Stat 732: Statistical Inference: Final exam

Due on April 25, 2014 at 3:10pm

Professor Tokdar

Anh Le

```r
library(mvtnorm)  # For bivariate normal
```

1 Question 1

In order for the 95% interval to have the same frequentist and Bayesian interpretations, we use the reference prior $\pi(\mu, \sigma^2) = {}^1/_{\sigma^2}$.
We reformulate the model as: $X_i = Z_i\mu + \epsilon_i$, with $\epsilon_i \sim N(0, \sigma^2)$, and $Z_i = 1 \forall i$.
With this model formulation, we can apply the result stated in HW 3, problem 1 and get the posterior distribution $\pi(\mu, \sigma^2|X) \sim N\chi^{-2}(\bar{X}, n, n-1, \frac{1}{n-1}\sum(x_i - \bar{X})^2)$.
Steps to generate posterior predictive:

- Generate (μ^*, σ^{2*}) from $\pi(\mu, \sigma^2|X) = N\chi^{-2}(\bar{X}, n, n-1, \frac{1}{n-1}\sum(x_i - \bar{X})^2)$
- Generate x^* from $N(\mu^*, \sigma^{2*})$
- Repeat to get a sample of many x^*

```r
rm(list = ls())
X <- c(45, 73, 100, 105, 112, 115, 120, 120, 123, 130, 150, 150, 150, 150, 159,
    160, 203, 220, 220, 235, 298)
Xbar <- mean(X)
n <- length(X)

# Generate sigma2 and mu
sigma2 <- sum((X - Xbar)^2)/rchisq(10000, df = n - 1)
mu <- rnorm(10000, mean = Xbar, sd = sqrt(sigma2/n))

# Generate samples of posterior predictive
X.star <- rnorm(10000, mean = mu, sd = sqrt(sigma2))

# Get the 95% interval
quantile(X.star, probs = c(0.025, 0.975))

##   2.5%  97.5%
##  24.21 276.50
```

2 Question 2

2.1 Part A

Below is the code to calculate the statistic D:

```r
rm(list = ls())
data <- c(45, 73, 100, 105, 112, 115, 120, 120, 123, 130, 150, 150, 150, 150,
    159, 160, 203, 220, 220, 235, 298)

Z <- (data - mean(data))/sd(data)
Z.star <- unique(Z)

d <- max(abs(sapply(Z.star, function(z.star) pnorm(z.star) - mean(Z <= z.star))))
d

## [1] 0.1905
```

2.2 Part B

The KS test only works for continuous distribution. While expenditure can be thought of as continuous, we do have ties in our data (which is not generated by a continuous distribution).

3 Question 3

3.1 Part A

Under H_0, we have:

$$
\begin{aligned}
f(X|\mu,\sigma^2) &= \prod \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(X_i-\mu)^2}{2\sigma^2}\right) && (1)\\
&= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum(X_i-\bar{X}+\bar{X}-\mu)^2}{2\sigma^2}\right) && (2)\\
&= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum(X_i-\bar{X})^2+\sum(\bar{X}-\mu)^2}{2\sigma^2}\right) && (3)\\
&= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum(n-1)s_X^2+\sum(\bar{X}-\mu)^2}{2\sigma^2}\right) && (4)
\end{aligned}
$$

Therefore, when conditioned on $(\bar{X}, s_X)$, $f(X|\mu,\sigma^2,\bar{X},s_X)$ is simply a constant $\forall X$. Given $X \in \mathbb{R}^n, t = \bar{X}, s = s_X$, we have:

$$
\begin{aligned}
R(t,s) &= \left\{X \in \mathbb{R}^n : \sum X_i = n\bar{X}, \sum(X_i-\bar{X})^2 = (n-1)s_X^2\right\} && (5)\\
&= \left\{X \in \mathbb{R}^n : \frac{1}{n}\sum X_i = \bar{X}, \frac{1}{n-1}\sum(X_i-\bar{X})^2 = s_X^2\right\} && (6)
\end{aligned}
$$

As shown above, $\forall X$ that have mean $\bar{X}$ and sample variance s_X^2 (i.e., $\forall X \in R(t,s)$), the conditional density of X is constant. Therefore, the conditional distribution of X is uniform over $R(t,s)$.

3.2 Part B

$$R(0,1) = \left\{ Z \in \mathbb{R}^n : \sum Z_i = 0, \frac{1}{n-1} \sum (X_i - \bar{X})^2 = 1 \right\} \tag{7}$$

All Z have mean 0 and variance 1 (due to standardization)
$\therefore$ they all belong to $R(t,s)$ by default
$\therefore$ Z is uniformly distributed over $R(0,1) \iff f(Z) = \text{const}\ \forall Z$

To verify that $f(Z) = \text{const}\ \forall Z$ we want to find the distribution $f(Z) = \prod f(Z_i) = \prod f(\frac{X_i - \bar{X}}{s_X})$, then calculate $f(Z)$ from multiple samples of Z based on a grid of different (μ, σ^2). We expect the simulation result to produce similar $f(Z)$.
First, finding $f(Z_i)$. Consider:

$$X_i - \bar{X} = X_i - \frac{1}{n}\sum_{j=1}^{n} X_j = (1 - \frac{1}{n})X_i - \frac{1}{n}\sum_{j \neq i} X_j \tag{8}$$

We see that $X_i - \bar{X}$ is a linear combination of normally distributed variables, thus it is also normally distributed. We want to find its mean and variance.

$$\mathrm{E}(X_i - \bar{X}) = 0 \tag{9}$$

$$\mathrm{Var}(X_i - \bar{X}) = \sigma^2 \left[\left(1 - \frac{1}{n}\right)^2 + \frac{n-1}{n^2} \right] \tag{10}$$

$$= \sigma^2 \left(1 - \frac{1}{n}\right) \tag{11}$$

$$\Rightarrow X_i - \bar{X} \sim N(0, \sigma^2 \left(1 - \frac{1}{n}\right)) \tag{12}$$

$$\Rightarrow \frac{X_i - \bar{X}}{\sigma\sqrt{1 - {}^1\!/_n}} \sim N(0,1) \tag{13}$$

We also know the sampling distribution of sample variance S_X^2 as follows:

$$\frac{(n-1)S_X^2}{\sigma^2} \sim \chi^2_{(n-1)} \tag{14}$$

Therefore,

$$\frac{Z_i}{\sqrt{1 - \frac{1}{n}}} = \frac{\frac{X_i - \bar{X}}{\sigma\sqrt{1 - {}^1\!/_n}}}{\sqrt{\frac{(n-1)S_X^2}{\sigma^2} \Big/ n-1}} \tag{15}$$

$$= \frac{N(0,1)}{\sqrt{\chi^2_{(n-1)} / (n-1)}} \sim t_{(n-1)} \tag{16}$$

Based on this, we have the distribution of $Z_i = t_{(n-1)}(\sqrt{n/n-1})$. The following code will generate samples of $Z = (Z_i, \ldots, Z_n)$ for different values of (μ, σ^2), then compare $f(Z)$.

```
simulation <- function(mean, sd) {
  samples_of_logfZ <- c()
  for (i in 1:1000) {
    # Generate a vector z with 100 z_i
    X <- rnorm(100, mean=mean, sd=sd)
    Z <- (X - mean(X)) / sd(X)
    n <- length(Z)

    # Calculate f(Z)
    logfZ <- sum(log(dt(x=sqrt(n/(n-1))*Z, df=n-1)*sqrt(n/(n-1))))
    samples_of_logfZ <- c(samples_of_logfZ, logfZ)
  }
  # We summarize the "similarity" of 1000 f(Z)
  # by looking at the sample variance
  return(mean( (samples_of_logfZ - mean(samples_of_logfZ))^2 ))
}

# We apply the simulation to a grid of different mu and sigma
finalResult <- apply(expand.grid(seq(-20,20), seq(1,50)), 1,
                     function(tuple) simulation(tuple[1], tuple[2]))

# We report the highest sample variance of 1000 f(Z) (or the least similarity)
# across all different values of mu and sigma
max(finalResult)

## [1] 0.01495
```

The difference is very small between $\log f(Z)$. Thus, we can say that $f(Z) = \text{const} \sim Unif$ over $R(0, 1)$

3.3 Part C

We know that under H_0, Z is uniformly distributed over $R(0, 1)$. Therefore, we can generate N samples of $Z^{(i)}$ and calculate $D^{(i)}$ for each sample.
Then we approximate p-value $= Pr(D > d_{\text{observed}}|H_0) \approx \frac{1}{N}\sum_{i=1}^{N} I(D^{(i)} > d_{\text{observed}})$

3.4 Part D

```
# We have to observed d calculated from question 2a
d

## [1] 0.1905

simulation_3d <- function() {
    X <- sample(21, replace = TRUE)
    Z <- (X - mean(X))/sd(X)
    Z.star <- unique(Z)
    d <- max(abs(sapply(Z.star, function(z.star) pnorm(z.star) - mean(Z <= z.star))))
    return(d)
}

simulated.D <- replicate(1000, simulation_3d())
mean(simulated.D > d)

## [1] 0.065
```

4 Question 4

4.1 Part A

We have:

$$l_X(\mu, \sigma^2, \alpha) = n \log 2 - \frac{n}{2}\log(2\pi) - n\log\sigma + \sum\left(\frac{-(x_i-\mu)^2}{2\sigma^2}\right) + \sum \log\Phi\left(\frac{\alpha(x_i-\mu)}{\sigma}\right) \tag{17}$$

Take partial derivative wrt α, plug in $(\hat{\mu}, \hat{\sigma}^2, \hat{\alpha} = 0)$

$$\begin{aligned}
\frac{\partial}{\partial\alpha} l_X &= \sum \frac{1}{\Phi\left(\frac{\alpha(x_i-\mu)}{\sigma}\right)}\phi\left(\frac{\alpha(x_i-\mu)}{\sigma}\right)\frac{x-\mu}{\sigma} && (18)\\
&= \sum \frac{\phi(0)}{\Phi(0)}\frac{x-\mu}{\sigma} && (19)\\
&= \frac{\phi(0)}{\Phi(0)}\sigma\sum(x-\bar{X}) && (20)\\
&= 0 && (21)
\end{aligned}$$

□

Similarly, take partial derivative wrt μ and σ^2 while plugging in the other estimates. Notice that when $\alpha = 0$ is plugged in, the likelihood is just the regular Normal distribution, therefore $(\hat{\mu}, \hat{\sigma}^2)$ are just MLE of Normal.

Finding $\hat{\mu}$:

$$
\begin{aligned}
\frac{\partial}{\partial \mu} l_X &= \frac{\partial}{\partial \mu}\left[const - \sum \left(\frac{(x_i-\mu)^2}{2\sigma^2}\right)\right] && (22)\\
&= -\frac{1}{2\sigma^2}\sum \frac{\partial}{\partial \mu}(x_i^2 - 2x_i\mu + \mu^2) && (23)\\
&= -\frac{1}{2\sigma^2}\sum(-2x_i + 2\mu) && (24)\\
&= 0 && (25)\\
\Rightarrow \hat{\mu} &= \sum x_i/n = \bar{X} && (26)
\end{aligned}
$$

□

Finding $\hat{\sigma}^2$:

$$
\begin{aligned}
\frac{\partial}{\partial \sigma^2} l_X &= \frac{\partial}{\partial \sigma^2}\left[const - n\log\sigma - \sum \left(\frac{(x_i-\bar{X})^2}{2\sigma^2}\right)\right] && (27)\\
&= \frac{\partial}{\partial \sigma^2}\left[const - \frac{n}{2}\log\sigma^2 - \frac{(n-1)s_X^2}{2\sigma^2}\right] && (28)\\
&= \frac{-n/2}{\sigma^2} + \frac{(n-1)s_X^2}{2\sigma^2} && (29)\\
&= 0 && (30)\\
\Rightarrow \frac{(n-1)s_X^2}{2\hat{\sigma}^2} &= \frac{-n/2}{\hat{\sigma}^2} && (31)\\
\hat{\sigma}^2 &= \frac{n-1}{n}s_X^2 && (32)
\end{aligned}
$$

□

4.2 Part B

We will do a multivariate pdf transformation. Denote $X = (\mu, \sigma^2)^T$ and $Y = (\mu, \xi = \log(\sigma))^T$. We have:

$$
\begin{aligned}
X_1 &= Y_1 && (33)\\
X_2 &= \exp(2Y_2) && (34)\\
D_y &= \begin{pmatrix} 1 & 0 \\ 0 & 2\exp(2Y_2) \end{pmatrix} && (35)
\end{aligned}
$$

Then we can derive the reparameterized prior:

$$
\begin{aligned}
\pi(\mu, \xi) = f_{Y_1,Y_2} &= f_{X_1,X_2}(Y_1, exp(2Y_2)) \times |detD_y| && (36)\\
&= \frac{1}{\exp(2Y_2)} \times 2\exp(2Y_2) && (37)\\
&= 2 && (38)
\end{aligned}
$$

Therefore, $\pi(\theta) \propto q(\alpha)$

4.3 Part C

4.3.1 (i)

To do Laplace approximation of $\pi(\mu, \xi|X, \alpha)$ we numerically optimize $h(\mu, \xi|\alpha) = l_X(\mu, \xi|\alpha) + \log \pi(\mu, \xi|\alpha)$. Since $\pi(\mu, \xi|\alpha)$ is a constant, this is equivalent to maximizing $l_X(\mu, \xi|\alpha)$.
Since we are conditioning on α, we don't have to worry about the optimization process of $l_X(\mu, \xi|\alpha)$ getting stuck on $\alpha = 0$. So good guesses for μ and ξ can be their sample equivalent.

$$\mu^* = \bar{X} \tag{39}$$

$$\xi^* = \log\left(\sqrt{\frac{n-1}{n} s_X^2}\right) \tag{40}$$

4.3.2 (ii)

Since we know the likelihood, prior, and posterior, we can calculate the marginal likelihood using the following trick:

$$f(X|\alpha) = \frac{f(X|\mu^*, \xi^*, \alpha) \times \pi(\mu^*, \xi^*|\alpha)}{\pi(\mu^*, \xi^*|X, \alpha)} \tag{41}$$

$$\approx \frac{SN_X(\mu^*, \xi^*, \alpha) \times 2}{N_{\mu^*, \xi^*}(\hat{m}, V)} \tag{42}$$

4.4 Part D

4.4.1 (i)

We have:

$$\pi(\alpha|X) = \frac{f(X|\alpha) \times \pi(\alpha)}{\sum f(X|\alpha_i) \times \pi(\alpha_i)} \tag{43}$$

$$= \frac{f(X|\alpha)}{\sum f(X|\alpha_i)} \tag{44}$$

Bayes' tail area for $\alpha = 0$ is $2 \times \min(Pr(\alpha > 0|X), Pr(\alpha < 0|X))$, where:

$$Pr(\alpha > 0|X) = \sum_{i:\alpha_i > 0} \pi(\alpha_i|X) \tag{45}$$

$$Pr(\alpha < 0|X) = \sum_{i:\alpha_i < 0} \pi(\alpha_i|X) \tag{46}$$

```
rm(list=ls())
data <- read.csv('food_logf.csv')
data$f <- exp(data$logf)
2*min(with(data, sum(f[alpha < 0]) / sum(f)),
      with(data, sum(f[alpha > 0]) / sum(f)))

## [1] 0.201
```

4.4.2 (ii)

Given $\hat{\alpha} = 2.5$, we want to maximize

$$h(\mu, \sigma^2) = l_X(\mu, \sigma^2|\alpha) = \sum_i \left(\log 2 + \log \text{dnorm}(x_i|\mu, \sigma) + \log \Phi\left(\frac{2.5(x_i - \mu)}{\sigma}\right)\right)$$

```
X <- c(45, 73, 100, 105, 112, 115,
          120, 120, 123, 130, 150, 150,
          150, 150, 159, 160, 203, 220,
          220, 235, 298)
obj.function <- function(argmin) {
  mu <- argmin[1]
  sigma <- argmin[2]
  return(-sum(log(dnorm(X, mean=mu, sd=sigma))) -
    sum(log(pnorm(2.5*(X-mu)/sigma))))
}

mu.start <- mean(X)
sigma.start <- sqrt( sum((X - mean(X))^2) / length(X))
argmin <- nlm(obj.function, c(mu.start, sigma.start), hessian=TRUE)

# mu hat and sigma hat
argmin$estimate

## [1] 87.91 83.76
```

The maximum *a posteriori* estimate $(\hat{\mu}, \hat{\sigma}, \hat{\alpha}) = (87.91, 83.76, 2.5)$
Bonus Power of the K-S test $= Pr(D > d|\hat{\mu}, \hat{\sigma}^2, \hat{\alpha})$. We can calculate power by simulation as follows.

```r
fun_genSN <- function(row) {
    mu <- row[1]
    sigma <- row[2]
    alpha <- row[3]
    Z1 <- rnorm(1)
    Z2 <- rnorm(1)
    return(mu + sigma * Z1 * sign(alpha * Z1 - Z2))
}

fun_genSample <- function(parameters) {
    replicate(21, fun_genSN(parameters))
}

# 1000 columns, each is a sample of 21 X
simulated.X <- replicate(1000, fun_genSample(c(argmin$estimate, 2.5)), simplify = "matri
scaled.X <- apply(simulated.X, 2, function(col) scale(col))

fun_findKSstatistic <- function(col) {
    Z.star <- unique(col)
    d <- max(abs(sapply(Z.star, function(z.star) pnorm(z.star) - mean(col <=
        z.star))))
    return(d)
}

simulated.D <- apply(scaled.X, 2, fun_findKSstatistic)

# Observed d is 0.1905
mean(simulated.D > 0.1905)

## [1] 0.106
```

4.4.3 (iii)

$$R = \frac{q(\alpha = 0|X)}{q(\alpha = 0)} \tag{47}$$

$$= q(\alpha = 0|X) \tag{48}$$

$$= \frac{f(X|\alpha = 0)}{\sum f(X|\alpha_i)} \tag{49}$$

```
with(data, f[alpha == 0]/sum(f))

## [1] 0.03538
```

4.5 Part E

Steps to generate posterior predictive sample:

- Generate (μ^*, σ^*) from $\pi(\mu, \sigma|X, \alpha) = N_2(\hat{m}, V)$
- Generate X_{n+1} from $SN(\mu^*, \sigma^*, \hat{\alpha})$

Bonus: The code below gives a numerical evaluation of the interval:

```
post.parameter <- rmvnorm(n=10000,
  mean=argmin$estimate, sigma=solve(argmin$hessian))
post.parameter <- cbind(post.parameter, 2.5)

post.pred.sample <- apply(post.parameter, 1, fun_genSN)
quantile(post.pred.sample, probs=c(0.025, 0.975))

##  2.5% 97.5%
## 118.6 281.2
```

4.6 Part F

We calculate the Bayes factor by computing the marginal likelihood under M_0 and M_1 as follows:

$$\log BF_{01}(X) = \log f(X|M_0) - \log f(X|M_1) \tag{50}$$

$$\log f(X|M_0) = \sum \log f(X_i|\mu^*, \sigma^{2*}) + \log \pi(\mu^*, \sigma^{2*}) - \log \pi(\mu^*, \sigma^2|X) \tag{51}$$

$$= \sum \log N_{X_i}(\mu^*, \sigma^{2*}) + \log(\frac{c}{\sigma^{2*}}) - \log N\chi^{-2}_{\mu^*,\sigma^*}(\bar{X}, n, n-1, \frac{1}{n-1}\sum(x_i - \bar{X})^2) \tag{52}$$

$$\log f(X|M_1) = \sum \log f(X_i|\mu^*, \sigma^{2*}, \alpha^*) + \log \pi(\mu^*, \sigma^{2*}, \alpha^*) - \log \pi(\mu^*, \sigma^{2*}, \alpha^*|X) \tag{53}$$

$$\tag{54}$$

Use $\alpha^* = 2.5$

$$\tag{55}$$

$$= \sum \log SN_{X_i}(\mu^*, \sigma^{2*}, \alpha = 2.5) + \log(\frac{ck}{\sigma^{2*}}) - \log N_{\mu^*,\sigma^*}(\hat{m}, V) \tag{56}$$

```
X <- c(45, 73, 100, 105, 112, 115,
          120, 120, 123, 130, 150, 150,
          150, 150, 159, 160, 203, 220,
          220, 235, 298)
mu.star <- mean(X)
sigma.star <- sd(X)
n <- length(X)

# Function to find the density of normal inv chisq
dnorminvchisq <- function(mu, sigma2, m, k, r, s, p = 1) {
  (r*s^2 / 2)^(r/2) / gamma(r/2) *
    (2*pi)^(-p/2) * (sigma2/k)^(-1/2) *
    sigma2^(-(r+p+2)/2) * exp( -((mu - m)^2 * k + r*s^2) / (2*sigma2) )
}

lf0 <- sum(log(dnorm(X, mu.star, sigma.star))) -
  log(dnorminvchisq(mu.star, sigma.star^2, mean(X), n, n-1, sd(X)))
lf1 <- n*log(2) + sum(log(dnorm(X, mu.star, sigma.star))) +
  sum(log(pnorm(2.5*(X-mu.star)/sigma.star))) -
  log(dmvnorm(c(mu.star,sigma.star),
              mean=argmin$estimate, sigma=solve(argmin$hessian)))

bf01 <- exp(lf0 - lf1)
bf01

## [1] 2.139e+11
```