Regression Analysis Final Project

Madison Poore

2024-12-03

0.1 Introduction

Coloradoans are typically seasoned professionals at driving in snow. However, there are some roads in Colorado that become particularly difficult to navigate after a storm. Snow storms cause snow (and ice) to accumulate on the roads which lead to a higher amount of traffic accidents. If we are able to predict the depth of snow on the roads from a storm, we can determine how to take proper safety measures. For example, based on how much snow we expect to "stick", we could determine how many snow plows are needed for proper snow removal.

This specific paper takes an observational dataset from Kaggle that provides snow data over 4 years on Vail Pass. I chose this dataset since the combination between ski traffic and snowy roads leads to an absurd amount of accidents. Further, there is relatively consistent snow throughout the winters on this pass and this allows for lots of snow-filled observations. This dataset was originally collected with avalanches in mind, though the number of observations with an avalanche occurring were few enough to not provide much information.

There is no prior research required to understand the results of this paper, though a knowledge of simple regression analysis would be helpful.

Research Question: Can we predict snow depth based on season, snow water equivalent, precipitation accumulation, minimum temperature and maximum temperature?

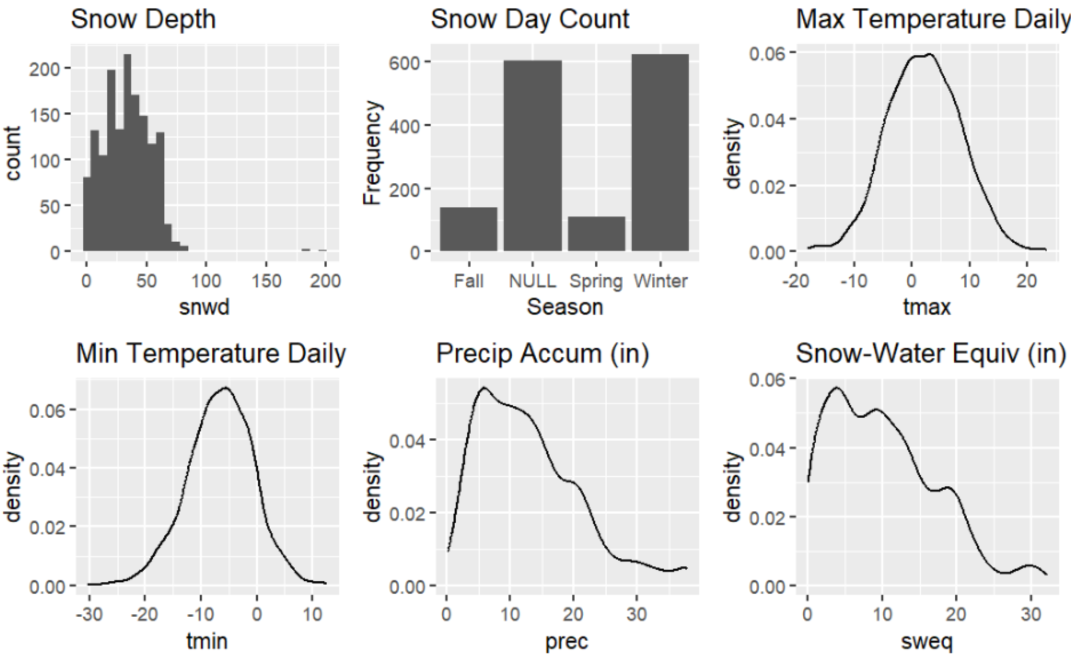
0.2 Results

Data Exploration

Here is a table describing each variable used for the regression model.

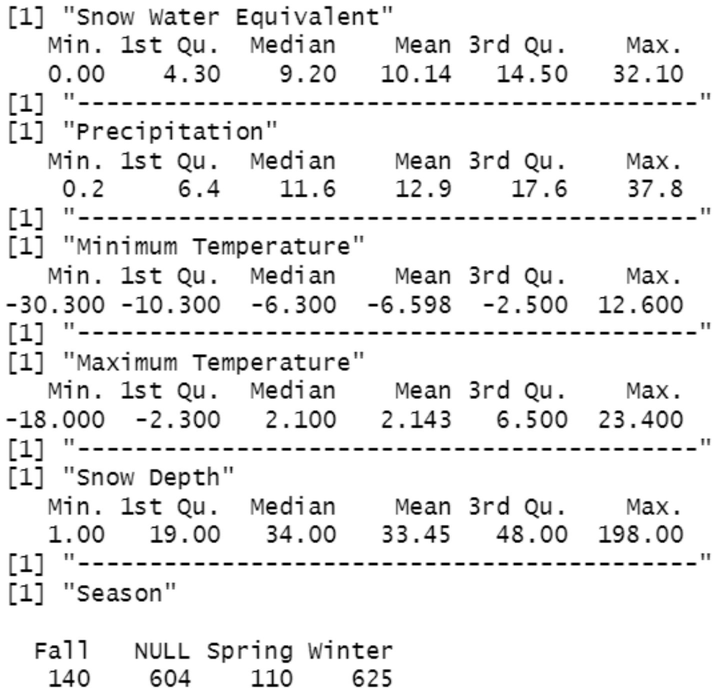
Variable	Description
Season	Character; Spring, Summer, Fall, Winter.
Snow-Water Equivalent	Numeric Variable; Measures the amount of water present in snowfall. That is, when the snow melts, how much water will it turn into? This is measured in inches.
Precipitation Accumulation	Numeric Variable; This measures the amount of precipitation within a given period of time. In this case, it is in number of inches in 1 day.
Minimum Temperature	Numeric Variable; Measured in Celsius, temperature of the air at it's coldest in one day
Maximum Temperature	Numeric Variable; Measured in Celsius, maximum temperature of the air for a day
Snow Depth	Numeric Variable; Depth of snow, measured in inches.

As a portion of the initial data exploration, here are all 6 visual summaries of the variables described in the table above. Reference the **Appendix Section I** for code used to generate this output. Also note this is computed after necessary data cleaning and transformations.



It is important to note the Snow Day Count has a category called `NULL`, which is a cause of deleting elements that have a `snwd` ≤ 0. This `NULL` category therefore represents summer days with no snow, which happens to be all summer days. The continuous variables in this data set are uni-modal (mostly). There are some outliers in the `snwd` which represent days with heavy snow. Temperature seems normally distributed while precipitation and snow water equivalent seem to be skew right distributions.

To take a more in-depth exploration of this data, we can look at numerical summaries. Please reference **Appendix Section II** for the code related to these results.

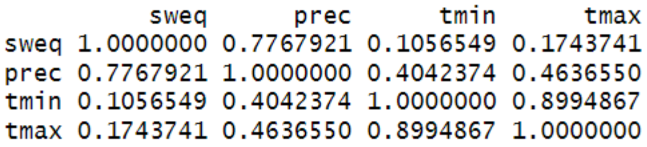


```
[1] "Snow Water Equivalent"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    4.30    9.20   10.14   14.50   32.10
[1] "-------------------------------------------"
[1] "Precipitation"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.2     6.4    11.6    12.9    17.6    37.8
[1] "-------------------------------------------"
[1] "Minimum Temperature"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-30.300 -10.300  -6.300  -6.598  -2.500  12.600
[1] "-------------------------------------------"
[1] "Maximum Temperature"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-18.000  -2.300   2.100   2.143   6.500  23.400
[1] "-------------------------------------------"
[1] "Snow Depth"
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   19.00   34.00   33.45   48.00  198.00
[1] "-------------------------------------------"
[1] "Season"

  Fall   NULL Spring Winter 
   140    604    110    625 
```

The numerical summaries are presented in the following order: `sweq`, `precip`, `tmin`, `tmax`, `snwd`, `season` . Note once again that we can think of the `NULL` variables as representative of summer days with no observation of snow. It is important to note that the Minimum Temperature and Maximum Temperature are both recorded in Celsius. Note that the maximum temperature on days with snow tend to be around 0 degrees Celsius (which is freezing point).

Collinearity Check

Now that we have performed an initial data exploration, we can check for collinearity and see if we need to amputate variables. Please reference **Appendix Section III** for code referenced related to collinearity. We have five potential regressors, four of which are numerical. So, we need to check between our four numerical variables whether or not they are collinear. We do this by computing a correlation matrix in R, then looking for values in the matrix that are not on the diagonal and 'close' to 1. Here is our correlation matrix:



```
          sweq      prec      tmin      tmax
sweq 1.0000000 0.7767921 0.1056549 0.1743741
prec 0.7767921 1.0000000 0.4042374 0.4636550
tmin 0.1056549 0.4042374 1.0000000 0.8994867
tmax 0.1743741 0.4636550 0.8994867 1.0000000
```

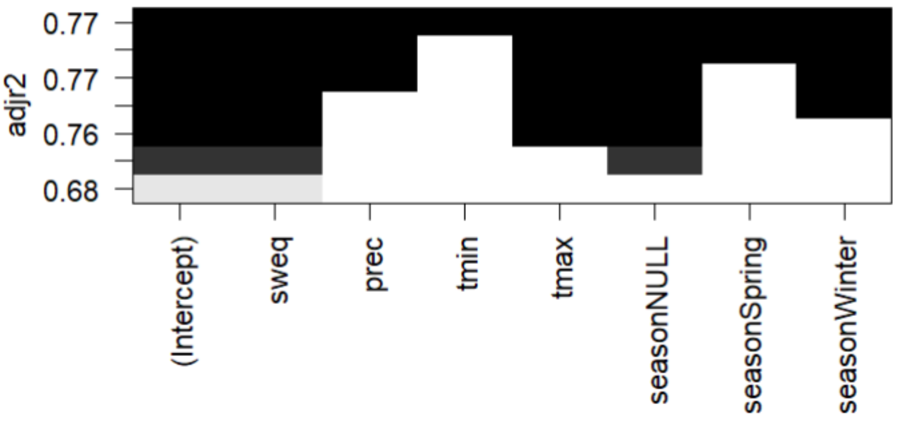
We expect the diagonal to be 1 since each variable is entirely co-linear with itself. We notice the off-diagonal entries are generally low with the exception of a high collinearity between `tmin` and `tmax`. Let's just take note of the collinearity for now, in order to maintain a reasonable number of variables. This collinearity issue will later resolve itself in the variable selection section of this paper.

Variable Selection

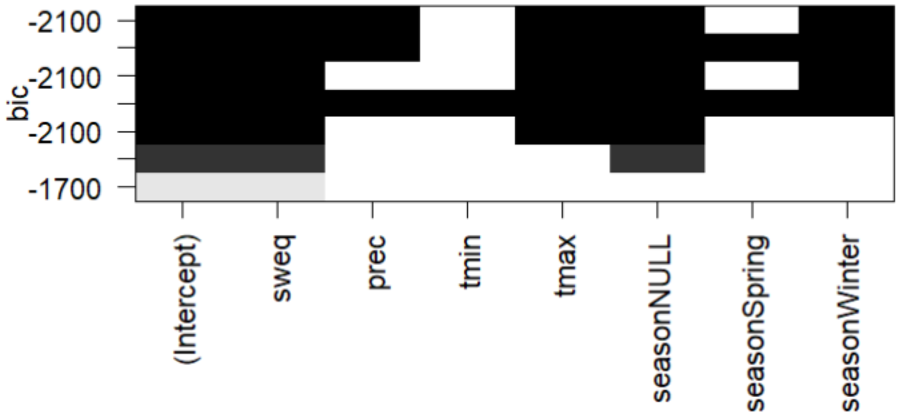
Moving on to variable selection, we begin by iterating through two activities: searching for the 'best' model, and selecting the 'best' model. Please reference **Appendix Section IV** for code related to variable selection.

That is, we want to see if we can simplify this model as much as reasonably possible. The code in the appendix references the best subset variable selection process. I chose to use R_a^2, BIC statistic, and Mallow's C_p to help pick an optimal model. I will favor simplicity if there is not a unanimous choice of regressors. This is in hopes of keeping computational expense as low as possible (and avoiding previously discussed collinearity).

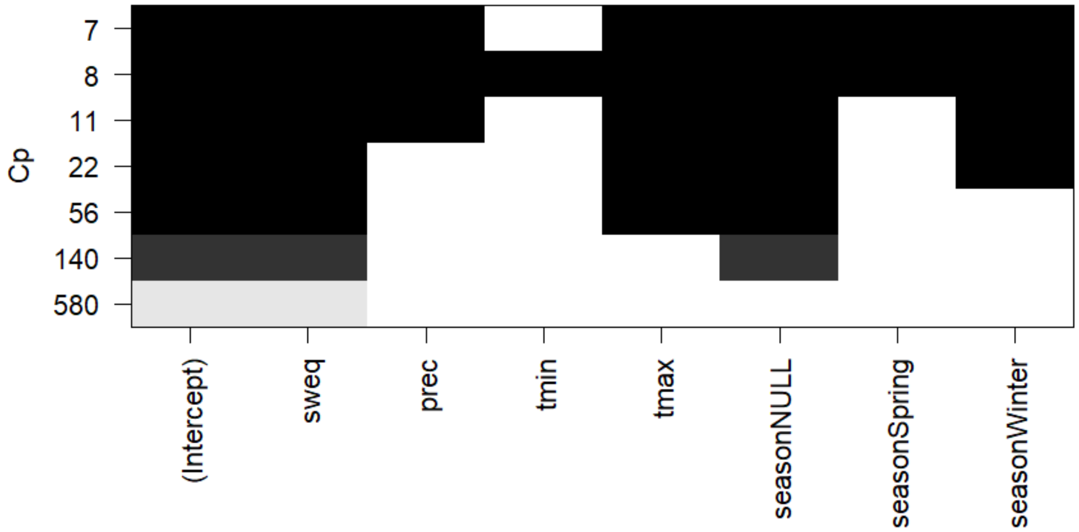
According to the R_a^2, the optimal model includes the intercept and all regressors in the complete model. Here is the graphic that determines that, where we can read off the colored in boxes in the top row as the regressors included in the model.



According to the BIC statistic, the optimal model includes the intercept, `sweq`, `prec`, `tmax`, summer, and winter seasons. Though it does not make sense to include some but not all categorical variables. Again, we reach this conclusion by reading off the top row of the graphic and looking for 'colored in' boxes that represent inclusion. From the BIC statistic, we may conclude the optimized model is the complete model without `tmin`.



According to Mallow's C_p, the optimal model includes: the intercept, `sweq`, `prec`, `tmax`, `season`.



Multiple variable selection processes suggest that we should include all possible variables except from `tmin`. This also resolves the colinearity issue briefly mentioned above.

So, our model will look something like this:

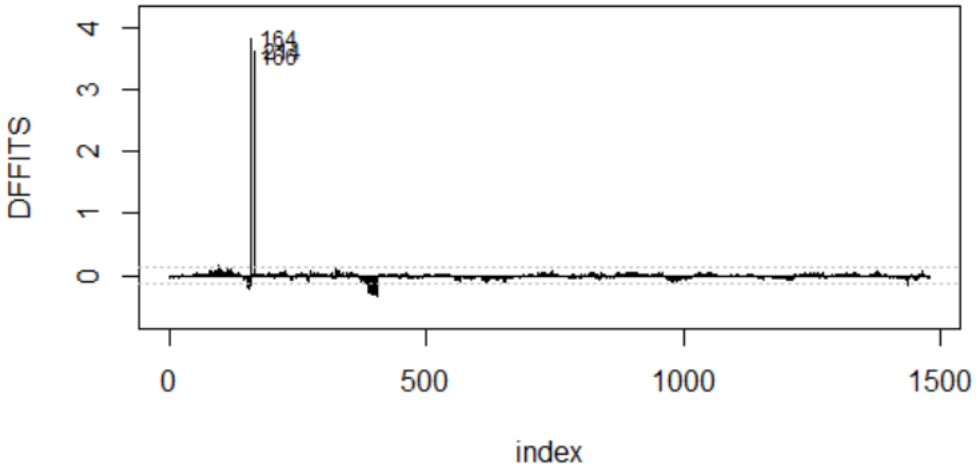
$$Y = \beta_0 + \beta_1 X_{\text{sweq}} + \beta_2 X_{\text{prec}} + \beta_3 X_{\text{tmax}} + \beta_4 \texttt{seasonNULL} D_1 + \beta_5 \texttt{seasonSpring} D_2 + \beta_6 \texttt{seasonWinter} D_3$$

D_i is the indicator variable referring to season. For example, if the observation was recorded in spring, $D_1, D_3 = 0$ and $D_2 = 1$.

Now we can move on to check assumptions

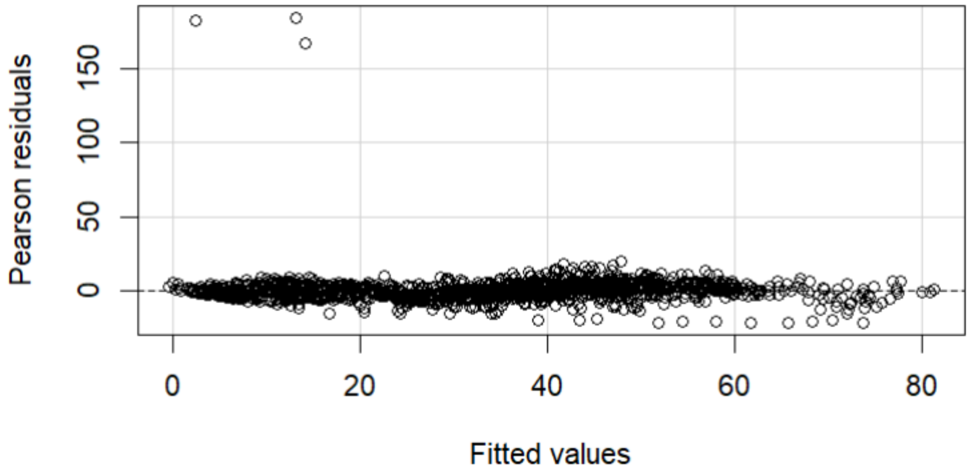
Checking for Influential Observations

Please reference **Appendix Section V** for code used in this section. Using an index plot, we can see we have 3 outliers in our model. The appendix references a further investigation of the outliers. The first two observations say there is snow in the summer, which we can certainly filter out. The third outlier is a Fall snow, and nothing looks improperly recorded. We test whether or not this observation affects the model substantially (again, in the appendix). The appendix shows that the third observation does not affect the model if we include it or if we don't, so we can continue our work. Here is the index plot that identified the outliers.



Checking Structure

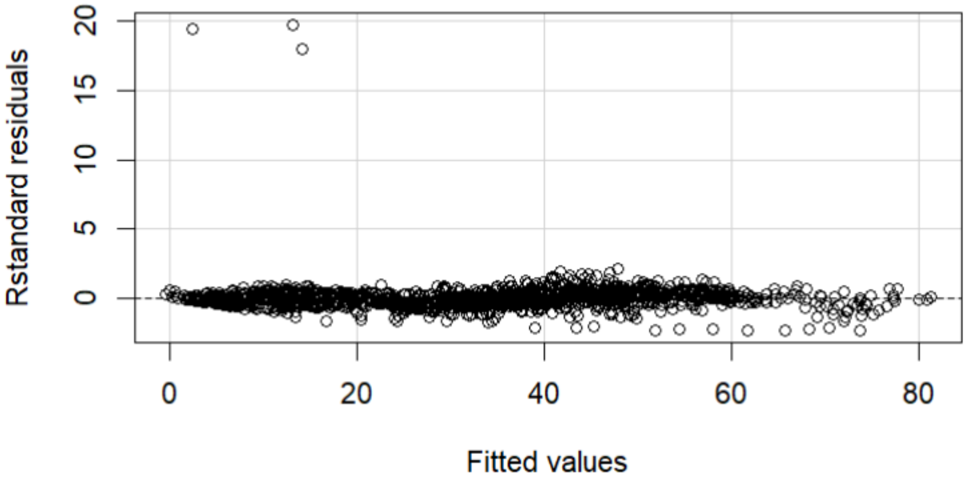
Please reference **Appendix Section VI** to see code for the assumption section of this report. To check the structure of our model, we can look at the `residualPlot()` of our model. If our model seems to (at least) approximately sit along 0, then we can verify our structure assumption. Again we can see the outliers, though we do not have a method to filter them out since they seem properly recorded.



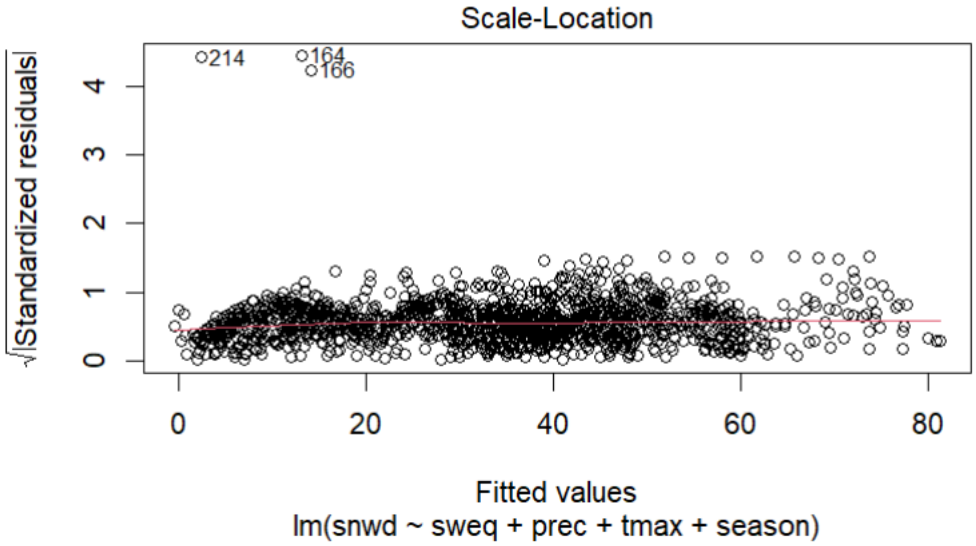
Since the residual plot doesn't provide any substantial cause for concern, we can conclude that our structure assumptions are valid. There are a few outliers, though the majority of the data lies along 0, so we don't have much cause for concern. Data is reasonably along $y = 0$ and evenly distributed, so we can move on.

Checking Error & Variance Assumptions

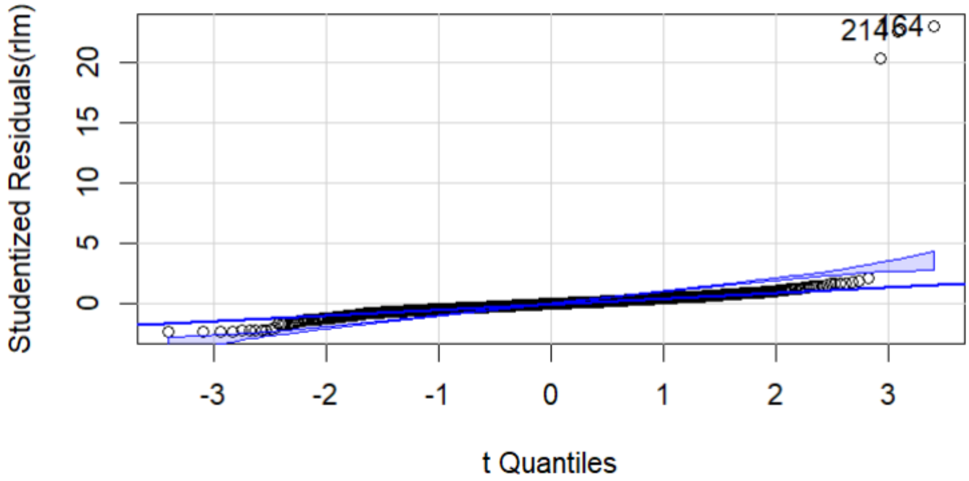
Now we can check if our assumption of constant variance was valid. We are looking for most of the data to be equally distributed around 0. See **Appendix Section VI** for the code to generate the standardized residual plot here:



There **might** be a slight variation in the distribution of measurements. It is fairly difficult to see whether or not there is a distinguishable variability in the data. So, we can look at a scale-location plot for the fitted model. Here is the scale-location plot:



Since the red line is very flat, the variance does seem to be constant. Once again, we do not have much reason for concern with our constant variance assumption. We can now move on to checking for normality of our errors. We do this by generating a **Q-q** plot. Again reference **Appendix Section VI** for code associated to this.



There is substantial cause for concern here since the points are outside the 95% confidence envelopes of the qunatiles. But, we can do a Shapiro-Wilk to confirm or deny our beliefs. The Shapiro-Wilk test says the p-value is $2.2e{-}16 \approx 0$. So, we reject our hypothesis that the errors are normally distributed. However, we have about $1,500$ observations, and we can say because of the Central Limit Theorem, the errors will not be an issue. This is because $1,500$ observations will allow for the errors to be approximately normal since we have observed enough instances. We can move on to check for influential observations.

Relevant Inference

Please reference **Appendix Section VII** for code associated to relevant inference. The two 'competing' models that we came across in our variable selection process are determined by including or excluding the `tmin` variable. So, we will consider the model with `tmin` the complete model, and the model without `tmin` the reduced model.

Our `anova` function call tells us that the reduced model is preferred to the complete model. That is, we should use the model **without** `tmin` because the p-value ≈ 0.3 and the test statistic is ≈ 1.05. Our null hypothesis is that the coefficient of `tmin` is 0. The alternative hypothesis is that the coefficient of `tmin` is not 0. So, we fail to reject H_0 and we don't have sufficient evidence to use H_a.

Now we can determine the direction and magnitude of the associations between regressors and the response. Please reference **Appendix Section VIII**. Here is a summary of the results

Regressor	Coefficient	Interpretation
`sweq`	2.1719	Positive association; an increase in 1 unit of `sweq` increases the response by 2.17. This is a pretty substantial impact.
`prec`	0.1820	Positive association; an increase in `prec` by 1 unit increases the response by 0.1820. This isn't a very substantial impact, though we will keep this in the model for the time being.
`tmax`	-0.4506	Negative association; an increase in `tmax` by 1 unit decreases the response by 0.4506. This is a reasonably sized impact (not too large or small).
`seasonNULL`	-3.9976	Negative association; the observation being in `seasonNULL` decreases the response by 3.9976. This is expected and substantial.
`seasonSpring`	3.3402	Positive association; if the observation is in `seasonSpring`, the response increases by 3.3402. This is a significant association.
`seasonWinter`	6.0487	Positive association; if the observation is in `seasonWinter`, the response increases by 6.0487. This is a very substantial association.

0.3 Conclusion & Further Efforts

The reduced linear model–which regresses Snow Depth on Snow-Water-Equivalent, Precipitation, maximum temperature, and season–can provide an idea of how much snow we expect to 'stick' on Vail Pass given observational weather data. This can be helpful because if the temperature is too high and it is still snowing, it can be important for The

Colorado Department of Transportation to save money by not deploying snow plows. On the flip side, it could be helpful to know how much snow to expect from a storm and subsequently how to deploy a necessary number of snow plows.

A potential improvement for future investigation could include associating time within the regression model somehow. It would be helpful to know when it will snow the most. Though time-series regression was not covered in this course, further investigation into the subject could be helpful. To continue on the simple linear regression route, there may be variables that were not represented in the Kaggle data set that may have more influence on snow depth.

Another extension of this project would be using snow depth and weather conditions to predict likelihood of an avalanche reaching the road.

0.4 Appendix

0.5 Section I

```r
# initializing necessary libraries that are used throughout the report
library(ggplot2)
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
#reading in file
df_raw <- read.csv("C:/Users/poore/Downloads/training_data.csv")

#filtering out days that have a reported snow depth greater than 0.
df <- df_raw |> subset(SNWD.I.1..in. > 0)

#renaming variables
names(df) <- c("x","date","sweq","prec","tobs","tmax","tmin","tavg","snwd","batt1","batt2","ba

#transforming date into season
df$date <- as.Date(df$date)

#read in original format and output season (Spring,Summer,Fall,Winter)
get_season <- function(date) {
  month <- as.numeric(format(date, "%m"))
  day <- as.numeric(format(date, "%d"))
  if ((month == 12 && day >= 21) || (month <= 2) || (month == 3 && day < 20)) {
    return("Winter")
  } else if ((month == 3 && day >= 20) || (month == 6 && day < 21)) {
    return("Spring")
  } else if ((month == 6 && day >= 21) || (month == 9 && day < 22)) {
    return("Summer")
  } else if ((month == 9 && day >= 22) || (month == 12 && day < 21)) {
    return("Fall")}}

#use previous function and apply it to the dataframe
df <- df |>
  mutate(season = as.character(sapply(date, get_season)))
```

```r
# Load necessary libraries
options(repos = c(CRAN = "https://cran.rstudio.com/"))
install.packages("gridExtra")
```

```
Installing package into 'C:/Users/poore/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
```

```
package 'gridExtra' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
    C:\Users\poore\AppData\Local\Temp\RtmpgRtMa7\downloaded_packages
```

```r
library(gridExtra)
```

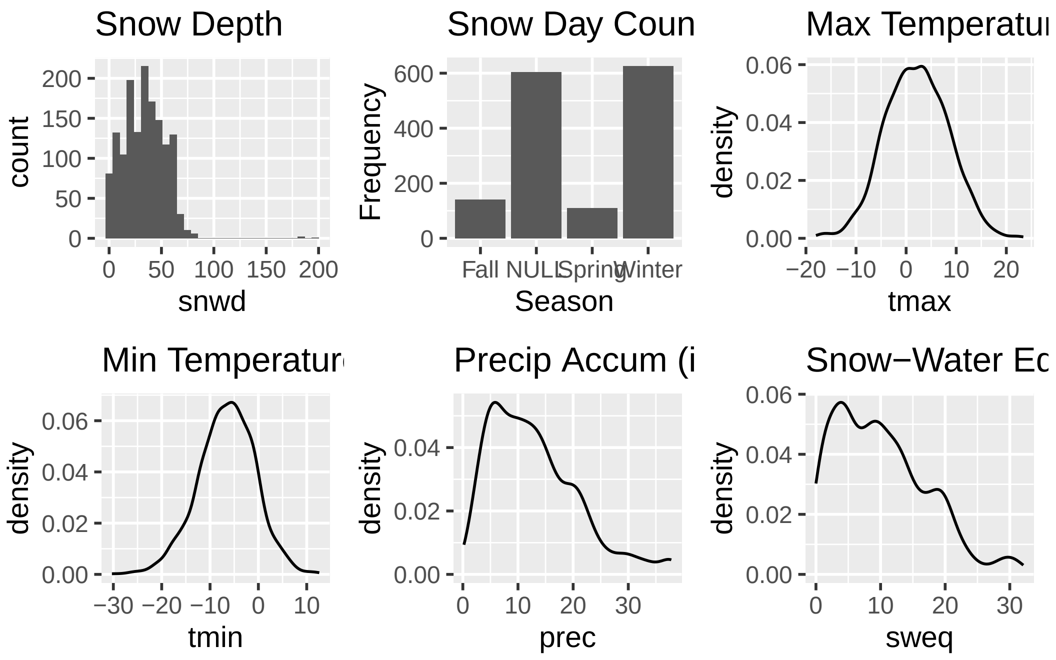
```
Warning: package 'gridExtra' was built under R version 4.4.2


Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':

    combine
```

```r
df <- df[ , !sapply(df, is.null)]

# Define each plot
p1 <- ggplot(df, aes(x = snwd)) +
      geom_histogram() +
      labs(title = "Snow Depth")

p2 <- ggplot(df, aes(x = season)) +
      geom_histogram(stat="count") +
      labs(title = "Snow Day Count",
           x = "Season",
           y = "Frequency")
```

```
Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
`binwidth`, `bins`, and `pad`
```

```r
p3 <- ggplot(df, aes(x = tmax)) +
      geom_density() +
      labs(title="Max Temperature Daily")

p4 <- ggplot(df, aes(x = tmin)) +
      geom_density() +
      labs(title="Min Temperature Daily")

p5 <- ggplot(df, aes(x = prec)) +
      geom_density() +
      labs(title="Precip Accum (in)")

p6 <- ggplot(df, aes(x = sweq)) +
      geom_density() +
     labs(title="Snow-Water Equiv (in)")

grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 3, nrow = 2)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



0.6 Section II

```
#snow water equivalent
print("Snow Water Equivalent")
```

```
[1] "Snow Water Equivalent"
```

```
summary(df$sweq)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00    4.30    9.20   10.14   14.50   32.10 
```

```
print("-------------------------------------------")
```

```
[1] "-------------------------------------------"
```

```
#precipitation accumulation
print("Precipitation")
```

```
[1] "Precipitation"
```

```
summary(df$prec)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    0.2     6.4    11.6    12.9    17.6    37.8 
```

```
print("-------------------------------------------")
```

```
[1] "-------------------------------------------"
```

```
#min temperature
print("Minimum Temperature")
```

```
[1] "Minimum Temperature"
```

```
summary(df$tmin)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-30.300 -10.300  -6.300  -6.598  -2.500  12.600 
```

```
print("-------------------------------------------")
```

```
[1] "-------------------------------------------"
```

```
#max temperature
print("Maximum Temperature")
```

```
[1] "Maximum Temperature"
```

```
summary(df$tmax)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-18.000  -2.300   2.100   2.143   6.500  23.400 
```

```
print("-------------------------------------------")
```

```
[1] "-------------------------------------------"
```

```
#snow depth
print("Snow Depth")
```

```
[1] "Snow Depth"
```

```
summary(df$snwd)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00   19.00   34.00   33.45   48.00  198.00 
```

```
print("-------------------------------------------")
```

```
[1] "-------------------------------------------"
```

```
print("Season")
```

```
[1] "Season"
```

```
#season
table(df$season)
```

```

  Fall   NULL Spring Winter 
   140    604    110    625 
```

0.7 Section III

Check for Collinearity

```
cor(df[, c("sweq", "prec", "tmin","tmax")])
```

```
          sweq      prec      tmin      tmax
sweq 1.0000000 0.7767921 0.1056549 0.1743741
prec 0.7767921 1.0000000 0.4042374 0.4636550
tmin 0.1056549 0.4042374 1.0000000 0.8994867
tmax 0.1743741 0.4636550 0.8994867 1.0000000
```

0.8 Section IV

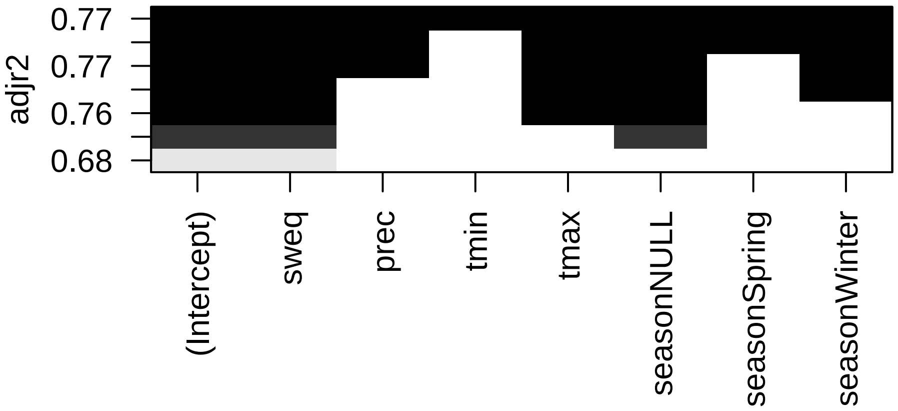
```
if(!require(caret, quietly = TRUE)) {
  install.packages("caret", repos = "https://cran.rstudio.com/")
  library(caret)
}
```

```
Warning: package 'caret' was built under R version 4.4.2
```

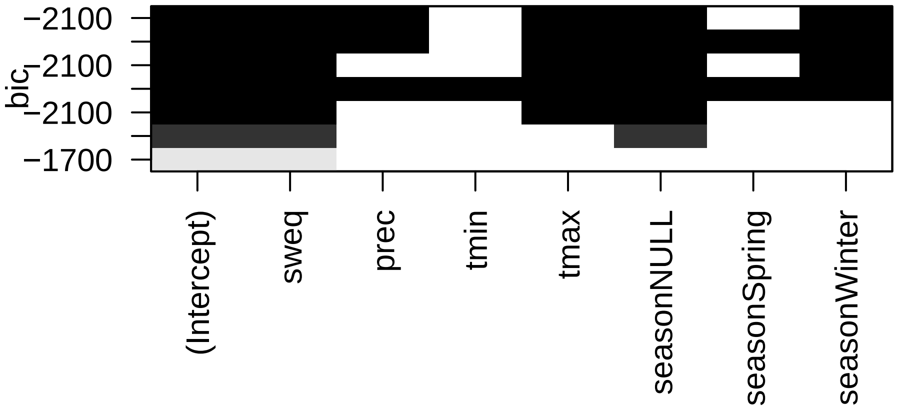
```
if(!require(leaps, quietly = TRUE)) {
  install.packages("leaps", repos = "https://cran.rstudio.com/")
  library(leaps)
}
```

```
Warning: package 'leaps' was built under R version 4.4.2
```

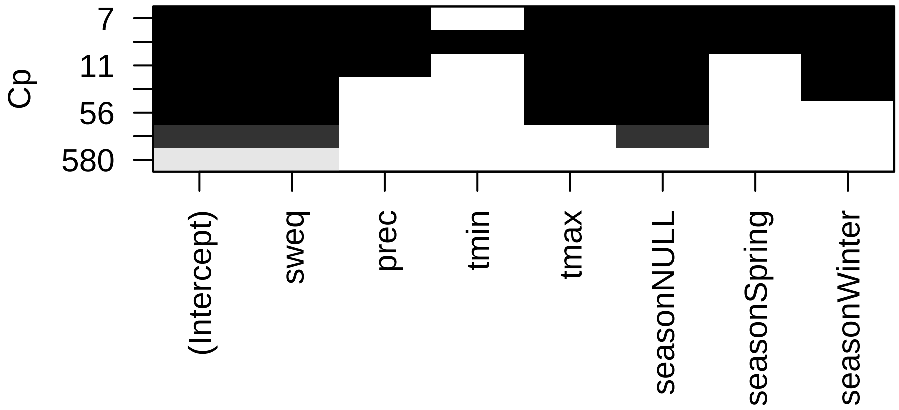
```
complete_lm <- lm(snwd ~ sweq + prec + tmin + tmax + season,data = df)

rs <- regsubsets(snwd ~ sweq + prec + tmin + tmax +season,data=df)



# R^2_a
plot(rs,scale="adjr2")
```



```
plot(rs,scale="bic")
```



```r
plot(rs,scale="Cp")
```

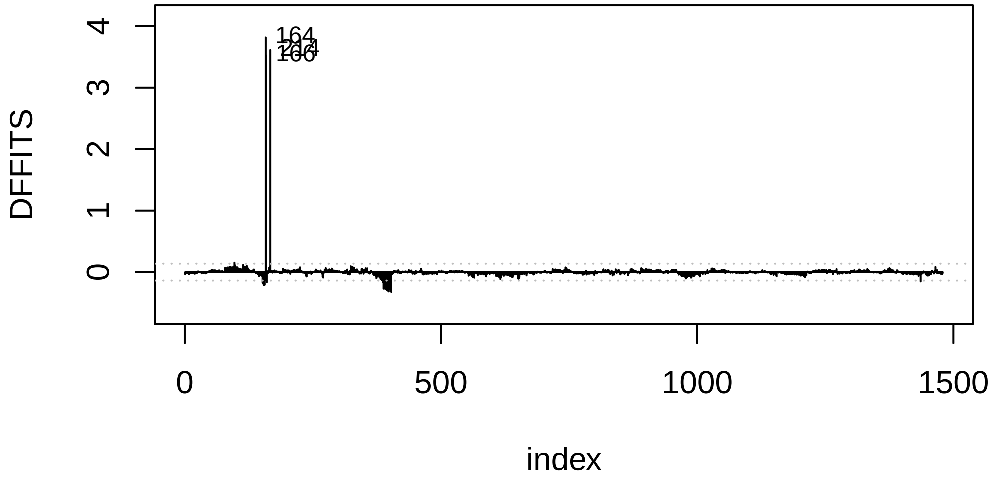


0.9 Section V

```r
library(api2lm)
```

```
Warning: package 'api2lm' was built under R version 4.4.2
```

```r
rlm <- lm(snwd ~ sweq + prec + tmax + season, data=df)
dffits_plot(rlm, id_n = 3)
```



```
options(digits=4)
df[c(164,166,214),c("season","snwd","prec","tmin","tmax","sweq")]
```

```
    season snwd prec tmin tmax sweq
197   NULL    2 31.3 11.6 21.9    0
201   NULL    5 31.3 12.6 23.4    0
344   Fall   26  8.1 -7.9  3.0    6
```

```
rlm <- lm(snwd ~ sweq + prec + tmax + season, data=df)
df2 <- df[-c(164,166)]
rlm2 <- lm(snwd ~ sweq + prec + tmax + season, data=df2)
coef_compare(rlm,rlm2)
```

```
             Model 1 Model 2 pct_change
(Intercept)    8.870   8.870      0.000
Std. Error     0.837   0.837      0.000

sweq          2.1719  2.1719     0.0000
Std. Error    0.0594  0.0594     0.0000

prec           0.182   0.182      0.000
Std. Error     0.059   0.059      0.000
```

```
tmax         -0.4506 -0.4506     0.0000
Std. Error    0.0502  0.0502     0.0000

seasonNULL    -3.998  -3.998      0.000
Std. Error     0.965   0.965      0.000

seasonSpring    3.34    3.34       0.00
Std. Error      1.32    1.32       0.00

seasonWinter   6.049   6.049      0.000
Std. Error     0.918   0.918      0.000
```

```r
df <- df[-c(164,166)]
rlm <- rlm2
```

0.10 Section VI

```r
#installing necessary packages
if(!require(lmtest, quietly = TRUE)) {
  install.packages("lmtest",
                   repos = "https://cran.rstudio.com/")
  library(lmtest)
}
```

```
Warning: package 'lmtest' was built under R version 4.4.2
```

```
Warning: package 'zoo' was built under R version 4.4.2
```

```

Attaching package: 'zoo'
```

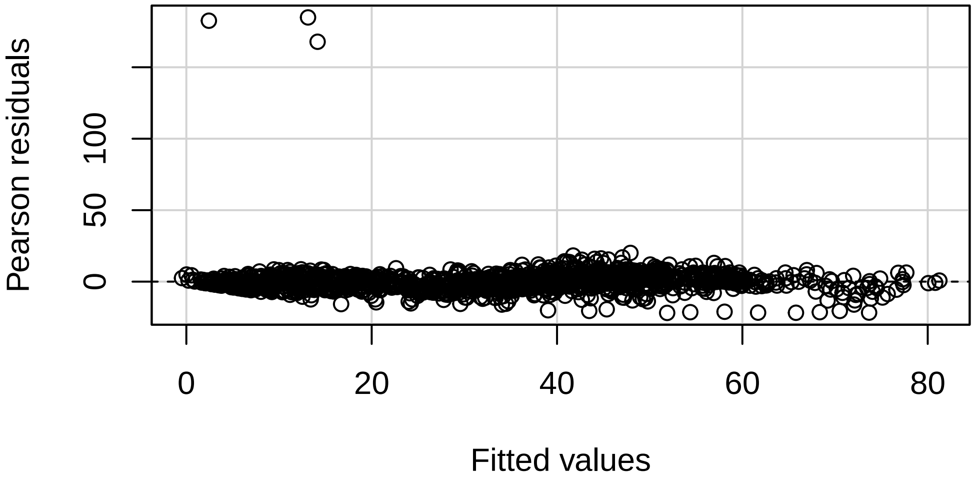
```
The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```
if(!require(car, quietly = TRUE)) {
  install.packages("car",
                   repos = "https://cran.rstudio.com/")
  library(car)
}
```

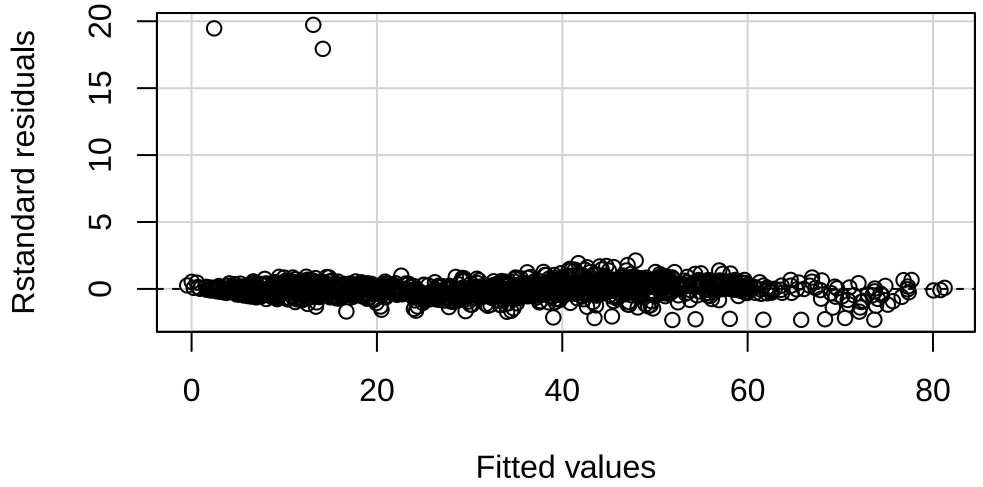
```
Attaching package: 'car'
```

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

    recode
```

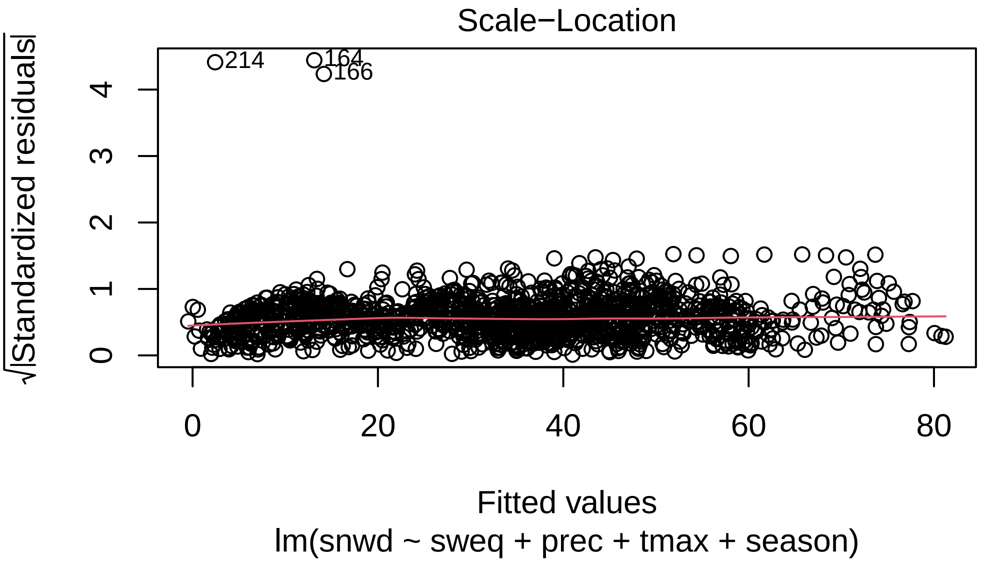
```
#fitting our model from section IV.
residualPlot(rlm2,quadratic=FALSE)
```



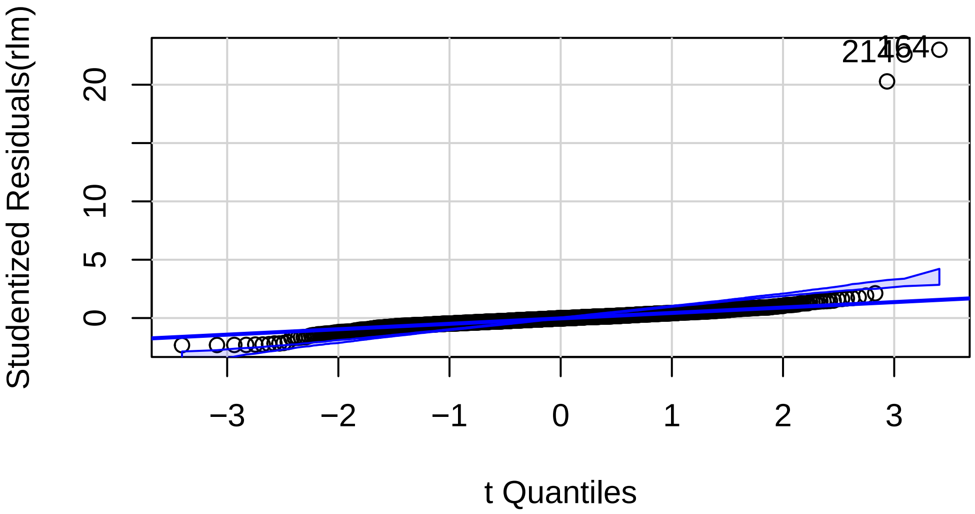
```
residualPlot(rlm,type="rstandard",quadratic=FALSE)
```



```
plot(rlm, which=3)
```



```
qqPlot(rlm)
```



```
164 214
158 167
```

```
shapiro.test(rstandard(rlm))
```

```


    Shapiro-Wilk normality test

data:  rstandard(rlm)
W = 0.4, p-value <2e-16
```

0.11 Section VII

```
anova(rlm,complete_lm)
```

```
Analysis of Variance Table

Model 1: snwd ~ sweq + prec + tmax + season
Model 2: snwd ~ sweq + prec + tmin + tmax + season
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1   1472 132767
2   1471 132672  1      94.6 1.05   0.31
```

0.12 Section VIII

```
summary(rlm)
```

```

Call:
lm(formula = snwd ~ sweq + prec + tmax + season, data = df2)

Residuals:
   Min     1Q Median     3Q    Max
-21.88  -3.12  -0.25   2.56 184.87

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    8.8700     0.8373   10.59  < 2e-16 ***
sweq           2.1719     0.0594   36.58  < 2e-16 ***
prec           0.1820     0.0590    3.08   0.0021 **
tmax          -0.4506     0.0502   -8.97  < 2e-16 ***
seasonNULL    -3.9976     0.9651   -4.14  3.6e-05 ***
seasonSpring   3.3402     1.3184    2.53   0.0114 *
seasonWinter   6.0487     0.9175    6.59  6.0e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.5 on 1472 degrees of freedom
Multiple R-squared:  0.77,  Adjusted R-squared:  0.769
F-statistic:  821 on 6 and 1472 DF,  p-value: <2e-16
```