- Linear regression requires that we specify the model's form before beginning the modeling process. For example, in our previous discussion, prior to creating a model we had to determine which predictors to include in our model. We also had to decide whether we would include polynomial or log-transformed variables in our model and whether we would consider interaction effects.

CASE STUDY: PREDICTING BLOOD PRESSURE

Now that we have a better understanding of how to build, evaluate, and improve a linear regression model, let's put some of the principles we learned in the previous sections to use. Suppose you are freelancing as a data science consultant with a small community clinic in Chicago. The care providers at the clinic are concerned about the prevalence of hypertension among their patient population. If left untreated for a sustained period of time, high blood pressure can lead to significant medical complications such as heart attack, stroke, or kidney disease. To raise awareness of the issue, the clinic would like you to develop a model that predicts blood pressure, based on anonymized health metrics and limited lifestyle information about their patients. The clinic's goal is to use this model to develop an interactive self-service patient portal that provides a patient's estimated blood pressure based on their health metrics and lifestyle.

You are provided with data for 1,475 patients collected by the clinic over the last 12 months. The data that you will be using in this case study is real-world data collected by the U.S. Centers for Disease Control and Prevention as part of its National Health and Nutrition Examination Survey (NHANES). Extensive data from this survey is available through the `RNHANES` package. The variables in our dataset are as follows:

- *`systolic`* is the systolic blood pressure of the patient. The unit of measure is millimeters of mercury (mmHg). This is the dependent variable that we want to predict.
- *`weight`* is the measured weight of the patient in kilograms (kg).
- *`height`* is the measured height of the patient in centimeters (cm).
- *`bmi`* is the body mass index of the patient. This provides a sense of how underweight or overweight a patient is.
- *`waist`* is the measured circumference of a patient's waist in centimeters (cm).
- *`age`* is the self-reported age of the patient.
- *`diabetes`* is a binary indictor of whether the patient has diabetes (1) or not (0).
- *`smoker`* is a binary indicator of whether the patient smokes cigarettes regularly (1) or not (0).
- *`fastfood`* is a self-reported count of how many fast-food meals the patient has had in the past week.

Importing the Data

We begin by reading our data using the `read_csv()` function from the *tidyverse* package.

```
> library(tidyverse)

> health <- read_csv("health.csv")
```

We successfully imported the 1,475 observations and 9 variables. To get a quick view of our data, we use the `glimpse()` command to show us our variable names, data types, and some sample data.

```
> glimpse(health)

Observations: 1,475
Variables: 9
$ systolic <dbl> 100, 112, 134, 108, 128, 102, 126, 124, 166, 138, 118, 124, 96, 116,...
$ weight   <dbl> 98.6, 96.9, 108.2, 84.8, 97.0, 102.4, 99.4, 53.6, 78.6, 135.5, 72.3,...
$ height   <dbl> 172.0, 186.0, 154.4, 168.9, 175.3, 150.5, 157.8, 162.4, 156.9, 180.2...
$ bmi      <dbl> 33.3, 28.0, 45.4, 29.7, 31.6, 45.2, 39.9, 20.3, 31.9, 41.7, 28.6, 31...
$ waist    <dbl> 120.4, 107.8, 120.3, 109.0, 111.1, 130.7, 113.2, 74.6, 102.8, 138.4,...
$ age      <dbl> 43, 57, 38, 75, 42, 63, 58, 26, 51, 61, 47, 52, 64, 55, 72, 80, 71, ...
$ diabetes <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,...
$ smoker   <dbl> 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0,...
$ fastfood <dbl> 5, 0, 2, 1, 1, 3, 6, 5, 0, 1, 0, 3, 0, 1, 0, 5, 0, 2, 1, 3, 2, 0, 12...
```

As we discussed earlier, systolic will be the response variable, and the other variables will be our predictors. Notice that all the variables were imported as numeric (*dbl* to be precise). However, we do know that the diabetes and smoker variables are actually categorical values. So, we need to convert these variables to factors by using the `as.factor()` function.

```
> health <- health %>%
  mutate(diabetes=as.factor(diabetes)) %>%
  mutate(smoker=as.factor(smoker))
```

Exploring the Data

Now that we have our data, let's explore our data. We start by using the `summary()` function to get a statistical summary of the numeric variables in our data.

```
> summary(health)

    systolic         weight           height           bmi             waist      
 Min.   : 80.0   Min.   : 29.10   Min.   :141.2   Min.   :13.40   Min.   : 56.2  
 1st Qu.:114.0   1st Qu.: 69.15   1st Qu.:163.8   1st Qu.:24.10   1st Qu.: 88.4  
 Median :122.0   Median : 81.00   Median :170.3   Median :27.90   Median : 98.9  
 Mean   :124.7   Mean   : 83.56   Mean   :170.2   Mean   :28.79   Mean   :100.0  
 3rd Qu.:134.0   3rd Qu.: 94.50   3rd Qu.:176.8   3rd Qu.:32.10   3rd Qu.:109.5  
 Max.   :224.0   Max.   :203.50   Max.   :200.4   Max.   :62.00   Max.   :176.0  

      age          diabetes smoker      fastfood    
 Min.   :20.00   0:1265   0:770   Min.   : 0.00  
 1st Qu.:34.00   1: 210   1:705   1st Qu.: 0.00  
 Median :49.00                    Median : 1.00  
 Mean   :48.89                    Mean   : 2.14  
 3rd Qu.:62.00                    3rd Qu.: 3.00  
 Max.   :80.00                    Max.   :22.00  
```

Looking at the statistical distribution for our response variable *`systolic`*, we see that the mean and median are relatively close, suggesting that the data is normally distributed. Using a histogram, we can get a visual representation of the distribution (Figure 4.8).

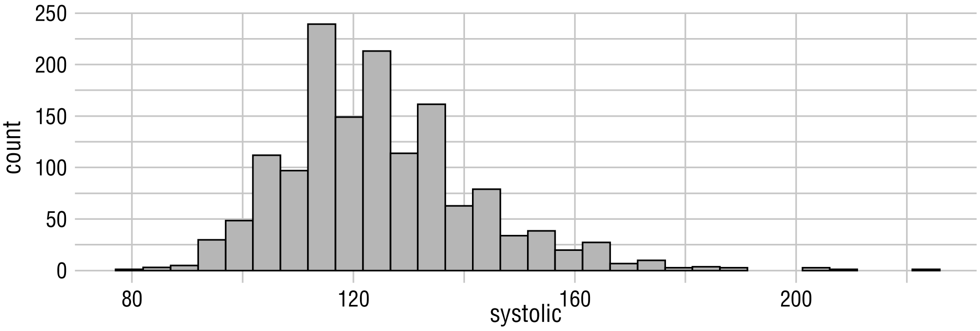

Figure 4.8 The systolic blood pressure data for this population appears to be normally distributed.

```
> health %>%
   ggplot() +
      geom_histogram(mapping=aes(x=systolic), fill = "lightblue", color =
"black") +
      theme_minimal()
```

The histogram shows that the data for the response variable is normally distributed. Now, let's also take a look at the statistical distributions of the predictor variables using a set of histograms. We do this by using the *tidyverse* keep(), gather(), and facet_wrap() functions (Figure 4.9).

```
> health %>%
   select(-systolic) %>%
   keep(is.numeric) %>%
   gather() %>%
   ggplot() +
       geom_histogram(mapping = aes(x=value,fill=key), color = "black") +
        facet_wrap(~ key, scales = "free") +
theme_minimal()
```

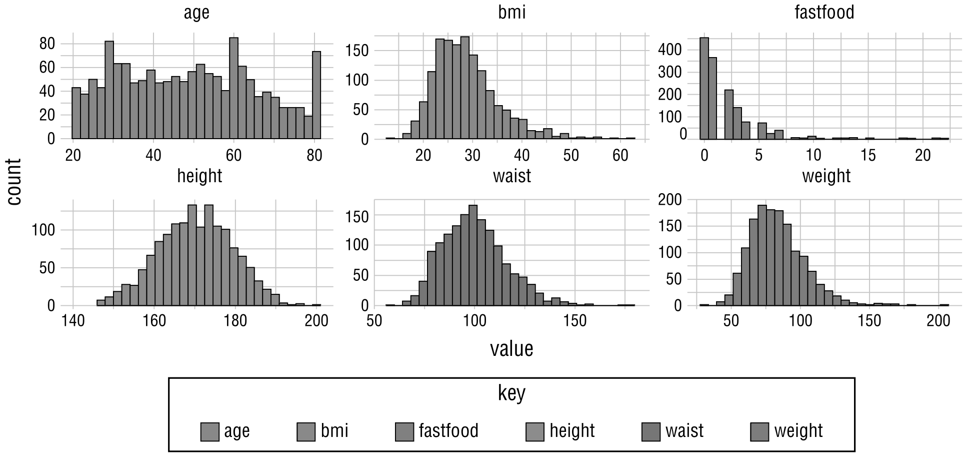

Figure 4.9 Distributions of dependent variables in the health dataset

We see a near uniform distribution for our *age* predictor. This means that our data is representative of patients across a wide age spectrum. This is to be expected. The *fastfood* variable is right-skewed. Most of our patients consume fast food as a meal less than five times a week. The rest of our predictors are normally distributed. From visual inspection, there are no obvious outliers in our data that need to be dealt with.

The next thing we need to do as part of the data exploration process is to look at the correlation between our continuous variables. To do this, we use the `cor()` function, which was introduced earlier.

```
> cor(health[, c("systolic","weight","height","bmi","waist","age","fastfood")])

            systolic      weight      height         bmi       waist         age    fastfood
systolic  1.00000000  0.10021386  0.02301030  0.09054668  0.16813021  0.40170911 -0.08417538
weight    0.10021386  1.00000000  0.40622019  0.89152826  0.89928820 -0.02217221  0.05770725
height    0.02301030  0.40622019  1.00000000 -0.03848241  0.14544676 -0.12656952  0.10917107
bmi       0.09054668  0.89152826 -0.03848241  1.00000000  0.91253710  0.03379844  0.01003525
waist     0.16813021  0.89928820  0.14544676  0.91253710  1.00000000  0.19508769 -0.02167324
age       0.40170911 -0.02217221 -0.12656952  0.03379844  0.19508769  1.00000000 -0.30089756
fastfood -0.08417538  0.05770725  0.10917107  0.01003525 -0.02167324 -0.30089756  1.00000000
```

Looking at the *systolic* column, we can see that the *age* predictor has the strongest correlation with systolic blood pressure. This is followed by *waist* size and *weight*, both of which are weakly correlated. It is interesting to note the negative correlation between *fastfood* consumption and *systolic* blood pressure. This seems unusual and counter-intuitive; however, the negative correlation is quite low, so it will not significantly impact our model.

Fitting the Simple Linear Regression Model

In the previous two sections, we imported and explored our data. From our exploration, we discovered that the *age* predictor has the strongest correlation to our response. So, we will begin by building a simple linear regression model using the *age* as the predictor and *systolic* as the response.

```
> health_mod1 <- lm(data=health, systolic~age)

> summary(health_mod1)

Call:
lm(formula = systolic ~ age, data = health)

Residuals:
    Min      1Q  Median      3Q     Max
-42.028 -10.109  -1.101   8.223  98.806
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 104.34474    1.28169   81.41   <2e-16 ***
age           0.41698    0.02477   16.84   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.14 on 1473 degrees of freedom
Multiple R-squared:  0.1614,  Adjusted R-squared:  0.1608
F-statistic: 283.4 on 1 and 1473 DF,  p-value: < 2.2e-16
```

Our results show that our predictors are significant. The coefficient for age tells us that for every 0.4-year increase in a patient's age, we should expect his or her systolic blood pressure to increase by 1 point. This means that, on average, the older a patient is, the higher their blood pressure.

Looking at our model diagnostics, we see that our residual standard error is low and our F-statistic is statistically significant. These are both good indicators of model fit. However, our multiple R-squared tells us that our model explains only 16 percent of the variability in the response. Let's see if we can do better by introducing additional predictors to the model.

Fitting the Multiple Linear Regression Model

For our multiple linear regression model, we will begin with all the predictors in our data and `systolic` as the response.

```
> health_mod2 <- lm(data=health, systolic~.)

> summary(health_mod2)

Call:
lm(formula = systolic ~ ., data = health)

Residuals:
    Min      1Q  Median      3Q     Max
-41.463 -10.105  -0.765   8.148 100.398

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 163.30026   33.52545   4.871 1.23e-06 ***
weight        0.55135    0.19835   2.780  0.00551 **
height       -0.39201    0.19553  -2.005  0.04516 *
bmi          -1.36839    0.57574  -2.377  0.01759 *
```

```
waist        -0.00955    0.08358  -0.114  0.90905
age           0.43345    0.03199  13.549  < 2e-16 ***
diabetes1     2.20636    1.26536   1.744  0.08143 .
smoker1       1.13983    0.90964   1.253  0.21039
fastfood      0.17638    0.15322   1.151  0.24985
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.99 on 1466 degrees of freedom
Multiple R-squared:  0.1808,       Adjusted R-squared:  0.1763
F-statistic: 40.44 on 8 and 1466 DF,  p-value: < 2.2e-16
```

The results show that the coefficient estimates for *weight*, *height*, *bmi*, *age*, and *diabetes* are significant in the model. Our model diagnostics also show a slight reduction in our residual standard error, a slight increase in our adjusted R-squared and significant F-statistic that is greater than 0. Overall, this model provides a better fit than our previous model. Let's now run some additional diagnostic tests against our new model.

The first test we run is the test for zero mean of residuals.

```
> mean(health_mod2$residuals)

[1] -1.121831e-15
```

Our residual mean is very close to zero, so our model passes this test.

Next, we test for normality of residuals (Figure 4.10).

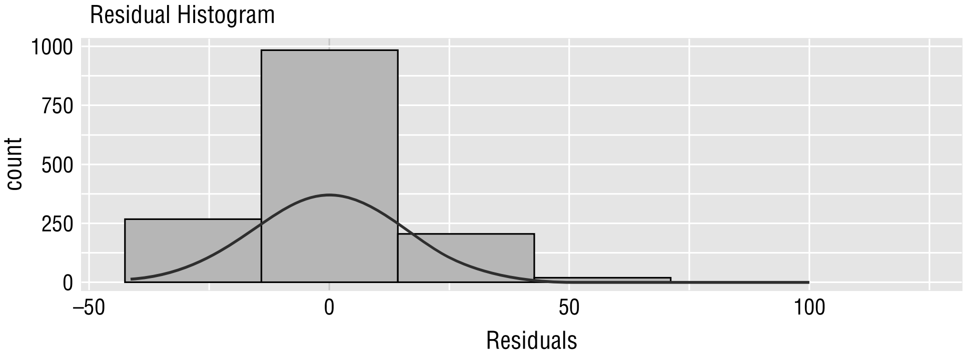

Figure 4.10 Histogram of residuals produced using the `ols_plot_resid_hist()` function

```
> library(olsrr)

> ols_plot_resid_hist(health_mod2)
```

The residual plot is normally distributed with a slight right skew. This is close enough to a normal distribution to satisfy our test.

Next, we test for the presence of heteroscedasticity in our residuals (Figure 4.11).

```
> ols_plot_resid_fit(health_mod2)
```

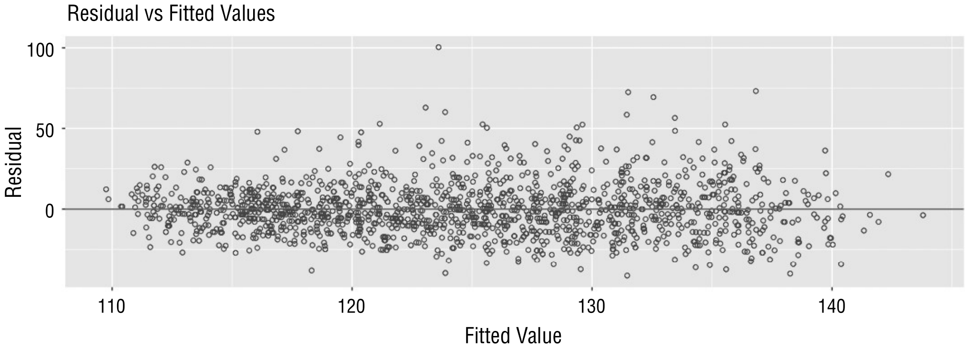

Figure 4.11 Scatterplot of residuals produced using the `ols_plot_resid_fit()` function

Our plot shows an even distribution of points around the origin line. There is no heteroscedasticity in the distribution of our residuals versus fitted values.

Next, we run a test for residual autocorrelation.

```
> library(car)

> durbinWatsonTest(health_mod2)

 lag Autocorrelation D-W Statistic p-value
   1     -0.01985291      2.038055   0.456
 Alternative hypothesis: rho != 0
```

With a Durbin-Watson statistic of 2.04 and a p-value greater than 0.05, we cannot reject the null hypothesis that "no first order autocorrelation exists." Therefore, we can say that our residuals are not autocorrelated.

The next diagnostic test we run is a check for influential points in our data by generating a chart of Cook's distance function for our dataset (Figure 4.12).

```
> ols_plot_cooksd_chart(health_mod2)
```

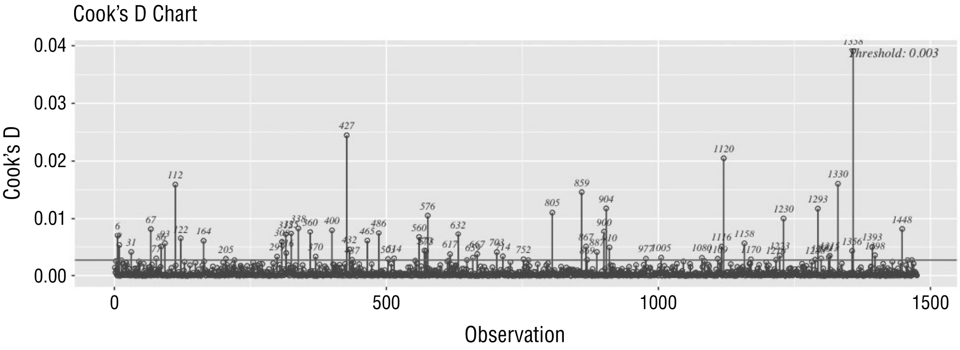

Figure 4.12 Cook's distance chart for the health dataset produced using the `ols_plot_cooksd_chart()` function

Our plot shows that there are indeed several influential points in our data. Observation 1358 stands out from the rest. Let's take a look at the observed values for that observation:

```
> health[1358,]

# A tibble: 1 x 9
  systolic weight height   bmi waist   age diabetes smoker fastfood
     <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <fct>    <fct>     <dbl>
1      184   146.   180.  44.9  140.    26 0         0           14
```

and compare those values to the statistical summary of our entire dataset, shown here:

```
> summary(health)

    systolic         weight           height           bmi             waist
 Min.   : 80.0   Min.   : 29.10   Min.   :141.2   Min.   :13.40   Min.   : 56.2
 1st Qu.:114.0   1st Qu.: 69.15   1st Qu.:163.8   1st Qu.:24.10   1st Qu.: 88.4
 Median :122.0   Median : 81.00   Median :170.3   Median :27.90   Median : 98.9
 Mean   :124.7   Mean   : 83.56   Mean   :170.2   Mean   :28.79   Mean   :100.0
 3rd Qu.:134.0   3rd Qu.: 94.50   3rd Qu.:176.8   3rd Qu.:32.10   3rd Qu.:109.5
 Max.   :224.0   Max.   :203.50   Max.   :200.4   Max.   :62.00   Max.   :176.0

      age         diabetes smoker     fastfood
 Min.   :20.00   0:1265   0:770   Min.   : 0.00
 1st Qu.:34.00   1: 210   1:705   1st Qu.: 0.00
 Median :49.00                    Median : 1.00
 Mean   :48.89                    Mean   : 2.14
 3rd Qu.:62.00                    3rd Qu.: 3.00
 Max.   :80.00                    Max.   :22.00
```

We can see that the values for *weight*, *bmi*, *height*, *age*, and *fastfood* are significantly different for observation 1358 compared to the average and median of those variables across the entire dataset.

Let's also take a look at the statistical distribution of the rest of the outliers and compare those to the statistical distribution of the data without the outliers. To do so, we will need a list of all the observations that make up our influential points. We first need to get a list of the index values for those observations. This is done by referring to the *observation* column of the *outlier* attribute from Cook's distance function.

```
> outlier_index <-
as.numeric(unlist(ols_plot_cooksd_chart(health_mod2)$outliers[,"observation"]))

> outlier_index

 [1]    6    9   31   67   77   86   93  112  122  164  205  299  308  315  316  325
[17]  338  360  370  400  427  432  437  465  486  503  514  560  570  573  576  617
[33]  632  659  667  703  714  752  805  859  867  869  887  900  904  910  977 1005
[49] 1080 1109 1116 1120 1158 1170 1216 1223 1230 1288 1293 1299 1313 1315 1330 1356
[65] 1358 1393 1398 1448
```

There are 68 observations in the list. Now that we have the outlier index values, we use the `summary()` command to compare the two datasets. First, let's look at a statistical summary of only the outlier points:

```
> summary(health[outlier_index,])

    systolic         weight           height           bmi            waist
 Min.   : 86.0   Min.   : 29.10   Min.   :144.2   Min.   :13.40   Min.   : 56.20
 1st Qu.:109.0   1st Qu.: 68.92   1st Qu.:159.5   1st Qu.:23.60   1st Qu.: 92.35
 Median :163.0   Median : 82.20   Median :167.2   Median :32.00   Median :111.20
 Mean   :149.4   Mean   : 91.73   Mean   :167.2   Mean   :32.26   Mean   :109.81
 3rd Qu.:174.0   3rd Qu.:109.03   3rd Qu.:174.2   3rd Qu.:38.42   3rd Qu.:124.92
 Max.   :224.0   Max.   :203.50   Max.   :193.3   Max.   :62.00   Max.   :172.20

      age         diabetes smoker    fastfood
 Min.   :21.00   0:44     0:29   Min.   : 0.000
 1st Qu.:41.75   1:24     1:39   1st Qu.: 0.000
 Median :56.00                   Median : 1.000
 Mean   :55.50                   Mean   : 2.897
 3rd Qu.:68.00                   3rd Qu.: 3.000
 Max.   :80.00                   Max.   :18.000
```

Next, let's compare that to a summary of the points in the dataset excluding the outliers.

```
> summary(health[-outlier_index,])

    systolic         weight           height           bmi             waist       
 Min.   : 80.0   Min.   : 41.10   Min.   :141.2   Min.   :16.00   Min.   : 65.60  
 1st Qu.:114.0   1st Qu.: 69.15   1st Qu.:164.0   1st Qu.:24.10   1st Qu.: 88.15  
 Median :122.0   Median : 81.00   Median :170.4   Median :27.80   Median : 98.50  
 Mean   :123.5   Mean   : 83.17   Mean   :170.3   Mean   :28.63   Mean   : 99.56  
 3rd Qu.:134.0   3rd Qu.: 94.10   3rd Qu.:176.8   3rd Qu.:31.90   3rd Qu.:108.80  
 Max.   :182.0   Max.   :180.20   Max.   :200.4   Max.   :59.00   Max.   :176.00  

      age         diabetes smoker     fastfood     
 Min.   :20.00   0:1221   0:741   Min.   : 0.000  
 1st Qu.:34.00   1: 186   1:666   1st Qu.: 0.000  
 Median :48.00                    Median : 1.000  
 Mean   :48.57                    Mean   : 2.103  
 3rd Qu.:62.00                    3rd Qu.: 3.000  
 Max.   :80.00                    Max.   :22.000  
```

We can see a slight to moderate difference in the mean and median between each of the variable pairs. While the minimum and maximum values for most pairs are similar, we see a significant difference with the minimum and maximum values of the weight variable. To improve our model, we should remove these influential points from our dataset. However, for us to be able to refer to the original data, let's create a new version of our dataset from the original without outliers. We call this new dataset *health2*.

```
> health2 <- health[-outlier_index,]
```

The final diagnostic test that we run is the test for multicollinearity.

```
> ols_vif_tol(health_mod2)

# A tibble: 8 x 3
  Variables Tolerance   VIF
  <chr>         <dbl> <dbl>
1 weight       0.0104 96.1 
2 height       0.0522 19.2 
3 bmi          0.0125 80.0 
4 waist        0.0952 10.5 
5 age          0.588   1.70
6 diabetes1    0.887   1.13
7 smoker1      0.840   1.19
8 fastfood     0.896   1.12
```

With a VIF well above 5.0 for *weight*, *height*, *bmi*, and *waist*, it's obvious that we have a problem with multicollinearity. This is not surprising, considering that *bmi* is calculated as *weight* divided by the square of *height* and that waist size is highly correlated with a person's weight. To resolve our multicollinearity problem, we need to either combine the impacted variables or drop some of them. Since *weight* has the lowest tolerance among the four predictors, we choose to drop the other three and keep *weight*.

With the changes we've made to our data and the new insight we have about our model, let's build a new multiple linear regression model.

```
> health_mod3 <- lm(data=health2, systolic ~ weight+age+diabetes)

> summary(health_mod3)

Call:
lm(formula = systolic ~ weight + age + diabetes, data = health2)

Residuals:
    Min      1Q  Median      3Q     Max
-38.825  -9.004  -0.177   8.222  49.679

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 96.62591    1.93014  50.062  < 2e-16 ***
weight       0.09535    0.01870   5.100 3.87e-07 ***
age          0.38372    0.02218  17.297  < 2e-16 ***
diabetes1    2.62446    1.11859   2.346   0.0191 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.59 on 1403 degrees of freedom
Multiple R-squared:  0.2128,  Adjusted R-squared:  0.2111
F-statistic: 126.4 on 3 and 1403 DF,  p-value: < 2.2e-16
```

All our predictors are significant, and all our model diagnostics show an improvement over the previous model. Our model now explains 21 percent of the variability in the response. This is still rather low, so let's try to see whether we can further improve our model.

The next two things we consider are the possibility of an interaction effect between our predictors and the possibility that there is a nonlinear relationship between some of our predictors and the response.

It is reasonable to expect that there may be interactions between *weight* and *diabetes* and between *age* and *diabetes*, so we will incorporate those possible interactions into our model. We learned how to specify this earlier using the `*` operator.

It is also reasonable to expect that the relationship between *age* and hypertension may not be constant at all age levels. As a patient gets older, there very well may be an accelerated relationship between *age* and *systolic* blood pressure. To account for this possibility, we will need to introduce nonlinear predictors into our model. To do so, we add two new variables to our *health2* data — *age*^{2}, which we call *age2*, and `log(`*age*`)`, which we call *lage*.

```
> health2 <- health2 %>%
  mutate(age2=age^2,
          lage=log(age))
```

To build our next model, we again use the `ols_step_both_p()` function from the *olsrr* package to perform variable selection. We provide as input our original dataset, along with four interaction effects between diabetes and four other dependent variables: *weight*, *age*, *age2*, and *lage*.

```
> ols_step_both_p(
   model = lm(
     data = health2,
     systolic ~ weight * diabetes + age * diabetes + age2 * diabetes
     + lage * diabetes
   ),
   pent = 0.2,
   prem = 0.01,
   details = FALSE
 )

Final Model Output
------------------

                              Model Summary
--------------------------------------------------------------------------
R                            0.467         RMSE                     13.551
R-Squared                    0.218         Coef. Var                10.969
Adj. R-Squared               0.216         MSE                     183.636
Pred R-Squared               0.213         MAE                      10.626
--------------------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error
```

```
                                 ANOVA
-------------------------------------------------------------------------------
                  Sum of
                 Squares          DF    Mean Square      F          Sig.
-------------------------------------------------------------------------------
Regression      71747.979          4      17936.995     97.677     0.0000
Residual       257457.582       1402        183.636
Total          329205.561       1406
-------------------------------------------------------------------------------

                             Parameter Estimates
-------------------------------------------------------------------------------
 model             Beta     Std. Error   Std. Beta   t      Sig     lower      upper
--------------------------------------------------------------------------------
(Intercept)       142.588      14.796                 9.637   0.000   113.563    171.612
lage              -16.720       5.364      -0.411    -3.117   0.002   -27.243     -6.197
age                 0.750       0.119       0.830     6.295   0.000     0.516      0.983
weight:diabetes0    0.096       0.019       0.209     5.077   0.000     0.059      0.134
weight:diabetes1    0.124       0.020       0.253     6.136   0.000     0.084      0.164
-------------------------------------------------------------------------------

                           Stepwise Selection Summary
-------------------------------------------------------------------------------
                         Added/               Adj.
Step     Variable       Removed    R-Square  R-Square    C(p)        AIC       RMSE
-------------------------------------------------------------------------------
   1   diabetes:age2   addition      0.200     0.199    30.1580   11362.6333   13.6970
   2      weight       addition      0.217     0.215     2.3790   11335.0892   13.5588
   3     diabetes      addition      0.217     0.215     3.0660   11335.7725   13.5573
   4       lage        addition      0.217     0.214     5.0560   11337.7626   13.5621
   5     diabetes       removal      0.217     0.214     4.3590   11337.0698   13.5636
   6       age2        addition      0.217     0.214     6.3590   11337.0698   13.5636
   7      weight        removal      0.200     0.198    33.8080   11364.2895   13.7002
   8  weight:diabetes  addition      0.217     0.214     5.4730   11338.1811   13.5641
   9   diabetes:age2    removal      0.217     0.215     3.4960   11336.2045   13.5594
  10       age         addition      0.218     0.216     3.1620   11335.8602   13.5529
  11       age2        removal       0.218     0.216     1.8100   11334.5121   13.5512
-------------------------------------------------------------------------------
```

Our output suggests a slight improvement over the previous model. The model now explains 21.6 percent of the variability in the response. This is better than what we started with but still rather low, suggesting limitations with the data. To get a model that better explains the variability in our response, we would need more predictors that correlate with the response. For example, we might want to include information about gender, family medical history, and exercise habits in our model.

However, it is also important to note that when working with behavioral data, it is common to run into difficulties building a model that explains most of the variability in the response. This is as a result of the unpredictable nature of human behavior.

Looking at the coefficient estimates from our output, we see that *lage*, *age*, *weight:diabetes0*, and *weight:diabetes1* are all significant. This suggests that there is a nonlinear relationship between age and blood pressure. It also shows that there is an interaction between weight and diabetes. The weight and diabetes interactions can be interpreted as follows: for patients without diabetes, a 1kg increase in weight results in an increase in systolic blood pressure of 0.96 points. However, for patients with diabetes, a 1kg increase in weight results in a 1.24 point increase in systolic blood pressure.

EXERCISES

1. You are working with a movie production company to evaluate the potential success of new feature films. As you begin your work, you gather data elements about all feature films released in the past 10 years. Identify five data elements that you think would be useful to gather for analysis. Characterize your expectations for each variable, stating whether you believe it would be positively correlated or negatively correlated with box office revenue and whether you believe each correlation would be relatively strong, moderate, or weak.
2. Using the blood pressure dataset from the use case in this chapter, produce a correlation plot. Use the `corrplot.mixed` function and generate a plot that shows the correlation coefficients visually above the diagonal and numerically below the diagonal. Provide an interpretation of your results.
3. You are working with college admission data and trying to determine whether you can predict a student's future GPA based upon their college admission test score. The test is scored on a scale of 0–100, while GPA is measured on a scale of 0.0–4.0.

```
Call:
lm(formula = gpa ~ test)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3050 -0.1237  0.0525  0.1412  0.2000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.695000   0.531954   1.307   0.2392
test        0.033000   0.006205   5.318   0.0018 **
---
```