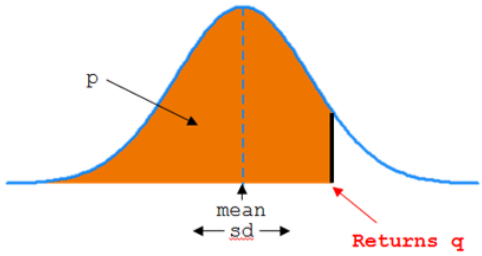
Chapter 3 Workshop

Table of contents

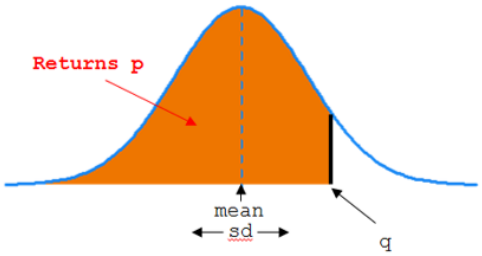
Distribution functions in R

For any distribution, there are four key R functions. In Figure 1, they are demonstrated for the normal distribution (`norm`).

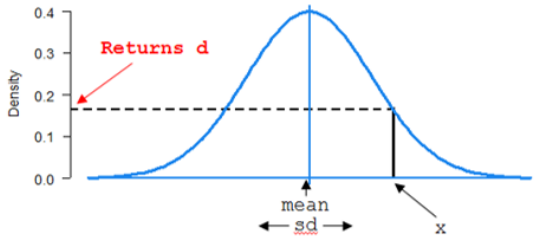
`qnorm(p,mean=0, sd=1)`
calculates quantiles (*x* or *z* values) from probabilities.



`pnorm(q,mean=0, sd=1)`
calculates probabilities from quantiles (*x* or *z* values).



`dnorm(x,mean=0, sd=1)`
calculates the density (*d*, the height of the curve) from *x* or *z* values.



`rnorm(n,mean=0, sd=1)` gives *n* random draws from a normal distribution.

```
> rnorm(5,mean=0,sd=1)
[1] 0.03974299 0.81851262 0.20344821 0.95838098 0.62207198
```

Figure 1: The four R function for the normal probability density function.

Dataset `Prestige`

We will be using a well-known dataset called `Prestige` from the `car` R package. This dataset deals with prestige ratings of Canadian Occupations. The `Prestige` dataset has 102 rows and 6 columns. The observations are occupations.

This data frame contains the following columns:

- **`education`** - Average education of occupational incumbents, years, in 1971.
- **`income`** - Average income of incumbents, dollars, in 1971.
- **`women`** - Percentage of incumbents who are women.
- **`prestige`** - Pineo-Porter prestige score for occupation, from a social survey conducted in the mid-1960s.
- **`census`** - Canadian Census occupational code.
- **`type`** - Type of occupation. A factor with levels: bc, Blue Collar; prof, Professional, Managerial, and Technical; wc, White Collar. (includes four missing values).

Exercise 3.1

For a standard normal variable z , obtain the area between -1.8 and 2.1.

```
pnorm(2.1, mean=0, sd=1) - pnorm(-1.8, mean=0, sd=1)
```

Note that the `mean=0`, `sd=1` are the defaults for `pnorm` function, so don't need to be specified.

```
pnorm(2.1) - pnorm(-1.8)
```

Exercise 3.2

Plot the `prestige` scores data as a histogram and show the theoretical normal curve fitted to the data.

```
library(tidyverse)
library(car)

Prestige |>
  ggplot() +
  aes(prestige) +
  geom_histogram(aes(y=after_stat(density)), bins=10) +
  stat_function(
    fun = dnorm,
    args = list(mean = mean(Prestige$prestige),
                sd = sd(Prestige$prestige) ),
    geom = "line")
```

Let's try a square-root transformation

```
library(tidyverse)
library(car)

Prestige |>
  ggplot() +
  aes(sqrt(prestige)) +
  geom_histogram(aes(y=after_stat(density)), bins=10) +
  stat_function(
    fun = dnorm,
    args = list(mean = mean(sqrt(Prestige$prestige)),
                sd = sd(sqrt(Prestige$prestige)) ),
    geom = "line") +
  ggtitle("Square-root prestige")
```

Exercise 3.3

Let's look at the `prestige` scores variable to see how well it conforms with a normal distribution.

First, make a normal quantile plot.

```
Prestige |>
  ggplot() +
  aes(sample=prestige) +
  stat_qq() +
  stat_qq_line()
```

The x-axis are theoretical quantiles of a normal distribution; the y-axis are the quantiles of the actual data.

If the data conformed perfectly to a normal distribution, the points would lie perfectly along the line.

The above plot shows that these data conform pretty well to the normal. There is very often some departure in the 'tails' at either end, like there is here. Here's a plot of data that were *actually simulated from a normal distribution* for comparison:

```
set.seed(111)

data.frame(x = rnorm(
  n = nrow(Prestige),
  mean = mean(Prestige$prestige),
  sd = sd(Prestige$prestige)
  )) |>
  ggplot() +
  aes(sample = x) +
  stat_qq() +
  stat_qq_line()
```

Now, we'll do some tests for whether `prestige` scores show a "significant" departure from the normal distribution.

The null hypothesis is that the data came from a normal distribution. A small p-value (say, < 0.05) would lead us to reject the null hypothesis and conclude that the data are unlikely to have come from a normal distribution. A large p-value (> 0.05) means we have no evidence of non-normality.

First, the Shapiro-Wilk test.

```
shapiro.test(Prestige$prestige)
```

Here, the null hypothesis is rejected, so the data are unlikely to have come from a normal.

The Kolmogorov-Smirnov test can also be used. It differs from the Shapiro-Wilk in that you specify the mean and SD of the distribution (here using the sample mean and SD).

```
ks.test(Prestige$prestige,
        "pnorm",
        mean(Prestige$prestige),
        sd(Prestige$prestige) )
```

Here we have a discrepancy. S-W rejected the null hypothesis, and K-S did not. It is well-known that S-W is generally more powerful (i.e., more likely to reject a false null hypothesis).

At any rate, I don't believe in "true" distributions. Would I feel comfortable using a normal here? Possibly. It all depends on the context. Remember, there are no true models, only useful ones.

We can try the square-root transformed `prestige` with .

```
shapiro.test(sqrt(Prestige$prestige))
```

No significant departure from normality for the sqrt-transformed data.

Exercise 3.4

Examine the fit of non-normal distributions for `prestige` scores data.

Log-normal

```
library(fitdistrplus)

m1 <- fitdist(Prestige$prestige, "lnorm")

plot(m1)
```

Gamma

```
library(fitdistrplus)

m2 <- fitdist(Prestige$prestige, "gamma")

plot(m2)
```

Weibull

```
library(fitdistrplus)

m3 <- fitdist(Prestige$prestige, "weibull")

plot(m3)
```

```
descdist(Prestige$prestige)
```

More graphing examples are here (R code file).