MA575 Report 3

C2 Team 4: Zhiquan Shen, Katherine Albrecht, Jiachen Chen, Jiahe Zhang, Xinzhi Zhang.

10/22/2021

Introduction

For this analysis, we used a transformed dataset nearly identical to the one we created for Report 2. The total area burned (*totarea*) was calculated by summing over all observations with matching values on day, month, DC, DMC, $FFMC$, and ISI. The weather variables RH, *temperature*, and *wind* were calculated as weighted means based on area burned when the summed area burned is greater than 0. For days where the summed area burned is equal to zero, a simple mean was used. The outcome variable *logarea* was calculated as $logarea = log(totarea + 1)$.

Part 1. Preperation

In Report 2, the only variables that showed a significant correlation with *logarea* were DC, DMC, $FFMC$, and *temperature*. We considered this set of predictors plus wind as our initial candidate covariates. While wind does not show a significant relationship with *logarea*, the observed relationship may be affected by the inverse correlation between temperature and wind. Hence, we believe that higher winds could possibly promote fire spread after controlling temperature.

```
# Load visualization packages
library(carData)
library(car)
library(ggplot2)
library(hrbrthemes)
library(GGally)
library(dplyr)
library(gridExtra)
```

```
# Read data from csv file
firedatafull <- read.csv("forestfires.csv",header=TRUE)
```

```
# create dataset summing area in responses with matching month, day, DC, DMC, FFMC, and ISI values
firedatafull$dateFWIinds <- as.factor(paste(firedatafull$month, firedatafull$day,
firedatafull$DMC, firedatafull$ISI, firedatafull$FFMC, firedatafull$DC))
data2 <- group_by(firedatafull, dateFWIinds)
data3 <- summarise(data2, DC=mean(DC), DMC=mean(DMC), FFMC= mean(FFMC), ISI = mean(ISI),
temp= weighted.mean(temp,area), wind=weighted.mean(wind,area), RH = weighted.mean(RH,area),
totarea = sum(area))
data3 <- data3[data3$totarea > 0,]
data4 <- summarise(data2, DC=mean(DC), DMC=mean(DMC), FFMC= mean(FFMC), ISI = mean(ISI),
temp= mean(temp), wind=mean(wind), RH = mean(RH), totarea = sum(area))
data4 <- data4[data4$totarea == 0,]
data5 <- rbind(data3, data4)
data5$logarea <- log(data5$totarea+1)
data5$month <- as.factor(substr(data5$dateFWIinds,1,3))

attach(data5)
cor.test(logarea,DC)
```

```
##
##  Pearson's product-moment correlation
##
## data:  logarea and DC
## t = 3.6746, df = 234, p-value = 0.0002953
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1091274 0.3508097
## sample estimates:
##       cor
## 0.2335729
```

```
cor.test(logarea,DMC)
```

```
##
##  Pearson's product-moment correlation
##
## data:  logarea and DMC
## t = 3.1284, df = 234, p-value = 0.001981
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.07456782 0.31987649
## sample estimates:
##       cor
## 0.2003604
```

```
cor.test(logarea,FFMC)
```

```
##
##  Pearson's product-moment correlation
##
## data:  logarea and FFMC
## t = 2.8493, df = 234, p-value = 0.004772
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.05674152 0.30371364
## sample estimates:
##       cor
## 0.1831152
```

```
cor.test(logarea,ISI)
```

```
##
##  Pearson's product-moment correlation
##
## data:  logarea and ISI
## t = 0.98108, df = 234, p-value = 0.3276
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.06422199  0.19014974
## sample estimates:
##       cor
## 0.0640034
```

```
cor.test(logarea,temp)
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  logarea and temp
## t = 2.4848, df = 234, p-value = 0.01366
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03331815 0.28225769
## sample estimates:
##       cor
## 0.1603364
```

```
cor.test(logarea,wind)
```

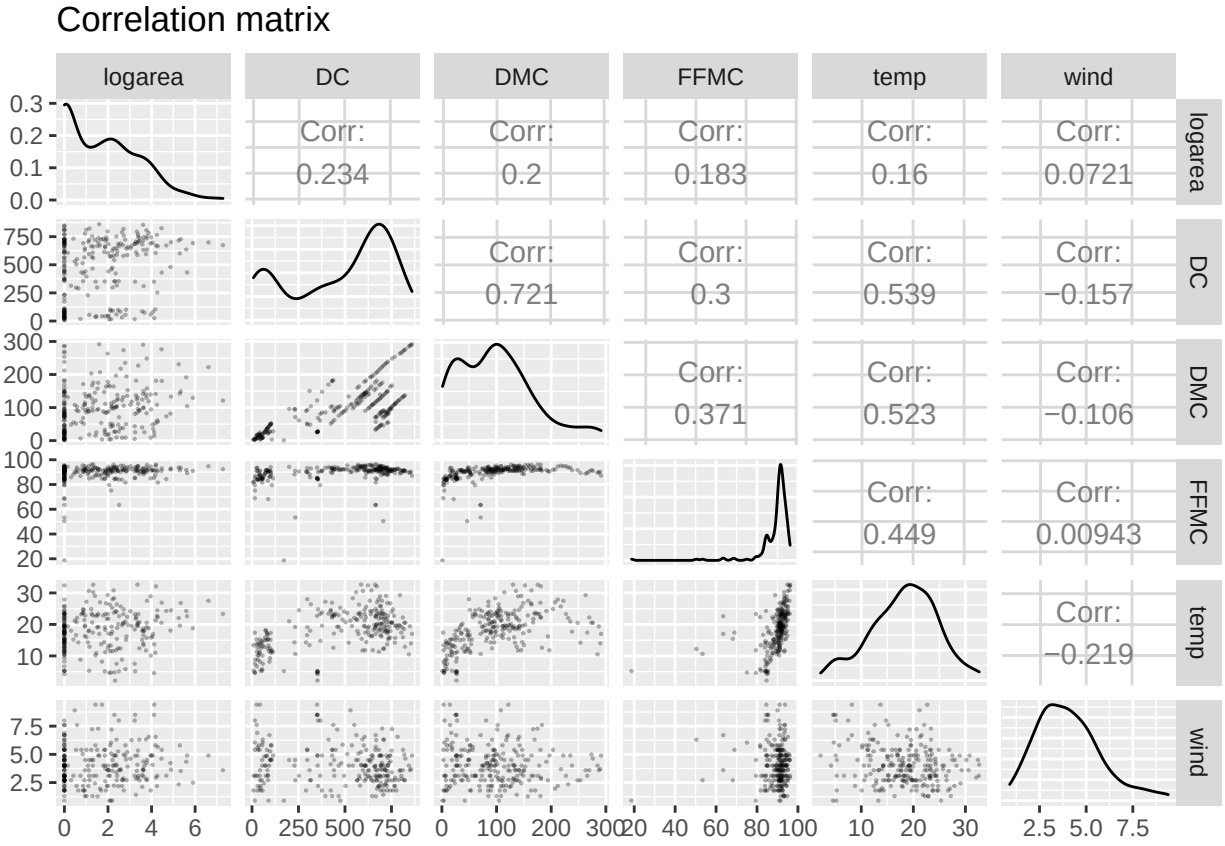
```
##
##  Pearson's product-moment correlation
##
## data:  logarea and wind
## t = 1.106, df = 234, p-value = 0.2699
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.05610509  0.19798930
## sample estimates:
##        cor
## 0.07211205
```

```
cor.test(logarea,RH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  logarea and RH
## t = -1.777, df = 234, p-value = 0.07686
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.23956269  0.01249225
## sample estimates:
##        cor
## -0.1153923
```

Part 2. Correlation plots for the total area burned dataset

```
plotdata <- data.frame(logarea, DC, DMC, FFMC, temp, wind)
ggpairs(plotdata, lower = list(continuous = wrap("points", alpha = 0.3, size=0.1))) +
  ggtitle("Correlation matrix")
```

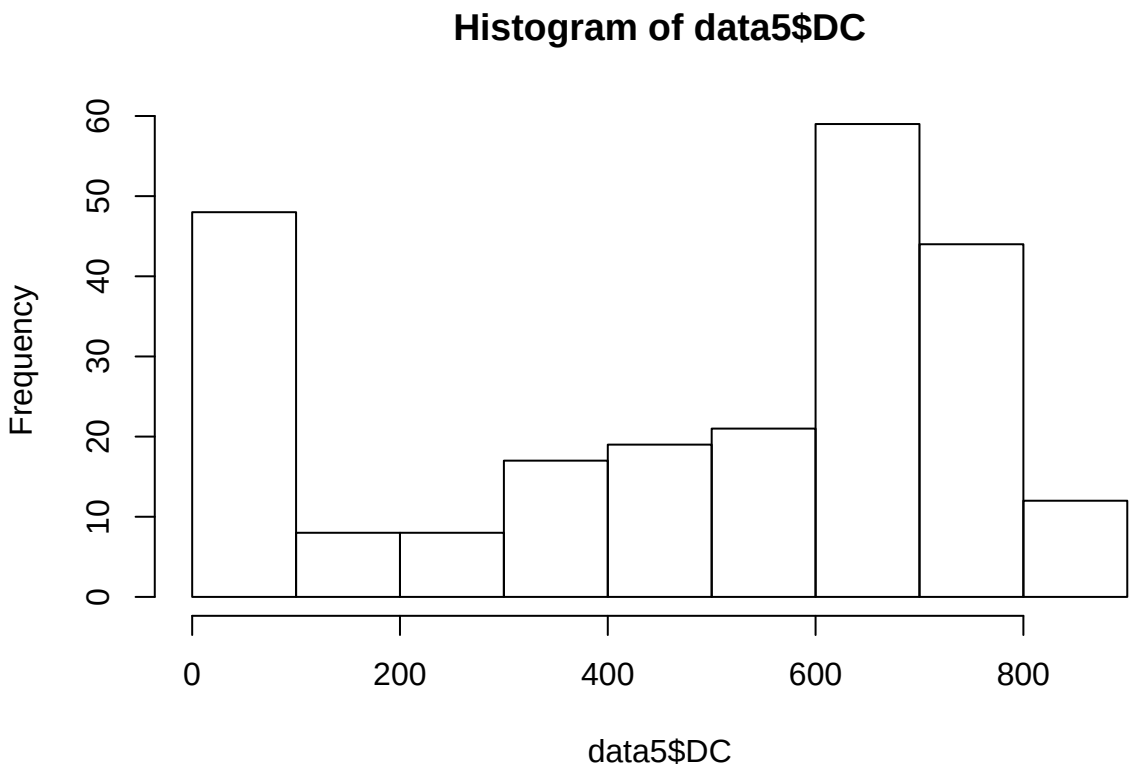



```
detach(data5)
```

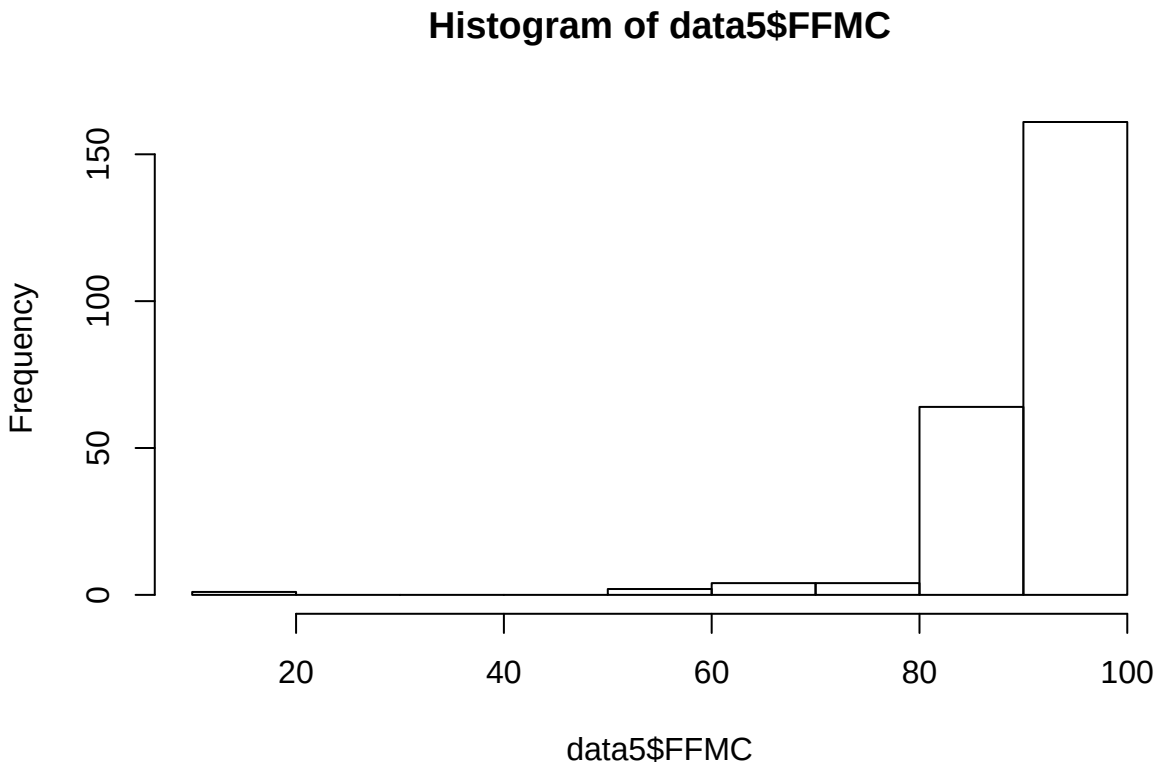
The correlation matrix to the full transformed dataset is shown above.

Since the distribution of DC shows a smaller peak at around 100, we explored the possibility of splitting the dataset into low and high *DC* models. It is possible that the controlled burns are more likely to be started on days with low *DC*. Controlled and uncontrolled fires are likely to have different characteristics, so creating separate models could potentially improve the model accuracy to the data. However, after building separate scatterplot matrices, only one variable was correlated with logarea in each model, which makes it unlikely for any good multiple linear regression models to fit to either data sets.

```
#histogram of DC
hist(data5$DC)
```




```
hist(data5$FFMC)
```




```
lowDC <- data5[which(data5$DC<250|data5$FFMC < 80),]
highDC <- data5[which(data5$DC>250&data5$FFMC>=80),]
print('Summary of total area burned when DC is low (<250)')
```

```
## [1] "Summary of total area burned when DC is low (<250)"
```

```
summary(lowDC$totarea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   7.818   7.465  64.200
```

```
print('Days with low DC by month')
```

```
## [1] "Days with low DC by month"
```

```
summary(lowDC$month)
```

```
## apr aug dec feb jan jul jun mar may nov oct sep
##   7   2   0  17   2   1   4  25   2   1   0   2
```

```
print('Summary of total area burned when DC is high (>= 250)')
```

```
## [1] "Summary of total area burned when DC is high (>= 250)"
```

```
summary(highDC$totarea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.17    5.44   35.55   25.63 1448.61
```

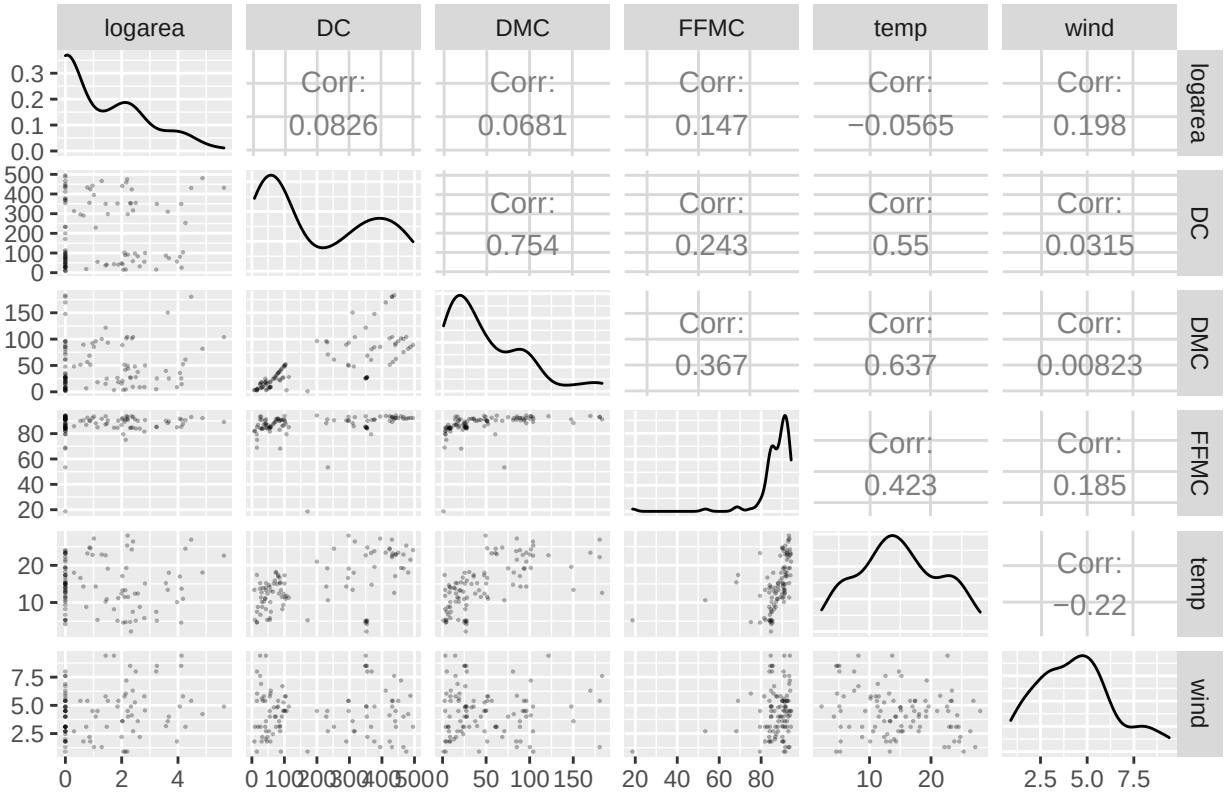
```
print('Days with high DC by month')
```

```
## [1] "Days with high DC by month"
```

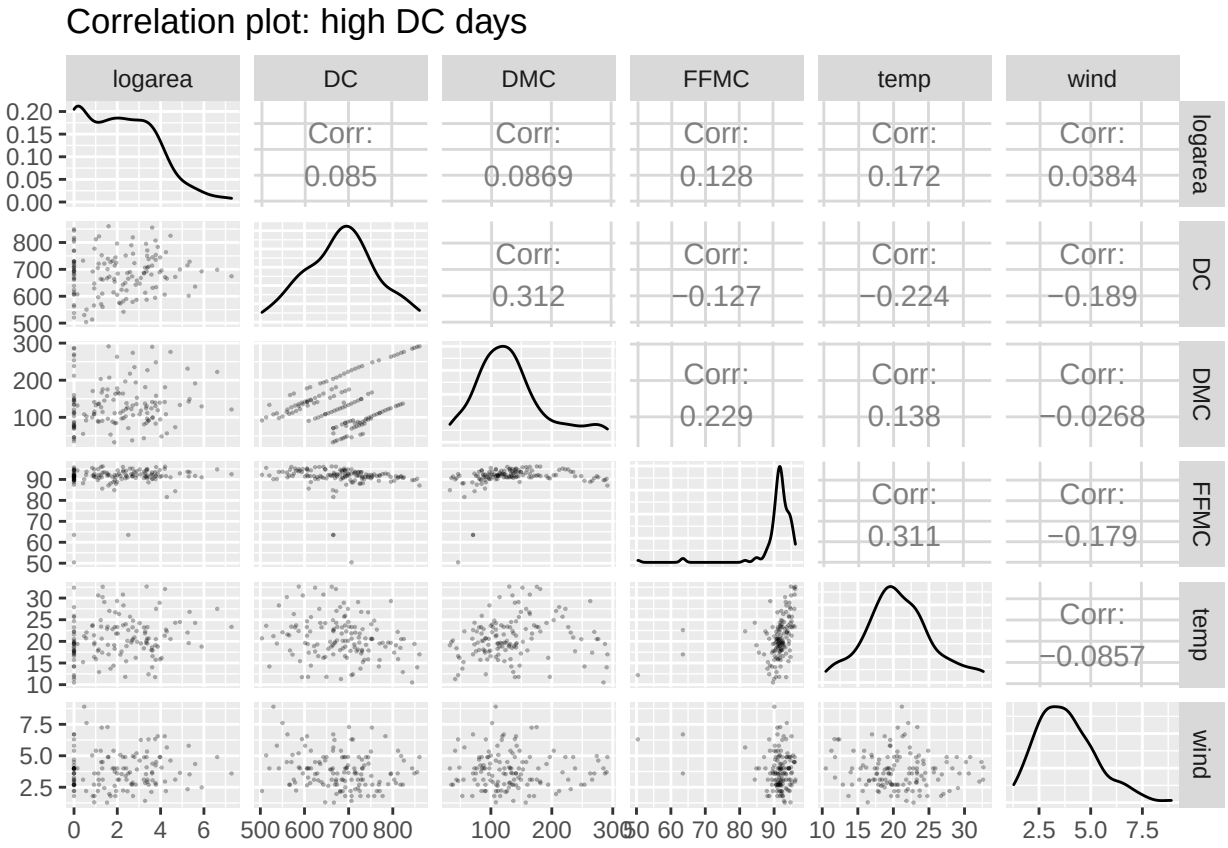
```
summary(highDC$month)
```

```
## apr aug dec feb jan jul jun mar may nov oct sep
##   0  72   6   1   0  25   9   0   0   0   7  53
```

```
#try splitting data by moisture -- controlled burns likely to be started at lower moisture
#content and may behave differently

moist <- data5[which(data5$DC<500),]
dry <- data5[-which(data5$DC<500),]
attach(moist)
moistdata <- data.frame(logarea, DC, DMC, FFMC, temp, wind)
ggpairs(moistdata, lower = list(continuous = wrap("points", alpha = 0.3, size=0.1))) +
  ggtitle("Correlation plot: low DC days")
```




```
detach(moist)

attach(dry)
drydata <- data.frame(logarea, DC, DMC, FFMC, temp, wind)
ggpairs(drydata, lower = list(continuous = wrap("points", alpha = 0.3, size=0.1))) +
   ggtitle("Correlation plot: high DC days")
```




```
detach(dry)
```

Based on the original scatterplot matrix, we are able to see that most of the predictors are highly correlated with one another. This is likely to cause problems when fitting a regression model. The strongest correlation is between DC and DMC (r = 0.721). Therefore, only one of these two predictors should be included in the model. We selected DC, since it shows stronger correlation with logarea based on its correlation coefficient (0.234>0.200).

Hence, the predictors included in our final model are: DC, $FFMC$, *temp* and *wind*. We also incorporate the quadratic term of *temp* in order to fit a more precise trend. The outcome $log(totarea + 1)$ is denoted as *logarea hereafter*.

Part 3. Analyzing the Multiple Linear Regression Model

```
# multiple linear model
attach(data5)
m.mls <- lm(logarea ~ DC + FFMC + temp +I(temp^2) + wind, data=data5)
summary(m.mls)
```

```
##
## Call:
## lm(formula = logarea ~ DC + FFMC + temp + I(temp^2) + wind, data = data5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9035 -1.2010 -0.2657  1.1530  5.2586
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.7493158  1.3360628  -0.561  0.57545
## DC           0.0014157  0.0004648   3.046  0.00259 **
## FFMC         0.0283174  0.0156747   1.807  0.07214 .
## temp        -0.1453420  0.0777557  -1.869  0.06286 .
## I(temp^2)    0.0041491  0.0020446   2.029  0.04357 *
## wind         0.0911154  0.0611470   1.490  0.13757
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.581 on 230 degrees of freedom
## Multiple R-squared:  0.0958, Adjusted R-squared:  0.07614
## F-statistic: 4.874 on 5 and 230 DF,  p-value: 0.0002943
```

```
round(confint(m.mls), digits=4)
```

```
##               2.5 % 97.5 %
## (Intercept) -3.3818 1.8832
## DC           0.0005 0.0023
## FFMC        -0.0026 0.0592
## temp        -0.2985 0.0079
## I(temp^2)    0.0001 0.0082
## wind        -0.0294 0.2116
```

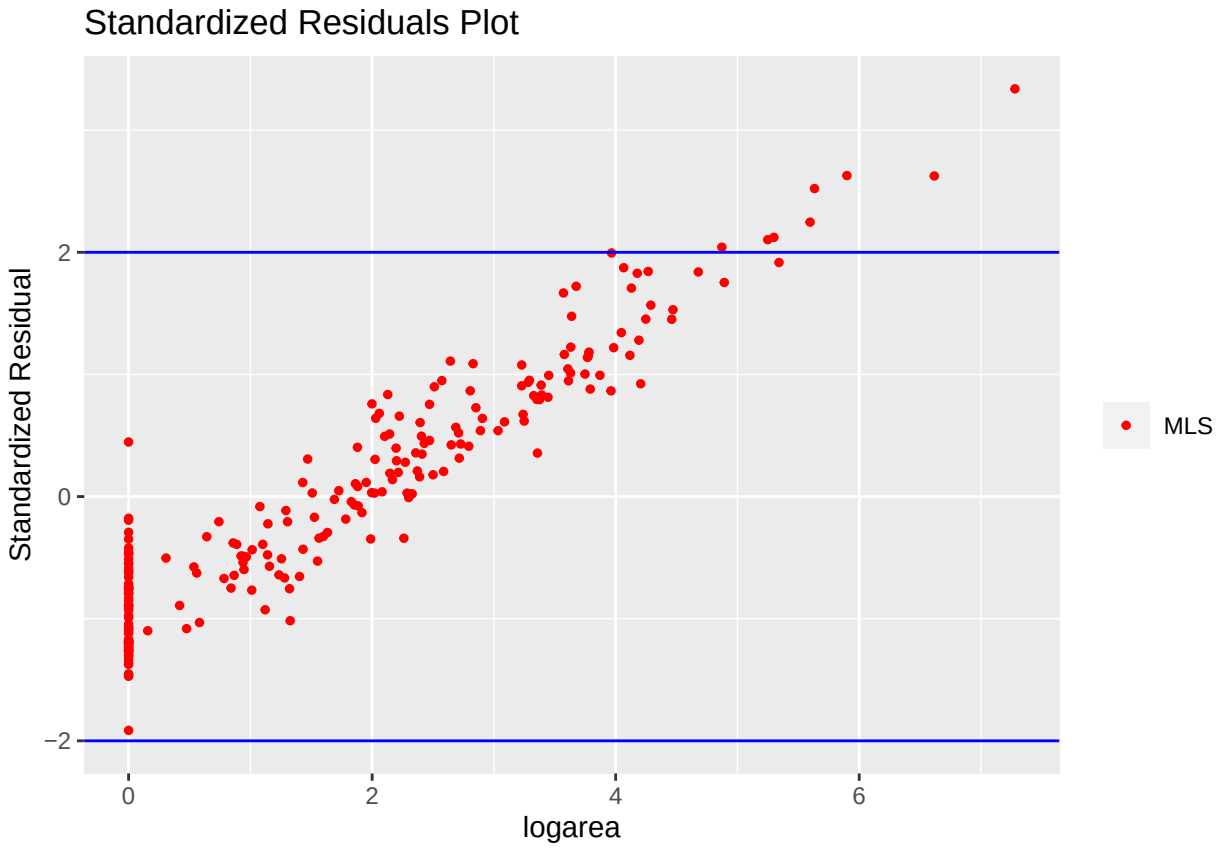
Fitting the multiple linear regression model with the method of least squares (MLS), we noticed that the R-square is 0.0958, which implies that the predictors included in this model could only account for 10% of the variability in *logarea*. Although the R-square is not strong enough, the model can still be regarded as a good fit considering the significance of predictors and the concordance with assumptions of the model (details will be shown).

Under the significance level of 0.05, DC and $I(temp^2)$ are significant predictors for the outcome *logarea*, and the model can be written as: $logarea = \beta_1 * DC + \beta_2 * I(temp^2)$. We notice that the p-value for 1 is 0.00259(<0.05) with a t-statistic on 230 degrees of freedom of 3.046, indicating that if we perform a hypothesis test with $H_0 : \beta_1 = 0$ vs $H_a : \beta_1 \neq 0$, we will reject the null hypothesis and conclude that $1 \neq 0$, at significance level of 0.05. Similarly, the p-value for 2 is 0.04357(<0.05) with a t-statistic on 230 degrees of freedom of 2.029, indicating that if we perform a hypothesis test with $H_0 : \beta_2 = 0$ vs $H_a : \beta_2 \neq 0$, we will reject the null hypothesis and conclude that $\beta_2 \neq 0$ at significance level of 0.05.
The estimated intercept in this model is not significant and thus can be regarded as zero. The coefficient estimate for $DC(\beta_1)$ is 0.0014 (95% $CI : [0.0005, 0.0023]$), and for 1 unit increase in (weighted) DC, we would expect the total area burned in the park to increase by approximately 0.14% (re-calculated based on the increase of the logarithm-transformed total area burned in the park). The coefficient estimate for $I(temp^2)(\beta_2)$ is 0.0041 (95% $CI : [0.0001, 0.0082]$), and for 1 unit increase in (weighted) $I(temp^2)$, we would expect the total area burned in the park to increase by approximately 0.42% (re-calculated based on the increase of the logarithm-transformed total area burned in the park). Although the effect size of significant predictors is small and the linear relationship between the outcome and predictors is weak, we see a linear relationship that $logarea = 0.0014 * DC + 0.0041 * I(temp^2)$.

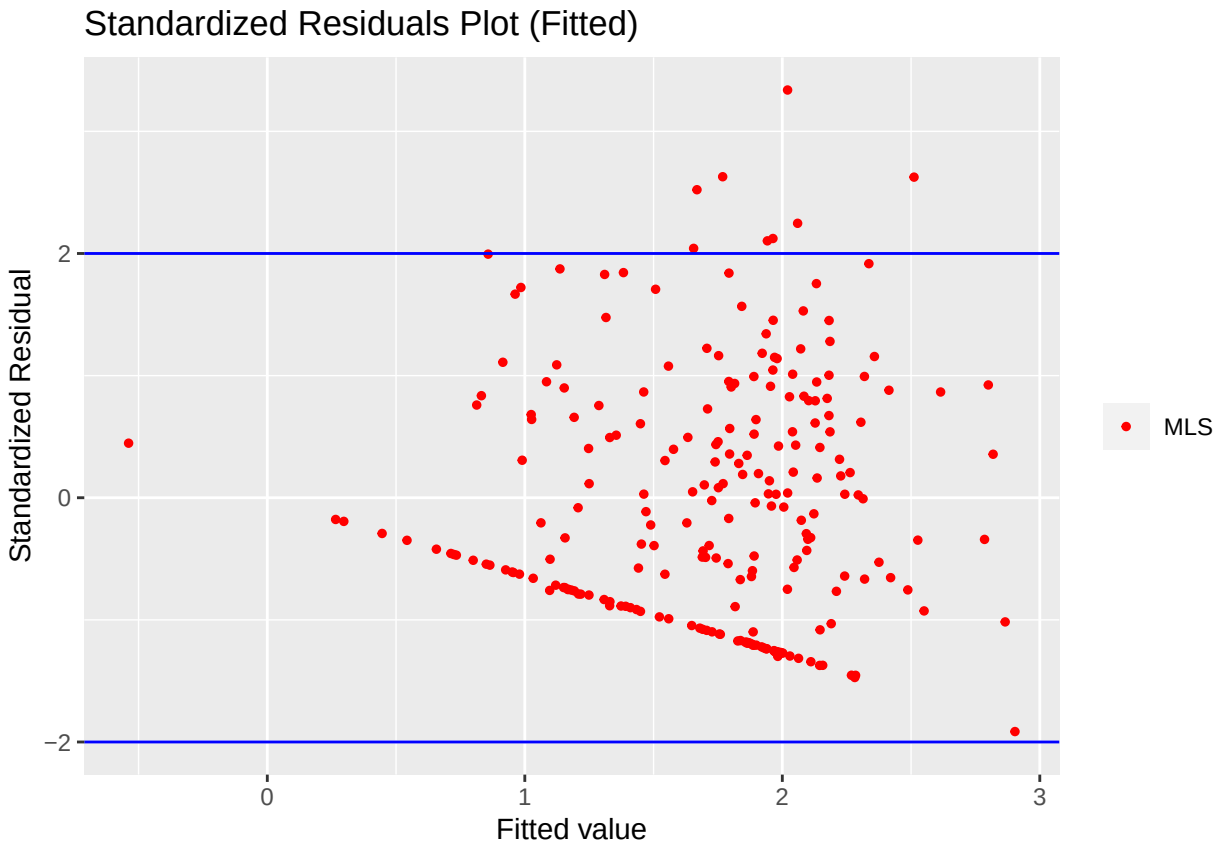
Part 4. Goodness of Fit for the Model

```
# Standard Residuals vs log(totarea+1)
StanResMLS <- rstandard(m.mls)
dataMLS <- data.frame(logarea,StanResMLS)

ggplot() +
  geom_point(data=dataMLS, aes(x=logarea, y=StanResMLS, color = "MLS"), size = 1) +
```

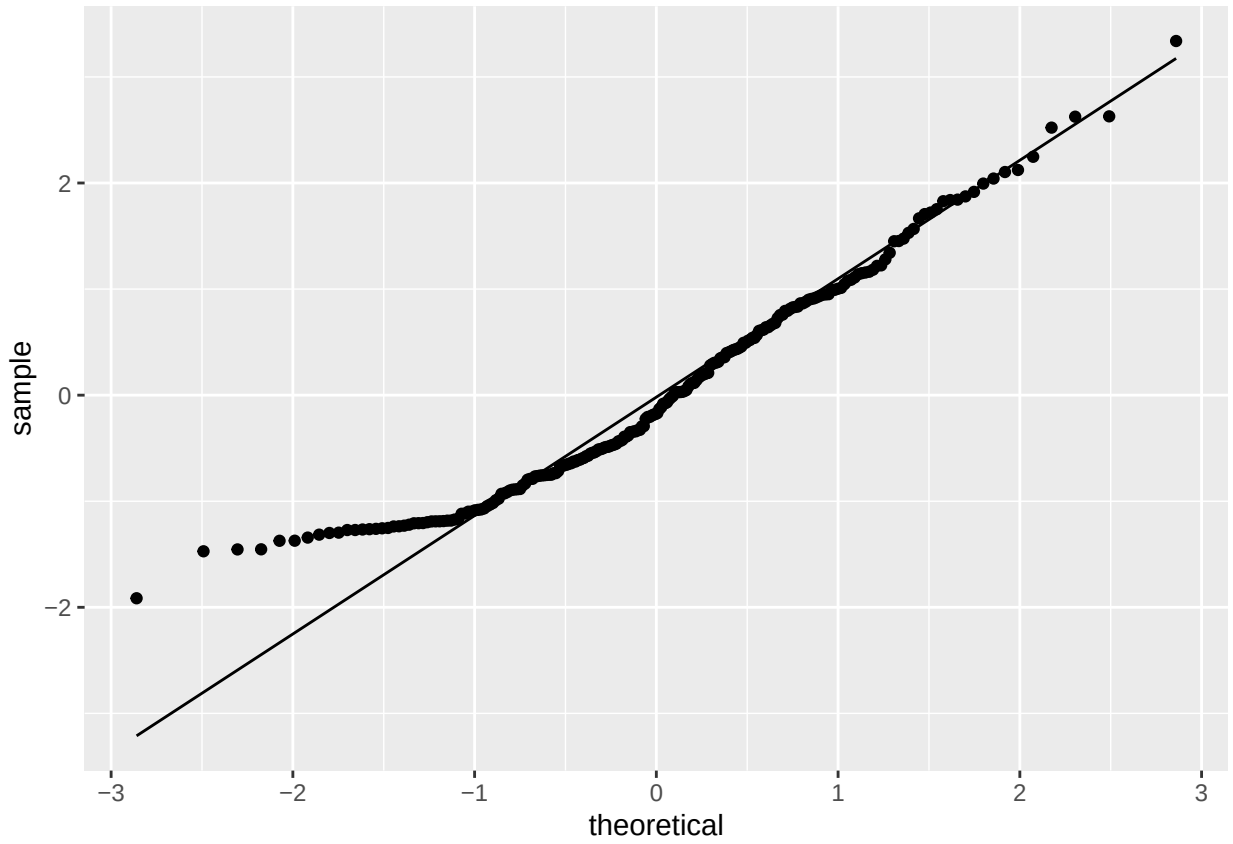
```
  geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
  scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("red")) +
  labs(y = "Standardized Residual") + ggtitle("Standardized Residuals Plot")
```



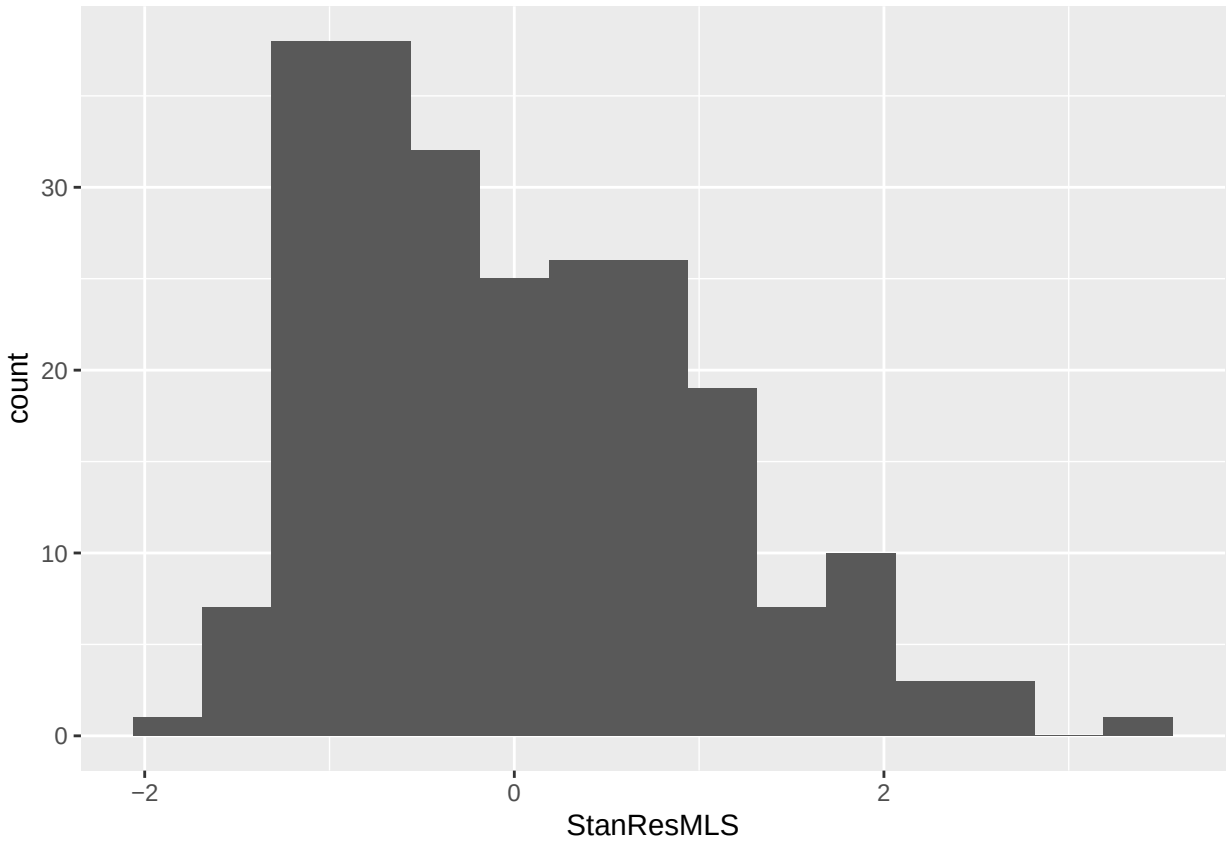

```
# Standarized Residuals vs Fitted
Fitted = fitted(m.mls)
dataMLSFitted <- data.frame(Fitted,StanResMLS)

ggplot() +
  geom_point(data=dataMLSFitted, aes(x=Fitted, y=StanResMLS, color = "MLS"), size = 1) +
  geom_hline(yintercept=2,color='blue') + geom_hline(yintercept=-2, color='blue') +
  scale_color_manual(name = element_blank(), labels = c("MLS"), values = c("red")) +
  labs(y = "Standardized Residual") + labs(x = "Fitted value") +
  ggtitle("Standardized Residuals Plot (Fitted) ")
```




```
# QQ Plot  MLS
p <- ggplot(data.frame(StanResMLS), aes(sample = StanResMLS))
p + stat_qq() + stat_qq_line()
```




```
# Histogram of  MLS
p1 <- ggplot(data = data.frame(StanResMLS), aes(x = StanResMLS)) + geom_histogram(bins = 15)
grid.arrange( p1, ncol=1)
```




```
# Clean up
detach(data5)
```

Based on the first two plots above (standardized residual versus *logarea*, standardized residual versus fitted value), we observe that most of the standardized residuals fall between -2 and 2 with only few outliers are above 2. This suggestes that the residuals are normally distributed. Besides, on the Q-Q plot, the points tend to align to the straight line except those at the tails. Hence, we can conclude that the distribution of residuals is approximately normal. The patterns in the histogram of standardized residuals are consistent with the patterns in the Q-Q plot, thus we can achieve the same conclusion. Obtain from the plot of standardized residual versus fitted value, again, we find that most points fall between -2 and 2; the variance of standardized residuals for each fixed fitted value is not constant yet does not differ too much. Therefore, we conclude the assumption that the variance of the residuals is homogeneous across fitted values is not perfectly satisfied. Considering that the heteroscedastic is not severe, and the effect of reduction in sample size from the transformed data, we think this model is a good fit overall but still has much space for further improvement.