Flexible Regression Models

David M. Vock

PubH 7485/8485

Example IHDP Data

- The Infant Health and Development Program (IHDP) targeted low-birth-weight, premature infants.
- The study, conducted from 1985-1988, was a randomized trial. The treatment group received intensive high-quality child care and home visits.
- We are interested in studying only those from the treatment group who were sufficiently compliant with the intervention. This is, of course, a nonrandom subset of the treatment group in the population so there are important prognostic differences among the control group and this subset of treatment.
- Outcome is child's IQ at 36 months
- Data source: Hill JL. "Bayesian Nonparametric Modeling for Causal Inference" Journal of Computational and Graphical Statistics 20(1):217-240 DOI:10.1198/jcgs.2010.08162

Summarize Baseline Covariates

```
##                                          Stratified by treat
##                                           0                 1
##   n                                          561                67
##   IQ at 36 mo. (mean (SD))                 84.55 (19.94)    101.37 (15.94)
##   Treatment group (mean (SD))               0.00 (0.00)       1.00 (0.00)
##   Birth weight (mean (SD))               1789.26 (465.09) 1774.15 (449.19)
##   Mother's age (mean (SD))                 25.01 (6.13)      25.94 (5.80)
##   Neo-natal health index (mean (SD))       99.61 (15.68)    101.09 (15.96)
##   Birth order (mean (SD))                   1.96 (1.17)       1.82 (1.09)
##   Parity (mean (SD))                        1.91 (1.14)       1.79 (1.11)
##   Previous premature births (mean (SD))     0.25 (0.61)       0.18 (0.49)
##   Cigarettes/day (mean (SD))                3.89 (7.07)       4.85 (8.36)
##   Drinks/week (mean (SD))                   0.35 (1.43)       0.82 (3.60)
##   Mother's PPVT score (mean (SD))          81.00 (21.05)     82.93 (18.85)
##   Female (mean (SD))                        0.51 (0.50)       0.51 (0.50)
##   Twins (%)
##      0                                     497 (88.6)         54 (80.6)
##      1                                      35 ( 6.2)         10 (14.9)
##      2                                      29 ( 5.2)          3 ( 4.5)
##   Marital status (%)
##      1                                     279 (49.7)         35 (52.2)
##      2                                     237 (42.2)         26 (38.8)
##      3                                      43 ( 7.7)          6 ( 9.0)
##      4                                       2 ( 0.4)          0 ( 0.0)
##   Living status (%)
##      1                                     335 (59.7)         41 (61.2)
##      2                                     164 (29.2)         18 (26.9)
##      3                                       5 ( 0.9)          1 ( 1.5)
##      4                                      57 (10.2)          7 (10.4)
##   Primary language (%)
##      1                                     529 (94.3)         65 (97.0)
##      2                                      18 ( 3.2)          0 ( 0.0)
##      3                                       2 ( 0.4)          0 ( 0.0)
```

Unadjusted (Associational) Treatment Effect

```
## [1] "ATE (SE) = 16.8 (2.8)"
```

```
## [1] "95% CI: 11.4, 22.3"
```

Adjusted Treatment Effect

- Some imbalance among groups for key covariates
- Could estimate the ATE (causal treatment effect) by fitting a model which adjusts for the other potential confounders. We will only include main effects and linear terms. (NB: we will discuss the merits of this later)
- We will collapse values of some categorical covariates with small frequencies

Adjusted Treatment Effect

```
summary(m1)
```

```
##
## Call:
## lm(formula = iqsb.36 ~ . - mlt.birtF - b.marryF - languageF,
##     data = ihdp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.297  -8.931   0.322   8.991  52.645
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    56.505028   7.218858   7.827 2.26e-14 ***
## treat          13.658437   1.970802   6.930 1.08e-11 ***
## bw              0.002970   0.001314   2.260  0.02417 *
## momage          0.039073   0.137615   0.284  0.77656
## nnhealth        0.109941   0.039512   2.782  0.00556 **
## birth.o        -0.366990   1.186503  -0.309  0.75720
## parity         -1.977900   1.183381  -1.671  0.09516 .
## moreprem       -1.349465   1.120616  -1.204  0.22898
## cigs           -0.002597   0.091897  -0.028  0.97746
## alcohol        -0.079484   0.357161  -0.223  0.82397
## ppvt.imp        0.119932   0.044709   2.683  0.00751 **
## female          1.023101   1.225492   0.835  0.40413
## livwhoF2        0.750597   2.254144   0.333  0.73926
## livwhoF3        1.098846   6.557446   0.168  0.86698
## livwhoF4       -2.429416   2.700213  -0.900  0.36863
## whenprenF1      1.579942   3.309679   0.477  0.63327
## whenprenF2      0.422621   3.332991   0.127  0.89914
## whenprenF3      0.116226   4.223073   0.028  0.97805
## momed4F2       -0.089615   1.650703  -0.054  0.95672
## momed4F3        5.389108   2.001691   2.692  0.00729 **
## momed4F4       11.421857   2.210704   5.166 3.78e-07 ***
```

Standard Error of the ATE

- IF we believe that the homoskedastic assumption (and independence), then the standard error that is part of the software output is valid
- BUT let's estiamte a robust standard error using the bootstrap
- Demonstrate how to use boot package

First create function which returns ATE

```
ate.stat <- function(data, indices){
  data.boot <- data[indices,]

  m1.boot <- lm(iqsb.36 ~ .
    - mlt.birtF - b.marryF - languageF,
    data = data.boot)
  return(coef(m1.boot)[2])
}
```

Call boot package

```
set.seed(1101985) # bootstrapping is random - don't forget to set a seed!
boot.results <- boot(data=ihdp, statistic=ate.stat, R=1000, parallel="multicore", ncpus=8) # on Mac OS
# boot(data=ihdp, statistic=ate.stat, R=1000, parallel="snow", ncpus=2) # on Windows OS
boot.results ; #head(boot.results$t)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = ihdp, statistic = ate.stat, R = 1000, parallel = "multicore",
##     ncpus = 8)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 13.65844 -0.01656407    1.748839
```

```
# store results in matrix
ATE <- c(ATE, m1$coefficients[2]); SE <- c(SE, sd(boot.results$t))
LB <- c(LB, sort(boot.results$t)[25]); UB <- c(UB, sort(boot.results$t)[975])
estimator_name <- c(estimator_name, "Main Effects, Linear Only")
print(paste0("ATE (SE) = ", round(ATE[2], 1), " (", round(SE[2], 1), ")"))
```

```
## [1] "ATE (SE) = 13.7 (1.7)"
```

```
print(paste0("95% CI:", round(LB[2], 1), ", ", round(UB[2], 1)))
```

```
## [1] "95% CI:10.1, 17.2"
```

Model mispsecification

- Proposed framework is dependent on getting the "right" regression model for the outcome
- The initial model proposed here is somewhat limiting (only linear main effcts)
- May want to examine more flexible approaches
- We will summarize a handful of different ideas and their practical implementation in R; this is not meant to be comprehensive

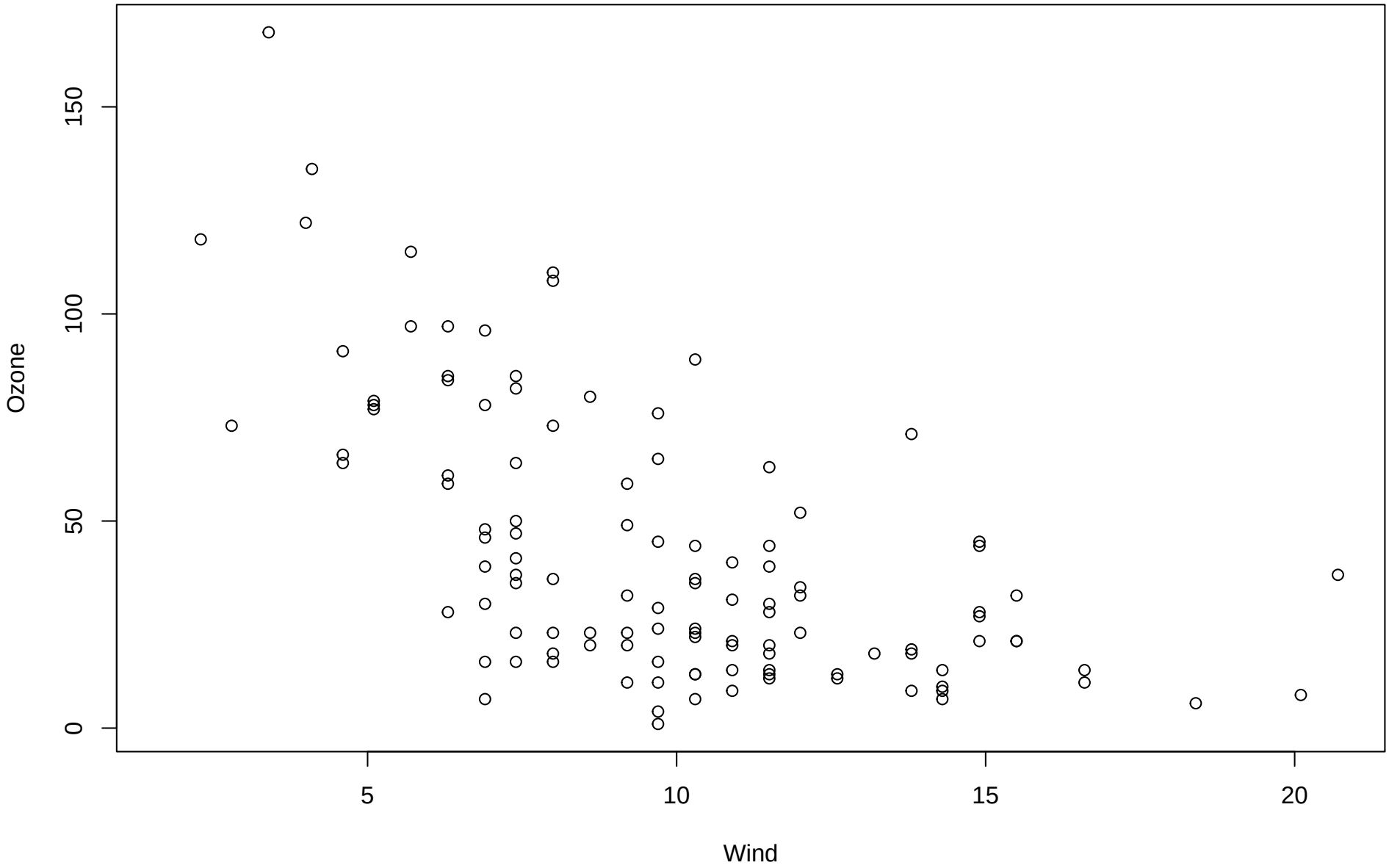
Diagnosisng Nonlinear Relationships

- In (multiple) linear regression, we typically assume that the response is linearly related to the covariates
- Diagnose departures from normality using scatter plot of covariate and response or residual plot
- Plots may be uninformative as we shall see shortly

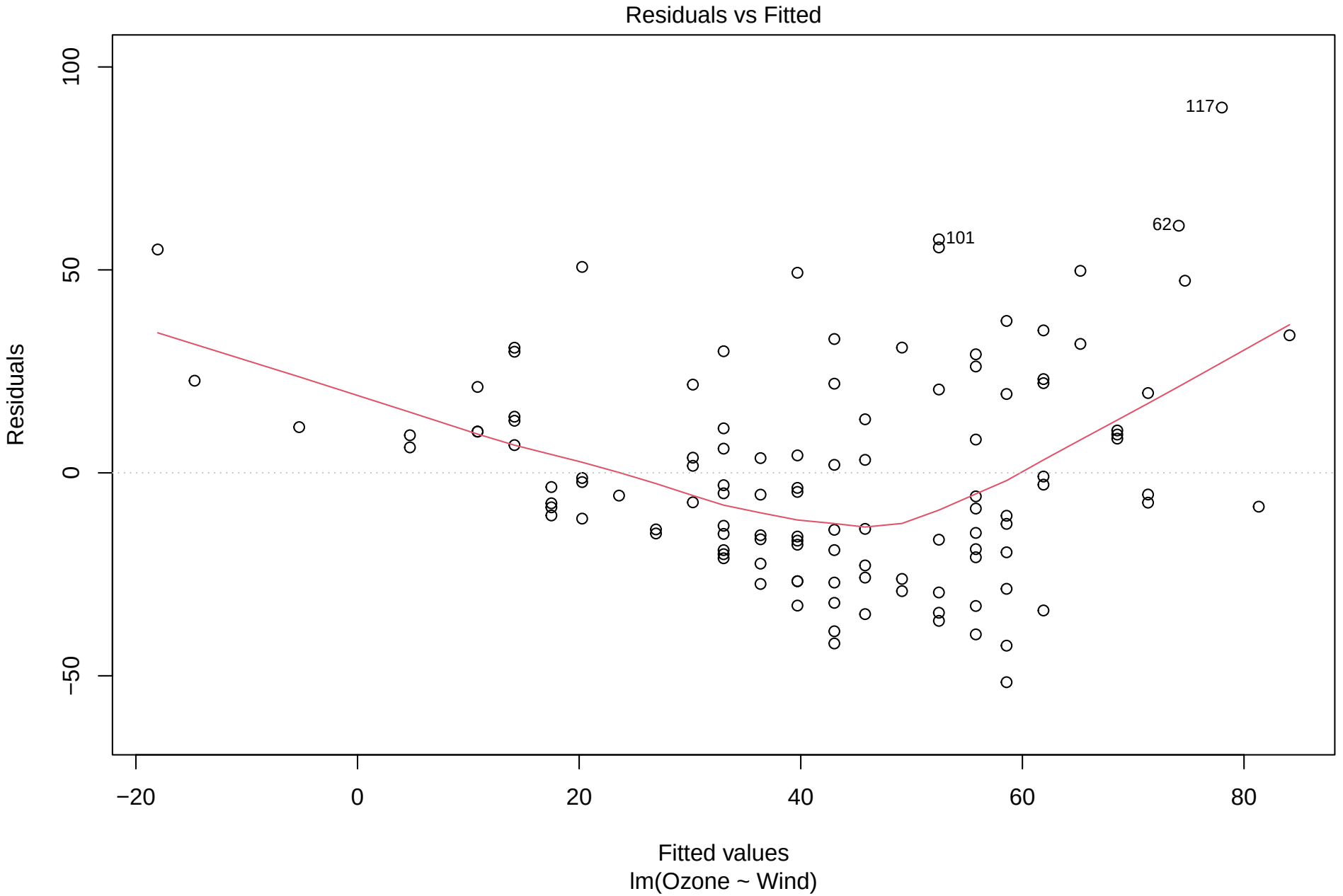
Example Dataset

- Ozone level (ppb) in New York City measured daily from May to September in 1973
- Covariates include Wind (mph), Temperature (F), Solar radiation (Langley), and calendar day
- 153 calendar days included
- Focus on relationship between Wind and Ozone

Scatter Plot of Data



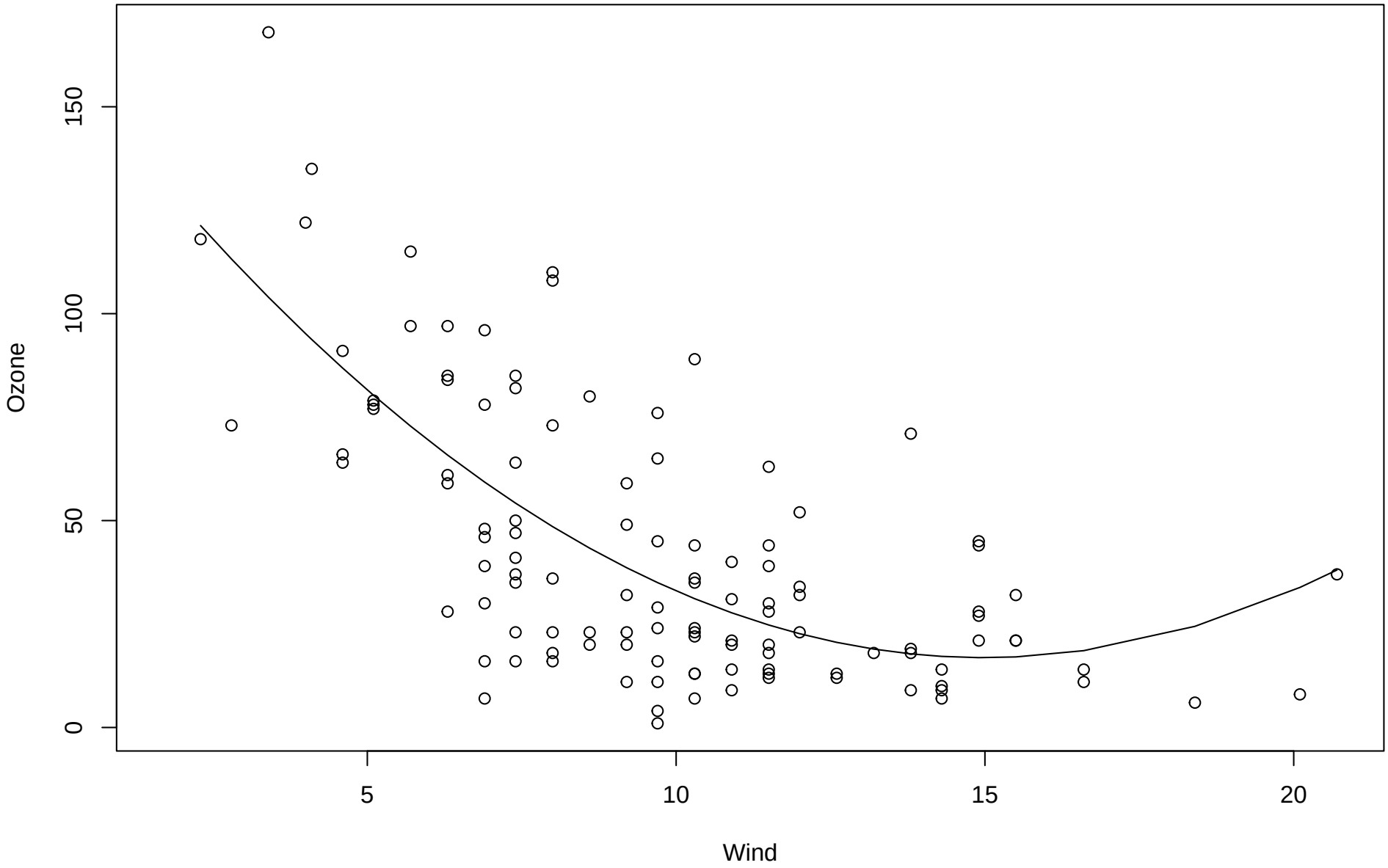
Residual Plot from Fitting a Linear Model



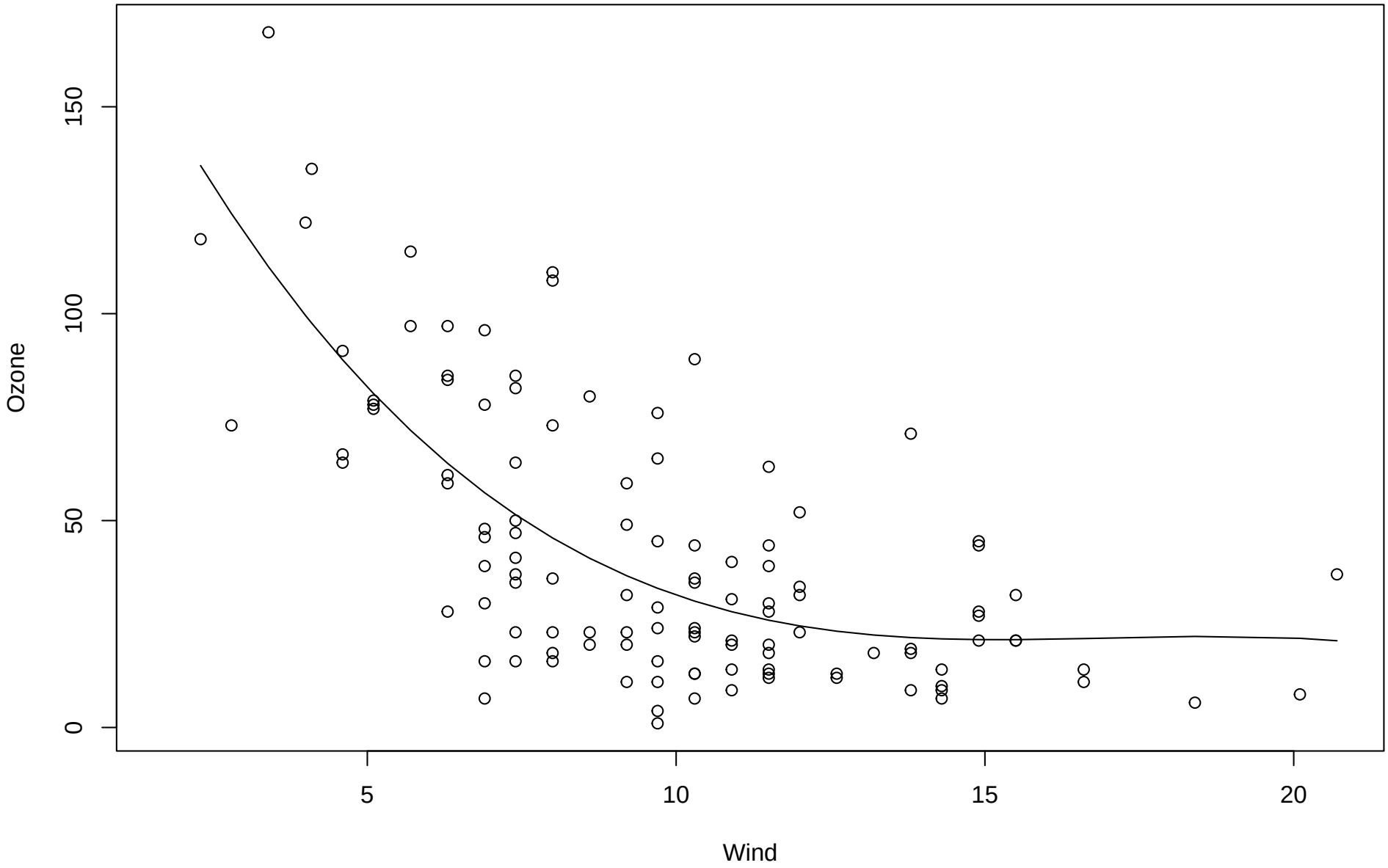
What to Do?

- Clearly a linear relationship is not sufficient to capture the relationship between wind and ozone
- Standard approaches to handle nonlinear relationships would include incorporating a quadratic terms or transforming the response/predictor
- Quadratic fit may be inadequate: How many polynomial terms should we include (e.g., cubic, quartic, etc.)?
- The challenge with very high order polynomials is that they tend to be rather "wiggly'' and have unusual tail behavior.

Quadratic Fit



Cubic Fit



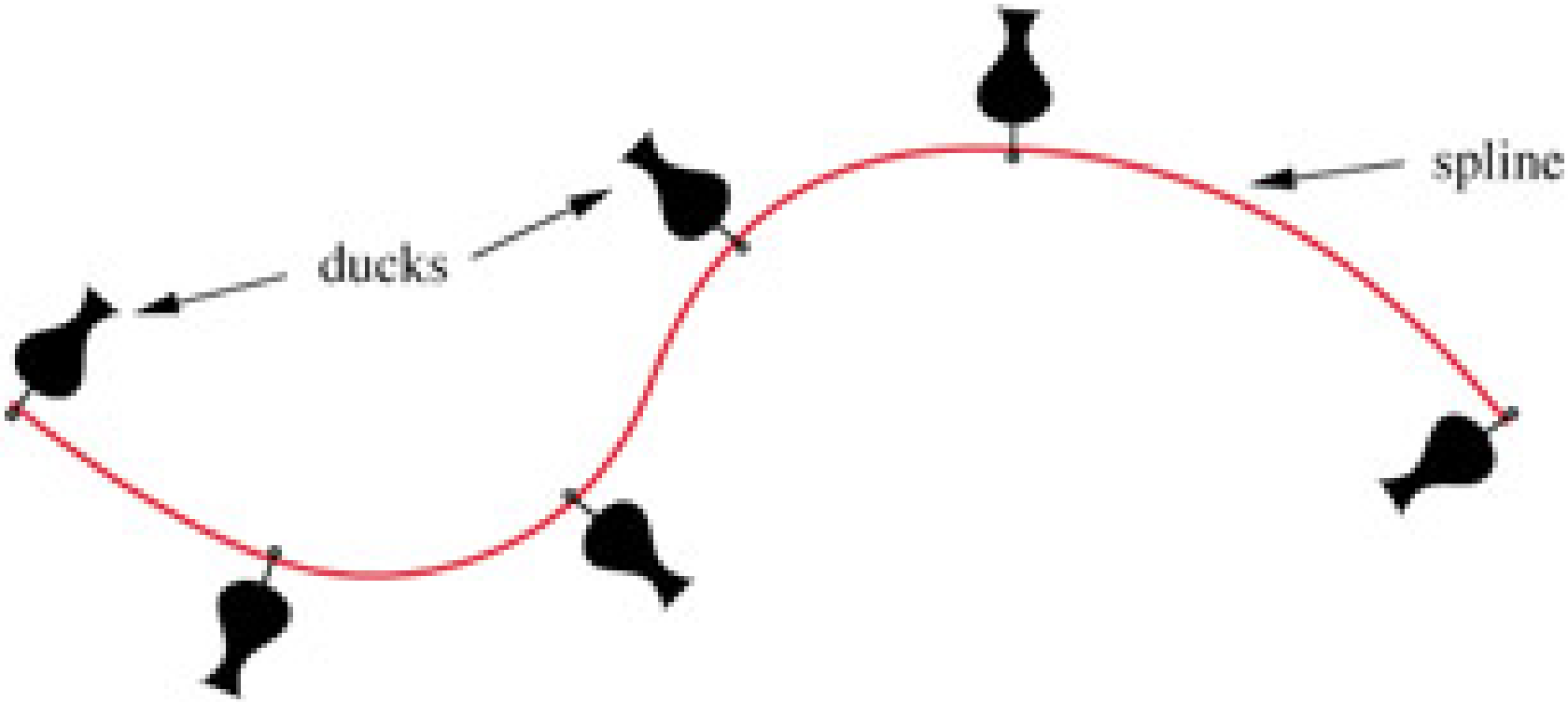
Key Citation and Warning

- Harrell, F.E. (2015) "Regression Modeling Strategies: With Applications to Linear Models. Logistic Regression, and Survival Models." 2nd Ed. Springer
- The discussion today will be highly applied based on my experience in collaborative settings
- Prioritize practicability (i.e., getting a good answer now over a perfect one in a month) over mathematical elegance

(Draftman) Splines

- From Wikipedia
- Consists of a long strip fixed in position at a number of points that relaxes to form and hold a smooth curve passing through those points for the purpose of transferring that curve to another material
- Used for creating engineering designs
- The splines were held in place with lead weights. The elasticity of the spline material combined with the constraint of the control points, or knots, would cause the strip to take the shape that minimized the energy required for bending it between the fixed points, this being the smoothest possible shape

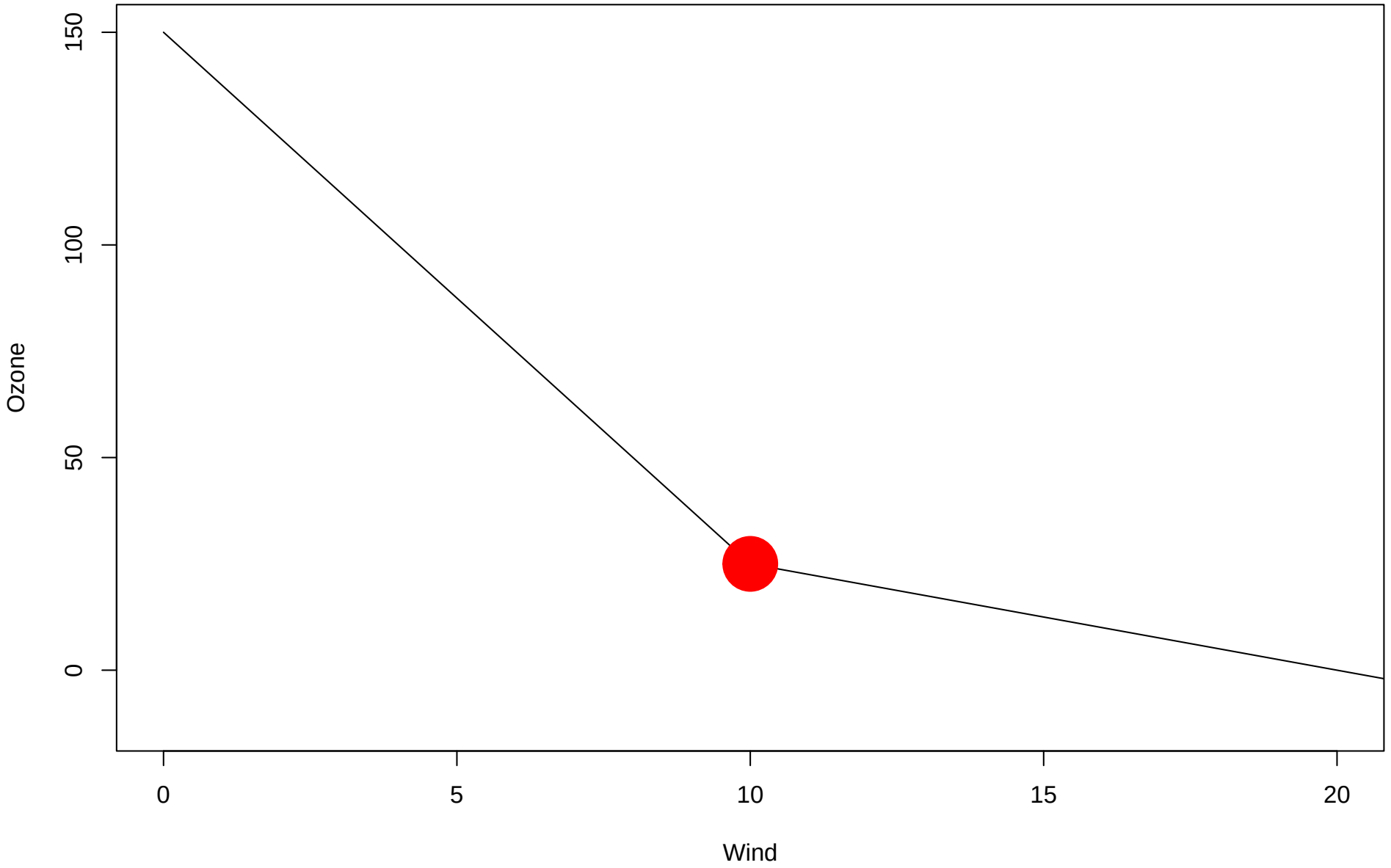
(Draftman) Splines



Linear Splines

- We could allow the linear function to change at a certain point.
- Of course this model is unreasonable (not continuous)
- A slightly more reasonable model would allow the slope to change at a specific point but ensure continuity. How many parameters are in this model?
- The change point is known as a "knot"
- Although we want to allow the slope to change we still want the function to be continuous
- We will eventually consider more sophisticated functions besides piecewise linear
- Assume that the location of the knot point is fixed (i.e., known, not estimated) by the analyst

Graph of Linear Spline with Single Knot



Single Knot - Parameterization

- Of course we want to find the linear spline function which bests fit the data
- What are the three parameters that we must estimate in this model?

Single Knot - Parameterization

- Let X be the covariate of interest (Wind speed here) and Y the response of interest (Ozone) and s the knot point
- We can estimate the best linear spline model by fitting a usual multiple regression model with covariates X and $\max(0, X - s)$
- Note that $\max(0, Z)$ is often denoted by Z_+
- The model we are fitting is $Y_i = \beta_0 + \beta_1 X_i + \beta_2 (X - k_1)_+ + \epsilon_i$ where $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$
- Note that β_2 is the difference in slopes and β_1 is the slope for values of X less than the knot

Single Knot - Parameterization R Code

```
knot.pt <- 10
Wind.s1 <- pmax(Wind - knot.pt, 0)
```

Single Knot - Model Output

```
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)    141.99      10.555  13.452 3.301e-25
## Wind           -11.63       1.282  -9.073 4.255e-15
## Wind.s1         11.09       2.048   5.414 3.506e-07
```

Note that the slope after 10 mph is -11.63 + 11.09 = -0.54

Single Knot - Model Interpretation

The average ozone level in NYC declines 11.63 ppb for every one mile per hour increase in the wind speed up to 10 mph. After 10 mph, the relationship between wind speed and ozone is dramatically attenuated; the average ozone level only declines 0.54 ppb for each one mph increase in wind speed after 10 mph.

Standard Errors/Confidence Intervals

- Once we know the standard error, how can we calculate a 95% CI for the parameter?
- The standard error for $\hat{\beta}_1$ can be read off the table. How could we get the standard error for $\hat{\beta}_1 + \hat{\beta}_2$, our estimate of the slope after 10 mph?

Standard Errors/Confidence Intervals

- Standard error for $\hat{\beta}_1 + \hat{\beta}_2$

```
se.part2 <- sqrt(vcov(model_ls)[2, 2] + vcov(model_ls)[3, 3] +
print(se.part2, digits = 3)
```

```
## [1] 1.11
```

- Confidence interval β_1

```
##     Wind     Wind
## -14.147   -9.121
```

- Confidence interval $\beta_1 + \beta_2$

```
##    Wind    Wind
## -2.725   1.635
```

Single Knot - Testing if the Spline is Necessary

- A test of whether or not β_2 is significantly different from zero is a test for whether or not the relationship between wind and ozone is significantly nonlinear
- Here we have substantial evidence that the relationship between ozone and wind is nonlinear and changes at 10 mph ($p < 0.001$)

Multiple Knot Points - Paremetrization

- No reason why we couldn't allow the slope to change at multiple points
- For an arbitrary number of knot points $(s_1, \ldots, s_K)$ we would fit a multiple linear regression model - what should the covariates be in this case?
- Note the number of additional covariates (beyond those needed for a linear model) is equal to K

Multiple Knot Points - Output

- Fit a model with knot points at 7.5 mph and 15 mph. Interpret this model

```
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)      171.19     15.916  10.756 5.814e-19
## Wind             -17.05      2.384  -7.152 9.362e-11
## Wind.s1           13.65      2.993   4.559 1.318e-05
## Wind.s2            3.58      3.548   1.009 3.152e-01
```

Multiple Knot Points - Selection

- Until know I've been pretty careless about how to pick knot points
- How might you compare the model with one knot point as compared to two know points?
- How would you compare two models with the same knot points but with different locations?

Multiple Knot Points - Testing if the Spline is Necessary

- Test of whether or not the relationship between ozone and wind speed is linear is equivalent to testing whether or not $\beta_2 = 0$ and $\beta_3 = 0$ simultaneously
- This is a composite hypothesis test not two simultaneous tests

```
model_lin <- lm(Ozone ~ Wind)
anova(model_lin, model_ls2)
```

```
## Analysis of Variance Table
##
## Model 1: Ozone ~ Wind
## Model 2: Ozone ~ Wind + Wind.s1 + Wind.s2
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    114 79859
## 2    112 63121  2     16738 14.849 1.904e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 ' ' 0.1 '
```

Robert Frost: Mending Wall

- Something there is that doesn't love a wall
- Good fences make good neighbors

Robert Frost: Mending Wall Applied to Splines

- Something there is that doesn't love a smooth function
- (Piecewise) Linear functions make good and interpretable models

Cubic Splines: Intro

- Idea is that a cubic function is fairly flexible but we want the cubic function to be allowed to change at different knot points
- How many parameters must we fit to determine a cubic function? If we have 4 different knot points how many parameters is that? (HINT: way too many)
- To reduce the number of parameters we need to estimate we impose some restrictions just as we did with the linear spline model. Specifically we assume that

1) The function is continuous
2) The first derivative is continuous
3) The second derivative is continuous
4) The function is linear outside the end knot points

- Known as a restricted cubic spline model

Cubic Splines Parameterization

- Don't all those restrictions make this impossible to work with? Surprisingly, no!
- Only require $K-2$ additional covariates (beyond a linear model) to fit this restricted cubic spline model
- That is, we assume $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_{K-1} X_{K-1} + \epsilon$ where $X_1 = X$ and $X_{j+1} = (X - s_j)_+^3 - (X - s_{K-1})_+^3 (s_K - s_j)/(s_K - s_{K-1}) + (X - s_K)_+^3 (s_{K-1} - s_j)/(s_K - s_{K-1})$

Cubic Splines in the rms package

- Isn't this a lot of coding? Yes, but luckily the rms package takes care of this for us.
- Here the knot locations are specified using the quantiles of the distribution

```
library(rms)
model_rms <- ols(Ozone ~ rcs(Wind, 5), data = airquality)
```

Cubic Splines - Choosing the Number of Knot Points

- Pertinent question throughout is how to choose the number and location of the knot points
- Larger number of knot points: greater flexibility but we risk overfitting the data
- If we let the number of knot points be large then we really need to control for overfitting using some form of variable selection or coefficient shrinkage
- Need to do that smartly - actually want to control the "wiggliness" of the nonlinear function which may not be controlled by setting some coefficients equal to zero
- Most simulation studies have shown that 4 or 5 knot points is sufficient to model most nonlinear relationships

Cubic Splines - Choosing the Location of Knot Points

- In the absence of strong subject-area prior knowledge on the functional form of the covariate typically choose knot points based on the percentiles of the covariate distribution
- Typically the smallest and largest knot points are the 5^{th} and 95^{th} percentile of the distribution
- "Interior"" knots are equally spaced percentiles
- For 5 knot points the knots would be chosen using the 5^{th}, 27.5^{th}, 50^{th}, 72.5^{th}, and 95^{th} percentiles or the covariates
- For 4 knot points the knots would be chosen using the 5^{th}, 35^{th}, 65^{th}, and 95^{th} percentiles or the covariates
- This is the default in the rms package. Second argument gives number of knot points

Cubic Splines Output

```
model_rms$coef
```

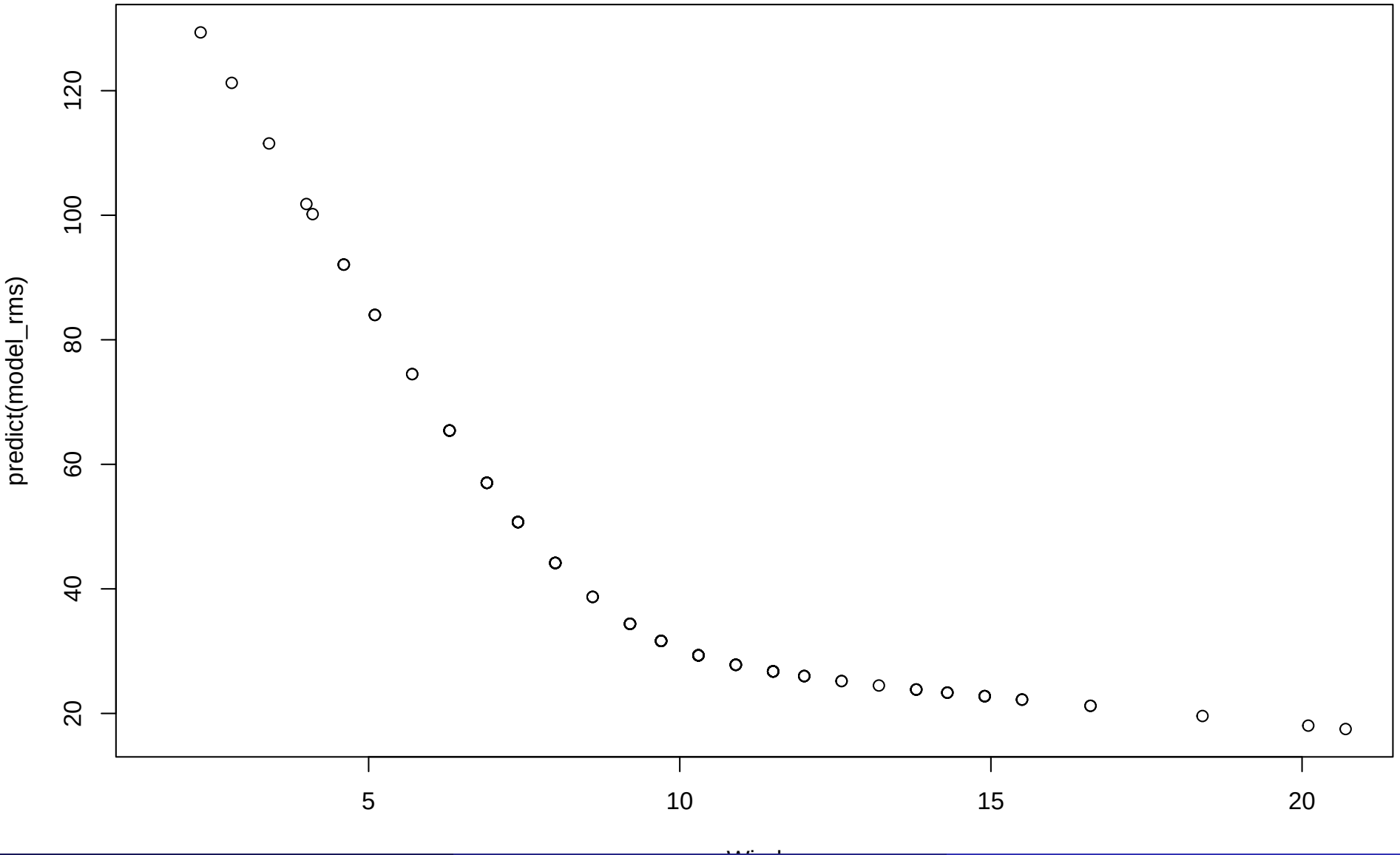
```
##  Intercept       Wind      Wind'     Wind''    Wind'''
## 166.64249 -16.21039  21.88926 -21.48347 -31.30685
```

Cubic Splines Interpretation

- Impossible to interpret the actual coefficients of the model
- In some cases if we are just adjusting for this covariate in a multivariable model, the interpretability doesn't matter. THIS IS THE CASE FOR OUTCOME REGRESSION MODELS IN CAUSAL INFERENCE
- One possibility is to present a graphical representation of the relationship between the covariate and the outcome (e.g., plot predicted value versus the covariate)
- But graphical representations are difficult to summarize in say an abstract

Cubic Splines Output - Graph

```
plot(Wind, predict(model_rms))
```



Cubic Splines - Tesing if the Relationship is Nonlinear

- A test of whether or not the relationship between wind speed and ozone is nonlinear is fairly interpretable but doesn't tell us much more
- A limitation here is that we have not accounted for the fact that the knot points were not chosen a priori.

```
anova(model_rms)
```

```
##                 Analysis of Variance          Response: Ozo
##
##  Factor     d.f. Partial SS MS           F     P
##  Wind          4  63181.49   15795.3718 28.30 <.0001
##   Nonlinear    3  17897.44    5965.8138 10.69 <.0001
##  REGRESSION    4  63181.49   15795.3718 28.30 <.0001
##  ERROR       111  61961.57     558.2124
```

IHDP Data Set

- Fit a main effects model as before but use restricted cubic splines with 4 knot points for the continuous factors
- We are going to punt on the question of variable selection for now (both for the nonlinear terms and main effects)
- Consider including interaction terms once we have discussed variable selection

Adjusted Treatment Effect with RCS

```
m4 <- lm(iqsb.36 ~ treat +
    rcs(bw, 4) + rcs(momage, 4) + rcs(nnhealth, 4) + rcs(ppvt.
    birth.o + parity + moreprem + cigs + alcohol +
    female + mlt.birtaltF + b.marryaltF + livwhoF + languageal
    momed4F + momraceF + workdur.imp, data=ihdp)
```

Adjusted Treatment Effect with RCS

```
summary(m4)
```

```
##
## Call:
## lm(formula = iqsb.36 ~ treat + rcs(bw, 4) + rcs(momage, 4) +
##     rcs(nnhealth, 4) + rcs(ppvt.imp, 4) + birth.o + parity +
##     moreprem + cigs + alcohol + female + mlt.birtaltF + b.marryaltF +
##     livwhoF + languagealtF + whenprenF + momed4F + momraceF +
##     workdur.imp, data = ihdp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.423  -8.925   0.431   8.785  54.437
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 54.901807  17.331031   3.168 0.001615 **
## treat                       13.621841   1.985936   6.859 1.75e-11 ***
## rcs(bw, 4)bw                 0.011026   0.004939   2.233 0.025943 *
## rcs(bw, 4)bw'               -0.011987   0.010287  -1.165 0.244374
## rcs(bw, 4)bw''               0.033904   0.063842   0.531 0.595579
## rcs(momage, 4)momage         0.081460   0.617454   0.132 0.895086
## rcs(momage, 4)momage'        0.057670   2.103695   0.027 0.978139
## rcs(momage, 4)momage''      -0.423691   5.678726  -0.075 0.940550
## rcs(nnhealth, 4)nnhealth     0.124728   0.094938   1.314 0.189426
## rcs(nnhealth, 4)nnhealth'    0.071922   0.236999   0.303 0.761637
## rcs(nnhealth, 4)nnhealth''  -1.108561   1.527867  -0.726 0.468394
## rcs(ppvt.imp, 4)ppvt.imp    -0.049852   0.141620  -0.352 0.724956
## rcs(ppvt.imp, 4)ppvt.imp'    0.812447   0.633596   1.282 0.200246
## rcs(ppvt.imp, 4)ppvt.imp''  -2.075853   1.678242  -1.237 0.216606
## birth.o                     -0.321957   1.194870  -0.269 0.787678
## parity                      -1.984076   1.188254  -1.670 0.095498 .
## moreprem                    -1.458756   1.129501  -1.292 0.197032
## cigs                        -0.002011   0.003730  -0.539 0.589891
```

Adjusted Treatment Effect with RCS

```
m4alt <- ols(iqsb.36 ~ treat +
    rcs(bw, 4) + rcs(momage, 4) + rcs(nnhealth, 4) + rcs(ppvt.imp, 4) +
    birth.o + parity + moreprem + cigs + alcohol +
    female + mlt.birtaltF + b.marryaltF + livwhoF + languagealtF + whenprenF +
    momed4F + momraceF + workdur.imp, data=ihdp)
anova(m4alt)
```

```
##                 Analysis of Variance          Response: iqsb.36
##
##  Factor          d.f. Partial SS   MS           F     P
##  treat             1  1.046989e+04 10469.886926 47.05 <.0001
##  bw                3  2.064346e+03   688.115484  3.09 0.0266
##   Nonlinear        2  1.091881e+03   545.940533  2.45 0.0869
##  momage            3  3.399153e+01    11.330509  0.05 0.9848
##   Nonlinear        2  2.267301e+01    11.336503  0.05 0.9503
##  nnhealth          3  1.776792e+03   592.264076  2.66 0.0473
##   Nonlinear        2  4.555588e+02   227.779384  1.02 0.3599
##  ppvt.imp          3  1.994583e+03   664.860902  2.99 0.0306
##   Nonlinear        2  3.749808e+02   187.490408  0.84 0.4311
##  birth.o           1  1.615676e+01    16.156761  0.07 0.7877
##  parity            1  6.204389e+02   620.438940  2.79 0.0955
##  moreprem          1  3.711875e+02   371.187545  1.67 0.1970
##  cigs              1  1.024280e-01     0.102428  0.00 0.9829
##  alcohol           1  7.833212e+00     7.833212  0.04 0.8512
##  female            1  1.328952e+02   132.895159  0.60 0.4400
##  mlt.birtaltF      1  5.157329e+01    51.573289  0.23 0.6304
##  b.marryaltF       1  6.219191e+02   621.919066  2.79 0.0951
##  livwhoF           3  5.084328e+02   169.477605  0.76 0.5159
##  languagealtF      1  4.003783e+03  4003.782524 17.99 <.0001
##  whenprenF         3  2.038306e+02    67.943545  0.31 0.8216
##  momed4F           3  4.285545e+03  1428.515149  6.42 0.0003
##  momraceF          2  8.043474e+03  4021.736948 18.07 <.0001
##  workdur.imp       1  3.804594e+02   380.459399  1.71 0.1915
##  TOTAL NONLINEAR   8  1.987712e+03   248.464051  1.12 0.3499
```

Adjusted Treatment Effect with RCS Summary

- Use bootstrap SE as with linear main effects model

Call boot package

```
set.seed(1101985) # bootstrapping is random - don't forget to set a seed!
boot.results <- boot(data=ihdp, statistic=ate.stat.rcs, R=1000, parallel="multicore", ncpus=8) # on Mac OS
# boot(data=ihdp, statistic=ate.stat, R=1000, parallel="snow", ncpus=2) # on Windows OS
boot.results ; #head(boot.results$t)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = ihdp, statistic = ate.stat.rcs, R = 1000, parallel = "multicore",
##     ncpus = 8)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 13.62184 -0.02411376    1.777412
```

```
# store results in matrix
ATE <- c(ATE, m4$coefficients[2]); SE <- c(SE, sd(boot.results$t))
LB <- c(LB, sort(boot.results$t)[25]); UB <- c(UB, sort(boot.results$t)[975])
estimator_name <- c(estimator_name, "Main Effects, RCS")
print(paste0("ATE (SE) = ", round(ATE[3], 1), " (", round(SE[3], 1), ")"))
```

```
## [1] "ATE (SE) = 13.6 (1.8)"
```

```
print(paste0("95% CI:", round(LB[3], 1), ", ", round(UB[3], 1)))
```

```
## [1] "95% CI:10, 17.2"
```

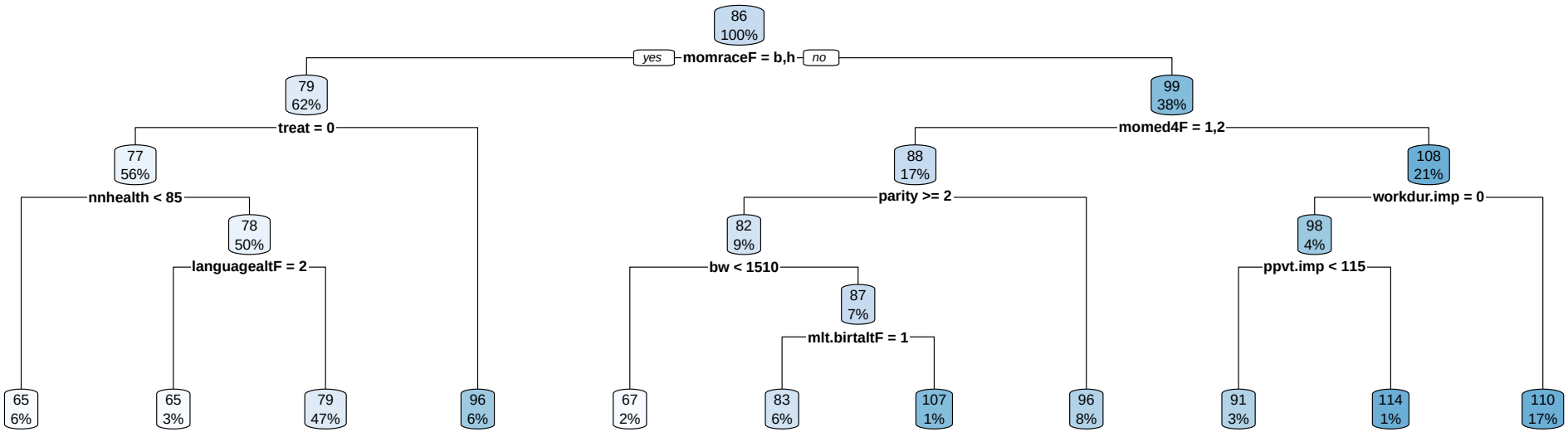
Classification and Regression Trees

- Classification and Regression Trees (CART) were originally introduced by Leo Breiman in 1984.
- It remains one of the most popular machine learning algorithms because of its simplicity and interpretability.
- The main idea of CART is that we are trying to approximate any function $f(x)$ by a piecewise constant $\hat{f}(x)$ using recursive partitioning.
- More simply, the goal is to create a model that predicts the value of a target variable based on several input variables.

Building a Classfication Tree

- Here, we are using the packages "rpart" and "rpart.plot" to produce a regression tree:

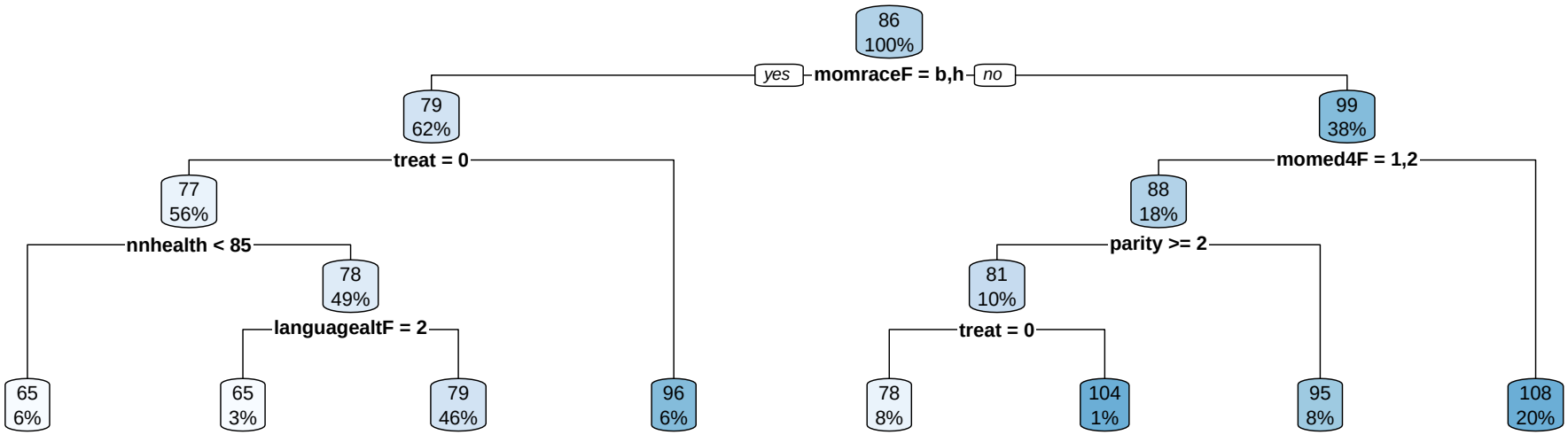
```
fit <- rpart(iqsb.36 ~ .,
  data = ihdp[, -which(colnames(ihdp) %in% c("mlt.birtF", "b.m
  method = "anova")
rpart.plot(fit)
```



- Note: method = "anova" is used here for a regression tree. The complexity parameter "cp" is used to control how far to grow the tree.

Using Single Trees

- If we use 600/628 observations, and build the regression tree the same way as previously:



- The variable "treat" became a split in the tree to the right, and we now have fewer terminal nodes

Random Forest

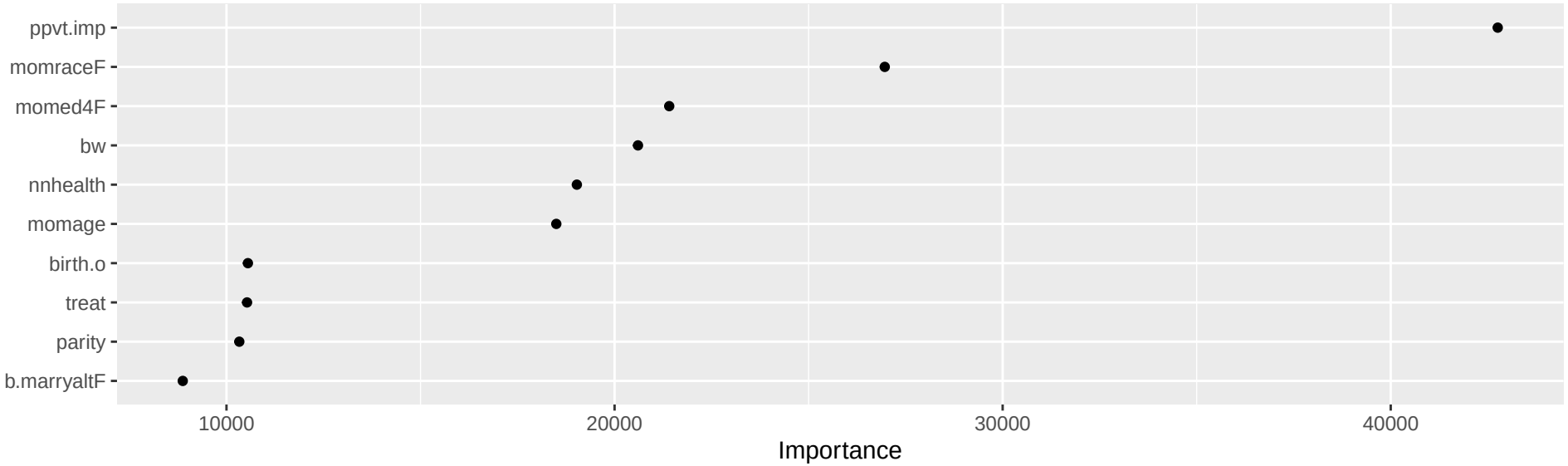
- Often times, using a single tree as a model can be unstable and give weak predictions. An easy way to improve on the prediction accuracy is to use multiple version of it to form a final classifier; this is the logic behind ensemble learning methods.
- Random Forest is an ensemble learning method proposed by Leo Breiman in 2001.
- The main idea behind Random Forest:
  1. Take sample with replacement of size "n" from the original dataset (bootstraped sample)
  2. Grow a tree but at each possible split select only a subset of variables to possibly split on ($p/3$ for regression and $\sqrt{p}$ for classification). Typically grow bigger trees than we would for a single tree
  3. Repeat steps 1 and 2 many times (~1,000). The collection of trees is called a forest.
  4. Decide a final predicted outcome by combining the results across all of the trees (an average in regression)

Random Forest

- Lot's of different software implementations in R for random forests including randomforestSRC, randomForest, ranger
- Will focus on using the ranger package; this pacakge does not have great graphics internally so we will use the pdp package to supplement

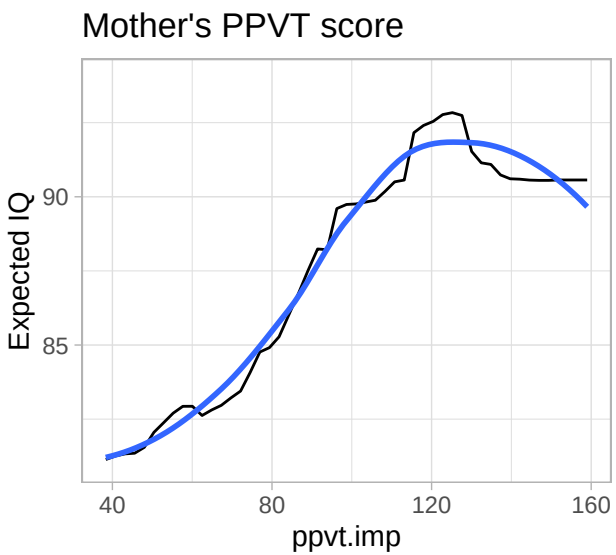
Random Forest implementation

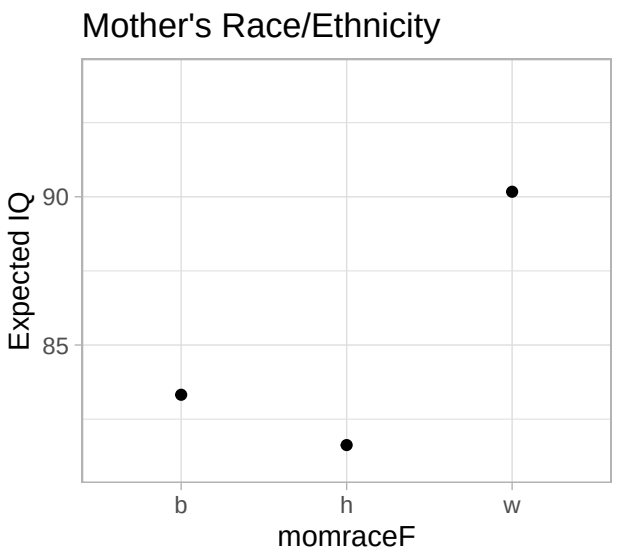
- Variable importance measures the variance of the responses for regression

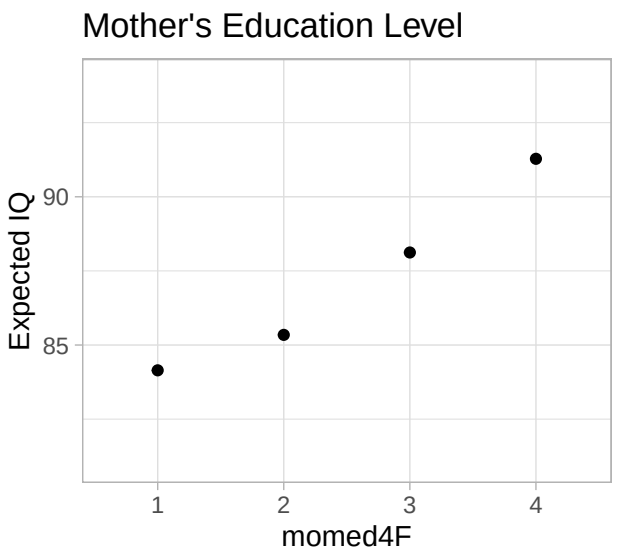


Partial dependence plot

- Heuristically, the partial dependence plots examine how the expected response changes as a function of a predictor, holding all other predictors constant

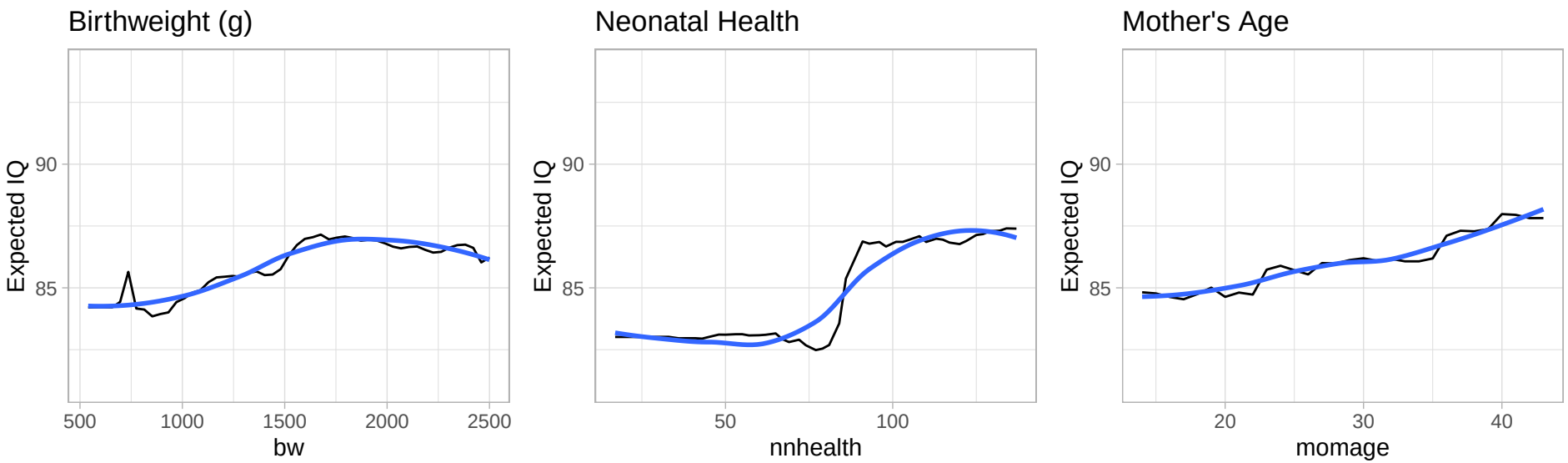






Partial dependence plot

- Heuristically, the partial dependence plots examine how the expected response changes as a function of a predictor, holding all other predictors constant



Obtain Predicted Value For Treatment/Control Group

- Note: should use out-of-bag (oob) estimates for those who actually received treatment level of interest

```
ihdp1 <- ihdp0 <- ihdp
ihdp1$treat <- 1; ihdp0$treat <- 0
pred1 <- predict(m2, data = ihdp1)$predictions
pred1[ihdp$treat == 1] <- m2$predictions[ihdp$treat == 1]
pred0 <- predict(m2, data = ihdp0)$predictions
pred0[ihdp$treat == 0] <- m2$predictions[ihdp$treat == 0]
```

Esimate of ATE

```
mean(pred1) - mean(pred0)
```

```
## [1] 7.724343
```

Use Bootstrap to Obtain Standard Error Estimates

- First, create function to return ATE

```
ate.rf <- function(data, freq){
  m2.boot <- ranger(iqsb.36 ~ .  - mlt.birtF - b.marryF - languageF, data=data,
  case.weights = freq, seed = 1101985)

  data1 <- data0 <- data
  data1$treat <- 1; data0$treat <- 0
  pred1 <- as.vector(predict(m2.boot, data = data1)$predictions)
  pred1[data$treat == 1] <- as.vector(m2.boot$predictions)[data$treat == 1]
  pred0 <- as.vector(predict(m2.boot, data = data0)$predictions)
  pred0[data$treat == 0] <- as.vector(m2.boot$predictions)[data$treat == 0]
  ate.rf <- weighted.mean(pred1, w = freq, na.rm = TRUE) -
    weighted.mean(pred0, w = freq, na.rm = TRUE)
  #return(c(m2.boot$predictions, pred1, pred0, freq))
  return(ate.rf)
}
```

Parallelizing with boot

```
set.seed(1101985) # bootstrapping is random - don't forget to set a seed!
start <- proc.time()
boot.results <- boot(data=ihdp, statistic=ate.rf, R = 1000,
  stype = "f", parallel="multicore", ncpus=8) # on Mac OS
proc.time() - start
```

```
##     user   system  elapsed
## 1245.329   52.544  230.754
```

```
boot.results
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = ihdp, statistic = ate.rf, R = 1000, stype = "f",
##     parallel = "multicore", ncpus = 8)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
## t1* 7.684887 -0.9728844    1.367006
```

Results Summary

```
## [1] "ATE (SE) = 7.7 (1.4)"
```

```
## [1] "95% CI (4, 9.6)"
```

Random Forest Part 2

- Some researchers have argued that should fit separate random forests in the treatment and control groups
- Effectively this is the same as splitting on treatment first
- May not be wise when one group (treat $= 1$ here) is fairly small

```
## [1] "ATE (SE) = 15.1 (1.6)"
```

```
## [1] "95% CI (12.3, 18.6)"
```

Putting it All Together

Method	ATE
Unadjusted Analysis	16.8
Main Effects, Linear Only	13.7
Main Effects, RCS	13.6
Random Forest	7.7
Random Forest; Separate Models	15.1