UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Dipartimento di scienze economiche,
aziendali, matematiche e statistiche
“Bruno de Finetti”

Non parametric statistics

Introduction

Francesco Pauli

A.A. 2015/2016

Indice

Hubble constant

Hubble's law states that:

1. Objects observed in deep space (extragalactic space, > 10 megaparsecs (Mpc)) are found to have a Doppler shift interpretable as relative velocity away from Earth;
2. This Doppler-shift-measured velocity is approximately proportional to their distance from the Earth for galaxies up to a few hundred megaparsecs away.

It consitutes a fundamental piece of evidence in favour of the hypotheses of expansion of th euniverse.

In formulas, if

- x is the distance of a galaxy (Mpc, $1p = 3.09e13$ km)
- y is the doppler shift measured for the same galaxy (km/s)

Hubble's law states that

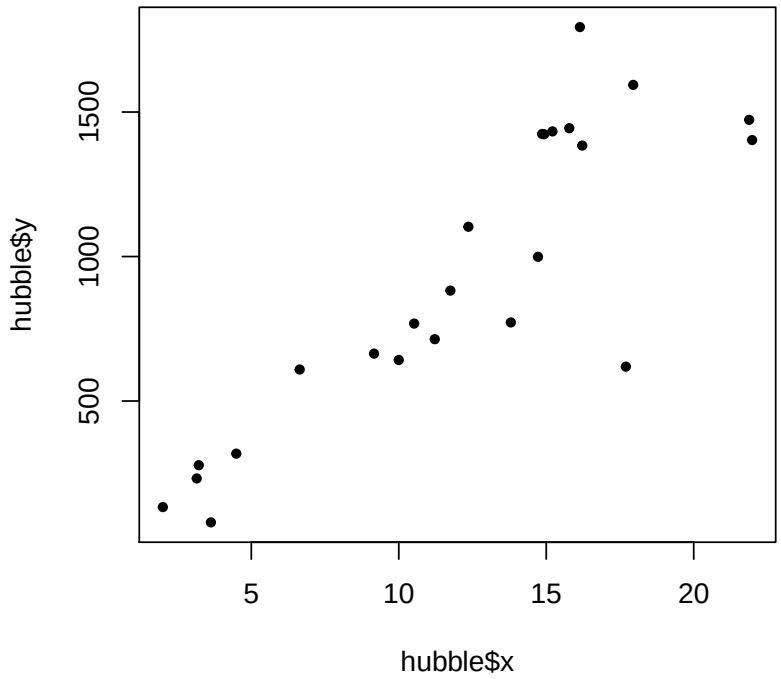
$$y = \beta x$$

Data

We consider data on distances and velocities of 24 galaxies containing Cepheid stars, from the Hubble space telescope key project to measure the Hubble constant.

```
library(gamair)
data(hubble)
head(hubble)
```

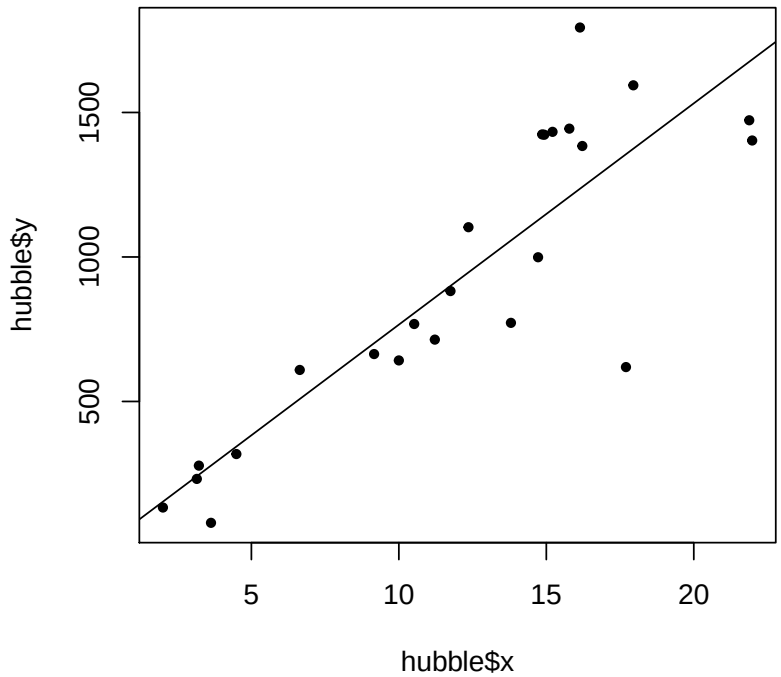
```
    Galaxy    y     x
1  NGC0300  133  2.00
2  NGC0925  664  9.16
3 NGC1326A 1794 16.14
4  NGC1365 1594 17.95
5  NGC1425 1473 21.88
6  NGC2403  278  3.22
```



Data

We consider data on distances and velocities of 24 galaxies containing Cepheid stars, from the Hubble space telescope key project to measure the Hubble constant.

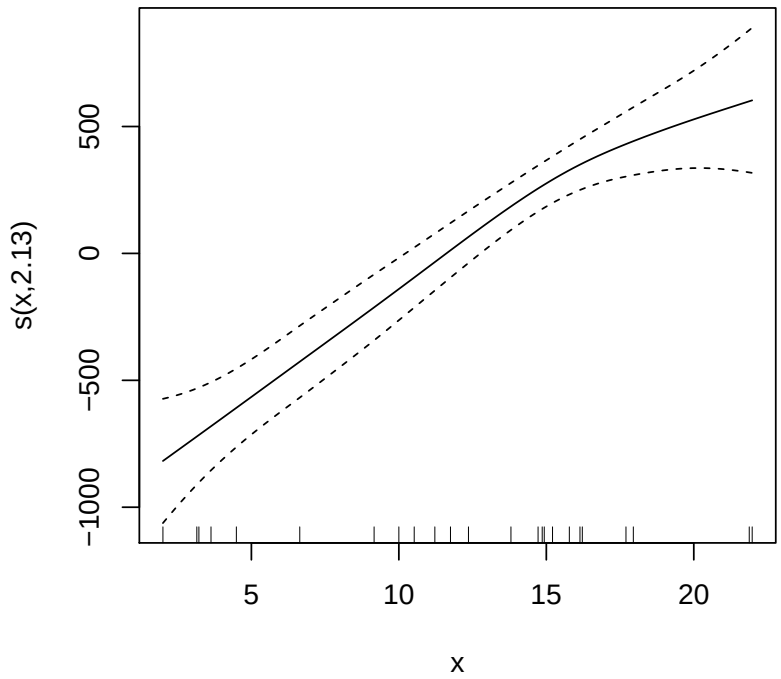
```
fit1=lm(y~x-1,data=hubble)
coef(fit1)

       x
76.58117

confint(fit1)

     2.5 %   97.5 %
x 68.37937 84.78297
```



A check on Hubble's law

We may use a semiparametric regression to check to what extent the data are compatible with the linearity of the relationship implied by Hubble's law.

```
fitGam=gam(y~s(x),data=hubble)
summary(fitGam)



Family: gaussian
Link function: identity

Formula:
y ~ s(x)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    924.4       51.9   17.81 4.25e-14

(Intercept) ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
       edf Ref.df     F  p-value
s(x) 2.132   2.63 26.94 2.89e-08 ***
---
Signif. codes:
```

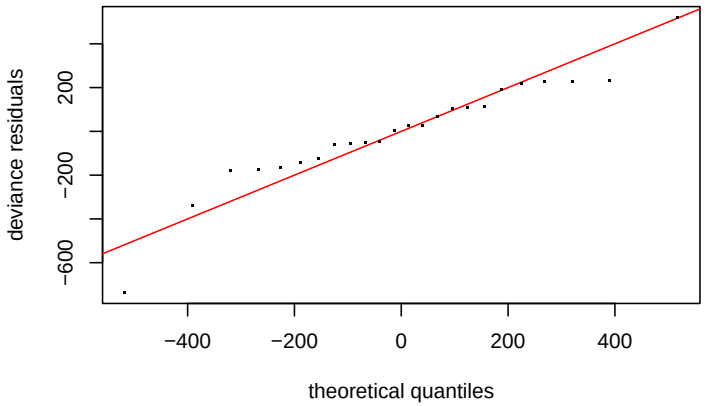


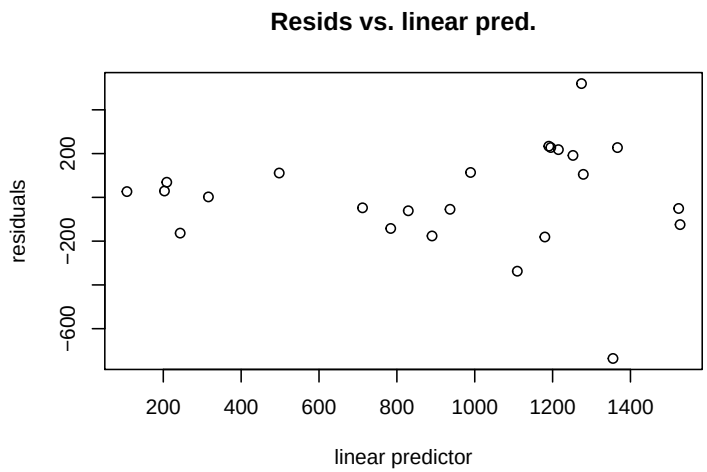
A check on Hubble's law

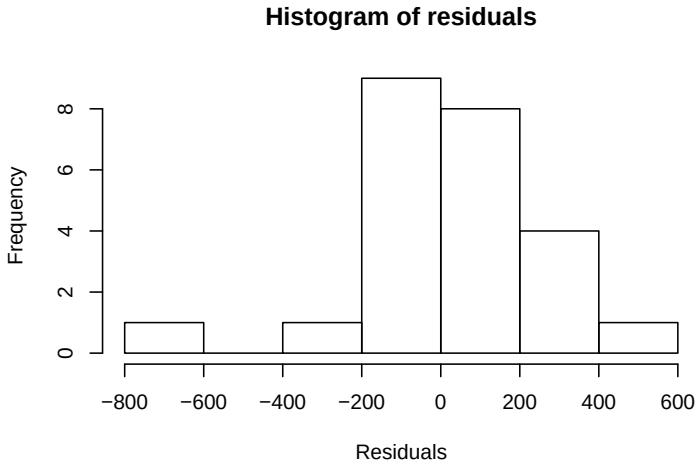
We now want to compare the two models

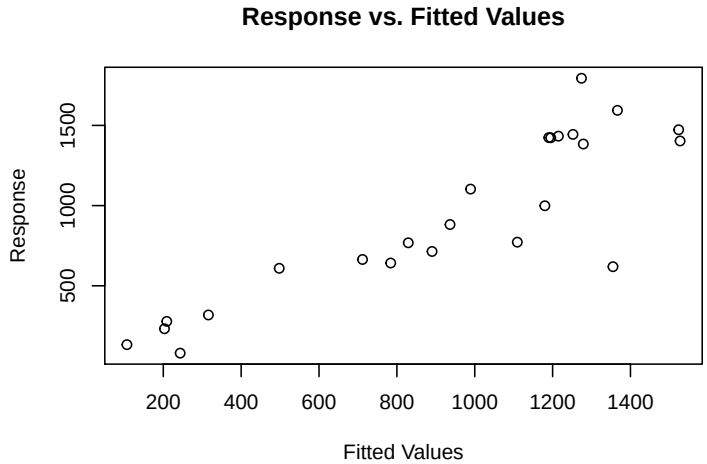
```
fitGamLin=gam(y~x,data=hubble)
AIC(fitGam,fitGamLin)


                df      AIC
fitGam    4.131733 338.8537
fitGamLin 3.000000 339.7999


anova(fitGamLin,fitGam,test="F")


Analysis of Deviance Table

Model 1: y ~ x
Model 2: y ~ s(x)
  Resid. Df Resid. Dev     Df Deviance      F
1    22.000    1541869
2    20.868    1348862 1.1317   193008 2.6385
  Pr(>F)
1
2 0.1162
```

A check of the model









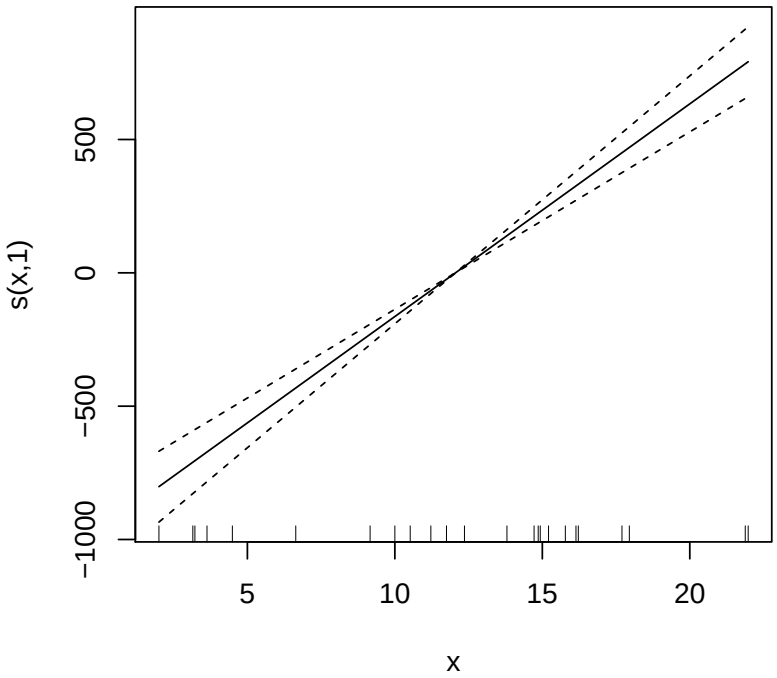
A check of the model (continua)

```
Method: GCV   Optimizer: magic
Smoothing parameter selection converged after 5 iterations.
The RMS GCV score gradiant at convergence was 1.456957 .
The Hessian was positive definite.
The estimated model rank was 10 (maximum possible: 10)
Model rank =  10 / 10

Basis dimension (k) checking results. Low p-value (k-index<1) m
indicate that k is too low, especially if edf is close to k'.

       k'  edf k-index p-value
s(x) 9.00 2.13    1.00    0.44
```

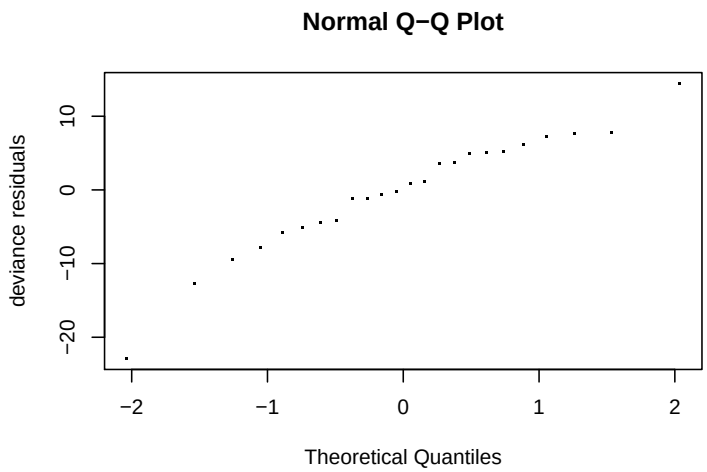
A better model

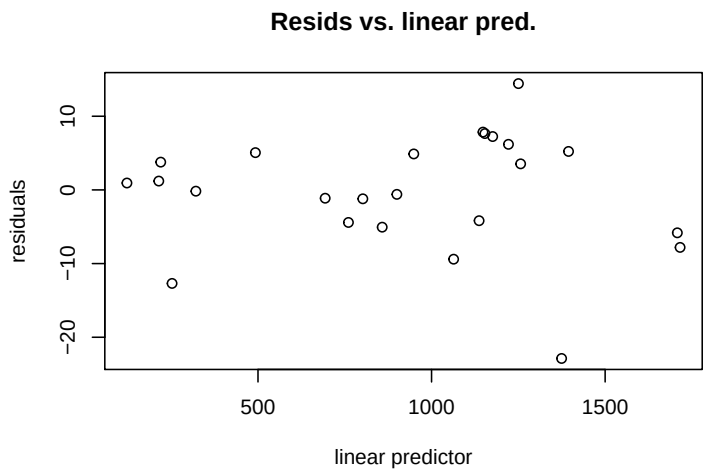
We may use a semiparametric regression to check to what extent the data are compatible with the linearity of the relationship implied by Hubble's law.

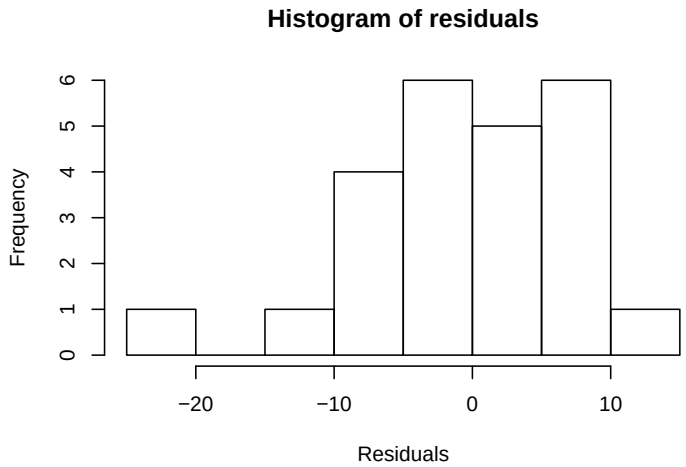
```
fitGam=gam(y~s(x),data=hubble,family=quasi(var=mu))
summary(fitGam)



Family: quasi
Link function: identity

Formula:
y ~ s(x)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   924.37      48.42   19.09 3.52e-15

(Intercept) ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
     edf Ref.df     F p-value
s(x)   1      1 145.7  <2e-16 ***
---
Signif. codes:
```

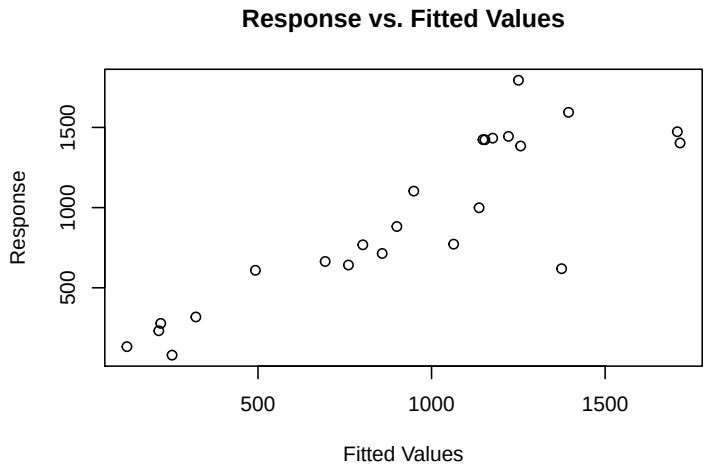


A check of the model









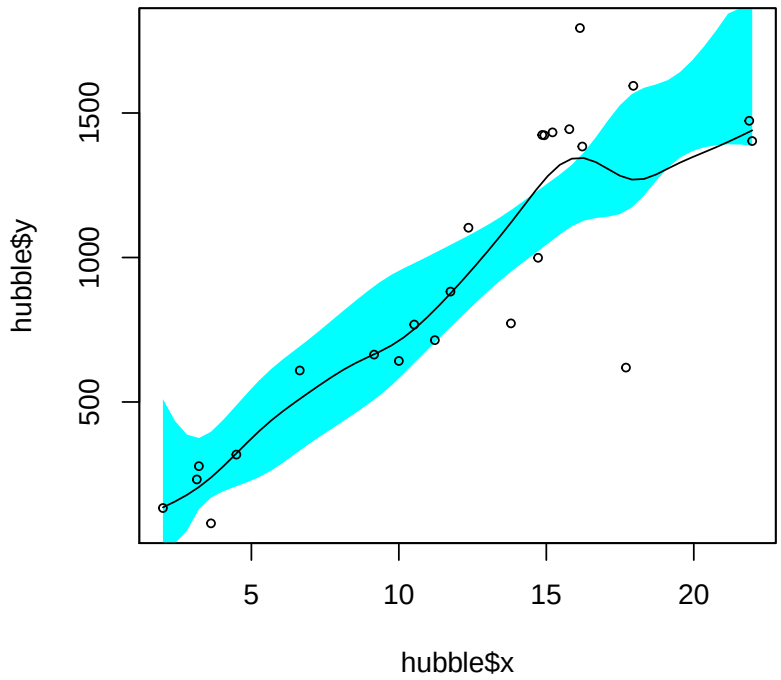
A check of the model (continua)

```
Method: GCV   Optimizer: outer newton
full convergence after 9 iterations.
Gradient range [-3.456696e-05,-3.456696e-05]
(score 72.3071 & scale 60.87159).
Hessian positive definite, eigenvalue range [3.456634e-05,3.456
Model rank =  10 / 10

Basis dimension (k) checking results. Low p-value (k-index<1) m
indicate that k is too low, especially if edf is close to k'.

        k'   edf k-index p-value
s(x) 9.000 1.000   0.989    0.42
```

A better method?

In order to check for linearity we may use the kernel regression approach.

```
library(sm)


Warning:  package 'sm' was built under R version
3.3.1
Package 'sm', version 2.2-5.4:  type help(sm) for
summary information

sm.regression(hubble$x, hubble$y,
              model = "linear")


Test of linear model:  significance =  0.267
```

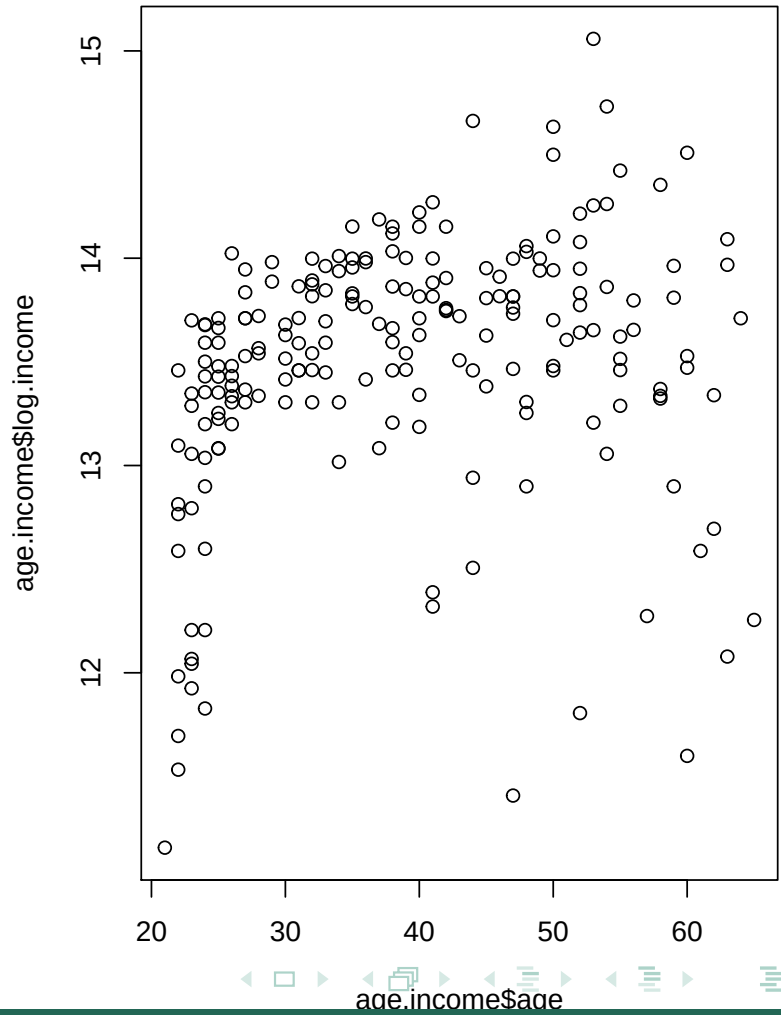


Indice

The relationship between age and income

We consider data on (log)income and age

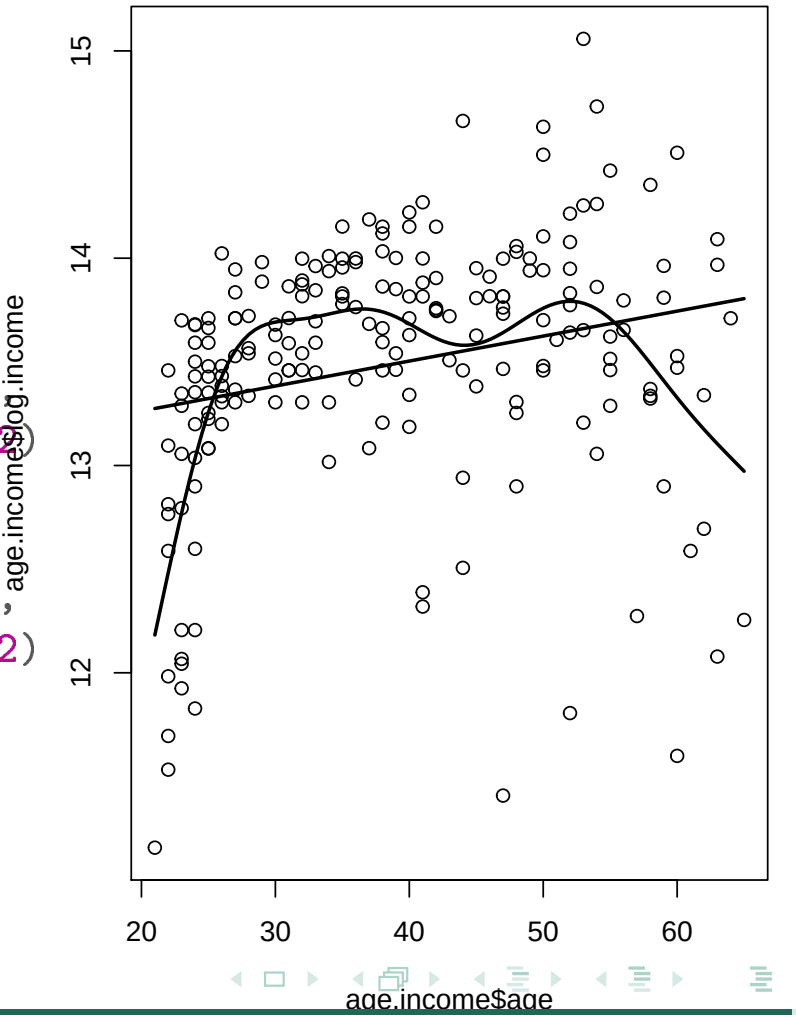
```
data(age.income)
age.income[1,]
plot(age.income$age,
     age.income$log.income)
```



The relationship between age and income

We consider data on (log)income and age

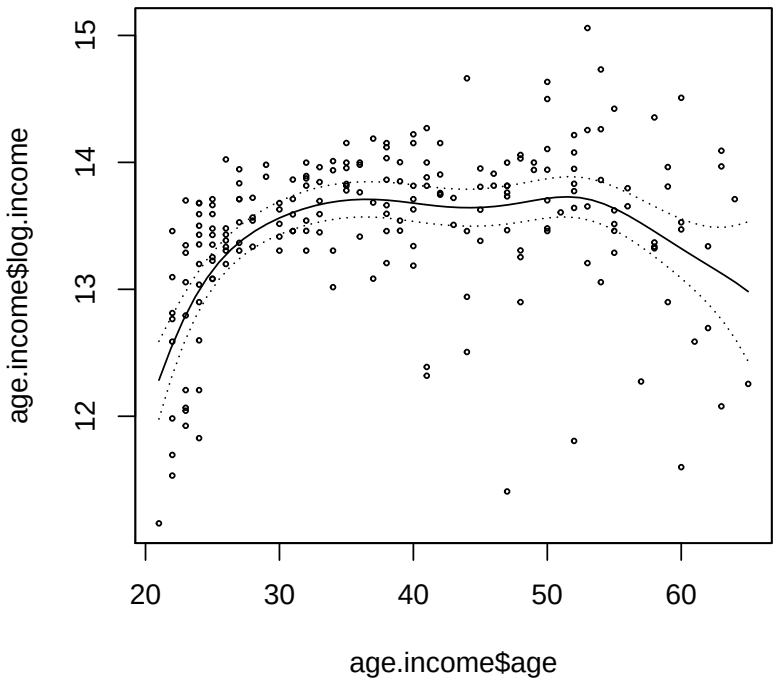
```
fit=gam(log.income~age,data=age.income)
curve(predict(fit,newdata=data.frame(age=x)),
                type="response"),ad=TRUE,lwd=2)

fit=gam(log.income~s(age),data=age.income)
curve(predict(fit,newdata=data.frame(age=x)),
                type="response"),ad=TRUE,lwd=2)
```



An alternative

As an alternative we may use the kernel regression approach.

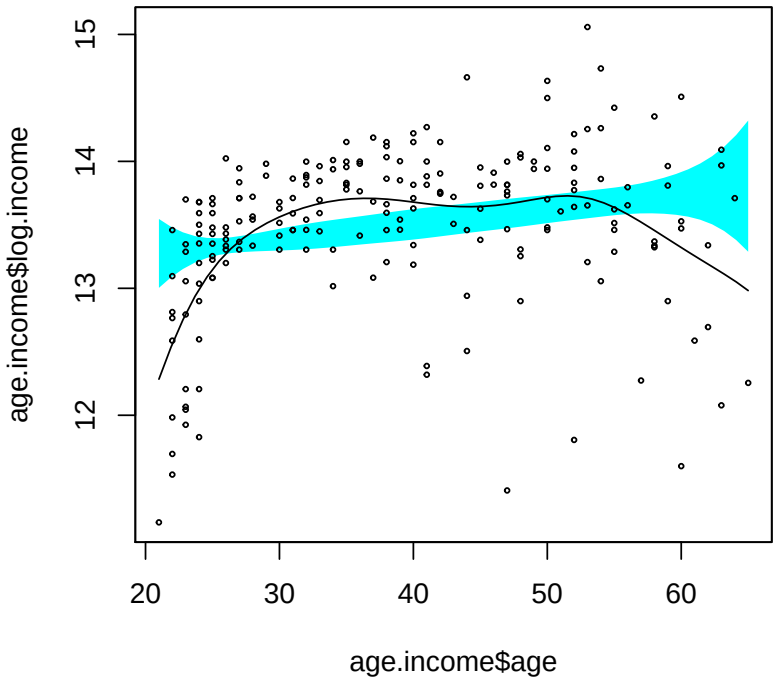
```
library(sm)
sm.regression(age.income$age,
              age.income$log.income,
              se = TRUE)
```



An alternative

We can also use th ekernel regression framework to check linearity

```
library(sm)
sm.regression(age.income$age,
              age.income$log.income,
              model = "linear")

Test of linear model:  significance =  0
```



Indice

Epidemiology: relationship between pollutants and mortality

It is widely believed that high pollutant concentrations may lead to higher mortality.

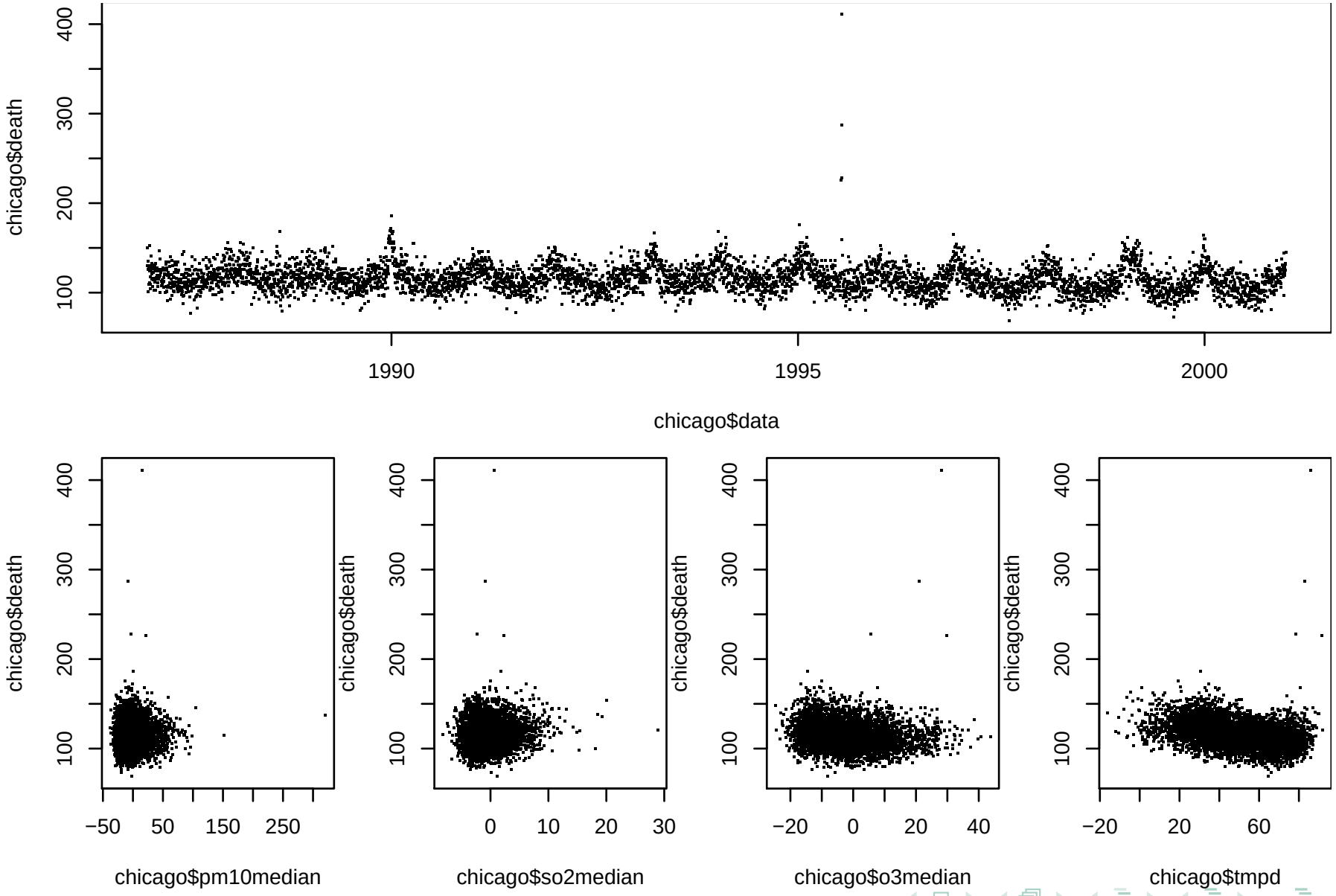
In analyzing such a relationship, it must be kept in mind that mortality is also affected by other factors such as, for example, the temperature.

Air pollution and death in Chicago

The following variables were observed daily from 1/1/1987 to 31/12/2000

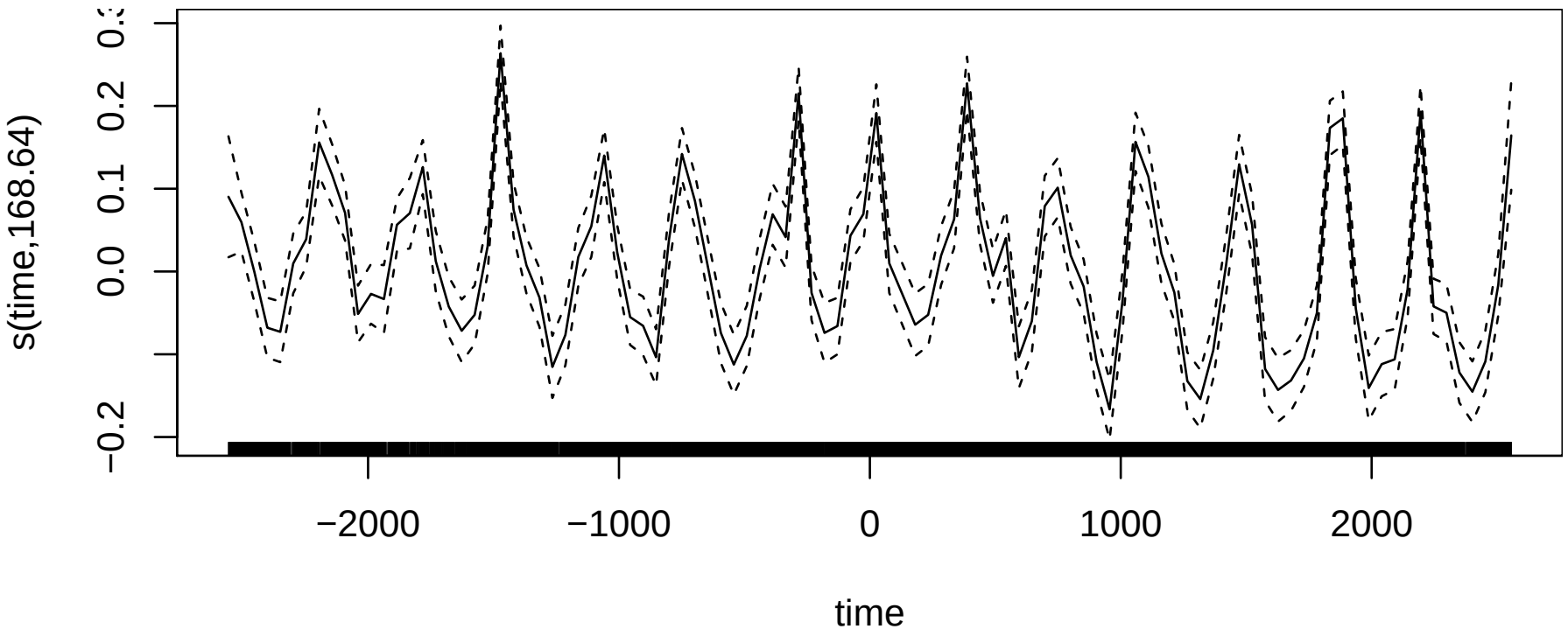
- `death` total deaths (per day).
- `pm10medianmedian` particles in $2.5 - 10$ per cubic m
- `pm25medianmedian` particles < 2.5 mg per cubic m (more dangerous).
- `o3medianOzone` in parts per billion
- `so2median` Median Sulpher dioxide measurement
- `time` time in days
- `tmpd` temperature in fahrenheit

A look at the data



A first model

```
fit1=gam(death~s(time,bs="cr",k=200)+
           pm10median+so2median+o3median+
           tmpd,
         data=chicago,family=poisson)
par(mar=c(5,4,0,0))
plot(fit1)
```



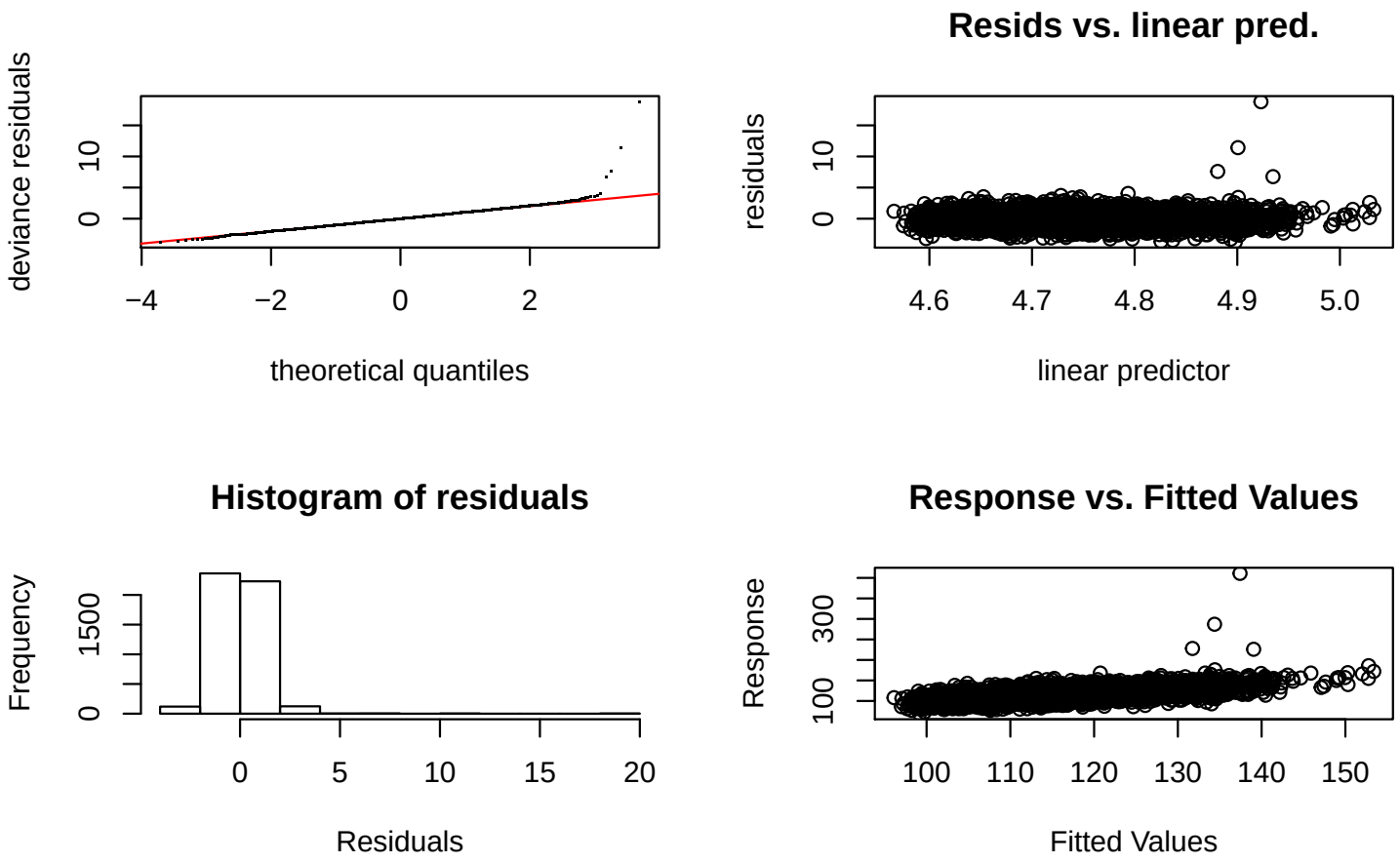
A first model (continua)

```
summary(fit1)



Family: poisson
Link function: log

Formula:
death ~ s(time, bs = "cr", k = 200) + pm10median + so2median +
    o3median + tmpd

Parametric coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 4.7006394  0.0091690 512.666  < 2e-16 ***
pm10median  0.0004321  0.0000889   4.861 1.17e-06 ***
so2median   0.0008184  0.0005523   1.482    0.138
o3median    0.0009306  0.0002032   4.581 4.63e-06 ***
tmpd        0.0009318  0.0001784   5.223 1.76e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
          edf Ref.df Chi.sq p-value
s(time) 168.6    188   2090  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.351   Deviance explained = 38.7%
UBRE = 0.25467  Scale est. = 1         n = 4841
```

A first model (continua)

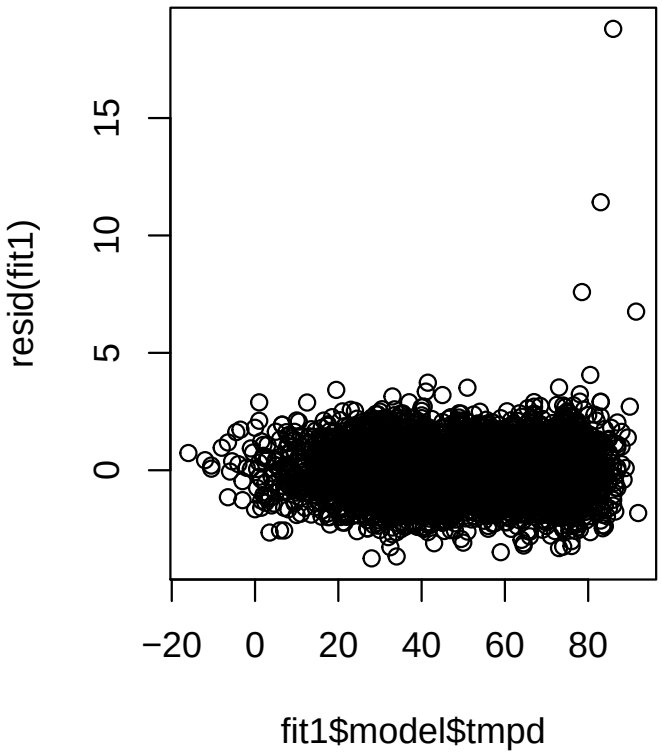
A first model

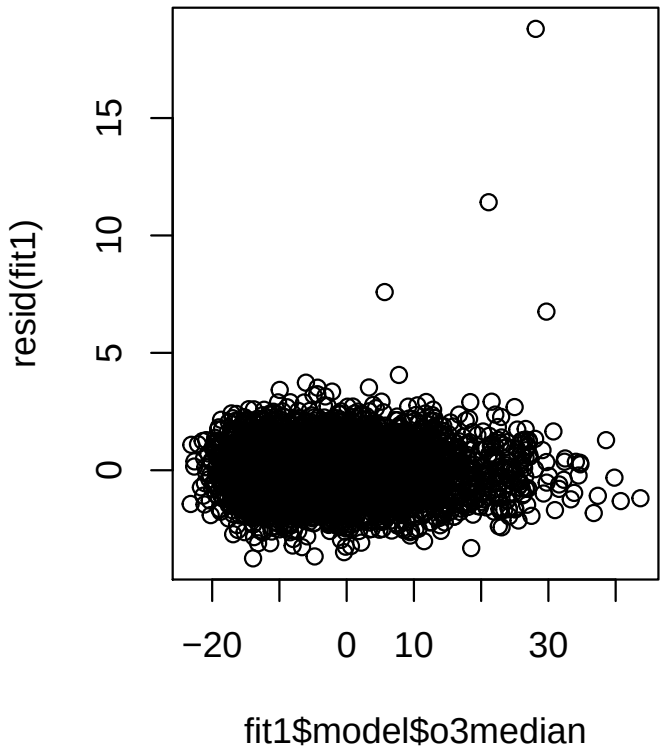
```
gam.check(fit1)
```



A look at the highest residuals

```
fit1$model[sort.list(resid(fit1),decreasing=TRUE)[1:4],]
chicago[3110:3120,]
par(mfrow=c(1,2))
plot(fit1$model$tmpd,resid(fit1))
plot(fit1$model$o3median,resid(fit1))
```

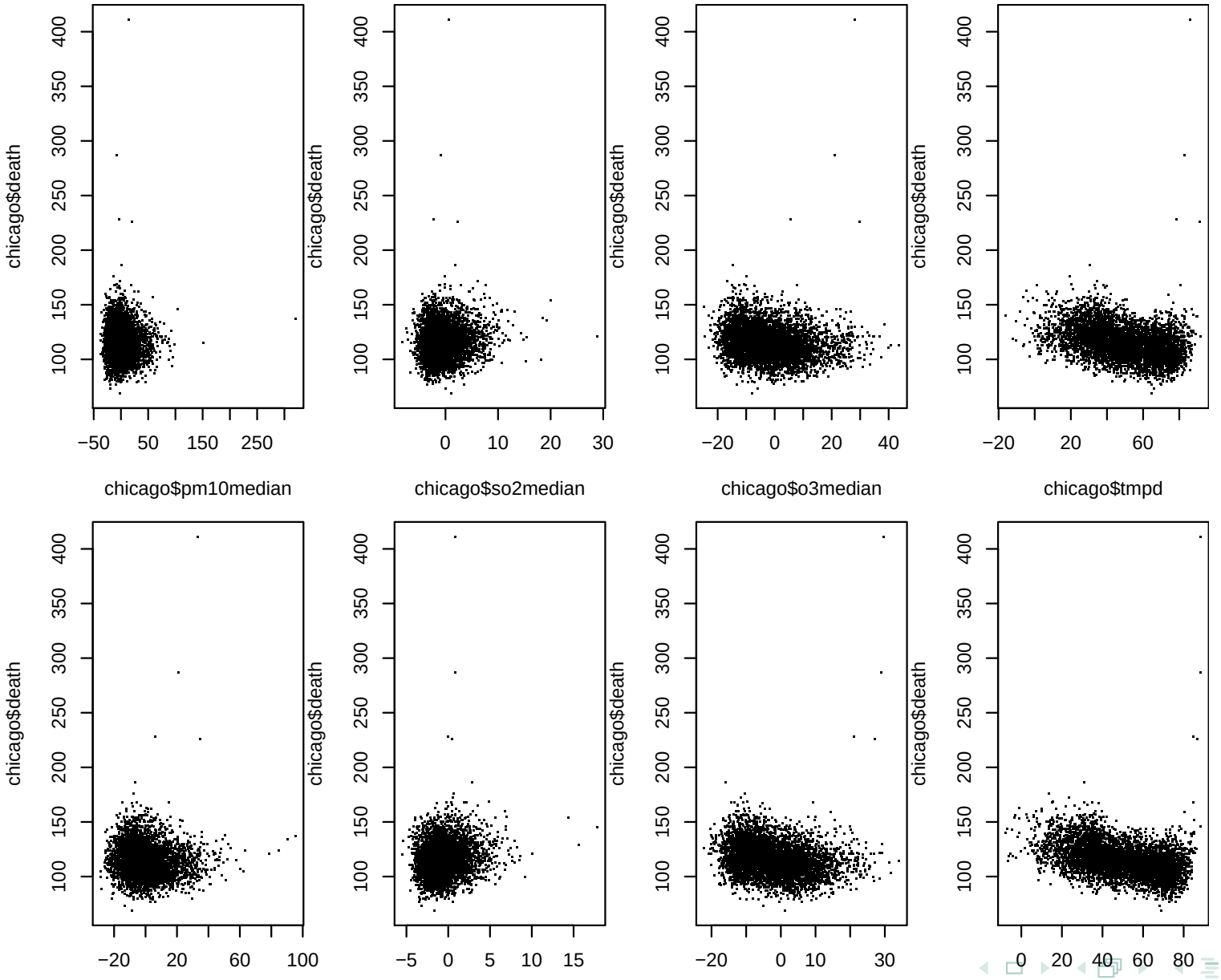




Lagged data

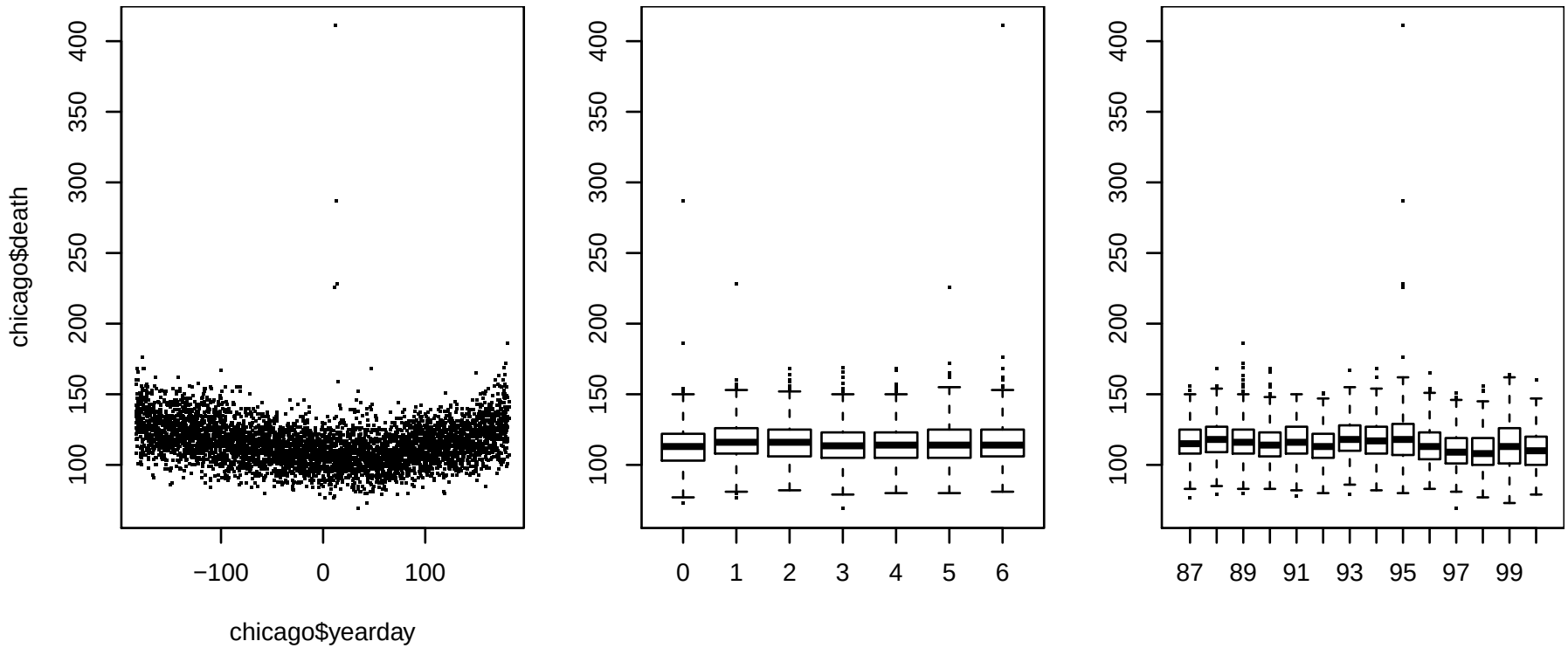
```
meanlag3=function(x){
  n=length(x)
  c(NA,NA,NA,(x[1:(n-3)]+x[2:(n-2)]+x[3:(n-1)]+x[4:n])/4)
}
chicago$pm10medianLag=meanlag3(chicago$pm10median)
chicago$pm25medianLag=meanlag3(chicago$pm25median)
chicago$o3medianLag=meanlag3(chicago$o3median)
chicago$so2medianLag=meanlag3(chicago$so2median)
chicago$tmpdLag=meanlag3(chicago$tmpd)
```

A look at the data: lagged covariates



A better (?) model for time

```
chicago$data=seq(as.Date("1987/1/1"), as.Date("2000/12/31"), "days")
chicago$yearday=as.POSIXlt(chicago$data)$yday-366/2
chicago$weekday=as.POSIXlt(chicago$data)$wday
chicago$year=as.POSIXlt(chicago$data)$year
```

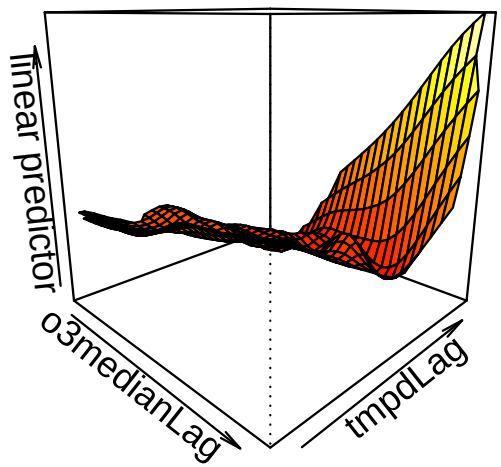


Alternative models

```
fit2=gam(death~s(time,bs="cr",k=200)+
           pm10medianLag+so2medianLag+o3medianLag+
           tmpdLag,
         data=chicago,family=poisson)
fit3=gam(death~s(yearday,bs="cr")+
           pm10medianLag+so2medianLag+o3medianLag+
           s(tmpdLag),
         data=chicago,family=poisson)
fit4=gam(death~s(yearday,bs="cr")+
           so2medianLag+s(pm10medianLag)+
           s(o3medianLag,tmpdLag,k=40),
         data=chicago,family=poisson)
fit5=gam(death~s(yearday,bs="cr")+
           so2medianLag+s(pm10medianLag,bs="cr",k=6)+
           te(o3medianLag,tmpdLag,k=8) ,
         data=chicago,family=poisson)
```

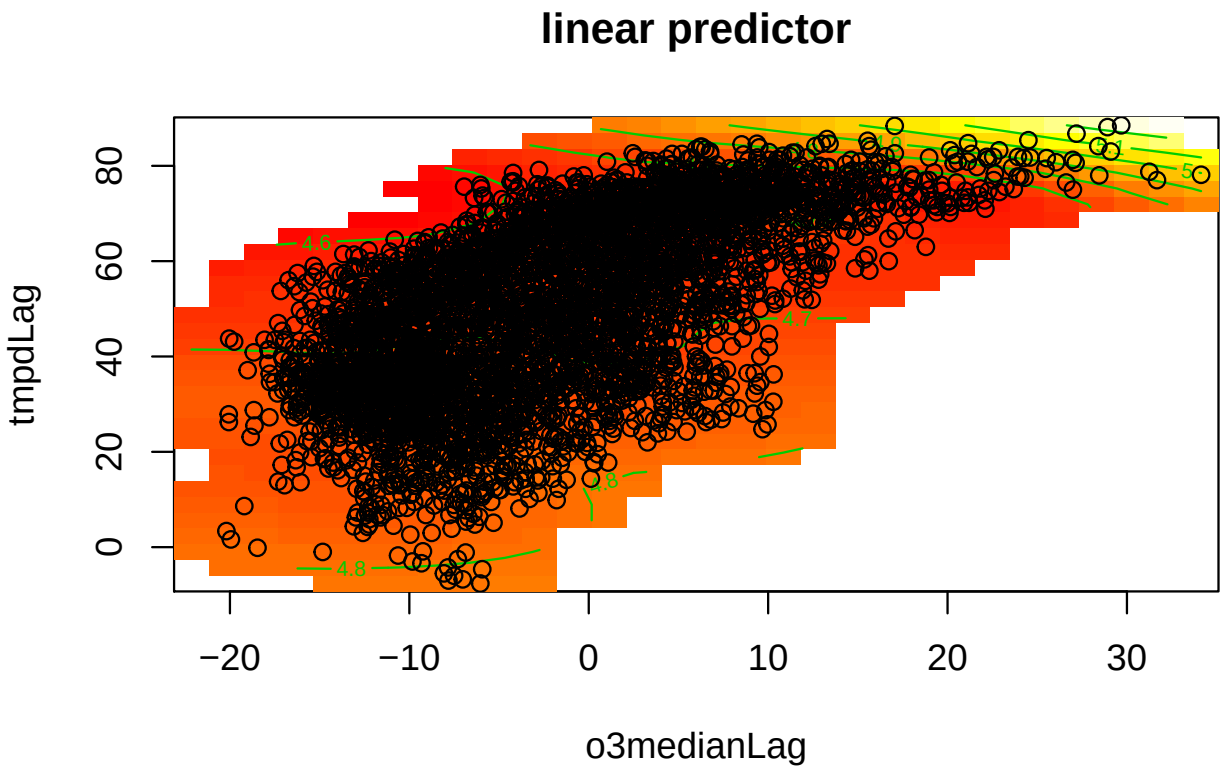
Alternative models

```
vis.gam(fit4,c("o3medianLag","tmpdLag"),theta=45,too.far=0.07)
```



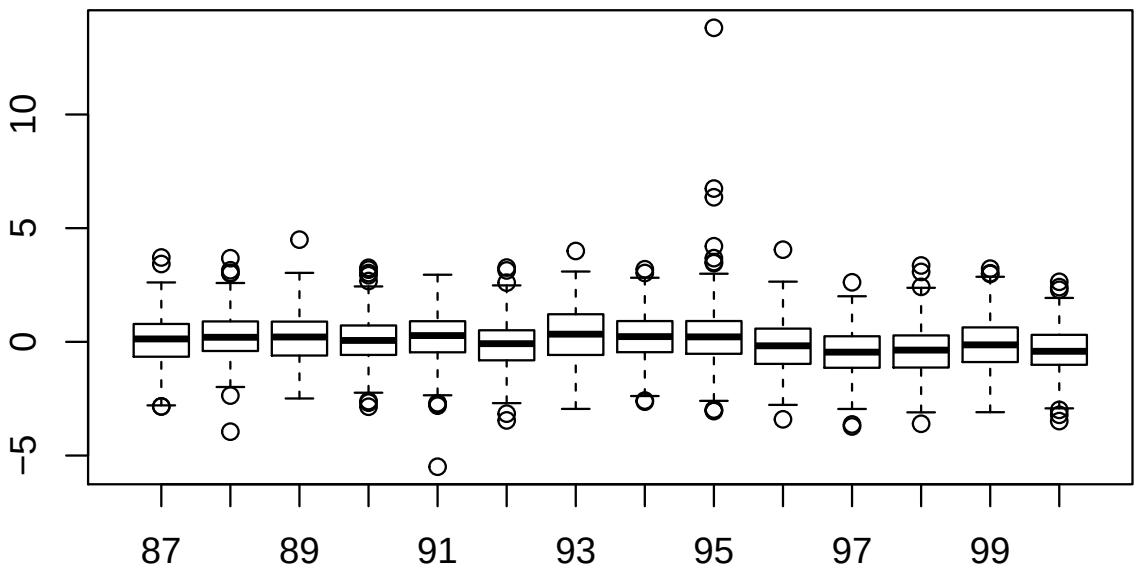
Alternative models (continua)

```
vis.gam(fit4,c("o3medianLag","tmpdLag"),plot.type="contour",too.far=0.07)
points(fit4$model$o3medianLag,fit4$model$tmpdLag)
```



Consider also the year (or other ways to include a time trend)?

```
plot(as.factor(na.omit(chicago[,-c(3,10)])$year),residuals(fit4))
```



Indice

CO_2 measured at southpole

We consider CO_2 concentrations at South pole

Data in co2s include

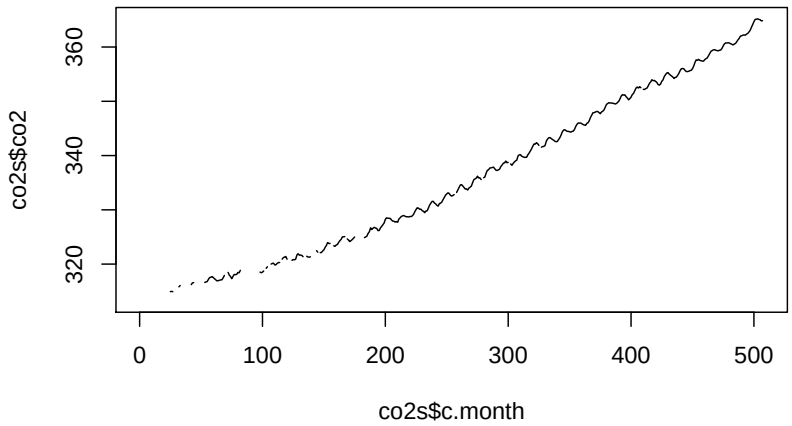
- co2 atmospheric CO2 concentration in parts per million
- `c.month` cumulative number of months since Jan 1957
- `month` month of year

CO_2 measured at southpole

We consider CO_2 concentrations at South pole

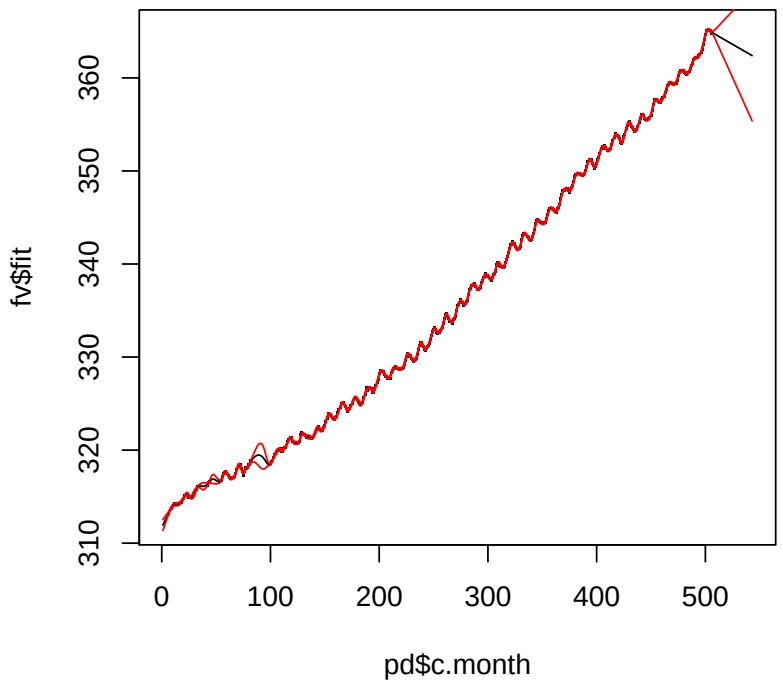
Data in co2s include

- co2 atmospheric CO2 concentration in parts per million
- `c.month` cumulative number of months since Jan 1957
- `month` month of year



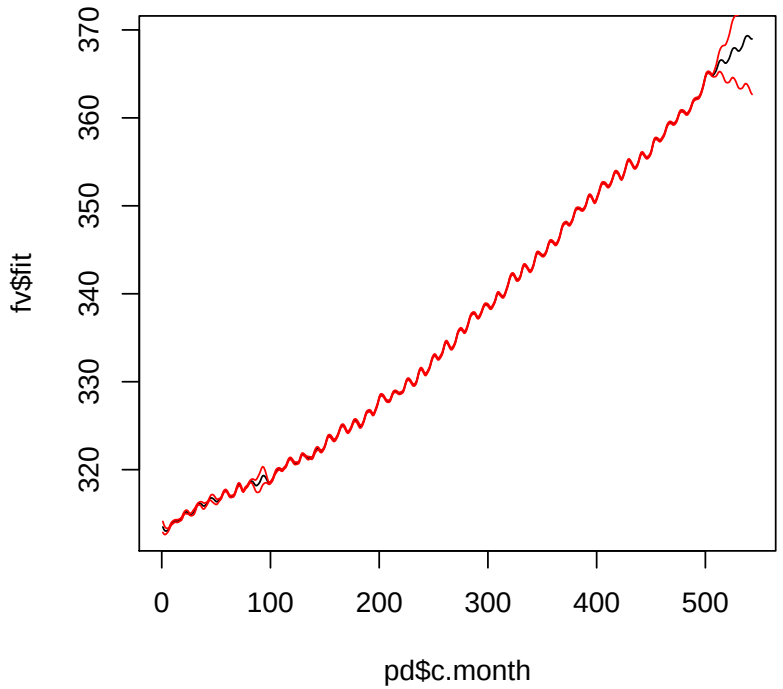
A model for trend prediction

```
b=gam(co2~s(c.month,k=300,bs="cr"),data=co2s)
plot(c.month,co2,type="l")
n=nrow(co2s)
pd <- data.frame(c.month=1:(n+36))
fv <- predict(b,pd,se=TRUE)
plot(pd$c.month,fv$fit,type="l")
lines(pd$c.month,fv$fit+2*fv$se,col=2)
lines(pd$c.month,fv$fit-2*fv$se,col=2)
```



A model for trend prediction 2

```
b2 <- gam(co2~s(month,bs="cc")+
            s(c.month,bs="cr",k=300),
          data=co2s,
          knots=list(month=seq(1,13,length=10)))

pd2 <- data.frame(c.month=1:(n+36),
                  month=rep(1:12,length.out=n+36))
fv <- predict(b2,pd2,se=TRUE)
plot(pd$c.month,fv$fit,type="l")
points(co2s$c.month,co2s$co2,pch=".")
lines(pd$c.month,fv$fit+2*fv$se,col=2)
lines(pd$c.month,fv$fit-2*fv$se,col=2)
```



Indice

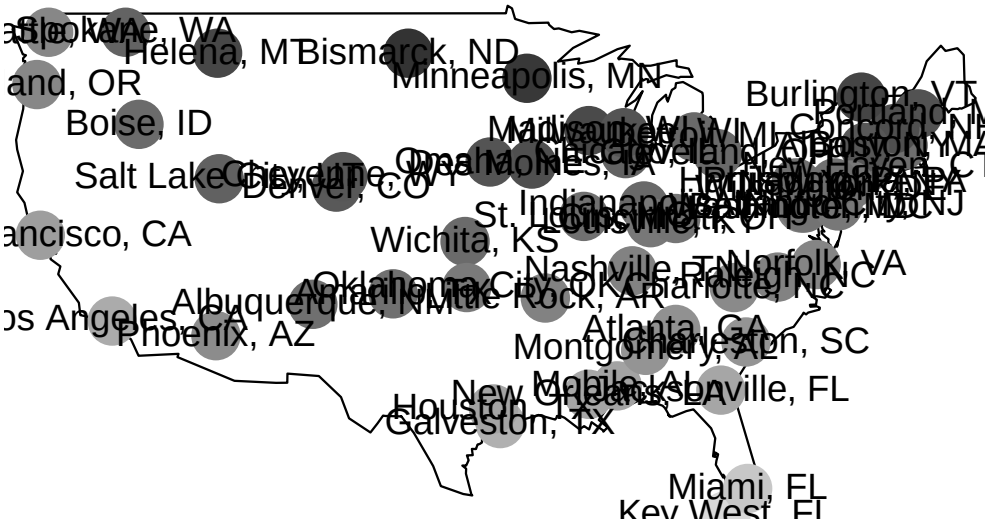
Data on temperatures of US cities

Dataset `uscities` contains minimum temperatures of some US cities together with their longitude and latitude.

```
library(SemiPar)
library(maps)
data(ustemp)
ustemp$min.temp=round((ustemp$min.temp - 32) / (9/5),1)
attach(ustemp)
grey.levs <- min.temp+20
col.vec <- paste("grey",as.character(grey.levs),sep="")
map("usa")
points(-longitude,latitude,
     col=col.vec,pch=16,cex=3,xlim=c(-130,-60))
text(-longitude,latitude,as.character(city))
detach(ustemp)
```

Data on temperatures of US cities

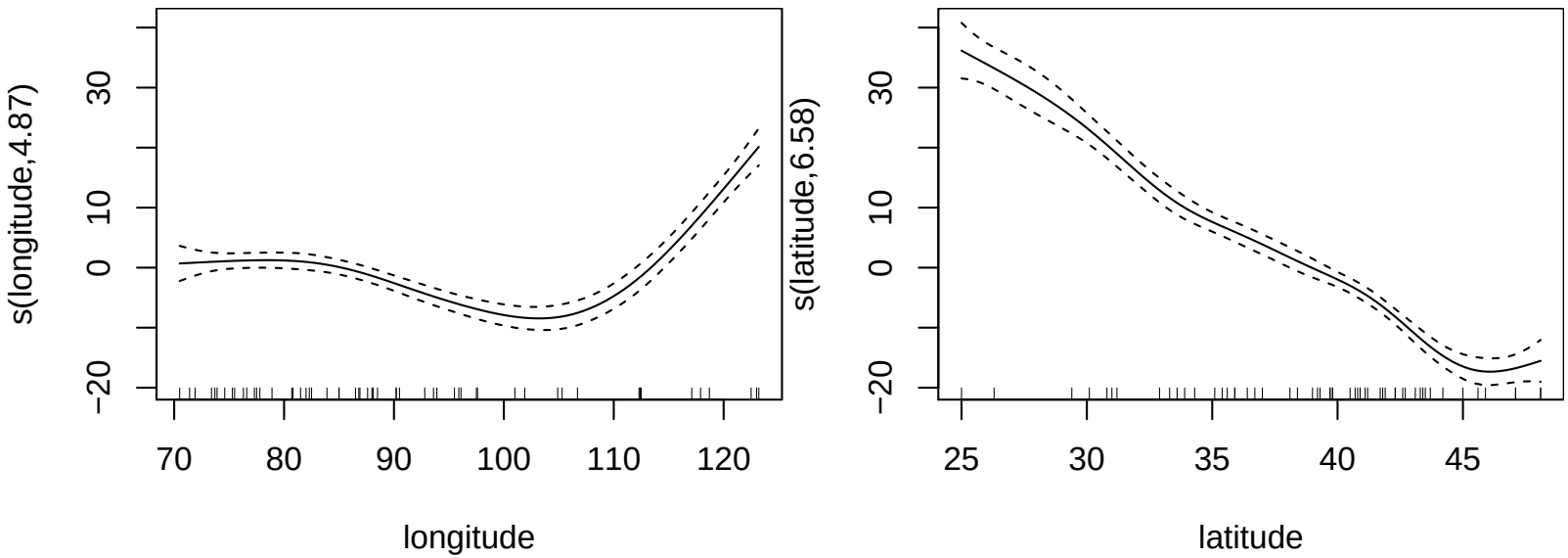
Dataset `uscities` contains minimum temperatures of some US cities together with their longitude and latitude.



Spatial model for temperature

It is expected that temperature varies with latitude, also the longitude may be relevant, we can model this phenomena as

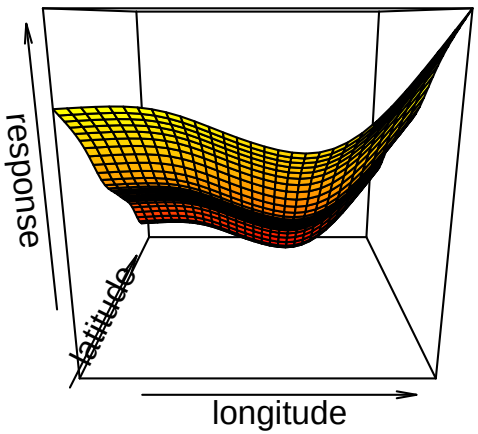
```
fitSep=gam(min.temp~s(longitude)+s(latitude),data=ustemp)
par(mfrow=c(1,2),mar=c(5,4,0,0))
plot(fitSep)
```

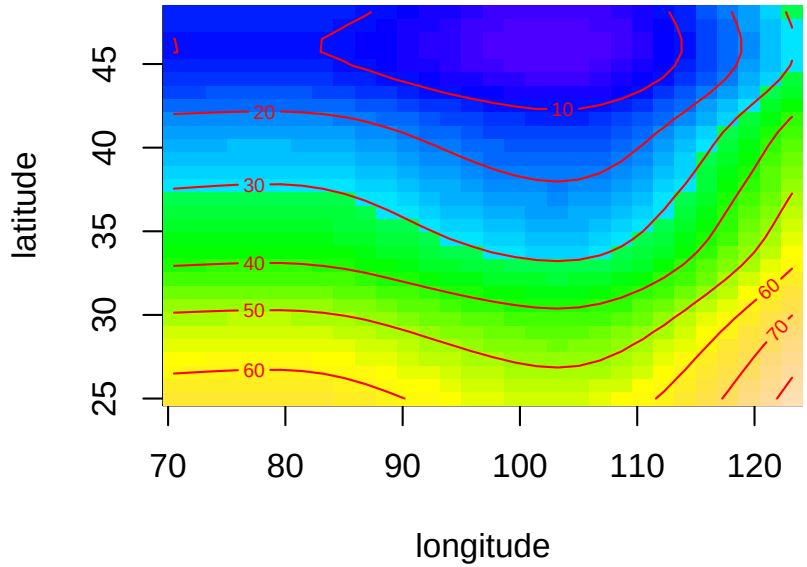


Model for temperatures: spatial trend

We can look at the spatial trend through the commands

```
par(mfrow=c(1,2),mar=c(5,4,0,0))
vis.gam(fitSep,type="response")
vis.gam(fitSep,type="response",
        plot.type="contour",color="topo")
```

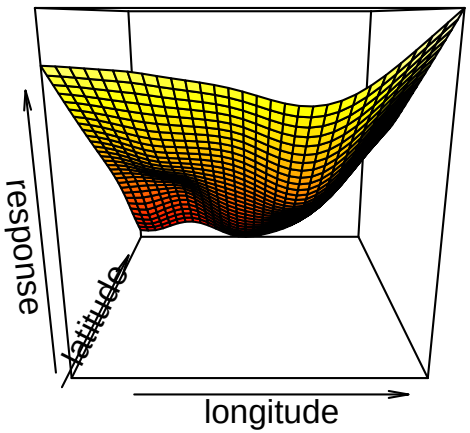


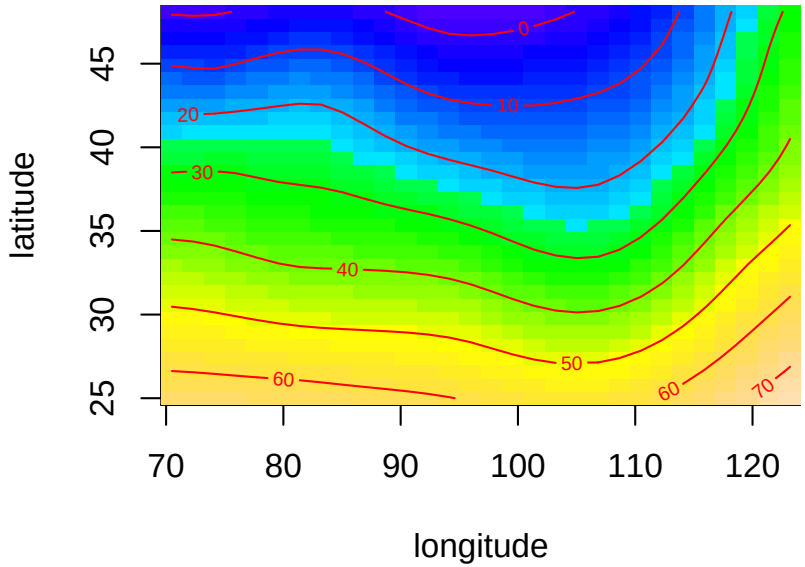


Bivariate spatial model for temperature

It is expected that temperature varies with latitude, also the longitude may be relevant, we can model this phenomena as

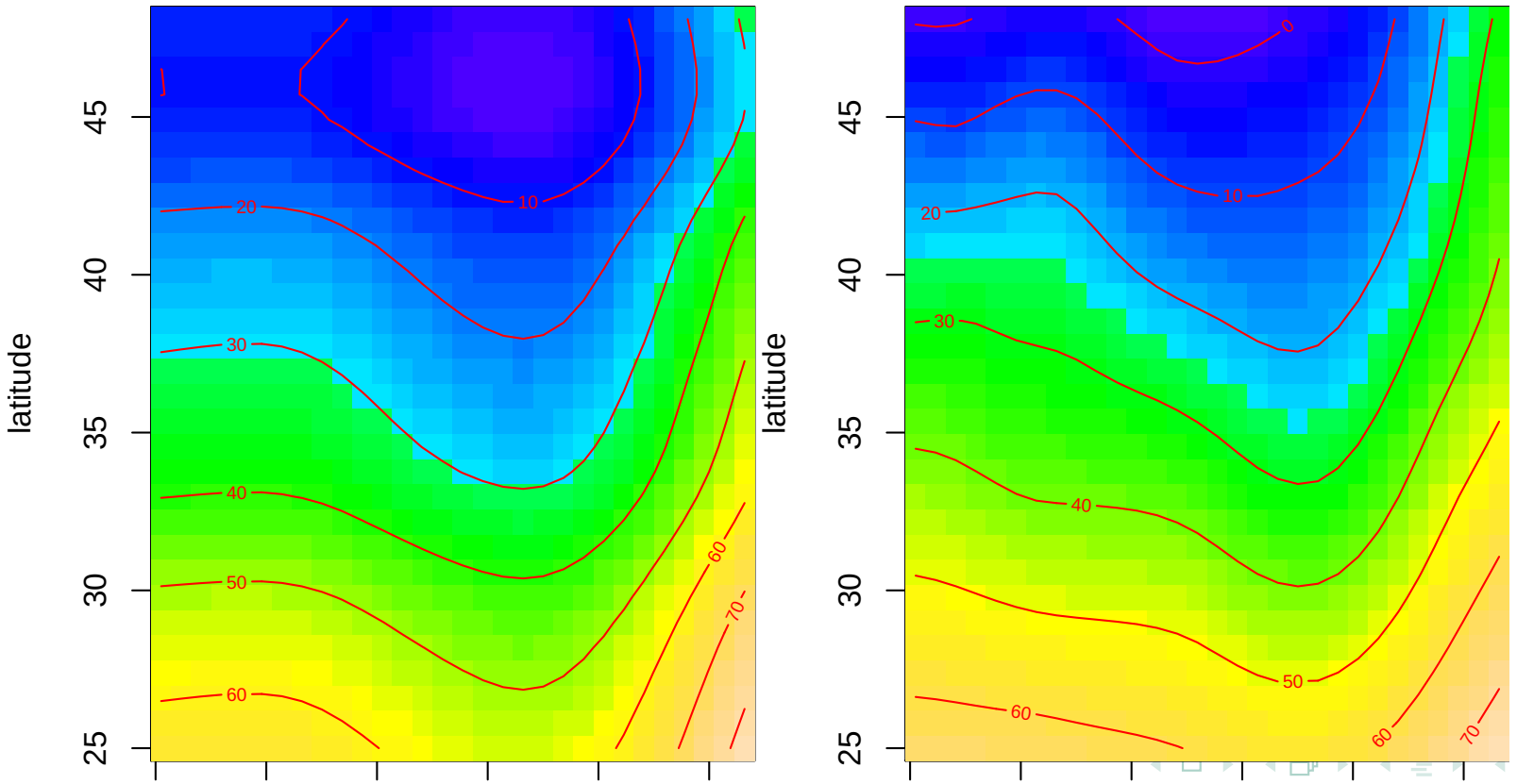
```
fitJoint=gam(min.temp~s(longitude,latitude),data=ustemp)
par(mfrow=c(1,2),mar=c(5,4,0,0))
vis.gam(fitJoint,type="response")
vis.gam(fitJoint,type="response",
        plot.type="contour",color="topo")
```





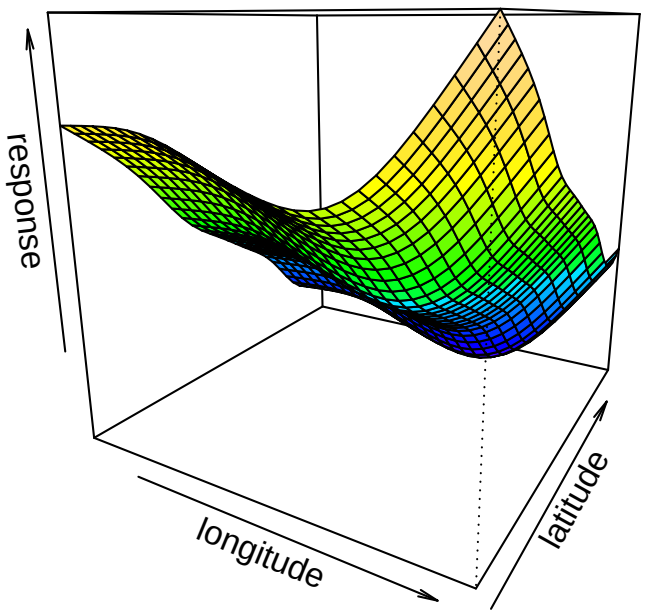
Comparison of models

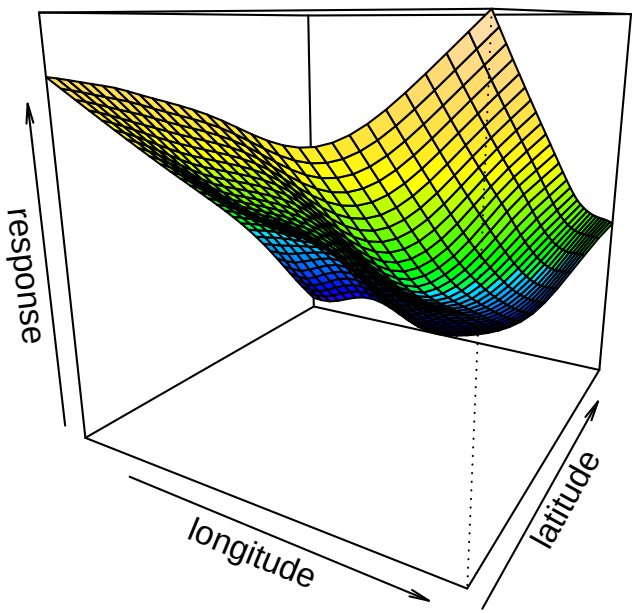
```
vis.gam(fitSep,type="response",
        plot.type="contour",color="topo")
vis.gam(fitJoint,type="response",
        plot.type="contour",color="topo")
```



Comparison of models

```
vis.gam(fitSep,type="response",
        color="topo",theta=30)
vis.gam(fitJoint,type="response",
        color="topo",theta=30)
```

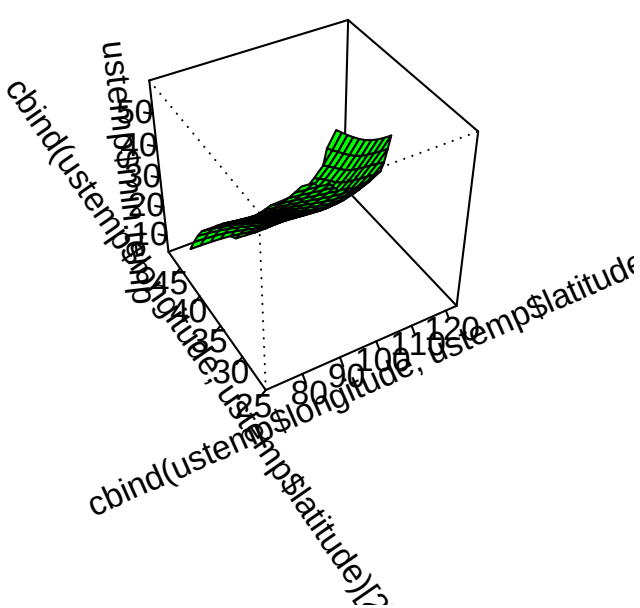


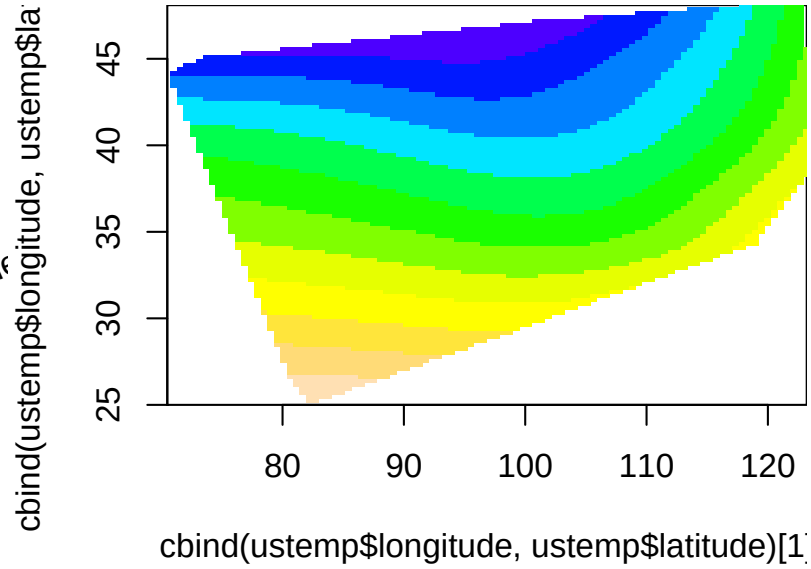


Bivariate kernel regression for US temperatures

The surface trend can be estimated through kernel regression

```
par(mfrow=c(1,2),mar=c(5,4,0,0))
fitK=sm.regression(cbind(ustemp$longitude,ustemp$latitude),
                   ustemp$min.temp)
fitK=sm.regression(cbind(ustemp$longitude,ustemp$latitude),
                   ustemp$min.temp,
                   display="image",ngrid=100)
```





```
library(mgcv)
library(rworldmap)
newmap <- getMap(resolution = "low")

lonseq=sort(-seq(min(ustemp$longitude),max(ustemp$longitude),le
latseq=seq(min(ustemp$latitude),max(ustemp$latitude),length=100
z=outer(lonseq,latseq,FUN=function(x,y) predict(fitSep,newdata=
plot(newmap,xlim = range(lonseq),ylim = range(latseq),asp = 1)
contour(lonseq,latseq,z,add=TRUE,col="red")
points(-ustemp$longitude,ustemp$latitude,
     col=col.vec,pch=16,cex=3,xlim=c(-130,-60))
text(-ustemp$longitude,ustemp$latitude,as.character(city))
```

Indice

Mackarel population

Assessment of fish stocks is substantial for management, but is also very difficult because of the high mobility.

We consider the problem of estimation of mackarel population near west coast of UK.

Fish population estimation based on egg production

An effective method is based on egg production:

- egg number is surveyed (easier than surveying fishes)
  - Eggs are usually sampled by hauling a fine meshed net up from the sea bed to the sea surface
  - The number of eggs, of the target species, in the sample is then counted
  - the volume of water sampled is known
- egg production rates per kg of adult fish can be assessed from adults caught in trawls

hence one works out the number (or more often mass) of adult fish required to produce this number or production rate.

Survey data on eggs

A data frame with 16 columns. Each row corresponds to one sample of eggs.

- `egg.count` The number of stage I eggs in this sample.
- `egg.dens` The number of stage I eggs per m^2 per day (calculated from `egg.count` and information about sampling net size, and egg stage duration).
- `b.depth` The sea bed depth at the sampling location.
- `c.dist` The distance from the sample location to the 200m contour.
- `lon` The longitude of the sample station in degrees east.
- `lat` The latitude of the sample station in degrees north.
- `time` The time of day (in hours) at which the sample was taken.
- `salinity` The salinity of the water at the sampling location.
- `flow` Reading from the flow meter attached to the sampling net.

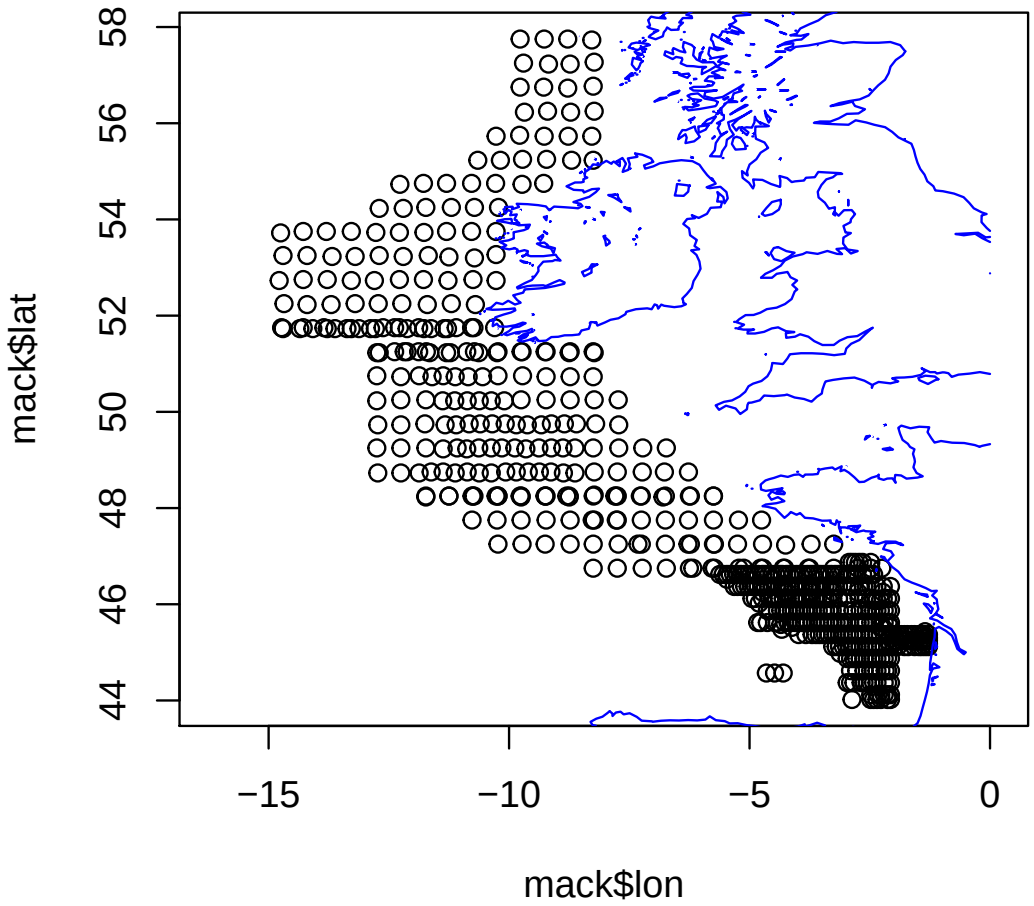
Survey data on eggs (continua)

- `s.depth` The depth that the sampling net started sampling from (the net is dropped to this depth and then hauled up to the surface, filtering eggs etc out of the water as it goes).
- `temp.surf` Temperature at sea surface at the sampling location.
- `temp.20m` The temperature 20m down at the sampling location.
- `net.area` The area of the sampling net in square metres.
- `country` Country responsible for the boat that took this sample.
- `vessel` A code identifying the boat that took this sample.
- `vessel.haul` A code uniquely identifying this sample, given that the vessel is known.

A look at the data: sample locations

```
data(mack)
data(coast)
plot(mack$lon,mack$lat,asp=1)
lines(coast$lon,coast$lat,col=1)
```

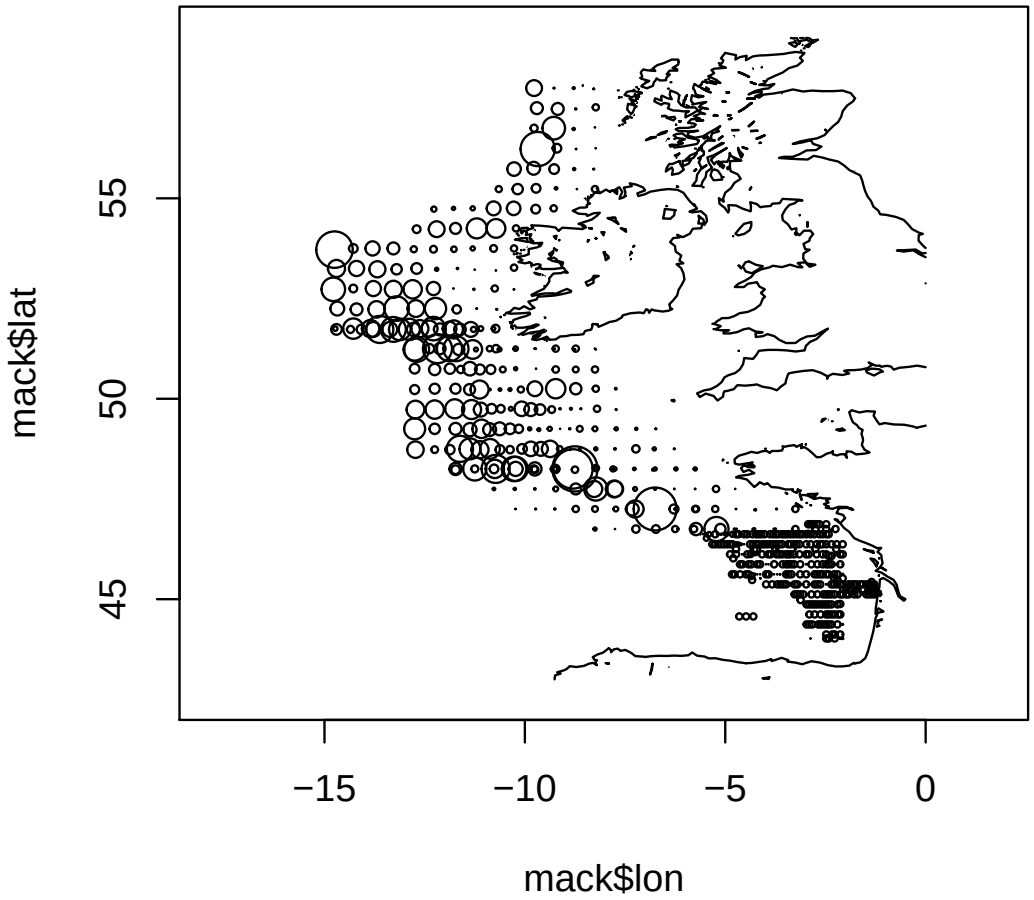
(Note: coast contains the coast profile.)



A look at the data: egg counts

```
data(mack)
data(coast)
symbols(mack$lon,mack$lat,
        circles=mack$egg.count,
        inches=0.1,asp=1)
lines(coast$lon,coast$lat,
      col="blue")
```

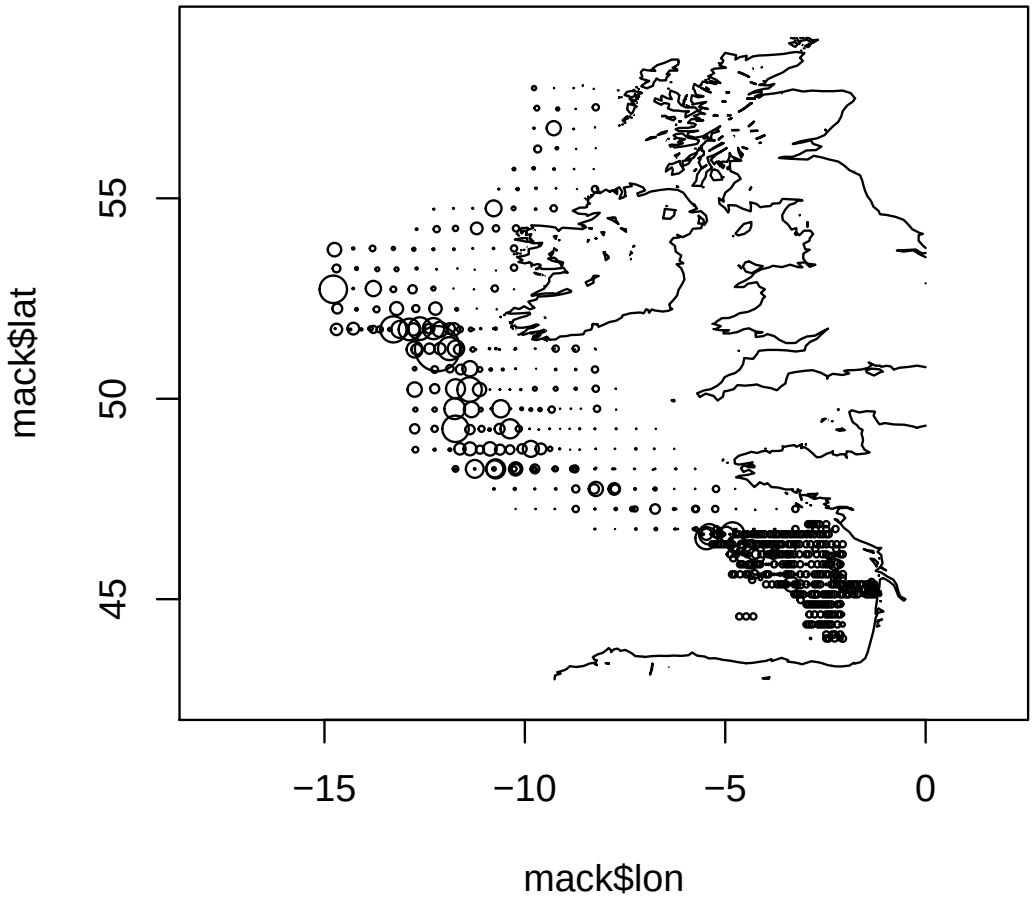
(Note: coast contains the coast profile.)



A look at the data: egg densities

```
data(mack)
data(coast)
symbols(mack$lon,mack$lat,
        circles=mack$egg.dens,
        inches=0.1,asp=1)
lines(coast$lon,coast$lat,
      col="blue")
```

(Note: coast contains the coast profile.)



Model

A reasonable model assumption may be that the `egg.counts` are Poisson distributed with mean λ proportional to the net area, that is

$$Y_i \sim \text{Poisson}(\lambda_i)$$

$$\lambda_i = g_i \times [\texttt{net.area}]_i$$

hence in terms of the linear predictor

$$\log(\lambda_i) = \log(g_i) + \log([\texttt{net.area}]_i)$$

so

- $\log([\texttt{net.area}]_i)$ will be treated as an offset
- $\log(g_i)$ wil be modeled as a function of predictors

Penalization shrinks the smoother, to what?

Recall the spline representation

$$f(x) = \beta_1 + \beta_2 x[+\beta_2 x^2 + \beta_3 x^3] + \sum_{j=1}^{K} b_j B_j(x)$$

and the penalized objective function

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \mathbf{b}^T S \mathbf{b}$$

if λ increases indefinitely, then the b_j are shrunk to 0, this means that the spline tends toward polynomial driven by the β_j.

It may be convenient to have a representation such that as *lambda* increases indefinitely, the function shrinks to 0 (that is, on the limit the spline does not contribute to the fit).

This can be done by adding the β_j to the penalization.

Penalization shrinks the smoother, to what?

Recall the spline representation

$$f(x) = \beta_1 + \beta_2 x[+\beta_2 x^2 + \beta_3 x^3] + \sum_{j=1}^{K} b_j B_j(x)$$

and the penalized objective function

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \mathbf{b}^T S \mathbf{b}$$

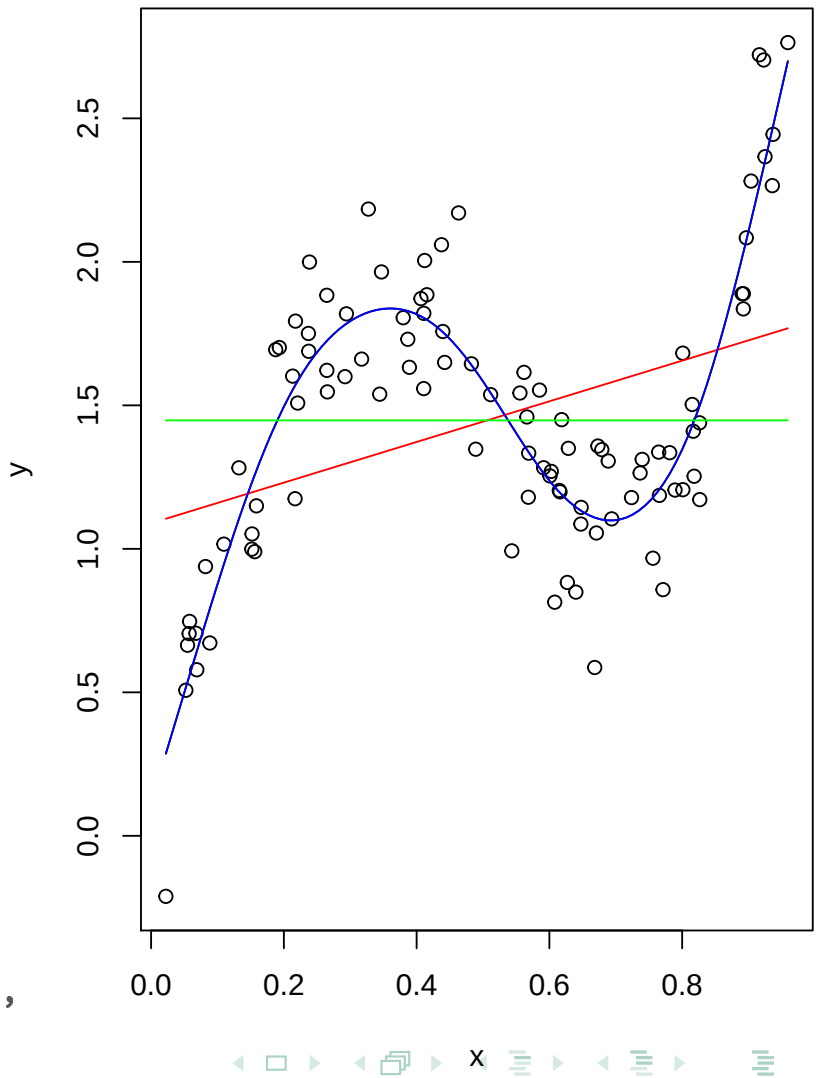
if λ increases indefinitely, then the b_j are shrunk to 0, this means that the spline tends toward polynomial driven by the β_j.

This can be done by adding the β_j to the penalization.

$$\begin{bmatrix} \beta & \mathbf{b} \end{bmatrix} \begin{bmatrix} \delta I & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} \beta \\ \mathbf{b} \end{bmatrix}$$

Shrinking to zero using mgcv::gam

```
x=sort(runif(100,0,1))
m=3*x+sin(2*pi*x)
y=m+rnorm(100,0,0.5*sd(m))
plot(x,y)

fit=gam(y~s(x,bs="tp"))
fit1=gam(y~s(x,bs="tp"),sp=10^10)
fit2=gam(y~s(x,bs="ts"))
fit3=gam(y~s(x,bs="ts"),sp=10^10)
plot(x,y)
curve(predict(fit,
              newdata=data.frame(x=x)),
      add=TRUE)
curve(predict(fit1,
              newdata=data.frame(x=x)),
      add=TRUE,col="red")
curve(predict(fit2,
              newdata=data.frame(x=x)),
      add=TRUE,col="blue")
curve(predict(fit3,newdata=data.frame(x=x)),
      add=TRUE,col="green")
```

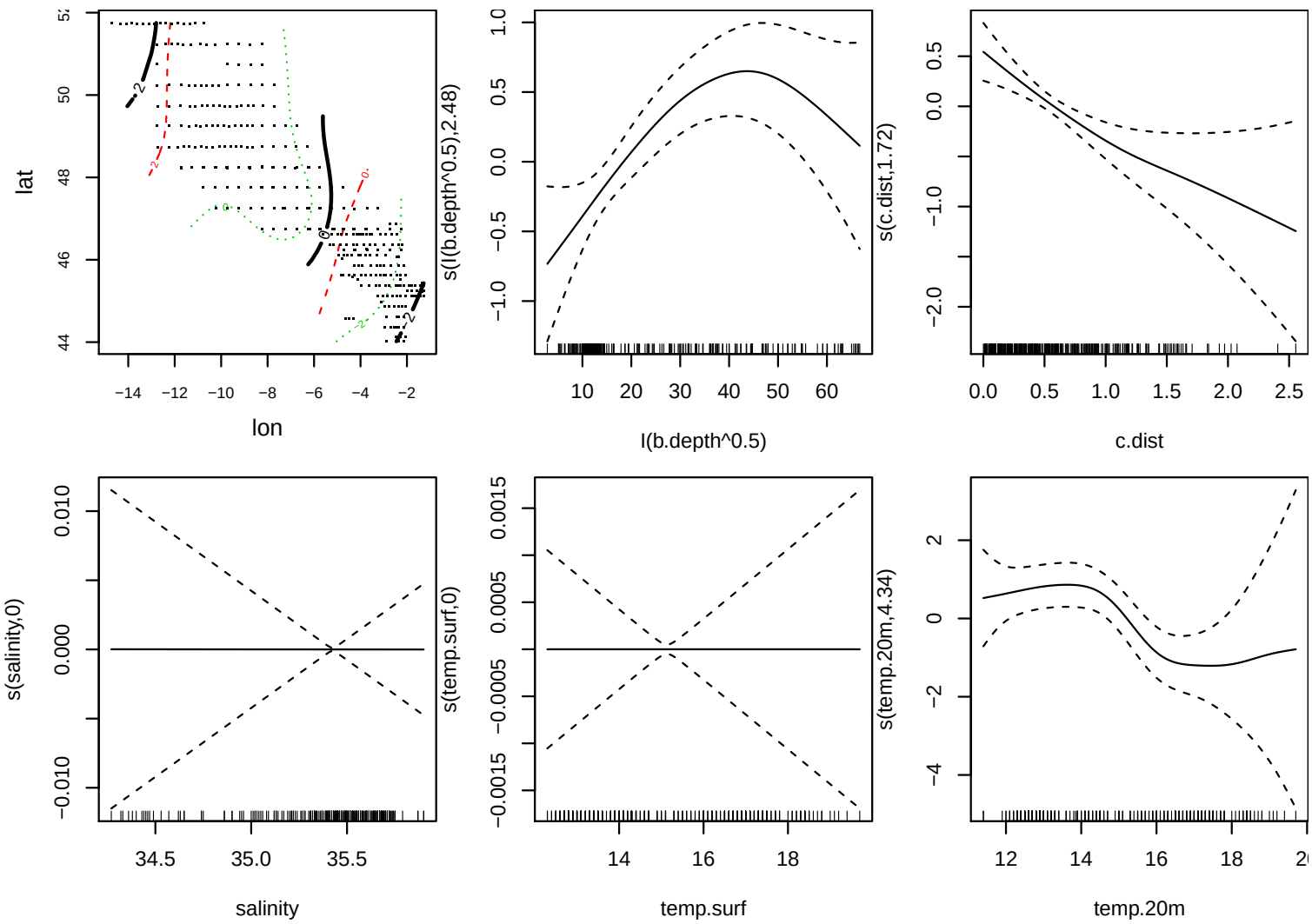


Estimate a model for mackerels

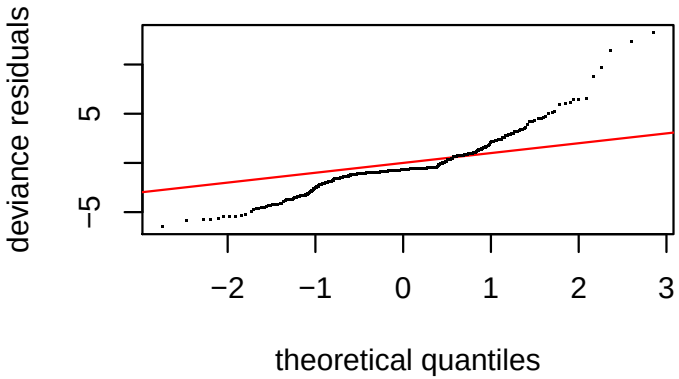
```
mack$log.net.area <- log(mack$net.area)
gm <- gam(egg.count ~ s(lon,lat,bs="ts")+
            s(I(b.depth^.5),bs="ts")+
            s(c.dist,bs="ts")+
            s(salinity,bs="ts")+
            s(temp.surf,bs="ts")+
            s(temp.20m,bs="ts")+
            offset(log.net.area),
          data=mack,
          family=poisson,
          scale=-1,gamma=1.4)
```

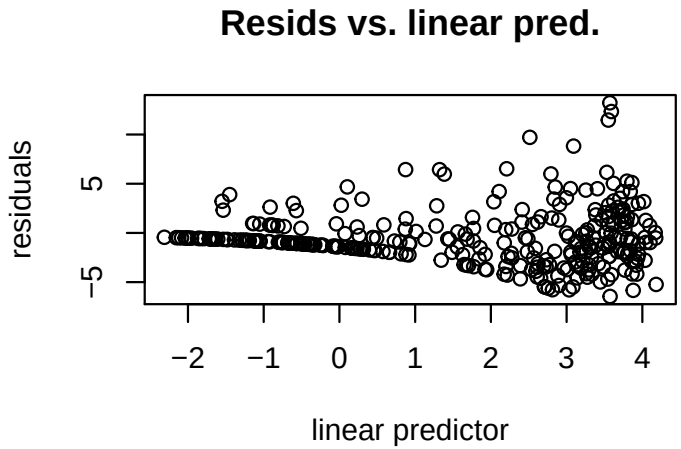
- `scale=-1` forces estimation of the scale parameter (we allow for overdispersion)
- `gamma=1.4` makes each edf count as 1.4 in the GCV scores, thus forces oversmoothing

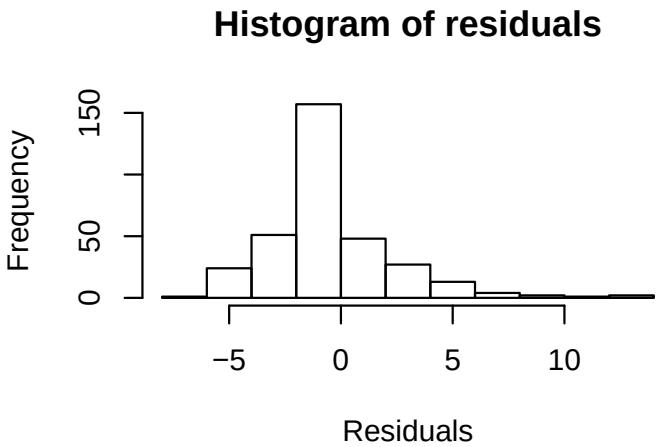
Model results

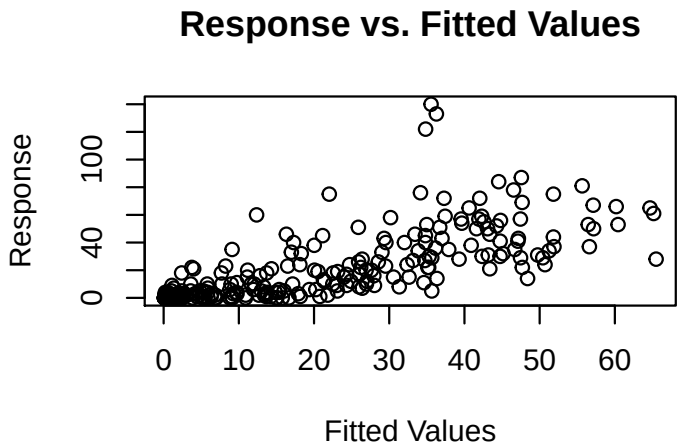


Model results









Model results

```
Family: poisson
Link function: log

Formula:
egg.count ~ s(lon, lat, bs = "ts") + s(I(b.depth^0.5), bs = "ts") +
    s(c.dist, bs = "ts") + s(salinity, bs = "ts") + s(temp.surf,
    bs = "ts") + s(temp.20m, bs = "ts") + offset(log.net.area)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.1696     0.1646   19.25   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                        edf Ref.df     F  p-value
s(lon,lat)        1.160e+01     29 1.926 3.47e-09 ***
s(I(b.depth^0.5)) 2.482e+00      9 2.319 2.12e-06 ***
s(c.dist)         1.724e+00      9 1.728 2.32e-07 ***
s(salinity)       6.884e-05      9 0.000  0.79980
s(temp.surf)      1.830e-06      9 0.000  1.00000
s(temp.20m)       4.337e+00      9 1.717  0.00135 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.571   Deviance explained = 77.2%
GCV = 9.6434  Scale est. = 10.648    n = 330
```

Estimate a model for mackerels

We drop salinity but keep temperature at surface since salinity had many missing data, so the fit is now on a different datset

```
gm2 <- gam(egg.count ~ s(lon,lat,bs="ts")+
             s(I(b.depth^.5),bs="ts")+
             s(c.dist,bs="ts")+
             s(temp.surf,bs="ts")+
             s(temp.20m,bs="ts")+
             offset(log.net.area),
           data=mack,
           family=poisson,
           scale=-1,gamma=1.4)
```

Estimate a model for mackerels (continua)

We loog at plots of the results and conclude that temperature at surface can be dropped

```
gm3 <- gam(egg.count ~ s(lon,lat,bs="ts")+
             s(I(b.depth^.5),bs="ts")+
             s(c.dist,bs="ts")+
             s(temp.20m,bs="ts")+
             offset(log.net.area),
           data=mack,
           family=poisson,
           scale=-1,gamma=1.4)
```

The model seem to have only significant contributions.

Estimate a model for mackerels (continua)

```
Family: poisson
Link function: log

Formula:
egg.count ~ s(lon, lat, bs = "ts") + s(I(b.depth^0.5), bs = "ts") +
    s(c.dist, bs = "ts") + s(temp.20m, bs = "ts") + offset(log.net.area)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.7088     0.1111   24.39   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                     edf Ref.df     F  p-value
s(lon,lat)        19.530     29 4.689  < 2e-16 ***
s(I(b.depth^0.5))  3.204      9 5.097 2.92e-12 ***
s(c.dist)          5.144      9 3.251 8.87e-07 ***
s(temp.20m)        5.552      9 3.074 9.33e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.665   Deviance explained = 81.8%
GCV = 7.0875  Scale est. = 8.0163     n = 634
```

Estimate a model for mackerels (continua)

The scale parameter is quite high, we may consider an alternative model such as the negaive binomial

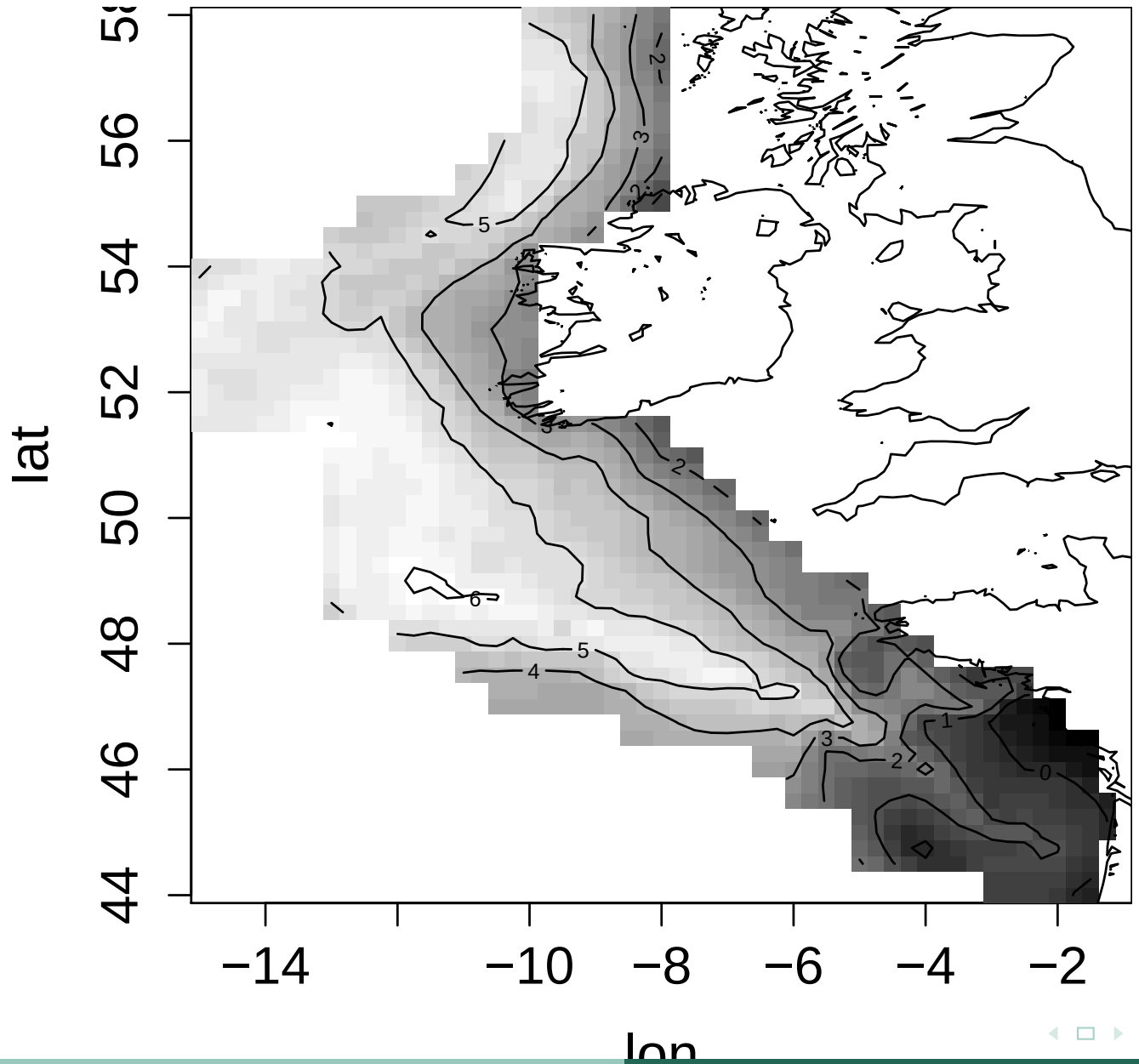
```
gm4<-gam(egg.count ~ s(lon,lat,bs="ts",k=40) +
           s(I(b.depth^.5),bs="ts") +
           s(c.dist,bs="ts") +
           s(temp.20m,bs="ts") +
           offset(log.net.area),
         data=mack,
         family=negbin(1),
         control=gam.control(maxit=100),gamma=1.4)
```

upon checking the residuals, in particular residuals versus fitted, the negative binomial specification seems to overstate variability for high fitted values.

Visualization of results

```
data(mackp)
mackp$log.net.area <- 0*mackp$lon # make offset column
lon<-seq(-15,-1,1/4);lat<-seq(44,58,1/4)
zz<-array(NA,57*57)
zz[mackp$area.index]<-predict(gm3,mackp)
image(lon,lat,matrix(zz,57,57),col=gray(0:32/32),
cex.lab=1.5,cex.axis=1.4)
contour(lon,lat,matrix(zz,57,57),add=TRUE)
lines(coast$lon,coast$lat,col=1)
```

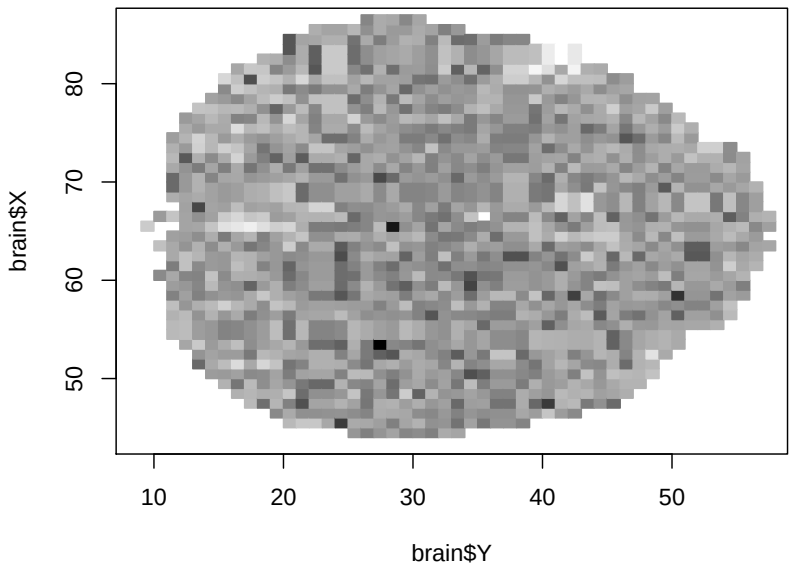
Visualization of results (continua)



Indice

Brain scan

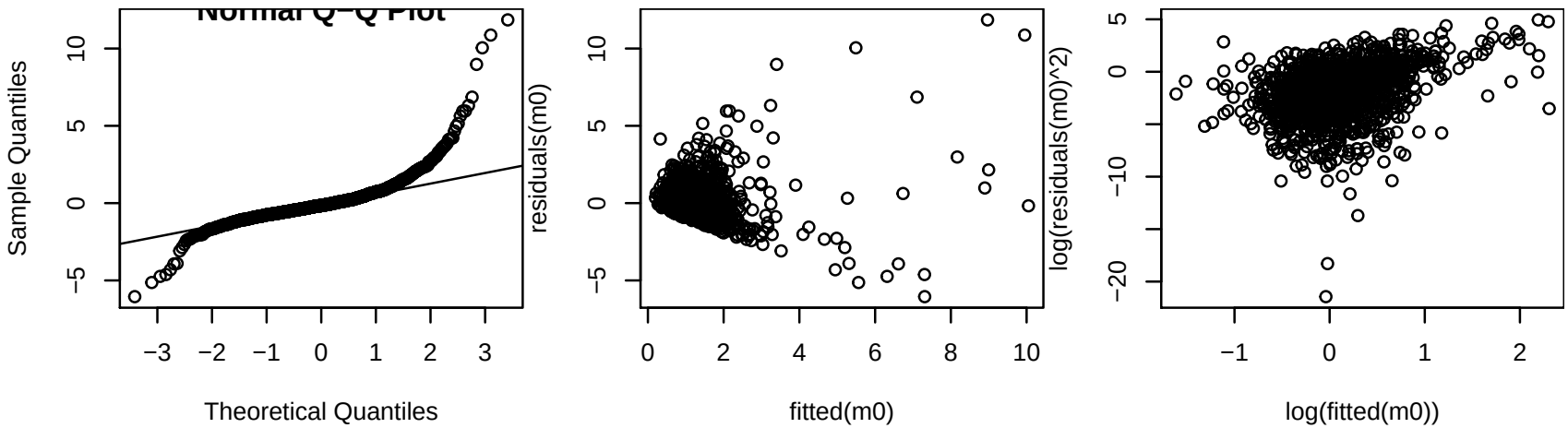
Observations of brain activity level at 1564 locations are available.



Models

A gaussian model is inadequate as residual plots quite clearly show

```
m0=gam(medFPQ~s(Y,X,k=100),data=brain)
par(mfrow=c(1,3),mar=c(5,4,0,0))
qqnorm(residuals(m0))
qqline(residuals(m0))
plot(fitted(m0),residuals(m0))
plot(log(fitted(m0)),log(residuals(m0)^2))
```



Models

The residuals suggest that the variance increases proportionally to the square of the mean

```
lm(log(residuals(m0)^2)~log(fitted(m0)))



Call:
lm(formula = log(residuals(m0)^2) ~ log(fitted(m0)))

Coefficients:
    (Intercept)  log(fitted(m0))
         -1.961            1.912
```

This leaves to possibilities

- a Gaussian model for the 4th root of y

```
m1=gam(medFPQ^.25~s(Y,X,k=100),data=brain)
```

- a Gamma model (with log-link to guaranteee positivity)

```
m2=gam(medFPQ~s(Y,X,k=100),data=brain,
       family=Gamma(link=log))
```

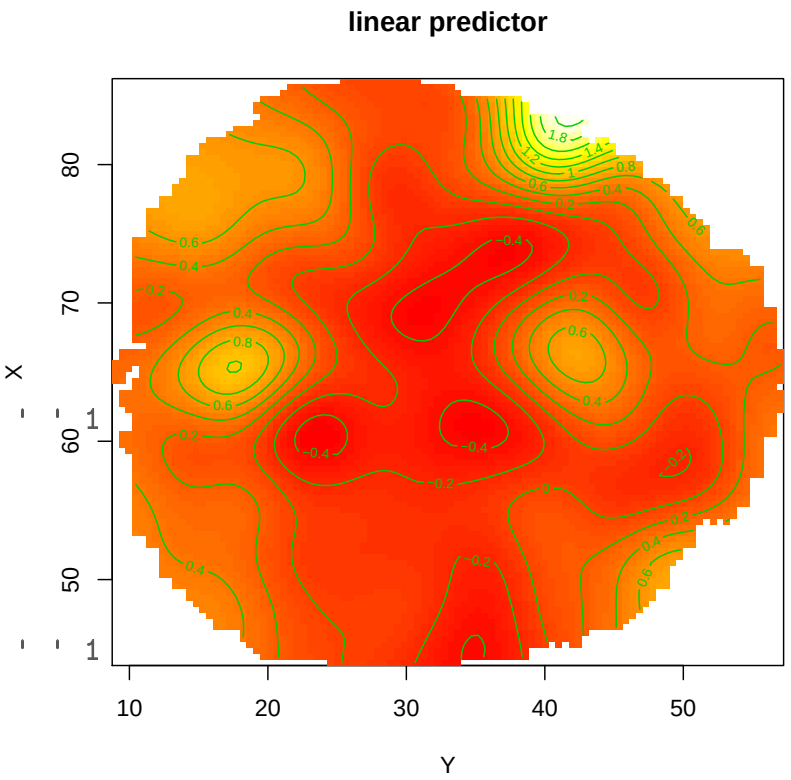
Both models are good (check residuals), we prefer the second since it is

A look at the gamma model

```
summary(m2)
```

```
Family: Gamma
Link function: log

Formula:
medFPQ ~ s(Y, X, k = 100)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.12031    0.01922   6.258 5.07e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
         edf Ref.df     F p-value
s(Y,X) 60.61  77.82 4.794  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.307   Deviance explained = 26.4%
GCV = 0.62169  Scale est. = 0.57802    n = 1564
```

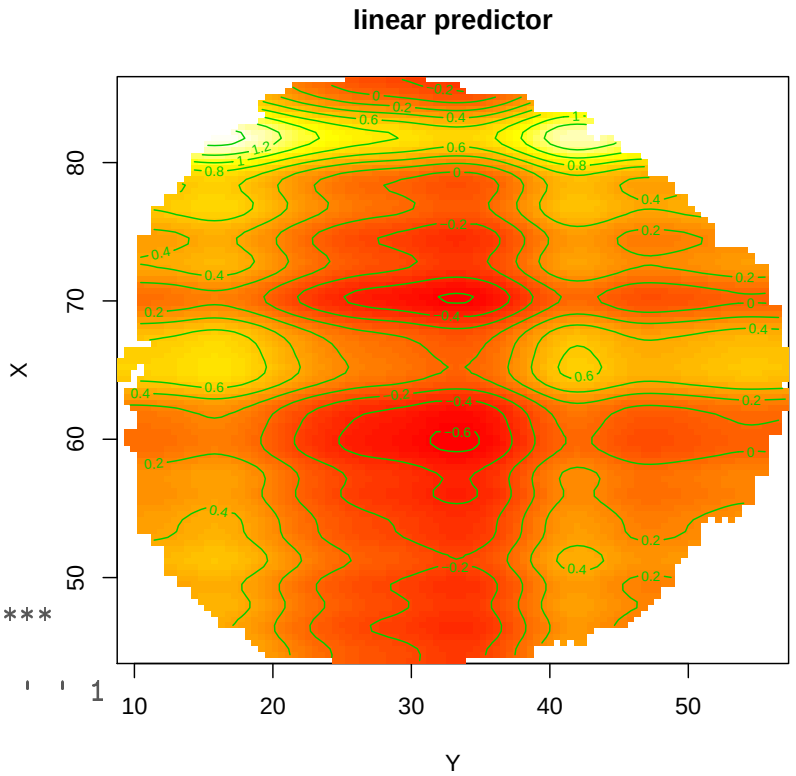
```
vis.gam(m2,plot.type="contour",too.far=0.02,n.grid=100)
```



A simpler model

```
vis.gam(m3,plot.type="contour",too.far=0.02,n.grid=100)
```

```
m3=gam(medFPQ~s(Y,k=30)+s(X,k=30),data=brain,
       family=Gamma(link=log))
m3$gcv.ubre


   GCV.Cp
0.6453502


anova(m2,m3,test="F")


Analysis of Deviance Table

Model 1: medFPQ ~ s(Y, X, k = 100)
Model 2: medFPQ ~ s(Y, k = 30) + s(X, k = 30)
  Resid. Df Resid. Dev      Df Deviance     F    Pr(>F)
1    1502.4     897.22
2    1533.2     970.00 -30.836  -72.775 4.083 8.422e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



A nested model

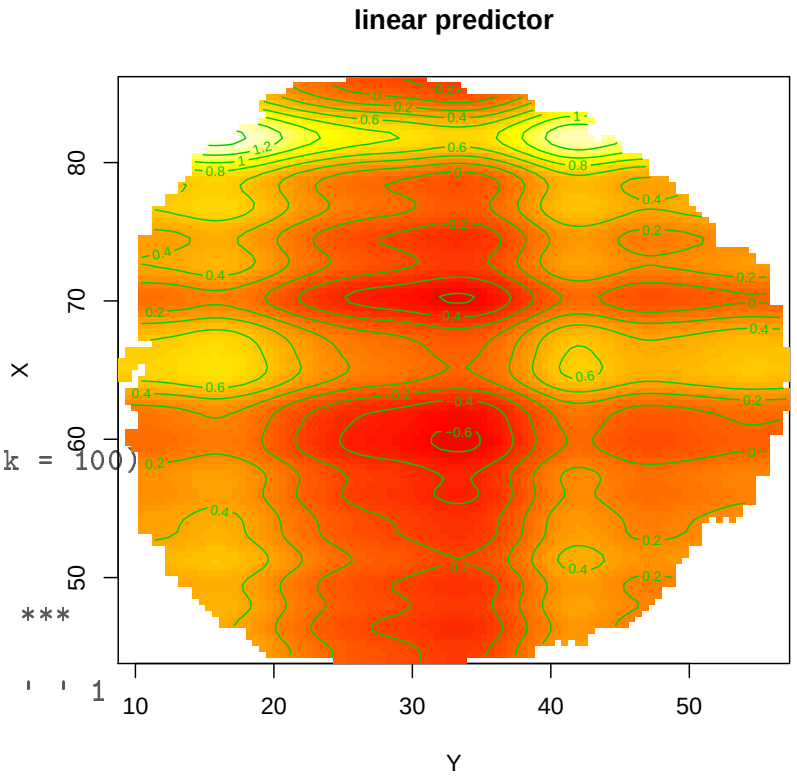
```
vis.gam(m3,plot.type="contour",too.far=0.02,n.grid=100)
```

```
m4=gam(medFPQ~s(Y,k=30)+s(X,k=30)+s(Y,X,k=100),data=brain,
       family=Gamma(link=log))
m4$gcv.ubre
```

```
   GCV.Cp
0.6175244
```

```
anova(m4,m3,test="F")
```

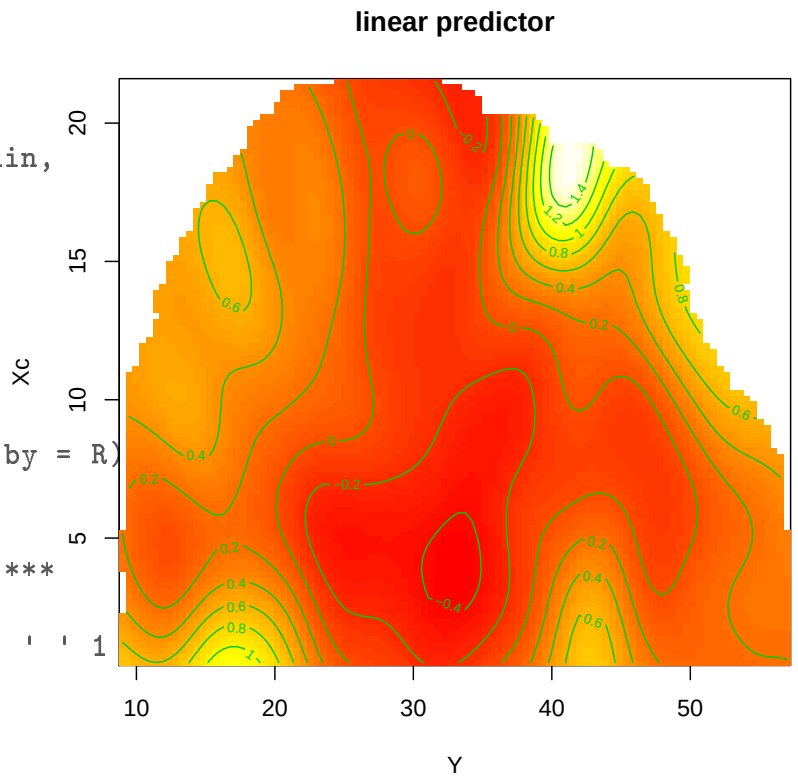
```
Analysis of Deviance Table

Model 1: medFPQ ~ s(Y, k = 30) + s(X, k = 30) + s(Y, X, k = 100)
Model 2: medFPQ ~ s(Y, k = 30) + s(X, k = 30)
  Resid. Df Resid. Dev      Df Deviance      F    Pr(>F)
1    1496.2      883.9
2    1533.2      970.0 -37.016    -86.1 4.1575 3.075e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Symmetry

```
vis.gam(ms1,plot.type="contour",too.far=0.04,n.grid=100)
```

```
brain$Xc=abs(brain$X-64.5)
brain$R=1*(brain$X<64.5)
ms1=gam(medFPQ~s(Y,Xc,k=100),data=brain,
        family=Gamma(link=log))
ms2=gam(medFPQ~s(Y,Xc,k=100)+s(Y,Xc,k=100,by=R),data=brain,
        family=Gamma(link=log))
anova(ms1,ms2,test="F")


Analysis of Deviance Table

Model 1: medFPQ ~ s(Y, Xc, k = 100)
Model 2: medFPQ ~ s(Y, Xc, k = 100) + s(Y, Xc, k = 100, by = R)
  Resid. Df Resid. Dev     Df Deviance      F    Pr(>F)
1    1511.6     948.08
2    1467.8     850.79 43.763   97.291 4.0972 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



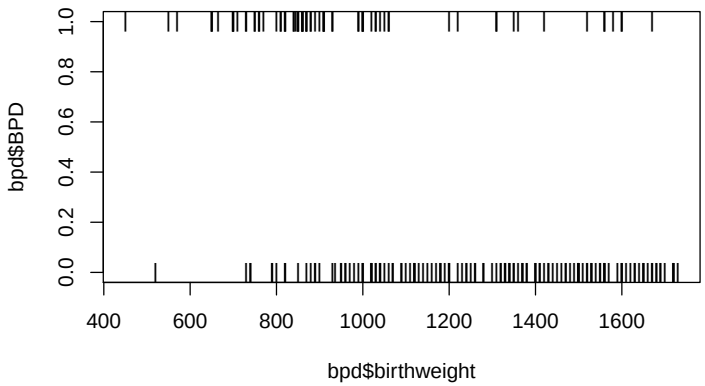
Indice

BPD data

Bronchopulmonary dysplasia (BPD) is a lung disease typical of premature babies, its presence may be related to birthweight.

For 223 babies we have observed

- birthweight
- presence of bronchopulmonary dysplasia (BPD)



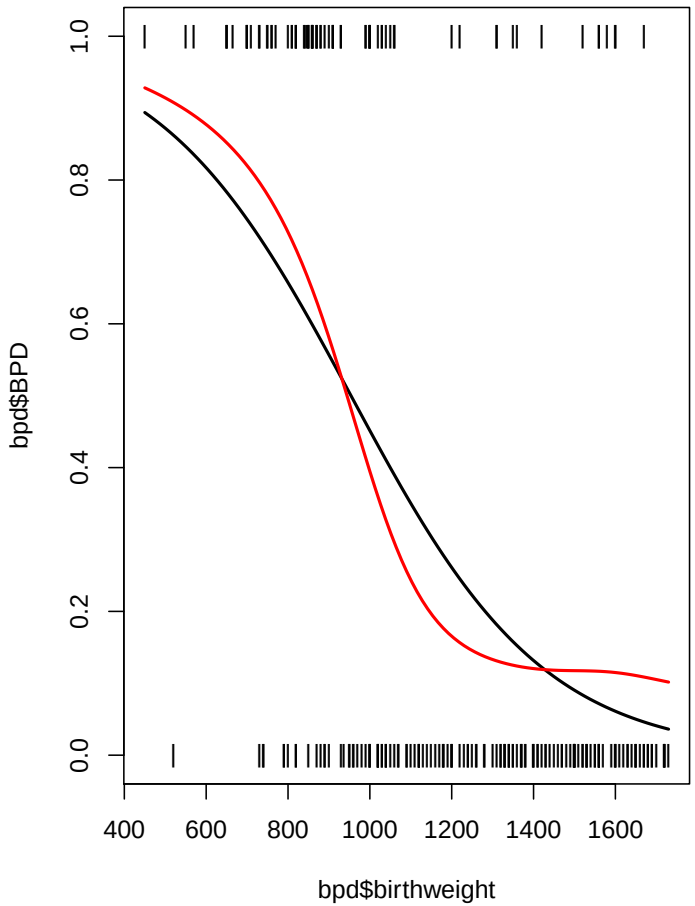
BPD data

The relationship between BPD and birthweight can be first examined through a GLM.

```
library(gamair)
data(bpd)
bpd[1,]
plot(bpd$birthweight,bpd$BPD,pch="|")
fit=gam(BPD~birthweight,data=bpd,family=binomial)
curve(predict(fit,newdata=data.frame(birthweight=x),
              type="response"),ad=TRUE,lwd=2)
```

... or, of course, we may use a non parametric model

```
fitS=gam(BPD~s(birthweight),data=bpd,
         family=binomial)
curve(predict(fitS,
              newdata=data.frame(birthweight=x),
              type="response"),ad=TRUE,lwd=2,col="red")
```



BPD data: kernel regression

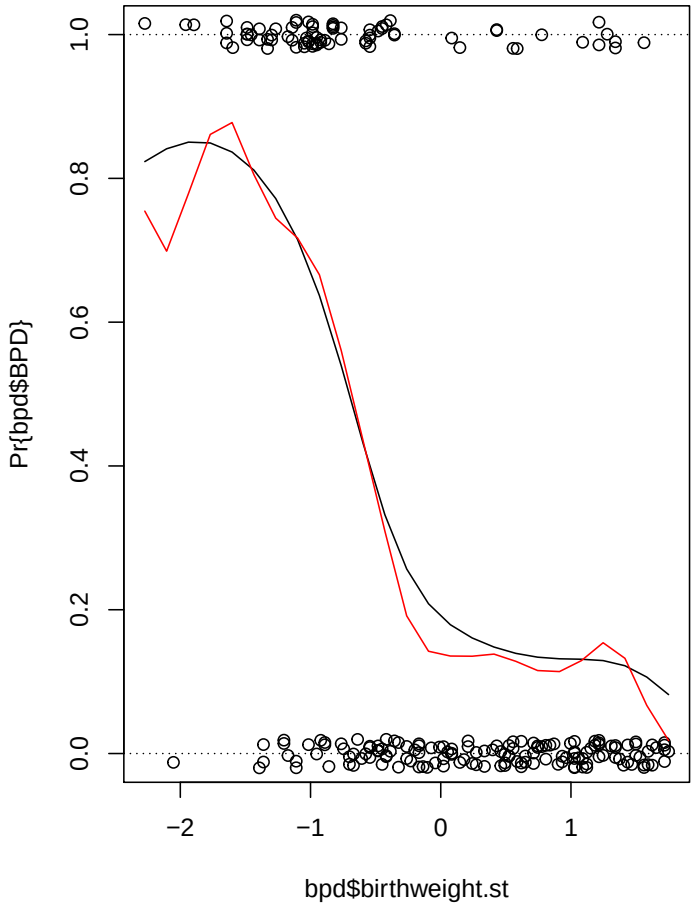
We can use the kernel regression approach

```
library(sm)
M=mean(bpd$birthweight)
S=sd(bpd$birthweight)
bpd$birthweight.st=(bpd$birthweight-M)/S
sm.binomial(bpd$birthweight.st,
            bpd$BPD,
            h=0.5)
```

Note that we need to standardize the covariate for numerical stability.

Note alsothat the smoothing parameter must be choosen, we try here an alternative value

```
sm.binomial(bpd$birthweight.st,
            bpd$BPD,
            h=0.25,add=TRUE,col="red",pch=NA)
```



Crestedlark

Indice

An application

Research Article

The Too-Much-Talent Effect: Team Interdependence Determines When More Talent Is Too Much or Not Enough

Roderick I. Swaab[1], Michael Schaerer[1], Eric M. Anicich[2], Richard Ronay[3], and Adam D. Galinsky[2]

[1]Organisational Behaviour Area, INSEAD, Fontainebleau, France; [2]Management Department, Columbia University; and [3]Department of Social and Organizational Psychology, VU University Amsterdam

Psychological Science
2014, Vol. 25(8) 1581–1591
DOI: 10.1177/0956797614537280

pss.sagepub.com

Scientific hypotheses

From the abstract:

Five studies examined the **relationship between talent and team performance**. Two survey studies found that **people believe there is a linear and nearly monotonic relationship between talent and performance**: Participants expected that more talent improves performance and that this relationship never turns negative. However, building off research on status conflicts, **we predicted that talent facilitates performance–bt only up to a point, after which the benefits of more talent decrease and eventually become detrimental as intrateam coordination suffers**.

Scientific hypotheses

From the abstract:

We also predicted that the **level of task interdependence is a key determinant** of when more talent is detrimental rather than beneficial. Three archival studies revealed that the **too-much-talent effect emerged when team members were interdependent (football and basketball) but not independent (baseball)**. Our basketball analysis also established the mediating role of team coordination. When teams need to come together, more talent can tear them apart.

Data and model

In order to check the validity of the above hypotheses data were collected for basket, baseball and soccer team on

- percentage of top players in a team (in a year) (t)

  (where top player is defined as being above a certain quantile of an appropriate ranking)
- performance of the team (in the year): percentage or number of wins
- control variables are also considered

A linear model for panel data is then considered where the variable t is included as

$$\beta_1 t + \beta_2 t^2$$

A conclusion on whether the relationship between percentage of top player is lienar or not is then drawn based on the significance of β_2 and the shape of the estimted curve

$$\hat{\beta}_1 t + \hat{\beta}_2 t^2$$

Results: NBA

Table 4. The Impact of Talent on Basketball Teams' Performance in Study 3 (n = 297)

Predictor	Model 1	Model 2	Model 3
Talent	0.35** (0.10)	1.61*** (0.42)	0.91* (0.43)
Talent-squared	—	–1.83** (0.56)	–1.23* (0.57)
Intrateam coordination	—	—	0.10*** (0.01)
Free-throw percentage	0.54 (0.29)	0.56 (0.27)	0.17 (0.25)
Roster size	–0.00 (0.00)	–0.00 (0.00)	–0.01* (0.00)
Games played	–0.00 (0.00)	–0.00 (0.00)	0.00 (0.00)
Lagged performance	0.32*** (0.05)	0.34*** (0.05)	0.22*** (0.05)
Intercept	–0.12 (0.27)	–0.30 (0.26)	0.28 (0.24)
R^2	.34	.38	.51
F	$F(5, 29) = 14.11$***	$F(6, 29) = 14.93$***	$F(7, 29) = 28.84$***

Note: Standard errors are reported in parentheses.
*p < .05. **p < .01. ***p < .001.

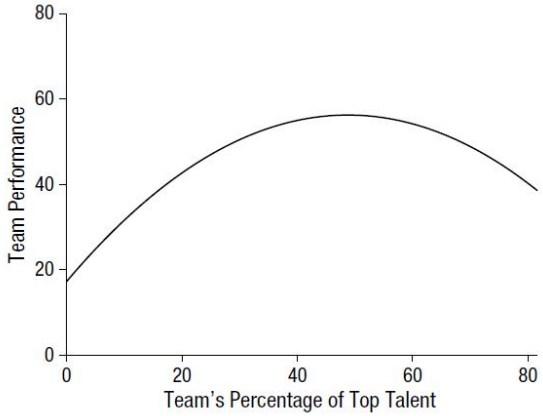


Fig. 3. Results from Study 3: team performance in the National Basketball Association from 2002 through 2012 as a function of the percentage of top talent on the team's roster.

Results: MLB

Table 7. The Impact of Talent on Baseball Teams' Performance in Study 4 (n = 300)

Predictor	Model 1	Model 2
Talent	0.41*** (0.03)	0.70*** (0.15)
Talent-squared	—	–0.42 (0.21)
Roster size	0.00 (0.00)	0.00 (0.00)
Games played	0.00 (0.01)	0.00 (0.01)
Lagged performance	0.21*** (0.05)	0.20*** (0.04)
Intercept	–0.33 (1.68)	–0.38 (1.71)
R^2	.60	.61
F	$F(4, 29) = 78.85$***	$F(5, 29) = 93.36$***

Note: Standard errors are reported in parentheses.
***$p < .001$.

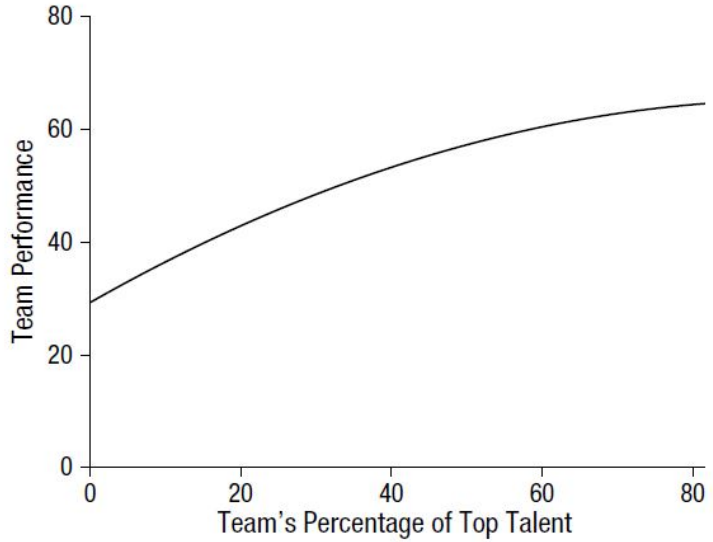
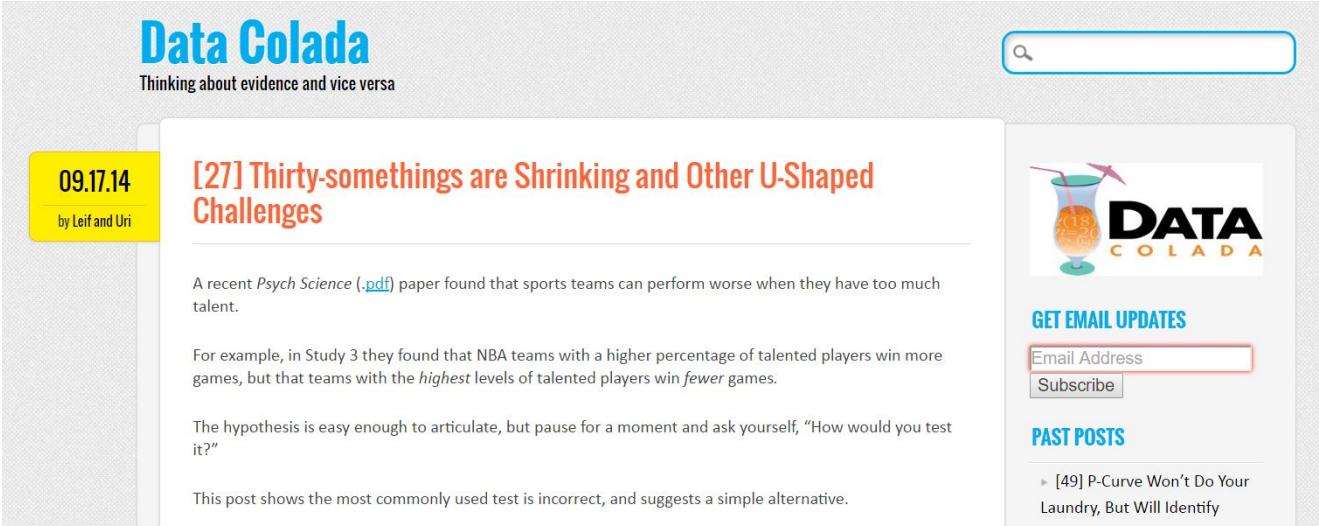


Fig. 6. Results from Study 4: team performance in Major League Baseball from 2002 through 2012 as a function of the percentage of top talent on the team's roster.

A criticism

A criticism on the paper was put forward by Leif Nelson and Uri Simonsohn



Data Colada

Thinking about evidence and vice versa

09.17.14

by Leif and Uri

[27] Thirty-somethings are Shrinking and Other U-Shaped Challenges

A recent *Psych Science* (.pdf) paper found that sports teams can perform worse when they have too much talent.

For example, in Study 3 they found that NBA teams with a higher percentage of talented players win more games, but that teams with the *highest* levels of talented players win *fewer* games.

The hypothesis is easy enough to articulate, but pause for a moment and ask yourself, "How would you test it?"

This post shows the most commonly used test is incorrect, and suggests a simple alternative.

DATA COLADA

GET EMAIL UPDATES

Email Address

Subscribe

PAST POSTS

[49] P-Curve Won't Do Your Laundry, But Will Identify

A criticism

A criticism on the paper was put forward by Leif Nelson and Uri Simonsohn

---

The main criticism was that it is dangerous to draw a conclusion on the sahpe of the relationship based on the fit of a quadratic curve,which is almost forced to suggest a *U*-shape.

Fig 2a. Age and Height

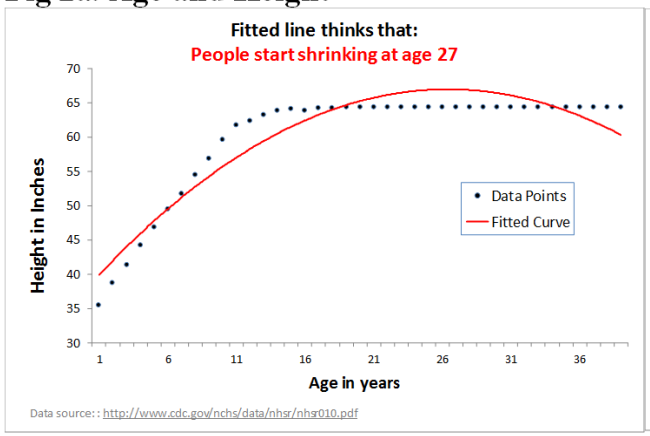
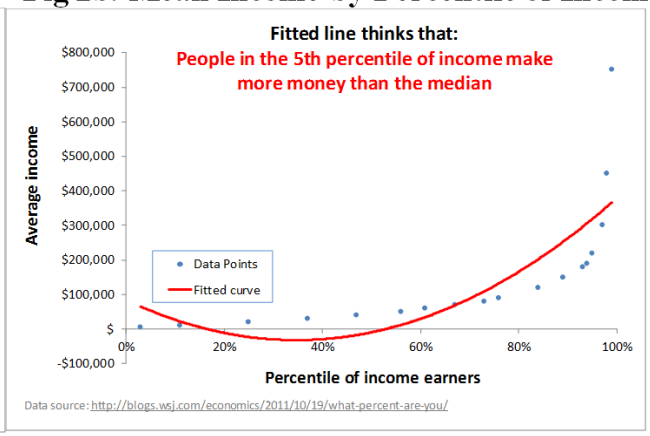


Fig 2b. Mean Income by Percentile of Income



A criticism

A criticism on the paper was put forward by Leif Nelson and Uri Simonsohn

---

The main criticism was that it is dangerous to draw a conclusion on the sahpe of the relationship based on the fit of a quadratic curve,which is almost forced to suggest a *U*-shape.

They suggest an alternative procedure

(1) Run the quadratic regression

(2) Find the point where the resulting u-shape maxes out.

(3) Now run a linear regression up to that point, and another from that point onwards.

(4) Test whether the second line is negative and significant.

A criticism

They suggest an alternative procedure

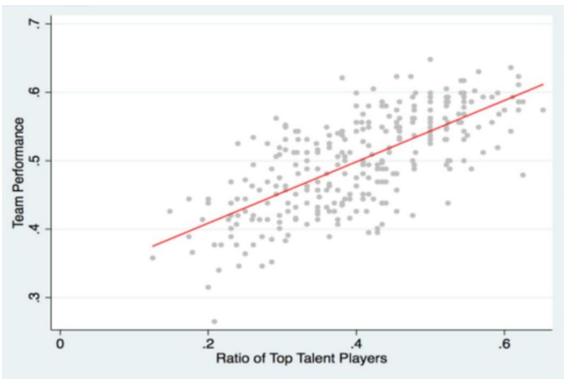
(1) Run the quadratic regression

(2) Find the point where the resulting u-shape maxes out.

(3) Now run a linear regression up to that point, and another from that point onwards.

(4) Test whether the second line is negative and significant.

The authors performed such an analysis obtaining



Baseball

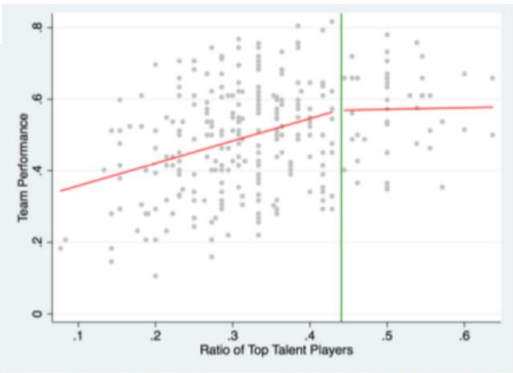


Figure 3b. NBA performance – top talent (33%). S&N test reveals that the first slope is significant and positive ($p \leq .001$) and that the second slope is not significant ($p = .48$).

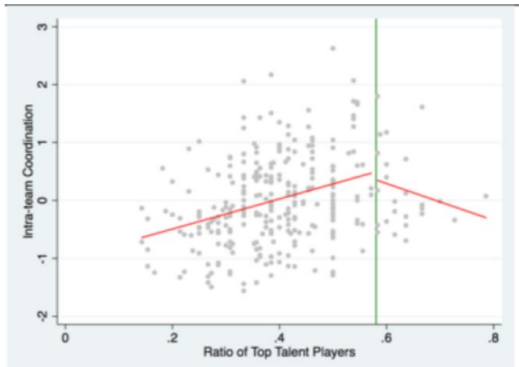
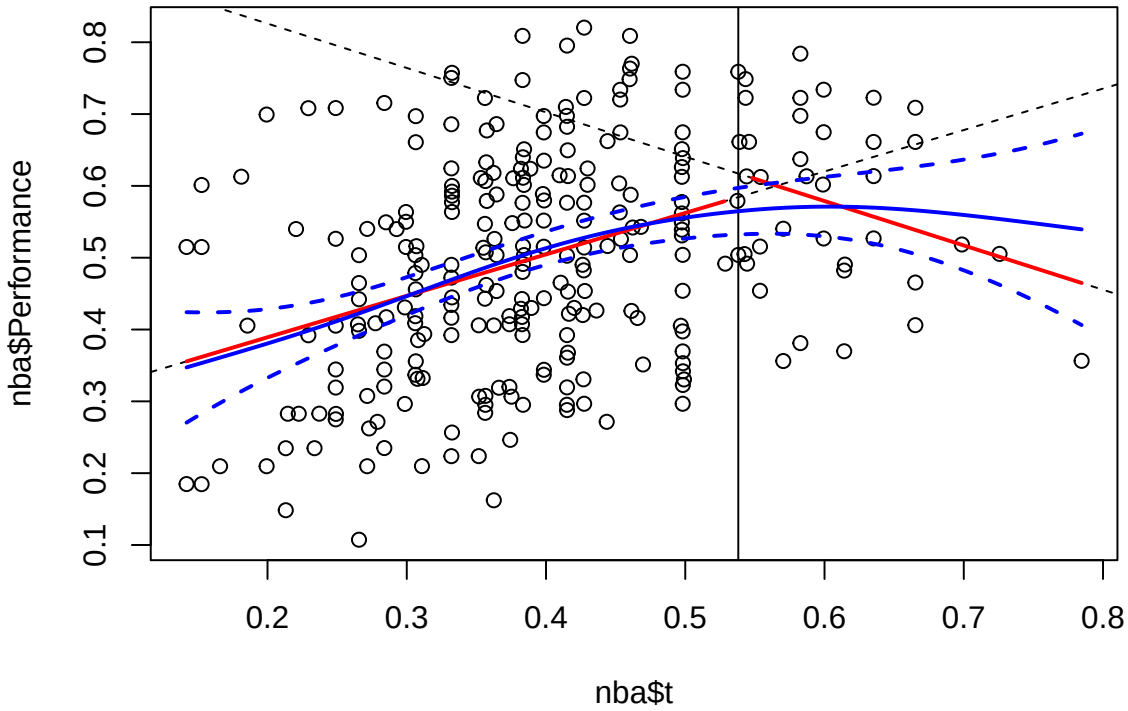
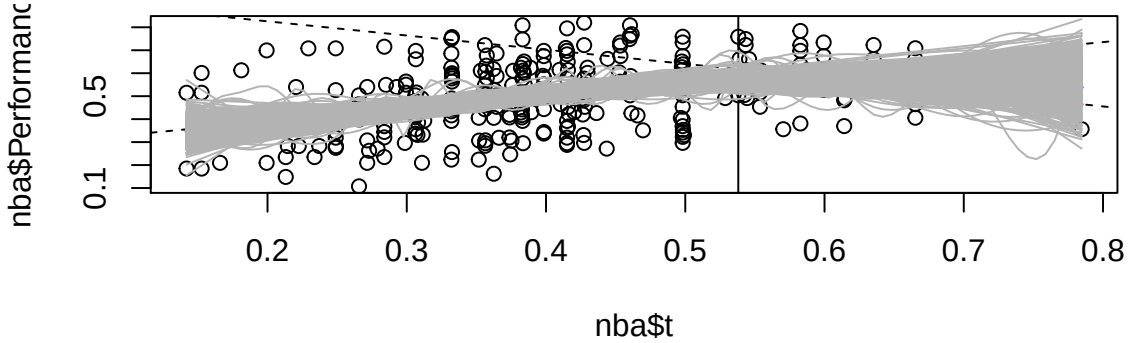


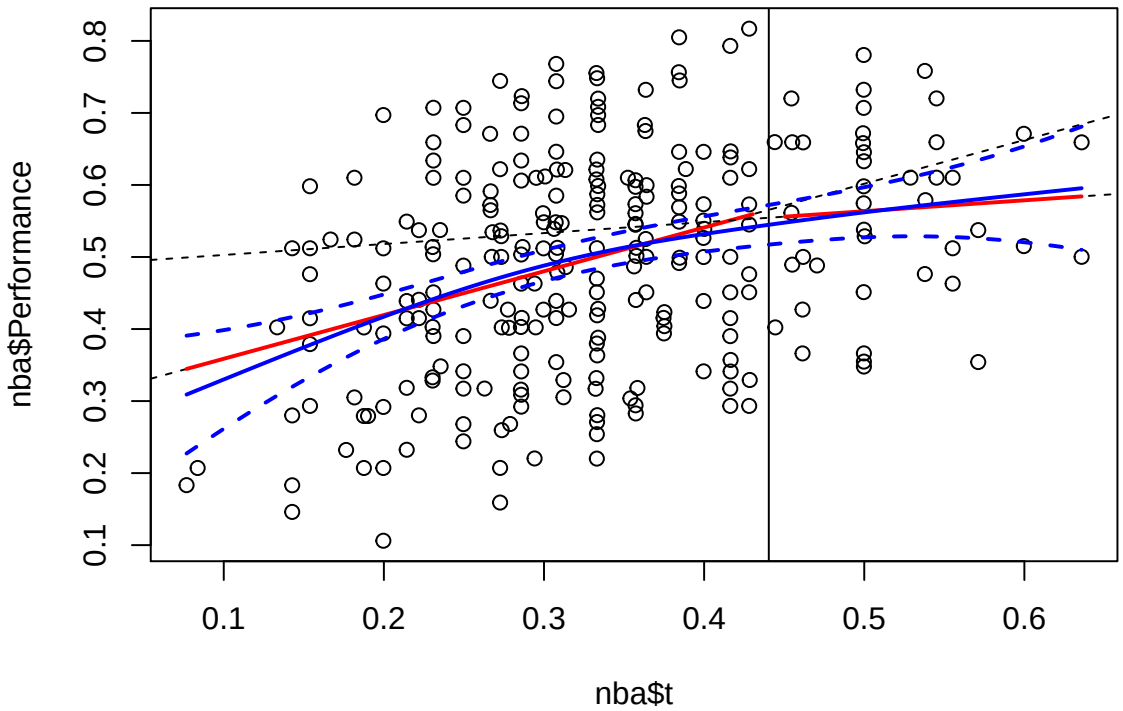
Figure 2c. NBA coordination – top talent (40%). S&N test reveals that the first slope is significant and positive ($p \leq .001$) and that the second slope is significant and negative ($p = .039$).

GAM analysis on the NBA data (more or less)





GAM analysis on the NBA data (more or less)



GAM analysis on the baseball data (more or less)