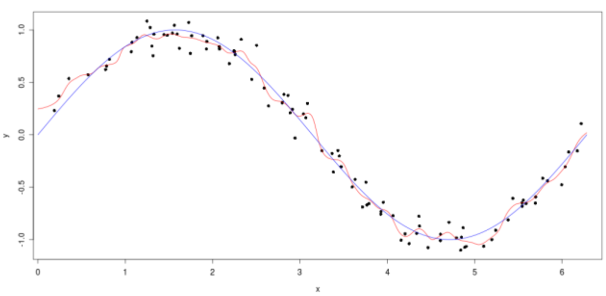
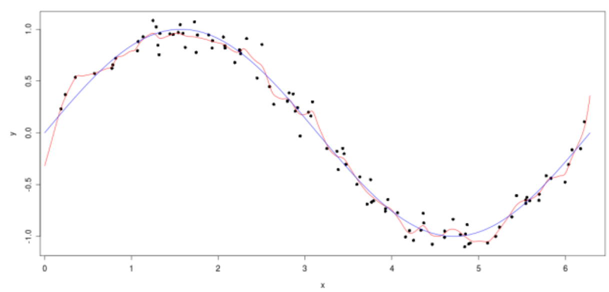
EKREM KOCAGUNELI – STAT745- HW#7

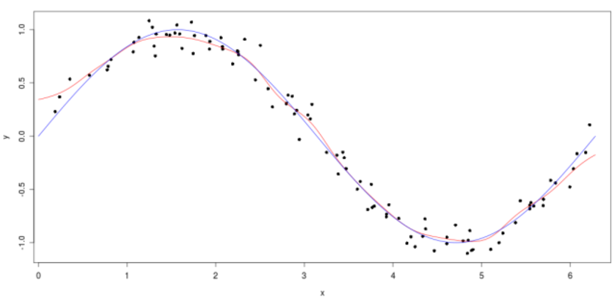
Question 1: On the left d=0 figures and on the right d=1 figures are given.



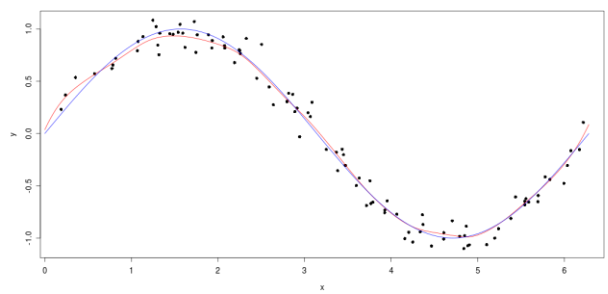
λ=0.01



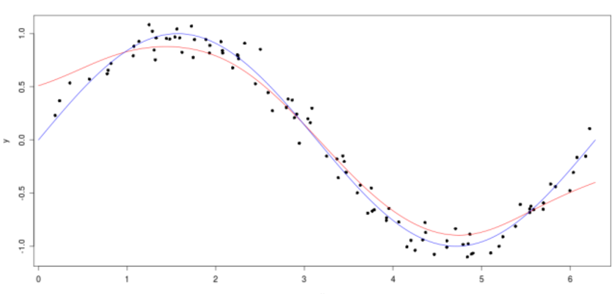
λ=0.01



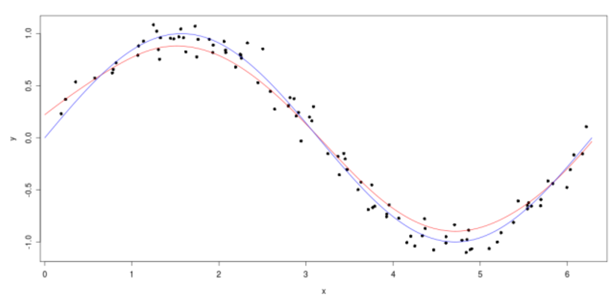
λ=0.1



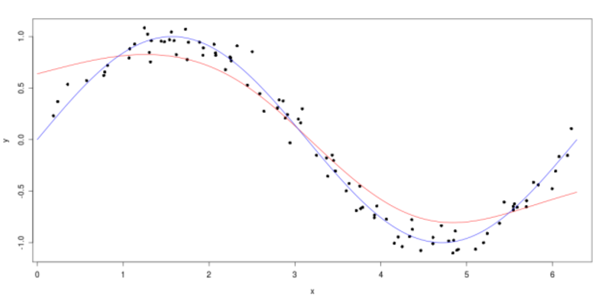
λ=0.1



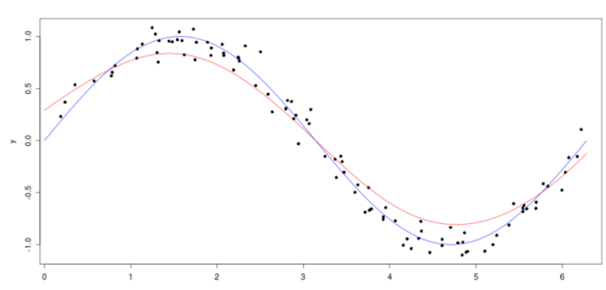
λ=0.5



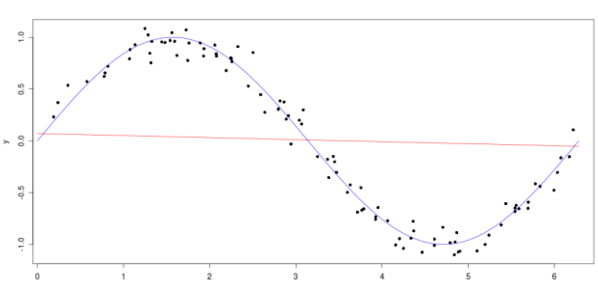
λ=0.5



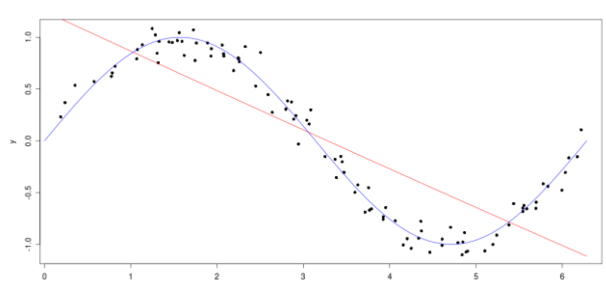
λ=1



λ=1



λ=100



λ=100

Question 2: On the previous page the figures for loess d=0 and d=1 are given. Depending on those figures we can comment on both 2 things: 1) Bias-Variance change in loess w.r.t. the value of λ and 2) the changes in loess w.r.t. d.

1) **Changes w.r.t. λ:** As we can see from the figures, the small values of λ make the loess too sensitive, i.e. for low λ values loess (both d=0 and d=1) have high variance and low bias. However, as we increase the λ values, the variance starts to decrease and the bias starts to increase. For extreme values of λ (such as 100) the loess reduces to a simple linear regression, i.e. too high bias and very low variance.

2) **Changes w.r.t. d:** The difference in loess between d=0 and d=1 is that d=0 does not have an intercept term whereas d=1 has an intercept term (e.g. intercept term for d=1 in $y=\theta_1*x + \theta_0$ is θ_0). The advantage of having an intercept term affects the success of loess particularly in the end points: When we look at the plots, we see that loess with d=1 is a more successful fit on the left and right ends of the plots in comparison to d=0. The importance of having an intercept term is more evident for high values of λ (i.e. when we have high bias and low variance condition): When we look at loess fits for λ=100, we see that the intercept is very important and d=1 is a much better fit than d=0.

THE CODE IS GIVEN BELOW:

```
bls<-function(x0,x,y){
 x0%*%solve(t(x)%*%x,t(x)%*%y)
}
knn<-function(x0,x,y,k){
   x=as.matrix(x)
   p=dim(x)[2]
   n=dim(x)[1]
   dis=rep(0,n)
   for(i in 1:p){
     dis= (x0[i]-x[,i])^2+dis
   }
   ind=order(dis)[1:k]
   mean(y[ind])
}
```

EKREM KOCAGUNELI – STAT745- HW#7

```
loess0<-function(x0,x,y,kern,lam){
  x=as.matrix(x)
  p=dim(x)[2]
  n=dim(x)[1]
  dis=rep(0,n)
  for(i in 1:p){
    dis= (x0[i]-x[,i])^2+dis
  }

  ##My code is below
  w = kern(dis,lam)
  return (sum(w*y)/sum(w))
  ##Your code here

}

loess1<-function(x0,x,y,kern,lam){
  x=as.matrix(x)
  p=dim(x)[2]
  n=dim(x)[1]
  dis=rep(0,n)
  for(i in 1:p){
    dis= (x0[i]-x[,i])^2+dis
  }
  ##My code is below
  w = kern(dis,lam)
  ## below we calculate the teta1
  ky = sum(w*y)
  kx = sum(w*x)
  kxy = sum(w*x*y)
  k = sum(w)
```

```
  kxx = sum(w*x*x)

  teta1 = (kxy*k - kx*ky)/(kxx*k - kx*kx)

  ## below we calculate the teta0

  teta0 = (ky*kxx - kxy*kx)/(k*kxx - kx*kx)


  return(teta1*x0 + teta0)

  ##Your code here

}


kern<-function(x,lam){

  exp(-x/lam)/lam

}

######

## sin(x) Regression Example

######

set.seed(100)

x=runif(100,0,2*pi)

y=sin(x)+rnorm(100,,0.1)


xgrid=seq(0,2*pi,length=500)

n=length(xgrid)

ygrid=vector(length=n)

k=15

lam=0.01

for(i in 1:n){

  ##ygrid[i]= loess0(xgrid[i],x,y,kern,lam) ##loess d=0 , d=1

  ygrid[i]= loess1(xgrid[i],x,y,kern,lam) ##loess d=0 , d=1

}

plot(x,y,pch=16)

lines(xgrid,ygrid,col=c("red"))

lines(xgrid,sin(xgrid),col=c("blue"))
```