Lab 6: Nonlinear Models

We will continue to use the `Wage` data set in the `ISLR` package to predict `wage` for $3,000$ mid-atlantic male workers.

```
library(ISLR)
library(tidyverse)
library(knitr)

str(Wage)
```

```
## 'data.frame':    3000 obs. of  11 variables:
##  $ year      : int  2006 2004 2003 2003 2005 2008 2009 2008 2006
2004 ...
##  $ age       : int  18 24 45 43 50 54 44 30 41 52 ...
##  $ maritl    : Factor w/ 5 levels "1. Never Married",..: 1 1 2 2 4 2
2 1 1 2 ...
##  $ race      : Factor w/ 4 levels "1. White","2. Black",..: 1 1 1 3
1 1 4 3 2 1 ...
##  $ education : Factor w/ 5 levels "1. < HS Grad",..: 1 4 3 4 2 4 3 3
3 2 ...
##  $ region    : Factor w/ 9 levels "1. New England",..: 2 2 2 2 2 2 2
2 2 2 ...
##  $ jobclass  : Factor w/ 2 levels "1. Industrial",..: 1 2 1 2 2 2 1
2 2 2 ...
##  $ health    : Factor w/ 2 levels "1. <=Good","2. >=Very Good": 1 2
1 2 1 2 2 1 2 2 ...
##  $ health_ins: Factor w/ 2 levels "1. Yes","2. No": 2 2 1 1 1 1 1 1
1 1 ...
##  $ logwage   : num  4.32 4.26 4.88 5.04 4.32 ...
##  $ wage      : num  75 70.5 131 154.7 75 ...
```

0.1 Polynomial Regression and Step Functions

1. Fit a degree-4 polynomial regression model predicting `wage` based on `age`. Inspect your model and describe the fit. [**Hint**: you can use the `step_poly` function to create your polynomials.]

2. Choose your degree of polynomial using a cross validation approach. What degree model would you pick?

3. Fit a step function for `age` predicting `wage` with 4 cut points. You can use the function `step_discretize` to change your quantitative variable into a categorical one. Let `step_discretize` automatically choose the cut locations based on your data.

0.2 Regression Splines

To fit regression splines, we will use `step_bs` and `step_ns` in the recipe. The `step_bs` function generates a matrix of basis functions for regression splines (defaults cubic) based on a vector of knots or a specified degree of freedom. The `ns` function is the same for natural splines.

We can use either of these functions with our usual linear model.

```
linear_spec <- linear_reg()

## automatically chosen knots
spline_rec <- recipe(y ~ x, data = df) |>
  step_bs(degree = 3, deg_free = 6) ## cubic spline with 2 knots &
        intercept

## user specified knots
spline_rec2 <- recipe(y ~ x, data = df) |>
  step_bs(degree = 3, options = list(knots = c(0, 5))) ## cubic spline
        with 2 knots & intercept

bs_workflow <- workflow() |>
  add_model(linear_spec) |>
  add_recipe(spline_rec)

bs_fit <- fit(bs_workflow, data = df)
```

1. Fit `wage` on `age` using a cubic regression spline with knots at ages $25, 40, 60$.

2. Fit `wage` on `age` using a cubic regression spline with 6 degrees of freedom and knots chosen uniformly on the quantiles of the data (this is how `step_bs` does it by default).

3. Fit `wage` on `age` using a natural cubic regression spline with 6 degrees of freedom and knots chosen uniformly on the quantiles of the data.

4. Create a scatter plot of `wage` vs `age` with all three of your fitted splines overlayed as well as your chosen polynomial model (either by anova or CV). Comment on the

shapes. [**Hint:**`predict` over a grid of `age` values might be helpful.]

0.3 GAMs

1. Fit a GAM using natural spline functions of `year` and `age`, treating `education` as a categorical predictor.