Polynomial regression

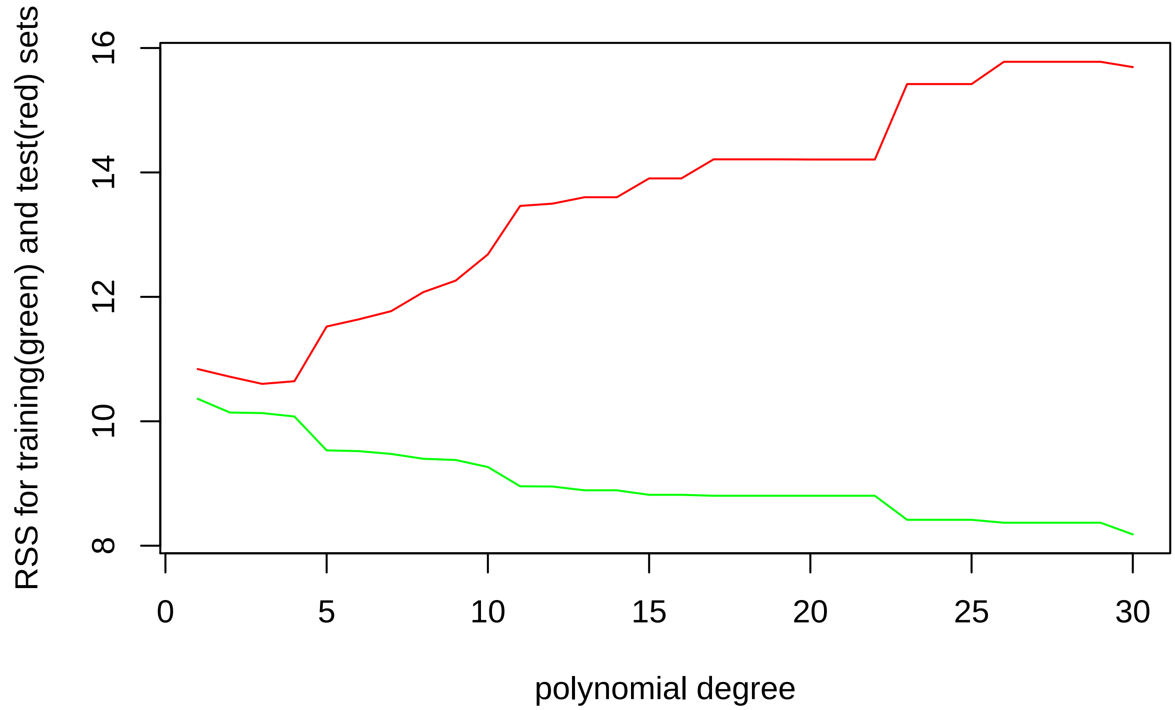
```
# prepare data
adv <- read.csv("~/GitRepos/MachineLearning2014/hw1/Advertising.csv")
adv$X <- NULL
smpsNum <- dim(adv)[1]
trainSize <- smpsNum * 2 / 3
testSize <- smpsNum - trainSize
trainInds <- sample(1:smpsNum, size = trainSize)
adv.train <- adv[trainInds, ]
adv.test <- adv[-trainInds, ]

#funcs
meanRss <- function(reals, preds) {
  return(mean((reals - preds)^2))
}

# get data for plots
trainRss <- c()
testRss <- c()
aic <- c()
bic <- c()
x <- c()
for (i in 1:30) {
  x[i] <- i
  l <- lm(Sales ~ poly(TV, i, raw=TRUE), data=adv.train)
  trainRss[i] <- meanRss(adv.train$Sales, predict(l, adv.train))
  testRss[i] <- meanRss(adv.test$Sales, predict(l, adv.test))
  aic[i] <- AIC(l)
  bic[i] <- BIC(l)
}
```

Let's plot graphs of training_set_rrs and test_set_rss vs polynomial_degree:

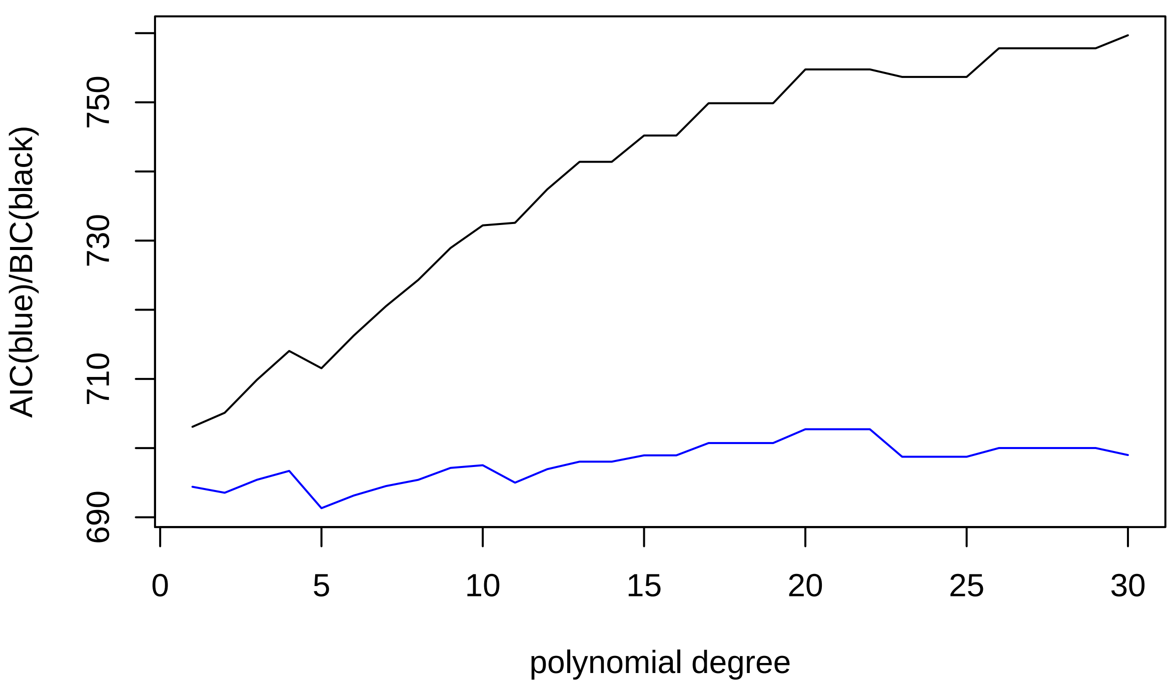
```
plot(x, testRss, type="l", col="red", ylim=range(c(testRss, trainRss)), xlab="polynomial degree", ylab=
lines(x, trainRss, col="green")
```



Analysing the figure above one can notice the higher polynomial degree the lower RSS for training data set and the higher RSS for test data set. So, increasing polynomial degree we make out model better fit training data and worse fit any new data. This looks like model overfitting.

Next figure shows graphs for AIC and BIC vs polynomial_degree:

```
plot(x, aic, type="l", col="blue", ylim=range(c(aic, bic)), xlab="polynomial degree", ylab="AIC(blue)/BI
lines(x, bic, col="black")
```



As we can see, increasing the degree we increase both measures (especially BIC) and, as a result, decrease model's quality.