

```
## This section of the code is used to sort particles into broader  
polymer classes
```

```
# To use this code, you will need a csv file with the following columns.  
Make sure to name the columns exactly as shown below. Do not use spaces  
or special characters or change the case of the letters in the column  
names. If you do, the code will not work as intended. The column names in  
order are Particle_ID, Identification, HQI, Width_in_microns,  
Height_in_microns, and Area_in_square_microns
```

```
# If you don't already have the tidyverse package installed in R, do so  
with the following line of code  
install.packages("tidyverse")
```

```
# Copy and paste the following line of code to load the tidyverse package  
library(tidyverse)
```

```
# Copy and paste the following code to select a csv file to edit  
MPs=read.csv(file.choose(), header = T)
```

```
# Set the data frame  
df <- as_tibble(MPs)
```

```
# Copy and paste the following code to group into polymer categories  
MPs_mod <- df %>%
```

```
  mutate(Polymer = case_when(  
    grepl("LCP/PTFE", Identification) ~ "LCP/PTFE",  
    grepl("LCP/PPS", Identification) ~ "LCP/PPS",  
    grepl("PA46/PTFE", Identification) ~ "PA/PTFE",  
    grepl("PA6/ABS", Identification) ~ "PA/ABS",  
    grepl("PA6/ASA", Identification) ~ "PA/ASA",  
    grepl("PA6/PET", Identification) ~ "PA/PET",  
    grepl("PA6/PP", Identification) ~ "PA/PP",  
    grepl("PA66/PE", Identification) ~ "PA/PE",  
    grepl("PA66/PTFE", Identification) ~ "PA/PTFE",  
    grepl("PBT/ASA", Identification) ~ "PBT/Acrylate",  
    grepl("PBT/PET", Identification) ~ "PBT/PET",  
    grepl("PBT/PTFE", Identification) ~ "PBT/PTFE",  
    grepl("PC/ABS", Identification) ~ "PC/ABS",  
    grepl("PC/ASA", Identification) ~ "PC/Acrylate",  
    grepl("PC/PBT", Identification) ~ "PC/PBT",  
    grepl("PC/PET", Identification) ~ "PC/PET",  
    grepl("PC/SAN", Identification) ~ "PC/SAN",  
    grepl("PC/Siloxan", Identification) ~ "PC/Siloxan",  
    grepl("PC/PTFE", Identification) ~ "PC/PTFE",  
    grepl("PE/EVA", Identification) ~ "PE/EVA",  
    grepl("PEEK/PTFE", Identification) ~ "PEEK/PTFE",  
    grepl("PEI/PTFE", Identification) ~ "PEI/PTFE",  
    grepl("PESU/PTFE", Identification) ~ "PESU/PTFE",  
    grepl("PLA/PC", Identification) ~ "PLA/PC",  
    grepl("POM/PDMS", Identification) ~ "POM/PDMS",  
    grepl("POM/PTFE", Identification) ~ "POM/PTFE",  
    grepl("PP/EPDM", Identification) ~ "PP/EPDM",  
    grepl("PP/EVA", Identification) ~ "PP/EVA",
```

```

grepl("PP/TPE", Identification) ~ "PP/TPE",
grepl("PP/TPO", Identification) ~ "PP/TPO",
grepl("PPA/PTFE", Identification) ~ "PPA/PTFE",
grepl("PPE/PA/PS", Identification) ~ "PPE/PA/PS",
grepl("PPE/PS", Identification) ~ "PPE/PS",
grepl("Polycarbonate/Acrylonitrile Butadiene Styrene Blend",
Identification, ignore.case=TRUE) ~ "PC/ABS",
grepl("Polyvinyl Chloride-Phenolic-Nitrile Butadiene Rubber",
Identification, ignore.case=TRUE) ~ "PVC/P/NBR",
grepl("PVC-P-NBR", Identification) ~ "PVC/P/NBR",
grepl("Blend of Polypropylen and Ethylen / Propylen",
Identification, ignore.case=TRUE) ~ "PP/EPR",
grepl("Polyphenylene Ether + High Impact Polystyrene",
Identification, ignore.case=TRUE) ~ "PPE/PS",
grepl("PPE + HIPS", Identification) ~ "PPE/PS",
grepl("Blend of Polyphenylene Ether And Polyamide", Identification,
ignore.case=TRUE) ~ "PA/PE",
grepl("PP + PA", Identification) ~ "PA/PP",
grepl("Polycarbonate / Polybutylene Terephthalate", Identification,
ignore.case=TRUE) ~ "PC/PBT",
grepl("PC + PBT", Identification) ~ "PC/PBT",
grepl("Acrylonitrile Styrene Acrylate", Identification,
ignore.case=TRUE) ~ "Acrylate",
grepl("Styrene Acrylonitrile", Identification, ignore.case=TRUE) ~
"SAN",
grepl("Sheet Molding Compound", Identification, ignore.case=TRUE) ~
"SMC",
grepl("Polyphenylene Ether", Identification, ignore.case=TRUE) ~
"PPE",
grepl("ACRYLAT HOTMELT", Identification) ~ "Acrylate",
grepl("Styrene Maleic Anhydride Copolymer", Identification,
ignore.case=TRUE) ~ "SMA",
grepl("PCTFE", Identification) ~ "PCTFE",
grepl("acrylonitrile butadiene styrene", Identification,
ignore.case=TRUE) ~ "ABS",
grepl("acrylic", Identification, ignore.case=TRUE) ~ "Acrylic",
grepl("polymethylmethacrylate", Identification, ignore.case=TRUE) ~
"Acrylic",
grepl("cellulose acetate", Identification, ignore.case=TRUE) ~
"Cellulose acetate",
grepl("ethylene vinylacetate", Identification, ignore.case=TRUE) ~
"Ethylene Vinyl Acetate",
grepl("ethylene vinyl acetate", Identification, ignore.case=TRUE) ~
"Ethylene Vinyl Acetate",
grepl("ethylene vinyl alcohol", Identification, ignore.case=TRUE) ~
"Ethylene Vinyl Alcohol",
grepl("ethylene vinylalcohol", Identification, ignore.case=TRUE) ~
"Ethylene Vinyl Alcohol",
grepl("nylon", Identification, ignore.case=TRUE) ~ "PA",
grepl("polyamide", Identification, ignore.case=TRUE) ~ "PA",
grepl("polyacrylonitrile", Identification, ignore.case=TRUE) ~
"PAN",
grepl("polybutylene terephthalate", Identification,
ignore.case=TRUE) ~ "PBT",

```

```

    grepl("polycarbonate", Identification, ignore.case=TRUE) ~ "PC",
    grepl("polyether ether ketone", Identification, ignore.case=TRUE) ~
"PEEK",
    grepl("polyetheretherketone", Identification, ignore.case=TRUE) ~
"PEEK",
    grepl("polyester", Identification, ignore.case=TRUE) ~ "PET",
    grepl("polyethylene terephthalate", Identification,
ignore.case=TRUE) ~ "PET",
    grepl("polyethylenterephthalate", Identification, ignore.case=TRUE)
~ "PET",
    grepl("POLY(ETHYLENE TEREPHTHALATE)", Identification,
ignore.case=TRUE) ~ "PET",
    grepl("polylactic acid", Identification, ignore.case=TRUE) ~ "PLA",
    grepl("polylactide", Identification, ignore.case=TRUE) ~ "PLA",
    grepl("polyoxymethylene", Identification, ignore.case=TRUE) ~
"POM",
    grepl("polypropylene", Identification, ignore.case=TRUE) ~ "PP",
    grepl("POLY(PROPYLENE)", Identification, ignore.case=TRUE) ~ "PP",
    grepl("polyphenylene sulfone", Identification, ignore.case=TRUE) ~
"PPSU",
    grepl("Polyphenyl sulfone", Identification, ignore.case=TRUE) ~
"PPSU",
    grepl("polystyrene", Identification, ignore.case=TRUE) ~ "PS",
    grepl("polysulfone", Identification, ignore.case=TRUE) ~ "PSU",
    grepl("polyurethane", Identification, ignore.case=TRUE) ~ "PU",
    grepl("Poly(vinyl alcohol)", Identification, ignore.case=TRUE) ~
"PVA",
    grepl("Polyvinyl alcohol", Identification, ignore.case=TRUE) ~
"PVA",
    grepl("Thermoplastic Polyurethane", Identification,
ignore.case=TRUE) ~ "PU",
    grepl("polyvinyl chloride", Identification, ignore.case=TRUE) ~
"PVC",
    grepl("polyvinylchloride", Identification, ignore.case=TRUE) ~
"PVC",
    grepl("structural reaction-injection moulding", Identification,
ignore.case=TRUE) ~ "S-RIM",
    grepl("polyethylene", Identification, ignore.case=TRUE) ~ "PE",
    grepl("poly(ethylene)", Identification, ignore.case=TRUE) ~ "PE",
    grepl("POLY(ETHYLENE), low density", Identification,
ignore.case=TRUE) ~ "PE",
    grepl("POLY(ETHYLENE OXIDE)", Identification, ignore.case=TRUE) ~
"PEG",
    grepl("rubber", Identification, ignore.case=TRUE) ~ "Rubber",
    grepl("KAUTSCHUK HOTMELT", Identification) ~ "Rubber",
    grepl("POLY(DIMETHYLSILOXANE)", Identification, ignore.case=TRUE) ~
"Silicone",
    grepl("silicone", Identification, ignore.case=TRUE) ~ "Silicone",
    grepl("PMMA", Identification) ~ "Acrylic",
    grepl("PEVA", Identification) ~ "Ethylene Vinyl Acetate",
    grepl("EVAc", Identification) ~ "Ethylene Vinyl Acetate",
    grepl("EVOH", Identification) ~ "Ethylene Vinyl Alcohol",
    grepl("POLY(VINYL ALCOHOL)", Identification) ~ "PVOH",
    grepl("PEEK", Identification) ~ "PEEK",

```

```

grepl("PTFE", Identification) ~ "PTFE",
grepl("PARA", Identification) ~ "Polyarylamide",
grepl("PPSU", Identification, ignore.case=TRUE) ~ "PPSU",
grepl("S-RIM", Identification) ~ "S-RIM",
grepl("tire", Identification, ignore.case=TRUE) ~ "Rubber",
grepl("EPDM", Identification) ~ "Rubber",
grepl("XNBR", Identification) ~ "Rubber",
grepl("HNBR", Identification) ~ "Rubber",
grepl("PEBA", Identification) ~ "PEBA",
grepl("PVDF", Identification) ~ "PVDF",
grepl("SMMA", Identification) ~ "Acrylate",
grepl("EFEP", Identification) ~ "EFEP",
grepl("PESU", Identification) ~ "PESU",
grepl("ETFE", Identification) ~ "ETFE",
grepl("MABS", Identification) ~ "MABS",
grepl("SEBS", Identification) ~ "SEBS",
grepl("ABS", Identification) ~ "ABS",
grepl("EAA", Identification) ~ "Ethylene Acrylic Acid",
grepl("EVA", Identification) ~ "Ethylene Vinyl Acetate",
grepl("PEA", Identification) ~ "Ethylene Vinyl Acetate",
grepl("PAN", Identification) ~ "PAN",
grepl("PBT", Identification) ~ "PBT",
grepl("PET", Identification) ~ "PET",
grepl("PLA", Identification) ~ "PLA",
grepl("POM", Identification) ~ "POM",
grepl("PSU", Identification) ~ "PSU",
grepl("PUR", Identification) ~ "PU",
grepl("TPU", Identification) ~ "PU",
grepl("PVC", Identification) ~ "PVC",
grepl("ECO", Identification) ~ "Rubber",
grepl("SBR", Identification) ~ "Rubber",
grepl("NBR", Identification) ~ "Rubber",
grepl("EVM", Identification) ~ "Rubber",
grepl("IIR", Identification) ~ "Rubber",
grepl("MVQ", Identification) ~ "Rubber",
grepl("FKM", Identification) ~ "Rubber",
grepl("ACM", Identification) ~ "Rubber",
grepl("FVMQ", Identification) ~ "Silicone",
grepl("TFM", Identification) ~ "Modified PTFE",
grepl("SMA", Identification) ~ "SMA",
grepl("OBC", Identification) ~ "OBC",
grepl("CSM", Identification) ~ "CSM",
grepl("AEM", Identification) ~ "Acrylate",
grepl("LCP", Identification) ~ "LCP",
grepl("PHA", Identification) ~ "PHA",
grepl("TPC", Identification) ~ "TPC",
grepl("PFA", Identification) ~ "PFA",
grepl("PVA", Identification) ~ "PVA",
grepl("MFA", Identification) ~ "MFA",
grepl("FEP", Identification) ~ "FEP",
grepl("TPS", Identification) ~ "TPS",
grepl("ASA", Identification) ~ "Acrylate",
grepl("PPS", Identification) ~ "PPS",
grepl("COC", Identification) ~ "COC",

```

```

grepl("PTT", Identification) ~ "PTT",
grepl("TPO", Identification) ~ "TPO",
grepl("PPP", Identification) ~ "PPP",
grepl("PPE", Identification) ~ "PPE",
grepl("SMC", Identification) ~ "SMC",
grepl("PPA", Identification) ~ "PPA",
grepl("EBA", Identification) ~ "Acrylate",
grepl("TPV", Identification) ~ "TPV",
grepl("PEI", Identification) ~ "PEI",
grepl("TPE", Identification) ~ "TPE",
grepl("SAN", Identification) ~ "SAN",
grepl("ASA", Identification) ~ "Acrylate",
grepl("PA", Identification) ~ "PA",
grepl("PP", Identification) ~ "PP",
grepl("pent", Identification, ignore.case=TRUE) ~ "Not Plastic",
grepl("K70 PE", Identification) ~ "Not Plastic",
grepl("PE", Identification) ~ "PE",
grepl("PU", Identification) ~ "PU",
grepl("PS", Identification) ~ "PS",
grepl("PC", Identification) ~ "PC",
grepl("SR", Identification) ~ "Silicone",
grepl("VE", Identification) ~ "Resin",
grepl("UP", Identification) ~ "Resin",
grepl("EP", Identification) ~ "Resin",
grepl("PF", Identification) ~ "Resin",
grepl("UF", Identification) ~ "Resin",
grepl("CM", Identification) ~ "CM",
grepl("TP", Identification) ~ "TP",
grepl("PO ", Identification) ~ "PO",
grepl("CR ", Identification) ~ "Rubber",
grepl("SB", Identification) ~ "SB",
grepl("cane", Identification, ignore.case=TRUE) ~ "Not Plastic",
grepl("CA", Identification) ~ "Cellulose acetate",
grepl("NOT IDENTIFIED", Identification, ignore.case=TRUE) ~ "Not
Identified",
.default = "Not Plastic"))

```

```
## To create a variable for whether this is a mixed plastic
```

```
# Set the data frame
```

```
df <- as_tibble(MPs_mod)
```

```
# Group into mixed polymer categories
```

```
MPs_mod <- df %>%
```

```

mutate(Mixed_Plastic = case_when(
  grepl("LCP/PTFE", Identification) ~ "Mixed Plastic",
  grepl("LCP/PPS", Identification) ~ "Mixed Plastic",
  grepl("PA46/PTFE", Identification) ~ "Mixed Plastic",
  grepl("PA6/ABS", Identification) ~ "Mixed Plastic",
  grepl("PA6/PET", Identification) ~ "Mixed Plastic",
  grepl("PA6/PP", Identification) ~ "Mixed Plastic",
  grepl("PA66/PE", Identification) ~ "Mixed Plastic",
  grepl("PA66/PTFE", Identification) ~ "Mixed Plastic",
  grepl("PBT/ASA", Identification) ~ "Mixed Plastic",
  grepl("PBT/PET", Identification) ~ "Mixed Plastic",

```

```

grepl("PBT/PTFE", Identification) ~ "Mixed Plastic",
grepl("PC/ABS", Identification) ~ "Mixed Plastic",
grepl("PC/ASA", Identification) ~ "Mixed Plastic",
grepl("PC/PBT", Identification) ~ "Mixed Plastic",
grepl("PC/PET", Identification) ~ "Mixed Plastic",
grepl("PC/SAN", Identification) ~ "Mixed Plastic",
grepl("PC/Siloxan", Identification) ~ "Mixed Plastic",
grepl("PC/PTFE", Identification) ~ "Mixed Plastic",
grepl("PE/EVA", Identification) ~ "Mixed Plastic",
grepl("PEEK/PTFE", Identification) ~ "Mixed Plastic",
grepl("PEI/PTFE", Identification) ~ "Mixed Plastic",
grepl("PESU/PTFE", Identification) ~ "Mixed Plastic",
grepl("PLA/PC", Identification) ~ "Mixed Plastic",
grepl("POM/PDMS", Identification) ~ "Mixed Plastic",
grepl("POM/PTFE", Identification) ~ "Mixed Plastic",
grepl("PP/EPDM", Identification) ~ "Mixed Plastic",
grepl("PP/EVA", Identification) ~ "Mixed Plastic",
grepl("PP/TPE", Identification) ~ "Mixed Plastic",
grepl("PP/TPO", Identification) ~ "Mixed Plastic",
grepl("PPA/PTFE", Identification) ~ "Mixed Plastic",
grepl("PPE/PA/PS", Identification) ~ "Mixed Plastic",
grepl("PPE/PS", Identification) ~ "Mixed Plastic",
grepl("Polycarbonate/Acrylonitrile Butadiene Styrene Blend",
Identification, ignore.case=TRUE) ~ "Mixed Plastic",
grepl("Polyvinyl Chloride-Phenolic-Nitrile Butadiene Rubber",
Identification, ignore.case=TRUE) ~ "Mixed Plastic",
grepl("PVC-P-NBR", Identification) ~ "Mixed Plastic",
grepl("Blend of Polypropylen and Ethylen / Propylen",
Identification, ignore.case=TRUE) ~ "Mixed Plastic",
grepl("Polyphenylene Ether + High Impact Polystyrene",
Identification, ignore.case=TRUE) ~ "Mixed Plastic",
grepl("PPE + HIPS", Identification) ~ "Mixed Plastic",
grepl("Blend of Polyphenylene Ether And Polyamide", Identification,
ignore.case=TRUE) ~ "Mixed Plastic",
grepl("PP + PA", Identification) ~ "Mixed Plastic",
grepl("Polycarbonate / Polybutylene Terephthalate", Identification,
ignore.case=TRUE) ~ "Mixed Plastic",
grepl("PC + PBT", Identification) ~ "Mixed Plastic",
.default = "No"))

```

```
## To assign confidence levels based on HQI
```

```
# Set the data frame
```

```
df <- as_tibble(MPs_mod)
```

```
# Copy and paste the following code to group into confidence categories
based on HQI
```

```
MPs_mod <- df %>%
```

```
  mutate(Confidence = case_when(
    MPs_mod$HQI >=700 ~ "High",
    MPs_mod$HQI <700 & MPs_mod$HQI >=400 ~ "Medium",
    MPs_mod$HQI <400 & MPs_mod$HQI >=200 ~ "Low",
    MPs_mod$HQI <200 ~ "Not Identified"))
```

```
## To assign MPs as Fluoropolymers compounds
```

```

# Set the data frame
df <- as_tibble(MPs_mod)

# Copy and paste the following code to group into polymer categories
MPs_mod_Fluoropolymer <- df %>%
  mutate(Fluoropolymers_Compounds = case_when(
    grepl("LCP/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PA46/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PA66/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PBT/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PC/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PEEK/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PEI/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PESU/PTFE", Identification) ~ "Fluoropolymer",
    grepl("POM/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PPA/PTFE", Identification) ~ "Fluoropolymer",
    grepl("PCTFE", Identification) ~ "Fluoropolymer",
    grepl("PTFE", Identification) ~ "Fluoropolymer",
    grepl("PVDF", Identification) ~ "Fluoropolymer",
    grepl("EFEP", Identification) ~ "Fluoropolymer",
    grepl("ETFE", Identification) ~ "Fluoropolymer",
    grepl("FKM", Identification) ~ "Fluoropolymer",
    grepl("FVMQ", Identification) ~ "Fluoropolymer",
    grepl("TFM", Identification) ~ "Fluoropolymer",
    grepl("PFA", Identification) ~ "Fluoropolymer",
    grepl("MFA", Identification) ~ "Fluoropolymer",
    grepl("FEP", Identification) ~ "Fluoropolymer",
    .default = "No"))

# Save the file
write.csv(MPs_mod_Fluoropolymer, file = "*FILE NAME*_Processed.csv",
row.names = FALSE)

## To get plastic polymer counts based on HQI
# Filter out any non-plastic and unidentified compounds
df <- subset(MPs_mod_Fluoropolymer, MPs_mod_Fluoropolymer$Confidence ==
"Low" | MPs_mod_Fluoropolymer$Confidence == "Medium" |
MPs_mod_Fluoropolymer$Confidence == "High")
MPs_mod_plastics <- df %>%
  filter(!((Polymer == 'Not Identified') | (Polymer == 'Not Plastic'))))
df <-MPs_mod_plastics

# Save file
write.csv(df, file = "*FILE NAME*_LMH_Processed.csv", row.names = FALSE)

# Create a function that will count objects in a group
my_func <- function(df,group){
  df %>%
    group_by(!as.name(group)) %>%
    summarise(my_count = n()) %>%
    arrange(desc(my_count))}

# Count the number of MPs in each group for HQI confidence levels of Low,
Medium, and High

```

```

my_func(df,"Polymer")

# Save the file
write.csv(my_func(df,"Polymer"), file = "*FILE NAME*_HQI_LMH_Count.csv",
row.names = FALSE)

## Count the number of MPs in each group for HQI confidence levels of
only Medium and High
df <- subset(MPs_mod_plastics, MPs_mod_plastics$Confidence == "Medium" |
MPs_mod_plastics$Confidence == "High")
# Save file
write.csv(df, file = "*FILE NAME*_MH_Processed.csv", row.names = FALSE)

my_func(df,"Polymer")
write.csv(my_func(df,"Polymer"), file = "*FILE NAME*_HQI_MH_Count.csv",
row.names = FALSE)

## To count Fluoropolymer compounds
# Subset only Fluoropolymer compounds
df <- subset(MPs_mod_Fluoropolymer,
MPs_mod_Fluoropolymer$Fluoropolymers_Compounds == 'Fluoropolymer')
MPs_mod_Fluoropolymer_only <- df

# Count number of Fluoropolymer compounds for low, medium, and high HQI
df <- subset(MPs_mod_Fluoropolymer_only,
MPs_mod_Fluoropolymer_only$Confidence == "Low" |
MPs_mod_Fluoropolymer_only$Confidence == "Medium" |
MPs_mod_Fluoropolymer_only$Confidence == "High")
my_func(df,"Identification")

# Save the file
write.csv(my_func(df,"Identification"), file = "*FILE
NAME*_Fluoropolymer_HQI_LMH_Count.csv", row.names = FALSE)

# Count number of Fluoropolymer compounds for only medium and high HQI
df <- subset(MPs_mod_Fluoropolymer_only,
MPs_mod_Fluoropolymer_only$Confidence == "Medium" |
MPs_mod_Fluoropolymer_only$Confidence == "High")
my_func(df,"Identification")

# Save the file
write.csv(my_func(df,"Identification"), file = "*FILE
NAME*_Fluoropolymer_HQI_MH_Count.csv", row.names = FALSE)

```