Assignment24

John Bute

2024-11-07

R Assignment 25

1. Generate a simulated data set as follows

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

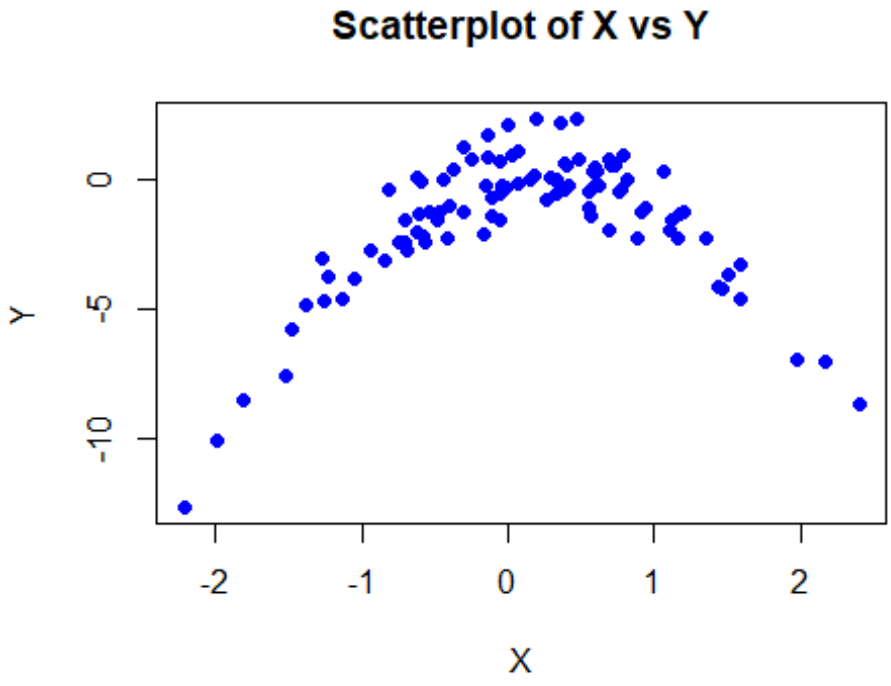
What is n and what is p? Write out the model used to generate the data in equation form.

In this dataset, n = 100 (number of observations) and p = 1 (Number of predictors, or X)

Thus, Y = X - 2X ^2 + epsilon, where epsilon represents the random term error.

2. Create a scatterplot of X against Y . Comment on what you find.

```
plot(x, y, main = "Scatterplot of X vs Y", xlab = "X", ylab = "Y", pch = 19,
col = "blue")
```



From what we see, we notice that there is a parabolic trend between X and Y, which is expected as we have a quadratic term in our equation (-2X^2) which introduces curviness in the relationship.

3. Set a random seed, and then compute the LOOCV errors that result from fitting the following four models using least squares: •i. $Y = \beta0 + \beta1X + \varepsilon$. •ii. $Y = \beta0 + \beta1X1 + \beta2X2 + \varepsilon$. •iii. $Y = \beta0 + \beta1X1 + \beta2X2 + \beta3X3 + \varepsilon$. •iv. $Y = \beta0 + \beta1X1 + \beta2X2 + \beta3X3 + \beta4X4 + \varepsilon$. Note you may find it helpful to use the data.frame() function to create a single data set containing both X and Y .

```
library(boot)

data <- data.frame(x = x, y = y)

compute_loocv <- function(model) {
  cv.glm(data, model)$delta[1]
}
set.seed(2)
model_1 <- glm(y ~ x, data = data)
loocv_1 <- compute_loocv(model_1)
model_2 <- glm(y ~ poly(x, 2), data = data)
loocv_2 <- compute_loocv(model_2)
model_3 <- glm(y ~ poly(x, 3), data = data)
loocv_3 <- compute_loocv(model_3)
model_4 <- glm(y ~ poly(x, 4), data = data)
loocv_4 <- compute_loocv(model_4)

loocv_results <- data.frame(Model = c("Linear", "Quadratic", "Cubic",
"Quartic"),
                            LOOCV_Error = c(loocv_1, loocv_2, loocv_3,
loocv_4))
print(loocv_results)

##       Model LOOCV_Error
## 1    Linear   7.2881616
## 2 Quadratic   0.9374236
## 3     Cubic   0.9566218
## 4   Quartic   0.9539049
```

4. Repeat 3. using another random seed, and report your results. Are your results the same as what you got in 3.? Why

```
set.seed(3)

loocv_1_new <- compute_loocv(model_1)
loocv_2_new <- compute_loocv(model_2)
loocv_3_new <- compute_loocv(model_3)
loocv_4_new <- compute_loocv(model_4)

loocv_results_new <- data.frame(Model = c("Linear", "Quadratic", "Cubic",
"Quartic"),
```

```
                                  LOOCV_Error = c(loocv_1_new, loocv_2_new,
loocv_3_new, loocv_4_new))
print(loocv_results_new)

##       Model LOOCV_Error
## 1    Linear   7.2881616
## 2 Quadratic   0.9374236
## 3     Cubic   0.9566218
## 4   Quartic   0.9539049
```

The results are the same, as LOOCV is deterministic for a given dataset and model, thus the results will not change. Since, LOOCV splits the datasets at leaves out one observation each time, there are no random elements in the process.

5. Which of the models in 3. had the smallest LOOCV error? Is this what you expected? Explain your answer. The quadratic has the smallest LOOCV, which is what I was expecting since the data showed a parabolic form and the equation that generated the data has a quadaratic term as its highest order. Therefore, quadratic models should provide the best fit for the data.

6. Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in 3. using OLS. Do these results agree with the conclusions drawn based on the cross-validation results?

```
summary(model_1)

##
## Call:
## glm(formula = y ~ x, data = data)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.6254     0.2619  -6.205 1.31e-08 ***
## x             0.6925     0.2909   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(model_2)

##
## Call:
## glm(formula = y ~ poly(x, 2), data = data)
##
```

```
## 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(model_3)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 3), data = data)
##
## 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(model_4)
```

```
##
## Call:
## glm(formula = y ~ poly(x, 4), data = data)
##
## 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
```

For the first Model, only the intercept and X are statistcally significant, but the high LOOCV error suggests that the model does not perform well.

For the quadaratic model, all terms are significant as their p-values are < 0.001. Furthermore, this model has the lowest LOOCV out of all the models, which suggests that this is the best model for capturing the relationship between X and Y by including a quadartic term.

For the cubic model, the X, X^2 and intercept are highly significant values, but the higher term X^3 is not significant, with a p-value of 0.784. Thus, the summary of this model suggests that adding the X^3 term was unnecessary and hurts the LOOCV result

Finally, for the last model, the X, X^2 and intercept are highly significant values, but the higher terms X^3 and X^4 are not siginificant with 0.784 nad 0.193 p-values respectiviely, thus indicating that these terms do not help our model, and lead to a higher LOOCV.

In all, these results correspond well to the CV results we obtained.