Stat 230: Lab 5

Dr. Irene Vrbik

Last updated: 04 November, 2021

Contents

1 Introduction

In class we have introduced one of the most important distributions in statistics: *the Normal (AKA Gaussian) distribution*. If a rv X follows a normal distribution with parameters μ and σ, then we write $X \sim N(\mu, \sigma)$, and the pdf of X is

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \qquad -\infty < x < \infty, \tag{1}$$

where μ is the **mean** of X and σ is the **standard deviation** of X. Alternatively, the normal distribution may be parameterized by it's variance; that is some sources may use $N(\mu, \sigma^2)$ instead of $N(\mu, \sigma)$. While I usually like to parameterize the normal distribution in lecture using variance, I will try and stick to standard deviation in this lab in order to remain consistent with the normal functions (`dnorm`, `pnorm`, `qnorm` and `rnorm`) in R.

2 R functions for the Normal Distribution

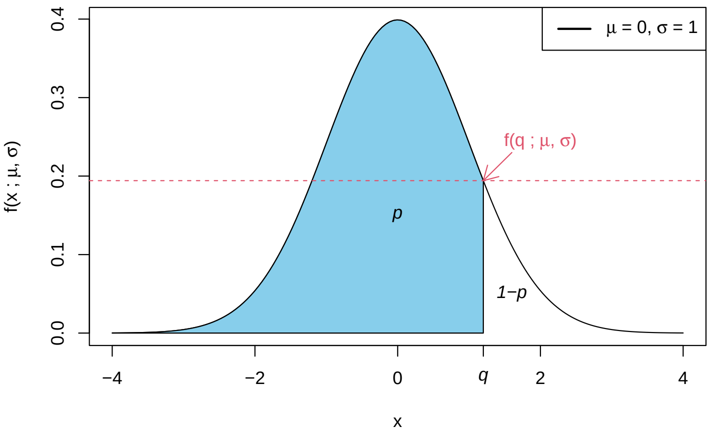
2.1 Sparknotes on `d/p/q norm` functions

> Please notice that the third argument in these function is standard deviation, not variance; that is `sd` = σ, `sd` $\neq \sigma^2$.

Table 1: Summary of functions. Note that the value being returned by the function are highlighted in red in the **Probability** column.

Distribution	**Command**	**Probability**
$X \sim \text{Normal}(\mu, \sigma^2)$	`dnorm(`x`, mean=`μ`, sd=`σ`)`	$f(x;\mu,\sigma^2)^*$
	`pnorm(`q`, mean=`μ`, sd=`σ`, prob=`p`)`	$\text{P}(X \leq q) = p$
	`pnorm(`q`, mean=`μ`, sd=`σ`, prob=`p`, lower.tail=FALSE)`	$\text{P}(X > q) = p$
	`qnorm(`p`, mean=`μ`), sd=`σ`)`	$\text{P}(X \leq q) = p$

*Note that the `dnorm` function does not provide the $P(X = x)$; recall the $P(X = x) = 0$ for all x.



Consider the following equation:

$$P(X \leq q) = p \tag{2}$$

If you want to find p for some value of q use:

`pnorm(q, mean, sd)` $= p$ (value will be returned by R)

If you want to find $1 - p$ for some value of q use:

`pnorm(q, mean, sd, lower.tail=FALSE)` $= 1 - p$ (value will be returned by R)

If you want to find q for some probability of p use:

`qnorm(p, mean, sd)` $= q$ (value will be returned by R)

If you want to evaluate (1) for some q, that is if you want to calculate $f(q; \mu, \sigma)$ use:

`dnorm(q, mean, sd)` $= \frac{1}{\texttt{sd}\sqrt{2\pi}} e^{-(q-\texttt{mean})^2/2\texttt{sd}^2}$ (value returned by R)

2.2 Standard Normal Distribution

A special case of this distribution is when $\mu = 0$ and $\sigma^2 = 1$, we call this the *Standard Normal*. If you do not specify a `mean` and `sd` into the `*norm` functions, it will assume the standard normal (that is, it will assume `mean=0`, and `sd=1`).

2.3 `dnorm` PDF of the Normal Distribution

To evaluate the pdf of the normal distribution, i.e. (1), for particular values of x, μ and σ, we use the `dnorm` function:

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
```

where:

`x` value/vector of x value(s)

`mean` means value μ

`sd` standard value σ

Note that the `dnorm` function evaluates the pdf of the normal distribution (given in equation (1)) at a particular value of x. As discussed in lecture, $f(x; \mu, \sigma^2)$ does *not* provide $P(X = x)$ [1]. It is only once we integrate $f(x; \mu, \sigma^2)$ that we can find probabilities of the form $P(X > x)$, $P(a < X < b)$, and $P(X < x)$ (which is the same as $P(X \leq x)$).

```
dnorm(x=4.21, mean=7, sd=3)

## [1] 0.08629352

# same as
(1/(3*sqrt(2*pi)))*exp(-(4.21-7)^2/(2*3^2))

## [1] 0.08629352
```

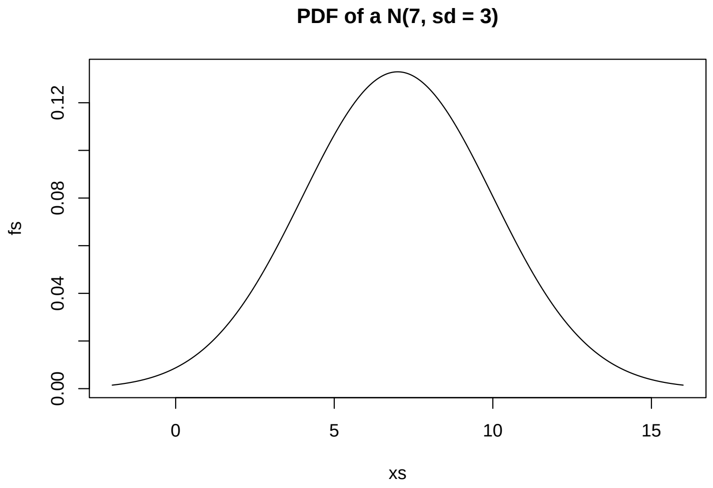
Notice that vectors can be fed into this function:

```
dnorm(x=c(4.21, 7.1, 3.2, -2), mean=7, sd=3)

## [1] 0.086293516 0.132906902 0.059619472 0.001477283
```

Recall that these are not yet probabilities! In fact the $P(X = x)$ for any x where X is a continuous RV is-0. It is only once we integrate the pdf do we get probabilities! For fun let's plot a sequence of theses $f(x; \mu, \sigma)$ values on a plot (although you should already know what it will look like).

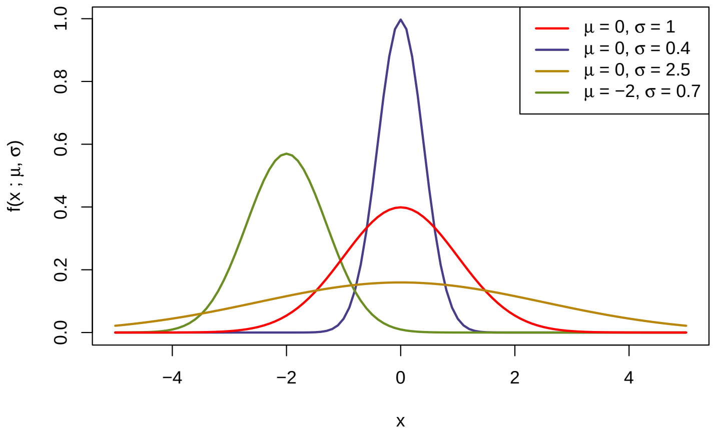
```
# create a sequence of numbers from -2 to 16,
# incrementing by 0.01
xs <- seq(from=-2, to=16, by=0.01)
xs[1:6]

## [1] -2.00 -1.99 -1.98 -1.97 -1.96 -1.95

# evaluate the pdf of a N(7,3) for these values of x
fs <- dnorm(xs, mean=7, sd=3)
# plot them using type = "l"
plot(xs, fs, type="l", main="PDF of a N(7, sd = 3)")
```

[1] $P(X = x)$ is equal to 0 for all x for continuous random variables



This produces a bell-shaped symmetric curve centered at 7 (as one would expect). To produce a more smooth version of this curve we could alternatively use the `curve` function. This will draw a curve corresponding to a function (provided in the first arguments) over the interval `[from` (second argument), `to` (third argument)`]`. Notice how this method does not require me to created a sequence vector of x values. Let's see how the normal curve would look for a range of values of μ and σ (to have them all appear on a single plot I am using the argument `add=TRUE`):

```
curve(dnorm(x, mean=0, sd=0.4), from=-5, to=5, col="darkslateblue", lwd=2,
      ylab=  expression(paste(f*"("* x *" ; " * mu *", "*sigma*")")))
curve(dnorm(x, mean=-2, sd=0.7), from=-5, to=5, col="olivedrab", lwd=2, add=TRUE)
curve(dnorm(x, mean=0, sd=1), from=-5, to=5, col="red", lwd=2, add=TRUE)
curve(dnorm(x, mean=0, sd=2.5), from=-5, to=5, col="darkgoldenrod", lwd=2, add=TRUE)
legend("topright", col=c("red", "darkslateblue", "darkgoldenrod", "olivedrab"),lwd= 2,
       legend = c(expression(mu*" = 0, "*sigma*" = 1"),
                  expression(mu*" = 0, "*sigma*" = 0.4"),
                  expression(mu*" = 0, "*sigma*" = 2.5"),
                  expression(mu*" = -2, "*sigma*" = 0.7")))
```



2.4 `pnorm` Finding probabilities for Normal Distribution

The `pnorm` function returns the CDF of the normal distribution. That is

```
pnorm(q, mean, sd)
```

returns $P(X \le \texttt{q})$ where $X \sim Norm(\mu = \texttt{mean}, \sigma = \texttt{sd})$. Alternatively, if we want to find $P(X > \texttt{q})$, or equivalently $P(X \ge \texttt{q})$, we use:

```
pnorm(q, mean, sd, lower.tail = FALSE)
```

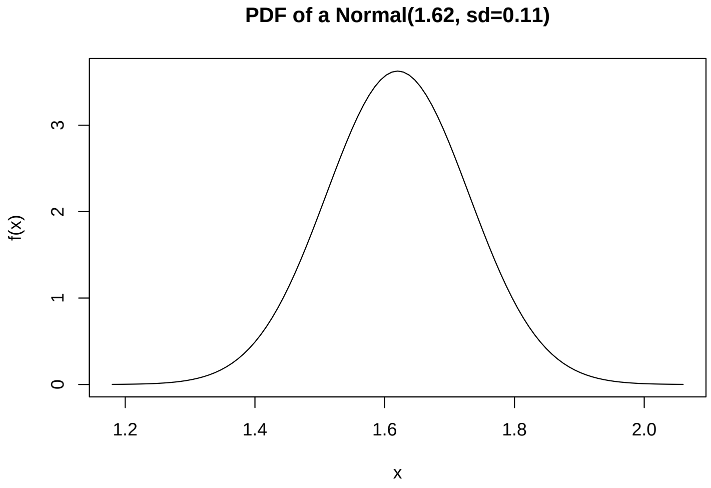
2.4.1 Height of Irish Women

Let's return to the height of Irish women example that we looked at in class.

Example 2.1. Consider the example of the height of women in Ireland, which we assume is normally distributed with mean 1.62m and standard deviation 0.11m.

We can plot the curve (this time I'll use the `curve` function to save us from having to create a vector of x values)

```
# plots f(x) from x=1.18,..2.06 (4 sd on each side of the mean)
# where f(x) is the pdf of a N(1.62, sd=0.11) RV
curve(dnorm(x, mean=1.62, sd=0.11), from=1.18, to= 2.06, ylab="f(x)",
      main="PDF of a Normal(1.62, sd=0.11)")
```



Let's recalculate the probabilities we computed in using the z-table in class:

- Find the probability that a randomly selected woman in Ireland is shorter than 1.52m $P(X \le 1.52)$ where $X \sim N(\mu = 1.62, \sigma = 0.11)$

  ```
  pnorm(1.52, mean=1.62, sd=0.11)

  ## [1] 0.1816511
  ```

- Find the probability that a randomly selected woman in Ireland is taller than 1.72m $P(X > 1.72)$ where $X \sim N(\mu = 1.62, \sigma = 0.11)$

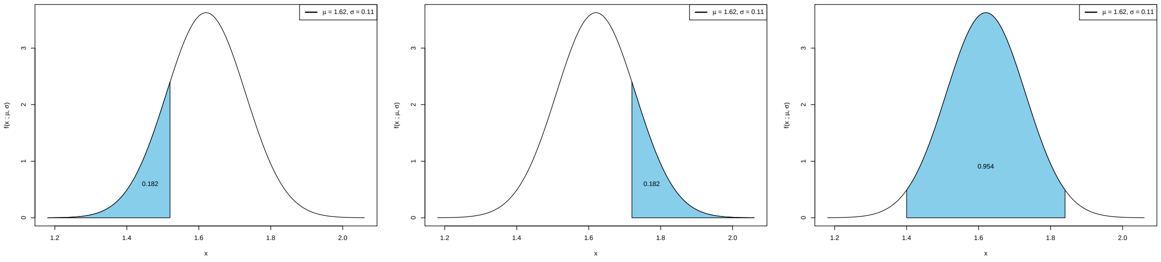
```
pnorm(1.72, mean=1.62, sd=0.11, lower.tail = FALSE)

## [1] 0.1816511
```

Recall $P(X > 1.72) = 1 - P(X \leq 1.72)$ so we could have alternatively calculated is using:

```
1 - pnorm(1.72, mean=1.62, sd=0.11)

## [1] 0.1816511
```

- Find the probability that women are between 1.40 and 1.84 meters tall. $P(1.40 \leq X \leq 1.84) = P(X \leq 1.84) - P(X \leq 1.40)$

```
pnorm(1.84, mean=1.62, sd=0.11) - pnorm(1.40, mean=1.62, sd=0.11)

## [1] 0.9544997
```



Notice that the above solutions bypasses the need to standardize these values into standard normal random variables. That is, in class we needed to transform $X \sim N(\mu, \sigma^2)$ into $Z \sim N(0, 1)$, so that we could use the Z-tables to look up probabilities. Recall the standardization formula:

$$Z = \frac{X - \mu}{\sigma}.$$

Another way to say this, is that if we want to ask questions of the form $P(X \leq x)$ where $X \sim N(\mu, \sigma)$ we can reframe them into questions of the form $P(Z \leq z)$ where $Z \sim N(0, 1)$ (the "standard" normal distribution). This is the only way we are able to find probabilities for the normal distribution "on paper". If however we are calculating these probabilities with R there is no need to do this!

For fun, let's verify that the answers are the same and visualize them side by side:

- Find the probability that a randomly selected woman in Ireland is shorter than 1.52m

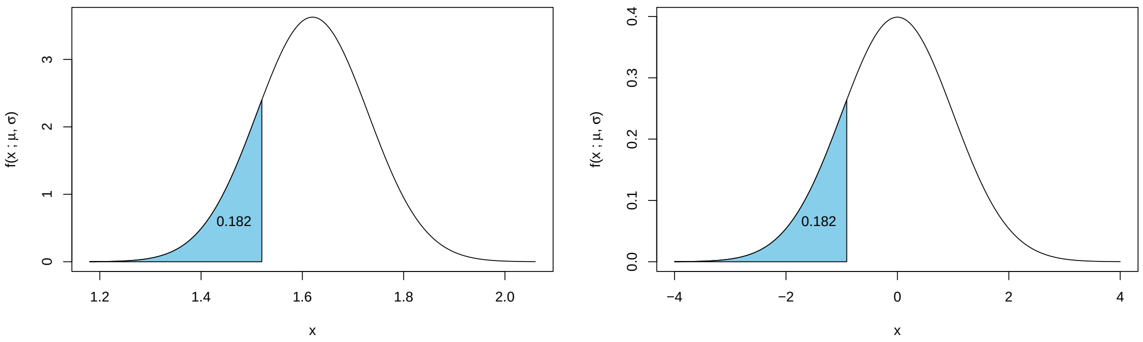
$$\begin{aligned} P(X \leq 1.52) &= P\left(Z \leq \frac{x - \mu}{\sigma}\right) && \text{where } X \sim N(\mu = 1.62, \sigma = 0.11) \\ &= P(Z \leq \frac{1.52 - 1.62}{0.11}) \\ &= P(Z \leq -0.9090909) && \text{where } Z \sim N(\mu = 0, \sigma = 1) \end{aligned}$$

```
z = (1.52-1.62)/0.11 # z-score
pnorm(z, mean=0, sd=1)

## [1] 0.1816511
```

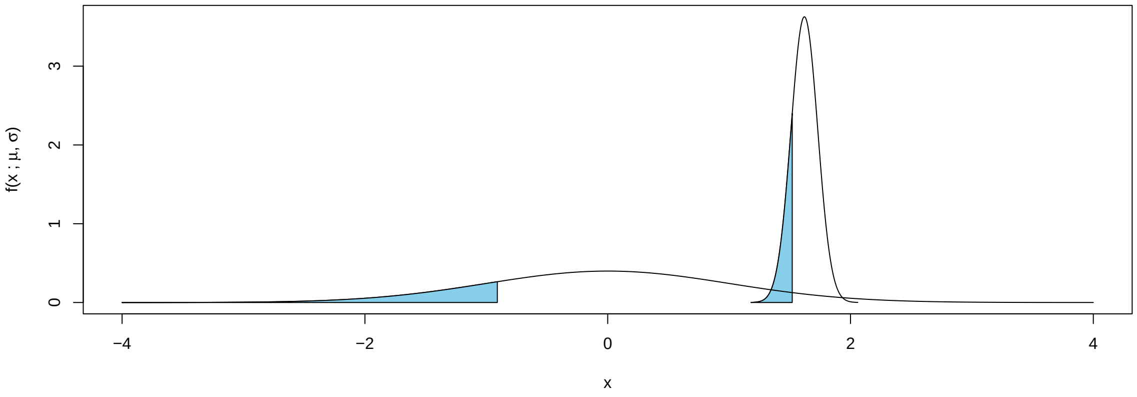
Recall since `mean=0` and `sd=1` is the default values, we can just type:

```
pnorm(z)

## [1] 0.1816511
```

That is $P(X \leq 1.52) = P(Z \leq -0.9090909)$, where $X \sim N(\mu = 1.62, \sigma = 0.11)$ and Z is the standard normal. Let's see a visualization of this (code not shown).



To see them on the same scale:



2.5 `rnorm` Simulating from a Normal Distribution

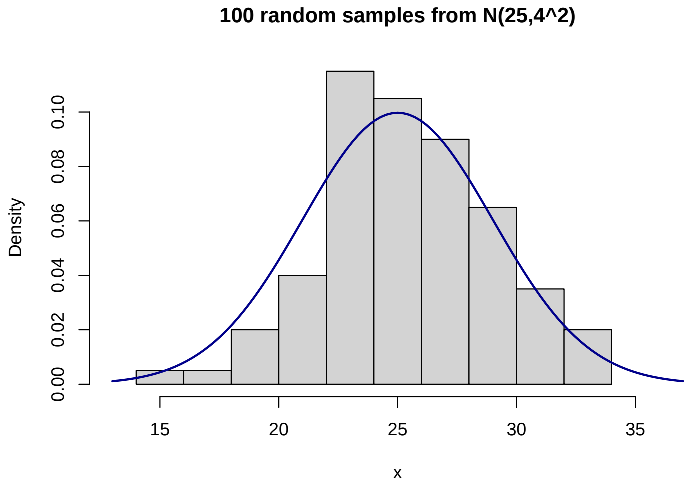
We can simulate observations from a normal distribution in a similar manner as in previous labs. For example, to draw 100 samples from a normal distribution with mean equal to 4 and standard deviation equal to 0.1 we use:

```
rnorm(n, mean = 4, sd = 0.1)
```

```
set.seed(123)
ndata <- rnorm(100, mean=25,sd=4)
mean(ndata)

## [1] 25.36162

sd(ndata)

## [1] 3.651264
```

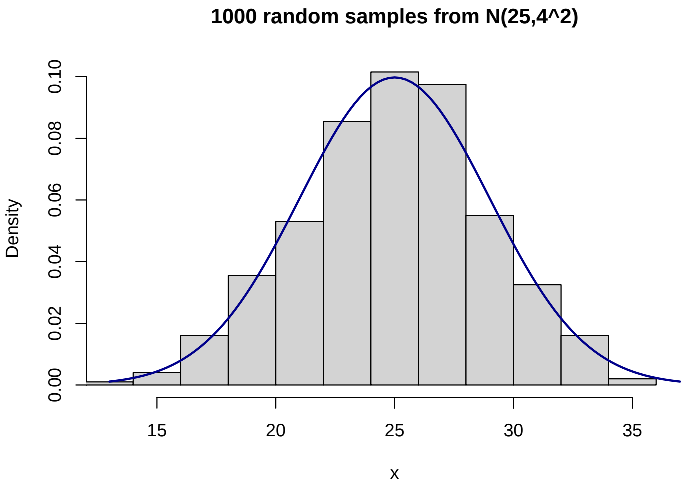
To see how the empirical distribution compares to the true pdf, run the following code:

```
hist(ndata, main="100 random samples from N(25,4^2)", freq=FALSE,
     xlab="x", xlim=c(25-3*4,25+3*4))
curve(dnorm(x,25, 4), add=TRUE, col="darkblue", lwd=2)
```



Of course, since this is just a sample, we don't see the perfectly symmetric bell shape that is exemplified by the smooth curve denoting the true pdf. You'll notice that the larger the sample, the closer the histogram will look to the curve:

```
set.seed(1000)
x1000 <- rnorm(1000, mean=25,sd=4)
hist(x1000, main="1000 random samples from N(25,4^2)",
     freq=FALSE, xlab="x", xlim=c(25-3*4,25+3*4))
curve(dnorm(x,25, 4), add=TRUE, col="darkblue", lwd=2)
```


1000 random samples from N(25,4^2)
Density
0.00
0.02
0.04
0.06
0.08
0.10
15
20
25
30
35
x