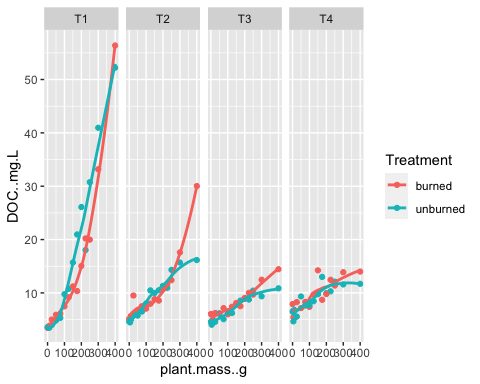
Cody

Cody Spiegel

4/22/2022

###DOC  
import and do a loop to clean up all files and make stacked data in single df

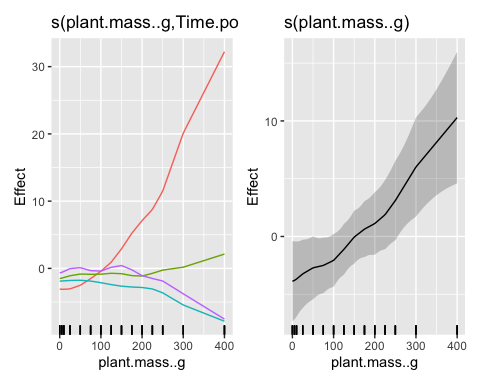
## import treatment IDs  
IDs<-read.csv("/Users/tobyspiegel2/Desktop/My\_pyromania/Pyromania/data/treatment.IDs.csv")  
  
##### grab files in a list  
Total.DOC.files <- list.files(path="/Users/tobyspiegel2/Desktop/My\_pyromania/Pyromania/data/DOC.TN", pattern = "csv$", full.names = T)  
  
##### what are the file names, sans extensions using package 'tools'  
file.names<-file\_path\_sans\_ext(list.files(path="/Users/tobyspiegel2/Desktop/My\_pyromania/Pyromania/data/DOC.TN", pattern = "csv$", full.names = F))  
  
############ formatting all data in for loop  
 for(i in 1:length(Total.DOC.files))  
 {  
 data<-read.csv(Total.DOC.files[i], sep=",")  
 data<-data[,c(1,3,4)] # removed columns we don't need  
 data$File<-Total.DOC.files[i]  
 colnames(data)<-c("Tank", "DOC..mg.L", "TN..mg.L", "File")  
 data$Tank<- IDs$Tank  
 data$Tank<-as.numeric(as.character(data$Tank)) # make the column of samples all numeric  
 data <- data[!is.na(as.numeric(as.character(data$Tank))),] # remove all rows that aren't numeric/tanks  
 data$Treatment<-IDs$Treatment  
 data$plant.mass..g<-IDs$plant.mass..g  
 make.names(assign(paste(file.names[i], sep=""), data)) # make the file name the name of new df for loop df  
 }  
  
########## this is the end of the loop  
  
#Combine files from loop to single df  
DOC.df<-rbind(DOC\_T1, DOC\_T2, DOC\_T3, DOC\_T4)  
DOC.df$Treatment=as.factor(DOC.df$Treatment)  
  
DOC.df$File<-substr(DOC.df$File, 64, 74)   
# alternative way to code the above  
#give the 10th-24th character of the file name, removing the rest  
#DOC.df$File<-substr(DOC.df$File, 13, 27)   
  
#alternatively  
# remove the 9 letters ('^.) at start   
# remove the 4 letters (.$') at end  
#DOC.df$File<-gsub('^.........|....$', '', DOC.df$File)   
  
# if equals DOC\_T0\_11052021 then, T0, if not then T1  
DOC.df$Time.point<- as.factor(  
 ifelse (DOC.df$File=="DOC\_T1.csv", "T1",  
 ifelse (DOC.df$File=="DOC\_T2.csv", "T2",   
 ifelse(DOC.df$File=="DOC\_T3.csv", "T3", "T4"))))  
  
#rearrange  
library(dplyr)  
DOC.df<- DOC.df %>%   
 dplyr::select(File, Time.point, Treatment, Tank, plant.mass..g, DOC..mg.L, TN..mg.L)   
  
#visualize  
ggplot(DOC.df, aes(x = plant.mass..g, y = DOC..mg.L, colour = Treatment)) +  
 geom\_point() +  
 facet\_grid(~Time.point)+  
 geom\_smooth(method = 'loess', se = FALSE)



library(mgcv)  
library(gratia)  
#A single common smoother plus group-level smoothers that have the same wiggliness (model GS)  
m1.gam=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, k=8, m=1) + s(plant.mass..g, Time.point, k=8, bs="fs", m=1), data=DOC.df, method="REML")  
summary.gam(m1.gam)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, k = 8, m = 1) + s(plant.mass..g,   
## Time.point, k = 8, bs = "fs", m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.46776 1.70457 6.141 2.07e-08 \*\*\*  
## Treatmentunburned 0.06059 0.40248 0.151 0.881   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g) 4.412 7 1.923 <2e-16 \*\*\*  
## s(plant.mass..g,Time.point) 21.799 31 25.369 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.931 Deviance explained = 94.6%  
## -REML = 302.81 Scale est. = 4.8597 n = 120

draw(m1.gam)



k.check(m1.gam)

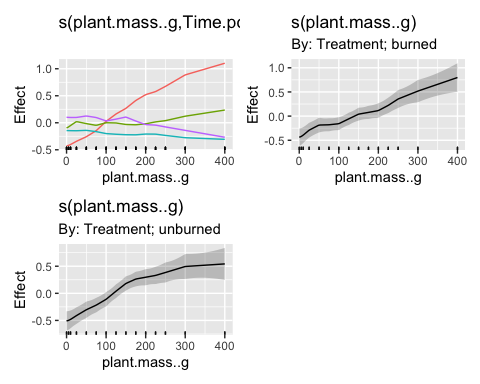
## k' edf k-index p-value  
## s(plant.mass..g) 7 4.412291 1.530985 1  
## s(plant.mass..g,Time.point) 32 21.799093 1.530985 1

I think this one is the best fit

#first model is that the reference level of mass is modeled as a constant term (it is the intercept), with the effect of mass for the other levels being smooth differences from this constant.  
m2.gam=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, by=Treatment, k=8, m=1) + s(plant.mass..g, Time.point, k=11, bs="fs", m=1), data=DOC.df, family= Gamma("log"), method="REML", select = T)  
summary(m2.gam)

##   
## Family: Gamma   
## Link function: log   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, by = Treatment, k = 8,   
## m = 1) + s(plant.mass..g, Time.point, k = 11, bs = "fs",   
## m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.18840 0.08174 26.773 <2e-16 \*\*\*  
## Treatmentunburned -0.02554 0.02667 -0.958 0.341   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g):Treatmentburned 5.494 7 5.428 <2e-16 \*\*\*  
## s(plant.mass..g):Treatmentunburned 5.336 7 5.333 <2e-16 \*\*\*  
## s(plant.mass..g,Time.point) 21.975 43 9.960 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.936 Deviance explained = 96%  
## -REML = 242.69 Scale est. = 0.021344 n = 120

draw(m2.gam)



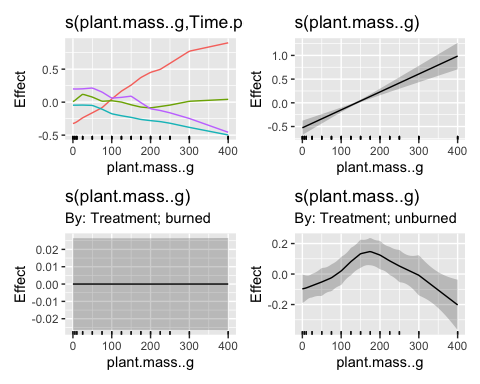
k.check(m2.gam)

## k' edf k-index p-value  
## s(plant.mass..g):Treatmentburned 7 5.494071 1.263143 1  
## s(plant.mass..g):Treatmentunburned 7 5.336489 1.263143 1  
## s(plant.mass..g,Time.point) 44 21.974796 1.263143 1

#in the second model, add s(mass), which then models the smooth effect of mass in the reference level. Now, the by smooths model smooth differences from this no-longer-constant reference smooth.  
m2.gamB=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, bs="tp", k=8) + s(plant.mass..g, by=Treatment, k=8, m=1) + s(plant.mass..g, Time.point, k=11, bs="fs", m=1), data=DOC.df, method="REML", family= Gamma(link = "log"), select = T)  
summary(m2.gamB)

##   
## Family: Gamma   
## Link function: log   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, bs = "tp", k = 8) +   
## s(plant.mass..g, by = Treatment, k = 8, m = 1) + s(plant.mass..g,   
## Time.point, k = 11, bs = "fs", m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.18919 0.08214 26.653 <2e-16 \*\*\*  
## Treatmentunburned -0.02606 0.02693 -0.968 0.336   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g) 9.808e-01 7 6.880 < 2e-16 \*\*\*  
## s(plant.mass..g):Treatmentburned 4.393e-04 7 0.000 0.175769   
## s(plant.mass..g):Treatmentunburned 4.229e+00 7 2.802 0.000192 \*\*\*  
## s(plant.mass..g,Time.point) 2.268e+01 43 9.702 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.941 Deviance explained = 95.7%  
## -REML = 235.11 Scale est. = 0.02176 n = 120

draw(m2.gamB)

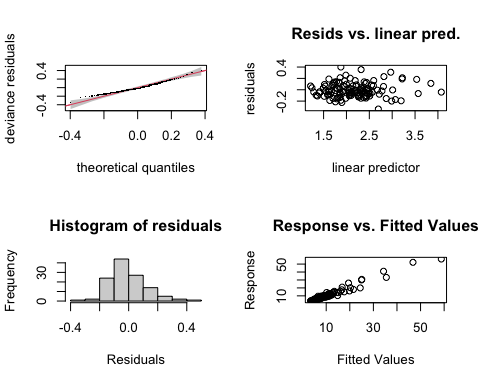


# the second model, all the levels of Treatment respond similarly to mass hence there are no large deviations from the smooth for the reference level of Treatment and hence the terms are not significant. the model is set up like ANOVA contrasts (estimate an effect for the reference level then have differences between individual levels and the reference), then fit the model like model 2  
  
#In the first model, the effect of mass for the reference level was constant, so the difference smooths picked up the actual non-linear effect of mass and hence were significantly different from zero

AIC(m2.gam, m2.gamB)

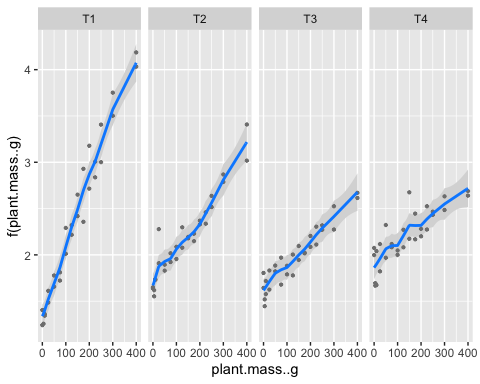
## df AIC  
## m2.gam 36.45139 432.9314  
## m2.gamB 31.79645 430.1705

#gamB seems to be the better fit  
gam.check(m2.gamB, rep=1000)



##   
## Method: REML Optimizer: outer newton  
## full convergence after 18 iterations.  
## Gradient range [-7.241876e-05,0.0001773987]  
## (score 235.1078 & scale 0.02175981).  
## Hessian positive definite, eigenvalue range [3.094414e-05,61.42111].  
## Model rank = 67 / 67   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value  
## s(plant.mass..g) 7.00e+00 9.81e-01 1.25 1  
## s(plant.mass..g):Treatmentburned 7.00e+00 4.39e-04 1.25 1  
## s(plant.mass..g):Treatmentunburned 7.00e+00 4.23e+00 1.25 1  
## s(plant.mass..g,Time.point) 4.40e+01 2.27e+01 1.25 1

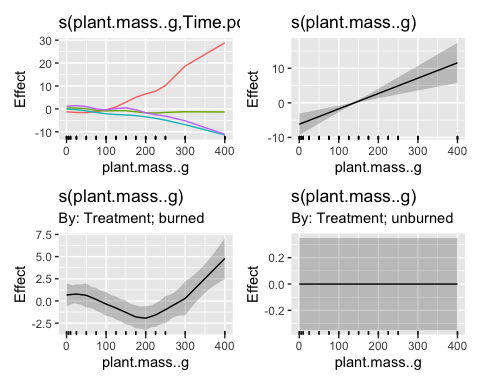
library(visreg)  
visreg(m2.gamB, xvar = "plant.mass..g",  
 by = "Time.point", data = DOC.df,  
 method = "REML", gg=T)



m3.gam=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, bs="tp", k=8) + s(plant.mass..g, by=Treatment, k=8, m=1) + s(plant.mass..g, Time.point, k=8, bs="fs", m=1), data=DOC.df, method="REML", select = T)  
summary(m3.gam)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, bs = "tp", k = 8) +   
## s(plant.mass..g, by = Treatment, k = 8, m = 1) + s(plant.mass..g,   
## Time.point, k = 8, bs = "fs", m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.46776 1.70203 6.15 2.21e-08 \*\*\*  
## Treatmentunburned 0.06059 0.35702 0.17 0.866   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g) 9.394e-01 7 2.208 < 2e-16 \*\*\*  
## s(plant.mass..g):Treatmentburned 4.609e+00 7 3.698 2.89e-05 \*\*\*  
## s(plant.mass..g):Treatmentunburned 2.070e-05 7 0.000 0.102   
## s(plant.mass..g,Time.point) 2.434e+01 31 32.372 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.945 Deviance explained = 96%  
## -REML = 292.41 Scale est. = 3.8239 n = 120

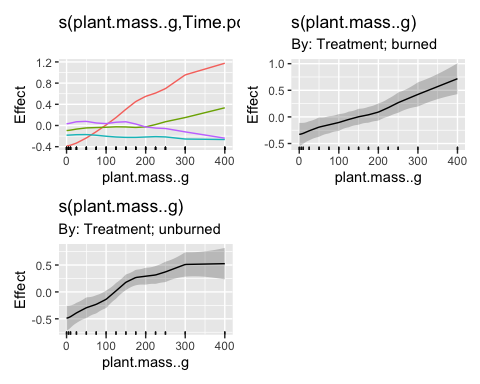
draw(m3.gam)



#with link= log  
m4.gam=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, by=Treatment, k=8, m=1) + s(plant.mass..g, Time.point, k=8, bs="fs", m=1),family = gaussian(link = "log"), data=DOC.df, method="REML", select=T)  
summary(m4.gam)

##   
## Family: gaussian   
## Link function: log   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, by = Treatment, k = 8,   
## m = 1) + s(plant.mass..g, Time.point, k = 8, bs = "fs", m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.17339 0.09300 23.369 <2e-16 \*\*\*  
## Treatmentunburned 0.03042 0.04256 0.715 0.477   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g):Treatmentburned 4.576 7 3.517 <2e-16 \*\*\*  
## s(plant.mass..g):Treatmentunburned 5.038 7 4.193 <2e-16 \*\*\*  
## s(plant.mass..g,Time.point) 17.406 31 31.844 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.952 Deviance explained = 96.4%  
## -REML = 283.07 Scale est. = 3.342 n = 120

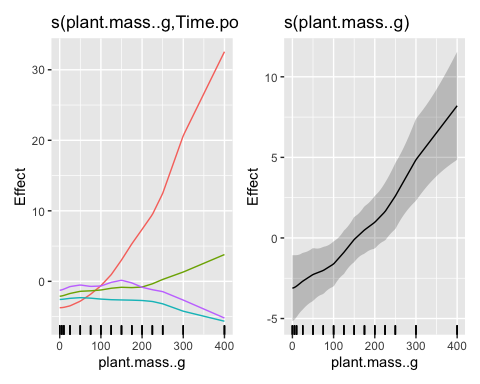
draw(m4.gam)



m5.gam=gam(DOC..mg.L~ Treatment + s(plant.mass..g, bs="tp", k=8, m=1)+ s(plant.mass..g, Time.point, bs="fs", k=8, m=1),#time point specific trends  
 data= DOC.df,  
 methods= 'REML',  
 select=TRUE)  
summary(m5.gam)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, bs = "tp", k = 8, m = 1) +   
## s(plant.mass..g, Time.point, bs = "fs", k = 8, m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.46776 1.70463 6.141 1.86e-08 \*\*\*  
## Treatmentunburned 0.06059 0.40454 0.150 0.881   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g) 3.105 7 3.513 <2e-16 \*\*\*  
## s(plant.mass..g,Time.point) 19.009 31 24.769 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.93 Deviance explained = 94.4%  
## GCV = 6.1443 Scale est. = 4.9096 n = 120

draw(m5.gam)

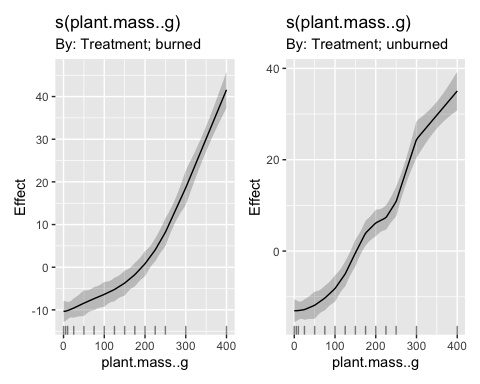


Separate models for each time point

DOC\_T1$Treatment=as.factor(DOC\_T1$Treatment)  
m1=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, by= Treatment, k=8, m=1), data= DOC\_T1, method= "REML")  
summary.gam(m1)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, by = Treatment, k = 8,   
## m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 14.0557 0.5796 24.25 1.27e-13 \*\*\*  
## Treatmentunburned 2.6145 0.8196 3.19 0.00597 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g):Treatmentburned 6.395 7 79.71 <2e-16 \*\*\*  
## s(plant.mass..g):Treatmentunburned 6.315 7 87.83 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.976 Deviance explained = 98.7%  
## -REML = 85.055 Scale est. = 5.0384 n = 30

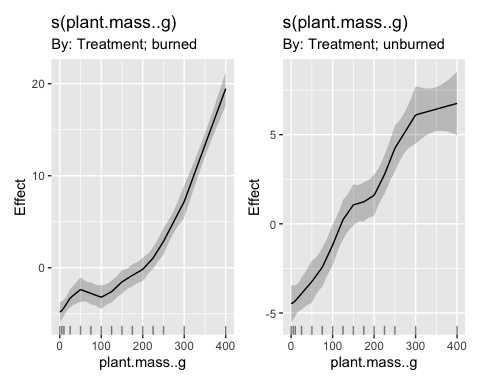
draw(m1)



DOC\_T2$Treatment=as.factor(DOC\_T2$Treatment)  
m2=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, by= Treatment, k=8, m=1), data= DOC\_T2, method= "REML")  
summary.gam(m2)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, by = Treatment, k = 8,   
## m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.2690 0.2513 40.863 <2e-16 \*\*\*  
## Treatmentunburned -0.9426 0.3554 -2.652 0.0174 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g):Treatmentburned 6.520 7 83.51 <2e-16 \*\*\*  
## s(plant.mass..g):Treatmentunburned 5.461 7 31.70 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.966 Deviance explained = 98.1%  
## -REML = 59.395 Scale est. = 0.94729 n = 30

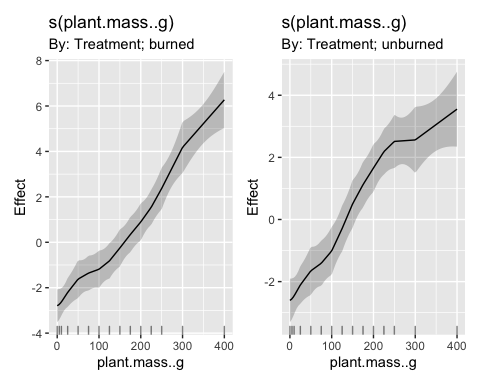
draw(m2)



DOC\_T3$Treatment=as.factor(DOC\_T3$Treatment)  
m3=gam(DOC..mg.L ~ Treatment + s(plant.mass..g, by= Treatment, k=8, m=1), data= DOC\_T3, method= "REML")  
summary.gam(m3)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## DOC..mg.L ~ Treatment + s(plant.mass..g, by = Treatment, k = 8,   
## m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.9565 0.1746 45.568 < 2e-16 \*\*\*  
## Treatmentunburned -0.8431 0.2469 -3.414 0.00319 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g):Treatmentburned 5.513 7 31.85 <2e-16 \*\*\*  
## s(plant.mass..g):Treatmentunburned 4.951 7 20.75 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.929 Deviance explained = 95.7%  
## -REML = 43.734 Scale est. = 0.4573 n = 30

draw(m3)

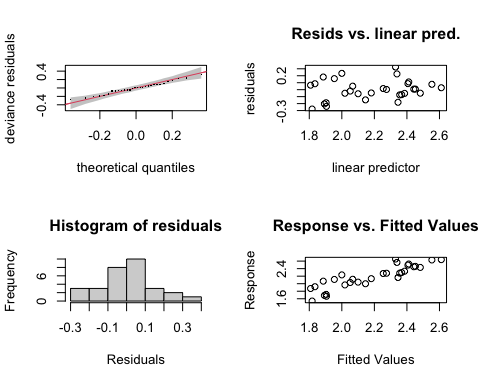


Not sure why T4 is being difficult…

DOC\_T4$Treatment=as.factor(DOC\_T4$Treatment)  
m4=gam(log(DOC..mg.L) ~ Treatment + s(plant.mass..g, by= Treatment, k=8, m=1), data= DOC\_T4, method= "REML")  
summary.gam(m4)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## log(DOC..mg.L) ~ Treatment + s(plant.mass..g, by = Treatment,   
## k = 8, m = 1)  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.21422 0.04373 50.638 <2e-16 \*\*\*  
## Treatmentunburned -0.05664 0.06184 -0.916 0.37   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Approximate significance of smooth terms:  
## edf Ref.df F p-value   
## s(plant.mass..g):Treatmentburned 3.520 7 4.857 5.92e-05 \*\*\*  
## s(plant.mass..g):Treatmentunburned 3.494 7 4.578 8.56e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## R-sq.(adj) = 0.694 Deviance explained = 77.9%  
## -REML = -1.0078 Scale est. = 0.02868 n = 30

gam.check(m4, rep=1000)



##   
## Method: REML Optimizer: outer newton  
## full convergence after 5 iterations.  
## Gradient range [-5.689671e-11,5.286438e-12]  
## (score -1.007834 & scale 0.02868005).  
## Hessian positive definite, eigenvalue range [1.231644,14.48156].  
## Model rank = 16 / 16   
##   
## Basis dimension (k) checking results. Low p-value (k-index<1) may  
## indicate that k is too low, especially if edf is close to k'.  
##   
## k' edf k-index p-value   
## s(plant.mass..g):Treatmentburned 7.00 3.52 0.72 0.045 \*  
## s(plant.mass..g):Treatmentunburned 7.00 3.49 0.72 0.040 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

draw(m4)

