Problem 1: This code generates 2000 sets of 5 variables, takes the sample mean of those 5 variables, and reports an estimate of the variance of the sample means at the end. Each row has the structure that $X_1 \sim N(0,5)$, and $X_2, ..., X_5 \sim N(0,1)$. Therefore,

$$\begin{aligned} \text{var}\left(\frac{1}{n}(X_1 + \sum_{i=2}^{5} X_i)\right) &= \frac{1}{n^2}\left(\text{var}(X_1) + \sum_{i=2}^{5} \text{var}(X_i)\right) \\ &= \frac{1}{n^2}\left(5+1+1+1+1\right) \\ &= \frac{9}{25} = .36 \end{aligned}$$

So entry entry of `M` will be a $N(0, .36)$ random variable, and answer (4) will be the approximate output.

Problem 2: This program generates 1000 pairs of datasets of size $n = 20$. One of the datasets are iid $N(1,1)$ and the other are iid $N(0,1)$. A 95% confidence interval for $\mu_1 - \mu_2$ is formed for each of the 1000 datasets and `I` stores whether or not the confidence interval contained the true value for $\mu_1 - \mu_2$, which is 1 in this case. At the end the proportion of these confidence intervals that contained 1 is printed, which should be approximately .95.

Problem 3: First of all, this is an exactly quadratic function, so Newton-Raphson should converge to an extrema after a single iteration. Since this is a strictly non-positive function, 0 is clearly an extrema which is attained at $x = k$, which is where the algorithm should be after 1 iteration. This can be verified explicitly as well, since

$$f'(x) = -2(x-k)$$

and

$$f''(x) = -2$$

So, starting from x_0, the next iteration takes you to

$$x_0 - \frac{2(x_0 - k)}{2} = x_0 - (x_0 - k) = k$$

Problem 4: Bisection begins by forming two new intervals based on splitting the bracket at the midpoint. So the two candidates for a new bracket are $(1/2, 3/4)$ and $(3/4, 1)$. From the information given in the problem there is not a sign change in f over the interval $(1/2, 3/4)$ and there is a sign change over the interval $(3/4, 1)$. Therefore, the new bracket will be $(3/4, 1)$.

Problem 5: Following from bilinearity of covariance,

$$\text{cov}(X+Y, X+Z) = \text{cov}(X,X) + \text{cov}(X,Z) + \text{cov}(Y,X) + \text{cov}(Y,Z) = \text{var}(X) = \lambda_1$$

which follows form the independence of X, Y, Z. A short R program to estimate this covariance, for particular values of λ_1, λ_2, denoted by `L1,L2` is

```
X <- rpois(1000, L1)
Y <- rpois(1000, L1)
Z <- rpois(1000, L2)
cov(X+Y,X+Z)
```

Problem 6: The code here estimates $\text{cov}(X, |X|)$, where $X \sim N(0,1)$. The output should then be around $\text{cov}(X, |X|)$, which is

$$\text{cov}(X, |X|) = E(X \cdot |X|)$$

since X has mean 0. This can be calculated by using the law of total expectation, conditioning on whether or not X is positive:

$$\begin{aligned} E(X \cdot |X|) &= \frac{1}{2}\left(E(X \cdot |X| \mid X < 0) + E(X \cdot |X| \mid X > 0)\right) \\ &= \frac{1}{2}\left(-E(X^2) + E(X^2)\right) \\ &= 0 \end{aligned}$$

this is also intuitive because large values of $|X|$ can either be associated with very large values of X or very small values of X, so there couldn't possibly be any linear association between $|X|$ and X. So the approximate output should be 0. Notice this is an example of when uncorrelated does not imply independent.

Problem 7: For the first two blanks, you need to enter the R code to calculate the two estimators, which would be

```
1/mean(X)
```

and

```
sqrt(1/var(X))
```

respectively. The MSE is the average squared difference of the estimator of from the truth, so the second two blanks need to be filled in by the true value for λ, which is `1` in this case.

Problem 8: No, this does not mean that $\hat{\theta}_1$ will necessarily outperform $\hat{\theta}_2$ in terms of MSE, since

$$MSE(\hat{\theta}) = \text{bias}(\hat{\theta})^2 + \text{var}(\hat{\theta})$$

So, if $\hat{\theta}_2$ had substantially lower variance, then it could outperform $\hat{\theta}_1$ in terms of MSE.

Problem 9: We need to calculate the log-likelihood and its first two derivatives for this problem. The log-likelihood is

$$\ell(\theta) = \log(p_\theta(x)) = \text{const} + \frac{1}{2}\log(\theta) - \frac{\theta}{2x}$$

so the derivative of the log-likelihood (the score function) is

$$\ell'(\theta) = \frac{1}{2\theta} - \frac{1}{2x}$$

and the second derivative is

$$\ell''(\theta) = -\frac{1}{\theta}$$

So the first iteration of Newton Raphson will take you to

$$\begin{aligned}
\theta_1 &= \theta_0 - \frac{\ell'(\theta_0)}{\ell''(\theta_0)} \\
&= \theta_0 + \frac{\frac{1}{2\theta_0} - \frac{1}{2x}}{\frac{1}{\theta_0}} \\
&= \theta_0 + \frac{1}{2} - \frac{\theta_0}{2x} \\
&= .1 + \frac{1}{2} - \frac{.1}{2x} \\
&= .6 - \frac{.1}{2x}
\end{aligned}$$

Problem 10: For the inversion method the first step is to calculate the CDF:

$$F(x) = \int_0^x \sin(y)dy = 1 - \cos(x)$$

It is straightforward to invert this function and see that

$$F^{-1}(y) = \cos^{-1}(1 - y)$$

so the R code to implement this is

```
rSin <- function(n) acos(1 - runif(n))
```

To do the rejection sampling optimally, we first need to determine

$$M = \max_{x \in (0,\pi/2)} \frac{p(x)}{g(x)} = \max_{x \in (0,\pi/2)} \frac{\pi \sin(x)}{2}$$

Since $\sin(x)$ is always at most 1, and $\sin(\pi/2) = 1$, this implies that

$$M = \pi/2$$

So $\frac{p(X)}{Mg(X)} = p(X) = \sin(X)$ an optimal rejection sampling algorithm will accept a candidate draw, $X \sim g$, when

$$U \leq \sin(X)$$

where U is Uniform(0,1). The optimal acceptance rate for rejection sampling is $1/M$, which is this case is $2/\pi \approx .636$.

Problem 11: This integral must evaluate to 0, because sinh(x) is an odd function– that is, $\sinh(x) = -\sinh(-x)$, and we are integrating over a region that is symmetric around 0, so the area to the left of 0 and the area to the right of 0 must cancel out.

This integral can be re-written as

$$I = 2\pi \int_{-\pi}^{\pi} \sinh(x) p(x) dx$$

where $p(x) = 1/2\pi$ is the Uniform($-\pi$,π) density. So,

$$I = 2\pi E(\sinh(U))$$

where $U \sim$ Uniform$(-\pi, \pi)$. This expectation can be estimated by monte carlo by appealing to the law of large numbers. The following program does this:

```
U <- runif(1e4, -pi, pi)
2*pi*mean( sinh(U) )
```

Problem 12: Using the hint, if $X \sim p$, we can think of X in terms of the following algorithm:

1. Generate $U \sim$ Uniform$(0, 1)$
2. If $U \leq .5$, then generate $X \sim N(-2, 1/2)$
3. If $U > .5$, generate $X \sim N(2, 1/2)$.

The following code does this for a pre-specified sample size, `n`:

```
U <- runif(n)
X <- rep(0,n)
w <- which(U <= .5)
X[w] <- rnorm(length(w), mean=-2, sd=sqrt(1/2))
X[-w] <- rnorm(n-length(w), mean=2, sd=sqrt(1/2))
```

Problem 13: Similarly to problem 11, this can be written as an expectation against the uniform(0,1) distribution:

$$E(\sin(e^{1+U^2}))$$

Using $g(x; \alpha, \beta)$, the beta(α, β) density, as the trial density, define the importance function to be $w(x) = p(x)/g(x) = 1/g(x)$. Sample from $X \sim g$, and then estimate $E(w(X) \cdot \sin(e^{1+X^2}))$ by appealing to the law of large numbers. The following R program does this:

```
IS <- function(k, a, b)
{
f <- function(x) sin(exp(1+x^2))
p <- function(x) dunif(x)
g <- function(x) dbeta(x,a,b)
w <- function(x) p(x)/g(x)
X <- rbeta(k, a, b)
Q <- w(X)*f(X)
return( c(mean(Q), var(Q)/k) )
}
```

We would choose α, β (`a,b` in the above program) by calculating `IS(k,a,b)` for a fixed `k` (say 10000), and comparing the standard error of the resulting estimate. The combination of α, β that corresponds to the smallest standard error would be considered the optimal choice.

Problem 14: Denoting the $N(0, \mu^2)$ density by $\phi_{\sigma^2}(x)$, this integral can be re-written as

$$\int_{-\infty}^{\infty} x^2 \frac{p(x)}{\phi_{\sigma^2}(x)} \phi_{\sigma^2}(x) dx$$

Letting $w(x) = \frac{p(x)}{\phi_{\sigma^2}(x)}$, this integral is the same as

$$E(X^2 w(X))$$

where $X \sim N(0, \sigma^2)$. This expectation can be estimated by appealing to the law of large numbers. The following R code carries out the importance sampling:

```
IS <- function(k, v)
{
   f <- function(x) x^2
   p <- function(x) exp(-x)/( (1+exp(-x))^2 )
   g <- function(x) dnorm(x,mean=0,sd=sqrt(v))
   w <- function(x) p(x)/g(x)
   X <- rnorm(k, mean=0, sd=sqrt(v))
   Q <- w(X)*f(X)
   return( c(mean(Q), var(Q)/k) )
}
```

To choose the optimal σ^2, you would call `IS` for some fixed `k` (say, 10000), and a grid of σ^2 values (say .01 up to 10, by .1), and see which value minimizes the variance of the integral approximation.

Problem 15: The partial derivatives of $f(x, y)$ are

$$\partial f / \partial x = (2x - 4y) f(x, y)$$

and

$$\partial f / \partial y = (2y - 4x) f(x, y)$$

The second derivatives are

$$\partial^2 f / \partial x^2 = (2 + (2x - 4y)^2) f(x, y),$$

$$\partial^2 f / \partial y^2 = (2 + (2y - 4x)^2) f(x, y),$$

and

$$\partial^2 f / \partial x \partial y = (-4 + (2x - 4y)(2y - 4x)) f(x, y)$$

plugging into $x = 1, y = 1$ in accordance with the start values, the gradient at x_0 is

$$\nabla f(x_0) = (-2/e, -2/e)$$

and the hessian matrix at x_0 is

$$\nabla^2 f(x_0) = \begin{pmatrix} 6/e & 0 \\ 0 & 6/e \end{pmatrix}$$

The determinant of this matrix is $36/e^2$, so the inverse hessian is

$$\left(\nabla^2 f(x_0)\right)^{-1} = \frac{e^2}{36} \begin{pmatrix} 6/e & 0 \\ 0 & 6/e \end{pmatrix} = \begin{pmatrix} e/6 & 0 \\ 0 & e/6 \end{pmatrix}$$

So the first iteration of Newton-Raphon will move you to

$$\begin{aligned} x_0 - \left(\nabla^2 f(x_0)\right)^{-1} \nabla f(x_0) &= x_0 - \begin{pmatrix} e/6 & 0 \\ 0 & e/6 \end{pmatrix} (-2/e, -2/e) \\ &= (1,1) + \Big((2/e) \cdot (e/6) + 0 \cdot 2/e, 0 \cdot 2/e + (2/e) \cdot (e/6)\Big) \\ &= (1,1) + (1/3, 1/3) \\ &= (4/3, 4/3) \end{aligned}$$

Problem 16: Remember the posterior is equal to the likelihood multiplied by the prior. To begin, the likelihood is nothing but the joint density, so

$$\begin{aligned} L(\sigma^2) &= \prod_{i=1}^{n} p(x_i|\sigma^2) \\ &= \prod_{i=1}^{n} (2\pi\sigma^2)^{1/2} \cdot e^{-x_i^2/2\sigma^2} \\ &= (2\pi\sigma^2)^{n/2} \text{exp} \left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} x_i^2 \right) \\ &\propto \sigma^n \text{exp} \left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} x_i^2 \right) \end{aligned}$$

In this case the exponential density is the prior, so it is equal to

$$p(\sigma^2) = \lambda e^{-\lambda\sigma^2}$$

Therefore

$$
\begin{aligned}
p(\sigma^2|x_1,...,x_n) &= L(\sigma^2)\cdot p(\sigma^2)\\
&= L(\sigma^2)\cdot \lambda e^{-\lambda\sigma^2}\\
&\propto \sigma^n \text{exp}\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n} x_i^2\right)\cdot \lambda e^{-\lambda\sigma^2}\\
&= \lambda\sigma^n \text{exp}\left(-\lambda\sigma^2-\frac{1}{2\sigma^2}\sum_{i=1}^{n} x_i^2\right)
\end{aligned}
$$

is something proportional to the posterior distribution. Letting $g(\sigma^2)$ be the gamma(α, β) density and letting post(σ^2) be the expression proportional to the posterior distribution, we would begin by determining

$$M = \max_{\sigma^2>0} \frac{\text{post}(\sigma^2)}{g(\sigma^2)}$$

Next you can generate a draw from post(σ^2) by

1. Generate U from the Uniform(0,1) distribution
2. Generate Y from the gamma(α, β) distribution
3. If $U \leq \frac{\text{post}(Y)}{M\cdot g(Y)}$ then accept Y as a draw from post(σ^2)

Repeat this process until the desired sample size is achieved.