Hardcoat Acrylic Film Degradation Data Assembly, Cleaning, EDA, Summary

Abhi Devathi

10/3/2017

Summary

This project asks us to clean data used for Hardcoat Acrylic Film Degradation project that was done in the Solar Durability and Lifetime Extension (SDLE) Center. The data is in many subfolders and not all of it is cleaned well, so it my task to take the data in this directory and clean it and make a big data frame. On that data frame, I will determine a structural equation model (SEM), network models of the degradation pathways using the netSEM package developd by the SDLE Center.

Part 1

Color Data

The first part is to clean the data, which is the bulk of the report. I chose to first clean the color data.

Loading Given Files

To begin, I load the key file, and the exposure files, making adjustments as needed.

```
key <- read.csv("./AcrylicHardcoats/acryhc-key.csv", sep = ",", header = TRUE)
exposures <- read.csv("./AcrylicHardcoats/acryhc-exp.csv", sep = ",", header = TRUE)
```

I notice that some of the variables are mislabeled, and we can have better names for other variables. So I do that cleaning.

```
exposures <- exposures %>%
  dplyr::rename(mASTMG154 = mASTG154) # This was labeled incorrectly in CSV
# Renaming the column names in the key file for easier names
key <- key %>%
  dplyr::rename(ID = Sample.Number, material = Product.Name, exposure = Exposure,
              retain = Step.Number.Retained)

# Let's try changing the material column to two columns,
# the second as substrate for better analysis

# Because the format is "XXXX-ABC"
rearrange.cols <- str_split_fixed(key$material, "-",2)
rearrange.cols <- as.data.frame(rearrange.cols)
names(rearrange.cols) <- list("substrate", "material")
# adding PURE level for no coating
levels(rearrange.cols$substrate) <- c(levels(rearrange.cols$substrate), "PURE")
for (i in 1:nrow(rearrange.cols)) { # iterate through all the samples and

  # check if samples are pure because material is 3 letters long if so.
  if(str_length(as.character(rearrange.cols$substrate[i])) <= 3) {
    # adjust the new substrate columns
    rearrange.cols$material[i] <- rearrange.cols$substrate[i]
    # plug in PURE if there is no coating as new substrate
    rearrange.cols$substrate[i] <- "PURE"
  }
}
key$material <- NULL # We want to overwrite the material column
key <- cbind(key, rearrange.cols) # merge the columns for an updated keyfile
# For Future use.
write.csv(key, "./AcrylicHardcoats/acryhc-key-updated.csv", row.names = FALSE)
```

Now, we want to compile all the color data. It is important to collect the data by ID and step as those are the two most important identifying pieces of data for each measurement that we need to keep track of. The rest comes from the key file.

```
## First we concatenate all the color data into a single data frame
## and make sure the column names are the same

# Reading in the Color Data for step 0
setwd("./AcrylicHardcoats/color/step0/")
filenames.step0 <- list.files("./", pattern = ".csv") # Find all files in the step0 folder
files.step0 <- lapply(filenames.step0, read.csv) # Reads each file into a list of dataframes
names(files.step0) <- filenames.step0

color.step0 <- NULL # New Data Frame for step0 color data
for (i in 1:length(filenames.step0)) {
  files.step0[[i]]$step <- 0 # Assign step to the data frame.
  color.step0 <- rbind(color.step0, files.step0[[i]])
}

# Reading in the Color Data for step 1
setwd("../step1/")
filenames.step1 <- list.files("./", pattern = ".csv")
files.step1 <- lapply(filenames.step1, read.csv)
names(files.step1) <- filenames.step1

color.step1 <- NULL
for (i in 1:length(filenames.step1)) {
 files.step1[[i]]$step <- 1
 color.step1 <- rbind(color.step1, files.step1[[i]])
}
```

This process is repeated for all of the steps, but I will not show it because it is redundant.

Combining all the intermediary datasets.

Finally I do some final edits, and remove intermediary data sets.

Note that there are incorrectly labeled sample IDs, which I chose to remove from the data entirely. Upon inspection, I found that there were only about 30 observations that were incorrectly labeled, so the work to rename each one was not worth preserving those data points (in my opinion). Therefore, those data points were just omitted from the models.

```r
color.binded <- rbind(color.step0, color.step1, color.step2,
                      color.step3, color.step4)
rm(color.step0, color.step1, color.step2, color.step3, color.step4)
rm(files.step0,files.step1, files.step2, files.step3, files.step4)
rm(filenames.step0, filenames.step1, filenames.step2,
   filenames.step3, filenames.step4)
rm(rearrange.cols)

## Removing values with incorrectly labeled samples
color.binded <- color.binded %>%
  dplyr::rename(lstar = L., astar = a., bstar = b.,
                YI = YI.E313..D65.10., Haze = Haze...D65.10) %>%
  dplyr::filter(nchar(as.character(ID)) == 10)

setwd("./AcrylicHardcoats/data/")

# This removes samples that are size 10 but do not exist in the key file
color.all <- merge(key, color.binded, by.x = "ID", all.x = TRUE)

# There are some entries where the baseline is measured again in step 4
# But there's no extra dose so we will change the step value to 0
# To not affect the analysis
for (i in 1:nrow(color.all)) {
  if (color.all$retain[i] == 0)
    color.all$step[i] <- 0
}
write.csv(color.all, file = "compiledColorData.csv", row.names = FALSE)
```

After this last chunk, I finally have all the color data compiled into one data frame, which I then save as a csv so that I can access it later without having to rely on my workspace to get it.

FTIR Data

Next, I want to incorporate the FTIR Data into the data frame. To do this, I must find only the peak information necessary, because to include every single peak would not be useful for the model we are trying to build. If we are to correlate every single wavenumber, then that would be a viable option.

The process is really similar, I go through each step folder in the FTIR folder and make an aggregate data frame of all the peaks that we want.

```r
# We first load in all the data that we need to make the analysis easy for us.

setwd("./AcrylicHardcoats/")
key <- read.table("acryhc-key-updated.csv", sep = ",", header = TRUE)
setwd("./data/")
color.data <- read.table("compiledColorData.csv", header = TRUE, sep = ",")

# We read in the FTIR Data so we can do some cleaning on it.

setwd("../FTIR/step0")
spc.step0 <- read.table(file = "step0.csv", sep = ",", header = TRUE)

setwd("../step1")
filenames.step1 <- list.files(".")
```

```r
files.step1 <- lapply(filenames.step1, read.csv, sep = ",")
files.step1[[2]]$Wavenumber <- NULL
spc.step1 <- cbind(files.step1[[1]], files.step1[[2]])

setwd("../step2")
filenames.step2 <- list.files(".")
files.step2 <- lapply(filenames.step2, read.csv, sep = ",")
files.step2[[2]]$Wavenumber <- NULL
spc.step2 <- cbind(files.step2[[1]], files.step2[[2]])

setwd("../step3")
filenames.step3 <- list.files(".")
files.step3 <- lapply(filenames.step3, read.csv, sep = ",")
spc.step3.outdoor <- files.step3[[1]]
spc.step3.indoor <- files.step3[[2]]

setwd("../step4")
filenames.step4 <- list.files(".")
files.step4 <- lapply(filenames.step4, read.csv, sep = ",")
spc.step4 <- files.step4[[1]]
```

At this point, I have all the FTIR data loaded in the session. Now I just need to extract the absorbance at the peaks necessary. To do this, I make a peak finder function to help.

```r
# Function that finds the max value within a range of the given frequency
# Takes in the spectra, a wavenumber column, a given frequency, and an adjustable range.
# Returns the max absorbance (peak) and the wavenumber at which it occurs in a vector
peak_finder <- function(spccolumn, wavenumber, frequency, range = 40){
  # Find the closest value to the given frequency
  index <- which(abs(wavenumber - frequency) == min(abs(wavenumber - frequency)))
  # Find the range of wavenumbers
  indices <- (index - range/2):(index + range/2)

  # Find Index of Max Absorbance within the range of the peak
  absorbance.max <- max(spccolumn[indices])

  wavenumber.max <- wavenumber[which(spccolumn == absorbance.max)]
  return(c(absorbance.max, wavenumber.max))
}
```

And I apply this function in the right way to get all the peak data necessary for this data frame.

```r
# Set of peaks we need to find
peaks <- c(1250, 1700, 2900, 3350)

#Creating the data frame so we can use it for later
columns <- c("ID","step","ftir.1250", "ftir.1700", "ftir.2900", "ftir.3350",
             "ftir.1250.wavenum", "ftir.1700.wavenum", "ftir.2900.wavenum",
             "ftir.3350.wavenum")
df.peaks <- setNames(data.frame(matrix(ncol = length(columns), nrow = 0)), columns)
df.peaks[columns] <- sapply(df.peaks[columns], as.numeric)

# Wavenumber for step0-step3.indoor (checked after inspection)
wavenumber.long <- spc.step0$Wavenumber
# Wavenumber for step3.outdoor - step4
```

```r
wavenumber.short <- spc.step4$Wavenumber

# Iterate through all the steps in the data.
spc.step0$Wavenumber <- NULL # only iterate through the spc.step0 DATA
sample.IDs <- colnames(spc.step0)

for (i in 1:length(spc.step0[1,])){
  sample.ID <- as.character(sample.IDs[i])

  # Below, we store the actual peak value.
  ftir.1250 <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                           frequency = peaks[1])[1]
  ftir.1700 <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                           frequency = peaks[2])[1]
  ftir.2900 <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                           frequency = peaks[3])[1]
  ftir.3350 <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                           frequency = peaks[4])[1]

  # Here, we store the wavenumber at which the peak occurs
  ftir.1250.wavenum <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                                   frequency = peaks[1])[2]
  ftir.1700.wavenum <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                                   frequency = peaks[2])[2]
  ftir.2900.wavenum <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                                   frequency = peaks[3])[2]
  ftir.3350.wavenum <- peak_finder(spc.step0[,i], wavenumber = wavenumber.long,
                                   frequency = peaks[4])[2]

  step <- 0

  # Each observation as a row in the big data frame, and we update.
  each.observation <- list(sample.ID, step, as.numeric(ftir.1250),
                           as.numeric(ftir.1700), as.numeric(ftir.2900),
                           as.numeric(ftir.3350), as.numeric(ftir.1250.wavenum),
                           as.numeric(ftir.1700.wavenum),
                           as.numeric(ftir.2900.wavenum),
                           as.numeric(ftir.3350.wavenum))
  df.peaks[nrow(df.peaks) + 1,] <- each.observation
}
```

Again, only step 0 is shown as it's redundant to show all 4 steps, so they will be hidden.

Finally, we merge the FTIR peak data with the color data to update our data frame.

```r
# Here we are merging the color data and the FTIR data to make the full, cleaned
# Data frame for analysis later on.
colorPlusFTIR <- merge(color.data, df.peaks, by = c("ID", "step"), all.x = TRUE)
setwd("./AcrylicHardcoats/data/")
write.csv(df.peaks, file = "ftirPeakData.csv", row.names = FALSE)
write.csv(colorPlusFTIR, file = "colorPlusFTIR.csv", row.names = FALSE)
```

Converting the Exposure to a Dose Basis

We don't just want to look at the number of hours a sample was exposed, what's important is how much energy was given to the sample to cause degradation. To do that, we must convert to a dose basis.

First, we set up the data and load it in as needed to make sure we do this right. The first thing being done is to make sure the exposure file is converted into a cumulative sum instead of number of hours per step, as the cumulative sum is more important in determining how much energy was put into the sample.

```
setwd("./AcrylicHardcoats/data/")
colorPlusFTIR <- read.csv("colorPlusFTIR.csv", header = TRUE, sep = ",")

setwd("../")
exposures <- read.csv("acryhc-exp.csv", header = TRUE, sep = ",")

# Replace the per step hour time with a cumulative sum
# So the data frame reads "at step 3, sample SA was exposed for x hours"
# Start at 2 to skip the step column (we don't want to accumulate the number of steps)
for (i in 2:ncol(exposures)){
  exposures[,i] <- cumsum(exposures[,i])
}

# Rename the columns for more readable variable names,
# and gather the exposures into a single column with tidyr
exposures <- exposures %>%
  dplyr::rename(step = Steps, mASTMG154 = mASTG154, '1x' = X1x, '5x' = X5x) %>%
  tidyr::gather(exposure, hours, baseline:'5x', na.rm = TRUE)

# Merge the exposure file with previously cleaned data
dat.0 <- merge(exposures, colorPlusFTIR, by = c("exposure", "step"), na.rm = FALSE,
               all.y = TRUE)
```

Here we make the actual data frame of doses. The values are taken from the information file where the different exposures are cross-correlated to a specific dose value. The values are then stored into a data frame in the following way. Then the final dat frame is built by merging everything together.

```
# All Doses are in MegaJoules per meter^2
# This data frame contains the dose and doseType value for each exposure type
doses <- data.frame(c('mASTMG154', 1,   0.21835, 'uva340'),
                    c('ASTMG154',  1,   8/12*0.21835, 'uva340'), #4 hours of dark
                    c('ASTMG155',  1,   .09382, 'uva340'),
                    c('1x',        1,   .04599, 'fullSpectrum'),
                    c('5x',       1,   NA, 'fullSpectrum'),
                    c('baseline', 1,  0,         NA),
                    c('HF',       1,  NA,        NA))
doses <- as.data.frame(t(doses))
names(doses) <- list('exposure', 'hours', 'dose', 'doseType')
# The next four lines are needed to remove any funky business from R
# and to keep a clean data frame.
setwd("./AcrylicHardcoats/data/")
write.csv(doses, 'doses.csv', row.names = FALSE)
rm(doses)
doses <- read.csv('doses.csv', header = TRUE, sep = ",")

# Here we initialize the dose and doseType columns of the big data frame
dat.0$dose <- NA
dat.0$doseType <- NA
# Now it's time to add the doses to the big data frame!
for (i in 1:nrow(dat.0)) {
  exp.specific <- as.character(dat.0$exposure[i]) # Find the exposure of this obs

  # Find the index of the corresponding exposure in the doses data frame.
  which.dose <- which(as.character(doses$exposure) == exp.specific)
  dose.onehour <- doses$dose[which.dose] # Find the dose value per hour of exposure
  # Enter in the total dose by multiplying hours of exposure by dose per hour
  dat.0$dose[i] <- dat.0$hours[i]*dose.onehour*doses$hours[which.dose]
  # Enter in the doseType to the big data frame
  dat.0$doseType[i] <- as.character(doses$doseType[which.dose])
  # change to factor because R is funky!
  dat.0$doseType <- factor(dat.0$doseType, levels = c("uva340", 'tuv280400', 'fullSpectrum'))
}

# Finally, we notice exposure became first column, but we want ID to be first
# So we switch them.
dat.0[,c("exposure","ID")] <- dat.0[,c("ID","exposure")]
dat.0 <- dat.0 %>%
  dplyr::rename(ID = exposure, exposure = ID)

data.final <- dat.0
rm(dat.0)
write.csv(data.final, "completeDataFrame.csv", row.names = FALSE)
```

After doing this, the final data frame is finally created and stored as a csv. We can remove everything from the workspace and just load the important data sets to free up space.

```
rm(list = ls())
data.all <- read.table("./AcrylicHardcoats/data/completeDataFrame.csv", sep = ",",
                       header = TRUE)
```

Questions and Answers

Question 1: What are the dimensions of your data frame?

```
# Base function that returns the dimension of an object
dim(data.all)
```

```
## [1] 804  22
```

Answer 1: The dimension of the data frame is 804 by 22.

Question 2: Show the Head and Tail of your Data Frame:

```
head(data.all)
```

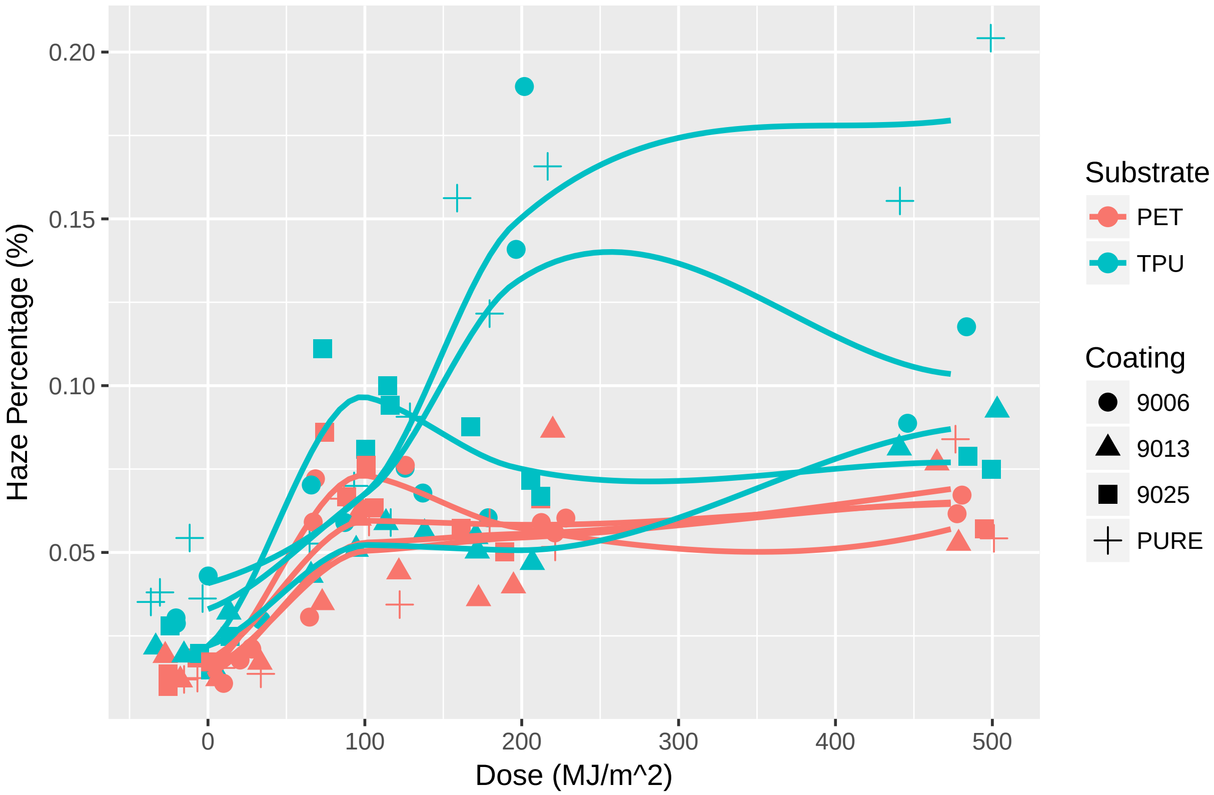
```
##           ID step hours exposure retain substrate material lstar astar
## 1 sa22076.02    0     0       1x      2      9013      PET 95.88 -0.04
## 2 sa22073.01    0     0       1x      1      9006      TPU 96.71 -0.05
## 3 sa22085.02    0     0       1x      2      9025      TPU 96.78 -0.03
## 4 sa22079.02    0     0       1x      2      9013      TPU 96.76 -0.04
## 5 sa22081.02    0     0       1x      2      9025      PET 95.81 -0.18
## 6 sa22085.01    0     0       1x      1      9025      TPU 96.81 -0.03
##   bstar   YI Haze ftir.1250 ftir.1700 ftir.2900 ftir.3350
## 1  1.14 2.13  1.2        NA        NA        NA        NA
## 2  0.36 0.63  2.9        NA        NA        NA        NA
## 3  0.34 0.60  2.5        NA        NA        NA        NA
## 4  0.36 0.63  2.2        NA        NA        NA        NA
## 5  0.95 1.66  1.8        NA        NA        NA        NA
## 6  0.38 0.67  2.8        NA        NA        NA        NA
##   ftir.1250.wavenum ftir.1700.wavenum ftir.2900.wavenum ftir.3350.wavenum
## 1                NA                NA                NA                NA
## 2                NA                NA                NA                NA
## 3                NA                NA                NA                NA
## 4                NA                NA                NA                NA
## 5                NA                NA                NA                NA
## 6                NA                NA                NA                NA
##   dose     doseType
## 1    0 fullSpectrum
## 2    0 fullSpectrum
## 3    0 fullSpectrum
## 4    0 fullSpectrum
## 5    0 fullSpectrum
## 6    0 fullSpectrum
```

```r
tail(data.all)
```

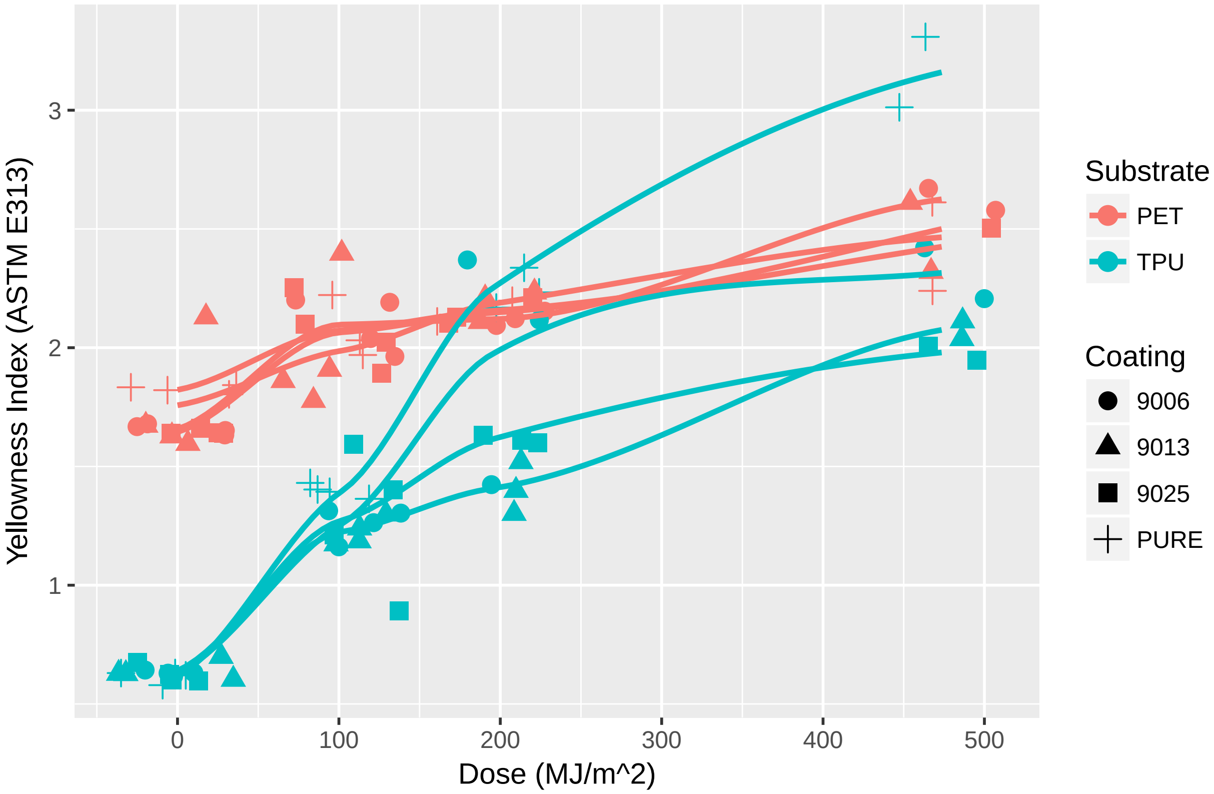
```
##             ID step hours  exposure retain substrate material lstar astar
## 799 sa22077.04    4  9513 mASTMG154      4      9013      TPU 96.31 -0.01
## 800 sa22088.05    4  9513 mASTMG154      4      PURE      TPU 96.55  0.04
## 801 sa22080.04    4  9513 mASTMG154      4      9025      PET 94.00 -0.52
## 802 sa22077.05    4  9513 mASTMG154      4      9013      TPU 96.56  0.02
## 803 sa22071.04    4  9513 mASTMG154      4      9006      TPU 96.28 -0.01
## 804 sa22083.05    4  9513 mASTMG154      4      9025      TPU 96.66  0.05
##     bstar   YI Haze ftir.1250 ftir.1700 ftir.2900 ftir.3350
## 799  0.47 0.87 13.8  0.514005  0.626885 0.0884927 0.0987322
## 800  0.45 0.86  5.5  0.394757  0.443066 0.1541110 0.0560826
## 801  2.86 5.06 45.2  0.162470  0.147467 0.0335728 0.0258007
## 802  0.40 0.75  9.5  0.494839  0.601998 0.0860713 0.0907124
## 803  0.50 0.92  5.8  0.412927  0.490545 0.1455800 0.0596604
## 804  0.43 0.83  4.1  0.419461  0.468163 0.1007050 0.0666429
##     ftir.1250.wavenum ftir.1700.wavenum ftir.2900.wavenum
## 799           1213.61           1737.45                NA
## 800           1213.61           1737.45                NA
## 801           1213.61           1735.59                NA
## 802           1213.61           1737.45                NA
## 803           1213.61           1737.45                NA
## 804           1213.61           1737.45                NA
##     ftir.3350.wavenum     dose doseType
## 799                NA 2077.164   uva340
## 800                NA 2077.164   uva340
## 801                NA 2077.164   uva340
## 802                NA 2077.164   uva340
## 803                NA 2077.164   uva340
## 804                NA 2077.164   uva340
```

Question 3: Plot the YI and Haze as a function of Dose for each material for the 1x exposure. do you notice any differences between substrates and the coatings on the substrates?

Haze for 1x Outdoor Samples against Dose



Yellowness Index for 1x Outdoor Samples against Dose



Answer 3:

For the Haze data, we see that there is more of a spread in the change in Haze percentage by coating in the TPU substrate samples. Their haze also increases generally higher than the PET samples. Although the Haze in the PET samples do increase, it is not nearly as much as the TPU samples and the coating does not seem to have an effect on the increase of the Haze PET samples. It seems that the coating does have a mitigatin effect on the degradation of TPU, evidenced by the fact that the PURE TPU has a substantially large Haze % increase and the coated samples increase less.

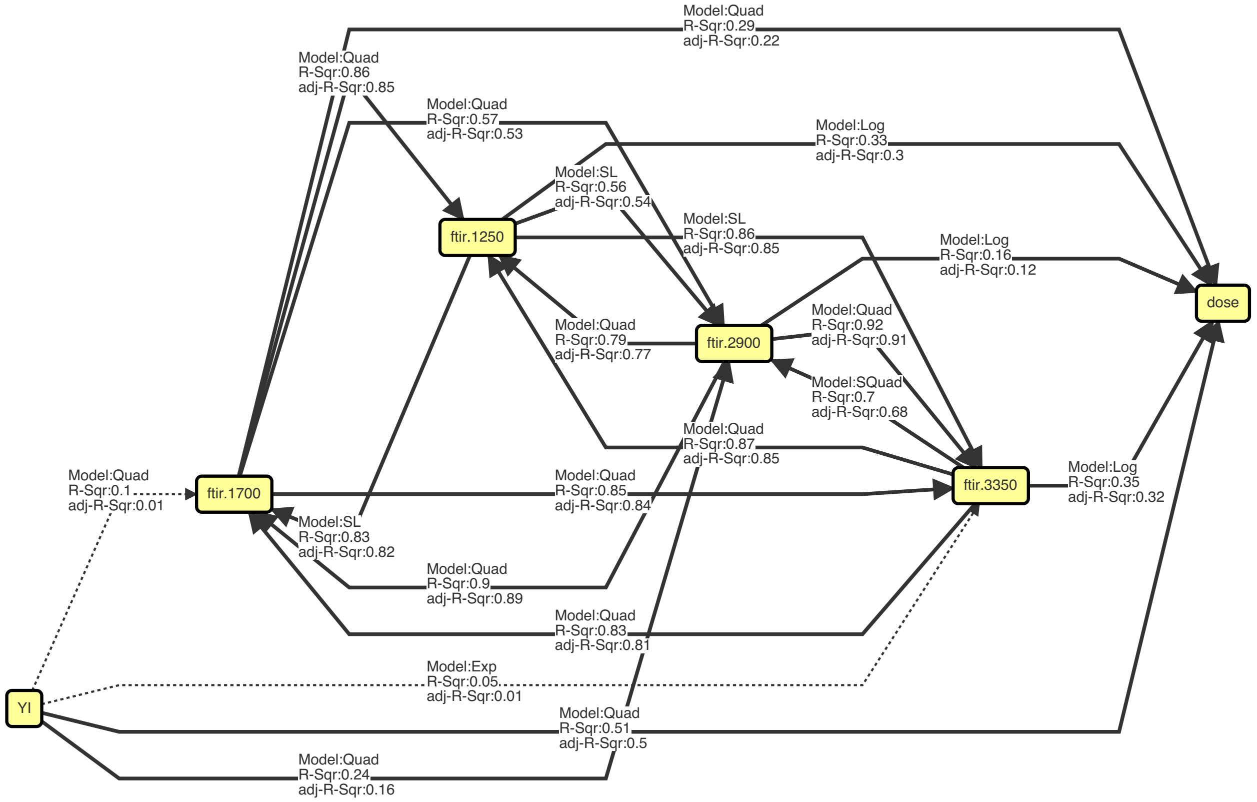
For the YI data, we see some similar results. To start off, the PET samples are more yellow than the TPU samples. However, the increase in Yellowness is much greater for the TPU samples than the PET samples. Further, similar to what was shown in the Haze plot, there is much more increase in Yellowness for the PURE TPU samples than the coated TPU samples, indicating the coating does have a mitigating effect of degradation on the TPU samples. However, the coating seems to have no effect on the degradation of the PET samples.

Part 2: netSEM Modeling

Question 1

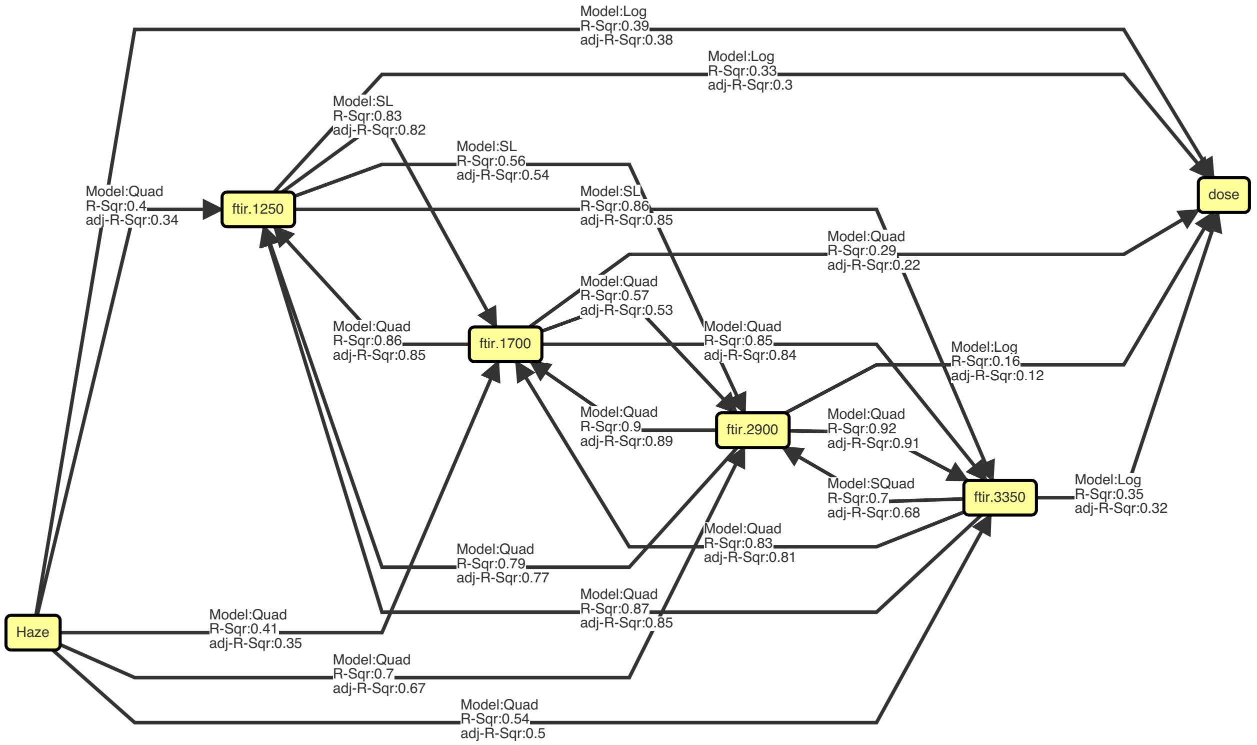
Here, we input our data frame into the netSEM model to make a network response model for the different mechanisms at play. As the Introduction file states, Irradiance on a dose basis is the stressor, YI or Haze is the response, and the FTIR peak values are the mechanistic variables. We filter, input the data and create the model as follows.

```r
model.YI1x <- data.all %>%
  dplyr::filter(exposure == '1x') %>%
  dplyr::select(c('dose', 'YI', 'ftir.1250', 'ftir.1700', 'ftir.2900', 'ftir.3350')) %>%
  netSEMm()
model.haze1x <- data.all %>%
  dplyr::filter(exposure == '1x') %>%
  dplyr::select(c('dose', 'Haze', 'ftir.1250', 'ftir.1700', 'ftir.2900', 'ftir.3350')) %>%
  netSEMm()
plot(model.YI1x, cutoff = 0.1)
```



```r
plot(model.haze1x, cutoff = 0.1)
```

```
##
## The cutoff value is lower than all of the adjusted R-sqr values: Only solid lines
```

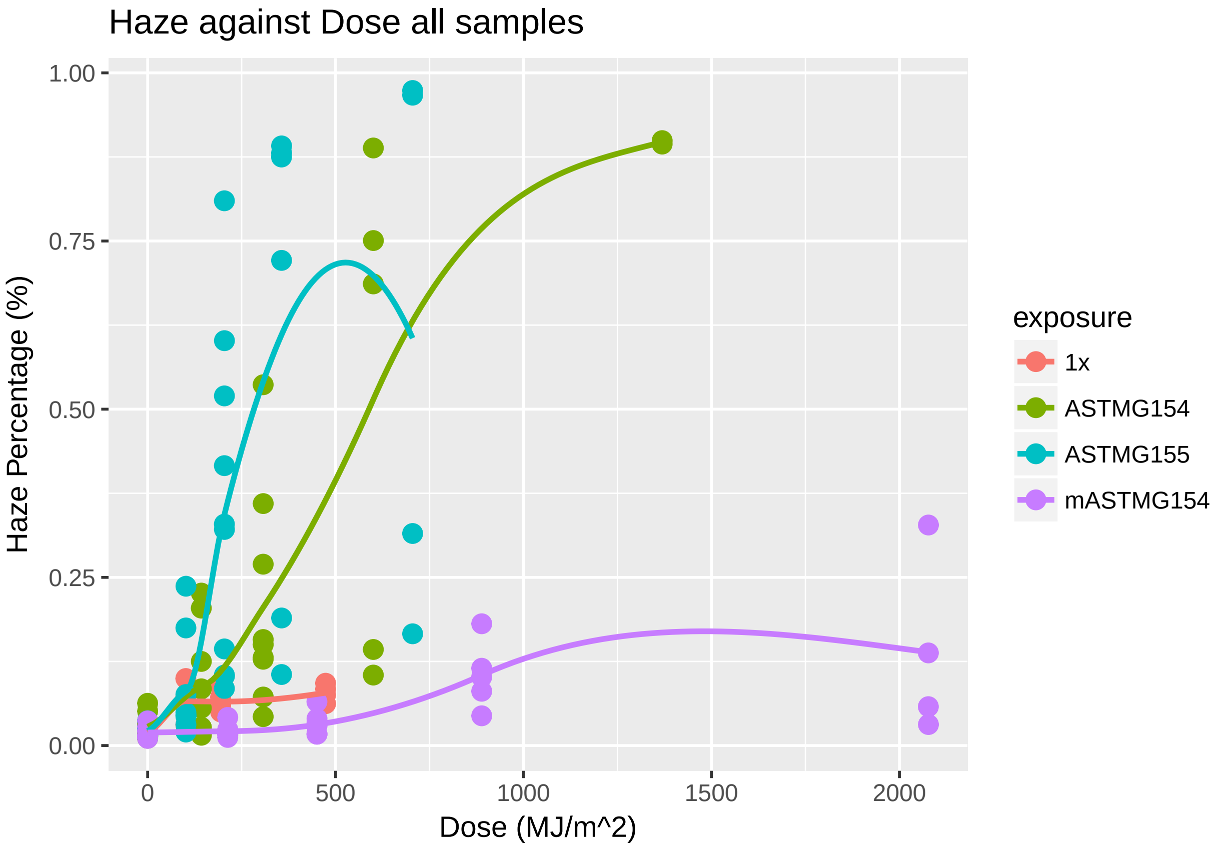


These are just two network models, where the first one relates YI to Dose through the FTIR Peak values for the 1x outdoor exposures and the second relates Haze % to Dose for the same samples. In each model you can see the Rˆ2 and adjusted Rˆ2 values, which are measures of how well the model relates the two variables. It also says what kind of model was used to calculate these values. Quad means quadratic, log means logarithmic, and so on.

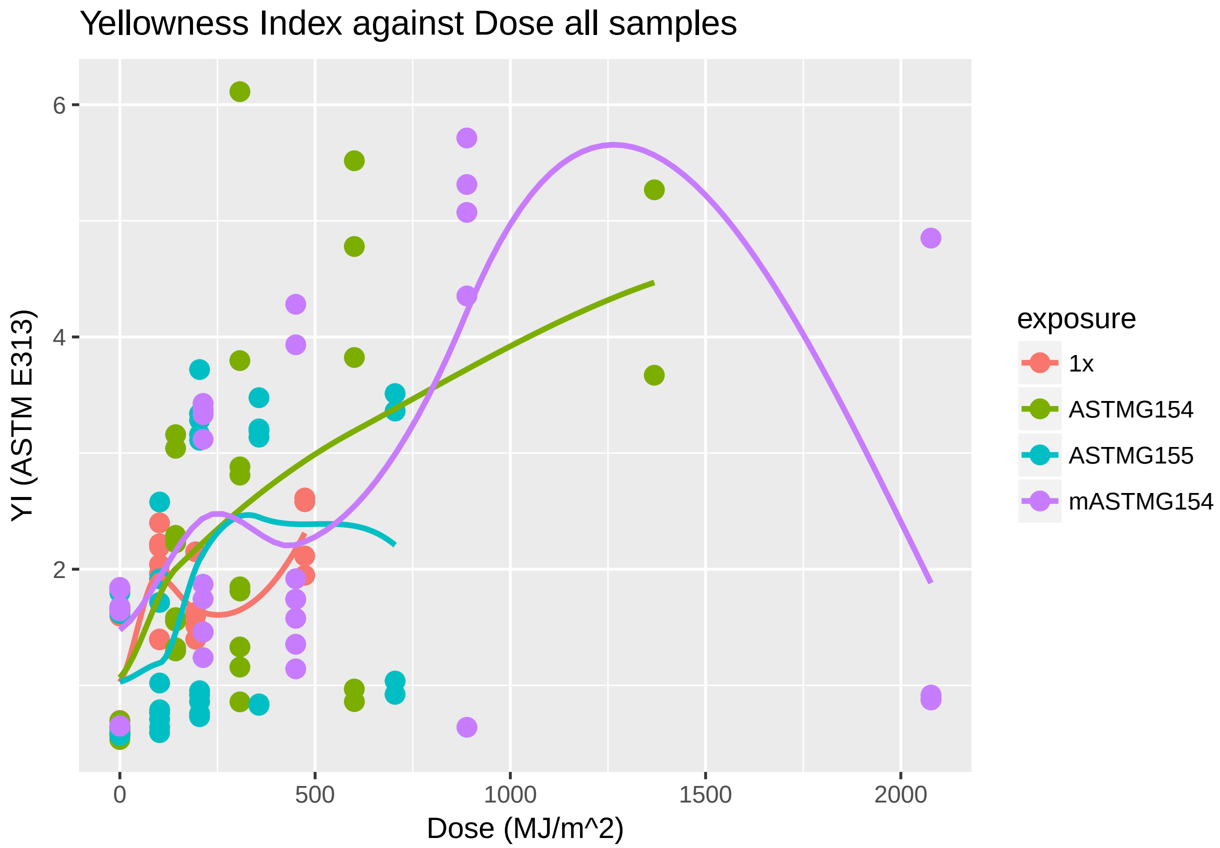
Question 2

Now, let's pick which indoor exposure is most related to 1x outdoor exposure.

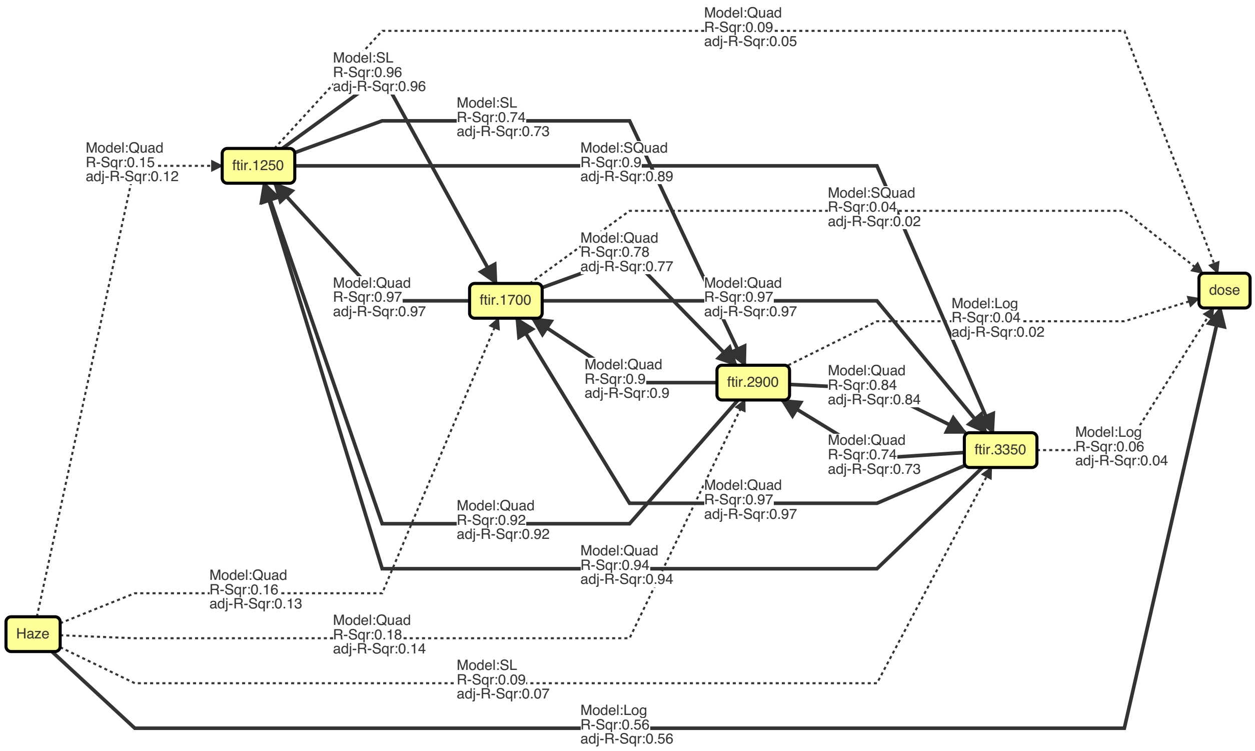
```
## `geom_smooth()` using method = 'loess'
```

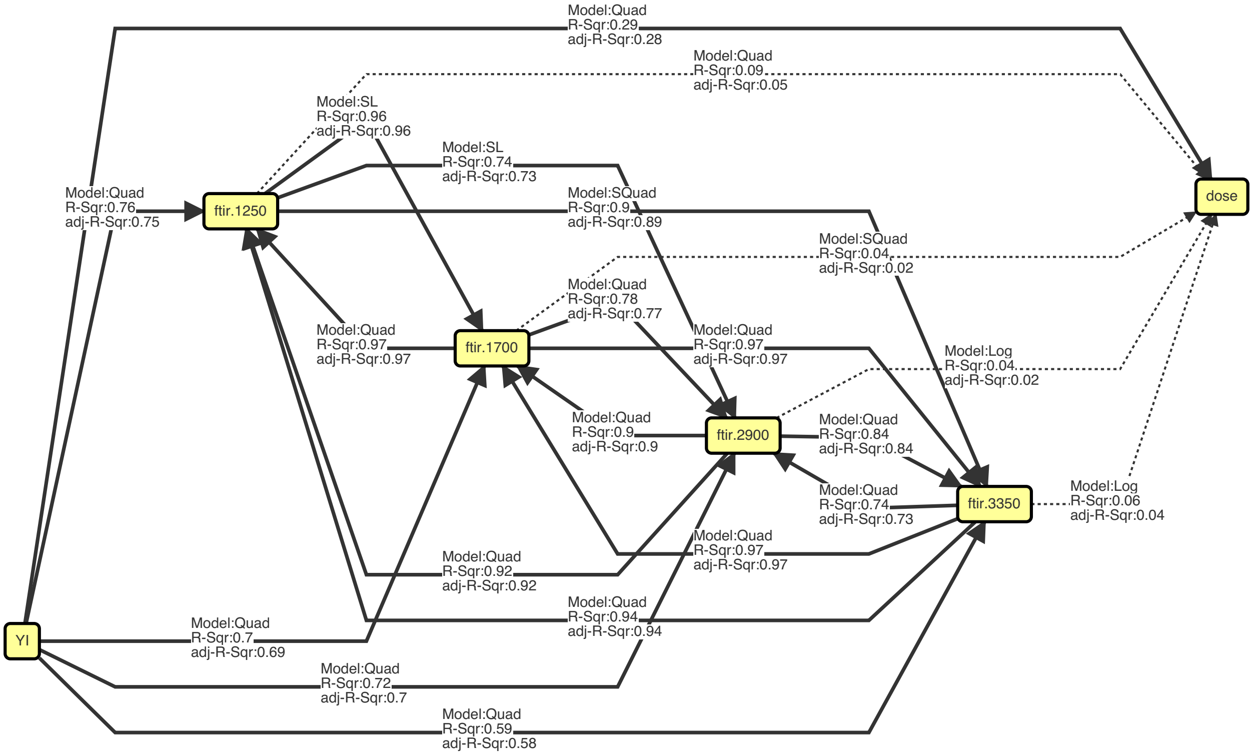


```
## `geom_smooth()` using method = 'loess'
```



Looking at the shapes of the two curves, it seems that 1x is most similar to mASTMG154 exposure. So, we will make a netSEM model for the mASTMG154 exposed samples with YI and Haze as responses, dose as stressor, and ftir peak values as mechanisms.





Upon inspection, you can see that the Haze to Dose model is better correlated than the YI to Dose model. These models help us understand how each of these measurements are related, and organizes all the relationships between stressors, mechanisms, and responses for readers to understand what's going on in mathematical models at a broader level. Further, the R^2 value helps you understand which exposures are more consistent in their degradation of a given material. For this example, you can see that the Haze % and dose are correlated with R^2 = 0.56, which means although it is not a great model, allows you to see that this exposure highlights the Haze response more than the 1x outdoor samples, which had an R^2 = 0.39.