The normal distribution

The normal distribution is the most important distribution in statistics, as a consequence of the central limit theorem and other properties. Parameterised by the mean, μ, and the variance, σ^2, the probability density function is

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

The distribution of F doesn't have a closed form, meaning it is difficult to sample from it with the same technique used for the gamma and exponential distributions. Instead, we use a different idea.

If (Φ, V) have joint density $f(\phi, v)$, and we define $X(\Phi, V)$ and $Y(\Phi, V)$ such that (X, Y) is a 1-1 function of (Φ, V), then (X, Y) has joint density

$$g(x, y) = f(\phi(x, y), v(x, y)) \left| \frac{\partial(\phi, v)}{\partial(x, y)} \right|.$$

With this in mind, with sensible choices of Φ, V, X, Y, we can obtain the normal distribution as a function of simpler distributions.

Using the result of Problem 9 below, we can take independent uniform variables U_1, U_2, define $\Phi = 2\pi U_1$ and $V = -2\ln(1 - U_2)$, and then obtain independent normally distributed random variables

$$X = \mu_1 + \sigma\sqrt{V}\cos\Phi,$$

$$Y = \mu_2 + \sigma\sqrt{V}\sin\Phi.$$

Programs

The normal pdf is implemented as `normal_pdf`.

We can sample from the normal distribution with the following code:

```
mu <- 0
sigma <- 1
n <- 100

normal_samples <- normal_distribution_sampler(n, mu, sigma^2)
head(normal_samples)
```

```
## [1] -0.77682970  0.18613445 -1.78910517  1.94268559 -1.06754990 -0.05067528
```

R also has a built-in normal distribution with pdf `dnorm`, which can be sampled with `rnorm`. We can test that this matches our implementation using a one-sample Kolmogorov-Smirnov test:

```
ks_results <- ks.test(normal_samples, "dnorm", mean = mu, sd = sigma)
ks_results[["p.value"]]
```

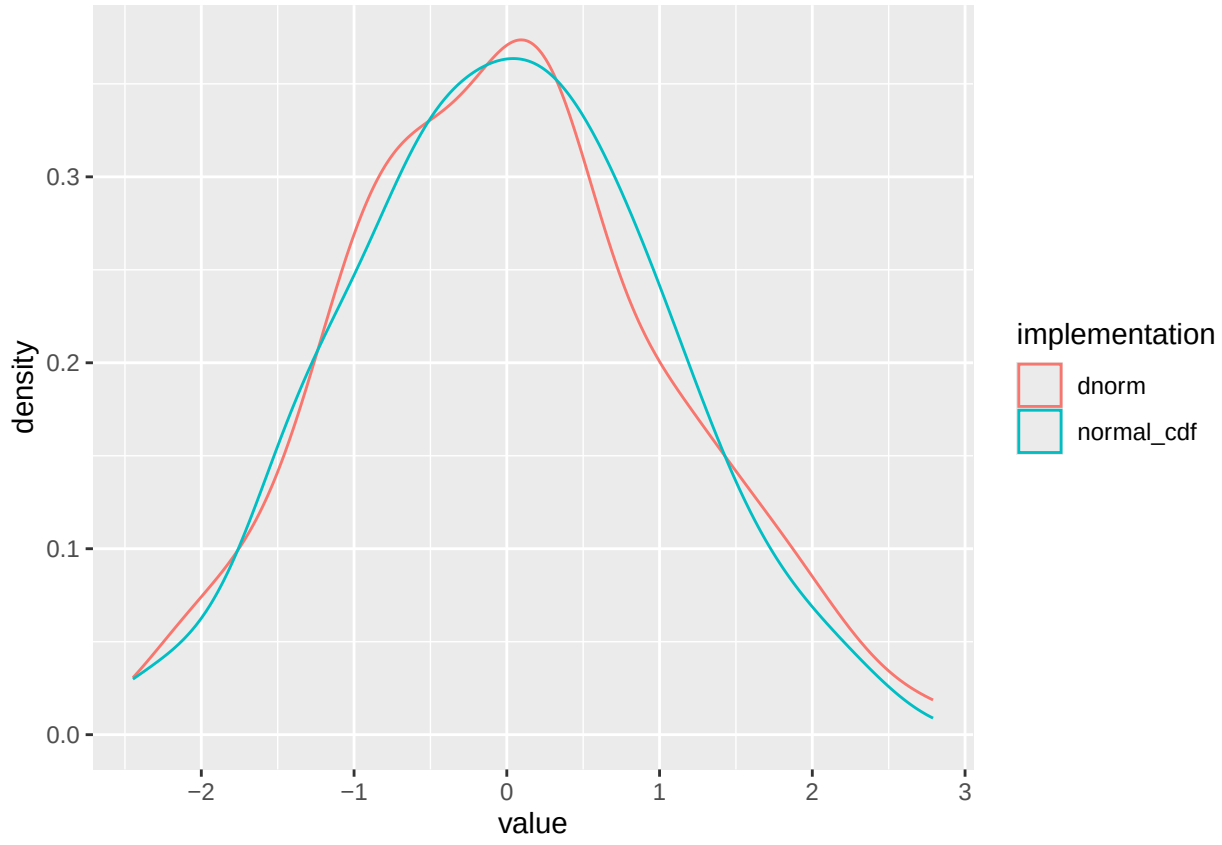
```
## [1] 3.765226e-82
```

In this instance, since $p < 0.05$, we conclude that the distributions match.

We can also plot the distributions against each other:

```
dat <- data.frame(implementation =
                    factor(rep(c("normal_cdf", "dnorm"), each = n)),
                  value = c(normal_samples,
                            rnorm(n, mean = mu, sd = sigma)))

ggplot(dat, aes(x = value, colour = implementation)) + geom_density()
```



Here, we have some noise from our relatively low number of samples, but the distributions clearly have the same shape.

Problems

Problem 9

Show that if $h(\phi, v) = \frac{1}{4\pi} e^{-v/2}$, $0 \leq \phi \leq 2\pi$, $v > 0$, and if we define

$$X = \mu_1 + \sigma\sqrt{V}\cos\Phi,$$

$$Y = \mu_2 + \sigma\sqrt{V}\sin\Phi,$$

then X, Y are independent $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$ random variables, i.e.,

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{\{(x-\mu_1)^2+(y-\mu_2)^2\}}{2\sigma^2}}, -\infty < x, y < \infty$$

Solution We can compute

$$V = (\frac{X-\mu_1}{\sigma})^2 + (\frac{Y-\mu_2}{\sigma})^2.$$

Using V, we can find

$$A = \sin\Phi = \frac{Y-\mu_2}{\sigma\sqrt{V}},$$
$$B = \cos\Phi = \frac{X-\mu_1}{\sigma\sqrt{V}},$$

and thus

$$\Phi = \begin{cases} \arctan(\frac{A}{B}) & 0 < A, B \\ \frac{\pi}{2} & 0 = B < A \\ \arctan(\frac{A}{B}) + \pi & B < 0 \\ \frac{3\pi}{2} & A < B = 0 \\ \arctan(\frac{A}{B}) + 2\pi & A < 0 < B \end{cases}$$

Now, we can compute

$$\begin{aligned} g(x,y) &= f(\phi(x,y), v(x,y)) \left|\frac{\partial(\phi, v)}{\partial(x,y)}\right| \\ &= \frac{1}{4\pi} e^{-\frac{\{(x-\mu_1)^2+(y-\mu_2)^2\}}{2\sigma^2}} \times \frac{2}{\sigma^2} \\ &= f(x|\mu_1, \sigma^2) f(y|\mu_2, \sigma^2), \end{aligned}$$

as required.

Problem 10

Explain how to construct an 80% confidence interval for μ.

Solution With $\hat{\mu} = \frac{\sum x_i}{n}$ as the sample mean, and $\hat{\sigma}^2 = \frac{\sum (x_i - \hat{\mu})^2}{n-1}$ as the sample variance, it's a well-known fact that $t = \frac{\hat{\mu}-\mu}{\hat{\sigma}/\sqrt{n}}$ follows the student t distribution with $n-1$ degrees of freedom. If t_α is the α^{th} quantile of this distribution, then a (symmetric) α% confidence interval for μ is given by rearranging:

$$[\hat{\mu} - t_{\alpha/2}\frac{\hat{\sigma}}{\sqrt{n}}, \hat{\mu} + t_{1-\alpha/2}\frac{\hat{\sigma}}{\sqrt{n}}].$$

We define the following function to compute a α% confidence interval from given normal samples:

```
confidence_interval <- function(samples, signficance) {
  n <- length(samples)
  mean <- sum(samples) / n
  std_dev <- sqrt(sum((samples - mean)^2) / (n - 1))
  # `qt` is R's built-in quantile function for the student t distribution.
  t_lower <- qt((1 - signficance) / 2, n - 1)
  t_upper <- qt((1 + signficance) / 2, n - 1)
```

```
  lower <- mean + t_lower * (std_dev / sqrt(n))
  upper <- mean + t_upper * (std_dev / sqrt(n))
  c(lower, upper)
}
```

Problem 11

For $\mu = 0$, generate a sample of size $n = 100$ from distribution $N(\mu, 1)$ and check whether the confidence interval does indeed contain μ. Repeat this procedure 25 times and display the results in a table with four columns, containing the sample mean, the lower and upper bound of the confidence interval, and an indicator of whether or not the interval contained the true mean. How many times did the interval not contain μ?

Solution We use the following program:

```
n <- 100
mu <- 0
sigma <- 1
dat <- data.frame(index = integer(),
                  sample_mean = character(),
                  lower_bound = character(),
                  upper_bound = character(),
                  in_interval = logical())
for (i in 1: 25) {
  sample <- normal_distribution_sampler(n, mean = mu, variance = sigma^2)
  sample_mean <- sum(sample) / n
  sample_ci <- confidence_interval(sample, 0.8)
  in_interval <- (sample_ci[1] < mu) & (mu < sample_ci[2])
  dat[nrow(dat) + 1, ] <- c(i,
                            round(sample_mean, 5),
                            round(sample_ci, 5),
                            in_interval)
}

knitr::kable(dat)
```

index	sample_mean	lower_bound	upper_bound	in_interval
1	0.04342	-0.10328	0.19012	1
2	0.12679	-0.00724	0.26082	1
3	0.13559	0.0043	0.26688	0
4	0.12158	-0.00574	0.24889	1
5	0.02886	-0.1143	0.17202	1
6	-0.07894	-0.20249	0.04462	1
7	-0.10972	-0.23406	0.01462	1
8	0.12017	-0.00127	0.24161	1
9	0.16381	0.0468	0.28081	0
10	0.12072	0.00118	0.24026	0
11	-0.09183	-0.22398	0.04032	1
12	0.02105	-0.11849	0.16059	1
13	0.08428	-0.03784	0.20641	1
14	0.07757	-0.04621	0.20135	1
15	0.03039	-0.0719	0.13267	1

index	sample_mean	lower_bound	upper_bound	in_interval
16	-0.08948	-0.20418	0.02522	1
17	0.17692	0.05088	0.30295	0
18	-0.06998	-0.20258	0.06263	1
19	0.11971	-0.02406	0.26348	1
20	-0.09455	-0.23368	0.04458	1
21	0.00254	-0.12548	0.13055	1
22	0.09488	-0.04185	0.23161	1
23	0.00227	-0.13447	0.13902	1
24	-0.04822	-0.18323	0.08678	1
25	-0.20863	-0.33045	-0.08682	0

Of these, 5 samples didn't contain the true mean, or 20%.

Problem 12

If questions 10 and 11 were to be repeated with $n = 50$ and $\mu = 4$, how many times would you expect the confidence interval not to contain μ?

Solution By definition of the 80% confidence interval, we expect that 20% of times we compute the confidence interval, the true mean will lie outside it, independent of sample size and parameters. In this case, we can test this, with a large number of repetitions to take advantage of the law of large numbers:

```
n <- 50
mu <- 4
sigma <- 1
no_of_successes <- 0
dat <- data.frame(index = integer(),
                  sample_mean = character(),
                  lower_bound = character(),
                  upper_bound = character(),
                  in_interval = logical())
for (i in 1: 100000) {
  sample <- normal_distribution_sampler(n, mean = mu, variance = sigma^2)
  sample_mean <- sum(sample) / n
  sample_ci <- confidence_interval(sample, 0.8)
  no_of_successes <- no_of_successes + ((sample_ci[1] < mu)
                                        & (mu < sample_ci[2]))
}
```

In this case, 19.987% of confidence intervals didn't contain the true mean, which is about what we expected.