

Bayesian Inference and Decision Theory

Unit 4: Introduction to Monte Carlo
Approximation

Learning Objectives for Unit 4

- Describe the Monte Carlo method and explain why it works
- Estimate common aspects of a distribution by sampling from the distribution
 - Expected value
 - Variance and standard deviation
 - Quantiles and credible intervals
 - Other interesting quantities
- Use Monte Carlo to predict future observations conditional on past observations
- Apply posterior predictive sampling to evaluate adequacy of a model and investigate departures from the model



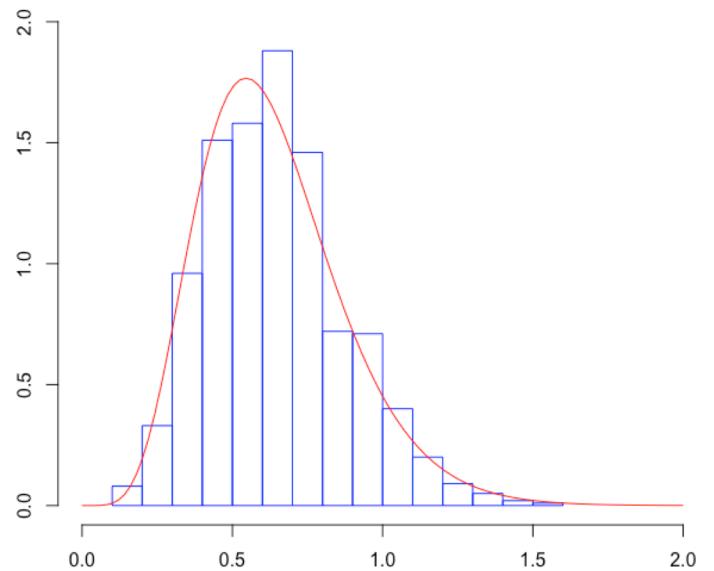
The Monte Carlo Method

- Monte Carlo is a class of computational methods based on random sampling
 - Coined by John von Neumann and Stanislaw Ulam as a code name for secret research conducted during World War II
- Bayesian Monte Carlo uses statistics to do statistics
- Availability of general-purpose Monte Carlo methods is an important reason for increase in popularity of Bayesian methods



How Monte Carlo Works

- Often we wish to use data to answer a question for which no exact answer is available, e.g.:
 - Find a posterior distribution $g(\Theta|X)$ that cannot be computed exactly
 - Estimate some feature of a posterior distribution $g(\Theta|X)$ that cannot be computed exactly
- Sometimes we can draw a sample from $g(\Theta|X)$ or a closely related distribution, and use the sample to find an approximate answer to the question of interest
 - Monte Carlo can give accurate approximations when exact calculations are intractable
- In this module we will examine Direct Monte Carlo (dMC), the simplest Monte Carlo method



Example: Manufacturing Defects

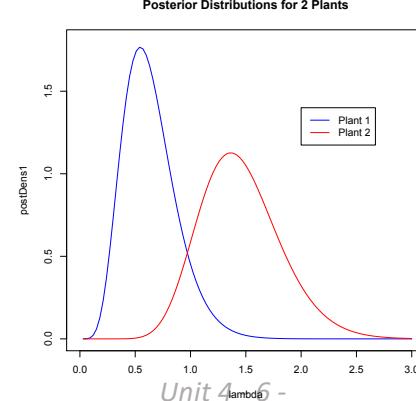
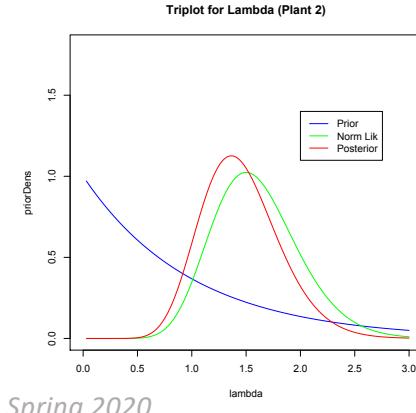
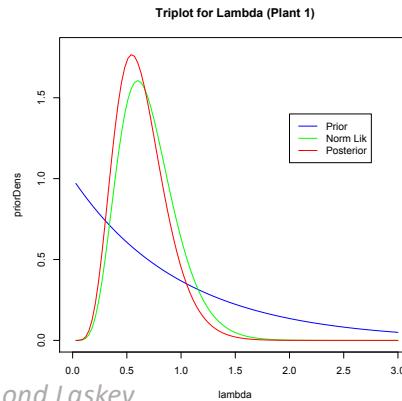
	Plant # 1	Plant # 2
Week 1: M	1	3
Week 1: T	0	0
Week 1: W	0	1
Week 1: H	0	0
Week 1: F	0	1
Week 2: M	0	4
Week 2: T	1	1
Week 2: W	1	2
Week 2: H	1	2
Week 2: F	2	1

- An organization manufactures equipment at two different plants. Each day for a period of two weeks, items from each plant were randomly selected and tested. This table shows the number of defective items in each of these samples.

- Questions of interest:
 - Does one of the plants have a higher defect rate than the other?
 - Is a Poisson distribution with constant rate a good model for the defect counts at each plant?
- We will use the Monte Carlo method to address these questions

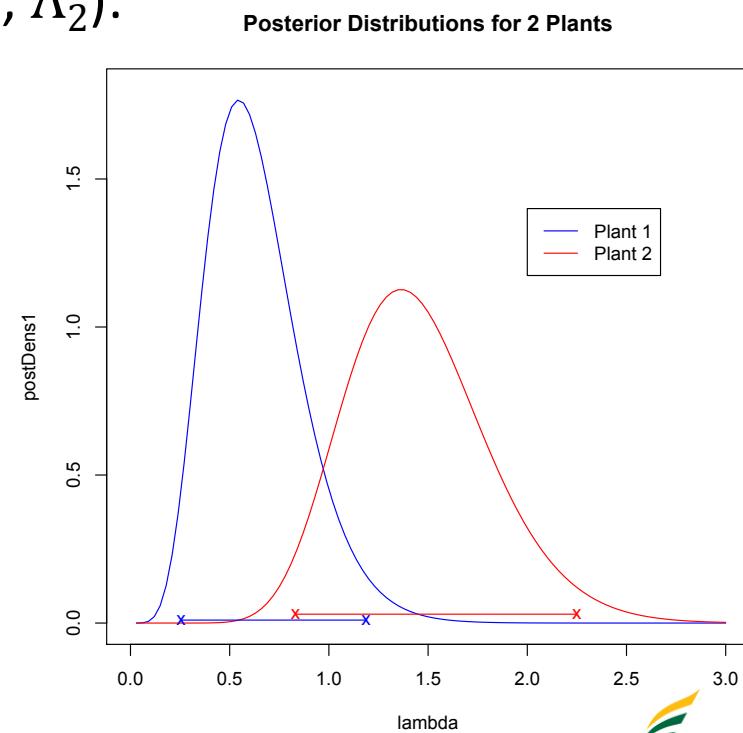
Posterior Distribution for Plant Defect Rates

- Likelihood: Defects at each plant are iid Poisson random variables with plant-specific defect rates Λ_1 and Λ_2
- Joint prior distribution of (Λ_1, Λ_2) : Λ_1 and Λ_2 are independent Gamma(1,1) RVs
 - Expected value 1 defect per day, 90% interval [0.05, 3.0] defects per day
- Observations:
 - Plant 1 has 6 defects in 10 days
 - Plant 2 has 15 defects in 10 days
- Posterior distributions for the two defect rates:
 - Plant 1: Gamma(7, 1/11)
 - Plant 2: Gamma(16, 1/11)



Comparing the Defect Rates

- Joint posterior distribution of defect rates (Λ_1, Λ_2):
 - Λ_1 and Λ_2 are independent
 - $\Lambda_1 | \text{data from Plant 1} \sim \text{Gamma}(7, 1/11)$
 - $\Lambda_2 | \text{data from Plant 2} \sim \text{Gamma}(16, 1/11)$
- Posterior 95% credible intervals
 - Plant 1: [0.26, 1.18]
 - Plant 2: [0.83, 2.25]
 - Credible intervals are shown as x-----x
- Does Plant 2 have a higher defect rate?
 - Finding separate credible intervals for Λ_1 and Λ_2 does not answer this question
 - We need to find $\Pr(\Lambda_2 - \Lambda_1 > 0 | \text{data})$
 - How do we find this probability?

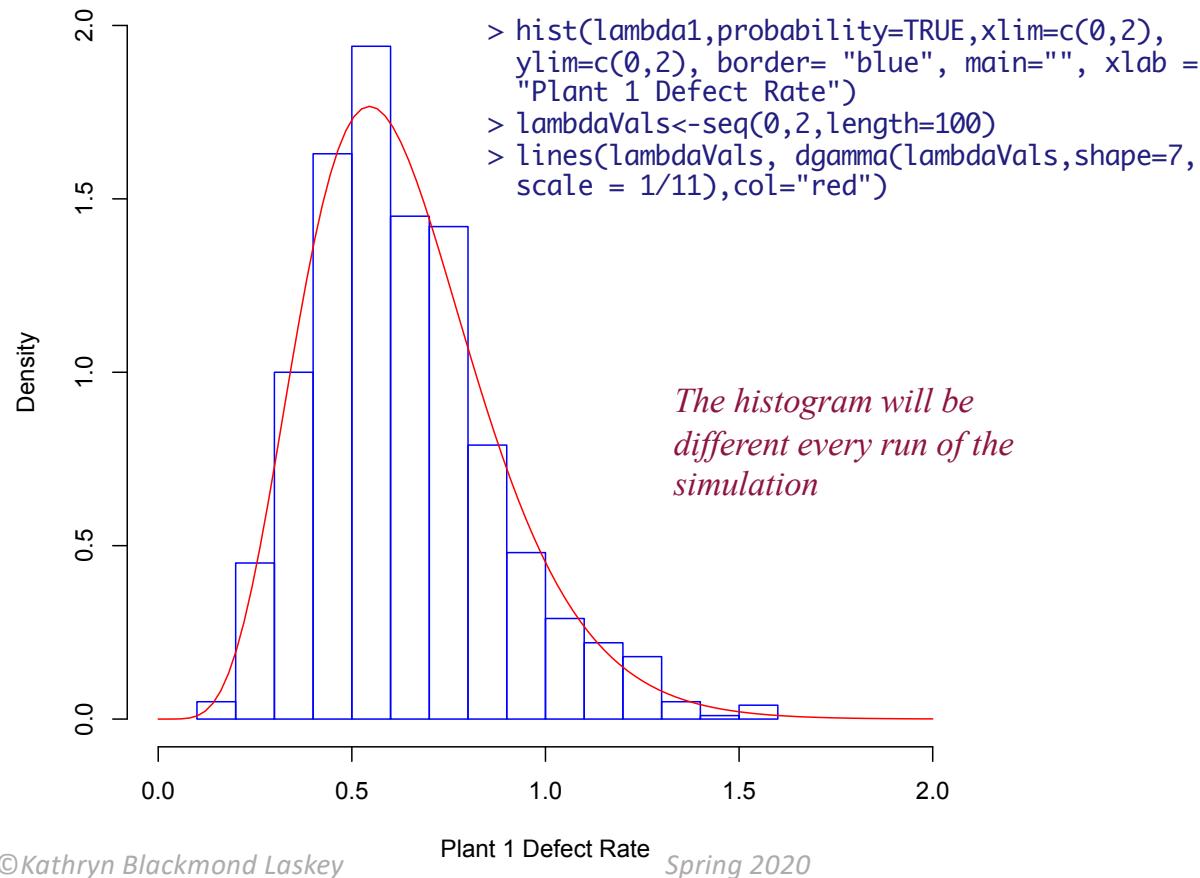


Using Monte Carlo to Estimate $\Pr(\Lambda_2 - \Lambda_1 > 0 | \text{data})$

- Find the joint posterior distribution for the two defect rates:
 - Λ_1 and Λ_2 are independent
 - $\Lambda_1 | \text{data from Plant 1} \sim \text{Gamma}(7, 1/11)$
 - $\Lambda_2 | \text{data from Plant 2} \sim \text{Gamma}(16, 1/11)$
- Draw a random sample from the joint posterior distribution for (Λ_1, Λ_2)
 - `sampleSize=1000` #Number of random draws
 - `lambda1<-rgamma(sampleSize, shape=7, scale=1/11)` #Sample from Plant 1 posterior
 - `lambda2<-rgamma(sampleSize, shape=16, scale=1/11)` #Sample from Plant 2 posterior
- Calculate frequency of samples in which Plant 2 has a larger defect rate
 - `diff=lambda2-lambda1` #Difference in defect rates
 - `probPlant2Worse = sum(diff>0) / length(diff)` #Proportion Plant 2 is worse
- Results of 10 repetitions of this procedure (10,000 total random draws):
 - 0.973, 0.975, 0.978, 0.977, 0.977, 0.974, 0.972, 0.966, 0.974, 0.962
 - Grand mean is 0.973; sample standard deviation is 0.005
 - Plant 2 has a larger defect rate in about 97% of the samples



Why Does Monte Carlo Work?



- Chart compares histogram of 1000 random samples from Plant 1 posterior distribution with true density function
- As the number of observations grows larger, empirical frequencies more closely approximate theoretical frequencies
- We can use empirical frequencies to estimate density/mass function



Monte Carlo Integration and Bayesian Statistics

- Often we want to find the posterior expected value of some function of the parameter: $E[h(\Theta) | X]$, e.g.:
 - Posterior mean $E[\Theta | X]$
 - Posterior variance $E[(\Theta - E[\Theta|X])^2 | X]$
 - Difference between two parameters $E[\Theta_1 - \Theta_2 | X]$
 - Many other examples
- This expectation can be written as

$$E[h(\Theta) | X] = \int_{\Theta} h(\theta)g(\theta|x)d\mu(\theta)$$

- This integral is often intractable
- We can use Monte Carlo to estimate this integral



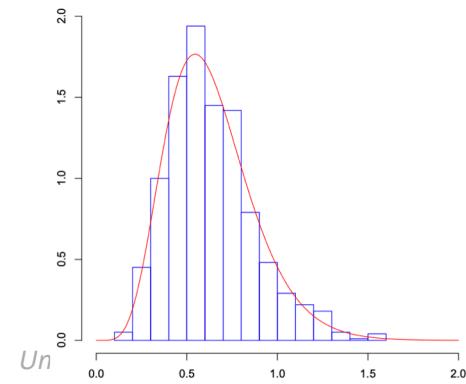
Direct Monte Carlo Estimation

$$E[h(\underline{\Theta}) | \underline{X}] \approx \frac{1}{N} \sum_i h(\underline{\Theta}_i)$$

- Our goal is to find an expected value of some random variable

$$E[h(\theta) | X] = \int_{\theta} h(\theta) g(\theta | x) d\mu(\theta)$$

- Often we cannot find a closed-form expression for this integral
- If we can simulate from $g(\theta | x)$ then we can estimate the integral by Direct Monte Carlo (dMC), the simplest form of Monte Carlo estimation
- How dMC works:
 - Simulate $\Theta_1, \Theta_2, \dots, \Theta_N$ from $g(\theta | x)$
 - Use estimator $E[h(\theta) | X] \approx \frac{1}{N} \sum_i h(\Theta_i)$
 - This converges to the desired integral as $N \rightarrow \infty$

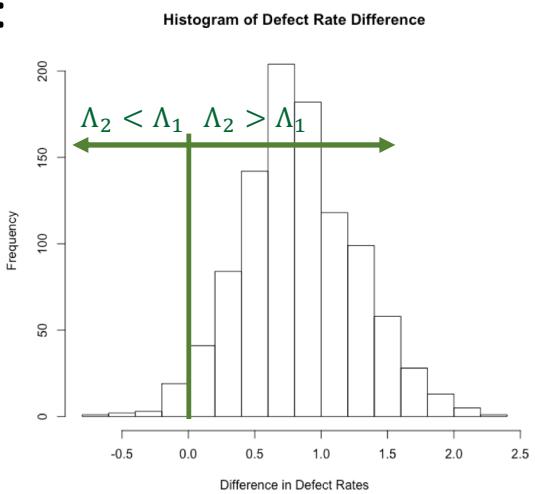


Direct Monte Carlo and the Two Plants

- Our goal is to draw inferences about the two defect rates $\Lambda = (\Lambda_1, \Lambda_2)$
 - Is the defect rate higher at the second plant?
 - Equivalently, is $\Lambda_2 - \Lambda_1 > 0$?
- We can formulate this question as a posterior expected value:

- $$h(\underline{\Lambda}) = \begin{cases} 1 & \text{if } \Lambda_2 > \Lambda_1 \\ 0 & \text{if } \Lambda_2 \leq \Lambda_1 \end{cases}$$
- $$\Pr(\Lambda_2 > \Lambda_1 | X = x) = \int_{\lambda_2 > \lambda_1} \int g(\underline{\lambda} | x) d\lambda_1 d\lambda_2 = \int_{\lambda_1, \lambda_2} h(\underline{\lambda}) g(\underline{\lambda} | x) d\underline{\lambda} = E[h(\underline{\Lambda}) | X = x]$$

- To estimate this expected value by dMC:
 - Simulate (Λ_1, Λ_2) from $g((\lambda_1, \lambda_2) | X = x)$
 - Calculate sample average $\frac{1}{N} \sum_i h(\underline{\Lambda}_i) = \frac{1}{N} \sum_{\Lambda_2 > \Lambda_1} 1 = \frac{\#(\Lambda_2 > \Lambda_1)}{N}$



Example Monte Carlo Estimators

- Expected value: $\hat{\mu} = \frac{1}{n} \sum_i \Theta_i$
- Variance: $\hat{\sigma}^2 = \frac{1}{n} \sum_i (\Theta_i - \hat{\mu})^2$ Standard deviation: $\hat{\sigma} = \sqrt{\frac{1}{n} \sum_i (\Theta_i - \hat{\mu})^2}$ (for unbiased estimator of σ^2 , divide by $n-1$)
- Probability that Θ is less than or equal to a (cdf of Θ at a): $\hat{\Pr}(\Theta_i \leq a) = \frac{\#\left[\Theta_i \leq a\right]}{n}$
- q th quantile: empirical q th quantile
- Expected value of any function $h(\Theta)$: $\hat{h} = \frac{1}{n} \sum_i h(\Theta_i)$



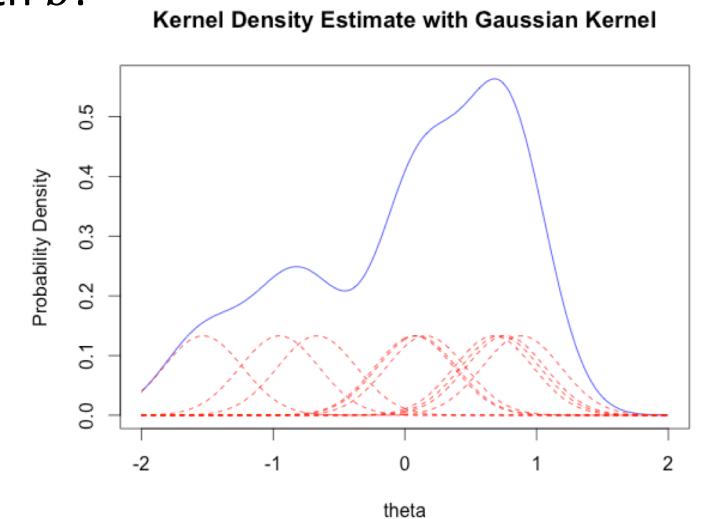
Kernel Density Estimator

- A kernel density estimator estimates a continuous density function from a sample (x_1, x_2, \dots, x_n) of observations
 - Kernel density estimator with kernel K and bandwidth b :

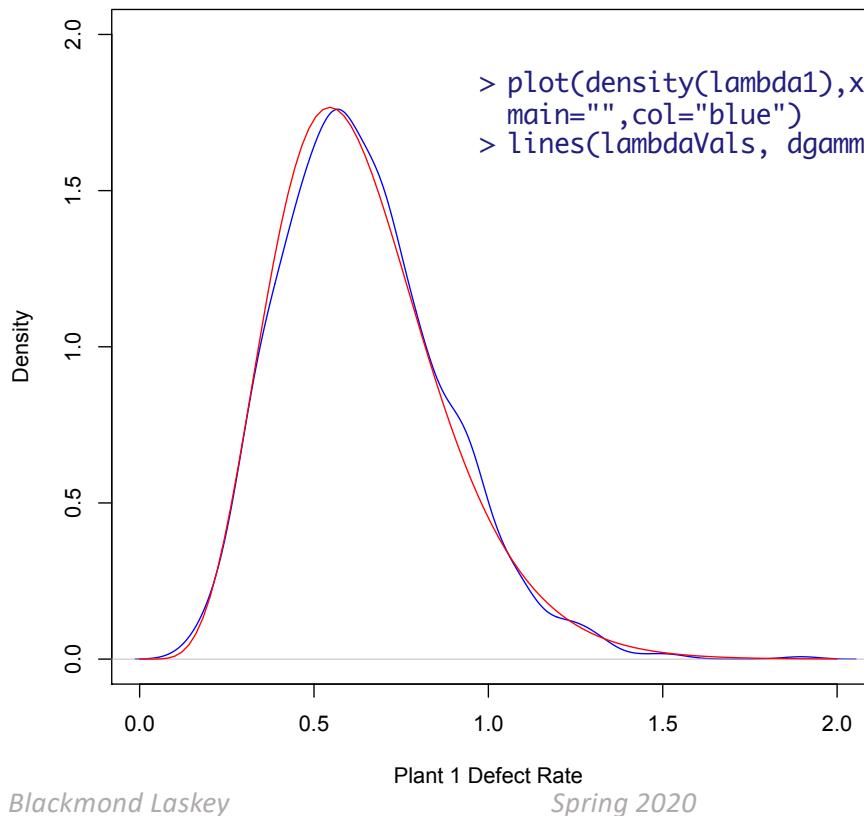
$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{b} K\left(\frac{x - x_i}{b}\right)$$

- Kernel is a location-scale family of density functions

- *This plot is a kernel density estimator with Gaussian kernel*
 - Average of N Gaussian densities
 - The i^{th} Gaussian density is centered at observation x_i
 - All the Gaussian densities have the same standard deviation b (usually chosen to be small)
 - The kernel density estimator has larger values where observations are most concentrated



Kernel Density Estimator for Plant 1 Defects (with Comparison to Exact Gamma Posterior Density)

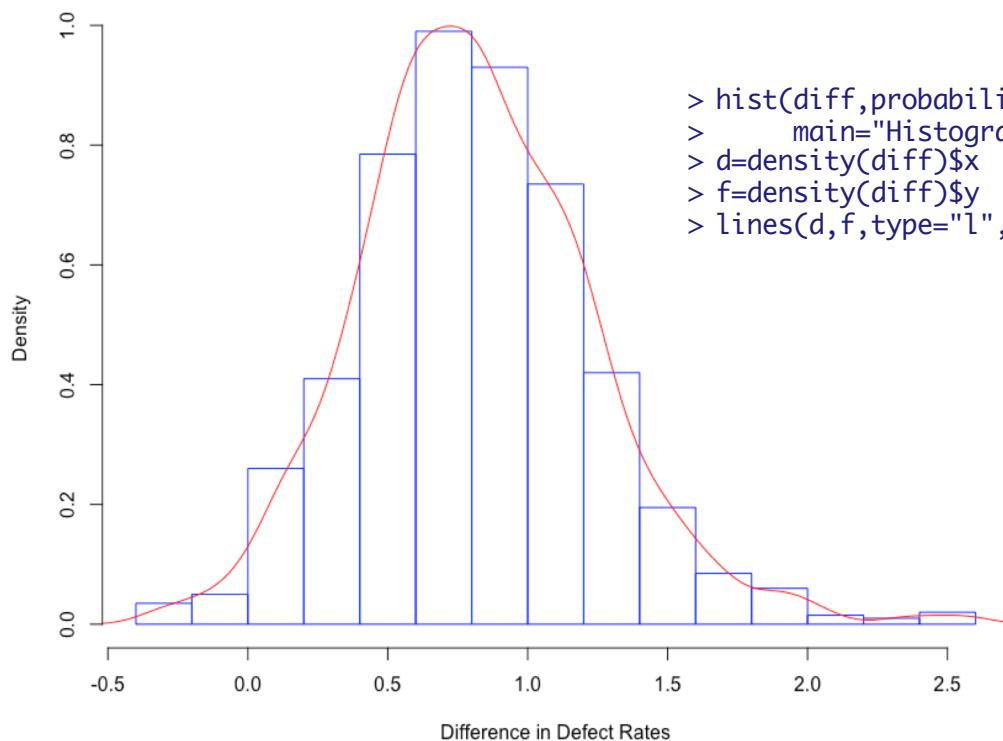


- Chart compares kernel density approximation (blue) for 1000 random samples with exact Gamma density function (red)
 - Use R function `density` to compute a kernel density object
- As sample size grows large, kernel density estimator closely approximates theoretical density function



Kernel Density Plot for Difference in Defect Rates

Histogram and Kernel Density Estimate



```
> hist(diff,probability=TRUE, border="blue",xlab="Difference in Defect Rates",
>       main="Histogram and Kernel Density Estimate")
> d=density(diff)$x    # points at which density is estimated
> f=density(diff)$y    # estimated density values
> lines(d,f,type="l",col="red")
```



How Accurate is our Monte Carlo Estimate?

- Suppose we are estimating the mean μ of the posterior distribution of Θ by dMC
- The Central Limit theorem implies that $\hat{\mu} = \frac{1}{n} \sum_i \Theta_i$ is approximately normally distributed with mean μ and variance σ^2/n , where n is the number of samples and σ^2 is the posterior variance of Θ
- We can estimate the accuracy of the dMC estimate by calculating the sample variance of our sample $\Theta_1, \Theta_2, \dots, \Theta_n$ and dividing by n
 - Suppose we draw 100 samples and calculate a sample standard deviation of 3.0
 - The standard deviation of our Monte Carlo estimate is approximately equal to $\sqrt{3.0^2/100} = 0.3$
- This method works to estimate the variance of a direct Monte Carlo estimate of the expected value of any function $h(\Theta)$



Achieving a Desired Accuracy for dMC

- If we need a certain accuracy, we can draw a sample of size n , estimate the variance, and then estimate how many additional samples we need to meet our target accuracy
 - Suppose we need to estimate the posterior mean μ to within ± 0.1
 - A 95% interval is about ± 2 standard deviations, so for an interval of ± 0.1 , we need a standard deviation of about 0.05
 - Suppose we draw 100 samples and calculate a sample standard deviation of 3.0
 - The standard deviation of our Monte Carlo estimate is approximately equal to $\sqrt{3.0^2/100} = 0.3$
 - To achieve a standard deviation of about 0.05, we solve $\sqrt{3.0^2/x} = 0.05$ for x
 - We need a total sample size of about 3600, because $\sqrt{3.0^2/3600} = 0.05$
- This method can be used to plan the number of samples for a dMC estimate of the expected value of any function $h(\Theta)$

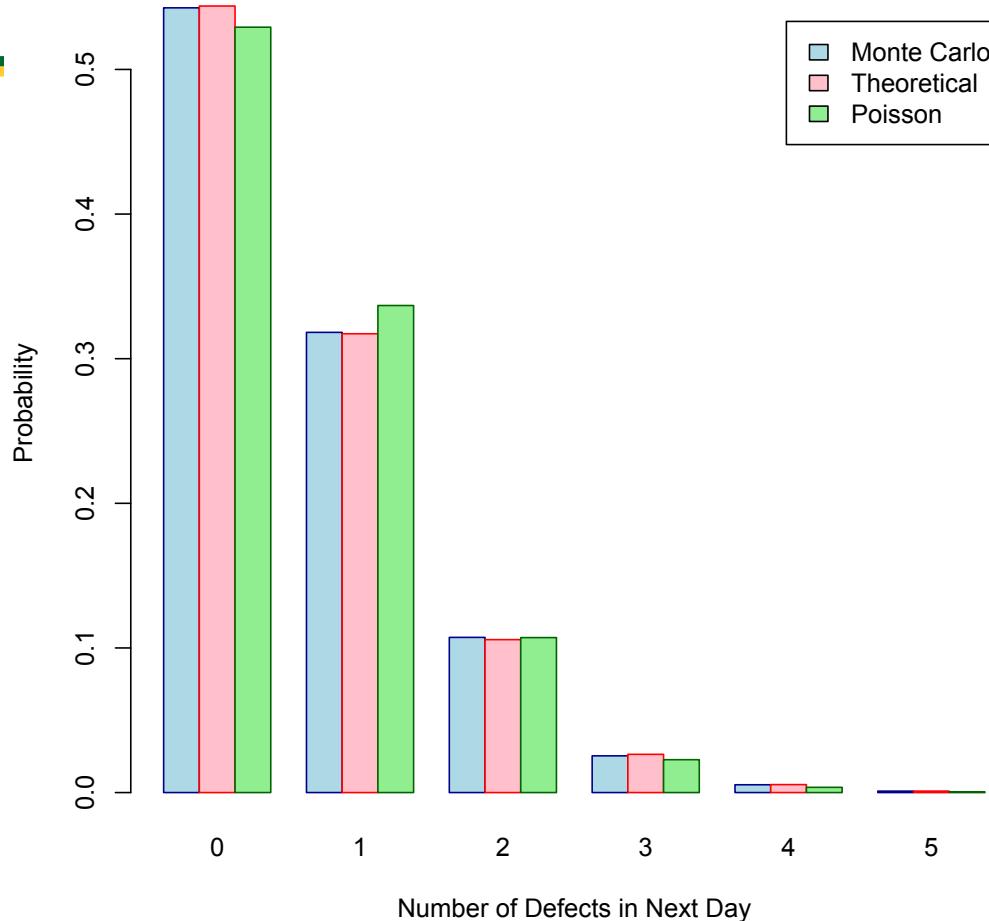


Estimating the Predictive Distribution by Direct Monte Carlo

- To simulate defects on the next day at Plant i (for $i=1,2$) by direct Monte Carlo:
Repeat until desired sample size has been reached:
 - Sample defect rate Λ_i from posterior distribution for Plant i
 - Sample number of defects X_i from Poisson distribution with rate Λ_i
- This method takes into account both uncertainty in Λ_i and uncertainty in X_i given Λ_i
- We can use our Monte Carlo sample to estimate the predictive distribution
 - We can compare our results with the exact negative binomial predictive distribution



Predicting Defects Next Day at Plant 1



Monte Carlo estimate with 10,000 samples accurately estimates theoretical negative Binomial predictive distribution

- Size: $\alpha=7$,
- Prob: $(1+\beta)^{-l} = (1+11^{-1})^{-l} = 0.917$

Poisson distribution is too narrow but not too inaccurate for one-day-ahead prediction

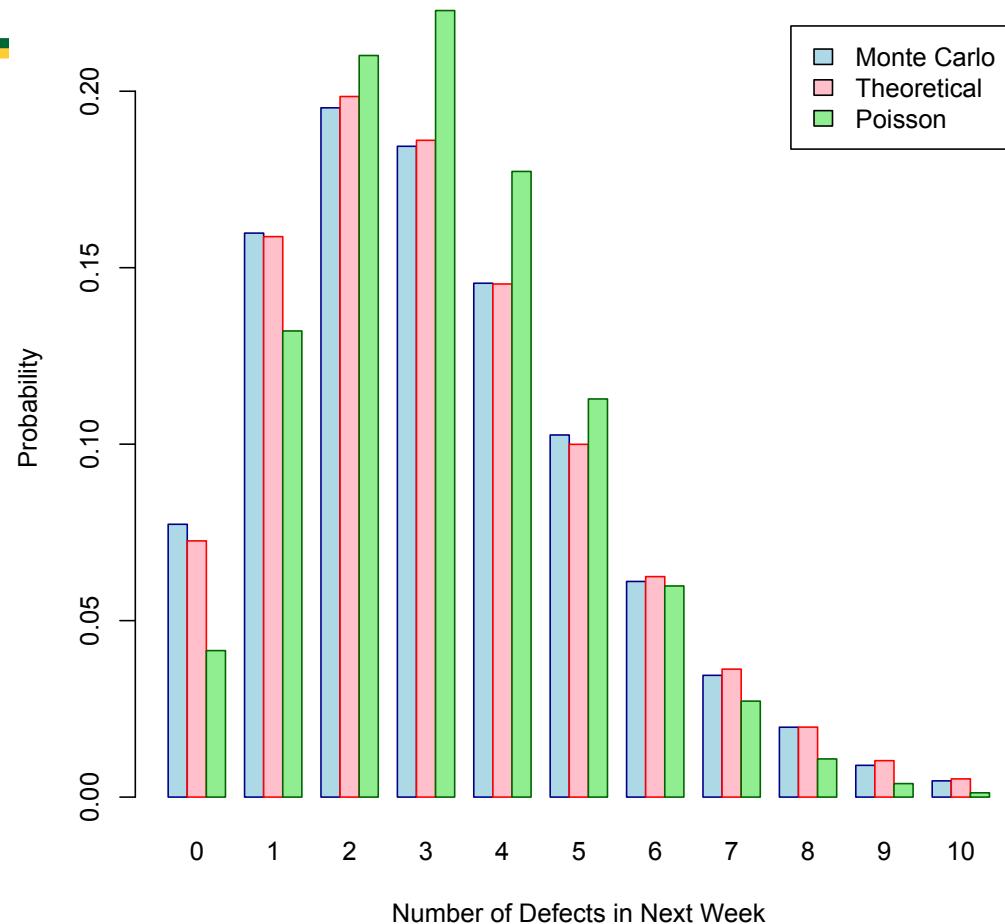


R Code for Posterior Predictive Simulation (1 day)

```
1. #Simulate the predictive distribution for daily defects in Plant 1
2. numSim=10000          #Monte Carlo Sample Size
3. predictedDefects<-NULL
4. for (i in 1:numSim) {
5.   lambda1<-rgamma(1,shape=7,scale=1/11) #Simulate defect rate
6.   predictedDefects[i]<-rpois(1,lambda1) #Simulate defects
7. }
8. defectProbMC<-table(predictedDefects)/numSim      #Monte Carlo estimate of predictive distribution
9. defectProbTh<-dnbinom(0:5,size=7,prob=1/(1+1/11)) #Theoretical Poisson-Gamma predictive
10. defectProbPs<-dpois(0:5,7/11)                      #Poisson distribution with point estimate
11. defectProb <- matrix(nrow=3,ncol=6)                 #Make a matrix
12. defectProb[1,] <- defectProbMC[1:6]                  #Plot only 0 to 5 defects
13. defectProb[2,] <- defectProbTh
14. defectProb[3,] <- defectProbPs
15. barplot(defectProb,main="Predicting Defects Next Day at Plant 1", xlab="Number of Defects in Next Day",
   ylab="Probability", col=c("lightblue","pink","lightgreen"),
   border=c("darkblue","red","darkgreen"),names=0:5, beside=TRUE,legend=c("Monte
   Carlo","Theoretical","Poisson"))
```



Predicting Defects Next Week at Plant 1



- Monte Carlo estimate with 10,000 samples accurately estimates theoretical negative Binomial predictive distribution
 - Size: $\alpha=7$,
 - Prob: $(1+n\beta)^{-l} = \left(1 + \frac{7}{11}\right)^{-1} = 0.688$
- Poisson distribution substantially underestimates uncertainty in predicting a week of defects

R Code for Posterior Predictive Simulation (5 days)

```
1. #Simulate the predictive distribution for weekly defects in Plant 1
2. numSim=10000          #Monte Carlo Sample Size
3. predictedDefects<-NULL
4. for (i in 1:numSim) {
5.     lambda1<-rgamma(1,shape=7,scale=1/11) #Simulate defect rate
6.     predictedDefects[i]<-sum(rpois(5,lambda1)) #Simulate defects
7. }
8. defectProbMC<-table(predictedDefects)/numSim      #Monte Carlo estimate of predictive distribution
9. defectProbTh<-dnbinom(0:10,size=7,prob=1/(1+5/11)) #Theoretical Poisson-Gamma predictive
10. defectProbPs<-dpois(0:10,35/11)                  #Poisson distribution with point estimate
11. defectProb <- matrix(nrow=3,ncol=11)             #Make a matrix
12. defectProb[1,] <- defectProbMC[1:11]              #Plot only 0 to 10 defects
13. defectProb[2,] <- defectProbTh
14. defectProb[3,] <- defectProbPs
15. barplot(defectProb,main="Predicting Defects Next Week at Plant 1", xlab="Number of Defects in Next Week",
   ylab="Probability", col=c("lightblue","pink","lightgreen"),
   border=c("darkblue","red","darkgreen"),names=0:10, beside=TRUE,legend=c("Monte
Carlo","Theoretical","Poisson"))
```

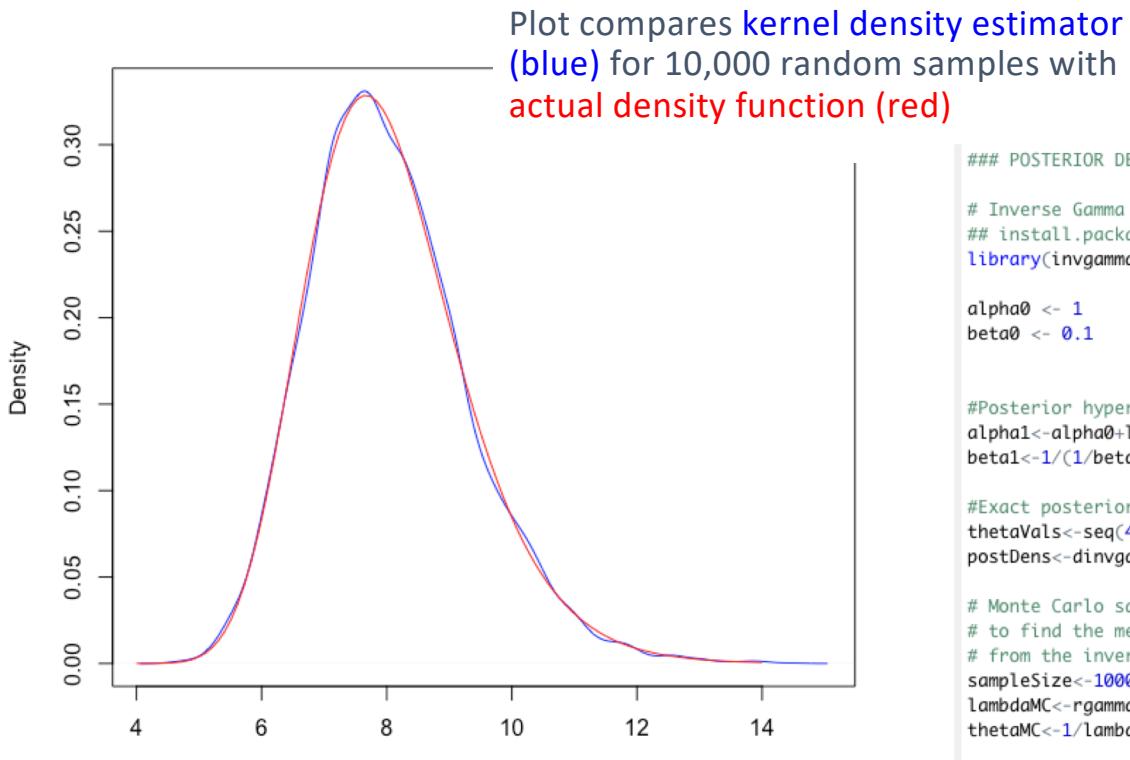


Monte Carlo Example: Intervals Between Cars

- Data collected on time intervals between cars passing a fixed point on M1 motorway near Bedfordshire, UK on 23 March 1985
 - 41 cars
 - 40 inter-arrival intervals
 - Total time from first to last car is 312 seconds
- Model:
 - Inter-arrival intervals have exponential distribution with parameter Θ
 - Prior distribution for Θ is inverse gamma distribution with shape $\alpha=1$ and inverse-scale $\beta=0.1$
 - Mean of Θ is $E[\Theta|\alpha, \beta] = \frac{1}{\beta(\alpha-1)} = \infty$
 - 50th percentile of Θ is 14.4 seconds between cars; 90% interval is [3.3, 195] seconds between cars
 - Mean of Θ^{-1} is $E[\Theta^{-1}|\alpha, \beta] = \alpha\beta = 0.1$ cars per second (1 car per 10 seconds)
- Exact posterior distribution of Θ is inverse gamma with parameters
 - Shape $\alpha^* = \alpha + n = 1 + 40 = 41$ and
 - Inverse-scale $\beta^* = 1/(\beta^{-1} + \sum_i x_i) = 1/(0.1^{-1} + 312) = 0.031$
- We will compare direct Monte Carlo approximation with exact inverse gamma posterior distribution



Posterior Density for Mean Time Between Cars



```
### POSTERIOR DENSITY - MONTE CARLO AND EXACT COMPARISON

# Inverse Gamma density function
## install.packages("invgamma") # install inverse-gamma package (only do this once)
library(invgamma) # load inverse-gamma functions

alpha0 <- 1 # prior shape
beta0 <- 0.1 # prior inverse-scale (remember that invgamma package calls this the scale)

#Posterior hyperparameters for inverse Gamma prior with shape alpha0 and inverse-scale beta0
alpha1<-alpha0+length(intervals) # Posterior shape (# intervals is 40)
beta1<-1/(1/beta0+sum(intervals)) # Posterior scale (total from 1st to last car is 312 seconds)

#Exact posterior density function
thetaVals<-seq(4,14,length=100)
postDens<-dinvgamma(thetaVals,shape=alpha1,scale=beta1)

# Monte Carlo sample - we sample the rate from a gamma distribution and invert
# to find the mean of the exponential distribution. We could also sample directly
# from the inverse-gamma distribution
sampleSize<-10000
lambdaMC<-rgamma(sampleSize,shape=alpha1,scale=beta1) # Sample rate
thetaMC<-1/lambdaMC # inverse of the rate is the mean

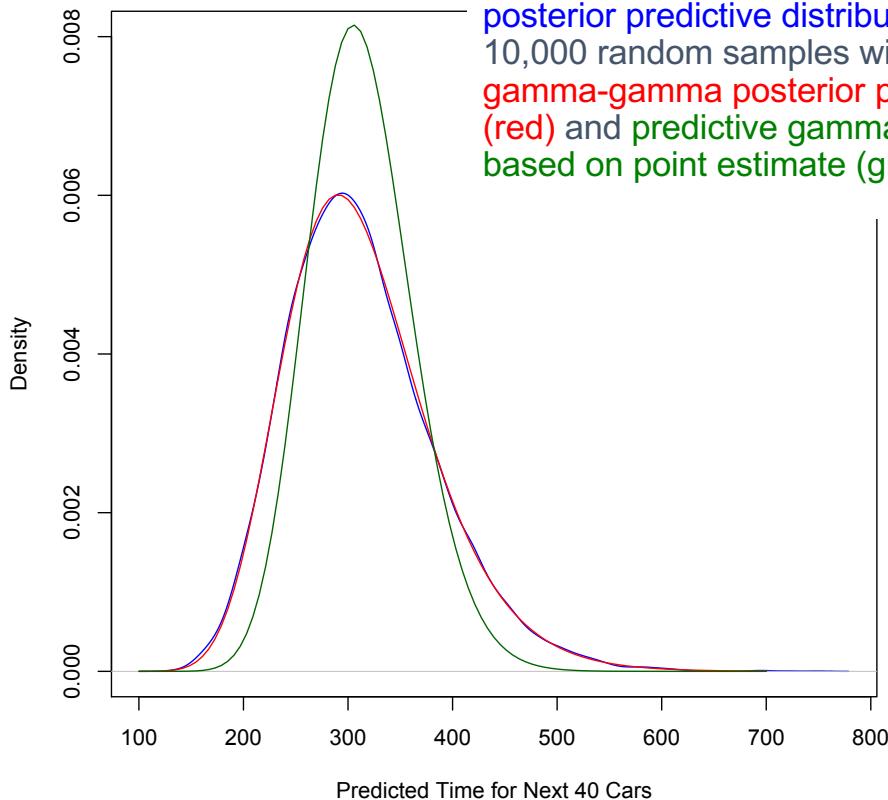
#Comparison plot
plot(density(thetaMC), col="blue",main="",xlab="Mean Time Between Cars")
lines(thetaVals,postDens,col="red")
```

Predicting the Next 40 Cars

- The theoretical predictive distribution of the time $Y = \sum_i X_i$ until the next $n=40$ cars pass is a gamma-gamma distribution with:
 - $\text{shape1} = 41$ (shape parameter of posterior distribution)
 - $\text{rate1} = 322$ (1/inverse-scale parameter of posterior distribution)
 - $\text{shape2} = 40$ (we are predicting time to next 40 events)
- To simulate this predictive distribution by Monte Carlo:
Repeat until desired sample size has been reached:
 - Simulate the rate Λ_i from a gamma distribution with shape 41 and rate 322
 - Calculate the mean $\Theta_i = 1/\Lambda_i$ of the exponential inter-arrival distribution
 - Simulate 40 observations from the exponential distribution with mean Θ_i or rate Λ_i
 - The `rexp` function in R uses the rate parameter
 - Calculate the sum of the 40 observations to get the total time



Predictive Distribution: Next 40 Cars



```
### PREDICTIVE DENSITY - MONTE CARLO AND EXACT COMPARISON

#Monte Carlo posterior predictive distribution for next 40 cars
# Note that R parameterizes exponential distribution using the rate
timeForNext40Cars<-NULL
for (i in 1:sampleSize) {
  next40Intervals<-rexp(40,rate=lambdaMC[i])
  timeForNext40Cars[i]<-sum(next40Intervals)
}

#Theoretical predictive density for 40 exponential observations with inverse-Gamma prior
library(BAEssd) # Package containing gamma-gamma density function
timeTo40<-seq(100,700,length=100) # range of values for predicting
shape1 = alpha1 # Shape parameter for posterior distribution
rate1 = 1/beta1 # Rate (1/inverse-scale) parameter for posterior distribution
shape2 = 40 # Number of events to predict

thPred=dggamma(timeTo40, shape1, rate1, shape2)

#Comparison plot - theoretical and Monte Carlo predictive distributions
plot(density(timeForNext40Cars), # Kernel density estimator
      col="blue",main="",xlab="Predicted Time for Next 40 Cars",
      ylim=c(0,.008))
lines(timeTo40,thPred,col="red") # Gamma-gamma predictive distribution
lines(timeTo40, # Compare with Poisson using point estimate
      dgamma(timeTo40,shape=40,scale=312/40),col="darkgreen")
```

Evaluating Model Adequacy

- “All models are wrong, but some models are useful” – George Box
- Models are convenient formalisms for:
 - analyzing data
 - formulating & evaluating hypotheses about data generation process
 - making predictions
- Components of model building:
 - Estimation – infer parameters conditional on truth of model
 - Criticism – confront model with data, ask whether model is consonant with it
 - Revision – decide how to change the model based on criticism, modeling goals, modeling resources
- When model is very implausible, conclusions drawn from model can be very misleading
- Model criticism:
 - Is my model good enough?
 - In what ways is it not good enough?

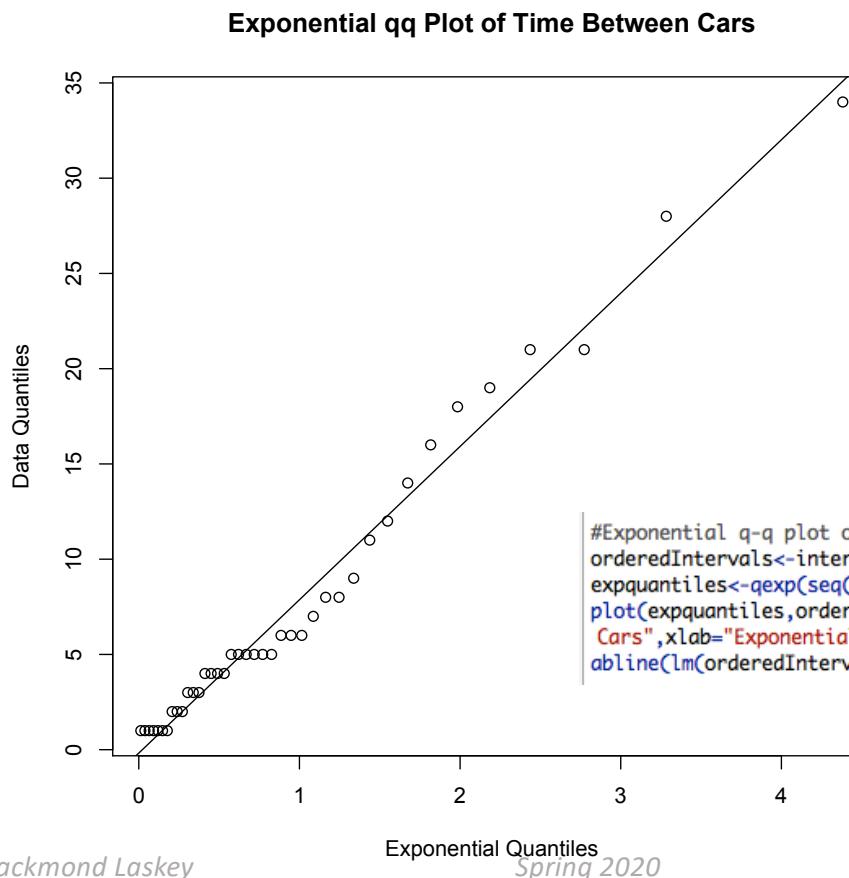
David Blei: <https://www.cs.princeton.edu/courses/archive/fall11/cos597C/lectures/ppc.pdf>

Model Evaluation Approaches

- Informal qualitative evaluation
- Formal frequentist goodness-of-fit tests
- Bayesian posterior predictive model evaluation



Qualitative Evaluation: q-q Plot



Highway traffic example:
Conclusion from q-q Plot:
Exponential distribution seems
to fit reasonably well
(but lots of repeated values for
small intervals)

Frequentist Tests for Model Adequacy

- Ingredients:
 - Assumed model H from parametric family (want to evaluate adequacy of H)
 - Observed data x (assumed drawn from $p(\cdot | H, \theta)$)
 - Hypothetical replications x^{rep} drawn from the same distribution $p(\cdot | H, \theta)$
 - (Imagine same experiment repeated with same model and same unknown θ)
 - Test statistic $T(x)$ (measures discrepancy from model; large values mean model fits poorly)
- Distribution of $T(x)$ must be (exactly or approximately) independent of parameter θ
- “p-value” p_H is defined as:
$$p_H = P(T(x^{rep}) > T(x) | H, \theta) \approx P(T(x^{rep}) > T(x) | H)$$
 - Inference is based on the sampling distribution of the hypothetical replications
 - Because θ is unknown, evaluating the p-value p_H requires distribution of $T(x)$ to be independent of θ
- Example: chi-square goodness of fit test:
 - $T(x) = \sum_i \frac{(O_i - E_i)^2}{E_i}$ has (given commonly assumed conditions) approximately chi-square distribution



Bayesian Posterior Predictive Model Checks

- From Rubin (1984):
 - Given observed data, X_{obs} , what would we expect to see in hypothetical replications of the study that generated X_{obs} ? Intuitively, if the model specifications are appropriate, we would expect to see something similar to what we saw this time, at least similar in “relevant ways.” This statement, which is essentially a fundamental premise of frequency inference, seems to me so basic that it needs no defense. Unlike the frequentist, the Bayesian, though, will condition on all observed values.
- Frequentist goodness-of-fit tests require finding a test statistic $T(X)$ with exactly or approximately independent of parameter θ
- A Bayesian uses a “discrepancy measure” whose distribution may depend on θ
 - Bayesian p-value is based on posterior predictive distribution of X_{obs}
 - X_{obs} should appear “typical” under the posterior predictive distribution
- The posterior predictive model checking process:
 - Define discrepancy $D(X)$ [or $D(X; \theta)$] that formalizes “relevant” deviations from model
 - Define “replication” of current study
 - Calculate frequency distribution of $D(X)$ [or $D(X; \theta)$] in hypothetical future replications
 - Use posterior predictive distribution conditional on observed data X_{obs} and model
 - Evaluate whether X_{obs} is “typical”

Example: Mondays at the Plant

- Notice that the two highest defect days at Plant #2 occur on Mondays
 - 3 in Week 1, 4 in Week 2, total 7 Monday defects out of 15 total defects
- It is known that some facilities have higher injury and error rates on Mondays
- Does Plant 2 have a **problem with Mondays**, or is it just sampling variability?
 - Our model cannot answer this question!
- We might address this question with a frequentist test
 - Observed data: 7 defects on Mondays, 8 defects on other days
 - Null hypothesis: expect 4 times as many defects Tuesday-Friday as on Monday
 - Chi-square test statistic:
 - $\chi^2 = \frac{(7-3)^2}{3} + \frac{(8-12)^2}{12} = 6.67$ *Expect 3 defects on Mondays; 12 defects on other days*
 - Chi-square p-value
 - $P(\chi^2_{df=1} \geq 6.67) = 0.0098$
 - Mondays look unusual (but interpret p-values for post-hoc tests cautiously!)
- How would we approach this with Bayesian posterior predictive check?

	Plant # 1	Plant # 2
Week 1: M	1	3
Week 1: T	0	0
Week 1: W	0	1
Week 1: H	0	0
Week 1: F	0	1
Week 2: M	0	4
Week 2: T	1	1
Week 2: W	1	2
Week 2: H	1	2
Week 2: F	2	1

Monday Effect: Bayesian Posterior Predictive Check

- We are concerned about a possible “Monday effect.”
- We looked at a commonly used frequentist model evaluation tool
 - We did a chi-square test of hypothesis that Mondays have same frequency as other days
 - The p-value was about 0.01, which suggests the model of equal frequencies may be problematic
- Now we consider a Bayesian test using the posterior predictive distribution
 - We need to pick a discrepancy measure
 - We do not need to find one (like the chi-square test) with a distribution approximately independent of λ
 - Suggestion: compare average number of defects on Mondays with average number of defects on other days
- Discrepancy measure to evaluate whether defects on Mondays at Plant 2 are unusually high according to our model:
$$D(X) = (\text{Avg Monday defects} - \text{Avg defects on other days})$$
$$D(X_{obs}) = (7/2 - 8/8) = 2.5$$
- In posterior predictive replications, how often is $D(X) > 2.5$?



Implementing Posterior Predictive Model Check

- We defined a discrepancy measure to evaluate whether defects on Mondays at Plant 2 are unusually high according to our model:

$$D(X) = (\text{Average Monday defects} - \text{Average defects on other days})$$

- To implement our model check:

Calculate $D(X_{obs}) = 7/2 - 8/8 = 2.5$ # avg Monday defects minus avg other-day defects

For $i = 1$ to desired_sample_size {

 Sample defect rate Λ_i from posterior distribution for Plant 2

 Sample number of defects X_{Mi} on the first two Mondays as sum of two Poisson observations
 with rate Λ_i

 Sample number of defects X_{Ri} on the remaining days as sum of eight Poisson observations
 with rate Λ_i

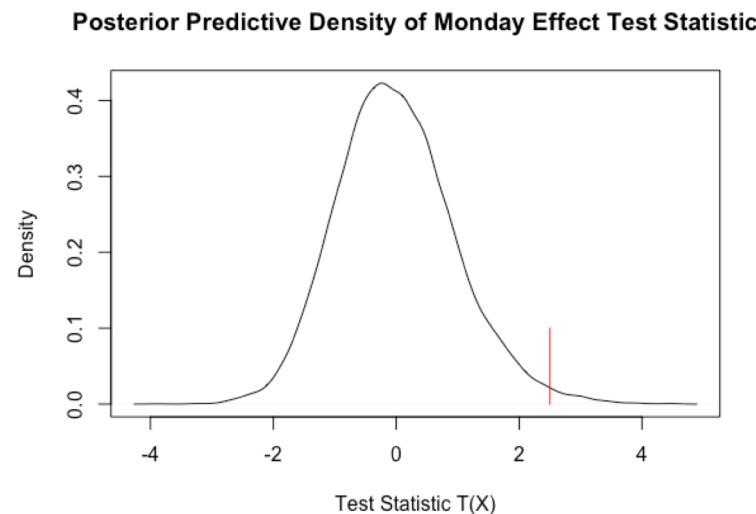
 Compute $D_i(X) = X_{Mi}/2 - X_{Ri}/8$ # avg Monday defects minus avg other-day defects
 }

- How often is $D_i(X)$ greater than or equal to $D(X_{obs}) = 2.5$?



Results of Posterior Predictive Model Check of Monday Effect

- In 20 repetitions of simulation with 10000 draws from the posterior predictive distribution, the Bayesian posterior predictive p-value averaged 0.012 with a standard deviation of 0.0011
- Therefore, according to our model, discrepancy of 2.5 between Mondays & other days is very unusual
- This is evidence that Mondays do not have the same defect rate as other days at Plant 2
- This calls into question the assumption that defect counts at Plant 2 are iid Poisson(Λ_2) random variables
- We must remember this is a post-hoc test and should be verified by taking additional data for Plant 2



High Defect Days with No Pattern

- What if a third plant had the same defect counts as Plant 2, but there was no special significance to the days with high defects?
- We still have more high defect days than expected under the Poisson distribution, but is it a big enough difference to question the iid model for Plant 3?
- An appropriate posterior predictive check for this situation is to compare **the 2 worst days** with the other days

Set $D(X_{obs}) = 2.5$, average defects on 2 worst days – average defects on other days

Repeat until desired sample size has been reached {

Sample defect rate Λ_3 from posterior distribution for Plant 3

For $k = 1, \dots, 10$ {

 Sample number of defects X_{k3} from Poisson distribution with rate Λ_3 }

 Calculate $D_{Wi}(X) =$ average number of defects on the two worst days

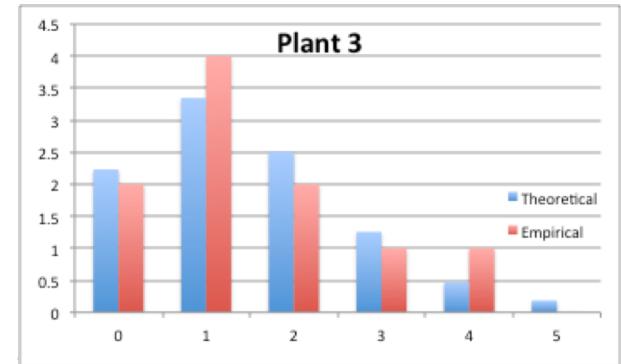
 Calculate $D_{Ri}(X) =$ average number of defects on the remaining days

 Compute $D_i(X) = D_{Wi}(X) - D_{Ri}(X)$, the difference between these averages

}

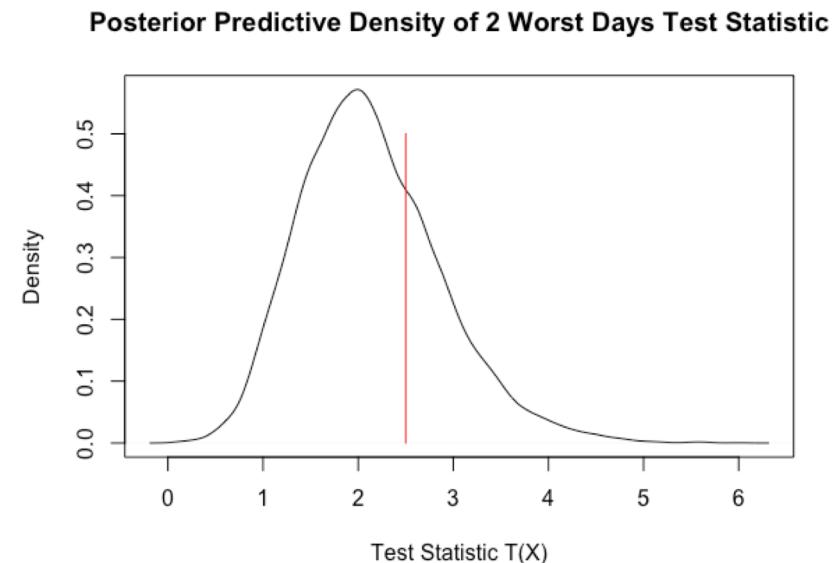
- How often is $D_i(X)$ greater than or equal to $D(X_{obs}) = 2.5$?

	Plant # 3
Week 1: M	0
Week 1: T	3
Week 1: W	1
Week 1: H	0
Week 1: F	1
Week 2: M	2
Week 2: T	1
Week 2: W	2
Week 2: H	4
Week 2: F	1



Results of Posterior Predictive Model Check of Worst Days Effect

- In 20 repetitions of simulation with 10000 draws from the posterior predictive distribution, the Bayesian posterior predictive p-value averaged 0.328 with a standard deviation of 0.039
- Therefore, according to our model, 7 defects on the 2 worst days is not too unusual
- Our model expects more defects on the two worst days than on two specific days



Posterior Predictive Checks: Comparison

- We applied posterior predictive model checking to evaluate our model for defects at the plants
- For Plant 2 defects:
 - We were concerned that the defect rate might be higher on Mondays
 - We simulated a large sample of “hypothetical replications” of our study, drawing the rate Λ_2 from the posterior distribution and the defects as iid Poisson(Λ_2)
 - We found that a difference of 2.5 between average Monday defects and average defects on other days is very unusual
 - This calls our iid Poisson(Λ_2) model for Plant 2 into question
- For Plant 3 defects:
 - We were concerned that there might be more high-defect days than expected under the Poisson distribution
 - We simulated a large sample of “hypothetical replications” of our study, drawing the rate Λ_3 from the posterior distribution and the defects as iid Poisson(Λ_3)
 - We found that a difference of 2.5 in average defects on the worst two days and average defects on other days is not too unusual
 - Our iid Poisson(Λ_3) model for Plant 3 is not called into question

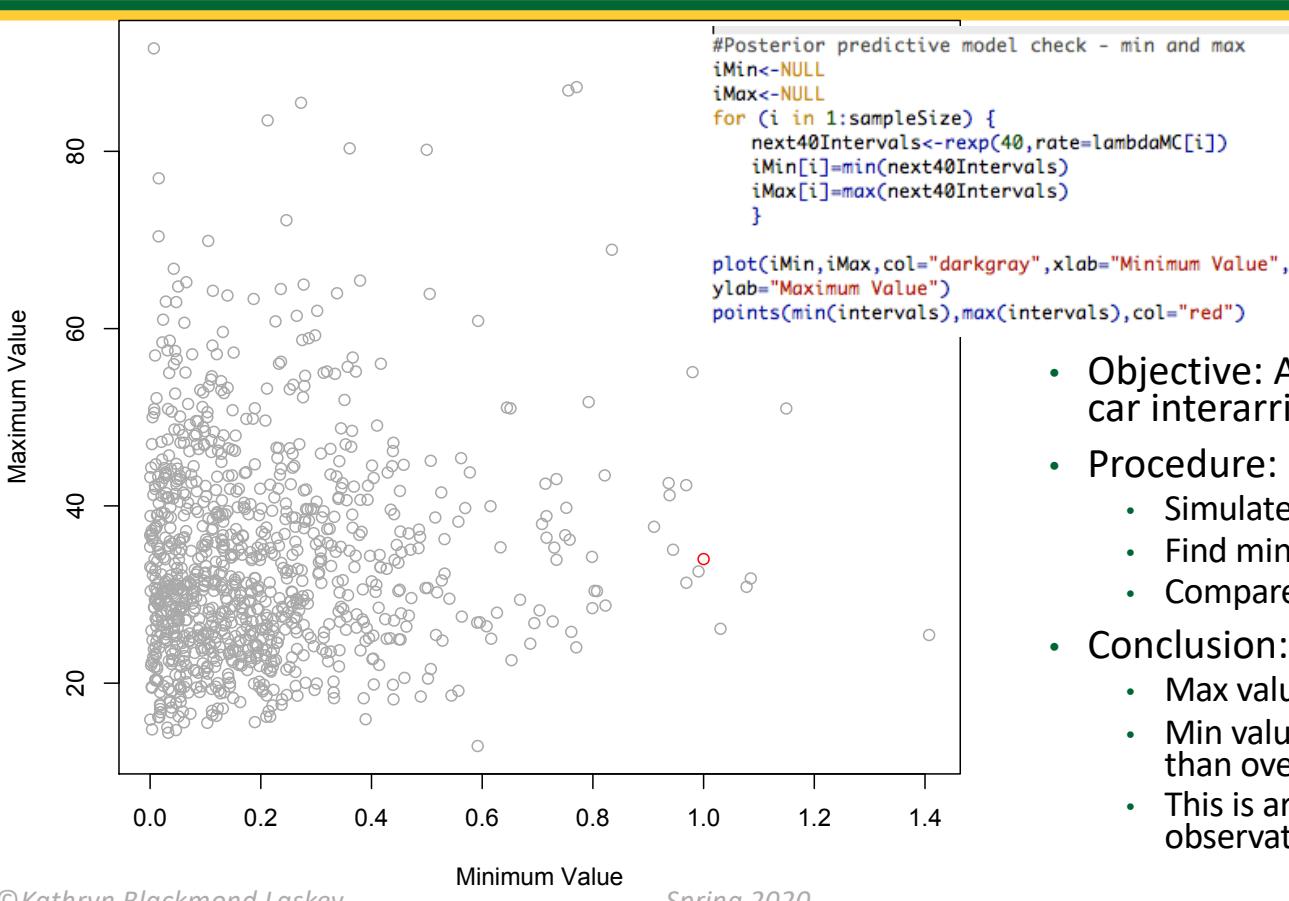


Posterior Predictive Model Checking: General Procedure

- Think about which departures from modeling assumptions it is important to evaluate
- Choose discrepancy measure D
 - Function of sample (X_1, X_2, \dots, X_n)
 - Design $D(X_1, X_2, \dots, X_n)$ to be diagnostic of important departures from model assumptions
 - Do not use sufficient statistic (posterior parameters are estimated to fit sufficient statistic!)
 - Examples:
 - To diagnose asymmetry in model that assumes symmetry, we might compare $D_1 = X_{0.75} - X_{0.5}$ with $D_2 = X_{0.5} - X_{0.25}$
 - To diagnose outliers, we might use $D = X_{0.95} - X_{0.05}$
- Compute discrepancy measure $D_{obs} = D(X_1, X_2, \dots, X_n)$ for observed sample
- Generate m posterior predictive samples $(X_1^{(i)}, X_2^{(i)}, \dots, X_n^{(i)}), i = 1, \dots, m$ as follows:
 - Generate $\Theta^{(i)}$ from posterior distribution of Θ given the observations X_1, X_2, \dots, X_n
 - Generate n iid observations $(X_1^{(i)}, X_2^{(i)}, \dots, X_n^{(i)})$ from likelihood $f(x|\Theta^{(i)})$
 - Calculate test statistic $D^{(i)} = D(X_1^{(i)}, X_2^{(i)}, \dots, X_n^{(i)})$ for i^{th} posterior predictive sample
- Compare test statistic from the actual sample with the test statistics $D^{(1)}, \dots, D^{(m)}$ from posterior predictive samples
- If actual test statistic is highly atypical of the simulated values, this calls the model into question



Posterior Predictive Check for Highway Traffic: Min and Max



- Objective: Assess whether min and max values in car interarrival times are unusually extreme
- Procedure:
 - Simulate many data sets
 - Find min and max of each
 - Compare with min and max of observations
- Conclusion:
 - Max value seems typical
 - Min value is unusual (observed value greater than over 99% of sampled values)
 - This is artifact of 1-second minimum in observations

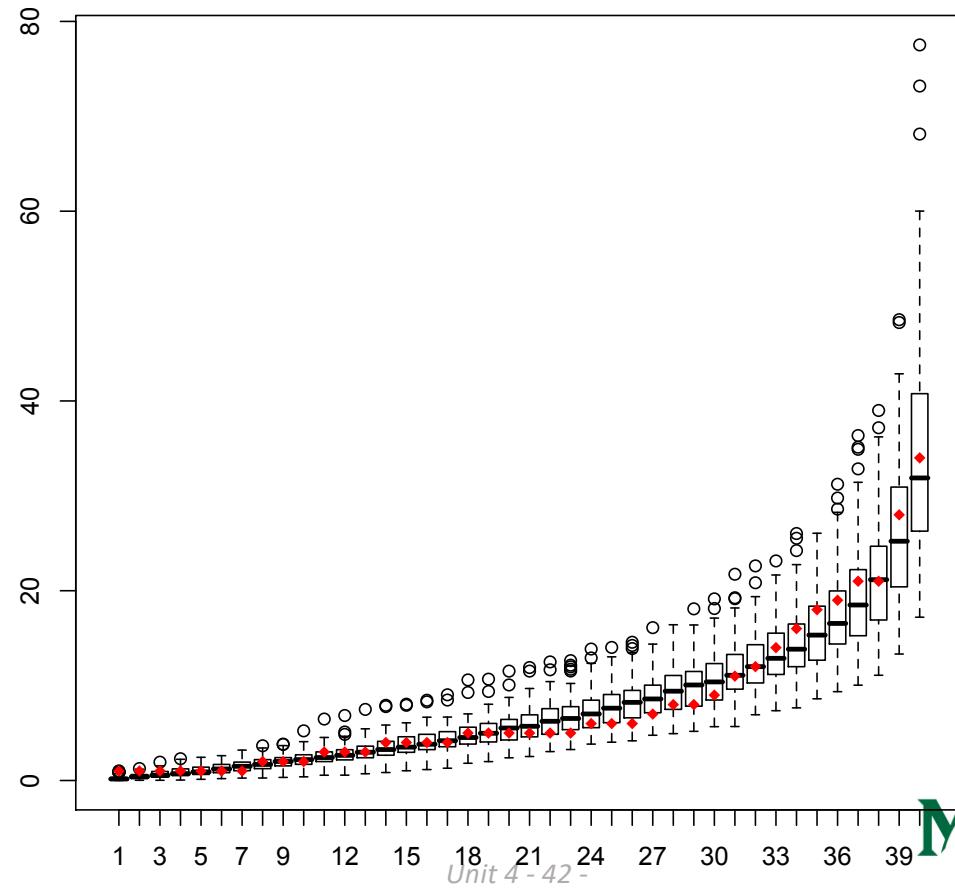


Highway Interval Data: Boxplot of Order Statistics from 1000 Posterior Predictive Samples

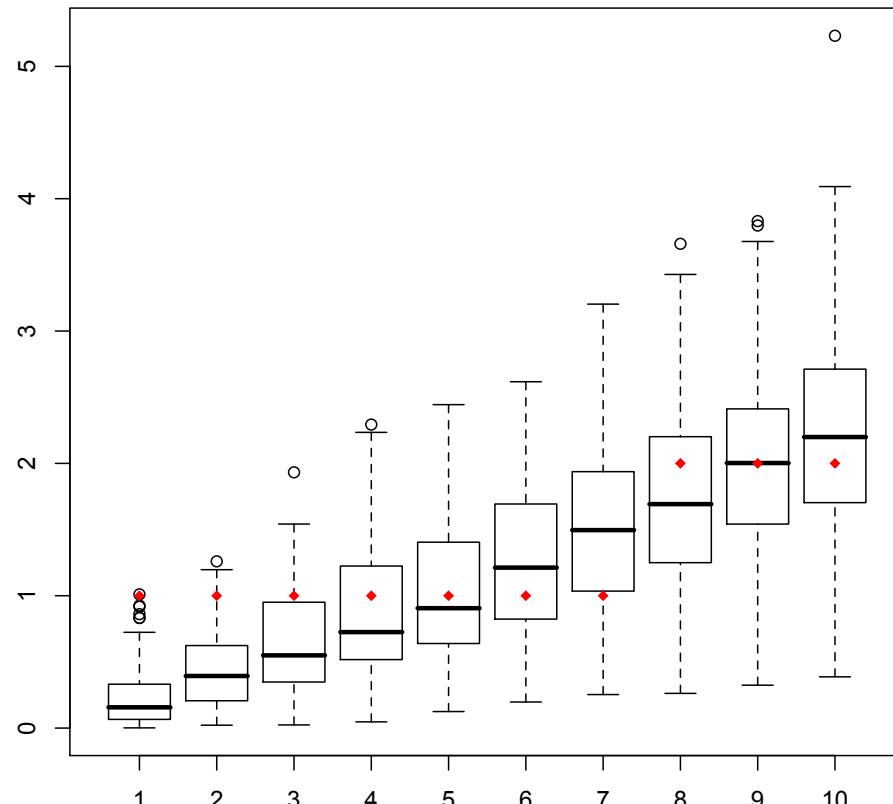
```
#Posterior predictive comparison of order statistics
predQuantilesMC<-matrix(nrow=sampleSize,ncol=length(intervals))
for (i in 1:sampleSize) {
  next40Intervals<-rexp(40,rate=lambdaMC[i])
  predQuantilesMC[i,]<-next40Intervals[order(next40Intervals)]
}

#Make a list out of the 40 MC quantiles
qq=list()
for (i in 1:length(intervals)) {
  qq=c(qq,list(predQuantilesMC[,i]))}

#grand boxplot
boxplot(qq)  #boxplot
points(1:40,orderedIntervals,col="red",pch=18)
```



Order Statistics 1 - 10 Show Rounding Effects



Posterior Predictive Model Evaluation: Recap

- Posterior predictive simulation can be used to evaluate whether a model is adequate for the problem
- Basic idea: if our model is correct, hypothetical replications of our data should be similar in relevant ways to the observed sample
 - Develop measures deviation of model from expected behavior for important ways our model could be in error
 - Simulate hypothetical replications with parameter drawn from posterior distribution and data drawn from likelihood
 - If our observed sample is very unusual this is cause to question the model
- We considered several examples of posterior predictive model evaluation



Monte Carlo: Review

- Monte Carlo can be used to approximate many aspects of a posterior distribution
 - Density function (histogram / kernel density estimator)
 - Expected values (mean, variance, CDF values)
- For many common conjugate families, we can sample directly from the posterior distribution
- Monte Carlo approximation is a convenient way to estimate quantities that are difficult to calculate analytically or numerically
- Monte Carlo can be used to compute predictive distribution for future samples
- Posterior predictive simulation can be useful for diagnosing and evaluating the severity of departures from the assumed model



Random and Pseudorandom Numbers

- Random numbers generated by software are actually pseudorandom numbers
 - A pseudorandom number generator (PRNG) is an algorithm for generating sequences of numbers that pass many tests for randomness
 - These sequences are not actually random!
- A PRNG is initialized with a seed
 - When initialized with a given seed, the PRNG always produces the same sequence of pseudorandom numbers
 - This allows us to reproduce a simulation exactly
 - From an n -bit seed, the sequence repeats after no more than 2^n numbers
- In R we set the seed using `set.seed(seed)`
- Hardware random number generators (HRNG) generate random numbers from a physical process
 - Sequences from a HRNG are not reproducible
 - Some HRNGs (e.g., quantum devices) produce numbers science cannot distinguish from truly random
 - For true random numbers (randomness from atmospheric noise) see <http://www.random.org>

Anyone who considers arithmetical methods of producing random numbers is, of course, in a state of sin. - John von Neumann



Bayesian Monte Carlo: Preview of Coming Attractions

- This unit covered direct Monte Carlo sampling from conjugate posterior distributions
- We need methods for approximating the posterior and predictive distributions for a broader range of problems where we cannot apply dMC
- In later units we will study Markov Chain Monte Carlo (MCMC)
 - MCMC is a general-purpose approach to approximating posterior distributions for a very broad range of inference problems
 - The idea of MCMC is to design a Markov chain whose stationary distribution is the posterior distribution we want to estimate
 - Consecutive samples from such a Markov chain are not independent, but a large enough sample approximates the stationary distribution
 - (For some problems “large enough” is very large!)
- MCMC methods entered the statistics community in the 1980s and have grown increasingly popular
- The two most common MCMC methods are Gibbs sampling (Chapter 6 of Hoff) and Metropolis-Hastings sampling (Chapter 10 of Hoff)



Summary and Synthesis

- Monte Carlo can be used to approximate many aspects of a posterior distribution
 - Density function (histogram / kernel density estimator)
 - Expected values (mean, variance, CDF values)
- For many common conjugate families, we can sample directly from the posterior distribution
- Monte Carlo approximation is a convenient way to estimate quantities that are difficult to calculate analytically or numerically
- Monte Carlo can be used to compute predictive distribution for future samples
- Posterior predictive simulation can be useful for diagnosing and evaluating the severity of departures from the assumed model

