# CWRU DSCI351: Project 3

*Anish Mitra*

*November 14, 2018*

## Contents



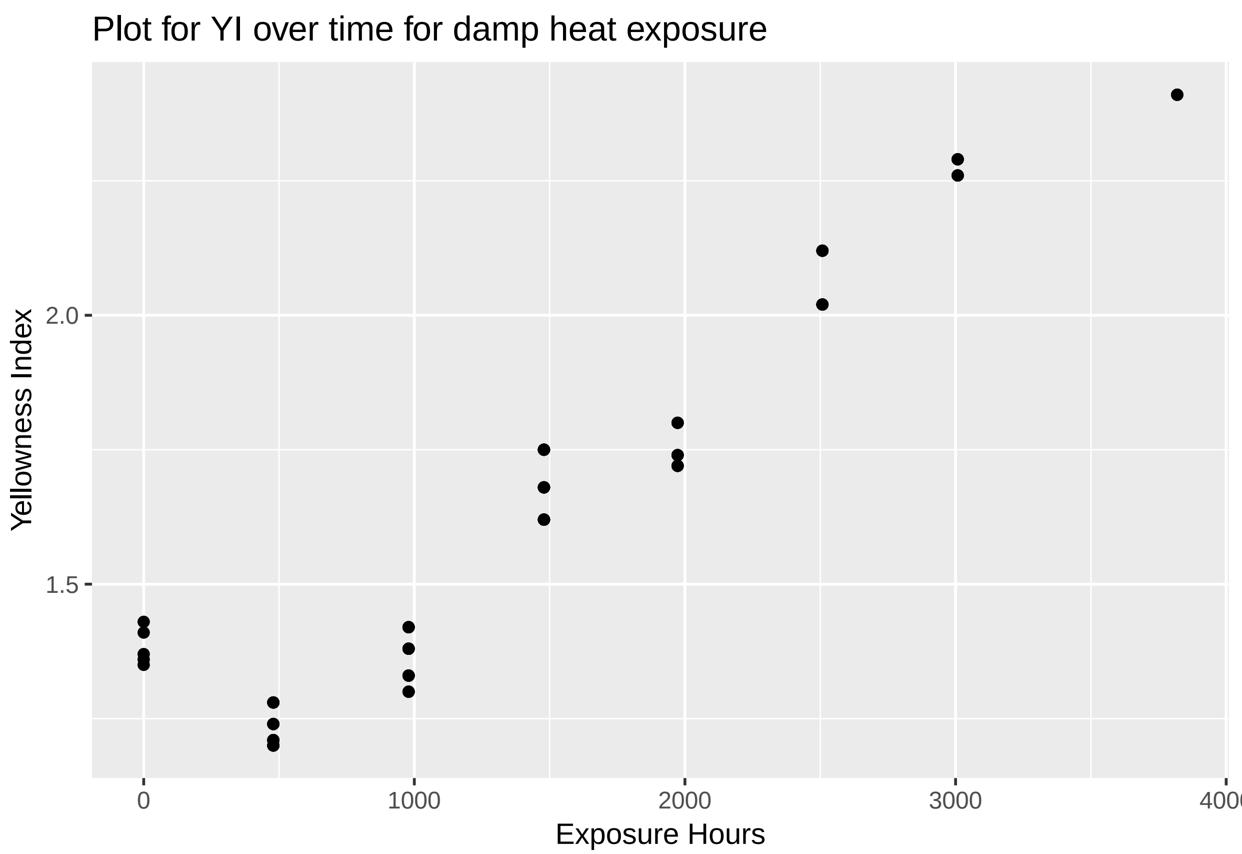
# 2 Project 3: Accelerated Ageing of PV Backsheets

## 2.1 1. Visualize the YI and gloss of one of the samples over time, what trends do you notice?

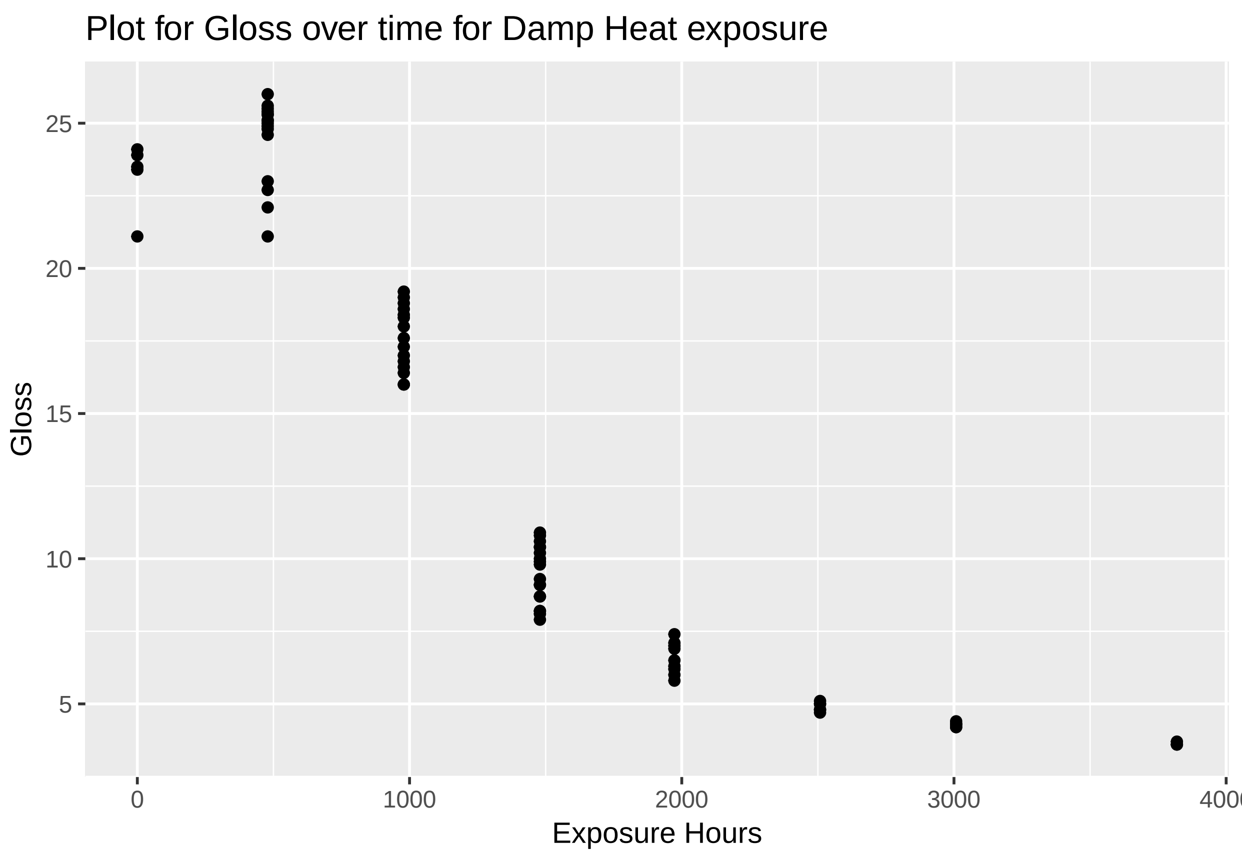
```r
# Loading useful packages from R
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(hyperSpec))
suppressPackageStartupMessages(library(segmented))
suppressPackageStartupMessages(library(MASS))
suppressPackageStartupMessages(library(lme4))
suppressPackageStartupMessages(library(segmented))

# Reading in the data
optivaldata <- data.table::fread("H:/Git/18f-dsci351-351m-451-axm949/1-assignments/proj/proj3/DSCI-352-

# Defining an array which contains the different exposure types
exposures <- unique(optivaldata$Exposure_Type)
```

```r
# Picking the sampleID sa31001 in Damp Heat visualization

# Plotting for YI over time for damp heat exposure
optivaldata %>%
  filter(SampleID == "sa31005") %>%
  filter(Exposure_Type == "Xenon#1") %>%
```

```r
  ggplot(aes(y = Yellowness_Index,
             x = Exposure_Hours)) +
  geom_point() +
  ggtitle("Plot for YI over time for damp heat exposure") +
  xlab("Exposure Hours") + ylab("Yellowness Index")
```




```r
# From the graph, the yellowness index appears to be increasing linearly over time.

# Plot for Gloss over time for damp heat exposure
optivaldata %>%
  filter(SampleID == "sa31005") %>%
  filter(Exposure_Type == "Xenon#1") %>%
  ggplot(aes(x = Exposure_Hours, y = Gloss_60degree)) +
  geom_point() +
  ggtitle("Plot for Gloss over time for Damp Heat exposure") +
  xlab("Exposure Hours") + ylab("Gloss")
```




```
# From the graph, the gloss appears to be decreasing non-linearly over time.
```

## 2.2 2. How do YI and gloss relate to the material degradation?

We know that polymers degrade and become yellow due to light or chemical exposure, it is the Yellowness Index that is used to act as a degradation indicator of polymers. This means increasing the yellowness index would be indicative of degradation.

We are also given that the smoother the surface, the larger the gloss value, and this means that the polymers are presumed to become more and more coarse during the degradation process. So decreasing gloss would be indicatice of increasing coarseness, which is indicative of degradation. This means decreasing the gloss would be indivative of degradation.

## 2.3 3. Develop degradation models

### 2.3.1 Develop a linear model for each material with exposure to each condition.

```
#For Yellowness Index,

# This creates an array with all the samples
samples <- unique(optivaldata$SampleID)
# This loop will build linear models for the yellowness index of all the samples at all exposures

# These arrays also store all the slopes for all the models
```

```
dampheatyislopes <- rep(NULL, length(samples))
xenon1yislopes <- rep(NULL, length(samples))
xenon2yislopes <- rep(NULL, length(samples))


for(i in exposures){
  for (j in samples) {
    lm.fit <-
  lm(Yellowness_Index ~ Exposure_Hours,
     data = optivaldata %>%
  filter(SampleID == j) %>%
  filter(Exposure_Type == i))
    assign(paste("lm.fit", j, i, sep = "."), lm.fit)
    if (i == exposures[1]){
      dampheatyislopes[which(samples == j)] <-
        coef(lm.fit)[[2]]
    } else if (i == exposures[2]){
      xenon1yislopes[which(samples == j)] <-
        coef(lm.fit)[[2]]
    } else if (i == exposures[3]){
      xenon2yislopes[which(samples == j)] <-
        coef(lm.fit)[[2]]
    }
  }
}

# For gloss,

# This loop will build linear models for the gloss of all the samples at all exposures

# These arrays also store all the slopes for all the models

dampheatglossslopes <- rep(NULL, length(samples))
xenon1glossslopes <- rep(NULL, length(samples))
xenon2glossslopes <- rep(NULL, length(samples))
for(i in exposures){
  for (j in samples) {
    lm.fit <-
  lm(Gloss_60degree ~ Exposure_Hours,
     data = optivaldata %>%
  filter(SampleID == j) %>%
  filter(Exposure_Type == i))
    assign(paste("lm.fit", j, i, "gloss", sep = "."), lm.fit)
    if (i == exposures[1]){
      dampheatglossslopes[which(samples == j)] <-
        coef(lm.fit)[[2]]
    } else if (i == exposures[2]){
      xenon1glossslopes[which(samples == j)] <-
        coef(lm.fit)[[2]]
    } else if (i == exposures[3]){
      xenon2glossslopes[which(samples == j)] <-
        coef(lm.fit)[[2]]
    }
```

```
  }
}
```

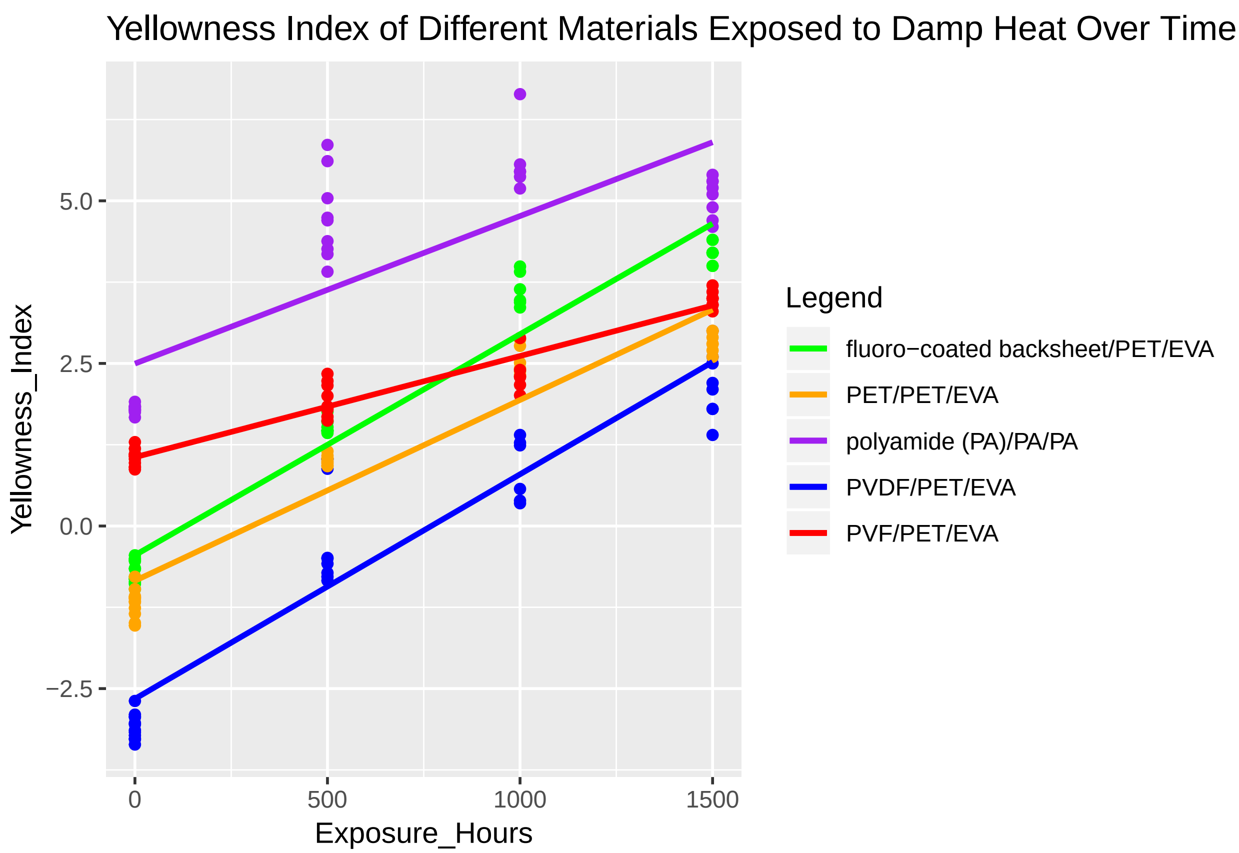
### 2.3.2 For each exposure, plot the data and linear model for all the materials on one plot.

Create data frames for each material for each exposure.

```
sampledata <- read.csv("H:/Git/18f-dsci351-351m-451-axm949/1-assignments/proj/proj3/DSCI-352-optival_da
#sampleID sa31001
test <- sampledata %>%
  filter(SampleID == "sa31001") %>%
  dplyr::select(Exposure_Type, Exposure_Hours, Yellowness_Index, Gloss_60degree)
dampheatsa31001 <- test[test$Exposure_Type=="DampHeat", 1:4]
xenon1sa31001 <- test[test$Exposure_Type=="Xenon#1", 1:4]
xenon2sa31001 <- test[test$Exposure_Type=="Xenon#2", 1:4]

#sampleID sa31002
test2 <- sampledata %>%
  filter(SampleID == "sa31002") %>%
  dplyr::select(Exposure_Type, Exposure_Hours, Yellowness_Index, Gloss_60degree)
dampheatsa31002 <- test2[test2$Exposure_Type=="DampHeat", 1:4]
xenon1sa31002 <- test2[test2$Exposure_Type=="Xenon#1", 1:4]
xenon2sa31002 <- test2[test2$Exposure_Type=="Xenon#2", 1:4]

#sampleID sa31003
test3 <- sampledata %>%
  filter(SampleID == "sa31003") %>%
  dplyr::select(Exposure_Type, Exposure_Hours, Yellowness_Index, Gloss_60degree)
dampheatsa31003 <- test3[test3$Exposure_Type=="DampHeat", 1:4]
xenon1sa31003 <- test3[test3$Exposure_Type=="Xenon#1", 1:4]
xenon2sa31003 <- test3[test3$Exposure_Type=="Xenon#2", 1:4]

#sampleID sa31004
test4 <- sampledata %>%
  filter(SampleID == "sa31004") %>%
  dplyr::select(Exposure_Type, Exposure_Hours, Yellowness_Index, Gloss_60degree)
dampheatsa31004 <- test4[test4$Exposure_Type=="DampHeat", 1:4]
xenon1sa31004 <- test4[test4$Exposure_Type=="Xenon#1", 1:4]
xenon2sa31004 <- test4[test4$Exposure_Type=="Xenon#2", 1:4]

#sampleID sa31005
test5 <- sampledata %>%
  filter(SampleID == "sa31005") %>%
  dplyr::select(Exposure_Type, Exposure_Hours, Yellowness_Index, Gloss_60degree)
dampheatsa31005 <- test5[test5$Exposure_Type=="DampHeat", 1:4]
xenon1sa31005 <- test5[test5$Exposure_Type=="Xenon#1", 1:4]
xenon2sa31005 <- test5[test5$Exposure_Type=="Xenon#2", 1:4]
```

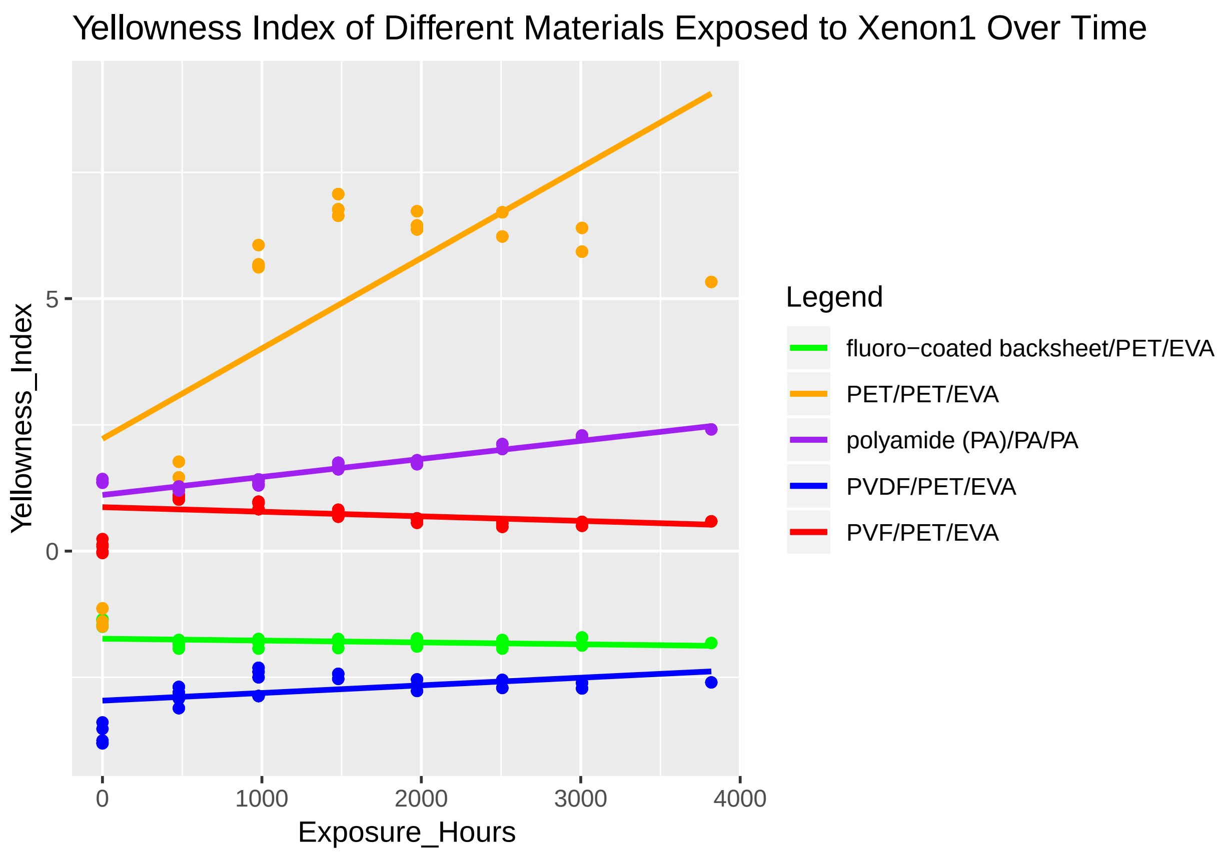
```
# FOr YI,

# Damp heat exposure
ggplot(data = dampheatsa31001, aes(x = Exposure_Hours, y = Yellowness_Index)) + geom_point(col = "blue")
aes(x = Exposure_Hours, y = Yellowness_Index),
```

```
col = "green") + geom_point(data = dampheatsa31003,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "orange") + geom_point(data = dampheatsa31004,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "red") + geom_point(data = dampheatsa31005,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "purple") + ggtitle("Yellowness Index of Different Materials Exposed to Damp Heat Over Time") + s
data = dampheatsa31002,
method = "lm",
se = FALSE,
aes(color = "fluoro-coated backsheet/PET/EVA")
) + stat_smooth(
data = dampheatsa31001,
method = "lm",
se = FALSE,
aes(color = "PVDF/PET/EVA")
) + stat_smooth(
data = dampheatsa31003,
method = "lm",
se = FALSE,
aes(color = "PET/PET/EVA")
) + stat_smooth(
data = dampheatsa31004,
method = "lm",
se = FALSE,
aes(color = "PVF/PET/EVA")
) + stat_smooth(
data = dampheatsa31005,
method = "lm",
se = FALSE,
aes(color = "polyamide (PA)/PA/PA")
) + scale_colour_manual(name = "Legend",
values = c("green", "orange", "purple", "blue", "red"))
```



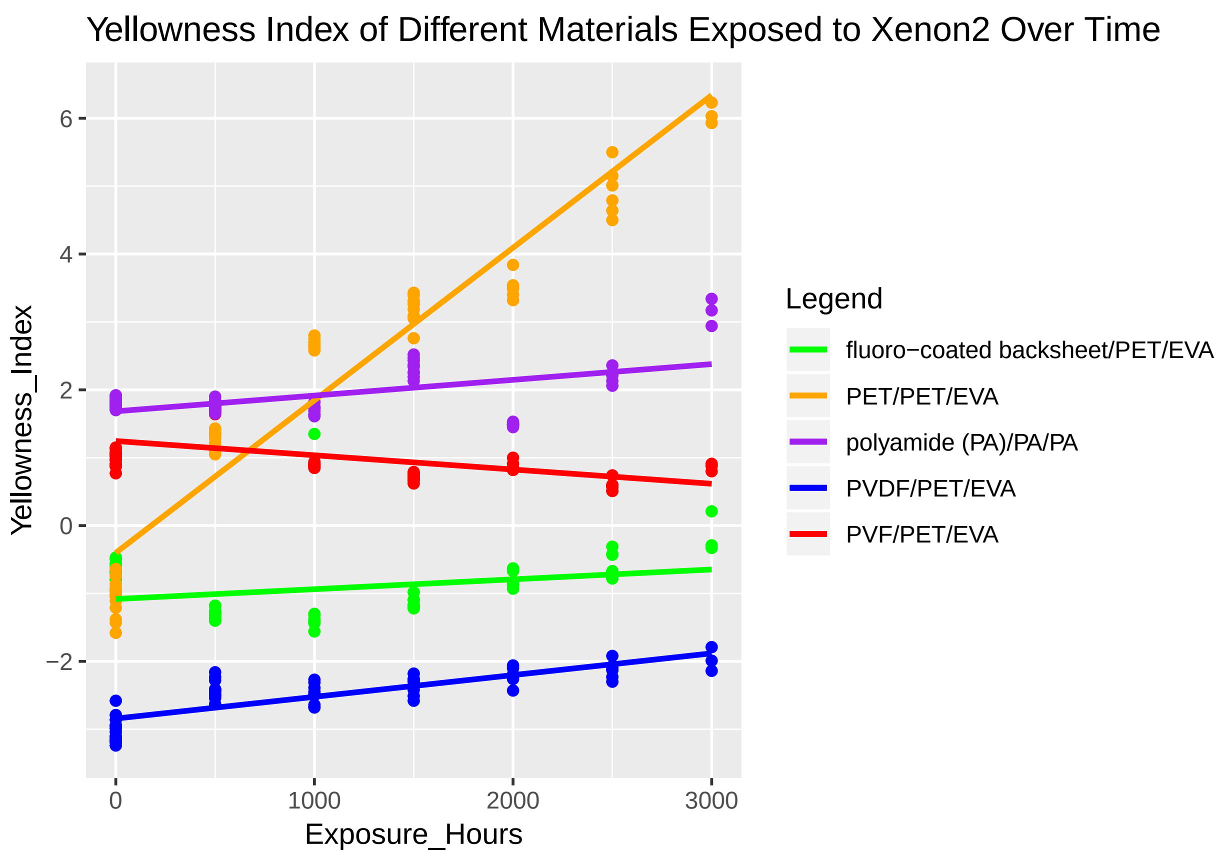

```
# Xenon1 exposure
ggplot(data = xenon1sa31001, aes(x = Exposure_Hours, y = Yellowness_Index)) + geom_point(col = "blue")
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "green") + geom_point(data = xenon1sa31003,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "orange") + geom_point(data = xenon1sa31004,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "red") + geom_point(data = xenon1sa31005,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "purple") + ggtitle("Yellowness Index of Different Materials Exposed to Xenon1 Over Time") + stat_
data = xenon1sa31002,
method = "lm",
se = FALSE,
aes(color = "fluoro-coated backsheet/PET/EVA")
) + stat_smooth(
data = xenon1sa31001,
method = "lm",
se = FALSE,
aes(color = "PVDF/PET/EVA")
) + stat_smooth(
data = xenon1sa31003,
method = "lm",
se = FALSE,
aes(color = "PET/PET/EVA")
) + stat_smooth(
data = xenon1sa31004,
```

```
method = "lm",
se = FALSE,
aes(color = "PVF/PET/EVA")
) + stat_smooth(
data = xenon1sa31005,
method = "lm",
se = FALSE,
aes(color = "polyamide (PA)/PA/PA")
) + scale_colour_manual(name = "Legend",
values = c("green", "orange", "purple", "blue", "red"))
```



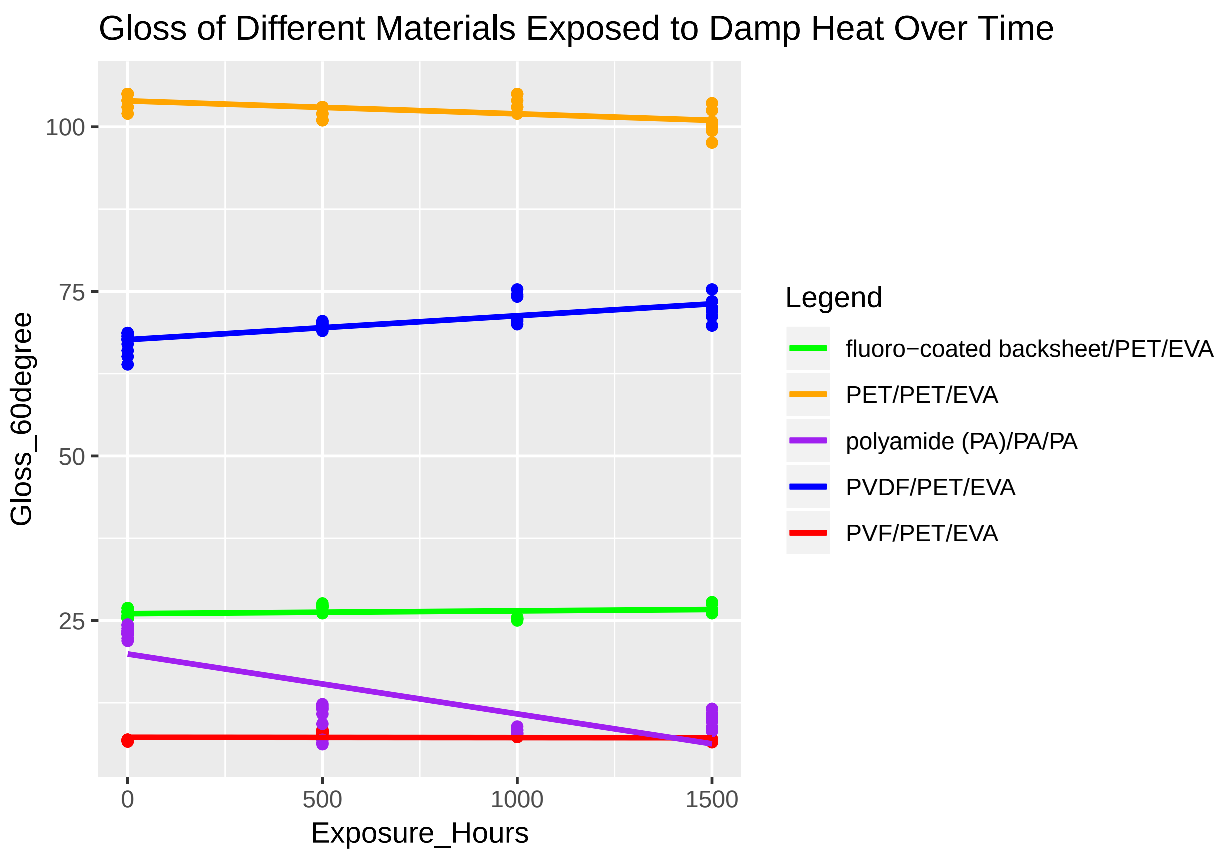

```
# Xenon2 exposure

ggplot(data = xenon2sa31001, aes(x = Exposure_Hours, y = Yellowness_Index)) + geom_point(col = "blue")
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "green") + geom_point(data = xenon2sa31003,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "orange") + geom_point(data = xenon2sa31004,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "red") + geom_point(data = xenon2sa31005,
aes(x = Exposure_Hours, y = Yellowness_Index),
col = "purple") + ggtitle("Yellowness Index of Different Materials Exposed to Xenon2 Over Time") + stat_
data = xenon2sa31002,
method = "lm",
se = FALSE,
aes(color = "fluoro-coated backsheet/PET/EVA")
```

```r
) + stat_smooth(
data = xenon2sa31001,
method = "lm",
se = FALSE,
aes(color = "PVDF/PET/EVA")
) + stat_smooth(
data = xenon2sa31003,
method = "lm",
se = FALSE,
aes(color = "PET/PET/EVA")
) + stat_smooth(
data = xenon2sa31004,
method = "lm",
se = FALSE,
aes(color = "PVF/PET/EVA")
) + stat_smooth(
data = xenon2sa31005,
method = "lm",
se = FALSE,
aes(color = "polyamide (PA)/PA/PA")
) + scale_colour_manual(name = "Legend",
values = c("green", "orange", "purple", "blue", "red"))
```



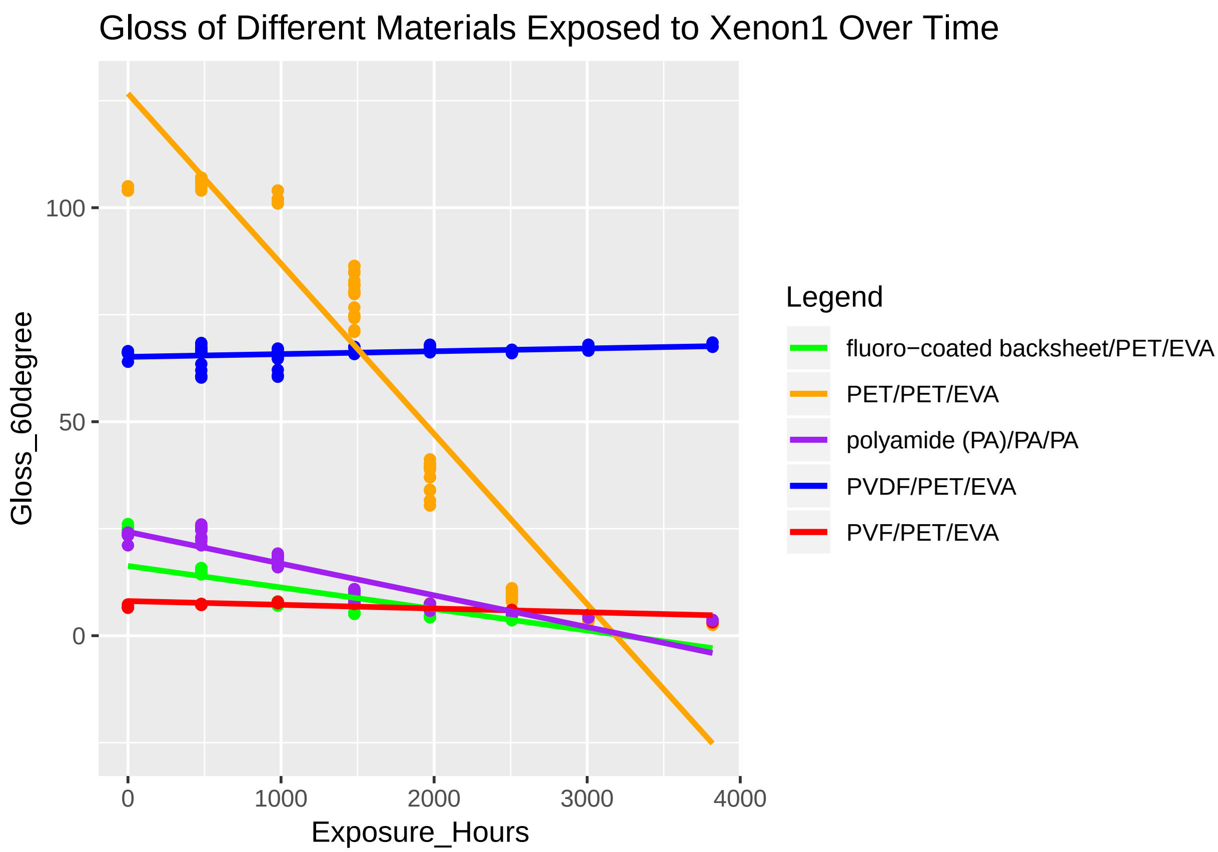

```r
# For gloss,

# Damp heat exposure
```

```
ggplot(data = dampheatsa31001, aes(x = Exposure_Hours, y = Gloss_60degree)) + geom_point(col = "blue")
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "green") + geom_point(data = dampheatsa31003,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "orange") + geom_point(data = dampheatsa31004,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "red") + geom_point(data = dampheatsa31005,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "purple") + ggtitle("Gloss of Different Materials Exposed to Damp Heat Over Time") + stat_smooth(
data = dampheatsa31002,
method = "lm",
se = FALSE,
aes(color = "fluoro-coated backsheet/PET/EVA")
) + stat_smooth(
data = dampheatsa31001,
method = "lm",
se = FALSE,
aes(color = "PVDF/PET/EVA")
) + stat_smooth(
data = dampheatsa31003,
method = "lm",
se = FALSE,
aes(color = "PET/PET/EVA")
) + stat_smooth(
data = dampheatsa31004,
method = "lm",
se = FALSE,
aes(color = "PVF/PET/EVA")
) + stat_smooth(
data = dampheatsa31005,
method = "lm",
se = FALSE,
aes(color = "polyamide (PA)/PA/PA")
) + scale_colour_manual(name = "Legend",
values = c("green", "orange", "purple", "blue", "red"))
```



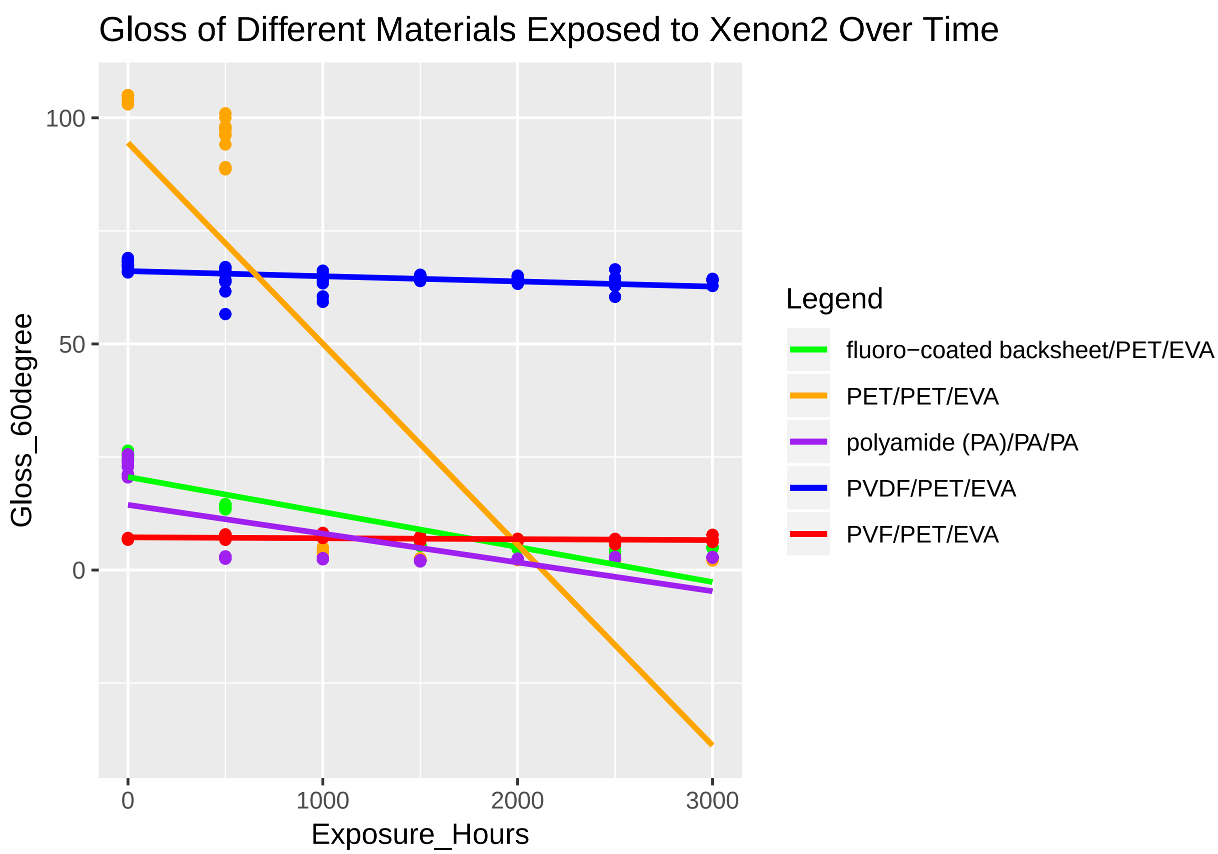

```
# Xenon1 exposure
ggplot(data = xenon1sa31001, aes(x = Exposure_Hours, y = Gloss_60degree)) + geom_point(col = "blue")  +
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "green") + geom_point(data = xenon1sa31003,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "orange") + geom_point(data = xenon1sa31004,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "red") + geom_point(data = xenon1sa31005,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "purple") + ggtitle("Gloss of Different Materials Exposed to Xenon1 Over Time") + stat_smooth(
data = xenon1sa31002,
method = "lm",
se = FALSE,
aes(color = "fluoro-coated backsheet/PET/EVA")
) + stat_smooth(
data = xenon1sa31001,
method = "lm",
se = FALSE,
aes(color = "PVDF/PET/EVA")
) + stat_smooth(
data = xenon1sa31003,
method = "lm",
se = FALSE,
aes(color = "PET/PET/EVA")
) + stat_smooth(
data = xenon1sa31004,
```

```
method = "lm",
se = FALSE,
aes(color = "PVF/PET/EVA")
) + stat_smooth(
data = xenon1sa31005,
method = "lm",
se = FALSE,
aes(color = "polyamide (PA)/PA/PA")
) + scale_colour_manual(name = "Legend",
values = c("green", "orange", "purple", "blue", "red"))
```




```
# Xenon2 exposure

ggplot(data = xenon2sa31001, aes(x = Exposure_Hours, y = Gloss_60degree)) + geom_point(col = "blue")  +
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "green") + geom_point(data = xenon2sa31003,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "orange") + geom_point(data = xenon2sa31004,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "red") + geom_point(data = xenon2sa31005,
aes(x = Exposure_Hours, y = Gloss_60degree),
col = "purple") + ggtitle("Gloss of Different Materials Exposed to Xenon2 Over Time") + stat_smooth(
data = xenon2sa31002,
method = "lm",
se = FALSE,
aes(color = "fluoro-coated backsheet/PET/EVA")
```

```
) + stat_smooth(
data = xenon2sa31001,
method = "lm",
se = FALSE,
aes(color = "PVDF/PET/EVA")
) + stat_smooth(
data = xenon2sa31003,
method = "lm",
se = FALSE,
aes(color = "PET/PET/EVA")
) + stat_smooth(
data = xenon2sa31004,
method = "lm",
se = FALSE,
aes(color = "PVF/PET/EVA")
) + stat_smooth(
data = xenon2sa31005,
method = "lm",
se = FALSE,
aes(color = "polyamide (PA)/PA/PA")
) + scale_colour_manual(name = "Legend",
values = c("green", "orange", "purple", "blue", "red"))
```



#### 2.3.3 Discuss the differences you notice between the data and the models.

Most of the data is linear which is why linear models are used. However, there are some nonlinear models which will be further discussed in part (5).

#### 2.3.4 For each exposure describe which material performs the best and worst, explain your conclusions.

We know that increasing the yellowness index and decreasing the gloss indicates degradation. This means the best materials will have the best materials will have the smallest absolute value of the slope to indicate the smallest rate of increase or decrease in yellowness index or gloss respectively while the worst materials will have the highest absolute value of slope to indicate the highest rate of increase or decrease in yellowness index or gloss respectively. I will rank each material from 1-5 with 1 being the best and 5 being the worst performer by comparing the coefficients of the slopes of the linear models. The cumulative rank from 2-10 will indicate which has been the overall best performer and which the worst.

```r
# This function will return the name of the material with an input of the sampleID.
returnMaterial <- function(sampleid){
  if (sampleid == "sa31001") {
    return("PVDF/PET/EVA")
  } else if (sampleid == "sa31002") {
    return("fluoro-coated backsheet/PET/EVA")
  } else if (sampleid == "sa31003") {
    return("PET/PET/EVA")
  } else if (sampleid == "sa31004") {
      return("PVF/PET/EVA")
  } else if (sampleid == "sa31005") {
    return("polyamide (PA)/PA/PA")
  }
}
```

```r
# Creating an array with scores of the yellowness index performance under damp heat conditions
dampheatyiscore <- rep(NULL, length(samples))
# Arranging the materials based on increasing slope
for (i in 1:length(sort(dampheatyislopes))) {
    message <- paste(returnMaterial(samples[which(dampheatyislopes == sort(dampheatyislopes)[i])]),
                samples[which(dampheatyislopes == sort(dampheatyislopes)[i])])
    if (i == 1) {
      message <- paste("BEST ", message)
    } else if (i == length(sort(dampheatyislopes))){
      message <- paste("WORST", message)
    } else {
      message <- paste("     ", message)
    }
    # Giving a score to each material
    dampheatyiscore[which(dampheatyislopes == sort(dampheatyislopes)[i])] <-
      i
    print(message)
}
```

```
## [1] "BEST  PVF/PET/EVA sa31004"
## [1] "      polyamide (PA)/PA/PA sa31005"
## [1] "      PET/PET/EVA sa31003"
## [1] "      fluoro-coated backsheet/PET/EVA sa31002"
## [1] "WORST PVDF/PET/EVA sa31001"
```

```r
# Creating an array with scores of the gloss performance under damp heat conditions
dampheatglossscore <- rep(NULL, length(samples))
# Arranging the materials based on decreasing slope
for (i in 1:length(dampheatglossslopes)) {
    message <- paste(returnMaterial(samples[which(dampheatglossslopes == sort(dampheatglossslopes, decr
                  samples[which(dampheatglossslopes == sort(dampheatglossslopes, decreasing = TRUE)[i])])
    if (i == 1) {
      message <- paste("BEST ", message)
    } else if (i == length(sort(dampheatyislopes))){
      message <- paste("WORST", message)
    } else {
      message <- paste("     ", message)
    }
    # Giving a score to each material
    dampheatglossscore[which(dampheatglossslopes == sort(dampheatglossslopes, decreasing = TRUE)[i])] <-
      i
    print(message)
}
```

```
## [1] "BEST  PVDF/PET/EVA sa31001"
## [1] "      fluoro-coated backsheet/PET/EVA sa31002"
## [1] "      PVF/PET/EVA sa31004"
## [1] "      PET/PET/EVA sa31003"
## [1] "WORST polyamide (PA)/PA/PA sa31005"
```

```r
# Overall scores for Damp Heat conditions
dampheatscore <- dampheatglossscore + dampheatyiscore

# Printing out the score for each material
for (i in 1:length(dampheatscore)){
  print(paste("Score for",
              returnMaterial(samples[i]),
              samples[i], ":", dampheatscore[i]))
}
```

```
## [1] "Score for PVDF/PET/EVA sa31001 : 6"
## [1] "Score for fluoro-coated backsheet/PET/EVA sa31002 : 6"
## [1] "Score for PET/PET/EVA sa31003 : 7"
## [1] "Score for PVF/PET/EVA sa31004 : 4"
## [1] "Score for polyamide (PA)/PA/PA sa31005 : 7"
```

```r
# Thus, it can be concluded PVF/PET/EVA(sa31004) is the best performer under damp heat and PET/PET/EVA(
```

```r
# Creating an array with scores of the yellowness index performance under xenon1 exposure
xenon1yiscore <- rep(NULL, length(samples))
# Arranging the materials based on increasing slope
for (i in 1:length(sort(xenon1yislopes))) {
    message <- paste(returnMaterial(samples[which(xenon1yislopes == sort(xenon1yislopes)[i])]),
                  samples[which(xenon1yislopes == sort(xenon1yislopes)[i])])
    if (i == 1) {
      message <- paste("BEST ", message)
    } else if (i == length(sort(xenon1yislopes))){
      message <- paste("WORST", message)
    } else {
      message <- paste("     ", message)
```

```
    }
    # Giving a score to each material
    xenon1yiscore[which(xenon1yislopes == sort(xenon1yislopes)[i])] <-
      i
    print(message)
}
```

```
## [1] "BEST  PVF/PET/EVA sa31004"
## [1] "      fluoro-coated backsheet/PET/EVA sa31002"
## [1] "      PVDF/PET/EVA sa31001"
## [1] "      polyamide (PA)/PA/PA sa31005"
## [1] "WORST PET/PET/EVA sa31003"
```

```
# Creating an array with scores of the gloss performance under xenon1 exposure
xenon1glossscore <- rep(NULL, length(samples))
# Arranging the materials based on decreasing slope
for (i in 1:length(xenon1glossslopes)) {
    message <- paste(returnMaterial(samples[which(xenon1glossslopes == sort(xenon1glossslopes, decreasi
                samples[which(xenon1glossslopes == sort(xenon1glossslopes, decreasing = TRUE)[i])])
    if (i == 1) {
      message <- paste("BEST ", message)
    } else if (i == length(sort(xenon1yislopes))){
      message <- paste("WORST", message)
    } else {
      message <- paste("     ", message)
    }
    # Giving a score to each material
    xenon1glossscore[which(xenon1glossslopes == sort(xenon1glossslopes, decreasing = TRUE)[i])] <-
      i
    print(message)
}
```

```
## [1] "BEST  PVDF/PET/EVA sa31001"
## [1] "      PVF/PET/EVA sa31004"
## [1] "      fluoro-coated backsheet/PET/EVA sa31002"
## [1] "      polyamide (PA)/PA/PA sa31005"
## [1] "WORST PET/PET/EVA sa31003"
```

```
# Overall scores for xenon1 exposure
xenon1score <- xenon1glossscore + xenon1yiscore

# Printing out the score for each material
for (i in 1:length(xenon1score)){
  print(paste("Score for",
              returnMaterial(samples[i]),
              samples[i], ":", xenon1score[i]))
}
```

```
## [1] "Score for PVDF/PET/EVA sa31001 : 4"
## [1] "Score for fluoro-coated backsheet/PET/EVA sa31002 : 5"
## [1] "Score for PET/PET/EVA sa31003 : 10"
## [1] "Score for PVF/PET/EVA sa31004 : 3"
## [1] "Score for polyamide (PA)/PA/PA sa31005 : 8"
```

```
# Thus, it can be concluded PVF/PET/EVA(sa31004) is the best performer under xenon1 and PET/PET/EVA(sa3
```

```r
# Creating an array with scores of the yellowness index performance under xenon1 exposure
xenon2yiscore <- rep(NULL, length(samples))
# Arranging the materials based on increasing slope
for (i in 1:length(sort(xenon2yislopes))) {
    message <- paste(returnMaterial(samples[which(xenon2yislopes == sort(xenon2yislopes)[i])]),
                samples[which(xenon2yislopes == sort(xenon2yislopes)[i])])
    if (i == 1) {
      message <- paste("BEST ", message)
    } else if (i == length(sort(xenon2yislopes))){
      message <- paste("WORST", message)
    } else {
      message <- paste("     ", message)
    }
    # Giving a score to each material
    xenon2yiscore[which(xenon2yislopes == sort(xenon2yislopes)[i])] <-
      i
    print(message)
}
```

```
## [1] "BEST  PVF/PET/EVA sa31004"
## [1] "      fluoro-coated backsheet/PET/EVA sa31002"
## [1] "      polyamide (PA)/PA/PA sa31005"
## [1] "      PVDF/PET/EVA sa31001"
## [1] "WORST PET/PET/EVA sa31003"
```

```r
# Creating an array with scores of the gloss performance under xenon2 exposure
xenon2glossscore <- rep(NULL, length(samples))
# Arranging the materials based on decreasing slope
for (i in 1:length(xenon2glossslopes)) {
    message <- paste(returnMaterial(samples[which(xenon2glossslopes == sort(xenon2glossslopes, decreasi
                samples[which(xenon2glossslopes == sort(xenon2glossslopes, decreasing = TRUE)[i])])
    if (i == 1) {
      message <- paste("BEST ", message)
    } else if (i == length(sort(xenon2yislopes))){
      message <- paste("WORST", message)
    } else {
      message <- paste("     ", message)
    }
    # Giving a score to each material
    xenon2glossscore[which(xenon2glossslopes == sort(xenon2glossslopes, decreasing = TRUE)[i])] <-
      i
    print(message)
}
```

```
## [1] "BEST  PVF/PET/EVA sa31004"
## [1] "      PVDF/PET/EVA sa31001"
## [1] "      polyamide (PA)/PA/PA sa31005"
## [1] "      fluoro-coated backsheet/PET/EVA sa31002"
## [1] "WORST PET/PET/EVA sa31003"
```

```r
# Overall scores for xenon2 exposure
xenon2score <- xenon2glossscore + xenon2yiscore

# Printing out the score for each material
for (i in 1:length(xenon2score)){
```

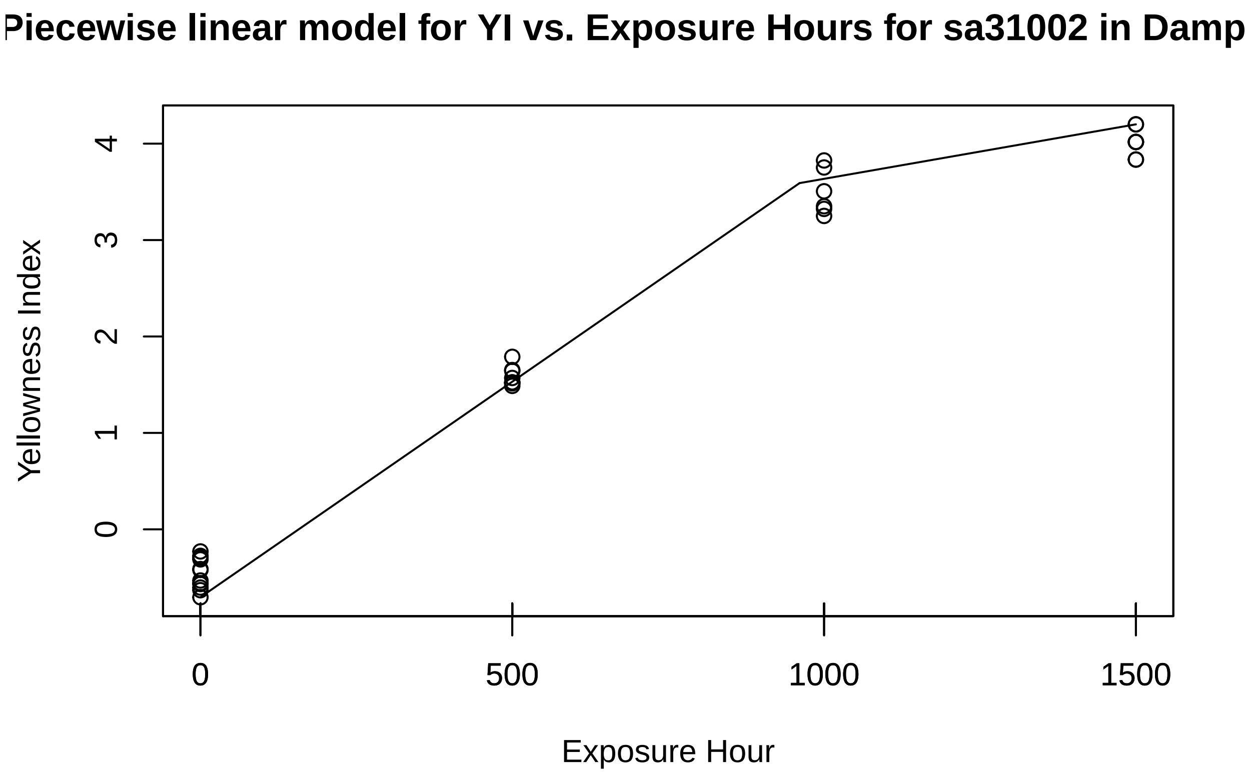
```r
  print(paste("Score for",
              returnMaterial(samples[i]),
              samples[i], ":", xenon2score[i]))
}
```

```
## [1] "Score for PVDF/PET/EVA sa31001 : 6"
## [1] "Score for fluoro-coated backsheet/PET/EVA sa31002 : 6"
## [1] "Score for PET/PET/EVA sa31003 : 10"
## [1] "Score for PVF/PET/EVA sa31004 : 2"
## [1] "Score for polyamide (PA)/PA/PA sa31005 : 6"
```

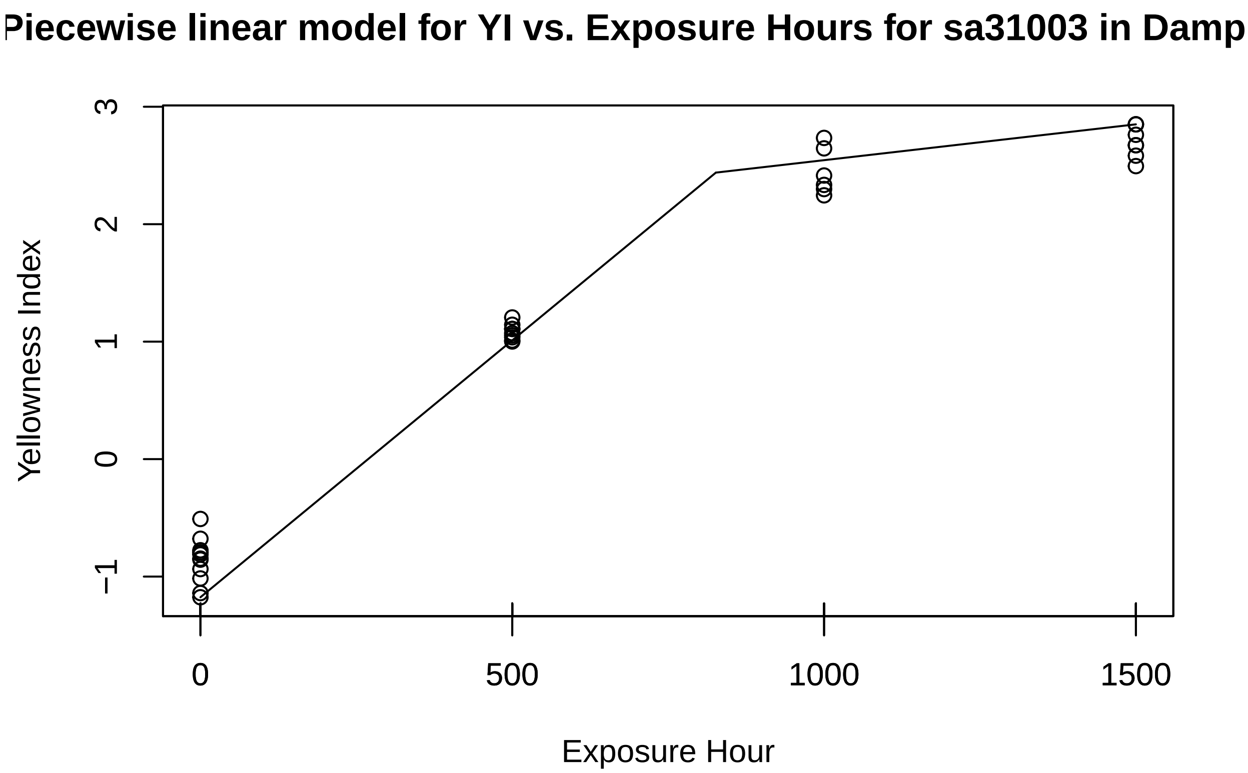
```r
# Once again, PVF/PET/EVA(sa31004) is the best under xenon 2 exposure and PET/PET/EVA(sa31003) the wors
```

### 2.3.5 Do any of the models appear non-linear? If so, try and improve their fit with a power transformation or a piecewise linear model.

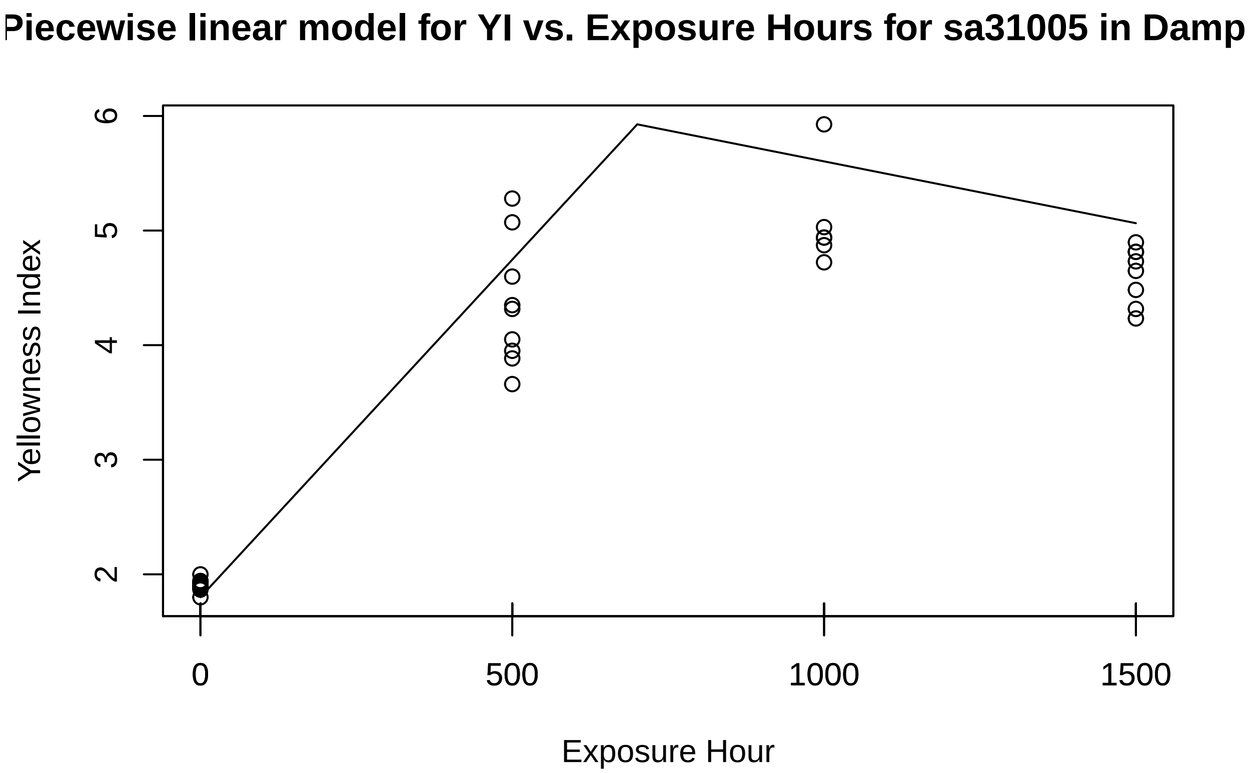
```r
# We cannot use box cox transformations as there are negative values so we use piecewise linear models.
# sa31002 for damp heat exposure
plot(dampheatsa31002$Exposure_Hours,
     dampheatsa31002$Yellowness_Index,
     main = "Piecewise linear model for YI vs. Exposure Hours for sa31002 in Damp Heat",
     xlab = "Exposure Hour",
     ylab = "Yellowness Index", yaxt = "n")
par(new = TRUE)
# Creating a piecewise linear model
dampheatsa31002exposurehours <- dampheatsa31002$Exposure_Hours
dampheatsa31002yi <- dampheatsa31002$Yellowness_Index

lmpiecewisedampheatsa31002 <-
  segmented(lm(dampheatsa31002yi ~
                 dampheatsa31002exposurehours),
            seg.Z = ~ dampheatsa31002exposurehours,
            psi = 999)

plot(lmpiecewisedampheatsa31002, xlab = "",
     ylab = "")
```



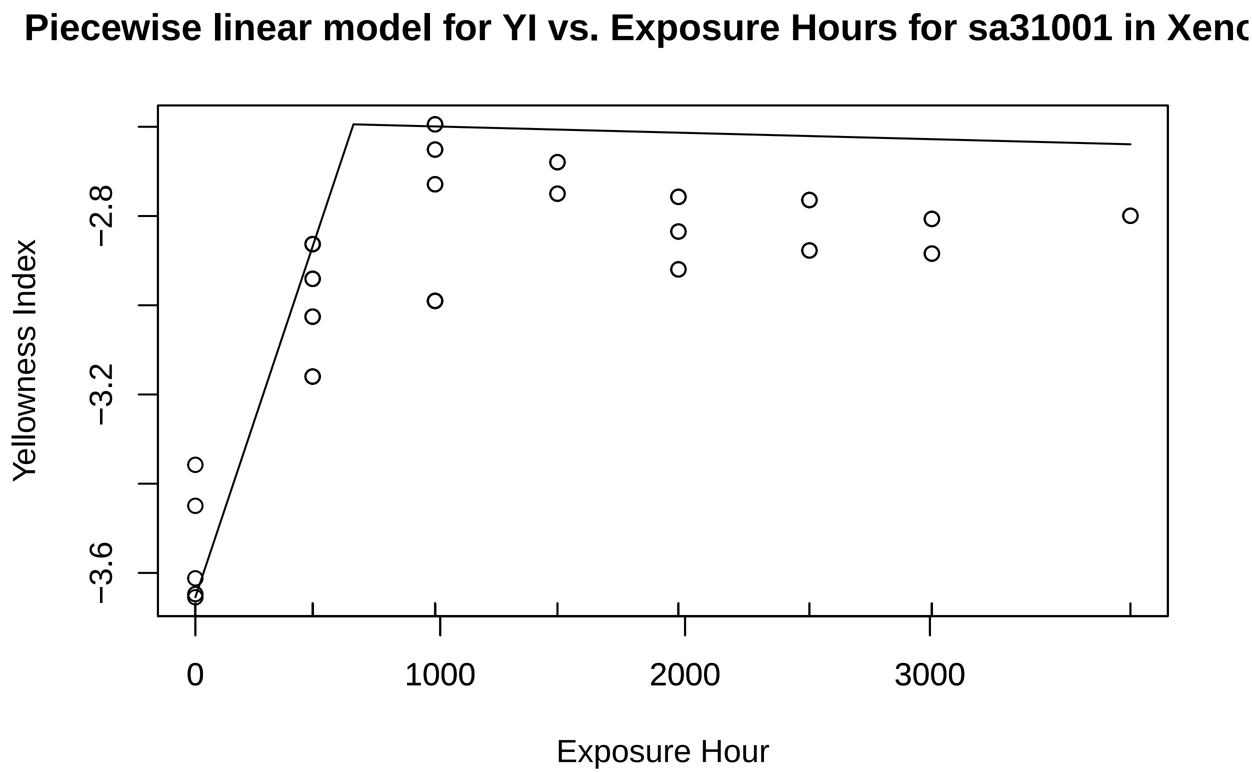

```
# Making a dataframe to be used by ggplot
gglmpiecewisedampheatsa31002 <-
  data.frame(Exposure_Hours = dampheatsa31002$Exposure_Hours,
                      Yellowness_Index = broken.line(lmpiecewisedampheatsa31002)$fit)

# sa31003 for damp heat exposure
plot(dampheatsa31003$Exposure_Hours,
     dampheatsa31003$Yellowness_Index,
     main = "Piecewise linear model for YI vs. Exposure Hours for sa31003 in Damp Heat",
     xlab = "Exposure Hour",
     ylab = "Yellowness Index", yaxt = "n")
par(new = TRUE)
# Creating a piecewise linear model
dampheatsa31003exposurehours <- dampheatsa31003$Exposure_Hours
dampheatsa31003yi <- dampheatsa31003$Yellowness_Index

lmpiecewisedampheatsa31003 <-
  segmented(lm(dampheatsa31003yi ~
                 dampheatsa31003exposurehours),
            seg.Z = ~ dampheatsa31003exposurehours,
            psi = 999)

plot(lmpiecewisedampheatsa31003, xlab = "",
     ylab = "")
```



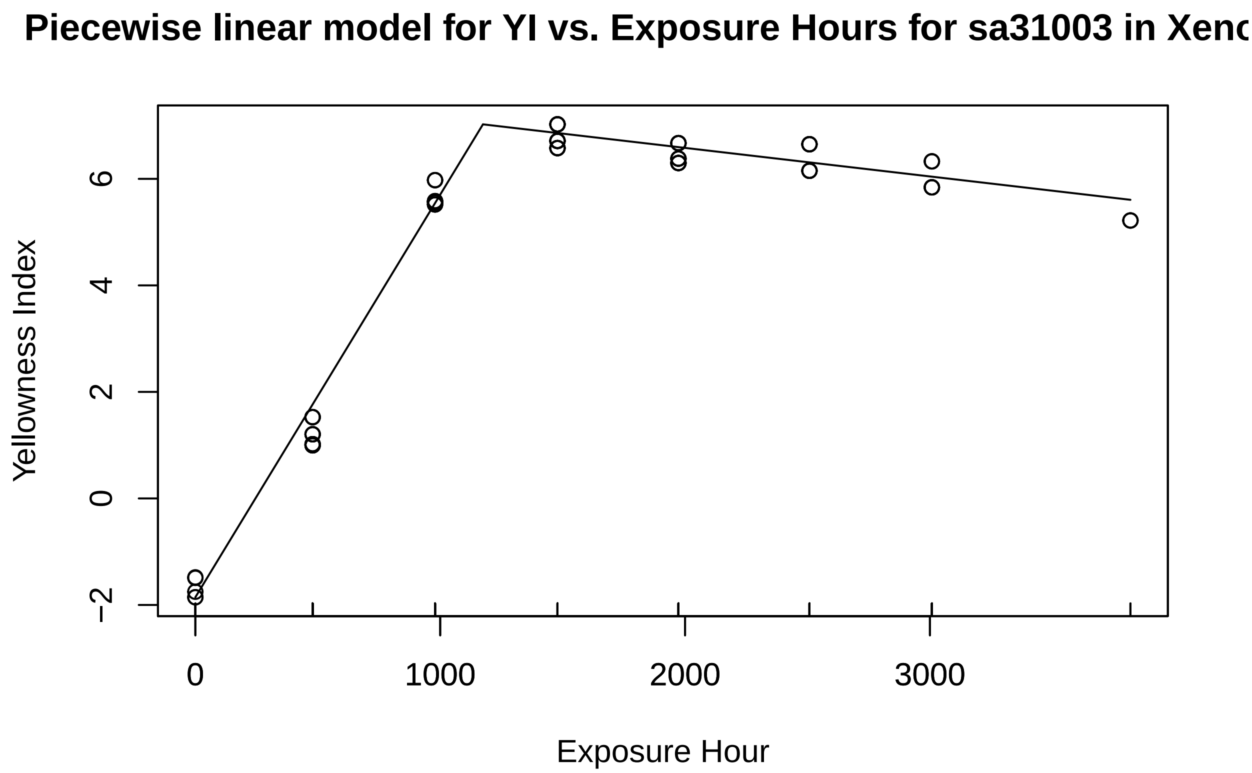

```
# Making a dataframe to be used by ggplot
gglmpiecewisedampheatsa31003 <-
  data.frame(Exposure_Hours = dampheatsa31003$Exposure_Hours,
                     Yellowness_Index = broken.line(lmpiecewisedampheatsa31003)$fit)

# sa31005 for damp heat exposure
plot(dampheatsa31005$Exposure_Hours,
     dampheatsa31005$Yellowness_Index,
     main = "Piecewise linear model for YI vs. Exposure Hours for sa31005 in Damp Heat",
     xlab = "Exposure Hour",
     ylab = "Yellowness Index", yaxt = "n")
par(new = TRUE)
# Creating a piecewise linear model
dampheatsa31005exposurehours <- dampheatsa31005$Exposure_Hours
dampheatsa31005yi <- dampheatsa31005$Yellowness_Index

lmpiecewisedampheatsa31005 <-
  segmented(lm(dampheatsa31005yi ~
                 dampheatsa31005exposurehours),
            seg.Z = ~ dampheatsa31005exposurehours,
            psi = 1000)

plot(lmpiecewisedampheatsa31005, xlab = "",
     ylab = "")
```



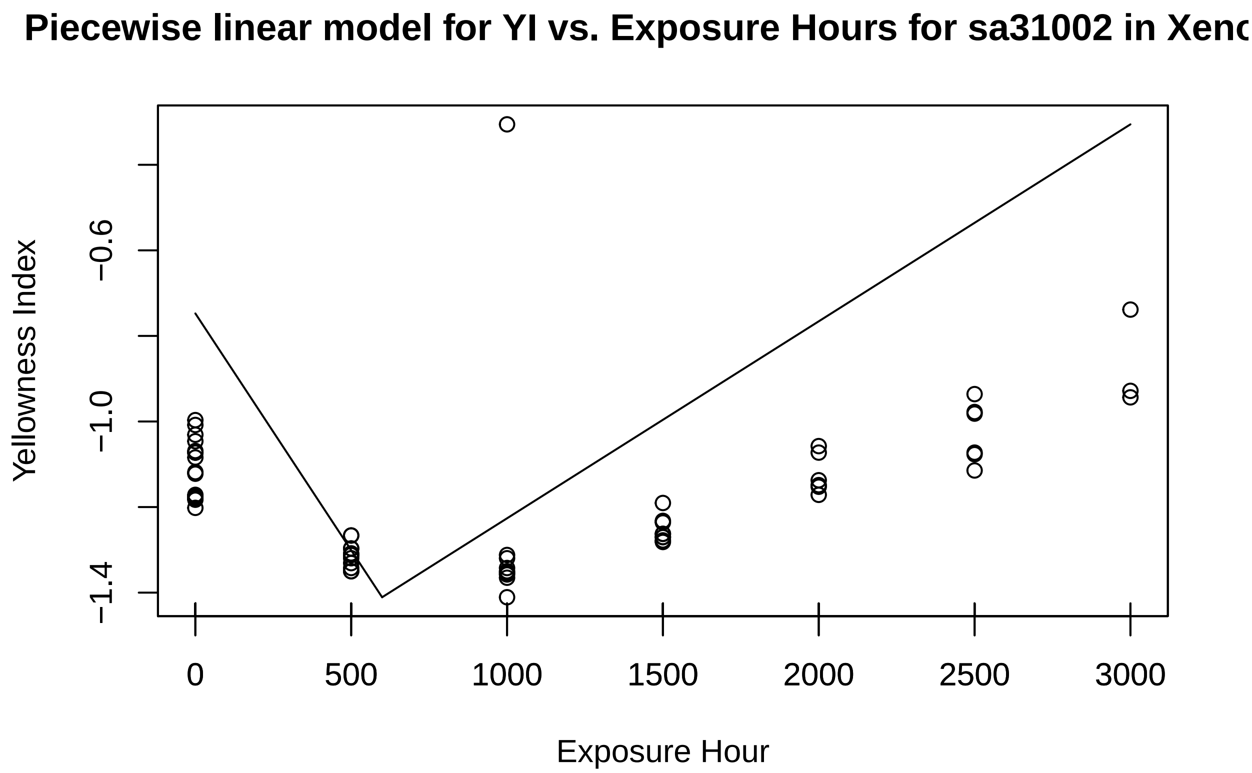

```
# Making a dataframe to be used by ggplot
gglmpiecewisedampheatsa31005 <-
  data.frame(Exposure_Hours = dampheatsa31005$Exposure_Hours,
                       Yellowness_Index = broken.line(lmpiecewisedampheatsa31005)$fit)

# sa31001 for xenon1 exposure
plot(xenon1sa31001$Exposure_Hours,
     xenon1sa31001$Yellowness_Index,
     main = "Piecewise linear model for YI vs. Exposure Hours for sa31001 in Xenon1",
     xlab = "Exposure Hour",
     ylab = "Yellowness Index", yaxt = "n")

par(new = TRUE)
# Creating a piecewise linear model
xenon1sa31001exposurehours <- xenon1sa31001$Exposure_Hours
xenon1sa31001yi <- xenon1sa31001$Yellowness_Index

lmpiecewisexenon1sa31001 <-
  segmented(lm(xenon1sa31001yi ~
                 xenon1sa31001exposurehours),
             seg.Z = ~ xenon1sa31001exposurehours,
             psi = 1000)

plot(lmpiecewisexenon1sa31001, xlab = "",
     ylab = "")
```



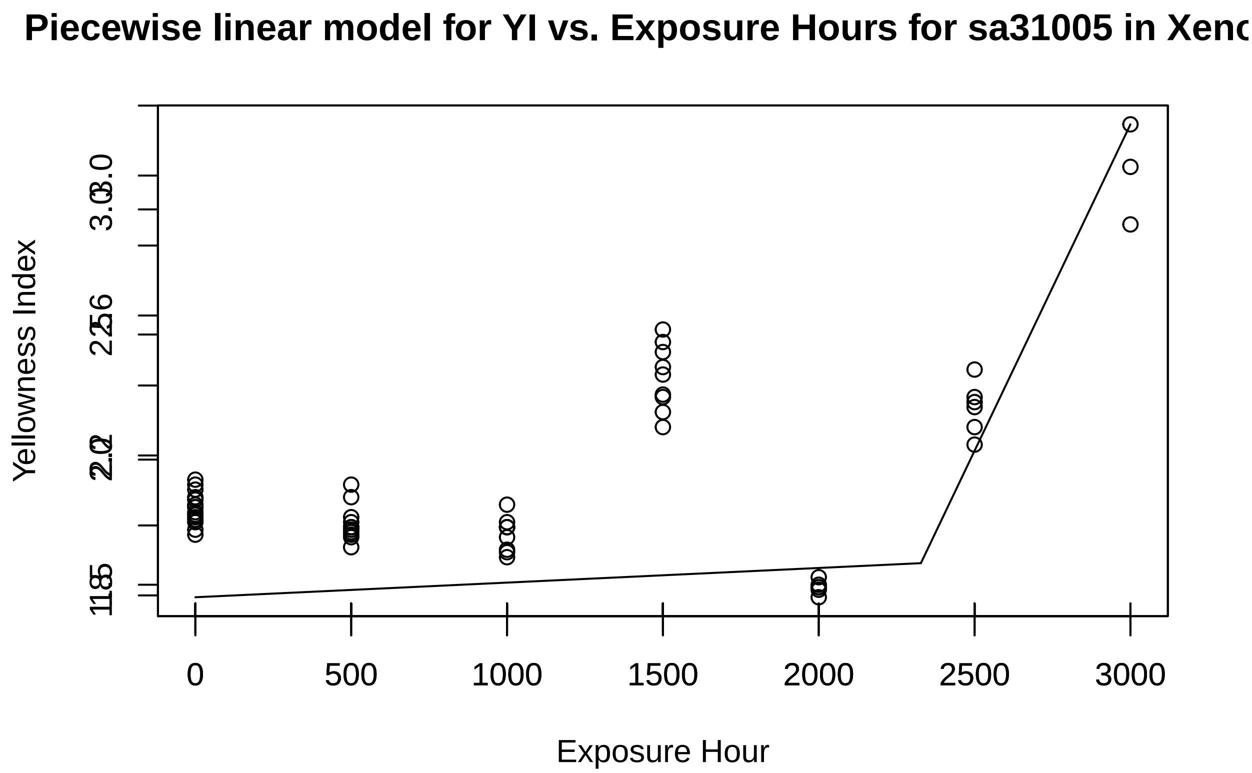

```r
# Making a dataframe to be used by ggplot
gglmpiecewisexenon1sa31001 <-
  data.frame(Exposure_Hours = xenon1sa31001$Exposure_Hours,
                   Yellowness_Index = broken.line(lmpiecewisexenon1sa31001)$fit)

# sa31003 for xenon1 exposure
plot(xenon1sa31003$Exposure_Hours,
     xenon1sa31003$Yellowness_Index,
     main = "Piecewise linear model for YI vs. Exposure Hours for sa31003 in Xenon1",
     xlab = "Exposure Hour",
     ylab = "Yellowness Index", yaxt = "n")

par(new = TRUE)
# Creating a piecewise linear model
xenon1sa31003exposurehours <- xenon1sa31003$Exposure_Hours
xenon1sa31003yi <- xenon1sa31003$Yellowness_Index

lmpiecewisexenon1sa31003 <-
  segmented(lm(xenon1sa31003yi ~
                 xenon1sa31003exposurehours),
             seg.Z = ~ xenon1sa31003exposurehours,
             psi = 1000)

plot(lmpiecewisexenon1sa31003, xlab = "",
     ylab = "")
```




```r
# Making a dataframe to be used by ggplot
gglmpiecewisexenon1sa31003 <-
  data.frame(Exposure_Hours = xenon1sa31003$Exposure_Hours,
                     Yellowness_Index = broken.line(lmpiecewisexenon1sa31003)$fit)

# sa31002 for xenon2 exposure
plot(xenon2sa31002$Exposure_Hours,
     xenon2sa31002$Yellowness_Index,
     main = "Piecewise linear model for YI vs. Exposure Hours for sa31002 in Xenon2",
     xlab = "Exposure Hour",
     ylab = "Yellowness Index", yaxt = "n")

par(new = TRUE)
# Creating a piecewise linear model
xenon2sa31002exposurehours <- xenon2sa31002$Exposure_Hours
xenon2sa31002yi <- xenon2sa31002$Yellowness_Index

lmpiecewisexenon2sa31002 <-
  segmented(lm(xenon2sa31002yi ~
                 xenon2sa31002exposurehours),
            seg.Z = ~ xenon2sa31002exposurehours,
            psi = 1000)

plot(lmpiecewisexenon2sa31002, xlab = "",
     ylab = "")
```



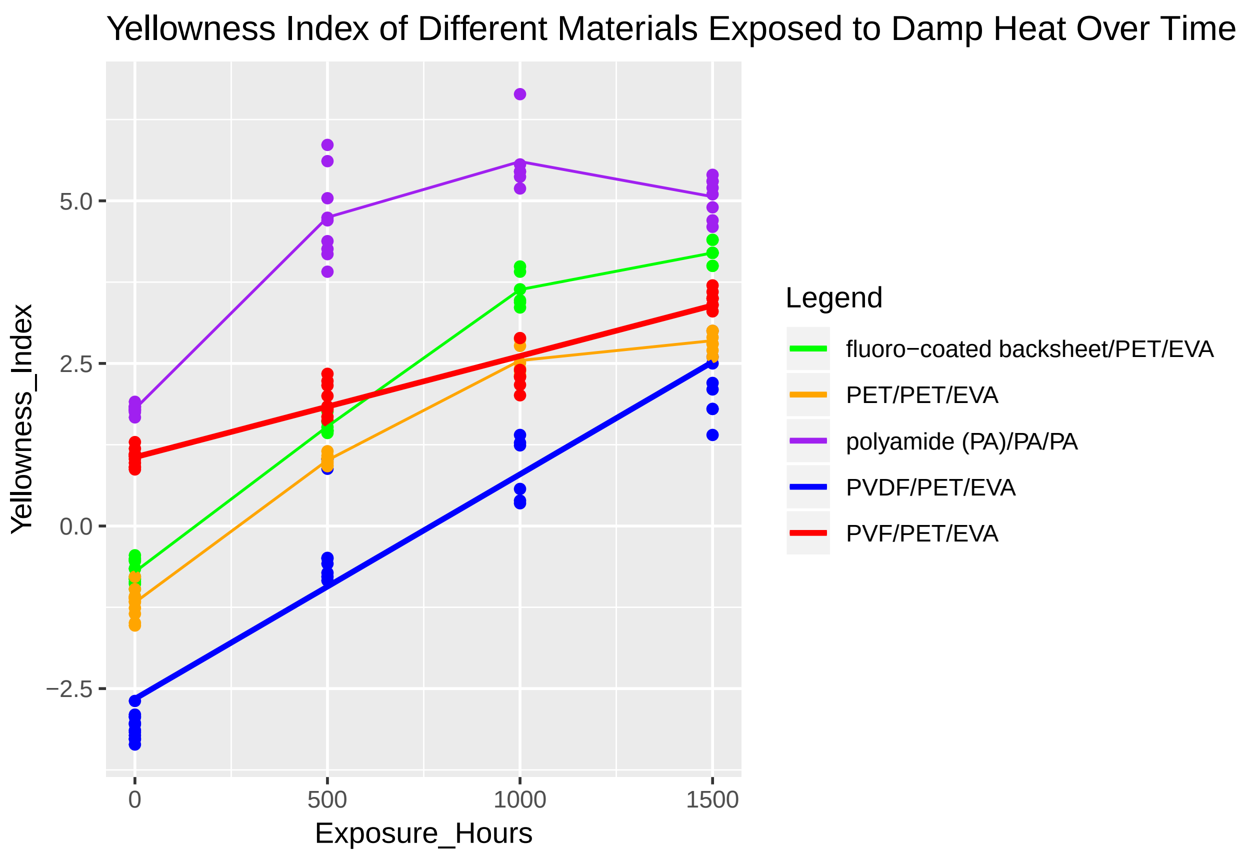

```r
# Making a dataframe to be used by ggplot
gglmpiecewisexenon2sa31002 <-
  data.frame(Exposure_Hours = xenon2sa31002$Exposure_Hours,
                    Yellowness_Index = broken.line(lmpiecewisexenon2sa31002)$fit)


#  sa31005 for xenon2 exposure
plot(xenon2sa31005$Exposure_Hours,
     xenon2sa31005$Yellowness_Index,
     main = "Piecewise linear model for YI vs. Exposure Hours for sa31005 in Xenon2",
     xlab = "Exposure Hour",
     ylab = "Yellowness Index")

par(new = TRUE)
# Creating a piecewise linear model
xenon2sa31005exposurehours <- xenon2sa31005$Exposure_Hours
xenon2sa31005yi <- xenon2sa31005$Yellowness_Index

lmpiecewisexenon2sa31005 <-
  segmented(lm(xenon2sa31005yi ~
                 xenon2sa31005exposurehours),
             seg.Z = ~ xenon2sa31005exposurehours,
             psi = 1000)

plot(lmpiecewisexenon2sa31005, xlab = "",
     ylab = "")
```




```r
# Making a dataframe to be used by ggplot
gglmpiecewisexenon2sa31005 <-
  data.frame(Exposure_Hours = xenon2sa31005$Exposure_Hours,
                       Yellowness_Index = broken.line(lmpiecewisexenon2sa31005)$fit)
```

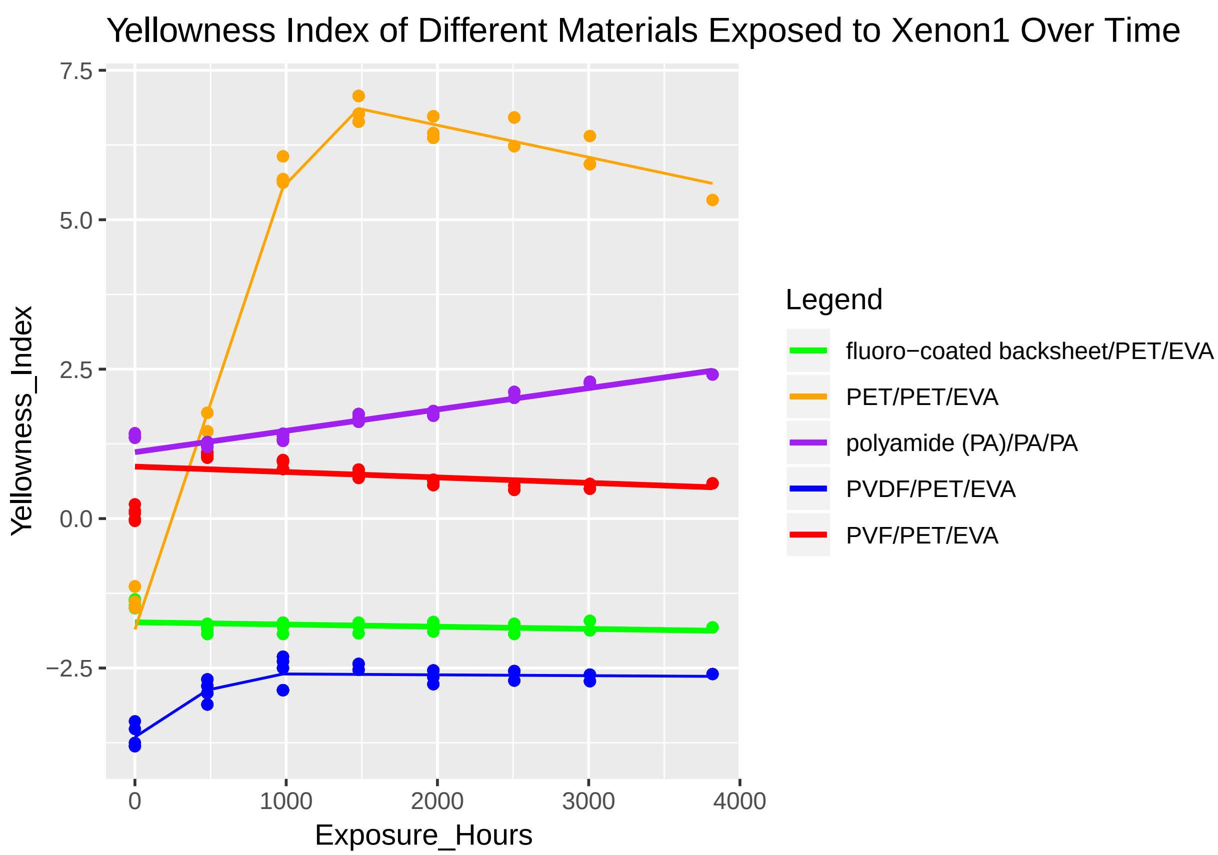
```r
# For yellowness index,

# Damp Heat
ggplot(data = dampheatsa31001,
       aes(x = Exposure_Hours,
           y = Yellowness_Index)) +
  geom_point(col = "blue")  +
  geom_point(data = dampheatsa31002,
             aes(x = Exposure_Hours,
                 y = Yellowness_Index),
             col = "green") +
  geom_point(data = dampheatsa31003,
             aes(x = Exposure_Hours,
                 y = Yellowness_Index),
             col = "orange") +
  geom_point(data = dampheatsa31004,
             aes(x = Exposure_Hours,
                 y = Yellowness_Index),
             col = "red") +
  geom_point(data = dampheatsa31005,
             aes(x = Exposure_Hours,
```

```r
                 y = Yellowness_Index),
             col = "purple") +
  ggtitle("Yellowness Index of Different Materials Exposed to Damp Heat Over Time") +
  geom_line(data = gglmpiecewisedampheatsa31002,
            aes(x = Exposure_Hours,
                y = Yellowness_Index,
                color = "fluoro-coated backsheet/PET/EVA")) +
  stat_smooth(data = dampheatsa31001,
              method = "lm",
              se = FALSE,
              aes(color = "PVDF/PET/EVA")) +
  geom_line(data = gglmpiecewisedampheatsa31003,
            aes(x = Exposure_Hours,
                y = Yellowness_Index,
                color = "PET/PET/EVA")) +
  stat_smooth(data = dampheatsa31004,
              method = "lm",
              se = FALSE,
              aes(color = "PVF/PET/EVA")) +
  geom_line(data = gglmpiecewisedampheatsa31005,
            aes(x = Exposure_Hours,
                y = Yellowness_Index,
                color = "polyamide (PA)/PA/PA")) +
  scale_colour_manual(name="Legend",
                      values=c("green",
                               "orange",
                               "purple",
                               "blue",
                               "red"))
```



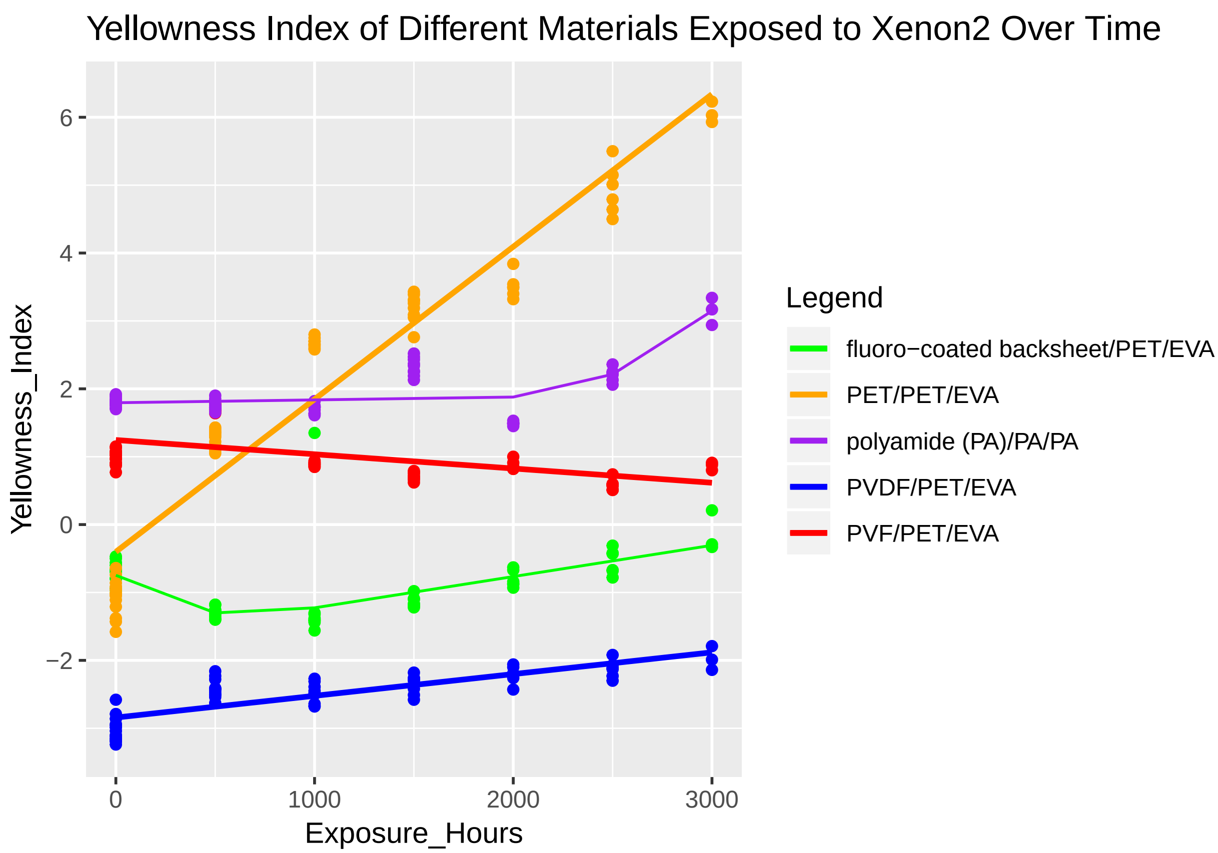

```
# Xenon1
ggplot(data = xenon1sa31001,
       aes(x = Exposure_Hours,
           y = Yellowness_Index)) +
  geom_point(col = "blue")  +
  geom_point(data = xenon1sa31002,
             aes(x = Exposure_Hours,
                 y = Yellowness_Index),
             col = "green") +
  geom_point(data = xenon1sa31003,
             aes(x = Exposure_Hours,
                 y = Yellowness_Index),
             col = "orange") +
  geom_point(data = xenon1sa31004,
             aes(x = Exposure_Hours,
                 y = Yellowness_Index),
             col = "red") +
  geom_point(data = xenon1sa31005,
             aes(x = Exposure_Hours,
                 y = Yellowness_Index),
             col = "purple") +
  ggtitle("Yellowness Index of Different Materials Exposed to Xenon1 Over Time") +
  stat_smooth(data = xenon1sa31002,
              method = "lm",
              se = FALSE,
              aes(color = "fluoro-coated backsheet/PET/EVA"))+
```

```
  geom_line(data = gglmpiecewisexenon1sa31001,
            aes(x = Exposure_Hours,
                y = Yellowness_Index,
                color = "PVDF/PET/EVA")) +
  geom_line(data = gglmpiecewisexenon1sa31003,
            aes(x = Exposure_Hours,
                y = Yellowness_Index,
                color = "PET/PET/EVA")) +
  stat_smooth(data = xenon1sa31004,
              method = "lm",
              se = FALSE,
              aes(color = "PVF/PET/EVA")) +
  stat_smooth(data = xenon1sa31005,
              method = "lm",
              se = FALSE,
              aes(color = "polyamide (PA)/PA/PA")) +
  scale_colour_manual(name="Legend",
                      values=c("green",
                               "orange",
                               "purple",
                               "blue",
                               "red"))
```




```
# Xenon2

ggplot(data = xenon2sa31001,
```

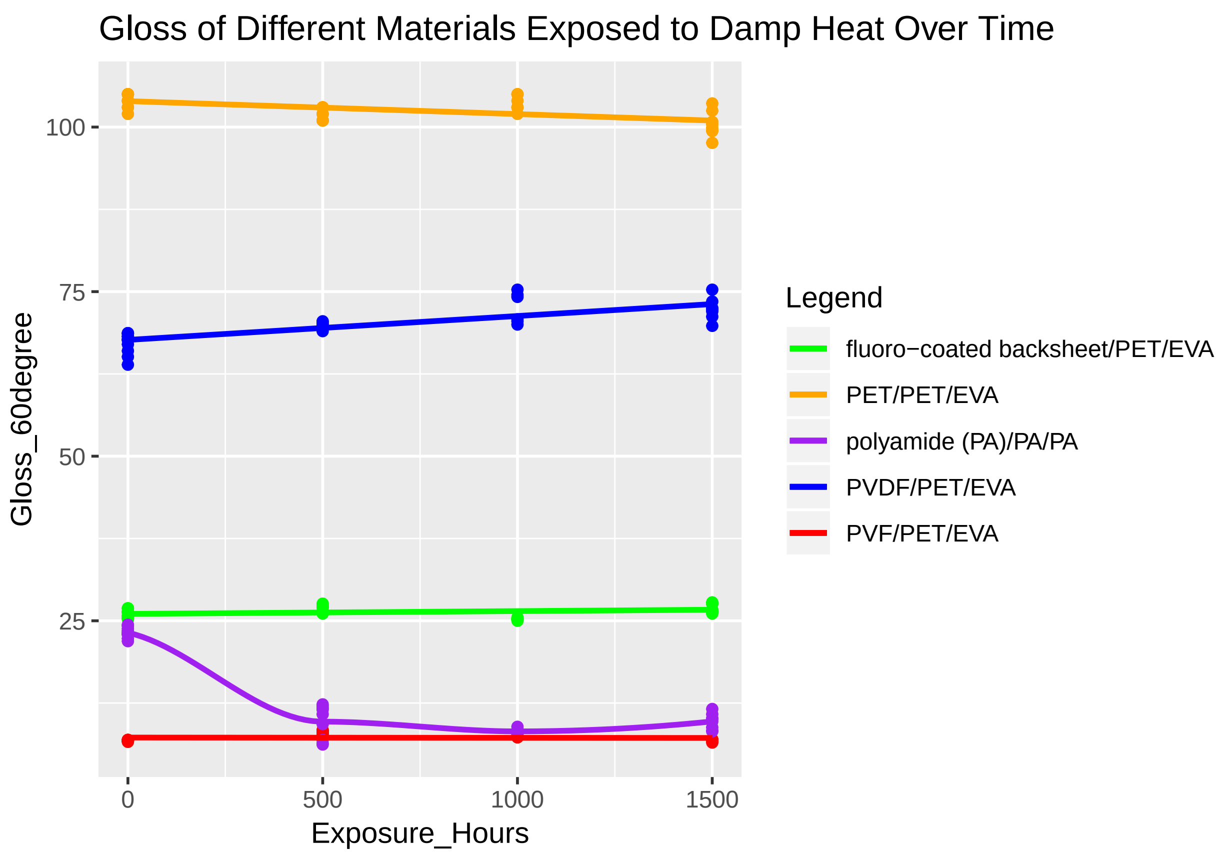
```r
    aes(x = Exposure_Hours,
        y = Yellowness_Index)) +
geom_point(col = "blue")  +
geom_point(data = xenon2sa31002,
           aes(x = Exposure_Hours,
               y = Yellowness_Index),
           col = "green") +
geom_point(data = xenon2sa31003,
           aes(x = Exposure_Hours,
               y = Yellowness_Index),
           col = "orange") +
geom_point(data = xenon2sa31004,
           aes(x = Exposure_Hours,
               y = Yellowness_Index),
           col = "red") +
geom_point(data = xenon2sa31005,
           aes(x = Exposure_Hours,
               y = Yellowness_Index),
           col = "purple") +
ggtitle("Yellowness Index of Different Materials Exposed to Xenon2 Over Time") +
geom_line(data = gglmpiecewisexenon2sa31002,
          aes(x = Exposure_Hours,
              y = Yellowness_Index,
              color = "fluoro-coated backsheet/PET/EVA"))+
stat_smooth(data = xenon2sa31001,
            method = "lm",
            se = FALSE,
            aes(color = "PVDF/PET/EVA")) +
stat_smooth(data = xenon2sa31003,
            method = "lm",
            se = FALSE,
            aes(color = "PET/PET/EVA")) +
stat_smooth(data = xenon2sa31004,
            method = "lm",
            se = FALSE,
            aes(color = "PVF/PET/EVA")) +
geom_line(data = gglmpiecewisexenon2sa31005,
          aes(x = Exposure_Hours,
              y = Yellowness_Index,
              color = "polyamide (PA)/PA/PA")) +
scale_colour_manual(name="Legend",
                    values=c("green",
                             "orange",
                             "purple",
                             "blue",
                             "red"))
```




```r
# For gloss,

# Damp exposure
ggplot(data = dampheatsa31001,
       aes(x = Exposure_Hours,
           y = Gloss_60degree)) +
  geom_point(col = "blue")  +
  geom_point(data = dampheatsa31002,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "green") +
  geom_point(data = dampheatsa31003,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "orange") +
  geom_point(data = dampheatsa31004,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "red") +
  geom_point(data = dampheatsa31005,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "purple") +
  ggtitle("Gloss of Different Materials Exposed to Damp Heat Over Time") +
  stat_smooth(data = dampheatsa31002,
              method = "lm",
```

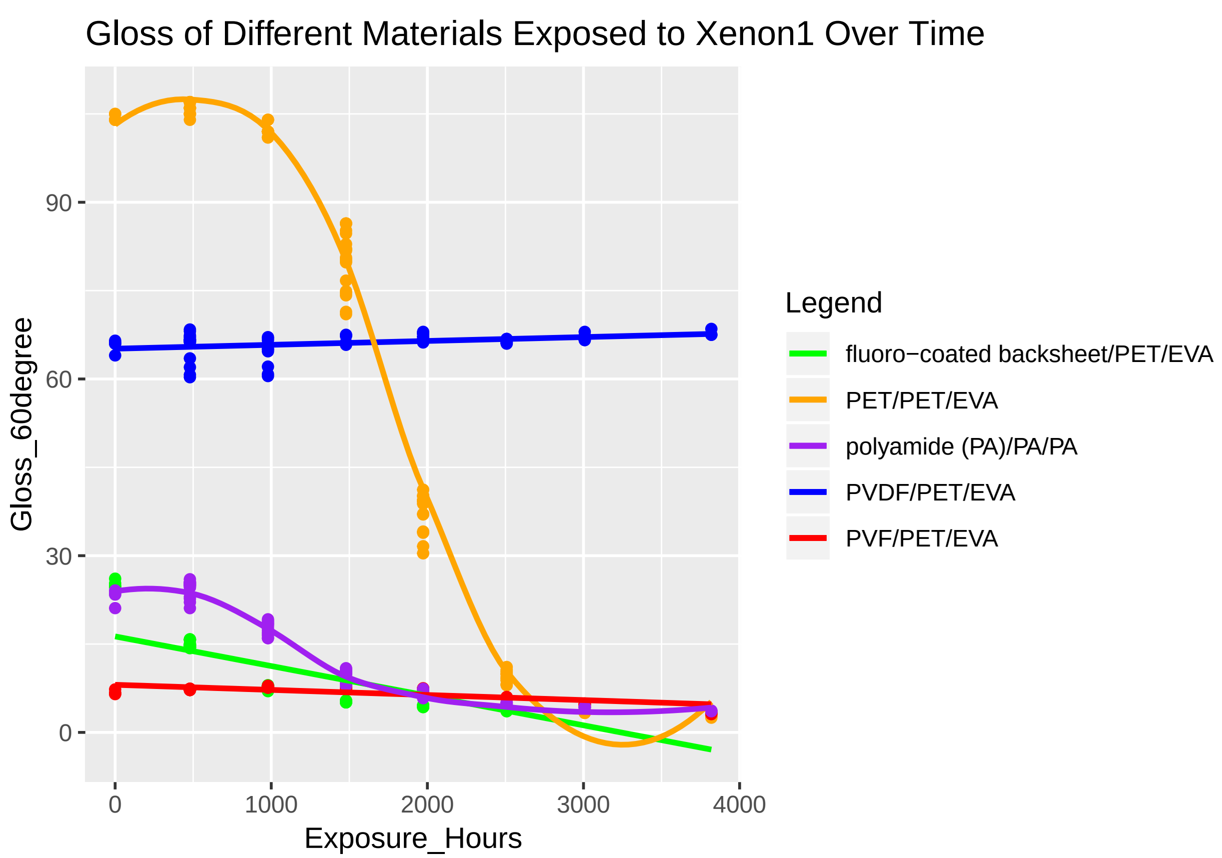
```r
              se = FALSE,
              aes(color = "fluoro-coated backsheet/PET/EVA")) +
  stat_smooth(data = dampheatsa31001,
              method = "lm",
              se = FALSE,
              aes(color = "PVDF/PET/EVA")) +
  stat_smooth(data = dampheatsa31003,
              method = "lm",
              se = FALSE,
              aes(color = "PET/PET/EVA")) +
  stat_smooth(data = dampheatsa31004,
              method = "lm",
              se = FALSE,
              aes(color = "PVF/PET/EVA")) +
  stat_smooth(data = dampheatsa31005,
              method = "loess",
              se = FALSE,
              aes(color = "polyamide (PA)/PA/PA")) +
  scale_colour_manual(name="Legend",
                      values=c("green",
                               "orange",
                               "purple",
                               "blue",
                               "red"))
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -7.5

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1007.5

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 5.5459e-017

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 2.5e+005
```



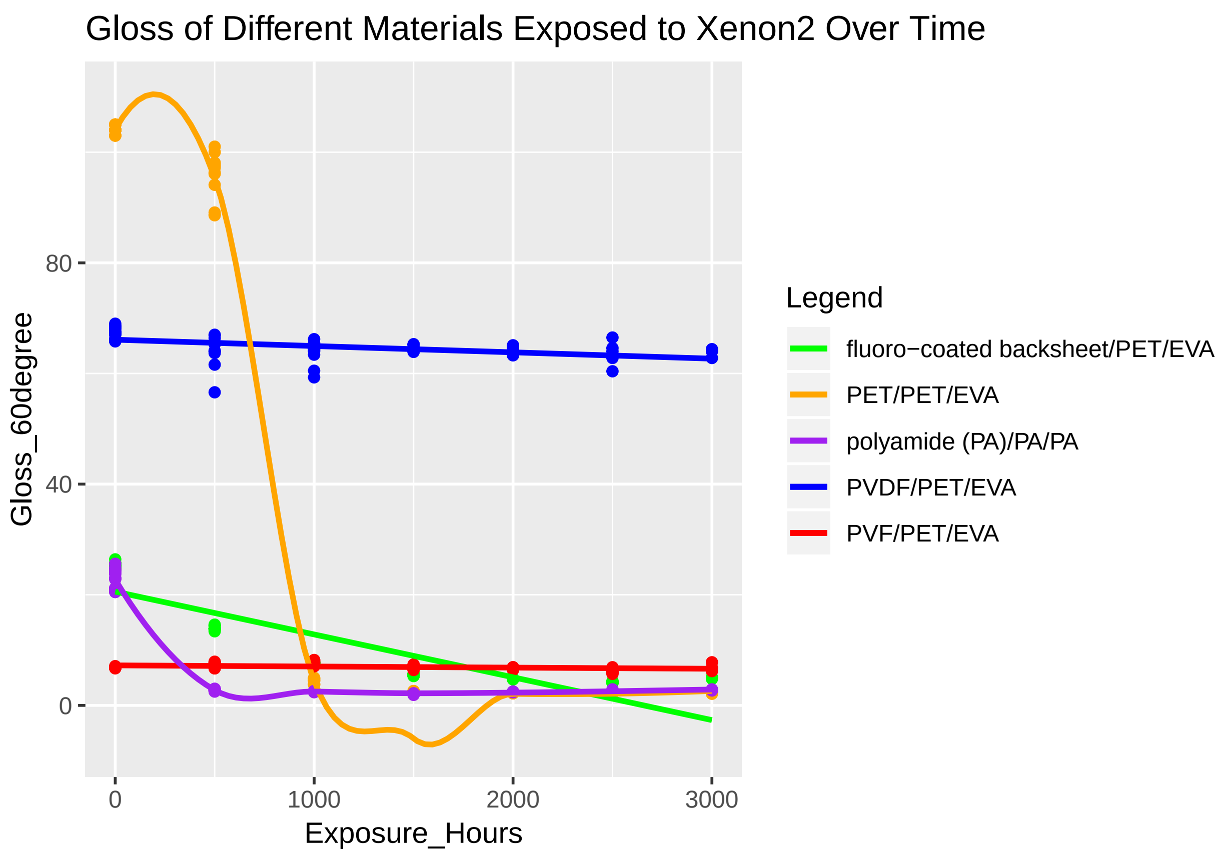

```r
# Xenon1
ggplot(data = xenon1sa31001,
       aes(x = Exposure_Hours,
           y = Gloss_60degree)) +
  geom_point(col = "blue")  +
  geom_point(data = xenon1sa31002,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "green") +
  geom_point(data = xenon1sa31003,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "orange") +
  geom_point(data = xenon1sa31004,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "red") +
  geom_point(data = xenon1sa31005,
             aes(x = Exposure_Hours,
                 y = Gloss_60degree),
             col = "purple") +
  ggtitle("Gloss of Different Materials Exposed to Xenon1 Over Time") +
  stat_smooth(data = xenon1sa31002,
              method = "lm",
              se = FALSE,
              aes(color = "fluoro-coated backsheet/PET/EVA")) +
```

```
  stat_smooth(data = xenon1sa31001, method = "lm",
              se = FALSE,
              aes(color = "PVDF/PET/EVA")) +
  stat_smooth(data = xenon1sa31003,
              method = "loess",
              se = FALSE,
              aes(color = "PET/PET/EVA")) +
  stat_smooth(data = xenon1sa31004, method = "lm",
              se = FALSE,
              aes(color = "PVF/PET/EVA")) +
  stat_smooth(data = xenon1sa31005,
              method = "loess",
              se = FALSE,
              aes(color = "polyamide (PA)/PA/PA")) +
  scale_colour_manual(name="Legend",
                      values=c("green",
                               "orange",
                               "purple",
                               "blue",
                               "red"))
```




```
# Xenon2

ggplot(data = xenon2sa31001,
       aes(x = Exposure_Hours,
           y = Gloss_60degree)) +
```

```
geom_point(col = "blue")  +
geom_point(data = xenon2sa31002,
           aes(x = Exposure_Hours,
               y = Gloss_60degree),
           col = "green") +
geom_point(data = xenon2sa31003,
           aes(x = Exposure_Hours,
               y = Gloss_60degree),
           col = "orange") +
geom_point(data = xenon2sa31004,
           aes(x = Exposure_Hours,
               y = Gloss_60degree),
           col = "red") +
geom_point(data = xenon2sa31005,
           aes(x = Exposure_Hours,
               y = Gloss_60degree),
           col = "purple") +
ggtitle("Gloss of Different Materials Exposed to Xenon2 Over Time") +
stat_smooth(data = xenon2sa31002,
            method = "lm",
            se = FALSE,
            aes(color = "fluoro-coated backsheet/PET/EVA")) +
stat_smooth(data = xenon2sa31001,
            method = "lm",
            se = FALSE,
            aes(color = "PVDF/PET/EVA")) +
stat_smooth(data = xenon2sa31003,
            method = "loess",
            se = FALSE,
            aes(color = "PET/PET/EVA")) +
stat_smooth(data = xenon2sa31004,
            method = "lm",
            se = FALSE,
            aes(color = "PVF/PET/EVA")) +
stat_smooth(data = xenon2sa31005,
            method = "loess",
            se = FALSE,
            aes(color = "polyamide (PA)/PA/PA")) +
scale_colour_manual(name="Legend",
                    values=c("green",
                             "orange",
                             "purple",
                             "blue",
                             "red"))
```




## 2.4 4. For each of the 5 backsheet materials, use Hyperspec and plot the FTIR spectra of each backsheet at the beginning and at the end of each of the 3 exposure type. Discuss what peak changes you note. You may want to normalize the two spectra so their peak amplitudes match for most of the peaks.

```
# Creating a function that normalizes the spectra
normalizeSpectra <- function(spectra) {
  # Finds the maximum amplitude
  maximumamplitude <- max(spectra[, -1])
  normalizedspectra <- spectra
  # Divides the amplitudes by the maximum and thus normalizes it
  normalizedspectra[, -1] <- spectra[, -1]/maximumamplitude
  return(normalizedspectra)
}
# Visual inspection was done to make sure that the maximum spectum was always stationary


# Creating a function to plot the spectra
# Input exposure will be "dampheat", "xenon#1" or "xenon#2"
# Input material will be "PA", "PET" or "PVF"
# Input time will be "beginning" or "end"
# The inputs are not case sensitive
plotSpectra <- function(exposure, material, time) {
```

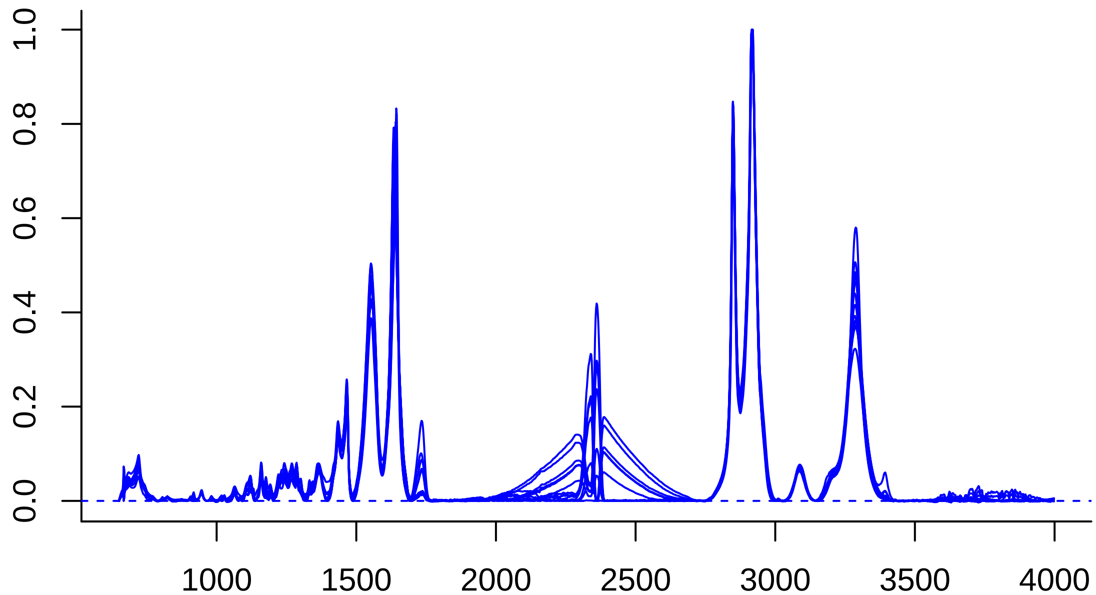
```r
setwd("H:/Git/18f-dsci351-351m-451-axm949/1-assignments/proj/proj3/")
# The chosen material determines the beginning and end exposure hours
if (toupper(exposure) == "DAMPHEAT") {
  path <- "FTIR/DampHeat-FTIR"
  ends <- "1500hr"
  beginning <- "0hr"
} else if (toupper(exposure) == "XENON#1") {
  path <- "FTIR/XENON1-FTIR"
  beginning <- "0HR"
  ends <- "2000HR"
} else if (toupper(exposure) == "XENON#2") {
  path <- "FTIR/XENON2-FTIR"
  ends <- "2000HR"
  beginning <- "0HR"
}

# Gets files of the selected material
materialfilter <- grep(paste0(" ", material, " "),
                       list.files(path = path),
                       value = TRUE, ignore.case = TRUE)
# Gets files towards the end of the exposure
endfiles <- grep(ends, materialfilter,
                 value = TRUE, ignore.case = TRUE)
# Gets files towards the beginning of the exposure
endfiles <- grep(beginning,
                        materialfilter,
                        value = TRUE,
                        ignore.case = TRUE)
# commandforend/beginning will be the text of the command that when executed,
# will combine the spectra

# .end is for spectra at the end of exposure
# .beginning for spectra at the beginning
commandforend <- "collapse("
for (i in endfiles) {
  # Reads the data
  raw <- read.csv(paste(path, i, sep = "/"))
  # Normalizes the spectra
  raw <- normalizeSpectra(raw)
  # Creates a hyperSpec object
  spec <- new("hyperSpec",
              wavelength = raw[, 1],
              spc = t(raw[, -1]))
  # This gets rid of troublesome whitespaces and dashes
  assign(gsub(" |-", "", i), spec)
  # Seperates all arguments by commas besides the last one
  if (i == endfiles[1]) {
    separator <- ""
  } else {
    separator <- ", "
  }
  # Combines text for the command
  commandforend <- paste(commandforend, gsub(" |-", "", i),
```

```r
                            sep = separator)
  }
  # Closes the command's text with a parenthesis
  commandforend <- paste(commandforend, ")")
  # Combines multiple spectra into a single file
  spectraend <- eval(parse(text = commandforend))

  # Now for spectra in the beginning

  commandforbeginning <- "collapse("
  for (j in endfiles) {
    # Reads the raw data
    raw <- read.csv(paste(path, j, sep = "/"))
    # Normalizes the spectra
    raw <- normalizeSpectra(raw)
    # Creates a hyperSpec object
    spec <- new("hyperSpec",
                wavelength = raw[, 1],
                spc = t(raw[, -1]))
    assign(gsub(" |-", "", j), spec)
    # Seperates all the arguments besides the final one with commas
    if (j == endfiles[1]) {
      separator <- ""
    } else {
      separator <- ", "
    }
    # Combines text for the command
    commandforbeginning <- paste(commandforbeginning,
                                  gsub(" |-", "", j),
                                sep = separator)
  }
  # Closes the command's text with a parenthesis
  commandforbeginning <- paste(commandforbeginning, ")")
  # Combines multiple spectra into a single file
  spectrabeginning <- eval(parse(text = commandforbeginning))

  if (toupper(time) == "END"){
    # Plots spectra at the end of the exposure
    hyperSpec::plotspc(spectraend, col = "red")
} else if (toupper(time) == "BEGINNING") {
    # Plots spectra at the beginning of the exposure
    hyperSpec::plotspc(spectrabeginning, col = "blue")
}
}
```

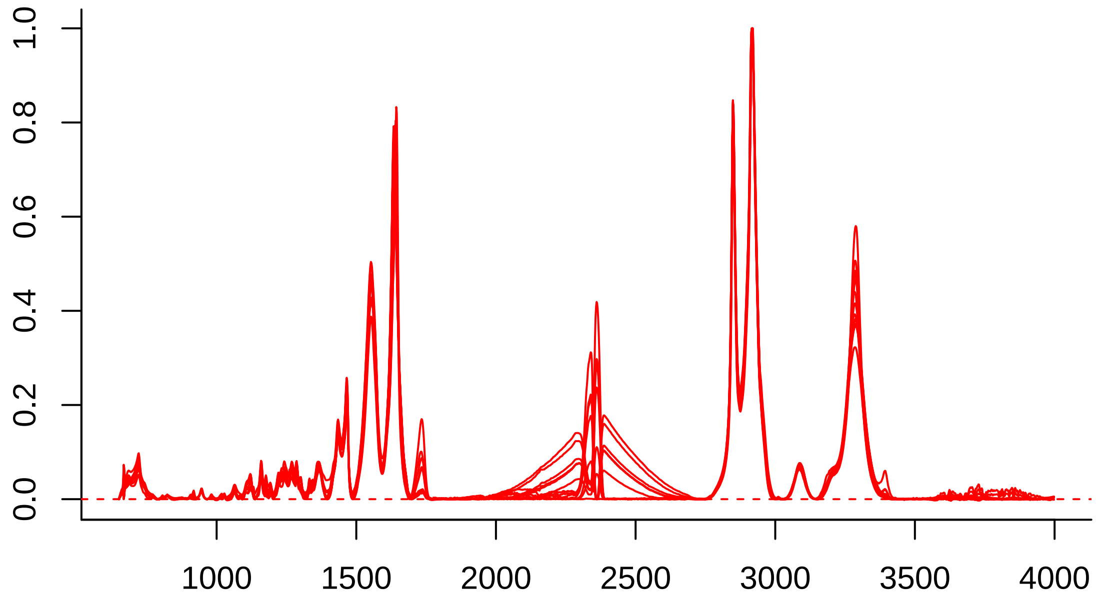
```r
# Creates an array containing different backsheets
materials <- c("PA", "PET", "PVF")
# Creates an array containing the times which we desire
times <- c("beginning", "end")
for (i in exposures) {  # For the ith exposure
  for (j in materials) {  # For the jth material
    for (k in times) {  # for the kth time (end or beginning)
      # prints the title of the graph
      print(paste("Spectra plot for",
```

```
                  j, "in",
                  i, "at the", k))
      # Plots the spectra
      suppressWarnings(plotSpectra(i, j, k))
    }
  }
}
```

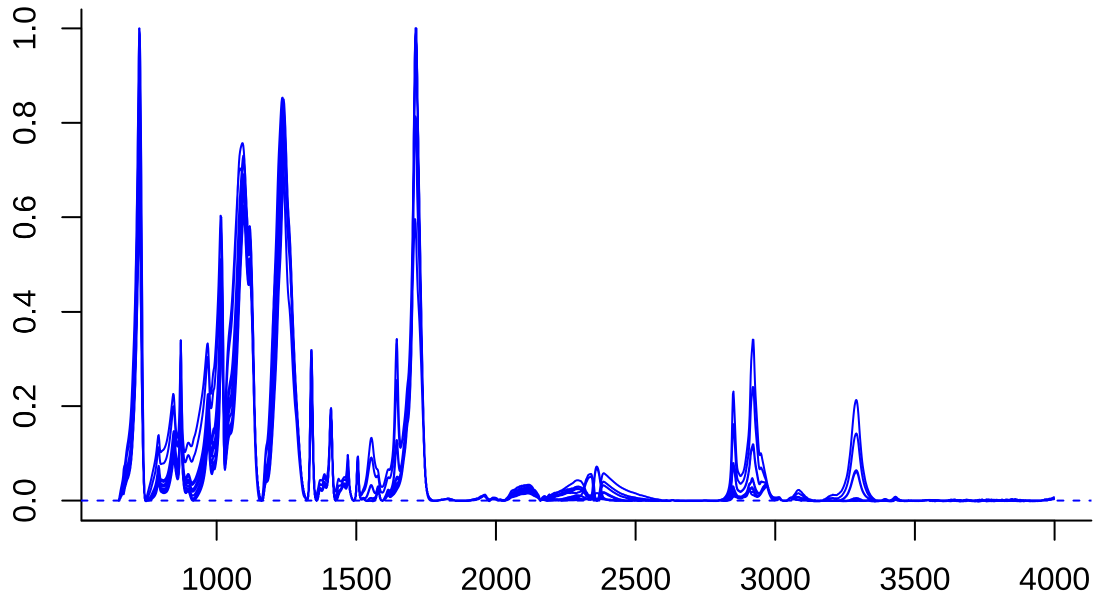
```
## [1] "Spectra plot for PA in DampHeat at the beginning"
```



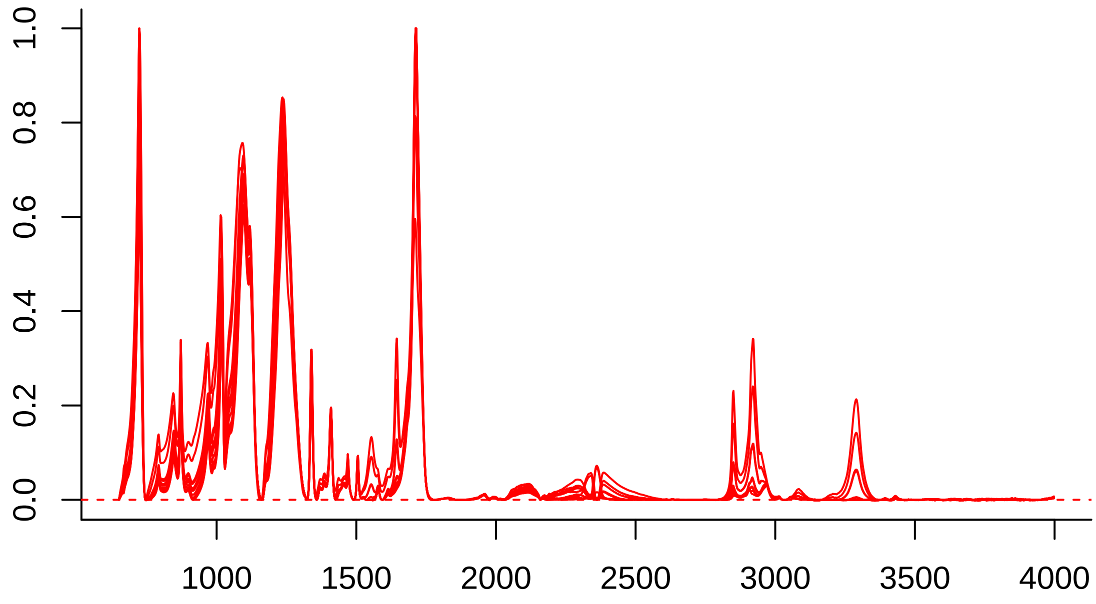

```
## [1] "Spectra plot for PA in DampHeat at the end"
```



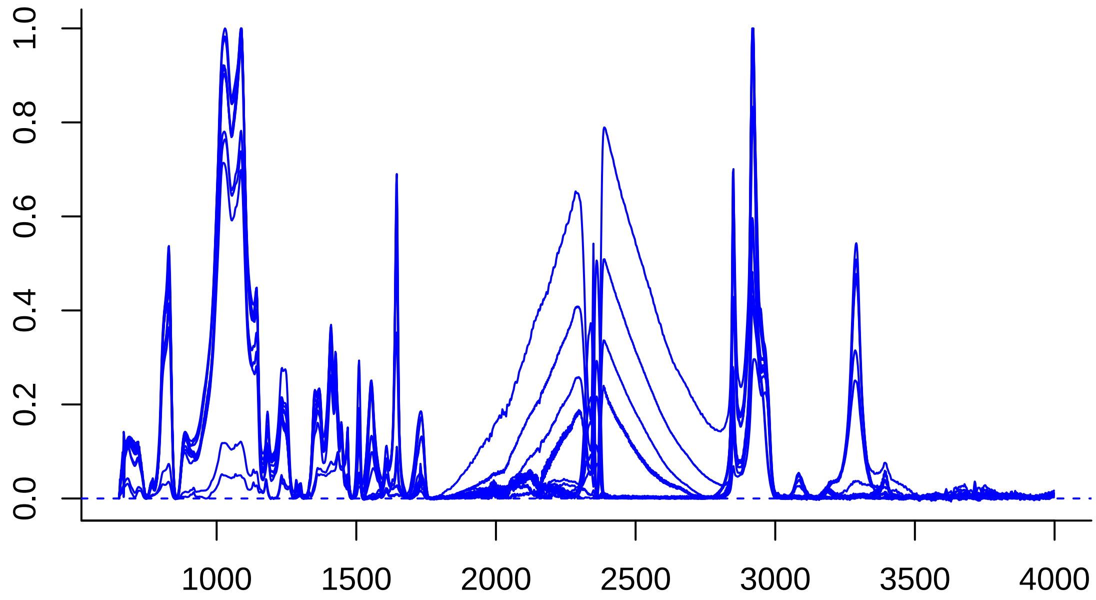

```
## [1] "Spectra plot for PET in DampHeat at the beginning"
```



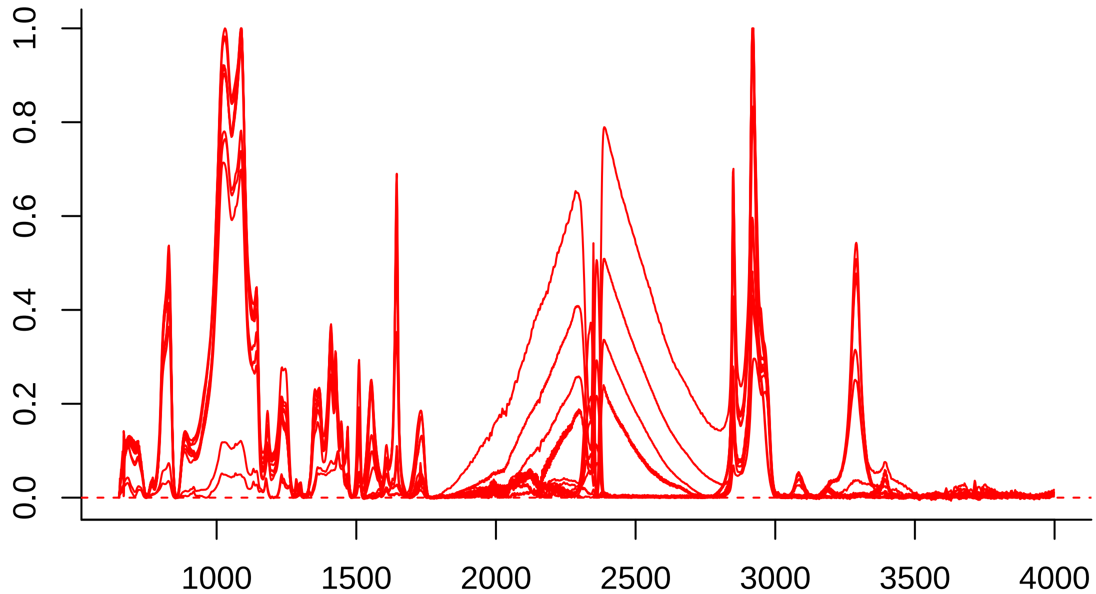

```
## [1] "Spectra plot for PET in DampHeat at the end"
```



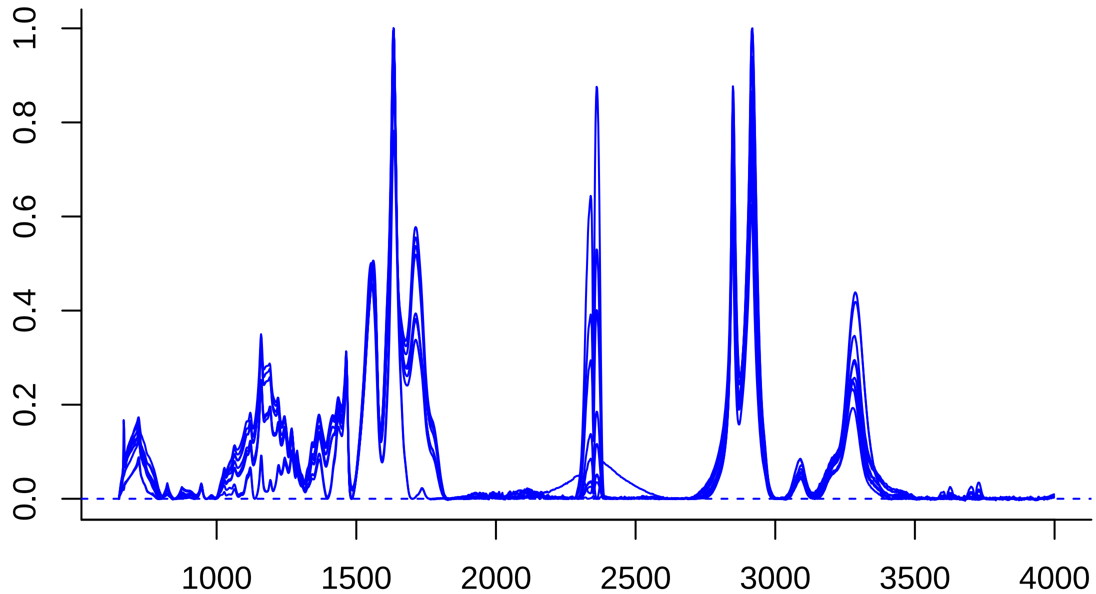

```
## [1] "Spectra plot for PVF in DampHeat at the beginning"
```



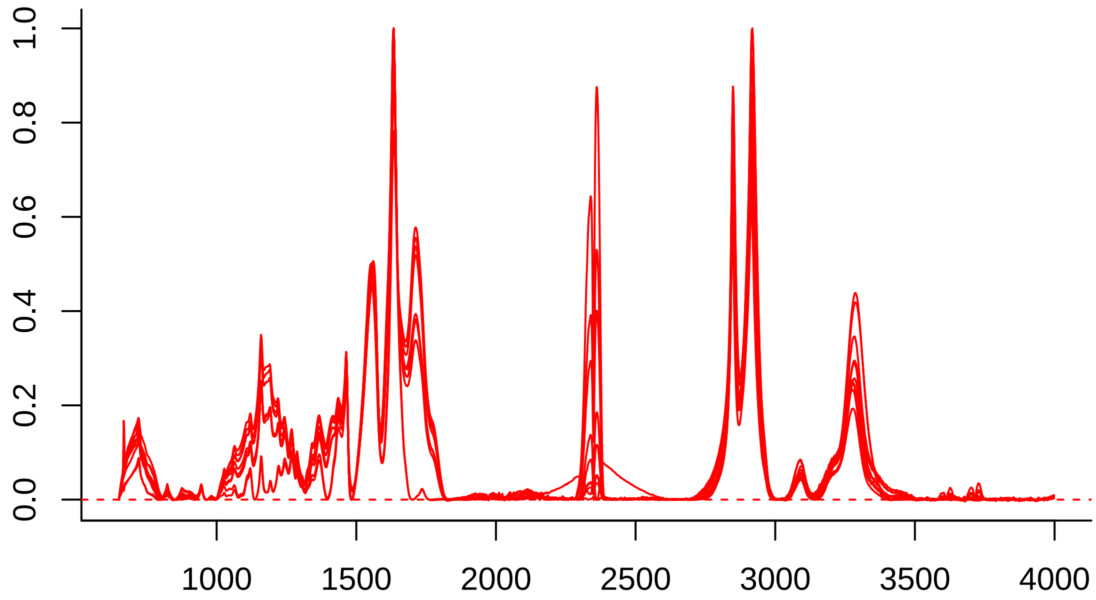

```
## [1] "Spectra plot for PVF in DampHeat at the end"
```



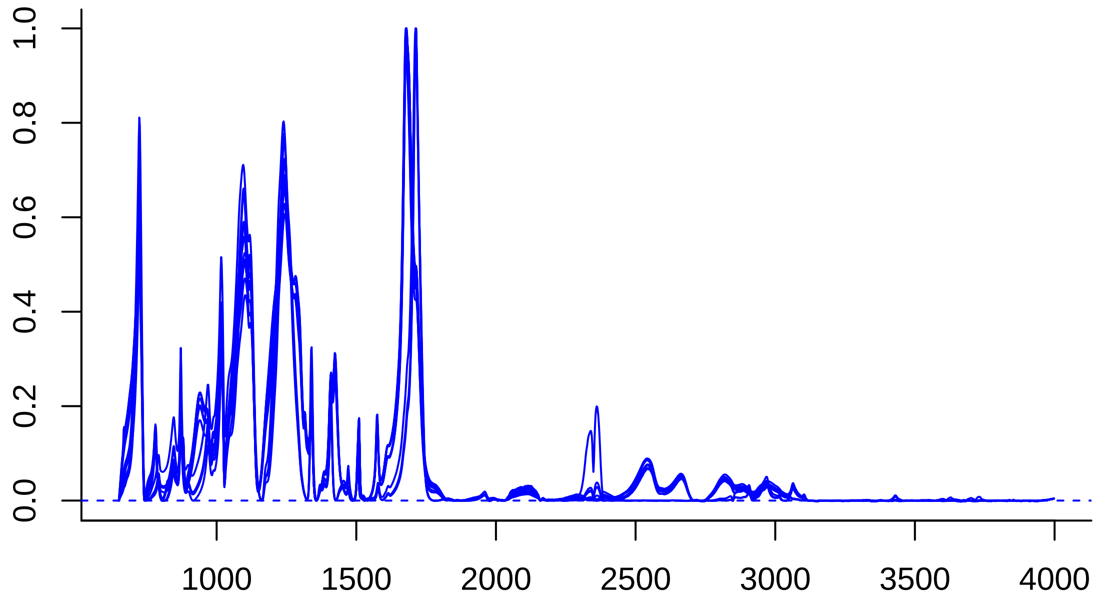

```
## [1] "Spectra plot for PA in Xenon#1 at the beginning"
```



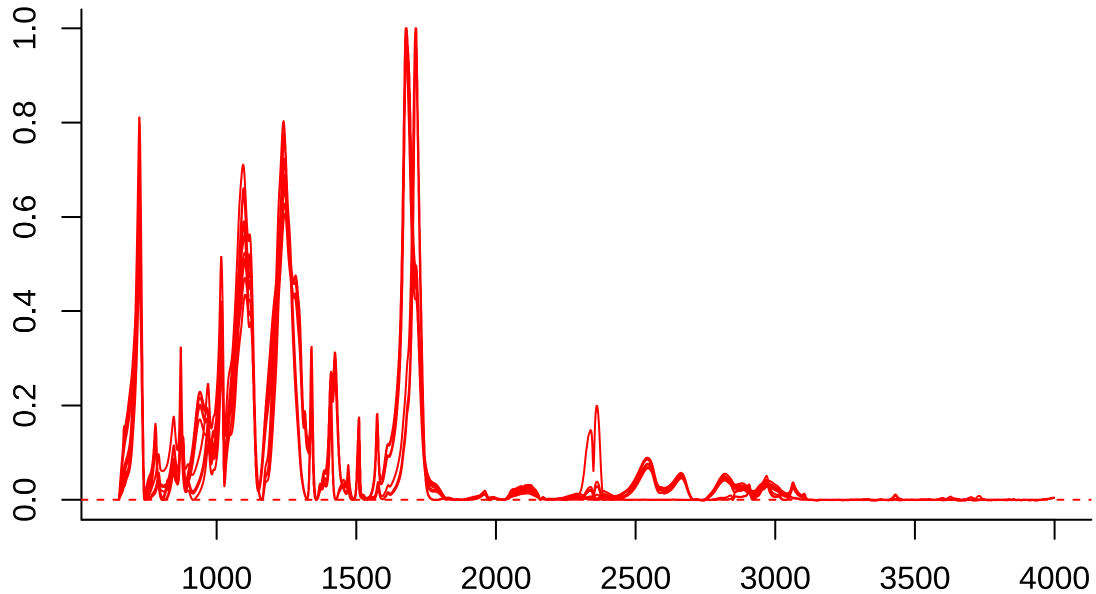

```
## [1] "Spectra plot for PA in Xenon#1 at the end"
```



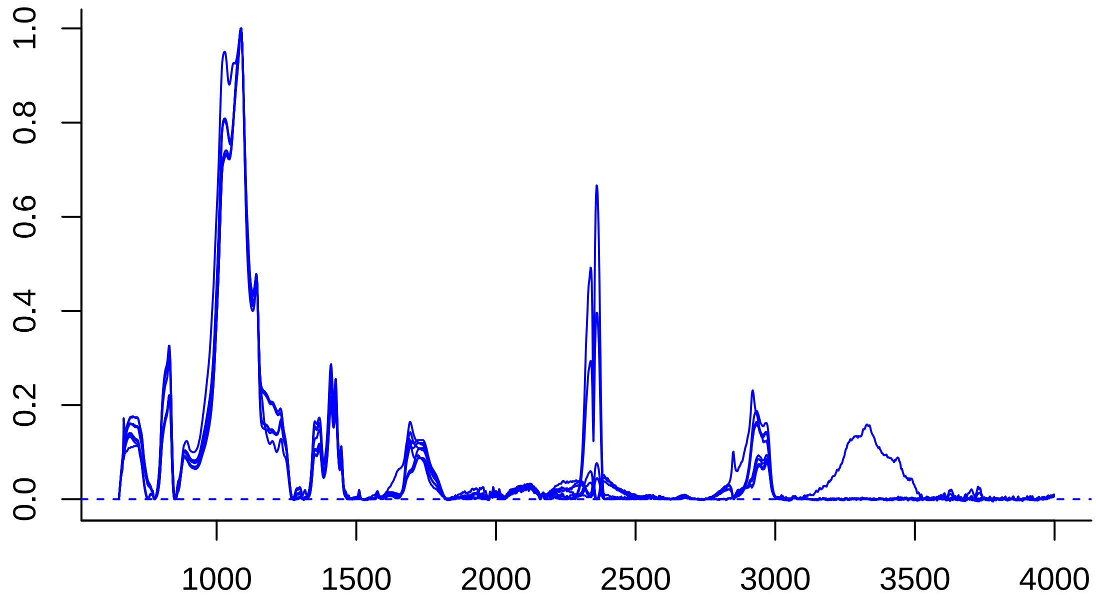

```
## [1] "Spectra plot for PET in Xenon#1 at the beginning"
```



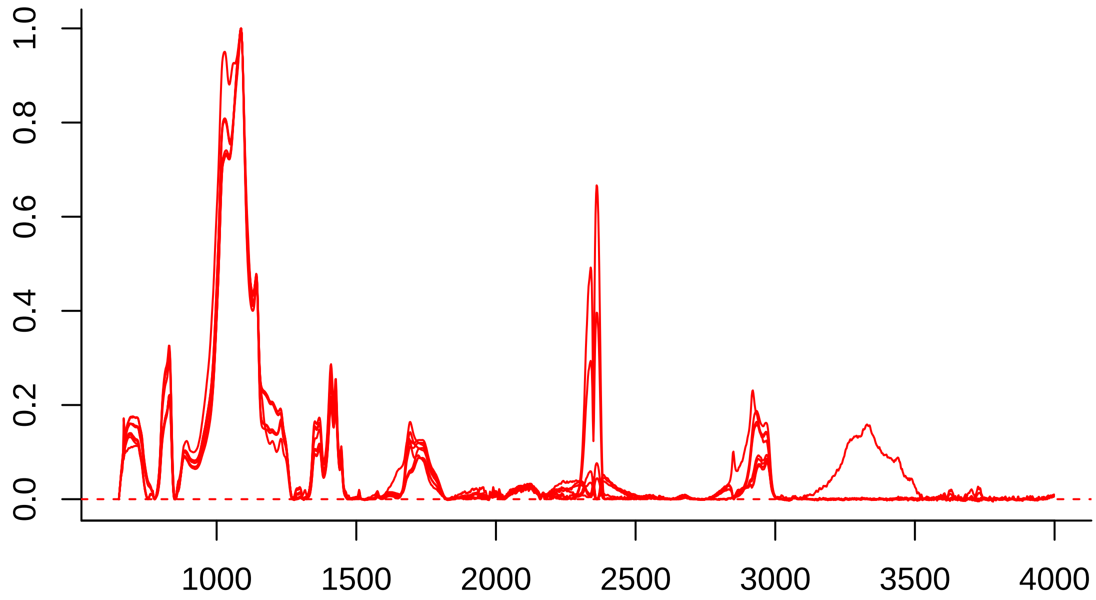

```
## [1] "Spectra plot for PET in Xenon#1 at the end"
```



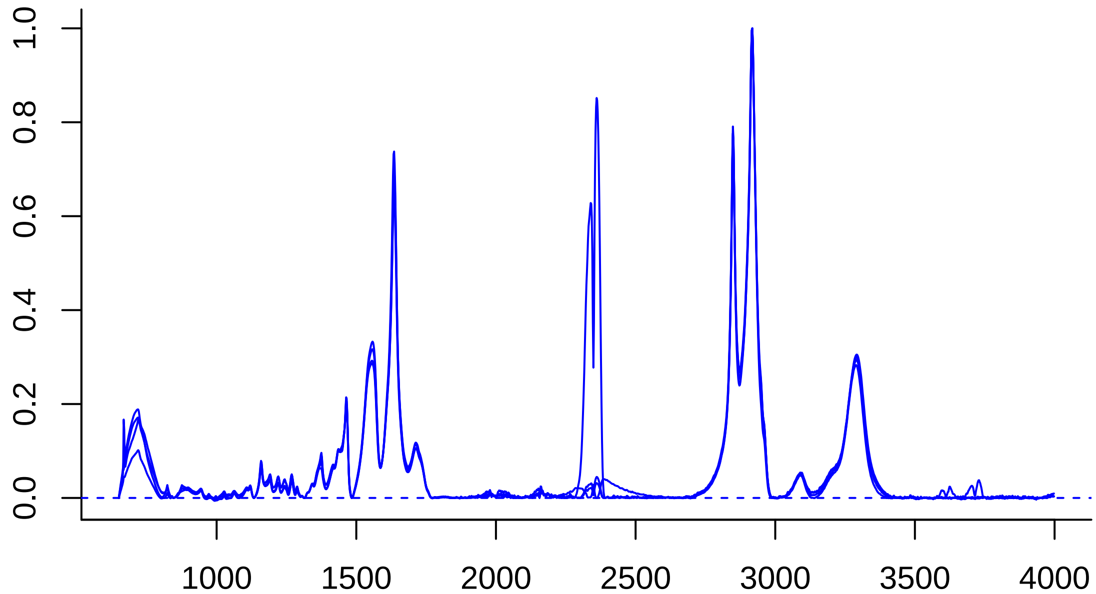

```
## [1] "Spectra plot for PVF in Xenon#1 at the beginning"
```



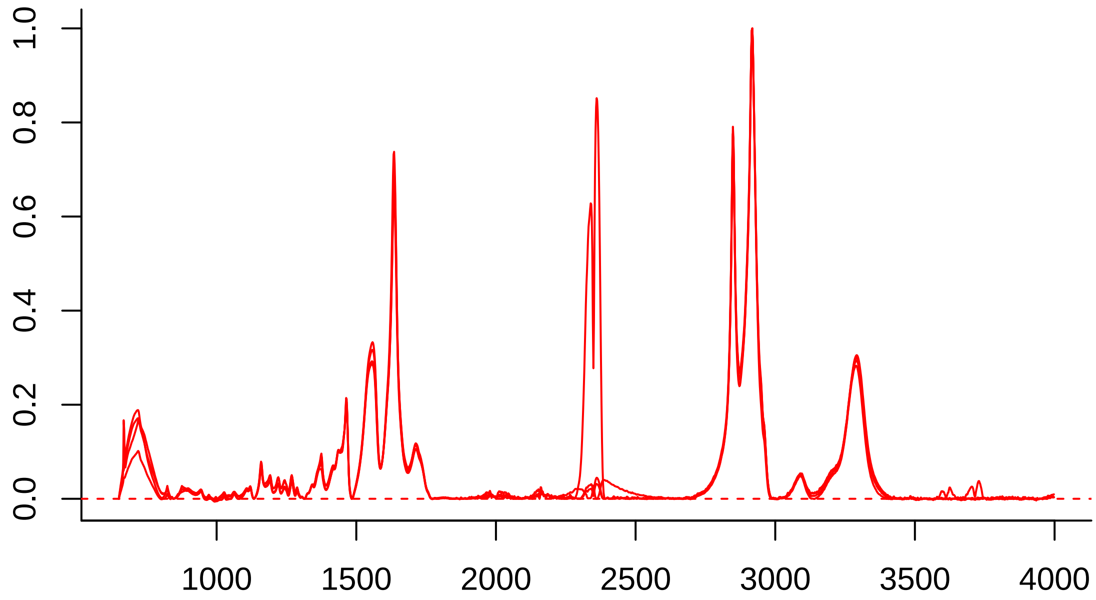

```
## [1] "Spectra plot for PVF in Xenon#1 at the end"
```



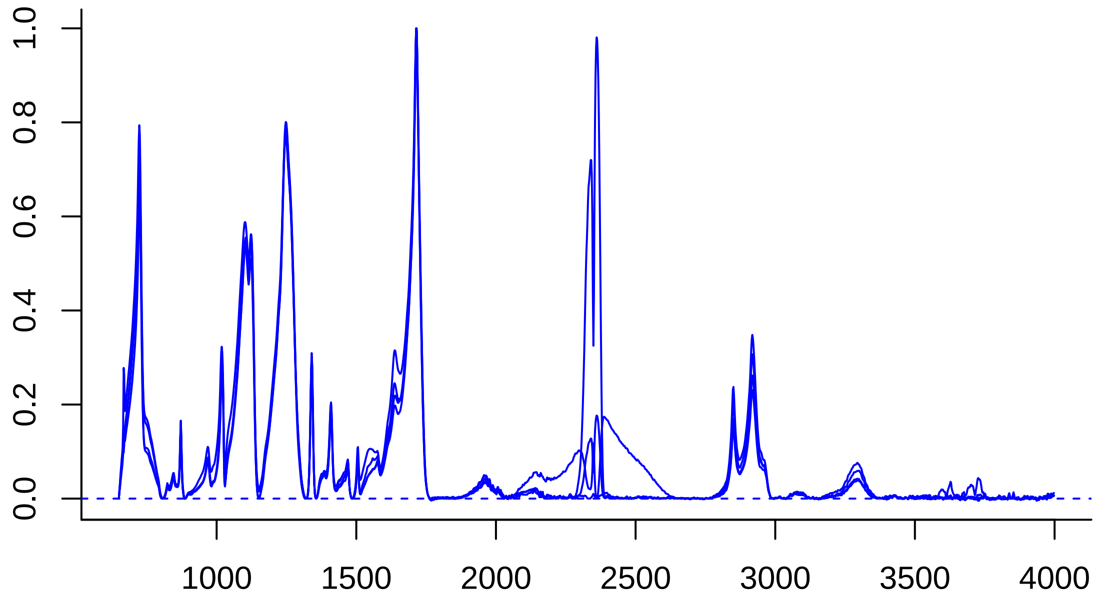

```
## [1] "Spectra plot for PA in Xenon#2 at the beginning"
```



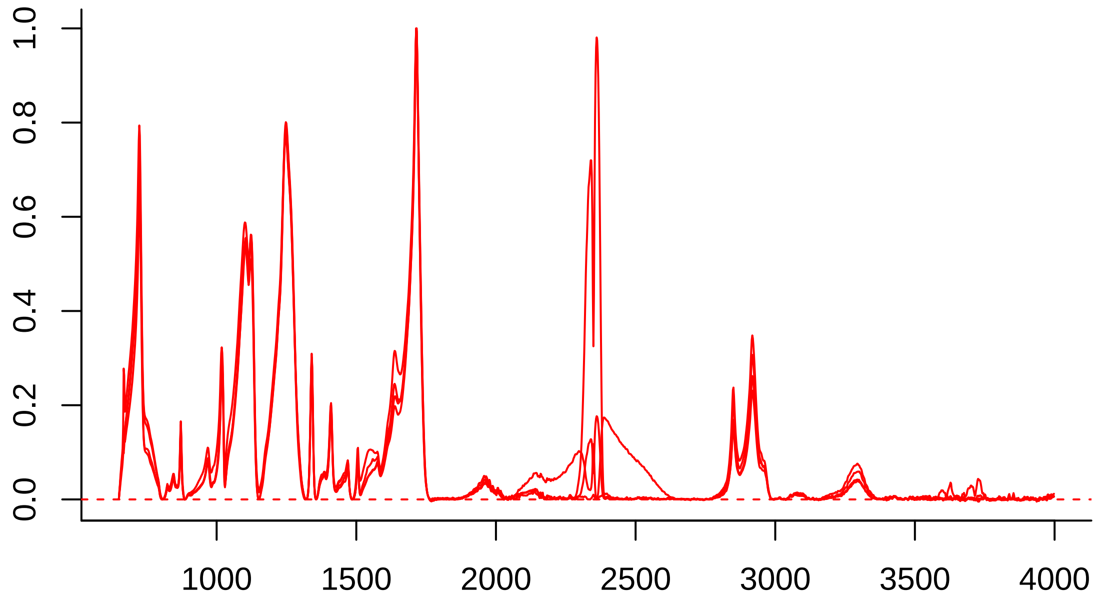

```
## [1] "Spectra plot for PA in Xenon#2 at the end"
```



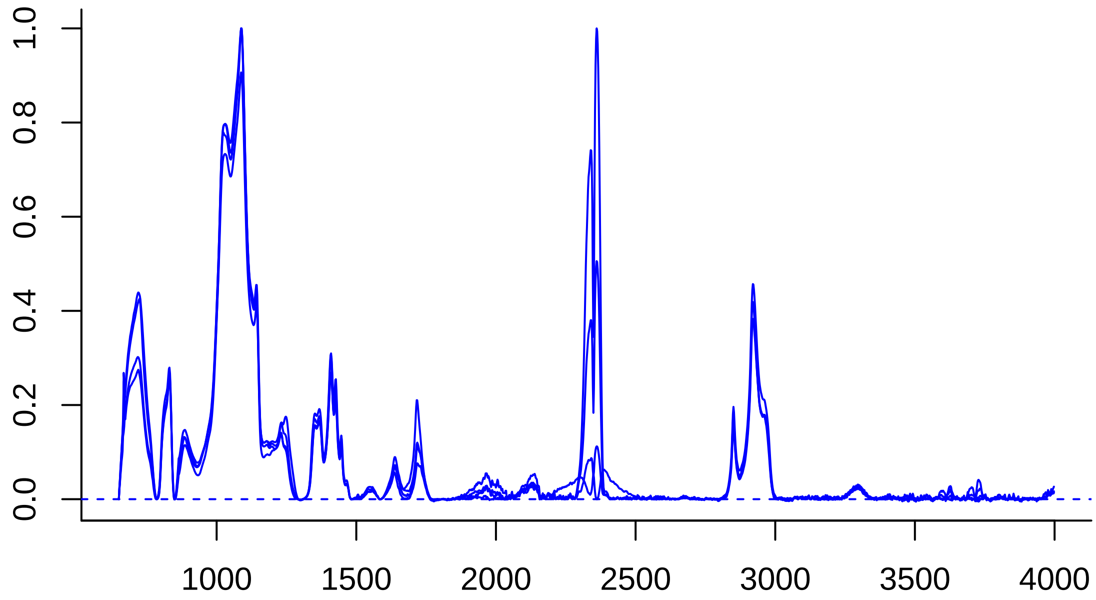

```
## [1] "Spectra plot for PET in Xenon#2 at the beginning"
```



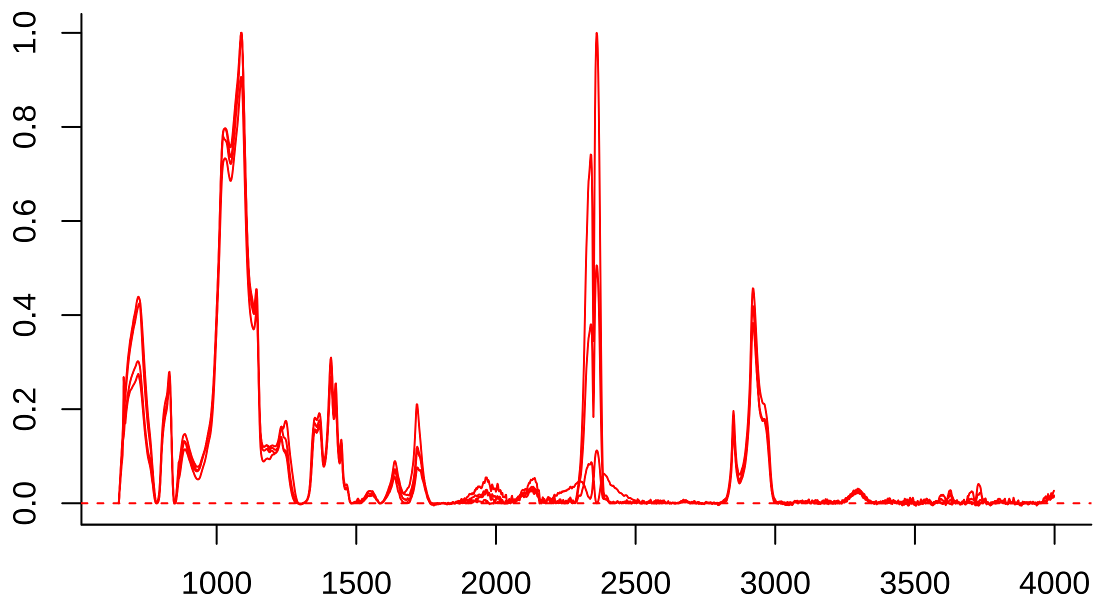

```
## [1] "Spectra plot for PET in Xenon#2 at the end"
```




```
## [1] "Spectra plot for PVF in Xenon#2 at the beginning"
```




```
## [1] "Spectra plot for PVF in Xenon#2 at the end"
```




## 2.5 5. Explain which backsheet you feel is the best overall performer based on this data set

```
# The final score is calculated by adding the ranks at each exposure
overallscore <- dampheatscore + xenon1score + xenon2score
# Displays the final scores for all materials
for (i in 1:length(overallscore)){
  print(paste("Final score for",
              returnMaterial(samples[i]),
              samples[i], ":", overallscore[i]))
}
```

```
## [1] "Final score for PVDF/PET/EVA sa31001 : 16"
## [1] "Final score for fluoro-coated backsheet/PET/EVA sa31002 : 17"
## [1] "Final score for PET/PET/EVA sa31003 : 27"
## [1] "Final score for PVF/PET/EVA sa31004 : 9"
## [1] "Final score for polyamide (PA)/PA/PA sa31005 : 21"
```

```
# This means PVF/PET/EVA(sa31004) is the best overall performer according to me.
```