Statistical linear regression models

Brian Caffo, Jeff Leek, Roger Peng
Johns Hopkins Bloomberg School of Public Health

Basic regression model with additive Gaussian errors.

- Least squares is an estimation tool, how do we do inference?
- Consider developing a probabilistic model for linear regression

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

- Here the ϵ_i are assumed iid $N(0, \sigma^2)$.
- Note, $E[Y_i \mid X_i = x_i] =: \mu_i = \beta_0 + \beta_1 x_i$
- Note, $Var(Y_i \mid X_i = x_i) = \sigma^2$.

Fragwuerdige Annahme!

- Likelihood equivalent model specification is that the Y_i are independent $N(\mu_i, \sigma^2)$.

```
Aus aehnlichem Grund:

Y_i = beta0 + beta1 * X_i + e_i
Var(Y_i) = var(beta0 + beta1 * X_i + e_i)

und alles ausser e_i ist konstant in
diesem Term, also = var(ei) = sigma^2,
weil e_i ja iid N(0, sigma^2)
```

```
Weil, basierend auf obiger Formel:
Y_i = beta0 + beta1 * X_i + e_i
also
E[Y_i] = E[beta0 + beta1 * X_i + e_i]
E[Y_i] = E[beta0 + beta1 * X_i] + E[e_i]

Weil die Condition X_i=x_i gegeben ist, ist der erste
Teil nicht-random, also einfach

E[Y_i] = beta0 + beta1 * X_i + E[e_i]
und E[e_i] = 0, weil e_i ja iid N(0,sigma^2)
```

Likelihood

beta=(beta0, beta1)

*Das ">" sollte ein griechisches L sein

$$\mathcal{L}^{*}(\beta, \sigma) = \prod_{i=1}^{n} \left\{ (2\pi\sigma^2)^{-1/2} \exp\left(- \frac{1}{2\sigma^2} (y_i - \mu_i)^2 \right) \right\}$$

wo ist beta hier??

so that the twice the negative log (base e) likelihood is

Produkt-Zeichen ist muehsam, daher log nehmen:

$$-2 \log\{\mathcal{L}^{*}(\beta, \sigma)\} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \mu_i)^2 + n \log(\sigma^2)$$

-2, um einfacher zu rechnen(wegen ^-1/2 in erster Formel)

Discussion

- Maximizing the likelihood is the same as minimizing -2 log likelihood
- The least squares estimate for $\mu_i = \beta_0 + \beta_1 x_i$ is exactly the maximimum likelihood estimate (regardless of σ)

Also unter der Annahme der Gauss-Normalverteilung ist Max Likelihood Estimate das selbe wie Minimieren der least squares, die durch beta0 und beta1 bestimmt sind.

Recap

- Model $Y_i = \mu_i + \epsilon_i = \beta_0 + \beta_1 X_i + \epsilon_i$ where ϵ_i are iid $N(0, \sigma^2)$
- ML estimates of β_0 and β_1 are the least squares estimates

$$\hat{\beta}_1 = Cor(Y, X) \frac{Sd(Y)}{Sd(X)} \quad \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

- $E[Y \mid X = x] = \beta_0 + \beta_1 x$
- $Var(Y \mid X = x) = \sigma^2$

Interpretting regression coefficients, the itc

Interceptor

- β_0 is the expected value of the response when the predictor is 0

$$E[Y|X = 0] = \beta_0 + \beta_1 \times 0 = \beta_0$$

- Note, this isn't always of interest, for example when $X = 0$ is impossible or far outside of the range of data. (X is blood pressure, or height etc.)

In solchen Faellen macht es keinen Sinn, den Intercept zu interpretieren (sagt er; ?! Was, wenn man es auf normalisierte Daten bezieht?)

- Consider that

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i = \beta_0 + a\beta_1 + \beta_1(X_i - a) + \epsilon_i = \tilde{\beta}_0 + \beta_1(X_i - a) + \epsilon_i$$

Trick: a abziehen und wieder addieren

dh X_i um ein a verschieben. Nun hat beta0 sich geaendert, nicht aber beta1, the slope

So, shifting you X values by value a changes the intercept, but not the slope.

- Often a is set to $\bar{X}$ so that the intercept is interpretted as the expected response at the average X value.

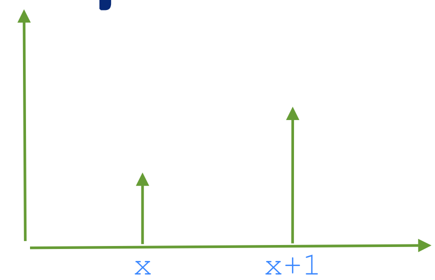
=> ist das nicht aehnlich wie einfach Standardisieren/normalisieren?

Siehe Diamonds, unten:
Ja, stimmt, aber man verliert den einfachen Bezug zur Einheit der abhaengigen Variable.
Besser, wenn man einfach zentriert (mean abzieht): dann bleibt die Streuung und damit die Bedeutung der Abweichung bei der unabh.Variablen gleich.

Nach Standardisieren entspricht eine Einheit auf der X-Achse einer Einheit auf der Y-Achse, also eine Standardabweichung.

Interpretting regression coefficients, the slope

- β_1 is the expected change in response for a 1 unit change in the predictor



$$E[Y \mid X = x+1] - E[Y \mid X = x] = \beta_0 + \beta_1(x+1) - (\beta_0 + \beta_1 x) = \beta_1$$

- Consider the impact of changing the units of X.

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i = \beta_0 + \frac{\beta_1}{a}(X_i a) + \epsilon_i = \beta_0 + \tilde{\beta}_1(X_i a) + \epsilon_i$$

- Therefore, multiplication of X by a factor a results in dividing the coefficient by a factor of a. logisch
- Example: X is height in m and Y is weight in kg. Then β_1 is $\mathrm{kg/m}$. Converting X to cm implies multiplying X by $100\mathrm{cm/m}$. To get β_1 in the right units, we have to divide by $100\mathrm{cm/m}$ to get it to have the right units.

$$X\mathrm{m} \times \frac{100\mathrm{cm}}{\mathrm{m}} = (100X)\mathrm{cm} \quad \text{and} \quad \beta_1 \frac{\mathrm{kg}}{\mathrm{m}} \times \frac{1\mathrm{m}}{100\mathrm{cm}} = \left(\frac{\beta_1}{100}\right)\frac{\mathrm{kg}}{\mathrm{cm}}$$

-> keep track of the units!

Using regression coeficients for prediction

- If we would like to guess the outcome at a particular value of the predictor, say X, the regression model guesses

predicted y:
y^(X) =
$$\hat{\beta}_0 + \hat{\beta}_1 X$$

warum jetzt …^ ??

- Note that at the observed value of Xs, we obtain the predictions

$$\hat{\mu}_i = \hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

Y^_i ist nicht dasselbe wie Y_i:
Ersterer ist der vorhergesagte Wert
fuer X_i, letzterer der tatsaechliche,
observed Datenpunkt fuer X_i.

- Remember that least squares minimizes

$$\sum_{i=1}^{n} (Y_i - \mu_i)$$

for μ_i expressed as points on a line

Und was ist der Unterschied zwischen
mu_i und mu^_i?

See Video 01_05_b, right at the end.

Von naechster Lektion:
der Hut (^) heisst einfach ‘estimated’,
also estimated interceptor, estimated
slope.

Example

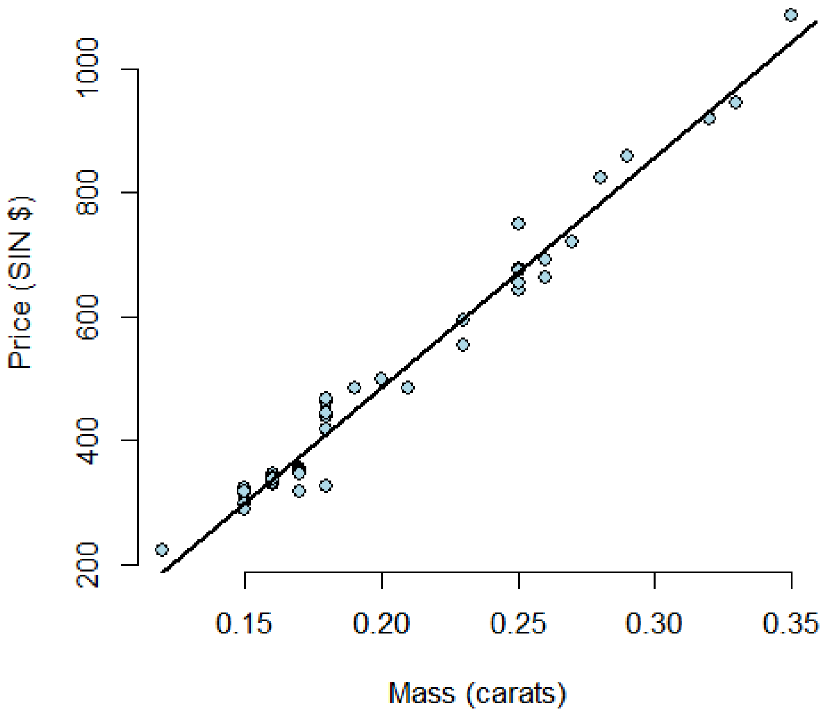
`diamond` data set from `UsingR`

Data is diamond prices (Signapore dollars) and diamond weight in carats (standard measure of diamond mass, 0.2 g). To get the data use `library(UsingR); data(diamond)`

Plotting the fitted regression line and data

```
data(diamond)
plot(diamond$carat, diamond$price,
     xlab = "Mass (carats)",
     ylab = "Price (SIN $)",
     bg = "lightblue",
     col = "black", cex = 1.1, pch = 21,frame = FALSE)
abline(lm(price ~ carat, data = diamond), lwd = 2)
```

The plot



Fitting the linear regression model

```
fit <- lm(price ~ carat, data = diamond)
coef(fit)
```

```
(Intercept)       carat
     -259.6      3721.0
```

==>>
- We estimate an expected 3721.02 (SIN) dollar increase in price for every carat increase in mass of diamond.
- The intercept -259.63 is the expected price of a 0 carat diamond.
  nicht sehr sinnvoll

Getting a more interpretable intercept

```
fit2 <- lm(price ~ I(carat - mean(carat)), data = diamond)
coef(fit2)
```

zentrieren!

```
            (Intercept) I(carat - mean(carat))
                  500.1                 3721.0
```

Thus $500.1 is the expected price for the average sized diamond of the data (0.2042 carats).

Und jetzt hat das Intercept eine Bedeutung!

Changing scale

- A one carat increase in a diamond is pretty big, what about changing units to 1/10th of a carat?
- We can just do this by just dividing the coeficient by 10.
  - We expect a 372.102 (SIN) dollar change in price for every 1/10th of a carat increase in mass of diamond.
- Showing that it's the same if we rescale the Xs and refit

```
fit3 <- lm(price ~ I(carat * 10), data = diamond)
coef(fit3)
```

```
  (Intercept) I(carat * 10)
       -259.6         372.1
```

Predicting the price of a diamond

```
newx <- c(0.16, 0.27, 0.34)
coef(fit)[1] + coef(fit)[2] * newx
```

```
[1]  335.7  745.1 1005.5
```

Dasselbe, einfacher mit Convenience-Fct von R:

```
predict(fit, newdata = data.frame(carat = newx))
```

Variable muss selben Namen haben wie in der fit-Variable,
daher hier ein Rename im Dataframe

```
     1      2      3 
 335.7  745.1 1005.5 
```

Predicted values at the observed Xs (red) and at the new Xs (lines)