November 8, 2015

Setting up, displaying and interpreting a regression model

1. Regression to the mean.

Considering a DGP $t = x + \epsilon$, where $x \sim N(\mu, \sigma^2)$ and $\epsilon \sim N(0, \tau^2)$, we want to prove that for an observation $x = \mu + \sigma$, the predicted mean of t given this observation –that is, $\mathbb{E}(t \mid x = \mu + \sigma) = y(\mu + \sigma)$– is less that one standard deviation away from the mean of the marginal distribution of t.

We can get the mean and standard deviation of the marginal distribution of t considering that $\mathbb{E}(t) = \mathbb{E}(x) + \mathbb{E}(\epsilon) = \mu$ and $\text{var}(t) = \text{var}(x) + \text{var}(\epsilon) - \text{cov}(x, \epsilon)$. As $\text{cov}(x, \epsilon) = 0$ by construction, we know that $t \sim N(\mu, \sigma^2 + \tau^2)$.

Now, we want to prove that,

$y(\mu + \sigma) \leq \mathbb{E}(t) + \text{var}(t)$

$\mu + \sigma \leq \mu + \sqrt{\sigma^2 + \tau^2}$

$\sigma \leq \sqrt{\sigma^2 + \tau^2}$

$\sigma^2 \leq \sigma^2 + \tau^2$

Which will always hold in the presence of some noise with variance $\tau^2 \geq 0$. We have proved that for an observation $x = \mu + \sigma$, the predicted mean of t given this observation is less that one standard deviation (of t) away from the mean of the marginal distribution of t. This is equivalent to say that, once we standardize variables x and t, deviations of the input x are associated with smaller deviations of the output t. For this statement we provide the followign intuition.

If we plotted the fitted regression line for data from the DGP $t = x + \epsilon$, we would get, on average, a coefficient for parameter x of 1, which would give the fitted line a slope of 1. However, if we standardized both x and t, so that both had mean 0 and standard deviation 1, we would allways get a coefficient for parameter x (or slope) lower than 1, unless we had no error ($\tau = 0$). We can estimate the coefficient for parameter x as:

$\hat{w_1} = \rho \, \frac{\sigma_t}{\sigma_x}$

Where ρ is the sample correlation of t and x, and σ_t and σ_t the sample standard deviation of t and x respectively. Once the the variables have been standardized, $\sigma_t = \sigma_t = 1$, and the estimate for w is just ρ, which we compute as $\rho = \frac{Cov(x,t)}{\sigma\sqrt{\sigma^2+\tau^2}}$. Thus, the higher the variance of the error term τ, the lower the correlation between x and t, and the lower the estimate for w. And a movement of one standard deviation in x, which is just one, would cause a movement of less than one in the predicted value of t.

$E(t|x = 1) = \frac{Cov(x,t)}{\sigma\sqrt{\sigma^2+\tau^2}} \times 1 \leq 1$

2. Earnings Analysis from Gelman & Hill

How many individuals were removed from the dataset?

A total of 82 individuals were removed for having what appeared to be contaminated (>98% by a large margin) height or weight data (outside reasonable ranges), 791 individuals were removed because salary data was not available (NA) or 0.

What is your preferred model and why?

```
combinedzmodel <- lm(formula = log(z.data.earn) ~
  z.data.weight +
  z.data.female +
  z.data.weight:z.data.female)
```

This model best fits the data and having standardized the variables means we can interpret the intercept as the average salary for men of average weight. The intercept plus the z.data.female coefficient is the average salary for females.

On which grounds do you chose it?

We chose the above because the R-squared value (fit to the data) is the highest achieved from all the variations and the coefficients have a high significance level. The coefficients have much greater significance than the same model using height in place of weight. Regressing earnings on height did produce significant results, but height is correlated with weight and using weight produces a model of greater significance which suggests that the effects (realized earnings) captured by height are actually driven by weights. We included an interaction variable because the effect of weight appeared to have different distributions for males and females.

What are the interpretations of the coefficients?

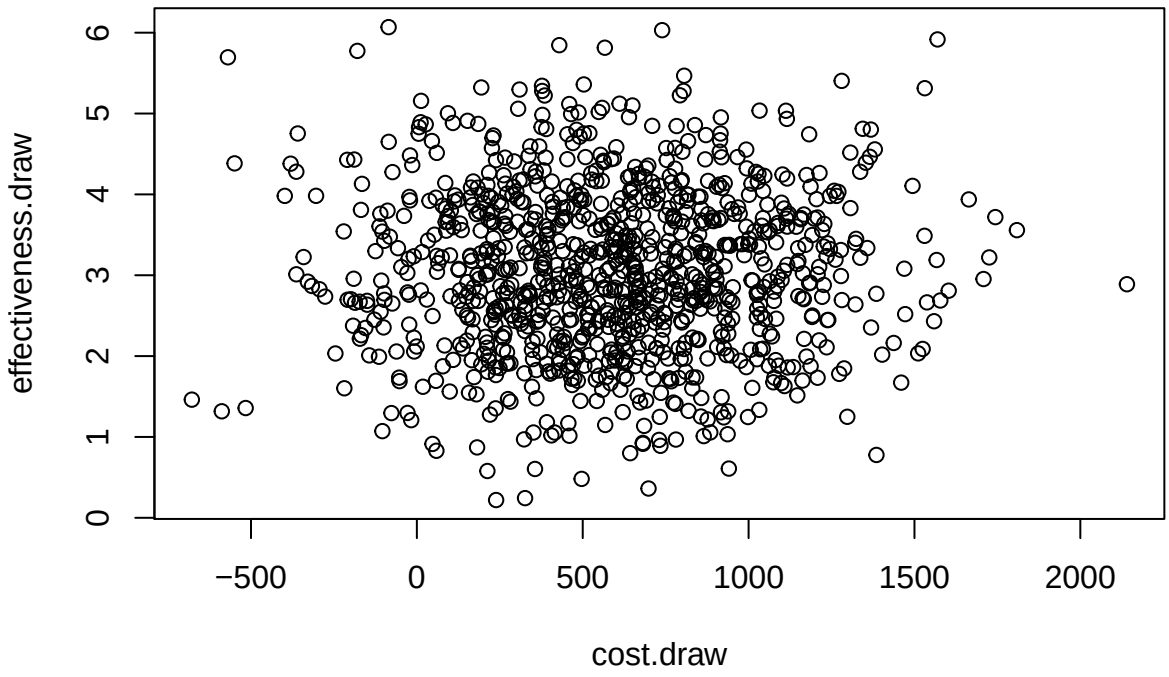
```
# > summary(combinedzmodel)
#
# Call:
# lm(formula = log(z.data.earn) ~ z.data.weight + z.data.female +
#     z.data.weight:z.data.female)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -4.2236 -0.3808  0.1450  0.5659  2.8433
#
# Coefficients:
#                             Estimate Std. Error t value Pr(>|t|)
# (Intercept)                  9.94985    0.04739 209.944  < 2e-16 ***
# z.data.weight                0.26391    0.08936   2.953 0.003208 **
# z.data.female               -0.49223    0.06044  -8.145 9.79e-16 ***
# z.data.weight:z.data.female -0.41468    0.11906  -3.483 0.000514 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.8835 on 1154 degrees of freedom
# Multiple R-squared:  0.09136, Adjusted R-squared:  0.089
# F-statistic: 38.68 on 3 and 1154 DF,  p-value: < 2.2e-16

# > exp(combinedzmodel$coefficients[[1]])
# 20949.12
# > exp(combinedzmodel$coefficients[[1]] + combinedzmodel$coefficients[[2]])
# 27276.11
# > exp(combinedzmodel$coefficients[[1]] + combinedzmodel$coefficients[[3]])
# 12805.34
# Affect for female one sd away from mean
# > exp(combinedzmodel$coefficients[[1]] +
#     combinedzmodel$coefficients[[2]] +
#     combinedzmodel$coefficients[[3]] +
#     combinedzmodel$coefficients[[4]])
# 11013.25
```

Average earnings for males of average weight is $20949.12. Average earnings for females of average weight is $12805.34. For females, being heavier than the mean by one standard deviation decreases the earnings average to $11013.25. For males, this effect is reversed where being heavier than the mean by one standard deviation increases earnings by $6327 to $27276.11. These results are all significant at or greater than the 99% level.

3. Inference for the ratio of parameters: a (hypothetical) study compares the costs and effectiveness of two different medical treatments.

(a) Create 1000 simulation draws of the cost difference and the effectiveness difference, and make a scatterplot of these draws.

```
##
## Attaching package: 'metRology'
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```



(b) Use simulation to come up with an estimate, 50% interval, and 95% interval for the incremental cost-effectiveness ratio.

- Estimate => 198.57
- 50% interval for cost-effectiveness ratio => 138.8713 to 233.9063
- 95% interval for cost-effectiveness ratio => -130.2227 to 258.3142

(c) Repeat this problem, changing the standard error on the difference in effectiveness to 2.0.

- Estimate => 194.8522

- 50% interval for cost-effectiveness ratio => 189.4548 to 203.0848
- 95% interval for cost-effectiveness ratio => 197.4547 to 205.8919

4. Replicate step by step the analysis done in Section 7.3 of Gelman and Hill on predicting electoral results.

```
# Model
# election outcome = constant + democratic_share + incumbency
#   1) constant - self-explanatory
#   2) democratic_share - continuous
#   3) incumbency - categorical: 1 for dem incumbent, -1 for rep incumbent, 0 for open
#
setwd("~/Box Sync/abarciausksas/myfiles/Statistical Modelling and Inference/week4")
library(foreign)

data90 <- read.table('1990.asc')
data88 <- read.table('1988.asc')
data86 <- read.table('1986.asc')
vote <- cbind(data86[,3:5], data88[,3:5], data90[,3:5])
years <- seq(86,90,2)
cnames <- list()
for (y in 1:length(years)) {
  cnames <- append(cnames, lapply(c('incumbency', 'demvote', 'repvote'), paste0, '.', years[y]))
}
cnames <- unlist(cnames)
colnames(vote) <- cnames

vote <- subset(vote, vote[,'demvote.88'] > 0)
vote <- subset(vote, vote[,'demvote.86'] > 0)
vote.88 <- vote[,'demvote.88'] / (vote[,'demvote.88'] + vote[,'repvote.88'])
vote.86 <- vote[,'demvote.86'] / (vote[,'demvote.86'] + vote[,'repvote.86'])

fit.88 <- lm(vote.88 ~ vote.86 + vote[,'incumbency.88'])

# Simulation for inferences and predictions of new data points
#
# Create n.tilde which is the number of congressional districts (that haven't been eliminated)
n.tilde <- length(vote.88)
# X.tilde holds the observable inputs for our prediction,
# that is, an intercept, the vote in 88 and whether a candidata for election is an incumbent.
#
X.tilde <- cbind(rep(1, n.tilde), vote.88, vote[,'incumbency.90'])

# We then simulate 1000 times
library(arm)
n.sims <- 1000
# Generate 1000 simulations of beta0, beta1, beta2
sim.88 <- sim(fit.88, n.sims)
# Initiate an array of predicted outcomes
y.tilde <- array(NA, c(n.sims, n.tilde))
# The predicted outcome for the s-th simulation is a random normal draw with
#   mean X.tilde (vector of inputs for each district) %*% vector of simulated coeffcitions from the si
#
```

```r
for (s in 1:n.sims) {
  y.tilde[s,] <- rnorm(n.tilde, X.tilde %*% slot(sim.88, name='coef')[s,], slot(sim.88, name='sigma')[s]
}
# Total number of districts predicted to be won by democrats
dems.tilde <- rowSums(y.tilde > .5)
# > mean(dems.tilde)
# [1] 250.459
# > sd(dems.tilde)
# [1] 5.206864

# compute these predictions by writing a custom R function
Pred.88 <- function (X.pred, lm.fit) {
  n.pred <- dim(X.pred)[1]
  sim.88 <- sim(lm.fit, 1)
  y.pred <- rnorm(n.pred, X.pred %*% t(slot(sim.88, name='coef')), slot(sim.88, name='sigma'))
  return (y.pred)
}

y.tilde <- replicate (1000, Pred.88(X.tilde, fit.88))
dems.tilde <- replicate (1000, Pred.88 (X.tilde, fit.88) > .5)
```