ANLY-511 Case Study Group 9

Optimizing Study Time

Sonali Pednekar (ssp88), Rui Qiu (rq47), Zhaoyuan Qiu (zq37)

Georgetown University
Oct 22, 2021 (updated: 2021-10-21)

Setup

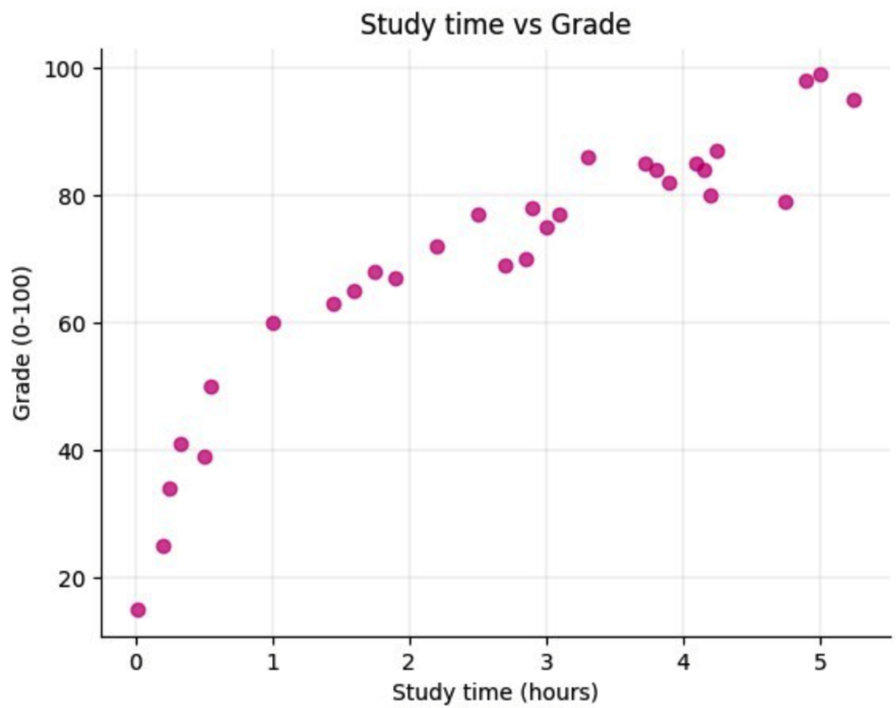
Introduction

What is a Maximum Likelihood Estimation?

- Maximum likelihood estimation (MLE) is a method of estimating the parameters of an assumed probability distribution, given some observed data.
- If the likelihood function is differentiable, the derivative test for determining maxima can be applied.
- The ordinary least squares estimator maximizes the likelihood of the linear regression model.

Introduction

time (h)	grade (0-100)
0,01	15
0,2	25
0,25	34
0,33	41
0,5	39
0,55	50
1	60
1,45	63
1,6	65
1,75	68
1,9	67
2,2	72
2,5	77
2,7	69
2,85	70



Beautiful dummy data 😁

Linear Model (SLR)

SLR

This is the model that best describes the problem at hand

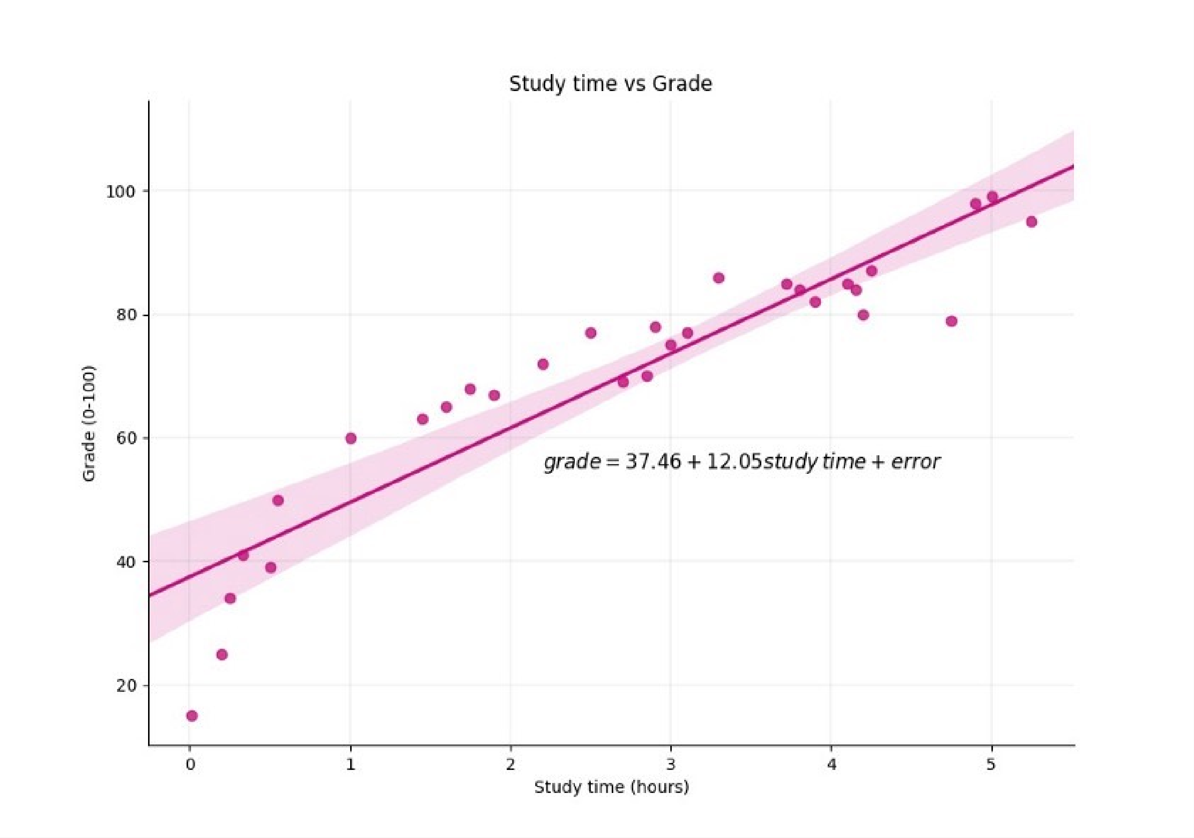
$$\underset{\text{independent}}{grade} = \underbrace{\beta_0 + \beta_1}_{\text{parameters}} \underset{\text{predictor}}{study\, time} + error$$

In this equation,

- Grade is an independent variable
- Predictor is the study time.
- The parameters, beta0 and beta1 are the coefficients

SLR

Fit a linear model to the dataset on any statistical software.



SLR

OLS Regression Results

Dep. Variable:	grade	R-squared:	0.856
Model:	OLS	Adj. R-squared:	0.851
Method:	Least Squares	F-statistic:	166.2
Date:	Sun, 30 Dec 2018	Prob (F-statistic):	2.70e-13
Time:	18:04:49	Log-Likelihood:	-104.46
No. Observations:	30	AIC:	212.9
Df Residuals:	28	BIC:	215.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	37.4571	2.905	12.892	0.000	31.506	43.409
time	12.0495	0.935	12.892	0.000	10.135	13.964

Omnibus:	6.882	Durbin-Watson:	0.653
Prob(Omnibus):	0.032	Jarque-Bera (JB):	5.216
Skew:	-0.955	Prob(JB):	0.0737
Kurtosis:	3.722	Cond. No.	6.55

Least Squares method was used to fit the model.

Parameter Estimation

Parameter Estimation

Recall Likelihood Function

Given output y, the likelihood function of the parameter θ is defined as,

$$L(\theta; y) = P(Y = y|\theta)$$

Simple Linear Regression

For SLR model,

$$y = \beta_0 + \beta_1 x + \sigma$$

Four hypotheses are raised,

1. Data points are mutually independent.
2. Dataset follows a normal distribution.
3. The error term σ follows a normal distribution whose mean equals to 0.
4. The output y is continuous.

Parameter Estimation

Likelihood Function in SLR

Given the two parameters β_0, β_1, input variable $X = x_i$ and error term σ, the likelihood function is built as,

$$L(\beta_0, \beta_1, \sigma; y) = P(Y = y|X = x_i; \beta_0, \beta_1, \sigma)$$

Conditional Density of $y|x$

In order to find the β_0, β_1, which maximize $L(\beta_0, \beta_1, \sigma; y)$, let's take a look at the density function of y.

Parameter Estimation

According to the hypothesis 1, 2 and 4, y should be continuously and normally distributed.

Therefore,

$$f(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu_y)^2}{2\sigma^2}}$$

Here $\mu_y = \mu_{\beta_0} + \mu_{\beta_1 x} + \mu_\sigma = \beta_0 + \beta_1 \mu_x + \mu_\sigma$

According to the hypothesis 3, $\mu_\sigma = 0$. Therefore, $\mu_y = \beta_0 + \beta_1 \mu_x$.

$$f(y|x_i, \beta_0, \beta_1, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-(\beta_0+\beta_1 x_i))^2}{2\sigma^2}}$$

Parameter Estimation

Probability of Observations

According to the hypothesis 1, the probability of all observed points are independent. The overall probability equals to the products of every point.

$$\prod_{i=1}^{n} P(y|X = x_i; \beta_0, \beta_1, \sigma) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i-(\beta_0+\beta_1 x_i))^2}{2\sigma^2}}$$
$$= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n e^{-\frac{\sum_{i=1}^{n}(y_i-(\beta_0+\beta_1 x_i))^2}{2\sigma^2}}$$

To maximize $\prod_{i=1}^{n} P(y|X = x_i; \beta_0, \beta_1, \sigma)$ equals to minimize $\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2$, which is exactly the **sum of square error**.

Therefore, the likelihood function can be updated as

$$L(\beta_0, \beta_1, x_i; y) = -\sum_{i=1}^{n}(y_i - (\beta_0 + \beta_1 x_i))^2 = -SSE$$

Estimate β_0

In order to find β_0 and β_1, which maximize $L(\beta_0, \beta_1, x_i; y)$, partial derivative methods will be used.

$$\frac{\partial L}{\partial \beta_0} = 2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\downarrow$$

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \beta_1 x_i}{n} = \overline{y} - \beta_1 \overline{x}$$

Parameter Estimation

Estimate β_1

$$\frac{\partial L}{\partial \beta_1} = 2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)x_i = 0$$

$$\downarrow$$

$$\begin{aligned}
\sum_{i=1}^{n}(x_i y_i - \beta_0 x_i - \beta_1 x_i^2) &= \sum_{i=1}^{n}(x_i y_i - (\bar{y} - \beta_1 \bar{x})x_i - \beta_1 x_i^2) \\
&= \sum_{i=1}^{n} x_i y_i - \bar{y}\sum_{i=1}^{n} x_i + \beta_1(\bar{x}\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i^2) \\
&= 0
\end{aligned}$$

$$\downarrow$$

$$\hat{\beta}_1 = \frac{\bar{y}\sum_{i=1}^{n}\frac{x_i}{n} - \sum_{i=1}^{n}\frac{x_i y_i}{n}}{\bar{x}\sum_{i=1}^{n}\frac{x_i}{n} - \sum_{i=1}^{n}\frac{x_i^2}{n}} = \frac{\bar{x}\bar{y} - \overline{xy}}{\bar{x}^2 - \overline{x^2}} = \frac{\mathrm{Cov}(x, y)}{\mathrm{Var}(x)}$$

Experiment with Similar Data

Housing Prices (1)

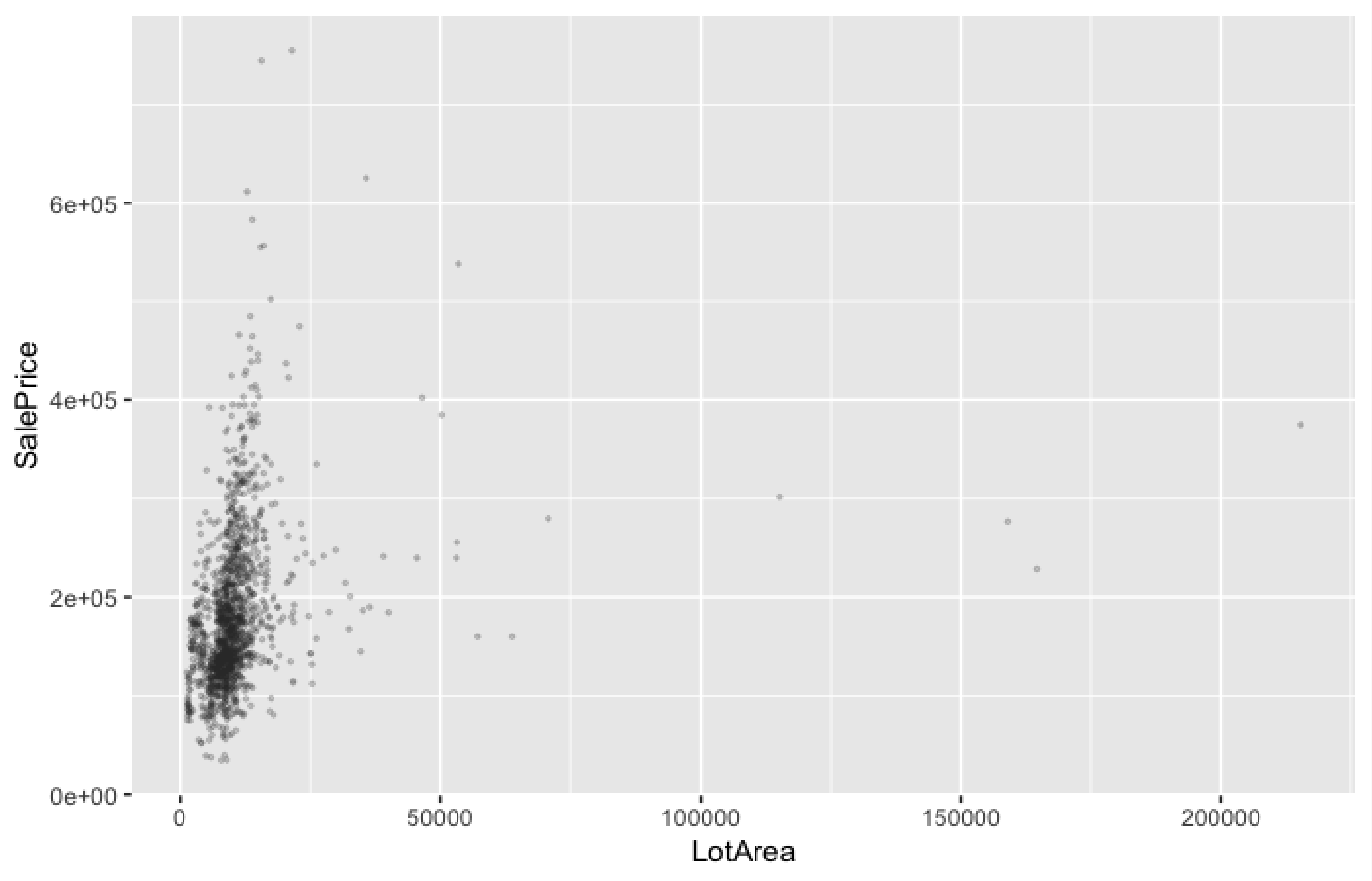
- Let's just jump into a Kaggle data

> data source: https://www.kaggle.com/c/home-data-for-ml-course

- we extract the train.csv and only pay attention to one variable `LotArea` and the response `SalePrice`

Housing Prices (2)

```r
dat <- read_csv("train.csv") %>% select(LotArea, SalePrice)
ggplot(dat, aes(x = LotArea, y = SalePrice)) +
  geom_point(alpha = .2, size = .5, colour = "grey20")
```

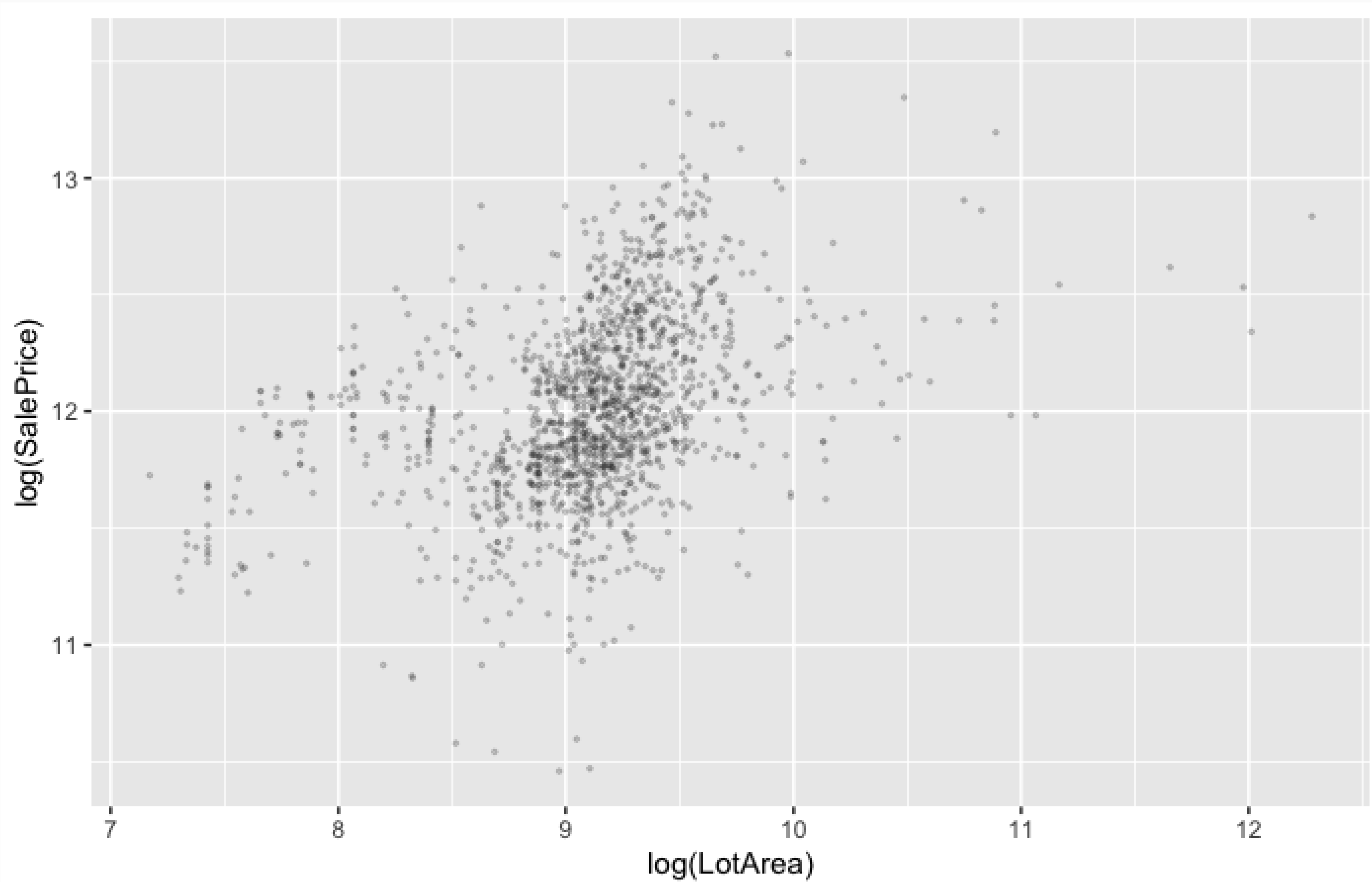


Housing Prices (3)

- not so good, try do some transformation(s) (not required though)
- but always do this, cuz the more "normal" our variables are, the "normal" fit the model will be (it's also one of the assumptions of a linear model and you can always use q-q plot to check that).

Housing Prices (4)

```r
ggplot(dat, aes(x = log(LotArea), y = log(SalePrice))) +
  geom_point(alpha = .2, size = .5, colour = "grey20")
```



Housing Prices (5)

```
lm_mod <- linear_reg() %>%
  set_engine("lm")
lm_fit <- lm_mod %>%
  fit(log(SalePrice) ~ log(LotArea), data = dat)
lm_fit
```

```
## parsnip model object
##
## Fit time:  7ms
##
## Call:
## stats::lm(formula = log(SalePrice) ~ log(LotArea), data = data)
##
## Coefficients:
##  (Intercept)  log(LotArea)
##       9.2113        0.3087
```

Housing Prices (6)

```
tidy(lm_fit)
```

```
## # A tibble: 2 × 5
##   term          estimate std.error statistic  p.value
##   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      9.21     0.169       54.5 0
## 2 log(LotArea)     0.309    0.0185      16.7 3.47e-57
```

$$\log(\text{SalePrice}) = \beta_0 + \beta_1 \times \log(\text{LotArea})$$

- so the estimate of intercept β_0 and the estimate of `LotArea` term β_1 are `9.21` and `0.309` respectively.
- how are we gonna get those two estimates?
- the core idea is to take the likelihood of all observations (which is the product of all pdfs of these observations). And remember, the we "assume" the data is normally distributed, that's the reason why we can use normal distribution as the pdf here.

Housing Prices (7)

- then, we take the log of the likelihood. Why? because of calculus. recall that if we want to find the maximizer/minimizer of a product of some polynomials, very computationally expensive. But if we take the log, the product of polynomials will turn to the sum of polynomials.
- eventually, finding the maximizer/minimizer $=$ solve for the equation where (first derivative $=$ 0).

Housing Prices (8)

```
beta_1_est <- cov(log(dat$LotArea), log(dat$SalePrice)) / var(log(dat$LotArea))
beta_1_est
```

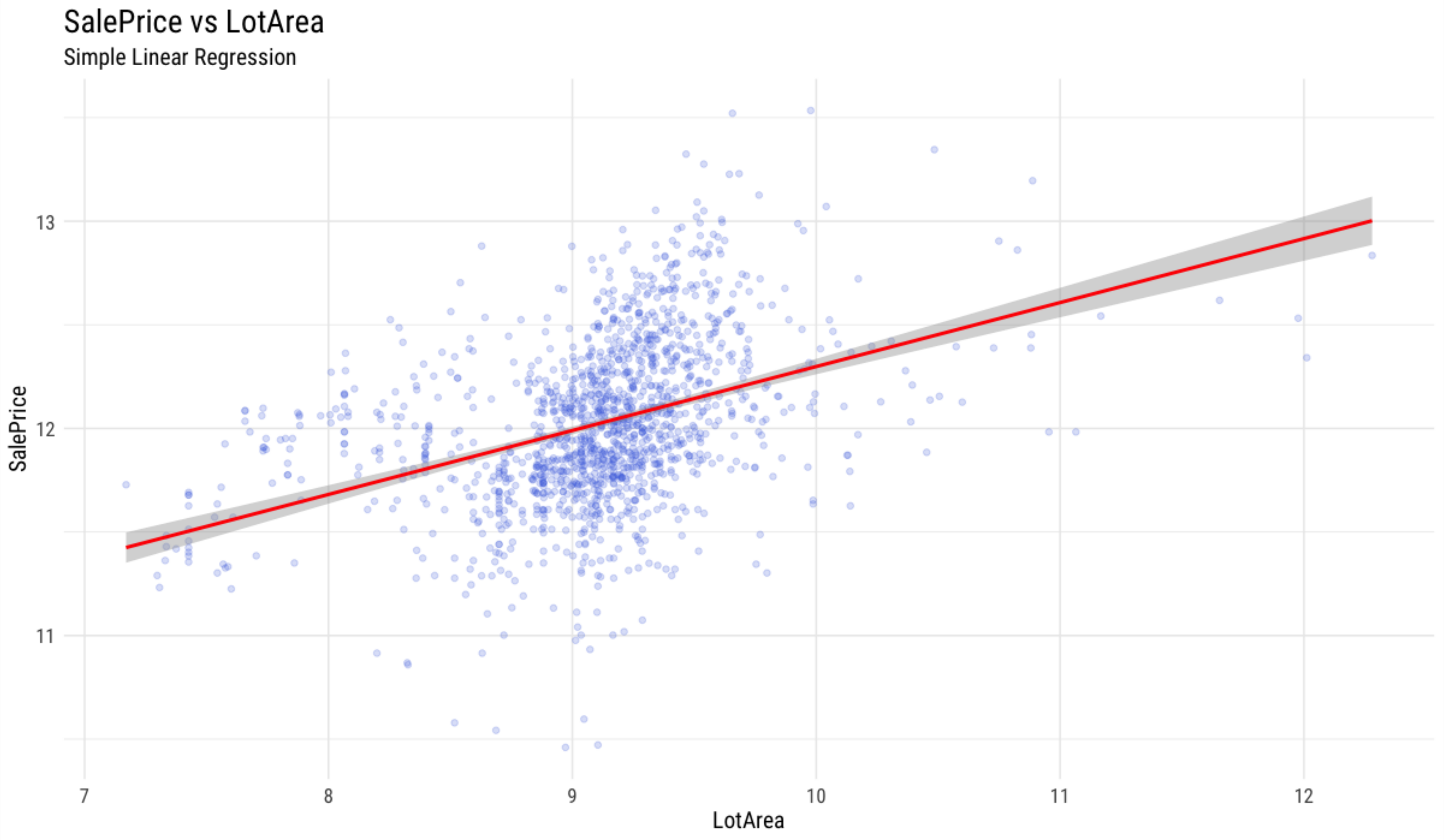
```
## [1] 0.3087226
```

```
beta_0_est <- mean(log(dat$SalePrice)) - beta_1_est * mean(log(dat$LotArea))
beta_0_est
```

```
## [1] 9.21133
```

Housing Prices (9)

Redraw the model in a more stylish manner:



Homebrew vs Package

Response vs Prediction (1)

Show 7 entries Search:

	LotArea	SalePrice	log LotArea	log SalePrice	Estimated log(SalePrice)
1	8450	208500	9.04192172035122	12.247694320221	12.002774817351
2	9600	181500	9.16951837745593	12.109010932687	12.0421667831259
3	11250	223500	9.32812340763257	12.3171666930358	12.0911317330095
4	9550	140000	9.16429643347478	11.8493977015914	12.0405546512468
5	14260	250000	9.56521369396829	12.4292161968444	12.1643268515716
6	14115	143000	9.55499334068759	11.870599909242	12.1611715980111
7	10084	307000	9.21870528830781	12.6346030265693	12.0573518918334

Showing 1 to 7 of 1,460 entries

Previous 1 2 3 4 5 … 209 Next

SLR in Matrix Form

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 X_1 \\ \beta_0 + \beta_1 X_2 \\ \vdots \\ \beta_0 + \beta_1 X_n \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$$\begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \\ \vdots \\ \hat{Y}_n \end{bmatrix} = \begin{bmatrix} 1 & X_1 \\ 1 & X_2 \\ \vdots & \vdots \\ 1 & X_n \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Homebrew MLE Estimation

```
homebrew_slr <- function(x, y){
  beta_1_est <- cov(x, y) / var(x)
  beta_0_est <- mean(y) - beta_1_est * mean(x)
  return(c(beta_0_est, beta_1_est))
}
homebrew_slr(dat$`log LotArea`, dat$`log SalePrice`)
```

```
## [1] 9.2113297 0.3087226
```

Response vs Prediction (2)

Show 7 entries Search:

	log LotArea	log SalePrice	Estimated log(SalePrice)	Estimated log(SalePrice) *
1	9.04192172035122	12.247694320221	12.002774817351	12.002774817351
2	9.16951837745593	12.109010932687	12.0421667831259	12.0421667831259
3	9.32812340763257	12.3171666930358	12.0911317330095	12.0911317330095
4	9.16429643347478	11.8493977015914	12.0405546512468	12.0405546512468
5	9.56521369396829	12.4292161968444	12.1643268515716	12.1643268515716
6	9.55499334068759	11.870599909242	12.1611715980111	12.1611715980111
7	9.21870528830781	12.6346030265693	12.0573518918334	12.0573518918334

Showing 1 to 7 of 1,460 entries

Previous 1 2 3 4 5 … 209 Next

Summary

Summary

The model in the article is a Simple Linear Regression (SLR) model, it only considers one variable and one continuous response.

However, an SLR model usually "over-simplifies" the real world. (It might be true for some extreme cases.) You see, a single factor does not lead to a certain result. This world does not operate like this.

For a more complex case, we might want to consider Multiple Linear Regression (MLR) with more than one variables. In some cases, a Generalized Linear Model is considered to fit the data. That is, of course, another topic we should discuss.

Scripts and slides used in the presentation are available on GitHub: **rexarski/case-study-mle**.

Slides created via the R package **xaringan**.

The chakra comes from remark.js, **knitr**, and R Markdown.