

Nonparametric Project of Agricultural Productivity in the U.S. Spatial Gam

Sofia Moroni*

2023-06-27

Contents

1 Model 1: PREDICT ON INTERACTION BETWEEN COORDINATES	2
2 Model 2: PREDICT ON INTERACTION BETWEEN COORDINATES AND POPULATION	6
3 Prediction	11

```
library(ISLR2)
```

```
## Warning: il pacchetto 'ISLR2' è stato creato con R versione 4.1.3
```

```
library(car)
```

```
## Warning: il pacchetto 'carData' è stato creato con R versione 4.1.3
```

```
library(sp)
library(mgcv)
library(rgl)
```

```
## Warning: il pacchetto 'rgl' è stato creato con R versione 4.1.3
```

```
library(splines)
library(pbapply)
```

```
## Warning: il pacchetto 'pbapply' è stato creato con R versione 4.1.3
```

```
library(devtools)
```

```
## Warning: il pacchetto 'devtools' è stato creato con R versione 4.1.3
```

```
## Warning: il pacchetto 'usethis' è stato creato con R versione 4.1.3
```

*sofia.moroni@mail.polimi.it

```

library(visreg)

## Warning: il pacchetto 'visreg' è stato creato con R versione 4.1.3

library(ggplot2)

## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.1.3

library(mgcViz)

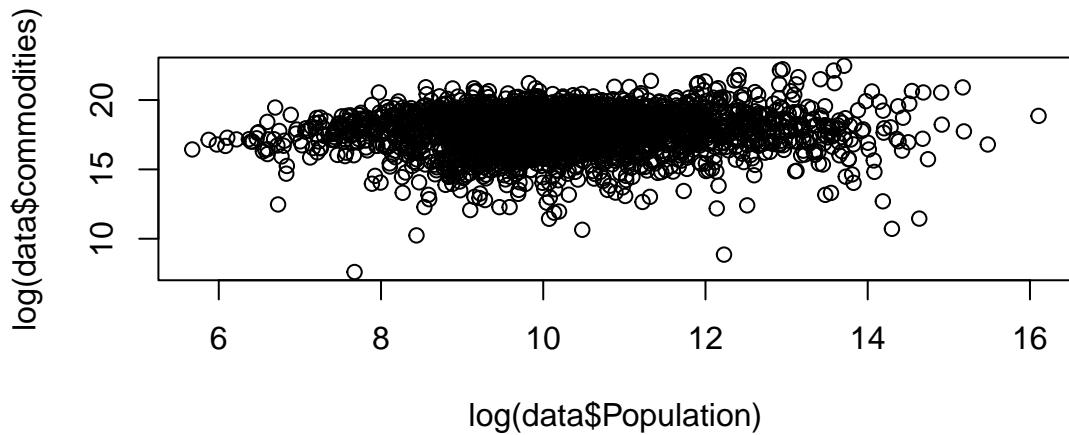
## Warning: il pacchetto 'mgcViz' è stato creato con R versione 4.1.3

## Warning: il pacchetto 'qgam' è stato creato con R versione 4.1.3

data_path = file.path('data')
output_path = file.path('results')
data =
  read.table(
    file.path(data_path, 'commodities_county_df.csv'),
    header = T,
    sep = ','
  )

plot(log(data$Population),log(data$commodities))

```



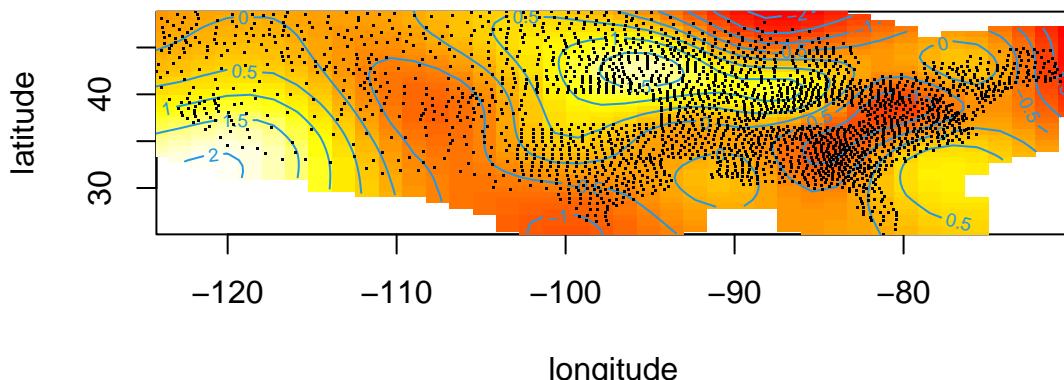
1 Model 1: PREDICT ON INTERACTION BETWEEN COORDINATES

```
mod2d <- mgcv::gam(log(commodities) ~ s(y,x), data = data, method = "REML")
summary(mod2d)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## log(commodities) ~ s(y, x)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.80467   0.02508 709.9 <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(y,x) 27.9  28.92 32.63 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.26  Deviance explained = 26.8%
## -REML = 4524.6  Scale est. = 1.6752    n = 2663
```

```
par(mfrow=c(1,1))
plot(mod2d,scheme=2,pages=0,select = 1,
     main = "Smooth coefficients of interaction",
     xlab = "longitude",ylab = "latitude")
```

Smooth coefficients of interaction

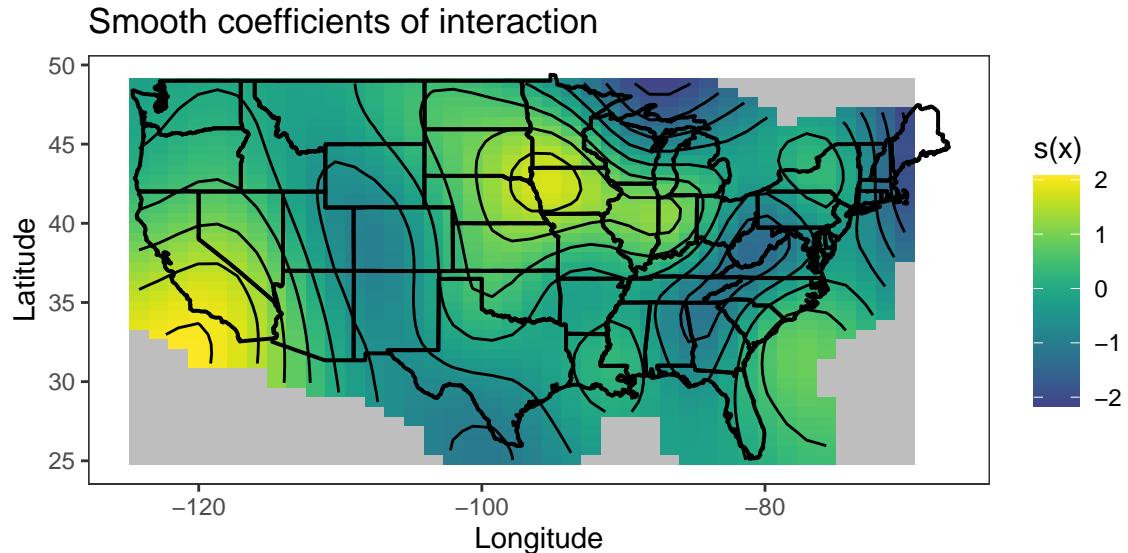


```
b <- getViz(mod2d)
pl <- plot(sm(b, 1)) + l_fitRaster() + l_fitContour() +
```

```

geom_polygon(data = map_data ("state"),
             aes(x=long, y = lat,group=group),
             fill=NA,color="black",lwd = 0.7,inherit.aes = F) +
ggtitle("Smooth coefficients of interaction") +
xlab("Longitude") + ylab("Latitude")
pl

```



```

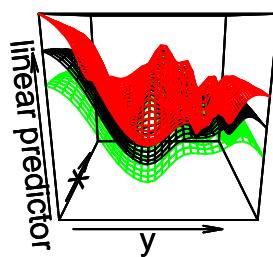
x_grid <- map_data ("state")$long
y_grid <- map_data ("state")$lat

```

```

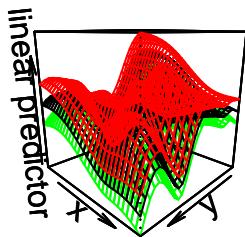
# Make the perspective plot with error surfaces
vis.gam(mod2d, view = c("y", "x"),
        plot.type = "persp", se = 2)

```



red/green are ± 2 s.e.

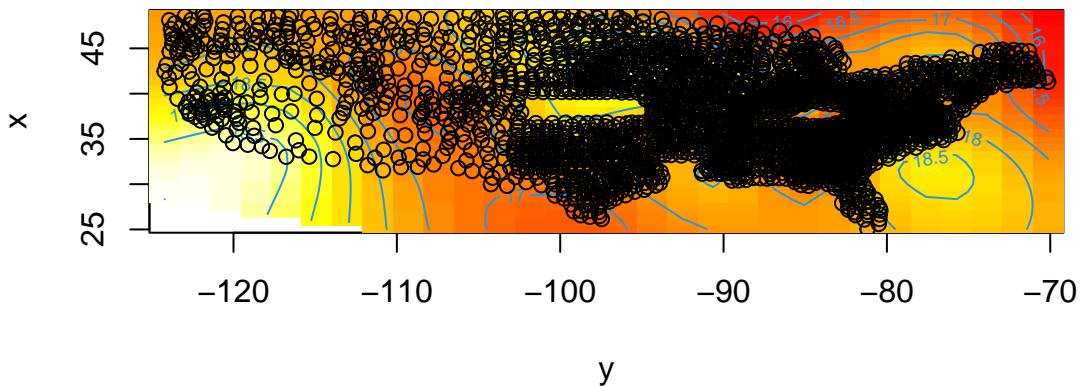
```
# Rotate the same plot
vis.gam(mod2d, view = c("y", "x"),
        plot.type = "persp", se = 2, theta = 135)
```



red/green are ± 2 s.e.

```
# Make plot with 10% extrapolation
vis.gam(mod2d, view = c("y", "x"),
         plot.type = "contour", too.far = 0.25)
dat1 <- data
coordinates(dat1) <- c("y", "x")
points(dat1)
```

linear predictor



2 Model 2: PREDICT ON INTERACTION BETWEEN COORDINATES AND POPULATION

```

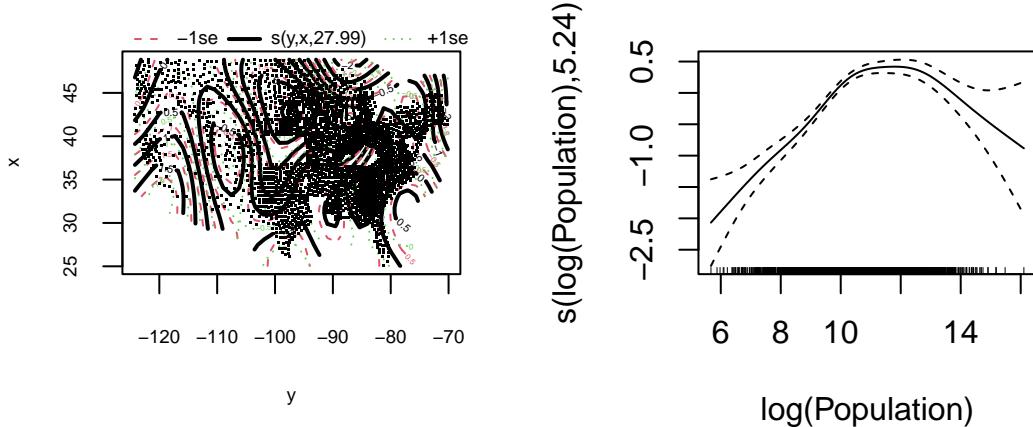
tensor_mod2 <- gam(log(commodities) ~ s(y, x)+s(log(Population)),
                     data = data, method = "REML")

# Summarize and plot
summary(tensor_mod2)

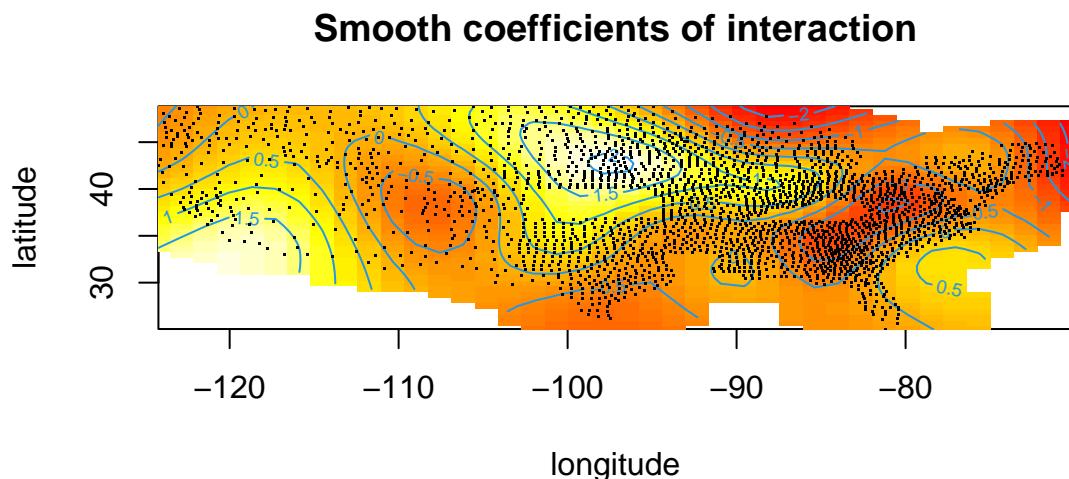
## 
## Family: gaussian
## Link function: identity
##
## Formula:
## log(commodities) ~ s(y, x) + s(log(Population))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.80467   0.02406    740   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(y,x)      27.987  28.93 42.02 <2e-16 ***
## s(log(Population)) 5.236   6.44 35.52 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.319  Deviance explained = 32.8%
## -REML = 4422.6  Scale est. = 1.5418   n = 2663

plot(tensor_mod2,pages=1)

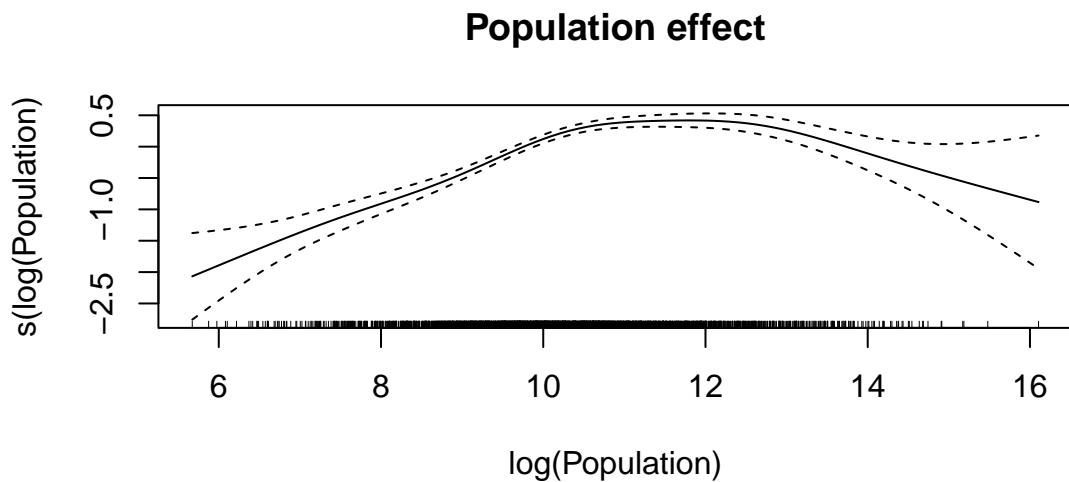
```



```
plot(tensor_mod2,scheme=2,pages=0,select = 1,
      main = "Smooth coefficients of interaction",
      xlab = "longitude",ylab = "latitude")
```

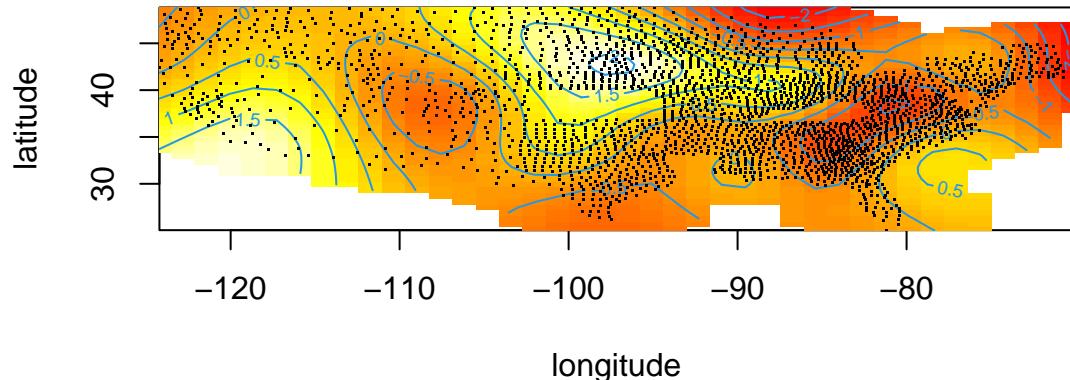


```
plot(tensor_mod2,scheme=2,pages=0,select = 2,
      main = "Population effect", ylab = "s(log(Population))")
```

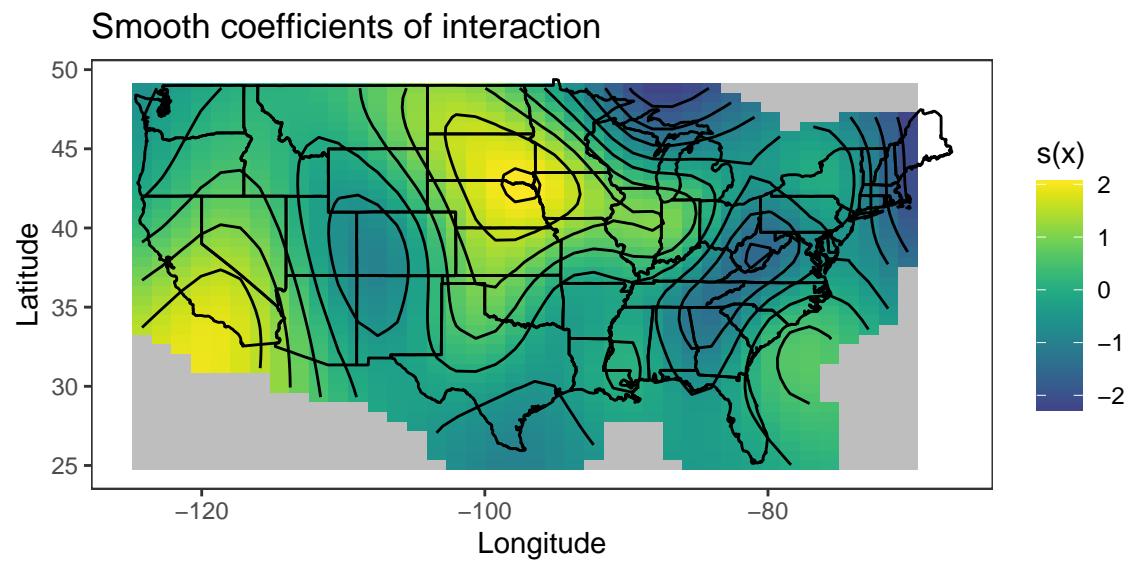


```
par(mfrow=c(1,1))
plot(tensor_mod2,scheme=2,pages=0,select = 1,
      main = "Smooth coefficients of interaction",
      xlab = "longitude",ylab = "latitude")
```

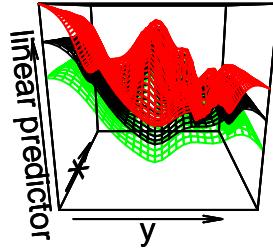
Smooth coefficients of interaction



```
b <- getViz(tensor_mod2)
pl <- plot(sm(b, 1)) + l_fitRaster() + l_fitContour() +
  geom_polygon(data = map_data ("state"),
    aes(x=long, y = lat,group=group),
    fill=NA,color="black",lwd = 0.5,inherit.aes = F) +
  ggtitle("Smooth coefficients of interaction") +
  xlab("Longitude") + ylab("Latitude")
pl
```

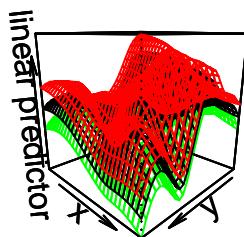


```
vis.gam(tensor_mod2, view = c("y", "x"),
  plot.type = "persp", se = 2)
```



red/green are ± 2 s.e.

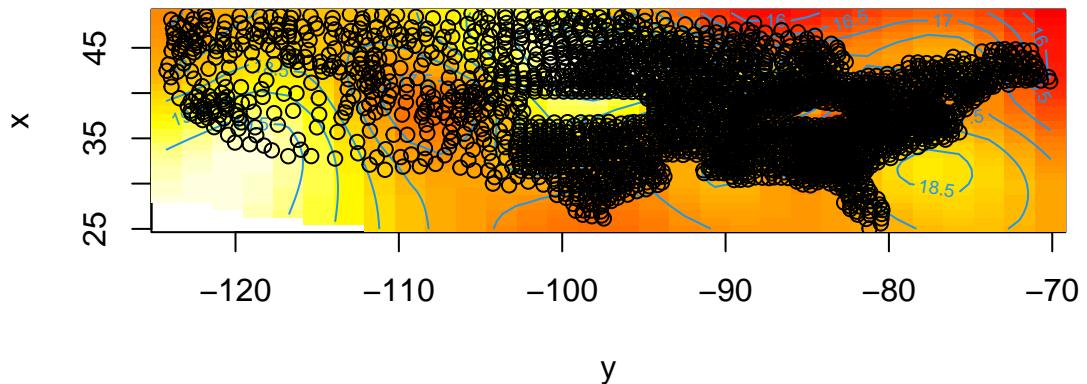
```
# Rotate the same plot
vis.gam(tensor_mod2, view = c("y", "x"),
         plot.type = "persp", se = 2, theta = 135)
```



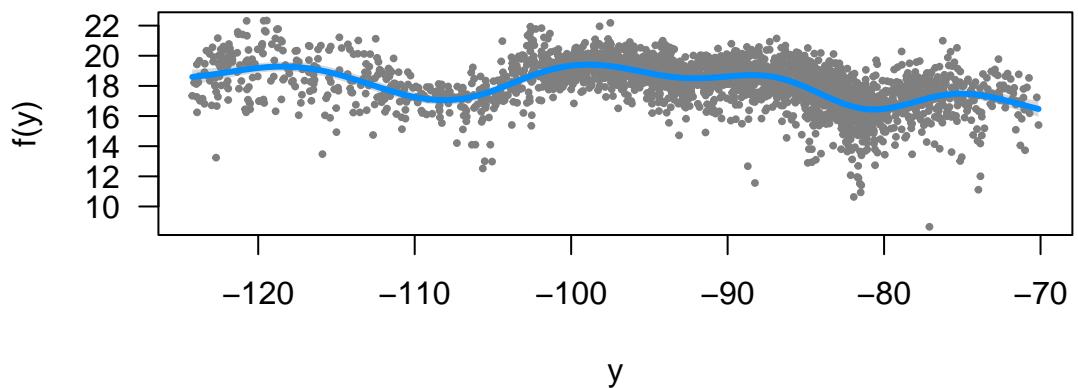
red/green are ± 2 s.e.

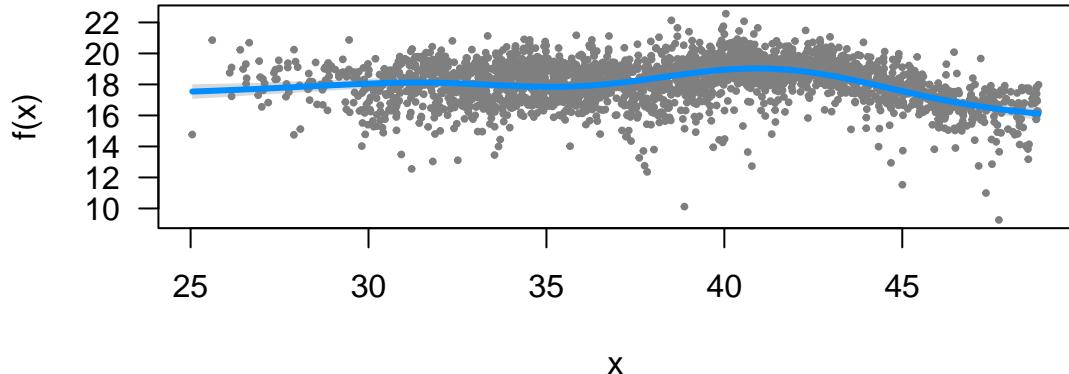
```
# Make plot with 10% extrapolation
vis.gam(tensor_mod2, view = c("y", "x"),
         plot.type = "contour", too.far = 0.25)
points(dat1)
```

linear predictor



```
visreg(tensor_mod2)
```





3 Prediction

Now I try to predict the sales of Kentucky

```
x_grid <- seq(-89,-81, by = 0.01)
y_grid <- seq(36,40, by = 0.01)
grid <- expand.grid(x_grid,y_grid)
grid <- data.frame(x=grid$Var2, y=grid$Var1)
preds <- predict(mod2d,newdata=grid)
#preds
df <- cbind(grid,"log(Sales)" =as.data.frame(preds))
ggplot(as.data.frame(df), aes(y,x, col=preds)) +
  geom_point(alpha=0.5) +
  scale_color_gradient(low="yellow", high="red") +
  coord_cartesian(xlim=c(-89, -81), ylim = c(36, 40))+
  geom_polygon( data=map_data("county"), aes(x=long, y=lat, group=group),
                color="black", fill="lightblue", alpha=0.1 ) +
  guides(color = guide_legend(title = "log(Sales)"))
```

