Final Project

2022-12-10

Import data and load packages

```
firedata <- read.csv("forestfires.csv", header=T)

library(ggplot2)
library(ggpubr)
library(hrbrthemes)
library(GGally)
library(gridExtra)
library(car)
library(glmnet)
library(Matrix)
library(ggfortify)
library(reshape2)

theme_update(plot.title = element_text(hjust = 0.5))

X <- as.numeric(firedata$X)
Y <- as.numeric(firedata$Y)
ISI <- as.numeric(firedata$ISI)
temp <- as.numeric(firedata$temp)
RH <- as.numeric(firedata$RH)
wind <- as.numeric(firedata$wind)
rain <- as.numeric(firedata$rain)
area <- as.numeric(firedata$area)
month <- as.numeric(firedata$month)
```

Re-code Variables

```
# Re-code Month Variable into Season Variable (categorical variable)
season <- numeric()
season[firedata$month=="dec" | firedata$month=="jan" | firedata$month=="feb"] <- 0 # winter
season[firedata$month=="mar" | firedata$month=="apr" | firedata$month=="may"] <- 1 # spring
season[firedata$month=="jun" | firedata$month=="jul" | firedata$month=="aug"] <- 2 # summer
season[firedata$month=="sep" | firedata$month=="oct" | firedata$month=="nov"] <- 3 # fall
firedata$season <- season

# Re-code Day Variable
day <- numeric()
day[firedata$day=="mon"] <- 1
day[firedata$day=="tue"] <- 2
day[firedata$day=="wed"] <- 3
day[firedata$day=="thu"] <- 4
day[firedata$day=="fri"] <- 5
day[firedata$day=="sat"] <- 6
day[firedata$day=="sun"] <- 7
firedata$day <- day
```

Randomize rows (before splitting into training and validation)

Use set.seed() function for reproducibility

```
set.seed(1)
sample <- sample(nrow(firedata))
firedata <- firedata[sample,]
```

Form Training and Validation Data Sets (split data 50-50)

```
firedata_train <- firedata[1:259,] # 50% of the data
firedata_valid <- firedata[260:517,] # 50% of the data
```

Assign Variables in Training Data Set

```
X <- firedata_train$X
Y <- firedata_train$Y
month <- firedata_train$month
season <- firedata_train$season
day <- firedata_train$day
temp <- firedata_train$temp
RH <- firedata_train$RH
wind <- firedata_train$wind
rain <- firedata_train$rain
area <- firedata_train$area
ISI <- firedata_train$ISI
```

Summary Statistics

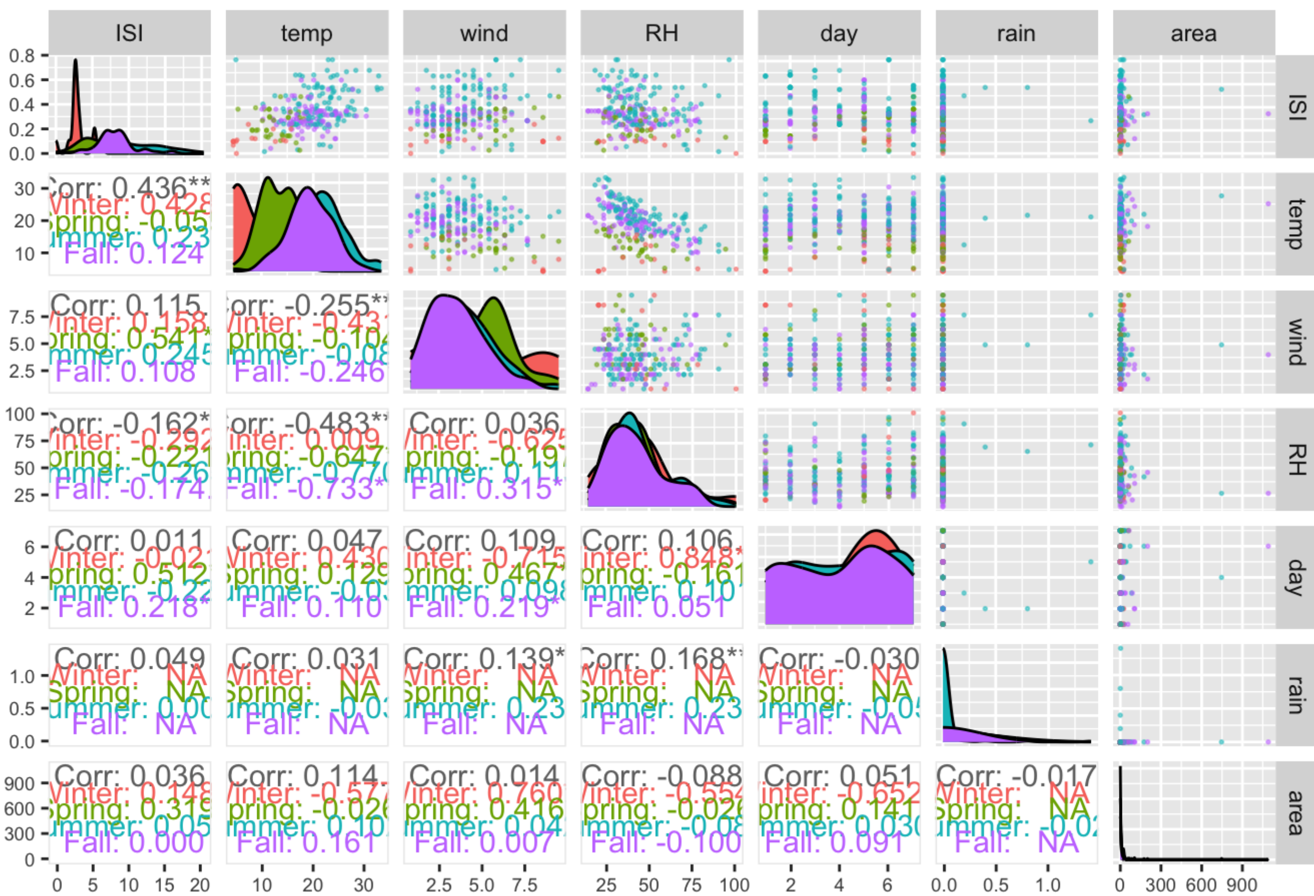
```
summary(firedata_train)
```

```
##        X               Y            month                day      
##  Min.   :1.000   Min.   :2.000   Length:259         Min.   :1.00  
##  1st Qu.:3.000   1st Qu.:4.000   Class :character   1st Qu.:2.00  
##  Median :4.000   Median :4.000   Mode  :character   Median :5.00  
##  Mean   :4.618   Mean   :4.251                      Mean   :4.17  
##  3rd Qu.:6.000   3rd Qu.:5.000                      3rd Qu.:6.00  
##  Max.   :9.000   Max.   :9.000                      Max.   :7.00  
##       FFMC             DMC               DC             ISI       
##  Min.   :18.70   Min.   :  1.10   Min.   :  9.3   Min.   : 0.00  
##  1st Qu.:90.10   1st Qu.: 69.15   1st Qu.:417.6   1st Qu.: 6.30  
##  Median :91.60   Median :108.00   Median :661.8   Median : 8.40  
##  Mean   :90.39   Mean   :110.80   Mean   :538.1   Mean   : 8.69  
##  3rd Qu.:92.65   3rd Qu.:141.25   3rd Qu.:713.9   3rd Qu.:10.55  
##  Max.   :96.20   Max.   :291.30   Max.   :860.6   Max.   :20.30  
##       temp              RH              wind           rain        
##  Min.   : 4.60   Min.   : 15.00   Min.   :0.90   Min.   :0.00000  
##  1st Qu.:15.40   1st Qu.: 32.00   1st Qu.:2.70   1st Qu.:0.00000  
##  Median :19.10   Median : 41.00   Median :3.60   Median :0.00000  
##  Mean   :18.85   Mean   : 43.74   Mean   :3.98   Mean   :0.01081  
##  3rd Qu.:22.90   3rd Qu.: 51.50   3rd Qu.:5.15   3rd Qu.:0.00000  
##  Max.   :33.10   Max.   :100.00   Max.   :9.40   Max.   :1.40000  
##       area              season     
##  Min.   :   0.000   Min.   :0.000  
##  1st Qu.:   0.000   1st Qu.:2.000  
##  Median :   0.000   Median :2.000  
##  Mean   :  14.871   Mean   :2.116  
##  3rd Qu.:   5.815   3rd Qu.:3.000  
##  Max.   :1090.840   Max.   :3.000  
```

Scatterplot and Correlation Matrix

```
data <- data.frame(ISI, temp, wind, RH, day, rain, area)
ggpairs(data, upper = list(continuous = wrap("points", alpha = 0.5, size = 0.3)),
        mapping = ggplot2::aes(color = factor(season, labels = c("Winter", "Spring", "Su
mmer", "Fall"))),
        lower = list(continuous = wrap('cor', size = 4))) +
        theme(axis.text = element_text(size = 7)) +
        labs(title = "Figure 1: Scatterplot and Correlation Matrix")
```

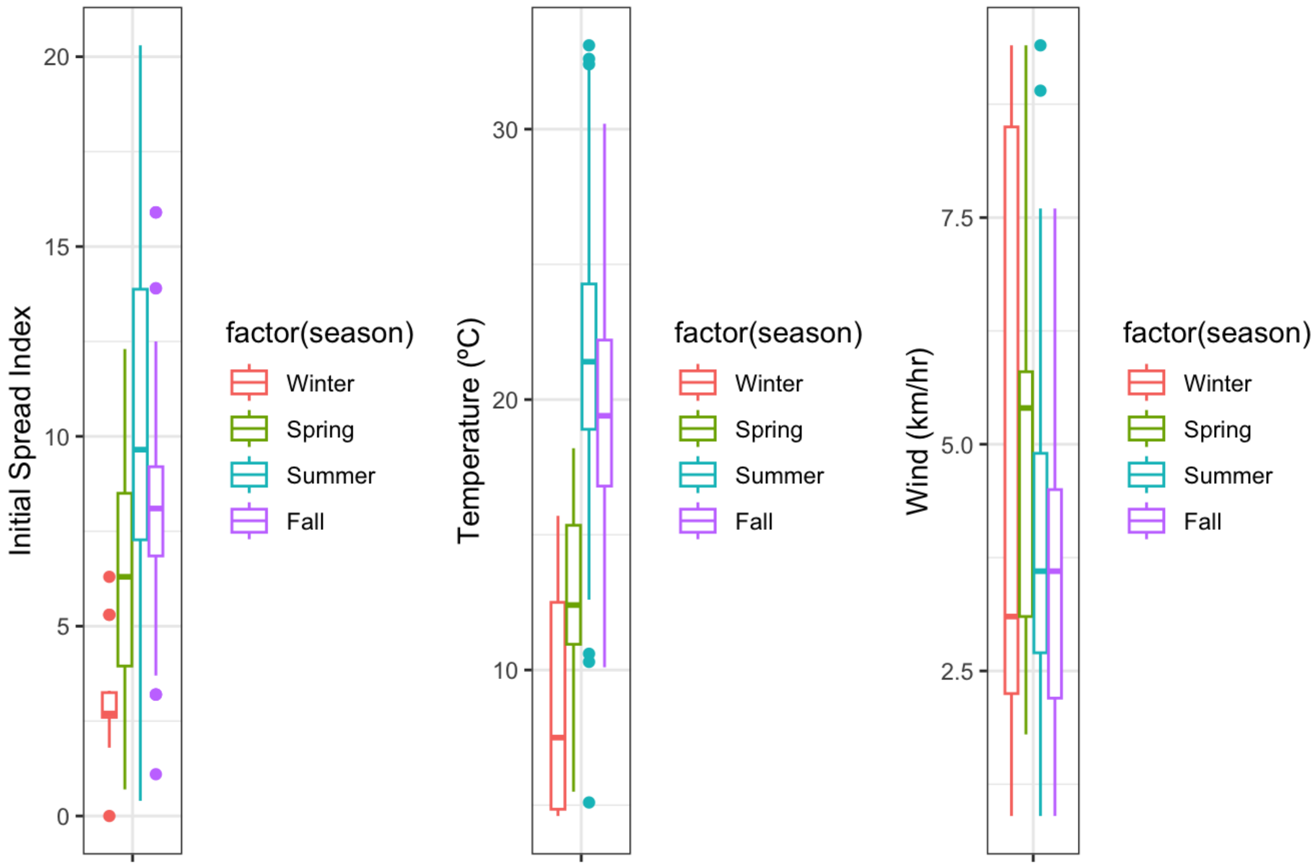
Figure 1: Scatterplot and Correlation Matrix



Boxplots

```
data1 <- data.frame(ISI, season)
data1_melt <- melt(data1, id = "season")
p1 <- ggplot(data1_melt, aes(x = variable, y = value, color = factor(season))) + geom_bo
xplot() + theme_bw() +
      theme(axis.text.x=element_blank()) + labs(x = "", y = "Initial Spread Index") +
      scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

data2 <- data.frame(temp, season)
data2_melt <- melt(data2, id = "season")
p2 <- ggplot(data2_melt, aes(x = variable, y = value, color = factor(season))) + geom_bo
xplot() + theme_bw() +
      theme(axis.text.x=element_blank()) + labs(x = "", y = "Temperature (ºC)") +
      scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

data3 <- data.frame(wind, season)
data3_melt <- melt(data3, id = "season")
p3 <- ggplot(data3_melt, aes(x = variable, y = value, color = factor(season))) + geom_bo
xplot() + theme_bw() +
      theme(axis.text.x=element_blank()) + labs(x = "", y = "Wind (km/hr)") +
      scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

grid.arrange(p1, p2, p3, nrow = 1, top = "Figure 2: Boxplots of Selected Variables by Se
ason")
```

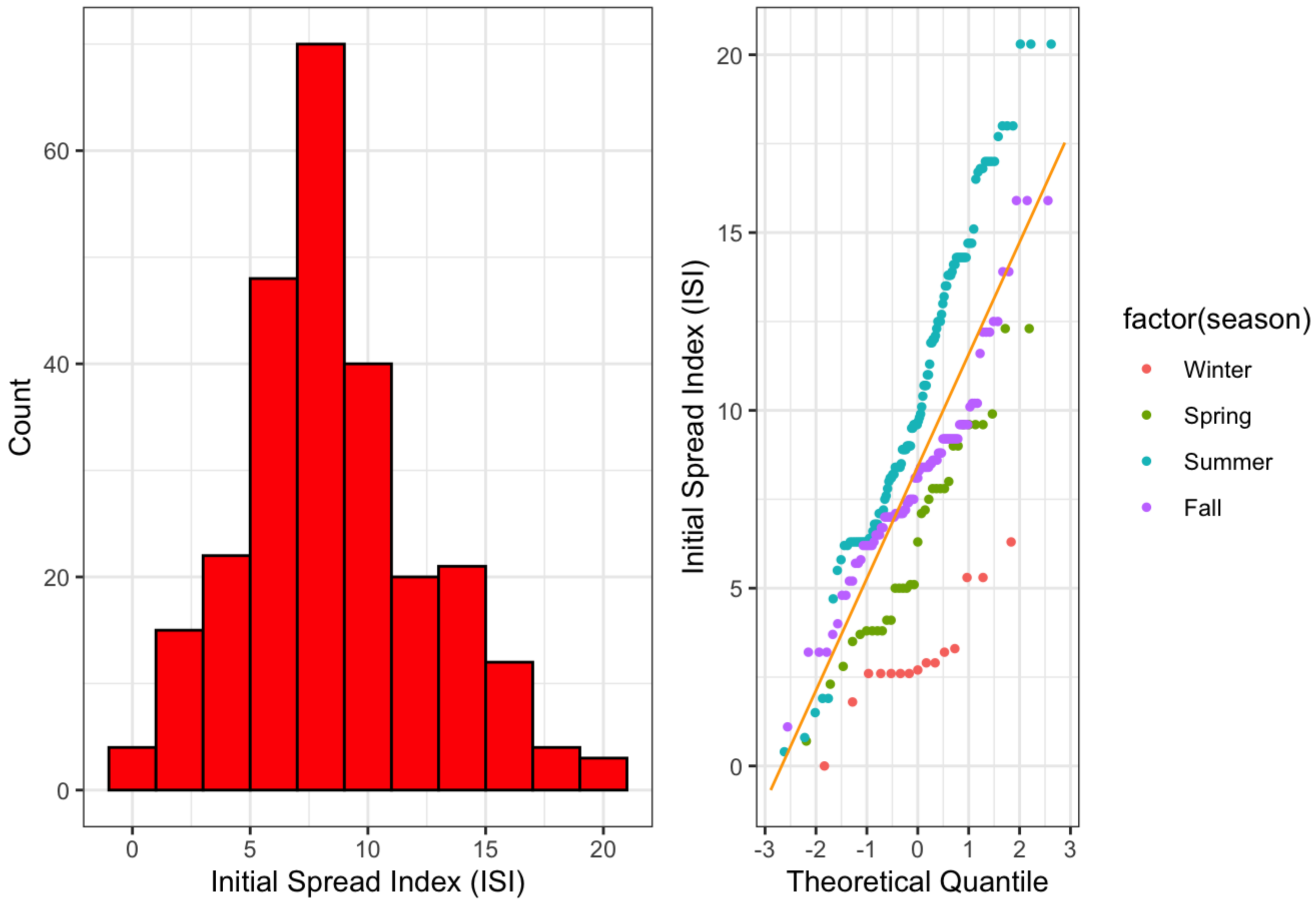
Figure 2: Boxplots of Selected Variables by Season



Combined Plot of Histogram and Q-Q Plot for Response Variable (ISI)

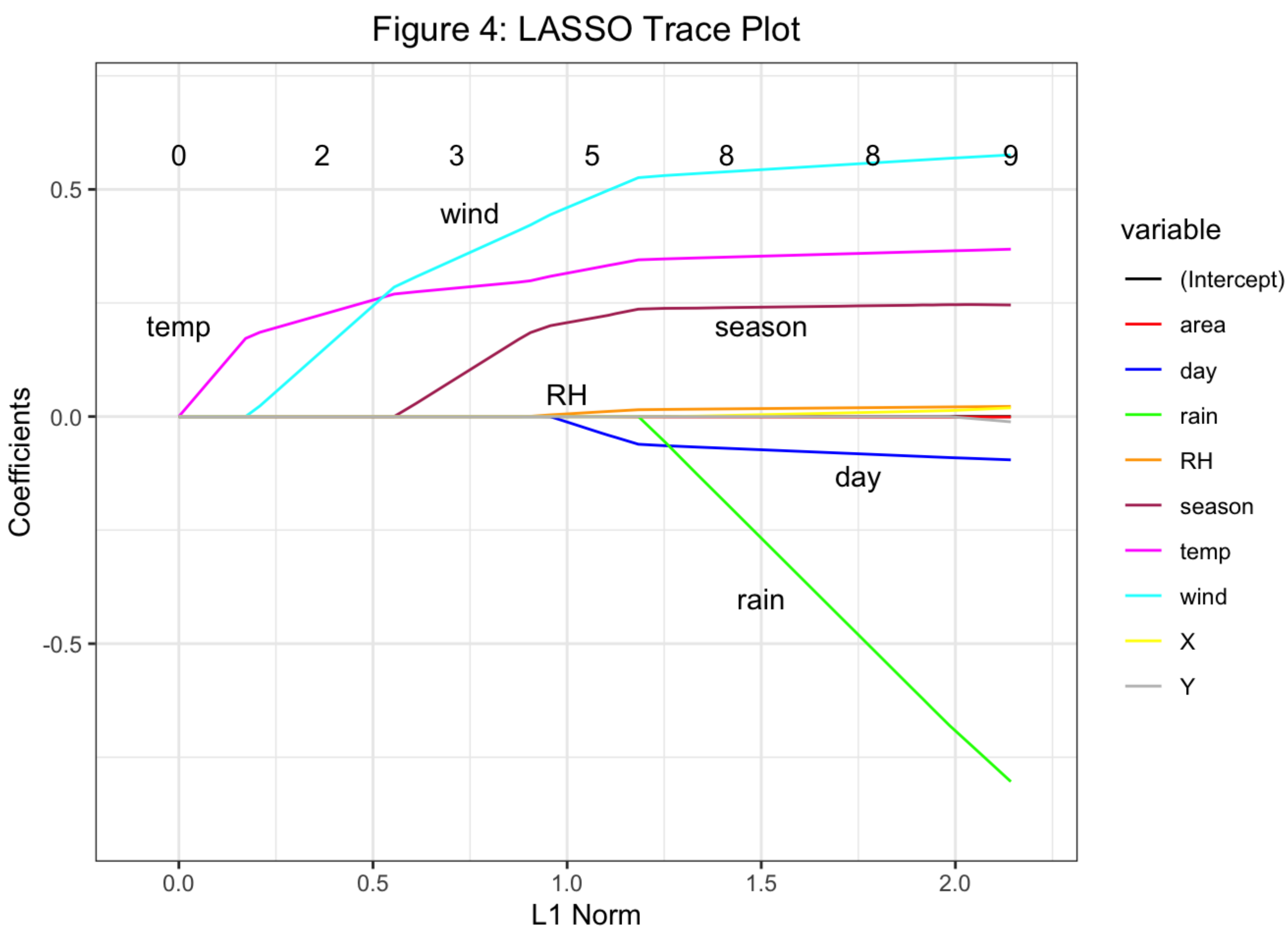
```r
# Histogram of Response Variable (ISI)
ISI_data <- data.frame(ISI)
p1 <- ggplot(ISI_data, aes(x = ISI, color = season)) +
  geom_histogram(binwidth = 2, color = "black", fill = "red") +
  labs(x = "Initial Spread Index (ISI)", y = "Count") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

# Q-Q Plot for Response Variable (ISI)
ISI_data <- data.frame(ISI, factor(season))
p2 <- ggplot(ISI_data, aes(sample = ISI, color = factor(season))) +
  stat_qq(size = 1) +
  geom_qq_line(color = "orange") +
  labs(x = "Theoretical Quantile", y = "Initial Spread Index (ISI)") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

# Combined Plot of Histogram and Q-Q Plot for ISI
grid.arrange(p1, p2, nrow = 1, top = "Figure 3: Histogram and Q-Q Plot of Initial Spread
Index (ISI)")
```

Figure 3: Histogram and Q-Q Plot of Initial Spread Index (ISI)



LASSO

```
x <- model.matrix(ISI~X+Y+season+day+temp+RH+wind+rain+area, firedata_train)
y <- ISI
fit <- glmnet::glmnet(x, y, alpha = 1)

# Plot coefficients vs. lasso penalty
pallete <- c('black', 'red', 'blue', 'green', 'orange','maroon','magenta','cyan', 'yello
w', 'gray')
Lasso <- autoplot(fit, cex = 0.5, xlim = c(-0.1,2.2), ylim = c(-0.9,0.7)) +
        scale_colour_manual(values = pallete) +
        labs(title = "Figure 4: LASSO Trace Plot") +
        theme_bw() +
        theme(plot.title = element_text(hjust = 0.5)) +
        annotate("text", x = 0.0, y = 0.2, label = "temp") +
        annotate("text", x = 0.75, y = 0.45, label = "wind") +
        annotate("text", x = 1.5, y = 0.2, label = "season") +
        annotate("text", x = 1.75, y = -0.13, label = "day") +
        annotate("text", x = 1.5, y = -0.4, label = "rain") +
        annotate("text", x = 1, y = 0.05, label = "RH")
Lasso
```



Ordinary Least Squares Model (Using Training Data)

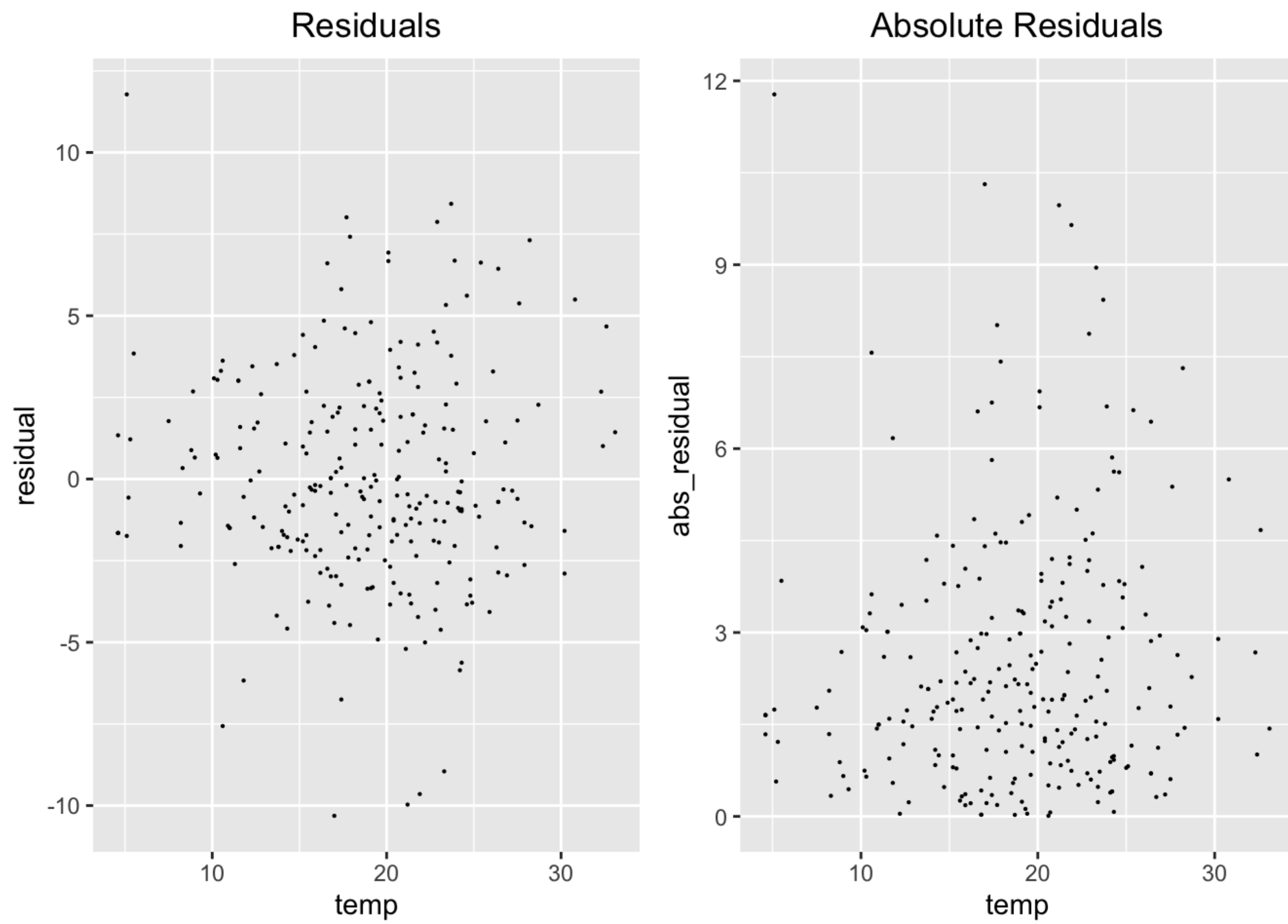
(ISI vs. Temperature, Wind, Season (categorical))

```
model_train <- lm(ISI~temp + wind + factor(season))
summary(model_train)
```

```
##
## Call:
## lm(formula = ISI ~ temp + wind + factor(season))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3138  -1.9106  -0.4068   1.9419  11.7792
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.81674    1.13269  -0.721 0.471540
## temp             0.18017    0.04805   3.749 0.000220 ***
## wind             0.49891    0.11667   4.276 2.69e-05 ***
## factor(season)1  2.43919    1.03958   2.346 0.019732 *
## factor(season)2  5.52504    1.09541   5.044 8.71e-07 ***
## factor(season)3  3.71688    1.05245   3.532 0.000491 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.296 on 253 degrees of freedom
## Multiple R-squared:  0.3332, Adjusted R-squared:   0.32
## F-statistic: 25.28 on 5 and 253 DF,  p-value: < 2.2e-16
```

Construct Weighted Least Squares (WLS) Model

```
# Residuals
residual <- residuals(model_train)

# Absolute value of residual
abs_residual <- abs(residuals(model_train))

data <- data.frame(temp, residual, abs_residual)

p11 <- ggplot(data, aes(x = temp, y = residual)) + geom_point(size = 0.1) +
  ggtitle("Residuals")

p12 <- ggplot(data, aes(x = temp, y = abs_residual)) + geom_point(size = 0.1) +
  ggtitle("Absolute Residuals")

grid.arrange(p11, p12, ncol = 2)
```



Weighted Least Squares

```
# Calculate fitted values from a regression of absolute residuals vs predictors
wts <- 1 / fitted(lm(abs(residuals(model_train))~temp + wind + factor(season)))^2

# Fit a WLS model using weights = 1 / (fitted values)^2
wls_train <- lm(ISI~temp + wind + factor(season), weights = wts)
summary(wls_train)
```

```
##
## Call:
## lm(formula = ISI ~ temp + wind + factor(season), weights = wts)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2008 -0.8448 -0.2301  0.9023  3.9199
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.48549    0.62121   0.782 0.435218
## temp             0.17613    0.04141   4.253 2.97e-05 ***
## wind             0.34494    0.09921   3.477 0.000597 ***
## factor(season)1  1.61480    0.60547   2.667 0.008146 **
## factor(season)2  4.85487    0.73355   6.618 2.16e-10 ***
## factor(season)3  3.04910    0.60483   5.041 8.82e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.295 on 253 degrees of freedom
## Multiple R-squared:  0.5031, Adjusted R-squared:  0.4932
## F-statistic: 51.22 on 5 and 253 DF,  p-value: < 2.2e-16
```

Weighted Least Squares Diagnostics

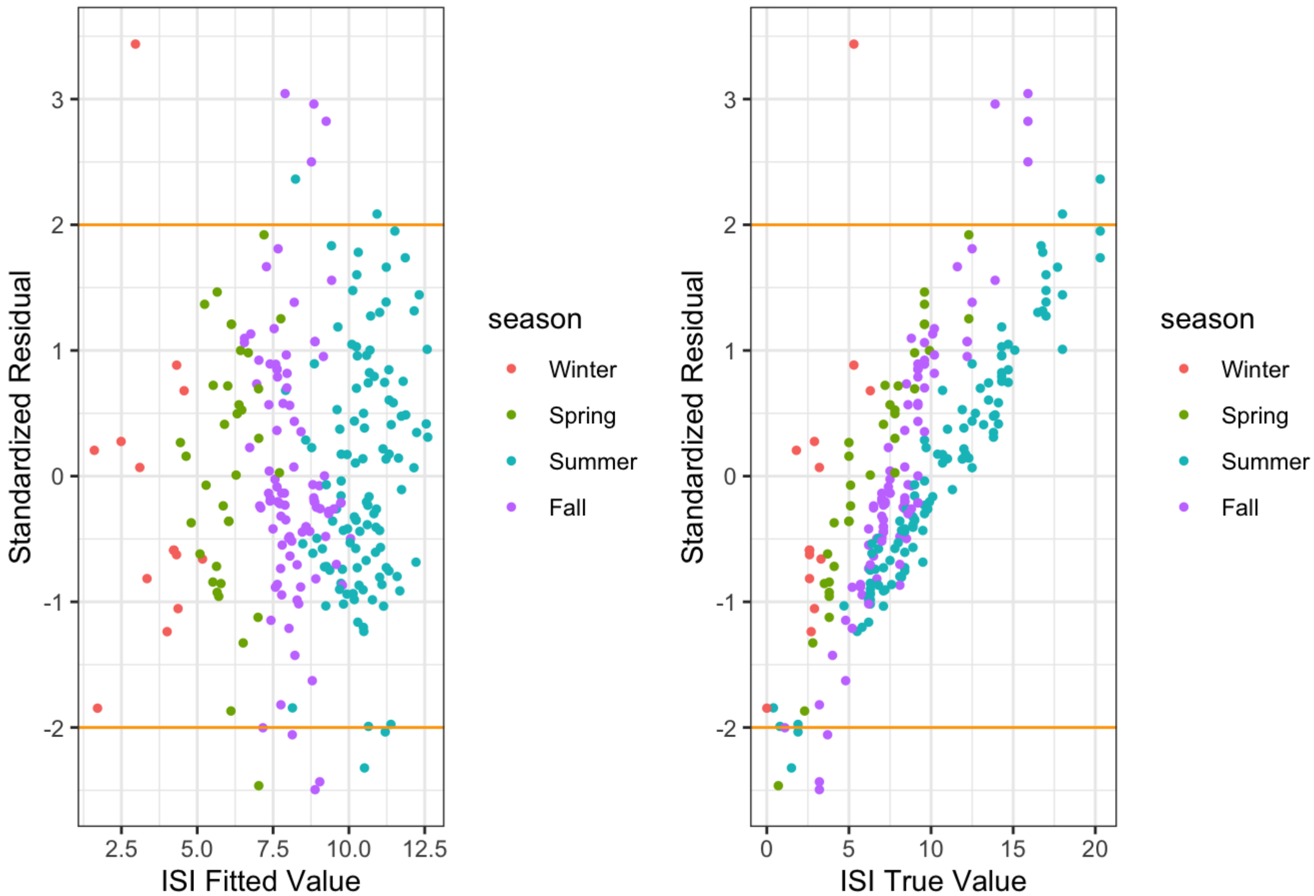
```
stand_resid_train <- rstandard(wls_train)
fitted_train <- fitted(wls_train)
leverages_train <- hatvalues(wls_train)
student_resid_train <- rstudent(wls_train)
season <- factor(season)
index_train <- seq(1, 259, 1)

data_diagnostics_train <- data.frame(stand_resid_train, fitted_train, leverages_train, s
tudent_resid_train, ISI, season, index_train)
```

Weighted Least Squares Combined Standardized Residuals

Plots

```
# Standardized Residuals vs. ISI Fitted Values Plot
p13 <- ggplot() + geom_point(data = data_diagnostics_train, aes(x = fitted_train, y = st
and_resid_train, col = season), size = 1) +
  geom_hline(yintercept = 2, color = "orange") + geom_hline(yintercept = -2, color = "or
ange") +
  labs(x = "ISI Fitted Value", y = "Standardized Residual") +
  scale_y_continuous(breaks = seq(-3, 3, 1)) +
  theme_bw() +
  scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall")) +
  theme(plot.title = element_text(hjust = 0.5))

# Standardized Residuals vs. ISI True Values Plot
p14 <- ggplot() + geom_point(data = data_diagnostics_train, aes(x = ISI, y = stand_resid
_train, col = season), size = 1) +
  geom_hline(yintercept = 2, color = "orange") + geom_hline(yintercept = -2, color = "or
ange") +
  labs(x = "ISI True Value", y = "Standardized Residual") +
  scale_y_continuous(breaks = seq(-3, 3, 1)) +
  theme_bw() +
  scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall")) +
  theme(plot.title = element_text(hjust = 0.5))

# Combined Standardized Residuals Plots
grid.arrange(p13, p14, nrow = 1, top = "Figure 5: Weighted Least Squares Standardized Re
siduals Plots")
```

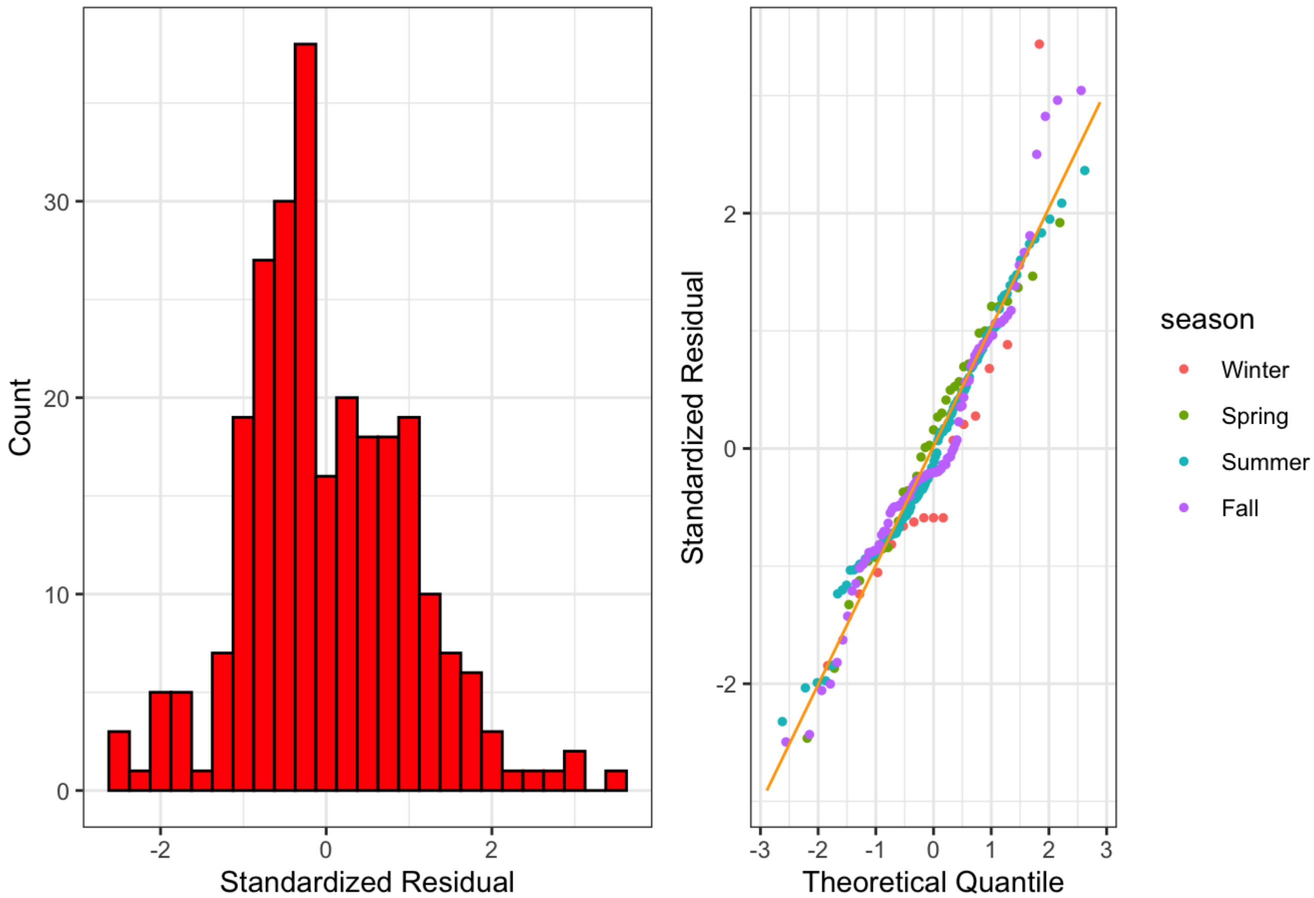
Figure 5: Weighted Least Squares Standardized Residuals Plots



Weighted Least Squares Combined Histogram and Q-Q Plots for Standardized Residuals

```
# Histogram of Standardized Residuals
p15 <- ggplot(data_diagnostics_train, aes(x = stand_resid_train)) +
  geom_histogram(binwidth = 0.25, color = "black", fill = "red") +
  labs(x = "Standardized Residual", y = "Count") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

# Q-Q Plot
p16 <- ggplot(data_diagnostics_train, aes(sample = stand_resid_train, color = season)) +
  stat_qq(size = 1) +
  geom_qq_line(color = "orange") +
  labs(x = "Theoretical Quantile", y = "Standardized Residual") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

# Combined Histogram and Q-Q Plots for Standardized Residuals
grid.arrange(p15, p16, nrow = 1, top = "Figure 6: Weighted Least Squares Histogram and Q
-Q Plot of Standardized Residuals")
```

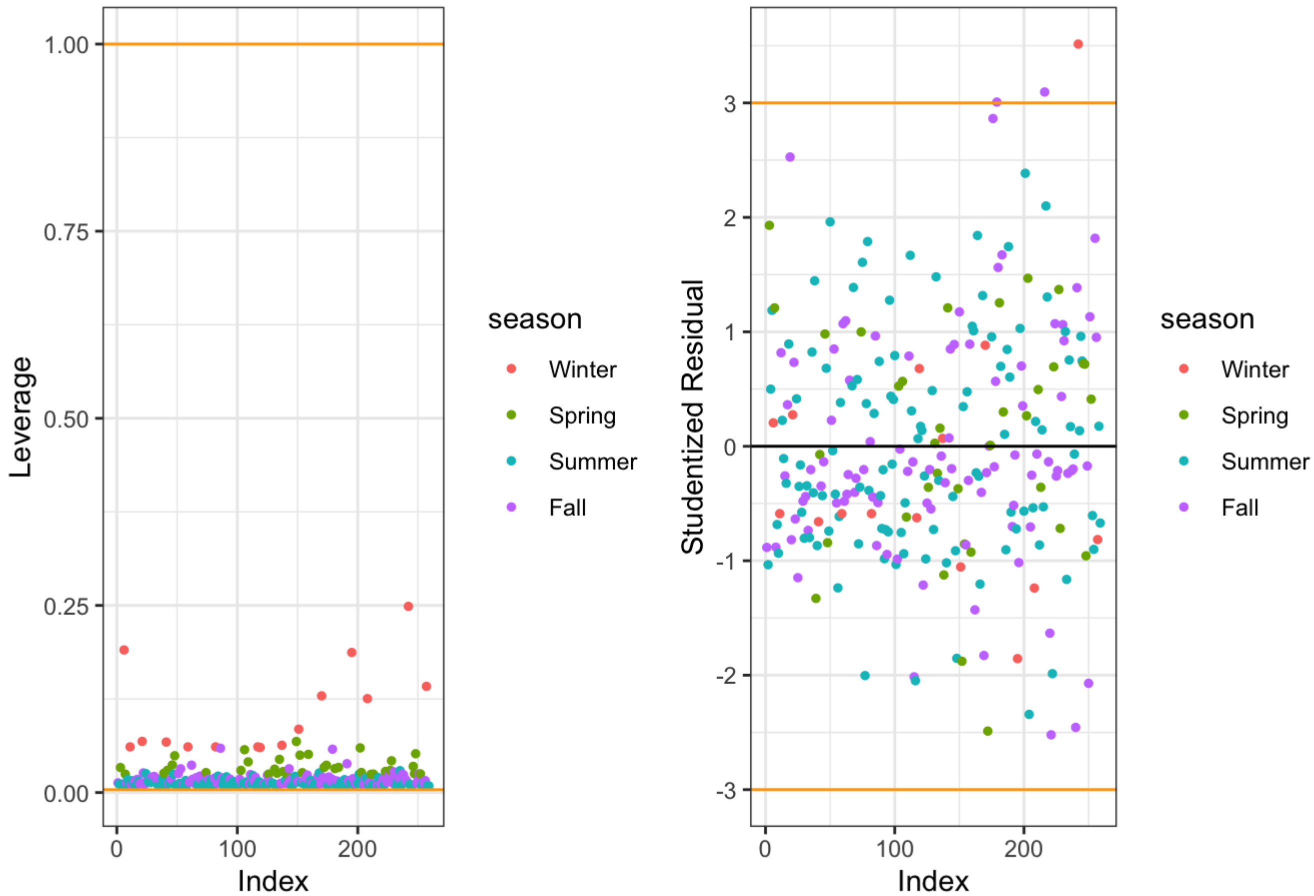
Figure 6: Weighted Least Squares Histogram and Q-Q Plot of Standardized Residuals



Weighted Least Squares Combined Outlier Detection Plots

```
# Plot of Leverages (to detect outliers in the x-space)
p17 <- ggplot() + geom_point(data = data_diagnostics_train, aes(x = index_train, y = lev
erages_train, color = season), size = 1) +
  geom_hline(yintercept = 1 / length(index_train), color = "orange") + geom_hline(yinter
cept = 1, color = "orange") +
  labs(x = "Index", y = "Leverage") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

# Plot of Studentized Residuals (to detect outliers in the y-space)
p18 <- ggplot() + geom_point(data =  data_diagnostics_train, aes(x = index_train, y = st
udent_resid_train, color = season), size = 1) +
  geom_hline(yintercept = -3, color = "orange") + geom_hline(yintercept = 3, color = "or
ange") +
  geom_hline(yintercept = 0, color = "black") +
  scale_y_continuous(breaks = seq(-3, 3, 1)) +
  labs(x = "Index", y = "Studentized Residual") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

# Combined Outlier Detection Plots
grid.arrange(p17, p18, nrow = 1, top = "Figure 7: Weighted Least Squares Leverage and St
udentized Residuals Plots")
```

Figure 7: Weighted Least Squares Leverage and Studentized Residuals Plots



Remove Outliers from Training Data

```
ISI_rev <- ISI[-c(which(student_resid_train > 3 | student_resid_train < -3 | leverages_t
rain > 0.20))]
temp_rev <- temp[-c(which(student_resid_train > 3 | student_resid_train < -3 | leverages
_train > 0.20))]
season_rev <- season[-c(which(student_resid_train > 3 | student_resid_train < -3 | lever
ages_train > 0.20))]
wind_rev <- wind[-c(which(student_resid_train > 3 | student_resid_train < -3 | leverages
_train > 0.20))]

data_rev <- data.frame(ISI_rev, temp_rev, season_rev, wind_rev)
```

Ordinary Least Squares (After Removing Outliers)

```
model_train_rev <- lm(ISI_rev~temp_rev + wind_rev + factor(season_rev))
summary(model_train_rev)
```

```
##
## Call:
## lm(formula = ISI_rev ~ temp_rev + wind_rev + factor(season_rev))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4305  -1.8896  -0.2194   1.9993  11.5992
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.15175    1.15311  -0.999 0.318845
## temp_rev             0.17259    0.04769   3.619 0.000358 ***
## wind_rev             0.52872    0.11627   4.548 8.47e-06 ***
## factor(season_rev)1  2.72435    1.05085   2.593 0.010089 *
## factor(season_rev)2  5.90582    1.11346   5.304 2.49e-07 ***
## factor(season_rev)3  3.94860    1.07194   3.684 0.000282 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.25 on 250 degrees of freedom
## Multiple R-squared:  0.3452, Adjusted R-squared:  0.3321
## F-statistic: 26.36 on 5 and 250 DF,  p-value: < 2.2e-16
```

Weighted Least Squares (After Removing Outliers)

```
# Calculate fitted values from a regression of absolute residuals vs predictors
wts_rev <- 1 / fitted(lm(abs(residuals(model_train_rev))~temp_rev + wind_rev + factor(se
ason_rev)))^2

# Fit a WLS model using weights = 1 / (fitted values)^2
wls_train_rev <- lm(ISI_rev~temp_rev + wind_rev + factor(season_rev), weights = wts_rev)
summary(wls_train_rev)
```

```
##
## Call:
## lm(formula = ISI_rev ~ temp_rev + wind_rev + factor(season_rev),
##     weights = wts_rev)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3921 -0.8349 -0.1332  0.9386  4.2414
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.02478    0.52694  -0.047  0.96252
## temp_rev             0.14573    0.03674   3.967 9.52e-05 ***
## wind_rev             0.43562    0.09313   4.678 4.75e-06 ***
## factor(season_rev)1  2.09493    0.54417   3.850  0.00015 ***
## factor(season_rev)2  5.71852    0.65353   8.750 3.22e-16 ***
## factor(season_rev)3  3.69531    0.52024   7.103 1.27e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.278 on 250 degrees of freedom
## Multiple R-squared:  0.6068, Adjusted R-squared:  0.5989
## F-statistic: 77.15 on 5 and 250 DF,  p-value: < 2.2e-16
```

Weighted Least Squares Validation

```
# Residuals for training data
resid_train_rev <- resid(wls_train_rev)

# Prediction for validation data
data <- data.frame(temp_rev = firedata_valid$temp, wind_rev = firedata_valid$wind, seaso
n_rev = factor(firedata_valid$season))
predict_valid <- predict(wls_train_rev, se.fit = TRUE, newdata = data)
resid_valid <- firedata_valid$ISI - predict_valid$fit
```

Mean Square Error for training data

```
mean((resid_train_rev)^2) # MSE for training data is 10.37
```

```
## [1] 10.37022
```

```
mean((resid_valid)^2) # MSE for validation data is 20.33
```

```
## [1] 20.33095
```

Relative Mean Square Error (can multiply by 100 to convert to a %)

```
mean((resid_train_rev)^2) / mean((ISI_rev)^2) # Relative MSE for training data is approx
imately 0.1144 (11.44%)
```

```
## [1] 0.1143969
```

```
mean((resid_valid)^2) / mean((firedata_valid$ISI)^2) # Relative MSE for validation data
is approximately 0.1801 (18.01%)
```

```
## [1] 0.1800988
```

Remove outlier from validation data and repeat previous analysis

Look for an outlier in the validation dataset

```
summary(resid_valid)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -9.02113 -1.50526  0.01292  0.64127  2.25668 45.38571
```

```
head(sort(resid_valid, decreasing = T), n = 10)
```

```
##        89       185        91       125        55       165        73       120
## 45.385705 14.374467 12.436678 10.314259  9.578777  9.514876  8.815458  8.714074
##       230        22
##  8.437971  8.350027
```

```
tail(sort(resid_valid, decreasing = T), n = 10)
```

```
##        42       127       102        56       248       132        82       213
## -4.992446 -5.196388 -5.238697 -5.593002 -5.685932 -6.331777 -7.732187 -7.792788
##       169       218
## -9.021130 -9.021130
```

Remove the largest residual (nearly 3x the second largest residual in magnitude)

```
predict <- predict_valid$fit
predict_valid_rev <- predict[-c(which(resid_valid > 15))]
resid_valid_rev <- resid_valid[-c(which(resid_valid > 15))]
ISI_valid_rev <- firedata_valid$ISI[-c(which(resid_valid > 15))]
season_valid_rev <- firedata_valid$season[-c(which(resid_valid > 15))]
```

Repeat the previous analysis (mean square error/relative mean square error) with this outlier removed

Mean Square Error for validation data

```
mean((resid_valid_rev)^2) # MSE for validation data (without outlier) is 12.40
```

```
## [1] 12.39503
```

Relative Mean Square Error for validation data (can multiply by 100 to convert to a %)

```
mean((resid_valid_rev)^2) / mean((ISI_valid_rev)^2) # Relative MSE for validation data
(without outlier) is approximately 0.1226 (12.26%)
```

```
## [1] 0.1226246
```

Create data frame with validation observations and predicted values

```
test <- data.frame(ISI_valid_rev, factor(season_valid_rev), predict_valid_rev, 1:length
(predict_valid_rev));
colnames(test)[1] = "ISI"
colnames(test)[2] = "Season"
colnames(test)[3] = "Prediction"
colnames(test)[4] = "Index"
```

Combined Validation Plots

```
# Plot Initial Spread Index vs Prediction for Validation Data Set
p19 <- ggplot(data = test, aes(x = ISI, y = Prediction, color = Season)) + geom_point()
+
  geom_abline(intercept = 0, slope = 1, color = "orange") +
  labs(x = "ISI True Value", y = "ISI Predicted Value") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_discrete(labels = c("Winter", "Spring", "Summer", "Fall"))

# Further Comparisons of Predicted Values vs. True Values for Validation Data
p20 <- ggplot(data = test, aes(x = Index)) +
  geom_line(aes(y = ISI, color = "ISI")) +
  geom_line(aes(y = Prediction, color = "Prediction"), linetype = "twodash") +
  scale_color_manual(name = element_blank(), labels = c("True ISI","Predicted ISI"),
                     values = c("pink", "steelblue")) + labs(y = "") +
  labs(x = "Index", y = "ISI") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

# Combined Validation Plots
grid.arrange(p19, p20, ncol = 1, top = "Figure 8: Weighted Least Squares Validation and
Prediction")
```