HW9

Thulasiram Ruppa Krishnan

March 23, 2019

```
library(ggplot2)
library(reshape2)
library(stringr)
library(scales)
library(plyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:plyr':
##
##     here
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:scales':
##
##     alpha
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
# For Grid Arrange
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.5.3
```

```
library(grid)
library(ggplot2)
library(lattice)
```

```
rm(list=ls())
# Step  1:  Load    the data
my_aq<- airquality

# Step  2:  Clean   the data
my_aq<-na.omit(my_aq)

my_aq$Month<-as.factor(my_aq$Month)

my_aq$Day<-as.factor(my_aq$Day)

my_aq<-cbind.data.frame(my_aq,"date"=as.Date(gsub(" ","",paste(str_pad(my_aq$Month,2,side="left"
,pad = "0"),"-",str_pad(my_aq$Day,2,side="left",pad = "0"),"-1973")),"%m-%d-%Y"))

my_aq.m <- melt(my_aq,id.vars = "date", measure.vars = c("Ozone", "Solar.R","Wind","Temp"))
my_aq.m <- ddply(my_aq.m, .(variable), transform, rescale = rescale(value))

str(my_aq)
```

```
## 'data.frame':    111 obs. of  7 variables:
##  $ Ozone  : int  41 36 12 18 23 19 8 16 11 14 ...
##  $ Solar.R: int  190 118 149 313 299 99 19 256 290 274 ...
##  $ Wind   : num  7.4 8 12.6 11.5 8.6 13.8 20.1 9.7 9.2 10.9 ...
##  $ Temp   : int  67 72 74 62 65 59 61 69 66 68 ...
##  $ Month  : Factor w/ 5 levels "5","6","7","8",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Day    : Factor w/ 31 levels "1","2","3","4",..: 1 2 3 4 7 8 9 12 13 14 ...
##  $ date   : Date, format: "1973-05-01" "1973-05-02" ...
```

```
summary(my_aq)
```

```
##      Ozone          Solar.R           Wind            Temp       Month
##  Min.   :  1.0   Min.   :  7.0   Min.   : 2.30   Min.   :57.00   5:24
##  1st Qu.: 18.0   1st Qu.:113.5   1st Qu.: 7.40   1st Qu.:71.00   6: 9
##  Median : 31.0   Median :207.0   Median : 9.70   Median :79.00   7:26
##  Mean   : 42.1   Mean   :184.8   Mean   : 9.94   Mean   :77.79   8:23
##  3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:11.50   3rd Qu.:84.50   9:29
##  Max.   :168.0   Max.   :334.0   Max.   :20.70   Max.   :97.00
##
##       Day          date
##  7      : 5   Min.   :1973-05-01
##  9      : 5   1st Qu.:1973-06-14
##  13     : 5   Median :1973-07-28
##  16     : 5   Mean   :1973-07-22
##  17     : 5   3rd Qu.:1973-09-01
##  18     : 5   Max.   :1973-09-30
##  (Other):81
```

```
#Create trainign and test data set
nrows <-nrow(my_aq)
nrows
```

```
## [1] 111
```

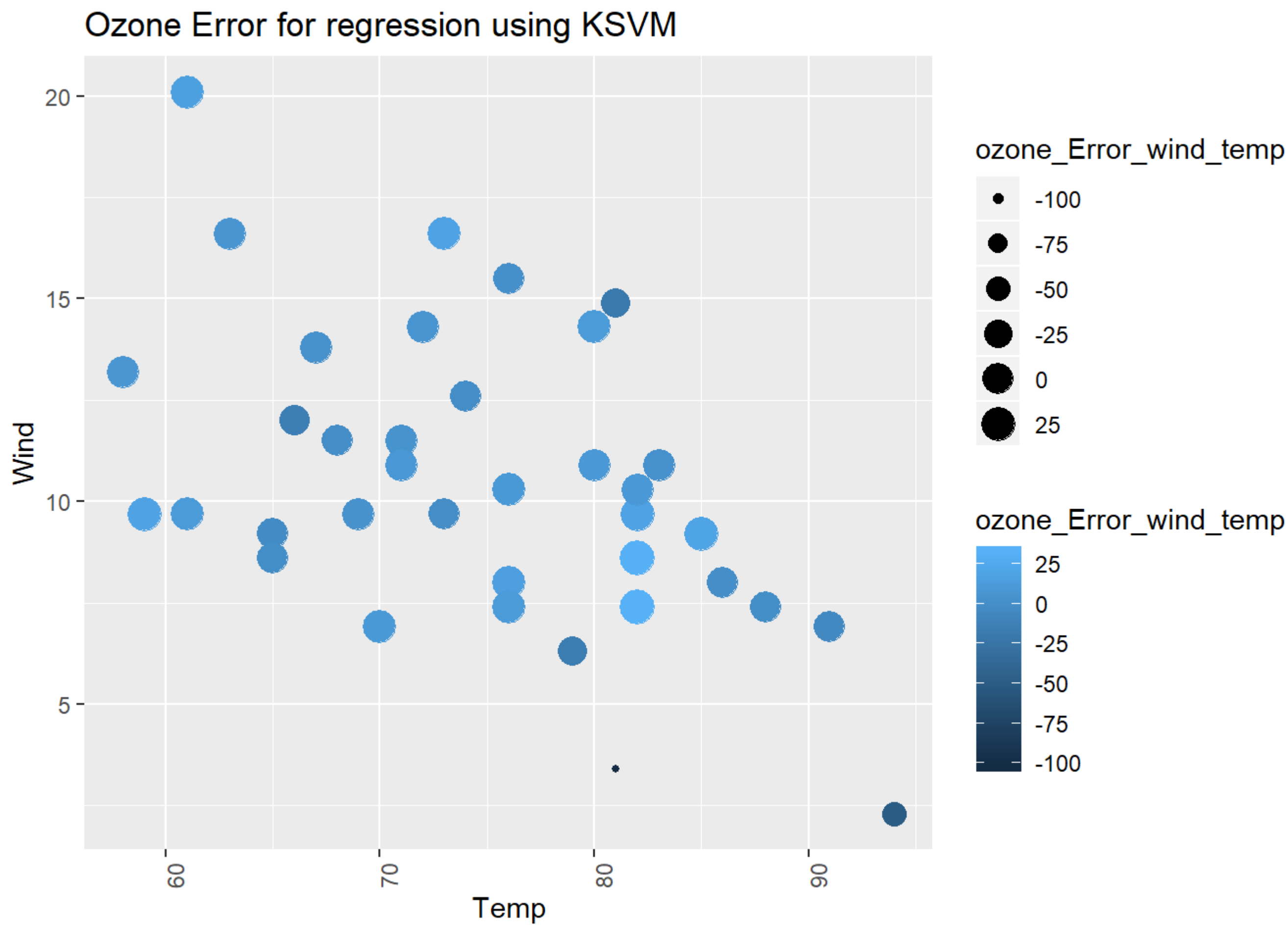
```
cutpoint <-floor(nrows/3*2)
cutpoint
```

```
## [1] 74
```

```
rand <- sample(1:nrows)

my_aq.train <- my_aq[rand[1:cutpoint],]
my_aq.test <- my_aq[rand[(cutpoint+1):nrows],]
```

```
#Build a model by using all variable
mq_aq.model.all.var <- ksvm(Ozone ~ Wind+Temp+Solar.R+Month, data=my_aq.train)
#mq_aq.model.all.var
# Test the model using test dataset
my_aq.test$ozone_pred_all_var <- predict(mq_aq.model.all.var,my_aq.test)

# store the prediction results for test dataset
my_aq.test.results.all.var <- table(my_aq.test$ozone_pred_all_var ,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_all_var <- (my_aq.test$ozone_pred_all_var-my_aq.test$Ozone)^2
mse.ozone.all.var <- mean(my_aq.test$ozone_squaredError_all_var)

#Build a model by using wind,temp,solar variable
mq_aq.model.wind.temp.solar <- ksvm(Ozone ~ Wind+Temp+Solar.R, data=my_aq.train)
#mq_aq.model.all.var
# Test the model using test dataset
my_aq.test$ozone_pred_wind_temp_solar <- predict(mq_aq.model.wind.temp.solar,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.temp.solar<- table(my_aq.test$ozone_pred_wind_temp_solar,my_aq.test$Ozon
e)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredErrorwind_temp_solar <- (my_aq.test$ozone_pred_wind_temp_solar-my_aq.tes
t$Ozone)^2
mse.ozone.wind.temp.solar <- mean(my_aq.test$ozone_squaredErrorwind_temp_solar)

#Build a model by using Temp and Wind variable
mq_aq.model.wind.temp <- ksvm(Ozone ~ Wind+Temp, data=my_aq.train)
#mq_aq.model.wind.temp
# Test the model using test dataset
my_aq.test$ozone_wind_temp <- predict(mq_aq.model.wind.temp,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.temp <- table(my_aq.test$ozone_wind_temp,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_wind_temp <- (my_aq.test$ozone_wind_temp-my_aq.test$Ozone)^2
my_aq.test$ozone_Error_wind_temp <- (my_aq.test$ozone_wind_temp-my_aq.test$Ozone)
mse.ozone.wind.temp <- mean(my_aq.test$ozone_squaredError_wind_temp)


#Build a model by using Temp variable
mq_aq.model.temp <- ksvm(Ozone ~ Temp, data=my_aq.train)
#mq_aq.model.temp
# Test the model using test dataset
my_aq.test$ozone_temp <- predict(mq_aq.model.temp,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.temp <- table(my_aq.test$ozone_temp,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_temp <- (my_aq.test$ozone_temp-my_aq.test$Ozone)^2
mse.ozone.temp <- mean(my_aq.test$ozone_squaredError_temp)

#Build a model by using Wind variable
```

```
mq_aq.model.wind <- ksvm(Ozone ~ Wind, data=my_aq.train)
#mq_aq.model.wind
# Test the model using test dataset
my_aq.test$ozone_wind <- predict(mq_aq.model.wind,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind <- table(my_aq.test$ozone_wind,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_wind <- (my_aq.test$ozone_wind-my_aq.test$Ozone)^2
mse.ozone.wind <- mean(my_aq.test$ozone_squaredError_wind)


#Build a model by using Solar.R variable
mq_aq.model.solar <- ksvm(Ozone ~ Solar.R, data=my_aq.train)
#mq_aq.model.solar
# Test the model using test dataset
my_aq.test$ozone_solar <- predict(mq_aq.model.solar,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.solar <- table(my_aq.test$ozone_solar,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_solar <- (my_aq.test$ozone_solar-my_aq.test$Ozone)^2
mse.ozone.solar <- mean(my_aq.test$ozone_squaredError_solar)


#scatter    chart (using    ggplot   geom_point),    with    the x-axis   representing    the tem
p,   the
#y-axis representing    the wind,   the size and color   of   each    dot representing    the erro
r

ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=ozone_Error_wind_temp,color=ozone_Error_wind_temp)) + theme
(legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Temp", y="Wind",title = "Ozone Error for regression using KSVM")
```

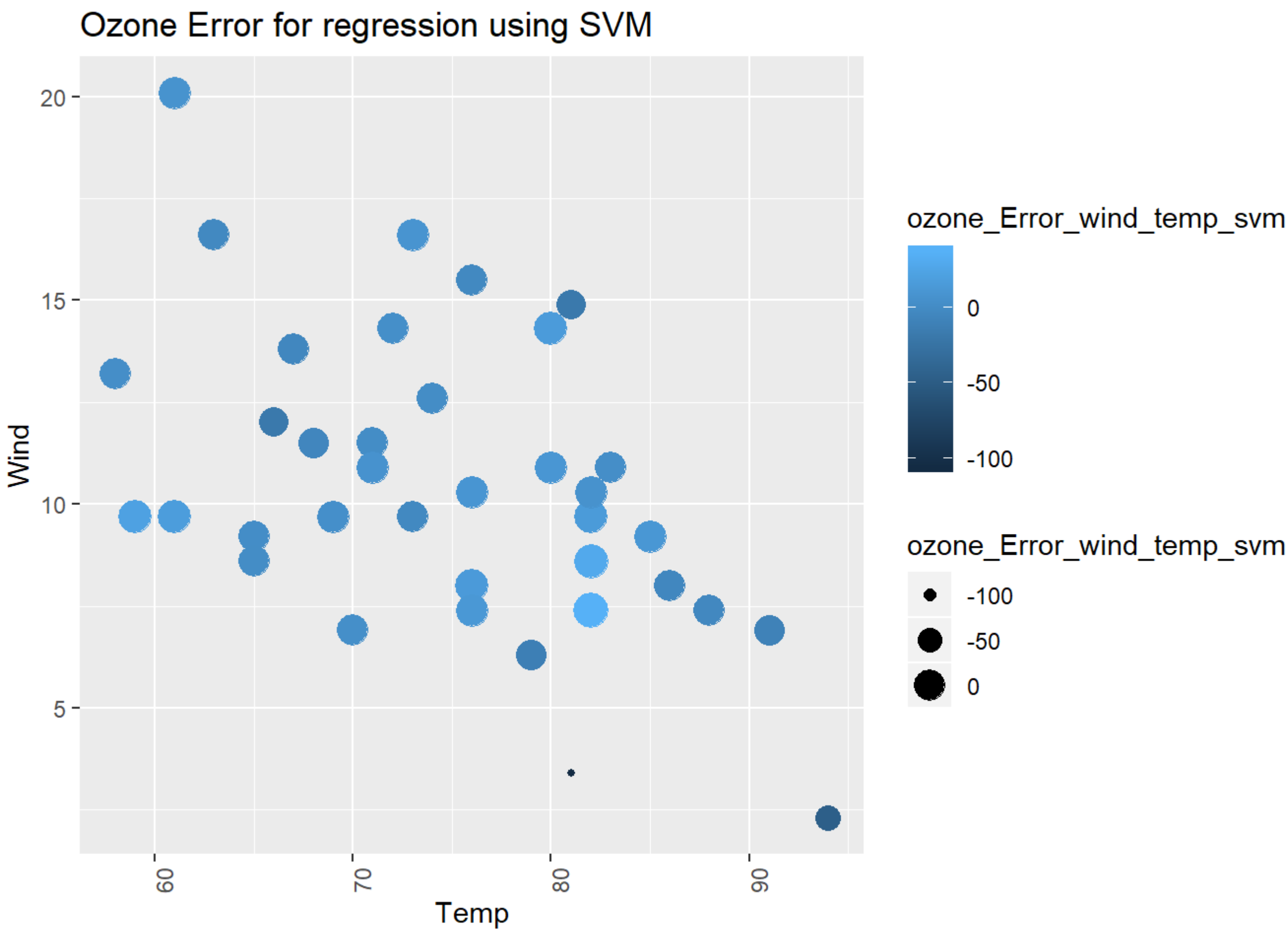



```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.5.3
```

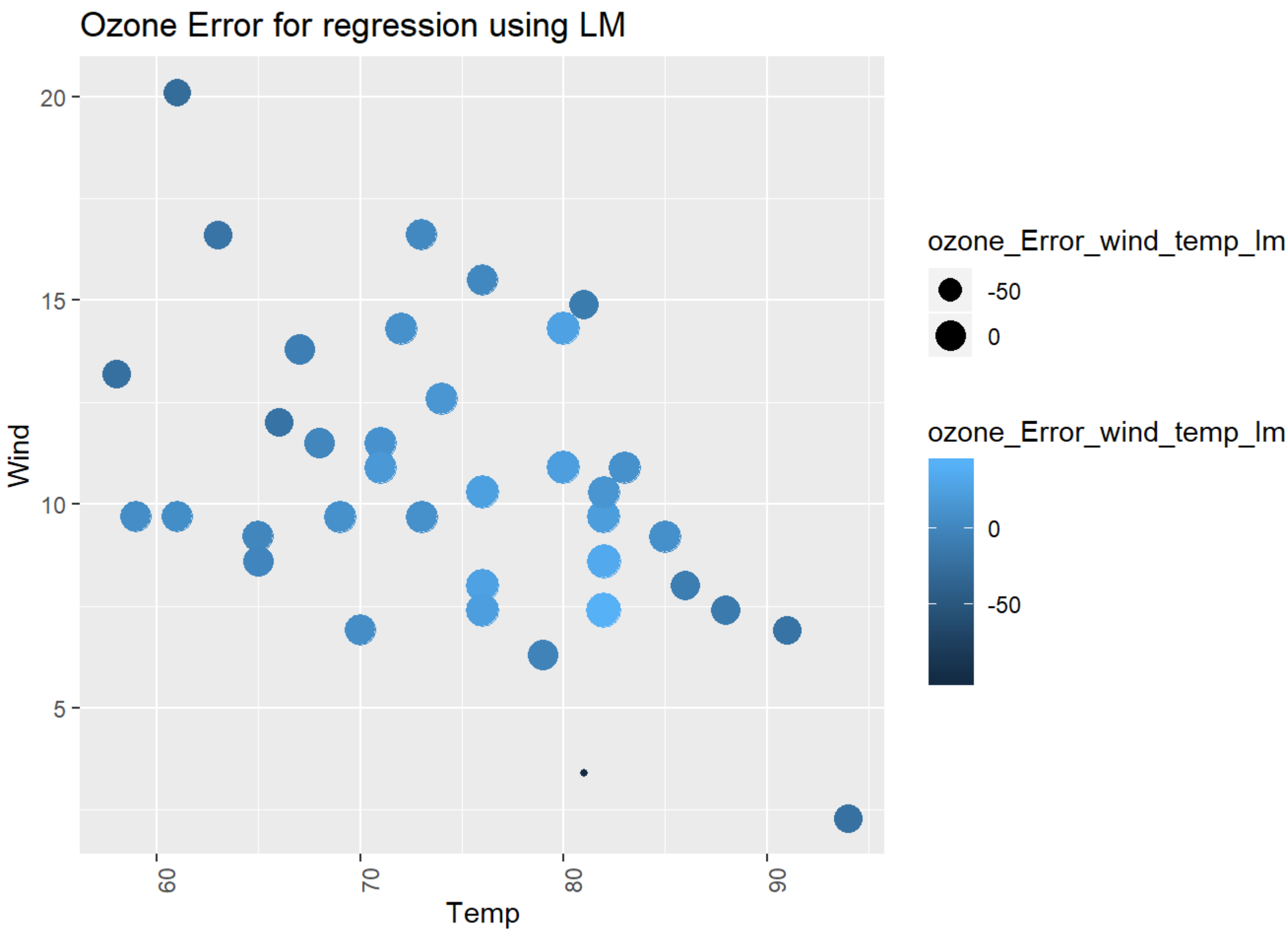
```
#Build a model by using all variable
mq_aq.svm.model.all.var <- svm(Ozone ~ Wind+Temp+Solar.R+Month, data=my_aq.train)
#mq_aq.svm.model.all.var
# Test the model using test dataset
my_aq.test$ozone_pred_all_var_svm <- predict(mq_aq.svm.model.all.var,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.all.var.svm <- table(my_aq.test$ozone_pred_all_var_svm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_all_var_svm <- (my_aq.test$ozone_pred_all_var_svm-my_aq.test$Ozon
e)^2
mse.ozone.all.var.svm <- mean(my_aq.test$ozone_squaredError_all_var_svm)


#Build a model by using wind,temp,solar variable
mq_aq.svm.model.wind.temp.solar <- svm(Ozone ~ Wind+Temp+Solar.R, data=my_aq.train)
#mq_aq.model.all.var
# Test the model using test dataset
my_aq.test$ozone_pred_wind_temp_solar_svm <- predict(mq_aq.svm.model.wind.temp.solar,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.temp.solar.svm<- table(my_aq.test$ozone_pred_wind_temp_solar_svm,my_aq.t
est$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredErrorwind_temp_solar_svm <- (my_aq.test$ozone_pred_wind_temp_solar_svm-m
y_aq.test$Ozone)^2
mse.ozone.wind.temp.solar.svm <- mean(my_aq.test$ozone_squaredErrorwind_temp_solar_svm)


#Build a model by using Temp and Wind variable
mq_aq.svm.model.wind.temp <- svm(Ozone ~ Wind+Temp, data=my_aq.train)
#mq_aq.model.wind.temp
# Test the model using test dataset
my_aq.test$ozone_wind_temp_svm <- predict(mq_aq.svm.model.wind.temp,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.temp.svm <- table(my_aq.test$ozone_wind_temp_svm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_wind_temp_svm <- (my_aq.test$ozone_wind_temp_svm-my_aq.test$Ozone)
^2
my_aq.test$ozone_Error_wind_temp_svm <- (my_aq.test$ozone_wind_temp_svm-my_aq.test$Ozone)
mse.ozone.wind.temp.svm <- mean(my_aq.test$ozone_squaredError_wind_temp_svm)


#Build a model by using Temp variable
mq_aq.svm.model.temp <- svm(Ozone ~ Temp, data=my_aq.train)
#mq_aq.model.temp
# Test the model using test dataset
my_aq.test$ozone_temp_svm <- predict(mq_aq.svm.model.temp,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.temp.svm <- table(my_aq.test$ozone_temp_svm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
```

```
my_aq.test$ozone_squaredError_temp_svm <- (my_aq.test$ozone_temp_svm-my_aq.test$Ozone)^2
mse.ozone.temp.svm <- mean(my_aq.test$ozone_squaredError_temp_svm)

#Build a model by using Wind variable
mq_aq.svm.model.wind <- svm(Ozone ~ Wind, data=my_aq.train)
#mq_aq.model.wind
# Test the model using test dataset
my_aq.test$ozone_wind_svm <- predict(mq_aq.svm.model.wind,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.svm <- table(my_aq.test$ozone_wind_svm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_wind_svm <- (my_aq.test$ozone_wind_svm-my_aq.test$Ozone)^2
mse.ozone.wind.svm <- mean(my_aq.test$ozone_squaredError_wind_svm)


#Build a model by using Solar.R variable
mq_aq.svm.model.solar <- svm(Ozone ~ Solar.R, data=my_aq.train)
#mq_aq.model.solar
# Test the model using test dataset
my_aq.test$ozone_solar_svm <- predict(mq_aq.svm.model.solar,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.solar.svm <- table(my_aq.test$ozone_solar_svm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_solar_svm <- (my_aq.test$ozone_solar_svm-my_aq.test$Ozone)^2
mse.ozone.solar.svm <- mean(my_aq.test$ozone_squaredError_solar_svm)

#scatter    chart (using    ggplot   geom_point),    with    the x-axis   representing    the tem
p,   the
#y-axis representing    the wind,    the size and color   of   each    dot representing    the erro
r

ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=ozone_Error_wind_temp_svm,color=ozone_Error_wind_temp_svm))
 + theme(legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Temp", y="Wind",title = "Ozone Error for regression using SVM")
```


Ozone Error for regression using SVM
Wind
Temp
20
15
10
5
60
70
80
90
ozone_Error_wind_temp_svm
0
-50
-100
ozone_Error_wind_temp_svm
-100
-50
0

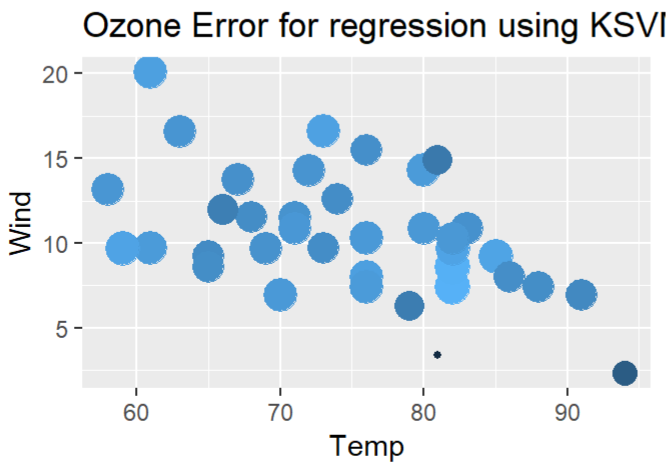
```
#Build a model by using all variable
mq_aq.lm.model.all.var <- lm(Ozone ~ Wind+Temp+Solar.R+Month, data=my_aq.train)
#mq_aq.lm.model.all.var
# Test the model using test dataset
#not able to predict the test data with lm using all variable
#my_aq.test$ozone_pred_all_var_lm <- predict(mq_aq.lm.model.all.var,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
#my_aq.test.results.all.var.lm <- table(pred,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
#my_aq.test$ozone_squaredError_all_var_lm <- (my_aq.test$ozone_pred_all_var_lm-my_aq.test$Ozone)
^2
#mse.ozone.all.var.lm <- mean(my_aq.test$ozone_squaredError_all_var_lm)



#Build a model by using wind,temp,solar variable
mq_aq.lm.model.wind.temp.solar <- svm(Ozone ~ Wind+Temp+Solar.R, data=my_aq.train)
#mq_aq.model.all.var
# Test the model using test dataset
my_aq.test$ozone_pred_wind_temp_solar_lm <- predict(mq_aq.lm.model.wind.temp.solar,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.temp.solar.lm<- table(my_aq.test$ozone_pred_wind_temp_solar_lm,my_aq.tes
t$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredErrorwind_temp_solar_lm <- (my_aq.test$ozone_pred_wind_temp_solar_lm-my_
aq.test$Ozone)^2
mse.ozone.wind.temp.solar.lm <- mean(my_aq.test$ozone_squaredErrorwind_temp_solar_lm)


#Build a model by using Temp and Wind variable
mq_aq.lm.model.wind.temp <- lm(Ozone ~ Wind+Temp, data=my_aq.train)
#mq_aq.model.wind.temp
# Test the model using test dataset
my_aq.test$ozone_wind_temp_lm <- predict(mq_aq.lm.model.wind.temp,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.temp.lm <- table(my_aq.test$ozone_wind_temp_lm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_wind_temp_lm <- (my_aq.test$ozone_wind_temp_lm-my_aq.test$Ozone)^2
my_aq.test$ozone_Error_wind_temp_lm <- (my_aq.test$ozone_wind_temp_lm-my_aq.test$Ozone)
mse.ozone.wind.temp.lm <- mean(my_aq.test$ozone_squaredError_wind_temp_lm)


#Build a model by using Temp variable
mq_aq.lm.model.temp <- lm(Ozone ~ Temp, data=my_aq.train)
#mq_aq.model.temp
# Test the model using test dataset
my_aq.test$ozone_temp_lm <- predict(mq_aq.lm.model.temp,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.temp.lm <- table(my_aq.test$ozone_temp_lm,my_aq.test$Ozone)
```

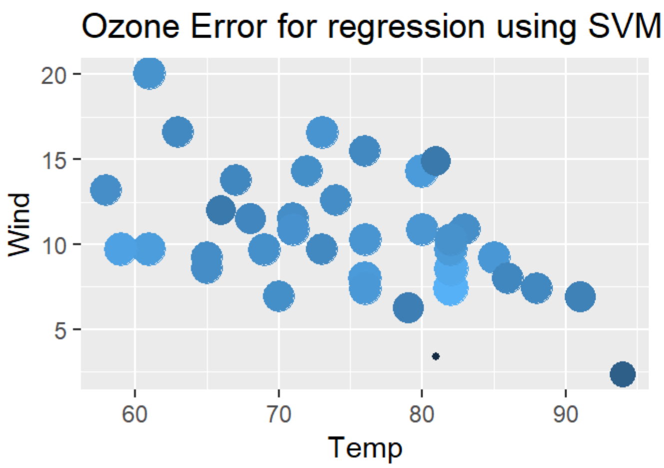
```
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_temp_lm <- (my_aq.test$ozone_temp_lm-my_aq.test$Ozone)^2
mse.ozone.temp.lm <- mean(my_aq.test$ozone_squaredError_temp_lm)

#Build a model by using Wind variable
mq_aq.lm.model.wind <- lm(Ozone ~ Wind, data=my_aq.train)
#mq_aq.model.wind
# Test the model using test dataset
my_aq.test$ozone_wind_lm <- predict(mq_aq.lm.model.wind,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.wind.lm <- table(my_aq.test$ozone_wind_lm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_wind_lm <- (my_aq.test$ozone_wind_lm-my_aq.test$Ozone)^2
mse.ozone.wind.lm <- mean(my_aq.test$ozone_squaredError_wind_lm)


#Build a model by using Solar.R variable
mq_aq.lm.model.solar <- lm(Ozone ~ Solar.R, data=my_aq.train)
#mq_aq.model.solar
# Test the model using test dataset
my_aq.test$ozone_solar_lm <- predict(mq_aq.lm.model.solar,my_aq.test)
#table(pred,my_aq.test$Ozone)
# store the prediction results for test dataset
my_aq.test.results.solar.lm <- table(my_aq.test$ozone_solar_lm,my_aq.test$Ozone)
# Square of Error - Support Vector Machine
my_aq.test$ozone_squaredError_solar_lm <- (my_aq.test$ozone_solar_lm-my_aq.test$Ozone)^2
mse.ozone.solar.lm <- mean(my_aq.test$ozone_squaredError_solar_lm)

#scatter    chart (using    ggplot  geom_point),    with    the x-axis  representing    the tem
p,   the
#y-axis representing    the wind,   the size and color  of  each    dot representing    the erro
r

ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=ozone_Error_wind_temp_lm,color=ozone_Error_wind_temp_lm)) +
 theme(legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Temp", y="Wind",title = "Ozone Error for regression using LM")
```

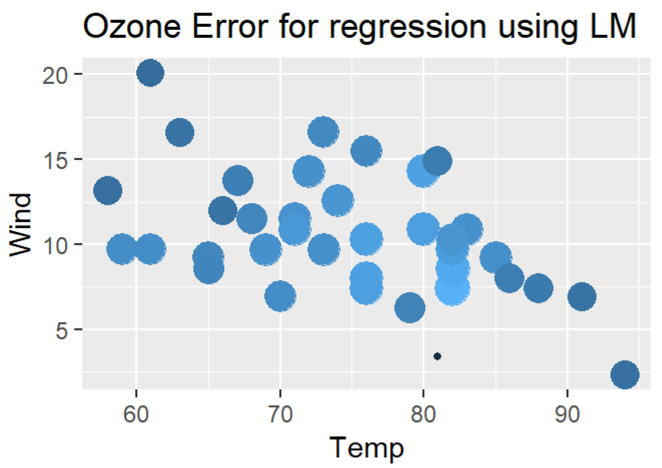



```
ggplot_ksvm <-ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=ozone_Error_wind_temp,color=ozone_Error_wind_temp)) +
  labs(x = "Temp", y="Wind",size="Error",color="Error", title = "Ozone Error for regression usin
g KSVM") + guides(size=FALSE, color=FALSE)

ggplot_svm <-ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=ozone_Error_wind_temp_svm,color=ozone_Error_wind_temp_svm))
 +
  labs(x = "Temp", y="Wind",size="Error",color="Error",title = "Ozone Error for regression using
SVM") + guides(size=FALSE, color=FALSE)

ggplot_lm <-ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=ozone_Error_wind_temp_lm,color=ozone_Error_wind_temp_lm)) +
  labs(x = "Temp", y="Wind",size="Error",color="Error",title = "Ozone Error for regression using
LM") + guides(size=FALSE, color=FALSE)

#Arrange all the scatter plots from different models in one Grid
text <-textGrob("This Grid showing the Error plot (predicted - Actual) from various regression m
ethod. Regression method used is KSVM, SVM and LM ")
#leg <- ggplot(my_aq.test) + theme(aes(size=ozone_Error_wind_temp_lm,color=ozone_Error_wind_temp
_lm),legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1))

grid.arrange(text, ggplot_ksvm, ggplot_svm, ggplot_lm, ncol=2)
```

- Actual) from various regression method. Regre








```
#Calculate Mean Ozone
mean_ozone <- mean(my_aq.train$Ozone)

my_aq.train$mean_oz <- mean_ozone

# function to determine wheather the ozone level is good/bad
flag_good_ozone <-function(Ozone)  {
  if(Ozone < mean_ozone) 0 else 1
}

flag_pred_good_ozone <-function(goodOzone)  {
  if(goodOzone < 0.5) 0 else 1
}

flag_pred_good_ozone_nb <-function(goodOzone)  {
  if(goodOzone ==1) 0 else 1
}


flag_pred_results <-function(goodOzone,goodOzonePred)  {
  if(goodOzone == goodOzonePred) 0 else 1
}

# Create a varaiable to house good/nbad ozone indicator
my_aq.train$goodOzone <- sapply(my_aq.train$Ozone,flag_good_ozone)
my_aq.test$goodOzone <- sapply(my_aq.test$Ozone,flag_good_ozone)
```

```
#Build a model by using all variable
mq_aq.goz.model.all.var <- ksvm(goodOzone ~ Wind+Temp+Solar.R+ Month, data=my_aq.train)
#mq_aq.model.all.var
# Test the model using test dataset
my_aq.test$goodOzone_pred_all_var <- predict(mq_aq.goz.model.all.var,my_aq.test)
my_aq.test$goodOzone_pred_all_var <- sapply(my_aq.test$goodOzone_pred_all_var,flag_pred_good_ozo
ne)
my_aq.test$goodOzone_pred_all_var_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_pred_all_var)
#table(pred,my_aq.test$goodOzone)
# store the prediction results for test dataset
#my_aq.goz.test.results.all.var <- table(pred,my_aq.test$goodOzone)
# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_all_var <- (my_aq.test$goodOzone_pred_all_var-my_aq.test$goodO
zone)^2
mse.goodOzone.all.var <- mean(my_aq.test$goodOzone_squaredError_all_var)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_all_var_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 91.89189
```

```
#Build a model by using wind,temp,solar variable
mq_aq.goz.model.wind.temp.solar <- ksvm(goodOzone ~ Wind+Temp+Solar.R, data=my_aq.train)
#mq_aq.model.all.var
# Test the model using test dataset
my_aq.test$goodOzone_pred_wind_temp_solar <- predict(mq_aq.goz.model.wind.temp.solar,my_aq.test)
my_aq.test$goodOzone_pred_wind_temp_solar <- sapply(my_aq.test$goodOzone_pred_wind_temp_solar,fl
ag_pred_good_ozone)
my_aq.test$goodOzone_pred_wind_temp_solar_results <- mapply(flag_pred_results,my_aq.test$goodOzo
ne,my_aq.test$goodOzone_pred_wind_temp_solar)
#table(pred,my_aq.test$goodOzone)
# store the prediction results for test dataset
#my_aq.goz.test.results.wind.temp.solar<- table(pred,my_aq.test$goodOzone)
# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredErrorwind_temp_solar <- (my_aq.test$goodOzone_pred_wind_temp_solar-m
y_aq.test$goodOzone)^2
mse.goodOzone.wind.temp.solar <- mean(my_aq.test$goodOzone_squaredErrorwind_temp_solar)
# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_solar_results==0,1])/length(my_aq.test[,1
])*100
```

```
## [1] 97.2973
```

```
#Build a model by using Temp and Wind variable
mq_aq.goz.model.wind.temp <- ksvm(goodOzone ~ Wind+Temp, data=my_aq.train)
#mq_aq.model.wind.temp
# Test the model using test dataset
my_aq.test$goodOzone_wind_temp <- predict(mq_aq.goz.model.wind.temp,my_aq.test)
my_aq.test$goodOzone_wind_temp <- sapply(my_aq.test$goodOzone_wind_temp,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_wind_temp_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_
aq.test$goodOzone_wind_temp)
#table(pred,my_aq.test$goodOzone)
# store the prediction results for test dataset
#my_aq.goz.test.results.wind.temp <- table(pred,my_aq.test$goodOzone)
# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind_temp <- (my_aq.test$goodOzone_wind_temp-my_aq.test$goodOz
one)^2
my_aq.test$goodOzone_Error_wind_temp <- (my_aq.test$goodOzone_wind_temp-my_aq.test$goodOzone)
mse.goodOzone.wind.temp <- mean(my_aq.test$goodOzone_squaredError_wind_temp)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 94.59459
```

```
#Build a model by using Temp variable
mq_aq.goz.model.temp <- ksvm(goodOzone ~ Temp, data=my_aq.train)
#mq_aq.model.temp
# Test the model using test dataset
my_aq.test$goodOzone_temp <- predict(mq_aq.goz.model.temp,my_aq.test)
my_aq.test$goodOzone_temp <- sapply(my_aq.test$goodOzone_temp,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_temp_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_aq.te
st$goodOzone_temp)
#table(pred,my_aq.test$goodOzone)
# store the prediction results for test dataset
#my_aq.goz.test.results.temp <- table(pred,my_aq.test$goodOzone)
# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_temp <- (my_aq.test$goodOzone_temp-my_aq.test$goodOzone)^2
mse.goodOzone.temp <- mean(my_aq.test$goodOzone_squaredError_temp)
# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_temp_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 91.89189
```

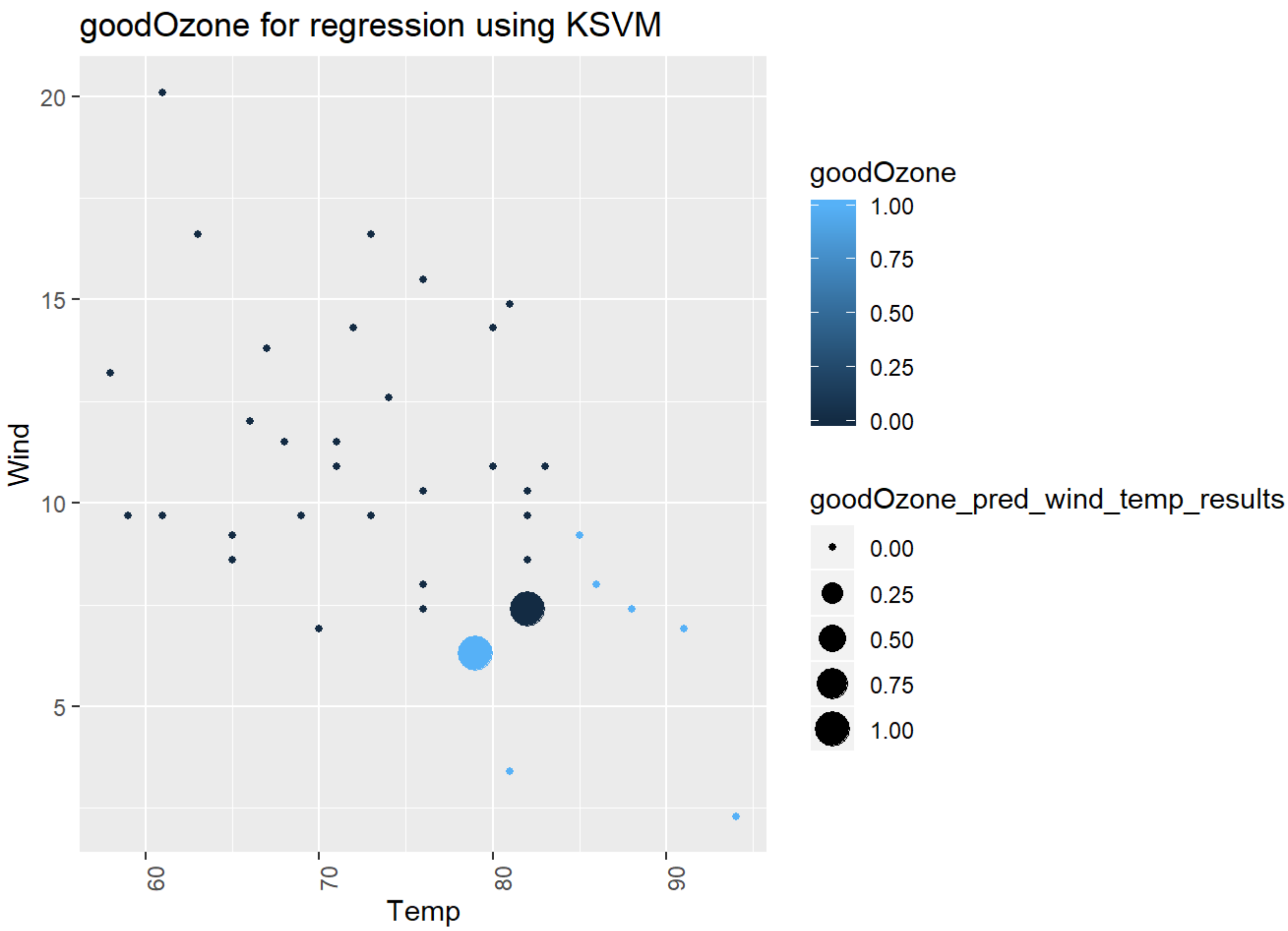
```
#Build a model by using Wind variable
mq_aq.goz.model.wind <- ksvm(goodOzone ~ Wind, data=my_aq.train)
#mq_aq.model.wind
# Test the model using test dataset
my_aq.test$goodOzone_wind <- predict(mq_aq.goz.model.wind,my_aq.test)
my_aq.test$goodOzone_wind <- sapply(my_aq.test$goodOzone_wind,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_wind_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_aq.te
st$goodOzone_wind)
#table(pred,my_aq.test$goodOzone)
# store the prediction results for test dataset
#my_aq.goz.test.results.wind <- table(pred,my_aq.test$goodOzone)
# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind <- (my_aq.test$goodOzone_wind-my_aq.test$goodOzone)^2
mse.goodOzone.wind <- mean(my_aq.test$goodOzone_squaredError_wind)
# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 86.48649
```

```
#Build a model by using Solar.R variable
mq_aq.goz.model.solar <- ksvm(goodOzone ~ Solar.R, data=my_aq.train)
#mq_aq.model.solar
# Test the model using test dataset
my_aq.test$goodOzone_solar <- predict(mq_aq.goz.model.solar,my_aq.test)
my_aq.test$goodOzone_solar <- sapply(my_aq.test$goodOzone_solar,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_solar_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_aq.t
est$goodOzone_solar)
#table(pred,my_aq.test$goodOzone)
# store the prediction results for test dataset
#my_aq.goz.test.results.solar <- table(pred,my_aq.test$goodOzone)
# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_solar <- (my_aq.test$goodOzone_solar-my_aq.test$goodOzone)^2
mse.goodOzone.solar <- mean(my_aq.test$goodOzone_squaredError_solar)
# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_solar_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 70.27027
```

```
#scatter    chart (using    ggplot  geom_point),    with    the x-axis  representing    the tem
p,   the
#y-axis representing    the wind,   the size and color  of  each    dot representing    the erro
r

ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_results,color=goodOzone)) + theme(l
egend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using KSVM")
```




```
library(e1071)
#Build a model by using all variable
mq_aq.goz.svm.model.all.var <- svm(goodOzone ~ Wind+Temp+Solar.R+ Month, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_pred_all_var_svm <- predict(mq_aq.goz.svm.model.all.var,my_aq.test)
my_aq.test$goodOzone_pred_all_var_svm <- sapply(my_aq.test$goodOzone_pred_all_var_svm,flag_pred_
good_ozone)
my_aq.test$goodOzone_pred_all_var_results_svm <- mapply(flag_pred_results,my_aq.test$goodOzone,m
y_aq.test$goodOzone_pred_all_var_svm)


# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_all_var_svm <- (my_aq.test$goodOzone_pred_all_var_svm-my_aq.te
st$goodOzone)^2
mse.goodOzone.all.var.svm <- mean(my_aq.test$goodOzone_squaredError_all_var_svm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_all_var_results_svm==0,1])/length(my_aq.test[,1])*10
0
```

```
## [1] 91.89189
```

```
#Build a model by using wind,temp,solar variable
mq_aq.goz.svm.model.wind.temp.solar <- svm(goodOzone ~ Wind+Temp+Solar.R, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_pred_wind_temp_solar_svm <- predict(mq_aq.goz.svm.model.wind.temp.solar,my_
aq.test)
my_aq.test$goodOzone_pred_wind_temp_solar_svm <- sapply(my_aq.test$goodOzone_pred_wind_temp_sola
r_svm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_wind_temp_solar_svm_results <- mapply(flag_pred_results,my_aq.test$goo
dOzone,my_aq.test$goodOzone_pred_wind_temp_solar_svm)


# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredErrorwind_temp_solar_svm <- (my_aq.test$goodOzone_pred_wind_temp_sol
ar_svm-my_aq.test$goodOzone)^2
mse.goodOzone.wind.temp.solar_svm <- mean(my_aq.test$goodOzone_squaredErrorwind_temp_solar_svm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_solar_svm_results==0,1])/length(my_aq.test
[,1])*100
```

```
## [1] 97.2973
```

```
#Build a model by using Temp and Wind variable
mq_aq.goz.svm.model.wind.temp <- svm(goodOzone ~ Wind+Temp, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_wind_temp_svm <- predict(mq_aq.goz.svm.model.wind.temp,my_aq.test)
my_aq.test$goodOzone_wind_temp_svm <- sapply(my_aq.test$goodOzone_wind_temp_svm,flag_pred_good_o
zone)
my_aq.test$goodOzone_pred_wind_temp_svm_results <- mapply(flag_pred_results,my_aq.test$goodOzon
e,my_aq.test$goodOzone_wind_temp_svm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind_temp_svm <- (my_aq.test$goodOzone_wind_temp_svm-my_aq.tes
t$goodOzone)^2
my_aq.test$goodOzone_Error_wind_temp_svm <- (my_aq.test$goodOzone_wind_temp_svm-my_aq.test$goodO
zone)
mse.goodOzone.wind.temp_svm <- mean(my_aq.test$goodOzone_squaredError_wind_temp_svm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_svm_results==0,1])/length(my_aq.test[,1])*
100
```

```
## [1] 94.59459
```

```
#Build a model by using Temp variable
mq_aq.goz.svm.model.temp <- svm(goodOzone ~ Temp, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_temp_svm <- predict(mq_aq.goz.svm.model.temp,my_aq.test)
my_aq.test$goodOzone_temp_svm <- sapply(my_aq.test$goodOzone_temp_svm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_temp_svm_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_temp_svm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_temp_svm <- (my_aq.test$goodOzone_temp_svm-my_aq.test$goodOzon
e)^2
mse.goodOzone.temp_svm <- mean(my_aq.test$goodOzone_squaredError_temp_svm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_temp_svm_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 91.89189
```

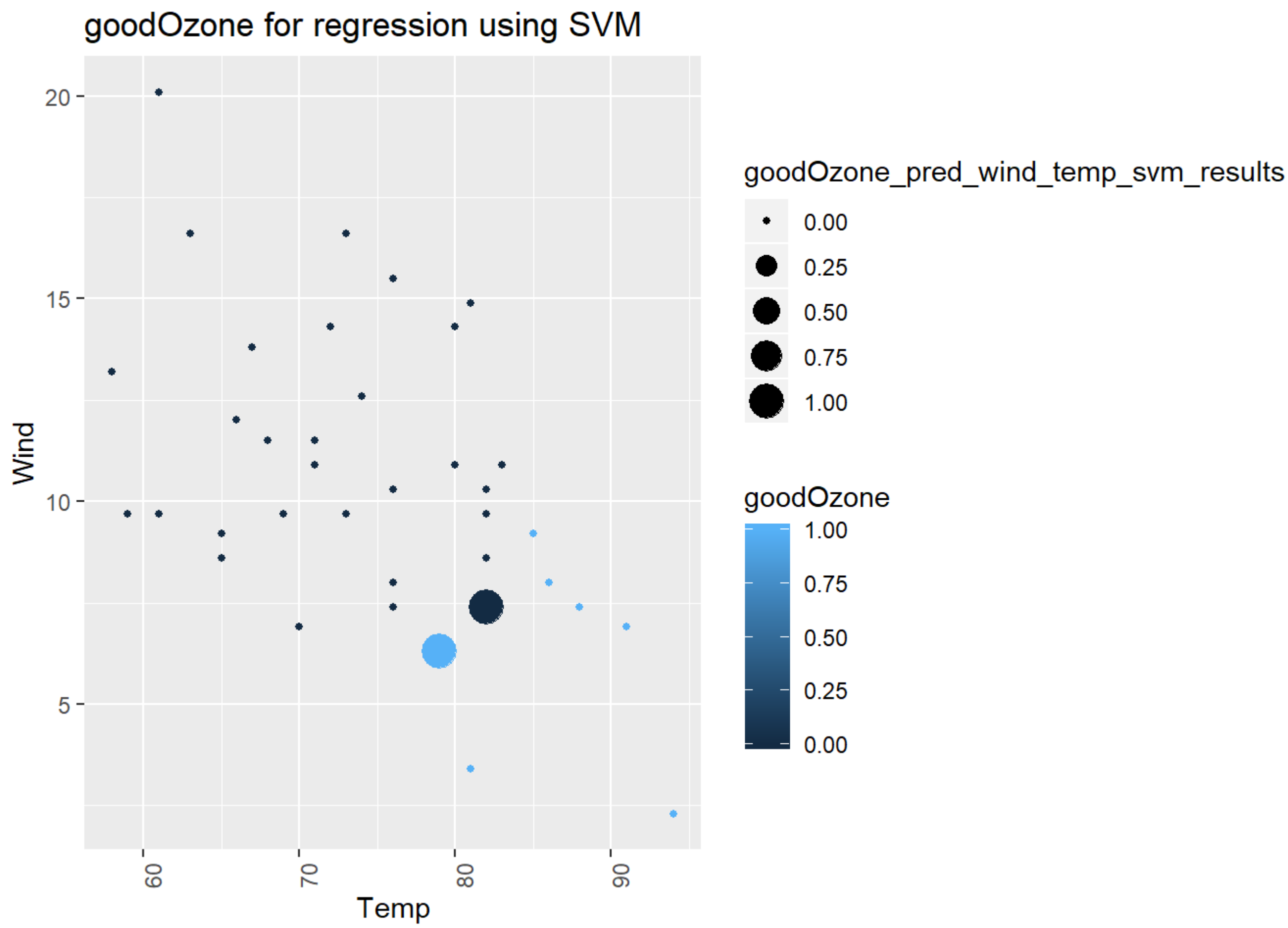
```
#Build a model by using Wind variable
mq_aq.goz.svm.model.wind <- svm(goodOzone ~ Wind, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_wind_svm <- predict(mq_aq.goz.svm.model.wind,my_aq.test)
my_aq.test$goodOzone_wind_svm <- sapply(my_aq.test$goodOzone_wind_svm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_wind_svm_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_wind_svm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind_svm <- (my_aq.test$goodOzone_wind_svm-my_aq.test$goodOzon
e)^2
mse.goodOzone.wind_svm <- mean(my_aq.test$goodOzone_squaredError_wind_svm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_svm_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 86.48649
```

```
#Build a model by using Solar.R variable
mq_aq.goz.svm.model.solar <- svm(goodOzone ~ Solar.R, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_solar_svm <- predict(mq_aq.goz.svm.model.solar,my_aq.test)
my_aq.test$goodOzone_solar_svm <- sapply(my_aq.test$goodOzone_solar_svm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_solar_svm_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_
aq.test$goodOzone_solar_svm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_solar_svm <- (my_aq.test$goodOzone_solar_svm-my_aq.test$goodOz
one)^2
mse.goodOzone.solar_svm <- mean(my_aq.test$goodOzone_squaredError_solar_svm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_solar_svm_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 81.08108
```

```
#scatter    chart (using    ggplot   geom_point),    with    the x-axis   representing    the tem
p,   the
#y-axis representing    the wind,   the size and color  of  each    dot representing    the erro
r

ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_svm_results,color=goodOzone)) + the
me(legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using SVM")
```




```
#Build a model by using all variable
mq_aq.goz.lm.model.all.var <- lm(goodOzone ~ Wind+Temp+Solar.R+ Month, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_pred_all_var_lm <- predict(mq_aq.goz.lm.model.all.var,my_aq.test)
my_aq.test$goodOzone_pred_all_var_lm <- sapply(my_aq.test$goodOzone_pred_all_var_lm,flag_pred_go
od_ozone)
my_aq.test$goodOzone_pred_all_var_results_lm <- mapply(flag_pred_results,my_aq.test$goodOzone,my
_aq.test$goodOzone_pred_all_var_lm)


# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_all_var_lm <- (my_aq.test$goodOzone_pred_all_var_lm-my_aq.test
$goodOzone)^2
mse.goodOzone.all.var.lm <- mean(my_aq.test$goodOzone_squaredError_all_var_lm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_all_var_results_lm==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 83.78378
```

```
#Build a model by using wind,temp,solar variable
mq_aq.goz.lm.model.wind.temp.solar <- lm(goodOzone ~ Wind+Temp+Solar.R, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_pred_wind_temp_solar_lm <- predict(mq_aq.goz.lm.model.wind.temp.solar,my_a
q.test)
my_aq.test$goodOzone_pred_wind_temp_solar_lm <- sapply(my_aq.test$goodOzone_pred_wind_temp_solar
_lm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_wind_temp_solar_lm_results <- mapply(flag_pred_results,my_aq.test$good
Ozone,my_aq.test$goodOzone_pred_wind_temp_solar_lm)


# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredErrorwind_temp_solar_lm <- (my_aq.test$goodOzone_pred_wind_temp_sola
r_lm-my_aq.test$goodOzone)^2
mse.goodOzone.wind.temp.solar_lm <- mean(my_aq.test$goodOzone_squaredErrorwind_temp_solar_lm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_solar_lm_results==0,1])/length(my_aq.test
[,1])*100
```

```
## [1] 91.89189
```

```
#Build a model by using Temp and Wind variable
mq_aq.goz.lm.model.wind.temp <- lm(goodOzone ~ Wind+Temp, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_wind_temp_lm <- predict(mq_aq.goz.lm.model.wind.temp,my_aq.test)
my_aq.test$goodOzone_wind_temp_lm <- sapply(my_aq.test$goodOzone_wind_temp_lm,flag_pred_good_ozo
ne)
my_aq.test$goodOzone_pred_wind_temp_lm_results <- mapply(flag_pred_results,my_aq.test$goodOzone,
my_aq.test$goodOzone_wind_temp_lm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind_temp_lm <- (my_aq.test$goodOzone_wind_temp_lm-my_aq.test
$goodOzone)^2
my_aq.test$goodOzone_Error_wind_temp_lm <- (my_aq.test$goodOzone_wind_temp_lm-my_aq.test$goodOzo
ne)
mse.goodOzone.wind.temp_lm <- mean(my_aq.test$goodOzone_squaredError_wind_temp_lm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_lm_results==0,1])/length(my_aq.test[,1])*1
00
```

```
## [1] 94.59459
```

```
#Build a model by using Temp variable
mq_aq.goz.lm.model.temp <- lm(goodOzone ~ Temp, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_temp_lm <- predict(mq_aq.goz.lm.model.temp,my_aq.test)
my_aq.test$goodOzone_temp_lm <- sapply(my_aq.test$goodOzone_temp_lm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_temp_lm_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_temp_lm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_temp_lm <- (my_aq.test$goodOzone_temp_lm-my_aq.test$goodOzone)
^2
mse.goodOzone.temp_lm <- mean(my_aq.test$goodOzone_squaredError_temp_lm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_temp_lm_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 91.89189
```

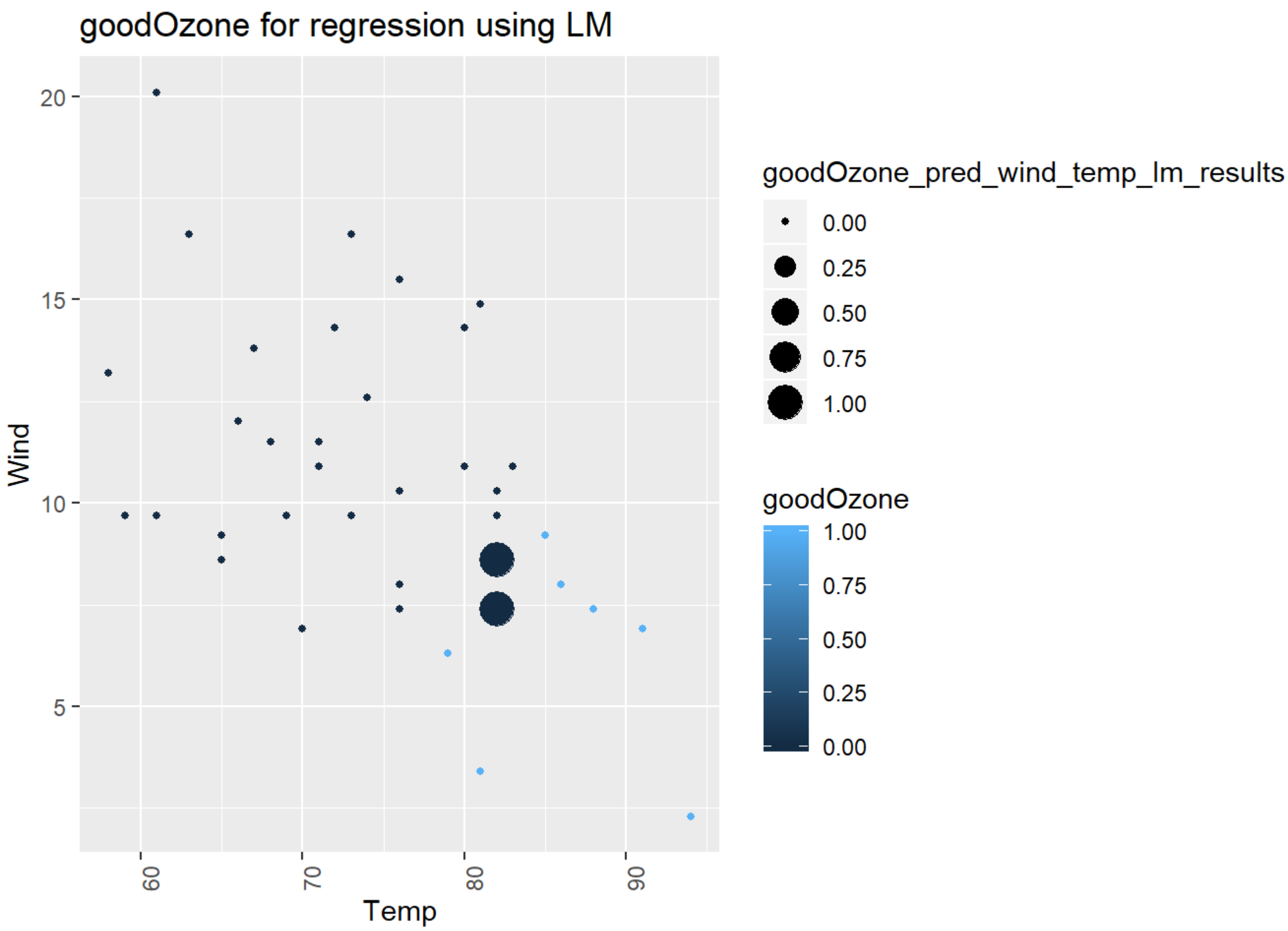
```
#Build a model by using Wind variable
mq_aq.goz.lm.model.wind <- lm(goodOzone ~ Wind, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_wind_lm <- predict(mq_aq.goz.lm.model.wind,my_aq.test)
my_aq.test$goodOzone_wind_lm <- sapply(my_aq.test$goodOzone_wind_lm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_wind_lm_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_wind_lm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind_lm <- (my_aq.test$goodOzone_wind_lm-my_aq.test$goodOzone)
^2
mse.goodOzone.wind_lm <- mean(my_aq.test$goodOzone_squaredError_wind_lm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_lm_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 86.48649
```

```
#Build a model by using Solar.R variable
mq_aq.goz.lm.model.solar <- lm(goodOzone ~ Solar.R, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_solar_lm <- predict(mq_aq.goz.lm.model.solar,my_aq.test)
my_aq.test$goodOzone_solar_lm <- sapply(my_aq.test$goodOzone_solar_lm,flag_pred_good_ozone)
my_aq.test$goodOzone_pred_solar_lm_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_solar_lm)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_solar_lm <- (my_aq.test$goodOzone_solar_lm-my_aq.test$goodOzon
e)^2
mse.goodOzone.solar_lm <- mean(my_aq.test$goodOzone_squaredError_solar_lm)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_solar_lm_results==0,1])/length(my_aq.test[,1])*100
```

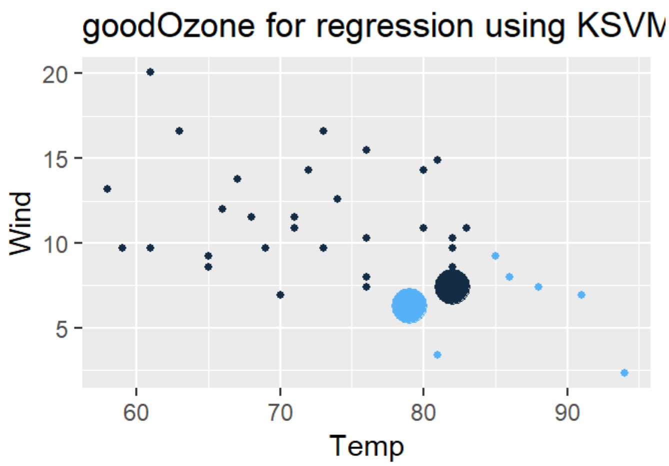
```
## [1] 70.27027
```

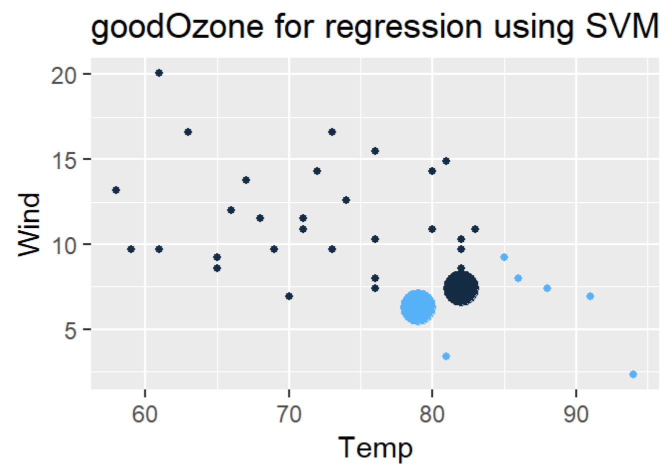
```
#scatter    chart (using    ggplot   geom_point),    with    the x-axis   representing    the tem
p,   the
#y-axis representing    the wind,   the size and color  of  each    dot representing    the erro
r

ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_lm_results,color=goodOzone)) + them
e(legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using LM")
```

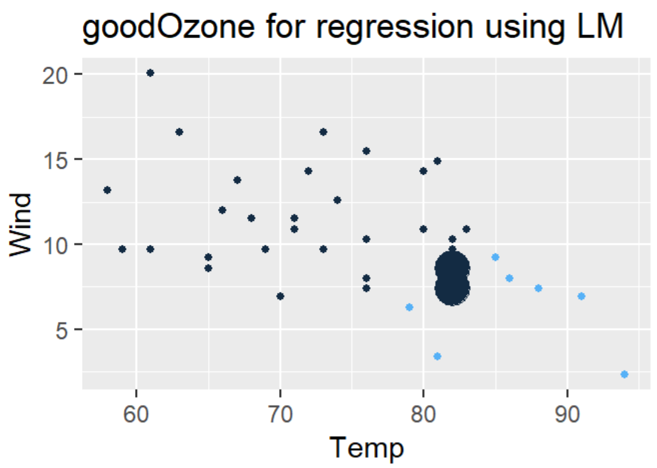



```
ggplot_ksvm_goodOzone <-ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_results,color=goodOzone)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using KSVM")+ guides(size=FALSE, c
olor=FALSE)

ggplot_svm_goodOzone <-ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_svm_results,color=goodOzone)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using SVM") + guides(size=FALSE, c
olor=FALSE)

ggplot_lm_goodOzone <-ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_lm_results,color=goodOzone)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using LM") + guides(size=FALSE, co
lor=FALSE)

#Arrange all the scatter plots from different models in one Grid
text <-textGrob("This Grid showing the accuracy of the goodOzone prediction between KSVM, SVM an
d LM ")
#leg <- ggplot(my_aq.test) + theme(aes(size=ozone_Error_wind_temp_lm,color=ozone_Error_wind_temp
_lm),legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1))

grid.arrange(text, ggplot_ksvm_goodOzone, ggplot_svm_goodOzone, ggplot_lm_goodOzone, ncol=2)
```

accuracy of the goodOzone prediction between








```
library(e1071)
#Build a model by using all variable
mq_aq.goz.nb.model.all.var <- naiveBayes(goodOzone ~ Wind+Temp+Solar.R+Month, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_pred_all_var_nb <- predict(mq_aq.goz.nb.model.all.var,my_aq.test[,1:5],type
="raw")
my_aq.test$goodOzone_pred_all_var_nb <- sapply(max.col(my_aq.test$goodOzone_pred_all_var_nb,"las
t"),flag_pred_good_ozone_nb)
my_aq.test$goodOzone_pred_all_var_results_nb <- mapply(flag_pred_results,my_aq.test$goodOzone,my
_aq.test$goodOzone_pred_all_var_nb)


# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_all_var_nb <- (my_aq.test$goodOzone_pred_all_var_nb-my_aq.test
$goodOzone)^2
mse.goodOzone.all.var.nb <- mean(my_aq.test$goodOzone_squaredError_all_var_nb)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_all_var_results_nb==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 91.89189
```

```
#Build a model by using wind,temp,solar variable
mq_aq.goz.nb.model.wind.temp.solar <- naiveBayes(goodOzone ~ Wind+Temp+Solar.R, data=my_aq.trai
n)

# Test the model using test dataset
my_aq.test$goodOzone_pred_wind_temp_solar_nb <- predict(mq_aq.goz.nb.model.wind.temp.solar,my_a
q.test[,2:4],type="raw")
my_aq.test$goodOzone_pred_wind_temp_solar_nb <- sapply(max.col(my_aq.test$goodOzone_pred_wind_te
mp_solar_nb,"last"),flag_pred_good_ozone_nb)
my_aq.test$goodOzone_pred_wind_temp_solar_nb_results <- mapply(flag_pred_results,my_aq.test$good
Ozone,my_aq.test$goodOzone_pred_wind_temp_solar_nb)


# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredErrorwind_temp_solar_nb <- (my_aq.test$goodOzone_pred_wind_temp_sola
r_nb-my_aq.test$goodOzone)^2
mse.goodOzone.wind.temp.solar_nb <- mean(my_aq.test$goodOzone_squaredErrorwind_temp_solar_nb)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_solar_nb_results==0,1])/length(my_aq.test
[,1])*100
```

```
## [1] 97.2973
```

```
#Build a model by using Temp and Wind variable
mq_aq.goz.nb.model.wind.temp <- naiveBayes(goodOzone ~ Wind+Temp, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_wind_temp_nb <- predict(mq_aq.goz.nb.model.wind.temp,my_aq.test[,3:4],type=
"raw")
my_aq.test$goodOzone_wind_temp_nb <- sapply(max.col(my_aq.test$goodOzone_wind_temp_nb,"last"),fl
ag_pred_good_ozone_nb)
my_aq.test$goodOzone_pred_wind_temp_nb_results <- mapply(flag_pred_results,my_aq.test$goodOzone,
my_aq.test$goodOzone_wind_temp_nb)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind_temp_nb <- (my_aq.test$goodOzone_wind_temp_nb-my_aq.test
$goodOzone)^2
my_aq.test$goodOzone_Error_wind_temp_nb <- (my_aq.test$goodOzone_wind_temp_nb-my_aq.test$goodOzo
ne)
mse.goodOzone.wind.temp_nb <- mean(my_aq.test$goodOzone_squaredError_wind_temp_nb)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_temp_nb_results==0,1])/length(my_aq.test[,1])*1
00
```

```
## [1] 94.59459
```

```
#Build a model by using Temp variable
mq_aq.goz.nb.model.temp <- naiveBayes(goodOzone ~ Temp, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_temp_nb <- predict(mq_aq.goz.nb.model.temp,my_aq.test[,4],type="raw")
```

```
## Warning in predict.naiveBayes(mq_aq.goz.nb.model.temp, my_aq.test[, 4], :
## Type mismatch between training and new data for variable 'Temp'. Did you
## use factors with numeric labels for training, and numeric values for new
## data?
```

```
my_aq.test$goodOzone_temp_nb <- sapply(max.col(my_aq.test$goodOzone_temp_nb,"last"),flag_pred_go
od_ozone_nb)
my_aq.test$goodOzone_pred_temp_nb_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_temp_nb)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_temp_nb <- (my_aq.test$goodOzone_temp_nb-my_aq.test$goodOzone)
^2
mse.goodOzone.temp_nb <- mean(my_aq.test$goodOzone_squaredError_temp_nb)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_temp_nb_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 81.08108
```

```
#Build a model by using Wind variable
mq_aq.goz.nb.model.wind <- naiveBayes(goodOzone ~ Wind, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_wind_nb <- predict(mq_aq.goz.nb.model.wind,my_aq.test[,3],type="raw")
```

```
## Warning in predict.naiveBayes(mq_aq.goz.nb.model.wind, my_aq.test[, 3], :
## Type mismatch between training and new data for variable 'Wind'. Did you
## use factors with numeric labels for training, and numeric values for new
## data?
```

```
my_aq.test$goodOzone_wind_nb <- sapply(max.col(my_aq.test$goodOzone_wind_nb,"last"),flag_pred_go
od_ozone_nb)
my_aq.test$goodOzone_pred_wind_nb_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_wind_nb)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_wind_nb <- (my_aq.test$goodOzone_wind_nb-my_aq.test$goodOzone)
^2
mse.goodOzone.wind_nb <- mean(my_aq.test$goodOzone_squaredError_wind_nb)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_wind_nb_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 81.08108
```

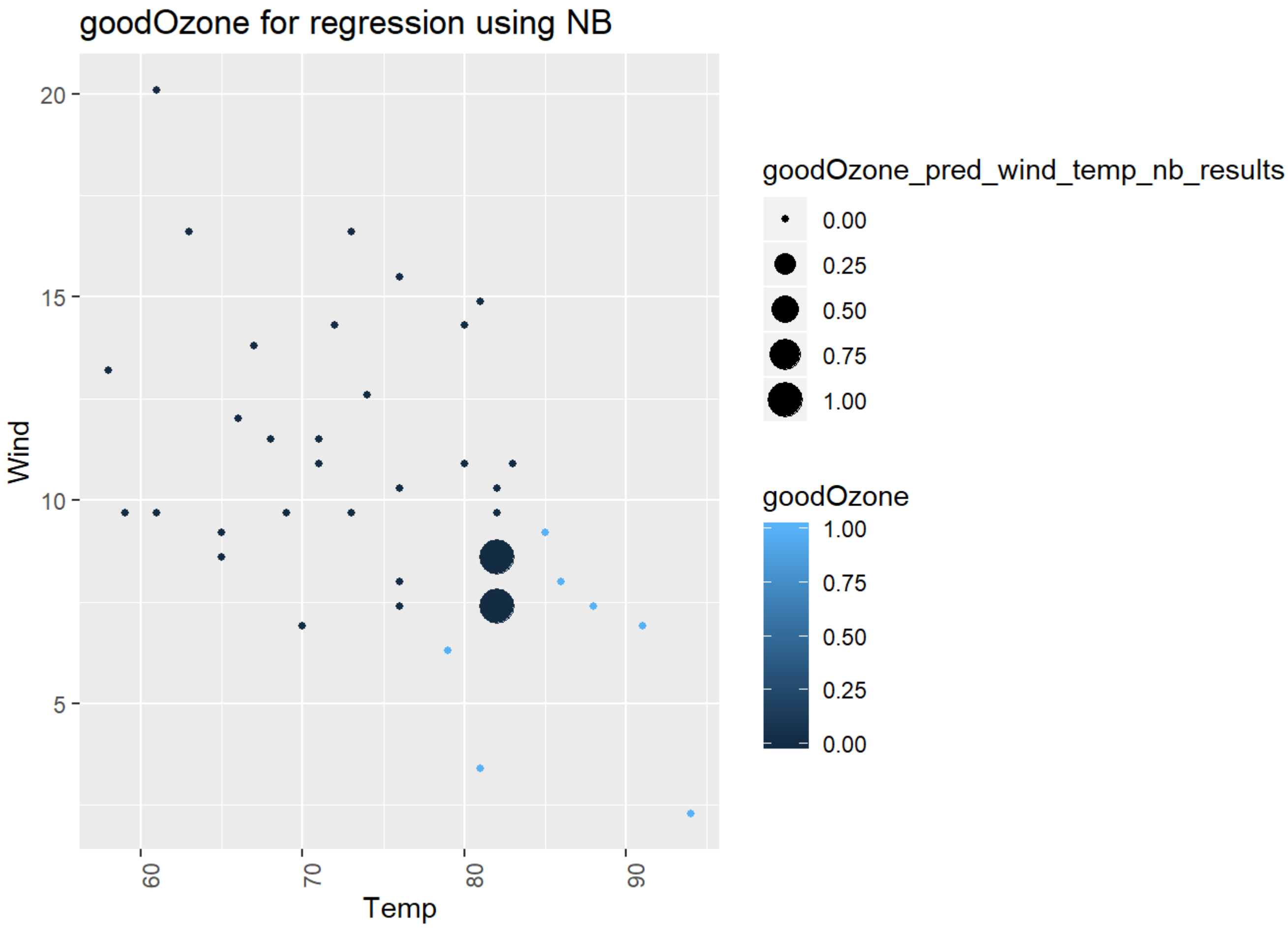
```
#Build a model by using Solar.R variable
mq_aq.goz.nb.model.solar <- naiveBayes(goodOzone ~ Solar.R, data=my_aq.train)

# Test the model using test dataset
my_aq.test$goodOzone_solar_nb <- predict(mq_aq.goz.nb.model.solar,my_aq.test[,2],type="raw")
```

```
## Warning in predict.naiveBayes(mq_aq.goz.nb.model.solar, my_aq.test[, 2], :
## Type mismatch between training and new data for variable 'Solar.R'. Did you
## use factors with numeric labels for training, and numeric values for new
## data?
```

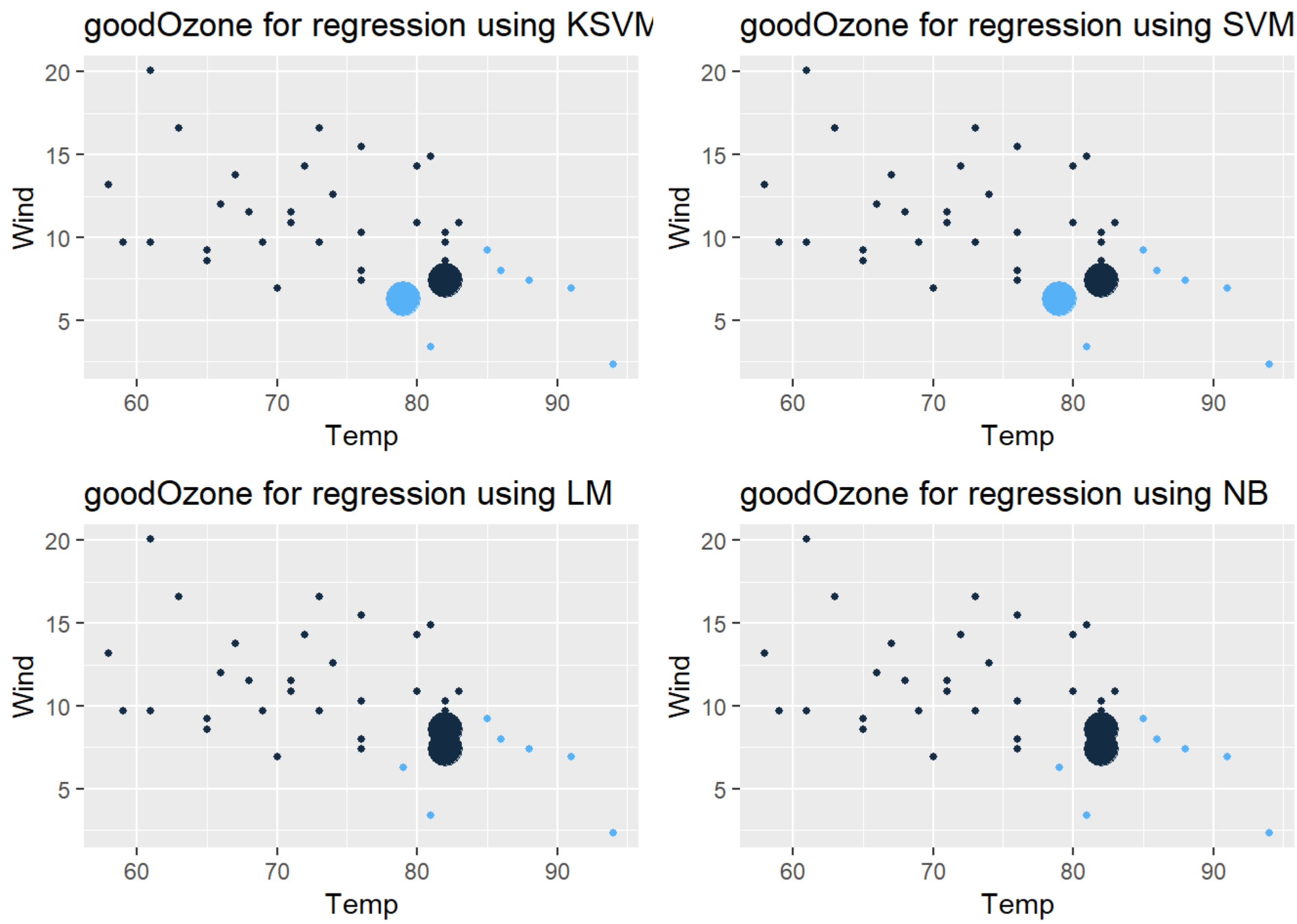
```
my_aq.test$goodOzone_solar_nb <- sapply(max.col(my_aq.test$goodOzone_solar_nb,"last"),flag_pred_
good_ozone_nb)
my_aq.test$goodOzone_pred_solar_nb_results <- mapply(flag_pred_results,my_aq.test$goodOzone,my_a
q.test$goodOzone_solar_nb)

# Square of Error - Support Vector Machine
my_aq.test$goodOzone_squaredError_solar_nb <- (my_aq.test$goodOzone_solar_nb-my_aq.test$goodOzon
e)^2
mse.goodOzone.solar_nb <- mean(my_aq.test$goodOzone_squaredError_solar_nb)

# % success in prediction
length(my_aq.test[my_aq.test$goodOzone_pred_solar_nb_results==0,1])/length(my_aq.test[,1])*100
```

```
## [1] 81.08108
```

```
#scatter   chart (using   ggplot  geom_point),   with   the x-axis  representing   the tem
p,   the
#y-axis representing   the wind,   the size and color  of  each   dot representing   the erro
r

ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_nb_results,color=goodOzone)) + them
e(legend.position = "right",axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using NB")
```




```
ggplot_nb_goodOzone <- ggplot(my_aq.test) +
  geom_point(aes(x=Temp,y=Wind,size=goodOzone_pred_wind_temp_nb_results,color=goodOzone)) +
  labs(x = "Temp", y="Wind",title = "goodOzone for regression using NB") + guides(size=FALSE, co
lor=FALSE)

# Now looking at the difference for all models
grid.arrange( ggplot_ksvm_goodOzone, ggplot_svm_goodOzone, ggplot_lm_goodOzone,ggplot_nb_goodOzo
ne, ncol=2)
```




```
# After comparing all the models to predict good and bad ozone . It appears that KSVM and SVM mo
del is giving better results with predicting good and bad Ozone
```