# THE IMPORTANCE OF CORRECT MODEL IDENTIFICATION

This simulation illustrates why it is important to identify the correct model. We simulate 5000 replicated data sets of size 1000 from a polynomial regression model with modelled mean

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2$$

with $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1.2$. We record the estimates of the regression coefficients and $\sigma^2$ for the following models:

- **Over-simple:** `fit1`
$$\beta_0 + \beta_1 x_{i1}$$
- **Correct:** `fit2`
$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2$$
- **Over-complex:** `fit3`
$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \beta_3 x_{i1}^3$$

where the predictor $X_{i1}$ has a normal distribution with mean 5 and variance 1.

```
nreps<-5000
library(MASS)
set.seed(423)
X<-seq(0,10,by=0.01)
X<-cbind(X,X^2,X^3)
n<-nrow(X)
be<-rep(0,4)
be[2:4]<-c(2,1.2,0)
mean.vec<-cbind(rep(1,n),X)%*%be
```

The following code fits the three models for each of the 5000 replicates, records the estimates for the three models, and predictions for all of the data points.

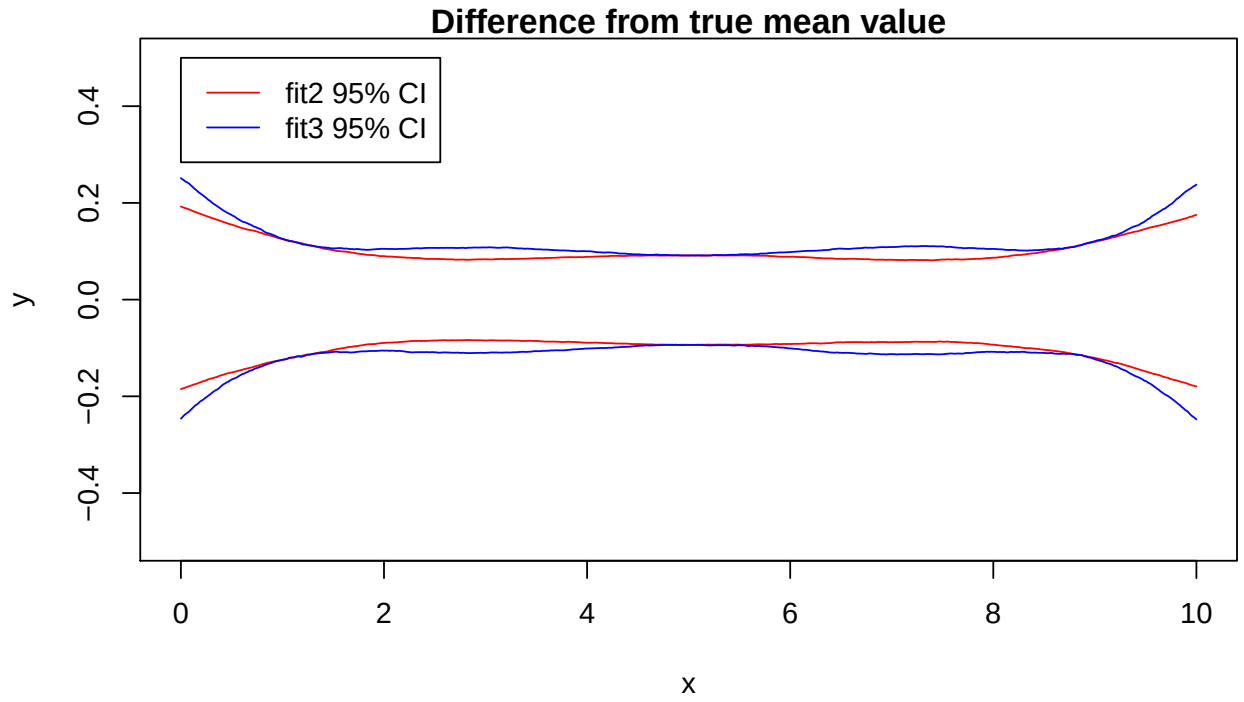
```
ests.mat1<-ests.mat2<-ests.mat3<-matrix(0,nrow=nreps,ncol=4)
bias.mat1<-bias.mat2<-bias.mat3<-matrix(0,nrow=nreps,ncol=4)
fitted.vals1<-fitted.vals2<-fitted.vals3<-matrix(0,nreps,n)
sigma.ests<-matrix(0,nreps,3)
for(i in 1:nreps){
    Y<-mean.vec+rnorm(n)
    fit1<-lm(Y~X[,1])
    ests.mat1[i,]<-c(coef(fit1),0,0)
    bias.mat1[i,]<-c(coef(fit1),0,0)-c(0,be[2:3],0)
    fitted.vals1[i,]<-fitted(fit1)
    sigma.ests[i,1]<-summary(fit1)$sigma^2
    fit2<-lm(Y~X[,1:2])
    ests.mat2[i,]<-c(coef(fit2),0)
    bias.mat2[i,]<-c(coef(fit2),0)-c(0,be[2:4])
    fitted.vals2[i,]<-fitted(fit2)
    sigma.ests[i,2]<-summary(fit2)$sigma^2
    fit3<-lm(Y~X)
    ests.mat3[i,]<-coef(fit3)
    bias.mat3[i,]<-coef(fit3)-c(0,be[2:4])
    fitted.vals3[i,]<-fitted(fit3)
    sigma.ests[i,3]<-summary(fit3)$sigma^2
}
```

We now summarize the biases (scaled by $\sqrt{n}$), variances and mean square errors (scaled by $n$):

```
bias1<-apply(bias.mat1,2,mean);bias2<-apply(bias.mat2,2,mean);bias3<-apply(bias.mat3,2,mean)
sqrt(n)*rbind(bias1,bias2,bias3)

:                [,1]         [,2]          [,3]          [,4]
: bias1 -632.12265814 379.65643460 -37.966300847  0.000000e+00
: bias2    0.04985534  -0.02675668   0.002018281  0.000000e+00
: bias3    0.05109832  -0.02825200   0.002392297 -2.493443e-05

var1<-apply(ests.mat1,2,var);var2<-apply(ests.mat2,2,var);var3<-apply(ests.mat3,2,var)
n*rbind(var1,var2,var3)

:           [,1]       [,2]       [,3]        [,4]
: var1  3.999533  0.1186894 0.00000000 0.000000000
: var2  9.222077  1.9327762 0.01778853 0.000000000
: var3 16.502553 12.2278861 0.65556719 0.002821731

mse1<-var1+bias1^2;mse2<-var2+bias2^2;mse3<-var3+bias3^2; n*rbind(mse1,mse2,mse3)

:              [,1]         [,2]         [,3]        [,4]
: mse1 3.995831e+05 1.441391e+05 1441.4400000 0.000000000
: mse2 9.224563e+00 1.933492e+00    0.0177926 0.000000000
: mse3 1.650516e+01 1.222868e+01    0.6555729 0.002821732
```

We now compare the fitted values for `fit2` and `fit3` to the true model mean; the model `fit1` produces biased fitted values, so we omit that comparison and examine deviations from the true value.

```
true.values<-cbind(1,X) %*% be
pred.var2<-apply(fitted.vals2,2,quantile,prob=c(0.025,0.975))
pred.var3<-apply(fitted.vals3,2,quantile,prob=c(0.025,0.975))
par(mar=c(4,4,1,2));plot(X[,1],true.values-true.values,type='n',xlab='x',ylab='y',ylim=range(-.5,.5))
lines(X[,1],pred.var2[1,]-true.values,col='red');lines(X[,1],pred.var2[2,]-true.values,col='red')
lines(X[,1],pred.var3[1,]-true.values,col='blue');lines(X[,1],pred.var3[2,]-true.values,col='blue')
legend(0,0.5,c('fit2 95% CI','fit3 95% CI'),col=c('red','blue'),lty=c(1,1))
title('Difference from true mean value')
```




Thus the estimator and prediction variance and MSE are **larger** for the over-complex model that includes the spurious cubic term.