Question 2

Set up:

```
library(readr)
library(tibble)
library(MASS)
data(Boston)
Boston = as_tibble(Boston)

set.seed(42)
boston_index = sample(1:nrow(Boston), size = 400)
train_boston = Boston[boston_index, ]
test_boston = Boston[-boston_index, ]
```

Fitting Model:

```
fit = lm(medv ~ . ^ 2, data = train_boston)
```

fitting 2 more models (one smaller and one bigger than fit):

```
fit_smaller <- lm(medv ~ . , data = train_boston)
fit_larger <- lm(medv ~ . ^ 2 + I(crim^2)+I(zn^2)+I(indus^2)+I(chas^2)+I(nox^2),data = train_boston)
```

Defining RMSE function and complexity function:

```
# Calculates RMSE

rmse = function (actual,predicted){
  sqrt(mean((actual -  predicted)^2))
}

get_rmse = function (model,data, response){
  rmse (actual = subset(data, select = response, drop = T),
        predicted = predict(model,data))
}

# Calculates the number of parameters, excluding the variance parameter
# of the model:
parameter_number = function (model) {
  length(coef(model)) - 1
}
```

Now let us obtain the train RMSE,test RMSE and number of parameters (excluding variance parameter) for each model defined above:

```
# (a) train RMSE
fit_train_RMSE <- get_rmse (model = fit, data = train_boston, response = 'medv')
fit_smaller_train_RMSE <- get_rmse (model = fit_smaller, data = train_boston, response = 'medv')
fit_larger_train_RMSE <- get_rmse (model = fit_larger, data = train_boston, response = 'medv')

# (b) test RMSE
```

```
fit_test_RMSE <- get_rmse (model = fit, data = test_boston, response = 'medv')
fit_smaller_test_RMSE <- get_rmse (model = fit_smaller, data = test_boston, response = 'medv')
fit_larger_test_RMSE <- get_rmse (model = fit_larger, data = test_boston, response = 'medv')

# (c) the number of parameters, excluding the variance parameter:
fit_number <- parameter_number(fit)
fit_smaller_number <- parameter_number(fit_smaller)
fit_larger_number <- parameter_number(fit_larger)
```

Sumarising the models:

```
library(flextable)
table <- data.frame(
  Model = c("fit_smaller", "fit","fit_larger"),
  model_train_RMSE = c(fit_smaller_train_RMSE,fit_train_RMSE, fit_larger_train_RMSE),
  model_test_RMSE = c(fit_smaller_test_RMSE,fit_test_RMSE, fit_larger_test_RMSE),
  model_number = c (fit_smaller_number,fit_number,fit_larger_number))


t<-delete_part(flextable(table), part = "header")
t<- add_header(t,top=T,model_train_RMSE="model train RMS",model_test_RMSE= 'model test RMSE',model_numb



m<- colformat_num(t,j=c(2,3),digits = 7)
j<- colformat_num(m,j=c(4),digits = 0)



autofit(align(j,align = "center", part = "all"))
```

	model train RMS	model test RMSE	number of parameters
fit_smaller	4.6754649	4.7677458	13
fit	2.6416852	2.8132861	91
fit_larger	2.6220123	2.8339406	96

We see that 'fit_smaller' is the least flexible model, and 'fit_larger' is the most flexible model.
Moreover, we see that the Train RMSE decrease as flexibility increases. The Test RMSE is smallest for 'fit', so it is the model we believe to perform the best on data not used to train the model.
In general, 'fit_smaller' has high values of Train RMSE and Test RMSE. Hence underfitting.
fit_larger seem to have overfitting.
To conclude, fit is the best model out of these 3.