Here is the output of the R code:

|  |
| --- |
| > #HW12 Assignment and program to help you  >  > library(faraway)  > #install.packages("caret") #install this package if needed  > library(caret)  Loading required package: lattice  Attaching package: ‘lattice’  The following object is masked from ‘package:faraway’:  melanoma  Loading required package: ggplot2  > set.seed(13245) #use this seed  >  > head(cars,1L)  speed dist  1 4 2  >  > attach(cars) #n=50  > # sorting dataset by distance for graphing purposes  > cars <- cars[order(dist),]  > cars  speed dist  1 4 2  3 7 4  2 4 10  6 9 10  12 12 14  5 8 16  10 11 17  7 10 18  13 12 20  24 15 20  4 7 22  14 12 24  8 10 26  16 13 26  20 14 26  25 15 26  11 11 28  15 12 28  27 16 32  29 17 32  39 20 32  9 10 34  17 13 34  18 13 34  21 14 36  36 19 36  28 16 40  30 17 40  32 18 42  19 13 46  37 19 46  40 20 48  31 17 50  41 20 52  26 15 54  45 23 54  33 18 56  42 20 56  22 14 60  43 20 64  44 22 66  38 19 68  46 24 70  34 18 76  23 14 80  35 18 84  50 25 85  47 24 92  48 24 93  49 24 120  >  > windows(7,7)  > plot(x=cars$dist,y=cars$speed)  >  > ##-------------------------------------------------------------##  >  > #The researcher is interested in predicting speed based on knowing stopping distance  > #fit a polynomial to the data - use degree 1, 2, 3, or 4?  > #use cross-validation since overfitting is a concern  > #ASSIGNMENT: use 5-fold cross validation to obtain the choice of degree 1, 2, 3, or 4  >  >  > #Here is the r-code for a polynomial of degree 4 and plotting the fitted curve  > #You can use the code below and just repeat for a polynomials of degree 1, 2, and 3  >  > #Fit a polynomial of degree 4  > poly4<- lm(speed~dist+I(dist^2)+I(dist^3)+I(dist^4), data=cars)  > summary(poly4) #summary of results from fitting a polynomial of degree 4  Call:  lm(formula = speed ~ dist + I(dist^2) + I(dist^3) + I(dist^4),  data = cars)  Residuals:  Min 1Q Median 3Q Max  -6.8557 -1.9194 0.2788 2.0023 5.5300  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 3.430e+00 2.364e+00 1.451 0.154  dist 4.806e-01 2.521e-01 1.906 0.063 .  I(dist^2) -4.909e-03 8.633e-03 -0.569 0.572  I(dist^3) 2.151e-05 1.109e-04 0.194 0.847  I(dist^4) -1.494e-08 4.657e-07 -0.032 0.975  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 2.917 on 45 degrees of freedom  Multiple R-squared: 0.7206, Adjusted R-squared: 0.6957  F-statistic: 29.01 on 4 and 45 DF, p-value: 5.984e-12  > plot(x=cars$dist,y=cars$speed)  > lines(x=cars$dist,y=poly4$fitted, type="l", col="red")  >  > #Compute the cross-validation metrics for degree 4  > # Define training control  > train.control <- trainControl(method = "cv", number = 5)  > # Train the model  > CVpoly4 <- train(speed~dist+I(dist^2)+I(dist^3)+I(dist^4),data = cars, method = "lm",  + trControl = train.control)  > # Summarize the results  > print(CVpoly4)  Linear Regression  50 samples  1 predictor  No pre-processing  Resampling: Cross-Validated (5 fold)  Summary of sample sizes: 41, 40, 39, 40, 40  Resampling results:  RMSE Rsquared MAE  4.399993 0.6739873 3.032101  Tuning parameter 'intercept' was held constant at a value of TRUE  > ## |
|  |
| |  | | --- | | > | |

Looking at the results, since degree 4 has the least standard error, I will choose degree 4