HAB Analysis

Nikolas Polsinelli

2025-01-09

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library("tidymodels")
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.2.0 --
## v broom        1.0.7     v rsample      1.2.1
## v dials        1.3.0     v tune         1.2.1
## v infer        1.0.7     v workflows    1.1.4
## v modeldata    1.4.0     v workflowsets 1.1.0
## v parsnip      1.2.1     v yardstick    1.3.1
## v recipes      1.1.0
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library("ggforce")
library("ggplot2")
alldata<-read_csv("allData.csv")%>%
  as_tibble()
```

```
## Rows: 100628 Columns: 9
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: ","
## dbl (9): Time, Temperature, Pressure, X Axis Acceleration, Y Axis Accelerati...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
alldata
```

```
## # A tibble: 100,628 x 9
##     Time Temperature Pressure `X Axis Acceleration` `Y Axis Acceleration`
##    <dbl>       <dbl>    <dbl>                 <dbl>                 <dbl>
##  1  2010        18.1     997.                  0.06                     0
##  2  3069        18.1     997.                  0.06                     0
##  3  4128        18.1     997.                  0.06                     0
##  4  5187        18.1     997.                  0.06                     0
##  5  6246        18.1     997.                  0.06                     0
##  6  7305        18.1     997.                  0.06                     0
##  7  8364        18.1     997.                  0                        0
##  8  9423        18.1     997.                  0.06                     0
##  9 10482        18.1     997.                  0.06                     0
## 10 11541        18.1     997.                  0.06                     0
## # i 100,618 more rows
## # i 4 more variables: `Z Axis Acceleration` <dbl>, CO2 <dbl>, TVOC <dbl>,
## #   DataFileNumber <dbl>
```

```
alldata2<-read_csv("allData2.csv")%>%
  as_tibble()
```

```
## Rows: 89284 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (9): Time, Temperature, Pressure, X Axis Acceleration, Y Axis Accelerati...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
alldata2
```

```
## # A tibble: 89,284 x 9
##        Time Temperature Pressure `X Axis Acceleration` `Y Axis Acceleration`
##       <dbl>       <dbl>    <dbl>                 <dbl>                 <dbl>
##  1 87846916       -7.33     170.                  0.06                  0.06
##  2 87845845       -7.33     170.                  0.06                  0.06
##  3 87849067       -7.34     170.                  0.06                  0.06
##  4 87844774       -7.33     170.                  0.06                  0
##  5 87843702       -7.32     170.                  0.06                  0.06
##  6 87842630       -7.32     170.                  0.06                  0.06
##  7 87839417       -7.3      170.                  0                     0.06
##  8 87841559       -7.32     170.                  0.06                  0.06
##  9 87840488       -7.32     170.                  0.06                  0.06
## 10 87838346       -7.3      170.                  0.06                  0.06
## # i 89,274 more rows
## # i 4 more variables: `Z Axis Acceleration` <dbl>, CO2 <dbl>, TVOC <dbl>,
## #   DataFileNumber <dbl>
```

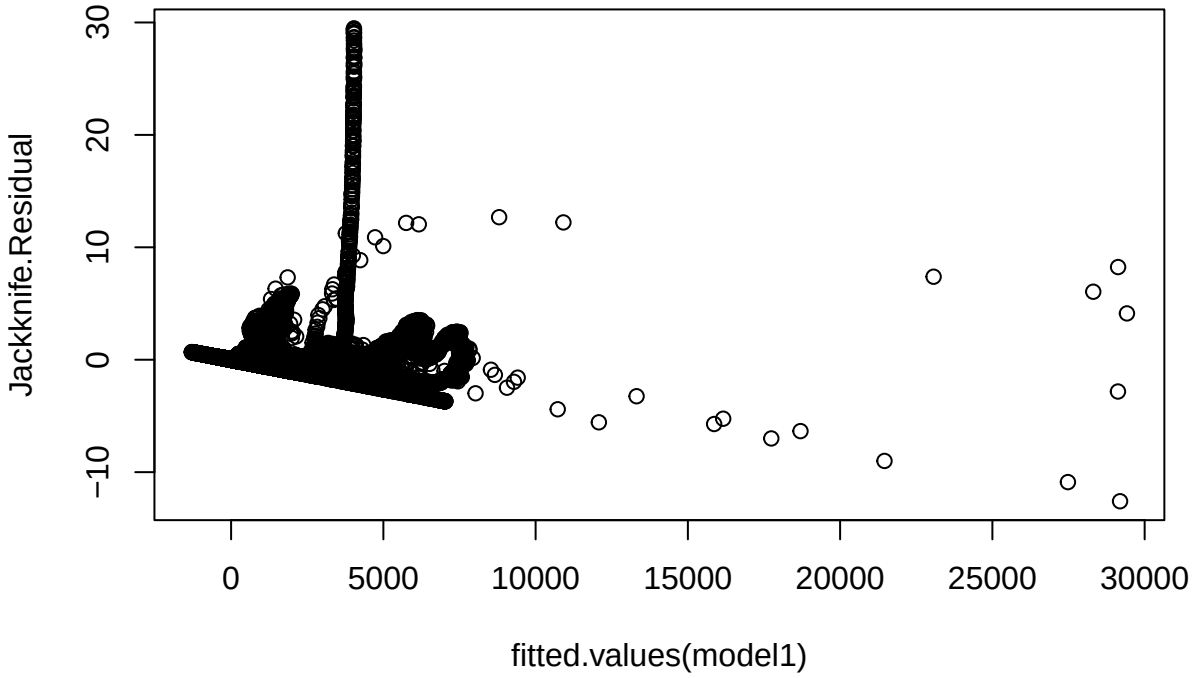
```
model1<-lm(TVOC~CO2+Pressure+Temperature+Time,data=alldata2)
summary(model1)
```

```
##
## Call:
## lm(formula = TVOC ~ CO2 + Pressure + Temperature + Time, data = alldata2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -23768   -376    -62    287  55861
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.468e+03  7.782e+01   83.12   <2e-16 ***
## CO2          4.961e-01  4.323e-03  114.76   <2e-16 ***
## Pressure    -8.948e+00  9.342e-02  -95.78   <2e-16 ***
## Temperature  9.810e+01  1.045e+00   93.84   <2e-16 ***
## Time        -4.427e-06  3.513e-07  -12.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1905 on 89279 degrees of freedom
## Multiple R-squared:  0.3201, Adjusted R-squared:  0.3201
## F-statistic: 1.051e+04 on 4 and 89279 DF,  p-value: < 2.2e-16
```

```
residerror<-sqrt(deviance(model1)/df.residual(model1))
```

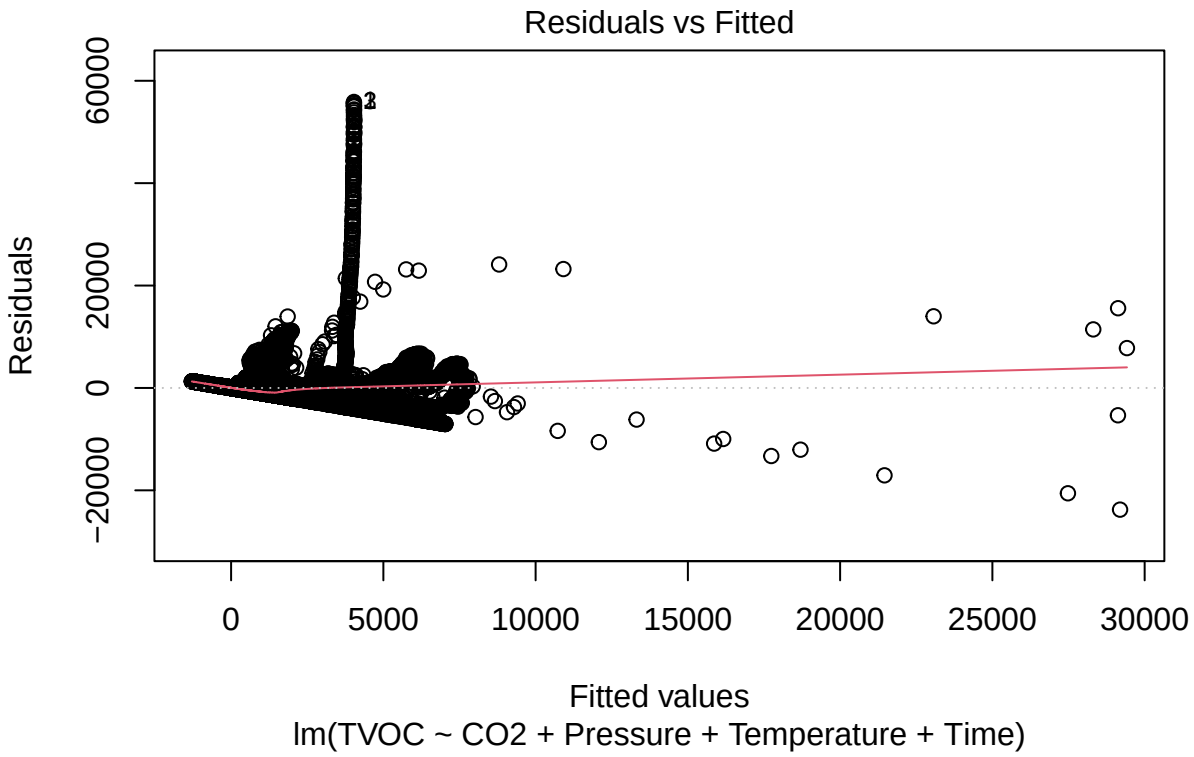
#{r} #ggplot(alldata2)+ # geom_point(aes(x=fitted.values(model),y=resid(model)))+ # geom_hline(yintercept=2*residerror,color="blue")+ # geom_hline(yintercept=-2*residerror,color="blue")+ # geom_hline(yintercept=3*residerror,color="red")+ # geom_hline(yintercept=-3*residerror,color="red") This residual plot is exceedingly problematic. Clear patterns, lots of outliers. This indicates a need to adjust the model. #{r} #round(sort(cooks.distance(model)),4) #

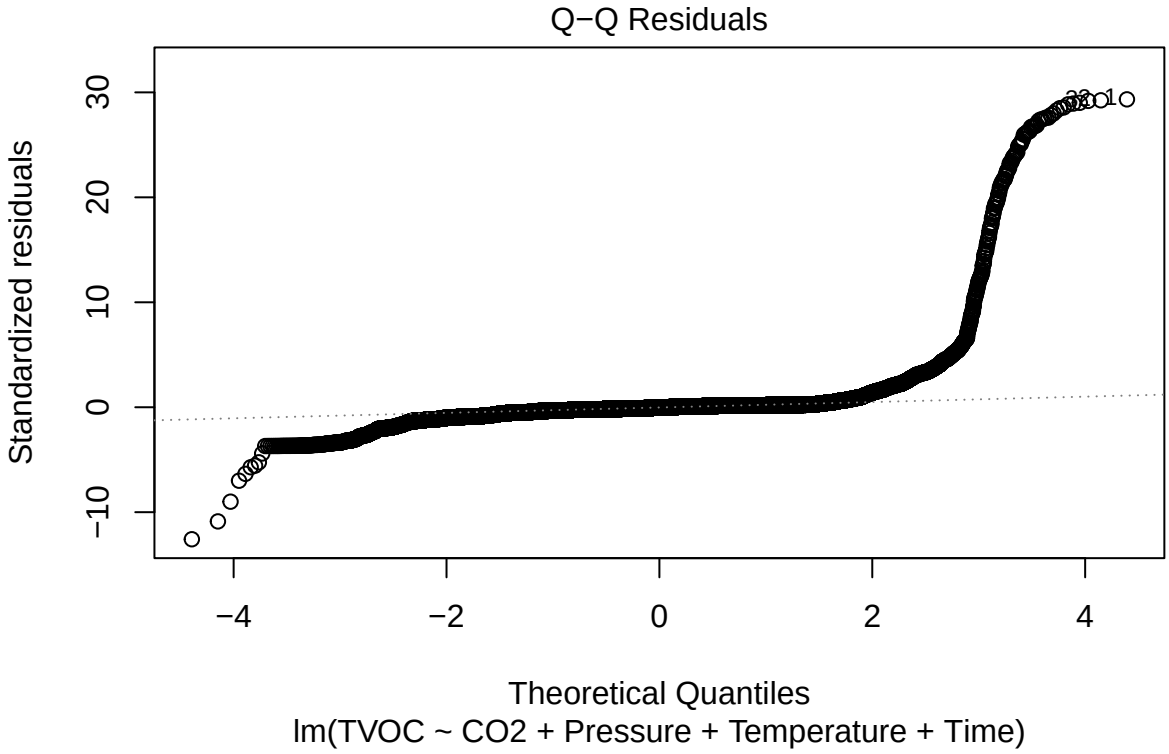
```
Jackknife.Residual<-rstudent(model1)
plot(fitted.values(model1),Jackknife.Residual)
```

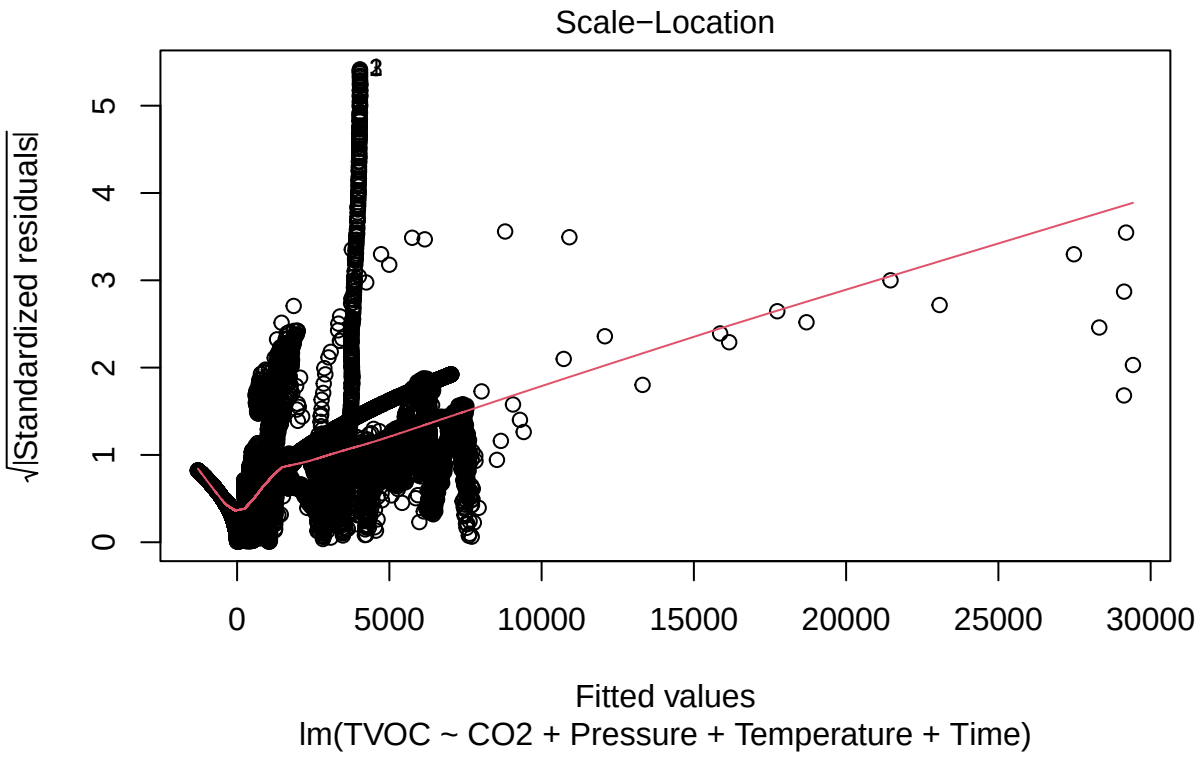


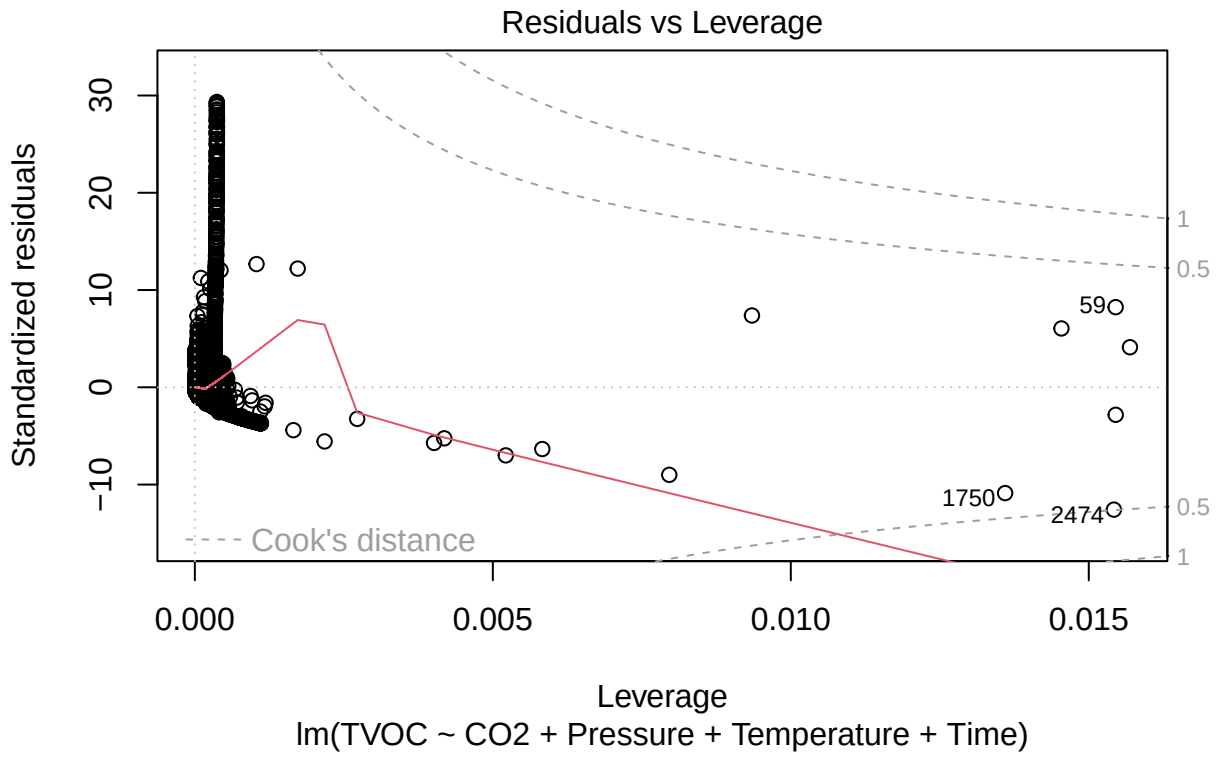

```
#ggplot(alldata2)+
#  geom_point(aes(x=fitted.values(model),y=Jackknife.Residual))+
 # geom_hline(yintercept=qt(df=95,.95),color="blue")+
 # geom_hline(yintercept=-qt(df=95,.95),color="blue")
```

```
plot(model1)
```


Residuals vs Fitted
Residuals
60000
20000
0
−20000
0
5000
10000
15000
20000
25000
30000
Fitted values
lm(TVOC ~ CO2 + Pressure + Temperature + Time)


Q−Q Residuals
Standardized residuals
30
20
10
0
−10
−4
−2
0
2
4
Theoretical Quantiles
lm(TVOC ~ CO2 + Pressure + Temperature + Time)


Scale−Location
√|Standardized residuals|
0
1
2
3
4
5
0
5000
10000
15000
20000
25000
30000
Fitted values
lm(TVOC ~ CO2 + Pressure + Temperature + Time)



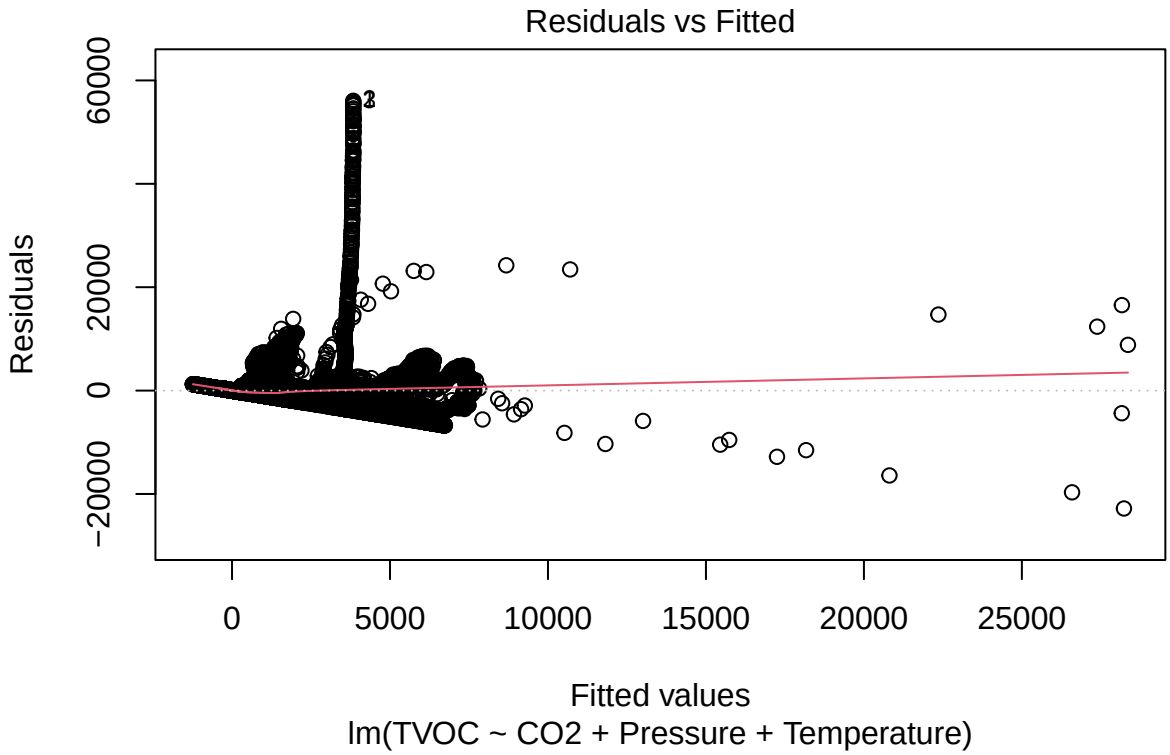

```
model2<-lm(TVOC~CO2+Pressure+Temperature,data=alldata2)
summary(model2)
```

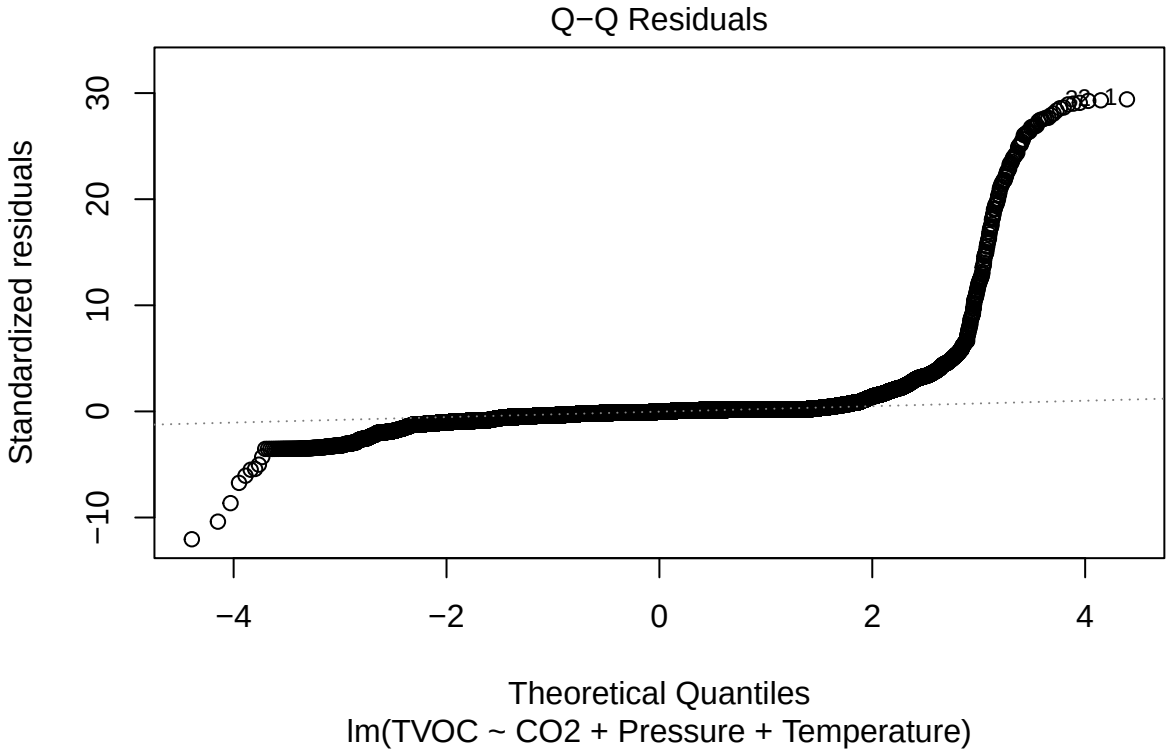
```
##
## Call:
## lm(formula = TVOC ~ CO2 + Pressure + Temperature, data = alldata2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -22806   -371    -57    291  56058
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.735e+03  5.167e+01  110.99   <2e-16 ***
## CO2          4.756e-01  4.009e-03  118.64   <2e-16 ***
## Pressure    -8.238e+00  7.459e-02 -110.44   <2e-16 ***
## Temperature  9.326e+01  9.732e-01   95.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1906 on 89280 degrees of freedom
## Multiple R-squared:  0.3189, Adjusted R-squared:  0.3189
## F-statistic: 1.394e+04 on 3 and 89280 DF,  p-value: < 2.2e-16
```

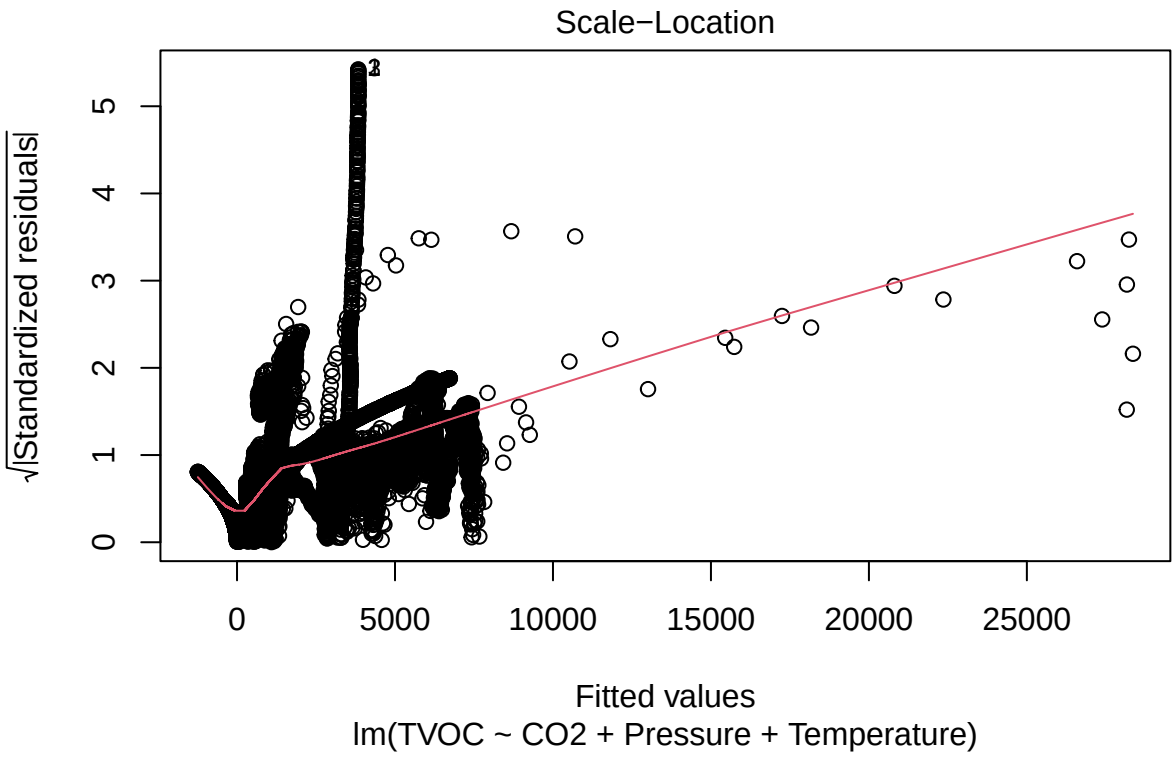
```
residerror2<-sqrt(deviance(model2)/df.residual(model2))
```

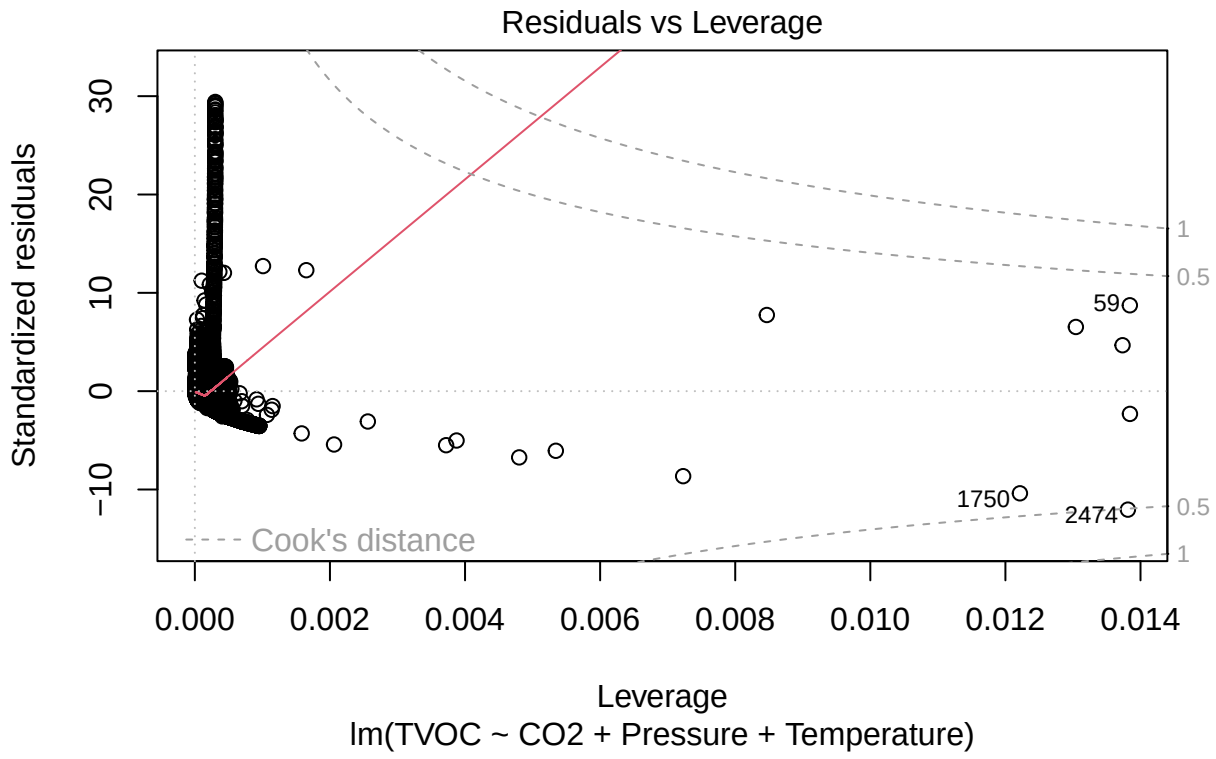
```
#ggplot(alldata2)+
 # geom_point(aes(x=fitted.values(model2),y=resid(model2)))+
 # geom_hline(yintercept=2*residerror,color="blue")+
 # geom_hline(yintercept=-2*residerror,color="blue")+
 # geom_hline(yintercept=3*residerror,color="red")+
 # geom_hline(yintercept=-3*residerror,color="red")
```

```
plot(model2)
```




Q−Q Residuals
Standardized residuals
30
20
10
0
−10
−4
−2
0
2
4
Theoretical Quantiles
lm(TVOC ~ CO2 + Pressure + Temperature)


Scale−Location
√|Standardized residuals|
0
1
2
3
4
5
0
5000
10000
15000
20000
25000
Fitted values
lm(TVOC ~ CO2 + Pressure + Temperature)




```
library("moments")
skewness(Jackknife.Residual)
```

```
## [1] 15.06723
```

```
kurtosis(Jackknife.Residual)
```

```
## [1] 351.7435
```

```
model3<-lm(TVOC~Pressure+Temperature+Time,data=alldata2)
summary(model3)
```

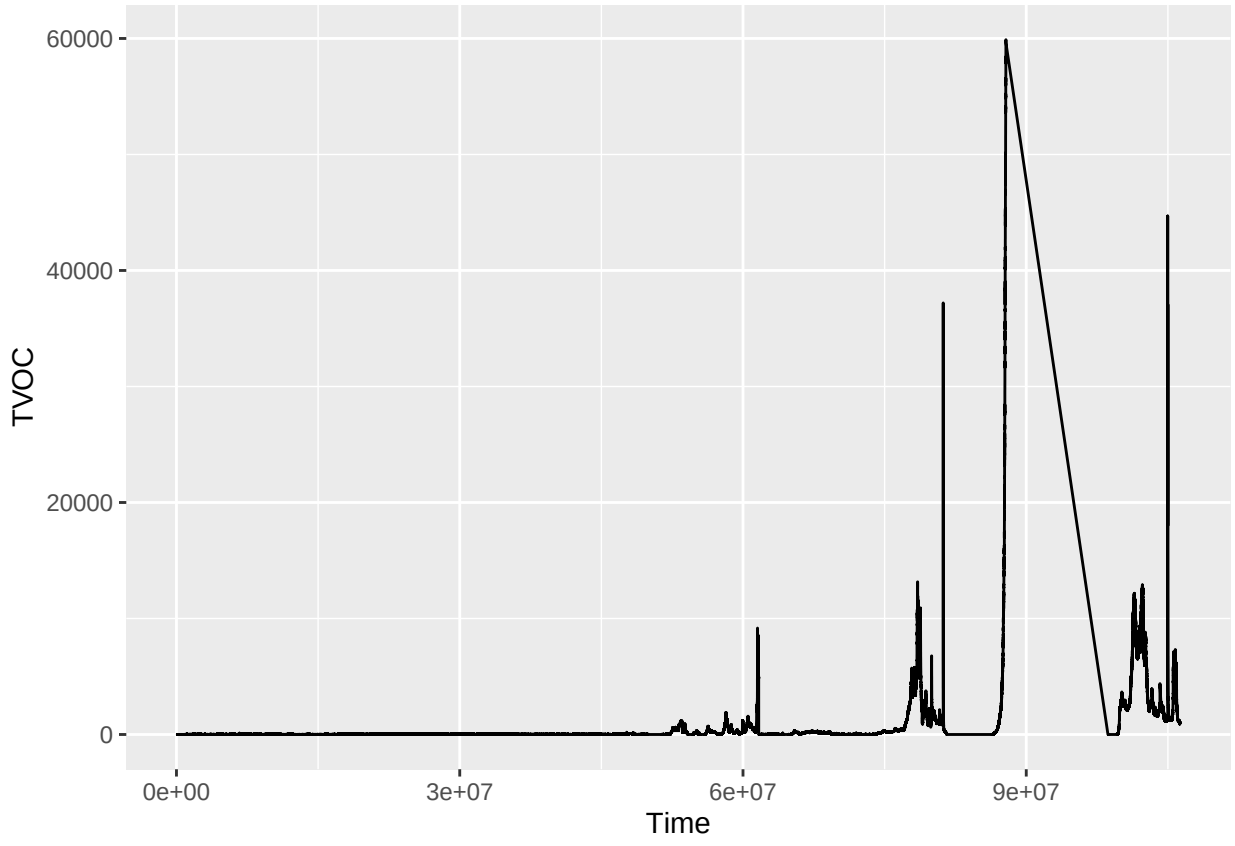
```
##
## Call:
## lm(formula = TVOC ~ Pressure + Temperature + Time, data = alldata2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8489   -706      7    341  55245
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.196e+03  8.332e+01   74.36   <2e-16 ***
## Pressure    -9.462e+00  9.995e-02  -94.66   <2e-16 ***
```

```
## Temperature  1.196e+02  1.102e+00  108.60   <2e-16 ***
## Time         1.075e-05  3.487e-07   30.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2040 on 89280 degrees of freedom
## Multiple R-squared:  0.2199, Adjusted R-squared:  0.2198
## F-statistic:  8387 on 3 and 89280 DF,  p-value: < 2.2e-16
```

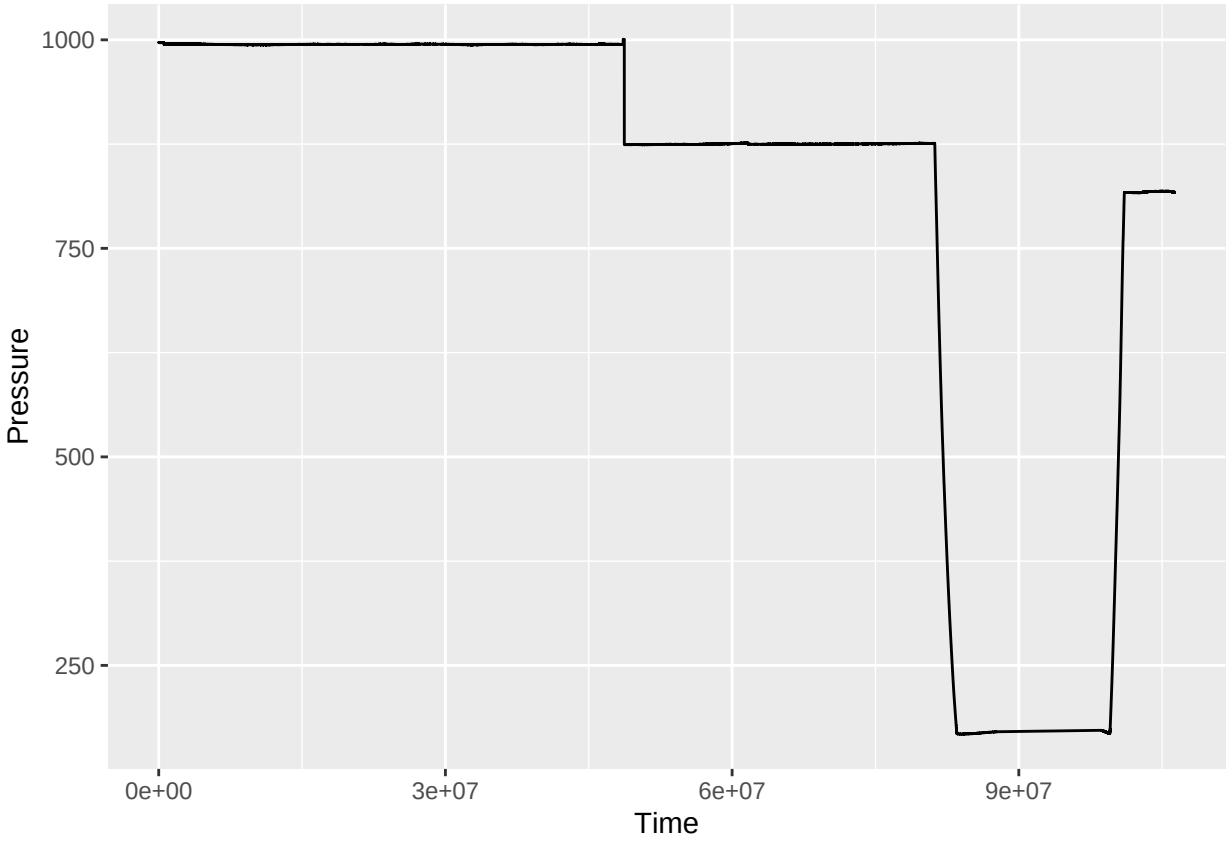
```
residerror3<-sqrt(deviance(model3)/df.residual(model3))
```

```
#ggplot(alldata2)+
#  geom_point(aes(x=fitted.values(model3),y=resid(model3)))+
 # geom_hline(yintercept=2*residerror,color="blue")+
 # geom_hline(yintercept=-2*residerror,color="blue")+
 # geom_hline(yintercept=3*residerror,color="red")+
 # geom_hline(yintercept=-3*residerror,color="red")
```

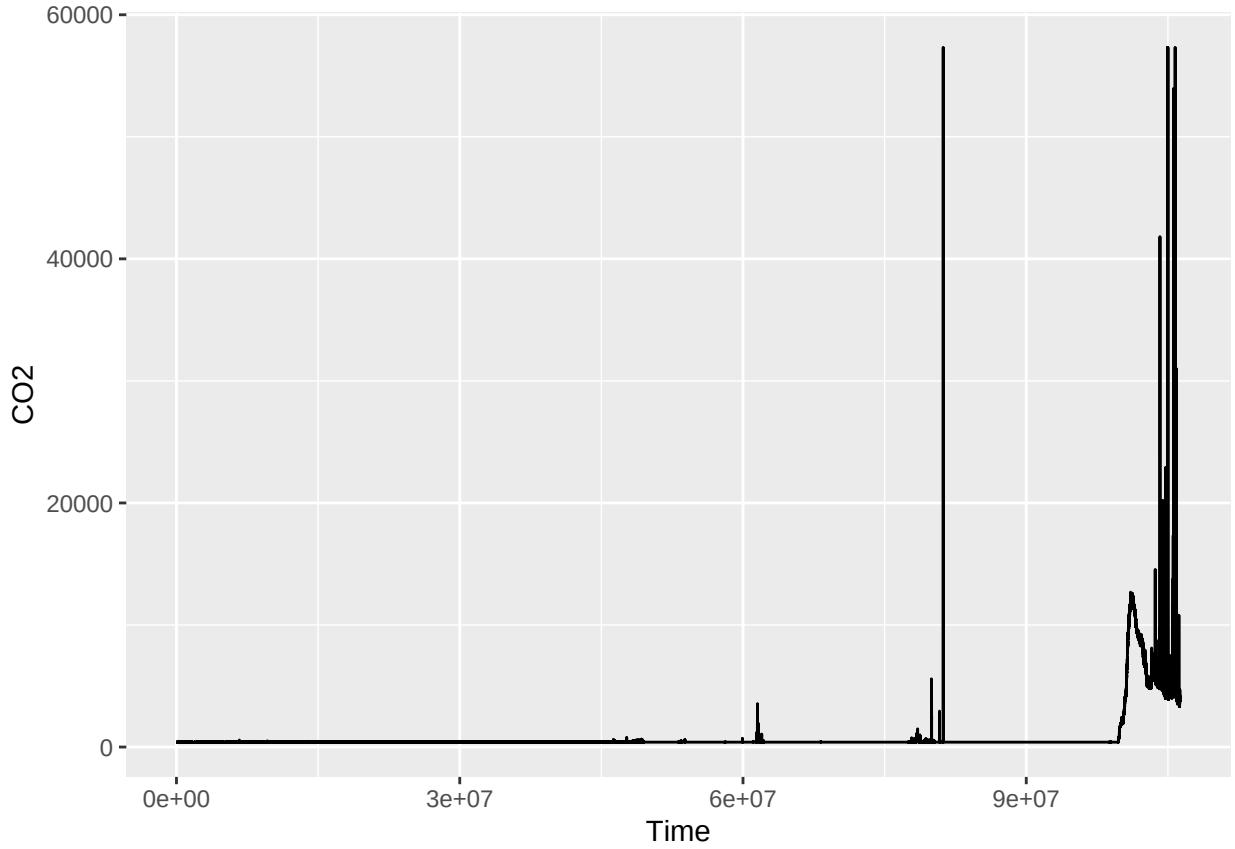
```
ggplot(alldata2, aes(x = Time, y = TVOC)) +
  geom_line()
```



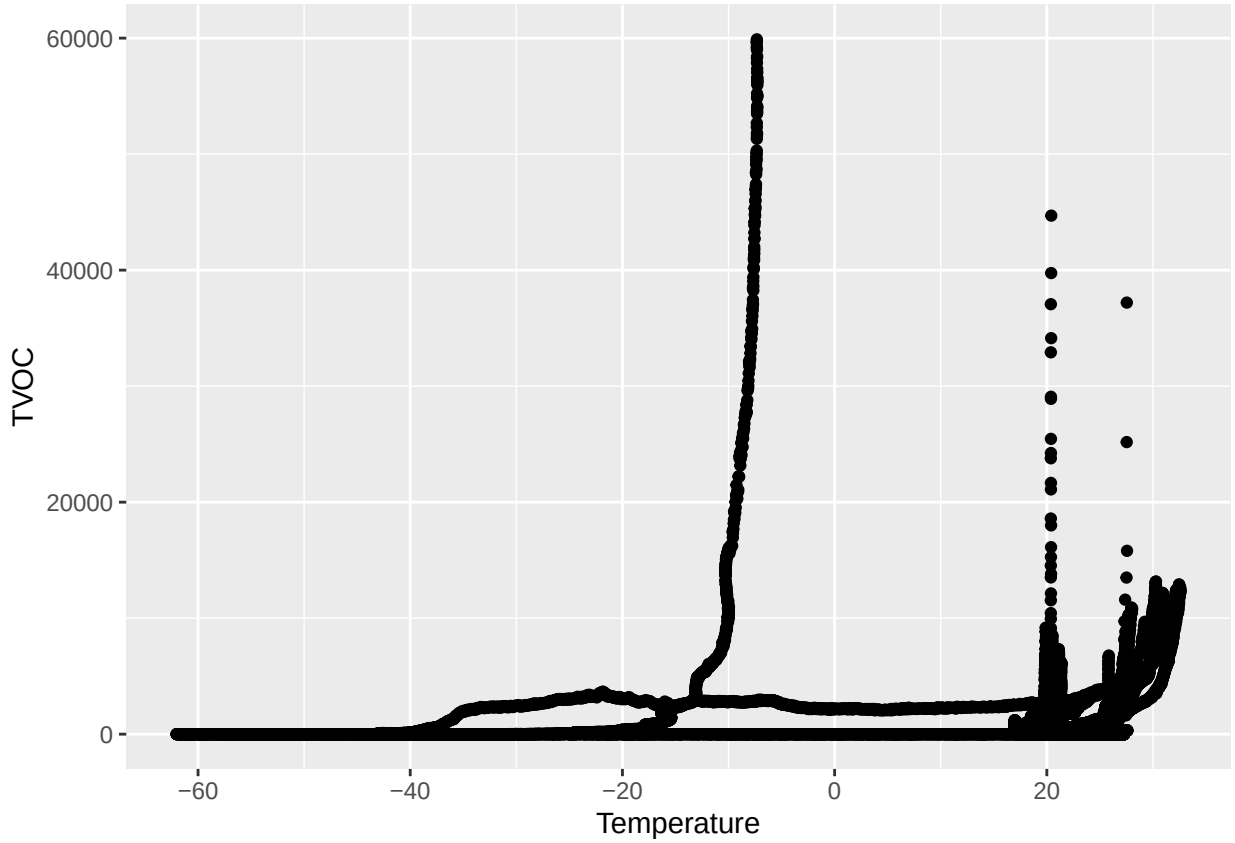

```
ggplot(alldata2, aes(x=Time, y=Pressure))+geom_line()
```



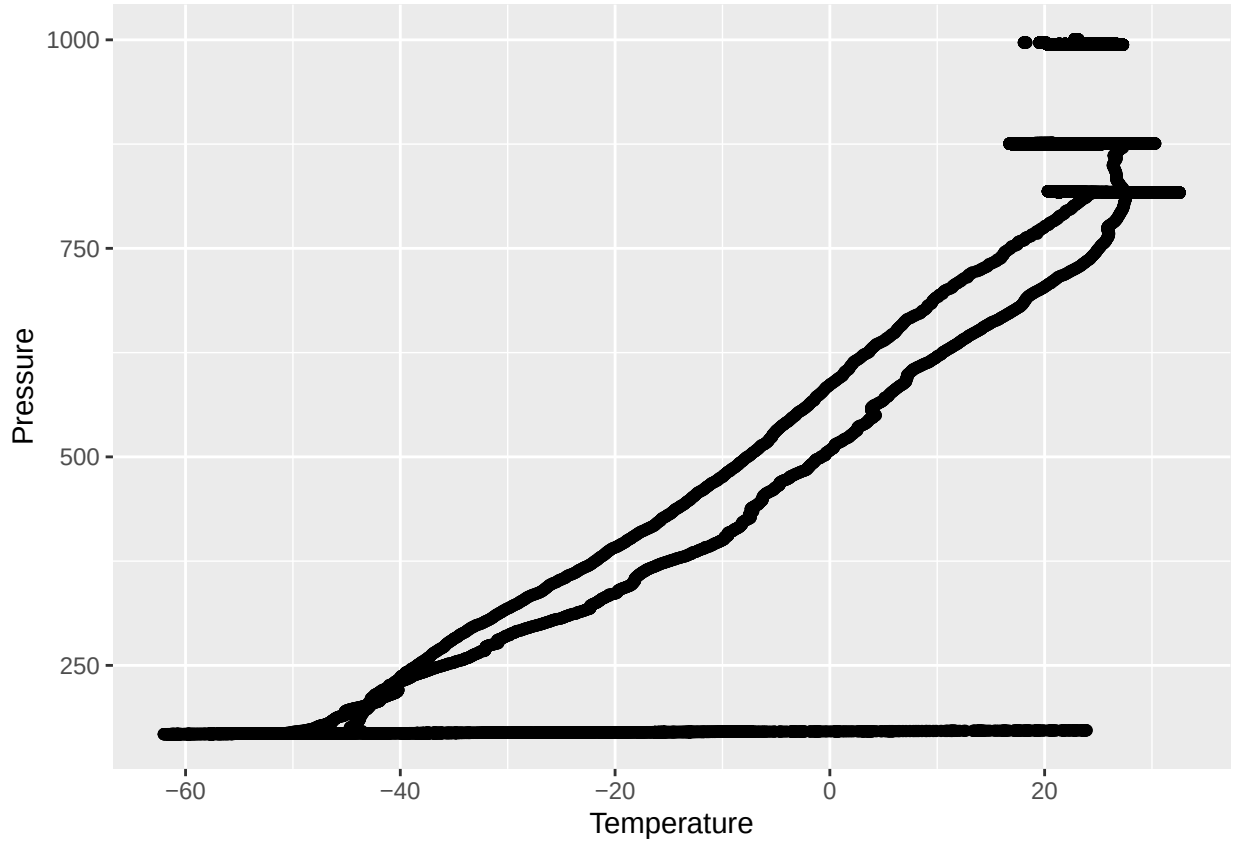

```
ggplot(alldata2, aes(x=Time, y=CO2))+geom_line()
```



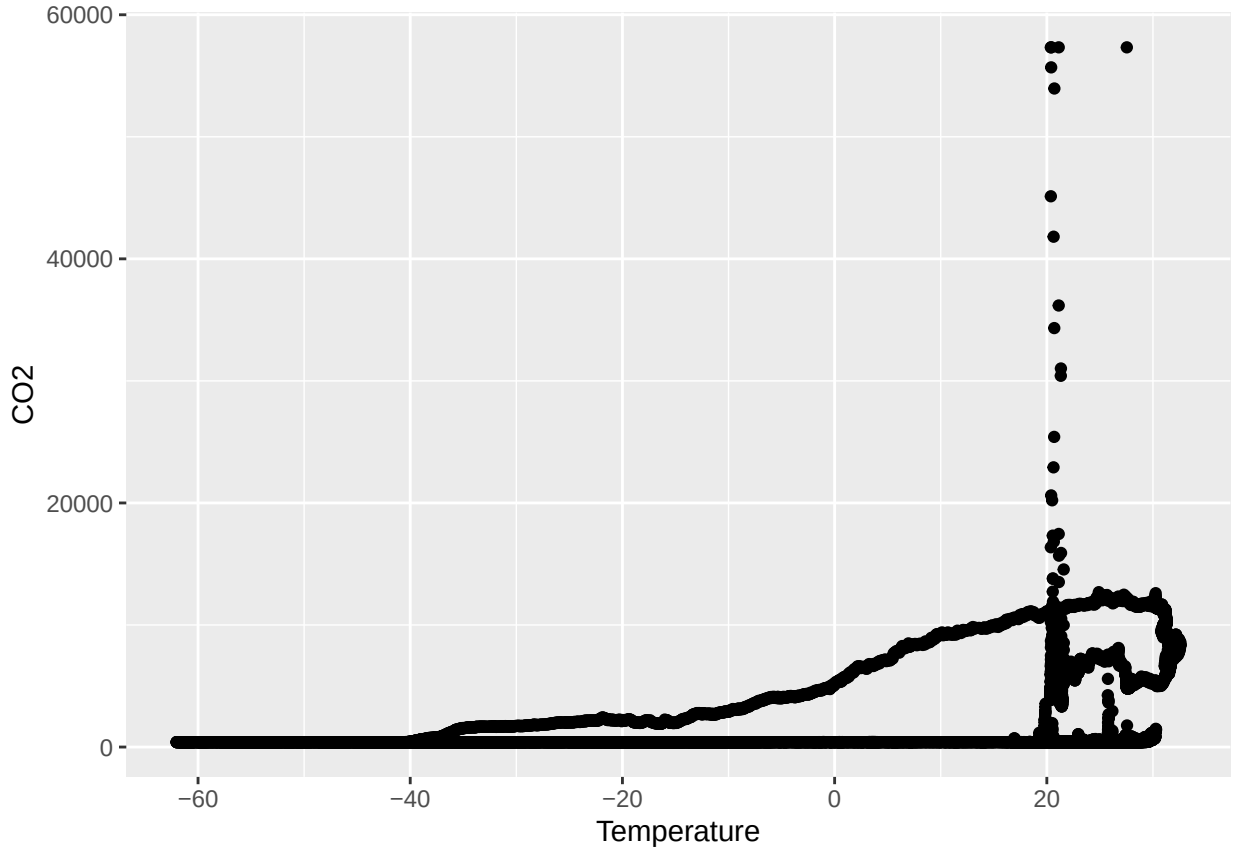

```
ggplot(alldata2, aes(x = Temperature, y = TVOC)) +
  geom_point()
```




```
ggplot(alldata2, aes(x=Temperature, y=Pressure))+geom_point()
```




```
ggplot(alldata2, aes(x=Temperature, y=CO2))+geom_point()
```




```
hab_split<-initial_split(alldata,prop=.75)
train<-training(hab_split)
test<-testing(hab_split)
train
```

```
## # A tibble: 75,471 x 9
##         Time Temperature Pressure `X Axis Acceleration` `Y Axis Acceleration`
##        <dbl>       <dbl>    <dbl>                 <dbl>                 <dbl>
##  1   3519357        26.6   995.                   0.06                  0
##  2 106168295        21.2   817.                   0.06                 -0.62
##  3   1933704        25.4   995.                   0.06                  0
##  4  23868476        22.3   995.                   0.06                  0
##  5  37221961        20.9   995.                   0.06                  0
##  6  20415284        22.1   994.                   0.06                  0
##  7   4125546        26.5   995.                   0.06                  0
##  8  62312421        24.1   875.                   0.06                  0.06
##  9 107768147        -0.5    -0.16                 NA                    NA
## 10  69142233        23.0   875.                   0.06                  0.06
## # i 75,461 more rows
## # i 4 more variables: `Z Axis Acceleration` <dbl>, CO2 <dbl>, TVOC <dbl>,
## #   DataFileNumber <dbl>
```

```
test
```

```
## # A tibble: 25,157 x 9
```

```
##     Time Temperature Pressure `X Axis Acceleration` `Y Axis Acceleration`
##    <dbl>       <dbl>    <dbl>                 <dbl>                 <dbl>
##  1 16838        18.2     997.                  0.06                  0
##  2 20015        18.2     997.                  0.06                  0
##  3 24254        18.2     997.                  0.06                  0
##  4 26372        18.2     997.                  0.06                  0
##  5 28492        18.2     997.                  0.06                  0
##  6 34496        19.5     997.                  0.06                  0.5
##  7 37673        19.5     997.                  0.06                  0.44
##  8 38732        19.5     997.                  0.06                  0.44
##  9 47206        19.6     997.                  0.06                  0.44
## 10 54619        19.6     997.                  0.06                  0.37
## # i 25,147 more rows
## # i 4 more variables: `Z Axis Acceleration` <dbl>, CO2 <dbl>, TVOC <dbl>,
## #   DataFileNumber <dbl>
```

```
get_upper_fence<-function(x){
  quantile(x,.75)+(1.5*IQR(x))
}

get_lower_fence<-function(x){
  quantile(x,.25)-(1.5*IQR(x))
}
train<-train%>%
  filter_at(vars(CO2,TVOC,Pressure,Temperature),
            all_vars(.>get_lower_fence(.)&
                       .<get_upper_fence(.)))
model<-fit(object=linear_reg(),formula=TVOC~CO2+Pressure+Temperature+Time,data=train)
model
```

```
## parsnip model object
##
##
## Call:
## stats::lm(formula = TVOC ~ CO2 + Pressure + Temperature + Time,
##     data = data)
##
## Coefficients:
## (Intercept)          CO2     Pressure  Temperature         Time
##   6.378e+02    3.084e-01   -8.715e-01    4.178e+00    1.305e-06
```

```
summary(model)
```

```
##              Length Class      Mode
## lvl           0     -none-     NULL
## spec          6     linear_reg list
## fit          12     lm         list
## preproc       1     -none-     list
## elapsed       1     -none-     list
## censor_probs  0     -none-     list
```

```
model<-fit(object=linear_reg(),formula=TVOC~CO2+Pressure+Temperature+Time,data=train)
predict(model,new_data=test)
```

```
## # A tibble: 25,157 x 1
##    .pred
##    <dbl>
##  1 -31.6
##  2 -31.5
##  3 -31.4
##  4 -31.4
##  5 -31.4
##  6 -25.8
##  7 -25.9
##  8 -25.8
##  9 -25.7
## 10 -25.6
## # i 25,147 more rows
```

```
model_results<-test%>%mutate(predict(model,new_data=test))
model%>%tidy()
```

```
## # A tibble: 5 x 5
##   term            estimate     std.error statistic   p.value
##   <chr>              <dbl>         <dbl>     <dbl>     <dbl>
## 1 (Intercept) 638.         38.6              16.5  4.27e- 61
## 2 CO2           0.308       0.0901            3.42 6.16e-  4
## 3 Pressure     -0.872       0.0194          -45.0  0
## 4 Temperature   4.18        0.285             14.7  1.63e- 48
## 5 Time          0.00000130  0.0000000545      23.9  9.79e-126
```

```
coef(model)
```

```
## NULL
```

```
model_results
```

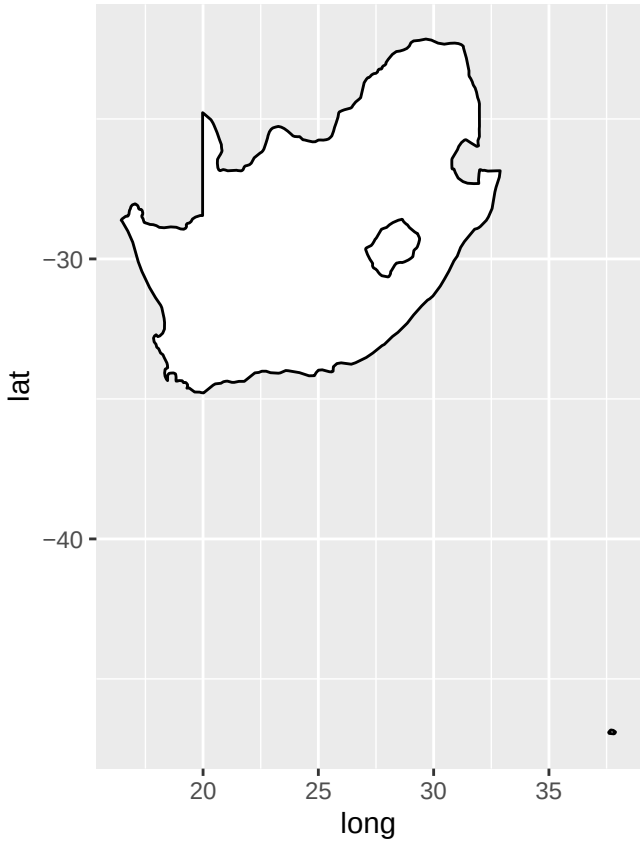
```
## # A tibble: 25,157 x 10
##     Time Temperature Pressure `X Axis Acceleration` `Y Axis Acceleration`
##    <dbl>       <dbl>    <dbl>                 <dbl>                 <dbl>
##  1 16838        18.2     997.                  0.06                  0
##  2 20015        18.2     997.                  0.06                  0
##  3 24254        18.2     997.                  0.06                  0
##  4 26372        18.2     997.                  0.06                  0
##  5 28492        18.2     997.                  0.06                  0
##  6 34496        19.5     997.                  0.06                  0.5
##  7 37673        19.5     997.                  0.06                  0.44
##  8 38732        19.5     997.                  0.06                  0.44
##  9 47206        19.6     997.                  0.06                  0.44
## 10 54619        19.6     997.                  0.06                  0.37
## # i 25,147 more rows
## # i 5 more variables: `Z Axis Acceleration` <dbl>, CO2 <dbl>, TVOC <dbl>,
## #   DataFileNumber <dbl>, .pred <dbl>
```

```
library ("maps")
```

```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
#get the SA map data
sa_map<-map_data("world","South Africa")
#plot the map
ggplot(sa_map,
       aes(x=long,y=lat,group=group))+geom_polygon(fill="white",color="black")+
  coord_quickmap()
```




```
library("rpart.plot")
```

```
## Loading required package: rpart
```

```
##
## Attaching package: 'rpart'
```

```
## The following object is masked from 'package:dials':
##
##     prune
```

```
library("caret")
```

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

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked from 'package:yardstick':
##
##     precision, recall, sensitivity, specificity
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
datarange<-read_csv("data_with_range.csv")%>%
  as_tibble()
```

```
## Rows: 99398 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Range
## dbl (9): Time, Temperature, Pressure, X Axis Acceleration, Y Axis Accelerati...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
datarange$Range <- as.factor(datarange$Range)

datarange$Range %>% unique()
```

```
## [1] Acceptable Concerning Dangerous
## Levels: Acceptable Concerning Dangerous
```

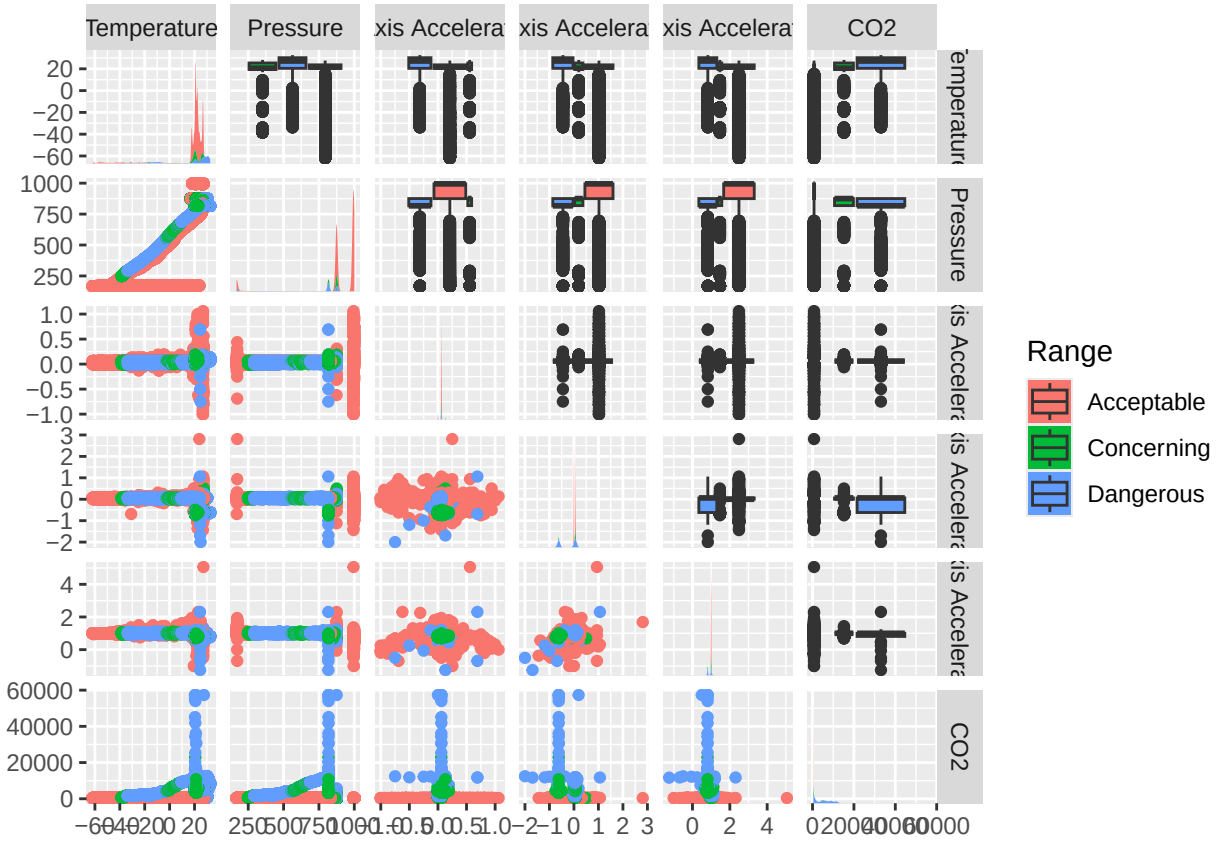
```
rangesplit <-initial_split(datarange, prop = 0.75)
train <-training(rangesplit)
test <-testing(rangesplit)
```

```
#plot different graphs of the values
ggplot(datarange, aes(x = .panel_x, y = .panel_y)) + geom_point(aes(color = Range)) + geom_autodensity(a
```

```
## Warning: Removed 60684 rows containing non-finite outside the scale range
## (`stat_autodensity()`).
```

```
## Warning: Removed 151710 rows containing missing values or values outside the scale range
## (`stat_boxplot()`).

## Warning: Removed 151710 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



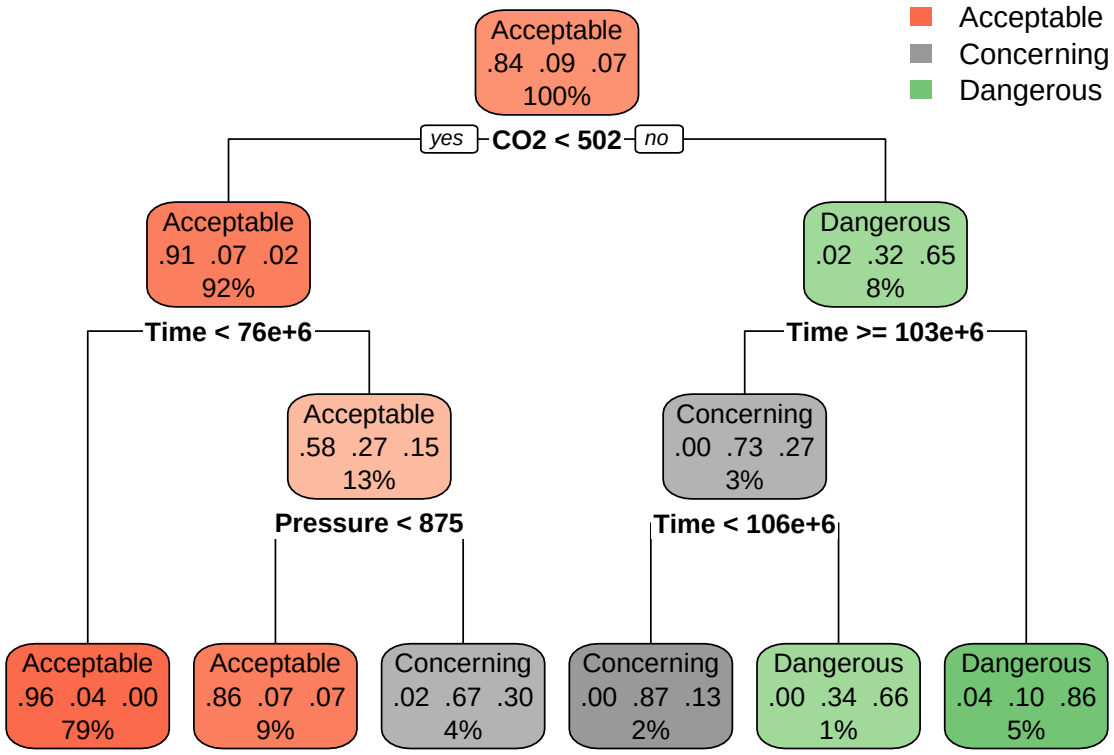

```
#minn is nodes to split the tree, mode is classification or regression, depth is maximum depth of tree
dtree_mod <-decision_tree(mode = "classification", min_n = 4, tree_depth = 3)
dtree_mod_fit <-fit(object = dtree_mod, formula = Range ~ Temperature+CO2+Pressure+Time, data = train)
dtree_mod_fit
```

```
## parsnip model object
##
## n=66972 (7576 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 66972 10776 Acceptable (0.8390969360 0.0917249000 0.0691781640)
##    2) CO2< 501.5 61852  5774 Acceptable (0.9066481278 0.0724794671 0.0208724051)
##      4) Time< 7.607167e+07 53197  2140 Acceptable (0.9597721676 0.0397390830 0.0004887494) *
##      5) Time>=7.607167e+07 8655  3634 Acceptable (0.5801270942 0.2737146158 0.1461582900)
##       10) Pressure< 875.455 5740   790 Acceptable (0.8623693380 0.0710801394 0.0665505226) *
##       11) Pressure>=875.455 2915   954 Concerning (0.0243567753 0.6727272727 0.3029159520) *
##    3) CO2>=501.5 5120  1778 Dangerous (0.0230468750 0.3242187500 0.6527343750)
```

```
##       6) Time>=1.034551e+08 1821   492 Concerning (0.0000000000 0.7298187809 0.2701812191)
##        12) Time< 1.055318e+08 1345   180 Concerning (0.0000000000 0.8661710037 0.1338289963) *
##        13) Time>=1.055318e+08 476   164 Dangerous (0.0000000000 0.3445378151 0.6554621849) *
##       7) Time< 1.034551e+08 3299   449 Dangerous (0.0357684147 0.1003334344 0.8638981510) *
```

```
dtree_mod_fit %>% extract_fit_engine() %>% rpart.plot()
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
predictions <-predict(dtree_mod_fit, new_data = test)
confusionMatrix(data = predictions$.pred_class, reference = test$Range)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Acceptable Concerning Dangerous
##   Acceptable      18776         770      2684
##   Concerning         19        1119       336
##   Dangerous          40         141       965
##
## Overall Statistics
```

```
##
##                Accuracy : 0.8394
##                  95% CI : (0.8348, 0.844)
##     No Information Rate : 0.7579
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4816
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Acceptable Class: Concerning Class: Dangerous
## Sensitivity                     0.9969           0.55123          0.24216
## Specificity                     0.4258           0.98444          0.99133
## Pos Pred Value                  0.8446           0.75916          0.84206
## Neg Pred Value                  0.9775           0.96103          0.87260
## Prevalence                      0.7579           0.08169          0.16036
## Detection Rate                  0.7556           0.04503          0.03883
## Detection Prevalence            0.8946           0.05932          0.04612
## Balanced Accuracy               0.7113           0.76784          0.61674
```

```
dtree_mod_fit %>% extract_fit_engine() %>% varImp()
```

```
##               Overall
## CO2          7109.064
## Pressure     6556.054
## Temperature  7033.191
## Time        10348.453
```

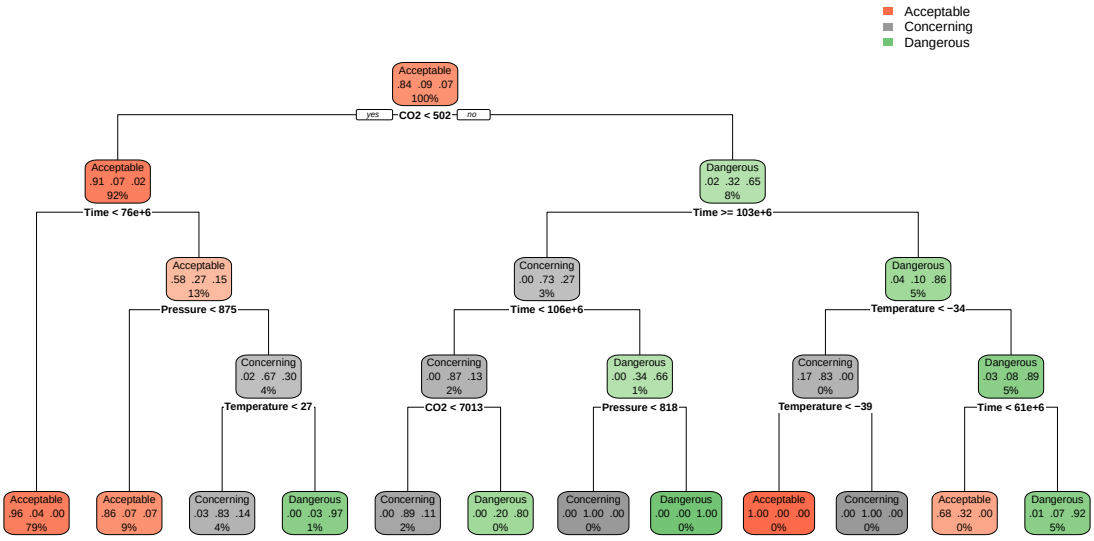
```
#all these variables seem fairly important based on the above
dtree_mod_hyper <-decision_tree(mode = "classification", min_n = 1, tree_depth = 4, cost_complexity = 0
dtree_mod_hyper_fit <-dtree_mod_hyper %>% fit(Range ~ Temperature + CO2+Pressure+Time, data = train)
dtree_mod_hyper_fit
```

```
## parsnip model object
##
## n=66972 (7576 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 66972 10776 Acceptable (0.8390969360 0.0917249000 0.0691781640)
##    2) CO2< 501.5 61852  5774 Acceptable (0.9066481278 0.0724794671 0.0208724051)
##      4) Time< 7.607167e+07 53197  2140 Acceptable (0.9597721676 0.0397390830 0.0004887494) *
##      5) Time>=7.607167e+07 8655  3634 Acceptable (0.5801270942 0.2737146158 0.1461582900)
##       10) Pressure< 875.455 5740   790 Acceptable (0.8623693380 0.0710801394 0.0665505226) *
##       11) Pressure>=875.455 2915   954 Concerning (0.0243567753 0.6727272727 0.3029159520)
##         22) Temperature< 26.905 2349   405 Concerning (0.0302256279 0.8275862069 0.1421881652) *
##         23) Temperature>=26.905 566    17 Dangerous (0.0000000000 0.0300353357 0.9699646643) *
##    3) CO2>=501.5 5120  1778 Dangerous (0.0230468750 0.3242187500 0.6527343750)
##      6) Time>=1.034551e+08 1821   492 Concerning (0.0000000000 0.7298187809 0.2701812191)
##       12) Time< 1.055318e+08 1345   180 Concerning (0.0000000000 0.8661710037 0.1338289963)
```

```
##         24) CO2< 7013 1295   140 Concerning (0.0000000000 0.8918918919 0.1081081081) *
##         25) CO2>=7013 50    10 Dangerous (0.0000000000 0.2000000000 0.8000000000) *
##       13) Time>=1.055318e+08 476   164 Dangerous (0.0000000000 0.3445378151 0.6554621849)
##         26) Pressure< 817.875 164     0 Concerning (0.0000000000 1.0000000000 0.0000000000) *
##         27) Pressure>=817.875 312     0 Dangerous (0.0000000000 0.0000000000 1.0000000000) *
##      7) Time< 1.034551e+08 3299   449 Dangerous (0.0357684147 0.1003334344 0.8638981510)
##       14) Temperature< -33.755 104    18 Concerning (0.1730769231 0.8269230769 0.0000000000)
##         28) Temperature< -38.68 18     0 Acceptable (1.0000000000 0.0000000000 0.0000000000) *
##         29) Temperature>=-38.68 86     0 Concerning (0.0000000000 1.0000000000 0.0000000000) *
##       15) Temperature>=-33.755 3195   345 Dangerous (0.0312989045 0.0766823161 0.8920187793)
##         30) Time< 6.145367e+07 102    33 Acceptable (0.6764705882 0.3235294118 0.0000000000) *
##         31) Time>=6.145367e+07 3093   243 Dangerous (0.0100226317 0.0685418687 0.9214354995) *
```

```
dtree_mod_hyper_fit %>% extract_fit_engine() %>% rpart.plot()
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
predictions2 <-predict(dtree_mod_hyper_fit, new_data = test)
confusionMatrix(data = predictions2$.pred_class, reference = test$Range)
```

```
## Confusion Matrix and Statistics
```

```
##
##             Reference
## Prediction   Acceptable Concerning Dangerous
##   Acceptable      18804        780      2685
##   Concerning         19       1186       162
##   Dangerous          12         64      1138
##
## Overall Statistics
##
##                Accuracy : 0.8502
##                  95% CI : (0.8457, 0.8546)
##     No Information Rate : 0.7579
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5144
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Acceptable Class: Concerning Class: Dangerous
## Sensitivity                     0.9984           0.58424          0.28557
## Specificity                     0.4239           0.99207          0.99636
## Pos Pred Value                  0.8444           0.86759          0.93740
## Neg Pred Value                  0.9880           0.96406          0.87955
## Prevalence                      0.7579           0.08169          0.16036
## Detection Rate                  0.7567           0.04773          0.04579
## Detection Prevalence            0.8961           0.05501          0.04885
## Balanced Accuracy               0.7111           0.78815          0.64096
```

```
dtree_mod_hyper_fit %>% extract_fit_engine() %>% varImp()
```

```
##               Overall
## CO2          7749.262
## Pressure     7134.363
## Temperature  7953.216
## Time        11203.160
```

```
folds <-vfold_cv(train, v = 5)
folds
```

```
## #  5-fold cross-validation
## # A tibble: 5 x 2
##   splits               id
##   <list>               <chr>
## 1 <split [59638/14910]> Fold1
## 2 <split [59638/14910]> Fold2
## 3 <split [59638/14910]> Fold3
## 4 <split [59639/14909]> Fold4
## 5 <split [59639/14909]> Fold5
```

```
dtree_mod <-decision_tree(mode = "classification") # an empty model
tuned <-tune_grid(dtree_mod, preprocessor = Range ~ Time+Pressure+CO2+Temperature, resamples = folds)
```

```
## Warning: No tuning parameters have been detected, performance will be evaluated
## using the resamples with no tuning. Did you want to [tune()] parameters?
```

```
tuned$.metrics
```

```
## [[1]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.869 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.776 Preprocessor1_Model1
## 3 brier_class multiclass     0.126 Preprocessor1_Model1
##
## [[2]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.868 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.781 Preprocessor1_Model1
## 3 brier_class multiclass     0.127 Preprocessor1_Model1
##
## [[3]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.871 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.779 Preprocessor1_Model1
## 3 brier_class multiclass     0.125 Preprocessor1_Model1
##
## [[4]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.866 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.781 Preprocessor1_Model1
## 3 brier_class multiclass     0.129 Preprocessor1_Model1
##
## [[5]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.865 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.777 Preprocessor1_Model1
## 3 brier_class multiclass     0.130 Preprocessor1_Model1
```

```
collect_metrics(tuned)
```

```
## # A tibble: 3 x 6
##   .metric     .estimator  mean     n  std_err .config
```

```
##   <chr>       <chr>      <dbl> <int>    <dbl> <chr>
## 1 accuracy    multiclass 0.868     5 0.00106  Preprocessor1_Model1
## 2 brier_class multiclass 0.127     5 0.000978 Preprocessor1_Model1
## 3 roc_auc     hand_till  0.779     5 0.00108  Preprocessor1_Model1
```

```
tunable(dtree_mod)
```

```
## # A tibble: 3 x 5
##   name            call_info        source     component     component_id
##   <chr>           <list>           <chr>      <chr>         <chr>
## 1 tree_depth      <named list [2]> model_spec decision_tree main
## 2 min_n           <named list [2]> model_spec decision_tree main
## 3 cost_complexity <named list [2]> model_spec decision_tree main
```

```
dtree_mod_tune <-decision_tree(mode = "classification", tree_depth = tune(), min_n = tune(), cost_comple
tree_grid_auto <-grid_regular(tree_depth(), min_n(), cost_complexity())
tree_grid_auto
```

```
## # A tibble: 27 x 3
##    tree_depth min_n cost_complexity
##         <int> <int>           <dbl>
##  1          1     2    0.0000000001
##  2          8     2    0.0000000001
##  3         15     2    0.0000000001
##  4          1    21    0.0000000001
##  5          8    21    0.0000000001
##  6         15    21    0.0000000001
##  7          1    40    0.0000000001
##  8          8    40    0.0000000001
##  9         15    40    0.0000000001
## 10          1     2    0.00000316
## # i 17 more rows
```

```
tree_grid_manual <-expand.grid(tree_depth = 2:5, min_n = seq(1, 51, 10), cost_complexity = c(0,.1,.2,.3)
tree_grid_manual
```

```
## # A tibble: 96 x 3
##    tree_depth min_n cost_complexity
##         <int> <dbl>           <dbl>
##  1          2     1               0
##  2          3     1               0
##  3          4     1               0
##  4          5     1               0
##  5          2    11               0
##  6          3    11               0
##  7          4    11               0
##  8          5    11               0
##  9          2    21               0
## 10          3    21               0
## # i 86 more rows
```

```
tuning_data <-tune_grid(dtree_mod_tune, preprocessor = Range ~ CO2+Temperature+Pressure+Time,resamples=
tuning_data %>% collect_metrics()
```

```
## # A tibble: 81 x 9
##    cost_complexity tree_depth min_n .metric     .estimator  mean     n  std_err
##              <dbl>      <int> <int> <chr>       <chr>      <dbl> <int>    <dbl>
##  1    0.0000000001          1     2 accuracy    multiclass 0.797     5 0.00165
##  2    0.0000000001          1     2 brier_class multiclass 0.178     5 0.00133
##  3    0.0000000001          1     2 roc_auc     hand_till  0.590     5 0.00119
##  4    0.0000000001          8     2 accuracy    multiclass 0.885     5 0.000841
##  5    0.0000000001          8     2 brier_class multiclass 0.112     5 0.000697
##  6    0.0000000001          8     2 roc_auc     hand_till  0.826     5 0.00125
##  7    0.0000000001         15     2 accuracy    multiclass 0.894     5 0.000674
##  8    0.0000000001         15     2 brier_class multiclass 0.105     5 0.000669
##  9    0.0000000001         15     2 roc_auc     hand_till  0.837     5 0.00127
## 10    0.0000000001          1    21 accuracy    multiclass 0.797     5 0.00165
## # i 71 more rows
## # i 1 more variable: .config <chr>
```

```
tuning_data %>% select_best(metric = "accuracy")
```

```
## # A tibble: 1 x 4
##   cost_complexity tree_depth min_n .config
##             <dbl>      <int> <int> <chr>
## 1    0.0000000001         15     2 Preprocessor1_Model03
```

```
tuning_data %>% show_best(metric = "accuracy")
```

```
## # A tibble: 5 x 9
##   cost_complexity tree_depth min_n .metric  .estimator  mean     n  std_err
##             <dbl>      <int> <int> <chr>    <chr>      <dbl> <int>    <dbl>
## 1    0.0000000001         15     2 accuracy multiclass 0.894     5 0.000674
## 2    0.00000316           15     2 accuracy multiclass 0.894     5 0.000674
## 3    0.0000000001         15    21 accuracy multiclass 0.894     5 0.000878
## 4    0.00000316           15    21 accuracy multiclass 0.894     5 0.000878
## 5    0.0000000001         15    40 accuracy multiclass 0.893     5 0.000892
## # i 1 more variable: .config <chr>
```

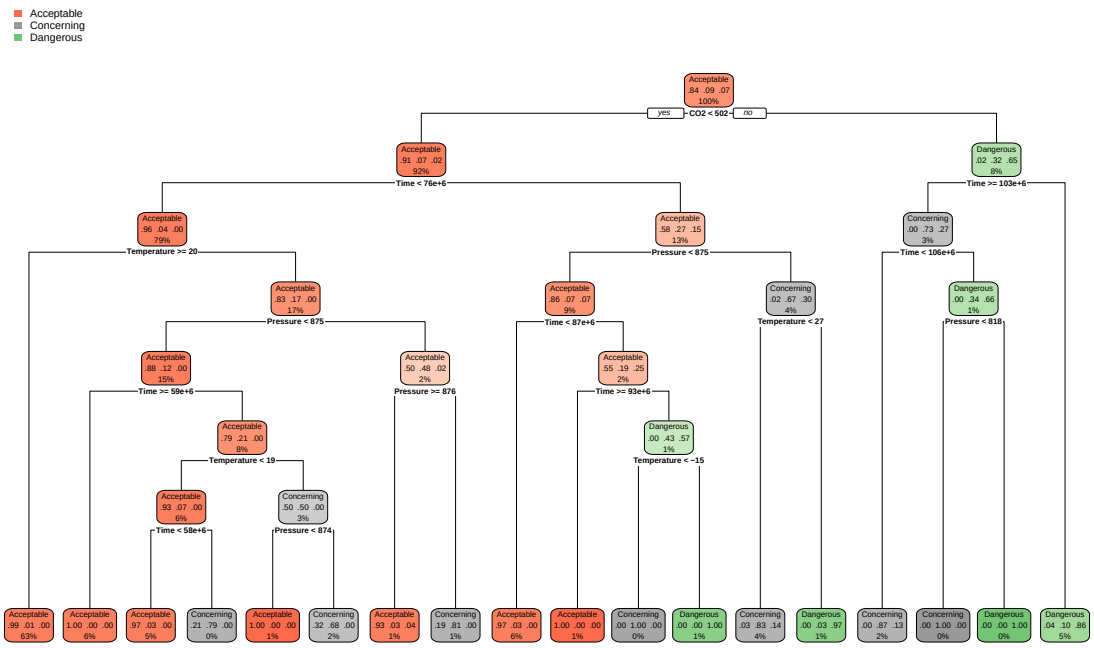
```
dtree_mod_betta <-decision_tree(mode = "classification", min_n = 2, tree_depth = 15,cost_complexity=0.0
dtree_mod_betta_fit <-fit(object = dtree_mod, formula = Range ~ Temperature+CO2+Pressure+Time, data = t
dtree_mod_betta_fit
```

```
## parsnip model object
##
## n=66972 (7576 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 66972 10776 Acceptable (0.8390969360 0.0917249000 0.0691781640)
##     2) CO2< 501.5 61852  5774 Acceptable (0.9066481278 0.0724794671 0.0208724051)
```

```
##       4) Time< 7.607167e+07 53197  2140 Acceptable (0.9597721676 0.0397390830 0.0004887494)
##         8) Temperature>=20.285 41898   233 Acceptable (0.9944388754 0.0055611246 0.0000000000) *
##         9) Temperature< 20.285 11299  1907 Acceptable (0.8312240021 0.1664749093 0.0023010886)
##          18) Pressure< 875.495 9787  1149 Acceptable (0.8825993665 0.1174006335 0.0000000000)
##            36) Time>=5.884288e+07 4252     4 Acceptable (0.9990592662 0.0009407338 0.0000000000) *
##            37) Time< 5.884288e+07 5535  1145 Acceptable (0.7931345980 0.2068654020 0.0000000000)
##              74) Temperature< 18.655 3801   275 Acceptable (0.9276506183 0.0723493817 0.0000000000)
##               148) Time< 5.797473e+07 3613   126 Acceptable (0.9651259341 0.0348740659 0.0000000000)
##               149) Time>=5.797473e+07 188    39 Concerning (0.2074468085 0.7925531915 0.0000000000) *
##              75) Temperature>=18.655 1734   864 Concerning (0.4982698962 0.5017301038 0.0000000000)
##               150) Pressure< 874.435 456     0 Acceptable (1.0000000000 0.0000000000 0.0000000000) *
##               151) Pressure>=874.435 1278   408 Concerning (0.3192488263 0.6807511737 0.0000000000) *
##          19) Pressure>=875.495 1512   758 Acceptable (0.4986772487 0.4841269841 0.0171957672)
##            38) Pressure>=876.095 637    46 Acceptable (0.9277864992 0.0313971743 0.0408163265) *
##            39) Pressure< 876.095 875   163 Concerning (0.1862857143 0.8137142857 0.0000000000) *
##       5) Time>=7.607167e+07 8655  3634 Acceptable (0.5801270942 0.2737146158 0.1461582900)
##        10) Pressure< 875.455 5740   790 Acceptable (0.8623693380 0.0710801394 0.0665505226)
##          20) Time< 8.691794e+07 4298   145 Acceptable (0.9662633783 0.0302466263 0.0034899953) *
##          21) Time>=8.691794e+07 1442   645 Acceptable (0.5527045770 0.1927877947 0.2545076283)
##            42) Time>=9.325133e+07 797     0 Acceptable (1.0000000000 0.0000000000 0.0000000000) *
##            43) Time< 9.325133e+07 645   278 Dangerous (0.0000000000 0.4310077519 0.5689922481)
##              86) Temperature< -15.255 278     1 Concerning (0.0000000000 0.9964028777 0.0035971223) *
##              87) Temperature>=-15.255 367     1 Dangerous (0.0000000000 0.0027247956 0.9972752044) *
##        11) Pressure>=875.455 2915   954 Concerning (0.0243567753 0.6727272727 0.3029159520)
##          22) Temperature< 26.905 2349   405 Concerning (0.0302256279 0.8275862069 0.1421881652) *
##          23) Temperature>=26.905 566    17 Dangerous (0.0000000000 0.0300353357 0.9699646643) *
##     3) CO2>=501.5 5120  1778 Dangerous (0.0230468750 0.3242187500 0.6527343750)
##       6) Time>=1.034551e+08 1821   492 Concerning (0.0000000000 0.7298187809 0.2701812191)
##        12) Time< 1.055318e+08 1345   180 Concerning (0.0000000000 0.8661710037 0.1338289963) *
##        13) Time>=1.055318e+08 476   164 Dangerous (0.0000000000 0.3445378151 0.6554621849)
##          26) Pressure< 817.875 164     0 Concerning (0.0000000000 1.0000000000 0.0000000000) *
##          27) Pressure>=817.875 312     0 Dangerous (0.0000000000 0.0000000000 1.0000000000) *
##       7) Time< 1.034551e+08 3299   449 Dangerous (0.0357684147 0.1003334344 0.8638981510) *
```

```
dtree_mod_betta_fit %>% extract_fit_engine() %>% rpart.plot()
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
predictionsbetta <-predict(dtree_mod_betta_fit, new_data = test)
confusionMatrix(data = predictionsbetta$.pred_class, reference = test$Range)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Acceptable Concerning Dangerous
##   Acceptable      18546         157      2548
##   Concerning        249        1776       174
##   Dangerous          40          97      1263
##
## Overall Statistics
##
##                Accuracy : 0.8686
##                  95% CI : (0.8643, 0.8728)
##     No Information Rate : 0.7579
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6085
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Acceptable Class: Concerning Class: Dangerous
```

```
## Sensitivity                 0.9847          0.87488          0.31694
## Specificity                 0.5503          0.98146          0.99343
## Pos Pred Value              0.8727          0.80764          0.90214
## Neg Pred Value              0.9197          0.98879          0.88392
## Prevalence                  0.7579          0.08169          0.16036
## Detection Rate              0.7463          0.07147          0.05082
## Detection Prevalence        0.8552          0.08849          0.05634
## Balanced Accuracy           0.7675          0.92817          0.65519
```

```
dtree_mod_betta_fit %>% extract_fit_engine() %>% varImp()
```

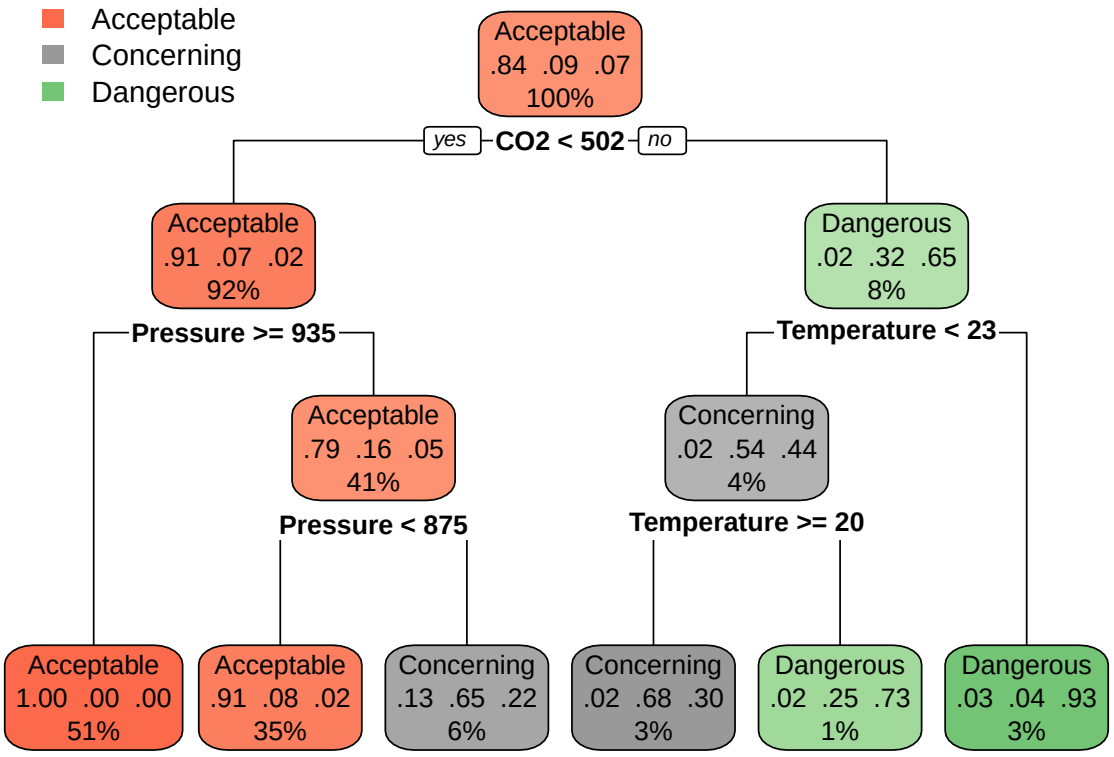
```
##                Overall
## CO2           7796.842
## Pressure      9624.150
## Temperature   9955.989
## Time         14149.008
```

```
detree_mod <-decision_tree(mode = "classification", min_n = 4, tree_depth = 3)
detree_mod_fit <-fit(object = detree_mod, formula = Range ~ Temperature+CO2+Pressure, data = train)
detree_mod_fit
```

```
## parsnip model object
##
## n=66972 (7576 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 66972 10776 Acceptable (0.83909694 0.09172490 0.06917816)
##    2) CO2< 501.5 61852  5774 Acceptable (0.90664813 0.07247947 0.02087241)
##      4) Pressure>=935.115 34289     0 Acceptable (1.00000000 0.00000000 0.00000000) *
##      5) Pressure< 935.115 27563  5774 Acceptable (0.79051627 0.16264558 0.04683815)
##       10) Pressure< 875.455 23428  2166 Acceptable (0.90754653 0.07614820 0.01630528) *
##       11) Pressure>=875.455 4135  1436 Concerning (0.12744861 0.65272068 0.21983071) *
##    3) CO2>=501.5 5120  1778 Dangerous (0.02304688 0.32421875 0.65273437)
##      6) Temperature< 22.995 2898  1326 Concerning (0.02035887 0.54244306 0.43719807)
##       12) Temperature>=20.305 1960   625 Concerning (0.02091837 0.68112245 0.29795918) *
##       13) Temperature< 20.305 938   255 Dangerous (0.01918977 0.25266525 0.72814499) *
##      7) Temperature>=22.995 2222   147 Dangerous (0.02655266 0.03960396 0.93384338) *
```

```
detree_mod_fit %>% extract_fit_engine() %>% rpart.plot()
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
predictions3 <-predict(detree_mod_fit, new_data = test)
confusionMatrix(data = predictions3$.pred_class, reference = test$Range)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Acceptable Concerning Dangerous
##   Acceptable      18622         524      2677
##   Concerning        186        1416       474
##   Dangerous          27          90       834
##
## Overall Statistics
##
##                Accuracy : 0.8399
##                  95% CI : (0.8353, 0.8445)
##     No Information Rate : 0.7579
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.502
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: Acceptable Class: Concerning Class: Dangerous
```

```
## Sensitivity                 0.9887          0.69754          0.20928
## Specificity                 0.4678          0.97108          0.99439
## Pos Pred Value              0.8533          0.68208          0.87697
## Neg Pred Value              0.9296          0.97304          0.86815
## Prevalence                  0.7579          0.08169          0.16036
## Detection Rate              0.7494          0.05698          0.03356
## Detection Prevalence        0.8782          0.08354          0.03827
## Balanced Accuracy           0.7283          0.83431          0.60184
```

```
detree_mod_fit %>% extract_fit_engine() %>% varImp()
```

```
##              Overall
## CO2         7079.562
## Pressure    8013.870
## Temperature 7630.415
```

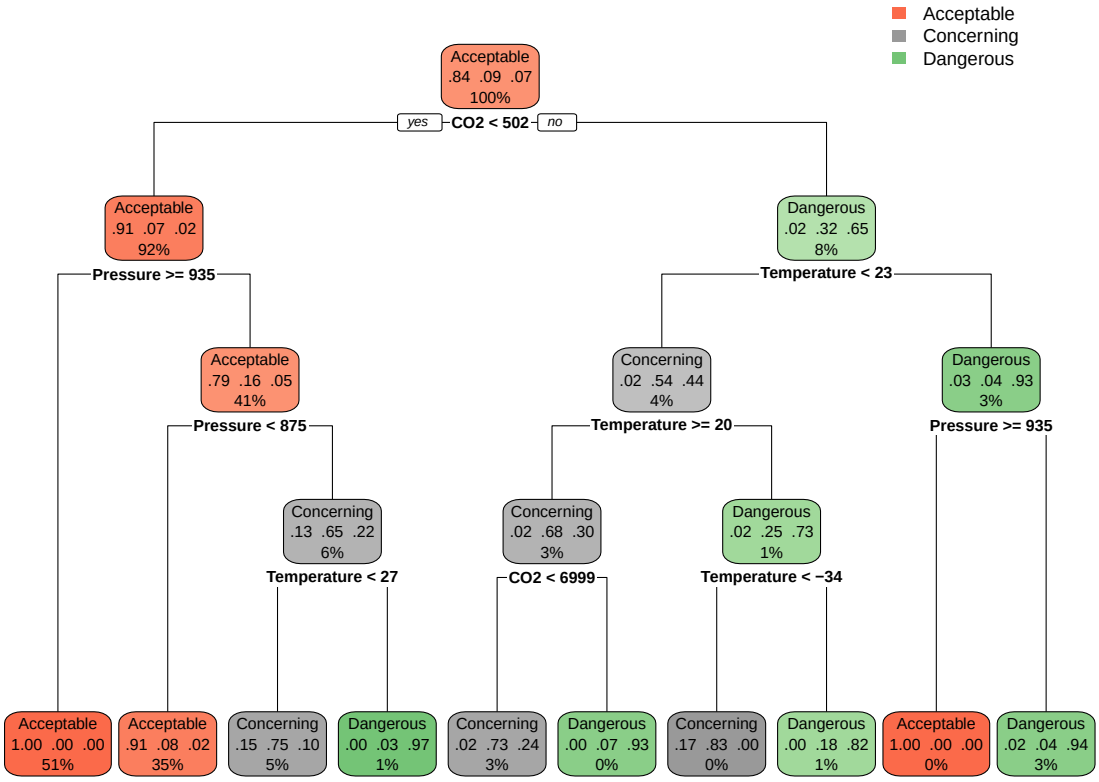
```
#all these variables seem fairly important based on the above
detree_mod_hyper <-decision_tree(mode = "classification", min_n = 1, tree_depth = 4, cost_complexity = 0
detree_mod_hyper_fit <-detree_mod_hyper %>% fit(Range ~ Temperature + CO2+Pressure, data = train)
detree_mod_hyper_fit
```

```
## parsnip model object
##
## n=66972 (7576 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 66972 10776 Acceptable (0.83909694 0.09172490 0.06917816)
##    2) CO2< 501.5 61852  5774 Acceptable (0.90664813 0.07247947 0.02087241)
##      4) Pressure>=935.115 34289     0 Acceptable (1.00000000 0.00000000 0.00000000) *
##      5) Pressure< 935.115 27563  5774 Acceptable (0.79051627 0.16264558 0.04683815)
##       10) Pressure< 875.455 23428  2166 Acceptable (0.90754653 0.07614820 0.01630528) *
##       11) Pressure>=875.455 4135  1436 Concerning (0.12744861 0.65272068 0.21983071)
##         22) Temperature< 26.905 3569   887 Concerning (0.14766041 0.75147100 0.10086859) *
##         23) Temperature>=26.905 566    17 Dangerous (0.00000000 0.03003534 0.96996466) *
##    3) CO2>=501.5 5120  1778 Dangerous (0.02304688 0.32421875 0.65273437)
##      6) Temperature< 22.995 2898  1326 Concerning (0.02035887 0.54244306 0.43719807)
##       12) Temperature>=20.305 1960   625 Concerning (0.02091837 0.68112245 0.29795918)
##         24) CO2< 6998.5 1802   478 Concerning (0.02275250 0.73473918 0.24250832) *
##         25) CO2>=6998.5 158    11 Dangerous (0.00000000 0.06962025 0.93037975) *
##       13) Temperature< 20.305 938   255 Dangerous (0.01918977 0.25266525 0.72814499)
##         26) Temperature< -33.755 104    18 Concerning (0.17307692 0.82692308 0.00000000) *
##         27) Temperature>=-33.755 834   151 Dangerous (0.00000000 0.18105516 0.81894484) *
##      7) Temperature>=22.995 2222   147 Dangerous (0.02655266 0.03960396 0.93384338)
##       14) Pressure>=935.065 22     0 Acceptable (1.00000000 0.00000000 0.00000000) *
##       15) Pressure< 935.065 2200   125 Dangerous (0.01681818 0.04000000 0.94318182) *
```

```
detree_mod_hyper_fit %>% extract_fit_engine() %>% rpart.plot()
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
## To silence this warning:
```

```
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
predictions4 <-predict(detree_mod_hyper_fit, new_data = test)
confusionMatrix(data = predictions4$.pred_class, reference = test$Range)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Acceptable Concerning Dangerous
##   Acceptable      18633        524      2677
##   Concerning        188       1434       266
##   Dangerous          14         72      1042
##
## Overall Statistics
##
##                Accuracy : 0.8495
##                  95% CI : (0.8449, 0.8539)
##     No Information Rate : 0.7579
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5304
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
## Statistics by Class:
##
##                      Class: Acceptable Class: Concerning Class: Dangerous
## Sensitivity                     0.9893           0.70640          0.26148
## Specificity                     0.4678           0.98011          0.99588
## Pos Pred Value                  0.8534           0.75953          0.92376
## Neg Pred Value                  0.9330           0.97404          0.87594
## Prevalence                      0.7579           0.08169          0.16036
## Detection Rate                  0.7498           0.05771          0.04193
## Detection Prevalence            0.8786           0.07598          0.04539
## Balanced Accuracy               0.7286           0.84325          0.62868
```

```
detree_mod_hyper_fit %>% extract_fit_engine() %>% varImp()
```

```
##              Overall
## CO2         7653.973
## Pressure    8398.010
## Temperature 8459.076
```

```
folds1 <-vfold_cv(train, v = 5)
folds1
```

```
## #  5-fold cross-validation
## # A tibble: 5 x 2
##   splits                id
##   <list>                <chr>
## 1 <split [59638/14910]> Fold1
## 2 <split [59638/14910]> Fold2
## 3 <split [59638/14910]> Fold3
## 4 <split [59639/14909]> Fold4
## 5 <split [59639/14909]> Fold5
```

```
detree_mod <-decision_tree(mode = "classification") # an empty model
tuned1 <-tune_grid(detree_mod, preprocessor = Range ~ Pressure+CO2+Temperature, resamples = folds)
```

```
## Warning: No tuning parameters have been detected, performance will be evaluated
## using the resamples with no tuning. Did you want to [tune()] parameters?
```

```
tuned1$.metrics
```

```
## [[1]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.850 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.708 Preprocessor1_Model1
## 3 brier_class multiclass     0.144 Preprocessor1_Model1
##
## [[2]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
```

```
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.848 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.715 Preprocessor1_Model1
## 3 brier_class multiclass     0.145 Preprocessor1_Model1
##
## [[3]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.847 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.707 Preprocessor1_Model1
## 3 brier_class multiclass     0.146 Preprocessor1_Model1
##
## [[4]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.845 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.715 Preprocessor1_Model1
## 3 brier_class multiclass     0.147 Preprocessor1_Model1
##
## [[5]]
## # A tibble: 3 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    multiclass     0.845 Preprocessor1_Model1
## 2 roc_auc     hand_till      0.712 Preprocessor1_Model1
## 3 brier_class multiclass     0.148 Preprocessor1_Model1
```

```
collect_metrics(tuned1)
```

```
## # A tibble: 3 x 6
##   .metric     .estimator  mean     n  std_err .config
##   <chr>       <chr>      <dbl> <int>    <dbl> <chr>
## 1 accuracy    multiclass 0.847     5 0.000920 Preprocessor1_Model1
## 2 brier_class multiclass 0.146     5 0.000764 Preprocessor1_Model1
## 3 roc_auc     hand_till  0.711     5 0.00173  Preprocessor1_Model1
```

```
tunable(detree_mod)
```

```
## # A tibble: 3 x 5
##   name            call_info        source     component     component_id
##   <chr>           <list>           <chr>      <chr>         <chr>
## 1 tree_depth      <named list [2]> model_spec decision_tree main
## 2 min_n           <named list [2]> model_spec decision_tree main
## 3 cost_complexity <named list [2]> model_spec decision_tree main
```

```
detree_mod_tune <-decision_tree(mode = "classification", tree_depth = tune(), min_n = tune(), cost_compl
tree_grid_auto1 <-grid_regular(tree_depth(), min_n(), cost_complexity())
tree_grid_auto1
```

```
## # A tibble: 27 x 3
```

```
##    tree_depth min_n cost_complexity
##         <int> <int>           <dbl>
##  1          1     2    0.0000000001
##  2          8     2    0.0000000001
##  3         15     2    0.0000000001
##  4          1    21    0.0000000001
##  5          8    21    0.0000000001
##  6         15    21    0.0000000001
##  7          1    40    0.0000000001
##  8          8    40    0.0000000001
##  9         15    40    0.0000000001
## 10          1     2    0.00000316
## # i 17 more rows
```

```
tree_grid_manual1 <-expand.grid(tree_depth = 2:5, min_n = seq(1, 51, 10), cost_complexity = c(0,.1,.2,.3
tree_grid_manual1
```

```
## # A tibble: 96 x 3
##    tree_depth min_n cost_complexity
##         <int> <dbl>           <dbl>
##  1          2     1               0
##  2          3     1               0
##  3          4     1               0
##  4          5     1               0
##  5          2    11               0
##  6          3    11               0
##  7          4    11               0
##  8          5    11               0
##  9          2    21               0
## 10          3    21               0
## # i 86 more rows
```

```
tuning_data <-tune_grid(detree_mod_tune, preprocessor = Range ~ CO2+Temperature+Pressure,resamples=folds
tuning_data %>% collect_metrics()
```

```
## # A tibble: 81 x 9
##    cost_complexity tree_depth min_n .metric     .estimator  mean     n  std_err
##              <dbl>      <int> <int> <chr>       <chr>      <dbl> <int>    <dbl>
##  1    0.0000000001          1     2 accuracy    multiclass 0.797     5 0.00186
##  2    0.0000000001          1     2 brier_class multiclass 0.178     5 0.00171
##  3    0.0000000001          1     2 roc_auc     hand_till  0.590     5 0.00127
##  4    0.0000000001          8     2 accuracy    multiclass 0.872     5 0.00103
##  5    0.0000000001          8     2 brier_class multiclass 0.122     5 0.00103
##  6    0.0000000001          8     2 roc_auc     hand_till  0.787     5 0.00185
##  7    0.0000000001         15     2 accuracy    multiclass 0.888     5 0.000639
##  8    0.0000000001         15     2 brier_class multiclass 0.111     5 0.000695
##  9    0.0000000001         15     2 roc_auc     hand_till  0.824     5 0.00160
## 10    0.0000000001          1    21 accuracy    multiclass 0.797     5 0.00186
## # i 71 more rows
## # i 1 more variable: .config <chr>
```

```
tuning_data %>% select_best(metric = "accuracy")
```

```
## # A tibble: 1 x 4
##   cost_complexity tree_depth min_n .config
##             <dbl>      <int> <int> <chr>
## 1    0.0000000001         15     2 Preprocessor1_Model03
```

```
tuning_data %>% show_best(metric = "accuracy")
```

```
## # A tibble: 5 x 9
##   cost_complexity tree_depth min_n .metric  .estimator  mean     n  std_err
##             <dbl>      <int> <int> <chr>    <chr>      <dbl> <int>    <dbl>
## 1    0.0000000001         15     2 accuracy multiclass 0.888     5 0.000639
## 2    0.00000316           15     2 accuracy multiclass 0.888     5 0.000639
## 3    0.0000000001         15    21 accuracy multiclass 0.887     5 0.000521
## 4    0.00000316           15    21 accuracy multiclass 0.887     5 0.000521
## 5    0.0000000001         15    40 accuracy multiclass 0.885     5 0.000683
## # i 1 more variable: .config <chr>
```

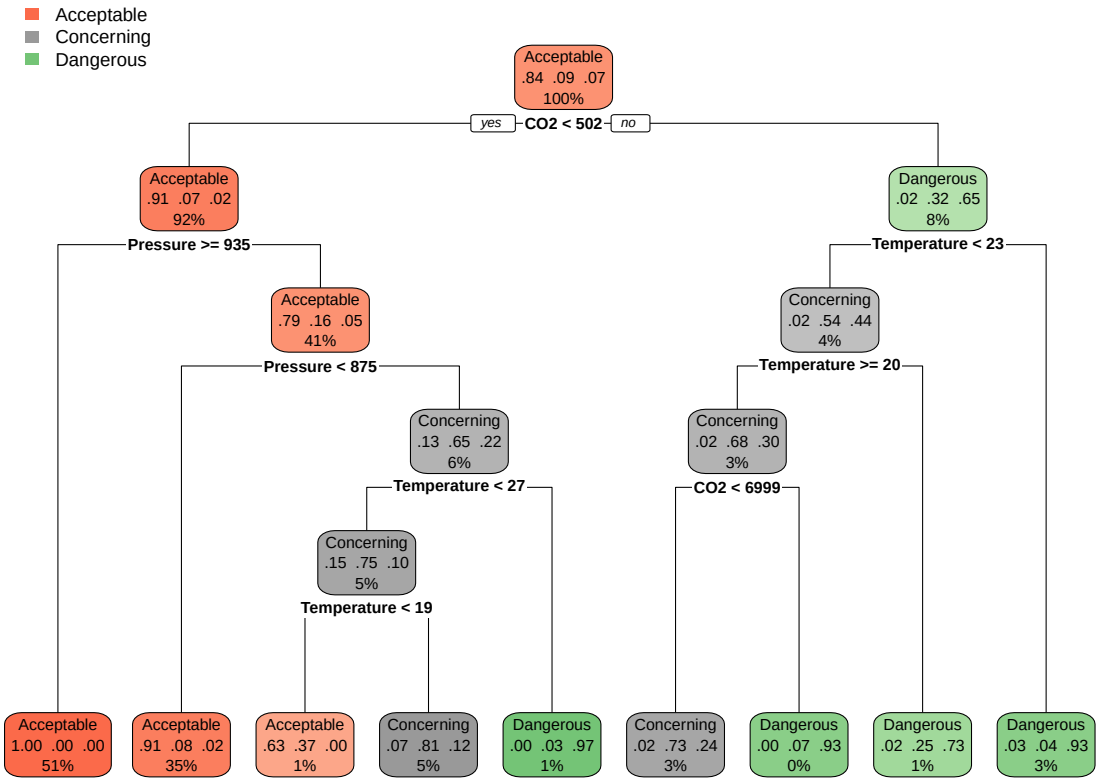
```
detree_mod_betta <-decision_tree(mode = "classification", min_n = 2, tree_depth = 15,cost_complexity=0.
detree_mod_betta_fit <-fit(object = detree_mod, formula = Range ~ Temperature+CO2+Pressure, data = trai
detree_mod_betta_fit
```

```
## parsnip model object
##
## n=66972 (7576 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 66972 10776 Acceptable (0.83909694 0.09172490 0.06917816)
##    2) CO2< 501.5 61852  5774 Acceptable (0.90664813 0.07247947 0.02087241)
##      4) Pressure>=935.115 34289     0 Acceptable (1.00000000 0.00000000 0.00000000) *
##      5) Pressure< 935.115 27563  5774 Acceptable (0.79051627 0.16264558 0.04683815)
##       10) Pressure< 875.455 23428  2166 Acceptable (0.90754653 0.07614820 0.01630528) *
##       11) Pressure>=875.455 4135  1436 Concerning (0.12744861 0.65272068 0.21983071)
##         22) Temperature< 26.905 3569   887 Concerning (0.14766041 0.75147100 0.10086859)
##           44) Temperature< 19.325 472   173 Acceptable (0.63347458 0.36652542 0.00000000) *
##           45) Temperature>=19.325 3097   588 Concerning (0.07361963 0.81013884 0.11624152) *
##         23) Temperature>=26.905 566    17 Dangerous (0.00000000 0.03003534 0.96996466) *
##    3) CO2>=501.5 5120  1778 Dangerous (0.02304688 0.32421875 0.65273437)
##      6) Temperature< 22.995 2898  1326 Concerning (0.02035887 0.54244306 0.43719807)
##       12) Temperature>=20.305 1960   625 Concerning (0.02091837 0.68112245 0.29795918)
##         24) CO2< 6998.5 1802   478 Concerning (0.02275250 0.73473918 0.24250832) *
##         25) CO2>=6998.5 158    11 Dangerous (0.00000000 0.06962025 0.93037975) *
##       13) Temperature< 20.305 938   255 Dangerous (0.01918977 0.25266525 0.72814499) *
##      7) Temperature>=22.995 2222   147 Dangerous (0.02655266 0.03960396 0.93384338) *
```

```
detree_mod_betta_fit %>% extract_fit_engine() %>% rpart.plot()
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
```

```
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```




```
predictions1betta <-predict(detree_mod_betta_fit, new_data = test)
confusionMatrix(data = predictions1betta$.pred_class, reference = test$Range)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   Acceptable Concerning Dangerous
##   Acceptable      18722         581      2677
##   Concerning         86        1351       266
##   Dangerous          27          98      1042
##
## Overall Statistics
##
##                Accuracy : 0.8497
##                  95% CI : (0.8452, 0.8541)
##     No Information Rate : 0.7579
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5251
##
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
## Statistics by Class:
##
##                      Class: Acceptable Class: Concerning Class: Dangerous
## Sensitivity                     0.9940           0.66552          0.26148
## Specificity                     0.4584           0.98457          0.99401
## Pos Pred Value                  0.8518           0.79331          0.89289
## Neg Pred Value                  0.9606           0.97067          0.87573
## Prevalence                      0.7579           0.08169          0.16036
## Detection Rate                  0.7534           0.05437          0.04193
## Detection Prevalence            0.8845           0.06853          0.04696
## Balanced Accuracy               0.7262           0.82505          0.62774
```

```
detree_mod_betta_fit %>% extract_fit_engine() %>% varImp()
```

```
##              Overall
## CO2         7588.090
## Pressure    8415.637
## Temperature 8535.516
```