# Climate Analysis Appendix

Max Vo

## Data Loading and Preliminary Analysis

```
data = na.omit(read.csv("/cloud/project/climate_change.csv"))

# For Consistency in Year Based/Month Based Analyses, we remove the first 8 months
# of the dataset (as the data randomly begins from the 5th month of 1983)

# Therefore, our dataset starts from 1984
climate <- data[-(1:8),]

#brief summary; head()/taiL()
head(climate,10)
```

```
##    Year Month    MEI    CO2     CH4     N2O  CFC.11  CFC.12      TSI Aerosols
## 9  1984     1 -0.339 344.05 1658.98 304.130 197.219 363.359 1365.426   0.0451
## 10 1984     2 -0.565 344.77 1656.48 304.194 197.759 364.296 1365.662   0.0416
## 11 1984     3  0.131 345.46 1655.77 304.285 198.249 365.044 1366.170   0.0383
## 12 1984     4  0.331 346.77 1657.68 304.389 198.723 365.692 1365.566   0.0352
## 13 1984     5  0.121 347.55 1649.33 304.489 199.233 366.317 1365.778   0.0324
## 14 1984     6 -0.142 346.98 1634.13 304.593 199.858 367.029 1366.096   0.0302
## 15 1984     7 -0.138 345.55 1629.89 304.722 200.671 367.893 1366.114   0.0282
## 16 1984     8 -0.179 343.20 1643.67 304.871 201.710 368.843 1365.978   0.0260
## 17 1984     9 -0.082 341.35 1663.60 305.021 202.972 369.800 1365.867   0.0239
## 18 1984    10  0.016 341.68 1674.65 305.158 204.407 370.782 1365.787   0.0220
##      Temp
## 9   0.089
## 10  0.013
## 11  0.049
## 12 -0.019
## 13  0.065
## 14 -0.016
## 15 -0.024
## 16  0.034
## 17  0.025
## 18 -0.035
```

```
tail(climate,10)
```

```
##     Year Month    MEI    CO2     CH4     N2O  CFC.11  CFC.12      TSI Aerosols
## 299 2008     3 -1.635 385.97 1792.84 321.295 245.430 535.979 1365.673   0.0034
## 300 2008     4 -0.942 387.16 1792.57 321.354 245.086 535.648 1365.715   0.0033
## 301 2008     5 -0.355 388.50 1796.43 321.420 244.914 535.399 1365.717   0.0031
## 302 2008     6  0.128 387.88 1791.80 321.447 244.676 535.128 1365.673   0.0031
## 303 2008     7  0.003 386.42 1782.93 321.372 244.434 535.026 1365.672   0.0033
```

```
## 304 2008     8 -0.266 384.15 1779.88 321.405 244.200 535.072 1365.657   0.0036
## 305 2008     9 -0.643 383.09 1795.08 321.529 244.083 535.048 1365.665   0.0043
## 306 2008    10 -0.780 382.99 1814.18 321.796 244.080 534.927 1365.676   0.0046
## 307 2008    11 -0.621 384.13 1812.37 322.013 244.225 534.906 1365.707   0.0048
## 308 2008    12 -0.666 385.56 1812.88 322.182 244.204 535.005 1365.693   0.0046
##      Temp
## 299 0.447
## 300 0.278
## 301 0.283
## 302 0.315
## 303 0.406
## 304 0.407
## 305 0.378
## 306 0.440
## 307 0.394
## 308 0.330
```

```r
summary(climate)
```

```
##       Year          Month             MEI               CO2
##  Min.   :1984   Min.   : 1.00   Min.   :-1.6350   Min.   :341.4
##  1st Qu.:1990   1st Qu.: 3.75   1st Qu.:-0.4125   1st Qu.:353.8
##  Median :1996   Median : 6.50   Median : 0.2250   Median :362.3
##  Mean   :1996   Mean   : 6.50   Mean   : 0.2573   Mean   :363.8
##  3rd Qu.:2002   3rd Qu.: 9.25   3rd Qu.: 0.8197   3rd Qu.:373.8
##  Max.   :2008   Max.   :12.00   Max.   : 3.0010   Max.   :388.5
##       CH4            N2O            CFC.11          CFC.12           TSI
##  Min.   :1630   Min.   :304.1   Min.   :197.2   Min.   :363.4   Min.   :1365
##  1st Qu.:1726   1st Qu.:308.6   1st Qu.:247.5   1st Qu.:478.0   1st Qu.:1366
##  Median :1766   Median :311.7   Median :259.0   Median :528.9   Median :1366
##  Mean   :1753   Mean   :312.6   Mean   :253.5   Mean   :501.3   Mean   :1366
##  3rd Qu.:1787   3rd Qu.:317.0   3rd Qu.:267.3   3rd Qu.:540.7   3rd Qu.:1366
##  Max.   :1814   Max.   :322.2   Max.   :271.5   Max.   :543.8   Max.   :1367
##     Aerosols             Temp
##  Min.   :0.00160   Min.   :-0.2820
##  1st Qu.:0.00280   1st Qu.: 0.1288
##  Median :0.00550   Median : 0.2510
##  Mean   :0.01535   Mean   : 0.2600
##  3rd Qu.:0.01200   3rd Qu.: 0.4100
##  Max.   :0.14940   Max.   : 0.7390
```

```r
# structure of dataframe
str(climate)
```

```
## 'data.frame':    300 obs. of  11 variables:
##  $ Year    : int  1984 1984 1984 1984 1984 1984 1984 1984 1984 1984 ...
##  $ Month   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MEI     : num  -0.339 -0.565 0.131 0.331 0.121 -0.142 -0.138 -0.179 -0.082 0.016 ...
##  $ CO2     : num  344 345 345 347 348 ...
##  $ CH4     : num  1659 1656 1656 1658 1649 ...
##  $ N2O     : num  304 304 304 304 304 ...
##  $ CFC.11  : num  197 198 198 199 199 ...
##  $ CFC.12  : num  363 364 365 366 366 ...
##  $ TSI     : num  1365 1366 1366 1366 1366 ...
##  $ Aerosols: num  0.0451 0.0416 0.0383 0.0352 0.0324 0.0302 0.0282 0.026 0.0239 0.022 ...
##  $ Temp    : num  0.089 0.013 0.049 -0.019 0.065 -0.016 -0.024 0.034 0.025 -0.035 ...
```

```r
# checking for any duplicate data (there is none)
nrow(climate)==nrow(unique(climate))
```

```
## [1] TRUE
```

## Exploratory Analysis

**FIGURE 2: Testing Correlation Between Variables**

```r
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(dplyr)
```

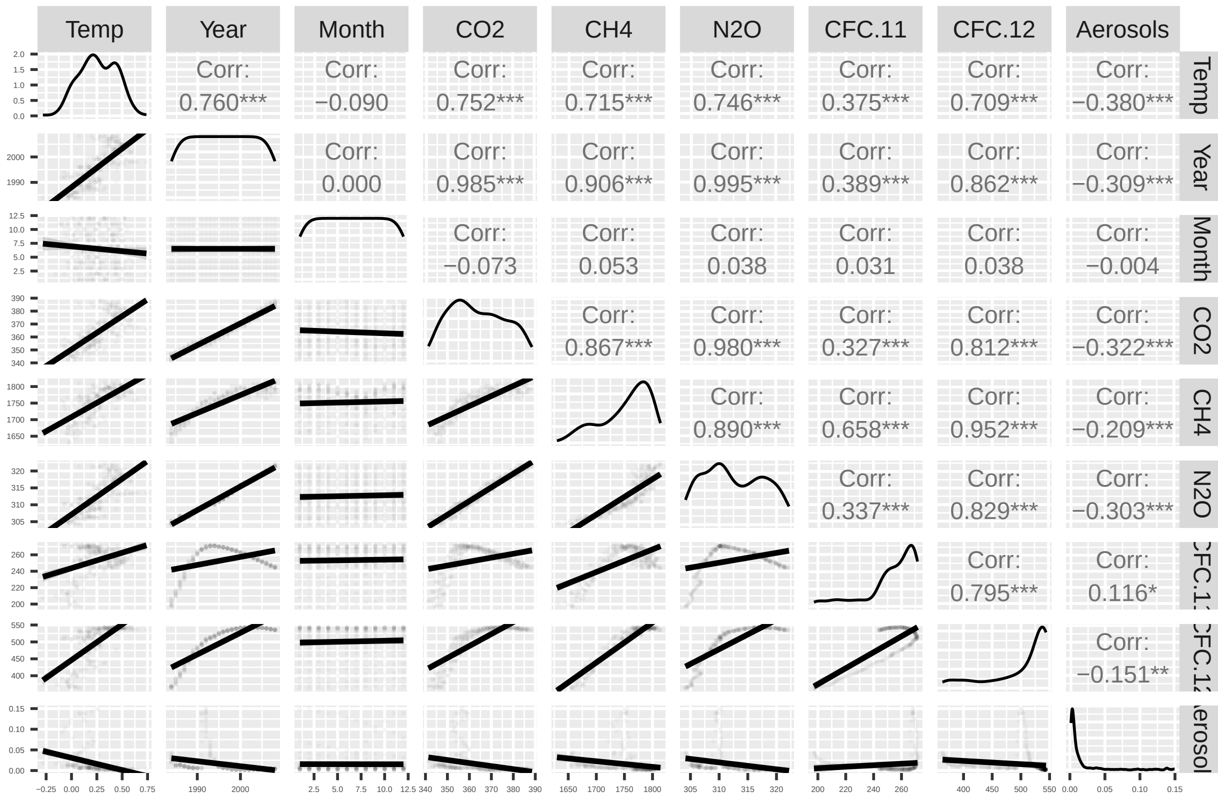
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

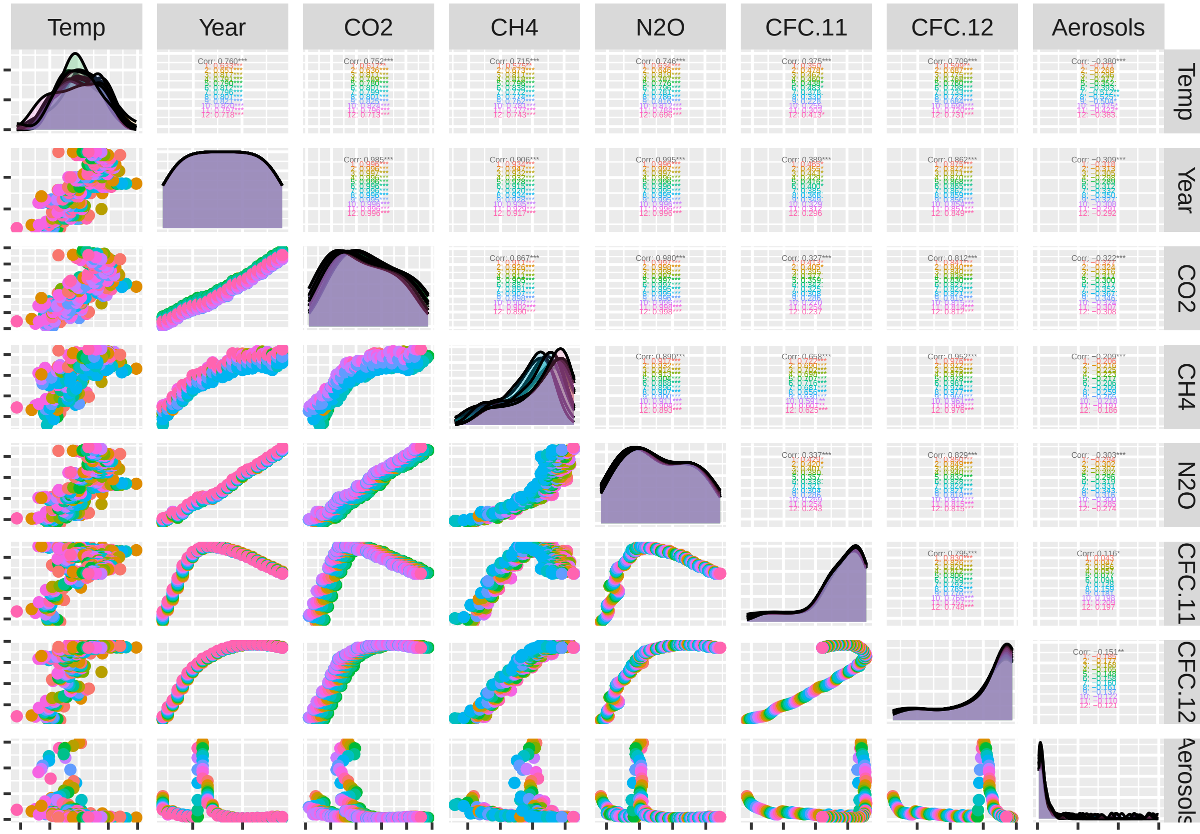
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)

### GROUPINGS OF THESE VECTORS EXPLAINED IN THE PAPER ###
variables <- c("Temp","Year","Month","MEI","CO2","CH4","N2O","CFC.11","CFC.12", "TSI", "Aerosols")
variables_of_interest <- c("Temp","Year","Month","CO2","CH4","N2O","CFC.11","CFC.12", "Aerosols")
variables_of_interest2 <- c("Temp","Year","CO2","CH4","N2O","CFC.11","CFC.12", "Aerosols")

# General Scatter Plot Matrix
climate %>%
  ggpairs(columns = variables_of_interest,
          upper = list(continuous = wrap('cor', size = 3)),
          lower = list(continuous = wrap('smooth',size = .1, alpha = 0.03))) +
  theme_grey() +
  theme(axis.text = element_text(size = 3)) +
  labs(title = "FIGURE 2: Scatter Plot Matrix")
```

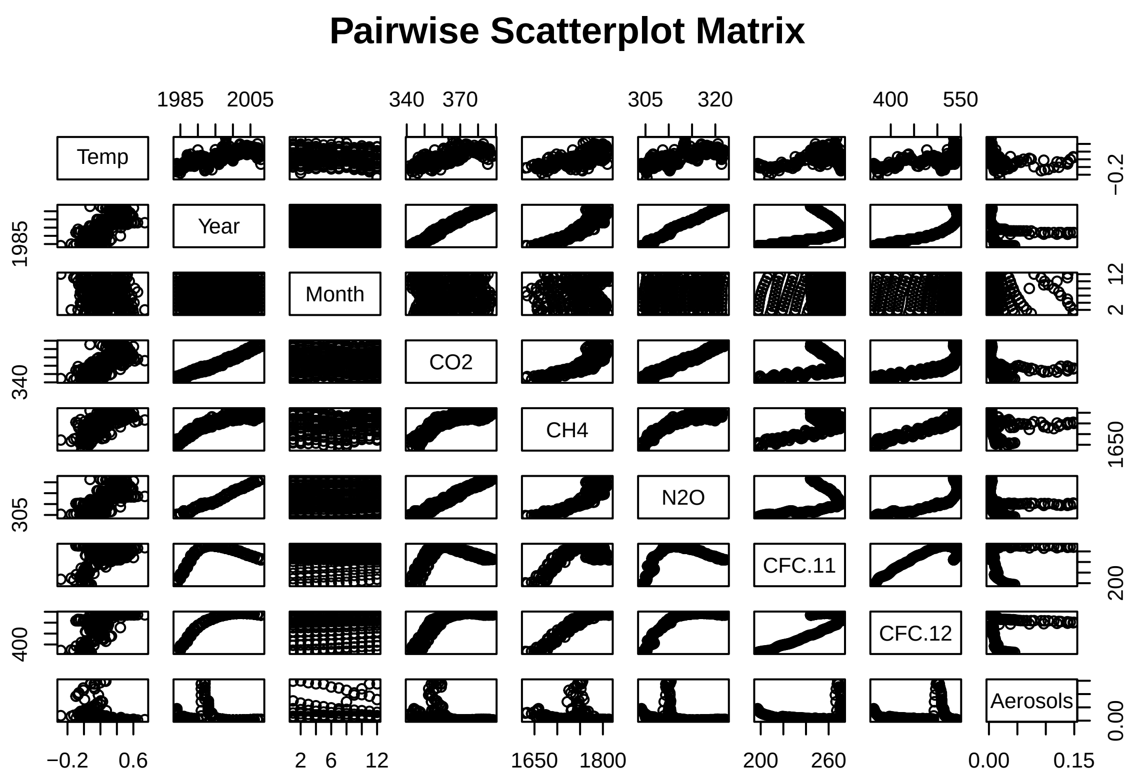
FIGURE 2: Scatter Plot Matrix




```r
# Scatter Plot Matrix Accounting for Categorical Variable
climate %>%
  ggpairs(columns = variables_of_interest2,
          aes(color = factor(Month)),
          upper = list(continuous = wrap('cor', size = 1)),
          lower = list(combo = list(continuous = "smooth", discrete = "boxplot"),
                       size = 0.1, alpha = 0.1),
          diag = list(continuous = wrap('densityDiag', alpha = 0.2))) +
  theme_grey() +
  theme(axis.text = element_text(size = 1))
```




```r
# Lower Half is Correlations; Diagonal Is Density Functions; Upper Half is Corr Values


pairs(climate[variables_of_interest], main = "Pairwise Scatterplot Matrix")
```



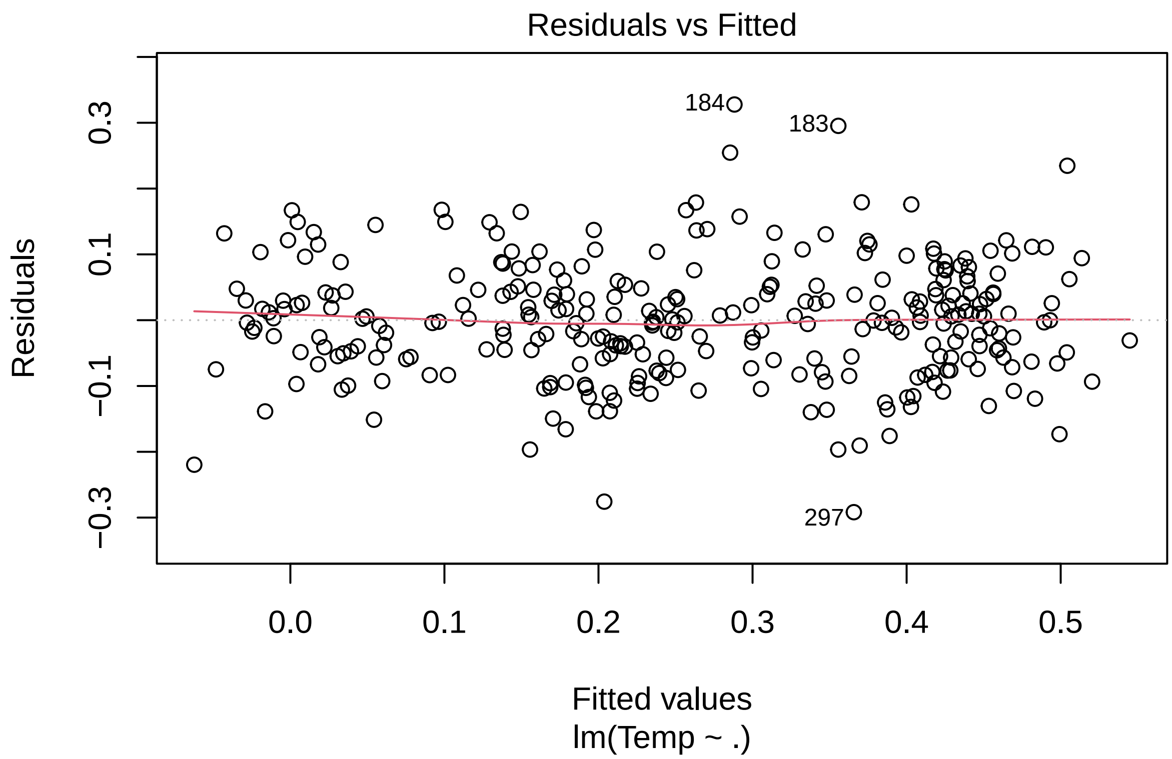

## Testing Linear and Logistic ~ the plots are not actually used, but inform decisions

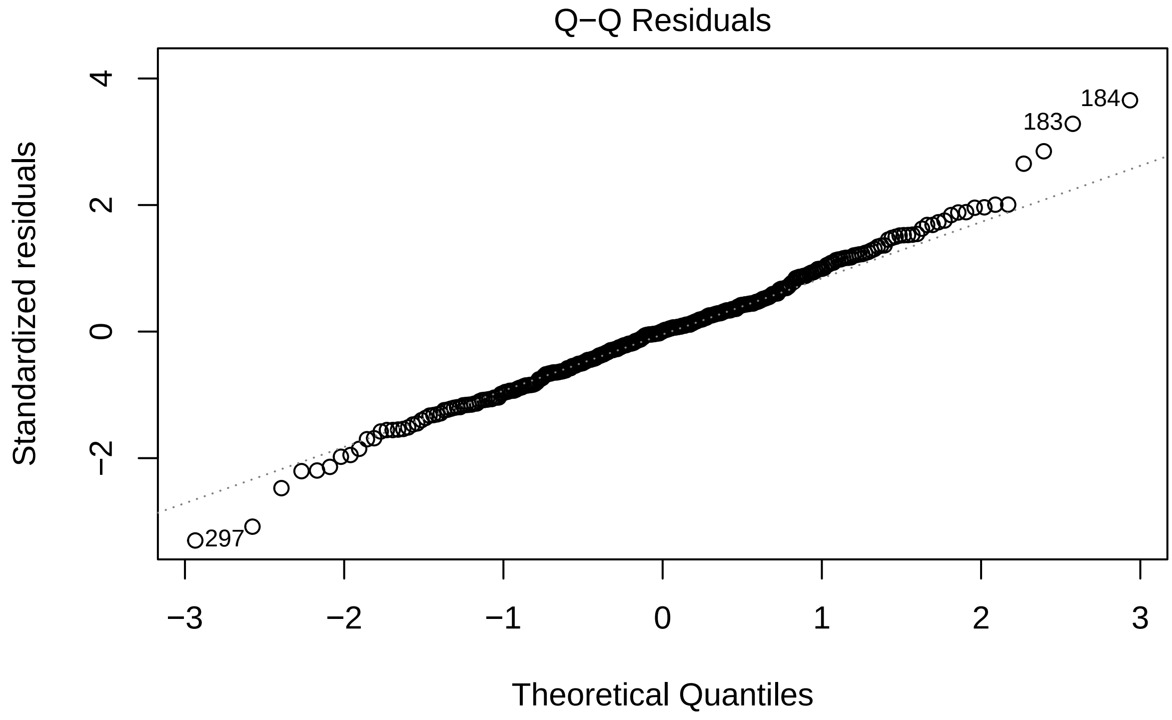
```r
linear_climate_model = lm(Temp ~ ., data = climate)
summary(linear_climate_model)
```

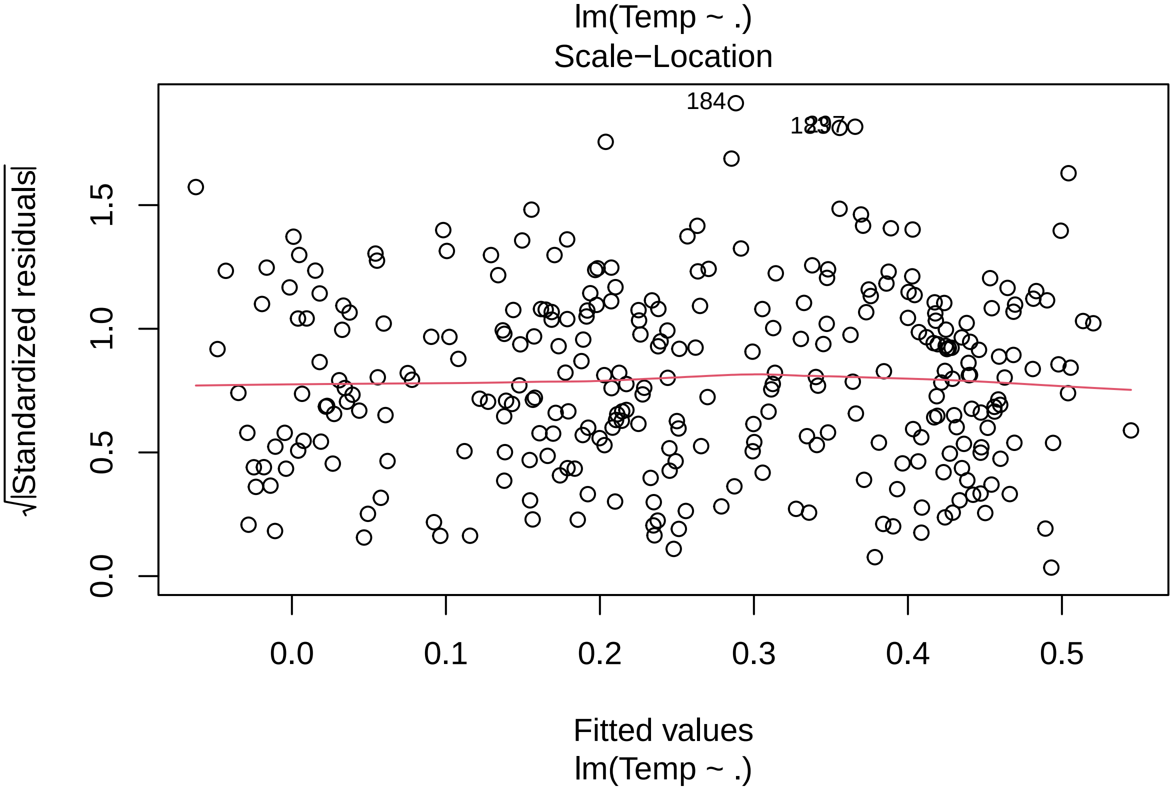
```
##
## Call:
## lm(formula = Temp ~ ., data = climate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29175 -0.05693  0.00049  0.04889  0.32774
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.170e+02  5.334e+01  -2.193  0.02912 *
## Year         6.073e-05  1.927e-02   0.003  0.99749
## Month       -4.667e-03  2.064e-03  -2.262  0.02447 *
## MEI          6.716e-02  6.224e-03  10.790  < 2e-16 ***
## CO2          1.962e-03  3.146e-03   0.624  0.53329
## CH4         -3.832e-05  5.123e-04  -0.075  0.94041
## N2O         -3.048e-03  1.859e-02  -0.164  0.86988
## CFC.11      -5.325e-03  2.001e-03  -2.661  0.00823 **
## CFC.12       3.414e-03  1.416e-03   2.410  0.01656 *
## TSI          8.571e-02  1.897e-02   4.519 9.06e-06 ***
## Aerosols    -1.762e+00  2.239e-01  -7.869 7.20e-14 ***
## ---
```

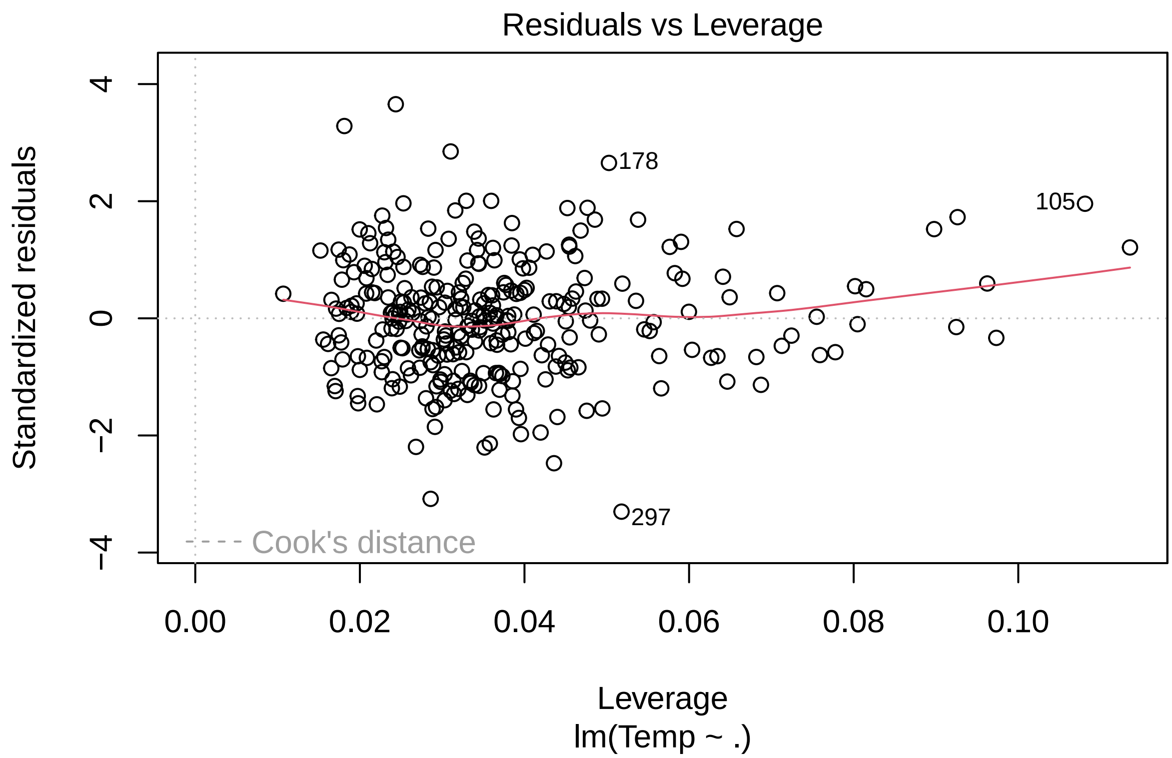
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09076 on 289 degrees of freedom
## Multiple R-squared:  0.7548, Adjusted R-squared:  0.7464
## F-statistic: 88.98 on 10 and 289 DF,  p-value: < 2.2e-16
```

```r
# Exploratory Visualization of the Response Variable ~ FIGURES 2a-2d
plot(linear_climate_model)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(Temp ~ .)
184
183
297


Scale−Location
√|Standardized residuals|
Fitted values
lm(Temp ~ .)
184
183
297

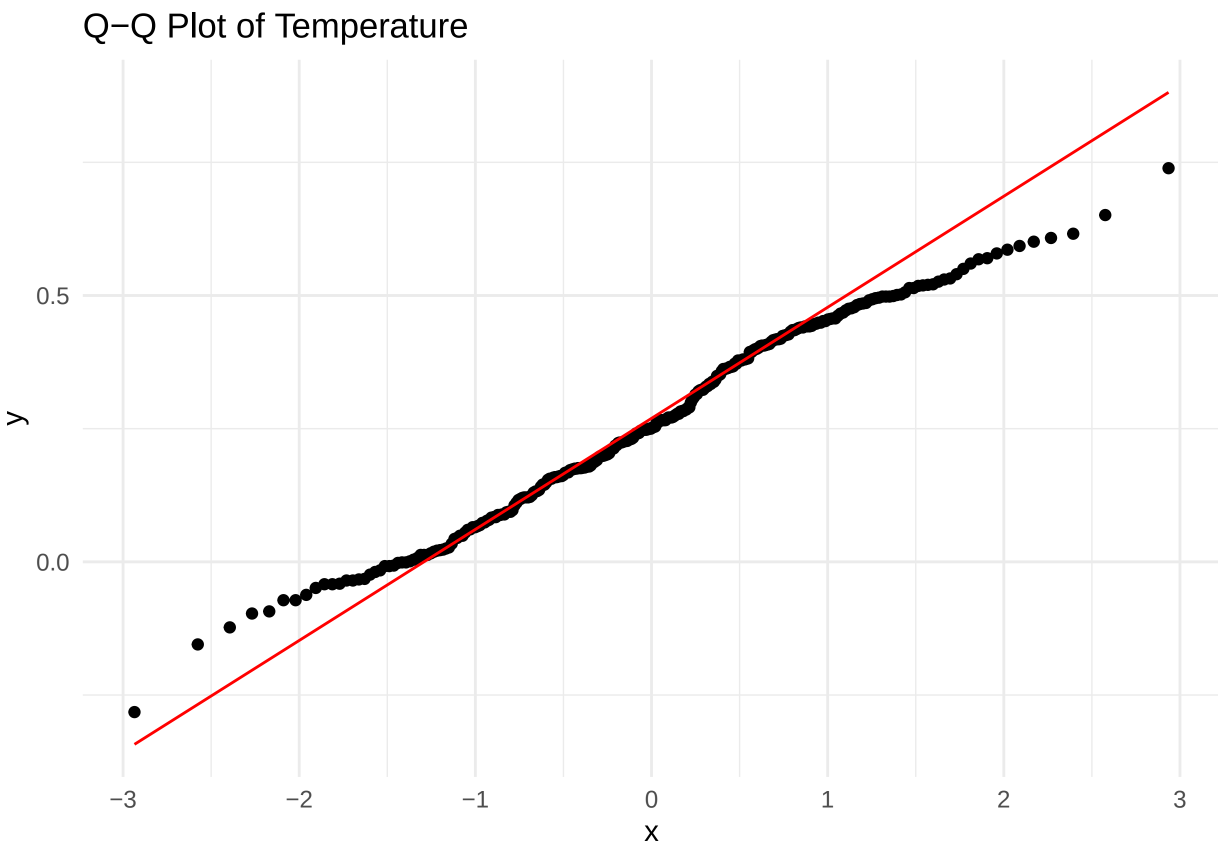



```r
############## PROB NEEDS FIXING ################
binomial_climate_model <- climate %>%
  mutate(HighTemp = ifelse(Temp < mean(Temp), 1, 0)) %>%
  glm(HighTemp ~ . - Temp, data = ., family = "binomial") # deal with multi-collinearity

# balancing sensitivity and specificity of response variable
summary(binomial_climate_model)
```

```
##
## Call:
## glm(formula = HighTemp ~ . - Temp, family = "binomial", data = .)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.565e+02  2.172e+03   0.164  0.86961
## Year         7.452e-01  7.711e-01   0.966  0.33384
## Month        1.364e-01  7.870e-02   1.733  0.08311 .
## MEI         -1.469e+00  3.199e-01  -4.591  4.4e-06 ***
## CO2         -1.206e-02  1.224e-01  -0.099  0.92149
## CH4          8.272e-03  2.024e-02   0.409  0.68275
## N2O         -9.541e-01  7.335e-01  -1.301  0.19336
## CFC.11       1.356e-01  9.458e-02   1.434  0.15167
## CFC.12      -9.209e-02  6.096e-02  -1.511  0.13088
## TSI         -1.131e+00  7.638e-01  -1.481  0.13869
## Aerosols     3.811e+01  1.222e+01   3.118  0.00182 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 415.77  on 299  degrees of freedom
## Residual deviance: 168.15  on 289  degrees of freedom
## AIC: 190.15
##
## Number of Fisher Scoring iterations: 6
```

```r
# NEED TO USE ROC CURVE OR OTHER METHOD TO DETERMINE THRESHOLDS OR
# CLUSTERING/HIERARCHICAL METHODS INSTEAD

##### BRIEF VISUALIZATION W/ GGPLOT #####
ggplot(climate, aes(sample = Temp)) +
  geom_qq() +
  geom_qq_line(col = "red") +
  ggtitle("Q-Q Plot of Temperature") +
  theme_minimal()
```




## Testing LDA ~ Confusion Table

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
lda_climate_model = lda(Temp ~ ., data = climate)
summary(lda_climate_model)
```

```
##         Length Class  Mode
## prior    238   -none- numeric
## counts   238   -none- numeric
## means   2380   -none- numeric
## scaling  100   -none- numeric
## lev      238   -none- character
## svd       10   -none- numeric
## N          1   -none- numeric
## call       3   -none- call
## terms      3   terms  call
## xlevels    0   -none- list
```

```r
# Create a confusion matrix
predictions <- predict(lda_climate_model)
confusion_matrix <- table(Actual = climate$Temp, Predicted = predictions$class)
### WE DID NOT PRINT THE OUTPUTS AS IT IS INCOLCUSIVE AND PRINTS TOO MUCH DATA
```

## Deciding on The Best Model

```r
library(leaps)


#### OLD CODE THAT USES ALL VARIABLES ####
predictor_names <- names(climate)[-which(names(climate) == "Temp")]
all_subsets <- regsubsets(Temp ~ ., data = climate,
                          nvmax = length(predictor_names), method = "exhaustive")
# Get the list of all subsets
all_subsets_list <- summary(all_subsets)$which; all_subsets_list
```

```
##    (Intercept)  Year Month   MEI   CO2   CH4   N2O CFC.11 CFC.12   TSI Aerosols
## 1         TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  FALSE  FALSE FALSE    FALSE
## 2         TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  FALSE  FALSE FALSE    FALSE
## 3         TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  FALSE  FALSE FALSE     TRUE
## 4         TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  FALSE  FALSE  TRUE     TRUE
## 5         TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  FALSE  FALSE  TRUE     TRUE
## 6         TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE   TRUE   TRUE  TRUE     TRUE
## 7         TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE   TRUE   TRUE  TRUE     TRUE
## 8         TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE   TRUE   TRUE  TRUE     TRUE
## 9         TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE   TRUE   TRUE  TRUE     TRUE
## 10        TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE   TRUE   TRUE  TRUE     TRUE
```

```r
#### ~ you would use predictor_names in the for loop instead

included_vars <- c("Year", "MEI", "CO2", "CH4", "N2O", "CFC.11", "CFC.12", "Aerosols")
formula_str <- paste("Temp ~", paste(included_vars, collapse = " + "))
formula <- as.formula(formula_str)
all_subsets <- regsubsets(formula, data = climate,
                           nvmax = length(included_vars), method = "exhaustive")

####### WE CAN CHANGE THE METHOD OF SEARCH but since the algorithm returns the best
# model of each size (number of parameters 2-10 or number of predictors 1-9),
```

```r
# so the results do not depend on a penalty model for model size: it doesn't make
# any difference whether you want to use AIC, BIC, CIC, DIC #######

# Initialize a dataframe to store model specifications, MSE, and adjR2
model_info <- data.frame(
  model = character(),
  MSE = numeric(),
  R2 = numeric(),
  adjR2 = numeric(),
  stringsAsFactors = FALSE
)

# Extract Values for Every Model Size (1-9 predictors)
for (i in 1:length(included_vars)) {
  # Extract the coefficients for the best model of size i
  model_coef <- coef(all_subsets, id = i)
  model_formula <- as.formula(paste("Temp ~",
                  paste(names(model_coef)[-1], collapse = "+"))) #create formulas
  # Fit Models
  fit <- lm(model_formula, data = climate)

  # Calculate MSE
  predictions <- predict(fit, newdata = climate)
  mse <- mean((climate$Temp - predictions)^2)
  # Get R2 Values
  r2 <- summary(fit)$r.squared
  adj_r2 <- summary(fit)$adj.r.squared

  # Store model information
  model_info <- rbind(model_info, data.frame(
    model = deparse(model_formula),
    MSE = mse,
    R2 = r2,
    adjR2 = adj_r2,
    stringsAsFactors = FALSE
  ))
}

model_info
```

```
##                                                                  model         MSE
## 1                                                          Temp ~ Year 0.013655576
## 2                                                    Temp ~ Year + MEI 0.011683088
## 3                                         Temp ~ Year + MEI + Aerosols 0.009825178
## 4                              Temp ~ MEI + CFC.11 + CFC.12 + Aerosols 0.009338815
## 5                        Temp ~ MEI + CO2 + CFC.11 + CFC.12 + Aerosols 0.009240142
## 6                 Temp ~ Year + MEI + CO2 + CFC.11 + CFC.12 + Aerosols 0.009074121
## 7           Temp ~ Year + MEI + CO2 + N2O + CFC.11 + CFC.12 + Aerosols 0.009008043
## 8 Temp ~ Year + MEI + CO2 + CH4 + N2O + CFC.11 + CFC.12 + Aerosols 0.008995450
##          R2     adjR2
## 1 0.5781145 0.5766987
## 2 0.6390540 0.6366234
## 3 0.6964536 0.6933771
## 4 0.7114797 0.7075675
```

```
## 5 0.7145282 0.7096732
## 6 0.7196573 0.7139165
## 7 0.7216988 0.7150272
## 8 0.7220878 0.7144476
```

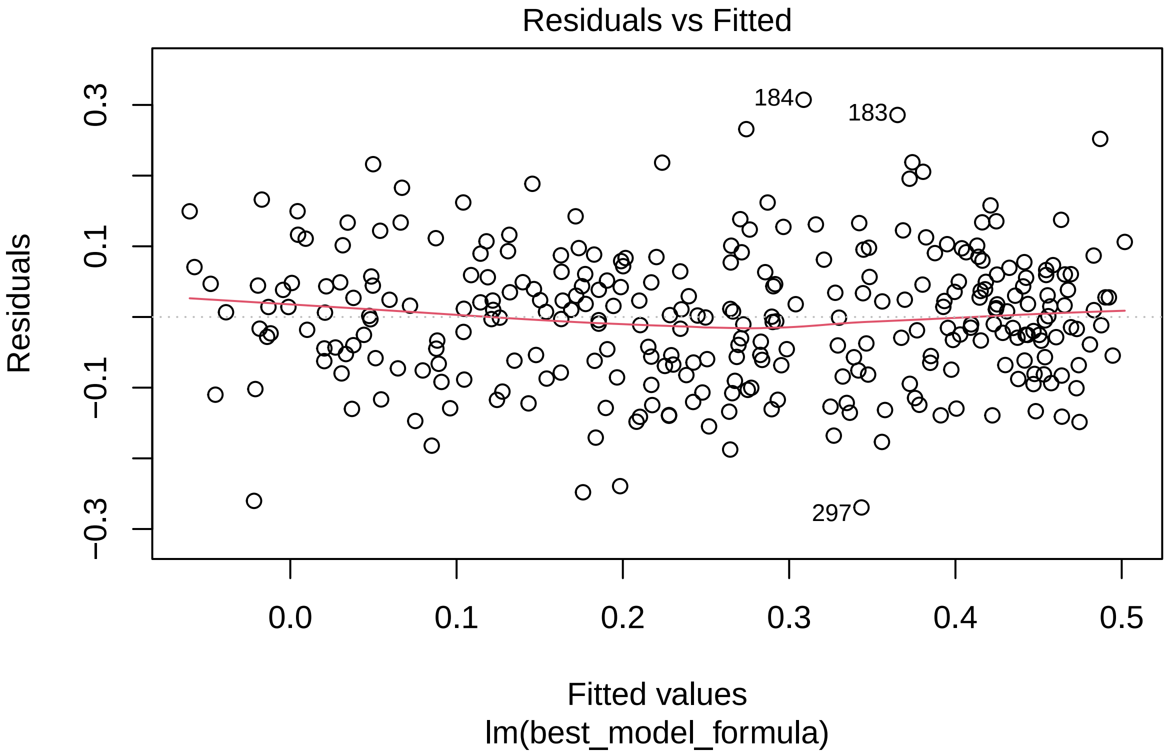
## Selection Criteria

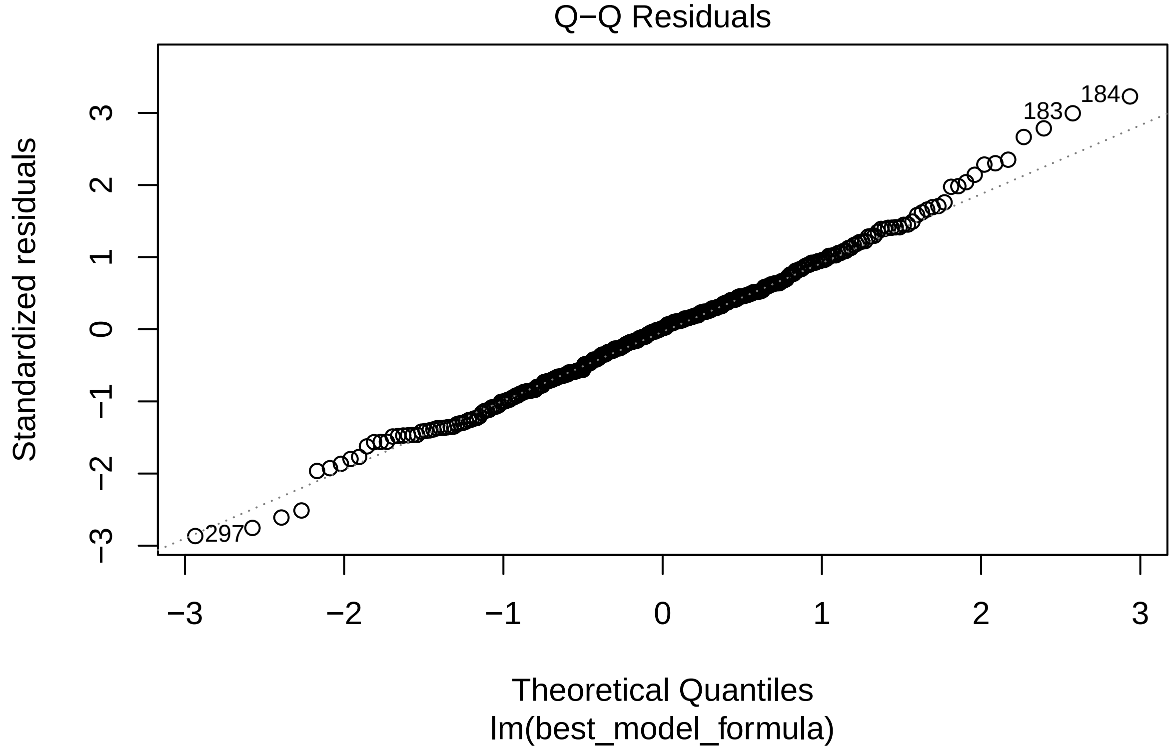
```r
# Regsubsets Identifies the Best Model at each # of predictors (1-9);
# We decide on the best model by considering the model with the lowest MSE that
# does not overfit the data (the adjusted R2 is not significantly smaller than the R2)

filter <- model_info[model_info$adjR2 >= (0.95 * model_info$R2), ]

# Choose the model with the lowest MSE from the filtered models
best_model <- filter[which.min(filter$MSE), ]

print(best_model)
```

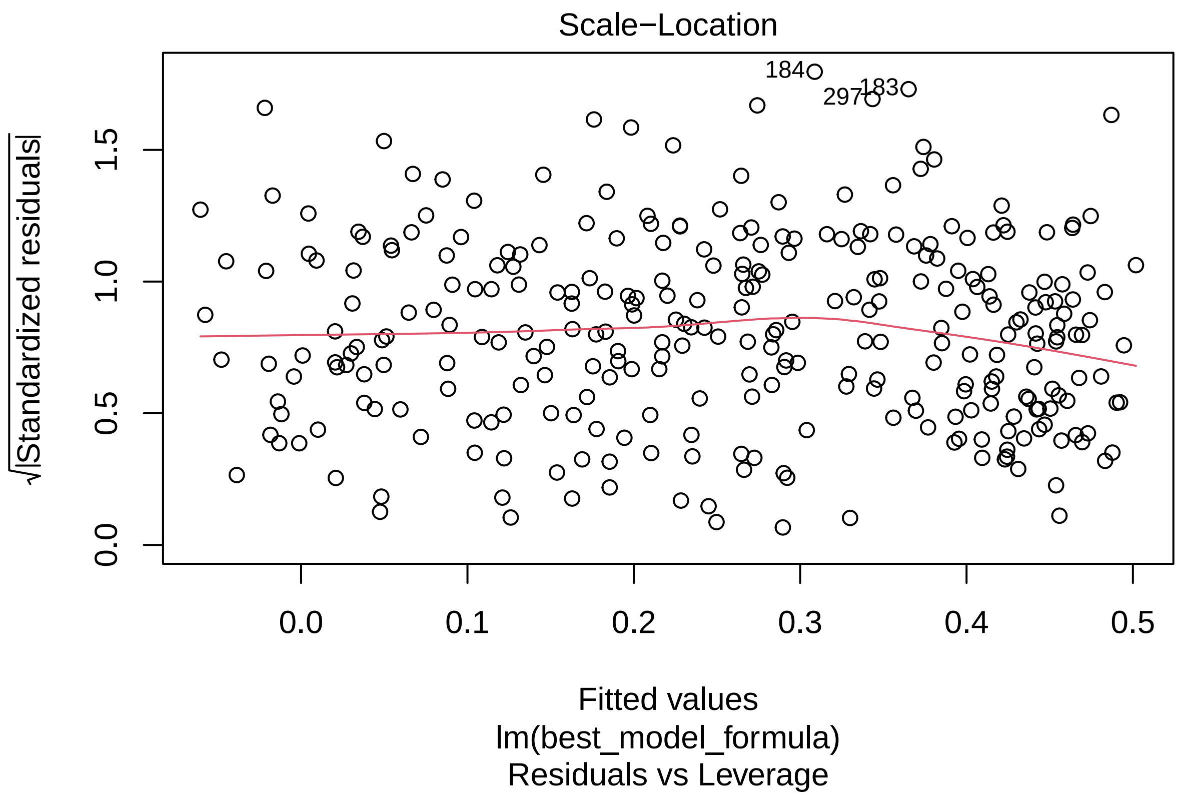
```
##                                                              model        MSE
## 8 Temp ~ Year + MEI + CO2 + CH4 + N2O + CFC.11 + CFC.12 + Aerosols 0.00899545
##          R2     adjR2
## 8 0.7220878 0.7144476
```

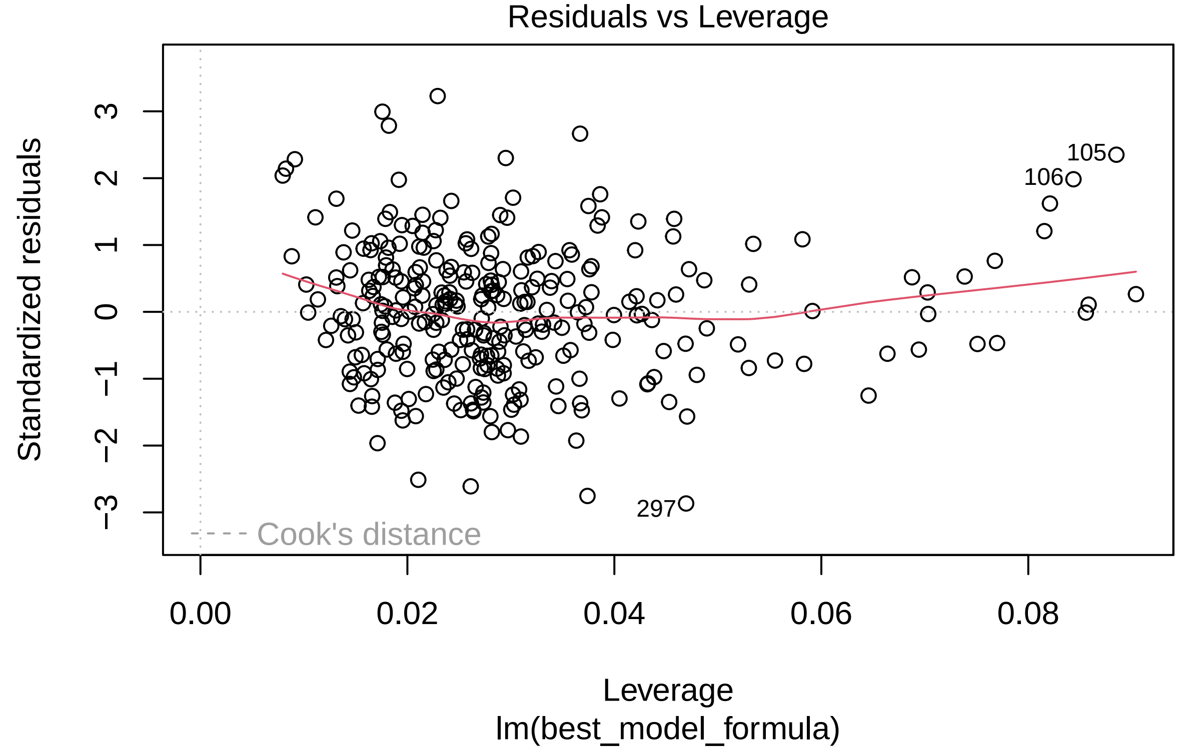
## Analyzing Our Model: Figure 3

```r
best_model_formula <- as.formula(best_model$model)


# Fit the best model
best_fit <- lm(best_model_formula, data = climate)
# FIGURE 3 (a-d)
plot(best_fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(best_model_formula)
184
183
297


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(best_model_formula)
184
183
297




```
# Analysis ~ t-test
summary(best_fit)
```

```
##
## Call:
## lm(formula = best_model_formula, data = climate)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.269500 -0.064557  0.001331  0.057791  0.307282 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 69.0353507 25.7736212   2.679  0.00782 ** 
## Year        -0.0398327  0.0148996  -2.673  0.00793 ** 
## MEI          0.0618198  0.0065293   9.468  < 2e-16 ***
## CO2          0.0097461  0.0030543   3.191  0.00157 ** 
## CH4         -0.0003409  0.0005341  -0.638  0.52381    
## N2O          0.0217293  0.0136837   1.588  0.11338    
## CFC.11      -0.0066150  0.0020318  -3.256  0.00126 ** 
## CFC.12       0.0053424  0.0013216   4.042 6.78e-05 ***
## Aerosols    -1.7626952  0.2339216  -7.535 6.20e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.0963 on 291 degrees of freedom
## Multiple R-squared:  0.7221, Adjusted R-squared:  0.7144 
## F-statistic: 94.51 on 8 and 291 DF,  p-value: < 2.2e-16
```

```r
qt(1-0.05/2, 300-8-1)
```

```
## [1] 1.96815
```

```r
# F-stat overall significance of the model
summary(best_fit)$fstatistic[1]
```

```
##    value 
## 94.51168
```

```r
qf(1-0.05,df1 = 8, df2 = 300-8-1)
```

```
## [1] 1.970285
```

```r
anova_table <- anova(best_fit)
print(anova_table)
```

```
## Analysis of Variance Table
## 
## Response: Temp
##            Df Sum Sq Mean Sq  F value    Pr(>F)    
## Year        1 5.6137  5.6137 605.3399 < 2.2e-16 ***
## MEI         1 0.5917  0.5917  63.8094 3.220e-14 ***
## CO2         1 0.0151  0.0151   1.6299   0.20274    
## CH4         1 0.0260  0.0260   2.8026   0.09519 .  
## N2O         1 0.0184  0.0184   1.9817   0.16028    
## CFC.11      1 0.0030  0.0030   0.3268   0.56797    
## CFC.12      1 0.2172  0.2172  23.4208 2.115e-06 ***
## Aerosols    1 0.5266  0.5266  56.7824 6.195e-13 ***
## Residuals 291 2.6986  0.0093                       
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Calculating F-val
mean_sq_model <- sum(anova_table$`Mean Sq`[1:8])
```

```r
F_value <- mean_sq_model /anova_table$`Mean Sq`[9]
# F-stat compares the variability explained by the predictors (Mean Sq Model)
# with the variability not explained by the model (Mean Sq Residuals)
F_value
```

```
## [1] 756.0935
```

## F-test Lack of Fit

```r
residuals <- residuals(best_fit)
fitted_values <- fitted(best_fit)

# Calculate lack-of-fit sum of squares
n <- length(residuals)
mean_residuals <- mean(residuals)
lack_of_fit_ss <- sum((residuals - mean_residuals)^2)

# Calculate residual sum of squares
residual_ss <- sum(residuals^2)

# Degrees of freedom for the lack-of-fit test
df_lack_of_fit <- n - length(coefficients(best_fit)) #Adjust for number of coefficients

# Degrees of freedom for residuals
df_residuals <- df.residual(best_fit)

# Calculate F-value for lack of fit
F_lack_of_fit <- (lack_of_fit_ss / df_lack_of_fit) / (residual_ss / df_residuals)
F_critical <- qf(1 - 0.05, df_lack_of_fit, df_residuals)

# Calculate p-value for lack of fit
p_value_lack_of_fit <- pf(F_lack_of_fit, df_lack_of_fit, df_residuals, lower.tail = FALSE)

# Print results
cat("F-critical for LOF:", F_critical, "\n")
```

```
## F-critical for LOF: 1.213079
```

```r
cat("F-value for lack of fit:", F_lack_of_fit, "\n")
```

```
## F-value for lack of fit: 1
```

## Cross Validation

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
# Leave-One-Out-Cross-Validation
train_control_loocv <- trainControl(method = "LOOCV")
loocv <- train(best_model_formula, data = climate, method = "lm", trControl = train_control_loocv)

# Calculate MSE for LOOCV
```

```r
mse1 <- loocv$results$RMSE^2
cat("LOOCV MSE:", mse1, "\n")
```

```
## LOOCV MSE: 0.009562674
```

```r
# K-fold cross-validation
train_control_kfold <- trainControl(method = "cv", number = 10)
kfold <- train(best_model_formula, data = climate, method = "lm", trControl = train_control_kfold)

# Calculate MSE for K-fold CV
mse2 <- kfold$results$RMSE^2
cat("K-fold MSE:", mse2, "\n")
```

```
## K-fold MSE: 0.009360922
```

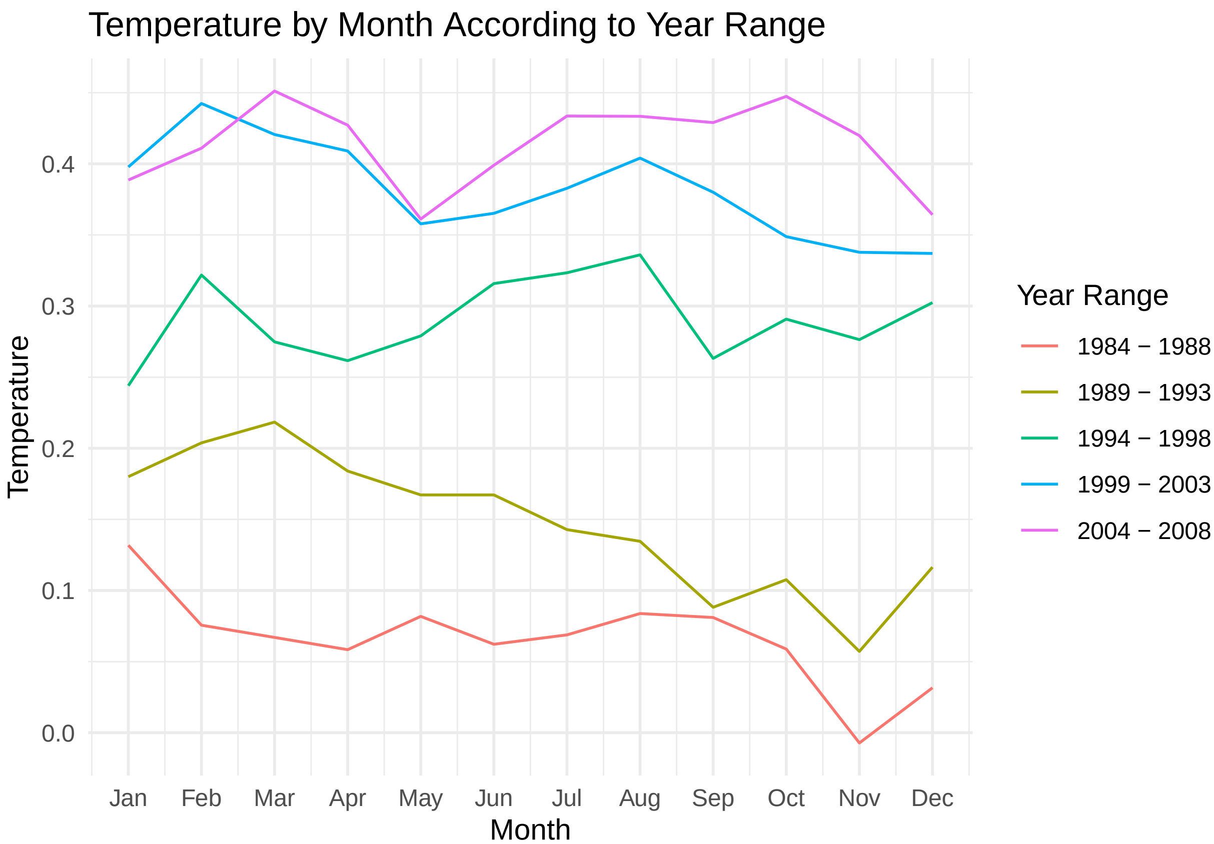
## Additional EXPLORATORY Analysis

```r
split_month <- climate %>%
  split(.$Month) %>%
  lapply(function(.) {
    .[order(.$Year, decreasing = FALSE), ]
  })

# split_month
### NOT PRINTED AS THE OUTPUT IS TOO LONG
```

```r
aggregate_temperature <- function(df) {

  # Create a new column that groups years into sets of 5
  df <- df %>%
    mutate(YearGroup = (row_number() - 1) %/% 5 + 1)

  # Calculate the average temperature for each group and rename the year group
  result <- df %>%
    group_by(YearGroup) %>%
    summarise(
      YearRange = paste(min(Year), max(Year), sep = " - "),
      Temperature = mean(Temp),
      Month = first(Month) # unchanged
    ) %>%
    ungroup()

  return(result)
}

# Apply the function to each split_month dataframe
bymonth <- lapply(split_month, aggregate_temperature)
# bymonth
### NOT PRINTED AS THE OUTPUT IS TOO LONG
```

## FIGURE 1 Exploration of How the Categorical Variable of Month Affects Temperature by Year

```
by_yeargroup <- list()

for (i in 1:5) {
  new_df <- do.call(rbind, lapply(bymonth, function(df) df[i, ]))
  by_yeargroup[[i]] <- new_df
}
# by_yeargroup
### NOT PRINTED AS THE OUTPUT IS TOO LONG

plot <- ggplot() +
  labs(title = "Temperature by Month According to Year Range",
       x = "Month",
       y = "Temperature") +
  scale_color_discrete(name = "Year Range") +
  scale_x_continuous(breaks = 1:12, labels = month.abb) +  # Set x-axis breaks and labels
  theme_minimal()

# Iterate through each data frame in the by_yeargroup list and add a geom_line() for each
for (i in 1:length(by_yeargroup)) {
  plot <- plot + geom_line(data = by_yeargroup[[i]], aes(x = Month,
                                     y = Temperature, color = factor(YearRange)))
}

plot
```


Temperature by Month According to Year Range
Temperature
0.0
0.1
0.2
0.3
0.4
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month
Year Range
1984 − 1988
1989 − 1993
1994 − 1998
1999 − 2003
2004 − 2008