Inference in regression

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

Recall our model and fitted values

- Consider the model

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

- $\epsilon \overset{iid}{\sim} N(0, \sigma^2)$.
- We assume that the true model is known.
- We assume that you've seen confidence intervals and hypothesis tests before.
- $\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$ estimated beta^0
- $\hat{\beta}_1 = Cor(Y, X) \frac{Sd(Y)}{Sd(X)}$. estimated beta^1

Review

- Statistics like $\frac{\hat{\theta}-\theta}{\hat{\sigma}_{\hat{\theta}}}$ often have the following properties.
  1. Is normally distributed and has a finite sample Student's T distribution if the estimated variance is replaced with a sample estimate (under normality assumptions).
  2. Can be used to test $H_0 : \theta = \theta_0$ versus $H_a : \theta >, <, \neq \theta_0$.
  3. Can be used to create a confidence interval for θ via $\hat{\theta} \pm Q_{1-\alpha/2}\hat{\sigma}_{\hat{\theta}}$ where $Q_{1-\alpha/2}$ is the relevant quantile from either a normal or T distribution.
- In the case of regression with iid sampling assumptions and normal errors, our inferences will follow very similarily to what you saw in your inference class.
- We won't cover asymptotics for regression analysis, but suffice it to say that under assumptions on the ways in which the X values are collected, the iid sampling model, and mean model, the normal results hold to create intervals and confidence intervals

Standard errors (conditioned on X)

einfacher waere es mit linearer Algebra

$$\mathrm{Var}(\hat{\beta}_1) = \mathrm{Var}\left(\frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(X_i - \bar{X})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} \right)$$

beta estimator einsetzen

$$= \frac{\mathrm{Var}\left(\sum_{i=1}^{n} Y_i(X_i - \bar{X})\right)}{\left(\sum_{i=1}^{n}(X_i - \bar{X})^2\right)^2}$$

$$= \frac{\sum_{i=1}^{n} \sigma^2(X_i - \bar{X})^2}{\left(\sum_{i=1}^{n}(X_i - \bar{X})^2\right)^2}$$

$$= \frac{\sigma^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

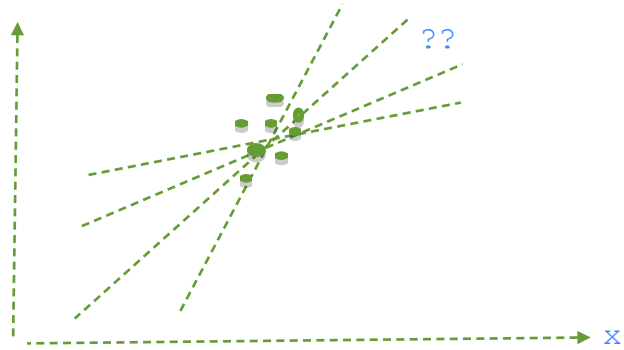
Rechnungshilfe: von erster zu zweiter Zeile:

sum(y_i - y_) = 0
sum(x_i - x_) = 0

sum((x_i - x_)(y_i - y_)) = sum((x_i - x_)y_i
oder = sum((y_i - y_)x_i

Varianz in beta^1 ist am kleinsten, wenn wir viel Varianz im Prediktor (den Xs) haben!
Varianz in X ist also wichtig!

Weil wie sollen wir eine Steigung schaetzen, wenn die X alle zusammenclustern:



Hingegen bei den Y sollte das sigma moeglichst klein sein, um eine gute Prediction machen zu koennen.

Results

- $\sigma^2_{\hat{\beta}_1} = \mathrm{Var}(\hat{\beta}_1) = \sigma^2 / \sum_{i=1}^{n} (X_i - \bar{X})^2$
- $\sigma^2_{\hat{\beta}_0} = \mathrm{Var}(\hat{\beta}_0) = \left(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n} (X_i - \bar{X})^2} \right) \sigma^2$
- In practice, σ is replaced by its estimate.

weil wir sigma ja nicht kennen

- It's probably not surprising that under iid Gaussian errors

$$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}_{\hat{\beta}_j}}$$

j??
j aus {0, 1} fuer einfache lin.Regr?

follows a t distribution with $n - 2$ degrees of freedom and a normal distribution for large n.

- This can be used to create confidence intervals and perform hypothesis tests.

Example diamond data set

```
library(UsingR); data(diamond)
y <- diamond$price; x <- diamond$carat; n <- length(y)
beta1 <- cor(y, x) * sd(y) / sd(x)
beta0 <- mean(y) - beta1 * mean(x)
e <- y - beta0 - beta1 * x
sigma <- sqrt(sum(e^2) / (n-2))
ssx <- sum((x - mean(x))^2)
seBeta0 <- (1 / n + mean(x) ^ 2 / ssx) ^ .5 * sigma
seBeta1 <- sigma / sqrt(ssx)
tBeta0 <- beta0 / seBeta0; tBeta1 <- beta1 / seBeta1
pBeta0 <- 2 * pt(abs(tBeta0), df = n - 2, lower.tail = FALSE)
pBeta1 <- 2 * pt(abs(tBeta1), df = n - 2, lower.tail = FALSE)
coefTable <- rbind(c(beta0, seBeta0, tBeta0, pBeta0), c(beta1, seBeta1, tBeta1, pBeta1))
colnames(coefTable) <- c("Estimate", "Std. Error", "t value", "P(>|t|)")
rownames(coefTable) <- c("(Intercept)", "x")
```

```
t-Statistiken fuer H1: beta0 != 0 (zu H0: beta0 = 0)
resp.  H1: beta1 != 0 (zu H0: beta1 = 0)

falls man einen Wert>1 kriegt,
hat man die falsche Seite genommen!
```

Example continued

```
coefTable
```

```
            Estimate Std. Error t value   P(>|t|)
(Intercept)   -259.6      17.32  -14.99 2.523e-19
x             3721.0      81.79   45.50 6.751e-40
```

einfacher gehts mit lm:

```
fit <- lm(y ~ x);
summary(fit)$coefficients
```

```
            Estimate Std. Error t value  Pr(>|t|)
(Intercept)   -259.6      17.32  -14.99 2.523e-19
x             3721.0      81.79   45.50 6.751e-40
```

Getting a confidence interval

```
sumCoef <- summary(fit)$coefficients
sumCoef[1,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[1, 2]
```
beta0 Std.Err of beta0

```
[1] -294.5 -224.8
```

und dasselbe fuer beta1:

```
sumCoef[2,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[2, 2]
```

```
[1] 3556 3886
```

With 95% confidence, we estimate that a 0.1 carat increase in diamond size results in a 355.6 to 388.6 increase in price in (Singapore) dollars.

Prediction of outcomes

- Consider predicting Y at a value of X
  - Predicting the price of a diamond given the carat
  - Predicting the height of a child given the height of the parents
- The obvious estimate for prediction at point x_0 is

$$\hat{\beta}_0 + \hat{\beta}_1 x_0$$

- A standard error is needed to create a prediction interval.
- There's a distinction between intervals for the regression line at point x_0 and the prediction of what a y would be at point x_0.
- Line at x_0 se, $\hat{\sigma}\sqrt{\frac{1}{n} + \frac{(x_0 - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}}$
- Prediction interval se at x_0, $\hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{X})^2}{\sum_{i=1}^{n}(X_i - \bar{X})^2}}$

Das Prediction interval im fuer Punkt x_0 ist also etwas groesser.

Fuer die Prediction-Line: wenn n gross ist und wir naehe beim Mean schauen, kann der Fehler fast 0 werden.

Jedoch fuer das Intervall nie (die 1 geht nie weg): das Intervall kann nie kleiner werden als das Sigma, die Streuung aller Punkte.

Vgl. Bild auf uebernaechster Folie.

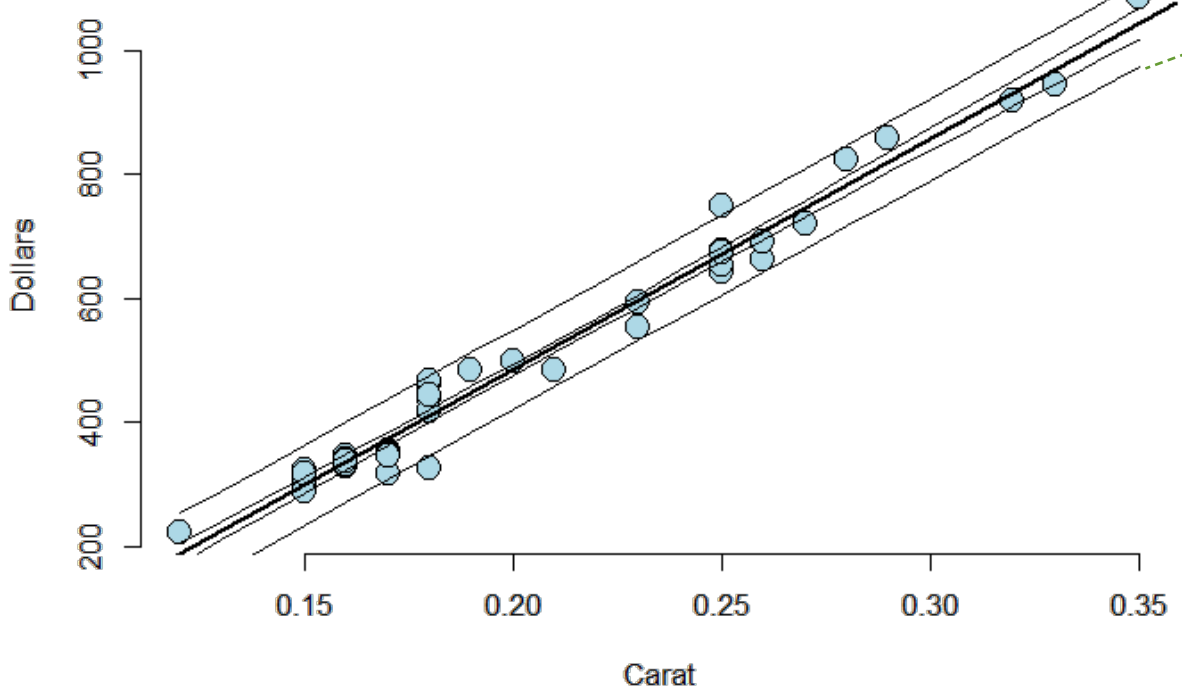
Plotting the prediction intervals

```
plot(x, y, frame=FALSE,xlab="Carat",ylab="Dollars",pch=21,col="black", bg="lightblue", cex=2)
abline(fit, lwd = 2)
xVals <- seq(min(x), max(x), by = .01)
yVals <- beta0 + beta1 * xVals
se1 <- sigma * sqrt(1 / n + (xVals - mean(x))^2/ssx)      se wird kleiner, je naeher x am
se2 <- sigma * sqrt(1 + 1 / n + (xVals - mean(x))^2/ssx)  mean x_ ist!
lines(xVals, yVals + 2 * se1)
lines(xVals, yVals - 2 * se1)
lines(xVals, yVals + 2 * se2)
lines(xVals, yVals - 2 * se2)
```

```
2 ~= 1.96 (und der t-Wert ist sogar noch etwas groesser)

Korrekt waere: qt(0.975, df= n-2)
```

Plotting the prediction intervals



Wenn man einen Punkt da draussen vor-hersagen muesste, also weit weg vom Mean, wird der Fehler viel groesser.

Prediction der ganzen Regressionslinie ist viel sicherer als einzelner Punkte!
Darum ist das Linien-Intervall viel kleiner.
Weil alle Punkte zusammen zur Prediction beitragen.

Auch wenn wir aber noch so viele Punkte haben, die Unsicherheit bei der Prediction eines einzelnen Punktes ist nie kleiner als die Streuung all dieser Punkte um die Linie!
Daher die 1 in der Formel zwei Folien weiter oben.

Discussion

- Both intervals have varying widths.
  - Least width at the mean of the Xs.
- We are quite confident in the regression line, so that interval is very narrow.
  - If we knew β_0 and β_1 this interval would have zero width. aber wir wissen nur beta^0 und beta^1
- The prediction interval must incorporate the variabilibity in the data around the line.
  - Even if we knew β_0 and β_1 this interval would still have width.

In R

```
newdata <- data.frame(x = xVals)
p1 <- predict(fit, newdata, interval = ("confidence"))
p2 <- predict(fit, newdata, interval = ("prediction"))
plot(x, y, frame=FALSE,xlab="Carat",ylab="Dollars",pch=21,col="black", bg="lightblue", cex=2)
abline(fit, lwd = 2)
lines(xVals, p1[,2]); lines(xVals, p1[,3])
lines(xVals, p2[,2]); lines(xVals, p2[,3])
```

In R