Project.Final

Leslie Speight Youtsey and Liz Weatherup

11/13/2020

Title : Fluxes in the ocean

Abstract:

Introduction:

Toolbox discussion:

Description:

Implementation:

Reading csv data in

```
rm(list=ls(all=TRUE))  #Housekeeping: clear out old files

options(warn = -1)
knitr::opts_chunk$set(echo = T, fig.height=6, fig.width=8, warning = F, message = F)
```

```
bats_flux <- read.csv("bats_flux.csv")

delete.na <- function(DF, n=0) {
  DF[rowSums(is.na(DF)) <= n,]
} #Function that takes rows out that contains NAs
bats_flux.noNA <- delete.na(bats_flux) # Take out rows containing NAs

# Too many observations taken out, lets subset a new data frame

Data <- subset(bats_flux,select = c("cr","dep","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2","M_avg","C_avg","N_avg","P_avg"))
Data.noNA <- delete.na(Data)

# now we can use Data.noNA to compare each flux to one another, since no flux contains NAs
```

Here is what our data looks like

```
head(Data)
```

	cr	dep	yymmdd1	yymmdd2	Lat2	Lat2.1	Long1	Long2	M_avg
	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	10003	150	19881216	19881219	31.737	31.708	64.242	64.000	59.00
2	10004	150	19890125	19890128	31.677	31.635	64.202	64.270	67.90
3	10004	200	19890125	19890128	31.677	31.635	64.202	64.270	63.70
4	10004	300	19890125	19890128	31.677	31.635	64.202	64.270	55.70
5	10004	400	19890125	19890128	31.677	31.635	64.202	64.270	27.45
6	10005	150	19890217	19890220	31.679	31.693	64.202	64.883	274.80

6 rows | 1-10 of 13 columns

Subsetting each flux out seperatly

```
# separate each flux into its own dataframe
# delete rows that contain NAs
C_flux.data <- delete.na(subset(Data,select = c("cr","dep","yymmdd1","yymmdd2","Lat2"
,"Lat2.1","Long1","Long2","C_avg")))
N_flux.data <- delete.na(subset(Data,select = c("cr","dep","yymmdd1","yymmdd2","Lat2"
,"Lat2.1","Long1","Long2","N_avg")))
P_flux.data <- delete.na(subset(Data,select = c("cr","dep","yymmdd1","yymmdd2","Lat2"
,"Lat2.1","Long1","Long2","P_avg")))
M_flux.data <- delete.na(subset(Data,select = c("cr","dep","yymmdd1","yymmdd2","Lat2"
,"Lat2.1","Long1","Long2","M_avg")))
```

Subseting each depth and Renaming Columns

```
# Subset each depth and rename columns
library(tidyverse)
library(dplyr)
# use tideyverse instead
C_flux.depth.150 <- subset(C_flux.data, dep == 150)
C_flux.depth.150 <- rename(C_flux.depth.150,c("dep.150" = "dep"))
C_flux.depth.150 <- rename(C_flux.depth.150, c("C_avg.150" = "C_avg" ))
C_flux.depth.200 <- subset(C_flux.data, dep == 200)
C_flux.depth.200 <- rename(C_flux.depth.200, c("dep.200"="dep"))
C_flux.depth.200 <- rename(C_flux.depth.200, c("C_avg.200" ="C_avg"))
C_flux.depth.300 <- subset(C_flux.data, dep == 300)
C_flux.depth.300 <- rename(C_flux.depth.300, c("dep.300"="dep"))
C_flux.depth.300 <- rename(C_flux.depth.300, c("C_avg.300" ="C_avg"))
C_flux.depth.400 <- subset(C_flux.data, dep == 400)
C_flux.depth.400 <- rename(C_flux.depth.400, c("dep.400"="dep"))
C_flux.depth.400 <- rename(C_flux.depth.400, c("C_avg.400" ="C_avg"))
```

```
N_flux.depth.150 <- subset(N_flux.data, dep == 150)
N_flux.depth.150 <- rename(N_flux.depth.150, c("dep.150"="dep"))
N_flux.depth.150 <- rename(N_flux.depth.150, c("N_avg.150" ="N_avg"))
N_flux.depth.200 <- subset(N_flux.data, dep == 200)
N_flux.depth.200 <- rename(N_flux.depth.200, c("dep.200"="dep"))
N_flux.depth.200 <- rename(N_flux.depth.200, c("N_avg.200" ="N_avg"))
N_flux.depth.300 <- subset(N_flux.data, dep == 300)
N_flux.depth.300 <- rename(N_flux.depth.300, c("dep.300"="dep"))
N_flux.depth.300 <- rename(N_flux.depth.300, c("N_avg.300" ="N_avg"))
N_flux.depth.400 <- subset(N_flux.data, dep == 400)
N_flux.depth.400 <- rename(N_flux.depth.400, c("dep.400"="dep"))
N_flux.depth.400 <- rename(N_flux.depth.400, c("N_avg.400" ="N_avg"))
```

```
P_flux.depth.150 <- subset(P_flux.data, dep == 150)
P_flux.depth.150 <- rename(P_flux.depth.150, c("dep.150"="dep"))
P_flux.depth.150 <- rename(P_flux.depth.150, c("P_avg.150" ="P_avg"))
P_flux.depth.200 <- subset(P_flux.data, dep == 200)
P_flux.depth.200 <- rename(P_flux.depth.200, c("dep.200"="dep"))
P_flux.depth.200 <- rename(P_flux.depth.200, c("P_avg.200" ="P_avg"))
P_flux.depth.300 <- subset(P_flux.data, dep == 300)
P_flux.depth.300 <- rename(P_flux.depth.300, c("dep.300"="dep"))
P_flux.depth.300 <- rename(P_flux.depth.300, c("P_avg.300" ="P_avg"))
P_flux.depth.400 <- subset(P_flux.data, dep == 400)
P_flux.depth.400 <- rename(P_flux.depth.400, c("dep.400"="dep"))
P_flux.depth.400 <- rename(P_flux.depth.400, c("P_avg.400" ="P_avg"))
```

```
M_flux.depth.150 <- subset(M_flux.data, dep == 150)
M_flux.depth.150 <- rename(M_flux.depth.150, c("dep.150"="dep"))
M_flux.depth.150 <- rename(M_flux.depth.150, c("M_avg.150" ="M_avg"))
M_flux.depth.200 <- subset(M_flux.data, dep == 200)
M_flux.depth.200 <- rename(M_flux.depth.200, c("dep.200"="dep"))
M_flux.depth.200 <- rename(M_flux.depth.200, c("M_avg.200" ="M_avg"))
M_flux.depth.300 <- subset(M_flux.data, dep == 300)
M_flux.depth.300 <- rename(M_flux.depth.300, c("dep.300"="dep"))
M_flux.depth.300 <- rename(M_flux.depth.300, c("M_avg.300" ="M_avg"))
M_flux.depth.400 <- subset(M_flux.data, dep == 400)
M_flux.depth.400 <- rename(M_flux.depth.400, c("dep.400"="dep"))
M_flux.depth.400 <- rename(M_flux.depth.400, c("M_avg.400" ="M_avg"))
```

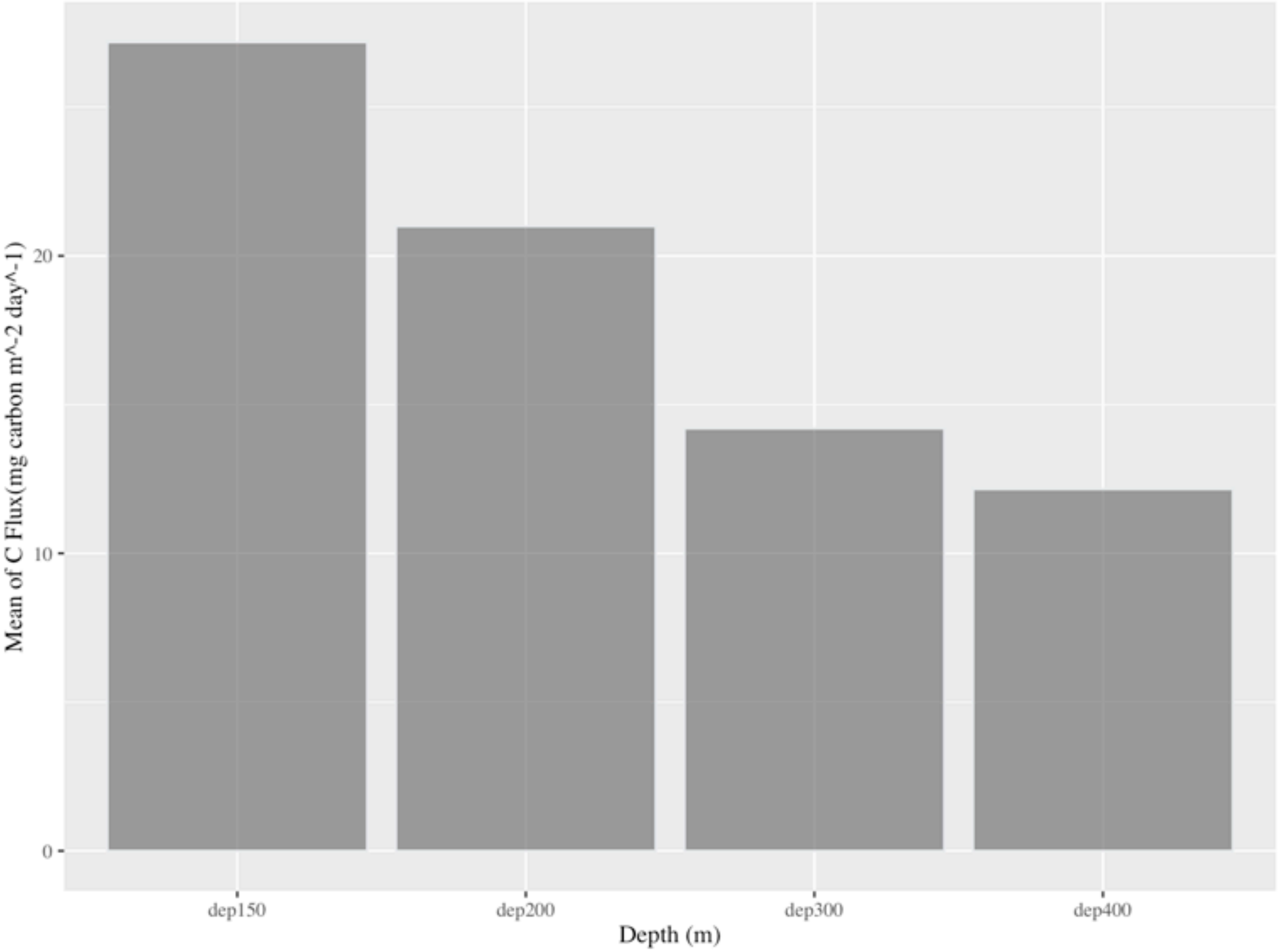
Bar plots to access relationship between fluxes and depths

Mean and SE of each depth of C-flux

```
dep150 <- subset(C_flux.data, dep=="150") #focusing on mean of C-avg at depth 150

muCflux150 <- mean(dep150$C_avg, data=dep150)

s.e150c <- muCflux150 * sqrt(10)  #need to find s.e

dep200 <- subset(C_flux.data, dep=="200") #focusing on mean of C-avg at depth 200

muCflux200 <- mean(dep200$C_avg, data=dep200)

s.e200c <- muCflux200 * sqrt(10)

dep300 <- subset(C_flux.data, dep=="300") #focusing on mean of C-avg at depth 300

muCflux300 <- mean(dep300$C_avg, data=dep300)

s.e300c <- muCflux300 * sqrt(10)

dep400 <- subset(C_flux.data, dep=="400") #focusing on mean of C-avg at depth 400

muCflux400 <- mean(dep400$C_avg, data=dep400)

s.e400c <- muCflux400 * sqrt(10)

bar.Cflux <- data.frame(c("dep150","dep200","dep300","dep400"), c(muCflux150,muCflux2
00,muCflux300,muCflux400))
colnames(bar.Cflux)[1] <- "Depth"
colnames(bar.Cflux)[2] <- "C_fluxMean"
```

Graph of C-flux

```
library(ggplot2)
library(dplyr)
library(viridisLite)

Viridis_palette <- viridis(40)[seq(1,40, 2)]
xval <- bar.Cflux$Depth #character objects need quotes
yval <- bar.Cflux$`C_flux Mean`

ggplot()+geom_col(bar.Cflux, mapping=aes(x= Depth, y = C_fluxMean), color="#e9ecef",
alpha=0.6,) +
xlab("Depth (m)") + ylab("Mean of C Flux(mg carbon m^-2 day^-1)") +

 theme(text = element_text(family = 'serif')) +
scale_fill_manual(values = Viridis_palette)
```

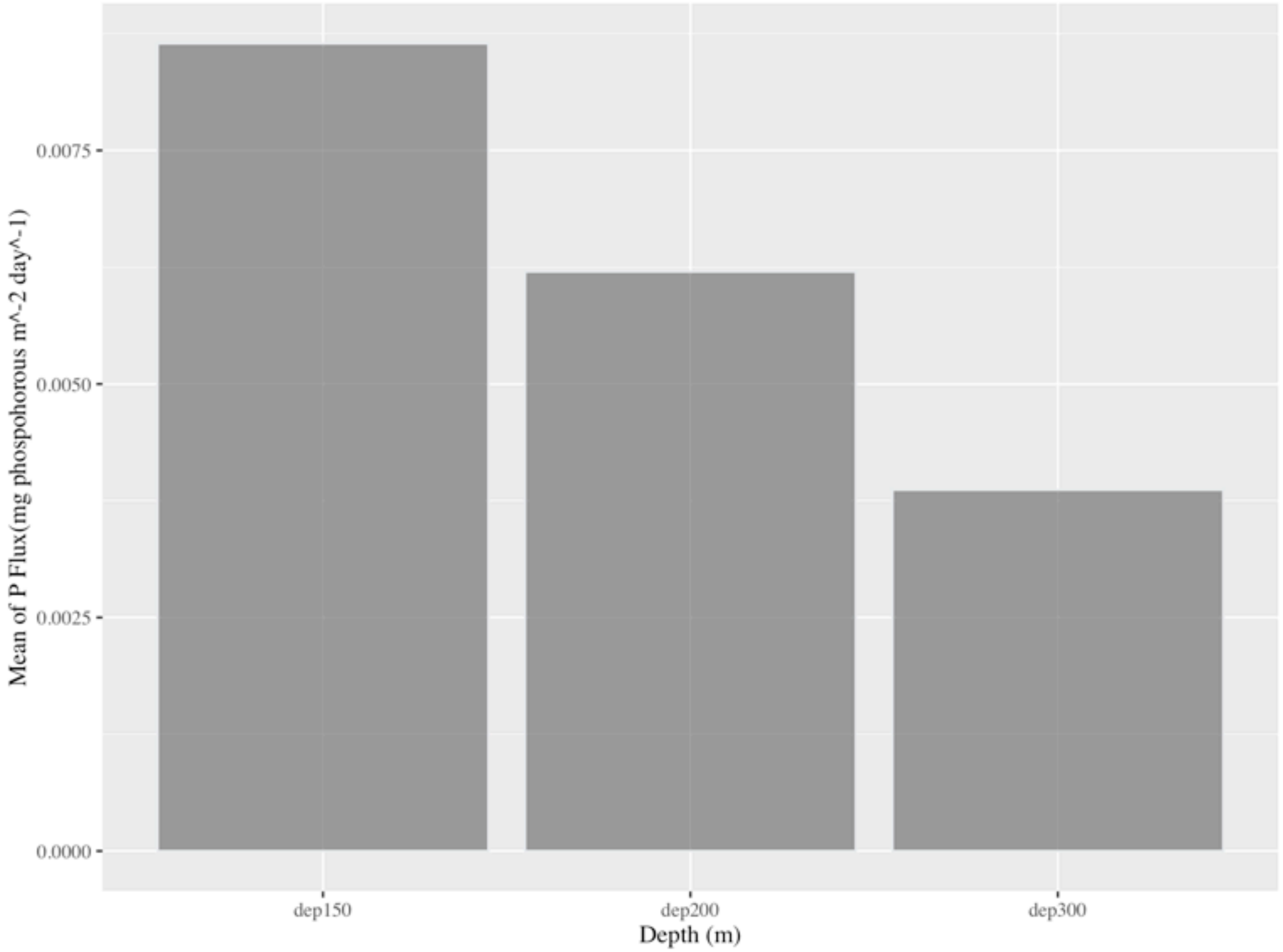


Mean and SE of each depth of P-flux

```
dep150p <- subset(P_flux.data, dep=="150") #focusing on mean of P-avg at depth 150

muPflux150 <- mean(dep150p$P_avg, data=dep150p)

s.e150p <- muPflux150 * sqrt(10)  #need to find s.e

dep200p <- subset(P_flux.data, dep=="200") #focusing on mean of P-avg at depth 200

muPflux200 <- mean(dep200p$P_avg, data=dep200p)

s.e200p <- muPflux200 * sqrt(10)

dep300p <- subset(P_flux.data, dep=="300") #focusing on mean of P-avg at depth 300

muPflux300 <- mean(dep300p$P_avg, data=dep300p)

s.e300p <- muPflux300 * sqrt(10)

bar.Pflux <- data.frame(c("dep150","dep200","dep300"), c(muPflux150,muPflux200,muPflu
x300))
colnames(bar.Pflux)[1] <- "Depth"
colnames(bar.Pflux)[2] <- "P_fluxMean"
```

Graph of P-flux

```
library(ggplot2)
library(dplyr)
library(viridisLite)

Viridis_palette <- viridis(40)[seq(1,40, 2)]
xval <- bar.Pflux$Depth #character objects need quotes
yval <- bar.Pflux$`P_flux Mean`

ggplot()+geom_col(bar.Pflux, mapping=aes(x= Depth, y = P_fluxMean), color="#e9ecef",
alpha=0.6,) +
xlab("Depth (m)") + ylab("Mean of P Flux(mg phospohorous m^-2 day^-1)") +

 theme(text = element_text(family = 'serif')) +
scale_fill_manual(values = Viridis_palette)
```

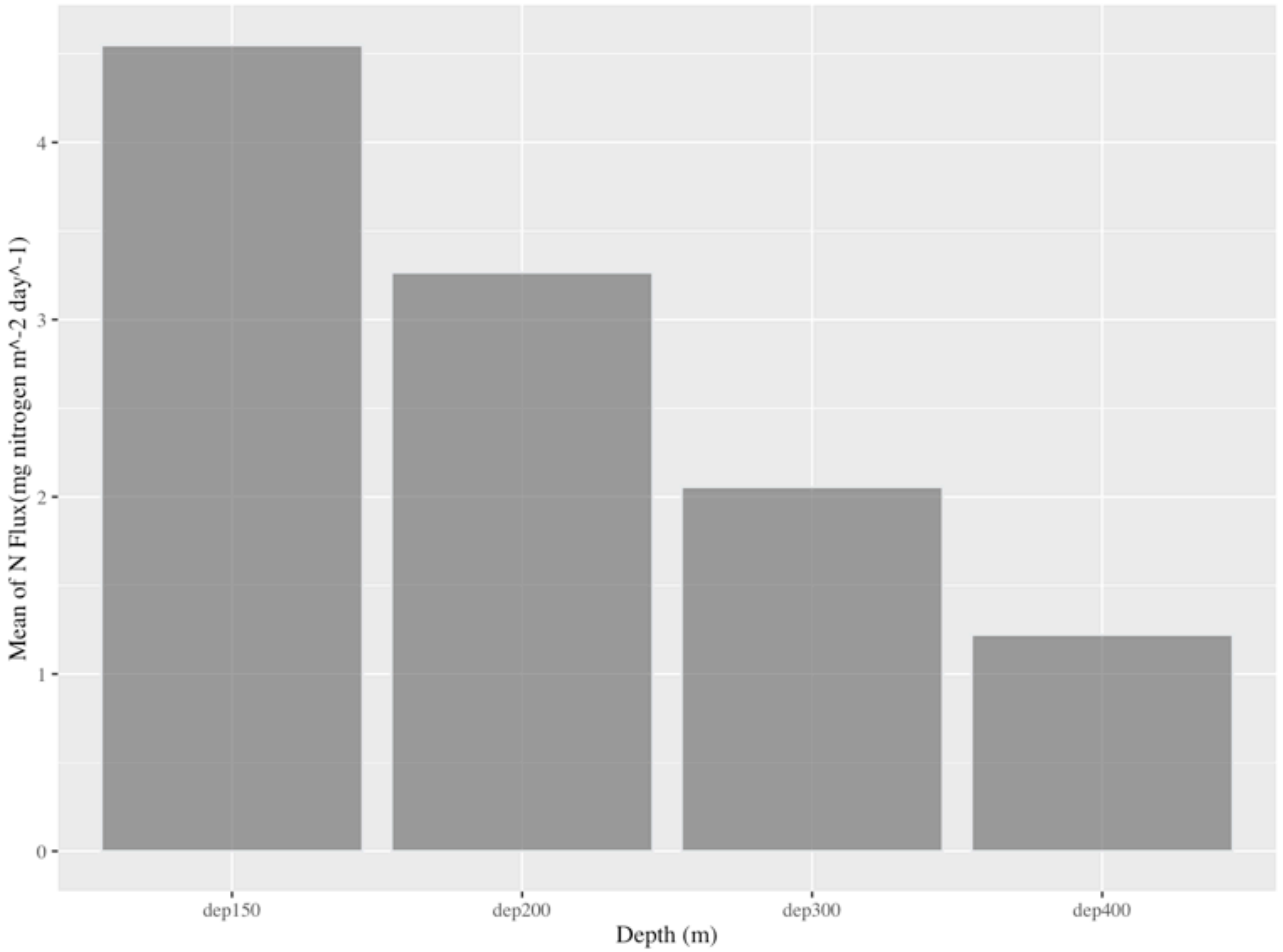


Mean and SE of each depth of N-flux

```
dep150n <- subset(N_flux.data, dep=="150") #focusing on mean of N-avg at depth 150

muNflux150 <- mean(dep150n$N_avg, data=dep150n)

s.e150n <- muNflux150 * sqrt(10)  #need to find s.e

dep200n <- subset(N_flux.data, dep=="200") #focusing on mean of N-avg at depth 200

muNflux200 <- mean(dep200n$N_avg, data=dep200n)

s.e200n <- muNflux200 * sqrt(10)

dep300n <- subset(N_flux.data, dep=="300") #focusing on mean of N-avg at depth 300

muNflux300 <- mean(dep300n$N_avg, data=dep300n)

s.e300n <- muNflux300 * sqrt(10)

dep400n <- subset(N_flux.data, dep=="400") #focusing on mean of N-avg at depth 400

muNflux400 <- mean(dep400n$N_avg, data=dep400n)

s.e400n <- muNflux400 * sqrt(10)

bar.Nflux <- data.frame(c("dep150","dep200","dep300","dep400"), c(muNflux150,muNflux2
00,muNflux300,muNflux400))
colnames(bar.Nflux)[1] <- "Depth"
colnames(bar.Nflux)[2] <- "N_fluxMean"
```

Graph of N-flux

```
library(ggplot2)
library(dplyr)
library(viridisLite)

Viridis_palette <- viridis(40)[seq(1,40, 2)]
xval <- bar.Nflux$Depth #character objects need quotes
yval <- bar.Nflux$`N_flux Mean`

ggplot()+geom_col(bar.Nflux, mapping=aes(x= Depth, y = N_fluxMean), color="#e9ecef",
alpha=0.6,) +
xlab("Depth (m)") + ylab("Mean of N Flux(mg nitrogen m^-2 day^-1)") +

 theme(text = element_text(family = 'serif')) +
scale_fill_manual(values = Viridis_palette)
```

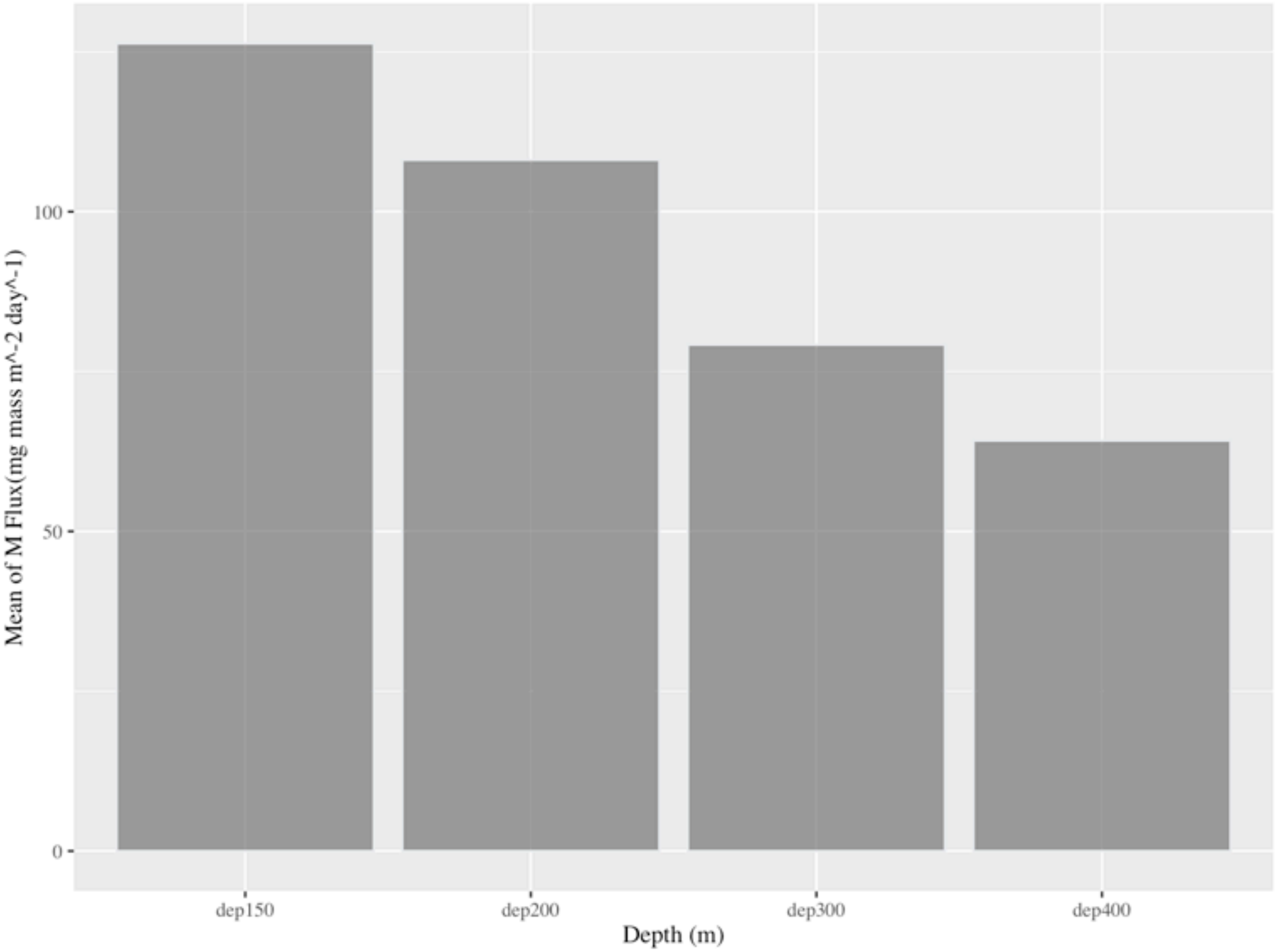


Mean and SE of each depth of M-flux

```
dep150m <- subset(M_flux.data,dep=="150") #focusing on mean of C-avg at depth 150

muMflux150 <- mean(dep150m$M_avg, data=dep150m)

s.e150m <- muMflux150 * sqrt(10)  #need to find s.e

dep200m <- subset(M_flux.data, dep=="200") #focusing on mean of C-avg at depth 200

muMflux200 <- mean(dep200m$M_avg, data=dep200m)

s.e200m <- muMflux200 * sqrt(10)

dep300m <- subset(M_flux.data, dep=="300") #focusing on mean of C-avg at depth 300

muMflux300 <- mean(dep300m$M_avg, data=dep300m)

s.e300m <- muMflux300 * sqrt(10)

dep400m <- subset(M_flux.data, dep=="400") #focusing on mean of C-avg at depth 400

muMflux400 <- mean(dep400m$M_avg, data=dep400m)

s.e400m <- muMflux400 * sqrt(10)

bar.Mflux <- data.frame(c("dep150","dep200","dep300","dep400"), c(muMflux150,muMflux2
00,muMflux300,muMflux400))
colnames(bar.Mflux)[1] <- "Depth"
colnames(bar.Mflux)[2] <- "M_fluxMean"
```

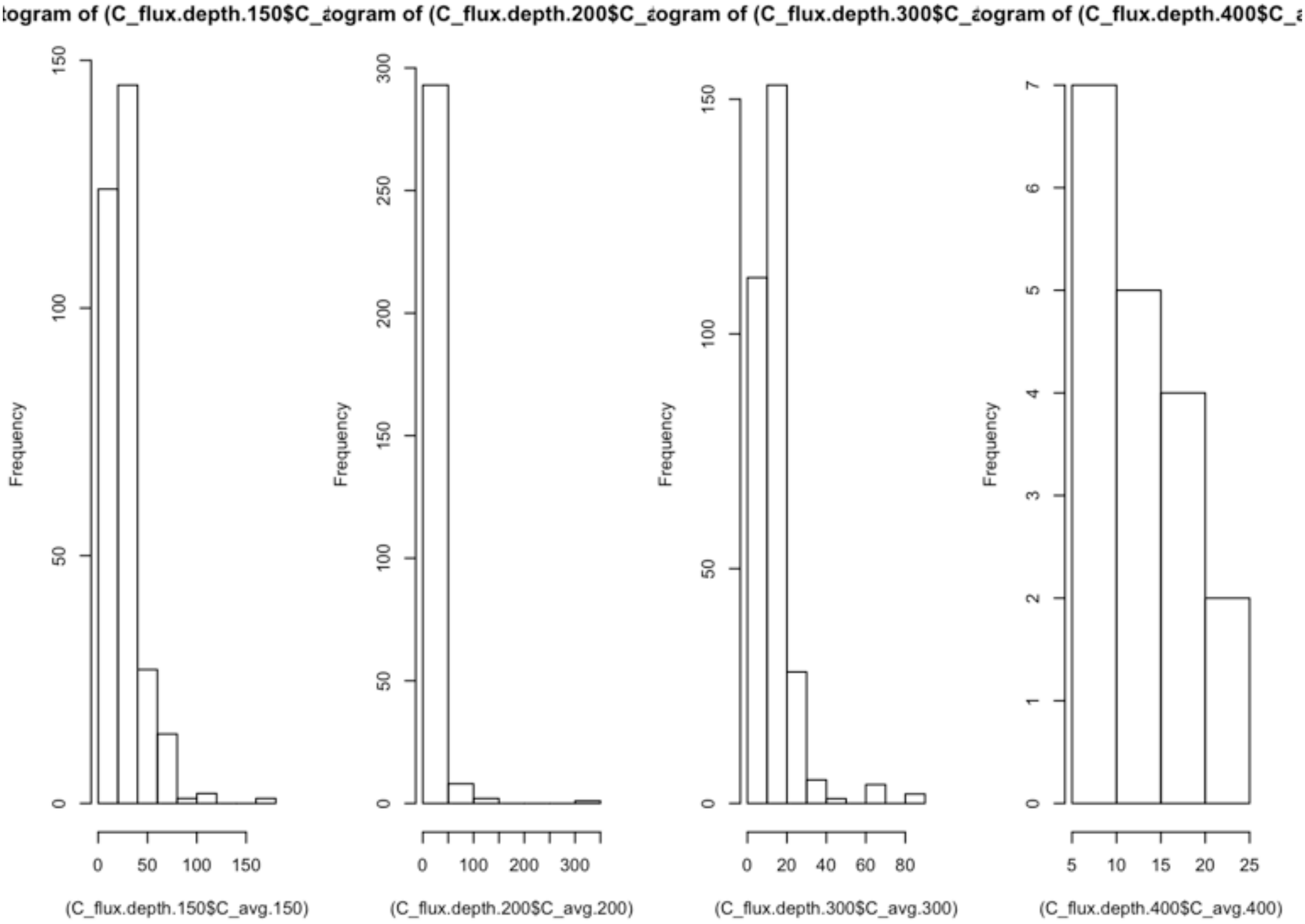
Graph of M-flux

```
library(ggplot2)
library(dplyr)
library(viridisLite)

Viridis_palette <- viridis(40)[seq(1,40, 2)]
xval <- bar.Mflux$Depth #character objects need quotes
yval <- bar.Mflux$`M_flux Mean`

ggplot()+geom_col(bar.Mflux, mapping=aes(x= Depth, y = M_fluxMean), color="#e9ecef",
alpha=0.6,) +
xlab("Depth (m)") + ylab("Mean of M Flux(mg mass m^-2 day^-1)") +

 theme(text = element_text(family = 'serif')) +
scale_fill_manual(values = Viridis_palette)
```



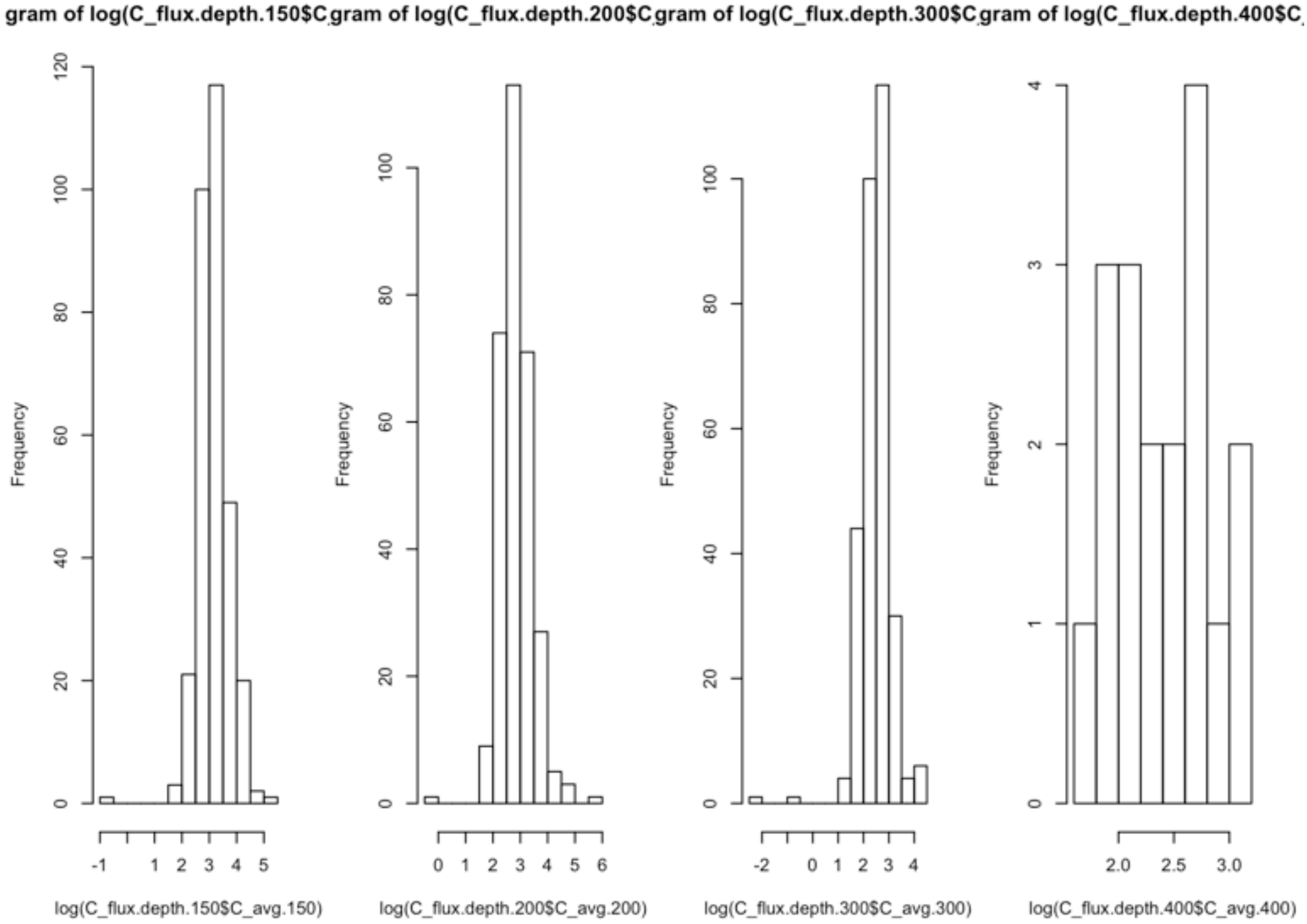
Distribution and box plot of our data by depth

```
# Checking for normaility and no outliers in Carbon Flux
par(mfrow=c(1,4))
hist((C_flux.depth.150$C_avg.150))
hist((C_flux.depth.200$C_avg.200))
hist((C_flux.depth.300$C_avg.300))
hist((C_flux.depth.400$C_avg.400))
```

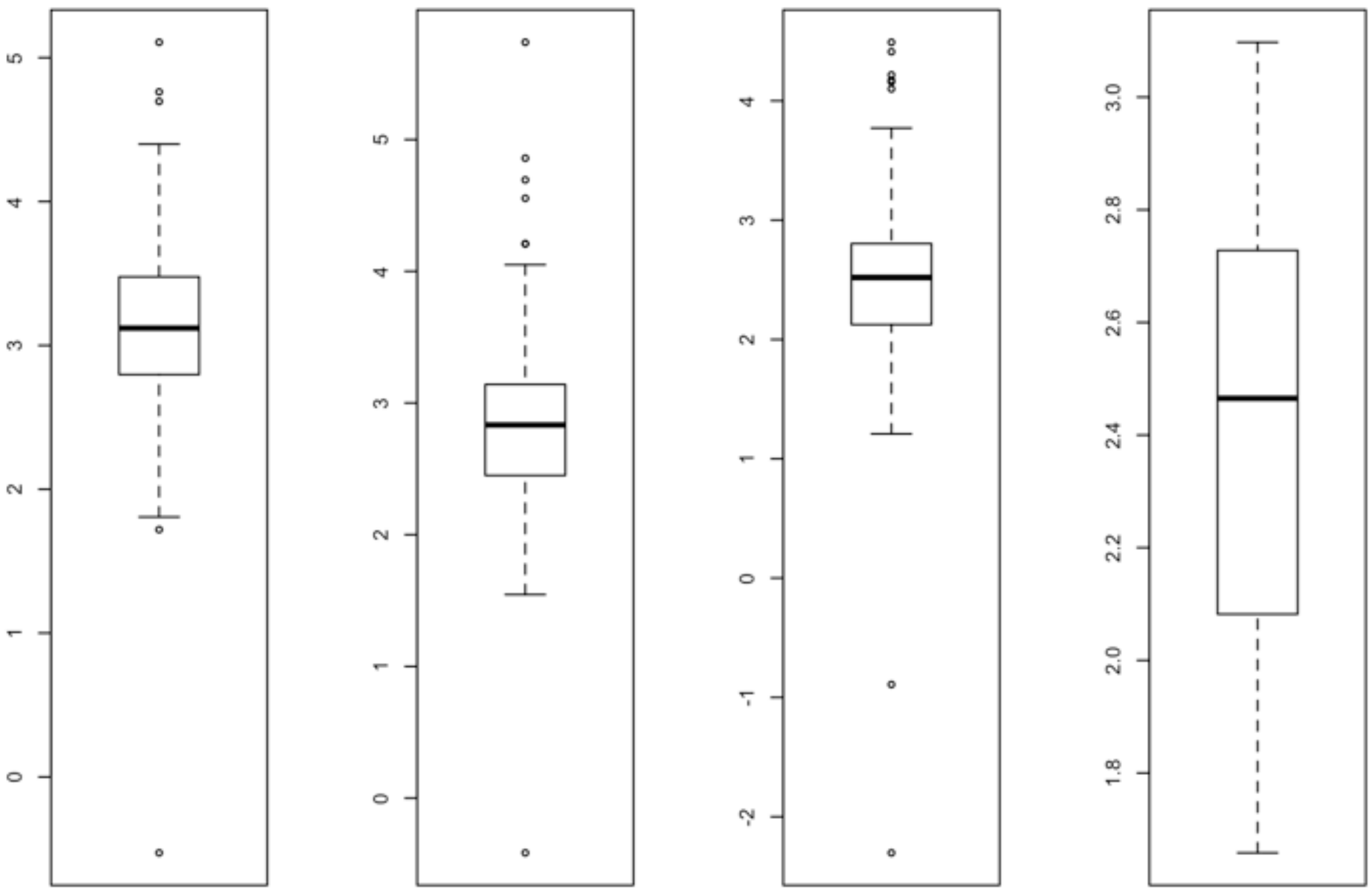


togram of (C_flux.depth.150$C_ togram of (C_flux.depth.200$C_ togram of (C_flux.depth.300$C_ togram of (C_flux.depth.400$C_

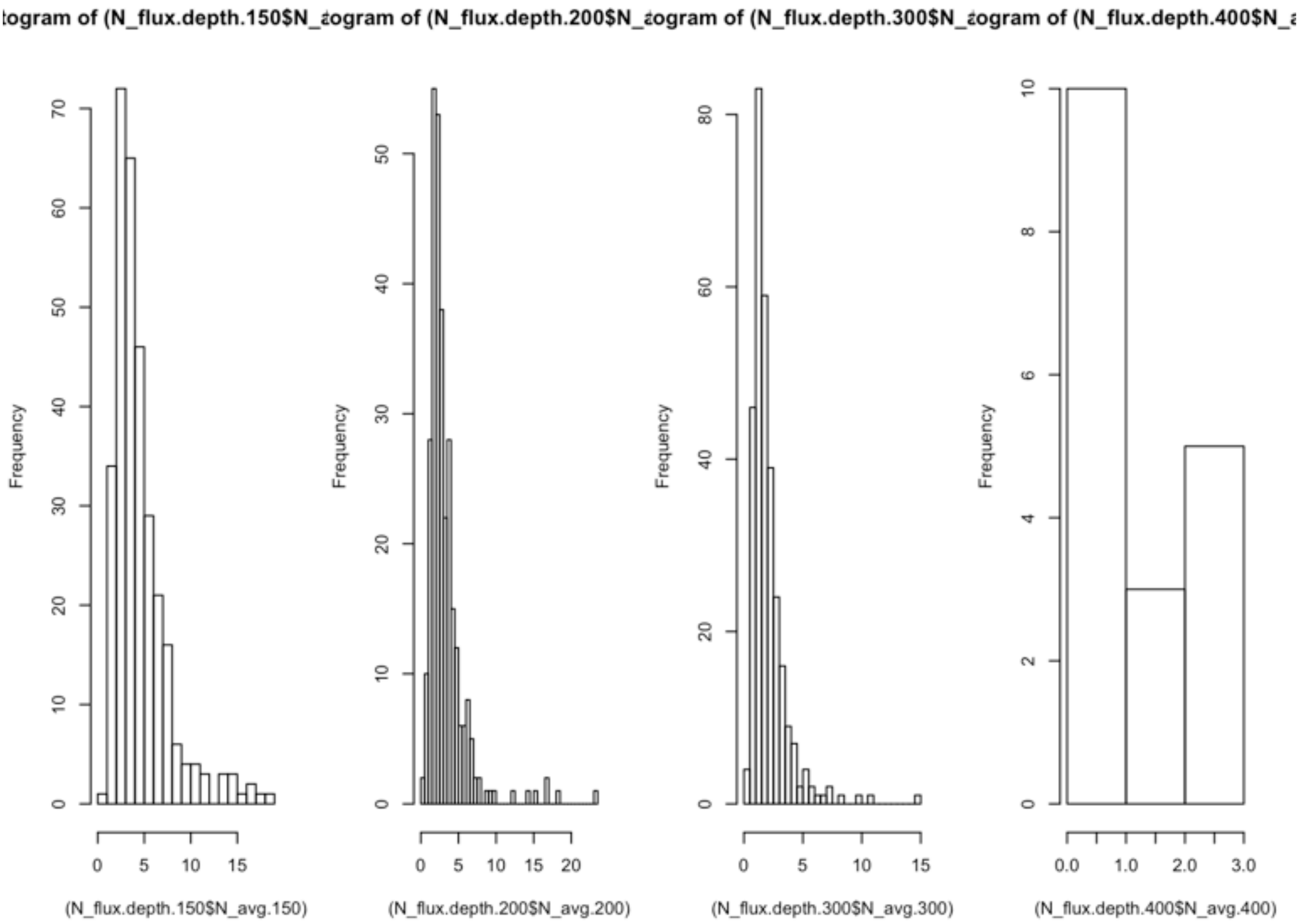
```
# using natural log to Transform the data
par(mfrow=c(1,4))
hist(log(C_flux.depth.150$C_avg.150))
hist(log(C_flux.depth.200$C_avg.200))
hist(log(C_flux.depth.300$C_avg.300))
hist(log(C_flux.depth.400$C_avg.400))
```



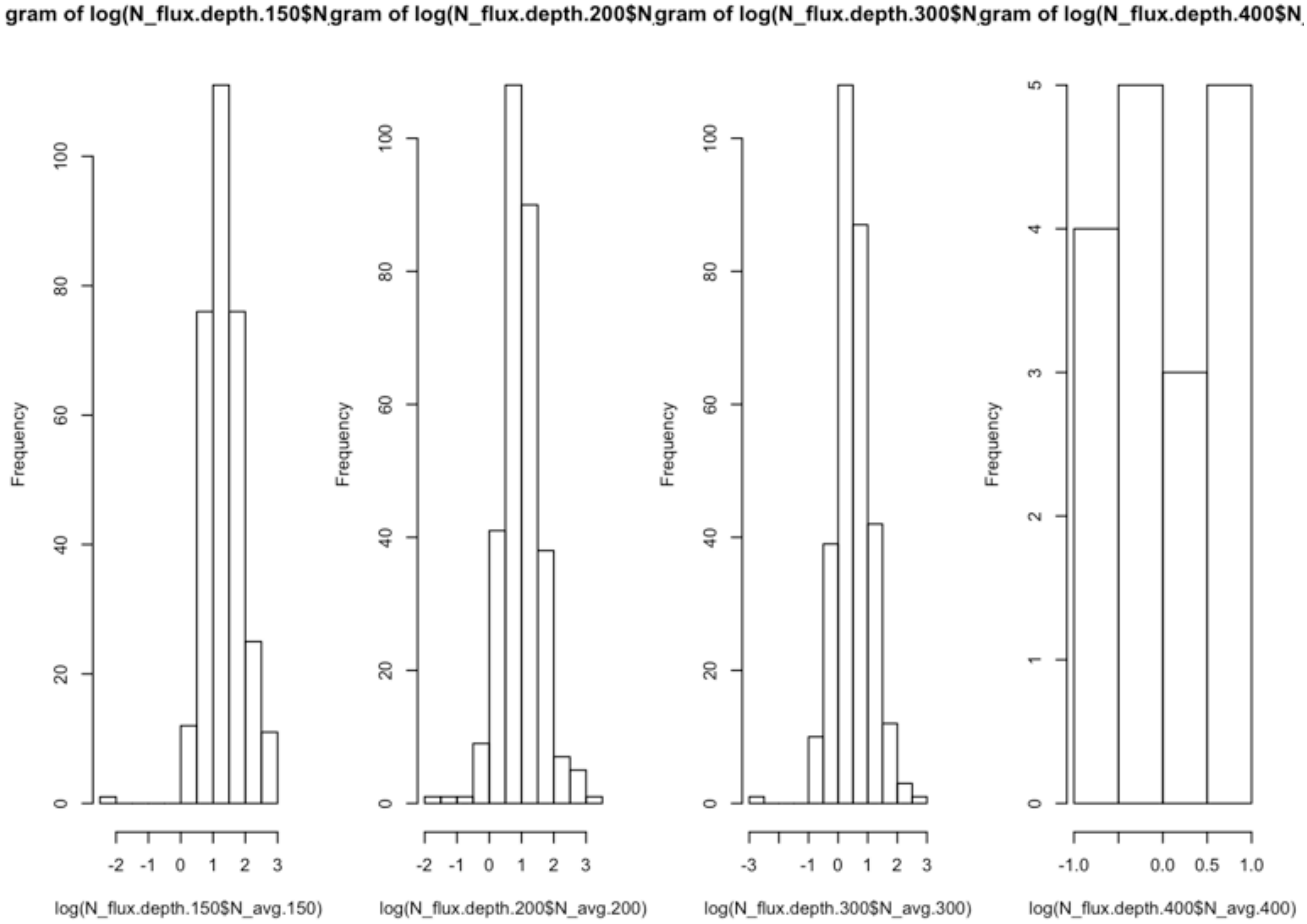
```
# Checking for outliers
par(mfrow=c(1,4))
boxplot(log(C_flux.depth.150$C_avg.150))
boxplot(log(C_flux.depth.200$C_avg.200))
boxplot(log(C_flux.depth.300$C_avg.300))
boxplot(log(C_flux.depth.400$C_avg.400))
```



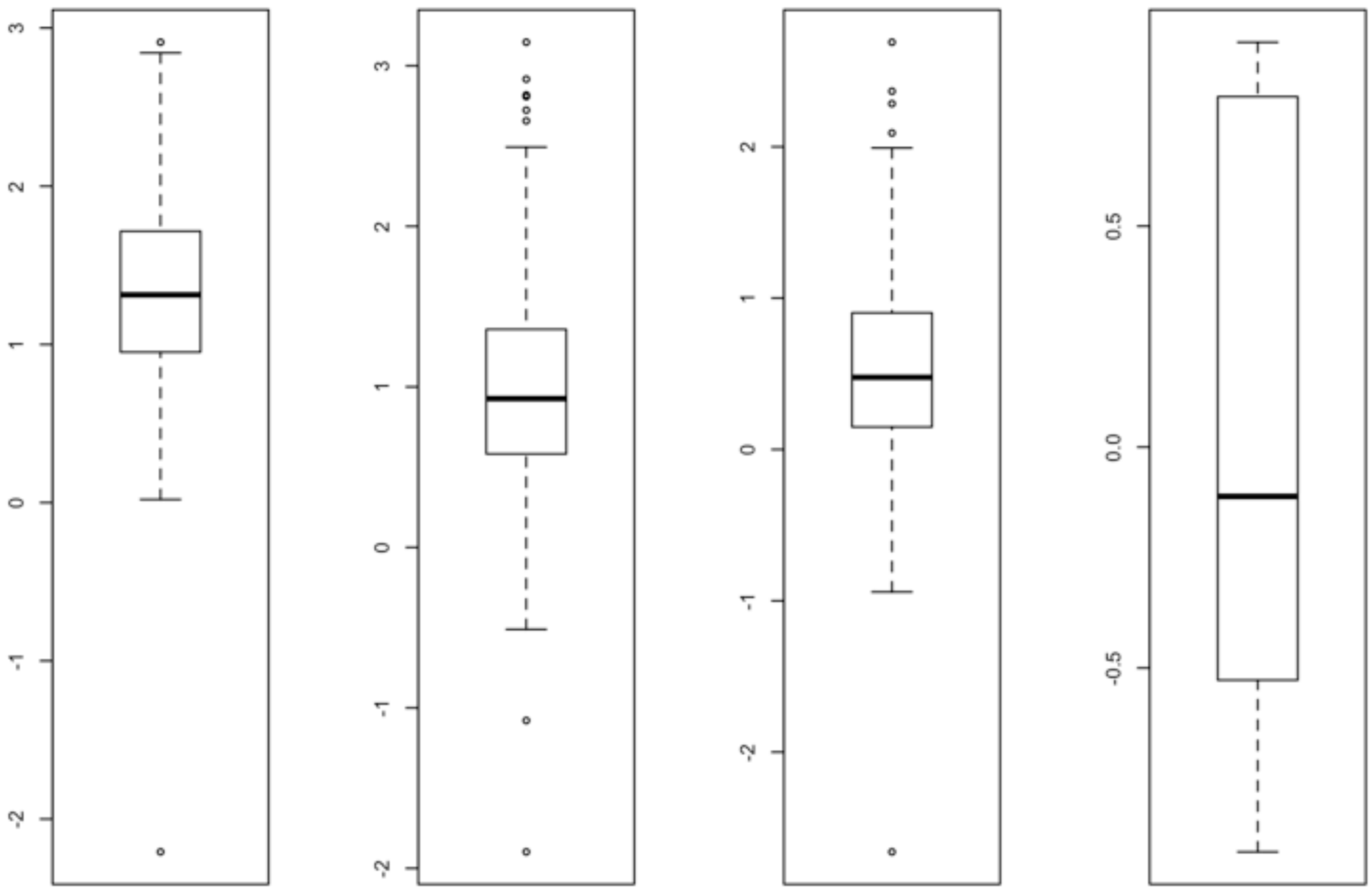
```
# Checking for normaility and no outliers in Nitrogen Flux
par(mfrow=c(1,4))
hist((N_flux.depth.150$N_avg.150), breaks = "FD")
hist((N_flux.depth.200$N_avg.200), breaks = "FD")
hist((N_flux.depth.300$N_avg.300), breaks = "FD")
hist((N_flux.depth.400$N_avg.400), breaks = "FD")
```



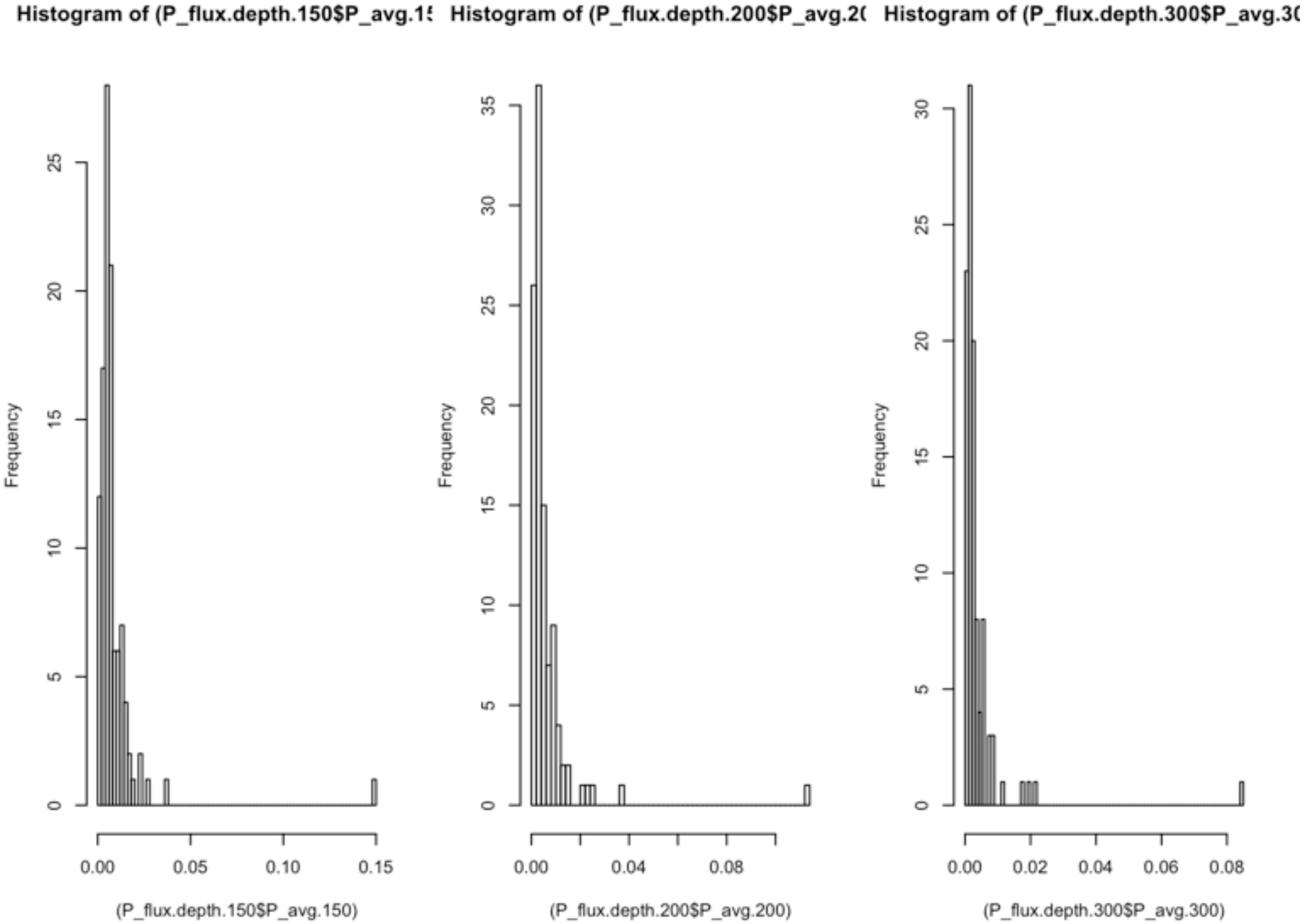
```
# using natural log to Transform the data
par(mfrow=c(1,4))
hist(log(N_flux.depth.150$N_avg.150))
hist(log(N_flux.depth.200$N_avg.200))
hist(log(N_flux.depth.300$N_avg.300))
hist(log(N_flux.depth.400$N_avg.400))
```



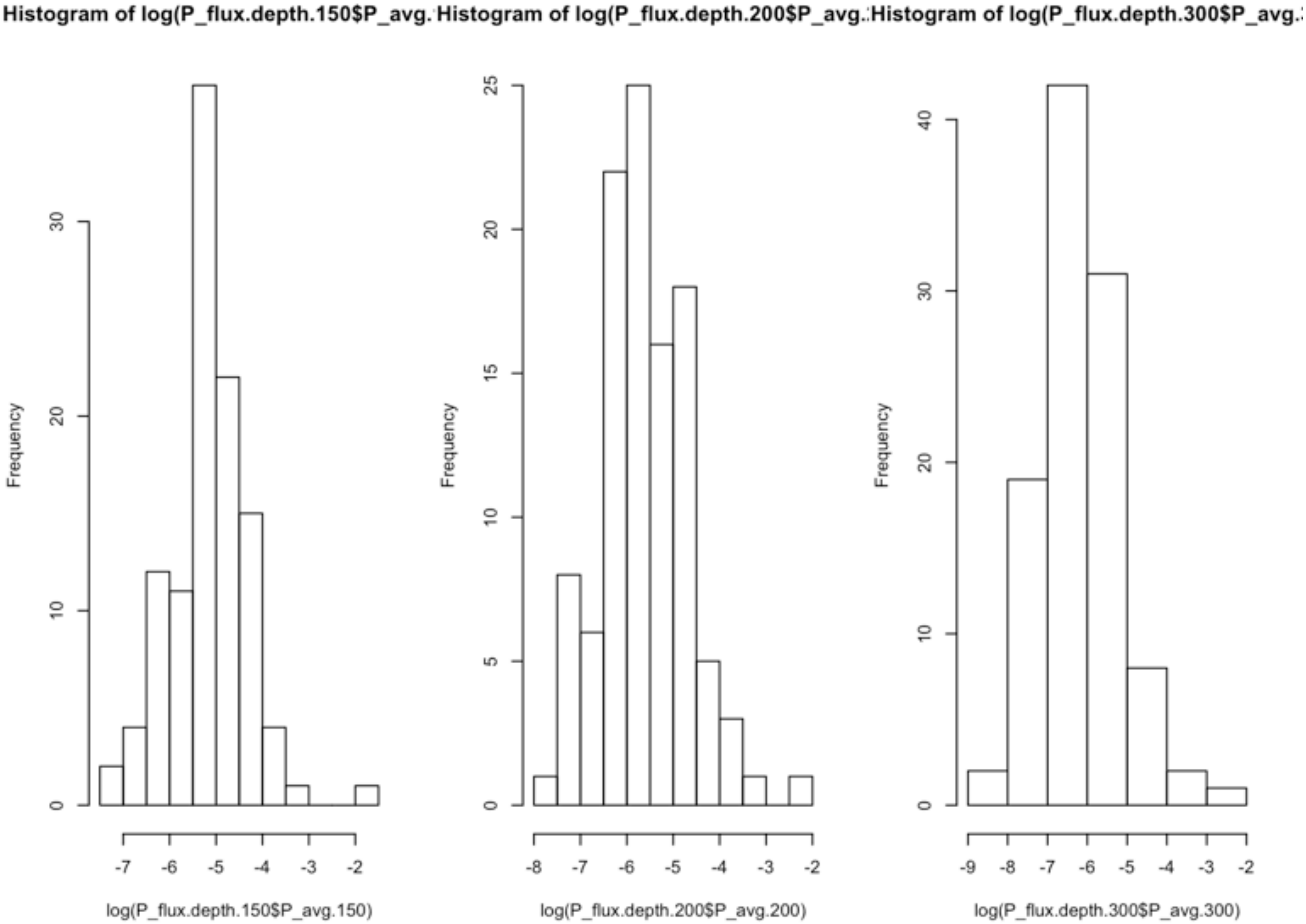
```
# Checking for outliers
par(mfrow=c(1,4))
boxplot(log(N_flux.depth.150$N_avg.150))
boxplot(log(N_flux.depth.200$N_avg.200))
boxplot(log(N_flux.depth.300$N_avg.300))
boxplot(log(N_flux.depth.400$N_avg.400))
```



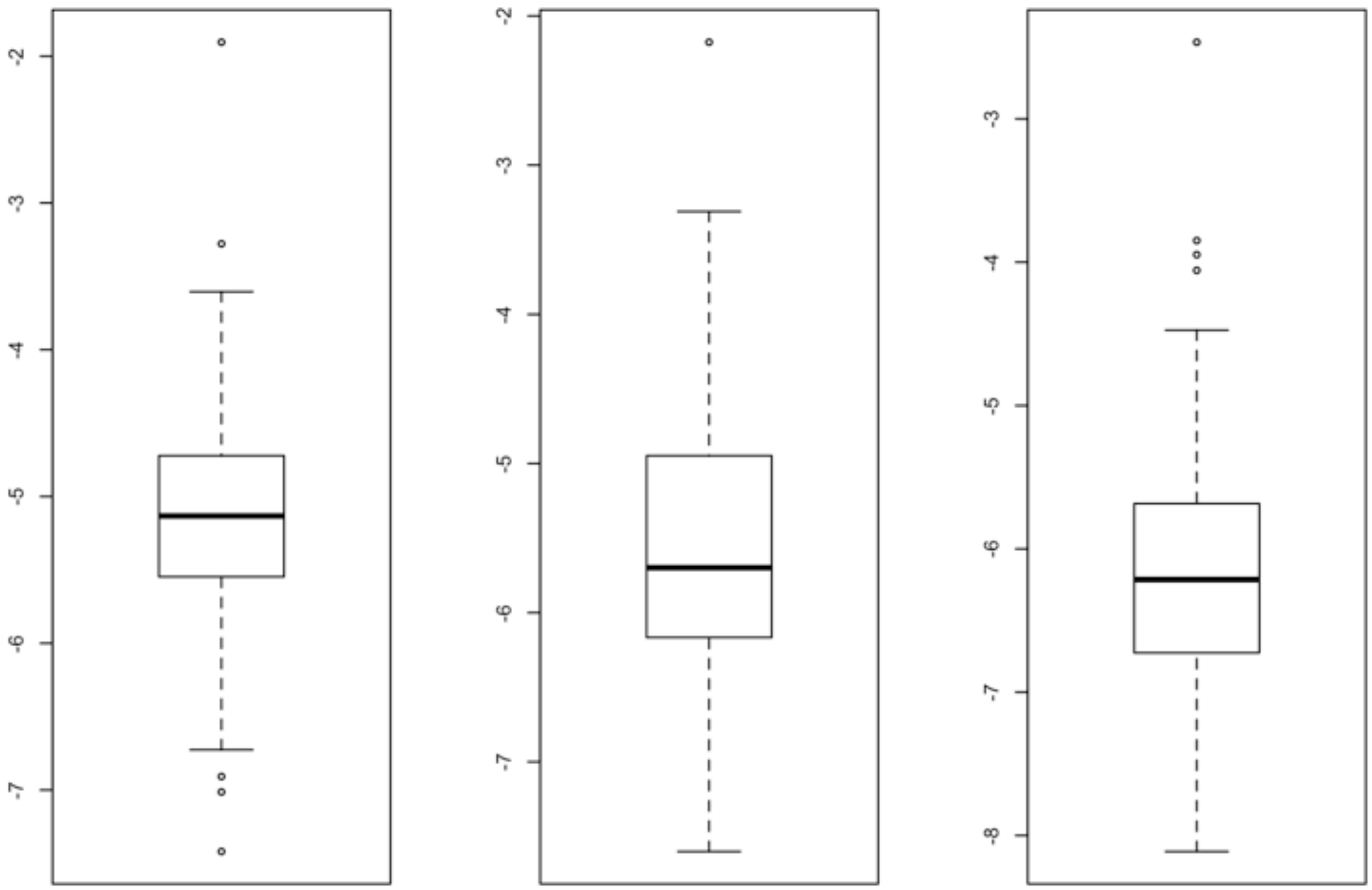
```
# Checking for normaility and no outliers in Phosporous Flux
par(mfrow=c(1,3))
hist((P_flux.depth.150$P_avg.150), breaks = "FD")
hist((P_flux.depth.200$P_avg.200), breaks = "FD")
hist((P_flux.depth.300$P_avg.300), breaks = "FD")
```



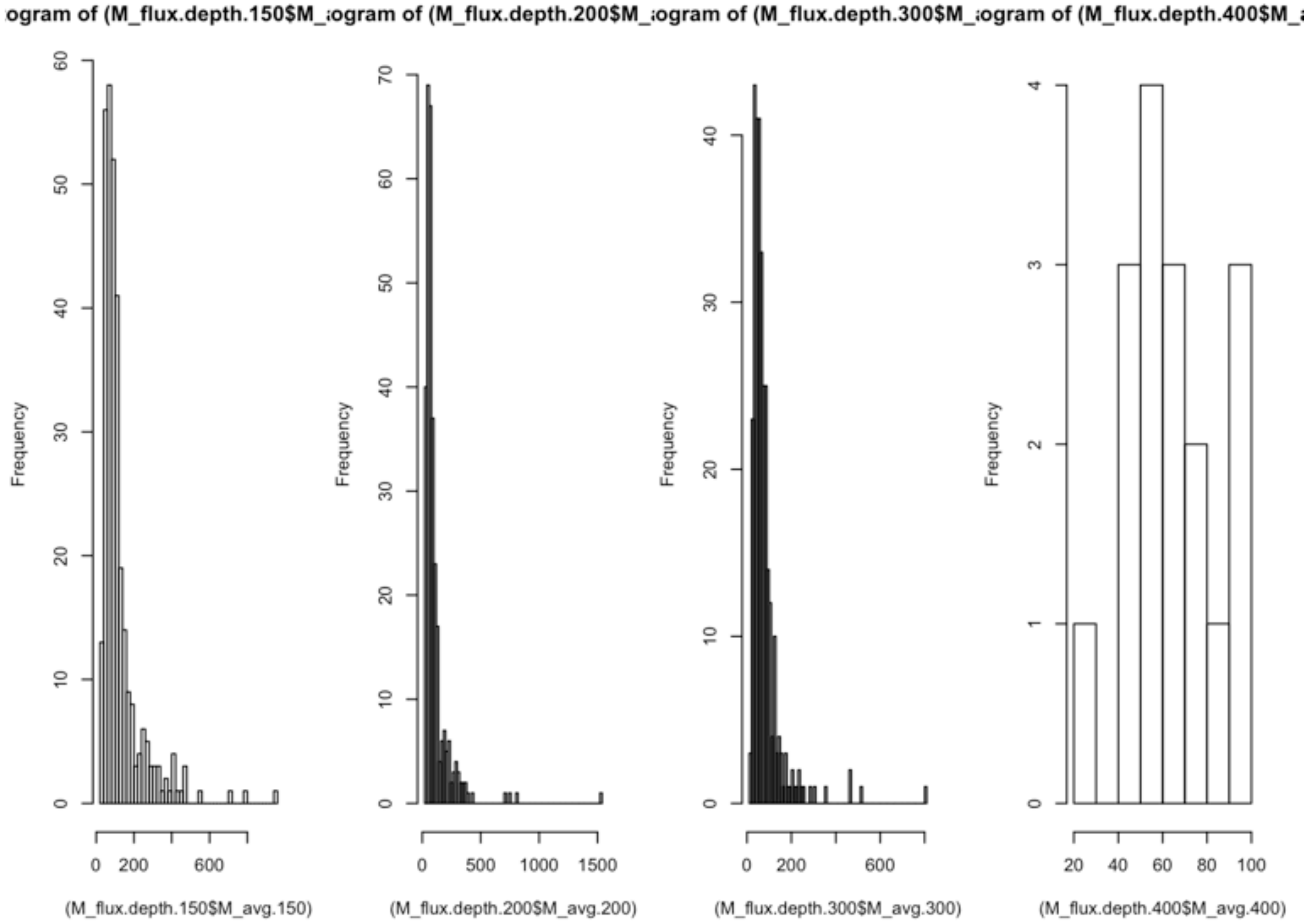
```
# using natural log to Transform the data
par(mfrow=c(1,3))
hist(log(P_flux.depth.150$P_avg.150))
hist(log(P_flux.depth.200$P_avg.200))
hist(log(P_flux.depth.300$P_avg.300))
```



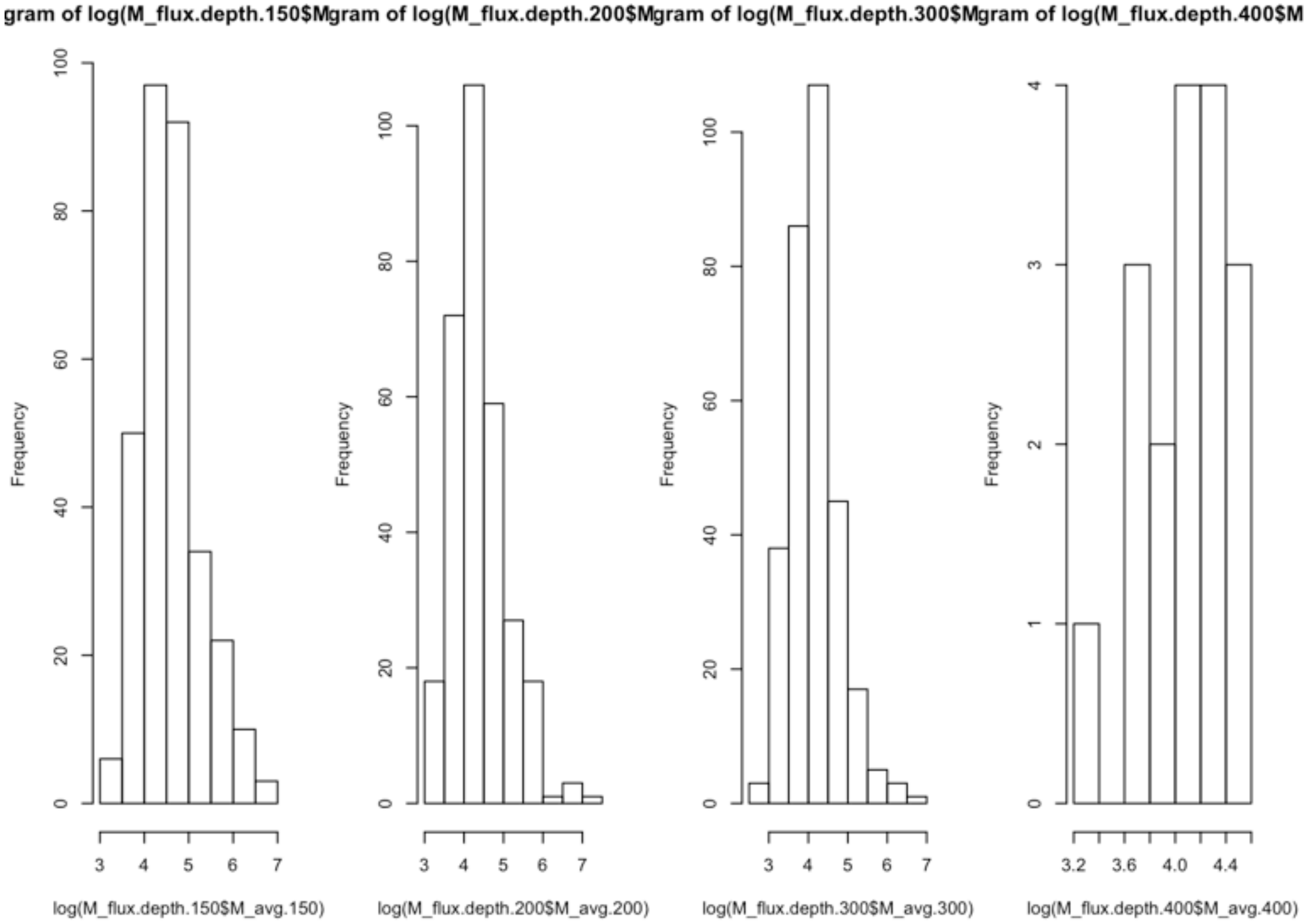
```
# Checking for outliers
par(mfrow=c(1,3))
boxplot(log(P_flux.depth.150$P_avg.150))
boxplot(log(P_flux.depth.200$P_avg.200))
boxplot(log(P_flux.depth.300$P_avg.300))
```



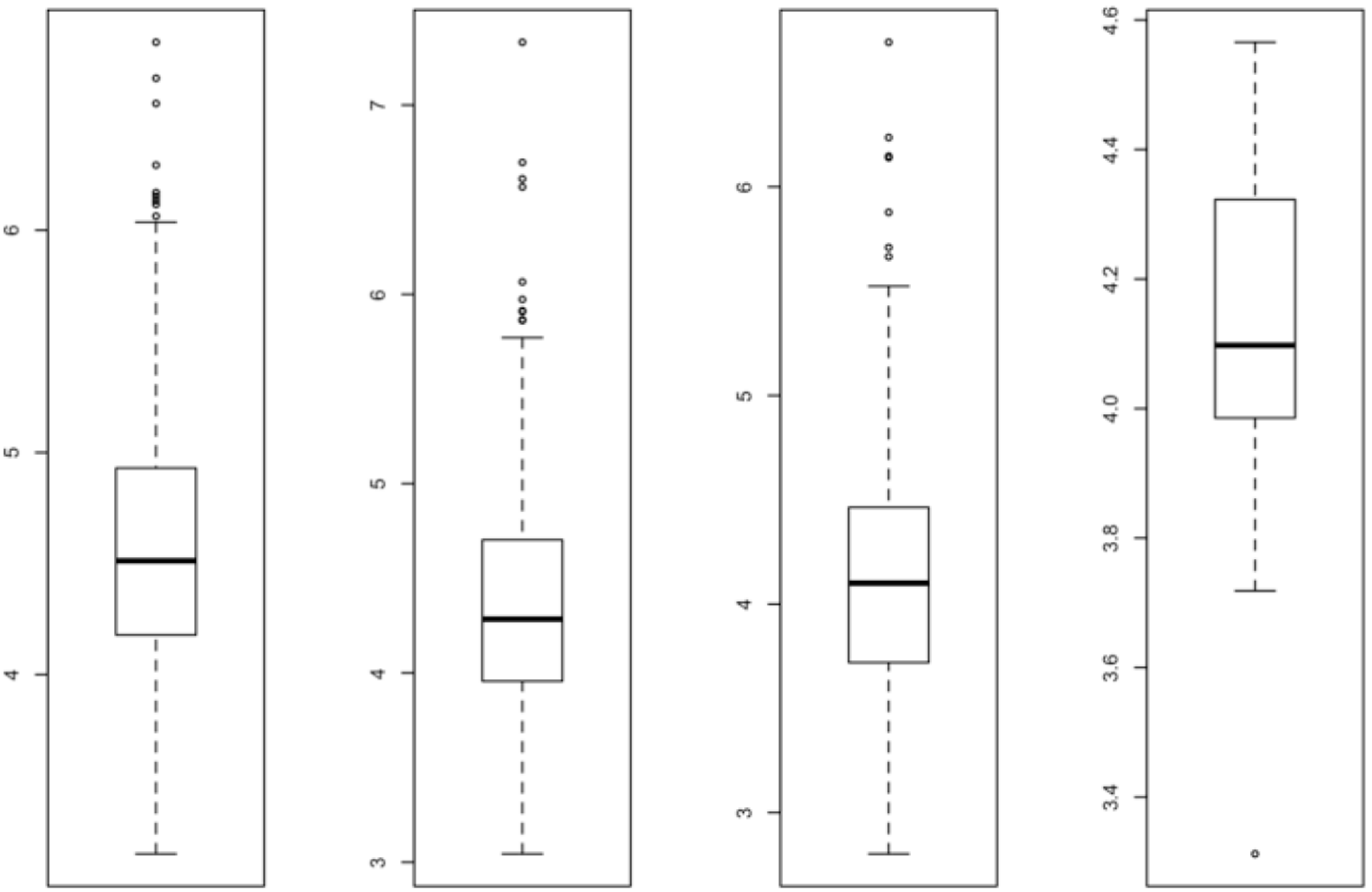
```r
# Checking for normaility and no outliers in Mass Flux
par(mfrow=c(1,4))
hist((M_flux.depth.150$M_avg.150), breaks = "FD")
hist((M_flux.depth.200$M_avg.200), breaks = "FD")
hist((M_flux.depth.300$M_avg.300), breaks = "FD")
hist((M_flux.depth.400$M_avg.400), breaks = "FD")
```



```
# using natural log to Transform the data
par(mfrow=c(1,4))
hist(log(M_flux.depth.150$M_avg.150))
hist(log(M_flux.depth.200$M_avg.200))
hist(log(M_flux.depth.300$M_avg.300))
hist(log(M_flux.depth.400$M_avg.400))
```



```
# Checking for outliers
par(mfrow=c(1,4))
boxplot(log(M_flux.depth.150$M_avg.150))
boxplot(log(M_flux.depth.200$M_avg.200))
boxplot(log(M_flux.depth.300$M_avg.300))
boxplot(log(M_flux.depth.400$M_avg.400))
```



Since many of the 400 m depth distrubutions were not normal, even after transformation and have small n, they will not be used in the hypothesis testing

Paired sample t test for each depth and the depth below it

This is a paired test because samples were collected on the same cruise and at the same location. So for each flux value at one depth, there was a flux value collected at a different depth at the same location! Data frames were merged based on cruise, year, and location to have the same number of values.

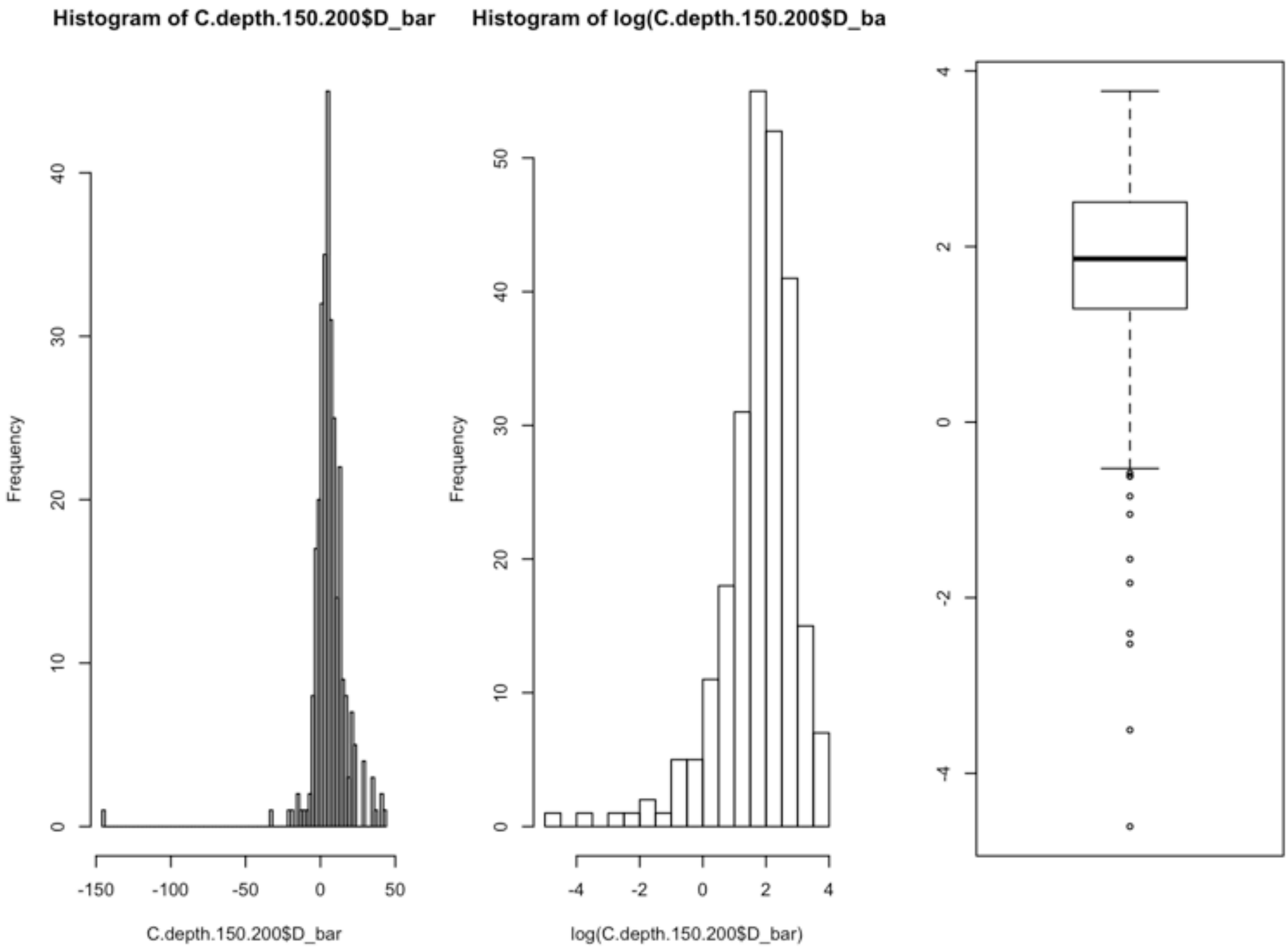
For a paired t test to be preformed it must meet these assumptions:

$\overline{Y_1}$ - $\overline{Y_2}$ $= \overline{D}$

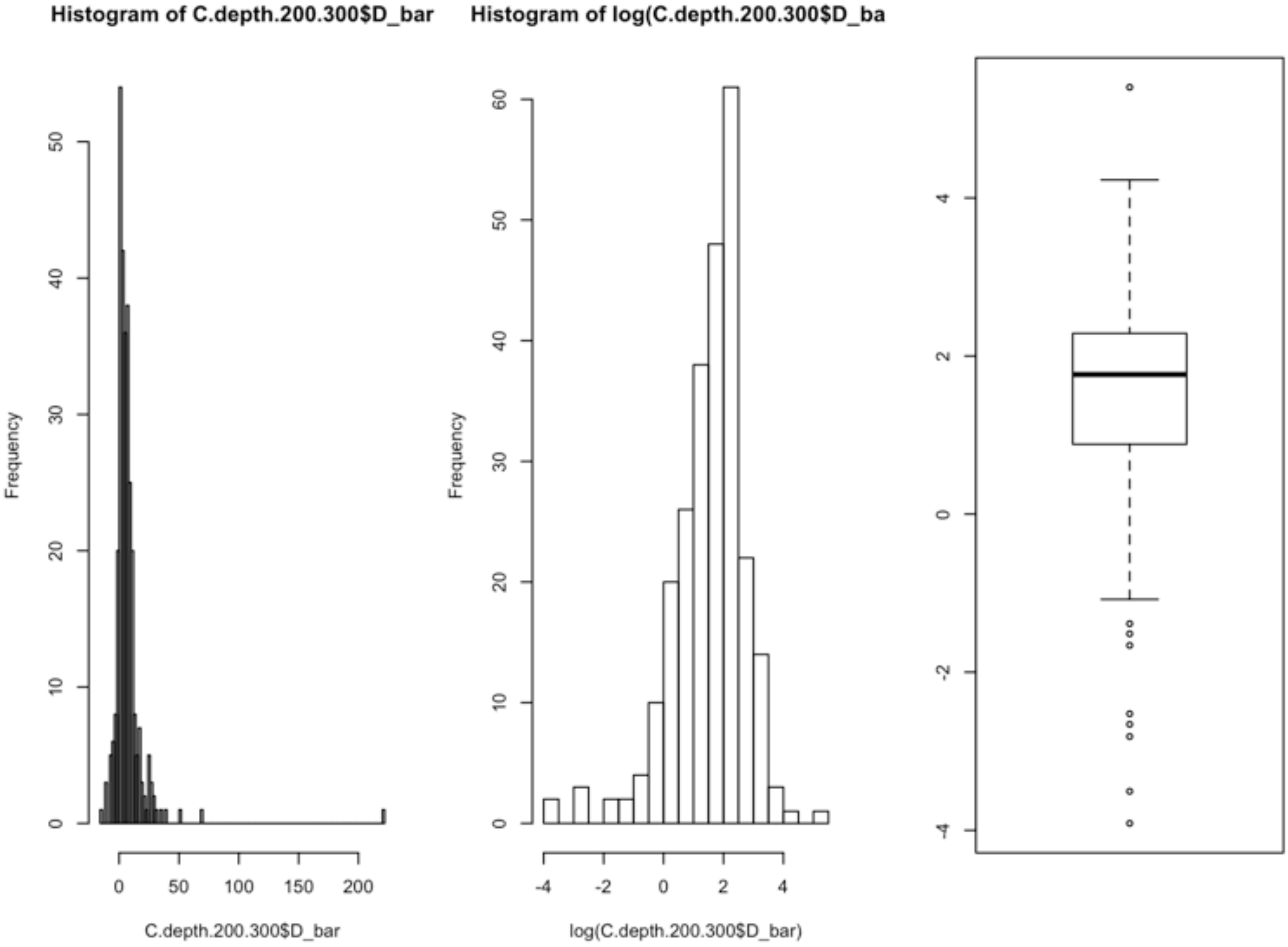
-The dependent variable must be continuous (interval/ratio). ✓ -The observations are independent of one another. ✓ -The dependent variable ($\overline{D}$) should be approximately normally distributed. ✓ -The dependent variable ($\overline{D}$) should not contain any outliers. □

```
#create D-bar for each paired group
C.depth.150.200 <- merge(C_flux.depth.150,C_flux.depth.200,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
C.depth.150.200$D_bar <- C.depth.150.200$C_avg.150 - C.depth.150.200$C_avg.200

C.depth.200.300 <- merge(C_flux.depth.200,C_flux.depth.300,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
C.depth.200.300$D_bar <- C.depth.200.300$C_avg.200 - C.depth.200.300$C_avg.300

N.depth.150.200 <- merge(N_flux.depth.150,N_flux.depth.200,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
N.depth.150.200$D_bar <- N.depth.150.200$N_avg.150 - N.depth.150.200$N_avg.200

N.depth.200.300 <- merge(N_flux.depth.200,N_flux.depth.300,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
N.depth.200.300$D_bar <- N.depth.200.300$N_avg.200 - N.depth.200.300$N_avg.300

P.depth.150.200 <- merge(P_flux.depth.150,P_flux.depth.200,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
P.depth.150.200$D_bar <- P.depth.150.200$P_avg.150 - P.depth.150.200$P_avg.200

P.depth.200.300 <- merge(P_flux.depth.200,P_flux.depth.300,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
P.depth.200.300$D_bar <- P.depth.200.300$P_avg.200 - P.depth.200.300$P_avg.300

M.depth.150.200 <- merge(M_flux.depth.150,M_flux.depth.200,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
M.depth.150.200$D_bar <- M.depth.150.200$M_avg.150 - M.depth.150.200$M_avg.200

M.depth.200.300 <- merge(M_flux.depth.200,M_flux.depth.300,by = c("cr","yymmdd1","yymmdd2","Lat2","Lat2.1","Long1","Long2"))
M.depth.200.300$D_bar <- M.depth.200.300$M_avg.200 - M.depth.200.300$M_avg.300
```

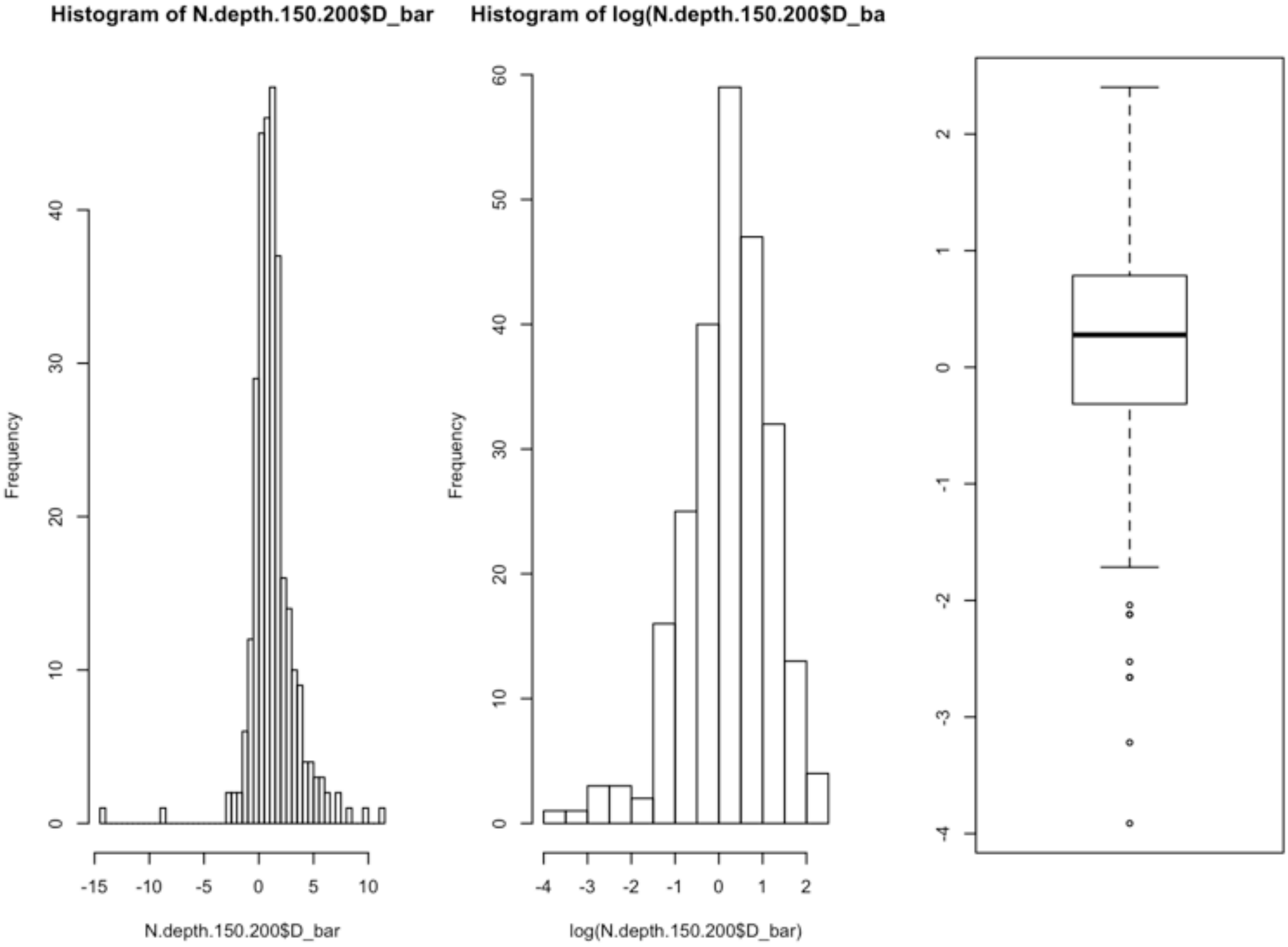
```
#check distribution of D-bar (difference between paried depth samples)

# Carbon
par(mfrow=c(1,3))
hist(C.depth.150.200$D_bar, breaks = "FD")
hist(log(C.depth.150.200$D_bar), breaks = "FD") #try lo transform
boxplot(log(C.depth.150.200$D_bar)) #lots of outliers
```



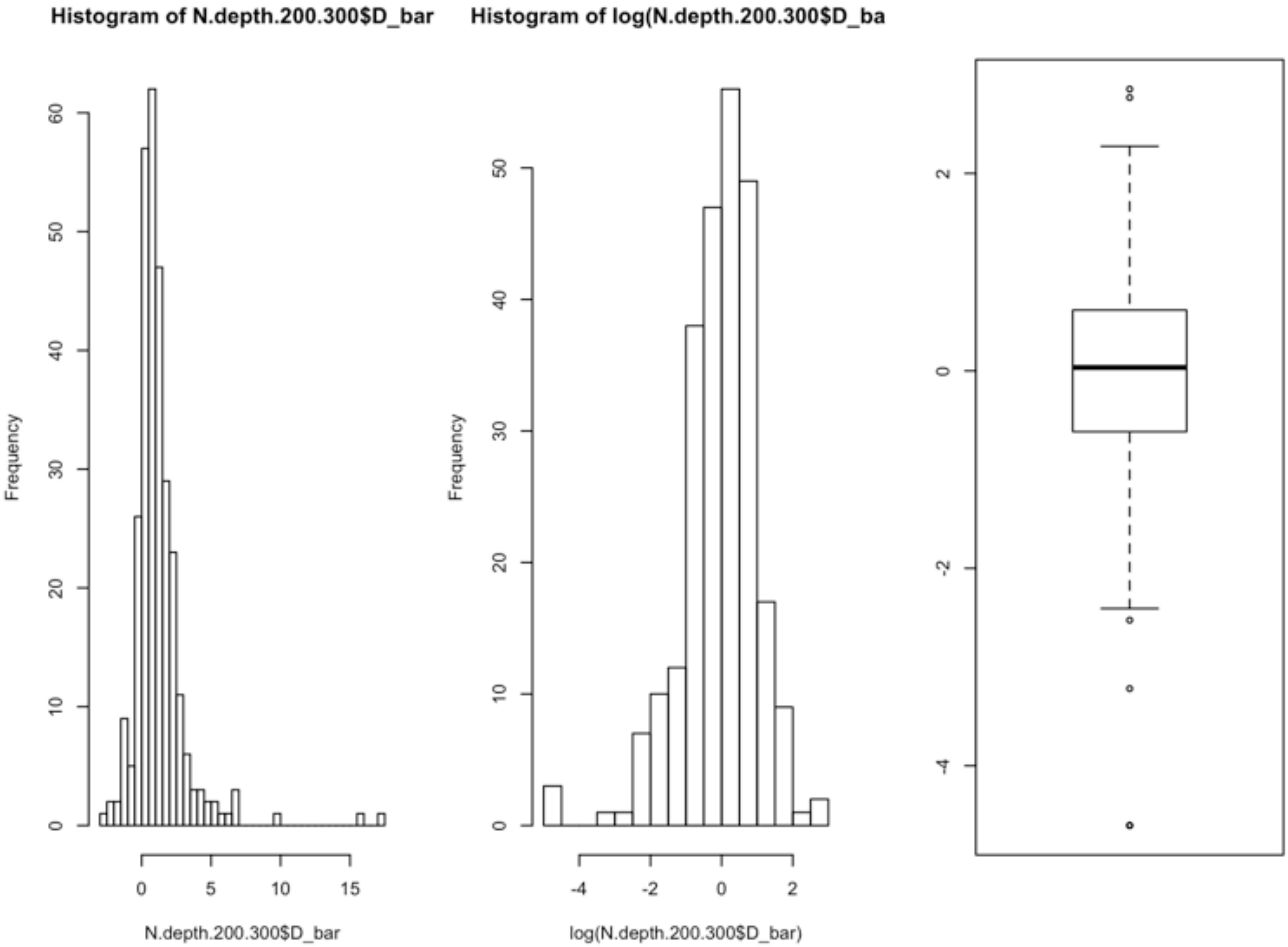
```
par(mfrow=c(1,3))
hist(C.depth.200.300$D_bar, breaks = "FD")
hist(log(C.depth.200.300$D_bar), breaks = "FD") #try lo transform
boxplot(log(C.depth.200.300$D_bar)) #lots of outliers
```



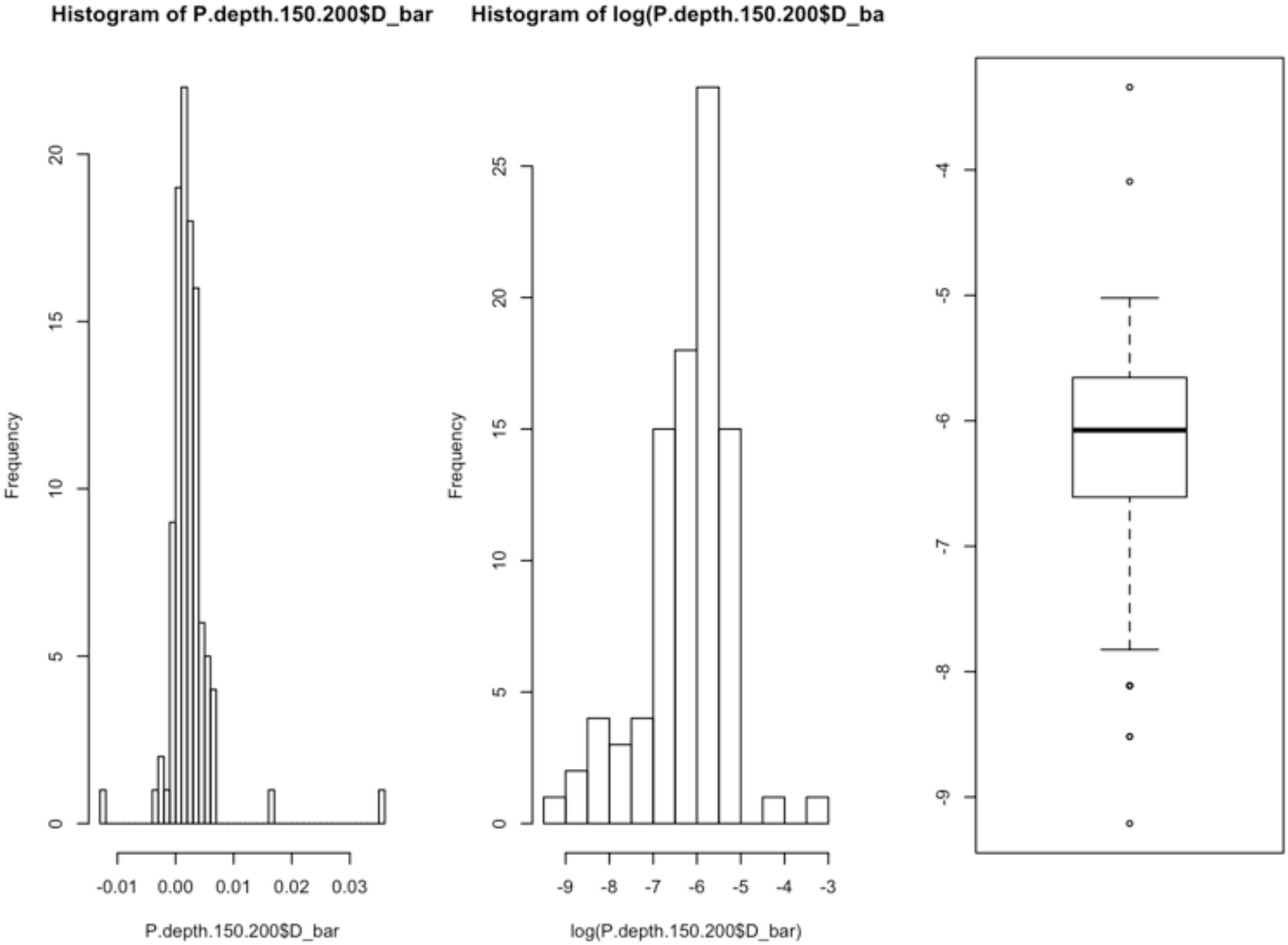
```
# Nitrogen
par(mfrow=c(1,3))
hist(N.depth.150.200$D_bar, breaks = "FD")
hist(log(N.depth.150.200$D_bar), breaks = "FD") #try lo transform
boxplot(log(N.depth.150.200$D_bar)) #lots of outliers
```



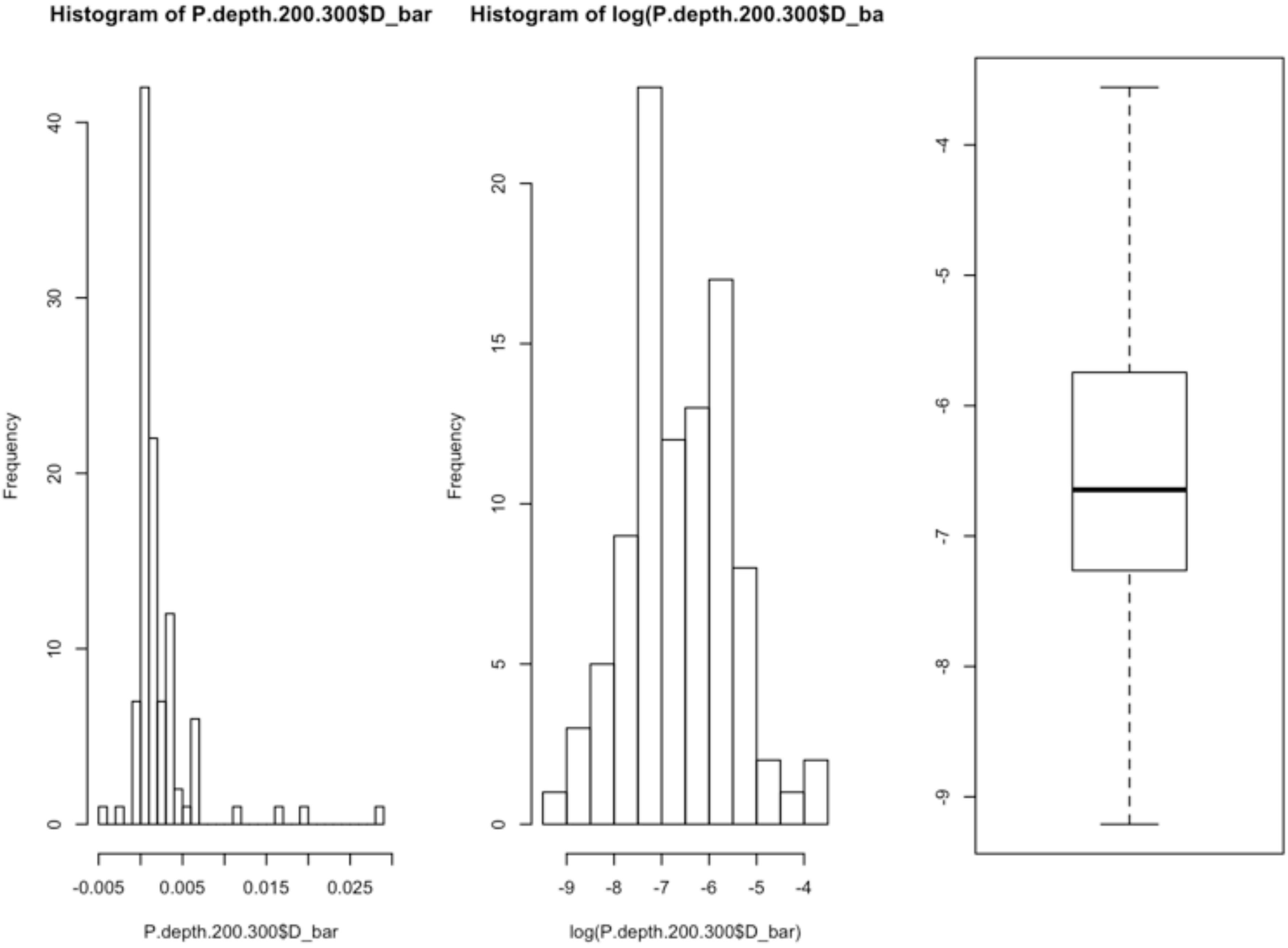
```
par(mfrow=c(1,3))
hist(N.depth.200.300$D_bar, breaks = "FD")
hist(log(N.depth.200.300$D_bar), breaks = "FD") #try lo transform
boxplot(log(N.depth.200.300$D_bar)) #lots of outliers
```



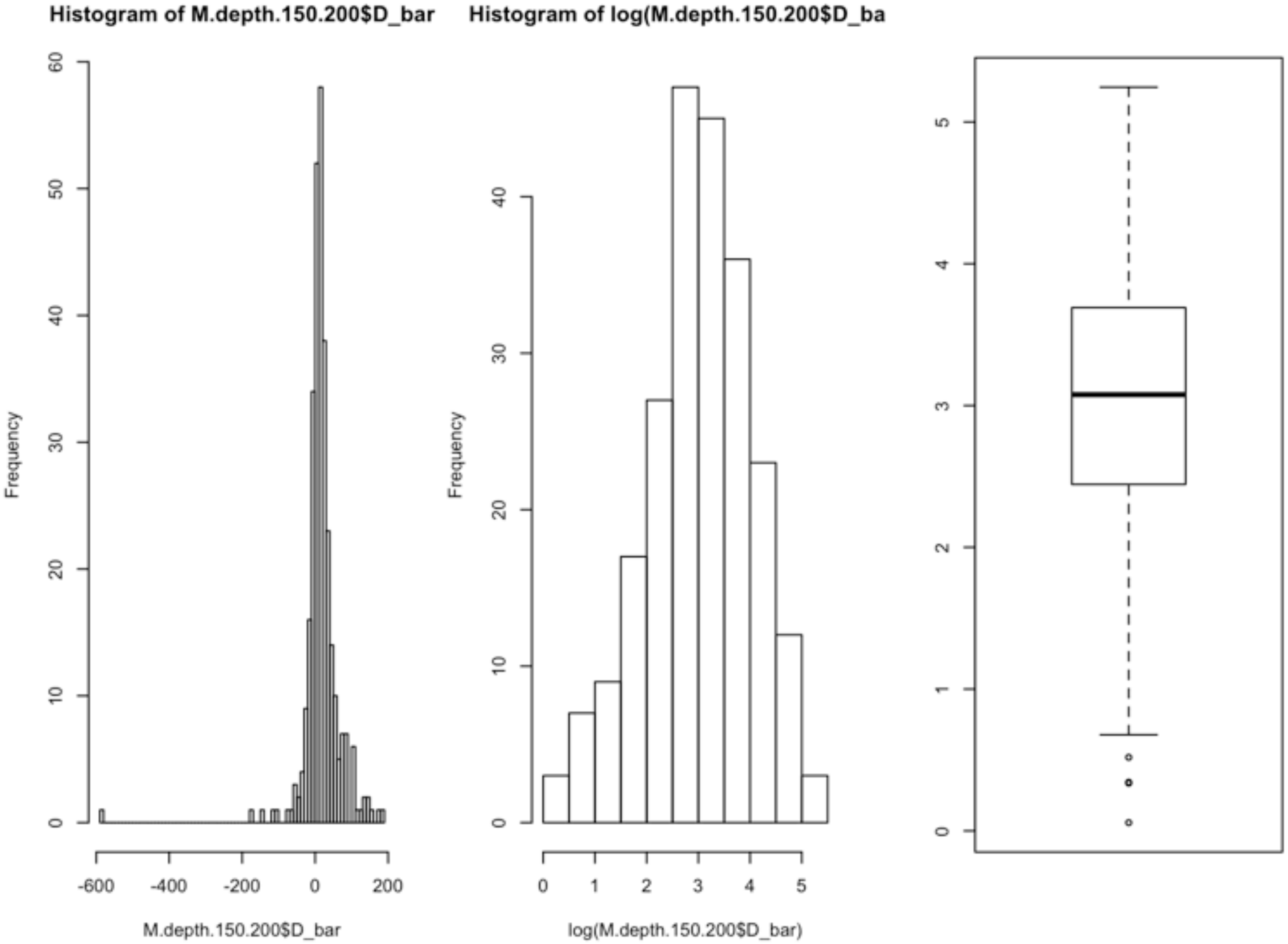
```
# Phosphorous
par(mfrow=c(1,3))
hist(P.depth.150.200$D_bar, breaks = "FD")
hist(log(P.depth.150.200$D_bar), breaks = "FD") #try lo transform
boxplot(log(P.depth.150.200$D_bar)) #lots of outliers
```



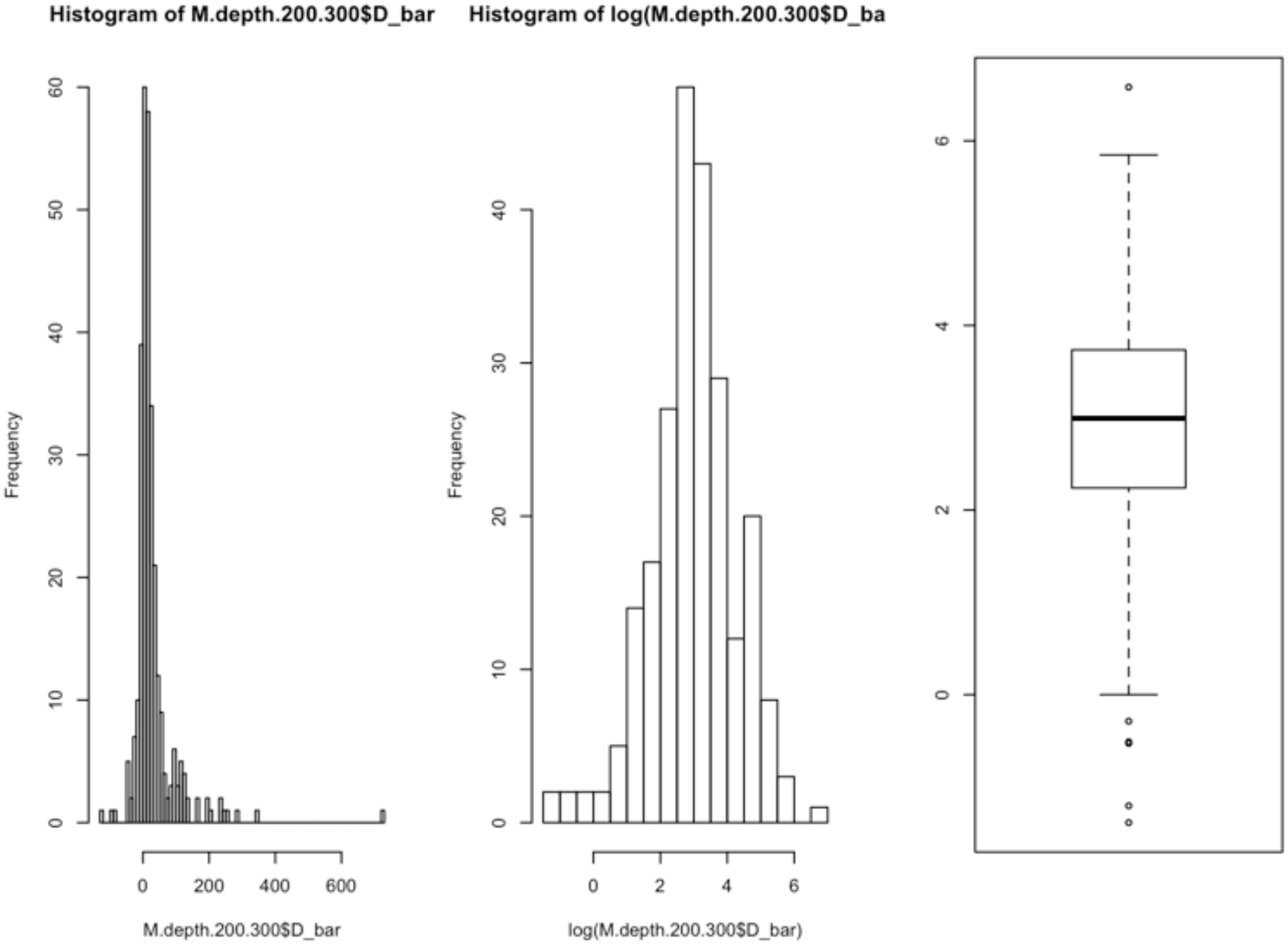
```
par(mfrow=c(1,3))
hist(P.depth.200.300$D_bar, breaks = "FD")
hist(log(P.depth.200.300$D_bar), breaks = "FD") #try lo transform
boxplot(log(P.depth.200.300$D_bar)) #lots of outliers
```



```
# Mass
par(mfrow=c(1,3))
hist(M.depth.150.200$D_bar, breaks = "FD")
hist(log(M.depth.150.200$D_bar), breaks = "FD") #try lo transform
boxplot(log(M.depth.150.200$D_bar)) #lots of outliers
```



```
par(mfrow=c(1,3))
hist(M.depth.200.300$D_bar, breaks = "FD")
hist(log(M.depth.200.300$D_bar), breaks = "FD") #try lo transform
boxplot(log(M.depth.200.300$D_bar)) #lots of outliers
```



(1.1)

μ_1 = mean Carbon flux at depth 150 μ_2 = mean Carbon flux at depth 200

H_0 : Carbon flux does not change between 150 and 200 meter. $\mu_1 = \mu_2$ H_1 : Carbon flux is lower at 200 meter then at 150 meter. $\mu_1 > \mu_2$

```
t.test(log(C.depth.150.200$C_avg.150) ,log(C.depth.150.200$C_avg.200), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(C.depth.150.200$C_avg.150) and log(C.depth.150.200$C_avg.200)
## t = 14.668, df = 302, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2578256       Inf
## sample estimates:
## mean of the differences
##               0.2905015
```

(1.2)

μ_1 = mean Carbon flux at depth 200 μ_2 = mean Carbon flux at depth 300

H_0 : Carbon flux does not change between 200 and 300 meter. $\mu_1 = \mu_2$ H_1 : Carbon flux is lower at 300 meter then at 200 meter. $\mu_1 > \mu_2$

```
t.test(log(C.depth.200.300$C_avg.200) ,log(C.depth.200.300$C_avg.300), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(C.depth.200.300$C_avg.200) and log(C.depth.200.300$C_avg.300)
## t = 13.324, df = 299, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3254642       Inf
## sample estimates:
## mean of the differences
##               0.3714623
```

(2.1)

μ_1 = mean Nitrogen flux at depth 150 μ_2 = mean Nitrogen flux at depth 200

H_0 : Nitrogen flux does not change between 150 and 200 meter. $\mu_1 = \mu_2$ H_1 : Nitrogen flux is lower at 200 meter then at 150 meter. $\mu_1 > \mu_2$

```
t.test(log(N.depth.150.200$N_avg.150) ,log(N.depth.150.200$N_avg.200), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(N.depth.150.200$N_avg.150) and log(N.depth.150.200$N_avg.200)
## t = 14.918, df = 300, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3202794       Inf
## sample estimates:
## mean of the differences
##                0.360109
```

(2.2)

μ_1 = mean Nitrogen flux at depth 200 μ_2 = mean Nitrogen flux at depth 300

H_0 : Nitrogen flux does not change between 200 and 300 meter. $\mu_1 = \mu_2$ H_1 : Nitrogen flux is lower at 300 meter then at 200 meter. $\mu_1 > \mu_2$

```
t.test(log(N.depth.200.300$N_avg.200) ,log(N.depth.200.300$N_avg.300), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(N.depth.200.300$N_avg.200) and log(N.depth.200.300$N_avg.300)
## t = 15.938, df = 297, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.4047791       Inf
## sample estimates:
## mean of the differences
##               0.4515234
```

(3.1)

μ_1 = mean Phosporous flux at depth 150 μ_2 = mean Phosporous flux at depth 200

H_0 : Phosporous flux does not change between 150 and 200 meter. $\mu_1 = \mu_2$ H_1 : Phosporous flux is lower at 200 meter then at 150 meter. $\mu_1 > \mu_2$

```
t.test(log(P.depth.150.200$P_avg.150) ,log(P.depth.150.200$P_avg.200), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(P.depth.150.200$P_avg.150) and log(P.depth.150.200$P_avg.200)
## t = 10.941, df = 105, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3862101       Inf
## sample estimates:
## mean of the differences
##               0.4552613
```

(3.2)

μ_1 = mean Phosporous flux at depth 200 μ_2 = mean Phosporous flux at depth 300

H_0 : Phosporous flux does not change between 200 and 300 meter. $\mu_1 = \mu_2$ H_1 : Phosporous flux is lower at 300 meter then at 200 meter. $\mu_1 > \mu_2$

```
t.test(log(P.depth.200.300$P_avg.200) ,log(P.depth.200.300$P_avg.300), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(P.depth.200.300$P_avg.200) and log(P.depth.200.300$P_avg.300)
## t = 12.446, df = 104, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.4504533       Inf
## sample estimates:
## mean of the differences
##               0.5197648
```

(4.1)

μ_1 = mean Mass flux at depth 150 μ_2 = mean Mass flux at depth 200

H_0 : Mass flux does not change between 150 and 200 meter. $\mu_1 = \mu_2$ H_1 : Mass flux is lower at 200 meter then at 150 meter. $\mu_1 > \mu_2$

```
t.test(log(M.depth.150.200$M_avg.150) ,log(M.depth.150.200$M_avg.200), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(M.depth.150.200$M_avg.150) and log(M.depth.150.200$M_avg.200)
## t = 11.049, df = 303, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.1613635       Inf
## sample estimates:
## mean of the differences
##               0.1896883
```

(4.2)

μ_1 = mean Mass flux at depth 200 μ_2 = mean Mass flux at depth 300

H_0 : Mass flux does not change between 200 and 300 meter. $\mu_1 = \mu_2$ H_1 : Mass flux is lower at 300 meter then at 200 meter. $\mu_1 > \mu_2$

```
t.test(log(M.depth.200.300$M_avg.200) ,log(M.depth.200.300$M_avg.300), alternative =
"greater", paired = TRUE, var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  log(M.depth.200.300$M_avg.200) and log(M.depth.200.300$M_avg.300)
## t = 12.054, df = 300, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.2143744       Inf
## sample estimates:
## mean of the differences
##               0.2483726
```

Simple Linear Regression via LS

How does each flux predict Mass Flux?

```
# plotting each nutrient flux against mass flux

#plot(Data.noNA) # we see some relationships between mass flux and the rest of the nu
trient fluxes
par(mfrow=c(1,3))
plot(log(Data.noNA$C_avg),log(Data.noNA$M_avg)) #good linearity, but clumping
lmfit_C.M <- lm(log(M_avg) ~ log(C_avg), data = Data.noNA)
summary(lmfit_C.M)
```

```
##
## Call:
## lm(formula = log(M_avg) ~ log(C_avg), data = Data.noNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.83440 -0.25212 -0.04393  0.18902  2.05452
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.00513    0.08106   24.74   <2e-16 ***
## log(C_avg)   0.83365    0.02835   29.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3784 on 303 degrees of freedom
## Multiple R-squared:  0.7405, Adjusted R-squared:  0.7396
## F-statistic: 864.6 on 1 and 303 DF,  p-value: < 2.2e-16
```

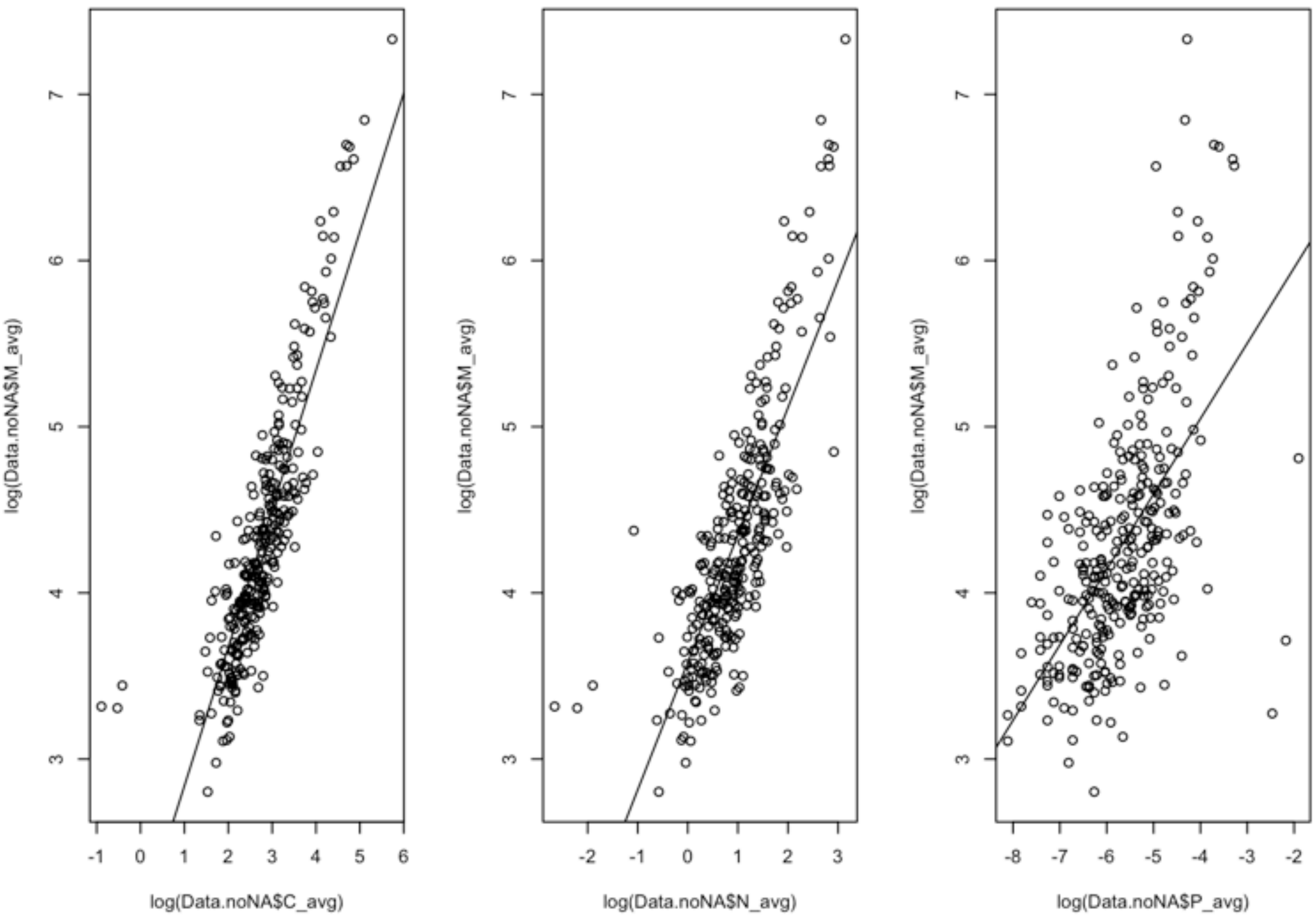
```
abline(lmfit_C.M)

plot(log(Data.noNA$N_avg),log(Data.noNA$M_avg)) # alright linearity, still clumping
lmfit_N.M <- lm(log(M_avg) ~ log(N_avg), data = Data.noNA)
summary(lmfit_N.M)
```

```
##
## Call:
## lm(formula = log(M_avg) ~ log(N_avg), data = Data.noNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96805 -0.31451 -0.06096  0.25532  1.77284
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.58148    0.03984   89.89   <2e-16 ***
## log(N_avg)   0.76636    0.03269   23.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4429 on 303 degrees of freedom
## Multiple R-squared:  0.6446, Adjusted R-squared:  0.6434
## F-statistic: 549.6 on 1 and 303 DF,  p-value: < 2.2e-16
```

```
abline(lmfit_N.M)

plot(log(Data.noNA$P_avg),log(Data.noNA$M_avg)) # worse, still clumping
lmfit_P.M <- lm(log(M_avg) ~ log(P_avg), data = Data.noNA)
summary(lmfit_P.M)
```

```
##
## Call:
## lm(formula = log(M_avg) ~ log(P_avg), data = Data.noNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.47045 -0.37370 -0.05109  0.33124  2.41221
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.86337    0.19945   34.41   <2e-16 ***
## log(P_avg)   0.45385    0.03482   13.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5946 on 303 degrees of freedom
## Multiple R-squared:  0.3592, Adjusted R-squared:  0.3571
## F-statistic: 169.9 on 1 and 303 DF,  p-value: < 2.2e-16
```

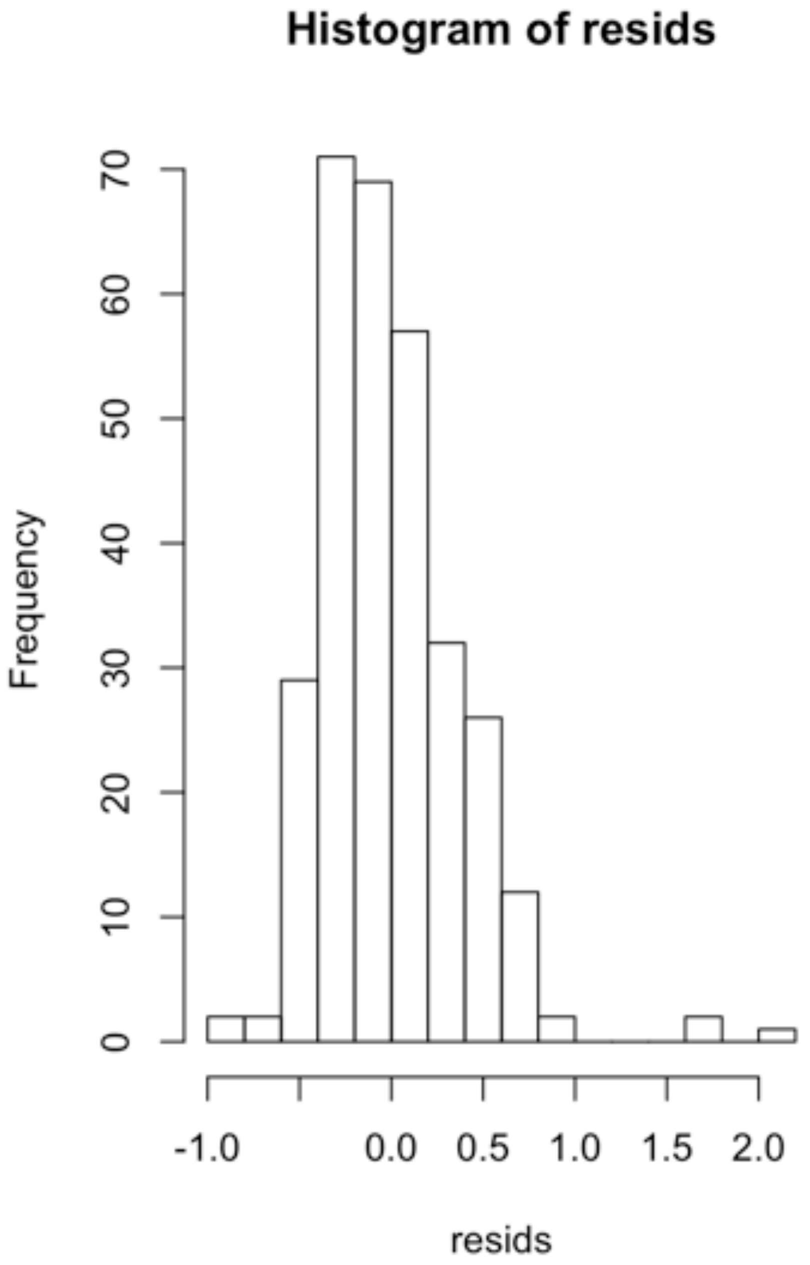
```
abline(lmfit_P.M)
```

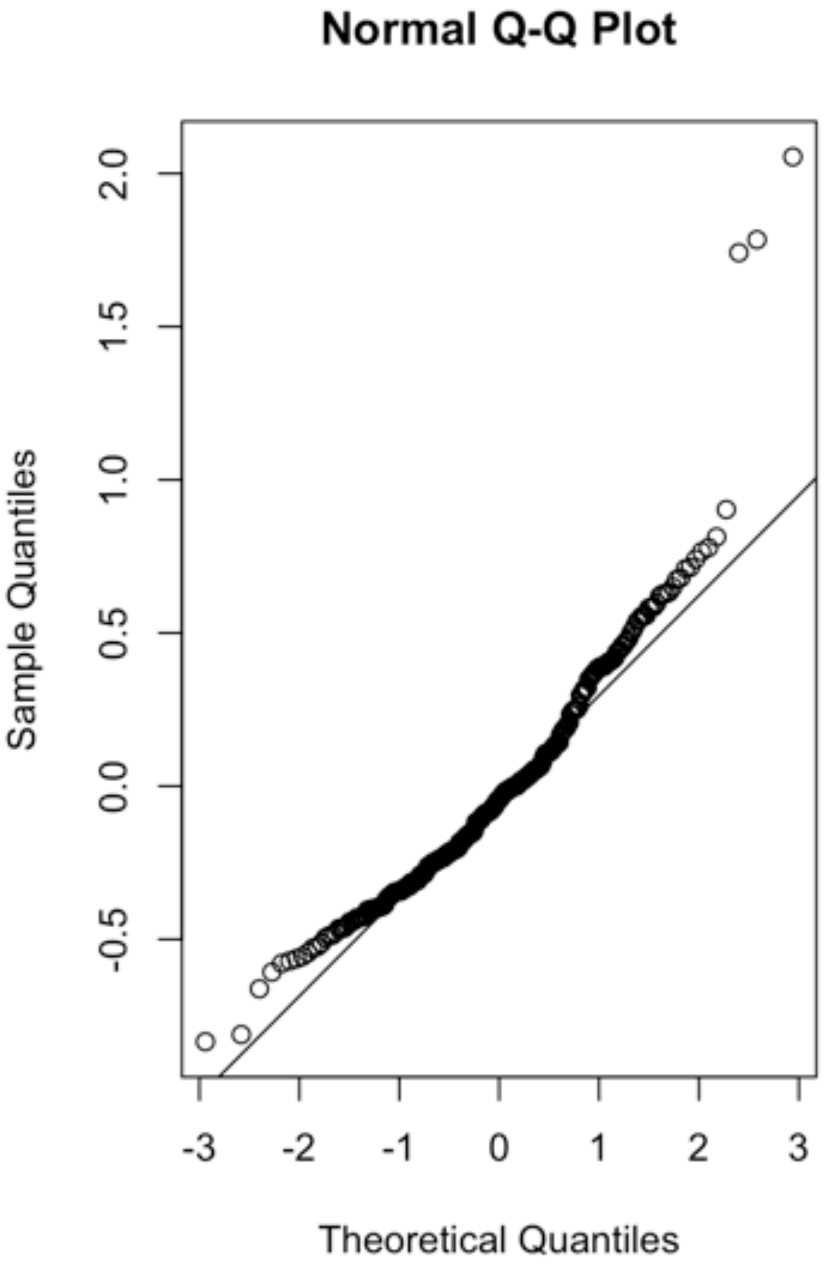


comment on these outputs

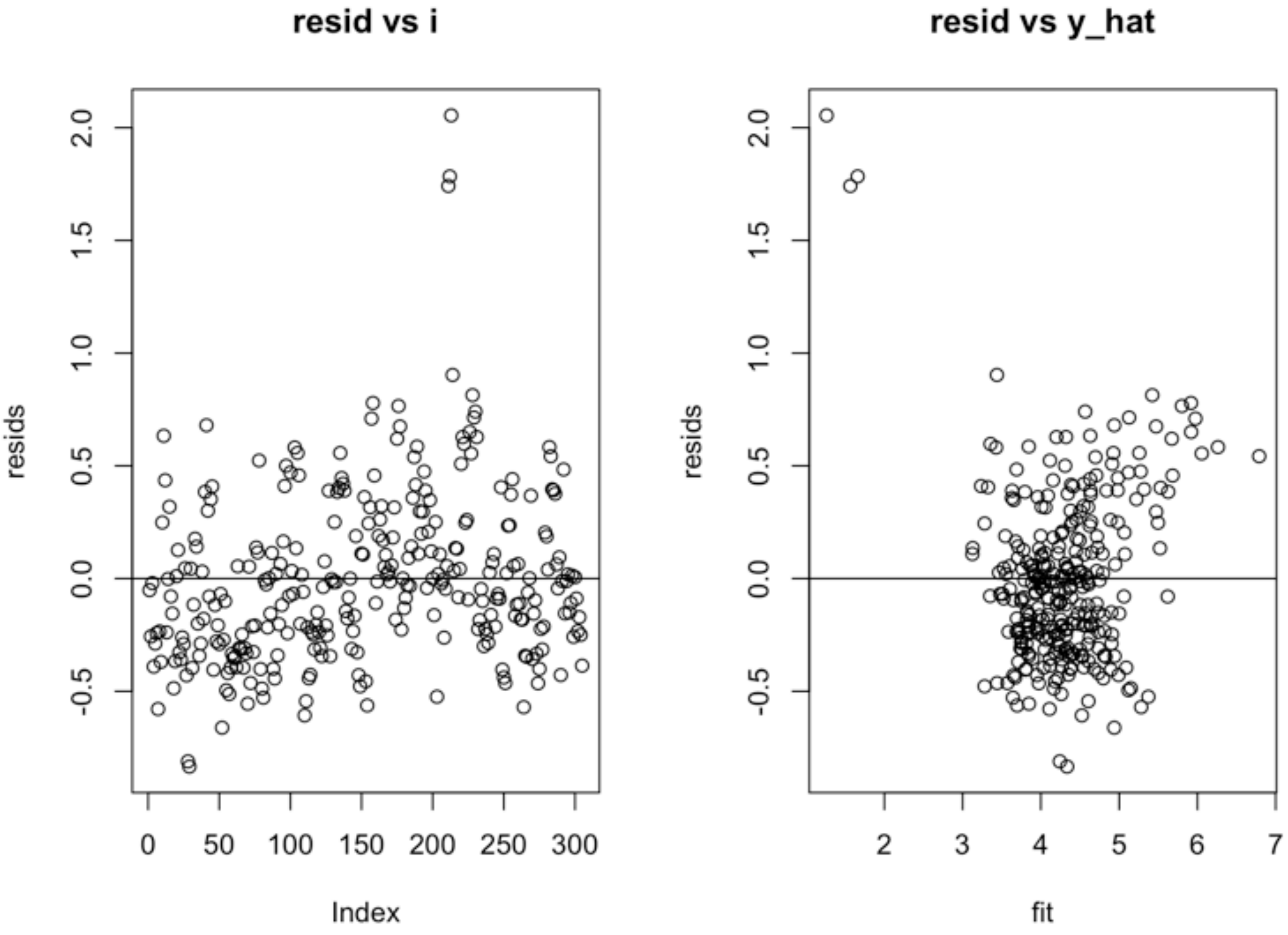
Carbon flux seems to be the best predictor of Mass flux. Lets run some EDA.

```
resids <- resid( lmfit_C.M ) # extract epsilon_hats
fit <- fitted ( lmfit_C.M ) # extract y_hats

par(mfrow=c(1,2))

hist(resids, breaks=20)
qqnorm(resids)
qqline(resids) # add straight line from true normal
```





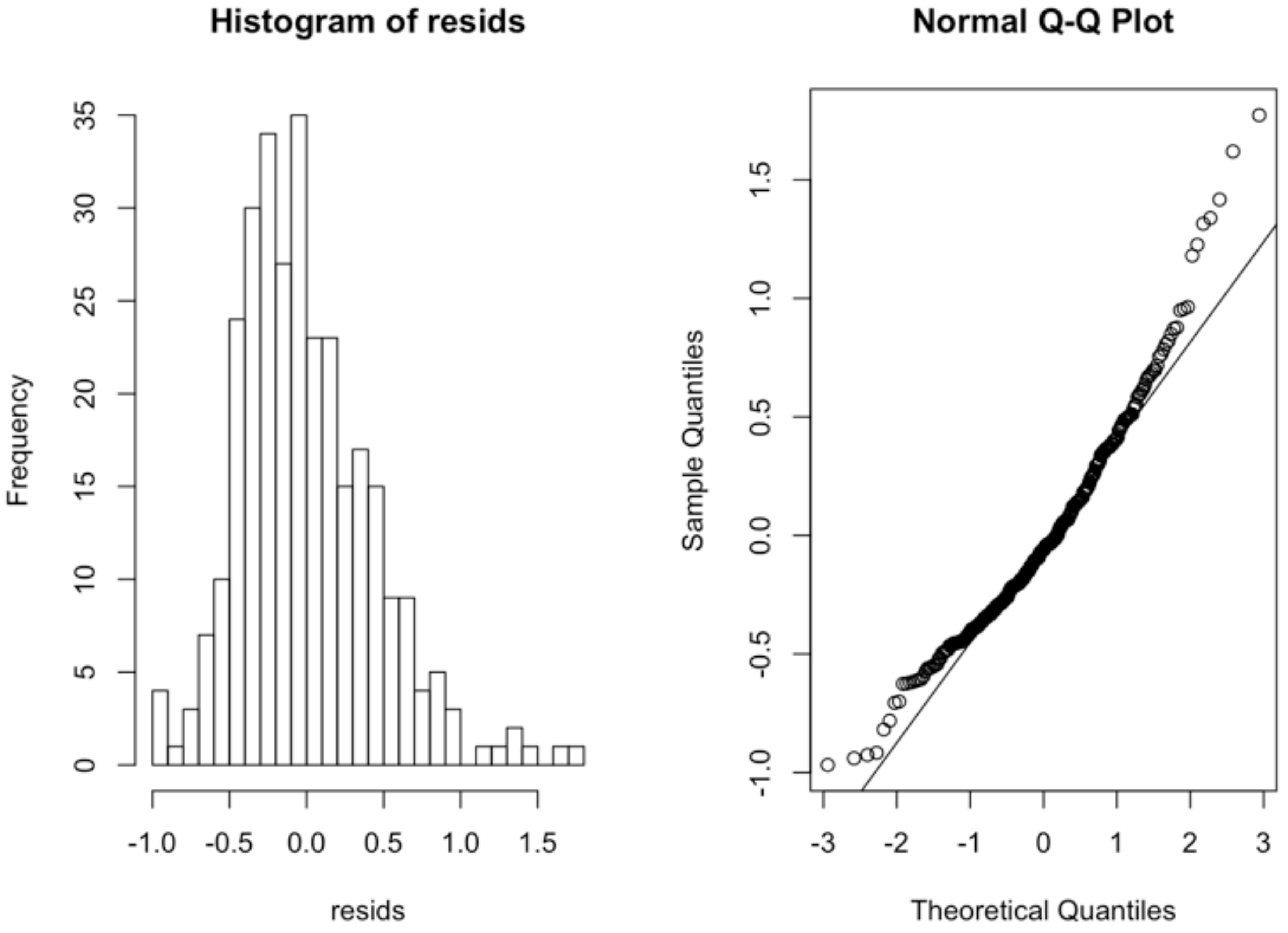
```
plot(resids, main="resid vs i")
abline(h=0) # mean of epsilon_hat

plot(x=fit, y=resids, main="resid vs y_hat")
abline(h=0)
```



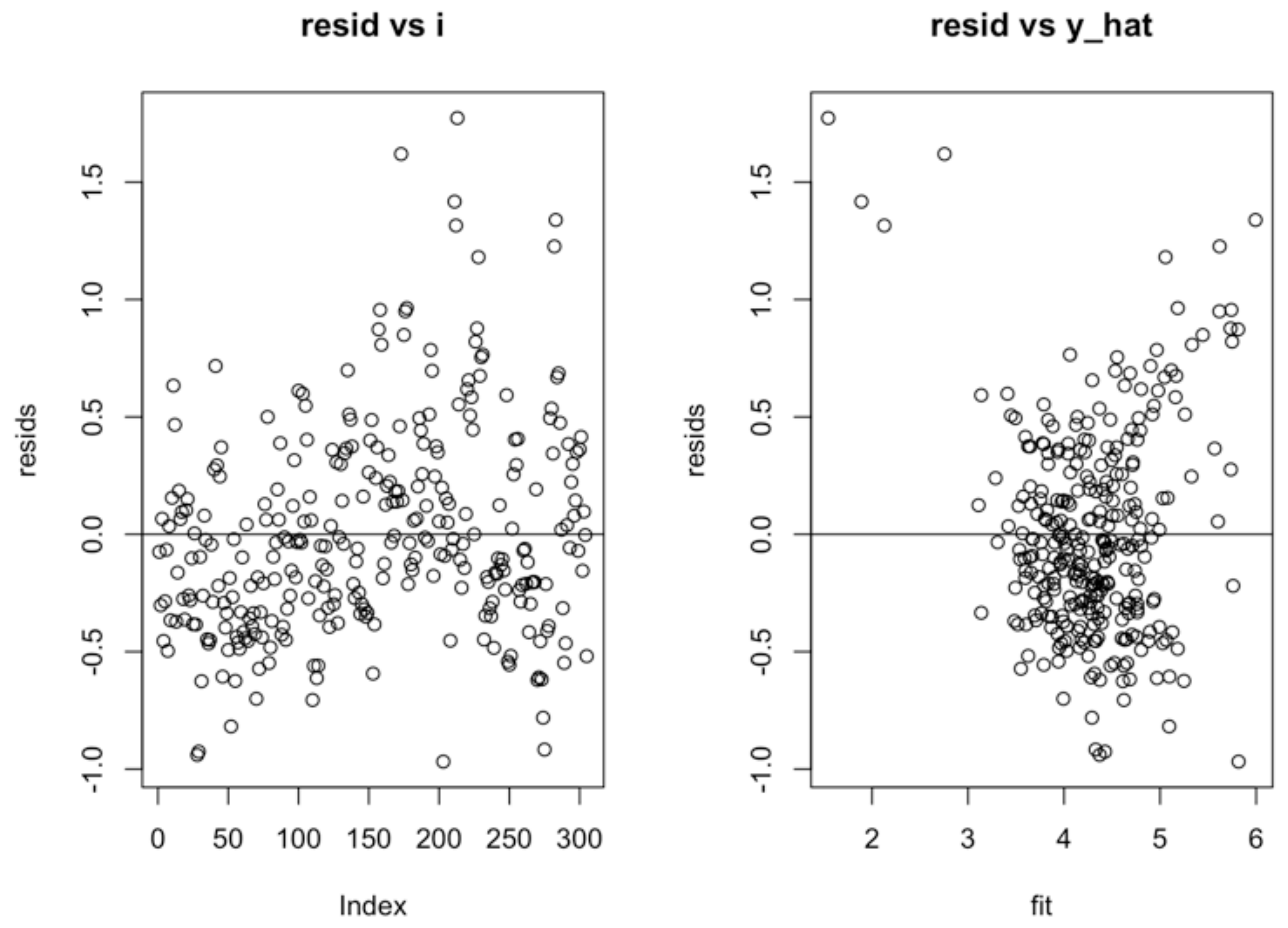
Comment on these

Lets compare these EDA’s to that of Nitrogen Flux and Mass Flux. Nitrogen Flux was also a strong predictor of Mass Flux.

```
resids <- resid( lmfit_N.M ) # extract epsilon_hats
fit <- fitted ( lmfit_N.M ) # extract y_hats

par(mfrow=c(1,2))

hist(resids, breaks=20)
qqnorm(resids)
qqline(resids) # add straight line from true normal
```



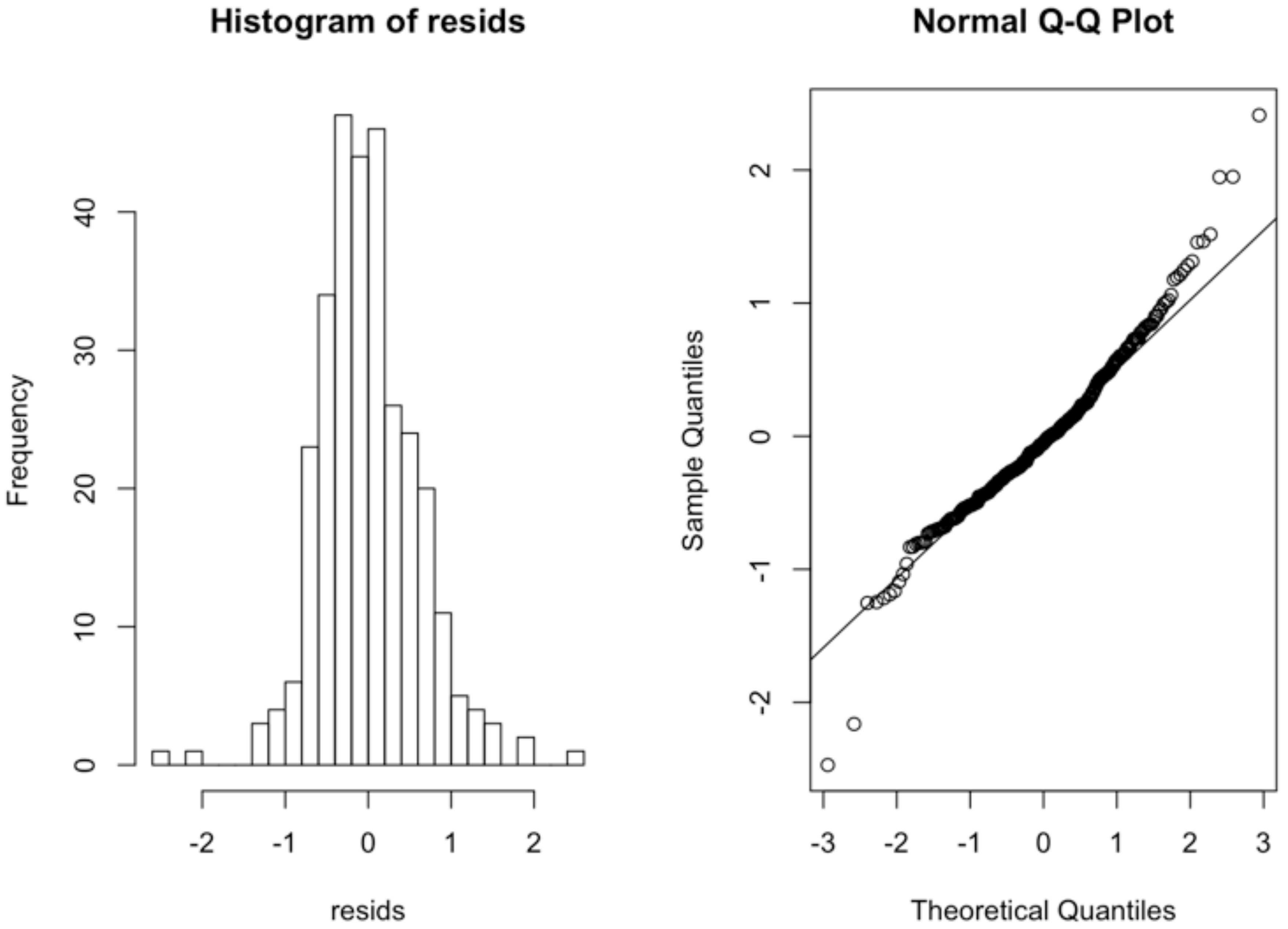
```
plot(resids, main="resid vs i")
abline(h=0) # mean of epsilon_hat

plot(x=fit, y=resids, main="resid vs y_hat")
abline(h=0)
```



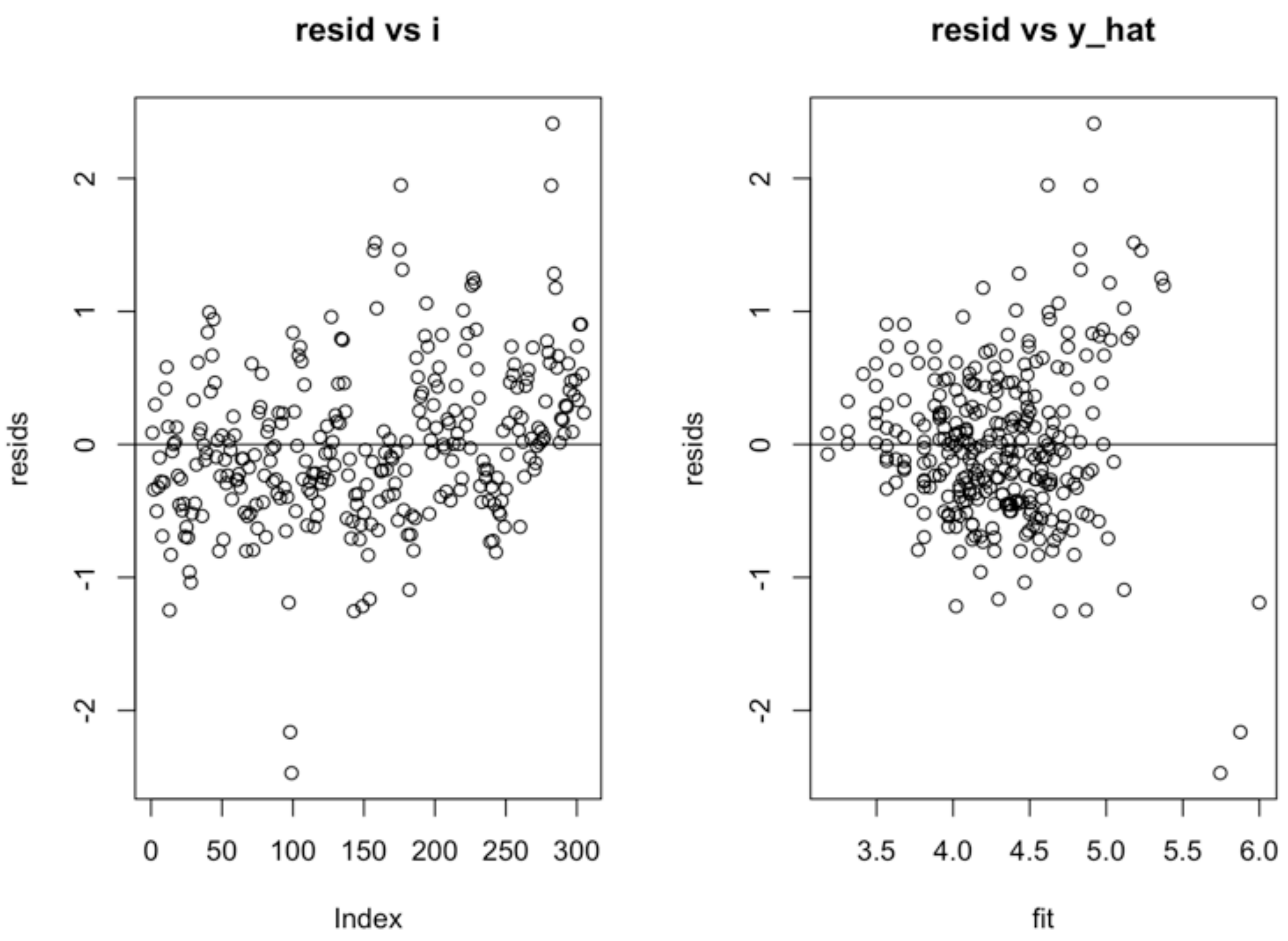
comment of these

Lets now compare these to Phosphorous Flux, a weak predcitor of Mass Flux.

```
resids <- resid( lmfit_P.M ) # extract epsilon_hats
fit <- fitted ( lmfit_P.M ) # extract y_hats

par(mfrow=c(1,2))

hist(resids, breaks=20)
qqnorm(resids)
qqline(resids) # add straight line from true normal
```



```
plot(resids, main="resid vs i")
abline(h=0) # mean of epsilon_hat

plot(x=fit, y=resids, main="resid vs y_hat")
abline(h=0)
```



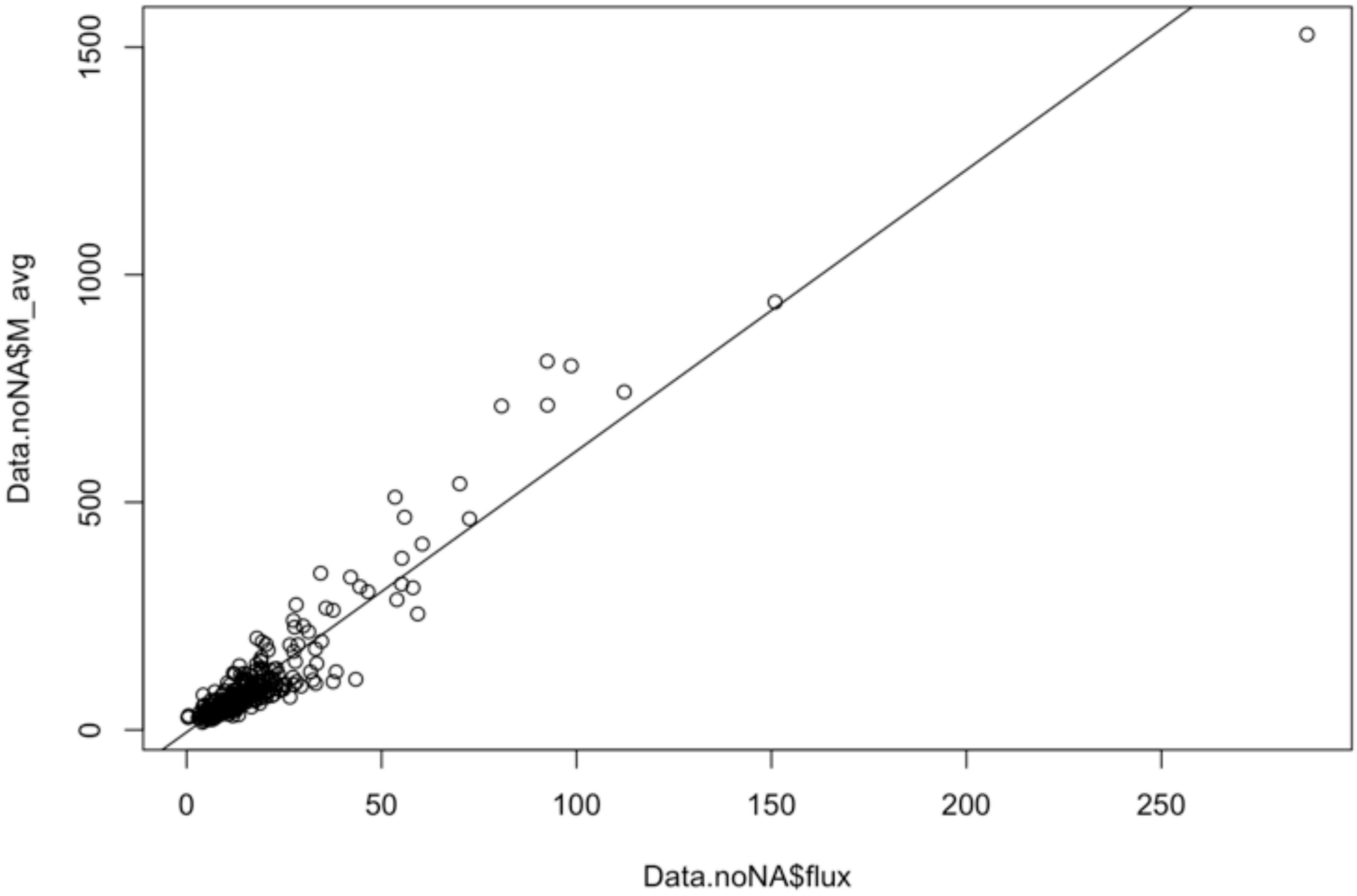
comment on these

Lets combine the fluxes

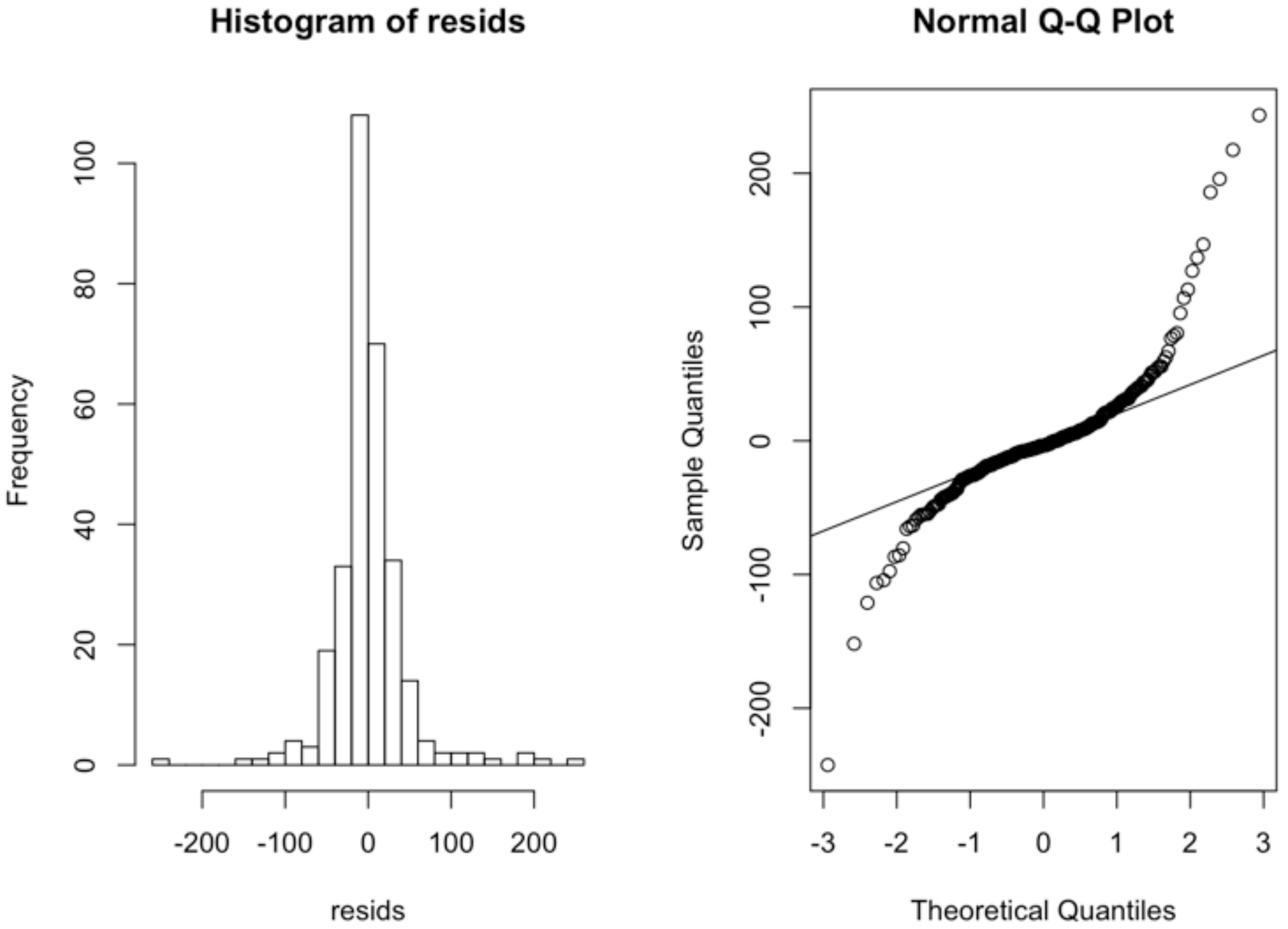
```
Data.noNA$flux <- Data.noNA$C_avg - Data.noNA$N_avg - Data.noNA$P_avg #combine the fluxes

lmfit.flux <- lm(M_avg ~ flux,
          data=Data.noNA)
summary(lmfit.flux)
```

```
## 
## Call:
## lm(formula = M_avg ~ flux, data = Data.noNA)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -242.439  -16.541   -3.581   12.990  243.455 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -5.1176     3.2751  -1.563    0.119    
## flux          6.1788     0.1114  55.453   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 44.92 on 303 degrees of freedom
## Multiple R-squared:  0.9103, Adjusted R-squared:   0.91 
## F-statistic:  3075 on 1 and 303 DF,  p-value: < 2.2e-16
```

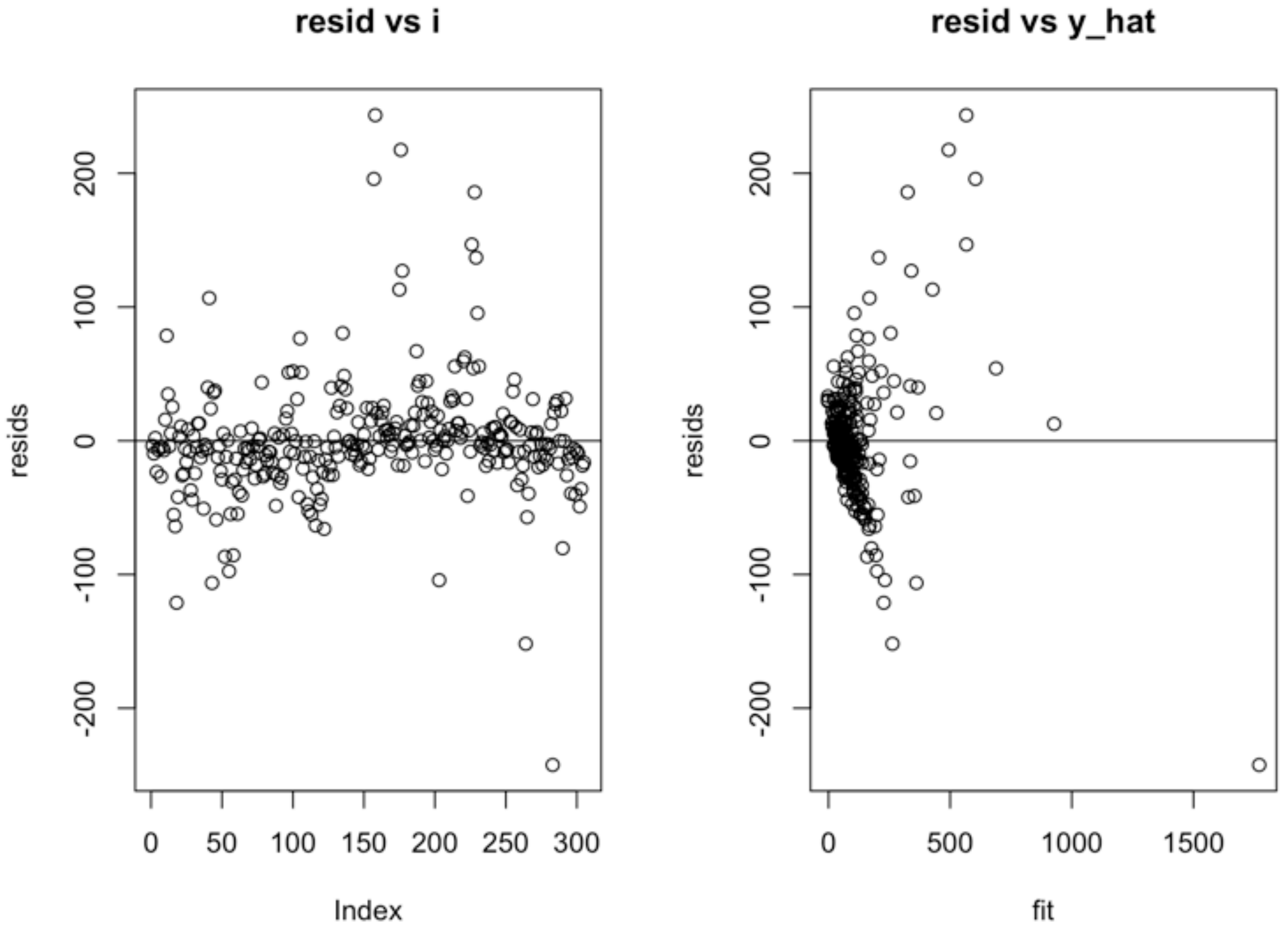
```
plot(Data.noNA$flux, Data.noNA$M_avg)
abline(lmfit.flux)
```



```
resids <- resid( lmfit.flux ) # extract epsilon_hats
fit <- fitted ( lmfit.flux ) # extract y_hats

par(mfrow=c(1,2))

hist(resids, breaks=20)
qqnorm(resids)
qqline(resids) # add straight line from true normal
```



```
plot(resids, main="resid vs i")
abline(h=0) # mean of epsilon_hat

plot(x=fit, y=resids, main="resid vs y_hat")
abline(h=0)
```



comment on this

How does depth affect flux?

```
# dept v. average for C_avg

par(mfrow=c(1,4))

plot(Data.noNA$dep, log(Data.noNA$C_avg))
lmfit_Dep.C <- lm(log(C_avg) ~ dep, data = Data.noNA)
summary(lmfit_Dep.C)
```

```
##
## Call:
## lm(formula = log(C_avg) ~ dep, data = Data.noNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6065 -0.4061 -0.0544  0.3198  2.9091
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.8268299  0.1447146  26.444  < 2e-16 ***
## dep         -0.0049867  0.0006469  -7.708 1.84e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7011 on 303 degrees of freedom
## Multiple R-squared:  0.164,  Adjusted R-squared:  0.1612
## F-statistic: 59.42 on 1 and 303 DF,  p-value: 1.839e-13
```

```
abline(lmfit_Dep.C)

plot(Data.noNA$dep, log(Data.noNA$N_avg))
lmfit_Dep.N <- lm(log(N_avg) ~ dep, data = Data.noNA)
summary(lmfit_Dep.N)
```

```
##
## Call:
## lm(formula = log(N_avg) ~ dep, data = Data.noNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5182 -0.3957 -0.0525  0.3702  2.1214
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.1679437  0.1428700  15.174   <2e-16 ***
## dep         -0.0057132  0.0006387  -8.945   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6922 on 303 degrees of freedom
## Multiple R-squared:  0.2089, Adjusted R-squared:  0.2063
## F-statistic: 80.02 on 1 and 303 DF,  p-value: < 2.2e-16
```

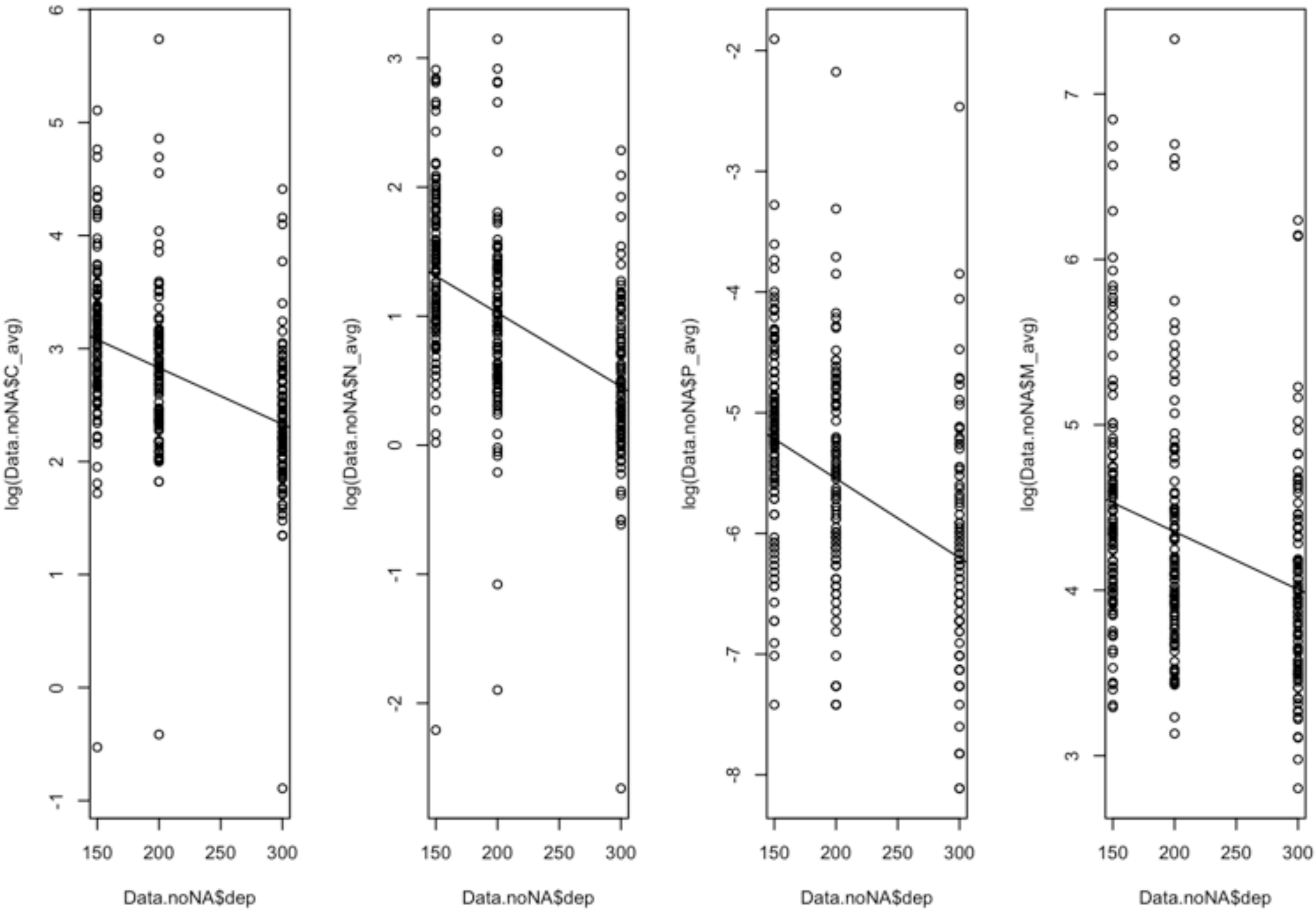
```
abline(lmfit_Dep.N)

plot(Data.noNA$dep, log(Data.noNA$P_avg))
lmfit_Dep.P <- lm(log(P_avg) ~ dep, data = Data.noNA)
summary(lmfit_Dep.P)
```

```
##
## Call:
## lm(formula = log(P_avg) ~ dep, data = Data.noNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2015 -0.5223 -0.0115  0.5191  3.7380
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.2310162  0.1840231 -22.992  < 2e-16 ***
## dep         -0.0065736  0.0008226  -7.991 2.83e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8916 on 303 degrees of freedom
## Multiple R-squared:  0.1741, Adjusted R-squared:  0.1713
## F-statistic: 63.85 on 1 and 303 DF,  p-value: 2.83e-14
```

```
abline(lmfit_Dep.P)

plot(Data.noNA$dep, log(Data.noNA$M_avg))
lmfit_Dep.M <- lm(log(M_avg) ~ dep, data = Data.noNA)
summary(lmfit_Dep.M)
```

```
##
## Call:
## lm(formula = log(M_avg) ~ dep, data = Data.noNA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2363 -0.4836 -0.1391  0.3049  2.9777
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.0517835  0.1466243  34.454  < 2e-16 ***
## dep         -0.0034891  0.0006555  -5.323 1.99e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7104 on 303 degrees of freedom
## Multiple R-squared:  0.08552,    Adjusted R-squared:  0.0825
## F-statistic: 28.34 on 1 and 303 DF,  p-value: 1.989e-07
```

```
abline(lmfit_Dep.M)
```

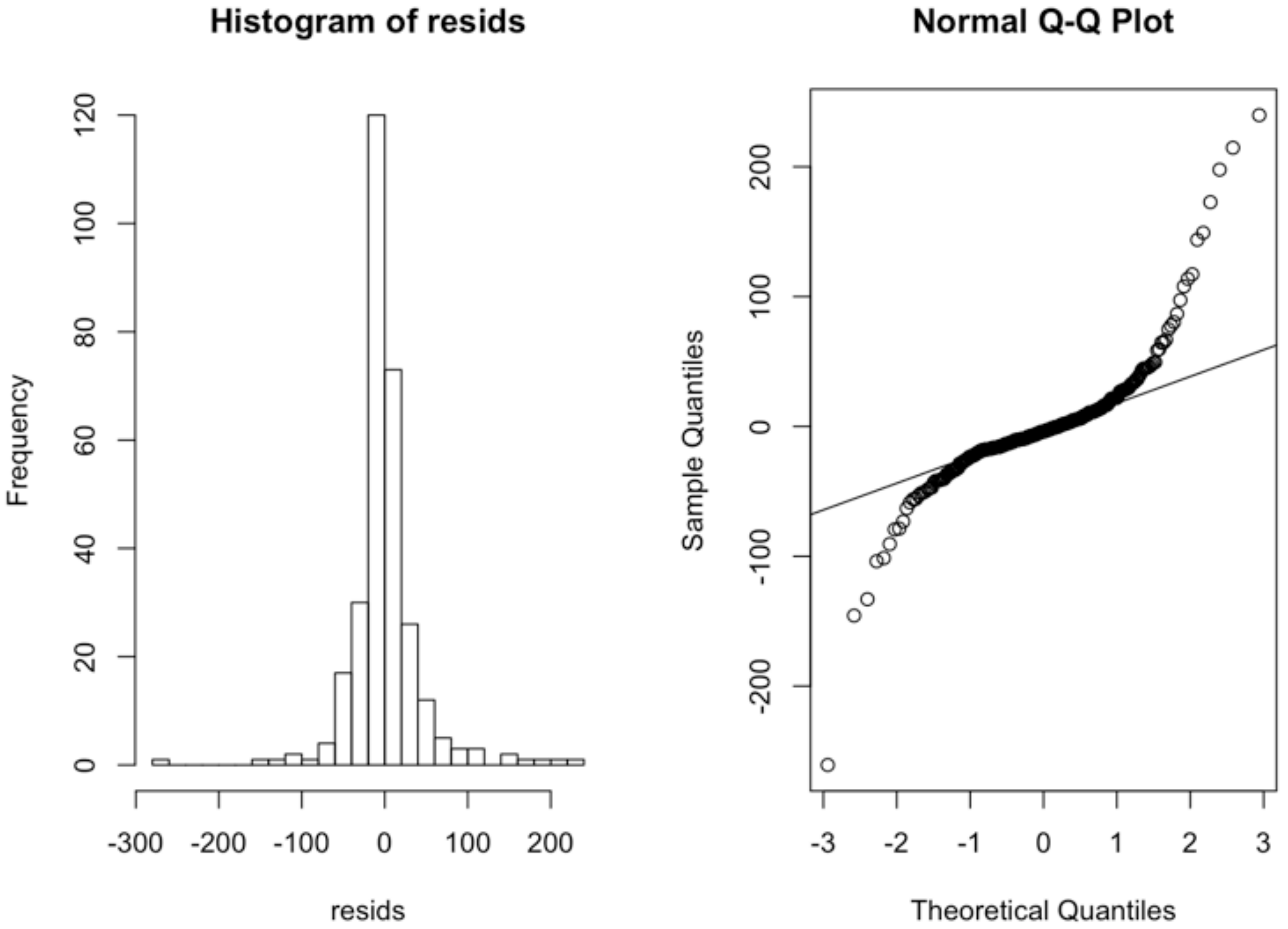


Multiple regression

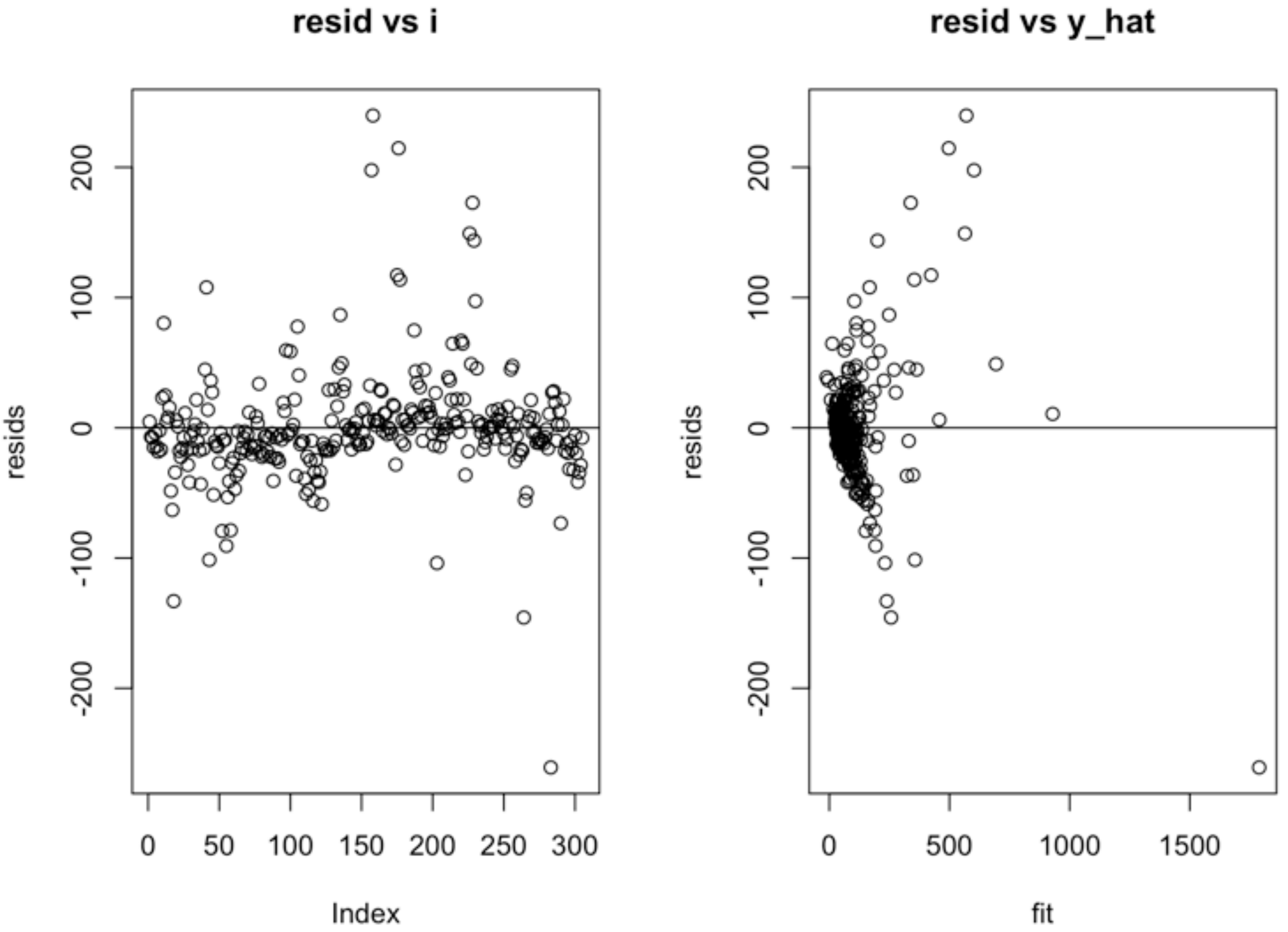
```
lmfit.MR <- lm(M_avg ~ flux + dep, data = Data.noNA)
summary(lmfit.MR)
```

```
##
## Call:
## lm(formula = M_avg ~ flux + dep, data = Data.noNA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -260.800  -16.536   -3.988   11.201  239.720
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -32.99883   10.03280  -3.289  0.00112 **
## flux          6.25389    0.11298  55.354  < 2e-16 ***
## dep           0.12338    0.04202   2.936  0.00358 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.37 on 302 degrees of freedom
## Multiple R-squared:  0.9128, Adjusted R-squared:  0.9122
## F-statistic:  1580 on 2 and 302 DF,  p-value: < 2.2e-16
```

```
resids <- resid( lmfit.MR ) # extract epsilon_hats
fit <- fitted ( lmfit.MR ) # extract y_hats

par(mfrow=c(1,2))

hist(resids, breaks=20)
qqnorm(resids)
qqline(resids) # add straight line from true normal
```



```
plot(resids, main="resid vs i")
abline(h=0) # mean of epsilon_hat

plot(x=fit, y=resids, main="resid vs y_hat")
abline(h=0)
```



which one is better?

```
smaller <- lmfit.flux; larger <- lmfit.MR
anova(smaller, larger)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	303	611381.3	NA	NA	NA	NA
2	302	594417.2	1	16964.09	8.618789	0.003582699

2 rows

Comment of this output

ANCOVA

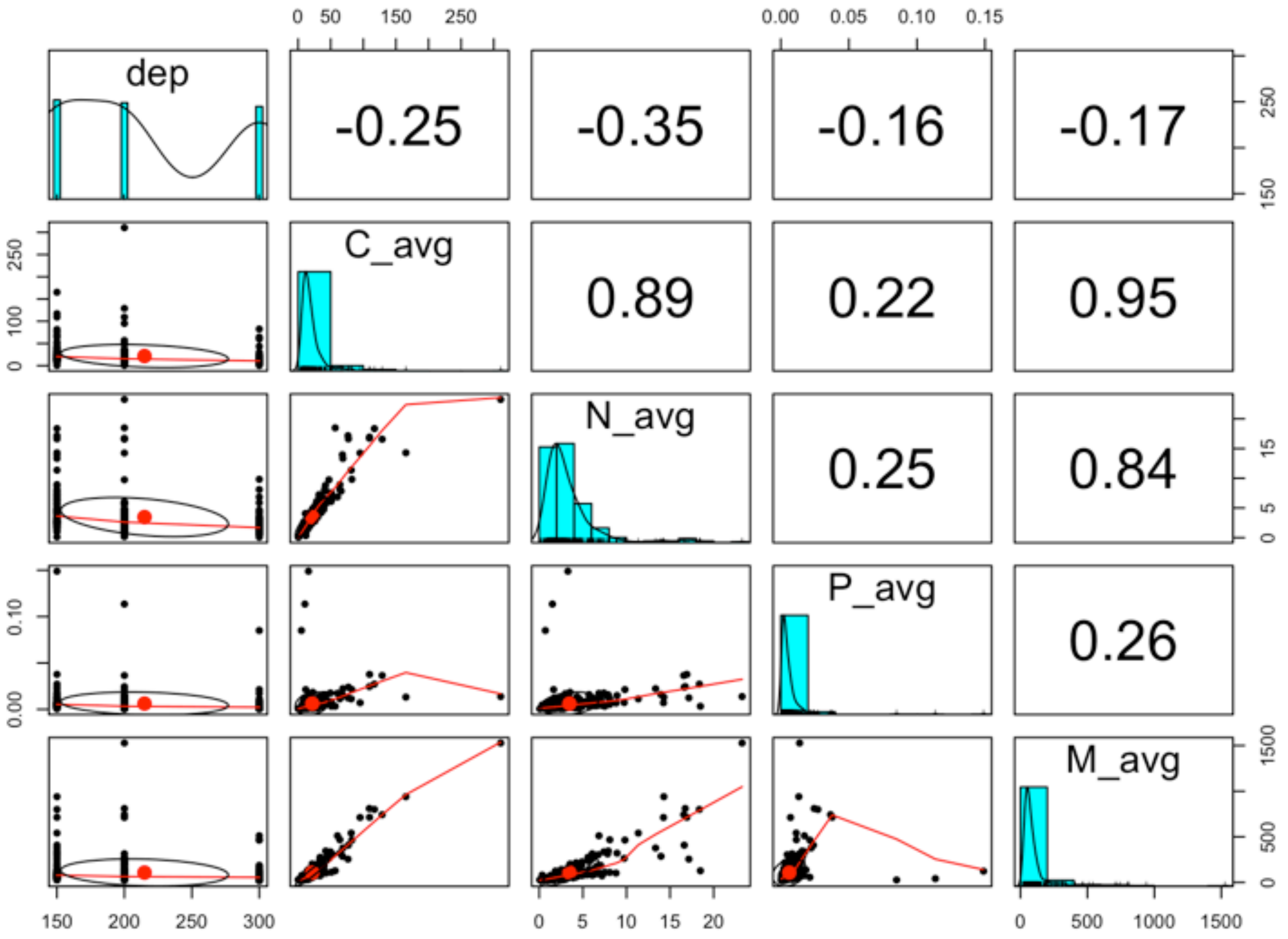
```r
#we want to make predictions on the Mass average flux.

# we want to compare this against depth and C,N,P fluxes

# Depth is categorical
# N,P,C are numerical

library(psych) # for `pairs.panels()`
library(lattice)

ANOCOVA.data <- subset(Data.noNA, select = c("dep", "M_avg", "C_avg","N_avg","P_avg")
)
as.character(ANOCOVA.data$dep) # need to make dep categorical not numerical
```

```
##   [1] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [13] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [25] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [37] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [49] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [61] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [73] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [85] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
##  [97] "150" "200" "300" "150" "150" "200" "300" "150" "200" "300" "150" "200"
## [109] "300" "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200"
## [121] "300" "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200"
## [133] "300" "150" "150" "200" "300" "150" "200" "300" "150" "200" "300" "150"
## [145] "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300" "150"
## [157] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
## [169] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
## [181] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
## [193] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
## [205] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
## [217] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
## [229] "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300"
## [241] "150" "200" "300" "150" "200" "150" "200" "300" "150" "200" "300" "150"
## [253] "200" "300" "150" "200" "300" "150" "200" "200" "150" "200" "300" "150"
## [265] "200" "300" "150" "200" "300" "150" "200" "300" "150" "200" "300" "150"
## [277] "200" "300" "150" "200" "300" "150" "200" "150" "200" "300" "150" "200"
## [289] "300" "150" "200" "300" "150" "200" "300" "150" "200" "300" "150" "200"
## [301] "300" "150" "200" "300" "150"
```

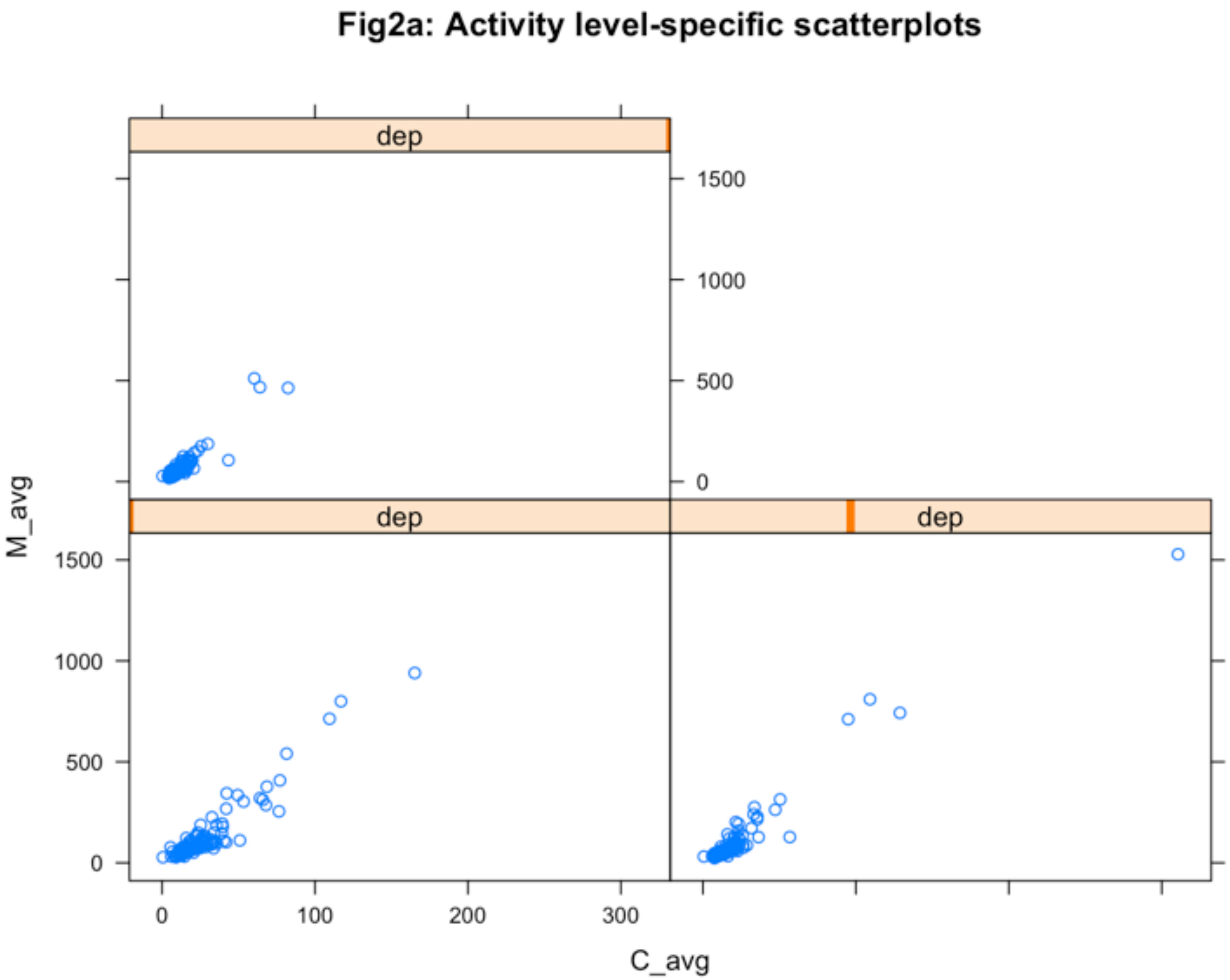
```r
attach(ANOCOVA.data)

summary(ANOCOVA.data)
```

```
##       dep             M_avg              C_avg            N_avg       
##  Min.   :150.0   Min.   :  16.49   Min.   :  0.41   Min.   : 0.070  
##  1st Qu.:150.0   1st Qu.:  45.57   1st Qu.: 10.07   1st Qu.: 1.600  
##  Median :200.0   Median :  64.90   Median : 15.09   Median : 2.510  
##  Mean   :214.9   Mean   : 107.31   Mean   : 21.67   Mean   : 3.464  
##  3rd Qu.:300.0   3rd Qu.: 101.87   3rd Qu.: 23.15   3rd Qu.: 4.100  
##  Max.   :300.0   Max.   :1527.97   Max.   :310.63   Max.   :23.260  
##      P_avg         
##  Min.   :0.000300  
##  1st Qu.:0.001900  
##  Median :0.003500  
##  Mean   :0.006229  
##  3rd Qu.:0.006600  
##  Max.   :0.148900  
```

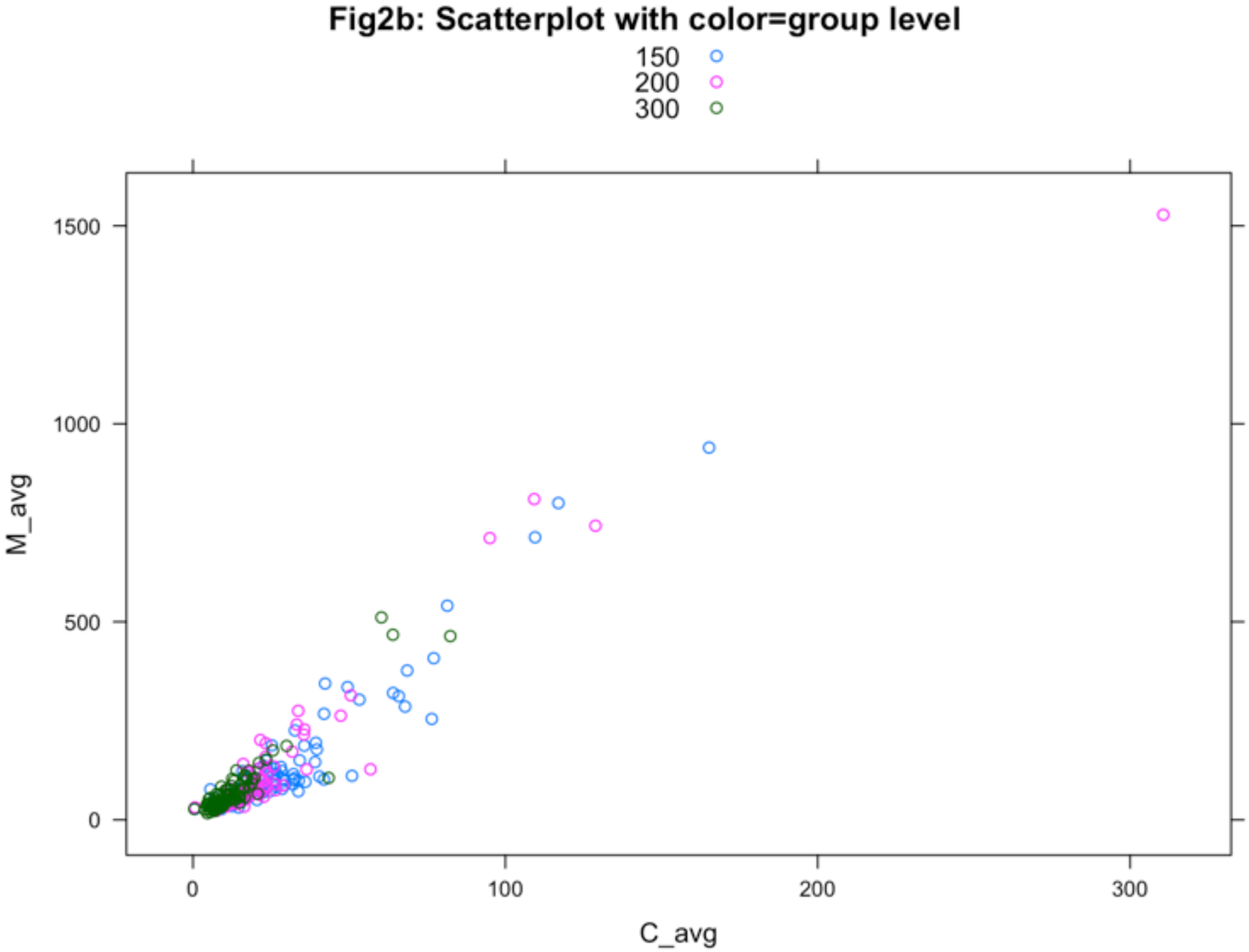
```
pairs.panels(data.frame(dep,C_avg,N_avg, P_avg, M_avg))
```



```
xyplot(M_avg ~ C_avg | dep,
                  main="Fig2a: Activity level-specific scatterplots"
           )
```

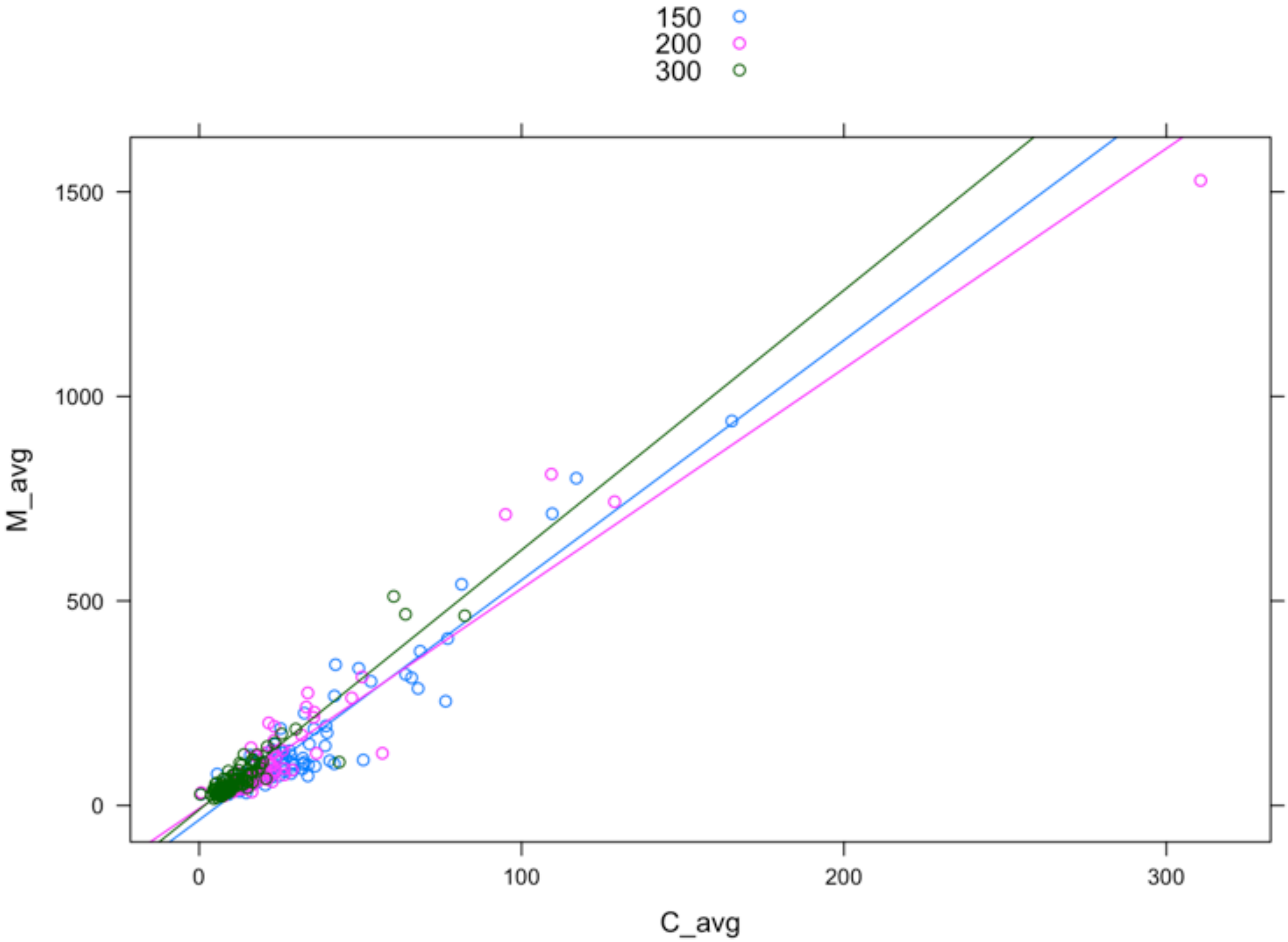


```
    xyplot(M_avg ~ C_avg, groups=dep,
                  auto.key=TRUE,
                  main="Fig2b: Scatterplot with color=group level"
           )
```



```
    xyplot(M_avg ~ C_avg, groups=dep,
                  type=c("p","r"), # `p` for _points_, `r` for _regression line_
                                   # see https://stackoverflow.com/questions/12972039
/plotting-xyplot-with-regression-line-on-lattice-graphics for more
                  auto.key=TRUE,
                  main="Fig3: Three standalone depth level-specific `lm()` fits"
           )
```

Fig3: Three standalone depth level-specific `lm()` fits

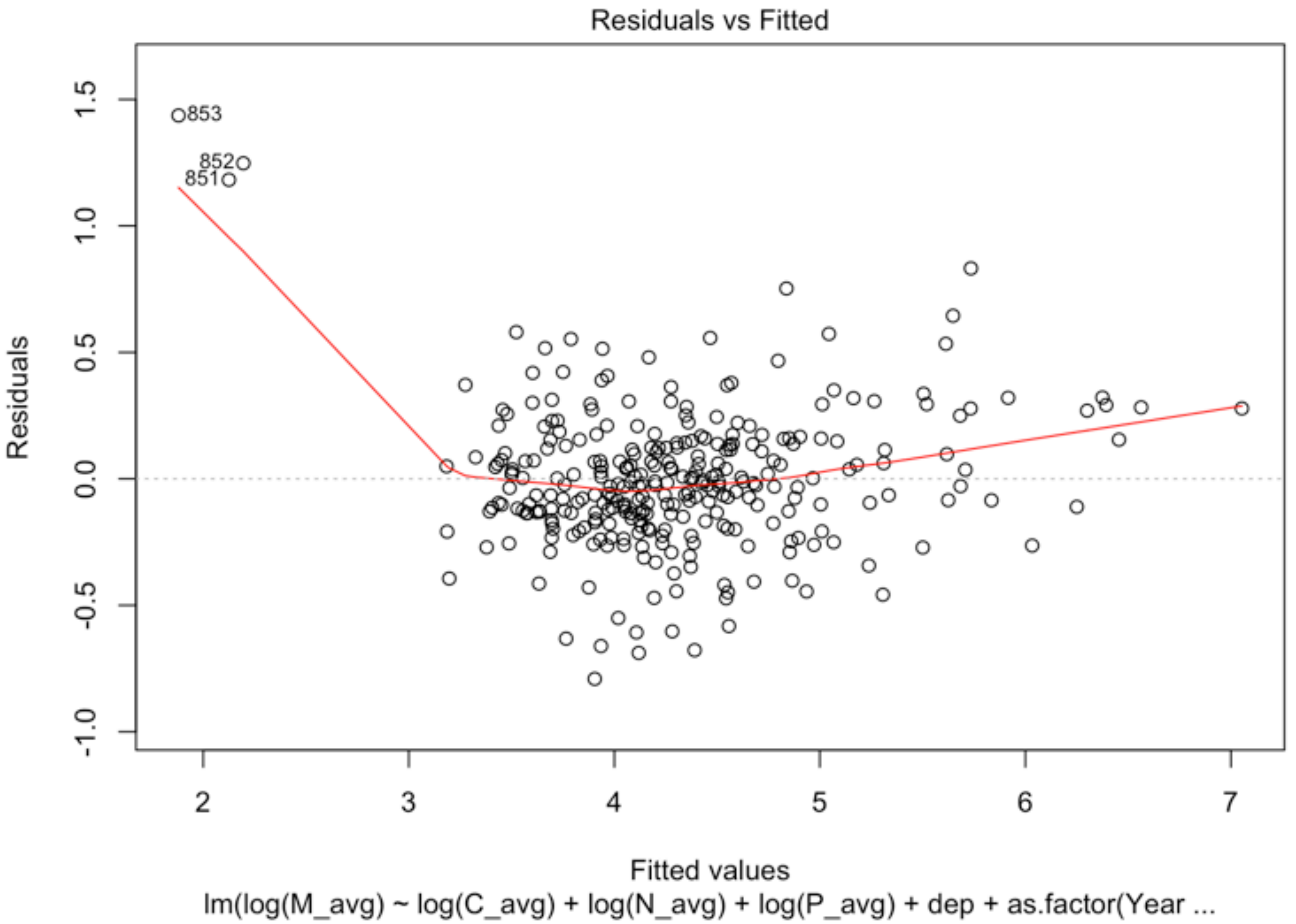


```
#reformat yymmdd so that r can better use it blocking factors

# only yymmdd1 was used to create new year and month columns since yymmdd1 and yymmdd
2 sampled were only a couple days apart and were averaged out in flux columns

Data.noNA$Date <- as.Date(paste(substr(Data.noNA$yymmdd1,1,4),
                                        substr(Data.noNA$yymmdd1,5,6),
                                        substr(Data.noNA$yymmdd1,7,8), sep = "-"),
                                  format = '%Y-%m-%d')

Data.noNA$Year <- substr(Data.noNA$yymmdd1,1,4)

#install.packages("lubridate")
library(lubridate)
Data.noNA$month <- as.Date(paste(substr(Data.noNA$yymmdd1,5,6),
                             substr(Data.noNA$yymmdd1,7,8), sep = "-"),
                       format = '%m-%d')
Data.noNA$month <- round_date(Data.noNA$month, unit = "month")
Data.noNA$month <- format(Data.noNA$month,format = "%Y-%b-%d")
Data.noNA$month <- substr(paste(Data.noNA$month),6,8)
```

```
ANOCOVA.data <- subset(Data.noNA, select = c("dep", "M_avg", "C_avg","N_avg","P_avg",
"Year", "month"))
ANOCOVA.data <- as.vector(ANOCOVA.data)
ANCOVA <- lm(log(M_avg) ~ log(C_avg) + log(N_avg) + log(P_avg) + dep + as.factor(Year
) + as.factor(month),data = ANOCOVA.data)
summary(ANCOVA)
```

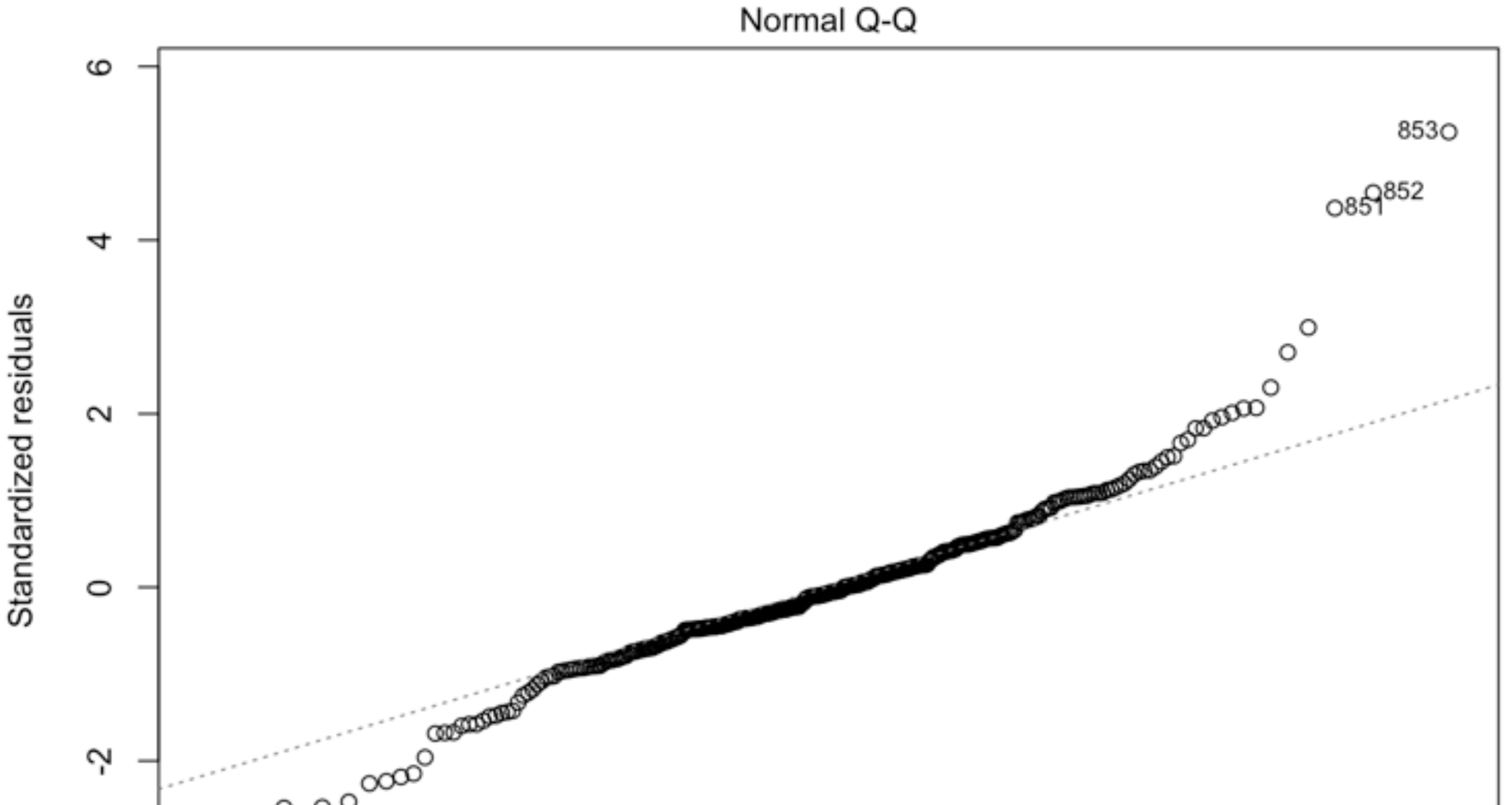
```
##
## Call:
## lm(formula = log(M_avg) ~ log(C_avg) + log(N_avg) + log(P_avg) +
##     dep + as.factor(Year) + as.factor(month), data = ANOCOVA.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79080 -0.13761 -0.01913  0.13800  1.43640
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.1133046  0.3096395  10.055  < 2e-16 ***
## log(C_avg)           0.7104828  0.0953704   7.450 1.18e-12 ***
## log(N_avg)          -0.0154473  0.0893634  -0.173 0.862888
## log(P_avg)           0.1342152  0.0280565   4.784 2.79e-06 ***
## dep                  0.0008642  0.0003428   2.521 0.012248 *
## as.factor(Year)2006  0.0308480  0.1219263   0.253 0.800451
## as.factor(Year)2007 -0.0660690  0.1188095  -0.556 0.578594
## as.factor(Year)2008  0.0231154  0.1239489   0.186 0.852195
## as.factor(Year)2009  0.0406626  0.1181832   0.344 0.731057
## as.factor(Year)2010  0.0780500  0.1244893   0.627 0.531197
## as.factor(Year)2011  0.2954345  0.1219114   2.423 0.016014 *
## as.factor(Year)2012  0.4328774  0.1200874   3.605 0.000370 ***
## as.factor(Year)2013  0.1907860  0.1180029   1.617 0.107053
## as.factor(Year)2014  0.1115483  0.1224068   0.911 0.362928
## as.factor(Year)2015  0.2994775  0.1207850   2.479 0.013750 *
## as.factor(month)Aug -0.5090202  0.0904291  -5.629 4.41e-08 ***
## as.factor(month)Dec -0.4301884  0.0830806  -5.178 4.30e-07 ***
## as.factor(month)Feb -0.0916026  0.0795033  -1.152 0.250231
## as.factor(month)Jan -0.4878071  0.1898550  -2.569 0.010708 *
## as.factor(month)Jul -0.4706758  0.0759314  -6.199 2.04e-09 ***
## as.factor(month)Jun -0.4504058  0.0921921  -4.886 1.74e-06 ***
## as.factor(month)Mar  0.0141537  0.0983848   0.144 0.885715
## as.factor(month)May -0.4274599  0.0834863  -5.120 5.70e-07 ***
## as.factor(month)Nov -0.2829892  0.0891804  -3.173 0.001676 **
## as.factor(month)Oct -0.3367639  0.0923424  -3.647 0.000317 ***
## as.factor(month)Sep -0.3769393  0.0804426  -4.686 4.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2917 on 279 degrees of freedom
## Multiple R-squared:  0.858,  Adjusted R-squared:  0.8453
## F-statistic: 67.46 on 25 and 279 DF,  p-value: < 2.2e-16
```

```
plot(ANCOVA)
```

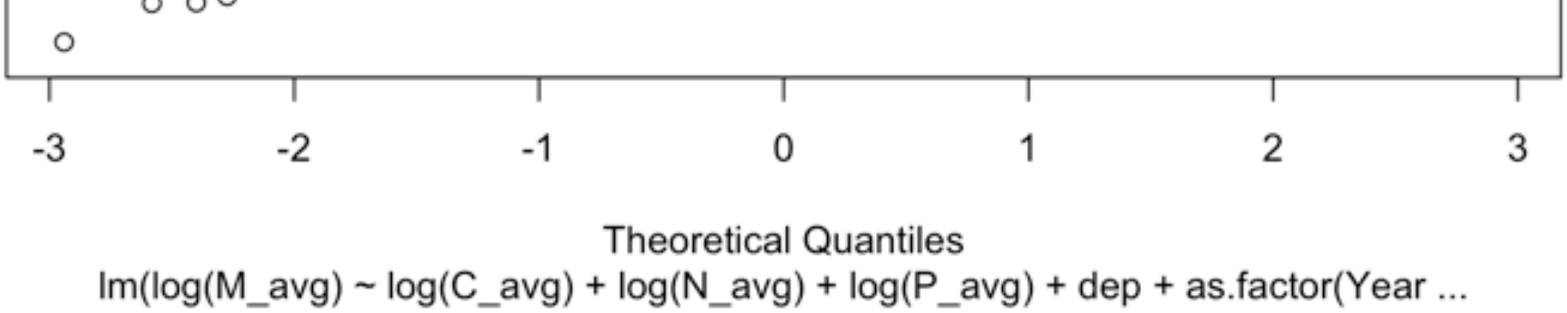


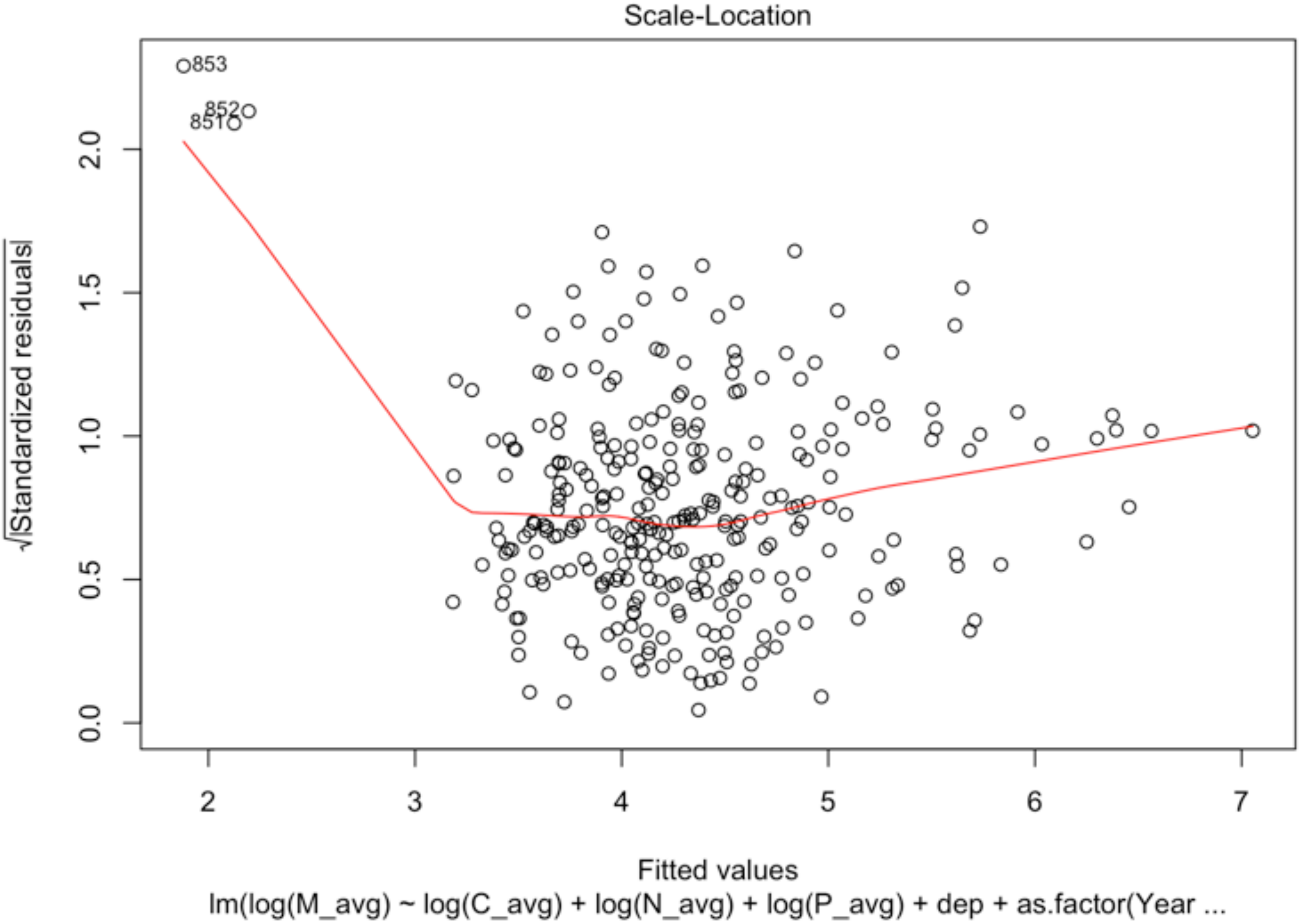
Residuals vs Fitted

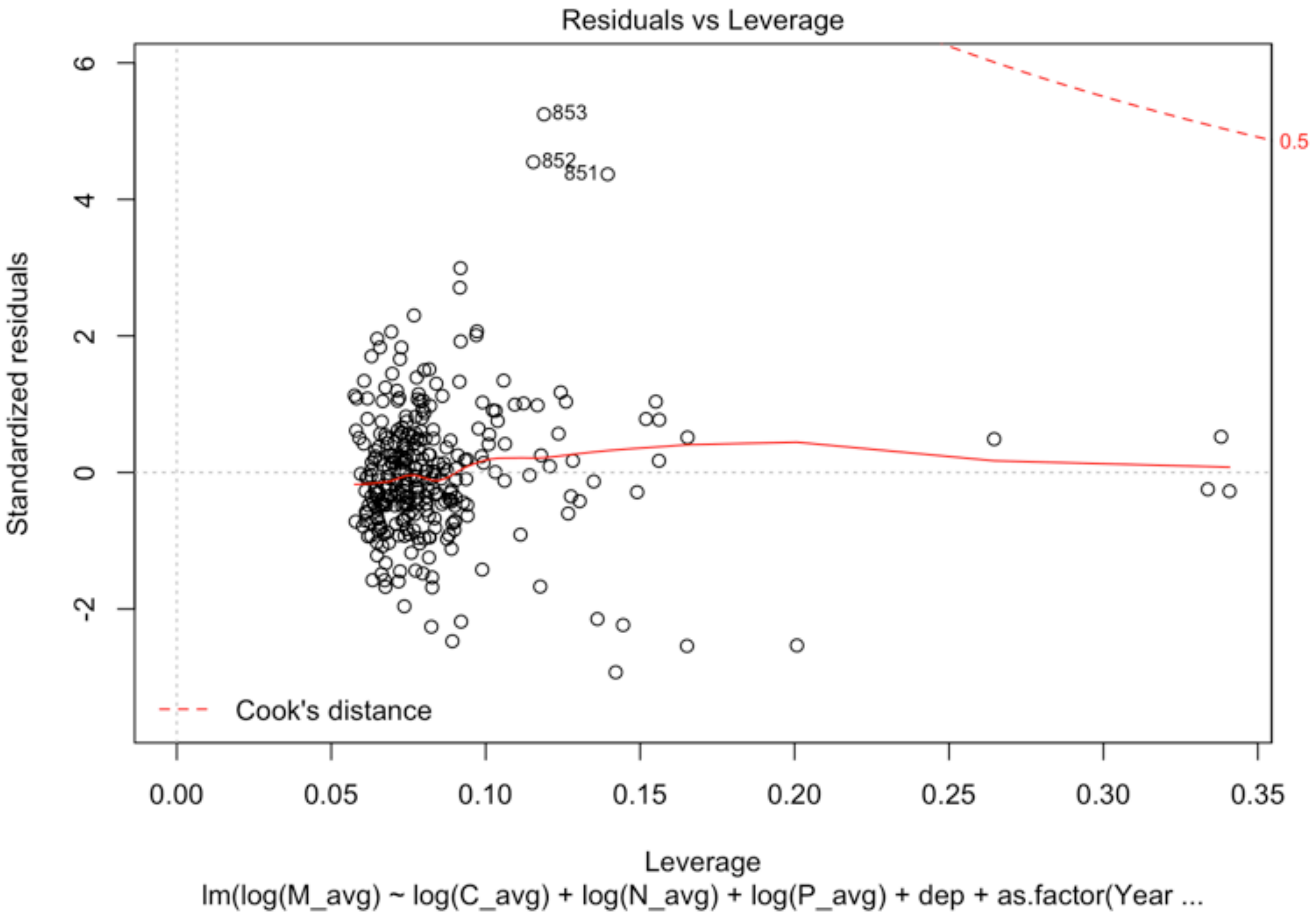
lm(log(M_avg) ~ log(C_avg) + log(N_avg) + log(P_avg) + dep + as.factor(Year ...



Normal Q-Q


-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(log(M_avg) ~ log(C_avg) + log(N_avg) + log(P_avg) + dep + as.factor(Year ...


Scale-Location
853
852
851
0.0
0.5
1.0
1.5
2.0
√|Standardized residuals|
2
3
4
5
6
7
Fitted values
lm(log(M_avg) ~ log(C_avg) + log(N_avg) + log(P_avg) + dep + as.factor(Year ...



```
resids <- resid(ANCOVA)
fit <-fitted(ANCOVA)
```

Results discussion:

Limitations:

References: