Residence Time Experiment

Emmi Mueller

July 23, 2019

##Setup

##Set up figure theme

```
my.cols <- RColorBrewer::brewer.pal(n = 4, name = "Greys")[3:4]

# Set theme for figures in the paper
theme_set(theme_classic() +
  theme(axis.title = element_text(size = 16),
        axis.title.x = element_text(margin = margin(t = 15, b = 15)),
        axis.title.y = element_text(margin = margin(l = 15, r = 15)),
        axis.text = element_text(size = 14),
        axis.text.x = element_text(margin = margin(t = 5)),
        axis.text.y = element_text(margin = margin(r = 5)),
        #axis.line.x = element_line(linewidth = 1),
        #axis.line.y = element_line(linewidth = 1),
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.ticks.x = element_line(linewidth = 1),
        axis.ticks.y = element_line(linewidth = 1),
        axis.ticks.length = unit(.1, "in"),
        panel.border = element_rect(color = "black", fill = NA, linewidth = 1.5),
        legend.text = element_text(size = 10),
        strip.text = element_text(size = 14),
        strip.background = element_blank()
        ))
```

```
#Define Inputs

#Tau = general design file for experiment, paired Day 0 and 20
Tau <- read.csv("data/RTLC/2021_RTLC_Tau_Combined.csv", header = TRUE)
Tau_ctrl <- Tau[50:51,]
Tau <- Tau[1:49,]
Tau$Tau <- as.numeric(Tau$Tau)
Tau$Set <- as.character(Tau$Set)
Tau$Day_0_Seq[Tau$Day_0_Seq == ""] <- NA
Tau$Day_20_Seq[Tau$Day_20_Seq == ""] <- NA

#Tau_Seq = Full list of sequenced samples, non-paired Day 0 and 20
Tau_Seq <- read.csv("data/RTLC/2021_RTLC_Tau_Seq.csv", header = TRUE)

#Import Shared Files
#OTUs_B <- read.otu(shared = "mothur/RTLC.final.shared", cutoff = "0.03")
OTUs <- read.otu(shared = "mothur/RTLC_A+B/RTLC.final.shared", cutoff = "0.03")
```

```
rownames(OTUs)[rownames(OTUs) == "GSF3365_RTLC"] <- "GSF3365_RTLC_002"

#Import Taxonomy
#OTU.tax.B <- read.tax(taxonomy = "mothur/RTLC.final.taxonomy", format = "rdp")
OTU.tax <- read.tax(taxonomy = "mothur/RTLC_A+B/RTLC.final.taxonomy", format = "rdp")

#Abundance data
#Tau <- N_fxn(read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_N.csv", header = TRUE), Tau)
Tau <- N_fxn(read.csv("data/RTLC/RTLC_S2/20210628_RTLC_S2_N_M1gate.csv", header = TRUE), Tau)
Tau <- N_fxn(read.csv("data/RTLC/RTLC_S3/20210723_RTLC_S3_N_M1gate.csv", header = TRUE), Tau)
Tau <- N_fxn(read.csv("data/RTLC/RTLC_S4/20210904_RTLC_S4_N_M1gate.csv", header = TRUE), Tau)


#Biofilm data
Tau <- OT_fxn(read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_Otoole.csv", header = TRUE), Tau)
Tau <- OT_fxn(read.csv("data/RTLC/RTLC_S2/20210628_RTLC_S2_Otoole.csv", header = TRUE), Tau)
Tau <- OT_fxn(read.csv("data/RTLC/RTLC_S3/20210723_RTLC_S3_Otoole.csv", header = TRUE), Tau)
Tau <- OT_fxn(read.csv("data/RTLC/RTLC_S4/20210904_RTLC_S4_Otoole.csv", header = TRUE), Tau)


#BP data
Tau <- as.data.frame(BP_fxn(read.csv("data/RTLC/RTLC_S1/20210602_RTLC_S1_BP.csv", header = FALSE), Tau,
Tau <- as.data.frame(BP_fxn(read.csv("data/RTLC/RTLC_S2/20210628_RTLC_S2_BP.csv", header = FALSE), Tau,
Tau <- as.data.frame(BP_fxn(read.csv("data/RTLC/RTLC_S3/20210723_RTLC_S3_BP.csv", header = FALSE), Tau,
Tau <- as.data.frame(BP_fxn(read.csv("data/RTLC/RTLC_S4/20210904_RTLC_S4_BP.csv", header = FALSE), Tau,
Tau$ind_P <- Tau$uMChr/Tau$N


#Biolog data
Tau <- EP_fxn(read.csv("data/RTLC/eco.data_rt.txt", header = TRUE, sep = "\t"), Tau)

Tau$Set <- as.factor(Tau$Set)


Tau$Tau <- log(((10^Tau$Tau)/60), 10)

Tau$N_turn <- (20 * 24)/(10^Tau$Tau)

turnover <- log(20*24, 10)

Tau$Turn <- Tau$Tau <= log(24*20, 10)

Tau$Pump <- Tau$Tau < 1.6
```

##Read in all EcoPlate 48 hour data and create a resource by sample matrix (wbs - well response by site) and environmental matrix (env - Tau only)

```
EcoPlate <- read.csv("data/RTLC/eco.data_rt_S1_48.txt", header = TRUE, sep = "\t")
EcoPlate <- rbind(EcoPlate, read.csv("data/RTLC/eco.data_rt_S2_48.txt", header = TRUE, sep = "\t"))
EcoPlate <- rbind(EcoPlate, read.csv("data/RTLC/eco.data_rt_S3_48.txt", header = TRUE, sep = "\t"))
EcoPlate <- rbind(EcoPlate, read.csv("data/RTLC/eco.data_rt_S4_48.txt", header = TRUE, sep = "\t"))
EcoPlate_env <- subset(EcoPlate, select = c(Tau, Set))
rownames(EcoPlate) <- EcoPlate$Tau
EcoPlate <- subset(EcoPlate, select =-c(Tau, NumRes, Avg, Set, Hours))
```

##Load Resource Use data for first 72 hours

```
EcoPlate_72 <- read.csv("data/RTLC/eco.data_rt_S1_0.txt", header = TRUE, sep = "\t")
file_list <- c("data/RTLC/eco.data_rt_S1_24.txt", "data/RTLC/eco.data_rt_S1_48.txt", "data/RTLC/eco.data

for(file in file_list){
  EcoPlate_72 <- rbind(EcoPlate_72, read.csv(file, header = TRUE, sep = "\t"))
}

long_EP <- EcoPlate_72 %>% gather(C_Source, OD, X2.Hydroxy.Benzoic.Acid:Avg)
```

##Load slopes of resource use data (48-24 hours)

```
EcoPlate_S1 <- read.csv("data/RTLC/eco.data_rt_S1_48.txt", header = TRUE, sep = "\t")
EcoPlate_S1 <- cbind(EcoPlate_S1[,1:2], (EcoPlate_S1[,4:34] - read.csv("data/RTLC/eco.data_rt_S1_24.txt
EcoPlate_S1[,4:33] <- EcoPlate_S1[,4:33]/48

EcoPlate_S2 <- read.csv("data/RTLC/eco.data_rt_S2_48.txt", header = TRUE, sep = "\t")
EcoPlate_S2 <- cbind(EcoPlate_S2[,1:2], (EcoPlate_S2[,4:34] - read.csv("data/RTLC/eco.data_rt_S2_24.txt
EcoPlate_S2[,4:33] <- EcoPlate_S2[,4:33]/48

EcoPlate_S3 <- read.csv("data/RTLC/eco.data_rt_S3_48.txt", header = TRUE, sep = "\t")
EcoPlate_S3 <- cbind(EcoPlate_S3[,1:2], (EcoPlate_S3[,4:34] - read.csv("data/RTLC/eco.data_rt_S3_24.txt
EcoPlate_S3[,4:33] <- EcoPlate_S3[,4:33]/48

EcoPlate_S4 <- read.csv("data/RTLC/eco.data_rt_S4_48.txt", header = TRUE, sep = "\t")
EcoPlate_S4 <- cbind(EcoPlate_S4[,1:2], (EcoPlate_S4[,4:34] - read.csv("data/RTLC/eco.data_rt_S4_24.txt
EcoPlate_S4[,4:33] <- EcoPlate_S4[,4:33]/48

EcoPlate_Slope <- rbind(EcoPlate_S1, EcoPlate_S2, EcoPlate_S3, EcoPlate_S4)

ResType <- read.csv("code/resource_type.txt", header = TRUE, sep = "\t")

colnames(EcoPlate_Slope) <- c("Tau", "Set", "2-Hydroxy.Benzoic.Acid", "4-Hydroxy.Benzoic.Acid", "alpha-


long_EP_slope <- EcoPlate_Slope %>% gather(C_Source, Slope, "2-Hydroxy.Benzoic.Acid":"Tween.80")

ResType <- distinct(ResType)

long_EP_slope$Type <- NA

for(row in rownames(long_EP_slope)){
  long_EP_slope[row,"Type"] <- ResType$Type[ResType$Resource == long_EP_slope[row,"C_Source"]]
}
```

##Clean and rarefy the OTU table

```
#Remove samples with fewer than 10000 reads
coverage <- rowSums(OTUs)
cutoff <- 10000
lows <- which(coverage < cutoff)
if (length(lows) > 0){
  OTUs <- OTUs[-which(coverage < cutoff),]
}
```

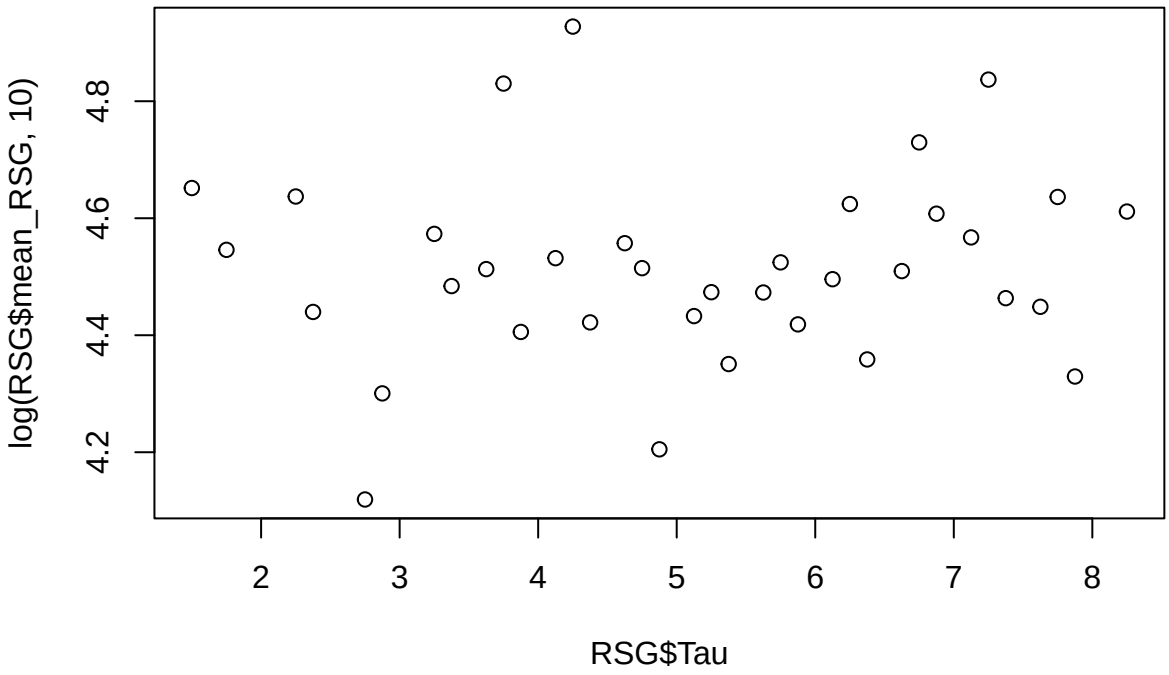
```r
#Remove OTUs with less thatn 2 reads across all samples
OTUs <- OTUs[,which(colSums(OTUs) > 2)]

#Generate rarefacation plot
otu.min <- min(rowSums(OTUs))

# png("./output/OTU_rarefaction.png", width = 800, height = 480)
# par(mar = c(1, 1, 1, 1) + 0.1)
#   rarecurve(x = OTUs, step = 100, sample = otu.min, col = "blue", cex = 0.6, las = 1)
# #  abline(0,1, col = "red")
# #  text(1500, 1500, "1:1", pos = 2, col = "red")
# dev.off()

#rarefy OTUs to the sample with the lowest number of reads then remove OTUs with 0 reads after rarefact
set.seed(47405)
OTU_r <- rrarefy(OTUs, otu.min)
OTU_r <- OTU_r[,-which(colSums(OTU_r) == 0)]
rm(OTUs)

Tau_20 <- subset(Tau[is.na(Tau$Day_20_Seq) == 0,])
Tau_Seq$Day <- as.factor(Tau_Seq$Day)
Tau_Seq$Tau <- as.numeric(Tau_Seq$Tau)
```

```
## Warning: NAs introduced by coercion
```

```r
#Generate OTU tables with p/a, relative abundance
OTUsREL <-decostand(OTU_r, method = "total")
OTUsREL_20 <- OTUsREL[rownames(OTUsREL) %in% unique(Tau$Day_20_Seq),]
OTUsREL_20 <- OTUsREL_20[,which(colSums(OTUsREL_20) > 0)]
rownames(OTUsREL_20) <- subset(Tau_Seq, Day == 20)$Tau
#
OTUs.PA <- decostand(OTU_r, method = "pa")
# OTUsPA_20<- OTUs.PA[rownames(OTUs.PA) %in% unique(Tau$Day_20_Seq),]
# OTUsPA_20 <- OTUsPA_20[,which(colSums(OTUsPA_20) > 0)]
# rownames(OTUsPA_20) <- subset(Tau_Seq, Day == 20)$Tau


OTUsr_20 <- OTU_r[rownames(OTU_r) %in% unique(Tau$Day_20_Seq),]
OTUsr_20 <- OTUsr_20[,which(colSums(OTUsr_20) >0)]
rownames(OTUsr_20) <- subset(Tau_Seq, Day == 20)$Tau

OTU.tax.20 <- OTU.tax[OTU.tax$OTU %in% unique(colnames(OTUsr_20)),]

Tau_Seq$ACE <- S.ace(OTU_r)

#
S <- as.data.frame(cbind(row.names(OTU_r), unname(S.cal(OTU_r))))
Tau <- S_fxn(S, Tau)
Tau$Day_20.S <- as.numeric(Tau$Day_20.S)
Tau$Day_0.S <- as.numeric(Tau$Day_0.S)

SimpE <- as.data.frame(cbind(row.names(OTU_r), unname(SimpE.cal(OTU_r))))
Tau <- SimpE_fxn(SimpE, Tau)
Tau$Day_0.SimpE <- as.numeric(Tau$Day_0.SimpE)
```

```r
Tau$Day_20.SimpE <- as.numeric(Tau$Day_20.SimpE)
```

```r
RSG <- read.csv("./data/RTLC/RTLC_meanRSG.csv", header = TRUE)
RSG$mean_RSG <- as.numeric(RSG$mean_RSG)

plot(log(RSG$mean_RSG, 10) ~ RSG$Tau)
```



#Figure 1. Community Structure

##Cstat - Microbial abundance

```r
# test <- as.data.frame(seq(0, 0.1, 0.0001))
# colnames(test) <- "n"
# for(i in seq(0, 0.1, 0.0001)){
#   N_gam_re <- gam(log_N ~ s(Tau, sp = i) + s(Set, bs = "re"), family = gaussian(link = "identity"), d
#   test[test$n == i, "AIC"] <- N_gam_re$aic
#   test[test$n == i, "REML"] <- N_gam_re$gcv.ubre[[1]]
# }
#
# plot(test$AIC ~ test$n)
# plot(test$REML ~ test$n, ylim = c(0,30))

N_gam <- gam(log_N ~ s(Tau), family = gaussian(link = "identity"), data = Tau, method = "REML")
N_gam_re <- gam(log_N ~ s(Tau) + s(Set, bs = "re"), family = gaussian(link = "identity"), data = Tau, m

AIC(N_gam, N_gam_re)
```

```
##                df      AIC
## N_gam    6.447302 33.06401
## N_gam_re 9.736133 16.89868
```

```
anova(N_gam, N_gam_re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: log_N ~ s(Tau)
## Model 2: log_N ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
## 1    29.684     3.6910
## 2    25.998     1.9624 3.6865   1.7287 3.332e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

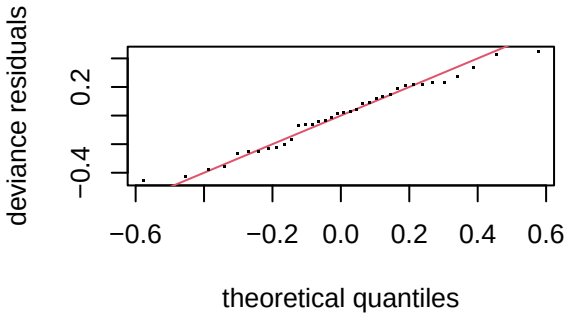
```
summary(N_gam_re)
```

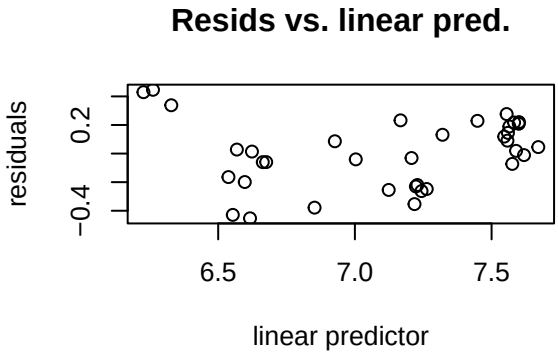
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log_N ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.115      0.153    46.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F  p-value
## s(Tau) 4.693  5.771 17.04  < 2e-16 ***
## s(Set) 1.777  2.000  9.70 0.000235 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.759   Deviance explained = 80.3%
## -REML = 13.189  Scale est. = 0.068783  n = 36
```

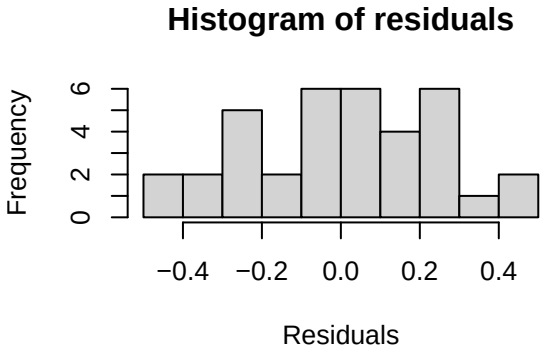
```
k.check(N_gam_re)
```

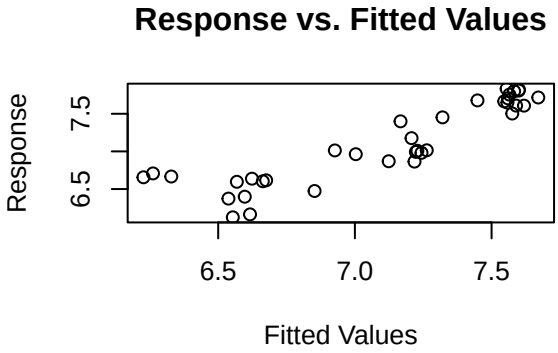
```
##        k'      edf  k-index p-value
## s(Tau)  9 4.693224 1.276555    0.95
## s(Set)  3 1.776724       NA      NA
```

```
gam.check(N_gam_re)
```










```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-6.707242e-06,7.847398e-07]
## (score 13.18909 & scale 0.06878258).
## Hessian positive definite, eigenvalue range [0.4752446,17.25415].
## Model rank =  13 / 13
##
## 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(Tau) 9.00 4.69    1.28    0.94
## s(Set) 3.00 1.78      NA      NA
```

```
mean(summary(N_gam_re)$s.table[,4])
```

```
## [1] 0.0001177233
```

```
gam.vcomp(N_gam_re)
```

```
##
## Standard deviations and 0.95 confidence intervals:
##
##          std.dev      lower     upper
## s(Tau) 0.4665748 0.13525409 1.6095044
## s(Set) 0.2536385 0.08568036 0.7508427
## scale  0.2622643 0.19582227 0.3512499
```

```
##
## Rank: 3/3
```

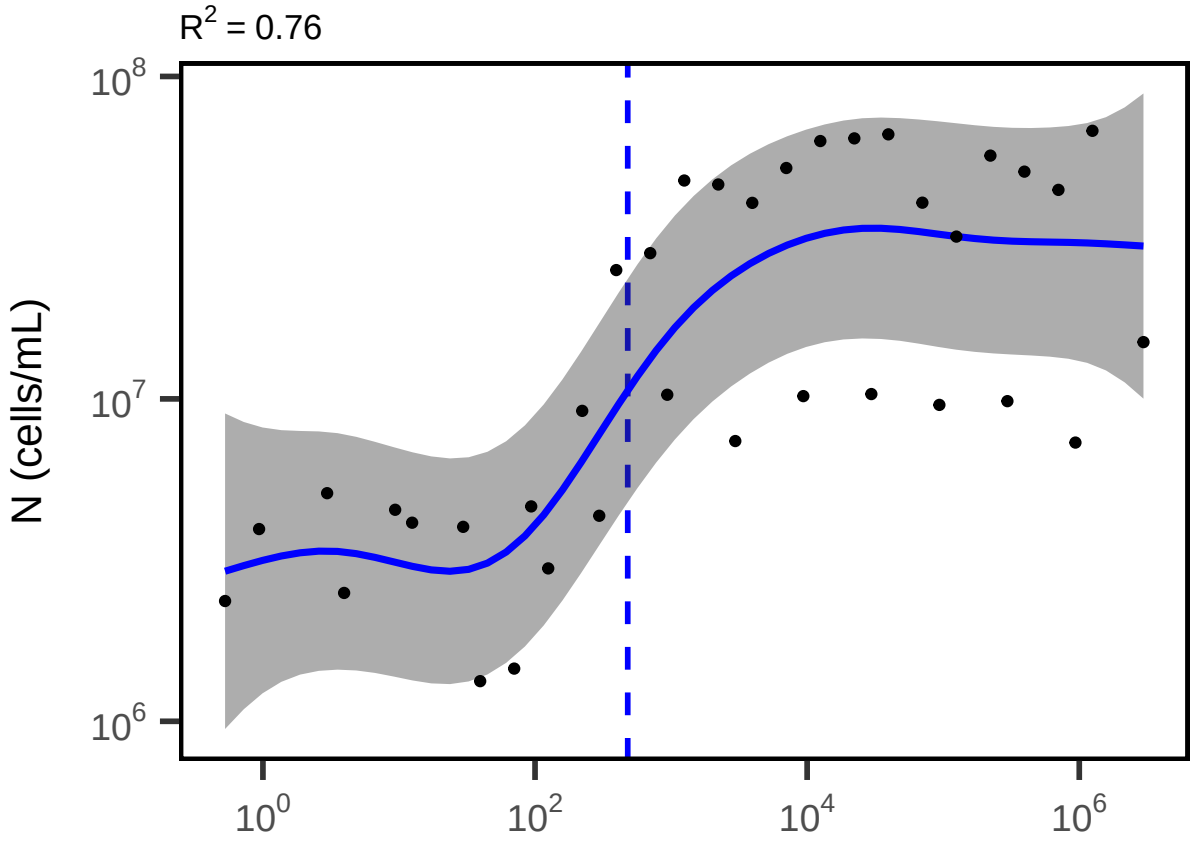
```
N_Tau <- predict_gam(N_gam_re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit))+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = Tau, aes(x = Tau, y = log_N))+
  theme(axis.title.x = element_blank())+
  ylab("N (cells/mL)")+
  scale_x_continuous(labels = label_math(10^.x))+
  scale_y_continuous(breaks = c(6, 7, 8), labels = label_math(10^.x))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(N_gam_re)$r.sq, 2))))
```

```
## Warning in mgcv::predict.gam(model, new_data, exclude = exclude_terms, se.fit =
## TRUE): factor levels 1 not in original fit
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
N_Tau
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```



```
ggsave("./output/RTLC_N.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/RTLC_N.png")
```

```
## Saving 6.5 x 4.5 in image
```

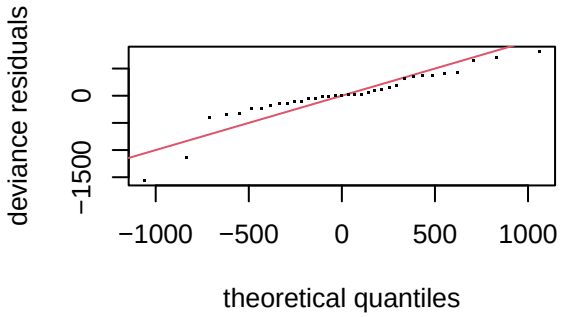
```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

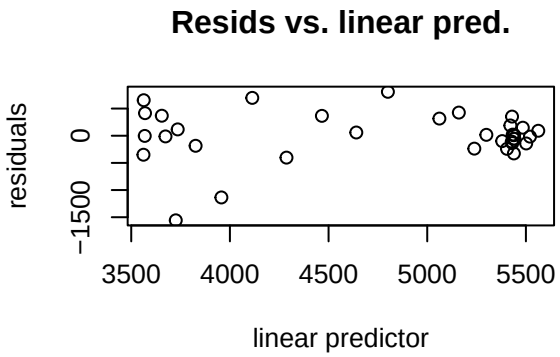
##Cstat - Observed Richness

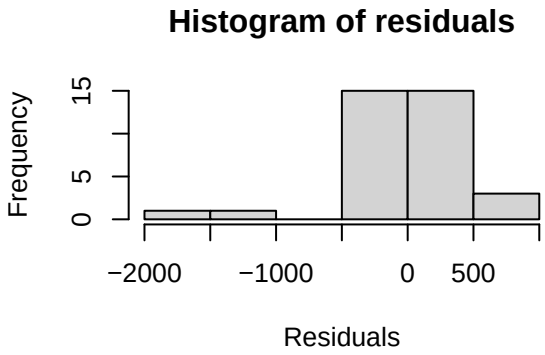
```
# test <- as.data.frame(seq(0, 0.1, 0.0001))
# colnames(test) <- "n"
# for(i in seq(0, 0.1, 0.0001)){
#   S_gam_re <- gam(log(Day_20.S, 10) ~ s(Tau, sp = i) + s(Set, bs = "re"), family = gaussian(link = "i
#   test[test$n == i, "AIC"] <- S_gam_re$aic
#   test[test$n == i, "REML"] <- S_gam_re$gcv.ubre[[1]]
# }
#
# plot(test$AIC ~ test$n)
# plot(test$REML ~ test$n, ylim = c(-50,-20))

S_gam <- gam(Day_20.S ~ s(Tau, k = 14), family = gaussian(link = "identity"), data = Tau, method = "REM
S_gam_re <- gam(Day_20.S ~ s(Tau)  + s(Set, bs = "re"), family = gaussian(link = "identity"), data = Tau

k.check(S_gam)
```

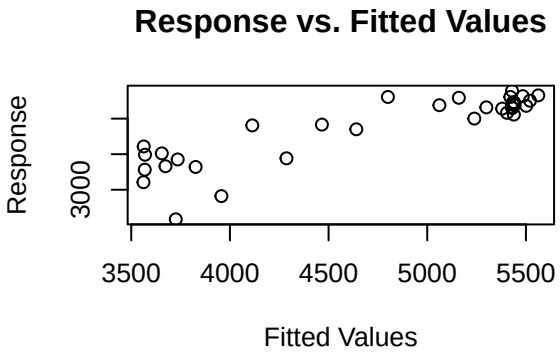
```
##        k'      edf  k-index p-value
## s(Tau) 13 4.942489 1.158869   0.805
```

```
gam.check(S_gam)
```










```
##
## Method: REML   Optimizer: outer newton
## full convergence after 4 iterations.
## Gradient range [-0.0001617916,1.904589e-05]
## (score 259.3033 & scale 234590.3).
## Hessian positive definite, eigenvalue range [0.6175485,16.74482].
## Model rank =  14 / 14
##
## 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(Tau) 13.00  4.94    1.16     0.8
```

```
AIC(S_gam, S_gam_re)
```

```
##                 df      AIC
## S_gam     8.101273 541.8116
## S_gam_re 10.954306 535.4213
```

```
anova(S_gam, S_gam_re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Day_20.S ~ s(Tau, k = 14)
## Model 2: Day_20.S ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev     Df Deviance Pr(>Chi)
## 1    26.740    6816611
```

```
## 2    23.449    4824718 3.2906  1991893  0.01517 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

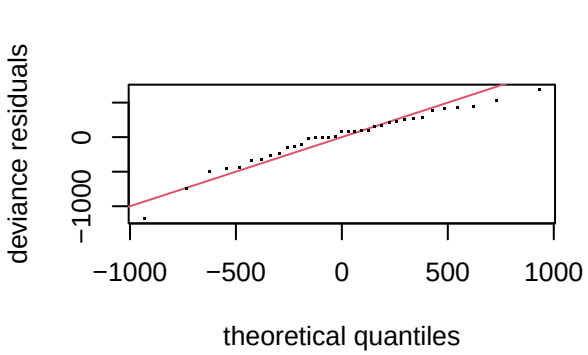
```
summary(S_gam_re)
```

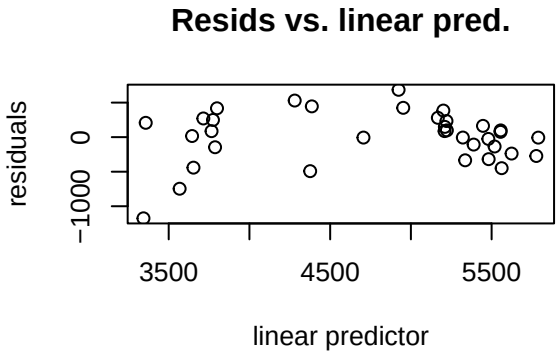
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Day_20.S ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4790.2      161.7   29.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df      F p-value
## s(Tau) 5.064  6.138 18.225  <2e-16 ***
## s(Set) 2.294  3.000  2.947  0.0186 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.792   Deviance explained = 83.7%
## -REML = 257.71  Scale est. = 1.8109e+05  n = 35
```

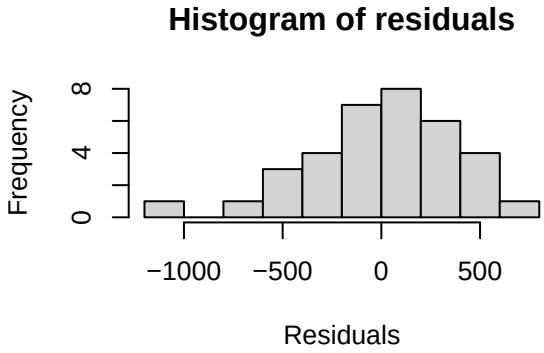
```
k.check(S_gam_re)
```

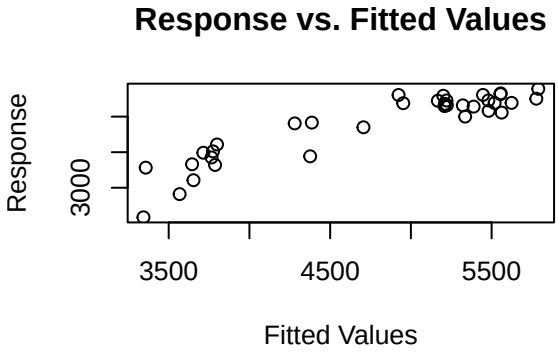
```
##        k'      edf  k-index p-value
## s(Tau)  9 5.063830 1.204698   0.855
## s(Set)  4 2.294089       NA      NA
```

```
gam.check(S_gam_re)
```










```
##
## Method: REML   Optimizer: outer newton
## full convergence after 4 iterations.
## Gradient range [-0.0001428268,1.919717e-05]
## (score 257.706 & scale 181093.9).
## Hessian positive definite, eigenvalue range [0.5611962,16.84203].
## Model rank =  14 / 14
##
## 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(Tau) 9.00 5.06     1.2    0.85
## s(Set) 4.00 2.29      NA      NA
```

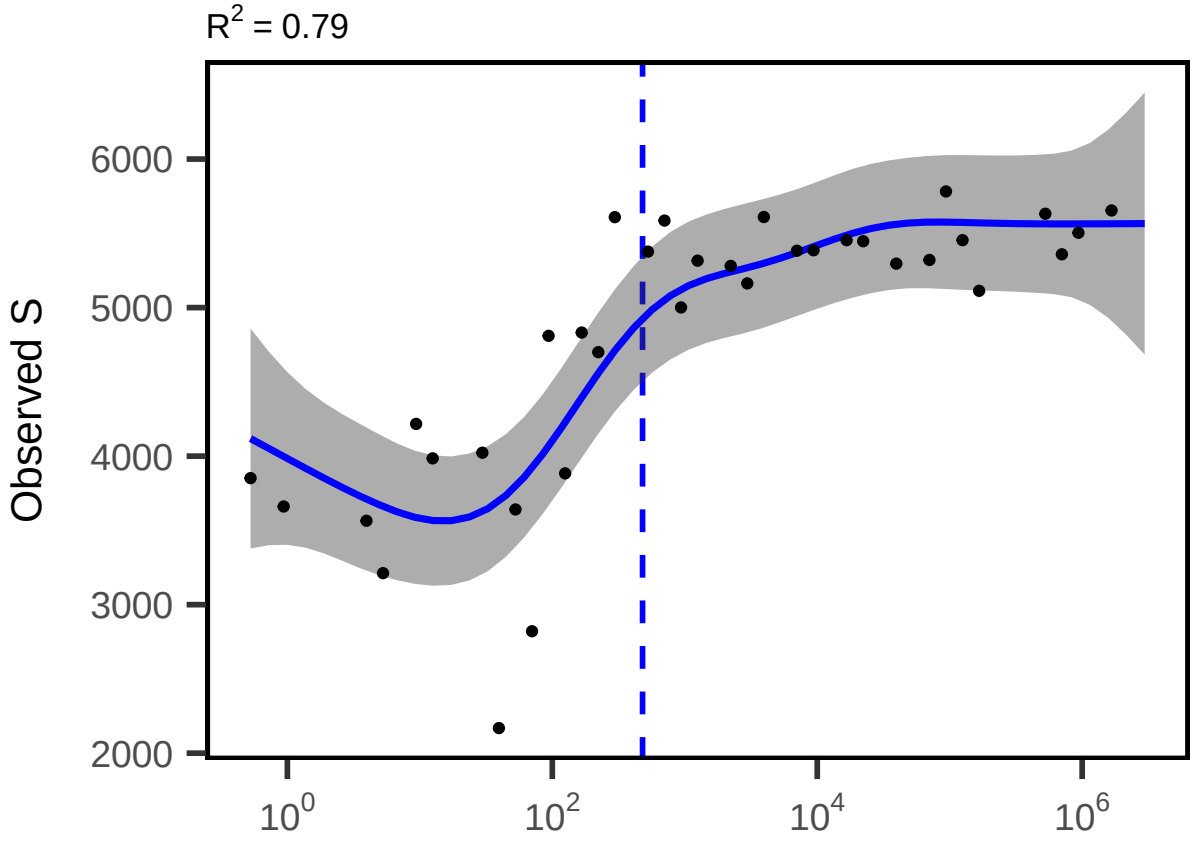
```
mean(summary(S_gam_re)$s.table[,4])
```

```
## [1] 0.009281618
```

```
S.ob_Tau <- predict_gam(S_gam_re, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit))+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = Tau, aes(x = Tau, y = Day_20.S))+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
#  scale_y_continuous(labels = label_math(expr = 10^.x, format = force))+
```

```
  ylab("Observed S")+
  theme(axis.title.x = element_blank())+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(S_gam_re)$r.sq, 2))))


S.ob_Tau
```

```
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/RTLC_S.pdf")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/RTLC_S.png")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```

#Cstat - Simpson's Evenness (E[1/D])

```
# test <- as.data.frame(seq(0, 0.1, 0.0001))
# colnames(test) <- "n"
# for(i in seq(0, 0.1, 0.0001)){
#   SimpE_gam <- gam(Day_20.SimpE ~ s(Tau, sp = i), family = gaussian(link = "identity"), data = Tau, m
#   test[test$n == i, "AIC"] <- SimpE_gam$aic
#   test[test$n == i, "REML"] <- SimpE_gam$gcv.ubre[[1]]
```

```r
# }
#
# plot(test$AIC ~ test$n)
# plot(test$REML ~ test$n)

SimpE_gam <- gam(Day_20.SimpE ~ s(Tau, k = 15), family = gaussian(link = "identity"), data = Tau, metho
SimpE_gam_re <- gam(Day_20.SimpE ~ s(Tau) + s(Set, bs = "re"), family = gaussian(link = "identity"), da

AIC(SimpE_gam, SimpE_gam_re)
```

```
##                     df       AIC
## SimpE_gam    8.147650 -195.5583
## SimpE_gam_re 7.715705 -195.1028
```

```r
anova(SimpE_gam, SimpE_gam_re, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Day_20.SimpE ~ s(Tau, k = 15)
## Model 2: Day_20.SimpE ~ s(Tau) + s(Set, bs = "re")
##   Resid. Df Resid. Dev       Df    Deviance Pr(>Chi)
## 1    26.677  0.0048176
## 2    27.253  0.0050026 -0.57592 -0.00018506   0.1612
```

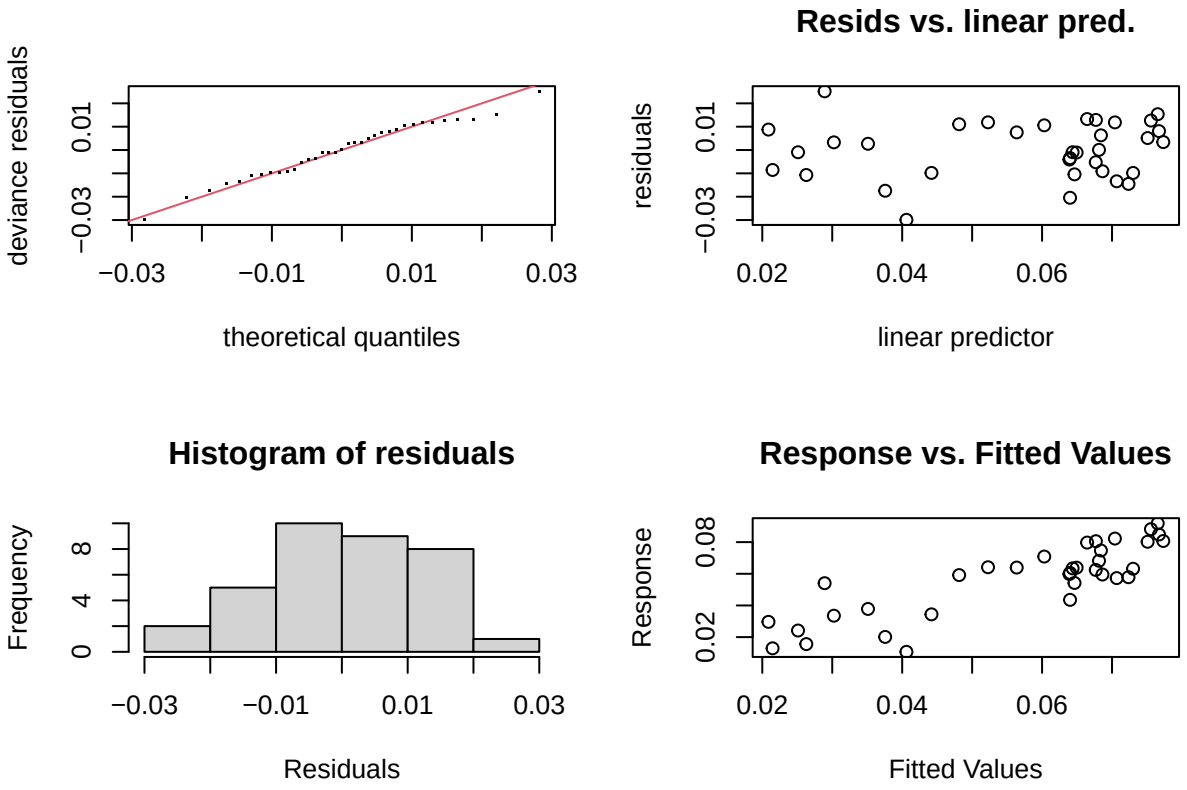
```r
summary(SimpE_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Day_20.SimpE ~ s(Tau, k = 15)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.056801   0.002178   26.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F  p-value
## s(Tau) 4.972  6.148 11.51 1.83e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.674   Deviance explained = 72.2%
## -REML = -88.306  Scale est. = 0.00016596  n = 35
```

```r
k.check(SimpE_gam)
```

```
##        k'      edf  k-index p-value
## s(Tau) 14 4.972393 0.963992     0.4
```

```
gam.check(SimpE_gam)
```




```
##
## Method: REML   Optimizer: outer newton
## full convergence after 7 iterations.
## Gradient range [-1.075916e-08,-8.738219e-09]
## (score -88.30582 & scale 0.0001659648).
## Hessian positive definite, eigenvalue range [0.7307918,16.75017].
## Model rank =  15 / 15
##
## 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(Tau) 14.00  4.97    0.96    0.32
```

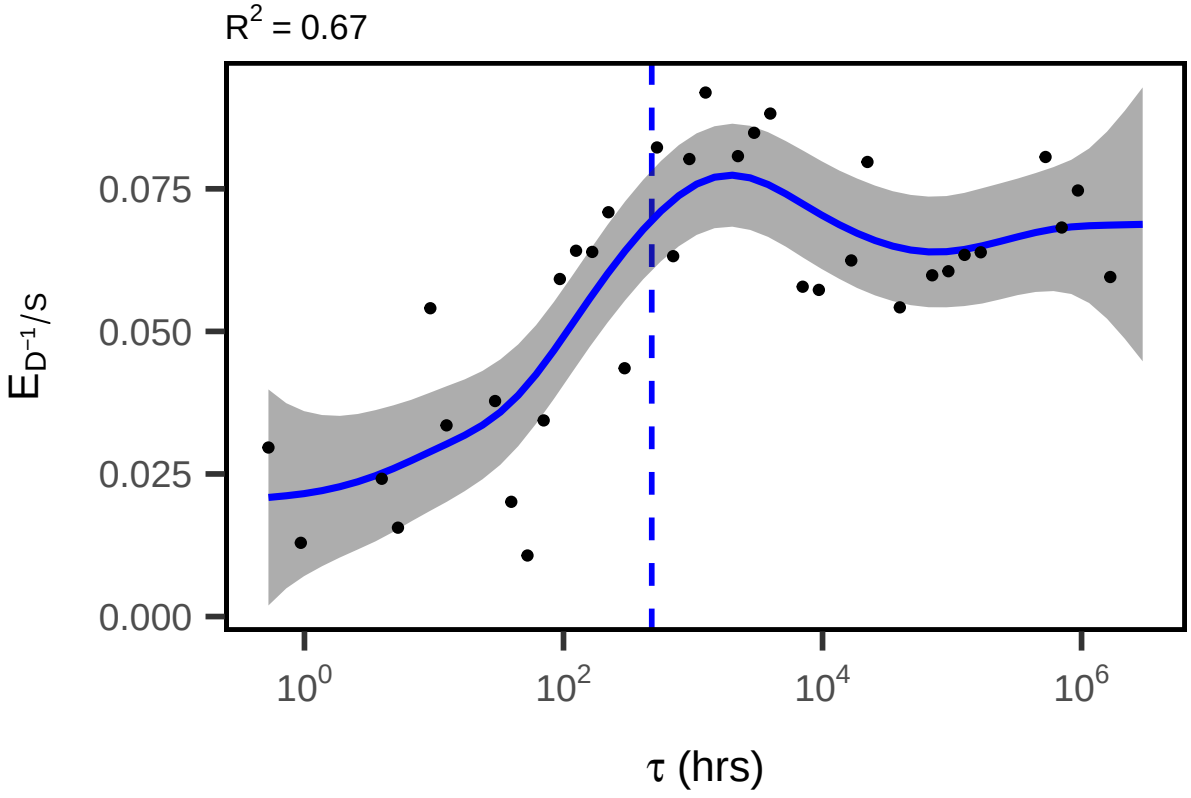
```
mean(summary(SimpE_gam)$s.table[,4])
```

```
## [1] 1.832464e-06
```

```
SimpE_Tau <- predict_gam(SimpE_gam) %>%
  ggplot(aes(Tau, fit))+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = Tau, aes(x = Tau, y = Day_20.SimpE))+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  ylab(expression("E"[D^"-1"/S]))+
```

```
  xlab(expression(paste(tau, " (hrs)")))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(SimpE_gam)$r.sq, 2))))

SimpE_Tau
```

```
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/RTLC_SimpE.pdf")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/RTLC_SimpE.png")
```

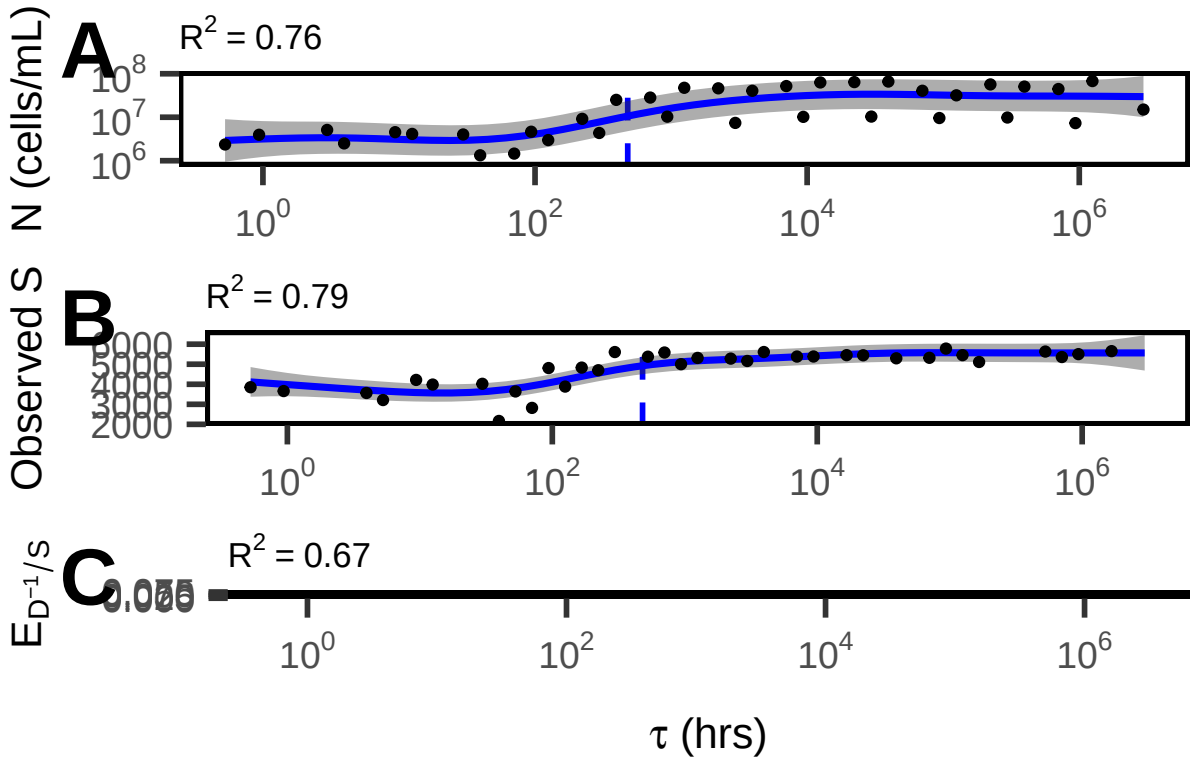
```
## Saving 6.5 x 4.5 in image
## Warning: Removed 14 rows containing missing values (`geom_point()`).
```

##Draw Figure

```
ggdraw()+
  draw_plot(N_Tau, x = 0, y = 0.65, width = 1, height = 0.3) +
  draw_plot(S.ob_Tau, x = 0, y = 0.35, width = 1, height = 0.3) +
  draw_plot(SimpE_Tau, x = 0, y = 0.0, width = 1, height = 0.35)+
  draw_plot_label(label = c("A", "B", "C"), size = 30, x = c(0.06,0.06,0.06), y = c(0.96,0.65,0.35))+
  theme(plot.background = element_rect(fill="white", color = NA))
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 14 rows containing missing values (`geom_point()`).
## Removed 14 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/CommStr_Cstat.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/CommStr_Cstat.png", width = 5, height = 10)
```

#Figure 2. Community composition

```
tau.db <- vegdist(OTUsr_20, method = "bray", upper = TRUE, diag = TRUE)

tau.pcoa <- cmdscale(tau.db, eig = TRUE, k = 3)
explainvar1 <- round(tau.pcoa$eig[1] / sum(tau.pcoa$eig), 3) *100
explainvar2 <- round(tau.pcoa$eig[2] / sum(tau.pcoa$eig), 3) *100
explainvar3 <- round(tau.pcoa$eig[3] / sum(tau.pcoa$eig), 3) *100
sum.eig <- sum(explainvar3, explainvar2, explainvar1)

tau.plot <- as.data.frame(tau.pcoa$points)
tau.plot <- cbind(tau.plot, Tau_20)
tau.plot$Pump <- factor(tau.plot$Pump, levels=c("TRUE", "FALSE"))
tau.plot$Turn <- factor(tau.plot$Turn, levels=c("TRUE", "FALSE"))

spe.corr <- add.spec.scores(tau.pcoa, OTUsREL_20, method = "cor.scores")$cproj
corrcut <- 0.7
imp.spp <- as.data.frame(spe.corr[abs(spe.corr[,1]) >= corrcut | abs(spe.corr[,2]) >= corrcut,])
imp.spp <- na.omit(imp.spp)
```

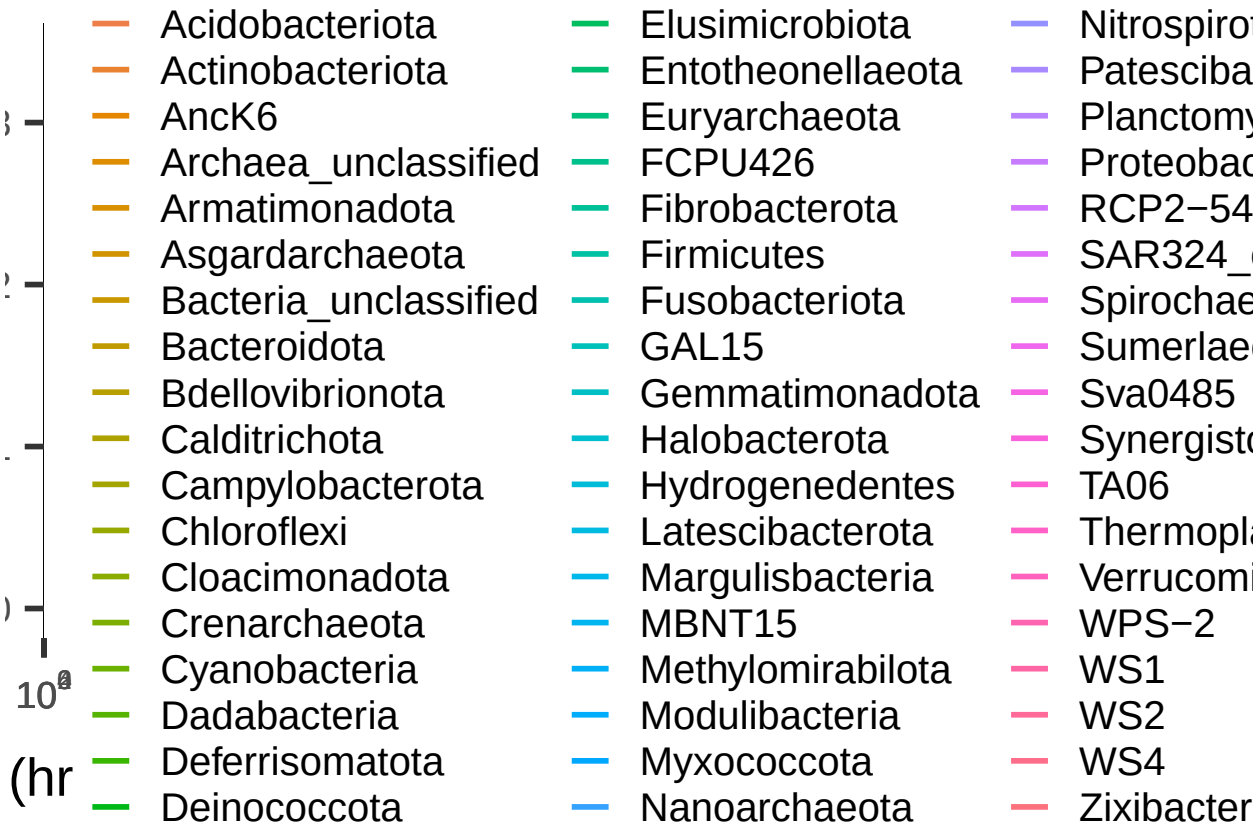
```
imp.tax <- subset(OTU.tax, OTU == rownames(imp.spp)[1])

x <- 2
while (x <= nrow(imp.spp)){
  imp.tax <- rbind(imp.tax, subset(OTU.tax, OTU == rownames(imp.spp)[x]))
  x <- x + 1
}

imp.spp <- cbind(imp.spp, imp.tax)

imp.otus <- as.data.frame(row.names(OTUsREL_20))
imp.spp.otu <- c()
for(otu in unique(imp.spp$OTU)){
    imp.otus <- cbind(imp.otus, OTUsREL_20[,otu])
    imp.spp.otu <- c(imp.spp.otu, otu)
}
colnames(imp.otus) <- c("Tau", imp.spp.otu)
imp.otus$Tau <- log(((10^as.numeric(imp.otus$Tau))/60), 10)
```

```
Dim1_sites <- tau.plot$V1
Dim2_sites <- tau.plot$V2

DNA_corr <- as.data.frame(matrix(NA, dim(OTUsREL_20)[2], 4))
row.names(DNA_corr) <- colnames(OTUsREL_20)
colnames(DNA_corr)  <- c("V1_rho", "V1_p.value", "V2_rho", "V2_p.value")

for (i in 1:dim(OTUsREL_20)[2]){
  out.i <- cor.test(OTUsREL_20[,i], tau.plot$V1, method="spearman",
                    exact=FALSE)
  DNA_corr[i,1] <- out.i$estimate
  DNA_corr[i,2] <- out.i$p.value

  out.i <- cor.test(OTUsREL_20[,i], tau.plot$V2, method="spearman",
                    exact=FALSE)
  DNA_corr[i,3] <- out.i$estimate
  DNA_corr[i,4] <- out.i$p.value
}

DNA_corr     <- na.omit(DNA_corr)
DNA_corr$V1_BH <- p.adjust(DNA_corr$V1_p.value, method = "BH")
DNA_corr$V2_BH <- p.adjust(DNA_corr$V2_p.value, method = "BH")
DNA_corr_V2 <- DNA_corr[abs(DNA_corr$V2_rho) > 0.7, ]
DNA_corr_V1 <- DNA_corr[abs(DNA_corr$V1_rho) > 0.7, ]

V1.tax <- subset(OTU.tax, OTU == rownames(DNA_corr_V1)[1])
V2.tax <- subset(OTU.tax, OTU == rownames(DNA_corr_V2)[1])

x <- 2
while (x <= nrow(DNA_corr_V1)){
  V1.tax <- rbind(V1.tax, subset(OTU.tax, OTU == rownames(DNA_corr_V1)[x]))
  x <- x + 1
}
DNA_corr_V1 <- cbind(DNA_corr_V1, V1.tax)
```

```
x <- 2
while (x <= nrow(DNA_corr_V2)){
  V2.tax <- rbind(V2.tax, subset(OTU.tax, OTU == rownames(DNA_corr_V2)[x]))
  x <- x + 1
}

DNA_corr_V2 <- cbind(DNA_corr_V2, V2.tax)

DNA_corr_V2_neg <- subset(DNA_corr_V2, DNA_corr_V2$V2_rho < -0.7)
DNA_corr_V2_neg <- DNA_corr_V2_neg[,c("OTU","V2_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V2_neg, "./data/DNA_corr_V2_neg.csv")
DNA_corr_V2_pos <- subset(DNA_corr_V2, DNA_corr_V2$V2_rho > 0.7)
DNA_corr_V2_pos <- DNA_corr_V2_pos[,c("OTU","V2_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V2_pos, "./data/DNA_corr_V2_pos.csv")

DNA_corr_V1_neg <- subset(DNA_corr_V1, DNA_corr_V1$V1_rho < -0.7)
DNA_corr_V1_neg <- DNA_corr_V1_neg[,c("OTU","V1_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V1_neg, "./data/DNA_corr_V1_neg.csv")
DNA_corr_V1_pos <- subset(DNA_corr_V1, DNA_corr_V1$V1_rho > 0.7)
DNA_corr_V1_pos <- DNA_corr_V1_pos[,c("OTU","V1_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V1_pos, "./data/DNA_corr_V1_pos.csv")

DNA_V2_family_neg <- data.frame()
for(x in unique(DNA_corr_V2_neg$Family)){
  DNA_V2_family_neg <- rbind(DNA_V2_family_neg, c(x, mean(subset(DNA_corr_V2_neg, DNA_corr_V2_neg$Family
}
colnames(DNA_V2_family_neg) <- c("Family", "rho", "n")
DNA_V2_family_neg$n <- as.numeric(DNA_V2_family_neg$n)
DNA_V2_family_neg$rho <- as.numeric(DNA_V2_family_neg$rho)

DNA_V2_family_pos <- data.frame()
for(x in unique(DNA_corr_V2_pos$Family)){
  DNA_V2_family_pos <- rbind(DNA_V2_family_pos, c(x, mean(subset(DNA_corr_V2_pos, DNA_corr_V2_pos$Family
}
colnames(DNA_V2_family_pos) <- c("Family", "rho", "n")
DNA_V2_family_pos$n <- as.numeric(DNA_V2_family_pos$n)
DNA_V2_family_pos$rho <- as.numeric(DNA_V2_family_pos$rho)

DNA_V1_family_neg <- data.frame()
for(x in unique(DNA_corr_V1_neg$Family)){
  DNA_V1_family_neg <- rbind(DNA_V1_family_neg, c(x, mean(subset(DNA_corr_V1_neg, DNA_corr_V1_neg$Family
}
colnames(DNA_V1_family_neg) <- c("Family", "rho", "n")
DNA_V1_family_neg$n <- as.numeric(DNA_V1_family_neg$n)
DNA_V1_family_neg$rho <- as.numeric(DNA_V1_family_neg$rho)

DNA_V1_family_pos <- data.frame()
for(x in unique(DNA_corr_V1_pos$Family)){
  DNA_V1_family_pos <- rbind(DNA_V1_family_pos, c(x, mean(subset(DNA_corr_V1_pos, DNA_corr_V1_pos$Family
}
colnames(DNA_V1_family_pos) <- c("Family", "rho", "n")
DNA_V1_family_pos$n <- as.numeric(DNA_V1_family_pos$n)
DNA_V1_family_pos$rho <- as.numeric(DNA_V1_family_pos$rho)
```

```
# taxa <- as.data.frame(imp.otus$Tau)
# for(x in unique(OTU.tax.20$Phylum)){
#   list <- data.frame(rep(0, 35))
#   colnames(list) <- c(as.character(x))
#   for(y in OTU.tax.20$OTU){
#     if(subset(OTU.tax.20, OTU.tax.20$OTU == y)$Phylum == x){
#       list[,x] <- list[,x] + as.data.frame(OTUsr_20[,y])
#     }
#   }
#   taxa[,as.character(x)] <- list[,x]
# }
# rownames(taxa) <- taxa$`imp.otus$Tau`
# taxa.20 <- taxa[,-1]
# write.csv(taxa.20, "./data/taxa.20.csv")

taxa.20 <- read.csv("./data/taxa.20.csv", header = TRUE)
rownames(taxa.20) <- taxa.20$X
taxa.20 <- taxa.20[,-1]
taxa.20.rel <- decostand(taxa.20, method = "total")
taxa.20.rel <- cbind(imp.otus$Tau, taxa.20.rel)
colnames(taxa.20.rel) <- c("Tau", unique(OTU.tax.20$Phylum))
taxa.20.long <- gather(taxa.20.rel, phylum, REL, Proteobacteria:Modulibacteria)

phylum_plot <- ggplot(taxa.20.long, aes(x = Tau, y = REL, group = phylum))+
  geom_smooth(method = "loess", se = F, aes(color = phylum, group = phylum))+
  ylab("Relative abundance")+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  labs(color = "Phylum")+
  xlab(expression(paste(tau, " (hrs)")))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 15), axis.title = ele
  guides(color = guide_legend(ncol = 3))
phylum_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggsave("./output/RTLC_rel_phylum.pdf")
```

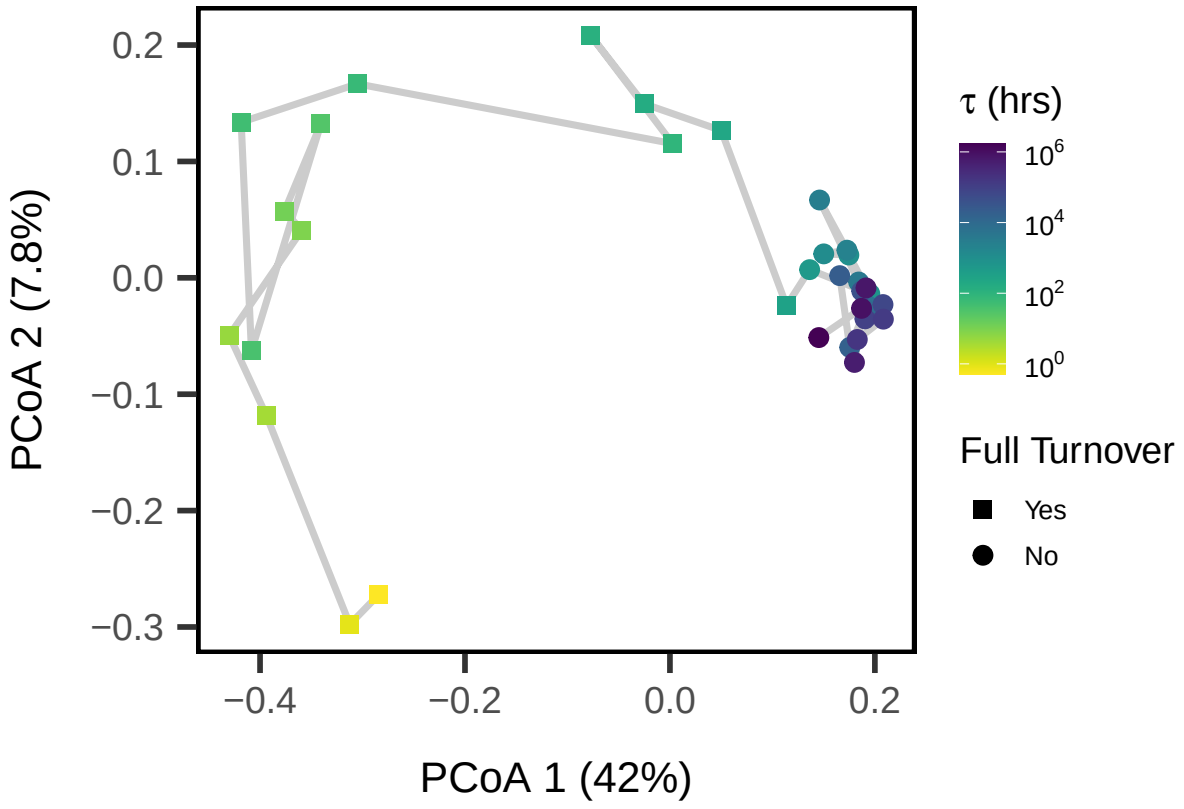
```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

```
ggsave("./output/RTLC_rel_phylum.png", width = 12, height = 8)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

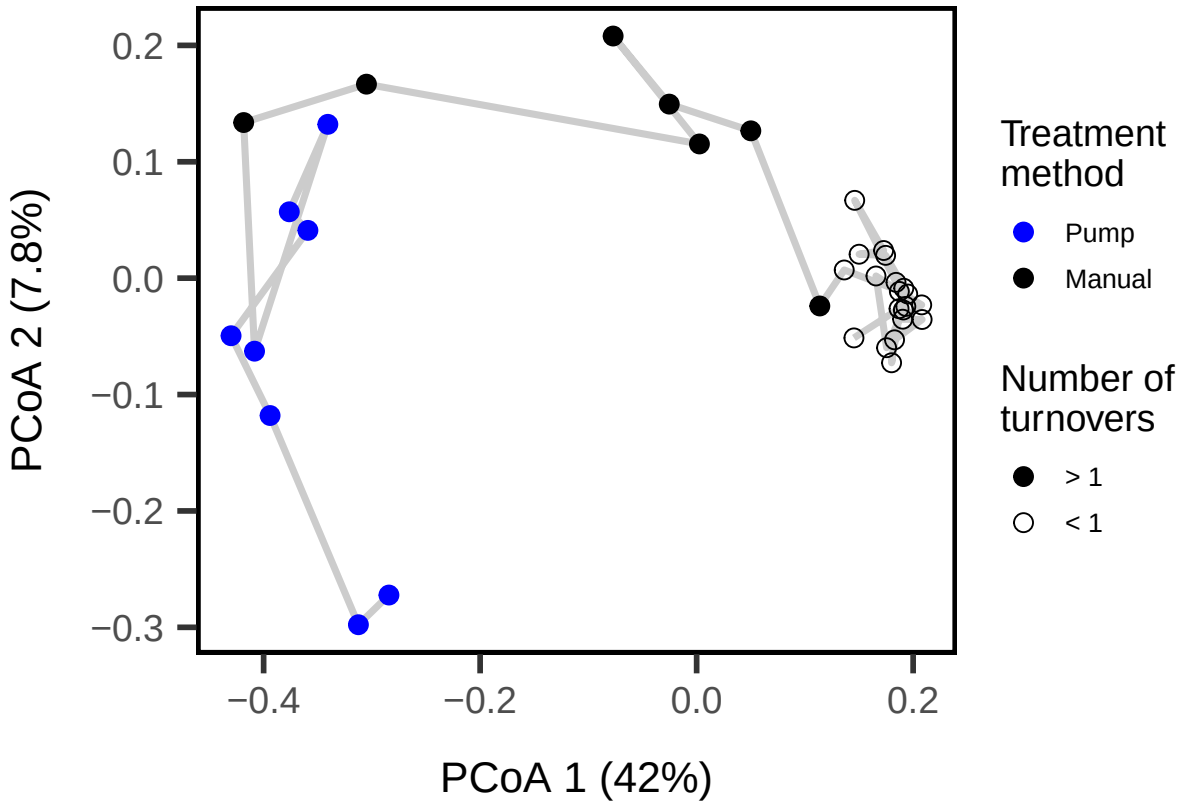
```
Dim1 <- c()
Dim2 <- c()
Dim3 <- c()
for(x in unique(imp.spp$Phylum)){
  Dim1 <- c(Dim1, mean(as.numeric(subset(imp.spp, imp.spp$Phylum == x)$Dim1)))
  Dim2 <- c(Dim2, mean(as.numeric(subset(imp.spp, imp.spp$Phylum == x)$Dim2)))
  Dim3 <- c(Dim3, mean(as.numeric(subset(imp.spp, imp.spp$Phylum == x)$Dim3)))
}

spe.corr.means <- as.data.frame(unique(imp.spp$Phylum))
spe.corr.means <- cbind(spe.corr.means, as.vector(Dim1), as.vector(Dim2), as.vector(Dim3))
colnames(spe.corr.means) <- c("Phylum", "Dim1", "Dim2", "Dim3")

tau.plot.order <- tau.plot[order(as.numeric(tau.plot$Tau)),]
OTUsr_20.order <- OTUsr_20[order(as.numeric(row.names(OTUsr_20))),]

PCoA <- ggplot(tau.plot.order, aes(x = V1, y = V2))+
  geom_path(linewidth = 1.25, color = alpha("black", 0.2))+
```

```
  geom_point(cex = 3, aes(shape = Turn, color = as.numeric(Tau)))+
  xlab(paste("PCoA 1 (", explainvar1, "%)", sep = ""))+
  ylab(paste("PCoA 2 (", explainvar2, "%)", sep = ""))+
  labs(shape = "Full Turnover", color = expression(paste(tau,  " (hrs)")))+
  scale_shape_manual(values = c(15, 19), labels = c("Yes", "No"))+
  scale_color_viridis(direction = -1, labels = label_math(expr = 10^.x, format = force))+
  theme(legend.title = element_text(size = 14))

PCoA
```




```
ggsave("./output/RTLC_BC_PCoA.pdf")

## Saving 6.5 x 4.5 in image
ggsave("./output/RTLC_BC_PCoA.png", width = 8, height = 5)
```

```
OTU_pumpturn <- ggplot(tau.plot.order, aes(x = V1, y = V2))+
  geom_path(linewidth = 1.25, color = alpha("black", 0.2))+
  xlab(paste("PCoA 1 (", explainvar1, "%)", sep = ""))+
  ylab(paste("PCoA 2 (", explainvar2, "%)", sep = ""))+
  geom_point(cex = 3, aes(shape = Turn, color = Pump))+
  labs(shape = "Number of\nturnovers", color = "Treatment\nmethod")+
  scale_color_manual(values = c("blue", "black"), labels = c("Pump", "Manual"))+
  scale_shape_manual(values = c(19, 1), labels = c("> 1", "< 1"))+
  theme(legend.title = element_text(size = 14))

OTU_pumpturn
```




```
ggsave("./output/RTLC_BC_OTUpumpturn.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_BC_OTUpumpturn.png", width = 8, height = 5)
```

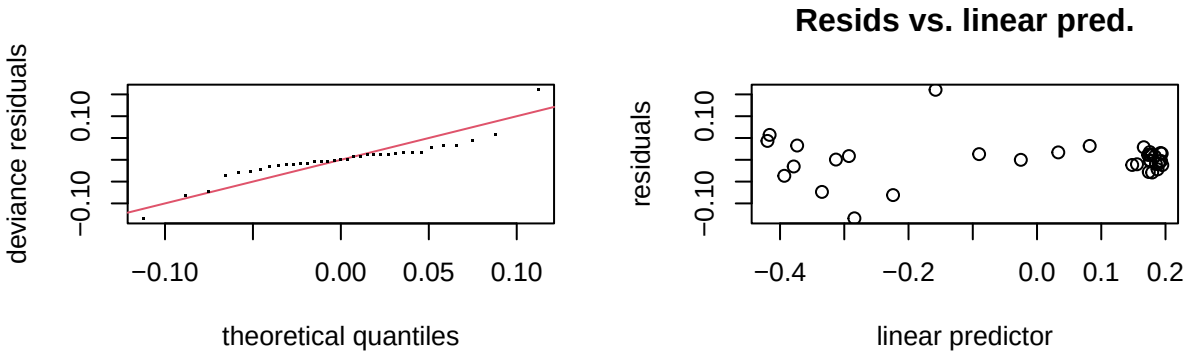
```
PCoA1_gam <- gam(V1 ~ s(Tau), family = gaussian(link = "identity"), data = tau.plot, method = "REML")
mean(summary(PCoA1_gam)$s.table[,4])
```

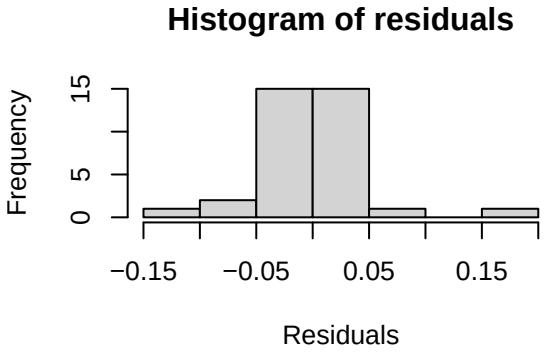
```
## [1] 0
```

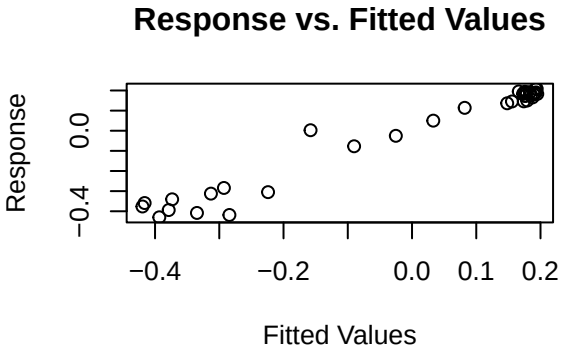
```
AIC(PCoA1_gam)
```

```
## [1] -98.60803
```

```
gam.check(PCoA1_gam)
```







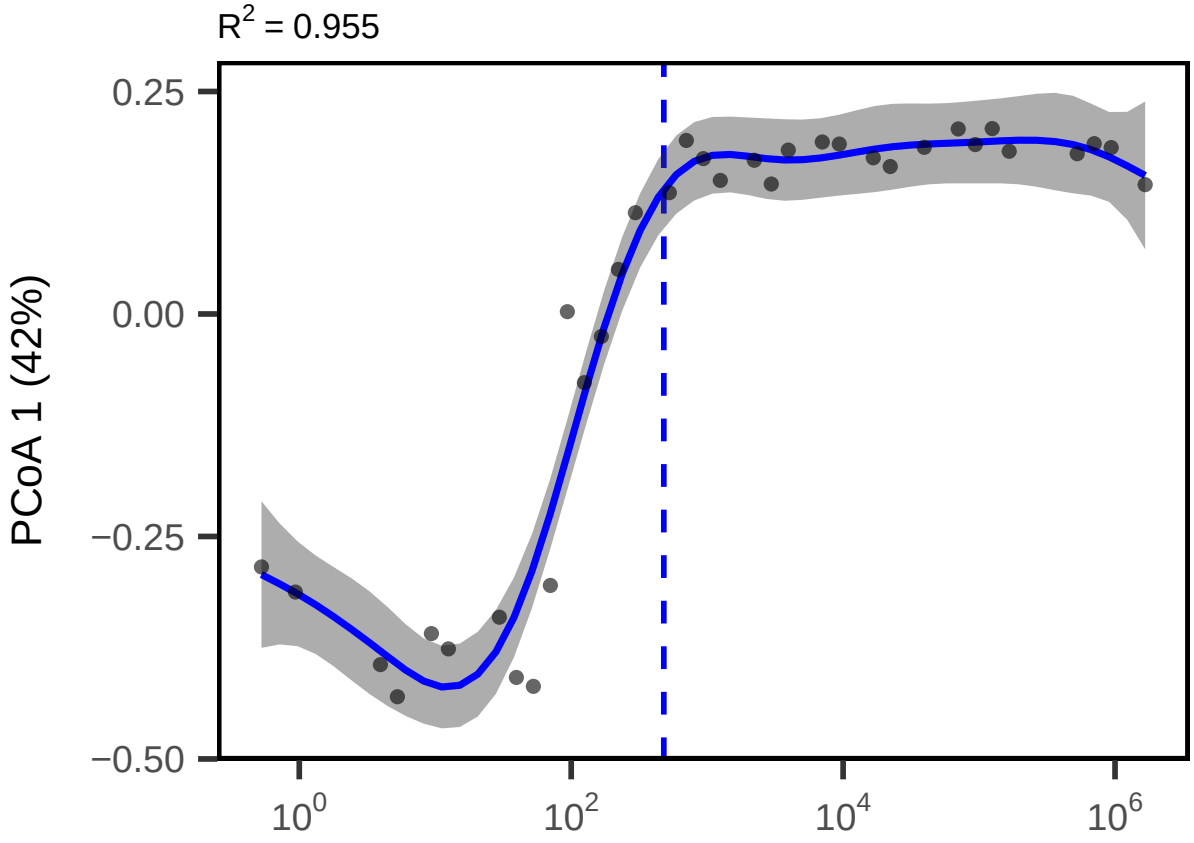

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-8.951339e-09,3.323744e-09]
## (score -36.68023 & scale 0.002628751).
## Hessian positive definite, eigenvalue range [1.97988,17.21017].
## Model rank =  10 / 10
##
## Basis dimension (k) checking 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(Tau) 9.00 7.42    0.88    0.17
```

```
summary(PCoA1_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## V1 ~ s(Tau)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.493e-17  8.666e-03       0        1
##
```

```
## Approximate significance of smooth terms:
##          edf Ref.df    F p-value
## s(Tau) 7.422  8.378 87.1  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.955   Deviance explained = 96.5%
## -REML = -36.68  Scale est. = 0.0026288  n = 35
```

```
PCoA1 <- predict_gam(PCoA1_gam) %>%
  ggplot(aes(Tau, fit))+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = tau.plot, aes(x = Tau, y = V1), size = 2, alpha = 0.6)+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  ylab(paste("PCoA 1 (", explainvar1, "%)", sep = ""))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(PCoA1_gam)$r.sq, 3))))+
  theme(axis.title.x = element_blank())

PCoA1
```




```
ggsave("./output/RTLC_BC_PCoA1.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_BC_PCoA1.png", width = 6.5, height = 5)

PCoA2_gam <- gam(V2 ~ s(Tau), family = gaussian(link = "identity"), data = tau.plot, method = "REML")
```

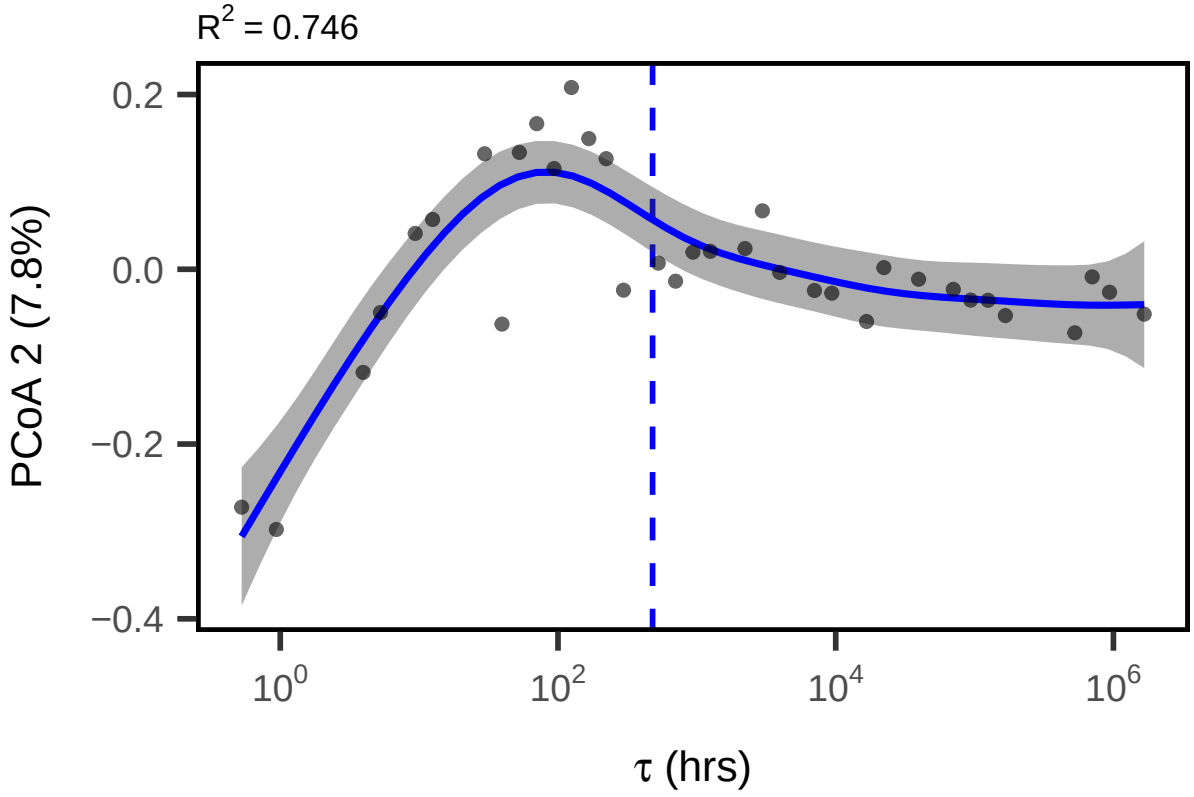
```r
summary(PCoA2_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## V2 ~ s(Tau)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.171e-17  8.908e-03       0        1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(Tau) 5.351  6.452 15.66  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.746   Deviance explained = 78.6%
## -REML = -40.918  Scale est. = 0.0027774  n = 35
```

```r
mean(summary(PCoA2_gam)$s.table[,4])
```

```
## [1] 0
```

```r
PCoA2 <- predict_gam(PCoA2_gam) %>%
  ggplot(aes(Tau, fit))+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = tau.plot, aes(x = Tau, y = V2), size = 2, alpha = 0.6)+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  xlab(expression(paste(tau, " (hrs)")))+
  ylab(paste("PCoA 2 (", explainvar2, "%)", sep = ""))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(PCoA2_gam)$r.sq, 3))))

PCoA2
```




```
ggsave("./output/RTLC_BC_PCoA2.pdf")
```

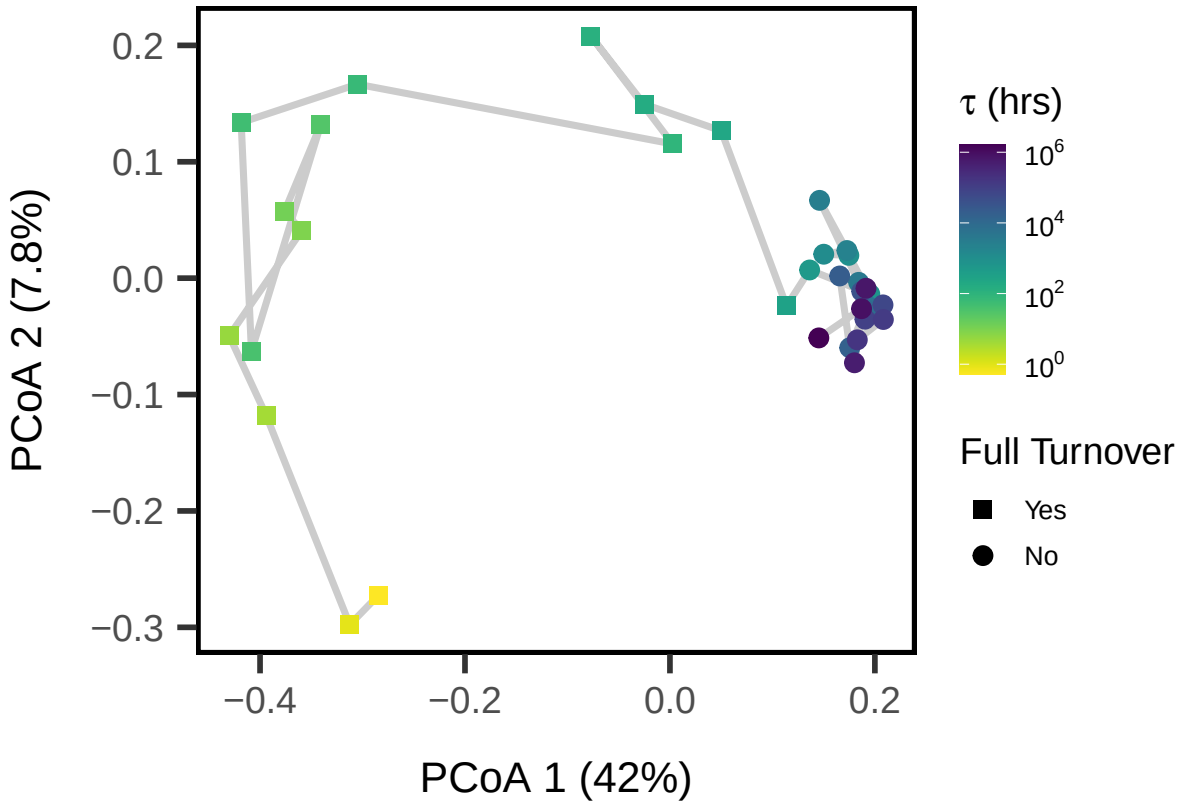
```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_BC_PCoA2.png")
```

```
## Saving 6.5 x 4.5 in image
```

##Draw Figure

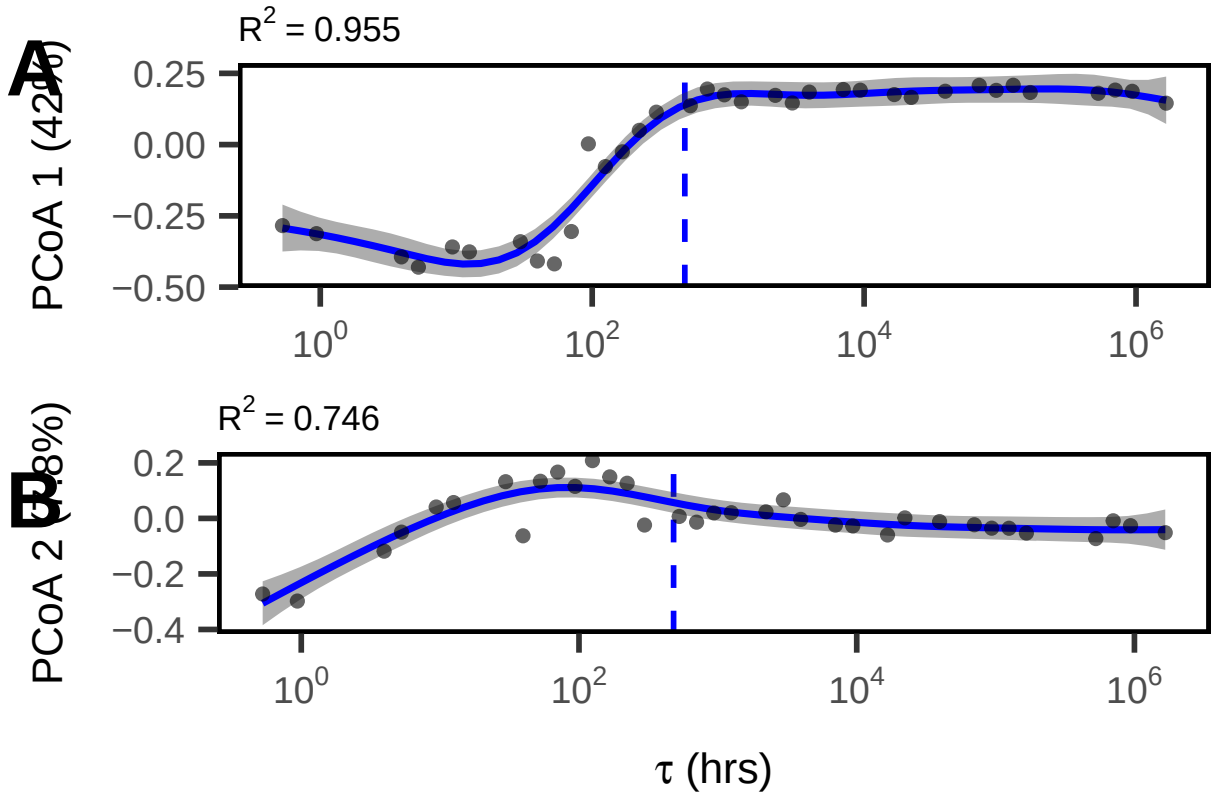
```
ggdraw()+
  draw_plot(PCoA, x = 0, y = 0, width = 1, height = 01) +
  theme(plot.background = element_rect(fill="white", color = NA))
```




```
ggsave("./output/CommPCoA.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/CommPCoA.png", width = 8, height = 10)

ggdraw()+
  draw_plot(PCoA1, x = 0, y = 0.55, width = 1, height = 0.45) +
  draw_plot(PCoA2, x = 0, y = 0, width = 1, height = 0.55) +
  theme(plot.background = element_rect(fill="white", color = NA))+
  draw_plot_label(label = c("A", "B"), size = 30, x = c(0,0), y = c(0.98,0.48))
```




```
ggsave("./output/CommPCoA_axes.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/CommPCoA_axes.png", width = 8, height = 10)
```

#Figure 4. Community Function - P

##Cstat - Biomass Production (uM C/hr)

```
# BP_lm <- lm(uMChr ~ Tau, data = Tau)
# BP_poly <- lm(uMChr ~ poly(Tau, 2, raw = TRUE), data = Tau)
# BP_gam <- gam(uMChr ~ s(Tau), family = gaussian(link = "identity"), data = Tau, method = "REML")
# BP_gam_gm <- gam(uMChr ~ s(Tau), family = Gamma(link = "log"), data = Tau, method = "REML")
#
# AIC(BP_lm, BP_poly, BP_gam, BP_gam_gm)

BP_gam_re_sp <- gam(log(uMChr, 10) ~ s(Tau) + s(Set, bs = "re"), family = gaussian(link = "identity"), d

# BP_gam_gm_re <- gam(uMChr ~ s(Tau) + s(Set, bs = "re"), family = Gamma(link = "log"), data = Tau, met
# BP_gam_gm_re_sp <- gam(uMChr ~ s(Tau, sp = 0.2) + s(Set, bs = "re"), family = Gamma(link = "log"), da

# anova(BP_gam_gm, BP_gam_gm_re, test = "Chisq")

summary(BP_gam_re_sp)
```

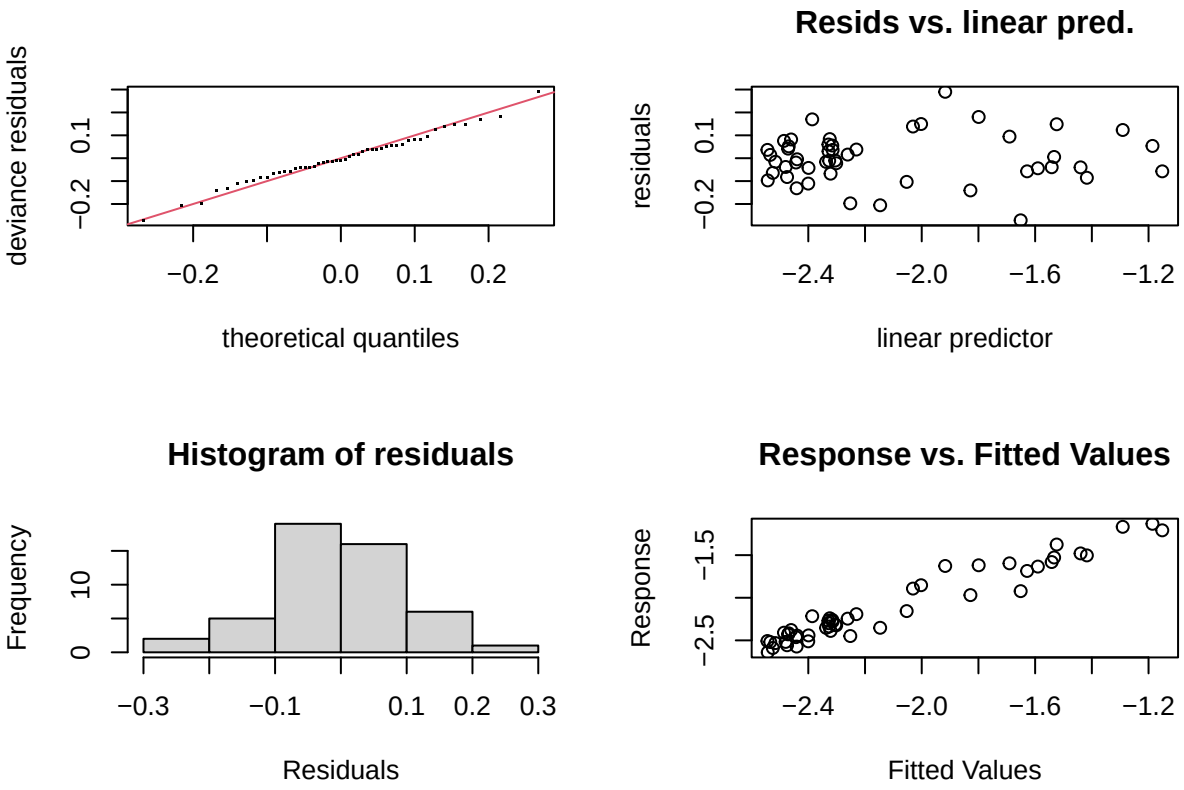
```
##
## Family: gaussian
```

```
## Link function: identity
##
## Formula:
## log(uMChr, 10) ~ s(Tau) + s(Set, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.1159     0.0536  -39.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df      F  p-value
## s(Tau) 5.042  6.161 96.966  < 2e-16 ***
## s(Set) 2.666  3.000  8.319 7.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.929   Deviance explained =   94%
## -REML = -22.399  Scale est. = 0.013289  n = 49
```

```
k.check(BP_gam_re_sp)
```

```
##        k'      edf  k-index p-value
## s(Tau)  9 5.042344 1.137607  0.7875
## s(Set)  4 2.666262       NA      NA
```

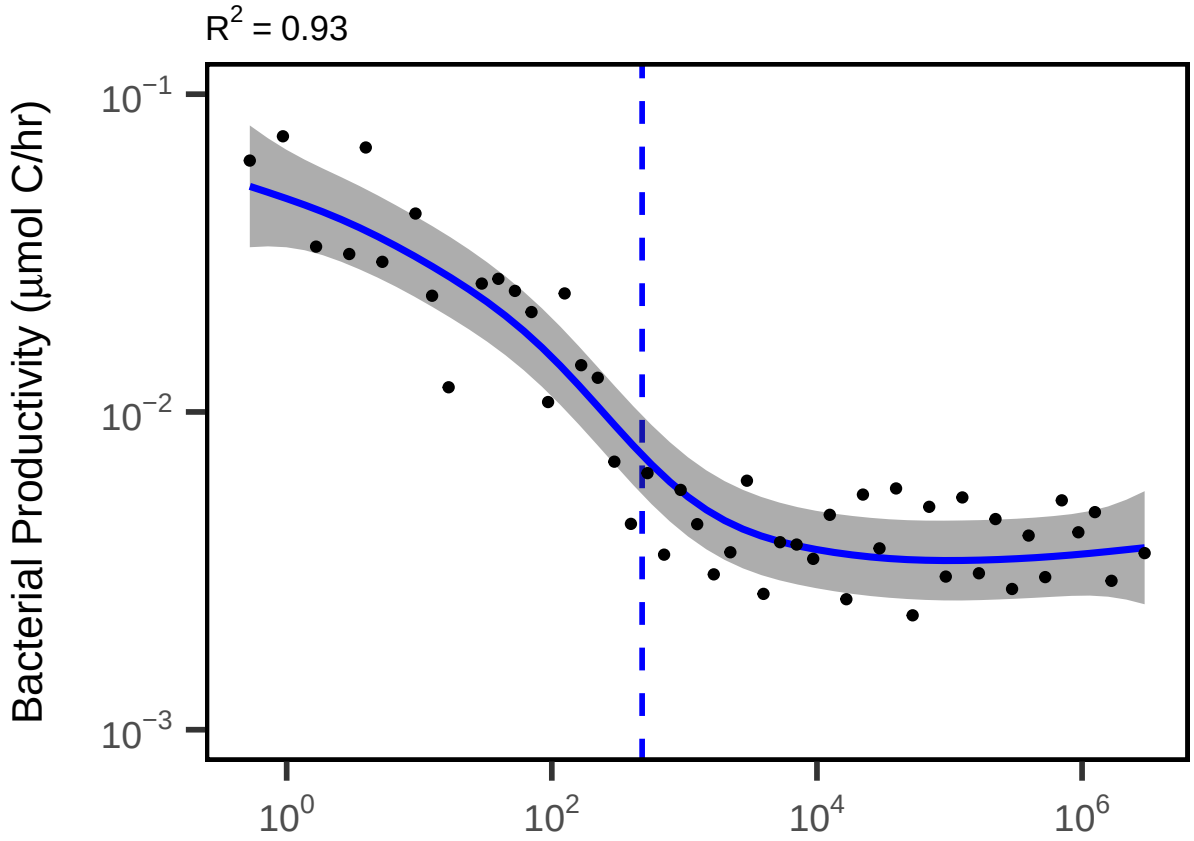
```
gam.check(BP_gam_re_sp)
```




```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-2.109971e-09,7.289502e-10]
## (score -22.39941 & scale 0.01328936).
## Hessian positive definite, eigenvalue range [1.085797,23.76268].
## Model rank =  14 / 14
##
## 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(Tau) 9.00 5.04    1.14    0.77
## s(Set) 4.00 2.67      NA      NA
```

```
BP_Tau <- predict_gam(BP_gam_re_sp, exclude_terms = "s(Set)") %>%
  filter(Set == "1") %>%
  ggplot(aes(Tau, fit))+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = Tau, aes(x = Tau, y = log(uMChr, 10)))+
#  xlab(expression(paste(tau, " (hrs)")))+
  ylab(expression(paste("Bacterial Productivity (", mu, "mol C/hr)")))+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  scale_y_continuous(limits = c(-3, -1), breaks = c(-1, -2, -3), labels = label_math(expr = 10^.x, forma
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(BP_gam_re_sp)$r.sq, 2))))+
#  theme(plot.margin = unit(c(1.5, 0.2, 0, 0.2), "cm"), axis.title.x = element_blank())+
```

```
  theme(axis.title.x = element_blank())

BP_Tau
```




```
ggsave("./output/RTLC_BP.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_BP.png", width = 6.5, height = 5)
```

##Average well response at 48 hours

```
Avg_lm <- lm(AvgRes ~ Tau, data = Tau)
Avg_lm2 <- lm(log(AvgRes, 10) ~ Tau, data = Tau)
Avg_poly <- lm(AvgRes ~ poly(Tau, 2, raw = TRUE), data = Tau)
Avg_log_re <- lmer(AvgRes ~ Tau + (1|Set), data = Tau)
Avg_gam <- gam(AvgRes ~ s(Tau), data = Tau, family = gaussian(link = "identity"), method = "REML")

AIC(Avg_lm, Avg_lm2, Avg_poly, Avg_log_re, Avg_gam)
```

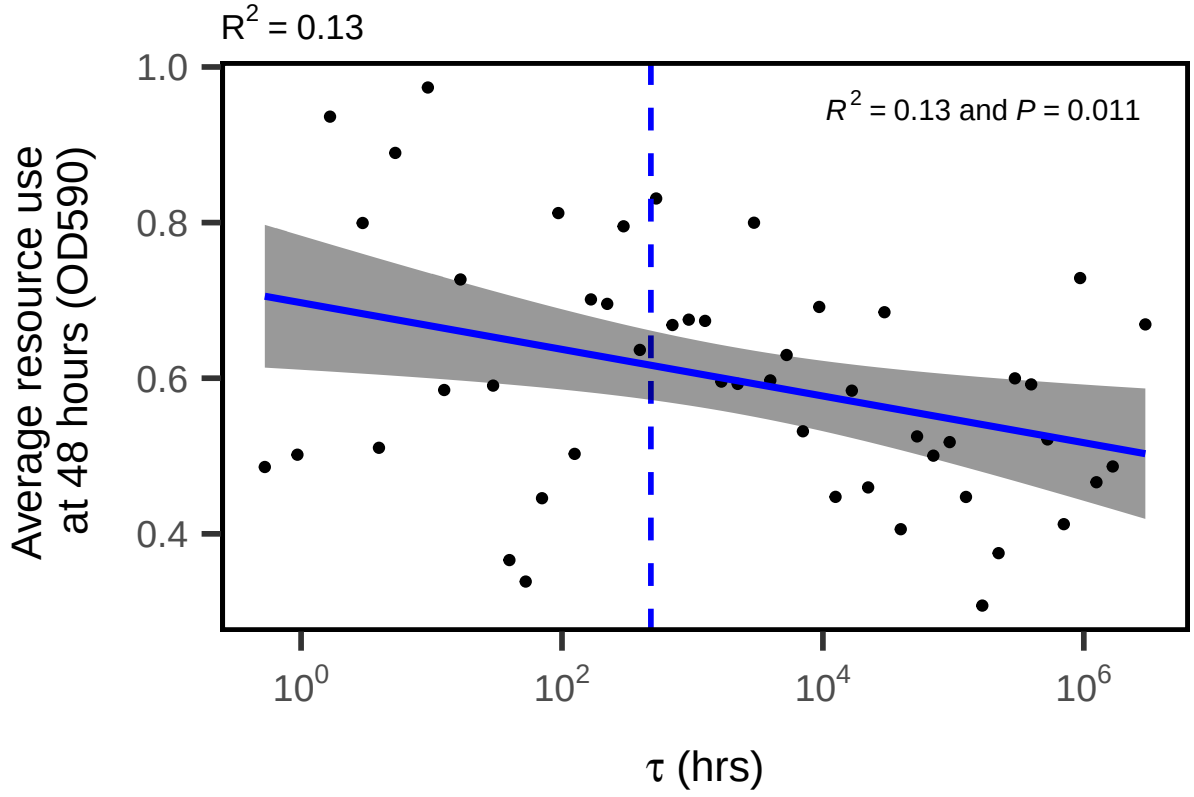
```
##                  df       AIC
## Avg_lm     3.000000 -44.79279
## Avg_lm2    3.000000 -73.10215
## Avg_poly   4.000000 -43.00174
## Avg_log_re 4.000000 -49.97815
## Avg_gam    3.000094 -44.79265
```

```r
summary(Avg_lm)
```

```
##
## Call:
## lm(formula = AvgRes ~ Tau, data = Tau)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.306727 -0.107065 -0.003979  0.070843  0.305724
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.69702    0.04284  16.272   <2e-16 ***
## Tau         -0.02998    0.01132  -2.647    0.011 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1471 on 47 degrees of freedom
## Multiple R-squared:  0.1298, Adjusted R-squared:  0.1112
## F-statistic: 7.008 on 1 and 47 DF,  p-value: 0.01101
```

```r
AvgRes <- ggplot(data = Tau, aes(x = Tau, y = AvgRes))+
  geom_point()+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  xlab(expression(paste(tau, " (hrs)")))+
  ylab("Average resource use \n at 48 hours (OD590)")+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  geom_smooth(method = "lm", formula = y ~ x, color = "blue", cex = 1.25, fill = alpha("black", 0.4))+
  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
               formula = y~x, parse = TRUE, size = 4, label.x = "right")+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(Avg_lm)$r.sq, 2))))

AvgRes
```

```
## Warning: `stat(rr.label)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(rr.label)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
ggsave("./output/RTLC_AvgRes.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_AvgRes.png", width = 6.5, height = 5)
```

```
N_RSG <- read.csv("./data/RTLC/2021_RTLC_RSG_N.csv", header = TRUE)
N_RSG$DorPer <- (N_RSG$N-N_RSG$RSG)/N_RSG$N
N_RSG$Tau <- log(((10^N_RSG$Tau)/60), 10)

Dorm_gam <- gam(data = N_RSG, (DorPer * 100) ~ s(Tau), family = gaussian(link = "identity"), method = "
Dorm_gam_re <- gam(data = N_RSG, (DorPer * 100) ~ s(Tau) + s(Set, bs = "re"), family = gaussian(link =
summary(Dorm_gam)
```

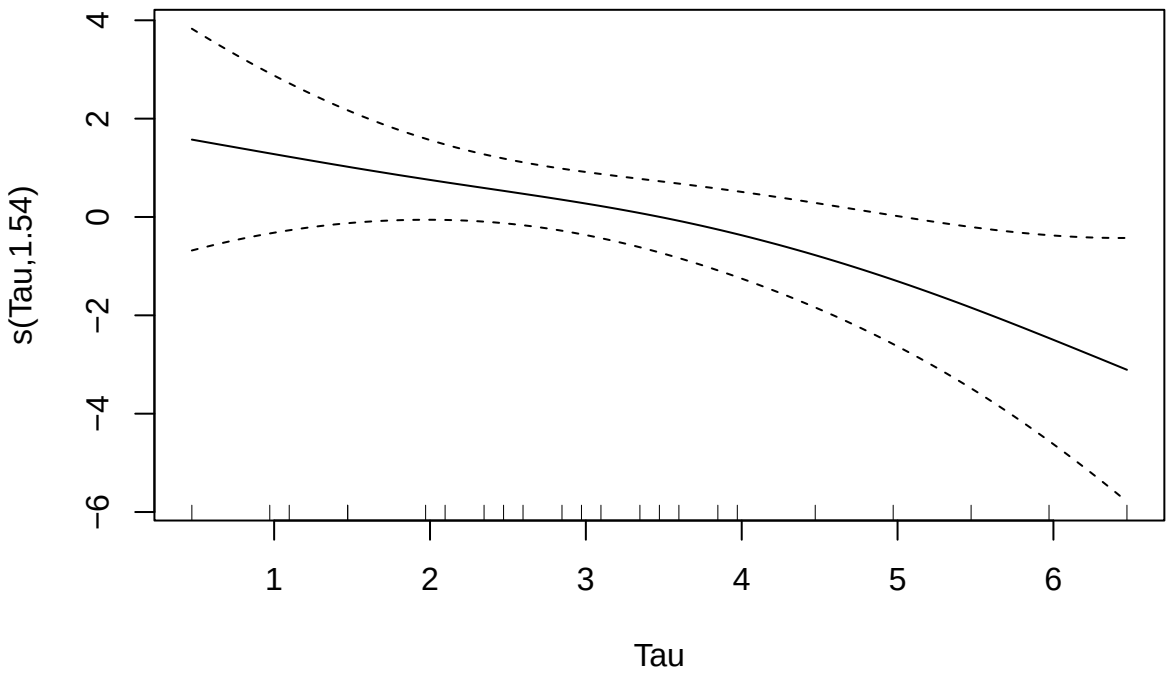
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## (DorPer * 100) ~ s(Tau)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  88.9460     0.5083     175   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(Tau) 1.537  1.894 3.303  0.0817 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.18   Deviance explained = 23.8%
## -REML = 51.977  Scale est. = 5.9421    n = 23
```

```
AIC(Dorm_gam, Dorm_gam_re)
```

```
##                   df      AIC
## Dorm_gam    3.893889 111.3576
## Dorm_gam_re 4.253419 111.4931
```

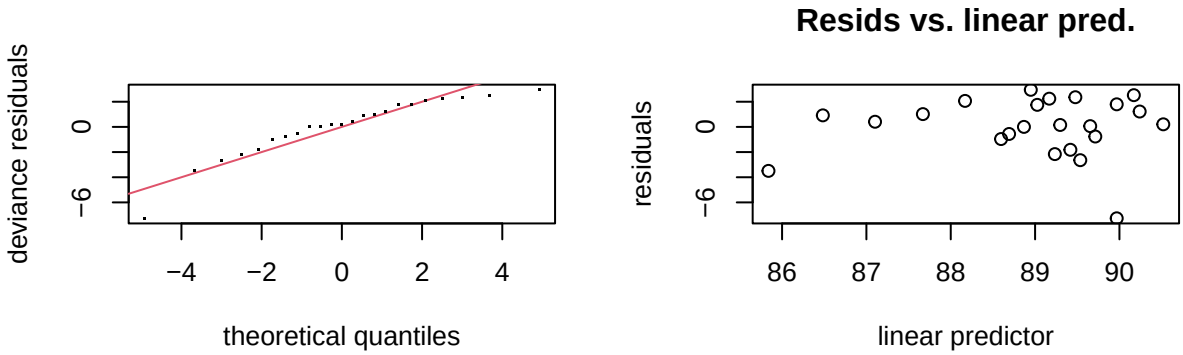
```
plot(Dorm_gam)
```

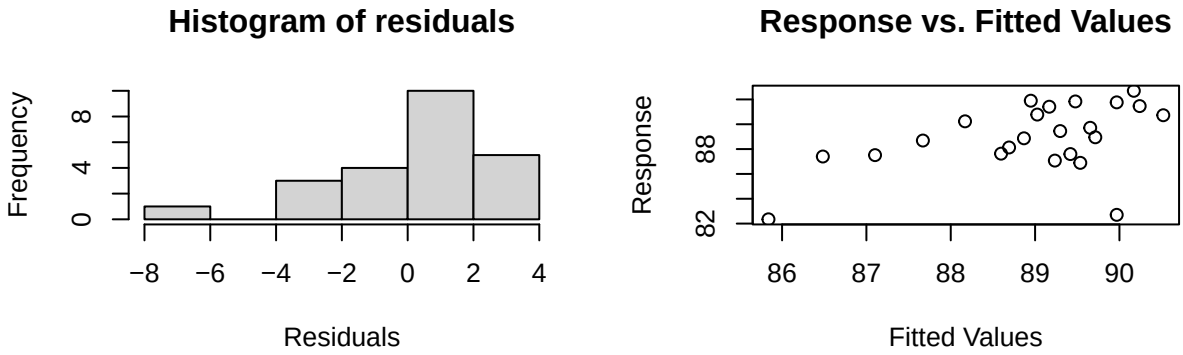



```
k.check(Dorm_gam)
```

```
##        k'      edf  k-index p-value
## s(Tau)  9 1.537469 1.065716  0.5475
```

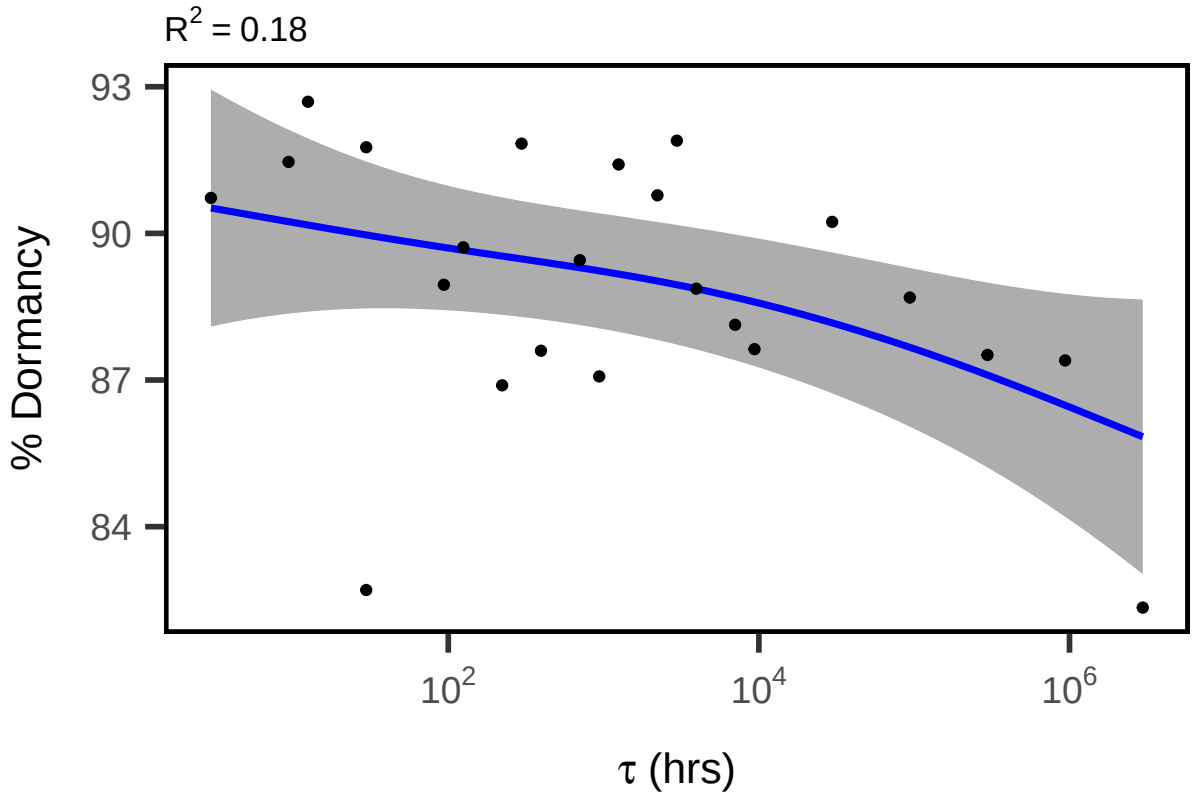
```
gam.check(Dorm_gam)
```






```
##
## Method: REML   Optimizer: outer newton
## full convergence after 8 iterations.
## Gradient range [-1.064124e-07,1.82711e-07]
## (score 51.97675 & scale 5.942076).
## Hessian positive definite, eigenvalue range [0.05069509,10.50691].
## Model rank =  10 / 10
##
## 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(Tau) 9.00 1.54    1.07    0.56
```

```
Dorm_Tau <- predict_gam(Dorm_gam) %>%
  ggplot(aes(Tau, fit))+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = N_RSG, aes(x = Tau, y = (DorPer * 100)))+
  xlab(expression(paste(tau, " (hrs)")))+
  ylab("% Dormancy")+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(Dorm_gam)$r.sq, 2))))

Dorm_Tau
```




```
ggsave("./output/RTLC_Dorm.pdf")
```

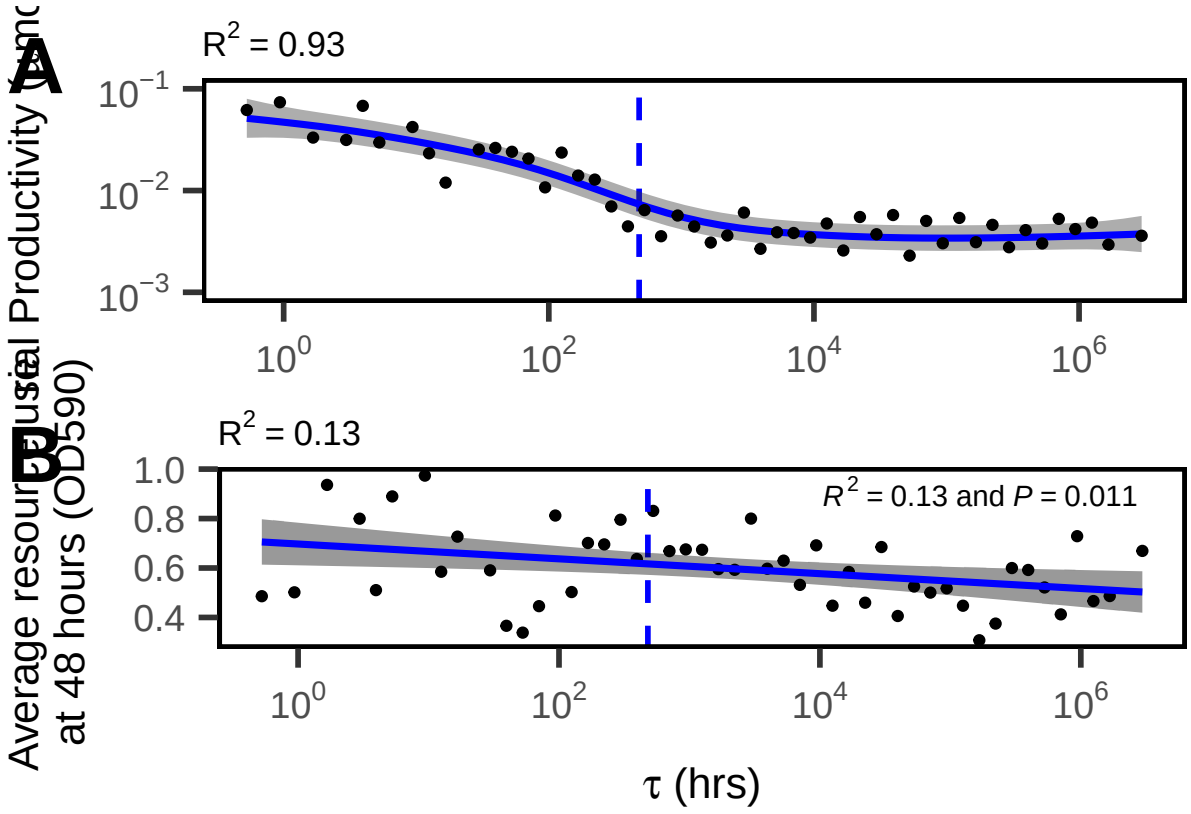
```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_Dorm.png")
```

```
## Saving 6.5 x 4.5 in image
```

#Draw figure

```
ggdraw()+
  draw_plot(BP_Tau, x = 0, y = 0.55, width = 1, height = 0.45) +
  draw_plot(AvgRes, x = 0, y = 0, width = 1, height = 0.55) +
  draw_plot_label(label = c("A", "B"), size = 30, x = c(0.02,0.02), y = c(1,0.55))+
  theme(plot.background = element_rect(fill="white", color = NA))
```




```
ggsave("./output/Comm_Func.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/Comm_Func.png", width = 8, height = 10)
```

#Figure 5. Significant resource use

```
csource_sig <- c()
for(c in unique(long_EP_slope$C_Source)){
  csource.lm <- lm(data = subset(long_EP_slope, long_EP_slope$C_Source == c), Slope ~ Tau)
  if(summary(csource.lm)$coefficients[2,4] < (0.05/31)){
    csource_sig <- c(csource_sig, c)
    assign(paste("lm.", gsub("-", "_", c), sep = ""), csource.lm)
  }
}
csource_sig
```

```
##  [1] "4-Hydroxy.Benzoic.Acid"          "D-Galactonic.Acid.gamma-Lactone"
##  [3] "D-Glucosaminic.Acid"             "D-Xylose"
##  [5] "L-Arginine"                      "L-Asparagine"
##  [7] "L-Serine"                        "L-Threonine"
##  [9] "Putrescine"                      "Pyruvic.Acid.Methyl.Ester"
## [11] "Tween.40"                        "Tween.80"
```

```
ep.mods <- as_tibble(rbind.data.frame(
  tidy(lm.4_Hydroxy.Benzoic.Acid) %>% add_column(Resource = "4-Hydroxy Benzoic Acid"),
  tidy(lm.D_Galactonic.Acid.gamma_Lactone) %>% add_column(Resource = expression(paste("D-Galactonic Aci
```

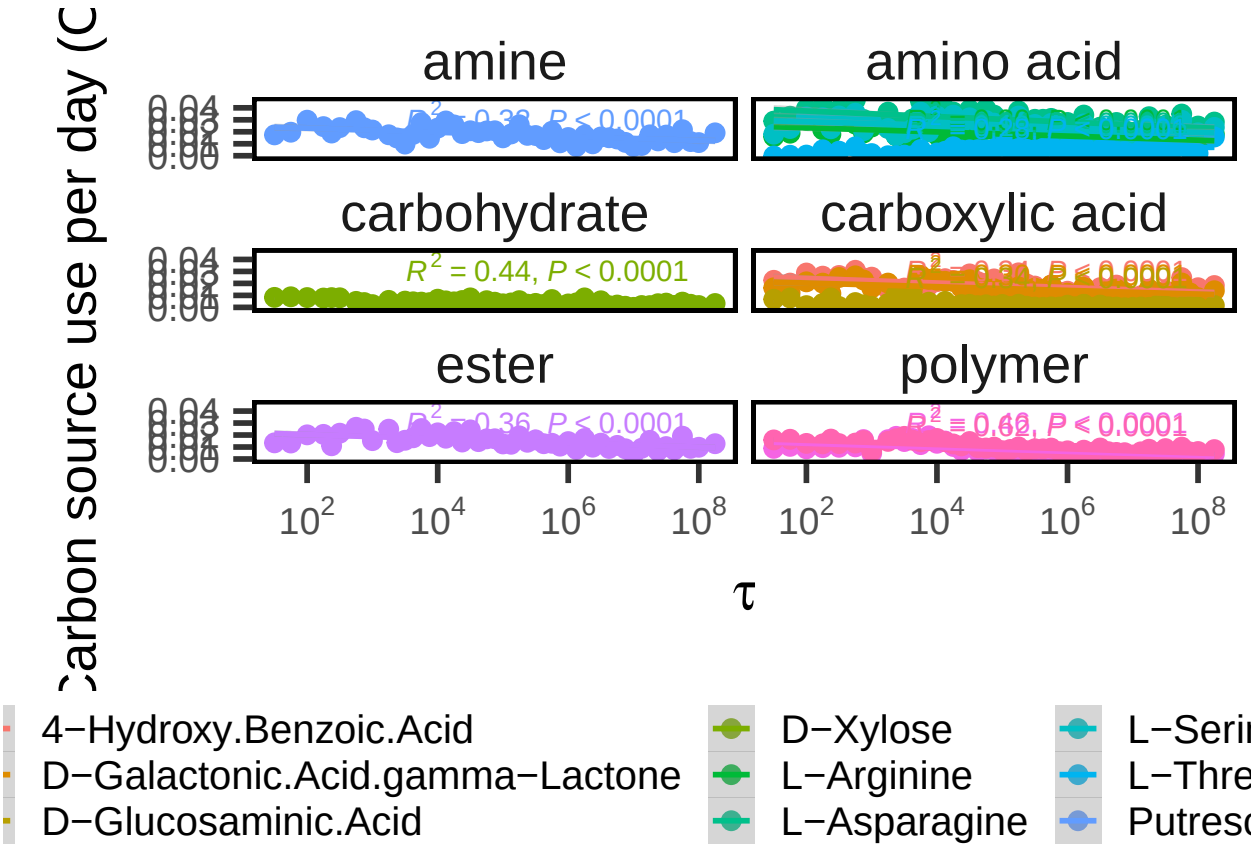
```
  tidy(lm.D_Glucosaminic.Acid) %>% add_column(Resource = "D-Glucosaminic Acid"),
  tidy(lm.D_Xylose) %>% add_column(Resource = "D-Xylose"),
  tidy(lm.L_Arginine) %>% add_column(Resource = "L-Arginine"),
  tidy(lm.L_Asparagine) %>% add_column(Resource = "L-Asparagine"),
  tidy(lm.L_Serine) %>% add_column(Resource = "L-Serine"),
  tidy(lm.L_Threonine) %>% add_column(Resource = "L-Threonine"),
  tidy(lm.Putrescine) %>% add_column(Resource = "Putrescine"),
  tidy(lm.Pyruvic.Acid.Methyl.Ester) %>% add_column(Resource = "Pyruvic Acid Methyl Ester"),
  tidy(lm.Tween.40) %>% add_column(Resource = "Tween 40"),
  tidy(lm.Tween.80) %>% add_column(Resource = "Tween 80")
))

ep.mods %>%
  group_by(Resource)%>%
  rename("Term" = term,
         "Estimate" = estimate,
         "Std. Error" = std.error,
         "Statistic" = statistic,
         "p-value" = p.value) %>%
  select(Resource, everything()) %>%
  pander(round = 4)
```

Table 1: Table continues below

Resource	Term	Estimate	Std. Error	Statistic
4-Hydroxy Benzoic Acid	(Intercept)	0.0279	0.0021	13.48
4-Hydroxy Benzoic Acid	Tau	-0.0019	4e-04	-4.939
paste(“D-Galactonic Acid”, gamma, “-Lactone”)	(Intercept)	0.0234	0.0019	12.55
paste(“D-Galactonic Acid”, gamma, “-Lactone”)	Tau	-0.0015	3e-04	-4.461
D-Glucosaminic Acid	(Intercept)	0.0061	9e-04	6.919
D-Glucosaminic Acid	Tau	-8e-04	2e-04	-4.866
D-Xylose	(Intercept)	0.0084	7e-04	11.31
D-Xylose	Tau	-8e-04	1e-04	-6.054
L-Arginine	(Intercept)	0.0268	0.0021	12.75
L-Arginine	Tau	-0.0017	4e-04	-4.447
L-Asparagine	(Intercept)	0.0419	0.0025	16.74
L-Asparagine	Tau	-0.0023	5e-04	-4.958
L-Serine	(Intercept)	0.0313	0.0024	13.16
L-Serine	Tau	-0.0015	4e-04	-3.355
L-Threonine	(Intercept)	0	0.0011	-0.0169
L-Threonine	Tau	9e-04	2e-04	4.268
Putrescine	(Intercept)	0.0264	0.002	13.25
Putrescine	Tau	-0.0017	4e-04	-4.649
Pyruvic Acid Methyl Ester	(Intercept)	0.0239	0.0018	13.07
Pyruvic Acid Methyl Ester	Tau	-0.0017	3e-04	-5.149
Tween 40	(Intercept)	0.0165	0.0015	11.31
Tween 40	Tau	-0.0017	3e-04	-6.268
Tween 80	(Intercept)	0.0191	0.0011	17.65
Tween 80	Tau	-0.0018	2e-04	-8.81

p-value
0
0
0
1e-04
0
0
0
0
0
1e-04
0
0
0
0.0016
0.9866
1e-04
0
0
0
0
0
0
0
0

```
Csource_sig <- ggplot(data = long_EP_slope[long_EP_slope$C_Source %in% csource_sig,], aes(x = Tau, y = S
  geom_point(size = 3)+
  geom_smooth(method = "lm", formula = y~x)+
  stat_poly_eq(aes(label = paste(stat(rr.label), "*\", \"*", stat(p.value.label), sep = "")),
               formula = y~x, parse = TRUE, size = 4, p.digits = 4, label.x = 0.9, label.y = c(0.95, 0.9
  xlab(expression(tau))+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  ylab("Carbon source use per day (OD600)")+
  labs(color = "Carbon Source")+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 15), axis.title = eler
  theme(legend.position="bottom", strip.text = element_text(size=20))+
  facet_wrap(~Type, ncol = 2, nrow = 3)


Csource_sig
```




```
ggsave("./output/RTLC_Res_Csource_sig.pdf", width = 15, height =15)
ggsave("./output/RTLC_Res_Csource_sig.png", width = 15, height = 15)
```

#Figure 6. Community Function - Resource Use #Bray-Curtis distance Resource use

```
EcoPlate_20 <- EcoPlate[rownames(OTUsr_20),]
EP.db.20 <- vegdist(EcoPlate_20, method = "bray")
EP.db <-vegdist(EcoPlate, method = "bray")
EP.pcoa <- cmdscale(EP.db, eig = TRUE)

EPREL <- EcoPlate
for(i in 1:nrow(EcoPlate)){
  EPREL[i,] = EcoPlate[i,]/sum(EcoPlate[i,])
}

EP.pcoa <- add.spec.scores(EP.pcoa, EPREL, method = "pcoa.scores")

explainvar1.ep <- round(EP.pcoa$eig[1] / sum(EP.pcoa$eig), 3) * 100
explainvar2.ep <- round(EP.pcoa$eig[2] / sum(EP.pcoa$eig), 3) * 100
explainvar3.ep <- round(EP.pcoa$eig[3] / sum(EP.pcoa$eig), 3) * 100

types <- read.csv("data/RTLC/EcoPlate/Csourcetypes.csv", header = TRUE, sep = ",")

EP.plot <- as.data.frame(EP.pcoa$points)
EP.plot <- cbind(EP.plot, Tau$Tau, Tau$Turn, Tau$Pump)
colnames(EP.plot) <- c("V1", "V2", "Tau", "Turn", "Pump")
EP.plot$Turn <- factor(EP.plot$Turn, levels = c("TRUE", "FALSE"))
```

```r
EP.plot$Pump <- factor(EP.plot$Pump, levels = c("TRUE", "FALSE"))

EP.cproj <- as.data.frame(EP.pcoa$cproj)
EP.cproj <- cbind(EP.cproj, as.data.frame(row.names(EP.pcoa$cproj)))
spe.corr.ep <- add.spec.scores(EP.pcoa, EPREL, method = "cor.scores")$cproj
spe.corr.ep <- as.data.frame(cbind(spe.corr.ep,types$Type))
corrcut.ep <- 0.7
imp.spp.ep <- as.data.frame(spe.corr.ep[abs(as.numeric(spe.corr.ep[,1])) >= corrcut.ep | abs(as.numeric

Dim1.ep <- c()
Dim2.ep <- c()
for(x in unique(spe.corr.ep$V3)){
  Dim1.ep <- c(Dim1.ep, mean(as.numeric(subset(spe.corr.ep, spe.corr.ep$V3 == x)$Dim1)))
  Dim2.ep <- c(Dim2.ep, mean(as.numeric(subset(spe.corr.ep, spe.corr.ep$V3 == x)$Dim2)))
}

spe.corr.ep.means <- as.data.frame(unique(spe.corr.ep$V3))
spe.corr.ep.means <- cbind(spe.corr.ep.means, as.vector(Dim1.ep), as.vector(Dim2.ep))
colnames(spe.corr.ep.means) <- c("C_types", "Dim1", "Dim2")

fit <- envfit(EP.pcoa, EPREL, perm = 999)
fit
```

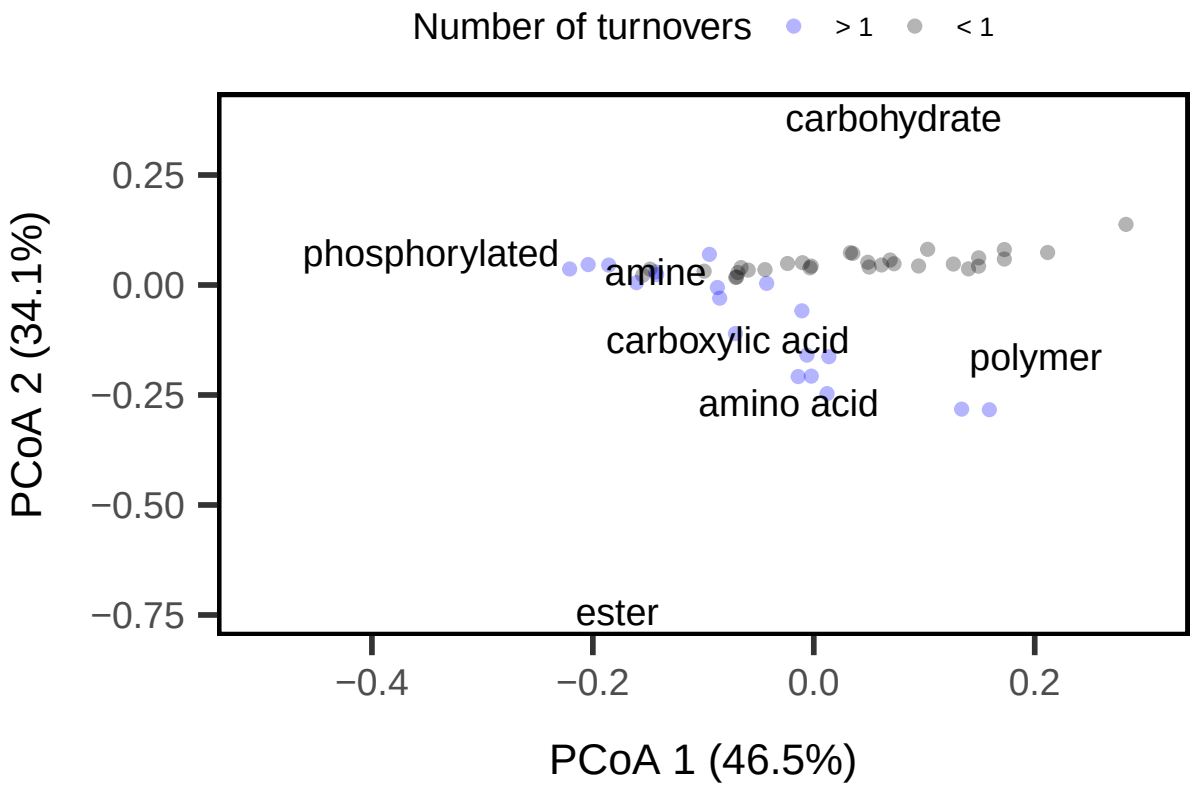
```
##
## ***VECTORS
##
##                                    Dim1     Dim2     r2 Pr(>r)
## X2.Hydroxy.Benzoic.Acid         -0.53492  0.84490 0.1134  0.067 .
## X4.Hydroxy.Benzoic.Acid         -0.26706 -0.96368 0.3872  0.001 ***
## alpha.Cyclodextrin               0.64510  0.76410 0.3523  0.001 ***
## alpha.D.Lactose                 -0.37648  0.92642 0.0706  0.204
## alpha.Ketobutyric.Acid           0.72908  0.68443 0.0871  0.114
## beta.Methyl.D.Glucoside          0.09447  0.99553 0.8586  0.001 ***
## D.Cellulobiose                   0.09011  0.99593 0.7406  0.001 ***
## D.Galactonic.Acid.gamma.Lactone -0.24631 -0.96919 0.4149  0.001 ***
## D.Galacturonic.Acid             -0.56686 -0.82381 0.1035  0.079 .
## D.Glucosaminic.Acid              0.06496 -0.99789 0.6408  0.001 ***
## D.Malic.Acid                    -0.51610  0.85653 0.4792  0.001 ***
## D.Mannitol                       0.85018  0.52650 0.2565  0.002 **
## D.Xylose                        -0.31648 -0.94860 0.4320  0.001 ***
## D.L.alpha.Glycerol.Phosphate    -0.99153 -0.12987 0.0814  0.140
## gamma.Hydroxybutyric.Acid        0.35206 -0.93598 0.0296  0.490
## Glucose.1.Phosphate             -0.88900  0.45790 0.2000  0.006 **
## Glycogen                         0.20096  0.97960 0.7179  0.001 ***
## Glycyl.L.Glutamic.Acid          -0.86095 -0.50869 0.0089  0.808
## i.Erythritol                     0.63253  0.77454 0.0426  0.376
## Itaconic.Acid                    0.34198  0.93971 0.0862  0.140
## L.Arginine                      -0.31516 -0.94904 0.5569  0.001 ***
## L.Asparagine                     0.13250 -0.99118 0.8257  0.001 ***
## L.Phenylalanine                 -0.71913  0.69488 0.0234  0.581
## L.Serine                         0.13939 -0.99024 0.4696  0.001 ***
## L.Threonine                      0.11754  0.99307 0.3818  0.001 ***
## N.Acetyl.D.Glucosamine           0.07591  0.99711 0.7650  0.001 ***
## Phenylethylamine                 0.03510  0.99938 0.6041  0.001 ***
```

```
## Putrescine                      -0.35529 -0.93476 0.6147  0.001 ***
## Pyruvic.Acid.Methyl.Ester       -0.20120 -0.97955 0.5817  0.001 ***
## Tween.40                         0.03070 -0.99953 0.8875  0.001 ***
## Tween.80                         0.13958 -0.99021 0.9341  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
EP.plot$Tau <- as.numeric(EP.plot$Tau)
typeof(EP.plot$Tau)
```

```
## [1] "double"
```

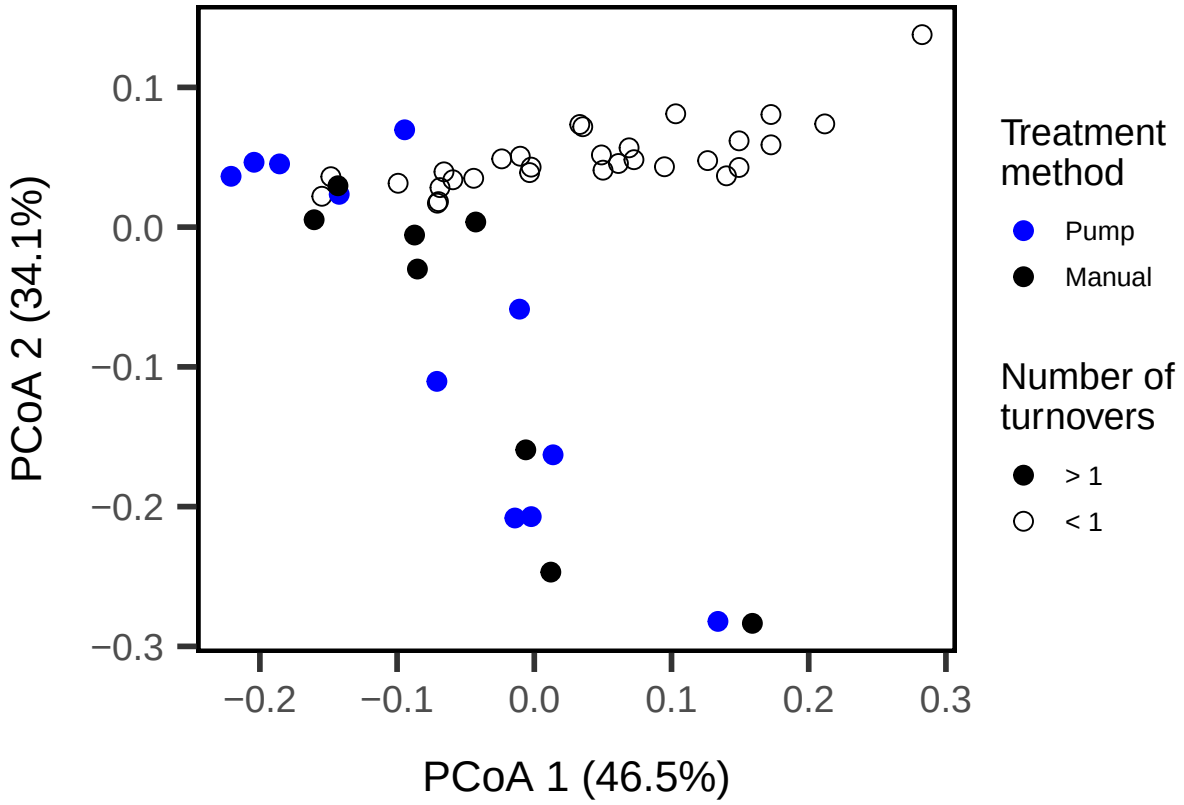
```
EP_BC <- ggplot(EP.plot, aes(x = V1, y = V2, color = Turn))+
  geom_point(cex = 2)+
  xlab(paste("PCoA 1 (", explainvar1.ep, "%)", sep = ""))+
  ylab(paste("PCoA 2 (", explainvar2.ep, "%)", sep = ""))+
  xlim(c(-0.5, 0.3))+
  labs(color = "Number of turnovers")+
  theme(legend.position = "top", legend.title = element_text(size = 14))+
  scale_color_manual(values = c(alpha("blue", 0.3), alpha("black", 0.3)), labels = c("> 1", "< 1"))+
  geom_text(size = 5, data = spe.corr.ep.means, aes(x = Dim1, y = Dim2, label = C_types), color = "blac

EP_BC
```




```
ggsave("./output/RTLC_Res_BC_PCoA.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_Res_BC_PCoA.png", width = 8, height = 5)

EP_BC_turn <- ggplot(EP.plot, aes(x = V1, y = V2))+
  geom_point(cex = 3, aes(color = Pump, shape = Turn))+
  xlab(paste("PCoA 1 (", explainvar1.ep, "%)", sep = ""))+
  ylab(paste("PCoA 2 (", explainvar2.ep, "%)", sep = ""))+
  labs(shape = "Number of\nturnovers", color = "Treatment\nmethod")+
  scale_color_manual(values = c("blue", "black"), labels = c("Pump", "Manual"))+
  scale_shape_manual(values = c(19, 1), labels = c("> 1", "< 1"))+
  theme(legend.title = element_text(size = 14))

EP_BC_turn
```




```
ggsave("./output/RTLC_Res_BC_PCoA_turn.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_Res_BC_PCoA_turn.png", width = 8, height = 5)

EcoPlate_env$Set <- c(rep(1, 13), rep(2,12), rep(3,10), rep(4,14))

EP.dist <- vegdist(EcoPlate, method = "bray")
Env.dist <- vegdist(scale(EcoPlate_env[,-2]), method = "euclid")

mantel(EP.dist, Env.dist)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = EP.dist, ydis = Env.dist)
##
## Mantel statistic r: 0.3918
##       Significance: 0.001
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0833 0.1062 0.1244 0.1580
## Permutation: free
## Number of permutations: 999
```

```
PC1_lm <- lm(V1 ~ Tau, data = EP.plot)
PC1_lm2 <- lm(V1 ~ Tau + Turn, data = EP.plot)
PC1_lm3 <- lm(V1 ~ Tau + Turn + Tau:Turn, data = EP.plot)
summary(PC1_lm)
```

```
##
## Call:
## lm(formula = V1 ~ Tau, data = EP.plot)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.187219 -0.064078 -0.005091  0.059682  0.219325
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.108353   0.029095  -3.724 0.000524 ***
## Tau          0.032876   0.007692   4.274  9.3e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09994 on 47 degrees of freedom
## Multiple R-squared:  0.2799, Adjusted R-squared:  0.2646
## F-statistic: 18.27 on 1 and 47 DF,  p-value: 9.298e-05
```

```
anova(PC1_lm, PC1_lm2, PC1_lm3)
```

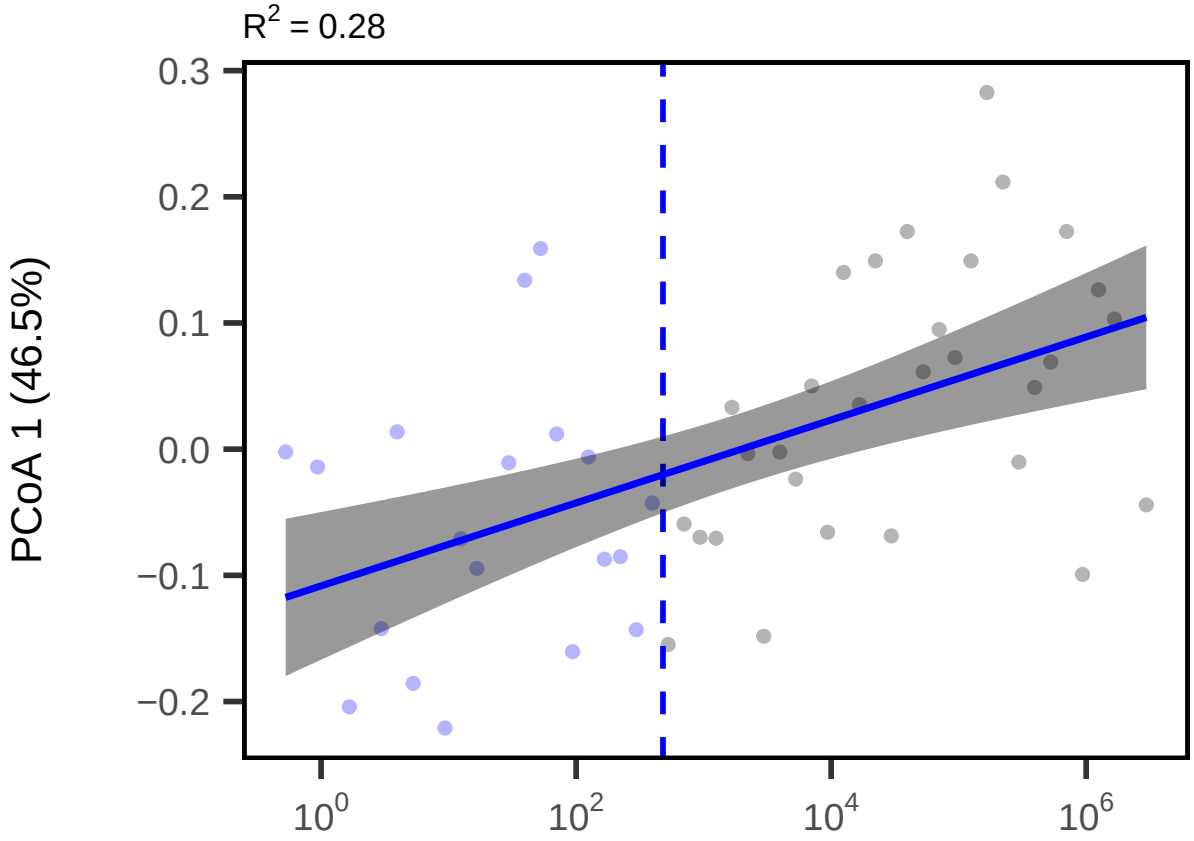
```
## Analysis of Variance Table
##
## Model 1: V1 ~ Tau
## Model 2: V1 ~ Tau + Turn
## Model 3: V1 ~ Tau + Turn + Tau:Turn
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     47 0.46941
## 2     46 0.46781  1 0.0015963 0.1570 0.6938
## 3     45 0.45746  1 0.0103552 1.0186 0.3182
```

```
AIC(PC1_lm, PC1_lm2, PC1_lm3)
```

```
##         df       AIC
## PC1_lm   3 -82.70094
## PC1_lm2  4 -80.86786
```

```
## PC1_lm3  5 -79.96467
```

```
Res_PCoA1 <- ggplot(EP.plot, aes(x = Tau, y = V1))+
  geom_point(cex = 2, aes(color = Turn))+
  ylab(paste("PCoA 1 (", explainvar1.ep, "%)\n", sep = ""))+
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  xlab(expression(paste(tau, " (hrs)")))+
  labs(color = "Number of turnovers")+
  geom_smooth(method = "lm", formula = y~x, color = "blue", cex = 1.25, fill = alpha("black", 0.4))+
  scale_color_manual(values = c(alpha("blue", 0.3), alpha("black", 0.3)), labels = c("> 1", "< 1"))+
#  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
#               formula = y~x, parse = TRUE, size = 4)+
  theme(legend.position = "none")+
  theme(axis.title.x = element_blank())+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(PC1_lm)$r.sq, 2))))

Res_PCoA1
```

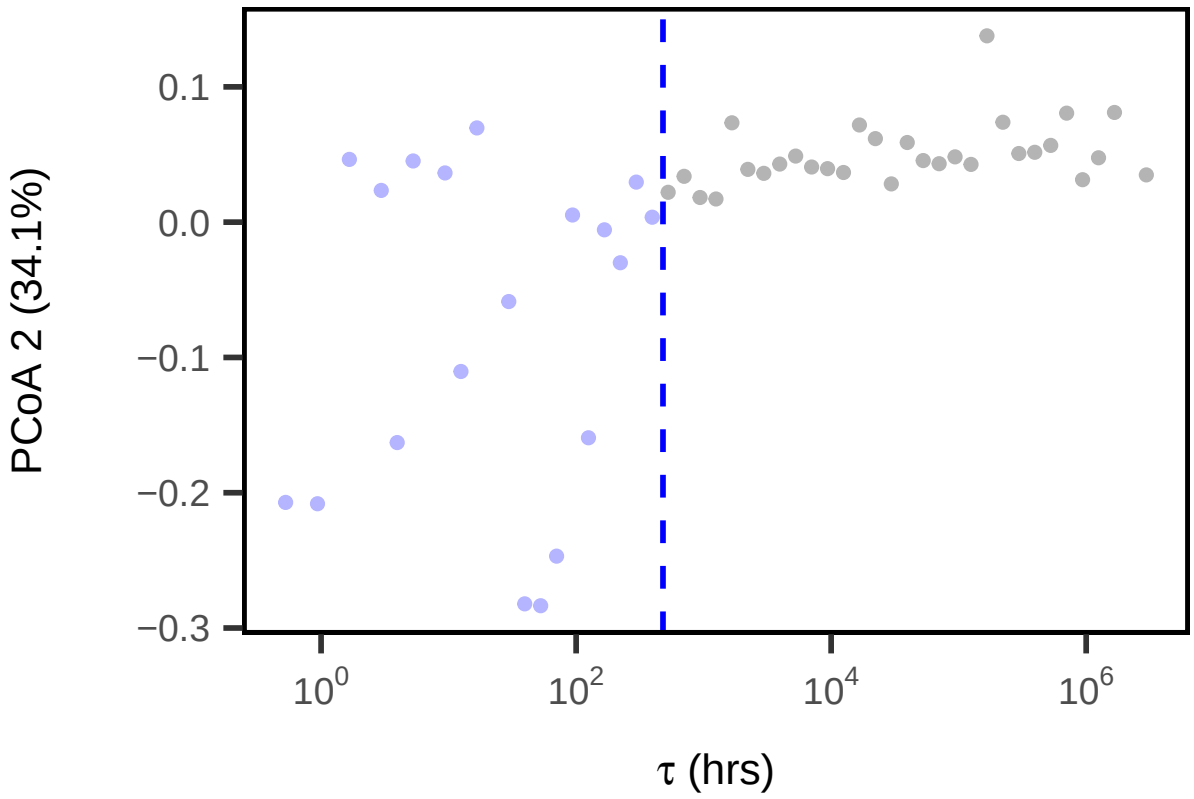



```
ggsave("./output/RTLC_Res_BC_PCoA1.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_Res_BC_PCoA1.png", width = 6.5, height = 5)

Res_PCoA2 <- ggplot(EP.plot, aes(x = Tau, y = V2))+
  geom_point(cex = 2, aes(color = Turn))+
  ylab(paste("PCoA 2 (", explainvar2.ep, "%)\n", sep = ""))+
```

```
  geom_vline(xintercept = turnover, linetype = "dashed", color = "blue", size = 1)+
  xlab(expression(paste(tau, " (hrs)")))+
  labs(color = "Number of turnovers")+
#  geom_smooth(method = "lm", formula = y~x, color = "blue", cex = 1.25, fill = alpha("black", 0.4))+
  scale_color_manual(values = c(alpha("blue", 0.3), alpha("black", 0.3)), labels = c("> 1", "< 1"))+
#  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
#               formula = y~x, parse = TRUE, size = 4)+
  theme(legend.position = "none")+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))
#  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(PC2_lm)$r.sq, 2))))

Res_PCoA2
```



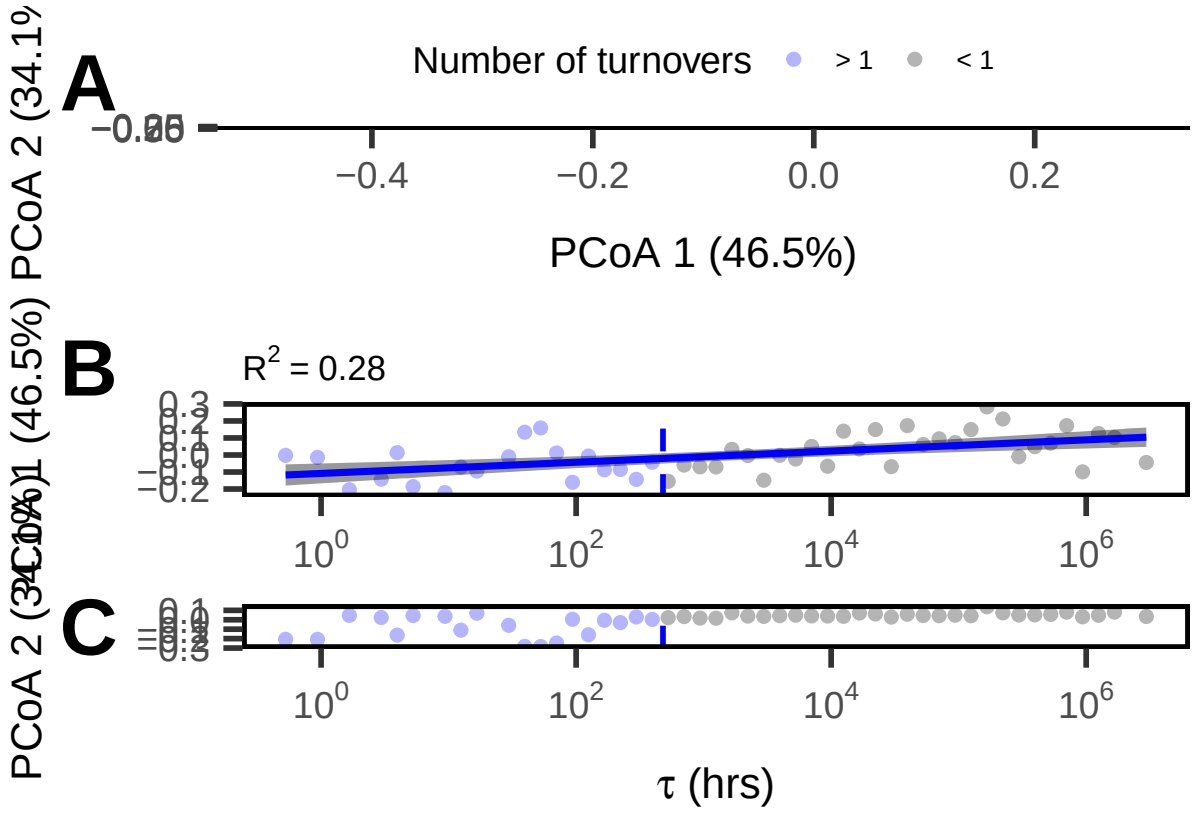

```
Dim1_sites <- EP.plot$V1
Dim2_sites <- EP.plot$V2

EP_corr <- as.data.frame(matrix(NA, dim(EPREL)[2], 4))
row.names(EP_corr) <- colnames(EPREL)
colnames(EP_corr)  <- c("V1_rho", "V1_p.value", "V2_rho", "V2_p.value")

for (i in 1:dim(EPREL)[2]){
  out.i <- cor.test(EPREL[,i], EP.plot$V1, method="spearman",
                    exact=FALSE)
  EP_corr[i,1] <- out.i$estimate
  EP_corr[i,2] <- out.i$p.value
```

```
  out.i <- cor.test(EPREL[,i], EP.plot$V2, method="spearman",
                    exact=FALSE)
  EP_corr[i,3] <- out.i$estimate
  EP_corr[i,4] <- out.i$p.value
}

EP_corr     <- na.omit(EP_corr)
EP_corr$V1_BH <- p.adjust(EP_corr$V1_p.value, method = "BH")
EP_corr$V2_BH <- p.adjust(EP_corr$V2_p.value, method = "BH")
EP_corr_V2 <- EP_corr[abs(EP_corr$V2_rho) > 0.7, ]
EP_corr_V1 <- EP_corr[abs(EP_corr$V1_rho) > 0.7, ]

V1.tax <- subset(OTU.tax, OTU == rownames(DNA_corr_V1)[1])
V2.tax <- subset(OTU.tax, OTU == rownames(DNA_corr_V2)[1])

x <- 2
while (x <= nrow(DNA_corr_V1)){
  V1.tax <- rbind(V1.tax, subset(OTU.tax, OTU == rownames(DNA_corr_V1)[x]))
  x <- x + 1
}
DNA_corr_V1 <- cbind(DNA_corr_V1, V1.tax)

x <- 2
while (x <= nrow(DNA_corr_V2)){
  V2.tax <- rbind(V2.tax, subset(OTU.tax, OTU == rownames(DNA_corr_V2)[x]))
  x <- x + 1
}

DNA_corr_V2 <- cbind(DNA_corr_V2, V2.tax)

DNA_corr_V2_neg <- subset(DNA_corr_V2, DNA_corr_V2$V2_rho < -0.7)
DNA_corr_V2_neg <- DNA_corr_V2_neg[,c("OTU","V2_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V2_neg, "./data/DNA_corr_V2_neg.csv")
DNA_corr_V2_pos <- subset(DNA_corr_V2, DNA_corr_V2$V2_rho > 0.7)
DNA_corr_V2_pos <- DNA_corr_V2_pos[,c("OTU","V2_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V2_pos, "./data/DNA_corr_V2_pos.csv")

DNA_corr_V1_neg <- subset(DNA_corr_V1, DNA_corr_V1$V1_rho < -0.7)
DNA_corr_V1_neg <- DNA_corr_V1_neg[,c("OTU","V1_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V1_neg, "./data/DNA_corr_V1_neg.csv")
DNA_corr_V1_pos <- subset(DNA_corr_V1, DNA_corr_V1$V1_rho > 0.7)
DNA_corr_V1_pos <- DNA_corr_V1_pos[,c("OTU","V1_rho", "Domain", "Phylum", "Class", "Order", "Family", "
write.csv(DNA_corr_V1_pos, "./data/DNA_corr_V1_pos.csv")

DNA_V2_family_neg <- data.frame()
for(x in unique(DNA_corr_V2_neg$Family)){
  DNA_V2_family_neg <- rbind(DNA_V2_family_neg, c(x, mean(subset(DNA_corr_V2_neg, DNA_corr_V2_neg$Family
}
colnames(DNA_V2_family_neg) <- c("Family", "rho", "n")
DNA_V2_family_neg$n <- as.numeric(DNA_V2_family_neg$n)
DNA_V2_family_neg$rho <- as.numeric(DNA_V2_family_neg$rho)

DNA_V2_family_pos <- data.frame()
```

```
for(x in unique(DNA_corr_V2_pos$Family)){
  DNA_V2_family_pos <- rbind(DNA_V2_family_pos, c(x, mean(subset(DNA_corr_V2_pos, DNA_corr_V2_pos$Family
}
colnames(DNA_V2_family_pos) <- c("Family", "rho", "n")
DNA_V2_family_pos$n <- as.numeric(DNA_V2_family_pos$n)
DNA_V2_family_pos$rho <- as.numeric(DNA_V2_family_pos$rho)

DNA_V1_family_neg <- data.frame()
for(x in unique(DNA_corr_V1_neg$Family)){
  DNA_V1_family_neg <- rbind(DNA_V1_family_neg, c(x, mean(subset(DNA_corr_V1_neg, DNA_corr_V1_neg$Family
}
colnames(DNA_V1_family_neg) <- c("Family", "rho", "n")
DNA_V1_family_neg$n <- as.numeric(DNA_V1_family_neg$n)
DNA_V1_family_neg$rho <- as.numeric(DNA_V1_family_neg$rho)

DNA_V1_family_pos <- data.frame()
for(x in unique(DNA_corr_V1_pos$Family)){
  DNA_V1_family_pos <- rbind(DNA_V1_family_pos, c(x, mean(subset(DNA_corr_V1_pos, DNA_corr_V1_pos$Family
}
colnames(DNA_V1_family_pos) <- c("Family", "rho", "n")
DNA_V1_family_pos$n <- as.numeric(DNA_V1_family_pos$n)
DNA_V1_family_pos$rho <- as.numeric(DNA_V1_family_pos$rho)
```

##Draw Figure

```
ggdraw()+
  draw_plot(EP_BC, x = 0, y = 0.625, width = 1, height = 0.375) +
  draw_plot(Res_PCoA1, x = 0, y = 0.325, width = 1, height = 0.3) +
  draw_plot(Res_PCoA2, x = 0, y = 0.0, width = 1, height = 0.325)+
  draw_plot_label(label = c("A", "B", "C"), size = 30, x = c(0.06,0.06,0.06), y = c(0.98,0.65,0.35))+
  theme(plot.background = element_rect(fill="white", color = NA))
```




```
ggsave("./output/CommRes.pdf")
```

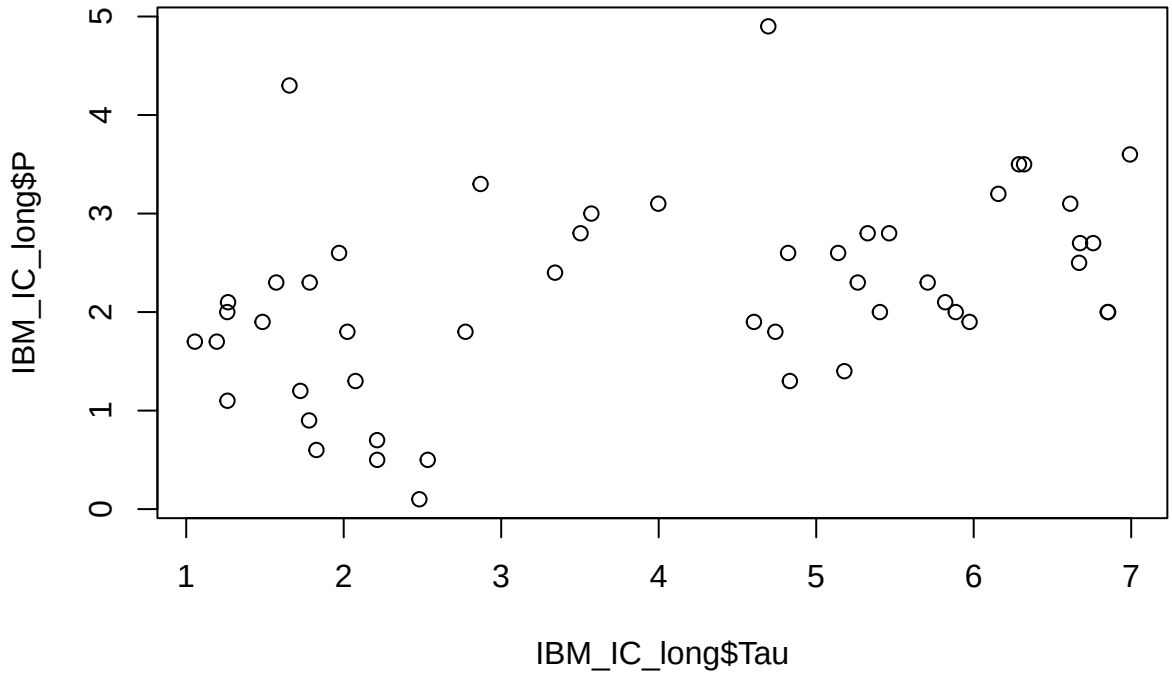
```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/CommRes.png", width = 5, height = 10)
```

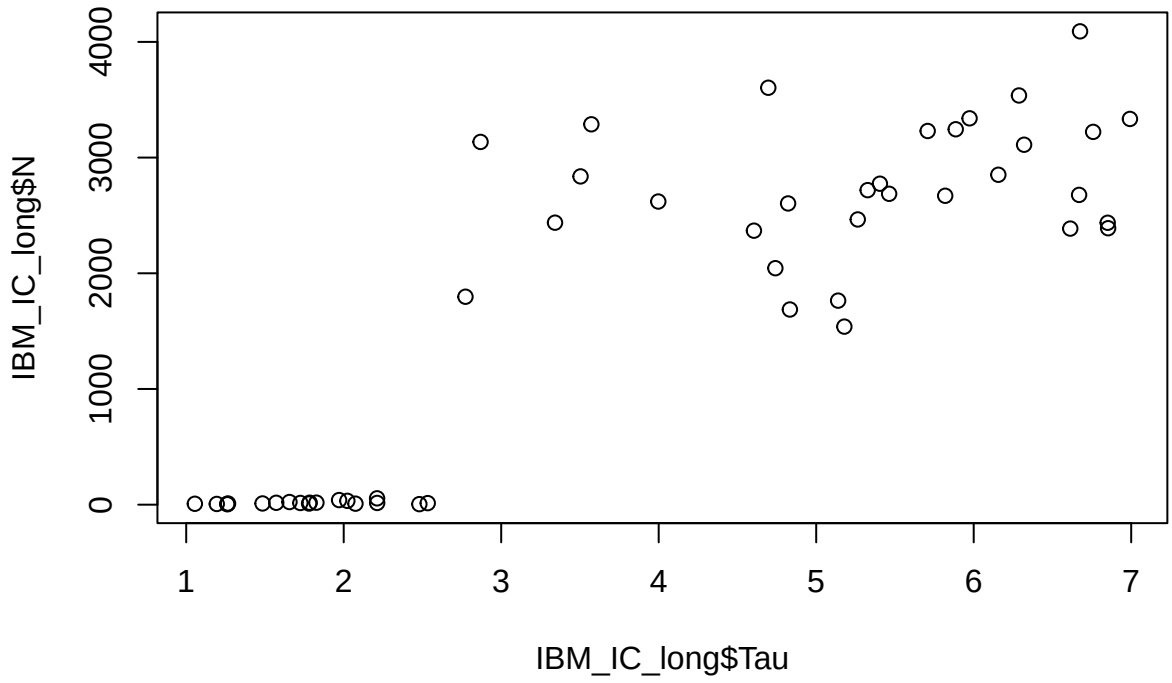
#Figure 7. IBMs

```
IBM_IC <- read.csv("../residence-time/Model/results/data/SimData.csv", header = TRUE, sep = ",")
IBM_IC_long <- as.data.frame(unique(IBM_IC$sim))
colnames(IBM_IC_long) <- c("sim")
IBM_IC_long$ct <- rep(1000, 50)

for(x in unique(IBM_IC_long$sim)){
  IBM_IC_long[IBM_IC_long$sim == x, "V"] <- mean(subset(IBM_IC, IBM_IC$sim == x)$V)
  IBM_IC_long[IBM_IC_long$sim == x, "Q"] <- mean(subset(IBM_IC, IBM_IC$sim == x)$Q)
  IBM_IC_long[IBM_IC_long$sim == x, "N"] <- mean(subset(IBM_IC, IBM_IC$sim == x)$total.abundance)
  IBM_IC_long[IBM_IC_long$sim == x, "R"] <- mean(subset(IBM_IC, IBM_IC$sim == x)$resource.particles)
  IBM_IC_long[IBM_IC_long$sim == x, "P"] <- mean(subset(IBM_IC, IBM_IC$sim == x)$ind.production)
  IBM_IC_long[IBM_IC_long$sim == x, "Dorm"] <- mean(subset(IBM_IC, IBM_IC$sim == x)$dormant.total.abunda
  IBM_IC_long[IBM_IC_long$sim == x, "S"] <- mean(subset(IBM_IC, IBM_IC$sim == x)$species.richness)
}

IBM_IC_long$N_turn <- IBM_IC_long$ct/(IBM_IC_long$V/IBM_IC_long$Q)
IBM_IC_long$turn <- IBM_IC_long$N_turn > 1
IBM_IC_long$ct_f <- as.factor(IBM_IC_long$ct)
IBM_IC_long$Tau <- log(IBM_IC_long$V/IBM_IC_long$Q, 10)
```

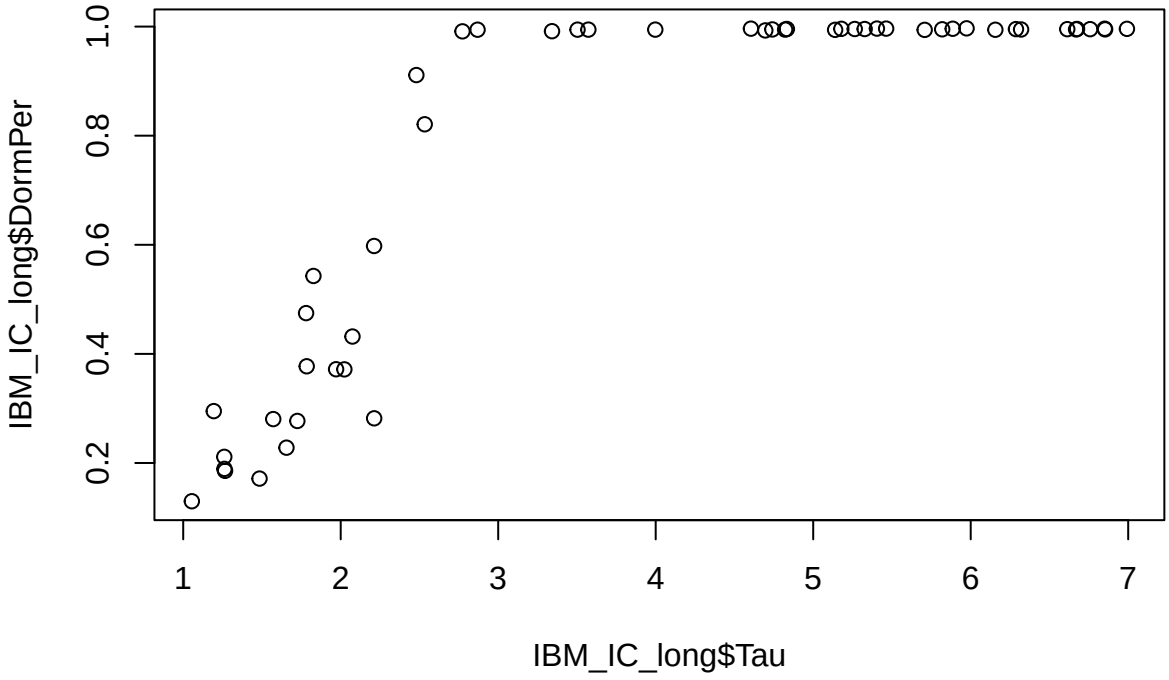
```r
IBM_IC_long$DormPer <- IBM_IC_long$Dorm/IBM_IC_long$N

plot(IBM_IC_long$P ~ IBM_IC_long$Tau)
```




```r
plot(IBM_IC_long$N ~ IBM_IC_long$Tau)
```




```
plot(IBM_IC_long$DormPer ~ IBM_IC_long$Tau)
```

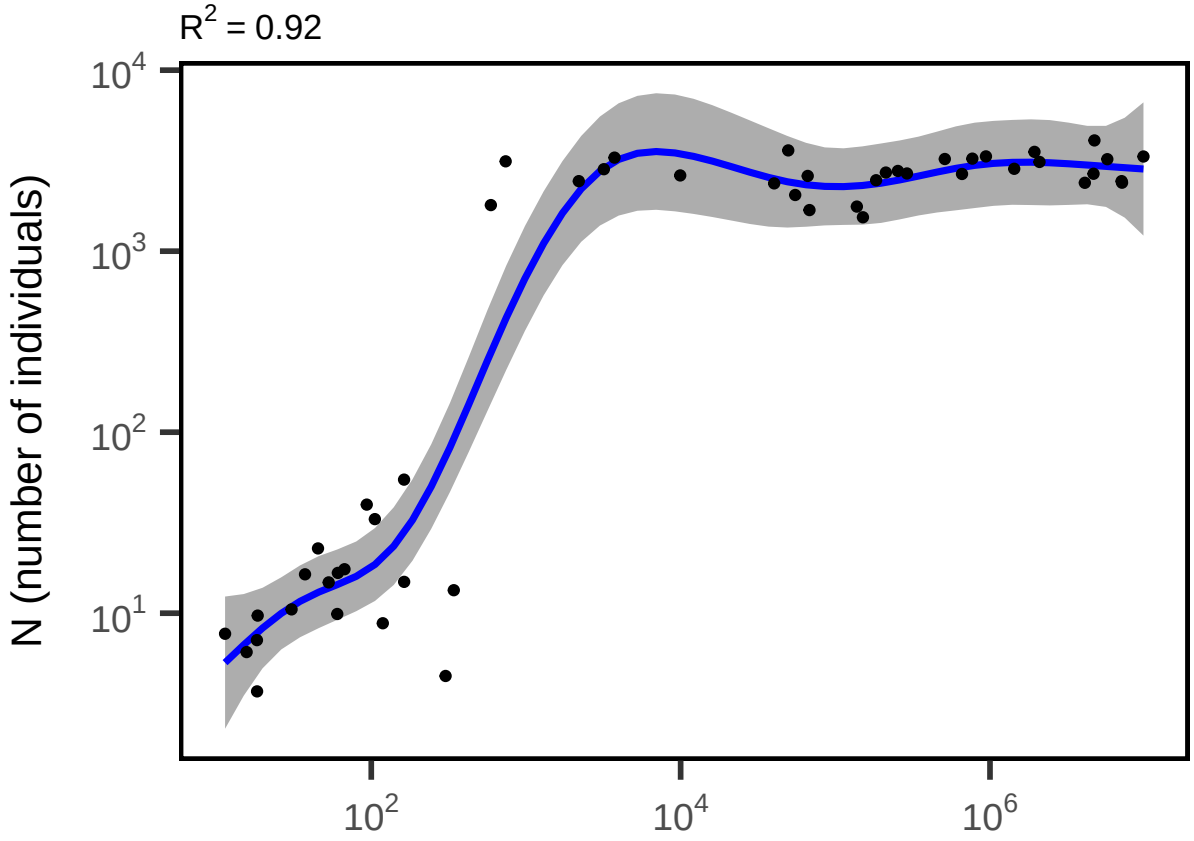


##IBM - N

```
N_IBM_gam <- gam(log(N, 10) ~ s(Tau), family = gaussian(link = "identity"), data = subset(IBM_IC_long,

summary(N_IBM_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(N, 10) ~ s(Tau)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.54566    0.04598   55.36   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(Tau) 6.275  7.405 76.65  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.921   Deviance explained = 93.1%
## -REML = 25.985  Scale est. = 0.10572   n = 50
```

```
N_IBM <- predict_gam(N_IBM_gam) %>%
  ggplot(aes(Tau, fit))+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = subset(IBM_IC_long, IBM_IC_long$N > 0), aes(x = Tau, y = log(N, 10)))+
  ylab("N (number of individuals)")+
  theme(axis.title.x = element_blank())+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(N_IBM_gam)$r.sq, 2))))

N_IBM
```




```
ggsave("./output/IBM_N.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/IBM_N.png", width = 6.5, height = 5)
```

#IBM - S

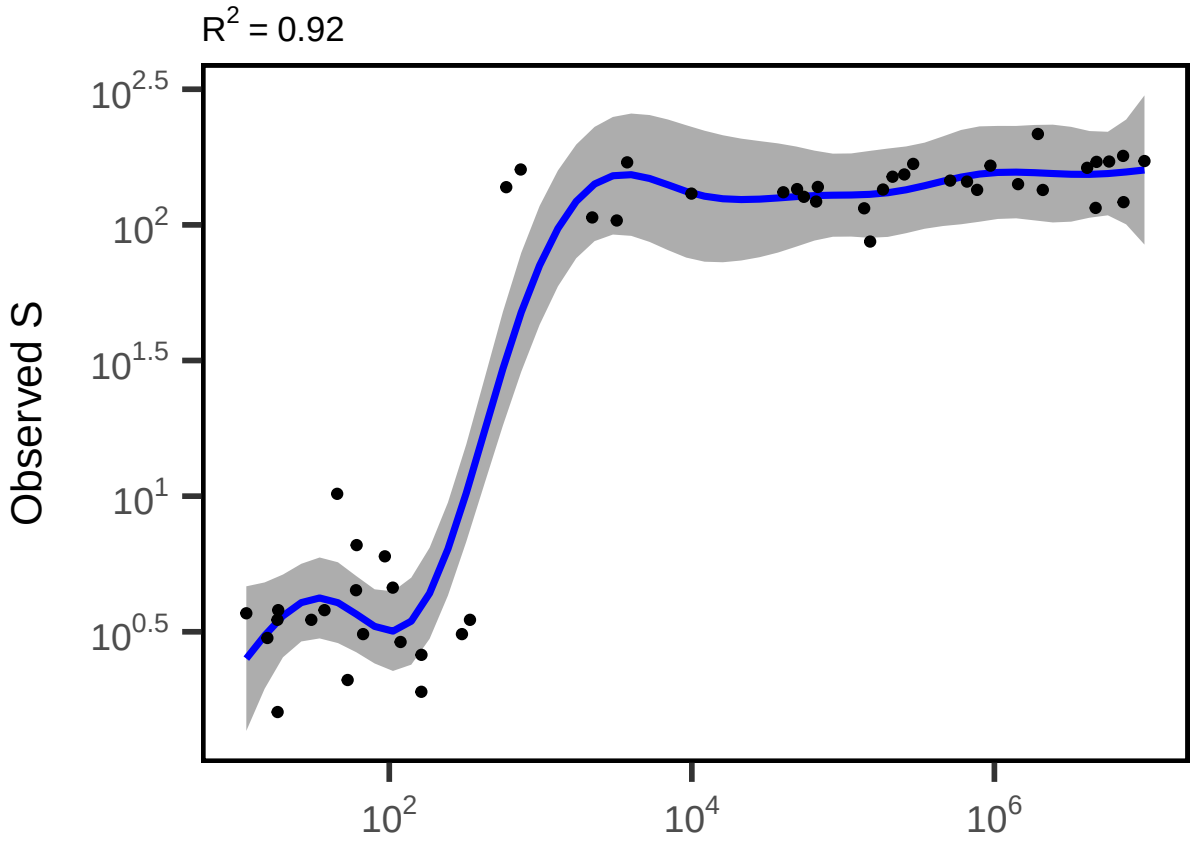
```
S_IBM_gam <- gam(log(S, 10) ~ s(Tau), family = gaussian(link = "identity"), data = subset(IBM_IC_long,
```

```
summary(S_IBM_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
```

```
## Formula:
## log(S, 10) ~ s(Tau)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.54105    0.03109   49.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(Tau) 7.477  8.416 70.07  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.924   Deviance explained = 93.5%
## -REML = 10.543  Scale est. = 0.04833   n = 50
```

```
S_IBM <- predict_gam(S_IBM_gam) %>%
  ggplot(aes(Tau, fit))+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = subset(IBM_IC_long, IBM_IC_long$S > 0), aes(x = Tau, y = log(S, 10)))+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force))+
  ylab("Observed S")+
  theme(axis.title.x = element_blank())+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(S_IBM_gam)$r.sq, 2))))

S_IBM
```




```
ggsave("./output/IBM_S.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/IBM_S.png", width = 6.5, height = 5)
```

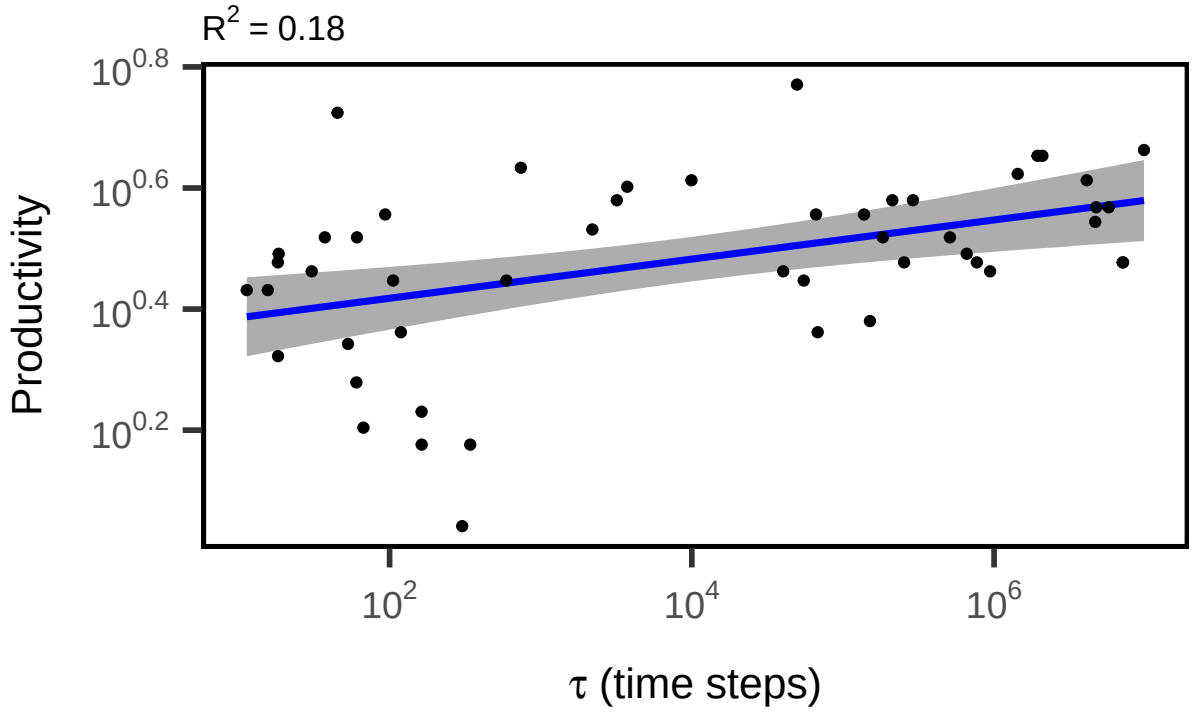
#IBM - P

```
P_IBM_gam <- gam(log(P + 1, 10) ~ s(Tau), family = gaussian(link = "identity"), data = IBM_IC_long, metl

summary(P_IBM_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(P + 1, 10) ~ s(Tau)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.48156    0.01878   25.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##        edf Ref.df     F p-value
## s(Tau)   1      1 11.76 0.00125 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.18   Deviance explained = 19.7%
## -REML = -24.88  Scale est. = 0.017641  n = 50
```

```
P_IBM <- predict_gam(P_IBM_gam) %>%
  ggplot(aes(Tau, fit))+
  geom_smooth_ci(color = "blue", cex = 1.25, ci_alpha = 0.4)+
  geom_point(data = IBM_IC_long, aes(x = Tau, y = log(P + 1, 10)))+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force))+
  ylab("Productivity")+
  xlab(expression(paste(tau, " (time steps)")))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(P_IBM_gam)$r.sq, 2))))+
  theme(plot.margin = unit(c(1.5, 0.2, 0, 0.2), "cm"))

P_IBM
```




```
ggsave("./output/IBM_P.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/IBM_P.png", width = 6.5, height = 5)
```

#Figure S1. ##Total number of resources used at 48 hours

```
NR_lm <- lm(NumRes ~ Tau, data = Tau)
NR_lm_re <- lmer(NumRes ~ Tau + (1|Set), data = Tau)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
NR_lm_2 <- lm(NumRes ~ poly(Tau, 2, raw = TRUE), data = Tau)
summary(NR_lm)
```

```
##
## Call:
## lm(formula = NumRes ~ Tau, data = Tau)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2767 -1.2890  0.7017  1.7186  2.7273
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.301054   0.689976  41.017   <2e-16 ***
## Tau         -0.004654   0.182411  -0.026     0.98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.37 on 47 degrees of freedom
## Multiple R-squared:  1.385e-05,  Adjusted R-squared:  -0.02126
## F-statistic: 0.0006511 on 1 and 47 DF,  p-value: 0.9798
```

```
AIC(NR_lm)
```

```
## [1] 227.5776
```

```
AIC(NR_lm_2)
```

```
## [1] 229.3136
```

```
cor(Tau$Tau, Tau$NumRes)
```

```
## [1] -0.003721891
```

```
anova(NR_lm_re, NR_lm_2, NR_lm)
```

```
## refitting model(s) with ML (instead of REML)
```

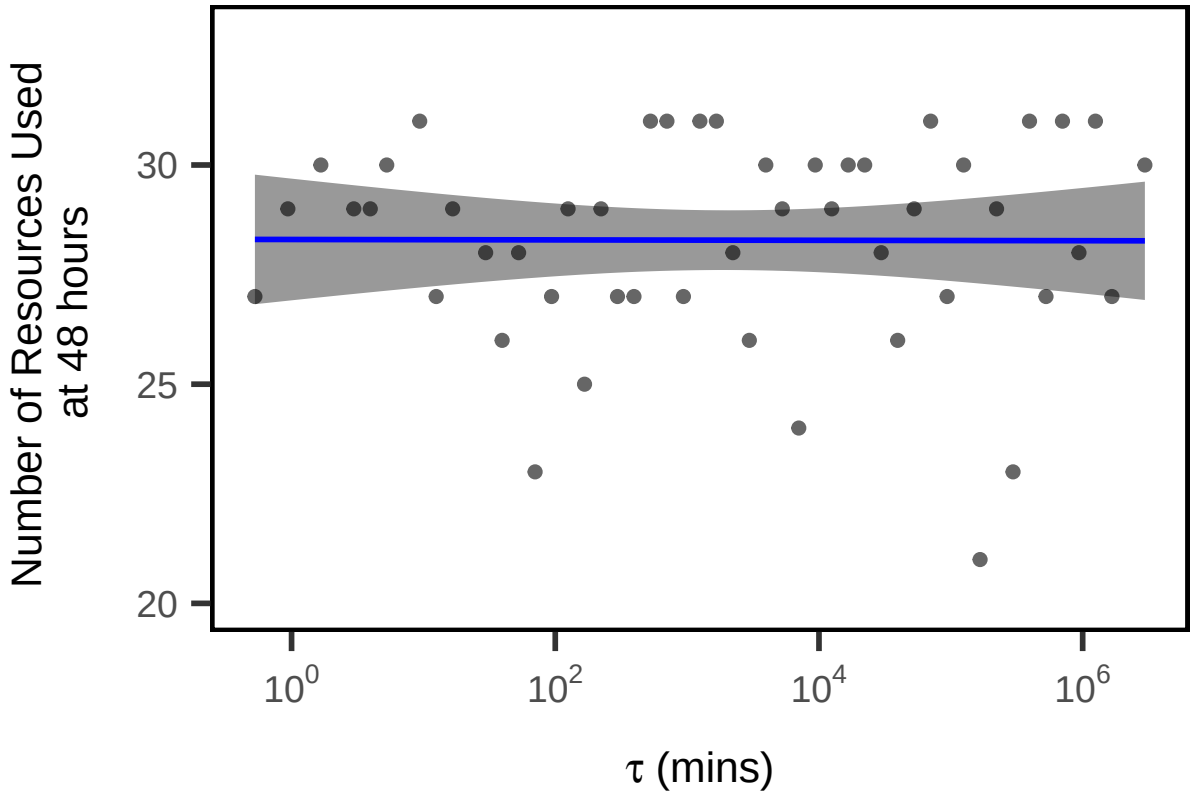
```
## Data: Tau
## Models:
## NR_lm: NumRes ~ Tau
## NR_lm_re: NumRes ~ Tau + (1 | Set)
## NR_lm_2: NumRes ~ poly(Tau, 2, raw = TRUE)
##          npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
## NR_lm       3 227.58 233.25 -110.79   221.58
## NR_lm_re    4 229.58 237.15 -110.79   221.58 0.000  1          1
## NR_lm_2     4 229.31 236.88 -110.66   221.31 0.264  0
```

```
NumRes <- ggplot(Tau, aes(x = Tau, y = NumRes))+
  geom_point(size = 2, alpha = 0.6)+
  ylim(c(20, 33))+
  xlab(expression(paste(tau, " (mins)")))+
  ylab("Number of Resources Used \n at 48 hours")+
```

```
  geom_smooth(method = "lm", formula = y~x, color = "blue", fill = alpha("black", 0.4))+
  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
               formula = y~x, parse = TRUE, size = 4)+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))


NumRes
```

```
## Warning in ci_f_ncp(stat, df1 = df1, df2 = df2, probs = probs): Upper limit
## outside search range. Set to the maximum of the parameter range.
```

```
## Warning: Computation failed in `stat_poly_eq()`
## Caused by error in `check_output()`:
## ! out[1] <= out[2] is not TRUE
```




```
ggsave("./output/RTLC_NumRes.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning in ci_f_ncp(stat, df1 = df1, df2 = df2, probs = probs): Upper limit
## outside search range. Set to the maximum of the parameter range.

## Warning in ci_f_ncp(stat, df1 = df1, df2 = df2, probs = probs): Computation
## failed in `stat_poly_eq()`
```

```
ggsave("./output/RTLC_NumRes.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning in ci_f_ncp(stat, df1 = df1, df2 = df2, probs = probs): Upper limit
## outside search range. Set to the maximum of the parameter range.

## Warning in ci_f_ncp(stat, df1 = df1, df2 = df2, probs = probs): Computation
## failed in `stat_poly_eq()`
```

#Pairwise OTU dist vs. Res dist

```
test.pw <- data.frame(site1=numeric(),
                      site2=numeric(),
                      otupw=numeric(),
                       ecopw=numeric())
rownames(EcoPlate) <- EcoPlate_env$Tau
x <- 2.5
y <- 3
for(x in row.names(OTUsr_20)){
  for(y in row.names(OTUsr_20)){
    test.pw <- rbind(test.pw, c(x, y, vegdist(OTUsr_20[c(x,y),], method = "bray", upper = TRUE, diag = 
  }
}

colnames(test.pw) <- c("site1", "site2", "otupw", "ecopw")
test.pw$site1 <- as.numeric(test.pw$site1)
test.pw$site1 <- log(((10^test.pw$site1)/60), 10)
test.pw$site2 <- as.numeric(test.pw$site2)
test.pw$site2 <- log(((10^test.pw$site2)/60), 10)
test.pw$otupw <- as.numeric(test.pw$otupw)
test.pw$ecopw <- as.numeric(test.pw$ecopw)
test.pw$taudiff <- test.pw$site2 - test.pw$site1
test.pw[1, "site1"]
```

```
## [1] 0.7218487
```

```
for(row in row(test.pw)){
  if((test.pw[row, "site1"] > 2.6 && test.pw[row, "site2"] > 2.6)){
    test.pw[row, "Turn"] <- "Less"
  }
  else if((test.pw[row, "site1"] < 2.6 && test.pw[row, "site2"] < 2.6)){
    test.pw[row, "Turn"] <- "More"
  }
  else{
    test.pw[row, "Turn"] <- NA
  }
}

pw_lm <- lm(otupw ~ ecopw * Turn, data = subset(test.pw, is.na(test.pw$Turn) == FALSE))
AIC(pw_lm)
```

```
## [1] -1125.702
```

```
summary(pw_lm)
```

```
##
## Call:
## lm(formula = otupw ~ ecopw * Turn, data = subset(test.pw, is.na(test.pw$Turn) ==
##     FALSE))
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32646 -0.02963  0.00962  0.05239  0.28715
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.229689   0.010676  21.515  < 2e-16 ***
## ecopw          0.057572   0.006883   8.364 4.01e-16 ***
## TurnMore       0.096774   0.018218   5.312 1.51e-07 ***
## ecopw:TurnMore 0.031114   0.009201   3.381 0.000766 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09785 on 621 degrees of freedom
## Multiple R-squared:  0.6028, Adjusted R-squared:  0.6009
## F-statistic: 314.2 on 3 and 621 DF,  p-value: < 2.2e-16
```

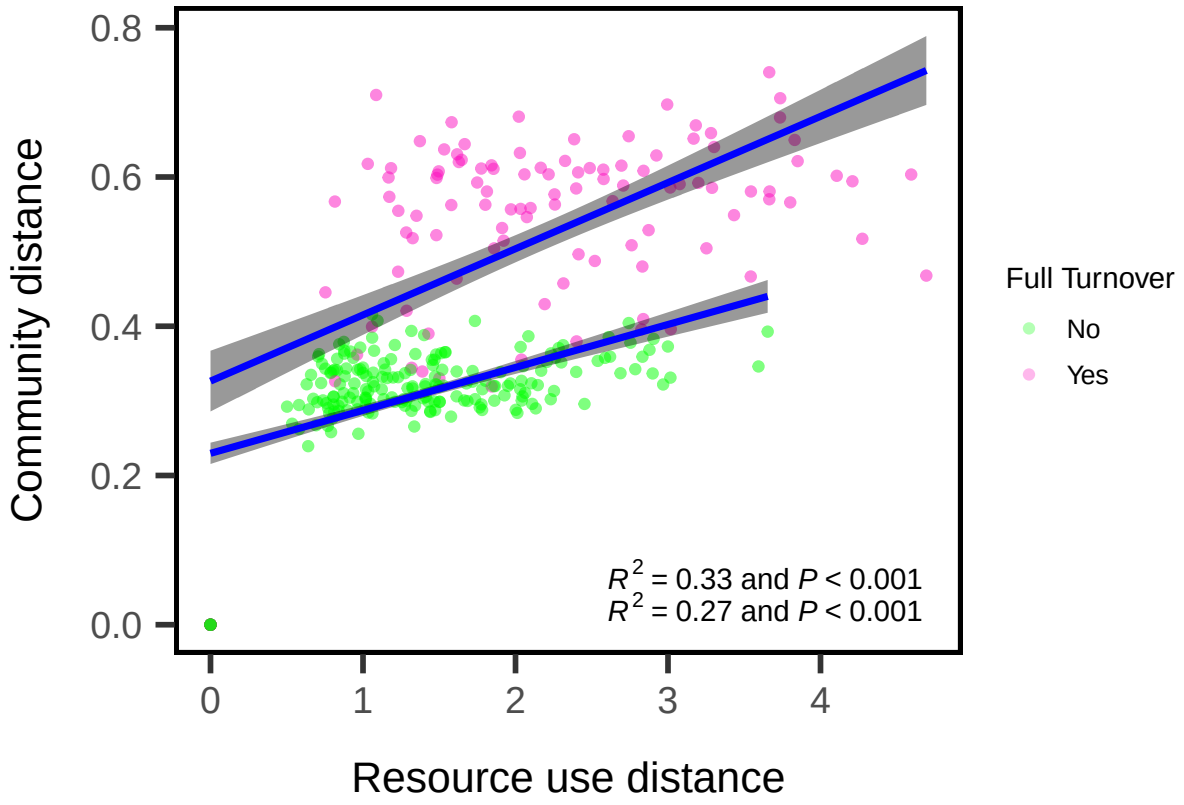
```
anova
```

```
## function (object, ...)
## UseMethod("anova")
## <bytecode: 0x00000254c7bca400>
## <environment: namespace:stats>
```

```
mantel(tau.db, EP.db.20)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = tau.db, ydis = EP.db.20)
##
## Mantel statistic r: 0.5564
##       Significance: 0.001
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.132 0.176 0.208 0.245
## Permutation: free
## Number of permutations: 999
```

```
pw.bc <- ggplot(data = subset(test.pw, is.na(test.pw$Turn) == FALSE), aes(x = ecopw, y = otupw, group =
  geom_point(aes(color = Turn))+
  xlab("Resource use distance")+
  ylab("Community distance")+
  scale_color_manual(values = c(alpha("green",0.3), alpha("#FC0FC0", 0.3)), labels = c("No", "Yes"))+
  labs(color = "Full Turnover")+
  geom_smooth(method = "lm", formula = y ~ x, color = "blue", cex = 1.25, fill = alpha("black", 0.4))+
  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
               formula = y~x, parse = TRUE, size = 4, label.x = "right", label.y= "bottom")

pw.bc
```




```
ggsave("./output/RTLC_BC_PW.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_BC_PW.png")
```

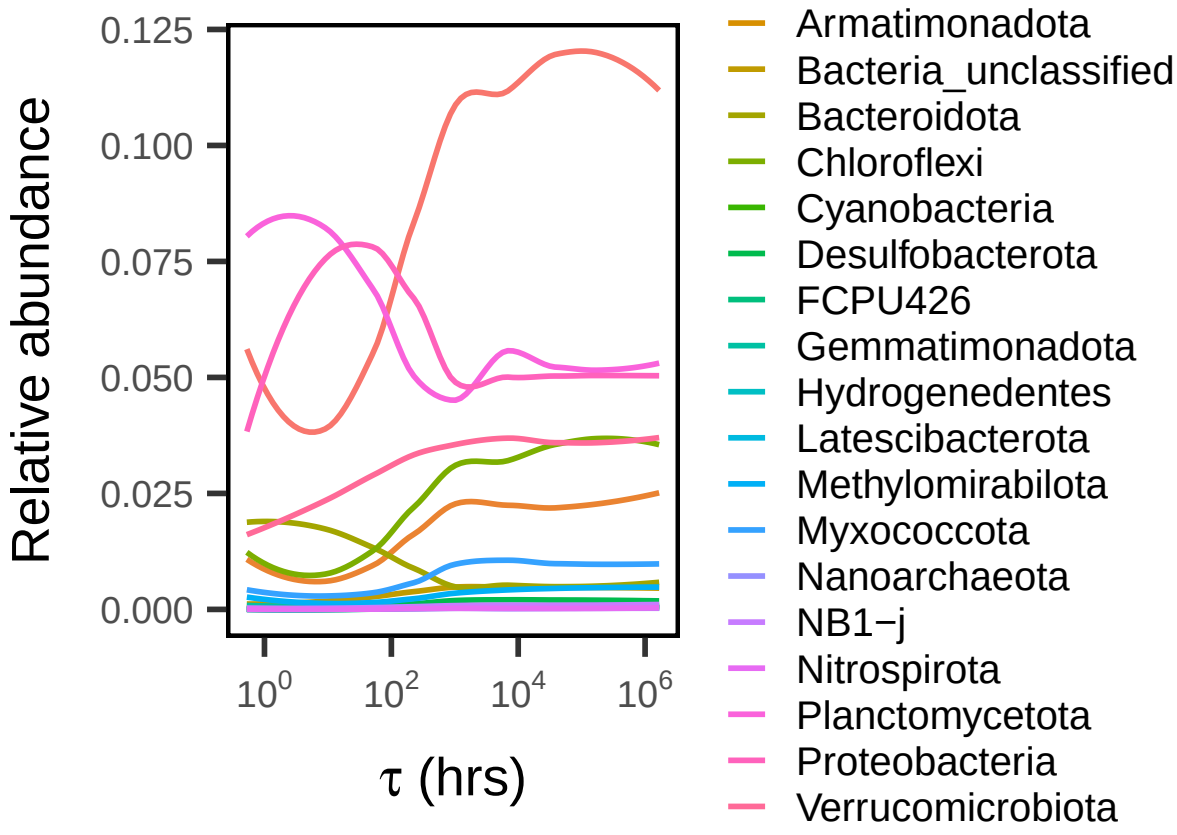
```
## Saving 6.5 x 4.5 in image
```

#Unused figures

```
for(x in unique(imp.spp$Phylum)){
  list <- data.frame(rep(0, 35))
  colnames(list) <- c(as.character(x))
  for(y in imp.spp$OTU){
    if(imp.spp[y, "Phylum"] == x){
      list[,x] <- list[,x] + as.data.frame(imp.otus[,y])
    }
  }
  imp.otus[,as.character(x)] <- list[,x]
}

imp.phylum <- cbind(imp.otus$Tau, imp.otus[ ,338:357])
colnames(imp.phylum) <- c("Tau", colnames(imp.otus[,338:357]))
imp.phylum.long <- gather(imp.phylum, phylum, REL, Proteobacteria:Hydrogenedentes)

imp.phylum.long$Tau <- as.numeric(imp.phylum.long$Tau)

imp_phylum_plot <- ggplot(imp.phylum.long, aes(x = Tau, y = REL, group = phylum))+
```

```
  geom_smooth(method = "loess", se = F, aes(color = phylum))+
  ylab("Relative abundance")+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  labs(color = "Phylum")+
  xlab(expression(paste(tau, " (hrs)")))+
  theme(legend.title = element_text(size = 20), legend.text = element_text(size = 15), axis.title = ele
imp_phylum_plot
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggsave("./output/RTLC_imp_phylum.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

```
ggsave("./output/RTLC_imp_phylum.png", width = 12, height = 8)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

##Individual Productivity (uM C/hr/Cell)

```
IP_lm <- lm(ind_P ~ Tau, data = Tau)
IP_lm_2 <- lm(log(ind_P, 10) ~ poly(Tau, 2, raw = TRUE), data = Tau)
IP_exp <- lm(log(ind_P, 10) ~ Tau, data = Tau)
summary(IP_lm)
```

```
##
## Call:
```

```
## lm(formula = ind_P ~ Tau, data = Tau)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.370e-09 -4.214e-09 -8.142e-10  2.701e-09  1.501e-08
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.390e-08  1.806e-09   7.700 5.92e-09 ***
## Tau         -2.918e-09  4.748e-10  -6.146 5.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.27e-09 on 34 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.5263, Adjusted R-squared:  0.5124
## F-statistic: 37.77 on 1 and 34 DF,  p-value: 5.568e-07
```

```
summary(IP_lm_2)
```

```
##
## Call:
## lm(formula = log(ind_P, 10) ~ poly(Tau, 2, raw = TRUE), data = Tau)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64692 -0.40112  0.03826  0.30816  0.90367
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -7.35852    0.21690 -33.926  < 2e-16 ***
## poly(Tau, 2, raw = TRUE)1 -0.90428    0.14912  -6.064 7.99e-07 ***
## poly(Tau, 2, raw = TRUE)2  0.08085    0.02257   3.583  0.00108 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.458 on 33 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.7566, Adjusted R-squared:  0.7419
## F-statistic: 51.29 on 2 and 33 DF,  p-value: 7.485e-11
```

```
summary(IP_exp)
```

```
##
## Call:
## lm(formula = log(ind_P, 10) ~ Tau, data = Tau)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9249 -0.2434  0.0704  0.2583  0.9861
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.89504    0.18219 -43.335  < 2e-16 ***
## Tau         -0.39088    0.04791  -8.159 1.62e-09 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5318 on 34 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.6619, Adjusted R-squared:  0.652
## F-statistic: 66.57 on 1 and 34 DF,  p-value: 1.621e-09
```

```
AIC(IP_lm)
```

```
## [1] -1266.297
```

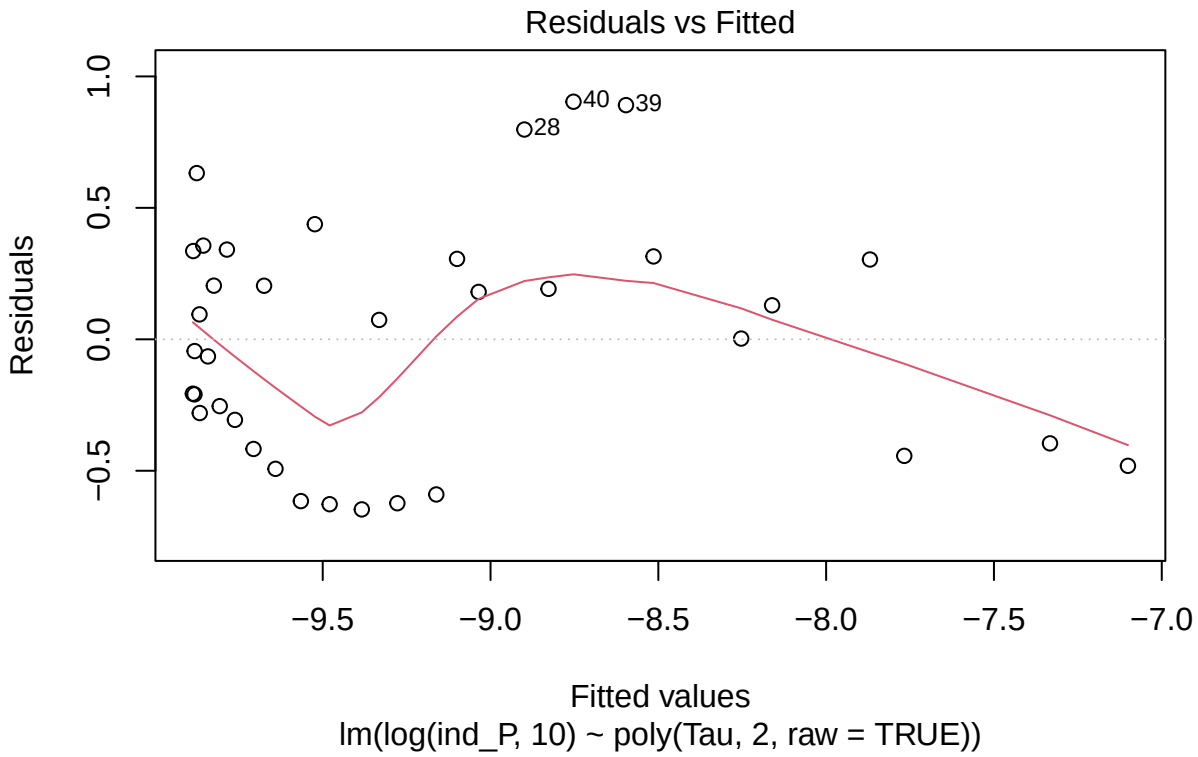
```
AIC(IP_lm_2)
```

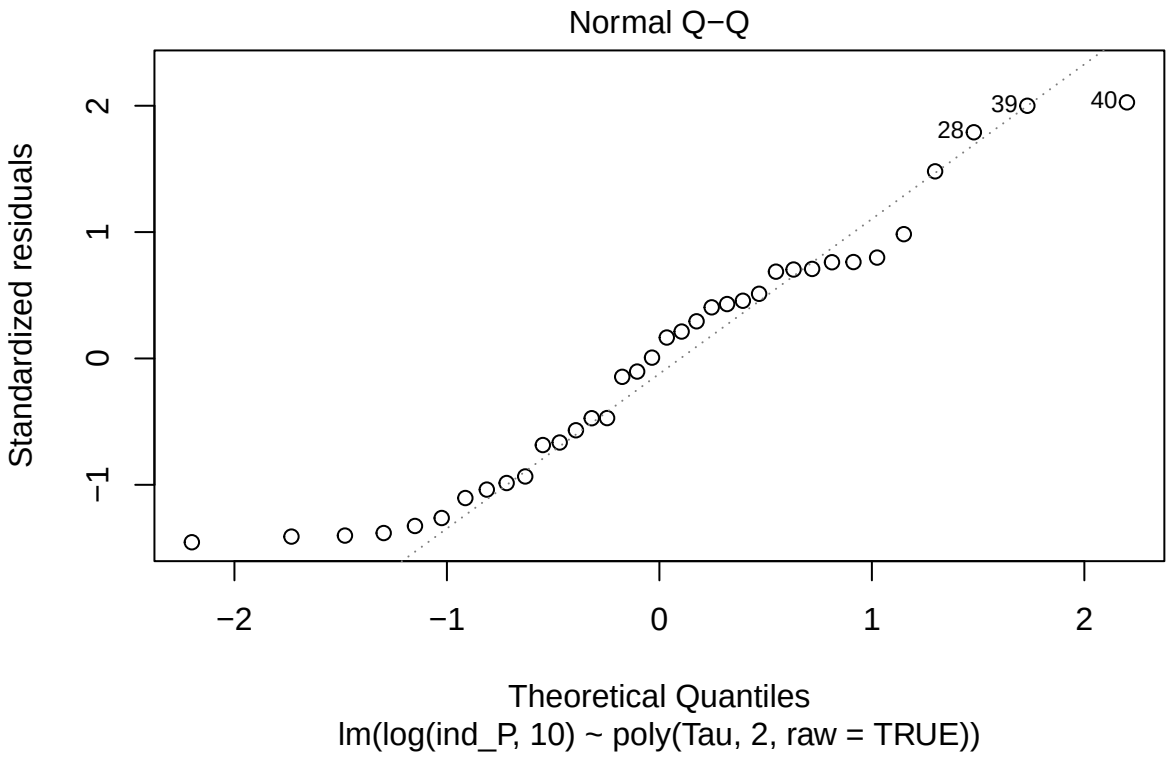
```
## [1] 50.80736
```

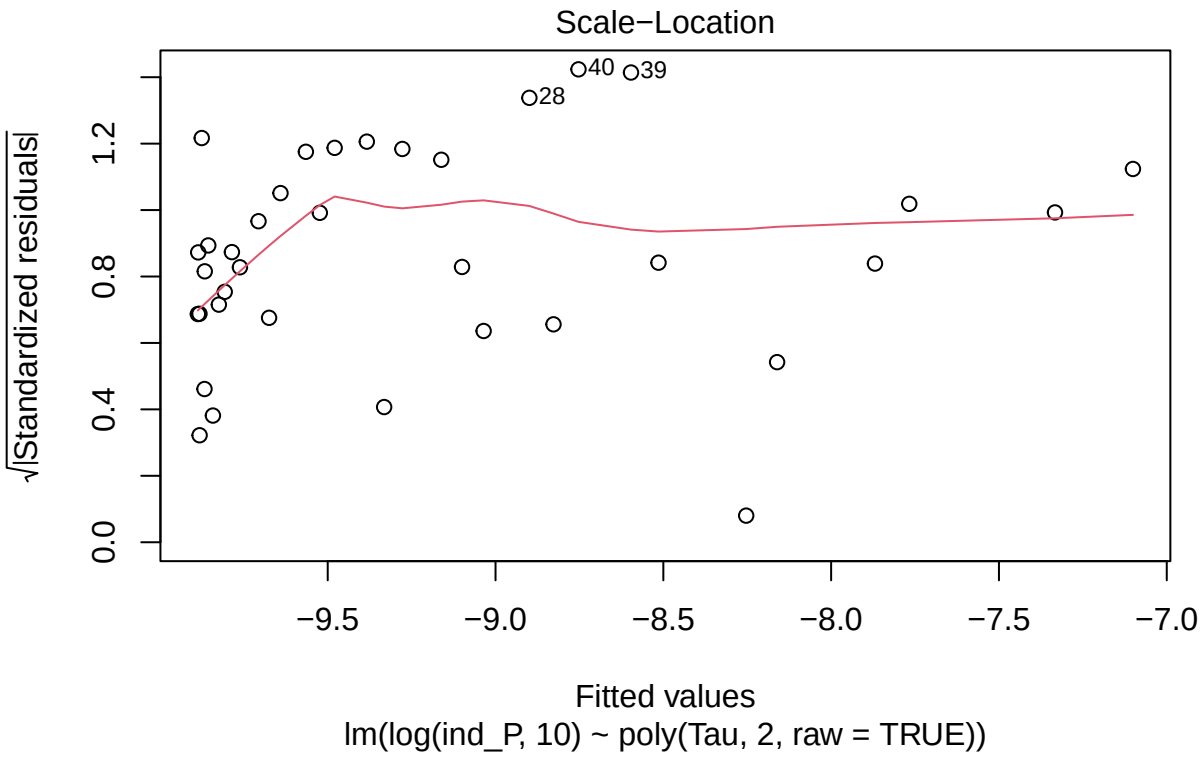
```
AIC(IP_exp)
```

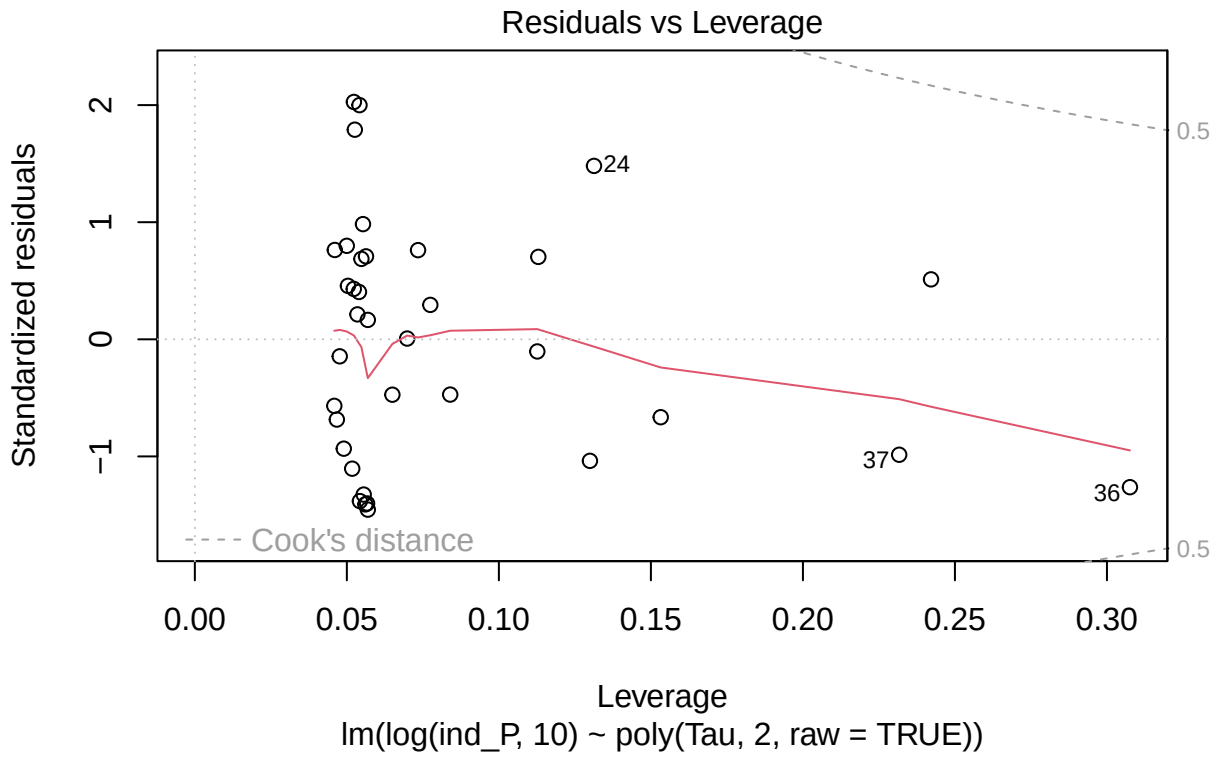
```
## [1] 60.63569
```

```
plot(IP_lm_2)
```

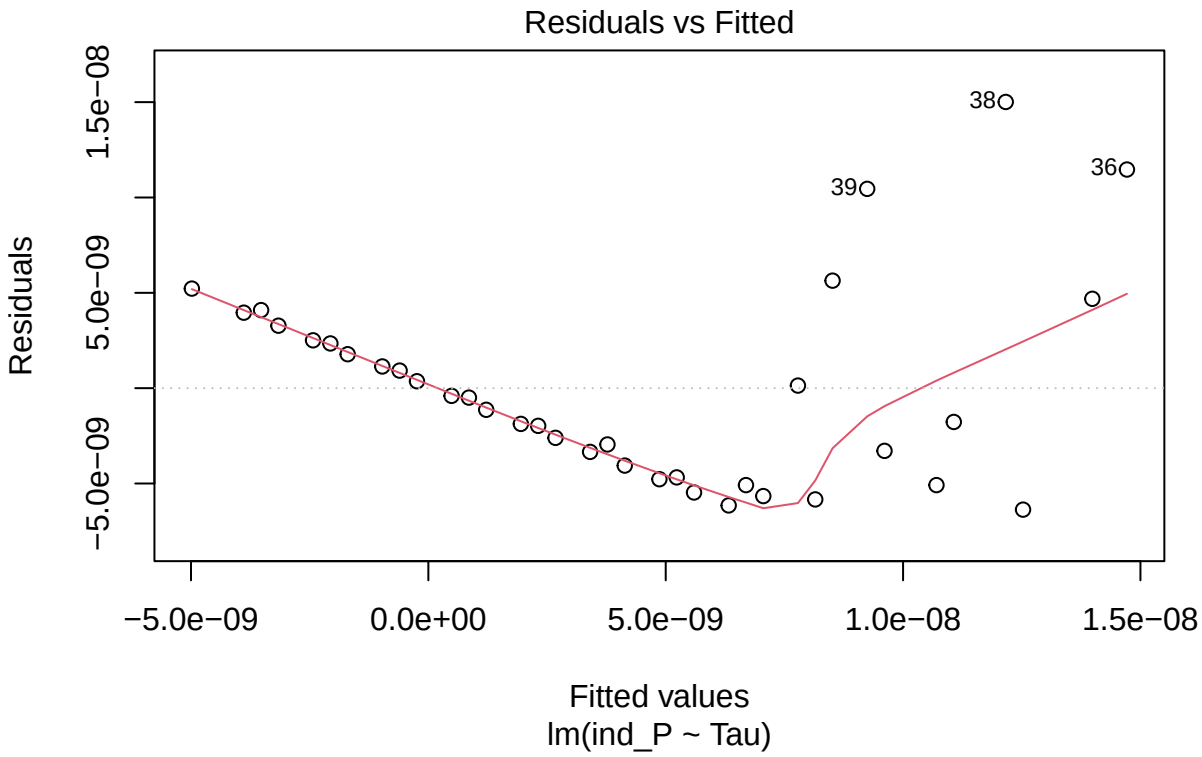


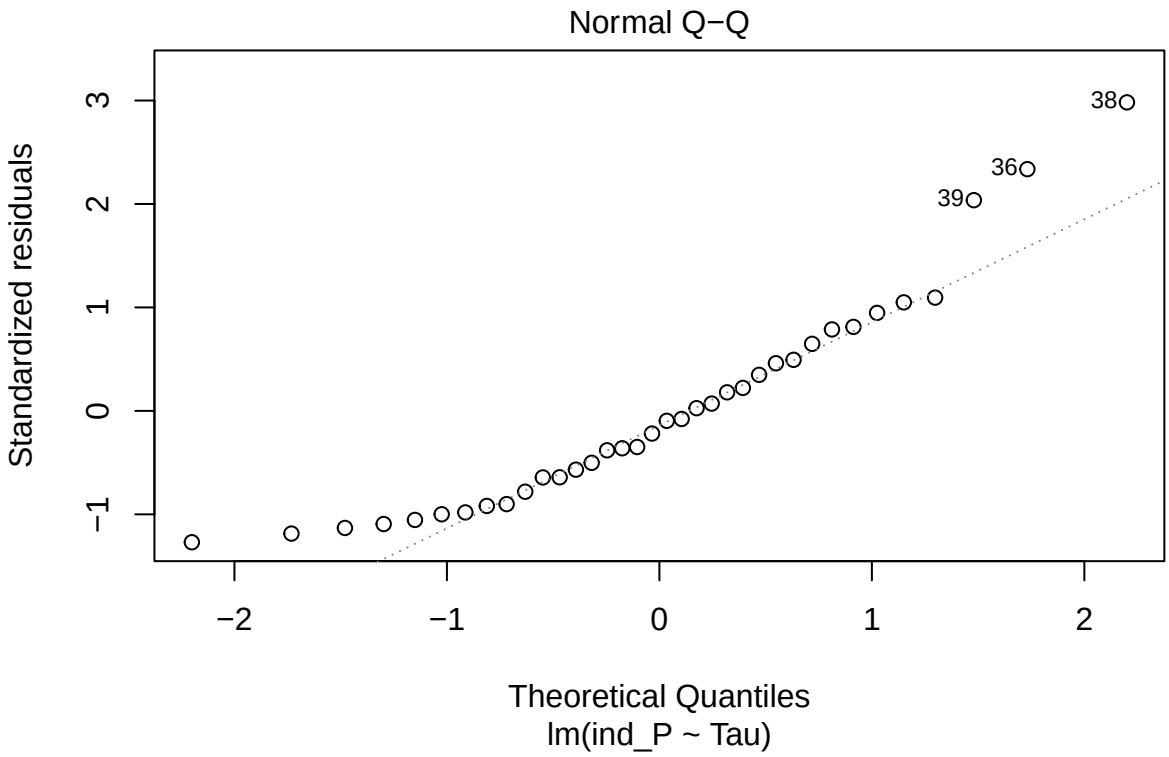

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(ind_P, 10) ~ poly(Tau, 2, raw = TRUE))
40
39
28

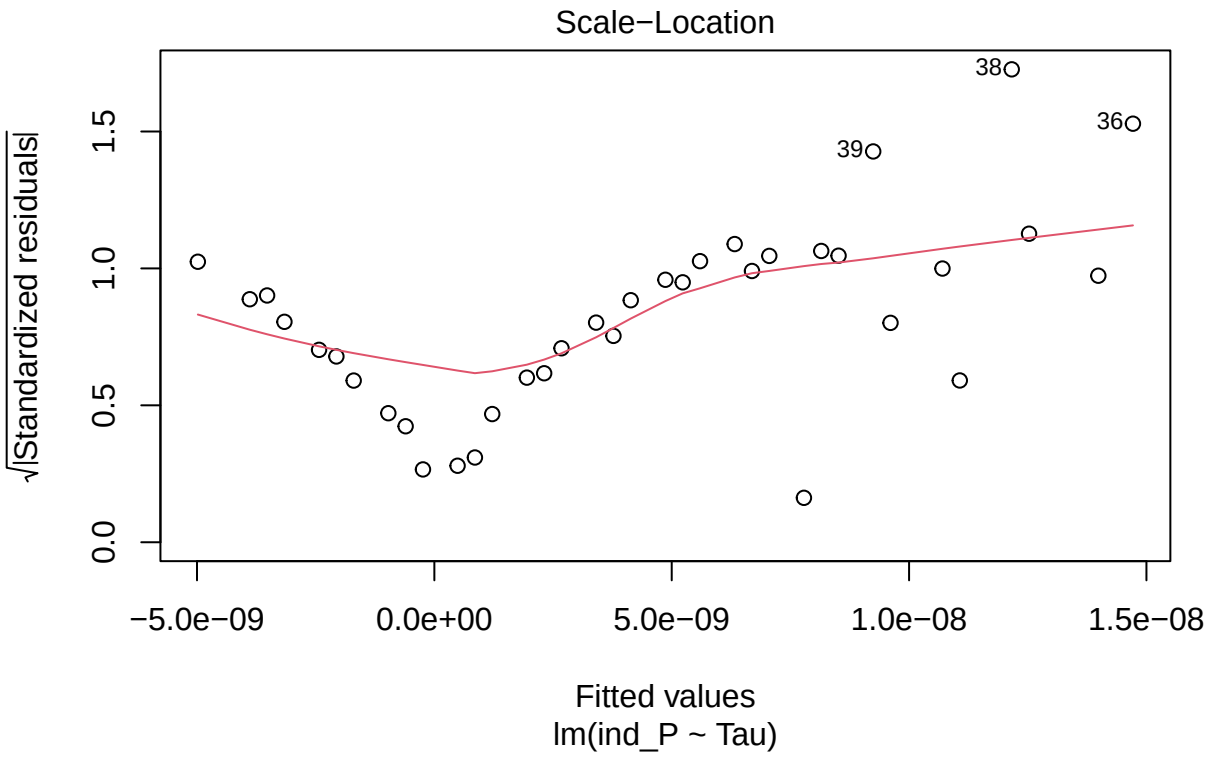

Scale−Location
40
39
28
√|Standardized residuals|
0.0
0.4
0.8
1.2
−9.5
−9.0
−8.5
−8.0
−7.5
−7.0
Fitted values
lm(log(ind_P, 10) ~ poly(Tau, 2, raw = TRUE))

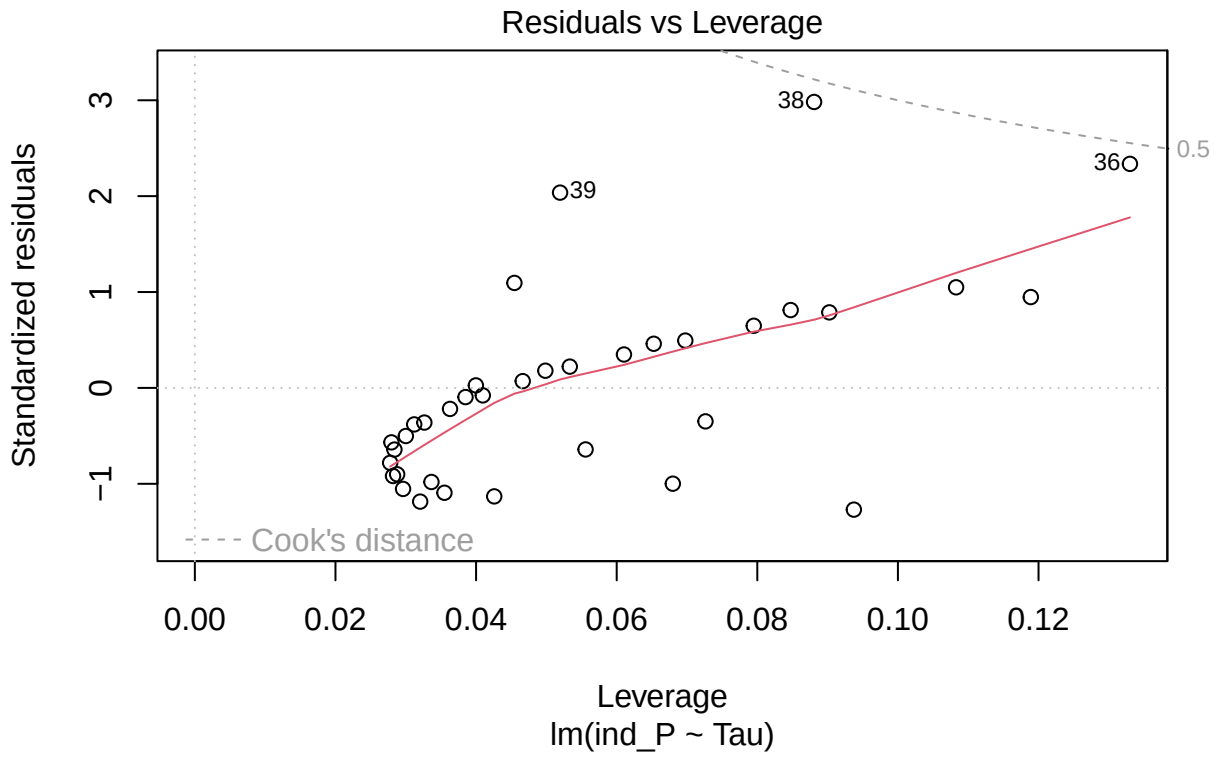

Residuals vs Leverage
Standardized residuals
2
1
0
−1
24
37
36
0.5
0.5
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Leverage
lm(log(ind_P, 10) ~ poly(Tau, 2, raw = TRUE))

```
plot(IP_lm)
```

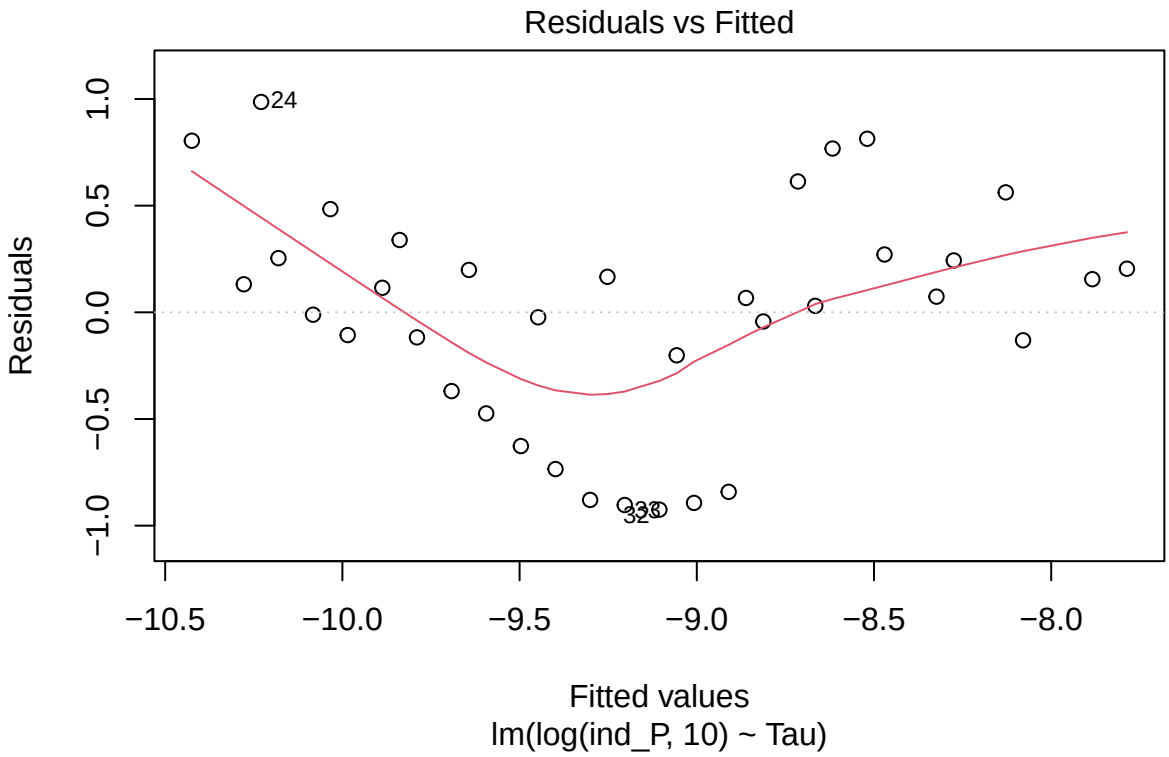

Residuals vs Fitted
Residuals
1.5e−08
5.0e−09
−5.0e−09
38
36
39
−5.0e−09
0.0e+00
5.0e−09
1.0e−08
1.5e−08
Fitted values
lm(ind_P ~ Tau)

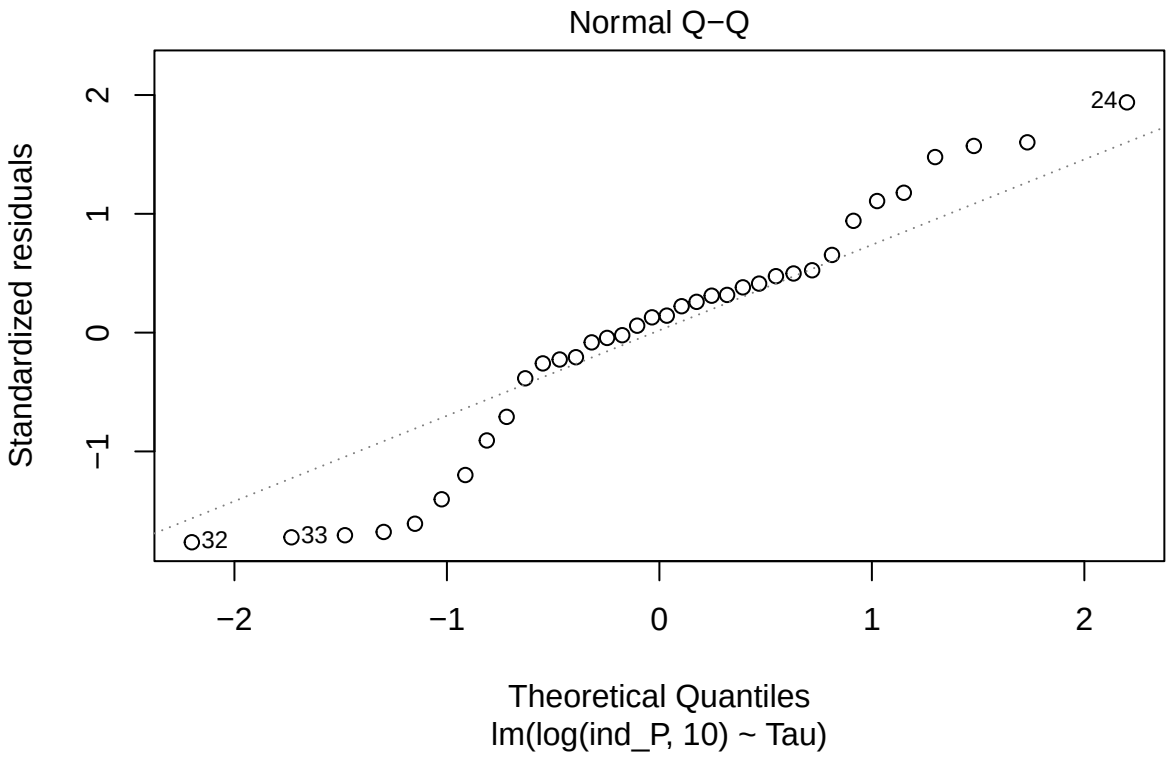

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(ind_P ~ Tau)
38
36
39
−2
−1
0
1
2
−1
0
1
2
3

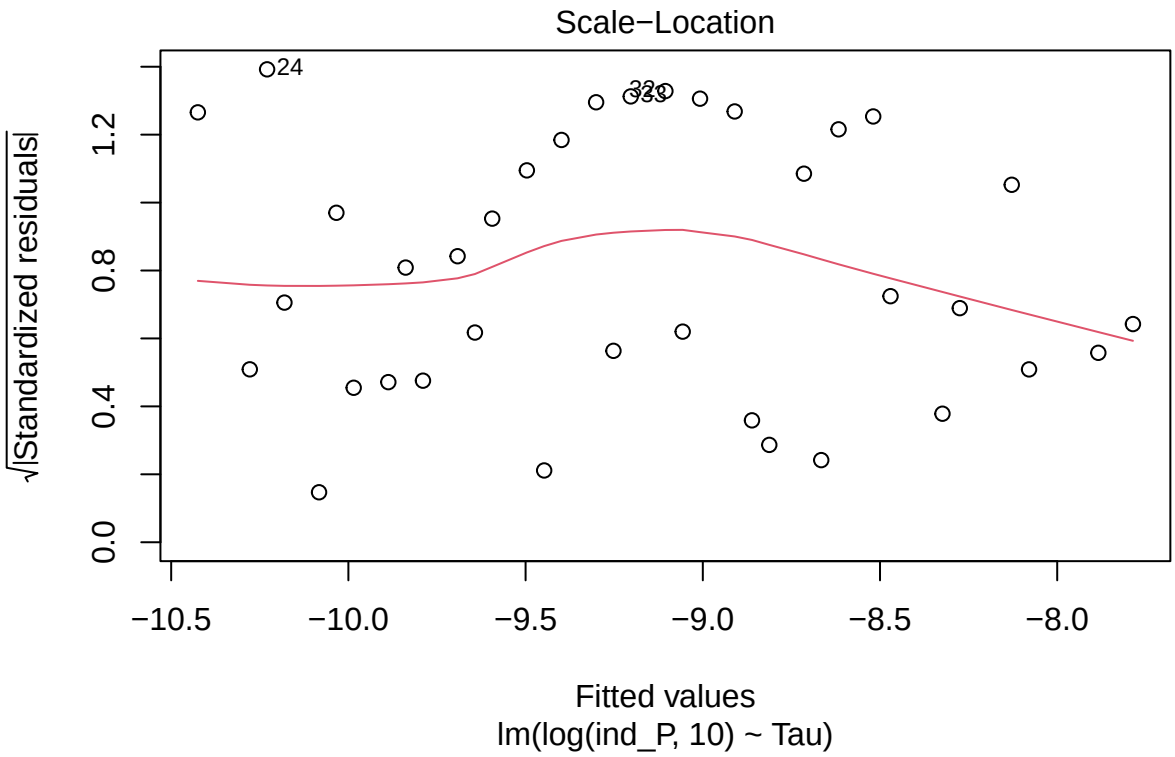

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
38
36
39
−5.0e−09
0.0e+00
5.0e−09
1.0e−08
1.5e−08
Fitted values
lm(ind_P ~ Tau)

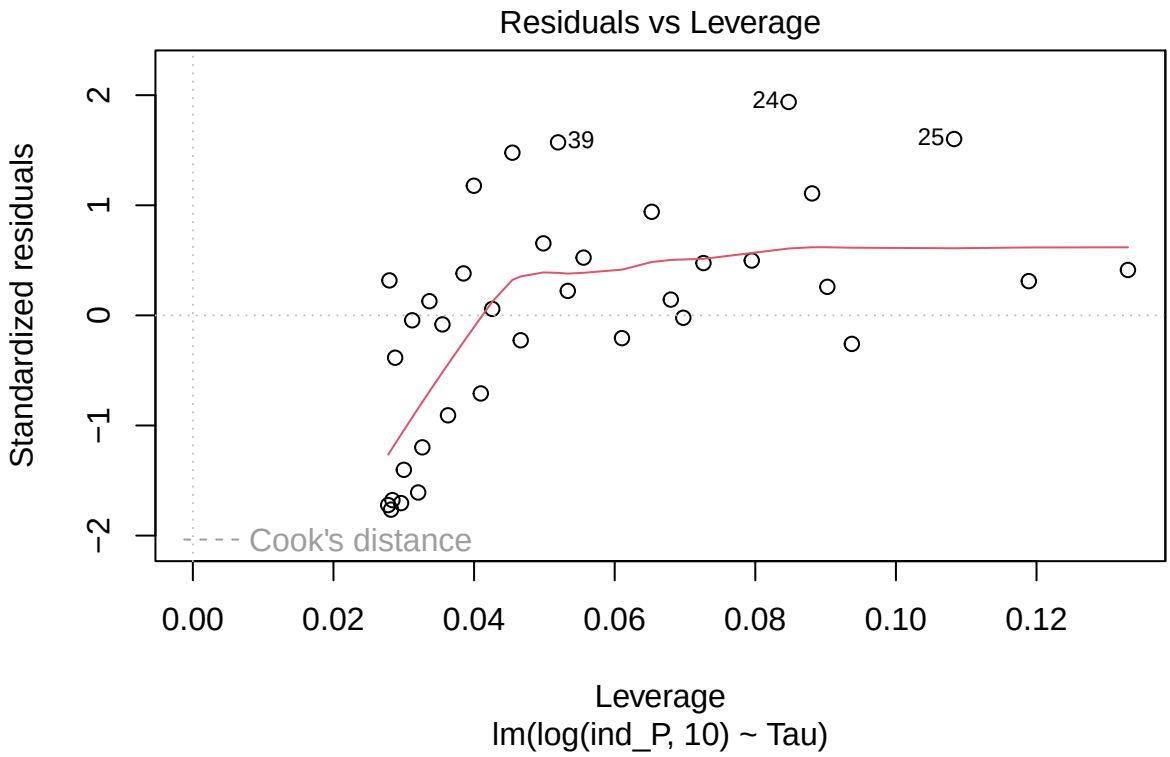

Residuals vs Leverage
Standardized residuals
Leverage
lm(ind_P ~ Tau)
Cook's distance
0.5
38
39
36
0.00
0.02
0.04
0.06
0.08
0.10
0.12
−1
0
1
2
3

```
plot(IP_exp)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
24
33
32
−10.5
−10.0
−9.5
−9.0
−8.5
−8.0
Fitted values
lm(log(ind_P, 10) ~ Tau)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(ind_P, 10) ~ Tau)
24
33
32
−2
−1
0
1
2


Scale−Location
24
√|Standardized residuals|
0.0
0.4
0.8
1.2
−10.5
−10.0
−9.5
−9.0
−8.5
−8.0
Fitted values
lm(log(ind_P, 10) ~ Tau)

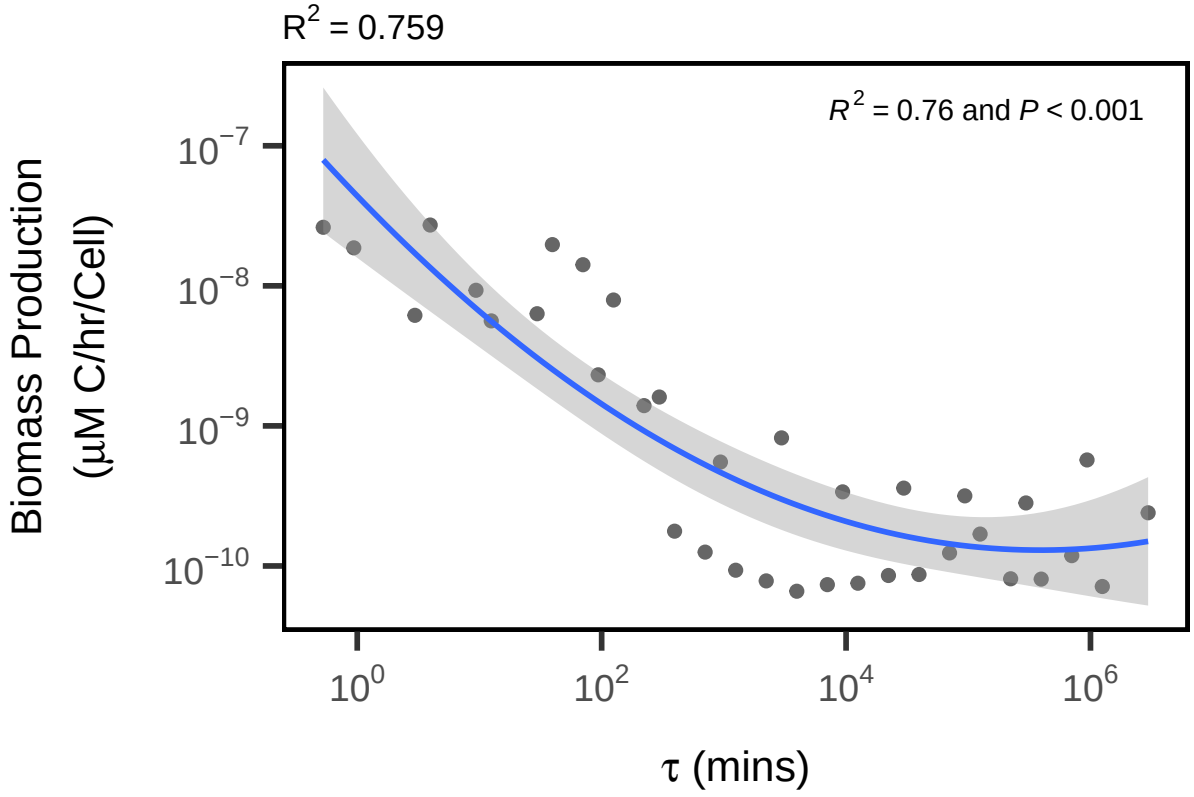



```
IP <- ggplot(Tau, aes(x = Tau, y = log(ind_P, 10)))+
  geom_point(size = 2, alpha = 0.6)+
  xlab(expression(paste(tau, " (mins)")))+
  ylab(expression(atop("Biomass Production", paste("(", mu, "M C/hr/Cell)"))))+
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE))+
  stat_poly_eq(aes(label =  paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
               label.x = "right", label.y = "top", formula = y ~ poly(x, 2, raw = TRUE), parse = TRUE, s
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(N_gam_re)$r.sq, 3))))

IP
```

```
## Warning: Removed 13 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 13 rows containing non-finite values (`stat_poly_eq()`).
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/RTLC_IP.pdf")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 13 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 13 rows containing non-finite values (`stat_poly_eq()`).
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/RTLC_IP.png")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 13 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 13 rows containing non-finite values (`stat_poly_eq()`).
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

##Biofilm Formation

```
OT_N_re <- lmer(log(OT/N) ~ Tau + (1|Set), data = Tau)
OT_N <- lm(log(OT/N) ~ Tau, data = Tau)

summary(OT_N_re)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(OT/N) ~ Tau + (1 | Set)
##    Data: Tau
##
```

```
## REML criterion at convergence: 81.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.15272 -0.68457  0.04512  0.58332  1.89456
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Set      (Intercept) 1.1474   1.0712
##  Residual             0.4135   0.6431
## Number of obs: 36, groups:  Set, 3
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -17.9853     0.6573 -27.361
## Tau          -0.3882     0.0595  -6.525
##
## Correlation of Fixed Effects:
##     (Intr)
## Tau -0.296
```

```
summary(OT_N)
```

```
##
## Call:
## lm(formula = log(OT/N) ~ Tau, data = Tau)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0661 -0.7784 -0.2485  0.8859  2.0412
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -18.16900    0.37750  -48.13  < 2e-16 ***
## Tau          -0.33356    0.09927   -3.36  0.00193 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.102 on 34 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.2493, Adjusted R-squared:  0.2272
## F-statistic: 11.29 on 1 and 34 DF,  p-value: 0.001935
```

```
AIC(OT_N, OT_N_re)
```

```
##         df      AIC
## OT_N     3 113.0918
## OT_N_re  4  89.8529
```

```
anova(OT_N_re, OT_N)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: Tau
## Models:
## OT_N: log(OT/N) ~ Tau
## OT_N_re: log(OT/N) ~ Tau + (1 | Set)
```

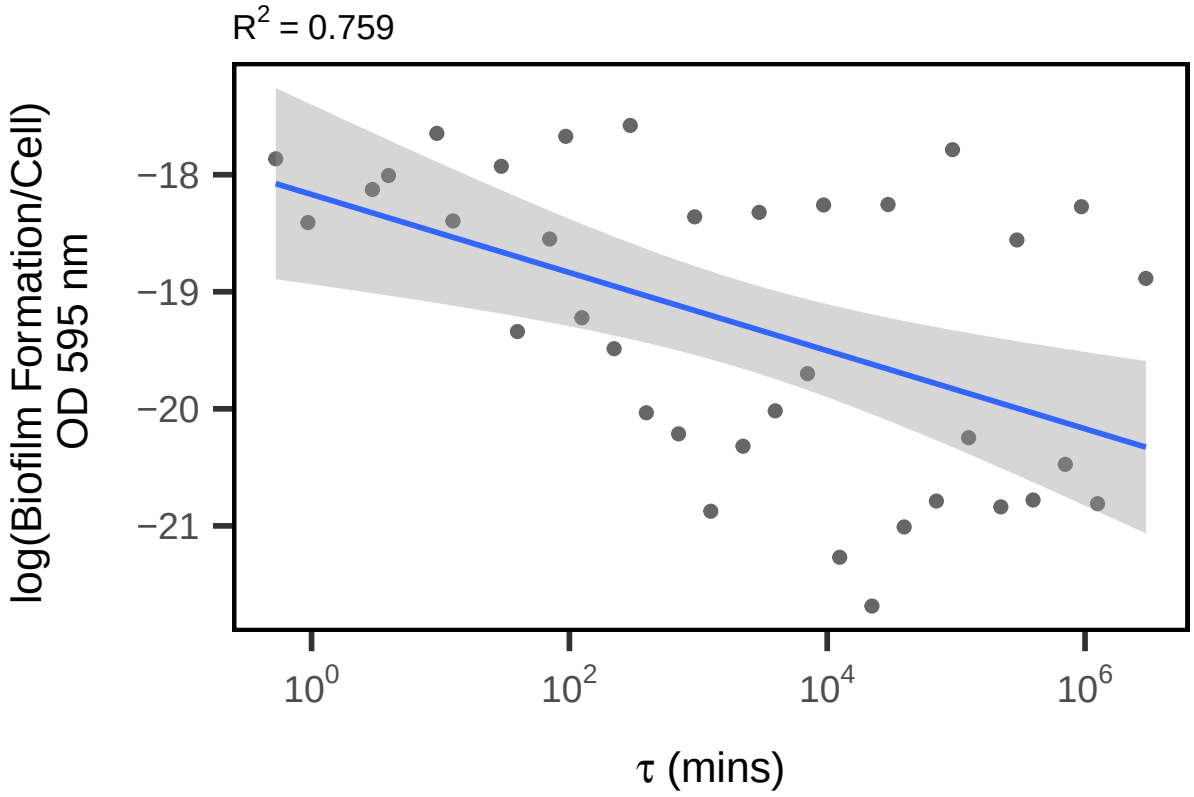
```
##         npar     AIC     BIC  logLik deviance Chisq Df Pr(>Chisq)
## OT_N       3 113.092 117.842 -53.546  107.092
## OT_N_re    4  86.721  93.055 -39.361   78.721 28.37  1  1.002e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Sig <- car::Anova(OT_N_re, test.statistic = "F")

OT_Tau <- ggplot(Tau, aes(x = Tau, y = log(OT/N)))+
  geom_point(size = 2, alpha = 0.6)+
  xlab(expression(paste(tau, " (mins)")))+
  ylab("log(Biofilm Formation/Cell) \n OD 595 nm")+
  geom_smooth(method = "lm")+
  labs(title = paste(" P =", Sig$`Pr(>F)`[1]))+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  labs(title = bquote("R"^2~ "=" ~ .(signif(summary(N_gam_re)$r.sq, 3))))

OT_Tau
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 13 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/RTLC_OT.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 13 rows containing non-finite values (`stat_smooth()`).
## Removed 13 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/RTLC_OT.png")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

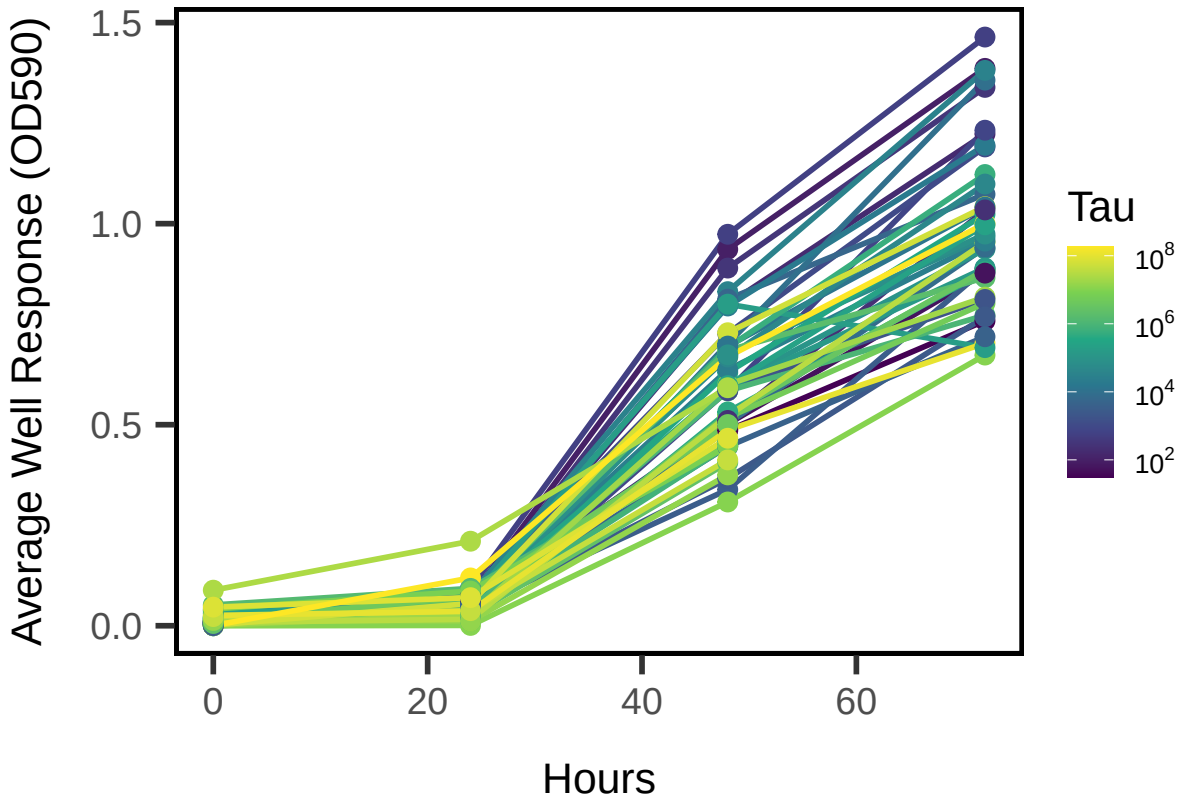
```
## Warning: Removed 13 rows containing non-finite values (`stat_smooth()`).
## Removed 13 rows containing missing values (`geom_point()`).
```

##Plot resource use vs. Time for all Tau

```
Avg_Hr <- ggplot(subset(long_EP, long_EP$C_Source =="Avg"), aes(x = Hours, y = OD, group = Tau, color =
  geom_line(size = 1)+
  geom_point(size = 3)+
  xlab("Hours")+
  ylab("Average Well Response (OD590)")+
  labs(group = "Tau")+
  scale_color_continuous(type = "viridis", labels = label_math(expr = 10^.x, format = force))+
  theme(legend.title = element_text(size = 16))

Avg_Hr
```

```
## Warning: Removed 9 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 9 rows containing missing values (`geom_point()`).
```



```
ggsave("./output/RTLC_AvgRes_Hr.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 9 rows containing missing values (`geom_line()`).
## Removed 9 rows containing missing values (`geom_point()`).
```

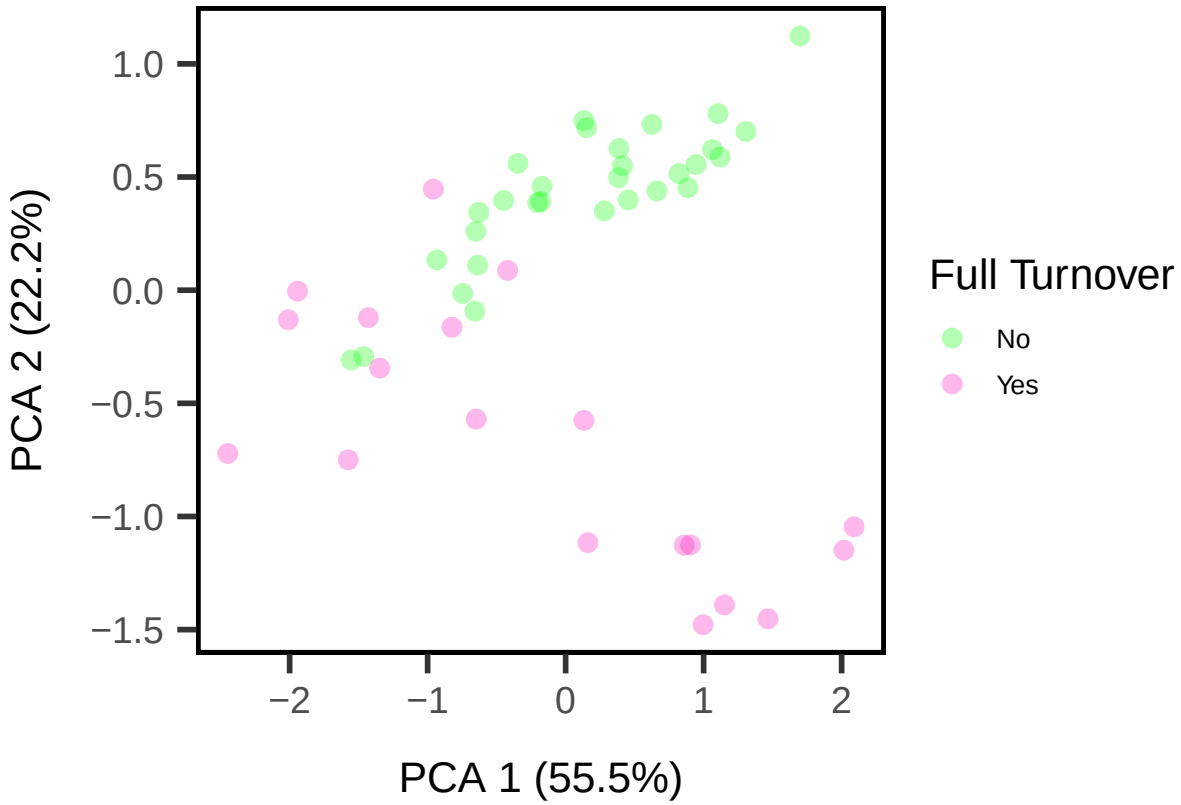
```
ggsave("./output/RTLC_AvgRes_Hr.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 9 rows containing missing values (`geom_line()`).
## Removed 9 rows containing missing values (`geom_point()`).
```

#Euclidean distance Resource use

```
EP.eu <-vegdist(EcoPlate, method = "euclidean")
EP.pca <- cmdscale(EP.eu, eig = TRUE)
EP.pca <- add.spec.scores(EP.pca, EcoPlate, method = "pcoa.scores")

explainvar1 <- round(EP.pca$eig[1] / sum(EP.pca$eig), 3) * 100
explainvar2 <- round(EP.pca$eig[2] / sum(EP.pca$eig), 3) * 100
explainvar3 <- round(EP.pca$eig[3] / sum(EP.pca$eig), 3) * 100

EP.plot <- as.data.frame(EP.pca$points)
EP.plot <- cbind(EP.plot, Tau$Tau, Tau$Turn)
colnames(EP.plot) <- c("V1", "V2", "Tau", "Turn")
EP.cproj <- as.data.frame(EP.pca$cproj)
EP.cproj <- cbind(EP.cproj, as.data.frame(row.names(EP.pca$cproj)))

EP_EU <- ggplot(EP.plot, aes(x = V1, y = V2, colour = Turn))+
  geom_point(cex = 3)+
  xlab(paste("PCA 1 (", explainvar1, "%)", sep = ""))+
  ylab(paste("PCA 2 (", explainvar2, "%)", sep = ""))+
  labs(color = "Full Turnover")+
  scale_color_manual(values = c(alpha("green",0.3), alpha("#FC0FC0", 0.3)), labels = c("No", "Yes"))+
  theme(legend.title = element_text(size = 16))


EP_EU
```




```
ggsave("./output/RTLC_Res_Eu_PCA.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/RTLC_Res_Eu_PCA.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
Res_heat <- as.data.frame(colnames(EcoPlate))
colnames(Res_heat) <- c("Res")
Res_heat$N <- colSums(EcoPlate)
rownames(EcoPlate) <- c(EcoPlate_env$Tau)


for (res in Res_heat$Res){
  x = 0
  for(site in rownames(EcoPlate)){
    x = x + (EcoPlate[site, res] * as.numeric(site))
  }
  Res_heat$Tau[Res_heat$Res == res] <- x/Res_heat$N[Res_heat$Res == res]
}

colnames(Res_heat) <- c("Res", "N", "Weight")

Res_heat <- Res_heat[order(Res_heat$Weight),]

EcoPlate <- cbind(EcoPlate_env, EcoPlate)
```

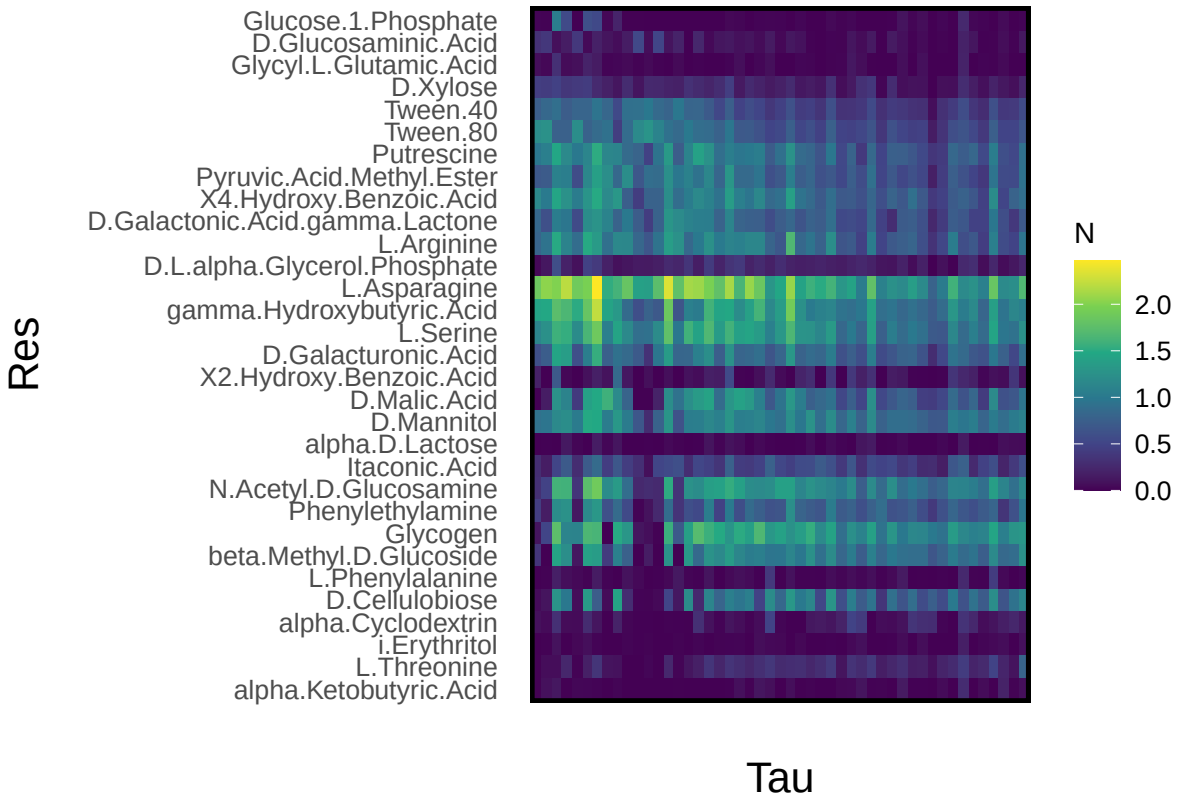
```r
colnames(EcoPlate)
```

```
##  [1] "Tau"                           "Set"
##  [3] "X2.Hydroxy.Benzoic.Acid"       "X4.Hydroxy.Benzoic.Acid"
##  [5] "alpha.Cyclodextrin"            "alpha.D.Lactose"
##  [7] "alpha.Ketobutyric.Acid"        "beta.Methyl.D.Glucoside"
##  [9] "D.Cellulobiose"                "D.Galactonic.Acid.gamma.Lactone"
## [11] "D.Galacturonic.Acid"           "D.Glucosaminic.Acid"
## [13] "D.Malic.Acid"                  "D.Mannitol"
## [15] "D.Xylose"                      "D.L.alpha.Glycerol.Phosphate"
## [17] "gamma.Hydroxybutyric.Acid"     "Glucose.1.Phosphate"
## [19] "Glycogen"                      "Glycyl.L.Glutamic.Acid"
## [21] "i.Erythritol"                  "Itaconic.Acid"
## [23] "L.Arginine"                    "L.Asparagine"
## [25] "L.Phenylalanine"               "L.Serine"
## [27] "L.Threonine"                   "N.Acetyl.D.Glucosamine"
## [29] "Phenylethylamine"              "Putrescine"
## [31] "Pyruvic.Acid.Methyl.Ester"     "Tween.40"
## [33] "Tween.80"
```

```r
#########
eco_long <- gather(EcoPlate, Res, N, X2.Hydroxy.Benzoic.Acid:Tween.80, factor_key = TRUE)
eco_long$Res <- factor(eco_long$Res, levels = rev(as.list(Res_heat$Res)))
print(levels(eco_long$Res))
```

```
##  [1] "alpha.Ketobutyric.Acid"        "L.Threonine"
##  [3] "i.Erythritol"                  "alpha.Cyclodextrin"
##  [5] "D.Cellulobiose"                "L.Phenylalanine"
##  [7] "beta.Methyl.D.Glucoside"       "Glycogen"
##  [9] "Phenylethylamine"              "N.Acetyl.D.Glucosamine"
## [11] "Itaconic.Acid"                 "alpha.D.Lactose"
## [13] "D.Mannitol"                    "D.Malic.Acid"
## [15] "X2.Hydroxy.Benzoic.Acid"       "D.Galacturonic.Acid"
## [17] "L.Serine"                      "gamma.Hydroxybutyric.Acid"
## [19] "L.Asparagine"                  "D.L.alpha.Glycerol.Phosphate"
## [21] "L.Arginine"                    "D.Galactonic.Acid.gamma.Lactone"
## [23] "X4.Hydroxy.Benzoic.Acid"       "Pyruvic.Acid.Methyl.Ester"
## [25] "Putrescine"                    "Tween.80"
## [27] "Tween.40"                      "D.Xylose"
## [29] "Glycyl.L.Glutamic.Acid"        "D.Glucosaminic.Acid"
## [31] "Glucose.1.Phosphate"
```

```r
eco_long$Tau <- as.factor(eco_long$Tau)

ecoplate_heat <- ggplot(eco_long, aes(Tau, Res, fill = N))+
  geom_tile()+
  theme(axis.ticks.x = element_blank(), axis.ticks.y = element_blank(), axis.text.x = element_blank(), a
  scale_fill_continuous(type = "viridis")

ecoplate_heat
```




```
ggsave("./output/res_heatmap.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggsave("./output/res_heatmap.png")
```

```
## Saving 6.5 x 4.5 in image
```

#Unused figures - Sequence Based

##Generate heat map of Day 20 communities with BC distance

```
OTUsr_20_or <- OTUsr_20[order(rownames(OTUsr_20)),]

tau.db <- vegdist(OTUsr_20_or, method = "bray", upper = TRUE, diag = TRUE)
order <- rev(attr(tau.db, "Labels"))

pdf(file = "./output/RTLC_BC_heat.pdf")
jpeg(file = "./output/RTLC_BC_heat.jpeg")
levelplot(as.matrix(tau.db) [, order], aspect = "iso", col.regions = inferno, xlab = "log(Tau, 10)", yla
dev.off()
```

```
## pdf
##   2
```

##Want OTUs by pH of site heat map to show horseshoe

```
# OTU_r_trim <- OTU_r[1:69,]
# OTU_heat <- as.data.frame(colnames(OTU_r_trim))
# colnames(OTU_heat) <- c("OTU")
```

```
# OTU_heat$N <- colSums(OTU_r_trim)
#
#
# for (otu in OTU_heat$OTU){
#   x = 0
#   for(site in rownames(OTU_r_trim)){
#     x = x + (OTU_r_trim[site, otu] * Tau_Seq$Tau[Tau_Seq$Seq_Sample == site])
#   }
#   OTU_heat$Tau[OTU_heat$OTU == otu] <- x/OTU_heat$N[OTU_heat$OTU == otu]
# }
#
# OTU_heat <- OTU_heat[order(OTU_heat$Tau),]
#
# OTU_heat <- OTU_heat[order(OTU_heat$Weight),]
#
# colnames(OTU_heat) <- c("OTU", "N", "Weight")
#
# OTU_heat <- cbind(OTU_heat, Tau_Seq[1:69,"Tau"])
#
# plot(x = as.numeric(OTUsr_20$Tau), y = OTUsr_20$Otu00628)
# #########
#
# colnames(OTUsr_20)[1] <- c("Tau")
# data_long <- gather(OTUsr_20, OTU, N, Otu00001:Otu33565, factor_key = TRUE)
# data_long$OTU <- factor(data_long$OTU, levels = rev(as.list(OTU_heat$OTU)))
# print(levels(data_long$OTU))
#
# map <- ggplot(data_long, aes(Tau, OTU, fill = log(N, 10)))+
#   geom_tile()+
#   theme(axis.ticks.x = element_blank(), axis.ticks.y = element_blank(), axis.text.x = element_blank()
#   scale_fill_continuous(type = "viridis", labels = label_math(expr = 10^.x, format = force))
#
# map
#
# ggsave("./output/map.pdf")
# ggsave("./output/map.png")
```

##Species Turnover (Whittaker's Turnover between two sites)

```
#Betaw = (S1 - gamma) + (S2 - gamma)/mean(S1, S2)
Betaw <- function(site1 = "", site2 = ""){
  site1 <- t(OTUs.PA[site1,])
  site2 <- t(OTUs.PA[site2,])
  site1 <- as.data.frame(subset(site1, select = site1 > 0))
  site2 <- as.data.frame(subset(site2, select = site2 > 0))
  gamma <- as.numeric(length(intersect(colnames(site1), colnames(site2))))
  bw <- ((ncol(site1) - gamma) + (ncol(site2) - gamma))/mean(ncol(site1), ncol(site2))
  return(bw)
}

Tau$Betaw <- NA
row <- 1
while(row < nrow(Tau)){
  if(!is.na(Tau[row, "Day_0_Seq"]) && !is.na(Tau[row, "Day_20_Seq"])){
```

```
    Tau[row, "Betaw"] <- Betaw(Tau[row, "Day_0_Seq"], Tau[row, "Day_20_Seq"])
  }
  row = row + 1
}

Bw_lm <- lm(Betaw ~ Tau, data = Tau)
Bw_poly <- lm(Betaw ~ poly(Tau, 2, raw = TRUE), data = Tau)
Bw_gam <- gam(Betaw ~ s(Tau), family = gaussian(link = "identity"), data = Tau, method = "REML")
Bw_gam_re <- gam(Betaw ~ s(Tau) + s(Set, bs = "re"), family = gaussian(link = "identity"), data = Tau, m

AIC(Bw_lm, Bw_poly, Bw_gam, Bw_gam_re)
```

```
##                 df       AIC
## Bw_lm     3.000000 -89.05706
## Bw_poly   4.000000 -92.55955
## Bw_gam    4.637063 -90.68502
## Bw_gam_re 6.673616 -92.38580
```

```
anova(Bw_lm, Bw_poly, Bw_gam, Bw_gam_re)
```

```
## Analysis of Variance Table
##
## Model 1: Betaw ~ Tau
## Model 2: Betaw ~ poly(Tau, 2, raw = TRUE)
## Model 3: Betaw ~ s(Tau)
## Model 4: Betaw ~ s(Tau) + s(Set, bs = "re")
##   Res.Df      RSS      Df  Sum of Sq      F  Pr(>F)
## 1 21.000 0.021595
## 2 20.000 0.017000 1.00000  0.0045948 6.2292 0.02226 *
## 3 19.888 0.017450 0.11228 -0.0004496
## 4 18.405 0.013576 1.48313  0.0038740 3.5411 0.06164 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(Bw_lm)
```

```
##
## Call:
## lm(formula = Betaw ~ Tau, data = Tau)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.043882 -0.029849  0.003575  0.019310  0.073312
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.88843    0.01233  72.052  < 2e-16 ***
## Tau          0.02025    0.00371   5.458 2.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03207 on 21 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.5865, Adjusted R-squared:  0.5668
## F-statistic: 29.79 on 1 and 21 DF,  p-value: 2.05e-05
```

```r
summary(Bw_poly)
```

```
##
## Call:
## lm(formula = Betaw ~ poly(Tau, 2, raw = TRUE), data = Tau)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05262 -0.01756 -0.00701  0.01367  0.07978
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.865849   0.014833  58.372  < 2e-16 ***
## poly(Tau, 2, raw = TRUE)1  0.045750   0.011475   3.987 0.000725 ***
## poly(Tau, 2, raw = TRUE)2 -0.004402   0.001893  -2.325 0.030711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02915 on 20 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.6745, Adjusted R-squared:  0.642
## F-statistic: 20.72 on 2 and 20 DF,  p-value: 1.335e-05
```

```r
summary(Bw_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Betaw ~ s(Tau)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.944977   0.006176     153   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F  p-value
## s(Tau) 2.112  2.637 14.48 5.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.63   Deviance explained = 66.6%
## -REML = -40.122  Scale est. = 0.00087741  n = 23
```

```r
summary(Bw_gam_re)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Betaw ~ s(Tau) + s(Set, bs = "re")
```

```
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.947210   0.009776   96.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df      F  p-value
## s(Tau) 1.714  2.125 16.619 6.99e-05 ***
## s(Set) 1.881  3.000  1.644   0.0745 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.689   Deviance explained =   74%
## -REML = -40.77  Scale est. = 0.00073763  n = 23
```

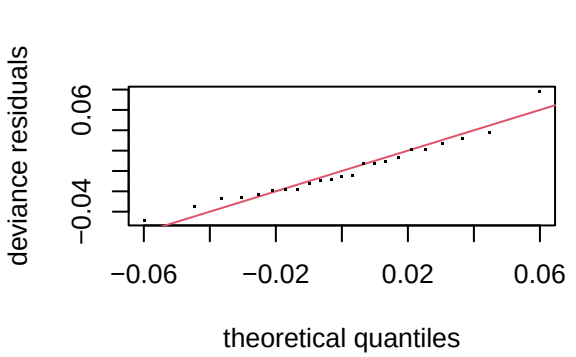
```
k.check(Bw_gam)
```

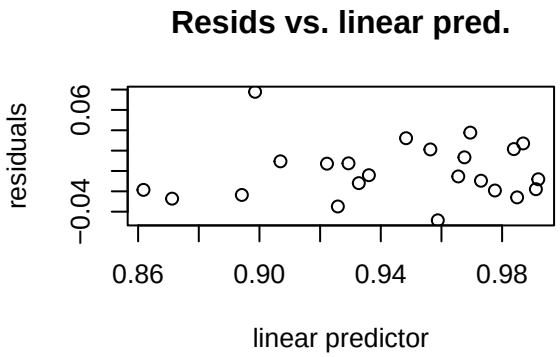
```
##        k'      edf  k-index p-value
## s(Tau)  9 2.112284 1.177311   0.725
```

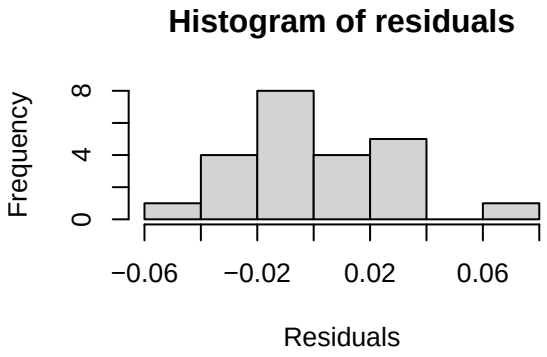
```
k.check(Bw_gam_re)
```

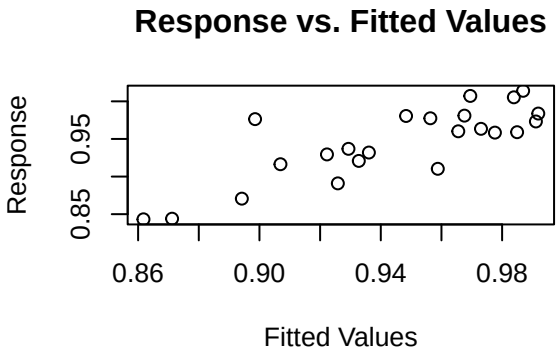
```
##        k'      edf  k-index p-value
## s(Tau)  9 1.714201 1.101156   0.645
## s(Set)  4 1.881211       NA      NA
```

```
gam.check(Bw_gam)
```



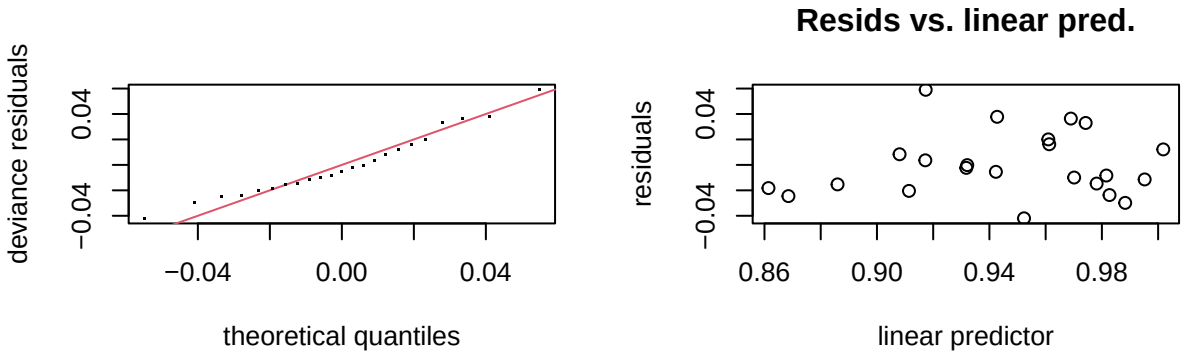


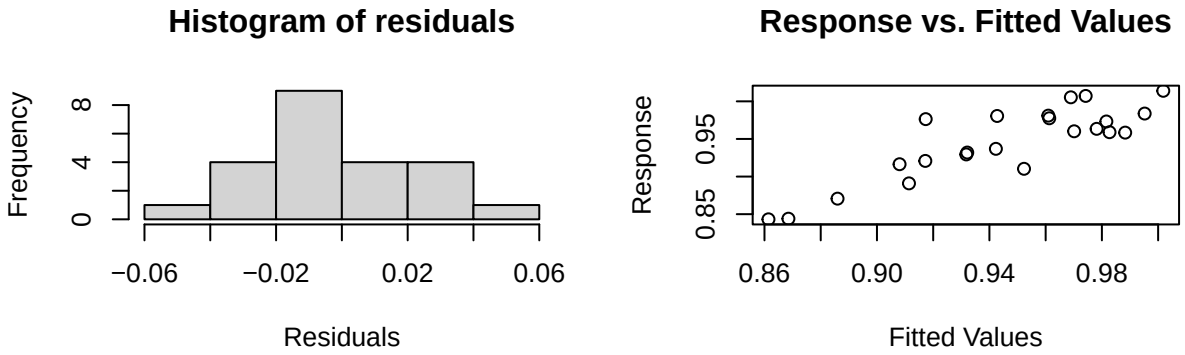





```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-7.916098e-09,3.706147e-10]
## (score -40.12205 & scale 0.000877411).
## Hessian positive definite, eigenvalue range [0.3944595,10.53061].
## Model rank =  10 / 10
##
## Basis dimension (k) checking 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(Tau) 9.00 2.11    1.18    0.74
```

```
gam.check(Bw_gam_re)
```



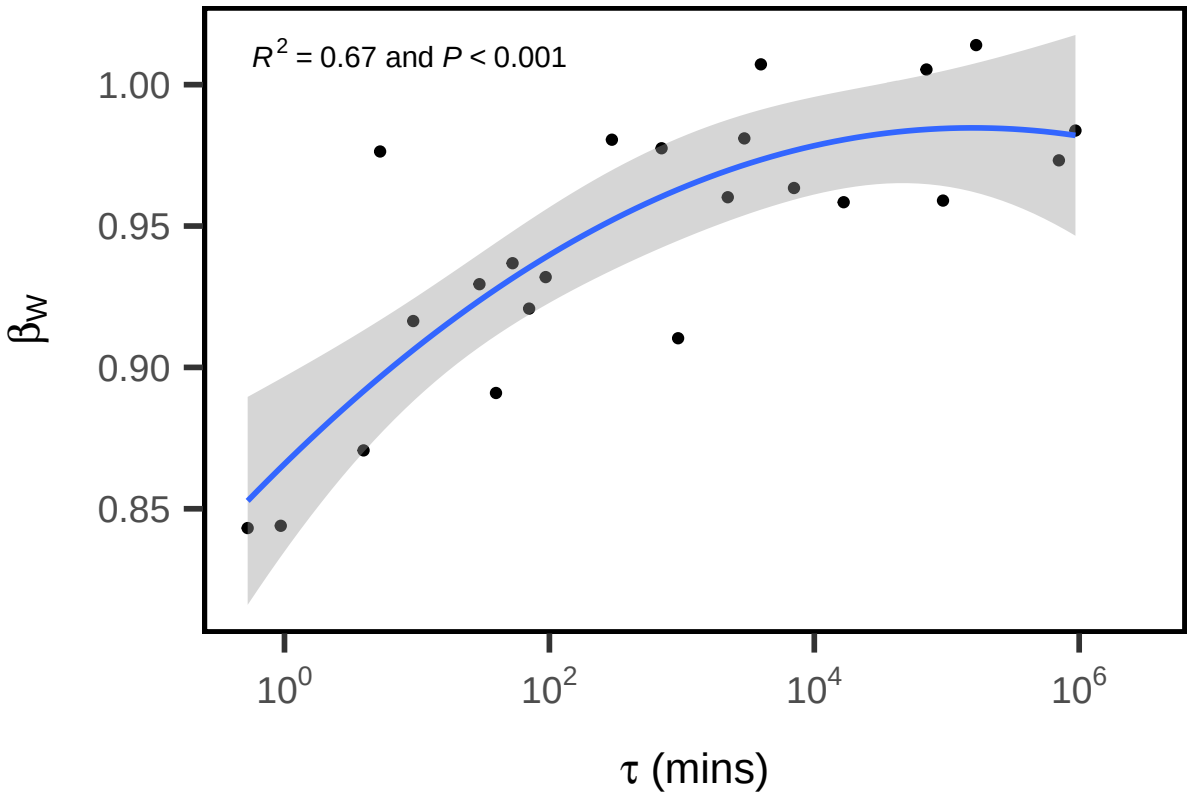



```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-1.367557e-08,3.907063e-09]
## (score -40.77026 & scale 0.0007376278).
## Hessian positive definite, eigenvalue range [0.1647294,10.60114].
## Model rank =  14 / 14
##
## Basis dimension (k) checking 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(Tau) 9.00 1.71     1.1    0.61
## s(Set) 4.00 1.88      NA      NA
```

```
betaw <- ggplot(data = Tau, aes(x = Tau, y = Betaw))+
  geom_point()+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force))+
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE))+
  xlab(expression(paste(tau, " (mins)")))+
  stat_poly_eq(aes(label = paste(stat(rr.label), "*\" and \"*", stat(p.value.label), sep = "")),
               label.x = "left", label.y = "top", formula = y~poly(x, 2, raw = TRUE), parse = TRUE, size =
  ylab(expression(beta[W]))

betaw
```

```
## Warning: Removed 26 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 26 rows containing non-finite values (`stat_poly_eq()`).
## Warning: Removed 26 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/RTLC_Bw.pdf")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 26 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 26 rows containing non-finite values (`stat_poly_eq()`).
## Warning: Removed 26 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/RTLC_Bw.png")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 26 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 26 rows containing non-finite values (`stat_poly_eq()`).
## Warning: Removed 26 rows containing missing values (`geom_point()`).
```

#Unused Plots - IBMs

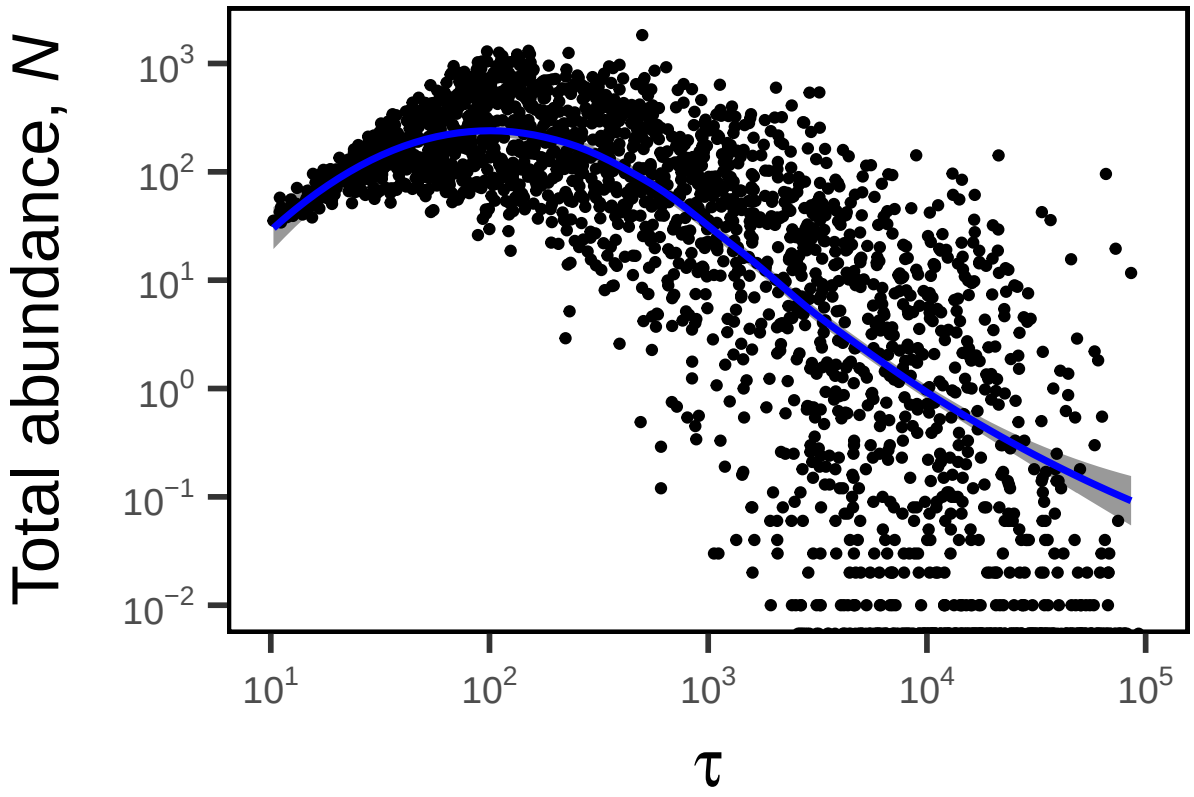
```
#IBM <- read.csv("data/IBM/SimData.csv", header = TRUE, sep = ",")

#IBM_2 <- subset(IBM, log(IBM$V, 10) <= 2 & log(IBM$V, 10) > 1)
#IBM_sum <- as.data.frame(unique(IBM_2$sim))
#colnames(IBM_sum) <- c("sim")
```

```r
#for(x in unique(IBM_sum$sim)){
#  IBM_sum[IBM_sum$sim == x, "N"] <- mean(subset(IBM_2, IBM_2$sim == x)$total.abundance)
#  IBM_sum[IBM_sum$sim == x, "V"] <- mean(subset(IBM_2, IBM_2$sim == x)$V)
#  IBM_sum[IBM_sum$sim == x, "Q"] <- mean(subset(IBM_2, IBM_2$sim == x)$Q)
#  IBM_sum[IBM_sum$sim == x, "S"] <- mean(subset(IBM_2, IBM_2$sim == x)$species.richness)
#  IBM_sum[IBM_sum$sim == x, "E"] <- mean(subset(IBM_2, IBM_2$sim == x)$simpson.e, na.rm = TRUE)
#  IBM_sum[IBM_sum$sim == x, "P"] <- mean(subset(IBM_2, IBM_2$sim == x)$ind.production, na.rm = TRUE)
#  IBM_sum[IBM_sum$sim == x, "B"] <- mean(subset(IBM_2, IBM_2$sim == x)$whittakers.turnover, na.rm = TR
#  IBM_sum[IBM_sum$sim == x, "Res"] <- mean(subset(IBM_2, IBM_2$sim == x)$avg.per.capita.efficiency1e, 
#}

#write.csv(IBM_sum, "data/IBM/IBM_sum.csv", row.names = FALSE)

IBM_sum <- read.csv("data/IBM/IBM_sum.csv", header = TRUE, sep = ",")
```

```r
N_IBM <- ggplot(data = subset(IBM_sum, log(IBM_sum$V, 10) <= 2 & log(IBM_sum$V, 10) > 1), aes(y = log(N
  geom_point()+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(1, 5))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force))+
  xlab(expression(paste(tau)))+
  ylab(expression(paste("Total abundance, ", italic("N"))))+
  stat_smooth(method = "loess", color = "blue", cex = 1.25, level = 0.95, fill = alpha("black", 0.4))+
  theme(axis.title = element_text(size = 25))

N_IBM
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 295 rows containing non-finite values (`stat_smooth()`).
```




```
ggsave("./output/IBM_N.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 295 rows containing non-finite values (`stat_smooth()`).
```

```
ggsave("./output/IBM_N.png", width = 6.5, height = 5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 295 rows containing non-finite values (`stat_smooth()`).
```

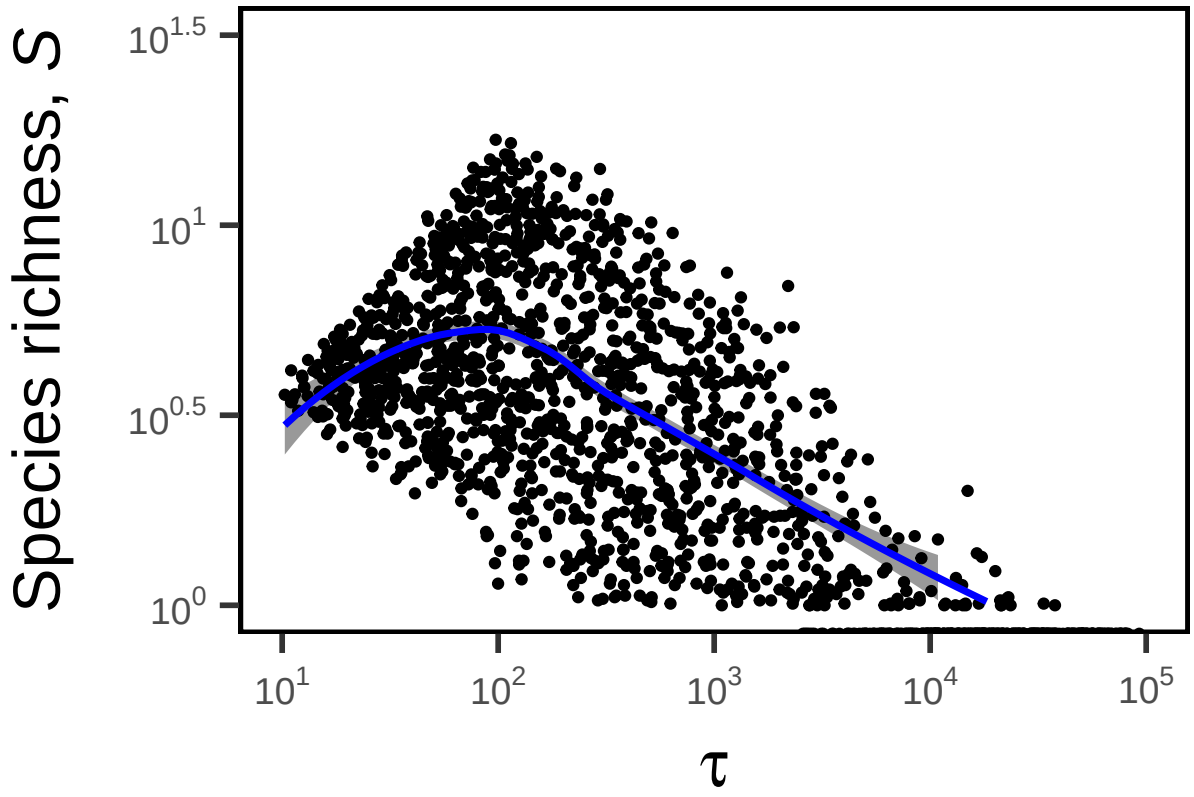
```
S_IBM <- ggplot(data = subset(IBM_sum, log(IBM_sum$V, 10) <= 2 & log(IBM_sum$V, 10) > 1), aes(y = log(S
  geom_point()+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(1, 5))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(0, 1.5))+
  xlab(expression(paste(tau)))+
  ylab(expression(paste("Species richness, ", italic("S"))))+
  stat_smooth(method = "loess", color = "blue", cex = 1.25, level = 0.95, fill = alpha("black", 0.4))+
  theme(axis.title = element_text(size = 25))

S_IBM
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 971 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 676 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 7 rows containing missing values (`geom_smooth()`).
```




```
ggsave("./output/IBM_S.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 971 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 676 rows containing missing values (`geom_point()`).
## Warning: Removed 7 rows containing missing values (`geom_smooth()`).
```

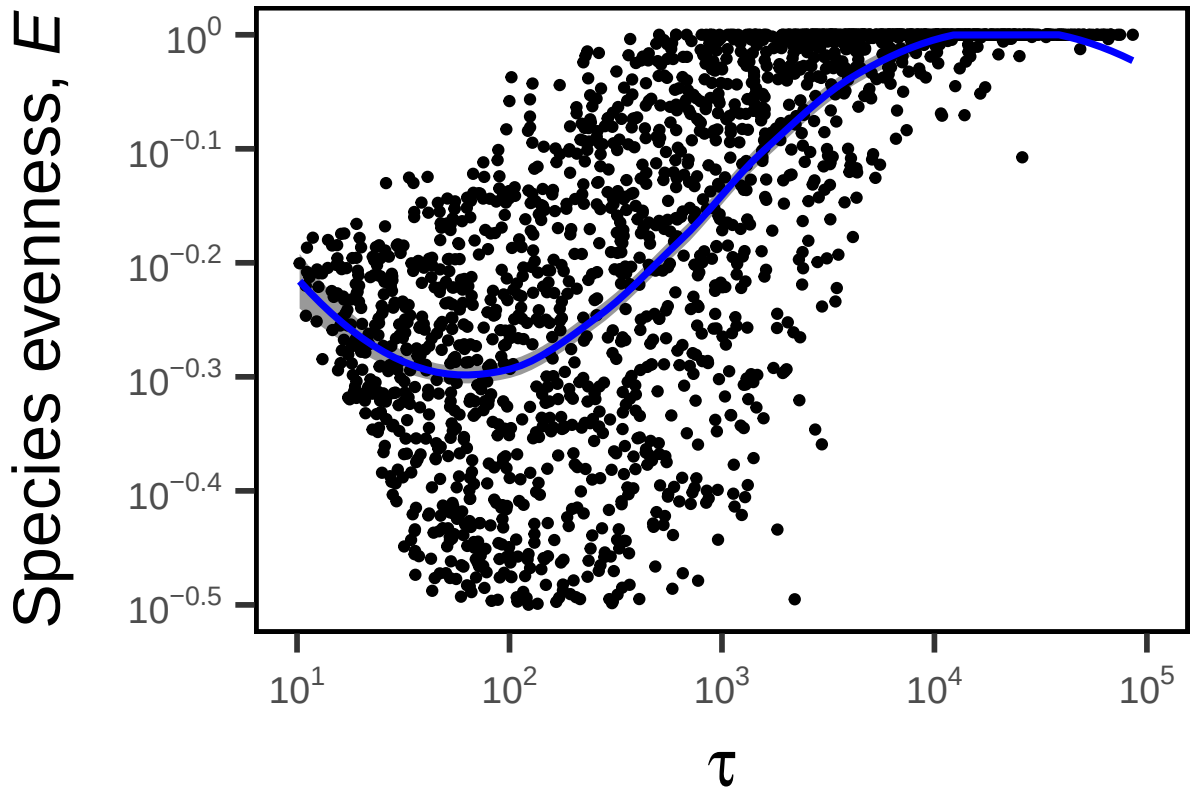
```
ggsave("./output/IBM_S.png", width = 6.5, height = 5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 971 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 676 rows containing missing values (`geom_point()`).
## Warning: Removed 7 rows containing missing values (`geom_smooth()`).
```

```
E_IBM <- ggplot(data = subset(IBM_sum, log(IBM_sum$V, 10) <= 2 & log(IBM_sum$V, 10) > 1), aes(y = log(E
  geom_point()+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(1, 5))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(-0.5, 0))+
  xlab(expression(paste(tau)))+
  ylab(expression(paste("Species evenness, ", italic("E"))))+
  stat_smooth(method = "loess", color = "blue", cex = 1.25, level = 0.95, fill = alpha("black", 0.4))+
  theme(axis.title = element_text(size = 25))
```

```
E_IBM
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 410 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 410 rows containing missing values (`geom_point()`).
## Warning: Removed 9 rows containing missing values (`geom_smooth()`).
```




```
ggsave("./output/IBM_E.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 410 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 410 rows containing missing values (`geom_point()`).
## Warning: Removed 9 rows containing missing values (`geom_smooth()`).
```

```
ggsave("./output/IBM_E.png", width = 6.5, height = 5)
```

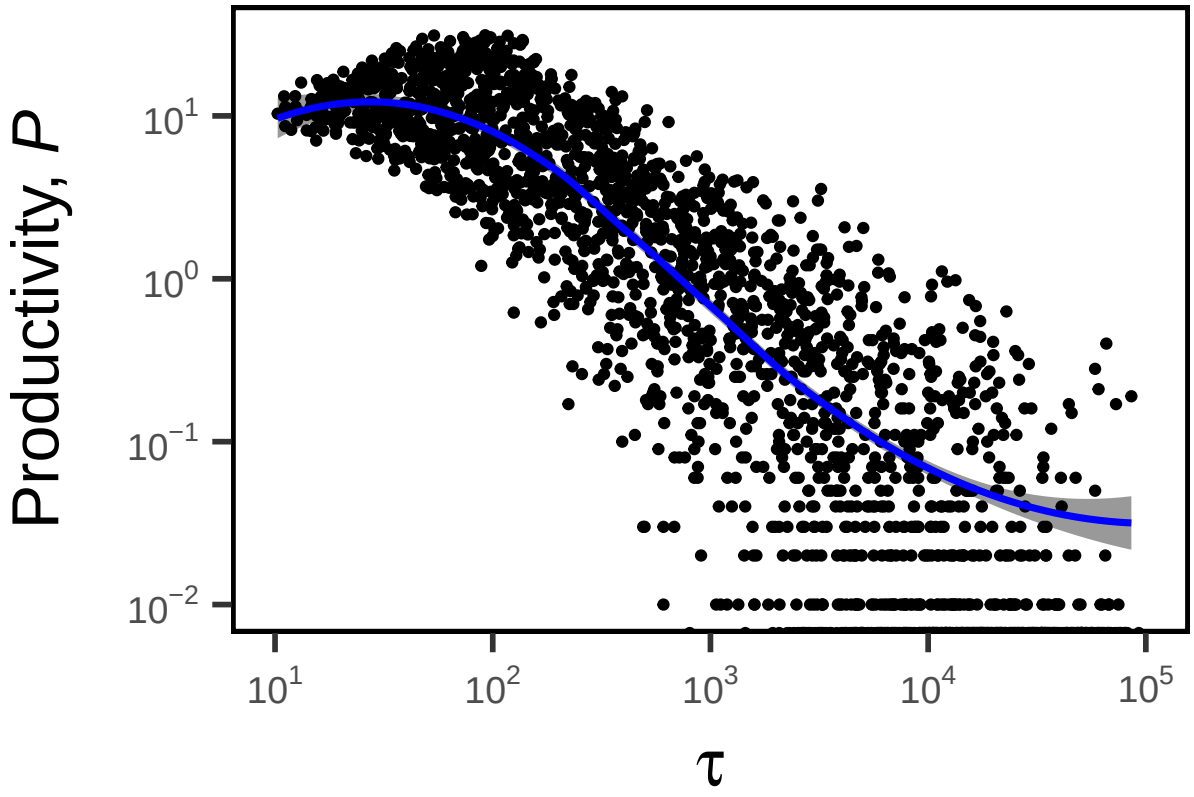
```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 410 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 410 rows containing missing values (`geom_point()`).
## Warning: Removed 9 rows containing missing values (`geom_smooth()`).
```

```
P_IBM <- ggplot(data = subset(IBM_sum, log(IBM_sum$V, 10) <= 2 & log(IBM_sum$V, 10) > 1), aes(y = log(P
  geom_point()+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(1, 5))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(-2, 1.5))+
  xlab(expression(paste(tau)))+
  ylab(expression(paste("Productivity, ", italic("P"))))+
  stat_smooth(method = "loess", color = "blue", cex = 1.25, level = 0.95, fill = alpha("black", 0.4))+
  theme(axis.title = element_text(size = 25))

P_IBM
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 436 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/IBM_P.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 436 rows containing non-finite values (`stat_smooth()`).
## Removed 11 rows containing missing values (`geom_point()`).
```

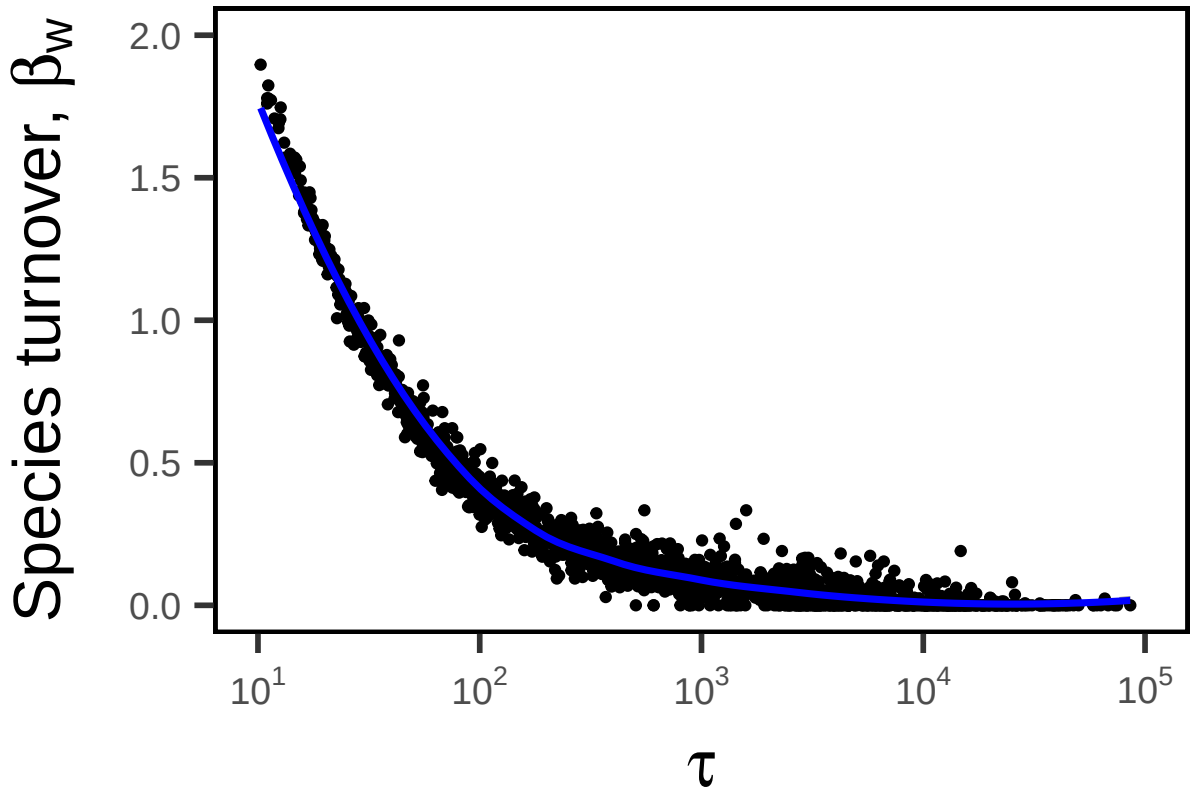
```
ggsave("./output/IBM_P.png", width = 6.5, height = 5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 436 rows containing non-finite values (`stat_smooth()`).
## Removed 11 rows containing missing values (`geom_point()`).
```

```
B_IBM <- ggplot(data = subset(IBM_sum, log(IBM_sum$V, 10) <= 2 & log(IBM_sum$V, 10) > 1), aes(y = B, x =
  geom_point()+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(1, 5))+
  ylim(c(0,2))+
  xlab(expression(paste(tau)))+
  ylab(expression(paste("Species turnover, ", beta[w])))+
  stat_smooth(method = "loess", color = "blue", cex = 1.25, level = 0.95, fill = alpha("black", 0.4))+
  theme(axis.title = element_text(size = 25))

B_IBM
```

```
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 430 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 430 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/IBM_B.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
## Warning: Removed 430 rows containing non-finite values (`stat_smooth()`).
## Removed 430 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/IBM_B.png", width = 6.5, height = 5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

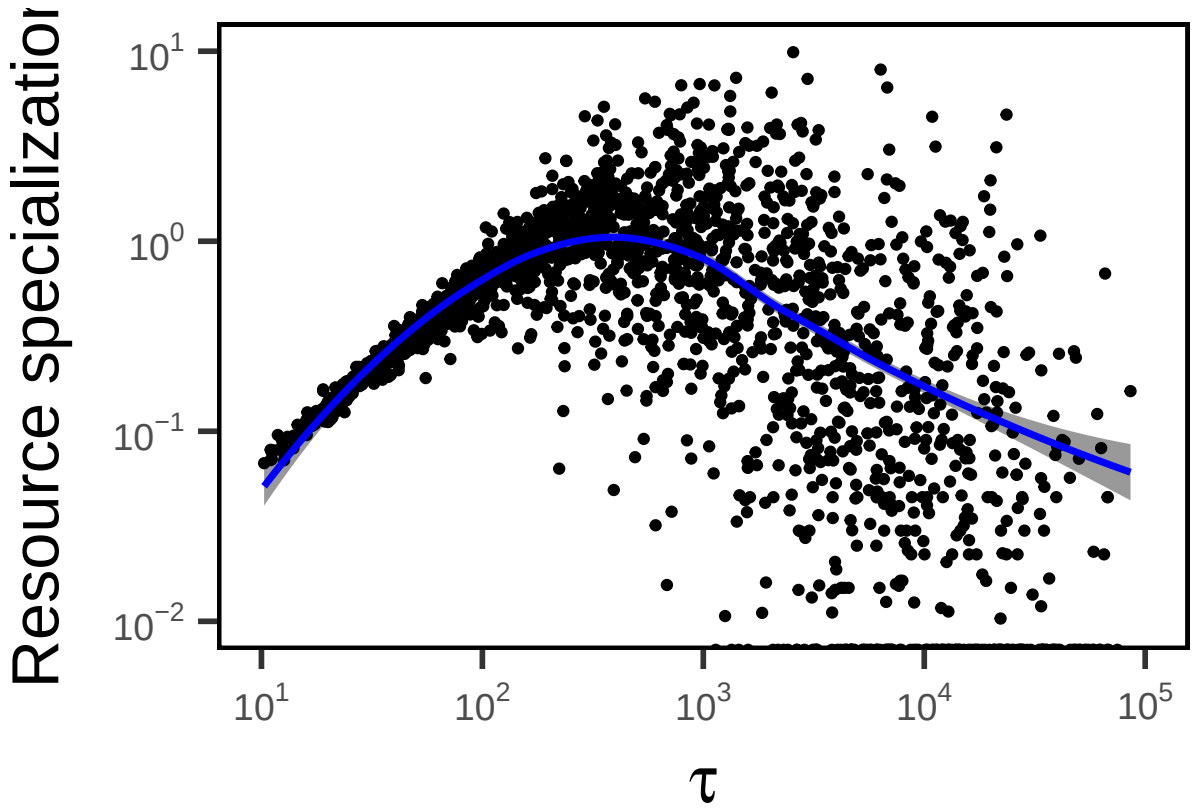
```
## Warning: Removed 430 rows containing non-finite values (`stat_smooth()`).
## Removed 430 rows containing missing values (`geom_point()`).
```

```
Res_IBM <- ggplot(data = subset(IBM_sum, log(IBM_sum$V, 10) <= 2 & log(IBM_sum$V, 10) > 1), aes(y = log
  geom_point()+
  scale_x_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(1, 5))+
  scale_y_continuous(labels = label_math(expr = 10^.x, format = force), limits = c(-2, 1))+
  xlab(expression(paste(tau)))+
  ylab("Resource specialization")+
  stat_smooth(method = "loess", color = "blue", cex = 1.25, level = 0.95, fill = alpha("black", 0.4))+
  theme(axis.title = element_text(size = 25))

Res_IBM
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 471 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 335 rows containing missing values (`geom_point()`).
```




```
ggsave("./output/IBM_Res.pdf")
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 471 rows containing non-finite values (`stat_smooth()`).
## Removed 335 rows containing missing values (`geom_point()`).
```

```
ggsave("./output/IBM_Res.png", width = 6.5, height = 5)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 471 rows containing non-finite values (`stat_smooth()`).
## Removed 335 rows containing missing values (`geom_point()`).
```