STATS 330 Assignment 4

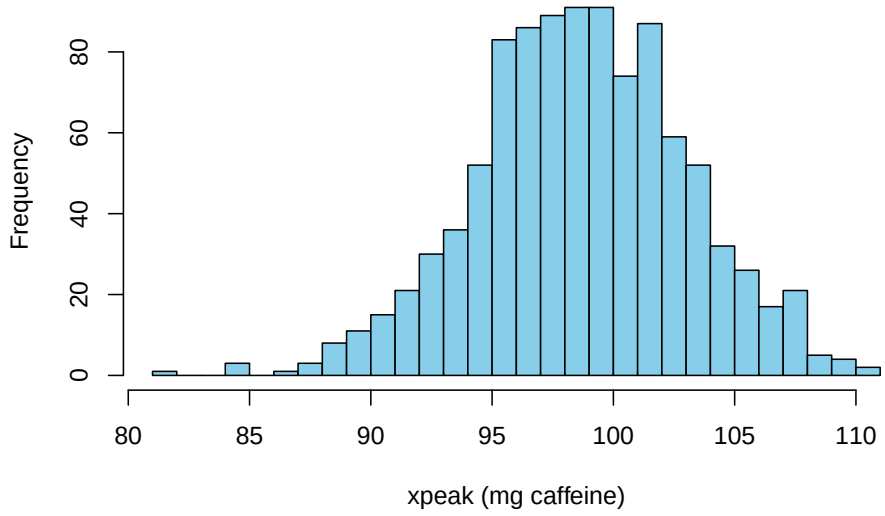
Anish Hota

2025-06-03

1A

```
xpeaks <- numeric(1000)
for (b in 1:1000) {
  responses <- numeric(5)
  for (i in 1:5) {
    vec <- c(rep(1, Caffeine2.df$Agrade[i]), rep(0, Caffeine2.df$n[i] - Caffeine2.df$Agrade[i]))
    resample <- sample(vec, 300, replace = TRUE)
    responses[i] <- sum(resample)
  }
  boot.df <- data.frame(
    caffeine = Caffeine2.df$caffeine,
    yes = responses,
    n = Caffeine2.df$n
  )
  mod.boot <- glm(cbind(yes, n - yes) ~ caffeine + I(caffeine^2),
                  family = binomial, data = boot.df)

  coefs <- coef(mod.boot)
  beta1 <- coefs["caffeine"]
  beta2 <- coefs["I(caffeine^2)"]
  if (!is.na(beta1) && !is.na(beta2) && beta2 != 0) {
    xpeaks[b] <- -beta1 / (2 * beta2)
  } else {
    xpeaks[b] <- NA
  }
}
xpeaks <- na.omit(xpeaks)
hist(xpeaks, breaks = 30, col = "skyblue", main = "Bootstrap Estimates of xpeak",
     xlab = "xpeak (mg caffeine)")
```



1B

```
confidence_interval <- quantile(xpeaks, probs = c(0.025, 0.975))
confidence_interval
```

```
##      2.5%     97.5%
##  89.87184 107.27713
```

1C

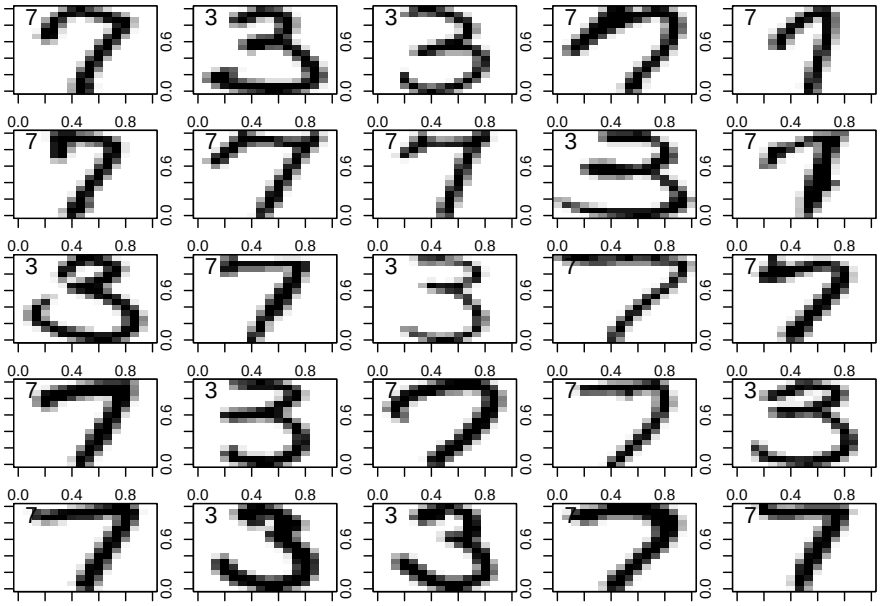
The confidence intervals in both cases are very similar to one another, nearly identical. Meaning that this bootstrap conveyed data very well.

2a

```
train.df = read.table("train.txt")
names(train.df) = c("D", paste("V", 1:256, sep=""))
```

2B

```
par(mfrow=c(5,5), mar = c(1,1,1,1))
for(k in 1:25){
z = matrix(unlist(train.df[k,-1]), 16,16)
zz = z
for(j in 16:1)zz[,j]=z[,17-j]
image(zz, col = gray((32:0)/32))
box()
text(0.1,0.9,train.df$D[k], cex=1.5)
}
```



I would look at the bottom left corner, the 7s seem to barely touch it, while the threes curve up in that corner.

2C

```
train.df$Y <- ifelse(train.df$D == 7, 1, 0)
correlations <- sapply(train.df[, 2:257], function(v) cor(v, train.df$Y))
variations <- names(sort(abs(correlations), decreasing = TRUE))[1:20]
variations
```

```
##  [1] "V185" "V170" "V105" "V220" "V235" "V201" "V229" "V120" "V219" "V230"
## [11] "V104" "V189" "V205" "V169" "V234" "V121" "V204" "V186" "V173" "V221"
```

It seems that pixels that are towards the bottom have the most variation.

2D

```
formula <- as.formula(paste("Y ~", paste(variations, collapse = " + ")))
Full.mod <- glm(formula, data = train.df, family = binomial)
logits <- predict(Full.mod, type = "link")
```

2E

```
probs <- predict(Full.mod, type = "response")
predicted <- ifelse(probs > 0.5, 1, 0)
actual <- train.df$Y
in_sample_error <- mean(predicted != actual)
in_sample_error
```

```
## [1] 0.01074444
```

2F

```
null.model = glm(Y~1, data=train.df, family=binomial)
#step(Full.mod, direction = "backward")
step_mod <- step(null.model, formula(Full.mod),direction="both", trace=0)
probs_step <- predict(step_mod, type = "response")
preds_step <- ifelse(probs_step > 0.5, 1, 0)
step_error <- mean(preds_step != actual)
step_error
```

```
## [1] 0.01304682
```

The step error seems to similar to the in sample error above, just slightly higher, so it is slightly better.

2G

```
PE.fun <- function(model, data, K = 10) {
  n <- nrow(data)
  folds <- sample(rep(1:K, length.out = n))
  errors <- numeric(K)

  for (k in 1:K) {
    train_fold <- data[folds != k, ]
    test_fold <- data[folds == k, ]

    mod_k <- glm(formula(model), data = train_fold, family = binomial)
    probs_k <- predict(mod_k, newdata = test_fold, type = "response")
    preds_k <- ifelse(probs_k > 0.5, 1, 0)
    actual_k <- test_fold$Y
    errors[k] <- mean(preds_k != actual_k)
  }

  mean(errors)
}

cv_error <- PE.fun(step_mod, train.df)
cv_error
```

```
## [1] 0.01455666
```

This sample error is slightly higher than before so it is a better measure.

2H

```
test.df <- read.table("test.txt")
names(test.df) <- c("D", paste("V", 1:256, sep=""))
test.df$Y <- ifelse(test.df$D == 7, 1, 0)

test_probs <- predict(step_mod, newdata = test.df, type = "response")
test_preds <- ifelse(test_probs > 0.5, 1, 0)
test_error <- mean(test_preds != test.df$Y)
test_error
```

```
## [1] 0.02875399
```

This is slightly higher so it is a better fit for this model.