

Statistics 406 Exam – November 17, 2005

1. For each of the following, what do you expect the value of A to be after executing the program? Briefly state your reasoning for each part.

(a)

```
X <- array(rexp(10*1000), c(10,1000))
M <- colMeans(X)
A <- mean(M)
```

Solution: A should be around 1 since the sample mean of *iid* data has the same expected value as the individual observations, and the expected value of a standard exponential observation is 1.

(b)

```
X <- array(rnorm(10*1000), c(10,1000))
M <- colMeans(X)
A <- var(M)
```

Solution: A should be around $1/10$ since the variance of \bar{X} is σ^2/n , where $n = 10$ is the sample size, and σ^2 is the variance of one observation ($\sigma^2 = 1$ here since the data are standard normal).

(c)

```
X <- array(rnorm(10*1000), c(10,1000))
M <- colMeans(X)
B <- var(M)

Y <- array(rnorm(20*1000), c(20,1000))
M <- colMeans(Y)
C <- var(M)

A <- B/C
```

Solution: A should be around 2, since cutting the sample size in half doubles the variance of the sample mean.

(d)

```
X <- array(rnorm(10*1000), c(10,1000))
B <- apply(X, 2, var)
A <- mean(B)
```

Solution: A should be around 1, since the sample variance is unbiased, and the population variance of standard normal data is 1.

(e)

```
X <- array(rexp(50*1000), c(50,1000))
B <- colMeans(X)
C <- apply(X, 2, var)
D <- sqrt(50)*(B - 1)/sqrt(C)
A <- mean(D > 0)
```

Solution: A should be around 1/2, since D is approximately standard normal for large n by the central limit theorem, and a standard normal draw has probability 1/2 of being positive.

(f)

```
X <- rnorm(1000)
Y <- rnorm(1000)
A <- cov(X+Y, X-Y)
```

Solution: A should be around zero, since

$$\begin{aligned}\text{cov}(X + Y, X - Y) &= \text{var}(X) - \text{cov}(X, Y) + \text{cov}(X, Y) - \text{var}(Y) \\ &= \text{var}(X) - \text{var}(Y) \\ &= 0\end{aligned}$$

2. Suppose we plan to use simulation to estimate the probability that the sample variance is further than 0.2 from the population variance, based on a normal sample of size 40. What statement should ??? be replaced with in the following program? Select one from alternatives a-e provided below.

```
X <- array(rnorm(40*1000), c(40,1000))
V <- apply(X, 2, var)
A <- ???
```

- (a) `abs(mean(V-1 > 0.2))`
- (b) `mean(abs(V-1) > 0.2)`
- (c) `mean(V > 1.2)`
- (d) `abs(mean(V) - 0.8)`
- (e) `sum(abs(V-1) < 0.2)`

Solution: The answer is b, so the correct program is:

```

X <- array(rnorm(40*1000), c(40,1000))
V <- apply(X, 2, var)
A <- mean(abs(V-1) > 0.2)

```

`abs(V-1)` is the distance between the sample variance and population variance, and `mean(... > 0.2)` calculates the proportion of the distances that are greater than 0.2.

3. How could an estimate of the variance of the sample standard deviation for an *iid* sample of 100 standard normal values be obtained from the results of the following program?

```

n <- 100
X <- array(rnorm(n*1000), c(n, 1000))
Q <- apply(X, 2, sd)
A <- mean(Q)-1
B <- mean((Q-1)^2)

```

Solution: Since MSE is equal to the variance plus the square of the bias, the variance is equal to MSE minus the square of the bias. Thus an estimate of the variance can be obtained using $B - A^2$. We gave full credit for `var(Q)` also, but this wasn't what we had in mind.

4. What is the statistical meaning of the value of `n` that results from the following program? Be as complete as possible, but keep your comments focused on the statistical meaning of `n`. If the value 0.5 in the program is replaced with 1, do you expect the value of `n` to increase or decrease? Briefly state your reasoning.

```

n <- 10

while (1)
{
  X <- array(rnorm(n*1000), c(n,1000))
  Y <- 0.5 + array(rnorm(n*1000), c(n,1000))

  MX <- colMeans(X)
  MY <- colMeans(Y)

  VX <- apply(X, 2, var)
  VY <- apply(Y, 2, var)

  S <- (MY-MX) / sqrt(VX/n + VY/n)
  Q <- mean(S > 2)
}

```

```

    if (Q > 0.8) { break }
    n <- n+1
}

```

Solution: n is the sample size at which there is 80% power for detecting a mean difference of $1/2$ between two groups of size n . The data are normal with variance 1 within each group, and have a mean difference of 0.5 (so the data are generated from an alternative distribution). A two-sample Z-test is used with the null hypothesis rejected when the test statistic value exceeds two.

If 0.5 is replaced with 1 then the alternative distribution becomes more different from the null distribution, making it easier to detect the difference with high confidence. Thus the sample size n will decrease.

5. What is the statistical meaning of the values in A and B? Do you expect A and B to be increasing sequences of numbers, decreasing sequences of numbers, or approximately constant sequences of numbers (answer separately for A and B).

```

A <- NULL
B <- NULL

for (n in c(10,20,40,80,160))
{
  X <- array(rnorm(n*1000), c(n,1000))

  M <- colMeans(X)
  C <- apply(X, 2, var)
  D <- 1.96*sqrt(C)/sqrt(n)

  c <- ((M-D < 0) & (M+D > 0))
  d <- mean(c)
  A <- c(A, d)
  B <- c(B, 2*mean(D))
}

```

Solution: The values in A are estimated coverage probabilities and the values in B are estimated average confidence interval widths. The confidence intervals are constructed using the usual plug-in rule $\bar{X} \pm 1.96\hat{\sigma}/\sqrt{n}$, and the simulated data are *iid* standard normal.

The coverage probabilities would ideally be approximately constant at 0.95. In practice, since we are using the plug-in variance they will start off somewhat below 0.95 and

increase as n increases, but they will level off at 0.95. The widths will decrease as n increases, since a bigger sample size provides more precise estimates of the unknown population mean.

6. Suppose X contains a list of numbers that have been measured in an experiment. Our goal is to use the non-parametric bootstrap to construct a 95% confidence interval for the expected value of the population from which X was sampled. What statements should ****1****, ****2**** and ****3**** be replaced with in the following program?

```
n <- length(X)
ii <- ceiling(runif(n*1000))
A <- array(X[ii], c(n, 1000))
B <- colMeans(A)
C <- **1**
D <- C[**2**]
E <- C[**3**]
```

Solution: The correct program is

```
n <- length(X)
ii <- ceiling(runif(n*1000))
A <- array(X[ii], c(n, 1000))
B <- colMeans(A)
C <- sort(B)
D <- C[25]
E <- C[975]
```

The confidence interval is bounded by D and E.

7. Suppose we have the average A_n of X_1, \dots, X_n , but we do not have the X_i values used to calculate A_n . We then observe X_{n+1} and wish to calculate A_{n+1} without using X_1, \dots, X_n . Derive a formula for A_{n+1} based on n , A_n and X_{n+1} .

Solution:

$$A_{n+1} = \frac{nA_n + X_{n+1}}{n + 1}$$

8. What is the statistical meaning of the values in Q as computed by following program? What characteristic should the values in Q ideally have?

```

n <- 30

Q <- array(0, 1000)

for (j in (1:1000))
{
  X <- rnorm(n)
  Y <- rnorm(n)

  mx <- mean(X)
  my <- mean(Y)
  vx <- var(X)
  vy <- var(Y)
  A <- (mx - my) / sqrt(vx/n + vy/n)

  Z <- c(X, Y)

  B <- array(0, 1000)
  for (r in (1:1000))
  {
    ii <- order(runif(2*n))

    x <- Z[ii[1:n]]
    y <- Z[ii[(n+1):(2*n)]]

    mx <- mean(x)
    my <- mean(y)
    vx <- var(x)
    vy <- var(y)
    B[r] <- (mx - my) / sqrt(vx/n + vy/n)
  }

  Q[j] <- mean(abs(B) > abs(A))
}

```

Solution: The values in Q are permutation test p-values for testing whether the samples of X and Y have the same expected value. We are evaluating the performance of this testing procedure when the data are *iid* samples of size 30 each from the standard normal distribution. Since X and Y are simulated from a null distribution, the p-values in Q should be uniform.

9. Suppose we observe paired data $(X_1, Y_1), \dots, (X_{100}, Y_{100})$, and we wish to use the non-parametric bootstrap to construct a 95% confidence interval for $r = \text{cor}(X, Y)$. Describe the critical error in the following program, which has the goal of calculating

values `C1` and `C2` that form the lower and upper bounds of a 95% confidence interval. As a result of this error, where will the resulting confidence intervals tend to be centered?

```
i1 <- ceiling(n*runif(1000*n))
X1 <- array(X[i1], c(1000,n))

i2 <- ceiling(n*runif(1000*n))
Y1 <- array(Y[i2], c(1000,n))

C <- array(0, 1000)
for (j in (1:1000)) { C[j] <- cor(X1[j,], Y1[j,]) }
C <- sort(C)

C1 <- C[25]
C2 <- C[975]
```

Solution: The paired data must be resampled as pairs when bootstrapping. Thus a single index vector must be used when resampling. Since two independent index vectors were used here, the resampled data will be independent (between X and Y), with an average sample correlation coefficient of zero. Hence the confidence intervals will tend to be centered around zero, regardless of the population correlation coefficient r .