

Lab 1: SOLUTIONS

36-350, Statistical Computing

Friday, 2 September 2011

These solutions are deliberately terse.

1.

```
x1 <- rexp(100) # or rexp(n=100,rate=1)
mean(x1) # Typical value: 1.22361
sd(x1)   # Typical value: 1.233442
```
2.

```
x0.1 <- rexp(100,0.1) # or rexp(n=100,rate=0.1)
x0.5 <- rexp(100,0.5)
x5 <- rexp(100,5)
x10 <- rexp(100,10)
# Next lines are with an eye to the next problem
x.means <- c(mean(x0.1),mean(x0.5),mean(x1),mean(x5),mean(x10))
x.sds <- c(sd(x0.1),sd(x0.5),sd(x1),sd(x5),sd(x10))
x.means # Typical value: 11.66383  1.90842  1.22361  0.17206  0.09410
x.sds # Typical: 12.21557  1.66369  1.23344  0.22016  0.08533
```
3.

```
x.rates <- c(0.1,0.5,1,5,10)
plot(x=x.rates,y=x.means) # Means vs. rates
plot(x=x.rates,y=x.sds)  # Standard deviations vs. rates
plot(x=x.means,y=x.sds)  # Means vs. standard deviations
# instructions ambiguous about which variable is to be the horizontal axis
```

Explanation: As the third plot shows, the mean and the standard deviation are very nearly equal for all rates. The mean seems to be close to one over the rate. Probability theory says that the expectation value should be exactly one over the rate, and should exactly equal the population standard deviation.

4.

```
y <- rexp(1e6) # or rexp(1000000) or rexp(n=1e6,rate=1), etc.
mean(y) # Typical value: 0.9992264
sd(y) # Typical value: 0.9967186
```

5.

```
hist(y)
```

See Figure 1

This does *not* match $1 - e^{-x}$, which is an increasing function that tends towards 1 as x grows, not towards 0. However, $1 - e^{-x}$ is the cumulative

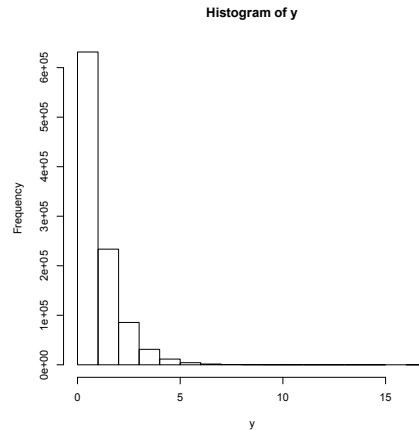


Figure 1: Histogram of y , from question 5.

distribution function, and the histogram should approximate the probability density function instead. We find the pdf $f(x)$ by differentiating the CDF $F(x)$ with respect to x , which gives e^{-x} , and this does have the right shape.

6. `y.mat <- matrix(y,nrow=1000) # or ncol=1000`
7. Because `hist` takes a vector argument, and a matrix has a vector lying underneath, it goes to the underlying vector. Thus `hist(y)` and `hist(y.mat)` give the same result.
8. `mean(y[,371])`
9. `y.col.means <- colMeans(y.mat) # or apply(y.mat,2,mean)`
`hist(y.col.means) # or even hist(colMeans(y.mat))`

The shape of this histogram (Figure 2) is a bell-shaped curve. This is because, by the central limit theorem, the mean of many independent, identically distributed variables approaches a Gaussian distribution, and the entries in each column are independent and identically distributed.

10. `mean(y[y>1]) # Typical value: 1.995996`
11. (10 points) Explain what these two commands do:

```
sum(y > 1)
mean(y[y.mat>1])
```

`y > 1` creates a Boolean, TRUE/FALSE vector, where each entry indicates whether the corresponding entry in `y` is strictly greater than 1. For

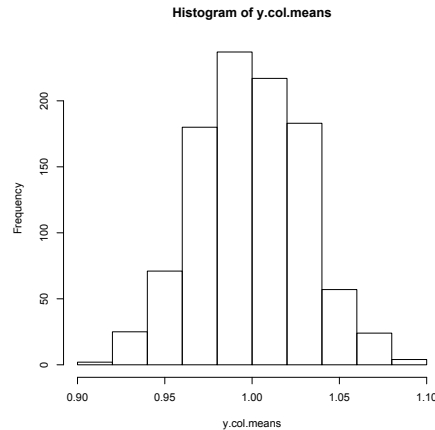


Figure 2: Histogram of the column means of `y.mat`.

purposes of arithmetic, `TRUE` is treated as 1 and `FALSE` as 0, so `sum(y > 1)` counts how many entries in `y` are strictly greater than 1. Typical value: 368398.

`y.mat > 1` creates a Boolean array which indicates which entries in `y.mat` are `> 1`. Then `y[y.mat>1]` tries to select the subset of elements of `y` where the Boolean entries are `TRUE`. Since `y` is a vector and not an array, the Boolean array is flattened out into a vector itself. `mean(y[y.mat > 1])` then takes the mean of the selected entries. This should be exactly the same as `mean(y[y>1])`, and it is.

12. `mean(y^2)` # Typical value: 1.9919
`(mean(y))^2 + (sd(y))^2` # Typical value: 1.991901
`(mean(y))^2 + (sd(y))^2*((1e6-1)/1e6)` # Typical value: 1.9919

Recall that for any random variable Y , with expectation $\mathbf{E}[Y]$ and variance $\text{Var}[Y]$

$$\text{Var}[Y] = \mathbf{E} \left[(Y - \mathbf{E}[Y])^2 \right] = \mathbf{E}[Y^2] - (\mathbf{E}[Y])^2$$

Re-arranging,

$$\mathbf{E}[Y^2] = (\mathbf{E}[Y])^2 + \text{Var}[Y]$$

and the variance is the square of the standard deviation. This also applies to the sample means and the sample variance. However, ordinarily when we ask R for the standard deviation of some data, we are interested in estimating the population standard deviation, so it divides by $n - 1$ rather than n . (See your 201/202 textbook.) This gives a value just a little too large here, and we need to scale it back down.