Systems Genetics
Lecture 02 - Primer in statistical modeling

Julien Gagneur

30 May 2020

Package

BiocStyle 2.10.0

Contents

1 Forewords

This lecture provides basic concepts in statistical modeling that are needed for the module Statistical Modeling in Systems Genetics.

As prerequisite for the module, we assume you are familiar with:

- Probabilities and Statistics, including hypothesis testing
- Linear Algebra
- Calculus, including multivariate calculus (Gradient, Hessian)

The exercises will be done in R. We recommend the module Data Analysis and Visualization in R. This very short tutorial can make you quickly familiar with basic R syntax:

https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

2 Notations and basics of probabilities

Lower cases are used for scalars (e.g. x), bold lower cases for vectors (e.g. $\mathbf{x}$) and bold upper cases for matrices (e.g. $\mathbf{X}$). The transpose of a matrix is denoted with a T-superscript (e.g.. $\mathbf{X}^\top$).

We indistinguishably denote $p(a)$:

- the probability of a logical event A to occur
- the probability of a discrete random variable A to take the value a
- the probability mass density of a continuous random variable A at the value a

The random variables a and b are *independent*, denoted $a \perp\!\!\!\perp b$, if and only if

$$p(a,b) = p(a)p(b) \tag{1}$$

Otherwise, the random variables a and b are *dependent*, denoted $a \not\perp\!\!\!\perp b$.

The joint probability of two events to occur (two random variables to take particular values) is denoted $p(a,b)$:

$$p(a,b) := p(a \wedge b) \tag{2}$$

The probability of either event a or b occur, denoted $p(a \vee b)$, equals to:

$$p(a \vee b) = p(a) + p(b) - p(a,b) \tag{3}$$

The conditional probability of an event a given that b occurs, denoted $p(a|b)$ and said “probability of a given b”, is defined as:

$$p(a|b) := \frac{p(a,b)}{p(b)} \tag{4}$$

It follows the Bayes theorem:

$$p(b|a) = \frac{p(a|b)p(b)}{p(a)} \tag{5}$$

These results are easy to recollect from an ensemblist point of view. Consider the union and intersection sets in Figure 1.

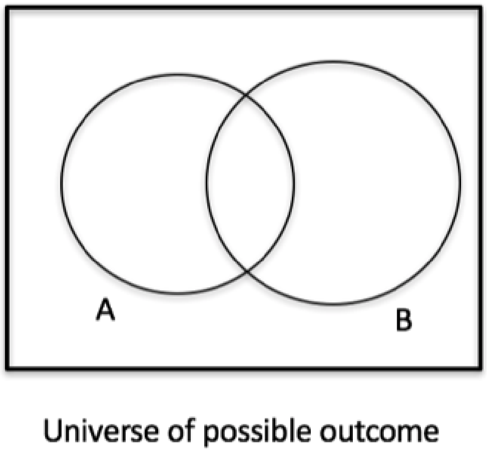

Figure 1: **Probability basics - Ensemblist point of view**

Moreover, these results generalize to discrete random variables and to probability mass densities of continuous random variables.

3 Maximum likelihood

3.1 Notations and definitions

We will model data as being generated from a stochastic process. Generally we will have data for n observations (e.g. individuals). The data will be multi-dimensional, with p the number of variables. For instance, we can observe for a given individual $p = 3$ quantities: her height, her weight, and whether she is affected by a specific disease. The i-th vector of observations will be denoted $\mathbf{x}_i$. The entire $n \times p$ matrix of data will be denoted $\mathbf{X}$. Hence, the rows of $\mathbf{X}$ are the transposed observation vectors and the columns correspond to the different variables.

A data generative model is a model of the stochastic process generating the data. The model may have multiple parameters. We often denote the vector of parameters $\boldsymbol{\theta}$. Specifically, the data generative model is a mathematical function $p(\mathbf{X}|\boldsymbol{\theta})$ that specifies the probability (or the probability mass function when data contains continuous quantities) of the data given the parameters.

The parameters of the model are not known. They will be estimated from the data. The *likelihood* is defined as:

$$\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}) := p(\mathbf{X}|\boldsymbol{\theta}) \tag{6}$$

The likelihood is hence nothing else than the probability of the data given specific values of the parameters but considered as a function of the parameters for a fixed dataset. The *maximum likelihood* estimate of the parameter is the one maximizing the likelihood. Hence, it is the value of parameters for which the data has highest probability. Estimates are denoted with a hat (e.g. $\hat{\boldsymbol{\theta}}$). In this lecture we will only consider maximum likelihood estimates.

Applying the maximum likelihood principle turns model parameter estimation into an optimization task for which we can make use of analytical (e.g. solving gradient=0) or numerical techniques (gradient descent, etc.).

Very often, the observations will be considered to be *independently and identically distributed* (short: i.i.d.) given the model parameters. Under this assumption, we have:

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1\dots n} p(\mathbf{x}_i|\boldsymbol{\theta}) \tag{7}$$

3.2 Modeling and parameter estimation strategy

We will proceed with the following steps:

1. **Write the data generative model**, i.e. the probability of the data given the parameters. Make sure to have clear notations distinguishing the data from the parameters.
2. **Simulate data** using the data generative model. This has several advantages: i) It makes the model concrete thereby allowing you to "debug" conceptual issues, ii) it allows generating ground truth data for which the data are ideally distributed and the values of the parameters known. This will be helpful to debug parameter estimation code and to empirically assess the robustness, bias, variance of the estimation procedure under various (yet ideal) conditions.
3. **Derive a parameter estimation procedure**. This is ideally but rarely a closed-form solution. Otherwise we will use numerical optimization.
4. **Implement the estimation and verify its accuracy using various simulated datasets**.
5. **Apply to real data**

We will see later on further techniques to assess whether the modeling assumptions seem to be reasonable for the data at hand.

3.3 Example: Univariate Gaussian

As an example, we consider the heights in cm of 20 individuals:

```
heights
##  [1] 178 162 178 178 169 150 156 162 165 189 173 157 154 162 162 161 168 156 169
## [20] 153
```

We will model the heights using the univariate Gaussian. The univariate Gaussian has two parameters, its mean and variance, which we aim to estimate.

3.3.1 Step 1. Data generative model

We denote $\{x_i\}_{i=1\dots n}$ the height values. We assume the data to be i.i.d. Our data generative model is:

$$p(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1\dots n} p(\mathbf{x}_i|\boldsymbol{\theta}) \tag{8}$$

$$= \prod_{i=1\dots n} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x_i-\mu)^2}{2\sigma^2}) \tag{9}$$

,

The parameters of the model are μ, the mean and σ, the standard deviation (or equivalently σ^2, the variance). These are to be estimated.

$$p(x|\mu,\sigma^2) = \mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{10}$$

3.3.2 Step 2. Simulate data

We can simulate data according to the model using the R function `rnorm()` which draws random samples of the normal distribution.

```r
simulate <- function(n, theta){
  x <- rnorm(n, mean=theta[["mu"]], sd=theta[["sigma"]])
  return(x)
}
n <- 20
x <- simulate(n, theta=c(mu=175, sigma=5))
x
##  [1] 174 177 176 179 175 178 180 172 169 175 174 172 173 172 179 181 180 173 181
## [20] 174
```

3.3.3 Step 3. Parameter estimation procedure

We consider maximum likelihood estimation of the parameters, i.e.:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta};\mathbf{X}) \tag{11}$$

Two tricks will simplify the calculations. The first one is to consider minimizing the negative log-likelihood $\mathrm{NLL}(\boldsymbol{\theta}) := -\log \mathcal{L}(\boldsymbol{\theta};\mathbf{X})$, which is equivalent to maximizing the likelihood. The second is to reparametrize the Gaussian using the precision, defined as the inverse of the variance ($\tau := 1/\sigma^2$).

The parameter estimates are obtained by setting the gradient of the negative log-likelihood with respect to the vector parameters $\boldsymbol{\theta} = (\mu,\tau)^T$ to $\mathbf{0}$:

$$\nabla_{\boldsymbol{\theta}} \mathrm{NLL}(\boldsymbol{\theta};\mathbf{X}) = \mathbf{0} \tag{12}$$

The NLL reads:

$$\mathrm{NLL} = -\frac{n}{2}\log(\tau) + \frac{\tau}{2}\sum_{i=1\ldots n}(x_i-\mu)^2 + \frac{n}{2}\log(2\pi) \tag{13}$$

Setting the derivative with respect to μ to 0 leads to:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1\ldots n} x_i \tag{14}$$

Injecting this estimation of μ and setting the derivative with respect to τ to 0 leads to:

$$\hat{\tau} = \frac{1}{\frac{1}{n}\sum_{i=1\ldots n}(x_i-\hat{\mu})^2} \tag{15}$$

and thus:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1\ldots n}(x_i-\hat{\mu})^2 \tag{16}$$

Hence the maximum likelihood estimate of μ is the sample mean and the maximum likelihood estimate of σ^2 is the so-called biased sample variance. One can show that the maximum likelihood approach systematically underestimates the variance of the distribution. This is also known as bias (Bishop 2007, 27).

3.3.4 Step 4. Implementation and empirical verification

We code our estimation procedure into a R function `estimate()`.

```r
estimate <- function(x){
  n <- length(x)
  theta_hat <- c(
    mu = mean(x),
    sigma = sqrt(mean((x-mean(x))^2))
  )
  return(theta_hat)
}
```

```r
n <- 20
x <- simulate(n, theta=c(mu=175, sigma=5))
theta_hat <- estimate(x)
theta_hat
##     mu  sigma
## 174.42   5.58
```

These estimates are not too far from the ground truth values. Our simple check is good enough for this didactic toy example. The code would allow you to investigate more systematically the relationships between estimates and ground truth with various values of the parameters and sample size n.

3.3.5 Step 5. Application to real data

We finally apply our estimation to the original dataset:

```
theta_hat <- estimate(heights)
theta_hat
##     mu  sigma
## 165.10   9.89
```

4 Assessing whether a distribution fits data with Q-Q plots

4.1 Motivation

The strategy described in section 1 allows assessing whether an estimation procedure returns reasonable estimates on simulated data. It does not assess however ether the simulation assumptions, i.e. the data generative model, is a reasonable model for the data at hand. One key modeling assumption of a data generative model is the choice of the distribution. The quantile-quantile plot is a graphical tool to assess whether a distribution fits the data reasonably.

As a concrete example, let's consider 50 data points coming from the uniform distribution in the [2,3] interval. If you assume your data comes from the uniform distribution in the [2,3] interval, you expect the first 10% of your data to fall in [2,2.1], the second 10% in [2.1,2.2] and so forth. A histogram could be used to visually assess this agreement. However, histograms are shaky because of possible low counts in every bin:

```
par(cex=0.7)
u <- runif(50, min=2, max=3) ## uniformly distributed data points
hist(u, bin=10, main="")
```

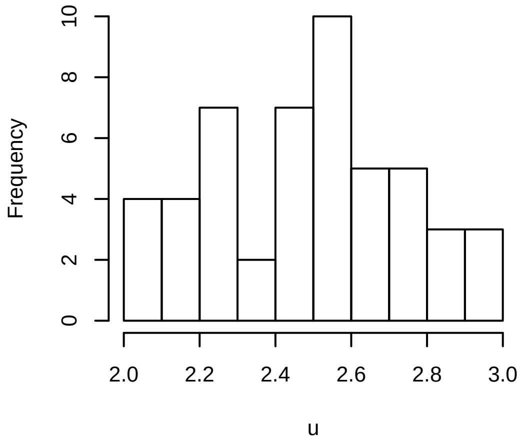

Figure 2: Histogram of u

Instead of the histogram, one could plot the deciles of the sample distribution against those of a theoretical distribution. Here are the deciles:

```
dec <- quantile(u, seq(0,1,0.1))
dec
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100% 
## 2.04 2.17 2.23 2.32 2.45 2.50 2.56 2.61 2.72 2.87 2.99
```

We can now compare these empirical deciles with the values of the theoretical ones.

```
par(cex=0.7)
plot(seq(2,3,0.1),
     dec,
     xlim=c(2,3), ylim=c(2,3),
     xlab="Deciles of the uniform distribution over [2,3]",
     ylab="Deciles of the dataset")
abline(0,1) ## diagonal y=x
```

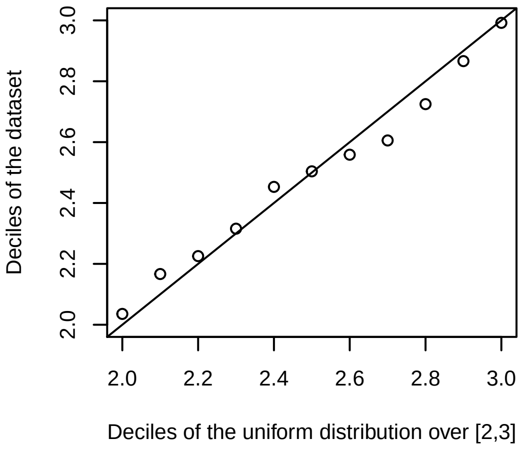

Figure 3: Decile scatterplot

Now we see a clear agreement between the expected values of the deciles of the theoretical distribution (x-axis) and those empirically observed (y-axis). The advantage of this strategy is that it also generalizes to other distributions (e.g. Normal), where the shape of the density can be difficult to assess with a histogram.

4.2 Definition

For a finite sample we can estimate the quantile for every data point, not just the deciles. The Q-Q plot scatter plots the quantiles of two distributions against each other. One way is to use as expected quantile $(r - 0.5)/N$ (Hazen, 1914), where r is the rank of the data point. The R function `ppoints` gives more accurate values:

```
par(cex=0.7)
plot(qunif(ppoints(length(u)), min=2, max=3), sort(u),
     xlim=c(2,3), ylim=c(2,3),
     xlab="Quantiles of the uniform distribution over [2,3]",
     ylab="Quantiles of the dataset")
abline(0,1) ## diagonal y=x
```

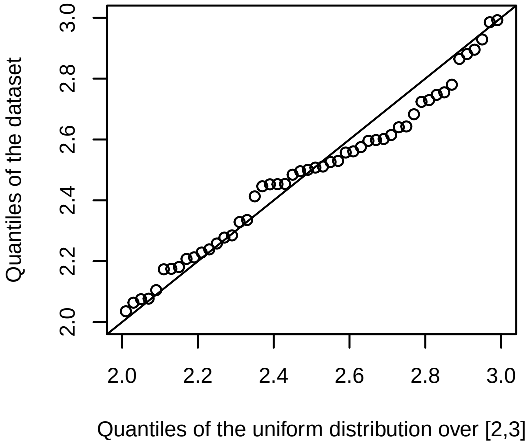

Figure 4: Q-Q plot of u against U[2,3]

In R, Q-Q plots between two datasets can be generated using the function `qqplot()`. In the special case of a normal distribution use the function `qqnorm()` and the function `qqline()`, which adds a line to the "theoretical" quantile-quantile plot passing through the first and third quartiles.

4.3 Examples

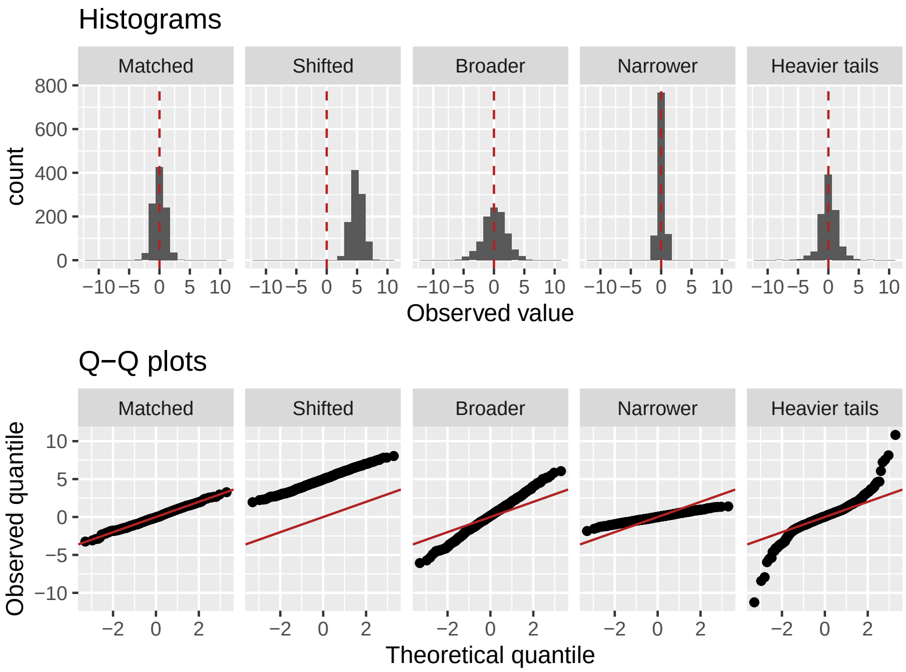

Figure 5: Various Q-Q plots

Figure 5 shows histograms (top row) and Q-Q plots against the normal distribution (bottom row) of datasets simulated according to:

i) the normal distribution $\mathcal{N}(0, 1)$
ii) a mean-shift $\mathcal{N}(5, 1)$
iii) a larger variance $\mathcal{N}(0, 4)$
iv) a smaller variance $\mathcal{N}(0, 1/4)$
v) a distribution with heavier tails (Student t with 2 degrees of freedom)

Generally, mean shifts affect the positions along the y-axis, changes of variance affect the slope, and heavier tails tend to give increasing deviations off the diagonal.

We conclude by checking normality of the `heights` dataset with the native R `qqnorm()` function:

```r
par(cex=0.7)
qqnorm(heights)
qqline(heights)
```

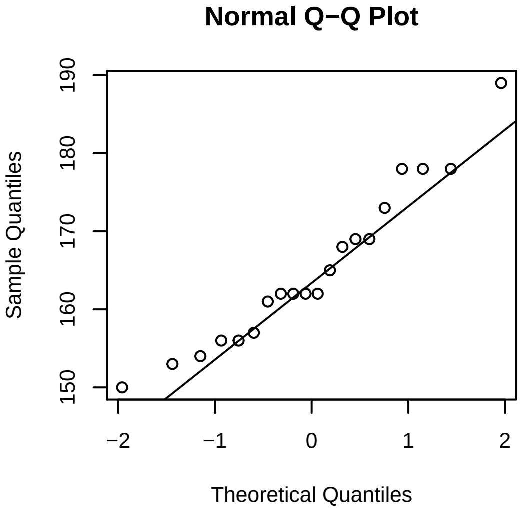

Figure 6: **Q-Q plot of heights against normal distribution**

5 Hypothesis testing

5.1 Motivation

Hypothesis testing is frequently used in quantitative genetics in general and in Systems Genetics in particular. We refer to elsewhere (e.g. Irizarry (2019)) for introduction to Hypothesis testing. Here we provide a refresher focusing on Student's t-test.

5.2 One-sample Student's t-test

We consider the one-sample t-test applied to the `heights` dataset above. The average height of females in Germany is 165 cm. We test whether the expectation of the underlying distribution of our 20-observation sample `heights` is significantly different than 165cm.

The Student's t statistics is defined as:

$$t = \frac{\bar{x} - \mu}{\widehat{\sigma}/\sqrt{n}} \quad \boxed{17}$$

where $\bar{x}$ is the sample mean, $\widehat{\sigma}$ is the unbiased estimate of the standard deviation, and μ is the population mean under the null hypothesis (here 165 cm).

5.3 P-value

The p-value is the probability that the test statistics would be the same as or more extreme than the actual observed results under the null hypothesis. "More extreme" can be taken as greater, lesser, or either way:

- For right-tail events: $P = p(t \geq t_{\text{obs}}|H_0)$
- For left-tail events: $P = p(t \leq t_{\text{obs}}|H_0)$
- For double tail events: $P = 2\min\{p(t \leq t_{\text{obs}}|H_0), p(t \geq t_{\text{obs}}|H_0)\}$

The null hypothesis is said to be rejected for sufficiently small p-values. Literature usually uses $P < 0.05$.

5.4 Student's t-distribution

Under the null hypothesis and assuming that the observations are normally and i.i.d., the t statistics follows a Student's t-distribution with $n-1$ degrees of freedom.

The Student's t-distribution has heavier tails than the normal distribution. This intuitively comes from the fact that, while the numerator of the t-statistics is normally distributed, the estimate of the standard deviation in the denominator is noisy. The smaller the sample size n, the noisier the estimate. Hence, the smaller the degrees of freedom, the heavier the tails. For infinite degrees of freedom, Student's t-distribution equals the normal distribution.

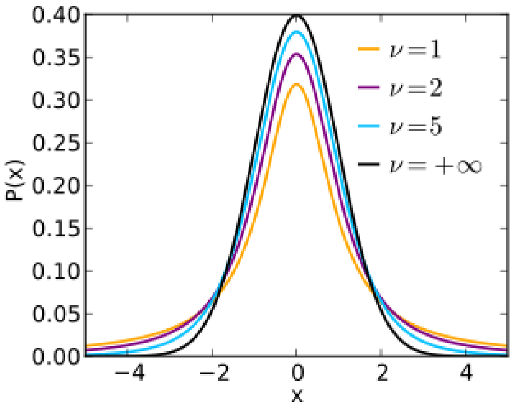

Figure 7: Student's t distribution (source: Wikipedia)

5.5 Application

We compute below the t-statistics and the two-sided p-value using the cumulative distribution function of Student's t distribution `pt()`. Note that `sd()` returns the unbiased standard deviation estimate.

```
n <- length(heights)
t_stat <- (mean(heights) - 165)/(sd(heights)/sqrt(n))
2*min(
  pt(t_stat, df= length(heights)-1),                      # p(t <= t_obs| H_0)
  pt(t_stat, df= length(heights)-1, lower.tail=FALSE)   # p(t >= t_obs| H_0)
)
## [1] 0.965
```

This matches the R implementation of the one-sample Student's t-test:

```
t.test(heights, mu=165)$p.value
## [1] 0.965
```

The p-value is not lesser than 0.05. We do not reject the null hypothesis that the expectation equals to 165 cm.

6 Multivariate normal distribution

6.1 Definition

The multivariate normal distribution (short MVN, also known as the multivariate Gaussian distribution) arises in many application contexts. We state here its basic properties which will be helpful for most of the models of the module.

The density $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of a MVN for a p-dimensional vector $\mathbf{x}$ with p-dimensional mean $\boldsymbol{\mu}$ and $p \times p$ covariance matrix $\boldsymbol{\Sigma}$ is given by:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\} \tag{18}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$.

We remind that the covariance of a multivariate random variable $\mathbf{x}$ is defined as:

$$\mathrm{cov}[\mathbf{x}] = \mathrm{E}[(\mathbf{x} - \mathrm{E}[\mathbf{x}])^\top(\mathbf{x} - \mathrm{E}[\mathbf{x}])] \tag{19}$$

The element (i, j) of $\boldsymbol{\Sigma}$ is such that:

$$\begin{aligned} \sigma_{i,j} &= \mathrm{E}[(x_i - \mu_i)(x_j - \mu_j)] \qquad (20) \\ &= \rho_{i,j}\sigma_i\sigma j \qquad (21) \end{aligned}$$

where $\rho_{i,j}$ is the Pearson correlation coefficient between x_i and x_j.

For MVNs (but not generally for any distributions), it holds that:

$$\rho_{i,j} = 0 \iff x_i \perp\!\!\!\perp x_j \tag{22}$$

6.2 Special cases

The MVN is often used to describe the joint distribution of multiple independent univariate normal variables.

Specifically, if $\{x_i\}_{i=1...n}$ are n independent normal random variables with means μ_i and variance σ_i^2, then

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{23}$$

,

where:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{pmatrix} \tag{24}$$

If, moreover, the x_i are i.i.d. (hence all means and variances are equal), then the covariance is proportional to the identity matrix $\mathbf{I}$:

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{x}|\mu\mathbf{1}, \sigma\mathbf{I}) \tag{25}$$

, where $\mathbf{1}$ is the vector of ones.

6.3 Geometry

The functional dependence of the MVN density on $\mathbf{x}$ is through the quadratic form:

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{26}$$

where the quantity Δ is called the *Mahalonobis distance* from $\mathbf{x}$ to $\boldsymbol{\mu}$. Because the covariance matrix of a well-defined MVN is symmetric positive definite, Δ and therefore the MVN density are constant along ellipsoid surfaces centered on $\boldsymbol{\mu}$ and whose axes are the eigendirections of $\boldsymbol{\Sigma}$ (Figure 8). Proofs can be found in (Bishop 2007).

Denoting $\mathbf{u}_1, ..., \mathbf{u}_p$ the eigenvectors of $\boldsymbol{\Sigma}$ with eigenvalues $\lambda_1 > ... > \lambda_p$, the quadratic form becomes:

$$\Delta^2 = \sum_i \frac{y_i^2}{\lambda_i} \tag{27}$$

where the y_i's are the coordinates of $\mathbf{x}$ in the coordinate system centered on μ and defined by the eigenvectors $\mathbf{u}_1, ..., \mathbf{u}_p$, i.e.:

$$y_i = \mathbf{u}_i^\top (\mathbf{x} - \boldsymbol{\mu}) \tag{28}$$

or, in matrix form:

$$\mathbf{y} = \mathbf{U}(\mathbf{x} - \boldsymbol{\mu}) \tag{29}$$

where the transposed eigenvectors are the rows of the matrix $\mathbf{U}$.

In this coordinate system (Figure 9), the MVN takes the form:

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_p \end{pmatrix}) \tag{30}$$

Hence, the change of coordinate leads to independent centered Gaussian variables. Their variances are the eigenvalues of $\mathbf{\Sigma}$.

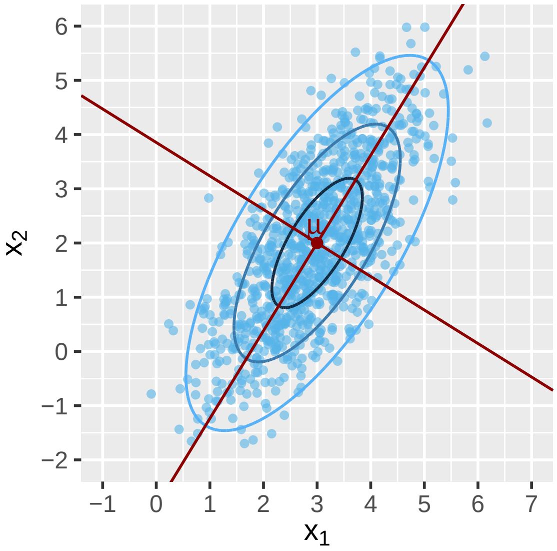

Figure 8: **Geometry of the mutivariate normal distribution**
1,000 random draws a bivariate MVN. Equiprobability curves are ellipses centered on the mean (red dot). Their axes (red) correspond to the eigenvectors of the covariance matrix.

6.4 Partitioned Gaussians

The following results will be instrumental for many models seen in the module. Proofs can be found in Bishop (2007) (page 85 to 87). You are not expected to know the formulae by heart but should be able to apply them.

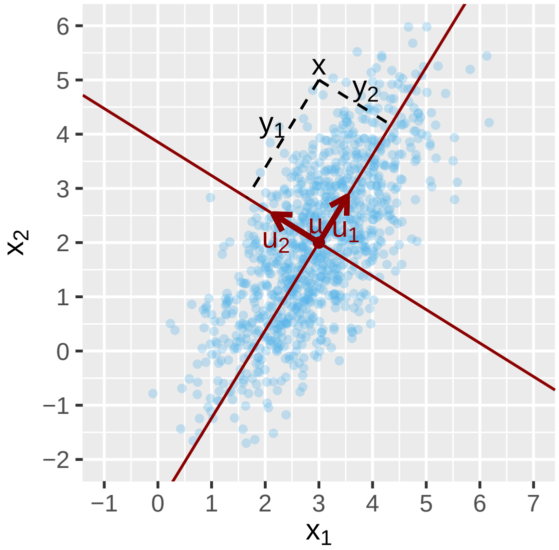

Figure 9: Transformed coordinates of an MVN
The coordinates y of a vector x in the coordinate systems defined by the eigenvectors (u1 and u2) and centered on the mean vector (mu)

Some expressions are more easily written using the precision matrix $\mathbf{\Lambda}$ defined as the inverse of the covariance matrix:

$$\mathbf{\Lambda} := \mathbf{\Sigma}^{-1} \tag{31}$$

Mind that the notation Λ for the precision matrix shall not to be confused with the notation λ_i for the eigenvalues of Σ.

Given a MVN $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma})$ with $\mathbf{\Lambda} = \mathbf{\Sigma}^{-1}$. We consider a partition of the p variables into two sets, leading to

$$\mathbf{x} = \left(\begin{array}{c} \mathbf{x}_a \\ \mathbf{x}_b \end{array} \right), \boldsymbol{\mu} = \left(\begin{array}{c} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{array} \right) \tag{32}$$

and

$$\mathbf{\Sigma} = \left(\begin{array}{cc} \mathbf{\Sigma}_{aa} & \mathbf{\Sigma}_{ab} \\ \mathbf{\Sigma}_{ba} & \mathbf{\Sigma}_{bb} \end{array} \right), \mathbf{\Lambda} = \left(\begin{array}{cc} \mathbf{\Lambda}_{aa} & \mathbf{\Lambda}_{ab} \\ \mathbf{\Lambda}_{ba} & \mathbf{\Lambda}_{bb} \end{array} \right) \tag{33}$$

The marginal distribution is a MVN with the following simple form:

$$p(\mathbf{x}_a) = \mathcal{N}\left(\mathbf{x}_a | \boldsymbol{\mu}_a, \mathbf{\Sigma}_{aa}\right) \tag{34}$$

The conditional distribution is also a MVN:

$$p(\mathbf{x}_a | \mathbf{x}_b) = \mathcal{N}\left(\mathbf{x} | \boldsymbol{\mu}_{a|b}, \mathbf{\Lambda}_{aa}^{-1}\right) \tag{35}$$

where

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \mathbf{\Lambda}_{aa}^{-1} \mathbf{\Lambda}_{ab} (\mathbf{x}_b - \boldsymbol{\mu}_b) \tag{36}$$

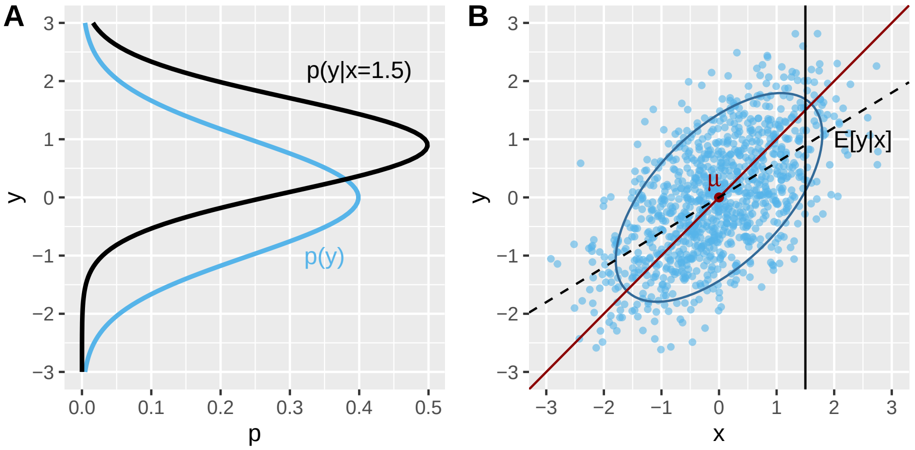

Figure 10: Marginal and conditional distributions of a MVN
(A) Marginal density p(y) (blue) and conditional density p(y|x=1.5) (black) for the bivariate MVN shown in panel B). (B) 1,000 random draws of a bivariate MVN, its mean (red dot), its first eigendirection (red), and one equiprobability curve (ellipse). The black dashed line marks the conditional expectation values E[y|x]. Note that E[y|x] passes by the MVN mean and is not as steep as the first eigendirection ("regression towards the mean"). The vertical line marks the value x=1.5.

6.5 Linear systems

Linear systems whereby the mean of a MVN depends linearly of another MVN often occur in the models we will study. The following properties are then useful.

Assuming:

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right) \quad 37$$

$$p(\mathbf{y}|\mathbf{x}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}\right) \quad 38$$

The marginal distribution $p(\mathbf{y})$ is again a MVN such that:

$$p(\mathbf{y}) = \mathcal{N}\left(\mathbf{y}|\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^{\top}\right) \quad 39$$

Moreover, the conditional distribution $p(\mathbf{x}|\mathbf{y})$ is also a MVN and it is such that:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\Sigma}\left\{\mathbf{A}^{\top}\mathbf{L}(\mathbf{y} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\right\}, \boldsymbol{\Sigma}\right) \quad 40$$

where

$$\boldsymbol{\Sigma} = \left(\boldsymbol{\Lambda} + \mathbf{A}^{\top}\mathbf{L}\mathbf{A}\right)^{-1} \quad 41$$

The derivations of these results can be found in Bishop (2007) (page 91 to 93).

7 Acknowledgements

This document was prepared with the help of Ines Scheller and Nils Wagner.

References

Bishop, Christopher M. 2007. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1st ed. Springer. https://www.microsoft.com/en-us/research/people/cmbishop/prml-book/.

Irizarry, R.A. 2019. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R*. Chapman & Hall/Crc Data Science Series. CRC Press. https://rafalab.github.io/dsbook/.