Octopus Pcrit Calculations

Kirt Onthank

8/10/2020

```
library(OTools)
library(xlsx)
library(segmented)
```

Making dataset to place Critical Oxygen Pressure (P_{CRIT}) data

```
pcrit.table=data.frame(octo=rep(0,14),Pc=rep(0,14),broken=rep(0,14))
```

Calculating P_{CRIT} data

Octopus 5

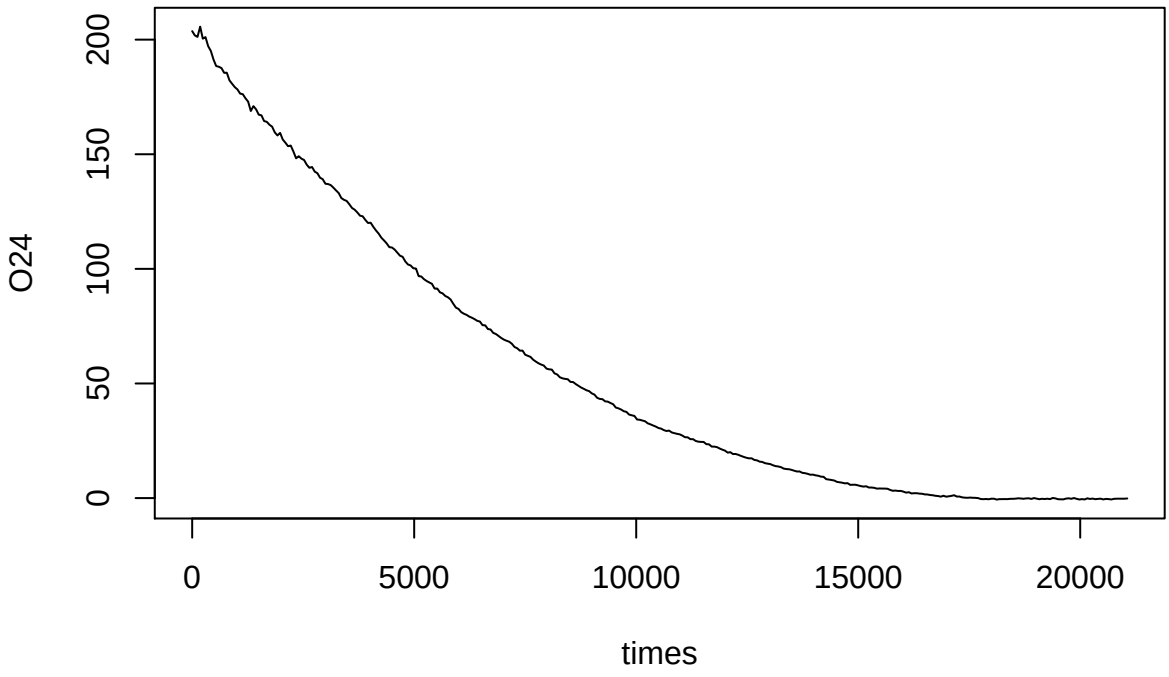
Reading in data

```
pcrit5=read.pyro("081914-09-05_Pcrit.txt")
```

```
## Warning in read.pyro("081914-09-05_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-09-05_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-09-05_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

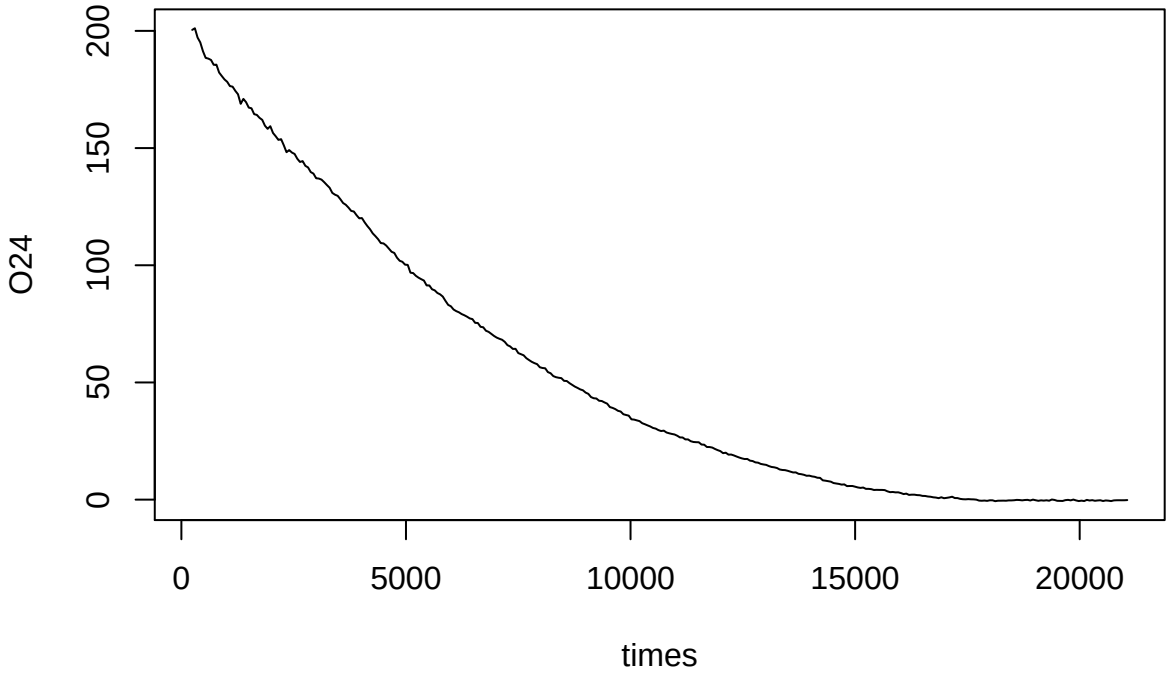
```
plot(O24~times,data=pcrit5,type="l")
```



###

Subsetting Data and replotting

```
pcrit5=pcrit5[5:nrow(pcrit5),]
plot(O24~times,data=pcrit5,type="l")
```



Calculating respiration rate

```
pcrit5.resp=closed.resp(pcrit5,volume=6,weight=193.6,channel=4,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit5$times==pcrit5.resp$time[1]):
  (which(pcrit5$times==pcrit5.resp$time[1])+
     length(pcrit5.resp$time)-1)
pcrit5.resp$kPa=umolO2L(pcrit5.resp$conc,temp=pcrit5$temp4[match],atmP=pcrit5$pressure[match])$kPa
pcrit5.resp=pcrit5.resp[,c(1,2,4,3)]
colnames(pcrit5.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit5.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit5.bin=aggregate(pcrit5.resp$resp~cut(pcrit5.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit5.bin)=c("bin","resp")
pcrit5.bin$bin=as.numeric(as.character(pcrit5.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

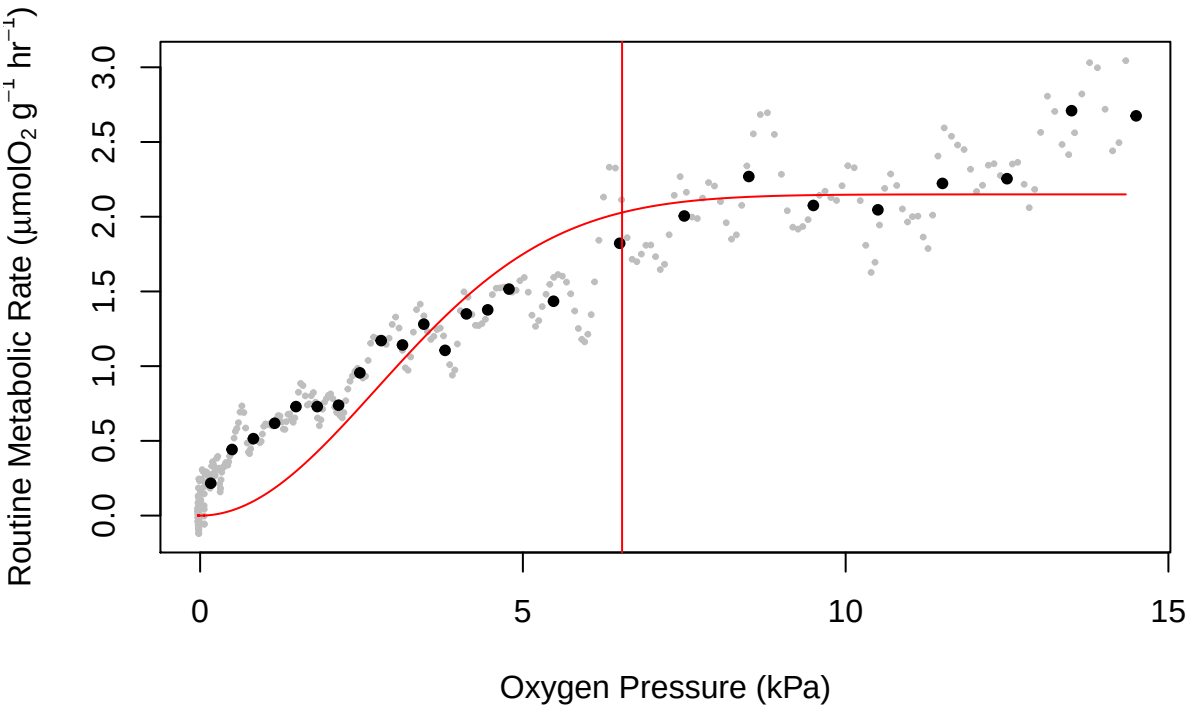
```
pcfit5=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit5.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit5)[2]
```

```
##       Pc
## 6.536773
```

```
kpa=pcrit5.resp$kPa
pcrit5.lm=lm(resp~kPa,data=pcrit5.resp)
pcrit5.seg=segmented(pcrit5.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                     control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit5.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
  ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*" g"^-1*" hr"^-1*")"))
points(resp~bin,data=pcrit5.bin,pch=20)
lines(pcrit5.resp$kPa[complete.cases(pcrit5.resp$resp)], predict(pcfit5),col="red")
abline(v=coef(pcfit5)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[1,]=c(5,coef(pcfit5)[2],pcrit5.seg$psi[2])
```

Octopus 8

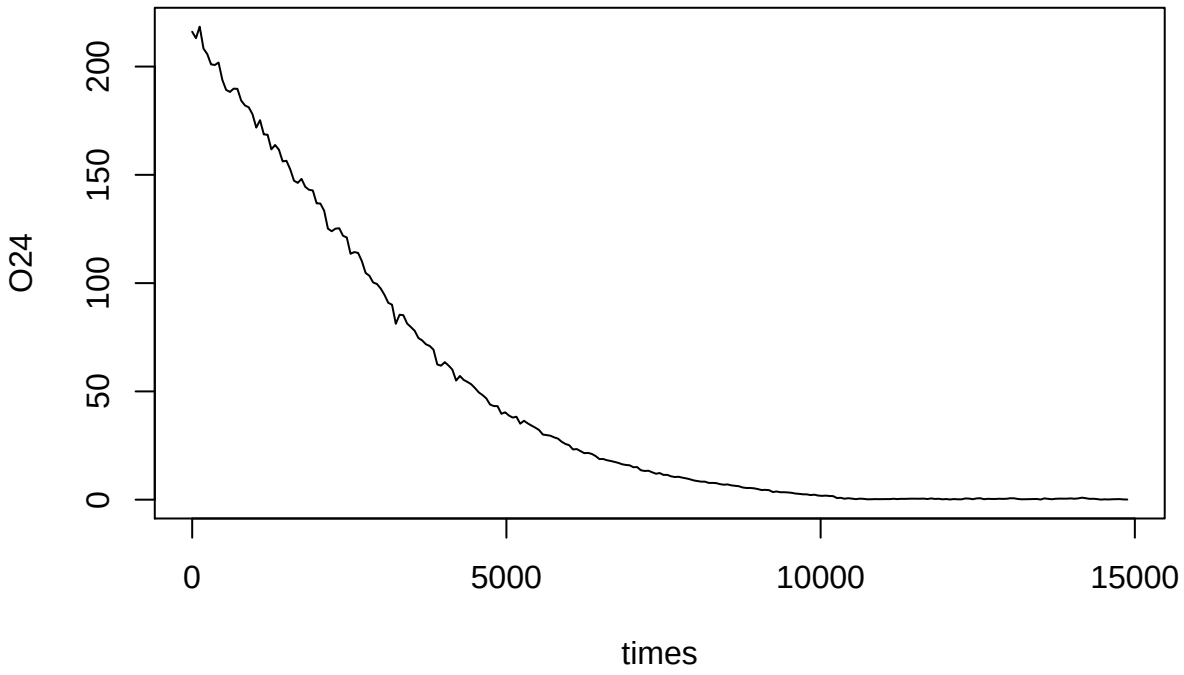
Reading in data

```
pcrit08=read.pyro("081814-12-08_Pcrit.txt")
```

```
## Warning in read.pyro("081814-12-08_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081814-12-08_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081814-12-08_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

```
plot(O24~times,data=pcrit08,type="l")
```



Calculating respiration rate

```
pcrit08.resp=closed.resp(pcrit08,volume=6,weight=184.2,channel=4,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit08$times==pcrit08.resp$time[1]):
  (which(pcrit08$times==pcrit08.resp$time[1])+
     length(pcrit08.resp$time)-1)
pcrit08.resp$kPa=umolO2L(pcrit08.resp$conc,temp=pcrit08$temp4[match],atmP=pcrit08$pressure[match])$kPa
pcrit08.resp=pcrit08.resp[,c(1,2,4,3)]
colnames(pcrit08.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit08.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit08.bin=aggregate(pcrit08.resp$resp~cut(pcrit08.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit08.bin)=c("bin","resp")
pcrit08.bin$bin=as.numeric(as.character(pcrit08.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

```
pcfit08=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit08.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit08)[2]
```

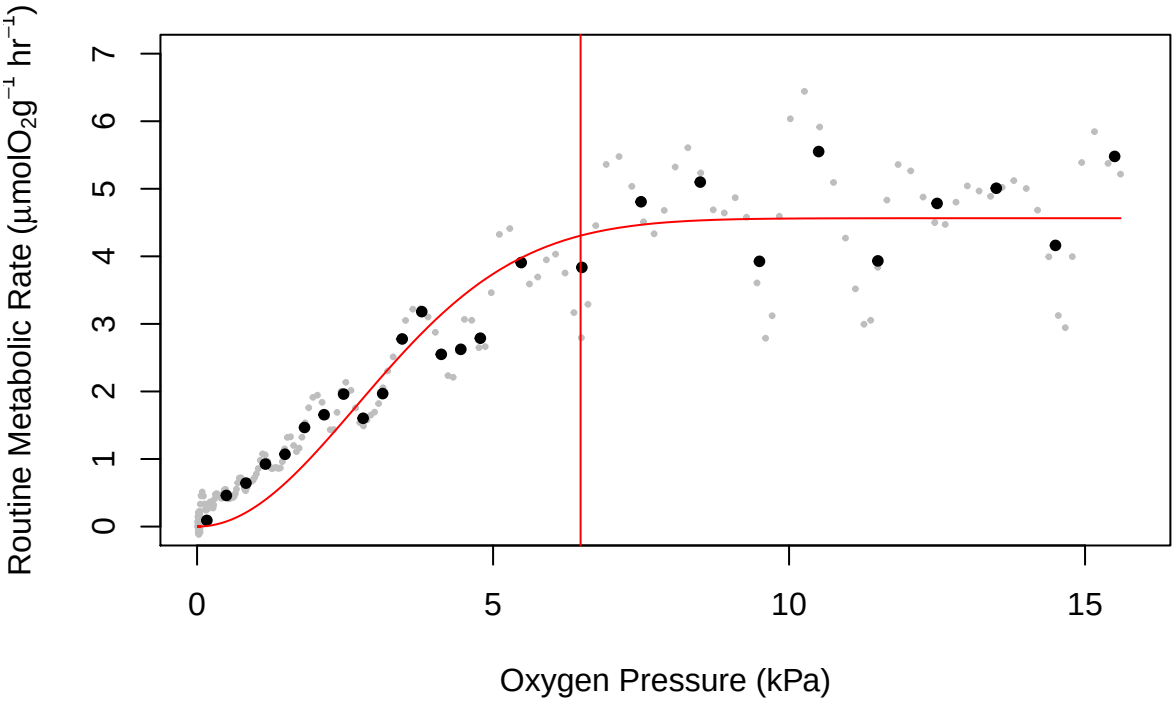
```
##       Pc
## 6.478113
```

```
kpa=pcrit08.resp$kPa
pcrit08.lm=lm(resp~kPa,data=pcrit08.resp)
```

```
pcrit08.seg=segmented(pcrit08.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit08.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,7))
points(resp~bin,data=pcrit08.bin,pch=20)
lines(pcrit08.resp$kPa[complete.cases(pcrit08.resp$resp)], predict(pcfit08),col="red")
abline(v=coef(pcfit08)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[2,]=c(08,coef(pcfit08)[2],pcrit08.seg$psi[2])
```

Generating Plot for paper

```
svg(file="Figure_2.svg",height=3.5,width=3.5,pointsize=6)
par(fig=c(0.04,1,0,1))
plot(resp~kPa,data=pcrit08.resp,pch=20,col="grey",cex=0.5,axes=F,ylab="",xlab="")
points(resp~bin,data=pcrit08.bin,pch=20)
lines(pcrit08.resp$kPa[complete.cases(pcrit08.resp$resp)], predict(pcfit08),col="red")
abline(v=coef(pcfit5)[2],col="red")
box(lwd=2)
axis(1,lwd=2,cex.axis=1.5)
axis(2,lwd=2,cex.axis=1.5,at=c(0,1,2,3))
mtext(expression("Routine Metabolic Rate ("*mu*"molO"[2]*" g"^-1*"hr"^-1*")"),
      side=2,cex=1.8,line=2.5)
mtext(expression("pO"[2]*" (kPa)"),side=1,cex=1.8,line=2.5)
dev.off()
```

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

Octopus 9

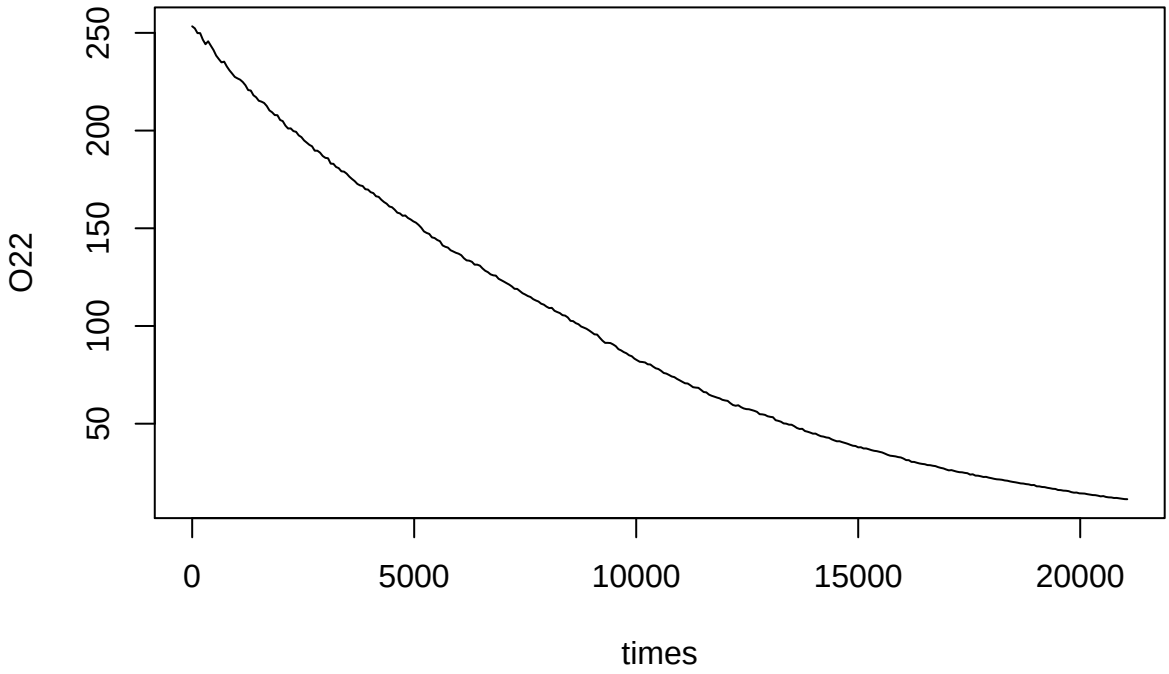
Reading in data

```
pcrit9=read.pyro("081914-09-05_Pcrit.txt")
```

```
## Warning in read.pyro("081914-09-05_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-09-05_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-09-05_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

```
plot(O22~times,data=pcrit9,type="l")
```



Calculating respiration rate

```
pcrit9.resp=closed.resp(pcrit9,volume=6,weight=157.8,channel=2,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit9$times==pcrit9.resp$time[1]):
  (which(pcrit9$times==pcrit9.resp$time[1])+
     length(pcrit9.resp$time)-1)
pcrit9.resp$kPa=umolO2L(pcrit9.resp$conc,temp=pcrit9$temp4[match],atmP=pcrit9$pressure[match])$kPa
pcrit9.resp=pcrit9.resp[,c(1,2,4,3)]
colnames(pcrit9.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit9.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit9.bin=aggregate(pcrit9.resp$resp~cut(pcrit9.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit9.bin)=c("bin","resp")
pcrit9.bin$bin=as.numeric(as.character(pcrit9.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

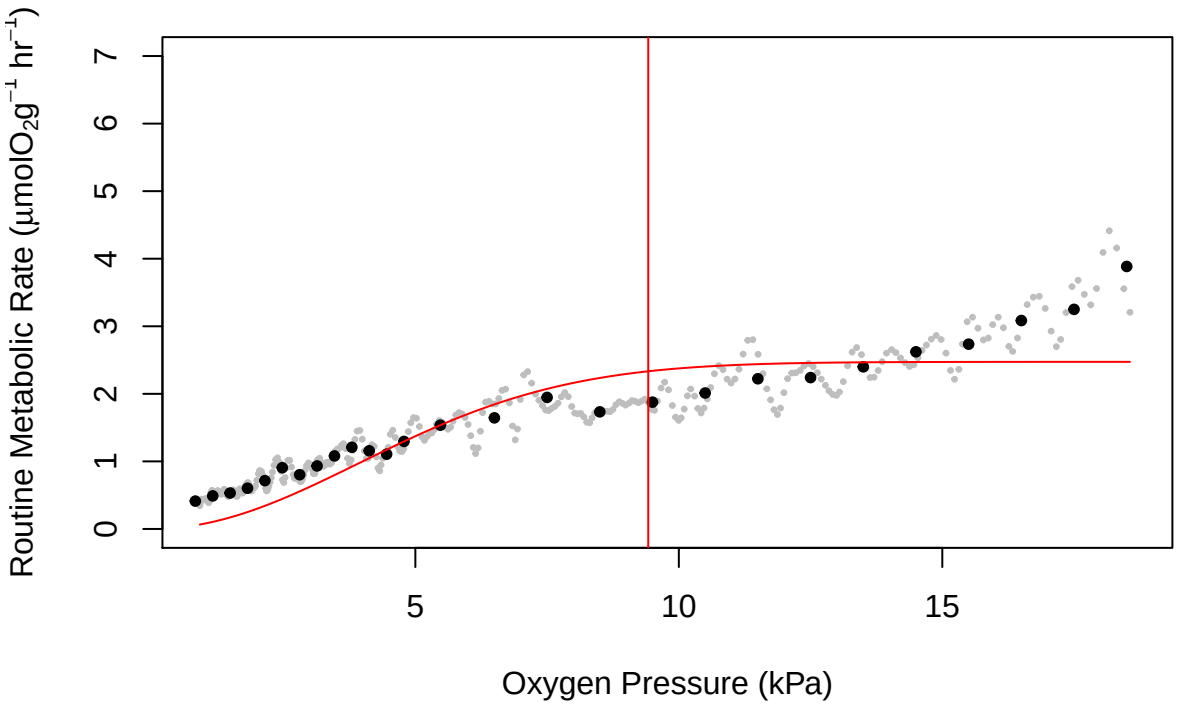
```
pcfit9=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit9.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit9)[2]
```

```
##      Pc
## 9.41952
```

```
kpa=pcrit9.resp$kPa
pcrit9.lm=lm(resp~kPa,data=pcrit9.resp)
pcrit9.seg=segmented(pcrit9.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                     control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit9.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,7))
points(resp~bin,data=pcrit9.bin,pch=20)
lines(pcrit9.resp$kPa[complete.cases(pcrit9.resp$resp)], predict(pcfit9),col="red")
abline(v=coef(pcfit9)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[3,]=c(9,coef(pcfit9)[2],pcrit9.seg$psi[2])
```

Octopus 10

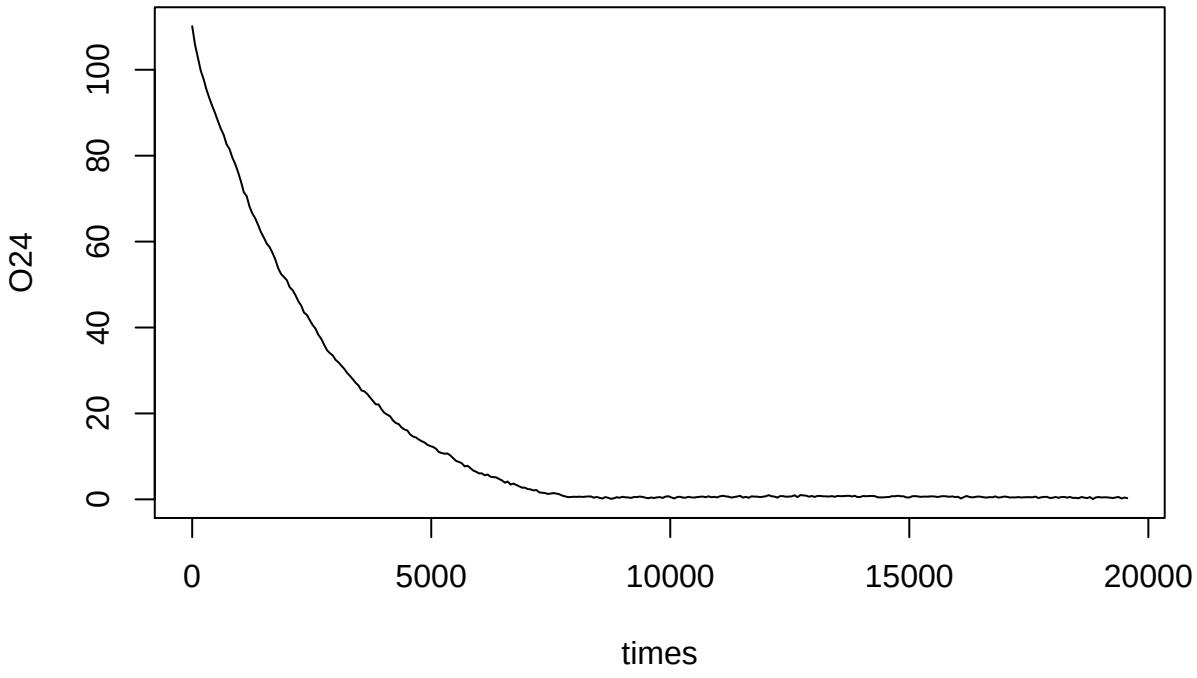
Reading in data

```
pcrit10=read.pyro("081814-15-10_Pcrit.txt")
```

```
## Warning in read.pyro("081814-15-10_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081814-15-10_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081814-15-10_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

```
plot(O24~times,data=pcrit10,type="l")
```



Calculating respiration rate

```
pcrit10.resp=closed.resp(pcrit10,volume=6,weight=262.9,channel=4,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit10$times==pcrit10.resp$time[1]):
  (which(pcrit10$times==pcrit10.resp$time[1])+
     length(pcrit10.resp$time)-1)
pcrit10.resp$kPa=umolO2L(pcrit10.resp$conc,temp=pcrit10$temp4[match],atmP=pcrit10$pressure[match])$kPa
pcrit10.resp=pcrit10.resp[,c(1,2,4,3)]
colnames(pcrit10.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit10.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit10.bin=aggregate(pcrit10.resp$resp~cut(pcrit10.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit10.bin)=c("bin","resp")
pcrit10.bin$bin=as.numeric(as.character(pcrit10.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

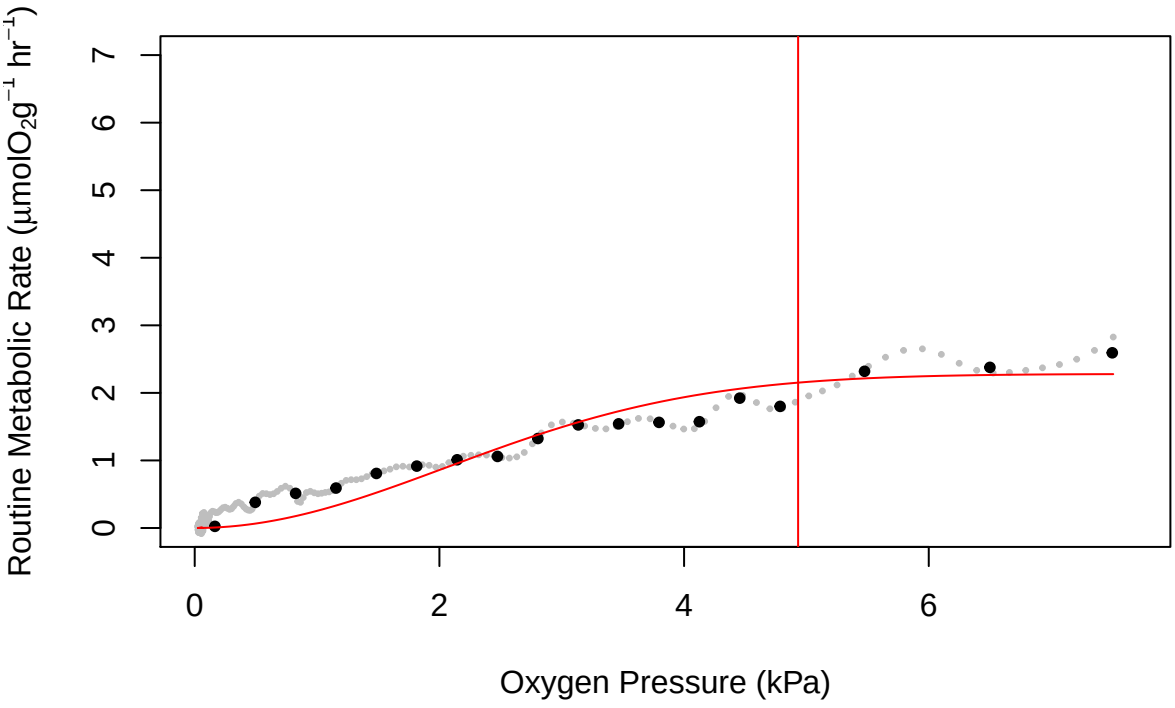
```
pcfit10=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit10.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit10)[2]
```

```
##       Pc
## 4.932009
```

```
kpa=pcrit10.resp$kPa
pcrit10.lm=lm(resp~kPa,data=pcrit10.resp)
pcrit10.seg=segmented(pcrit10.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit10.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,7))
points(resp~bin,data=pcrit10.bin,pch=20)
lines(pcrit10.resp$kPa[complete.cases(pcrit10.resp$resp)], predict(pcfit10),col="red")
abline(v=coef(pcfit10)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[4,]=c(10,coef(pcfit10)[2],pcrit10.seg$psi[2])
```

Octopus 12

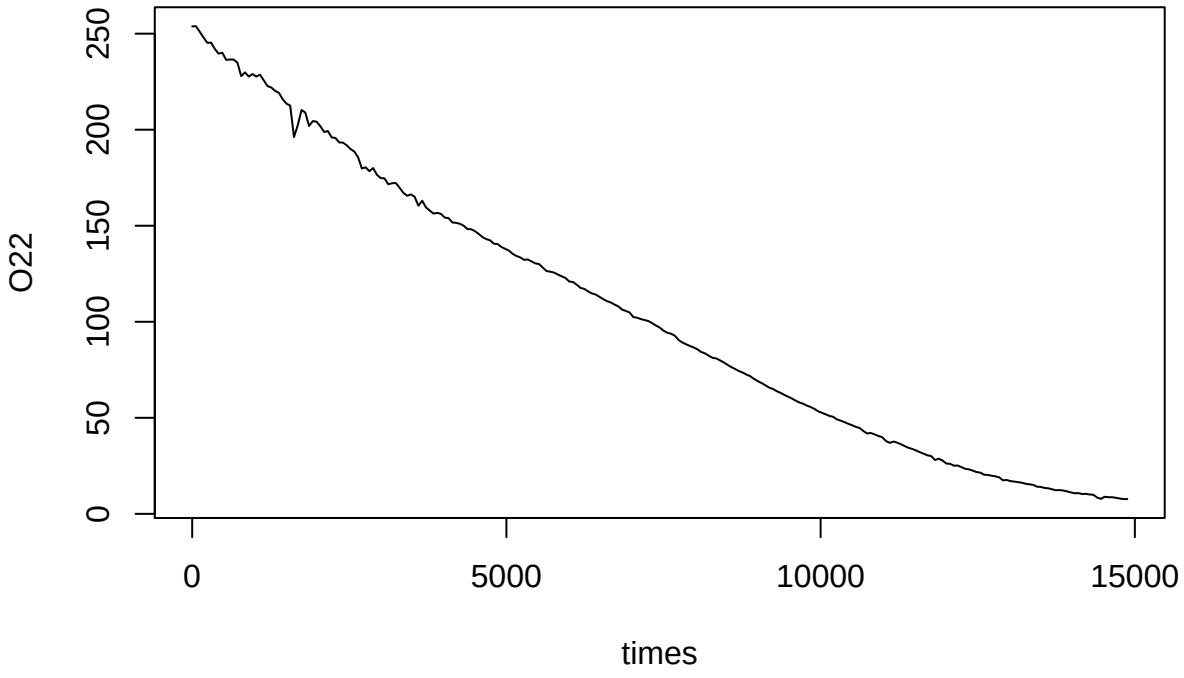
Reading in data

```
pcrit12=read.pyro("081814-12-08_Pcrit.txt")
```

```
## Warning in read.pyro("081814-12-08_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081814-12-08_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081814-12-08_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

```
plot(O22~times,data=pcrit12,type="l")
```



Calculating respiration rate

```
pcrit12.resp=closed.resp(pcrit12,volume=6,weight=149.6,channel=2,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit12$times==pcrit12.resp$time[1]):
  (which(pcrit12$times==pcrit12.resp$time[1])+
     length(pcrit12.resp$time)-1)
pcrit12.resp$kPa=umolO2L(pcrit12.resp$conc,temp=pcrit12$temp4[match],atmP=pcrit12$pressure[match])$kPa
pcrit12.resp=pcrit12.resp[,c(1,2,4,3)]
colnames(pcrit12.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit12.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit12.bin=aggregate(pcrit12.resp$resp~cut(pcrit12.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit12.bin)=c("bin","resp")
pcrit12.bin$bin=as.numeric(as.character(pcrit12.bin$bin))
```

Calculating P_{CRIT} using Weibull function and Broken Stick

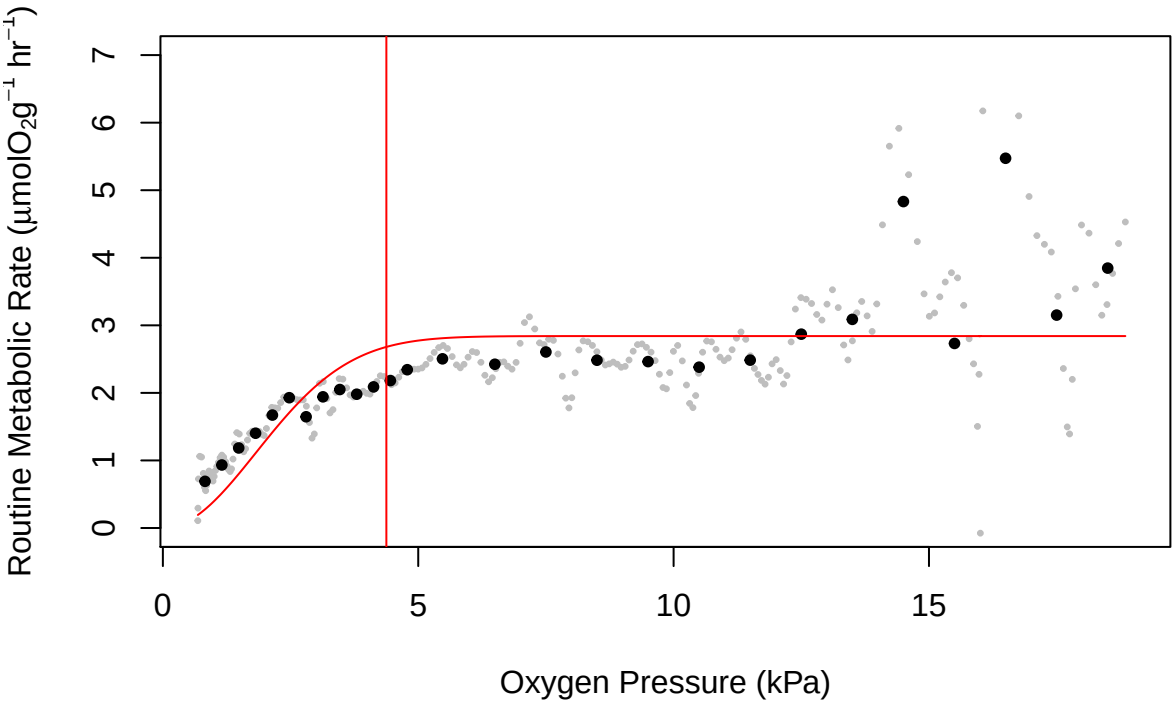
```
pcfit12=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit12.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit12)[2]
```

```
##       Pc
## 4.375791
```

```
kpa=pcrit12.resp$kPa
pcrit12.lm=lm(resp~kPa,data=pcrit12.resp)
pcrit12.seg=segmented(pcrit12.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit12.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,7))
points(resp~bin,data=pcrit12.bin,pch=20)
lines(pcrit12.resp$kPa[complete.cases(pcrit12.resp$resp)], predict(pcfit12),col="red")
abline(v=coef(pcfit12)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[5,]=c(12,coef(pcfit12)[2],pcrit12.seg$psi[2])
```

Octopus 15

Reading in data

```
pcrit15=read.pyro("081814-15-10_Pcrit.txt")
```

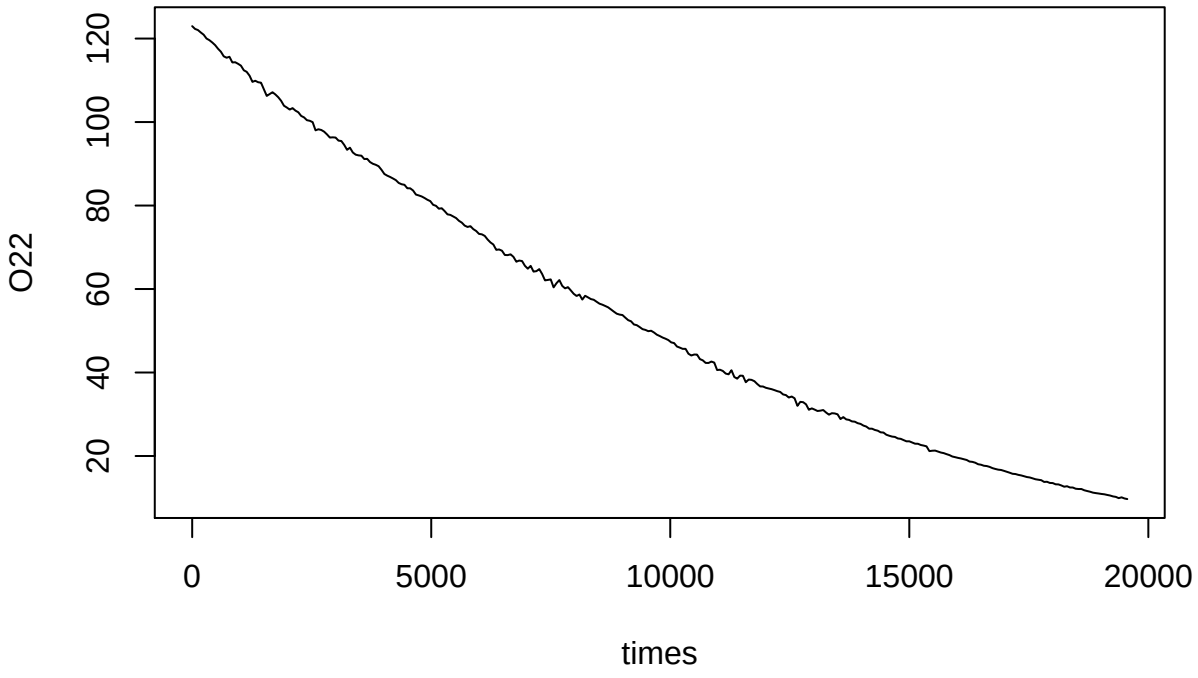
```
## Warning in read.pyro("081814-15-10_Pcrit.txt"): NAs introduced by coercion
```

```
## Warning in read.pyro("081814-15-10_Pcrit.txt"): NAs introduced by coercion
```

```
## Warning in read.pyro("081814-15-10_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

```
plot(O22~times,data=pcrit15,type="l")
```



Calculating respiration rate

```
pcrit15.resp=closed.resp(pcrit15,volume=6,weight=157.8,channel=2,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit15$times==pcrit15.resp$time[1]):
  (which(pcrit15$times==pcrit15.resp$time[1])+
     length(pcrit15.resp$time)-1)
pcrit15.resp$kPa=umolO2L(pcrit15.resp$conc,temp=pcrit15$temp4[match],atmP=pcrit15$pressure[match])$kPa
pcrit15.resp=pcrit15.resp[,c(1,2,4,3)]
colnames(pcrit15.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit15.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit15.bin=aggregate(pcrit15.resp$resp~cut(pcrit15.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit15.bin)=c("bin","resp")
pcrit15.bin$bin=as.numeric(as.character(pcrit15.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

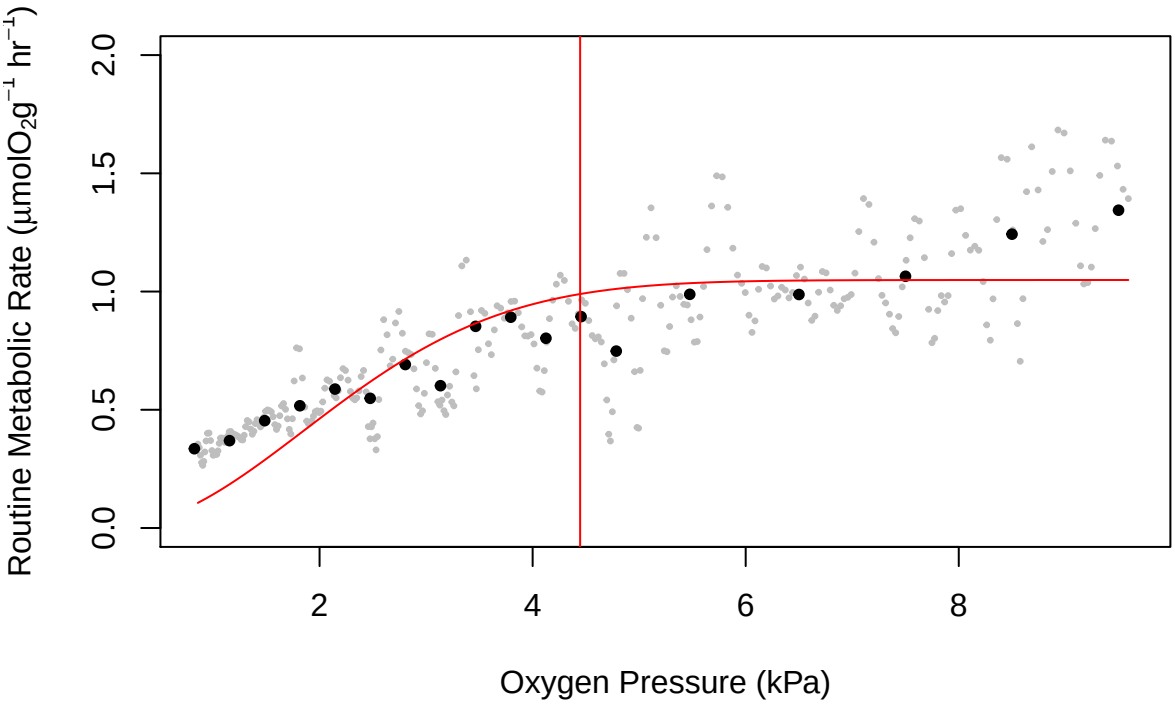
```
pcfit15=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit15.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit15)[2]
```

```
##       Pc
## 4.445873
```

```
kpa=pcrit15.resp$kPa
pcrit15.lm=lm(resp~kPa,data=pcrit15.resp)
pcrit15.seg=segmented(pcrit15.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit15.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,2))
points(resp~bin,data=pcrit15.bin,pch=20)
lines(pcrit15.resp$kPa[complete.cases(pcrit15.resp$resp)], predict(pcfit15),col="red")
abline(v=coef(pcfit15)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[6,]=c(15,coef(pcfit15)[2],pcrit15.seg$psi[2])
```

Octopus 16

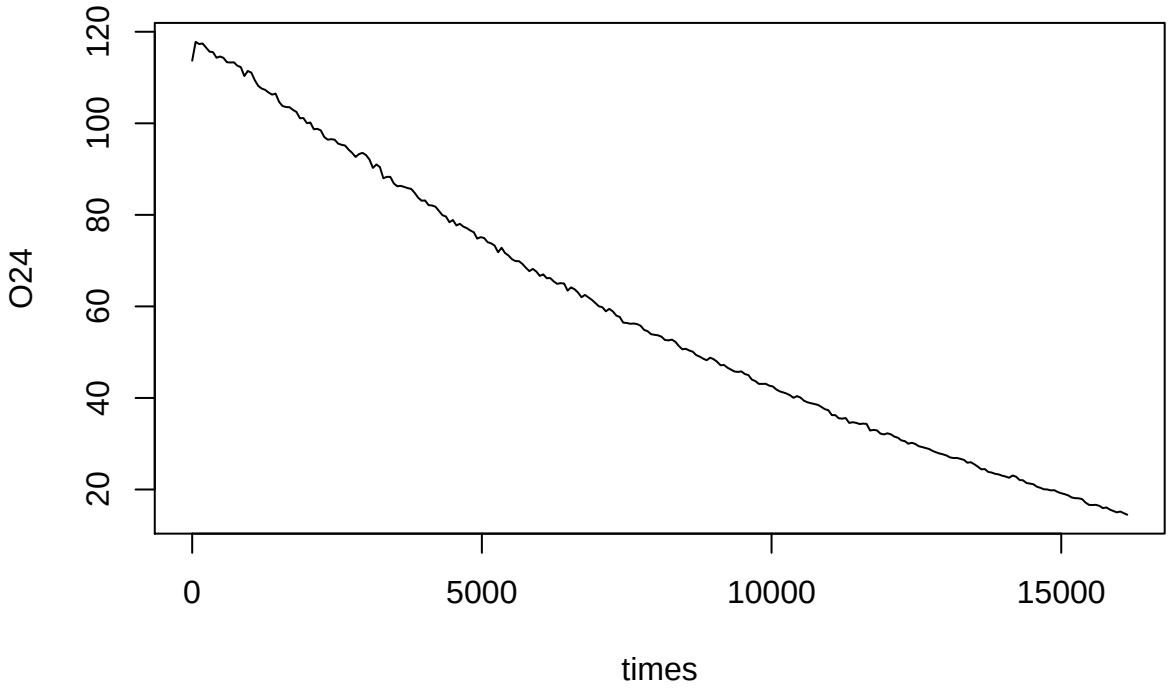
Reading in data

```
pcrit16=read.pyro("081914-17-16_Pcrit.txt")
```

```
## Warning in read.pyro("081914-17-16_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-17-16_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-17-16_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

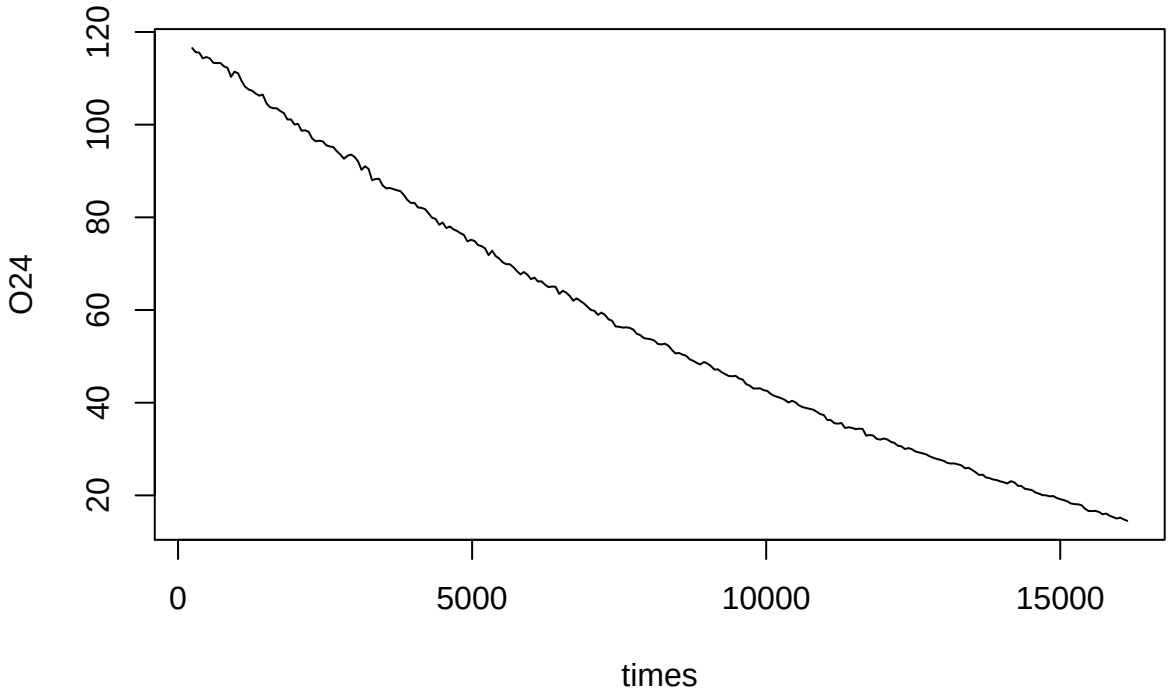
```
plot(O24~times,data=pcrit16,type="l")
```



###

Subsetting Data and replotting

```
pcrit16=pcrit16[5:nrow(pcrit16),]
plot(O24~times,data=pcrit16,type="l")
```



Calculating respiration rate

```
pcrit16.resp=closed.resp(pcrit16,volume=6,weight=184.2,channel=4,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit16$times==pcrit16.resp$time[1]):
  (which(pcrit16$times==pcrit16.resp$time[1])+
     length(pcrit16.resp$time)-1)
pcrit16.resp$kPa=umolO2L(pcrit16.resp$conc,temp=pcrit16$temp4[match],atmP=pcrit16$pressure[match])$kPa
pcrit16.resp=pcrit16.resp[,c(1,2,4,3)]
colnames(pcrit16.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit16.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit16.bin=aggregate(pcrit16.resp$resp~cut(pcrit16.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit16.bin)=c("bin","resp")
pcrit16.bin$bin=as.numeric(as.character(pcrit16.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

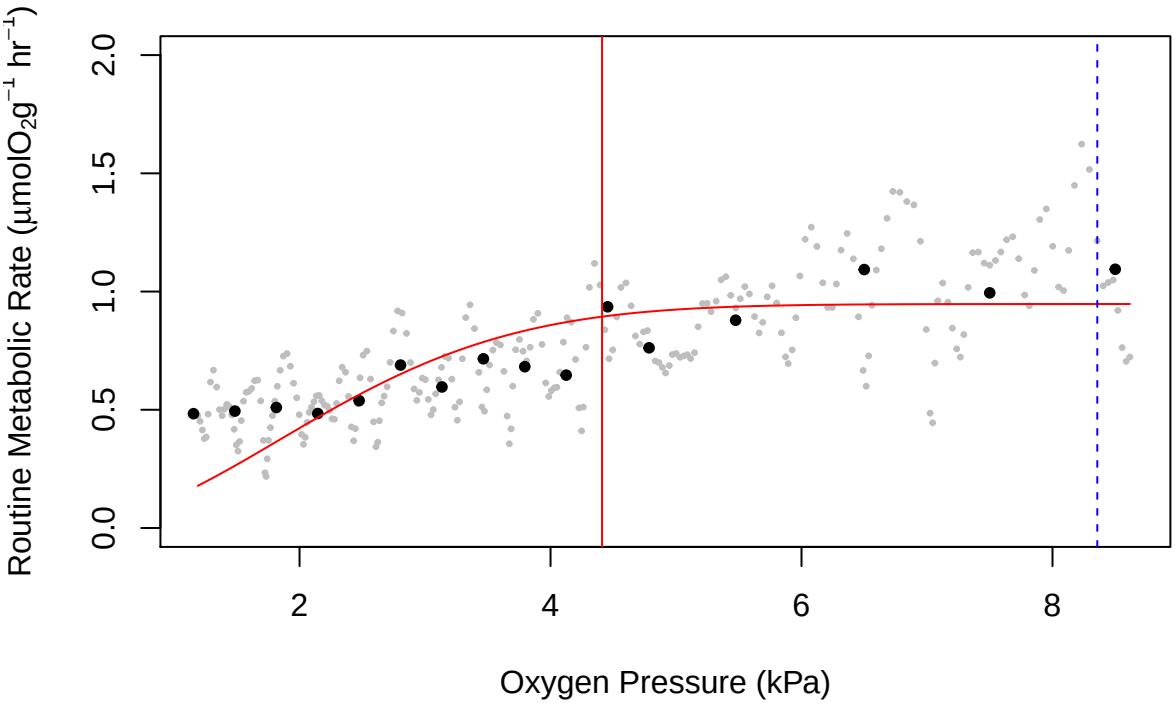
```
pcfit16=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit16.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit16)[2]
```

```
##       Pc
## 4.410653
```

```
kpa=pcrit16.resp$kPa
pcrit16.lm=lm(resp~kPa,data=pcrit16.resp)
pcrit16.seg=segmented(pcrit16.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit16.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,2))
points(resp~bin,data=pcrit16.bin,pch=20)
lines(pcrit16.resp$kPa[complete.cases(pcrit16.resp$resp)], predict(pcfit16),col="red")
abline(v=coef(pcfit16)[2],col="red")
abline(v=pcrit16.seg$psi[2],lty=2,col="blue")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[7,]=c(16,coef(pcfit16)[2],pcrit16.seg$psi[2])
```

Octopus 17

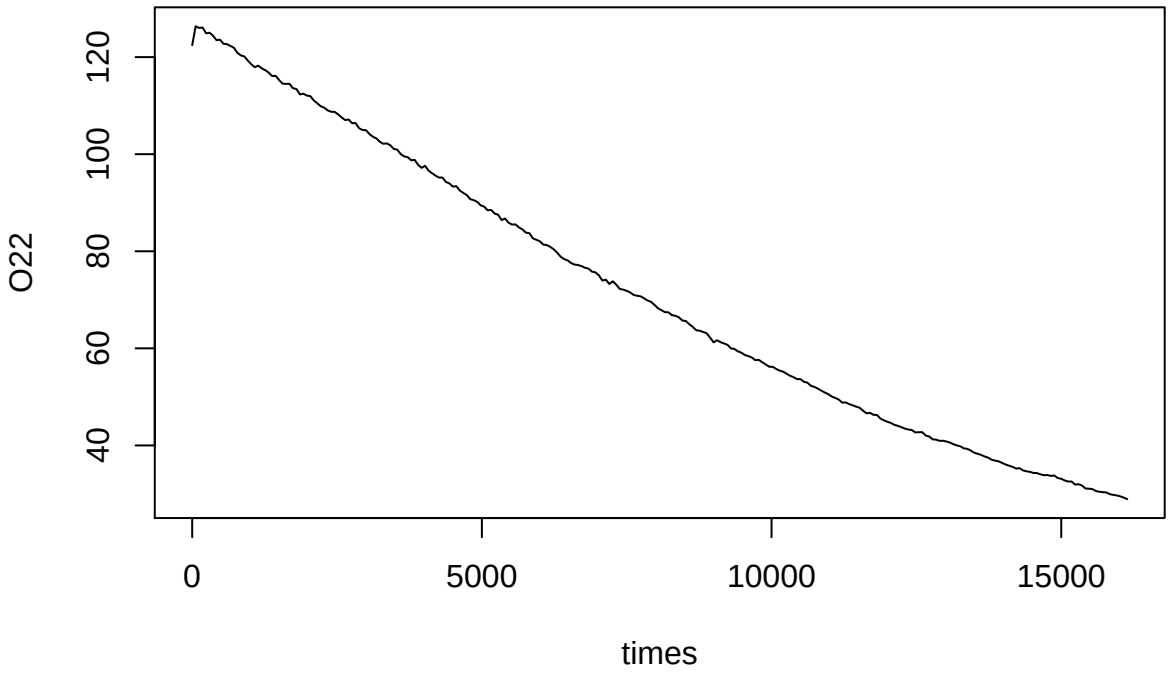
Reading in data

```
pcrit17=read.pyro("081914-17-16_Pcrit.txt")
```

```
## Warning in read.pyro("081914-17-16_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-17-16_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("081914-17-16_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

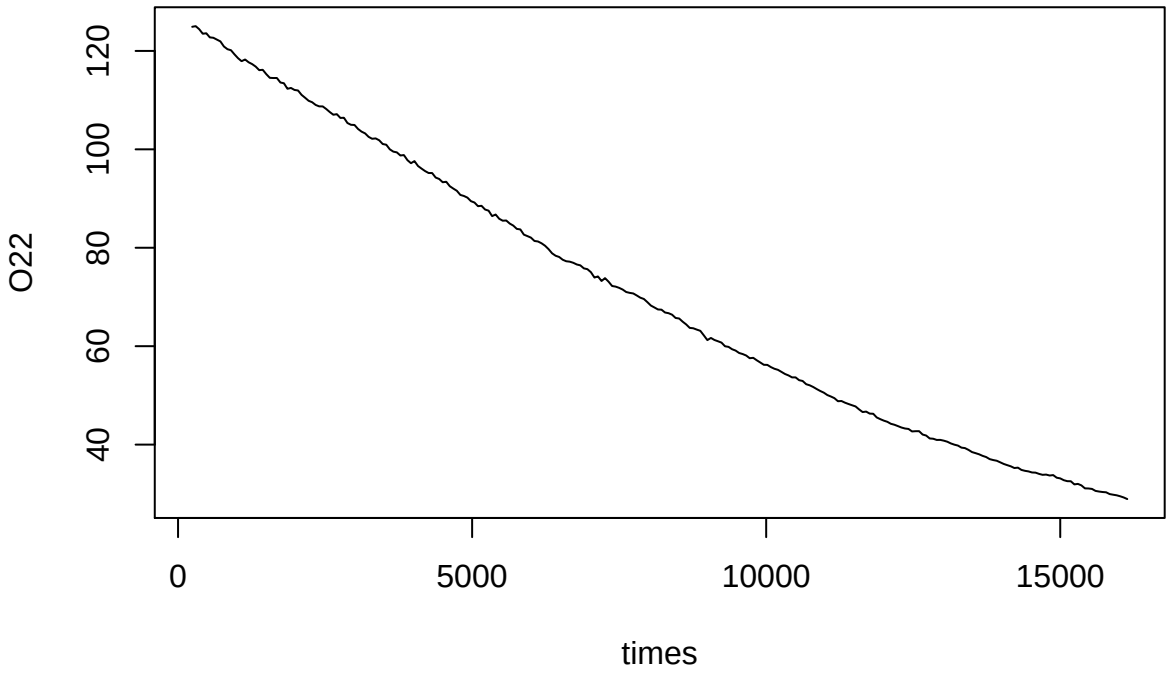
```
plot(O22~times,data=pcrit17,type="l")
```



###

Subsetting Data and replotting

```
pcrit17=pcrit17[5:nrow(pcrit17),]
plot(O22~times,data=pcrit17,type="l")
```



Calculating respiration rate

```
pcrit17.resp=closed.resp(pcrit17,volume=6,weight=184.2,channel=2,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcrit17$times==pcrit17.resp$time[1]):
  (which(pcrit17$times==pcrit17.resp$time[1])+
     length(pcrit17.resp$time)-1)
pcrit17.resp$kPa=umolO2L(pcrit17.resp$conc,temp=pcrit17$temp4[match],atmP=pcrit17$pressure[match])$kPa
pcrit17.resp=pcrit17.resp[,c(1,2,4,3)]
colnames(pcrit17.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcrit17.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcrit17.bin=aggregate(pcrit17.resp$resp~cut(pcrit17.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcrit17.bin)=c("bin","resp")
pcrit17.bin$bin=as.numeric(as.character(pcrit17.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

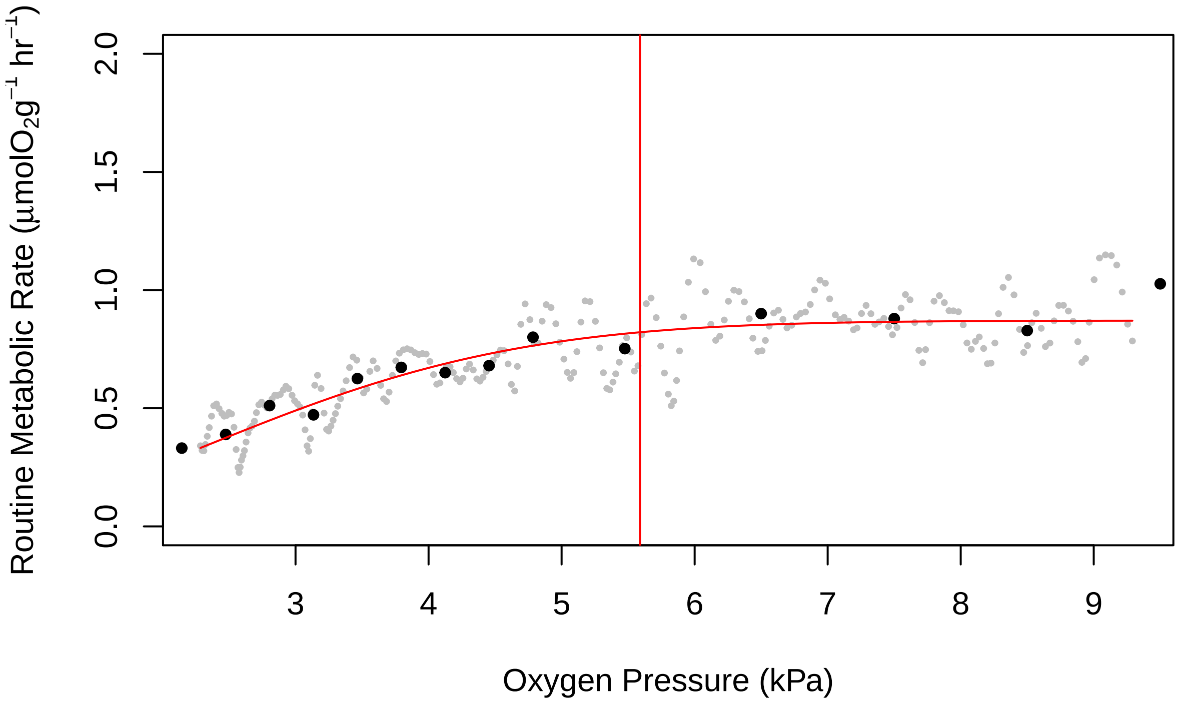
```
pcfit17=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcrit17.resp,start=list(Bm=1.5,Pc=5))
coef(pcfit17)[2]
```

```
##       Pc
## 5.589634
```

```
kpa=pcrit17.resp$kPa
pcrit17.lm=lm(resp~kPa,data=pcrit17.resp)
pcrit17.seg=segmented(pcrit17.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcrit17.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,2))
points(resp~bin,data=pcrit17.bin,pch=20)
lines(pcrit17.resp$kPa[complete.cases(pcrit17.resp$resp)], predict(pcfit17),col="red")
abline(v=coef(pcfit17)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[8,]=c(17,coef(pcfit17)[2],pcrit17.seg$psi[2])
```

Octopus L1

Reading in data

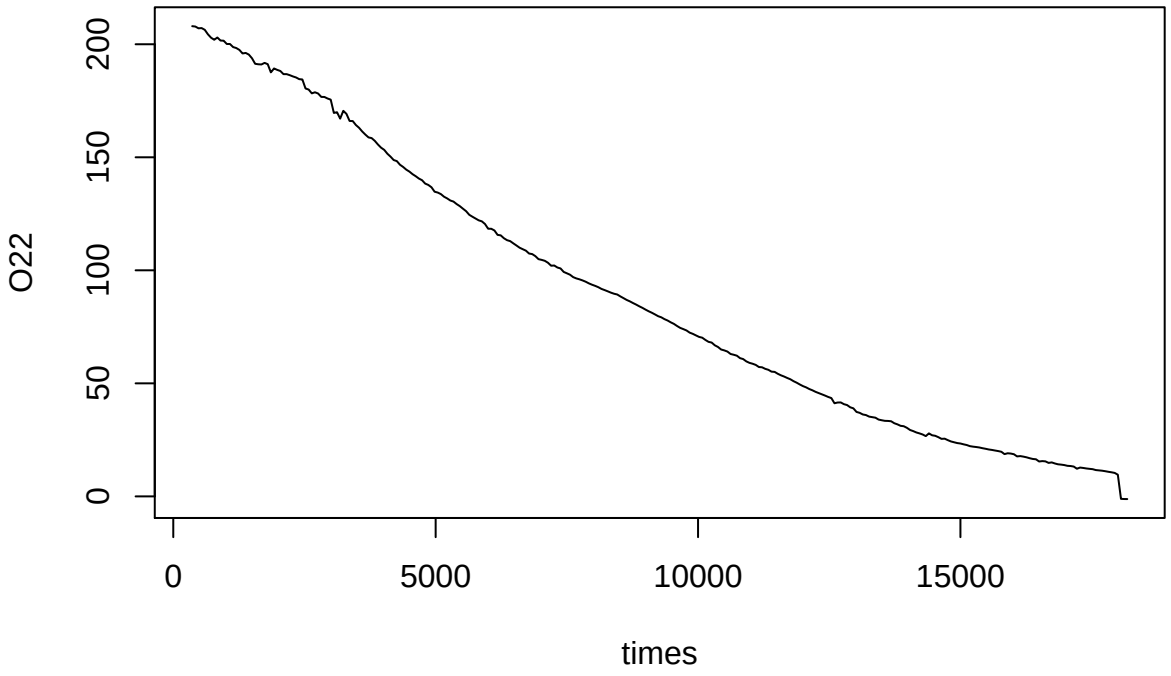
```
pcritL1=read.pyro("080514-L1-L2_Pcrit.txt")
```

```
## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion
```

```
## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

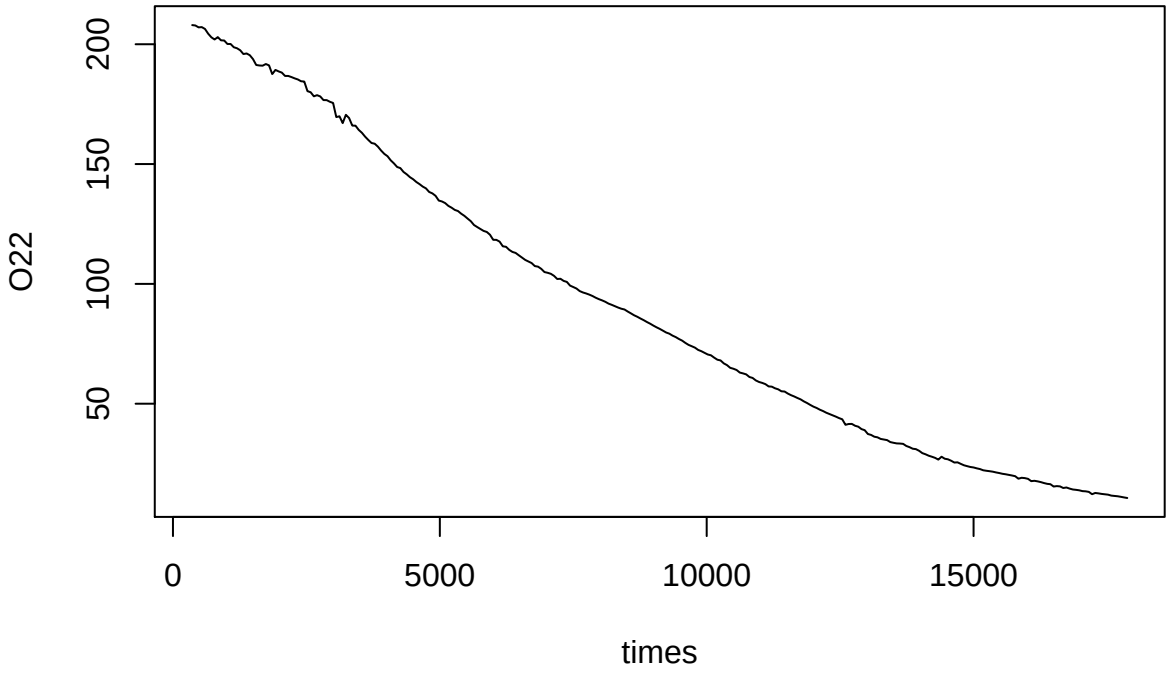
```
plot(O22~times,data=pcritL1,type="l")
```



###

Subsetting Data and replotting

```
pcritL1=pcritL1[1:(nrow(pcritL1)-5),]
plot(O22~times,data=pcritL1,type="l")
```



Calculating respiration rate

```
pcritL1.resp=closed.resp(pcritL1,volume=6,weight=187.1,channel=2,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcritL1$times==pcritL1.resp$time[1]):
  (which(pcritL1$times==pcritL1.resp$time[1])+
     length(pcritL1.resp$time)-1)
pcritL1.resp$kPa=umolO2L(pcritL1.resp$conc,temp=pcritL1$temp4[match],atmP=pcritL1$pressure[match])$kPa
pcritL1.resp=pcritL1.resp[,c(1,2,4,3)]
colnames(pcritL1.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcritL1.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcritL1.bin=aggregate(pcritL1.resp$resp~cut(pcritL1.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcritL1.bin)=c("bin","resp")
pcritL1.bin$bin=as.numeric(as.character(pcritL1.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

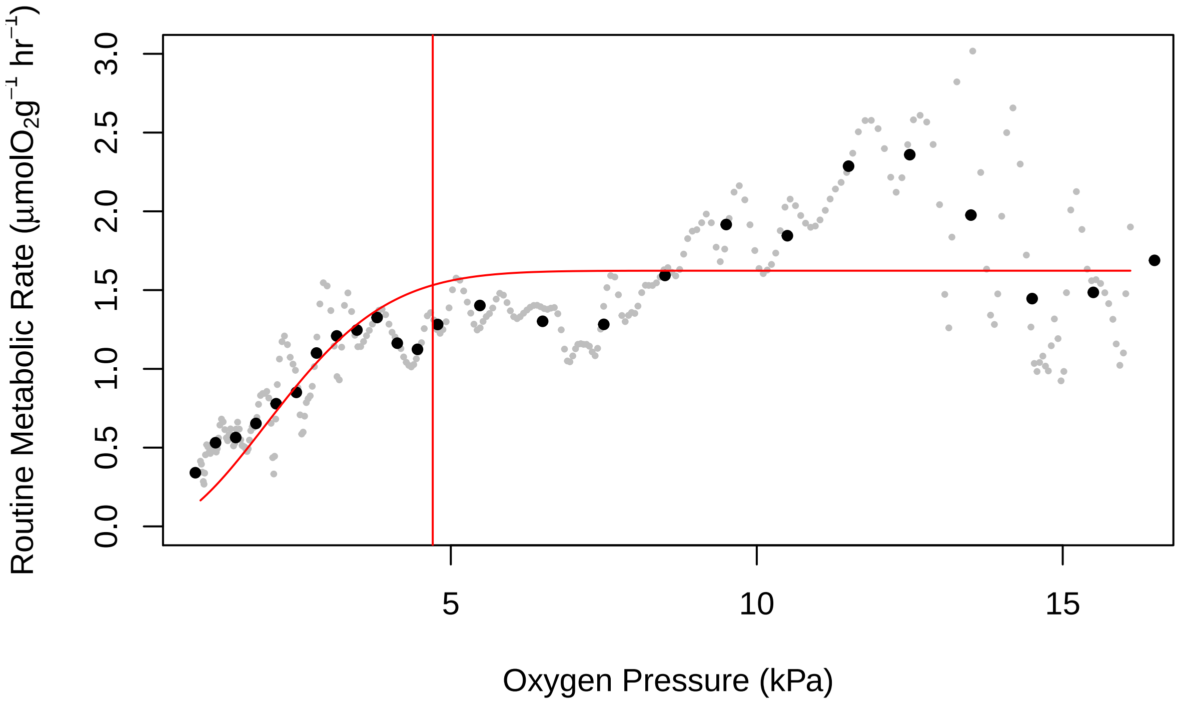
```
pcfitL1=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcritL1.resp,start=list(Bm=1.5,Pc=5))
coef(pcfitL1)[2]
```

```
##       Pc
## 4.704231
```

```
kpa=pcritL1.resp$kPa
pcritL1.lm=lm(resp~kPa,data=pcritL1.resp)
pcritL1.seg=segmented(pcritL1.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcritL1.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,3))
points(resp~bin,data=pcritL1.bin,pch=20)
lines(pcritL1.resp$kPa[complete.cases(pcritL1.resp$resp)], predict(pcfitL1),col="red")
abline(v=coef(pcfitL1)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[9,]=c("L1",coef(pcfitL1)[2],pcritL1.seg$psi[2])
```

Octopus L2

Reading in data

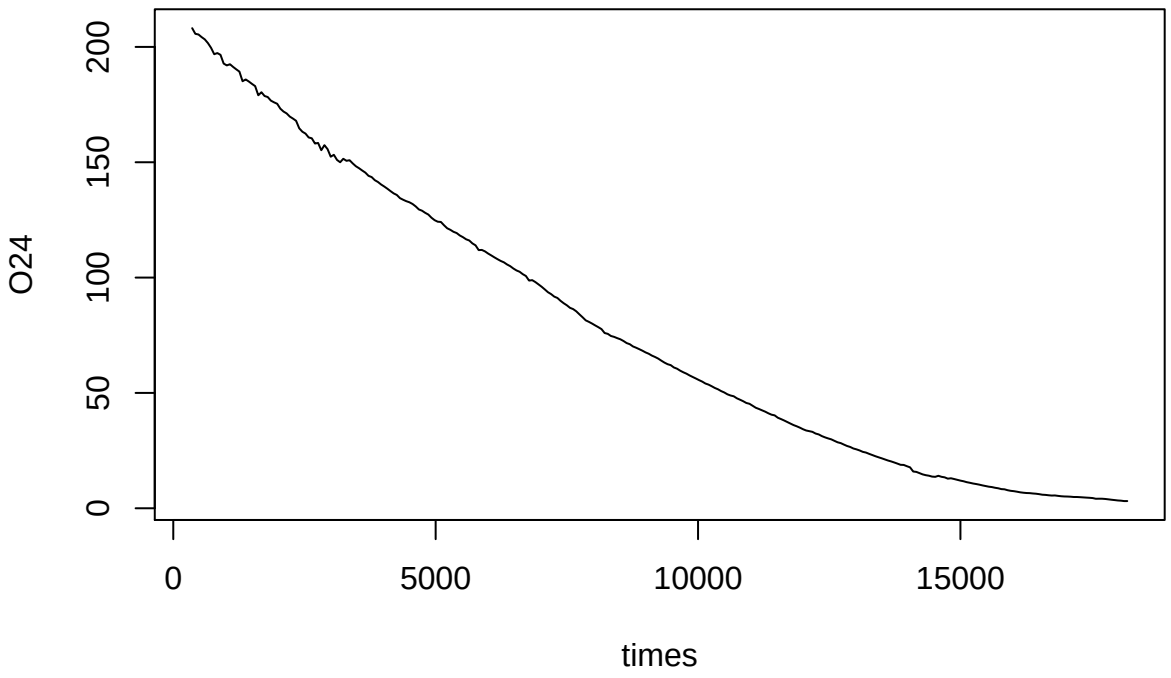
```
pcritL2=read.pyro("080514-L1-L2_Pcrit.txt")
```

```
## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion
```

```
## Warning in read.pyro("080514-L1-L2_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

```
plot(O24~times,data=pcritL2,type="l")
```



Calculating respiration rate

```
pcritL2.resp=closed.resp(pcritL2,volume=6,weight=160.3,channel=4,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcritL2$times==pcritL2.resp$time[1]):
  (which(pcritL2$times==pcritL2.resp$time[1])+
     length(pcritL2.resp$time)-1)
pcritL2.resp$kPa=umolO2L(pcritL2.resp$conc,temp=pcritL2$temp4[match],atmP=pcritL2$pressure[match])$kPa
pcritL2.resp=pcritL2.resp[,c(1,2,4,3)]
colnames(pcritL2.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcritL2.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcritL2.bin=aggregate(pcritL2.resp$resp~cut(pcritL2.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcritL2.bin)=c("bin","resp")
pcritL2.bin$bin=as.numeric(as.character(pcritL2.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

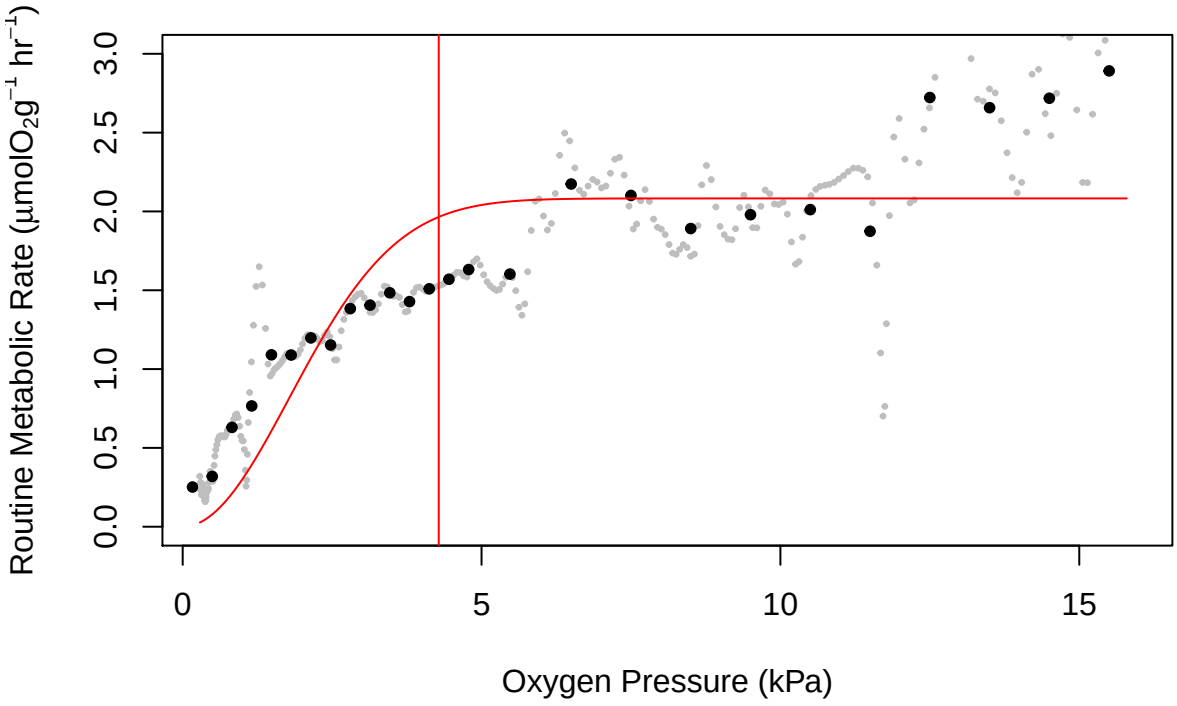
```
pcfitL2=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcritL2.resp,start=list(Bm=1.5,Pc=5))
coef(pcfitL2)[2]
```

```
##       Pc
## 4.284009
```

```
kpa=pcritL2.resp$kPa
pcritL2.lm=lm(resp~kPa,data=pcritL2.resp)
pcritL2.seg=segmented(pcritL2.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcritL2.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,3))
points(resp~bin,data=pcritL2.bin,pch=20)
lines(pcritL2.resp$kPa[complete.cases(pcritL2.resp$resp)], predict(pcfitL2),col="red")
abline(v=coef(pcfitL2)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[10,]=c("L2",coef(pcfitL2)[2],pcritL2.seg$psi[2])
```

Octopus L3

Reading in data

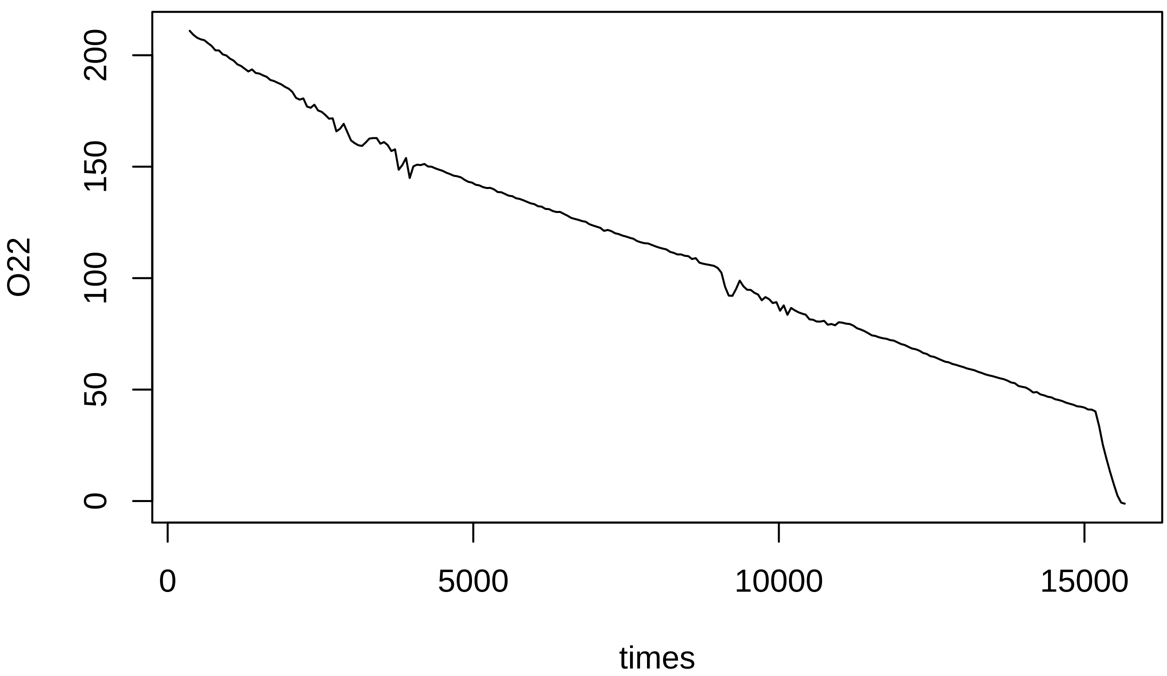
```
pcritL3=read.pyro("080514-L3-L4_Pcrit.txt")
```

```
## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion
```

```
## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

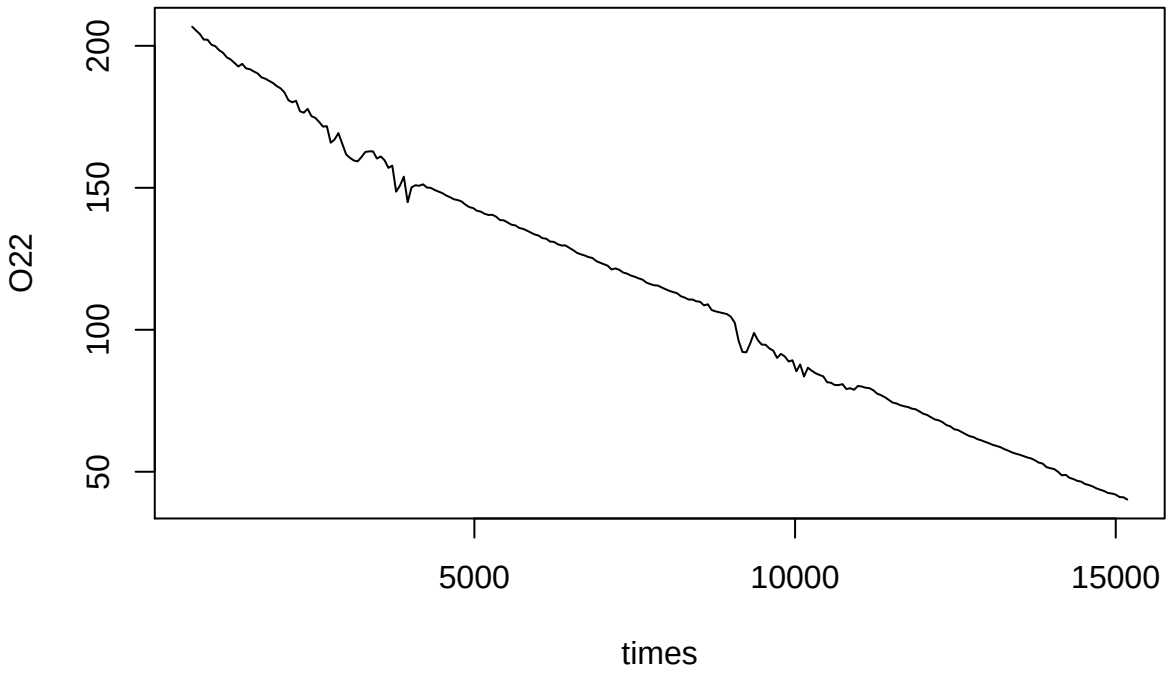
```
plot(O22~times,data=pcritL3,type="l")
```



###

Subsetting Data and replotting

```
pcritL3=pcritL3[5:(nrow(pcritL3)-8),]
plot(O22~times,data=pcritL3,type="l")
```



Calculating respiration rate

```
pcritL3.resp=closed.resp(pcritL3,volume=6,weight=132.3,channel=2,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcritL3$times==pcritL3.resp$time[1]):
  (which(pcritL3$times==pcritL3.resp$time[1])+
     length(pcritL3.resp$time)-1)
pcritL3.resp$kPa=umolO2L(pcritL3.resp$conc,temp=pcritL3$temp4[match],atmP=pcritL3$pressure[match])$kPa
pcritL3.resp=pcritL3.resp[,c(1,2,4,3)]
colnames(pcritL3.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcritL3.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcritL3.bin=aggregate(pcritL3.resp$resp~cut(pcritL3.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcritL3.bin)=c("bin","resp")
pcritL3.bin$bin=as.numeric(as.character(pcritL3.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

```
pcfitL3=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcritL3.resp,start=list(Bm=1.5,Pc=5))
coef(pcfitL3)[2]
```

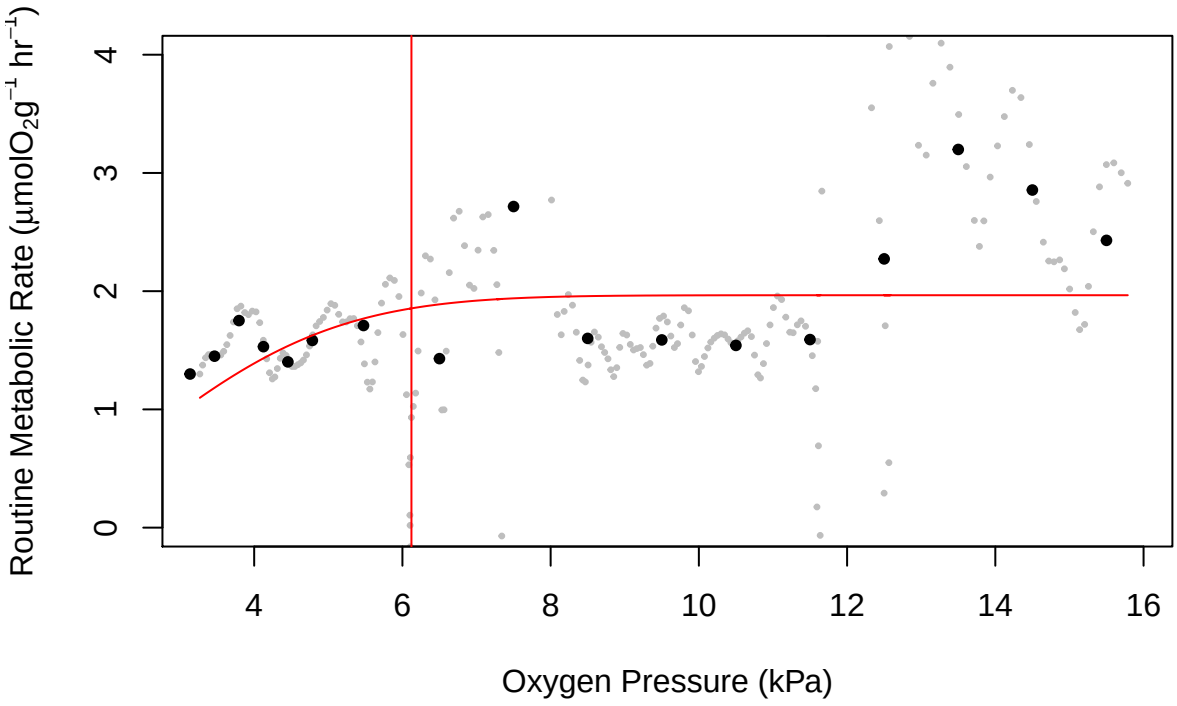
```
##       Pc
## 6.121861
```

```
kpa=pcritL3.resp$kPa
pcritL3.lm=lm(resp~kPa,data=pcritL3.resp)
```

```
pcritL3.seg=segmented(pcritL3.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcritL3.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,4))
points(resp~bin,data=pcritL3.bin,pch=20)
lines(pcritL3.resp$kPa[complete.cases(pcritL3.resp$resp)], predict(pcfitL3),col="red")
abline(v=coef(pcfitL3)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[11,]=c("L3",coef(pcfitL3)[2],pcritL3.seg$psi[2])
```

Octopus L4

Reading in data

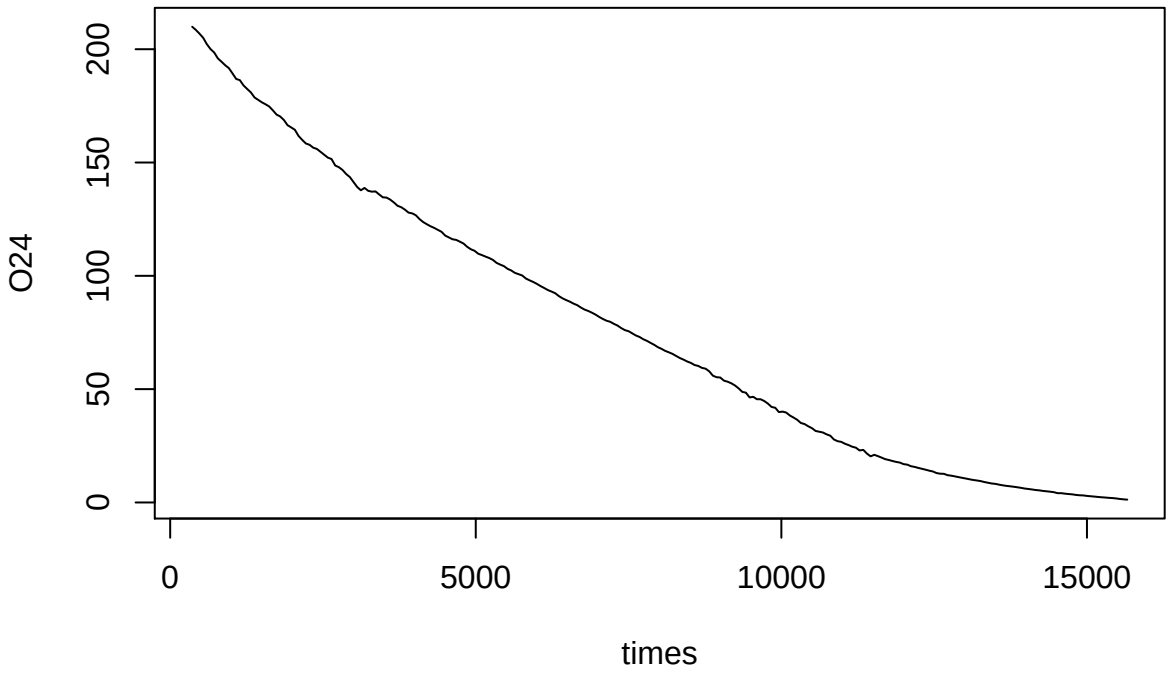
```
pcritL4=read.pyro("080514-L3-L4_Pcrit.txt")
```

```
## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion
```

```
## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion

## Warning in read.pyro("080514-L3-L4_Pcrit.txt"): NAs introduced by coercion
```

Plot of oxygen over time

```
plot(O24~times,data=pcritL4,type="l")
```



Calculating respiration rate

```
pcritL4.resp=closed.resp(pcritL4,volume=6,weight=149.3,channel=4,smooth=10)
```

Calculating oxygen pressure

```
match=which(pcritL4$times==pcritL4.resp$time[1]):
  (which(pcritL4$times==pcritL4.resp$time[1])+
     length(pcritL4.resp$time)-1)
pcritL4.resp$kPa=umolO2L(pcritL4.resp$conc,temp=pcritL4$temp4[match],atmP=pcritL4$pressure[match])$kPa
pcritL4.resp=pcritL4.resp[,c(1,2,4,3)]
colnames(pcritL4.resp)=c("time(s)","umolO2.per.L","kPa","resp")
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(pcritL4.resp$kPa,na.rm=T)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
pcritL4.bin=aggregate(pcritL4.resp$resp~cut(pcritL4.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(pcritL4.bin)=c("bin","resp")
pcritL4.bin$bin=as.numeric(as.character(pcritL4.bin$bin))
```

Calculating P_{CRIT} using Weibull function and broken stick

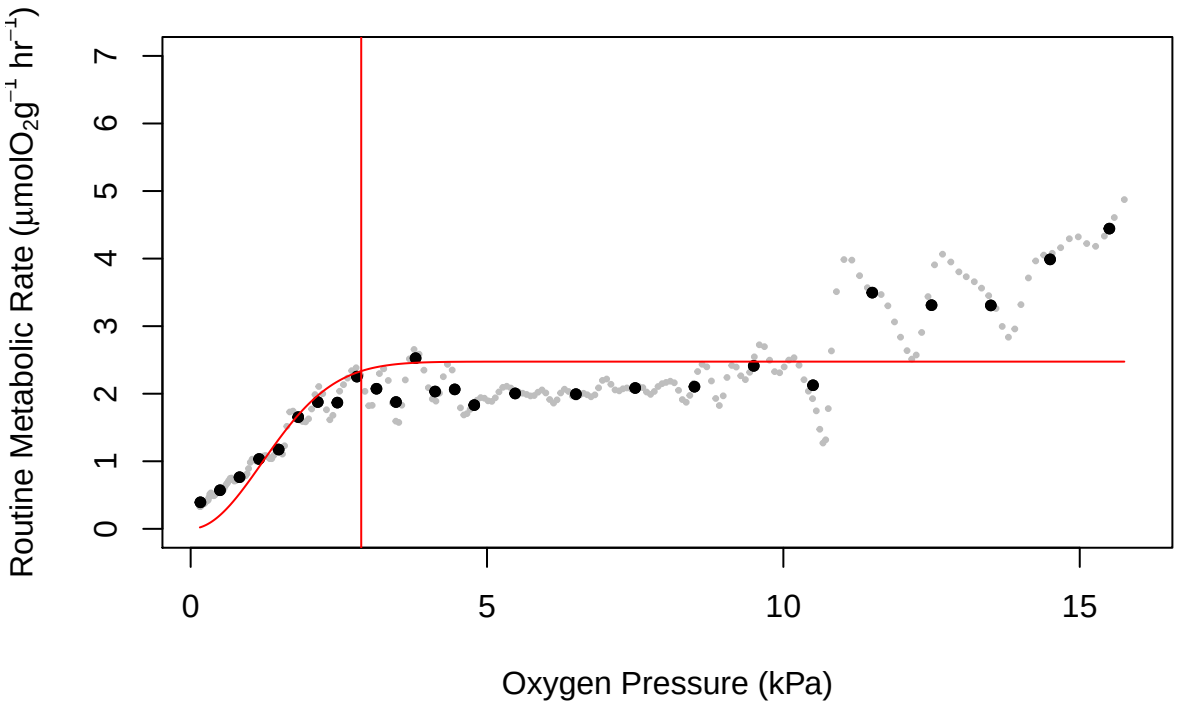
```
pcfitL4=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=pcritL4.resp,start=list(Bm=1.5,Pc=5))
coef(pcfitL4)[2]
```

```
##       Pc
## 2.877803
```

```
kpa=pcritL4.resp$kPa
pcritL4.lm=lm(resp~kPa,data=pcritL4.resp)
pcritL4.seg=segmented(pcritL4.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                      control=seg.control(display=FALSE))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=pcritL4.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,7))
points(resp~bin,data=pcritL4.bin,pch=20)
points(resp~bin,data=pcritL4.bin,pch=20)
lines(pcritL4.resp$kPa[complete.cases(pcritL4.resp$resp)], predict(pcfitL4),col="red")
abline(v=coef(pcfitL4)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[12,]=c("L4",coef(pcfitL4)[2],pcritL4.seg$psi[2])
```

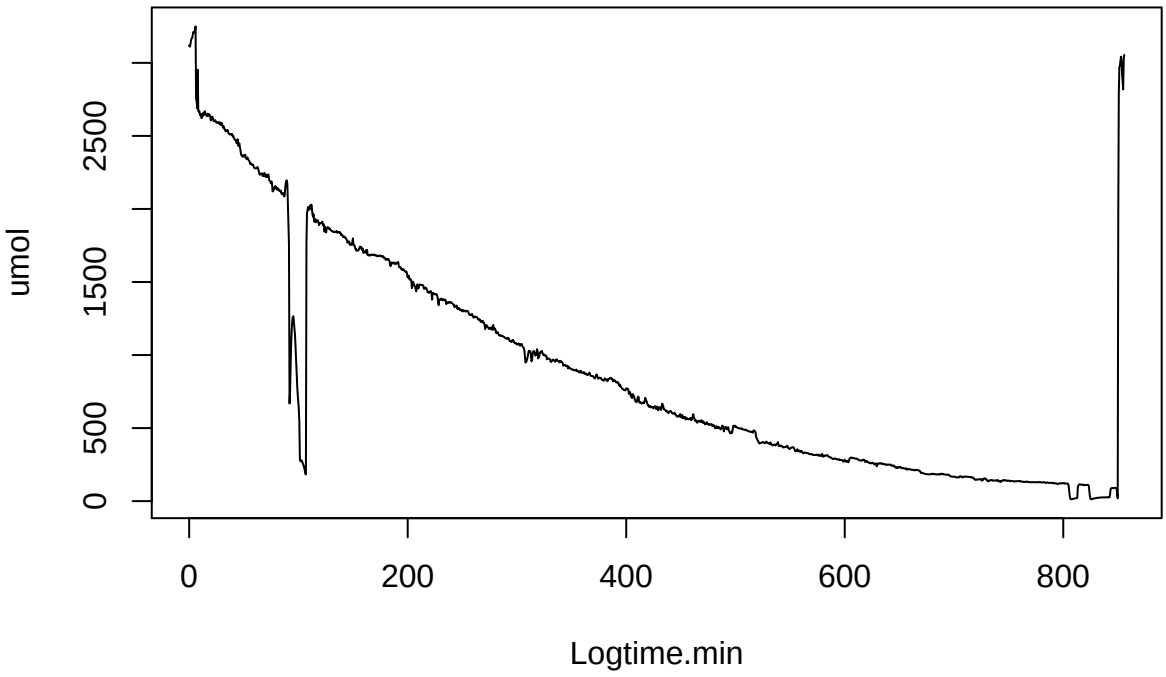
Octo Lloyd1

Reading in data

```
lloyd1=read.xlsx("pcrti1-2 data.xlsx",sheetIndex=1)
lloyd1=lloyd1[complete.cases(lloyd1$Logtime.min),]
```

Plot of oxygen over time

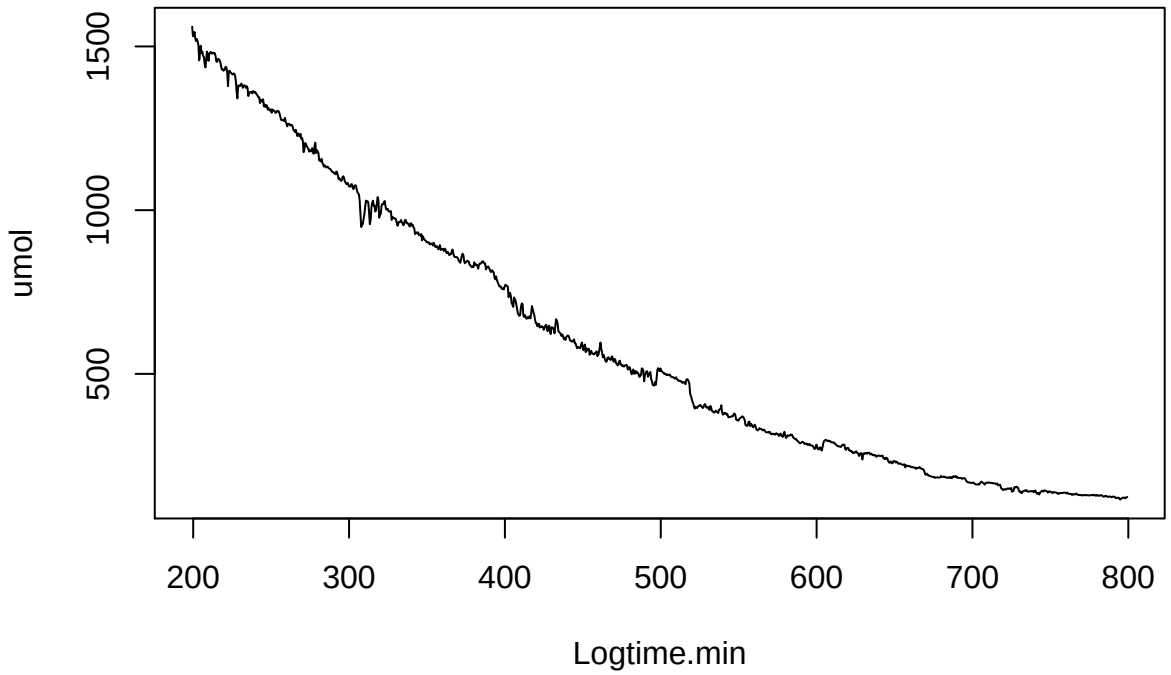
```
plot(umol~Logtime.min,data=lloyd1,type="l")
```



###

Subsetting Data and replotting

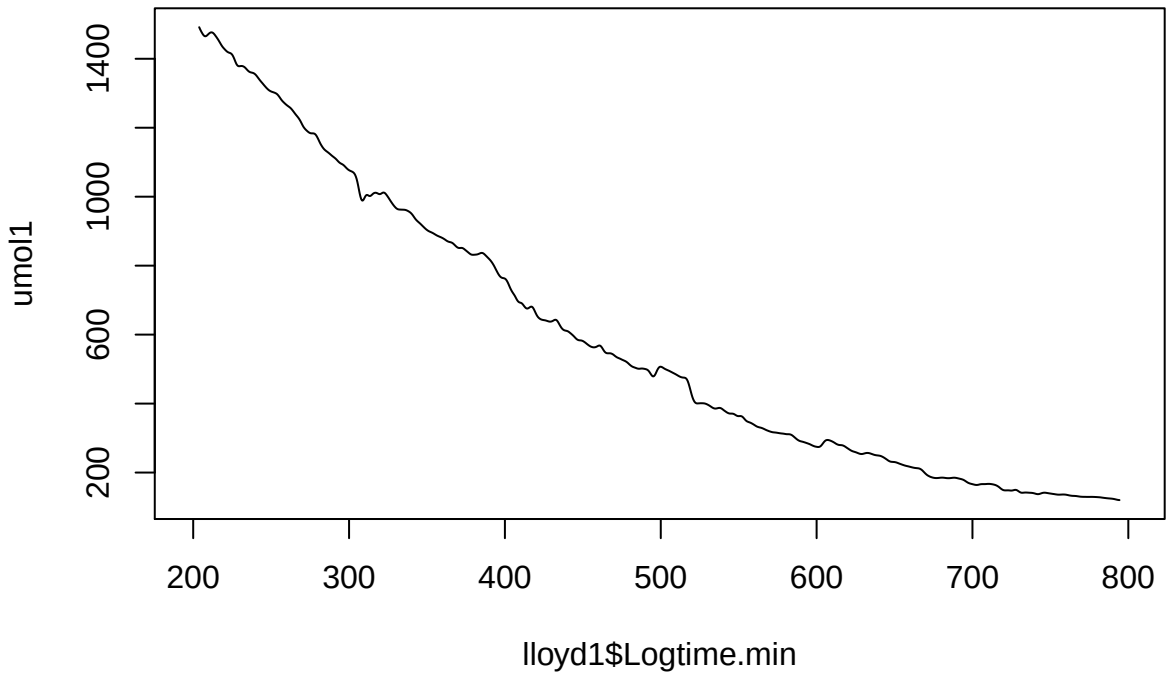
```
#lloyd1=lloyd1[c(20:175,200:(nrow(lloyd1)-110)),]
lloyd1=lloyd1[400:1600,]
plot(umol~Logtime.min,data=lloyd1,type="l")
```



###

Smoothing oxygen trace This is applying a 10-minute normally distributed weighted smoothing to the data. This is the same smoothing that was done on all previous data. In those cases it was done in the closed.resp() function.

```
umol1=lloyd1$umol
coeff = dnorm(seq(from = -3, to = 3, length.out = 20))/sum(dnorm(seq(from = -3, to = 3, length.out = 20)
umol1=filter(umol1, coeff, sides = 2)
plot(lloyd1$Logtime.min,umol1,type="l")
```



Calculating respiration rate

```
lloyd1.resp=data.frame(
  time=lloyd1$Logtime.min[2:nrow(lloyd1)]*60,
  umolO2.per.L=lloyd1$umol[2:nrow(lloyd1)]/10,
  kPa=lloyd1$kPa[2:nrow(lloyd1)],
  resp=(umol1[1:(length(umol1)-1)]-umol1[2:length(umol1)])/
        171.3/
        ((lloyd1$Logtime.min[2:nrow(lloyd1)]-lloyd1$Logtime.min[1:(nrow(lloyd1)-1)])/60)
)
```

Calculating P_{CRIT} using Weibull function and broken stick

```
lloyd1.fit=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=lloyd1.resp,start=list(Bm=1.5,Pc=5))
coef(lloyd1.fit)[2]
```

```
##       Pc
## 5.913762
```

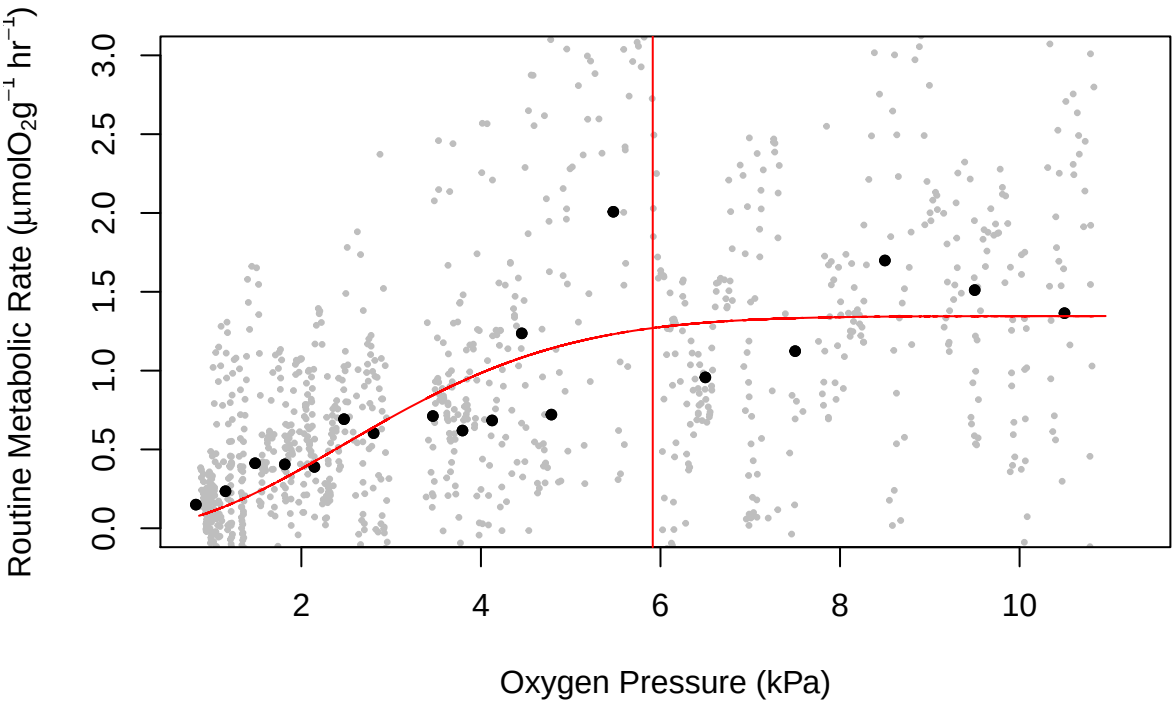
```
kpa=lloyd1.resp$kPa
lloyd1.lm=lm(resp~kPa,data=lloyd1.resp)
lloyd1.seg=segmented(lloyd1.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                     control=seg.control(display=FALSE))
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(lloyd1.resp$kPa)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
lloyd1.bin=aggregate(lloyd1.resp$resp~cut(lloyd1.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(lloyd1.bin)=c("bin","resp")
lloyd1.bin$bin=as.numeric(as.character(lloyd1.bin$bin))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=lloyd1.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,3))
points(resp~bin,data=lloyd1.bin,pch=20)
points(resp~bin,data=lloyd1.bin,pch=20)
lines(lloyd1.resp$kPa[complete.cases(lloyd1.resp$resp)], predict(lloyd1.fit),col="red")
abline(v=coef(lloyd1.fit)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[13,]=c("lloyd1",coef(lloyd1.fit)[2],lloyd1.seg$psi[2])
```

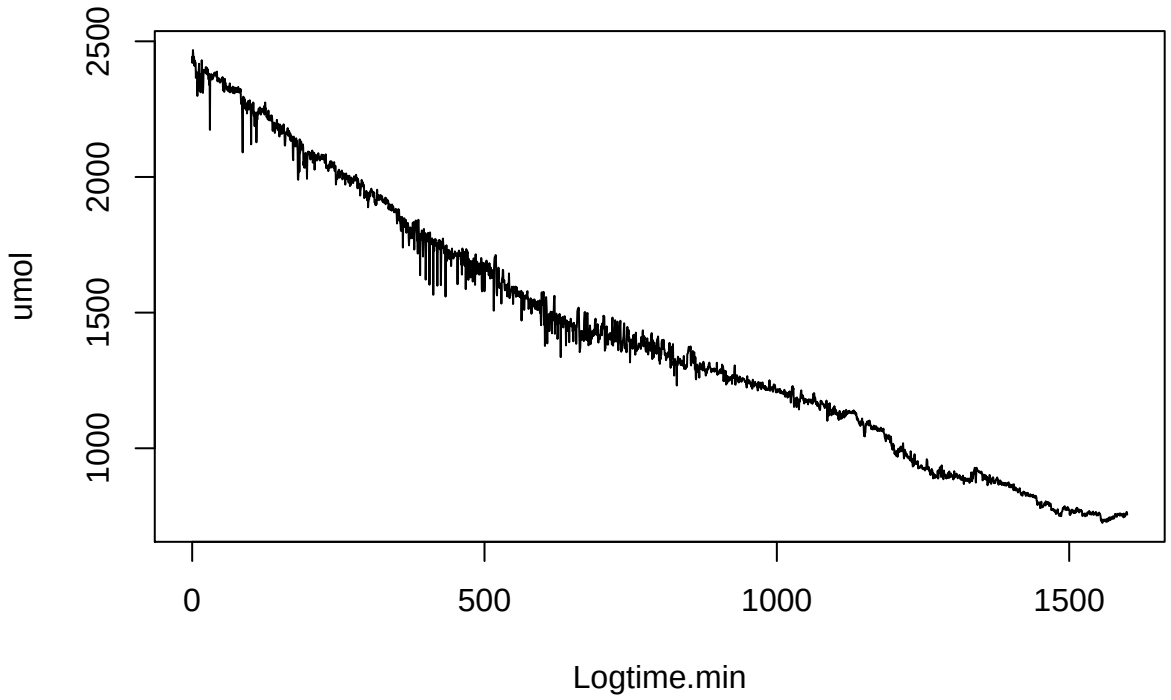
Octo lloyd2

Reading in data

```
lloyd2=read.xlsx("pcrti1-2 data.xlsx",sheetIndex=2)
lloyd2=lloyd2[complete.cases(lloyd2$Logtime.min),]
```

Plot of oxygen over time

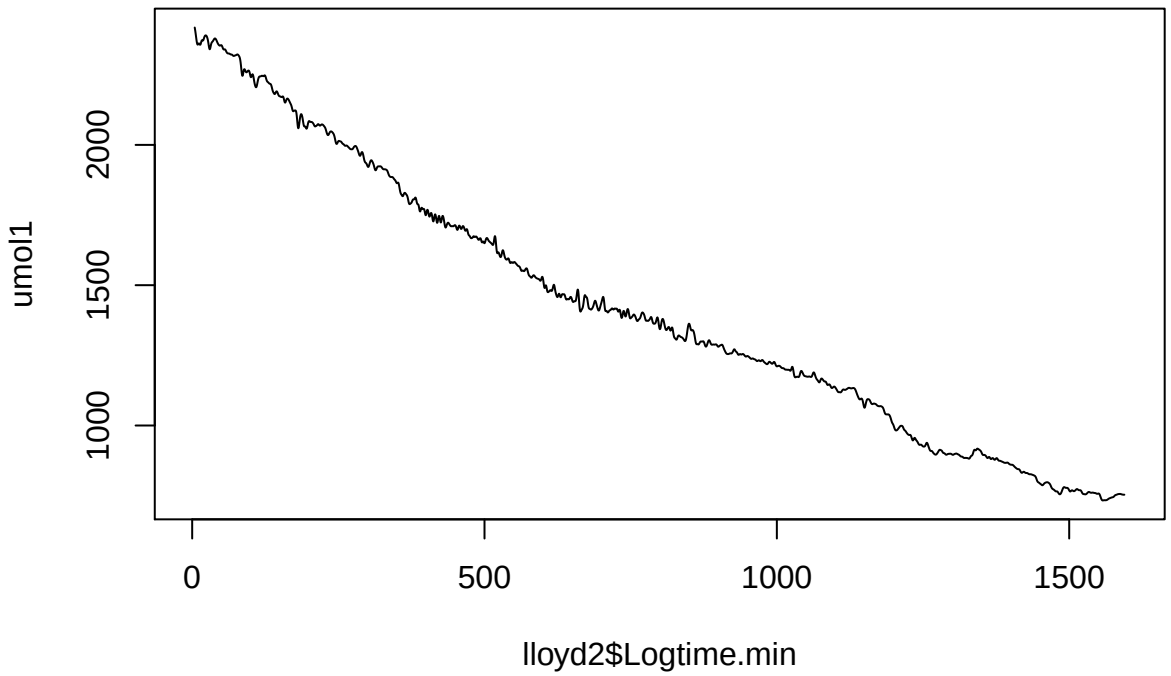
```
plot(umol~Logtime.min,data=lloyd2,type="l")
```



###

Smoothing oxygen trace This is applying a 10-minute normally distributed weighted smoothing to the data. This is the same smoothing that was done on all previous data. In those cases it was done in the closed.resp() function.

```
umol1=lloyd2$umol
coeff = dnorm(seq(from = -3, to = 3, length.out = 20))/sum(dnorm(seq(from = -3, to = 3, length.out = 20)
umol1=filter(umol1, coeff, sides = 2)
plot(lloyd2$Logtime.min,umol1,type="l")
```



Calculating respiration rate

```
lloyd2.resp=data.frame(
  time=lloyd2$Logtime.min[2:nrow(lloyd2)]*60,
  umolO2.per.L=lloyd2$umol[2:nrow(lloyd2)]/10,
  kPa=lloyd2$kPa[2:nrow(lloyd2)],
  resp=(umol1[1:(length(umol1)-1)]-umol1[2:length(umol1)])/
        83.6/
        ((lloyd2$Logtime.min[2:nrow(lloyd2)]-lloyd2$Logtime.min[1:(nrow(lloyd2)-1)])/60)
)
```

Calculating P_{CRIT} using Weibull function and broken stick

```
lloyd2.fit=nls(resp~Bm*(1-exp(-(kPa/(0.59*Pc))^2)),data=lloyd2.resp,start=list(Bm=1.5,Pc=5))
coef(lloyd2.fit)[2]
```

```
##       Pc
## 3.297945
```

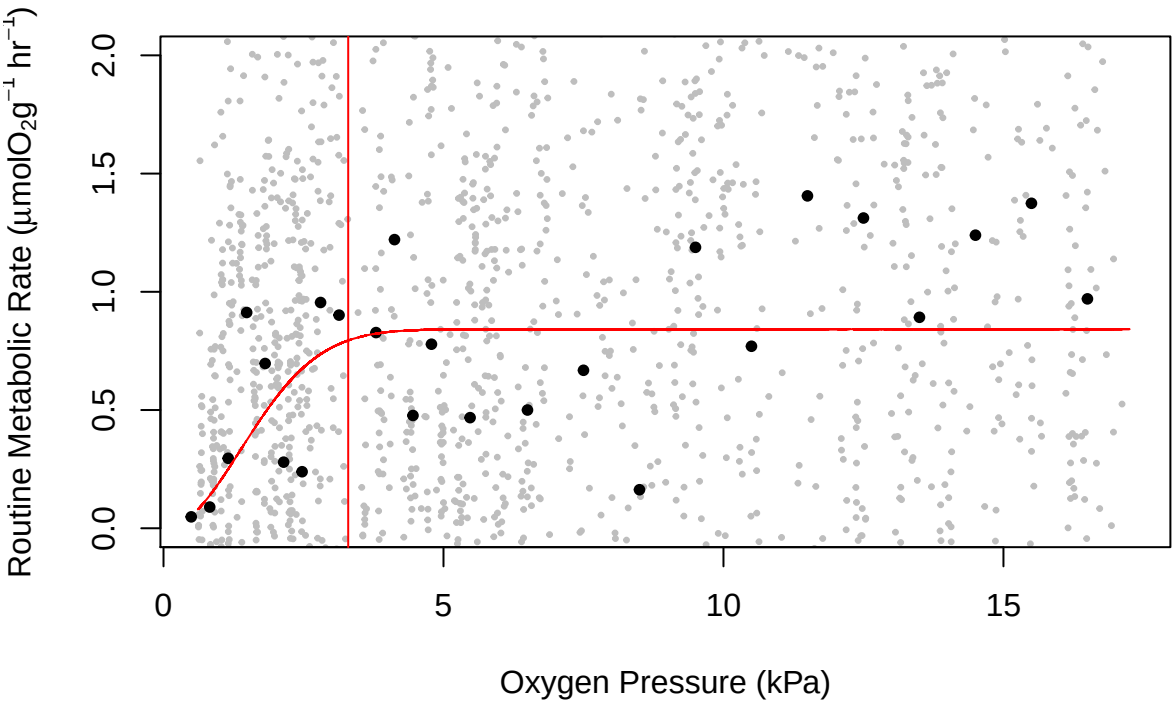
```
kpa=lloyd2.resp$kPa
lloyd2.lm=lm(resp~kPa,data=lloyd2.resp)
lloyd2.seg=segmented(lloyd2.lm,seg.Z=~kpa,psi=list(kpa=c(5)),
                     control=seg.control(display=FALSE))
```

Binning respiration rates by oxygen pressure

```
breaks=c(seq(from=0,to=5,by=0.33),seq(from=6,to=max(lloyd2.resp$kPa)+1,by=1))
tags=(breaks[1:(length(breaks)-1)]+breaks[2:length(breaks)])/2
lloyd2.bin=aggregate(lloyd2.resp$resp~cut(lloyd2.resp$kPa,breaks=breaks,labels=tags),FUN=c("mean"))
colnames(lloyd2.bin)=c("bin","resp")
lloyd2.bin$bin=as.numeric(as.character(lloyd2.bin$bin))
```

Plotting P_{CRIT} estimates and Weibull function regression

```
plot(resp~kPa,data=lloyd2.resp,pch=20,col="grey",cex=0.5,xlab="Oxygen Pressure (kPa)",
    ylab=expression("Routine Metabolic Rate ("*mu*"molO"[2]*"g"^-1*" hr"^-1*")"),ylim=c(0,2))
points(resp~bin,data=lloyd2.bin,pch=20)
lines(lloyd2.resp$kPa[complete.cases(lloyd2.resp$resp)], predict(lloyd2.fit),col="red")
abline(v=coef(lloyd2.fit)[2],col="red")
```



Adding P_{CRIT} estimates to table

```
pcrit.table[14,]=c("lloyd2",coef(lloyd2.fit)[2],lloyd2.seg$psi[2])
```

Null hypothesis significance testing

Formatting table

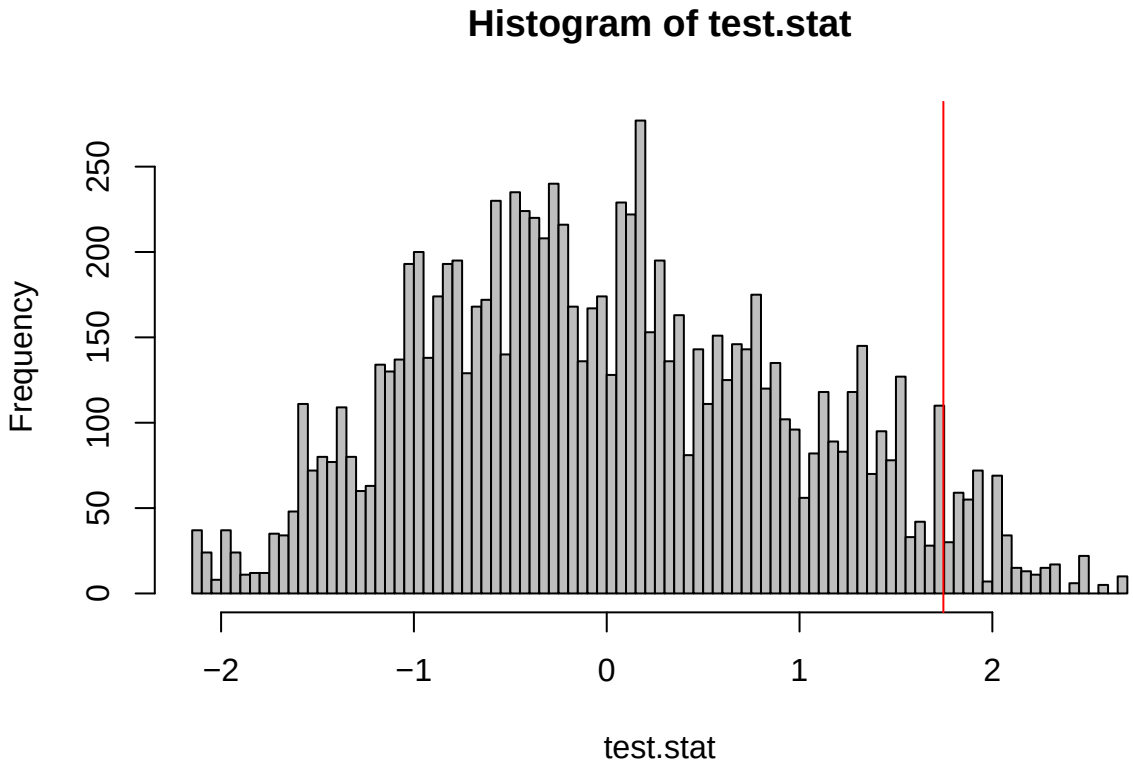
```
pcrit.table$Pc=as.numeric(pcrit.table$Pc)
pcrit.table$pco2=c(1500,700,1500,700,700,700,1500,1500,rep(700,6))
```

One-tailed permutation t-test of P_{CRIT} values

I am using a permutation test here because there are only 4 values at 1500 uatm

```
pcrit.table$set=1
pcrit.table$set[pcrit.table$pco2==1500]=2
set.seed(56)
obs.stat=
  mean(pcrit.table$Pc[pcrit.table$pco2==1500])-
  mean(pcrit.table$Pc[pcrit.table$pco2==700])

test.stat=0

for (i in 1:10000) {
  temp=pcrit.table
  temp$pco2=sample(temp$pco2)
  test.stat[i]=mean(temp$Pc[temp$pco2==1500])-mean(temp$Pc[temp$pco2==700])
}
```

```r
hist(test.stat,breaks=100,col="grey")
abline(v=obs.stat,col="red")
```




```r
test.p=sum(test.stat>=obs.stat)/length(test.stat)
test.p
```

```
## [1] 0.0445
```

Writing P_{CRIT} data out to .csv

```r
write.csv(pcrit.table[c(1,2,4)],"Pcrit_2014.csv",row.names = F)
```