Probability Distributions

Working with Common Statistical Distributions in R

Command	Meaning
rX()	Generate random vector from distribution X
dX()	Return the value of the PDF of distribution X
pX()	Return the value of the CDF of distribution X
qX()	Return the number at which the CDF hits input value [0,1]

Uniform distribution

- **d, p and q functions**

dunif(x, min, max)	Give the probability density function
punif(q, min, max)	Give the cumulative distribution function
qunif(p, min, max)	Give the quantile function

Example:

```
dunif(c(0:5),0,5)

punif(c(0:5),0,5)

qunif(c(0,0.25,0.5,0.75,1),0,5)
```

Generate Random Numbers from Uniform distribution

Uniform: Generate random numbers from uniform distribution
by runif(n,min,max)

Example

runif(5,0,2)

[1] 0.07076444 0.01870595 0.50100158 0.61309213 0.77972391

runif(5)

[1] 0.1705696 0.8001335 0.9218580 0.1200221 0.1836119

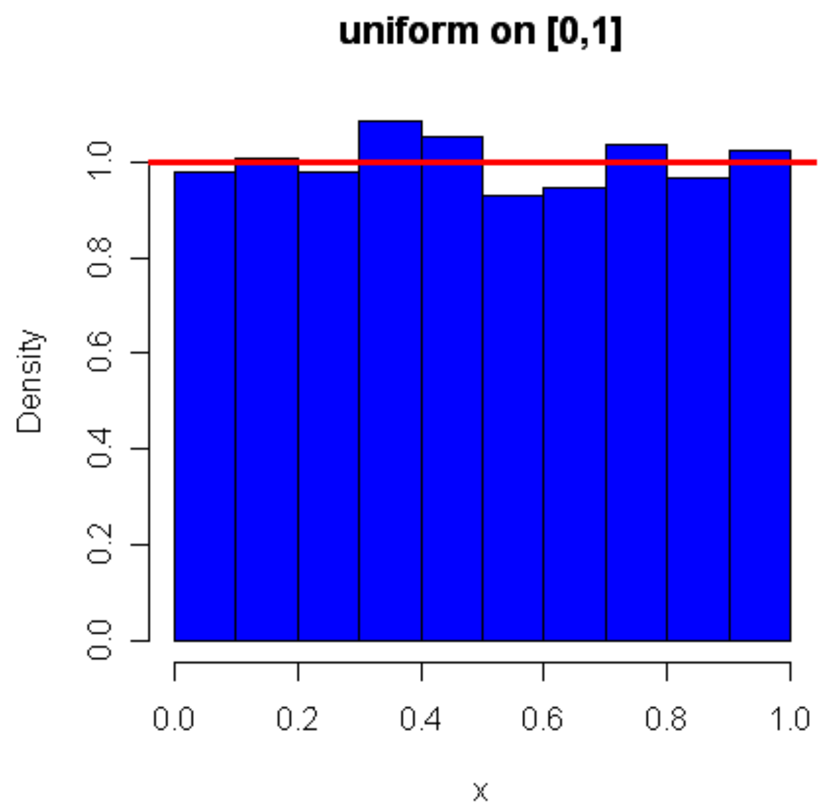
Uniform Distribution

To see the distribution with min=0 and max=1 (the default) we have

Generate 2000 random numbers from uniform distribution

Plot histogram

add density curve of uniform

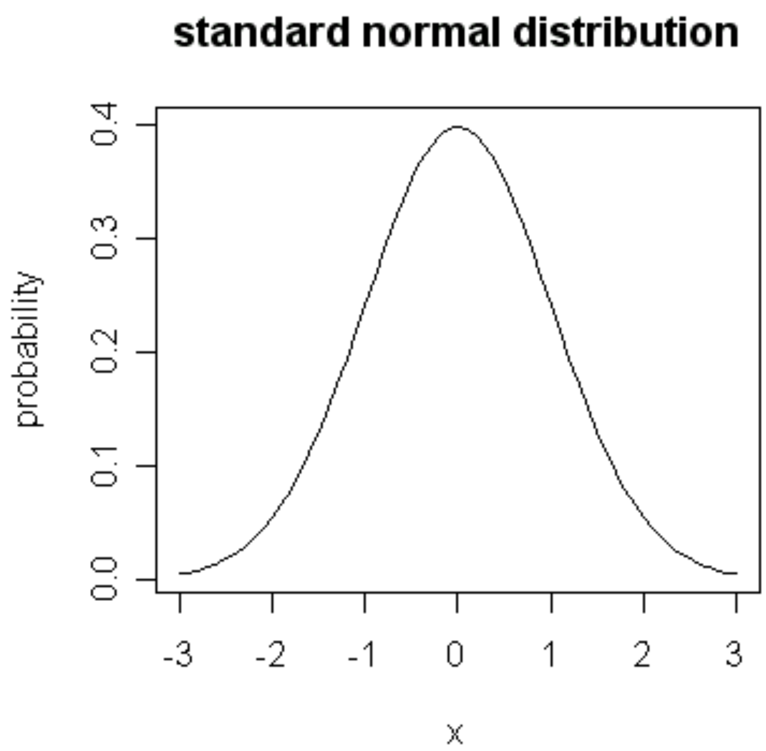


Normal distribution

- **d, p and q functions**

dnorm(x, mean, sd)
pnorm(q, mean, sd)
qnorm(p, mean, sd)

Example 1:
Plot the probability density function for stan normal distribution



Example 2
round(pnorm(c(-3,-1.96,0,1.96,3)),dig=3)

round(qnorm(c(0,0.05,0.5,0.95,1)),dig=2)

Generate random numbers from normal distribution

The function is called as rnorm(n,mean,sd) where one species the mean and the standard deviation.

Example

```
rnorm(1,100,16) # an IQ score
[1] 94.1719
rnorm(5,mean=280,sd=10)
[1] 277.2562 263.8982 264.3409 286.3676 279.1569
```

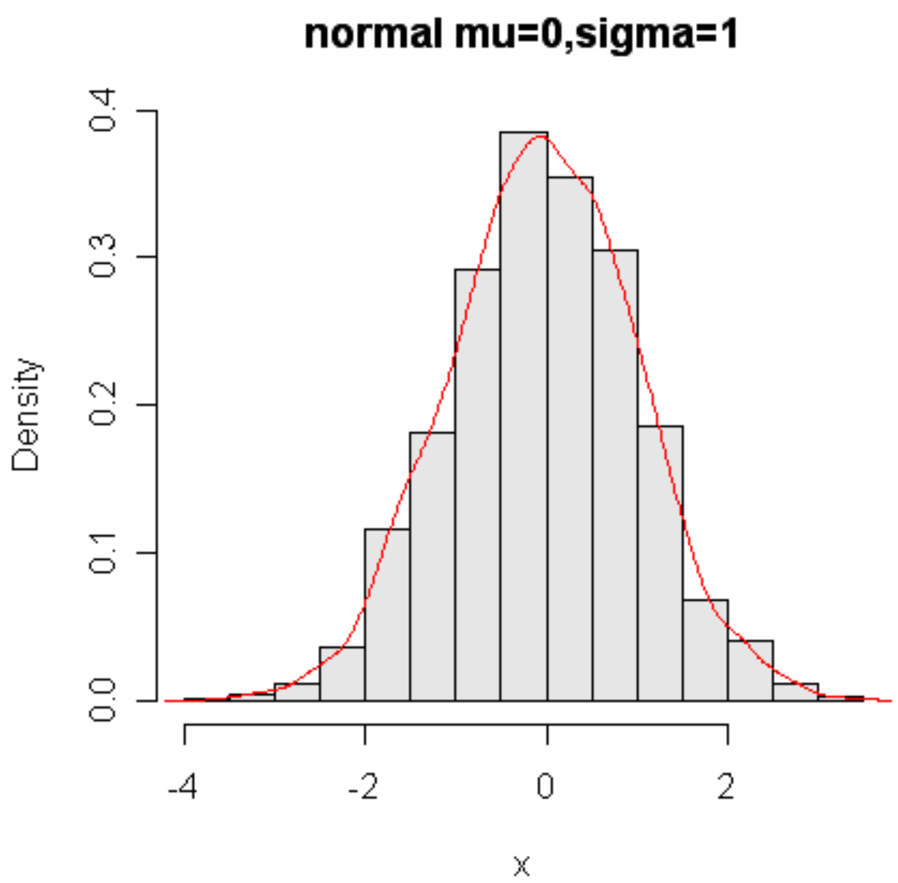
Normal Distribution

To see the shape for the defaults (mean 0, standard deviation 1)

#Generate 2000 random numbers from standard normal distribution

Plot histogram

Add estimated probability density function



Binomial distribution

- **d, p and q functions**

dbinom(x, n, prob)
pbinom(q, n, prob)
qbinom(p, n, prob)

x, vector of the number of successes
n, the number of trails
q, vector of the number of success
p, vector of probabilities
prob, probability of success on each trial

Example 1: Compute P(2 =< X <=4) for X Binomial(10,0.5)

Example 2: qbinom(0.5,10,0.5)

Generate random numbers from Binomial distribution

Binomial. The distribution of the number of successes in n independent Bernoulli trials where a Bernoulli trial results in success or failure, success with probability p.

rbinom(N, n, prob) N: number of observations, n: number of trials, probability of success on each trial.

A single Bernoulli trial is given with n=1 in the binomial

```
n=1, p=.5 # set the probability
rbinom(1,n,p) # different each time
[1] 1
rbinom(10,n,p) # 10 different such numbers
[1] 0 1 1 0 1 0 1 0 1 0
```

Generate random numbers from Binomial distribution

To generate binomial numbers, we simply change the value of n from 1 to the desired number of trials. For example, with 10 trials:

```
> n = 10; p=.5
> rbinom(1,n,p) # 6 successes in 10 trials
[1] 6
> rbinom(5,n,p) # 5 binomial number
[1] 6 6 4 5 4
```

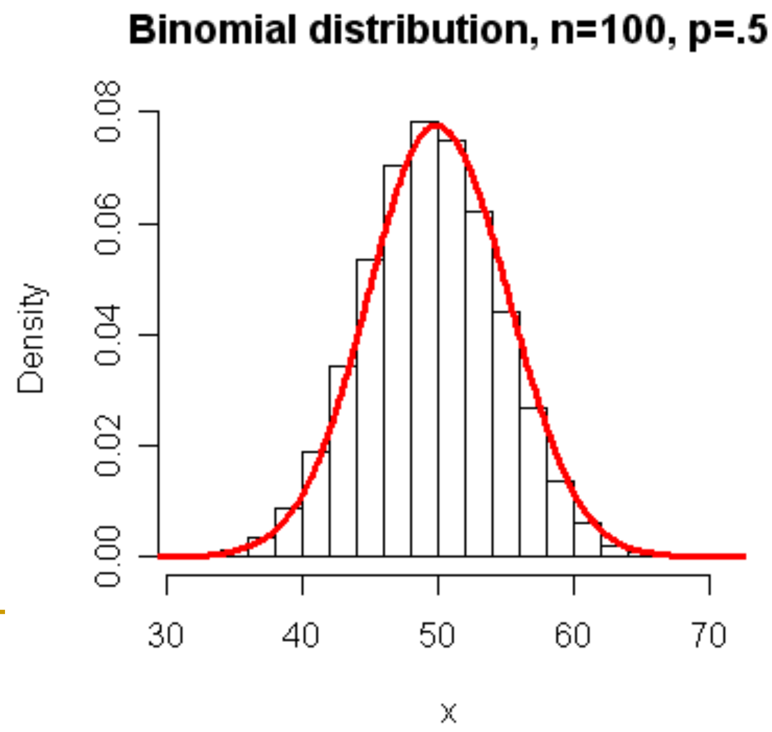
Binomial distribution

Mean and variance of the Binomial distribution B(n,p)

$$\mu = np \qquad \boldsymbol{\sigma^2 = np(1-p)}$$

For large values of *n,* the distributions of the count *X* is approximately normal. The mean and variance for the approximately normal distribution of *X* are *np* and *np(1-p).*

Use a simulation to demonstrate that the binomial distribution can be approximated by the normal distribution



Poisson Probability Distribution

- **d, p and q functions**

dpois(x, lambda)
ppois(q, lambda)
qpois(p, lambda)

Example: Records indicate that, on average, 3 breakdowns per day occur on an urban highway during morning rush hour. Assume that the distribution is poisson.

a. Find the probability that on any given day there will be fewer than two breakdowns on this highway during the morning rush hour.

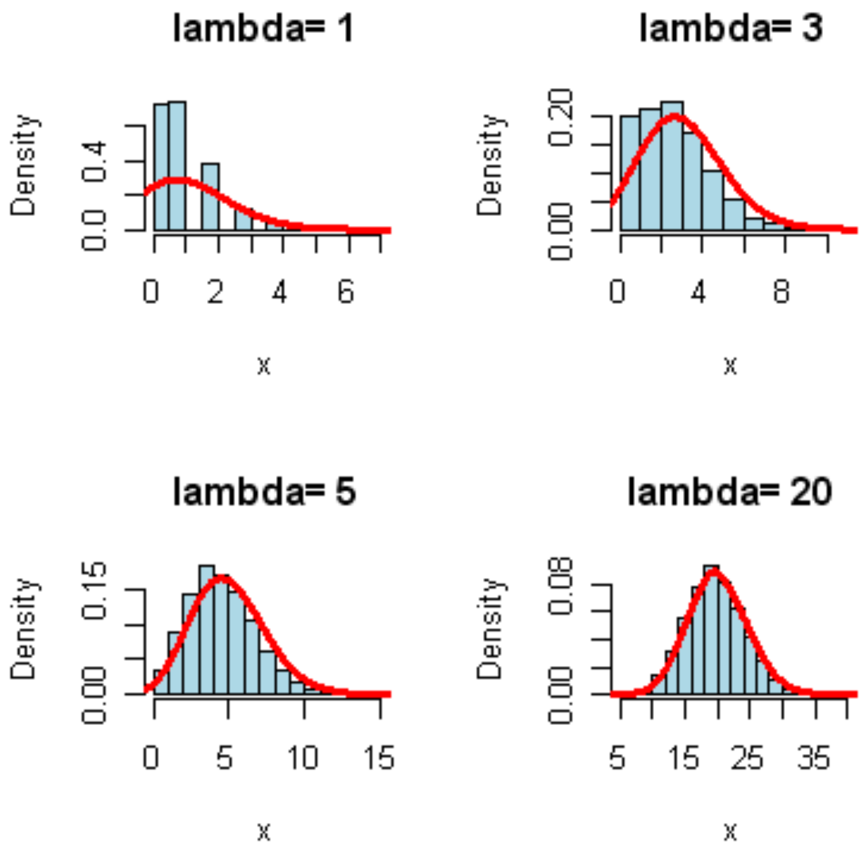
b. Find the probability that on any given day there will be more than four breakdowns on this high way during the morning rush hour?

a.

b.

Generate random numbers from Poisson distribution (rpois (n,lambda))

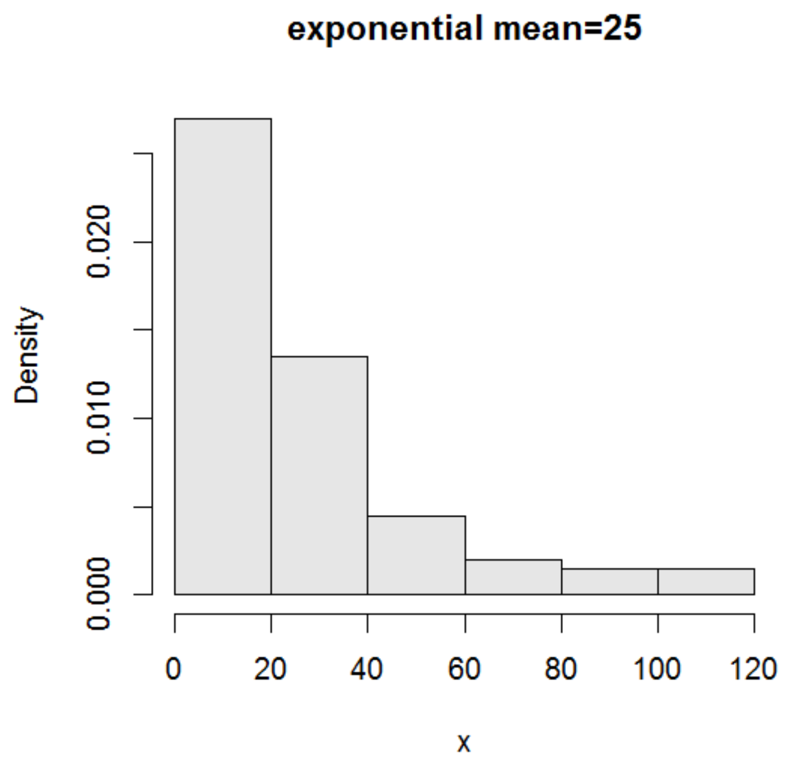
```
op <- par(mfrow=c(2,2))
N <- 10000
lambda<-c(1,3,5,20)
for (i in c(1:4))
{x <- rpois(N, lambda[i])
hist(x, xlim=c(min(x),max(x)),
probability=T, col="lightblue",
main=paste("lambda=", lambda[i] ))
lines(density(x,bw=1), col="red",
lwd=3)
}
par(op)
```



Exponential Distribution

dexp(x, rate)
pexp(q, rate)
qexp(p, rate)
rexp(n, rate)

x<-rexp(100,1/25)
hist(x,probability=TRUE,col=gray(.9),main="exponential mean=25")



Sampling with and without replacement using command "sample"

By default "sample" samples without replacement each object having equal chance of being picked. You need to specify replace=TRUE if you want to sample with replacement. Furthermore, you can specify separate probabilities for each if desired.

```
## Roll a die
> sample(1:6,10,replace=TRUE)
[1] 5 1 5 3 3 4 5 4 2 1
## toss a coin
> sample(c("H","T"),10,replace=TRUE)
[1] "H" "H" "T" "T" "T" "T" "H" "H" "T" "T"
## pick 6 of 54 (a lottery)
> sample(1:54,6) # no replacement
[1] 6 39 23 35 25 26
```

Appendix: Probability Distribution in R

Distribution	R Name	Possible Arguments
beta	`beta`	shape1, shape2, ncp
binomial	`binom`	size, prob
Cauchy	`cauchy`	location, scale
chi-squared	`chisq`	df, ncp
exponential	`exp`	rate
F	`f`	df1, df1, ncp
gamma	`gamma`	shape, scale
geometric	`geom`	prob
hypergeometric	`hyper`	m, n, k
log-normal	`lnorm`	meanlog, sdlog
logistic	`logis`	location, scale
negative binomial	`nbinom`	size, prob
normal	`norm`	mean, sd
Poisson	`pois`	lambda
Students t	`t`	df, ncp
uniform	`unif`	min, max
Weibull	`weibull`	shape, scale
Wilcoxon	`wilcox`	m, n

Probability Density Estimation

Outline

Methods of Parametric Estimation

- Method of Moments
- Maximum likelihood estimation

Methods of Non-Parametric Estimation

- Kernel density estimation

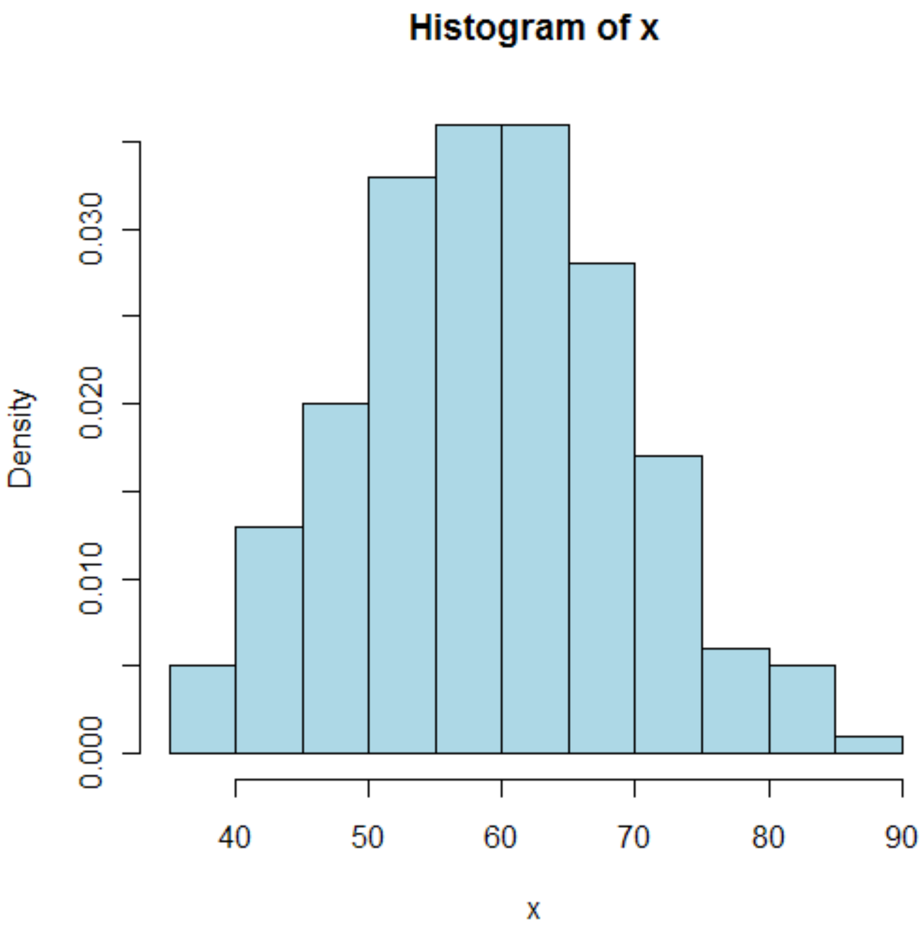
Example 1

- A random sample of 100 students' math scores from UIC is given as follows:

73 60 47 63 53 60 61 59 51 60 82 59 54 60 68 49 52 69 61 55 55 73 69 65 64 47 50 58 47 60 79 73 51 84 53 64 52 69 56 63 71 66 64 67 72 57 66 71 72 66 57 57 52 70 70 62 66 62 64 59 59 76 40 51 73 45 48 65 50 42 48 58 74 50 58 56 71 61 56 85 62 61 55 59 54 55 76 60 49 51 62 74 63 59 41 62 52 57 44 62 55 69 72 76 59 52 50 52 53 60 65 42 65 47 76 53 57 74 58 50 45 64 59 69 73 70 61 61 50 70 39 58 42 52 36 85 61 54 42 62 45 69 70 64 54 47 59 56 60 42 68 52 66 52 42 47 51 49 67 89 54 74 45 49 48 67 73 53 69 64 59 66 68 51 82 61 63 69 63 72 60 68 61 62 65 64 52 55 48 57 36 66 60 40 66 62 56 42 55 76

- Estimate the probability that a student's math score is higher than 80.

Example 1 (Continuity)



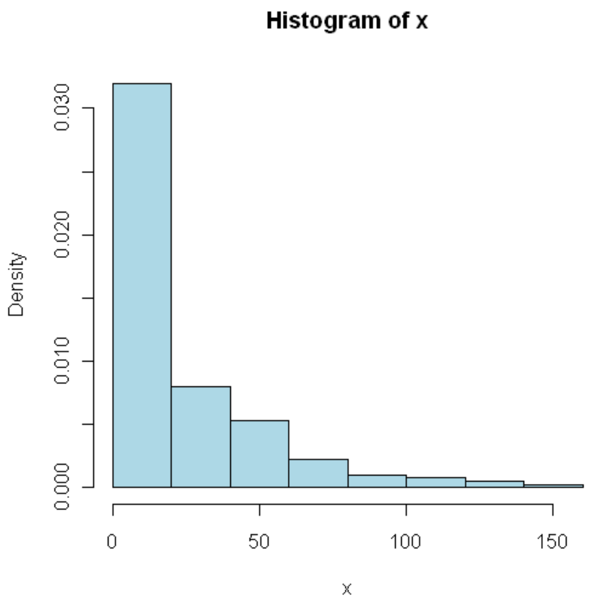
Example 2

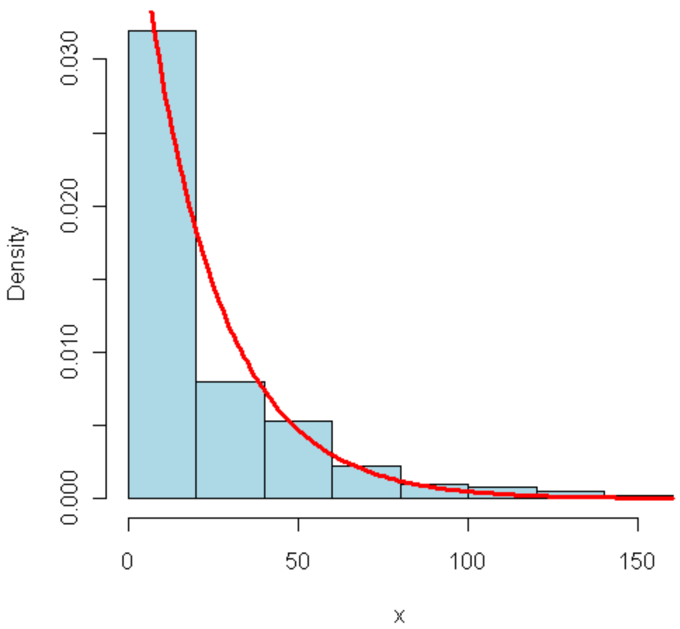
- **The amounts paid on workers compensation medical benefits are investigated. A random sample of 200 payments is given as follows: (in $100)**

2.3 2.8 0.3 13.1 21.1 1.5 1.5 12.3 15.9 4.2 0.1 0.3 3.5 13.5 1.2 10.9 95.4
14.2 23.3 14.9 14.9 41.6 10.0 74.2 0.3 35.0 3.3 1.8 33.3 8.9 2.2 42.7 8.5 7.1
0.2 102.5 4.2 0.3 0.5 16.1 1.4 42.4 19.5 6.1 0.7 11.9 5.7 52.0 23.1 20.3 0.9
15.3 34.5 1.3 23.3 43.2 8.4 1.3 0.1 0.1 57.0 6.3 50.9 0.1 0.5 22.5 129.8 7.0
91.2 2.7 0.2 50.5 8.5 25.8 47.2 0.1 4.4 14.4 92.7 2.9 1.5 8.2 96.0 10.9 1.7
142.7 27.6 9.6 31.5 0.8 54.3 14.7 36.8 4.7 14.7 10.7 42.5 21.2 0.1 36.1 3.1 0.9
17.1 62.5 26.3 1.5 19.0 74.0 0.0 0.5 12.0 5.8 15.6 34.6 55.1 23.1 0.3 35.4 3.2
21.1 9.6 33.3 41.5 7.5 0.7 32.7 40.8 12.9 33.8 100.9 0.1 57.8 7.5 20.5 28.4 45.1
34.4 68.7 2.9 2.2 0.1 66.4 123.2 46.2 15.8 0.1 2.0 0.2 8.2 5.9 19.4 0.5 9.0
7.2 10.7 51.7 0.1 1.2 0.1 5.1 38.8 5.0 47.4 4.4 2.4 25.4 11.4 37.3 11.0 46.7
13.3 63.9 0.1 17.2 11.9 35.3 6.0 4.4 4.4 0.3 28.0 60.3 66.6 9.2 51.5 117.6 23.4
10.1 0.1 4.8 75.8 10.9 19.5 1.5 2.9 9.3 11.2 14.4 1.5 1.2

- **What is the distribution of the amounts paid on workers compensation medical benefits?**

Example 2 (Continuity)

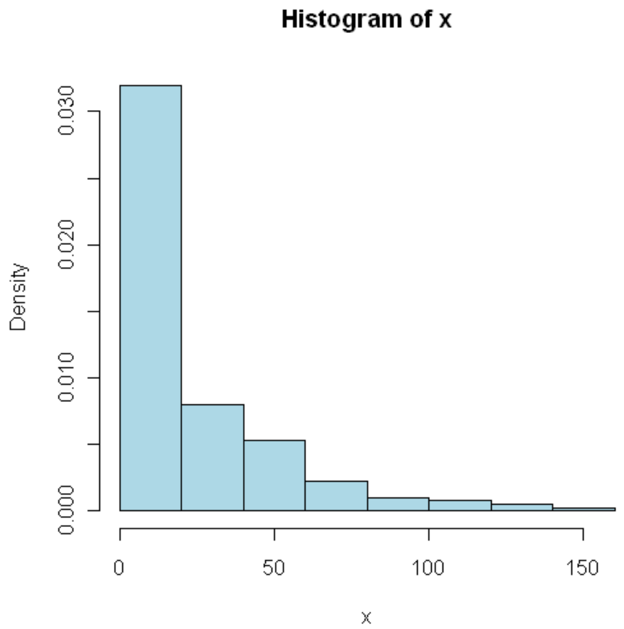


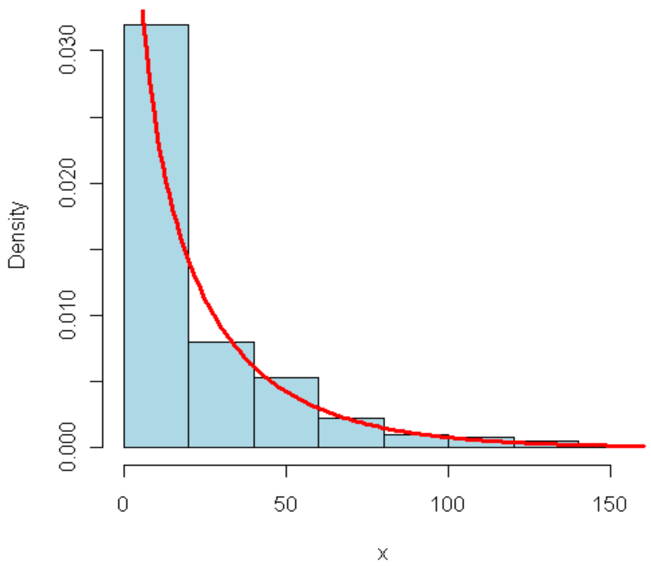


X~ Exponential (λ)

λ =?

Example 2 (Continuity)





$X \sim$ Gamma(α, β)

Example 3

- **An insurance company received the claim counts for a year as the following table**

No. of claims	**No. of policies**
0	**9048**
1	**905**
2	**45**
3	**2**
4+	**0**

Estimate λ for a Poisson model.

Moment

Suppose a random variable X has density $f(x;\theta)$. The k-th *theoretical moment* of this random variable is defined as

$$\mu_k = E(X^k) = \int_{-\infty}^{\infty} x^k f(x \mid \theta) dx.$$

where the random variable is continuous

Or

$$\mu_k = E(X^k) = \sum_x x^k f(x \mid \theta),$$

where the random variable is discrete. The sum is over all x_j with positive probabilities

Method of Moments Estimation (MME)

Let $X_1, X_2, \ldots, X_n$ be a r.s. from a population with pmf or pdf $f(x; \theta_1, \theta_2, \ldots, \theta_k)$. The MMEs are found by equating the first k population moments to corresponding sample moments and solving the resulting system of equations.

Population Moments

$$\mu_k = E\left[X^k\right]$$

Sample Moments

$$M_k = \frac{1}{n}\sum_{i=1}^{n} X_i^k$$

Method of Moments Estimation (MME)

$$\mu_1 = M_1 \qquad \mu_2 = M_2 \qquad \mu_3 = M_3$$

$$E(X) = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad E(X^2) = \frac{1}{n}\sum_{i=1}^{n} X_i^2 \qquad E(X^3) = \frac{1}{n}\sum_{i=1}^{n} X_i^3$$

so on…

Continue this until there are enough equations to solve for the unknown parameters

Example 4: Use the method of moment to estimate the parameters μ and σ^2 for the normal density.

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\},$$

based on a random sample $X_1, \cdots, X_n$.

Solution: The first and second theoretical moments for the normal distribution are

$$\mu_1 = E(X) = \mu \quad \text{and} \quad \mu_2 = E(X^2) = \mu^2 + \sigma^2$$

We have the method of moment estimate

$$\hat{\mu} = \overline{X} \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \overline{X}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

Example 5 : Suppose that X is a discrete random variable with the following probability mass function: where $0 \leq \theta \leq 1$ is a parameter. The following 10 independent observations

X	0	1	2	3
$P(X)$	$2\theta/3$	$\theta/3$	$2(1-\theta)/3$	$(1-\theta)/3$

were taken from such a distribution: (3,0,2,1,3,2,1,0,2,1). Please use the method of moment to find the estimate of θ.

Solution: The theoretical mean value is

Question:

$X \sim Exp(\theta)$. For a r.s of size n, find the MME of θ.

$X \sim Gamma(\alpha, \beta)$. For a r.s of size n, find the MMEs of α and β.

Drawbacks of MMES

- Although sometimes parameters are positive valued, MMEs can be negative.
- If moments does not exist, we cannot find MMEs.
- They are often (usually) not the best estimators available

Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation (MLE)

The MLE is the parameter point for which the observed sample is most likely.

step 1. The likelihood function $L(\theta; X) = \prod_{i=1}^{n} f(x_i; \theta)$

step 2. The parameter space (domain) Ω

step 3. Maximization: to find a $\theta^* \in \Omega$ such that

$$L(\theta^*; X) = \max_{\theta \in \Omega} L(\theta; X).$$

The value θ^* is called maximum likelihood estimator of θ.

and $L(\theta^*; X)$ is called maximum likelihood.

Actually, the log of the likelihood function is more convenient in the maximization step

$$l(\theta; X) = \log L(\theta; X) = \sum_{i=1}^{n} \log f(x_i, \theta), \ \theta \in \Omega.$$

Example 6. Let $X_1, ..., X_n$ be independent identically distributed random variables with pdf

$$f(x) = \frac{1}{\lambda} x^{\frac{1}{\lambda} - 1},$$

where $\lambda > 0$ and $0 < x \leq 1$.

a) Find the maximum likelihood estimator of λ.

Example **7.** (Laplace distribution).

$$f(x;\theta) = \frac{1}{2}e^{-|x-\theta|}, \quad -\infty < x < \infty, -\infty < \theta < \infty$$

Find the Maximum Likelihood Estimator of θ.

n

step 1. $l(\theta) =$

step 2. $\Omega =$

step 3.

ADVANTAGES OF MLE

- Often yields good estimates, especially for large sample size.

- Usually they are consistent estimators.

- Invariance property of MLEs

- Asymptotic distribution of MLE is Normal.

- Most widely used estimation technique.

DISADVANTAGES OF MLE

• Requires that the pdf or pmf is known except the value of parameters.

• MLE may not exist or may not be unique

.

• MLE may not be obtained explicitly (numerical or search methods may be required.). It is sensitive to the choice of starting values when using numerical estimation.

• MLEs can be heavily biased for small samples.

• The optimality properties may not apply for small samples.

Maximum Likelihood in R

MLE in many cases have explicit formula. For the studied examples, in the previous lectures, we are lucky that we can find the MLE by solving equations in closed form.

But life is never easy. In applications, we usually don’t have closed form solutions due to the complicated probability distribution or the nonlinearity of the equations obtained from maximum likelihood principles. In these situations, we can use a computer to solve the problem.

Many statistics software package has MLE as a standard procedure, but for the purpose of learning MLE and for the purpose of learning programming language, let us develop the code ourselves.

Maximum Likelihood in R

Maximum likelihood estimation involves an optimization problem.

Step 1: Use R to define your log likelihood function

Step 2: Use some optimization technique.
(uniroot , optimize, nlm, optim)

Sometimes you also need to write your score (the first derivative of the log likelihood) and or the hessian (the second derivative of the log likelihood).

Optimization Technique

uniroot (f, interval) #The function uniroot searches the interval from lower to upper for a root (i.e., zero) of the function f with respect to its first argument.
f: the function for which the root is sought.
Interval: a vector containing the end-points of the interval to be searched for the root.

optimize(f = , interval =) # optimize searches the interval from lower to upper for a minimum or maximum of the function f with respect to its first argument.
f: the function to be optimized. The function is either minimized or maximized over its first argument depending on the value of maximum.
Interval: a vector containing the end-points of the interval to be searched for the minimum

nlm() #carry out a minimization of the function f using a Newton-type algorithm
nlm(f, p)
f: the function to be minimized.
p: starting parameter values for the minimization

Example 10: The Cauchy distribution

The Cauchy density with location parameter θ is unknown and scale parameter σ=1 is

$$f(x) = \frac{1}{\pi[1 + (x - \theta)^2]}, \quad -\infty < x < \infty$$

Based on n observations $x_1, x_{2,\ldots}\ x_n$, the likelihood function is

$$L(\theta) = \prod_{i=1}^{n} f(x_i)$$

and the loglikelihood function is

$$\begin{aligned} \log L(\theta) = \ell(\theta) &= \sum_{i=1}^{n} \log f(x_i) \\ &= -n \log \pi - \sum_{i=1}^{n} \log[1 + (x - \theta)^2] \end{aligned}$$

Example: The Cauchy distribution

The log likelihood can be written in R either as

```
mlogl <- function(mu, x) {
sum(-dcauchy(x, location = mu, log = TRUE))
 }
```

using dcauchy to avoid having to know the formula for the densities

or as

```
mlogl2 <- function(mu, x) {
sum(log(1 + (x - mu)^2))
 }
```

using our knowledge of the Cauchy densities.

The first and second derivatives of the log likelihood are, respectively.

$$\frac{\partial \ell}{\partial \theta} = \sum_{i=1}^{n} \frac{2(x_i - \theta)}{1 + (x_i - \theta)^2} \quad \text{and} \quad \frac{\partial^2 \ell}{\partial \theta^2} = \sum_{i=1}^{n} \frac{2(x_i - \theta)^2 - 2}{(1 + (x_i - \theta)^2)^2}$$

It is hard to get analytic solution. Let's try to numerical solution obtained by R

The optimization procedure based on mlogl and mlogl2 produces the same results on the simulated data.

```
n <- 500
set.seed(42)
x <- rcauchy(n)
# generate random number from cauchy distribution with θ=0 and σ=1.

mu.start <- median(x) #starting parameter values for the minimization
```

```
out <- nlm(mlogl, mu.start, x = x)
mu.hat <- out$estimate
mu.hat
[1] 0.05710308
```

```
out2 <- nlm(mlogl2, mu.start, x = x)
mu.hat <- out$estimate
mu.hat
[1] 0.05710308
```

Alternative:

$$\frac{\partial \ell}{\partial \theta} = \sum_{i=1}^{n} \frac{2(x_i - \theta)}{1 + (x_i - \theta)^2} = 0$$

```
fmlog<-function(mu,x)  sum((x-mu)/(1+(x-mu)^2))
out <- uniroot(fmlog, c(-5,5), x=x)
out
$root
[1] 0.05708338
```

Alternative:

```
out2 <- optimize(mlogl2, c(-5,5), x =
x)
out2
$minimum
[1] 0.05710469
```

Example 11: The Cauchy Distribution with Two Parameters

$$f(x;\boldsymbol{\theta}) = \frac{\theta_2}{\pi[\theta_2^2 + (x-\theta_1)^2]}, \quad -\infty < \theta_1 < \infty, \ \ \theta_2 > 0.$$

Use the same simulation procedure

```
n <- 500;    set.seed(42);   x <- rcauchy(n)
```

Minus the log likelihood for the two-parameter Cauchy can be written

and the MLE calculated by

Using Derivative Information in Optimization

$$\frac{\partial \ell}{\partial \theta_1} = 2\sum_{i=1}^{n} \frac{(x_i - \theta_1)}{[\theta_2^2 + (x_i - \theta_1)^2]}$$

$$\frac{\partial \ell}{\partial \theta_2} = \frac{n}{\theta_2} - 2\sum_{i=1}^{n} \frac{\theta_2}{[\theta_2^2 + (x_i - \theta_1)^2]}$$

Using Derivative Information in Optimization

In its default mode of operation **nlm** uses derivatives calculated by finite differences. It will work better and faster if we supply the derivatives.

```
mlogl4 <- function(theta, x) {
```

defines minus the log like and adds an attribute "gradient", which is the gradient. So let’s see how it works.

```
out <- nlm(mlogl4, theta.start, x = x)
out$estimate
[1] 0.05654482 0.96209048
```

Example 12:

Suppose the observations $X_1, X_2, ..., X_n$ are from $N(\mu, \sigma^2)$ distribution (2 parameters: μ and σ^2).

The log likelihood function is

$$\sum_{i=1}^{n}\left(-\frac{(x_i-\mu)^2}{2\sigma^2}-\frac{1}{2}(\log\sigma^2)-\frac{1}{2}\log(2\pi)\right)$$

R solution:

```
xvec <-random(500,mean=4,sd=1.5)
```

Kernel density estimation

- Methods of Non-Parametric Estimation

Nonparametric estimation

- *What if I don't want to specify a simple parametric form?*
- *No assumption about the form of probability distribution.*

Histogram

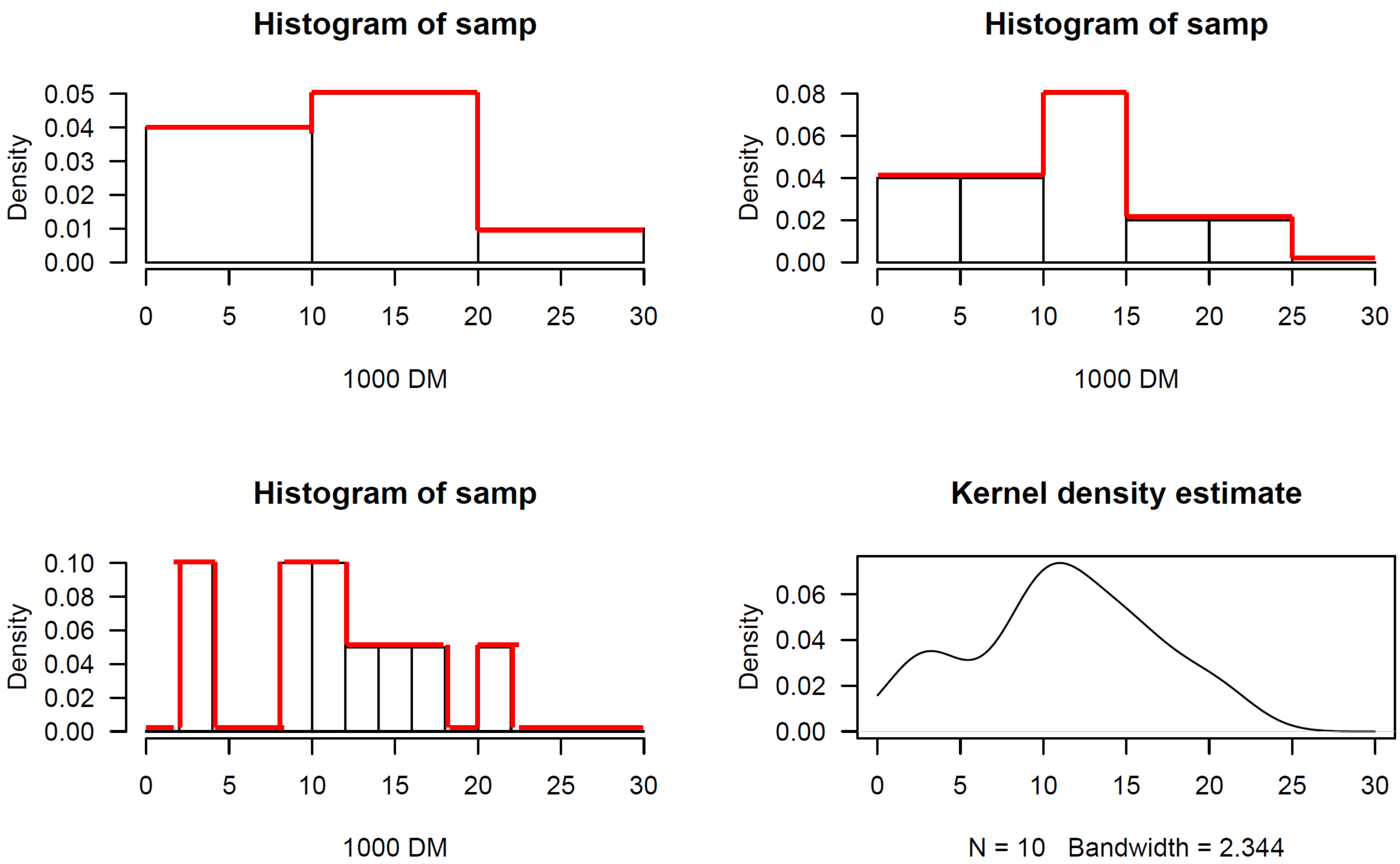
A well–known non–parametric estimator of the probability density function is the histogram.

Histogram has the advantage of simplicity but it also has disadvantage, such as

- ➢ lack of continuity
- ➢ an observation may be closer to an observation in the neighboring bin than it is to points in its own bin
- ➢ the estimated *p.d.f.* appears to be significantly different due to the choice of number of bins.

Histogram

Histograms with different bin widths and a kernel estimate of $f(x)$ for the same sample.



Local neighborhood density

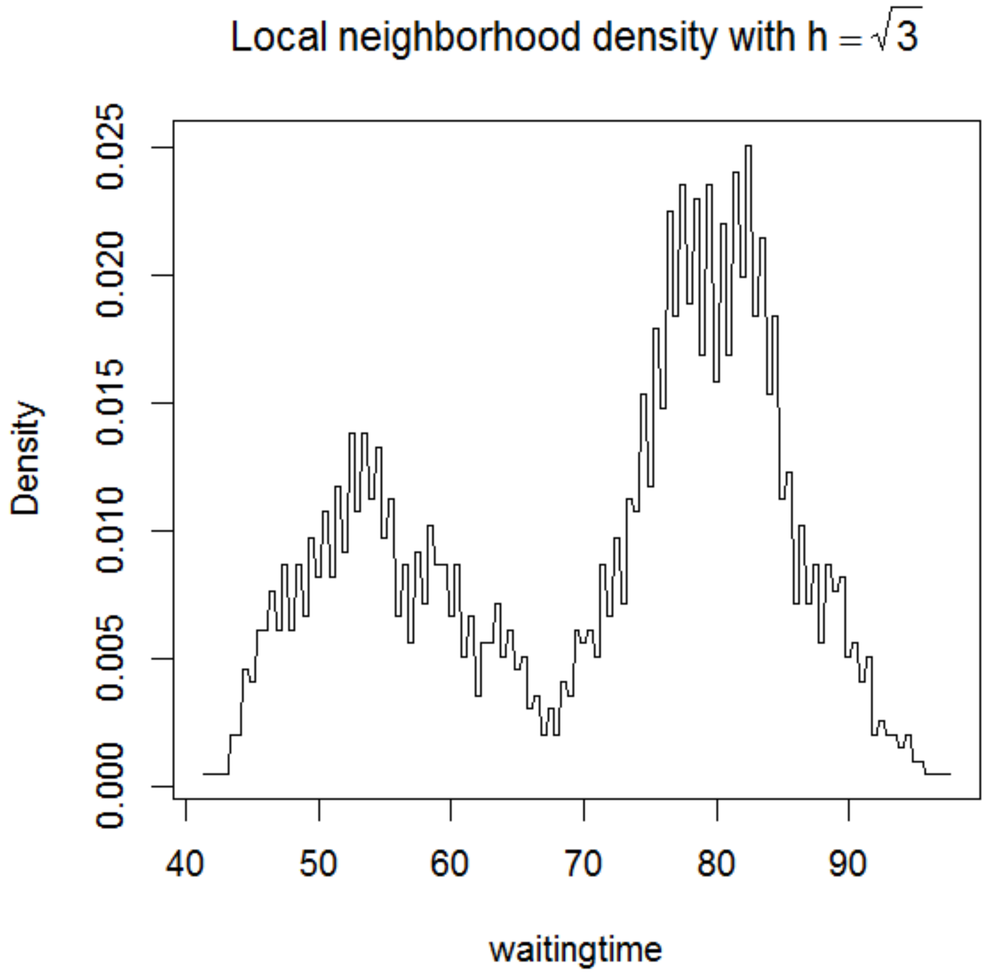
For example, consider estimating the density at a point x_0 by taking the local density of the points within distance h of x_0:

$$\hat{f}(x_0) = \frac{n^{-1} \sum_i I(|x_i - x_0| \leq h)}{2h}$$

This solves one problem of the histogram -namely, it ensures that no point further away from x_0 than x_i will contribute more than x_i does to the density estimate

Local neighborhood density: Old Faithful data

However, the resulting density estimate is still bumpy:



Kernel density estimate

Let's x the bumpiness: instead of giving every point in the neighborhood equal weight, let's assign a weight which dies off toward zero in a continuous fashion as we get further away from the target point x_0

Specially, consider estimators of the following form:

$$\hat{f}(x_0) = \frac{1}{nh} \sum_i K\left(\frac{x_i - x_0}{h}\right),$$

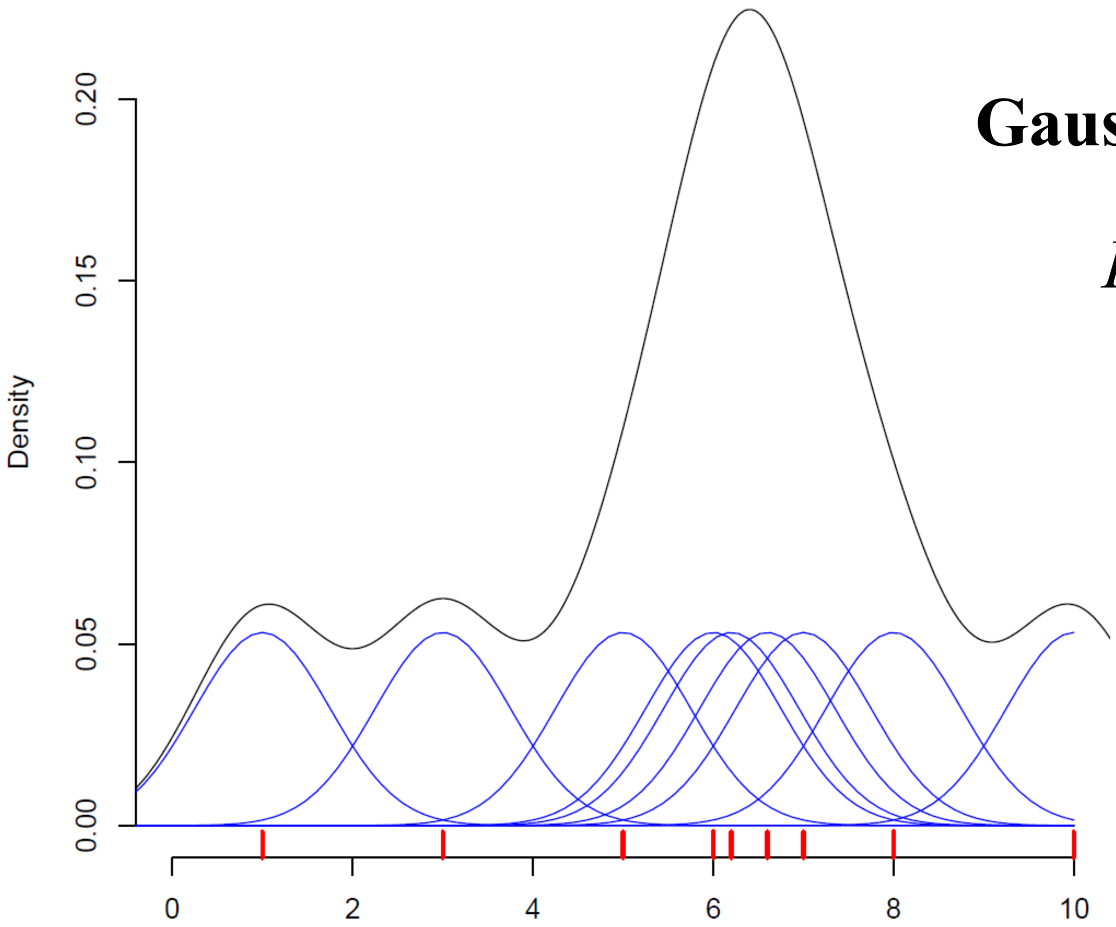
where h, which controls the size of the neighborhood around x_0, is the smoothing parameter.
The function K is called the kernel, and it controls the weight given to the observations $\{x_i\}$ at each point x_0 based on their Proximity.

Kernel properties

To yield meaningful estimates, a kernel function must satisfy four properties:

- $K(u) \geq 0$
- Symmetric about 0
- $\int K(u)du = 1$
- $\int u^2 K(u)du > 0$

Gaussian kernel: density estimate



Gaussian kernel

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2} \right)$$

Other kernels

One drawback of the Gaussian kernel is that its support runs over the entire real line; occasionally it is desirable that a kernel have compact support

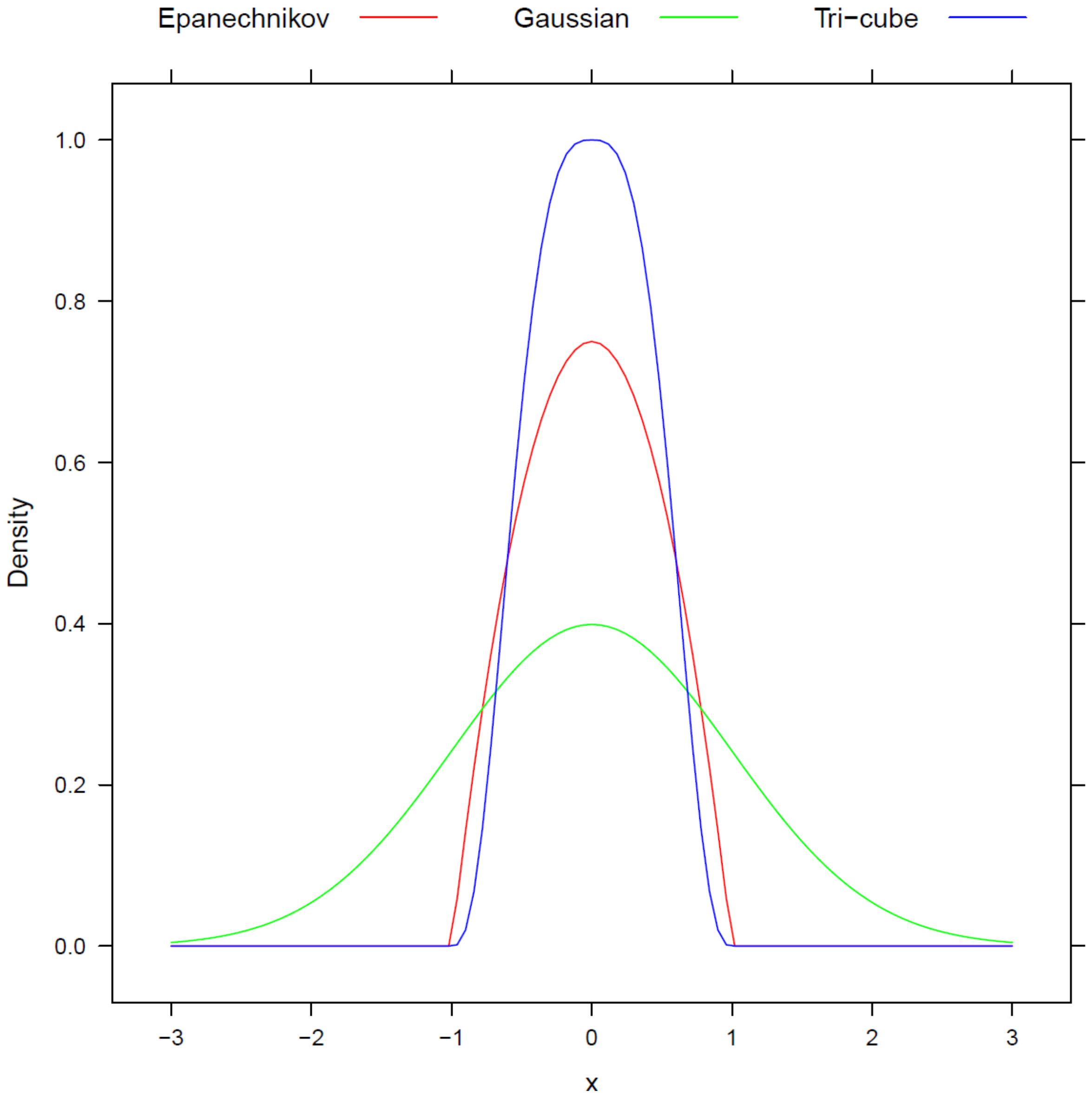
Two popular compact kernels are the Epanechnikov kernel:

$$K(u) = \begin{cases} \frac{3}{4}(1 - \boldsymbol{u^2}) & if |u| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

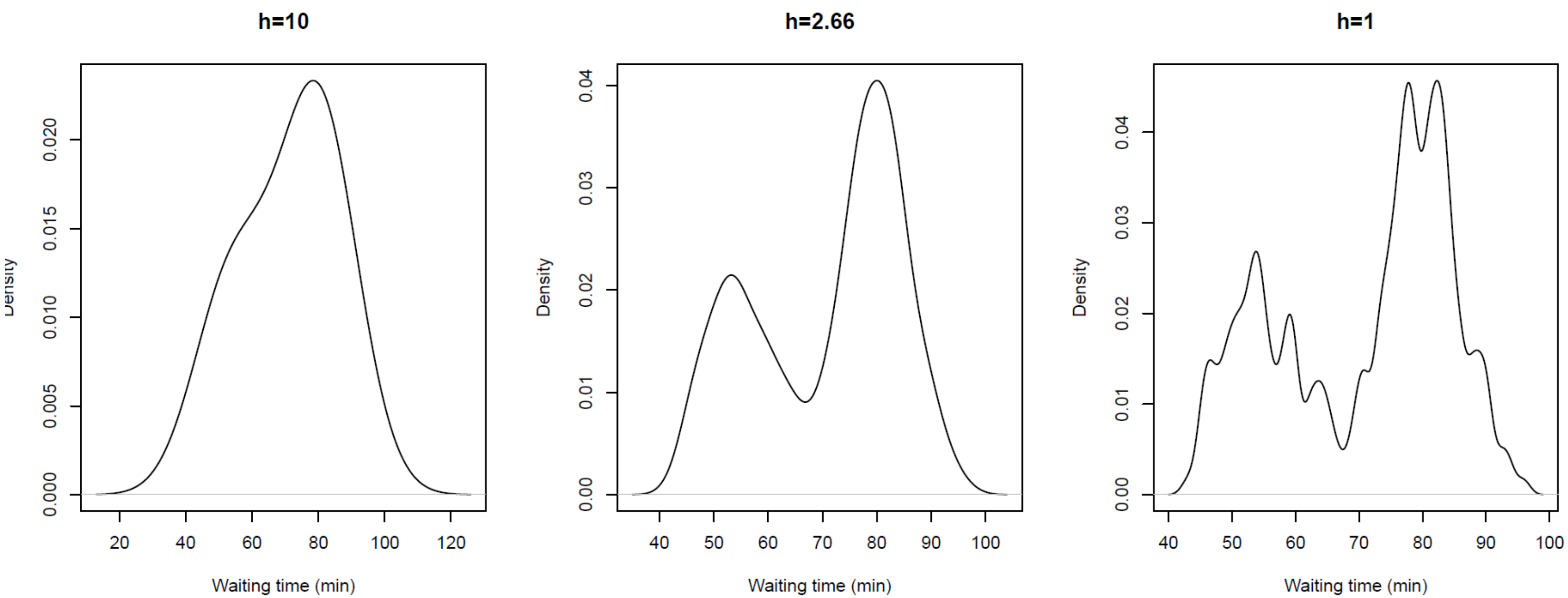
and the tri-cube kernel:

$$K(u) = \begin{cases} (1 - |u|^3)^3 & if |u| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Kernels: illustration



Effect of changing bandwidth



Kernel density estimate in R

Kernel density estimates are available in R via the density function:

```
d <- density(faithful$waiting)
plot(d)
```

By default, density uses a Gaussian kernel, but a large variety of other kernels are available by specifying the kernel Option.

Bandwidth specification

By default, density selects the bandwidth based on the normal reference rule

However, you can manually choose the bandwidth by specifying, for example, bw=4

You can also obtain automatic selection by cross-validation by specifying bw=‘ucv’

Multivariate densities (Optional)

It is straightforward to extend the idea of kernel density estimation to obtain multidimensional densities:

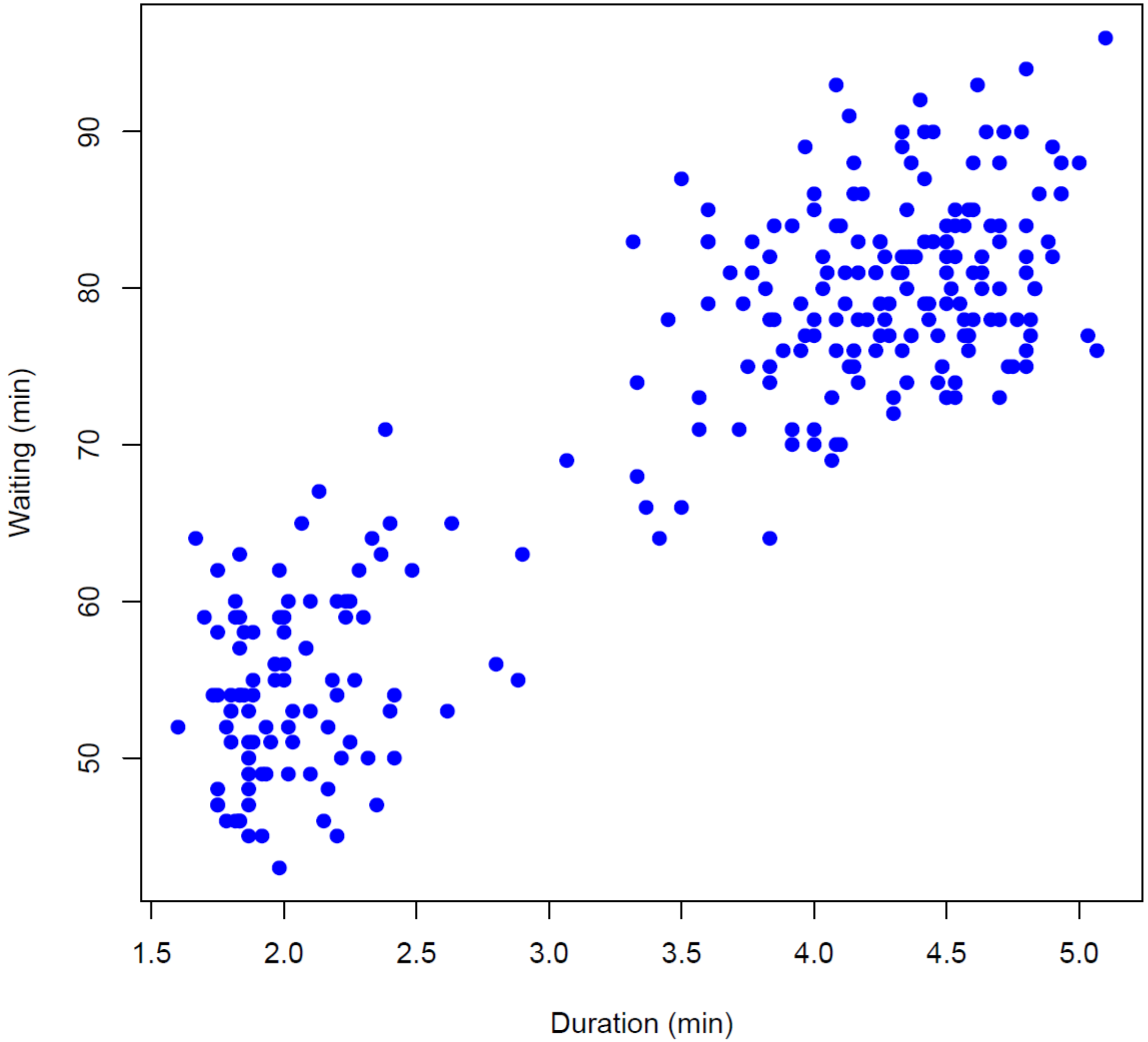
$$\hat{f}(\mathbf{x}_0) = \frac{1}{nh^p} \sum_i K\left(\frac{\|\mathbf{x}_i - \mathbf{x}_0\|}{h}\right),$$

where p is the dimension of *x*

This can be further generalized by allowing different bandwidths in each dimension:

$$\hat{f}(\mathbf{x}_0) = \frac{1}{n} \sum_i \prod_{j=1}^{p} \frac{1}{h_j} K\left(\frac{x_{ij} - x_{0j}}{h_j}\right)$$

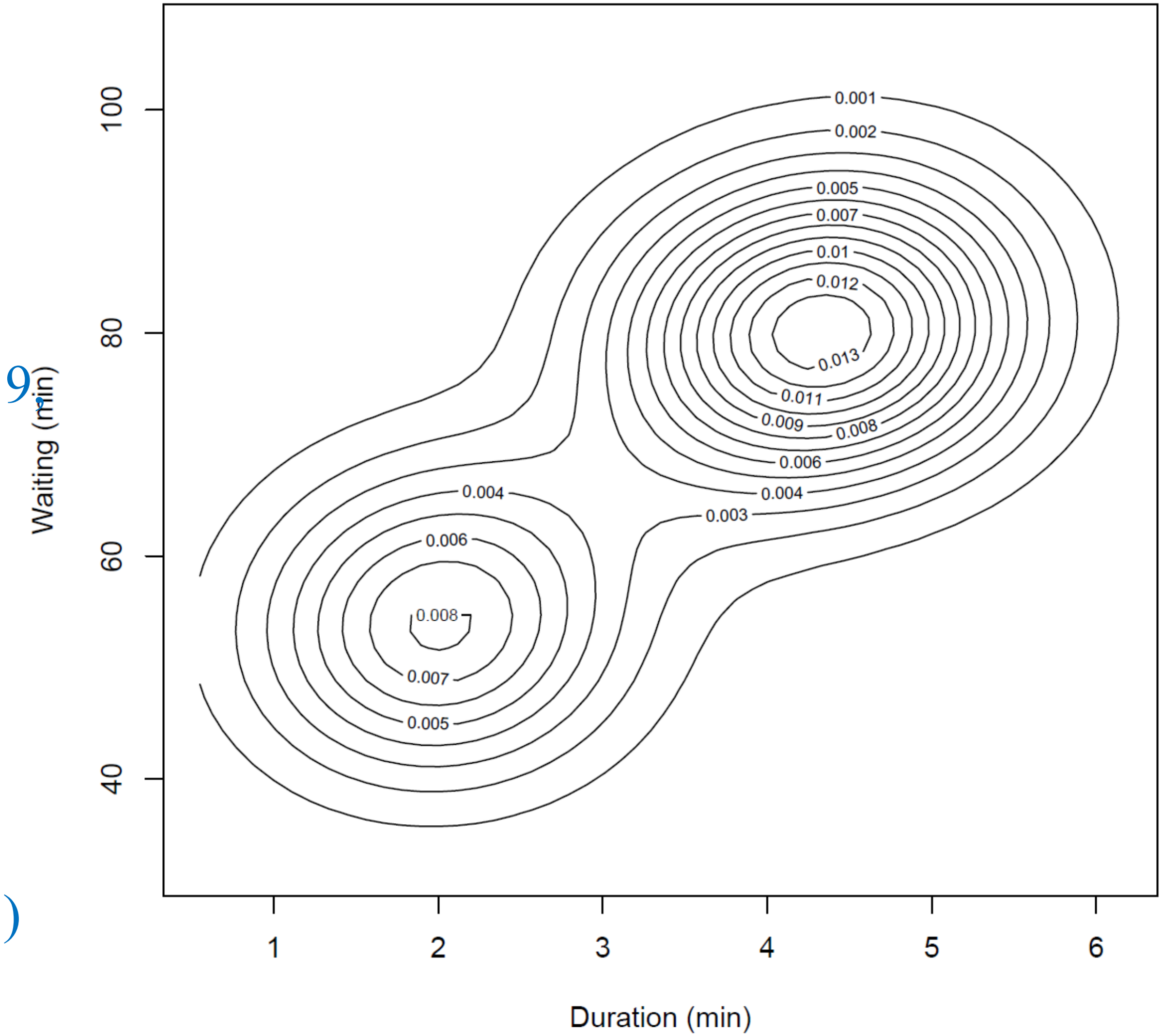
Old faithful data (Optional)



Old faithful 2D density estimate: contour plot

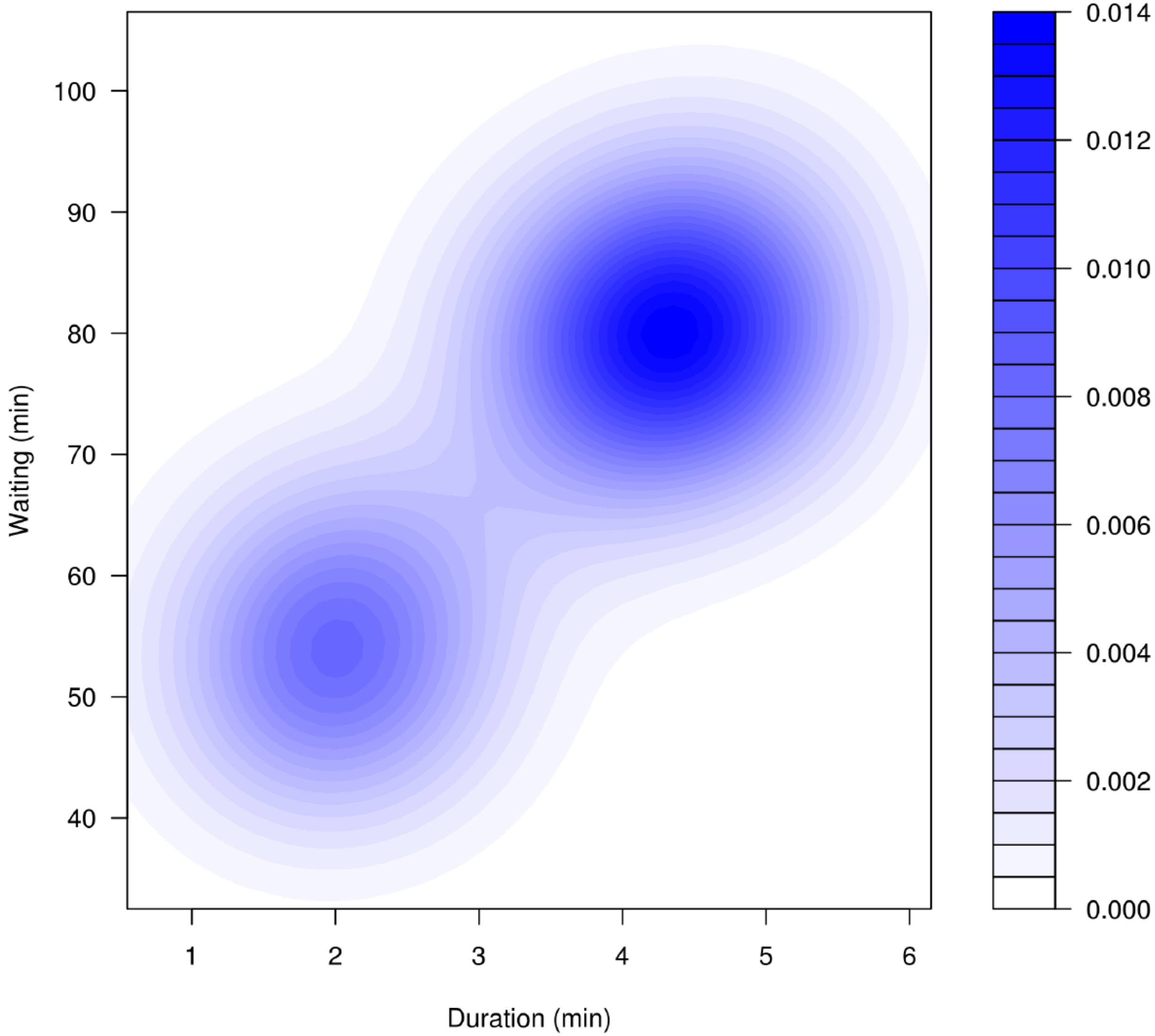
(Optional)

```
# 2D
library(KernSmooth)
with(faithful,
plot(eruptions,waiting,pch=19,
xlab="Duration
(min)",ylab="Waiting
(min)",col="blue"))
fit <- bkde2D(faithful,
bandwidth=c(0.7, 7))
contour(fit$x1, fit$x2,
fit$fhat,xlab="Duration
(min)",ylab="Waiting (min)")
```



Old faithful 2D density estimate: filled contour plot (Optional)

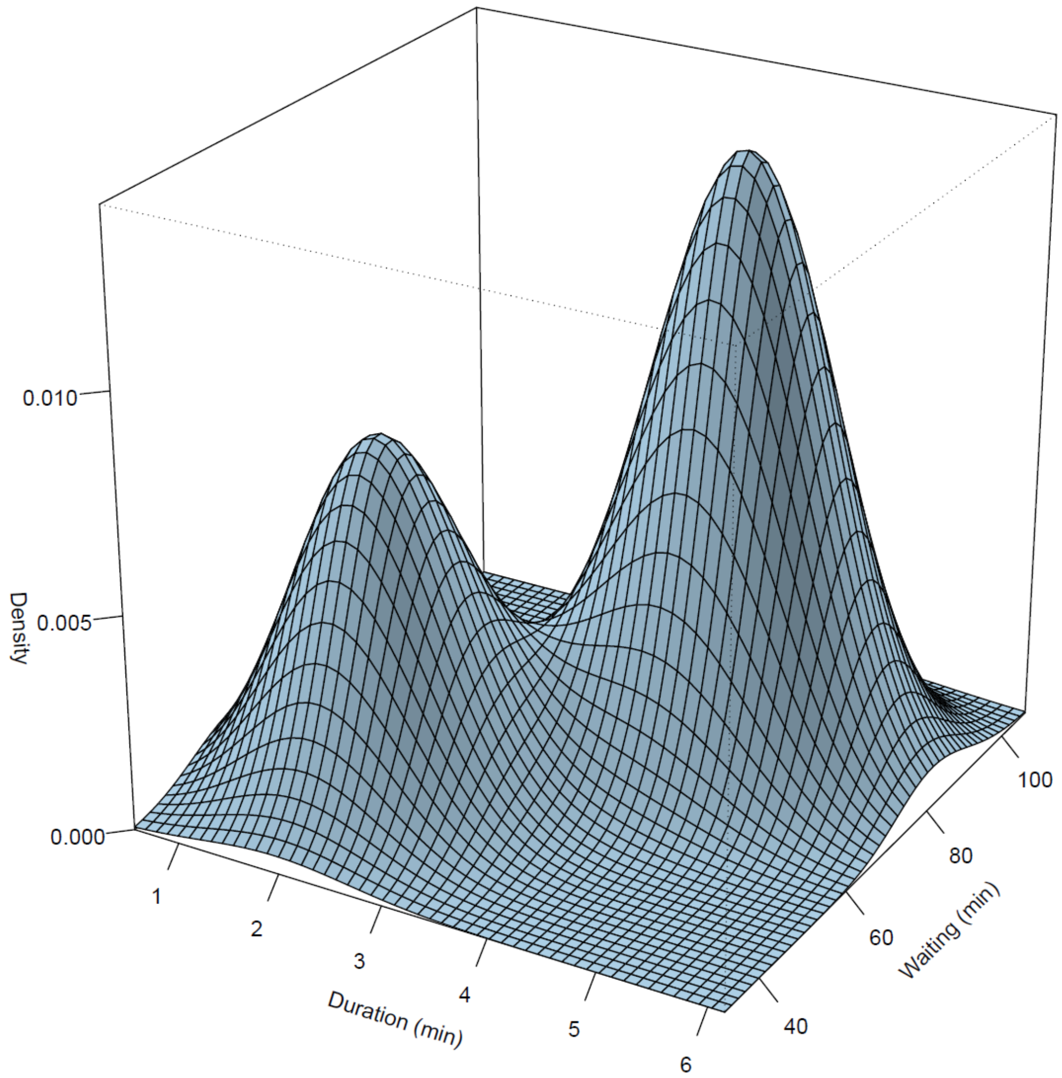
myPal <-colorRampPalette(c("white", "blue"))
filled.contour(fit$x1, fit$x2, fit$fhat,xlab="Duration (min)",ylab="Waiting (min)", color.palette=myPal)



Old faithful 2D density estimate: perspective plot

(Optional)

```
persp(fit$x1, fit$x2, fit$fhat,
shade=0.15, col="#BFE6FF",
theta=30, phi=25, d=5,
ticktype="detailed",
xlab="Duration
(min)",ylab="Waiting
(min)",zlab="Density")
require(rgl)
persp3d(fit$x1, fit$x2, fit$fhat,
col="#BFE6FF", xlab="Duration
(min)",ylab="Waiting
(min)",zlab="Density")
```



Limitations (Optional)

The density function is exclusively for one-dimensional kernel density estimation, but 2D density estimates like the ones just presented are available via the KernSmooth package.

The package is limited, however, in that it does not provide automatic methods for choosing bandwidths and it only extends to the 2D case.

Although one can easily write down an expression for the kernel density estimate in higher dimensions, the statistical properties of the estimator worsen rapidly as p grows.