Question 15: Problem 8 in Section 5.4

Julia Lee

10/12/2018

A

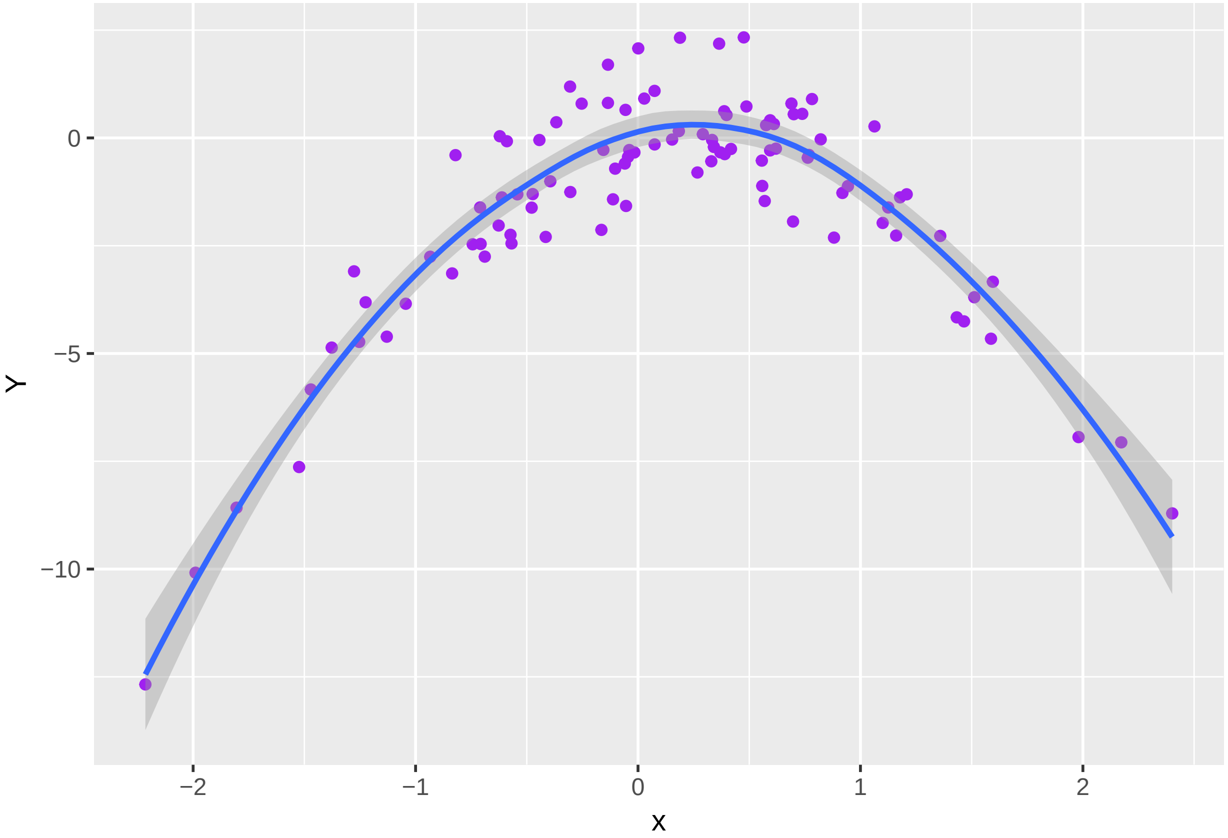
```
set.seed(1)
x <-rnorm(100)
Y<- x-(2*x^2)+rnorm(100)
```

$n = 100$ There are two parameters (x , $2x^2$) so $p = 2$ $y = x - 2(x^2) + \epsilon$

B

```
set.seed(1)
ggplot()+
  geom_point(aes(x=x,y=Y), color = "purple") +
  geom_smooth(aes(x=x,y=Y))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



There is what appears to be a negative quadratic relationship between X and Y. This makes sense because y = x - 2(x^2).

C

```
set.seed(64)
frame <- data.frame(Y, x )
mod_1 <- glm(Y~poly(x,1), data=frame)
mod_2 <- glm(Y~poly(x,2), data=frame)
mod_3 <-  glm(Y~poly(x,3), data=frame)
mod_4 <- glm(Y~poly(x,4), data=frame)

deltas = data.frame(delta1=0, delta2=0)
cv_error_LR <- cv.glm(frame, mod_1)$delta[1]
cv_error_quad <- cv.glm(frame, mod_2)$delta[1]
cv_error_cube <- cv.glm(frame, mod_3)$delta[1]
cv_error_4 <- cv.glm(frame, mod_4)$delta[1]



cv_error_LR
```

```
## [1] 7.288162
```

```
cv_error_quad
```

```
## [1] 0.9374236
```

```
cv_error_cube
```

```
## [1] 0.9566218
```

```
cv_error_4
```

```
## [1] 0.9539049
```

```
set.seed(1997)
frame <- data.frame(Y, x )
mod_1 <- glm(Y~poly(x,1), data=frame)
mod_2 <- glm(Y~poly(x,2), data=frame)
mod_3 <-  glm(Y~poly(x,3), data=frame)
mod_4 <- glm(Y~poly(x,4), data=frame)

deltas = data.frame(delta1=0, delta2=0)
cv_error_LR <- cv.glm(frame, mod_1)$delta[1]
cv_error_quad <- cv.glm(frame, mod_2)$delta[1]
cv_error_cube <- cv.glm(frame, mod_3)$delta[1]
cv_error_4 <- cv.glm(frame, mod_4)$delta[1]



cv_error_LR
```

```
## [1] 7.288162
```

```
cv_error_quad
```

```
## [1] 0.9374236
```

```
cv_error_cube
```

```
## [1] 0.9566218
```

```
cv_error_4
```

```
## [1] 0.9539049
```

E

The seed does not matter because LOOCV does not randomly divide into test and train. Here we see a sharp drop in the estimated test MSE between linear and quadratic fits. MSE increases again for the 3rd and 4th degree models. The quadratic model has the lowest MSE which makes sense because the underlying relationship between x and y is quadratic.

F

```
summary(mod_1)
```

```
##
## Call:
## glm(formula = Y ~ poly(x, 1), data = frame)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -9.5161  -0.6800   0.6812   1.5491   3.8183
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.550      0.260  -5.961 3.95e-08 ***
## poly(x, 1)     6.189      2.600   2.380   0.0192 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6.760719)
##
##     Null deviance: 700.85  on 99  degrees of freedom
## Residual deviance: 662.55  on 98  degrees of freedom
## AIC: 478.88
##
## Number of Fisher Scoring iterations: 2
```

```
summary(mod_2)
```

```
##
## Call:
## glm(formula = Y ~ poly(x, 2), data = frame)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.9650  -0.6254  -0.1288   0.5803   2.2700
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.5500     0.0958  -16.18  < 2e-16 ***
## poly(x, 2)1   6.1888     0.9580    6.46 4.18e-09 ***
## poly(x, 2)2 -23.9483     0.9580  -25.00  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9178258)
##
##     Null deviance: 700.852  on 99  degrees of freedom
## Residual deviance:  89.029  on 97  degrees of freedom
## AIC: 280.17
##
## Number of Fisher Scoring iterations: 2
```

```
summary(mod_3)
```

```
##
## Call:
## glm(formula = Y ~ poly(x, 3), data = frame)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9765  -0.6302  -0.1227   0.5545   2.2843
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.55002    0.09626 -16.102  < 2e-16 ***
## poly(x, 3)1   6.18883    0.96263   6.429 4.97e-09 ***
## poly(x, 3)2 -23.94830    0.96263 -24.878  < 2e-16 ***
## poly(x, 3)3   0.26411    0.96263   0.274    0.784
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9266599)
##
##     Null deviance: 700.852  on 99  degrees of freedom
## Residual deviance:  88.959  on 96  degrees of freedom
## AIC: 282.09
##
## Number of Fisher Scoring iterations: 2
```

```
summary(mod_4)
```

```
##
## Call:
## glm(formula = Y ~ poly(x, 4), data = frame)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0550  -0.6212  -0.1567   0.5952   2.2267
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.55002    0.09591 -16.162  < 2e-16 ***
## poly(x, 4)1   6.18883    0.95905   6.453 4.59e-09 ***
## poly(x, 4)2 -23.94830    0.95905 -24.971  < 2e-16 ***
## poly(x, 4)3   0.26411    0.95905   0.275    0.784
## poly(x, 4)4   1.25710    0.95905   1.311    0.193
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9197797)
##
##     Null deviance: 700.852  on 99  degrees of freedom
## Residual deviance:  87.379  on 95  degrees of freedom
## AIC: 282.3
##
## Number of Fisher Scoring iterations: 2
```

We see that the quadratic model has the lowest p-value and that the 3rd and 4th degree models are not significant at the 0.05 significant level.