Chapter_6_insurance_example

Fran Camacho

2025-02-06

Chapter 6 - Linear regression

Example from the book "Machine Learning with R", by Brett Lantz:

Predicting auto insurance claims costs using linear regression

Step 1 – collecting data

(The dataset for this example is a simulation created for this book based on demographics and traffic statistics from the United States government)

The insurance dataset includes 20,000 examples of beneficiaries enrolled in the hypothetical au- tomobile vehicle insurance plan. This is much smaller than the typical datasets used by practicing actuaries, especially for very rare outcomes, but the size has been reduced to allow analysis even on computers with limited memory.

```
# import the CSV file
insurance <- read.csv(file.path("Chapter06", "autoinsurance.csv"), stringsAsFactors = TRUE)
```

Step 2 – exploring and preparing the data

Structure of the dataset:

```
str(insurance)
```

```
## 'data.frame':    20000 obs. of  11 variables:
##  $ age                : int  19 30 39 64 33 27 62 39 67 38 ...
##  $ geo_area           : Factor w/ 3 levels "rural","suburban",..: 3 3 3 2 2 2 2 3 1 3 ...
##  $ vehicle_type       : Factor w/ 4 levels "car","minivan",..: 3 3 1 3 3 3 1 4 1 1 ...
##  $ est_value          : int  28811 52603 113870 35228 19190 22899 42172 29468 56469 103371 ...
##  $ miles_driven       : int  11700 12811 9784 17400 14665 12570 15539 10763 13811 11805 ...
##  $ college_grad_ind   : int  0 1 1 0 0 1 1 0 1 1 ...
##  $ speeding_ticket_ind: int  1 0 0 0 0 0 0 0 0 0 ...
##  $ hard_braking_ind   : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ late_driving_ind   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ clean_driving_ind  : int  0 1 0 1 1 0 1 1 0 1 ...
##  $ expenses           : num  0 6311 49684 0 0 ...
```

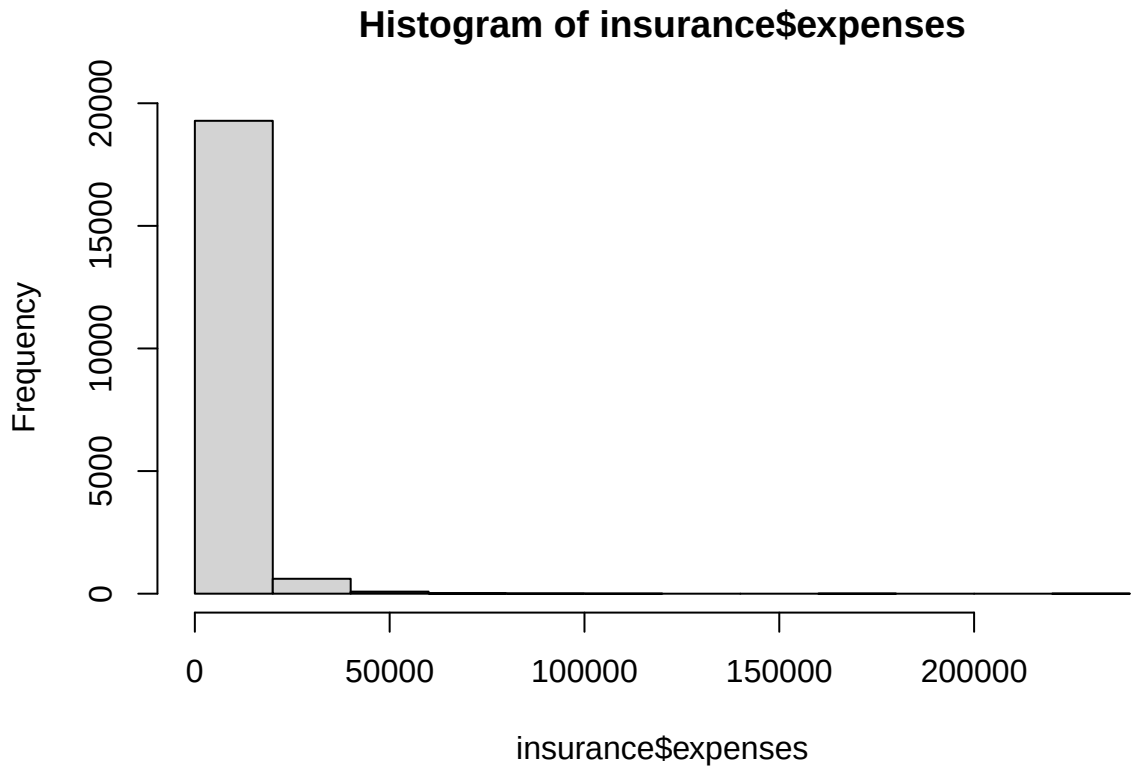
Our model's dependent variable is expenses, which measures the loss or damages each person claimed under the insurance plan for the year.

```
summary(insurance$expenses)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0    1709       0  232797
```

The minimum, first quartile, median, and third quartile are all zero, which implies that at least 75% of the beneficiaries had no expenses in the calendar year. The fact that the mean value is greater than the median gives us the sense that the distribution of insurance expenses is right- skewed, but the skew is likely to be quite extreme as the average expense was $1,709 while the maximum was $232,797. We can confirm this visually using a histogram:

```
hist(insurance$expenses)
```



As expected, the figure shows a right-skewed distribution with a huge spike at zero, reflecting the fact that only a small portion (about 8.6%) made an insurance claim. Among those that did claim a vehicle loss or damages, the tail end of the distribution extends far to the right, beyond $200,000 worth of expenses for the costliest injuries. Although this distribution is not ideal for linear regression, knowing this weakness ahead of time may help us design a better-fitting model later. For now, using only the distribution of expenses, we can say that the average beneficiary should be charged an annual premium of $1,709 for the insurer to break even, or about $150 a month per subscriber for a slight profit.

Before we add additional predictors, it is important to note that regression models require that every feature is numeric, yet we have two factor-type features in our data frame.

```
table(insurance$geo_area)
```

```
##
##    rural suburban    urban
##     3622     8727     7651
```

```
table(insurance$vehicle_type)
```

```
##
##     car minivan     suv   truck
##    5801     726    9838    3635
```

Here, we see that the data has been divided nearly evenly between urban and suburban areas, but rural is a much smaller portion of the data. Additionally, SUVs are the most popular vehicle type, followed by cars and trucks, with minivans in a distant fourth place. We will see how R's linear regression function handles these factor variables shortly.

Exploring relationships between features – the correlation matrix

Before fitting a regression model to data, it can be useful to determine how the independent variables are related to the dependent variable and each other. A correlation matrix provides a quick overview of these relationships. Given a set of variables, it provides a correlation for each pairwise relationship.

```
cor(insurance[c("age", "est_value", "miles_driven", "expenses")])
```

```
##                       age   est_value miles_driven     expenses
## age           1.000000000 -0.05990552   0.04812638 -0.009121269
## est_value    -0.059905524  1.00000000  -0.01804807  0.088100468
## miles_driven  0.048126376 -0.01804807   1.00000000  0.062146507
## expenses     -0.009121269  0.08810047   0.06214651  1.000000000
```

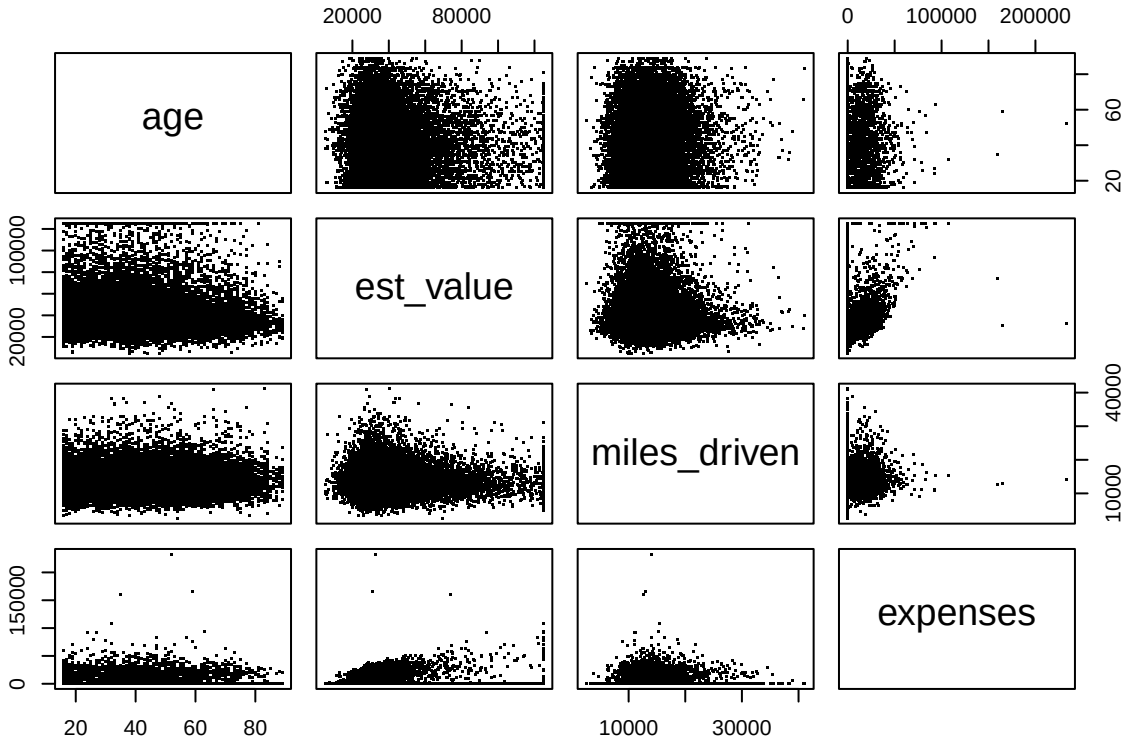
None of the correlations in the matrix are very strong, but the associations do match with common sense. For instance, age and expenses appear to have a weak negative correlation, meaning that as someone ages, their expected insurance cost goes down slightly—probably reflecting the greater driving experience. There are also positive correlations between est_value and expenses and miles_driven and expenses, which indicate that more valuable cars and more extensive driving lead to greater expenses.

Visualizing relationships between features – the scatterplot matrix

It can also be helpful to visualize the relationships between numeric features with scatterplots. Although we could create a scatterplot for each possible relationship, doing so for a large set of features quickly becomes tedious. An alternative is to create a scatterplot matrix (sometimes abbreviated as SPLOM).

Given the relatively large size of our insurance dataset, we'll set the plot character parameter pch = "." (just a dot) to make the visualization easier to read, and we'll limit the columns to the four variables of interest:

```
pairs(insurance[c("age", "est_value", "miles_driven", "expenses")], pch = ".")
```

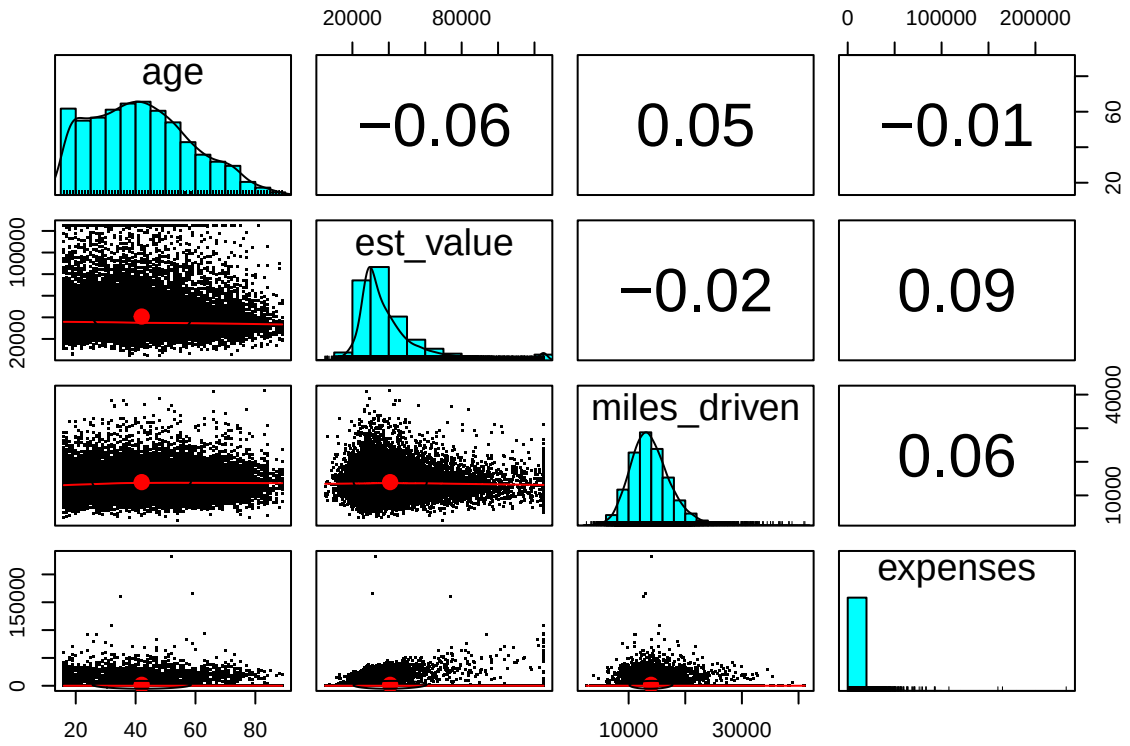


By adding more information to the plot, it can be made even more useful. An enhanced scatter plot matrix can be created with the pairs.panels() function in the psych package.

```
install.packages("psych")
```

```
## The following package(s) will be installed:
## - psych [2.4.12]
## These packages will be installed into "~/git/masterML/renv/library/linux-debian-bookworm/R-4.4/x86_64
##
## # Installing packages --------------------------------------------------------
## - Installing psych ...                          OK [linked from cache]
## Successfully installed 1 package in 11 milliseconds.
```

```
library(psych)
pairs.panels(insurance[c("age", "est_value", "miles_driven", "expenses")], pch = ".")
```



In the pairs.panels() output, the scatterplots above the diagonal are replaced with a correlation matrix. The diagonal now contains histograms depicting the distribution of values for each feature. Finally, the scatterplots below the diagonal are presented with additional visual information.

Step 3: Training a model on the data

To fit a linear regression model to data with R, the lm() function can be used:

```
#ins_model <- lm(expenses ~ age + geo_area + vehicle_type + est_value + miles_driven + college_grad_ind
#                 hard_braking_ind + late_driving_ind + clean_driving_ind, data = insurance)
ins_model <- lm(expenses ~ ., data = insurance)  #shorter
ins_model
```

```
##
## Call:
## lm(formula = expenses ~ ., data = insurance)
##
## Coefficients:
##         (Intercept)                  age     geo_areasuburban
##          -1.155e+03           -1.886e+00            1.911e+02
##       geo_areaurban  vehicle_typeminivan      vehicle_typesuv
##           1.691e+02            1.153e+02           -1.970e+01
##   vehicle_typetruck            est_value         miles_driven
##           2.157e+01            3.115e-02            1.190e-01
##    college_grad_ind  speeding_ticket_ind     hard_braking_ind
```

```
##          -2.504e+01            1.558e+02            1.185e+01
##    late_driving_ind    clean_driving_ind
##           3.625e+02           -2.390e+02
```

For instance, for each additional year of age, we would expect $1.89 lower insurance claims costs on average, assuming everything else is held equal. Similarly, we would expect $0.12 higher claims for each additional mile driven and $0.03 higher per dollar of insured value, all else equal.

You might notice that although we only specified 10 features in our model formula, there are 13 coefficients reported in addition to the intercept. This happened because the lm() function automatically applies dummy coding to each of the factor-type variables included in the model.

```
ins_model2 <- lm(expenses ~ ., data = insurance)
ins_model2
```

```
##
## Call:
## lm(formula = expenses ~ ., data = insurance)
##
## Coefficients:
##         (Intercept)                  age     geo_areasuburban
##          -1.155e+03           -1.886e+00            1.911e+02
##       geo_areaurban  vehicle_typeminivan      vehicle_typesuv
##           1.691e+02            1.153e+02           -1.970e+01
##   vehicle_typetruck             est_value         miles_driven
##           2.157e+01            3.115e-02            1.190e-01
##    college_grad_ind  speeding_ticket_ind     hard_braking_ind
##          -2.504e+01            1.558e+02            1.185e+01
##    late_driving_ind    clean_driving_ind
##           3.625e+02           -2.390e+02
```

In our model, R automatically held out the geo_arearural and vehicle_typecar variables, making rural car owners the reference group. Thus, urban dwellers have $169.11 more claims costs each year relative to rural areas and trucks cost the insurer an average of $21.57 more than cars per year.

Step 4 – evaluating model performance

```
summary(ins_model)
```

```
##
## Call:
## lm(formula = expenses ~ ., data = insurance)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6707  -1989  -1492  -1057 231252
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.155e+03  3.514e+02  -3.287  0.00101 **
## age                 -1.886e+00  3.145e+00  -0.600  0.54866
```

```
## geo_areasuburban     1.911e+02  1.432e+02   1.334  0.18210
## geo_areaurban        1.691e+02  1.579e+02   1.071  0.28402
## vehicle_typeminivan  1.153e+02  2.766e+02   0.417  0.67683
## vehicle_typesuv     -1.970e+01  1.180e+02  -0.167  0.86743
## vehicle_typetruck    2.157e+01  1.536e+02   0.140  0.88835
## est_value            3.114e-02  2.497e-03  12.475  < 2e-16 ***
## miles_driven         1.190e-01  1.433e-02   8.305  < 2e-16 ***
## college_grad_ind    -2.504e+01  1.156e+02  -0.217  0.82848
## speeding_ticket_ind  1.558e+02  1.402e+02   1.112  0.26624
## hard_braking_ind     1.185e+01  1.069e+02   0.111  0.91178
## late_driving_ind     3.625e+02  2.247e+02   1.614  0.10665
## clean_driving_ind   -2.390e+02  1.111e+02  -2.152  0.03140 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6995 on 19986 degrees of freedom
## Multiple R-squared:  0.01241,    Adjusted R-squared:  0.01176
## F-statistic: 19.31 on 13 and 19986 DF,  p-value: < 2.2e-16
```

There are three main ways to evaluate the performance, or fit, of our model:

1.The Residuals section provides summary statistics for the prediction errors.

Since a residual is equal to the true value minus the predicted value, the maximum error of 231252 suggests that the model under-predicted expenses by more than $230,000 for at least one observation.

On the other hand, the majority of errors are relatively small negative values, which means that we are over-estimating expenses for most enrollees. This is exactly how the insurance company can afford to cover the expenses for costly accidents.

2.Better explanation (for me) from another place:

https://blog.minitab.com/en/adventures-in-statistics-2/how-to-interpret-regression-analysis-results-p-values-and-coefficients

"The p-value for each term tests the **null hypothesis** that the coefficient is equal to zero (no effect). A low p-value (< 0.05) indicates that you can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be a meaningful addition to your model because changes in the predictor's value are related to changes in the response variable."

"Conversely, a larger (insignificant) p-value suggests that changes in the predictor are not associated with changes in the response."

3.The Multiple R-squared value (also called the coefficient of determination) provides a measure of how well our model explains the values of the dependent variable. (How well the regression model fits the data points).

It is similar to the correlation coefficient in that the closer the value is to 1.0, the better the model perfectly explains the data. Since the R-squared value is 0.01241, we know that the model explains about 1.2 percent of the variation in the dependent variable. Because models with more features always explain more variation, the Adjusted R-squared value corrects R-squared by penalizing models with a large number of independent variables. This is useful for comparing the performance of models with different numbers of explanatory variables.

Other explanations:

- w3schools:

"The value of R-Squared is always between 0 to 1 (0% to 100%).

A high R-Squared value means that many data points are close to the linear regression function line. A low R-Squared value means that the linear regression function line does not fit the data well."

In the book the author says:
"Given the preceding three performance indicators, our model is performing well enough."

?!

https://statisticsbyjim.com/regression/interpret-r-squared-regression :

R-squared is a goodness-of-fit measure for linear regression models. This statistic indicates the percentage of the variance in the dependent variable that the independent variables explain collectively. R-squared measures the strength of the relationship between your model and the dependent variable on a convenient 0 – 100% scale.

R-squared has Limitations

You cannot use R-squared to determine whether the coefficient estimates and predictions are biased, which is why you must assess the residual plots.

Others:

"R-squared is a statistical measure that indicates how much of the variation of a dependent variable is explained by an independent variable in a regression model."

"Adjusted R-squared, a modified version of R-squared, adds precision and reliability by considering the impact of additional independent variables that tend to skew the results of R-squared measurements."

"One misconception about regression analysis is that a low R-squared value is always a bad thing."

Step 5 – improving model performance

As mentioned previously, a key difference between regression modeling and other machine learning approaches is that regression typically leaves feature selection and model specification to the user.

Consequently, if we have subject-matter knowledge about how a feature is related to the outcome, we can use this information to inform the model specification and potentially improve the model's performance.

Model specification – adding nonlinear relationships

To add the nonlinear age to the model, we simply need to create a new variable:

```
insurance$age2 <- insurance$age^2
```

Then, when we produce our improved model, we'll add both age and age2 to the lm() formula using the form expenses ~ age + age2.

Model specification – adding interaction effects

To interact the hard braking indicator (hard_braking_ind) with the late driving indicator (late_driving_ind), we would write a formula in the form **expenses ~ hard_braking_ind*late_driving_ind**.

The * operator is a shorthand that instructs R to model like this: expenses ~ hard_braking_ind + late_driving_ind + hard_braking_ind:late_driving_ind.

The colon operator (:) in the expanded form indicates that hard_braking_ind:late_driving_ind is the interaction between the two variables. Note that the expanded form automatically also included the individual hardbraking_ind and late_driving_ind variables as well as their interaction.

Putting it all together – an improved regression model

```
ins_model2 <- lm(expenses ~ . + hard_braking_ind:late_driving_ind, data = insurance)
```

we summarize the results:

```
summary(ins_model2)
```

```
##
## Call:
## lm(formula = expenses ~ . + hard_braking_ind:late_driving_ind,
##     data = insurance)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6618  -1996  -1491  -1044 231358
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -5.350e+02  4.571e+02  -1.170   0.2419
## age                              -3.314e+01  1.537e+01  -2.157   0.0310 *
## geo_areasuburban                  1.788e+02  1.433e+02   1.248   0.2121
## geo_areaurban                     1.325e+02  1.587e+02   0.835   0.4040
## vehicle_typeminivan               1.717e+02  2.779e+02   0.618   0.5367
## vehicle_typesuv                  -8.006e+00  1.181e+02  -0.068   0.9460
## vehicle_typetruck                 2.643e+01  1.537e+02   0.172   0.8634
## est_value                         3.118e-02  2.496e-03  12.489   <2e-16 ***
## miles_driven                      1.187e-01  1.433e-02   8.289   <2e-16 ***
## college_grad_ind                  1.725e+01  1.174e+02   0.147   0.8832
## speeding_ticket_ind               1.551e+02  1.401e+02   1.107   0.2685
## hard_braking_ind                 -1.244e+01  1.098e+02  -0.113   0.9098
## late_driving_ind                  1.833e+02  2.842e+02   0.645   0.5189
## clean_driving_ind                -2.328e+02  1.111e+02  -2.096   0.0361 *
## age2                              3.432e-01  1.653e-01   2.076   0.0380 *
## hard_braking_ind:late_driving_ind  4.691e+02  4.617e+02   1.016   0.3096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6995 on 19984 degrees of freedom
## Multiple R-squared:  0.01267,    Adjusted R-squared:  0.01193
## F-statistic:  17.1 on 15 and 19984 DF,  p-value: < 2.2e-16
```

Although the R-squared and adjusted R-squared values didn't change much compared to the previous model, the new features present some interesting insights. In particular, the estimate for age is relatively large and negative (lower expenses) but age2 is relatively small and positive (higher expenses). However, because age squared grows faster than age, expenses will begin to rise for very high age groups. The overall effect is a U-shaped expense curve, where the youngest and oldest enrollees are predicted to have higher expenses. The interaction of hard_braking_ind and late_driving_ind is also interesting, as it is a relatively large positive. Although the inter- action is not statistically significant, the direction of the effect implies that driving late at night is especially dangerous if you are the type of driver that already drives dangerously.

?! (pues vale)

Making predictions with a regression model

After examining the estimated regression coefficients and fit statistics, we can also use the mod- el to predict the expenses of future enrollees on the insurance plan. To illustrate the process of making predictions, let's first apply the model to the original training data using the predict() function as follows:

```
insurance$pred <- predict(ins_model2, insurance)
```

This saves the predictions as a new vector named pred in the insurance data frame. We can then compute the **correlation** between the predicted and actual costs of insurance:

```
cor(insurance$pred, insurance$expenses)
```

```
## [1] 0.1125714
```

The correlation of 0.11 suggests a relatively weak linear relationship between the predicted and actual values, which is disappointing but not too surprising given the seemingly random nature of motor vehicle accidents. It can also be useful to examine this finding as a scatterplot.

The fol- lowing R commands plot the relationship and then add an identity line with an intercept equal to zero and a slope equal to one. The col, lwd, and lty parameters affect the line color, width, and type, respectively:

```
plot(insurance$pred, insurance$expenses)
abline(a = 0, b = 1, col = "red", lwd = 3, lty = 2)
```

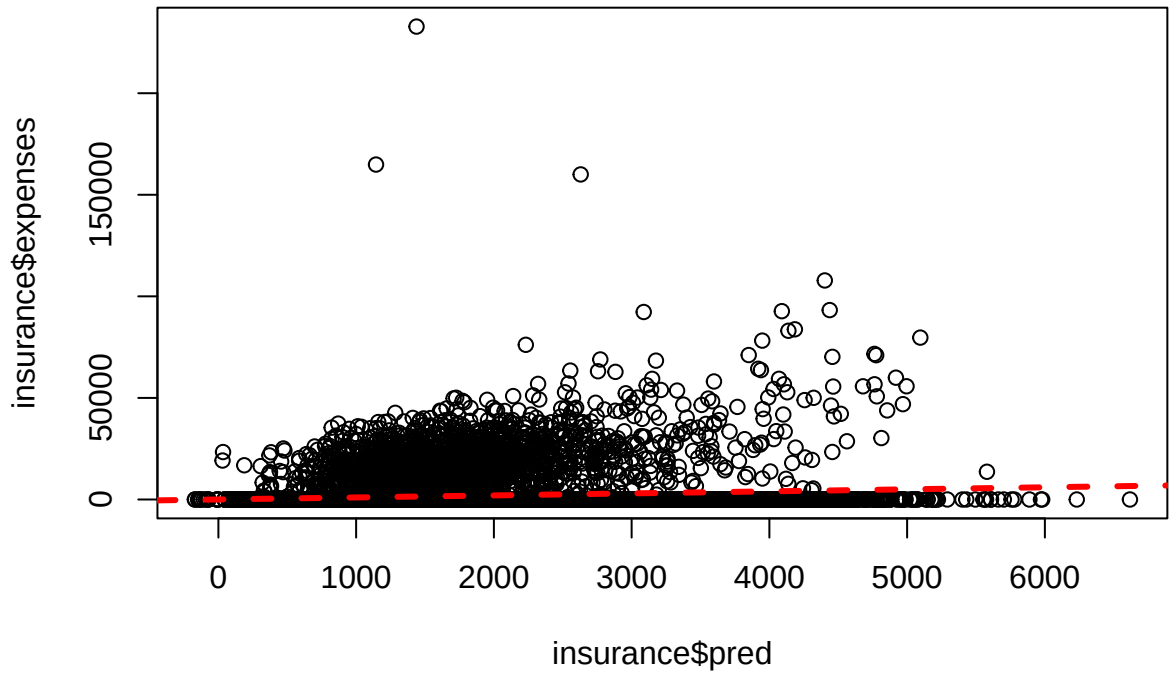


Figure 6.12: In this scatterplot, points falling on or near the diagonal dashed line where y = x indicate the predictions that were very close to the actual values

The off-diagonal points falling above the line are cases where the actual expenses are greater than expected, while cases falling below the line are those less than expected. We can see here that a small number of people with much larger-than-expected expenses are balanced by a huge number of people with slightly smaller-than-expected expenses.

Now, suppose you would like to forecast the expenses for potential new enrollees on the insur- ance plan. To do this, you must provide the predict() function a data frame with the prospective drivers' data.

For example, to estimate the insurance expenses for a 30-year-old, rural driver of a truck valued at $25,000 driven for about 14,000 miles annually with a clean driving record:

```
predict(ins_model2, data.frame(age = 30, age2 = 30^2, geo_area = "rural", vehicle_type = "truck", est_va
                               miles_driven = 14000, college_grad_ind = 0, speeding_ticket_ind = 0, ha
                               late_driving_ind = 0, clean_driving_ind = 1))
```

```
##        1
## 1015.059
```

Using this value, the insurance company would need to charge about $1,015 annually to break even for this demographic group. To compare the rate for someone who is otherwise similar except for a history of a recent accident, use the predict() function in much the same way:

```
predict(ins_model2, data.frame(age = 30, age2 = 30^2, geo_area = "rural", vehicle_type = "truck", est_va
                               miles_driven = 14000, college_grad_ind = 0, speeding_ticket_ind = 0, ha
                               late_driving_ind = 0, clean_driving_ind = 0))
```

```
##        1
## 1247.903
```

Note that the difference between these two values, **1015.059 − 1247.903 = -232.844**, is the same as the estimated regression model coefficient for clean_driving_ind. On average, drivers with a clean history are estimated to have about $232.84 less in expenses for the plan per year, all else being equal.

This illustrates the more general fact that the predicted expenses are a sum of each of the regression coefficients times their corresponding value in the prediction data frame. For instance, using the model's regression coefficient of 0.118748 for the miles driven, we can predict that adding 10,000 additional miles will result in an increase in expenses of 10,000 * 0.118748 = 1187.48, which can be confirmed as follows: