Gov 51: Nonlinear Relationships

Matthew Blackwell

Harvard University

Social pressure experiment

- We'll look at the Michigan experiment that was trying to see if social pressure affects turnout.
- Load the data and create an age variable:

```
social <- read.csv("data/social.csv")
social$age <- 2006 - social$yearofbirth
summary(social$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    20.0    41.0    50.0    49.8    59.0   106.0
```

```
social.neighbors <- subset(social,
                           neighbors == 1 | control == 1)
```

Linear regression are linear

$$\widehat{Y}_i = \widehat{\alpha} + \widehat{\beta}_1 X_i$$

- Standard linear regression can only pick up **linear** relationships.
- What if the relationship between X_i and Y_i is nonlinear?

Adding a squared term

- To allow for nonlinearity in age, add a squared term to the model:

$$\widehat{Y}_i = \widehat{\alpha} + \widehat{\beta}_1 \text{age}_i + \widehat{\beta}_2 \left(\text{age}_i^2 \right)$$

- We are now fitting a **parabola** to the data.
- In R, we need to wrap the squared term in `I()`:

```
fit.sq <- lm(primary2006 ~ age + I(age^2), data = social)
coef(fit.sq)
```

```
## (Intercept)         age    I(age^2)
##  -0.0816804   0.0122736  -0.0000808
```

- $\widehat{\beta}_2$: how the effect of age increases as age increases.

Predicted values from lm()

- We can get predicted values out of R using the `predict()` function:

```
predict(fit.sq, newdata = list(age = c(20, 21, 22)))
```

```
##     1     2     3
## 0.131 0.140 0.149
```

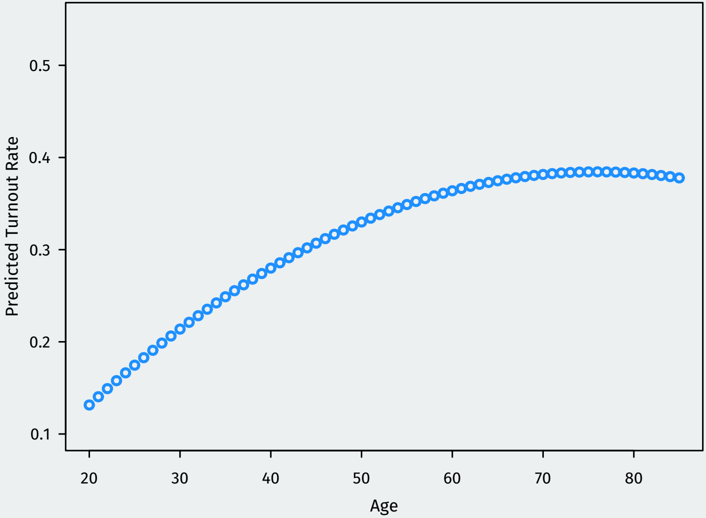
- Create a vector of ages to predict and save predictions:

```
age.vals <- 20:85
age.preds <- predict(fit.sq, newdata = list(age = age.vals))
```

- Plot the predictions:

```
plot(x = age.vals, y = age.preds, ylim = c(0.1, 0.55),
     xlab = "Age", ylab = "Predicted Turnout Rate",
     col = "dodgerblue", lwd = 2)
```

Plotting predicted values

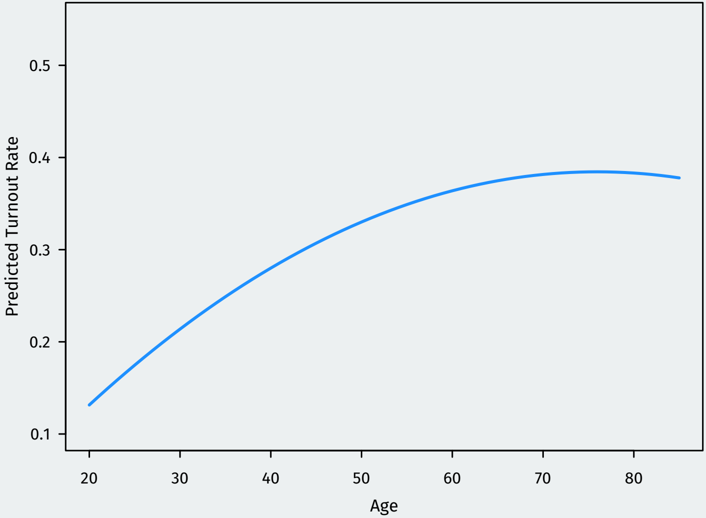


Plotting lines instead of points

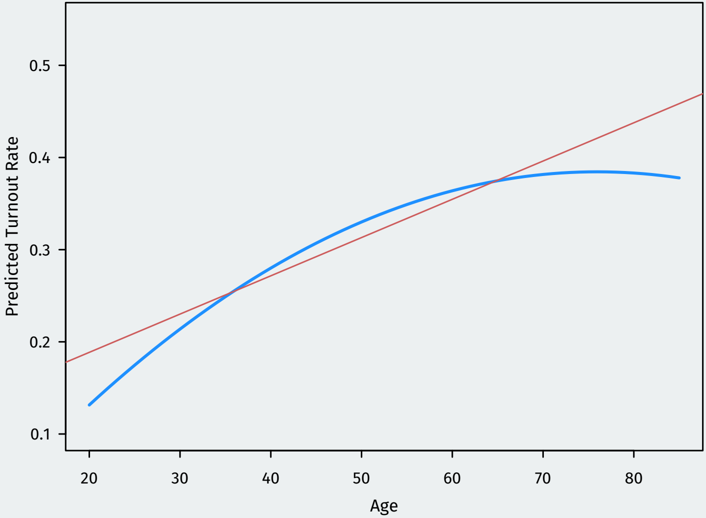
- If you want to connect the dots in your scatterplot, you can use the `type = "l"` ("line" type):

```
plot(x = age.vals, y = age.preds, ylim = c(0.1, 0.55),
     xlab = "Age", ylab = "Predicted Turnout Rate",
     col = "dodgerblue", lwd = 2, type = "l")
```

Plotting predicted values



Comparing to linear fit



Diagnosing nonlinearity

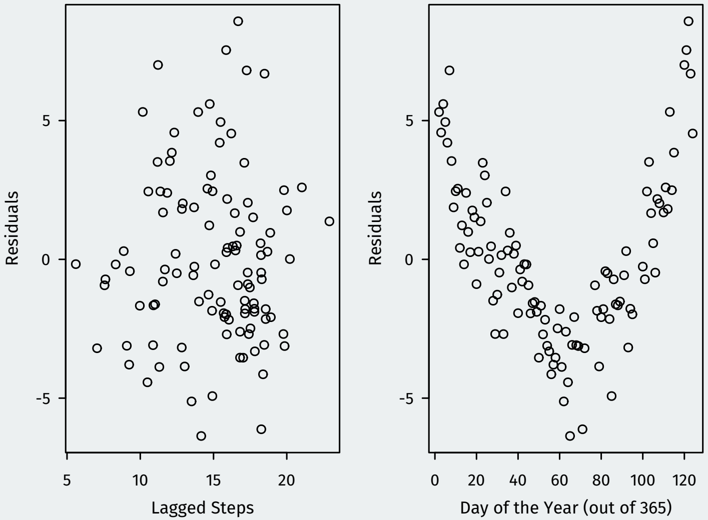
- One independent variable: just look at a scatterplot.
- With multiple independent variables, harder to diagnose.
- One useful tool: scatterplot of residuals versus independent variables.
- Example: my weight again

```
health <- read.csv("data/health2017.csv")
w.fit <- lm(weight ~ steps.lag + dayofyear, data = health)
```

Residual plot

```
plot(health$steps.lag, residuals(w.fit),
     xlab = "Lagged Steps", ylab = "Residuals")
plot(health$dayofyear, residuals(w.fit),
     xlab = "Day of the Year (out of 365)",
     ylab = "Residuals")
```

Residual plot



Add a squared term for a better fit

```
w.fit.sq <- lm(weight ~ steps.lag + dayofyear + I(dayofyear^2),
               data = health)
coef(w.fit.sq)
```

```
##    (Intercept)       steps.lag       dayofyear
##       177.4679          0.0521         -0.4439
## I(dayofyear^2)
##         0.0024
```

```
plot(health$steps.lag, residuals(w.fit.sq),
     xlab = "Lagged Steps", ylab = "Residuals")
plot(health$dayofyear, residuals(w.fit.sq),
     xlab = "Day of the Year (out of 365)",
     ylab = "Residuals")
```

Residual plot, redux