STAT 716 - Class 9/10 - 2018-11-05

Vitaly Druker

Non Linearity

Polynomial regression

global function - this can be an issue - especially at the outer edges

Step Functions

Step Functions localize everything, cut points categorical dummy variables

Basis functions

general funcionts

Regression Splines

piecewise polinomial regression

show page 272

piecwise cubic regression where to put knots?

spline - continuous on second derivative

Natural splines

the function is required to be linear at the boundary (in the region where X is smaller than the smallest knot, or larger than the largest knot).

Generalized Additive Models

We have discussed regular linear regression. We can standarize into GAM with the following equation:

$$y_i = \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i$$

Lab

```r
library(ISLR)
first_fit <- lm(wage ~ poly(age, 4), data = Wage)

coef(summary(first_fit))
```

```
##                  Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)    111.70361  0.7287409 153.283015 0.000000e+00
## poly(age, 4)1  447.06785 39.9147851  11.200558 1.484604e-28
## poly(age, 4)2 -478.31581 39.9147851 -11.983424 2.355831e-32
## poly(age, 4)3  125.52169 39.9147851   3.144742 1.678622e-03
## poly(age, 4)4  -77.91118 39.9147851  -1.951938 5.103865e-02
```

```
coef(summary(lm(wage ~age + I(age^2) + I(age^3) + I(age^4),
                data = Wage)))
```

```
##                  Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept) -1.841542e+02 6.004038e+01 -3.067172 0.0021802539
## age          2.124552e+01 5.886748e+00  3.609042 0.0003123618
## I(age^2)    -5.638593e-01 2.061083e-01 -2.735743 0.0062606446
## I(age^3)     6.810688e-03 3.065931e-03  2.221409 0.0263977518
## I(age^4)    -3.203830e-05 1.641359e-05 -1.951938 0.0510386498
```

Why don't the match? See here for an example: https://stackoverflow.com/questions/29999900/poly-in-lm-difference-between-raw-vs-orthogonal

```
coef(summary(lm(wage ~poly(age, 4, raw = T),
                data = Wage)))
```

```
##                            Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept)           -1.841542e+02 6.004038e+01 -3.067172 0.0021802539
## poly(age, 4, raw = T)1  2.124552e+01 5.886748e+00  3.609042 0.0003123618
## poly(age, 4, raw = T)2 -5.638593e-01 2.061083e-01 -2.735743 0.0062606446
## poly(age, 4, raw = T)3  6.810688e-03 3.065931e-03  2.221409 0.0263977518
## poly(age, 4, raw = T)4 -3.203830e-05 1.641359e-05 -1.951938 0.0510386498
```

Cuts

```
coef(summary(lm(wage ~cut(age, 4),
                data = Wage)))
```

```
##                          Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)             94.158392   1.476069 63.789970 0.000000e+00
## cut(age, 4)(33.5,49]    24.053491   1.829431 13.148074 1.982315e-38
## cut(age, 4)(49,64.5]    23.664559   2.067958 11.443444 1.040750e-29
## cut(age, 4)(64.5,80.1]   7.640592   4.987424  1.531972 1.256350e-01
```

Splines

```
library(splines)
library(dplyr)

# cubic splines

lm(wage~bs(age,knots=c(25,40,60)),data=Wage) %>%
  coef
```

```
##                      (Intercept) bs(age, knots = c(25, 40, 60))1
##                         60.49371                         3.98050
## bs(age, knots = c(25, 40, 60))2 bs(age, knots = c(25, 40, 60))3
```

```
##                          44.63098                         62.83879
## bs(age, knots = c(25, 40, 60))4 bs(age, knots = c(25, 40, 60))5
##                          55.99083                         50.68810
## bs(age, knots = c(25, 40, 60))6
##                          16.60614
```

We have 3 knots - how many degrees of freedom?

If it's piecewise cubic how many degrees of freedom? $4 + 4 + 4 + 4 = 16$

What do splines add? 1. Continuity 2. Continuity on 1st derivative 3. Continuiuty on 2nd derivative

Each constraint is a degree of freedom

remove $3 + 3 + 3 = 9$

Remaining $= 7$ which is what we see

```r
lm(wage~bs(age, df = 6),data=Wage) %>%
  coef
```

```
##      (Intercept) bs(age, df = 6)1 bs(age, df = 6)2 bs(age, df = 6)3
##         56.31384         27.82400         54.06255         65.82839
## bs(age, df = 6)4 bs(age, df = 6)5 bs(age, df = 6)6
##         55.81273         72.13147         14.75088
```

```r
attributes(bs(Wage$age, df = 6))
```

```
## $dim
## [1] 3000    6
##
## $dimnames
## $dimnames[[1]]
## NULL
##
## $dimnames[[2]]
## [1] "1" "2" "3" "4" "5" "6"
##
##
## $degree
## [1] 3
##
## $knots
##   25%   50%   75%
## 33.75 42.00 51.00
##
## $Boundary.knots
## [1] 18 80
##
## $intercept
## [1] FALSE
##
## $class
## [1] "bs"      "basis"  "matrix"
```

```r
lm(wage~bs(age),data=Wage) %>%
  coef
```

```
## (Intercept)    bs(age)1    bs(age)2    bs(age)3
```

```
##    58.68868   102.64368    48.76204    40.80330
```

```r
lm(wage~ns(age, knots=c(25,40,60)),data=Wage) %>% coef() %>% length()
```

```
## [1] 5
```

```r
lm(wage~bs(age, knots=c(25,40,60)),data=Wage) %>% coef() %>% length()
```

```
## [1] 7
```

```r
lm(wage~ns(age, knots=c(25,40,60)),data=Wage) %>% coef() %>% length()
```

```
## [1] 5
```

How many degrees of freedom do we get from add the natural spline contraint

only 2 degrees of freedom on the ends

- 4 degrees of freedom

2 + 4 + 4 + 2

- 2 - 3 - 2

```r
2 + 4 + 4 + 2 +
  - 2 - 3 - 2
```

```
## [1] 5
```

```r
ns_mod <- lm(wage~ns(age, df = 6),data=Wage)
bs_mod <- lm(wage~bs(age, df = 6),data=Wage)
poly_mod <- lm(wage~poly(age,6),data=Wage)

length(coef(ns_mod))
```

```
## [1] 7
```

```r
length(coef(bs_mod))
```

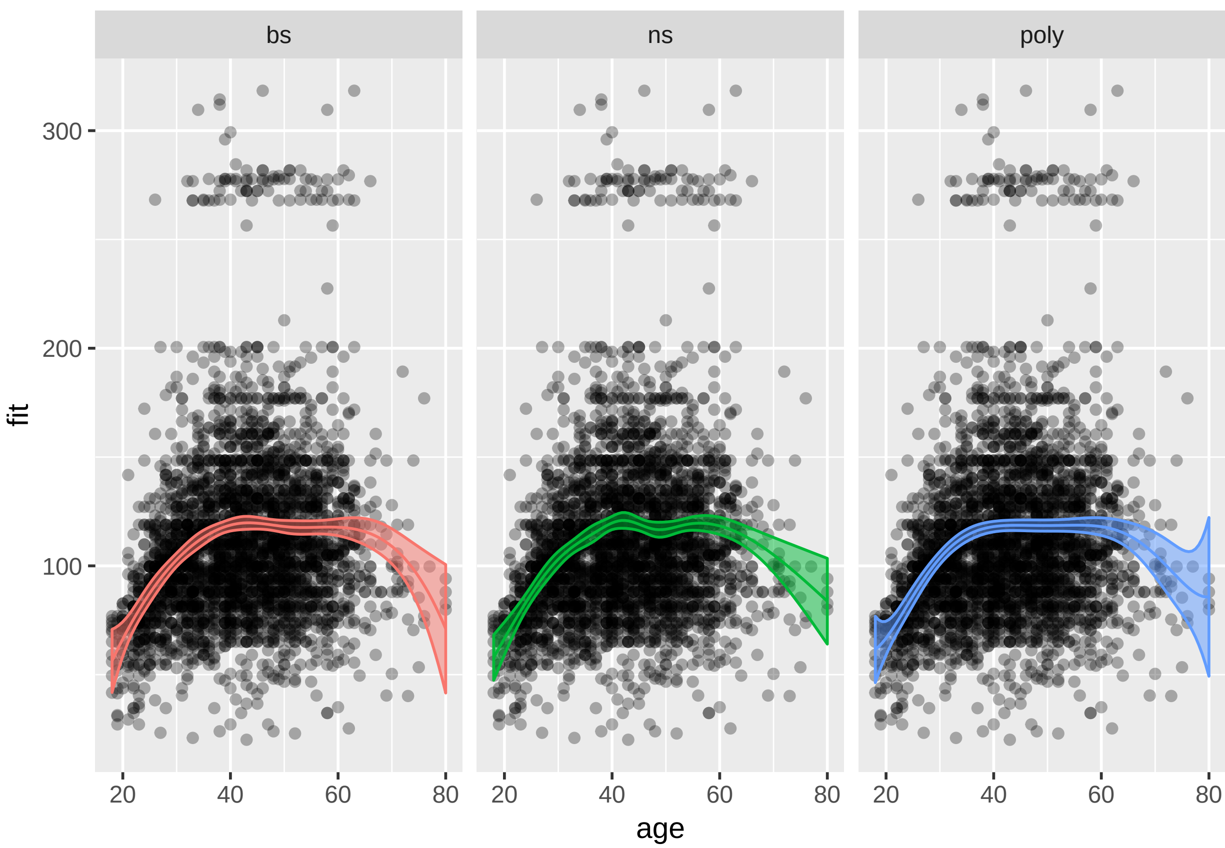
```
## [1] 7
```

```r
length(coef(poly_mod))
```

```
## [1] 7
```

```r
age_df  <- data.frame(age = seq(min(Wage$age), max(Wage$age), length.out =  100))

predict_se_fit <- function(model, newdata,...){

  predictions <- predict(model, newdata, se.fit = T)
  data.frame(
    fit   = predictions$fit,
    se.fit = predictions$se.fit
  )
}


all_models  <- bind_rows(
  age_df %>%
    nest() %>%
    mutate(mod = map(data, ~predict_se_fit(ns_mod, .)),
           model = "ns") %>%
```

```
    unnest(),
  age_df %>%
    nest() %>%
    mutate(mod = map(data, ~predict_se_fit(bs_mod, .)),
           model = "bs") %>%
    unnest(),
  age_df %>%
    nest() %>%
    mutate(mod = map(data, ~predict_se_fit(poly_mod, .)),
           model = "poly") %>%
    unnest()

) %>%
  mutate(fit_low = fit - 2*se.fit,
         fit_high = fit + 2*se.fit)

all_models %>%
  ggplot(aes(x = age, y = fit, fill = model, color = model)) +
  geom_point(aes(x = age, y = wage), data = Wage, inherit.aes = F, alpha = 0.3) +
  geom_ribbon(aes(ymax = fit_high, ymin = fit_low), alpha = 0.5) +
  geom_line() +
  facet_wrap(~model) +
  theme(legend.position = "none")
```



Smoothing Splines

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

Loss + penalty function

2nd dervative is a measure of roughness

The function g(x) that minimizes (7.11) can be shown to have some special properties: it is a piecewise cubic polynomial with knots at the unique values of x1,. . . ,xn, and continuous first and second derivatives at each knot. Furthermore, it is linear in the region outside of the extreme knots. In other words, the function g(x) that minimizes (7.11) is a natural cubic spline with knots at x1,. . . ,xn!

effective parameters

n to 2 as lamda goes to ininity

you can use loocv to calc

$$RSS_{cv} = \sum_{i=1}^{n}(\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{S_\lambda\}_{ii}})^2$$

$\hat{g}$ is the fitted values for the full spline

```
cv_smooth_spline <-with(Wage, smooth.spline(age,wage,cv=TRUE))
```

```
## Warning in smooth.spline(age, wage, cv = TRUE): cross-validation with non-
## unique 'x' values seems doubtful
```

```
gcv_smooth_spline <- with(Wage, smooth.spline(age,wage))
```

```
library(gam)
```

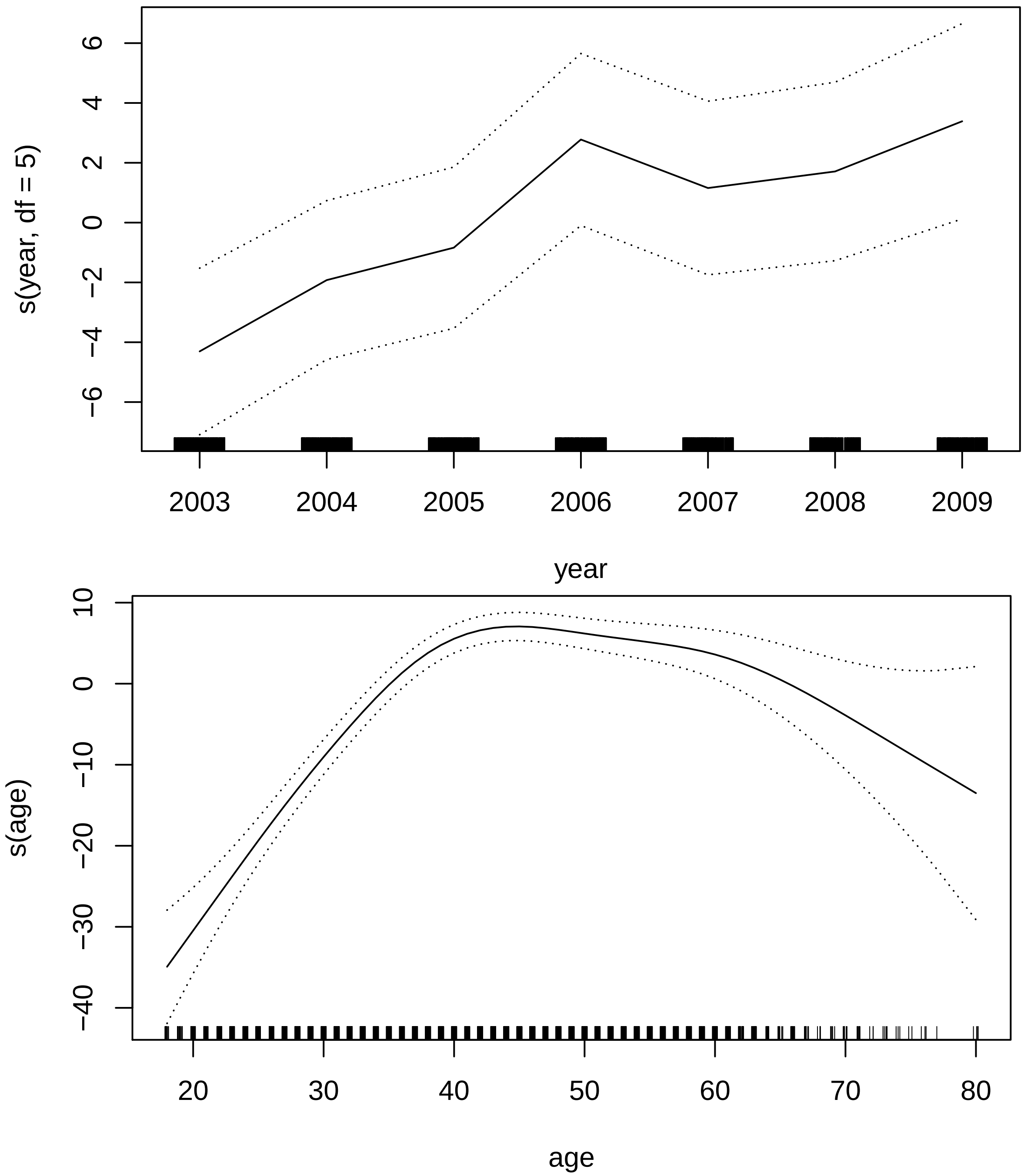
```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

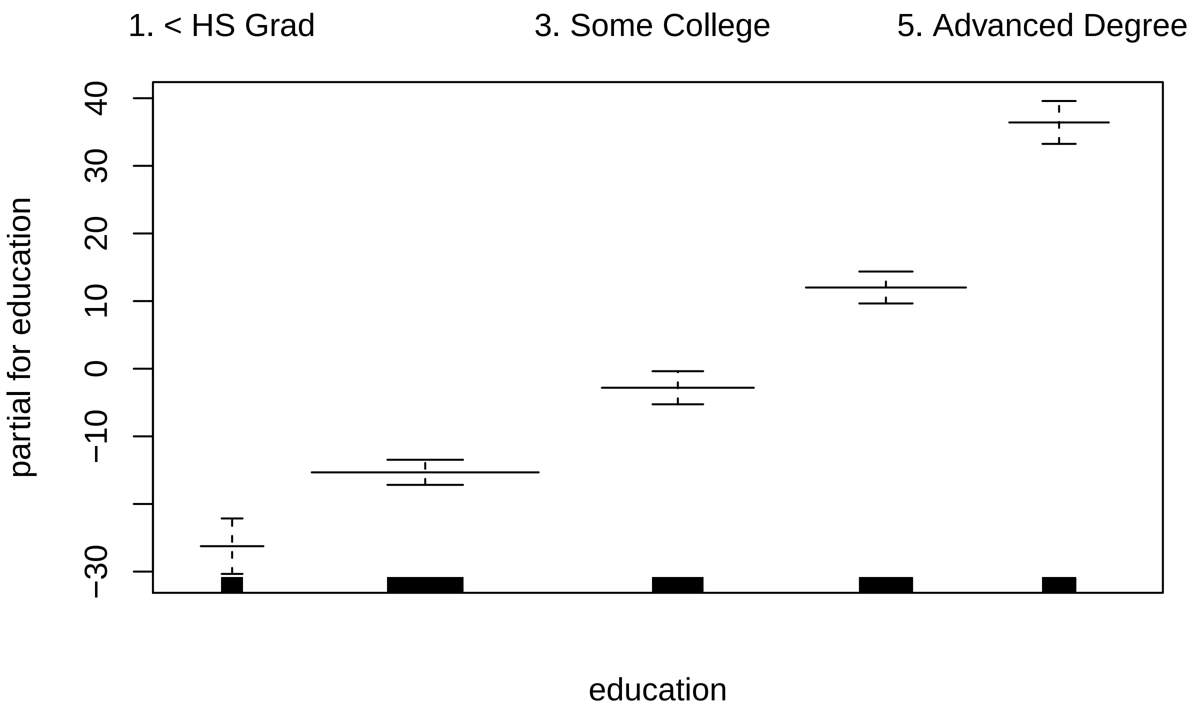
```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded gam 1.16
```

```
gam_mod <- gam(wage ~ s(year) + s(age) + education, data = Wage)
gam_mod <- gam(wage ~ s(year, df = 5) + s(age) + education, data = Wage)
plot(gam_mod, se = T)
```



s(year, df = 5)
6
4
2
0
−2
−4
−6
2003
2004
2005
2006
2007
2008
2009
year
s(age)
10
0
−10
−20
−30
−40
20
30
40
50
60
70
80
age

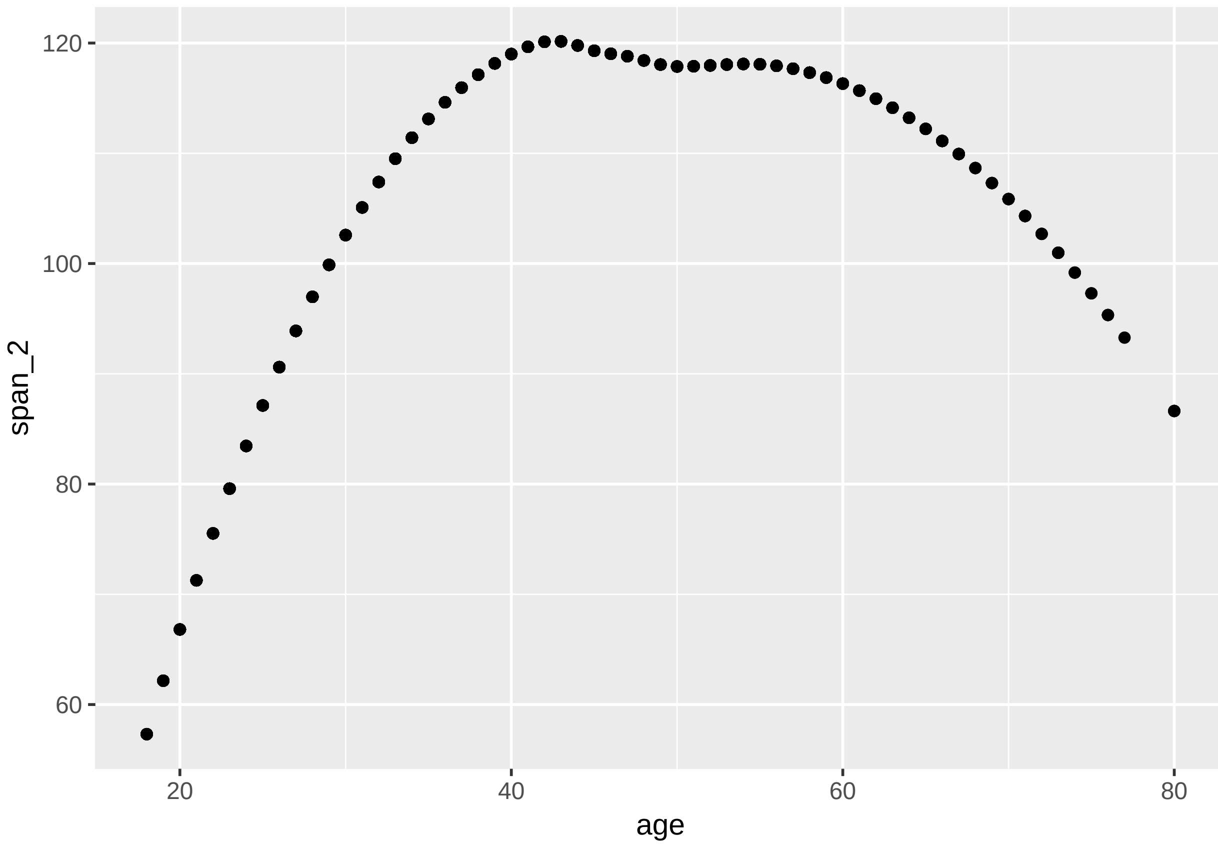


Local Regression

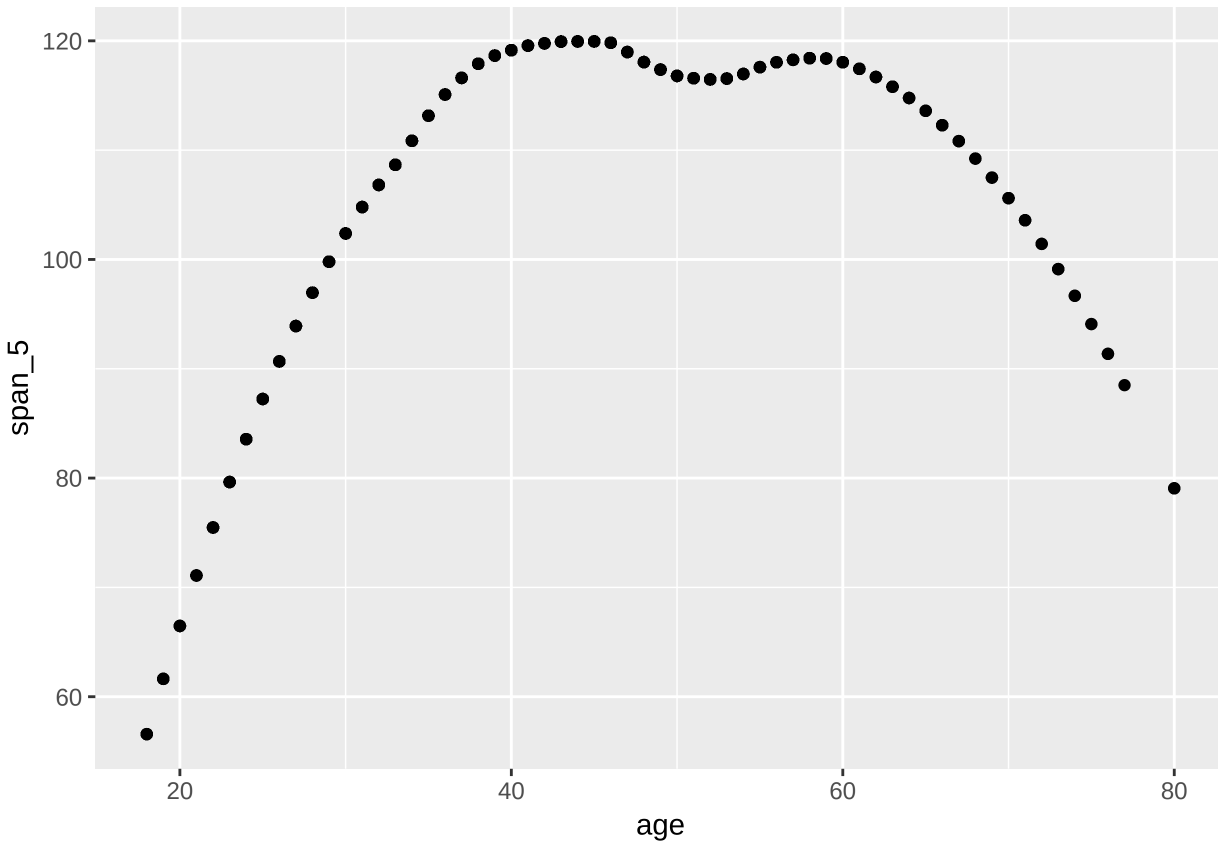
weighted regresion,

choose points within x_0. can be linear or w/e

```
data.frame(age = Wage$age,
span_2 = predict(loess(wage ~ age, spoan = .2, data = Wage))) %>%
  ggplot(aes(x = age, y = span_2)) +
  geom_point()
```




```
data.frame(age = Wage$age,
span_5 = predict(loess(wage ~ age, span = .5, data = Wage))) %>%
  ggplot(aes(x = age, y = span_5)) +
  geom_point()
```



Homework

Read Chapter 7 Question 9

Missing Data

Mean/Median Imputation

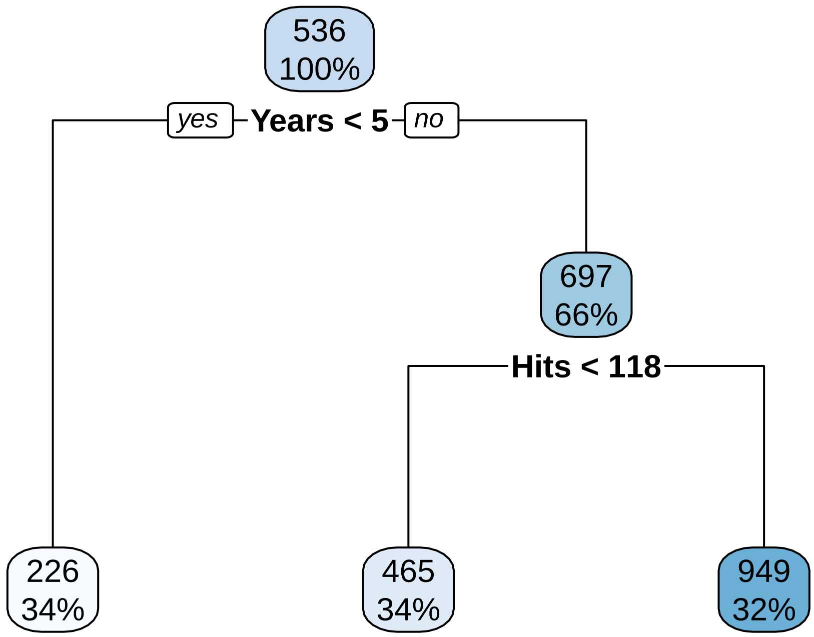
Modeling

https://topepo.github.io/caret/pre-processing.html#impute

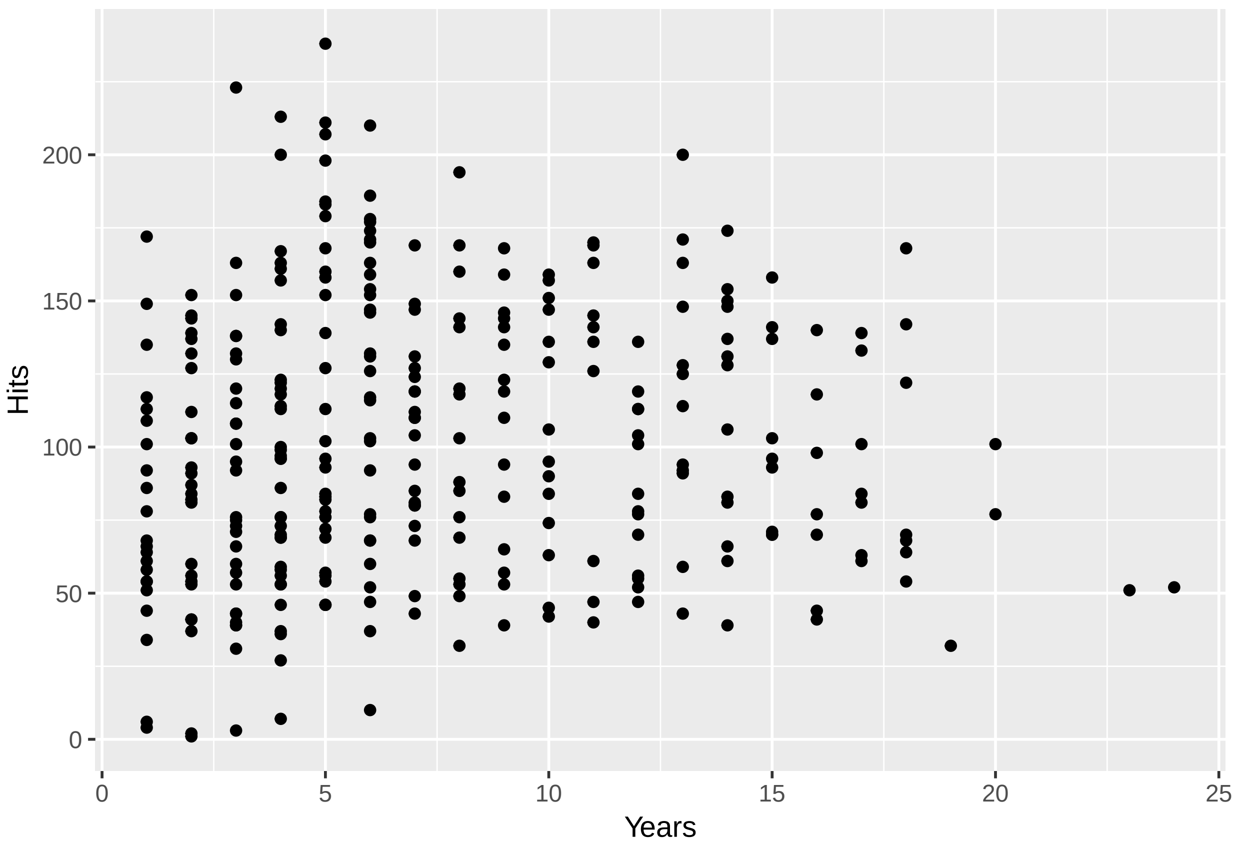
Tree Based Methods

This is a method of splitting up the predictor space into sections.

```
library(rpart)
library(rpart.plot)

rpart(Salary ~ Years + Hits, Hitters, control = rpart.control(maxdepth = 2)) %>%
  rpart.plot()
```




```r
Hitters %>%
  ggplot(aes(x = Years, y = Hits)) + geom_point()
```



Node - is a decision point

Terminal Node/Leaf - the final end of the tree or the predictor Internal Node

How does prediction work?

How do we make a tree?

1. Divide the predictor space into non overlapping regions. This means over *all* predictors.
2. Make a prediction for everyone that falls into a bucket

Ideally we could look at every single subset of features but that is not computationally feasable (akin to best subset) so we take a top down approach. Using recursive inarty splitting.

1. Pick a predictor and cut into 2 pieces by way of reducing RSS.

We do this by minimizing the joint RSS (it's the same as a step function). What does the RSS look like?

We then split one of those two regions to find the best next split.

This continues until a stoping criteria is reached (there are many different examples of stopping criteria that we will see in the lab)

Unfortunately this is a high variance method - we can't keep going or we will overfit the data.

Cost Complexity Pruning

What can we penalize?

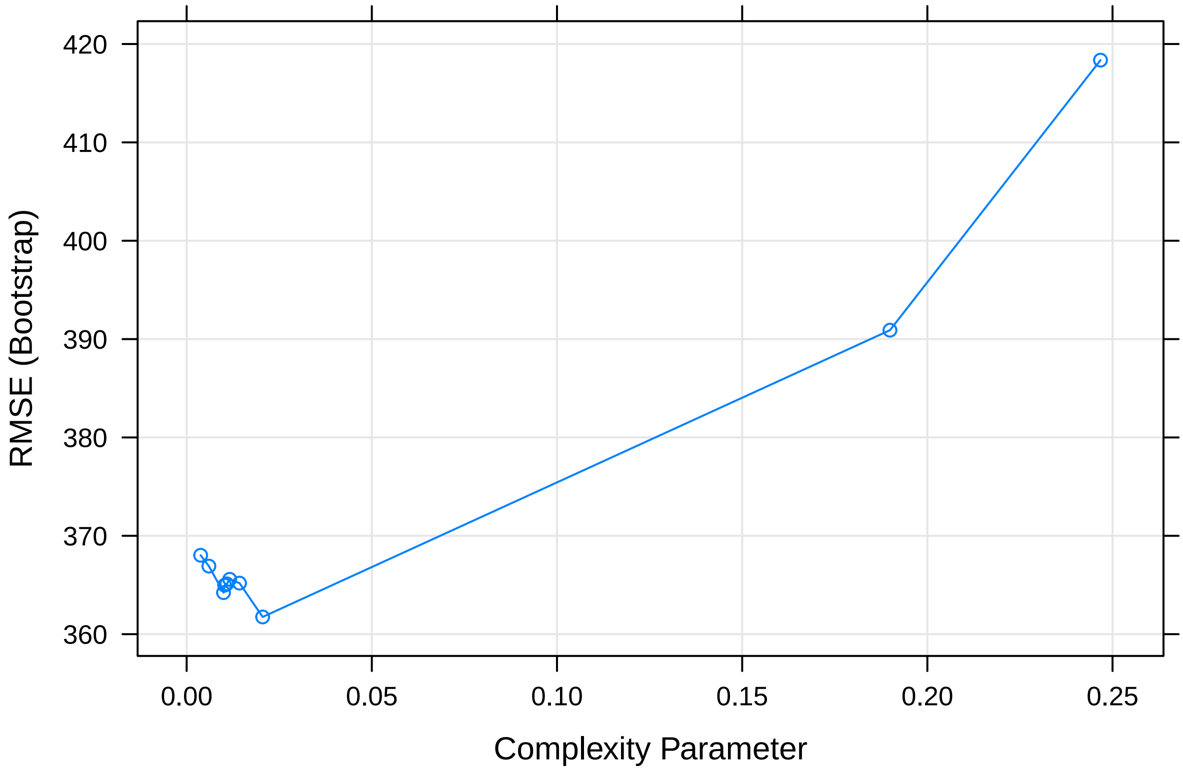
Prune back for a value that has a penalty of alpha * |T| (the number of terminal nodes).

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
d_hit <- Hitters %>%
  select(Salary, Years, Hits) %>%
  filter(complete.cases(.))
train(Salary ~ Years + Hits, d_hit, method = "rpart", tuneLength = 10) %>% plot
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```



Classification Trees

What error do we use?

Classification Error?

It is not sensative enough Gini Coefficient can be used:

$$G = \sum_{i}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Cross Entropy

$$\sum_{i}^{K} \hat{p}_{mk} log(\hat{p}_{mk})$$

You can use either Gini or Cross Entropy to build trees - but use classification error to prune them

Benefits

Very easy to explain Nice to display Handle both qual and quant variable easily.

Issues

Hi variability - How does pruning help with variability of the tree? Where is the higher variability?

Dealing with Issues of Trees

Bagging

Bootstrap Aggregation (or bagging) can help methods that have a high variance without loosing too much bias.

OOB Error Estimation - Natural cross validation.

Is the number of bootstraps a tuning parameter?

How do we interpert the models? You can look at the reduction of RSS when a variable is added and average it to see what's most important.

Random FOrests

Try to decorrelate the trees by using a random sample of m predictors

$m = \sqrt{p}$

If there is a strong predictor it will show up at the top of each bagged try so we try not to use it every time.

Homework

Read Chapter 8 Chapter 8 question 7