Rebecca Nauli

GSTAT Summary

Inductive Statistics

General procedure: Translate what is known about the phenomenon under investigation into a statistical model and the question of interest into a question about unknown parameters of the model.

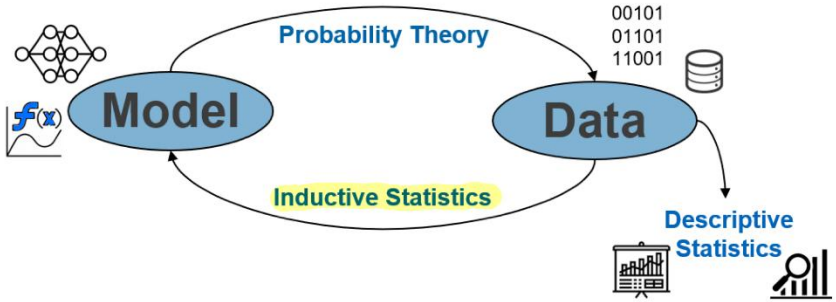


Figure 1: Relation of data and model in probability theory and inductive statistics

Definitions

Model: is an idealised picture of the population and also includes aspects of the data collection process.

Parametric models: a **specific family of distributions** is assumed **(normal, binomial, ...).** Only one or at most finitely many parameters are unknown.

Non-parametric models: are not models that have **no unknown parameters**, but the determination of individual parameters does not uniquely define the distribution. One can estimate:

- *Expected value*
- *Variance*
- *Median*
- *75% quantile*

Without having to make a specific distribution assumption. Non-parametric models make fewer assumptions and are therefore more universally applicable. However, the accuracy of the results is often reduced.

Terms:

Realizations:	e.g. the lifetime of a randomly selected component = realization of a rv X.
Random variable (rv):	A variable which can take different values by chance under certain constant condition is called random variable.
iid:	independent and identical distributed
μ:	mean
σ:	standard deviation (SD)
σ^2:	Variance (Var)

Examples and Distributions

- *Machine producing screw → normal distribution*
- *Fish in a lake (capture-recapture) → urn problem – hypergeometric distribution*
- *Duration of components → exponential distribution*

Simulation

replicate()

function() {}

Example 1

```
getpVals <- function(mu = 0, n = 10, nsim = 10000){
    replicate(nsim,
              t.test(rnorm(n, mean = mu, sd = 1), alternative="two.sided")$p.value)
  }
```

Example 2

```
sim <- function(n) {
  x <- rnorm(n, 0, 1)
  ci_z1 <- z.test(x = x, sigma.x = 1)$conf.int # true value, unknown in practice
  ci_z2 <- z.test(x = x, sigma.x = sd(x))$conf.int # cheating
  ci_t <- t.test(x = x)$conf.int # correct one
  c(in_z1 = ((ci_z1[1] <= 0) & (ci_z1[2] >= 0)),
    in_z2 = ((ci_z2[1] <= 0) & (ci_z2[2] >= 0)),
    in_t = ((ci_t[1] <= 0) & (ci_t[2] >= 0)))
}
```

Note: the difference between knowing the variance and estimating it is not so relevant in large samples but substantial in small samples.

Rebecca Nauli

Law of Large Numbers

Law of Large Numbers (LLN) — states that the arithmetic mean approaches the expected value with increasing n

With

$$\overline{X_n} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

$$E(\bar{X}_n) = \mu$$

$$Var(\bar{X}_n) = \frac{\sigma^2}{n}$$

Weak LLN — for any $\varepsilon > 0$

$$\lim_{n\to\infty} P(|\bar{X}_n - \mu| > \varepsilon) = 0$$

Central Limit Theorem (CLT)

As n increases, the arithmetic mean of $\boldsymbol{X_1, X_2, \ldots, X_n}$ **behaves like a normally distributed random variable** with expectation equal to the expectation of the X_i and variance converging to zero. → stronger statement than LLN

Consider a sequence $X1, X2, \ldots$ of independent identically distributed random variables with $E(Xi) = \mu$ and $Var(Xi) = \sigma^2$.

Then:

$$\sqrt{n} * (\overline{X_n} - \mu)\sigma \cdot\sim N(0,1)$$

or

$$X_n \cdot\sim N\ \mu, \sigma\ 2\ n$$

Summary:

- *With increasing n, sum and arithmetic mean of iid rv's behave – independently of the original distribution! – more and more like normally distributed random variables*
- *How large n has to be for a good approximation does depend on the original distribution of the Xi*
- *Rule of thumb: n Ø 25 or n Ø 30 I The approximation is already good for small n, if the shape of the original distribution is already very similar to the normal distribution*
- *Larger n required if distribution is strongly asymmetric or has heavy tails.*
- *The approximation "in the middle" of the distribution is better than in the tails for an accurate approximation of extreme quantiles we need a larger n*
- *There are also distributions that are so "wide" that the variance or even the expected value do not exist (e.g. Cauchy distribution) → CLT does not work!*

Parameter Estimation

Point Estimation

Point estimate: only a concrete **best guess (single value)** for one or more unknown parameter is wanted.
Estimate unknown parameters of distribution models for data

Model: $X_1, X_2, \ldots X_n$ drawn iid from some distribution with unknown parameters θ
Goal: Estimate θ

Properties:

- *Unbiasedness:* $E(T) = \theta$
- *Asymptotic unbiasedness:* $\lim_{n\to\infty} T(X_1, \ldots, X_n) = \theta$ $(weaker\ than\ unbiasedness)$
- *Efficiency, low* ***mean square error*** $\boldsymbol{MSE(T)} = E((T-\theta)^2) = Var(T) + Bias(T)^2$
- *Consistency:*
- *Robustness: usually with respect to outliers*

Plug-In

Idea: Estimate theoretical parameters by analogous quantities from the sample.
Simplest example: Estimate expectation E(X) by the sample arithmetic mean X¯n. Sometimes, we have to solve an equation, i.e. write a parameter as a function of something we can estimate from the sample. Example: Estimate $\theta = 1/E(X)$ *by* $T(X_1, \ldots, X_n) = 1/\bar{X}n$.

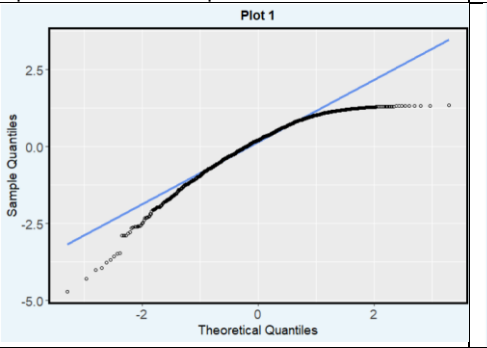
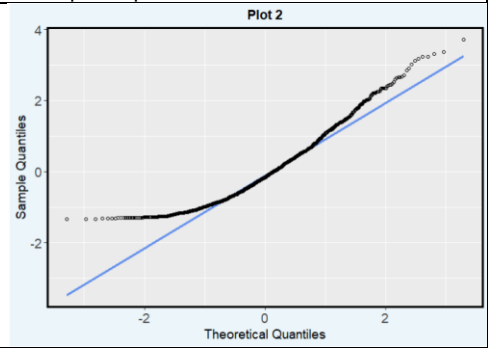
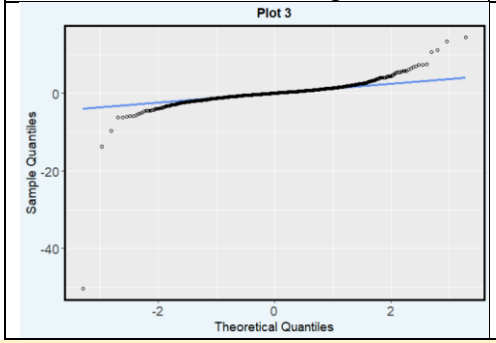
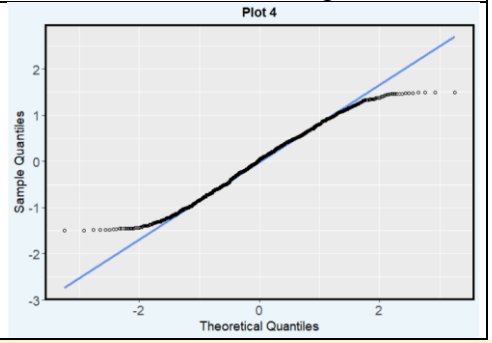
- *Simple method, often leads to reasonable estimators.*
- *Solution is usually not unique, as different sample quantities may correspond to the parameter.*
- *Desirable properties of estimator not necessarily guaranteed.*
- *May occasionally give impossible values.*

Note:

- *The variance of a consistent estimator does not converge to zero if the sample size goes to infinity → FALSE*

QQ-Plot

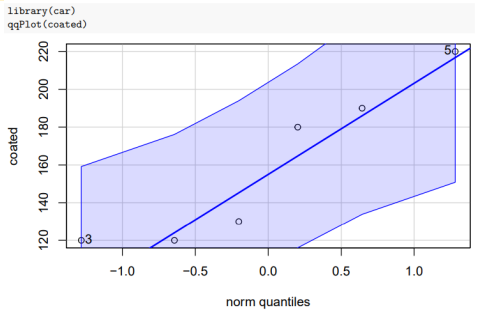
used to check whether two data sets have the same distribution

Left skewed	Right skewed
Upside down U-shape → below the curve	U-shape → points lie above curve
	
Long-tailed (heavier tails)	Short-tailed
Below on left and above on right	Above on left and below on right
	

library(car)

*qq**P**lot()*

```
library(car)
qqPlot(tires$Profile_A - tires$Profile_B)
```




```
## [1] 5 3
```

Maximum Likelihood Estimation

Idea: Estimate the parameter by the value for which the sample is most typical, i.e. choose θ to maximize

$L(\theta) = P_\theta(X_1, \ldots, X_n)$ (discrete case)

or

$L(\theta) = f_\theta(X_1, \ldots, X_n)$ (continuous case).

Usually simpler: maximize $\ell(\theta) = log\, L(\theta)$.

In the iid case, simplified by factorizing L, ℓ becomes a sum.

Maximization sometimes possible by analytical solution, in more complex situation usually solved numerically.

Under weak conditions, maximum likelihood estimates are consistent, asymptotically most efficient, asymptotically normal.

Log-likelihood

MLE

dlogis(), plogis(), qlogis(), rlogis()

Example 1

For m, we can just use

$$\hat{m}_{plugin} = \overline{X_n}$$

Another possibility would be to use the median. For s, we get from

$$Var(X) = \frac{s^2\pi^2}{3}$$

that

$$s = \sqrt{\frac{3 \cdot Var(X)}{\pi^2}}$$

so a plug-in estimate would be

$$\hat{s}_{plugin} = \sqrt{\frac{3 \cdot S_n^2}{\pi^2}},$$

where S_n^2 is the sample variance. Using this on our sample gives:

```
plug_in_estimates <- function(x) {
  c(mhat = mean(x), shat = sqrt(3*var(x)/pi^2))
}
plug_in_estimates(x)
```

```
##     mhat     shat
## 1.556956 2.399929
```

```r
llh <- function(par, x) {
  -sum(dlogis(x, location = par[1], scale = par[2], log = TRUE))
}

optim(par = plug_in_estimates(x), fn = llh, x = x, method = "BFGS")
```

```
## $par
##     mhat     shat
## 1.557601 2.512727
```

Example 2

Numerical Calculation of a Maximum Likelihood Estimate

Consider the continuous distribution with density:

$f_\theta(x) = \begin{cases} \theta x^{\theta-1} & x \in (0,1) \\ 0 & otherwise \end{cases}$

for $\theta \in (0, \infty)$.

A sample of size 250 from this distribution (with unknown θ) is available here:sample_dist.rda

Calculate the Maximum Likelihood Estimate of θ numerically using *"optim()"* with method equal to *"BFGS"*.

Hint: Ignore possible warnings about generated NaNs

Enter the MLE of θ with a precision of at least 3 decimal places.

```r
# question 9 - version 2 ----
n = 250
m_true = 3
s_true = 1

y <- rlogis(n = n, location = m_true , scale = s_true)

#starting values estimates
sample_mean <- mean(y) # m as expected value from the exercise
sample_s <- sqrt(3)/pi*sd(y) #rearranging from variance from exercise

starting_value <- c(sample_mean,sample_s)
starting_value

minus_llh <- function(par, y){
  -sum(dlogis(x = y, location = par[1], scale = par[2], log = TRUE))
}

optim(par = starting_value, fn = minus_llh, method = "BFGS", y = y)
```

a) What is the Maximum Likelihood estimate?

```r
mu_est_mle <- mean(x)
mu_est_mle
```

Example 3

Exercise 1: Maximum Likelihood Estimation

(3+3=6 Points)

Consider a continuous distribution with parameter $\theta > 0$. The density is given by

$$f_\theta(x) = \begin{cases} 2\theta^2 x e^{-(\theta x)^2}, & x > 0 \\ 0, & \text{otherwise} \end{cases}$$

on $(0, \infty)$.

a) Write an R function `llh <- function(theta, x){...}` that calculates the log-likelihood for a sample and **one** parameter value θ. The function should have the following two arguments:

argument `theta`: parameter value (scalar)
argument `x`: Values of the sample (vector)

Give the R function and the value of the log-likelihood for the sample `d <- c(5,4,3,2,5,4,2,6,2,1,1)` and $\theta = 0.2$.

```r
llh <- function(theta, x){sum(log( 2*theta^2*x*exp(-(theta*x)^2) )) }

d <- c(5,4,3,2,5,4,2,6,2,1,1)
llh(theta = 0.2, x = d)
```

```
## [1] -22.46174
```

b) Using R, determine the Maximum Likelihood Estimate of θ for the sample `d <- c(5,4,3,2,5,4,2,6,2,1,1)`. Also give the R code.

```r
minus_llh <- function(theta, x) {
  -llh(theta, x)
}

optim( par = 1,
       fn = minus_llh,
       method = 'BFGS',
       x = d)
```

```
## $par
## [1] 0.2793113
##
## $value
## [1] 20.47361
##
## $counts
## function gradient
##       20        7
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Rebecca Nauli

Bayes Estimators

Idea: Interpret parameter θ as a random variable with a known distribution, the so-called prior. This should reflect any previous knowledge on the parameter.

Calculate the **posterior distribution**, i.e. distribution of parameter given the data:

$$h(\theta|X_1,\dots,X_n) = \frac{f\,(X_1,\dots,X_n|\theta)\cdot g(\theta)}{f\,(X_1,\dots,X_n|\tilde{\theta})\cdot\, g(\tilde{\theta})d\tilde{\theta}}$$

The only case where we get probabilities for the parameter! Analytically tractable only for a few special pairs of prior and data distribution. Usually solved numerically using Markov Chain Monte Carlo.

With some lengthy manipulations, we obtain:

$$h(\mu \mid x_1,\dots,x_n) = \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left(-\frac{(\mu-\mu_1)^2}{2\sigma_1^2}\right)$$

with

$$\mu_1 = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}}\overline{x} + \frac{\frac{1}{\sigma_0^2}}{\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}}\mu_0$$

and

$$\sigma_1^2 = \frac{1}{\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}}.$$

The **posterior distribution** is again a **normal distribution**, with parameters that depend on

- μ_0 and σ_0^2 from the **prior**
- σ^2 (known) and
- the **sample** via $\overline{x}$

Beta-distributed a = shape1, b = shape2
dbeta(), pbeta(), qbeta(), rbeta()

Bayes estimators for the binomial: If $X \sim Bin(n,p)$ with known n and

$$p \sim Beta(a,b)$$ prior

(prior), then the **posterior distribution** of p given x is a $Beta(a',b')$-distribution with parameters

$$a' = a + x$$
$$b' = b + n - x.$$

The **posterior expectation** is then:

$$\frac{a'}{a'+b'} = \frac{a+x}{a+b+n} = \frac{a+b}{a+b+n}\left(\frac{a}{a+b}\right) + \frac{n}{a+b+n}\left(\frac{x}{n}\right),$$

i.e. a **weighted mean** of the **prior expectation** and the **proportion of successes in the sample**.

Example 1

```
n <- length(x)
mu0 <- 9.8 # prior mean
sigma0 <- 0.25 # prior sd
```

```
# Likelihood (depends on data, but not prior information)

sigma <- 0.2

likelihood <- function(mu) {
  prod(dnorm(x, mean = mu, sd = sigma))
}

lines(xgrid, sapply(xgrid, likelihood)/30, col = "red")

# Posterior (combines data and prior)

# posterior mu and sigma from the lectures
mu1 <- ((n / sigma ^ 2) * mu_est_mle +
          (1 / sigma0 ^ 2) * mu0) / (n / sigma ^ 2 + 1 / sigma0 ^ 2)
sigma1 <- sqrt(1 / (n / sigma ^ 2 + 1 / sigma0 ^ 2))

mu1
```

Rebecca Nauli

Confidence Intervals

Confidence intervals: an interval of plausible values for one or more unknown parameter is wanted. The interval should be as short as possible and should contain the **unknown parameter with high certainty**. → shorter intervals = more informative
gives not only a single point estimate but a whole range of values of the parameter which would be compatible with the data, Calculation of plausible values for parameters

Note:

- *This is why all other factors being the same, a 95% confidence interval will be wider than a 90% confidence interval. (more certainty = wider range)*
- *A 90% confidence interval is shorter than a 95% confidence interval calculated from the same data → TRUE*

Mathematically: given an iid sample $X_1, \dots, X_n$ from a distribution with some parameter θ, we want a lower bound $\hat{\theta}_{lower}(X_1, \dots, X_n)$ and an upper bound $\hat{\theta}_{upper}(X_1, \dots, X_n)$, such that for the interval

$$[\hat{\theta}_{lower}(X_1, \dots, X_n), \hat{\theta}_{upper}(X_1, \dots, X_n)]$$

we have

$$P(\hat{\theta}_{lower}(X_1, \dots, X_n) \leq \theta \leq \hat{\theta}_{upper}(X_1, \dots, X_n)) \geq 1 - \alpha$$

$1 - \alpha$ is called the coverage probability, or confidence level.

Confidence intervals for $expectation\ \mu\ of\ normal\ distribution$:

- *σ known:*

$$\left[\bar{x} - \frac{\sigma}{\sqrt{n}} q_{1-\frac{\alpha}{2}}, \bar{x} + \frac{\sigma}{\sqrt{n}} q_{1-\frac{\alpha}{2}}\right]$$

- *σ unknown:*

$$[\overline{X_n} - \frac{S_n}{\sqrt{n}} t_{n-1;1-\frac{\alpha}{2}}, \overline{X_n} + \frac{S_n}{\sqrt{n}} t_{n-1;1-\frac{\alpha}{2}}]$$

Length of CI:

$$2\frac{\sigma}{\sqrt{n}} q_{1-\frac{\alpha}{2}} = length$$

$$\Leftrightarrow n \geq \left(2\frac{\sigma}{length} q_{1-\frac{\alpha}{2}}\right)^2 \ for\ sample\ size$$

```
(2 * 0.02 / 0.001 * qnorm(0.95))^2
```

Also asymptotically valid for expectation of other distributions by using the Central Limit Theorem.

rejection region: 0.05% significance level, 95 quantile
acceptance region: >0.05% significance level

Sample arithmetic mean $\overline{X_n}$

$$\overline{X_n} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

standardised rv

$$Z_n = \sqrt{n}\frac{X_n - \mu}{\sigma} \Leftrightarrow Z_n \sim N(0,1)$$

Sample standard deviation (SD)

Sample SD

```
sds <- apply(x, MARGIN = 1, FUN = sd)
```

Student's t-distribution

dt(), pt(), qt(), rt() *df = degrees of freedom*

```
dd <- seq(-7, 7, 0.1)
lines(dd, dt(dd, df = 2), col = "green")
```

Example 1

Confidence Intervals

Given below are three confidence intervals for a parameter of an unknown distribution, all based on the same sample.

Match the intervals a,b,c to the 60%-, 95%- and 99% confidence levels.

Confidence interval a: [100, 386]
99% ✓

Confidence interval b: [196, 293]
60% ✓

Confidence interval c: [138, 350]
95% ✓

Example 2

c) A company would like to collect the average time needed for a certain assembly step. For this purpose, the times that 40 assemblers needed for this step were stopped. This resulted in an arithmetic mean of 12.73 min and an estimated sample standard deviation of 2.06 min. Calculate an (approximate) 99% confidence interval for the mean time required.

We calculate the CI based on the t distribution, i.e.

$$\left[\overline{x} \pm \frac{s_n}{\sqrt{n}} t_{n-1;1-\alpha/2}\right].$$

Alternatively, the normal quantiles may also be used.

```
12.73 + c(-1,1) * 2.06 / sqrt(40) * qt(0.995, df = 39)
```

```
## [1] 11.84799 13.61201
```

```
12.73 + c(-1,1) * 2.06 / sqrt(40) * qnorm(0.995)
```

```
## [1] 11.89101 13.56899
```

Example 3

Metal pin length

What is the average length of metal pins?
We want to estimate these with the arithmetic Mean. A sample of size 39 yields a mean length (arithmetic mean) of $\bar{x} = 38.5mm$.
It is known from previous studies that the length of the metal pins is normally distributed and that the producing machine operates with a known standard deviation of $\sigma = 1.6mm$.

Answer the questions below:

a. Give the lower bound of the 95% confidence interval for the expected metal pin length. Do NOT solve this problem with a bootstrap confidence interval. Give the result with at least 3 decimal places: 37.998 ✓

b. What is the minimum size of a sample that the 95% confidence interval for the mean pencil length is at most half as wide: 156 ✓

c. Given the following confidence interval: [38.205 mm; 38.795 mm]. What is the confidence interval? Select the percentage: 75 ✓

```r
# Given data
x_bar <- 38.5       # sample mean
sigma <- 1.6        # population standard deviation
n <- 39             # sample size

# Part (a) - Lower bound of the 95% confidence interval
# Calculate margin of error for 95% confidence level without using z-score explicitly
margin_error_95 <- qnorm((1 + 0.95) / 2) * (sigma / sqrt(n))
lower_bound_95 <- x_bar - margin_error_95
cat("Lower bound of the 95% confidence interval:", round(lower_bound_95, 3), "mm\n")

# Part (b) - Minimum sample size for the confidence interval to be half as wide
# Calculate the new margin of error as half of the original margin of error
new_margin_error <- margin_error_95 / 2

# Calculate the required sample size
required_n <- ((qnorm((1 + 0.95) / 2) * sigma) / new_margin_error) ^ 2
cat("Required sample size for half-width confidence interval:", ceiling(required_n), "\n")

# Part (c) - Confidence level for the interval [38.205, 38.795]
lower_bound <- 38.205
upper_bound <- 38.795

# Calculate the margin of error for the given interval
given_margin_error <- (upper_bound - lower_bound) / 2

# Find the confidence level by determining the probability that corresponds to the calculated z-score
z_given <- given_margin_error / (sigma / sqrt(n))
confidence_level <- 2 * pnorm(z_given) - 1
confidence_level_percentage <- confidence_level * 100
cat("Confidence level for the interval [38.205, 38.795]:", round(confidence_level_percentage, 2), "%\n")
```

Test used for CI calculations

z.test()
t.test()

Confidence Interval for Expected Length of Screws (Calculation in R)

Given is a sample of screws produced on a particular day. The sample size is **148** and the **screws' lengths** (in mm) are available in the file `screwlength.rda`. Determine the 90% confidence interval of the expected screw length. In addition, calculate point estimates of the expected value and the standard deviation of the screw length.

(Note: Do not use the bootstrap here!)

The length of the screws is given here: screwlength.rda

a. Enter the lower and upper bound of the confidence interval 90% confidence interval of the expected length. Give the result with at least 3 decimal places.
Lower endpoint: 49.975 ✓ Upper endpoint: 50.037 ✓

b. Enter your point estimates of the expected value and the standard deviation of the screw length. Give the result with at least 3 decimal places.
Estimator for expected value: 50.006 ✓ Estimated standard deviation: 0.229 ✓

```r
t.test(screwlength, alternative = "two.sided", conf.level = 0.90)
```

binom.test()
poisson.test()

b) In the last 5 years, in a certain area the number of earthquakes of a certain strengt was 55, 72, 64, 73, and 57, respectively. Calculate a 90% confidence interval for the mean number of earthquakes per year.

```r
x <- c(55, 72, 64, 73, 57)
poisson.test(x = sum(x), T = 5, conf.level = 0.9)
```

```
##
##  Exact Poisson test
##
## data:  sum(x) time base: 5
## number of events = 321, time base = 5, p-value < 2.2e-16
## alternative hypothesis: true event rate is not equal to 1
## 90 percent confidence interval:
##  58.42188 70.41467
## sample estimates:
## event rate
##       64.2
```

Bootstrap Confidence Intervals

Bootstrap: We use the Bootstrap to obtain confidence intervals without distribution assumption.

Idea: For a parameter estimated by a statistic $T(X_1, \dots, X_n)$ calculate T a large number of times on bootstrap samples drawn **with replacement** from the original sample.

Use quantiles of the empirical distribution of $T^*(x^1), \dots, T^*(x^B)$ as boundaries for the confidence interval.

More refined versions are available, e.g. the Bca variant.

library(boot)

boot(data =, statistic = , R =) *R = number of bootstrap replicates*

```
library(boot)
statistic <- function(dat, ind) cor(dat[ind, "A"], dat[ind, "B"])
boot_res <- boot(data = spatial, statistic = statistic, R = 1000)
hist(boot_res$t)
```

```
boot_res <- boot(galaxies, function(z, ind) median(z[ind]), R = 5000)
boot_res
```

boot.ci() *"perc" = percentile, "bca" = adjusted bootstrap percentile (Bca) method*

```
boot.ci(boot_res, type = c("perc", "bca"))
```

Example 1

Question 3
Not yet answered
Marked out of 2.00
Flag question

Confidence interval of a quantile of repair times (bootstrap simulation with R)

In a car repair shop, a 90% confidence interval is to be calculated for the 80% quantile of the time required for a transmission repair. Only the last 55 repairs are available.

The sample of repair times can be found here: repairtimes.rda

Set the RNG to 100 (*"set.seed(100)"*) before running your simulation. Use either "perc" or "bca".

Report the lower and upper bounds of the 90% confidence interval of the 80% quantile of the duration of repair times using 100'000 bootstrap samples. Give the result with at least 3 decimal places.

Lower bound: Upper bound:

```
# question 3 ----
set.seed(100) # to get the same result

library("boot") # bootstrap library

# statistic - here 80th percentile - this changes depending
Tstat <-  function(x, ind) {
  quantile(x[ind], 0.8)
}

#bootstrapping with 100'000 bootstrap samples
boot_res <- boot(data = repairtimes, statistic = Tstat, R = 100000)

# 90% CI for 80% quantile
boot.ci(boot_res, conf = 0.90, type = c("perc","bca"))
```

Example 2

Exercise 5: Bootstrap

$2 + 3 = 5$ Points

In many technical and scientific applications, the *coefficient of variation* is of interest. It is defined as the ratio of the standard deviation of a random variable divided by the expectation:

$$CV_X = \frac{\sigma_X}{E(X)}$$

This quantity is of course only defined for non-negative random variables (e.g., measurements of length, weight etc.) and relates scatter to expectation.

The coefficient of variation can be estimated using

$$T(x_1, \dots, x_n) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}}{\bar{x}}$$

which corresponds to the ratio of sample standard deviation and arithmetic mean, except that we divide by n and not by $n - 1$ when calculating the standard deviation.

Generate a sample as follows:

```
set.seed(20200123)
x <- rlnorm(20, meanlog = 1, sdlog = sqrt(log(2)))
```

a) Calculate the estimator T of the coefficient of variation for this sample.

```
T_stat <- sqrt(sum((x-mean(x))^2)/length(x))/mean(x)
T_stat
```

```
## [1] 0.9313037
```

b) Calculate a 90% bootstrap confidence interval for the coefficient of variation. Indicate how you chose the number of bootstrap samples.

```r
# BCa- and percentile interval using boot:

library(boot)

stat <- function(y, ind) {
  sqrt(sum((y[ind]-mean(y[ind]))^2)/length(y[ind]))/mean(y[ind])
}

boot.res <- boot(data = x, statistic = stat, R = 1000)
boot.res
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = x, statistic = stat, R = 1000)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 0.9313037 -0.03271554   0.1089688
```

```r
boot.ci(boot.res, conf = 0.9, type = c("perc", "bca"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.res, conf = 0.9, type = c("perc", "bca"))
##
## Intervals :
## Level     Percentile            BCa
## 90%   ( 0.7292,  1.0762 )   ( 0.7889,  1.1388 )
## Calculations and Intervals on Original Scale
```

```r
# percentile interval without boot

stat2 <- function(y) {
  sqrt(sum((y-mean(y))^2)/length(y))/mean(y)
}

boot.res2 <- replicate(1000, stat2(sample(x, size = length(x), replace = TRUE)))
quantile(boot.res2, probs = c(0.05, 0.95))
```

```
##        5%       95%
## 0.7395813 1.0675444
```

Rebecca Nauli

Statistical Hypothesis Testing

Hypothesis testing: verify statements about unknown parameters, e.g. whether an expected value is greater than 0 or whether the expected value is greater in one population than in another. → statistical significance

General Terminology

Test, whether certain values of parameters are compatible with the data

Generic recipe:

- *Determine the null hypothesis H0*
- *Determine the alternative hypothesis H1*
- *Determine the test statistic T and its distribution*
- *(Optionally) determine rejection/acceptance region*
- *Observe the realized value of the test statistic*
- *Calculate p-value*
- *Making a decision: reject H0 if p-value ≤ α or equivalently T takes a value in the rejection region.*

H_0: *Null hypothesis*
H_1: *Alternative hypothesis*

If p-value > α → H_0 *is not rejected*
If p-value < α → H_0 *is rejected*

$\alpha =$ 5% *or* 1% *significance level*

Note:

- *A large sample size n helps, when an existing effect should be significant in a statistical test. → correct*
- *If the result of a test is not significant, one may conclude that it is statistically proven that the effect is non-existent. → wrong*
- *5000 independent tests are conducted at a small significance level of α = 0.1%. Hence, we can conclude with great certainty that there is a real effect if at least one of the tests gives a significant result. → wrong*
- *When performing a t-test at 5% significance level, how does the probability of a Type I error change if the sample size is increased? → stays the same*
- *A p-value of 0.01 in a hypothesis test means that the probability of the null hypothesis being true is only 1%. → FALSE*
- *You perform a t-test of at 5% level. If you do not reject the null, you proved that with probability 95%, . H0 : μ = 0 μ = 0 → FALSE*

Power:

power.t.test() *1 parameter can be unknown and is called as NULL*

```
power.t.test(n = NULL, delta = 1, sd = 1, sig.level = 0.05, power = 0.99,
             alternative="one.sided", type = "one.sample")
```

Note:

- *If the variance stays the same, increasing the sample size n increases the power of the t-test*
- *Two-sided tests have a lower power than one-sided tests.*

One sample test

z-Test

to check whether the **expectation μ of a distribution is equal to a given value**.
Assumption: variance σ^2 was known

$$Z = \frac{\bar{X} - \mu_0}{\frac{\sigma}{\sqrt{n}}} = \sqrt{n} \cdot \frac{\overline{X} - \mu_0}{\sigma} \dot{\sim} N(0,1)$$

library(BSDA)
z.test() *inputs: sample,* σ, μ_0 *from* H_0 *and* H_1

```
library(BSDA)
x <- rnorm(20, mean = 0.5, sd = 1)
z.test(x = x, sigma.x = 1,
       alternative = "two.sided", conf.level = 0.95)
```

library(BSDA)
z.sum.test *input: arithmetic mean & σ*

```
zsum.test(mean.x = mean(x), sigma.x = 1, n.x = length(x),
          alternative = "two.sided", conf.level = 0.95)
```

t-Test

unknown σ substitute for sample standard deviation → estimate of **σ**

$$\widehat{\sigma} = s_n = \sqrt{\frac{1}{n-1}\sum (x_i - \overline{x})^2}$$

Test statistic T: for normal distribution → t-distributed with n − 1 degrees of freedom. For other distributions → approximately in so far as the CLT is applicable the sample is not to small & the distribution is not too skewed or too heavy-tailed

$$T = \frac{\overline{X} - \mu_0}{\frac{\widehat{\sigma}}{\sqrt{n}}} \sim t_{n-1}$$

t-distribution: somewhat wider than the normal distribution

The width of the region of acceptance now depends on the sample size. → The region of acceptance becomes narrower as the sample size increases.

library(BSDA)

t.test() *Performs one and two sample t-tests on vectors of data.*

```
t.test(fuel, mu = 8.2, alternative = "two.sided")
```

library(BSDA)

tsum.test() *only if arithmetic mean, sample SD & the sample size*

```
tsum.test(mean.x = 148, s.x = 3, n.x = 50, mu = 150,
          alternative = "less")
```

pt() *one-sided t-test, used when sample, test statistic t & α is known*

```
pt(-2.3, df = 17)
```

Note: For a sample of size 10, the t-test should only be applied if the observations come from a normal distribution. → Correct

Two-sample Test

Paired vs Unpaired

Paired: both observations were taken on the same experimental unit (or on very similar units), Perform t-test on differences in pairs.

- *before and after receiving a treatment,*
- *responses of patient i to two different treatments*
- *e.g. platelets accumulation in smokers with before & after measurements*
- *e.g. 2 tire profiles breaking test on the same 10 vehicles*
- *e.g. 15 seedlings growth height in nearly identical plants*

library(BSDA)

*t.test(x, y, paired = **TRUE**, ...)* *for paired test set paired to TRUE*

```
t.test(os$Linux, os$Windows, paired = TRUE, mu = 0,
       alternative = "two.sided")
t.test(neck$Before, neck$After, paired = TRUE)
```

Unpaired: two samples from different populations, not have to have the same sample size,

- *two groups with different conditions*
- *e.g. two forms of iron preparation divided for two groups of mice*

library(BSDA)

*t.test(x, y, paired = **FALSE**,...)* *for unpaired test set paired to FALSE*

```
# Supply the samples as x and y
t.test(x = blood_pressure$bp_decrease[blood_pressure$group == "treatment"],
       y = blood_pressure$bp_decrease[blood_pressure$group == "control"],
       paired = FALSE, alternative = "greater")
```

```
t.test(Score ~ Treatment, data = case0101,
       var.equal = FALSE, mu = 0, alternative = "two.sided")
```

Welch test

allows for different values of σ in the two populations

Note: var.equal = FALSE is the Welch Test (does not assume that the variances are equal)

```
# use formula notation
# the x sample is factor level 1 (treatment), y is factor level 2 (control)
levels(blood_pressure$group)
```

```
## [1] "treatment" "control"
```

```
t.test(bp_decrease ~ group, data = blood_pressure,
       alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  bp_decrease by group
## t = 1.6037, df = 15.591, p-value = 0.06442
## alternative hypothesis: true difference in means between group treatment and group c
## 95 percent confidence interval:
##  -0.476678       Inf
## sample estimates:
## mean in group treatment   mean in group control
##               5.0000000              -0.2727273
```

Note: if data is normally distributed use t-test if not you can use the Wilcoxon Rank-Sum Test (check with qqplot)

Other tests

Binomial Test

binom.test() — *Performs an exact test of a simple null hypothesis about the probability of success in a Bernoulli experiment.*
known X, n, p (probability) or CI level,, H_0 & H_1

```
binom.test(x = 15, n = 50, p = 1/2, alternative = "less")
```

```
qbinom(0.95, size = 20, prob = 0.5)

## [1] 14
sum(dbinom(14:20, size = 20, prob = 0.5))

## [1] 0.05765915
sum(dbinom(15:20, size = 20, prob = 0.5))

## [1] 0.02069473
```

Poisson Test

poisson.test() — *Performs an exact test of a simple null hypothesis about the rate parameter in Poisson distribution, or for the ratio between two rate parameters.*

```
load(file.path(baseDir, "insulate.RData"))
# Add up the flaws per vendor
counts <- tapply(insulate$flaws, insulate$supplier, sum)

# 5 poisson tests:
poistst <- function(x) poisson.test(x, T = 10, r = 7, alternative = "less")$p.value
pvals <- sapply(counts, poistst)
round(pvals, digits = 5)
```

Nonparametric Tests

Sign Test

- *for the median of a distribution.*
- *based on the signs of $X_i - m_0$, where m_0 is the hypothetical median from H_0.*

SIGN.test() — *This function will test a hypothesis based on the sign test and reports linearly interpolated confidence intervals for one sample problems.*

Wilcoxon Rank-Sum Test

- *t is a nonparametric counterpart to the **unpaired two-sample t-test.***
- *Ranks are calculated for combined sample and added up for one of the samples.*

*wilcox.test(..., **paired = FALSE**)* — *Performs one- and two-sample Wilcoxon tests on vectors of data; the latter is also known as 'Mann-Whitney' test.*

```
wilcox.test(Sand1, Sand2, alternative = "two.sided")

##
##  Wilcoxon rank sum exact test
##
## data:  Sand1 and Sand2
## W = 42, p-value = 0.005062
## alternative hypothesis: true location shift is not equal to 0
```

Coin version
library(coin)
wilcox_test()

```
library(coin)
dat <- data.frame(wt = c(ext, int),
                  feed = factor(c(rep("ext", length(ext)),
                                  rep("int", length(int))))
                  )

wilcox_test(wt ~ feed, data = dat,
            distribution = "exact")

##
##  Exact Wilcoxon-Mann-Whitney Test
##
## data:  wt by feed (ext, int)
## Z = -3.1279, p-value = 0.0009101
## alternative hypothesis: true mu is not equal to 0
```

Note: if data is normally distributed use t-test if not you can use the Wilcoxon Rank-Sum Test (check with qqplot)

Wilcoxon Signed Rank Test

- *is a nonparametric counterpart to the **paired two sample t-test***
- *Ranks of absolute values of differences are calculated and added up for positive sign.*

*wilcox.test(..., **paired = TRUE**)*

```
wilcox.test(tires$Profile_A, tires$Profile_B, paired = TRUE)

##
##  Wilcoxon signed rank exact test
##
## data:  tires$Profile_A and tires$Profile_B
```

Coin version
wilcoxsign_test()

```
library(coin)

wilcoxsign_test(tires$Profile_A ~ tires$Profile_B,
            alternative = "two.sided",
            distribution = "exact")

##
##  Exact Wilcoxon-Pratt Signed-Rank Test
##
## data:  y by x (pos, neg)
##   stratified by block
## Z = -2.2934, p-value = 0.01953
## alternative hypothesis: true mu is not equal to 0
```

Multiple Testing

t-test for each possible pair:

pairwise.t.test() *Calculate pairwise comparisons between group levels with corrections for multiple testing*

```
pairwise.t.test(pulp$bright, pulp$operator,
                p.adjust.method = "none", pool.sd = F)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  pulp$bright and pulp$operator
##
##   a      b      c
## b 0.5088 -      -
## c 0.1882 0.0055 -
## d 0.1360 0.0028 0.6811
##
## P value adjustment method: none
```

Significant: Differences between B – C and B – D.

Bonferroni correction

```
pairwise.t.test(pulp$bright, pulp$operator,
                p.adjust.method = "bonf", pool.sd = F)
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  pulp$bright and pulp$operator
##
##   a     b     c
## b 1.000 -     -
## c 1.000 0.033 -
## d 0.816 0.017 1.000
##
## P value adjustment method: bonferroni
```

In this case, the differences are still significant (but with higher p-values).

p.adjust() *Given a set of p-values, returns p-values adjusted using one of several methods.*

Chi-Square Test (x^2 – *Test for Contingency Tables*)

Test of Independence

Each cell contains the number of observations with the particular combination of values given by row and column.

chisq.test()

Example 1

```
pollutants_test <- chisq.test(pollutants)
pollutants_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  pollutants
## X-squared = 125.01, df = 12, p-value < 2.2e-16
```

Example 2

Exercise 4: Tests and confidence intervals

(2 + 2 + 2 = 6 Points)

In the questions regarding hypothesis tests, give the null and alternative hypotheses, the test used, the R code and the p-value and the test decision.

All questions are independent!

a) A survey among students in Zurich and Lausanne yielded the following distribution of subjects:

	Mathematics	Engineering	Chemistry	Economics	other	Total
Zurich	**95**	**300**	**160**	**250**	**320**	1125
Lausanne	**75**	**200**	**100**	**230**	**270**	875
Total	170	500	260	480	590	2000

Use a suitable test at a significance level of 5% to check whether the distributions of subjects are different between the students in both cities.
H0: Subject and city are independet or the distributions of the subject are the same for both cities.
We conduct a χ^2-test and get a p-value of 0.03752, so there is a significant difference.

```
tab<-rbind(c(95,300,160,250,320),
c(75,200,100,230,270))

chisq.test(tab)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab
## X-squared = 10.179, df = 4, p-value = 0.03752
```

```
mat <- rbind(c(55, 95), c(36, 122))
chisq.test(mat)
```

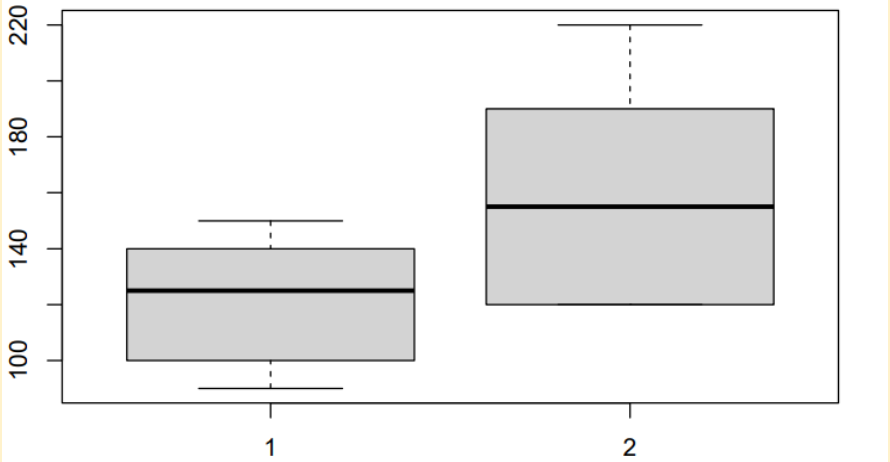
Test of Equal or Given Proportions

prop.test() *can be used for testing the null that the proportions (probabilities of success) in several groups are the same, or that they equal certain given values.)*

Visualisations

boxplot()

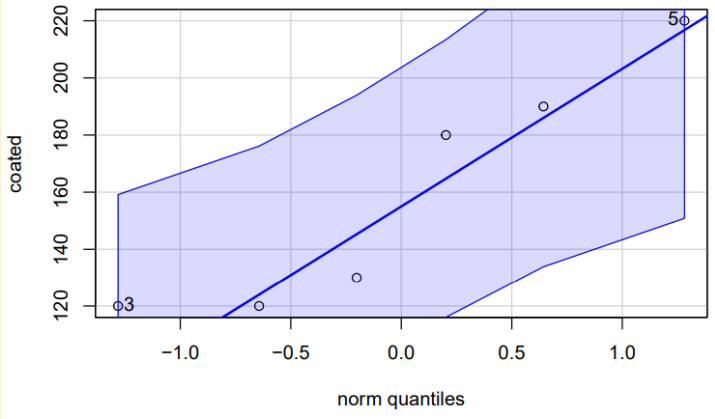
```
uncoated <- c(120, 130, 150, 90, 100, 140)
coated <- c(180, 130, 120, 190, 220, 120)
boxplot(uncoated, coated)
```



library(car)

qqPlot() or qqplot() → *second not from car library*

```
library(car)
qqPlot(coated)
```




```
## [1] 5 3
```

hist()

Rebecca Nauli

Repetition – Probability theory

Basics

- *A **random experiment** is an experiment or a situation in which the result is not predetermined.*
- *A repetition does usually not yield the same result.*
- *There is exactly one outcome and the different outcomes are mutually exclusive.*

Sample space:	the set of all possible outcomes, usually denoted by the symbol Ω
Events:	subsets of Ω
Random variable:	a variable which can take different values by chance under certain constant conditions. It is part of the sample space. $X : \Omega \to \mathbb{R}$

Discrete Case

Discrete variables:	part of the random variables, countable, only $\mathbb{N}$, Examples: humans, vehicles, infections, defects (you cannot have half a human, etc.)
Probability distribution	the distribution of a rv which indicated which values the rv takes with which probability.
Realizations:	all random variables from $X : x_1, x_2, x_3, x_4, \dots$
Probabilities:	corresponding to realisations of X, $p_1, p_2, p_3, p_4, \dots$
Probability function:	distribution of all probabilities of all random variables in a sample space. E.g. Normal distribution, binomial, etc.

- *NOTE: only for **discrete** rv! Otherwise use density function.*

Probability function: $p_i = p(x_i) = P(X = x_i)$

Probability of an event = 1 (ALWAYS!!)

$$\sum_i p_i = 1$$

Continuous Case

Continuous variables:	part of random variables, interval, possibly unbounded (±∞) Examples: temperature, weight, length, etc.)
PDF:	Probability **density function,** $f(x)$ = probability function in discrete case. not possible to assign probabilities to single value but interval (**continuous variables**).

1. $f(x) \geq 0, for\ all\ x \in \mathbb{R}$
2. f *is piecewise continuous*
3. $\int_{-\infty}^{\infty} f(x)dx = 1$ → *all probabilities are = 1*

$$P(a < X \leq b) = \int_a^b f(x)dx$$

Distribution function

CDF: The **cumulative distribution function (CDF)** calculates the cumulative probability for a given x-value. Use the CDF to determine the probability that a random observation that is taken from the population will be less than or equal to a certain value.

$$F(x) = P(X \leq x)$$

$CDF\ for\ discrete$:

$$\mathrm{F(x)} = \sum_{\mathrm{z \leq x}} \mathrm{p(z)}$$

$CDF\ for\ continuous$:

$$\mathrm{F(X)} = \int_{-\infty}^{\mathrm{x}} \mathrm{f(z)dz}$$

integrate()

Expected Value

Expected value: $E(X)$ of a rv is what one obtains on average with an infinite number of realizations.

$\boldsymbol{E(X)\ for\ discrete}\ rv$:

$$E(X) = \sum_i x_i P(X = x_i) = \sum_i x_i p(x_i) = \sum_i x_i p_i$$

$\boldsymbol{E(X) for\ continous}\ rv$:

$$E(X) = \int_{-\infty}^{\infty} x \cdot f(x)dx$$

Rules for transformations:

1. $Y = aX + b\ , with\ a, b \in \mathbb{R}$
- $E(Y) = E(aX + b) = a \cdot E(X) + b$
2. $E(X + Z) = E(X) + E(Z)$

Variance

Variance: $Var(X) = \sigma^2$ is a measure of dispersion of a random variable

Var*(X) *for discrete *rv with* $\mu = E(X)$:

$$Var(X) = E((X - E(X)^2) = E((X - \mu)^2) = \sum_i (x_i - \mu)^2 p(x_i)$$

Var*(X) *for continous *rv with* $\mu = E(X)$:

$$Var(X) = E((X - E(X)^2) = E((X - \mu)^2) = \int_{-\infty}^{\infty} (x - \mu)^2 \cdot f(x)dx$$

Simple: $Var(X) = E(X^2) - \big(E(X)\big)^2$

Discrete cases:

$$E(X^2) = \sum_i x_i^2 p(x_i)$$

Continous cases:

$$E(X^2) = \int_{-\infty}^{\infty} x^2 \cdot f(x)dx$$

Linear Transformation

1. $Y = aX + b$, *with* $a, b \in \mathbb{R}$

- $Var(Y) = Var(aX + b) = a^2 \cdot Var(X)$

2. $Var(X + Z) = E(X) + E(Z) + 2Cov(X, Z)$

Covariance

Covariance: of two random variables is a measure of linear dependence and defined as follows:

$$Cov(X, Y) = E((X - E(X))(Y - E(Y))$$

If $Cov(X, Z) = 0$ (always the case when X & Z independent:

$$Var(X + Z) = Var(X) + Var(Z)$$

Standard deviation

Standard deviation of X is the square root of the variance

$$sd_x = \sqrt{(Var(X)} = \sigma$$

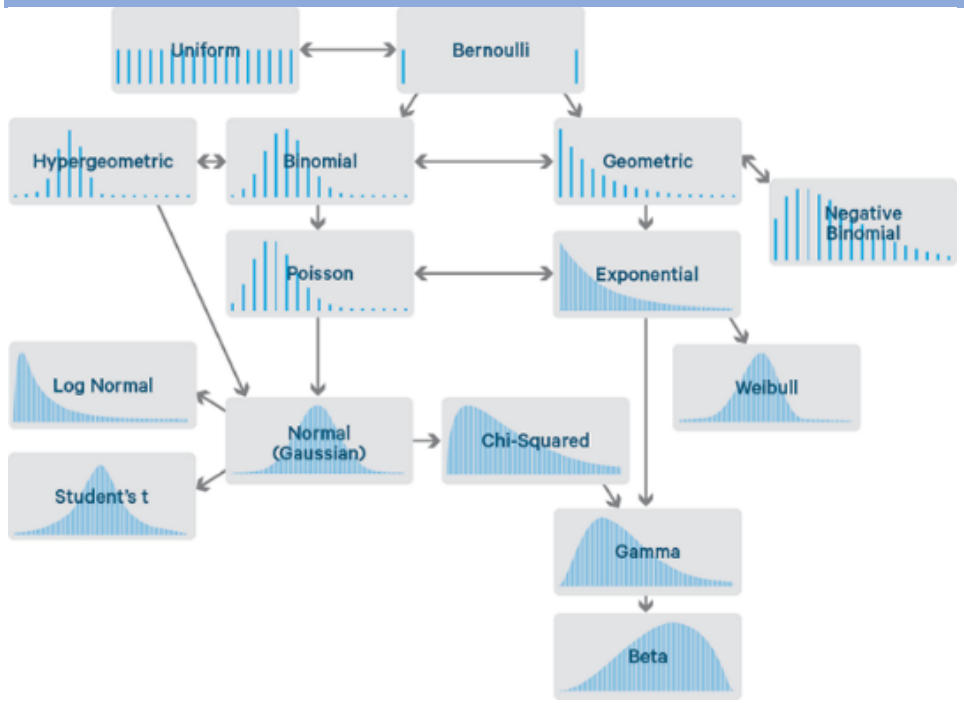
Important Distributions - Summary

d: discrete
c: continous

Notation	Name	d/c	Parameter	$E(X)$	$Var(X)$
$Bin(n, p)$	Binomial	d	$n \in \mathbb{N},\ p \in (0, 1)$	np	$np(1-p)$
$Poisson(\lambda)$	Poisson	d	$\lambda > 0$	λ	λ
$Geom(p)$	Geom.	d	$p \in (0, 1)$	$\frac{1-p}{p}$	$\frac{1-p}{p^2}$
$NBin(r, p)$	Neg. Bin.	d	$r \in \mathbb{N},\ p \in (0, 1)$	$\frac{r(1-p)}{p}$	$\frac{r(1-p)}{p^2}$
$Hyp(m, n, k)$	Hyperg.	d	$m, n, k \in \mathbb{N}$	$k\frac{m}{m+n}$	$k\frac{m}{m+n}\left(1 - \frac{m}{m+n}\right)\frac{n+m-k}{n+m-1}$
$Exp(\lambda)$	Exp.	c	$\lambda > 0$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
$U([a, b])$	Uniform	c	$a, b \in \mathbb{R},\ a < b$	$\frac{b+a}{2}$	$\frac{(b-a)^2}{12}$
$N(\mu, \sigma^2)$	Normal	c	$\mu \in \mathbb{R},\ \sigma > 0$	μ	σ^2

a = min
b = max

Distributions



Bernoulli Distribution

$$X \sim Bernoulli(p)$$

rv can be two values

$$P(X = 1) = p$$

or

$$P(X = 0) = 1 - p$$

Probability of success:

$$P = P(X = 1)$$

Sample:

$$\Omega = \{0, 1\}$$

Expected value:

$$E[X] = 0 \cdot (1 - p) + 1 \cdot p = p$$
$$E[X^2] = 1^2 \cdot P(X = 1) = 1 \cdot p = p$$

Variance:

$$Var[X] = E[X^2] - E[X]^2 = (0 - p)^2(1 - p) + (1 + p)^2 \cdot p = p(1 - p) = p \cdot q$$

Standard deviation:

Binominal Distribution

$$X \sim B(n, p)$$
$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k},$$

Sample:

$$\Omega = \{0, 1, 2, \dots, n\}(discrete)$$

Expected value:

$$\mu = E[X] = E(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} E(x_i) = \sum_{i=1}^{n} p = n \cdot p$$

Variance:

$$\sigma^2 = V[X] = Var(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} Var(x_i) = n \cdot p(1 - p)$$

Standard deviation:

$$\sigma = S(X) = \sqrt{npq}$$

Symmetrie für $p = 0.5$ bei wachsenden $p \neq 0.5$ immer symmetrisch falls $n \cdot p(1 - p) > 10$, gilt $Bin(n, p)$ als symmetrisch

rbinom()	*random deviates*
qbinom()	*quantile function*
dbinom()	*density function*
pbinom()	*distribution function*

Poisson Distribution

$$X \sim Pois(\lambda)$$
$$P(X = x) = \frac{\lambda^k \exp(-\lambda)}{x!}$$

Sample:

$$\Omega = \{0, 1, 2, \dots\}(discrete)$$

Expected value:

$$\lambda = E[X]$$ for small λ → stark rechtsschief

Variance:

$$Var[X] = \lambda$$ the larger λ → symmetrie bei $\lambda > 10$

Standard deviation:

$$SD[X] = \sqrt{Var[X]}$$

Geometric Distribution

$$X \sim Ge(p)$$

Expected value:

$$E[X] = \frac{1 - p}{p}$$

Variance:

$$Var[X] = \frac{1 - p}{p^2}$$

Standard deviation:

$$SD[X] = \sqrt{Var[X]} = \frac{\sqrt{1 - p}}{p}$$

Negative Binomial Distribution

$$X \sim NB(r, p)$$

Expected value:

$$E(X) = r\frac{(1 - p)}{p}$$

Variance:

$$V(X) = \frac{r(1 - p)}{p^2}$$

Hypergeometrical Distribution

$$X \sim H(N, M, n)$$

$$P(X = k) = \frac{\binom{M}{k} \cdot \binom{N-M}{n-k}}{\binom{N}{n}}$$

N: Objekte Total
M: Merkmale Total
$N - M$: Objekte anderer Sorte
n: gezogene Objekte ohne Zurücklegen

Sample:

$$\Omega = \{0, 1, \dots, N\}\ (endlich)$$

Expected value:

$$\mu = E[X] = n \cdot \frac{M}{N}$$

Variance:

$$\sigma^2 = V(x) = n \cdot \frac{M}{N} \cdot \left(1 - \frac{M}{N}\right) \cdot \left(\frac{N-n}{N-1}\right)$$

Standard deviation:

$$\sigma = S(X) = \sqrt{n \cdot \frac{M}{N}\left(1 - \frac{M}{N}\right)\left(\frac{N-n}{N-1}\right)} = \sqrt{V(X)}$$

Exponential Distribution

$$X \sim Exp(\lambda)$$

Expected value:

$$E(X) = \frac{1}{\lambda}$$

Variance

$$Var(X) = \frac{1}{\lambda^2}$$

pexp()	*use for CDF continuous*
dexp()	
rexp()	*use for CDF discrete*

Uniform Distribution

$$X \sim U(a, b)$$

Expected value:

$$E(X) = \frac{1}{2}(a + b)$$

Variance:

$$Var(X) = \frac{1}{12}(b - a)^2$$

Normal Distribution

$$X \sim N(\mu, \sigma^2)$$

Expected value:

$$E(X) = \mu$$

Variance:

$$Var(X) = \sigma^2$$

Standard deviation:

$$SD(X) = \sigma$$

pnorm()
dnorm()
rnorm()
qnorm()

Bayes theorem/statistics

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Law of total probability

If $B_1, B_2, B_3, \dots$ is a partition of the sample space S, then for any event A we have

$$P(A) = \sum_i P(A \cap B_i) = \sum_i P(A|B_i) \cdot P(B_i), i = number\ of\ events$$

Conditional probability:

$$P(A) = P(A|B) \cdot P(B) + P(A|B^C) \cdot P(B^C)$$

Multiplication probability rule:

$$P(A \cap B) = P(A|B) \cdot P(B)$$

Source for distributions: https://statproofbook.github.io/l/ToC#Normal%20distribution

Useful R functions

sapply()
apply()
lapply()
tapply()