Principles of Economics with R (PoE)

ryannthegeek

6/17/22

Table of contents

1 The Simple Linear Regression Model **2**

- 1.1 The Simple Linear Regression Model 2
  - 1.1.1 Assumptions of simple linear model 2
- 1.2 Example: Food Expenditure versus Income 3
- 1.3 Estimating a Linear Regression 4
- 1.4 Prediction with the Linear Regression Model 7
- 1.5 Repeated Samples to Assess Regression Coefficients 7
- 1.6 Estimated Variances and Covariance of Regression Coefficients 7
- 1.7 Non-Linear Relationships 8
  - 1.7.1 The quadratic model 8
  - 1.7.2 The log-linear model 11
    - Histogram of price 11
    - Histogram of log price 11
    - Drawing the fitted values curve of the log-linear model 13
- 1.8 Using Indicator Variables in a Regression 15
  - 1.8.1 fitting a regression model 16
- 1.9 Monte Carlo Simulation 16

List of Figures

1 a plot of wage against education 3
2 A scatter diagram for the food expenditure versus income 4
3 A regression on food expenditure against income 6
4 A scatter plot of sale price of 1080 houses in Baton Rouge, LA against square feet 9
5 Fitting a quadratic model to the br dataset 10
6 Fitting a quadratic model to the br dataset by specifying se = F 10
7 Histogram of price 11

8 Histogram of price 12
9 Fitting a quadratic model to the br dataset 14
10 Fitting a quadratic model to the br dataset 15

List of Tables

1 The Simple Linear Regression Model

1.1 The Simple Linear Regression Model

A simple linear regression model assumes that a linear relationship exists between the conditional expectation of a dependent variable y and an independent variable x.

The assumed relationship in a linear regression model has the form:

$$y_i = \beta_0 + \beta_1 x_i + e_i \tag{1}$$

where:

- y is the dependent variable
- x is the independent variable
- e is an error term
- σ^2 is the variance of the error term
- β_0 is the intercept parameter or coefficient
- β_1 is the slope parameter or coefficient
- i stands for the i^{th} observation in the data set, $i = 1, 2, ..., N$
- N is the number of observations in the data set.

The *predicted*, or estimated value of y given x is given by:

$$\hat{y} = \beta_0 + \beta_1 x$$

1.1.1 Assumptions of simple linear model

- The values of x are previously chosen (therefore, they are non-random).
- The variance of the error term σ^2 is the same for all values of x.
- There is no connection between one observation and another (no correlation between the error terms of two observations).
- The expected value of the error term for any value of x is zero.
- The error term is normally distributed.

```r
require(PoEdata)
data("cps_small")
attach(cps_small)
names(cps_small)
```

```
[1] "wage"    "educ"    "exper"   "female"  "black"   "white"   "midwest"
[8] "south"   "west"
```

```r
require(ggplot2)
ggplot() +
  geom_point(data = cps_small, aes(x = educ, y = wage)) +
  ggtitle("A plot of wage against education") +
  xlab("Education") +
  ylab("Wage")
```

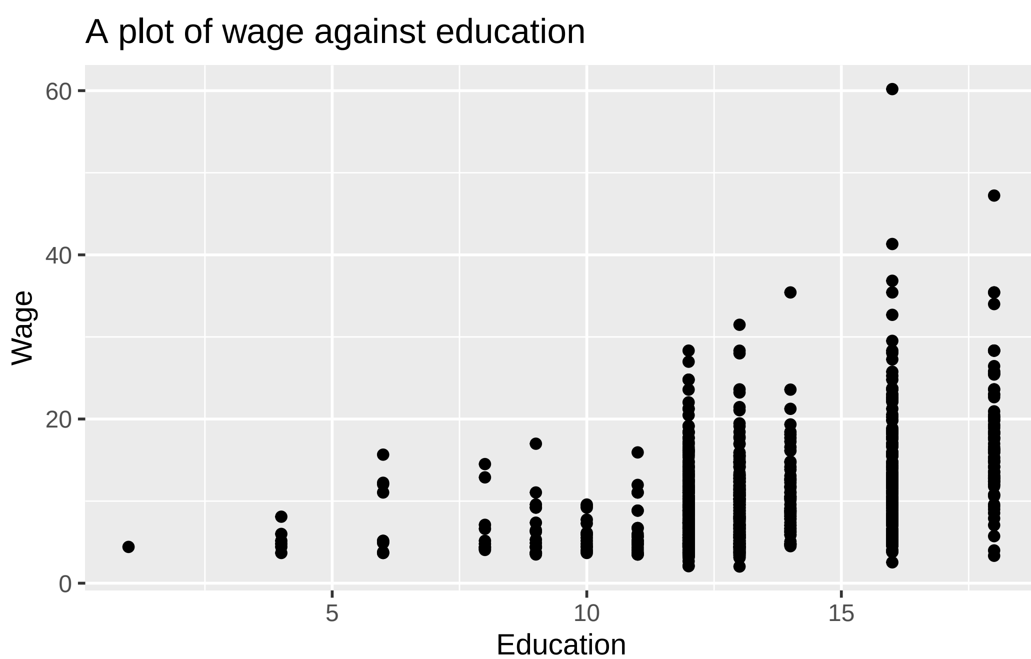

Figure 1: a plot of wage against education

1.2 Example: Food Expenditure versus Income

```r
data("food")
attach(food)
names(food)
```

```
[1] "food_exp" "income"
```

```r
max(food_exp);max(income)
```

```
[1] 587.66
```

```
[1] 33.4
```

```
ggplot() +
  geom_point(data = food, aes(x = income, y = food_exp)) +
  scale_x_continuous(name = "weekly income in $100", limits = c(0, 34)) +
  scale_y_continuous(name = "weekly food expenditure in $", limits = c(0, 588))
    +
  ggtitle("A scatter plot of food expenditure against income")
```

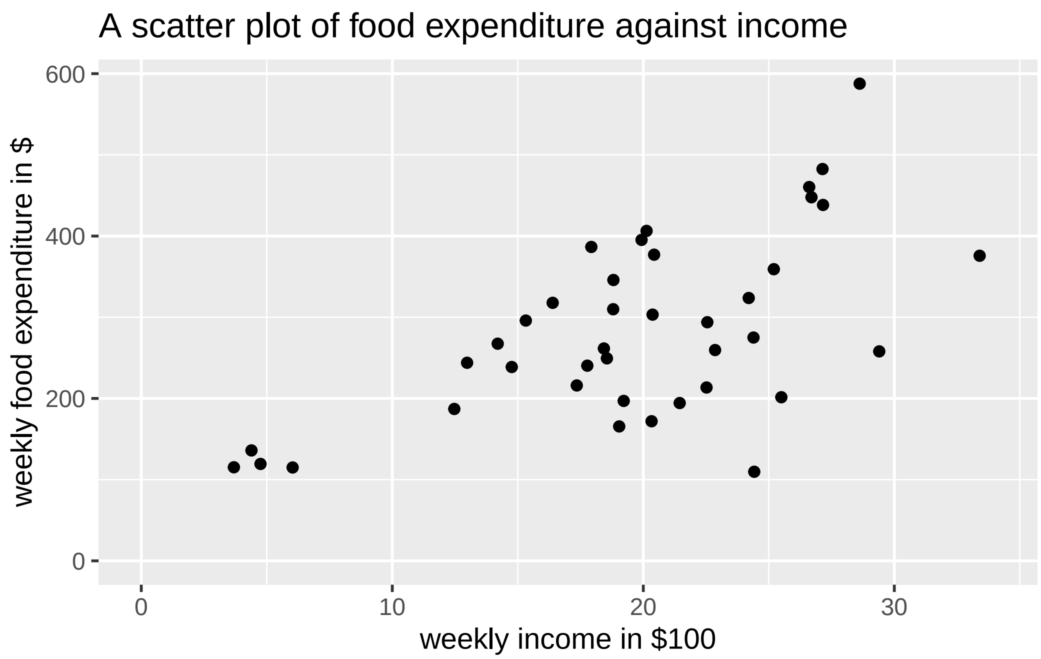

Figure 2: A scatter diagram for the food expenditure versus income

1.3 Estimating a Linear Regression

$$\text{foodexpenditure} = \beta_0 + \beta_1 \text{income} + e$$

```
m1 <- lm(food_exp ~ income, data = food)
b0 <- coef(m1)[[1]]
b1 <- coef(m1)[[2]]
summary_m1 <- summary(m1);summary_m1
```

```

Call:
lm(formula = food_exp ~ income, data = food)
```

```

Residuals:
     Min       1Q   Median       3Q      Max
-223.025  -50.816   -6.324   67.879  212.044

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   83.416     43.410   1.922   0.0622 .
income        10.210      2.093   4.877 1.95e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 89.52 on 38 degrees of freedom
Multiple R-squared:  0.385, Adjusted R-squared:  0.3688
F-statistic: 23.79 on 1 and 38 DF,  p-value: 1.946e-05
```

```
coef(m1)
```

```
(Intercept)      income
   83.41600    10.20964
```

The intercept parameter β_0 is usually of little importance in econometric models; we are mostly interested in the slope parameter β_1.
The estimated value of β_1 suggests that the **food expenditure** for an average family increases by 10.209643 when the **family income** increases by 1 unit, which in this case is $100.
The R function `geom_abline()` adds the regression line.

```
ggplot() +
  geom_point(data = food, aes(x = income, y = food_exp)) +
  scale_x_continuous(name = "weekly income in $100", limits = c(0, 34)) +
  scale_y_continuous(name = "weekly food expenditure in $", limits = c(0, 588))
    +
  geom_abline(intercept = b0, slope = b1, color = "skyblue", linetype = "solid",
     size = 1.5) +
  ggtitle("A regression on food expenditure against income")
```

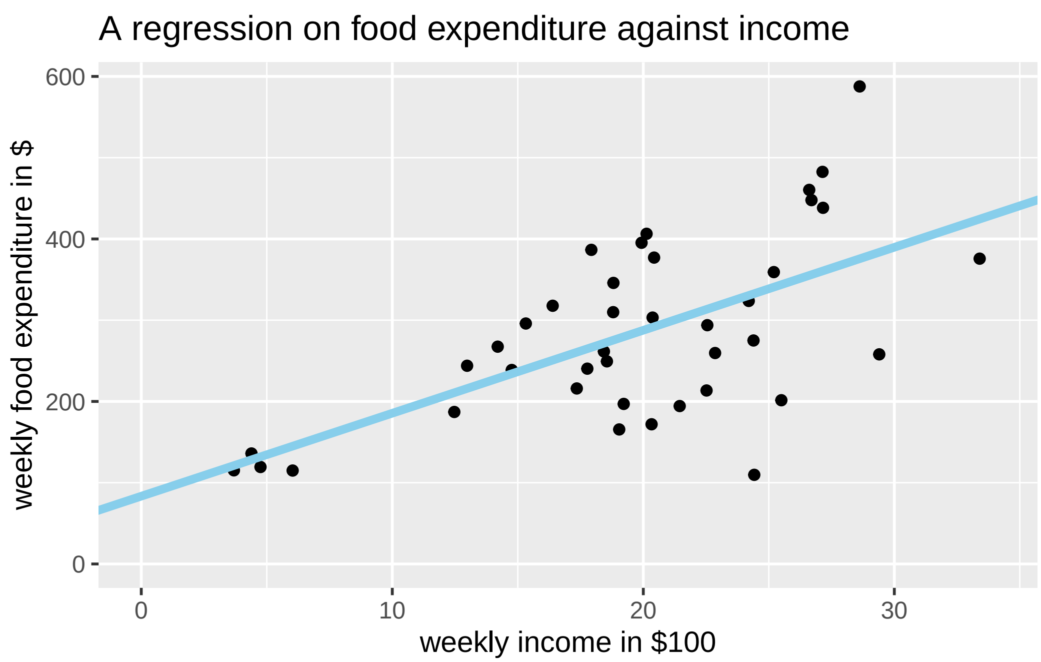

Figure 3: A regression on food expenditure against income

list the names of all results in each object

```
names(m1)
```

```
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

```
names(summary_m1)
```

```
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
m1$coefficients
```

```
(Intercept)      income
   83.41600    10.20964
```

```
summary_m1$coefficients
```

```
            Estimate Std. Error  t value     Pr(>|t|)
(Intercept) 83.41600  43.410163 1.921578 6.218242e-02
income      10.20964   2.093264 4.877381 1.945862e-05
```

1.4 Prediction with the Linear Regression Model

```
newx <- data.frame(income = c(20, 25, 27))
yhat <- predict(m1, newx)
names(yhat) <- c("income=$2000", "$2500", "$2700")
yhat  # prints the result
```

```
income=$2000        $2500        $2700
    287.6089     338.6571     359.0764
```

1.5 Repeated Samples to Assess Regression Coefficients

Let us construct a number of random sub samples from the food data and re-calculate β_0 and β_1. A random sub sample can be constructed using the function sample(), as the following example illustrates only for β_1.

```
N <- nrow(food);N # returns the number of observations in the dataset
```

```
[1] 40
```

```
C <- 50          # desired number of subsamples
S <- 38          # desired sample size

sumb1 <- 0 # initial value
for (i in 1:C){   # a loop over the number of subsamples
  set.seed(3*i)   # a different seed for each subsample
  subsample <- food[sample(1:N, size = S, replace = TRUE), ]
  m2 <- lm(food_exp ~ income, data = subsample)
  #sum b2 for all subsamples:
  sumb1 <- sumb1 + coef(m2)[[2]]
}
print(sumb1/C, digits = 3)
```

```
[1] 9.89
```

The result, $\beta_1 = 9.88$, is the average of 50 estimates of β_1

1.6 Estimated Variances and Covariance of Regression Coefficients

Many applications require estimates of the variances and covariances of the regression coefficients. R stores them in the a `matrix` `vcov()`:

```
varb0 <- vcov(m1)[1, 1];varb0
```

```
[1] 1884.442
```

```
varb1 <- vcov(m1)[2, 2];varb1
```

```
[1] 4.381752
```

```
covb0b1 <- vcov(m1)[1,2];covb0b1
```

```
[1] -85.90316
```

1.7 Non-Linear Relationships

1.7.1 The quadratic model

The quadratic model requires the square of the independent variable.

$$y_i = \beta_0 + \beta_1 x_i^2 + e_i$$

In R, independent variables involving mathematical operators can be included in a regression equation with the function `I()`.
The following example uses the dataset `br` from the package `PoEdata`, which includes the `sale prices` and the `surface area in square feet` of 1080 houses in Baton Rouge, LA.
`Price` is the `sale price in dollars`, and `sqft` is the `surface area in square feet`.

```
data(br) # sometimes attach() function doesn't work, use data() # just always
   use both!!!
attach(br)

ggplot() +
  geom_point(data = br, aes(x = sqft, y = price)) +
  xlab("Totalsquare feet") +
  ylab("Sale price in $") +
  ggtitle("A scatter plot of sale price of 1080 houses in Baton Rouge, LA
    against square feet")

m3 <- lm(price ~ I(sqft^2), data = br)
b0 <- coef(m3)[[1]]
b1 <- coef(m3)[[2]]
sqftx = c(2000, 4000, 6000) # given values for sqft
pricex = b0 + b1*sqftx^2 # prices corresponding to given sqft
DpriceDsqft <- 2*b1*sqftx # marginal effect of sqft on price
elasticity = DpriceDsqft*sqftx/pricex
par.df <- data.frame(b0, b1);par.df
```

```
        b0        b1
1 55776.57 0.0154213
```

```
data.df <- data.frame(sqftx, pricex, DpriceDsqft, elasticity);data.df
```

```
  sqftx    pricex DpriceDsqft elasticity
1  2000 117461.8    61.68521   1.050303
2  4000 302517.4   123.37041   1.631251
3  6000 610943.4   185.05562   1.817408
```

```
## draw a scatter diagram and see how the quadratic function fits the data

ggplot(data = br, aes(x = sqft, y = price)) +
  geom_point() + # add the quadratic curve to the scatter plot
  geom_smooth(method = "lm", formula = y ~ x + I(x^2)) +
  xlab("Totalsquare feet") +
  ylab("Sale price in $") +
  ggtitle("Fitting a quadratic model to the br dataset")

## we can remove the confidence interval by specifying se = F

ggplot(data = br, aes(x = sqft, y = price)) +
  geom_point() + # add the quadratic curve to the scatter plot
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = F) +
  xlab("Totalsquare feet") +
  ylab("Sale price in $") +
  ggtitle("Fitting a quadratic model to the br dataset")
```

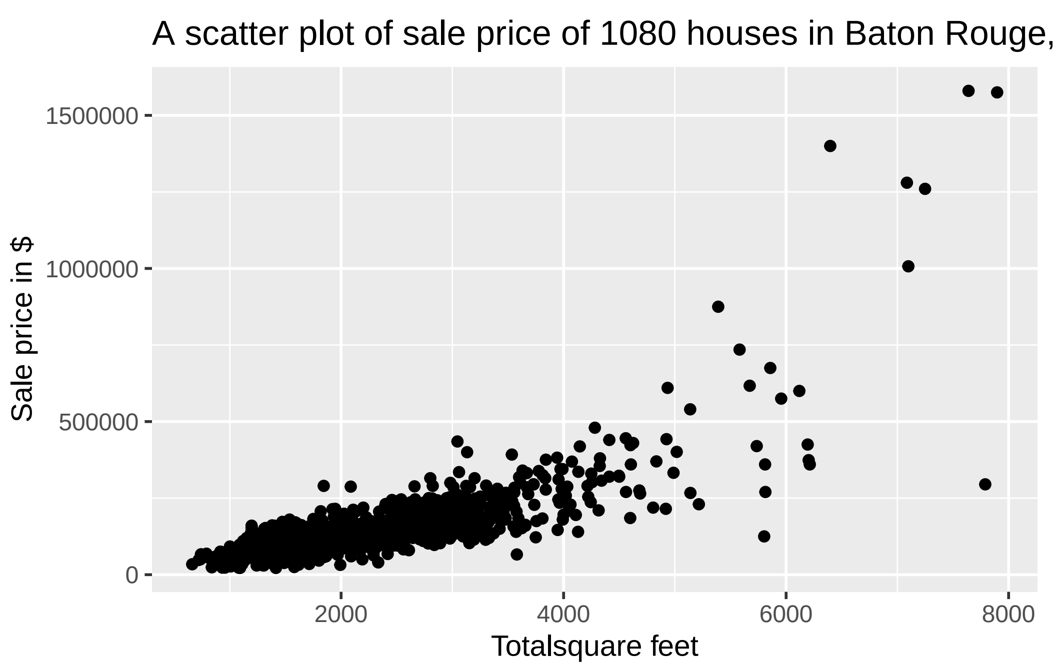

Figure 4: A scatter plot of sale price of 1080 houses in Baton Rouge, LA against square feet

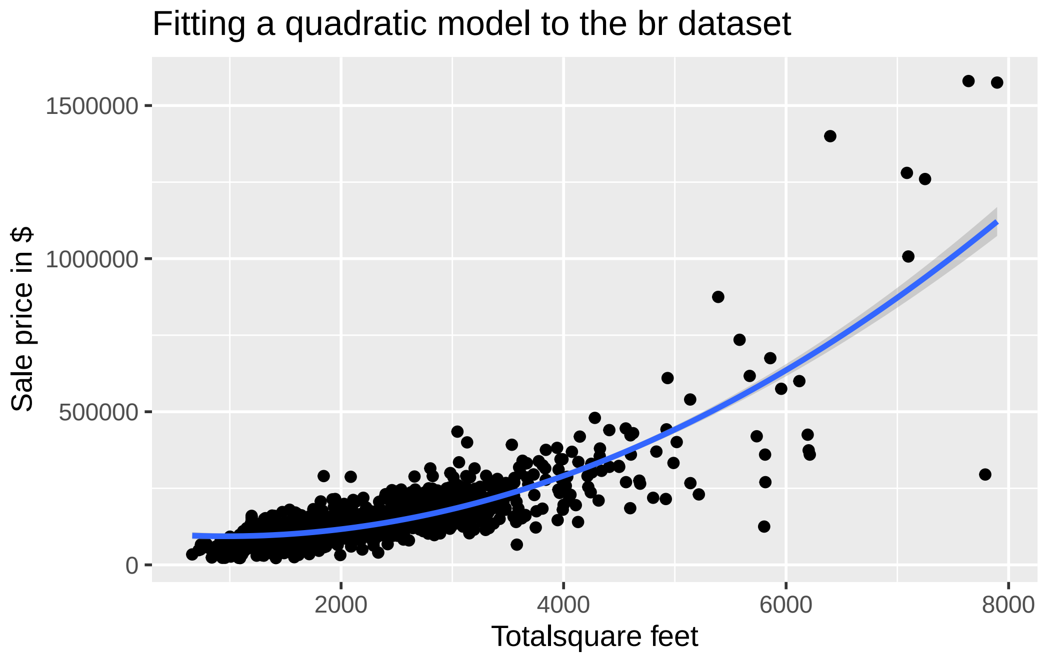

Figure 5: Fitting a quadratic model to the br dataset

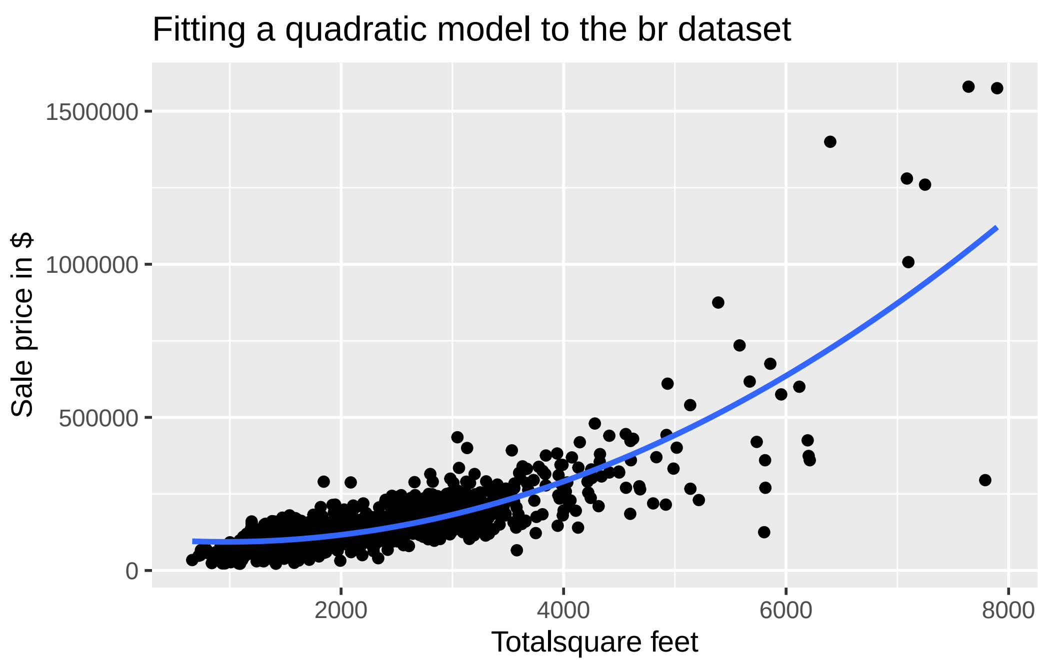

Figure 6: Fitting a quadratic model to the br dataset by specifying se = F

1.7.2 The log-linear model

The log-linear model regresses the log of the dependent variable on a linear expression of the independent variable.
The log linear model is given by:

$$log(y_i) = \beta_0 + \beta_1 x_i + e_i$$

One of the reasons to use the log of an independent variable is to make its distribution closer to the normal distribution

Histogram of price

```
ggplot(data = br, aes(x = price)) +
  geom_histogram()
```

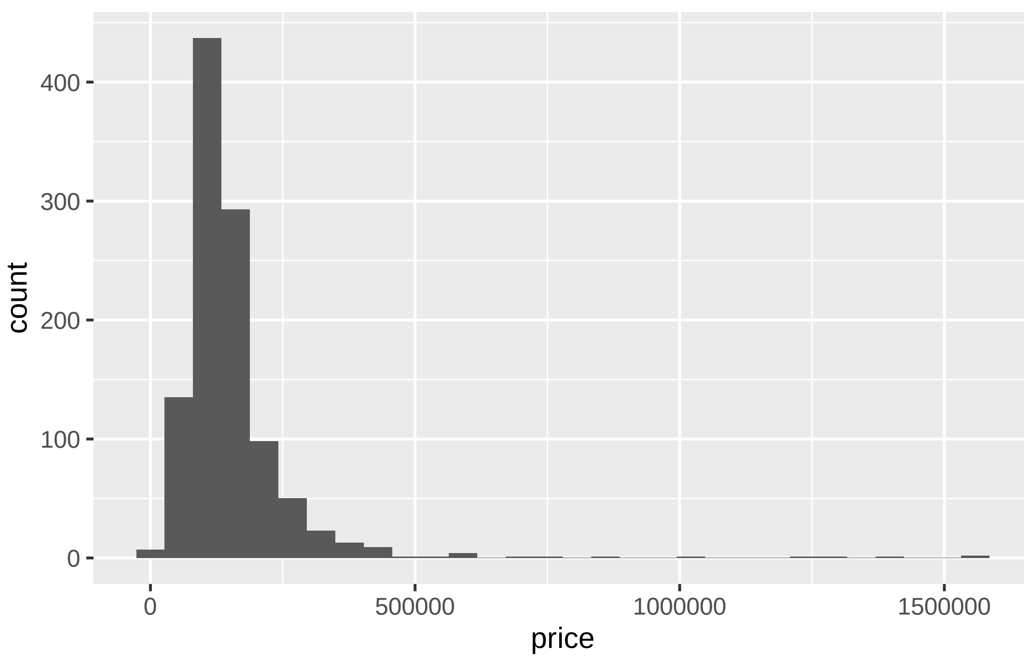

Figure 7: Histogram of price

Histogram of log price

```
ggplot(data = br, aes(x = log(price))) +
  geom_histogram()
```

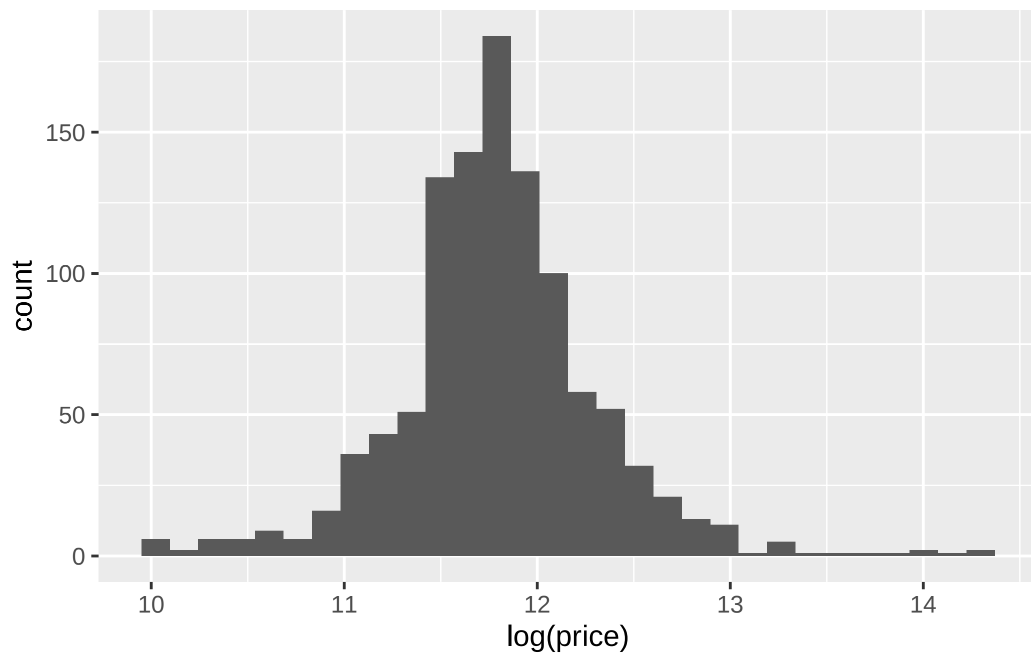

Figure 8: Histogram of price

```
##  It can be noticed that that the log is closer to the normal distribution.
```

We are interested in the *estimates* of the *coefficients* and their interpretation, in the *fitted values* of price, and in the *marginal effect* of an increase in sqft on price.

```
m4 <- lm(log(price) ~ sqft, data = br)
coef(m4)
```

```
 (Intercept)          sqft
1.083860e+01 4.112689e-04
```

```
summary(m4)
```

```

Call:
lm(formula = log(price) ~ sqft, data = br)

Residuals:
     Min       1Q   Median       3Q      Max
-1.48912 -0.13653  0.02876  0.18500  0.98066

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.084e+01  2.461e-02  440.46   <2e-16 ***
sqft        4.113e-04  9.708e-06   42.37   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3215 on 1078 degrees of freedom
Multiple R-squared:  0.6248,    Adjusted R-squared:  0.6244
F-statistic:  1795 on 1 and 1078 DF,  p-value: < 2.2e-16
```

The coefficients are $\beta_0 = 10.84$ and $\beta_1 = 0.00041$, showing that an increase in the surface area (sqft) of an apartment by one unit (1 sqft) increases the price of the apartment by 0.041%. Thus, for a house price of $100,000, an increase of 100 sqft will increase the price by approximately 100(0.041)%, which is equal to 4112.7.
In general, the marginal effect of an increase in x on y is

$$\frac{dy}{dx} = \beta_1 y$$

and the elasticity is:

$$\epsilon = \frac{dy}{dx}\frac{x}{y} = \beta_1 x$$

Drawing the fitted values curve of the log-linear model

```
ggplot(data = br, aes(x = sqft, y = price)) +
  geom_point() + # add the quadratic curve to the scatter plot
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = F) +
  xlab("Totalsquare feet") +
  ylab("Sale price in $") +
  ggtitle("Fitting a quadratic model to the br dataset")
```

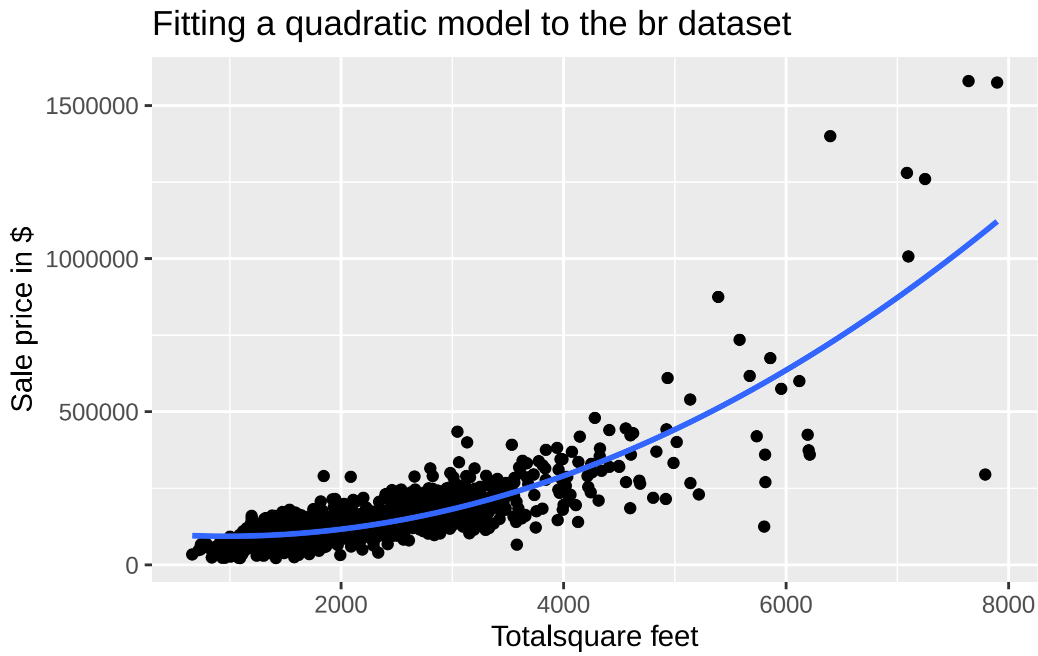

Figure 9: Fitting a quadratic model to the br dataset

```
ordat <- br[order(br$sqft), ] #order the dataset
plot(br$sqft, br$price, col = "grey")
lines(exp(fitted(m4)) ~ ordat$sqft,
      col = "blue", main = "Log-linear Model")
```

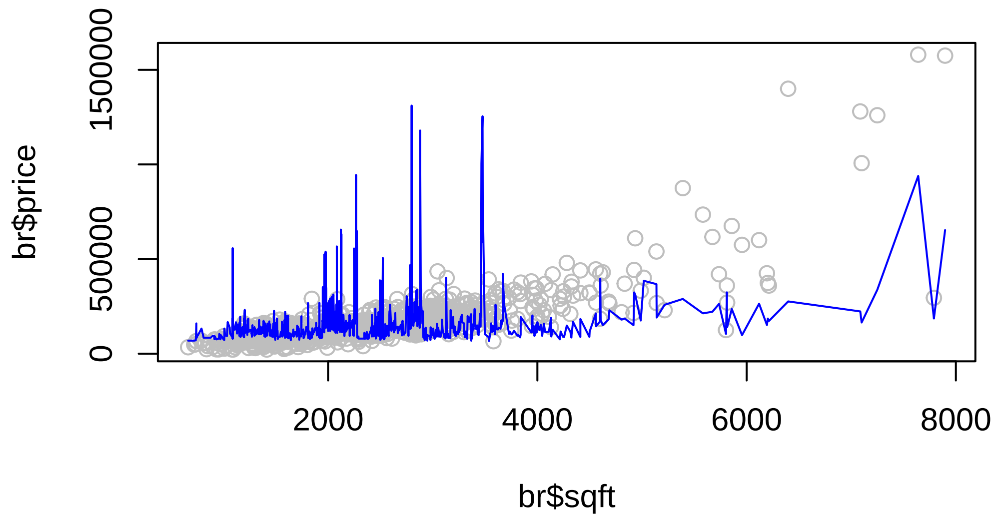

Figure 10: Fitting a quadratic model to the br dataset

```
b0 <- coef(m4)[[1]]
b1 <- coef(m4)[[2]]
#pick a few values for sqft:
sqftx <- c(2000, 3000, 4000)
#estimate prices for those and add one more:
pricex <- c(100000, exp(b0+b1*sqftx))
#re-calculate sqft for all prices:
sqftx <- (log(pricex)-b0)/b1
```

calculate and print elasticities:

```
(elasticities <- b1*sqftx) # the brackets makes sure it is printed
```

```
[1] 0.6743291 0.8225377 1.2338066 1.6450754
```

1.8 Using Indicator Variables in a Regression

An indicator, or binary variable marks the presence or the absence of some attribute of the observational unit, such as gender or race if the observational unit is an individual, or location if the observational unit is a house. In the data set utown, the variable utown is 1 if a house is close to the university and 0 otherwise. Here is a simple linear regression model that involves the variable `utown`:

$$\text{price}_i = \beta_0 + \beta_1 \text{utown}_i \tag{2}$$

The coefficient of such a variable in a simple linear model is equal to the difference between the average prices of the two categories; the intercept coefficient of the model is equal to the average price of the houses that are not close to university.

```
data("utown")
attach(utown)
price0bar <- mean(utown$price[which(utown$utown == 0)])
price1bar <- mean(utown$price[which(utown$utown == 1)])
prices.df <- data.frame(price0bar, price1bar);prices.df
```

```
  price0bar price1bar
1  215.7325  277.2416
```

The results are: $\overline{\text{price}} = 277.24$ close to university, and $\overline{\text{price}} = 215.73$ for those not close.

1.8.1 fitting a regression model

I now show that the same results yield the coefficients of the regression model

$$\text{price}_i = \beta_0 + \beta_1 \text{utown}_i$$

```
m5 <- lm(price ~ utown, data = utown)
b0 <- coef(m5)[[1]]
b1 <- coef(m5)[[2]]
```

The results are: $\overline{\text{price}} = \beta_1 = 215.73$ for non-university houses, and $\overline{\text{price}} = \beta_0 + \beta_1 = 277.24$ for university houses.

1.9 Monte Carlo Simulation