Advanced Data Mining

Kamiar Rahnama Rad

lecture 1

chapter 1

- statistical learning:
  - supervised: the presence of the outcome variable guides the learning process, such as regression and classification.
  - unsupervised: we observe only the features and have no measurements of the outcome, such as clustering and dimensionality reduction.

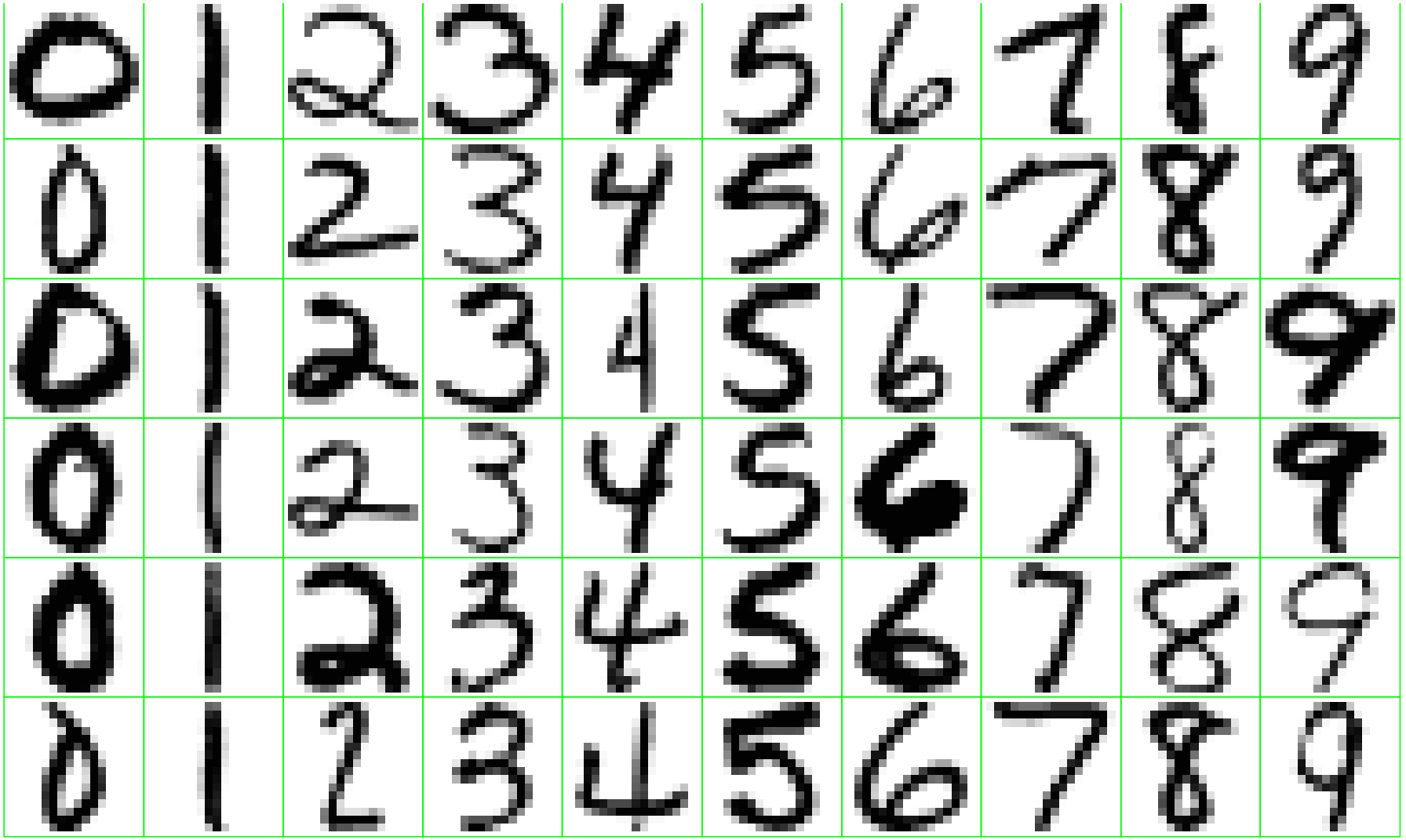


FIGURE 1.2. *Examples of handwritten digits from U.S. postal envelopes.*

- supervised learning/classification: identify the numbers in a handwritten zip code, from a digitized image.
- each image is a segment from a five digit zip code.
- images are 16×16 eight-bit greyscale maps.
- each pixel ranging in intensity from 0 to 255.
- predict, from the 16×16 matrix of pixel intensities, the identity of each image $(0, \ldots, 9)$.

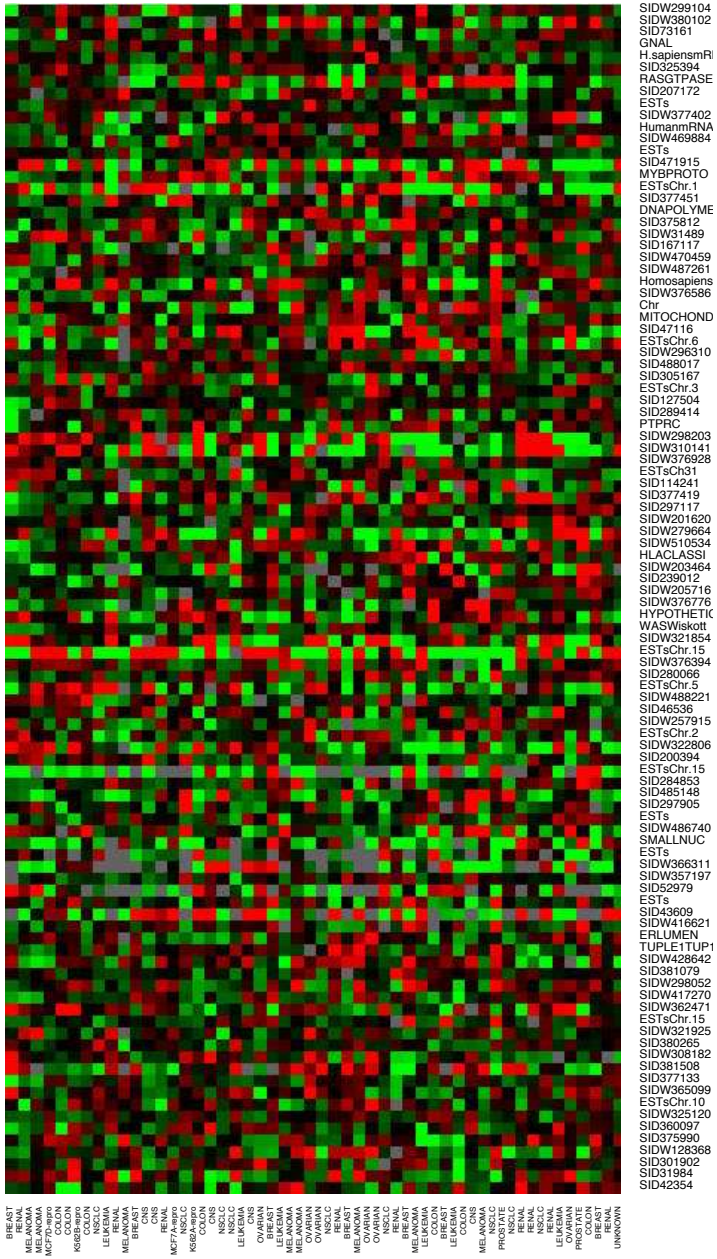


FIGURE 1.3. *DNA microarray data: expression matrix of* 6830 *genes (rows) and* 64 *samples (columns), for the human tumor data. Only a random sample of* 100 *rows are shown. The display is a heat map, ranging from bright green (negative, under expressed) to bright red (positive, over expressed). Missing values are gray. The rows and columns are displayed in a randomly chosen order.*

- 6830 genes (rows).
- 64 samples (columns) corresponding to cancer tumors from different patients.
- which samples are most similar to each other, in terms of their expression profile across genes? think of samples as points in 6830 dimensional space, which we want to cluster in some way.
- which genes are most similar to each other, in terms of their expression profiles across samples?

TABLE 1.1. *Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between* `spam` *and* `email`.

	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

- supervised learning/classification: predict wether an email is spam based on the words and punctuation marks in the email message.
- not all errors are equal: we want to avoid filtering out good email, while letting spam get through is not desirable but less serious in its consequences.

chapter 2

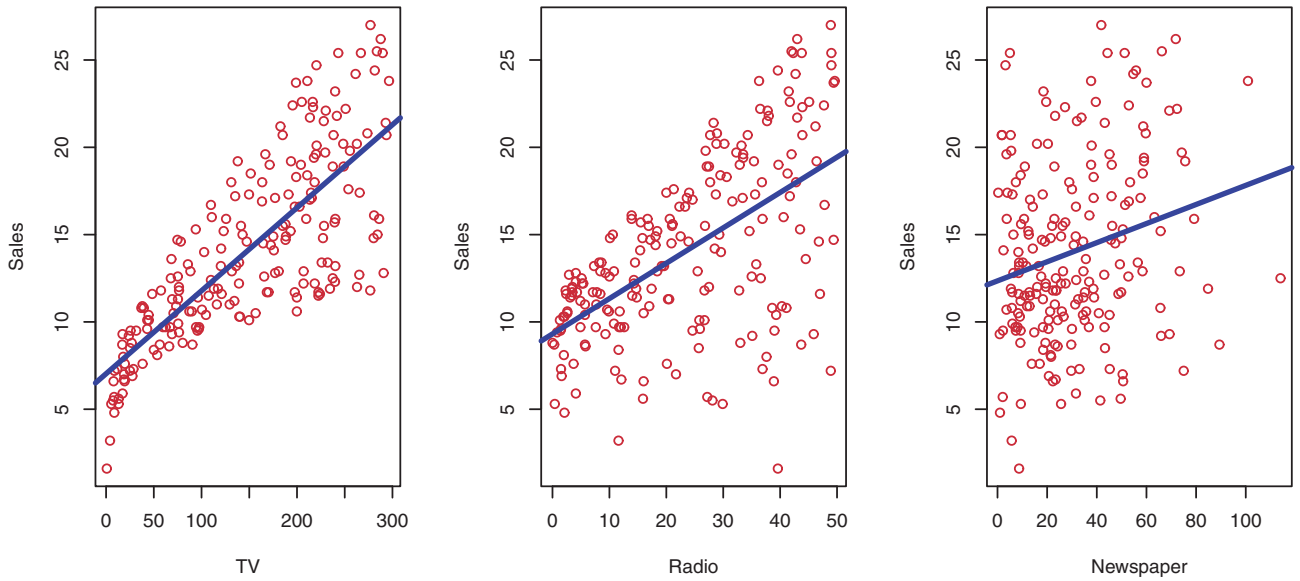


FIGURE 2.1. *The* `Advertising` *data set. The plot displays* `sales`*, in thousands of units, as a function of* `TV`*,* `radio`*, and* `newspaper` *budgets, in thousands of dollars, for* 200 *different markets. In each plot we show the simple least squares fit of* `sales` *to that variable, as described in Chapter 3. In other words, each blue line represents a simple model that can be used to predict* `sales` *using* `TV`*,* `radio`*, and* `newspaper`*, respectively.*

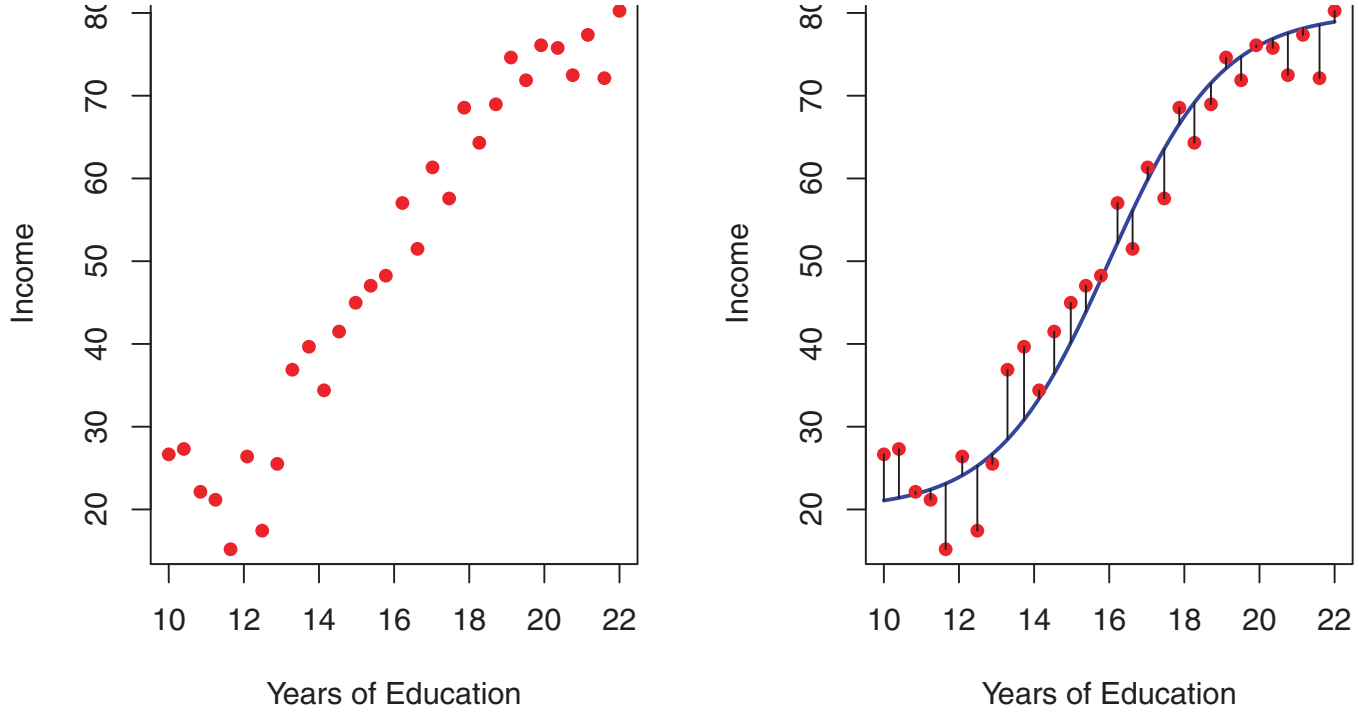


FIGURE 2.2. *The* `Income` *data set.* Left: *The red dots are the observed values of* `income` *(in tens of thousands of dollars) and* `years of education` *for* 30 *individuals.* Right: *The blue curve represents the true underlying relationship between* `income` *and* `years of education`*, which is generally unknown (but is known in this case because the data were simulated). The black lines represent the error associated with each observation. Note that some errors are positive (if an observation lies above the blue curve) and some are negative (if an observation lies below the curve). Overall, these errors have approximately mean zero.*

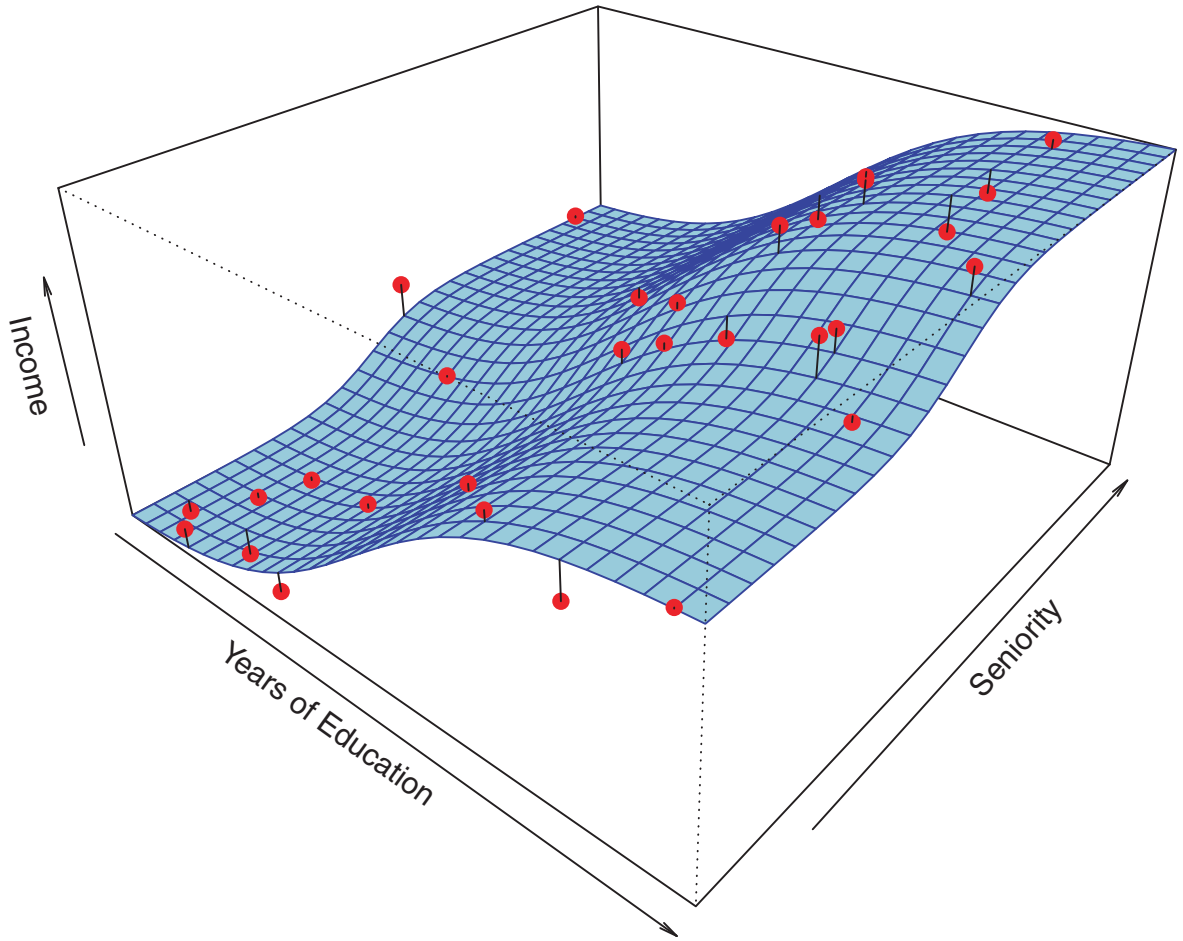


FIGURE 2.3. *The plot displays* income *as a function of* years of education *and* seniority *in the* Income *data set. The blue surface represents the true underlying relationship between* income *and* years of education *and* seniority*, which is known since the data are simulated. The red dots indicate the observed values of these quantities for* 30 *individuals.*

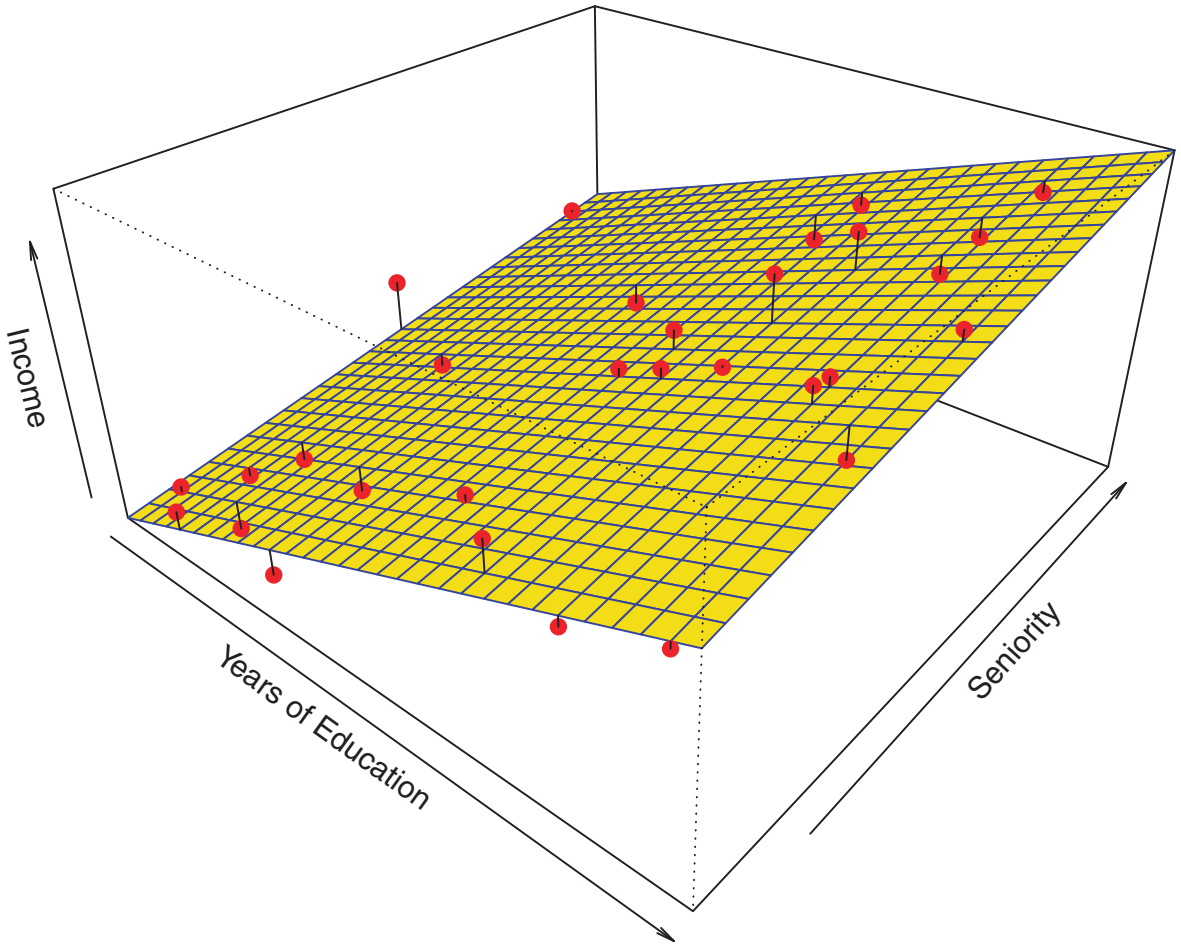


FIGURE 2.4. *A linear model fit by least squares to the* `Income` *data from Figure 2.3. The observations are shown in red, and the yellow plane indicates the least squares fit to the data.*

- there is some relationship between the p features (predictors, independent variables, input variables) $X = (X_1, \ldots, X_p)$ and the response (dependent variable, output) Y

$$Y = f(X) + \epsilon.$$

- ϵ is a random error term, independent of X, and has zero mean.
- $f(.)$ represents the systematic information that X provides about Y.
- $f(.)$ is unknown but we want to learn/estimate it from data.

- Why estimate f?
  - prediction: $\hat{Y} = \hat{f}(X)$.
  - accuracy of $\hat{Y}$ as a prediction for Y depends on:
    * reducible error: $\hat{f}$ is not a perfect estimate of f due to limited amount of data, and an inaccurate statistical model, e.g. linear regression when the relationship is nonlinear.
    * irreducible error: even if we have a perfect estimate, that is $\hat{Y} = f(X)$, our prediction will still have some error because Y is also a function of ϵ which cannot be predicted from X.

- $E_Y(Y - \hat{Y})^2$: average squared difference between the predicted and the actual value of Y:

$$E_Y(Y - \hat{Y})^2 \;\; = \;\; [f(X) - \hat{f}(X)]^2 + var(\epsilon)$$

- reducible error: $[f(X) - \hat{f}(X)]^2$
- irreducible error: $var(\epsilon)$
- take home message: *We must not expect more precision than the subject-matter admits.* The Nicomachean Ethics: Book I.

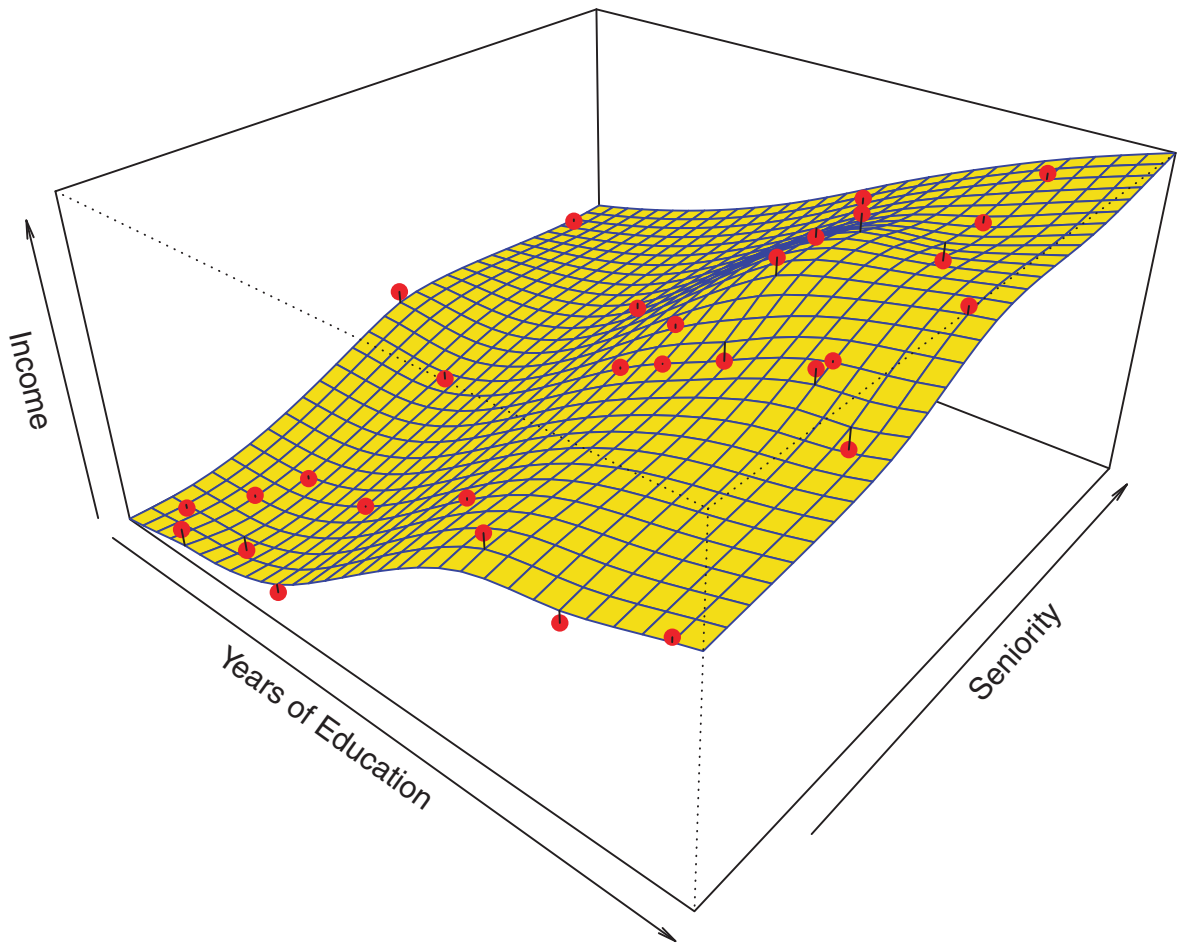


FIGURE 2.5. *A smooth thin-plate spline fit to the* `Income` *data from Figure 2.3 is shown in yellow; the observations are displayed in red. Splines are discussed in Chapter 7.*

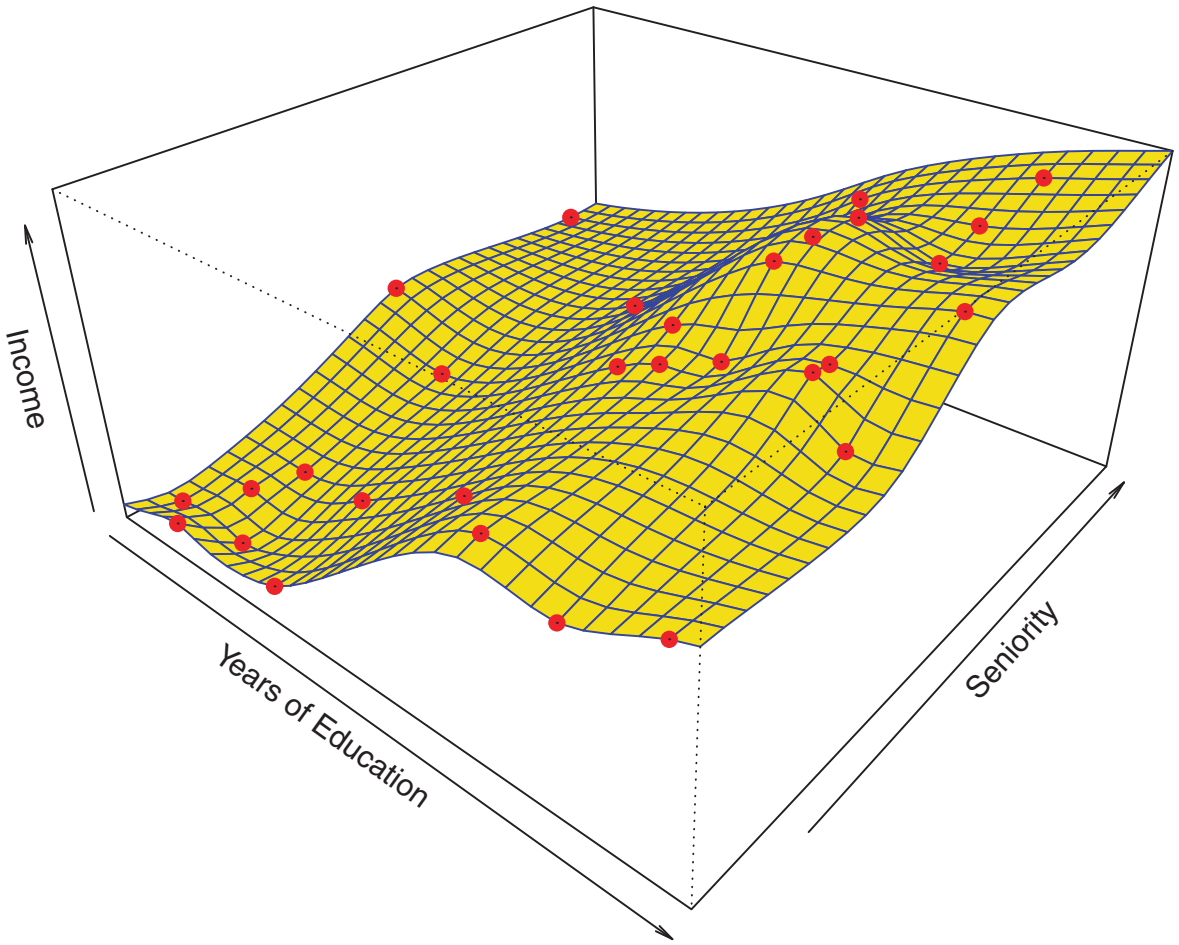


FIGURE 2.6. *A rough thin-plate spline fit to the* `Income` *data from Figure 2.3. This fit makes zero errors on the training data.*

- why estimate f?
  - inference:
    * 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?

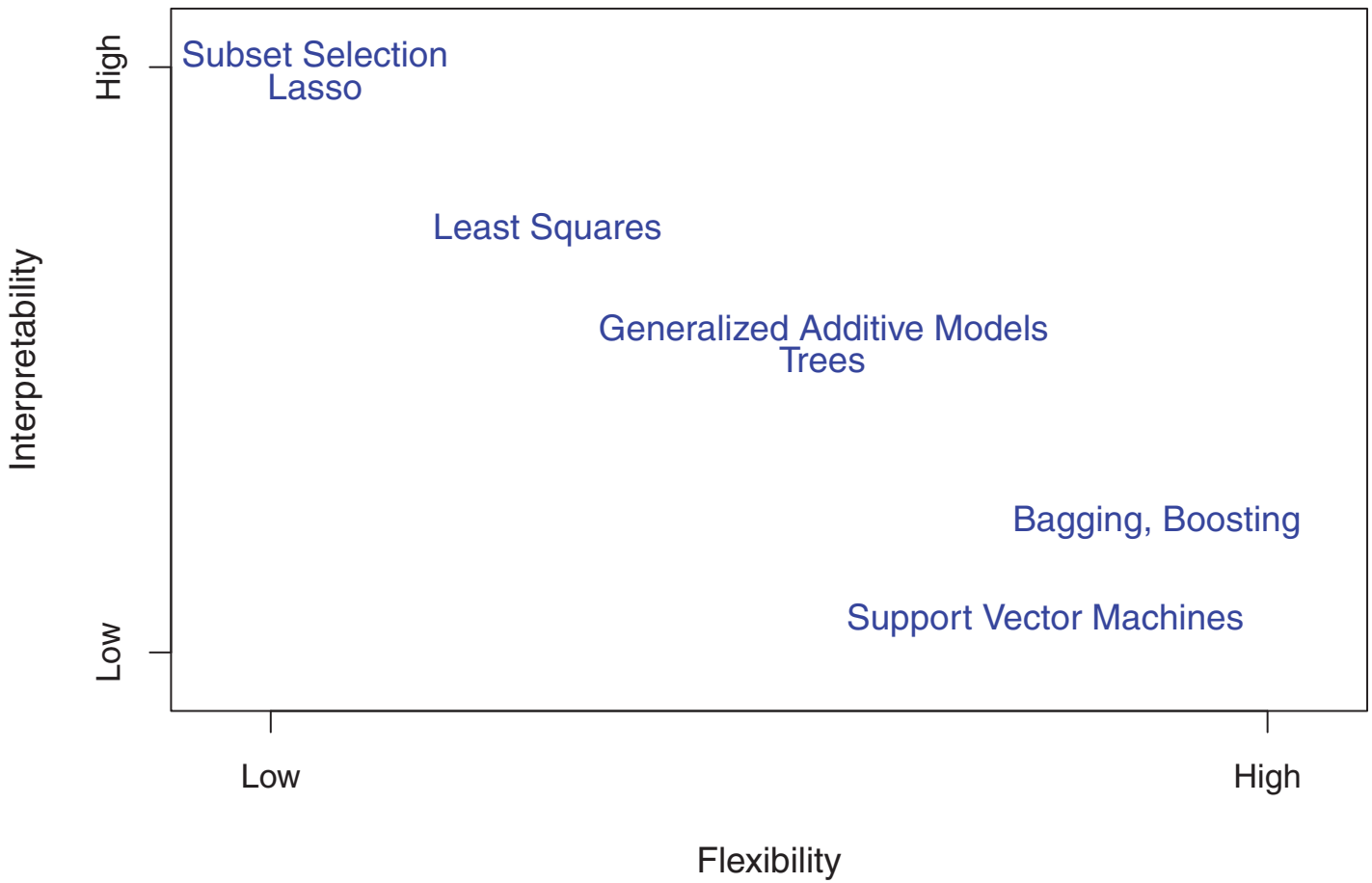


FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

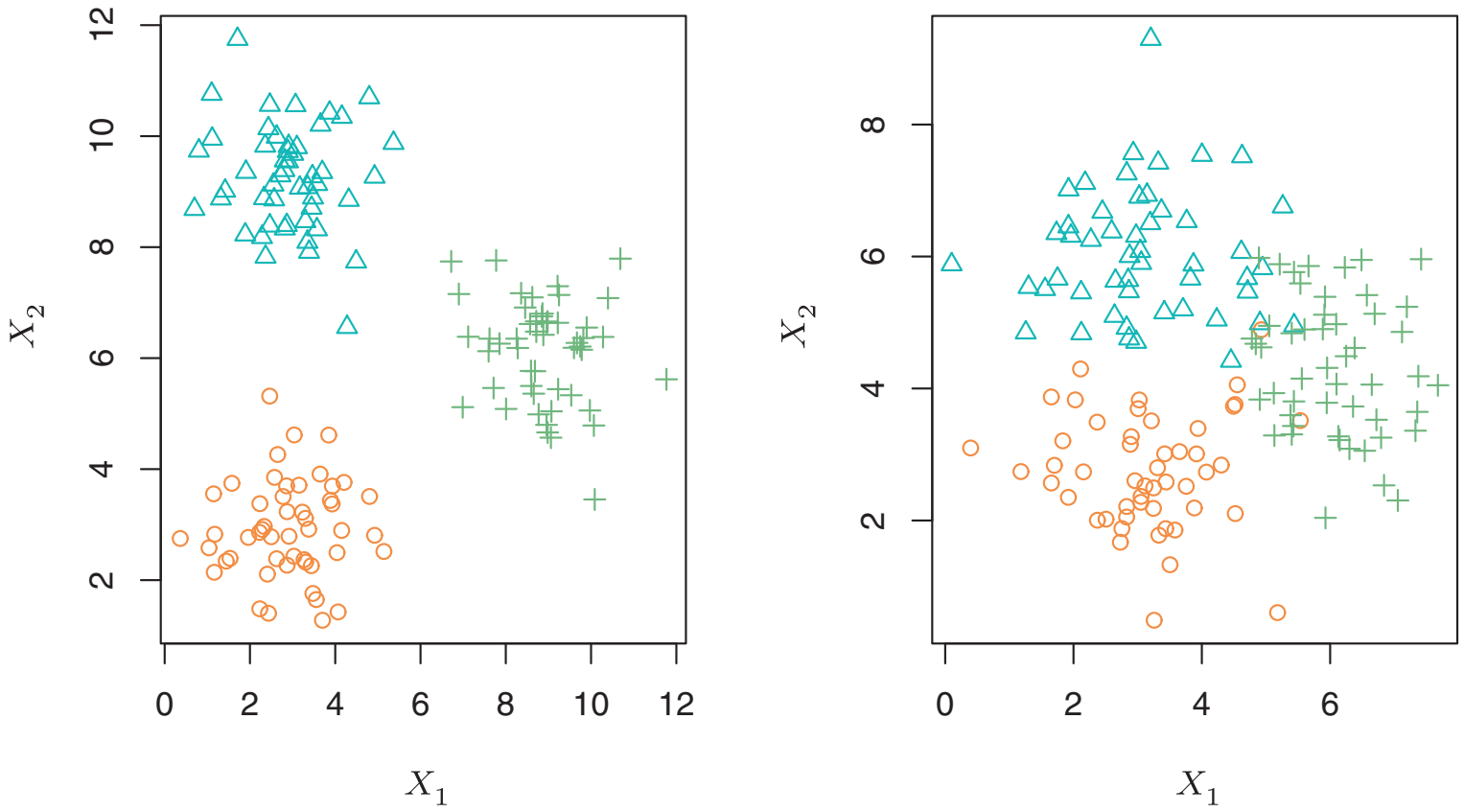


FIGURE 2.8. *A clustering data set involving three groups. Each group is shown using a different colored symbol.* Left: *The three groups are well-separated. In this setting, a clustering approach should successfully identify the three groups.* Right: *There is some overlap among the groups. Now the clustering task is more challenging.*

- training data is $\Big\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\Big\}$ where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$.
- or equivalently in vector and matrix format the training data is X, y where
  - X is a $n \times p$ matrix with $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ as its ith row
  - $y = (y_1, y_2, \ldots, y_n)^T$.
- example of supervised learning: given data X, y find $\hat{f}(.)$ such that $\hat{f}(x_i)$ is close to y_i.

- matrix algebra and notation...
- $y = X\beta + \epsilon$
- residual sum of squares $= \sum_{i=1}^{n}(y_i - x_i^T\hat{\beta})^2 = \|y - x\beta\|_2^2$
- what is n? what is p?

lecture 2

- how to assessing model accuracy? train vs. test.
- training MSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

- accuracy of predictions when $\hat{f}()$ applied to previously unseen data.
- test MSE:

$$\text{Ave}(y_0 - \hat{f}(x_0)),$$

where (x_0, y_0) is a previously unseen test observation not used to train the statistical learning method.

- typically, in practice we use all our data to train/learn/estimate the statistical model...cross validation to estimate test MSE (chapter 5)
- train MSE $\neq$ test MSE
- train MSE vs. degrees of freedom ?
- test MSE vs. degrees of freedom?
- degrees of freedom $\simeq$ a quantity that summarizes the flexibility $\simeq$ a measure for model complexity $\simeq$ number of parameters in the model
- overfitting: a less flexible model would have yielded a smaller test MSE

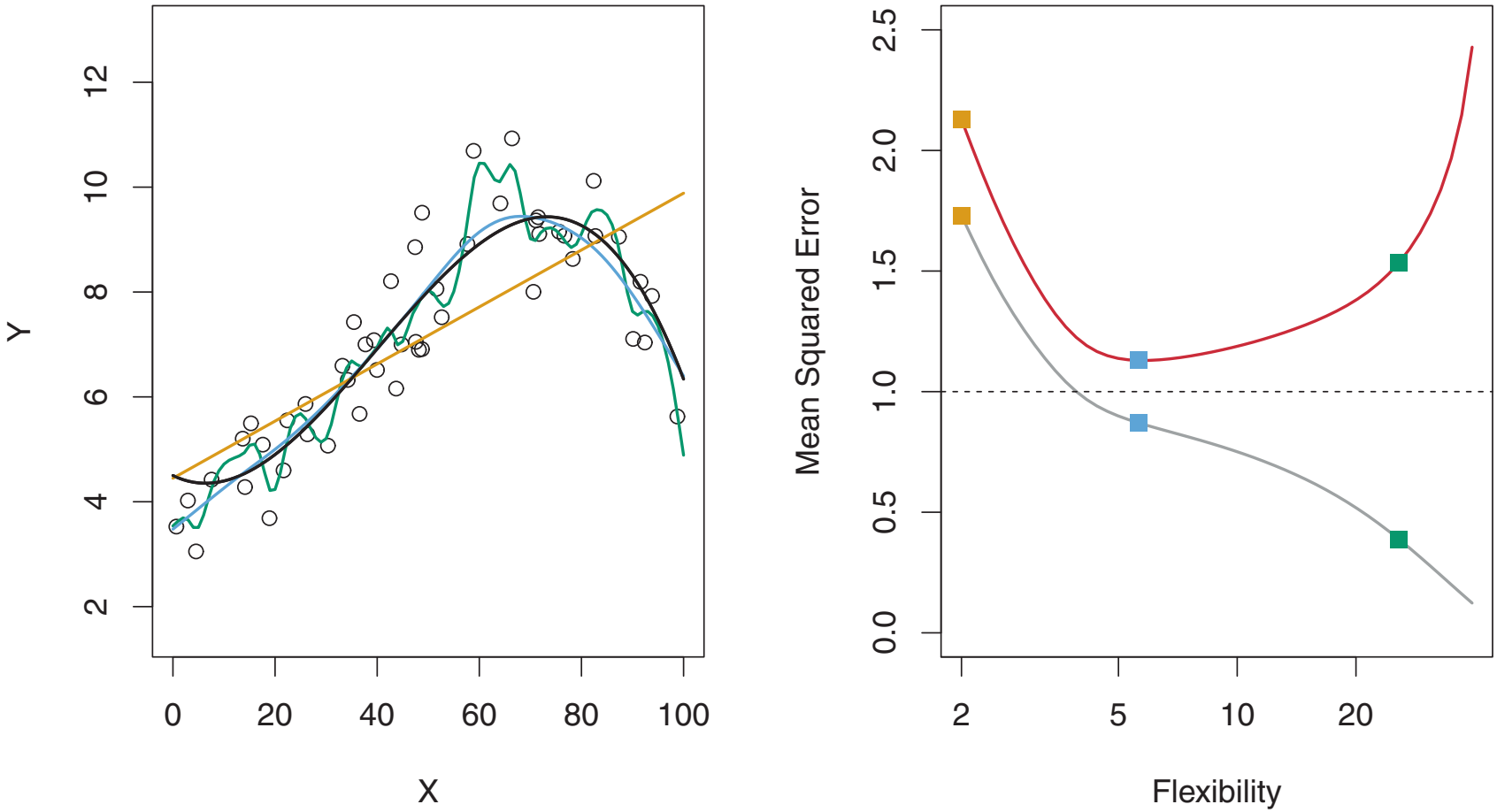


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

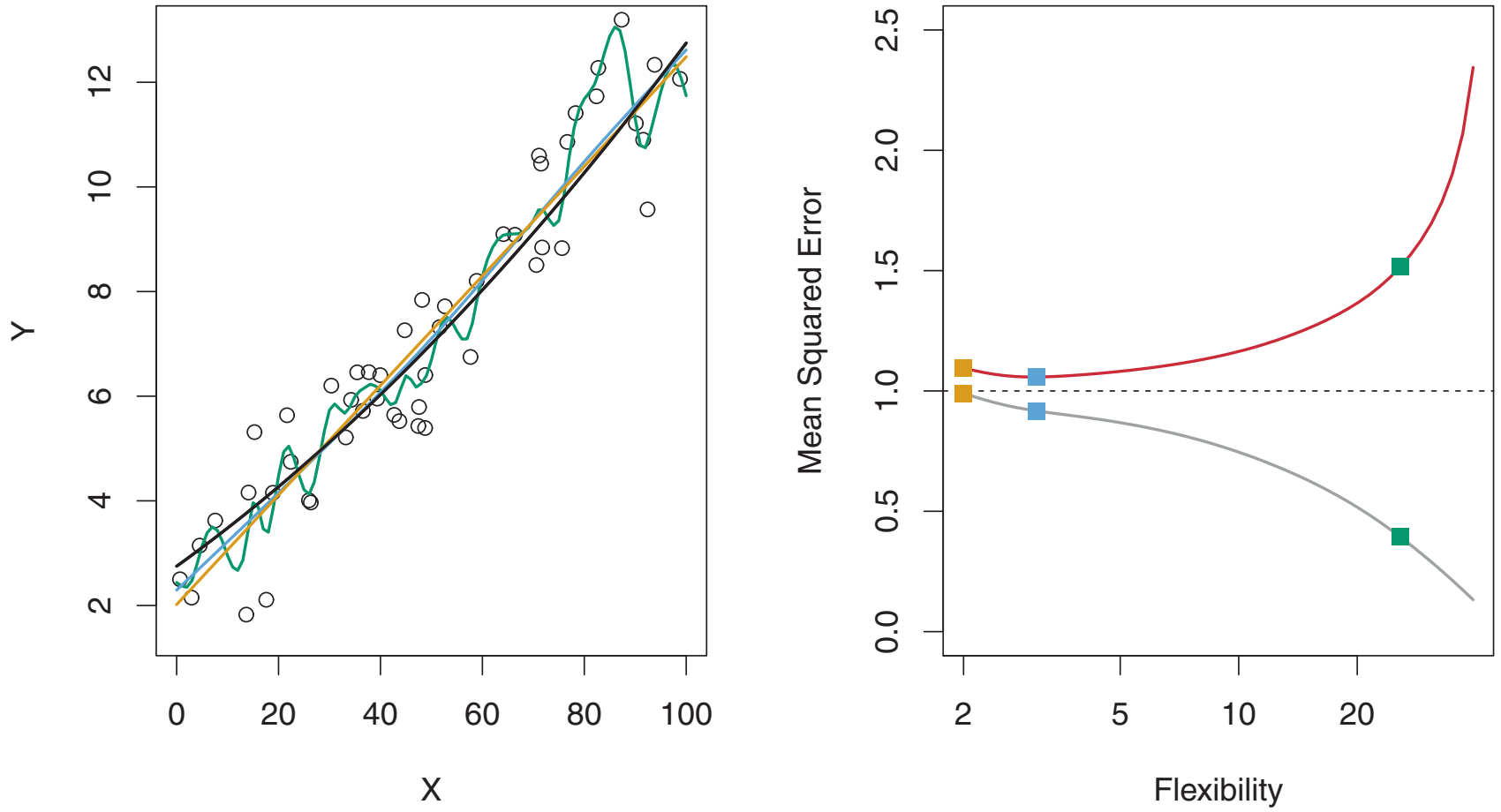


FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

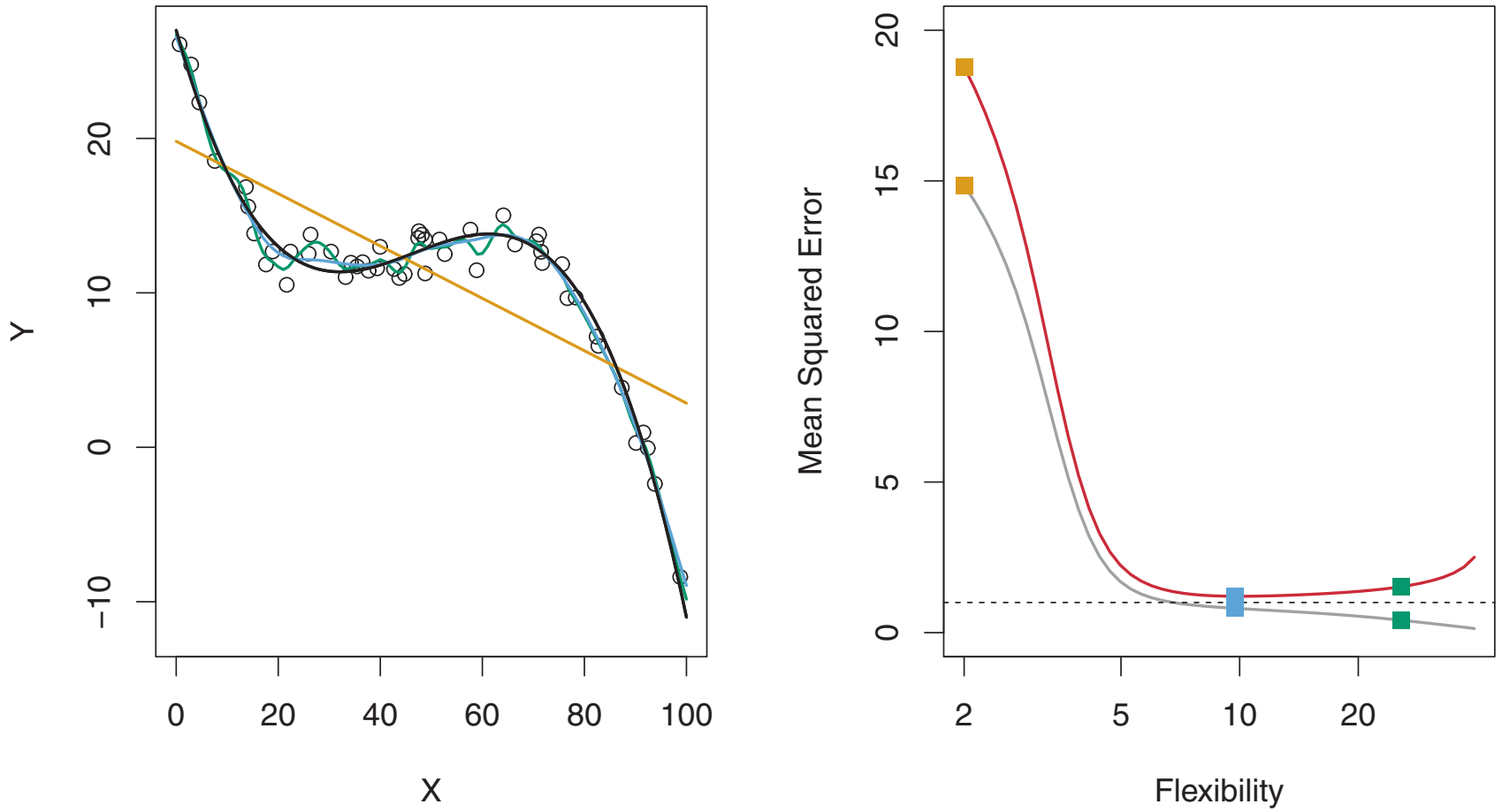


FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

- u-shape of test MSE: bias-variance trade-off
- $E\Big(y_0 - \hat{f}(x_0)\Big)^2 = \text{var}(\hat{f}(x_0)) + [\text{bias}(\hat{f}(x_0))]^2 + \text{var}(\epsilon)$
- expected test error: repeatedly estimate f using a large number of training sets, and test each estimate at x_0
- ideal: low bias and low variance
- variance: the amount by which $\hat{f}$ would change if we estimated using a different training data set.
- high variance: small changes in the training data results in large changes in $\hat{f}$.
- bias: error that is introduced by approximating an extremely complicated real life problem by a much simpler model.

- more flexible methods $\rightarrow$ small bias + high variance.
- low bias + high variance = curve that passes through every single training observaion
- low variance + high bias = fitting a horizontal line to the data

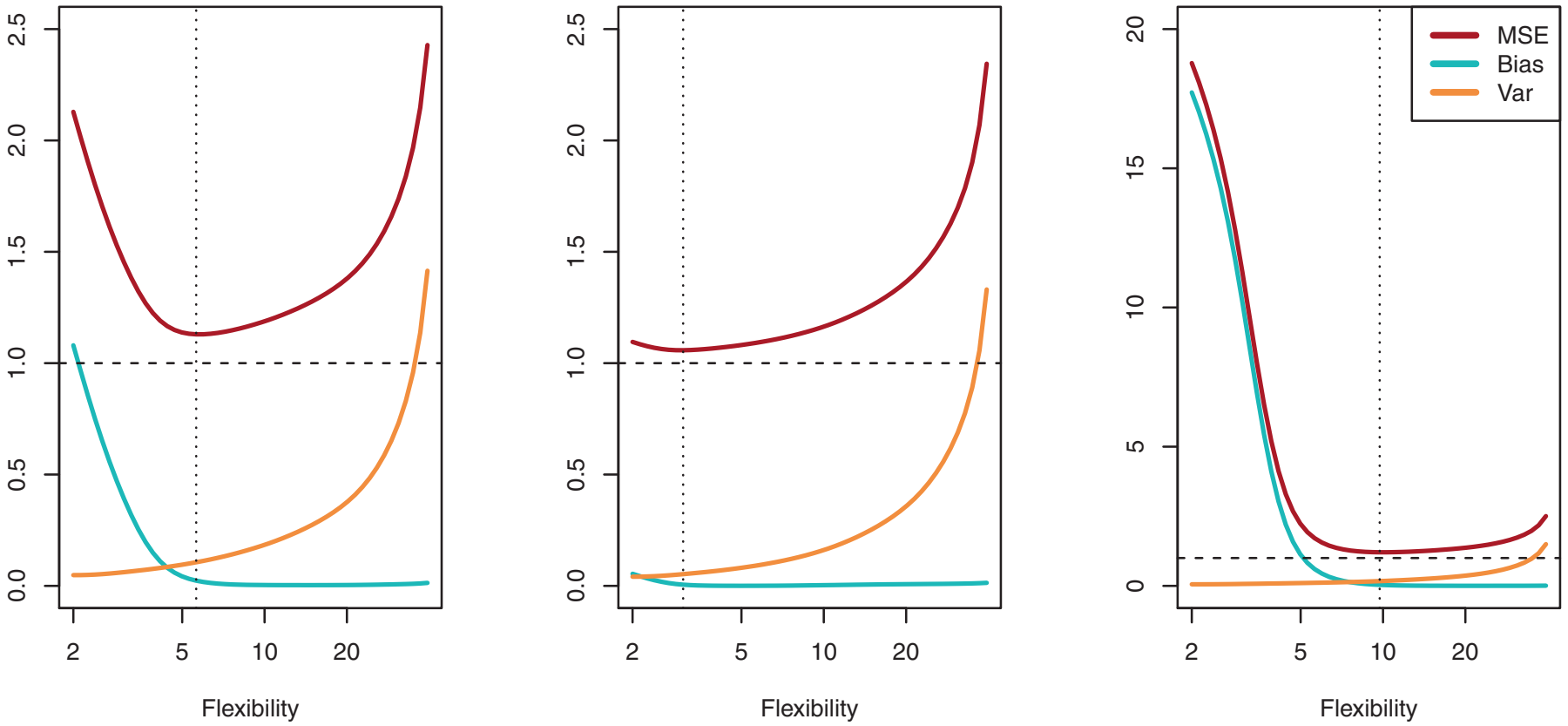


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

- classification: how to assessing model accuracy?
- basic concepts: error rate, indicator variable, training error, test error
- conditional probability: $\Pr(Y = j|X = x_0)$
- bayes classifier=unattainable gold standard $\sim$ irreducible error
- bayes decision boundary
- overall bayes error rate: $1 - E\Big(\max_j \Pr(Y = j|X = x_0)\Big)$

lecture 3

- k-nearest neighbors (knn):
  $\widehat{\Pr}(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$

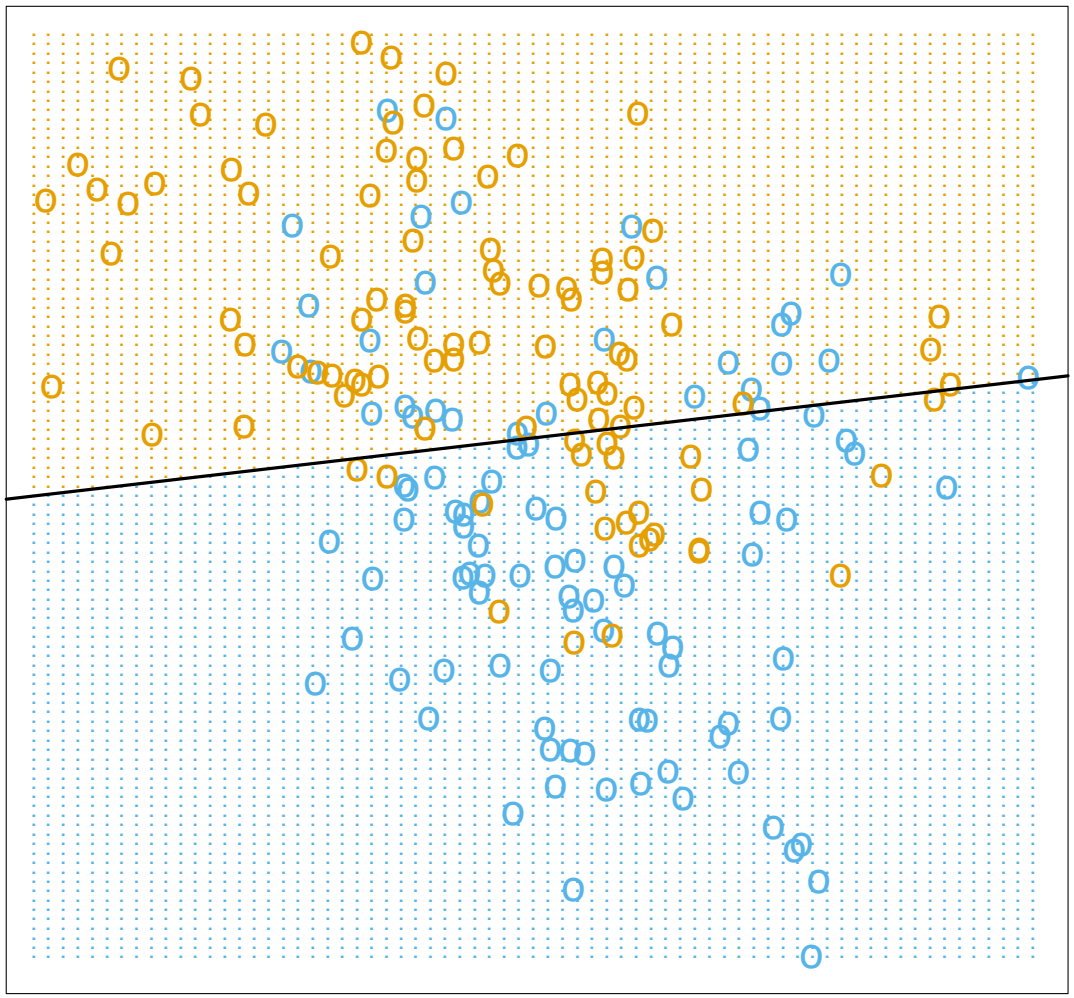


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$*. The orange shaded region denotes that part of input space classified as* ORANGE*, while the blue region is classified as* BLUE*.*

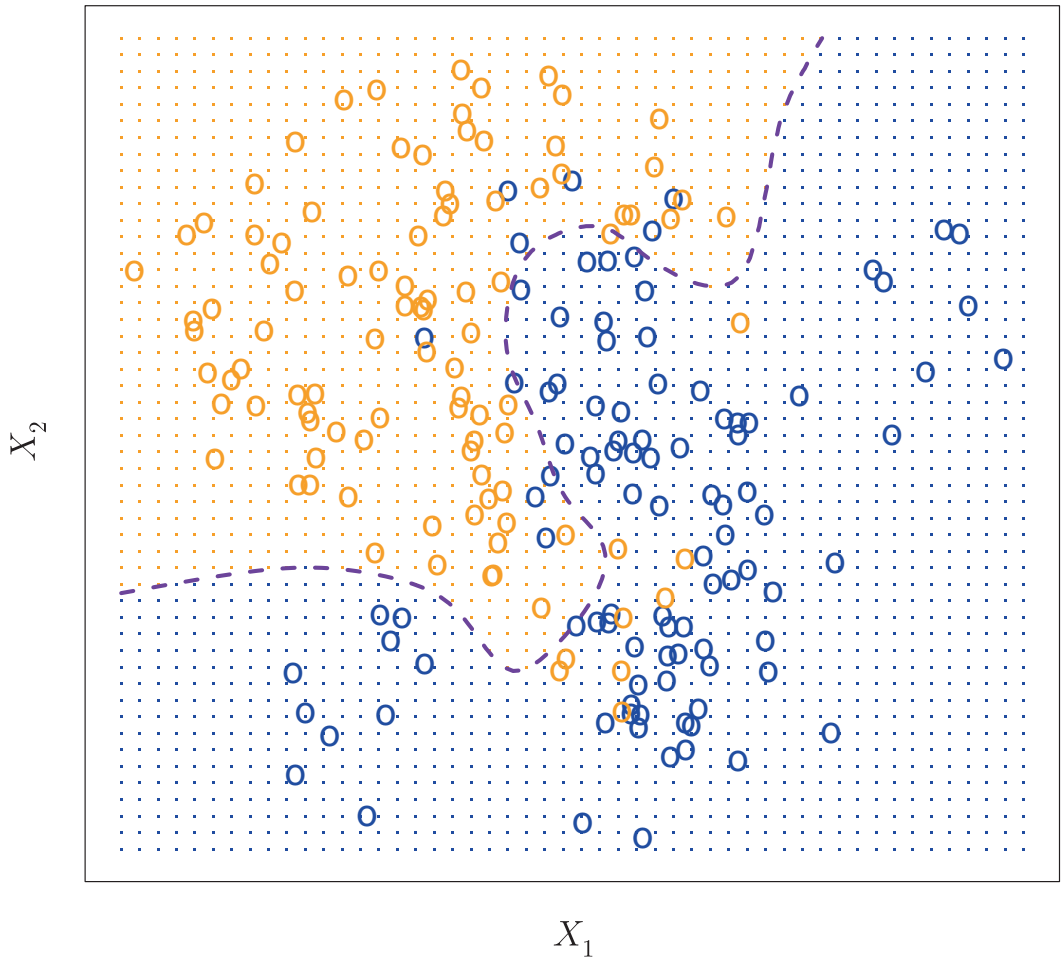


FIGURE 2.13. *A simulated data set consisting of* 100 *observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.*

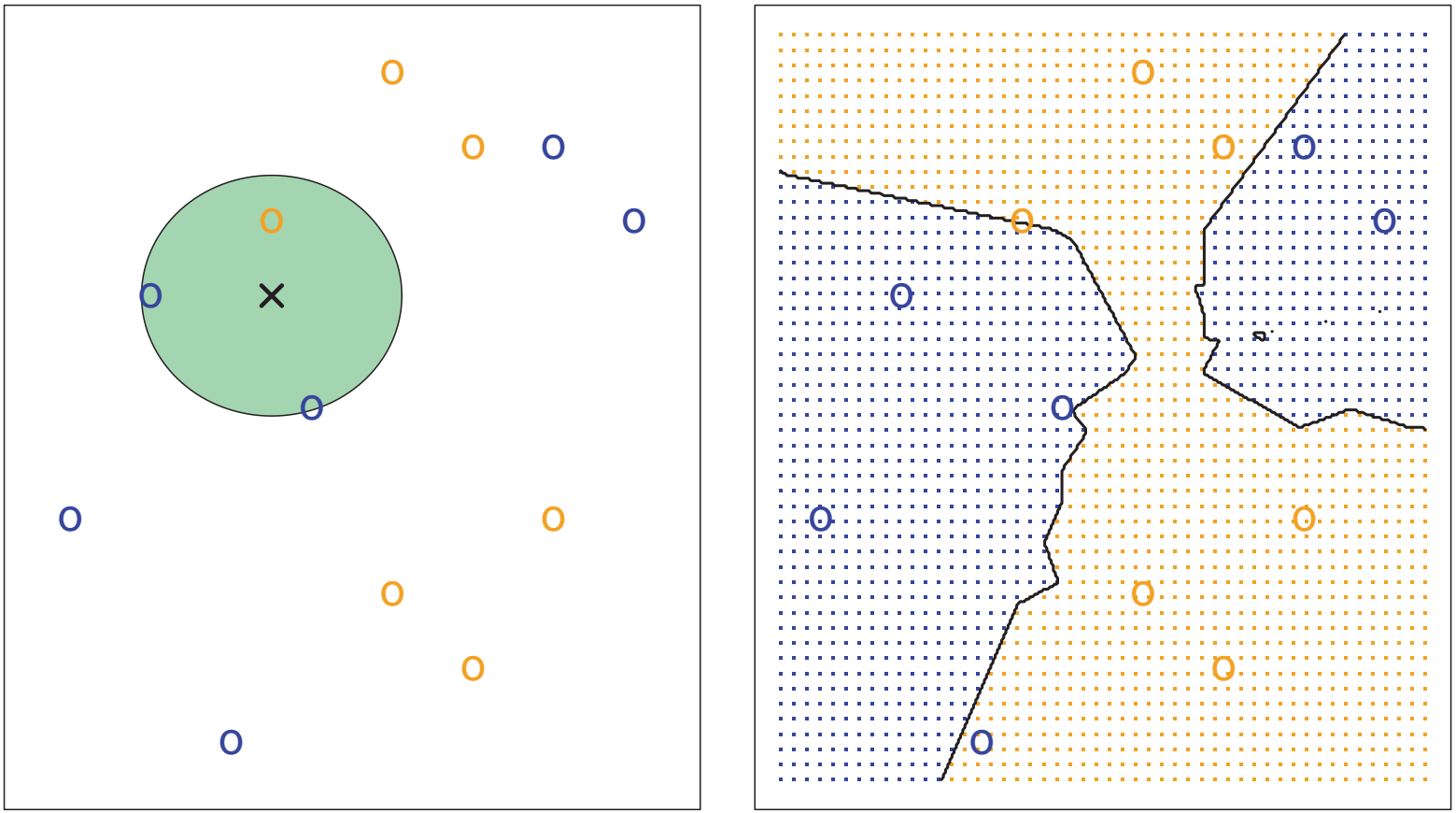


FIGURE 2.14. *The KNN approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

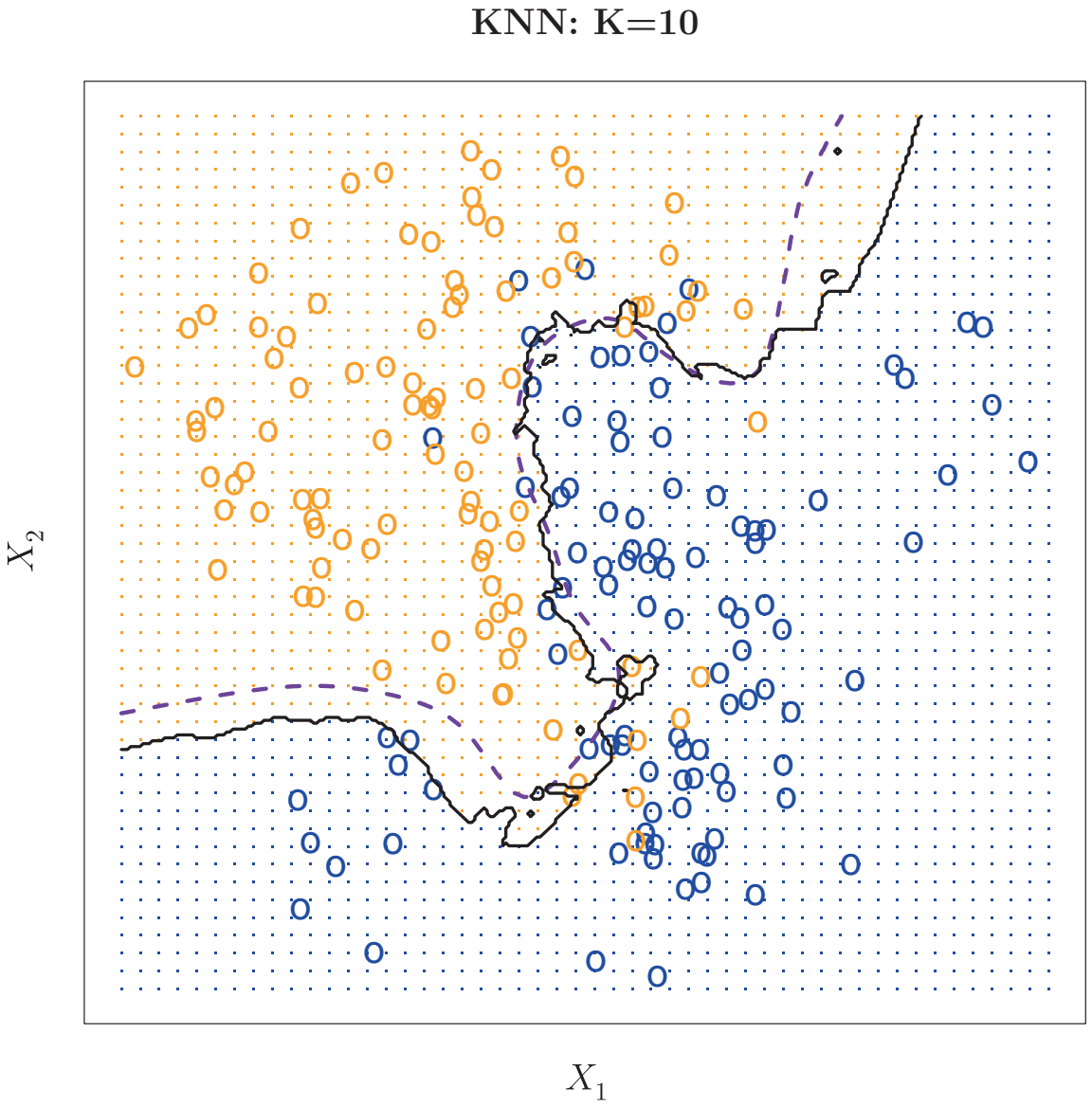


FIGURE 2.15. *The black curve indicates the KNN decision boundary on the data from Figure 2.13, using $K = 10$. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.*

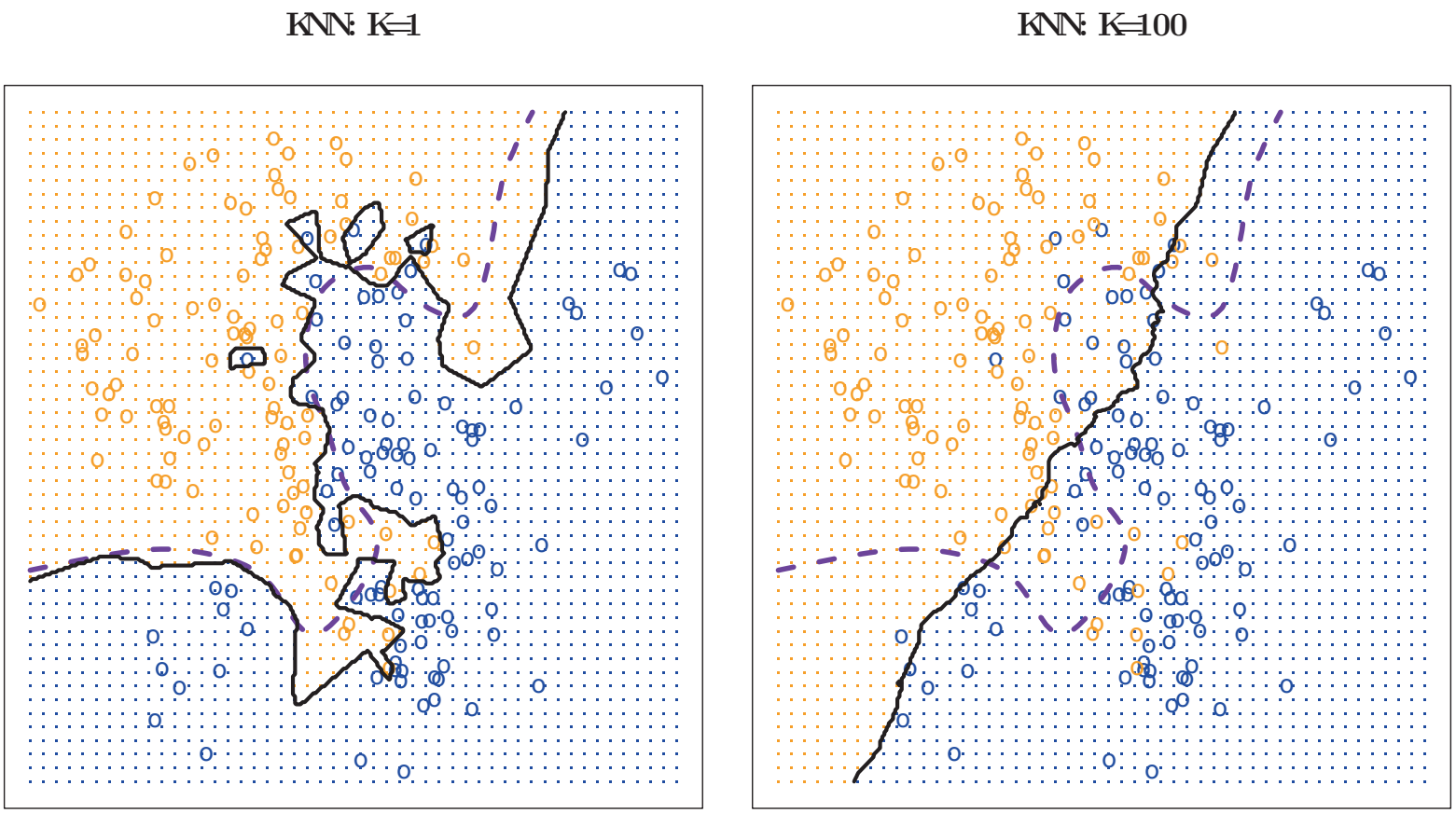


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.*

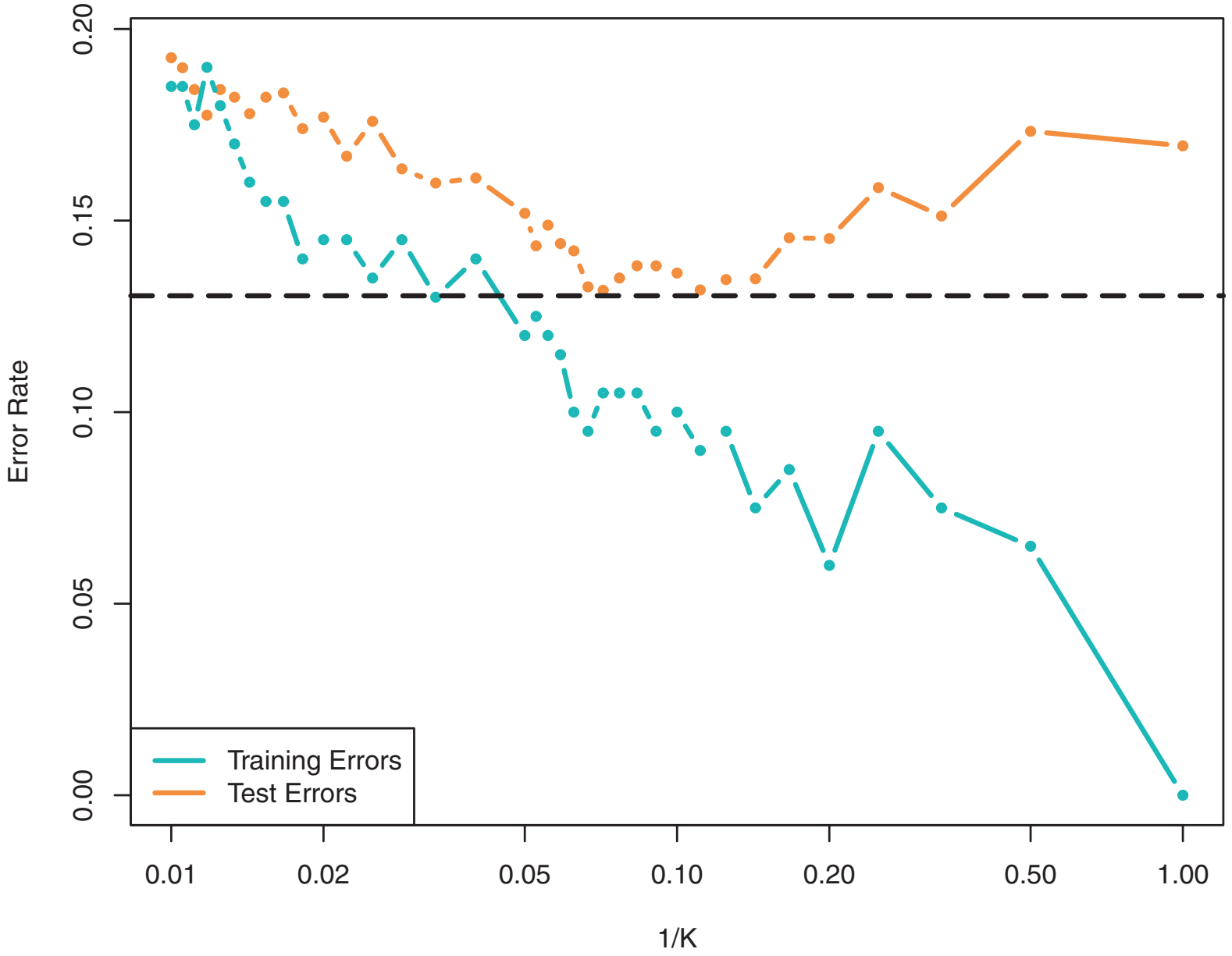


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using* $1/K$*) increases, or equivalently as the number of neighbors* K *decreases. The black dashed line indicates the Bayes error rate.*

R lab

- basic commands (an introduction to statistical learning)
- entering/building/indexing matrices (R reference)
- data analysis and graphics using R
  - vectors: concatenation, subsets, patterned, missing values, a factor is stored internally as a numeric vector with values 1,2,3,. . .,k.
  - data frames: a generalization of a matrix, in which different columns may have different modes. all elements of any column must, however, have the same mode, i.e. all numeric, or all factor, or all character, or all logical.
  - data(X)= copy X into the workspace
  - attach(a): a$s vs. s

chapter 3 : linear regression : lecture 4

- linear regression, RSS matrix notation
- $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ is assumed to come from:
  $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$
- $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- contour and three dimensional plots of RSS

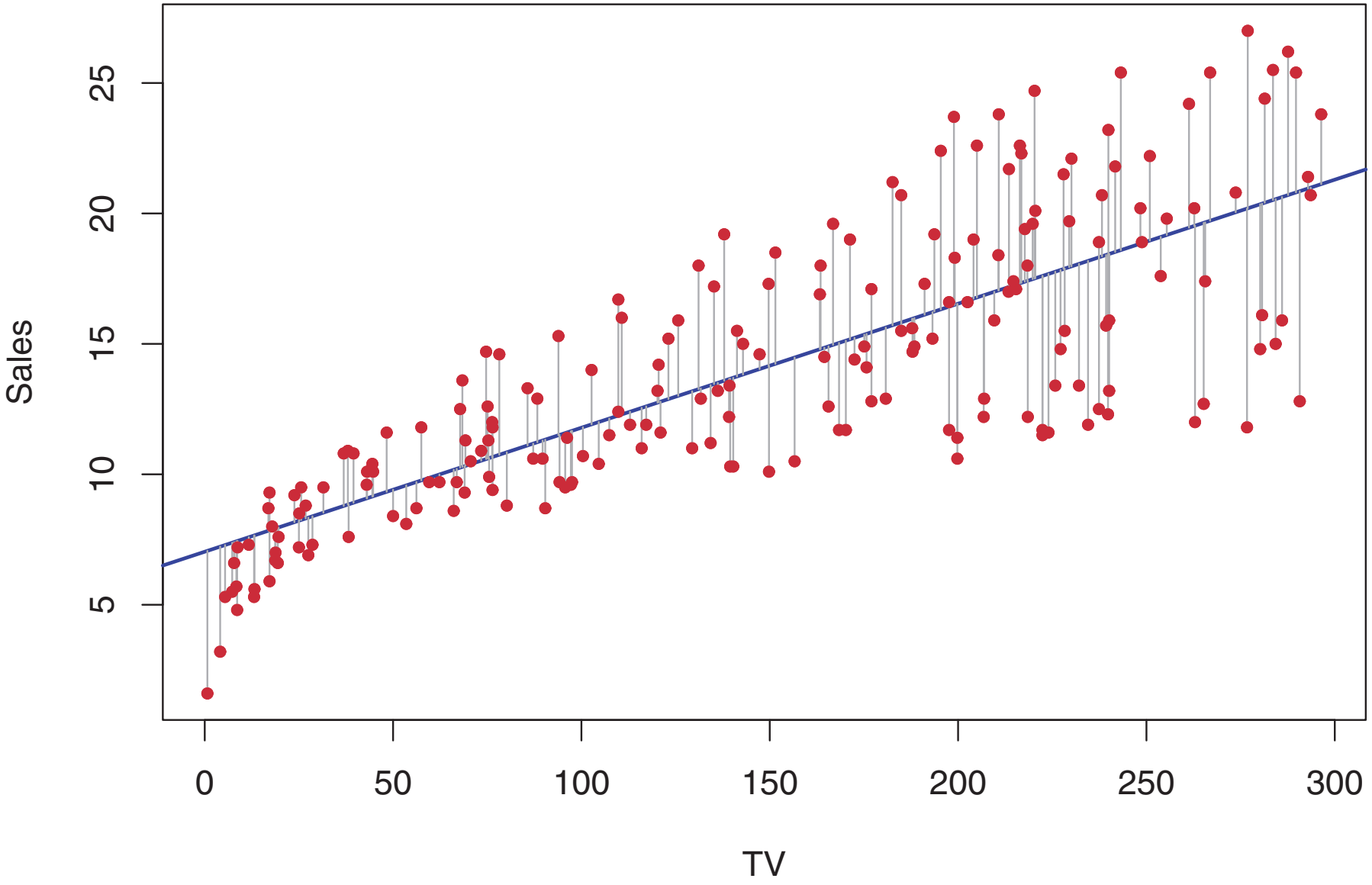


FIGURE 3.1. *For the* `Advertising` *data, the least squares fit for the regression of* `sales` *onto* `TV` *is shown. The fit is found by minimizing the sum of squared errors. Each grey line segment represents an error, and the fit makes a compromise by averaging their squares. In this case a linear fit captures the essence of the relationship, although it is somewhat deficient in the left of the plot.*

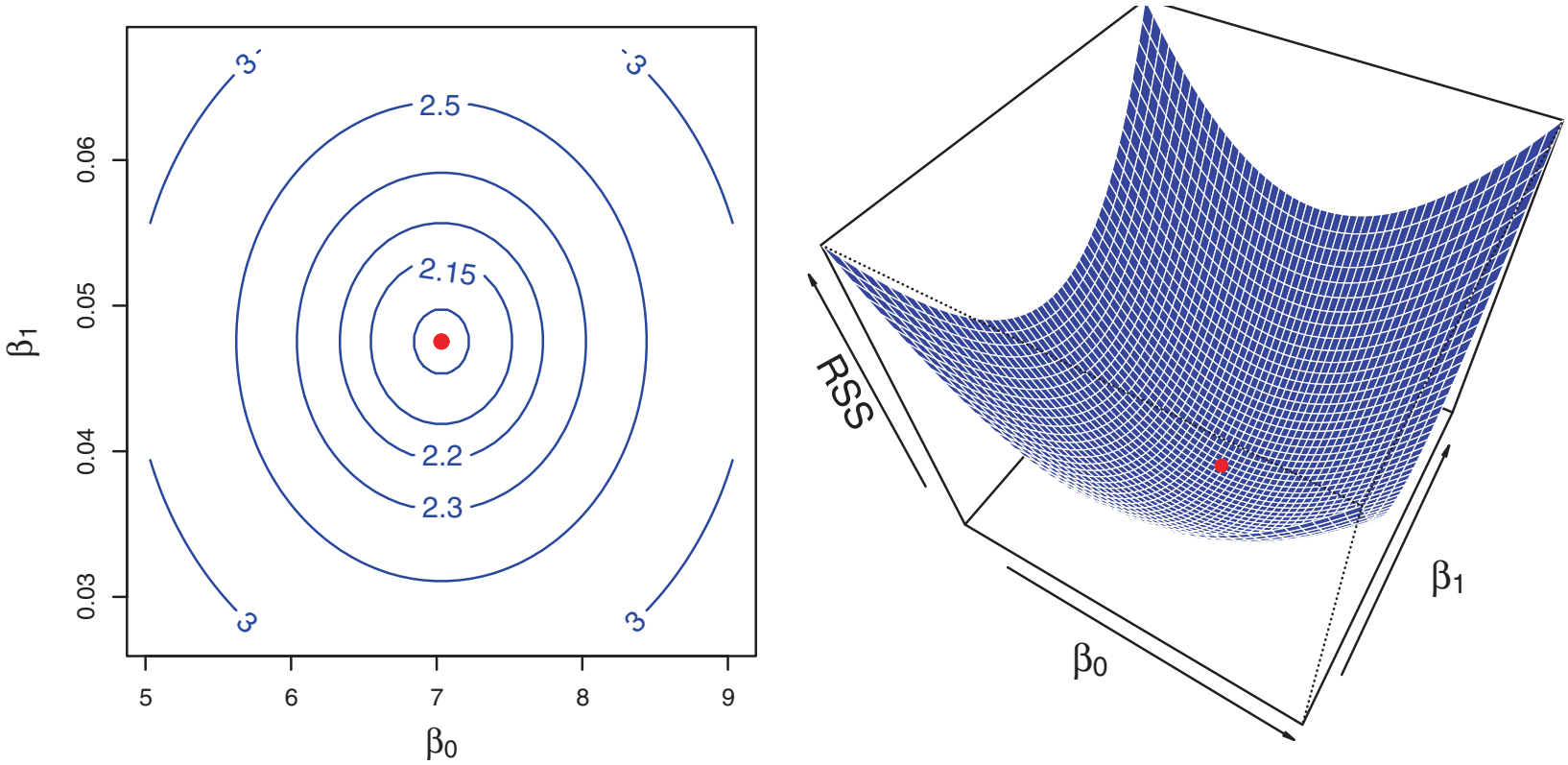


FIGURE 3.2. *Contour and three-dimensional plots of the RSS on the* `Advertising` *data, using* `sales` *as the response and* `TV` *as the predictor. The red dots correspond to the least squares estimates* $\hat{\beta}_0$ *and* $\hat{\beta}_1$*, given by (3.4).*

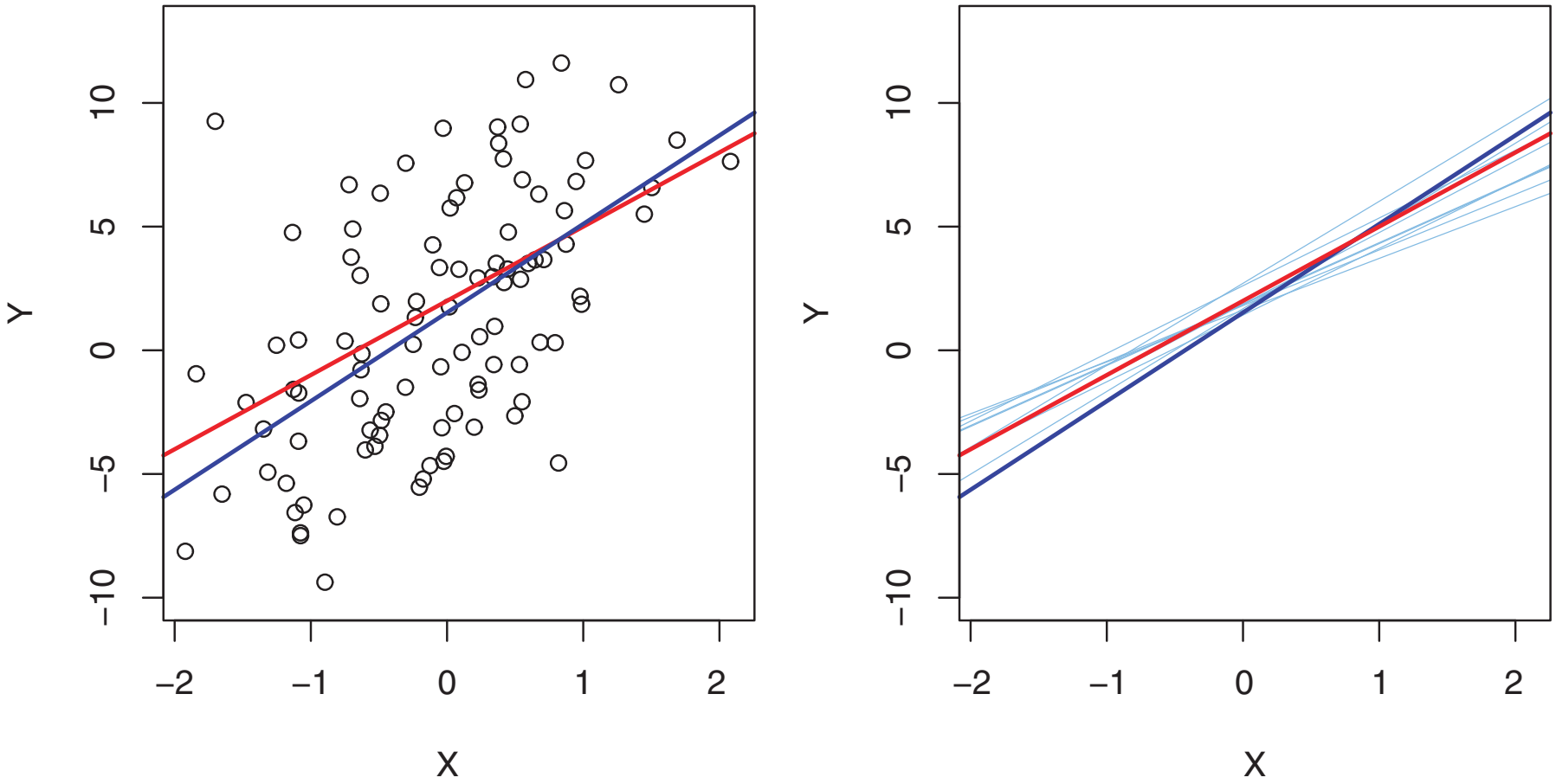


FIGURE 3.3. *A simulated data set.* Left: *The red line represents the true relationship,* $f(X) = 2 + 3X$*, which is known as the population regression line. The blue line is the least squares line; it is the least squares estimate for* $f(X)$ *based on the observed data, shown in black.* Right: *The population regression line is again shown in red, and the least squares line in dark blue. In light blue, ten least squares lines are shown, each computed on the basis of a separate random set of observations. Each least squares line is different, but on average, the least squares lines are quite close to the population regression line.*

	Coefficient	Std. error	t-statistic	p-value
Intercept	7.0325	0.4578	15.36	< 0.0001
TV	0.0475	0.0027	17.67	< 0.0001

TABLE 3.1. *For the* Advertising *data, coefficients of the least squares model for the regression of number of units sold on TV advertising budget. An increase of* \$1,000 *in the TV advertising budget is associated with an increase in sales by around 50 units (Recall that the* sales *variable is in thousands of units, and the* TV *variable is in thousands of dollars).*

Quantity	Value
Residual standard error	3.26
R^2	0.612
F-statistic	312.1

TABLE 3.2. *For the* `Advertising` *data, more information about the least squares model for the regression of number of units sold on TV advertising budget.*

- $y = \beta_0 + \beta_1 x + \epsilon$
- $SE(\hat{\beta}_0^2) = \sigma^2[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}]$, $SE(\hat{\beta}_1^2) = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$
- H_0 : There is no relationship between X and Y
- under H_0, t-statistics $= (\hat{\beta}_1 - 0)/SE(\hat{\beta}_1) \sim$ t-distribution with $n - 2$ degrees of freedom
- $RSE = \sqrt{\frac{RSS}{n-2}}$ where $RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$.
- RSE : if the model were correct and the true values of the unknown coefficients β_0 and β_1 were known exactly, any prediction of y on the basis of x would still off by about RSE on average.
- $R^2 = 1 - \frac{RSS}{TSS}$ =proportion of variability in y that can be explained using x, where $TSS = \sum_{i=1}^n (y_i - \bar{y})^2$.

- multiple linear regression: $y_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p + \epsilon_i$,
- collinearity? simple and multiple regression coefficients can be quiet different.
- example table 3.5: we tend to spend more on newspaper advertisement in markets where we spend more on radio advertising.

Simple regression of `sales` on `radio`

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	9.312	0.563	16.54	< 0.0001
`radio`	0.203	0.020	9.92	< 0.0001

Simple regression of `sales` on `newspaper`

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	12.351	0.621	19.88	< 0.0001
`newspaper`	0.055	0.017	3.30	< 0.0001

TABLE 3.3. *More simple linear regression models for the* `Advertising` *data. Coefficients of the simple linear regression model for number of units sold on* Top: *radio advertising budget and* Bottom: *newspaper advertising budget. A* \$1,000 *increase in spending on radio advertising is associated with an average increase in sales by around 203 units, while the same increase in spending on newspaper advertising is associated with an average increase in sales by around 55 units (Note that the* `sales` *variable is in thousands of units, and the* `radio` *and* `newspaper` *variables are in thousands of dollars).*

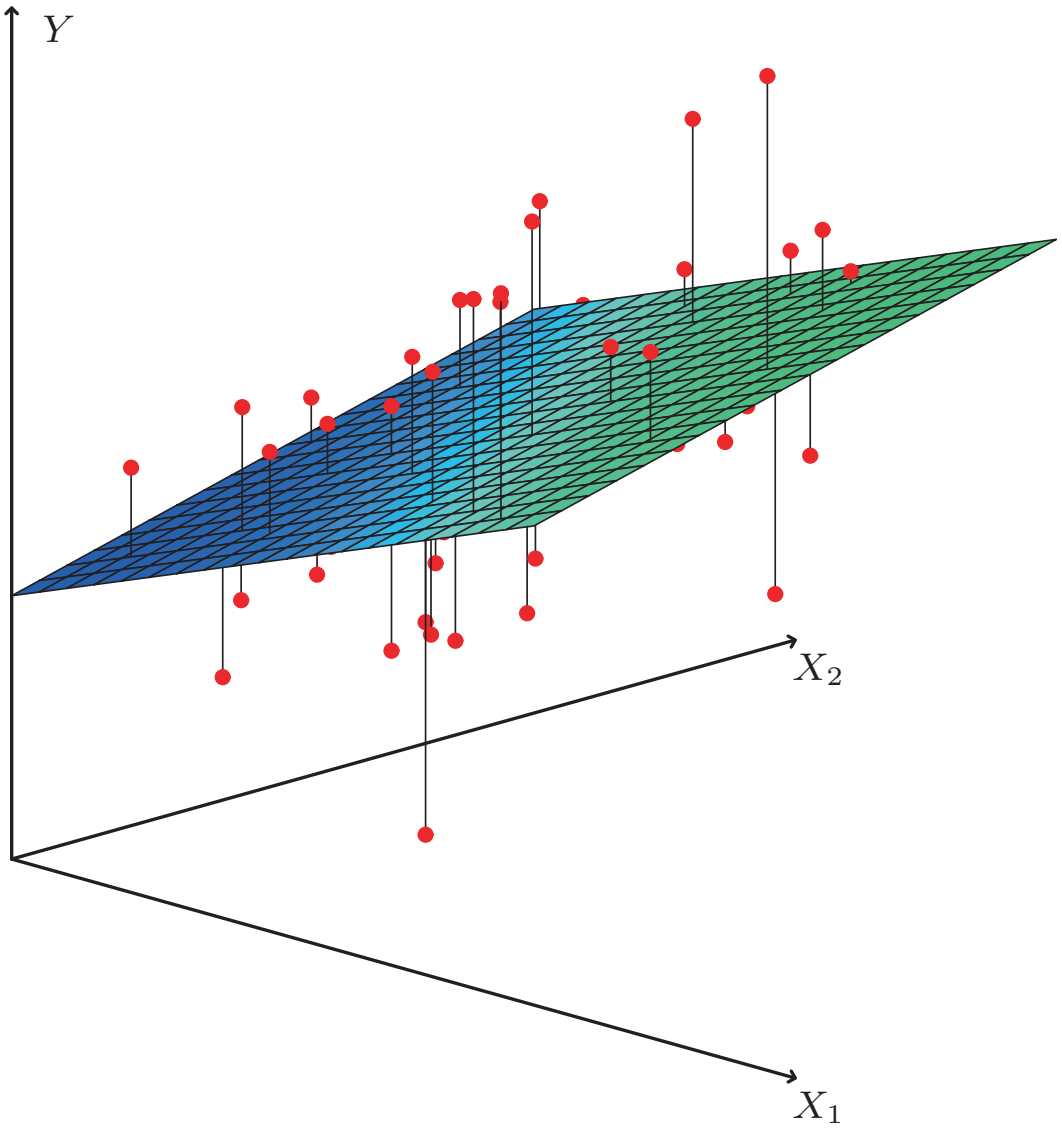


FIGURE 3.4. *In a three-dimensional setting, with two predictors and one response, the least squares regression line becomes a plane. The plane is chosen to minimize the sum of the squared vertical distances between each observation (shown in red) and the plane.*

	Coefficient	Std. error	t-statistic	p-value
Intercept	2.939	0.3119	9.42	< 0.0001
TV	0.046	0.0014	32.81	< 0.0001
radio	0.189	0.0086	21.89	< 0.0001
newspaper	-0.001	0.0059	-0.18	0.8599

TABLE 3.4. *For the* Advertising *data, least squares coefficient estimates of the multiple linear regression of number of units sold on radio, TV, and newspaper advertising budgets.*

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

TABLE 3.5. *Correlation matrix for* TV, radio, newspaper, *and* sales *for the* Advertising *data.*

- Is there a relationship between the response and the predictors?
- $H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$
- $F = \frac{(TSS-RSS)/p}{RSS/(n-p-1)}$ where $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ and $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.
- Reject H_0 if F is greater than 1.
- If $p > n$, F statistics cannot be used.

Quantity	Value
Residual standard error	1.69
R^2	0.897
F-statistic	570

TABLE 3.6. *More information about the least squares model for the regression of number of units sold on TV, newspaper, and radio advertising budgets in the* `Advertising` *data. Other information about this model was displayed in Table 3.4.*

- variable selection: which predictors/features are associated with the response
- how many models that contain subsets of p features?
- forward selection: start with no variable, iteratively add the variable that results in the lowest RSS. continue until some stopping rule is satisfied? AIC, BIC, adjusted-R^2
- backward selection: start with all variables in the model, iteratively remove variables with the highest p-value, fit the model again with the remaining variables, etc. continue until all variables have a p-value below some threshold.
- mixed selection: start with no variables, iteratively add variables until the p-value for some variable becomes large. remove that variable, continue adding, removing...

- model fit: R^2 will always increase when more variables are added to the model.
- $RSE = \sqrt{\frac{RSS}{n-p-1}}$, therefore the RSE can increase when a predictor is added while the RSS must decrease.
- non-linear terms?

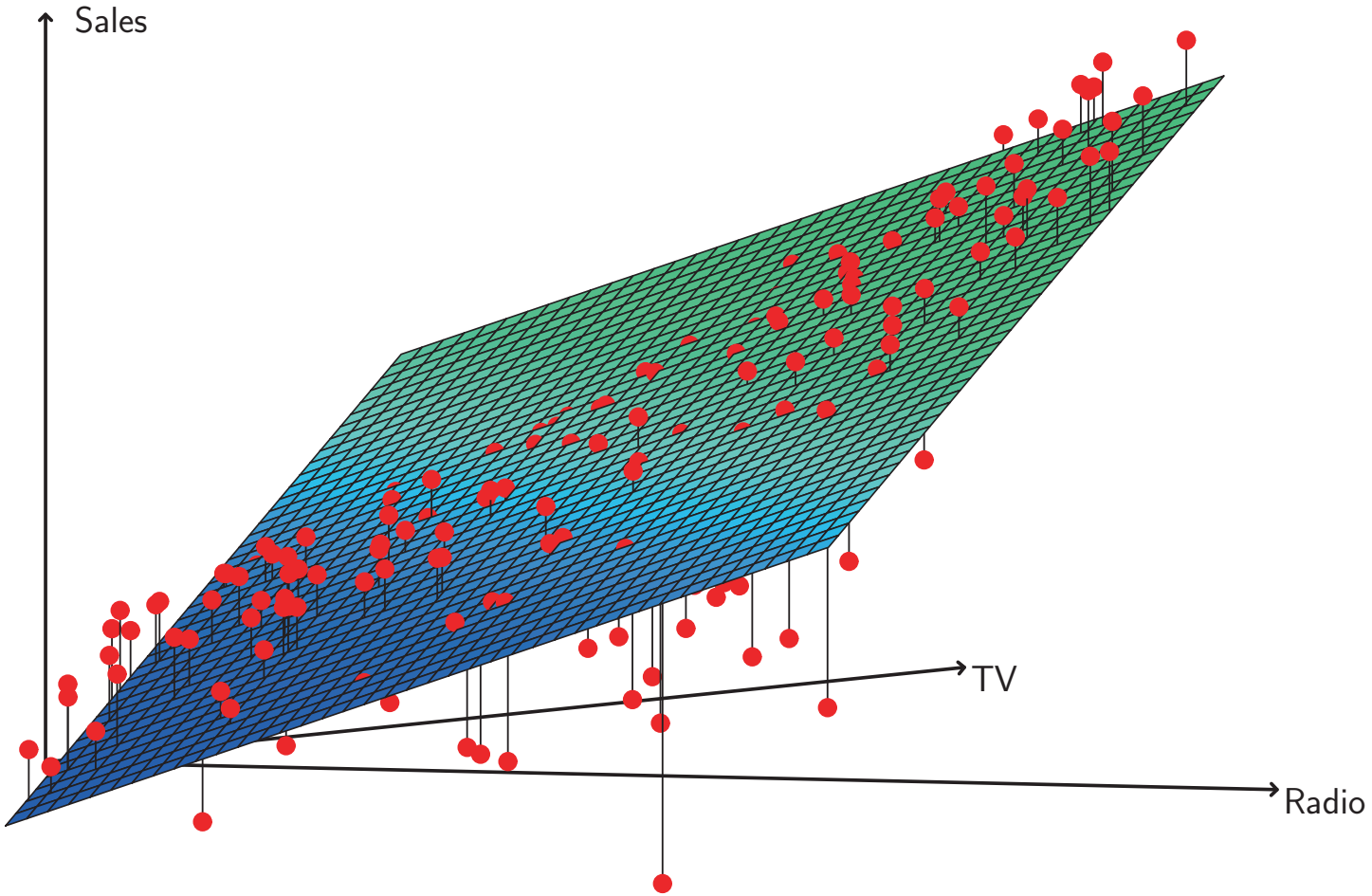


FIGURE 3.5. *For the* `Advertising` *data, a linear regression fit to* `sales` *using* `TV` *and* `radio` *as predictors. From the pattern of the residuals, we can see that there is a pronounced non-linear relationship in the data.*

- prediction interval versus confidence interval ?
- prediction interval $\sim$ reducible estimate (error in the estimate of $f(X)$) + irreducible error (individual data points vs. population mean)
- prediction interval is always wider than a confidence interval.

- Qualitative predictors with more than two levels.

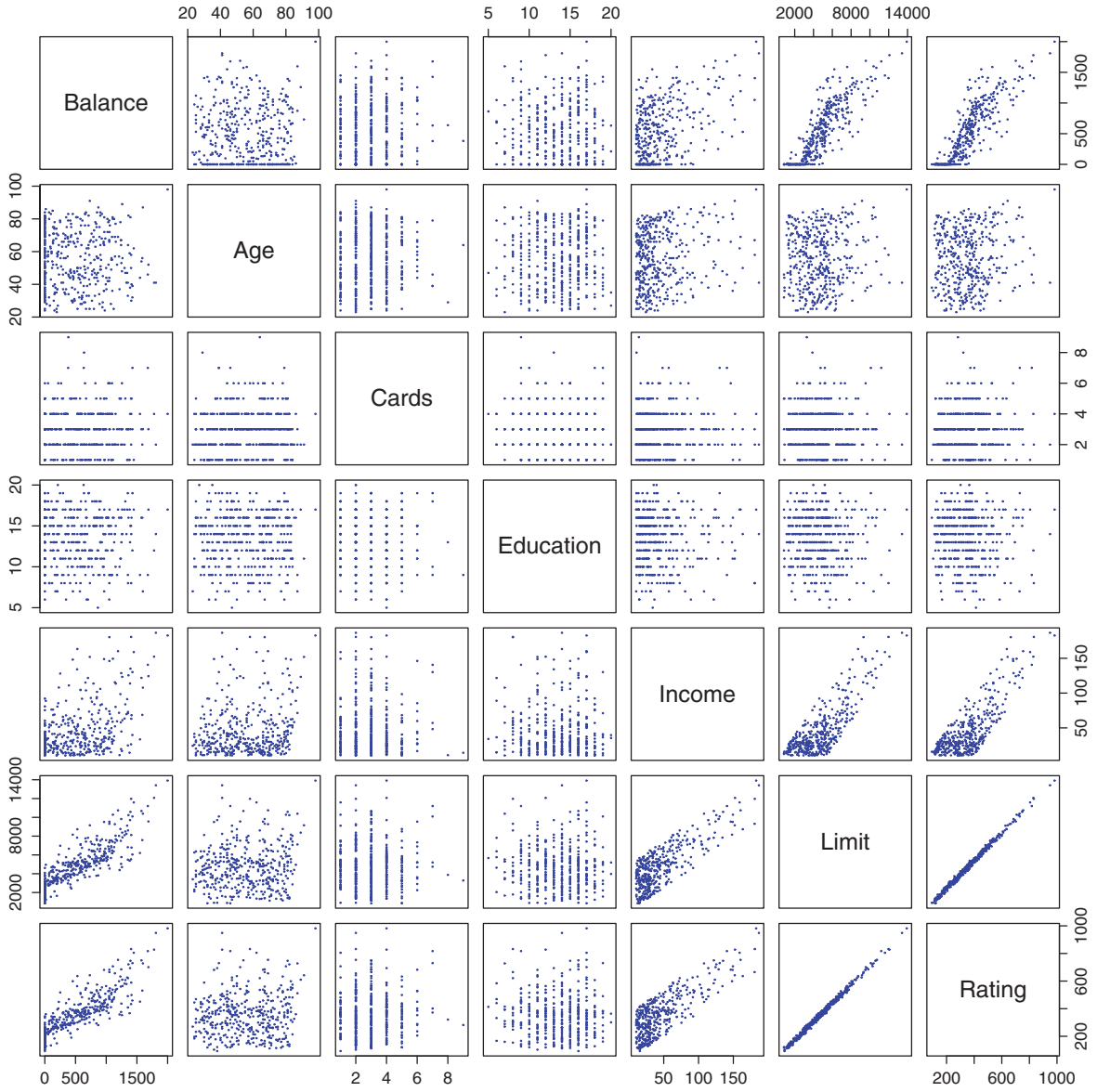


FIGURE 3.6. *The* `Credit` *data set contains information about* `balance`, `age`, `cards`, `education`, `income`, `limit`, *and* `rating` *for a number of potential customers.*

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male,} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ -1 & \text{if } i\text{th person is male} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 - \beta_1 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

$$x_{i1} = \begin{cases} 1 & \text{if } i\text{th person is Asian} \\ 0 & \text{if } i\text{th person is not Asian,} \end{cases} \tag{3.28}$$

and the second could be

$$x_{i2} = \begin{cases} 1 & \text{if } i\text{th person is Caucasian} \\ 0 & \text{if } i\text{th person is not Caucasian.} \end{cases} \tag{3.29}$$

Then both of these variables can be used in the regression equation, in order to obtain the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is Asian} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i\text{th person is Caucasian} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is African American.} \end{cases}$$

- interaction term. $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$. Adjusting X_2 will change the impact of X_1 on Y .
- if we include an interaction in a model, we should also include the main effects, even if the p-values associated with principle their coefficients are not significant.
- interaction between a qualitative and a quantitative variable
- non-linear Relationships

	Coefficient	Std. error	t-statistic	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
radio	0.0289	0.009	3.24	0.0014
TV×radio	0.0011	0.000	20.73	< 0.0001

TABLE 3.9. *For the* Advertising *data, least squares coefficient estimates associated with the regression of* sales *onto* TV *and* radio*, with an interaction term, as in (3.33).*

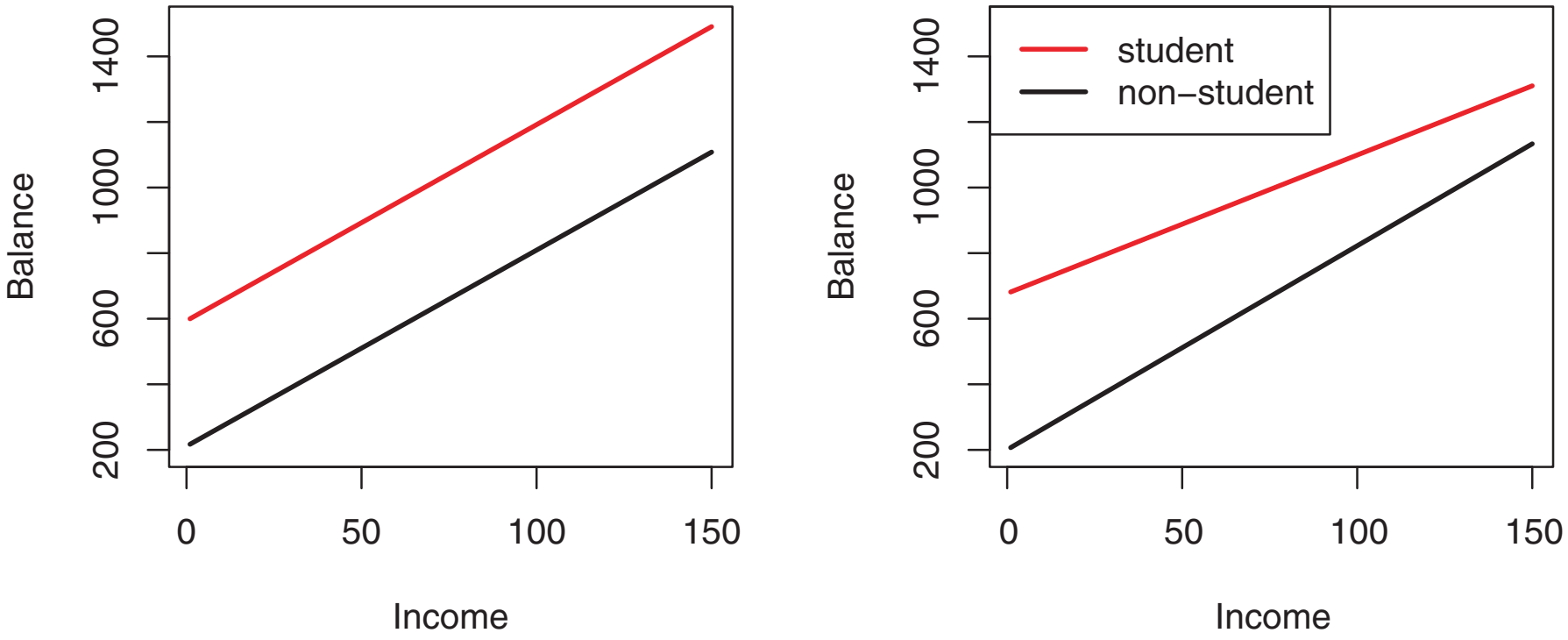


FIGURE 3.7. *For the* `Credit` *data, the least squares lines are shown for prediction of* `balance` *from* `income` *for students and non-students.* Left: *The model (3.34) was fit. There is no interaction between* `income` *and* `student`. Right: *The model (3.35) was fit. There is an interaction term between* `income` *and* `student`.

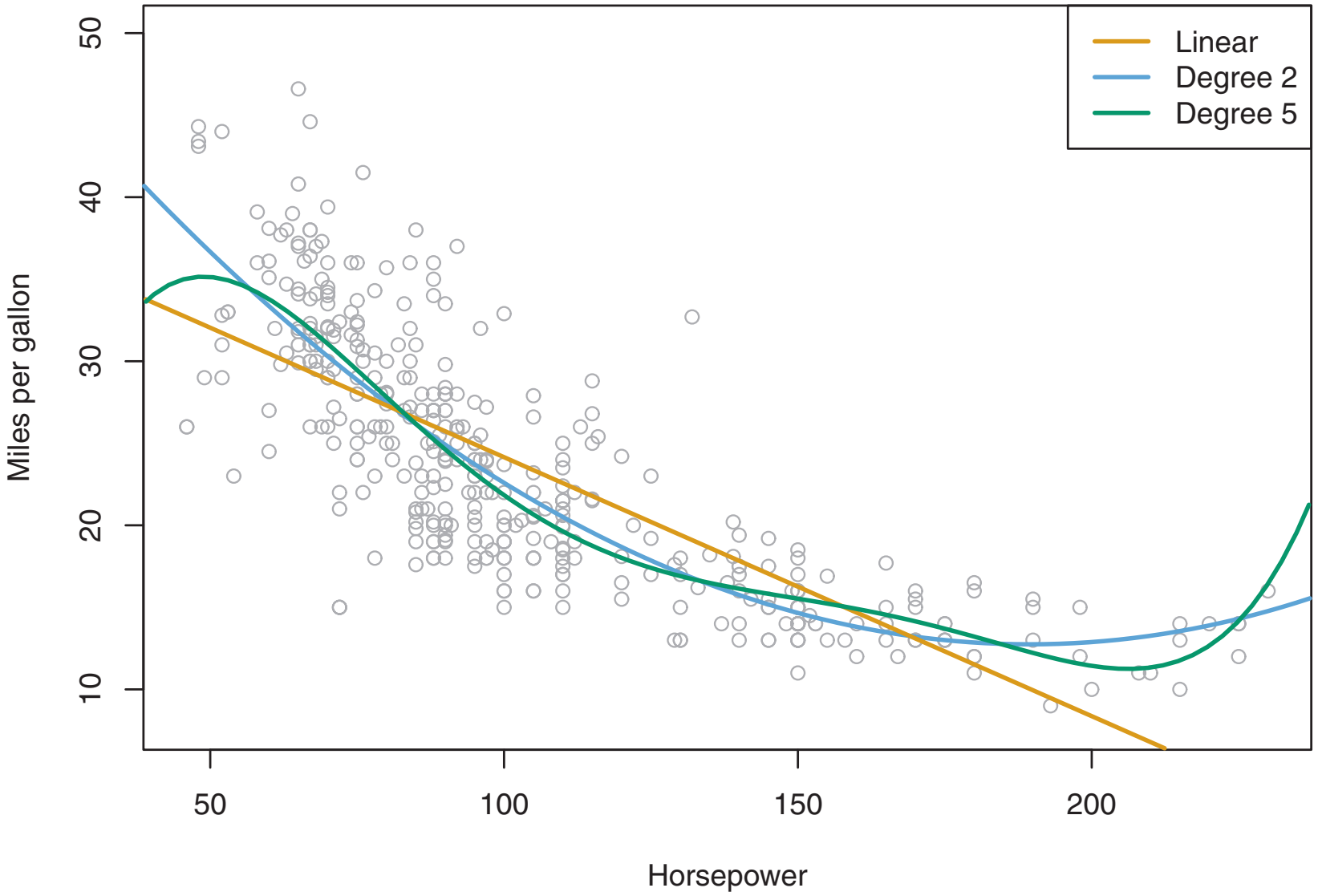


FIGURE 3.8. *The* `Auto` *data set. For a number of cars,* `mpg` *and* `horsepower` *are shown. The linear regression fit is shown in orange. The linear regression fit for a model that includes* $\texttt{horsepower}^2$ *is shown as a blue curve. The linear regression fit for a model that includes all polynomials of* `horsepower` *up to fifth-degree is shown in green.*

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	56.9001	1.8004	31.6	< 0.0001
`horsepower`	-0.4662	0.0311	-15.0	< 0.0001
`horsepower`^{2}	0.0012	0.0001	10.1	< 0.0001

TABLE 3.10. *For the* `Auto` *data set, least squares coefficient estimates associated with the regression of* `mpg` *onto* `horsepower` *and* `horsepower`^{2}.

- potential problems:
  - non-linearity of the data.
  - correlation of error terms.
  - non-constant variance of error terms.
  - outliers.
  - high leverage points.
  - collinearity.

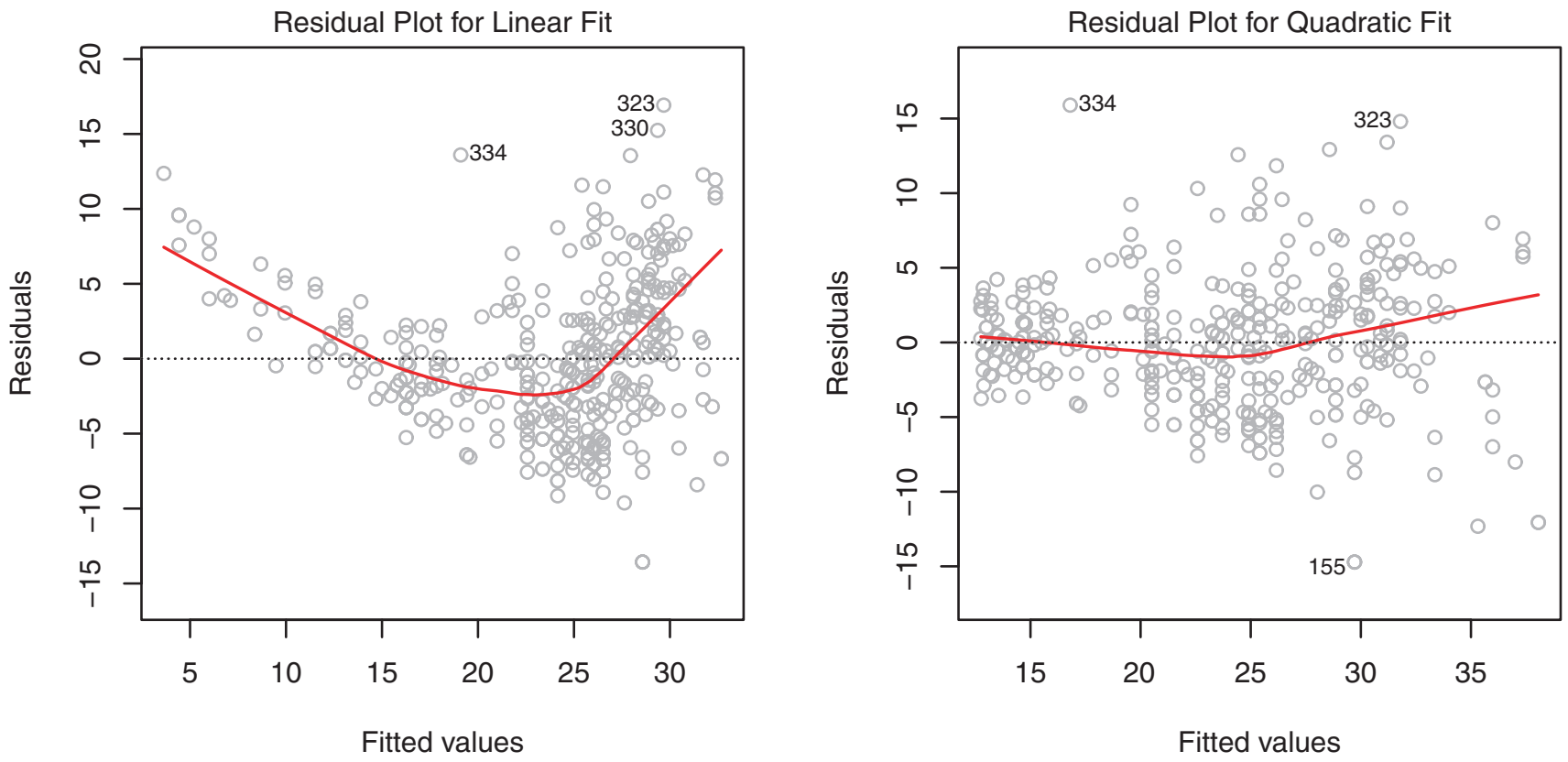
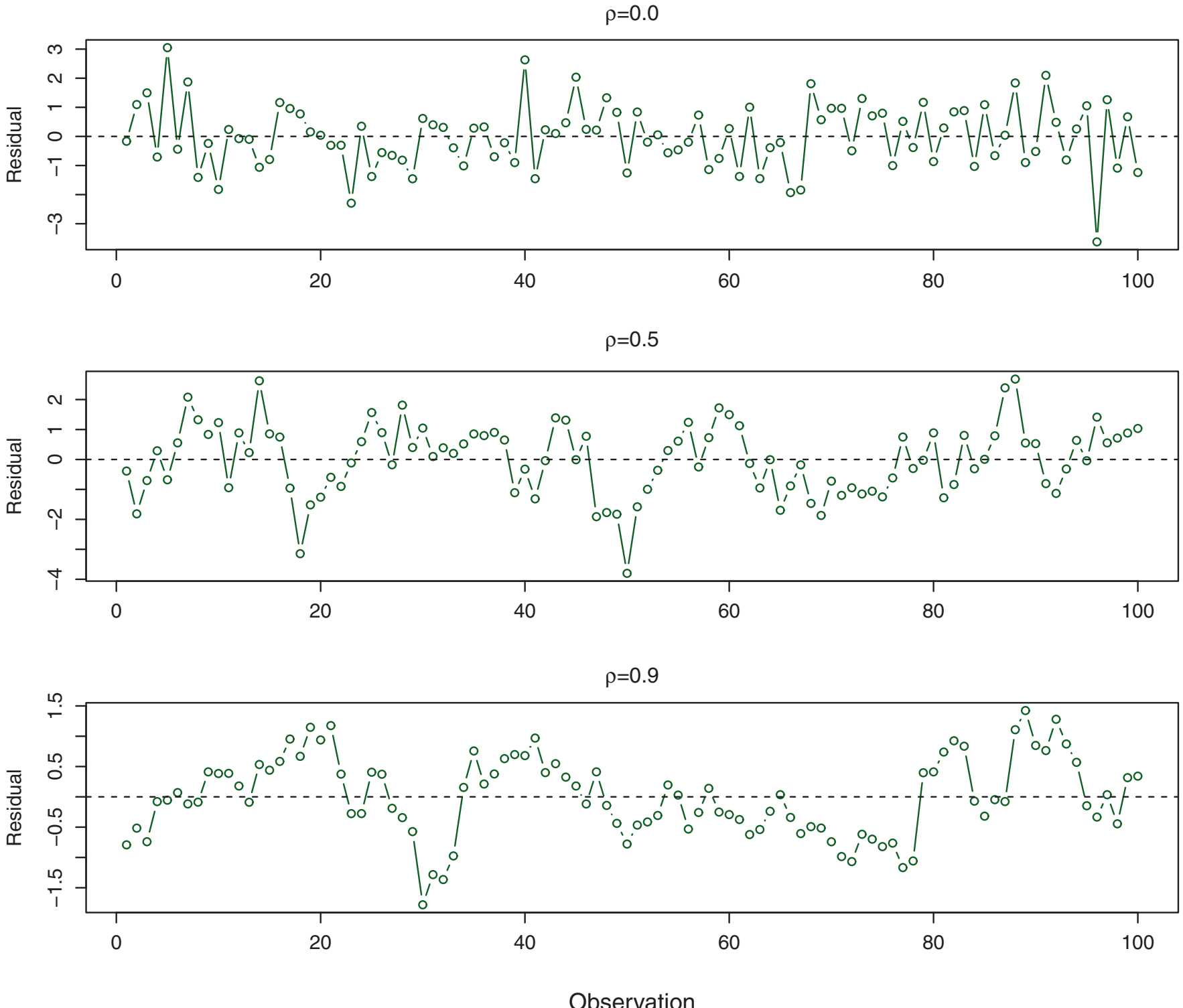


FIGURE 3.9. *Plots of residuals versus predicted (or fitted) values for the* `Auto` *data set. In each plot, the red line is a smooth fit to the residuals, intended to make it easier to identify a trend.* Left: *A linear regression of* `mpg` *on* `horsepower`*. A strong pattern in the residuals indicates non-linearity in the data.* Right: *A linear regression of* `mpg` *on* `horsepower` *and* $\texttt{horsepower}^2$*. There is little pattern in the residuals.*


ρ=0.0
Residual
3
2
1
0
−1
−3
0
20
40
60
80
100
ρ=0.5
Residual
2
1
0
−2
−4
0
20
40
60
80
100
ρ=0.9
Residual
1.5
0.5
−0.5
−1.5
0
20
40
60
80
100
Observation

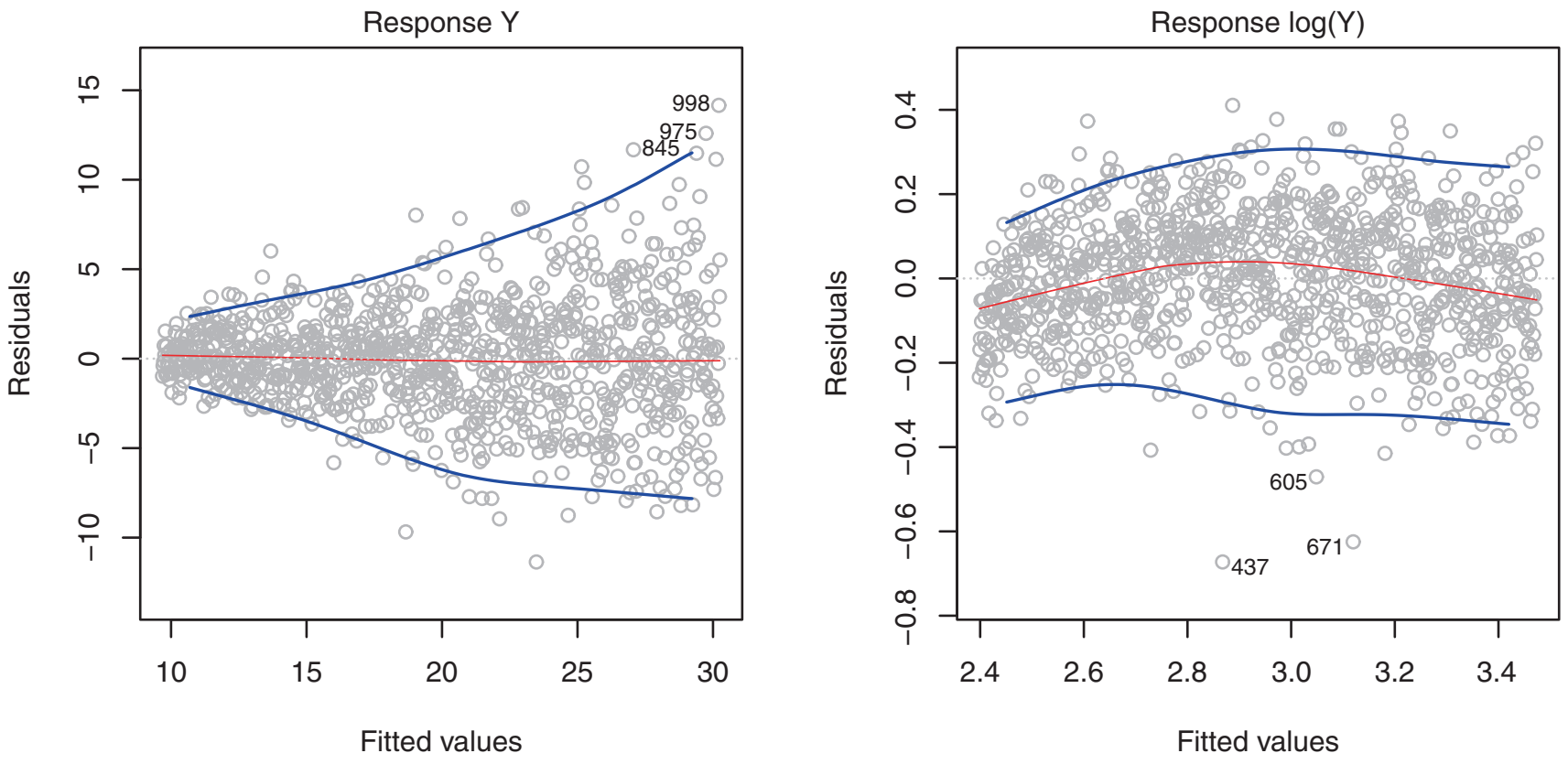


FIGURE 3.11. *Residual plots. In each plot, the red line is a smooth fit to the residuals, intended to make it easier to identify a trend. The blue lines track the outer quantiles of the residuals, and emphasize patterns.* Left: *The funnel shape indicates heteroscedasticity.* Right: *The predictor has been log-transformed, and there is now no evidence of heteroscedasticity.*

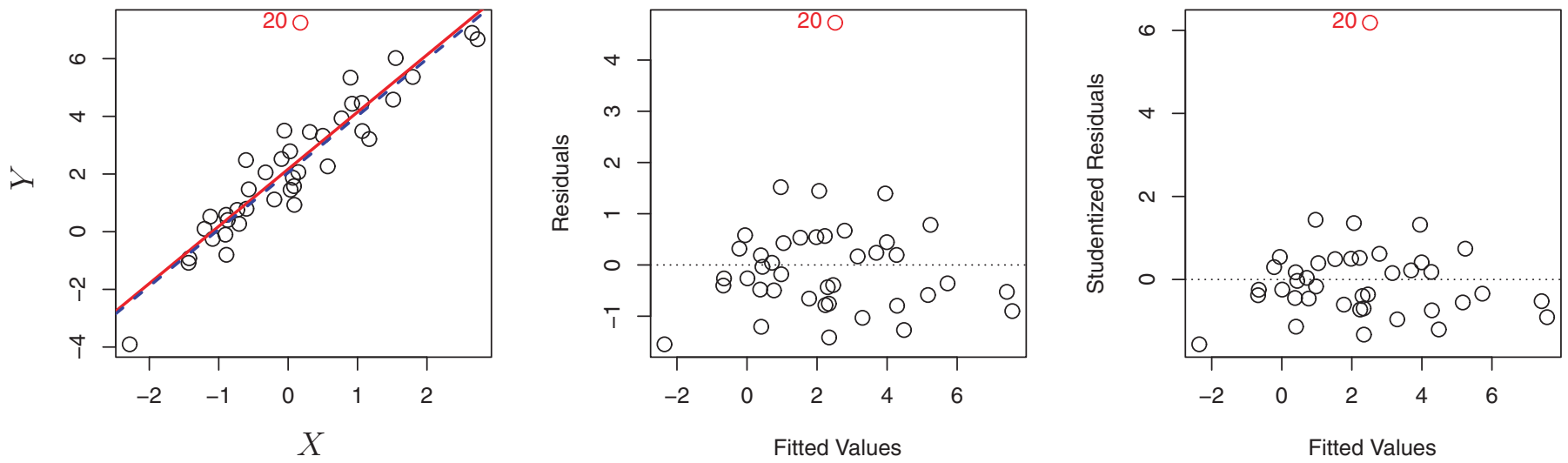


FIGURE 3.12. Left: *The least squares regression line is shown in red, and the regression line after removing the outlier is shown in blue.* Center: *The residual plot clearly identifies the outlier.* Right: *The outlier has a studentized residual of* 6*; typically we expect values between* -3 *and* 3.

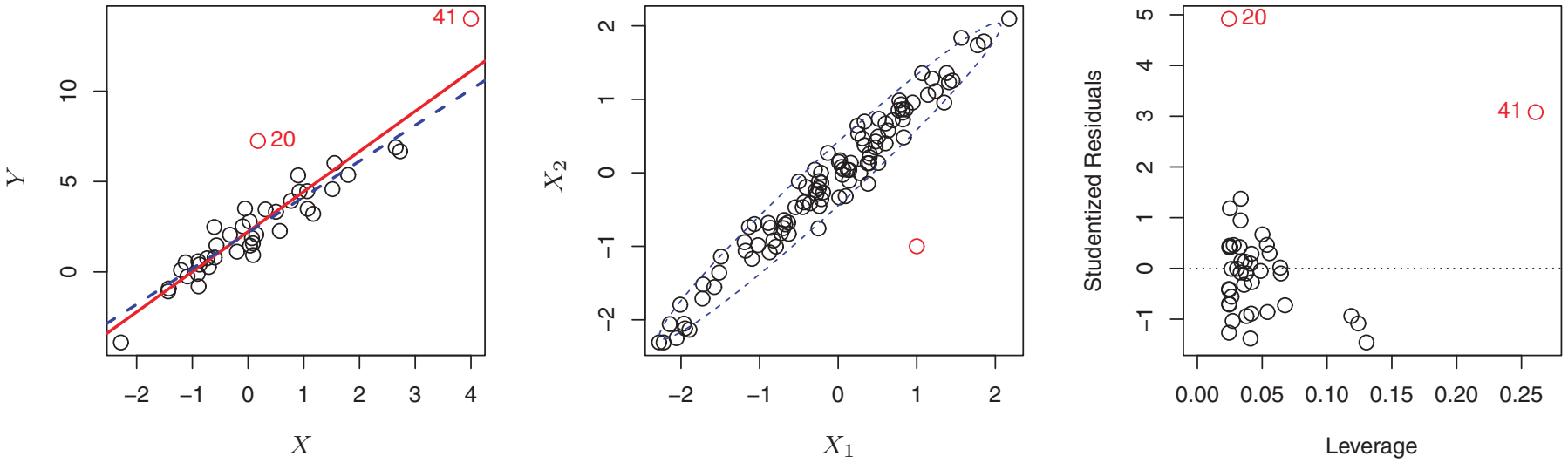


FIGURE 3.13. Left: *Observation 41 is a high leverage point, while 20 is not. The red line is the fit to all the data, and the blue line is the fit with observation 41 removed.* Center: *The red observation is not unusual in terms of its* X_1 *value or its* X_2 *value, but still falls outside the bulk of the data, and hence has high leverage.* Right: *Observation* 41 *has a high leverage and a high residual.*

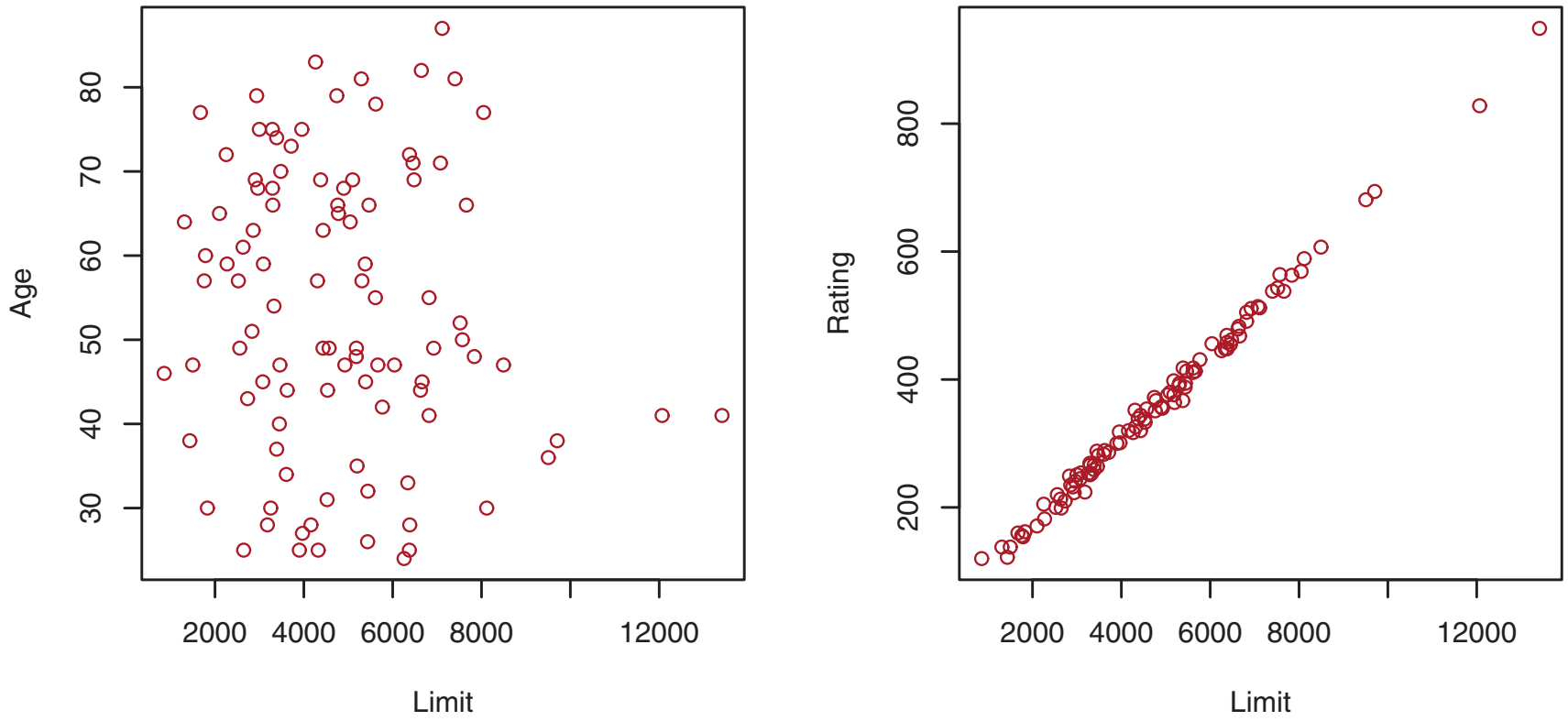


FIGURE 3.14. *Scatterplots of the observations from the* `Credit` *data set.* Left: *A plot of* `age` *versus* `limit`*. These two variables are not collinear.* Right: *A plot of* `rating` *versus* `limit`*. There is high collinearity.*

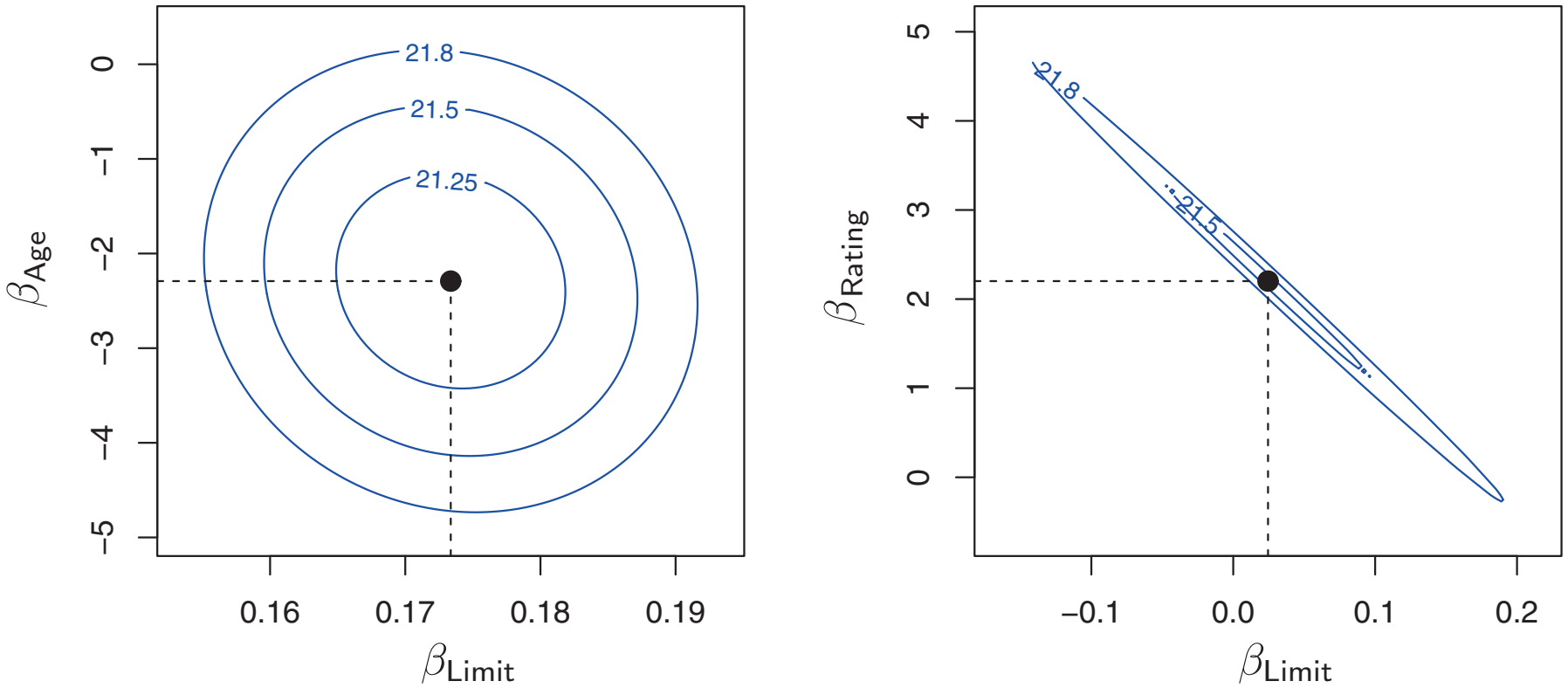


FIGURE 3.15. *Contour plots for the RSS values as a function of the parameters β for various regressions involving the* `Credit` *data set. In each plot, the black dots represent the coefficient values corresponding to the minimum RSS.* Left: *A contour plot of RSS for the regression of* `balance` *onto* `age` *and* `limit`*. The minimum value is well defined.* Right: *A contour plot of RSS for the regression of* `balance` *onto* `rating` *and* `limit`*. Because of the collinearity, there are many pairs* $(\beta_{\text{Limit}}, \beta_{\text{Rating}})$ *with a similar value for RSS.*

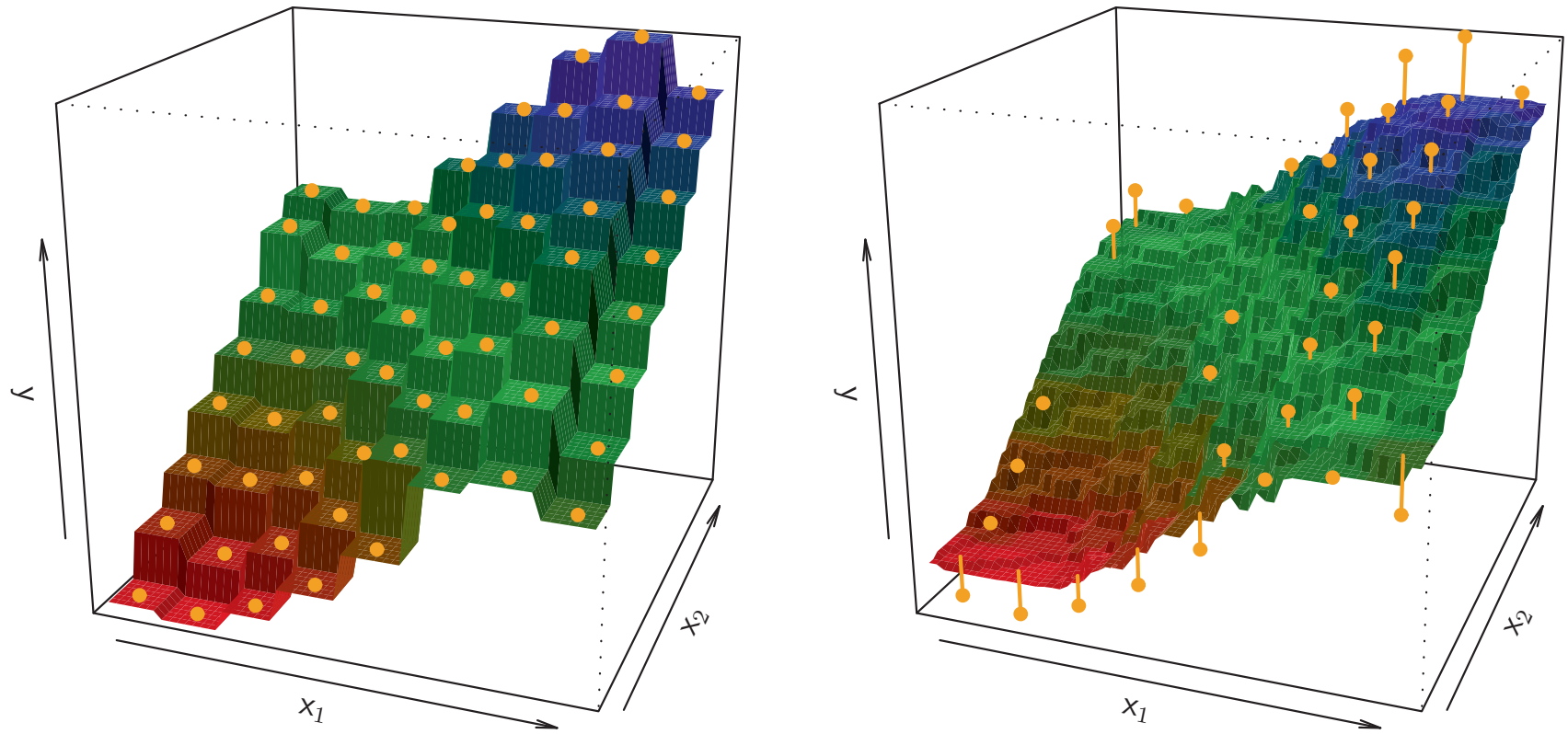


FIGURE 3.16. *Plots of $\hat{f}(X)$ using KNN regression on a two-dimensional data set with* 64 *observations (orange dots).* Left: $K = 1$ *results in a rough step function fit.* Right: $K = 9$ *produces a much smoother fit.*

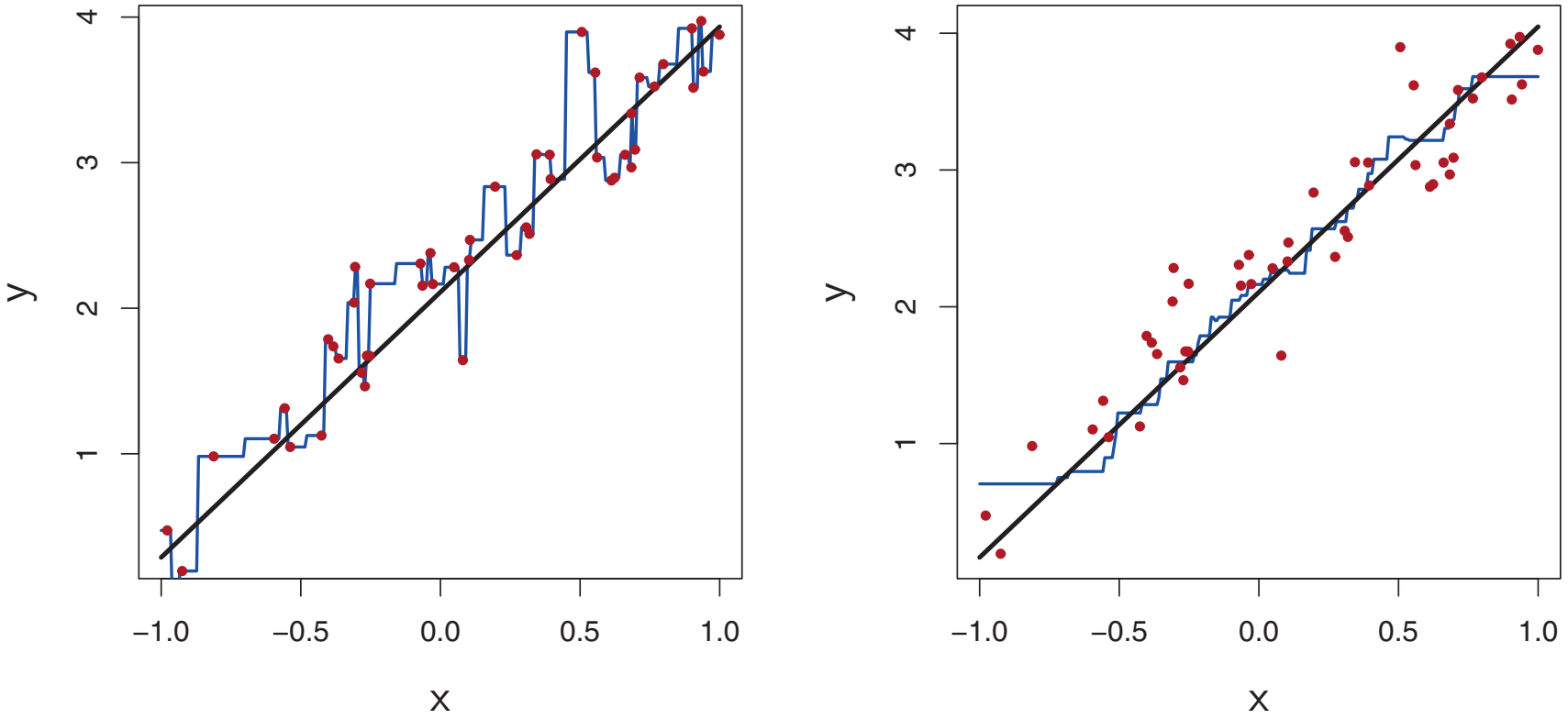


FIGURE 3.17. *Plots of $\hat{f}(X)$ using KNN regression on a one-dimensional data set with* 100 *observations. The true relationship is given by the black solid line.* Left: *The blue curve corresponds to $K = 1$ and interpolates (i.e. passes directly through) the training data.* Right: *The blue curve corresponds to $K = 9$, and represents a smoother fit.*

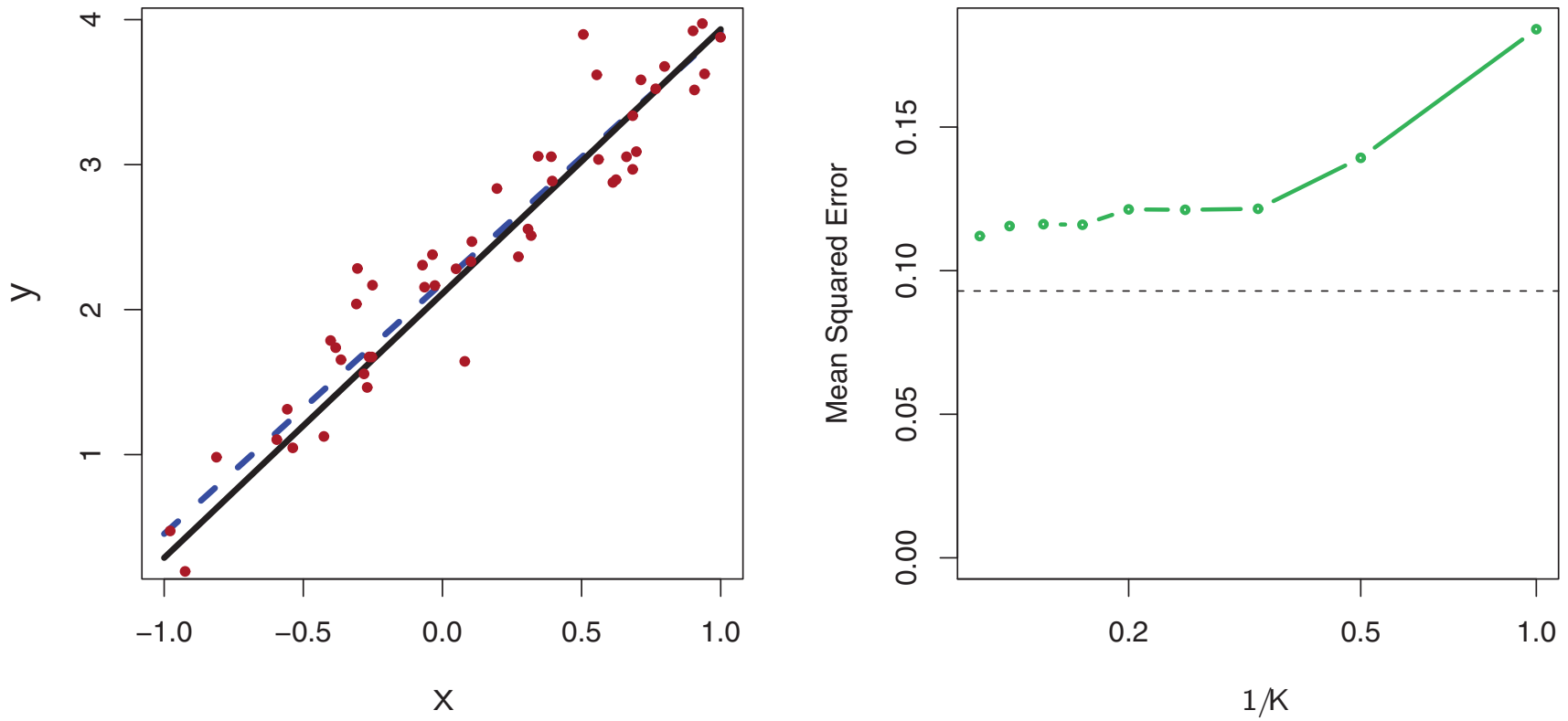


FIGURE 3.18. *The same data set shown in Figure 3.17 is investigated further.* Left: *The blue dashed line is the least squares fit to the data. Since* $f(X)$ *is in fact linear (displayed as the black line), the least squares regression line provides a very good estimate of* $f(X)$. Right: *The dashed 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* $1/K$.

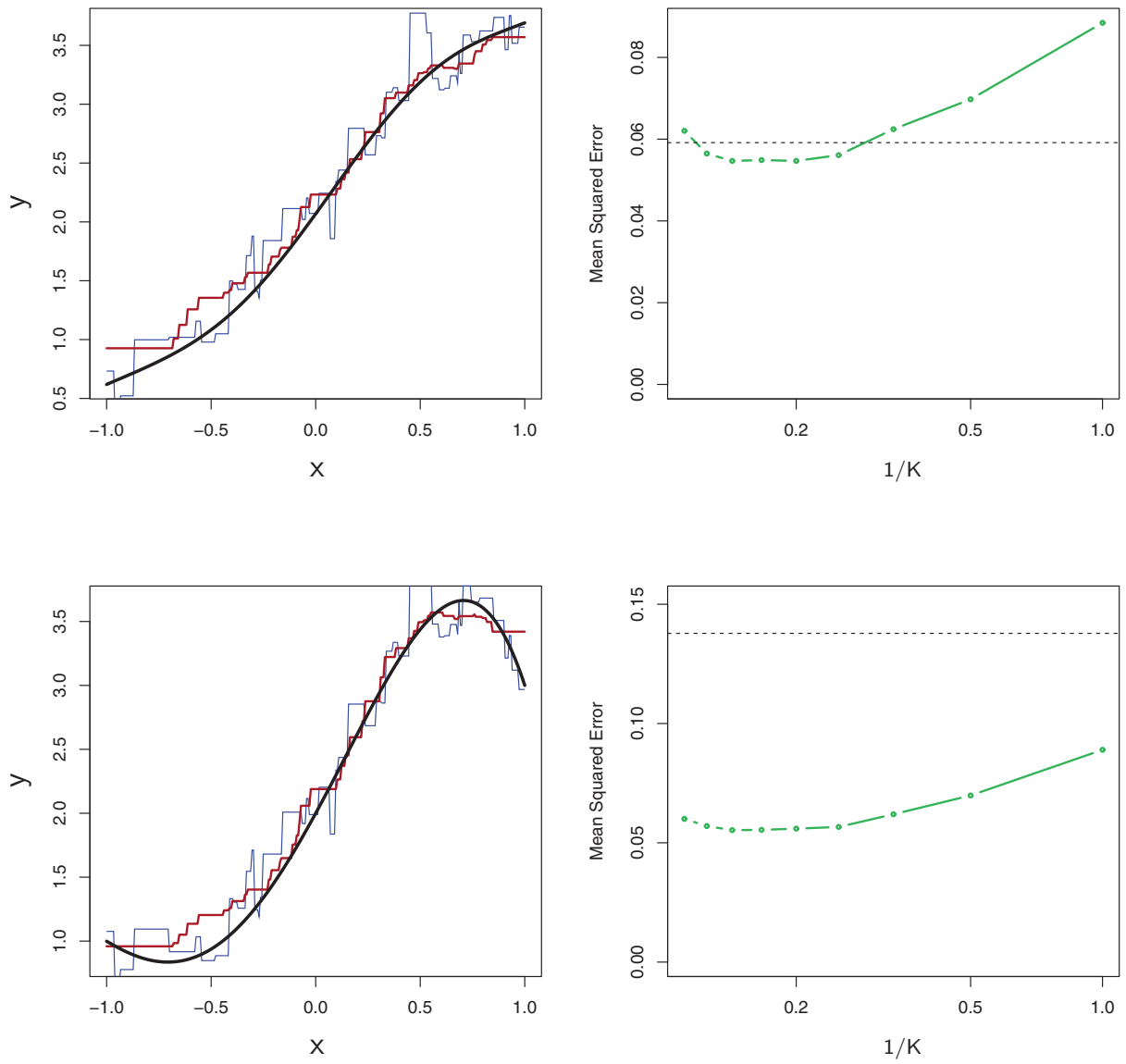


FIGURE 3.19. Top Left: *In a setting with a slightly non-linear relationship between X and Y (solid black line), the KNN fits with $K = 1$ (blue) and $K = 9$ (red) are displayed.* Top Right: *For the slightly non-linear data, the test set MSE for least squares regression (horizontal black) and KNN with 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.*

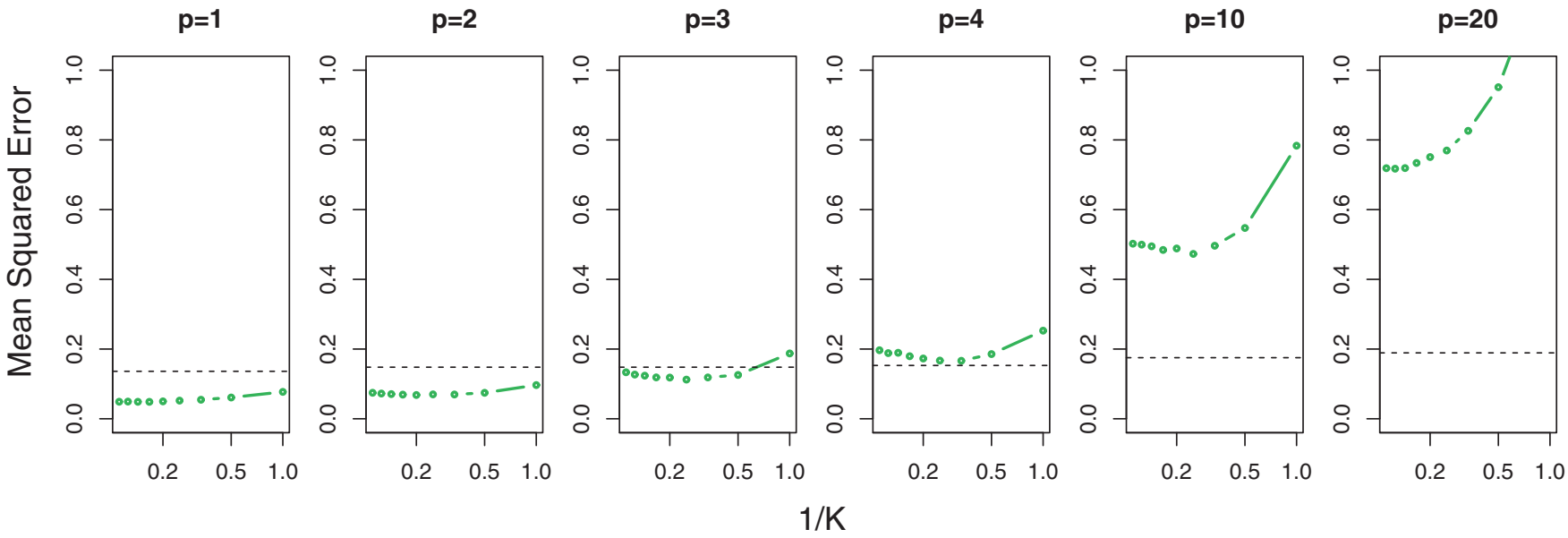


FIGURE 3.20. *Test MSE for linear regression (black dashed lines) and KNN (green curves) as the number of variables p increases. The true function is non–linear in the first variable, as in the lower panel in Figure 3.19, and does not depend on the additional variables. The performance of linear regression deteriorates slowly in the presence of these additional noise variables, whereas KNN's performance degrades much more quickly as p increases.*

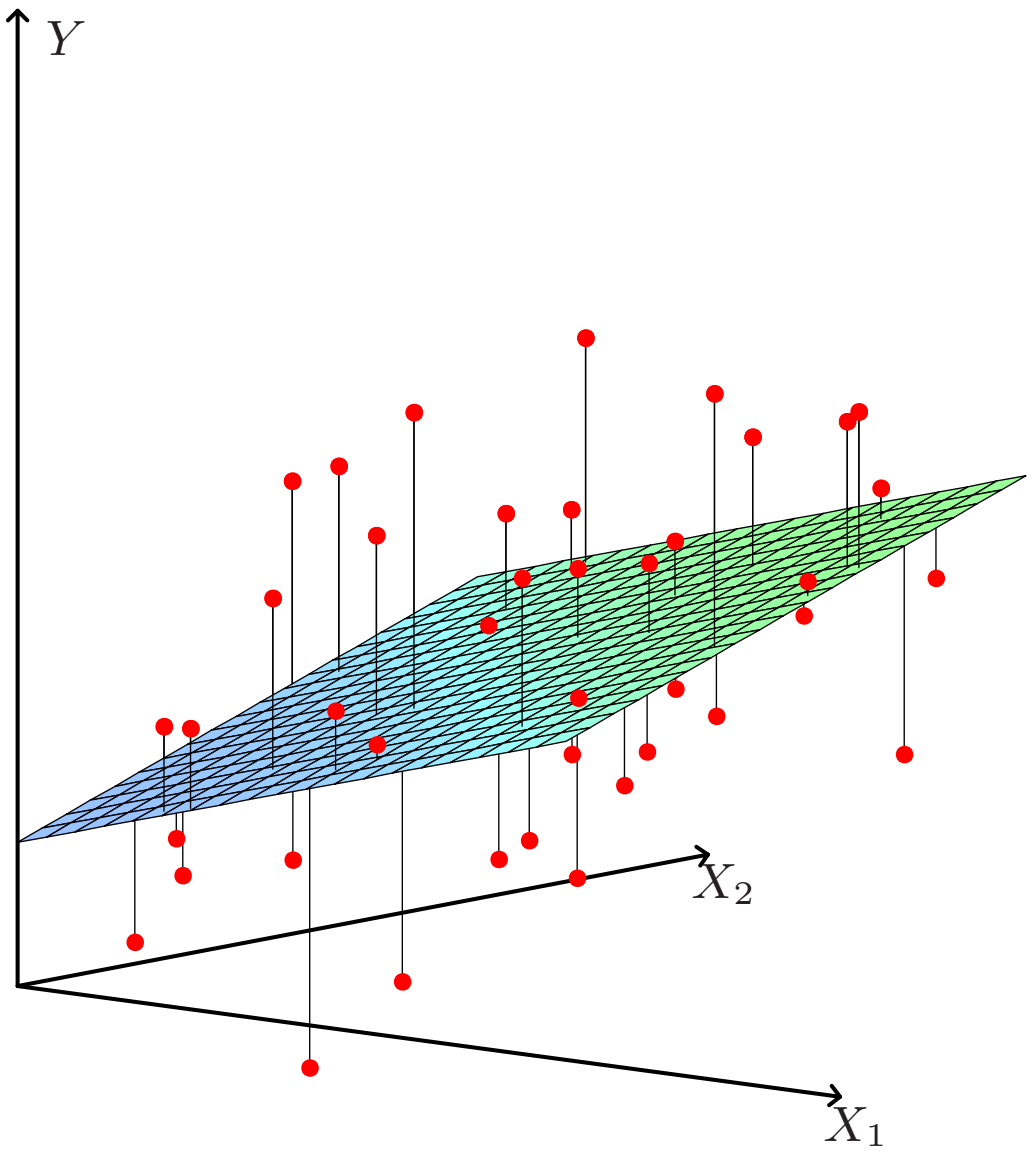


FIGURE 3.1. *Linear least squares fitting with* $X \in \mathbb{R}^2$. *We seek the linear function of* X *that minimizes the sum of squared residuals from* Y.

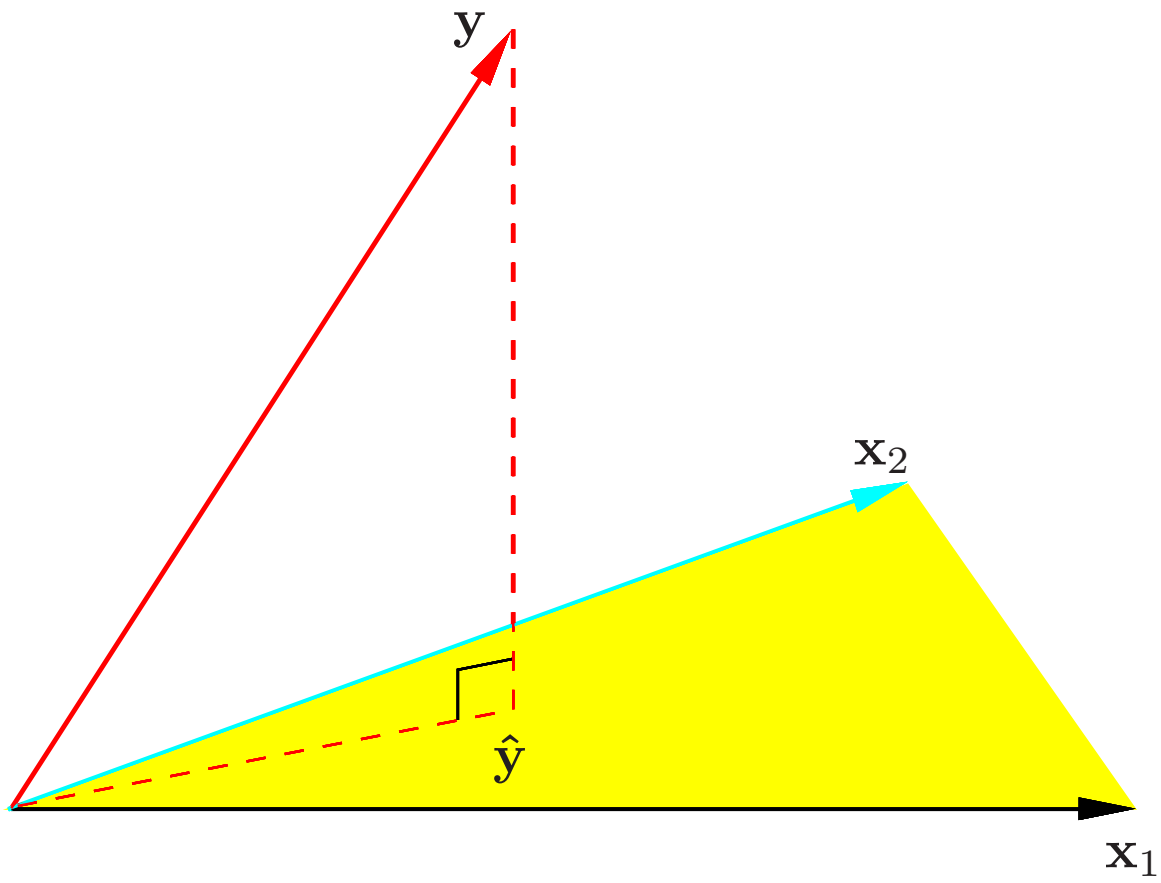


FIGURE 3.2. *The N-dimensional geometry of least squares regression with two predictors. The outcome vector $\mathbf{y}$ is orthogonally projected onto the hyperplane spanned by the input vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. The projection $\hat{\mathbf{y}}$ represents the vector of the least squares predictions*

chapter 4: lecture 7

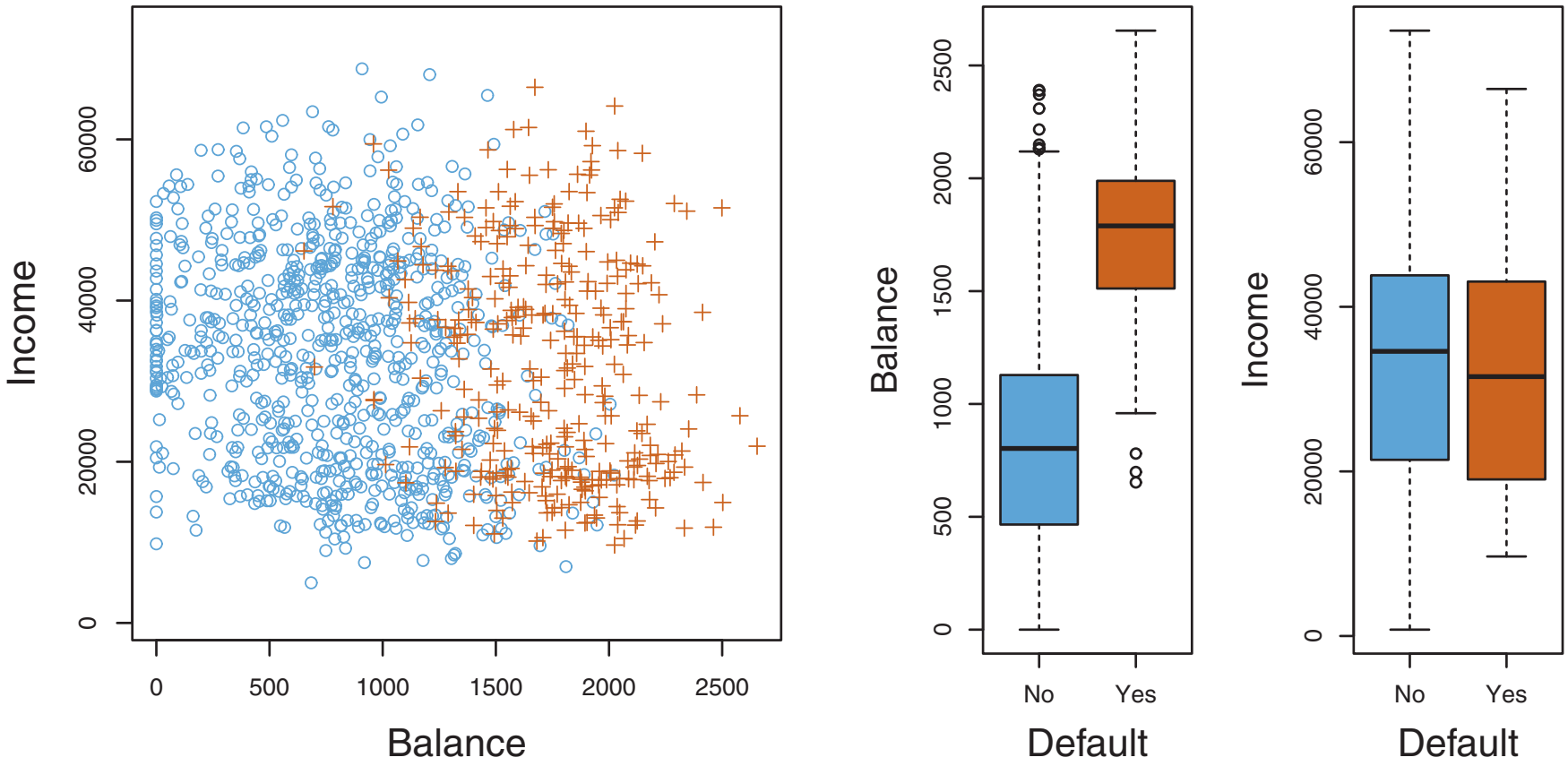


FIGURE 4.1. *The* `Default` *data set.* Left: *The annual incomes and monthly credit card balances of a number of individuals. The individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue.* Center: *Boxplots of* `balance` *as a function of* `default` *status.* Right: *Boxplots of* `income` *as a function of* `default` *status.*

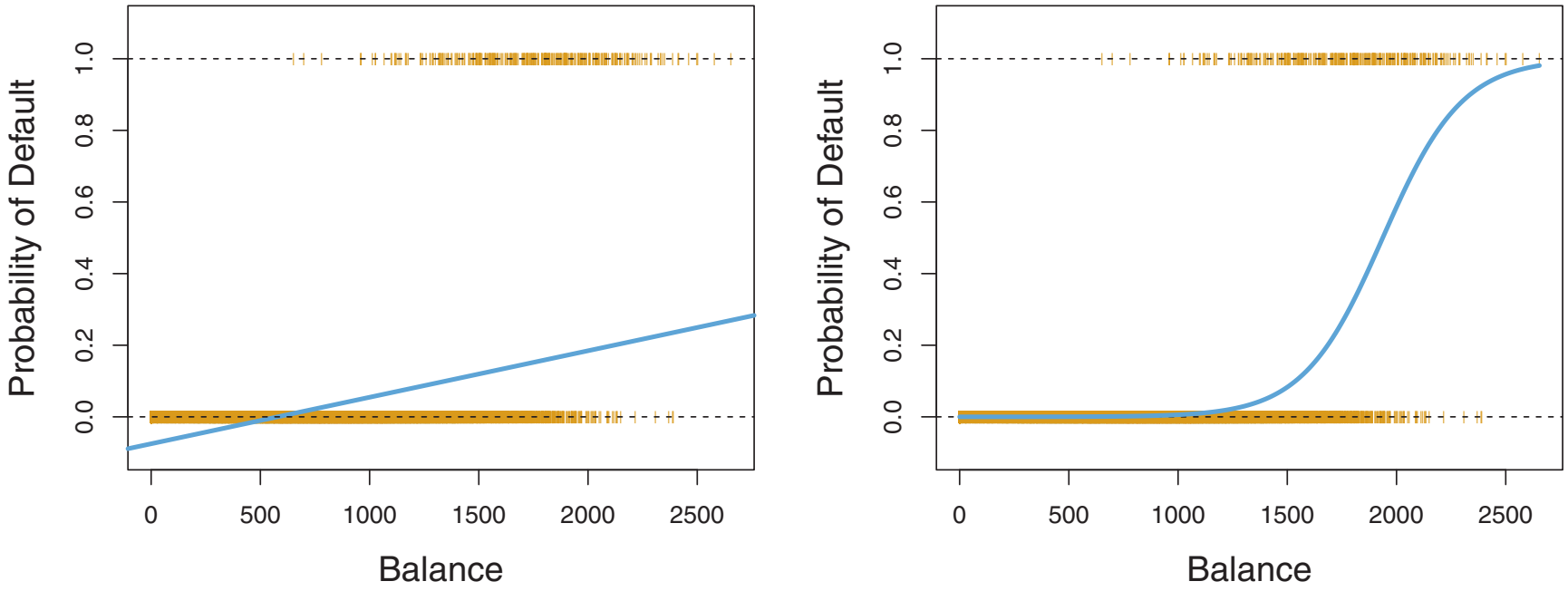


FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default` *(*`No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using logistic regression. All probabilities lie between* 0 *and* 1.

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	−10.6513	0.3612	−29.5	<0.0001
`balance`	0.0055	0.0002	24.9	<0.0001

TABLE 4.1. *For the* `Default` *data, estimated coefficients of the logistic regression model that predicts the probability of* `default` *using* `balance`*. A one-unit increase in* `balance` *is associated with an increase in the log odds of* `default` *by* 0.0055 *units.*

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	−3.5041	0.0707	−49.55	<0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

TABLE 4.2. *For the* `Default` *data, estimated coefficients of the logistic regression model that predicts the probability of* `default` *using student status. Student status is encoded as a dummy variable, with a value of* 1 *for a student and a value of* 0 *for a non-student, and represented by the variable* `student[Yes]` *in the table.*

	Coefficient	Std. error	Z-statistic	P-value
Intercept	−10.8690	0.4923	−22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	−0.6468	0.2362	−2.74	0.0062

TABLE 4.3. *For the* Default *data, estimated coefficients of the logistic regression model that predicts the probability of* default *using* balance, income, *and student status. Student status is encoded as a dummy variable* student[Yes], *with a value of* 1 *for a student and a value of* 0 *for a non-student. In fitting this model,* income *was measured in thousands of dollars.*

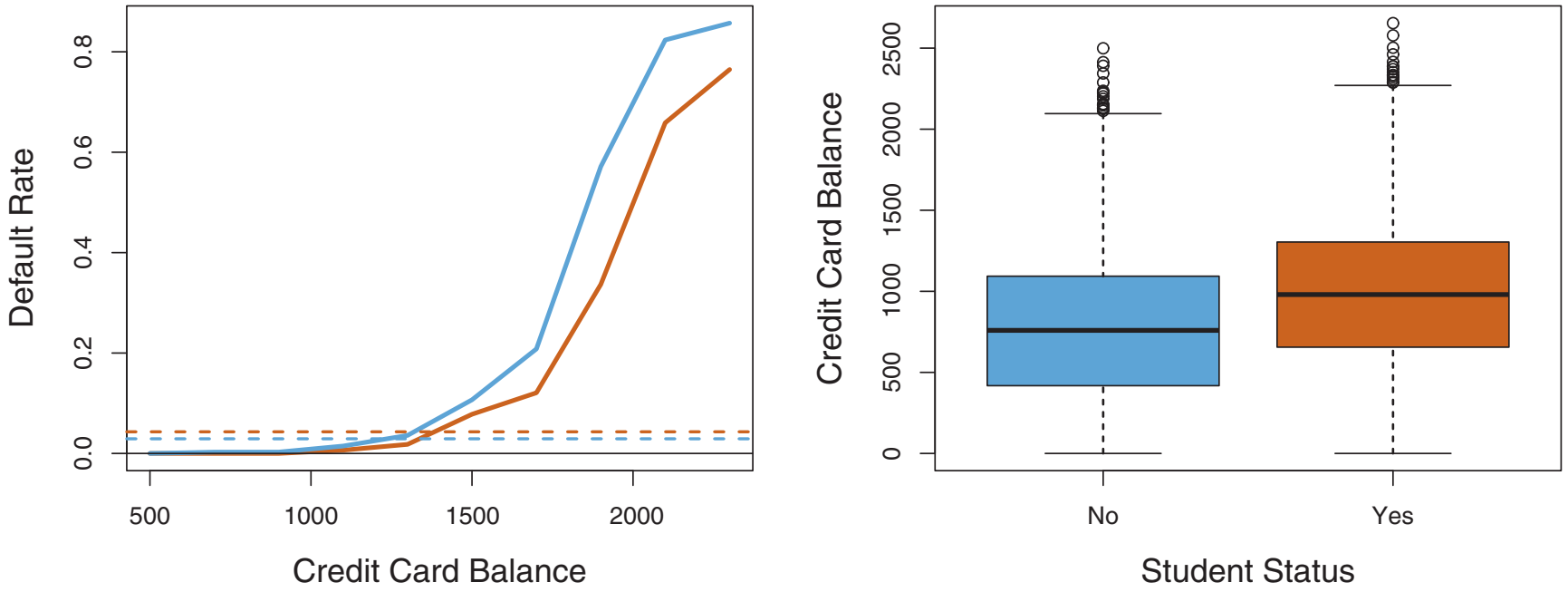


FIGURE 4.3. *Confounding in the* `Default` *data.* Left: *Default rates are shown for students (orange) and non-students (blue). The solid lines display default rate as a function of* `balance`*, while the horizontal broken lines display the overall default rates.* Right: *Boxplots of* `balance` *for students (orange) and non-students (blue) are shown.*

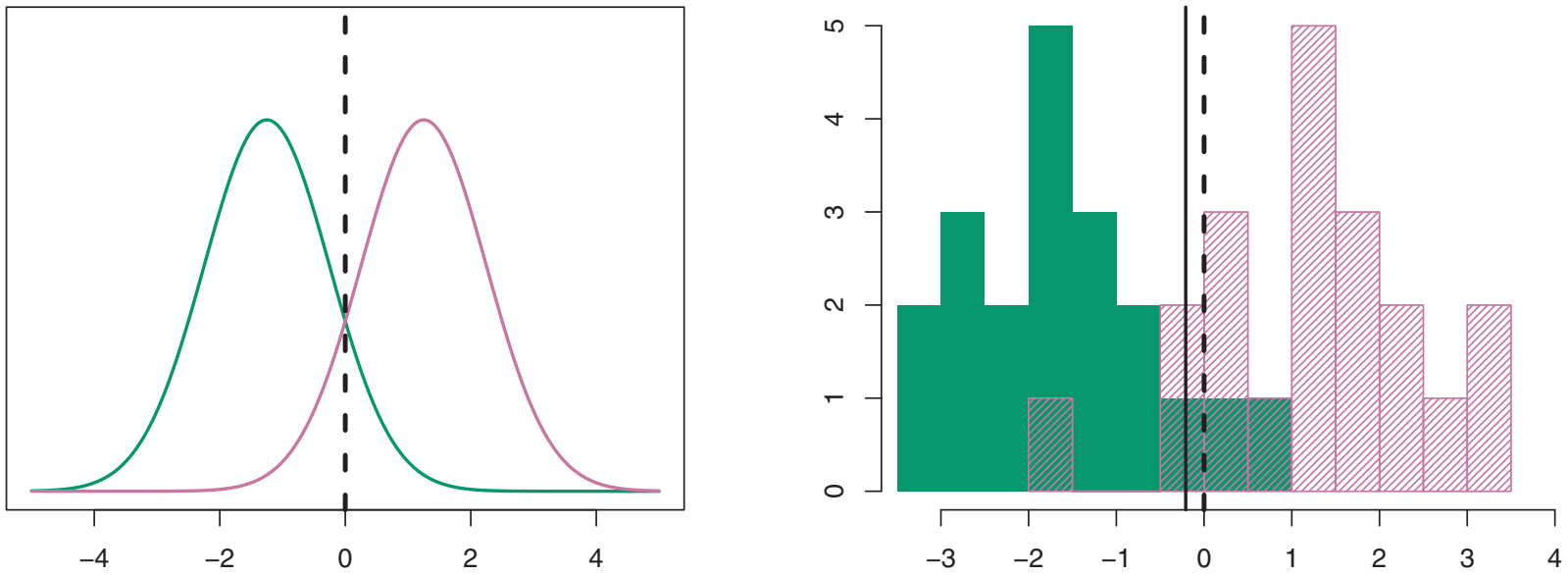


FIGURE 4.4. Left: *Two one-dimensional normal density functions are shown. The dashed vertical line represents the Bayes decision boundary.* Right: *20 observations were drawn from each of the two classes, and are shown as histograms. The Bayes decision boundary is again shown as a dashed vertical line. The solid vertical line represents the LDA decision boundary estimated from the training data.*

lecture 8

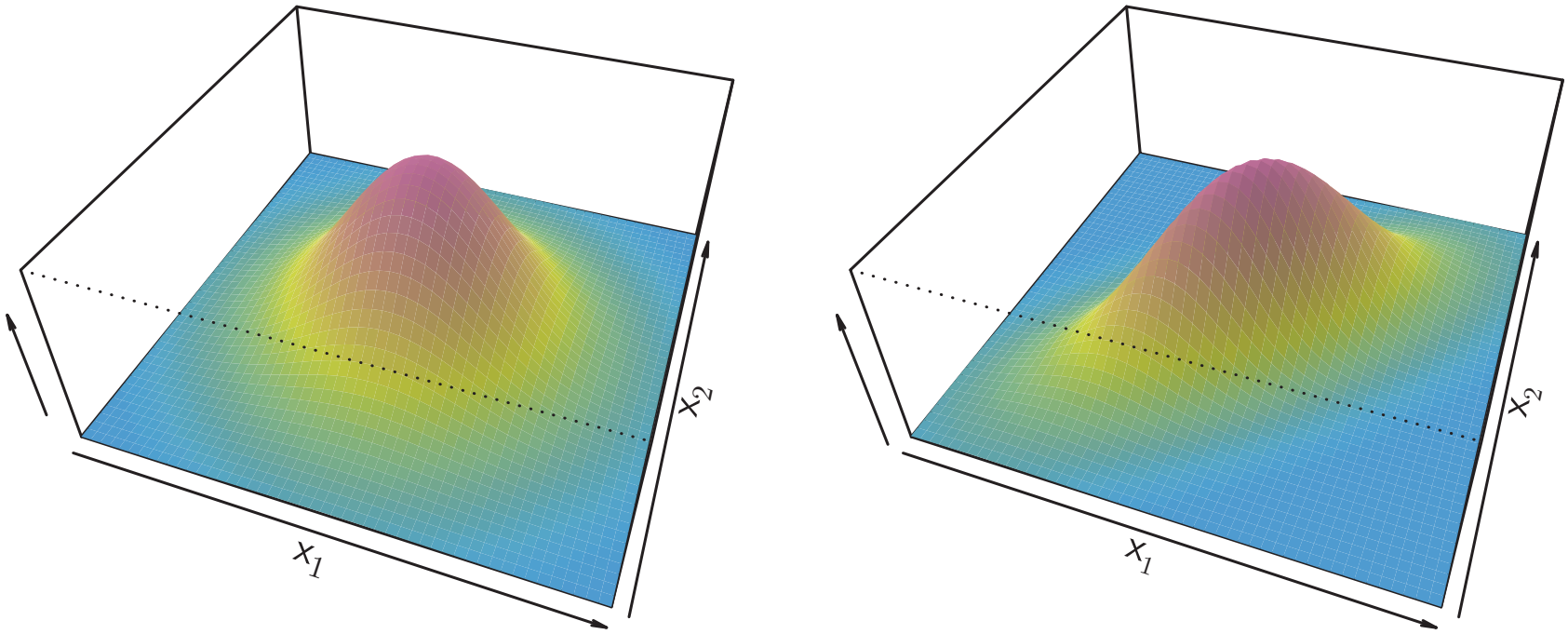


FIGURE 4.5. *Two multivariate Gaussian density functions are shown, with* $p = 2$. Left: *The two predictors are uncorrelated.* Right: *The two variables have a correlation of* 0.7.

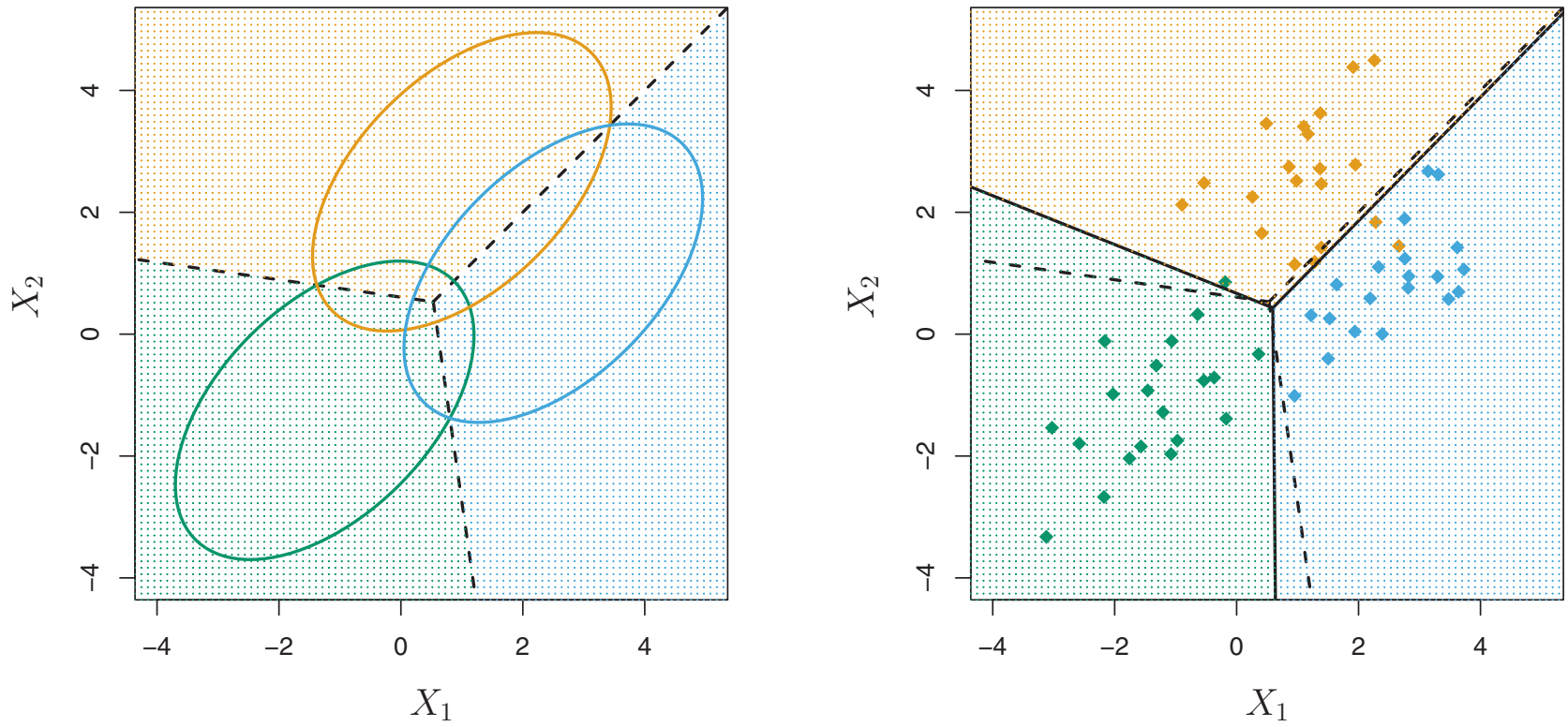


FIGURE 4.6. *An example with three classes. The observations from each class are drawn from a multivariate Gaussian distribution with $p = 2$, with a class-specific mean vector and a common covariance matrix.* Left: *Ellipses that contain* 95 % *of the probability for each of the three classes are shown. The dashed lines are the Bayes decision boundaries.* Right: 20 *observations were generated from each class, and the corresponding LDA decision boundaries are indicated using solid black lines. The Bayes decision boundaries are once again shown as dashed lines.*

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

TABLE 4.4. *A confusion matrix compares the LDA predictions to the true default statuses for the* 10,000 *training observations in the* `Default` *data set. Elements on the diagonal of the matrix represent individuals whose default statuses were correctly predicted, while off-diagonal elements represent individuals that were misclassified. LDA made incorrect predictions for* 23 *individuals who did not default and for* 252 *individuals who did default.*

		True default status		
		No	Yes	Total
Predicted default status	No	9,432	138	9,570
	Yes	235	195	430
	Total	9,667	333	10,000

TABLE 4.5. *A confusion matrix compares the LDA predictions to the true default statuses for the* 10,000 *training observations in the* `Default` *data set, using a modified threshold value that predicts default for any individuals whose posterior default probability exceeds* 20 %.

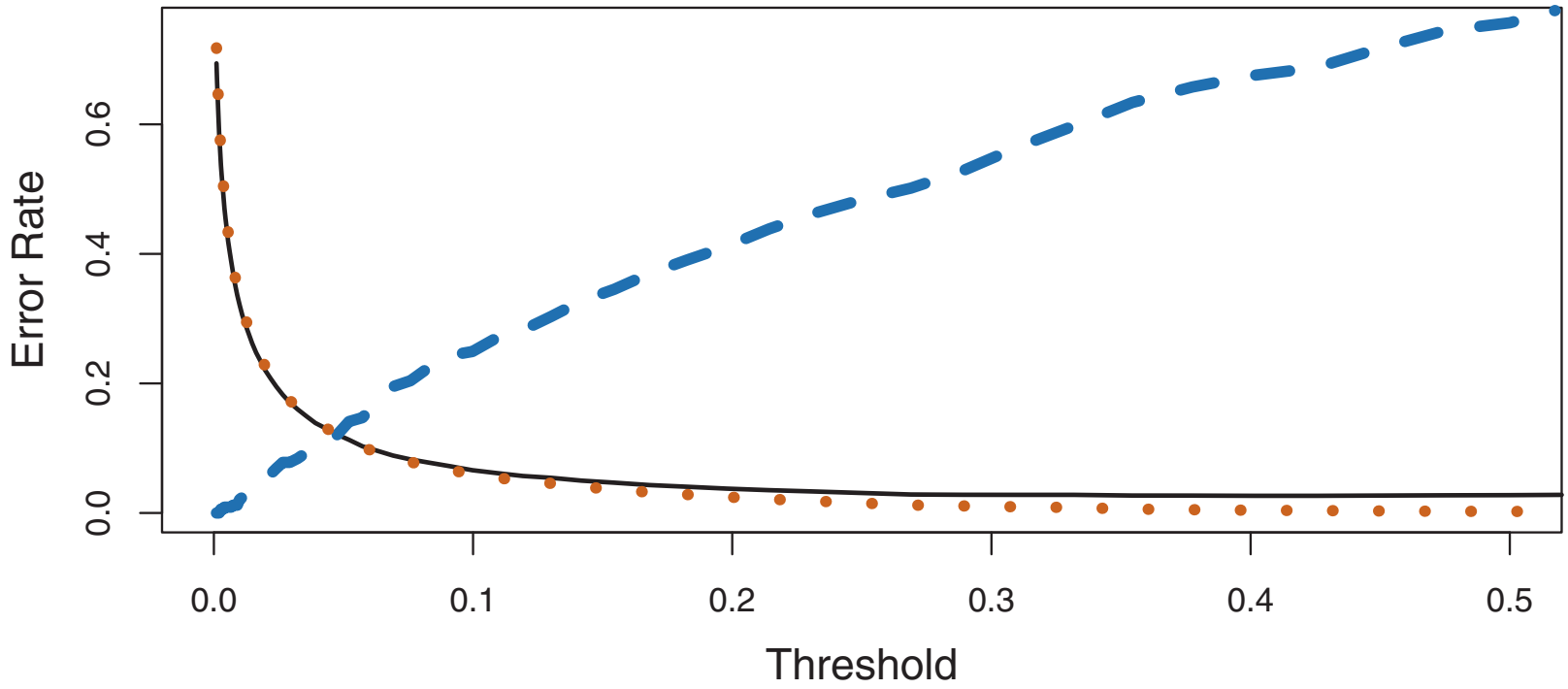


FIGURE 4.7. *For the* `Default` *data set, error rates are shown as a function of the threshold value for the posterior probability that is used to perform the assignment. The black solid line displays the overall error rate. The blue dashed line represents the fraction of defaulting customers that are incorrectly classified, and the orange dotted line indicates the fraction of errors among the non-defaulting customers.*

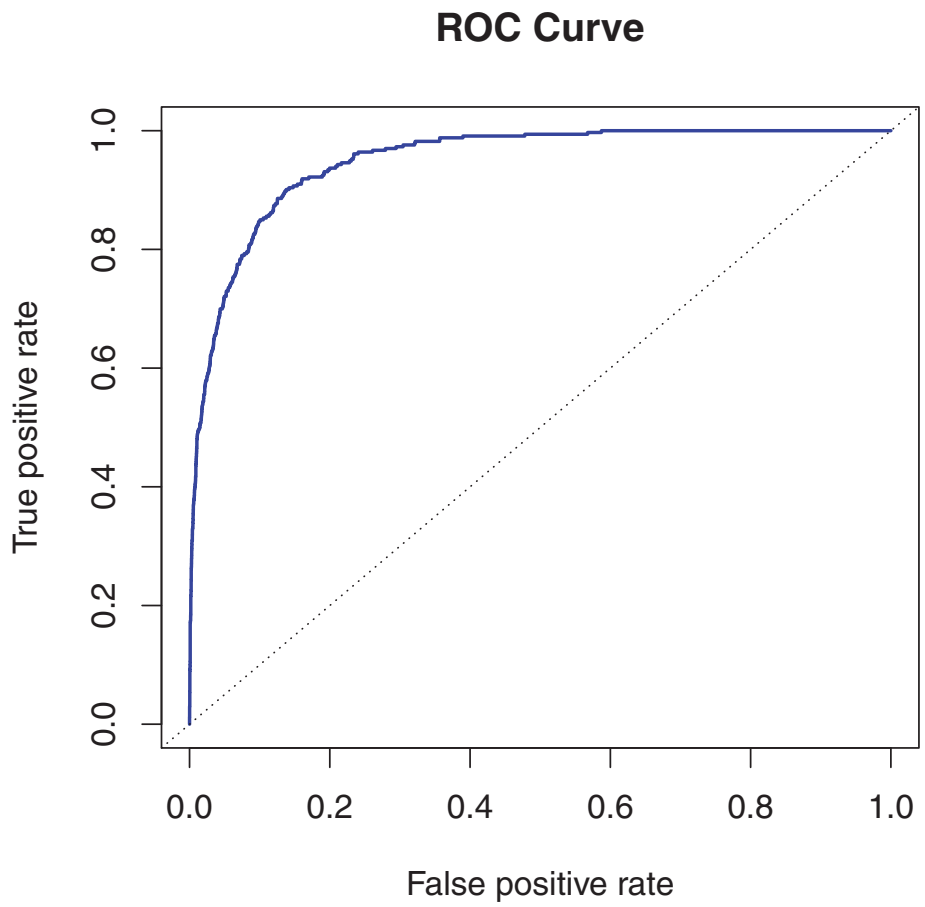


FIGURE 4.8. *A ROC curve for the LDA classifier on the* `Default` *data. It traces out two types of error as we vary the threshold value for the posterior probability of default. The actual thresholds are not shown. The true positive rate is the sensitivity: the fraction of defaulters that are correctly identified, using a given threshold value. The false positive rate is 1-specificity: the fraction of non-defaulters that we classify incorrectly as defaulters, using that same threshold value. The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The dotted line represents the "no information" classifier; this is what we would expect if student status and credit card balance are not associated with probability of default.*

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

TABLE 4.7. *Important measures for classification and diagnostic testing, derived from quantities in Table 4.6.*

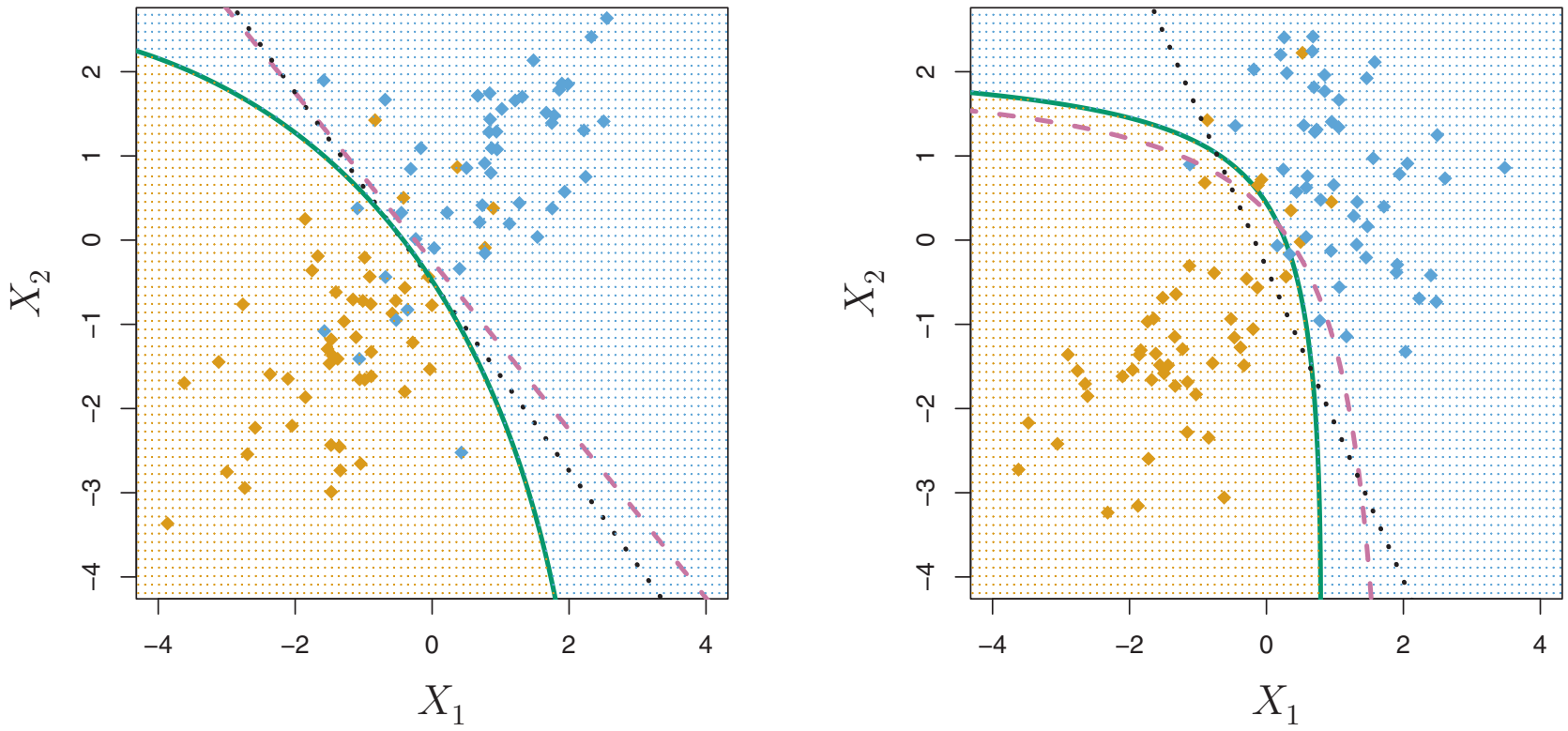


FIGURE 4.9. Left: *The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem with* $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$. *The shading indicates the QDA decision rule. Since the Bayes decision boundary is linear, it is more accurately approximated by LDA than by QDA.* Right: *Details are as given in the left-hand panel, except that* $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$. *Since the Bayes decision boundary is non-linear, it is more accurately approximated by QDA than by LDA.*

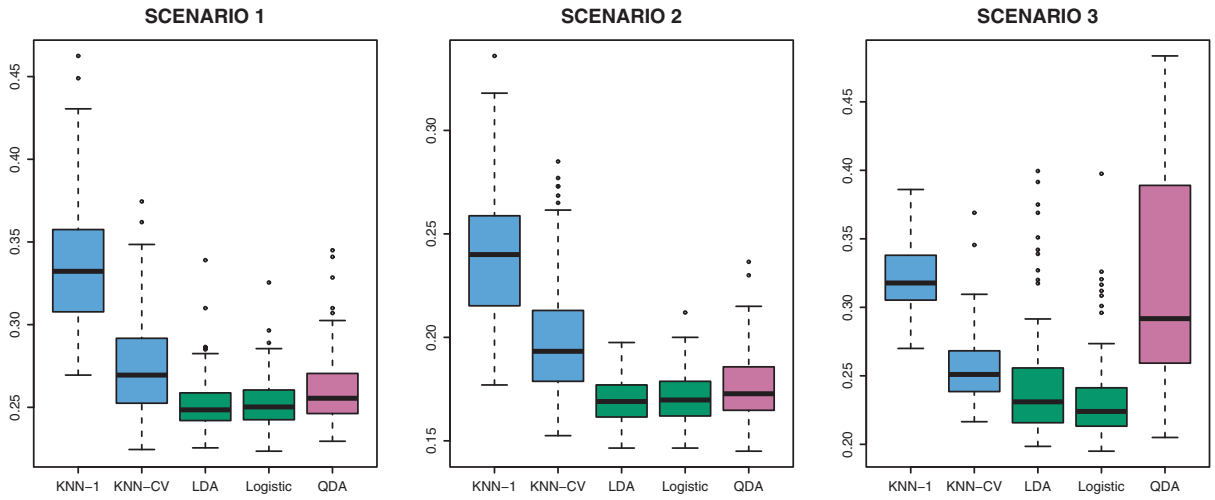


FIGURE 4.10. *Boxplots of the test error rates for each of the linear scenarios described in the main text.*

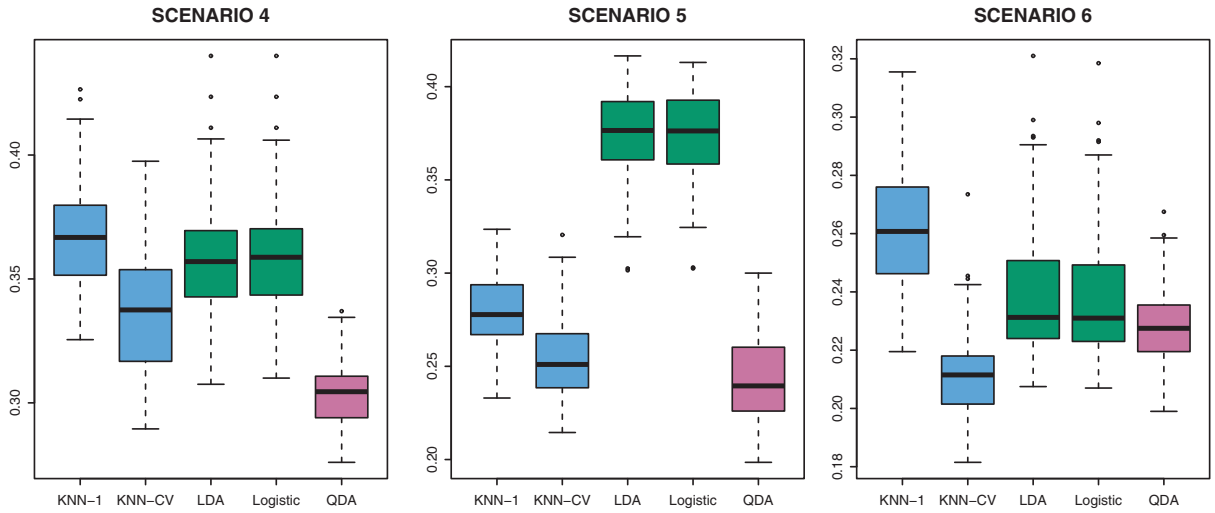


FIGURE 4.11. *Boxplots of the test error rates for each of the non-linear scenarios described in the main text.*

lecture 9

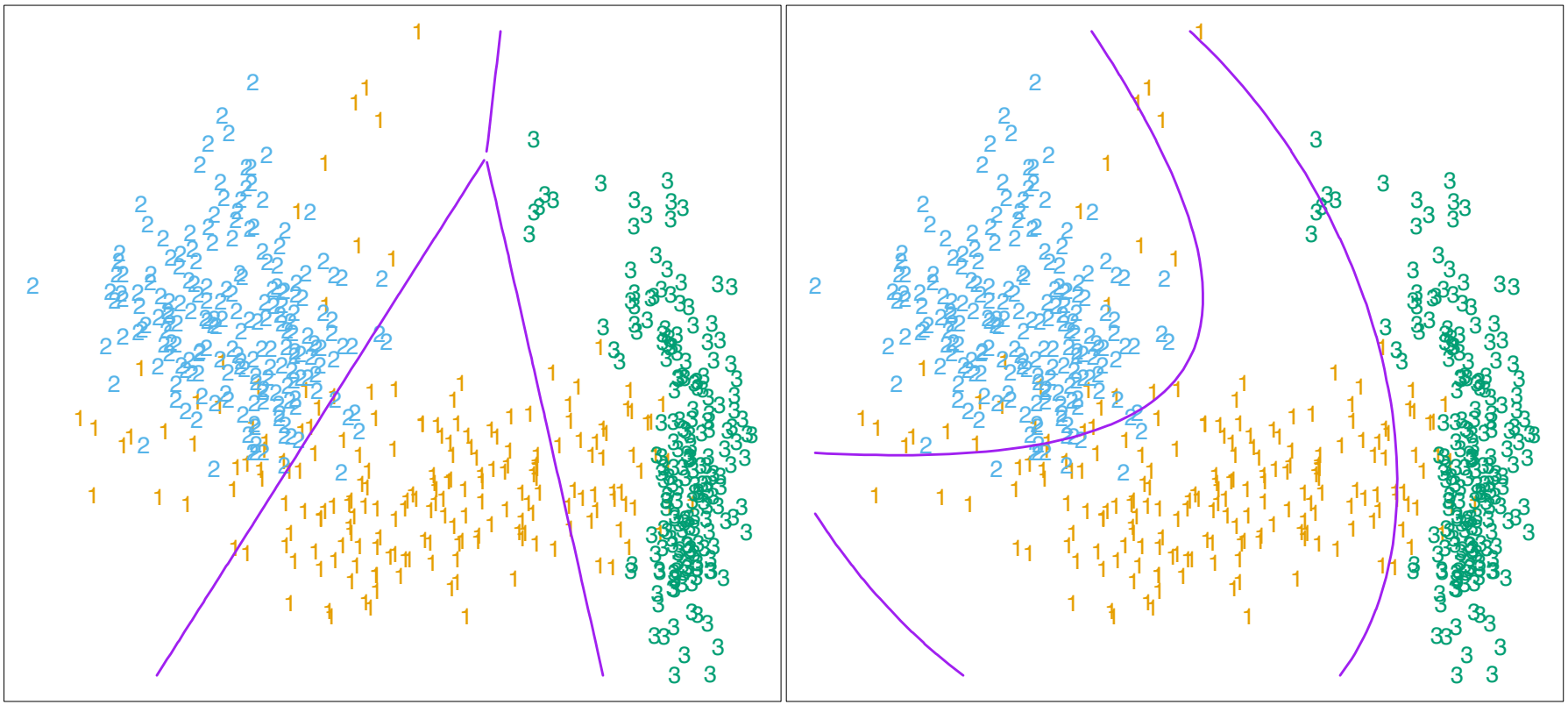


FIGURE 4.1. *The left plot shows some data from three classes, with linear decision boundaries found by linear discriminant analysis. The right plot shows quadratic decision boundaries. These were obtained by finding linear boundaries in the five-dimensional space* $X_1, X_2, X_1X_2, X_1^2, X_2^2$. *Linear inequalities in this space are quadratic inequalities in the original space.*

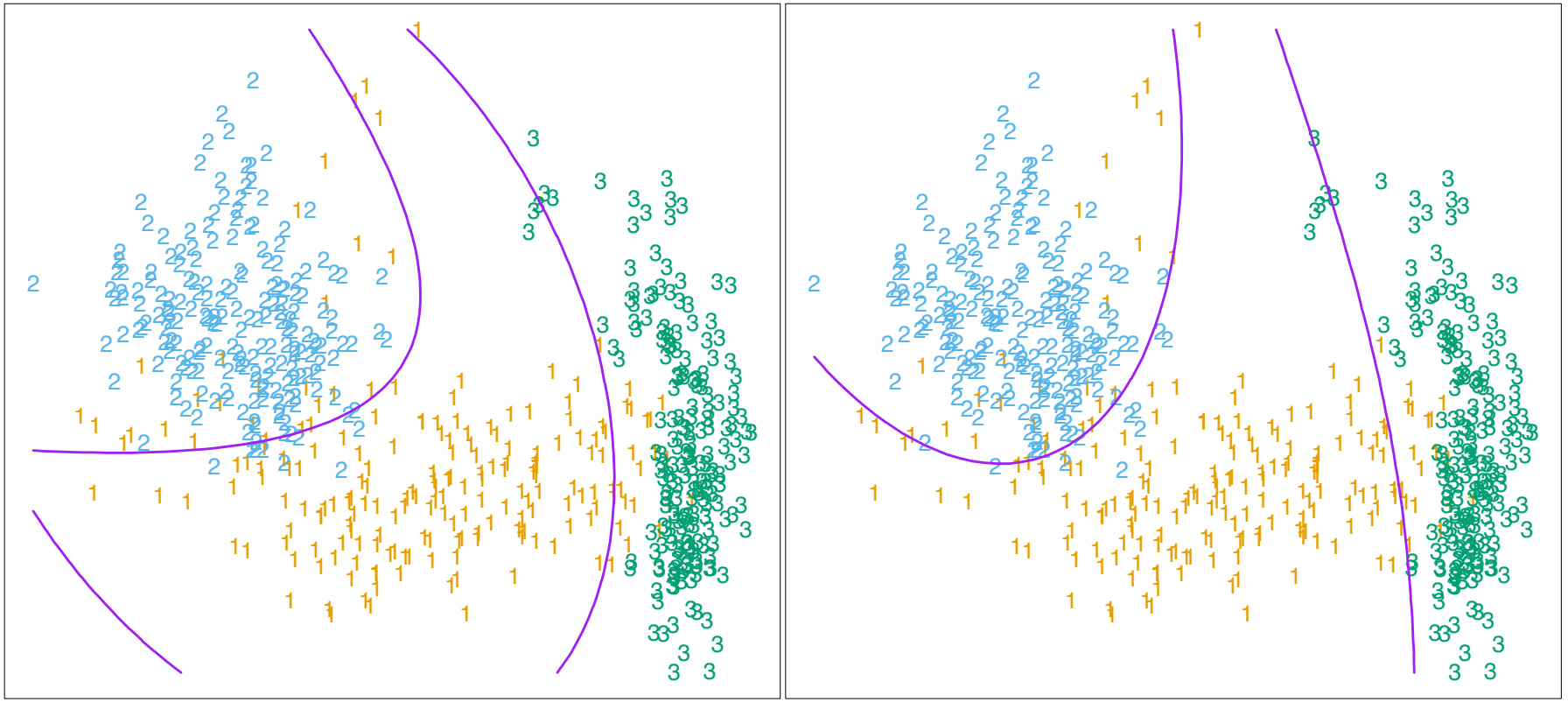


FIGURE 4.6. *Two methods for fitting quadratic boundaries. The left plot shows the quadratic decision boundaries for the data in Figure 4.1 (obtained using LDA in the five-dimensional space $X_1, X_2, X_1X_2, X_1^2, X_2^2$). The right plot shows the quadratic decision boundaries found by QDA. The differences are small, as is usually the case.*

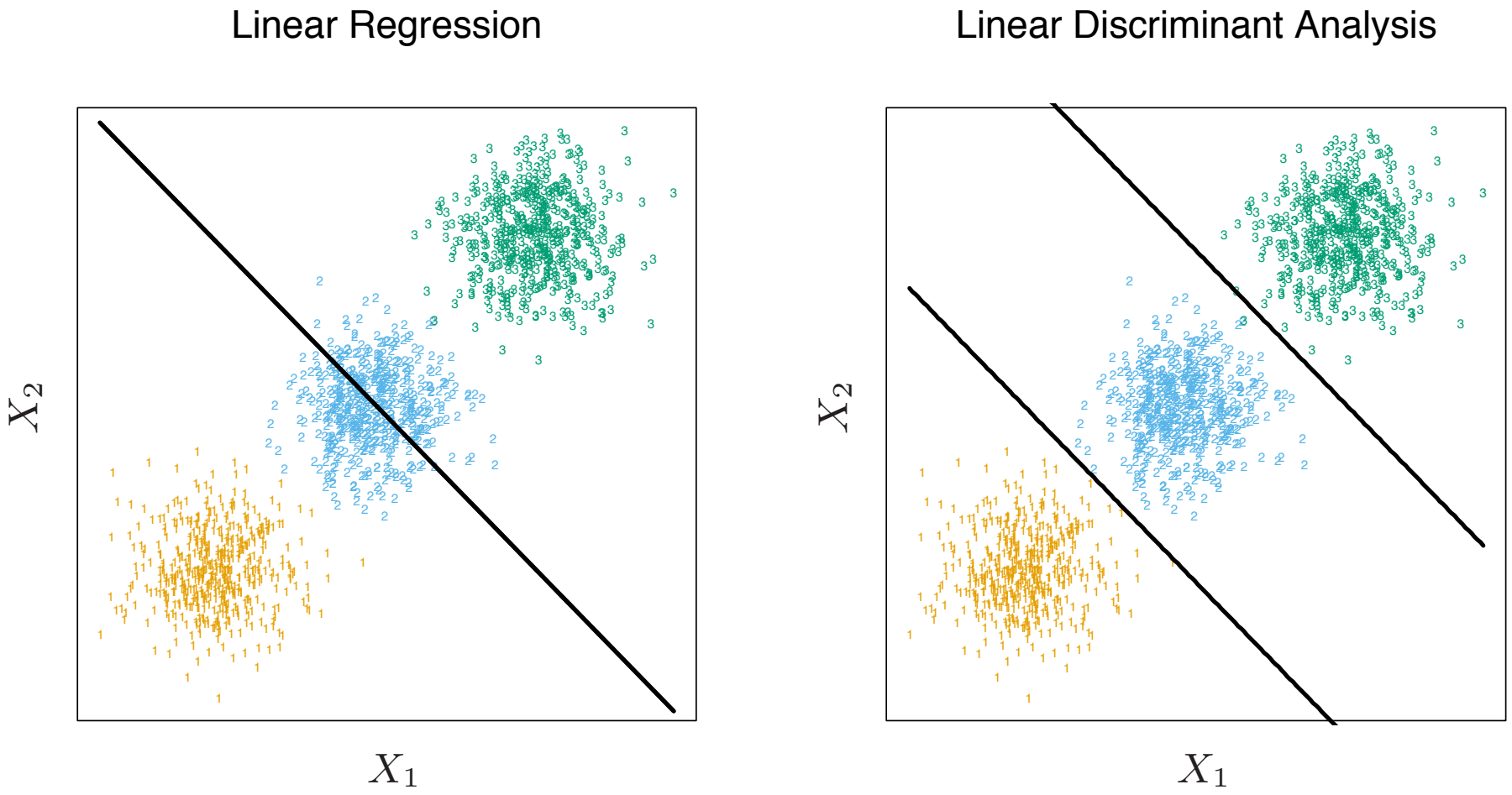


FIGURE 4.2. *The data come from three classes in* $\mathbb{R}^2$ *and are easily separated by linear decision boundaries. The right plot shows the boundaries found by linear discriminant analysis. The left plot shows the boundaries found by linear regression of the indicator response variables. The middle class is completely masked (never dominates).*

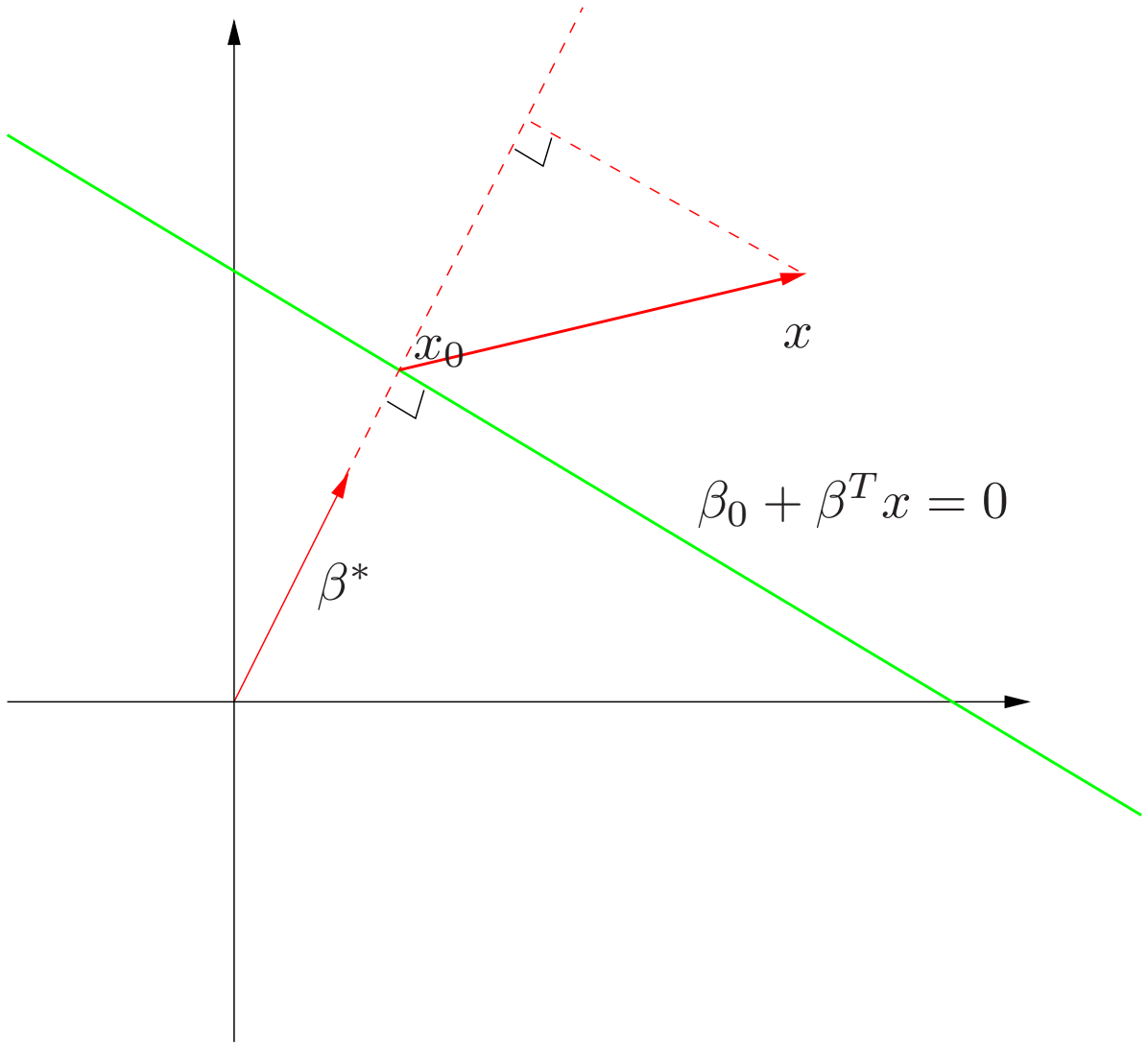


FIGURE 4.15. *The linear algebra of a hyperplane (affine set).*

chapter 5: resampling methods

- drawing samples from a training set and refitting a model
- computationally expensive
- two main approaches:
  - cross-validation: estimate test error
    - model assessment: evaluate model's performance
    - model selection: select proper level of flexibility
  - bootstrap: provide a measure of accuracy of a parameter estimate

- test error: average error in predicting the response to a measurement not used to train the model.
- test error estimate: hold out a subset of training data from the fitting process and evaluate the model of the held out data.
- available set of observations=training set + validation set (hold out set)

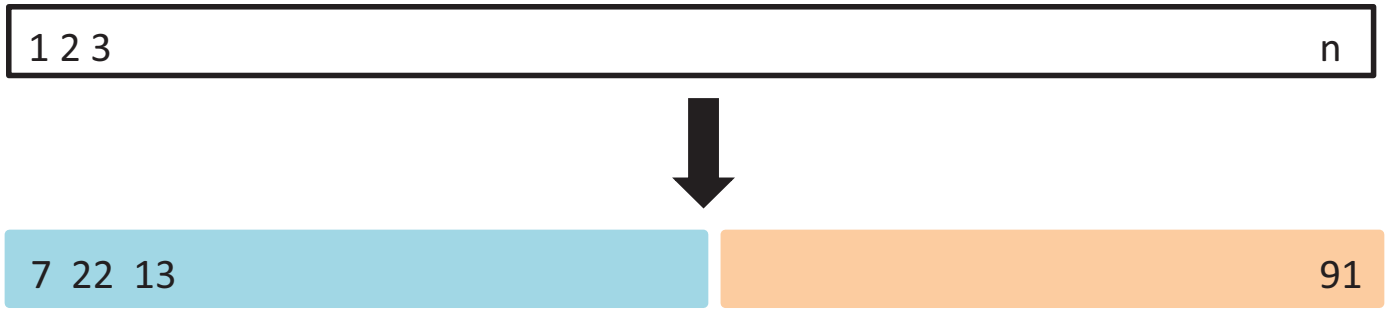


FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

potential drawbacks of the validation set approach:

- splitting is random $\rightarrow$ variability!
- only a subset of the data used to train $\rightarrow$ less data $\rightarrow$ model's performance decreases $\rightarrow$ validation set error rate overestimates the test error rate.

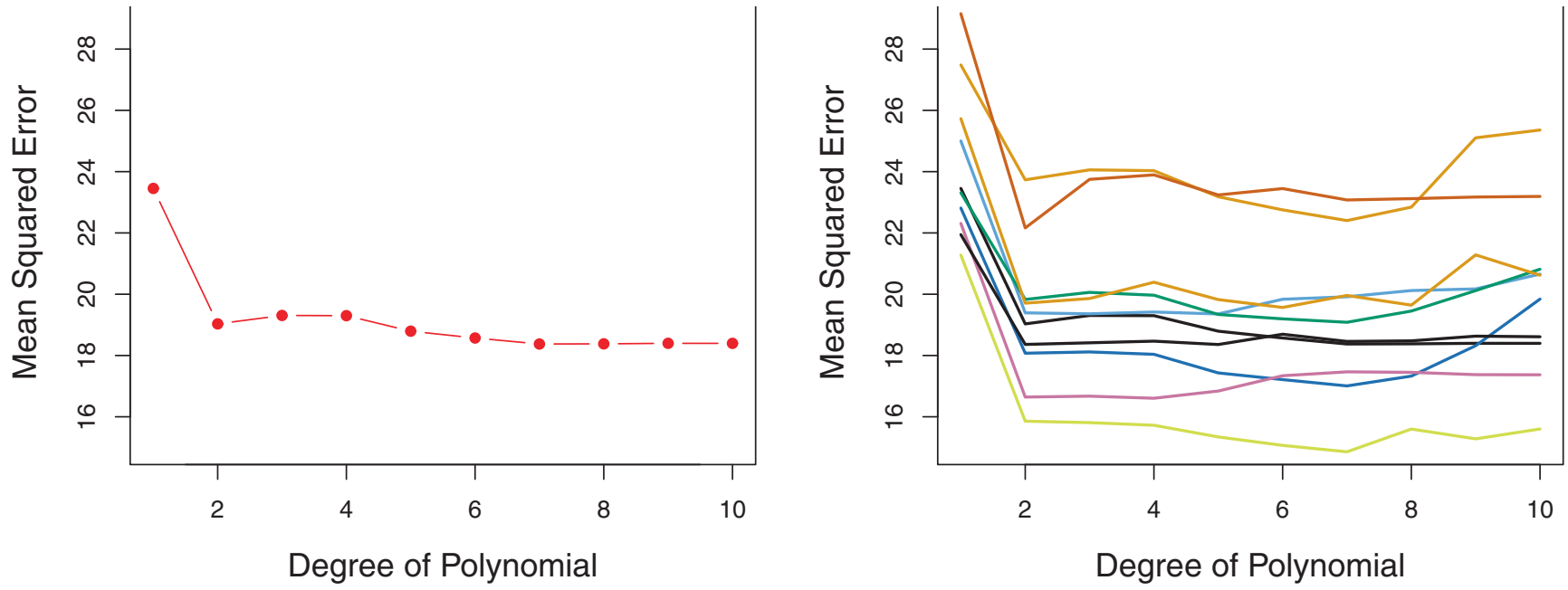


FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

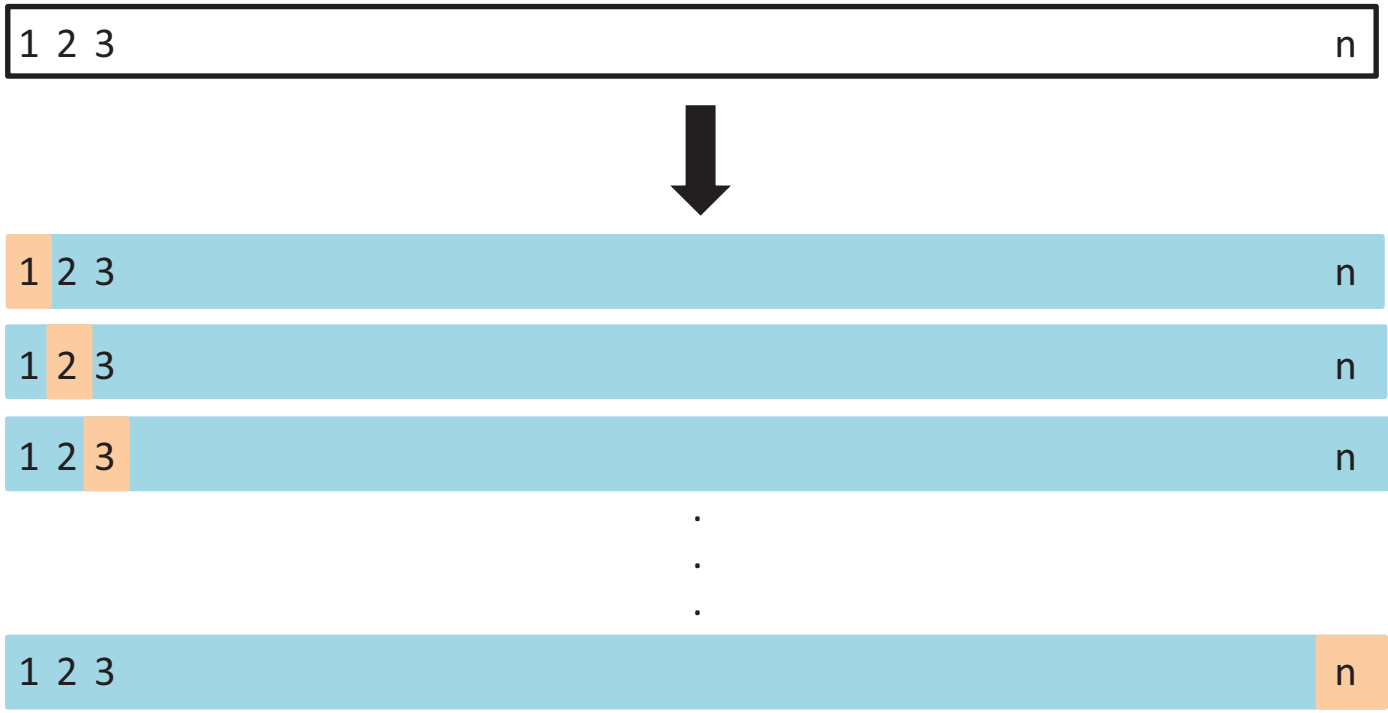


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

- $data = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- $data^{(-i)} = \{(x_1, y_1), \ldots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \ldots, (x_n, y_n)\}$
- $\hat{y}_i^{(-i)} =$ prediction for x_i using $data^{(-i)}$
- $CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i^{(-i)})^2$

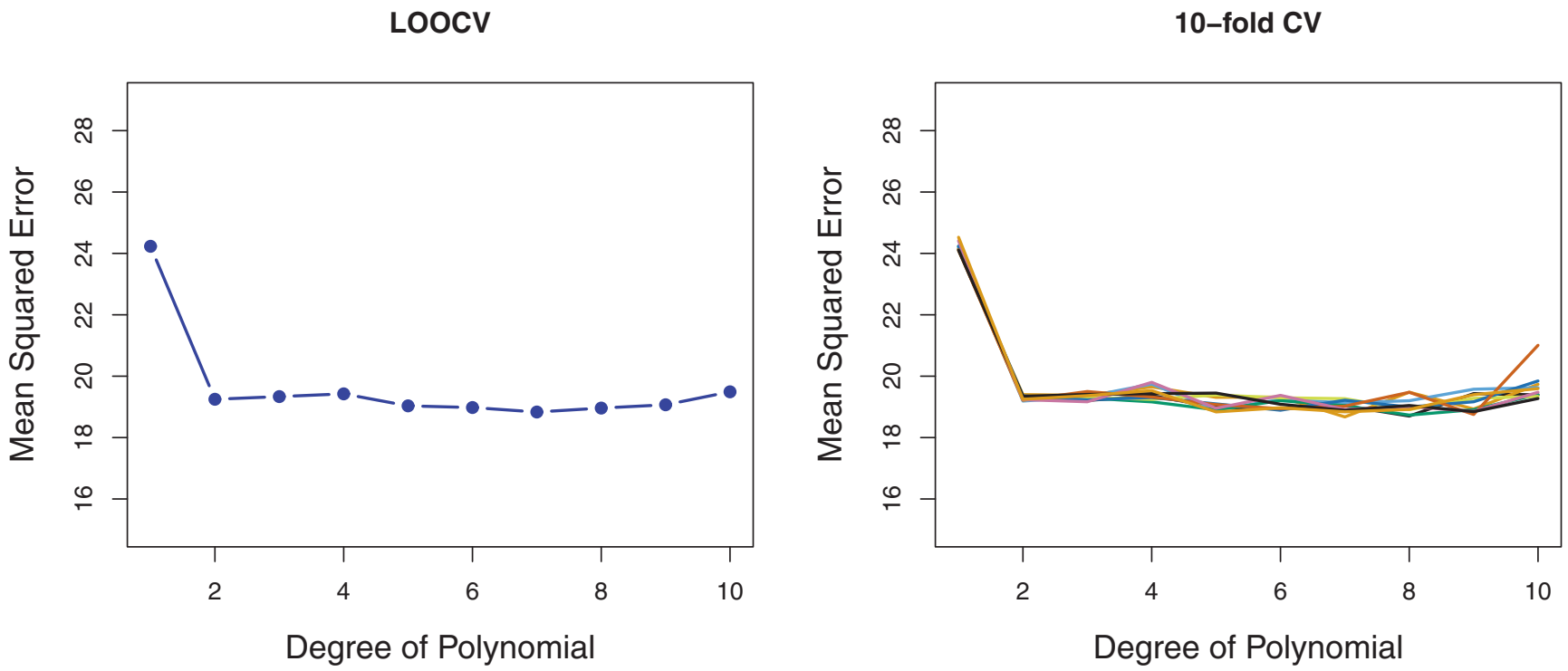


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10-*fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

- LOOCV: in general can computationally infeasible
- LOOCV: easy to compute for linear regression:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$

- $h_i = i$th diagonal of $H = X(X^TX)^{-1}X^T$ (hat matrix)
- h_i =leverage= amount that an observation influences its own fit.
- residuals for high-leverage points are inflated

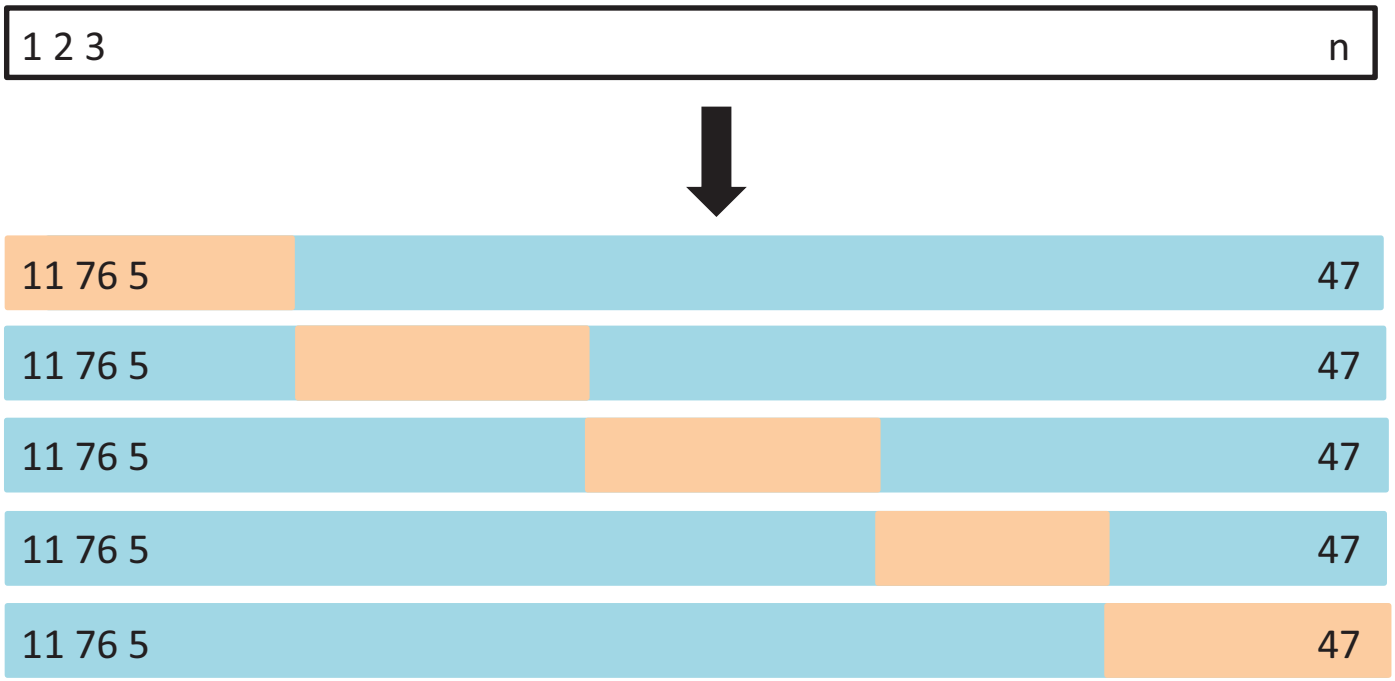


FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

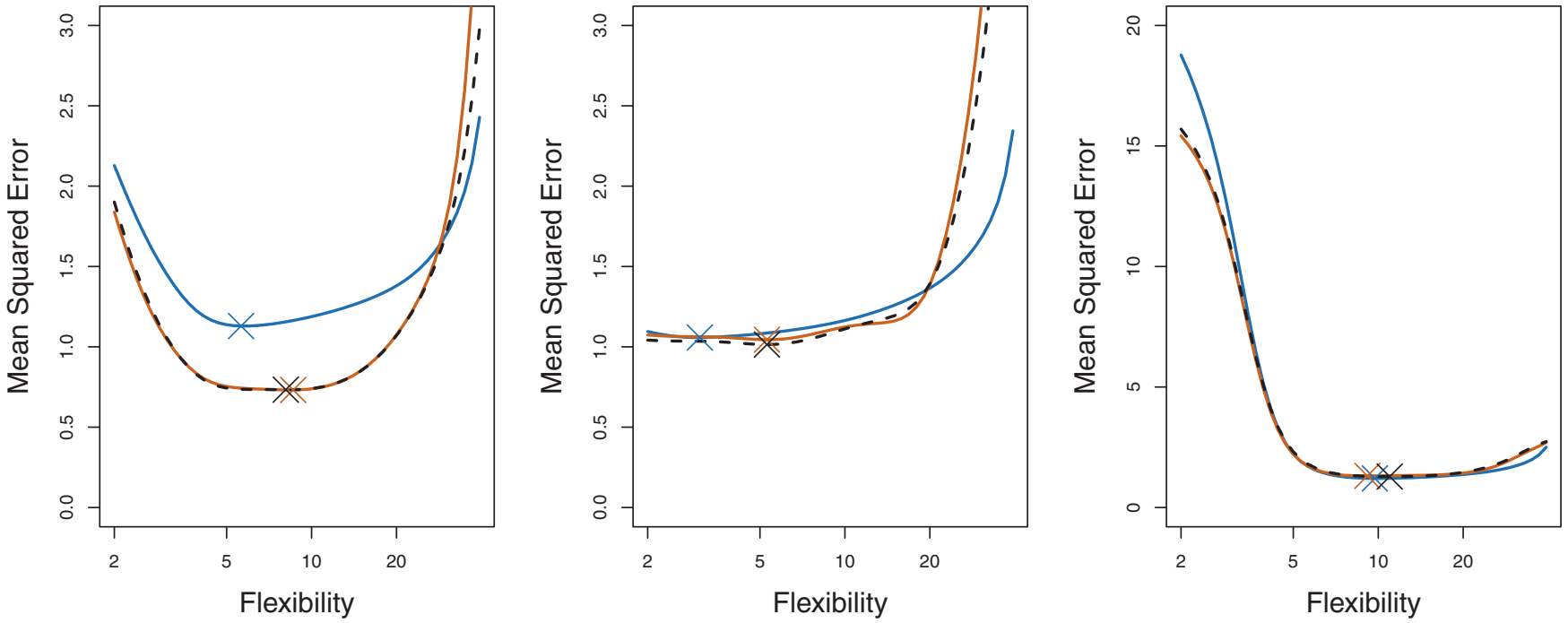


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9 (*left*), 2.10 (*center*), and 2.11 (*right*). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the* 10*-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.*

- model assessment: predictive power: actual estimate of test MSE is of interest
- CV tends to underestimate test MSE (in the moderate flexibility regime)
- model selection: location of the minimum point in the estimated test MSE is of interest
- CV vs. test MSE: minimizing flexibilities are close

cross-validation on classification problems:

- $CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq y_i^{(-i)})$

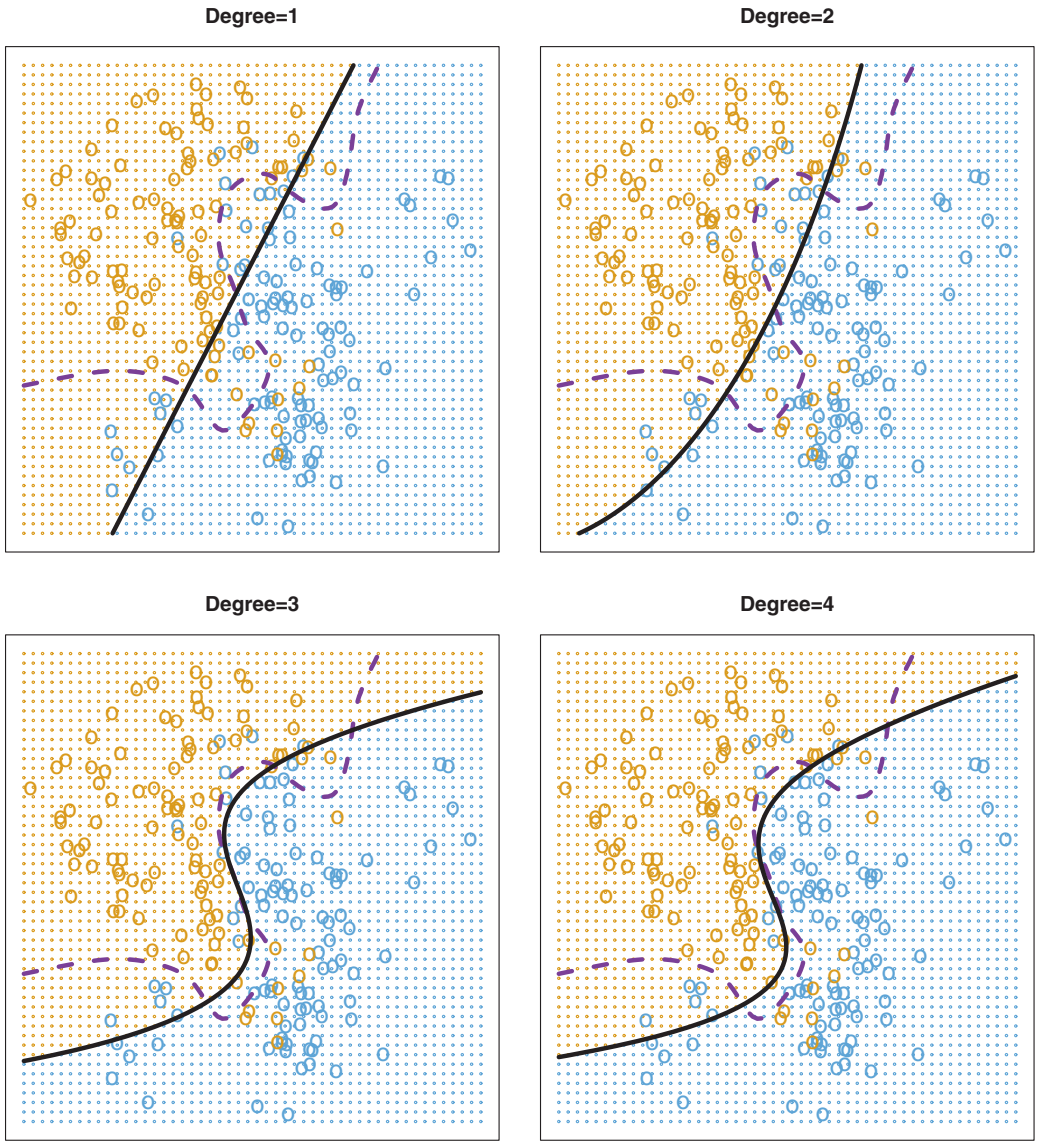


FIGURE 5.7. *Logistic regression fits on the two-dimensional classification data displayed in Figure 2.13. The Bayes decision boundary is represented using a purple dashed line. Estimated decision boundaries from linear, quadratic, cubic and quartic (degrees 1–4) logistic regressions are displayed in black. The test error rates for the four logistic regression fits are respectively* 0.201, 0.197, 0.160, *and* 0.162, *while the Bayes error rate is* 0.133.

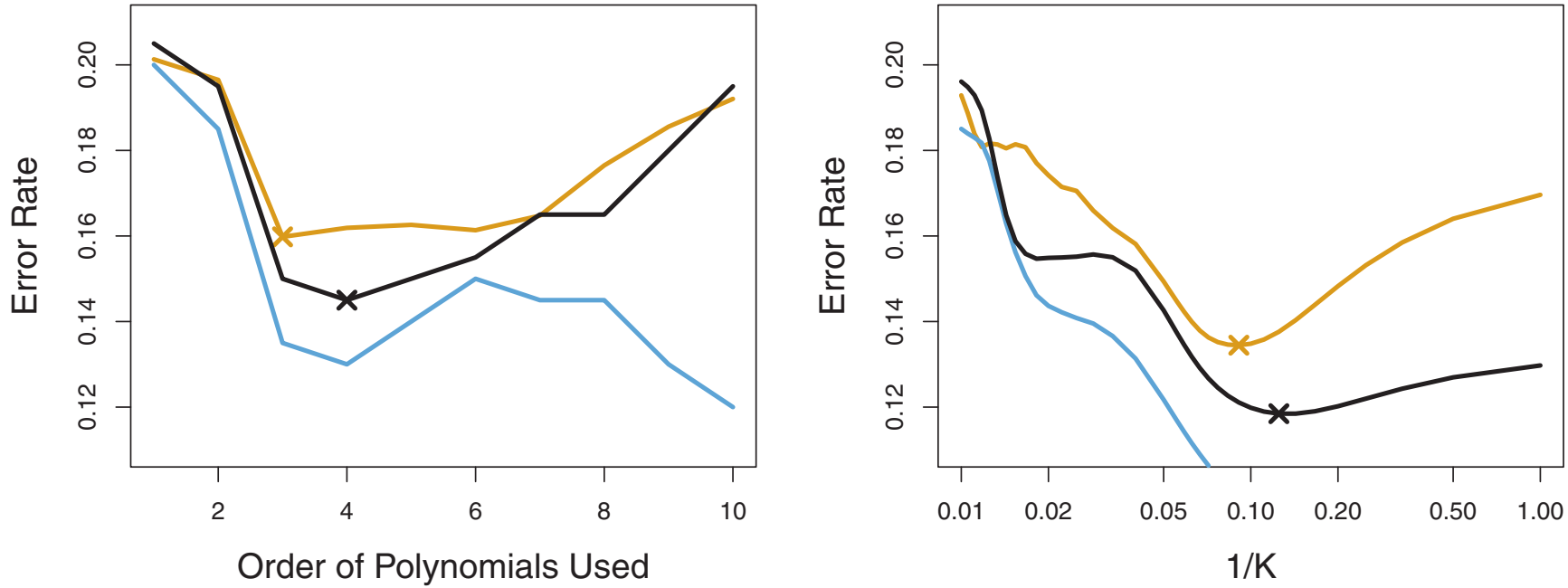


FIGURE 5.8. *Test error (brown), training error (blue), and* 10*-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the* x*-axis.* Right: *The KNN classifier with different values of* K*, the number of neighbors used in the KNN classifier.*

- CV tends to underestimate test MSE (in the moderate flexibility regime)

bootstrap:

- quantify uncertainty associated with a given estimator or statistical learning method
- example: estimate standard error of coefficients from a linear regression fit

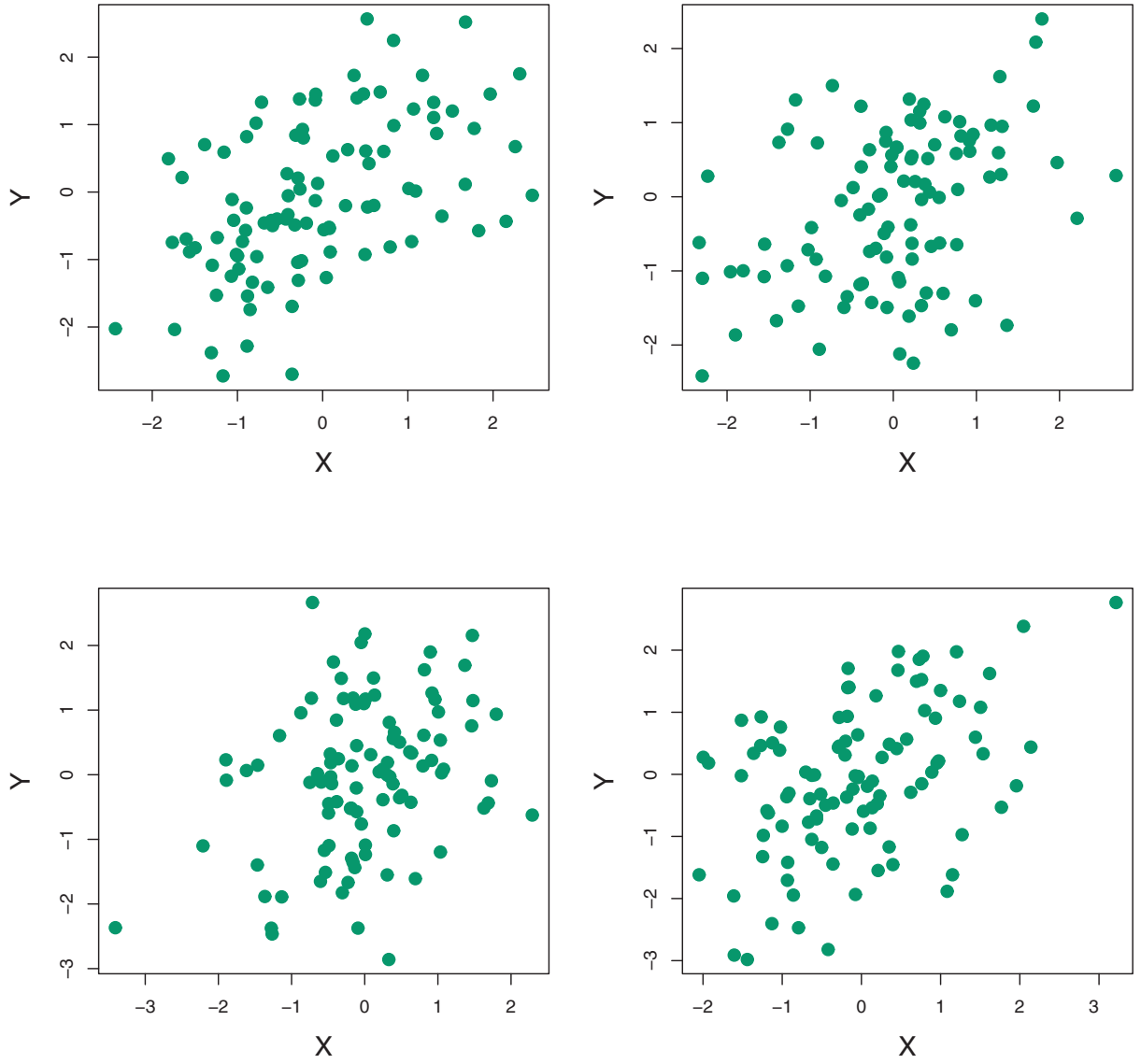


FIGURE 5.9. *Each panel displays* 100 *simulated returns for investments* X *and* Y*. From left to right and top to bottom, the resulting estimates for* α *are* 0.576, 0.532, 0.657, *and* 0.651.

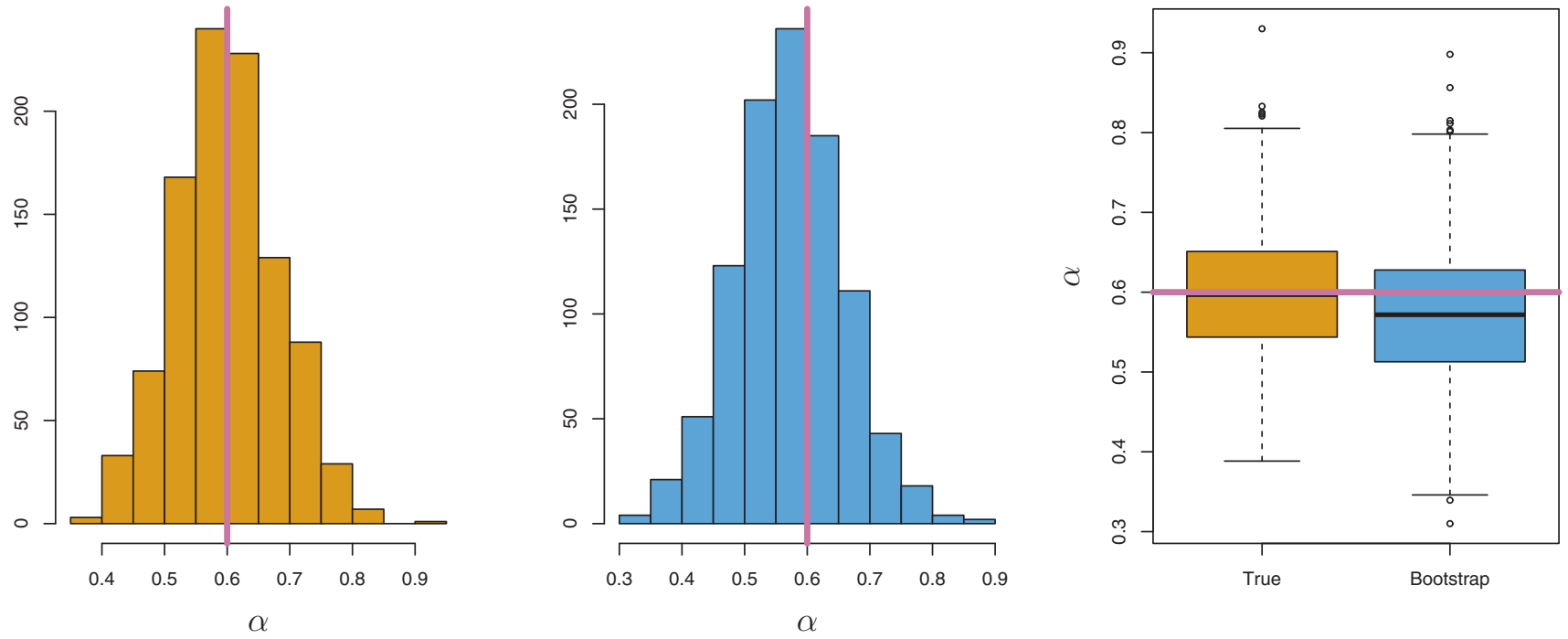


FIGURE 5.10. Left: *A histogram of the estimates of* α *obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of* α *obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of* α *displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of* α.

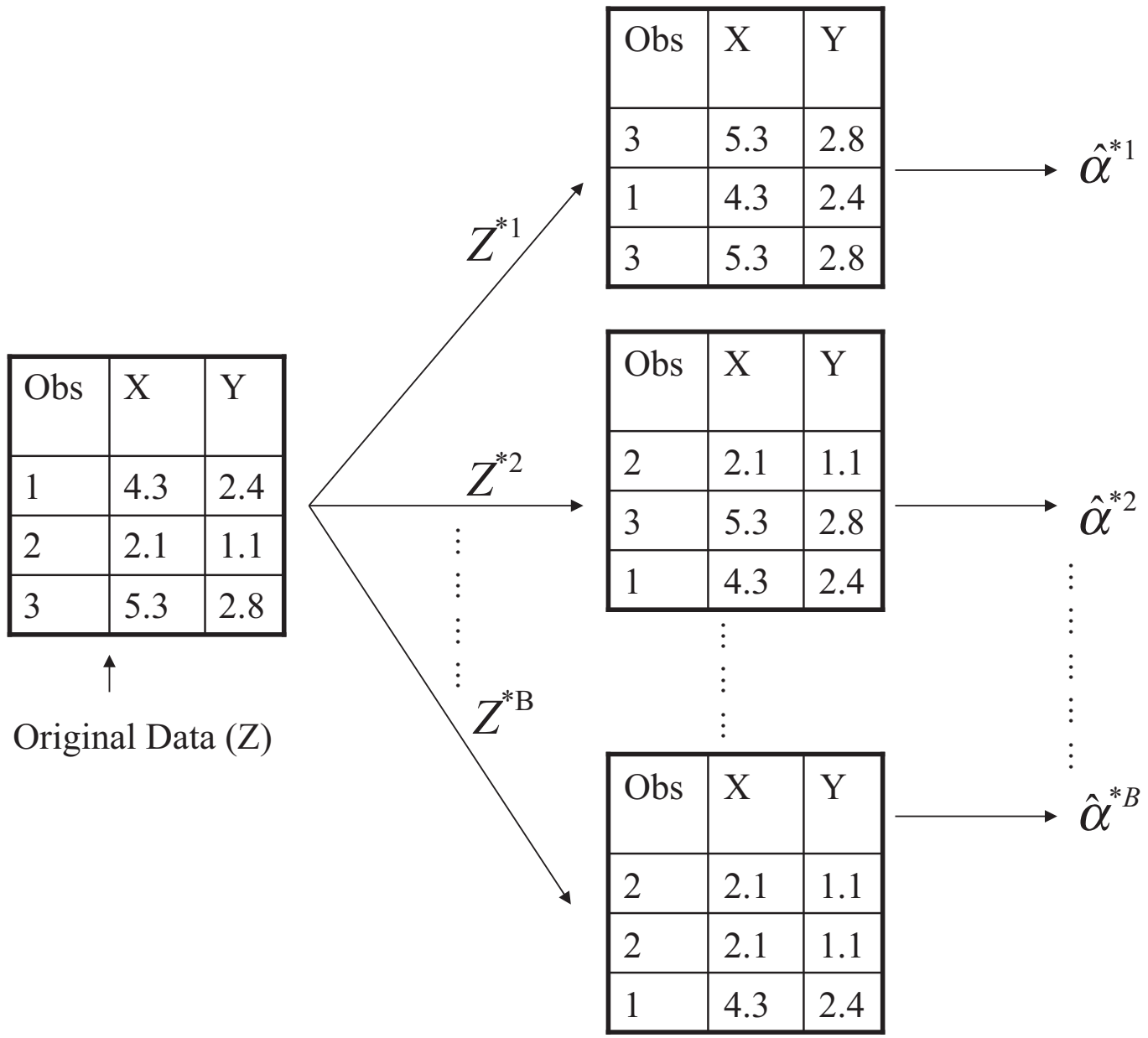


FIGURE 5.11. *A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.*

chapter 6

if n is not much larger than p, or if $p > n$

- poor predictions
- irrelevant variables lead to unnecessary complexity

- subset selection:
  - best subset selection
  - forward stepwise selection
  - backward stepwise selection
- shrinkage
  - ridge regression
- subset selection + shrinkage: lasso(least absolute shrinkage and selection operator)
- dimension reduction

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For $k = 1, 2, \ldots p$:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.

3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

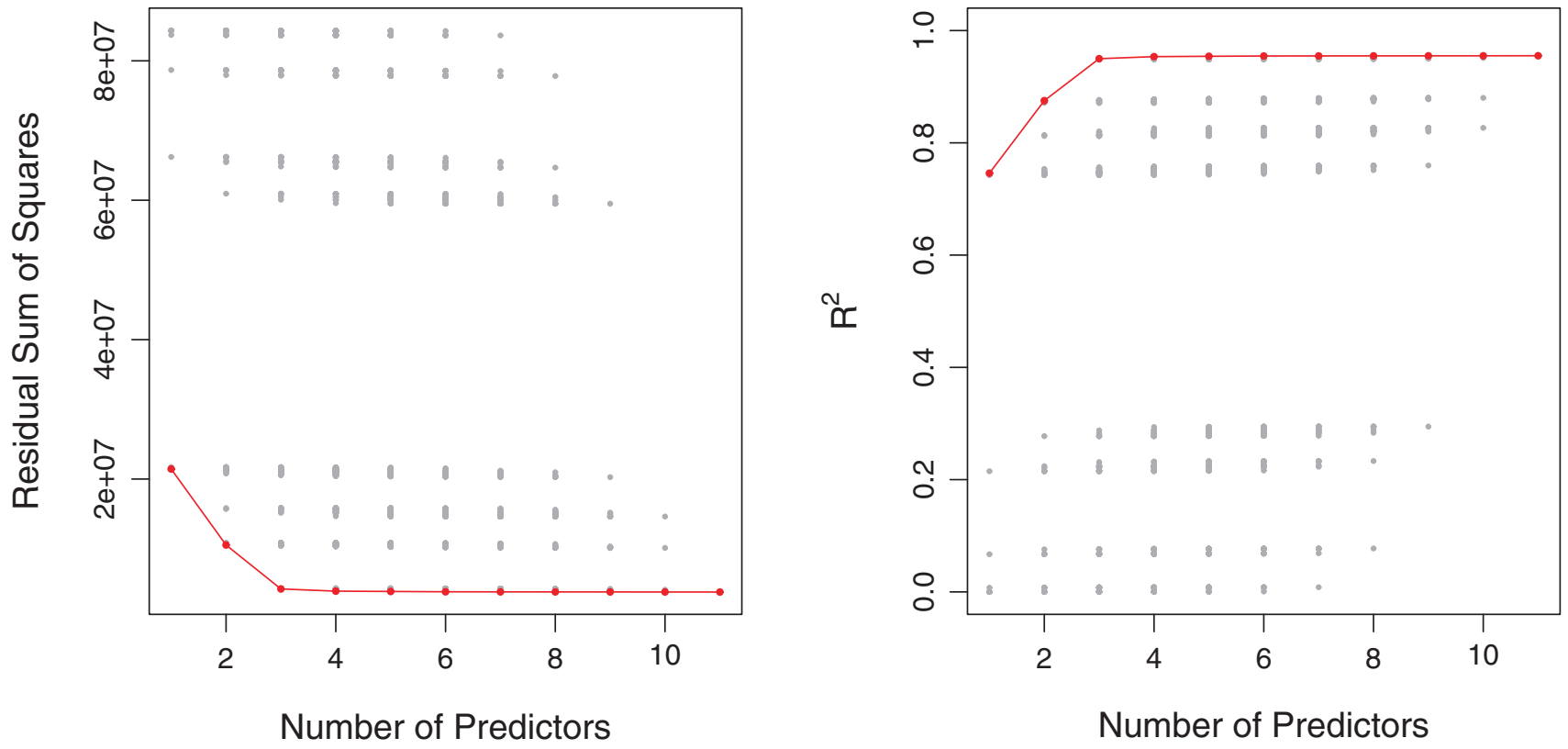


FIGURE 6.1. *For each possible model containing a subset of the ten predictors in the* `Credit` *data set, the RSS and* R^2 *are displayed. The red frontier tracks the* best *model for a given number of predictors, according to RSS and* R^2*. Though the data set contains only ten predictors, the* x*-axis ranges from* 1 *to* 11*, since one of the variables is categorical and takes on three values, leading to the creation of two dummy variables.*

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

# Variables	Best subset	Forward stepwise
One	`rating`	`rating`
Two	`rating`, `income`	`rating`, `income`
Three	`rating`, `income`, `student`	`rating`, `income`, `student`
Four	`cards`, `income`	`rating`, `income`,
	`student`, `limit`	`student`, `limit`

TABLE 6.1. *The first four selected models for best subset selection and forward stepwise selection on the* `Credit` *data set. The first three models are identical but the fourth models differ.*

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

- choosing the optimal model= minimum test error
  - test error $\simeq$ adjusted training error $\rightarrow$ C_p, AIC, BIC, Adjusted R^2
  - test error $\simeq$ cross validation error

- RSS= $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- $C_p = \frac{1}{n}\left(RSS + 2d\hat{\sigma}^2\right)$
- $\hat{\sigma}^2 =$ an estimate of the variance of the error ϵ
- $d =$ number of predictors (features) in the model
- adjusted for the fact that the training error tends to underestimate the test error

- $RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ and $TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$
- $AIC = \frac{1}{n\hat{\sigma}^2}\left(RSS + 2d\hat{\sigma}^2\right)$
- $BIC = \frac{1}{n\hat{\sigma}^2}\left(RSS + \log(n)d\hat{\sigma}^2\right)$
- Adjusted $R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$

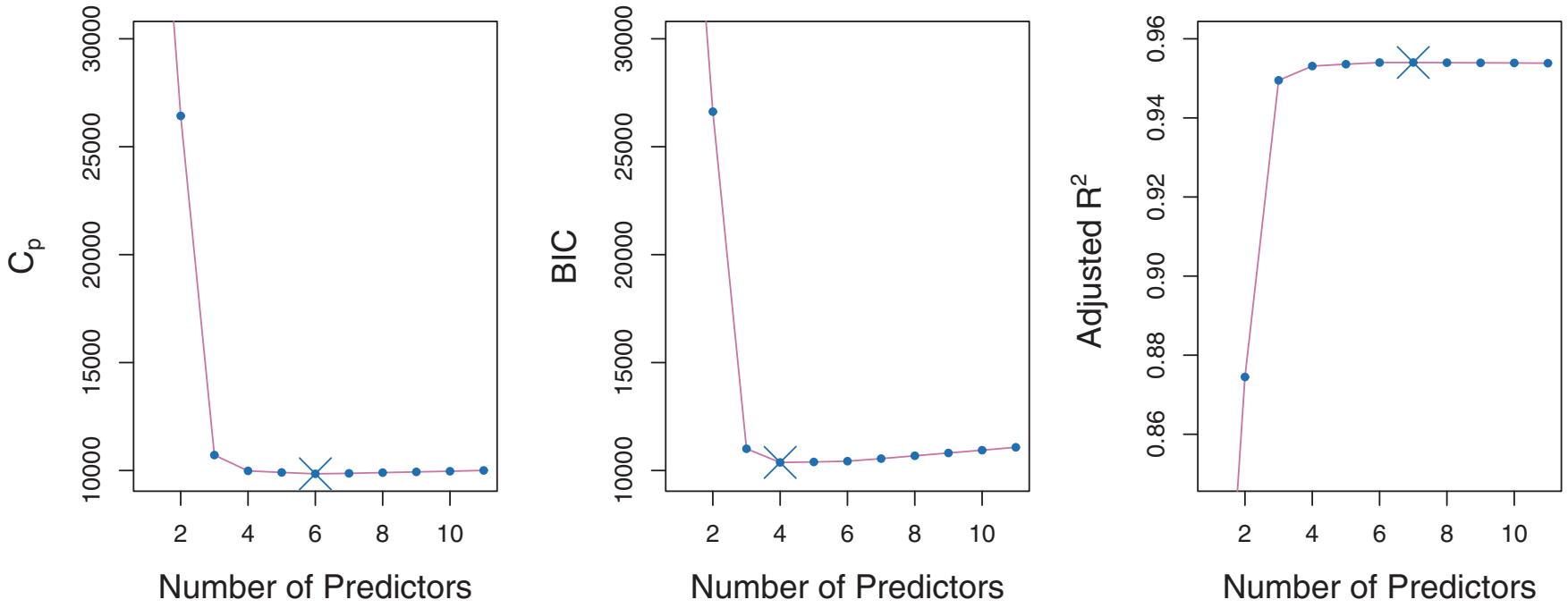


FIGURE 6.2. *C_p, BIC, and adjusted R^2 are shown for the best models of each size for the* `Credit` *data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.*

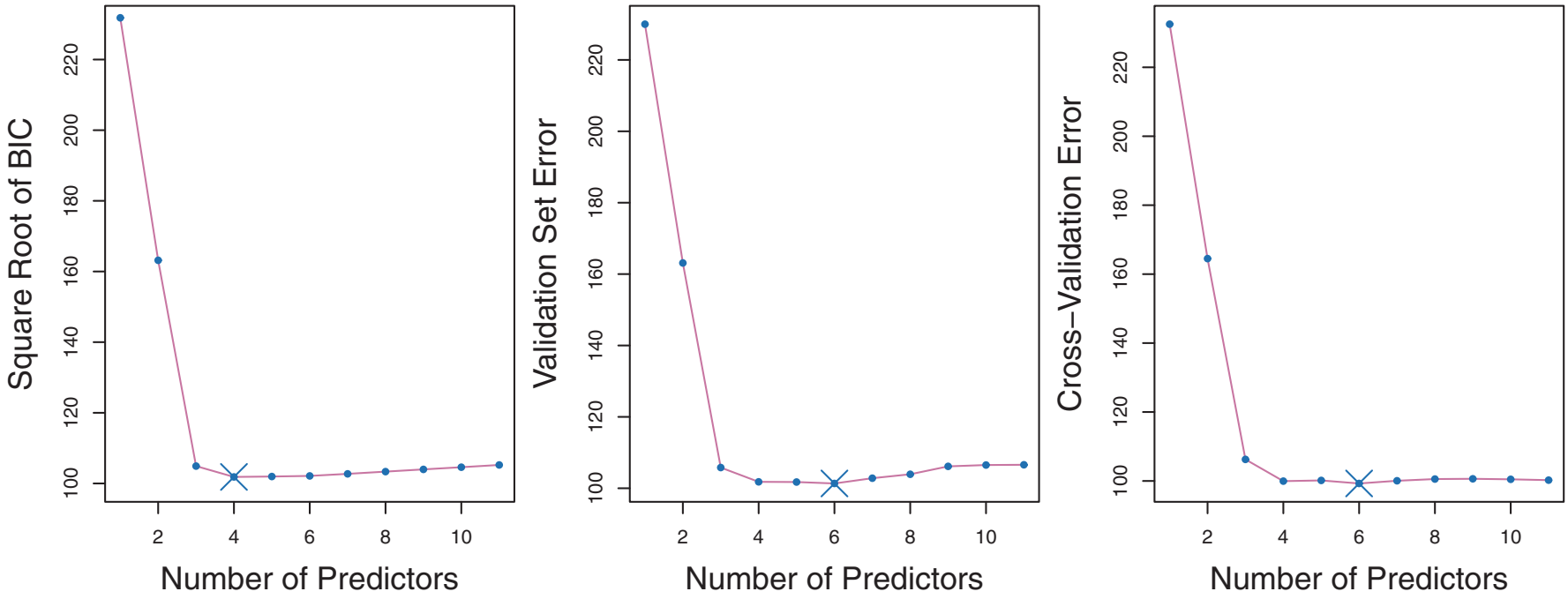


FIGURE 6.3. *For the* `Credit` *data set, three quantities are displayed for the best model containing d predictors, for d ranging from* 1 *to* 11. *The overall* best *model, based on each of these quantities, is shown as a blue cross.* Left: *Square root of BIC.* Center: *Validation set errors.* Right: *Cross-validation errors.*

lecture 12

ℓ_2-norm regularization, ridge regression:

- $$\begin{aligned}(\hat{\beta}_0, \hat{\beta}) &= \arg\min_{\beta_0,\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \\ &\quad \text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t\end{aligned}$$

- or more compactly,

$$\begin{aligned}(\hat{\beta}_0, \hat{\beta}) &= \arg\min_{\beta_0,\beta} \|y - \beta_0 1 - X\beta\|_2^2 \\ &\quad \text{subject to } \|\beta\|_2^2 \leq t,\end{aligned}$$

  where 1 is the vector of n ones.

- t is a kind of a "budget": it limits the sum of the squares of the parameter estimates.

- typically, we first standardize the predictors X so that:
  - each column is **centered**: $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$,
  - each column has unit variance $\frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1$.
- for convenience, **center** out come values y_i: $\frac{1}{n}\sum_{i=1}^{n} y_i = 0$.

- centering is convenient: we can omit the intercept term β_0.
- given a solution to the centered data, we can recover solutions for uncentered data:
  - $\hat{\beta}$ is the same,
  - $\hat{\beta}_0 = \bar{y} - \sum_{j=1}^{p} \bar{x}_j \hat{\beta}_j$.
- ...omit the intercept for the rest of the lectures

ridge regression:

- contraints problem:

$$\begin{aligned} \hat{\beta} &= \arg\min_{\beta} \|y - X\beta\|_2^2 \\ &\quad \text{subject to } \|\beta\|_2^2 \leq t \end{aligned}$$

- Lagrangian form:

$$\hat{\beta} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$$

- there is one-to-one correspondence between the constraint problem and the Lagrangian form.
- for each value t, there is a corresponding value λ that yields the same solution from the Lagrangian form.

ℓ_1-norm regularization, lasso regression:

- contraints problem:

$$\begin{aligned} \hat{\beta} &= \arg\min_{\beta} \|y - X\beta\|_2^2 \\ &\quad \text{subject to } \|\beta\|_1 \leq t \end{aligned}$$

- Lagrangian form:

$$\hat{\beta} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

- there is one-to-one correspondence between the constraint problem and the Lagrangian form.
- for each value t, there is a corresponding value λ that yields the same solution from the Lagrangian form.

why consider the an alternative to least-squares?

- prediction accuracy:
  - least-squares often has low bias and high variance.
  - prediction accuracy can sometimes be improved by
    - shrinking the regression coefficients: ridge
    - setting some coefficients to zero: lasso
- interpretation: with a large number of predictors, we like to identify a smaller subset that exhibit the strongest effect.

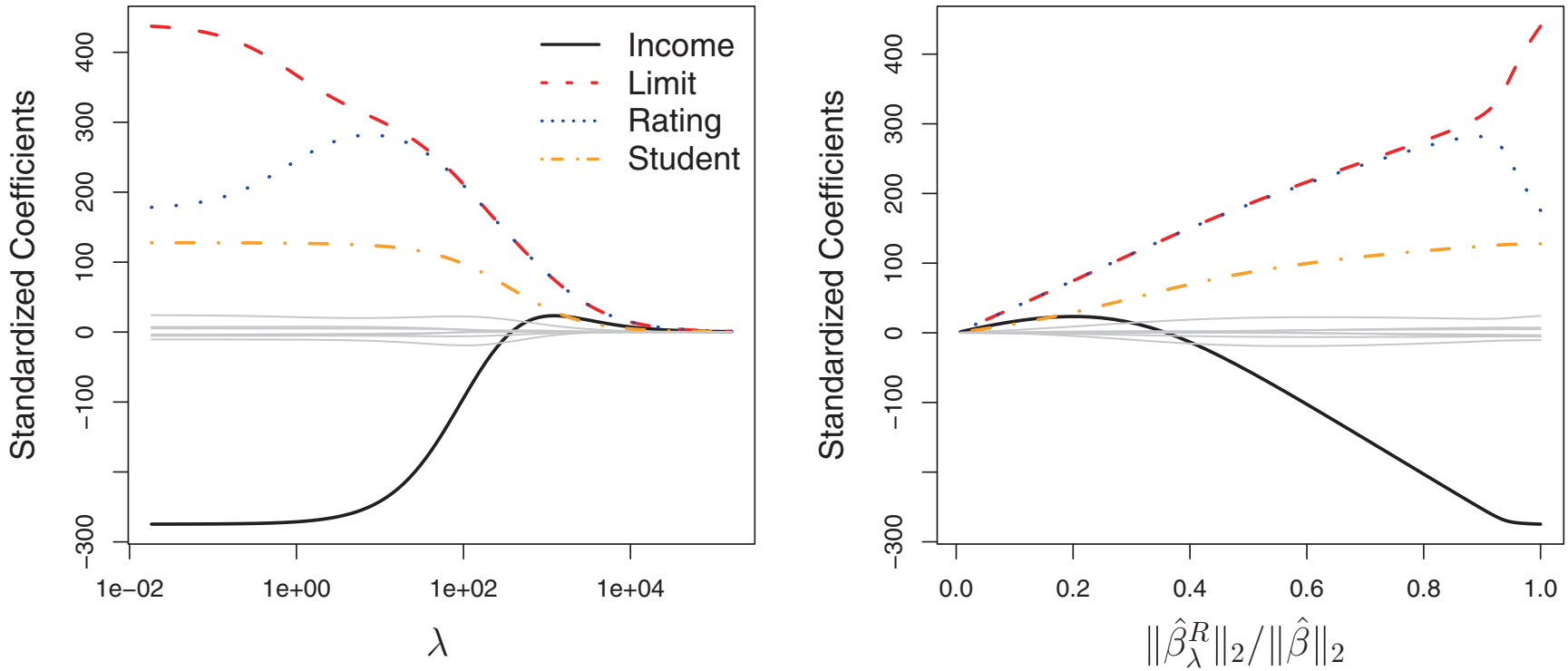


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

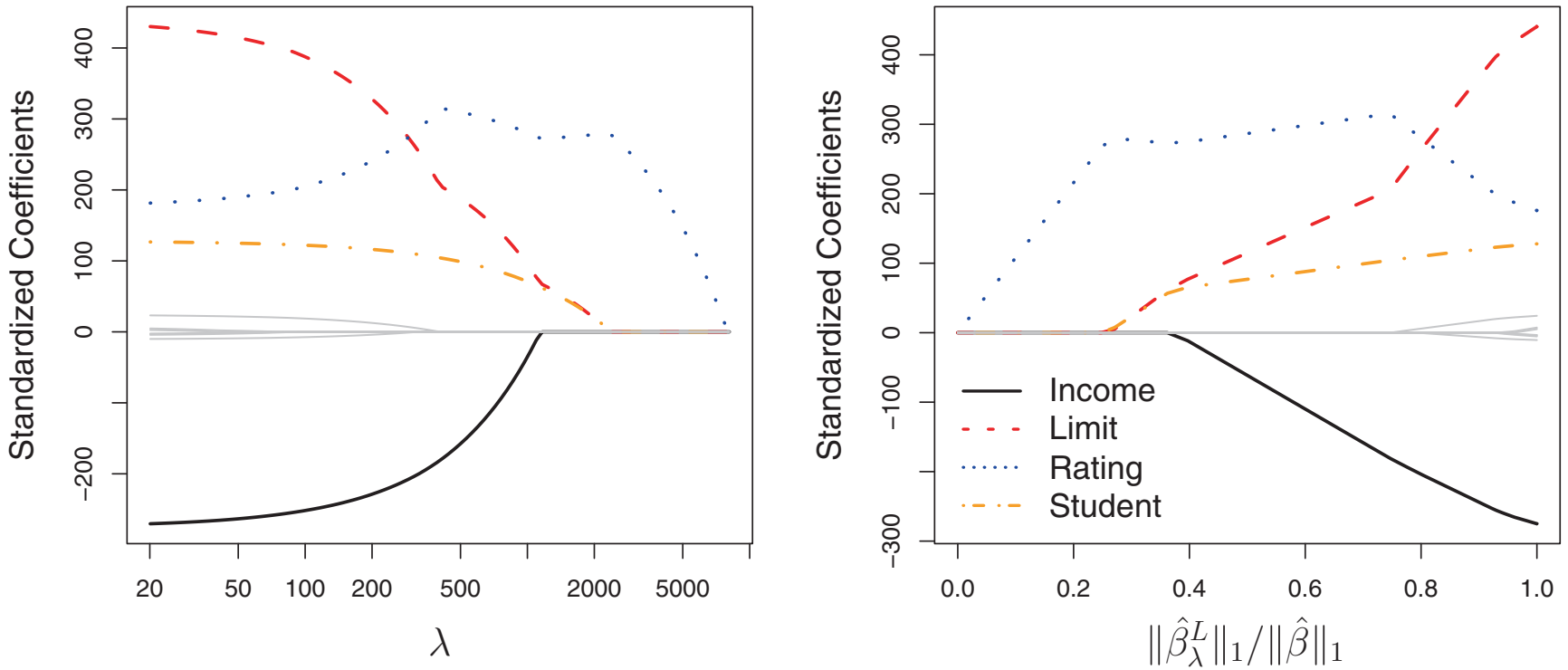


FIGURE 6.6. *The standardized lasso coefficients on the* `Credit` *data set are shown as a function of* λ *and* $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$.

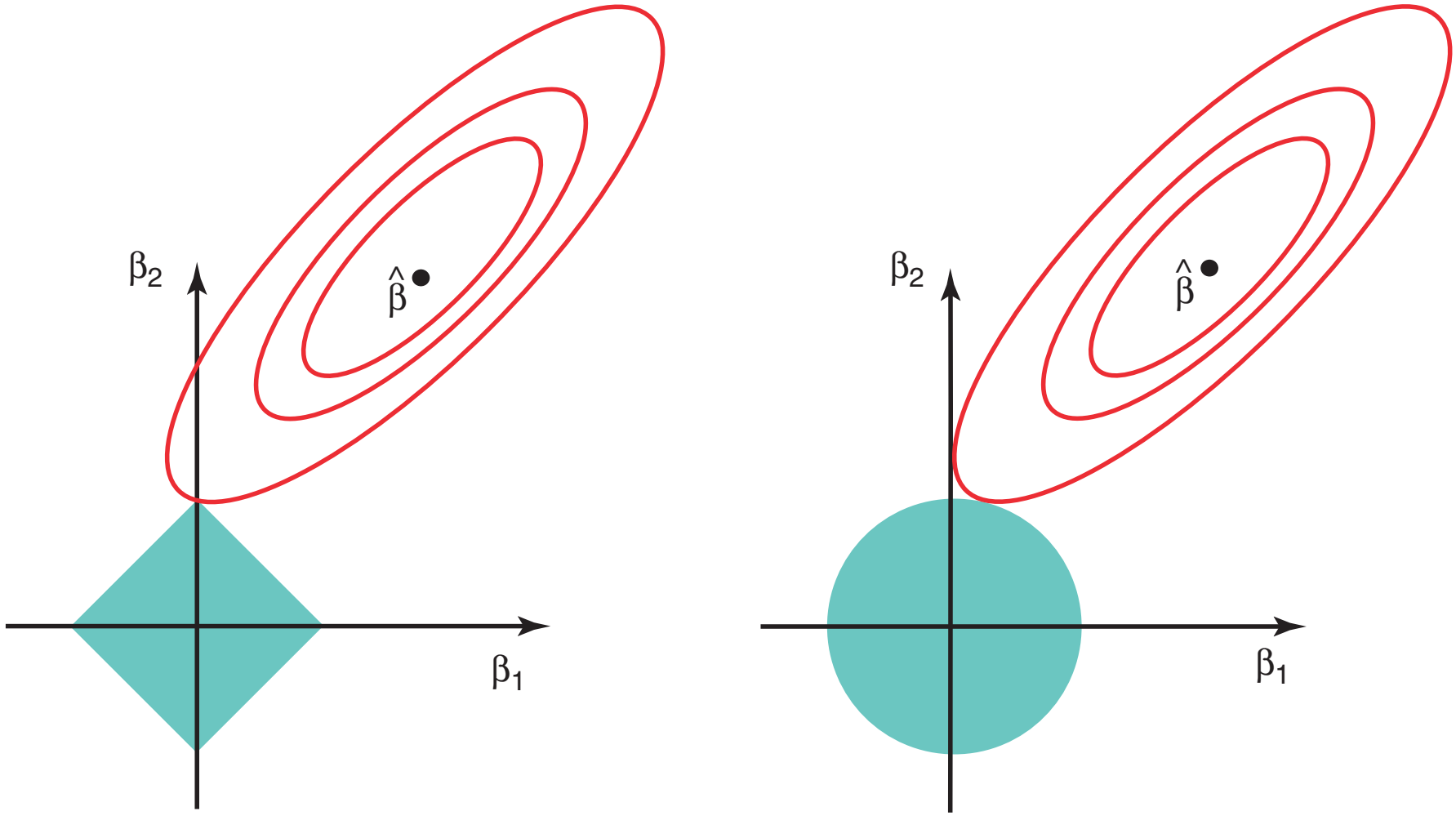


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

why lasso?

- encourage simplicity of interpretation and enforce sparsity
- "bet-on-sparsity principle":
  - true underlying model is sparse: ℓ_1 penalty (regularization) recovers it.
  - true underlying model is not sparse: ℓ_1 will not work well, however, in that instance, no method can do well.
- computational efficiency: with 100 observation and one millions features
  - without lasso: we have to estimate one million nonzero parameters: challenging.
  - with lasso: at most 100 parameters can be nonzero in the solution. computationally much easier.

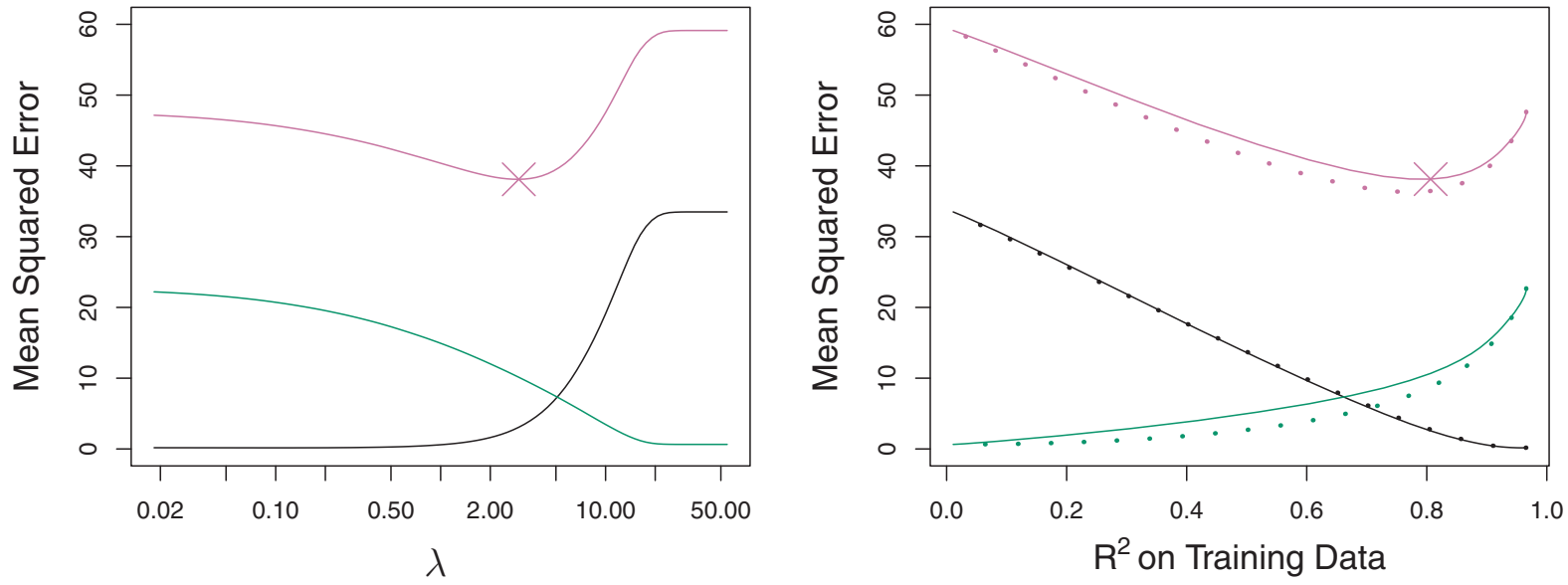


FIGURE 6.8. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on a simulated data set.* Right: *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). Both are plotted against their* R^2 *on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

true model: response is a function of all 45 features (predictors)

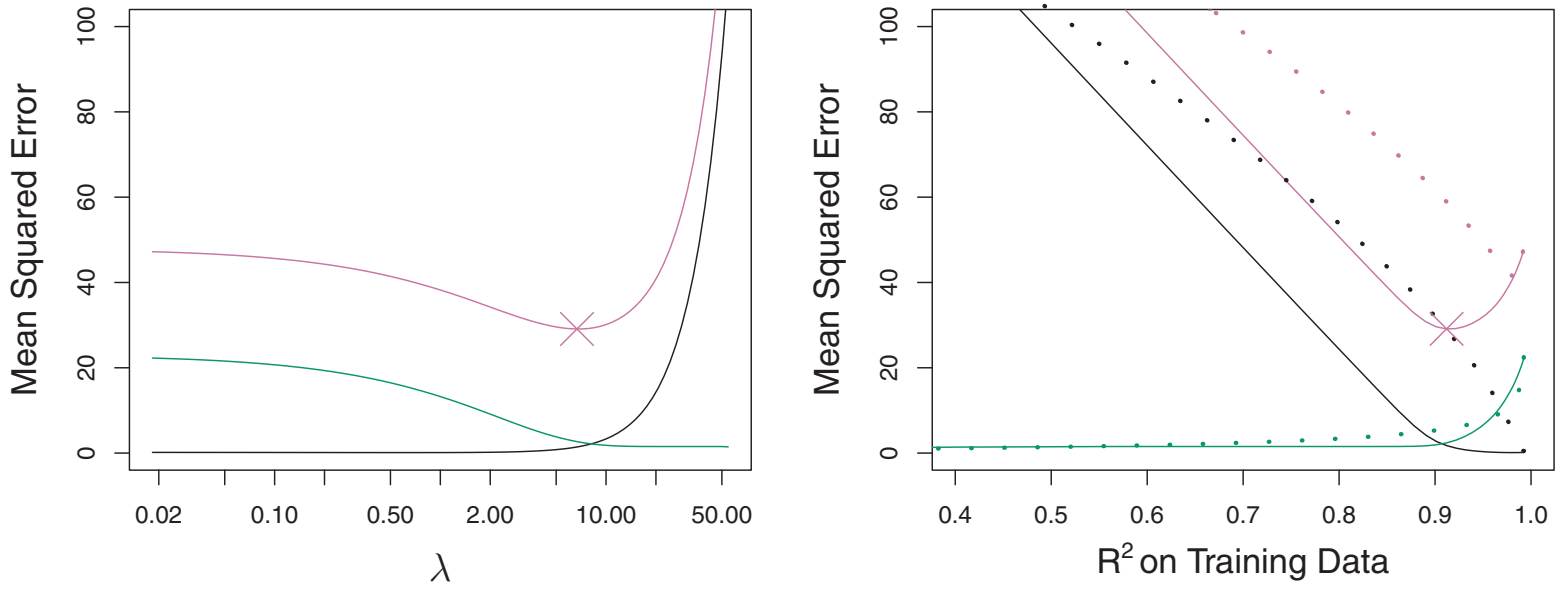


FIGURE 6.9. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Figure 6.8, except that now only two predictors are related to the response.* Right: *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). Both are plotted against their* R^2 *on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

true model: response is a function of only 2 out of the 45 features (predictors)

- simple special case for ridge and lasso
- bayesian interpretation

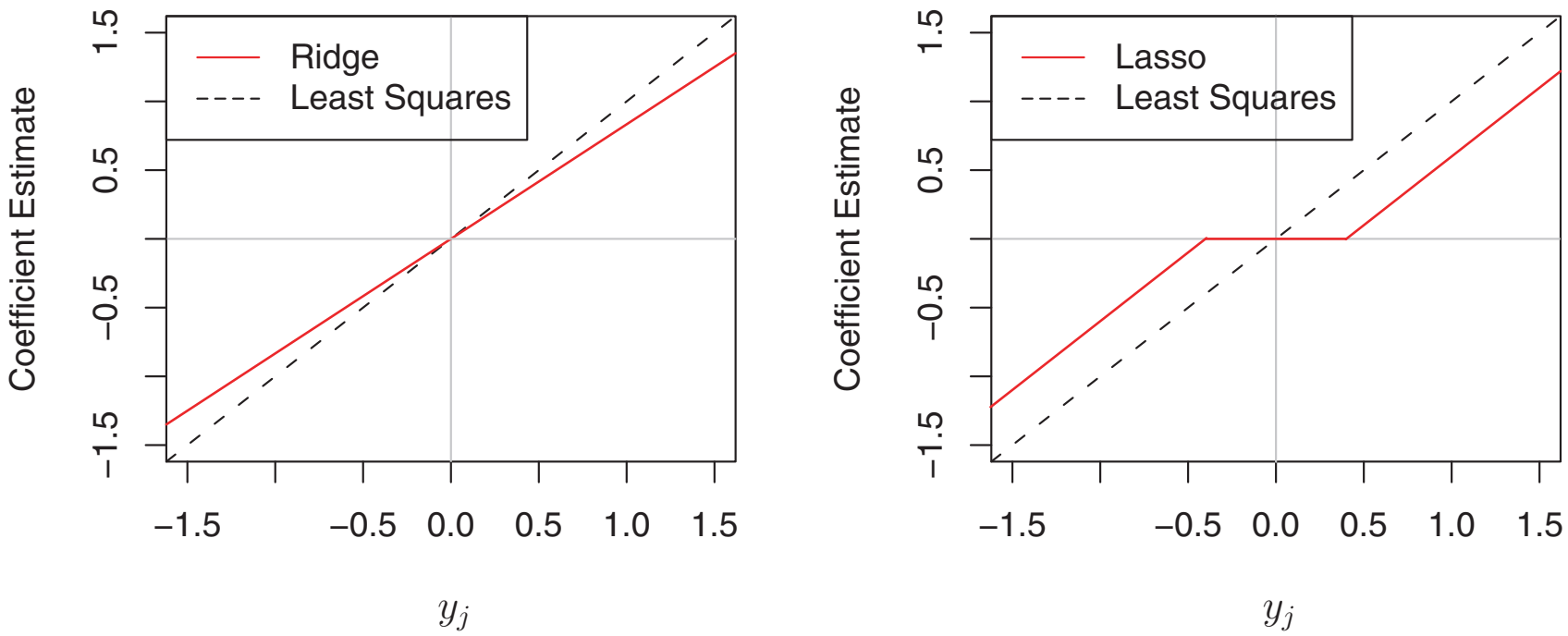


FIGURE 6.10. *The ridge regression and lasso coefficient estimates for a simple setting with* $n = p$ *and* $\mathbf{X}$ *a diagonal matrix with* 1*'s on the diagonal.* Left: *The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates.* Right: *The lasso coefficient estimates are soft-thresholded towards zero.*

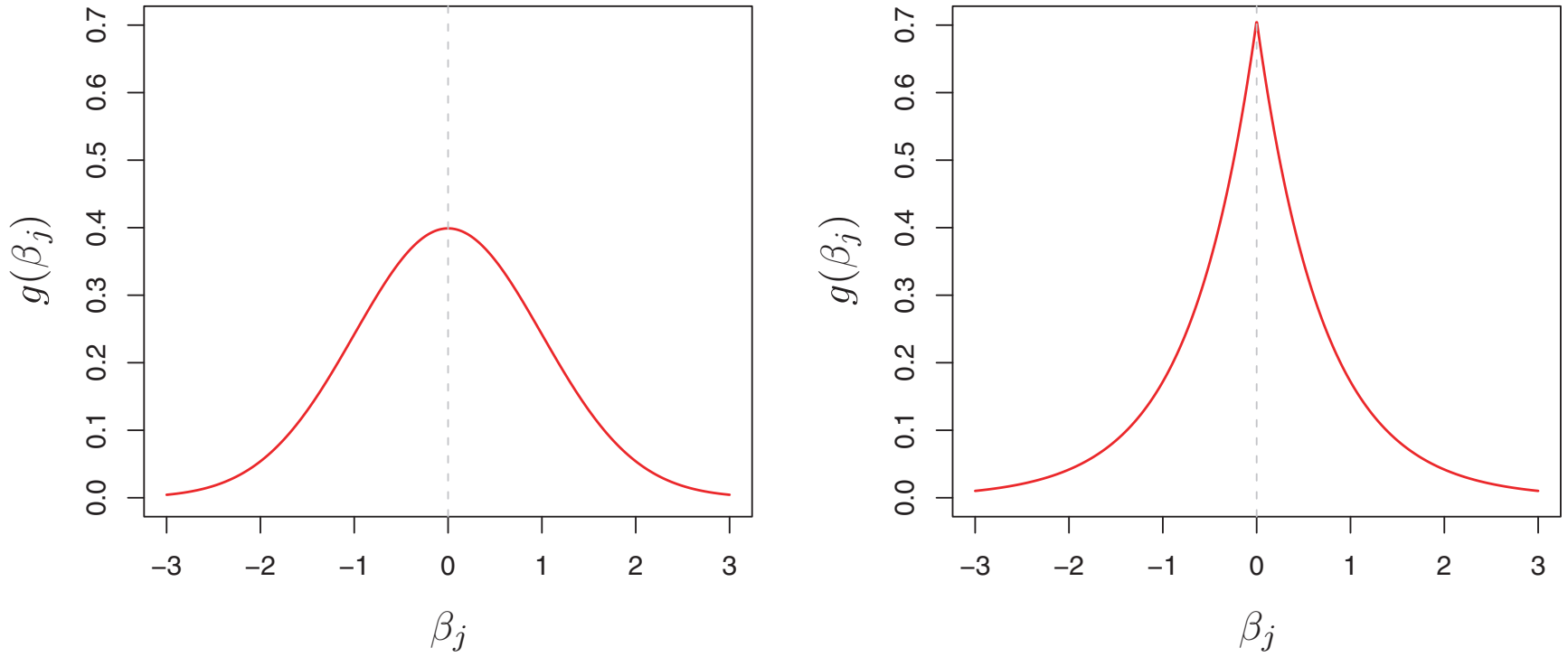


FIGURE 6.11. Left: *Ridge regression is the posterior mode for β under a Gaussian prior.* Right: *The lasso is the posterior mode for β under a double-exponential prior.*

- selecting the tuning parameter λ

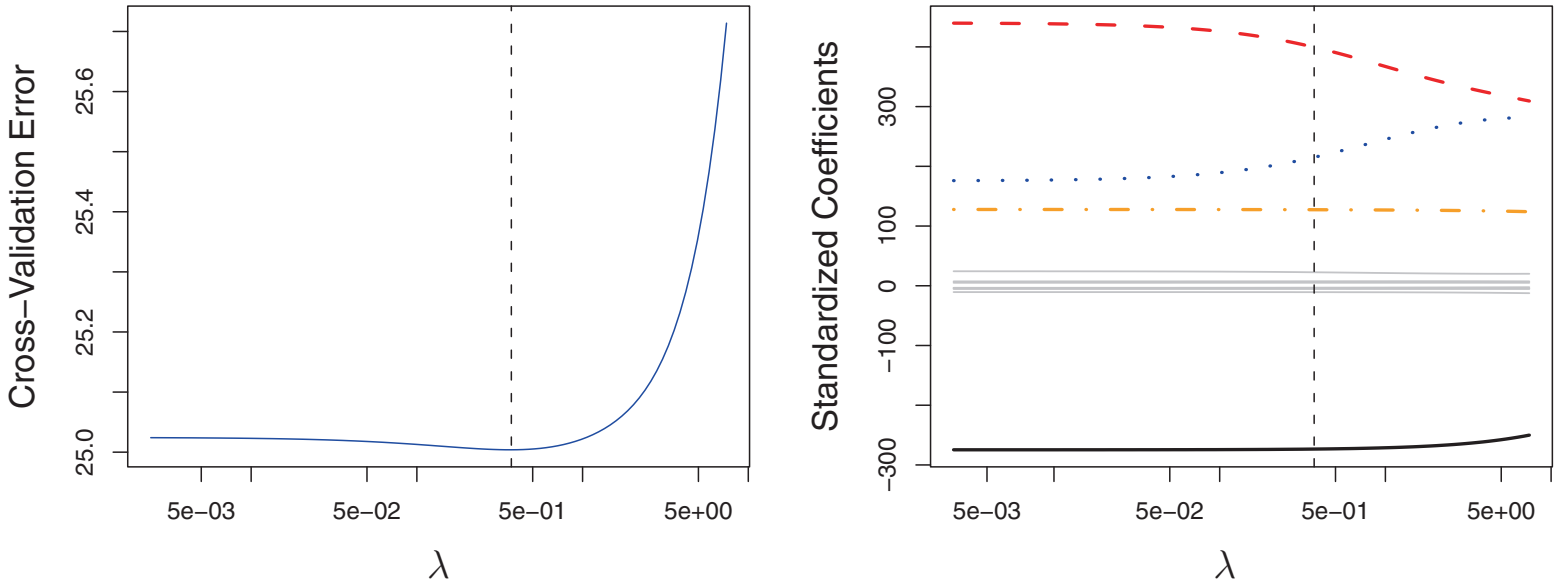


FIGURE 6.12. Left: *Cross-validation errors that result from applying ridge regression to the* `Credit` *data set with various value of* λ. Right: *The coefficient estimates as a function of* λ. *The vertical dashed lines indicate the value of* λ *selected by cross-validation.*

optimal fit involves a small amount of shrinkage

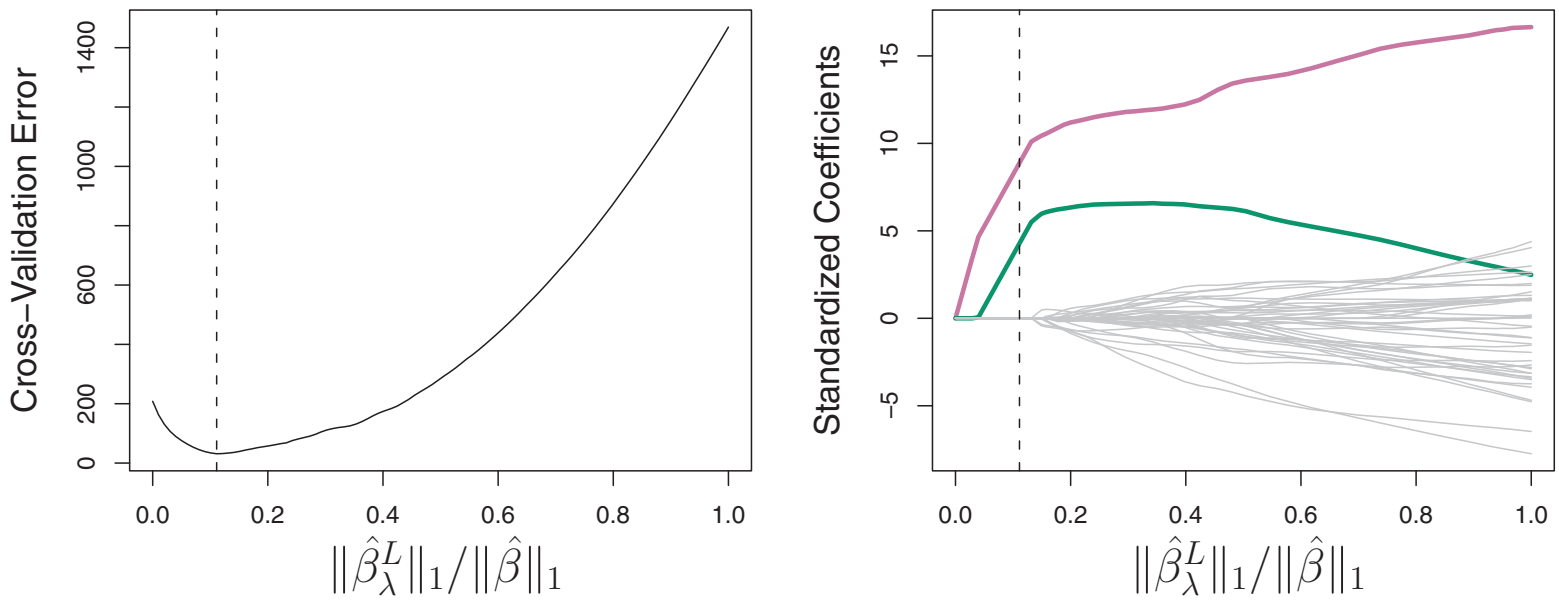


FIGURE 6.13. Left*: Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9.* Right: *The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

$p = 45$ and $n = 50$

high dimensional data

- predict blood pressure on the basis of
  - age, gender, and BMI $\rightarrow n \simeq 200$ and $p \simeq 3$
  - measurements for half a million single nucleotide polymorphisms $\rightarrow n \simeq 200$ and $p \simeq 500000$

- what goes wrong when $p > n$?
- always a perfect fit!
- when $p > n$, simple linear regression is too flexible.
- perfect fit $\rightarrow$ RSS=0 $\rightarrow$ $\hat{\sigma}^2 = 0$ $\rightarrow$ can't use C_p, AIC, BIC

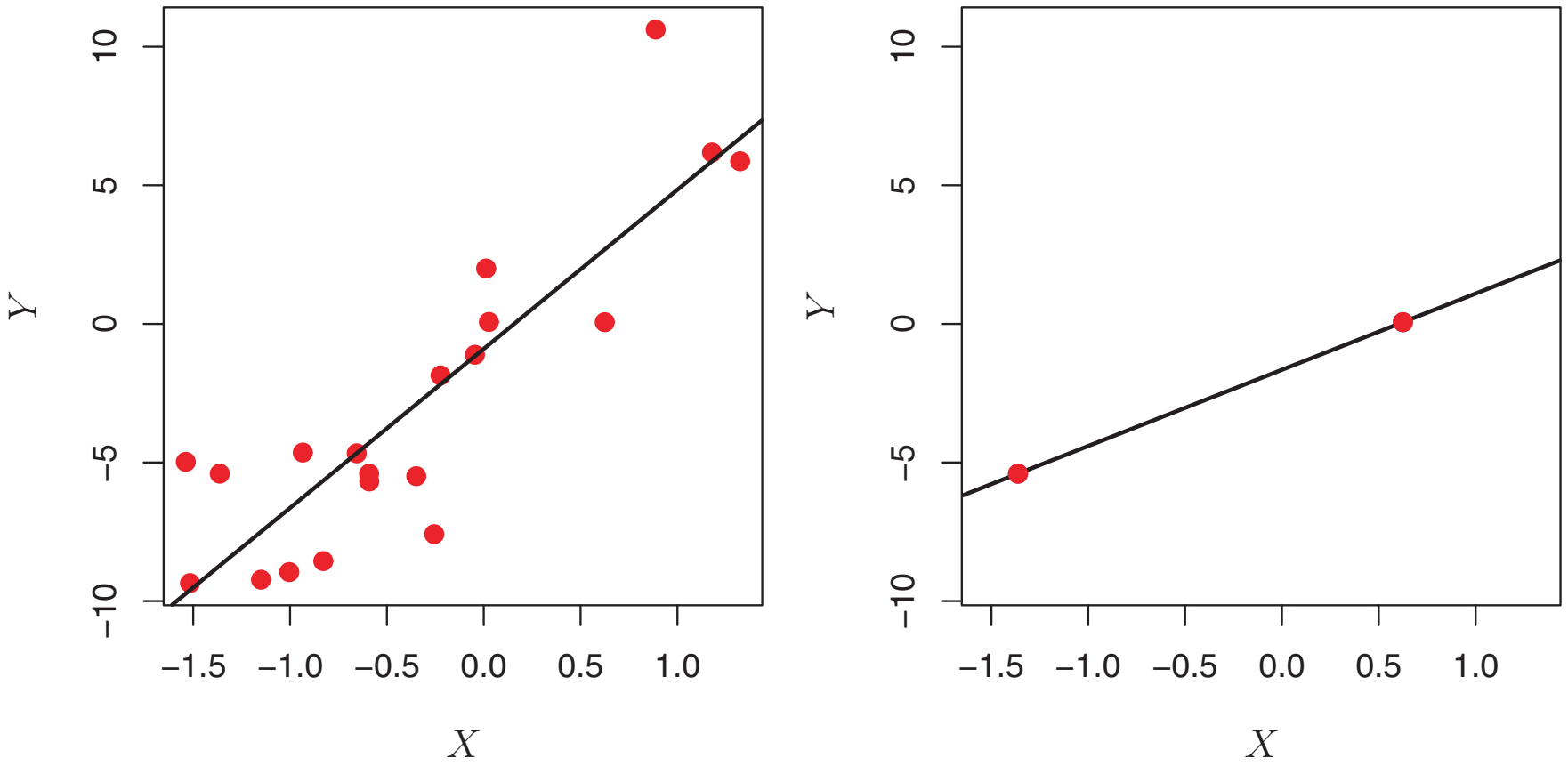


FIGURE 6.22. Left: *Least squares regression in the low-dimensional setting.* Right: *Least squares regression with $n = 2$ observations and two parameters to be estimated (an intercept and a coefficient).*

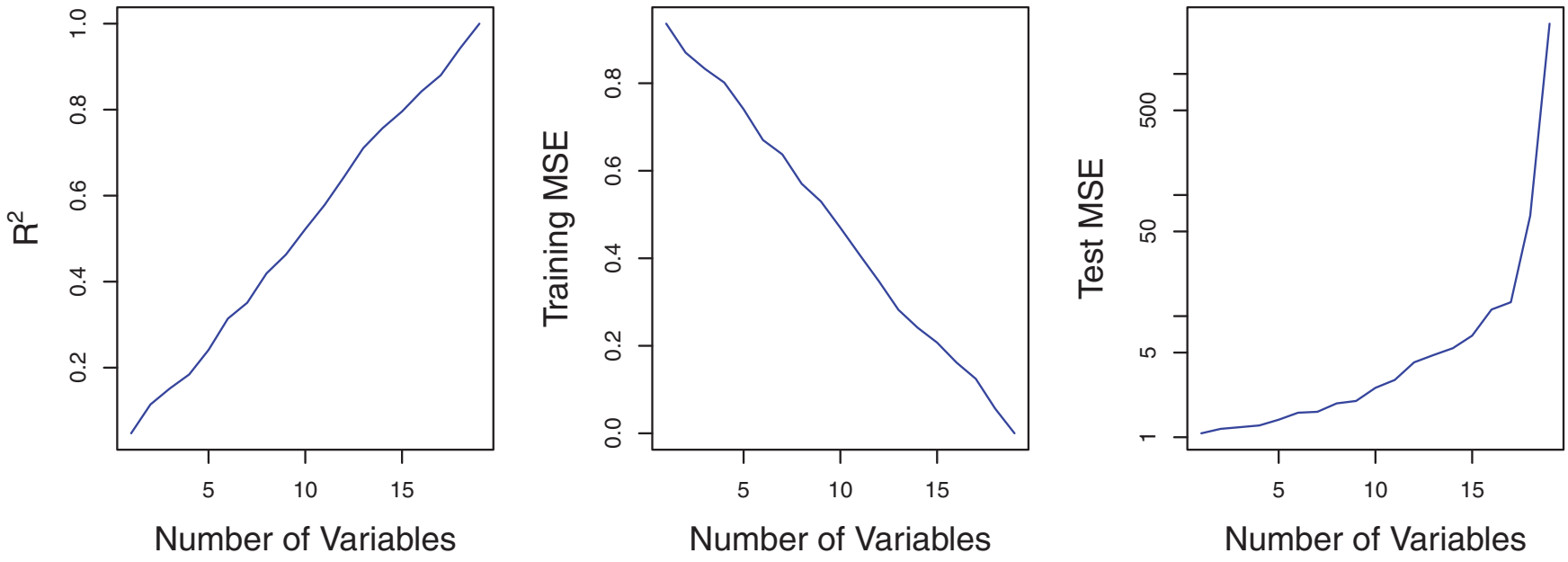


FIGURE 6.23. *On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model.* Left: *The R^2 increases to 1 as more features are included.* Center: *The training set MSE decreases to 0 as more features are included.* Right: *The test set MSE increases as more features are included.*

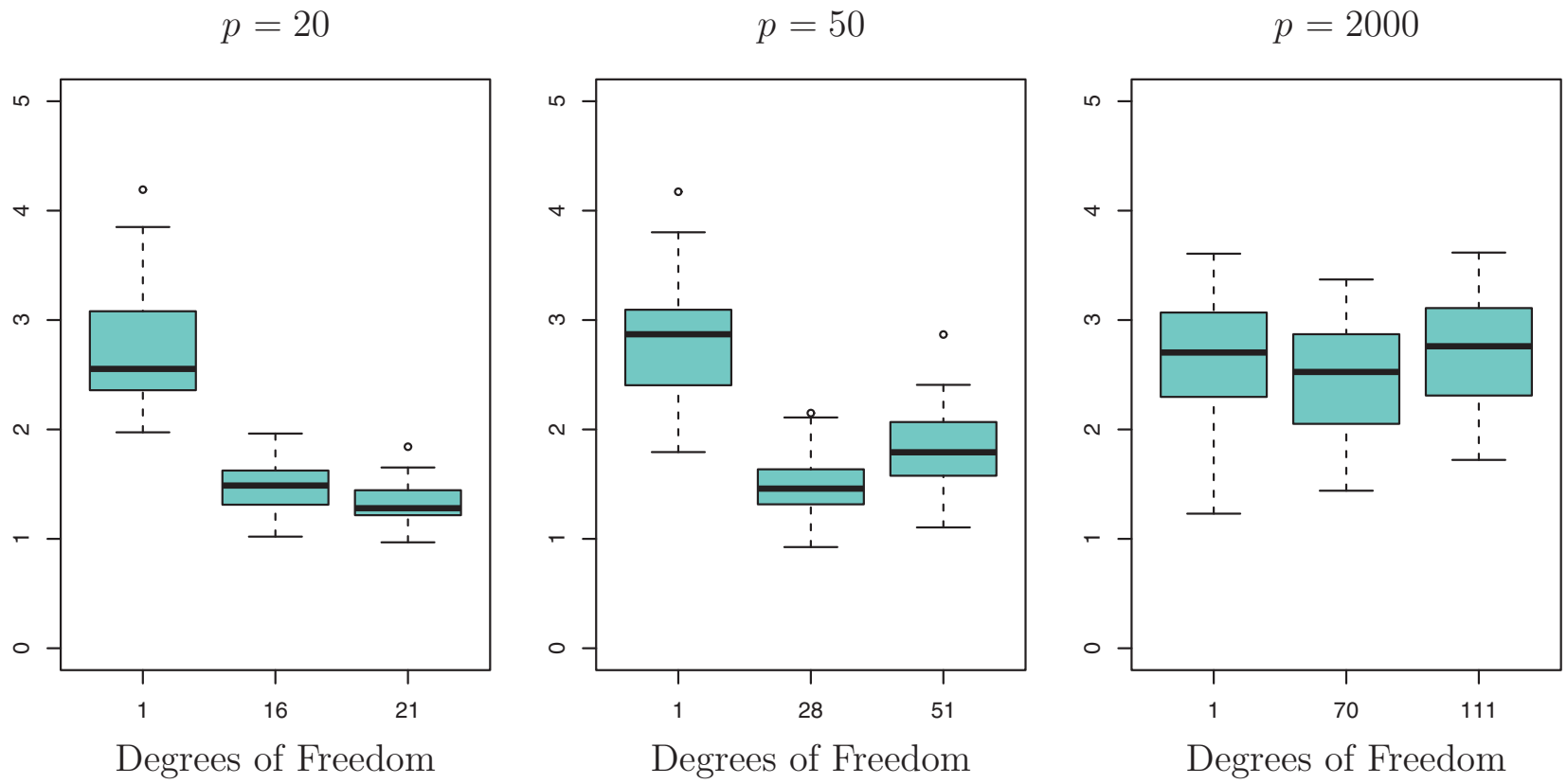


FIGURE 6.24. *The lasso was performed with $n = 100$ observations and three values of p, the number of features. Of the p features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter λ in (6.7). For ease of interpretation, rather than reporting λ, the* degrees of freedom *are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When $p = 20$, the lowest test MSE was obtained with the smallest amount of regularization. When $p = 50$, the lowest test MSE was achieved when there is a substantial amount of regularization. When $p = 2{,}000$ the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.*

chapter 8

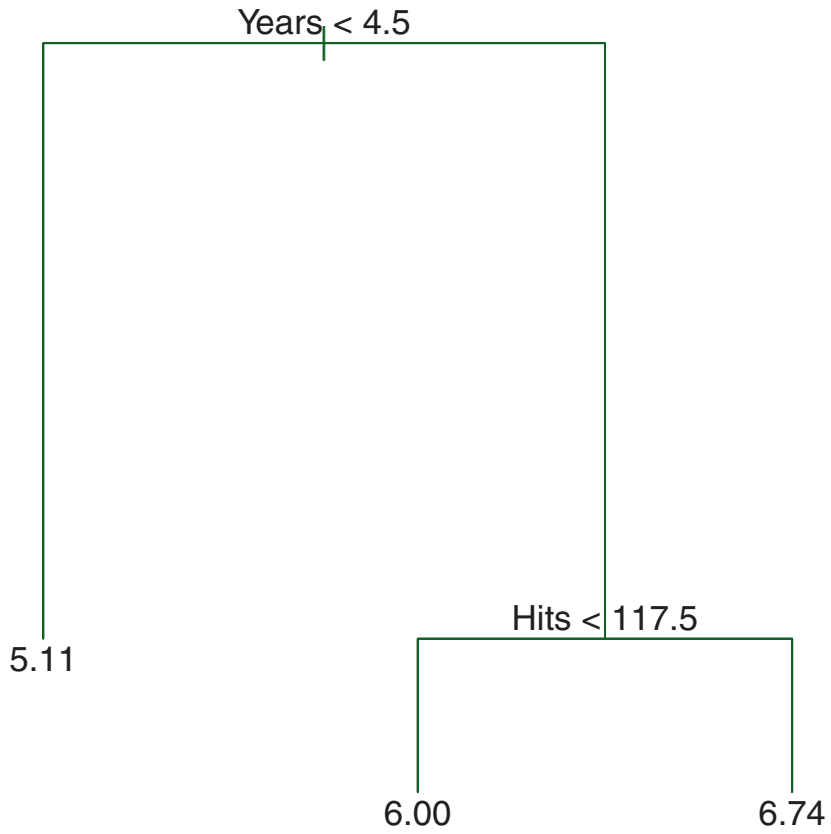


FIGURE 8.1. *For the* `Hitters` *data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year. At a given internal node, the label (of the form* $X_j < t_k$*) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to* $X_j \geq t_k$*. For instance, the split at the top of the tree results in two large branches. The left-hand branch corresponds to* `Years<4.5`*, and the right-hand branch corresponds to* `Years>=4.5`*. The tree has two internal nodes and three terminal nodes, or leaves. The number in each leaf is the mean of the response for the observations that fall there.*

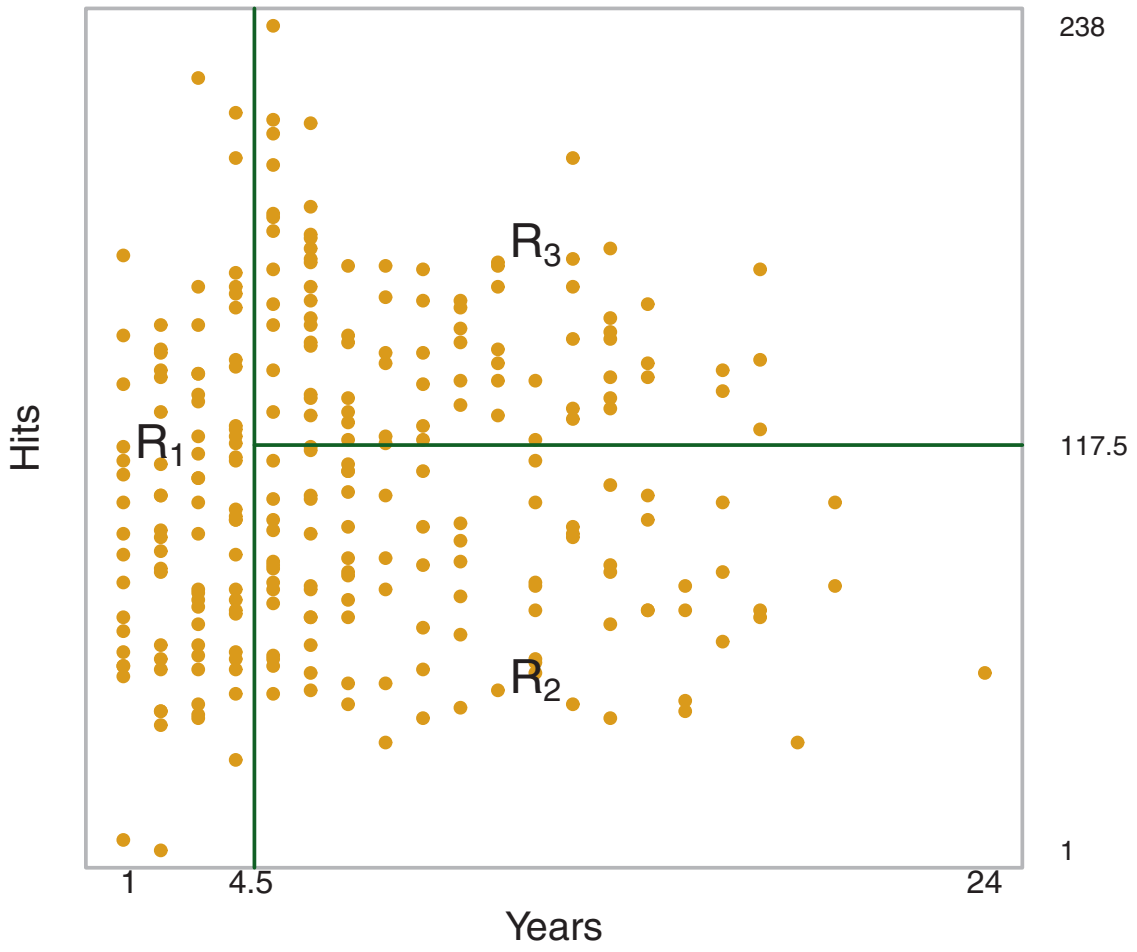


FIGURE 8.2. *The three-region partition for the* `Hitters` *data set from the regression tree illustrated in Figure 8.1.*

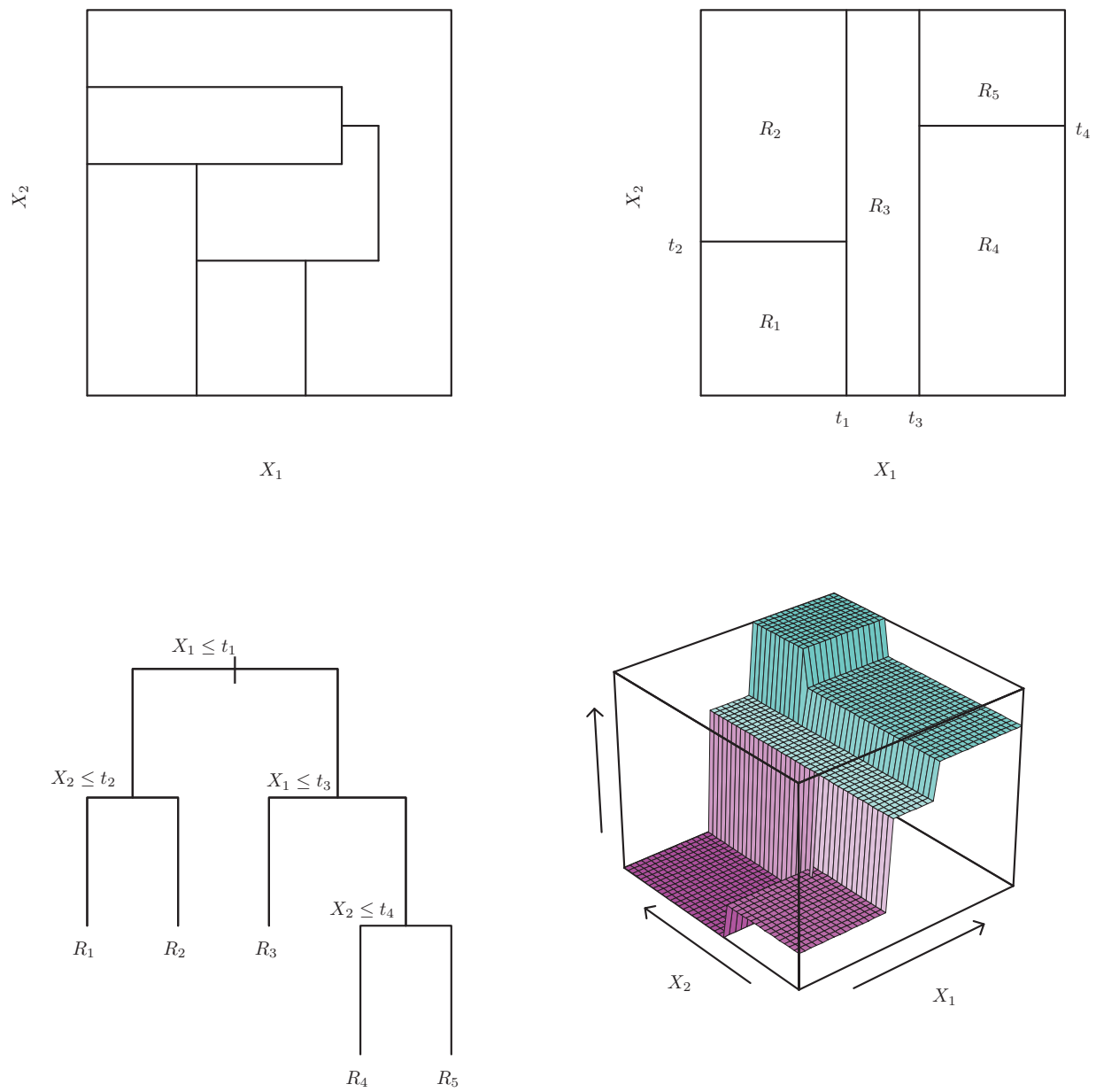


FIGURE 8.3. Top Left: *A partition of two-dimensional feature space that could not result from recursive binary splitting.* Top Right: *The output of recursive binary splitting on a two-dimensional example.* Bottom Left: *A tree corresponding to the partition in the top right panel.* Bottom Right: *A perspective plot of the prediction surface corresponding to that tree.*

Algorithm 8.1 *Building a Regression Tree*

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.

2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α.

3. Use K-fold cross-validation to choose α. That is, divide the training observations into K folds. For each $k = 1, \ldots, K$:

   (a) Repeat Steps 1 and 2 on all but the kth fold of the training data.

   (b) Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α.

   Average the results for each value of α, and pick α to minimize the average error.

4. Return the subtree from Step 2 that corresponds to the chosen value of α.

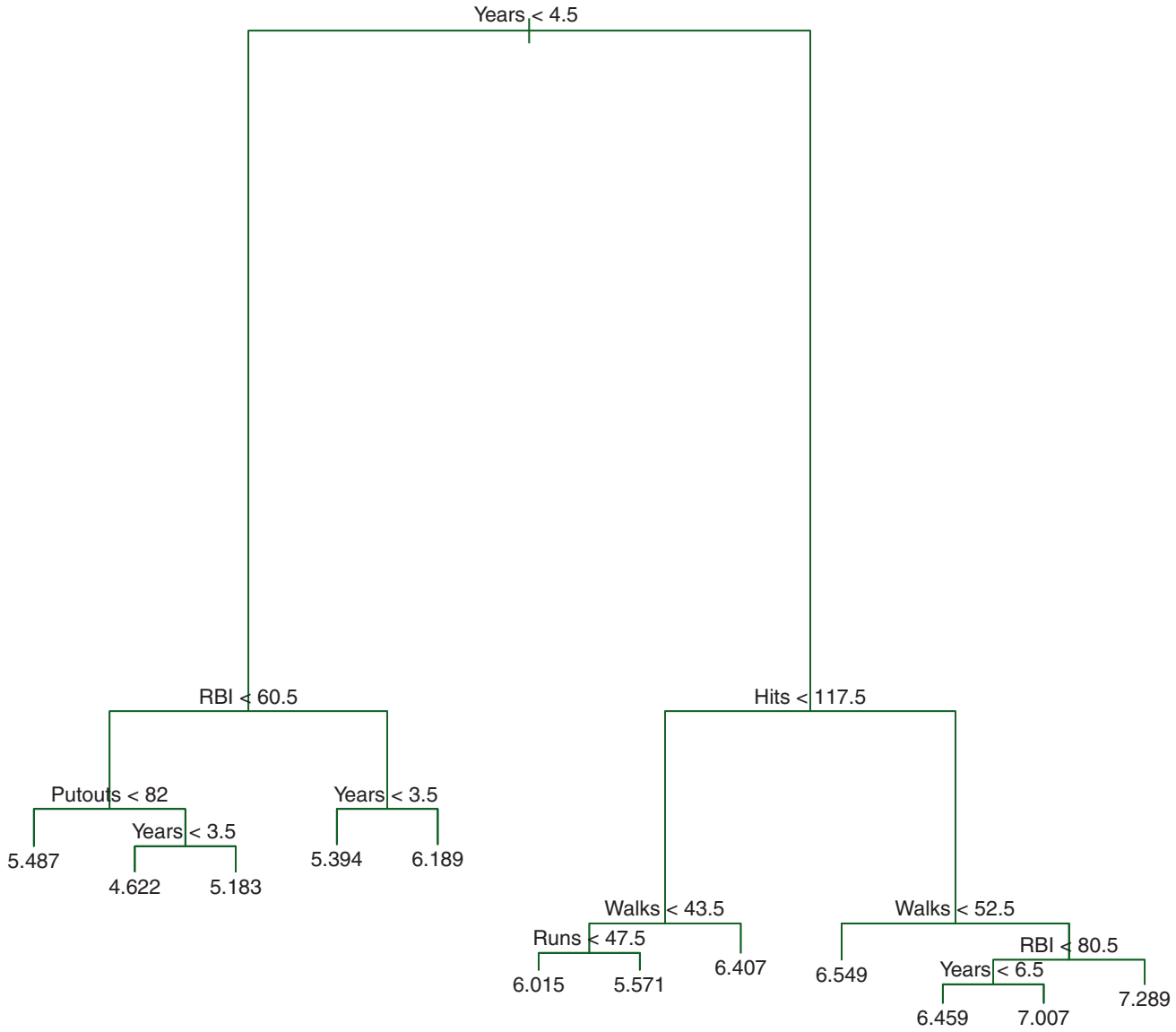


FIGURE 8.4. *Regression tree analysis for the* `Hitters` *data. The unpruned tree that results from top-down greedy splitting on the training data is shown.*

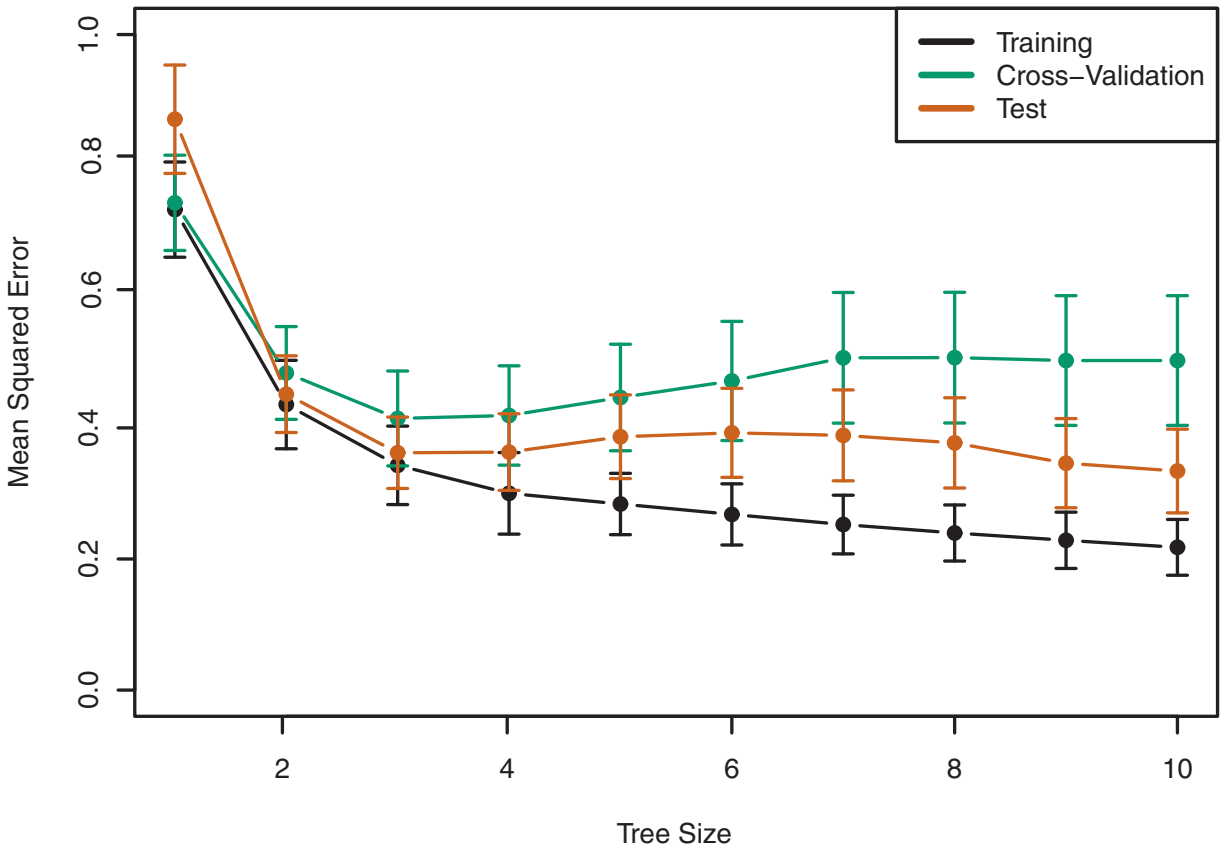


FIGURE 8.5. *Regression tree analysis for the* `Hitters` *data. The training, cross-validation, and test MSE are shown as a function of the number of terminal nodes in the pruned tree. Standard error bands are displayed. The minimum cross-validation error occurs at a tree size of three.*

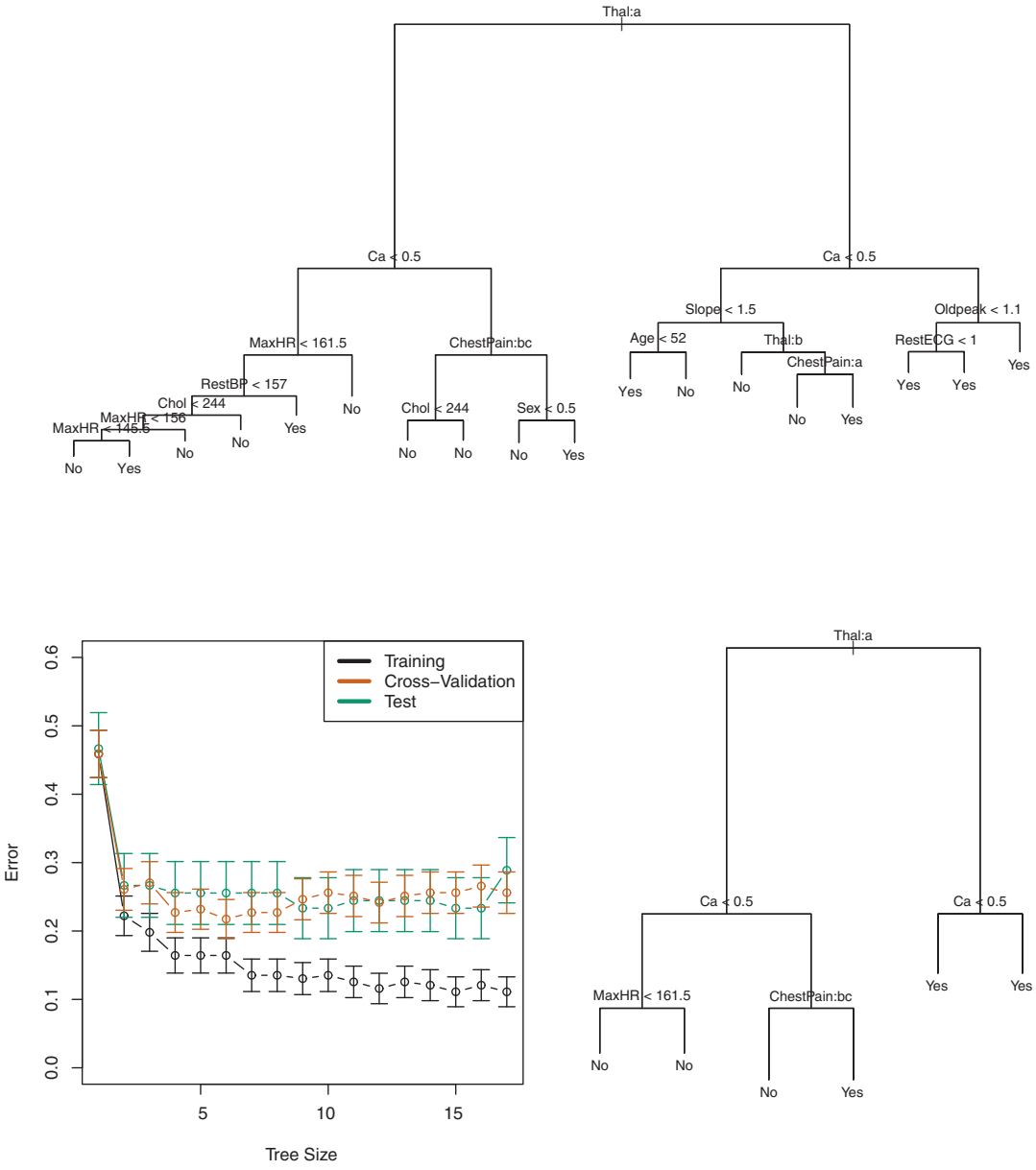


FIGURE 8.6. `Heart` *data.* Top: *The unpruned tree.* Bottom Left: *Cross -validation error, training, and test error, for different sizes of the pruned tree.* Bottom Right: *The pruned tree corresponding to the minimal cross-validation error.*

- misclassification error
- cross entropy
- gini index= $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'}$

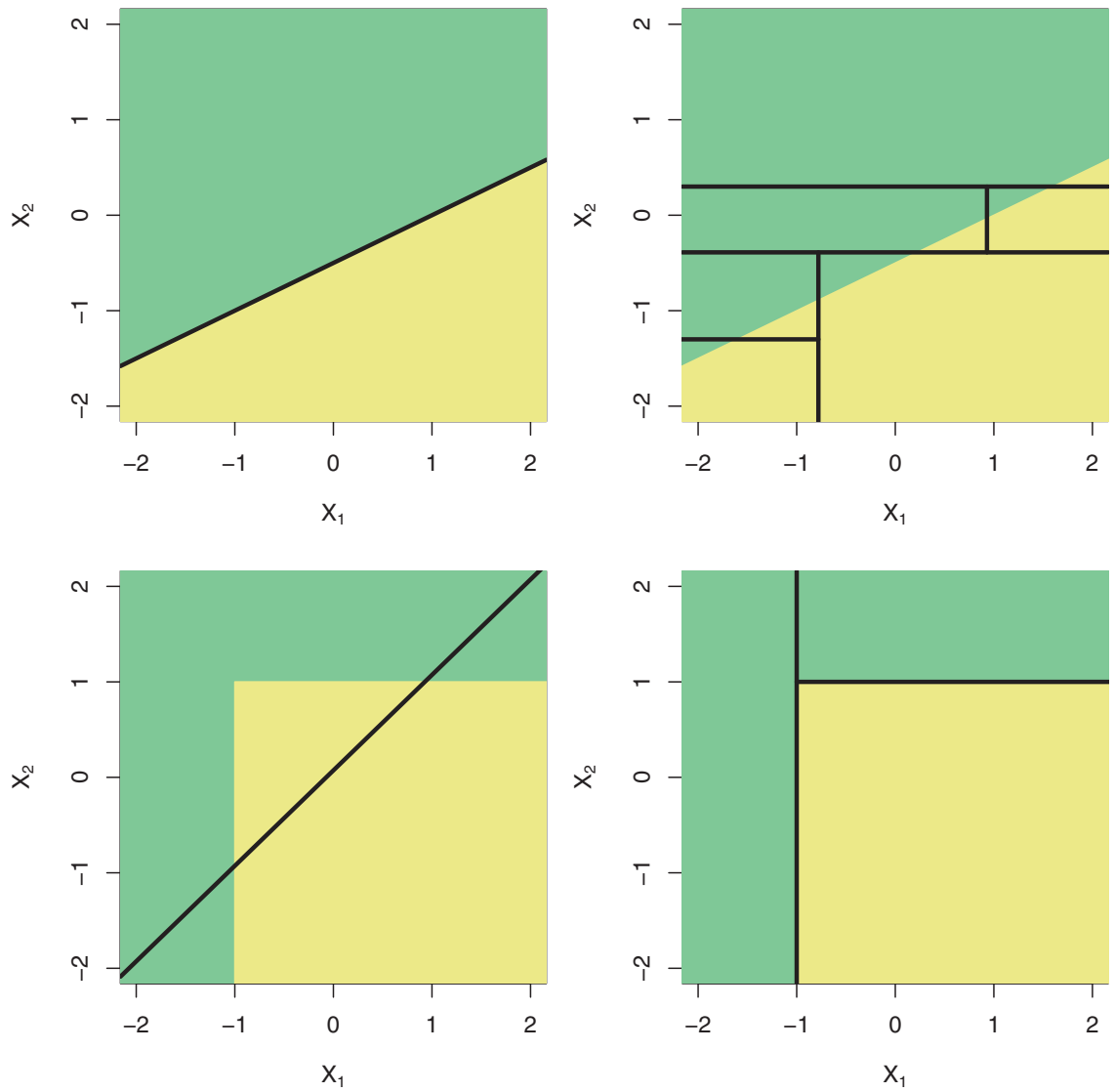


FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

- bagging, random forests, boosting use trees as building blocks to construct more powerful predictions.

bagging:

- $\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$
- trees are grown deep and are not pruned.

out-of-bag (oob) test error estimation:

- $\hat{f}^{*b}(x)$ is (roughly) using $\frac{2}{3}$ of the data
- rest $\frac{1}{3}$ of the data = oob observations for this tree
- cross validation?
  - how many trees didn't use the ith observation?
  - predict the response to the ith observation based on trees that didn't use it
  - oob error $\simeq$ estimate of the test error

bagging:

- improves prediction accuracy
- loses interpretability

bagging: importance of each predictor?

- remove all splits over a predictor for one tress. how much does the rss or gini index increase?
- averaged over all B trees.

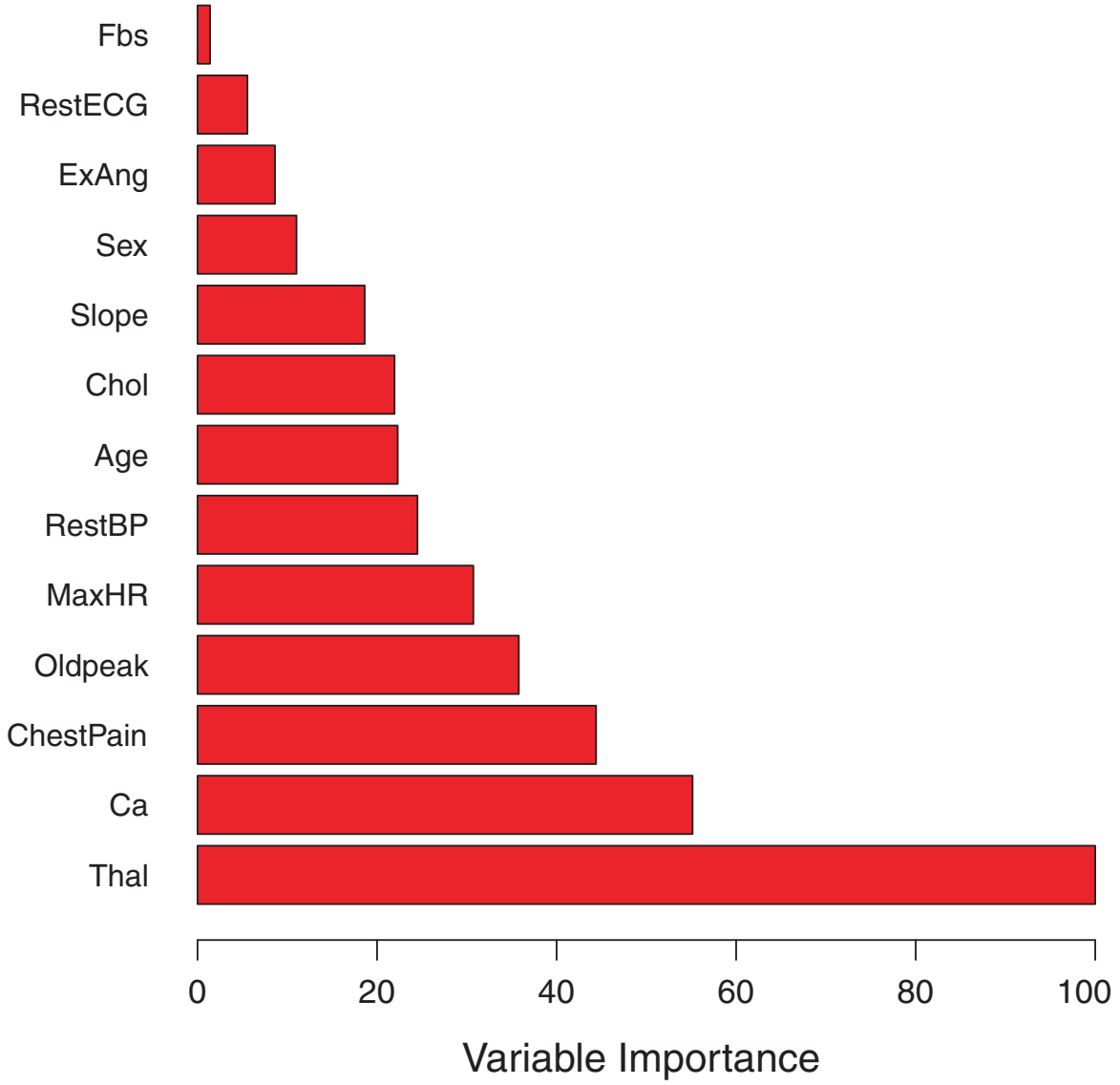


FIGURE 8.9. *A variable importance plot for the* `Heart` *data. Variable importance is computed using the mean decrease in Gini index, and expressed relative to the maximum.*

random forests:

- bagging: highly correlated trees
- random forests: de-correlate trees:
  - when growing a tree, for each possible new split, select random sample of m predictors
  - total number of predictors $= p$
  - typically $m \sim \sqrt{p}$

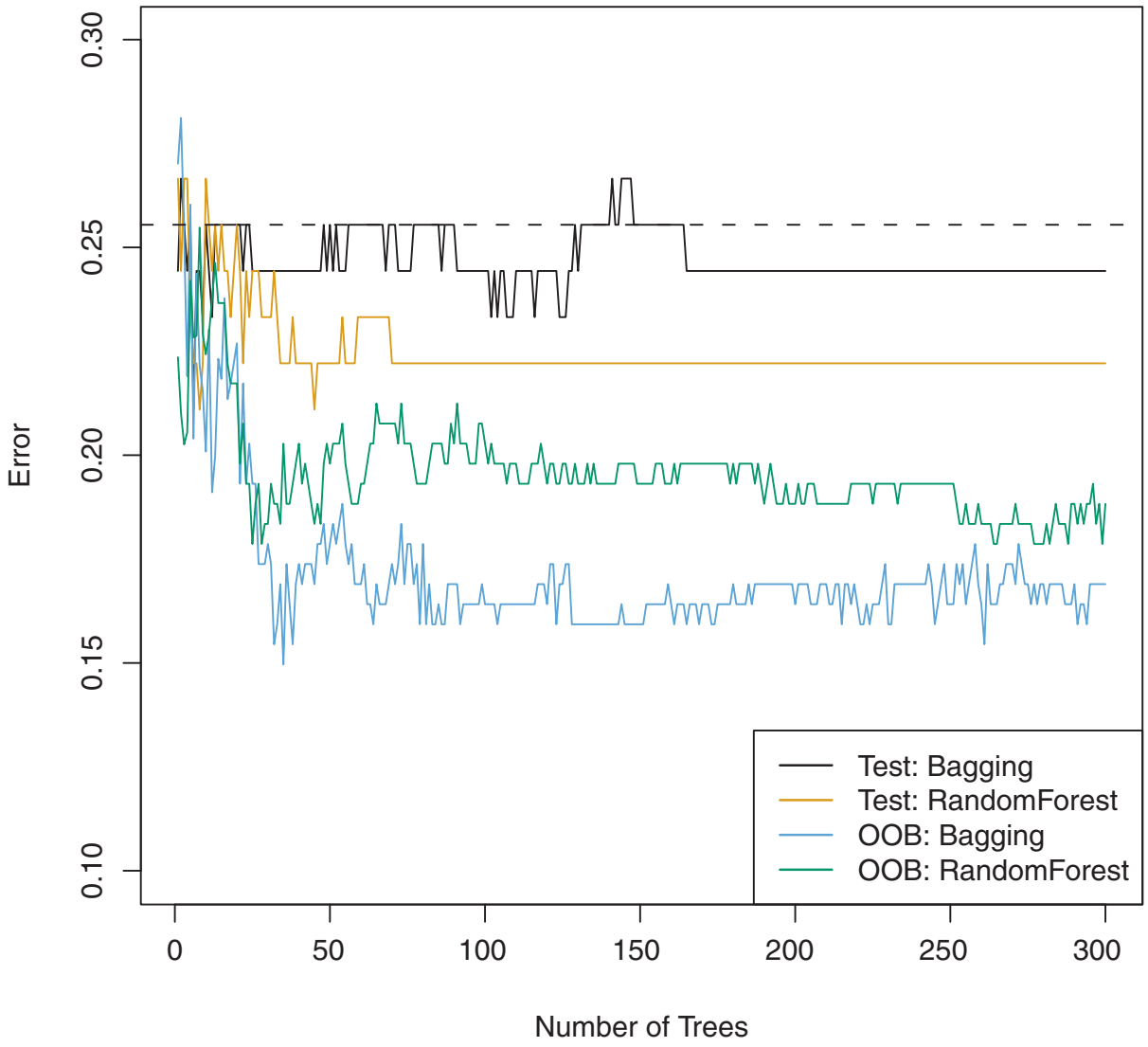


FIGURE 8.8. *Bagging and random forest results for the* `Heart` *data. The test error (black and orange) is shown as a function of* B*, the number of bootstrapped training sets used. Random forests were applied with* $m = \sqrt{p}$*. The dashed line indicates the test error resulting from a single classification tree. The green and blue traces show the OOB error, which in this case is considerably lower.*

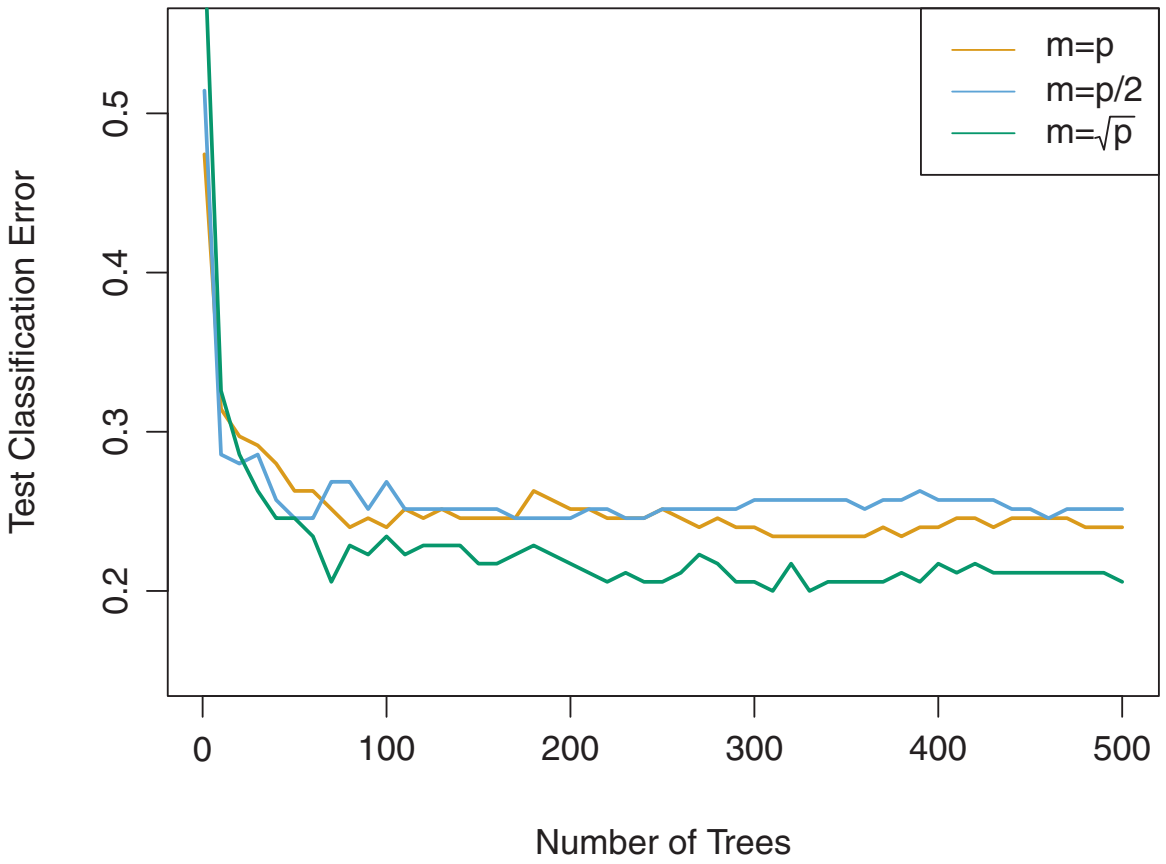


FIGURE 8.10. *Results from random forests for the 15-class gene expression data set with $p = 500$ predictors. The test error is displayed as a function of the number of trees. Each colored line corresponds to a different value of m, the number of predictors available for splitting at each interior tree node. Random forests ($m < p$) lead to a slight improvement over bagging ($m = p$). A single classification tree has an error rate of 45.7%.*

boosting:

- trees are learned one after the other
- each tree is fitted to a modified dataset
- each tree has only a few d terminal nodes:
  - large tree $\rightarrow$ overfitting
  - small tree $\sim$ weak learner and low flexibility
    - combine many weak classifiers $\rightarrow$ a powerful comittee

Algorithm 8.2 *Boosting for Regression Trees*

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.

2. For $b = 1, 2, \ldots, B$, repeat:

   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).

   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x). \tag{8.10}$$

   (c) Update the residuals,

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i). \tag{8.11}$$

3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{8.12}$$

boosting:

- large B leads to over fitting but very slowly
- λ (aka learning rate) $\simeq 0.1, 0.001$.
  - small λ means larger B
- number of splits $= d \sim 1, 2$
  - for $d = 1$, the boosted ensemble is fitting an additive model.

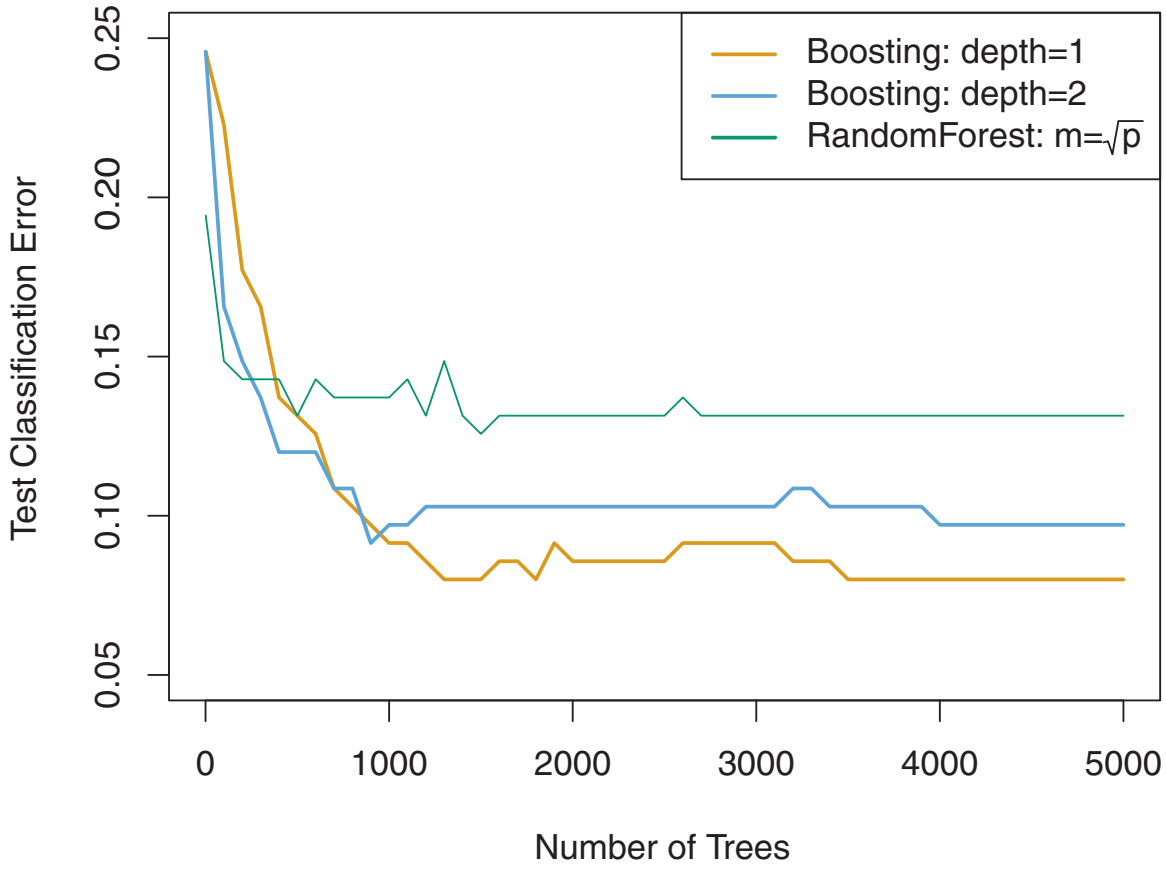


FIGURE 8.11. *Results from performing boosting and random forests on the 15-class gene expression data set in order to predict* cancer *versus* normal. *The test error is displayed as a function of the number of trees. For the two boosted models,* $\lambda = 0.01$. *Depth-1 trees slightly outperform depth-2 trees, and both outperform the random forest, although the standard errors are around 0.02, making none of these differences significant. The test error rate for a single tree is 24 %.*

chapter 10: unsupervised learning

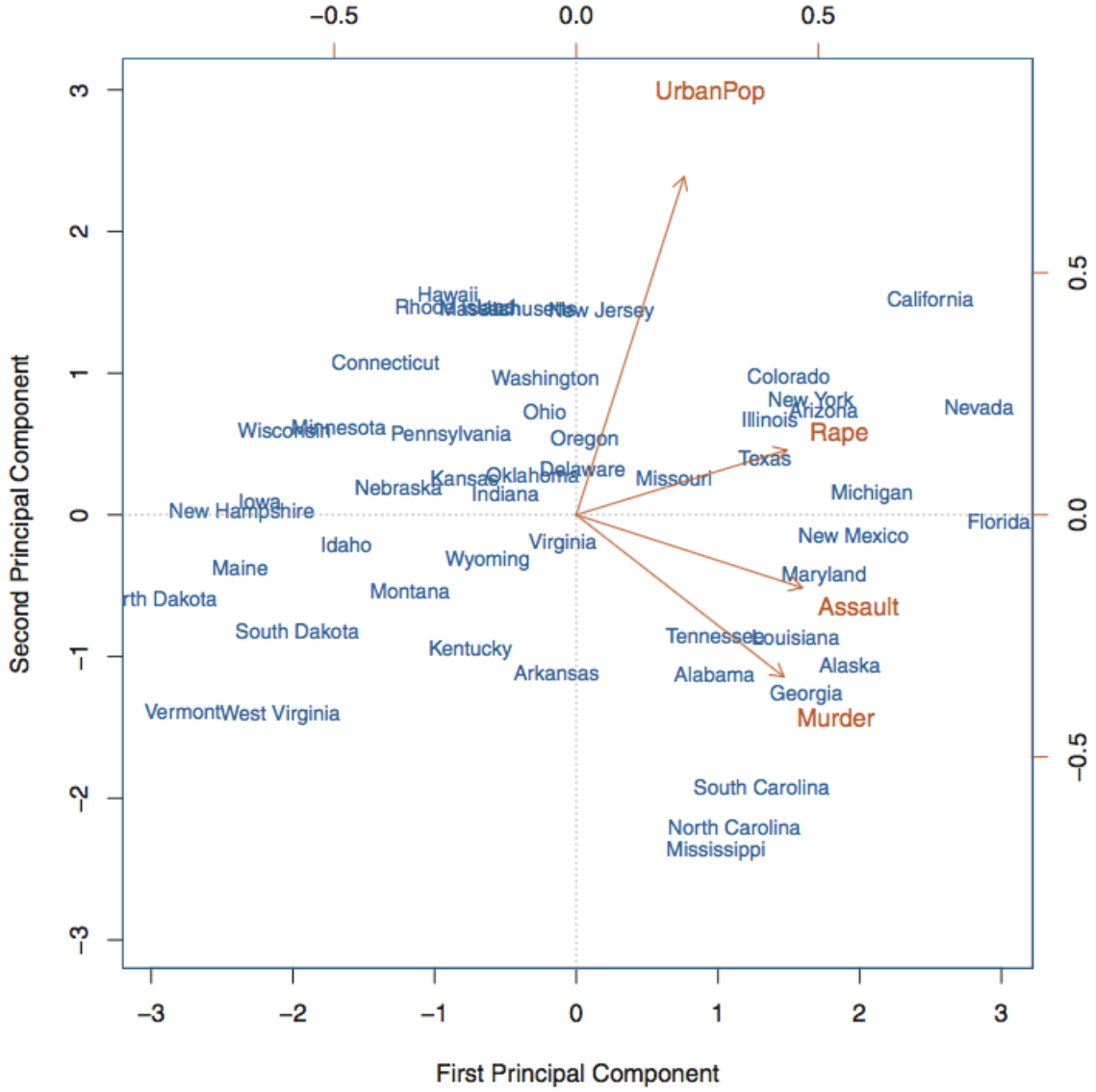


FIGURE 10.1. *The first two principal components for the* `USArrests` *data. The blue state names represent the scores for the first two principal components. The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for* `Rape` *on the first component is* 0.54*, and its loading on the second principal component* 0.17 *(the word* `Rape` *is centered at the point* $(0.54, 0.17)$*). This figure is known as a biplot, because it displays both the principal component scores and the principal component loadings.*

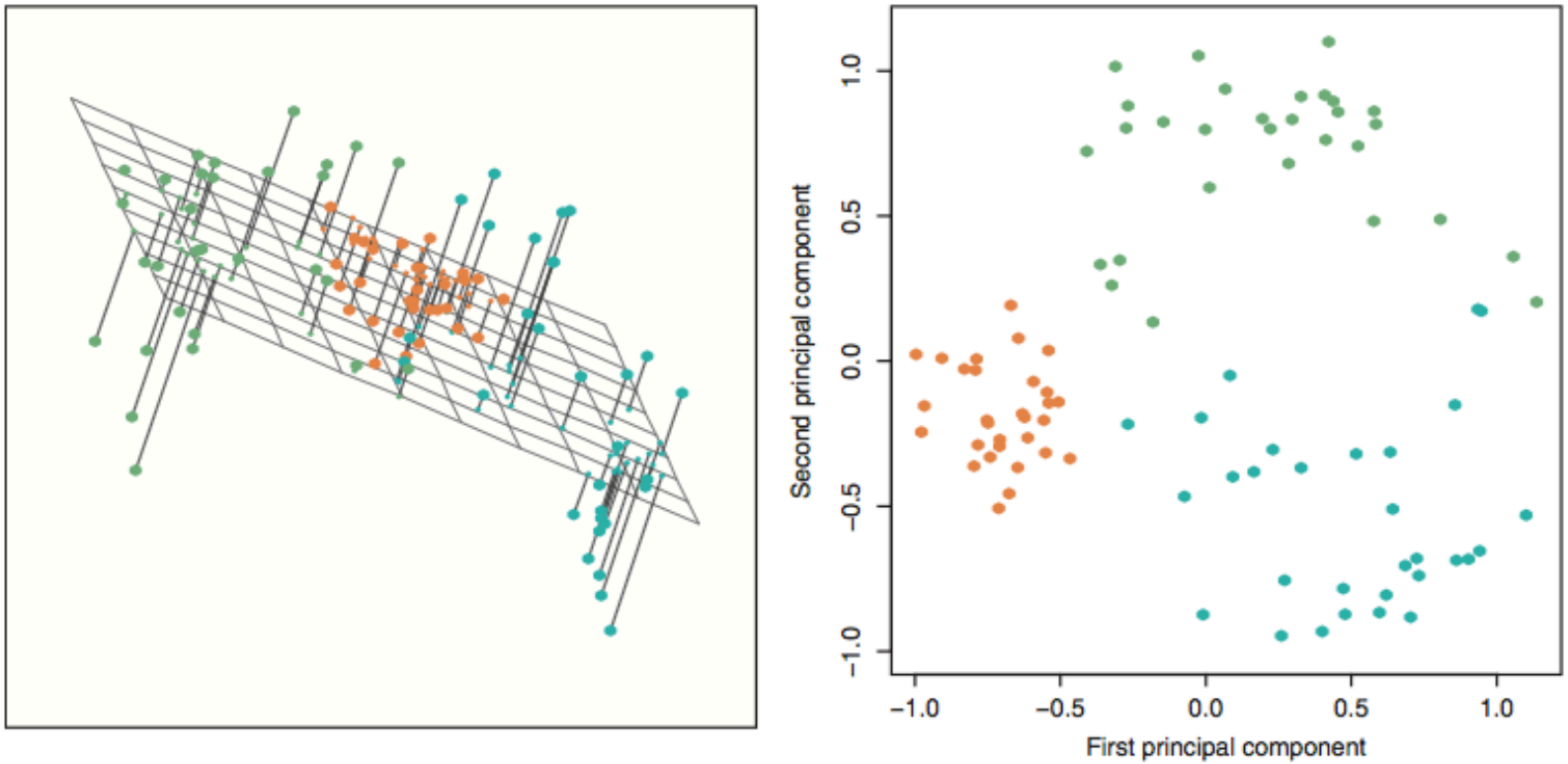


FIGURE 10.2. *Ninety observations simulated in three dimensions.* Left: *the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane.* Right: *the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.*

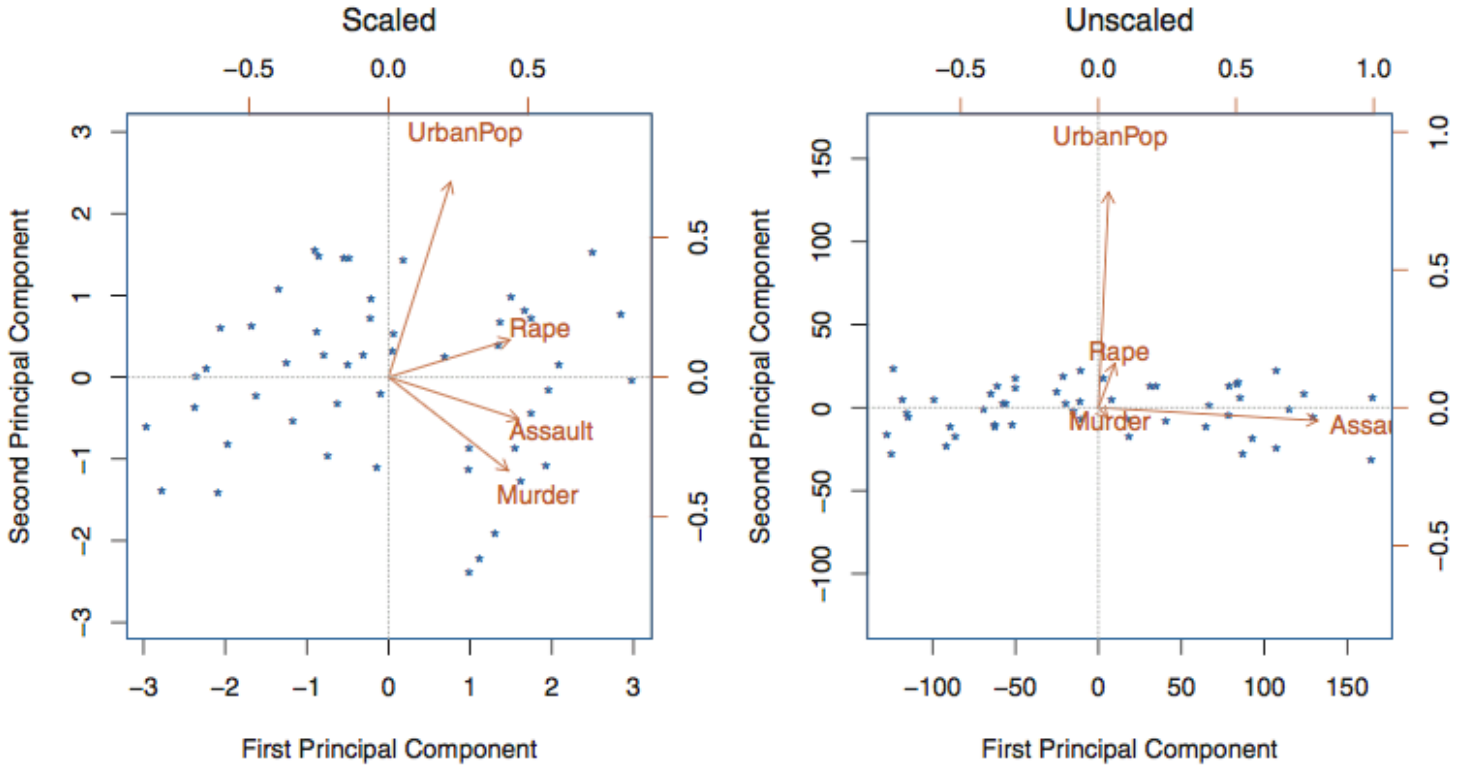


FIGURE 10.3. *Two principal component biplots for the* `USArrests` *data.* Left: *the same as Figure 10.1, with the variables scaled to have unit standard deviations.* Right: *principal components using unscaled data.* `Assault` *has by far the largest loading on the first principal component because it has the highest variance among the four variables. In general, scaling the variables to have standard deviation one is recommended.*

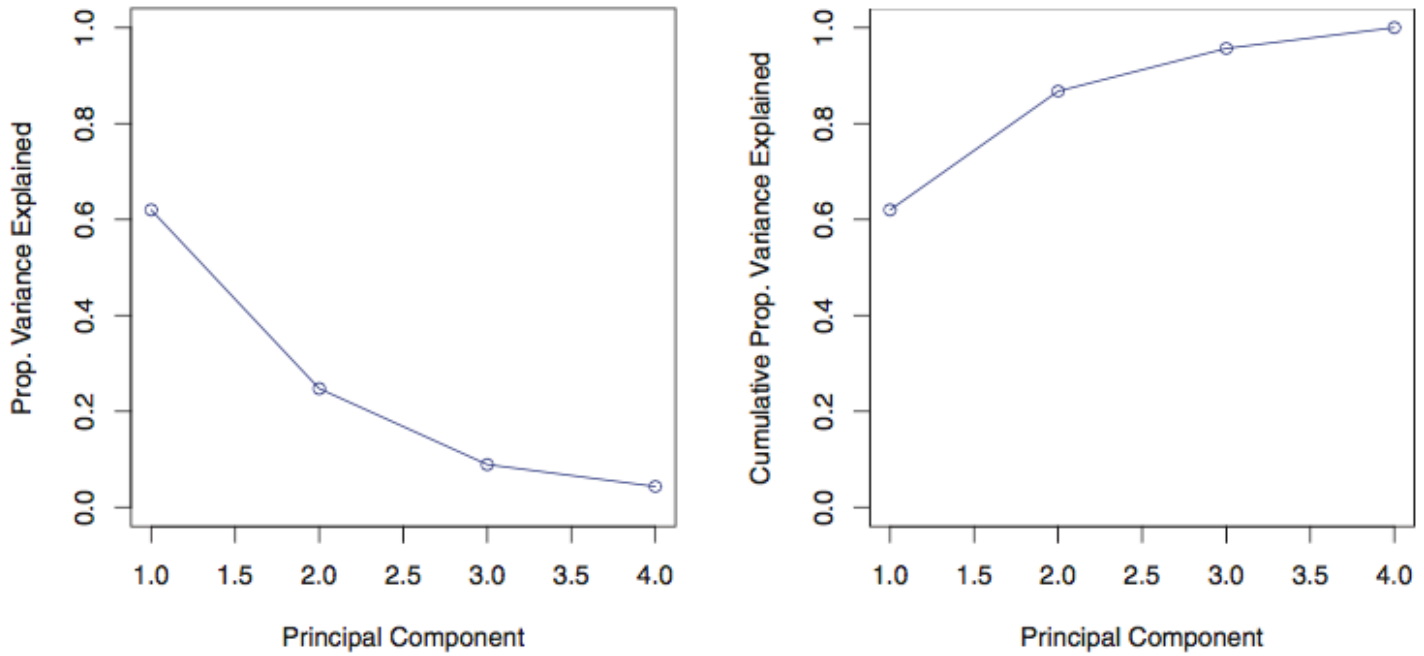


FIGURE 10.4. Left: *a scree plot depicting the proportion of variance explained by each of the four principal components in the* `USArrests` *data.* Right: *the cumulative proportion of variance explained by the four principal components in the* `USArrests` *data.*

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}.$$

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2,$$

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}.$$

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}.$$

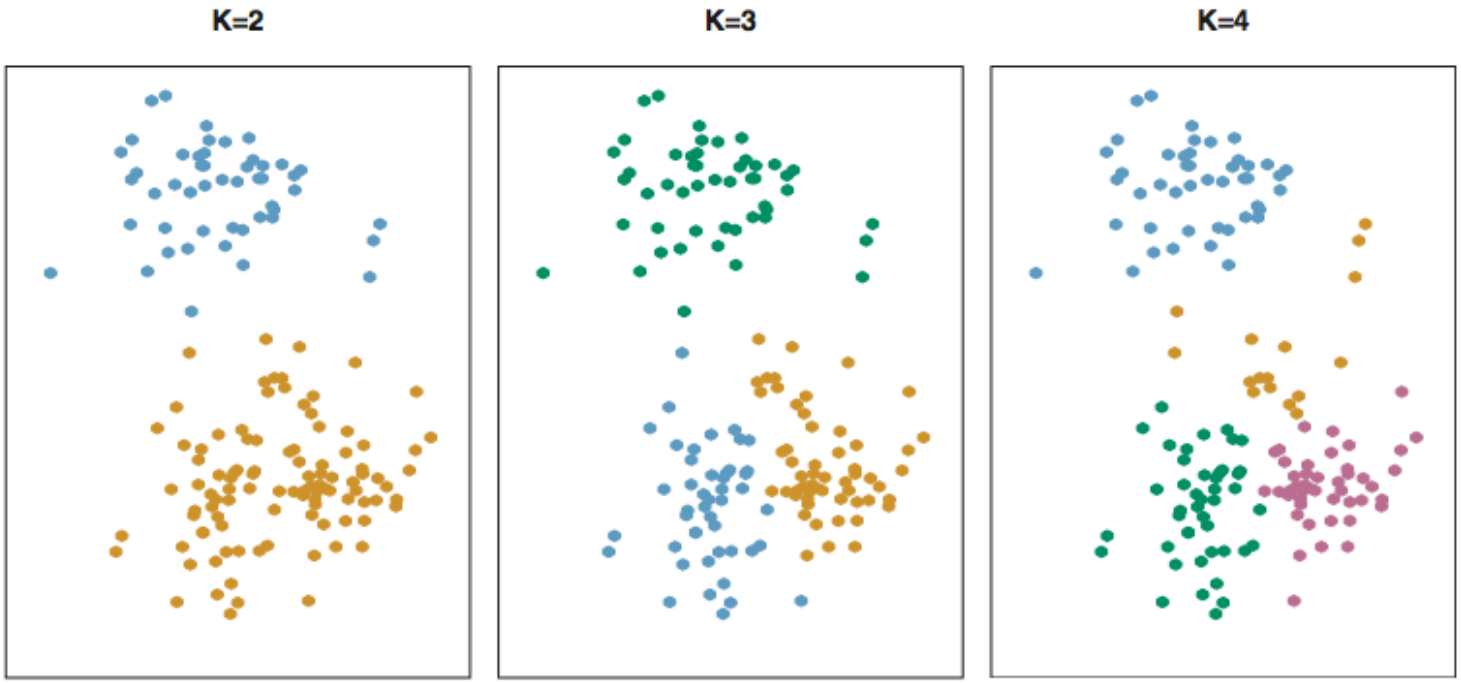


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.*

Algorithm 10.1 *K-Means Clustering*

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2,$$

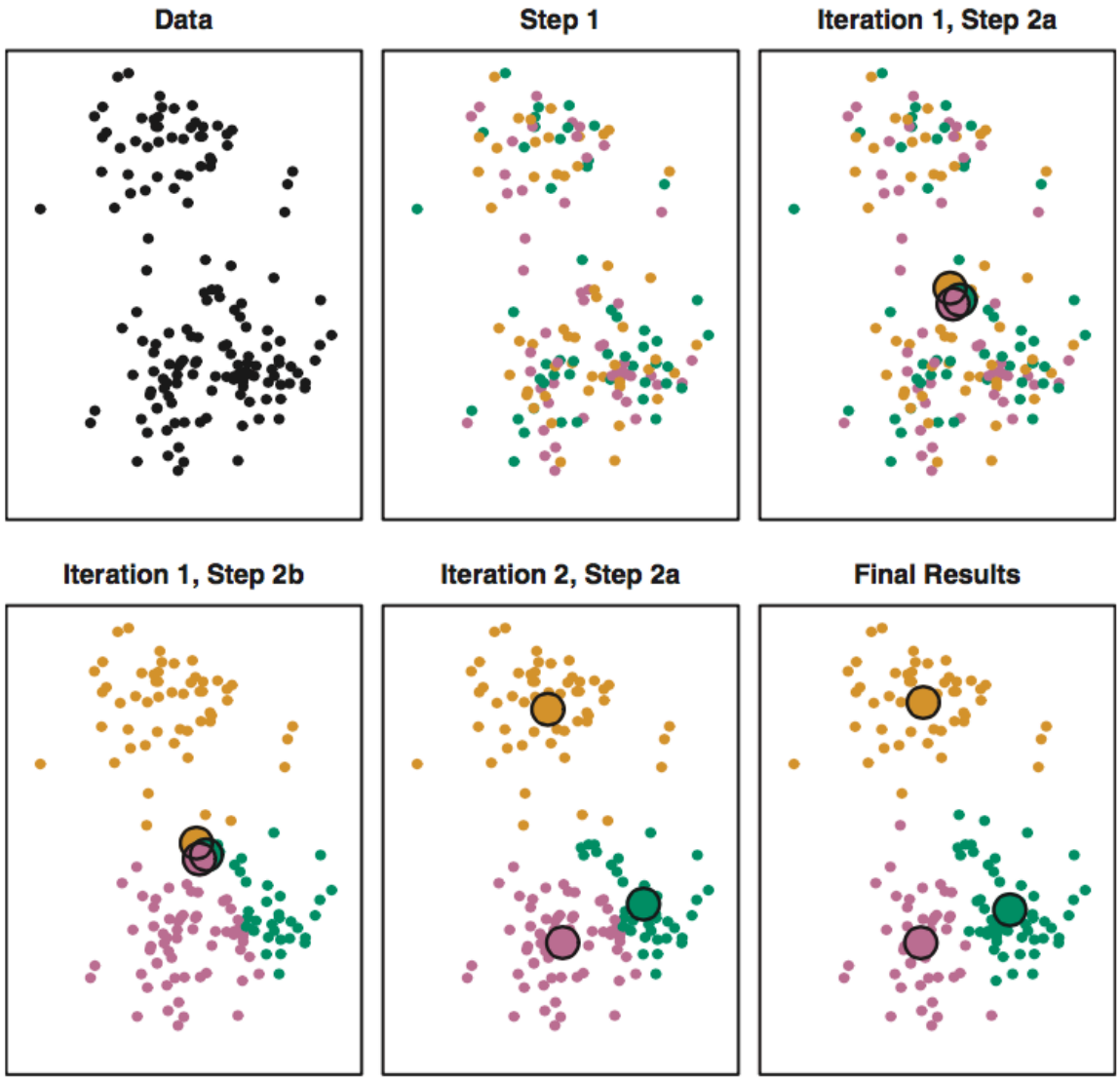


FIGURE 10.6. *The progress of the K-means algorithm on the example of Figure 10.5 with $K=3$.* Top left: *the observations are shown.* Top center: *in Step 1 of the algorithm, each observation is randomly assigned to a cluster.* Top right: *in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.* Bottom left: *in Step 2(b), each observation is assigned to the nearest centroid.* Bottom center: *Step 2(a) is once again performed, leading to new cluster centroids.* Bottom right: *the results obtained after ten iterations.*

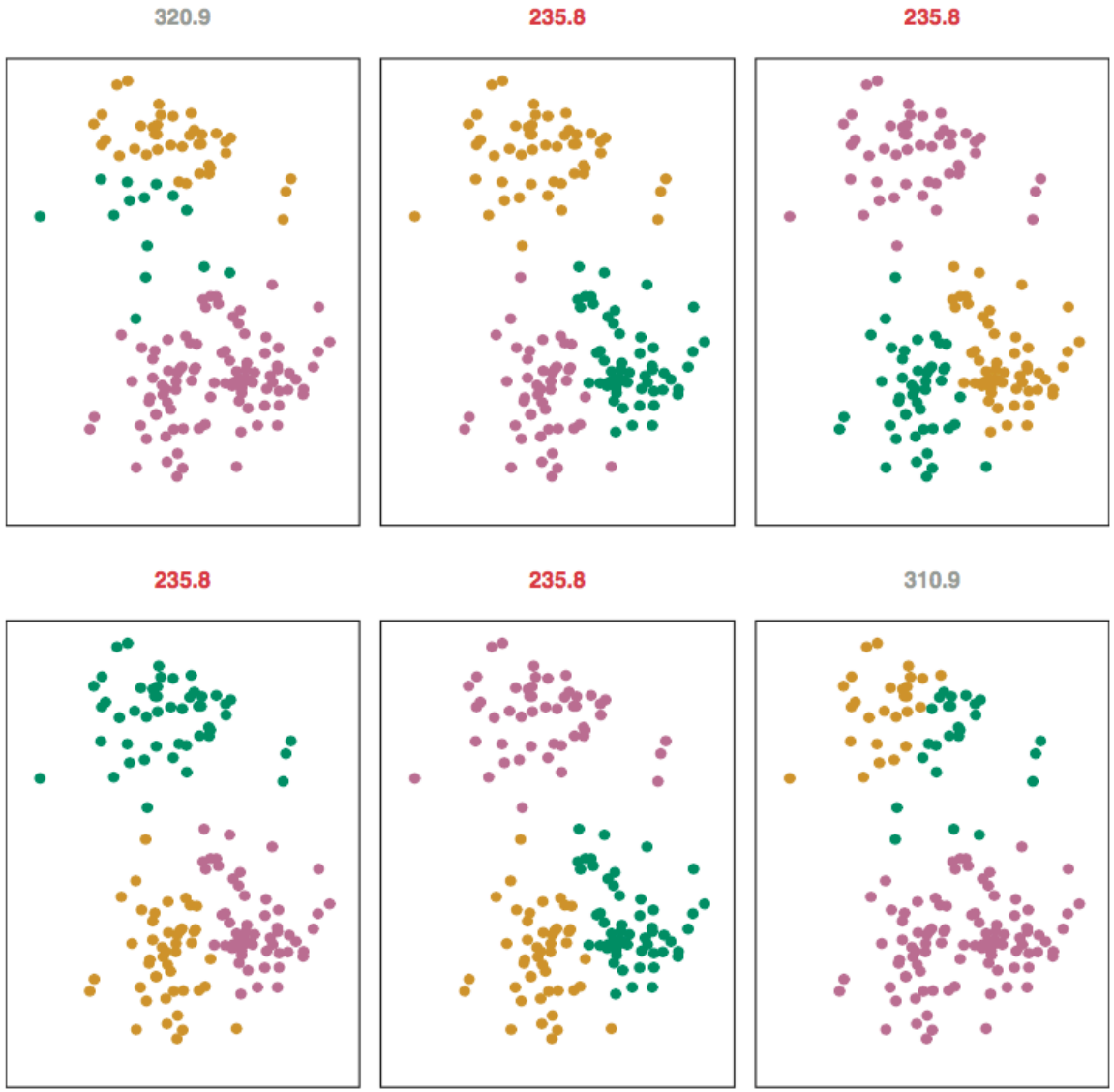


FIGURE 10.7. *K-means clustering performed six times on the data from Figure 10.5 with $K = 3$, each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8.*

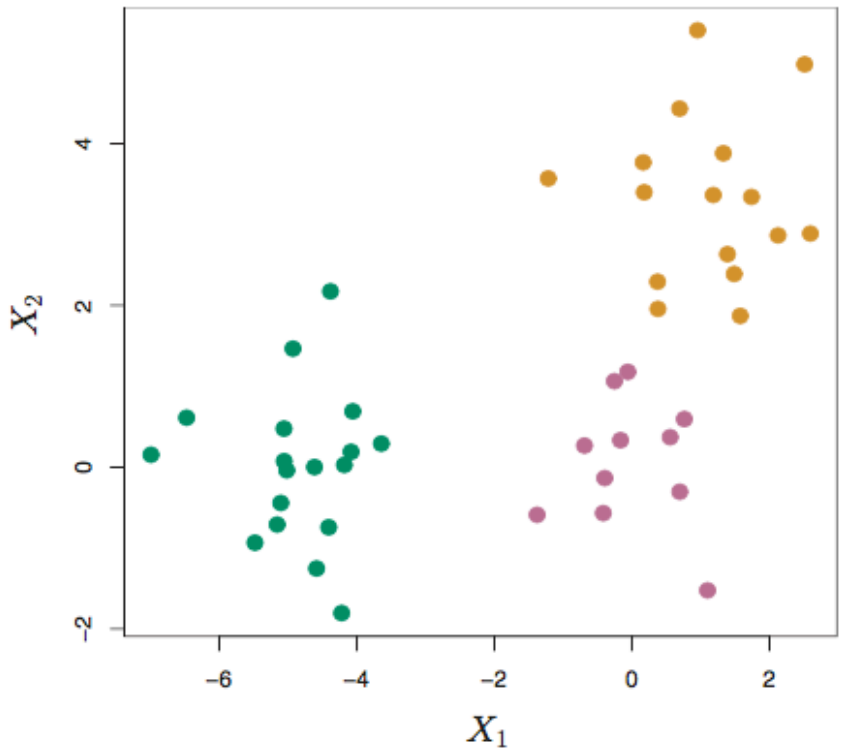


FIGURE 10.8. *Forty-five observations generated in two-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.*

hierarchical clustering:

- tree based representation: no need to pre specify the number of clusters
- agglomerative/bottom-up clustering:
  - leaf: each observation
  - fuse similar leaves into branches
  - fuse branches...
  - early fusion $\simeq$ similar groups of observations
  - later fusion $\neq$ similar groups of observations

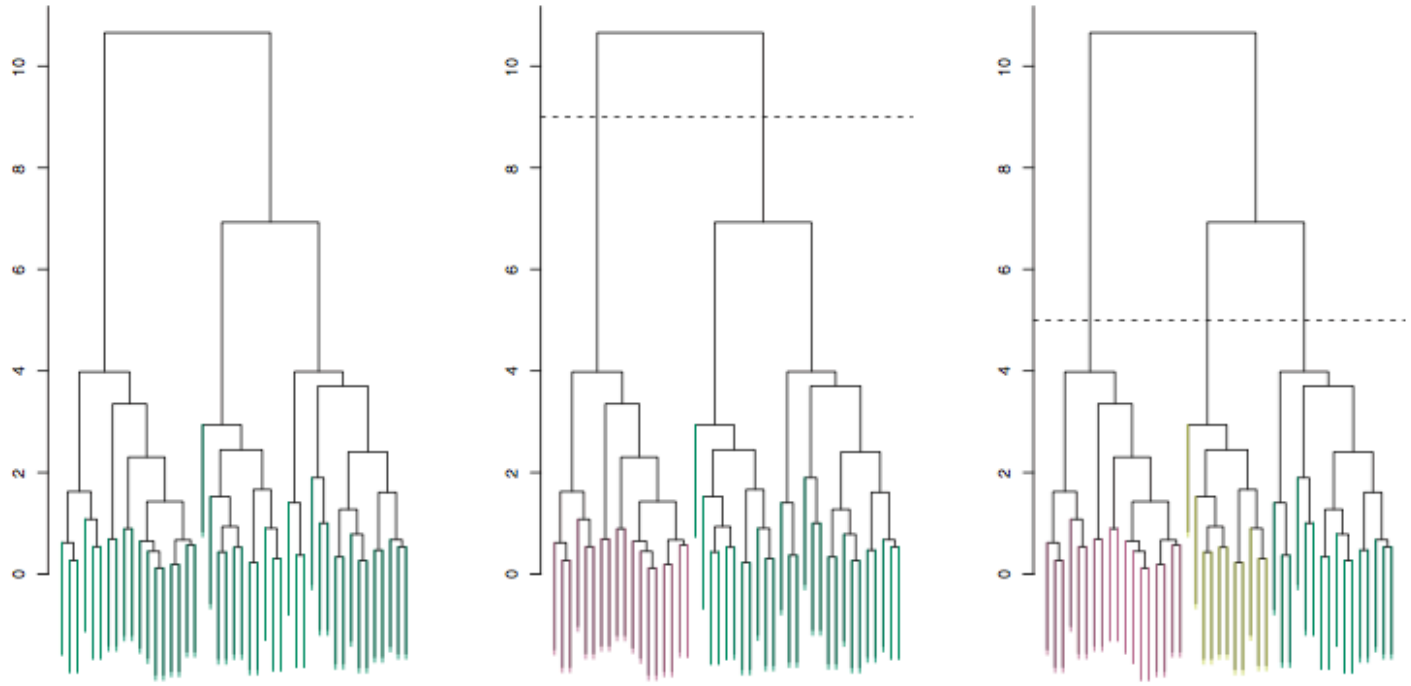


FIGURE 10.9. Left: *dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.* Right: *the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.*

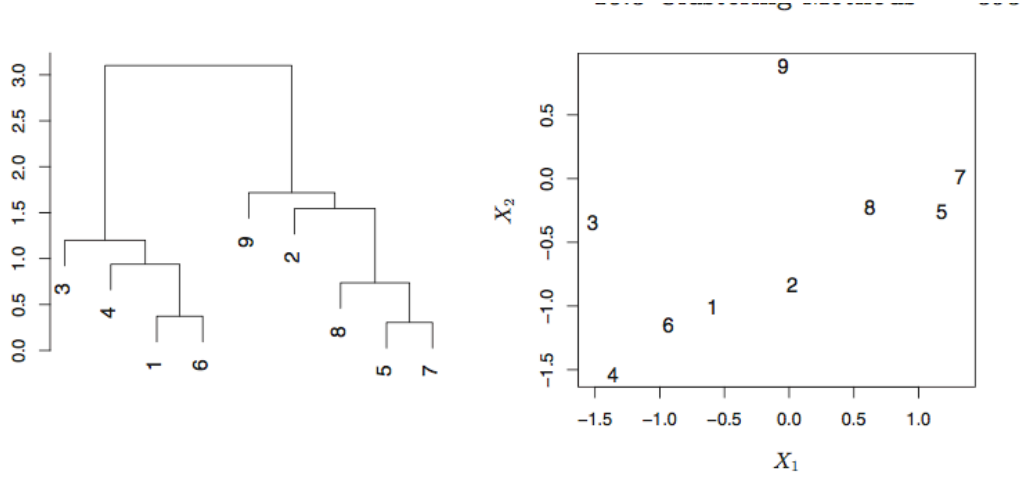


FIGURE 10.10. *An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.* Left: *a dendrogram generated using Euclidean distance and complete linkage. Observations* 5 *and* 7 *are quite similar to each other, as are observations* 1 *and* 6. *However, observation* 9 *is* no more similar to *observation* 2 *than it is to observations* $8, 5,$ *and* 7, *even though observations* 9 *and* 2 *are close together in terms of horizontal distance. This is because observations* $2, 8, 5,$ *and* 7 *all fuse with observation* 9 *at the same height, approximately* 1.8. Right: *the raw data used to generate the dendrogram can be used to confirm that indeed, observation* 9 *is no more similar to observation* 2 *than it is to observations* $8, 5,$ *and* 7.

- 5 and 7 similar ?
- 9 and 2 similar?
- 2^{n-1} possible reorderings of the dendugram

- hierarchical structure may fail:
  - group of people: 50-50 split of male and female
  - group of people: even split of american, japanese, and french
- best split into two groups?
- best split into three groups?

Algorithm 10.2 *Hierarchical Clustering*

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

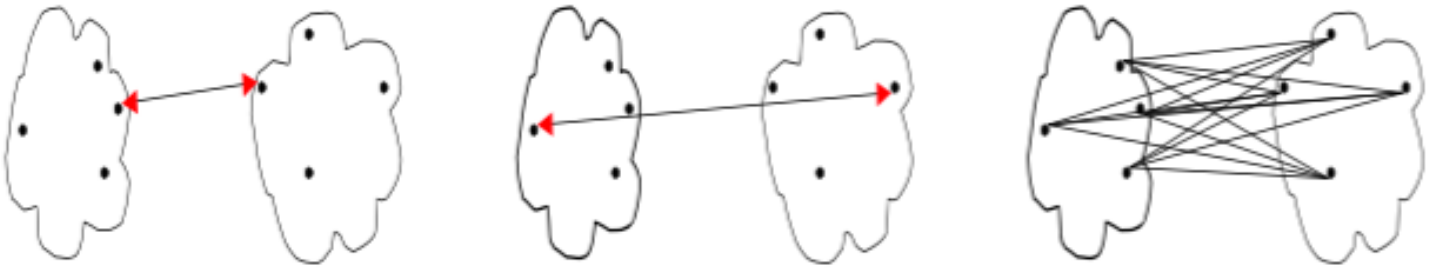
2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

TABLE 10.2. *A summary of the four most commonly-used types of linkage in hierarchical clustering.*



(a) MIN (b) MAX (c) Group Average

Figure 5.26. Definition of Cluster Proximity

point	x coordinate	y coordinate
p1	0.4005	0.5306
p2	0.2148	0.3854
p3	0.3457	0.3156
p4	0.2652	0.1875
p5	0.0789	0.4139
p6	0.4548	0.3022

Table 5.6. X-Y coordinates of six points.

	p1	p2	p3	p4	p5	p6
p1	0.0000	0.2357	0.2218	0.3688	0.3421	0.2347
p2	0.2357	0.0000	0.1483	0.2042	0.1388	0.2540
p3	0.2218	0.1483	0.0000	0.1513	0.2843	0.1100
p4	0.3688	0.2042	0.1513	0.0000	0.2932	0.2216
p5	0.3421	0.1388	0.2843	0.2932	0.0000	0.3921
p6	0.2347	0.2540	0.1100	0.2216	0.3921	0.0000

Table 5.7. Distance Matrix for Six Points

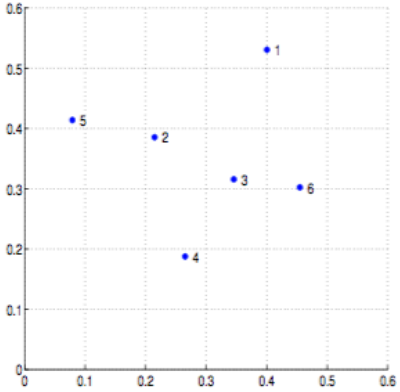
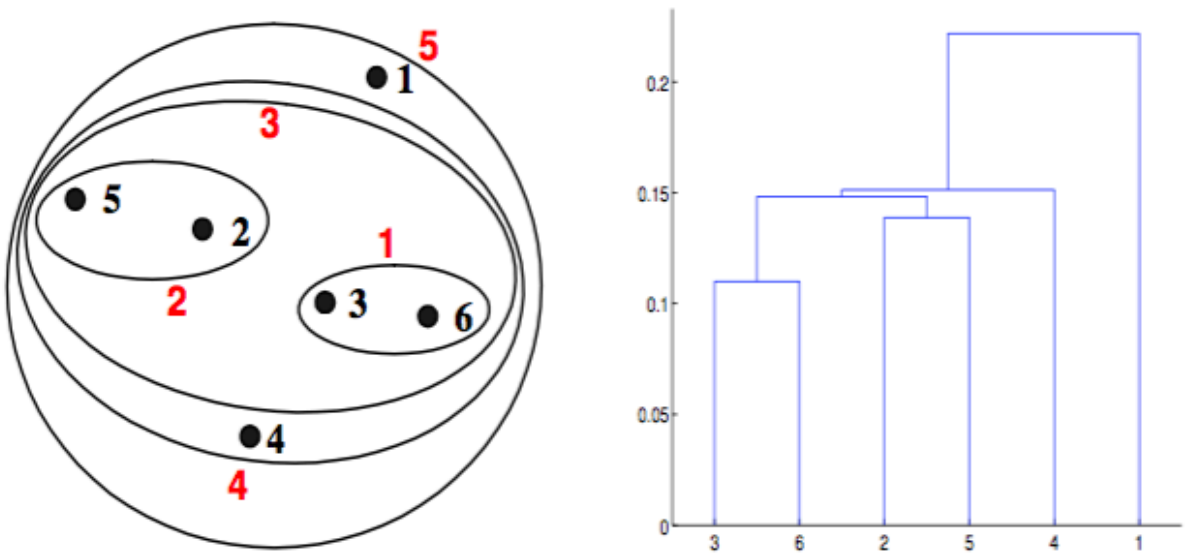


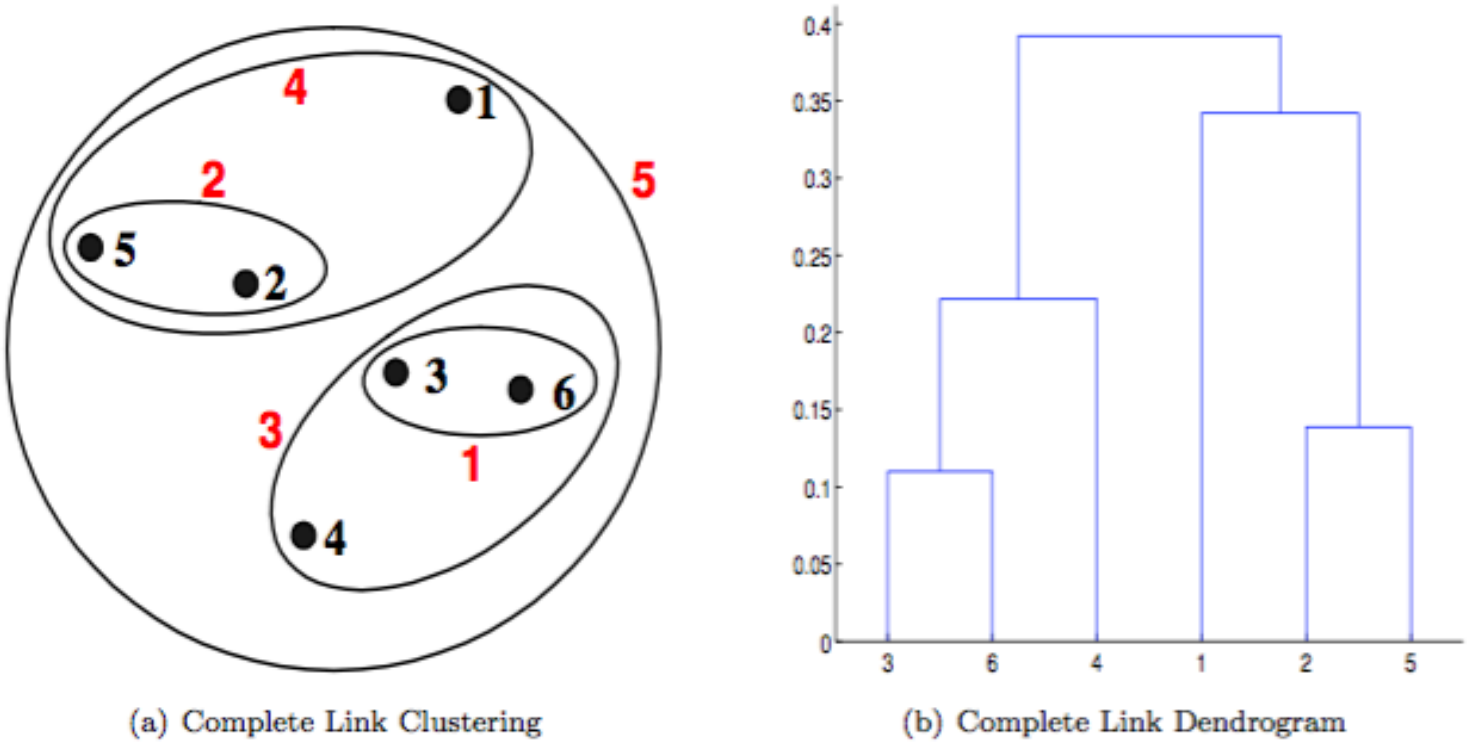
Figure 5.24. Set of Six Two-dimensional Points.



(a) Single Link Clustering (b) Single Link Dendrogram

Figure 5.27. Single Link Clustering of Six Points.

reflects the distance of the two clusters. For instance, from Table 5.7, we see that the distance between points 3 and 6 is 0.11, and that is the height at which they are joined into one cluster in the dendrogram. As another example, the distance between clusters $\{3,6\}$ and $\{2,5\}$ is given by $dist(\{3,6\},\{2,5\}) = min(dist(3,2), dist(6,2), dist(3,5), dist(6,5)) = min(0.1483, 0.2540, 0.2843, 0.3921) = 0.1483$.

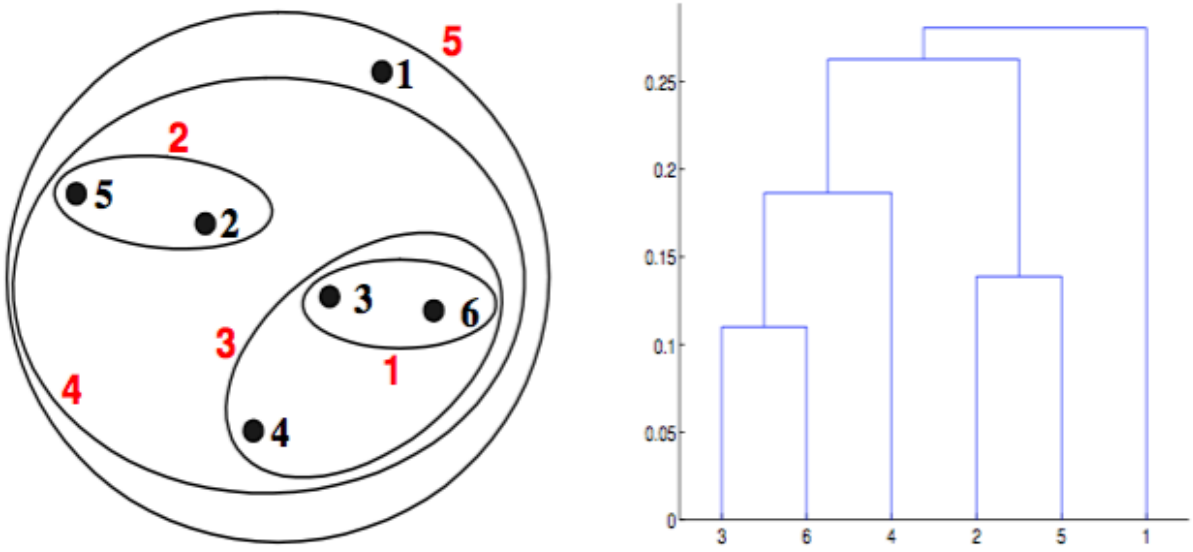


(a) Complete Link Clustering

(b) Complete Link Dendrogram

Figure 5.28. Complete Link Clustering of Six Points.

Figure 5.29 shows the results of applying group average to the sample data set of six points. To illustrate how group average works, we calculate the distance between some clusters. $dist(\{3,6,4\},\{1\}) = (0.2218 + 0.3688 + 0.2347)/(3 * 1) = 0.2751$. $dist(\{2,5\},\{1\}) = (0.2357 + 0.3421)/(2 * 1) = 0.2889$. $dist(\{3,6,4\},\{2,5\}) = (0.1483+0.2843+0.2540+0.3921+0.2042+0.2932)/(6*1) = 0.2637$. Because $dist(\{3,6,4\},\{2,5\})$ is smaller than $dist(\{3,6,4\},\{1\})$ and $dist(\{2,5\},\{1\})$, these two clusters are merged at the fourth stage.



(a) Group Average Clustering

(b) Group Average Dendrogram

Figure 5.29. Group Average Clustering of Six Points.

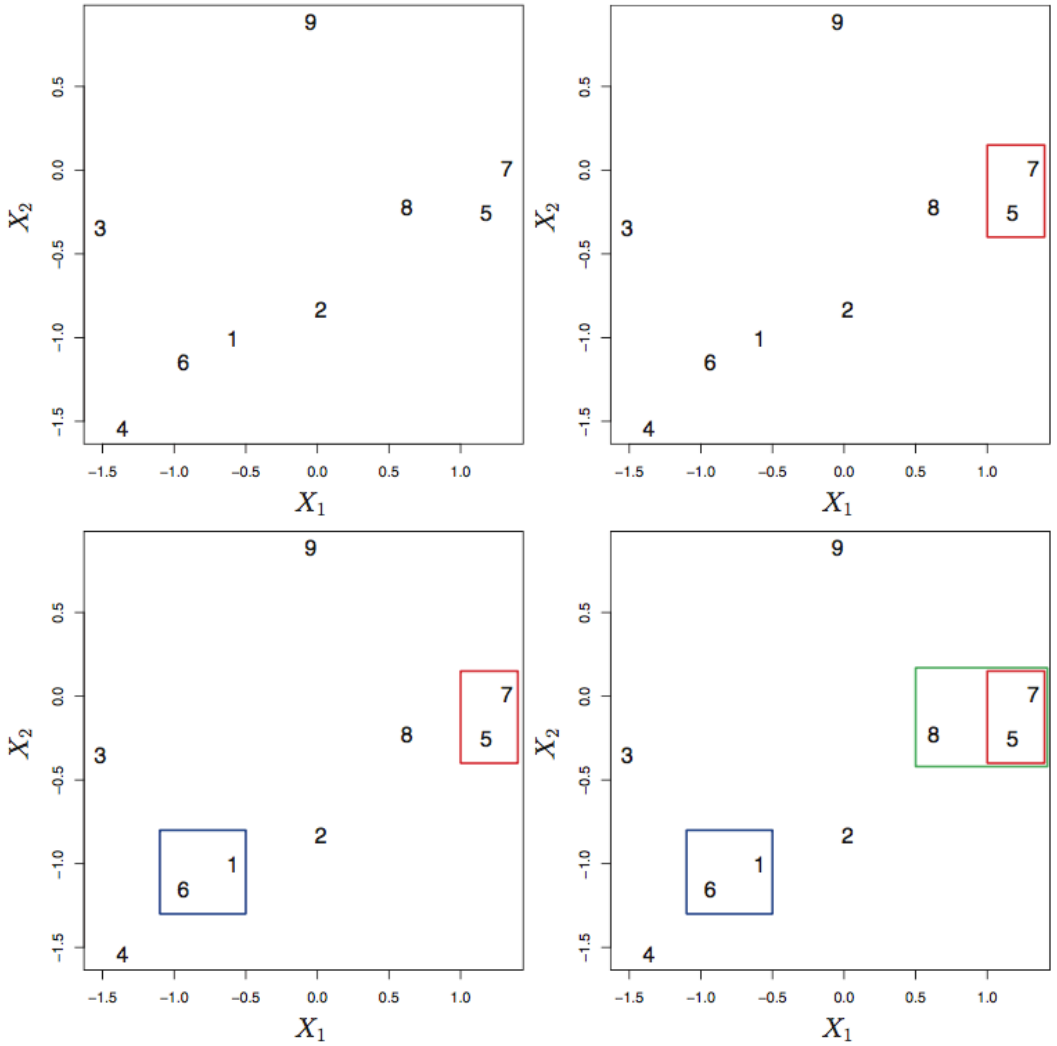


FIGURE 10.11. *An illustration of the first few steps of the hierarchical clustering algorithm, using the data from Figure 10.10, with complete linkage and Euclidean distance.* Top Left: *initially, there are nine distinct clusters,* $\{1\}, \{2\}, \ldots, \{9\}$. Top Right: *the two clusters that are closest together,* $\{5\}$ *and* $\{7\}$, *are fused into a single cluster.* Bottom Left: *the two clusters that are closest together,* $\{6\}$ *and* $\{1\}$, *are fused into a single cluster.* Bottom Right: *the two clusters that are closest together using* complete linkage, $\{8\}$ *and the cluster* $\{5, 7\}$, *are fused into a single cluster.*

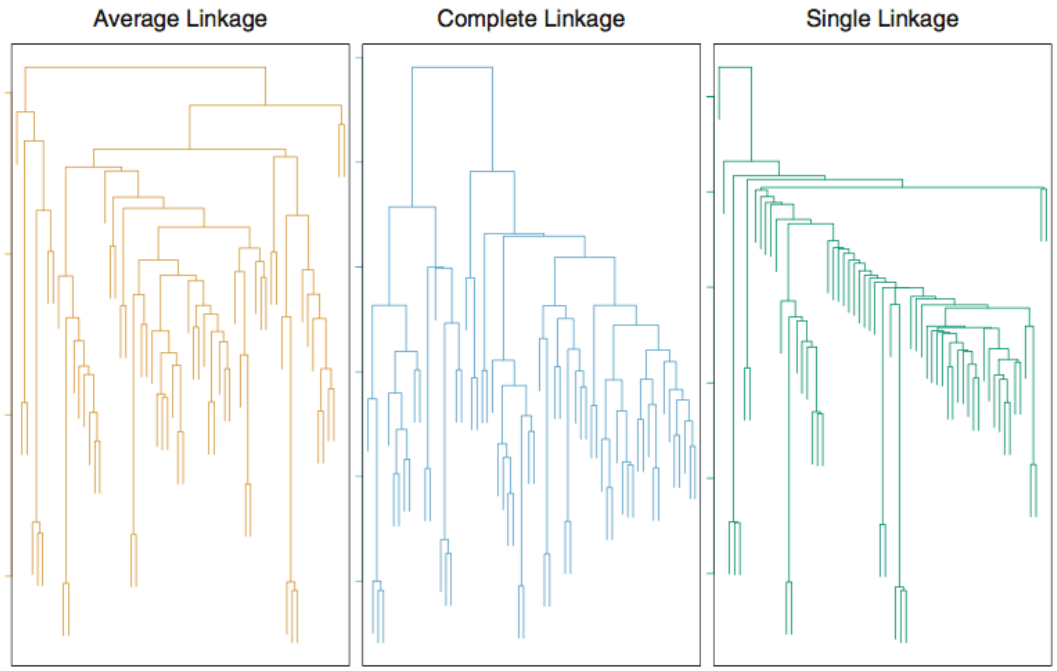


FIGURE 10.12. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

- choice of dissimilarity measure:
  - euclidian distance?
  - correlation-based distance?

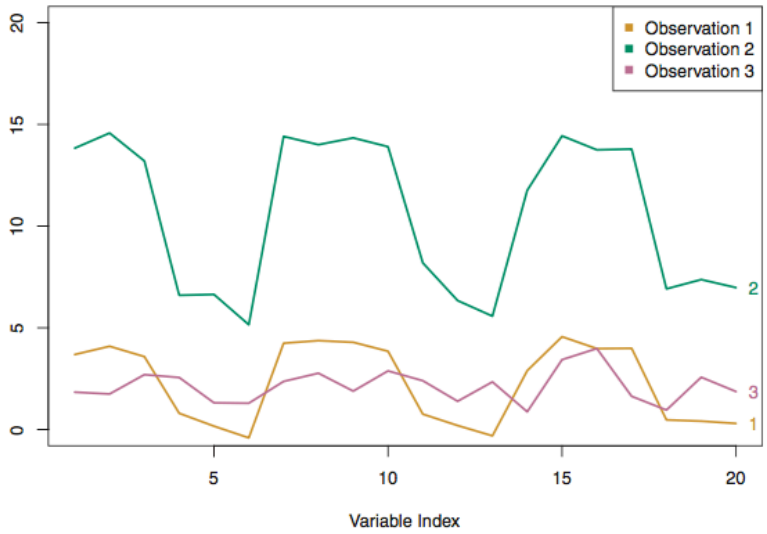


FIGURE 10.13. *Three observations with measurements on 20 variables are shown. Observations 1 and 3 have similar values for each variable and so there is a small Euclidean distance between them. But they are very weakly correlated, so they have a large correlation-based distance. On the other hand, observations 1 and 2 have quite different values for each variable, and so there is a large Euclidean distance between them. But they are highly correlated, so there is a small correlation-based distance between them.*

- whether or not variables should be scaled to have standard deviation one?

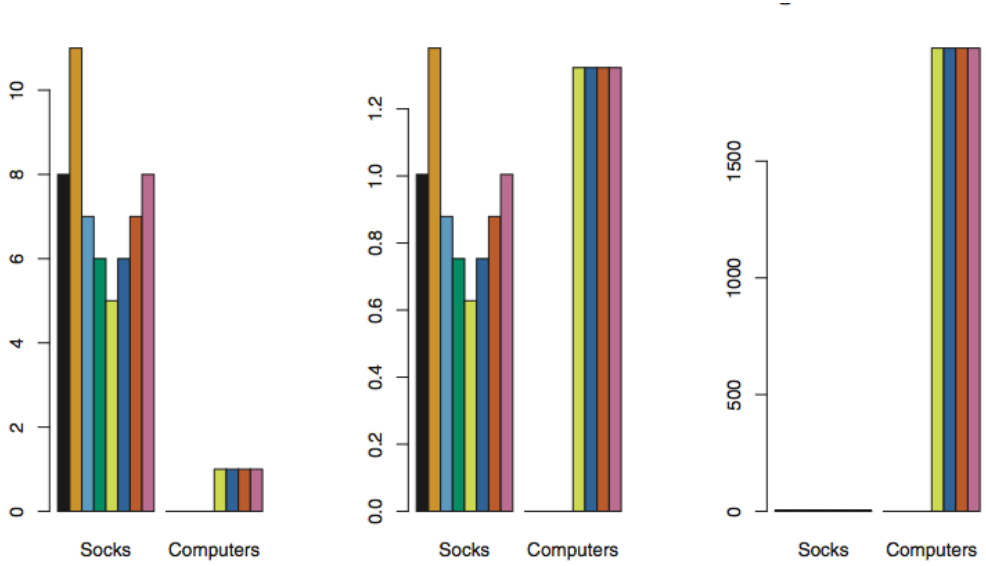


FIGURE 10.14. *An eclectic online retailer sells two items: socks and computers.* Left: *the number of pairs of socks, and computers, purchased by eight online shoppers is displayed. Each shopper is shown in a different color. If inter-observation dissimilarities are computed using Euclidean distance on the raw variables, then the number of socks purchased by an individual will drive the dissimilarities obtained, and the number of computers purchased will have little effect. This might be undesirable, since (1) computers are more expensive than socks and so the online retailer may be more interested in encouraging shoppers to buy computers than socks, and (2) a large difference in the number of socks purchased by two shoppers may be less informative about the shoppers' overall shopping preferences than a small difference in the number of computers purchased.* Center: *the same data is shown, after scaling each variable by its standard deviation. Now the number of computers purchased will have a much greater effect on the inter-observation dissimilarities obtained.* Right: *the same data are displayed, but now the y-axis represents the number of dollars spent by each online shopper on socks and on computers. Since computers are much more expensive than socks, now computer purchase history will drive the inter-observation dissimilarities obtained.*

practical considerations:

- standardardize observations?
- hierarchical clustering:
  - similarity measure?
  - type of linkage?
- how many clusters?

support vector machines

- maximal margin classifier (mmc)
- support vector classifier (svc)
- support vector machine (svm)
- svm vs. logistic
- more than two classes?

maximal margin classifier:

- hyperplane: $\left\{all\ x \in \mathbb{R}^p\ such\ that\ \beta_0 + \beta^\top x = 0\right\}$
- halfspace: $\left\{all\ x \in \mathbb{R}^p\ such\ that\ \beta_0 + \beta^\top x > 0\right\}$
- halfspace: $\left\{all\ x \in \mathbb{R}^p\ such\ that\ \beta_0 + \beta^\top x < 0\right\}$

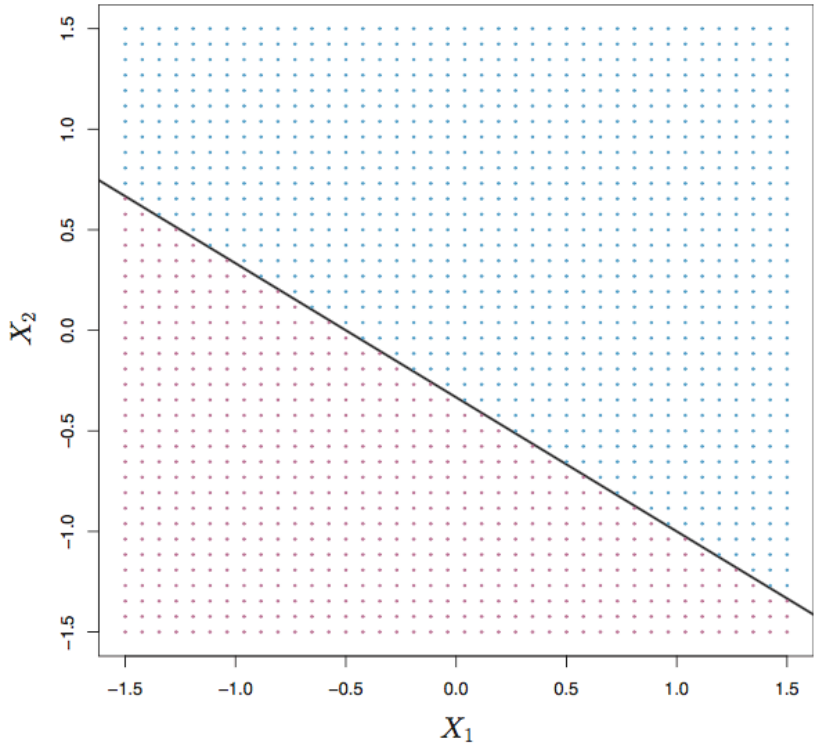


FIGURE 9.1. *The hyperplane* $1 + 2X_1 + 3X_2 = 0$ *is shown. The blue region is the set of points for which* $1 + 2X_1 + 3X_2 > 0$*, and the purple region is the set of points for which* $1 + 2X_1 + 3X_2 < 0$*.*

learning a separating hyperplane: find β_0 and β such that:

- $\beta_0 + \beta^\top x_i > 0$ if $y_i = 1$
- $\beta_0 + \beta^\top x_i < 0$ if $y_i = -1$
- $y_i(\beta_0 + \beta^\top x_i) > 0$ for $(x_1, y_1), \ldots, (x_n, y_n)$

how to classify a new test observation x?

- $\hat{y} = 1$ if $\beta_0 + x^\top \beta > 0$
- $\hat{y} = -1$ if $\beta_0 + x^\top \beta < 0$

magnitude of $f(x) = \beta_0 + \beta^\top x$?

- large $|f(x)| = x$ is far from hyperplane
- small $|f(x)| = x$ is near the hyperplane

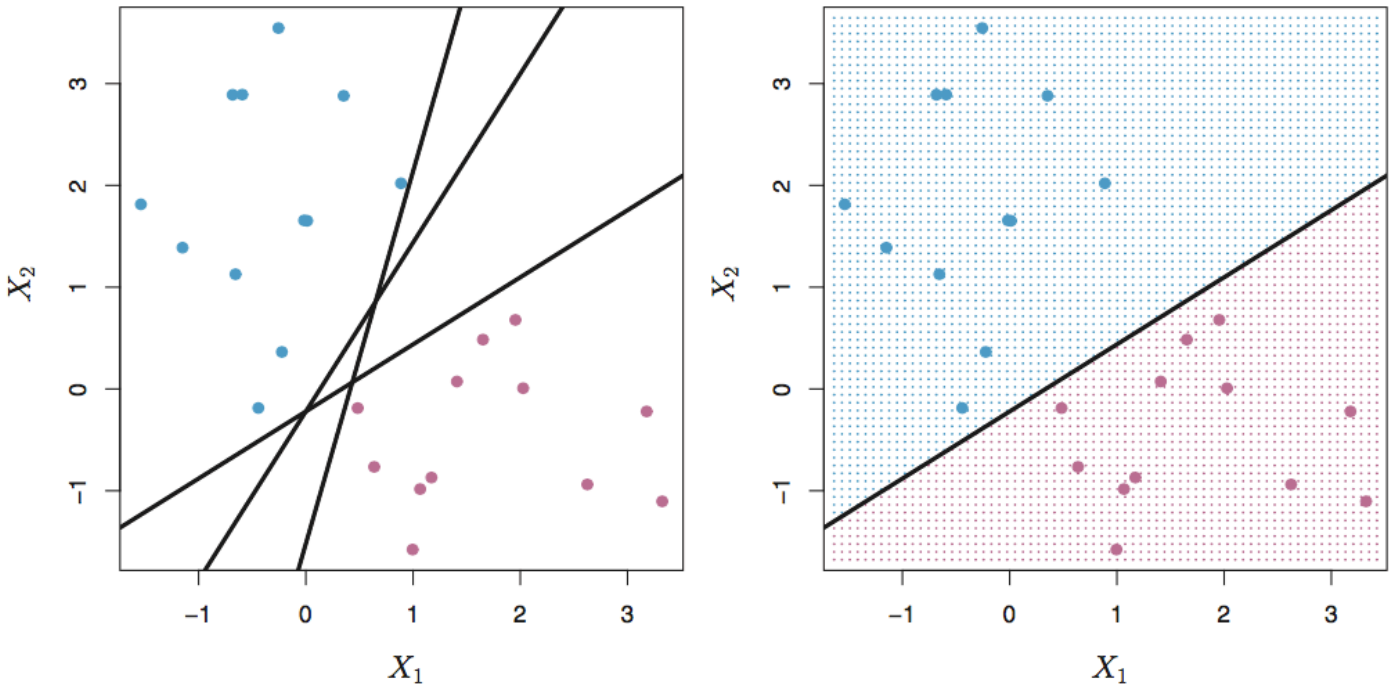


FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.*

- if separable $\rightarrow$ many separating hyperplanes
- maximal margin classifier: separating hyperplane is farthest from the training observation
- margin= minimal observation distance from the hyperplane

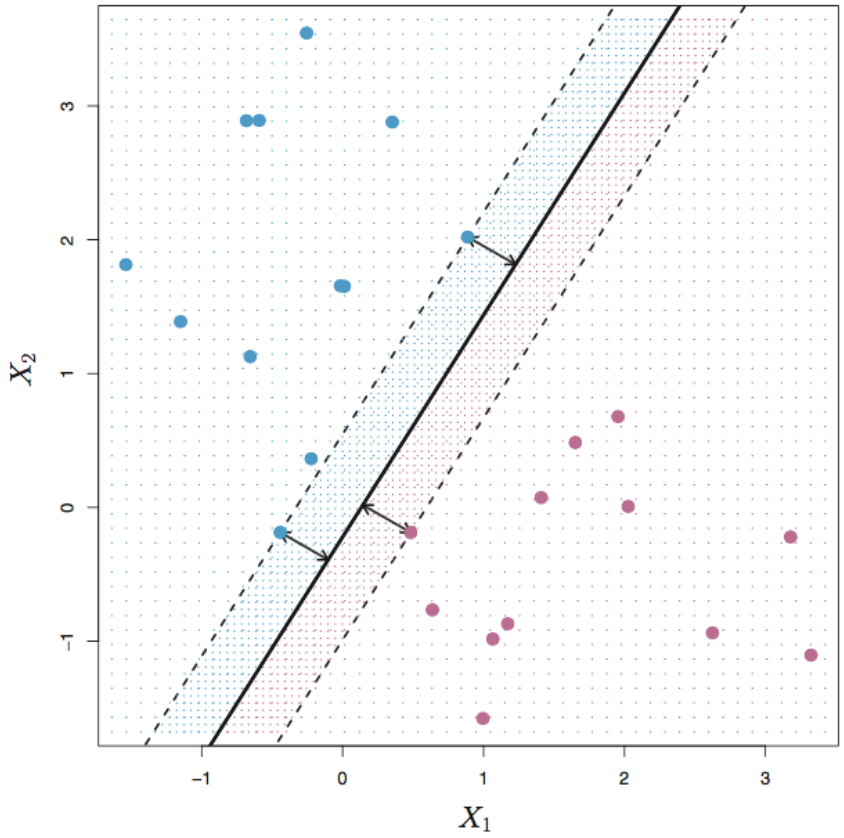


FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the margin is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.*

- support vectors= observation that support the maximal margin hyperplane
- maximal margin hyperplane $\sim$ support vectors $\sim$ small subset of the observations

$$
\begin{aligned}
\text{maximize}_{\beta_0,\beta} \quad & M \\
s.t.\ \|\beta\|_2^2 &= 1 \\
y_i\left(\beta_0 + \beta^\top x_i\right) &\geq M,\ \forall i = 1, \ldots, n
\end{aligned}
$$

- $y_i \left(\beta_0 + \beta^\top x_i\right) \geq 0$
  - observations are on the correct side

- $\|\beta\|_2 = 1$
  - perpendicular distance from hyperplane $= y_i \left(\beta_0 + \beta^\top x_i\right)$

- $y_i \left(\beta_0 + \beta^\top x_i\right) \geq M$
  - observations are at least M away from the border

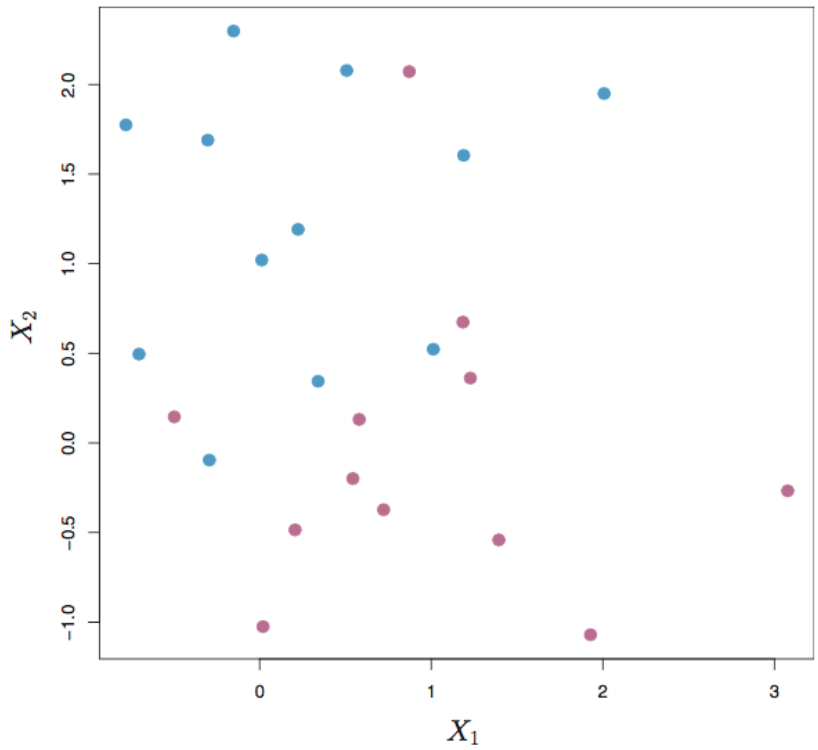


FIGURE 9.4. *There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.*

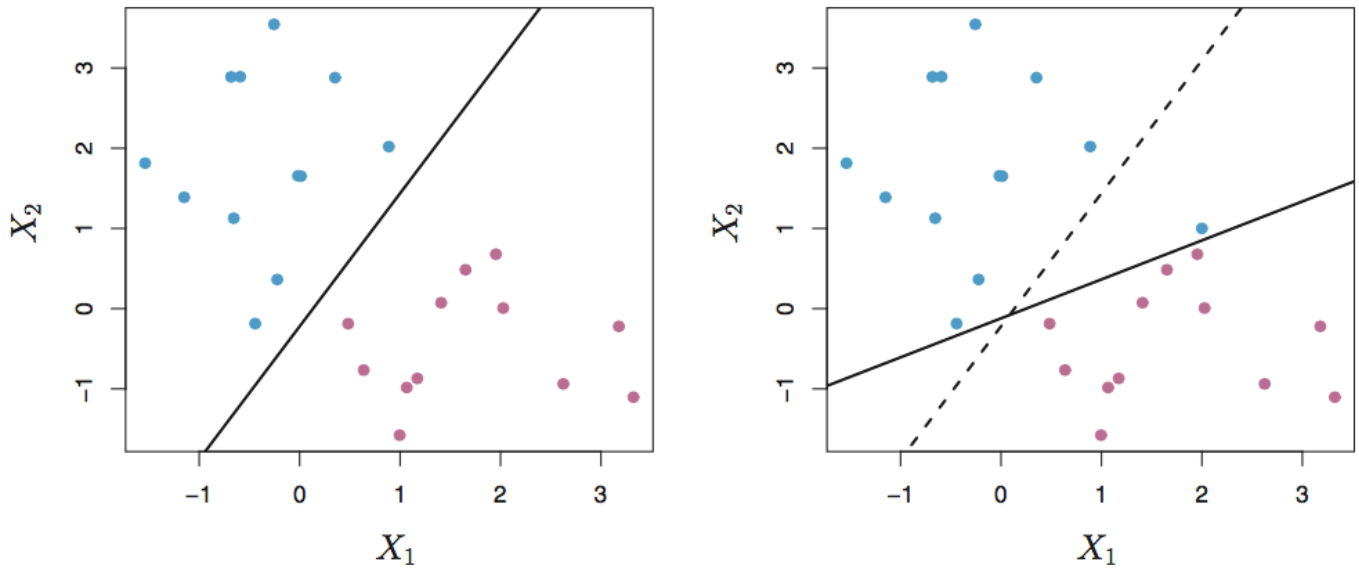


FIGURE 9.5. Left: *Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane.* Right: *An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.*

support vector classifier:

- more robust
- better classification of most observations

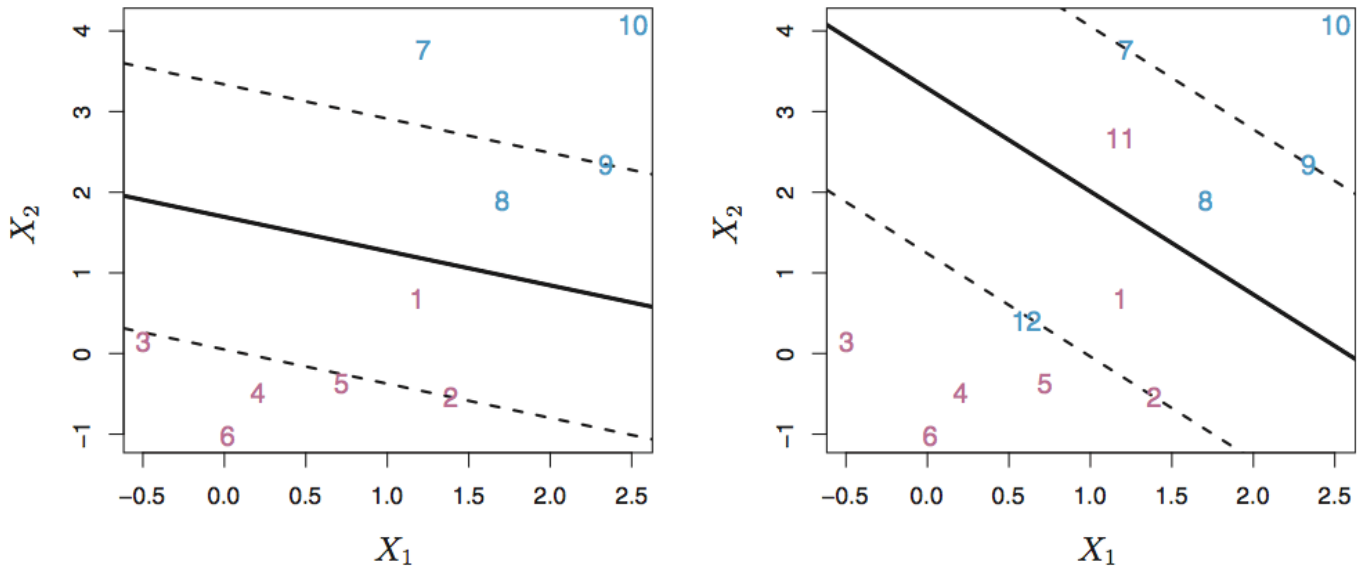


FIGURE 9.6. Left: *A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines.* Purple observations: *Observations* $3, 4, 5,$ *and* 6 *are on the correct side of the margin, observation* 2 *is on the margin, and observation 1 is on the wrong side of the margin.* Blue observations: *Observations* 7 *and* 10 *are on the correct side of the margin, observation* 9 *is on the margin, and observation* 8 *is on the wrong side of the margin. No observations are on the wrong side of the hyperplane.* Right: *Same as left panel with two additional points,* 11 *and* 12*. These two observations are on the wrong side of the hyperplane and the wrong side of the margin.*

$$
\begin{aligned}
\text{maximize}_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_n} \quad & M \\
s.t.\ \|\beta\|_2^2 &= 1 \\
y_i\left(\beta_0 + \beta^\top x_i\right) &\geq M(1-\epsilon_i),\ \forall i = 1,\ldots,n \\
\epsilon_i \geq 0, \quad & \sum_{i=1}^{n} \epsilon_i \leq C
\end{aligned}
$$

- C is a nonnegative tuning parameter
- M is the width of the margin
- $\epsilon_1, \ldots, \epsilon_n$ are slack variables

slack variables:

- $\epsilon_i = 0$: observation i is on the correct side of the margin

- if $\epsilon_i > 0$:
  1. $1 \geq \epsilon_i > 0$
     - observation i violated the margin
     - observation i is on the correct side of the hyperplane
  2. $\epsilon_i > 1$:
     - observation i violated the margin
     - observation i is on the wrong side of the hyperplane

tuning parameter C:

- $C \geq \sum_{i=1}^{n} \epsilon_i$: bounds the severity of violence
- $C = 0$: no violation = maximal margin hyperplane (if separable)

- observation i on the wrong side of the hyperplane: $\epsilon_i > 1$
- no more than C observations on the wrong side of the hyperplane
- chosen via cross-validation

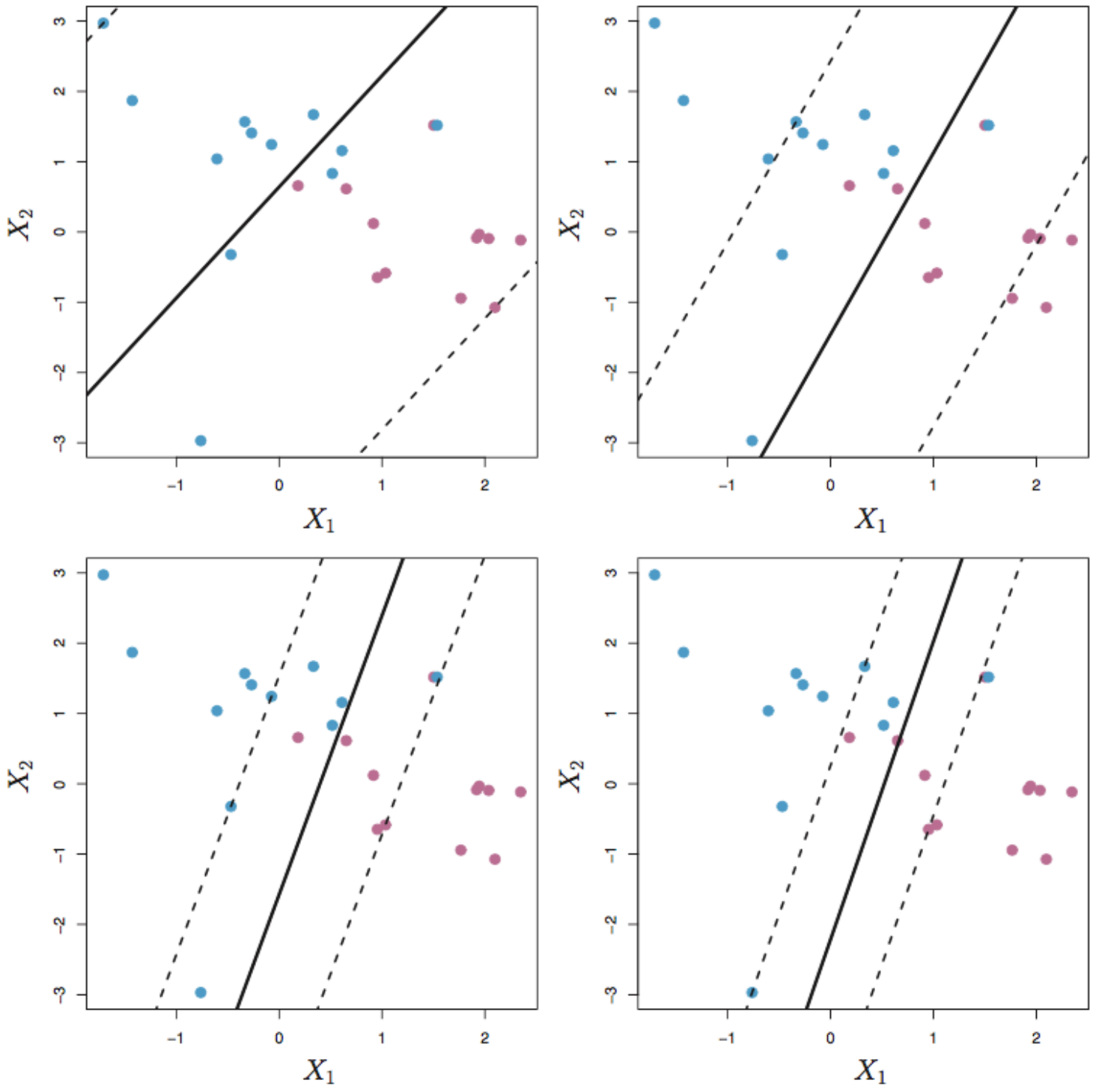
tuning parameter C:

- small C: classifier that highly fits the training data

- large C: classifier that fits the training data less hard

- robust: observations lying strictly on the correct side of the margin do not affect the support vector classifier
- LDA depends on all of the observations

- bias-variance trade-off ?


X_1
X_2
X_1
X_2
X_1
X_2
X_1
X_2

example of non-linear class boundaries:

$$
\begin{aligned}
\text{maximize}_{\beta_0,\beta_{11},\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n} \quad & M \\
s.t. \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 &= 1 \\
y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right) &\geq M(1-\epsilon_i),\ \forall i = 1,\ldots,n \\
\epsilon_i \geq 0, \qquad \sum_{i=1}^{n}\epsilon_i &\leq C
\end{aligned}
$$

- non-linear class boundaries: linear in the enlarged feature space
- higher order polynomial terms
- interaction terms

- huge number of features: computations become unmanageable

another look:

$$
\begin{aligned}
\text{maximize}_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_n} \quad & M \\
s.t.\ \|\beta\|_2^2 &= 1 \\
y_i\left(\beta_0 + \beta^\top x_i\right) &\geq M(1-\epsilon_i),\ \forall i = 1,\ldots,n \\
\epsilon_i \geq 0, \quad & \sum_{i=1}^{n} \epsilon_i \leq C
\end{aligned}
$$

more conveniently rephrased as:

$$\begin{aligned} \text{minimize}_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_n} \quad & \|\beta\|_2^2 \\ y_i\left(\beta_0 + \beta^\top x_i\right) \quad & \geq \quad 1 - \epsilon_i, \ \forall i = 1, \ldots, n \\ \epsilon_i \geq 0, \quad & \sum_{i=1}^{n} \epsilon_i \leq C \end{aligned}$$

- $\epsilon_i = 0$: observation i is on the correct side of the margin

- $1 \geq \epsilon_i > 0$: observation i violates the margin but is on the correct side of the hyperplane

- $\epsilon_i > 1$: observation i is on the wrong side of the hyperplane

- support vector machine = enlarged feature space + support vector classifier

- how to enlarge the feature space without adding computations?

- support vector classifier involves only inner products:

$$\langle x_i, x_{i'} \rangle = \sum_{i=1}^{p} x_{ij} x_{i'j} \tag{1}$$

- linear support vector classifier:

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle \tag{2}$$

- $\binom{n}{2}$ inner products $\langle x_i, x_{i'} \rangle$
- α_i is only non zero for the support vectors:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle \tag{3}$$

generalization of the inner product:

- $$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} = x^\top x'$$

- polynomial kernel of degree d

$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d = (1 + x^\top x')^d$$

- radial kernel

$$K(x_i, x_{i'}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2\right) = \exp\left(-\gamma \|x - x'\|_2^2\right)$$

support vector classifier = support vector machine with polynomial kernel $d = 1$:

$$K(x_i, x_{i'}) \quad = \quad (1 + x^\top x')^d$$

polynomial kernel $d = 2$: what is the implied feature space?

$$K(x_i, x_{i'}) \quad = \quad (1 + x^\top x')^d$$

radial kernel?

- test observation x^*: $f(x^*) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x^*, x_i)$.
- if $\|x^* - x_i\|_2^2$ is large, then x_i has little role in $f(x^*)$.
- radial kernel is local $\sim$ nearby training observations matter

why not just enlarge the feature space?

- kernels: compute only $\binom{n}{2}$ for all distinct pairs i, i'
- kernels: enlarge without explicitly enlarging the feature space
- example: radial kernels: the implicit feature space is infinite-dimensional

example: heart disease $\sim$ age + sex + chol + ten other predictors

- svm vs. lda
- remove 6 missing data
- randomly split:
  - 207 training
  - 90 test

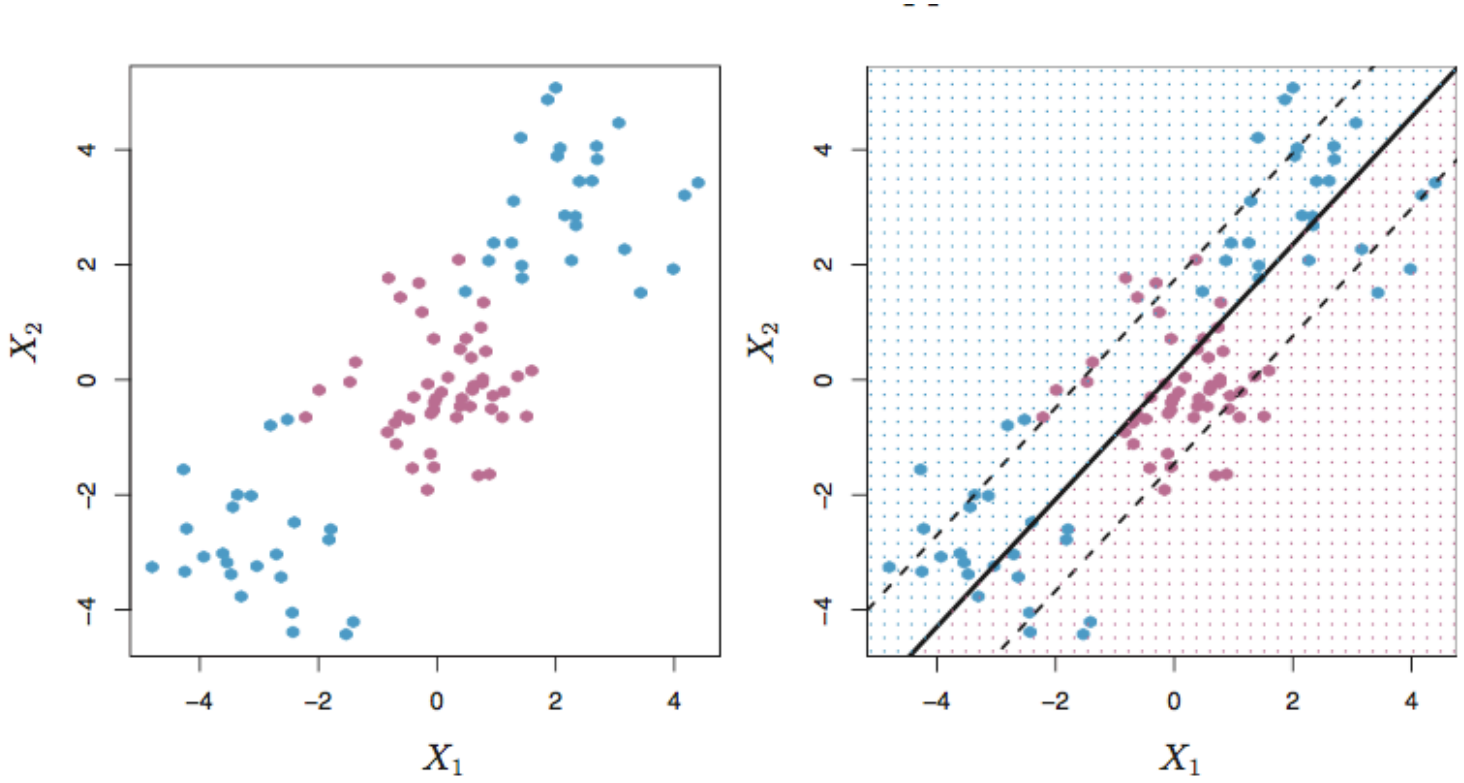


FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

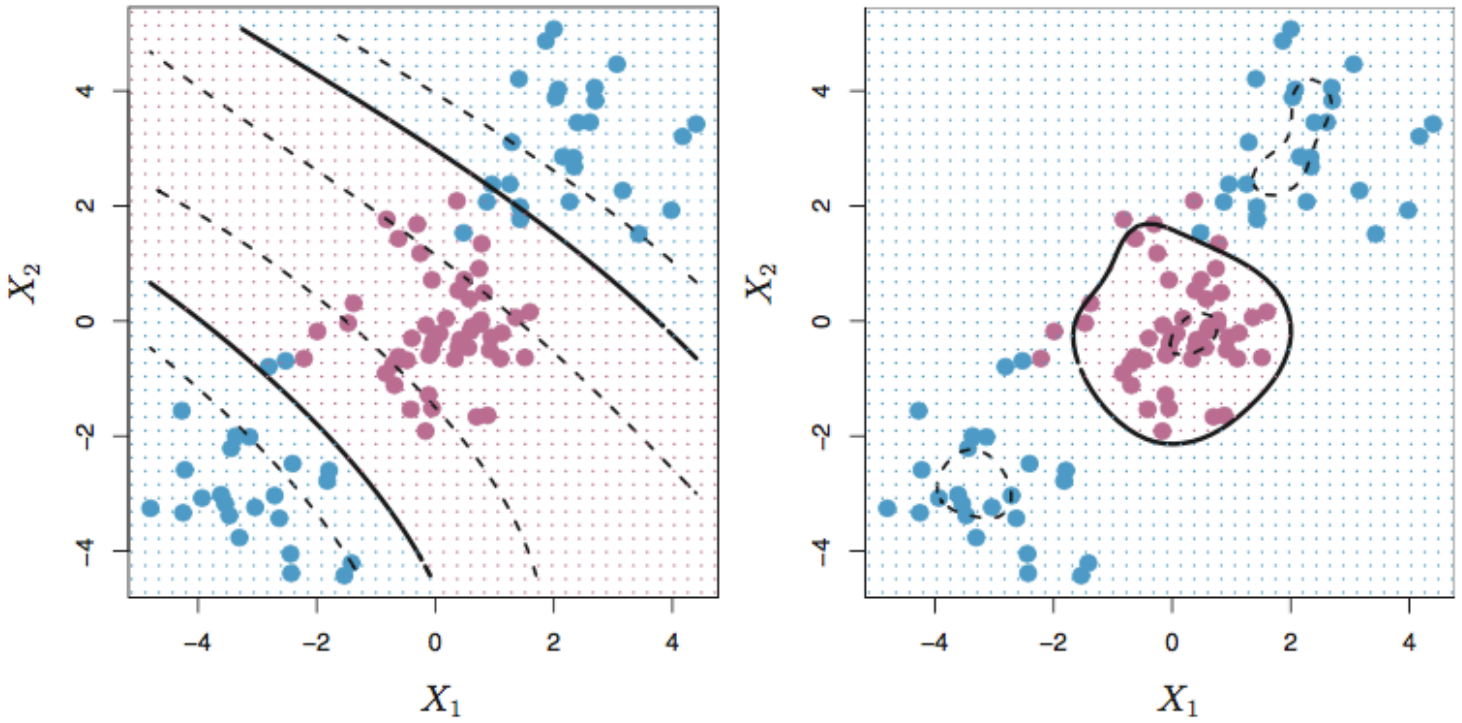


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

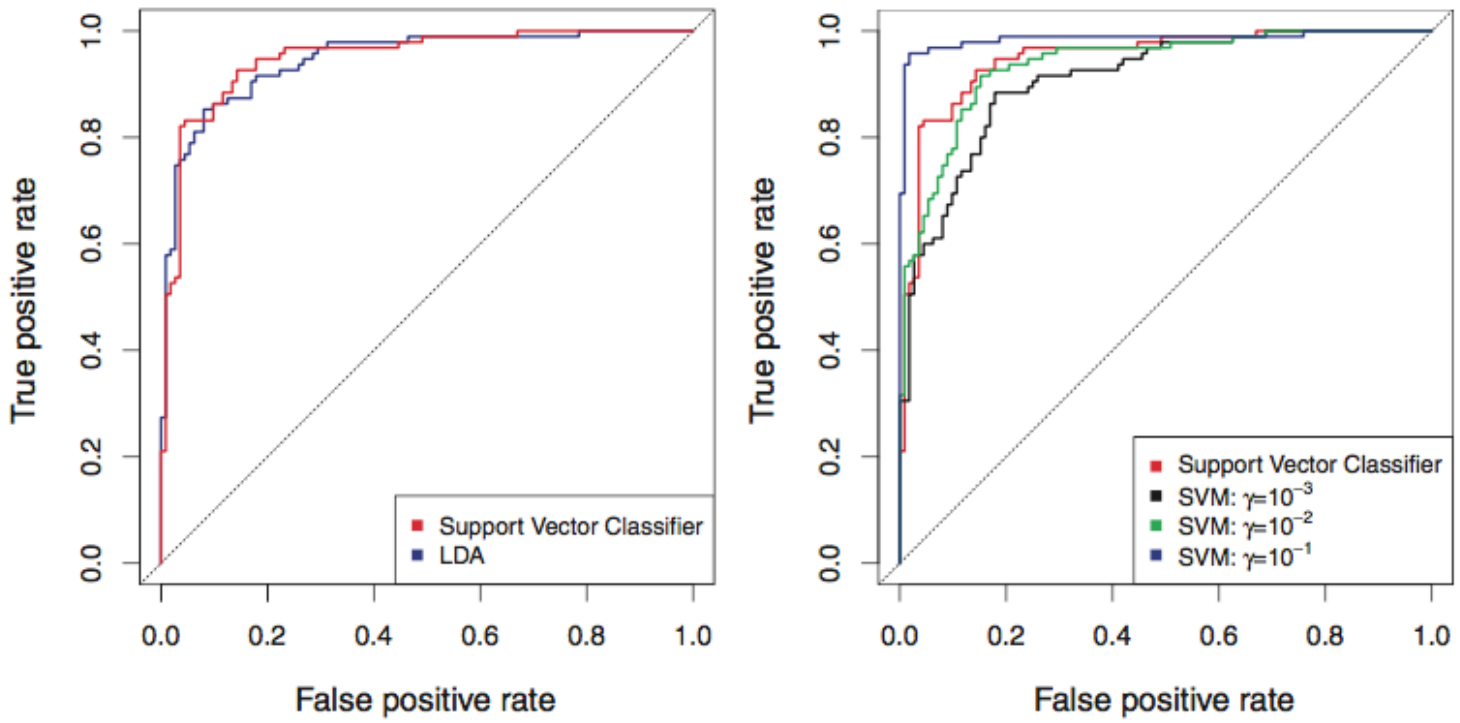


FIGURE 9.10. *ROC curves for the* `Heart` *data training set.* Left: *The support vector classifier and LDA are compared.* Right: *The support vector classifier is compared to an SVM using a radial basis kernel with* $\gamma = 10^{-3}$, 10^{-2}, *and* 10^{-1}.

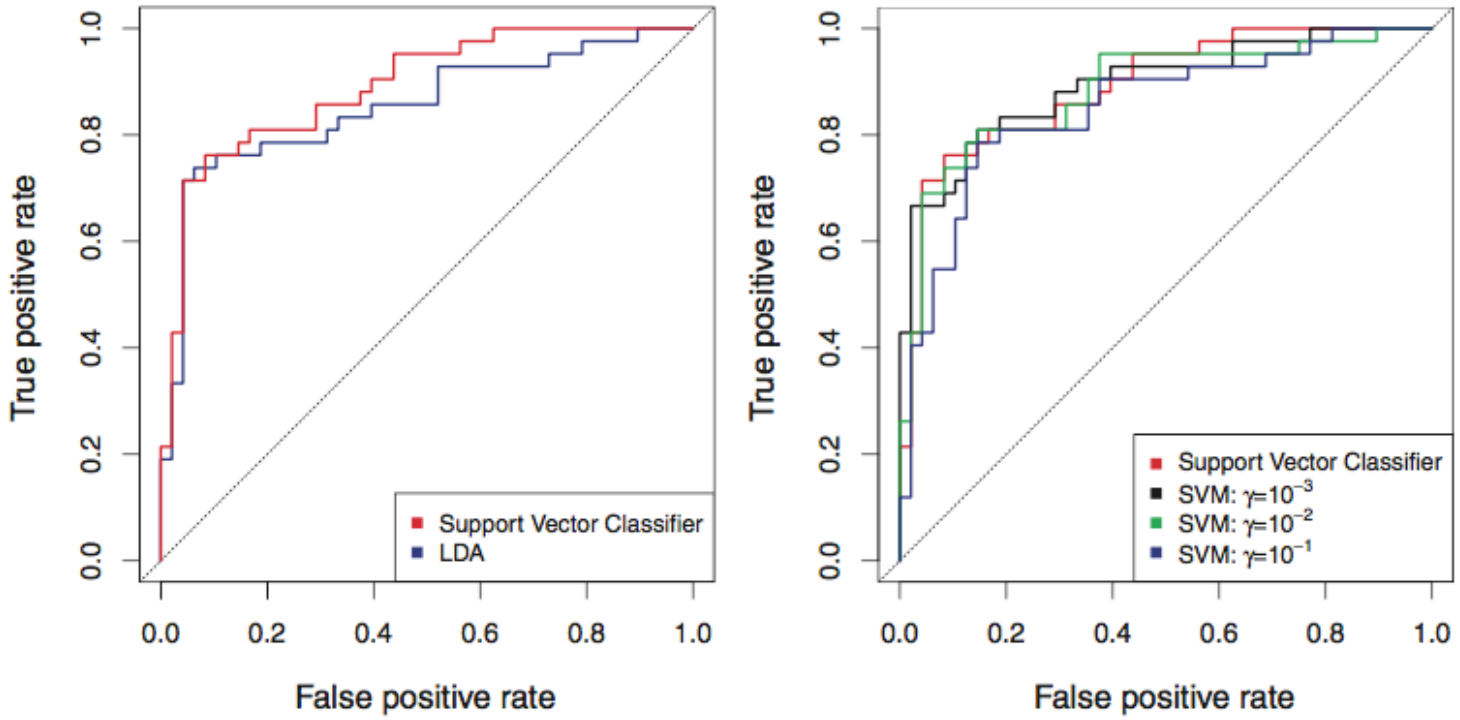


FIGURE 9.11. *ROC curves for the test set of the* `Heart` *data.* Left: *The support vector classifier and LDA are compared.* Right: *The support vector classifier is compared to an SVM using a radial basis kernel with* $\gamma = 10^{-3}$, 10^{-2}, *and* 10^{-1}.

more than two classes?

- one-versus-one classifier
- one-versus-all classifier

one-versus-one classifier:

- $\binom{K}{2}$ classifiers
- assign test to the class most frequently assigned

one-versus-all classifier:

- K classifiers: one versus $K - 1$
- assign test x^* to the class for which $f_k(x^*)$ is largest= high level of confidence

- find a hyperplane that data while allowing some violations: not new!
- expand feature space without added computational cost: new.

more conveniently rephrased as:

$$\begin{aligned} \text{minimize}_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_n} \quad & \|\beta\|_2^2 \\ y_i\left(\beta_0 + \beta^\top x_i\right) \quad & \geq \quad 1 - \epsilon_i, \ \forall i = 1, \ldots, n \\ \epsilon_i \geq 0, \quad & C \geq \sum_{i=1}^{n} \epsilon_i \end{aligned}$$

- $\epsilon_i = 0$: observation i is on the correct side of the margin
- $1 \geq \epsilon_i > 0$: observation i violates the margin but is on the correct side of the hyperplane
- $\epsilon_i > 1$: observation i is on the wrong side of the hyperplane

$$
\begin{aligned}
\text{minimize}_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_n} \quad & \|\beta\|_2^2 \\
y_i\left(\beta_0 + \beta^\top x_i\right) \quad & \geq \quad 1 - \epsilon_i,\ \forall i = 1,\ldots,n \\
\epsilon_i \geq 0, \quad & C \geq \sum_{i=1}^{n} \epsilon_i
\end{aligned}
$$

$$
\begin{aligned}
\text{minimize}_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_n} \quad & \|\beta\|_2^2 \\
\epsilon_i \quad & \geq \quad 1 - y_i \underbrace{\left(\beta_0 + \beta^\top x_i\right)}_{f(x_i)},\ \forall i = 1,\ldots,n \\
\epsilon_i \geq 0, \quad & C \geq \sum_{i=1}^{n} \epsilon_i
\end{aligned}
$$

svm:

$$\text{minimize}_{\beta_0,\beta} \quad \sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \|\beta\|_2^2$$

- λ large, β is small $\rightarrow$ more violations are tolerated + low-variance + high-bias
- λ small, β is large $\rightarrow$ less violations are tolerated + high-variance + low-bias

svm:

$$\text{minimize}_{\beta_0,\beta} \quad \underbrace{\sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)]}_{hinge\ loss} + \underbrace{\lambda \|\beta\|_2^2}_{penalty}$$

logistic ridge :

$$\text{minimize}_{\beta_0,\beta} \quad \underbrace{\sum_{i=1}^{n} \ln\left(1 + \exp(-y_i f(x_i))\right)}_{logistic\ loss} + \underbrace{\lambda \|\beta\|_2^2}_{penalty}$$

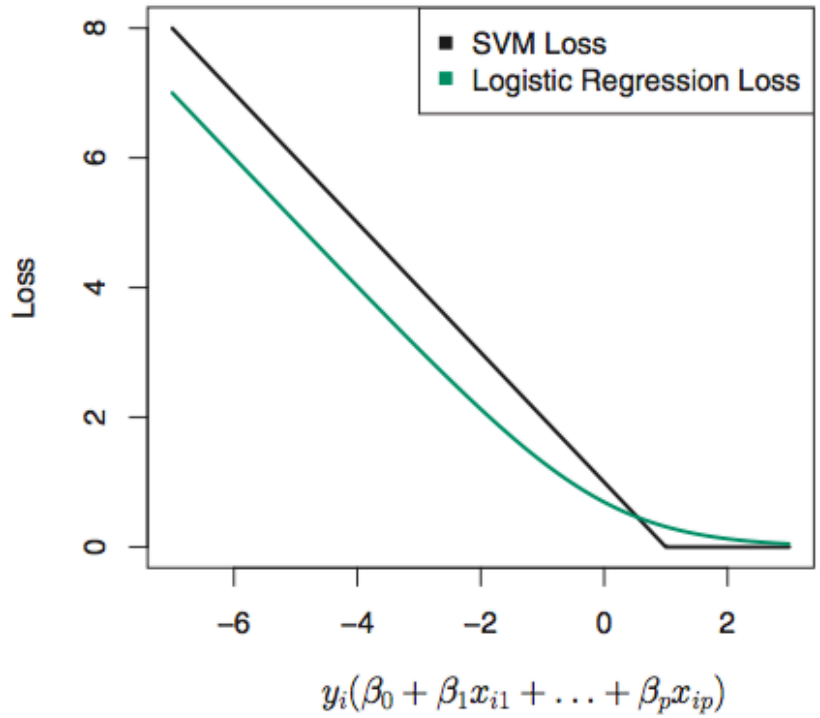


FIGURE 9.12. *The SVM and logistic regression loss functions are compared, as a function of* $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$. *When* $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})$ *is greater than 1, then the SVM loss is zero, since this corresponds to an observation that is on the correct side of the margin. Overall, the two loss functions have quite similar behavior.*