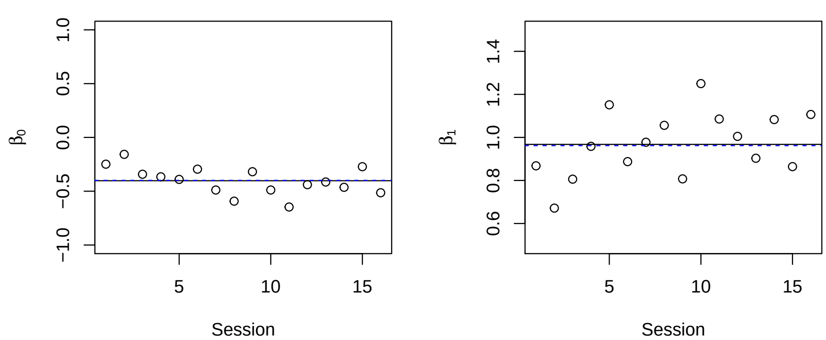
Solutions to SDT exercises

Kenneth Knoblauch
ECVP, Bremen

August 20, 2013

1. Gb.df is based on 16 sessions of 224 trials each. Estimate sensitivity and bias for each session individually and plot them as a function of session. Are there any learning trends? How variable are the measures?

```
load("GabResp.Rdata")  # load data file
Gb.df <- data.frame( Resp = GabResp %in% c("H", "FA"),
                     Stim = GabResp %in% c("H", "M"))
Gb.glm <- glm(Resp ~ Stim, binomial(probit), Gb.df)
x <- seq(0L, 15L) * 224 + 1
res.ses <- t(sapply(x, function(ix){
        coef(update(Gb.glm, data = Gb.df[seq(ix, ix + 223), ]))
}))
par(mfrow = c(1, 2))
plot(1:16, res.ses[, 1], ylim = c(-1, 1),
        xlab = "Session", ylab = expression(beta[0]))
abline(h = mean(res.ses[, 1]))
abline(h = coef(Gb.glm)[1], col = "blue", lty = 2)
plot(1:16, res.ses[, 2], ylim = c(0.5, 1.5),
        xlab = "Session", ylab = expression(beta[1]))
abline(h = coef(Gb.glm)[2], col = "blue", lty = 2)
abline(h = mean(res.ses[, 2]))
```



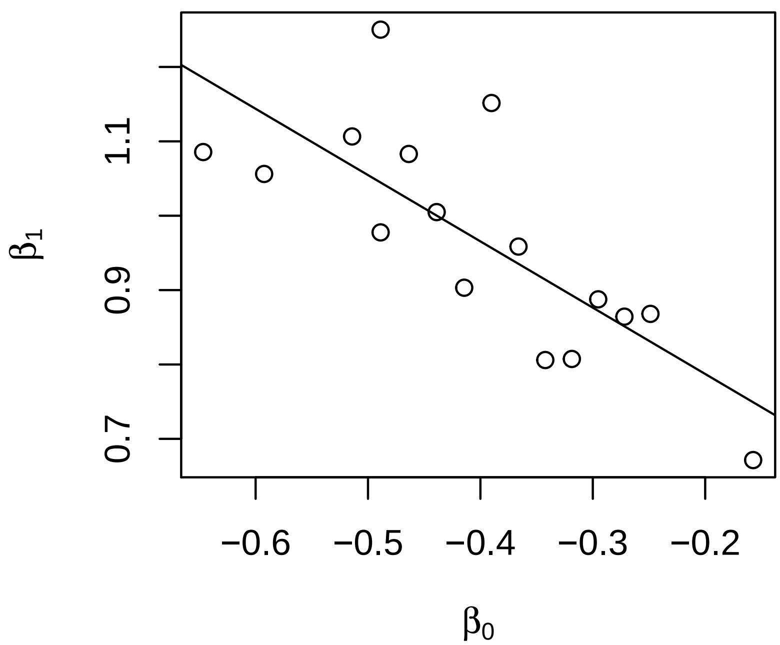

```
ses <- 1:16
summary(lm(res.ses ~ ses))

## Response (Intercept) :
##
```

```
## Call:
## lm(formula = `(Intercept)` ~ ses)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21094 -0.04242 -0.00083  0.05871  0.21716
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.28905    0.06146    -4.7  0.00034 ***
## ses         -0.01333    0.00636    -2.1  0.05456 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.117 on 14 degrees of freedom
## Multiple R-squared:  0.239,Adjusted R-squared:  0.185
## F-statistic:  4.4 on 1 and 14 DF,  p-value: 0.0546
##
##
## Response StimTRUE :
##
## Call:
## lm(formula = StimTRUE ~ ses)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.2085 -0.0969  0.0158  0.0597  0.2623
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.85304    0.07396   11.53  1.6e-08 ***
## ses          0.01348    0.00765    1.76      0.1 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.141 on 14 degrees of freedom
## Multiple R-squared:  0.182,Adjusted R-squared:  0.123
## F-statistic: 3.11 on 1 and 14 DF,  p-value: 0.0998
```

```
cbind(Mean = apply(res.ses, 2, mean),
                SD = apply(res.ses, 2, sd))
```

```
##                Mean     SD
## (Intercept) -0.4024 0.1298
## StimTRUE     0.9676 0.1506
```

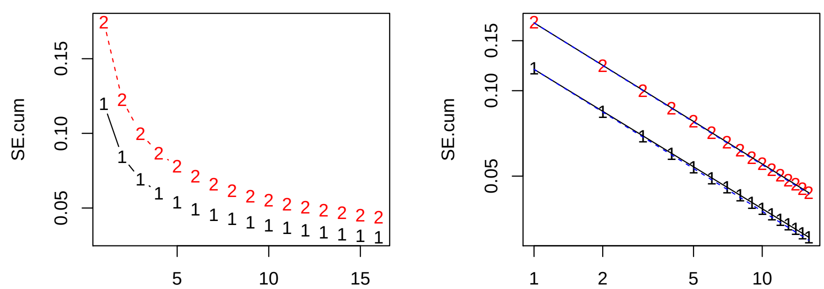
```r
cor(res.ses)

##             (Intercept) StimTRUE
## (Intercept)      1.0000  -0.7681
## StimTRUE        -0.7681   1.0000

plot(res.ses, xlab = expression(beta[0]), ylab = expression(beta[1]))
abline(lm(res.ses[, 2] ~ res.ses[, 1]))
```



2. Fit the SDT detection model to the Gb.df data set cumulatively by session, i.e., first the first session, then the first and second sessions, etc. How do the SEs of the parameter estimates vary with the number of trials?

```r
SE.cum <- t(sapply(x, function(ix){
  res <- update(Gb.glm, data = Gb.df[seq(1, ix + 223), ])
        coef(summary(res))[, 2]
}))
par(mfrow = c(1, 2))
```

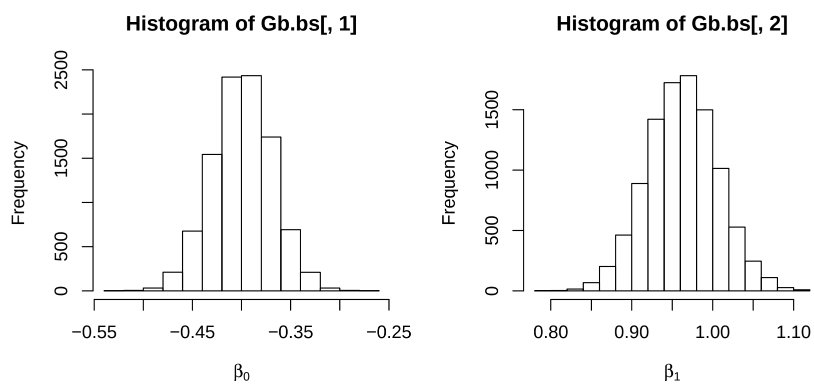
```
matplot(SE.cum, type = "b")
matplot(SE.cum, log = "xy")
(SE.cum.lm <- lm(log10(SE.cum) ~ log10(ses)))

##
## Call:
## lm(formula = log10(SE.cum) ~ log10(ses))
##
## Coefficients:
##              (Intercept)  StimTRUE
## (Intercept)  -0.925       -0.761
## log10(ses)   -0.491       -0.497

cc <- coef(SE.cum.lm)
lines(ses, 10^cc[1, 1] * ses^cc[2, 1])
lines(ses, 10^cc[1, 2] * ses^cc[2, 2])
lines(ses, 10^cc[1, 1] * ses^-0.5, col = "blue", lty = 2)
lines(ses, 10^cc[1, 2] * ses^-0.5, col = "blue", lty = 2)
```



3. Bootstrap standard errors. With the model object, Gb.glm, obtain the fitted values with the fitted method (fitted(Gb.glm)). Then, use these as probabilities with the rbinom function to generate new Bernoulli responses based on the fitted probabilities. Use these new simulated responses in place of the actual ones to fit a glm, estimating new parameters, * and *, and store them. Repeat this many time (e.g., 10000). Calculate the mean and SDs of the bootstrap parameter estimates. The difference between the mean and value obtained on the original data is an estimate of bias. The SDs are estimates of the SEs, the 0.025 and 0.975 quantiles of the distribution are 95estimates.

```
# should run faster if multiple cores present and run in parallel
library(parallel)
cl <- makeCluster(getOption("cl.cores", detectCores()))
N <- 10000
```

```r
Gb.fit <- fitted(Gb.glm)
Resp.bs <- rbinom(N * length(Gb.fit), 1, Gb.fit)
Resp.bs <- matrix(Resp.bs, nrow = length(Gb.fit), ncol = N)
clusterExport(cl, c("Resp.bs", "Gb.df"))
# this step will take some time
Gb.bs <- t(parSapply(cl, seq(N), function(ix){
  Gb.bt <- glm(Resp.bs[, ix] ~ Gb.df$Stim, binomial(probit))
        coef(Gb.bt)
}))
par(mfrow = c(1, 2))
hist(Gb.bs[, 1], xlab = expression(beta[0]))
hist(Gb.bs[, 2], xlab = expression(beta[1]))
```




```r
Bs.mn <- apply(Gb.bs, 2, mean)
bias <- coef(Gb.glm) -  Bs.mn
Bs.se <- apply(Gb.bs, 2, sd)
( Gb.bsCI <- t(apply(Gb.bs, 2, quantile, p = c(0.025, 0.975))) )

##                      2.5%   97.5%
## (Intercept)     -0.4590 -0.3408
## Gb.df$StimTRUE  0.8772  1.0489

#  compare w/ GLM method from vcov matrix
( Gb.CI <- confint(Gb.glm) )

## Waiting for profiling to be done...

##                2.5 %  97.5 %
## (Intercept) -0.4591 -0.3396
## StimTRUE     0.8763  1.0478
```