Vignette 01:

A primer on (multivariate) Gaussian distribution percentiles

Ty Stanford, Maddison Mellow, et al.

Table of contents

1 The normal distribution

Also referred to as the Gaussian distribution.

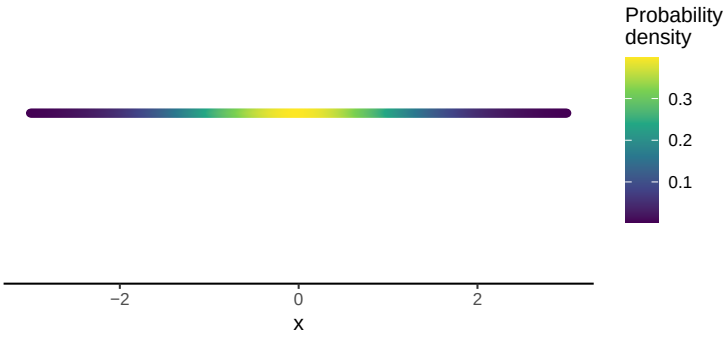
In this document we will start with the likely familiar one dimensional normal distribution and visually show its percentile "contours". Then we will introduce higher dimensional normal distributions and visually represent their respective percentile contours - that is a "*fence*" or boundary that contains the highest density of $100 \times p\%$ of the distribution.

2 One-dimensional Gaussian distribution

Often the normal distribution is depicted as a bell curve. The points on the x-axis are values the distribution can take while the curve represents the probabilistic density related to the possible values.

Another way the normal distribution could also be depicted is by a single line representing values the distribution can take (the x-axis) and the colour of the line shows the respective density value (can be thought of as a view top-down of the normal curve). For example the below shows a standard normal distribution (mean of 0 and variance of 1 written $\sim N_1(0, 1)$; the added 1 subscript to the "N" is to denote a 1-dimensional normal distribution).

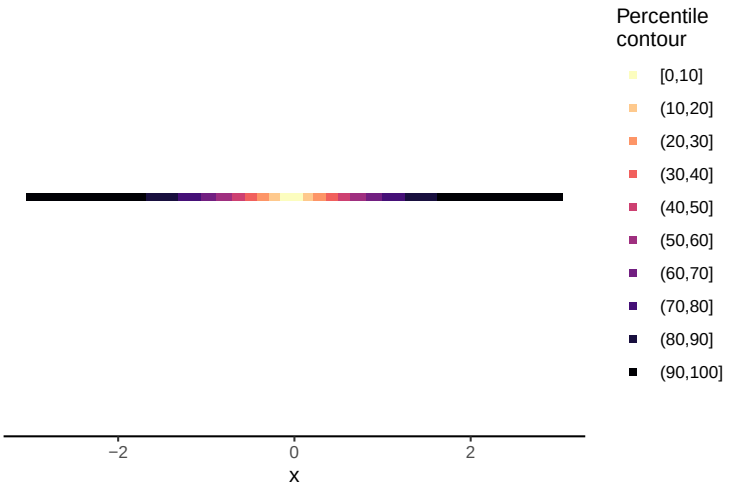
```
# 1-dim
x <- seq(-3, 3, 0.01)

tibble(x = x, y = 1, fx = dnorm(x)) %>%
  ggplot(., aes(x = x, y = y, col = fx)) +
  geom_point() +
  scale_colour_viridis_c() +
  theme_classic() + y_blank_theme() +
  labs(col = "Probability\ndensity")
```



From this we can move to quantifying the bounds/fence/contour(s) that contain the highest $100{\times}p\%$ of the distribution. Below demonstrates the fences/contours that contain the $100{\times}p\%$ percentiles for different values of p.

```
# 1-dim
x <- seq(-3, 3, 0.02)
x_mat <- matrix(x, ncol = 1)
# get_inequality_value(x_mat, 0, as.matrix(1))
# get_inequality_value(x_mat, 0, as.matrix(1), as_percentile = TRUE)
```

```r
tibble(
  x = x,
  y = 1,
  fx =
    get_inequality_value(x_mat, 0, as.matrix(1), as_percentile = TRUE)
) %>%
  mutate(fx_c = cut(fx, breaks = seq(0, 100, 10), include.lowest = TRUE)) %>%
  ggplot(., aes(x = x, y = y, col = fx_c)) +
  geom_point(shape = "square") +
  scale_colour_viridis_d(option = "A", direction = -1) +
  theme_classic() + y_blank_theme() +
  labs(col = "Percentile\ncontour")
```



Each of the percentile bands in the figure above, e.g., $(30, 40]$ = 30-40%, demonstrate the outside of that band creates a fence that encapsulates, for example 40%, the highest density of points from the mean value (0). The values of x (normal distribution values) that demarcate the highest 40% are approximately -0.52 and +0.52. i.e., we expect 40% of values from this distribution to lie between -0.52 and +0.52.

3 Two-dimensional Gaussian distribution

3.1 Uncorrelated variates

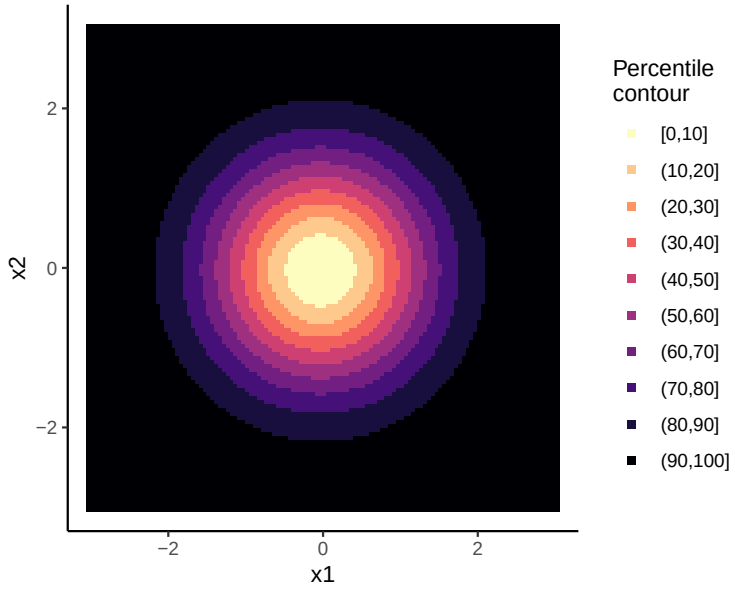
The first example of a 2D Gaussian distribution will use no covariance between the two variates (the 2-dimensions). This means the pairs of values (x_1, x_2) are unrelated to each other - the value of x_1 is unrelated to x_2, and vice-versa.

This may be written

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}_2 \left(\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right).$$

We plot an evenly generated grid of points for (x_1, x_2) below showing the contours like before (top-down type view) excvept we use the x-axis to depict (x_1, x_2) values and the y-axis to depict x_2 values.

```
# 2-dim independent
x_df <- expand_grid(x1 = seq(-3, 3, 0.05), x2 = seq(-3, 3, 0.05))
x_df$fx <-
    get_inequality_value(
      as.matrix(x_df),
      mean_vec = c(0, 0),
      covar_mat = diag(2),
      as_percentile = TRUE
    )
x_df$fx_c <- cut(x_df$fx, breaks = seq(0, 100, 10), include.lowest = TRUE)

x_df %>%
  ggplot(., aes(x = x1, y = x2, col = fx_c)) +
  geom_point(shape = "square") +
  scale_colour_viridis_d(option = "A", direction = -1) +
  theme_classic() +
  labs(col = "Percentile\ncontour")
```



Above we see that the contours radiate out from the mean at $(x_1, x_2) = (0, 0)$.

3.2 Correlated variates

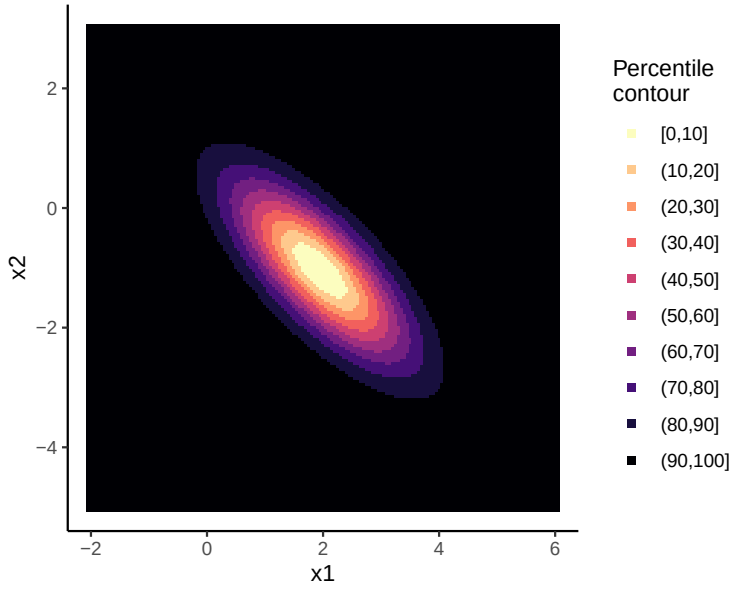
Now we consider a 2D Gaussian distribution with -0.75 correlation (-0.75 covariance too as the variance of x_1 and x_2 is 1) between the two variates. This means we expect a reasonably strong negative relationship between x_1 and x_2 - that is - for increasing values of x_1 we expect more-often-that-not decreasing values of x_2.

Note the below distribution also has a different mean vector than before.

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}_2 \left(\mu = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & -0.75 \\ -0.75 & 1 \end{bmatrix} \right).$$

```
# 2-dim independent
x_df <- expand_grid(x1 = seq(-2, 6, 0.05), x2 = seq(-5, 3, 0.05))
x_df$fx <-
    get_inequality_value(
      as.matrix(x_df),
      mean_vec = c(2, -1),
      covar_mat = matrix(c(1, -0.75, -0.75, 1), ncol = 2),
      as_percentile = TRUE
    )
x_df$fx_c <- cut(x_df$fx, breaks = seq(0, 100, 10), include.lowest = TRUE)

x_df %>%
  ggplot(., aes(x = x1, y = x2, col = fx_c)) +
```

```
  geom_point(shape = "square") +
  scale_colour_viridis_d(option = "A", direction = -1) +
  theme_classic() +
  labs(col = "Percentile\ncontour")
```



The above contours are *ellipses* which is the specific case of a 2D *ellipsoid*. *Ellipsoids* in 3D (or more) contain volume (or hyper-volume) as are to ellipses as spheres are to circles.

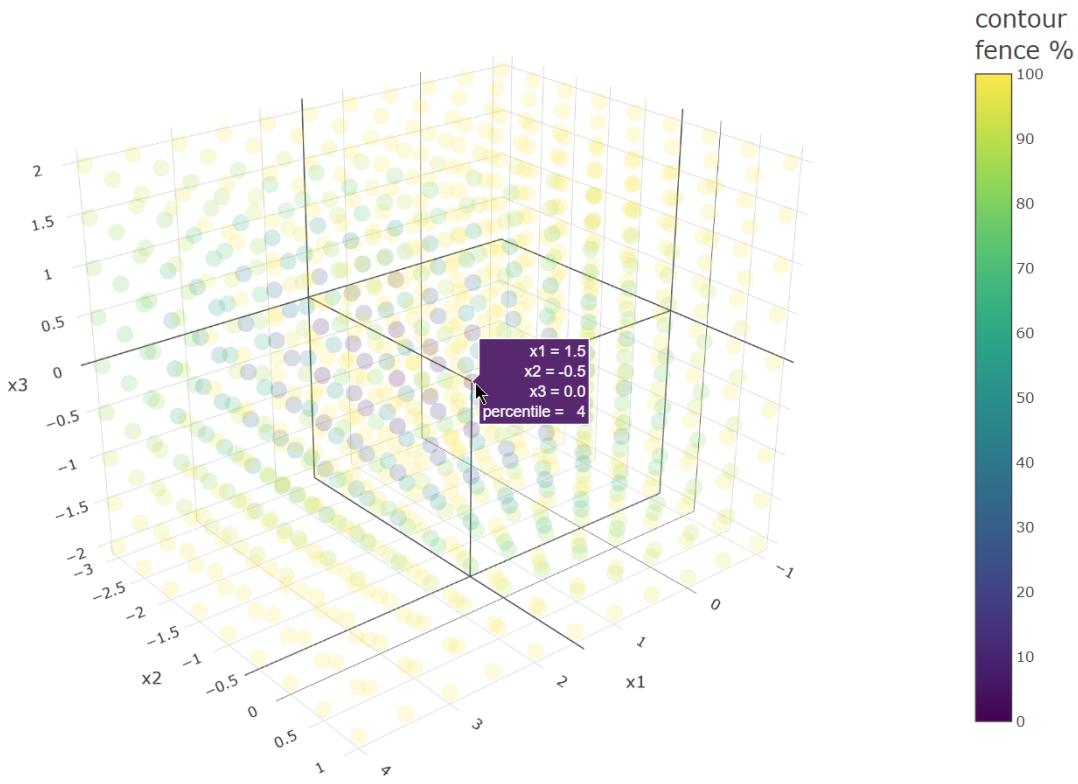
4 Three-dimensional Gaussian distribution

Three dimensions is about the limit of statically visualising Gaussian distributions. Below we produce a plot using $(x, y, z) = (x_1, x_2, x_3)$ and the colour represents the percentile contour for the following distribution:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \sim \mathcal{N}_3 \left(\mu = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & -0.75 & 0.25 \\ -0.75 & 1 & 0 \\ 0.25 & 0 & 1 \end{bmatrix} \right).$$

```
# 3 variates
x_df <- expand_grid(
  x1 = seq(-1, 4, 1/2),
  x2 = seq(-3, 1, 1/2),
  x3 = seq(-2, 2, 1/2)
)
x_df$fx <-
  get_inequality_value(
    as.matrix(x_df),
    mean_vec = c(2, -1, 0),
    covar_mat = matrix(c(1, -0.75, 0.25, -0.75, 1, 0, 0.25, 0, 1), ncol = 3),
    as_percentile = TRUE
  )
x_df$fx_c <- cut(x_df$fx, breaks = seq(0, 100, 10), include.lowest = TRUE)

# labels for plot
x_df$obs_labs <-
  paste0(
    "<br>x1 = ", sprintf("%1.1f", x_df[["x1"]]),
    "<br>x2 = ", sprintf("%1.1f", x_df[["x2"]]),
    "<br>x3 = ", sprintf("%1.1f", x_df[["x3"]]),
    "<br>percentile = ", sprintf("%3.0f", x_df[["fx"]])
  )

# make 3D points in plotly scatterplot
plty <-
  plot_ly() %>%
  add_trace(
    type = "scatter3d",
    mode = "markers",
    data = x_df,
    x = ~x1,
    y = ~x2,
    z = ~x3,
    opacity = 0.2,
    hovertext = ~obs_labs,
```

```
    hoverinfo = "text",
    hoverlabel = list(
      align = "right",
      bgcolor = map_cts_to_scale(x_df[["fx"]])
    ),
    marker = list(color = ~fx, showscale = TRUE, colorscale = "Viridis")
  )

### not plotting as html output
### but inserting screenshot below
# plty
```



5 More than three-dimensions

In higher dimensional spaces we can look at all univariate and all combinations of pair-wise variable plots of points to try to visualise and understand data structure.