

warmXtrophic Project: CN Analyses

Kara Dobson

July 15, 2021

Load in packages & data

```
# Clear all existing data
rm(list = ls())

# Load packages
library(tidyverse)
library(bbmle)
library(lme4)
library(fitdistrplus)

# Set working directory
L1_dir <- Sys.getenv("L1DIR")

# Read in data
cn <- read.csv(file.path(L1_dir, "CN/CN_L1.csv"))

# check species in dataframe
with(cn, table(cn$site, cn$species)) # keep sites in same dataframe for now bc unique spp at each site

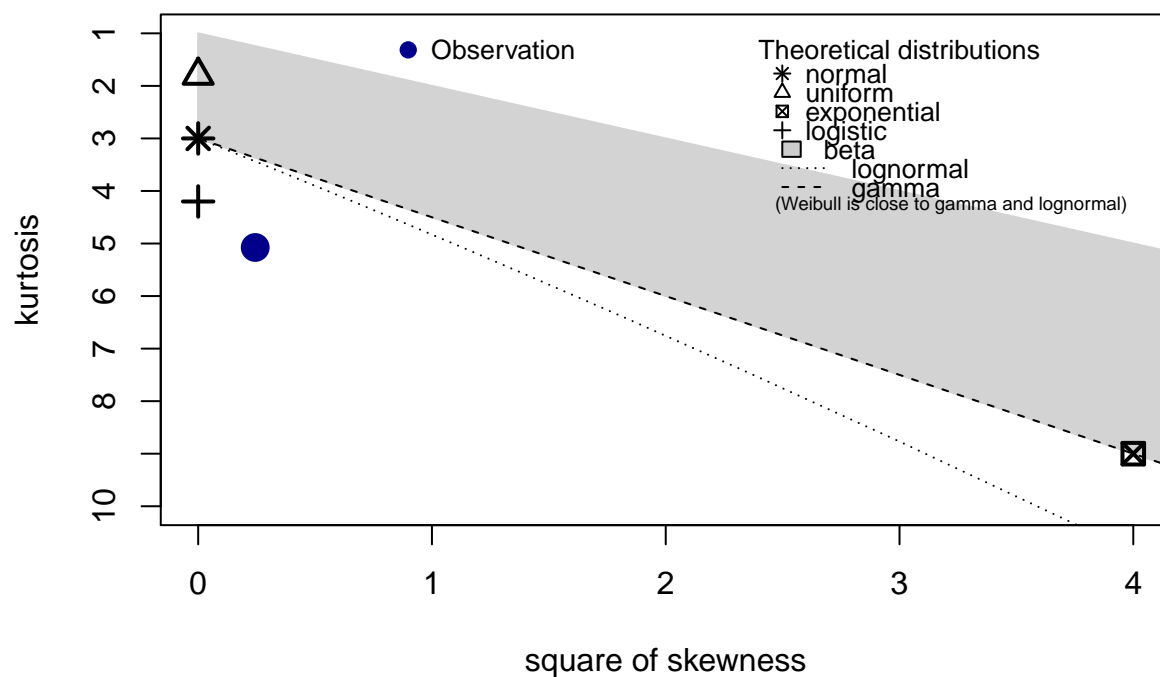
##
##          Cest Popr Soca
## kbs      0    0   73
## umbs    69   58    0

# removing NAs - two rows have NAs for both N and C, so this takes care of both
cn <- cn[!is.na(cn$carbon), ]
```

Data exploration: Carbon

```
descdist(cn$carbon, discrete = FALSE)
```

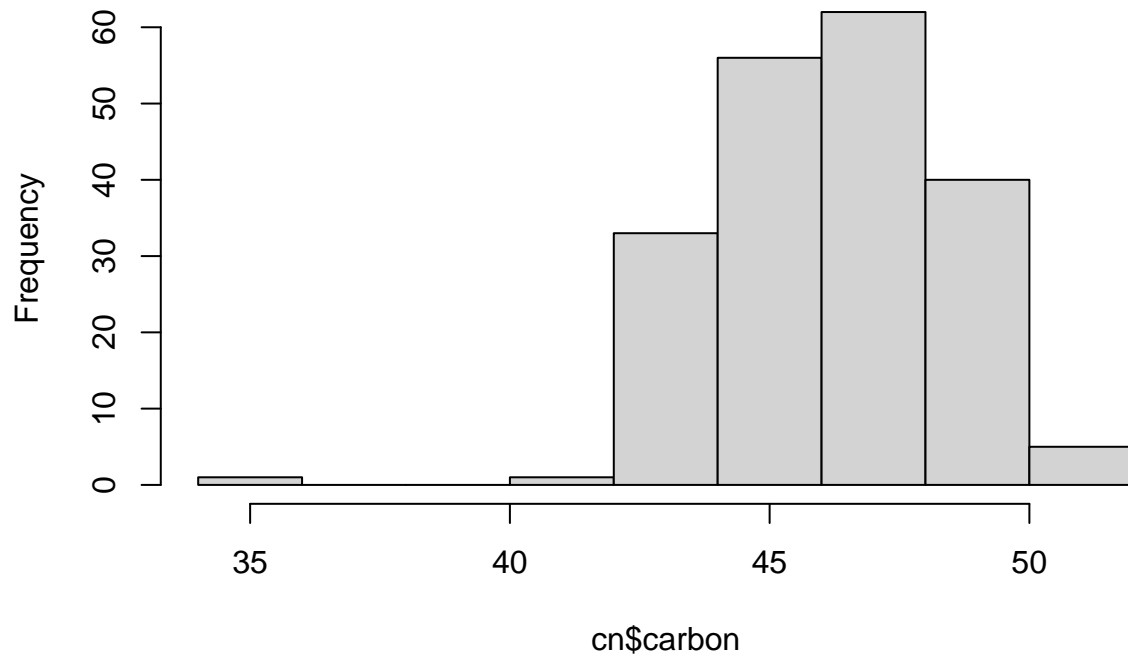
Cullen and Frey graph



```
## summary statistics
## -----
## min: 34.6899 max: 51.7902
## median: 46.3907
## mean: 46.2019
## estimated sd: 2.266163
## estimated skewness: -0.4940101
## estimated kurtosis: 5.075245
```

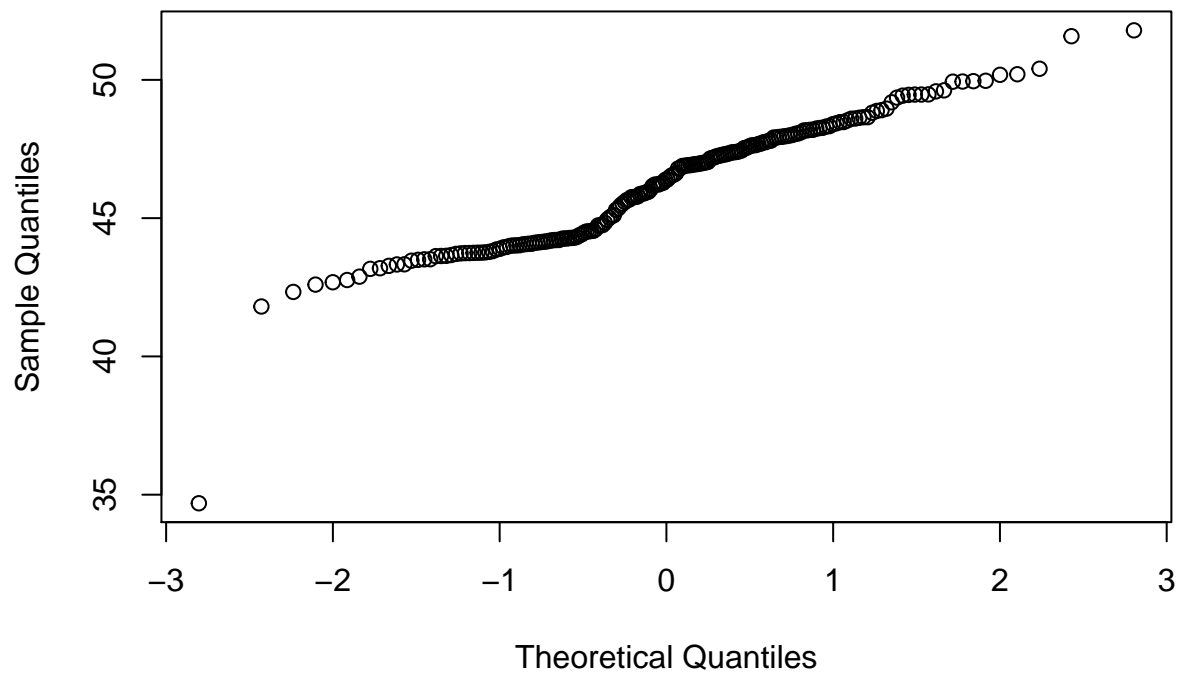
```
hist(cn$carbon)
```

Histogram of cn\$carbon



```
qqnorm(cn$carbon)
```

Normal Q-Q Plot

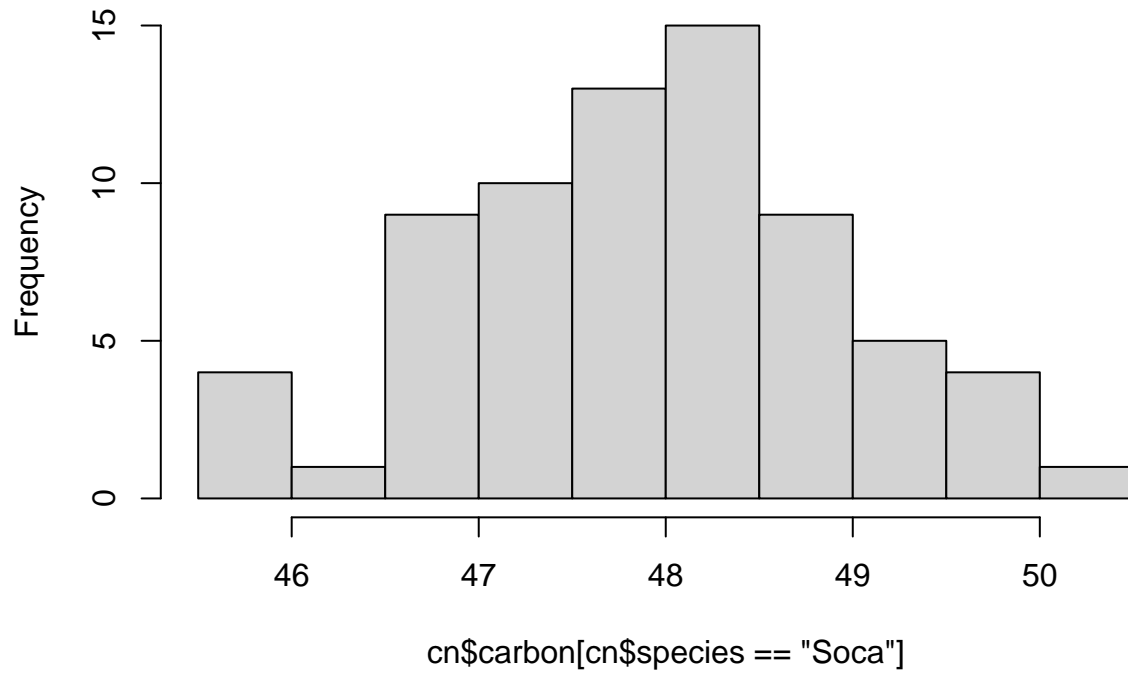


```
shapiro.test(cn$carbon)
```

```
##  
## Shapiro-Wilk normality test
```

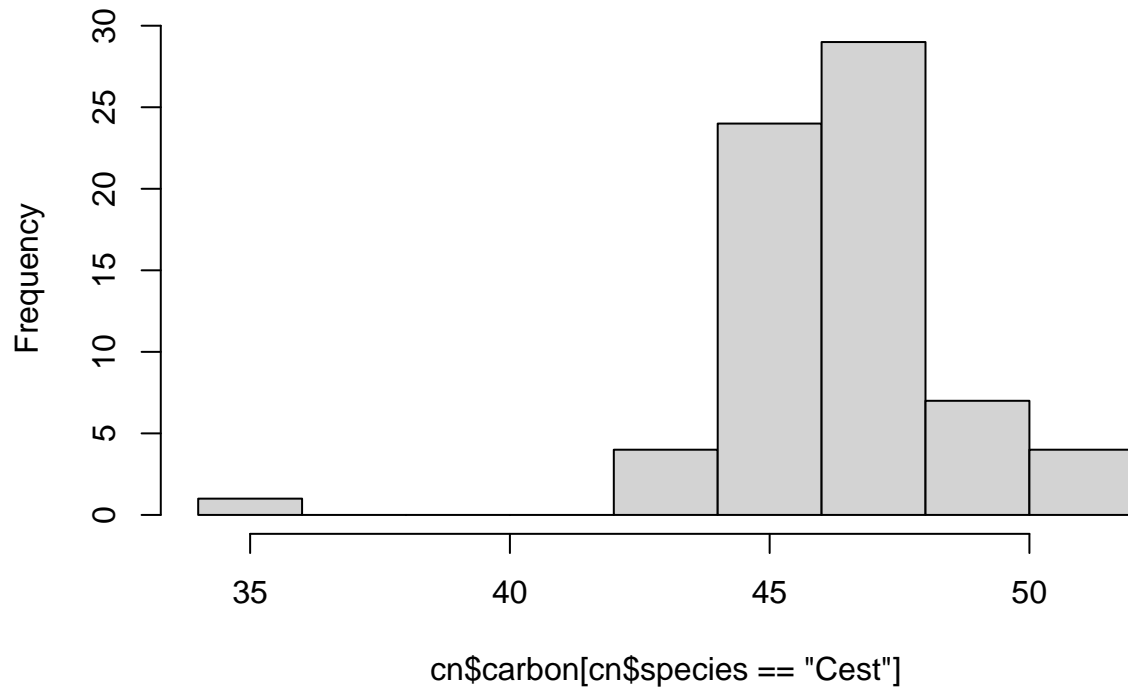
```
##  
## data:  cn$carbon  
## W = 0.9467, p-value = 1.028e-06  
# looking at each species separately  
hist(cn$carbon[cn$species == "Soca"])
```

Histogram of cn\$carbon[cn\$species == "Soca"]



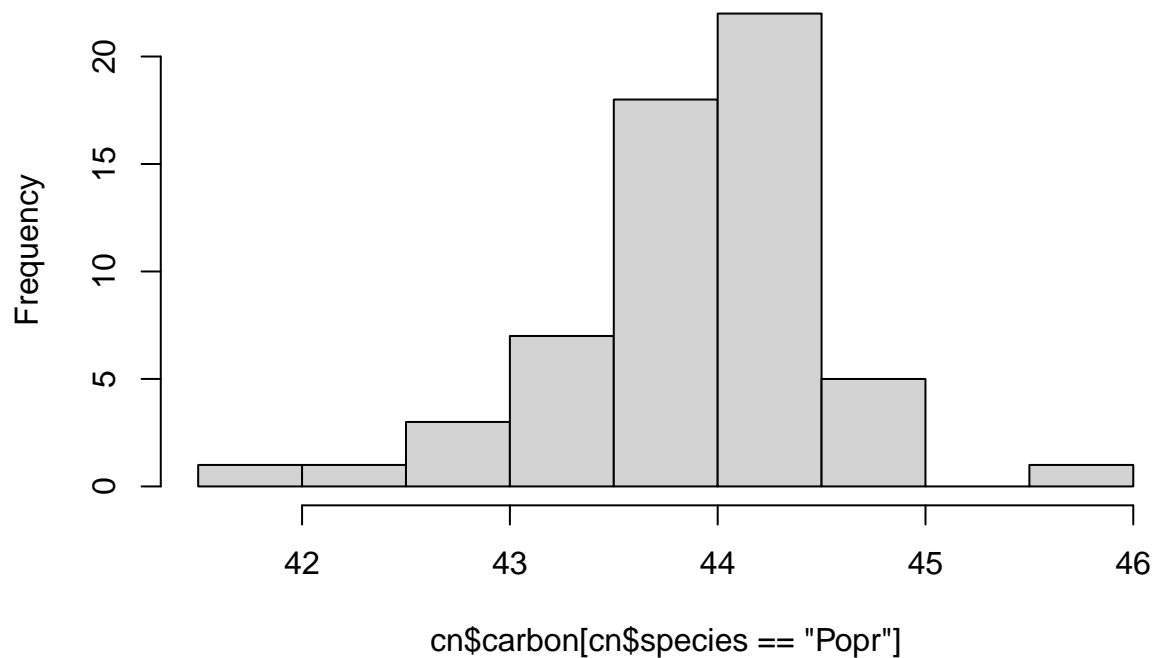
```
hist(cn$carbon[cn$species == "Cest"])
```

Histogram of `cn$carbon[cn$species == "Cest"]`



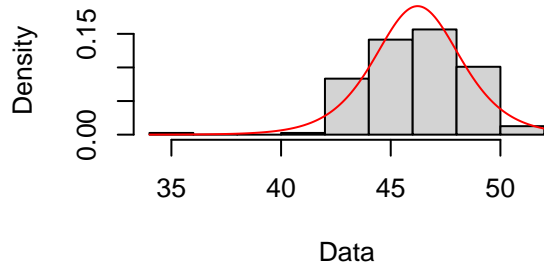
```
hist(cn$carbon[cn$species == "Popr"]) # skewness looks species specific
```

Histogram of `cn$carbon[cn$species == "Popr"]`

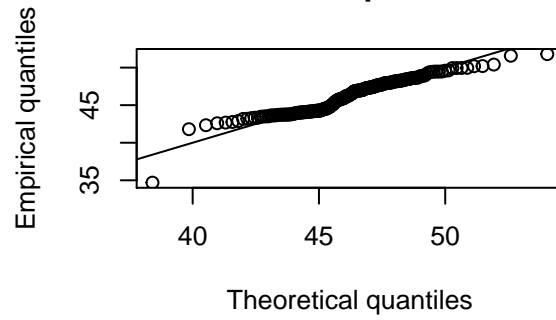


```
# logistic distribution?  
c.fit.logis <- fitdist(cn$carbon, "logis")  
plot(c.fit.logis)
```

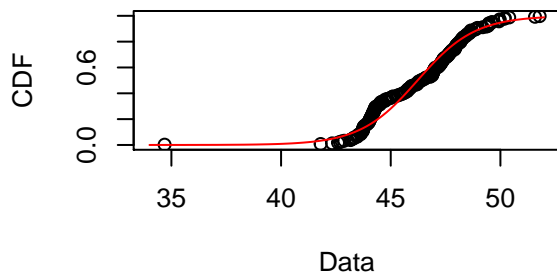
Empirical and theoretical dens.



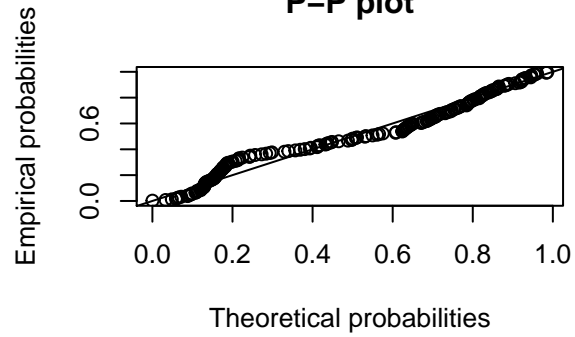
Q-Q plot



Empirical and theoretical CDFs

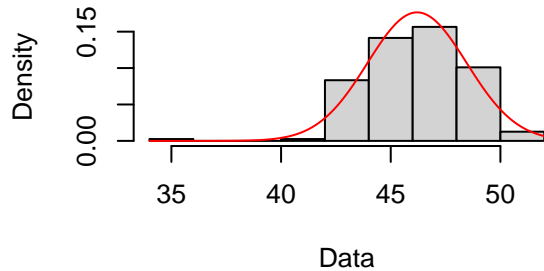


P-P plot

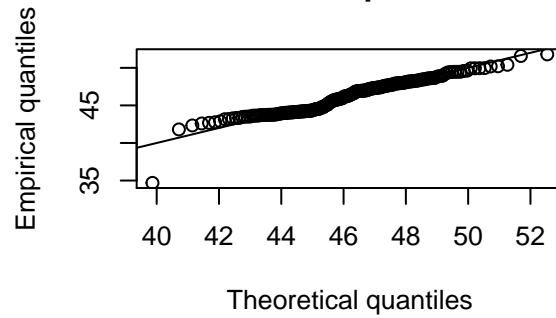


```
# normal distribution?  
c.fit.norm <- fitdlist(cn$carbon, "norm")  
plot(c.fit.norm)
```

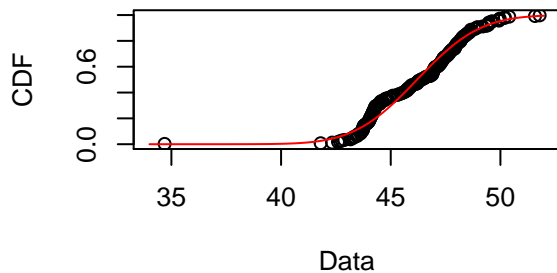
Empirical and theoretical dens.



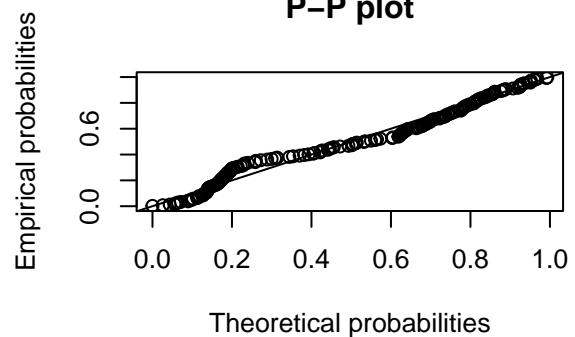
Q-Q plot



Empirical and theoretical CDFs



P-P plot

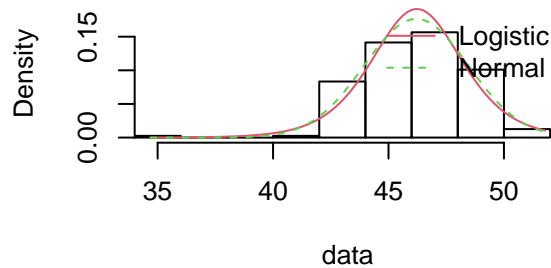


```

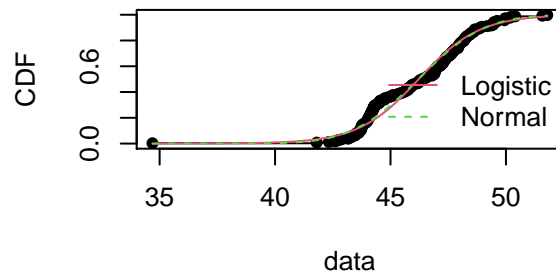
par(mfrow = c(2, 2))
plot.legend <- c("Logistic", "Normal")
denscomp(list(c.fit.logis, c.fit.norm), legendtext = plot.legend)
cdfcomp(list(c.fit.logis, c.fit.norm), legendtext = plot.legend)
qqcomp(list(c.fit.logis, c.fit.norm), legendtext = plot.legend)
ppcomp(list(c.fit.logis, c.fit.norm), legendtext = plot.legend)

```

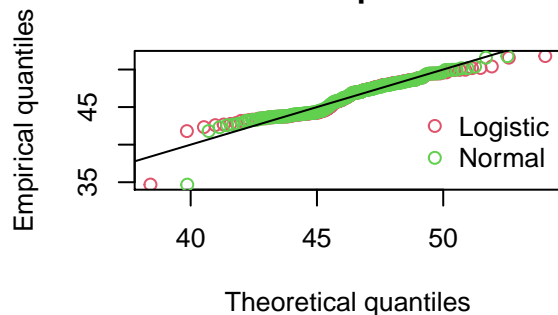
Histogram and theoretical densities



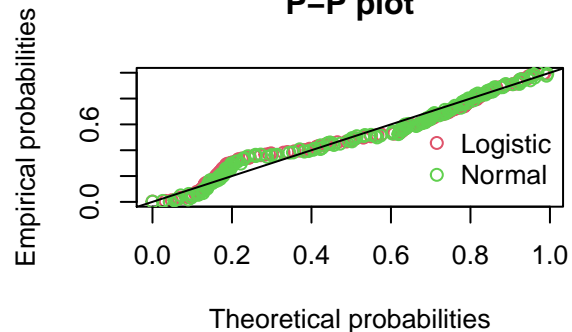
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```

gofstat(list(c.fit.logis, c.fit.norm), fitnames = c("Logistic", "Normal"))

```

```

## Goodness-of-fit statistics
##                               Logistic   Normal
## Kolmogorov-Smirnov statistic 0.1169881 0.1020134
## Cramer-von Mises statistic   0.4982147 0.3779116
## Anderson-Darling statistic   3.0473209 2.3546803
##
## Goodness-of-fit criteria
##                               Logistic   Normal
## Akaike's Information Criterion 891.4264 888.8601
## Bayesian Information Criterion 898.0030 895.4366
# Normal looks best

```

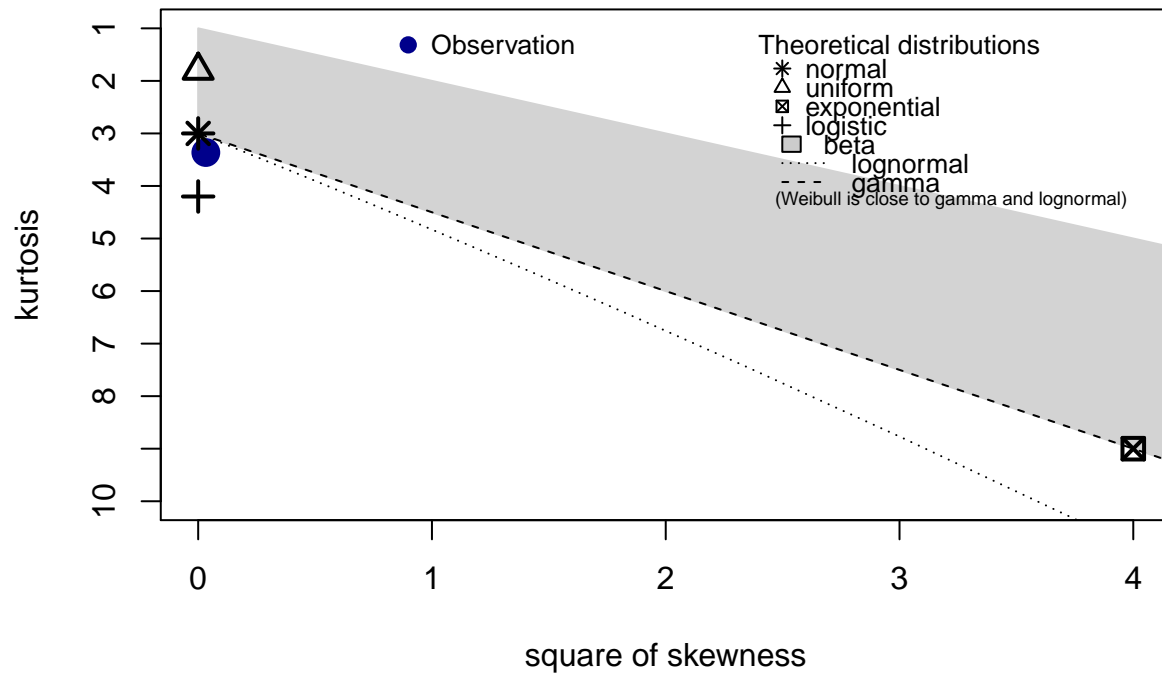
Data exploration: Nitrogen

```

descdist(cn$nitrogen, discrete = FALSE)

```

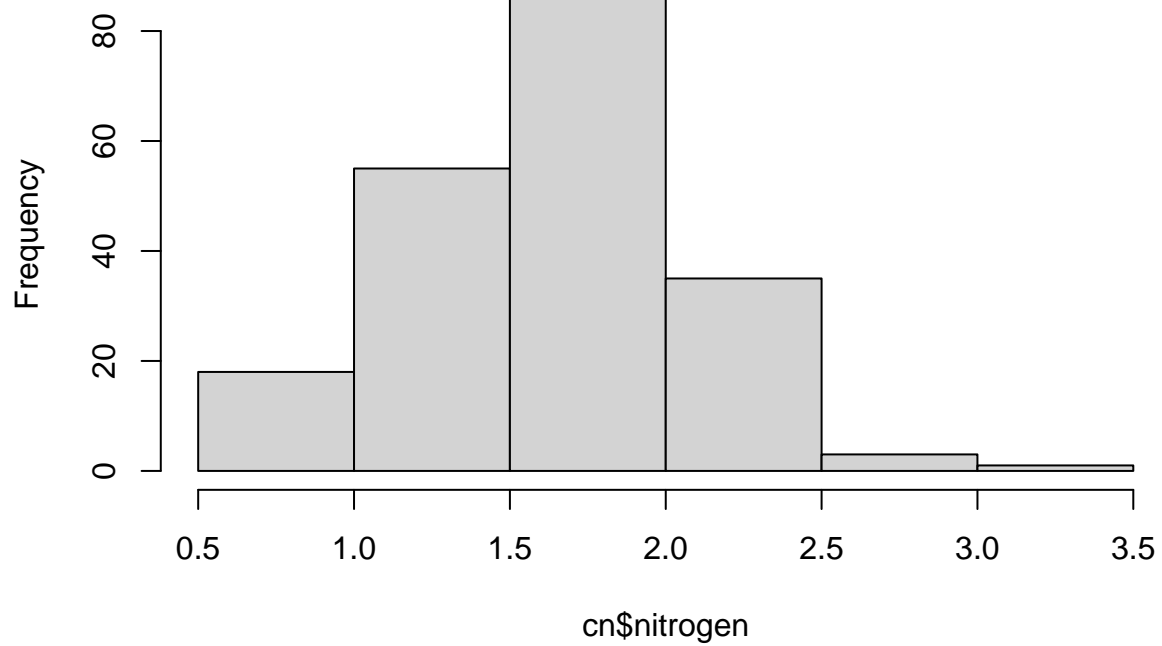
Cullen and Frey graph



```
## summary statistics
## -----
## min: 0.6773 max: 3.355
## median: 1.6766
## mean: 1.61771
## estimated sd: 0.4492647
## estimated skewness: 0.181173
## estimated kurtosis: 3.363488
```

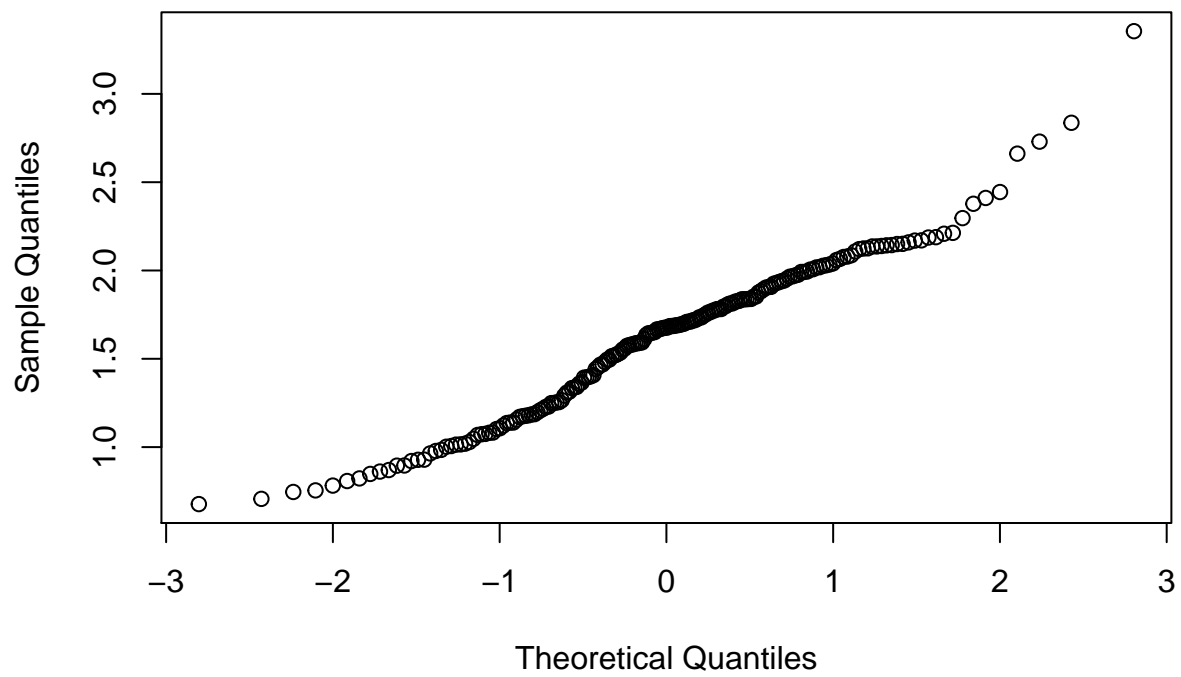
```
hist(cn$nitrogen)
```


Histogram of cn\$nitrogen



```
qqnorm(cn$nitrogen)
```

Normal Q-Q Plot

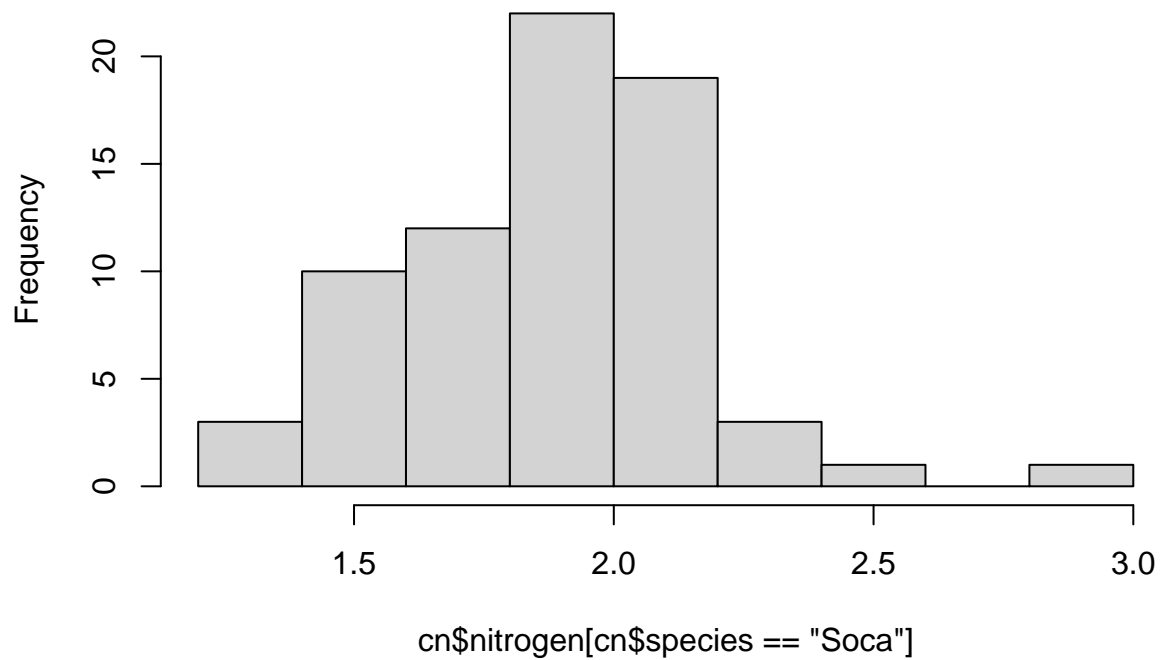


```
shapiro.test(cn$nitrogen)
```

```
##
```

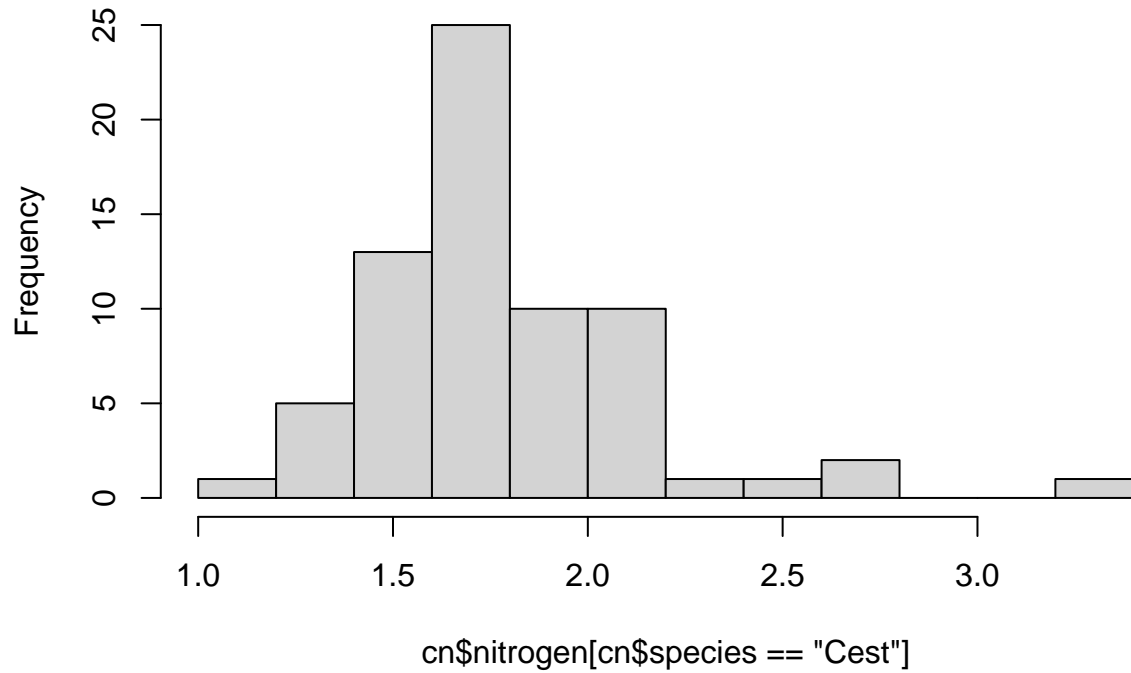
```
## Shapiro-Wilk normality test
##
## data: cn$nitrogen
## W = 0.97753, p-value = 0.002877
# looking at each species separately
hist(cn$nitrogen[cn$species == "Soca"])
```

Histogram of cn\$nitrogen[cn\$species == "Soca"]



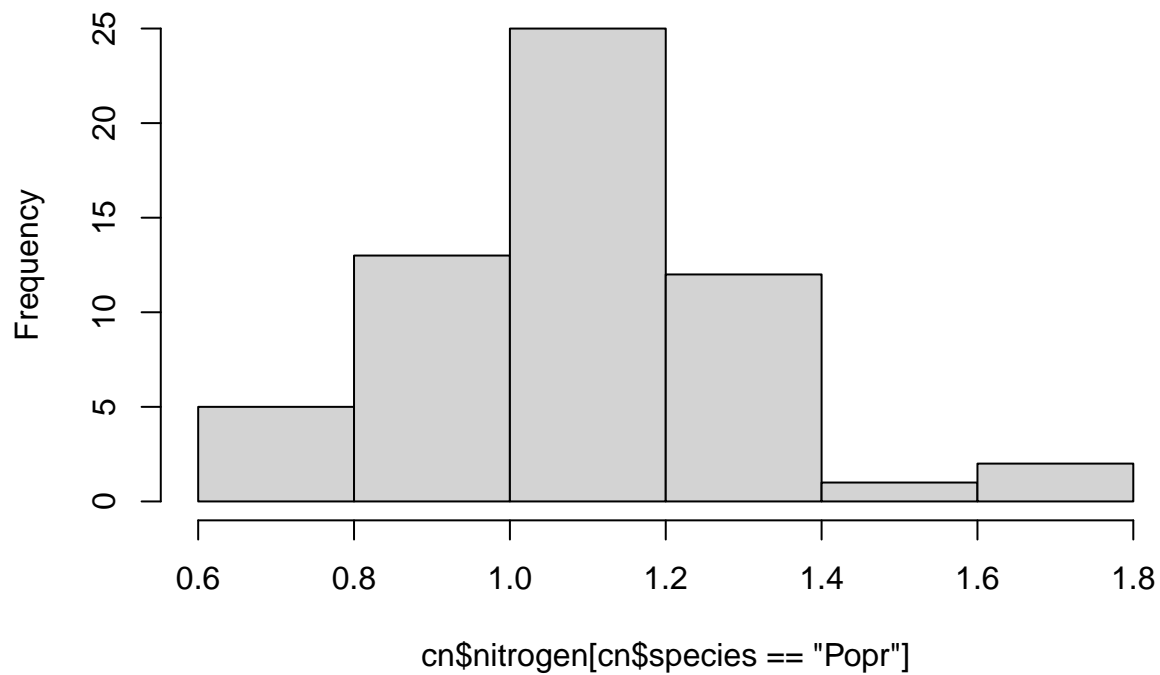
```
hist(cn$nitrogen[cn$species == "Cest"])
```

Histogram of `cn$nitrogen[cn$species == "Cest"]`



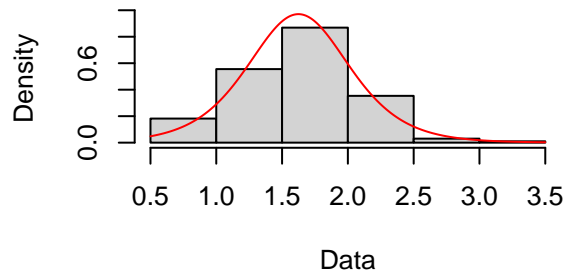
```
hist(cn$nitrogen[cn$species == "Popr"])
```

Histogram of `cn$nitrogen[cn$species == "Popr"]`

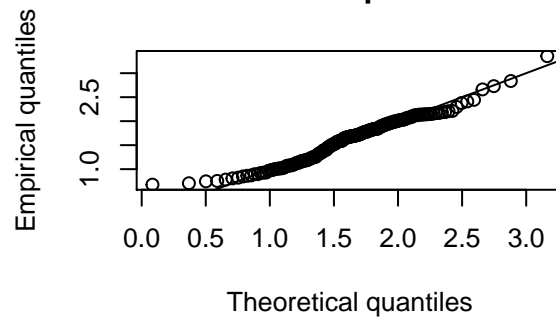


```
# logistic distribution?  
n.fit.logis <- fitdist(cn$nitrogen, "logis")  
plot(n.fit.logis)
```

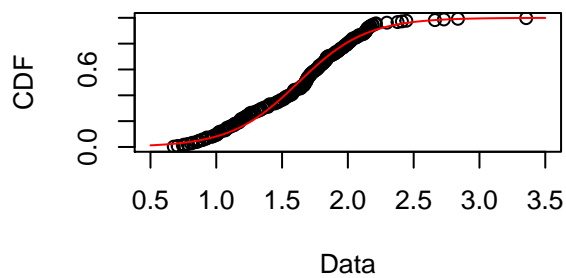
Empirical and theoretical dens.



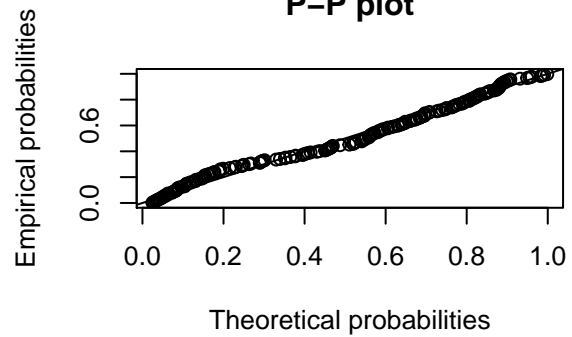
Q-Q plot



Empirical and theoretical CDFs

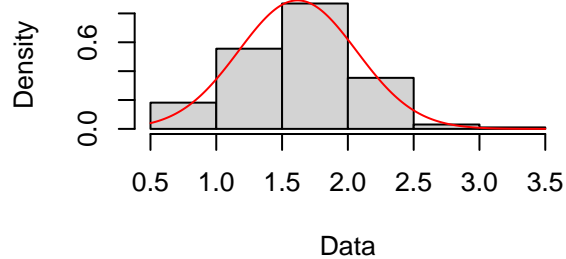


P-P plot

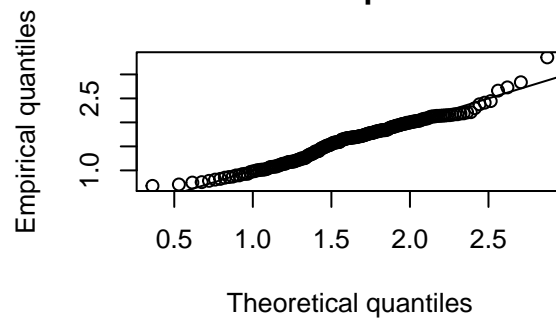


```
# normal distribution?  
n.fit.norm <- fitdlist(cn$nitrogen, "norm")  
plot(n.fit.norm)
```

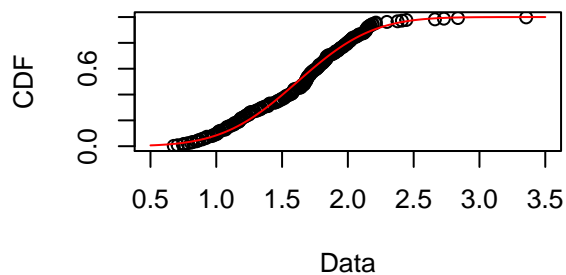
Empirical and theoretical dens.



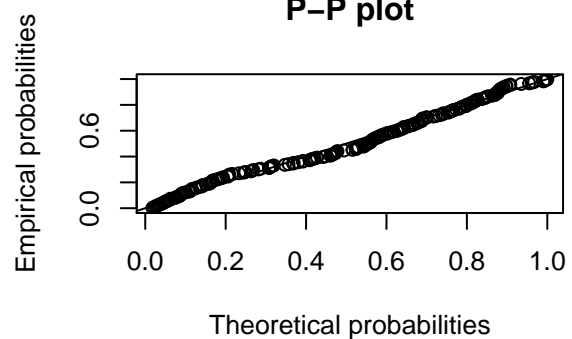
Q-Q plot



Empirical and theoretical CDFs

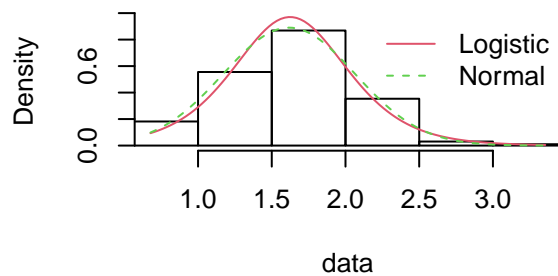


P-P plot

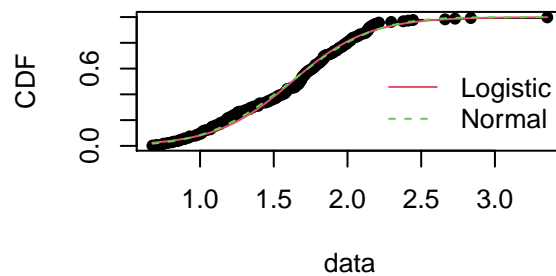


```
par(mfrow = c(2, 2))
plot.legend <- c("Logistic", "Normal")
denscomp(list(n.fit.logis, n.fit.norm), legendtext = plot.legend)
cdfcomp(list(n.fit.logis, n.fit.norm), legendtext = plot.legend)
qqcomp(list(n.fit.logis, n.fit.norm), legendtext = plot.legend)
ppcomp(list(n.fit.logis, n.fit.norm), legendtext = plot.legend)
```

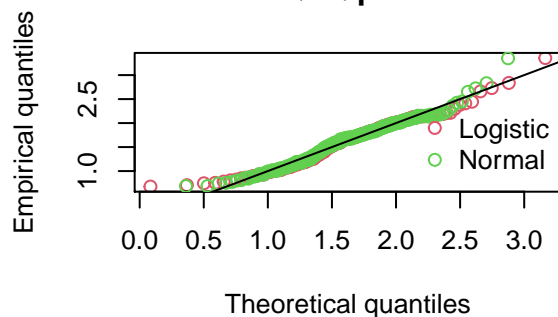
Histogram and theoretical densities



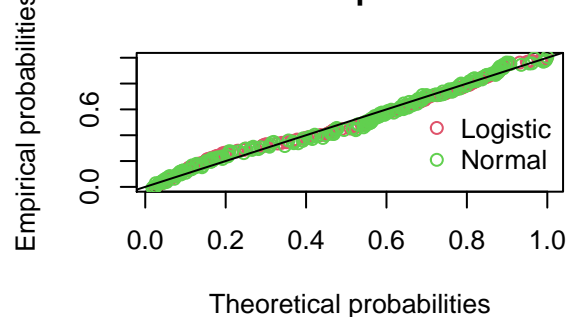
Empirical and theoretical CDFs



Q-Q plot



P-P plot



```
gofstat(list(n.fit.logis, n.fit.norm), fitnames = c("Logistic", "Normal"))
```

```
## Goodness-of-fit statistics
##                               Logistic    Normal
## Kolmogorov-Smirnov statistic 0.07015436 0.06920138
## Cramer-von Mises statistic   0.20825586 0.17291905
## Anderson-Darling statistic   1.40229004 1.10944863
##
## Goodness-of-fit criteria
##                               Logistic    Normal
## Akaike's Information Criterion 252.1265 248.0405
## Bayesian Information Criterion 258.7030 254.6170
```

```
# Normal looks best
```

Check assumptions

```
# check assumptions m1 <- lmer(carbon ~ state + insecticide + (1/plot), data =
# cn, REML=F)
```

updates end here - need to merge meta-data with cn data
code below is old

Old code

Model comparison: KBS Soca

```
# carbon m1a <- lm(carbon ~ state, data = cn_kbs) m1b <- lm(carbon ~ state +  
# insecticide, data = cn_kbs) m1c <- lmer(carbon ~ state + (1|plot), data =  
# cn_kbs) m1d <- lmer(carbon ~ state + insecticide + (1|plot), data = cn_kbs)  
# AICctab(m1a, m1b, m1c, m1d, weights = T) summary(m1a) summary(m1c) # plot as a  
# random effect nitrogen m2a <- lm(nitrogen ~ state, data = cn_kbs) m2b <-  
# lm(nitrogen ~ state + insecticide, data = cn_kbs) m2c <- lmer(nitrogen ~ state  
# + (1|plot), data = cn_kbs) m2d <- lmer(nitrogen ~ state + insecticide +  
# (1|plot), data = cn_kbs) AICctab(m2a, m2b, m2c, m2d, weights = T) summary(m2a)  
# summary(m2c) # plot as a random effect
```

Model comparison: UMBS Cest

```
# carbon m3a <- lm(carbon ~ state, data = cn_cest_umbs) m3b <- lm(carbon ~ state  
# + insecticide, data = cn_cest_umbs) m3c <- lmer(carbon ~ state + (1|plot), data  
# = cn_cest_umbs) m3d <- lmer(carbon ~ state + insecticide + (1|plot), data =  
# cn_cest_umbs) AICctab(m3a, m3b, m3c, m3d, weights = T) summary(m3a)  
# summary(m3c) # plot as a random effect nitrogen m4a <- lm(nitrogen ~ state,  
# data = cn_cest_umbs) m4b <- lm(nitrogen ~ state + insecticide, data =  
# cn_cest_umbs) m4c <- lmer(nitrogen ~ state + (1|plot), data = cn_cest_umbs) m4d  
# <- lmer(nitrogen ~ state + insecticide + (1|plot), data = cn_cest_umbs)  
# AICctab(m4a, m4b, m4c, m4d, weights = T) summary(m4a) summary(m4c) # plot as a  
# random effect
```

Model comparison: UMBS Popr

```
# carbon m5a <- lm(carbon ~ state, data = cn_popr_umbs) m5b <- lm(carbon ~ state  
# + insecticide, data = cn_popr_umbs) m5c <- lmer(carbon ~ state + (1|plot), data  
# = cn_popr_umbs) m5d <- lmer(carbon ~ state + insecticide + (1|plot), data =  
# cn_popr_umbs) AICctab(m5a, m5b, m5c, m5d, weights = T) summary(m5a)  
# summary(m5c) # plot as a random effect nitrogen m6a <- lm(nitrogen ~ state,  
# data = cn_popr_umbs) m6b <- lm(nitrogen ~ state + insecticide, data =  
# cn_popr_umbs) m6c <- lmer(nitrogen ~ state + (1|plot), data = cn_popr_umbs) m6d  
# <- lmer(nitrogen ~ state + insecticide + (1|plot), data = cn_popr_umbs)  
# AICctab(m6a, m6b, m6c, m6d, weights = T) summary(m6a) summary(m6c) # plot as a  
# random effect
```