Solutions for Series 1

1. If the elementary events have equal probability we can calculate the probability of event A as the number of elementary events in A divided by the total number of elementary events.

a) $\Omega = \{(1,1),(1,2),\dots,(1,6),(2,1),(2,2),\dots,(2,6),\dots,(6,6)\}$, $|\Omega| = 36$.

b) $P[\{\text{elementary event}\}] = \frac{1}{|\Omega|} = \frac{1}{36}$.

c) $E_1 = \{(1,6),(2,5),(3,4),(4,3),(5,2),(6,1)\}$;
Number of favourable cases: $|E_1| = 6$;
Number of possible cases: $|\Omega| = 36$;
$P[E_1] = \frac{|E_1|}{|\Omega|} = \frac{6}{36} = \frac{1}{6}$.

d) $E_2 = \{(1,1),(2,1),(1,2)\}$;
$P[E_2] = \frac{|E_2|}{|\Omega|} = \frac{3}{36} = \frac{1}{12}$.

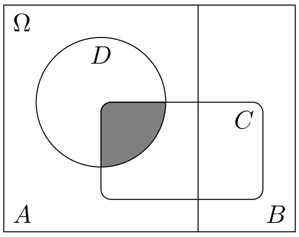
e) $E_3 = \{(1,1),(1,3),(1,5),(3,1),(3,3),(3,5),(5,1),(5,3),(5,5)\}$;
$P[E_3] = \frac{|E_3|}{|\Omega|} = \frac{9}{36} = \frac{1}{4}$.

f)

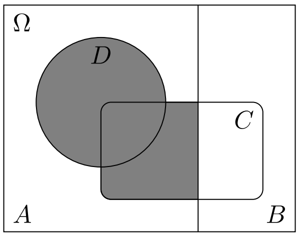
$$\begin{aligned} P[E_2 \cup E_3] =& \quad P[E_2] + P[E_3] - P[E_2 \cap E_3] \\ =& \quad P[E_2] + P[E_3] - P[\{(1,1)\}] \\ =& \quad \tfrac{3}{36} + \tfrac{9}{36} - \tfrac{1}{36} = \tfrac{11}{36}. \end{aligned}$$

2. a) The operators union ($\cup$), intersection ($\cap$) and complement (^{c}) operate on sets and the addition ($+$) on numbers. We get that (i) and (ii) are meaningful and (iii) and (iv) are not.

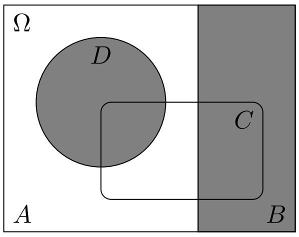
b) (i):



(ii):



(iii):



3. a) The event C occurs if either A or B occurs but not both. $C = (A \cup B) \setminus (A \cap B)$

b) $P[C] = P[(A \cup B) \setminus (A \cap B)] = P[A \cup B] - P[A \cap B] \overset{(*)}{=} P[A] + P[B] - 2P[A]P[B] = 0.1679$
Note that we made use of the fact that the events A and B are independent in the step $(*)$.

4. a) The solution is found dividing the number of smokers by the total number of participants:

$$P[\text{sm}] = \frac{1213}{6549} = 0.185.$$

b) We know that

$$P[A|B] = \frac{P[A \cap B]}{P[B]}.$$

In this case, A is the smoking fraction, while B is the high income fraction of the sample. We have

$$P[\text{hi}] = \frac{2115}{6549}.$$

while

$$P[\text{sm} \cap \text{hi}] = \frac{247}{6549}.$$

Then

$$P[\text{sm}|\text{hi}] = \frac{247}{6549} \cdot \frac{6549}{2115} = \frac{247}{2115} = 0.117.$$

c) To ask this question we have to answer the question if $P[\text{sm}] = P[\text{sm}|\text{hi}]$. If the conditional probability of being a smoker while having an high revenue is the same, then the two are independent. In this case we see from exercice a) and b) that these two qualities are dependent, *i.e.* the probability of smoking is lower for high revenue people than for the whole sample.

d) This question corresponds to asking if $P[\text{hsm}] = P[\text{hsm}|\text{wsm}]$, where "hsm" is the event of a husband smoking, "wsm" is the event of a wife smoking. We know that $P[\text{hsm}] = 0.3$ and $P[\text{wsm}] = 0.2$. Furthermore, we know that $P[\text{hsm} \cap \text{wsm}] = 0.08$. Then, we compute

$$P[\text{hsm}|\text{wsm}] = \frac{P[\text{hsm} \cap \text{wsm}]}{P[\text{wsm}]} = \frac{0.08}{0.2} = 0.4.$$

Hence the smoking status is dependent of that of the wife: if the wife smokes, the man has 40% of probability of smoking too.

5. a) Let X be the number of contaminated samples in one collective sample. The probability that a sample is contaminated is $\pi = 0.02$. Under the assumption that all samples are independent, X is binomially distributed: $X \sim \text{Bin}(n = 10, \pi = 0.02)$.
The probability not to find any contamination in the sample is given by

$$P[X = 0] = \binom{10}{0} \cdot 0.02^0 \cdot 0.98^{10} = 0.98^{10} = 0.8171.$$

In R, we can calculate $P[X = 0]$ as $>$ dbinom(0, size = 10, prob = 0.02) [1] 0.8170728
Another possible solution: each sample is clean with a probability of 0.98, independently of the other samples. Therefore we have

$$P[\text{all samples are clean}] = \prod_{i=1}^{10} P[i\text{-th sample is clean}] = 0.98^{10} = 0.8171.$$

b) The random variable Y can only have the values 1 or 11, because:

1. if all samples are clean, we are done after one analysis: $Y = 1$.
2. if at least one sample is contaminated, then the collective sample is contaminated and we need to check all 10 samples separately: $Y = 11$

Hence

$$P[Y = 1] = P[\text{no sample is contaminated}] = 0.8171\ ,$$
$$P[Y = 11] = 1 - P[Y = 1] = 0.1829\ .$$

c) The average number of analyses for one collective sample is given through the expectation value of Y:

$$E[Y] = \sum_{k=0}^{\infty} kP[Y = k] = 1 \cdot P[Y = 1] + 11 \cdot P[Y = 11] = 1 \cdot 0.8171 + 11 \cdot 0.1829 = 2.8293\ .$$

On average we save $10 - 2.8293 = 7.1707 \approx 7$ analyses.

6. We first prove the equality in case of discrete random variables. Then

$$\begin{aligned}\operatorname{Var}(X) &\overset{\text{def}}{=} \sum_{k=1}^{\infty}\left(x_k - E[X]\right)^2 p(x_k) = \sum_{k=1}^{\infty}\Big(x_k^2 + E[X]^2 - 2x_k E[X]\Big)\, p(x_k) \\ &= \underbrace{\sum_{k=1}^{\infty} x_k^2\, p(x_k)}_{E[X^2]} + E[X]^2 - 2E[X]\underbrace{\sum_{k=1}^{\infty} x_k\, p(x_k)}_{E[X]} = E[X^2] - E[X]^2.\end{aligned}$$

In case of a continuous random variable the proof is similar. We have

$$\begin{aligned}\operatorname{Var}(X) &\overset{\text{def}}{=} \int_{\mathbb{R}}\left(x - E[X]\right)^2 f(x)\,dx = \int_{\mathbb{R}}\Big(x^2 + E[X]^2 - 2xE[X]\Big)\, f(x)\,dx \\ &= \underbrace{\int_{\mathbb{R}} x^2\, f(x)\,dx}_{E[X^2]} + E[X]^2 - 2E[X]\underbrace{\int_{\mathbb{R}} x\, f(x)\,dx}_{E[X]} = E[X^2] - E[X]^2.\end{aligned}$$

7. The expectation value of a discrete random variable X can be calculated with the following formula:

$$E[X] = \sum_k k \cdot P[X = k].$$

As X is Poisson distributed with parameter λ, we know

$$P[X = k] = \frac{\lambda^k}{k!}e^{-\lambda}.$$

This gives us the equations:

$$\begin{aligned}E[X] = \sum_k k \cdot P[X = k] &= \sum_{k=0}^{\infty} k \cdot \frac{\lambda^k}{k!}e^{-\lambda} \\ &= e^{-\lambda}\sum_{k=1}^{\infty} k \cdot \frac{\lambda^k}{k!} \\ &= e^{-\lambda} \cdot \lambda \cdot \sum_{k=1}^{\infty} \frac{\lambda^{(k-1)}}{(k-1)!} \\ &= e^{-\lambda} \cdot \lambda \cdot \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \\ &= e^{-\lambda} \cdot \lambda \cdot e^{\lambda} \\ &= \lambda\end{aligned}$$

8. **a)** Haldane's model claims, that the number of crossovers is Poisson distributed with parameter $\lambda =$ "length of the chromosome in Morgan". Thus we have

$$P[X = 0] = \frac{\lambda^0}{0!} \cdot e^{-\lambda} = e^{-3.05} = 0.047.$$

If we solve this problem with R, we get the following solution:

```
> lambda <- 3.05
> ppois(0, lambda)
[1] 0.04735892
```

b) With the same arguments as in a) we have

$$P[k \geq 2] = 1 - P[k = 1] - P[k = 0] = 1 - \lambda \cdot e^{-\lambda} - e^{-\lambda} = 0.808.$$

We can solve this exercise also with R:

```
> 1 - ppois(1, lambda)
[1] 0.8081964
```

c) As for the Poisson distribution with parameter λ the expectation value is exactly λ, we get $E[X] = \lambda = 3.05$.

d) As k can only be a natural number, we have

$$P[k \geq 3.05] = P[k \geq 4] = 1 - P[k = 3] - P[k = 2] - P[k = 1] - P[k = 0] = 0.364.$$

A solution to solve this exercise with R is:

```
> 1 - ppois(lambda, lambda)
[1] 0.3639687
```

9. a) A recombination (event R) between the two genes happens if and only if there is an *odd* number of crossovers between them. Formally:

$$R = \{X \text{ is odd}\} = \{X = 1\} \cup \{X = 3\} \cup \{X = 5\} \cup \dots$$

b) By the consideration from task a), we have to calculate

$$P[R] = P[X = 1] + P[X = 3] + P[X = 5] + \dots$$

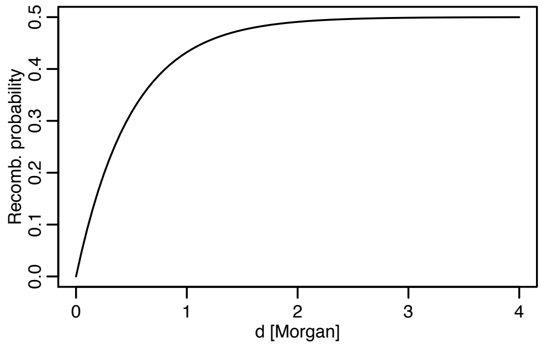
Since X is Poisson distributed with rate parameter $\lambda = d$ (Haldane's model!), we have $P[X = x] = e^{-d} \cdot \frac{d^x}{d!}$ and hence

$$P[R] = e^{-d} \cdot \left(d + \frac{d^3}{3!} + \frac{d^5}{5!} + \dots \right) = e^{-d} \sum_{k=0}^{\infty} \frac{d^{2k+1}}{(2k+1)!}$$

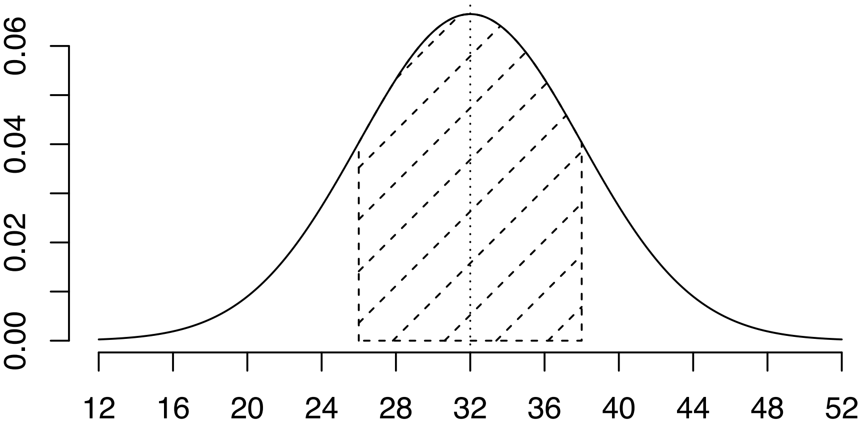
Using the hint, we find the recombination probability

$$P[R] = e^{-d} \sinh(d) \ .$$

A plot of this function shows that the recombination probability goes to $\frac{1}{2}$ as d grows, exactly as we expect:



10. a)



b) Let X be the lead content of the sample. It holds that

$$X \sim \mathcal{N}(\mu, \sigma^2) \qquad \text{with } \mu = 32 \text{ and } \sigma^2 = 6^2.$$

We solve the following problems with R. To calculate $P[X \leq 40]$ we use the following command:

```
> pnorm(40,mean = 32, sd = 6)
[1] 0.9087888
```

c) We calculate $P[X \leq 27]$.

```
> pnorm(27, mean = 32, sd = 6)
[1] 0.2023284
```

d) We chose c such that $P[X \leq c] = 0.975$.

```
> qnorm(0.975, mean = 32, sd = 6)
[1] 43.75978
```

e) This time we chose c such that $P[X \leq c] = 0.1$.

```
> qnorm(0.1, mean = 32, sd = 6)
[1] 24.31069
```

f) What is the probability of the area you draw in part a) of this exercise?

To calculate $P[26 \leq X \leq 38] = P[X \leq 38] - P[X < 26]$ we use the following command:

```
> pnorm(38, mean = 32, sd = 6) - pnorm(26, mean = 32, sd = 6)
[1] 0.6826895
```

11. **a)** If we calculate the sum of each row and each column, we see that the overall value of the probabilities is 1. We get the following table:

X/Y	1	2	3	$\sum$
1	0.05	0.08	0.12	0.25
2	0.14	0.19	0.09	0.42
3	0.22	0.08	0.03	0.33
$\sum$	0.41	0.35	0.24	1.00

The marignal distribution of X therefore is
$X = 1$ with probability 0.25
$X = 2$ with probability 0.42 and
$X = 3$ with probability 0.33.

The marignal distribution of Y therefore is
$Y = 1$ with probability 0.41
$Y = 2$ with probability 0.35 and
$Y = 3$ with probability 0.24.

b) The probability of X being on a low level is $P[X = 1] = 0.25$.
If we know the value of Y, the probability changes, as we now have a conditional probability. X can still take the same values (1,2 and 3), but as we already now that Y takes the value 2, only the middle column of the table is important to us. As we still need a total probability of 1, we need to adjust the probabilities, with which X takes its values. We do that with the formula

$$p_{X|Y=2}(i) = P[X = i|Y = 2] = \frac{P[X = i, Y = 2]}{P[Y = 2]} = \frac{p_{X,Y}(x, y)}{p_Y(2)} .$$

We have $p_{X|Y=2}(1) = P[X = 1|Y = 2] = \frac{0.08}{0.35} = 0.228571428571429$.

c) Gene X downregulates gene Y. Then for high X values, the higher Y values occur much less often than the smaller Y values. This is an indication of downregulation.

d) No, you can either show that by calculating the two values $p_{Y|X=1}(2)$ and $p_{Y|X=3}(2)$ or by argumentation.

In the cases of $p_{X,Y}(1, 2)$ and $p_{X,Y}(3, 2)$ we want to know the probabilities that Y takes the value 2 and X takes the value 1 (respectively 3). But we have just the probabilities what value X and Y are going to take.
In the case of $p_{Y|X=1}(2)$ and $p_{Y|X=3}(2)$ we already know the value of X. So we only need to know with which probability Y has the value 2, when X takes the value 1 (respectively 3).
As the random variables X and Y are dependent, there is a difference.

12. a) Importing the data set:

```
> d.pet <- read.table("count.txt", header = TRUE)
```

The argument `header = TRUE` tells R that the orignal dataset contains variable names on the first line.
The imported dataset `d.pet` is saved as a `data.frame`:

```
> class(d.pet)
[1] "data.frame"
```

It has 5000 rows and 4 columns (variables):

```
> dim(d.pet)
[1] 5000 4
```

The function `str()` shows the internal structure of the R-object.

```
> str(d.pet)
'data.frame': 5000 obs. of 4 variables:
$ a.v1: int 27 15 38 25 23 21 21 23 27 22 ...
$ a.v2: int 0 0 0 1 1 0 1 0 0 0 ...
$ b.v1: int 0 0 1 2 0 2 1 2 2 1 ...
$ c.v1: int 14 13 18 8 12 13 8 15 1 16 ...
```

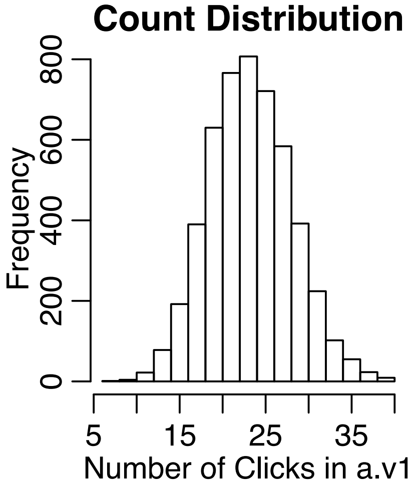
We can see the name of the variables, the saved type and a preview of the first entries. For example, the first variable is called `a.v1` and saved as an integer. In this dataset all variables of `d.pet` are saved as integers.

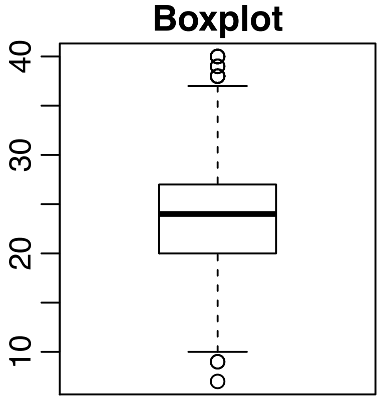
b) The characteristic numbers can be calculated as follows:

```
> mean(d.pet$a.v1) # Mean
[1] 23.708
> var(d.pet$a.v1) # Variance
[1] 23.13856
> summary(d.pet$a.v1) # Quantile, Minimum and Maximum
Min. 1st Qu. Median Mean 3rd Qu. Max.
7.00 20.00 24.00 23.71 27.00 40.00
```

c) Both the histogram and the boxplot represent the distribution of the observed click-count. However the histogram shows this distribution in its entirety by plotting the frequency of every possible number of clicks, whereas the boxplot might end up hiding some of the specific shape of the count distribution.

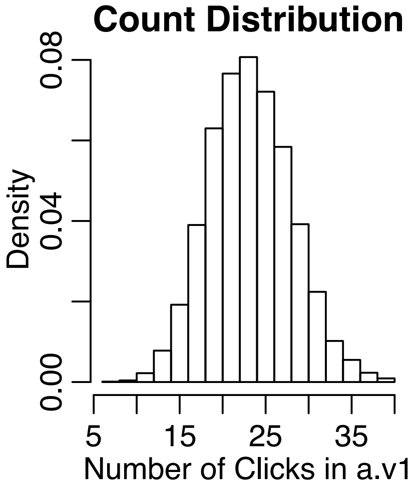
```
> d.pet <- read.table("count.txt", header = TRUE)
> par(mfrow = c(1,2)) # two plots in one window
> hist(d.pet$a.v1, main = "Count Distribution",xlab="Number of Clicks in a.v1")
> counts <- c(d.pet$a.v1)
> boxplot(counts, main = "Boxplot")
```





To obtain a scaled histogram we use the argument `probabilities = TRUE`.

```
> hist(d.pet$a.v1,probability = TRUE, main = "Count Distribution", xlab="Number of Clicks
in a.v1")
```



13. a) The number of test tubes of minor value X has a binomial distribution.

b) Here $X \sim \mathrm{Bin}(n, \pi)$ with $n = 50$ and $\pi = 0.1$. So we get

$$P[X = 3] = \binom{50}{3} 0.1^3 \cdot 0.9^{47} = 0.139.$$

c) We have again $X \sim \mathrm{Bin}(n, \pi)$ with $n = 50$ and $\pi = 0.1$. The probability that X is at most 3 is given by

$$P[X \leq 3] = \sum_{k=0}^{3} \binom{50}{k} (0.1)^k \cdot (0.9)^{50-k} = 0.25.$$

d) We use the central limit theorem (CLT). We approximate the cumulative distribution function (CDF) of X by the CDF of a normal distribution with mean $\mu = n\pi$ and variance $\sigma^2 = n\pi(1-\pi)$. So we get

$$P[X \leq 3] \approx 0.17.$$

We observe that the rule of thumb from the lecture notes is violated and so the approximation is rather imprecise. In this case it is better to use the real distribution.

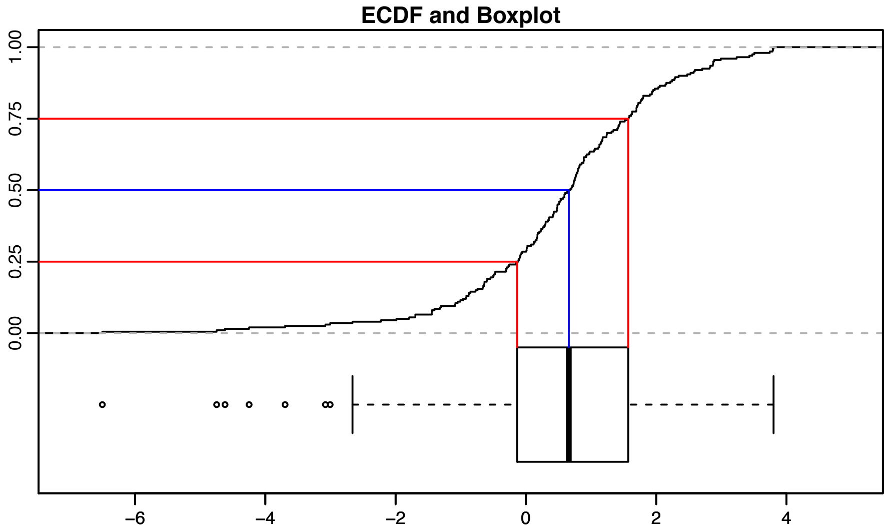
e) The manufacturer would like to define a critical bound K. If X lies below the bound the delivery is (with high probability) as desired and if X lies above K it is not. But as we saw in part (c), if he chooses $K = 3$ the probability that at most 3 test tubes are of minor value is only 25% even if the delivery is as desired. On the other hand, if he chooses K smaller than 3 the probability to reject the delivery becomes larger even if the delivery is as desired.

14. **a)**

$$a = 1 \qquad b = 3 \qquad c = 2 \qquad d = 4$$

b)

$$A = 2 \qquad B = 4 \qquad C = 3 \qquad D = 1$$



c)

15. **a)** The number of red ants X in the sample is binomially distributed with parameters $n = 5$ and $\pi = \frac{1}{10}$; hence we get

$$P[X = 3] = \binom{5}{3}\pi^3(1-\pi)^2 = 10 \cdot 0.1^3 \cdot 0.9^2 = 0.0081\ .$$

c) The number Y of red ants is binomially distributed with parameters $n = 150$ and $\pi = 0.1$, hence we have $E[Y] = n\pi = 15$ and $Var(Y) = n\pi(1-\pi) = 13.5$.

d) sWe can approximate the distribution of Y by a normal one with the same mean and variance:

$$Y \approx \mathcal{N}\left(n\pi, n\pi(1-\pi)\right) = \mathcal{N}\left(\mu = 15, \sigma^2 = 13.5\right) \quad \Rightarrow \quad Z = \frac{Y - 15}{\sqrt{13.5}} \approx \mathcal{N}(0,1)\ .$$

Hence we can calculate the probability that Y is between 15 and 20 using the cumulative distribution function of the normal distribution:

$$P\left[15 \leq Y \leq 20\right] = P\left[0 \leq Z \leq \frac{20-15}{\sqrt{13.5}}\right] \approx \Phi(1.36) - \Phi(0)$$

With R, we get the probability with either of the following approaches:

```
> pnorm(1.36) - pnorm(0)
[1] 0.413085
> pnorm(20, 15, sqrt(13.5)) - pnorm(15, 15, sqrt(13.5))
[1] 0.4132159 (where the second term is 0.5 in both cases, of course...)
```

16. **a)** > B = 1000; n = 50; la = 2; exp.value = 0.5; count = 0; > set.seed(4672) # Set the random number generator to a starting point. > for (i in 1:B) # Simulate B-times... x = rexp(n,la) # ...n random variables from the exponential distribution... conf.int = c(mean(x) - qnorm(0.95)*sd(x)/sqrt(n), mean(x) + qnorm(0.95)*sd(x)/sqrt(n)) # ...and calculate the confidence interval using the formula: # estimate +/- 1.64*standard-error-of-estimate # (which applies because of the central limit theorem). if(exp.value > conf.int[1] & exp.value < conf.int[2]) # Check if the CI contains the true mean. count = count + 1 # If so set the count plus 1. > prob = count/B # Calculate the probability that in our simulations > # the CI contained the true mean, using > # (# CIs-incl-true-mean)/(total# CIs-created)

After simulating n exponentially distributed random variables for B times, we find that in 890 of the cases our confidence interval did contain the true mean. That's a probability of 0.89 which is near our confidence level of 0.9.

b) In this case we obtain in 764 of the cases a confidence interval including the true mean. The fact that this number is lower as in a) does not surprise us. For if we chose a sample size of only $n = 5$ we expect that our results (CIs) will vary a lot. This is because in our calculation of the confidence interval we use the normal approximation. However, for small sample sizes this is not really appropriate.

17. We start by reading in the data set and extracting the different variables:

```
> fracture <- read.table("bone-fracture.csv", sep = ";", header = TRUE)

> conc <- fracture$conc

> dif <- fracture$dif

> no.cells <- fracture$no.cells

> hit <- fracture$hit
```

We then load the packages needed for fitting and Q-Q plots, and define the significance level needed for the calculation of the confidence intervals:

```
> library(car)

> library(MASS)

> alpha <- 0.05
```

Variable conc

From a histogram (see below), we guess that the data is approximately normally distributed. Fitting a normal distribution yields:

```
> (fit.conc <- fitdistr(conc, "normal"))

mean sd 3.41084484 0.38235027 (0.05407249) (0.03823503)
```

Note the numbers in brackets below the estimates: they denote the standard errors of the estimates. Hence we get lower and upper bounds of the 95% confidence intervals as follows:

```
> (conc.lower <- fit.conc$estimate - qnorm(1 - alpha/2)*fit.conc$sd)

mean sd
```

```
3.304865 0.307411
```

```
> (conc.upper <- fit.conc$estimate + qnorm(1 - alpha/2)*fit.conc$sd)
```

```
mean sd
```

```
3.5168250 0.4572895
```

In the above R-code sd is not the standard deviation but the estimated standard error.

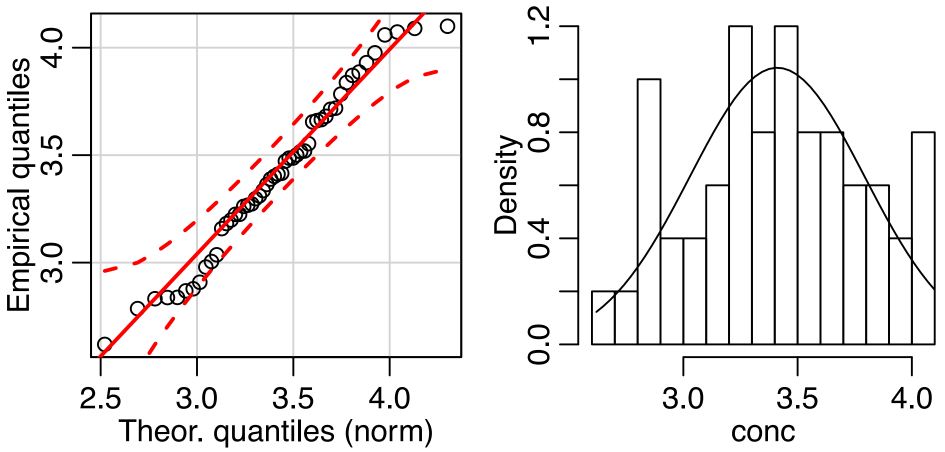
Checking the Q-Q plot and adding the estimated density into the histogram:

```
> par(mfrow = c(1, 2))
```

```
> qqPlot(conc, dist = "norm", mean = fit.conc$estimate["mean"], sd = fit.conc$estimate["sd"], xlab = "Theor. quantiles (norm)", ylab = "Empirical quantiles") > hist(conc, breaks = 20, freq = FALSE, main = "")
```

```
> x.val <- seq(min(conc), max(conc), length = 50)
```

```
> lines(x.val, dnorm(x.val, mean = fit.conc$estimate["mean"], sd = fit.conc$estimate["sd"]))
```



Remark for experienced R users: to avoid copying, pasting and adapting the R code above for the next three variables, we write a function which generates the Q-Q plot and the histogram:

```
> plot.density <- function(x, estimate, dist, ...)  par(mfrow = c(1, 2)) do.call(qqPlot, c(list(x = x, dist = dist), as.list(estimate), xlab = sprintf("Theor. quantiles (ylab = "Empirical quantiles")) hist(x, freq = FALSE, main = "", ...) if (is.integer(x)) x.val <- seq(min(x), max(x)) else x.val <- seq(min(x), max(x), length.out = 50) lines(x.val, do.call(sprintf("dc(list(x = x.val), as.list(estimate))))
```

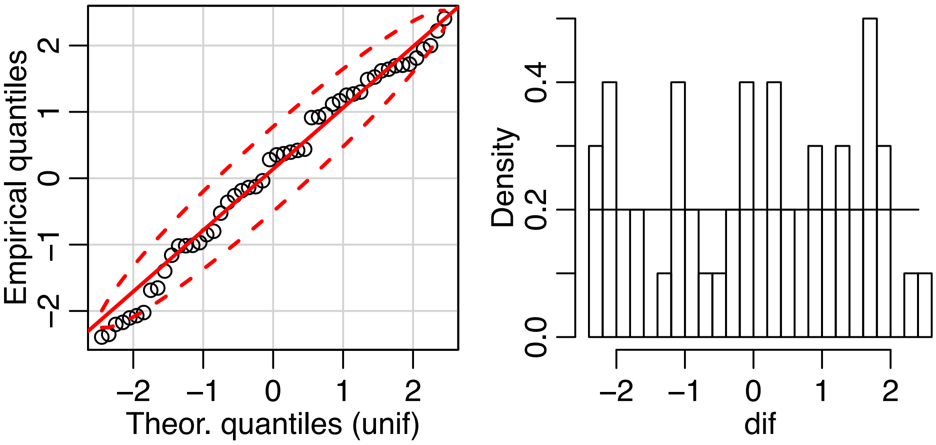
The plot above could then be generated with a single function call:

```
> plot.density(conc, fit.conc$estimate, "norm", breaks = 20, xlab = "conc")
```

Variable `dif`

Since no screw length in a 5 mm interval should be more likely than another one, we would expect the length differences to be approximately uniformly distributed in the interval $[-2.5\text{mm}, 2.5\text{mm}]$. There is no parameter to be estimated for the uniform distribution. We check our assumption with a Q-Q plot and plot the uniform density together with the histogram of the data:

```
> plot.density(dif, c(min = -2.5, max = 2.5), "unif", breaks = 20, xlab = "dif")
```

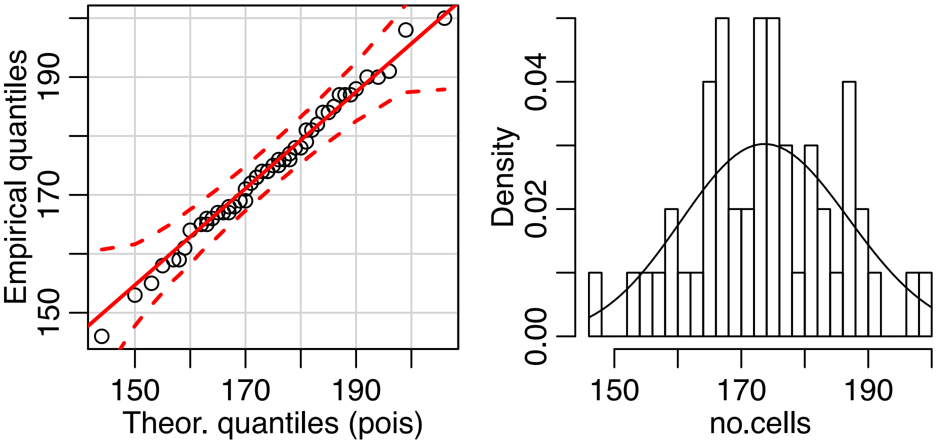


Variable `no.cells`
The distribution is concentrated around its mean. Since the data is discrete here, we fit a Poisson distribution and calculate the 95% confidence interval:

```
> (fit.cells <- fitdistr(no.cells, "poisson"))

lambda

174.160000

( 1.866333)

> (cells.lower <- fit.cells$estimate - qnorm(1 - alpha/2)*fit.cells$sd)

lambda

170.5021

> (cells.upper <- fit.cells$estimate + qnorm(1 - alpha/2)*fit.cells$sd)

lambda

177.8179
```

Checking the Q-Q plot and adding the estimated density into the histogram:

```
> plot.density(no.cells, fit.cells$estimate, "pois", breaks = 20, xlab = "no.cells")
```



Variable `hit`
`hit` is a Bernoulli variable; we only have to estimate its probability of being 1:

```
> (p <- mean(hit))
```

```
[1] 0.68
```

The standard error of the arithmetic mean is given by $\sigma/\sqrt{n}$, where σ denotes the standard deviation of the samples and n the sample size. Hence we can calculate the 95% confidence level for p as follows:

```
> (p.lower <- p - qnorm(1 - alpha/2)*sd(hit)/sqrt(length(hit)))
```

```
[1] 0.5493891
```

```
> (p.upper <- p + qnorm(1 - alpha/2)*sd(hit)/sqrt(length(hit)))
```

```
[1] 0.8106109
```

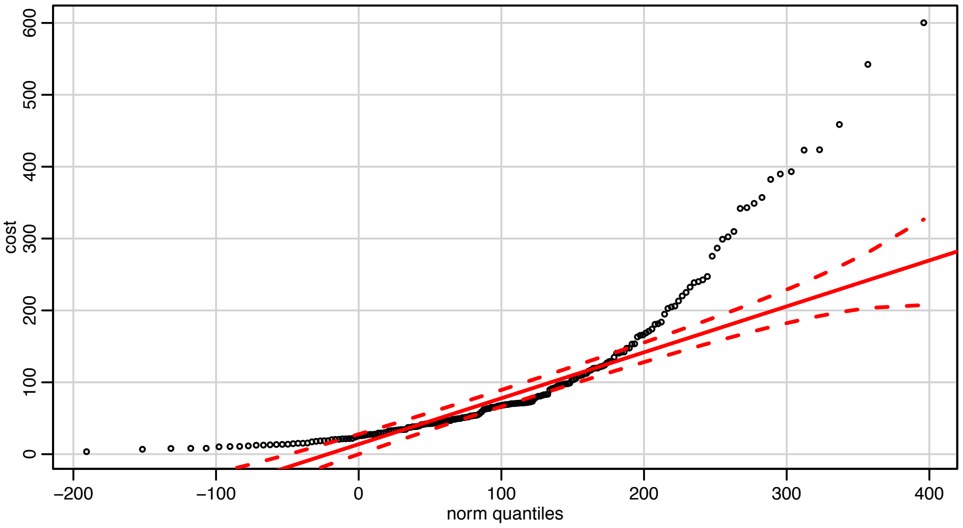
For Bernoulli variables, we can of course fit the empirical distribution exactly; a histogram for comparison is superfluous.

18. **a)** By looking at the shape of the histogram, we might guess that the data follow a poisson distribution. But this can not be since cost in CHF is a continuous variable which is rounded. Yet for a poisson distribution we need a variable which counts and only takes integers as values. So we try two other distributions (for continuous variables) we know: the exponential distribution and the normal distribution. We start with an automatic estimation of the parameters using the function `fitdistr()` from the package `MASS`:

```
> library(MASS)
> norm.fit <- fitdistr(cost, "normal")
```

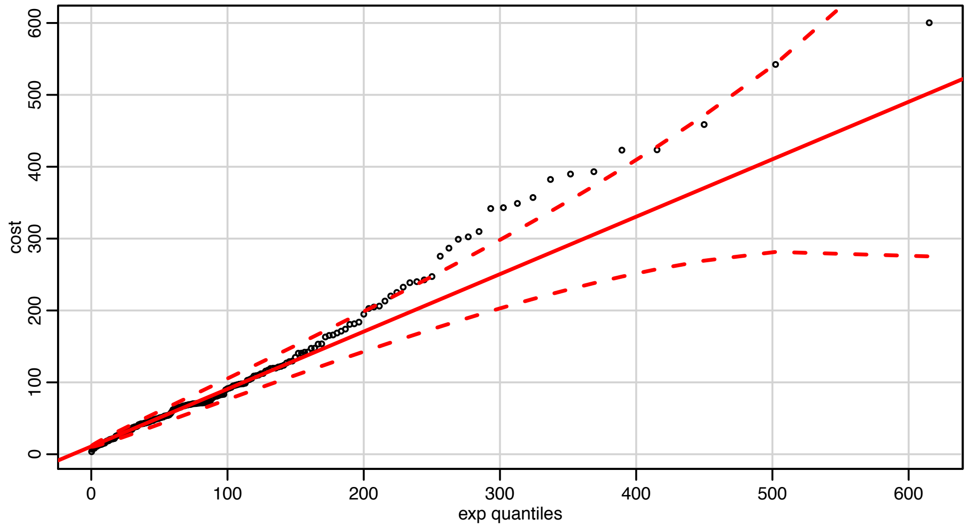
We start with testing the normal distribution. The Q-Q plot of the fitted distribution doesn't look good; this indicates that the choice of the normal distribution for this variable is not appropriate.

```
> library(car)
> qqPlot(cost, dist = "norm",
mean = norm.fit$estimate["mean"],
sd = norm.fit$estimate["sd"])
```



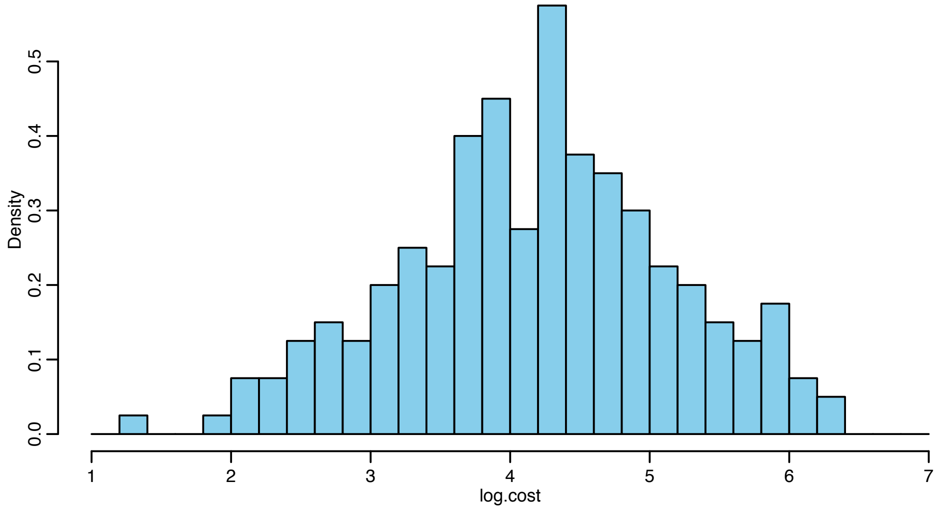
If we do the same for an exponential distribution we get:

```
> exp.fit <- fitdistr(cost, "exponential")
> qqPlot(cost, "exp", rate = exp.fit$estimate["rate"])
```



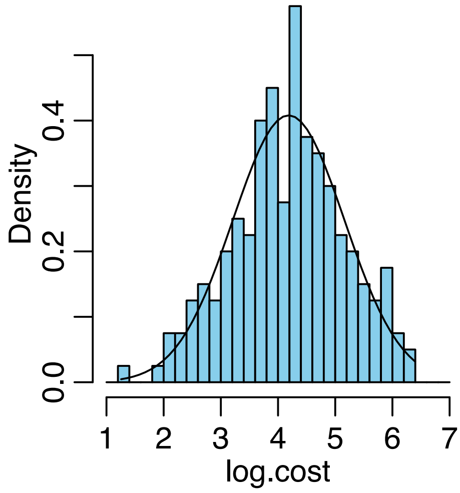
We see that the exponential distribution does not fit particularly well either.

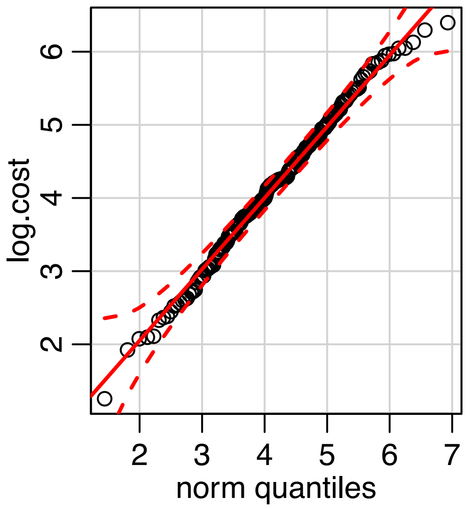
b) After the log-transformation, the data looks like this:



We fit a normal distribution to the transformed data set and check its Q-Q plot. Since the log-normal distribution fits the initial data well, it is not surprising that the normal distribution fits the log-transformed data well, as can be seen from both plots:

```
> par(mfrow = c(1, 2))
> norm.fit <- fitdistr(log.cost, "normal")
> hist(log.cost, freq = FALSE, breaks = seq(1, 7, by = 0.2), col = "skyblue", main = "")
> x.val <- seq(min(log.cost), max(log.cost), length = 50)
> lines(x.val, dnorm(x.val, mean = norm.fit$estimate["mean"], sd = norm.fit$estimate["sd"]))
> qqPlot(log.cost, dist = "norm", mean = norm.fitestimate[”mean”], sd = norm.fitestimate["sd"])
```





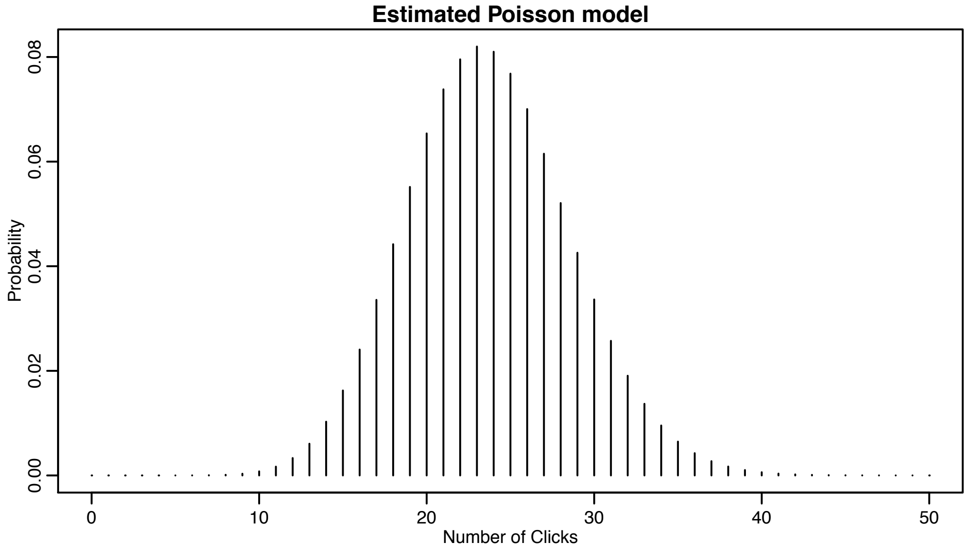
c) We can read off the parameters from the list `norm.fit`:

```
> norm.fit
mean sd
4.18541817 0.97764830
(0.06913017) (0.04888241)
```

Hence we have a normal distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ with mean $\hat{\mu} = 4.185$ and standard deviation $\hat{\sigma} = 0.978$.

19. a) The parameter λ of the Poisson distribution can be estimated by the mean of the counts of a.v1, because the MLE is exactly the mean of $a.v1$ (shown below). This gives $\lambda = 23.708$. The Poisson distribution is stored in the function `dpois()`. To plot the expected values we use the function `dpois()` with the estimated λ and calculate the probabilities of each number of clicks between 0 and 50.

```
> la <- mean(d.pet$a.v1)
> x <- 0:50
> expected <- dpois(x,lambda = la)
> plot(x, expected ,type = "h", ylab = "Probability", xlab = "Number of Clicks", main = "Estimated Poisson model")
```



In addition, let us show that the maximum likelihood estimator for the parameter λ is indeed the sample mean $\bar{x}$. The log-likelihood $l(\lambda)$ is given by

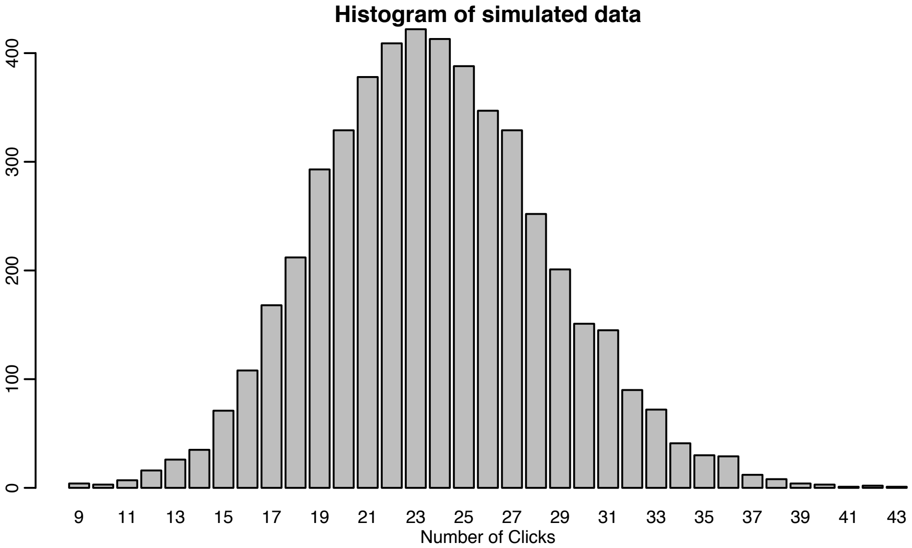
$$\begin{aligned} l(\lambda) &= \log \prod_{i=1}^{n} p(x_i; \lambda) = \sum_{i=1}^{n} \log(p(x_i; \lambda) \\ &= \sum_{i=1}^{n} \log(e^{-\lambda} \frac{\lambda^{x_i}}{x_i!}) \\ &= \sum_{i=1}^{n} \left[x_i \log(\lambda) - \lambda - \log(x_i!)\right] \\ &= \log(\lambda) n\bar{x} - \lambda n - \sum_{i=1}^{n} \log(x_i!). \end{aligned}$$

The MLE $\hat{\lambda} = \arg\max_{\lambda}(l(\lambda))$ can now be calculated by

$$\begin{aligned} l'(\lambda) &= \frac{n\bar{x}}{\lambda} - n \overset{!}{=} 0 \\ &\Rightarrow \hat{\lambda} = \bar{x}. \end{aligned}$$

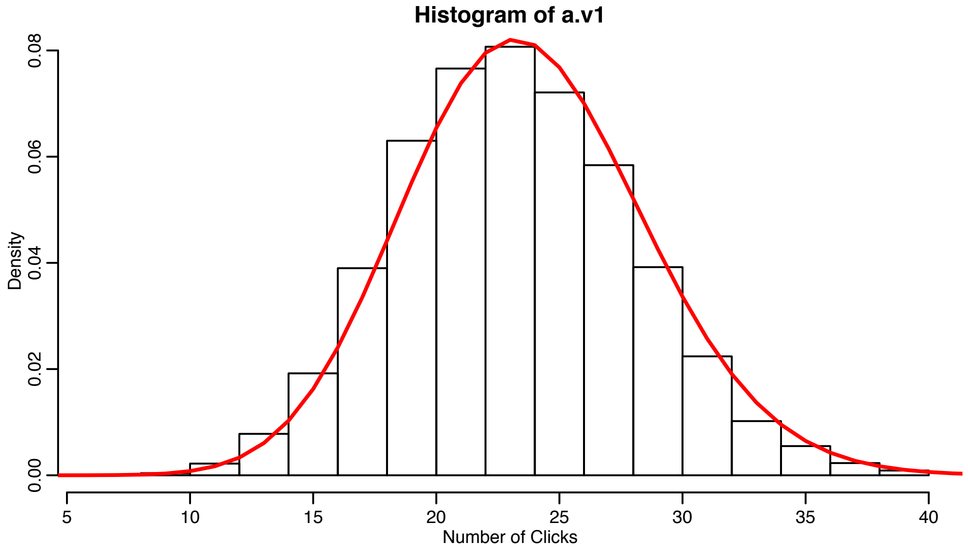
b) The function `rpois()` generats random numbers from a Poisson distribution. The estimated λ and the length of the series can be set as arguments.

```
> n <- nrow(d.pet) # Number of observations
> set.seed(4892) # Set seed for the random number generator.
> sim <- rpois(n, lambda = la) # Simulating a new series of counts
> barplot(table(sim), xlab="Number of Clicks", main="Histogram of simulated data")
```



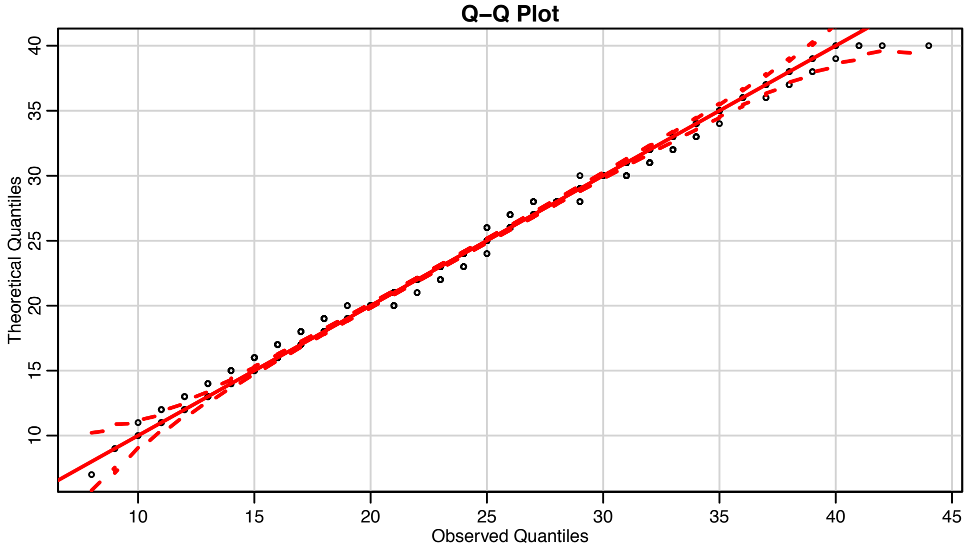
c) First we draw a histogram of the observed counts of `av.1`. The histogram can be scaled to probabilities with the additional option `probability = TRUE`. Afterwards we can add the curve of estimated counts with the function `lines()` .

```
> hist(d.pet$a.v1,probability = TRUE, main = "Histogram of a.v1", xlab = "Number of Clicks")
> x <- 0:50
> lines(x,dpois(x,lambda = la),col = "red", lwd = 2)
```



The red curve fits the histogram very well. Hence, the Poisson model describes the data well. A second possibility for checking the goodness of fit is the quantile-quantile plot (QQ-Plot). In this plot the quantiles of the observed counts are plotted against the quantiles of the counts from the Poisson model. If model and data fit well, we should see a straight line.

```
> library("car")
> sim.dist <- rpois(n, lambda= la)
> qqPlot(d.pet$a.v1,dist = "pois",lambda = la,main="Q-Q Plot", ylab = "Theoretical Quantiles", xlab = "Observed Quantiles")
```



The points fall along the line and into the dotted confidence region. This indicates a good fit too.

20. **a)** Note that the number of cured patients is a binomially distributed random variable, $X \sim \text{Bin}(n = 10, \pi = 0.3)$. Hence:

$$P(X = k) = \binom{10}{k} 0.3^k 0.7^{n-k}$$

$$P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2) = 0.7^{10} + \binom{10}{1} 0.3^1 0.7^9 + \binom{10}{2} 0.3^2 0.7^8 = 0.38$$

The solution can be derived in R in the following way:

```
> n <- 10
```