Statistical Methods for Data Science
CS 6313.001: Mini Project #1

Due on Thursday February 2, 2017 at 4pm

Instructor: Pankaj Choudhary

Hanlin He / Lizhong Zhang (hxh160630 / lxz160730)

Contents

Contribution

Generally speaking, two of us wrote this report together. To be specific, the work is split as follows:

- Hanlin He: P1(b)(c)(d), P2(a)(b)(f)
- Lizhong Zhang: P1(a)(e), P2(c)(d)(e)

Section 1 Answers

Problem 1.1 Random Variable X

(a) Compute E(X), Var(X) and P(X > 0.5)

Based on the conditions in the problem, we have:

$$f(x) = \begin{cases} 4x^3 & \text{if } 0 \le x < 1, \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We can calculate the following using formulas.

$$E(X) = \int_0^1 x f(x)\,\mathrm{d}x = \frac{4}{5} \times x^5 \Big|_0^1 = \frac{4}{5} = 0.8 \tag{2}$$

$$E(X^2) = \int_0^1 x^2 f(x)\,\mathrm{d}x = \frac{4}{6} \times x^6 \Big|_0^1 = \frac{2}{3} \tag{3}$$

$$Var(X) = E(X^2) - (E(X))^2 = \frac{2}{3} - \left(\frac{4}{5}\right)^2 \approx 0.02666667 \tag{4}$$

$$P(X > 0.5) = \int_{0.5}^1 f(x)\,\mathrm{d}x = \int_{0.5}^1 4x^3\,\mathrm{d}x = x^4\big|_{0.5}^1 = 0.9375 \tag{5}$$

(b) Simulate a Draw

From inverse transform method, we have:

$$U = F(X) = \int 4x^3\,\mathrm{d}x = x^4$$

Thus, $X = F^{-1}(U) = U^{\frac{1}{4}}$.

Therefore, to simulate a draw from the distribution of X, we can call **`runif()`** function in R as follow:

```
runif(1)^(1/4)

## [1] 0.7905004
```

(c) Approximate E(X), Var(X) and P(X > 0.5)

Using Monte Carlo simulation with 1,000 draws 5 times comes the results in table 1.

	Mean	Variance	P(X>0.5)
[1,]	0.7968975	0.02656562	0.935
[2,]	0.8066367	0.02570413	0.946
[3,]	0.7993725	0.02848914	0.933
[4,]	0.7960599	0.02745759	0.936
[5,]	0.7992016	0.02744726	0.929

Table 1: 5 Times 1000 Draws

(d) Repeat (c) with 10,000 draws.

Using Monte Carlo simulation with 10,000 draws 5 times comes the results in table 2.

	Mean	Variance	P(X>0.5)
[1,]	0.8019585	0.02683943	0.9398
[2,]	0.8017539	0.02565271	0.9418
[3,]	0.7997643	0.02633748	0.9370
[4,]	0.7984901	0.02708400	0.9329
[5,]	0.7980328	0.02707841	0.9356

Table 2: 5 Times 10000 Draws

(e) Compare Result in (a), (c), (d)

Based on the results computed in (a), we know that accurate values of $E(X)$, $Var(X)$ and $P(X > 0.5)$ theoretically.

Comparing the results of (c) and (d) to the theoretical values, we can find that, as the repetition of simulation increases, the simulated results approaches the theoretical results.

When we calculated the theoretical values of a distribution based on formulas, we were using implicit assumptions that the number of sample is enough large. Thus, in simulation, the more repeated times, the more accurate results we could get.

Problem 1.2 IQ test

(a) Compute the 95-th Percentile

Based on transformations from any Normal random variable to Standard Normal variable, $X = \mu + \sigma Z$. To compute the 95-th percentile of X is to compute the 95-th percentile of Z, in other word $x = \mu + \sigma(\Phi^{-1}(0.95))$.

According to Table A4 in Appendix of the textbook, $\Phi(1.64) = 0.9495 \approx 0.95$. Therefore, $\Phi^{-1}(0.95) = 1.64$

Thus, we have: $x = \mu + \sigma(\Phi^{-1}(0.95)) = \mu + \sigma(1.64) = 100 + 15 \times 1.64 = 124.6$.

(b) What does this mean?

If my IQ score equals the 95-th percentile, which is 124.6, it means, generally speaking, my IQ score is more than 95 percent of the total population, on which the IQ test has been taken, i.e. I'm smarter than 95% people among the population.

(c) Simulate a Draw

To simulate a draw from the distribution of IQ scores, we can call **`rnorm()`** function in R:

```
rnorm(1, mean = 100, sd = 15)

## [1] 96.57991
```

(d) Approximate the 95-th Percentile

Using Monte Carlo simulation with 1,000 draws 5 times comes the results in table 3.

95%	95%	95%	95%	95%
120.8349	125.8516	123.7723	125.7336	125.0442

Table 3: 5 Times 1000 Draws

(e) Repeat (d) with 10,000 draws

Using Monte Carlo simulation with 10,000 draws 5 times comes the results in table 4.

95%	95%	95%	95%	95%
124.4912	123.8312	125.0600	124.6121	124.8582

Table 4: 5 Times 10000 Draws

(f) Compare Result in (a), (c), (d)

In (a), we used the Table to calculate the 95-th percentile of a standard normal distribution, which is an approximation of a theoretical value when the population is infinite.

During the simulation, let X denotes the result in (d), and Y denotes the result in (e),

$$mean(x) = 124.2473 \qquad var(x) = 4.322004$$
$$mean(y) = 124.5705 \qquad var(y) = 0.2192413$$

we can see that the mean of (d) and (e) are both very close to the theoretical value calculated in (a), but the variance of (e) is much smaller than the variance of (d). In other word, when the number of draws (size of sample) increases, the 95-th percentile of sample becomes closer to the theoretical value.

This observation can be explained by the law of large number.

Section 2 R Code

```
################################################################################
# R code for problem 1
################################################################################

# 1 (b) Explain how you would simulate a draw from the distribution of X.
runif(1)^(1/4)

## [1] 0.9476318

# Define function getdraw(x,y) for (c) and (d).
# Arguments:
#       'x' is the number of draws to simulate.
#       'y' is the times to repeat.
# Result: Summary of E(X), Var(X) and P(X>5) of each simulation in a table.

getdraw <- function(x, y) {
    # Simulated x draws y times, result stored in variable 'draw'.
    draw <- replicate(y, runif(x)^(1/4))

    # Calculated the result of each 1000 draws,
    # i.e. apply calculation in each column.

    # Calculated the mean in each column.
    mean5 <- apply(draw, 2, mean)
```

```
    # Calculated the variance in each column.
    var5 <- apply(draw, 2, var)

    # Calculated the probability of X > 5 in each column.
    p5 <- apply(draw, 2, (function(x) sum(x > 0.5)/sum(x >= 0)))

    # Combine the result in a table.
    result <- cbind(mean5, var5, p5)

    # Rename column.
    colnames(result) <- c("Mean","Variance","P(X>0.5)")

    return(result)
}


# 1 (c) Approximate E(X), Var(X) and P(X > 0.5)
#       using Monte Carlo simulation with 1,000
#       draws 5 times. Summarize the results in a table.

getdraw(1000, 5)
```

```
##           Mean   Variance P(X>0.5)
## [1,] 0.8049452 0.02534549    0.948
## [2,] 0.8000916 0.02706675    0.944
## [3,] 0.8105327 0.02494554    0.947
## [4,] 0.7988951 0.02895235    0.930
## [5,] 0.8002070 0.02614280    0.939
```

```
# 1 (d) Repeat (c) with 10,000 draws.

getdraw(10000, 5)
```

```
##           Mean   Variance P(X>0.5)
## [1,] 0.7993287 0.02676855   0.9385
## [2,] 0.7975362 0.02755187   0.9352
## [3,] 0.8013925 0.02583196   0.9396
## [4,] 0.7990607 0.02680608   0.9379
## [5,] 0.7965577 0.02675823   0.9391
```

```
################################################################################
# R code for problem 2
################################################################################

# 2 (c) Explain how you would simulate a draw from the distribution
#       of the IQ scores.
rnorm(1, mean = 100, sd = 15)

## [1] 111.3283

# 2 (d) Approximate the 95-th percentile of the distribution
#       using Monte Carlo simulation with 1,000 draws 5 times.
replicate(5, quantile(rnorm(1000, mean = 100, sd = 15), prob = 0.95))

##      95%      95%      95%      95%      95%
## 125.4001 123.5218 123.6171 124.0018 122.0643

# 2 (e) Repeat (d) with 10,000 draws.
replicate(5, quantile(rnorm(10000, mean = 100, sd = 15), prob = 0.95))

##      95%      95%      95%      95%      95%
## 124.8571 123.6998 124.5491 125.4282 124.4076
```