Data-based Statistical Decision Model

Lecture 2 - The case of simple linear regression

Sungkyu Jung

Today’s topic

- Simple Linear Regression Models, with various levels of assumptions

---

Simple Linear Regression Models

Let’s recall the simple linear regression model from last time. This is a statistical model with two variables X and Y, where we try to predict Y from X.

1. The distribution of X is unspecified, possibly even deterministic;
2. $Y|X = \beta_0 + \beta_1 x + \varepsilon$,where ε is a noise variable;
3. ε has mean 0, a constant variance σ^2,
4. ε is uncorrelated with X and uncorrelated across observations.

To elaborate, with multiple data points, $(X_1, Y_1), (X_2, Y_2), \dots (X_n, Y_n)$ then the model says that, for each $i \in 1:n$,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

where the noise variables ε_i all have the same expectation (0) and the same variance (σ^2), and $Cov[\varepsilon_i, \varepsilon_j] = 0$ (unless $i = j$, of course).

The optimal linear predictor

$$m(x) = E[Y] + Cov[X, Y]/Var[X](x - E(X))$$

equals the model formula $\beta_0 + \beta_1 x$, *if the model is true*.

Estimation

1. From the optimal linear predictor of Y

- The optimal linear predictor of Y from X has slope $\beta_1 = Cov[X, Y]/Var[X]$, and intercept $\beta_0 = E[Y] - \beta_1 E[X]$.
- Use what’s sometimes called the “plug-in principle”, where we
  - find equations for the parameters which would hold if we knew the full distribution, and
  - “plug in” the sample versions of the population quantities.

Thus

$$\hat{\beta}_1 = \frac{c_{XY}}{s_X^2},$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}.$$

What can we say about these estimates?

---

2. From the objective function ("MSE")

- The true optimal slope and intercept are the ones which minimize the mean squared error: for $MSE(b_0, b_1) = E(Y - (b_0 + b_1 X))^2$,

$$(\beta_0, \beta_1) = \arg\min_{(b_0, b_1)} MSE(b_0, b_1).$$

- Since we do not know the distribution, we can't get β_0 and β_1 from the data. Instead, minimize the **in-sample, empirical or training MSE**.

$$\widehat{MSE}(b_0, b_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (b_0 + b_1 x_i))^2.$$

- If our samples are all independent, for any fixed (b_0, b_1), the law of large numbers tells us that $MSE(b_0, b_1) \to MSE(b_0, b_1)$ as $n \to \infty$. So it doesn't seem unreasonable to try minimizing the in-sample error, which we can compute, as a proxy for minimizing the true MSE, which we can't. Where does it lead us?

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{(b_0, b_1)} \widehat{MSE}(b_0, b_1).$$

- Minimizing the in-sample MSE leads to what is often called "normal equations" or "estimating equations".

We have

- Normal equations

$$\bar{y} - \hat{\beta}_0 - \hat{\beta}_1 \bar{x} = 0$$

$$\bar{xy} - \hat{\beta}_0 \bar{x} - \hat{\beta}_1 \bar{x^2} = 0$$

- Closed-form solutions

$$\hat{\beta}_1 = \frac{c_{XY}}{s_X^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

They are the same as the "plug-in" estimates!

Assumptions?

We haven't used any of the model assumptions, yet.

Existence and uniqueness

- The least-squares estimates (of the linear regression line) always exist, as long as $s_X^2 \neq 0$.
- They are always unique, again unless $s_X^2 \neq 0$.
- Notice that this existence and uniqueness assumes absolutely nothing about the data-generating process.
    - not assuming that the simple linear regression model is correct.

How good these estimates are?

To quantify how good $\hat{\beta}_1$ and $\hat{\beta}_0$ are, we need to know to what we are comparing the random quantities.

- We may simply compare those with the coefficients of the "optimal linear predictor", $\beta_1 = Cov[X,Y]/Var[X]$, and $\beta 0 = E[Y] - \beta_1 E[X]$.
- The comparison would be more meaningful, if we assume that the model was true.
- We will see
    - bias of $\hat{\beta}_1$ (in estimation of β_1)
    - variance (or standard error) of $\hat{\beta}_1$
    - consistency of $\hat{\beta}_1$
- We would very much like to know the sampling distribution of $\hat{\beta}_1$, but it is not possible, yet.

Constant plus noise representation

$$\hat{\beta}_1 = \beta_1 + \frac{1}{n}\sum_{i=1}^{n}\frac{(x_i - \bar{x})}{s_X^2}\epsilon_i,$$

$$\hat{\beta}_0 = \beta_0 + \frac{1}{n}\sum_{i=1}^{n}(1 + \frac{(x_i - \bar{x})}{s_X^2})\epsilon_i.$$

Note that the "noise" term is a weighted sum of uncorrelated noises.

Also note that we have used a model assumption here (which one?).

Bias and variance of the estimators

- The least-squared estimates are unbiased.

$$E(\hat{\beta}_0) = \beta_0$$

$$E(\hat{\beta}_1) = \beta_1$$

- Variances reduce like $1/n$.

$$Var(\hat{\beta}_0) = \frac{\sigma^2}{n}(1 + \bar{x}^2/s_X^2)$$

$$Var(\hat{\beta}_1) = \frac{\sigma^2}{n s_X^2}$$

- They are also consistent. Why?

Parameter interpretation

- σ^2: the variance of the noise around the regression line
- β_0: simply the expected value of Y when X is 0, $E[Y|X=0]$. This ensures that the line goes through the point $(E[X], E[Y])$.

- β_1: Mathematically, $\beta_1 = E[Y|X=x] - E[Y|X=x-1]$, or

$$\beta_1 = \frac{E[Y|X=x_2] - E[Y|X=x_1]}{x_2 - x_1}.$$

(Again, this is true only if the linear form of assumption is true.)

Note that this is *not causation*. Which statement may be wrong?

1. If we change X by 1, then on average Y will change by β_1.
2. If we select two sets of cases from the un-manipulated distribution where X differs by 1, we expect Y to differ by β_1.

The right way to interpret β_1 is not as the result of a change, but as an expected difference.

Predictions

We got into all this mess not because we want to know the numbers β_0 and β_1 for their own sake, but because we wanted to predict Y from X. How do we make those predictions, and how good are they?

- If we knew β_0 and β_1, and that $X = x$, then our prediction for Y would be $\beta_0 + \beta_1 x$. (In what sense?)
- Since we do not know β_0 and β_1, replace them with our best guess $\hat{\beta}_0$ and $\hat{\beta}_1$. This point prediction is the fitted value of Y at $X = x$:

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}_1 x.$$

How good the fitted values are?

Notice the fitted value $\hat{m}(x)$ is an estimate of $E[Y|X=x]$. The latter is $\beta_0 + \beta_1 x$ (if the assumptions are true), a perfectly deterministic quantity; and we do not know it!

- $\hat{m}(x)$ is random.

Using the model assumptions, by writing

$$\hat{m}(x) = \beta_0 + \beta_1 x + \frac{1}{n}\sum_{i=1}^{n}\left(1 + (x - \bar{x})\frac{x_i - \bar{x}}{s_X^2}\right)\varepsilon_i,$$

we can check

---

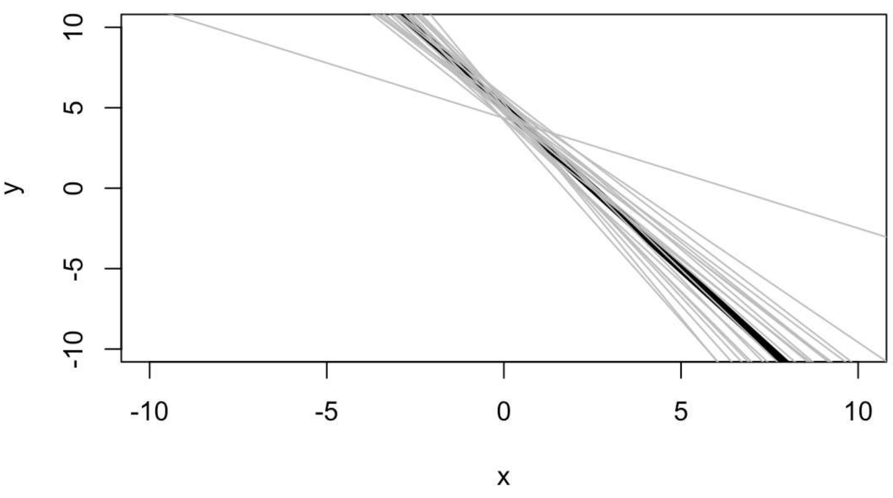
1. $\hat{m}(x)$ is unbiased; $E(\hat{m}(x)) = m(x)$.

2. $\hat{m}(x)$ has variance $Var(\hat{m}(x)) = \frac{\sigma^2}{n}(1 + \frac{(x-\bar{x})^2}{s_X^2})$.

Note:

1. The variance grows as σ^2 grows.
2. The larger n is, the smaller the variance.
3. The variance of our predictions is the sum of two terms:
    - variation of the height of $\bar{y}$,
    - variation around the slope.

Scatter of estimated least-squares regression lines (thin, grey) around the true regression line (thick, black). Notice how the estimated lines become more spread out as we move away from the center of the distribution.



Estimating σ^2

From

$$E(Y - (\beta_0 + \beta_1 X))^2 = \sigma^2 \quad = \text{MSE}(\beta_0, \beta_1),$$

we find the minimal value of the “in-sample MSE”

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x))^2 := \hat{\sigma}^2 \quad = \text{in-sample MSE}(\hat{\beta}_0, \hat{\beta}_1).$$

- $\hat{\sigma}^2$ is consistently estimating σ^2, but is slightly biased;

$$s^2 = \frac{n}{n-2}\hat{\sigma}^2,$$

is an unbiased estimator of σ^2. (This is also sometimes called the *MSE*.)

Residuals

The residual value at a data point is the difference between the actual value of the response y_i and the fitted value $\hat{m}(x_i)$:

$$e_i = y_i - \hat{m}(x_i) = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x).$$

- Compare this with the noise at x_i:

$$\epsilon_i = y_i - (\beta_0 + \beta_1 x_i)$$

- They are not the same; $e_i \neq \epsilon_i$.
- From the model assumptions, we have

$$E(\epsilon_i) = 0,\ Cov(\epsilon_i, X_i) = 0.$$

- Likewise, but not due to any assumptions,

$$\sum_{i=1}^{n} e_i = 0,\ \sum_{i=1}^{n} e_i(x_i - \bar{x}) = 0.$$

Limitations of Least Squares

- The results so far, almost exhaust the theory of statistical inference for least squares estimates in the simple linear regression model. We used just least squares and the simple linear regression model.
- However, we can’t get the sampling distribution of, say, $\hat{\beta}_1$.
- We need sampling distributions to form confidence intervals, evaluate the properties of hypothesis tests.
- If we add to the simple linear regression model the assumption that the ε_i are IID draws from a fixed, not-necessarily-Gaussian distribution, we might then try to use the central limit theorem to show that the weighted average tends towards a Gaussian. (n needs to be truly huge in some cases.)

R demonstration

The basic command for fitting a linear model by least squares in R is `lm`.

```
lm(y ~ x, data=df)
```

Here `df` is a data frame containing the data we want to fit a regression to, and the first part, the formula, tells `lm` that we want to regress the column of `df` called `y` on the column called `x`.

To play around, we play God, and simulate data, using the following function:

```
sim.linmod <- function(n, beta.0, beta.1, width, df) {
    # draw n points from a uniform distribution centered on 0
    x <- runif(n, min=-width/2, max=width/2)
    # draw n points from a t distribution with the given number of degrees
    # of freedom
    epsilon <- rt(n, df=df)
    # make y from a linear model
    y <- beta.0 + beta.1*x + epsilon
    # return the data frame
    return(data.frame(x=x, y=y))
}
```

What `lm` returns is a rather complicated object. If you just print it out, it seems to be only the intercept and the slope:

```
 # Make a very small simulated data set from our running examing
toy.data <- sim.linmod(n=10, beta.0=5, beta.1=-2, width=4, df=3)
# Fit the simple linear regression model to it by least squares
lm(y~x, data=toy.data)
```

```
##
## Call:
## lm(formula = y ~ x, data = toy.data)
##
## Coefficients:
## (Intercept)            x
##       4.520       -2.424
```

In fact, `lm` has done lots of calculations as part of fitting the model, and stored many of the results into the object it returns; R just doesn't print all of that, unless you make it.

```
names(lm(y~x, data=toy.data))
```

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

By saving the result in an R object, here called `toy.lm`, we can call it many times without re-computing it.

```
toy.lm <- lm(y~x, data=toy.data)
```

Now obtain the following:

- `coefficients(toy.lm)` : returns $\hat{\beta}_0$ and $\hat{\beta}_1$,
- `fitted(toy.lm)` : returns $\hat{m}(x_i)$, the fitted value of Y at $X = x_i$
- `residuals(toy.lm)` : returns e_i
- `predict(toy.lm, newdata)` : returns $\hat{m}(x)$ at points x in `newdata`

```
coefficients(toy.lm)
```

```
## (Intercept)           x
##    4.520476   -2.423980
```

```
fitted(toy.lm)
```

```
##          1          2          3          4          5          6
##  5.7851548  7.7132587  7.4649890  1.9810525  2.0863970  8.3186042
##          7          8          9         10
##  0.3596059  8.6439486 -0.3061479  5.6856641
```

```
residuals(toy.lm)
```

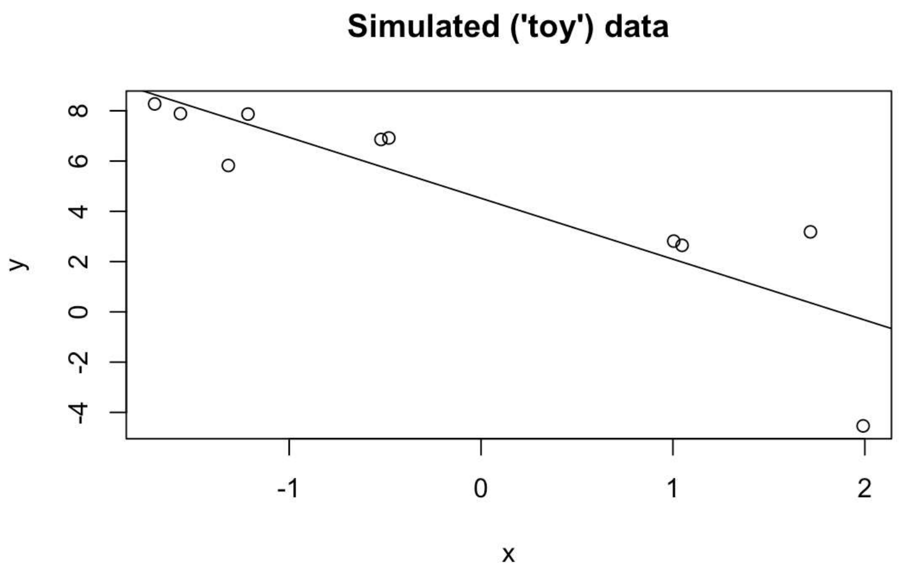
```
##          1          2          3          4          5          6
##  1.0733434 -1.8898530  0.4034083  0.6644482  0.7274961 -0.4315735
##          7          8          9         10
##  2.8230043 -0.3702680 -4.2315077  1.2315018
```

```
predict(toy.lm, newdata=data.frame(x=1:5))
```

```
##          1          2          3          4          5
##  2.0964956 -0.3274847 -2.7514651 -5.1754455 -7.5994258
```

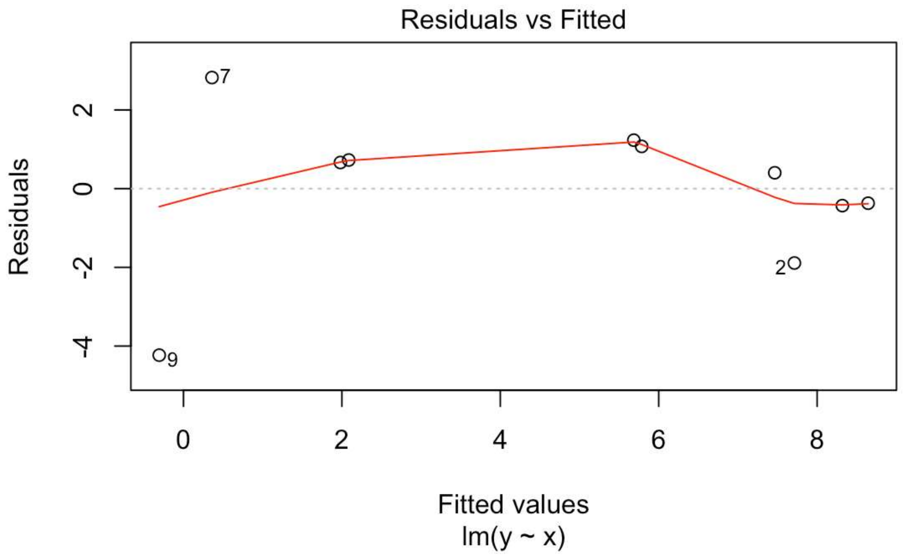
We may want to “see” how good the fit is.

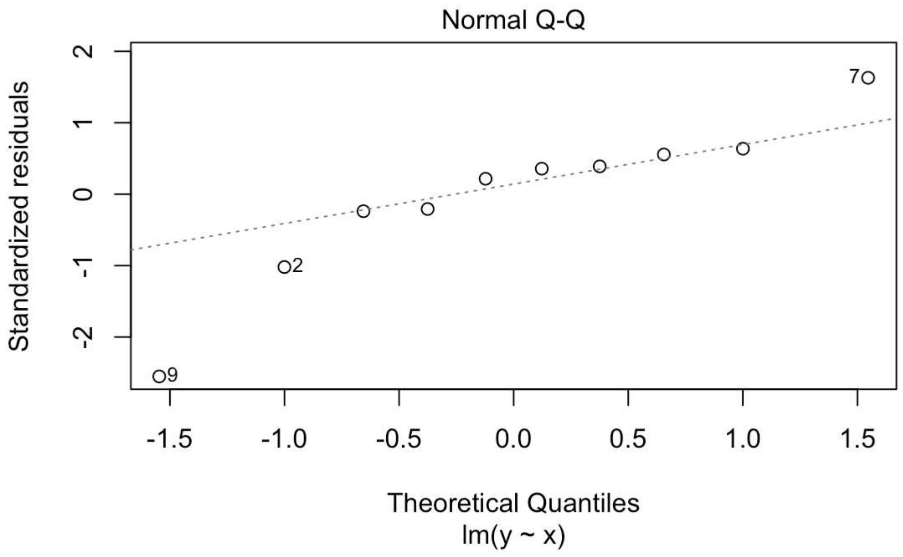
```
plot(y~x, data=toy.data, xlab="x",ylab="y", main="Simulated ('toy') data")
abline(toy.lm)
```

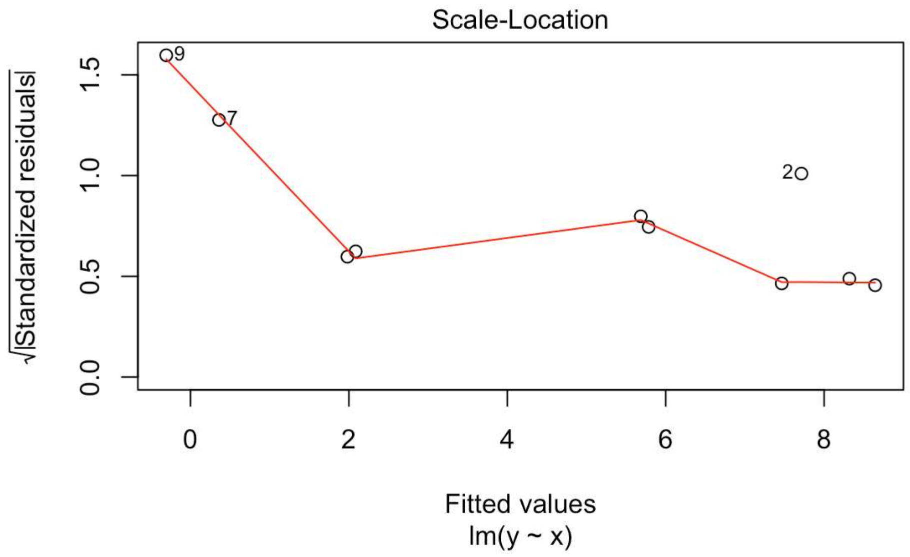


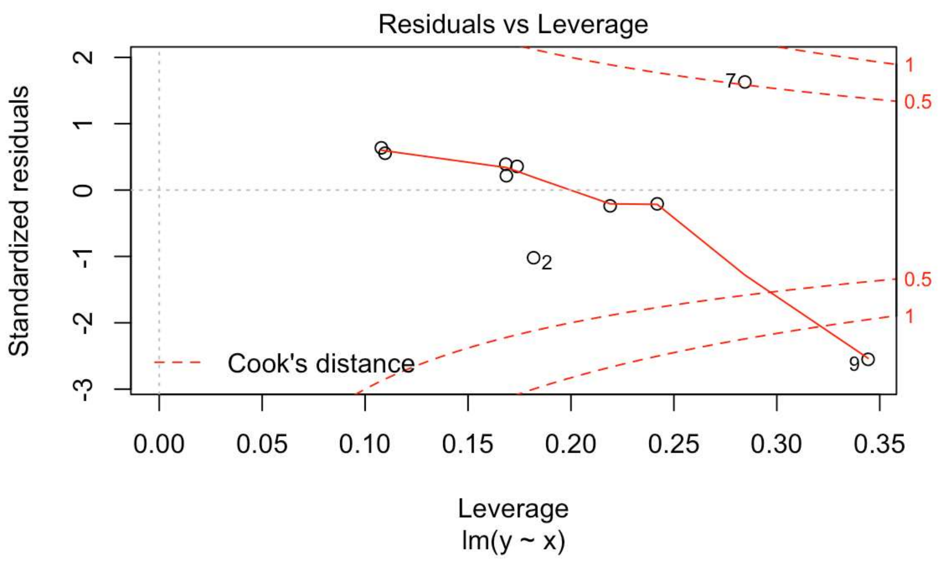
`plot` function gives a series diagnostic plots. We will return to them later.

```
plot(toy.lm)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x)
7
9
2
0
2
4
6
8
-4
-2


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(y ~ x)
7
2
9
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5


Scale-Location
√|Standardized residuals|
Fitted values
lm(y ~ x)
9
7
2
0.0
0.5
1.0
1.5

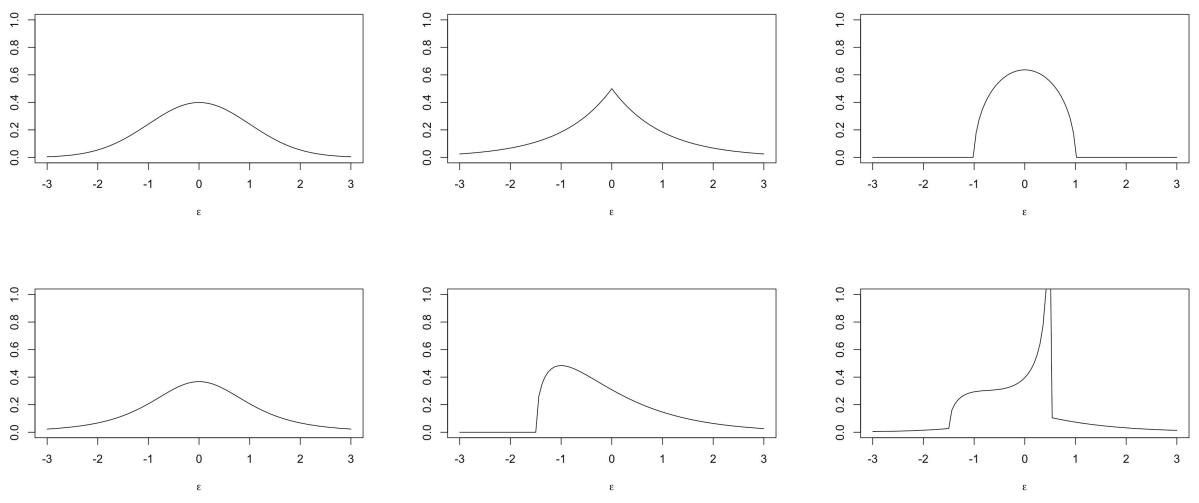


The Gaussian-Noise Simple Linear Regression Model

We have, so far, assumed comparatively little about the noise term ε.

- The advantage of this is that our conclusions apply to lots of different situations;
- the drawback is there's not much to say about our estimator or our predictions
- To be more precise on inference, we may add a distributional assumptions on ϵ:

Some possible noise distributions for the simple linear regression model, since all have $E[\varepsilon] = 0$, and could get any variance by scaling. Why Gaussian?



Why Gaussian noise?

1. Central limit theorem
2. Mathematical convenience

The Gaussian-Noise Simple Linear Regression Model

1. The distribution of X is unspecified, possibly even deterministic;
2. $Y|X = \beta_0 + \beta_1 x + \varepsilon$,where ε is a noise variable;
3. $\varepsilon \sim N(0, \sigma^2)$
4. ε is uncorrelated with X and uncorrelated across observations.

Estimation by maximizing the likelihood

Now with the Gaussian assumption, we have

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \;\; \epsilon_i \sim \; i.i.d. \; N(0, \sigma^2).$$

- The MLE (maximum likelihood estimate) is the parameter $(\beta_0, \beta_1, \sigma^2)$ that maximizes the likelyhood, a measure of how likely the data came from.
- The MLEs of $\beta_0, \beta_1, \sigma^2$ are exactly those of Least-squares estimates.
- MLEs are in general “most efficient” (in the sense that they have the smallest variance among all other estimators’ variances).

Advantages of the Gaussian model

1. Complete visualization of the model.
2. Sampling distributions of $\hat{\beta}_1, \hat{\beta}_0, \hat{m}(x)$.