MP Oyster Depuration Project

Garth C

January 27, 2022

## Load libraries

Load the libraries we’ll need. You’ll have to install these if you don”t already have them.

library(plyr)  
library(reshape2)  
library(ggplot2)  
library(glmmTMB)  
library(MuMIn)  
library(lme4)

## Loading required package: Matrix

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

theme1 <-  
 theme\_bw() +  
 theme(  
 text = element\_text(size = 7),  
 axis.text = element\_text(size = 7),  
 strip.background = element\_blank(),  
 strip.text = element\_text(size = 7),  
 legend.text = element\_text(size = 7),  
 panel.grid = element\_blank()  
 )

## Oyster data

Import the oyster depuration data and view it.

oysterdep <-   
 read.csv("oysterdep.csv")  
  
head(oysterdep)

## id sampleday size.cat table tank species sample.date width  
## 1 MIS1 0 twentyto50 Start Start C. gigas 11-03-2016 93  
## 2 MIS1 0 fiftyto100 Start Start C. gigas 11-03-2016 93  
## 3 MIS1 0 onehundredto500 Start Start C. gigas 11-03-2016 93  
## 4 MIS1 0 fivehundredto1000 Start Start C. gigas 11-03-2016 93  
## 5 MIS1 0 onethousandto5000 Start Start C. gigas 11-03-2016 93  
## 6 MIS2 0 twentyto50 Start Start C. gigas 11-03-2016 66  
## length depth cont.weight cont.dryweight dry.weight shell.dry.weight CI  
## 1 43 21 259.28 260.96 1.68 17.65 0.095  
## 2 43 21 259.28 260.96 1.68 17.65 0.095  
## 3 43 21 259.28 260.96 1.68 17.65 0.095  
## 4 43 21 259.28 260.96 1.68 17.65 0.095  
## 5 43 21 259.28 260.96 1.68 17.65 0.095  
## 6 40 21 262.04 263.63 1.59 12.62 0.126  
## time.in.oven time.out.oven dat.filter dat.count observer red.fib  
## 1 2016-11-05/ 17:03 2016-11-10/18:00 11-17-2016 11-29-2016 MD 0  
## 2 2016-11-05/ 17:03 2016-11-10/18:00 11-17-2016 11-29-2016 MD 0  
## 3 2016-11-05/ 17:03 2016-11-10/18:00 11-17-2016 11-29-2016 MD 0  
## 4 2016-11-05/ 17:03 2016-11-10/18:00 11-17-2016 11-29-2016 MD 0  
## 5 2016-11-05/ 17:03 2016-11-10/18:00 11-17-2016 11-29-2016 MD 0  
## 6 2016-11-05/ 17:03 2016-11-10/18:00 11-17-2016 11-28-2016 GC 0  
## yell.fib green.fib turq.fib blu.fib purp.fib gray.fib clear.fib pink.fib  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 1 0 0 0 0  
## 4 0 0 0 1 0 0 0 0  
## 5 0 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0 0  
## brown.fib orang.fib black.fib red.frag yell.frag green.frag turq.frag  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 1 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## blu.frag purp.frag gray.frag clear.frag pink.frag brown.frag orang.frag  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## black.frag whit.spher clear.spher  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## 6 0 0 0

Get all of the variables into the right format.

oysterdep$id <- as.factor(oysterdep$id)  
oysterdep$sampleday <- as.factor(oysterdep$sampleday)  
oysterdep$size.cat <- as.factor(oysterdep$size.cat)  
oysterdep$table <- as.factor(oysterdep$table)  
oysterdep$tank <- as.factor(oysterdep$tank)

Make a new dataframe to work with and rename size categories.

oysterdep2 <- oysterdep  
  
oysterdep2$size.cat <- factor(oysterdep2$size.cat,   
 levels=c("twentyto50", "fiftyto100",  
 "onehundredto500", "fivehundredto1000",   
 "onethousandto5000")) # reorder  
  
  
oysterdep2$size.cat <- mapvalues(oysterdep2$size.cat,   
 from = c("twentyto50", "fiftyto100",   
 "onehundredto500",   
 "fivehundredto1000",   
 "onethousandto5000"),   
 to = c("20-50", "50-100", "100-500",  
 "500-1000", "1000-5000")) # rename  
summary(oysterdep2$size.cat) # view

## 20-50 50-100 100-500 500-1000 1000-5000   
## 49 50 50 50 50

We need to melt the dataframe so that the particle types are organized by row instead of by column.

oysterdep3 <- melt(oysterdep2,  
 id.vars = c("id", "sampleday", "size.cat", "table",  
 "tank", "species", "sample.date", "length",   
 "width", "depth", "cont.weight",   
 "cont.dryweight", "dry.weight",   
 "shell.dry.weight", "CI", "time.in.oven",   
 "time.out.oven", "dat.filter", "dat.count",   
 "observer"))  
  
## Take out unnecessary columns  
  
oysterdep3 <- oysterdep3[,c(1:5,7:10,13:15,18:22)]  
  
head(oysterdep3) # now "variable" is particle category and "value is the count"

## id sampleday size.cat table tank sample.date length width depth  
## 1 MIS1 0 20-50 Start Start 11-03-2016 43 93 21  
## 2 MIS1 0 50-100 Start Start 11-03-2016 43 93 21  
## 3 MIS1 0 100-500 Start Start 11-03-2016 43 93 21  
## 4 MIS1 0 500-1000 Start Start 11-03-2016 43 93 21  
## 5 MIS1 0 1000-5000 Start Start 11-03-2016 43 93 21  
## 6 MIS2 0 20-50 Start Start 11-03-2016 40 66 21  
## dry.weight shell.dry.weight CI dat.filter dat.count observer variable  
## 1 1.68 17.65 0.095 11-17-2016 11-29-2016 MD red.fib  
## 2 1.68 17.65 0.095 11-17-2016 11-29-2016 MD red.fib  
## 3 1.68 17.65 0.095 11-17-2016 11-29-2016 MD red.fib  
## 4 1.68 17.65 0.095 11-17-2016 11-29-2016 MD red.fib  
## 5 1.68 17.65 0.095 11-17-2016 11-29-2016 MD red.fib  
## 6 1.59 12.62 0.126 11-17-2016 11-28-2016 GC red.fib  
## value  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 0

Get rid of the <100 micron particles as we”re not confident in these.

oysterdep3 <- subset(oysterdep3, size.cat != "20-50" & size.cat != "50-100")  
  
oysterdep3$size.cat <- as.character(oysterdep3$size.cat)  
oysterdep3$size.cat <- as.factor(oysterdep3$size.cat)  
  
cat.sums <- aggregate(oysterdep3$value ~   
 oysterdep3$variable, FUN = sum) # sums everything  
print(cat.sums)

## oysterdep3$variable oysterdep3$value  
## 1 red.fib 1  
## 2 yell.fib 3  
## 3 green.fib 4  
## 4 turq.fib 1  
## 5 blu.fib 52  
## 6 purp.fib 0  
## 7 gray.fib 0  
## 8 clear.fib 326  
## 9 pink.fib 2  
## 10 brown.fib 3  
## 11 orang.fib 3  
## 12 black.fib 156  
## 13 red.frag 0  
## 14 yell.frag 0  
## 15 green.frag 0  
## 16 turq.frag 0  
## 17 blu.frag 0  
## 18 purp.frag 0  
## 19 gray.frag 0  
## 20 clear.frag 2  
## 21 pink.frag 0  
## 22 brown.frag 0  
## 23 orang.frag 0  
## 24 black.frag 0  
## 25 whit.spher 5  
## 26 clear.spher 0

So we can remove the empty categories.

oysterdep4 <- subset(oysterdep3,   
 variable == "red.fib" |  
 variable == "yell.fib" |  
 variable == "green.fib" |  
 variable == "turq.fib" |  
 variable == "blu.fib" |  
 variable == "clear.fib" |  
 variable == "pink.fib" |  
 variable == "brown.fib" |  
 variable == "orang.fib" |  
 variable == "black.fib" |  
 variable == "clear.frag" |  
 variable == "whit.spher") # keeps all these rows  
  
oysterdep4$variable <- as.character(oysterdep4$variable)  
oysterdep4$variable <- as.factor(oysterdep4$variable) # gets rid of unused names  
  
summary(oysterdep4$variable) # view

## black.fib blu.fib brown.fib clear.fib clear.frag green.fib orang.fib   
## 150 150 150 150 150 150 150   
## pink.fib red.fib turq.fib whit.spher yell.fib   
## 150 150 150 150 150

Let”s make the category names a bit more useful for plotting

oysterdep4$variable <- mapvalues(oysterdep4$variable,  
 from = c(levels(oysterdep4$variable)),   
 to = c("Black fibres",   
 "Blue fibres",  
 "Brown fibres",  
 "Clear fibres",  
 "Clear fragments",  
 "Green fibres",  
 "Orange fibres",  
 "Pink fibres",  
 "Red fibres",  
 "Turquoise fibres",   
 "White spheres",  
 "Yellow fibres"))  
  
summary(oysterdep4$variable)

## Black fibres Blue fibres Brown fibres Clear fibres   
## 150 150 150 150   
## Clear fragments Green fibres Orange fibres Pink fibres   
## 150 150 150 150   
## Red fibres Turquoise fibres White spheres Yellow fibres   
## 150 150 150 150

## Oyster blanks

Now load in the blanks dataset and set it up for plotting, same as with the oyster data set. First we load in the data.

depblanks <-   
 read.csv("depblanks.csv")  
  
head(depblanks)

## id sampleday size.cat blankrun time.in.oven  
## 1 MControl1 (0,1) 0 twentyto50 1 2016-11-05/ 17:03  
## 2 MControl1 (0,1) 0 fiftyto100 1 2016-11-05/ 17:03  
## 3 MControl1 (0,1) 0 onehundredto500 1 2016-11-05/ 17:03  
## 4 MControl1 (0,1) 0 fivehundredto1000 1 2016-11-05/ 17:03  
## 5 MControl1 (0,1) 0 onethousandto5000 1 2016-11-05/ 17:03  
## 6 MControl2 (0,1) 0 twentyto50 1 2016-11-05/ 17:03  
## time.out.oven dat.filter dat.count observer red.fib yell.fib green.fib  
## 1 2016-11-10/18:00 11-17-2016 11-24-2016 GC 0 0 0  
## 2 2016-11-10/18:00 11-17-2016 11-24-2016 GC 0 0 0  
## 3 2016-11-10/18:00 11-17-2016 11-24-2016 GC 0 0 0  
## 4 2016-11-10/18:00 11-17-2016 11-24-2016 GC 0 0 0  
## 5 2016-11-10/18:00 11-17-2016 11-24-2016 GC 0 0 0  
## 6 2016-11-10/18:00 11-17-2016 11-24-2016 GC 0 0 0  
## turq.fib blu.fib purp.fib gray.fib clear.fib pink.fib brown.fib orang.fib  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 0 1 0 0 0  
## 4 0 0 0 0 1 0 0 0  
## 5 0 0 0 0 3 0 0 0  
## 6 0 0 0 0 0 0 0 0  
## black.fib red.frag yell.frag green.frag turq.frag blu.frag purp.frag  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## gray.frag clear.frag pink.frag brown.frag orang.frag black.frag whit.spher  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## clear.spher  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 0

depblanks$id <- as.factor(depblanks$id)  
depblanks$size.cat <- as.factor(depblanks$size.cat)  
depblanks$observer <- as.factor(depblanks$observer)  
depblanks$sampleday <- as.factor(depblanks$sampleday)

Now we make a new dataframe to work with and rename size categories.

depblanks2 <- depblanks  
  
depblanks2$size.cat <- factor(depblanks2$size.cat,   
 levels=c("twentyto50", "fiftyto100",  
 "onehundredto500", "fivehundredto1000",   
 "onethousandto5000")) # reorder  
  
  
depblanks2$size.cat <- mapvalues(depblanks2$size.cat,   
 from = c("twentyto50", "fiftyto100",   
 "onehundredto500",   
 "fivehundredto1000",   
 "onethousandto5000"),   
 to = c("20-50", "50-100", "100-500",  
 "500-1000", "1000-5000")) # rename  
summary(depblanks2$size.cat) # view

## 20-50 50-100 100-500 500-1000 1000-5000   
## 15 15 15 15 15

Again, we melt the dataframe so that the particle types are organized by row instead of by column.

depblanks3 <- melt(depblanks2,  
 id.vars = c("id", "size.cat", "sampleday", "blankrun",   
 "time.in.oven", "time.out.oven", "dat.filter",  
 "dat.count", "observer"))  
  
## Take out unnecessary columns  
  
depblanks3 <- depblanks3[c("id", "size.cat", "sampleday", "blankrun",   
 "dat.filter", "dat.count", "observer", "variable",  
 "value")]  
  
head(depblanks3) # now "variable" is particle category and "value is the count"

## id size.cat sampleday blankrun dat.filter dat.count observer  
## 1 MControl1 (0,1) 20-50 0 1 11-17-2016 11-24-2016 GC  
## 2 MControl1 (0,1) 50-100 0 1 11-17-2016 11-24-2016 GC  
## 3 MControl1 (0,1) 100-500 0 1 11-17-2016 11-24-2016 GC  
## 4 MControl1 (0,1) 500-1000 0 1 11-17-2016 11-24-2016 GC  
## 5 MControl1 (0,1) 1000-5000 0 1 11-17-2016 11-24-2016 GC  
## 6 MControl2 (0,1) 20-50 0 1 11-17-2016 11-24-2016 GC  
## variable value  
## 1 red.fib 0  
## 2 red.fib 0  
## 3 red.fib 0  
## 4 red.fib 0  
## 5 red.fib 0  
## 6 red.fib 0

Now sum by totals across all samples and particles sizes to get an idea of what we”re working with. First remove particles less than 100 microns as we”re not confident in them.

depblanks3 <- subset(depblanks3, size.cat != "20-50" & size.cat != "50-100")  
  
depblanks3$size.cat <- as.character(depblanks3$size.cat)  
depblanks3$size.cat <- as.factor(depblanks3$size.cat)  
  
cat.sums.blanks <- aggregate(depblanks3$value ~   
 depblanks3$variable, FUN = sum) # sums everything  
print(cat.sums.blanks)

## depblanks3$variable depblanks3$value  
## 1 red.fib 0  
## 2 yell.fib 0  
## 3 green.fib 0  
## 4 turq.fib 0  
## 5 blu.fib 4  
## 6 purp.fib 0  
## 7 gray.fib 0  
## 8 clear.fib 42  
## 9 pink.fib 2  
## 10 brown.fib 0  
## 11 orang.fib 0  
## 12 black.fib 0  
## 13 red.frag 0  
## 14 yell.frag 0  
## 15 green.frag 0  
## 16 turq.frag 0  
## 17 blu.frag 0  
## 18 purp.frag 0  
## 19 gray.frag 0  
## 20 clear.frag 0  
## 21 pink.frag 0  
## 22 brown.frag 0  
## 23 orang.frag 0  
## 24 black.frag 0  
## 25 whit.spher 0  
## 26 clear.spher 0

So we can remove everything but blue, clear, and pink fibres.

depblanks4 <- subset(depblanks3,   
 c(variable == "blu.fib" |  
 variable == "clear.fib" |  
 variable == "pink.fib")  
 ) # keep only these rows  
  
depblanks4$variable <- as.character(depblanks4$variable)  
depblanks4$variable <- as.factor(depblanks4$variable) # gets rid of those names  
  
summary(depblanks4$variable) # view

## blu.fib clear.fib pink.fib   
## 45 45 45

Let”s make the category names a bit more useful for plotting

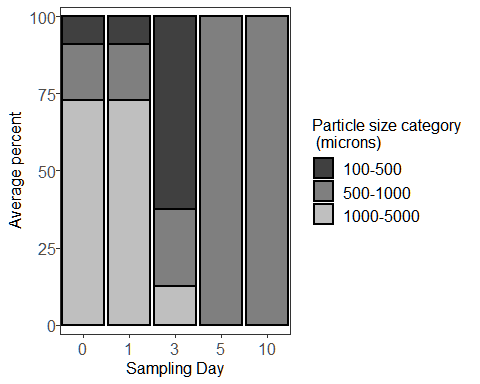
depblanks4$variable <- mapvalues(depblanks4$variable,  
 from = c(levels(depblanks4$variable)),   
 to = c("Blue fibres",  
 "Clear fibres",  
 "Pink fibres"))  
  
summary(depblanks4$variable)

## Blue fibres Clear fibres Pink fibres   
## 45 45 45

Next let”s make plots of the distribution of sizes and particle types and colours.

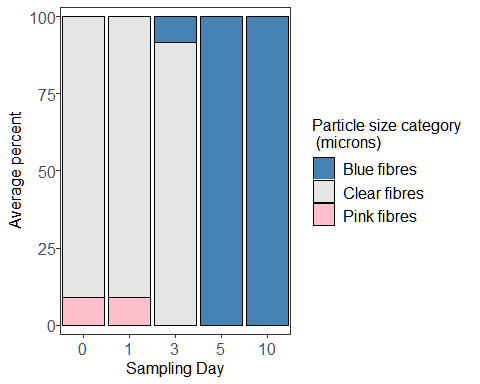
Starting with size

## First we need to make a new data frame of sums according to each size category  
  
blankssizesums <- ddply(depblanks4, c("id", "size.cat", "sampleday",   
 "blankrun", "dat.filter", "dat.count",   
 "observer"), summarize, sum = sum(value))  
  
  
  
## Now calculate proportions of each particle size category present for each  
## sampling day  
  
blanksprop <- ddply(blankssizesums,  
 c("sampleday", "size.cat"),   
 summarize,   
 sum = sum(sum))  
blanksprop$prop <- numeric(length = nrow(blanksprop))  
  
for(i in 1:nrow(blanksprop)) {  
 blanksprop$prop[i] <-  
 (blanksprop$sum[i] /  
 sum(blanksprop$sum[blanksprop$sampleday == blanksprop$sampleday[i]]))  
}  
  
blanksprop$size.cat <- as.character(blanksprop$size.cat)  
blanksprop$size.cat <- as.factor(blanksprop$size.cat)  
  
## Put the size categories in order  
  
blanksprop$size.cat <- factor(blanksprop$size.cat,  
 levels = c("10-100",  
 "100-500", "500-1000",  
 "1000-5000"))  
  
## Now plot  
  
ggplot(blanksprop, aes(x=sampleday, y=100\*prop, fill = size.cat)) +   
 geom\_col(size = 1, colour = "black") +   
 xlab("Sampling Day") +   
 ylab("Average percent") +  
 guides(fill=guide\_legend(title="Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(values = c("grey25", "grey50", "grey75")) +  
 theme(legend.text = element\_text(size=12), text = element\_text(size=12),   
 panel.spacing = unit(1, "lines"),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12,   
 margin = margin(t=0, r=0, b=0, l=2)),  
 strip.background = element\_blank(),   
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside") +  
 scale\_x\_discrete(expand = c(0,0.5)) +   
 scale\_y\_continuous(expand = c(0.03,0))



Now with particle type/colour

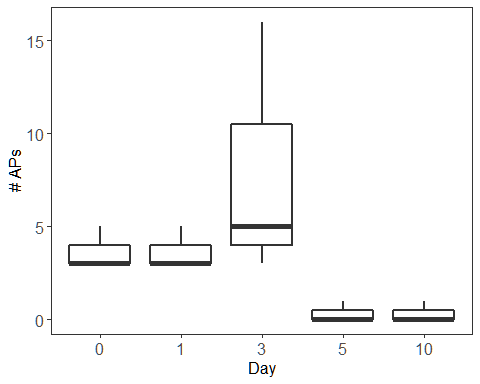
## First we need to make a new data frame of sums according to each particle  
## type/colour  
  
blankstypesums <- ddply(depblanks4, c("id", "sampleday", "blankrun",   
 "dat.filter", "dat.count", "observer",   
 "variable"),   
 summarize, sum = sum(value))  
  
## Now calculate proportions of each particle type/colour present for each   
## sampling day  
  
blankstypeprop <- ddply(blankstypesums,  
 c("sampleday", "variable"),   
 summarize,   
 sum = sum(sum))  
blankstypeprop$prop <- numeric(length = nrow(blankstypeprop))  
  
for(i in 1:nrow(blankstypeprop)) {  
 blankstypeprop$prop[i] <-  
 (blankstypeprop$sum[i] /  
 sum(blankstypeprop$sum[blankstypeprop$sampleday == blankstypeprop$sampleday[i]]))  
}  
  
blankstypeprop$variable <- as.character(blankstypeprop$variable)  
blankstypeprop$variable <- as.factor(blankstypeprop$variable)  
  
## Now plot  
  
ggplot(blankstypeprop, aes(x=sampleday, y=100\*prop, fill = variable)) +   
 geom\_col(size = 0.5, colour = "black") +   
 xlab("Sampling Day") +   
 ylab("Average percent") +  
 guides(fill=guide\_legend(title="Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(values = c("Steel Blue",  
 "Grey90",  
 "Pink")) +  
 theme(legend.text = element\_text(size=12), text = element\_text(size=12),   
 panel.spacing = unit(1, "lines"),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12,   
 margin = margin(t=0, r=0, b=0, l=2)),  
 strip.background = element\_blank(),   
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside") +  
 scale\_x\_discrete(expand = c(0,0.5)) +   
 scale\_y\_continuous(expand = c(0.03,0))



Take total sums and plot the results.

totalblanksums <- ddply(depblanks4, c("id", "sampleday", "observer",   
 "dat.filter", "blankrun"),  
 summarise, sum = sum(value)) # adds all the counts together  
  
## Plot according to count per filtration day  
  
ggplot(totalblanksums, aes(x=sampleday, y=sum)) +   
 geom\_boxplot(position=position\_dodge(), size=1) +   
 xlab("Day") +   
 ylab("# APs") +  
 guides(fill=FALSE) +  
 theme\_bw() +  
 theme(text = element\_text(size=12),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12),  
 strip.background = element\_blank(),  
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.



Now do summary stats for the blanks

## medians, range, and SD  
  
with(totalblanksums, tapply(sum, sampleday, median)) # takes median for each day

## 0 1 3 5 10   
## 3 3 5 0 0

with(totalblanksums, tapply(sum, sampleday, min)) # takes minimum for each day

## 0 1 3 5 10   
## 3 3 3 0 0

with(totalblanksums, tapply(sum, sampleday, max)) # takes maximum for each day

## 0 1 3 5 10   
## 5 5 16 1 1

with(totalblanksums, tapply(sum,   
 sampleday, sd)) # takes standard deviation for each day

## 0 1 3 5 10   
## 1.1547005 1.1547005 7.0000000 0.5773503 0.5773503

## Proportions of particles sizes  
  
sum(depblanks4$value[depblanks4$size.cat == "100-500"])/  
 sum(depblanks4$value)\*100

## [1] 35.41667

sum(depblanks4$value[depblanks4$size.cat == "500-1000"])/  
 sum(depblanks4$value)\*100

## [1] 25

sum(depblanks4$value[depblanks4$size.cat == "1000-5000"])/  
 sum(depblanks4$value)\*100

## [1] 39.58333

## This provides the percentage of total particles found accounted for by each   
## colour/type of particles  
blanksummarytable <- with(depblanks4, tapply(value, variable, sum))  
blanksummarytable <- blanksummarytable/sum(depblanks4$value)\*100  
print(blanksummarytable)

## Blue fibres Clear fibres Pink fibres   
## 8.333333 87.500000 4.166667

## Ajusted oyster data

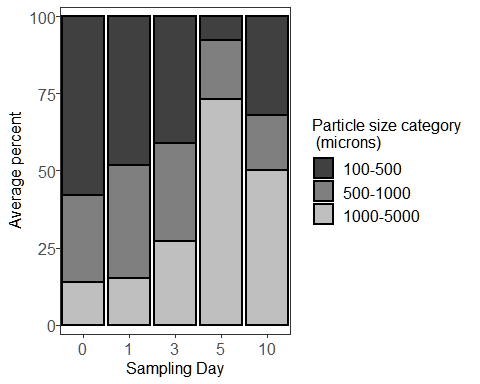
Now subtract the blank counts from the appropriate size and shape categories for the oyster counts.

depblanksmean <- ddply(depblanks4, c("size.cat", "sampleday", "dat.filter",   
 "variable"),  
 summarize, value.b = mean(value))  
  
depblanksmean$value.b <-  
 ceiling(depblanksmean$value.b) ## Round all blank averages up to the nearest whole number  
  
oysterdep5 <-  
 left\_join(oysterdep4, depblanksmean,   
 by = c("sampleday", "size.cat", "variable")) ## Join the 2 dataframes  
  
oysterdep5$value.b <- as.numeric(oysterdep5$value.b)  
oysterdep5$value.b[is.na(oysterdep5$value.b)] <- 0  
  
oysterdep4$variable <- as.factor(oysterdep4$variable)  
  
oysterdep5$finalcount <- (oysterdep5$value - oysterdep5$value.b)  
  
oysterdep5$finalcount[oysterdep5$finalcount < 0] <-  
 0 ## Change any negative values to 0  
  
summary(oysterdep5$finalcount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.2556 0.0000 24.0000

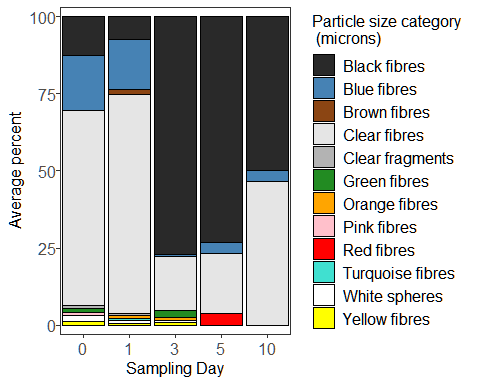
Now run all of the same code again but for the modified oyster counts.

## First we need to make a new data frame of sums according to each size category  
  
oystersizesums <- ddply(oysterdep5, c("id", "sampleday", "size.cat", "table",   
 "tank", "sample.date", "length", "width",  
 "depth", "dry.weight",   
 "shell.dry.weight", "CI", "dat.count",   
 "observer"),   
 summarize, sum = sum(finalcount))  
  
  
  
## Now calculate proportions of each particle size category present for each   
## sampling day  
  
oysterprop <- ddply(oystersizesums,  
 c("sampleday", "size.cat"),   
 summarize,   
 sum = sum(sum))  
oysterprop$prop <- numeric(length = nrow(oysterprop))  
  
for(i in 1:nrow(oysterprop)) {  
 oysterprop$prop[i] <-  
 (oysterprop$sum[i] /  
 sum(oysterprop$sum[oysterprop$sampleday == oysterprop$sampleday[i]]))  
}  
  
oysterprop$size.cat <- as.character(oysterprop$size.cat)  
oysterprop$size.cat <- as.factor(oysterprop$size.cat)  
  
oysterprop$size.cat <-   
 factor(oysterprop$size.cat,   
 levels = c("100-500", "500-1000", "1000-5000"))  
  
## Now plot  
  
ggplot(oysterprop, aes(x=sampleday, y=100\*prop, fill = size.cat)) +   
 geom\_col(size = 1, colour = "black") +   
 xlab("Sampling Day") +   
 ylab("Average percent") +  
 guides(fill=guide\_legend(title="Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(values = c("grey25", "grey50", "grey75")) +  
 theme(legend.text = element\_text(size=12), text = element\_text(size=12),   
 panel.spacing = unit(1, "lines"),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12,   
 margin = margin(t=0, r=0, b=0, l=2)),  
 strip.background = element\_blank(),   
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside") +  
 scale\_x\_discrete(expand = c(0,0.5)) +   
 scale\_y\_continuous(expand = c(0.03,0))



Now with particle type/colour

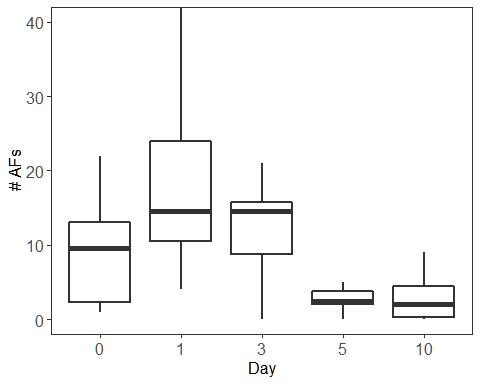
## First we need to make a new data frame of sums according to each particle  
## type/colour  
  
oystertypesums <- ddply(oysterdep5,   
 c("id", "sampleday", "table", "tank", "sample.date",   
 "length", "width", "depth", "dry.weight",  
 "shell.dry.weight", "CI", "dat.count", "observer",  
 "variable"),  
 summarize,   
 sum = sum(finalcount))  
  
  
  
## Now calculate proportions of each particle type/colour present for each  
## sampling day  
  
oystertypeprop <- ddply(oystertypesums,  
 c("sampleday", "variable"),   
 summarize,   
 sum = sum(sum))  
oystertypeprop$prop <- numeric(length = nrow(oystertypeprop))  
  
for(i in 1:nrow(oystertypeprop)) {  
 oystertypeprop$prop[i] <-  
 (oystertypeprop$sum[i] /  
 sum(oystertypeprop$sum[oystertypeprop$sampleday ==  
 oystertypeprop$sampleday[i]]))  
}  
  
oystertypeprop$variable <- as.character(oystertypeprop$variable)  
oystertypeprop$variable <- as.factor(oystertypeprop$variable)  
  
## Now plot  
  
  
ggplot(oystertypeprop, aes(x = sampleday, y = 100 \* prop, fill = variable)) +  
 geom\_col(size = 0.5, colour = "black") +  
 xlab("Sampling Day") +  
 ylab("Average percent") +  
 guides(fill = guide\_legend(title = "Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(  
 values = c(  
 "Grey16",  
 "Steel Blue",  
 "Saddle Brown",  
 "Grey90",  
 "Grey70",  
 "Forest Green",  
 "Orange",  
 "Pink",  
 "Red",  
 "Turquoise",  
 "White",  
 "Yellow"  
 )  
 ) +  
 theme(  
 legend.text = element\_text(size = 12),  
 text = element\_text(size = 12),  
 panel.spacing = unit(1, "lines"),  
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12,  
 margin = margin(  
 t = 0,  
 r = 0,  
 b = 0,  
 l = 2  
 )),  
 strip.background = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside"  
 ) +  
 scale\_x\_discrete(expand = c(0, 0.5)) +  
 scale\_y\_continuous(expand = c(0.03, 0))



Take total sums overall and plot the results.

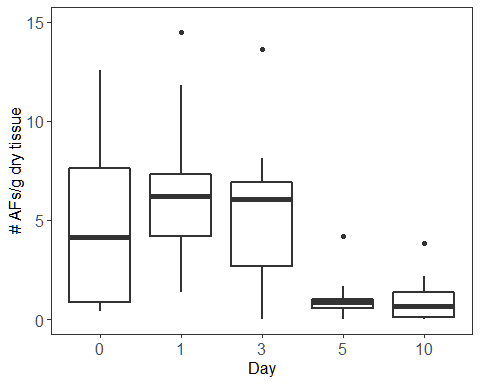
totalsums <- ddply(oysterdep5,   
 c("id", "sampleday", "table", "tank", "length", "width",   
 "depth", "dry.weight", "shell.dry.weight", "CI", "observer"),  
 summarise,   
 sum = sum(finalcount)) # adds all the counts together  
  
## Plot according to count per individual  
  
ggplot(totalsums, aes(x=sampleday , y=sum)) +   
 geom\_boxplot(position=position\_dodge(), size=1) +   
 xlab("Day") +   
 coord\_cartesian(ylim = c(0, 40)) +  
 ylab("# AFs") +  
 guides(fill=FALSE) +  
 theme\_bw() +  
 theme(text = element\_text(size=12),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12),  
 strip.background = element\_blank(),  
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.



## Plot according to concentration (particles/dry tissue weight)  
  
ggplot(totalsums, aes(x=sampleday , y=sum/dry.weight)) +   
 geom\_boxplot(position=position\_dodge(), size=1) +   
 xlab("Day") +   
 coord\_cartesian(ylim = c(0, 15)) +  
 ylab("# AFs/g dry tissue") +  
 guides(fill=FALSE) +  
 theme\_bw() +  
 theme(text = element\_text(size=12),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12),  
 strip.background = element\_blank(),  
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.



Let”s print out some summary stats.

## Summary stats according to count per individual  
  
with(totalsums, tapply(sum, sampleday, mean)) # takes mean for each day

## 0 1 3 5 10   
## 9.5 18.5 12.6 2.6 2.8

with(totalsums, tapply(sum, sampleday, min)) # takes minimum for each day

## 0 1 3 5 10   
## 1 4 0 0 0

with(totalsums, tapply(sum, sampleday, max)) # takes maximum for each day

## 0 1 3 5 10   
## 22 42 21 5 9

with(totalsums, tapply(sum,   
 sampleday, sd)) # takes standard deviation for each day

## 0 1 3 5 10   
## 7.947746 11.881358 6.801961 1.505545 3.047768

## Summary stats according to concentration  
  
with(totalsums, tapply(sum/dry.weight,   
 sampleday, mean)) # takes mean for each day

## 0 1 3 5 10   
## 4.781623 6.708133 5.484178 1.106082 1.000859

with(totalsums, tapply(sum/dry.weight,   
 sampleday, min)) # takes minimum for each day

## 0 1 3 5 10   
## 0.3891051 1.3513514 0.0000000 0.0000000 0.0000000

with(totalsums, tapply(sum/dry.weight,   
 sampleday, max)) # takes maximum for each day

## 0 1 3 5 10   
## 12.574850 14.482759 13.636364 4.201681 3.813559

with(totalsums, tapply(sum/dry.weight,   
 sampleday, sd)) # takes standard deviation for each day

## 0 1 3 5 10   
## 4.318497 3.882901 3.907880 1.175746 1.213063

## Proportions of particles sizes  
  
sum(oystersizesums$sum[oystersizesums$size.cat == "100-500"])/  
 sum(oystersizesums$sum)\*100

## [1] 45

sum(oystersizesums$sum[oystersizesums$size.cat == "500-1000"])/  
 sum(oystersizesums$sum)\*100

## [1] 31.52174

sum(oystersizesums$sum[oystersizesums$size.cat == "1000-5000"])/  
 sum(oystersizesums$sum)\*100

## [1] 23.47826

## This provides the percentage of total particles found accounted for by each   
## colour/type of particles  
oystersummarytable <- with(oystertypesums, tapply(sum, variable, sum))  
oystersummarytable <- oystersummarytable/sum(oystertypesums$sum)\*100  
print(oystersummarytable)

## Black fibres Blue fibres Brown fibres Clear fibres   
## 33.9130435 10.8695652 0.6521739 50.2173913   
## Clear fragments Green fibres Orange fibres Pink fibres   
## 0.4347826 0.8695652 0.6521739 0.2173913   
## Red fibres Turquoise fibres White spheres Yellow fibres   
## 0.2173913 0.2173913 1.0869565 0.6521739

## Water data

Now load in the water sample data.

depwater <-   
 read.csv("depwater.csv")  
  
head(depwater)

## id sampleday size.cat table tank time.in.oven  
## 1 Water Day 0 A 0 twentyto50 A 4 2017-01-09/17:00  
## 2 Water Day 0 A 0 fiftyto100 A 4 2017-01-09/17:00  
## 3 Water Day 0 A 0 onehundredto500 A 4 2017-01-09/17:00  
## 4 Water Day 0 A 0 fivehundredto1000 A 4 2017-01-09/17:00  
## 5 Water Day 0 A 0 onethousandto5000 A 4 2017-01-09/17:00  
## 6 Water Day 0 B 0 twentyto50 B 3 2017-01-09/17:00  
## time.out.oven dat.filter dat.count observer red.fib yell.fib green.fib  
## 1 2017-01-17/09:30 12-13-2017 12-20-2017 MD 0 0 0  
## 2 2017-01-17/09:30 12-13-2017 12-20-2017 MD 0 0 0  
## 3 2017-01-17/09:30 12-13-2017 12-20-2017 MD 0 0 0  
## 4 2017-01-17/09:30 12-13-2017 12-20-2017 MD 0 0 0  
## 5 2017-01-17/09:30 12-13-2017 12-20-2017 MD 0 0 0  
## 6 2017-01-17/09:30 12-13-2017 12-21-2017 MD 0 0 0  
## turq.fib blu.fib purp.fib gray.fib clear.fib pink.fib brown.fib orang.fib  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0 0  
## 5 0 0 0 0 1 0 0 0  
## 6 0 0 0 0 0 0 0 0  
## black.fib tot.fib red.frag yell.frag green.frag turq.frag blu.frag purp.frag  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0 0  
## 5 0 1 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0 0  
## gray.frag clear.frag pink.frag brown.frag orang.frag black.frag whit.spher  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## clear.spher  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 0

depwater$id <- as.factor(depwater$id)  
depwater$sampleday <- as.factor(depwater$sampleday)  
depwater$size.cat <- as.factor(depwater$size.cat)  
depwater$table <- as.factor(depwater$table)  
depwater$tank <- as.factor(depwater$tank)  
depwater$observer <- as.factor(depwater$observer)

Now we make a new dataframe to work with and rename size categories.

depwater2 <- depwater  
  
depwater2$size.cat <- factor(depwater2$size.cat,   
 levels=c("twentyto50", "fiftyto100",  
 "onehundredto500", "fivehundredto1000",   
 "onethousandto5000")) # reorder  
  
  
depwater2$size.cat <- mapvalues(depwater2$size.cat,   
 from = c("twentyto50", "fiftyto100",   
 "onehundredto500",   
 "fivehundredto1000",   
 "onethousandto5000"),   
 to = c("20-50", "50-100", "100-500",  
 "500-1000", "1000-5000")) # rename  
summary(depwater2$size.cat) # view

## 20-50 50-100 100-500 500-1000 1000-5000   
## 20 20 20 20 20

Again, we melt the dataframe so that the particle types are organized by row instead of by column.

depwater3 <- melt(depwater2,  
 id.vars = c("id", "sampleday", "size.cat", "table", "tank",   
 "time.in.oven", "time.out.oven", "dat.filter",  
 "dat.count", "observer"))  
  
## Take out unnecessary columns  
  
depwater3 <- depwater3[,-c(6:7)]  
  
head(depwater3) # now "variable" is particle category and "value is the count"

## id sampleday size.cat table tank dat.filter dat.count observer  
## 1 Water Day 0 A 0 20-50 A 4 12-13-2017 12-20-2017 MD  
## 2 Water Day 0 A 0 50-100 A 4 12-13-2017 12-20-2017 MD  
## 3 Water Day 0 A 0 100-500 A 4 12-13-2017 12-20-2017 MD  
## 4 Water Day 0 A 0 500-1000 A 4 12-13-2017 12-20-2017 MD  
## 5 Water Day 0 A 0 1000-5000 A 4 12-13-2017 12-20-2017 MD  
## 6 Water Day 0 B 0 20-50 B 3 12-13-2017 12-21-2017 MD  
## variable value  
## 1 red.fib 0  
## 2 red.fib 0  
## 3 red.fib 0  
## 4 red.fib 0  
## 5 red.fib 0  
## 6 red.fib 0

Now sum by totals across all samples and particles sizes to get an idea of what we”re working with. First remove particles <100 microns as we don”t trust these counts.

depwater3 <- subset(depwater3, size.cat != "20-50" & size.cat != "50-100")  
  
depwater3$size.cat <- as.character(depwater3$size.cat)  
depwater3$size.cat <- as.factor(depwater3$size.cat)  
  
cat.sums.blanks <- aggregate(depwater3$value ~   
 depwater3$variable, FUN = sum) # sums everything  
print(cat.sums.blanks)

## depwater3$variable depwater3$value  
## 1 red.fib 3  
## 2 yell.fib 0  
## 3 green.fib 0  
## 4 turq.fib 1  
## 5 blu.fib 4  
## 6 purp.fib 0  
## 7 gray.fib 0  
## 8 clear.fib 37  
## 9 pink.fib 0  
## 10 brown.fib 0  
## 11 orang.fib 0  
## 12 black.fib 0  
## 13 tot.fib 45  
## 14 red.frag 0  
## 15 yell.frag 0  
## 16 green.frag 0  
## 17 turq.frag 0  
## 18 blu.frag 0  
## 19 purp.frag 0  
## 20 gray.frag 0  
## 21 clear.frag 0  
## 22 pink.frag 0  
## 23 brown.frag 0  
## 24 orang.frag 0  
## 25 black.frag 0  
## 26 whit.spher 0  
## 27 clear.spher 0

So we can remove the yellow, green, purple, gray, pink, brown, orange, and black fibre categories, and all other categories as well.

depwater4 <- subset(depwater3,   
 c(variable == "red.fib" |   
 variable == "turq.fib" |   
 variable == "blu.fib" |  
 variable == "clear.fib")) # keep only these rows  
  
depwater4$variable <- as.character(depwater4$variable)  
depwater4$variable <- as.factor(depwater4$variable) # gets rid of those names  
  
summary(depwater4$variable) # view

## blu.fib clear.fib red.fib turq.fib   
## 60 60 60 60

Let”s make the category names a bit more useful for plotting

depwater4$variable <- mapvalues(depwater4$variable,  
 from = c(levels(depwater4$variable)),   
 to = c("Blue fibres", "Clear fibres",   
 "Red fibres", "Turquoise fibres"))  
  
## And reorder them  
depwater4$variable <- factor(depwater4$variable,   
 levels = c("Blue fibres", "Clear fibres",  
 "Red fibres", "Turquoise fibres"))  
  
summary(depwater4$variable)

## Blue fibres Clear fibres Red fibres Turquoise fibres   
## 60 60 60 60

## Water blanks

Now do all these same things for the water blanks.

waterblanks <- read.csv("waterblanks.csv")  
  
head(waterblanks)

## id size.cat time.in.oven time.out.oven  
## 1 Water Control Control 1 twentyto50 2017-01-09/17:00 2017-01-17/09:30  
## 2 Water Control Control 1 fiftyto100 2017-01-09/17:00 2017-01-17/09:30  
## 3 Water Control Control 1 onehundredto500 2017-01-09/17:00 2017-01-17/09:30  
## 4 Water Control Control 1 fivehundredto1000 2017-01-09/17:00 2017-01-17/09:30  
## 5 Water Control Control 1 onethousandto5000 2017-01-09/17:00 2017-01-17/09:30  
## 6 Water Control Control 2 twentyto50 2017-01-09/17:00 2017-01-17/09:30  
## dat.filter dat.count observer red.fib yell.fib green.fib turq.fib blu.fib  
## 1 12-13-2017 12-20-2017 MD 0 0 0 0 0  
## 2 12-13-2017 12-20-2017 MD 0 0 0 0 0  
## 3 12-13-2017 12-20-2017 MD 0 0 0 0 0  
## 4 12-13-2017 12-20-2017 MD 0 0 0 0 0  
## 5 12-13-2017 12-20-2017 MD 0 0 0 0 0  
## 6 12-13-2017 12-20-2017 MD 0 0 0 0 0  
## purp.fib gray.fib clear.fib pink.fib brown.fib orang.fib black.fib tot.fib  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0 0  
## red.frag yell.frag green.frag turq.frag blu.frag purp.frag gray.frag  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## clear.frag pink.frag brown.frag orang.frag black.frag whit.spher clear.spher  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0

waterblanks$id <- as.factor(waterblanks$id)  
waterblanks$size.cat <- as.factor(waterblanks$size.cat)  
waterblanks$observer <- as.factor(waterblanks$observer)

Now we make a new dataframe to work with and rename size categories.

waterblanks2 <- waterblanks  
  
waterblanks2$size.cat <- factor(waterblanks2$size.cat,   
 levels=c("twentyto50", "fiftyto100",  
 "onehundredto500", "fivehundredto1000",   
 "onethousandto5000")) # reorder  
  
  
waterblanks2$size.cat <- mapvalues(waterblanks2$size.cat,   
 from = c("twentyto50", "fiftyto100",   
 "onehundredto500",   
 "fivehundredto1000",   
 "onethousandto5000"),   
 to = c("20-50", "50-100", "100-500",  
 "500-1000", "1000-5000")) # rename  
summary(waterblanks2$size.cat) # view

## 20-50 50-100 100-500 500-1000 1000-5000   
## 3 3 3 3 3

Again, we melt the dataframe so that the particle types are organized by row instead of by column.

waterblanks3 <- melt(waterblanks2,  
 id.vars = c("id", "size.cat", "time.in.oven",   
 "time.out.oven", "dat.filter", "dat.count",   
 "observer"))  
  
head(waterblanks3) # now "variable" is particle category and "value is the count"

## id size.cat time.in.oven time.out.oven  
## 1 Water Control Control 1 20-50 2017-01-09/17:00 2017-01-17/09:30  
## 2 Water Control Control 1 50-100 2017-01-09/17:00 2017-01-17/09:30  
## 3 Water Control Control 1 100-500 2017-01-09/17:00 2017-01-17/09:30  
## 4 Water Control Control 1 500-1000 2017-01-09/17:00 2017-01-17/09:30  
## 5 Water Control Control 1 1000-5000 2017-01-09/17:00 2017-01-17/09:30  
## 6 Water Control Control 2 20-50 2017-01-09/17:00 2017-01-17/09:30  
## dat.filter dat.count observer variable value  
## 1 12-13-2017 12-20-2017 MD red.fib 0  
## 2 12-13-2017 12-20-2017 MD red.fib 0  
## 3 12-13-2017 12-20-2017 MD red.fib 0  
## 4 12-13-2017 12-20-2017 MD red.fib 0  
## 5 12-13-2017 12-20-2017 MD red.fib 0  
## 6 12-13-2017 12-20-2017 MD red.fib 0

Now sum by totals across all samples and particles sizes to get an idea of what we”re working with. First remove <100 micron particles.

waterblanks3 <- subset(waterblanks3,   
 size.cat != "20-50" &   
 size.cat != "50-100")  
  
waterblanks3$size.cat <- as.character(waterblanks3$size.cat)  
waterblanks3$size.cat <- as.factor(waterblanks3$size.cat)  
  
cat.sums.blanks <- aggregate(waterblanks3$value ~   
 waterblanks3$variable, FUN = sum) # sums everything  
print(cat.sums.blanks)

## waterblanks3$variable waterblanks3$value  
## 1 red.fib 0  
## 2 yell.fib 0  
## 3 green.fib 0  
## 4 turq.fib 0  
## 5 blu.fib 3  
## 6 purp.fib 0  
## 7 gray.fib 0  
## 8 clear.fib 0  
## 9 pink.fib 0  
## 10 brown.fib 0  
## 11 orang.fib 0  
## 12 black.fib 0  
## 13 tot.fib 3  
## 14 red.frag 0  
## 15 yell.frag 0  
## 16 green.frag 0  
## 17 turq.frag 0  
## 18 blu.frag 0  
## 19 purp.frag 0  
## 20 gray.frag 0  
## 21 clear.frag 0  
## 22 pink.frag 0  
## 23 brown.frag 0  
## 24 orang.frag 0  
## 25 black.frag 0  
## 26 whit.spher 0  
## 27 clear.spher 0

So we can remove the red, yellow, green, turquoise, purple, gray, clear, pink, brown, orange, and black fibre categories, and all other categories as well.

waterblanks4 <- subset(waterblanks3, variable == "blu.fib") # keep only these rows  
  
waterblanks4$variable <- as.character(waterblanks4$variable)  
waterblanks4$variable <- as.factor(waterblanks4$variable) # gets rid of those names  
  
summary(waterblanks4$variable) # view

## blu.fib   
## 9

Let”s make the category names a bit more useful for plotting

waterblanks4$variable <- mapvalues(waterblanks4$variable,  
 from = c(levels(waterblanks4$variable)),   
 to = "Blue fibres")  
  
summary(waterblanks4$variable)

## Blue fibres   
## 9

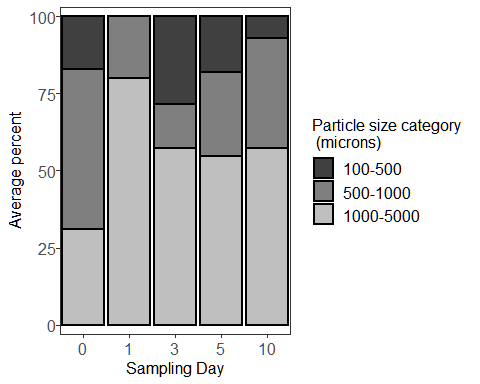
## Adjusted water data

Now subtract the blank counts from the appropriate size and shape categories for the water counts.

depwater4$dat.filter <- as.factor(depwater4$dat.filter)  
waterblanks4$dat.filter <- as.factor(waterblanks4$dat.filter)  
  
waterblanksmean <- ddply(waterblanks4, c("size.cat", "dat.filter",   
 "variable"),  
 summarize, value.b = mean(value))  
  
depwater5 <-  
 left\_join(depwater4, waterblanksmean,   
 by = c("dat.filter", "size.cat", "variable")) ## Join the 2 dataframes  
  
depwater5$value.b <- as.numeric(depwater5$value.b)  
depwater5$value.b[is.na(depwater5$value.b)] <- 0  
  
depwater5$variable <- as.factor(depwater5$variable)  
  
depwater5$finalcount <- (depwater5$value - depwater5$value.b)  
  
depwater5$finalcount[depwater5$finalcount < 0] <-  
 0 ## Change any negative values to 0  
  
summary(depwater5$finalcount)

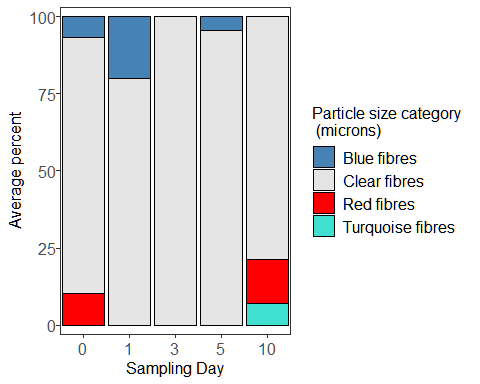
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.1792 0.0000 4.0000

## First we need to make a new data frame of sums according to each size category  
  
depwater5$sampleday <- factor(depwater5$sampleday,   
 levels =   
 c("0", "1", "3", "5", "10")) # put in order  
  
watersizesums <- ddply(depwater5,   
 c("id", "sampleday", "size.cat", "table", "tank",  
 "dat.filter", "dat.count", "observer"),   
 summarize,   
 sum = sum(finalcount))  
  
## Now calculate proportions of each particle size category present in each   
## sample and take the means for each sampling day  
  
waterprop <- ddply(watersizesums,  
 c("sampleday", "size.cat"),   
 summarize,   
 sum = sum(sum))  
  
waterprop$prop <- numeric(length = nrow(waterprop))  
  
for(i in 1:nrow(waterprop)) {  
 waterprop$prop[i] <-  
 (waterprop$sum[i] /  
 sum(waterprop$sum[waterprop$sampleday == waterprop$sampleday[i]]))  
}  
  
waterprop$size.cat <- as.character(waterprop$size.cat)  
waterprop$size.cat <- as.factor(waterprop$size.cat)  
  
## Put the size categories in order  
  
waterprop$size.cat <- factor(waterprop$size.cat,   
 levels = c("10-100", "100-500", "500-1000",   
 "1000-5000"))  
  
## Now plot  
  
ggplot(waterprop, aes(x=sampleday, y=100\*prop, fill = size.cat)) +   
 geom\_col(size = 1, colour = "black") +   
 xlab("Sampling Day") +   
 ylab("Average percent") +  
 guides(fill=guide\_legend(title="Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(values = c("grey25", "grey50", "grey75")) +  
 theme(legend.text = element\_text(size=12), text = element\_text(size=12),   
 panel.spacing = unit(1, "lines"),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12,   
 margin = margin(t=0, r=0, b=0, l=2)),  
 strip.background = element\_blank(),   
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside") +  
 scale\_x\_discrete(expand = expansion(0,0.5)) +   
 scale\_y\_continuous(expand = expansion(0.03,0))



Now with particle type/colour

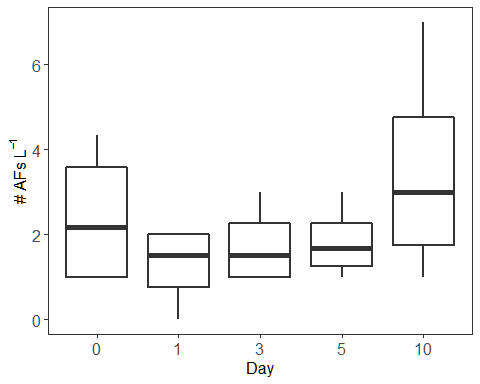
## First we need to make a new data frame of sums according to each particle  
## type/colour  
  
watertypesums <- ddply(depwater5, c("id", "sampleday", "table", "tank",   
 "dat.filter", "dat.count",   
 "observer", "variable"),   
 summarize, sum = sum(finalcount))  
  
  
  
## Now calculate proportions of each particle type/colour present in each   
## sample and take the means for each sampling day  
  
watertypeprop <- ddply(watertypesums,  
 c("sampleday", "variable"),   
 summarize,   
 sum = sum(sum))  
watertypeprop$prop <- numeric(length = nrow(watertypeprop))  
  
for(i in 1:nrow(watertypeprop)) {  
 watertypeprop$prop[i] <-  
 (watertypeprop$sum[i] /  
 sum(watertypeprop$sum[watertypeprop$sampleday ==  
 watertypeprop$sampleday[i]]))  
}  
  
watertypeprop$variable <- as.character(watertypeprop$variable)  
watertypeprop$variable <- as.factor(watertypeprop$variable)  
  
## Now plot  
  
ggplot(watertypeprop, aes(x=sampleday, y=100\*prop, fill = variable)) +   
 geom\_col(size = 0.5, colour = "black") +   
 xlab("Sampling Day") +   
 ylab("Average percent") +  
 guides(fill=guide\_legend(title="Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(values = c("steel blue", "grey90", "red", "turquoise")) +  
 theme(legend.text = element\_text(size=12), text = element\_text(size=12),   
 panel.spacing = unit(1, "lines"),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12,   
 margin = margin(t=0, r=0, b=0, l=2)),  
 strip.background = element\_blank(),   
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside") +  
 scale\_x\_discrete(expand = expansion(0,0.5)) +   
 scale\_y\_continuous(expand = expansion(0.03,0))



Take total sums and plot the results.

totalwatersums <- ddply(depwater5, c("id", "sampleday", "table", "tank",   
 "observer", "dat.filter"),  
 summarise, sum = sum(finalcount)) # adds all the counts together  
  
## Plot according to count per filtration day  
  
ggplot(totalwatersums, aes(x=sampleday, y=sum)) +   
 geom\_boxplot(position=position\_dodge(), size=1) +   
 xlab("Day") +  
 ylab(expression(paste("# AFs"~L^-1))) +  
 guides(fill=FALSE) +  
 theme\_bw() +  
 theme(text = element\_text(size=12),   
 axis.text.x = element\_text(size = 12),  
 axis.text.y = element\_text(size = 12),  
 strip.background = element\_blank(),  
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.



Now run the same summary stats for the water samples.

## medians, range, and SD  
  
with(totalwatersums, tapply(sum, sampleday, median)) # takes median for each sampleday

## 0 1 3 5 10   
## 2.166667 1.500000 1.500000 1.666667 3.000000

with(totalwatersums, tapply(sum, sampleday, min)) # takes minimum for each sampleday

## 0 1 3 5 10   
## 1 0 1 1 1

with(totalwatersums, tapply(sum, sampleday, max)) # takes maximum for each sampleday

## 0 1 3 5 10   
## 4.333333 2.000000 3.000000 3.000000 7.000000

with(totalwatersums, tapply(sum,   
 sampleday, sd)) # takes standard deviation for each sampleday

## 0 1 3 5 10   
## 1.6859990 0.9574271 0.9574271 0.8819171 2.6457513

## Proportions of particles sizes  
  
sum(depwater5$finalcount[depwater5$size.cat == "100-500"])/  
 sum(depwater5$finalcount)\*100

## [1] 13.95349

sum(depwater5$finalcount[depwater5$size.cat == "500-1000"])/  
 sum(depwater5$finalcount)\*100

## [1] 32.55814

sum(depwater5$finalcount[depwater5$size.cat == "1000-5000"])/  
 sum(depwater5$finalcount)\*100

## [1] 53.48837

## This provides the percentage of total particles found accounted for by each   
## colour/type of particles  
watersummarytable <- with(depwater5, tapply(value, variable, sum))  
watersummarytable <- watersummarytable/sum(depwater5$finalcount)\*100  
print(watersummarytable)

## Blue fibres Clear fibres Red fibres Turquoise fibres   
## 9.302326 86.046512 6.976744 2.325581

## Algae data

Now load in the algae sample data.

algae <-   
 read.csv("algae.csv")  
  
head(algae)

## id species size.cat red.fib yell.fib  
## 1 Algae Nov 9 #1Ts Tetraselmis suecica twentyto50 0 0  
## 2 Algae Nov 9 #1Ts Tetraselmis suecica fiftyto100 0 0  
## 3 Algae Nov 9 #1Ts Tetraselmis suecica onehundredto500 1 0  
## 4 Algae Nov 9 #1Ts Tetraselmis suecica fivehundredto1000 0 0  
## 5 Algae Nov 9 #1Ts Tetraselmis suecica onethousandto5000 0 0  
## 6 Algae Nov 9 # 2Cm Chaetoceros muelleri twentyto50 0 0  
## green.fib turq.fib blu.fib purp.fib gray.fib clear.fib pink.fib brown.fib  
## 1 0 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 1 0 0  
## 4 0 0 0 0 0 2 0 0  
## 5 0 0 0 0 0 3 0 0  
## 6 0 0 0 0 0 0 0 0  
## orang.fib black.fib red.frag yell.frag green.frag turq.frag blu.frag  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## purp.frag gray.frag clear.frag pink.frag brown.frag orang.frag black.frag  
## 1 0 0 0 0 0 0 0  
## 2 0 0 0 0 0 0 0  
## 3 0 0 0 0 0 0 0  
## 4 0 0 0 0 0 0 0  
## 5 0 0 0 0 0 0 0  
## 6 0 0 0 0 0 0 0  
## whit.spher clear.spher  
## 1 0 0  
## 2 0 0  
## 3 0 0  
## 4 0 0  
## 5 0 0  
## 6 0 0

algae$id <- as.factor(algae$id)  
algae$size.cat <- as.factor(algae$size.cat)  
algae$species <- as.factor(algae$species)

Now we make a new dataframe to work with and rename size categories.

algae2 <- algae  
  
algae2$size.cat <- factor(algae2$size.cat,   
 levels=c("twentyto50", "fiftyto100",  
 "onehundredto500", "fivehundredto1000",   
 "onethousandto5000")) # reorder  
  
  
algae2$size.cat <- mapvalues(algae2$size.cat,   
 from = c("twentyto50", "fiftyto100",   
 "onehundredto500",   
 "fivehundredto1000",   
 "onethousandto5000"),   
 to = c("20-50", "50-100", "100-500",  
 "500-1000", "1000-5000")) # rename  
summary(algae2$size.cat) # view

## 20-50 50-100 100-500 500-1000 1000-5000   
## 9 9 9 9 9

Again, we melt the dataframe so that the particle types are organized by row instead of by column.

algae3 <- melt(algae2,  
 id.vars = c("id", "size.cat", "species"))  
  
head(algae3) # now "variable" is particle category and "value is the count"

## id size.cat species variable value  
## 1 Algae Nov 9 #1Ts 20-50 Tetraselmis suecica red.fib 0  
## 2 Algae Nov 9 #1Ts 50-100 Tetraselmis suecica red.fib 0  
## 3 Algae Nov 9 #1Ts 100-500 Tetraselmis suecica red.fib 1  
## 4 Algae Nov 9 #1Ts 500-1000 Tetraselmis suecica red.fib 0  
## 5 Algae Nov 9 #1Ts 1000-5000 Tetraselmis suecica red.fib 0  
## 6 Algae Nov 9 # 2Cm 20-50 Chaetoceros muelleri red.fib 0

Now sum by totals across all samples and particles sizes to get an idea of what we”re working with. First remove particles <100 microns as we don”t trust these counts.

algae3 <- subset(algae3, size.cat != "20-50" & size.cat != "50-100")  
  
algae3$size.cat <- as.character(algae3$size.cat)  
algae3$size.cat <- as.factor(algae3$size.cat)  
  
cat.sums.blanks <- aggregate(algae3$value ~   
 algae3$variable, FUN = sum) # sums everything  
print(cat.sums.blanks)

## algae3$variable algae3$value  
## 1 red.fib 1  
## 2 yell.fib 0  
## 3 green.fib 0  
## 4 turq.fib 1  
## 5 blu.fib 1  
## 6 purp.fib 0  
## 7 gray.fib 0  
## 8 clear.fib 46  
## 9 pink.fib 0  
## 10 brown.fib 0  
## 11 orang.fib 0  
## 12 black.fib 0  
## 13 red.frag 0  
## 14 yell.frag 0  
## 15 green.frag 0  
## 16 turq.frag 0  
## 17 blu.frag 0  
## 18 purp.frag 0  
## 19 gray.frag 0  
## 20 clear.frag 0  
## 21 pink.frag 0  
## 22 brown.frag 0  
## 23 orang.frag 0  
## 24 black.frag 0  
## 25 whit.spher 0  
## 26 clear.spher 0

So we can remove the yellow, green, purple, gray, pink, brown, orange, and black fibre categories, and all other categories as well.

algae4 <- subset(algae3, c(variable == "red.fib" |   
 variable == "turq.fib" |  
 variable == "blu.fib" |  
 variable == "clear.fib")) # keep only these rows  
  
algae4$variable <- as.character(algae4$variable)  
algae4$variable <- as.factor(algae4$variable) # gets rid of those names  
  
summary(algae4$variable) # view

## blu.fib clear.fib red.fib turq.fib   
## 27 27 27 27

Let”s make the category names a bit more useful for plotting

algae4$variable <- mapvalues(algae4$variable,  
 from = c(levels(algae4$variable)),   
 to = c("Blue fibres", "Clear fibres",   
 "Red fibres", "Turquoise fibres"))  
  
## And reorder them  
algae4$variable <- factor(algae4$variable,   
 levels = c("Blue fibres", "Clear fibres",  
 "Red fibres", "Turquoise fibres"))  
  
summary(algae4$variable)

## Blue fibres Clear fibres Red fibres Turquoise fibres   
## 27 27 27 27

## Adjusted algae data

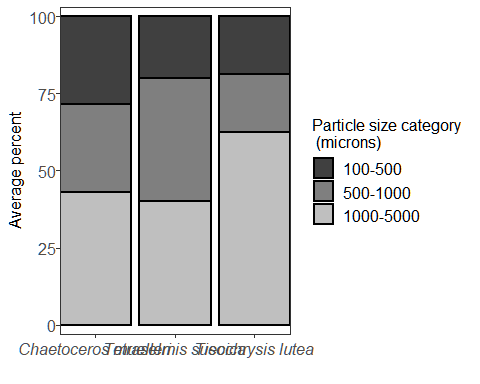
Now subtract the blank counts from the appropriate size and shape categories for the algae counts.

algae5 <-  
 left\_join(algae4, waterblanksmean,   
 by = c("size.cat", "variable")) ## Join the 2 dataframes  
  
algae5$value.b <- as.numeric(algae5$value.b)  
algae5$value.b[is.na(algae5$value.b)] <- 0  
  
algae5$variable <- as.factor(algae5$variable)  
  
algae5$finalcount <- (algae5$value - algae5$value.b)  
  
algae5$finalcount[algae5$finalcount < 0] <-  
 0 ## Change any negative values to 0  
  
summary(algae5$finalcount)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.4537 0.0000 15.0000

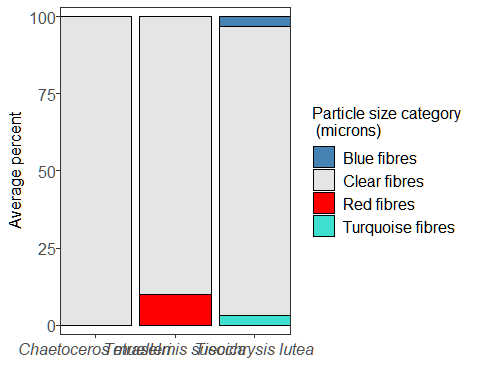
Now plot particle size proportions for algae.

## First we need to make a new data frame of sums according to each size category  
  
algaesizesums <-   
 ddply(algae5,   
 c("id", "size.cat", "species"),   
 summarize, sum = sum(finalcount))  
  
  
  
## Now calculate proportions of each particle size category  
  
algaeprop <- ddply(algaesizesums,  
 c("size.cat", "species"),   
 summarize,   
 sum = sum(sum))  
  
for(i in 1:nrow(algaeprop)) {  
 algaeprop$prop[i] <-  
 (algaeprop$sum[i] /  
 sum(algaeprop$sum[algaeprop$species == algaeprop$species[i]]))  
}  
  
algaeprop$size.cat <- as.character(algaeprop$size.cat)  
algaeprop$size.cat <- as.factor(algaeprop$size.cat)  
  
## Put the size categories in order  
  
algaeprop$size.cat <- factor(algaeprop$size.cat,   
 levels = c("100-500", "500-1000",   
 "1000-5000"))  
  
## Now plot  
  
ggplot(algaeprop, aes(x=species, y=100\*prop, fill = size.cat)) +   
 geom\_col(size = 1, colour = "black") +   
 xlab("") +   
 ylab("Average percent") +  
 guides(fill=guide\_legend(title="Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(values = c("grey25", "grey50", "grey75")) +  
 theme(legend.text = element\_text(size=12), text = element\_text(size=12),   
 panel.spacing = unit(1, "lines"),   
 axis.text.x = element\_text(size = 12,  
 face = "italic"),  
 axis.text.y = element\_text(size = 12,   
 margin = margin(t=0, r=0, b=0, l=2)),  
 strip.background = element\_blank(),   
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside") +  
 scale\_x\_discrete(expand = c(0,0)) +   
 scale\_y\_continuous(expand = c(0.03,0))



Now with particle type/colour

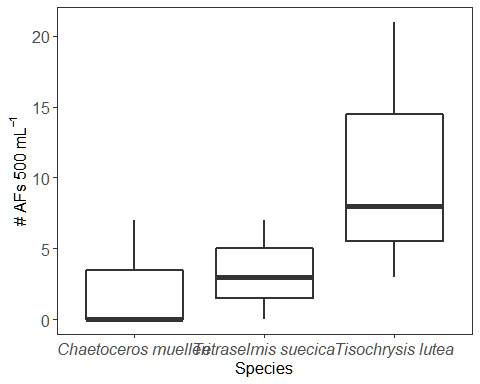
## First we need to make a new data frame of sums according to each particle  
## type/colour  
  
algaetypesums <-   
 ddply(algae5,   
 c("id",   
 "variable",  
 "species"),  
 summarize, sum = sum(finalcount))  
  
  
  
## Now calculate proportions of each particle type/colour present in each   
## sample and take the means for each sampling day  
  
algaetypeprop <- ddply(algaetypesums,  
 c("variable",  
 "species"),   
 summarize,   
 sum = sum(sum))  
algaetypeprop$prop <- numeric(length = nrow(algaetypeprop))  
  
for(i in 1:nrow(algaetypeprop)) {  
 algaetypeprop$prop[i] <-  
 (algaetypeprop$sum[i] /  
 sum(algaetypeprop$sum[algaetypeprop$species ==   
 algaetypeprop$species[i]]))  
}  
  
watertypeprop$variable <- as.character(watertypeprop$variable)  
watertypeprop$variable <- as.factor(watertypeprop$variable)  
  
## Now plot  
  
ggplot(algaetypeprop, aes(x=species, y=100\*prop, fill = variable)) +   
 geom\_col(size = 0.5, colour = "black") +   
 xlab("") +   
 ylab("Average percent") +  
 guides(fill=guide\_legend(title="Particle size category \n (microns)")) +  
 theme\_bw() +  
 scale\_fill\_manual(values = c("steel blue", "grey90", "red", "turquoise")) +  
 theme(legend.text = element\_text(size=12), text = element\_text(size=12),   
 panel.spacing = unit(1, "lines"),   
 axis.text.x = element\_text(size = 12,  
 face = "italic"),  
 axis.text.y = element\_text(size = 12,   
 margin = margin(t=0, r=0, b=0, l=2)),  
 strip.background = element\_blank(),   
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank(),  
 strip.placement = "outside") +  
 scale\_x\_discrete(expand = c(0,0)) +   
 scale\_y\_continuous(expand = c(0.03,0))



Take total sums and plot the results.

totalalgaesums <- ddply(algae5,   
 c("id", "species"),  
 summarise, sum = sum(finalcount)) # adds all the counts together  
  
## Plot according to count per filtration day  
  
ggplot(totalalgaesums, aes(x=species, y=sum)) +   
 geom\_boxplot(position=position\_dodge(), size=1) +   
 xlab("Species") +  
 ylab(expression(paste("# AFs 500"~mL^-1))) +  
 guides(fill=FALSE) +  
 theme\_bw() +  
 theme(text = element\_text(size=12),   
 axis.text.x = element\_text(size = 12,  
 face = "italic"),  
 axis.text.y = element\_text(size = 12),  
 strip.background = element\_blank(),  
 panel.grid.major = element\_blank(),   
 panel.grid.minor = element\_blank())

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.



Now run the same summary stats for the algae samples.

## medians, range, and SD  
  
with(totalalgaesums, tapply(sum, species, median)) # takes median for each species

## Chaetoceros muelleri Tetraselmis suecica Tisochrysis lutea   
## 0 3 8

with(totalalgaesums, tapply(sum, species, min)) # takes minimum for each species

## Chaetoceros muelleri Tetraselmis suecica Tisochrysis lutea   
## 0 0 3

with(totalalgaesums, tapply(sum, species, max)) # takes maximum for each species

## Chaetoceros muelleri Tetraselmis suecica Tisochrysis lutea   
## 7 7 21

with(totalalgaesums, tapply(sum,   
 species, sd)) # takes standard deviation for each species

## Chaetoceros muelleri Tetraselmis suecica Tisochrysis lutea   
## 4.041452 3.511885 9.291573

## Proportions of particles sizes  
  
sum(algae5$finalcount[algae5$size.cat == "100-500"])/  
 sum(algae5$finalcount)\*100

## [1] 20.40816

sum(algae5$finalcount[algae5$size.cat == "500-1000"])/  
 sum(algae5$finalcount)\*100

## [1] 24.4898

sum(algae5$finalcount[algae5$size.cat == "1000-5000"])/  
 sum(algae5$finalcount)\*100

## [1] 55.10204

## This provides the percentage of total particles found accounted for by each   
## colour/type of particles  
algaesummarytable <- with(algae5, tapply(value, variable, sum))  
algaesummarytable <- algaesummarytable/sum(algae5$finalcount)\*100  
print(algaesummarytable)

## Blue fibres Clear fibres Red fibres Turquoise fibres   
## 2.040816 93.877551 2.040816 2.040816

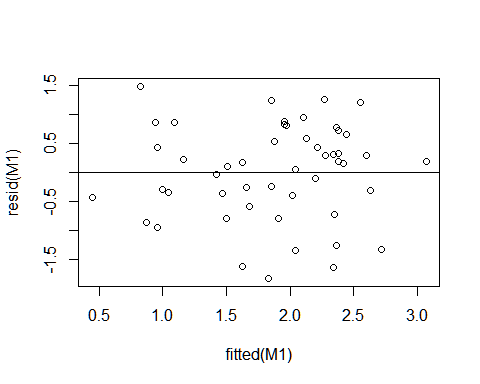
## Oyster statistics

Now fit a generalized linear mixed model (GLMM) to the oyster data using the package glmmTMB to determine differences over time. We”ll specify AP count as the response variable, sampling day and oyster width as predictors, and tank nested within table as the random, or mixed, effects.

# Gaussian  
  
totalsums$sampleday <- as.character(totalsums$sampleday)  
totalsums$sampleday <- as.numeric(totalsums$sampleday)  
  
M1 <-   
 glmmTMB(log(sum + 1) ~ sampleday + width + (1 | table/tank),  
 data = totalsums) ## Fits a GLMM  
  
summary(M1) # model output

## Family: gaussian ( identity )  
## Formula: log(sum + 1) ~ sampleday + width + (1 | table/tank)  
## Data: totalsums  
##   
## AIC BIC logLik deviance df.resid   
## 136.8 148.3 -62.4 124.8 44   
##   
## Random effects:  
##   
## Conditional model:  
## Groups Name Variance Std.Dev.   
## tank:table (Intercept) 3.274e-09 5.722e-05  
## table (Intercept) 3.547e-02 1.883e-01  
## Residual 6.818e-01 8.257e-01  
## Number of obs: 50, groups: tank:table, 11; table, 5  
##   
## Dispersion estimate for gaussian family (sigma^2): 0.682   
##   
## Conditional model:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.55916 1.10303 0.507 0.6122   
## sampleday -0.18711 0.04486 -4.171 3.03e-05 \*\*\*  
## width 0.02500 0.01422 1.758 0.0787 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(resid(M1) ~ fitted(M1)) # check assumptions  
abline(0,0)



Model assumptions seem to be met.

Sample day significant (p<0.001), oyster width not significant (p = 0.08).

## Plot model predictions

predictions <-   
 data.frame(sampleday = seq(from = 0,  
 to = 10,  
 length.out = 1000),  
 width = with(totalsums,  
 sample(c(min(width),  
 median(width),  
 max(width)),  
 1000,  
 replace = TRUE)),  
 table = rep(NA, 1000), # ignore random effects  
 tank = rep(NA, 1000))  
  
new.predict <-  
 predict(M1,  
 newdata = predictions,  
 re.form = NA,  
 type = "link",  
 se.fit = TRUE)  
  
predictions$mean <- exp(new.predict$fit) - 1  
predictions$upper <- exp(with(new.predict, fit + 1.96\*se.fit)) - 1  
predictions$lower <- exp(with(new.predict, fit - 1.96\*se.fit)) - 1  
  
predictions$width <- as.factor(predictions$width)  
  
ggplot() +  
 geom\_ribbon(data = predictions,  
 aes(x = sampleday,  
 ymin = lower,  
 ymax = upper,  
 fill = width),  
 alpha = 0.5) +  
 geom\_line(data = predictions,  
 aes(x = sampleday,  
 y = mean,  
 colour = width),  
 size = 1) +  
 geom\_point(data = totalsums,  
 aes(x = sampleday,  
 y = sum),  
 size = 0.75) +  
 scale\_colour\_brewer(type = "qual",  
 palette = "Set2",  
 name = "Width (mm)") +  
  
 scale\_fill\_brewer(type = "qual",  
 palette = "Set2",  
 name = "Width (mm)") +  
 labs(x = "Day",  
 y = "Number of anthropogenic particles") +  
 theme1

