

Confidence, prediction and tolerance intervals in classical and Bayesian settings

Master Thesis in Biostatistics (STA495)

by

Sonja Hartnack
16-745-283

supervised by PD Dr. Malgorzata Roos

Zurich, May 2019

Confidence, prediction and tolerance intervals in classical and Bayesian settings

Sonja Hartnack

Version May 13, 2019

Contents

Abstract	1
1 Introduction to classical and Bayesian intervals	3
2 Confidence intervals	7
2.1 Confidence intervals for a normal distribution	7
2.2 Confidence intervals for a proportion based on a binomial distribution	10
3 Prediction intervals	17
3.1 Prediction intervals for a normal distribution	17
3.2 Prediction intervals for a binomial distribution	19
4 Tolerance intervals	26
4.1 Non-parametric tolerance intervals based on order statistics	26
4.2 Tolerance intervals for a normal distribution	27
4.3 Tolerance intervals for a binomial distribution	28
5 Contrasting different types of intervals	31
5.1 Simulated data sets: normality assumption	31
5.2 Behaviour subject to increasing sample size: normal distribution	34
5.3 Behaviour subject to increasing sample size: binomial distribution	39
6 Discussion	44
7 Appendix	46
7.1 Distribution of sample mean	46
7.2 Distribution of a sample proportion	46
7.3 Sum of variances	47
8 Acknowledgments	48
9 R code	49
Bibliography	81

Abstract

In good statistical practice it is recommended to limit the results not only to point estimates, but to present simultaneously interval estimates. The standard, well known and frequently taught interval in introductory statistical classes, is the confidence interval. It is designed to describe a single unknown parameter of a population with some uncertainty.

Another type of interval, the prediction interval, is occasionally presented in the context of regression analysis. The aim of the prediction interval is to predict future observation(s) in a population with a defined uncertainty. In contrast to these two intervals, tolerance intervals, capturing a specified proportion P of a population with some uncertainty, are rarely presented. Depending on the context and on the purpose of a study, different intervals are recommended. Presenting a confidence or a prediction interval, when a tolerance interval would be appropriate, engenders potentially a misuse of the former two intervals.

In this thesis the three different types of intervals are illustrated with simulated data and two real data sets from veterinary medicine. One data set of normally distributed data originates from a study on body weight loss in obese dogs. The second data set of binomial data comes from a study simulating fractures caused by horse kicks and different shoeing materials.

The intervals for a normal and a binomial distribution are interpreted from both a classical and a Bayesian perspective. Next to theoretical aspects, practical applications in R are presented. The asymptotic behaviour of the three intervals, subject to increasing sample size, is assessed. With increasing sample size, the width of confidence intervals decreases and approaches 0. The widths of the prediction and the tolerance intervals decrease initially, but will reach a stable plateau. The magnitude of the width of these two intervals will be influenced by the standard deviation and, additionally for the tolerance intervals, the captured proportion P .

It became also evident, that the assumption of normality needs to be checked carefully to avoid erroneous intervals. For binomial distributions, Bayesian approaches were found to be superior to Wald normal approximations which should be avoided.

As the three types of intervals serve for different purposes, the decision which interval to use should be context-driven and clearly justified. In order to avoid confusion and misunderstandings, all three types of intervals should be taught and presented in introductory statistical classes.

Chapter 1

Introduction to classical and Bayesian intervals

Statistics can be understood as a set of analytical tools designed to quantify uncertainty. This uncertainty originates from the situation when inferences pertaining to a whole population are desired, but only a subset of the population, a sample, is available to obtain such inferences. Thus, due to sampling variation or error, solutions are needed to deal appropriately with this uncertainty.

More precisely, in Maximum Likelihood approaches, the idea is to determine the most likely value of the parameter of interest. Assuming the probability distribution from which the sample was drawn, constructing a likelihood function, looking at its first derivative and setting this value to zero, allows to find this most likely value based on the data at hand. Unfortunately, this most likely value, the point estimate, will with high probability differ from the true parameter of interest of the population from which the sample was drawn. If the point estimate is unable to catch up the true parameter value, there is a need for interval estimates.

There are three main types of intervals estimates, e.g. confidence or credible (CI or CrI), prediction (PI) and tolerance intervals (TI) which have been developed serving for different purposes. Confidence or credible intervals are designed to describe a parameter with some uncertainty due to sampling errors. Prediction intervals aim to predict future observation(s) including some uncertainty. Tolerance intervals are constructed to capture a specified proportion of a population with a defined uncertainty.

There are two approaches in statistics: Bayesian and frequentist. Both allow providing different intervals. The first approach is named after Thomas Bayes who lived in the 18th century in the Age of Enlightenment. At that time, the development of the probability theory provided means to make probabilistic statements about the likelihood of events.

In a Bayesian setting, the underlying assumption is that there is a parameter distribution, while the data are fixed. Consequently, the parameter of interest is contained in a 95% credible interval with a 95% probability. In addition to the likelihood, containing information about the data, the prior information, needs to be considered. Positively formulated, it can be included as an already existing source of information. Based on these two factors, the posterior distribution is derived from which the credible interval can be read off. Bayesian statistics was contested in the 19th and 20th century and gained a wider acceptance among a number of statisticians with the recent increase in computational power.

The second approach was established about a century ago by pioneers like Ronald Fisher, Karl and Egon Pearson, and Jerzy Neyman. The overarching idea in the classical branch of statistics is to assume that the unknown true parameter of interest is fixed, and can be learned or approximated by repeatedly (*frequently*) drawn samples (of identical, independent observations) from the population. This parameter of interest could be a mean or a variance in the case of a normal, or a probability of an event in a binomial probability distribution. A 95% classical interval will

allude to the sampling experiment: “when sampling multiple times from ...”, “the true *parameter of interest* is included in 95% of the obtained intervals”.

Frequentist reasoning is still largely dominating medical statistics and is therefore considered as the *classical* approach which is typically taught in statistics.

All three types of intervals can be described in the generic form of adding and subtracting k times the standard error from the point estimate of the parameter of interest:

$$\widehat{parameter} \pm k \cdot SE(\widehat{parameter}). \quad (1.1)$$

Thus the width of the interval is based on the product of “ k ” times the standard error SE , and centered around the most likely value of the parameter of interest. The key difference for the three intervals lies in the specific definition of k .

The main idea behind equation (1.1), is expressed by the Wald approach. This approach is named after Abraham Wald and relies on an approximation with the normal distribution. It can be summarised as follows:

Assume that a random variable X follows a normal probability distribution with an unknown mean μ and a known variance σ^2 :

$$X \sim N(\mu, \sigma^2) \quad (1.2)$$

A standardized normal random variable is obtained by subtracting the mean μ from X and dividing by the standard deviation σ which in turn follows a standard normal distribution with a mean of 0 and a variance of 1.

$$\frac{X - \mu}{\sigma} = \frac{X - \mu}{\sqrt{\sigma^2}} \sim N(0, 1) \quad (1.3)$$

Consider now a sample x_1, \dots, x_n realisations of $X_1, \dots, X_n \stackrel{\text{iid}}{\sim} N(\mu, \sigma^2)$ and a sample mean

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Regarding the distribution of sample mean \bar{X} , it can be shown (see [Appendix 7.1](#)) that it follows a normal distribution with mean μ and variance $\frac{\sigma^2}{n}$.

$$\bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right) \quad (1.4)$$

In a standardized form this can be written as

$$\frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} = \frac{\bar{X} - \mu}{\sqrt{\frac{\sigma^2}{n}}} \sim N(0, 1) \quad (1.5)$$

Denote the $100(1 - \frac{\alpha}{2})$ th percentile of the standard normal distribution by $z_{1-\frac{\alpha}{2}}$ -value. For $\alpha = 5\%$ and a 95% confidence level $(1 - \alpha)$, $z_{1-\frac{\alpha}{2}}$ equals to 1.96. Furthermore it holds:

$$\mathbb{P}\left[|N(0, 1)| \leq z_{1-\frac{\alpha}{2}}\right] = 1 - \alpha \quad (1.6)$$

If equation (1.5) is considered for a sample, then

$$\mathbb{P}\left[\left|\frac{\bar{x} - \mu}{\sqrt{\frac{\sigma^2}{n}}}\right| \leq z_{1-\frac{\alpha}{2}}\right] = 1 - \alpha \quad (1.7)$$

and

$$\mathbb{P}\left[-z_{1-\frac{\alpha}{2}} \leq \frac{\bar{x} - \mu}{\sqrt{\frac{\sigma^2}{n}}} \leq z_{1-\frac{\alpha}{2}}\right] = 1 - \alpha \quad (1.8)$$

when subtracting \bar{x} , multiplying with $\sqrt{\frac{\sigma^2}{n}}$ it is possible to obtain a $(1 - \alpha)$ Wald confidence interval with a lower and upper bound for the unknown parameter μ :

$$\mathbb{P}\left[\bar{x} - z_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}} \leq \mu \leq \bar{x} + z_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}}\right] = 1 - \alpha \quad (1.9)$$

If σ is unknown, it will be replaced by the estimate $\hat{\sigma}$. In such a case $z_{1-\frac{\alpha}{2}}$ could be replaced by a quantile of a t_{n-1} distribution. In order to keep the argument concise, in what follows, however, we concentrate on the normal approximation only.

Thus, with $k = z_{1-\frac{\alpha}{2}}$ and $SE(\hat{\mu}) = \frac{\hat{\sigma}}{\sqrt{n}}$:

$$\hat{\mu} \pm k \cdot SE(\hat{\mu}) \quad (1.10)$$

In the binomial case, for a proportion p , and its sample estimate \hat{p} , following a Bernoulli probability distribution with $E\hat{p} = p$ and $Var(\hat{p}) = \frac{p(1-p)}{n}$ (see [Appendix 7.2](#)).

$$\hat{p} \sim N\left(p, \frac{p(1-p)}{n}\right) \quad (1.11)$$

and subsequently, using the standard normal approximation:

$$\frac{\hat{p} - p}{\sqrt{\frac{p(1-p)}{n}}} \sim N(0, 1) \quad (1.12)$$

In practice, however, one uses

$$\frac{\hat{p} - p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}} \sim N(0, 1) \quad (1.13)$$

Following the steps in equations (1.7) to (1.9) with equation (1.5) replaced by equation (1.12) we arrive at a $(1 - \alpha)$ Wald confidence interval for p :

$$\mathbb{P}\left[\hat{p} - z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq p \leq \hat{p} + z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right] = 1 - \alpha \quad (1.14)$$

Thus

$$\hat{p} \pm k \cdot SE(\hat{p}) \quad (1.15)$$

with $k = z_{1-\frac{\alpha}{2}}$ and $SE(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$.

Aim of the master thesis

The aim of this master thesis is to present, from a classical and a Bayesian perspective, the confidence/credible, prediction and tolerance intervals for normal and binomial distributions. Next to theoretical aspects, applied examples are presented including two real data sets from veterinary medicine and simulated data.

The following data sets are used. The first one originates from the study of (Flanagan *et al.*, 2017) describing a weight loss program in dogs. Obesity has become a concern in dogs too, and effective weight loss solutions are required to avoid negative health effects. In a 3-month prospective multicenter observational cohort study, different commercially available dry or wet weight loss diets were assessed for their effect on the percentage weight loss per dog. In total 926 dogs from 340 veterinary practices in 27 countries were involved. The study was sponsored by Royal Canin.

The second data set from (Sprick *et al.*, 2017) assesses the damage inflicted by an unshod hoof and by the horseshoe material steel on the long bones of horses. Due to welfare reasons, horses are increasingly kept in groups. During social interactions, kicks - particularly with the hind limbs - possibly cause fractures at the long bones, radii and tibiae, when loads are applied perpendicular to the longitudinal axis. In group holding systems shoeing with steel is often prohibited. A fracture of these long bones was and still is a potential reason to euthanase a horse. In the study (Sprick *et al.*, 2017) kicks with a comparable velocity of 8m/s were simulated using a drop impact test setup. The data for an impactor with steel and horn are used as an applied example.

With the aim to illustrate the three types of intervals for a normal distribution, samples are drawn from a random normal distribution with a true mean of 100 and a standard deviation of 10. Similarly, to illustrate the three types of intervals for a binomial distribution, samples are drawn from a random binomial distribution, with a true proportion of 0.5 and of 0.1. Additionally, simulated data originating from a normal, a bi-modal and a skewed are generated to assess the importance of the normality assumption with regard to the obtained intervals.

The master thesis is structured as follows. In the first chapter classical and Bayesian intervals are briefly introduced. In the second chapter confidence and credible intervals are described. Chapter three presents prediction intervals and chapter four is dedicated to tolerance intervals. In chapter five, the three types of intervals are contrasted. The first comparison is made with regard to the normality assumption in the case of normal, bi-modal and skewed data. The second focus is related to increasing sample size n as well as the number of predicted observations m . In chapter six the findings are discussed. The three different intervals are contrasted highlighting their appropriate use and interpretation. In the appendix, the formula for the distribution of sample mean and proportion, as well as the sum of variances are presented.

Chapter 2

Confidence intervals

Confidence intervals (CI) were first described by (Neyman, 1937). According to (Held and Sabanés Bové, 2014, p.57), the following definition for classical confidence intervals for an unknown parameter θ applies:

Definition of a classical confidence interval:

For identical and independent repetitions of the underlying statistical sampling experiment, a $(1 - \alpha) \cdot 100\%$ confidence interval will cover θ in $(1 - \alpha) \cdot 100\%$ of all cases.

In contrast in a Bayesian setting, a credible interval (CrI) is defined as (Held and Sabanés Bové, 2014, p.172):

Definition of a Bayesian credible interval:

For a fixed $(1 - \alpha) \in (0, 1)$ a $(1 - \alpha) \cdot 100\%$ credible interval is defined through two real numbers t_l and t_u that fulfill:

$$\int_{t_l}^{t_u} f(\theta|x)d\theta = 1 - \alpha \quad (2.1)$$

with $(1 - \alpha)$ the credible level for the interval $[t_l, t_u]$.

This implies that θ is contained in the $(1 - \alpha)$ credible interval with probability $(1 - \alpha)$.

2.1 Confidence intervals for a normal distribution

The definition of a classical confidence interval is demonstrated for a normal distribution in Figure 2.1. In this example, the aim is to provide a 95% confidence interval for an unknown mean μ . When sampling multiple times from a random normal distribution with a specified mean, e.g. of 100 and a specified standard deviation of 10, the true mean of 100 is included in 95% of the obtained confidence intervals. In 5% of the cases, the true parameter is not covered by the 95% CI (μ).

This derivation of a $(1 - \alpha) \cdot 100\%$ CI (μ) for the mean is based on equation (1.9) with sample standard deviation $\hat{\sigma}$, and reads:

$$\mathbb{P}\left[\bar{x} - z_{1-\frac{\alpha}{2}} \cdot SE(\hat{\mu}) \leq \mu \leq \bar{x} + z_{1-\frac{\alpha}{2}} \cdot SE(\hat{\mu})\right] = 1 - \alpha \quad (2.2)$$

The standard error $SE(\hat{\mu}) = \frac{\hat{\sigma}}{\sqrt{n}}$ of the sample mean depends on both the standard deviation $\hat{\sigma}$ and the sample size n . The standard error of the mean can also be considered as the accuracy of the sample mean \bar{x} estimate, indicating the uncertainty around the estimate of the mean value (Altman and Bland, 2005).

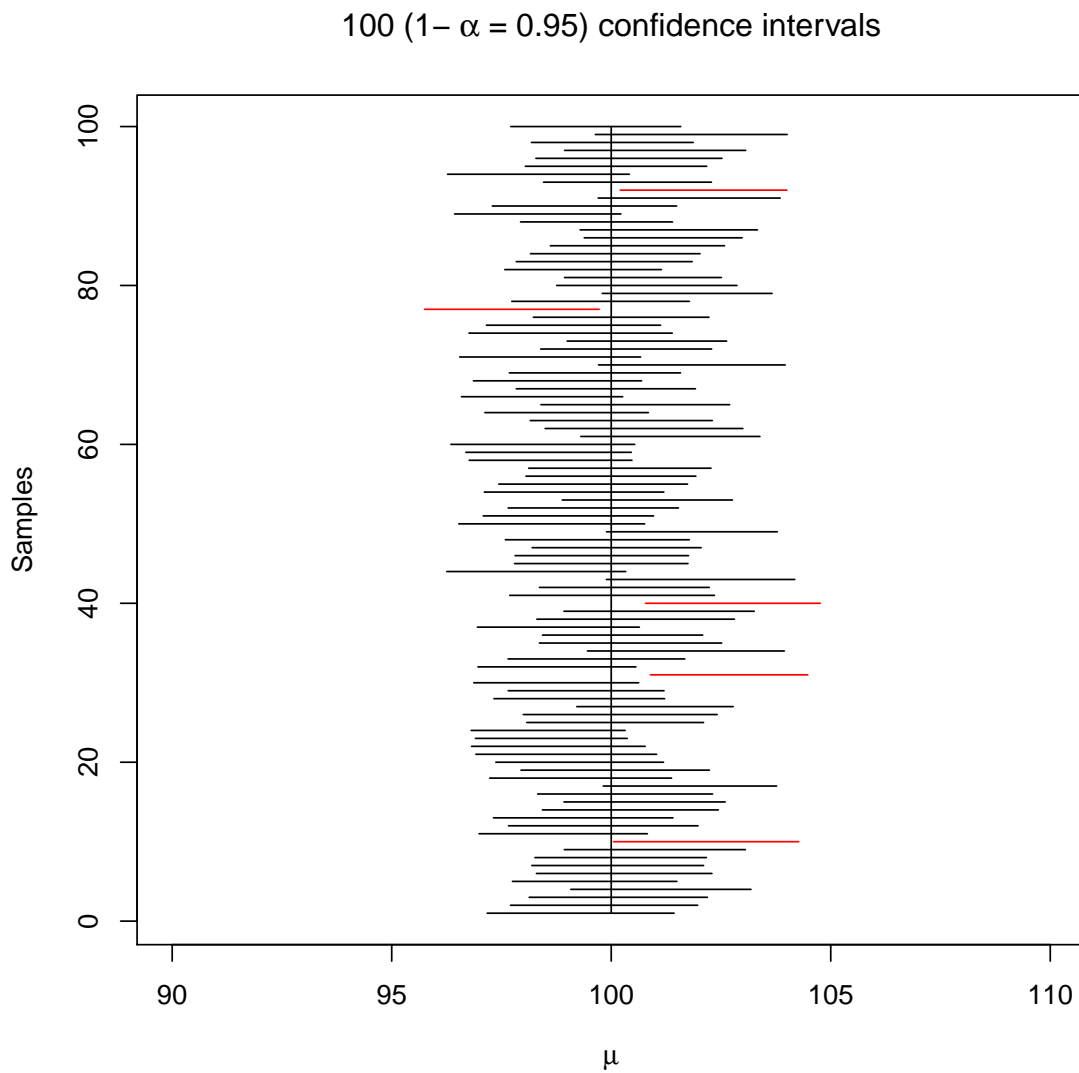


Figure 2.1: Illustration of confidence intervals $CI(\mu)$ for an unknown mean parameter μ . The confidence intervals which do not cover the true mean are colored in red (here 5 out of 100). Similar properties apply to $CI(p)$ discussed in section 2.2.

2.1.1 Normal distribution: example of applicability

In this example from (Flanagan *et al.*, 2017), a multicenter study sponsored by Royal Canin, the percentage of weight loss is evaluated in 926 dogs. The mean percentage weight loss is 11.4% with a standard deviation of 5.84%. The data are shown in Figures 2.2 and 2.3.

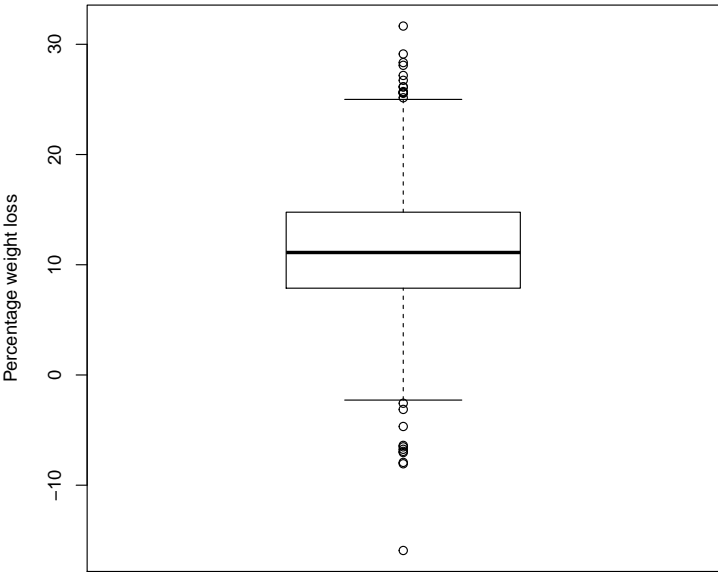


Figure 2.2: Boxplot of percentage of body weight loss in dogs based on the study of (Flanagan *et al.*, 2017).

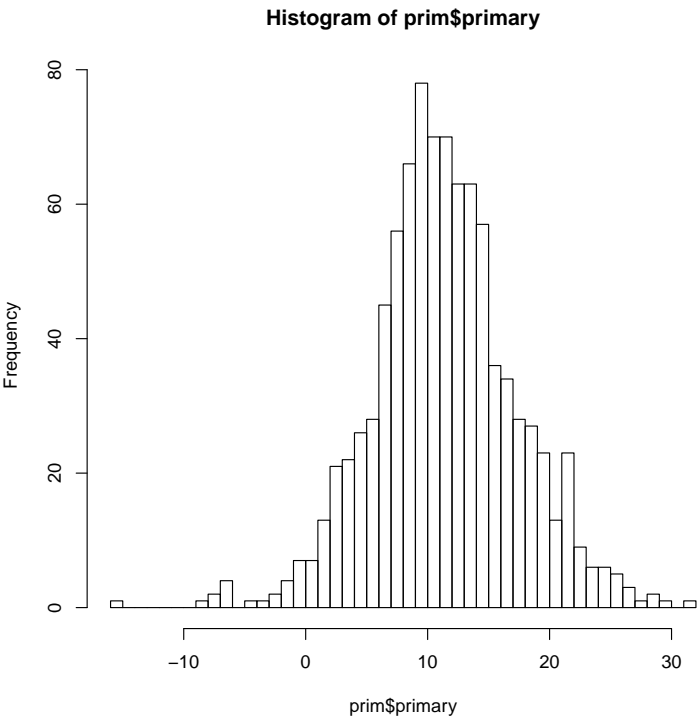


Figure 2.3: Histogram of percentage of body weight loss in dogs based on the study of (Flanagan *et al.*, 2017).

Based on the above described equation (2.2) the following own function was written in R with the arguments of the data (x) and the chosen $(1 - \alpha)$ level for the confidence interval.

```
my.CI.norm.func <- function(x, alpha){
  # x = data
  # 1 - alpha = confidence level
  SE <- sd(x) / sqrt(length(x))
  low.ci <- mean(x) - SE * qnorm((1-alpha/2),0,1)
  up.ci <- mean(x) + SE * qnorm((1-alpha/2),0,1)
  low.ci.f <- floor(low.ci * 100) /100
  up.ci.c <- ceiling(up.ci * 100) /100
  return(list(lower.ci= low.ci.f, upper.ci = up.ci.c))
}
```

R packages

In the following, the results of my.CI.norm.function will be compared with the functions *ci.mean()* (package **Publish**), *CI()* (package **Rmisc**), and *MeanCI()* (package **DescTools**) for constructing confidence intervals.

Package	Function	lower bound	upper bound
Wald	my.CI.norm.function	10.99	11.75
Publish	ci.mean()	10.99	11.74
Rmisc	CI()	10.99	11.74
DescTools	MeanCI()	10.99	11.74

Table 2.1: 95% CI (μ) for percentage body weight loss of 926 dogs in the (Flanagan *et al.*, 2017) study.

Interpretation

Thus for repeated samplings (i.e. independent, identical realisations of the experiment) of dogs in the (Flanagan *et al.*, 2017) study, the interval [10.99 to 11.75] according to my own function, or [10.99 to 11.74] according to the R packages **Publish**, **Rmisc** and **DescTools** will contain the (unkown) true percentage of body weight loss in 95% of the cases (see table 2.1).

2.2 Confidence intervals for a proportion based on a binomial distribution

The traditional (and still widely used) method for obtaining a confidence interval for an unknown parameter p , a proportion, based on a sample drawn from a binomial distribution, relies on a normal approximation.

Thus, based on the idea of a confidence interval obtained as described in the introduction, when \hat{p} is a point estimator for the probability of successes in a repeated Bernoulli trial based on x successes out of n trials

$$\hat{p} = \frac{x}{n} \quad (2.3)$$

the formula for the $(1 - \alpha)$ CI(p), similar to equation (1.9) with

$$\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = SE(\hat{p}) \quad (2.4)$$

it is possible to write

$$\mathbb{P}\left[\hat{p} - z_{1-\frac{\alpha}{2}} \cdot SE(\hat{p}) \leq p \leq \hat{p} + z_{1-\frac{\alpha}{2}} \cdot SE(\hat{p})\right] = 1 - \alpha \quad (2.5)$$

The $k = z_{1-\frac{\alpha}{2}}$ here is exactly the same as in equation (1.10).

Below is the own function providing Wald (or approximate normal) formula for a $(1 - \alpha) \cdot 100\%$ CI(p) implemented in R with the arguments of the data (x successes out of n trials) and the chosen confidence level $(1 - \alpha)$. The name “Wald” makes the link of the confidence interval to the Wald test for p obvious. The interval is the set of p_0 values having a p-value exceeding α in testing

$$H_0 : p = p_0 \quad \text{against} \quad H_A : p \neq p_0$$

using the test statistic

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}} \quad (2.6)$$

The Wald CI results from inverting the Wald test for p according to (Agresti and Coull, 1998). Based on the equation in (2.5), the following formula is implemented in R, based the number of successes x and trials n and the chosen confidence level $1 - \alpha$.

```
my.CI.Wald.function <- function(x, n, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # 1 - alpha = confidence level
  p <- x/n
  SE <- sqrt((p*(1-p))/n)
  low.ci <- p - SE * qnorm((1-alpha/2),0,1)
  up.ci <- p + SE * qnorm((1-alpha/2),0,1)
  low.ci.f <- floor(low.ci*1000) /1000
  up.ci.c <- ceiling(up.ci*1000) /1000
  return(c(lower=low.ci.f, upper=up.ci.c))
}
```

There are cases when this normal approximation is not appropriate, for example for very small or large proportions, i.e. when p is near the boundaries of 0 or 1. In such a case, $SE(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$ is close to 0. This might be a consequence of using \hat{p} as a midpoint in the normal approximation, when the binomial distribution is skewed. In this context, two other approaches are briefly mentioned: Clopper-Pearson exact and Wilson confidence intervals.

Clopper-Pearson exact confidence intervals

Clopper and Pearson developed a method named Clopper-Pearson method or “exact confidence interval” (Clopper and Pearson, 1934). In regard to the binomial distribution with the density function:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x} \quad (2.7)$$

The idea is to use two one-sided Binomial tests, one for the lower (P_{Lo}) and one for the upper (P_{Up}) confidence bound at the desired level of confidence. This confidence interval is based on inverting equal-tailed binomial tests of $H_0 : p = p_0$. The exact Clopper-Pearson CI is then the range from the endpoints P_{Lo} to P_{Up} satisfying the following conditions for x successes out of n trials:

$$\sum_{l=0}^x \binom{n}{l} P_{Up}^l (1 - P_{Up})^{n-l} = \frac{\alpha}{2} \quad (2.8)$$

$$\sum_{l=x}^n \binom{n}{l} P_{Lo}^l (1 - P_{Lo})^{n-l} = \frac{\alpha}{2} \quad (2.9)$$

For $x = 0$ the lower bound will be 0 and for $x = n$ the upper bound will be 1. This confidence interval is guaranteed to have coverage probability of at least $(1 - \alpha)$ for every possible value of p . According to (Brown *et al.*, 2001), for any fixed p , the actual coverage probability can be much larger than $(1 - \alpha)$ unless n is quite large. In this sense, the confidence interval is rather inaccurate.

The following function implements the Clopper-Pearson approach for an exact confidence interval in R with the arguments of the data (x successes out of n trials) and the chosen confidence level $(1 - \alpha)$.

```
my.CP.function <- function(x, n, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # 1 - alpha = confidence level
  Lo <- function(p){pbinom(x - 1, n, p) - (1-alpha/2)}
  Up <- function(p){pbinom(x, n, p) - alpha/2}
  Low <- uniroot(Lo, c(.001, .999))
  low.ci.f <- floor(Low$root*1000) / 1000
  upper <- uniroot(Up, c(.001, .999))
  upper.ci.c <- ceiling(upper$root * 1000) / 1000
  return(c(lower=low.ci.f, upper=upper.ci.c))
}
```

Wilson confidence intervals

To obtain confidence intervals according to (Wilson, 1927) one starts by first calculating A, B and C. Here x represents the number of successes out of n Bernoulli trials and $\hat{p} = x/n$ is the proportion of successes.

$$A = 2x + z_{1-\frac{\alpha}{2}}^2 \quad (2.10)$$

$$B = z_{1-\frac{\alpha}{2}} \sqrt{z_{1-\frac{\alpha}{2}}^2 + 4xq} \quad (2.11)$$

$$C = 2(n + z_{1-\frac{\alpha}{2}}^2) \quad (2.12)$$

The $(1 - \alpha) \cdot 100\%$ CI(p) is then given by:

$$\frac{A - B}{C} \quad to \quad \frac{A + B}{C} \quad (2.13)$$

Jeffreys Bayes methods

An alternative to the classical approach is the Bayesian one. The underlying idea is to combine the information contained in the actual data, the likelihood, with prior information or belief to obtain a posterior describing the parameter of interest.

Based on the conditional probability law, the probability of Y given θ is equal to the joint probability of Y and θ divided by the probability of θ :

$$P(Y|\theta) = \frac{P(Y, \theta)}{P(\theta)} \quad (2.14)$$

According to Bayes rule we obtain the posterior distribution $f(\theta|y)$ given the likelihood $f(y|\theta)$ and the prior distribution $f(\theta)$:

$$f(\theta|y) = \frac{f(y|\theta)f(\theta)}{\int f(y|\theta)f(\theta)d\theta} \quad (2.15)$$

According to (Held and Sabanés Bové, 2014, p.170), the denominator can be written as

$$\int f(y|\theta)f(\theta)d\theta = \int f(y, \theta)d\theta = f(y) \quad (2.16)$$

When $f(y)$, the marginal likelihood is omitted, the posterior distribution is proportional, with proportionality constant $\frac{1}{f(y)}$, to the product of the likelihood $f(y|\theta)$ and the density of the prior distribution $f(\theta)$.

To calculate the posterior distribution, the concept of conjugacy is important. Choosing as prior distribution a member of a specific family of distributions such that the posterior distribution belongs to the same family is called a *conjugate prior distribution* (Held and Sabanés Bové, 2014, p.179). For a binomial distribution, a beta distribution is a good choice for a conjugate prior. A beta distribution has also a support ranging from 0 to 1, similar to the binomial distribution.

In the case of a binomial density function with

$$f(x|\theta) = \binom{n}{x} \theta^x (1 - \theta)^{n-x} \quad (2.17)$$

and a beta prior distribution $\theta \sim Be(a, b)$ with suitably chosen parameters $a, b > 1$,

$$f(\theta) = \frac{1}{B(a, b)} \theta^{a-1} (1 - \theta)^{b-1} \quad (2.18)$$

assuming that the posterior distribution is proportional to the product of the likelihood times the prior distribution yields:

$$f(\theta|x) \propto \theta^x (1 - \theta)^{n-x} \theta^{a-1} (1 - \theta)^{b-1} \quad (2.19)$$

which can be simplified to

$$= \theta^{a+x-1} (1 - \theta)^{b+n-x-1} \quad (2.20)$$

In (Meeker and Hahn, 2017, p. 107), Jeffreys method, a Bayesian approach is proposed to construct a confidence or strictly speaking a credible interval for a proportion. It was shown by (Brown *et al.*, 2001) that the interval based on Jeffreys Bayesian approach is always contained in the exact confidence intervals based on the Clopper-Pearson approach. This finding highlights that Jeffreys approach is preferable because it is less conservative.

Here, the following formula for a two-sided approximate $100(1 - \alpha)\%$ binomial credible interval for p is proposed based on the shape parameters a and b of the beta distribution. Both are set to 0.5, thus considered to be “un-informative”, as so-called “Jeffreys priors”. For equi-tailed credible intervals, the corresponding $\frac{\alpha}{2}$ quantiles of the beta distribution are considered. For an applied

example, the prior $(\text{beta}(0.5, 0.5))$, the likelihood and the posterior distributions, as well as the limits of the credible interval are displayed in Figures 2.4.

$$\mathbb{P}[\text{qbeta}(\frac{\alpha}{2}; x + a, n - x + b), \text{qbeta}(1 - \frac{\alpha}{2}; x + a, n - x + b)] = 1 - \alpha \quad (2.21)$$

According to (Brown *et al.*, 2001) both the Wilson and the Jeffreys prior interval display excellence performance in terms of coverage probability. He considers the Jeffreys prior interval slightly superior with an average coverage being very close to the nominal level even for quite small n .

The following function implements Jeffreys approach for a binomial credible interval in R with the arguments of the data (x successes out of n trials) and the chosen credible level $(1 - \alpha)$.

```
my.CI.Jeffrey.function <- function(x, n, alpha){
# x = number of successes of an observed sample
# n = number of trials in observed sample
# 1 - alpha = credible level
a <- x + 0.5
b <- n - x + 0.5
low.ci <- qbeta(alpha/2, a, b)
up.ci <- qbeta((1-(alpha/2)), a, b)
low.ci.f <- floor(low.ci*1000) /1000
up.ci.c <- ceiling(up.ci*1000) /1000
return(c(lower=low.ci.f, upper=up.ci.c))
}
```

R packages

In the following, the results of these functions will be compared with the functions *binconf()* in the *Hmisc* package and the *binom.test()* in the *stats* package which uses a procedure first given in (Clopper and Pearson, 1934).

2.2.1 Example of applicability

One example is obtained from (Sprick *et al.*, 2017) with the objective to evaluate the damage inflicted by an unshod hoof and by the horseshoe material steel on the long bones of horses, i.e. radii and tibiae after a simulated kick. An impactor with a hoof horn or a steel head was dropped onto 16 prepared bones with a velocity of 8 m/s. The proportion of a long bone fracture was assessed and confidence or credible intervals were obtained. In the case of steel 12 out of 16 bones displayed a fracture, and 2 out of 16 bones with hoof horn.

Method/Package	Function	lower bound	upper bound
Wald	my.CI.Wald.function	0.537	0.963
Exact	my.CP.function	0.476	0.928
stats	binom.test()	0.476	0.927
Hmisc	binconf()	0.505	0.898
Jeffrey credible	my.CI.Jeffrey.function	0.509	0.91

Table 2.2: 95% CI (p) for steel

Method/Package	Function	lower bound	upper bound
Wald	my.CI.Wald.function	-0.038	0.288
Exact	my.CP.function	0.015	0.384
stats	binom.test()	0.016	0.383
Hmisc	binconf()	0.035	0.36
Jeffrey credible	my.CI.Jeffrey.function	0.026	0.345

Table 2.3: 95% CI (p) for horn**Interpretation of confidence intervals**

For repeated i.e. independent, identical realisations of the kick experiment with a steel impactor at a velocity of 8 m/s, the interval [47 to 93]% will contain the (unknown) true probability of a fracture in 95% of the cases ($n=16$) (see Table 2.2). Similarly, for an impactor with horn the interval [1 to 39] will contain the (unknown) true probability of a fracture in 95% of the cases ($n=16$) (see Table 2.3). The limitations of the Wald approach, relying on a normal approximation becomes evident, when looking at the lower bound, being negative which is impossible for a proportion.

Interpretation of credible intervals

In contrast, in a Bayesian approach, i.e. Jeffreys, the probability of a fracture lies between [50.9 to 91] for an impactor with steel and between [2.6 to 34.5] with horn with probability 95%.

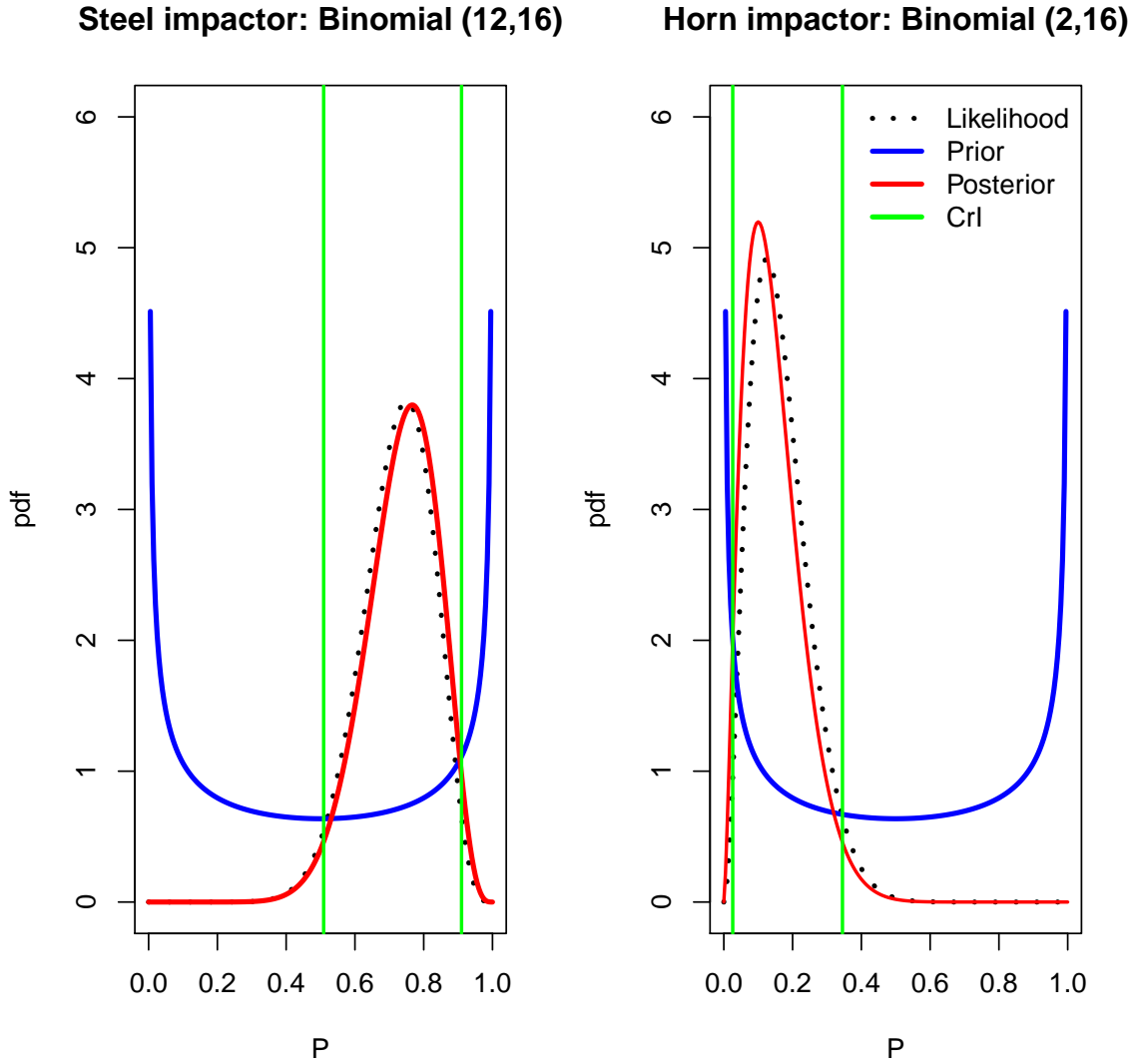


Figure 2.4: Density plots of the posterior distributions based the prior ($\text{Beta}(0.5, 0.5)$) and the data ($\text{Bin}(12,16)$ for a steel and $\text{Bin}(2,16)$) for an horn impactor in the example from (Sprick *et al.*, 2017). This prior distribution (blue) has an equal probability at 0 and 1, therefore considered to be un-informative. The total weight of Jeffreys prior for $a=0.5$ and $b=0.5$ equals to one observation. The likelihood (dotted black) and the posterior distribution (red) are similar. The $(1-\alpha = 0.95)$ credible intervals is presented green.

Chapter 3

Prediction intervals

Prediction intervals are appropriate to predict a future mean or another value - or as a specific case - a future single observation based on already observed (known) measurements. In the classical approach, a prediction interval for a single predicted future value will cover the true value in 95% of identical repetitions of the experiment. In a Bayesian approach, the unknown predicted value lies in a predicted interval with a 95% probability.

3.1 Prediction intervals for a normal distribution

Assume there are x_1, \dots, x_n realisations from $X \stackrel{iid}{\sim} N(\mu, \sigma^2)$. The aim is to predict a mean $\bar{y} = \bar{y}_{new}$ of a future sample y_1, \dots, y_m originating from independent additional realisations of a random normal variable $Y \stackrel{iid}{\sim} N(\mu, \sigma^2)$. Assume that σ^2 is known and μ is estimated by $\bar{x} = \bar{x}_{old}$ of an already observed sample x_1, \dots, x_n independent of y_i 's. In this situation, the two variances $Var(\bar{X})$ and $Var(\bar{Y})$ need to be taken into consideration. While $Var(\bar{X})$ is also relevant in the construction of a confidence interval, for the construction of prediction intervals $Var(\bar{Y})$ is also needed.

With

$$Var(\bar{X}) = \frac{1}{n}\sigma^2 \quad (3.1)$$

and

$$Var(\bar{Y}) = \frac{1}{m}\sigma^2 \quad (3.2)$$

it can be shown that, if \bar{X} and \bar{Y} are independent, then (see [Appendix 7.3](#))

$$Var(\bar{Y} - \bar{X}) = Var(\bar{Y}) + Var(\bar{X}) = \sigma^2 \left(\frac{1}{m} + \frac{1}{n} \right) \quad (3.3)$$

and

$$SE(\bar{y}_{new} - \bar{x}_{old}) = \hat{\sigma} \sqrt{\frac{1}{m} + \frac{1}{n}} \quad (3.4)$$

Then, approximately, for the new predicted mean \bar{y}_{new} and the available sample mean \bar{x}_{old} ,

$$\frac{\bar{y}_{new} - \bar{x}_{old}}{\hat{\sigma} \sqrt{\frac{1}{n} + \frac{1}{m}}} \sim N(0, 1) \quad (3.5)$$

Thus, similar to steps in equations (1.7) to (1.9), gives a prediction interval for the mean \hat{y}_{new} of m future observations based on an already observed sample x_1, \dots, x_n :

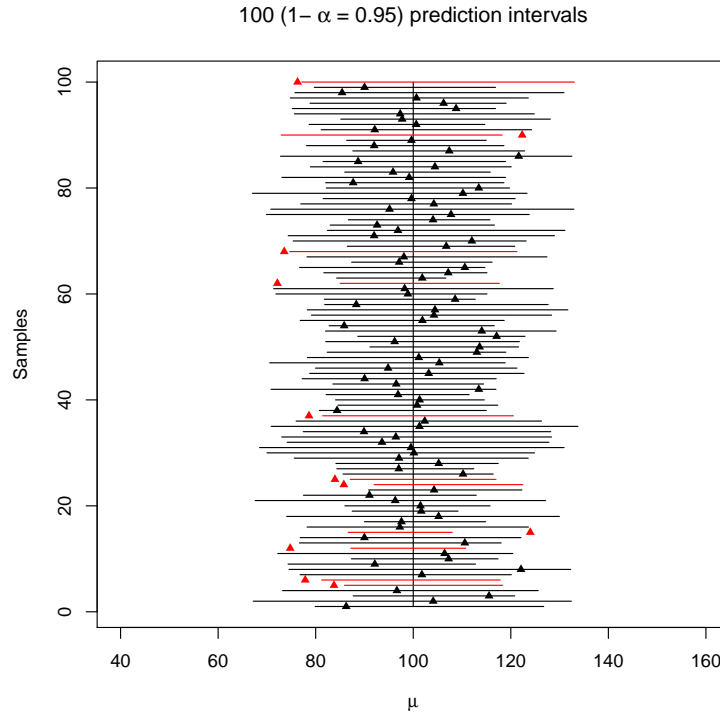


Figure 3.1: $(1 - \alpha = 0.95)$ prediction intervals for a single future value ($m = 1$), based on 100 random samples each with ($n = 10$), a true mean of 100 and a standard deviation of 10. The prediction intervals who do not cover the y_{new} , a newly generated random variable drawn from a normal distribution with a true mean of 100 and a standard deviation of 10, are colored in red (here 11 out of 100)

$$\mathbb{P}\left[\bar{x}_{old} - z_{1-\frac{\alpha}{2}} \cdot SE(\bar{y}_{new} - \bar{x}_{old}) \leq \bar{y}_{new} \leq \bar{x}_{old} + z_{1-\frac{\alpha}{2}} \cdot SE(\bar{y}_{new} - \bar{x}_{old})\right] = 1 - \alpha \quad (3.6)$$

This corresponds to equation (1.1), for $SE(\widehat{parameter}) = \hat{\sigma} \sqrt{\frac{1}{m} + \frac{1}{n}}$ and $k = z_{1-\alpha}$.

If the aim is to predict a single future observation, then $m = 1$ and the prediction interval for a single future observation y_{new} is

$$\mathbb{P}\left[\bar{x}_{old} - z_{1-\frac{\alpha}{2}} \cdot \hat{\sigma} \sqrt{1 + \frac{1}{n}} \leq y_{new} \leq \bar{x}_{old} + z_{1-\frac{\alpha}{2}} \cdot \hat{\sigma} \sqrt{1 + \frac{1}{n}}\right] = 1 - \alpha \quad (3.7)$$

3.1.1 Example of applicability

Based on equation (3.7), the following function can be implemented in R with the arguments of the already observed sample (mean, standard deviation, sample size), the size of the future sample and the chosen prediction level $(1 - \alpha)$.

```
my.PI.norm.function <- function(x, s, n, m, alpha){
  # x = mean of observed sample
  # s = standard deviation of observed sample
  # n = sample size of observed sample
  # m = sample size of future observations
```

```
# 1 - alpha = prediction level
SE <- s*sqrt((1/n) + (1/m))
low.ci <- x - SE * qnorm((1-alpha/2),0,1)
up.ci <- x + SE * qnorm((1-alpha/2),0,1)
low.ci.f <- floor(low.ci*100) /100
up.ci.c <- ceiling(up.ci*100) /100
return(c(lower=low.ci.f, upper=up.ci.c))
}
```

Subsequently this function is used to obtain a predicted value of the percent weight loss in one overweight dog (see section 2.1.1) and compared to the function `predict.lm()` in the `stats` package.

Function name	lower bound	upper bound
my.PI.norm.function	-0.11	22.84
predict.lm	-0.12	22.85

Table 3.1: $(1 - \alpha = 0.95)$ $PI(y_{new})$ for the percentage of weight loss in one overweight dog.

Interpretation

Based on the available sample of 926 overweight dogs, for independent and identical repetitions of the sampling experiment, y_{new} , the body weight loss in percent of a single future dog, will be covered by $[-0.11$ to $22.84]$ in 95 out of 100 cases.

3.2 Prediction intervals for a binomial distribution

We have an underlying $Be(p)$ process of iid generated successes. Similar to confidence intervals, for prediction intervals of a sample following a binomial distribution, a normal distribution approximation exists. If \hat{p} corresponds to the proportion in an available sample of size n , thus $n \cdot \hat{p}_{old}$ would be the number of successes in a Bernoulli trial, $m \cdot p_{new}$ would be the number of additionally successes generated by an independent Bernoulli trial of size m . Assuming that,

$$\hat{p}_{new} = \frac{1}{m} \sum_{i=1}^m I_i \quad (3.8)$$

$$Var(m\hat{p}_{new}) = Var\left(\sum_{i=1}^m I_i\right) = m\hat{p}_{new}(1 - \hat{p}_{new}) \quad (3.9)$$

If

$$Var(m\hat{p}_{new} - n\hat{p}_{old}) = Var(m\hat{p}_{new}) + Var(n\hat{p}_{old}) \quad (3.10)$$

then for $p_{new} = p_{old} = p$

$$mp(1 - p) + np(1 - p) = (m + n)p(1 - p) \quad (3.11)$$

Similar to equations (3.5) and (3.6),

$$\frac{m\hat{p}_{new} - n\hat{p}_{old}}{\sqrt{Var(m\hat{p}_{new} - n\hat{p}_{old})}} \sim N(0, 1) \quad (3.12)$$

and

$$\frac{m\hat{p} - n\hat{p}}{\sqrt{(m+n)p(1-p)}} \sim N(0,1) \quad (3.13)$$

A binomial prediction interval for the number of Bernoulli successes mp_{new} , based on the normal approximation, can be constructed as follows (Meeker and Hahn, 2017, p.122):

$$np_{old} - z_{1-\frac{\alpha}{2}} \sqrt{(m+n)p_{old}(1-p_{old})} \leq mp_{new} \leq np_{old} + z_{1-\frac{\alpha}{2}} \sqrt{(m+n)p_{old}(1-p_{old})} \quad (3.14)$$

Based on equation (3.14) the following R function was developed with the arguments of the available data (x successes out of n trials), the sample size m of the future sample and the chosen prediction level $(1 - \alpha)$.

```
my.PI.bin.function <- function(x, n, m, alpha){
# x = number of successes of an observed sample
# n = number of trials in observed sample
# m = sample size of future observations
# 1 - alpha = prediction level
p <- x/n
quant <- qnorm((1-alpha/2),0,1)
SE <- sqrt((m+n)*p*(1-p))
low.ci <- m*p - quant*SE
up.ci <- m*p + quant*SE
low.ci.f <- floor(low.ci)
up.ci.c <- ceiling(up.ci)
return(c(lower=low.ci.f, upper=up.ci.c))
}
```

Similar to relying on the normal approximation for binomial confidence intervals, where Bayesian credible intervals are preferable to classical approaches, Bayesian approaches exists also for the purpose of prediction. The underlying idea of predictive distributions (prior and post) is explained as follows.

3.2.1 Prior and posterior predictive distributions

Conceptionally, the prior predictive distribution of the data corresponds to what we expect before we actually see the data. The probability distribution $f(y_n)$, with y_n being the new expected data, is a marginal probability. It is possible to obtain it by integrating out all dependence on θ of the joint probability of y_n and θ .

$$f(y_n) = \int_0^1 f(y_n, \theta) d\theta \quad (3.15)$$

This can be rewritten using the conditional probability law, with the probability of y given θ is equal to the joint probability of y and θ divided by the probability of θ . With

$$P(Y|\theta) = \frac{P(Y, \theta)}{P(\theta)} \quad (3.16)$$

the prior predictive distribution is equal to the product of the likelihood $f(y_n|\theta)$ times the prior $P(\theta)$ and integrating it out over all parameter choices.

$$f(y_n) = \int_0^1 f(y_n|\theta) f(\theta) d\theta \quad (3.17)$$

In the context of a binomial distribution, with conjugate beta priors, this can be expressed as

$$f(y_n) = \int_0^1 \binom{n_n}{y_n} \theta^{y_n} (1 - \theta)^{n_n - y_n} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1} (1 - \theta)^{b-1} d\theta \quad (3.18)$$

rearranged into

$$= \binom{n_n}{y_n} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{\Gamma(a+y_n)\Gamma(b+n_n-y_n)}{\Gamma(a+b+n_n)} \int_0^1 \frac{\Gamma(a+b+n_n) \theta^{a+y_n-1} (1-\theta)^{b+n_n-y_n-1}}{\Gamma(a+y_n)\Gamma(b+n_n-y_n)} d\theta \quad (3.19)$$

and simplified to the prior predictive distribution

$$f(y_n) = \binom{n_n}{y_n} \frac{\text{Beta}(a+y_n, b+n_n-y_n)}{\text{Beta}(a, b)} \quad (3.20)$$

which can be directly implemented into R to calculate a prior predictive distribution as shown in Figure 3.2.

```
bin.pri.pred <- function(a,b,nn,yn){
# a = shape parameter a of a beta distribution
# b = shape parameter b of a beta distribution
# nn = sample size of new, not yet observed data
# yn = successes in new, not yet observed data
  return(exp(log(choose(nn, yn))+log(beta(a+yn,b+nn-yn))-log(beta(a,b))))
}
```

Conceptionally, once data have been observed, the posterior predictive distribution can be calculated. The question is, what value of new data y_n we would expect to obtain if the experiment would be repeated after old data y_o from a previous experiment have already been observed. Thus, with $f(y_n|y_o)$ being considered as a marginal probability which we could get from integrating out the joint probability of y_n and θ :

$$f(y_n|y_o) = \int_0^1 f(y_n, \theta|y_o) d\theta \quad (3.21)$$

Based on Bayes rule of conditional probability, this can be rewritten as

$$\int_0^1 f(y_n|\theta, y_o) \cdot f(\theta|y_o) d\theta \quad (3.22)$$

Regarding the first factor of $f(y_n|\theta, y_o)$, when we condition on θ , the new observation y_n is independent of y_o . Therefore,

$$\int_0^1 f(y_n|\theta) \cdot f(\theta|y_o) d\theta \quad (3.23)$$

The second factor, corresponds to the posterior distribution from the previous experiment. The first factor, $f(y_n|\theta)$ is the likelihood. After integrating out over all parameter choices we can obtain the posterior predictive distribution. When implementing this into R, the trick consists of updating the initially chosen a and b with the already observed data (y_o, n_o) .

$$a = a_o + y_o \quad (3.24)$$

and

$$b = b_o + n_o - y_o \quad (3.25)$$

In the context of a binomial distribution, the posterior predictive distribution can be expressed as

$$f(y_n|y_o) = \binom{n_n}{y_n} \frac{\Gamma(a_o + b_o + n_o)}{\Gamma(a_o + y_o)(b_o + n_o - y_o)} \frac{\Gamma(a_o + y_o + y_n)\Gamma(b_o + n_o - y_o + n_n - y_n)}{\Gamma(a_o + b_o + n_o + n_n)} \quad (3.26)$$

and simplified into

$$f(y_n|y_o) = \binom{n_n}{y_n} \frac{\text{Beta}(a_o + y_o + y_n, b_o + n_o - y_o + n_n - y_n)}{\text{Beta}(a_o + y_o, b_o + n_o - y_o)} \quad (3.27)$$

This can be directly implemented into R and is illustrated in Figure 3.2:

```
bin.post.pred <- function(a,b,nn,yn,no,yo){
# a = shape parameter a of a beta distribution
# b = shape parameter b of a beta distribution
# nn = sample size of new, not yet observed data
# yn = successes in new, not yet observed data
# no = sample size of old, already observed data
# yo = successes in old, already observed data
return(exp(log(choose(nn, yn))
            +log(beta(a+yo+yn,b+no-yo+nn-yn))
            -log(beta(a+yo,b+no-yo))))
}
```

Jeffreys Bayesian method for binomial prediction intervals

Jeffreys method can be viewed as a Bayesian binomial predictive distribution (Meeker and Hahn, 2017, p. 124), based on quantiles of the beta-binomial distribution considered as a Bayesian predictive distribution. It is an extension of the construction of a confidence interval, including also m , the size of the trials in a future Bernoulli experiment. Here the sample size parameter is n , the shape parameters a and b are set to 0.5 in Jeffreys approach. Similar to the posterior predictive distribution described above, the prior information is updated by the already observed data.

$$\mathbb{P}[qbetabinom(\frac{\alpha}{2}; m; x + a, n - x + b), qbetabinom(1 - \frac{\alpha}{2}; m; x + a, n - x + b)] = 1 - \alpha \quad (3.28)$$

```
dbetabinom <- function(x, size, a, b) {
  if (any(size <= 0))
    stop("size must be greater than 0")
  if (any(a <= 0))
    stop("a must be greater than 0")
  if (any(b <= 0))
    stop("b must be greater than 0")
  if (any(x < 0) || any(x > size))
    stop("x must be between 0 and size")
  exp(lbeta(x + a, size - x + b) - lbeta(a, b) +
      lchoose(size, x))
}
```

```

qbetabinom <- function(p, size, a, b) {
  if (any(size <= 0))
    stop("size must be greater than 0")
  if (any(a <= 0))
    stop("a must be greater than 0")
  if (any(b <= 0))
    stop("b must be greater than 0")
  if (any(p <= 0) || any(p >= 1))
    stop("p must be between 0 and 1")
  the.cumsum <- cumsum(dbetabinom(0:size, size, a, b))
  sapply(p, function(x) sum(the.cumsum < x))
}

```

Based on equation (3.11) for Jeffreys prediction interval the following R function was developed with the arguments of the available data (x successes out of n trials), the sample size m of the future sample and the chosen prediction level $(1 - \alpha)$.

```

my.Jeffrey.PI.function <- function(x, n, m, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # m = sample size of future observations
  # 1 - alpha = prediction level
  a <- x + 0.5
  b <- n - x + 0.5
  size <- m
  low.ci <- qbetabinom(alpha/2,size,a, b)
  up.ci <- qbetabinom((1-(alpha/2)),size,a, b)
  return(c(lower=low.ci, upper=up.ci))
}

```

3.2.2 Example of applicability

Applying the functions for prediction intervals based on the normal approximation and Jeffreys Bayesian approach to the bone fractures in the kick experiment with steel or horn impactor (2.2.1) gives the following results (predicting the number of fractures).

Impactor	lower bound	upper bound
steel	7	17
horn	-2	6

Table 3.2: $(1 - \alpha = 0.95)$ PI (mp_{new}) for number of fractures with steel and horn impactor based on a classical approach.

Impactor	lower bound	upper bound
steel	7	16
horn	0	7

Table 3.3: $(1 - \alpha = 0.95)$ PI (mp_{new}) for number of fractures with steel and horn impactor based on a Bayesian approach.

Interpretation

In a classical approach, based on the available sample of 16 kick experiments with either an steel or an horn impactor, for independent and identical repetitions of the sampling experiment, mp_{new} will be covered by [7 to 17] and by [-2 to 6] in 95 out of 100 cases for steel and horn respectively. For steel the larger possible bound with a possible maximum of 16 kicks and for horn the lower possible bound, as a minimum of 0 kicks, are not respected. This is attributed to the inadequate normal approximation used in the Wald approach.

In contrast, based on Jeffreys approach for a future sample with 16 bones being used in a kick experiment with either steel or horn, the number of predicted fractures would lie between 7 and 16 for steel and 0 and 7 for horn, respecting the possible boundaries.

The prediction intervals based on a normal approximation and Jeffreys approach are rather close for 16 future kicks with the steel impactor (see tables 3.2 and 3.3), for the steel impactor with a proportion p being close to 0, the normal approximation is not satisfactory, as the number of future predicted kicks cannot be negative. In general, the Bayesian prediction intervals are wider than the classical ones.

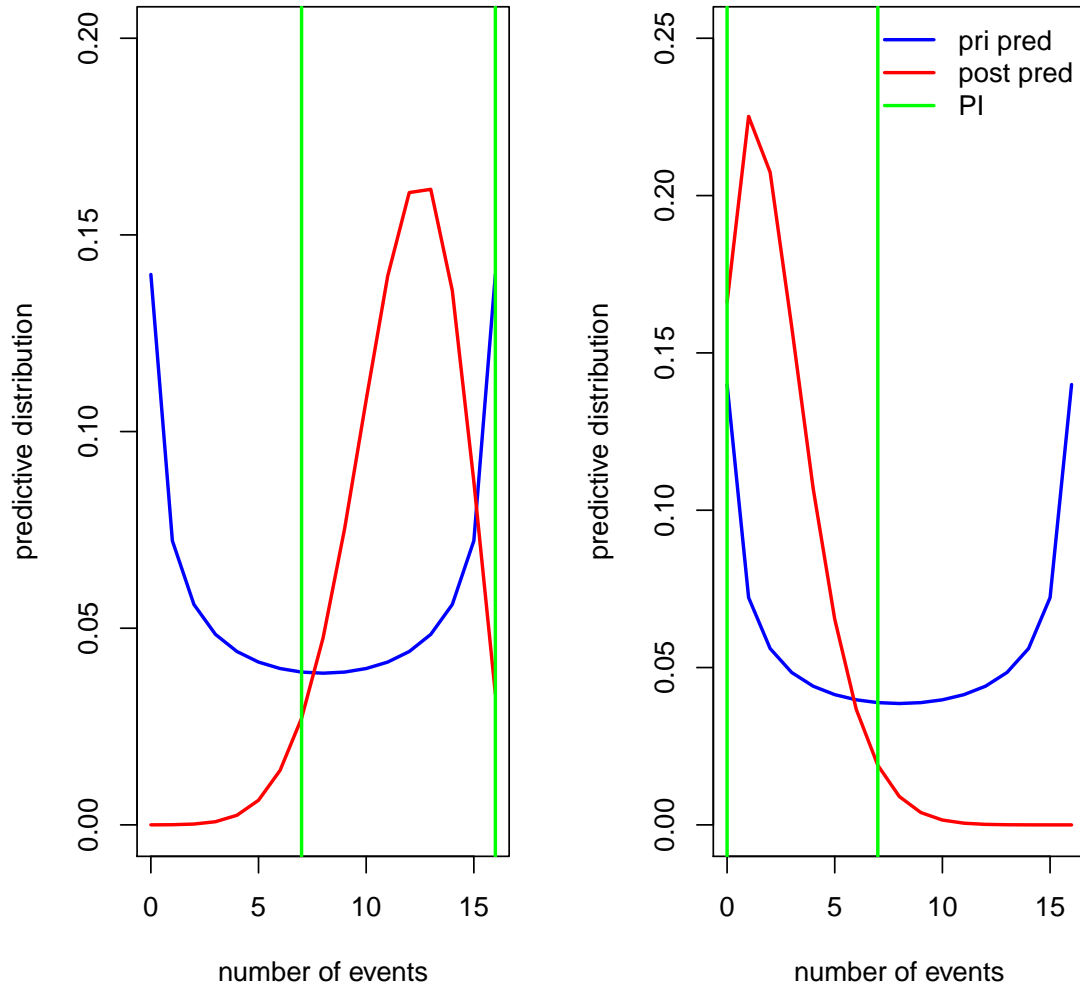


Figure 3.2: Density plots of the prior and posterior predictive distributions based the prior ($\text{Betabinomial}(0.5, 0.5)$) and the data ($\text{Bin}(12,16)$ for a steel and $\text{Bin}(2,16)$) for a horn impactor in the example from (Sprick *et al.*, 2017). This prior distribution has an equal probability at 0 and $n = 16$, therefore considered to be un-informative. The total weight of the prior, Jeffreys prior is $a = 0.5$ and $b = 0.5$ equals to one observation.

Chapter 4

Tolerance intervals

Additional to confidence and prediction intervals which have been presented so far, tolerance intervals can be applied. In some contexts, tolerance intervals are more appropriate. In order to showcase it, consider the following example: In the study from (Flanagan *et al.*, 2017) to assess factors influencing the weight loss in overweight dogs, the authors describe their primary outcome as percentage weight loss with $11.84 \pm 5.84\%$ (indicating the *mean* \pm *SD*). A 95% confidence interval, designed to describe the value of a parameter of interest - with some uncertainty - of the percentage weight loss would lie between (10.99 to 11.75). But this would tell very little about the weight loss in an individual dog. When aiming to predict the percentage weight loss in 1 or in 100 dogs, the corresponding 95% prediction intervals would lie between [-0.11 to 22.84] and [10.16 to 12.58]. Still for counseling, these two types of intervals, are not appropriate, since they are heavily influenced by the sample size. In the Flanagan study it would be more informative to describe an interval in the form of an upper and lower bound of a specified fraction of the studied dogs, i.e. 90 or 95%. This is a *tolerance interval*.

According to (Young, 2016), statistical tolerance intervals (TI) of the form $(1 - \alpha, P)$ provide bounds to capture at least a specified proportion P of the samples population with a given confidence level $(1 - \alpha)$. P is also called the *content* of the tolerance interval, and $(1 - \alpha)$ reflects the sampling variability in this classical approach. Different options, including non-parametric or distribution-free and parametric approaches to obtain tolerance intervals are possible. It is possible to construct two-sided tolerance intervals, bounded by an upper and a lower limit. In between these, a specified content or proportion is supposed to lie with a specified level of confidence. It is also possible to create one-sided tolerance intervals, with one single bound above or below a specified content is found with a defined confidence level.

According to (Young, 2010), for a continuous random variable X with a probability density function $f_X(\cdot; \theta)$ and a cumulative distribution function $F_X(\cdot; \theta)$ and θ a vector of parameters characterizing the distribution, then

$$C_X(L, U, \theta) = F_X(U; \theta) - F_X(L; \theta) \quad (4.1)$$

C_X is the coverage, i.e. with confidence $(1 - \alpha)$ of the two-sided interval with the lower bound L and the upper bound U . Thus,

$$\mathbb{P}[C_X(L, U, \theta) \geq P] \geq 1 - \alpha \quad (4.2)$$

4.1 Non-parametric tolerance intervals based on order statistics

Assume a random sample of ordered values ranging from $X_1 = x_1$, the lowest value to $X_n = x_n$, the highest value, with an underlying distribution function F_X which is a continuous, non-decreasing, probability function. The lower and upper non-parametric tolerance bounds L and U are

$$L = x_{(r)} \quad (4.3)$$

and

$$U = x_{(s)}. \quad (4.4)$$

with $x_{(j)}$ corresponding to the j -th value from the ordered sequence of the original x_1, \dots, x_n values.

In an example cited in (Young, 2010) related to the effects of a new drug on hours of sleep, a researcher wishes to claim with a specified confidence level $(1 - \alpha)$ that a proportion P of the subjects get sleep above a certain number of hours. The function `nptol.int()` in the package **tolerance** allows to evaluate a $(1 - \alpha = 0.95, P = 0.9)$ lower non-parametric tolerance interval. In this example, data describing the duration of sleep is appropriately described by a logistic distribution.

```
set.seed(100)
sleep.hours <- rlogis(n = 20, location = 5, scale = 1)
nptol.int(x = sleep.hours, alpha = 0.05, P = 0.9, side = 1)

##   alpha    P 1-sided.lower 1-sided.upper
## 1  0.05 0.9          2.18245          7.013099
```

Thus, being 95% confident, at least 90% of the subjects would get at least 2.18 hours of sleep. Regarding two-sided tolerance intervals based on order statistics, an approach based on a beta distribution is available. Otherwise, an approximation not using the beta distribution is described by (Scheffé and Tukey, 1944).

4.2 Tolerance intervals for a normal distribution

Additional to the non-parametric, distribution-free approaches, there are also parametric approaches for one- and two-sided tolerance intervals based on the normal distribution. These involve so-called tolerance factors, which correspond to k in equation (1.1).

To estimate the lower, L and upper, U limits of two-sided tolerance intervals for a normal distribution, similar to equation (1.10) $L = \bar{x} - k \cdot \hat{\sigma}$ and $U = \bar{x} + k \cdot \hat{\sigma}$ (Young, 2010). Here, according to (Young, 2010) k accounts for the sampling errors in the mean \bar{x} , $\hat{\sigma}$ the estimated standard deviation, the confidence level $(1 - \alpha)$ and the proportion of interest P . Solely in the one-sided setting, there is an exact solution for k :

$$k = \frac{1}{\sqrt{n}} t_{n-1, 1-\alpha}^*(\sqrt{n}z \cdot z_{1-\frac{\alpha}{2}}) \quad (4.5)$$

with n being the sample size, $t_{d, 1-\alpha}^*(\delta)$ being the $(1 - \alpha)$ -th quantile of a non-central t distribution with d degrees of freedom and non-centrality parameter δ , and $z_{1-\frac{\alpha}{2}}$ being the $100(1 - \frac{\alpha}{2})$ corresponding percentile of the standard normal distribution.

Based on (Howe, 1969), k can be defined for a two-sided normal tolerance interval by

$$k = uvw \quad (4.6)$$

with n the sample size

$$u = z_{1-\frac{\alpha}{2}} \sqrt{1 + n^{-1}} \quad (4.7)$$

with $\chi_{d; \alpha}^2$ being the α -the quantile of a χ^2 distribution with d degrees of freedom

$$v = \sqrt{\frac{n-1}{\chi_{n-1;\alpha}^2}} \quad (4.8)$$

$$w = \sqrt{1 + \frac{n-3 - \chi_{n-1;\alpha}^2}{2(n+1)^2}} \quad (4.9)$$

4.2.1 Examples of applicability

Simulated data set

When generating a random sample of 100 observations from a normal distribution with a mean of 0 and a standard deviation of 0.2, the package **tolerance** with the functions (*norm.tol()*), (*nptol.int()*) and (*bayesnormtol.int()*) provides a tolerance interval based on the function arguments of the data (x), the chosen $(1 - \alpha)$ confidence level, the proportion P of the population to be covered, and the option of a one- or a two-sided tolerance interval. For the Bayesian approach, additionally information on hyperparameters is required.

Approach	lower bound	upper bound
parametric	-0.382	0.383
non-parametric	-0.454	0.379
Bayes	-0.44	0.441

Table 4.1: ($1 - \alpha = 0.95, P = 0.9$) TI based on a classical normal, a non parametric and a Bayesian approach.

Percentage weight loss in overweight dogs (Flanagan *et al.*, 2017)

Approach	lower bound	upper bound
parametric	1.36	21.38
non-parametric	1.79	21.84
Bayesian	1.42	21.34

Table 4.2: ($1 - \alpha = 0.95, P = 0.9$) classical normal, non-parametric and Bayesian tolerance intervals of the percentage body weight loss in overweight dogs.

Interpretation

Thus, for the percentage weight loss of 926 dogs described in (Flanagan *et al.*, 2017), for independent and identical repetitions of the experiment, 90% of the dogs would be covered by the TI of [1.36 to 21.38] and [1.79 to 21.84] for classical normal and non-parametric approaches in 95 out of 100 cases. From a Bayesian perspective, the tolerance interval spans [1.42 to 21.34].

4.3 Tolerance intervals for a binomial distribution

Tolerance intervals for a binomial random variable, with a probability mass function $f_X(\cdot; \theta)$ and a cumulative distribution function $F_X(\cdot; \theta)$ are described in (Young, 2010). The coverage C_x of the two-sided interval $[L, U]$ for a discrete random variable x , the number of successes in a Bernoulli

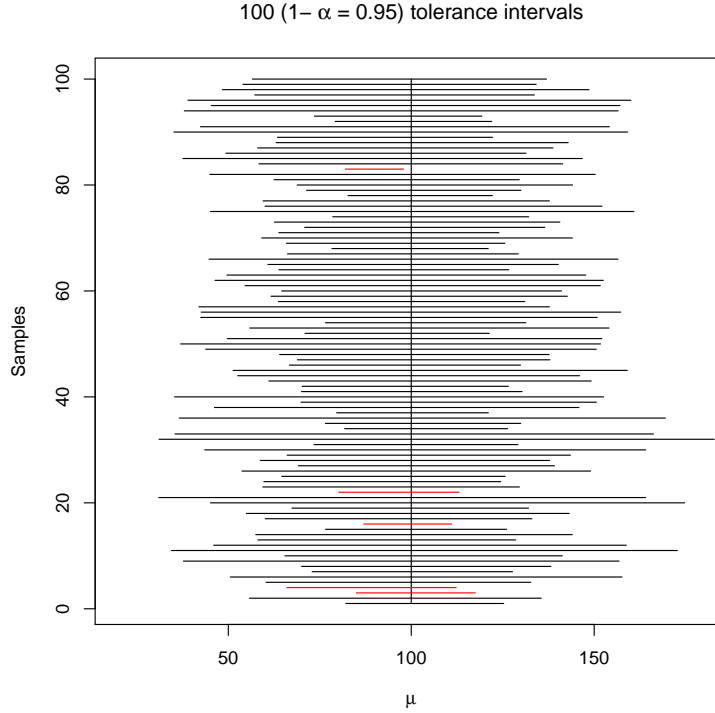


Figure 4.1: Illustration of $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals samples of size $n = 5$ drawn from a random normal distribution with a true mean of 100 and a sd of 10. The tolerance intervals which do not cover the chosen content of $P = 0.9$ colored in red (here 5 out of 100).

trial, when θ_L and θ_U are $100 \cdot (1 - \alpha)\%$ lower and upper confidence bounds, respectively, for the parameter θ , is given by

$$C_X(L, U; m, \theta_L, \theta_U) = F_X(U; m, \theta_U) - F_X(L; m, \theta_L) \quad (4.10)$$

In contrast to the situation of a normal distribution in equation (4.3), the coverage of a binomial tolerance, depends also on m , the future sample size.

One-sided $(1 - \alpha, P)$ lower or upper tolerance intervals, i.e. $[L, +\infty]$ and $[-\infty, U]$ for a future sample of size m , requires finding the largest integer L or the smallest integer U such that

$$\mathbb{P}\left[1 - F_X(L - 1; m; \theta_L) \geq P\right] \geq 1 - \alpha \quad (4.11)$$

and

$$\mathbb{P}\left[F_X(U; m; \theta_U) \geq P\right] \geq 1 - \alpha \quad (4.12)$$

For a two-sided interval

$$\mathbb{P}\left[C_X(L, U; m; \theta_L, \theta_U) \geq P\right] \geq 1 - \alpha \quad (4.13)$$

More specifically for a random binomial variable with a cumulative distribution function

$$F_X(x; n; p) = \sum_{i=0}^x \binom{n}{i} p^i (1 - p)^{(n-i)} \quad (4.14)$$

here $x = 0, \dots, n$ is the number of successes in a Bernoulli trial and $0 \leq p \leq 1$ is the proportion $p = \frac{x}{n}$. The tolerance limits are calculated using confidence bounds for p . A number of different approaches estimating the interval of a binomial proportion are discussed in (Brown *et al.*, 2001).

In the package `tolerance` the function (`bintol.int()`) provides one- and two-sided $(1 - \alpha, P)$ binomial tolerance intervals.

4.3.1 Examples of applicability

Simulated data set

In the example given in the `tolerance` package, for $x = 10$ successes in a Bernoulli trial of size $n = 1000$, a future sample of size $m = 50$, the $(1 - \alpha) = 95\%$ confidence and $P = 90\%$ tolerance intervals are shown.

```
#one-sided
bintol.int(x = 10, n = 1000, m = 50, alpha = 0.05, P = 0.9, side = 1)

##   alpha    P p.hat 1-sided.lower 1-sided.upper
## 1  0.05 0.9  0.01              0              2

#two-sided
bintol.int(x = 10, n = 1000, m = 50, alpha = 0.05, P = 0.9, side = 2)

##   alpha    P p.hat 2-sided.lower 2-sided.upper
## 1  0.05 0.9  0.01              0              2
```

A number of different methods to calculate the confidence bounds $\theta_L = p_L$ and $\theta_U = p_U$ is available: the large-sample method (“LS”) if $n \geq 50$ and $n\hat{p}$ and $n(1 - \hat{p})$ are both ≥ 10 . For smaller sample sizes ($n \leq 40$), when p is not too close to 0 or 1, Wilson’s method (“WS”) is appropriate. Another option includes the Clopper-Pearson or “exact” method. A Bayesian approach based on Jeffreys method based on a beta prior distribution for p is available.

Approach	method TI	lower bound
classical	Large sample	7
classical	Wilson	6
classical	Clopper Pearson	6
Bayesian	Bayesian	6

Table 4.3: One-sided $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals for minimum number of fractures due to the steel in 16 kick experiments.

Approach	method TI	lower bound
classical	Large sample	0
classical	Wilson	0
classical	Clopper Pearson	0
Bayesian	Bayesian	0

Table 4.4: One-sided $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals for the minimum number of fractures due to the horn impactor in 16 kick experiments.

Fractures in a kick experiment (see section 2.2.1)

Interpretation

Thus, from a classical perspective, with a 95% confidence, in 90% of the kick experiments with 16 kicks, there will be at least 7 fractures with steel (or no fractures with horn) for independent and identical repetitions of the experiment. Similar results are obtained from a Bayesian approach.

Chapter 5

Contrasting different types of intervals

In order to contrast confidence, prediction and tolerance intervals, we first replicate the examples suggested by (Gitlow and Awad, 2013). In this example three samples of simulated data with equal means and standard deviations are provided. In a second step we explore the asymptotic behaviour of the three interval types subject to growing sample size.

5.1 Simulated data sets: normality assumption

All three scenarios A, B and C are simulated to have the same mean (16) and standard deviation (12), but Figure 5.1 shows, that only scenario A is based on a random normal sample. Scenario B is designed to be bi-modal and scenario C to be skewed. Deviations from the normal distribution of scenarios B and C are also evident from Table 5.1 regarding skewness and kurtosis. Thus the later two scenarios are not normally distributed.

In Table 5.2 the $(1-\alpha = 0.95)$ confidence, $(1-\alpha = 0.95)$ prediction and the $(1-\alpha = 0.95, P = 0.9)$ tolerance intervals with the latter ones based on a normal distribution and on a non-parametric approach are shown. For the prediction intervals the size m of the predicted future observations was chosen to be equal to the original sample size $n = 300$.

Interpretation

The first three intervals, the confidence, prediction and tolerance intervals - obtained by the function *normtol.int()* in the **tolerance** package - are identical for the three scenarios. This result is not plausible as the three scenarios are clearly different. This is due to the underlying assumption of normality for the confidence, prediction and tolerance intervals (based on the function *normtol.int()*). Solely the non-parametric approach by the function *nptol.int()* allows to discern a difference between the three scenarios.

Table 5.1: Descriptive statistics of the three scenarios A, B and C

	A	B	C
n	300	300	300
mean	16	16	16
sd	12	12	12
min	-12.79	-3.78	4.46
max	47.18	35.54	81.41
skewness	0.04	0.13	1.99
kurtosis	2.74	1.32	8.11
IQR	16.15	23.05	12.32

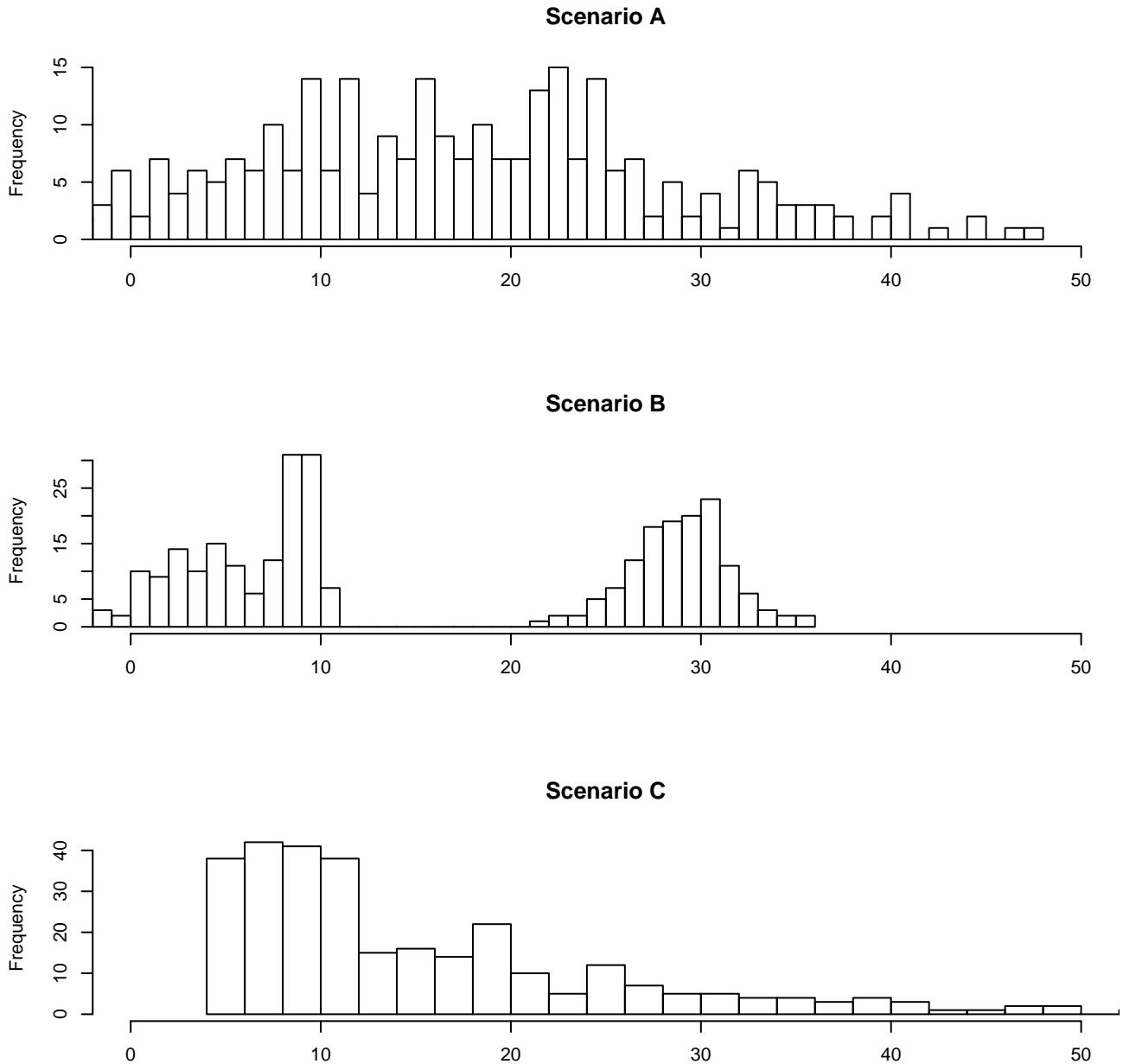


Figure 5.1: Histograms of simulated data sets: scenario A (normally distributed), B (bimodal distribution) and C (skewed distribution).

In conclusion, it is important to check the normality assumption. Otherwise, confidence, prediction and tolerance intervals might be misleading. Presenting the three types of intervals in Table 5.2 together helps also to highlight the differences between them. The confidence and the prediction intervals are of similar width - which might be due to choosing a sample size m of the predicted future observations being equal to the already observed sample of size n . In contrast, the width of the tolerance interval, for a proportion P of 0.9, is considerably larger than the width of the other two intervals.

Scenario	Function	Interval	P	lower bound	upper bound
A	my.CI.norm	CI		14.64	17.36
B	my.CI.norm	CI		14.64	17.36
C	my.CI.norm	CI		14.64	17.36
A	my.PI.norm	PI		14.07	17.93
B	my.PI.norm	PI		14.07	17.93
C	my.PI.norm	PI		14.07	17.93
A	normtol.int	TI	0.9	-5.21	37.21
B	normtol.int	TI	0.9	-5.21	37.21
C	normtol.int	TI	0.9	-5.21	37.21
A	nptol.int	TI	0.9	-6.19	39.01
B	nptol.int	TI	0.9	-0.22	32.61
C	nptol.int	TI	0.9	4.67	44.30

Table 5.2: Comparison of confidence, prediction and tolerance intervals for the three scenarios A, B and C with a $(1 - \alpha = 0.95)$ confidence and prediction level, and a $(1 - \alpha = 0.95, P = 0.9)$ tolerance interval. The parametric *normtol.int()* and non-parametric *nptol.int()* functions for the respective tolerance intervals originate from the **tolerance** package.

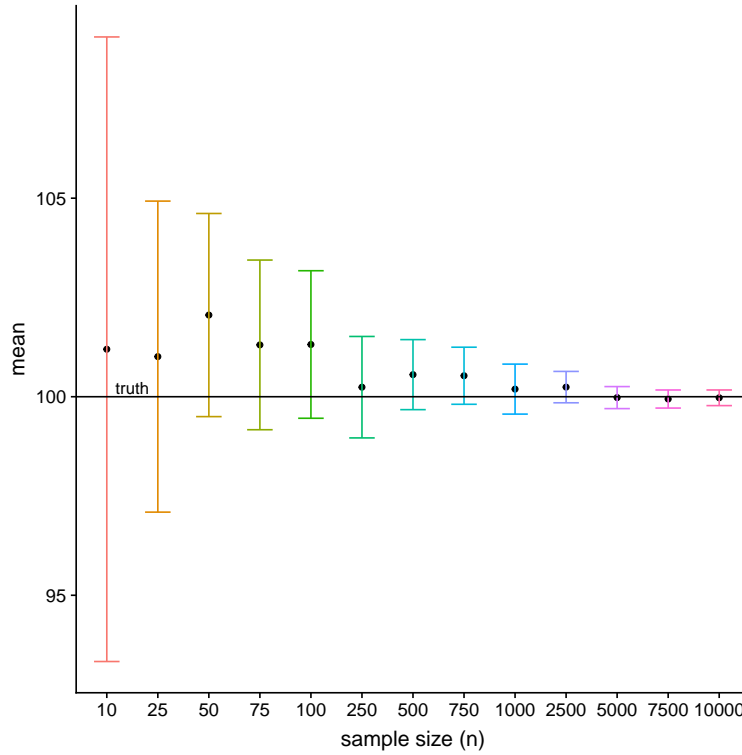


Figure 5.2: Illustration of the effect of sample size n on the width of $(1 - \alpha = 0.95)$ confidence intervals. The random sample was drawn from a normal distribution with a true mean of 100 (black line) and a standard deviation of 10 and subsequently subsetting to obtain samples ranging from $n = 10$ to $n = 10^4$.

5.2 Behaviour subject to increasing sample size: normal distribution

In order to assess the behaviour of the three intervals if the sample size n increases, a random sample $n = 10^4$ was drawn from a normal distribution with a true mean of 100 and a standard deviation of 10. Subsequently, subset samples ranging from $n = 10^1$ to $n = 10^4$ were taken.

5.2.1 Confidence intervals

For 95% confidence intervals, if the sample size n goes to infinity, the sampling error se will be close to 0 as $se = \frac{\hat{\sigma}}{\sqrt{n}}$. Subsequently, the width of the confidence interval

$$2 \cdot 1.96 \cdot \frac{\hat{\sigma}}{\sqrt{n}} \quad (5.1)$$

will be zero and the confidence interval approach the (true) mean value (see Figure 5.2).

5.2.2 Prediction intervals

Similar to the confidence intervals, the width of the prediction intervals decreases initially, but reaches a plateau earlier, i.e. after a sample size of 500 (see Figure 5.3). When n is increasing, the width for a $(1 - \alpha = 0.95)$ prediction interval for a single future observation ($m = 1$) is determined by referring to equation (3.4):

$$2 \cdot 1.96 \cdot \hat{\sigma} \sqrt{\frac{1}{1} + \frac{1}{n}} \quad (5.2)$$

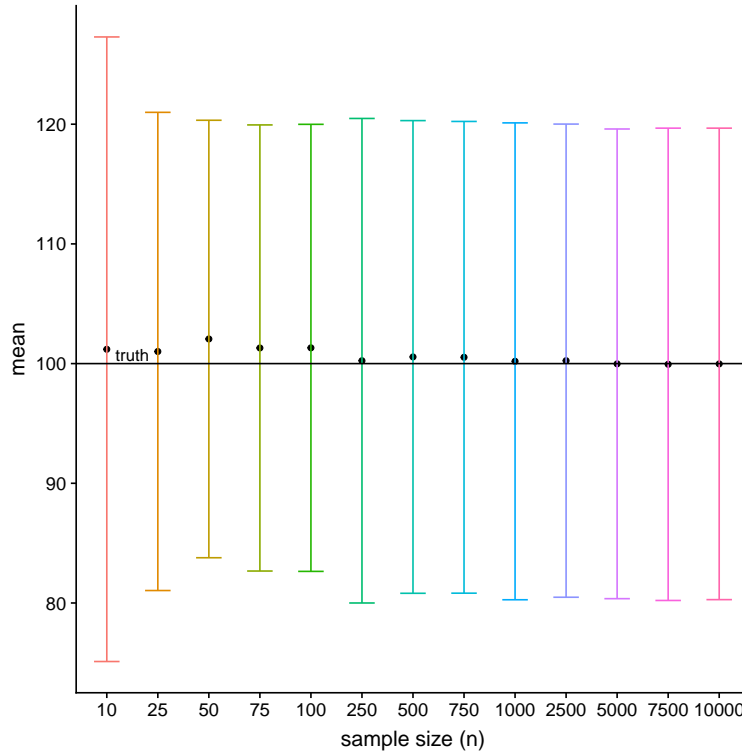


Figure 5.3: Illustration of the effect of sample size n on the width of $(1 - \alpha = 0.95)$ prediction intervals for a single future observation ($m = 1$). The random sample was drawn from a normal distribution with a true mean of 100 (black line) and a standard deviation of 10 and subsequently subsetting to obtain samples ranging from $n = 10$ to $n = 10^4$.

If n goes to infinity, then this becomes:

$$2 \cdot 1.96 \cdot \hat{\sigma} \quad (5.3)$$

Thus, for a chosen standard deviation σ of 10, the width of a $(1 - \alpha = 0.95)$ prediction interval will approach a minimum value of 39.2 or approximately four times the standard deviation. When, for a fixed $n = 100$, the sample size m of the future predicted samples is increased (see Figure 5.4), also a plateau is reached and the width $(1 - \alpha = 0.95)$ is determined by referring to equation (3.4):

$$2 \cdot 1.96 \cdot \hat{\sigma} \sqrt{\frac{1}{m} + \frac{1}{n}}$$

If m goes to infinity, for an already observed sample of size $n = 100$, the width of a $(1 - \alpha = 0.95)$ prediction interval will be

$$2 \cdot 1.96 \cdot \hat{\sigma} \sqrt{\frac{1}{100}} = 2 \cdot 1.96 \cdot 10 \cdot 0.1$$

Thus for an increasing number m of predicted future observations, with a $\sigma = 10$ and $n = 100$, the width of a $(1 - \alpha = 0.95)$ prediction interval will approach 3.92 or approximately four times the standard deviation divided by the \sqrt{n} .

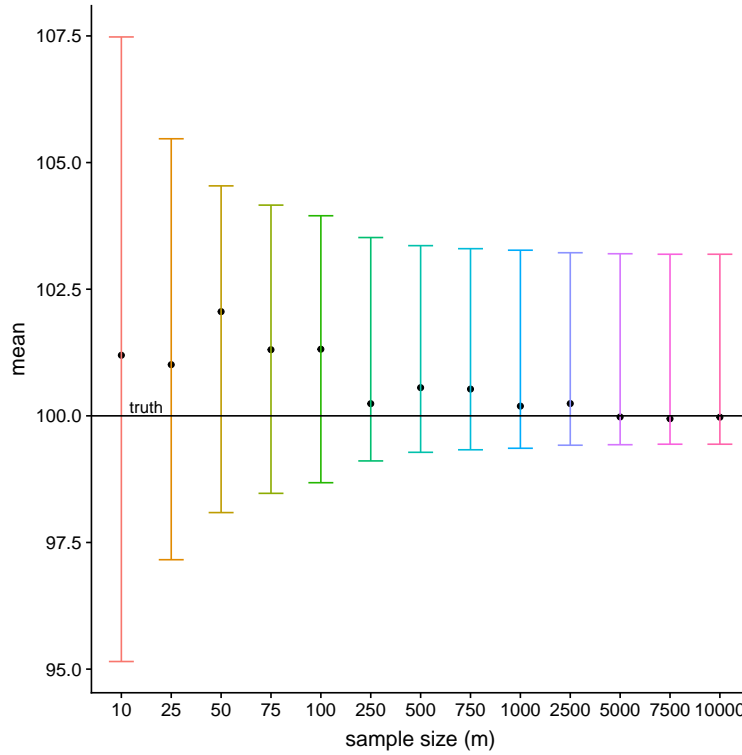


Figure 5.4: Illustration of the effect of sample size $m = 10$ to $m = 1000$ on the width of $(1 - \alpha = 0.95)$ prediction intervals based on an observed sample with a size of $n = 100$. The random sample was drawn from a normal distribution with a true mean of 100 (black line) and a standard deviation of 10.

5.2.3 Tolerance intervals

Empirically, based on Figure 5.5 the width of $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals for increasing n approaches a minimum width of 33.44. This is close to (Meeker and Hahn, 2017, p. 529), where the tabulated k factors for a tolerance interval containing a proportion of $P = 0.9$ is 1.64. Thus, when multiplying with 2 for the width and with 10 for σ , the minimum width of the tolerance intervals will approach $2 * 1.64 * 10 = 32.8$.

5.2.4 Comparison of CI, PI and TI

In Figure 5.6, the widths of all three types of intervals, subject to increasing sample size n are displayed. In Table 5.3, the $(1 - \alpha = 0.95)$ confidence, $(1 - \alpha = 0.95)$ prediction and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals for increasing sample size n ranging from 10 to 10000 are displayed. It becomes evident that the width of confidence intervals will be close to zero. This finding confirms that confidence, or credible intervals, are designed to describe a parameter with some uncertainty due to sampling errors. When the sample size going to infinity sampling error will vanish and there will be no uncertainty left.

While the width of both the prediction and the tolerance intervals is initially decreasing, the width will after sample sizes of approx. $n = 200$ or $n = 500$ reach a plateau. For the prediction of the location of a single $m = 1$ and increasing sample sizes n , the prediction intervals will be larger as predicting the intervals for increasing m based on a fixed sample size of $n = 100$. This is due to the fact that there is a lot of uncertainty where a single future observation could lie. In the second case, more information is already contained in the 100 already existing observation which will lead to a smaller width of the prediction interval.

For the chosen $(1 - \alpha = 0.95, P = 0.9)$ tolerance interval, the proportion P determines if the

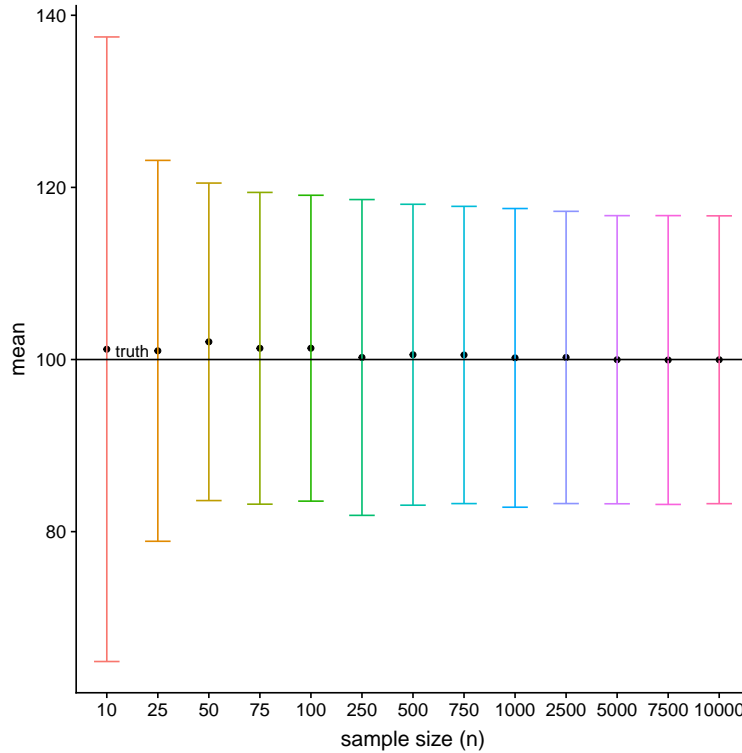


Figure 5.5: Illustration of the effect of sample size n on the width of $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals. The random sample was drawn from a normal distribution with a true mean of 100 (black line) and a standard deviation of 10 and subsequently subsetting to obtain samples ranging from $n = 10$ to $n = 10^4$.

width of the tolerance interval is larger than the width of the prediction interval or not. For all three intervals the width reduction is most important for the smaller sample sizes and then reaches a plateau.

sample size n	mean	sd	low ci	up ci	low pi	up pi	low ti	up ti
10	101.20	12.69	93.33	109.06	75.11	127.29	64.91	137.48
25	101.01	9.99	97.09	104.93	81.04	120.99	78.88	123.14
50	102.06	9.23	99.50	104.61	83.78	120.33	83.61	120.50
75	101.31	9.44	99.17	103.44	82.67	119.94	83.19	119.42
100	101.32	9.48	99.46	103.17	82.64	119.99	83.55	119.09
250	100.24	10.31	98.96	101.52	80.00	120.48	81.89	118.59
500	100.56	10.06	99.67	101.44	80.81	120.30	83.07	118.04
750	100.53	10.04	99.81	101.25	80.82	120.23	83.26	117.80
1000	100.19	10.16	99.56	100.82	80.27	120.11	82.84	117.55
2500	100.24	10.08	99.85	100.64	80.48	120.01	83.26	117.22
5000	99.98	10.01	99.70	100.26	80.36	119.60	83.24	116.72
7500	99.94	10.06	99.71	100.17	80.21	119.67	83.16	116.72
10000	99.97	10.05	99.78	100.17	80.28	119.67	83.25	116.69

Table 5.3: Comparison of $(1 - \alpha = 0.95)$ confidence intervals, $(1 - \alpha = 0.95)$ prediction intervals for one future observation ($m = 1$), and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals with increasing sample size n . The samples have been drawn from a normal distribution with a true mean of 100 and a standard deviation of 10 and subsequently subsetting to obtain samples ranging from $n = 10$ to $n = 10^4$.

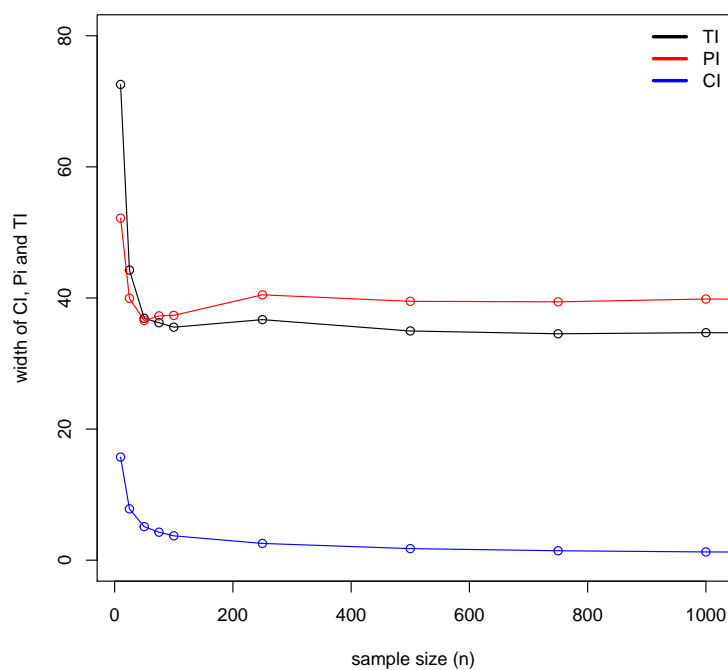


Figure 5.6: Illustration of the effect of sample size n on the width of $(1 - \alpha = 0.95)$ confidence intervals (blue), $(1 - \alpha = 0.95)$ prediction intervals for one future observation ($m = 1$) (red), and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals (black). The random sample was drawn from a normal distribution with a true mean of 100 and a standard deviation of 10 and subsequently subsetting to obtain samples ranging from $n = 10$ to $n = 10^4$.

5.3 Behaviour subject to increasing sample size: binomial distribution

In order to assess the behaviour of the three intervals, including Bayesian approaches, if the sample size n increases, random samples of size $n = 10^4$ with a true proportion of 0.5 and 0.1 were drawn from a binomial distribution. Subsequently, subset samples ranging from $n = 10$ to $n = 10^4$ were taken. Since the results for the prediction and tolerance intervals are given in the form of number of successes ($m * p$), in order to allow for a comparison, the resulting lower and upper limits were divided by the size ($n = m$) corresponding initial/future samples in order to obtain a proportion.

In Figures 5.7 and 5.8, the effect of increasing sample size n on $(1 - \alpha = 0.95)$ credible, $(1 - \alpha = 0.95)$ prediction intervals, a frequentist approach based on a normal approximation and a Bayesian approach based on Jeffreys, and on $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals for the true proportions of 0.5 and 0.1 are displayed. In Tables 5.4 to 5.7 the resulting intervals both in the form of a proportion but also in the form of number of (predicted) successes are presented.

n	x	Pr	low cri	up cri	low pi N	up pi N	low pi J	up pi J	low ti J	up ti J
10	3	0.30	0.09	0.61	-0.20	0.80	0.00	0.70	0.00	1.00
25	12	0.48	0.29	0.67	0.20	0.76	0.20	0.72	0.08	0.92
50	22	0.44	0.31	0.58	0.24	0.64	0.22	0.60	0.14	0.86
75	34	0.45	0.34	0.57	0.29	0.61	0.23	0.53	0.15	0.85
100	47	0.47	0.37	0.57	0.33	0.61	0.27	0.54	0.16	0.84
250	114	0.46	0.40	0.52	0.37	0.54	0.39	0.57	0.18	0.82
500	238	0.48	0.43	0.52	0.41	0.54	0.41	0.53	0.19	0.81
750	378	0.50	0.47	0.54	0.45	0.55	0.45	0.55	0.20	0.80
1000	496	0.50	0.46	0.53	0.45	0.54	0.46	0.55	0.20	0.80
2500	1256	0.50	0.48	0.52	0.47	0.53	0.48	0.54	0.21	0.79
5000	2483	0.50	0.48	0.51	0.48	0.52	0.48	0.52	0.21	0.79
7500	3717	0.50	0.48	0.51	0.48	0.51	0.49	0.52	0.22	0.78
10000	4998	0.50	0.49	0.51	0.49	0.51	0.49	0.51	0.22	0.78

Table 5.4: Comparison of $(1 - \alpha = 0.95)$ credible, $(1 - \alpha = 0.95)$ prediction and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals with increasing sample size $n(= m)$ for binomial samples with a true proportion of 0.5. n and x are the number of trials and successes, respectively. Pr is the estimated proportion, 'low and up cri' correspond to the credible intervals based on Jeffreys approach, 'low and up pi N' are the predictive intervals based on a normal approximation, 'low and up pi J' are predictive intervals based on Jeffreys approach and 'low and up ti J' are the tolerance intervals obtained by the function *bintol.int()* specifying Jeffreys approach of the **tolerance** package.

5.3.1 Comparison of CI, PI and TI

Similar to the case of the normal distribution, the width of the credible interval approaches 0 (see Figures 5.7 to 5.9 and Tables 5.4 to 5.7). Thus, also the credible intervals are best used to describe the location of a single parameter of interest. Regarding the classical and the Bayesian approach for the prediction intervals, it becomes evident that they yield similar results for larger sample sizes, approaching the true proportion (similar to the credible intervals). When looking at the number of future predicted successes, both from a classical and a Bayesian approach, the former ones consists of slightly larger intervals. The difference is exactly 1 observation less, presumably due to the prior information. Here to obtain the Bayesian predictive intervals, the formula of (Meeker and Hahn, 2017, p. 335) was used. A posterior predictive distribution in combination

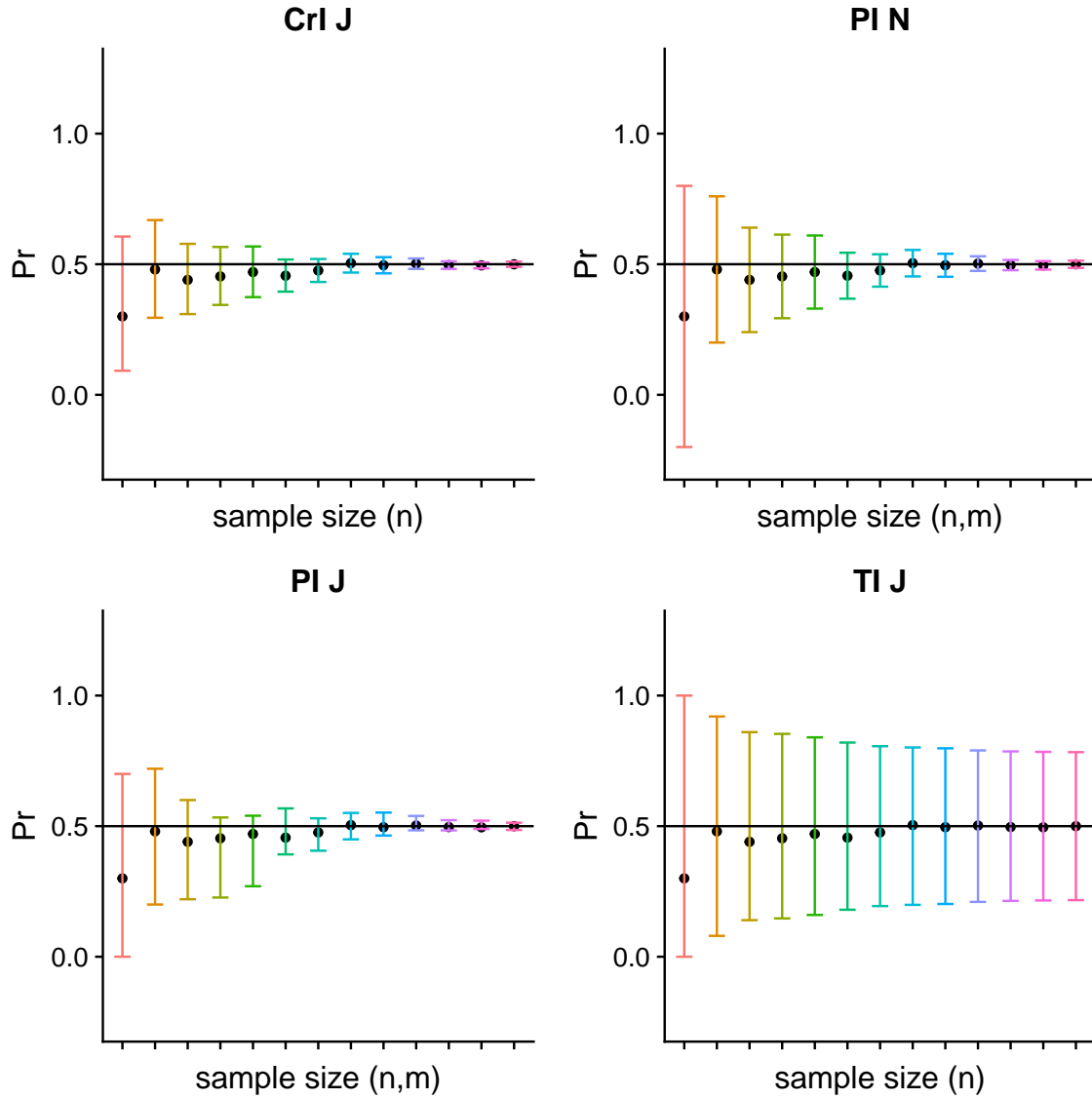


Figure 5.7: Different intervals for samples with a sample size n and size m of predicted observations ranging from 10 to 10000 drawn from a binomial distribution with true proportion of 0.5. CrI J: $(1 - \alpha = 0.95)$ credible intervals based on Jeffreys prior. PI N: $(1 - \alpha = 0.95)$ prediction intervals based on a normal approximation with $m = n$. PI J: $(1 - \alpha = 0.95)$ prediction intervals based on Jeffreys approach with $m = n$. TI J: $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals based on Jeffreys approach. In order to allow for comparisons, the prediction and tolerance intervals which predict the number of future successes $m * p$, the resulting intervals were divided by the corresponding sample size $m = n$, to obtain p . The black line indicates the true proportion.

with a prior predictive distribution, thus including prior information on already observed data, might further decrease the widths of the Bayesian prediction intervals. Furthermore, the lower limit of classical approach results in negative values up to a sample size n of 75. In contrast, the Bayesian approach is boundary respecting and therefore clearly preferable.

The width of the tolerance intervals is considerably larger, also for the number of successes $m * p$. Not surprisingly, the width of all confidence intervals is larger for a true proportion of 0.5, compared to a true proportion of 0.1.

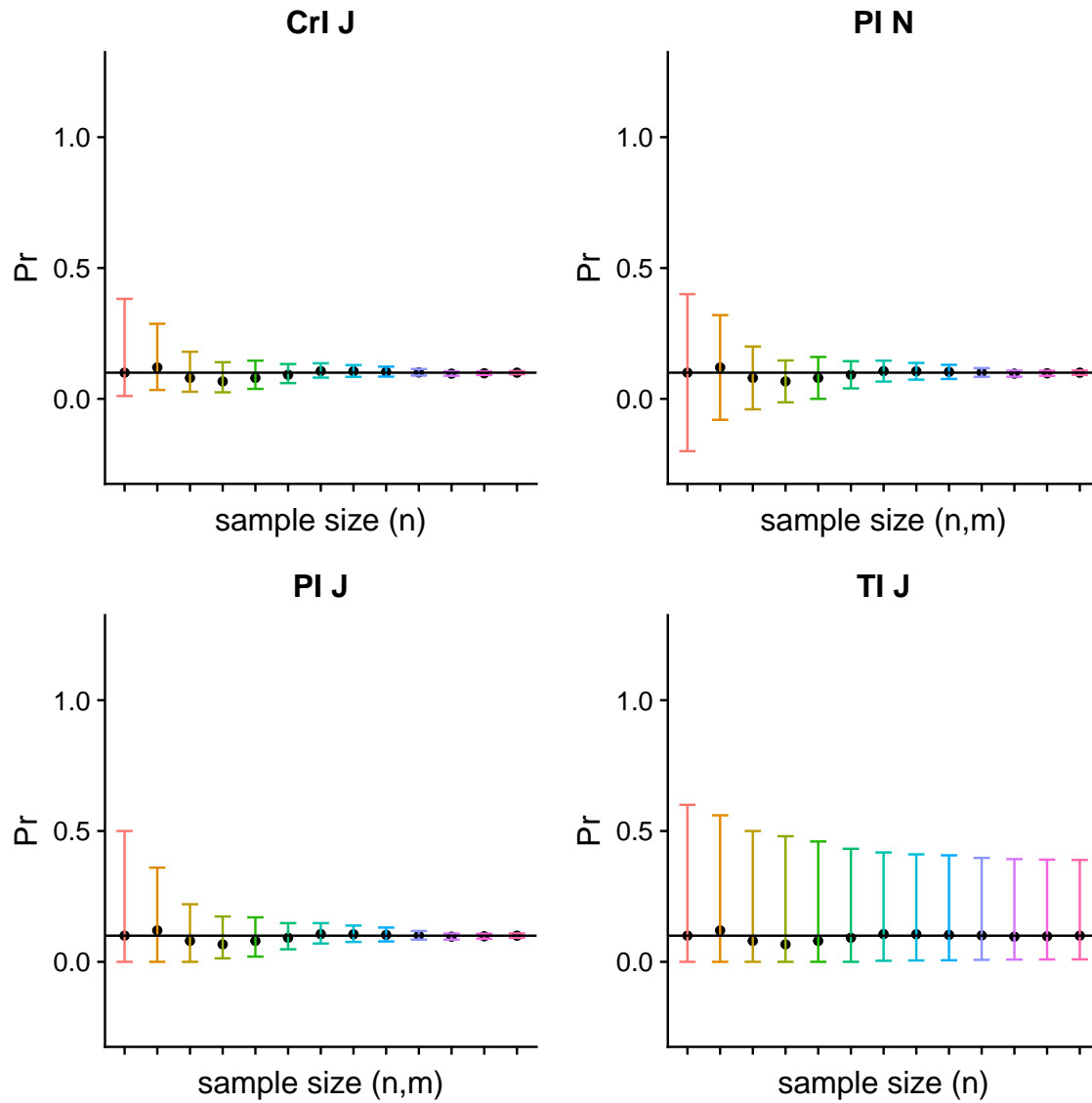


Figure 5.8: Different intervals for samples with a sample size n and size m of predicted observations ranging from 10 to 10000 drawn from a binomial distribution with true proportion of 0.1. CrI J: ($1 - \alpha = 0.95$) credible intervals based on Jeffreys prior. PI N: ($1 - \alpha = 0.95$) prediction intervals based on a normal approximation with $m = n$. PI J: ($1 - \alpha = 0.95$) prediction intervals based on Jeffreys approach with $m = n$. TI J: ($1 - \alpha = 0.95, P = 0.9$) tolerance intervals based on Jeffreys approach. In order to allow for comparisons, the prediction and tolerance intervals which predict the number of future successes $m * p$, the resulting intervals were divided by the corresponding sample size $m = n$, to obtain p . The black line indicates the true proportion.

n	x	Pr	low pi N	up pi N	low pi J	up pi J	low ti J	up ti J
10	3	0.30	-2	8	0	7	0	10
25	12	0.48	5	19	5	18	2	23
50	22	0.44	12	32	11	30	7	43
75	34	0.45	22	46	17	40	11	64
100	47	0.47	33	61	27	54	16	84
250	114	0.46	92	136	98	142	45	205
500	238	0.48	207	269	203	265	97	403
750	378	0.50	340	416	337	413	149	601
1000	496	0.50	452	540	464	552	202	798
2500	1256	0.50	1186	1326	1210	1348	525	1975
5000	2483	0.50	2385	2581	2419	2615	1069	3931
7500	3717	0.50	3596	3838	3668	3908	1617	5883
10000	4998	0.50	4859	5137	4854	5132	2167	7833

Table 5.5: Comparison of $(1 - \alpha = 0.95)$ prediction and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals with increasing sample size $n(= m)$ for binomial samples with a true proportion of 0.5. Presented are the number of successes ($m * p$).

n	x	Pr	low cri	up cri	low pi N	up pi N	low pi J	up pi J	low ti J	up ti J
10	1	0.10	0.01	0.38	-0.20	0.40	0.00	0.50	0.00	0.60
25	3	0.12	0.03	0.29	-0.08	0.32	0.00	0.36	0.00	0.56
50	4	0.08	0.03	0.18	-0.04	0.20	0.00	0.22	0.00	0.50
75	5	0.07	0.02	0.14	-0.01	0.15	0.01	0.17	0.00	0.48
100	8	0.08	0.04	0.15	0.00	0.16	0.02	0.17	0.00	0.46
250	23	0.09	0.06	0.13	0.04	0.14	0.05	0.15	0.00	0.43
500	53	0.11	0.08	0.14	0.07	0.15	0.07	0.15	0.00	0.42
750	79	0.11	0.08	0.13	0.07	0.14	0.08	0.14	0.01	0.41
1000	103	0.10	0.08	0.12	0.08	0.13	0.08	0.13	0.01	0.41
2500	252	0.10	0.09	0.11	0.08	0.12	0.08	0.12	0.01	0.40
5000	481	0.10	0.09	0.10	0.08	0.11	0.08	0.11	0.01	0.39
7500	732	0.10	0.09	0.10	0.09	0.11	0.09	0.11	0.01	0.39
10000	999	0.10	0.09	0.11	0.09	0.11	0.09	0.11	0.01	0.39

Table 5.6: Comparison of $(1 - \alpha = 0.95)$ credible, $(1 - \alpha = 0.95)$ prediction and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals with increasing sample size $n(= m)$ for binomial samples with a true proportion of 0.1. n and x are the number of trials and successes, respectively. Pr is the estimated proportion, 'low and up cri' correspond to the credible intervals based on Jeffreys approach, 'low and up pi N' are the predictive intervals based on a normal approximation, 'low and up pi J' are predictive intervals based on Jeffreys approach and 'low and up ti J' are the tolerance intervals obtained by the function `bintol.int()` specifying Jeffreys approach of the `tolerance` package.

n	x	Pr	low pi N	up pi N	low pi J	up pi J	low ti J	up ti J
10	1	0.10	-2	4	0	5	0	10
25	3	0.12	-2	8	0	9	2	23
50	4	0.08	-2	10	0	11	7	43
75	5	0.07	-1	11	1	13	11	64
100	8	0.08	0	16	2	17	16	84
250	23	0.09	10	36	12	37	45	205
500	53	0.11	33	73	35	74	97	403
750	79	0.11	55	103	57	104	149	601
1000	103	0.10	76	130	78	131	202	798
2500	252	0.10	210	294	212	295	525	1975
5000	481	0.10	423	539	425	540	1069	3931
7500	732	0.10	660	804	662	805	1617	5883
10000	999	0.10	915	1083	917	1084	2167	7833

Table 5.7: Comparison of $(1 - \alpha = 0.95)$ prediction and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals with increasing sample size $n(= m)$ for binomial samples with a true proportion of 0.5. Presented are the number of successes $(m * p)$.

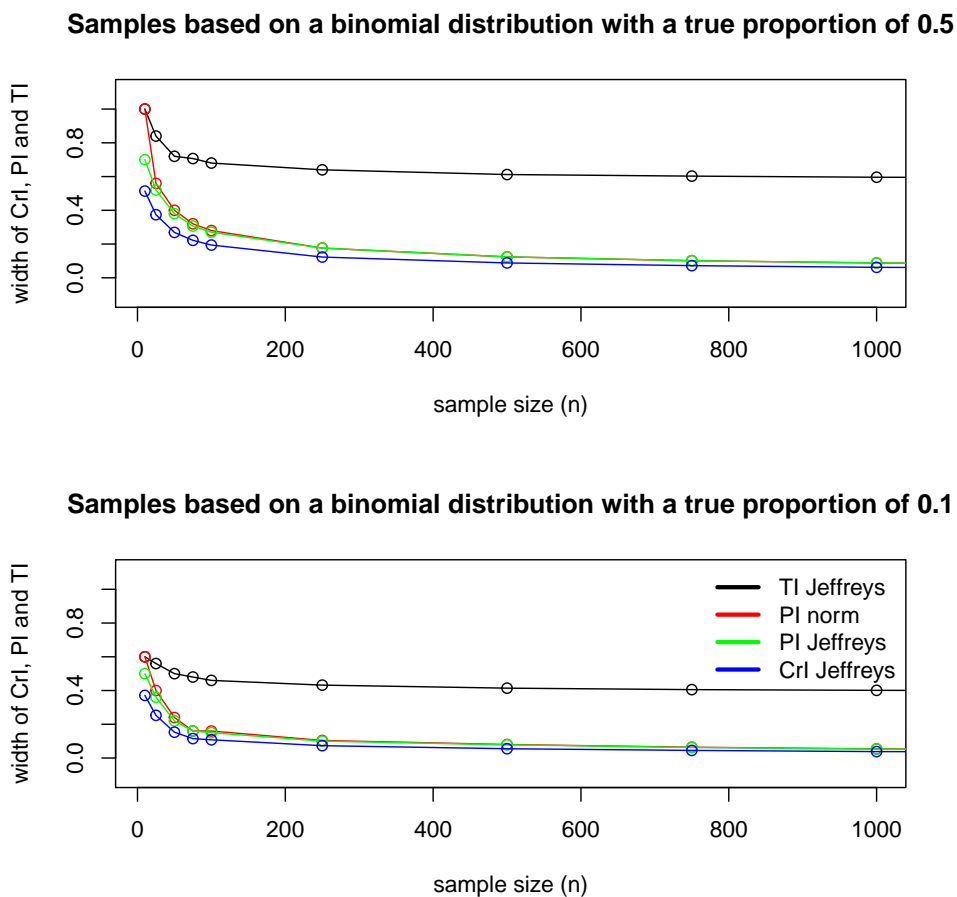


Figure 5.9: Illustration of the effect of sample size n on the width of $(1 - \alpha = 0.95)$ confidence intervals (blue), $(1 - \alpha = 0.95)$ prediction intervals for one future observation ($m = 1$) (red) and $(1 - \alpha = 0.95, P = 0.9)$ tolerance intervals (black). The random sample was drawn from a normal distribution with a true mean of 100 and a standard deviation of 10.

Chapter 6

Discussion

In this thesis confidence, prediction and tolerance intervals for the normal and the binomial distribution have been reviewed and illustrated with examples. All intervals have been considered from a classical and a Bayesian perspective. A few aspects became evident.

All three types of classical intervals can be considered in the generic form of

$$\widehat{parameter} \pm k \cdot SE(\widehat{parameter}).$$

For the confidence intervals, it holds that $k = 1.96$ and $se = \frac{\hat{\sigma}}{\sqrt{n}}$. For large sample sizes, se vanishes and the confidence interval is close to the parameter of interest. Thus, the width of a confidence interval is mainly driven by the sample size n . This is in line with its purpose to describe a single unknown parameter.

Similarly for the prediction intervals $k = 1.96$, but $se = \sigma \sqrt{\frac{1}{m} + \frac{1}{n}}$. With larger sample sizes n and m the prediction intervals will also decrease. Here, similar to the confidence intervals, the main purpose is to describe the location of a single parameter of interest. Since m and n are positive integers, even for large sample sizes of both n (and n), se will not vanish and σ will determine the width of the prediction interval. Thus, prediction intervals will always be wider than confidence intervals, because they do account for the variability in the observed and the future predicted data. Prediction of a single observation will also be associated with more uncertainty than predicting a large number of observations, which would be a prediction of the mean.

Regarding the tolerance intervals, according to (Young, 2010), k accounts for the sampling errors in the mean \bar{x} , $\hat{\sigma}$ the estimated standard deviation, the confidence level $(1 - \alpha)$ and the proportion of interest P . Even for large sample sizes, the width of the confidence interval, for a given P , will be determined by σ .

For tolerance intervals, an analytical solution for k is solely available for the simplest case, of a one-sided tolerance interval for a normal distribution. For the other intervals numerical approximations are needed and available in the **tolerance** package or tabulated for specific values of n , P and α .

The simulation with normal, skewed and bimodal data resulted in similar results for the confidence, prediction and tolerance intervals under the assumption of normality. This highlights again the need to carefully check the assumption of normality. If violated, intervals might be misleading. In such situations available non-parametric approaches should be used.

The usage of different approaches for the binomial distribution also confirms the stated severe applicability limitations of the widely used Wald confidence intervals (Brown *et al.*, 2001). For

the binomial distribution, the normal approximation or Wald methodology is not the best option. This pertains particularly to situations when the proportion p is close to 0 or 1. In these situations negative probabilities, might arise. Additionally, compared to the exact intervals based on the Clopper-Pearson approach for confidence intervals, a Bayesian alternative using Jeffreys prior was found to be superior. For prediction intervals, similarly, a Bayesian approach performed better than the Wald approximation.

Frequently, a classical confidence interval is misunderstood in the Bayesian sense. The usage of Bayesian intervals might also be advocated on the ground that the interpretation is more intuitive than the interpretation of the classical intervals based on repeated sampling.

Confidence intervals are taught in statistical classes, including introductory ones and are widely used or presumably even abused. Prediction intervals are less prominent. In most instances they will be taught and applied in the context of regression analysis, but not to predict future observations. In contrast, tolerance intervals are rarely presented, particularly in introductory classes. Tolerance intervals are widely used in engineering. Still there might be situations in medicine and veterinary medicine, when tolerance intervals are also relevant, i.e. to describe where a specific proportion of a population lies with a specified confidence. This information can only be conveyed by tolerance intervals, although in practice, confidence intervals might be presented as this is (not recommended) standard practice. With the advent of the **tolerance** package, the existing theoretical developments about the construction of tolerance intervals are now implemented in R.

In conclusion, as the three types of intervals serve for different purposes, the decision which interval to use should be context-driven and clearly justified. In order to avoid confusion and misunderstandings, all three types of intervals should be taught and presented.

Chapter 7

Appendix

7.1 Distribution of sample mean

Assume $X_1, \dots, X_n \stackrel{iid}{\sim} N(\mu, \sigma^2)$ and

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i.$$

Then

$$\begin{aligned} \mathbb{E} \bar{X} &= \frac{1}{n^2} \sum_{i=1}^n \mathbb{E} X_i = \frac{1}{n} n\mu = \mu \\ \text{Var} \bar{X} &= \frac{1}{n} \sum_{i=1}^n \text{Var} X_i = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n} \end{aligned}$$

7.2 Distribution of a sample proportion

Assume $X_1, \dots, X_n \stackrel{iid}{\sim} Be(p)$, a Bernoulli distribution with

$$\begin{aligned} X_i &= \begin{cases} 1 & \text{with } p \\ 0 & \text{with } 1 - p \end{cases} \\ \mathbb{E} X_i &= p \end{aligned}$$

$$\text{Var} X_i = p(1 - p)$$

Then,

$$\begin{aligned} \mathbb{E} \bar{X} &= \frac{1}{n} \sum_{i=1}^n \mathbb{E} X_i = \frac{1}{n} np = p \\ \text{Var} \bar{X} &= \frac{1}{n^2} \sum_{i=1}^n \text{Var} X_i = \frac{np(1 - p)}{n^2} = \frac{p(1 - p)}{n} \end{aligned}$$

7.3 Sum of variances

Proof that, if X and Y are independent*, then $Var(X - Y)$ is equal to $Var(X) + Var(Y)$

$$\begin{aligned} Var(X - Y) &= E((X - Y) - E(X - Y))^2 \\ &= E((X - E(X)) - (Y - E(Y)))^2 \\ &= E(X - E(X))^2 - E[2(X - E(X))(Y - E(Y))] + E(Y - E(Y))^2 \\ &\stackrel{*}{=} E(X - E(X))^2 + E(Y - E(Y))^2 \\ &= Var(X) + Var(Y). \end{aligned}$$

Chapter 8

Acknowledgments

I would like to express my gratitude to PD Malgorzata Roos for all the ideas, help, continuous support and guidance she has provided me during the work on this thesis. It was a pleasure to experience the enthusiasm she brought into this topic.

My sincere thanks also go to Prof. Paul R. Torgerson for given me the flexibility to do a masters in biostatistics.

To my friends and former fellow students of the Master Program in Biostatistics at the University of Zurich, - thank you for making the study of biostatistics so enjoyable. Thanks for the great discussions and for your help throughout the semesters. I would also like to thank the professors and lecturers from the Master Program in Biostatistics for their dedicated and passionate work.

I would also like to thank my parents, my husband and my daughter for their unconditional love and constant support and the patience with my ongoing studies.

Chapter 9

R code

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## Matrix products: default
##
## attached base packages:
## [1] stats      graphics  grDevices utils      datasets  methods   base
##
## other attached packages:
## [1] cowplot_0.9.4      moments_0.14      tolerance_1.3.0
## [4] Hmisc_4.2-0        Formula_1.2-3     survival_2.44-1.1
## [7] gridExtra_2.3      ggplot2_3.1.1     DescTools_0.99.28
## [10] Rmisc_1.5          plyr_1.8.4        lattice_0.20-38
## [13] Publish_2018.04.17 prodlim_2018.04.18 xtable_1.8-4
## [16] knitr_1.22
##
## loaded via a namespace (and not attached):
## [1] rgl_0.100.19      Rcpp_1.0.1
## [3] mvtnorm_1.0-10    assertthat_0.2.1
## [5] digest_0.6.18     mime_0.6
## [7] R6_2.4.0          backports_1.1.4
## [9] acepack_1.4.1     stats4_3.6.0
## [11] evaluate_0.13     highr_0.8
## [13] pillar_1.3.1      rlang_0.3.4
## [15] lazyeval_0.2.2    miniUI_0.1.1.1
## [17] rstudioapi_0.10   data.table_1.12.2
## [19] rpart_4.1-15      Matrix_1.2-17
## [21] checkmate_1.9.3   labeling_0.3
## [23] splines_3.6.0     webshot_0.5.1
## [25] stringr_1.4.0     foreign_0.8-71
## [27] htmlwidgets_1.3   munsell_0.5.0
## [29] shiny_1.3.2       compiler_3.6.0
## [31] httpuv_1.5.1      xfun_0.6
## [33] pkgconfig_2.0.2   base64enc_0.1-3
## [35] manipulate_1.0.1  htmltools_0.3.6
## [37] nnet_7.3-12       tidysselect_0.2.5
```

```
## [39] tibble_2.1.1          htmlTable_1.13.1
## [41] expm_0.999-4          crayon_1.3.4
## [43] dplyr_0.8.0.1         withr_2.1.2
## [45] later_0.8.0           MASS_7.3-51.4
## [47] grid_3.6.0           jsonlite_1.6
## [49] gtable_0.3.0          magrittr_1.5
## [51] scales_1.0.0          stringi_1.4.3
## [53] promises_1.0.1        latticeExtra_0.6-28
## [55] boot_1.3-22           lava_1.6.5
## [57] RColorBrewer_1.1-2    tools_3.6.0
## [59] manipulateWidget_0.10.0 glue_1.3.1
## [61] purrr_0.3.2           crosstalk_1.0.0
## [63] colorspace_1.4-1      cluster_2.0.9
```

```
## chapter 2 #####
#####
library(xtable)
library(Publish)
library(Rmisc)
library(DescTools)
library(ggplot2)
library(gridExtra)
library(Hmisc)
library(tolerance)
library(moments)
library(plyr)
library(ggplot2)
library(cowplot)
#####
prim <- read.csv("~/primary.csv", sep = ";")

# Figure 2.1 #
samsize <- 100
replicates <- 100
pval <- .05
set.seed(3105)

samples <- replicate(replicates, rnorm(100,100,10))
confint <- t(apply(samples, 2, function(x)
c(mean(x)-qt(1-pval/2,
  df=samsize-1)*sd(x)/sqrt(samsize),
  mean(x)+qt(1-pval/2, df=samsize-1)*sd(x)/sqrt(samsize))))

# Use red if mean outside interval
outside <- ifelse(confint[,1]>100 | confint[,2]<100, 2 , 1)
plot(c(100, 100), c(1, replicates), col="black", typ="l",
  main = expression(paste("100 (1- ",alpha, " = 0.95) confidence intervals")),
  ylab="Samples", xlim = c(90,110),
  xlab=expression(mu))
segments(confint[,1], 1:replicates, confint[,2], 1:replicates,
```

```

col=outside)

# Figure 2.2 #
boxplot(prim$primary, ylab = "Percentage weight loss")

# Figure 2.3 #
hist(prim$primary, breaks=50, ylab = "Frequency")

# Function #
my.CI.norm.func <- function(x, alpha){
  # x = data
  # 1 - alpha = confidence level
  SE <- sd(x) / sqrt(length(x))
  low.ci <- mean(x) - SE * qnorm((1-alpha/2),0,1)
  up.ci <- mean(x) + SE * qnorm((1-alpha/2),0,1)
  low.ci.f <- floor(low.ci * 100) /100
  up.ci.c <- ceiling(up.ci * 100) /100
  return(list(lower.ci= low.ci.f, upper.ci = up.ci.c))
}

# Table 2.1 #
Package <- c("Wald", "Publish", "Rmisc", "DescTools")
Function <- c("my.CI.norm.function", "ci.mean()", "CI()", "MeanCI()")

lower <- c(my.CI.norm.func(prim$primary,0.05)$lower.ci,
  round(as.numeric(ci.mean(prim$primary)[3]),2),
  round(as.numeric(CI(prim$primary)[3]),2),
  round(as.numeric(MeanCI(prim$primary)[2]),2))
upper <- c(my.CI.norm.func(prim$primary,0.05)$upper.ci,
  round(as.numeric(ci.mean(prim$primary)[4]),2),
  round(as.numeric(CI(prim$primary)[1]),2),
  round(as.numeric(MeanCI(prim$primary)[3]),2))

dat.ci <- as.data.frame(cbind(Package, Function, lower, upper))
names(dat.ci) <- c("Package","Function", "lower bound", "upper bound")
dat.cit <- xtable(dat.ci,
  caption= "95\\% CI ( $\mu$ ) for percentage body weight loss
  of 926 dogs in the \\citep{Flanagan2017} study.",
  label ="CI normal")
print(dat.cit, include.rownames=FALSE)

# Function #
my.CI.Wald.function <- function(x, n, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # 1 - alpha = confidence level
  p <- x/n
  SE <- sqrt((p*(1-p))/n)
  low.ci <- p - SE * qnorm((1-alpha/2),0,1)
  up.ci <- p + SE * qnorm((1-alpha/2),0,1)
  low.ci.f <- floor(low.ci*1000) /1000

```

```

up.ci.c <- ceiling(up.ci*1000) /1000
return(c(lower=low.ci.f, upper=up.ci.c))
}

# Function #
my.CP.function <- function(x, n, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # 1 - alpha = confidence level
  Lo <- function(p){pbinom(x - 1, n, p) - (1-alpha/2)}
  Up <- function(p){pbinom(x, n, p) - alpha/2}
  Low <- uniroot(Lo,c(.001,.999))
  low.ci.f <- floor(Low$root*1000) / 1000
  upper <- uniroot(Up,c(.001,.990))
  upper.ci.c <- ceiling(upper$root * 1000) / 1000
  return(c(lower=low.ci.f, upper=upper.ci.c))
}

# Function #
my.CI.Jeffrey.function <- function(x, n, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # 1 - alpha = credible level
  a <- x + 0.5
  b <- n - x + 0.5
  low.ci <- qbeta(alpha/2,a, b)
  up.ci <- qbeta((1-(alpha/2)),a, b)
  low.ci.f <- floor(low.ci*1000) /1000
  up.ci.c <- ceiling(up.ci*1000) /1000
  return(c(lower=low.ci.f, upper=up.ci.c))
}

# Table 2.2 #
Package <- c("Wald", "Exact", "stats", "Hmisc", "Jeffrey credible")
Function <- c("my.CI.Wald.function", "my.CP.function", "binom.test()",
             "binconf()", "my.CI.Jeffrey.function")

lower <- c(my.CI.Wald.function(12,16,0.05)[1],
           my.CP.function(12,16,0.05)[1],
           round(as.numeric(binom.test(12,16)$conf.int[1]),3),
           round(as.numeric(binconf(12,16,0.05)[2]),3),
           my.CI.Jeffrey.function(12,16,0.05)[1])
upper <- c(my.CI.Wald.function(12,16,0.05)[2],
           my.CP.function(12,16,0.05)[2],
           round(as.numeric(binom.test(12,16)$conf.int[2]),3),
           round(as.numeric(binconf(12,16,0.05)[3]),3),
           my.CI.Jeffrey.function(12,16,0.05)[2])

dat.steel <- as.data.frame(cbind(Package, Function, lower, upper))
names(dat.steel) <- c("Method/Package", "Function", "lower bound", "upper bound")
dat.s <- xtable(dat.steel,

```



```

        caption= "95\\% CI (p) for steel",
        label ="CI binomial 1")

print(dat.s, include.rownames=FALSE)

# Table 2.3 #
Package <- c("Wald", "Exact", "stats", "Hmisc", "Jeffrey credible")
Function <- c("my.CI.Wald.function", "my.CP.function", "binom.test()",
             "binconf()", "my.CI.Jeffrey.function")

lower <- c(as.numeric(my.CI.Wald.function(2,16,0.05)[1]),
           as.numeric(my.CP.function(2,16,0.05)[1]),
           (round(as.numeric(binom.test(2,16)$conf.int[1]),3)),
           round(as.numeric(binconf(2,16,0.05)[2]),3),
           as.numeric(my.CI.Jeffrey.function(2,16,0.05)[1]))
upper <- c(as.numeric(my.CI.Wald.function(2,16,0.05)[2]),
           as.numeric(my.CP.function(2,16,0.05)[2]),
           round(as.numeric(binom.test(2,16)$conf.int[2]),3),
           round(as.numeric(binconf(2,16,0.05)[3]),3),
           as.numeric(my.CI.Jeffrey.function(2,16,0.05)[2]))

dat.horn <- as.data.frame(cbind(Package, Function, lower, upper))
names(dat.horn) <- c("Method/Package", "Function", "lower bound", "upper bound")
dat.h <- xtable(dat.horn,
               caption= "95\\% CI (p) for horn",
               label ="CI binomial 2")

print(dat.h, include.rownames=FALSE)

# Figure 2.4 #
par(mfrow=c(1,2))
k = 12
n = 16

## x-axis for plotting
numSteps = 200
x = seq(0, 1, 1 / numSteps)

## Likelihood function
L = x^k * (1 - x)^(n - k)

## Just normalize likelihood to integrate to one (for purposes of plotting)
L = L / sum(L) * numSteps

### Uniform Prior
## Plot likelihood

plot(x, L, type = 'l', lwd = 3, lty = 3, ylim = c(0,6),
     main = "Steel impactor: Binomial (12,16)",
     xlab = "P", ylab = "pdf")

```

```

## Plot Beta(1,1) prior
lines(x, dbeta(x, 0.5, 0.5), lwd = 3, col = "blue")

## Plot posterior
lines(x, dbeta(x, k + 0.5, n - k + 0.5), lwd = 3, col = "red")
abline(v= 0.509, lwd = 2, col = "green")
abline(v= 0.91, lwd = 2, col = "green")

k = 2
n = 16

## x-axis for plotting
numSteps = 200
x = seq(0, 1, 1 / numSteps)

## Likelihood function
L = x^k * (1 - x)^(n - k)

## Just normalize likelihood to integrate to one (for purposes of plotting)
L = L / sum(L) * numSteps

### Uniform Prior
## Plot likelihood

plot(x, L, type = 'l', lwd = 3, lty = 3, ylim = c(0,6),
     main = "Horn impactor: Binomial (2,16)",
     xlab = "P", ylab = "pdf")

## Plot Beta(1,1) prior
lines(x, dbeta(x, 0.5, 0.5), lwd = 3, col = "blue")

## Plot posterior
lines(x, dbeta(x, k + 0.5, n - k + 0.5), lwd = 2, col = "red")
abline(v= 0.026, lwd = 2, col = "green")
abline(v= 0.345, lwd = 2, col = "green")

legend("topright", c("Likelihood", "Prior", "Posterior", "CrI"),
      bty = "n", lty = c(3, 1, 1, 1), lwd = 3,
      col = c("black", "blue", "red", "green"))

## chapter 3 #####
#####
# Function #
my.PI.norm.function <- function(x, s, n, m, alpha){
# x = mean of observed sample
# s = standard deviation of observed sample
# n = sample size of observed sample
# m = sample size of future observations, m=1 for a single observation
# 1 - alpha = prediction level
SE <- s*sqrt((1/n) + (1/m))
low.ci <- x - SE * qnorm((1-alpha/2),0,1)

```

```

up.ci <- x + SE * qnorm((1-alpha/2),0,1)
low.ci.f <- floor(low.ci*100) /100
up.ci.c <- ceiling(up.ci*100) /100
return(c(lower=low.ci.f, upper=up.ci.c))
}

# Figure 3.1 #
set.seed(531051972)
M <- 100
N <- 10
mydat <- matrix(rnorm(M*N,100,10),ncol = M)

x <- apply(mydat, 2, mean)
s <- apply(mydat, 2, sd)
m <- rep(1,100)
n <- rep(10,100)
alpha <- rep(0.05,100)

for (i in 1:N){
  lower <- my.PI.norm.function(x,s,n,m,alpha)[1:100]
  upper <- my.PI.norm.function(x,s,n,m,alpha)[101:200]
}

PIs <- as.data.frame(cbind(lower,upper))

set.seed(3105)
ynew <- rnorm(100,100,10)
# Use red if outside interval
outside <- ifelse(!(ynew > PIs[,1] & ynew < PIs[,2]), 2 , 1)
plot(c(100, 100), c(1, 100), col="black", typ="l",
     main = expression(paste("100 (1- ",alpha, " = 0.95) prediction intervals")),
     ylab="Samples", xlim = c(40,160),
     xlab=expression(mu))
points(ynew,1:100, pch = 17, col = outside, cex=0.8)
segments(PIs[,1], 1:100, PIs[,2], 1:100,
         col=outside)

# Table 3.1 #
x <- mean(prim$primary)
s <- sd(prim$primary)
n <- length(prim$primary)
alpha <- 0.05
m <- 1

pr.m <- lm(primary ~ 1, data = prim)
prci <- predict(pr.m,interval = "prediction")
prci[1,]

Function <- c("my.PI.norm.function", "predict.lm")

lower <- c(round(as.numeric(my.PI.norm.function(x, s, n, m, alpha)[1]),2),

```

```

        round(as.numeric(prci[1,2]),2))
upper <- c(round(as.numeric(my.PI.norm.function(x, s, n, m, alpha)[2]),2),
        round(as.numeric(prci[1,3]),2))

dat.pin <- as.data.frame(cbind(Function, lower, upper))
names(dat.pin) <- c("Function name", "lower bound", "upper bound")
dat.pint <- xtable(dat.pin,
                  caption="$(1-\\alpha=0.95)$ PI($y_{new}$) for
                  the percentage of weight loss in one overweight dog.",
                  label = "PI norm")

print(dat.pint, include.rownames=FALSE)

# Function #
my.PI.bin.function <- function(x, n, m, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # m = sample size of future observations
  # 1 - alpha = prediction level
  p <- x/n
  quant <- qnorm((1-alpha/2),0,1)
  SE <- sqrt((m+n)*p*(1-p))
  low.ci <- m*p - quant*SE
  up.ci <- m*p + quant*SE
  low.ci.f <- floor(low.ci)
  up.ci.c <- ceiling(up.ci)
  return(c(lower=low.ci.f, upper=up.ci.c))
}

# Function #
bin.pri.pred <- function(a,b,nn,yn){
  # a = shape parameter a of a beta distribution
  # b = shape parameter b of a beta distribution
  # nn = sample size of new, not yet observed data
  # yn = successes in new, not yet observed data
  return(exp(log(choose(nn, yn))+log(beta(a+yn,b+nn-yn))-log(beta(a,b))))
}

# Function #
bin.post.pred <- function(a,b,nn,yn,no,yo){
  # a = shape parameter a of a beta distribution
  # b = shape parameter b of a beta distribution
  # nn = sample size of new, not yet observed data
  # yn = successes in new, not yet observed data
  # no = sample size of old, already observed data
  # yo = successes in old, already observed data
  return(exp(log(choose(nn, yn))
            +log(beta(a+yo+yn,b+no-yo+nn-yn))
            -log(beta(a+yo,b+no-yo))))
}

```

```

# Function #
dbetabinom <- function(x, size, a, b) {
  if (any(size <= 0))
    stop("size must be greater than 0")
  if (any(a <= 0))
    stop("a must be greater than 0")
  if (any(b <= 0))
    stop("b must be greater than 0")
  if (any(x < 0) || any(x > size))
    stop("x must be between 0 and size")
  exp(lbeta(x + a, size - x + b) - lbeta(a, b) +
      lchoose(size, x))
}

# Function #
qbetabinom <- function(p, size, a, b) {
  if (any(size <= 0))
    stop("size must be greater than 0")
  if (any(a <= 0))
    stop("a must be greater than 0")
  if (any(b <= 0))
    stop("b must be greater than 0")
  if (any(p <= 0) || any(p >= 1))
    stop("p must be between 0 and 1")
  the.cumsum <- cumsum(dbetabinom(0:size, size, a, b))
  sapply(p, function(x) sum(the.cumsum < x))
}

# Function #
my.Jeffrey.PI.function <- function(x, n, m, alpha){
  # x = number of successes of an observed sample
  # n = number of trials in observed sample
  # m = sample size of future observations
  # 1 - alpha = prediction level
  a <- x + 0.5
  b <- n - x + 0.5
  size <- m
  low.ci <- qbetabinom(alpha/2,size,a, b)
  up.ci <- qbetabinom((1-(alpha/2)),size,a, b)
  return(c(lower=low.ci, upper=up.ci))
}

# Table 3.2 #
Impactor <- c("steel", "horn")

lower <- c(as.numeric(my.PI.bin.function(12,16,16,0.05)[1]),
          as.numeric(my.PI.bin.function(2,16,16,0.05)[1]))

upper <- c(as.numeric(my.PI.bin.function(12,16,16,0.05)[2]),
          as.numeric(my.PI.bin.function(2,16,16,0.05)[2]))

```



```

abline(v= 7, lwd = 2, col = "green")
abline(v= 16, lwd = 2, col = "green")

data.t.i <- list(n = 16, r = 2)

plot(yn.a, bin.pri.pred(prior.1$a, prior.1$b, n.a, yn.a), type="n",
     xlab = "number of events", ylab = "predictive distribution",
     ylim=c(0,0.25))
lines(yn.a, bin.pri.pred(prior.1$a, prior.1$b, n.a, yn.a),col="blue",
      lwd = 2)
lines(yn.a, bin.post.pred(prior.1$a, prior.1$b, n.a, yn.a, data.t.i$n,
                          data.t.i$r),col="red", lwd = 2)
abline(v= 0, lwd = 2, col = "green")
abline(v= 7, lwd = 2, col = "green")

legend("topright",c("pri pred", "post pred", "PI"),
      col=c("blue", "red", "green"),
      bty="n",lty=c(1),lwd=c(2))

## chapter 4 #####
#####
str(prim)
mean(prim$primary)
sd(prim$primary)
my.CI.norm.func(prim$primary, 0.05)
my.PI.norm.function(11.36778, 5.848243, 926, 100, 0.05)
my.PI.norm.function(11.36778, 5.848243, 926, 1, 0.05)
normtol.int(prim$primary, alpha = 0.05, P = 0.90, side = 2)

set.seed(100)
sleep.hours <- rlogis(n = 20, location = 5, scale = 1)
nptol.int(x = sleep.hours, alpha = 0.05, P = 0.9, side = 1)

# Table 4.1
set.seed(100)
xt <- rnorm(100,0,0.2)

para <- normtol.int(xt, alpha = 0.05, P = 0.9, side = 2)
notp <- nptol.int(xt, alpha = 0.05, P = 0.9, side = 2, method = "WALD")
bayesn <- bayesnormtol.int(xt, alpha = 0.05, P = 0.9,
                          side = 2, method = "EXACT",
                          hyper.par = list(mu.0 = 0,
                                           sig2.0 = 0.2, n.0 = 100, m.0 = 10))

approach <- c("parametric", "non-parametric", "Bayes")

lowert <- c(round(as.numeric(para[4]),3),
            round(as.numeric(notp[1,3]),3),
            round(as.numeric(bayesn[3]),3))

```

```

uppert <- c(round(as.numeric(para[5]),3),
            round(as.numeric(notp[1,4]),3),
            round(as.numeric(bayesn[4]),3))

dat.tol <- as.data.frame(cbind(approach,lowert, uppert))
names(dat.tol) <- c("Approach", "lower bound", "upper bound")
dat.toll <- xtable(dat.tol,
                  caption= "$ (1-\\alpha = 0.95, P = 0.9)$
TI based on a classical normal, a non parametric
and a Bayesian approach.",
                  label ="TI com")

print(dat.toll, include.rownames=FALSE)

# Figure 4.1 #
set.seed(4120191)
M <- 100
N <- 5
mydat <- matrix(rnorm(M*N,100,10),ncol = M)

TIs <- data.frame(matrix(nrow = 100, ncol = 2))
colnames(TIs) <- c("low.ti", "high.ti")

for (i in 1:100) {
  low.ti <- as.numeric(normtol.int(mydat[,i], 0.05, 0.9, side = 2)[4])
  high.ti <- as.numeric(normtol.int(mydat[,i], 0.05, 0.9, side = 2)[5])
  TIs[i,1] <- low.ti
  TIs[i,2] <- high.ti
}

content <- pnorm(TIs[,2],100,10) - pnorm(TIs[,1],100,10)
outside <- ifelse(content<0.9, 2 , 1)
plot(c(100, 100), c(1, 100), col="black", typ="l",
     main = expression(paste("100 (1- ",alpha, " = 0.95) tolerance intervals")),
     ylab="Samples", xlim = c(20,180),
     xlab=expression(mu))
segments(TIs[,1], 1:100, TIs[,2], 1:100,
         col=outside)

# Figure 4.2
prim.n <- normtol.int(prim$primary, 0.05, 0.9, 2, method = "HE")
prim.np <- nptol.int(prim$primary, 0.05, 0.9, 2)
prim.b <- bayesnormtol.int(prim$primary, alpha = 0.05, P = 0.9,
                          side = 2, method = "EXACT",
                          hyper.par = list(mu.0 = 11.4,
                                           sig2.0 = 5.8, n.0 = 926, m.0 = 10))

approach <- c("parametric", "non-parametric", "Bayesian")

lower <- c(as.numeric(prim.n[4]),
          (prim.np[3]),

```



```

    as.numeric(prim.b[3]))

upper <- c(as.numeric(prim.n[5]),
          (prim.np[4]),
          as.numeric(prim.b[4]))

dat.tolbmi <- as.data.frame(cbind(approach, lower, upper))
names(dat.tolbmi) <- c("Approach", "lower bound", "upper bound")
dat.tolbmis <- xtable(dat.tolbmi,
                      caption= "$ (1-\\alpha = 0.95, P=0.9)$
                      classical normal, non-parametric and
                      Bayesian tolerance intervals of the
                      percentage body weight loss in overweight dogs.",
                      label ="TI com")

print(dat.tolbmis, include.rownames=FALSE)

#one-sided
bintol.int(x = 10, n = 1000, m = 50, alpha = 0.05, P = 0.9, side = 1)

#two-sided
bintol.int(x = 10, n = 1000, m = 50, alpha = 0.05, P = 0.9, side = 2)

# Table 4.3 and Table 4.4 #
bint1s1 <- bintol.int(x = 12, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "LS")
bint2s1 <- bintol.int(x = 12, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "WS")
bint3s1 <- bintol.int(x = 12, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "CP")
bint4s1 <- bintol.int(x = 12, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "JF")

bint1h1 <- bintol.int(x = 2, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "LS")
bint2h1 <- bintol.int(x = 2, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "WS")
bint3h1 <- bintol.int(x = 2, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "CP")
bint4h1 <- bintol.int(x = 2, n = 16, m = 16, alpha = 0.05, P = 0.9, side = 1,
                     method = "JF")

Approach <- c(rep("classical",3), rep("Bayesian",1))

Function <- c("Large sample", "Wilson", "Clopper Pearson", "Bayesian")

lower <- c(round(as.numeric(bint1s1[4]),2),
          round(as.numeric(bint2s1[4]),2),
          round(as.numeric(bint3s1[4]),2),
          round(as.numeric(bint4s1[4]),2))

```

```

dat.bint <- as.data.frame(cbind(Approach,Function,lower))
names(dat.bint) <- c("Approach","method TI", "lower bound")
dat.bints <- xtable(dat.bint,
                    caption= "One-sided  $(1-\alpha = 0.95, P=0.9)$ 
                    tolerance intervals for minimum number of fractures
                    due to the steel in 16 kick experiments.",
                    label ="TI steel")

print(dat.bints, include.rownames=FALSE)

Approach <- c (rep("classical",3), rep("Bayesian",1))

Function <- c("Large sample", "Wilson", "Clopper Pearson", "Bayesian")

lower <- c(round(as.numeric(bint1h1[4]),2),
           round(as.numeric(bint2h1[4]),2),
           round(as.numeric(bint3h1[4]),2),
           round(as.numeric(bint4h1[4]),2))

dat.binh <- as.data.frame(cbind(Approach,Function,lower))
names(dat.binh) <- c("Approach","method TI", "lower bound")
dat.binhs <- xtable(dat.binh,
                    caption= "One-sided  $(1-\alpha = 0.95, P=0.9)$ 
                    tolerance intervals for the minimum number of
                    fractures due to the horn impactor in 16
                    kick experiments.",
                    label ="TI horn")

print(dat.binhs, include.rownames=FALSE)

## chapter 5 #####
#####
# Table 5.1, Figure 5.1, Table 5.2
# to generate a random normal sample
set.seed(12345)
a <- rnorm(300, 5.901, 4.2)

# to generate a bimodal random sample
set.seed(12345)
N <- 300
components <- sample(1:3,prob=c(0.3,0.5,0.2),size=N,replace=TRUE)
mus <- c(1,10,3)
sds <- sqrt(c(1,1,0.1))
b <- rnorm(n=N,mean=mus[components],sd=sds[components])

# to generate a skewed random sample
set.seed(12345)
c <- rexp(N,rate= 1)

# to obtain a standardized vector
my.x2z <- function(x){

```

```

return((x-mean(x))/sd(x))
}

# from standardized to normal with mu and sigma vector
my.z2y <- function(z,mu,sigma){
  return(mu+sigma*z)
}

az <- my.x2z(a)
a2y <- my.z2y(az,mu = 16,sigma = 12)
mean(a2y);sd(a2y)

bz <- my.x2z(b)
b2y <- my.z2y(bz,mu = 16,sigma = 12)
mean(b2y);sd(b2y)

cz <- my.x2z(c)
c2y <- my.z2y(cz,mu = 16,sigma = 12)
mean(c2y);sd(c2y)

# Table 5.1 #
A <- c(N, mean(a2y), sd(a2y),round(min(a2y),2), round(max(a2y),2),
       round(skewness(a2y),2), round(kurtosis(a2y),2), round(IQR(a2y),2))

B <- c(N, mean(b2y), sd(b2y),round(min(b2y),2), round(max(b2y),2),
       round(skewness(b2y),2), round(kurtosis(b2y),2), round(IQR(b2y),2))

C <- c(N, mean(c2y), sd(c2y),round(min(c2y),2), round(max(c2y),2),
       round(skewness(c2y),2), round(kurtosis(c2y),2), round(IQR(c2y),2))

Func <- c("n", "mean", "sd", "min", "max", "skewness", "kurtosis", "IQR")
descr <- as.data.frame(cbind(Func,A, B, C))

names(descr) <- c("", "A","B", "C")

descrt <- xtable(descr, caption= " Descriptive statistics of the
                        three scenarios A, B and C",
                        label ="descr scenario")

print(descrt, include.rownames= F, table.placement = "b",
      caption.placement = "top")

# Figure 5.1 #
par(mfrow=c(3,1))
hist(a2y, breaks = 50, xlim = c(0,50), main="Scenario A", xlab = "")
hist(b2y, breaks = 50, xlim = c(0,50), main="Scenario B", xlab = "")
hist(c2y, breaks = 50, xlim = c(0,50), main="Scenario C", xlab = "")

lower <- c(my.CI.norm.func(a2y,0.05)[1],
           my.CI.norm.func(b2y,0.05)[1],
           my.CI.norm.func(c2y,0.05)[1],

```

```

my.PI.norm.function(mean(a2y),sd(a2y),300,300,0.05)[1],
my.PI.norm.function(mean(b2y),sd(b2y),300,300,0.05)[1],
my.PI.norm.function(mean(c2y),sd(c2y),300,300,0.05)[1],
as.numeric(normtol.int(a2y, 0.05, 0.9, side = 2)[4]),
as.numeric(normtol.int(b2y, 0.05, 0.9, side = 2)[4]),
as.numeric(normtol.int(c2y, 0.05, 0.9, side = 2)[4]),
as.numeric(nptol.int(a2y, 0.05, 0.9, 2)[3]),
as.numeric(nptol.int(b2y, 0.05, 0.9, 2)[3]),
as.numeric(nptol.int(c2y, 0.05, 0.9, 2)[3]))

upper <- c(my.CI.norm.func(a2y,0.05)[2],
  my.CI.norm.func(b2y,0.05)[2],
  my.CI.norm.func(c2y,0.05)[2],
  my.PI.norm.function(mean(a2y),sd(a2y),300,300,0.05)[2],
  my.PI.norm.function(mean(b2y),sd(b2y),300,300,0.05)[2],
  my.PI.norm.function(mean(c2y),sd(c2y),300,300,0.05)[2],
  as.numeric(normtol.int(a2y, 0.05, 0.9, side = 2)[5]),
  as.numeric(normtol.int(b2y, 0.05, 0.9, side = 2)[5]),
  as.numeric(normtol.int(c2y, 0.05, 0.9, side = 2)[5]),
  as.numeric(nptol.int(a2y, 0.05, 0.9, 2)[4]),
  as.numeric(nptol.int(b2y, 0.05, 0.9, 2)[4]),
  as.numeric(nptol.int(c2y, 0.05, 0.9, 2)[4]))

# Tabl3 5.2 #
Scenario <- rep(c("A","B","C"),4)
Interval <- c("CI", "CI", "CI", "PI", "PI", "PI",
  "TI","TI","TI","TI","TI","TI")
Function <- c("my.CI.norm", "my.CI.norm", "my.CI.norm",
  "my.PI.norm", "my.PI.norm", "my.PI.norm",
  "normtol.int", "normtol.int", "normtol.int",
  "nptol.int", "nptol.int", "nptol.int")

P <- c(rep("",6), rep(0.9,6))

diff.int <- as.data.frame(cbind(Scenario, Function, Interval, P,
  lower, upper))
names(diff.int) <- c("Scenario", "Function","Interval", "P",
  "lower bound", "upper bound")

diff.ints <- xtable(diff.int,
  caption= "Comparison of confidence, prediction and
  tolerance intervals for the three scenarios A, B and
  C with a  $(1-\alpha = 0.95)$  confidence and prediction
  level, and a  $(1-\alpha = 0.95, P = 0.9)$  tolerance
  interval. The parametric \textit{normtol.int()}
  and non-parametric \textit{nptol.int()} functions
  for the respective tolerance intervals originate
  from the \texttt{tolerance} package.",
  label = "CI.ABC")

print(diff.ints, include.rownames=F)

```

```

# Figure 5.2 to 5.6, Table 5.3 #
set.seed(707)
a.6 <- rnorm(10000,100,10)
a.10000 <- a.6[1:10000]
a.7500 <- a.6[1:7500]
a.5000 <- a.6[1:5000]
a.2500 <- a.6[1:2500]
a.1000 <- a.6[1:1000]
a.750 <- a.6[1:750]
a.500 <- a.6[1:500]
a.250 <- a.6[1:250]
a.100 <- a.6[1:100]
a.75 <- a.6[1:75]
a.50 <- a.6[1:50]
a.25 <- a.6[1:25]
a.10 <- a.6[1:10]

samples <- factor(c(rep(10, 10),
                    rep(25, 25),
                    rep(50, 50),
                    rep(75, 75),
                    rep(100, 100),
                    rep(250, 250),
                    rep(500, 500),
                    rep(750, 750),
                    rep(1000, 1000),
                    rep(2500, 2500),
                    rep(5000, 5000),
                    rep(7500, 7500),
                    rep(10000, 10000)))

numbers <- c(a.10,a.25,a.50,a.75,a.100,a.250,a.500,a.750,
            a.1000,a.2500,a.5000,a.7500,a.10000)

samples <- data.frame(samples)

dat <- data.frame(numbers)
mydats <- data.frame(c(samples,dat))

cdata <- ddply(mydats, "samples", summarise,
              N      = as.integer(length(numbers)),
              mean   = mean(numbers),
              sd     = sd(numbers),
              se     = sd(numbers) / sqrt(N),
              qts    = qnorm(.975, 0,1),
              low.ci = mean - (sd(numbers) / sqrt(N))*(qnorm(.975, 0,1)),
              up.ci  = mean + (sd(numbers) / sqrt(N))*(qnorm(.975, 0,1)))

cdata$low.pim <- my.PI.norm.function(mean(a.100),sd(a.100),100,

```

```

                                c(10,25,50,75,100,250,500,750,1000,
                                2500,5000,7500,10000),0.05)[1:13]

cdata$up.pim    <- my.PI.norm.function(mean(a.100),sd(a.100),100,
                                c(10,25,50,75,100,250,500,750,1000,
                                2500,5000,7500,10000),0.05)[14:26]

cdata$low.pi    <- my.PI.norm.function(cdata$mean,cdata$sd,cdata$N,1,0.05)[1:13]
cdata$up.pi     <- my.PI.norm.function(cdata$mean,cdata$sd,cdata$N,1,0.05)[14:26]

lt1  <- as.numeric(normtol.int(a.10, alpha = 0.05, P = 0.9, side = 2)[4])
lt2  <- as.numeric(normtol.int(a.25, alpha = 0.05, P = 0.9, side = 2)[4])
lt3  <- as.numeric(normtol.int(a.50, alpha = 0.05, P = 0.9, side = 2)[4])
lt4  <- as.numeric(normtol.int(a.75, alpha = 0.05, P = 0.9, side = 2)[4])
lt5  <- as.numeric(normtol.int(a.100, alpha = 0.05, P = 0.9, side = 2)[4])
lt6  <- as.numeric(normtol.int(a.250, alpha = 0.05, P = 0.9, side = 2)[4])
lt7  <- as.numeric(normtol.int(a.500, alpha = 0.05, P = 0.9, side = 2)[4])
lt8  <- as.numeric(normtol.int(a.750, alpha = 0.05, P = 0.9, side = 2)[4])
lt9  <- as.numeric(normtol.int(a.1000, alpha = 0.05, P = 0.9, side = 2)[4])
lt10 <- as.numeric(normtol.int(a.2500, alpha = 0.05, P = 0.9, side = 2)[4])
lt11 <- as.numeric(normtol.int(a.5000, alpha = 0.05, P = 0.9, side = 2)[4])
lt12 <- as.numeric(normtol.int(a.7500, alpha = 0.05, P = 0.9, side = 2)[4])
lt13 <- as.numeric(normtol.int(a.10000, alpha = 0.05, P = 0.9, side = 2)[4])

ut1  <- as.numeric(normtol.int(a.10, alpha = 0.05, P = 0.9, side = 2)[5])
ut2  <- as.numeric(normtol.int(a.25, alpha = 0.05, P = 0.9, side = 2)[5])
ut3  <- as.numeric(normtol.int(a.50, alpha = 0.05, P = 0.9, side = 2)[5])
ut4  <- as.numeric(normtol.int(a.75, alpha = 0.05, P = 0.9, side = 2)[5])
ut5  <- as.numeric(normtol.int(a.100, alpha = 0.05, P = 0.9, side = 2)[5])
ut6  <- as.numeric(normtol.int(a.250, alpha = 0.05, P = 0.9, side = 2)[5])
ut7  <- as.numeric(normtol.int(a.500, alpha = 0.05, P = 0.9, side = 2)[5])
ut8  <- as.numeric(normtol.int(a.750, alpha = 0.05, P = 0.9, side = 2)[5])
ut9  <- as.numeric(normtol.int(a.1000, alpha = 0.05, P = 0.9, side = 2)[5])
ut10 <- as.numeric(normtol.int(a.2500, alpha = 0.05, P = 0.9, side = 2)[5])
ut11 <- as.numeric(normtol.int(a.5000, alpha = 0.05, P = 0.9, side = 2)[5])
ut12 <- as.numeric(normtol.int(a.7500, alpha = 0.05, P = 0.9, side = 2)[5])
ut13 <- as.numeric(normtol.int(a.10000, alpha = 0.05, P = 0.9, side = 2)[5])

cdata$low.ti <- c(lt1,lt2,lt3,lt4,lt5,lt6,lt7,lt8,lt9,lt10,lt11,lt12,lt13)
cdata$up.ti  <- c(ut1,ut2,ut3,ut4,ut5,ut6,ut7,ut8,ut9,ut10,ut11,ut12,ut13)

cdata <- as.data.frame(cdata)
cdata$width.ci <- cdata$up.ci - cdata$low.ci
cdata$width.pi <- cdata$up.pi - cdata$low.pi
cdata$width.ti <- cdata$up.ti - cdata$low.ti

pd <- position_dodge(0.78)
plot1 <- ggplot(cdata, aes(x=samples, y = mean, group = samples)) +
  #draws the means
  geom_point(position=pd) +
  #draws the CI error bars

```

```

    geom_errorbar(data=cdata,
                  aes(ymin= low.ci, ymax=up.ci,
                     color=samples), width=.5, position=pd)
h <- 100
plot2 <- plot1 + geom_hline(aes(yintercept=h))

plot2 + labs(x = "sample size (n)") + annotate(geom="text",
                                              label="truth",
                                              x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

pd <- position_dodge(0.78)
plot3 <- ggplot(cdata, aes(x=samples, y = mean, group = samples)) +
  #draws the means
  geom_point(position=pd) +
  #draws the CI error bars
  geom_errorbar(data=cdata,
                aes(ymin= low.pi, ymax=up.pi,
                   color=samples), width=.5, position=pd)
h <- 100
plot4 <- plot3 + geom_hline(aes(yintercept=h))

plot4 + labs(x = "sample size (n)") + annotate(geom="text",
                                              label="truth",
                                              x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

pd <- position_dodge(0.78)
plot5 <- ggplot(cdata, aes(x=samples, y = mean, group = samples)) +
  #draws the means
  geom_point(position=pd) +
  #draws the CI error bars
  geom_errorbar(data=cdata,
                aes(ymin= low.pim, ymax=up.pim,
                   color=samples), width=.5, position=pd)
h <- 100
plot6 <- plot5 + geom_hline(aes(yintercept=h))

plot6 + labs(x = "sample size (m)") + annotate(geom="text",
                                              label="truth",
                                              x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

pd <- position_dodge(0.78)
plot7 <- ggplot(cdata, aes(x=samples, y = mean, group = samples)) +
  #draws the means
  geom_point(position=pd) +
  #draws the CI error bars
  geom_errorbar(data=cdata,
                aes(ymin= low.ti, ymax=up.ti,
                   color=samples), width=.5, position=pd)

```

```

h <- 100
plot8 <- plot7 + geom_hline(aes(yintercept=h))
plot8 + labs(x = "sample size (n)") + annotate(geom="text",
                                              label="truth",
                                              x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

par(mfrow=c(1,1))
red.cdata <- subset(cdata, N<2500)
plot(width.ti ~ N, type = "b", col = "white",
     ylab = "width of CI, Pi and TI",
     xlab = "sample size (n)",
     #main = "Width of intervals as a function of sample size",
     data = red.cdata,
     ylim = c(0,80) )
points(cdata$N,cdata$width.ti, col = "black")
lines(cdata$N,cdata$width.ti, col = "black")
points(cdata$N,cdata$width.pi, col = "red")
lines(cdata$N,cdata$width.pi, col = "red")
points(cdata$N,cdata$width.ci, col = "blue")
lines(cdata$N,cdata$width.ci, col = "blue")

legend("topright", c("TI", "PI", "CI"), bty = "n",
      lty = c(1, 1, 1), lwd = 3, col = c("black", "red", "blue"))

contri <- as.data.frame(cbind(cdata$N,cdata$mean,cdata$sd,
                             cdata$low.ci,cdata$up.ci,
                             cdata$low.pi,cdata$up.pi,
                             cdata$low.ti,cdata$up.ti))
names(contri) <- c("sample size n", "mean", "sd",
                  "low ci", "up ci",
                  "low pi", "up pi",
                  "low ti", "up ti")

contrit <- xtable(contri, caption= "(1-\\alpha = 0.95)$
confidence intervals, (1-\\alpha = 0.95)$ prediction
intervals for one future observation (m=1)$, and
(1-\\alpha = 0.95, P = 0.9)$ tolerance intervals
with increasing sample size $n$. The samples have
been drawn from a normal distribution with a true
mean of 100 and a standard deviation of 10 and
subsequently subsetted to obtain samples ranging
from $n=10$ to $n=10^4$.",label = "contrast",
digits = c(0, 0, 2, 2, 2, 2, 2, 2, 2, 2))

print(contrit, include.rownames=F)

set.seed(3105)
b.6 <- rbinom(1000000,1,0.5)

```



```

bdata$up.b.pi <- lbu/bdata$N

bdata$low.bnrm.pi <- lbl
bdata$up.bnrm.pi <- lbu

l1 <- as.numeric(my.Jeffrey.PI.function(3,10,10,0.05)[1])/10
l2 <- as.numeric(my.Jeffrey.PI.function(11,25,25,0.05)[1])/25
l3 <- as.numeric(my.Jeffrey.PI.function(20,50,50,0.05)[1])/50
l4 <- as.numeric(my.Jeffrey.PI.function(28,75,75,0.05)[1])/75
l5 <- as.numeric(my.Jeffrey.PI.function(40,100,100,0.05)[1])/100
l6 <- as.numeric(my.Jeffrey.PI.function(120,250,250,0.05)[1])/250
l7 <- as.numeric(my.Jeffrey.PI.function(234,500,500,0.05)[1])/500
l8 <- as.numeric(my.Jeffrey.PI.function(375,750,750,0.05)[1])/750
l9 <- as.numeric(my.Jeffrey.PI.function(508,1000,1000,0.05)[1])/1000
l10 <-as.numeric(my.Jeffrey.PI.function(1279,2500,2500,0.05)[1])/2500
l11 <-as.numeric(my.Jeffrey.PI.function(2517,5000,5000,0.05)[1])/5000
l12 <-as.numeric(my.Jeffrey.PI.function(3788,7500,7500,0.05)[1])/7500
l13 <-as.numeric(my.Jeffrey.PI.function(4993,10000,10000,0.05)[1])/10000

u1 <- as.numeric(my.Jeffrey.PI.function(3,10,10,0.05)[2])/10
u2 <- as.numeric(my.Jeffrey.PI.function(11,25,25,0.05)[2])/25
u3 <- as.numeric(my.Jeffrey.PI.function(20,50,50,0.05)[2])/50
u4 <- as.numeric(my.Jeffrey.PI.function(28,75,75,0.05)[2])/75
u5 <- as.numeric(my.Jeffrey.PI.function(40,100,100,0.05)[2])/100
u6 <- as.numeric(my.Jeffrey.PI.function(120,250,250,0.05)[2])/250
u7 <- as.numeric(my.Jeffrey.PI.function(234,500,500,0.05)[2])/500
u8 <- as.numeric(my.Jeffrey.PI.function(375,750,750,0.05)[2])/750
u9 <- as.numeric(my.Jeffrey.PI.function(508,1000,1000,0.05)[2])/1000
u10 <-as.numeric(my.Jeffrey.PI.function(1279,2500,2500,0.05)[2])/2500
u11 <-as.numeric(my.Jeffrey.PI.function(2517,5000,5000,0.05)[2])/5000
u12 <-as.numeric(my.Jeffrey.PI.function(3788,7500,7500,0.05)[2])/7500
u13 <-as.numeric(my.Jeffrey.PI.function(4993,10000,10000,0.05)[2])/10000

bdata$prJ.l <- c(l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13)
bdata$prJ.u <- c(u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13)

l1n <- as.numeric(my.Jeffrey.PI.function(3,10,10,0.05)[1])
l2n <- as.numeric(my.Jeffrey.PI.function(11,25,25,0.05)[1])
l3n <- as.numeric(my.Jeffrey.PI.function(20,50,50,0.05)[1])
l4n <- as.numeric(my.Jeffrey.PI.function(28,75,75,0.05)[1])
l5n <- as.numeric(my.Jeffrey.PI.function(40,100,100,0.05)[1])
l6n <- as.numeric(my.Jeffrey.PI.function(120,250,250,0.05)[1])
l7n <- as.numeric(my.Jeffrey.PI.function(234,500,500,0.05)[1])
l8n <- as.numeric(my.Jeffrey.PI.function(375,750,750,0.05)[1])
l9n <- as.numeric(my.Jeffrey.PI.function(508,1000,1000,0.05)[1])
l10n <-as.numeric(my.Jeffrey.PI.function(1279,2500,2500,0.05)[1])
l11n <-as.numeric(my.Jeffrey.PI.function(2517,5000,5000,0.05)[1])
l12n <-as.numeric(my.Jeffrey.PI.function(3788,7500,7500,0.05)[1])
l13n <-as.numeric(my.Jeffrey.PI.function(4993,10000,10000,0.05)[1])

u1n <- as.numeric(my.Jeffrey.PI.function(3,10,10,0.05)[2])

```

```

u2n <- as.numeric(my.Jeffrey.PI.function(11,25,25,0.05)[2])
u3n <- as.numeric(my.Jeffrey.PI.function(20,50,50,0.05)[2])
u4n <- as.numeric(my.Jeffrey.PI.function(28,75,75,0.05)[2])
u5n <- as.numeric(my.Jeffrey.PI.function(40,100,100,0.05)[2])
u6n <- as.numeric(my.Jeffrey.PI.function(120,250,250,0.05)[2])
u7n <- as.numeric(my.Jeffrey.PI.function(234,500,500,0.05)[2])
u8n <- as.numeric(my.Jeffrey.PI.function(375,750,750,0.05)[2])
u9n <- as.numeric(my.Jeffrey.PI.function(508,1000,1000,0.05)[2])
u10n <- as.numeric(my.Jeffrey.PI.function(1279,2500,2500,0.05)[2])
u11n <- as.numeric(my.Jeffrey.PI.function(2517,5000,5000,0.05)[2])
u12n <- as.numeric(my.Jeffrey.PI.function(3788,7500,7500,0.05)[2])
u13n <- as.numeric(my.Jeffrey.PI.function(4993,10000,10000,0.05)[2])

bdata$prJ.ln <- c(l1n,l2n,l3n,l4n,l5n,l6n,l7n,l8n,l9n,l10n,l11n,l12n,l13n)
bdata$prJ.un <- c(u1n,u2n,u3n,u4n,u5n,u6n,u7n,u8n,u9n,u10n,u11n,u12n,u13n)

b2 <- bintol.int(x = 5, n = 10, bdata$N, alpha = 0.05, P = 0.9, side = 2,
  method = "JF")
bdata <- as.data.frame(bdata)

lt <- b2[1:13,4]
ut <- b2[1:13,5]
bdata$low.ti <- lt/bdata$N
bdata$up.ti <- ut/bdata$N
bdata$lt <- b2[1:13,4]
bdata$ut <- b2[1:13,5]

plot10 <- ggplot(bdata, aes(x=samples, y = Pr, group = samples))
+ ggtitle("CrI J") +
  #draws the means
  geom_point() +
  #draws the CI error bars
  geom_errorbar(data=bdata,
    aes(ymin= low.cri, ymax=up.cri,
      color=samples), width=.5) + ylim(-0.25, 1.25)

h <- 0.5
plot20 <- plot10 + geom_hline(aes(yintercept=h))

pa <- plot20 + labs(x = "sample size (n)") + annotate(geom="text",
  label="", x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

plot11 <- ggplot(bdata, aes(x=samples, y = Pr, group = samples))
+ ggtitle("PI N") +
  #draws the means
  geom_point() +
  #draws the CI error bars
  geom_errorbar(data=bdata,
    aes(ymin= low.b.pi, ymax=up.b.pi,
      color=samples), width=.5) + ylim(-0.25, 1.25)

```

```

h <- 0.5
plot21 <- plot11 + geom_hline(aes(yintercept=h))

pb <- plot21 + labs(x = "sample size (n,m)") + annotate(geom="text",
  label="", x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

plot12 <- ggplot(bdata, aes(x=samples, y = Pr, group = samples))
+ ggtitle("PI J") +
  #draws the means
  geom_point(position=pd) +
  #draws the CI error bars
  geom_errorbar(data=bdata,
    aes(ymin= prJ.l, ymax=prJ.u,
      color=samples), width=.5) + ylim(-0.25, 1.25)

h <- 0.5
plot22 <- plot12 + geom_hline(aes(yintercept=h))

pc <- plot22 + labs(x = "sample size (n,m)") + annotate(geom="text",
  label="", x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

plot13 <- ggplot(bdata, aes(x=samples, y = Pr, group = samples))
+ ggtitle("TI J") +
  #draws the means
  geom_point() +
  #draws the CI error bars
  geom_errorbar(data=bdata,
    aes(ymin= low.ti, ymax=up.ti,
      color=samples), width=.5) + ylim(-0.25, 1.25)

h <- 0.5
plot23 <- plot13 + geom_hline(aes(yintercept=h))

pd <- plot23 + labs(x = "sample size (n)") + annotate(geom="text",
  label="", x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

grid.arrange(pa + theme(axis.text.x = element_blank()),
  pb + theme(axis.text.x = element_blank()),
  pc + theme(axis.text.x = element_blank()),
  pd + theme(axis.text.x = element_blank()),
  ncol = 2, nrow = 2)

set.seed(3105)
b.6 <- rbinom(1000000,1,0.1)
b.10000 <- b.6[1:10000]
b.7500 <- b.6[1:7500]
b.5000 <- b.6[1:5000]
b.2500 <- b.6[1:2500]

```

[illegible]

```

ddata$up.b.pi <- lbu/ddata$N

ddata$low.bnorm.pi <- lbl
ddata$up.bnorm.pi <- lbu

ddata

l1 <- as.numeric(my.Jeffrey.PI.function(1,10,10,0.05)[1])/10
l2 <- as.numeric(my.Jeffrey.PI.function(3,25,25,0.05)[1])/25
l3 <- as.numeric(my.Jeffrey.PI.function(4,50,50,0.05)[1])/50
l4 <- as.numeric(my.Jeffrey.PI.function(5,75,75,0.05)[1])/75
l5 <- as.numeric(my.Jeffrey.PI.function(8,100,100,0.05)[1])/100
l6 <- as.numeric(my.Jeffrey.PI.function(23,250,250,0.05)[1])/250
l7 <- as.numeric(my.Jeffrey.PI.function(53,500,500,0.05)[1])/500
l8 <- as.numeric(my.Jeffrey.PI.function(79,750,750,0.05)[1])/750
l9 <- as.numeric(my.Jeffrey.PI.function(103,1000,1000,0.05)[1])/1000
l10 <-as.numeric(my.Jeffrey.PI.function(252,2500,2500,0.05)[1])/2500
l11 <-as.numeric(my.Jeffrey.PI.function(481,5000,5000,0.05)[1])/5000
l12 <-as.numeric(my.Jeffrey.PI.function(732,7500,7500,0.05)[1])/7500
l13 <-as.numeric(my.Jeffrey.PI.function(999,10000,10000,0.05)[1])/10000

u1 <- as.numeric(my.Jeffrey.PI.function(1,10,10,0.05)[2])/10
u2 <- as.numeric(my.Jeffrey.PI.function(3,25,25,0.05)[2])/25
u3 <- as.numeric(my.Jeffrey.PI.function(4,50,50,0.05)[2])/50
u4 <- as.numeric(my.Jeffrey.PI.function(5,75,75,0.05)[2])/75
u5 <- as.numeric(my.Jeffrey.PI.function(8,100,100,0.05)[2])/100
u6 <- as.numeric(my.Jeffrey.PI.function(23,250,250,0.05)[2])/250
u7 <- as.numeric(my.Jeffrey.PI.function(53,500,500,0.05)[2])/500
u8 <- as.numeric(my.Jeffrey.PI.function(79,750,750,0.05)[2])/750
u9 <- as.numeric(my.Jeffrey.PI.function(103,1000,1000,0.05)[2])/1000
u10 <-as.numeric(my.Jeffrey.PI.function(252,2500,2500,0.05)[2])/2500
u11 <-as.numeric(my.Jeffrey.PI.function(481,5000,5000,0.05)[2])/5000
u12 <-as.numeric(my.Jeffrey.PI.function(732,7500,7500,0.05)[2])/7500
u13 <-as.numeric(my.Jeffrey.PI.function(999,10000,10000,0.05)[2])/10000

ddata$prJ.l <- c(l1,l2,l3,l4,l5,l6,l7,l8,l9,l10,l11,l12,l13)
ddata$prJ.u <- c(u1,u2,u3,u4,u5,u6,u7,u8,u9,u10,u11,u12,u13)

l1n <- as.numeric(my.Jeffrey.PI.function(1,10,10,0.05)[1])
l2n <- as.numeric(my.Jeffrey.PI.function(3,25,25,0.05)[1])
l3n <- as.numeric(my.Jeffrey.PI.function(4,50,50,0.05)[1])
l4n <- as.numeric(my.Jeffrey.PI.function(5,75,75,0.05)[1])
l5n <- as.numeric(my.Jeffrey.PI.function(8,100,100,0.05)[1])
l6n <- as.numeric(my.Jeffrey.PI.function(23,250,250,0.05)[1])
l7n <- as.numeric(my.Jeffrey.PI.function(53,500,500,0.05)[1])
l8n <- as.numeric(my.Jeffrey.PI.function(79,750,750,0.05)[1])
l9n <- as.numeric(my.Jeffrey.PI.function(103,1000,1000,0.05)[1])
l10n <-as.numeric(my.Jeffrey.PI.function(252,2500,2500,0.05)[1])
l11n <-as.numeric(my.Jeffrey.PI.function(481,5000,5000,0.05)[1])
l12n <-as.numeric(my.Jeffrey.PI.function(732,7500,7500,0.05)[1])

```

```

l13n <- as.numeric(my.Jeffrey.PI.function(999,10000,10000,0.05)[1])

u1n <- as.numeric(my.Jeffrey.PI.function(1,10,10,0.05)[2])
u2n <- as.numeric(my.Jeffrey.PI.function(3,25,25,0.05)[2])
u3n <- as.numeric(my.Jeffrey.PI.function(4,50,50,0.05)[2])
u4n <- as.numeric(my.Jeffrey.PI.function(5,75,75,0.05)[2])
u5n <- as.numeric(my.Jeffrey.PI.function(8,100,100,0.05)[2])
u6n <- as.numeric(my.Jeffrey.PI.function(23,250,250,0.05)[2])
u7n <- as.numeric(my.Jeffrey.PI.function(53,500,500,0.05)[2])
u8n <- as.numeric(my.Jeffrey.PI.function(79,750,750,0.05)[2])
u9n <- as.numeric(my.Jeffrey.PI.function(103,1000,1000,0.05)[2])
u10n <- as.numeric(my.Jeffrey.PI.function(252,2500,2500,0.05)[2])
u11n <- as.numeric(my.Jeffrey.PI.function(481,5000,5000,0.05)[2])
u12n <- as.numeric(my.Jeffrey.PI.function(732,7500,7500,0.05)[2])
u13n <- as.numeric(my.Jeffrey.PI.function(999,10000,10000,0.05)[2])

ddata$prJ.ln <- c(l1n,l2n,l3n,l4n,l5n,l6n,l7n,l8n,l9n,l10n,l11n,l12n,l13n)
ddata$prJ.un <- c(u1n,u2n,u3n,u4n,u5n,u6n,u7n,u8n,u9n,u10n,u11n,u12n,u13n)

d2 <- bintol.int(x = 1, n = 10, ddata$N, alpha = 0.05, P = 0.9,
                side = 2,
                method = "JF")
d2

lt <- d2[1:13,4]
ut <- d2[1:13,5]

ddata <- as.data.frame(ddata)
ddata$low.ti <- lt/ddata$N
ddata$up.ti <- ut/ddata$N
ddata$lt <- b2[1:13,4]
ddata$ut <- b2[1:13,5]

plot10 <- ggplot(ddata, aes(x=samples, y = Pr, group = samples))
+ ggtitle("CrI J") +
  #draws the means
  geom_point() +
  #draws the CI error bars
  geom_errorbar(data=ddata,
               aes(ymin= low.cri, ymax=up.cri,
                   color=samples), width=.5) + ylim(-0.25, 1.25)
h <- 0.1
plot20 <- plot10 + geom_hline(aes(yintercept=h))

pf <- plot20 + labs(x = "sample size (n)") + annotate(geom="text",
  label="", x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")

```

```

plot11 <- ggplot(ddata, aes(x=samples, y = Pr, group = samples))
+ ggtitle("PI N") +
  #draws the means
  geom_point() +
  #draws the CI error bars
  geom_errorbar(data=ddata,
                aes(ymin= low.b.pi, ymax=up.b.pi,
                    color=samples), width=.5) + ylim(-0.25, 1.25)

h <- 0.1
plot21 <- plot11 + geom_hline(aes(yintercept=h))

pg <- plot21 + labs(x = "sample size (n,m)") + annotate(geom="text",
  label="", x=1.5, y=h, vjust=-0.2) + theme(legend.position="none")


plot12 <- ggplot(ddata, aes(x=samples, y = Pr, group = samples))
+ ggtitle("PI J") +
  #draws the means
  geom_point() +
  #draws the CI error bars
  geom_errorbar(data=ddata,
                aes(ymin= prJ.l, ymax= prJ.u,
                    color=samples), width=.5) + ylim(-0.25, 1.25)

h <- 0.1
plot22 <- plot12 + geom_hline(aes(yintercept=h))

ph <- plot22 + labs(x = "sample size (n,m)") + annotate(geom="text",
  label="",
  x=1.5,

plot13 <- ggplot(ddata, aes(x=samples, y = Pr, group = samples)) +
  ggtitle("TI J") +
  #draws the means
  geom_point() +
  #draws the CI error bars
  geom_errorbar(data=ddata,
                aes(ymin= low.ti, ymax=up.ti,
                    color=samples), width=.5) + ylim(-0.25, 1.25)

h <- 0.1
plot23 <- plot13 + geom_hline(aes(yintercept=h))

pi <- plot23 + labs(x = "sample size (n)") + annotate(geom="text",
  label="", x=1.5, y=h, vjust=-0.2)
+ theme(legend.position="none")

grid.arrange(pf + theme(axis.text.x = element_blank()),
  pg + theme(axis.text.x = element_blank()),
  ph + theme(axis.text.x = element_blank()),
  pi + theme(axis.text.x = element_blank()),

```



```

ncol = 2, nrow = 2)

contri2 <- as.data.frame(cbind(bdata$N,bdata$R,bdata$Pr,
                              bdata$low.cri,bdata$up.cri,
                              bdata$low.b.pi,bdata$up.b.pi,
                              bdata$prJ.l,bdata$prJ.u,
                              bdata$low.ti,bdata$up.ti))
names(contri2) <- c("n", "x", "Pr",
                   "low cri", "up cri",
                   "low pi N", "up pi N",
                   "low pi J", "up pi J",
                   "low ti J", "up ti J")

contrit2 <- xtable(contri2, caption= "(1-\\alpha = 0.95)$
credible, (1-\\alpha = 0.95)$ prediction and
(1-\\alpha = 0.95, P = 0.9)$ tolerance intervals
with increasing sample size $n (= m)$ for binomial
samples with a true proportion of 0.5. $n$ and $x$
are the number of trials and successes, respectively.
$Pr$ is the estimated proportion, 'low and up cri'
correspond to the credible intervals based on
Jeffreys approach, 'low and up pi N' are the
predictive intervals based on a normal approximation,
'low and up pi J' are predictive intervals based on
Jeffreys approach and 'low and up ti' are the
tolerance intervals obtained by the function
\\textit{bintol.int()} specifying Jeffreys
approach of the \\texttt{tolerance} package.",
label = "contrast2",
digits = c(0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2))

print(contrit2, include.rownames=F)

contri2a <- as.data.frame(cbind(bdata$N,bdata$R,bdata$Pr,
                              bdata$low.bnorm.pi,bdata$up.bnorm.pi,
                              bdata$prJ.ln,bdata$prJ.un,
                              bdata$lt,bdata$ut))
names(contri2a) <- c("n", "x", "Pr",
                   "low pi N", "up pi N",
                   "low pi J", "up pi J",
                   "low ti J", "up ti J")

contrit2a <- xtable(contri2a, caption= "(1-\\alpha = 0.95)$
prediction and (1-\\alpha = 0.95, P = 0.9)$
tolerance intervals with increasing sample size
$n (= m)$ for binomial samples with a true
proportion of 0.5. Presented are the number of
successes $(m*p)$.", label = "contrast2a",

```

```

        digits = c(0, 0, 0, 2, 0, 0, 0, 0, 0, 0))

print(contrit2a, include.rownames=F)

contrit3 <- as.data.frame(cbind(ddata$N,ddata$R,ddata$Pr,
                                ddata$low.cri,ddata$up.cri,
                                ddata$low.b.pi,ddata$up.b.pi,
                                ddata$prJ.l,ddata$prJ.u,
                                ddata$low.ti,ddata$up.ti))
names(contrit3) <- c("n", "x", "Pr",
                    "low cri", "up cri",
                    "low pi N", "up pi N",
                    "low pi J", "up pi J",
                    "low ti J", "up ti J")

contrit3 <- xtable(contrit3, caption= "Comparison of  $(1-\alpha = 0.95)$ 
credible,  $(1-\alpha = 0.95)$  prediction and
 $(1-\alpha = 0.95, P = 0.9)$  tolerance intervals
with increasing sample size  $n$  ( $= m$ ) for binomial
samples with a true proportion of 0.1.  $n$  and  $x$ 
are the number of trials and successes, respectively.
 $Pr$  is the estimated proportion, 'low and up cri'
correspond to the credible intervals based on
Jeffreys approach, 'low and up pi N' are the predictive
intervals based on a normal approximation, 'low and up pi J'
are predictive intervals based on Jeffreys approach
and 'low and up ti' are the tolerance intervals obtained
by the function bintol.int() specifying
Jeffreys approach of the tolerance package.",
label = "contrast3",
digits = c(0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2))

print(contrit3, include.rownames=F)

contrit3a <- as.data.frame(cbind(ddata$N,ddata$R,ddata$Pr,
                                ddata$low.bnorm.pi,ddata$up.bnorm.pi,
                                ddata$prJ.ln,ddata$prJ.un,
                                ddata$lt,ddata$ut))

names(contrit3a) <- c("n", "x", "Pr",
                    "low pi N", "up pi N",
                    "low pi J", "up pi J",
                    "low ti J", "up ti J")

contrit3a <- xtable(contrit3a, caption= "Comparison of  $(1-\alpha = 0.95)$ 
prediction and  $(1-\alpha = 0.95, P = 0.9)$  tolerance
intervals with increasing sample size  $n$  ( $= m$ ) for
binomial samples with a true proportion of 0.5. Presented
are the number of successes  $(m \cdot p)$ .",

```

```

        label ="contrast3a",
        digits = c(0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0))

print(contrit3a, include.rownames=F)

bdata <- as.data.frame(bdata)
bdata$width.ci <- bdata$up.cri - bdata$low.cri
bdata$width.pin <- bdata$up.b.pi - bdata$low.b.pi
bdata$width.piJ <- bdata$prJ.u - bdata$prJ.l
bdata$width.ti <- bdata$up.ti - bdata$low.ti

ddata <- as.data.frame(ddata)
ddata$width.ci <- ddata$up.cri - ddata$low.cri
ddata$width.pin <- ddata$up.b.pi - ddata$low.b.pi
ddata$width.piJ <- ddata$prJ.u - ddata$prJ.l
ddata$width.ti <- ddata$up.ti - ddata$low.ti

par(mfrow=c(2,1))
red.bdata <- subset(bdata, N<2500)
plot(width.ti ~ N, type = "b", col = "white",
      ylab = "width of CrI, PI and TI",
      xlab = "sample size (n)",
      main = "Samples based on a binomial distribution with a
      true proportion of 0.5",
      data =red.bdata,
      ylim = c(-0.125,1.125) )
points(bdata$N,bdata$width.ti, col = "black")
lines(bdata$N,bdata$width.ti, col = "black")
points(bdata$N,bdata$width.pin, col = "red")
lines(bdata$N,bdata$width.pin, col = "red")
points(bdata$N,bdata$width.piJ, col = "green")
lines(bdata$N,bdata$width.piJ, col = "green")
points(bdata$N,bdata$width.ci, col = "blue")
lines(bdata$N,bdata$width.ci, col = "blue")

red.ddata <- subset(ddata, N<2500)
plot(width.ti ~ N, type = "b", col = "white",
      ylab = "width of CrI, PI and TI",
      xlab = "sample size (n)",
      main = "Samples based on a binomial distribution with a
      true proportion of 0.1",
      data =red.ddata,
      ylim = c(-0.125,1.125) )
points(ddata$N,ddata$width.ti, col = "black")
lines(ddata$N,ddata$width.ti, col = "black")
points(ddata$N,ddata$width.pin, col = "red")
lines(ddata$N,ddata$width.pin, col = "red")
points(ddata$N,ddata$width.piJ, col = "green")
lines(ddata$N,ddata$width.piJ, col = "green")

```

```
points(ddata$N,ddata$width.ci, col = "blue")
lines(ddata$N,ddata$width.ci, col = "blue")

legend("topright", c("TI Jeffreys", "PI norm", "PI Jeffreys", "CrI Jeffreys"),
      bty = "n",
      lty = c(1, 1, 1,1), lwd = 3, col = c("black", "red", "green", "blue"))
```

Bibliography

- Agresti, A. and Coull, B. (1998). Approximate is better than Exact for interval estimation of binomial proportions. *The American Statistician*, **52**, 119–126. [11](#)
- Altman, D. and Bland, M. (2005). Standard deviations and standard errors. *BMJ*, **331**, 903. [7](#)
- Brown, L., Cai, T., and A., D. (2001). Interval estimation for a binomial proportion. *Statistical Science*, **16**, 101–133. [12](#), [13](#), [14](#), [29](#), [44](#)
- Clopper, C. and Pearson, E. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, **26**, 404–413. [11](#), [14](#)
- Flanagan, J., Bissot, T., M.-A., H., Moreno, B., Feugier, A., and German, A. (2017). Success of a weight loss plan for overweight dogs: The results of an international weight loss study. *PloS ONE*, **12**, e0184199. [6](#), [8](#), [9](#), [10](#), [26](#), [28](#)
- Gitlow, H. and Awad, H. (2013). Intro stats students need both confidence and tolerance (intervals). *The American Statistician*, **67**, 229–234. [31](#)
- Held, L. and Sabanés Bové, D. (2014). *Applied Statistical Inference*. Springer, Heidelberg. [7](#), [13](#)
- Howe, W. (1969). Two-sided tolerance limits for normal populations - some improvements. *Journal of the American Statistical Association*, **64**, 610–620. [27](#)
- Meeker, W. and Hahn, G.J. and Escobar, L. (2017). *Statistical Intervals. A guide for practitioners and researchers*. Wiley, New Jersey. [13](#), [20](#), [22](#), [36](#), [39](#)
- Neyman, J. (1937). Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society A*, **236**, 333–380. [7](#)
- Scheffé, H. and Tukey, J. (1944). A formula for sample sizes for population tolerance intervals. *The Annals of Mathematical Statistics*, **15**, 217. [27](#)
- Sprick, M., Furst, A., Baschnagel, F., Michel, S., Piskoty, G., Hartnack, S., and Jackson, M. (2017). The influence of aluminium, steel and polyurethane shoeing systems and of the unshod hoof on the injury risk of a horse kick. an ex vivo experimental study. *Veterinary and Comparative Orthopaedics and Traumatology*, **12**, 339–345. [6](#), [14](#), [16](#), [25](#)
- Wilson, E. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, **22**, 209–212. [12](#)
- Young, D. (2010). tolerance: An R package for estimating tolerance intervals. *Journal of Statistical Software*, **36**, . [26](#), [27](#), [28](#), [44](#)
- Young, D. (2016). Normal tolerance interval procedures in the tolerance package. *The R Journal*, **8**, 2. [26](#)

List of Figures

2.1	Illustration of confidence intervals	8
2.2	Boxplot of percentage of body weight loss in dogs	9
2.3	Histograms of percentage of body weight loss in dogs	9
2.4	Credible intervals: prior and posterior distributions	16
3.1	Illustration of prediction intervals.	18
3.2	Prediction intervals: prior and posterior predictive distributions.	25
4.1	Illustration of tolerance intervals	29
5.1	Histograms of scenarios A, B and C	32
5.2	Effects of sample size on the width of normal confidence intervals	34
5.3	Effects of sample size on the width of normal prediction intervals	35
5.4	Effects of size of m on the width of normal prediction intervals	36
5.5	Effects of sample size on the width of normal tolerance intervals	37
5.6	Effect of sample size on the width of normal confidence, prediction and tolerance intervals	38
5.7	Effects of sample size on the width of intervals for a binomial distribution ($P = 0.5$)	40
5.8	Effects of sample size on the width of intervals for a binomial distribution ($P = 0.1$)	41
5.9	Effect of sample size on the width of binomial confidence, prediction and tolerance intervals	43