# Bias Variance Assignment

*Matt Richey*

*2/18/2020*

## Assignment 1

- For the underlying "true" model, use polynomials of degree =1..4. In each case, repeat the process above with linear models of degree up to about 15 or so. Does the "optimal" flexibility correspond to the degree of the underlying true model?

Of course, build a ggplot version of Figure 2.12 of ISLR using your results from above.

```
f1 <- function(x) 1+x
f2 <- function(x) (x-1)*(x+1)
f3 <- function(x) x*(x-1)*(x-2)
f4 <- function(x) {(x-1.5)*(x-1)*x*(x+1)}
```

We are going to be repeating the process of building training data, so make a simple function

```
buildData <- function(func,sizeDS,sig,xMin = -3, xMax = 3){
  ##predictor
  x<-runif(sizeDS,xMin, xMax) # inputs
  ## Repsonse
  y<-func(x)+rnorm(sizeDS,0,sig) #realized values f(x)+noise
  ## Put in a data frame
  data.frame(x,y)
}
```

Build a Bias-Variance-MSE Calculator

```
biasVarTO.lm <- function(func,form,sizeDS,numDS,x0){
  allVals <- matrix(ncol=2,nrow=numDS)
  for(m in 1:numDS){
    ##the
    mod <- lm(formula(form),buildData(func,sizeDS,sig))
    pred <- predict(mod,newdata=data.frame(x=x0))
    allVals[m,1] <- pred
  }
  allVals[,2] <- func(x0)+rnorm(numDS,0,sig)

  allVals.df <- data.frame(pred=allVals[,1],true=allVals[,2])
  mse <- with(allVals.df,mean((pred-true)^2))
  var0 <- with(allVals.df,var(pred))
  bias2 <- with(allVals.df,mean(pred-true))^2
  noise <- sig^2
  c(mse,var0,bias2,noise)
}
```

Test it out....

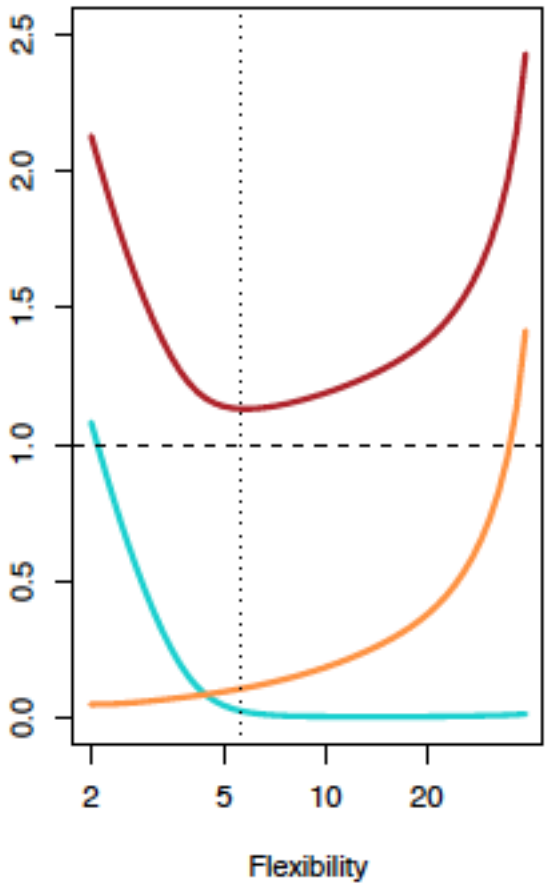



Figure 1: Figure 2.12

```
sig <- 2
sizeDS <- 100
numDS <- 200
x0 <- 0.5
```

```
form <- "y ~ x +I(x^2)"
theFunc <- f1
biasVarTO.lm(theFunc,form,sizeDS,numDS,x0)
```

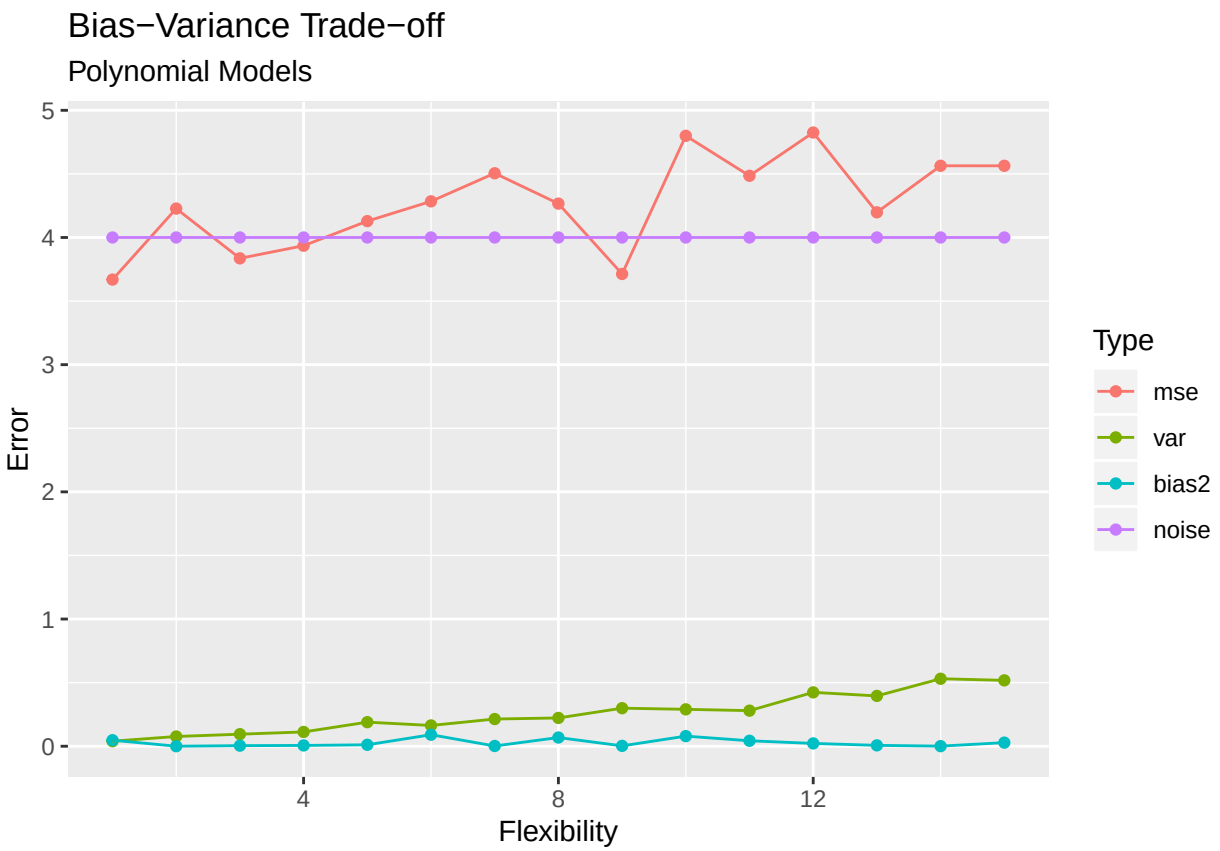
```
## [1] 3.59162517 0.09809709 0.30532403 4.00000000
```

Now run this on a variety of underlying models (linear, quadratic, cubic, quartic)

```
maxDegree <- 15
sizeDS <- 100
numReps <- 200 ## Increase this for more accuracy of the estimates
##Starter Formula
form0 <- "y ~ "

##A place to stash the results
res <- matrix(nrow=maxDegree,ncol=4)
for(k in 1:maxDegree){
  ##Build up the formula
  form0 <- sprintf("%s + I(x^%s)",form0,k)
  ##print(form0)
  res[k,] <-biasVarTO.lm(theFunc,form0,sizeDS,numDS,x0)
  ##print(res[k,])
}
```

Build a plot from this information.

```
res.df <- data.frame(flex=1:maxDegree,res)
names(res.df) <- c("flex","mse","var","bias2","noise")

res.df %>%
  gather(Type,err,mse:noise) %>%
  ##put these in order
  mutate(Type=factor(Type,levels=c("mse","var","bias2","noise"))) %>%
  ggplot()+
  geom_point(aes(flex,err,color=Type))+
    geom_line(aes(flex,err,color=Type))+
  labs(x="Flexibility",
       y="Error",
       title="Bias-Variance Trade-off",
      subtitle="Polynomial Models")
```




It looks as if the minimal MSE occurs somewhere around degree=1 or 2. As it should be.

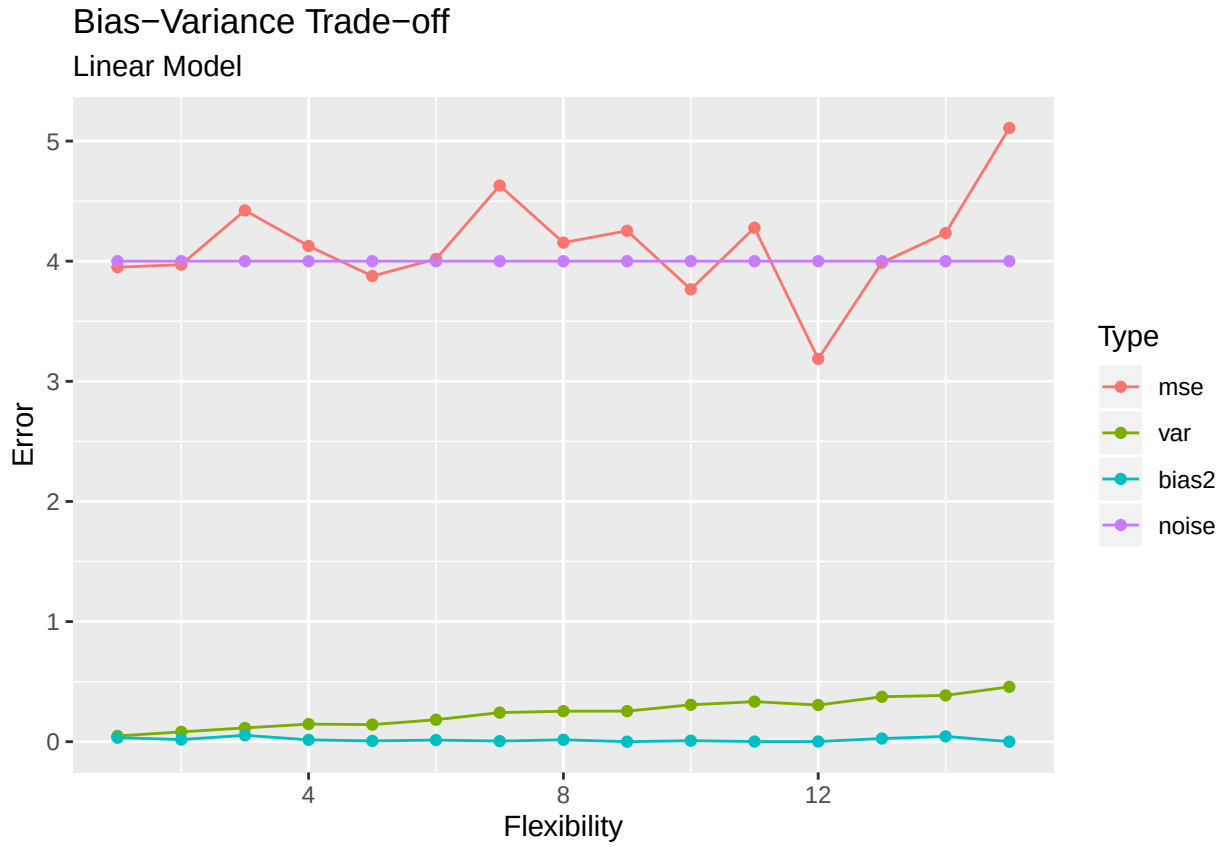
Now build a simple function to handle any underlying model. The arguments include all the relevant parameters (sizeDS etc). It returns the

```
buildRes <- function(maxDegree, theFunc,sizeDS,numReps){
  ##Starter Formula
  form0 <- "y ~ "

  ##A place to stash the results
  res <- matrix(nrow=maxDegree,ncol=4)
  for(k in 1:maxDegree){
    ##Build up the formula
    form0 <- sprintf("%s + I(x^%s)",form0,k)
    res[k,] <-biasVarTO.lm(theFunc,form0,sizeDS,numDS,x0)
  }
  res.df <- data.frame(flex=1:maxDegree,res)
  names(res.df) <- c("flex","mse","var","bias2","noise")

  res.df %>%
    gather(Type,err,mse:noise)
}
```

## Let it rip. . . . .

Use the linear function to start. Plot the results

```
 ##put these in order
buildRes(15,f1,100,50) %>%
  mutate(Type=factor(Type,levels=c("mse","var","bias2","noise"))) %>%
  ggplot()+
  geom_point(aes(flex,err,color=Type))+
    geom_line(aes(flex,err,color=Type))+
  labs(x="Flexibility",
       y="Error",
       title="Bias-Variance Trade-off",
      subtitle="Linear Model")
```

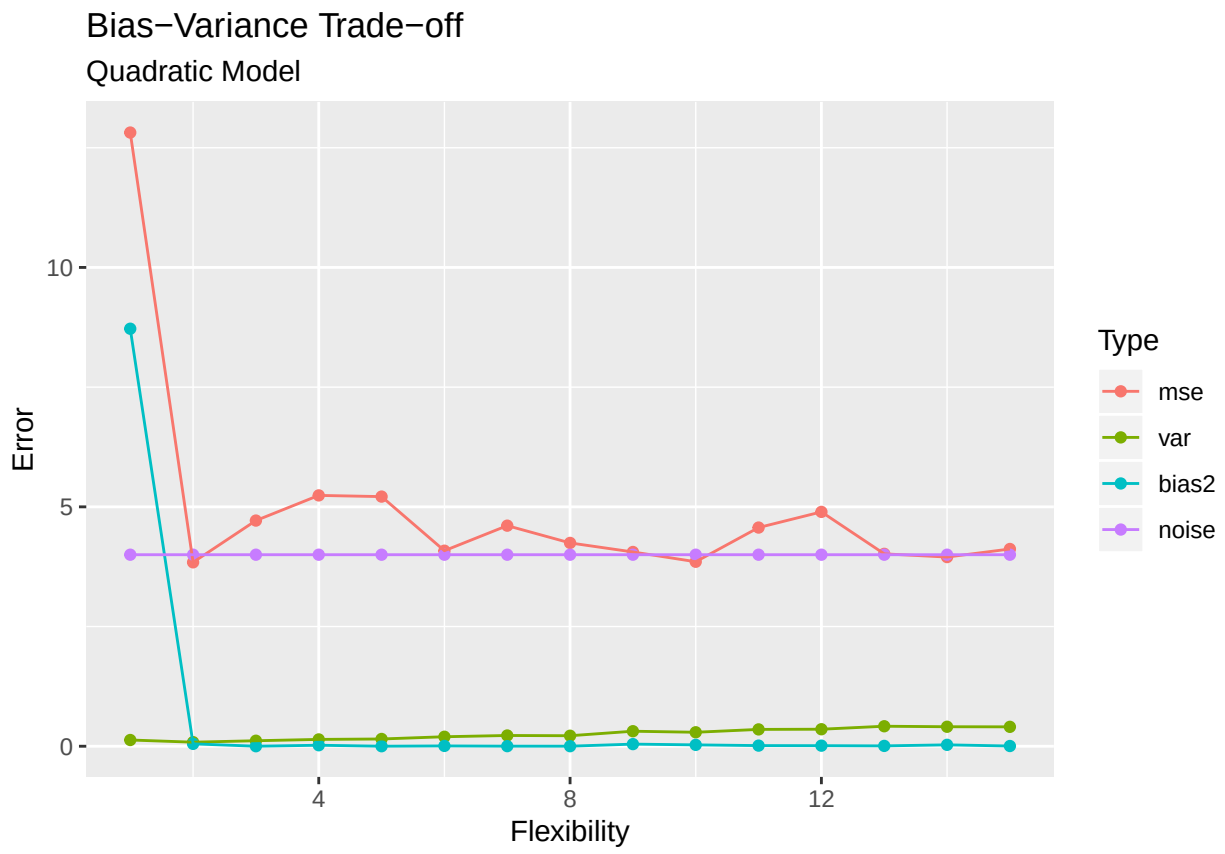



Looks minimal at Flex=1 or 2. Good.

Repeat for other degrees

Degree=2

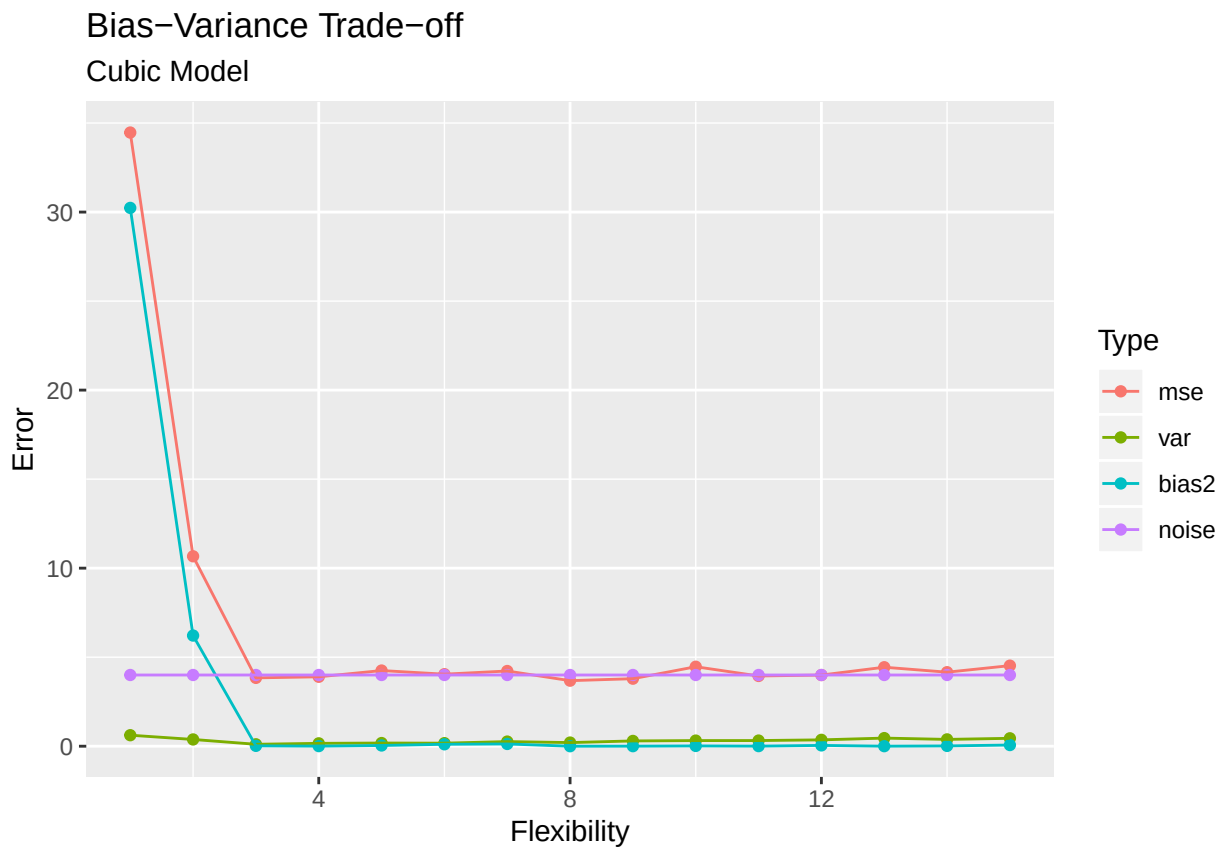
```
 ##put these in order
buildRes(15,f2,100,50) %>%
  mutate(Type=factor(Type,levels=c("mse","var","bias2","noise"))) %>%
  ggplot()+
  geom_point(aes(flex,err,color=Type))+
    geom_line(aes(flex,err,color=Type))+
  labs(x="Flexibility",
       y="Error",
       title="Bias-Variance Trade-off",
      subtitle="Quadratic Model")
```




Min MSE at 2 or so.

Degree=3

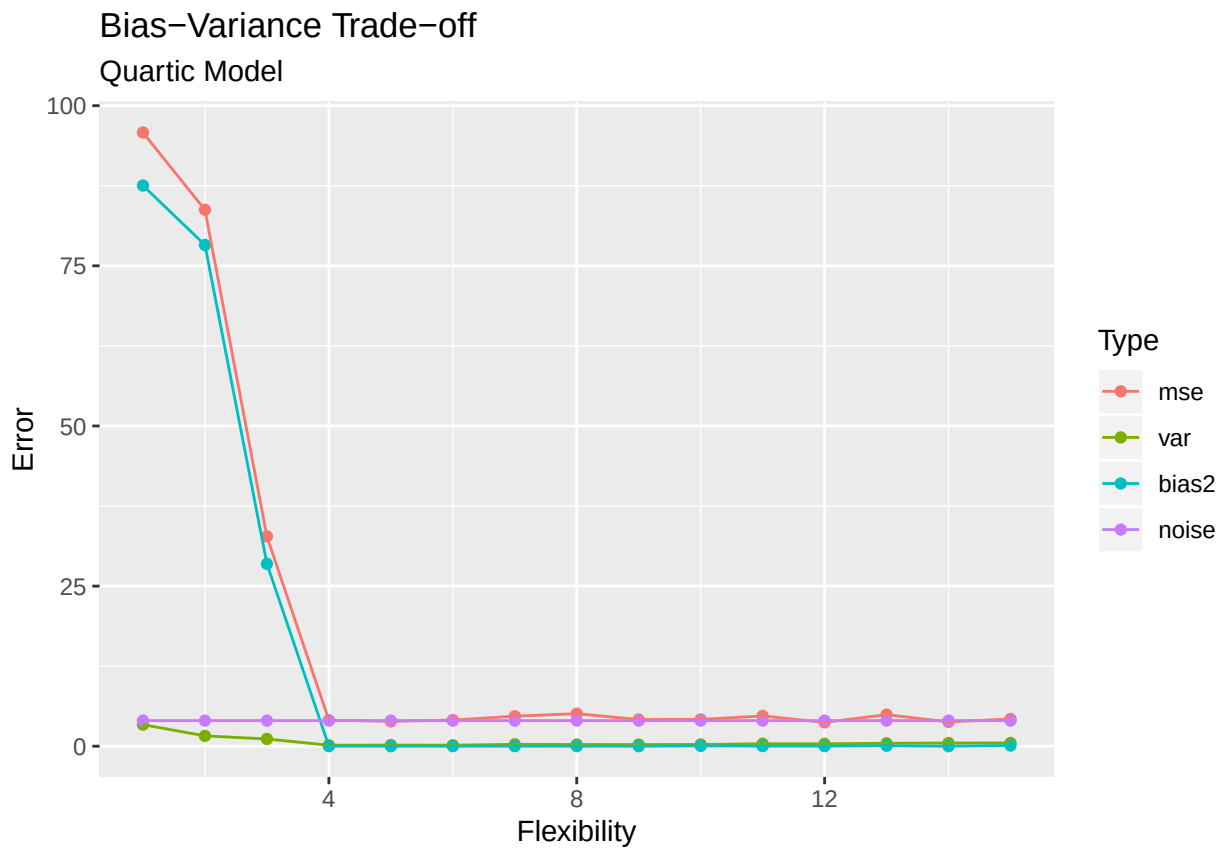
```
 ##put these in order
buildRes(15,f3,100,50) %>%
  mutate(Type=factor(Type,levels=c("mse","var","bias2","noise"))) %>%
  ggplot()+
  geom_point(aes(flex,err,color=Type))+
    geom_line(aes(flex,err,color=Type))+
  labs(x="Flexibility",
       y="Error",
       title="Bias-Variance Trade-off",
      subtitle="Cubic Model")
```




The minimal MSE is obtained first at Flex=2, as it should be.

Degree=4

```
 ##put these in order
buildRes(15,f4,100,50) %>%
  mutate(Type=factor(Type,levels=c("mse","var","bias2","noise"))) %>%
  ggplot()+
  geom_point(aes(flex,err,color=Type))+
    geom_line(aes(flex,err,color=Type))+
  labs(x="Flexibility",
       y="Error",
       title="Bias-Variance Trade-off",
      subtitle="Quartic Model")
```




The minimal MSE is obtained first at Flex=4, as it should be.

## Assignment 2

Repeat using KNN regression (i.e., using knn.reg). Note that in this case, flexibility increases as the control paramenter $k$ (=number of neighbors) decreases. Again, your goal is to build a version of Figure 2.12 of ISLR. Note, we usually put flexibilty on the horizontal axis with the lowest flexibility on the left and the highest on the right.

Be careful, the knn.reg function requires that you put the input data is a very specific form. This was described in the RMarkdown document from the first day of class.

## Setup

```
library(FNN)
```

Note it depends on the function buildData

```
biasVarTO3.knn <- function(kVal,func,sizeDS,numDS,x0){
  allVals <- matrix(ncol=2,nrow=numDS)
  for(m in 1:numDS){
    ##the
    train.df <- buildData(func,sizeDS,sig)
    train.X <- as.matrix(train.df[c("x")])
    test.X <- as.matrix(x0)
    train.Y <- as.matrix(train.df[c("y")])
```

```
    mod.knn <- knn.reg(train.X,test.X,train.Y,k=kVal)
    allVals[m,1] <- mod.knn$pred
  }
  allVals[,2] <- func(x0)+rnorm(numDS,0,sig)
  allVals.df <- data.frame(pred=allVals[,1],true=allVals[,2])
  mse <- with(allVals.df,mean((pred-true)^2))
  var0 <- with(allVals.df,var(pred))
  bias2 <- with(allVals.df,mean(pred-true))^2
  noise <- sig^2
  c(mse,var0,bias2,noise)
}
```

The buildData Function

```
buildData <- function(func,sizeDS,sig,xMin = -1, xMax = 1){
  ##predictor
  x<-runif(sizeDS,xMin, xMax) # inputs
  ## Repsonse
  y<-func(x)+rnorm(sizeDS,0,sig) #realized values f(x)+noise
  ## Put in a data frame
  data.frame(x,y)
}
```

## Example

```
f3 <- function(x) x*(x-1)*(x+1)
sizeDS <- 100
sig <-0.5
numReps <- 25
```

```
kVal <- 20
x0 <- 0.5
(vals <- biasVarTO3.knn(kVal,f3,sizeDS,numReps,x0))
```

```
## [1] 0.15659761 0.01264731 0.01985178 0.25000000
```

```
vals <- round(vals,3)
sprintf("MSE=%s, Var=%s, Bias^2=%s, Noise=%s",vals[1],vals[2],vals[3],vals[4])
```

```
## [1] "MSE=0.157, Var=0.013, Bias^2=0.02, Noise=0.25"
```

## Bias-Variance Plot for KNN.reg

```
func <- f3
sizeDS <- 50
numReps <- 500 ## Increase this for more accuracy of the estimates
##Starter Formula
```

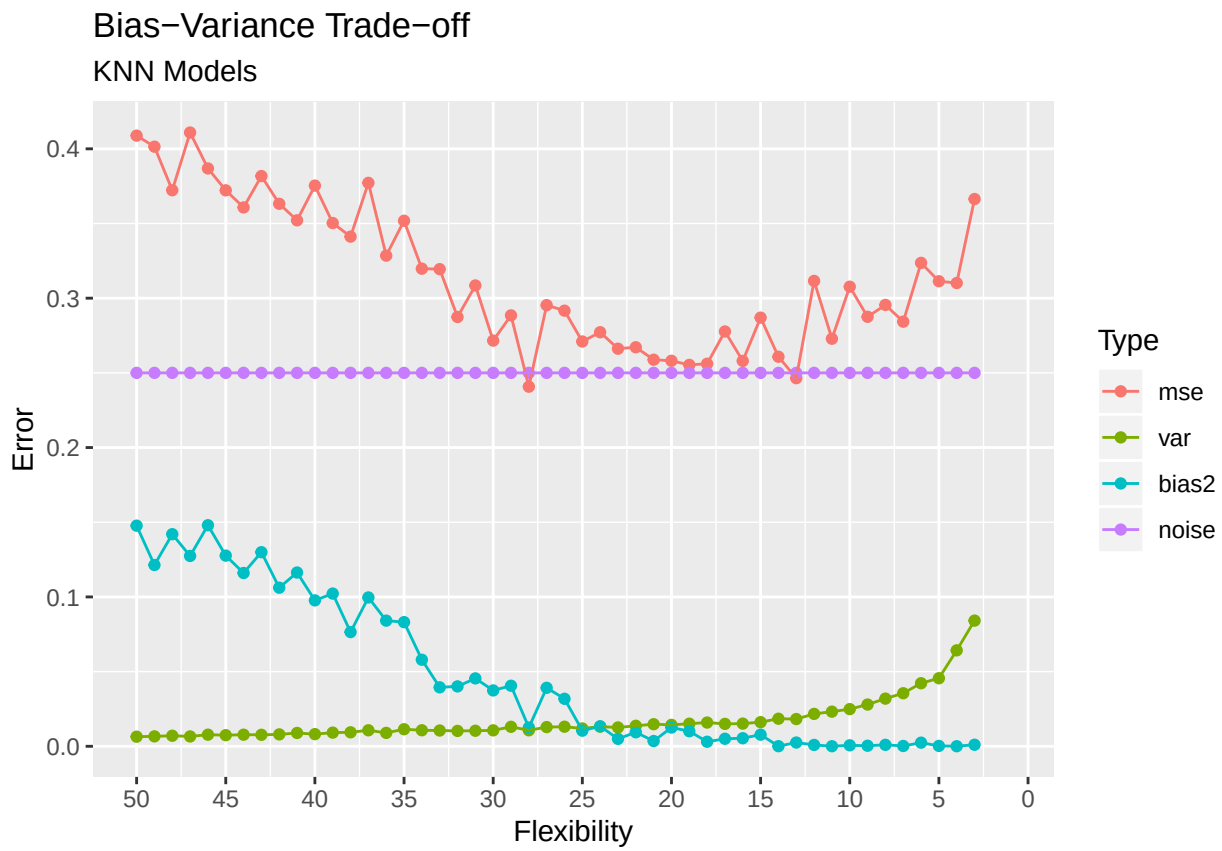
```
maxK <- 50
##A place to stash the results
res <- matrix(nrow=maxK,ncol=4)
##Skip k=2!
for(kval in 3:maxK){
  res[kval,] <- biasVarTO3.knn(kval,func,sizeDS,numReps,0.5)
}
```

Build a plot from this information.

```
res.df <- data.frame(flex=-(1:maxK),res)
names(res.df) <- c("flex","mse","var","bias2","noise")

res.df %>%
  gather(Type,err,mse:noise) %>%
  ##put these in order
  mutate(Type=factor(Type,levels=c("mse","var","bias2","noise"))) %>%
  ggplot()+
  geom_point(aes(flex,err,color=Type))+
    geom_line(aes(flex,err,color=Type))+
  scale_x_continuous(breaks=seq(-50,0,by=5),
                     labels=seq(50,0,by=-5))+
  labs(x="Flexibility",
       y="Error",
       title="Bias-Variance Trade-off",
      subtitle="KNN Models")
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```

```
## Warning: Removed 8 rows containing missing values (geom_path).
```




Note that this plotted using larger values of k on left (lower flexibility).

It looks as if the minimal MSE occurs somewhere around k=10-15.