Modeling concepts; least squares

PSTAT100 Summer 2023

This week: the simple linear model

- **Statistical models**
  - What makes a model ‘statistical’?
  - Goals of modeling: prediction, description, and inference
  - When to avoid models
- **The simple linear regression model**
  - Line of best fit by least squares
  - A model for the error distribution
  - The simple linear model
  - Interpretation of estimates
  - Uncertainty quantification

What makes a model?

> A model is an idealized representation of a system. You likely use models every day. For instance, a weather forecast is [based on] a model.

In the context of quantitative methods a model is typically a mathematical representation of some system.
Quick discussion:

- what are some examples of models you've encountered?
- in what sense are they models?
- are they statistical models?

What makes a model ‘statistical’?

One straightforward view is that a **statistical model** is simply a ***probabilistic representation of a data-generating process***. In other words, a probability distribution.
For a probability distribution to provide a sensible description of a data generating process:

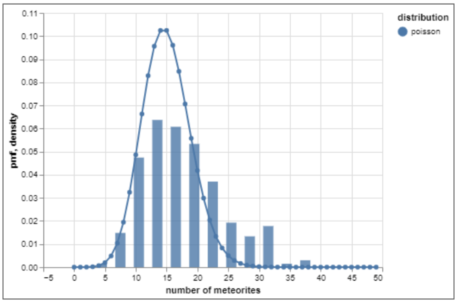
- one needs to be able to at least imagine collecting multiple datasets with the same basic structure
- observations must be subject to randomness in some sense

That’s why sampling is so important to statisticians. Probabilistic sampling ensures:

- the process by which data are collected is fixed and repeatable
- the method of selection and measurement of observational units induces randomness in the data

Univariate models

Suppose you have a dataset comprising the number of meteorites that hit earth on each of 225 days. A very simple model is that the counts are independent Poisson random variables.

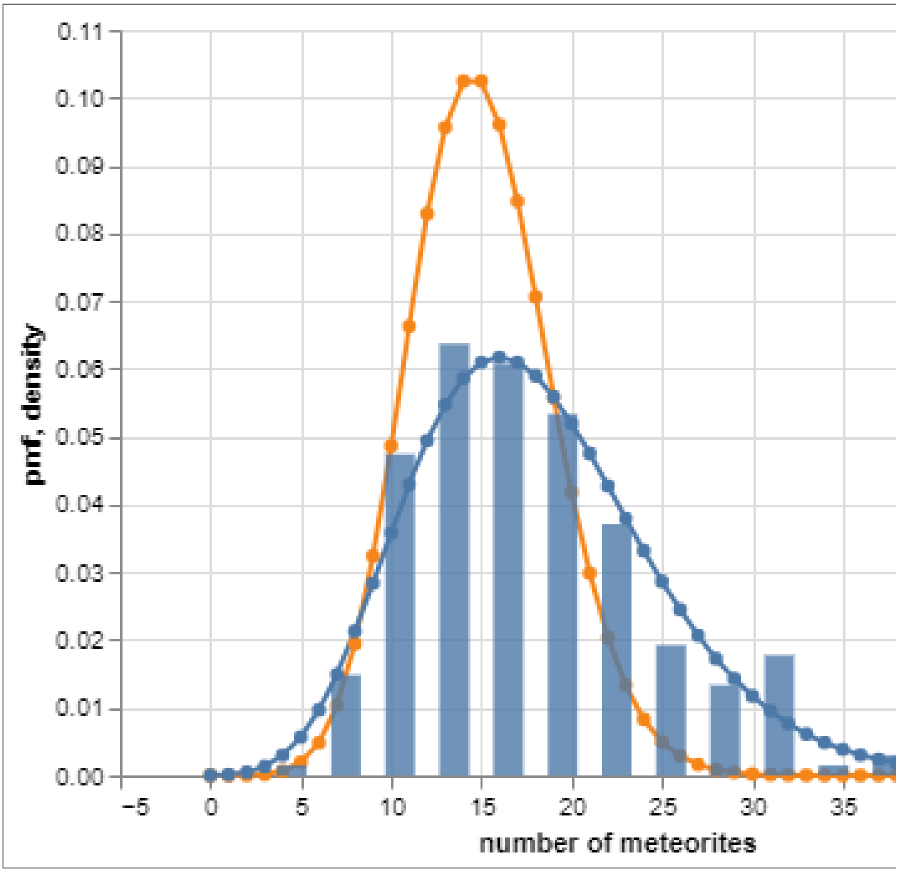


- the model is parametric, with a single parameter λ, given by the Poisson distribution: $f(x; \lambda) = \lambda^x e^{-\lambda}/x!$
- since $\mathbb{E}X = \lambda$ and observations are assumed independent and identically distributed, the

model is ‘fitted’ by
estimating $\lambda = \bar{x}$

Simple example

However, the negative binomial distribution provides a better ‘fit’:



- perhaps not surprising, considering it has *two* parameters

Why model the data-generating process?

A good description of a data-generating process usually captures two aspects of a system:

1. the deterministic aspects, allowing one to identify structure in the data; and
2. the random aspects or 'noise', allowing one to quantify uncertainty.

In our toy example, the better model captured both the most common value (a kind of structure) and the variation (noise).

Modeling goals

Models serve one of three main purposes:

- **Prediction**: predict new data before it is observed.
- **Inference**: make conclusions about a larger population than the observed data.
- **Description**: less common, but sometimes models provide a convenient description of observed data.

We probably wouldn't use a univariate model to make specific predictions, but we could for instance estimate the probability that over 40 meteorites (rarely observed) hit earth any given day.

Models you’ve seen already

The exploratory analysis techniques you’ve seen are actually very flexible models often used for descriptive purposes.

- Kernel density estimates are models for univariate data
- LOESS curves are models for trends in bivariate data
- Principal components are models for correlation structures

It’s a little tricky to see, but these correspond to *classes* of probability distributions rather than specific ones. That’s why they’re so flexible.

It's okay not to model data

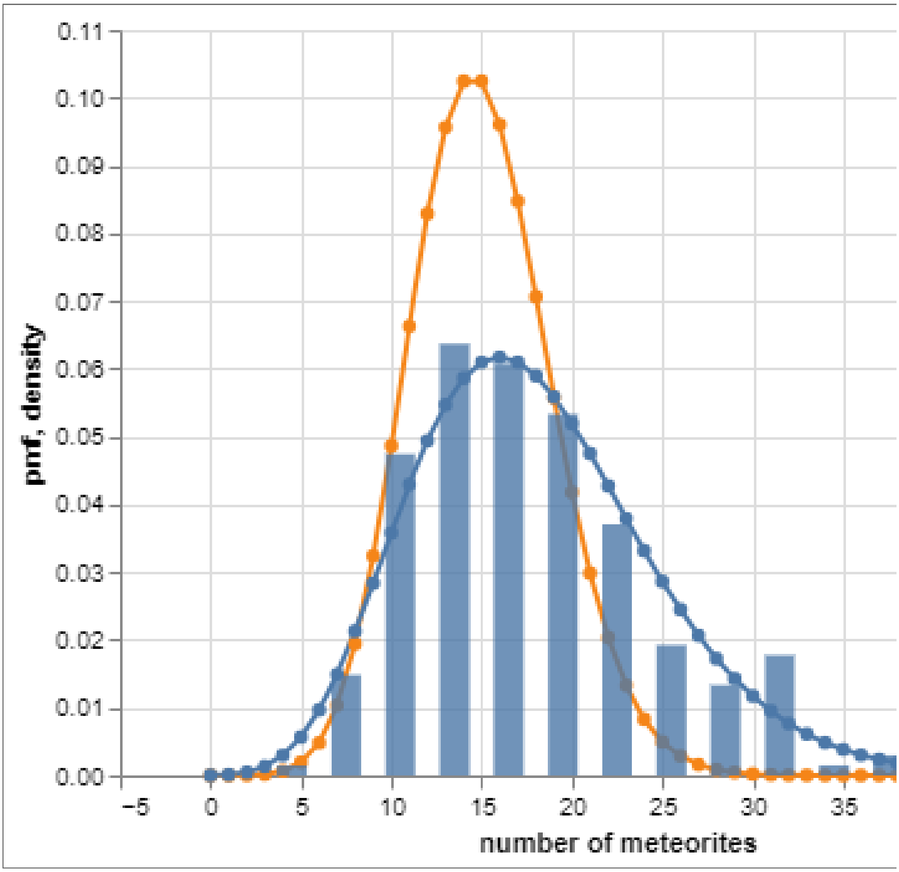
There are a lot of situations when modeling simply isn't appropriate or feasible.
Sketchy sampling: every statistical model makes some assumptions about the data collection process. These don't always need to hold exactly, but models could be untenable if:

- the way data were collected is highly opaque or nonsystematic
- the sampling design or measurement procedures have serious flaws or inconsistencies

Sparse data: model fitting is sensitive to the specific dataset observed, and may be unreliable if it's too sensitive. This commonly arises when:

- the model has a lot of parameters
- the dataset contains few observations (relative to the number of parameters)

Comment



These univariate models usually don’t occur to us as models in the fullest sense, because:

- No deterministic structure
- All variation is random

For this reason you can’t do much with them, so they seem a bit uninteresting.

Constructing more interesting models

Many models *also* involve an interesting deterministic component.
A frequent strategy is to form a *regression* model:

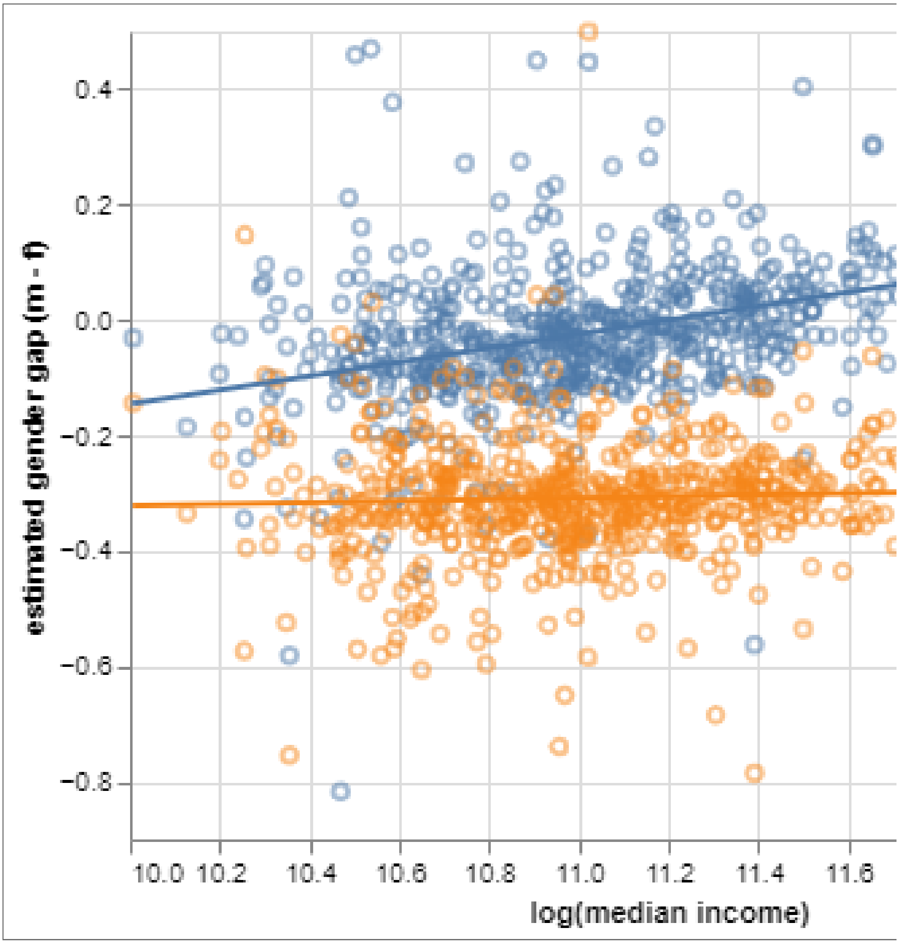
- assume a distributional form for a quantity of interest
- 'regress' the mean or other parameters of the distribution 'on' other variables – *i.e.,* express the former as a function of the latter

For instance:

- $\#\text{meteors each day} \sim \text{poisson}(\lambda)$
- $\lambda = g(\text{month})$

Linear models

A linear regression model is one in which the deterministic component is linear: the mean of a variable of interest is a linear function of one or more other variables.
For instance:



You may not have realized it at the time, but those lines are *simple linear regression models*: they describe the mean gaps as linear functions of log median income.

Remarks on terminology

- A linear regression model is *simple* if it involves only one variable of interest Y and one additional variable X
  - Y is the ‘response’, ‘dependent’, or ‘endogenous’ variable
  - X is the ‘explanatory’, ‘independent’, or ‘exogenous’ variable
- The *multiple* linear regression model involves multiple explanatory variables $X_1, X_2, \dots$
- *Multivariate* linear regression models involve multiple responses $Y_1, Y_2, \dots$
- “Linear regression model” is usually taken to mean a model in which Y is assumed to be *normal*, given X

Applications of linear models

Linear models can be used for prediction, inference, or both.

- Predict the gender achievement gaps for a new district based on median income in the district.
- Quantify the association between median income and achievement gaps.

We’re going to talk in detail about the model itself:

- definition
- estimation
- assumptions

Data setting

Let's first introduce the kind of data that the simple linear model describes.

- There are two variables, X and Y.
- The data values are n observations of these two variables:
  $(x_1, y_1), \ldots, (x_n, y_n)$

  The notation in tuples indicates the pairing of the values when measured on the same observational unit. If it helps, think of them as rows of an $n \times 2$ dataframe:

X	Y
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_n	y_n

Data setting

The notation above is just a mathematical description of data that looks like this:

In our notation, X would represent log median income, and Y would represent the math gap.

Data setting

The example SEDA data in tabular form are:

▶ Code

	log_income	gap
id		
600001	11.392048	-0.562855
600006	11.607236	0.061163
600011	10.704570	-0.015417

The tuples would be:

$$(\text{log\_income}_1, \text{gap}_1)\ ,\ (\text{log\_income}_2, \text{gap}_2)\ ,\ \dots\ ,\ (\text{log\_ir}$$

Or more specifically:

$$(11.392, -0.563)\ ,\ (11.607, 0.061)\ ,\ \dots\ ,\ (11.229, -0.04$$

Lines and data

A line in slope-intercept form is given by the equation:

$$y = ax + b$$

Data values never fall exactly on a line. So in general for every a, b:

$$y_i \neq ax_i + b$$

But we can describe any dataset as a line and a 'residual':

$$y_i = \underbrace{ax_i + b}_{\text{line}} + \underbrace{e_i}_{\text{residual}}$$

Lines and data

Here’s a picture:

Each *residual* is simply the vertical distance of a value of Y from the line:
$e_i = {\color{blue}y_i} - {\color{red}(ax_i + b)}$

Many possible lines

This makes it possible to express Y as a linear function of X.
However, the mathematical description is somewhat tautological, since for *any* a, b, there are residuals $e_1, \ldots, e_n$ such that
$y_i = ax_i + b + e_i$
In other words, there are **infinitely many possible lines**. So which values of a and b should be chosen for a given set of data values?

The least squares line

A sensible criterion is to find the line for which:

- the average residual $\bar{e}$ is zero; and
- the residuals vary the least.

If $\bar{e} = 0$, then the residual variance is proportional to the *sum of squared residuals*:

$$\frac{1}{n-1}\sum_{i=1}^{n}(e_i - \bar{e})^2 = \frac{1}{n-1}\sum_{i=1}^{n} e_i^2$$

So the values of a, b that minimize $\sum_i e_i$ give the 'best' line (in one sense of the word 'best'). This method is known as **least squares**.

$$(a^*, b^*) = \arg\min_{(a,b)} \left\{ \sum_{i=1}^{n} \underbrace{(y_i - (ax_i + b))^2}_{e_i^2} \right\}$$

Deriving the least squares line

Let’s try doing the derivation using univariate calculus.

Note: $e_i = y_i - a - bx_i$.

$$\frac{d}{da}\sum_i e_i^2 = \cdots$$

$$\frac{d}{db}\sum_i e_i^2 = \cdots$$

If $\frac{d}{da}\sum_i e_i^2 = 0$ and $\frac{d}{db}\sum_i e_i^2 = 0$ then…

Deriving the least squares line

Alternatively, the model can be written in matrix form as $\mathbf{y} = \mathbf{X}\mathbf{a} + \mathbf{e}$, where:

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{a}} + \underbrace{\begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix}}_{\mathbf{e}}$$

Then, the sum of squared residuals is:

$$\mathbf{e}'\mathbf{e} = (\mathbf{y} - \mathbf{X}\mathbf{a})'(\mathbf{y} - \mathbf{X}\mathbf{a})$$

Using vector calculus, one can show that:

$$\nabla_{\mathbf{a}}\mathbf{e}'\mathbf{e} = 0 \quad \Longrightarrow \quad 2\mathbf{X}'\mathbf{y} = 2\mathbf{X}'\mathbf{X}\mathbf{a} \quad \Longrightarrow \quad \mathbf{a} = (\mathbf{X}'\mathbf{X})$$

And that this is a minimum.

Calculating the least squares line

The solution $\mathbf{a} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ is known as the **ordinarly least squares** (OLS) estimate.

The `sklearn` implementation looks like this:

```python
# save explanatory variable and response variable separately as ar
x = regdata.log_income.values[:, np.newaxis]
y = regdata.gap.values

# configure regression module and fit model
slr = LinearRegression()
slr.fit(x, y)

# store estimates
print('slope: ', slr.coef_[0])
print('intercept: ', slr.intercept_)
```

```
slope:  0.12105696076155241
intercept:  -1.3561699570330028
```

Calculating the least squares line

The solution $\mathbf{a} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ is known as the **ordinarly least squares** (OLS) estimate.

The `statsmodels` implementation looks like this:

```
# add a column of ones (intercept)
x_aug = sm.tools.add_constant(x)

# fit model
slr = sm.OLS(endog = y, exog = x_aug)
slr_fit = slr.fit()

# return estimates
print('estimates: ', slr_fit.params)
```

```
estimates:  [-1.35616996  0.12105696]
```

Note a constant has to be added to `x`. Why?

Calculating the least squares line

Alternatively, `statsmodels` also has a wrapper around `sm.OLS` that allows models to be fit based on a dataframe and column names (much like `lm()` in R):

```python
import statsmodels.formula.api as smf

# fit model based on dataframe and formula
slr = smf.ols(formula = 'gap ~ log_income', data = regdata)
slr_fit = slr.fit()

# return estimates
slr_fit.params
```

```
Intercept    -1.356170
log_income    0.121057
dtype: float64
```

Calculating the least squares line

We can check the calculations by computing the closed-form expression:

```
# ols solution, by hand
x_mx = np.vstack([np.repeat(1, len(x)), x[:, 0]]).transpose() # X
xtx = x_mx.transpose().dot(x_mx) # X'X
xtx_inv = np.linalg.inv(xtx) # (X'X)^{-1}
xtx_inv.dot(x_mx.transpose()).dot(y) # (X'X)^{-1} X'y
```

```
array([-1.35616996,  0.12105696])
```

Error

×