

```

rm(list=ls())

data(ChickWeight)
?ChickWeight
summary(ChickWeight)
str(ChickWeight)

cw<-ChickWeight
class(cw$weight)
class(cw$Time)
class(cw$Chick)
class(cw$Diet)
cw$weight
cw$Time
cw$Chick
cw$Diet

w<-cw$weight
t<-as.integer(cw$Time)
h<-as.factor(cw$Chick)
hu<-factor(cw$Chick,ordered=F)
str(hu)
d<-cw$Diet
du<-factor(cw$Diet,ordered=F)

## linear
lm.ch<-lm(w~t+h+d, x=TRUE, y=TRUE)
summary(lm.ch)
dev.new()
par(mfrow=c(2,2))
plot(lm.ch)
require(lmtest)
resettest(lm.ch,power=2,type="fitted")
bptest(lm.ch)

require(np)
#system.time(npctest(w~t+h+d,model=lm.ch))
## np
#bw.ch<-npregbw(w~t+h+d)
system.time(bw.ch<-npregbw(w~t+h+d))
system.time(bw.ch<-npregbw(w~t+hu+d,bwmethod="cv.aic"))
summary(bw.ch)
?npregbw
np.ch<-npreg(bw.ch)
npplot(bw.ch)
npplot(bw.ch,gradients=T)
npplot(bw.ch,gradients=F,plot.errors.method = c("asymptotic"))
npplot(bw.ch,gradients=F,plot.errors.method = c("bootstrap"))
npplot(bw.ch,gradients=T,plot.errors.method = c("bootstrap"))
#npsigtest(bw.ch)

bw.ch1<-npregbw(w~t+h+d,regtype='ll')

```

```

npplot(bw.ch1)
npplot(bw.ch1,gradients=T)
npplot(bw.ch1,gradients=F,plot.errors.method = c("bootstrap"))
npplot(bw.ch1,gradients=T,plot.errors.method = c("bootstrap"))
#npsigtest(bw.ch1)

bw.chu<-npregbw(w~t+hu+d,regtype='ll')
np.chu<-npreg(bw.chu)
npplot(bw.chu)
npplot(bw.chu,gradients=T)
npplot(bw.chu,gradients=T,ylim=c(-10,10))
npplot(bw.chu,gradients=F,plot.errors.method = c("bootstrap"))
npplot(bw.chu,gradients=T,plot.errors.method = c("bootstrap"))
#npsigtest(bw.chu)

## Diet ##
nd1<-data.frame(t,hu,d)[3,]
fv1= predict(np.chu, newdata = nd1)
fv1
nd2<-nd1
str(nd2)
## Type of data !
str(nd2)
nd2[[3]]<-factor(4)
compare<-rbind(nd1,nd2)
rownames(compare)<-c("Dieta.1","Dieta.2")
compare

fv2= predict(np.chu, newdata = nd2)
fv2
imp<-c(fv1,fv2,fv2-fv1)
names(imp)<-c('Diet-1','Diet-4','Diff')
imp

```