## 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 $\tau^2 \geq 0$.

### 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 = 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?*

I chose it because all the coefficients have a high significance level. The coefficients have much greater 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. I 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?*

```
# Call:
# lm(formula = z.data.earn ~ z.data.weight + z.data.female + z.data.weight:z.data.female)
#
# Residuals:
```

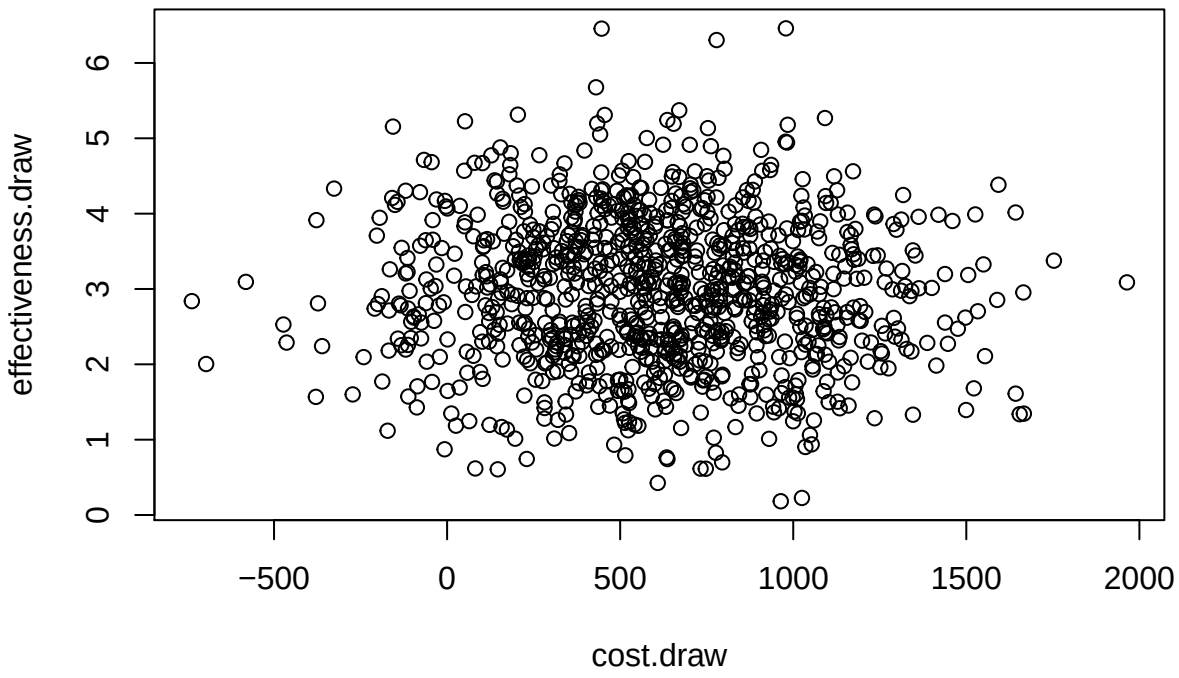
```
#    Min     1Q Median     3Q    Max
# -34855 -10950  -3428   6381 368828
#
# Coefficients:
#                            Estimate Std. Error t value Pr(>|t|)
# (Intercept)                   28485       1149  24.799  < 2e-16 ***
# z.data.weight                  6685       2166   3.087 0.002073 **
# z.data.female                -10827       1465  -7.392 2.77e-13 ***
# z.data.weight:z.data.female   -9967       2886  -3.454 0.000572 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 21410 on 1154 degrees of freedom
# Multiple R-squared:  0.0809,  Adjusted R-squared:  0.07851
# F-statistic: 33.86 on 3 and 1154 DF,  p-value: < 2.2e-16
```

Average earnings for males are $28485. Average earnings for females are $17658. For females, being heavier than the mean by one standard deviation decreases earnings by $9967. For males, this effect is reversed where being heavier than the mean by one standard deviation increases earnings by $6685. 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
#
```

```r
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
#
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)
```