Stat Computing HW3

Yaniv Bronshtein

4/9/2022

Library imports

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.5     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.1.1     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Problem 1.

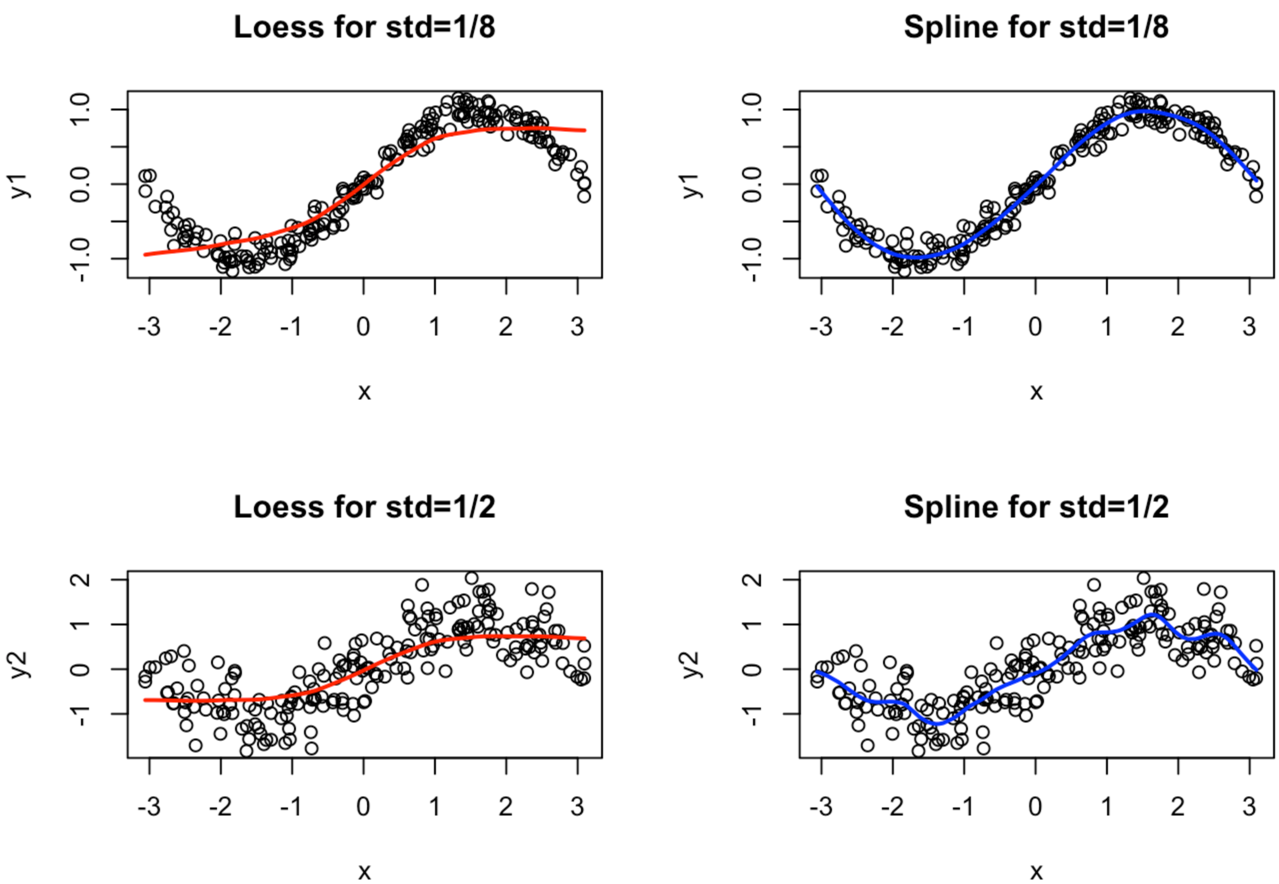
a. Generate 200 replicas of uniform [-pi,pi] and 200 normal with mean 0 and standard deviation 1/8. Set data x from this uniform, error epsilon from this normal distribution. The response y is by model: y = sin(x)+epsilon Fit the data with two types of smoothing techniques. Plot both the data and your fitted smooth curves. (b) The same as (a) except changing the standard deviation from 1/8 to 1/2. [Remark: Use a computer for you calculation; explain your analysis and results carefully]

```
set.seed(1)
x <- runif(n=200,min=-pi,max=pi)
e1 <- rnorm(n=200,mean=0,sd=(1/8))
y1 <- sin(x) + e1
```

```
set.seed(1)
e2 <- rnorm(n=200,mean=0,sd=(1/2))
y2 <- sin(x) + e2
```

Fit the data with two smoothing techniques

```
par(mfrow=c(2,2))
plot(x,y1,main="Loess for std=1/8")
lines(loess.smooth(x,y1),lwd=2,col='red')

plot(x,y1,main="Spline for std=1/8")
lines(smooth.spline(x,y1),lwd=2,col='blue')

plot(x,y2,main="Loess for std=1/2")
lines(loess.smooth(x,y2),lwd=2,col='red')

plot(x,y2,main="Spline for std=1/2")
lines(smooth.spline(x,y2),lwd=2,col='blue')
```



The above graphs show that the four graphs show a fit to the non-linear data using both a smoothing method and a kernal method. These are two of the more popular fitting tool with two graphs using standard deviation of 1/2 and 1/8 for each method. The fit is similiar for the two standard deviations but we see that the lower standard deviation is slightly less smooth and the higher std.dev is more smooth.

Based on the results, a spline fitting the data for error standard deviation of 1/8 provides the best fit. Loess provides the poorest fit for both values of standard deviation. Despite the difficulty of fitting for std=1/2, the spline consistently performs the better fit for both standard deviations

Problem 2.

a. Use a linear regression model to analyze the GAG in urine data in data frame GAGurine. Produce a chart to help a pediatrician to assess if a child's GAG concentration is 'normal or not (hint: plot in one graph the estimated line and confidence bands at different levels)
b. Consider using a smooth regression to analyze the GAG in urine data [Remark: See the data set named "GAGurine.csv" in the assignment. Use a computer for you calculation; explain your analysis and results carefully]

Read in the data

```
gag_urine_df <- read.csv('/Users/yanivbronshtein/Coding/Rutgers/Statistical_Computing
_Repo/data/GAGurine.csv')
```

Fit linear regression model

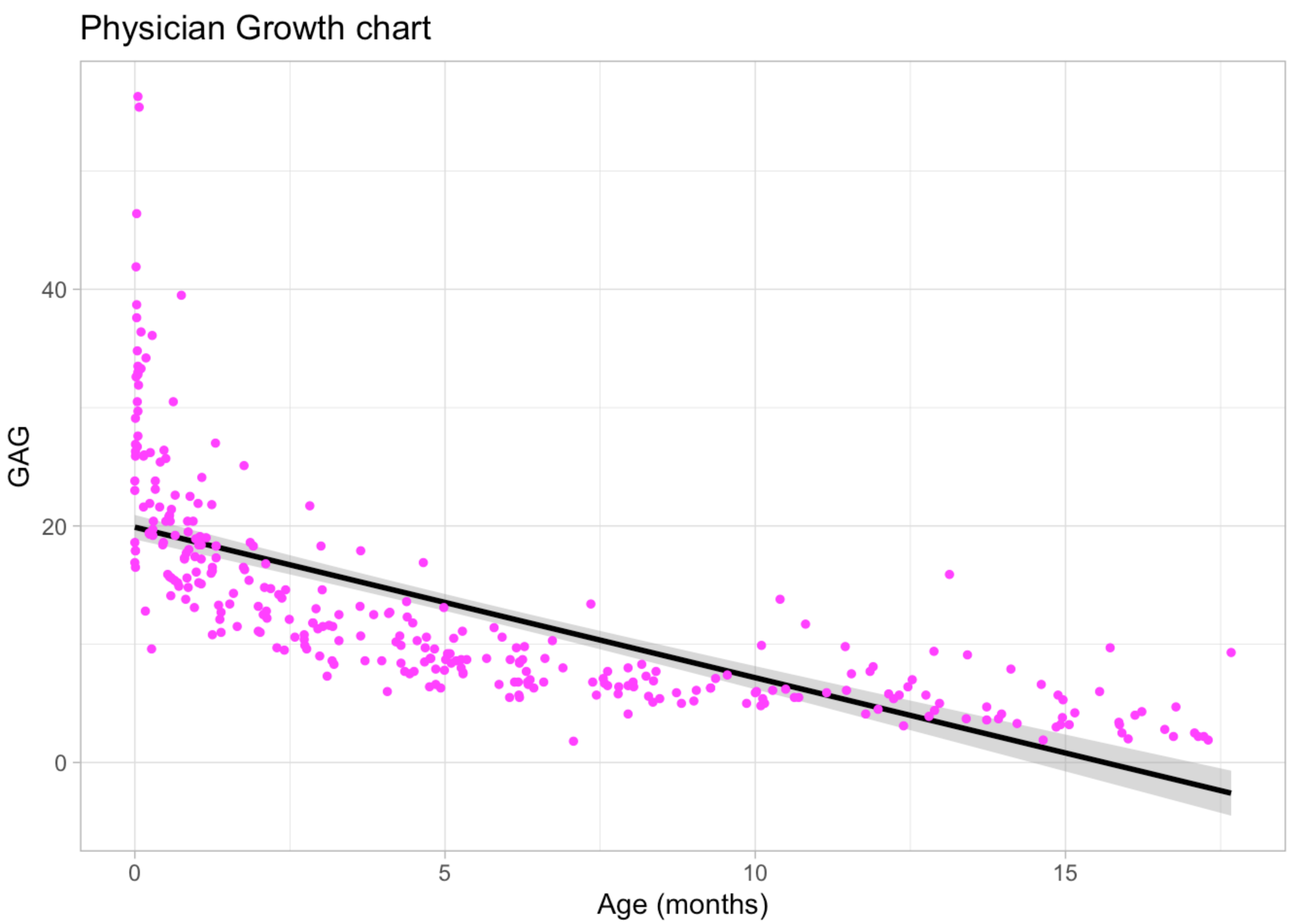
```
gag_lm <- lm(GAG~Age, data=gag_urine_df)
summary(gag_lm)
```

```
##
## Call:
## lm(formula = GAG ~ Age, data = gag_urine_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.950 -4.217 -1.596  2.477 36.470
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.89381    0.52553   37.85   <2e-16 ***
## Age         -1.27253    0.07242  -17.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.386 on 312 degrees of freedom
## Multiple R-squared:  0.4974, Adjusted R-squared:  0.4958
## F-statistic: 308.7 on 1 and 312 DF,  p-value: < 2.2e-16
```

**Generate the physician growth chart

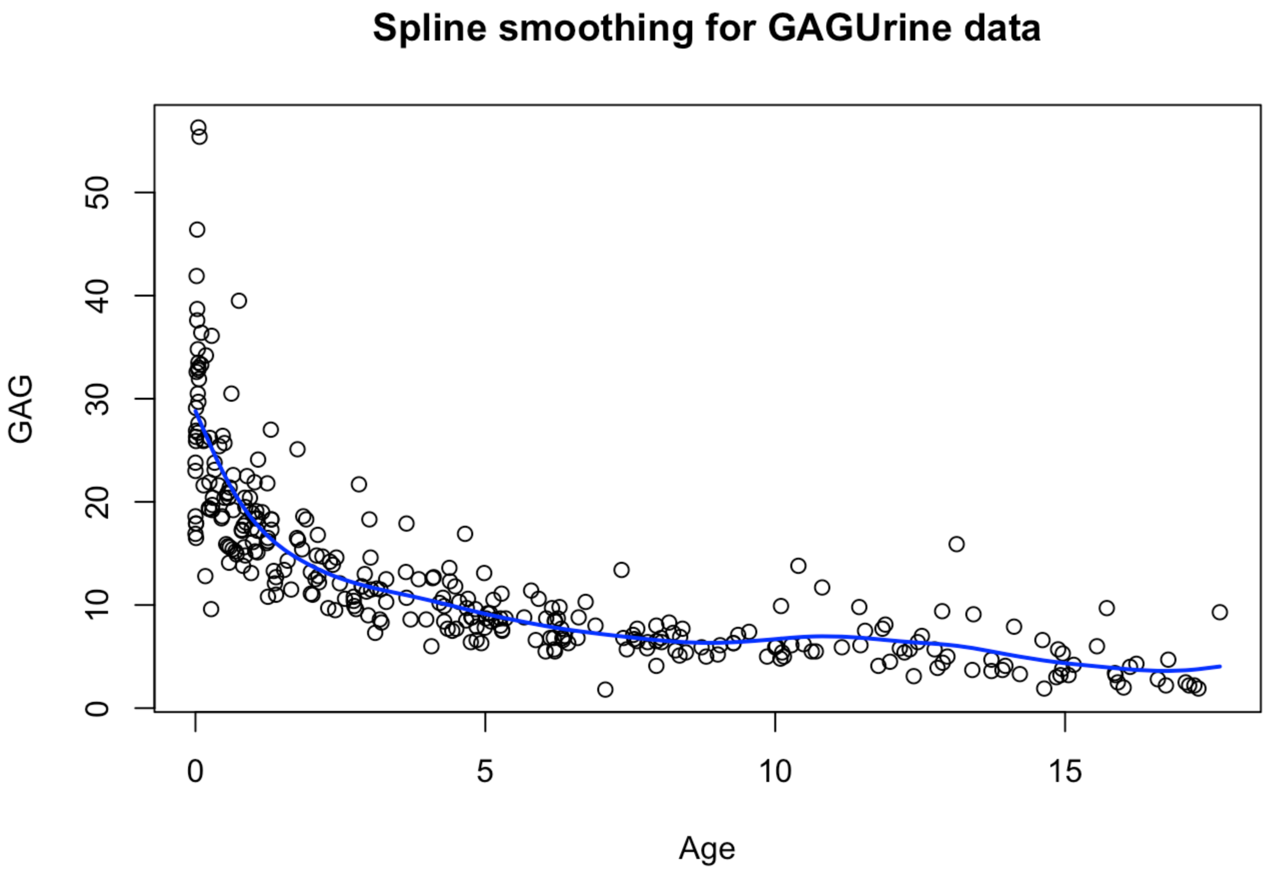
```
GAG_plot <- ggplot(gag_urine_df, aes(Age, GAG)) +
    geom_smooth(method = "lm", se = TRUE, col = "black") +
    geom_point(size = 1, col = "magenta") +
    labs(x = "Age (months)", y = "GAG") +
    ggtitle("Physician Growth chart") +
    theme_light()

GAG_plot
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The physician chart above shows that the data is poorly fit with a linear model. Most of the observations are not on the regression curve. **Fit the spline for the GAGURine data**

```
x <- gag_urine_df$Age
y <- gag_urine_df$GAG
plot(x=x,
     y=y,
     main="Spline smoothing for GAGUrine data",
     xlab="Age",
     ylab="GAG")
lines(smooth.spline(x,y),lwd=2,col='blue')
```



Problem 4

Write a computing code to calculate the integration from -5 to 5 of (x^3 – x$^{2)\exp\{-x}$2/2} using Monte Carlo simulation with N samples from a uniform distribution, for N = 10, 100, 1000. For each choice of N, repeat the experiment for 500 times, compute the variance and visualize the relationship between the variance and N. [Remark: Use a computer for you calculation; explain your results carefully]

```
# importing the modules


# limits of integration
a = -5
b = 5
N = 100

arr <- runif(N,a,b)


# variable to store sum of the functions of
# different values of x
integral <- 0.0

# function to calculate the sin of a particular
# value of x
f <- function(x){
  exponent <- -(x^2)/2
  return((x^3 - x^2)*exp(exponent))
}
# iterates and sums up values of different functions
# of x
for(elem in arr){
    integral <- integral + f(elem)
}
cat(integral)
```

```
## -20.34422
```

```
# we get the answer by the formula derived adobe
ans = 1.0*(b-a)/N*integral

# prints the solution
cat("The value calculated by monte carlo integration is.",ans)
```

```
## The value calculated by monte carlo integration is. -2.034422
```

Function to integrate

```
f <- function(x){
    exponent <- -(x^2)/2
    return((x^3 - x^2)*exp(exponent))
}
```

Monte Carlo integration

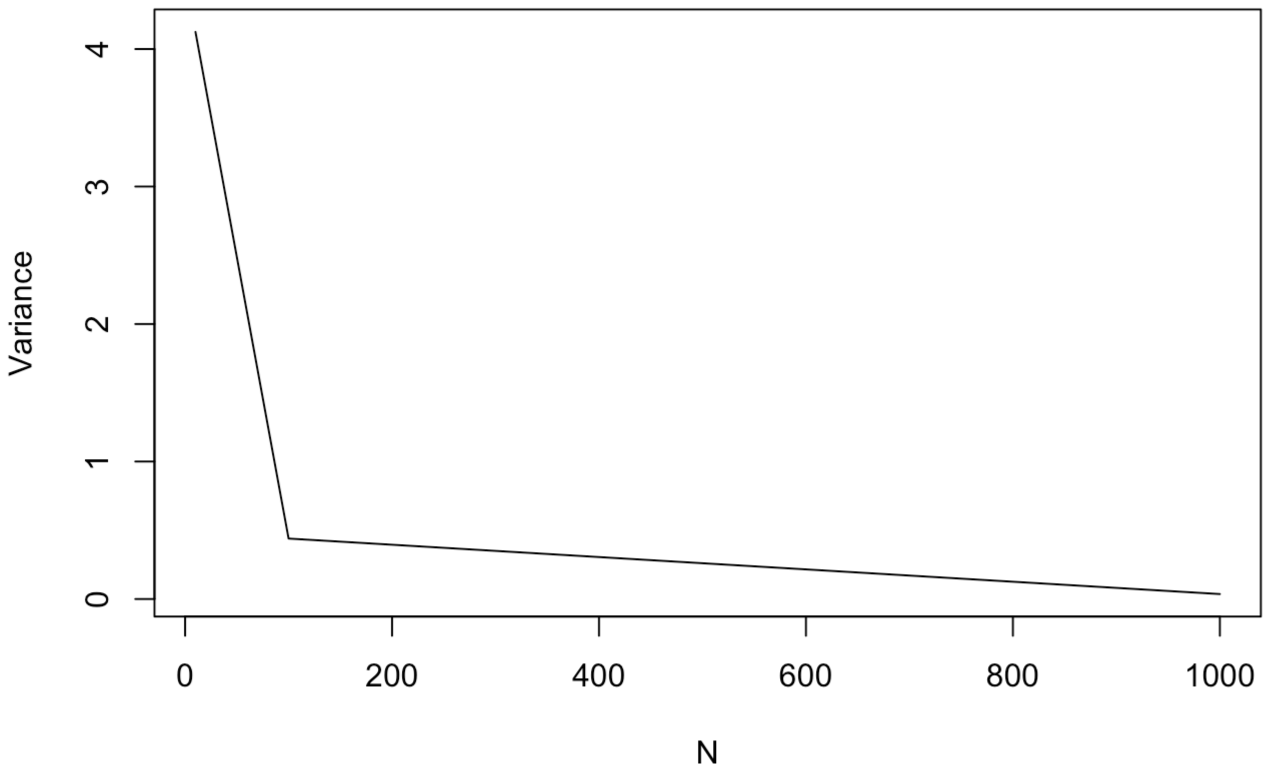
```
monte_carlo_integral <- function(N){
  a <- -5
  b <- 5
  arr <- runif(N,a,b)


  # variable to store sum of the functions of
  # different values of x
  integral <- 0.0


  # iterates and sums up values of different functions
  # of x
  for(elem in arr){
      integral <- integral + f(elem)
  }
  # we get the answer by the formula derived adobe
  ans = (b-a)/N*integral

  return(ans)
}
```

Compute the variances after 500 simulations

```
arr_N10 <- NULL; arr_N100 <- NULL; arr_N1000 <- NULL
for(i in 1:500) {
  arr_N10 <- c(arr_N10, monte_carlo_integral(10))
  arr_N100 <- c(arr_N100, monte_carlo_integral(100))
  arr_N1000 <- c(arr_N1000, monte_carlo_integral(1000))

}

var_N10 <- var(arr_N10)
var_N100 <- var(arr_N100)
var_N1000 <- var(arr_N1000)
```

Plot the variance

```
N <- c(10,100,1000)
Variance <- c(var_N10, var_N100, var_N1000)

plot(x=N, y=Variance, type='l')
```



Over 500 iterations, we compute the variance estimates and utilize the uniform distribution in our monte carlo simulation. As the value of N increases, the variance decreases and the accuracy of the integration value increases(verified using wolfram alpha).

Problem 3. Service times of a queuing system follow Exponential Distribution with an unknown parameter θ.
A sample of service times $X_1, X_2, \ldots, X_n$ is observed.

(a). Show that the Gamma(α, λ) family of prior distributions is conjugate.

Let $\theta \sim$ Gamma (α, λ) so that we can define our prior as follows:

$$f_\Theta(\theta) = \frac{\lambda^\alpha \theta^{\alpha-1} e^{-\lambda\theta}}{\Gamma(\alpha)}$$

The data given the parameter θ have an exponential distribution, so $X_i | \theta \sim \exp(\theta)$

$$f_{X_i|\Theta}(x_i|\theta) = \theta e^{-\theta x_i}$$

We write the joint distribution for n independent, identically distributed samples by:

$$f(X|\theta) = \theta^n e^{-\theta \sum x_i}$$

The posterior is proportional to the product of the likelihood and the prior.

So

$$f_{\Theta|X}(\theta|x) \propto f_{X|\Theta}(x|\theta) * f_\Theta(\theta)$$

$$\propto \theta^n e^{-\theta \Sigma x_i} * \frac{\lambda^\alpha \theta^{\alpha-1} e^{-\lambda\theta}}{\Gamma(\alpha)}$$

$$\propto \theta^{n+\alpha-1} e^{-\theta(\Sigma x_i + \lambda)} \frac{\lambda^\alpha}{\Gamma(\alpha)} \leftarrow \text{Omit}$$

$$\propto \theta^{n+\alpha-1} e^{-\theta(\Sigma x_i + \lambda)}$$

The pdf of a Gamma Distribution is:

$$f(x;\alpha;\lambda) = \frac{x^{\alpha-1} e^{-\lambda x} \lambda^\alpha}{\Gamma(\alpha)} \quad \text{for } x > 0 \quad \alpha, \lambda > 0.$$

$$f(\theta;\alpha;\lambda) = \frac{\theta^{\alpha-1} e^{-\lambda\theta} \lambda^\alpha}{\Gamma(\alpha)} \approx \theta^{\alpha-1} e^{-\theta(\lambda)}$$

So the λ parameter is $\Sigma x_i + \lambda$
and α parameter is $n+\alpha$.

Thus,

$$\Theta | X \sim \text{Gamma}(n+\alpha, \Sigma x_i + \lambda)$$

Since the product our prior and likelihood adhered to a Gamma distribution, we conclude that the Gamma family of priors is conjugate ∎

(b). Find the posterior parameters, posterior mean and variance (As functions of θ, α, λ, and X_i's)

From part (a), we have that

$$\theta | X \sim \text{Gamma}(n+\lambda, \Sigma X_i + \lambda)$$

$$\mu = E[\theta|X] = (n+\lambda)(\Sigma X_i + \lambda)$$

$$\sigma^2 = V[\theta|X] = (n+\lambda)(\Sigma X_i + \lambda)^2$$

(c). Suppose that we can allow $\alpha = 0$ and consider a prior density $\pi(\theta) = \frac{1}{\theta}$ for $\theta > 0$. Find the Posterior distribution, its mean, and its variance.

Posterior $\propto$ Likelihood * Prior

$$\propto \theta^n e^{-\theta \Sigma x_i} * \frac{1}{\theta}$$

$$\propto \theta^{n-1} e^{-\theta \Sigma x_i}$$

Now $\theta | X \sim \text{Gamma}(n, \Sigma X_i)$

$$\mu = E[\theta|X] = n * \Sigma X_i$$

$$\sigma^2 = V[\theta|X] = n * (\Sigma X_i)^2$$