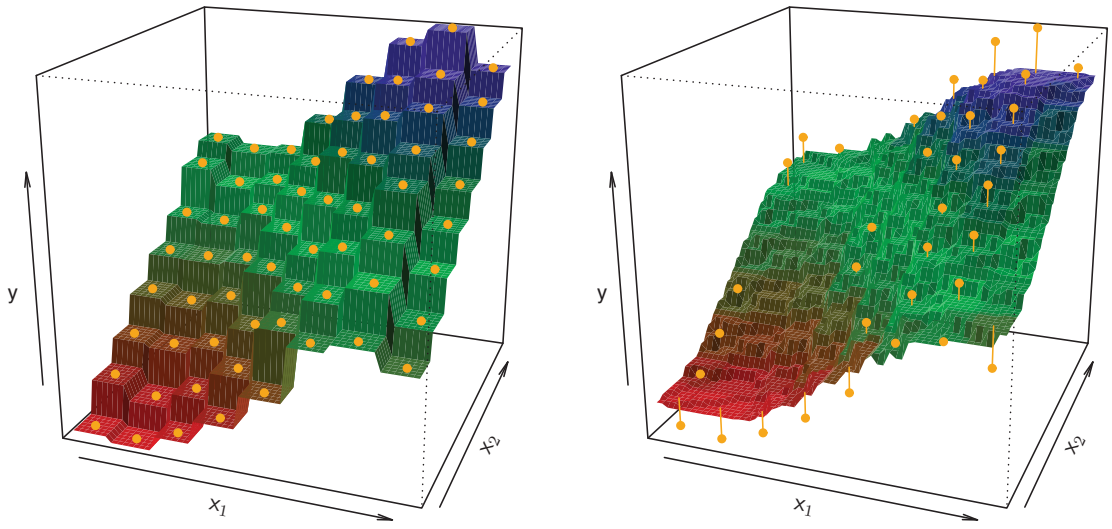
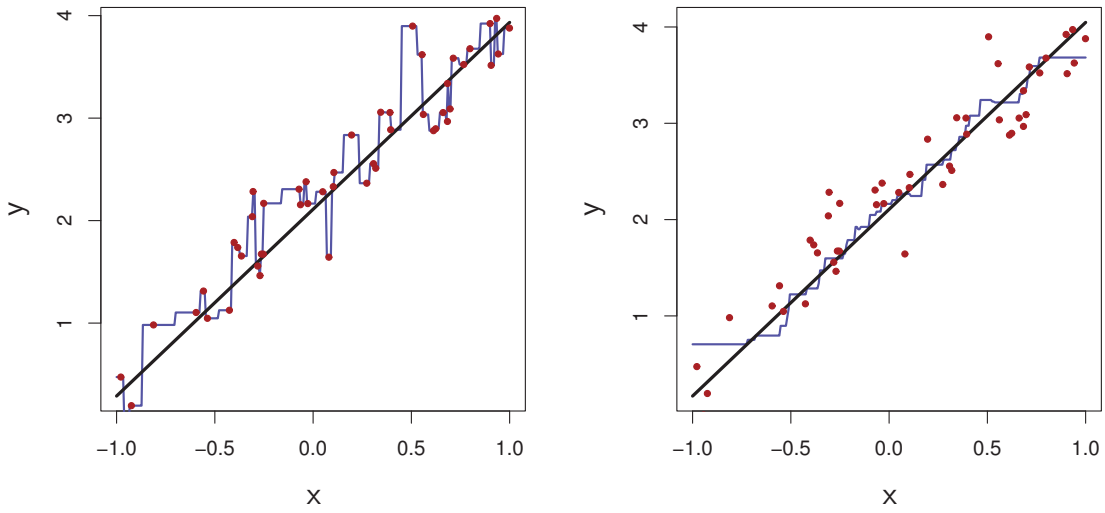
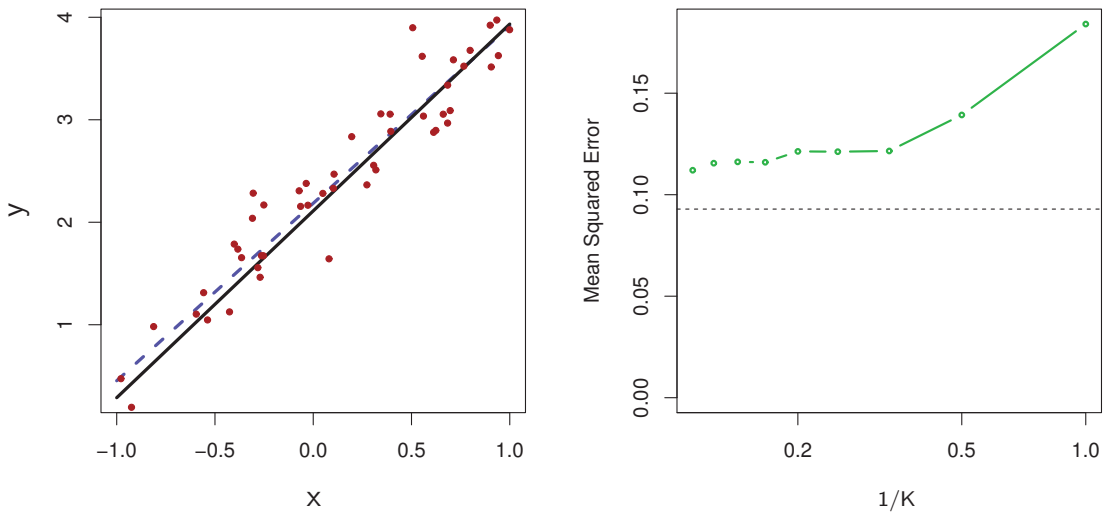


Plots of $\hat{f}(X)$ using KNN regression on a two-dimensional data set with 64 observations (orange dots). Left: $K = 1$ results in a rough step function fit. Right: $K = 9$ produces a much smoother fit.

- In what setting will a parametric approach such as least squares linear regression outperform a non-parametric approach such as KNN regression?
- The answer is simple: the parametric approach will outperform the non-parametric approach if the parametric form that has been selected is close to the true form of f.

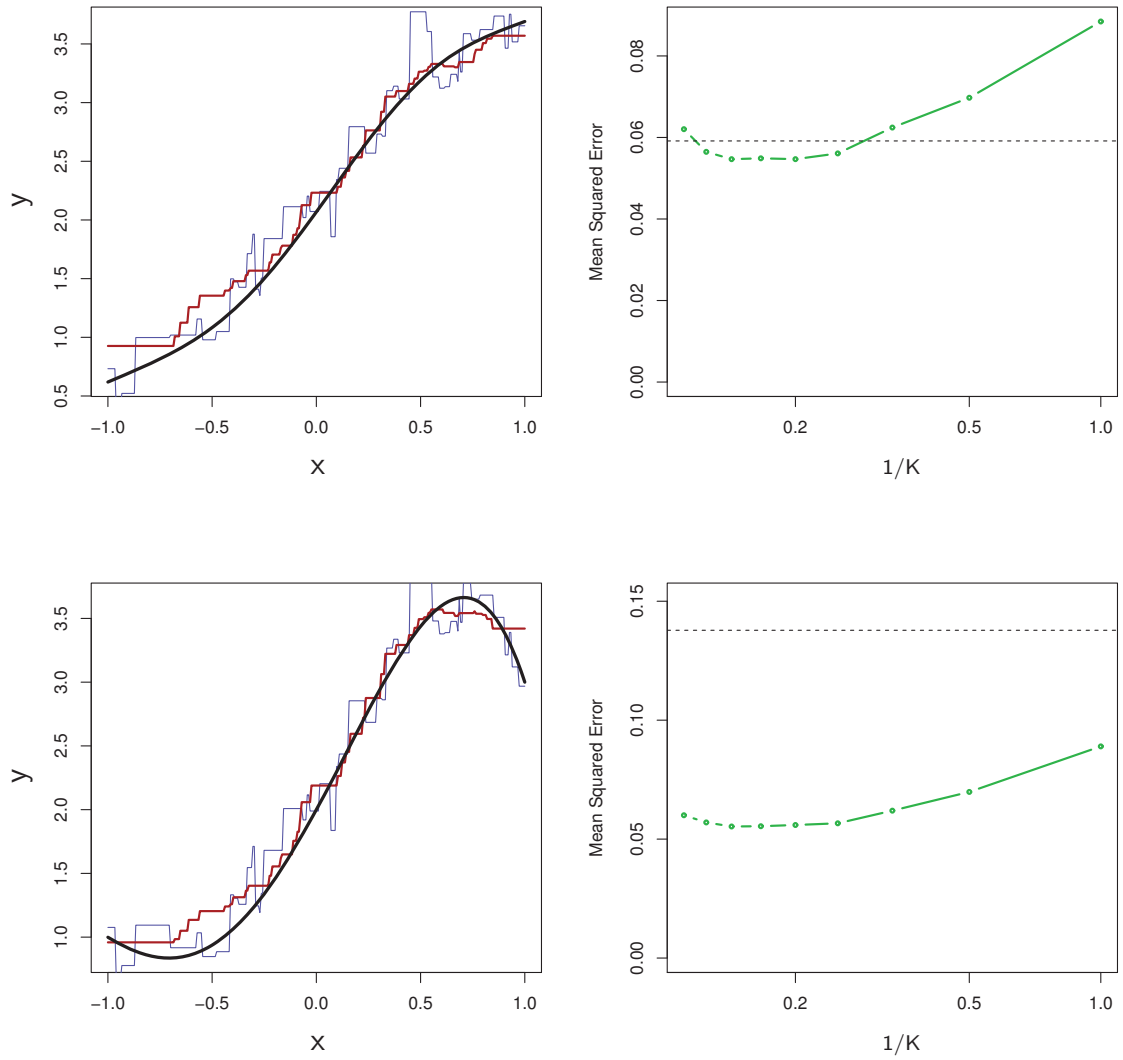


Plots of $\hat{f}(X)$ using KNN regression on a one-dimensional data set with 50 observations. The true relationship is given by the black solid line. Left: The blue curve corresponds to $K = 1$ and interpolates (i.e. passes directly through) the training data. Right: The blue curve corresponds to $K = 9$, and represents a smoother fit.



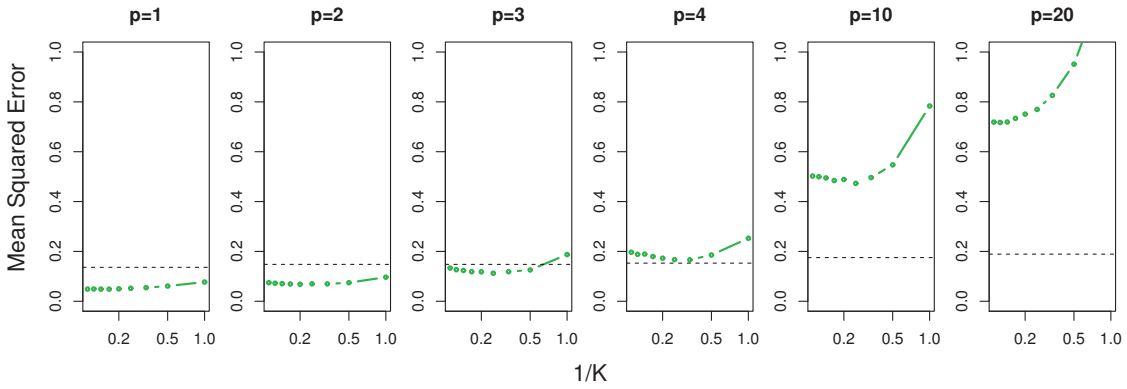
The same data set shown in the previous figure is investigated further. Left: The blue dashed line is the least squares fit to the data. Since $f(X)$ is in fact linear (displayed as the black line), the least squares regression line provides a very good estimate of $f(X)$. Right: The dashed horizontal line represents the least squares test set MSE, while the green solid line corresponds to the MSE for KNN as a function of $1/K$ (on the log scale). Linear regression achieves a lower test MSE than does KNN regression, since $f(X)$ is in fact linear. For KNN regression, the best results occur with a very large value of K, corresponding to a small value of $1/K$.

- In practice, the true relationship between X and Y is rarely exactly linear.



Top Left: In a setting with a slightly non-linear relationship between X and Y (solid black line), the KNN fits with $K = 1$ (blue) and $K = 9$ (red) are displayed. Top Right: For the slightly non-linear data, the test set MSE for least squares regression (horizontal black) and KNN with various values of $1/K$ (green) are displayed. Bottom Left and Bottom Right: As in the top panel, but with a strongly non-linear relationship between X and Y.

- In a real life situation in which the true relationship is unknown, one might suspect that KNN should be favored over linear regression because it will at worst be slightly inferior to linear regression if the true relationship is linear, and may give substantially better results if the true relationship is non-linear.
- But in reality, even when the true relationship is highly non-linear, KNN may still provide inferior results to linear regression.
- In higher dimensions, KNN often performs worse than linear regression.



Test MSE for linear regression (black dashed lines) and KNN (green curves) as the number of variables p increases. The true function is non–linear in the first variable, as in the lower panel in the previous figure, and does not depend on the additional variables. The performance of linear regression deteriorates slowly in the presence of these additional noise variables, whereas KNN's performance degrades much more quickly as p increases.

- As a general rule, parametric methods will tend to outperform non-parametric approaches when there is a small number of observations per predictor.
- Even when the dimension is small, we might prefer linear regression to KNN from an interpretability standpoint.

- If the test MSE of KNN is only slightly lower than that of linear regression, we might be willing to forego a little bit of prediction accuracy for the sake of a simple model that can be described in terms of just a few coefficients, and for which p-values are available.

Performing K-Nearest Neighbors Regression in R

Now we will perform KNN regression on the advertising data in order to predict sales on the basis of TV, radio, and newspaper budgets.

We perform KNN regression using the knnreg() function, which is part of the caret library. The function requires three inputs:

1. A matrix or data frame of training set predictors.
2. A numeric vector of outcomes.
3. A value for K, the number of nearest neighbors to be considered.

We begin by using the sample() function to split the set of observations into two halves, by selecting a random subset of 100 observations out of the original 200 observations.

```
> library(caret)
> Adv <- read.csv("Advertising.csv",header=T)
> attach(Adv)
> names(Adv)
[1] "X"         "TV"        "radio"     "newspaper" "sales"
> set.seed(1)
> train <- sample(200, 100)
```

We then use the cbind() function, short for column bind, to bind the TV, radio, and newspaper variables together into two matrices, one for the training set and the other for the test set.

Note: Because the KNN regression predicts the value of the response for a given test observation by identifying the observations that are nearest to it, the scale of the variables matters.

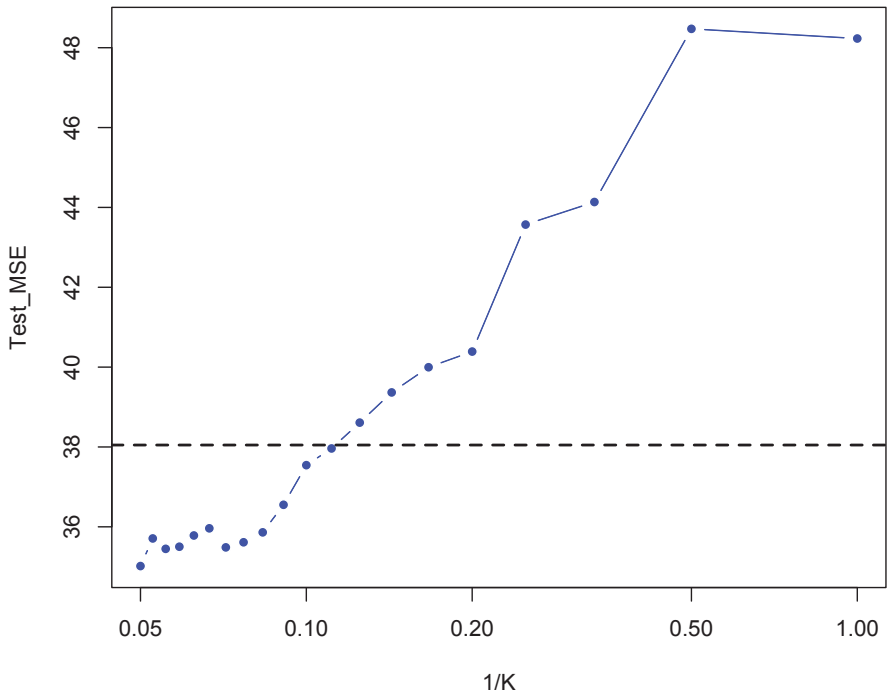
```
> train_X <- scale(cbind(TV, radio, newspaper)[train, ] )
> test_X <- scale(cbind(TV, radio, newspaper)[-train, ] )
> train_sales <- sales[train]
```

We now fit KNN regression using only the observations corresponding to the training set for different values of K. Then, we use the predict() function to estimate the response for observations corresponding to the test set and we use the mean() function to calculate the test MSE.

```
> Test_MSE_knn <- data.frame(matrix(NA, nrow=20, ncol=2))
> names(Test_MSE_knn) <- c("1/K", "Test_MSE")
>
> for(i in 1:20){
+         knn_fit <- knnreg(train_X, train_sales, k = i)
+         knn_pred <- predict(knn_fit, test_X)
+         Test_MSE_knn[i,1] <- 1/i
+         Test_MSE_knn[i,2] <- mean((sales - knn_pred)[-train]^2)
+ }
```

We also fit a linear regression to calculate the test MSE. We plot the test MSE of KNN and linear regression.

```
> lm_fit <- lm(sales ~ TV + radio + newspaper , data=Adv, subset=train)
> lm_pred <- predict(lm_fit, Adv)[-train]
> Test_MSE_lm <- mean((sales - lm_pred)[-train]^2)
>
> plot(Test_MSE_knn[,1:2], type="b", log="x", col="blue", pch=16)
> abline(Test_MSE_lm,0, col="black", lwd=2, lty="dashed")
```



The test set MSE for least squares regression (horizontal black) and KNN with various values of $1/K$ (blue) are displayed. We can see that KNN outperforms the linear regression for $K \geq 10$.