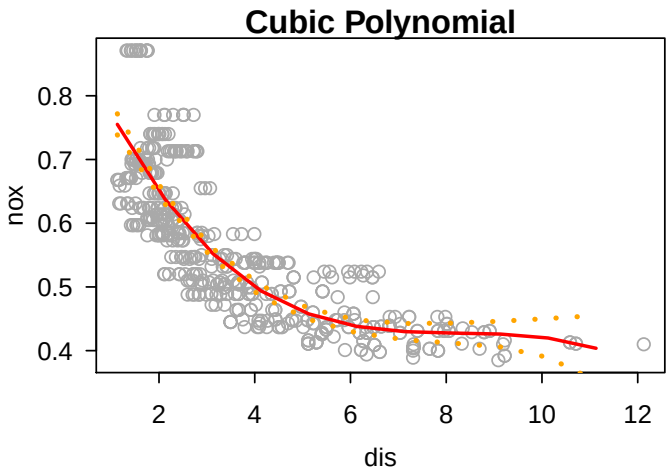
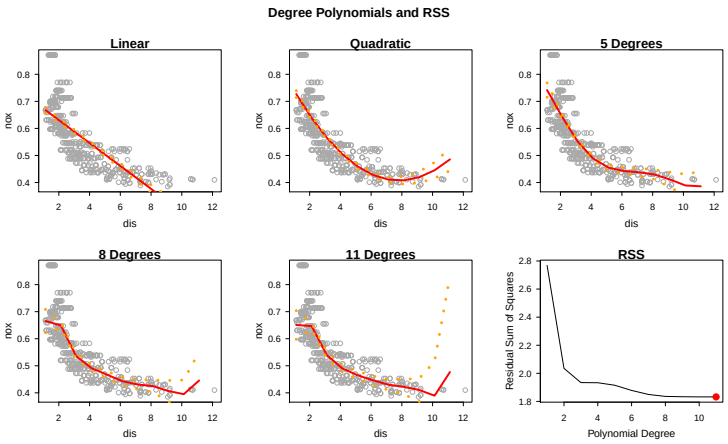
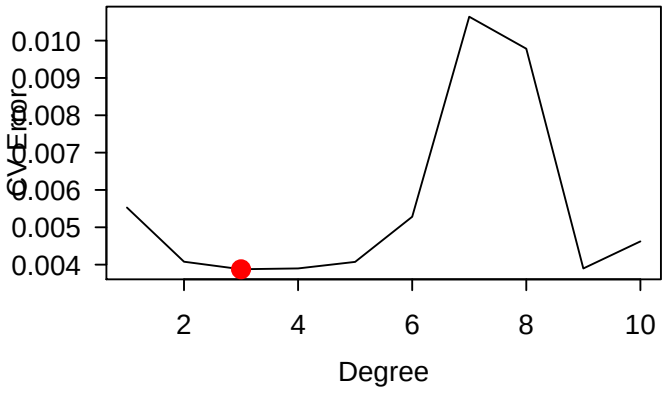
The results below are generated from an R script.

```
cubic_fit <- lm(nox ~ poly(dis, 3), data = Boston)
coef(summary(cubic_fit))

##                Estimate Std. Error t value   Pr(>|t|)
## (Intercept)     0.5547   0.002759 201.021  0.000e+00
## poly(dis, 3)1  -2.0031   0.062071 -32.271 1.597e-124
## poly(dis, 3)2   0.8563   0.062071  13.796  6.133e-37
## poly(dis, 3)3  -0.3180   0.062071  -5.124  4.275e-07

dislims <- range(dis)
dis_grid <- seq(from = dislims[1], to = dislims[2])
cubic_pred <- predict(cubic_fit, newdata = list(dis = dis_grid), se = TRUE)
se_bands <- cbind(cubic_pred$fit + 2*cubic_pred$se.fit,
                  cubic_pred$fit - 2*cubic_pred$se.fit)
par(mar = c(4.5,4.5,1,1), oma = c(0,0,2,0))
plot(dis, nox, xlim = dislims, col = "darkgrey", xlab = "dis", ylab = "nox")
title("Cubic Polynomial", outer = FALSE) # title that spans both plots
lines(dis_grid, cubic_pred$fit, lwd = 2, col = "red")
matlines(dis_grid, se_bands, lwd = 3, col = "orange", lty = 3)
```

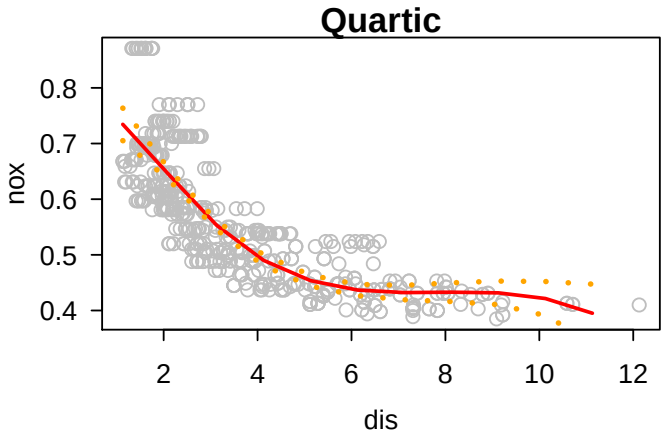




```r
prediction_error <- rep(0, 10)
for (i in 1:10){ # Run all the polynomial models and store them
  # Use the glm function for poly models instead of lm so we can use cv.glm
   poly_fit <- glm(nox ~ poly(dis, i), data = Boston)
   prediction_error[i] <- cv.glm(Boston, poly_fit, K = 10)$delta[1]}
par(mfrow = c(1,1),  mar = c(4.5,4.5,1,1), oma = c(0,0,2,0)) # plot it!
plot(1:10, prediction_error, xlab = "Degree", ylab = "CV Error", type = "l")
d.min <- which.min(prediction_error)
points(which.min(prediction_error), prediction_error[which.min(prediction_error)],
       col = "red", cex = 2, pch = 20)
```

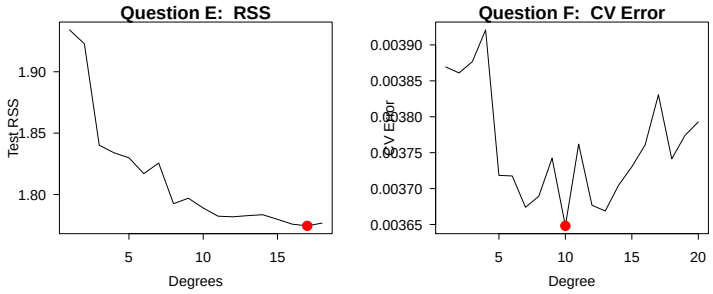



```r
spline_fit <- lm(nox ~ bs(dis, df = 4), data = Wage)
spline_pred <- predict(spline_fit, newdata = list(dis = dis_grid), se = TRUE)
par(mar = c(4.5,4.5,1,1), oma = c(0,0,2,0))
plot(dis, nox, col = "gray");title("Quartic", outer = FALSE)  # Plot the output
lines(dis_grid, spline_pred$fit, lwd = 2, col = "red")
lines(dis_grid, spline_pred$fit + 2* spline_pred$se, lwd = 3, col = "orange", lty = 3)
lines(dis_grid, spline_pred$fit - 2* spline_pred$se,lwd = 3, col = "orange", lty = 3)
```



```
attr(bs(dis, df = 4), "knots")

##   50%
## 3.207
```

```
# Code for *Question E*
RSS_reg_splines <- rep(NA, 18)
for (i in 3:20) {
    reg_spline_fit <- lm(nox ~ bs(dis, df = i), data = Boston)
    RSS_reg_splines[i] <- sum(reg_spline_fit$residuals^2)}
par(mfrow = c(1,2), mar = c(4.5,4.5,1,1), oma = c(0,0,4,0))
RSS <- RSS_reg_splines[-c(1, 2)]
plot(1:18, RSS, xlab = "Degrees", ylab = "Test RSS", type = "l")
d.min <- which.min(RSS); title("Question E:  RSS", outer = FALSE)
points(which.min(RSS), RSS[which.min(RSS)], col = "red", cex = 2, pch = 20)

# Code for *Question F*
prediction_error <- rep(0, 20); set.seed(232)
for (i in 1:20){ # Run all the polynomial models and store them
  # Use the glm function for poly models instead of lm so we can use cv.glm
   reg_spline_fit <- glm(nox ~ bs(dis, df = i), data = Boston)
   prediction_error[i] <- cv.glm(Boston, reg_spline_fit, K = 10)$delta[1]}
plot(1:20, prediction_error, xlab = "Degree", ylab = "CV Error", type = "l")
d.min <- which.min(prediction_error); title("Question F:  CV Error", outer = FALSE)
points(which.min(prediction_error), prediction_error[which.min(prediction_error)],
       col = "red", cex = 2, pch = 20)
```



The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 3.2.4 (2016-03-10)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.9.5 (Mavericks)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods   base
##
```

```
## other attached packages:
## [1] shiny_0.12.2 boot_1.3-18  MASS_7.3-45  ISLR_1.0     knitr_1.12.3
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.4       digest_0.6.9      mime_0.4          R6_2.1.1
##  [5] jsonlite_0.9.17   xtable_1.7-4      formatR_1.3       magrittr_1.5
##  [9] evaluate_0.8.3    highr_0.5.1       stringi_1.0-1     rstudioapi_0.3.1
## [13] tools_3.2.4       stringr_1.0.0     markdown_0.7.7    httpuv_1.3.3
## [17] rsconnect_0.4.1.4 htmltools_0.3.5

Sys.time()

## [1] "2016-05-01 10:02:35 EDT"
```