Lesson 26

Clark

Note that some of the examples from today are from https://m-clark.github.io/generalized-additive-models/

Sometimes it's been said that money doesn't buy happiness. Since I don't know how to meaningfully measure happiness, let's just make the (probably valid) assumption that the more you know about science the happier you will be!

To analyze this data perhaps we can take some data from the Programme for International Student Assessment along a few other indexies.

```
library(tidyverse)
library(GGally)
library(mgcv)
library(faraway)
dat <- read.csv(paste0("https://raw.githubusercontent.com/m-clark/",
          "generalized-additive-models/master/data/pisasci2006.csv"))

pisa = dat %>% select(-c(Issues,Explain,Evidence,Country))

pisa %>% str()
```

```
## 'data.frame':    65 obs. of  7 variables:
##  $ Overall : int  NA 391 527 511 382 510 390 434 534 438 ...
##  $ Interest: int  NA 567 465 507 612 503 592 523 469 591 ...
##  $ Support : int  NA 506 487 515 542 492 519 527 501 564 ...
##  $ Income  : num  0.599 0.678 0.826 0.835 0.566 0.831 0.637 0.663 0.84 0.673 ...
##  $ Health  : num  0.886 0.868 0.965 0.944 0.78 0.935 0.818 0.829 0.951 0.923 ...
##  $ Edu     : num  0.716 0.786 0.978 0.824 NA 0.868 0.646 0.778 0.902 0.764 ...
##  $ HDI     : num  0.724 0.773 0.92 0.866 NA 0.877 0.695 0.753 0.897 0.78 ...
```

Here Overall is the average science score for 15 year olds, interest is a measure of interest in science, support for scientific inquiry, an income index, a health index, and the Human Development Index.

We can explore

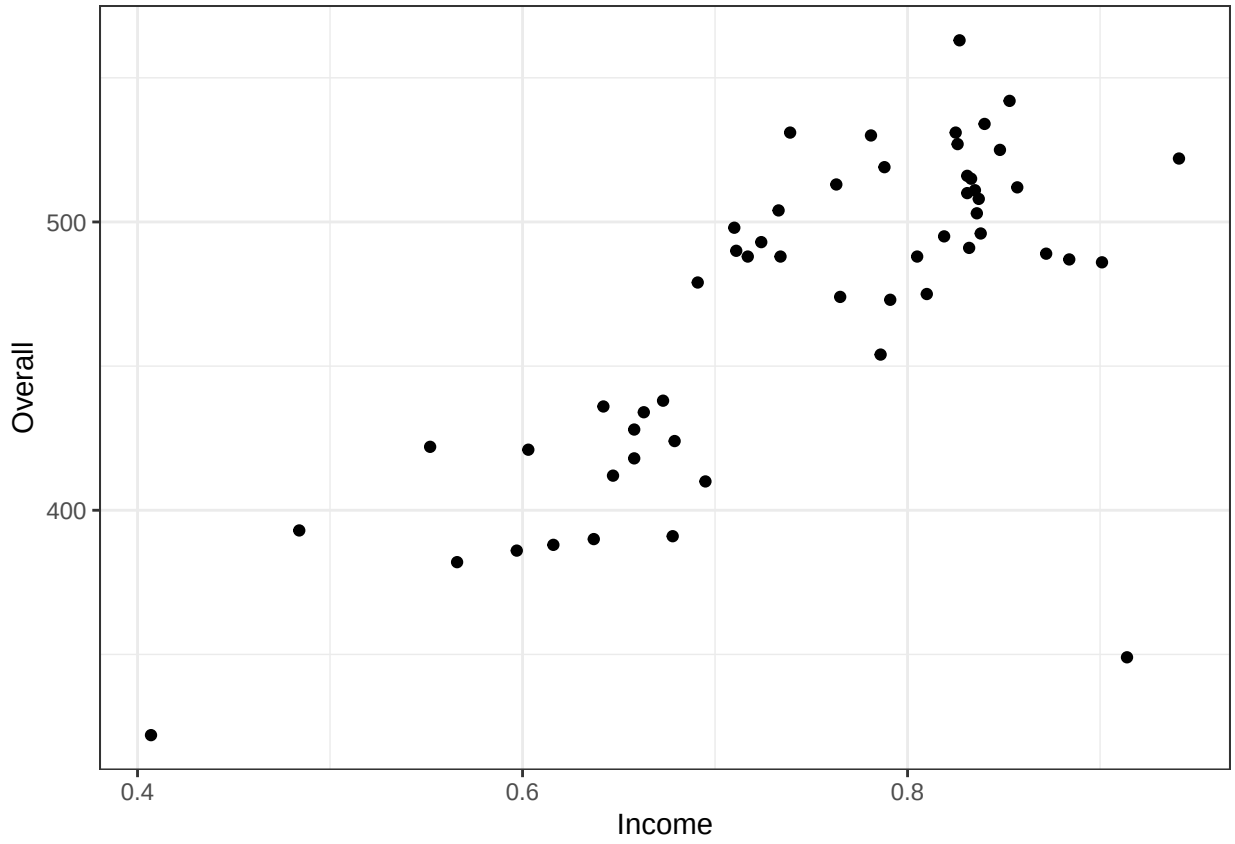
```
# Function to return points and geom_smooth
# allow for the method to be changed
my_fn <- function(data, mapping, method="loess", ...){
  p <- ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method=method, se=FALSE)
  p
}

pisa  %>% ggpairs(upper = list(continuous = wrap("cor", size = 3)),
                  lower = list(continuous =my_fn)) + theme_bw()
```

While we might start with a linear model, perhaps there are some concerns.

```
pisa %>% ggplot(aes(x=Income,y=Overall)) +
  geom_point() + theme_bw()
```

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```



Perhaps we want a model that allows for additional flexibility inside of our linear predictor. That is:

Now, this begs a lot of questions. How do we choose our $f_i()$ functions? How do we fit such a model? How do we conduct inference on this?

Well, if we are given data, one criterion we may use for a linear model is a penalized sum of squares. That is, we want to find α and $f_i()$ terms that minimize:

The λ term here controls the amount of 'wiggliness' of the fitted model. If $\lambda \to \infty$ then we want our second derivative to be zero (which would result in a linear f_i function, or, in other words $x\beta$). If $\lambda \to 0$ then we can fit models that perfectly interpolate our data. Typically, λ is selected via cross-validation.

In general, in can be shown, that a unique minimizer for the PRSS is an additive cubic spline model. That is, each of the functions $f_j(.)$ are cubic splines with knots at each of the unique values of x_{ij}.

Just as a quick example, the following basis functions would represent a cubic spline with two knots:

Without going too far down this rabbit hole, if we don't want to select knots we could consider a cubic smoothing spline can be written as

$$f(x) = \sum_{j=1}^{N} N_j(x)\theta_j$$

Here there are knots placed on each data point. $N_j(x)$ are a set of basis functions representing the spline and θ_j gives the weights of the basis function. The θ values are shrunk similar to a ridge regression based on the value of λ chosen.

These are adjusted slightly to become natural cubic splines which are essentially regressed (with a ridge type regularization) on the data to yield an estimate of $f(x)$

We can fit cubic splines via:

```
mod_gam1 = gam(Overall ~ s(Income, bs = "cr"), data = pisa)
summary(mod_gam1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Overall ~ s(Income, bs = "cr")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  470.444      4.082   115.3   <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(Income) 6.895  7.741 16.67  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =    0.7   Deviance explained = 73.9%
```

```
## GCV = 1053.7  Scale est. = 899.67    n = 54
```

Which we can visualize here:

```
library(mgcViz)
```

```
## Warning: package 'mgcViz' was built under R version 4.3.3
```

```
## Loading required package: qgam
```

```
## Warning: package 'qgam' was built under R version 4.3.3
```

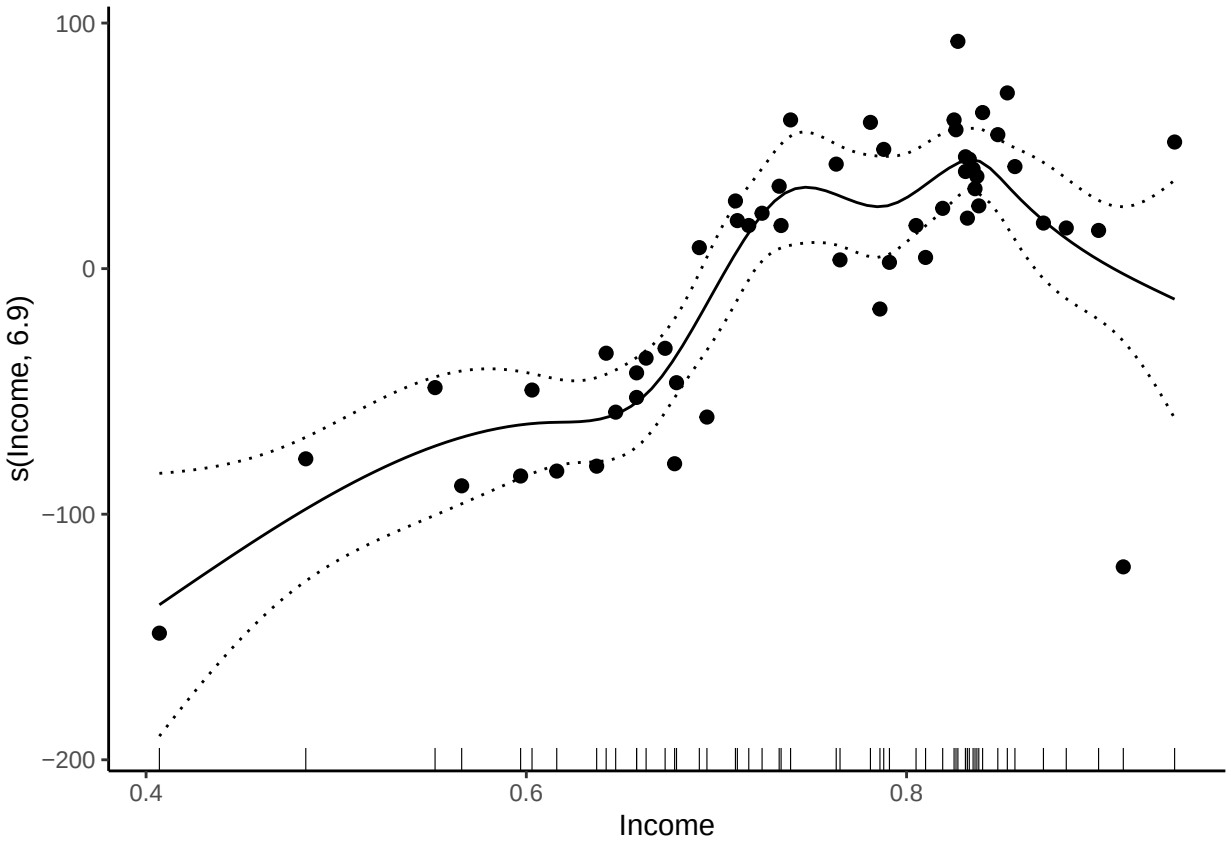
```
## Registered S3 method overwritten by 'mgcViz':
##   method from
##   +.gg   GGally
```

```
##
## Attaching package: 'mgcViz'
```

```
## The following objects are masked from 'package:stats':
##
##     qqline, qqnorm, qqplot
```

```
viz <- getViz(mod_gam1)

viz %>% plot() +
   l_points(shape=19,size=2) +
  l_fitLine(linetype = 1)  +
  l_ciLine(linetype = 3) +
  l_ciBar() +
  l_rug() +
  theme_classic()
```



One thing we note with the summary is we obtain p values for income. This is calculated comparing the deviance of the fitted model with the deviance of the null model after adjusting for loss of (effective) degrees of freedom. The degrees of freedom are estimated through looking at the smoothing matrix and in general a higher effective degrees of freedom means a more complex model.

GAMs can be compared to other models through AIC and can be compared to nested models via anova test.

More terms can be added by considering applying a cubic smoothing spline on the residuals. That is:

```
mod_gam2 = gam(Overall ~ s(Income) + s(Edu) + s(Health), data = pisa)
summary(mod_gam2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Overall ~ s(Income) + s(Edu) + s(Health)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  471.154      2.772     170   <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(Income) 7.593  8.415 8.826 1.29e-06 ***
## s(Edu)    6.204  7.178 3.308  0.00771 **
## s(Health) 1.000  1.000 2.736  0.10679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.863   Deviance explained = 90.3%
## GCV = 573.83  Scale est. = 399.5     n = 52
```

Here we can see:

```
viz <- getViz(mod_gam2)

viz %>% plot() +
   l_points(shape=19,size=2) +
  l_fitLine(linetype = 1)  +
  l_ciLine(linetype = 3) +
  l_ciBar() +
  l_rug() +
  theme_classic()
```

Looking at `Health` it looks like the spline isn't doing much, so maybe a linear fit is better.

```
mod_gam2B = update(mod_gam2, . ~ . - s(Health) + Health)
summary(mod_gam2B)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Overall ~ s(Income) + s(Edu) + Health
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    640.3      102.3   6.260 3.06e-07 ***
## Health        -189.5      114.6  -1.654    0.107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df     F  p-value
## s(Income) 7.593  8.415 8.826 1.29e-06 ***
## s(Edu)    6.204  7.178 3.308  0.00771 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.863   Deviance explained = 90.3%
## GCV = 573.83  Scale est. = 399.5     n = 52
```

The framework from above can be extended to a generalized version of the additive model via:

The algorithm for fitting the model is:

```
data(ozone)
gammgcv <- gam(O3 ~ s(temp) + s(ibh) + s(ibt), family=poisson, scale=-1, data=ozone)
summary(gammgcv)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## O3 ~ s(temp) + s(ibh) + s(ibt)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.29269    0.02305   99.45   <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(temp) 3.816  4.736 16.803  <2e-16 ***
## s(ibh)  3.737  4.568 10.594  <2e-16 ***
## s(ibt)  1.348  1.623  0.238   0.651
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.712   Deviance explained = 72.9%
## GCV = 1.5062  Scale est. = 1.4585    n = 330
```

```
#predict(gammgcv,type="response")
#plot(gammgcv,residuals=TRUE,select=1)
```

Let's talk through this here:

The `pim` dataset consists of 768 female Pima Indians. We want to predict the diabets test result from the other predictors.

- Take a random sample of size 100, make this your test test.
- Fit a GLM on your training set, evaluate it on your test set.
- Fit a GAM on your training set, evaluate it on your test set.
- Which model do you prefer?