[STAT 4610] HW-8

Michael Ghattas

10/24/2022

Chapter 7

```
library(MASS)

## Warning: package 'MASS' was built under R version 4.1.2

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.2

## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.5
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2

## Warning: package 'ggplot2' was built under R version 4.1.2

## Warning: package 'tibble' was built under R version 4.1.2

## Warning: package 'tidyr' was built under R version 4.1.2

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'purrr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.2

## Warning: package 'stringr' was built under R version 4.1.2

## Warning: package 'forcats' was built under R version 4.1.2

## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ dplyr::select() masks MASS::select()
```

```
library(ggplot2)
library(ggthemes)
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.1.2
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.1.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(splines)
```

Problem-9

```
set.seed(123)
theme_set(theme_tufte(base_size = 14) + theme(legend.position = 'top'))
data('Boston')
```

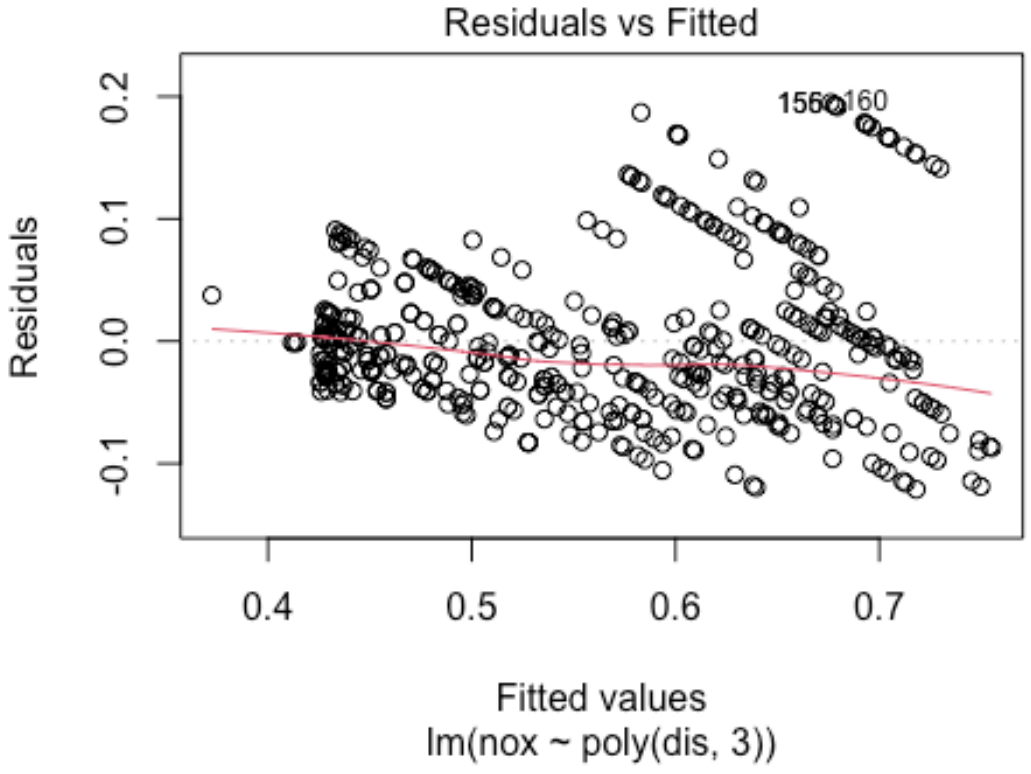
Part(a)

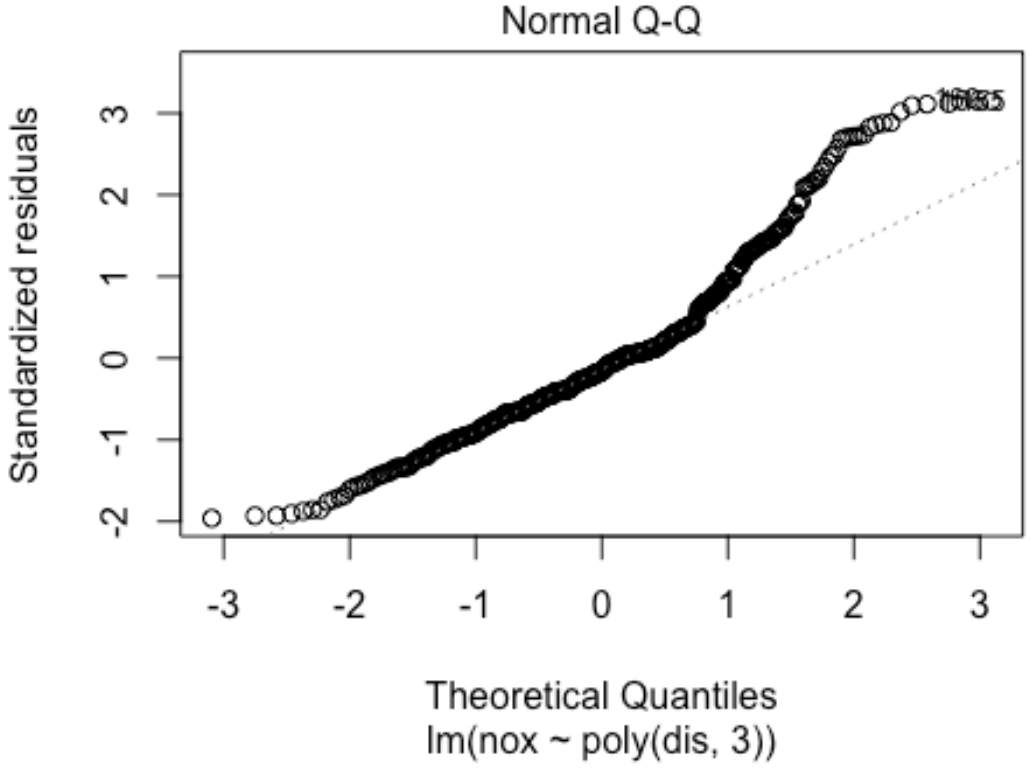
```
lMod <- lm(nox ~ poly(dis, 3), data = Boston)
summary(lMod)
```

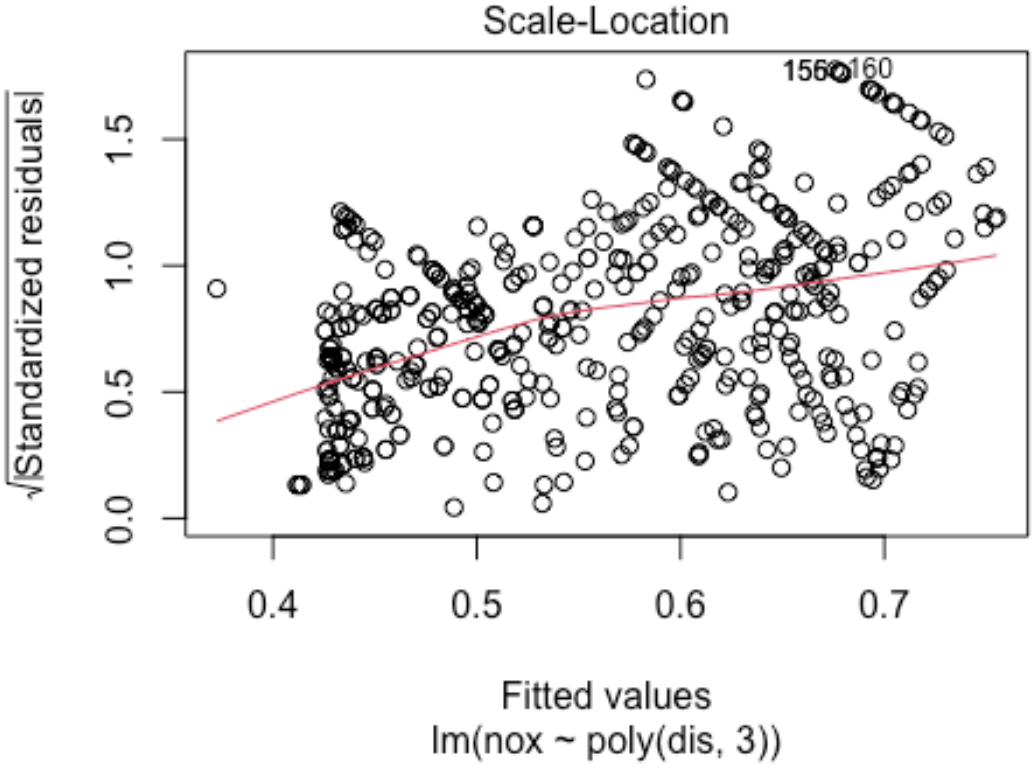
```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
```

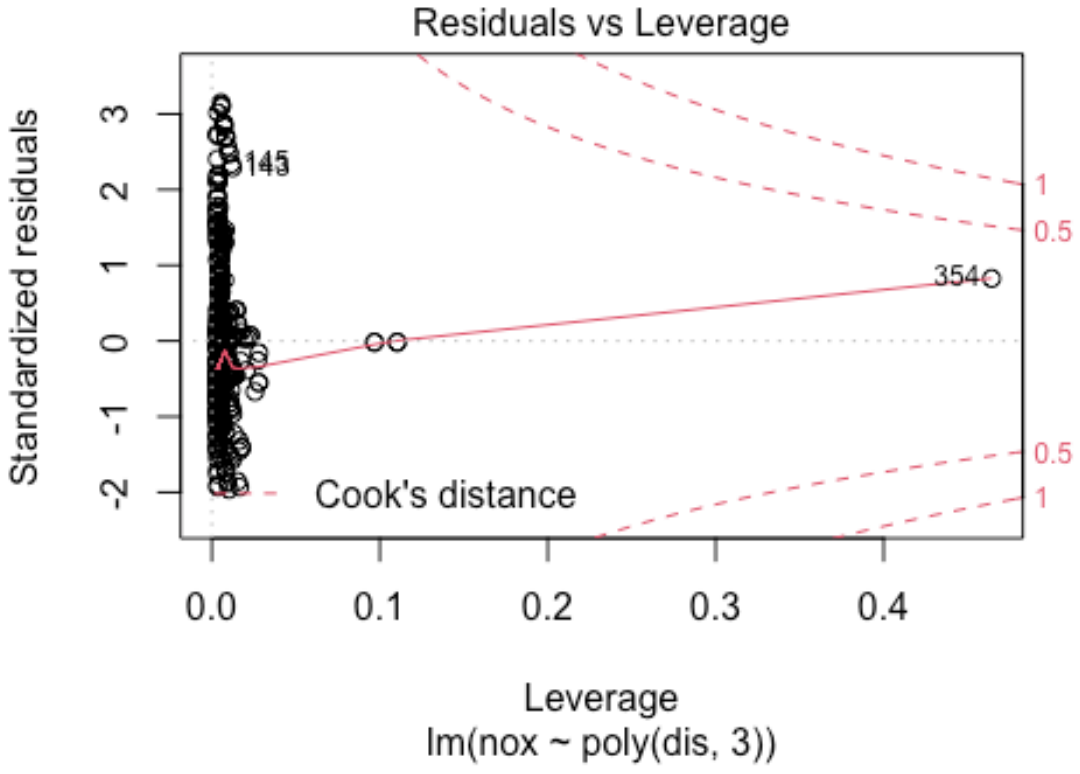
```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
## poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
## poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
## poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06207 on 502 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.7131
## F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
plot(lMod)
```

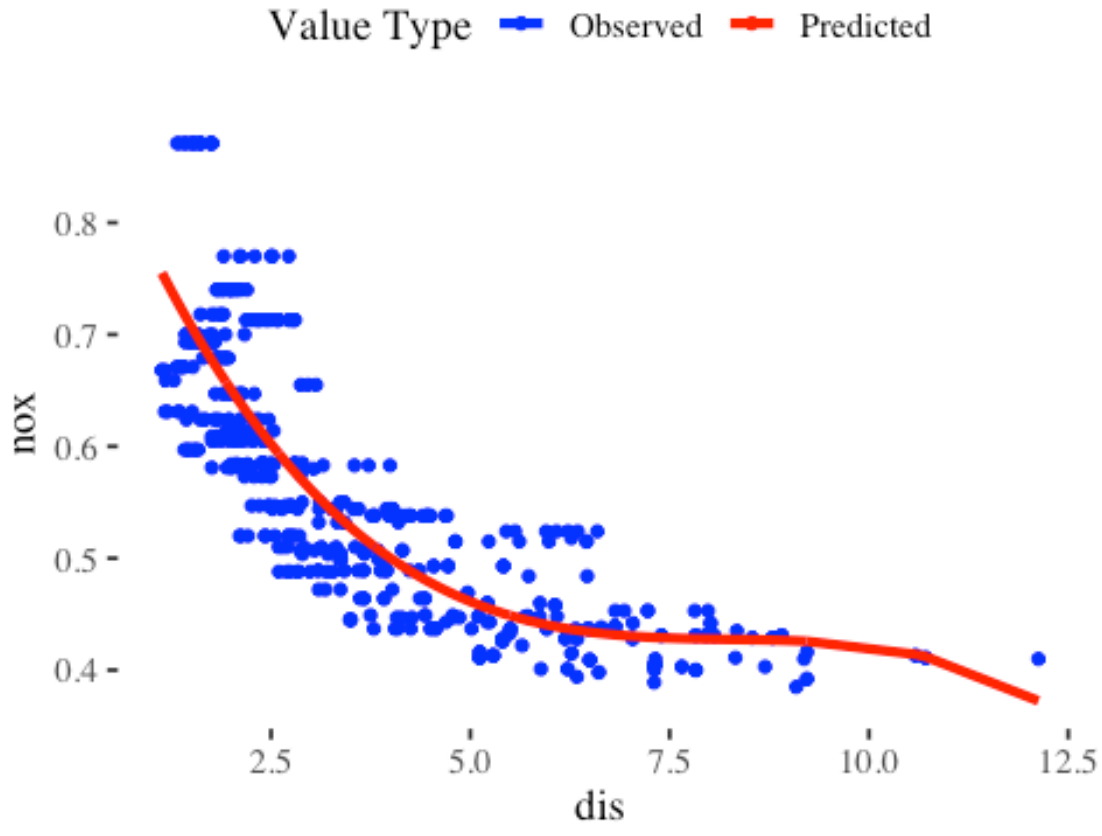

Residuals vs Fitted
Residuals
0.2
0.1
0.0
-0.1
156
160
0.4
0.5
0.6
0.7
Fitted values
lm(nox ~ poly(dis, 3))


Normal Q-Q
Standardized residuals
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(nox ~ poly(dis, 3))


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0.4
0.5
0.6
0.7
Fitted values
lm(nox ~ poly(dis, 3))




```
Boston %>% mutate(pred = predict(lMod, Boston)) %>% ggplot() +
geom_point(aes(dis, nox, col = '1')) + geom_line(aes(dis, pred, col = '2'),
size = 1.5) +
  scale_color_manual(name = 'Value Type', labels = c('Observed',
'Predicted'), values = c('blue', 'red'))
```

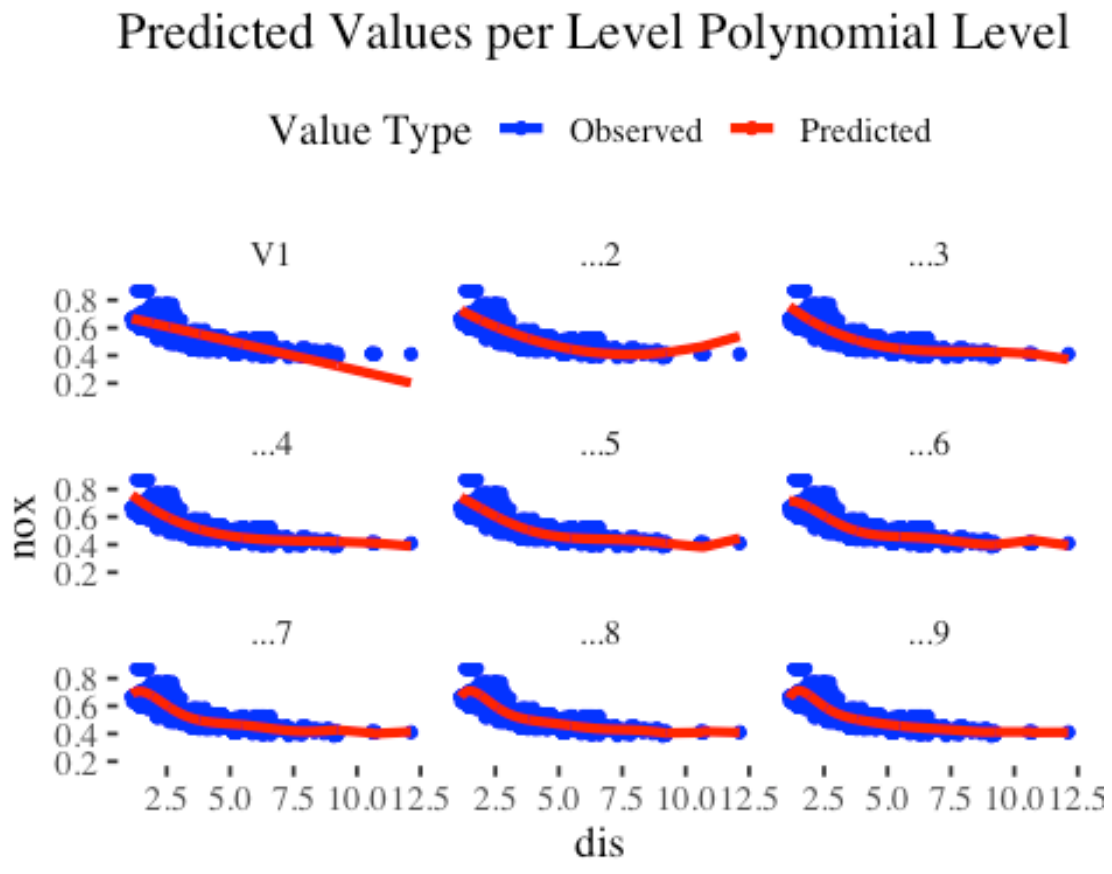


-> We can note from the summary that each power of the “dis” coefficient seems to be statistically significant. -> The fitted line seems to describe the data well.

Part(b)

```
errs <- list()
lMods <- list()

pred_df <- data_frame(V1 = 1:506)

## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## ℹ Please use `tibble()` instead.

for (i in 1:9) {
    lMods[[i]] <- lm(nox ~ poly(dis, i), data = Boston)
    preds <- predict(lMods[[i]])
    pred_df[[i]] <- preds
```

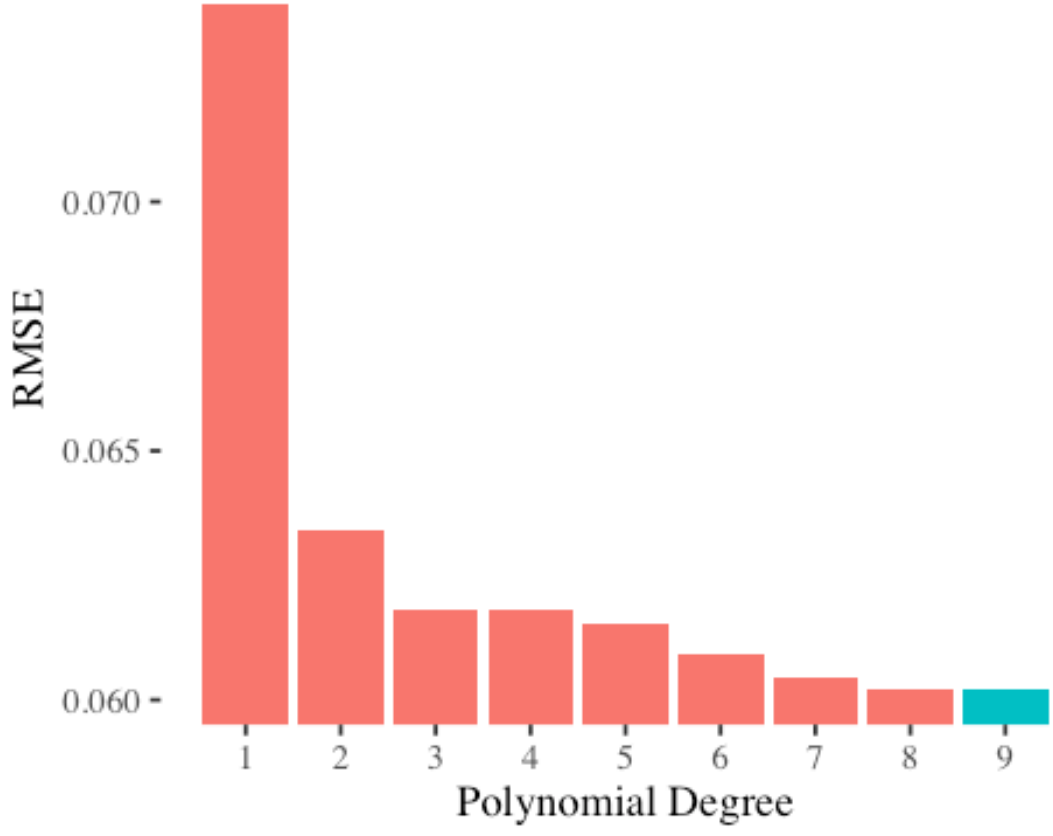
```
    errs[[i]] <- sqrt(mean((Boston$nox - preds)^2))
}

Boston %>% cbind(pred_df) %>% gather(Polynomial, prediction, -(1:14)) %>%
mutate(Polynomial = factor(Polynomial, levels =
unique(as.character(Polynomial)))) %>% ggplot() +
  ggtitle('Predicted Values per Level Polynomial Level') +
geom_point(aes(dis, nox, col = '1')) +  geom_line(aes(dis, prediction, col =
'2'), size = 1.5) +
  scale_color_manual(name = 'Value Type', labels = c('Observed',
'Predicted'), values = c('blue', 'red')) + facet_wrap(~ Polynomial, nrow = 3)
```




```
errs <- unlist(errs)

names(pred_df) <- paste('Level', 1:9)
```

```
data_frame(RMSE = errs) %>% mutate(Poly = row_number()) %>% ggplot(aes(Poly,
RMSE, fill = Poly == which.min(errs))) + geom_col() +  guides(fill = FALSE) +
scale_x_continuous(breaks = 1:9) +
    coord_cartesian(ylim = c(min(errs), max(errs))) + labs(x = 'Polynomial
Degree')
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



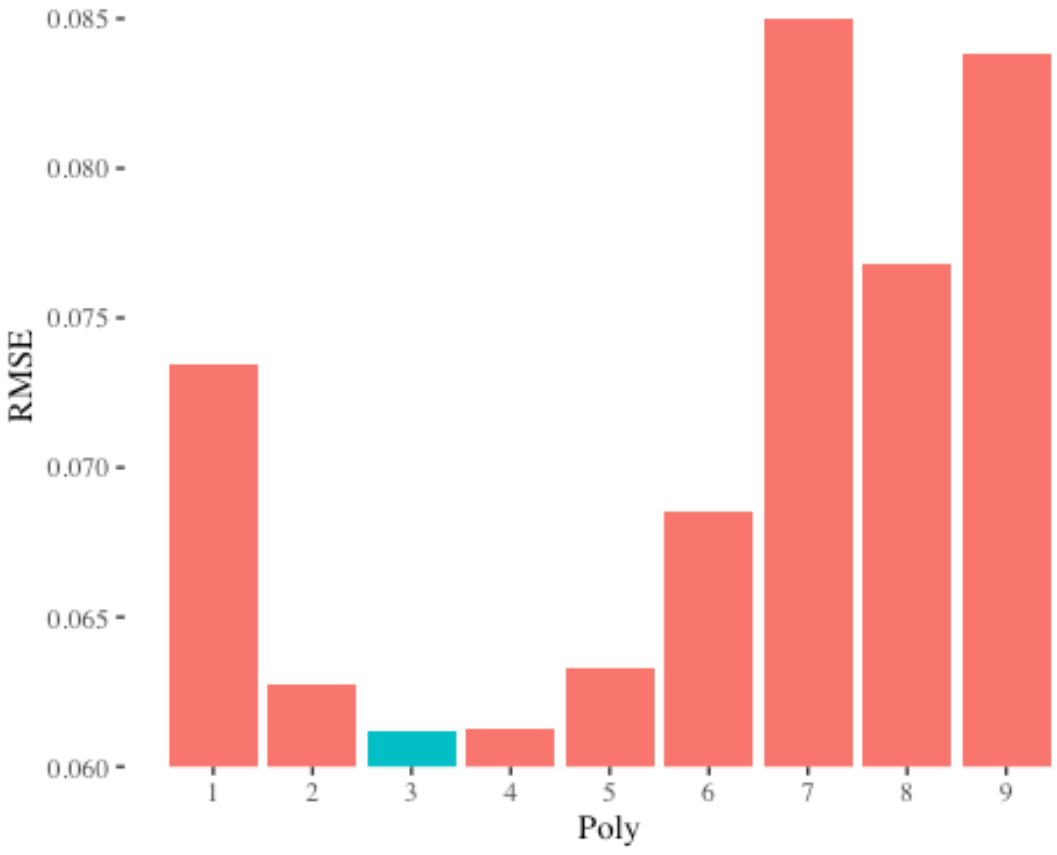
-> The model with the highest polynomial degree has the lowest RSS.

Part(c)

```
errs <- list()
folds <- sample(1:10, 506, replace = TRUE)
errs <- matrix(NA, 10, 9)
```

```r
for (k in 1:10) {
    for (i in 1:9) {
        model <- lm(nox ~ poly(dis, i), data = Boston[folds != k, ])
        pred <- predict(model, Boston[folds == k, ])
        errs[k, i] <- sqrt(mean((Boston$nox[folds == k] - pred)^2))
    }
}

errs <- apply(errs, 2, mean)

data_frame(RMSE = errs) %>% mutate(Poly = row_number()) %>% ggplot(aes(Poly,
RMSE, fill = Poly == which.min(errs))) + geom_col() + theme_tufte() +
guides(fill = FALSE) + scale_x_continuous(breaks = 1:9) +
    coord_cartesian(ylim = range(errs))
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



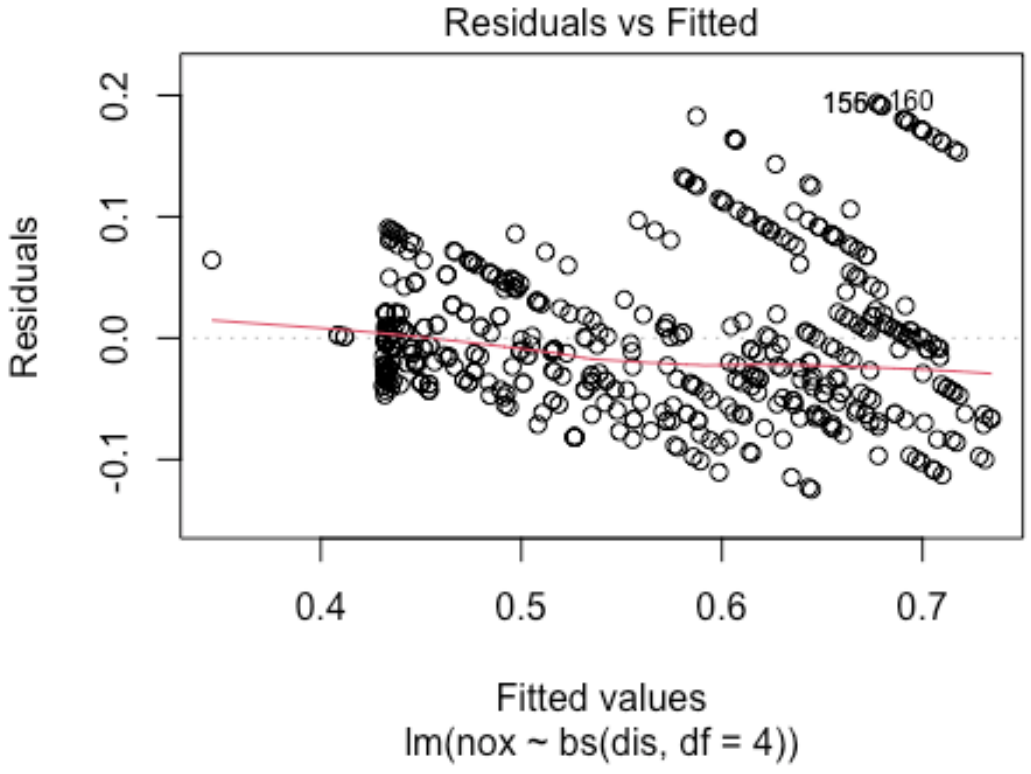
-> The model with polynomial degree 3 is the highest degree that has the lowest RMSE and thus does not show signs of over-fitting.

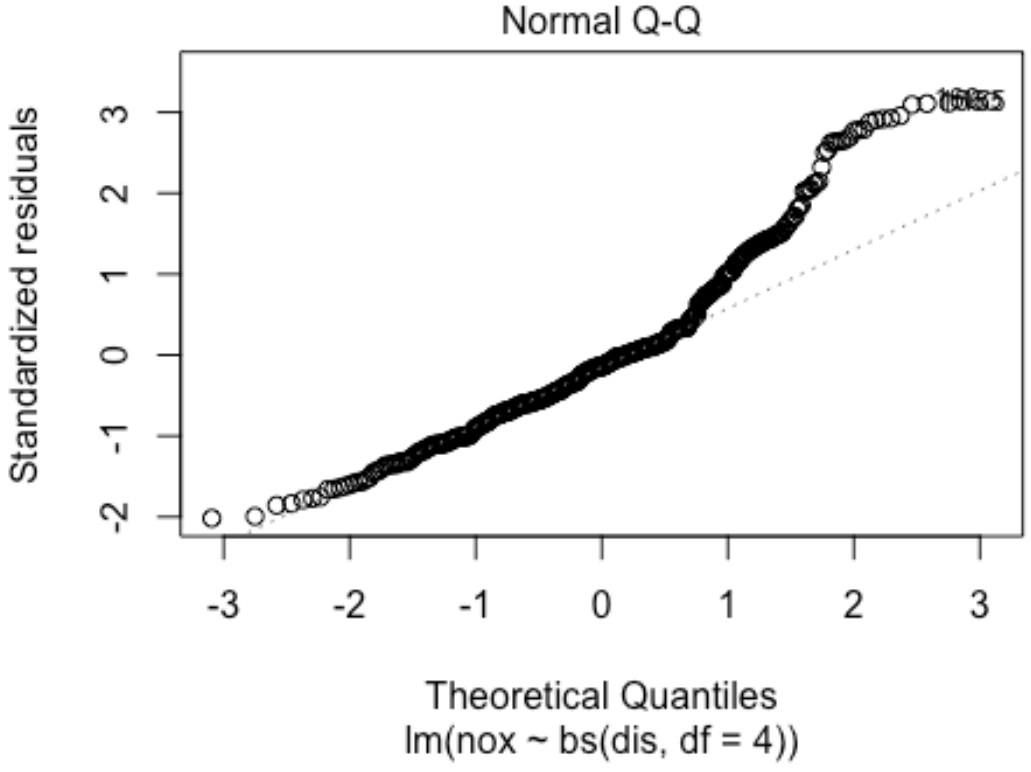
Part(d)

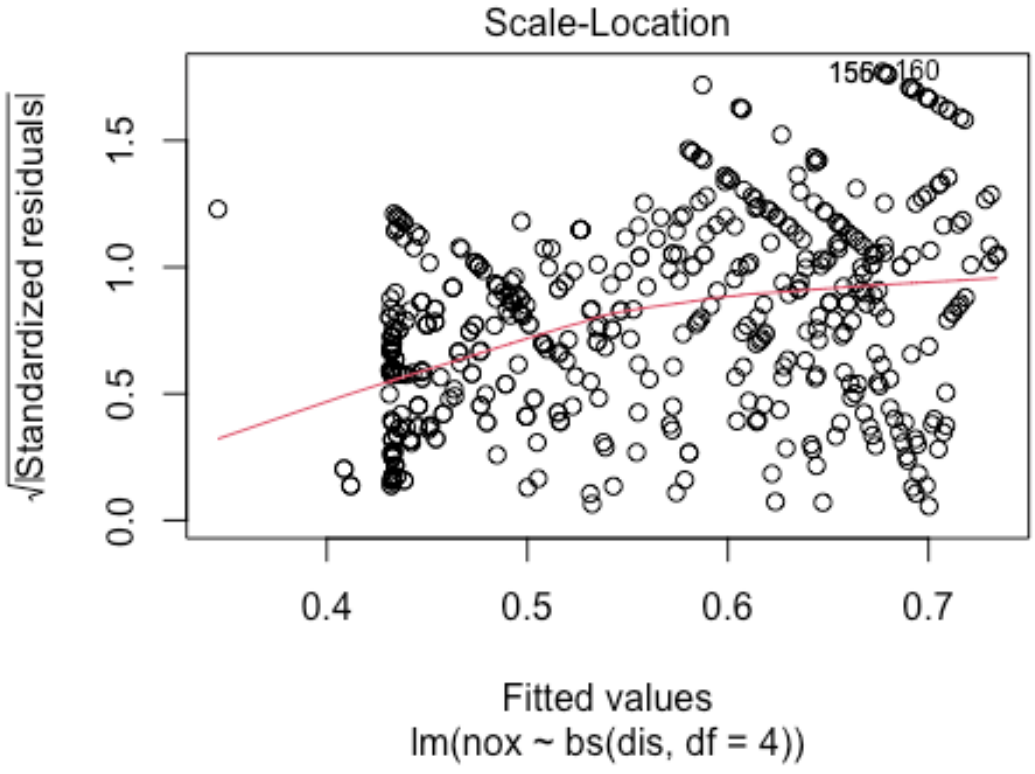
```
lMod <- lm(nox ~ bs(dis, df = 4), data = Boston)
summary(lMod)

##
## Call:
## lm(formula = nox ~ bs(dis, df = 4), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124622 -0.039259 -0.008514  0.020850  0.193891
##
```

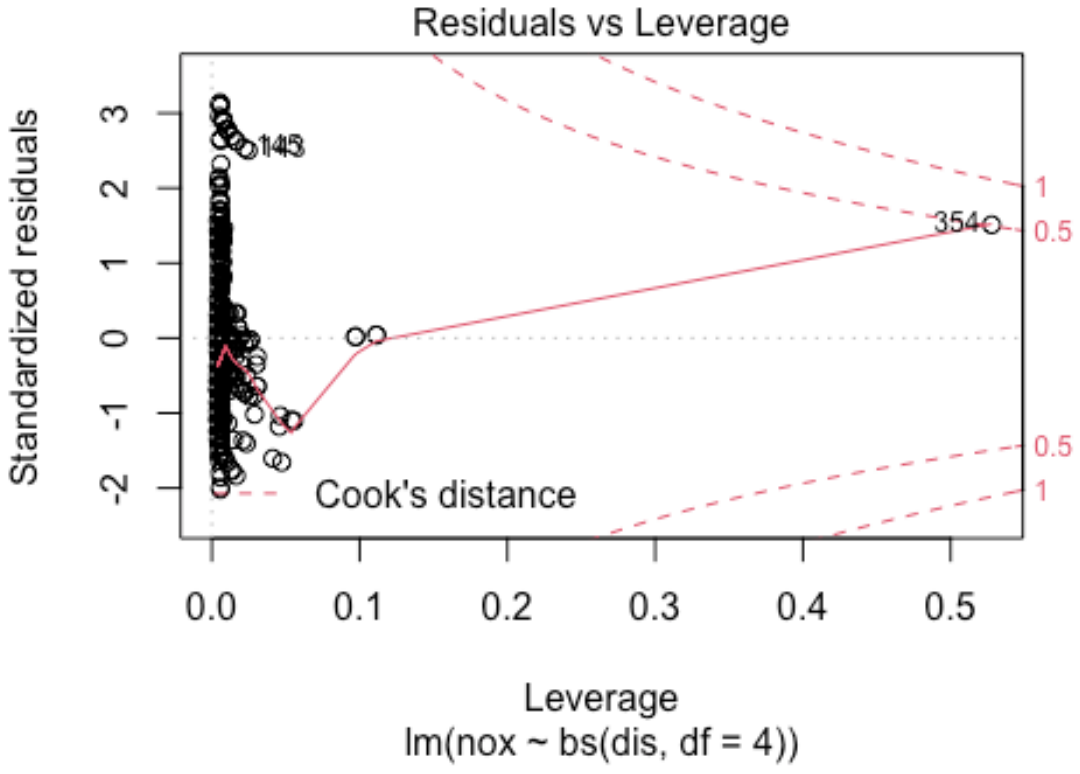
```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.73447    0.01460  50.306  < 2e-16 ***
## bs(dis, df = 4)1 -0.05810    0.02186  -2.658  0.00812 **
## bs(dis, df = 4)2 -0.46356    0.02366 -19.596  < 2e-16 ***
## bs(dis, df = 4)3 -0.19979    0.04311  -4.634 4.58e-06 ***
## bs(dis, df = 4)4 -0.38881    0.04551  -8.544  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06195 on 501 degrees of freedom
## Multiple R-squared:  0.7164, Adjusted R-squared:  0.7142
## F-statistic: 316.5 on 4 and 501 DF,  p-value: < 2.2e-16
```

```
plot(lMod)
```

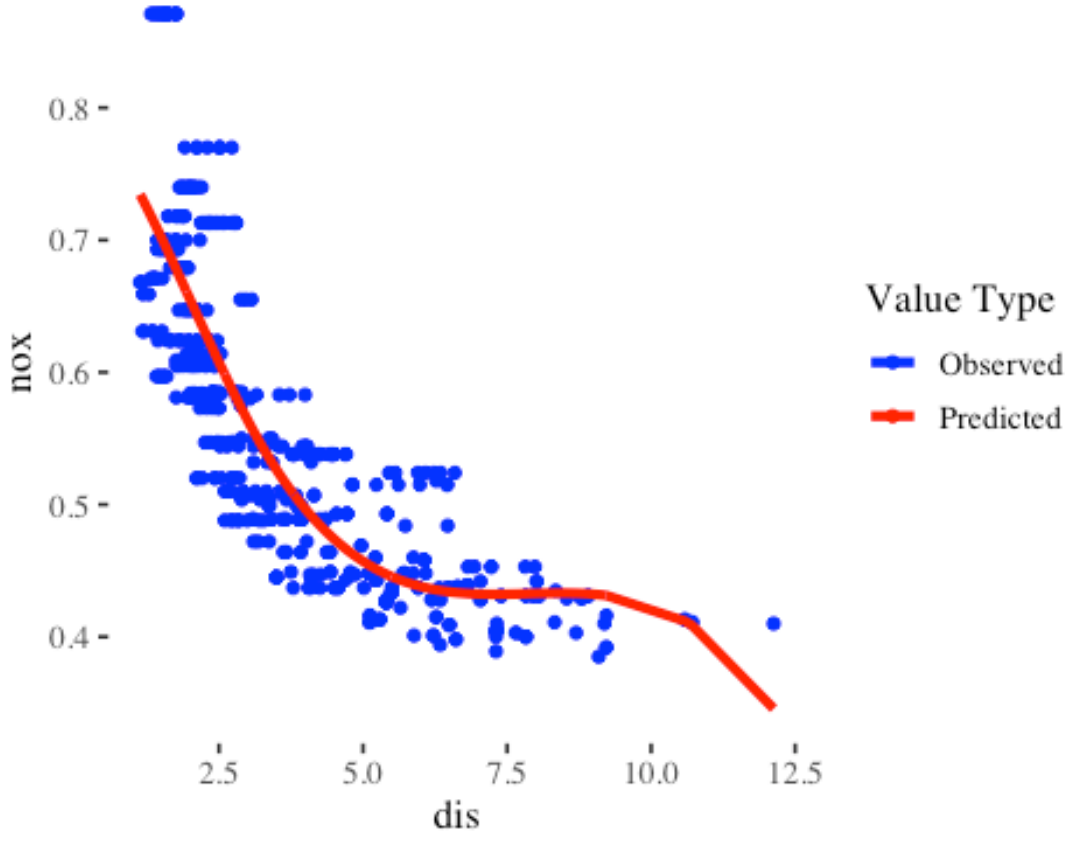

Residuals vs Fitted
Residuals
0.2
0.1
0.0
-0.1
156
160
0.4
0.5
0.6
0.7
Fitted values
lm(nox ~ bs(dis, df = 4))


Normal Q-Q
Standardized residuals
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(nox ~ bs(dis, df = 4))


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0.4
0.5
0.6
0.7
Fitted values
lm(nox ~ bs(dis, df = 4))




```
Boston %>% mutate(pred = predict(lMod)) %>% ggplot() + geom_point(aes(dis,
nox, col = '1')) + geom_line(aes(dis, pred, col = '2'), size = 1.5) +
  scale_color_manual(name = 'Value Type', labels = c('Observed',
'Predicted'), values = c('blue', 'red')) + theme_tufte(base_size = 13)
```



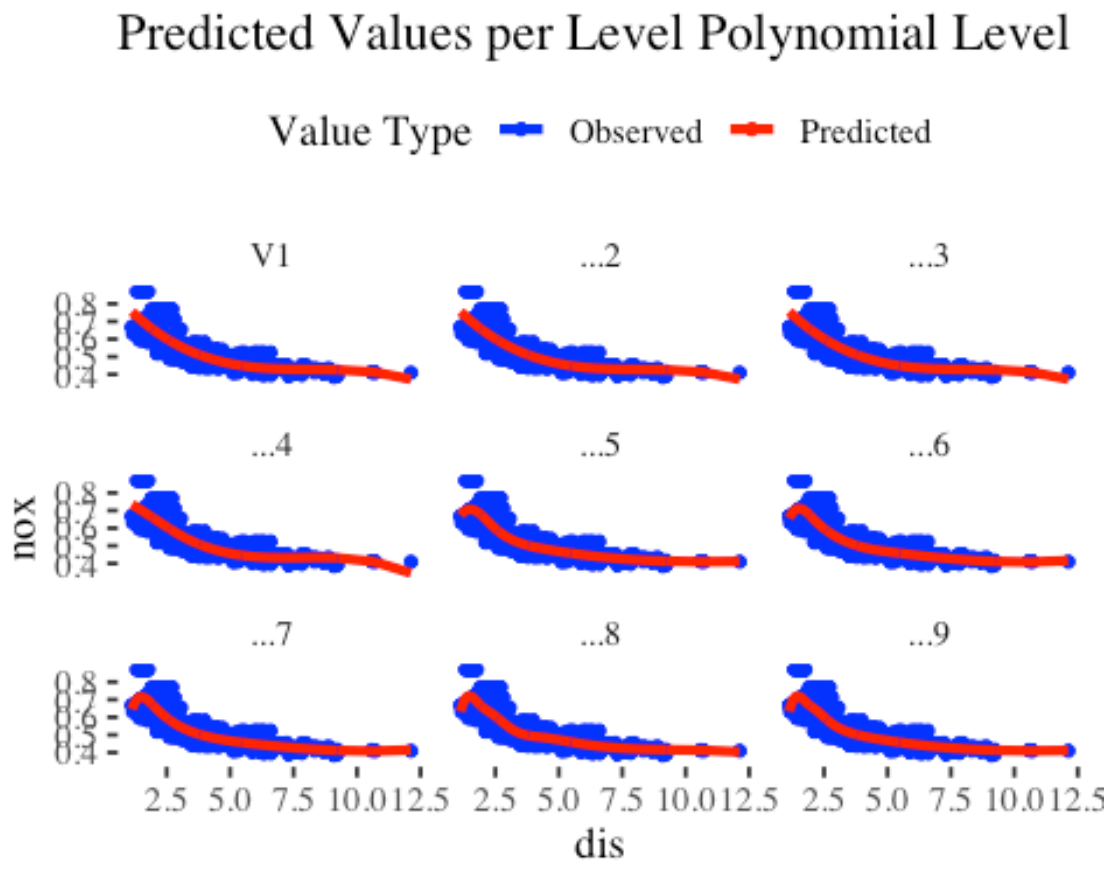
-> All the bases seem to be statistically significant for the model. -> The prediction line seems to fit the data.

Part(e)

```
errs <- list()
lMods <- list()

pred_df <- data_frame(V1 = 1:506)
for (i in 1:9) {
    lMods[[i]] <- lm(nox ~ bs(dis, df = i), data = Boston)
    preds <- predict(lMods[[i]])
    pred_df[[i]] <- preds
    errs[[i]] <- sqrt(mean((Boston$nox - preds)^2))
}
```

```
## Warning in bs(dis, df = i): 'df' was too small; have used 3

## Warning in bs(dis, df = i): 'df' was too small; have used 3
```

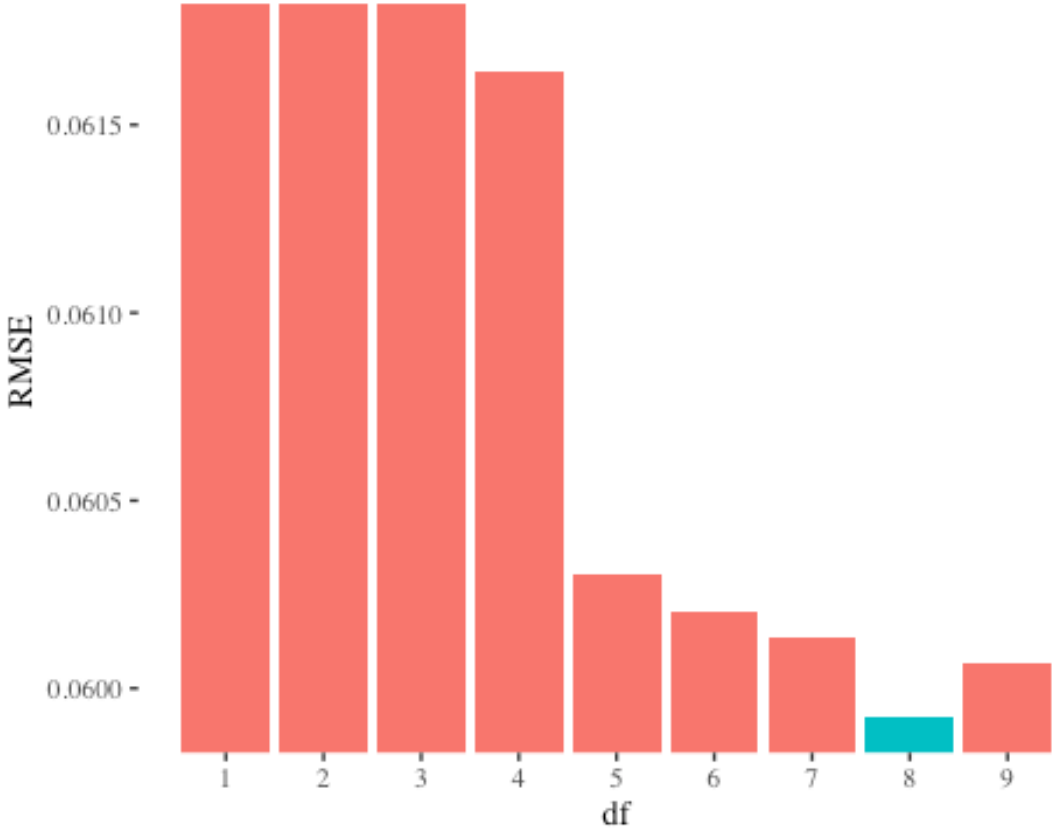
```
Boston %>% cbind(pred_df) %>% gather(df, prediction, -(1:14)) %>% mutate(df =
factor(df, levels = unique(as.character(df)))) %>% ggplot() +
ggtitle('Predicted Values per Level Polynomial Level') +
  geom_point(aes(dis, nox, col = '1')) + geom_line(aes(dis, prediction, col =
'2'), size = 1.5) +
  scale_color_manual(name = 'Value Type', labels = c('Observed',
'Predicted'), values = c('blue', 'red')) + facet_wrap(~ df, nrow = 3)
```




```
names(pred_df) <- paste(1:9, 'Degrees of Freedom')
data_frame(RMSE = unlist(errs)) %>% mutate(df = row_number()) %>%
ggplot(aes(df, RMSE, fill = df == which.min(errs))) + geom_col() +
```

```
guides(fill = FALSE) + theme_tufte() +
  scale_x_continuous(breaks = 1:9) + coord_cartesian(ylim = range(errs))
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



-> It seems that the model with high complexity is the best.

Part(f)

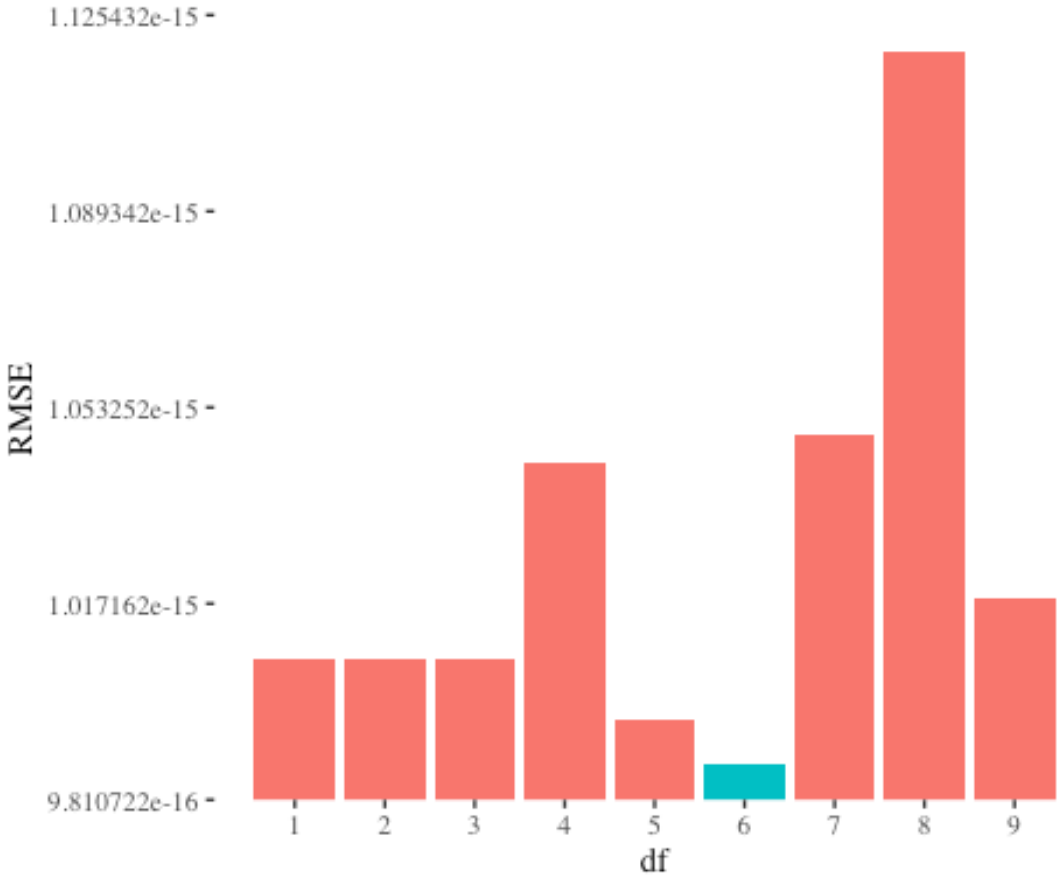
```
folds <- sample(1:10, size = 506, replace = TRUE)
errs <- matrix(NA, 10, 9)
lMods <- list()

for (k in 1:10) {
    for (i in 1:9) {
```

```
        lMods[[i]] <- lm(nox ~ bs(nox, df = i), data = Boston[folds != k, ])
        pred <- predict(lMods[[i]], Boston[folds == k, ])
        errs[k, i] <- sqrt(mean((Boston$nox[folds == k] - pred)^2))
    }
}
```

```
## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, degree = 3L, knots = numeric(0), Boundary.knots =
c(0.389, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(nox, df = i): 'df' was too small; have used 3
```

```
## Warning in bs(nox, degree = 3L, knots = numeric(0), Boundary.knots =
c(0.389, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(nox, degree = 3L, knots = numeric(0), Boundary.knots =
c(0.389, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(nox, degree = 3L, knots = c(`50%` = 0.538), Boundary.knots =
## c(0.389, : some 'x' values beyond boundary knots may cause ill-conditioned
bases

## Warning in bs(nox, degree = 3L, knots = c(`33.33333%` = 0.489, `66.66667%`
=
## 0.597: some 'x' values beyond boundary knots may cause ill-conditioned
bases

## Warning in bs(nox, degree = 3L, knots = c(`25%` = 0.448, `50%` = 0.538, :
some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(nox, degree = 3L, knots = c(`20%` = 0.4414, `40%` = 0.504, :
some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(nox, degree = 3L, knots = c(`16.66667%` = 0.437, `33.33333%`
=
## 0.489, : some 'x' values beyond boundary knots may cause ill-conditioned
bases

## Warning in bs(nox, degree = 3L, knots = c(`14.28571%` = 0.431, `28.57143%`
=
## 0.46, : some 'x' values beyond boundary knots may cause ill-conditioned
bases

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3
```

```
## Warning in bs(nox, df = i): 'df' was too small; have used 3

## Warning in bs(nox, df = i): 'df' was too small; have used 3

errs <- apply(errs, 2, mean)

data_frame(RMSE = errs) %>% mutate(df = row_number()) %>% ggplot(aes(df,
RMSE, fill = df == which.min(errs))) + geom_col() + theme_tufte() +
guides(fill = FALSE) + scale_x_continuous(breaks = 1:9) +
  coord_cartesian(ylim = range(errs))

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```



-> Once validated with out-of-sample data, we are able to choose a simpler model. -> As per the

polynomial validation process, here we can see that our choice is a complex model with the lowest RMSE, which does not show signs of over-fitting.

End.