DS Assignment
EDA in Indian Weather Data

Shreya Shettigar FMDS2122007
Anjali Jain FMDS2122004
Ajay Daniel Martin FMDS2122001

Details of the Data Set

- The IMD has six Regional Meteorological Centres, each under a Deputy Director General. These are located in **Chennai, Guwahati, Kolkata, Mumbai, Nagpur and New Delhi**. There is also a Meteorological Centre" in each state capital.
- Our Dataset Is a Collection of Different Data that are collected at weather stations on the ground that is Distributed in the Opensource for Education Purposes
- Our Dataset is form the Site Kaggle for with the name **Indian Weather and Astronomy Data which contains** Astronomical Data,Forecast data and Locations.
- Of the Above mentioned Datasets we chose Forecast.csv which contains Weather information for all the major cities in all of the 29 states of India. The file contains ~24000 rows and 34 columns.

- Our Basis of the Columns Selections is by the Fact that The weather of an area is due to four factors. They are **heat energy, air pressure, winds, and moisture.** Changes in these factors determine the kind of weather an area will have.
- The Weather units are saved both in International and American units Standards which will be later cleaned as per our requirements
- As for the Accuracy of the Data Collected and its time we have both the IST and The Time Epoch to indicate the Exact Point of time

Data sets Columns

Time_epoch	Unix time is a system for describing a point in time
Time	The point of time as measured
Temp	the degree or intensity of heat present at the time
is _day	To show Day(1) and Night (0)
Condition	The Weather Condition as measured
wind	The Strength of the Wind
Wind_degree	The numerical measure that measure the direction of the wind
Wind_dir	The Direction the wind originated from
Pressure	The Atmospheric pressure is an indicator of weather is high or low
Precip	precipitation is any product of the condensation of atmospheric water vapor that falls under gravitational pull from clouds
Humidity	The amount of water in air
cloud	The mass of water or ice substance in the atmosphere

Data sets Columns

Feelslike	a measurement of how hot or cold it really feels like outside.
Windchill	a measure of the rate of heat loss from skin that is exposed to the air
Heatindex	Indicates what the temperature feels like to the human body when relative humidity is combined with the air temperature.
Dewpoint	the temperature the air needs to be cooled to (at constant pressure) in order to achieve a relative humidity (RH) of 100%
Will_it_rain	Indicates if there is a chance ofrain for the coverage area during a 24 hour time period
Chance_of_rain	an expression of how likely it is to see rain over the coverage area during a 24 hour time period
Will_it_snow	Indicates if there is a chance of snow for the coverage area during a 24 hour time period
Chance_of_snow	an expression of how likely it is to see snow over the coverage area during a 24 hour time period
Vis	the clearness of the atmosphere and the maximum range at which objects and lights can be clearly sighted.
Gust	a sudden increase in wind speed above the average wind speed.
State	Name of the State
city	Name of the city from the State

Dependent and independents Variable of the dataset

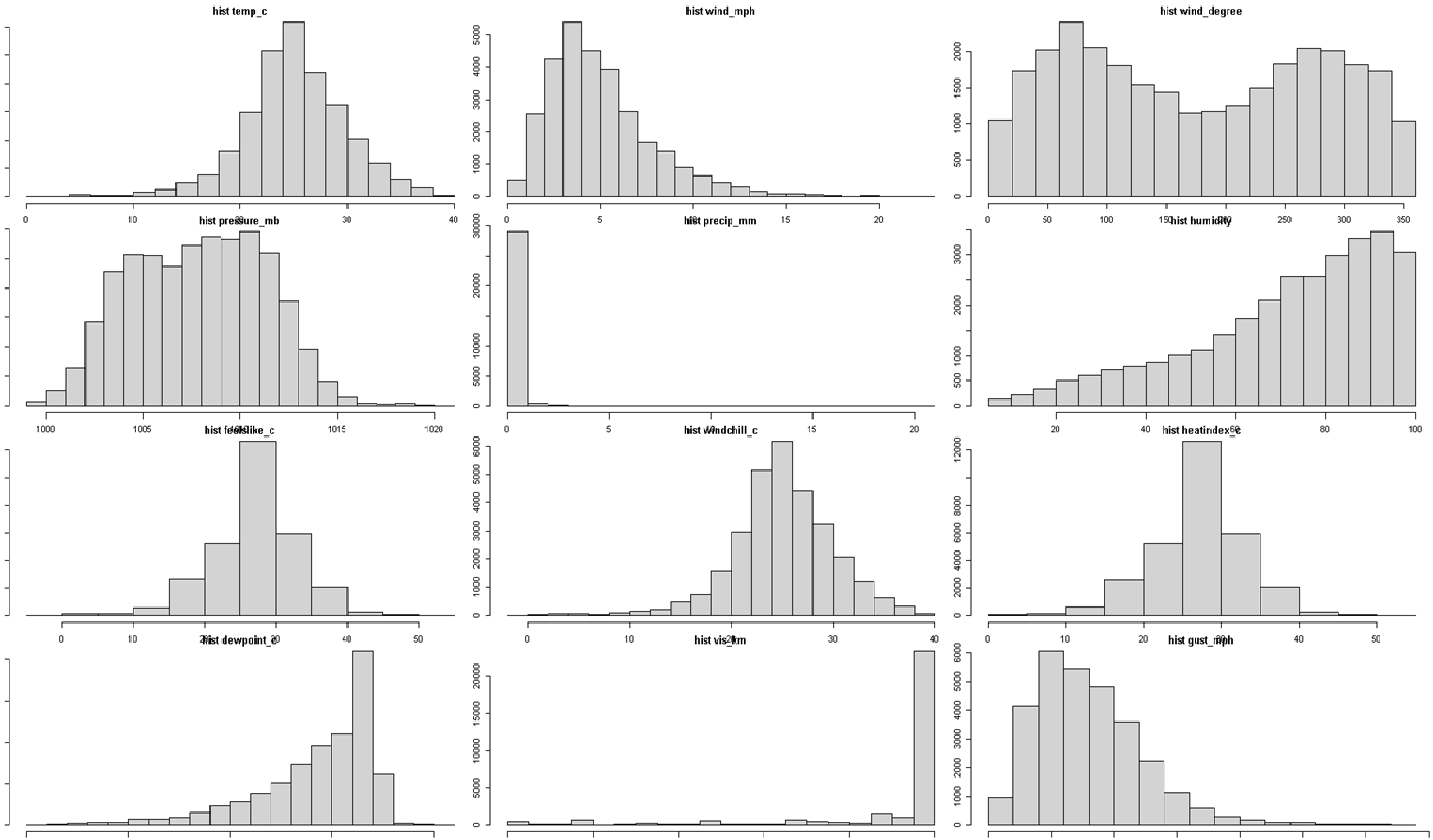
Now For Climate and its reading most of the variables are dependent on each other the relation Ship between the variable will be explored in the later part of the ppt

The Independent Variables in our dataset are **time,time_epoch, Is_day, Cloud, wind_dir, wind_degree, gust, feels_like, vis,** & **State**

The Dependent variables are **temp, condition, wind, pressure, precip, humidity, cloud, windchill, heatindex, dewpoint, will_it_rain, will_it_snow, Chance_of_rain & chance_of_snow**

Importing and Cleaning of the dataset

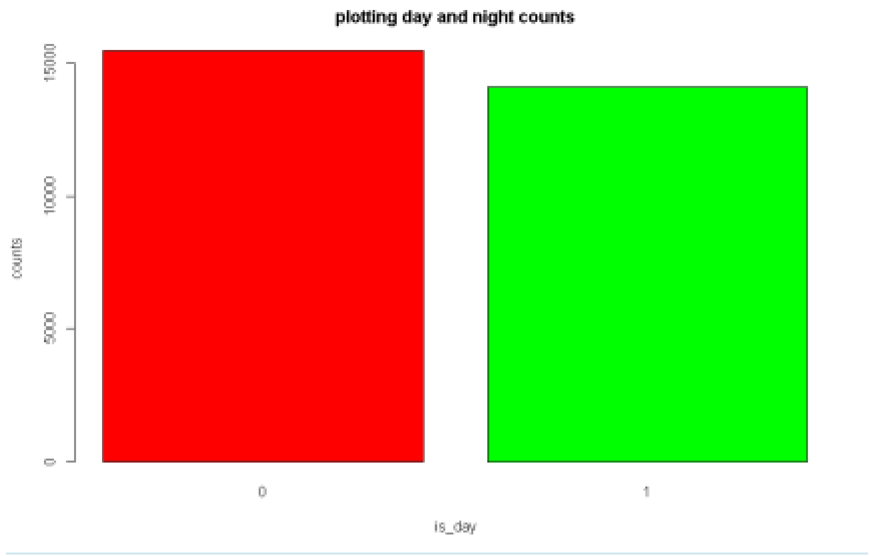
- forecast=read.csv("FinalForecastData.csv") Reading the Data
- #removing redundancy
- forecast=forecast[-c(5,9,13,15,19,21,23,25,26,28,27,29,31,33)] Removing the Us Standard units columns
- View(forecast)
- names(forecast)=make.names(names(forecast)) Validating names of the columns
- names(forecast)
- nrow(forecast) Number of Rows in the Matrix
- ncol(forecast) Number of Columns in the Matrix
- str(forecast)
- par(mfrow=c(4,3),mar=c(1,1,1,1)) visualising cols which are continuous
- continuous_distribution=forecast[c(4,7,8,10,11,12,14,15,16,17,18,19)] Continuous Distribution of required Column
- x=names(continuous_distribution)
- j=0
- for(i in continuous_distribution){
-   j = j + 1
-   hist(i,xlab=x[j],main=paste("hist",x[j]))
- }


hist temp_c
hist wind_mph
hist wind_degree
hist pressure_mb
hist precip_mm
hist humidity
hist feelslike_c
hist windchill_c
hist heatindex_c
hist dewpoint_c
hist vis_km
hist gust_mph

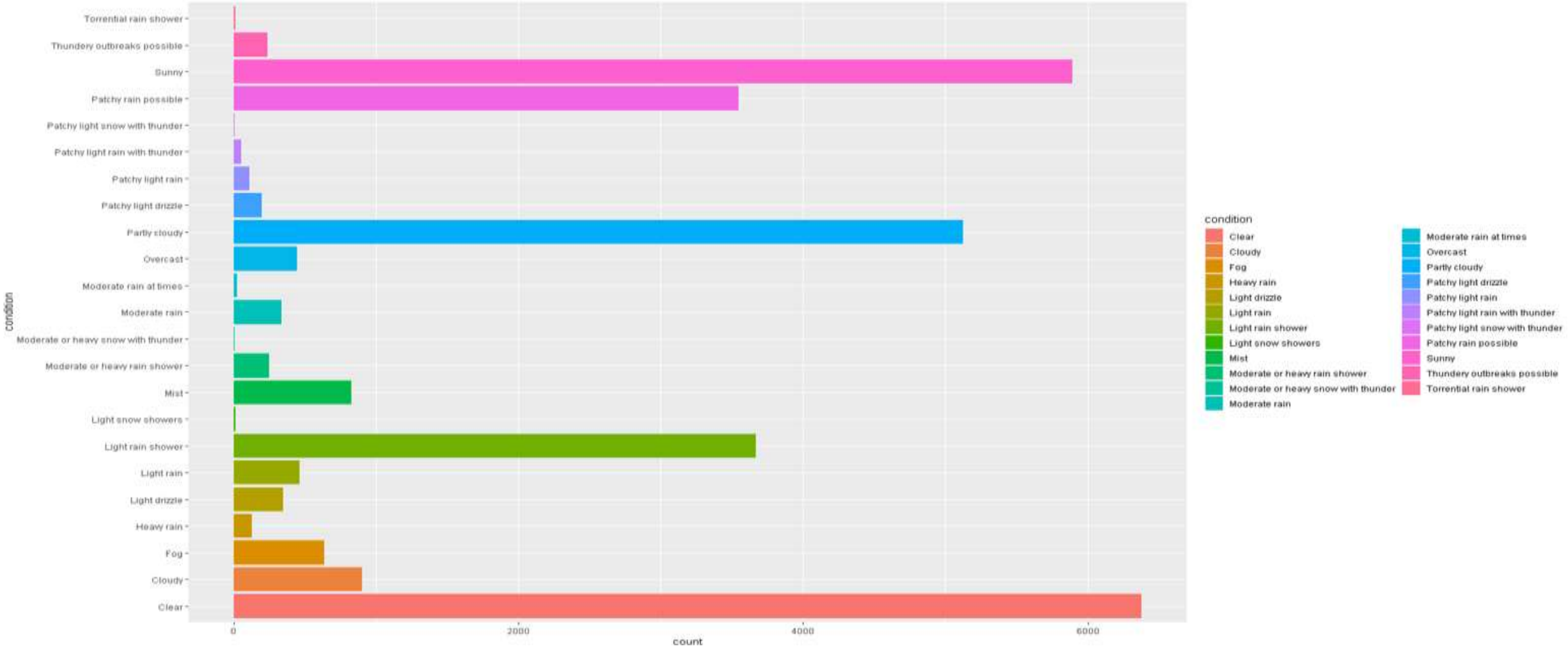
We did preliminary tests to confirm our assumptions regarding the data from visualizating the columns individually

- library(ggplot2)
  library(viridisLite)
  library(reshape2)
  #visualizing categorical variables
- barplot(table(forecast$is_day),xlab="is_day",ylab="counts",main="plotting day and night counts",col=c("red","green"))

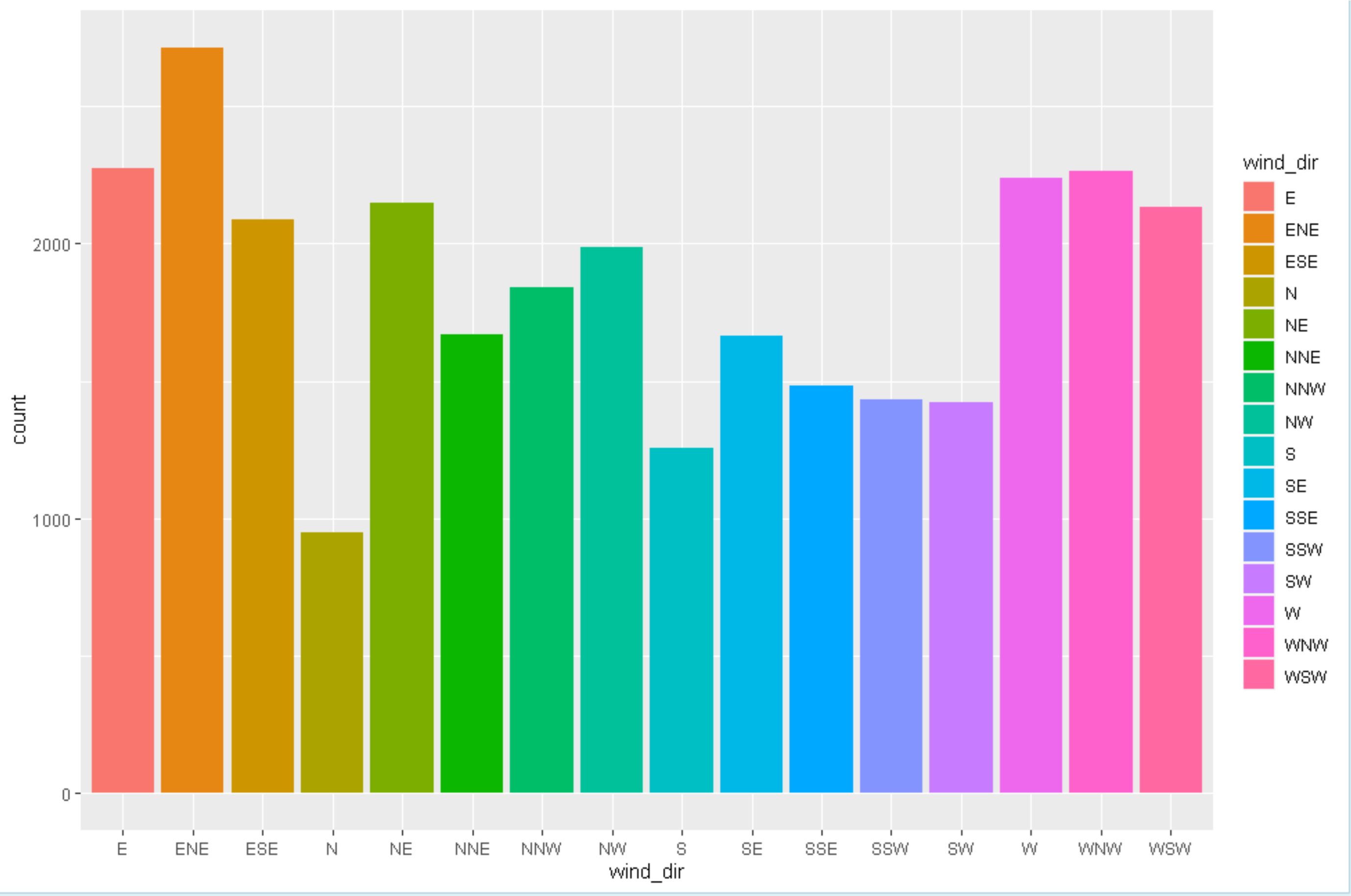
 Libraries used for Viz



#visualising weather conditions
ggplot(forecast,aes(y=condition,fill=condition,main="weather_conditions"))+geom_bar()

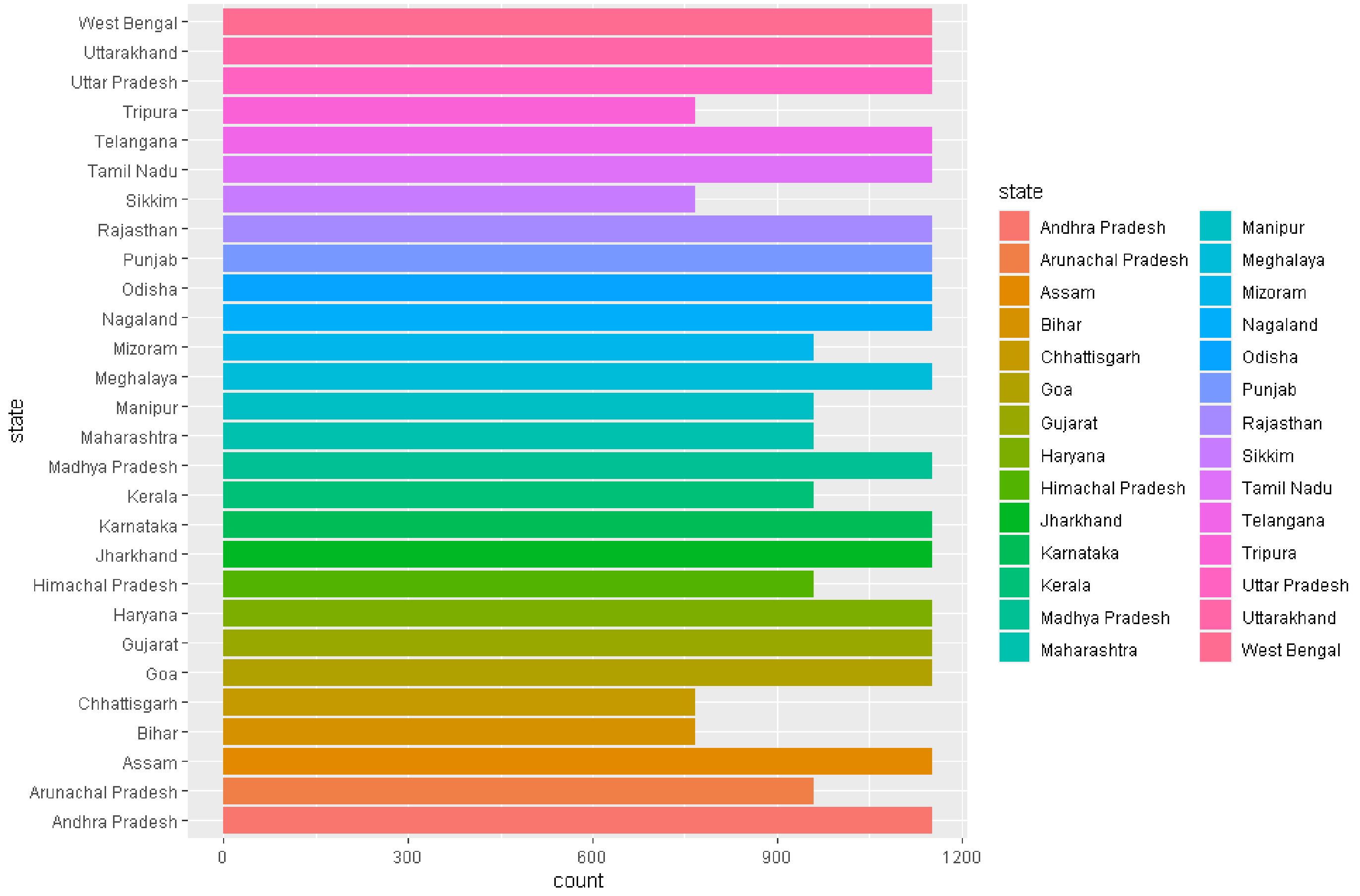


#visualising wind direction
ggplot(forecast,aes(x=wind_dir,fill=wind_dir,main="wind_Directions"))+geom_bar(
)



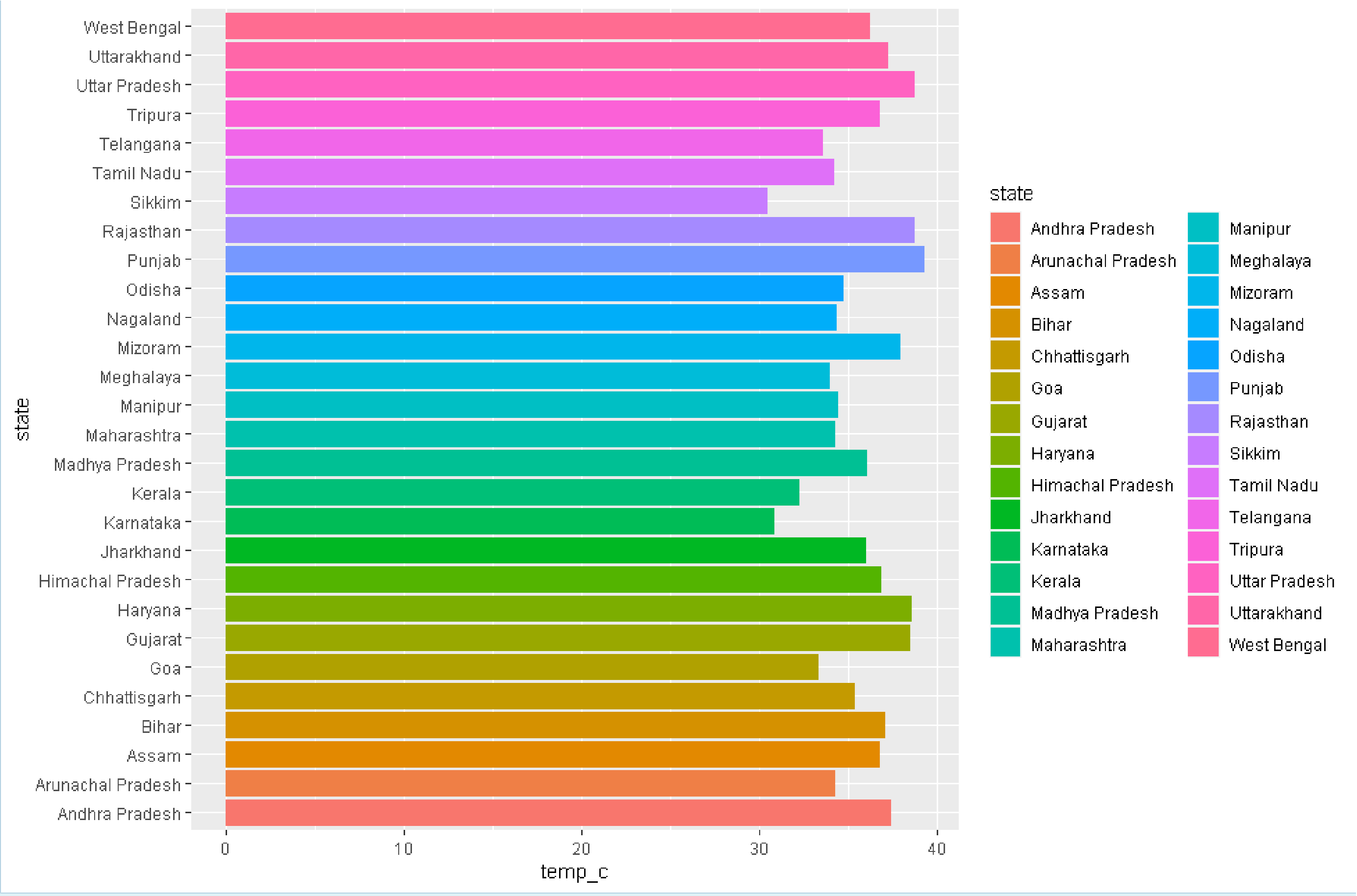
#visualising state distribution

```
ggplot(forecast,aes(y=state,fill=state,main="states"))+geom_bar()
```



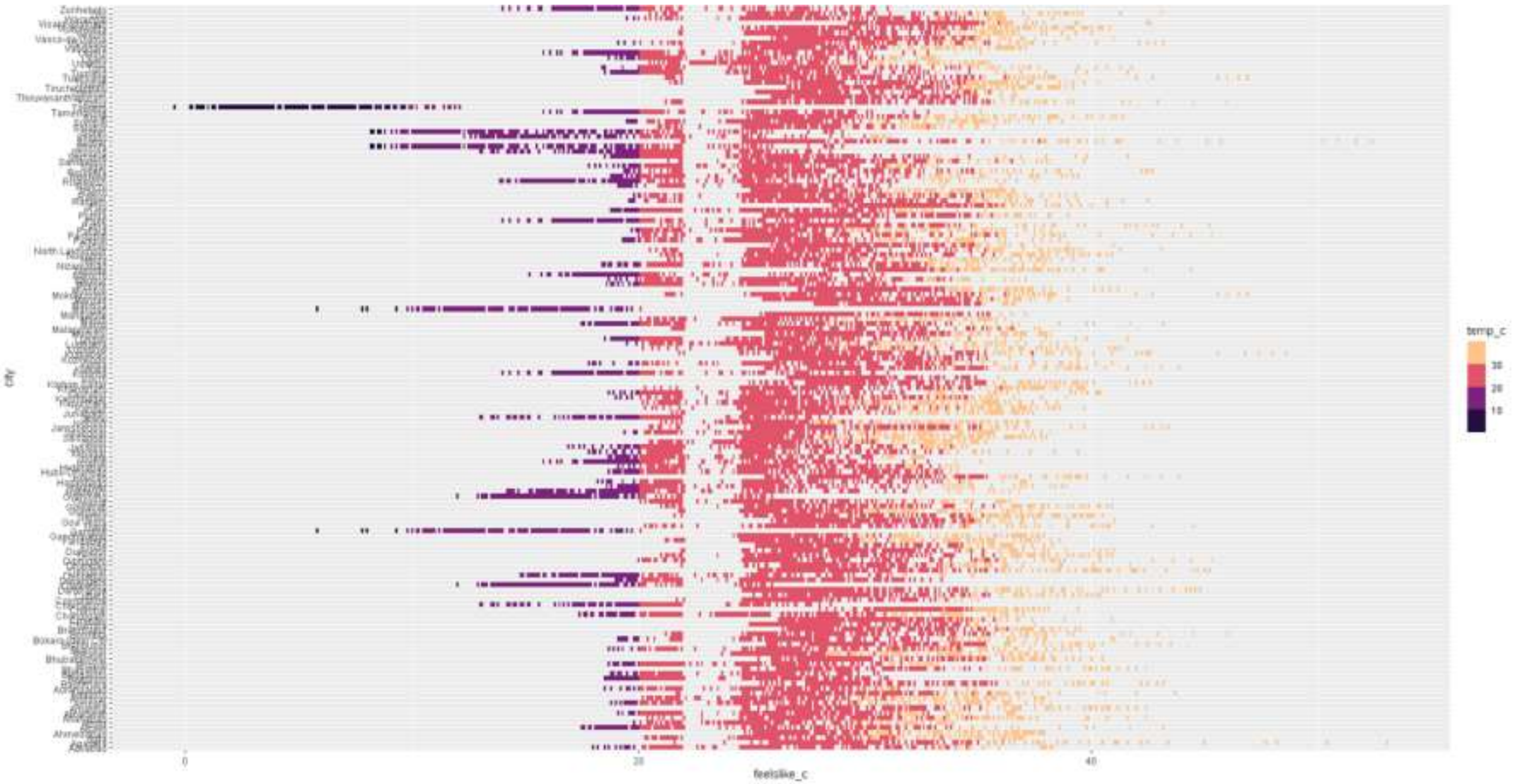
#state wise temperature

```
ggplot(forecast,aes(x=temp_c,y=state,fill=state))+geom_col(position=position_dodge())
```

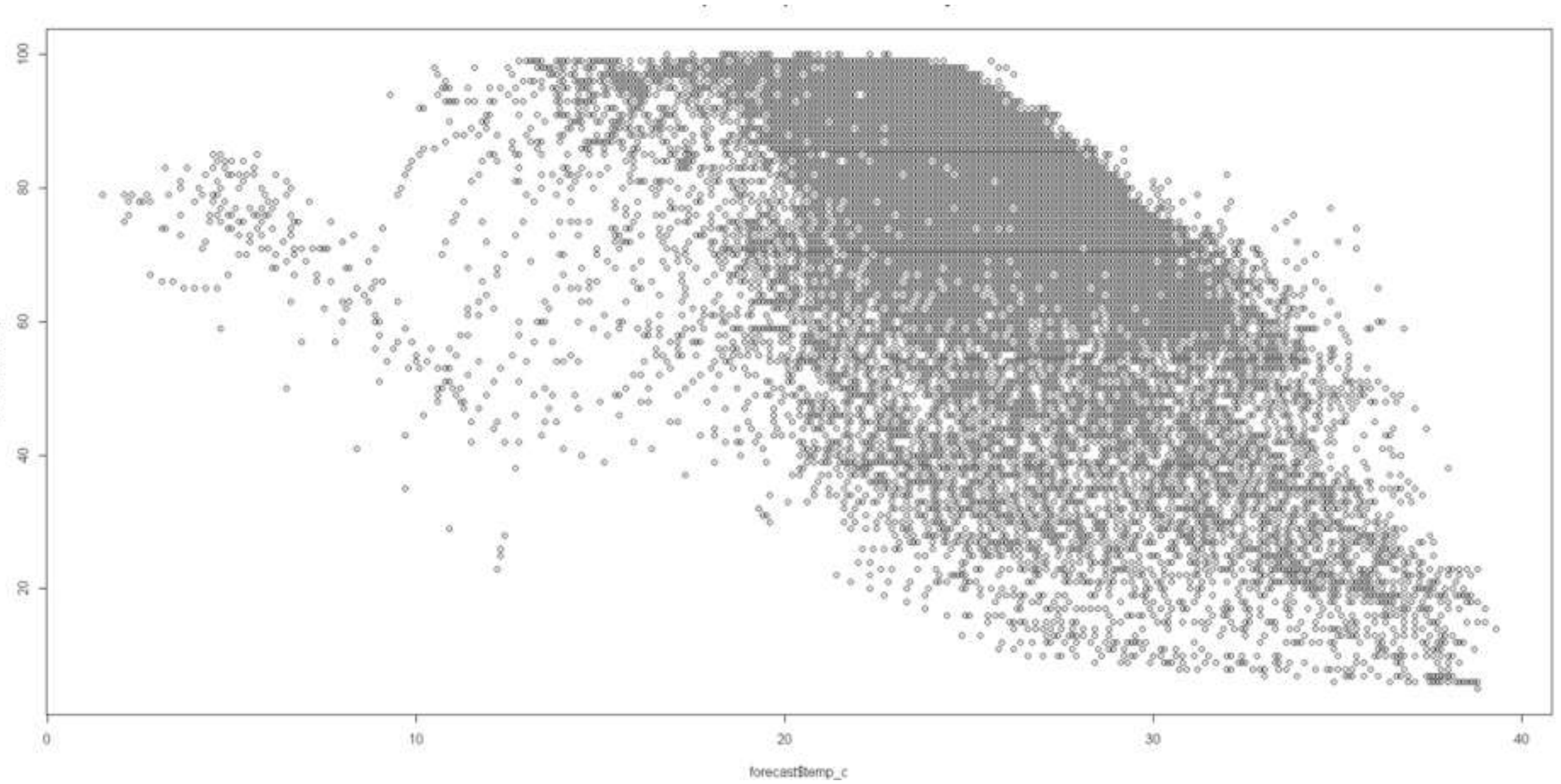


#Heatmap statewise using Temp and Feels Like
ggplot(data = forecast, aes(x=feelslike_c,y=city)) + geom_tile(aes(fill=temp_c)) + scale_fill_viridis_b(option ="magma")

This shows city wise what it feels like and what the actual Temperature is



```
#scatterplot
plot(forecast$temp_c,forecast$humidity,main = "scatter plot of temperature and humidity")
```



Correlation between the columns in the dataset

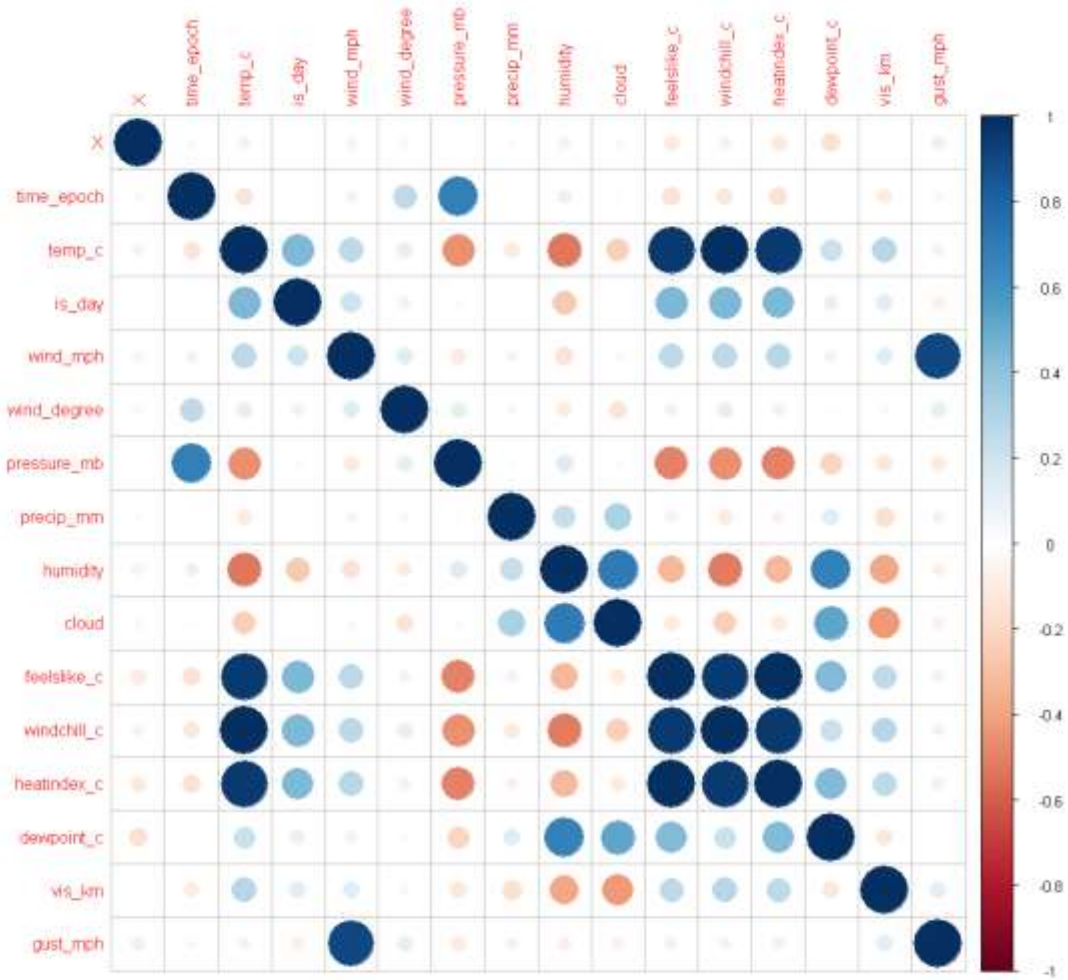
Now in weather except for time and day every other variable is dependent on the other variable we can predict temp using other variables

each and every other variable in this dataset is corelated to the other

```
#corellation matrixxx

corr_mat <- round(cor(forecast_new),2)

melted_corr_mat <- melt(corr_mat)

ggplot(data = melted_corr_mat, aes(x=Var1, y=Var2,fill=value)) + geom_tile()
```

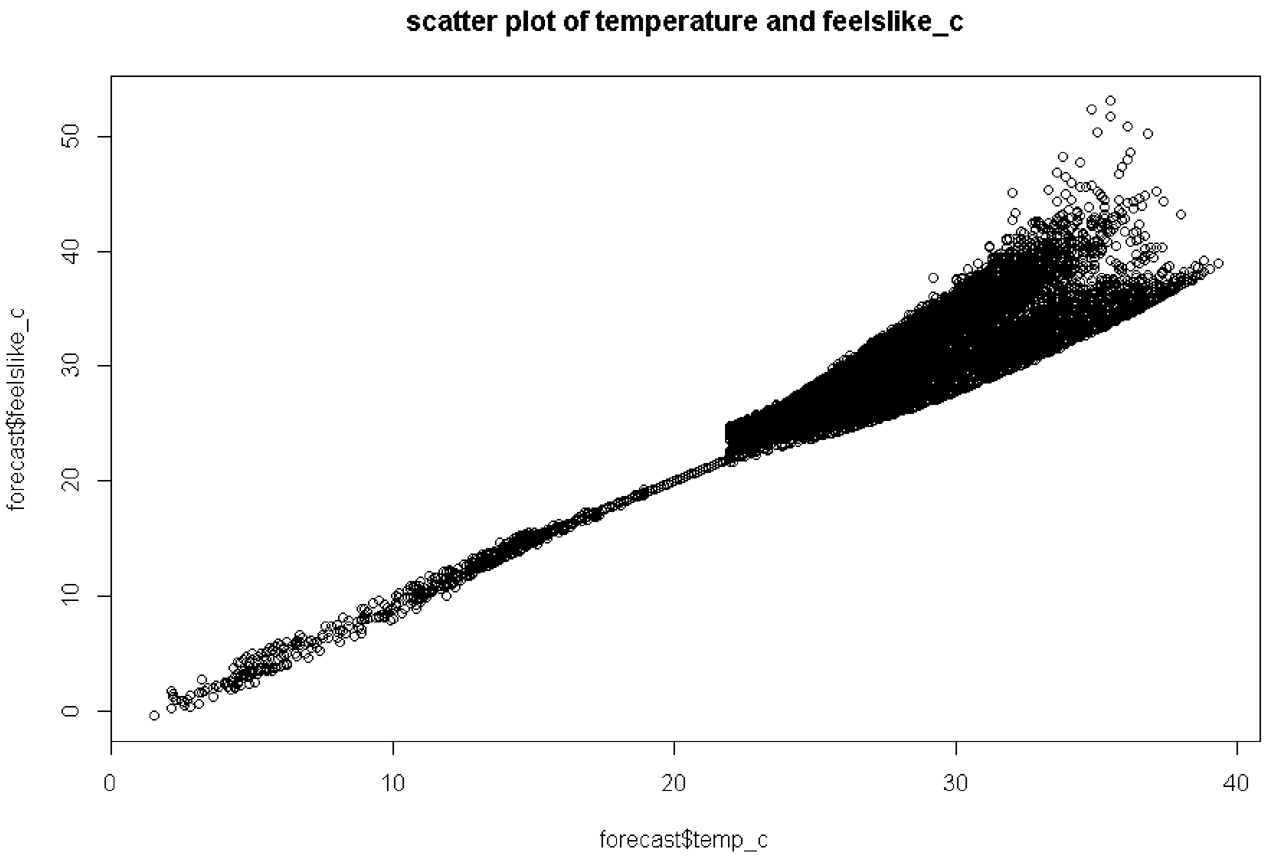
Selecting variables from the correlation

- From the correlation plot we can see that the relation between temperature is strong with Feels like, Wind chill and Heat index
- So we decided to choose Temp as the dependent variable and the remain three variable as independent
- And to confirm their relation we used scatter plots for each independent variable with the dependent variable(Temp_c)
- And the Scatterplot showed the Relation between Temp and its independent variable as Positive



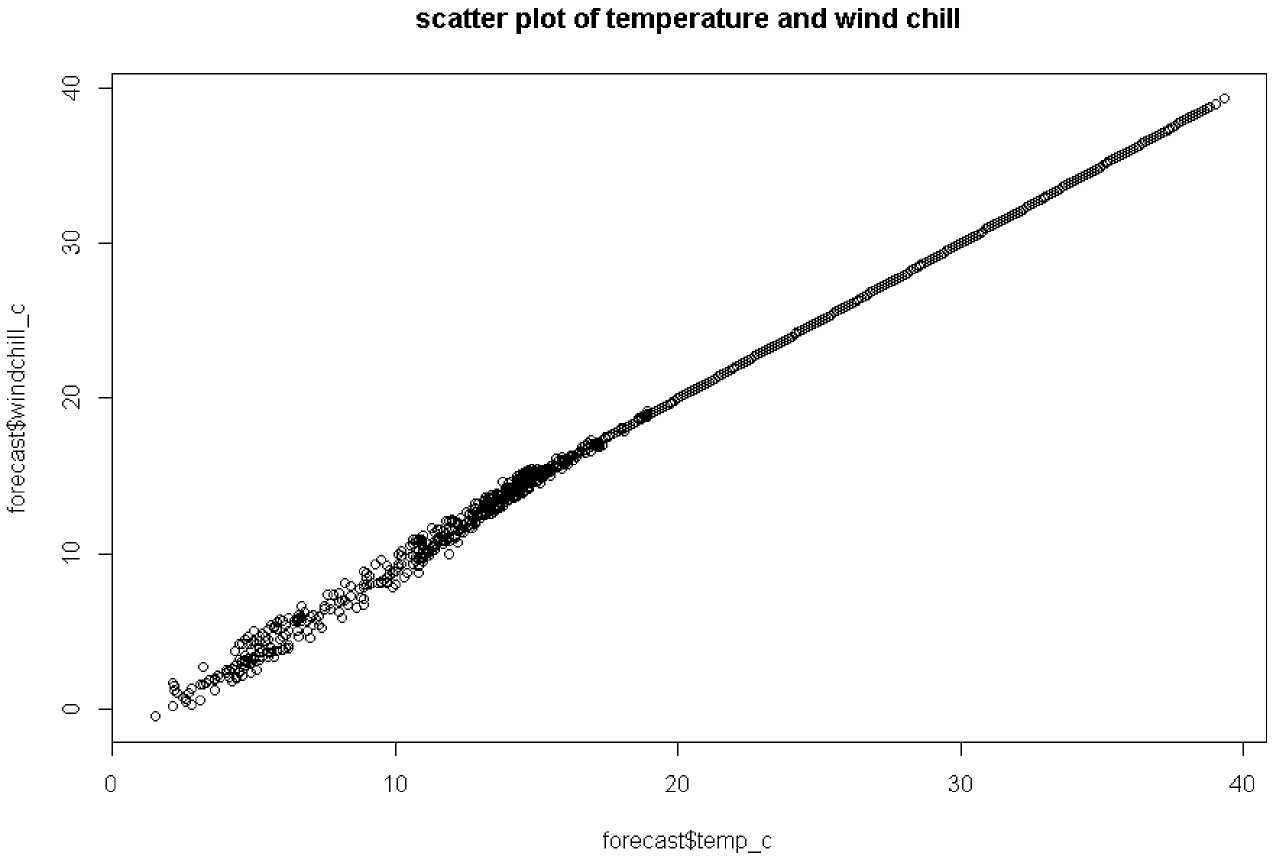
Relation between Temperature and Feels Like

plot(forecast$temp_c,forecast$feelslike_c,main = "scatter plot of temperature and feelslike_c")



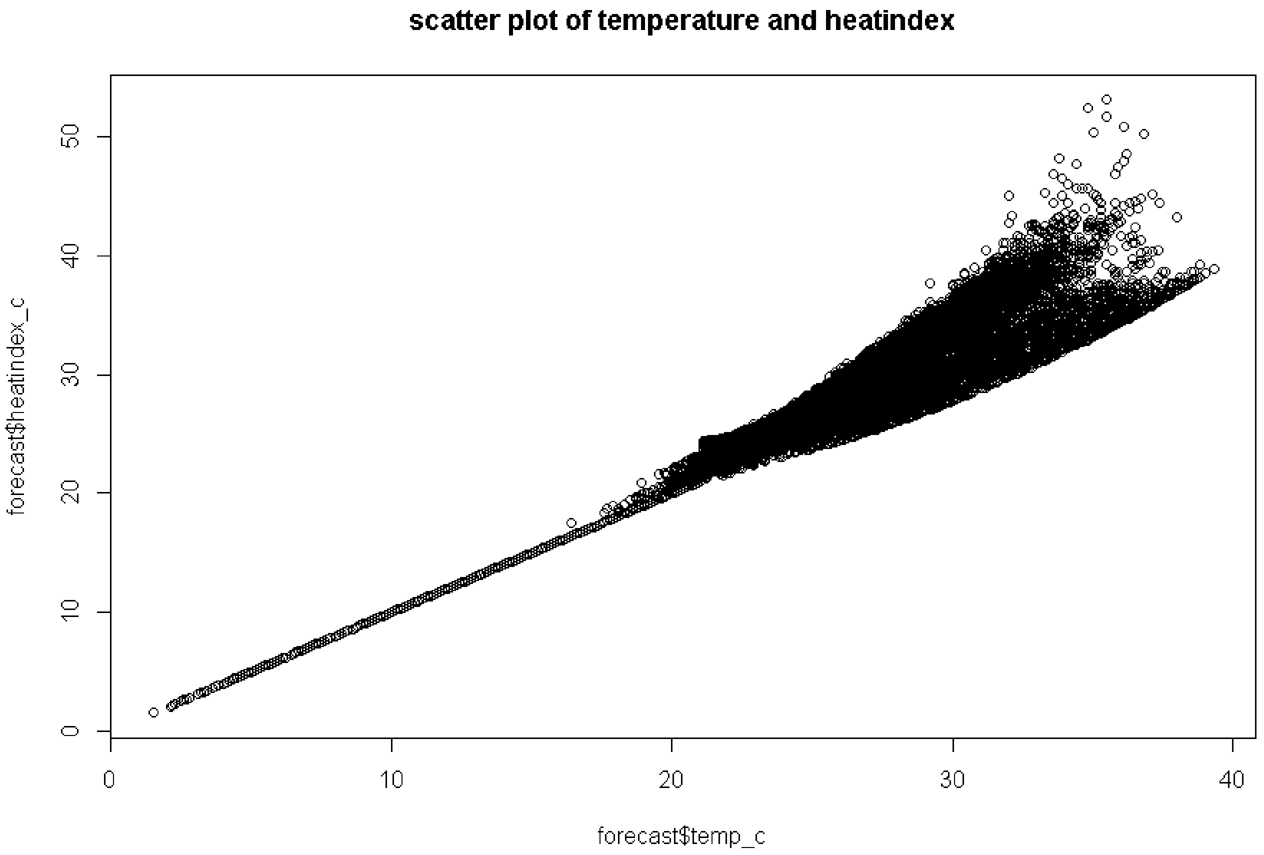
Relation between Temperature and wind chill

```
plot(forecast$temp_c,forecast$windchill_c,main = "scatter plot of temperature and wind chill")
```



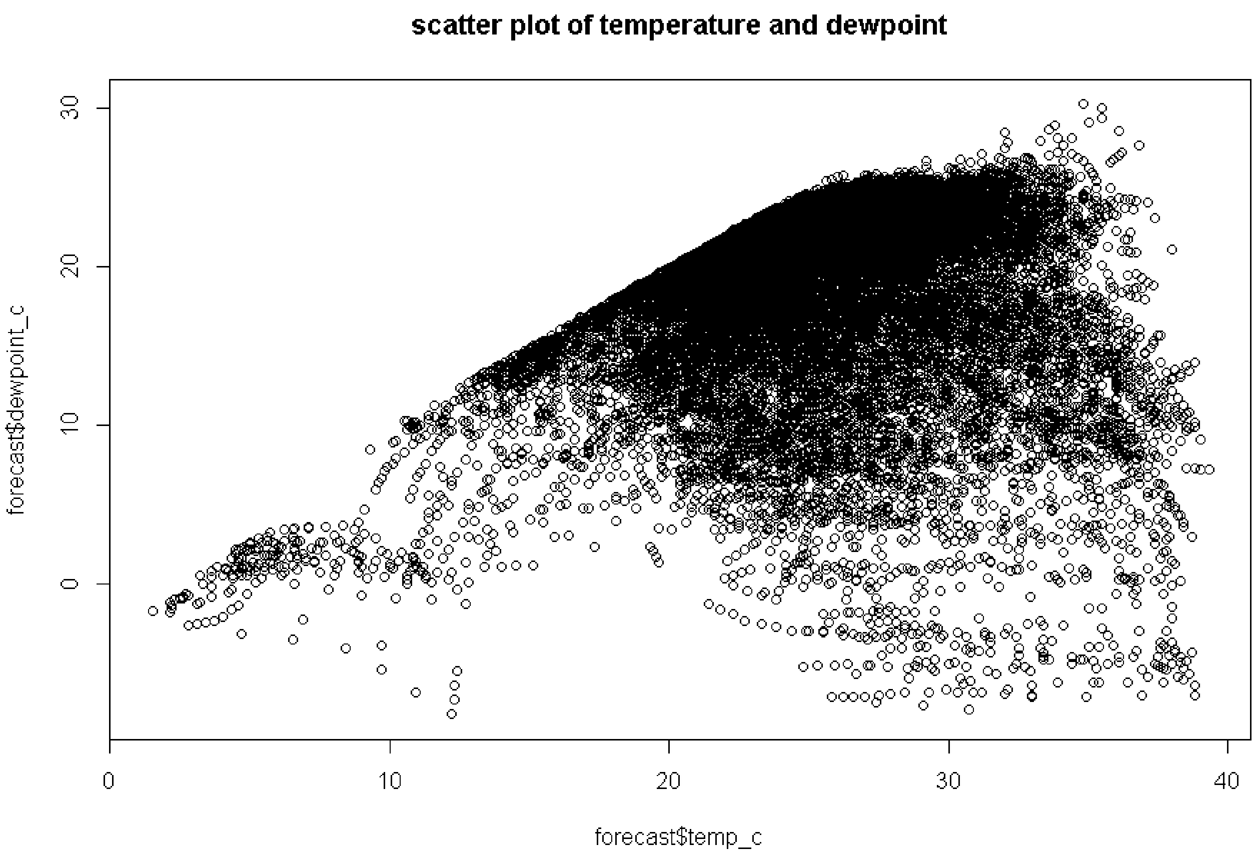
Relation between Temperature and Heatindex

```
plot(forecast$temp_c,forecast$heatindex_c,main = "scatter plot of temperature and heatindex")
```



Relation between Temperature and Dewpoint

```
plot(forecast$temp_c,forecast$dewpoint_c,main = "scatter plot of temperature and dewpoint")
```



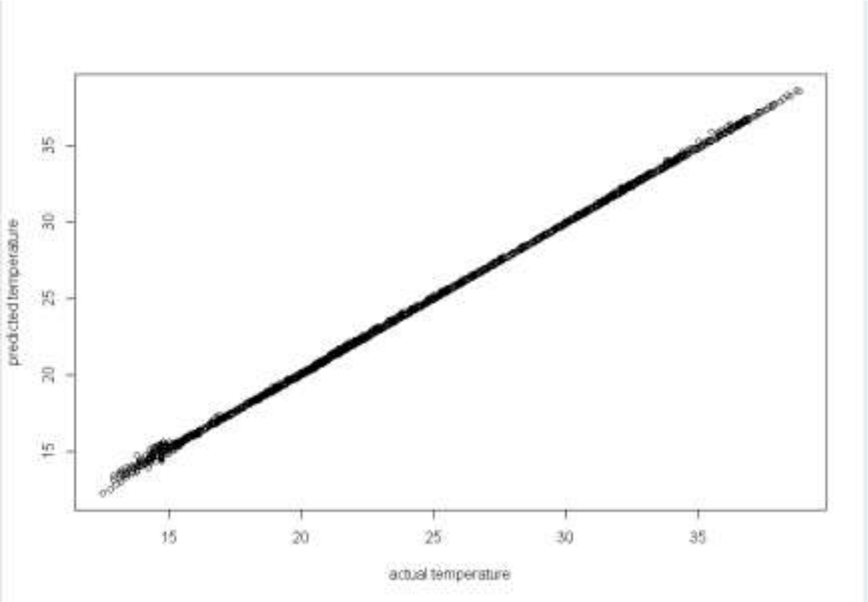
Regression model

```
forecast=read.csv("FinalForecastData.csv")
forecast=forecast[-c(5,9,13,15,19,21,23,25,26,28,27,29,31,33)]
View(forecast)
names(forecast)=make.names(names(forecast))
forecast=na.omit(forecast)
dim(forecast)
df=forecast[,c(4,14,15,16,17)]
View(df)
str(df)
split=sample(0.8*nrow(df))
training=df[split,]
head(training)
testing=df[-split,]
head(testing)
reg_model=lm(temp_c~.,data=training)
summary(reg_model)
prediction=predict(reg_model,testing)
data.frame(prediction,testing$temp_c)
```

```
> head(x)
      prediction testing.temp_c
23655   33.26279           33.3
23656   32.95878           33.0
23657   31.85431           31.9
23658   30.65797           30.7
23659   29.55350           29.6
23660   29.15587           29.2
>
```

- Reading the Dataset
- Removing Redundancy
- Structuring the new datasets Col names
- Removing the NA Values
- Filtering the unwanted cols for regression
- Compact display of the structure
- Splitting data sets into training and testing
- 80% training
- 20% Testing
- Applying Regression model with temp

```
plot(testing$temp_c,prediction,xlab="actual temperature",ylab="predicted temperature")
```



Now we can Predict the Temperature using its Independent Variables

```
x=data.frame(feelslike_c=24, windchill_c=25 ,heatindex_c=26 ,dewpoint_c=28)
predict(reg_model,x)
```

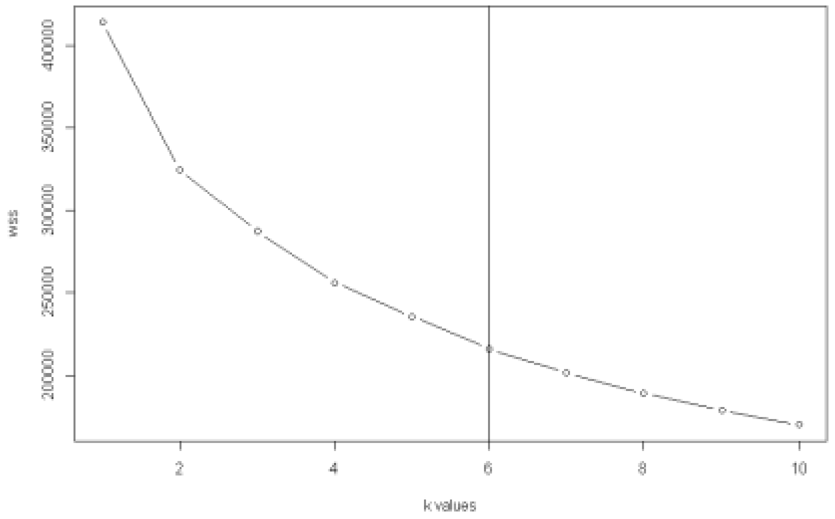
```
> predict(reg_model,x)
       1
24.81729
>
```

The Predicted Value of the temperature is 24.8 C

K – Means Clustering

using k means clustering we are teaching the machine to cluster the unlabled data into groups and further figure the useful data from the clusters

```
forecast_new=forecast[-c(1,2,3,6,9,20,21)]

View(forecast_new)

maximum=10

scal=scale(forecast_new)

wss=sapply(1:maximum,function(k){kmeans(scal,k,nstart=50,iter.max = 15)$tot.withinss})

plot(1:maximum,wss,type='b',xlab='k values')

abline(v=6)
```



km=kmeans(forecast_new,6,iter.max = 50)
km

```
> km
K-means clustering with 6 clusters of sizes 4072, 3807, 3728, 6809, 6160, 4992

Cluster means:
    temp_c    is_day wind_mph wind_degree pressure_mb   precip_mm humidity    cloud feelslike_c windchill_c
1 23.69804 0.4963163 4.395383   183.23232    1008.448 0.236502947 86.93296 74.05845    26.02257    23.69283
2 26.87040 0.5208826 5.305726   235.82637    1008.869 0.001113738 53.32046 10.33018    28.01763    26.86625
3 24.24254 0.4723712 5.275724   287.78702    1008.256 0.172988197 85.81599 73.98015    26.58887    24.24273
4 24.71964 0.4624761 5.066544   103.48759    1007.995 0.142979880 78.42782 58.24145    26.94990    24.69258
5 24.78481 0.4147727 4.576136    44.21981    1007.846 0.067112013 67.22175 31.16623    26.47498    24.75654
6 26.75136 0.5282452 5.945353   313.28365    1008.998 0.000713141 59.97977 12.17268    28.26707    26.75136
  heatindex_c dewpoint_c   vis_km gust_mph
1    26.15236   21.19990 8.439808 7.023649
2    28.15474   14.96194 9.971395 8.382217
3    26.73018   21.55231 8.569930 8.063063
4    27.10963   20.08368 8.943281 7.855647
5    26.61274   17.20766 9.607273 7.464578
6    28.41084   17.04399 9.925821 9.005629

Clustering vector:
   [1] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4 5 5 5 5 5 3 3 3 3 3 3 3 3 3 3 3 1 4 4 4 4 4 5 5 5 5 5 4 2 6 6 6 6 6 6 6
  [56] 6 6 6 6 6 6 6 6 6 2 4 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 2 4 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 2 1 4 4 4 4 4 4
 [111] 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 2 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4 2 6 2 4
 [166] 5 4 3 6 2 4 5 4 2 6 2 4 5 5 5 5 5 4 4 4 4 4 4 5 5 4 1 5 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 4 5 5 5 5 4 4 5 4 3 3
 [221] 3 3 3 3 3 3 3 3 3 3 3 3 1 4 5 5 5 5 4 4 1 1 2 3 1 4 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 1 1 3 3 3 3 3 3 3 2 1 4 4
 [276] 4 4 4 4 4 4 4 4 4 1 1 1 1 3 3 4 5 5 5 5 4 4 4 4 4 4 4 4 4 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2
 [331] 2 1 1 1 1 2 1 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [386] 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 1 1 1 1 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 6 6 6 6 6 2 2 2 2 3 3 3 3 3 3 3 3
 [441] 3 3 3 3 3 3 3 3 3 6 6 6 6 6 6 6 3 2 2 2 3 3 3 3 3 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 3 4 4 4 5 4 1 6 6 6 6 6 6 6 6
 [496] 6 3 3 3 3 3 3 3 3 1 1 1 1 1 2 3 6 6 6 6 6 6 6 3 3 3 6 6 3 3 3 3 3 2 3 3 3 3 3 3 3 6 6 6 6 6 6 6 6 2 4 5 4 4 1
 [551] 1 2 3 3 3 3 6 6 6 6 6 6 6 3 3 3 6 6 6 6 2 2 2 2 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3
 [606] 3 6 6 2 2 2 2 2 2 2 1 1 1 1 2 2 2 6 6 3 3 6 6 6 6 6 6 6 6 6 6 6 6 3 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3
 [661] 3 3 3 6 6 6 6 6 6 6 6 6 2 2 2 2 6 6 6 6 6 6 6 6 3 2 4 5 4 4 4 1 1 1 1 1 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 [716] 6 6 6 6 6 6 6 6 6 6 6 6 6 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 1 5 4 4 4 4 4 5 4 4 4 1 1 1 1 3 6 6
 [771] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 1 1 1 2 2 2 1 4 4
 [826] 5 5 5 4 4 4 4 4 4 4 4 4 4 1 1 2 2 2 6 6 6 6 6 6 6 2 4 5 5 4 4 4 4 4 1 1 1 1 1 6 6 6 6 6 6 6 6 6 6 6 6 6 2 4 5
 [881] 4 2 6 6 6 6 2 4 3 3 3 3 3 3 3 3 1 4 4 4 4 4 4 4 4 1 1 1 2 2 2 3 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 [936] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 3 6 6 6 6 6 6 2 4 5 5 5 5 4 4 4 4 4 1 1 1 1 1 3
 [991] 3 3 3 3 6 6 6 6 3 3
 [ reached getOption("max.print") -- omitted 28568 entries ]

Within cluster sum of squares by cluster:
[1]  5829982  6221229  6168480 14229777 12246331  6270764
 (between_SS / total_SS =  86.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

km$withinss
km$tot.withinss
km$iter

```
> km$withinss
[1]  5829982  6221229  6168480 14229777 12246331  6270764
> km$tot.withinss
[1] 50966563
> km$iter
[1] 5
```

km$centers

```
> km$centers
    temp_c    is_day wind_mph wind_degree pressure_mb    precip_mm humidity     cloud feelslike_c windchill_c heatindex_c
1 23.76345 0.4074885 4.988210    67.19757    1008.079 2.037662e-01 86.42960 73.221071    26.11781    23.74238    26.27801
2 25.58243 0.4418233 4.712195    53.67393    1007.659 3.358657e-04 59.10296 15.510396    27.11116    25.54692    27.25076
3 25.66590 0.4914537 5.785515   319.66907    1008.814 4.426079e-02 69.61533 30.894449    27.49124    25.66587    27.64367
4 24.66348 0.4946856 4.591839   144.20321    1008.226 1.416754e-01 78.24500 56.351108    26.82275    24.64745    26.95251
5 27.21614 0.5479734 5.579986   250.71147    1009.087 1.602931e-05 51.05381  7.861919    28.26123    27.21287    28.38951
6 24.04097 0.4863360 4.763411   238.46382    1008.194 2.379099e-01 86.65764 75.485071    26.41121    24.03902    26.55873
  dewpoint_c   vis_km gust_mph
1   21.19343 8.575264 7.806991
2   15.85356 9.961895 7.699420
3   18.76155 9.628846 8.838890
4   19.97337 8.978743 7.189587
5   14.52207 9.988825 8.636341
6   21.50610 8.327505 7.464853
> 
```

km$cluster=as.factor(km$cluster)
km$cluster

```
> km$cluster=as.factor(km$cluster)
> km$cluster
  [1] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4 5 5 5 5 5 3 3 3 3 3 3 3 3 3 3 3 1 4 4 4 4 4 5 5 5 5 5 4 2 6 6 6 6 6 6 6 6 6 6
 [59] 6 6 6 6 6 6 2 4 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 2 4 5 5 5 5 5 5 5 5 5 1 1 1 1 1 1 2 1 4 4 4 4 4 4 4 4 4 4 4 4
[117] 4 4 4 4 1 1 1 1 1 1 2 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 4 2 6 2 4 5 4 3 6 2 4 5 4 2
[175] 6 2 4 5 5 5 5 5 4 4 4 4 4 4 5 5 4 1 5 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 4 5 5 5 5 4 4 5 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[233] 1 4 5 5 5 5 4 4 1 1 2 3 1 4 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 1 1 3 3 3 3 3 3 3 2 1 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 3 3
[291] 4 5 5 5 5 4 4 4 4 4 4 4 4 4 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 2 1 4 4 4 4 4 4 4 4 1 1 1
[349] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 1 1 1
[407] 1 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 6 6 6 6 6 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 6 6 6 6 6 6 6 3 2 2 2 3 3 3 3
[465] 3 1 1 2 2 2 2 1 1 1 1 1 1 1 1 3 4 4 4 5 4 1 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 3 3 1 1 1 1 1 2 3 6 6 6 6 6 6 6 3 3 3 6
[523] 6 3 3 3 3 3 2 3 3 3 3 3 3 3 6 6 6 6 6 6 6 6 2 4 5 4 4 1 1 2 3 3 3 3 6 6 6 6 6 6 6 3 3 3 6 6 6 6 2 2 2 2 2 2 6 6 6 6
[581] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 6 6 2 2 2 2 2 2 2 1 1 1 1 2 2 2 6 6 3 3 6 6 6 6 6 6 6 6 6 6 3 6
[639] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 6 6 6 6 6 6 6 6 6 2 2 2 2 6 6 6 6 6 6 6 6 3 2 4 5 4 4 4 1 1 1 1 1
[697] 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 2 2 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 1 5
[755] 4 4 4 4 4 5 4 4 4 1 1 1 1 3 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[813] 6 6 6 6 1 1 1 2 2 2 1 4 4 5 5 5 4 4 4 4 4 4 4 4 4 4 1 1 2 2 2 6 6 6 6 6 6 6 2 4 5 5 4 4 4 4 4 1 1 1 1 1 6 6 6 6 6 6
[871] 6 6 6 6 6 6 6 2 4 5 4 2 6 6 6 6 2 4 3 3 3 3 3 3 3 3 1 4 4 4 4 4 4 4 4 1 1 1 2 2 2 3 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
[929] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 3 3 3 3 3 3 3 6 6 6 6 6 6 2 4 5 5 5 5 4 4 4 4 4 1 1
[987] 1 1 1 3 3 3 3 3 6 6 6 6 3 3
 [ reached getOption("max.print") -- omitted 28568 entries ]
Levels: 1 2 3 4 5 6
> 
```

```
library(ggfortify)
autoplot(km,forecast_new,frame=TRUE)
```