPanel for plotting outputs. Here, the user could select any combination of x and y variables from the dropdown and would be able to select the type of plot (Scatterplot, Histogram, Boxplot) they want to see in the dashboard

The server part contains the backend logic like taking the input from the ui and plotting with the data from that input. Here, the choice of plot is taken from ui and the Temperature related data is plotted for that selected plot using normal plot functions like plot, hist(), boxplot() with renderPlot().

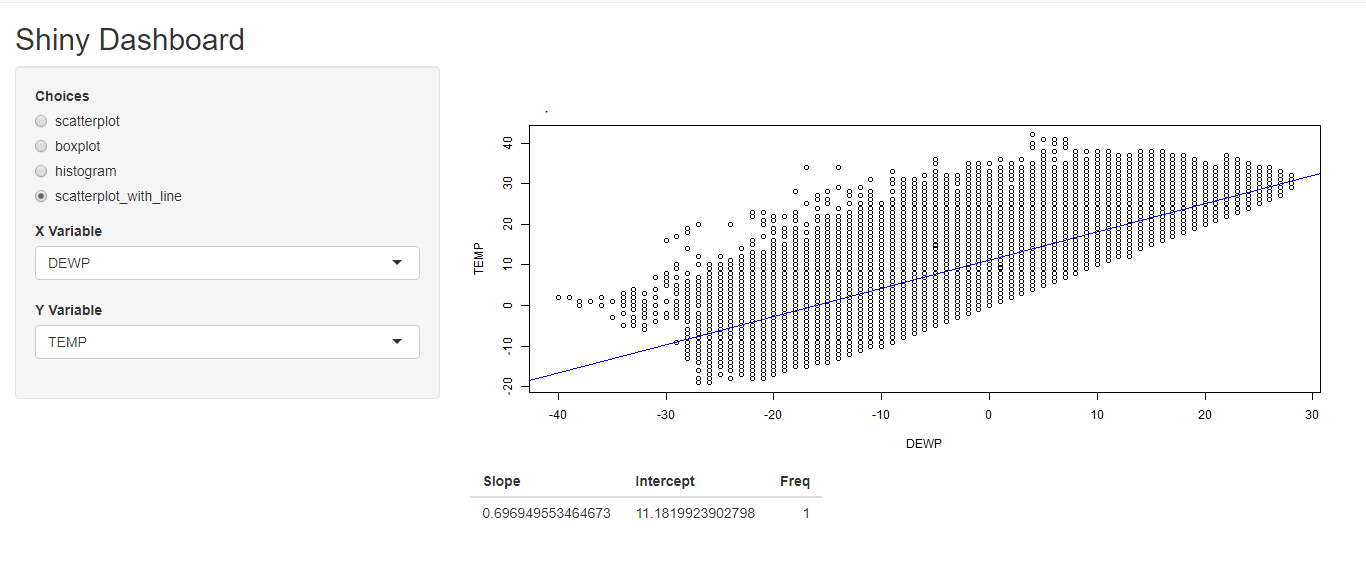






**3:**

Additionally, the ability to fit a linear regression model (the fitted line) to the scatterplot generated in the second part is provided as an option in the radio button list inside the mainPanel. A table with slope and intercept for the fitted line is also included below this scatterplot with fitted line using renderTable() function.

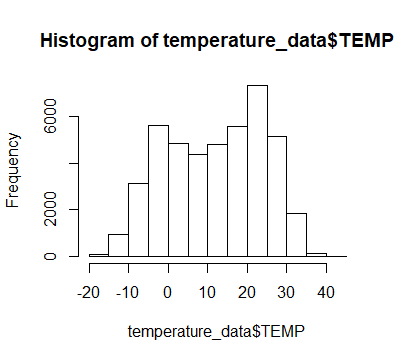


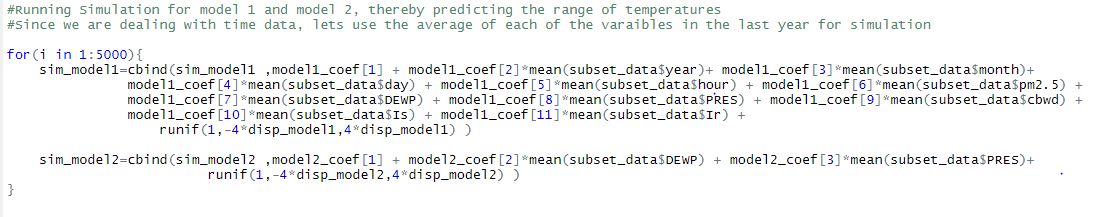
**4:**

We are going to predict the temperature for the subsequent years using at least two different models used in the part 1. The predictions are done with Monte Carlo simulations.

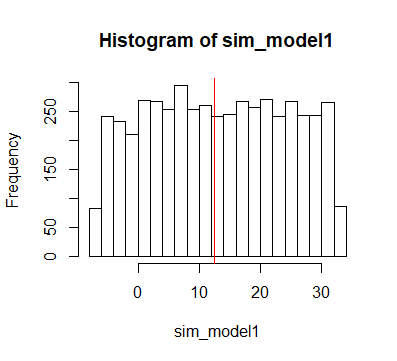
As we had time data (year, month, day, hour) in our dataset, we subset the data frame which contains only the last year’s data and use that while simulating.

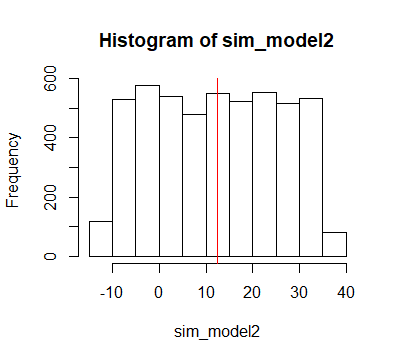
Model 1 (all the independent variables except the No (number) column and Model 2 (only DEWP and PRES as x variables) has been taken into account. Running two simulations using the above two models for predicting the subsequent temperatures. Have used uniform distribution for displacement as it the actual temperature distribution looks approximately uniform.

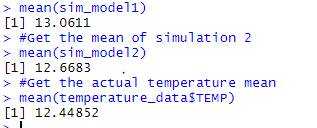




Now, plotted the histograms of both the simulated predictions and checked which simulation’s mean is closer to the actual temperature mean thereby inferring which model would be better.





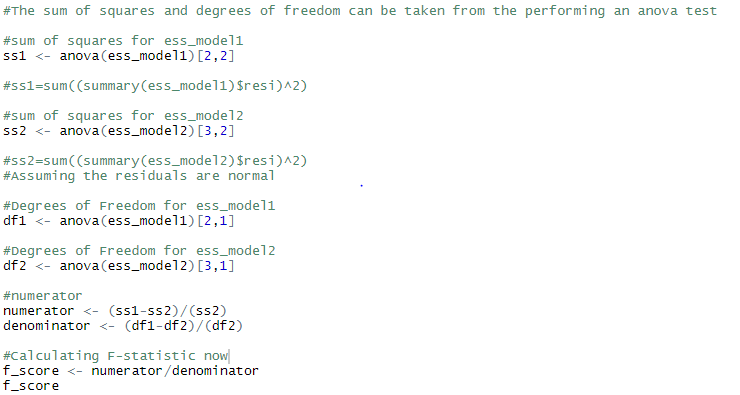


Based on the above **histogram distribution comparison and mean comparison** of the simulation of the two models, it could be seen that **mean of the second model** (**simpler model**) is **close** to the **actual temperature mean** and the **distribution** of the **second model** **closely resembles** the **actual temperature distribution**. So, **concluding that model 2 (the simpler model) performs the best.**

**Exercise 5:**

Now, we are going to construct two linear models one for Temperature TEMP as y variable and Pressure as x-variable (Simple linear model) and the other for Temperature TEMP as y and Pressure and Iws (Cumulative Wind speed) as x variables (multiple linear model). We were then asked to generate a distribution of ess statistic (f-score) by running a Monte Carlo simulation. The ess statistic can be computed using the residual sum of squares of both the models. The steps are as follows:

* Find the ess-statistic using the below formula which involves both the models. This would be used at last while checking where it would be falling in the generated ess statistic distribution. The residual sum of squares and the degrees of freedom could be easily found from running the anova test of each model.

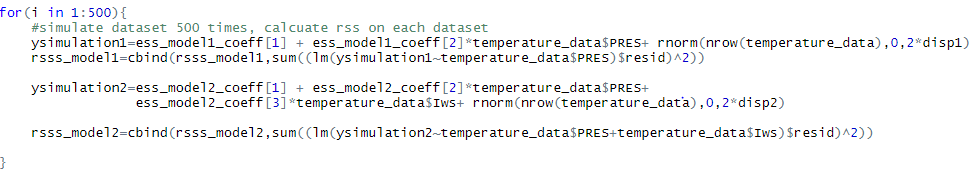




* Null Hypothesis: Simpler model is the better model

Alternate Hypothesis: Simpler model is not the better model (i.e,) the complicated model is the better model.

* Calculate the f-score with the given models using the formula below.
* Run a simulation for loop for any number of iterations (here 500 has been taken)
* For every iteration, simulate the first model’s prediction using its regression equation and compute the rss (Residual sum of squares) for that model with a certain displacement (here, rnorm distribution has been taken) and in a similar way, simulate the second model’s prediction using its regression equation and compute the rss for that model. The residual sum of squares values and degrees of freedom values could be easily taken from running an anova (Analysis of Variance) test for the respective models.
* The degrees of freedom would remain the same across iterations as the degrees of freedom is nothing but the total number of rows in the dataset (we have 43824 rows) minus the number of parameters in the model. Here the degrees of freedom for ess\_model 1 is 43822 whereas the degrees of freedom for ess\_model 2 is 43821.



* Now, Compute the f-score or ess statistic using the below formula:

**F statistic - To check which model is better** using the below formula:

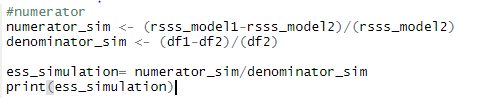
**F = ((SS1-SS2) / (SS2)) / ((DF1 - DF2) / (DF2))**

where SS1 = sum of squares of model1

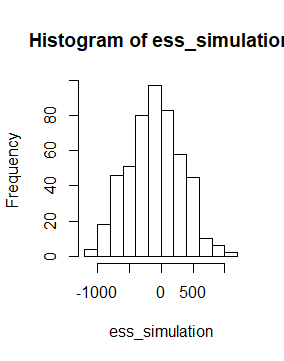
**SS2** = sum of squares of model2

**DF1** = Degrees of Freedom for model1 (**degree of freedom = number of observations - no of parameters involved in the model building**)

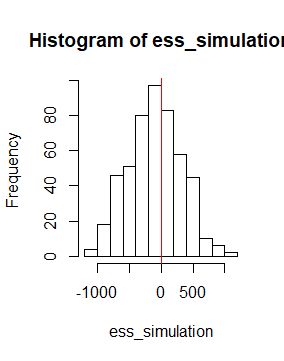
**DF2** = Degrees of Freedom for model2



* Plot a histogram for ess\_statistic vector which contains the ess statistic for 500 iterations. Look at the distribution.



* Cut the plot using the abline of ess statistic found before the simulation.



* If the line falls within 95%, we fail to reject the null hypothesis, so, the simpler model is better, otherwise complicated model is better.
* For us, we got the simpler model as the better model as the abline of main ess statistic in the above histogram fell within 95%.

**Conclusion:**

Thus, five tasks have been carried out with the first one we found the model 1 appropriate as it has the higher multiple R-squared value but model 2 was simple, though it had a little less multiple R-squared value. In the 2nd and 3rd, the shiny app was built with plots like scatterplots, histograms, boxplots and scatterplot with fitter linear regression line based on the user selection. In 4th task, we simulated the temperature predictions for the subsequent years using two different models with Monte Carlo simulations and found model 1 performed the best. In the final question, a distribution of ess test statistic had been generated with the two given models using Monte Carlo simulations and plotted and found the simpler model was the better fit for the data.