Lec 04: Indicator variables and Splines

MATH 456 - Spring 2016

Assigned Reading

Afifi: Chapter 9.3, Harrel Ch 2

Factor variable coding

- Better used term: Indicator variable
- Math notation: **I(gender == "Female")**.
- A.k.a reference coding
- For a nominal X with K categories, define K indicator variables.
  - Choose a reference (referent) category:
  - Leave it out
  - Use remaining K-1 in the regression.
  - Often, the largest category is chosen as the reference category.

Example: Binary indicator for gender

Consider the linear model of FEV on gender(x_1), height(x_2) and age(x_3) where gender interacts with both age and height. In other words, gender changes the relationship between height and FEV1, and the relationship between age and FEV1.

$$FEV1 \sim \beta_0 + \beta_1 * gender + \beta_2 * height + \beta_3 * age + \beta_4 * gender * height + \beta_5 * gender * age$$

If we let gender = 0 if the record is on a male, and gender = 1 if the record is on a female, then the model for males would be:

$$FEV1 \sim \beta_0 + \beta_2 * height + \beta_3 * age$$

and the model for females would be:

$$FEV1 \sim (\beta_0 + \beta_1) + (\beta_2 + \beta_4) * height + (\beta_3 + \beta_5) * age$$

Example: Religion against income and depression

Consider a log-linear model for the effect of marital status (X_2) on log income while controlling for age(X_1). This is called a log-linear model because the outcome has been log transformed.

$$log(Y_i) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2$$

```r
dep <- read.table("C:/GitHub/MATH456/data/Depress_020916.txt", sep="\t", header=TRUE)
names(dep) <- tolower(names(dep)) # I hate all captal variable names
levels(dep$marital)
```

```
## [1] "Divorced"      "Married"       "Never Married" "Separated"
## [5] "Widowed"
```

Marital status has 5 levels, so we would need 4 indicator variables. R always uses the first level of a factor variable as the reference level.

- Let $x_2 = 1$ when `marital='Married'`, and 0 otherwise,
- let $x_3 = 1$ when `marital='Never Married'`, and 0 otherwise,
- let $x_4 = 1$ when `marital='Separated'`, and 0 otherwise,
- let $x_5 = 1$ when `marital='Widowed'`, and 0 otherwise.

The mathematical model would look like:

$$log(Y)|X \sim \beta_0 + \beta_1 * x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5$$

Two levels of interpretation here.

1. The outcome is log transformed, so the interpretation has to be back-transformed.
2. The coefficients for the other levels of the categorical variable are in *comparison* to the reference level.

Interpretation of log-linear models Calculate the change in Y that corresponds to a one unit change in x_1. Since marital status is remaining constant, I will exclude it from the calculations below to save space and not to detract from the main point.

Write each equation down

$$log(Y)|x_1 = \beta_0 + \beta_1 x_1$$
$$log(Y)|(x_1 + 1) = \beta_0 + \beta_1 (x_1 + 1)$$

Find the difference

$$(log(Y)|x_1) - (log(Y)|(x_1 + 1)) = (\beta_0 + \beta_1 x_1) - (\beta_0 + \beta_1 (x_1 + 1))$$

and simplify.

$$log(\frac{Y|x_1}{Y|x_1 + 1}) = \beta_1$$
$$\frac{Y|x_1}{Y|x_1 + 1} = e^{\beta_1}$$

Each 1-unit increase in x_j multiplies the expected value of Y by $e^{\hat{\beta}_j}$.

Interpretation: $100\hat{\beta}_j$ is the expected **percentage** change in Y for a unit increase in x_j.

The nice thing about factor variables in R, is that the appropriate indicator variables are automatically created for you by the linear model (`lm()`) function.

```
summary(lm(log(income) ~ age + marital,data=dep))
```

```
##
## Call:
## lm(formula = log(income) ~ age + marital, data = dep)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62643 -0.46829  0.01535  0.45280  1.48175
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.065696   0.166760  18.384  < 2e-16 ***
## age                    -0.009043   0.002919  -3.098 0.002143 **
## maritalMarried          0.416653   0.124698   3.341 0.000944 ***
## maritalNever Married   -0.183156   0.141354  -1.296 0.196109
## maritalSeparated       -0.394544   0.223431  -1.766 0.078482 .
## maritalWidowed         -0.278352   0.173159  -1.607 0.109042
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7059 on 288 degrees of freedom
## Multiple R-squared:  0.1971, Adjusted R-squared:  0.1832
## F-statistic: 14.14 on 5 and 288 DF,  p-value: 2.236e-12
```

- For every year older, a persons income decreases by 1%. (`exp(-0.009)` = 0.99)
- Married individuals have a 52% higher income compared to those who are divorced. (`exp(-0.417)` = 1.52)
- Those who have never been married have 16% lower income compared to those who are divorced. (`exp(-0.183)` = 0.83)
- Separated individuals have 32% lower income compared to those who are divorced. (`exp(-0.394)` = 0.67)
- Widowed individuals have 24% lower income compared to those who are divorced. (`exp(-0.278)` = 0.76)

Other references on how to interpret regression parameters when they have been log transformed:

- http://www.ats.ucla.edu/stat/mult_pkg/faq/general/log_transformed_regression.htm
- http://www.kenbenoit.net/courses/ME104/logmodels2.pdf

On Your Own

On Your Own Create a model to analyze the relationship of education status to depression level as measured by CESD after controlling for age. Combine all education levels below a HS graduate into one reference category called "Up to HS" prior to analysis.

This is a seemingly simple request, but there are a lot of steps you must do to correctly analyze this question.

1. Ensure that you are using the analyzable version of the depression data set. It may be helpful to confirm that your recodes are correct by comparing your data management code file to mine dm_depress located on our course website.
2. Reference your Ch3 homework (or the solutions) if you need help collapsing educational categories.
3. Ensure that R is treating "Up to HS" as the reference category for education level. If it is not, use the `levels` argument of the `factor()` function to reorder your factor levels. This is also presented in the Ch3 solutions.
4. Consider a transformation of `CESD`. Explain and justify using graphical measures why you chose to, or chose not to, transform CESD prior to modeling.
5. Check the model fit by examining the residuals to see if the assumption that $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is upheld.
6. Identify any potential outliers. Explain why you think they are outliers. Examine their standardized residuals and leverage values.If any seem to stand out or have high values for either measure, exclude them from the analysis and re-run the model.
7. Once you have finalized your model, interpret ALL coefficients in context of the problem. State if any are significantly predictive of the outcome, provide p-values in your conclusion.
8. Does this model do well at all in predicting CESD? Answer this question using both the coefficient of determination and the ANOVA test of overall global fit (testing that all β's are 0)

Splines & other non-linear terms

References

- Afifi Section 9.4
- Harrell 2.4.3, pg 39 http://biostat.mc.vanderbilt.edu/tmp/course.pdf
- Harrell ch2 from second edition pdf in shared GDrive.
- https://www.youtube.com/watch?v=o_d4hmKhmsQ
- http://www.r-bloggers.com/thats-smooth/

[top]

Example 1: Simulated data. Example data pulled from _http://faculty.washington.edu/heagerty/Courses/b571/homework/spline-tutorial.q_

Suppose we have a predictor that takes the values 1:24

```
x <- c(1:24)
```

and there is an outcome variable that is predicted by the variable X, but in some non-linear fashion:

```
mu <- 10 + 5 * sin( x * pi / 24 )  - 2 * cos( (x-6)*4/24 )
```

But there is always some amount of error associated with real data.

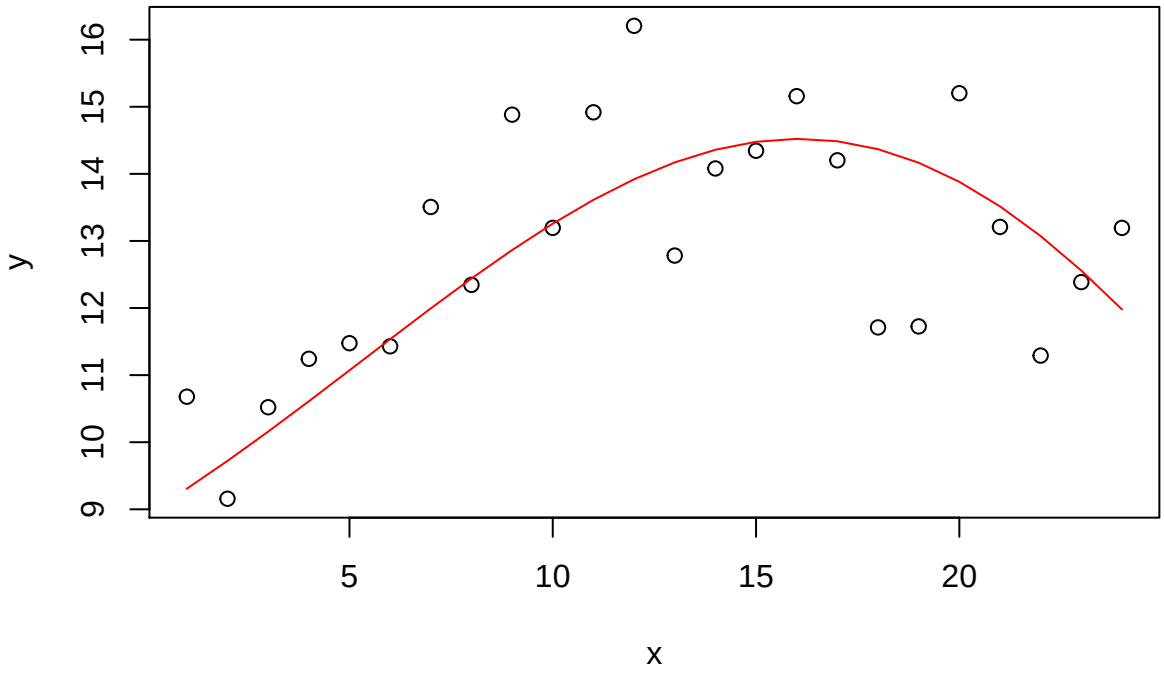
```
set.seed(42)
eee <- rnorm(length(mu))
```

So our simulated data then is the true trend + the noise.

```
y <- mu + eee
```

Let's look at the data, and the real mean trend without the random noise.

```
plot(y~x)
lines(x, mu, col="red" )
```



Let's look at ways to fit a model to this data.

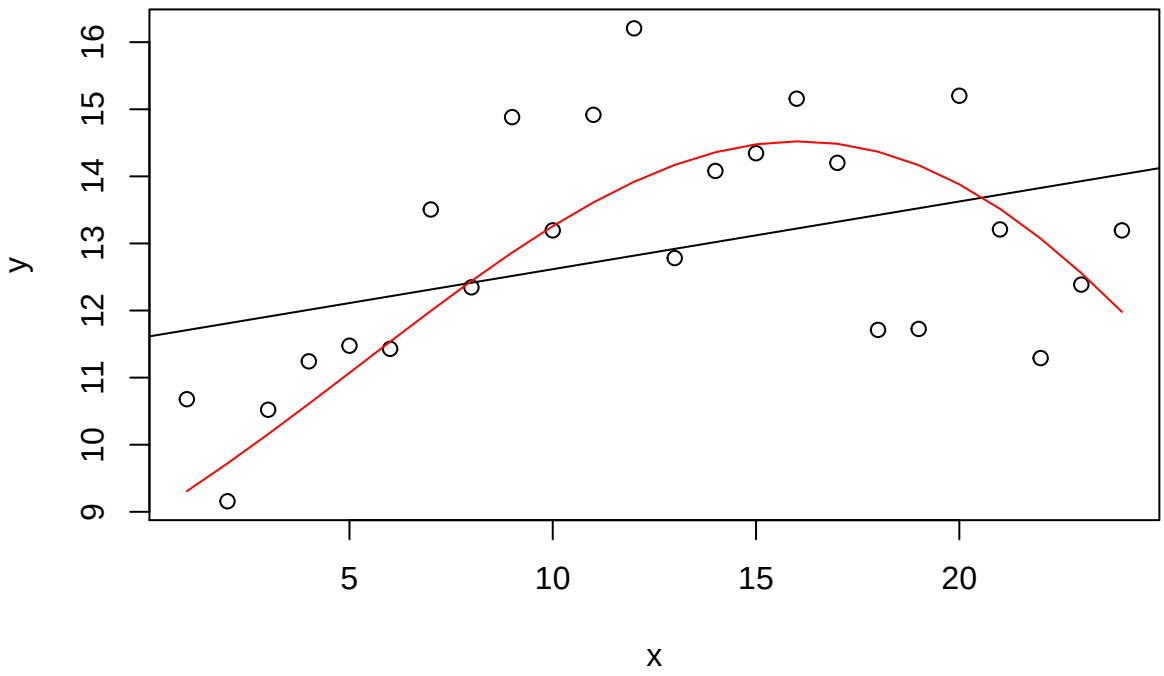
Linear

[top]

Ignore the trend and fit a linear model.

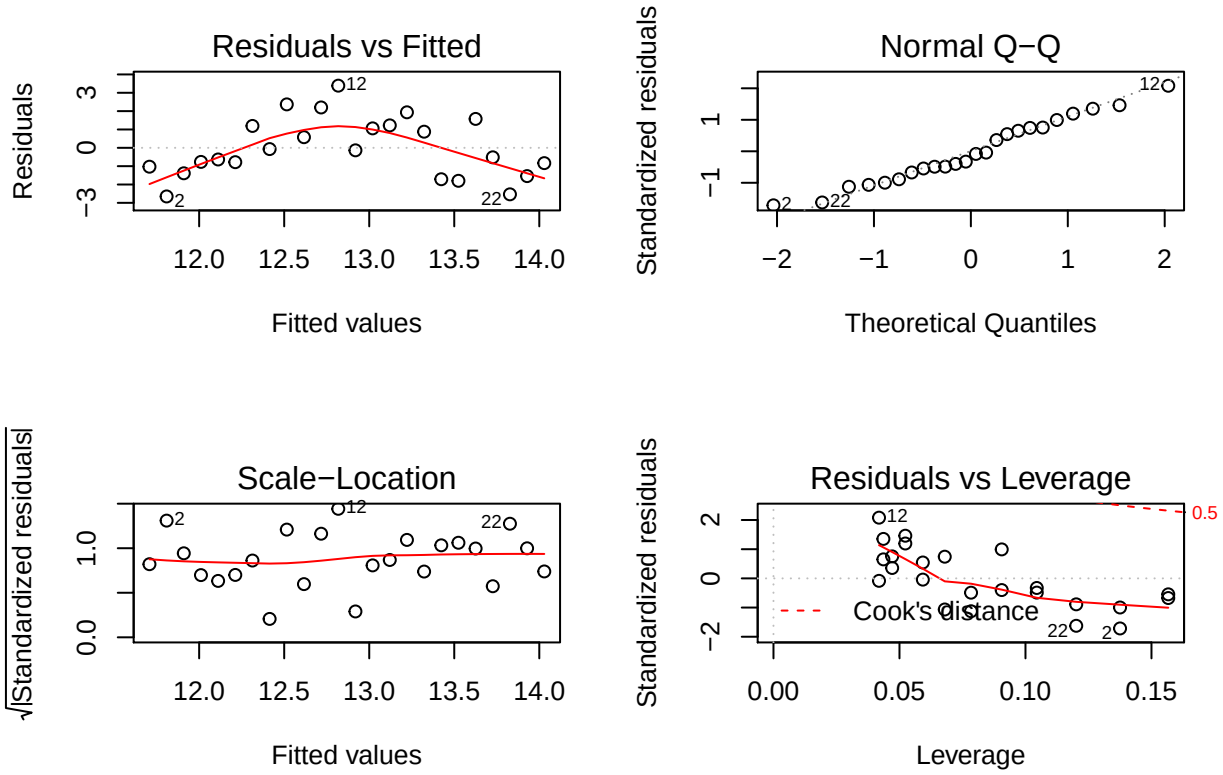
$$E(Y|X) = \beta_0 + \beta_1 X$$

```
fit.slr <- lm(y~x)
plot(y~x)
abline(fit.slr)
lines(x, mu, col="red" )
```



Undoubtedly not a good fit. Examining the residuals shows the non-constant variance clearly.

```
par(mfrow=c(2,2))
plot(fit.slr)
```



Piecewise linear splines

[top]

We allow the x axis to be divided into intervals, with a linear model fit within each interval. The breakpoints between intervals are called *knots*. This is where you are allowing the slope of the line to change. For example to break the x-axis into three sections we would use 2 knots. The model would look like.

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 (X-a)_+ + \beta_3 (X-b)_+$$

where $(u)_+$ contains the value of u when u is positive, and 0 otherwise.

Let's put knots at 6, 12, and 18.

```
x6 <- (x-6)
x6[ x6<0 ] <- 0

x12 <- (x-12)
x12[ x12<0 ] <- 0

x18 <- (x-18)
x18[ x18<0 ] <- 0
```

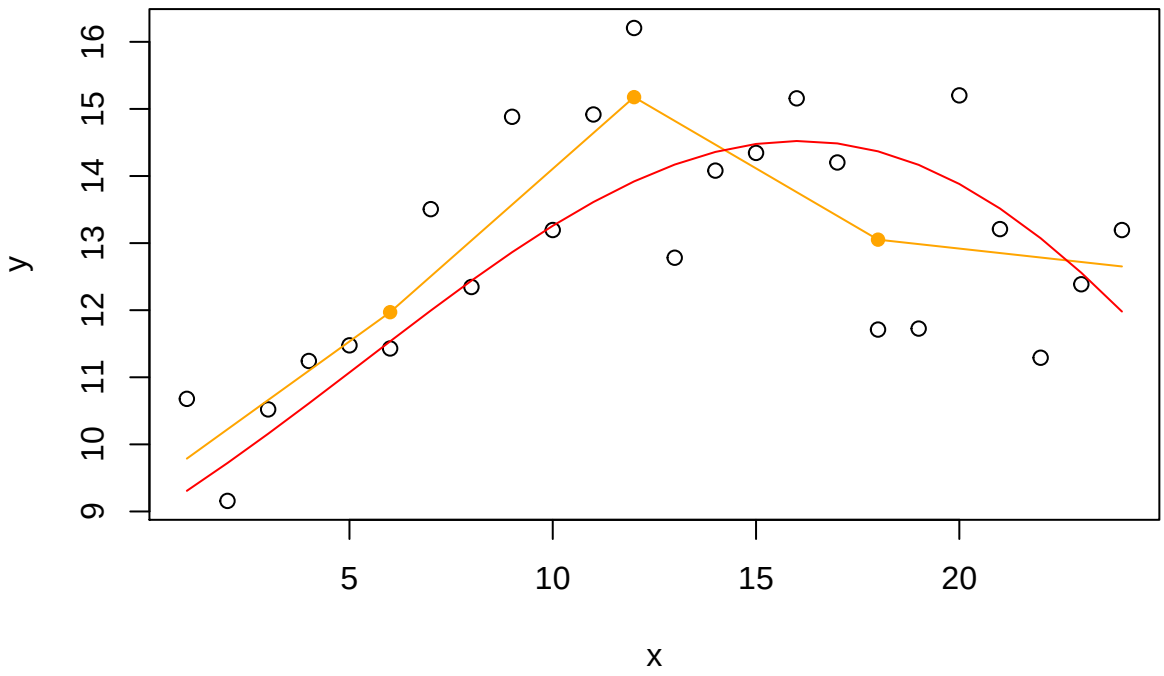
What does this data look like now?

```
t(cbind(x, x6, x12, x18)[8:20,])
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## x       8    9   10   11   12   13   14   15   16    17    18    19    20
## x6      2    3    4    5    6    7    8    9   10    11    12    13    14
## x12     0    0    0    0    0    1    2    3    4     5     6     7     8
## x18     0    0    0    0    0    0    0    0    0     0     0     1     2
```

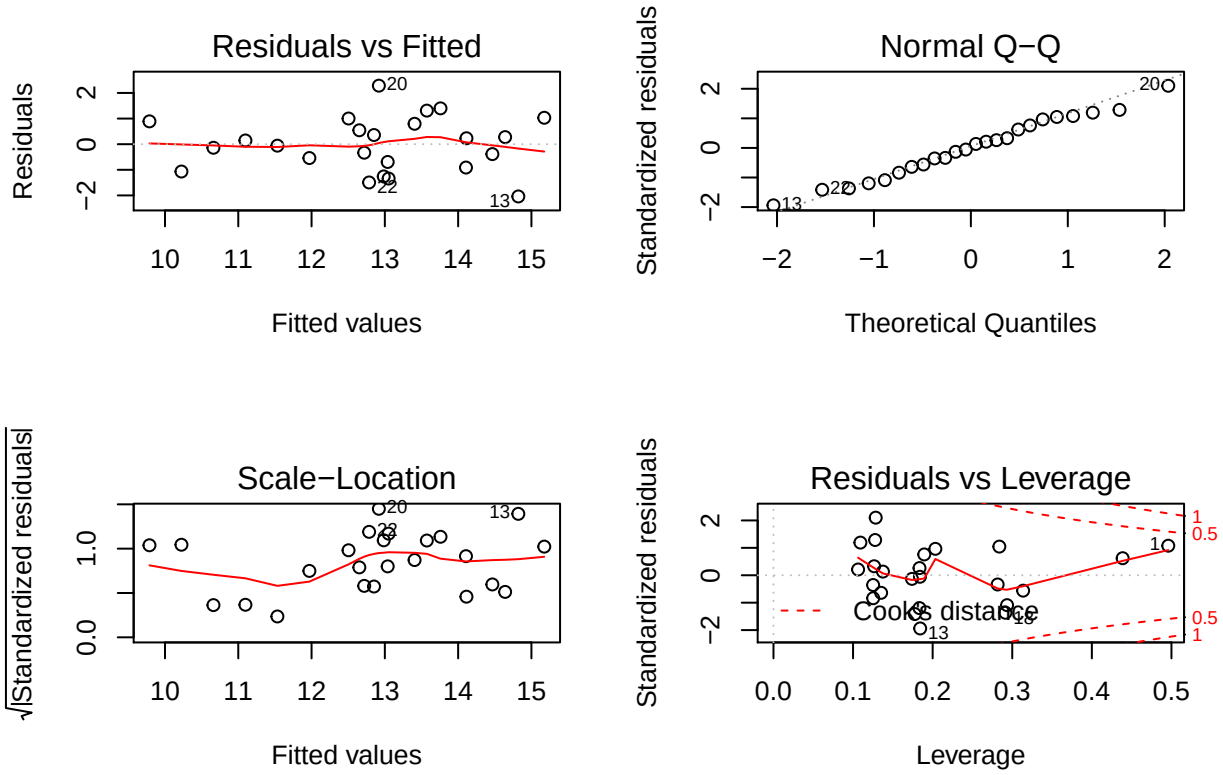
Now let's fit this model.

```
fit.lin.spline <- lm(y ~ x + x6 + x12 + x18)
plot(y~x)
lines(x, predict(fit.lin.spline), col="orange")
points(c(6, 12, 18), predict(fit.lin.spline)[c(6, 12, 18)], pch=16, col="orange")
lines(x, mu, col="red" )
```



Much closer than the linear model, but it still lacks the curvature that is present in the data. The residual plots look much better already.

```
par(mfrow=c(2,2))
plot(fit.lin.spline)
```



Powers

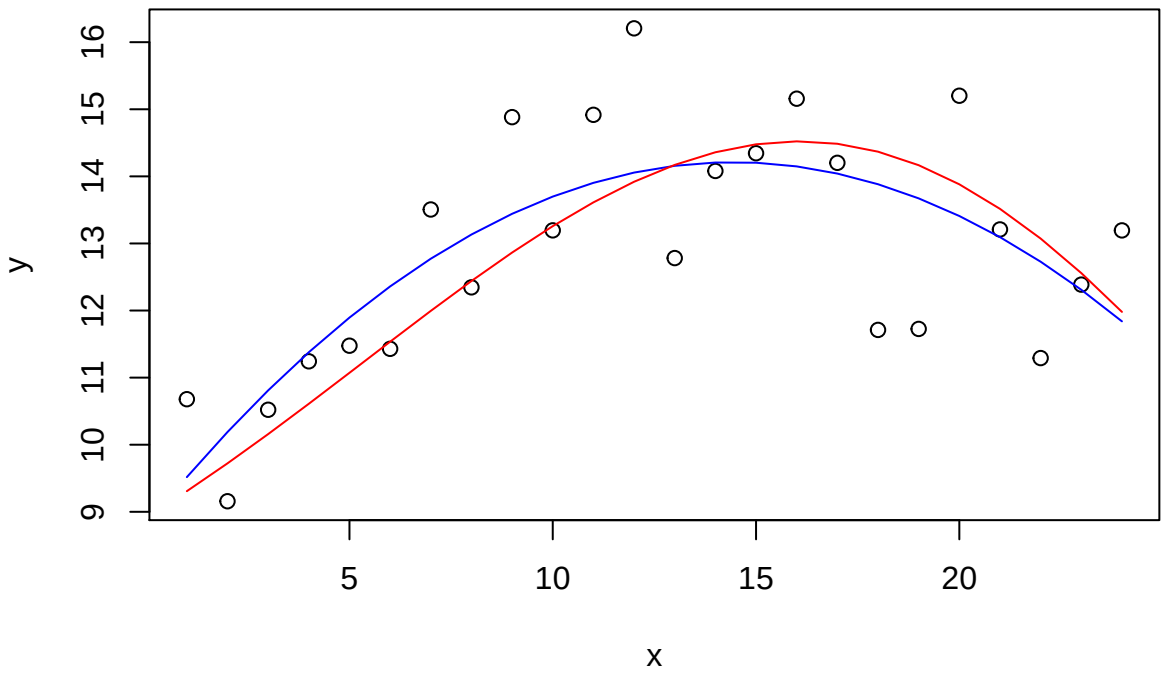
[top]

A non-linear effect can be as simple as adding a covariate at some power.

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 X^2$$

Testing $H_0 : \beta_2 = 0$ tests the null hypothesis that the effect of X_1 on Y is linear vs the effect is quadratic.

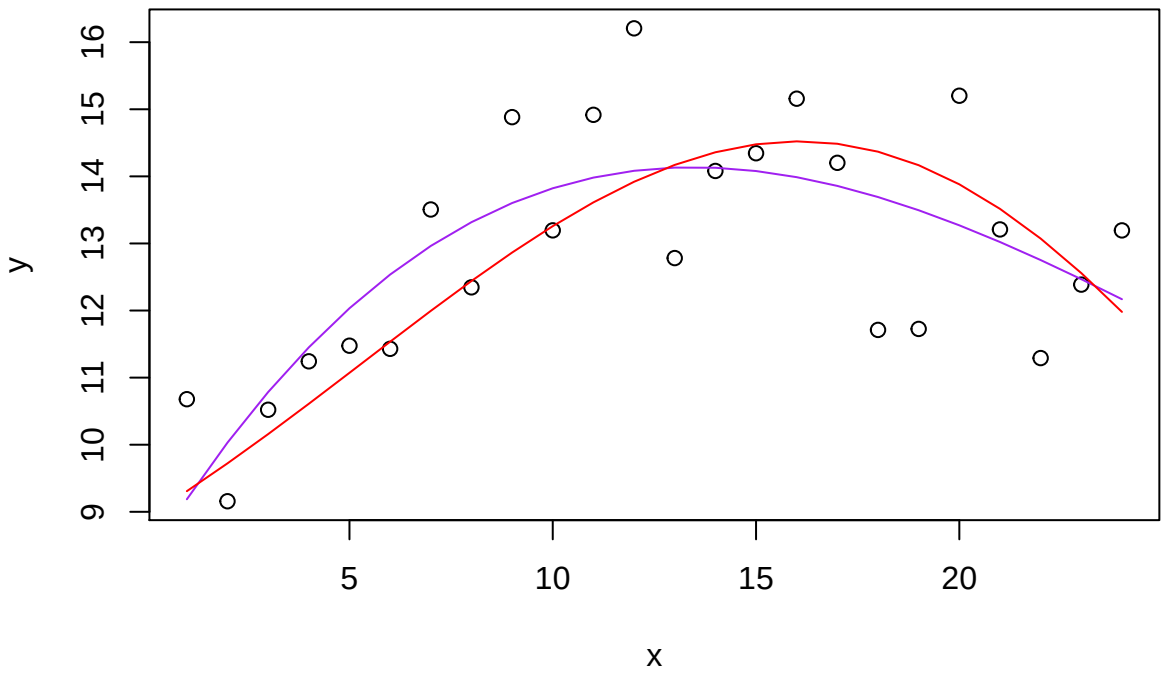
```
x.squared <- x^2
fit.sq <- lm(y~x + x.squared)
plot(y~x)
lines(x, predict(fit.sq), col="blue")
lines(x, mu, col="red" )
```



A cubic term could also be added.

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 x^3$$

```
x.cubed <- x^3
fit.cubic <- lm(y~x + x.squared + x.cubed)
plot(y~x)
lines(x, predict(fit.cubic), col="purple")
lines(x, mu, col="red" )
```



Adding this cubic term allows for another "wiggle" in the fitted line.

Cubic splines

[top]

Combining the two concepts allows for a very flexible polynomial model.

$$E(Y|X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 (X-a)_+^3 + \beta_5 (X-b)_+^3$$

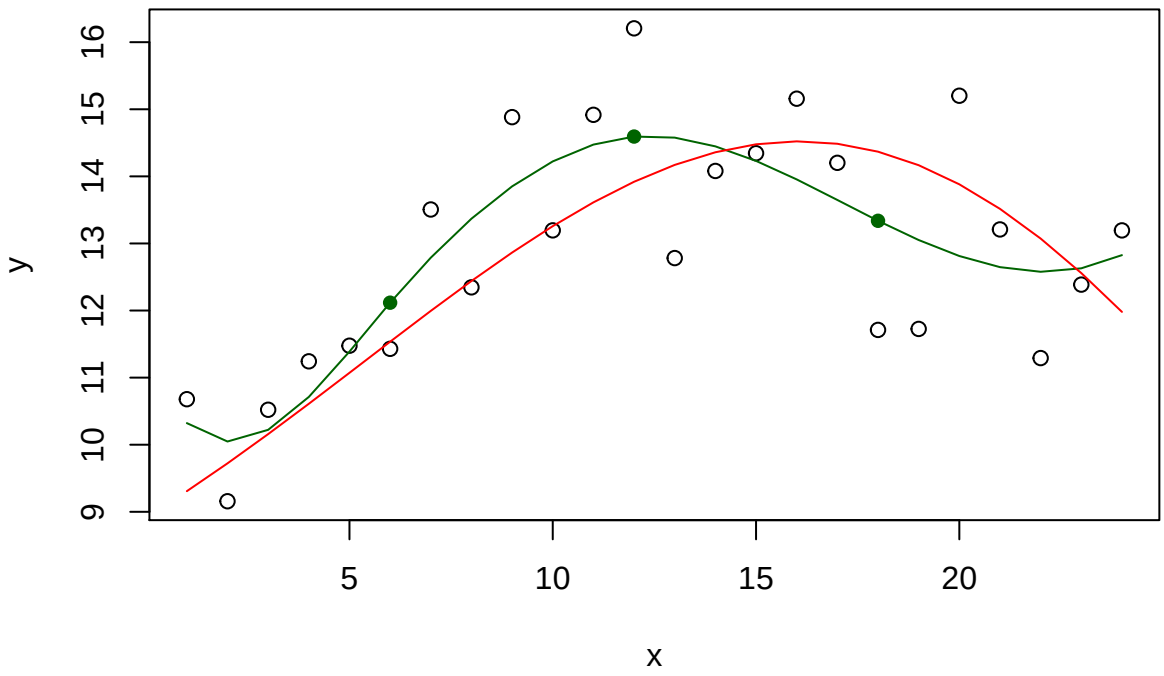
Using the knots at 6, 12, and 18 let's fit a cubic spline.

```
x6.cubed <- x6^3
x12.cubed <- x12^3
x18.cubed <- x18^3

fit.cub.spline <- lm(y ~ x + x.squared + x.cubed + x6.cubed + x12.cubed + x18.cubed)
```

Replot and look at the fitted model.

```
plot(y~x)
lines(x, predict(fit.cub.spline), col="darkgreen")
points(c(6, 12, 18), predict(fit.cub.spline)[c(6, 12, 18)], pch=16, col="darkgreen")
lines(x, mu, col="red" )
```



It seems like our model is fitting the data better, but sometimes there is a balance between a flexible model, and overfitting the data (when your model fits each point better than the true underlying average.)

Natural splines

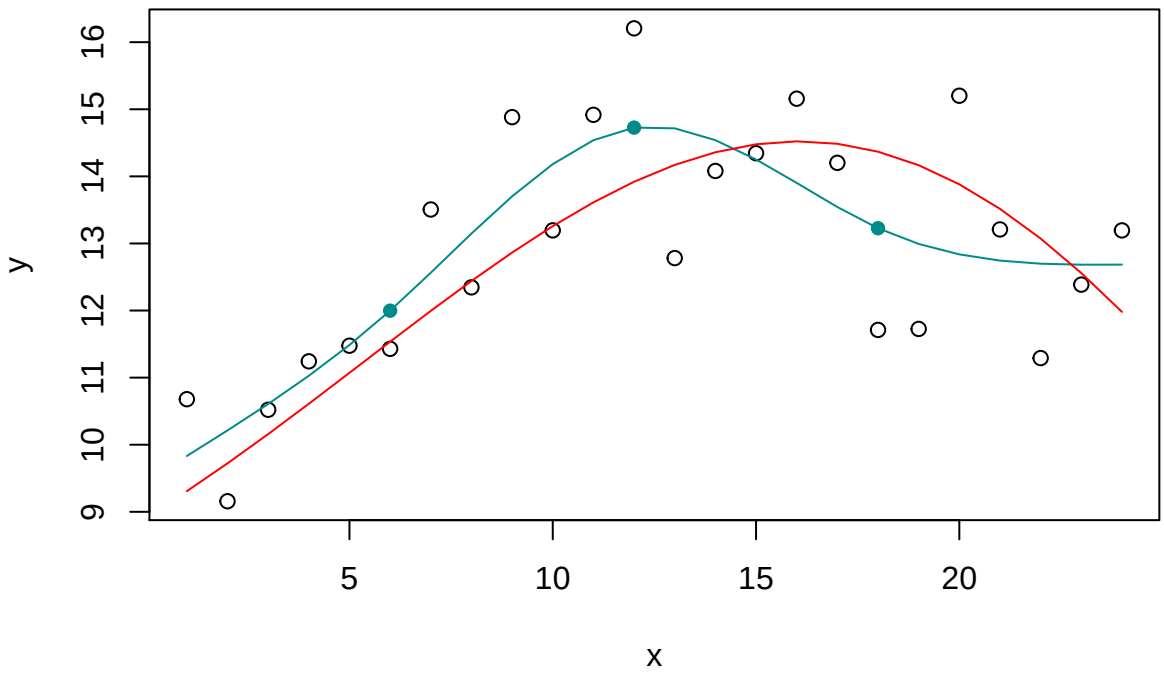
[top]

Also called *natural splines*, these models constrain the model to be linear in the tails. The model is difficult to write, and fit by hand so we will use the `splines` package.

```
library(splines)

fit.ns = lm( y ~ ns(x, knots=c(6,12,18) ) )

plot(y~x)
lines(x, predict(fit.ns), col="darkcyan")
points(c(6, 12, 18), predict(fit.ns)[c(6, 12, 18)], pch=16, col="darkcyan")
lines(x, mu, col="red" )
```



There are other methods of model fitting under the umbrella of *Nonparametric Regression*, these include kernel smoothing, smoothing splines, and the familiar LOWESS (locally weighted scatterplot smoothing) and LOESS (Local regression) models.

[top]

On Your Own

On Your Own

1. Using the `cars` data set built into R, build a model to predict the distance a car takes to stop based on how fast it was going.

2. Using the family lung function data, build a model to predict FEV1 to height for the oldest child.