Chapter 5
Ranking and Result Aggregation

Thomas Bartz-Beielstein, Olaf Mersmann, and Sowmya Chandrasekaran

Abstract This chapter explores different methods to analyze the results of Hyperparameter Tuning (HPT) experiments. Four different scenarios and two different approaches are presented. On the one hand, rankings and especially consensus rankings are introduced to aggregate the results of many different HPT results. On the other hand, statistical significance analysis and power analysis are used for a detailed analysis of single algorithms and pairwise algorithm comparisons. This chapter discusses issues with sample size determination, power calculations, hypotheses, and wrong conclusions from hypothesis testing. On top of the established methods, we add and explain severity, a frequentist approach that extends the classical concept of p-values. Mayo's concept of severity offers one solution to these issues, and one might achieve even better results by applying severity.

5.1 Comparing Algorithms

Aggregating the results of any kind of hyperparameter tuning or other large-scale modeling experiment poses its own set of challenges. Generally, we can differentiate between four settings (Bartz-Beielstein and Preuss 2011):

Definition 5.1 [Algorithm-Problem Designs]

Single Algorithm Single Problem (SASP): Analyzing the result of a single algorithm or learner on a single optimization problem or data set.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-981-19-5170-1_5.

T. Bartz-Beielstein (✉) · O. Mersmann · S. Chandrasekaran
Institute for Data Science, Engineering and Analytics, TH Köln, Gummersbach, Germany
e-mail: thomas.bartz-beielstein@th-koeln.de

O. Mersmann
e-mail: olaf.mersmann@th-koeln.de

S. Chandrasekaran
e-mail: sowmya.chandrasekaran@th-koeln.de

E. Bartz et al. (eds.), *Hyperparameter Tuning for Machine and Deep Learning with R*,
https://doi.org/10.1007/978-981-19-5170-1_5

Single Algorithm Multiple Problem (SAMP): Comparing the results of a single algorithm or learner on many different optimization problems or data sets.
Multiple Algorithm Single Problem (MASP): Comparing the results of multiple algorithms or learners on a single optimization problem or data set.
Multiple Algorithm Multiple Problem (MAMP): Comparing the results of multiple algorithms or learners on many different optimization problems or data sets.

The SASP setting is fundamentally different from the other three settings, because we are not *comparing* results but merely analyzing them. That is, we are evaluating the performance of an *optimization algorithm* $\mathcal{A}$ on a single problem instance π. In the second scenario, we have multiple problem instances $\pi_1, \ldots, \pi_p$. That means, the second setting is a generalization of the first setting, where we might want to check if our algorithm generalizes to different instances from the same domain or even generalizes to different domains. The third setting generalizes the first by introducing more algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_a$. Here, we want to compare the performance of these algorithms on a single problem instance and more than likely choose a "best" algorithm. Finally, the last scenario is a combination of the previous two, where we have a algorithms being benchmarked on p problem instances.

For now, we will ignore the challenges posed by the SASP and SAMP settings and focus on the comparison of multiple algorithms. We will denote the random performance measure we use to evaluate an algorithm with Y. Even for deterministic algorithms, it is justified to view this as a random variable since the result still heavily depends on the initial starting parameters, etc. We will assume that we have collected n Independent and Identically Distributed (IID) samples of our performance measure Y for each algorithm and performance metric. These are denoted with $y_1, \ldots, y_n$.

During all of the following discussions on comparing algorithms, we should always remember that the No Free Lunch theorem (Wolpert and Macready 1997) tells us there is no single best algorithm in both the learning and the optimization setting. We are interested in *comparing* algorithms and *choosing* one that is fit for purpose; we cannot hope to find a single "best" algorithm.

5.2 Ranking

When we are in the MASP setting, there are many established statistical frameworks to analyze the observed performance metrics; see for example Chiarandini and Goegebeur (2010) or Bartz-Beielstein (2015). Here, we will look at a somewhat different approach based on rankings as described in Mersmann et al. (2015). The advantage of ranking-based approaches is their scale invariance.

Consider the case where we have only two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$. For each algorithm, we observe n values of our performance metric

Algorithm $\mathcal{A}_1$: $y_1^{\mathcal{A}_1}, \ldots, y_n^{\mathcal{A}_1}$
Algorithm $\mathcal{A}_2$: $y_1^{\mathcal{A}_2}, \ldots, y_n^{\mathcal{A}_2}$

and we want to decide if $\mathcal{A}_1$ is

1. "better than or equal to" $\mathcal{A}_2$ (denoted by $\mathcal{A}_1 \succ \mathcal{A}_2$);
2. "similar to" $\mathcal{A}_2$ (denoted by $\mathcal{A}_1 \simeq \mathcal{A}_2$);
3. "worse than" $\mathcal{A}_2$ (denoted by $\mathcal{A}_1 \prec \mathcal{A}_2$).

Saying $\mathcal{A}$ is worse than $\mathcal{B}$ is nothing more than saying $\mathcal{B}$ is better than or equal to $\mathcal{A}$:

$$\mathcal{A} \prec \mathcal{B} \iff \mathcal{B} \succ \mathcal{A}.$$

We can also simplify when we consider two algorithms to be similar. We say two algorithms are similar if both are better than or equal to the other one:

$$\mathcal{A} \simeq \mathcal{B} \iff \mathcal{A} \succ \mathcal{B} \wedge \mathcal{B} \succ \mathcal{A}.$$

Therefore, it is enough to specify the binary relation $\succ$ if we want to decide if some algorithm *dominates* another algorithm. We call $\succ$ the *dominance relation* for our performance metric. One way would be using statistical hypothesis tests as discussed in Sect. 5.6.1 but if n is large,[1] it can be something as simple as the comparison of the mean performance measure attained by each algorithm. It is also possible to think of scenarios where we might be more interested in a consistent result. In these cases, we might compare the variance of the observed performance measures. Finally, if we are really only interested in the absolute best performance the algorithm can deliver, we should compare the minimal or maximal performance measure obtained. For a more detailed description of the different choices available, see Mersmann et al. (2010b). But for now, let's just assume that we are able to define such a dominance relation.

Our dominance relation can have the following useful properties:

reflexive: $\mathcal{A} \succ \mathcal{A}$ for all $\mathcal{A}$ under test. That is, every algorithm is better than or equal to itself. This is a property we want in any dominance relation.

antisymmetric: $\mathcal{A} \succ \mathcal{B} \wedge \mathcal{B} \succ \mathcal{A} \implies \mathcal{A} \simeq \mathcal{B}$. This is a weaker form of our "similar to" definition above that suffices for our further reasoning.

transitive: $\mathcal{A} \succ \mathcal{B}$ and $\mathcal{B} \succ C$, then $\mathcal{A} \succ C$.

complete: For all distinct pairs of algorithms, either $\mathcal{A} \succ B$ or $\mathcal{B} \succ \mathcal{A}$.

At a minimum, we want our relation to be *reflexive* and *transitive*. We call such a relation a *preorder* and it is the first step toward a relation that induces an order, i.e., gives us a meaningful comparison of all algorithms based on simple pairwise comparisons. Next, we want *antisymmetry* which gives us a *partial order* and finally if the partial order is *complete*, we get a *linear order*. A linear order has quite a few requirements which must be fulfilled. Instead, we could ask ourselves what are the minimum properties we would want? We would certainly want our relation to be *transitive* since otherwise we won't have a ranking, and we also want the relation

[1] And see below for reasons why maybe it shouldn't be too large.

to be *complete* so that we can compare all algorithm pairs. An order with just these properties is called a *weak order* and will become important later in our discussion of rankings.

Let's illustrate what we have so far with an example. Assume we have $a = 5$ algorithms and that we measured the performance of each algorithm $n = 15$ times. We can store these results in a 5×15 matrix. Each row stores the results for one algorithm and each column is one observation of the performance measure.

```
t(Y)
```

```
##            A_1       A_2      A_3      A_4      A_5
##  [1,] 11.25802  9.184456 10.91332 9.699683 9.533216
##  [2,] 11.44358  9.227654 11.13609 9.632939 9.339878
##  [3,] 11.49753  9.979770 10.69170 9.411786 9.480409
##  [4,] 11.45181  9.654419 10.87821 9.883699 9.456854
##  [5,] 11.34973  9.359485 10.86492 9.697134 9.416884
##  [6,] 11.67326  9.681364 10.55226 9.586506 9.064084
##  [7,] 11.35203  9.958682 11.04233 9.623864 9.254671
##  [8,] 11.77464 10.094786 11.07630 9.704412 9.397412
##  [9,] 11.35842  9.629653 10.96289 9.586520 9.082003
## [10,] 11.63193  9.664293 11.18674 8.969936 9.164142
## [11,] 11.82829  9.363114 10.88976 9.625530 9.388907
## [12,] 11.09319  9.807767 10.76145 9.495107 9.426344
## [13,] 11.31520  9.587748 11.12167 9.605600 9.365461
## [14,] 11.34429  9.806837 10.87282 9.684919 9.095553
## [15,] 11.49815  9.856715 11.32392 9.848501 9.052752
```

From these raw results, we could derive the incidence matrix of our dominance relation by comparing the mean performance of each algorithm:

```
I <- matrix(0, nrow(Y), nrow(Y))
rownames(I) <- colnames(I) <- rownames(Y)
for (i in 1:nrow(Y)) {
  for (j in 1:nrow(Y)) {
    I[i, j] <- mean(Y[i, ]) >= mean(Y[j, ])
  }
}
I
```

```
##     A_1 A_2 A_3 A_4 A_5
## A_1   1   1   1   1   1
## A_2   0   1   0   1   1
## A_3   0   1   1   1   1
## A_4   0   0   0   1   1
## A_5   0   0   0   0   1
```

And from that the dominance relation using the `relations` package (Meyer and Hornik 2022):

```
r_mean <- relation(incidence = I)
```

We can now check if it is a preorder, partial order, or a linear order:

```
relation_is_preorder(r_mean)

  ## [1] TRUE

relation_is_partial_order(r_mean)

  ## [1] TRUE

relation_is_linear_order(r_mean)

  ## [1] TRUE

relation_is_weak_order(r_mean)

  ## [1] TRUE
```

Not surprisingly, we find that the relation is indeed a *linear order*. Using a small helper function, we can pretty print the order

```
show_relation <- function(r) {
  classes <- relation_classes(r)
  class_names <- sapply(
    classes,
    function(x) paste0("{", paste(x, collapse = ", "), "}")
  )
  paste(class_names, collapse = " > ")
}

show_relation(r_mean)

  ## [1] "{A_1} > {A_3} > {A_2} > {A_4} > {A_5}"
```

As expected, algorithm $\mathcal{A}_1$ dominates all other algorithms since it has the highest mean performance of 11.4580052.

Let's see what happens if we use a more nuanced approach using hypothesis tests to derive our dominance relation

```
I <- matrix(0, nrow(Y), nrow(Y))
rownames(I) <- colnames(I) <- rownames(Y)
for (i in 1:nrow(Y)) {
  for (j in 1:nrow(Y)) {
    I[i, j] <- if (i != j) {
      t.test(Y[i, ], Y[j, ],
        paired = TRUE,
        alternative = "less"
      )$p.value > 0.05
    } else {
      1
    }
  }
}
r_ht <- relation(incidence = I)
show_relation(r_ht)

   ## [1] "{A_1} > {A_3} > {A_2, A_4} > {A_5}"
```

The resulting dominance relation is not a linear order, because it is not *antisymmetric* since $\mathcal{A}_2 \simeq \mathcal{A}_4$ but $\mathcal{A}_2 \neq \mathcal{A}_4$. It is however still a *weak order* since it is complete and transitive.

```
relation_is_preorder(r_ht)

   ## [1] TRUE

relation_is_partial_order(r_ht)

   ## [1] FALSE

relation_is_linear_order(r_ht)

   ## [1] FALSE

relation_is_weak_order(r_ht)

   ## [1] TRUE
```

While a ranking derived from a dominance relation does not give us as many insights as some of the more advanced techniques based on ANOVA or multiple comparison tests, it does extract the essential information we need. From a ranking, we can derive clear preferences for some algorithm or see that a group of algorithms performs similarly.

The real advantage of the ranking-based approach becomes apparent when we leave the MASP setting and go over to the MAMP setting. We can view the MAMP

setting as p[2] MASP settings. For each problem instance $\pi_i, \ldots, \pi_p$, we can derive a ranking of the algorithms with the above methodology. This amounts to each problem instance voicing its opinion about which algorithm is preferable. Why is this advantageous when compared to direct performance measure calculations? Because in most cases, the *scale* of our performance measure is specific to the problem instance. We cannot compare the performance measure observed on one problem instance with that on another problem instance. What we can compare are the obtained *ranks*. The ranking is scale-invariant and allows us to aggregate the results of many different MASP scenarios into one MAMP comparison.

5.3 Rank Aggregation

Before we dive into aggregation methods for rankings, let's look at a motivating toy example. An ice cream plant is trying to determine the favorite flavor of ice cream for kids. They let three children rank the flavors based on how well they like them and get the following result:

$$\begin{aligned}&\text{chocolate} \succ \text{vanilla} \succ \text{strawberry} \succ \text{cherry} \succ \text{blueberry}\\&\text{vanilla} \succ \text{strawberry} \succ \text{cherry} \succ \text{blueberry} \succ \text{chocolate}\\&\text{strawberry} \succ \text{cherry} \succ \text{blueberry} \succ \text{chocolate} \succ \text{vanilla}\end{aligned}$$

Here, the children are the "problem instances" and the ice cream flavors are the "algorithms" being ranked. If we simply average the rank for each flavor and then rank the flavors based on this average, we get the following (unsurprising) result:

$$\text{vanilla} \succ \text{strawberry} \succ \text{cherry} \succ \text{chocolate} \succ \text{blueberry} \tag{5.1}$$

Since blueberries are expensive and kids seem to dislike them, they rank last. In fact, we might have suspected that and not taken the flavor blueberry into account. If we remove the blueberry flavor from all three rankings and again calculate the average ranking, we get

$$\text{vanilla} \succ \text{strawberry} \succ \text{chocolate} \succ \text{cherry} \tag{5.2}$$

Notice how deleting the least liked flavor from the list resulted in cherry and chocolate switching positions. Surely, this is not the kind of behavior we would want. But in fact, if we remove the other fruit flavor (cherry), we get an average ranking of

$$\text{vanilla} \simeq \text{strawberry} \simeq \text{chocolate} \tag{5.3}$$

There appears to be no clear preference anymore!

[2] Remember p denotes the number of different problem instances in our MAMP setting.

We could also view this as a "fruit conspiracy". Strawberry, cherry, and blueberry, full well knowing that only strawberry has any chance of winning, are in cahoots and all enter the competition. By entering all three fruit flavors, the results seem to be skewed in their favor.

We will see that this is an unfortunate side effect of any so-called Consensus Method (CM) for rankings. We have seen in the previous section that, depending on our choice of dominance relation, we arrive at different rankings of our algorithms under test. This is to be expected, since we are ranking them based on different definitions of what we consider to be a better algorithm. We will see that there are different methods for deriving a consensus from our set of rankings and that methods offer different trade-offs between properties that the consensus fulfills. So, before we have seen the first consensus method, we need to accept the fact that from this point forward, we cannot objectively define the best algorithm. Instead, our statement of which algorithm is best depends on our subjective choice of a CM. But not all hope is lost. What we can define are criteria we would want, an ideal CM to have, and then make an informed choice about the trade-off between these criteria.

1. A CM that takes into account all rankings instead of mimicking one predetermined ranking is said to be *non-dictatorial*.
2. A CM that, given a fixed set of rankings, deterministically returns a complete ranking is called a *universal consensus method* or is said to have a *universal domain*.
3. A CM is *independent of irrelevant alternatives*, if given two sets of rankings $R = r_1, \ldots, r_k$ and $S = s_1, \ldots, s_k$ in which for every $i \in 1, \ldots, k$ the order of two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ in r_i and s_i is the same; the resulting consensus rankings rank $\mathcal{A}_1$ and $\mathcal{A}_2$ in the same order. Essentially, this means that introducing a further algorithm does not lead to a rank reversal between any of the already ranked algorithms. While this might seem highly desirable (see the above ice cream example), it is also a very strict requirement.
4. A CM which ranks an algorithm higher than another algorithm if it is ranked higher in a majority of the individual rankings fulfills the *majority criterion*.
5. A CM is called *Pareto efficient* if given a set of rankings in which for every ranking an algorithm ai is ranked higher than an algorithm aj, the consensus also ranks ai higher than aj.

No consensus method can meet all of these criteria because the independence of irrelevant alternatives (IIA) and the majority criterion are incompatible. But even if we ignore the majority criterion, there is no consensus method which fulfills the remaining criteria (Arrow 1950). So it is not surprising that if we choose different criteria for our CM, we may get very different consensus rankings.

At this point, we might ask ourselves why bother finding a consensus if it is subjective in the end. And to a certain extent that is true, but it still gives us valuable insights into which algorithms might warrant further investigation and which algorithms perform poorly. However, we have to take care that no accidental or intentional manipulation of the consensus takes place. This can easily happen if the IIA is not

fulfilled. Remember how introducing the irrelevant fruit flavors in our toy ice cream example changed the consensus drastically. By adding many similar algorithms or variants of one algorithm, we can skew our analysis and provoke unwanted rank reversals.

Generally, we can differentiate between positional and optimization-based methods. Positional methods calculate sums of scores for each algorithm $\mathcal{A}_i$ over all rankings. The final order is determined by the score obtained by each algorithm. This amounts to

$$\mathcal{A}_i \succ \mathcal{A}_j \iff si >= sj, \qquad \mathcal{A}_i \simeq \mathcal{A}_j \iff si = sj$$

with the score of algorithm $\mathcal{A}_i$ given by

$$s_i = \sum_{k=1}^{p} s(\mathcal{A}_i, r_{\pi_k}).$$

Here, s denotes a score function and r_{π_k} is the ranking inferred from problem instance π_k. The score function takes as arguments an algorithm and a ranking and returns the score of the algorithm in that ranking.

The simplest score function we might use assigns a value of one to the best algorithm in each ranking while all other algorithms get a value of zero. Although this is somewhat intuitive, undesirable consensus rankings can occur. Consider the situation with two different rankings of three algorithms:

$$\mathcal{A}_1 \succ \mathcal{A}_2 \succ \mathcal{A}_3 \quad \text{and} \quad \mathcal{A}_3 \succ \mathcal{A}_2 \succ \mathcal{A}_1.$$

Using the above score function, we would obtain the following scores:

$$s_1 = 1 + 0 = 1 \quad s_2 = 0 + 0 = 0 \quad s_3 = 0 + 1 = 1$$

which leads to the consensus ranking

$$\{\mathcal{A}_1 \simeq \mathcal{A}_2\} \succ \mathcal{A}_2.$$

This is counterintuitive since the two rankings are opposed and we'd expect them to cancel out and give

$$\{\mathcal{A}_1 \simeq \mathcal{A}_2 \simeq \mathcal{A}_2\}.$$

The Borda count method (de Borda 1781) solves this issue and assigns an algorithm one point for each algorithm that is not better than

$$s^{BC}(\mathcal{A}_i, r) = \sum_{i \neq j} \mathbf{I}(\mathcal{A}_i \succ \mathcal{A}_j).$$

In the case of no ties, it reduces the ranks of the data. For our example rankings above, we get

$$s_1 = 2 + 0 = 2 \quad s_2 = 1 + 1 = 2 \quad s_3 = 0 + 2 = 2$$

and the consensus ranking

$$\{\mathcal{A}_1 \simeq \mathcal{A}_2 \simeq \mathcal{A}_2\}$$

which is more intuitive than the previous result. Unfortunately, the Borda method does not fulfill the majority or the IIA criterion. It is still a popular consensus method because it can be easily implemented and understood. The main criticism voiced in the literature is that it implicitly, like all positional consensus methods, assumes a distance between the positions of a ranking.

A completely different approach is to frame the CM as an optimization problem where we want to find a ranking that minimizes a function of the distances to all of the individual rankings. Cook and Kress (1992) give a gentle introduction to this line of thought and present a wide variety of possible distance functions. Central to this is a notion of betweenness, expressed by pairwise comparisons. Here, we will focus on the axiomatically motivated symmetric difference distance function[3] originally proposed by Kemeny and Snell (1962), but the general procedure is the same regardless of the distance function chosen. First, we pick a set C of admissable consensus rankings. This could be the set of all *linear* or *weak orderings* of our algorithms. Then, we solve the following optimization problem:

$$\arg\min_{c \in C} L(c) = \arg\min_{c \in C} \sum_{i=1}^{p} d(c, r_{\pi_i})^{\ell}, \quad \ell \geq 1.$$

Setting $\ell = 1$ results in what is called a median consensus ranking and $\ell = 2$ results in a mean consensus ranking.

Let's revisit the ice cream example and see what the consensus is according to Borda or using the symmetric difference.

```
show_relation(child1)

   ## [1] "{chocolate} > {vanilla} > {strawberry} > {cherry} > {blueberry}"

show_relation(child2)

   ## [1] "{vanilla} > {strawberry} > {cherry} > {blueberry} > {chocolate}"

show_relation(child3)

   ## [1] "{strawberry} > {cherry} > {blueberry} > {chocolate} > {vanilla}"
```

[3] The symmetric difference counts the number of cases where $\mathcal{A}_i \succ \mathcal{A}_j$ is contained in one of the relations but not the other.

The Borda consensus among the three children is

```
ranks <- relation_ensemble(child1, child2, child3)
r_borda <- relation_consensus(ranks, "Borda")
show_relation(r_borda)

   ## [1] "{strawberry} > {vanilla} > {cherry} > {chocolate} > {blueberry}"
```

and the symmetric difference-based consensus among all linear orderings of the flavors is

```
r_sd <- relation_consensus(ranks, "symdiff/L")
show_relation(r_sd)

   ## [1] "{vanilla} > {strawberry} > {cherry} > {blueberry} > {chocolate}"
```

We see that the Borda consensus falls into the "fruit-gang trap" and ranks the strawberry flavor first. The symmetric difference-based consensus on the other hand ranks vanilla higher than strawberry because in two out of three rankings, it ranks higher than strawberry.

Unfortunately, we cannot give a general recommendation regarding the introduced consensus methods as each method offers a different trade-off of the consensus criteria (Saari and Merlin 2000). The symmetric difference combined with linear or weak orderings meet the majority criterion and thus cannot meet the IIA criterion simultaneously. However, on real data, as seen in the ice cream example, they rarely result in rank reversals if algorithms are added or dropped. The Borda count method does not fulfill either of these criteria. Saari and Merlin (2000) however showed that both methods always rank the respective winner above the loser of the other method.

Finally, it is important to note that consensus rankings generally do not admit nesting in a hierarchical structure. For example, separate consensus rankings could be of interest for problem instances with specific features. While this certainly is a valid and meaningful approach, one has to keep in mind that an overall consensus of these separate consensus rankings does not necessarily have to equal the consensus ranking directly generated based on all individual rankings.

5.4 Result Analysis

Many of the Machine Learning (ML) and Deep Learning (DL) methods are stochastic in nature as there is randomness involved as a part of optimization or learning. Hence, these methods could yield different results to the same data for every run. To access the performance of the model, one single evaluation may not be sufficient. To statistically evaluate the variance of the obtained results, multiple repeats have to be performed and the summary statistics of the performance measure are to be reported.

Generally, the performance of the ML and DL methods can be analyzed considering model quality and runtime. The model quality is determined using the Root Mean

Squared Error (RMSE) for the regression models and the Mean Mis-Classification Error (MMCE) for the classification models as discussed in Sect. 2.2.

Often, these quality metrics are compared among different algorithms to analyze their performances. Hence, the tuners aim to minimize these metrics. As these metrics can be affected by the algorithm's and tuner's stochastic nature, the experiment has to be repeated for a specific number of times. It enables better estimation of the model quality parameter using descriptive and Exploratory Data Analysis (EDA) tools. Also, statistical inference is highly recommended in understanding the underlying distribution of the model quality parameters.

EDA is a statistical methodology for analyzing data sets to summarize their main characteristics (Tukey 1977; Chambers et al. 1983). The EDA tools are employed to analyze and report the performance of the ML models. This includes both descriptive and graphical tools. The numerical measures include reporting the *mean*, *median*, *best*, *worst*, and *standard deviation* of the performance measures of the algorithms obtained for certain number of repeats. They measure the central tendency and the variability of the results. The graphical tools like histograms, and box and violin plots provide information about the shape and the distribution of the performance measures, respectively. These statistics are necessary, but are not always sufficient to evaluate the performances. Kleijnen (1997), Bartz-Beielstein et al. (2010), Myers et al. (2016), Montgomery (2017), and Gramacy (2020) are good starting points. More information about various techniques and best practices in analyzing the performance measures can be found in Bartz-Beielstein et al. (2020b).

5.5 Statistical Inference

Statistical inference means drawing conclusions from partial information of a population about the whole population using methods based on data analysis and probability theory. Statistical inference is recommended in making decisions about identifying the best algorithm and tuner. The key ingredient of statistical inference is hypothesis testing (Neyman 1950). As a part of pre-data analysis, the null hypothesis H_0 can be formulated as "There is no statistically significant difference between the compared algorithms", while the alternative hypothesis H_1 states that there exists a statistically significant difference between the compared algorithms. Hypothesis testing will be outlined in Sect. 5.6.1.

The hypothesis testing can be classified into *parametric* and *non-parametric* tests. For the case of parametric tests, the distributional assumptions have to be satisfied, one of which is Normal, Independent and Identically Distributed (NIID) data. If the distributional assumptions are not met, non-parametric tests are employed. For the case of single pairwise comparison, the most commonly used parametric test is the t-test (Sheskin 2003) and its non-parametric counter-part is the Wilcoxon-rank sum test (Hart 2001). And in case of multiple comparisons, one commonly used parametric test is the one-way *ANOVA* (Lindman 1974), while its non-parametric

test counter-part is the *Kruskal-Wallis rank sum test* (Kruskal and Wallis 1952). The following sections analyze parametric tests.

5.6 Definitions

5.6.1 Hypothesis Testing

Generally, hypothesis testing can be either one-sided or two-sided: $H_0 : \tau \leq 0$ versus $H_1 : \tau > 0$ (one-sided) or $H_0 : \tau = 0$ versus $H_1 : \tau \neq 0$ (two-sided), where H_0 and H_1 denote the corresponding hypotheses that will be explained in this section. For the purpose of performance comparison of the two methods, we consider a one-sided test and question whether method $\mathcal{A}$ is better than method $\mathcal{B}$. Let $p(\mathcal{A})$ and $p(\mathcal{B})$ represent the performance of method $\mathcal{A}$ and $\mathcal{B}$, respectively. If we consider a minimization problem, the smaller the values the better the performance of the method. For method $\mathcal{A}$ to be better than method $\mathcal{B}$, $p(\mathcal{A}) < p(\mathcal{B}) \Leftrightarrow p(\mathcal{B}) - p(\mathcal{A}) > 0 \Leftrightarrow \tau > 0$.

To state properties of the hypothesis, the symbol μ will be used for the mean, whereas the symbol τ denotes the difference between two means. For example, $\tau = \mu_1 - \mu_0$ or variations of the mean, e.g., $\tau = \mu + \Delta$.

Definition 5.2 (*One-sided Hypothesis Test*) The hypothesis is then formulated as

$$H_0 : \tau \leq 0 \text{ versus } H_1 : \tau > 0, \tag{5.4}$$

where τ denotes the range of possible values.

Definition 5.3 (*Test Statistic*) The test statistic $d(Y)$ reflects the distance from H_0 in the direction of H_1. Assuming the data follow a normal distribution, i.e., $Y \sim \mathcal{N}(\mu_0, \sigma^2)$, the test statistic reads

$$d(Y) = \sqrt{n}(\bar{Y} - \mu_0)/\sigma. \tag{5.5}$$

In the remainder of this chapter, we assume that data are NIID.

Definition 5.4 (*Cut-off Point: $c_{1-\alpha}$*) The $c_{1-\alpha}$ is a threshold value or the cut-off point.

Definition 5.5 (*Upper-tail of the Standard Normal Distribution: $u_{1-\alpha}$*) $u_{1-\alpha}$ denotes the value of the normal distribution which cuts off the upper-tail probability of α.

Based on the test statistic from Eq. 5.5, we can calculate the cut-off point $c_{1-\alpha}$: $d(Y) = \sqrt{n}(\bar{Y} - \mu_0)/\sigma = u_{1-\alpha} \Leftrightarrow \bar{Y} = \mu_0 + (u_{1-\alpha})\sigma/\sqrt{n} = c_{1-\alpha}$. When a test statistic is observed beyond the cut-off point, $d(Y) > c_{1-\alpha}$, we reject the H_0 at a significance level α. Otherwise the H_0 is not rejected.

This hypothesis test can lead to two kinds of errors based on the decision taken.

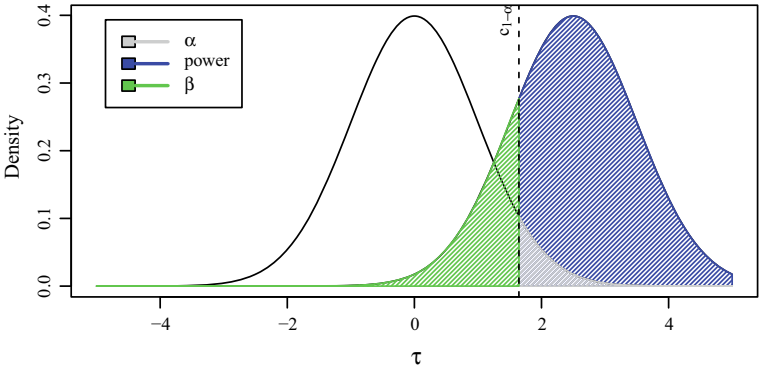

Fig. 5.1 Hypothesis test

Definition 5.6 (*Type I and II Errors*) They are the Type I and the Type II errors, which are pre-specified before the experiment is carried out.

1. A Type I error occurs while incorrectly rejecting the null hypothesis when it is true. The probability of committing a Type I error is called the significance level and is denoted as α. In other words, α is the acceptable probability for Type I error to occur, which is decided by the user. The Type I error can be represented as shown in Fig. 5.1. $\alpha = P_{H_0}(d(Y) > c_{1-\alpha})$.
2. A Type II error occurs while incorrectly rejecting the alternative hypothesis, when it is true: $\beta = P_{H_1}(d(Y) \leq c_{1-\alpha})$.

The notation $P_H(y)$ represents the probabilistic assignments under a model, i.e., the probability of y under the hypothesis H. The power $(1 - \beta)$ is the probability of correctly rejecting the null hypothesis when it is false.

Definition 5.7 (*Paired Samples*) Two samples $\mathcal{X}_1$ and $\mathcal{X}_2$ are considered *paired*, if there is a relation that assigns each element in $\mathcal{X}_1$ uniquely to one element in $\mathcal{X}_2$.

Example: Paired Samples

Therefore, we consider results from running deterministic optimization methods $\mathcal{A}$ and $\mathcal{B}$ paired, if they are using the same starting points (the starting points can be used for indexing the sample points). The starting points are randomly generated, using the same seed for each sample.

Example: Conjugate Gradient versus Nelder-Mead

We will consider the performance differences between two optimization methods. To enable replicability, we have chosen two optimization methods (optimizers) that are available "out of the box" in every R installation via the `optim` function. They are described in the R help system as follows (R Core Team 2022):

1. Method Conjugate Gradient (CG) is a conjugated gradients method based on Fletcher and Reeves (1964). Conjugate gradient methods will generally be more fragile than the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) method, but as they do not store a matrix they may be successful in much larger optimization problems.
2. Method Nelder and Mead Simplex Algorithm (NM) uses only function values and is robust but relatively slow (Nelder and Mead 1965). It will work reasonably well for non-differentiable functions.

CG and NM will be tested on the two-dimensional Rosenbrock function (Rosenbrock 1960). The function is defined by

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2. \tag{5.6}$$

It has a global minimum at $(x_1, x_2) = (1, 1)$. To keep the discussion focused, assume that results from $n = 100$ runs of each method are available, i.e., in total, 200 runs were performed. Let $y_{i,j}$ denote the result of the jth repetition of the ith method, i.e., the vector $y_{1,\cdot}$ represents 100 results of the CG runs.

We will consider the performance differences $d_j = y_{1,j} - y_{2,j}, \quad j = 1, \ldots, n$, with corresponding mean $\bar{d} = 9.02$. Based on

$$S_d = \left(\frac{\sum_{j=1}^{n} (d_j - \bar{d})^2}{n - 1} \right)^{1/2} \tag{5.7}$$

we can calculate the sample standard deviation of the differences as $S_d = 30.73$.

As $\bar{d}$ is positive, we can assume that method NM is superior. We are interested to see whether the difference in means is smaller or larger than μ_0 and formulate the test problem as

$$H_0 : \mu \leq \mu_0 \text{ versus } H_1 : \mu > \mu_0,$$

in our case: $\mu_0 = 0$. And, if H_0 is rejected then it signifies that NM outperforms CG for the given test function.

We will use the test statistic as defined in (5.5) which follows a standard normal distribution if H_0 is true ($\mu \leq \mu_0$). Then

$$P\left(\frac{\overline{Y} - \mu_0}{\sigma/\sqrt{n}} > u_{1-\alpha} \right) \leq \alpha, \text{ otherwise } P\left(\frac{\overline{Y} - \mu_0}{\sigma/\sqrt{n}} > u_{1-\alpha} \right) > \alpha, \tag{5.8}$$

where $u_{1-\alpha}$ denotes the cut-off point; see Definition 5.5.

The test $T(\alpha)$ results in rejecting the null hypothesis H_0 if $d(y) > u_{1-\alpha}$ and in not rejecting H_0 otherwise. For $\alpha = 0.025$ and $u_{1-\alpha} = 1.96$, we get $d(y) = (\overline{y} - \mu_0)/(\sigma/\sqrt{n}) = 2.93 > 1.96 = u_{1-\alpha}$, i.e., H_0 will be rejected.

A sample size of $n = 100$ was chosen without any statistical justification: it remains unclear whether ten samples might be sufficient or whether one thousand samples should have been used. The power calculation, which will be discussed next, provides a proven statistical tool to determine adequate sample sizes for planned experimentation.

5.6.2 Power

The power function that is used to calculate the power for several alternatives μ_1 is defined as

Definition 5.8 (*Power Function*)

$$\text{Pow}(\mu_1) = P_{\mu=\mu_1}\left(\frac{\overline{Y} - \mu_0}{\sigma/\sqrt{n}} > u_{1-\alpha}\right) = 1 - \Phi\left(u_{1-\alpha} - \frac{\mu_1 - \mu_0}{\sigma/\sqrt{n}}\right) \tag{5.9}$$

where $\mu_1 = \mu_0 + \Delta$ and Δ denotes the relevant difference.

In our example, we set up a one-sided test with $H_0 : \mu_0 = 0$ and the following parameters:

1. significance level: $\alpha = 0.025$
2. beta (1-power): $\beta = 1 - 0.8 = 0.2$
3. relevant difference: $\Delta = 10$
4. between-sample standard deviation: $\sigma = 30.73$.

The relationship between power and sample size is illustrated in Fig. 5.2.

5.6.3 p-Value

The p-value quantifies how strongly the data contradicts the null hypothesis, and it allows others to make judgments based on the significance level of their choice (Mayo 2018; Senn 2021).

Definition 5.9 (p *-value*) A p-value is the probability of observing an outcome as extreme or more extreme than the observed outcome $\bar{y}$ if the null hypothesis is true. It is defined as the α' value with

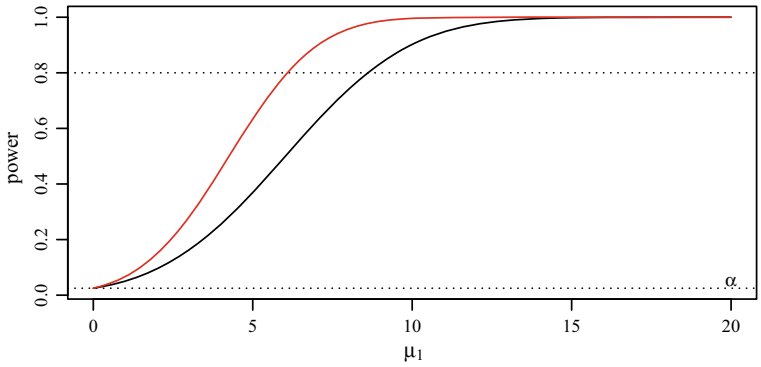

Fig. 5.2 Power for n = 100 (black) and n = 200 (red) for varying μ_1 values. This figure illustrates that larger sample sizes result in higher power

$$d(Y) > u_{1-\alpha'} \Leftrightarrow \alpha' = 1 - \Phi\left(\sqrt{n}(\bar{y} - \mu_0)/\sigma\right),$$

under the assumption that H_0 is true.

If an effect τ measures the true difference between the performance of two methods and y is a statistic used to measure the difference between methods, a one-sided p-value can be defined as

$$P_{\tau=0}(y \geq \bar{y}) \tag{5.10}$$

where $\bar{y}$ is the observed value of the statistic if H_0 is true. The p value can be used for going beyond the simple decision *reject* or *not reject*.

Senn (2002) claims that p-values are a perfectly reasonable way for scientists to communicate the results of a significance test, even when making decisions rather than conclusions. Small p-values indicate that either H_0 is not true or a very unlikely event has occurred (Fisher 1925).

Example: CG versus NM continued

Considering the CG versus NM example (Sect. 5.6.1), the observed difference $\bar{d} = 9.02$, and the corresponding p-value of 0.0017 is obtained.

5.6.4 Effect Size

The effect size is an easy scale-free approach to quantifying the size of the performance difference between the two methods.

Definition 5.10 (*Effect size*) The *effect size* is the standardized mean difference between the two methods, say $\mathcal{A}$ and $\mathcal{B}$ (Cohen 1977):

$$\text{Cohen's } d = \frac{\bar{y}_{\mathcal{A}} - \bar{y}_{\mathcal{B}}}{S_p} \tag{5.11}$$

$$S_p = \sqrt{\frac{(n_{\mathcal{B}} - 1)s_{\mathcal{B}}^2 + (n_{\mathcal{A}} - 1)s_{\mathcal{A}}^2}{n_{\mathcal{A}} + n_{\mathcal{B}} - 2}}, \tag{5.12}$$

where $\bar{y}_{\mathcal{A}}$ and $\bar{y}_{\mathcal{B}}$ is the sample mean of the method $\mathcal{A}$ and $\mathcal{B}$, respectively. The S_p is the pooled standard deviation, $n_{\mathcal{A}}, n_{\mathcal{B}}$ are the sample size of each method, and $s_{\mathcal{A}}, s_{\mathcal{B}}$ are the standard deviation of each method. As a guideline, Cohen suggested effect size as small (0.2), medium (0.5), and large (0.8) but with a strong caution to the applicability in different fields.

Hedges and Olkin (1985) identified that Cohen's d is biased and it slightly overestimates the standard deviation and introduced a correction measure as

$$\text{Hedge's } g = 1 - \frac{3}{4(n_{\mathcal{A}} + n_{\mathcal{B}}) - 9} \times \text{Cohen's } d. \tag{5.13}$$

Example: CG versus NM continued

Again, considering the CG versus NM example (Sect. 5.6.1), Cohen's d and Hedge's g, which are the standardized mean difference between the two methods, can be calculated using (5.11) and (5.13) as $d = 0.415$ and $g = 0.4134$, respectively. Both values indicate that the observed mean difference is of a smaller magnitude.

5.6.5 *Sample Size Determination and Power Calculations*

Adequate sample size is essential for comparing algorithms. Even for deterministic optimizers, it is recommended to perform several runs with varying starting points instead of using results from one run of each algorithm. But "the more the merrier" is not efficient in this context, because additional runs incur additional costs. Statistical inference provides tools for tackling this trade-off between cost and effectiveness.

5.6.5.1 Five Basic Factors

The usual point of view is that the sample size is the determined function of variability, statistical method, power, and difference sought. We consider a one-sided test as defined in Eq. 5.4.

Definition 5.11 (*Five basic factors*) While discussing sample size requirements, Senn (2021) introduced the following conventions regarding symbols:

α: the probability of a type I error, given that the null hypothesis is true.

β: the probability of a type II error, given that the alternative hypothesis is true.
Δ: the difference sought. In most cases, one speaks of the "relevant difference" and this in turn is defined "as the difference one would not like to miss". Notation: In hypothesis testing, Δ denotes a particular value within the range of possible values τ.
σ: the presumed standard deviation of the outcome.
n: the number of runs of each method. Because two methods are compared; the total number is $2 \times n$.

5.6.5.2 Sample Size

Based on the definition of the type II error rate for $1 - \beta$ for μ_1, the sample size can be calculated for the type II error rate, i.e.,

$$\Phi\left(u_{1-\alpha} - \frac{\mu_1 - \mu_0}{\sigma/\sqrt{n}}\right) = \beta \Leftrightarrow n = \frac{\sigma^2}{(\mu_1 - \mu_0)^2}(u_{1-\alpha} - u_\beta)^2 = \frac{\sigma^2}{\Delta^2}(u_{1-\alpha} - u_\beta)^2,$$

which gives an estimate of the required sample size $n = n(\alpha, \beta, \sigma, \mu_0, \mu_1) = n(\alpha, \beta, \sigma, \Delta)$.

Any four factors from Definition 5.11 are enough to determine the fifth factor uniquely. First, we consider the formula for sample size, n as a function of α, β, Δ, and σ. For a one-sided test of size α, the (approximate) formula for sample size is

$$n \approx 2 \times (u_{1-\alpha} + u_{1-\beta})^2 \sigma^2 / \Delta^2, \tag{5.14}$$

where $u_{1-\alpha}$ denotes the value of the normal distribution which cuts off the upper-tail probability of α.

Hence, for the CG versus NM example (Sect. 5.6.1), if the relevant difference is $\Delta = 10$ then approximately 148 completing runs per method are required.

Example: Sample size determination

Compare two optimization methods, say $\mathcal{A}$ = CG and $\mathcal{B}$ = NM. Therefore, we set up a one-sided test with the following parameters:

1. significance level: $\alpha = 0.05$
2. beta (1-power): $\beta = 1 - 0.8 = 0.2$
3. relevant difference: $\Delta = 200$
4. between-sample standard deviation: $\sigma = 450$.

We will use the function `getSampleSize` from the R package `SPOT` to determine the sample size n. All calculations shown in this chapter are implemented in this package.

```
library("SPOT")
nsamples <- round(getSampleSize(
  mu0 = 0, mu1 = 200,
  alpha = 0.05, beta = 0.2,
  sigma = 450,
  alternative = "one.sided"
), 0)
```

Based on Eq. 5.14, approximately $n = 63$ completing runs per method are required.

Although sample size calculation appears to be transparent and simple, there are several issues with this approach that will be discussed in the following.

5.6.6 Issues

In this section, we will consider issues with sample size determination, with power calculations, and with hypotheses and wrong conclusions from hypothesis testing. Our presentation (and especially the examples) is based on the discussion in Senn (2021).

Issues with sample size determination can be caused by the computation of the standard deviation, σ: This computation is a chicken or egg dilemma, because the between-sample standard deviation will be unknown until the result of the experiment is known. But the experiment must be planned before it can be run. Furthermore, Eq. 5.14 is only an approximate formula, because it is based on the assumption that the standard deviation is known. The experiments we use are based on using an estimate obtained from the examined sample.

There is no universal standard for a relevant difference Δ. This creates another problem in determining sample size, since significant differences are application-dependent.

Errors can cause issues with sample size determination, because the levels of α and β are relative: α is an actual value used to determine significance in analysis, while β is a theoretical value used for planning (Senn 2021). Frequently, the error values are chosen as $\alpha = 0.05$ and $\beta = 0.20$. However, in some cases, the value of β ought to be much lower, but if only a very small number of experiments are feasible, a very low value of β might not be realistic. The same considerations are true for α, because α and β cannot be reduced simultaneously without increasing the sample size.

In practice, sample size calculation might be flawed. For example, $n = 10$ or $n = 100$ are popular sample sizes, but they are often chosen without any justification. Some authors justify their selection by claiming that “this is done by everyone”.

In some situations, there is enough knowledge to plan an experiment, i.e., the number of experiments to be performed is known. Nuclear weapons tests are an extreme example of this situation.

Furthermore, Senn (2021) claims that the sample size calculation can be "an excuse for a sample size and not a reason". In practice, there is a usually undesirable tendency to "adjust" certain factors, notably the difference sought and sometimes the power, in light of *practical sample size requirements*.

Tip: Sample Size Determination

Perform pre-experimental runs to compute the (approximate) sample size before the full experiment is started.

In addition to issues with sample size determination, also issues with power calculations might arise. The fact that a sample size has been chosen which seemingly has 80% power does not guarantee that there is an 80% chance that there is an effect (alternative H_1 is true) (Senn 2021). Even if the whole experimental setup and process are correct, external failures can happen and that is outside of the experimenter's control: The methods or the algorithm may not work. Importantly, if an algorithm does not work we must recognize this; see the example in Sect. 5.8.2.1. But even if the algorithm is successful, it may not produce a relevant difference. Or, looking at another extreme, the algorithm might be better than planned for—so the sample size could have been chosen smaller. In addition, experimental errors might occur that are not covered by the assumptions made for the power (sample size) calculation. The calculations are made under the assumption that the experiment is performed correctly. Or, as Senn (2021) states: Sample size calculation does not allow for "acts of God" or dishonest or incompetent investigators. Thus, although we can affect the probability of success by adjusting the sample size, we cannot fix it.

Finally, there are issues with hypotheses and wrong conclusions based on hypothesis testing. Selecting the correct hypothesis pair, e.g., $H_0 : \tau \leq 0$ versus $H_1 : \tau > 0$ (one-sided) or $H_0 : \tau = 0$ versus $H_1 : \tau \neq 0$ (two-sided) is not always obvious.

In the context of clinical testing, Senn (2021) states that the following statement is a *surprisingly widespread piece of nonsense*:

> If we have performed a power calculation, then upon rejecting the null hypothesis, not only may we conclude that the treatment is effective but also that it has a clinically relevant effect.

Consider, for example, the comparison of an optimization method, $\mathcal{A}$ with a second one, say method $\mathcal{B}$, based on a two-sided test. Let τ be the true difference in performance ($\mathcal{A}$ versus $\mathcal{B}$). We then write the two hypotheses,

$$H_0 : \tau = 0 \text{ versus } H_1 : \tau \neq 0. \tag{5.15}$$

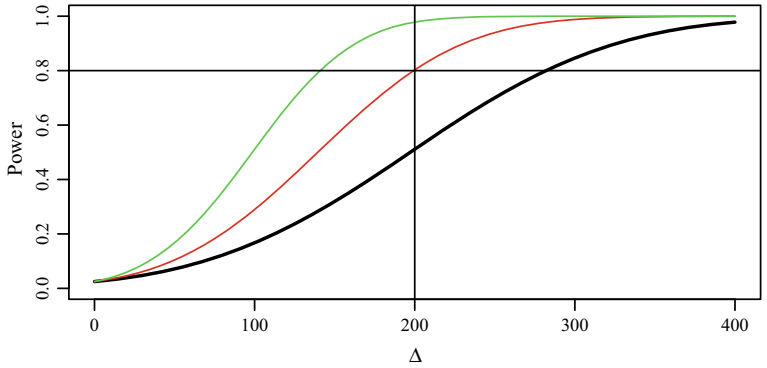

Fig. 5.3 Power as a function of the relevant difference Δ for a two-parallel-group experiment (black = 40, red = 80, and green = 160 runs). If the relevant difference Δ is 200, n = 80 runs per method are needed for 80% power

By rejecting the null hypothesis, we are in favor of the alternative, H_1, which states that there is a non-zero difference. The sign of this difference might indicate whether $\mathcal{A}$ is superior or inferior to $\mathcal{B}$.

Replacing Eq. 5.15 with

$$H_0 : \tau = 0 \text{ versus } H_1 : \tau \geq \Delta \tag{5.16}$$

would imply that we know one algorithm is better than the other *before* the experiments are performed. But this is usually not known prior to the experiment—the whole point of the experiment is to determine which algorithm performs better. Therefore, we will consider a one-sided test as specified in Eq. 5.4. This procedure will be exemplified in Sect. 5.8.

We have highlighted some important issues with sample size determination, power calculations, and hypotheses tests. Senn (2021) mentions many more, and the reader is referred to his discussion.

Tips

Plotting the power function for an experiment is recommended. This is illustrated in Fig. 5.3.

Last but not the least, issues with the "large n problem", i.e., the topic "large versus small samples", should be considered. Senn (2021), Sect. 13.2.8 states:

1. other things being equal, significant results are more indicative of efficacy if obtained from large experiments rather than small experiments.

2. But consider: if the sample size is increased, not only is the power of finding a relevant difference, Δ, increased, but the smallest detectable difference also decreases.

5.7 Severity

5.7.1 Motivation

Severity has been proposed as an approach to tackle the issues discussed in Sect. 5.6.6 by philosopher of science Mayo (1996). To explain the concept of severity, we start with an example that was taken from (Senn 2021).

Example: High Power

Consider an algorithm comparison using a one-sided test with $\alpha = 0.025$ but with a very high power, say 99%, for a target relevant difference of $\Delta = 200$. The standard deviation of the differences in the mean is taken to be 450. Note, except for drastically reducing the error of the second kind from $\beta = 0.2$ down to $\beta = 0.01$, this example is similar to the Example "Sample Size Determination" in Sect. 5.6.5. A one-sided hypothesis test as specified in Eq. 5.4 with the following parameters is performed:

1. significance level: $\alpha = 0.025$
2. power: $1 - \beta = 0.99$
3. relevant difference: $\Delta = 200$
4. between-sample standard deviation: $\sigma = 450$.

A standard power calculation, see Eq. 5.14, suggests $n \approx 186$ samples for each configuration, which we round up to $2 \times 200 = 400$ in total. This value gives a standard error for the difference of $450 \times \sqrt{2/200} = 45$.

We run the experiments (assuming unpaired, i.e., independent samples) and the result is significant, i.e., we have observed a difference of $\bar{y} = 90$. We get the p-value 0.0231.

How can we interpret the results from this experiment, e.g., the p-value? Although the p-value of 0.0231 is statistically significant, i.e., p-value $< \alpha$, we cannot conclude that the H_1 is true. The probability of occurrence of a type I error has to be acknowledged. The situation is shown in Fig. 5.4. Observing a $\bar{y} = 90$ is more likely under H_0 than under H_1. This is evident by comparing the height of the density curve at $\bar{y} = 90$ both under the H_0 and H_1, respectively. Hence, this is more likely to be the case of a type I error. Although the power is relatively high ($1 - \beta = 0.99$), it

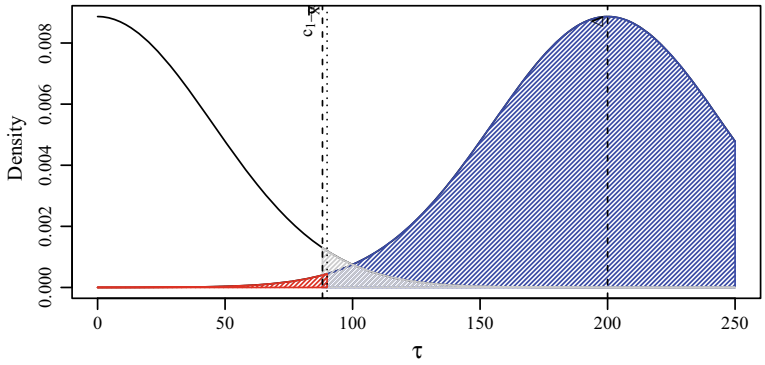

Fig. 5.4 Severity (red), type I error rate (gray), and power (blue). Since $\bar{y}$ is larger than $c_{1-\alpha}$, the null hypothesis is rejected

would be an error to claim that the experiment has an effect $\Delta = \mu_1 - \mu_0 = 200$.[4] There are two reasons:

1. This test did not make extensive use of $\bar{y}$, the actual difference observed. The actual difference observed is only used to calculate the test statistic and to decide whether the null hypothesis should be rejected.
2. Going beyond the simple decision *reject* or *not reject*, the p value can be used. The actual difference observed, $\bar{y} = 90$, is closer to 0 than to 200. Because 90 is farther away from 200 than from 0, this is far from good evidence that the true difference is as large as the relevant difference of 200.

Senn (2021) proposed ways for solving this problem, e.g., using a so called *point-estimate* of the true difference together with associated confidence limits, or using an irrelevant difference approach, or using severity.

5.7.2 Severity: Definition

Severity is a measure of the plausibility of a result which considers the decision and the data: after the decision is made, the severity of rejecting or not rejecting the null hypothesis can be calculated. It uses post-data information and provides means for answering the question:

> How can we ensure that the results are not only statistically but also scientifically relevant?

The concept of *Severity* was introduced by Mayo and Spanos (2006) (see also Mayo 2018):

[4] Note: $\mu_0 + \Delta = \mu_1$.

Table 5.1 Power $(1-\beta)$, significance level (α), p-value, and severity. P_{H_0} denotes the probability under the assumption that H_0 is true, whereas P_{H_1} denotes the probability under the assumption that H_1 is true

$1-\beta$	α	p-value	Severity
$P_{H_1}(Y > c_{1-\alpha})$	$P_{H_0}(Y > c_{1-\alpha})$	$P_{H_0}(Y > \bar{y})$	S_{nr}: $P_{H_1}(Y > \bar{y})$ S_r: $P_{H_1}(Y \leq \bar{y})$
$P_{H_1}(d(Y) > u_{1-\alpha})$	$P_{H_0}(d(Y) > u_{1-\alpha})$	$P_{H_0}(d(Y) > d(\bar{y}))$	S_{nr}: $P_{H_1}(d(Y) > d(\bar{y}))$ S_r: $P_{H_1}(d(Y) \leq d(\bar{y}))$

> The result that hypothesis H is not rejected is severe only if it is very unlikely that this result will also occur if H is false.

Severity offers a meta-statistical principle for evaluating the proposed statistical conclusions. It shows how well-tested (not how likely) hypotheses are. It is therefore an attribute of the entire test procedure. The severity of the test and the resulting outcome can be evaluated.

Definition 5.12 (*Severity*) Severity is defined separately for the non-rejection (S_{nr}) and the rejection (S_r) of the null hypothesis as in (5.17).

$$\begin{aligned} S_{nr} &= P_{H_1}(d(Y) > d(y)) \\ S_r &= P_{H_1}(d(Y) \leq d(y)). \end{aligned} \tag{5.17}$$

The S_{nr} values increase monotonically from 0 to 1 as a function of τ. The S_r values decrease monotonically from 1 to 0 as a function of τ. The closer the value is to 1, the more reliable is the decision made with the hypothesis test. The key difference between power and severity is that severity depends on the data and the test statistic, i.e., $d(y)$ instead of $c_{1-\alpha}$.

The severity is an analogous probability to Eq. 5.10 that considers non-zero τ values. The severity of rejection, which considers values in the other direction, $y \leq \bar{y}$ is calculated as

$$S_r(\tau') = P_{\tau=\tau'}(y \leq \bar{y}), \tag{5.18}$$

if H_0 is rejected and $S_{nr}(\tau') = P_{\tau=\tau'}(y \geq \bar{y})$, otherwise. Table 5.1 shows the relations between power $(1-\beta)$, significance level (α), p-value, and severity.

Example: High Power (Continued)

Figure 5.5 plots the severity for the given example against every possible value of the true difference in the performance τ (Senn 2021).

Labeled on the graph are values of $\tau = \bar{y}$, the observed difference, for which the severity is 0.5, and $\tau = \Delta$, the value used for planning. The severity of rejecting H_0 is only 0.0075 for this value. Figure 5.5 exhibits that $\tau > 200$ has a very low severity.

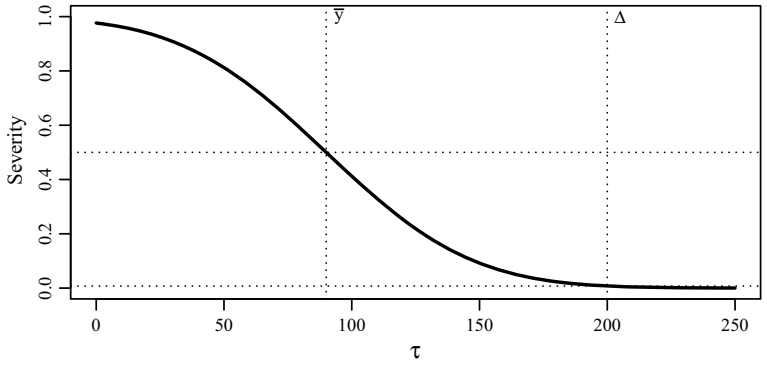

Fig. 5.5 Severity of rejecting H_0, S_R as a function of $\Delta = \mu_1 - \mu_0$. $S_R(0) = 1 - p$

The p-value, here 0.0231, is smaller than $\alpha = 0.025$. Note, in case of rejection H_0, severity is $1 - p$ for $\Delta = \mu_1 - \mu_0 = 0$. The severity of not rejecting the null hypothesis is the same as the p-value for $\Delta = 0$.

Example: Conjugate Gradient versus Nelder-Mead (Continued)

Let us now revisit the CG versus NM example (Sect. 5.6.1) and calculate the severity. Given the observed difference 9.02 and sample size n=100, the decision based on the p-value of 0.0017 is to reject H_0. Considering a target relevant difference of $\Delta = 10$, the severity of rejecting H_0 is 0.37 and is shown in the left panel in Fig. 5.6. The right panel in Fig. 5.6 shows the severity of rejecting H_0 as a function of τ. Based on the result of the hypothesis test for the given data, NM seems to outperform CG. And, claiming that the true difference is as large as or larger than 10 has a very low severity, whereas differences smaller than 7 are well supported by severity.

5.7.3 Two Examples

We will use two illustrative examples for severity calculations that are based on the discussions in Mayo (2018), Bönisch and Inderst (2020), and Senn (2021). In each example, 100 samples from a $\mathcal{N}(\mu, \sigma^2)$-distributed random variables are drawn, but with different means. The first example represents a situation in which the true difference is small compared to the variance in the data, whereas the second example represents a situation in which the difference is relatively large. The first example uses the sample mean $\mu_1 = 1e - 6$ (data set I), the second sample $\mu_2 = 3$ (data set

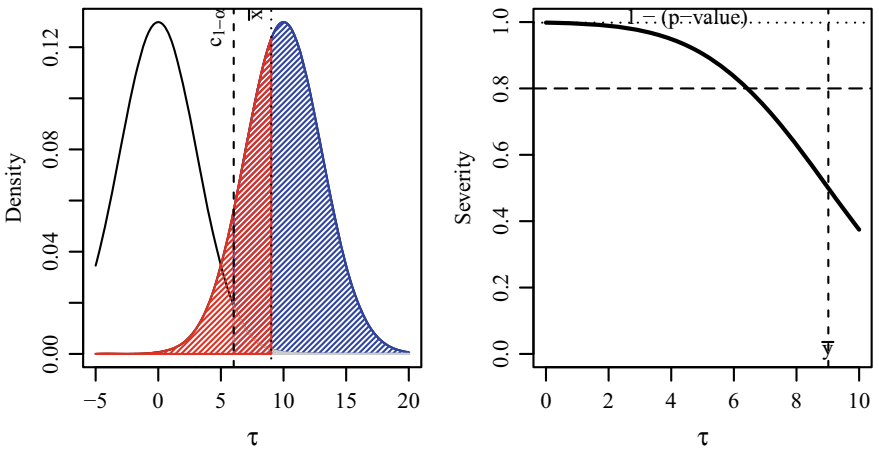

Fig. 5.6 Left: Severity of rejecting H_0 (red), power (blue) for a target relevant difference $\Delta = 10$. Right: Severity of rejecting H_0 as a function of τ

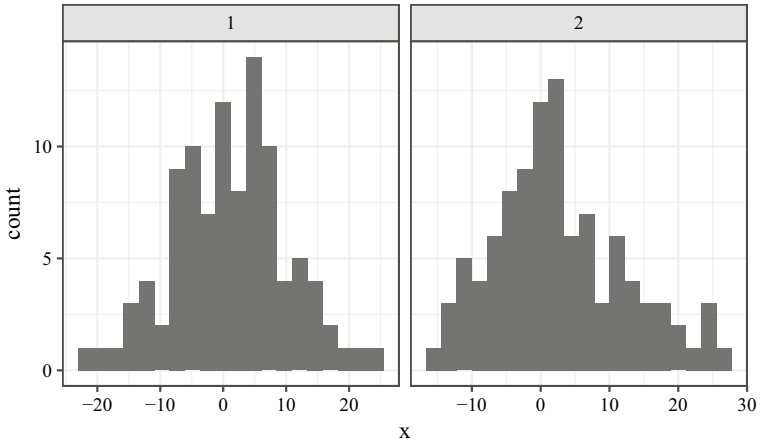

Fig. 5.7 Data sets I and II. Histograms showing artificial data. Left: mean = 1e-6; right: mean = 3. Standard deviation $\sigma = 10$ in both cases

II). The same standard deviation ($\sigma = 10$) is used in both cases. Histograms of the data are shown in Fig. 5.7.

In both examples, a one-sided test is performed as defined in (5.4) with the following parameters:

1. significance level: $\alpha = 0.05$
2. power: $1 - \beta = 0.8$
3. relevant difference: $\Delta = 2.5$
4. between-sample standard deviation: $\sigma = 10$.

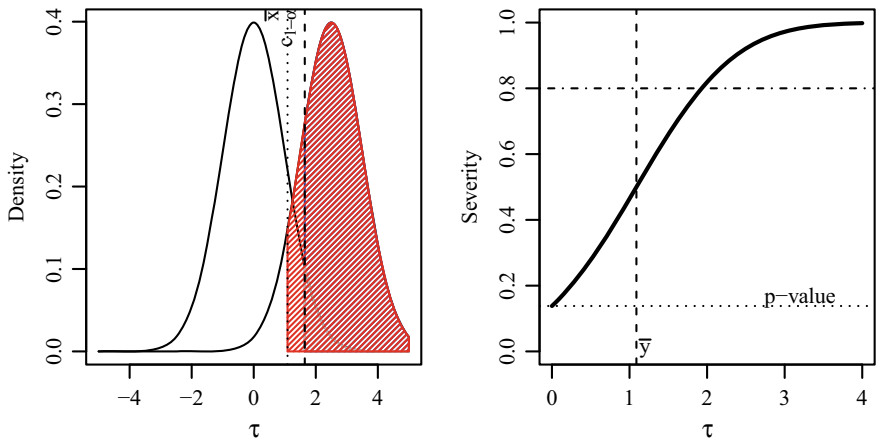

Fig. 5.8 Severity of not rejecting the null for a target relevant difference $\Delta = 2.5$. Right: Severity of not rejecting H_0 as a function of τ

Example: Data set I: Severity of not rejecting the null hypothesis

First, using data set I (100 samples from a $\mathcal{N}(\mu_1, \sigma^2)$ distributed random variable, with $\mu_1 = 1e-6$ and $\sigma = 10$), the severity of not rejecting the null hypothesis is analyzed. Assume, the value $\bar{y} = 1.0889$ was observed, i.e., a statistically not significant difference (p-value $0.1381046 > 0.05$) is observed. But it would be a mistake to conclude from this result that the size of the difference is zero.

Figure 5.8 illustrates this situation by applying the concept of severity. The right panel in Fig. 5.8 provides a graphic depiction of answers to the following question for different values of τ: if the actual difference is at least τ, what is the probability of the observed estimate being higher than the actually observed value of $\bar{y} = 1.089$?

The greater this probability, the stronger the observed evidence is against that particular τ value. For two numbers, the answer is already known: for $\tau = 0$, namely 0.14, which is the p-value, and for $\tau = \bar{y}$, which is 50%. The p-value indicates that the null hypothesis of "no difference" cannot be rejected for $\alpha = 0.05$.

Because of the high variance in the data, the histogram is relatively broad (see the left panel in Fig. 5.7). This is now directly reflected in the assessment of other possible τ values (other initial hypotheses for a difference). For example, the severity of the evidence only crosses the threshold of 80% at a τ-value of approximately 2. This can be seen on the vertical axis in the right panel in Fig. 5.8. Therefore, if the actual difference was at least 2, then there would be a probability of 80% of estimating a value higher than the observed value 1.09. Even if the null hypothesis is not rejected, it cannot be concluded that the magnitude of the difference is zero. With high severity (80%), it can be concluded that the differences larger than 2 are unlikely. The right panel in Fig. 5.8 shows the severity of evidence (vertical axis) for all initial hypotheses (horizontal axis).

Table 5.2 Data set I: result analysis

p-value	Decision	Power	Cohen's d	Hedge's g	Severity
0.14	H0 not rejected	0.8037649	0.1212286	0.1212286	$\Delta \geq 2$ are well supported

The result statistic is presented in Table 5.2. The effect size suggests that the difference is of a smaller magnitude.

Example: Data set II: Severity of rejecting the null hypothesis

Data set II, with $\mu_2 = 3$ and $\sigma = 10$, is used to analyze the severity of rejecting the null hypothesis, i.e., the statistically significant estimate of $\bar{y} = 2.62$, resulting in the null hypothesis (of "there is no difference") being rejected, is considered.

Asserting this is evidence for a difference of exactly $\bar{y} = 2.62$ is not justified. Besides the null hypothesis, no further hypotheses were tested, e.g., "is the difference exactly 2.62?" or "is the difference at least 2.62?". Statistically, it was shown that there is a very low probability that there is no positive difference. So the evidence strongly ("severely") argues against the lack of an effect.

In the following, the test result is used to evaluate further hypotheses, e.g., that the difference is "not higher than at most τ," where τ represents a possible difference of, say, $\bar{y} = 4$ or $\bar{y} = 5$. The central question in this context is: How strongly does the experimental evidence speak against such an alternative null hypothesis, i.e., a difference of at most τ? This situation is comparable to the question of whether the null hypothesis can be rejected with sufficient certainty. This question can only be answered with a probability of error that can be estimated.

The following results were inspired by Bönisch and Inderst (2020), who present a similar discussion in the context of "damage estimation". For $\tau = 0$, the probability that the observed estimate is less than the observed value $\bar{y} = 2.62$ is $1 - p = 99.56$ %. Applying these results to other hypotheses about the value of τ leads to results shown in Fig. 5.9: For example, if $\tau = 1.75$, the probability of observing a value smaller than $\bar{y}$ is 80.84%. For $\tau = 2.5$, the probability would decrease to 54.85%. Consider—similar to the power value of 0.8—an 80% threshold as a minimum requirement for severity, then an estimate of $\bar{y} = 2.62$ that there is a sufficient severity against a difference up to $\tau = 1.75$ is obtained (Table 5.3).

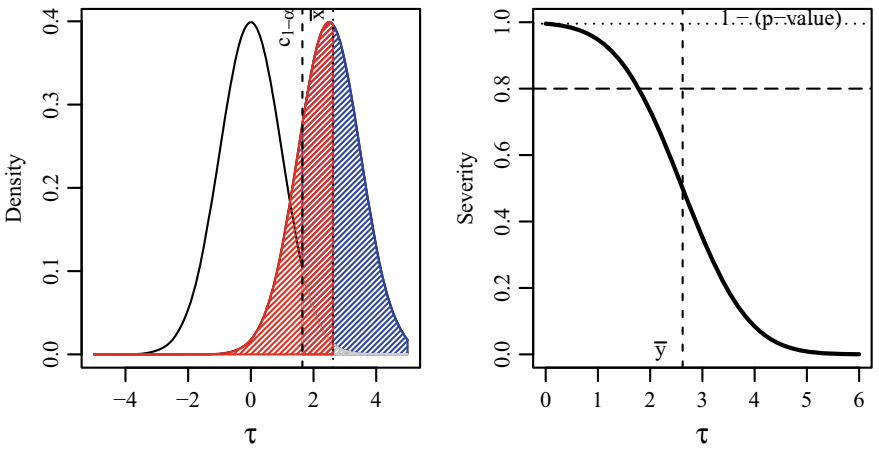

Fig. 5.9 Severity of rejecting H_0 (red), power (blue), and error (gray) for a target relevant difference $\Delta = 2.5$. Right: Severity of rejecting H_0 as a function of τ

Table 5.3 Data set II: result analysis

p-value	Decision	Power	Cohen's d	Hedge's g	Severity
0	H0 rejected	0.8037649	0.2737213	0.2737213	$\Delta \leq 2$ are well supported

5.7.4 Discussion of the 80% Threshold

Although a threshold of 80% was used in Fig. 5.9, it remains unclear at which threshold the level should be set. Demanding a severity of 90% has the consequence that even the assumption of a difference of at least 1.9 is not supported. Given that severity should also take into account domain-specific knowledge, a general value cannot be recommended. Visualizations such as Fig. 5.9 can help to get objective and rational results.

5.7.5 A Comment on the Normality Assumption

We discussed the extended classical hypothesis testing mechanism with Mayo's error statistics. Central tool in error statistics is severity, which allows a post-data analysis.

Severity can be applied to inferential statistics, no matter what the underlying distribution is Spanos (1999). We have discussed the normal distribution, because we will apply severity analysis to benchmarking (Sect. 5.8). In this context, the normality assumption holds, because most of the examples in this chapter use 50 or even more samples. The normality assumption is sometimes misunderstood: it does not require the population to resemble a normal distribution. It requires the sampling distribution of the mean difference should be approximately normal. In most cases, the central limit theorem will impart normality to the (hypothetical) distribution. This happens even to moderate n values, when the underlying population is not extremely asymmetric, e.g., caused by extreme outliers.

5.8 Severity: Application in Benchmarking

Now that we have the statistical tools available, i.e., power analysis plus error statistics (severity), we can evaluate their adequacy for scenarios in algorithm benchmarking.

The following experiments demonstrate how to perform a comparison of two algorithms. Our goal is not to provide a full comparison of many algorithms on many problems, i.e., MAMP, but to highlight important insights gained by severity. Therefore, two algorithms and three optimization problems were chosen. To cover the most important scenarios, three independent MASP studies will be performed. Each study compares two algorithms, say $\mathcal{A}$ and $\mathcal{B}$, on one problem instance.

The function `makeMoreFunList` from the R package `SPOTMisc` generates a list of functions presented in More et al. (1981), which is one of the most cited benchmark suites in optimization with more than 2000 citations. This list can be passed to the `runOptim` function, which performs the optimization. `runOptim` uses the arguments from Table 5.4.

We will compare the optimization methods CG and NM on the Rosenbrock, the Freudenstein and Roth, and Powell's Badly Scaled test function that were defined in More et al. (1981).

Table 5.4 `runOptim` arguments

Parameter	Description	Default value
`fl`	Function list	
`method`	The method used by `optim`: "Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", or "Brent".	"Nelder-Mead"
`n`	Repeats. If $n > 1$, different start points (randomized) will be used	2
`k`	Subset of benchmark functions	All implemented functions
`verbosity`	Level of information to be shown	0

5.8.1 Experiment I: Rosenbrock

5.8.1.1 Pre-experimental Planning

In our first experiment, we will use the Rosenbrock function; see Eq. 5.6. This is the first function in More et al. (1981)s study, so we will pass the argument $k = 1$ to the `runopt()` function. To estimate the number of function evaluations, a few pre-experimental runs of the algorithms are performed. These pre-experimental runs are also necessary for testing numerical instabilities, expected behavior, and correct implementations. In our case, $n = 20$ pre-experimental runs were performed.

```
library("SPOT")
set.seed(1)
k <- 1 # More function no. 1
n0 <- 20 # Pre-experimental runs
moreFl <- makeMoreFunList()
resCG0 <- runOptim(
  fl = moreFl,
  method = "CG",
  n = n0,
  k = k
)
resNM0 <- runOptim(
  fl = moreFl,
  method = "Nelder-Mead",
  n = n0,
  k = k
)
```

A data.frame with 20 observations is available for each algorithm, e.g., for CG:

```
str(resCG0)

  ## 'data.frame':    20 obs. of  3 variables:
  ##  $ f: num  1 1 1 1 1 1 1 1 1 1 ...
  ##  $ r: num  1 2 3 4 5 6 7 8 9 10 ...
  ##  $ y: num  0.10644 0.0686 0.00577 0.08434 3.65499 ...
```

Looking at the summary of the results is strongly recommended. R's `summary` is the first choice.

```
summary(resCG0$y)

  ##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
  ## 0.000939 0.068510 0.080510 0.549287 0.119892 3.654986


summary(resNM0$y)

  ##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
  ## 1.900e-08 5.740e-07 1.439e-06 1.622e-04 3.233e-06 3.206e-03
```

The summaries indicate that NM is superior and that CG has some outliers. A graphical inspection is shown in Fig. 5.10. Taking care of extreme outliers is recommended in further analysis.

We are interested in the mean difference in the methods' performances. The pre-experimental runs indicate that the difference is $\bar{y} = 0.55$. Because this value is positive, we can assume that method NM is superior. The standard deviation is $s_d = 1.14$. Based on Eq. 5.14, and with $\alpha = 0.05$, $\beta = 0.2$, and $\Delta = 0.5$, we can determine the number of runs for the full experiment with the `getSampleSize()` function.

For a relevant difference of 0.5, approximately 65 completing runs per algorithm are required. Figure 5.11 illustrates the situation for various Δ and three n values.

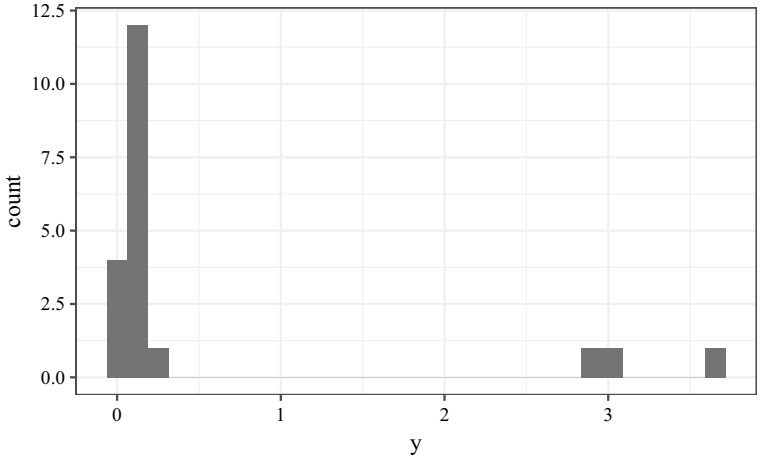

Fig. 5.10 Results from CG on Rosenbrock. Histogram to inspect outliers

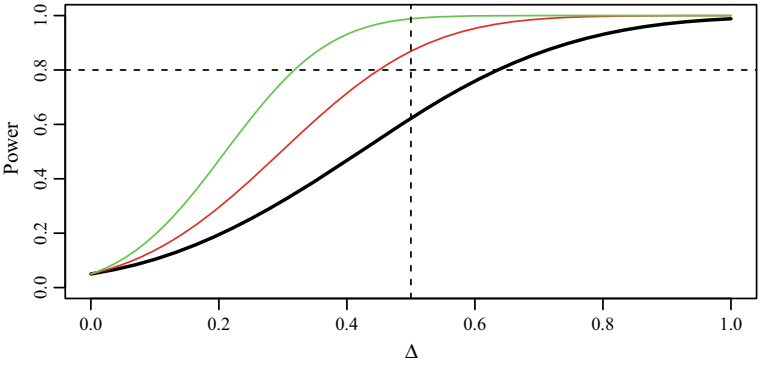

Fig. 5.11 Rosenbrock (function 1). Power as a function of the relevant difference Δ for a two-parallel-group experiment (black = 40, red = 80, and green = 160 runs). If the relevant difference is 0.5, n = 160 runs per algorithm are needed for 80% power

Although we do not know any "true" relevant difference for artificial (dimension less) test functions, we consider the distance $\Delta = 0.5$ as relevant and, to play safe, choose $n = 80$ algorithm runs for the full experiment.

5.8.1.2 Performing the Experiments on Rosenbrock

The full experiments can be conducted as follows. The 20 results from the pre-experimental runs will be "recycled", only 60 additional runs must be performed. How to combine existing results with new ones was discussed in Sect. 4.5.3. The corresponding code is similar to the code that was used for the pre-experimental experiments in Sect. 5.8.1.1.

```
summary(resCG$y)

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000053 0.037231 0.079467 0.681190 0.107427 4.332730

summary(resNM$y)

##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## 4.000e-09 1.430e-07 8.260e-07 8.207e-05 2.686e-06 3.206e-03
```

Figure 5.12 shows a histogram of the results.
The numerical summary of these results is

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000052 0.037222 0.079464 0.681108 0.107421 4.332730
```

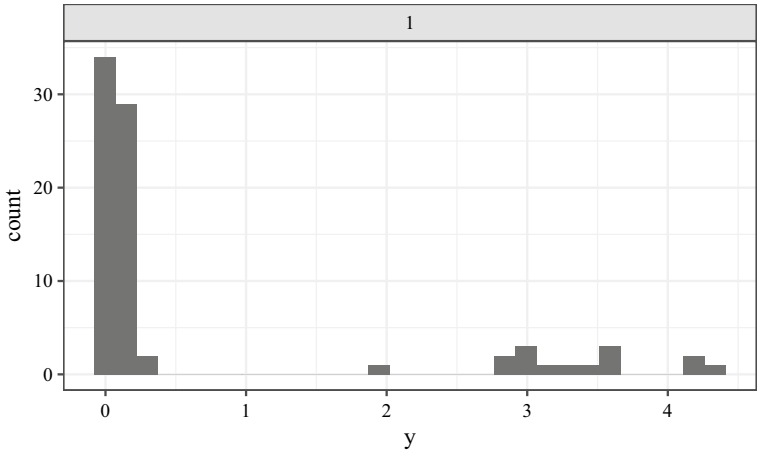

Fig. 5.12 Rosenbrock: Difference between CG and NM results (y = CG - NM)

Table 5.5 Experiment I: result analysis

p-value	Decision	Power	Cohen's d	Hedge's g	Severity
0	H0 rejected	0.961397	0.7348167	0.7313231	$\Delta \leq 0.5$ are well supported

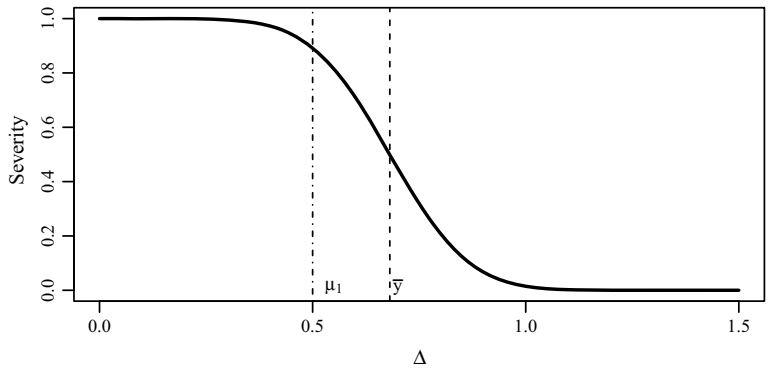

Fig. 5.13 Rosenbrock. Severity for rejecting H_0. Given data from the experiment, claiming that the true difference is as large or larger than 1.0 has a very low severity, whereas differences as large as 0.5 are well supported by severity

The sample mean of the differences is $\bar{y} = 0.68$. Obviously, NM is superior and there is a difference in performance. The question remains: how large is this difference? To answer this question, we will analyze results from these runs with severity.

The summary result statistic is presented in Table 5.5. The effect size suggests that the difference is of medium magnitude. The corresponding severity plot is shown in Fig. 5.13.

5.8.1.3 Discussion

Results indicate that the NM method is superior. Beyond the classical analysis based on EDA tools and hypothesis tests, severity allows further conclusions: It shows that performance differences smaller than 0.5 are well supported. Although the situation is clear, the final choice is up to the experimenter. They might include additional criteria such as run time, costs, and robustness in their final decision. And last but not the least: The question of whether a difference of 0.5 is of practical relevance is highly dependent on external factors.

5.8.2 *Experiment II: Freudenstein-Roth*

The two-dimensional Freudenstein and Roth Test Function (Freudenstein and Roth 1963), which is number $k = 2$ in More et al. (1981)s list, will be considered next. The function is defined as

$$f(x_1, x_2) = (x_1 - 13 + ((5 - x_2)x_2 - 2)x_2)^2 + (x_1 - 29 + ((1 + x_2)x_2 - 14)x_2)^2.$$

5.8.2.1 Pre-experimental Planning: Freudenstein and Roth

Similar to the study of the Rosenbrock function, 20 pre-experimental runs are performed. We take a look at the individual results.

```
summary(resCG0$y)

  ##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  ##  0.3928 19.6811 77.9517 57.7497 81.2694 86.3797

summary(resNM0$y)

  ##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  ##   48.98   48.98   48.98   48.98   48.98   48.98
```

The summaries indicate that NM is not able to find improved values. A floor effect occurred (Bartz-Beielstein 2006). The experiment is too difficult for NM. No further experiments will be performed, because NM is not able to find improvements. A re-parametrization of the NM (via hyperparameter tuning) is recommended, before additional experiments are performed.

Although CG appears to be superior and can find values as small as 0.3928, it has problems with outliers as can be seen in Fig. 5.14.

5.8.2.2 Discussion

An additional, experimental performance analysis (that focuses on the mean) is not recommended in this case, because the result is clear: CG outperforms NM.

5.8.3 *Experiment III: Powell's Badly Scaled Test Function*

Powell's two-dimensional Badly Scaled Test function, which is number $k = 3$ in More et al. (1981)s list, will be considered next.

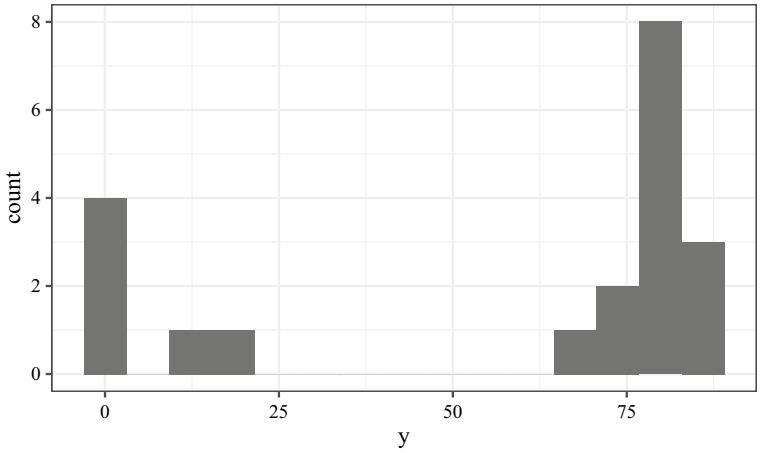

Fig. 5.14 Results from method CG on Freudenstein Roth. Histogram to inspect outliers

The function is defined as

$$f(x_1, x_2) = f_1^2 + f_2^2$$

with

$$f_1 = 1e4 x_1 x_2 - 1 \quad \text{and} \quad f_2 = \exp(-x_1) + \exp(-x_2) - 1.0001.$$

5.8.3.1 Pre-experimental Planning: Powell's Badly Scaled Test Function

First, we take a look at the individual results. The summaries do not clearly indicate which algorithm is superior. A graphical inspection is shown in Fig. 5.15. Both methods are able to find improvements, but both are affected by outliers. The pre-experimental runs indicate that the difference is $\bar{y} = -0.21$. Because this value is negative, we will continue the analysis under the assumption (hypothesis) that method CG is superior.

We are interested in the mean difference in the algorithms' performances.

The standard deviation is $s_d = 1.5$. Based on Eq. 5.14, and with $\alpha = 0.05$, $\beta = 0.2$, and $\Delta = 0.5$, we can determine the number of runs for the full experiment.

For a relevant difference of 0.5, approximately 112 completing runs per algorithm are required. Figure 5.16 illustrates the situation for various Δ and three n values.

Although we do not know any "true" relevant difference for artificial (dimension less) test functions, we consider a distance $\Delta = 0.5$ as relevant and choose $n = 120$ algorithm runs for the full experiment.

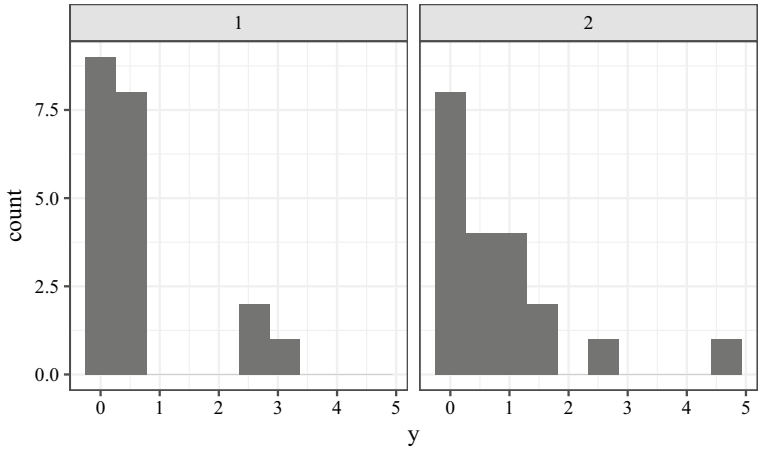

Fig. 5.15 Results from CG and NM on Powell's badly scaled test function. Histograms to inspect outliers

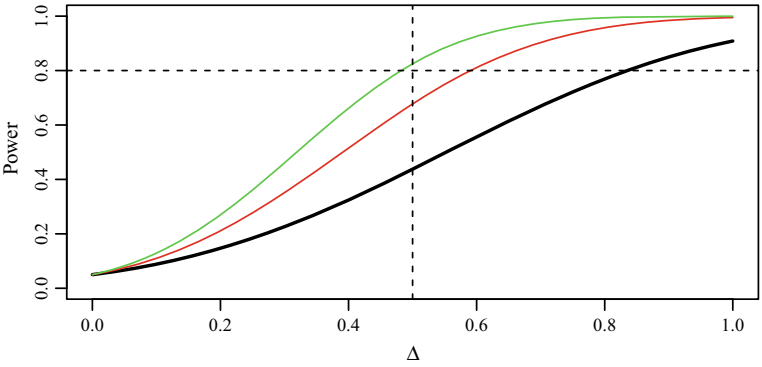

Fig. 5.16 Powell's badly scaled test function: Power as a function of the relevant difference Δ for a two-parallel-group experiment (black = 40, red =80, and green = 120 runs). If the relevant difference is 0.5, n = 120 runs per algorithm are needed for 80% power

5.8.3.2 Performing the Experiments: Powell's Badly Scaled Test Function

The full experiments can be conducted as follows. Results from the pre-experimental runs will be "recycled", only 100 additional runs must be performed.

A graphical inspection is shown in Fig. 5.17, which shows a histogram of the results. As expected, both algorithms are able to find improvements, but are affected by outliers.

Figure 5.18 shows a histogram of the differences. The numerical summary of these results is

```
## [1] "CG"
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.003772  0.094777  0.323066  0.998882  0.710185 22.565737
## [1] "NM"
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000026 0.110220 0.329906 0.757107 1.040480 8.932574
## [1] "diff"
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -8.81704 -0.25991 -0.02496  0.24177  0.36554 21.58666
```

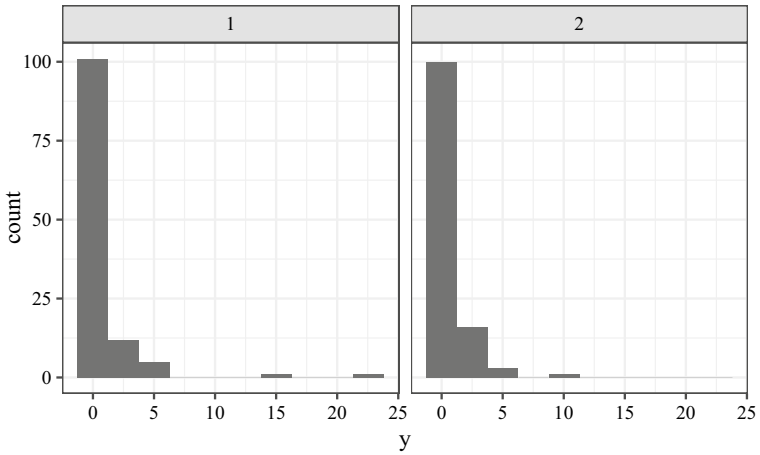

Fig. 5.17 Results from CG and NM on Powell's badly scaled test function (n = 120). Histograms to inspect outliers

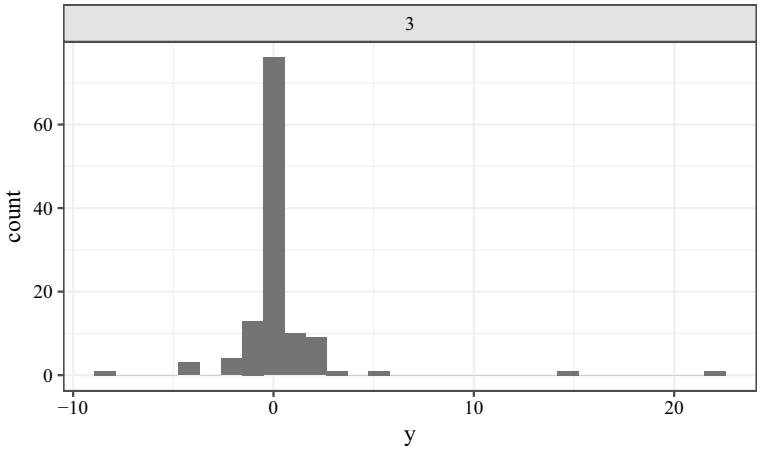

Fig. 5.18 Powell's badly scaled test function: Difference between CG and NM results (y = CG–NM)

Table 5.6 Experiment III: Result Analysis

p-value	Decision	Power	Cohen's d	Hedge's g	Severity
0.17	H0 not rejected	0.6274005	0.1184404	0.1180668	$\Delta \geq 0.5$ are well supported

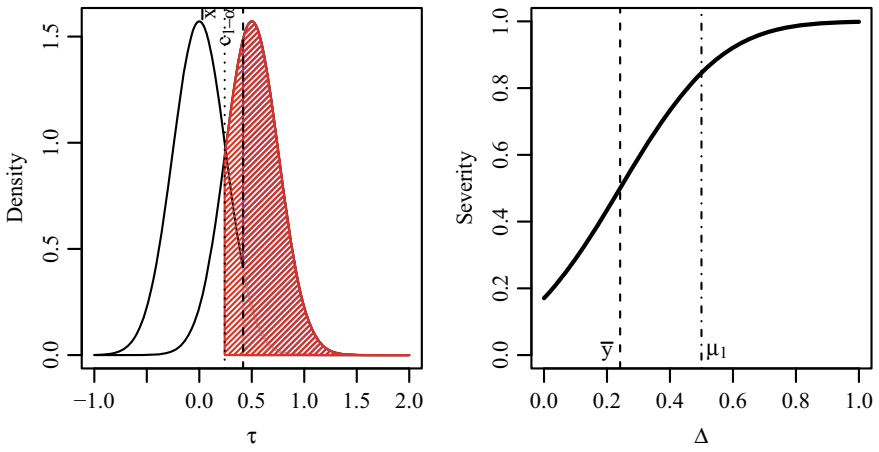

Fig. 5.19 Powell's badly scaled test function: Left: Severity of not rejecting H_0 (red), power (blue) for a target relevant difference of $\Delta = 0.5$. Right: Severity of not rejecting H_0 as a function of Δ

The summary result statistic is presented in Table 5.6. The effect size suggests that the difference is of a smaller magnitude. For the chosen $\Delta = 0.5$, the severity value is at 0.85 and thus it strongly supports the decision of not rejecting the H_0.

5.8.3.3 Discussion

Results from these runs can be analyzed using severity. The sample mean of the differences is $\bar{y} = 0.24$, so method NM might be superior. However, the median is negative. It is not obvious, which method is superior. The corresponding severity plot is shown in Fig. 5.19.

5.9 Summary and Discussion

Simply proving that there is a difference between the performance of the methods, e.g., by performing a one-sided test, is in many situations not sufficient: one needs to show that this difference is relevant. Severity was introduced as one way to tackle this problem.

A research question is necessary, e.g., if we are facing a real-world problem that has similar structural properties (discovered by landscape analysis; see Mersmann et al. 2011) as the artificial test function. Then, it might be interesting to see whether a gradient-based method (CG) is superior compared to a gradient-free method (NM). Even when theoretical results are available, they should be validated (numerical instabilities, dependencies on starting points, etc.).

Finally, at the end of this chapter, we may ask: *Why severity?* An optimization algorithm, e.g., $\mathcal{A}^+$, has achieved a high success rate with a test problem: the optimum can be determined in 96.3% of the cases. Consider the following situations:

- Let us first assume that an algorithm, say $\mathcal{A}^-$, which has no domain knowledge, only achieves such a high success rate as $\mathcal{A}^+$ in very rare exceptional cases. Is this score a good indication that $\mathcal{A}^+$ is well suited to solve this problem? In this case, based on the test results of $\mathcal{A}^+$ and $\mathcal{A}^-$, the conclusion would be justified.
- Next, suppose that Algorithm $\mathcal{A}^-$, which does not use domain knowledge, would have no problem having a score of up to 96%. Again, we can ask the same question: is this 96.3% score good evidence that $\mathcal{A}^+$ is well suited for this test problem? Based on information about the results of $\mathcal{A}^+$ and $\mathcal{A}^-$, in this case the conclusion would rather not be justified.

The severity provides a meta-statistical concept to identify these effects, which are also known in the literature as floor and ceiling effects (Cohen 1995).