Coursera Regression Models Quiz 2

Cheng-Han Yu

August 11, 2015

Question 1

Consider the following data with x as the predictor and y as as the outcome. `x <- c(0.61, 0.93, 0.83, 0.35, 0.54, 0.16, 0.91, 0.62, 0.62)` `y <- c(0.67, 0.84, 0.6, 0.18, 0.85, 0.47, 1.1, 0.65, 0.36)`

Give a P-value for the two sided hypothesis test of whether $\beta 1$ from a linear regression model is 0 or not.

Solution:

The easier way is using the the coefficient table from the summary of `lm` model.

```
x <- c(0.61, 0.93, 0.83, 0.35, 0.54, 0.16, 0.91, 0.62, 0.62)
y <- c(0.67, 0.84, 0.6, 0.18, 0.85, 0.47, 1.1, 0.65, 0.36)
fit <- lm(y ~ x)
coefTable <- coef(summary(fit))
(pval <- coefTable[2, 4])
## [1] 0.05296439
```

We can also sompute the P-value using the definitions and formulas as follows. The P-value will be the same as above.

```
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)
seBeta1 <- sigma / sqrt(ssx)
tBeta1 <- beta1 / seBeta1
(pBeta1 <- 2 * pt(abs(tBeta1), df = n - 2, lower.tail = FALSE))
## [1] 0.05296439
```

Question 2

Consider the previous problem, give the estimate of the residual standard deviation.

Solution:

Again, we can use the summary of the `lm` model to extract the the residual standard deviation, or we can compute it using the formula $\sqrt{\sum ni=1e2in-2}$, which is done in Question 1.

```
summary(fit)$sigma
## [1] 0.2229981
(sigma <- sqrt(sum(e ^ 2) / (n - 2)))
## [1] 0.2229981
```

Question 3

In the `mtcars` data set, fit a linear regression model of weight (predictor) on mpg (outcome). Get a 95% confidence interval for the expected mpg at the average weight. What is the lower endpoint?

Solution:

We can use the `predict()` function or the formula $E[\hat{y}] \pm t_{.975,n-2}\hat{\sigma}\sqrt{\frac{1}{n}+\frac{(x_0-\bar{X})^2}{\sum(X_i-\bar{X})^2}}$ at $x_0=\bar{X}$ to get the confidence interval.

```
data(mtcars)
y <- mtcars$mpg
x <- mtcars$wt
fit_car <- lm(y ~ x)
predict(fit_car, newdata = data.frame(x = mean(x)), interval = ("confidence"))
##        fit      lwr      upr
## 1 20.09062 18.99098 21.19027
yhat <- fit_car$coef[1] + fit_car$coef[2] * mean(x)
yhat + c(-1, 1) * qt(.975, df = fit_car$df) * summary(fit_car)$sigma /
sqrt(length(y))
## [1] 18.99098 21.19027
```

Question 4

Refer to the previous question. Read the help file for `mtcars`. What is the weight coefficient interpreted as?

Solution:

Since variable `wt` has unit (lb/1000), the coefficient is interpreted as the estimated expected change in mpg per 1,000 lb increase in weight.

Question 5

Consider again the `mtcars` data set and a linear regression model with mpg as predicted by weight (1,000 lbs). A new car is coming weighing 3000 pounds. Construct a 95% prediction interval for its mpg. What is the upper endpoint?

Solution:

We can simply use `predict()` function to get the prediction interval, or use the formula

$$\hat{y} \pm t_{.975,n-2}\hat{\sigma}\sqrt{1+\frac{1}{n}+\frac{(x_0-\bar{X})^2}{\sum(X_i-\bar{X})^2}}$$ at $x_0=3$.

```
predict(fit_car, newdata = data.frame(x = 3), interval = ("prediction"))
##        fit      lwr      upr
## 1 21.25171 14.92987 27.57355
yhat <- fit_car$coef[1] + fit_car$coef[2] * 3
yhat + c(-1, 1) * qt(.975, df = fit_car$df) * summary(fit_car)$sigma * sqrt(1 +
(1/length(y)) + ((3 - mean(x)) ^ 2 / sum((x - mean(x)) ^ 2)))
## [1] 14.92987 27.57355
```

Question 6

Consider again the `mtcars` data set and a linear regression model with mpg as predicted by weight (in 1,000 lbs). A “short” ton is defined as 2,000 lbs. Construct a 95% confidence interval for the expected change in mpg per 1 short ton increase in weight. Give the lower endpoint.

Solution:

We could change unit of the predictor from 1000 lbs to 2000 lbs.

```
fit_car2 <- lm(y ~ I(x/2))
sumCoef2 <- coef(summary(fit_car2))
(sumCoef2[2,1] + c(-1, 1) * qt(.975, df = fit_car2$df) * sumCoef2[2, 2])
## [1] -12.97262  -8.40527
```

Question 7

If my X from a linear regression is measured in centimeters and I convert it to meters what would happen to the slope coefficient?

Solution: It would get multiplied by 100. Simply consider the following example.

```
x <- c(0.61, 0.93, 0.83, 0.35, 0.54, 0.16, 0.91, 0.62, 0.62)
y <- c(0.67, 0.84, 0.6, 0.18, 0.85, 0.47, 1.1, 0.65, 0.36)
fit <- lm(y ~ x)
fit$coef[2]
##         x
## 0.7224211
x_meter <- x / 100
fit_meter <- lm(y ~ x_meter)
fit_meter$coef[2]
##  x_meter
## 72.24211
```

Question 8

I have an outcome, Y, and a predictor, X and fit a linear regression model with $Y=\beta_0+\beta_1 X+\epsilon$ to obtain $\hat{\beta}_0$ and $\hat{\beta}_1$. What would be the consequence to the subsequent slope and intercept if I were to refit the model with a new regressor, $X+c$ for some constant, c?

Solution:

The new intercept would be $\hat{\beta}_0-c\hat{\beta}_1$. Consider the following example.

```
x <- c(0.61, 0.93, 0.83, 0.35, 0.54, 0.16, 0.91, 0.62, 0.62)
y <- c(0.67, 0.84, 0.6, 0.18, 0.85, 0.47, 1.1, 0.65, 0.36)
fit <- lm(y ~ x)
fit$coef
## (Intercept)           x
##   0.1884572   0.7224211
x_c <- x + 10
fit_c <- lm(y ~ x_c)
fit_c$coef
## (Intercept)         x_c
##  -7.0357536   0.7224211
fit$coef[1] - 10 * fit$coef[2]
## (Intercept)
##   -7.035754
```

Question 9

Refer back to the `mtcars` data set with mpg as an outcome and weight (`wt`) as the predictor. About what is the ratio of the the sum of the squared errors, $\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^2$ when comparing a model with just an intercept (denominator) to the model with the intercept and slope (numerator)?

Solution:

$\hat{Y}_i=\bar{Y}$ when the fitted model has an intercept only.

```
data(mtcars)
y <- mtcars$mpg
x <- mtcars$wt
fit_car <- lm(y ~ x)
sum(resid(fit_car)^2) / sum((y - mean(y)) ^ 2)
## [1] 0.2471672
```

Question 10

Do the residuals always have to sum to 0 in linear regression?

Solution:

If an intercept is included, then they will sum to 0.

```
data(mtcars)
y <- mtcars$mpg
x <- mtcars$wt
fit_car <- lm(y ~ x)
sum(resid(fit_car))
```

```
## [1] -1.637579e-15
fit_car_noic <- lm(y ~ x - 1)
sum(resid(fit_car_noic))
## [1] 98.11672
fit_car_ic <- lm(y ~ rep(1, length(y)))
sum(resid(fit_car_ic))
## [1] -5.995204e-15
```