**Input and output: Explanatory or Response?**

- **Input:** predictors, covariates, explanatory variables, independent variables, features, inputs, or sometimes just variables, denoted by $\mathbf{X}$.
- **Output:** response variables, dependent variable, or outputs, denoted by $Y$.

**Regression or classification?**

- **Regression**: A response variable ($Y$) is quantitative.
- **Classification**: A response variable ($Y$) is categorical or qualitative.

- $n =$ the number of distinct data points, observations.
- $p =$ the number of variables
- $\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$ is a $n \times p$ matrix.

- $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{pmatrix}$ is a vector of length $p$, containing the $p$ measurements for the $i$th observation.
- $\mathbf{x}_j = \begin{pmatrix} x_{1j} \\ \vdots \\ x_{nj} \end{pmatrix}$ is a vector of length $n$, containing the $n$ measurements for the $j$th variable.
- Therefore, $\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_p)$

1.1. **Notation.**

1.2. **Statistical decision theory.**

- Let $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}$ with joint probability $P(x, y)$.
- We seek $f(X)$ to predict $Y$ given $X$. It requires a loss function to be minimized.
- The most common loss function is square loss $L(a, b) = (a - b)^2$:

$$L(Y, f(X)) = (Y - f(X))^2 \tag{1}$$

- The expected prediction error is

$$E(Y - f(X))^2 = E_X \left[ E_{Y|X} \left( [Y - f(X)]^2 | X \right) \right] \tag{2}$$

- We can minimize the expected predition error pointwise:

$$f(x) = \text{argmin}_\mu E_{Y|X}\left([Y-\mu]^2|X=x\right) = E(Y|X=x) \tag{3}$$

The conditional expectation is called the regression function.

- Two (nonparametric and parametric) approximation methods:
  - A knn regression method approximates the conditional expectation by a locally constant function:

$$\hat{f}_{knn}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i \tag{4}$$

  - A linear regression approximates the conditional expectation by a linear function $f(x) \approx x^T\beta$ and estimates the finite number of parameters $\beta$. The least squares solution for $\beta$ is

$$\beta = \text{argmin}_\beta E\left(Y - X^T\beta\right)^2 = \left[E(XX^T)\right]^{-1} E(XY) \tag{5}$$

- We may use another loss function, $L_1$ loss or absolute error loss $L_1(a,b) = |a-b|$:

$$L_1(Y, f(X)) = |Y - f(X)| \tag{6}$$

The solution is the conditional median of $Y$ given $X$:

$$f(x) = \text{argmin}_\mu E_{Y|X}\left(|Y-\mu||X=x\right) = median(Y|X=x) \tag{7}$$

- For classification problem, it is common to use a zero-one loss function $L(a,b) = I(a \neq b)$. Suppose $Y = C_1, \cdots, C_K$, one of $K$ possible classes.

$$E\left[L(Y, f(X))\right] = E_X\left[\sum_{k=1}^{K} L(C_k, f(X))P(C_k|X)\right] \tag{8}$$

By minimizing the expected prediction error pointwise,

$$\min_{g\in\{C_1,\cdots,C_K\}} \sum_{k=1}^{K} L(C_k,g)P(C_k|X) = \min_{g\in\{C_1,\cdots,C_K\}} \left[\sum_{k=1}^{K} I(C_k \neq g)P(C_k|X)\right] = \min_{g\in\{C_1,\cdots,C_K\}} [1-P(g|X)] \tag{9}$$

$$f(x) = \text{argmin}_{g\in\{C_1,\cdots,C_K\}} [1-P(g|X=x)] = \text{argmax}_g P(g|X=x) \tag{10}$$

that is called the Bayes classifier.

## 2. Regression

- Suppose that we observe a quantitative response $Y$ and $p$ different predictors, $X_1, X_2, \cdots, X_p$.
- We assume that there is some relationship between $Y$ and $\mathbf{X} = (X_1, X_2, \cdots, X_p)'$:

$$Y = f(\mathbf{X}) + \epsilon \tag{11}$$

  where $f$ is a fixed and unknown function.

- **Systematic information:** $f$
- **Random error term:** $\epsilon$

### 2.1. Systematic information and random error terms in a regression model.

Why estimate $f$? Prediction or/and inference

- Prediction: In many situations, a set of inputs $\mathbf{X}$ are readily available, but the output $Y$ cannot be easily obtained.
  - Let $\widehat{f}$ be an estimate for $f$.
  - We predict $Y$ by using
$$\widehat{Y} = \widehat{f}(\mathbf{X}) \tag{12}$$
  - When the purpose is prediction only, $\widehat{f}$ is treated as a blackbox, that is, we are not concerned with the exact form of $\widehat{f}$.
  - (Accuracy of prediction) For fixed $\widehat{f}$ and $\mathbf{X}$,
$$E\left[(Y - \widehat{Y})^2\right] = E\left[(f(\mathbf{X}) + \epsilon - \widehat{f}(\mathbf{X}))^2\right] = \underbrace{\left(f(\mathbf{X}) - \widehat{f}(\mathbf{X})\right)^2}_{\text{Reducible error}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible error}} \tag{13}$$
  - We want to learn statistical techniques for estimating $f$ with the aim of minimizing the reducible error.
- Inference: We want to understand the relationship between $\mathbf{X}$ and $Y$ , or more specifically, to understand how $Y$ changes as a function of $X_1, \cdots, X_p$.
  - Which predictors are associated with the response?

– What is the relationship between the response and each predictor?

– Can the relationship between Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?

2.2. **Purpose of estimating $f$.**

2.3. **Methods of estimating $f$: How to estimate $f$?** There are parametric and nonparametric methods.

2.3.1. *Parametric methods.*

- Make an assumption about a parametric functional form, or shape, of $f$ that includes parameters $\beta_0, \cdots, \beta_p$.
- Fit or train the model, that is, estimate the parameters $\beta_0, \cdots, \beta_p$.
- Using a parametric method, estimating $f$ is reduced to estimating the parameters $\beta_0, \cdots, \beta_p$.
- Potential disadvantage:
  (1) The chosen model may be far from the true model $\Rightarrow$ poor conclusions (prediction or/and interpretation)
  (2) Adding more flexibility (many parameters) $\Rightarrow$ Overfit the data.
- (Example) Linear regression model, GAM, etc.

2.3.2. *Example.* (Linear regression: polynomial regression)

```r
library(ISLR)
fit <- lm(mpg ~ horsepower, data = Auto)
summary(fit)
```

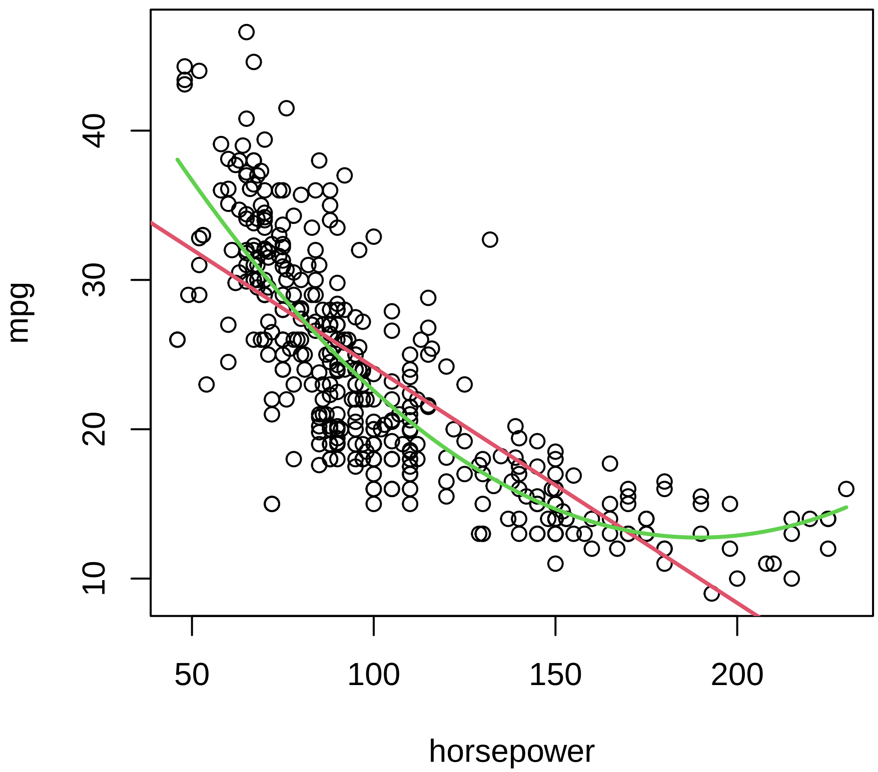
```

Call:
lm(formula = mpg ~ horsepower, data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max
-13.5710  -3.2592  -0.3435   2.7630  16.9240

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
fit2 <- lm(mpg ~ poly(horsepower, 2, raw = T), data = Auto)
summary(fit2)
```

```
Call:
lm(formula = mpg ~ poly(horsepower, 2, raw = T), data = Auto)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.7135  -2.5943  -0.0859   2.2868  15.8961 

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   56.9000997  1.8004268   31.60   <2e-16 ***
poly(horsepower, 2, raw = T)1 -0.4661896  0.0311246  -14.98   <2e-16 ***
poly(horsepower, 2, raw = T)2  0.0012305  0.0001221   10.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.374 on 389 degrees of freedom
Multiple R-squared:  0.6876,	Adjusted R-squared:  0.686 
F-statistic:   428 on 2 and 389 DF,  p-value: < 2.2e-16
```

```r
with(Auto, plot(horsepower, mpg))
abline(fit, col = 2, lwd = 2)
curve(coef(fit2)[1] + coef(fit2)[2] * x + coef(fit2)[3] * x^2, add = T,
    col = 3, lwd = 2)
```

FIGURE 1. Fitted curves of polynomial regression models: (Red) linear, and (Green) quadratic



### 2.3.3. *Nonparametric methods.*

- We do not make explicit assumptions about the functional form of $f$.
- Seek an estimate of $f$ that gets as close to the data points as possible without being too rough or wiggly.
- A very large number of observations (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate for $f$.
- (Example) kNN regression, Random Forest, etc.

#### 2.3.3.1. Example. (kNN regression)

```
library(ISLR)
library(caret)
fit <- knnreg(data.frame(horsepower = Auto$horsepower), Auto$mpg, k = 10)
xt <- seq(46, 230, by = 0.001)
yhat <- predict(fit, data.frame(horsepower = xt))
```

```
plot(xt, yhat, type = "l", col = "red", lwd = 2)
with(Auto, points(horsepower, mpg))
```

Figure 2. knn regression with k=10

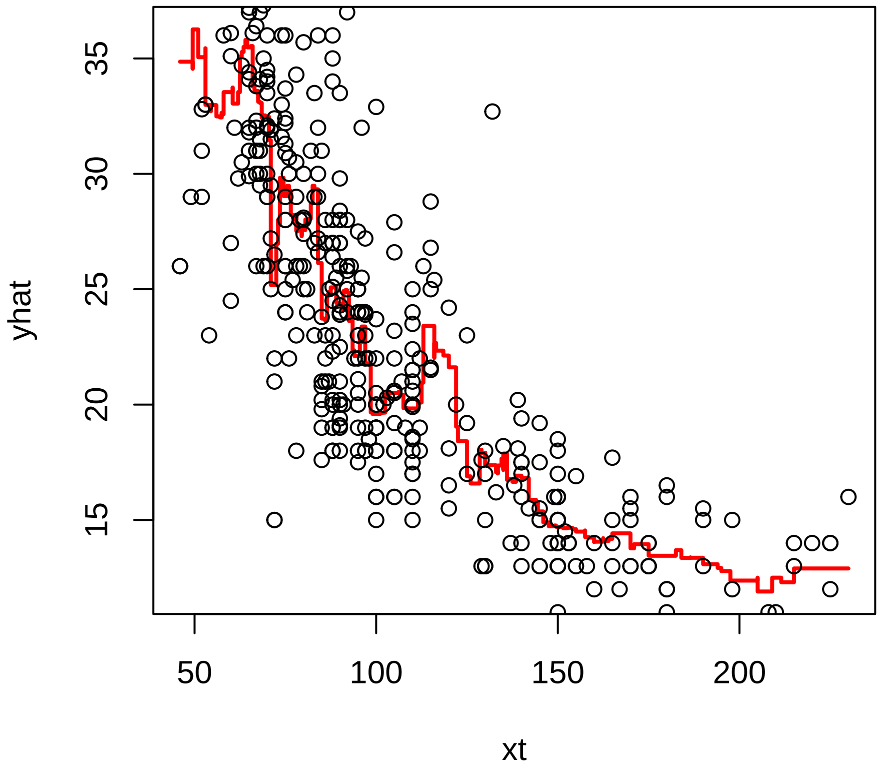
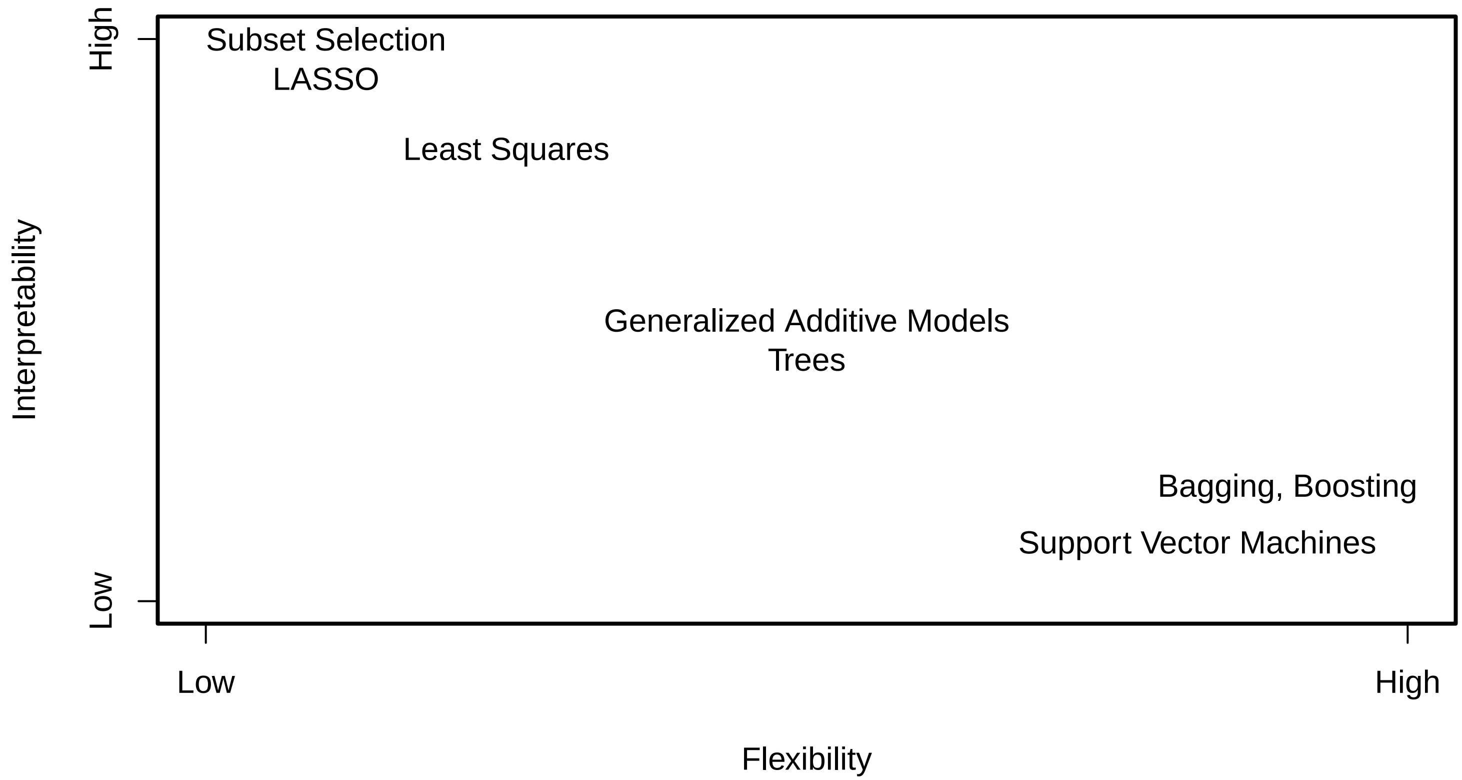


Figure 3. A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.




Why would we ever choose to use a more restrictive method instead of a very flexible approach?

- If we are mainly interested in inference, then restrictive models are much more interpretable.

- Even for prediction, highly flexible methods may suffer from overfitting and we often obtain more accurate predictions using a less flexible method.
- **The law of parsimony** tells us that when there are alternative explanations of events, the simplest one is likely to be correct.

### 2.4. The Trade-Off Between Prediction Accuracy and Model Interpretability.

**Measuring the Quality of Fit**

- A commonly used measure is the mean square error (MSE):

$$MSE(\widehat{f}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{f}(\mathbf{x}_i)\right)^2 \tag{14}$$

- **Training MSE and test MSE**: when we have training data $\{(y_1, \mathbf{x}_1), \cdots, (y_n, \mathbf{x}_n)\}$ and independent observation $(y_0, \mathbf{x}_0)$ that is not used for estimating $f$,

$$(training) \quad MSE(\widehat{f}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{f}(\mathbf{x}_i)\right)^2 \tag{15}$$

$$(test) \quad MSE(\widehat{f}) = Ave\left(y_0 - \widehat{f}(\mathbf{x}_0)\right)^2 \tag{16}$$

- A more flexible model tends to have a smaller training MSE than a simpler model. As model flexibility increases, training MSE will decrease, but test MSE may not.

- When a given method yields a small training MSE but a large test MSE, we are said to be overfitting the data.
- To assess a model accuracy, we need to calculate the test MSE rather than the training MSE.
- How can we go about trying to select a method that minimizes the test MSE?
- There are a variety of approaches to estimate the test MSE. One of important methods is cross-validation.

2.5. **Assessing Model Accuracy.**

2.5.1. *Example.* 100 observations are selected from $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$ where $\beta_0 = 1, \beta_1 = 2, \beta_2 = 3$, and $\epsilon \sim N(0, 1)$. Four predictors $x_1, \cdots, x_4$ are independently generated from a uniform distribution on (0,1).

FIGURE 4. Traing MSE (Black) and Test MSE (Red). The traing MSE decreases as the number of parameters increases while the test MSE is U-shaped and it has a minimum at 3 parameter model.

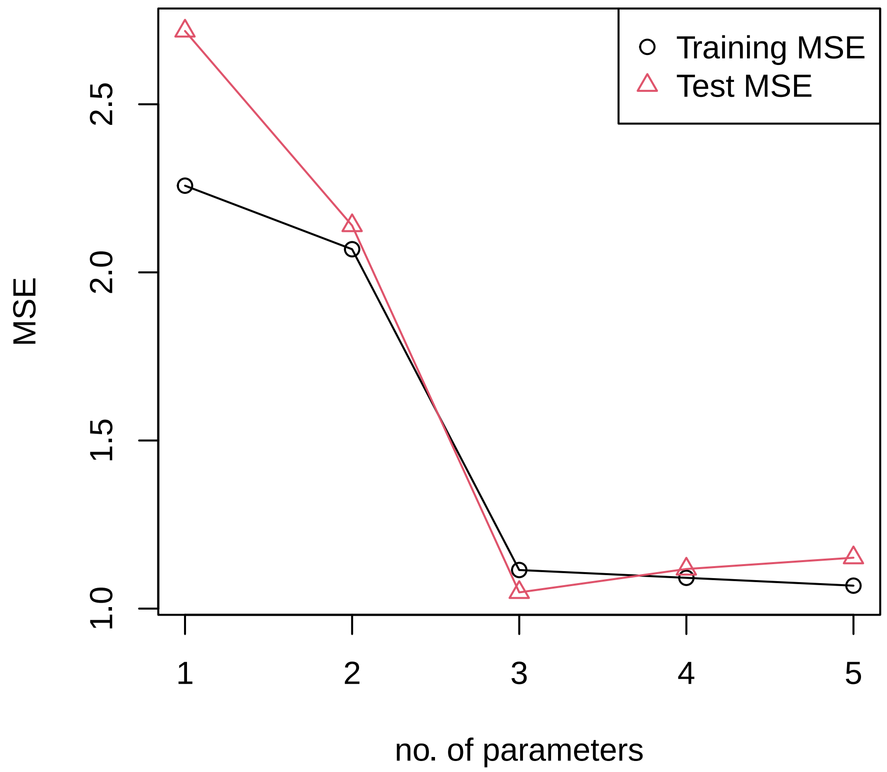


FIGURE 5. Left: Data simulated from $f$, shown in gray. Three estimates of $f$ are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and purple curves). Right: Training MSE (black curve), and test MSE (red curve). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

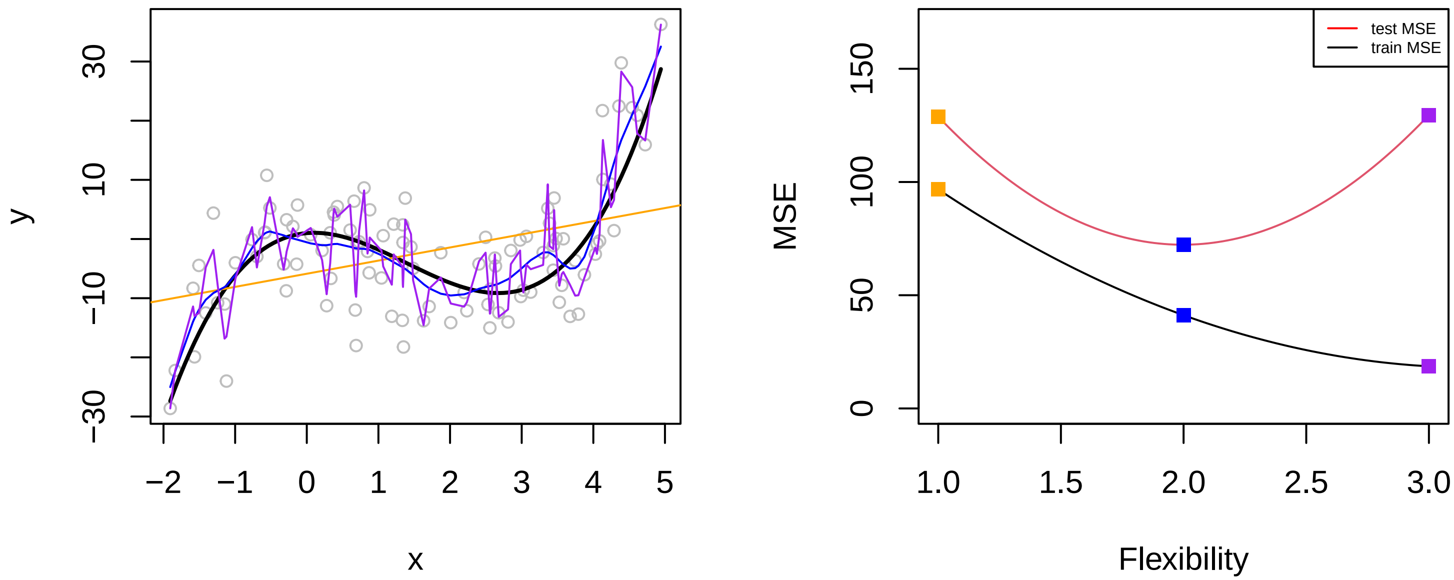
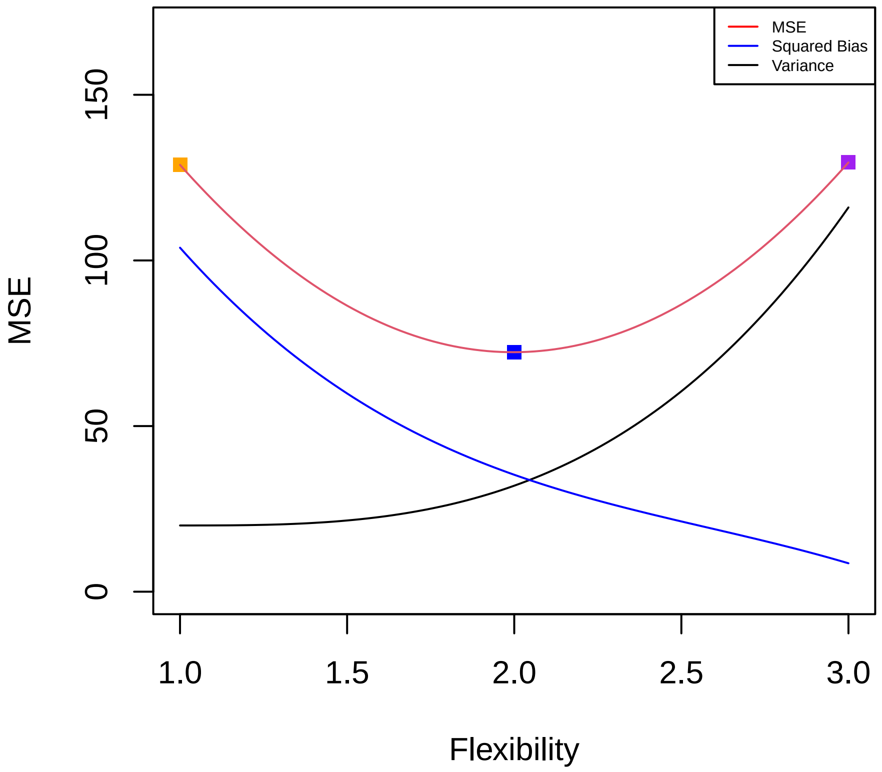


FIGURE 6. Bias-variance trade-off of test MSE. Bias tends to decrease as a model becomes more complex (Blue). Variance tends to increase as a model becomes more complex (Black).




2.6. **Model selection and bias-variance trade-off.**

- The U-shape observed in the test MSE curves turns out to be the result of two competing properties of statistical learning methods.
- The expected test MSE for a given $x_0$ is

$$E(y_0 - \widehat{f}(\mathbf{x}_0))^2 = \text{Var}(\widehat{f}(\mathbf{x}_0)) + Bias(\widehat{f}(\mathbf{x}_0))^2 + \text{Var}(\epsilon) \tag{17}$$

  that refers to the average test MSE obtained by repeatedly estimating $f$ using a large number of training sets and tested each at $x_0$.
- Variance refers to the amount by which $\widehat{f}$ would change if we estimated it using a different training data set. In general, more flexible statistical methods have higher variance.
- To minimize the expected test error, we need to find a statistical learning method having simultaneously low variance and low bias.

- Most of models have complexity parameters to be determined.

- We cannot use residual sum-of squares on the training data to determine the complexity parameters since we would always pick those gave interpolating fits and hence zero residuals. Such a model is unlikely to predict future data well at all.

We define the expected prediction error at $x_0$, test or generalization error, as

$$EPE(x_0) = E_{Y_0|X_0}\left[E_{\mathcal{T}} L(Y_0, \widehat{f}(X_0))|X_0 = x_0\right] \tag{18}$$

If $L(a,b) = (a-b)^2$, then

$$\begin{aligned}
EPE(x_0) =& E_{Y_0|X_0}\left[E_{\mathcal{T}}(Y_0 - \widehat{f}(x_0))^2|X_0 = x_0\right] && (19)\\
=& E_{Y_0|X_0}\left[(Y_0 - f(x_0))^2 + 2(Y_0 - f(x_0))E_{\mathcal{T}}(f(x_0) - \widehat{f}(x_0)) + E_{\mathcal{T}}(f(x_0) - \widehat{f}(x_0))^2|X_0 = x_0\right] && (20)\\
=& E_{Y_0|X_0}\left[(Y_0 - f(x_0))^2|X_0 = x_0\right] + 2\,\underbrace{E_{Y_0|X_0}\left[Y_0 - f(x_0)|X_0 = x_0\right]}_{=0} E_{\mathcal{T}}(f(x_0) - \widehat{f}(x_0)) && (21)\\
& + \underbrace{E_{\mathcal{T}}(f(x_0) - \widehat{f}(x_0))^2}_{=MSE_{\mathcal{T}}(\widehat{f}(x_0))} && (22)\\
=& \mathrm{Var}(Y_0|X_0 = x_0) + \mathrm{Var}_{\mathcal{T}}(\widehat{f}(x_0)) + \mathrm{Bias}^2_{\mathcal{T}}(\widehat{f}(x_0)) && (23)
\end{aligned}$$