# Statistical Comparison of Regions in the Swiss Feed Database

**Bachelor Thesis in Computer Science**
provided from:

**Martin Leimer**
Olten, Switzerland

Made at the
Department of Informatics
at the University of Zurich
Prof. Dr. M. Böhlen

Supervisor: Andrej Taliun
Delivery date: August $16^{th}$, 2012

## Acknowledgements

I would like to express my deepest appreciation to Dr. Andrej Taliun, who has guided, motivated and supported me through the entire development of this thesis. I also like to appreciate Prof. Dr. Michael Böhlen who has given me the opportunity to develop statistical tests on the Swiss Feed Database.

Many thanks also to Annelies Bracher and Peter Stoll, who proposed me useful literature and explained me some statistics. I also would like to acknowledge the guidance through the Agroscope company, where I could directly see how feed samples are stored, identified and measured. This remembered me once again, that my work will help researches doing their work and that it is not done on any artificial database, but on a real, frequent used database.

# Abstract

In this bachelor thesis we develop an approach how to find similar regions and how to calculate the probability to which degree two regions are equal. This calculation is based on nutrient measurements on feed samples in the Swiss Feed Database. We encounter this challenge using different statistical tests, which we finally implement on the Swiss Feed Database. Moreover, we will focus on an optimized implementation of the developed algorithm. The experimental evaluation reveals that the similarity probabilities of the *top-k similar regions* algorithm can be computed in reasonable time. Further, it will be shown that there indeed can be very similar regions.

## Zusammenfassung

In dieser Bachelorarbeit entwickeln wir ein Vorgehen um ähnliche Regionen zu finden und zu welcher Wahrscheinlichkeit dies der Fall ist. Die Berechnung basiert dabei auf Nährstoffmessungen von Futterproben in der Schweizerischen Futtermitteldatenbank. Wir entgegnen dieser Herausforderung mit der Anwendung verschiedener statistischer Tests, welche wir schliesslich auf der Schweizerischen Futtermitteldatenbank implementieren. Im Übrigen legen wir den Fokus auf eine optimierte Implementierung des entwickelten Algorithmus. Die experimentelle Evaluierung zeigt auf, dass die Ähnlichkeits-Wahrscheinlichkeiten vom *k meist ähnlichen Regionen*-Algorithmus in sinnvoller Zeit berechnet werden können. Im Übrigen werden wir aufzeigen dass in der Tat sehr ähnliche Regionen existieren können.

# Contents

1 Introduction 6

2 The Swiss Feed Database 7
- 2.1 Current online interface . . . . . . . . 7
- 2.2 The Swiss Feed Data . . . . . . . . 9

3 Region Comparison 12
- 3.1 Notation and preliminary definitions . . . . . . . . 15
- 3.2 Shapiro-Wilk-Test . . . . . . . . 17
- 3.3 F-Test . . . . . . . . 19
- 3.4 t-Test . . . . . . . . 21
  - 3.4.1 Two sample t-Test for equal variance . . . . . . . . 21
  - 3.4.2 Welch-t-Test . . . . . . . . 22
  - 3.4.3 Similarity probability . . . . . . . . 23

4 Top-k similar regions 25
- 4.1 Views . . . . . . . . 26
  - 4.1.1 top_k_similar_regions . . . . . . . . 27
- 4.2 region_summary_data . . . . . . . . 32
- 4.3 region . . . . . . . . 35
- 4.4 aggregated_data . . . . . . . . 35

5 Implementation 37

6 Evaluation 39
- 6.1 Impact on the number of measurements . . . . . . . . 39
- 6.2 Impact on the number of distinct nutrients . . . . . . . . 40
- 6.3 Impact on the number of distinct regions . . . . . . . . 40
- 6.4 Query execution plan . . . . . . . . 41
- 6.5 Top k (dis)similar regions . . . . . . . . 43

7 Conclusions and Future Work 45

References 46

# 1 Introduction

Comparison of regions based on the nutrient measurements is highly desirable and novel challenge to the Swiss Feed Database. It allows researches to find similar regions, on which they are able to make new conclusion about the nutrient distribution in Switzerland. Moreover, as measurements are costly, in case of strong similarities, regions may be less often measured in future, whereas new regions can be examined instead. This contributes to a more complete and meaningful Swiss Feed Database in future.

In this thesis we investigate statistical techniques in detail to compare the similarity of two sets of random variables and, as the core result, we develop an efficient approach to compare regions in the feed data. We employ a number of statistical tests: first, we use the Shapiro-Wilk-Test to verify whether the underlying measurements are normally distributed in both regions, second, we use the F-Test to compare the variances of measurements and, at last, we employ the t-Test to derive the final probability how much the two regions can be considered similar. Further, we extend our approach to find the top-k similar regions, i.e., for the given region we aim to find the most similar regions which are possibly separated by the big distances. Note, that for a human such a task is infeasible because of the high number (currently more than 1000) of distinct regions and nutrients. To ensure the low execution time, we first develop data views which aggregate the feed data at necessary projections and, next, we implement statistical procedures as user defined aggregate functions in PostgreSQL.

Finally, we evaluate our approach on the execution time of finding the top-k similar regions. It reveals that the computation can be done in reasonable time as long as the user selects only a few nutrients at once or defines other criteria a measurement must fulfil. Furthermore, we will take a focus on how the execution times varies if there are more or fewer measurements, distinct nutrients or locations in the Swiss Feed Database. It is shown, that the algorithm depends linearly on the number of them. Moreover, we show the existence of very similar and dissimilar regions for some example.

This thesis is organized as follows. In Section 2 the current Swiss Feed Database with its data is being presented. In Section 3, the statistical calculations are explained in detail. The algorithm how to find the top-k similar regions is shown in Section 4. Section 5 and 6 describes the implementation and the evaluation of the implemented algorithm. Finally, conclusions and forecasts are taken in Section 7.

# 2 The Swiss Feed Database

The Swiss Feed Database is a public service for researchers, farmers and agriculture companies supplied by Agroscope and University of Zurich. The database contains right now 920 different feed types and nearly four million nutrient measurements, mostly gathered in Switzerland but also in Europe. Each measurement contains information about the corresponding feed, nutrient, origin, harvest time and the feed sample it was taken from. Usually, multiple measurements on different nutrients are taken from the same feed sample. Furthermore, the same nutrient is tested more than once from the same feed sample in order to decrease erroneous measurements. However, a feed sample is not tested on every nutrient, as this is time consuming and costly.

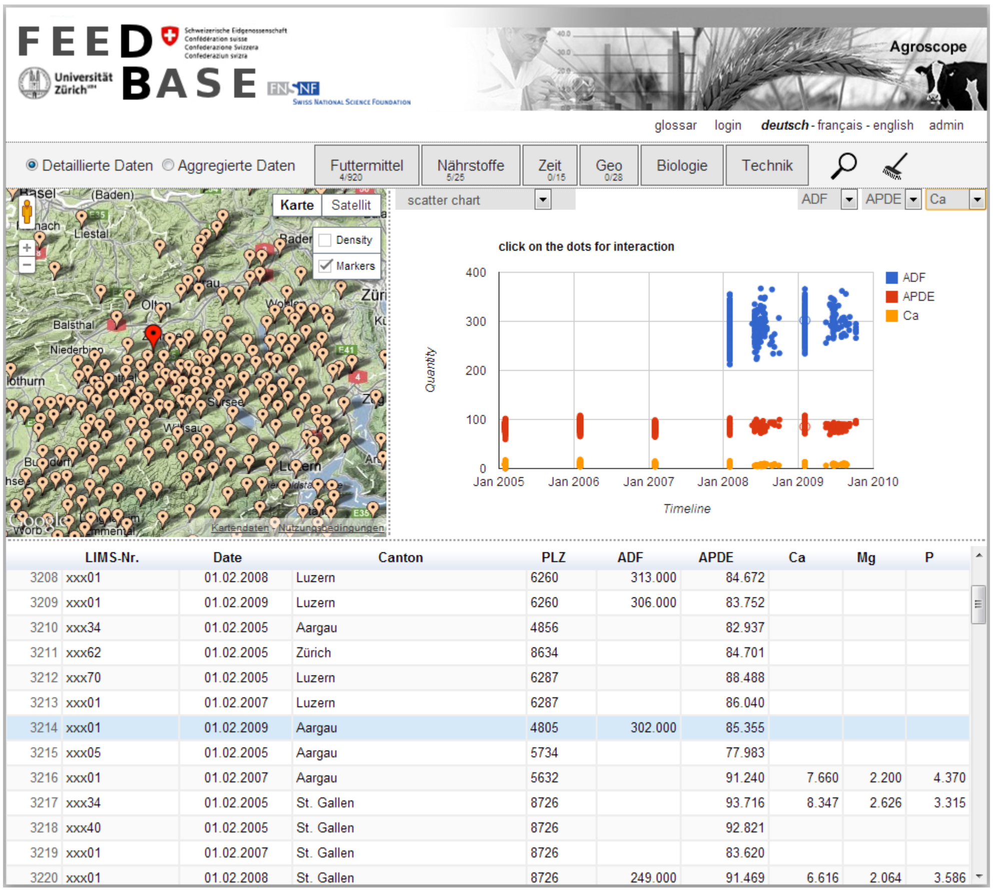
## 2.1 Current online interface

Users can access the database through an online interface, which is made available in different languages, in particular in English, French and German. Through this interface, users can makes searches on the measurements in the Swiss Feed Database.

A search request consists of a selection of feed types and nutrients measured for those. Moreover, the user may choose where the measurement must come from or when it must have been gathered.

The measurements which fulfil the search request will then be returned to the user, as seen in Figure 2.1. The result page is divided into three parts:

- Map: On the top left, each measurement is assigned with a flag at the location where it was gathered. If a user clicks on a flag, the corresponding row in the table below gets highlighted.
- Table: On the bottom, the table lists all feed samples which fulfil the search request with the measured nutrient quantities. Each feed sample is shown with parts of its LIMS-Nr. which uniquely identifies a feed sample, the date when the feed sample was gathered and its origin.
- Statistics: On the top right, different statistics are available for the user. In the row and scatter chart, the nutrients quantities are displayed over the time when they were gathered. The statistics of nutrients section discloses some informations about the measurements, like the average, variance, smallest and largest quantity of a nutrient which fulfils the search request. In addition, in the region comparison section, the user can select two regions and run the ANOVA F-Test or the in this thesis developed t-Test, which returns an approximate probability of how much the selected regions can be considered equal.



| | LIMS-Nr. | Date | Canton | PLZ | ADF | APDE | Ca | Mg | P |
|---|---|---|---|---|---|---|---|---|---|
| 3208 | xxx01 | 01.02.2008 | Luzern | 6260 | 313.000 | 84.672 | | | |
| 3209 | xxx01 | 01.02.2009 | Luzern | 6260 | 306.000 | 83.752 | | | |
| 3210 | xxx34 | 01.02.2005 | Aargau | 4856 | | 82.937 | | | |
| 3211 | xxx62 | 01.02.2005 | Zürich | 8634 | | 84.701 | | | |
| 3212 | xxx70 | 01.02.2005 | Luzern | 6287 | | 88.488 | | | |
| 3213 | xxx01 | 01.02.2007 | Luzern | 6287 | | 86.040 | | | |
| 3214 | xxx01 | 01.02.2009 | Aargau | 4805 | 302.000 | 85.355 | | | |
| 3215 | xxx05 | 01.02.2005 | Aargau | 5734 | | 77.983 | | | |
| 3216 | xxx01 | 01.02.2007 | Aargau | 5632 | | 91.240 | 7.660 | 2.200 | 4.370 |
| 3217 | xxx34 | 01.02.2005 | St. Gallen | 8726 | | 93.716 | 8.347 | 2.626 | 3.315 |
| 3218 | xxx40 | 01.02.2005 | St. Gallen | 8726 | | 92.821 | | | |
| 3219 | xxx01 | 01.02.2007 | St. Gallen | 8726 | | 83.620 | | | |
| 3220 | xxx01 | 01.02.2008 | St. Gallen | 8726 | 249.000 | 91.469 | 6.616 | 2.064 | 3.586 |

Figure 2.1: Result of a search request in the online interface of the Swiss Feed Database

## 2.2 The Swiss Feed Data

The Swiss Feed Database contains the following information:

- Feeds: A feed has a name and an alternative name. It is often assigned to a feed group which categorizes related feeds together. Each feed group has an id, a name and an id of the parent feed group which categorizes related feed groups if existent. The *feed_key* uniquely identifies a feed. Example:

| id | feed_key | name | alt_name | feed_group_id | parent_feed_group_id | feed_name |
|---|---|---|---|---|---|---|
| 1 | 668 | Ameisensäure | VE-UB | 19 | 18 | Säuren und Salze |
| 2 | 889 | Weizenkeime | GT-WEKM | 5 | 3 | Stärkegewinnung |
| 3 | 1302 | - | - | 18 | 1 | Zusatzstoffe |

- Nutrients: A nutrient has a name and often an abbreviation and a description. Sometimes it is related to a nutrient group which categorizes related nutrients. The *nutrient_key* uniquely identifies a feed. Example:

| id | nutrient_key | name | abbreviation | description | nutrient_group |
|---|---|---|---|---|---|
| 1 | 38 | Arachinsäure | C20 | Arachinsäure | Einzelfettsäuren |
| 2 | 355 | Zucker | ZUCK | Zucker alkohollöslich | Kohlenhydrate |
| 3 | 359 | Kalzium | CA | Kalzium | Mineralstoffe |

- Feed Samples: A feed sample has LIMS-Nr. Furthermore it is described with up to two lab descriptions, the way it was prepared and the source among other biological information. A feed sample is uniquely identified by the *lims_number* and the *sample_key* as well. Example:

| id | sample_key | lims_number | preparation_de | info_1 | info_2 | provenance |
|---|---|---|---|---|---|---|
| 1 | 30933 | 290688-9 | KEINE | 58.2 P5_Sevrage Sang | - | - |
| 2 | 61540 | 194158-4 | LYO-BR1 | 9615 K A FECES | A2 | DIG115 |

- Origins: An origin contains geographical information like postal code, city, canton, country, latitude, longitude, its corresponding altitude class and the region name and number it belongs to according the classification of Agridea. The *origin_key* uniquely identifies an origin. Example:

| id | origin_key | pc | city | canton | country | altitude | r_id | r_name | lat | lon |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1038 | 1085 | Vulliens | Vaud | Switzerland | 600 - 799 | 2 | VD | 46.62 | 6.79 |
| 2 | 1212 | 3054 | Schüpfen | Bern | Switzerland | < 600 | 4 | Mittelland | 47.03 | 7.37 |

- Times: A time contains a full date, together with its year and month and the season. A moment indicates the type of the time, meaning whether it stands for the harvesting, sampling, arrival or analysis time. The time is uniquely identified by the *time_key*. Example:

| **id** | **time_key** | **t_day** | **t_year** | **t_month** | **season_en** | **moment** |
|---|---|---|---|---|---|---|
| 1 | 64 | 2009-06-30 | 2009 | 6 | Summer | 1 |
| 2 | 9548 | 1993-09-25 | 1993 | 9 | Autumn | 2 |

- Measurements: A measurement is made on a feed sample and has a quantity, which is either converted to the dry matter basis or not. It inherits the corresponding LIMS-Nr. of the feed. In addition, it links the complementary time and origin where the feed sample was gathered, the nutrient which was analysed and the feed from which the measurement was taken from. The *measure_pkey* uniquely identifies a measurement. Example:

| **id** | **measure_pkey** | **lims_number** | **quantity** | **d_m_b** | **time** | **nutrient** | **origin** | **sample** | **feed** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 75414 | 09-22092-005 | 129 | false | 75 | 355 | 1324 | 250 | 1 |
| 2 | 3974244 | 05-14410273431 | 5.224039 | true | 9833 | 223 | 1184 | 3941 | 1 |

Descriptions and names are often translated into different languages, particular in English, German and French. Nevertheless, sometimes translations, short and alternative names as well as descriptions are missing. Furthermore, feed samples are only measured on few nutrients. This implies that in a search request with few and possible not so common nutrients, many measurements are dropped from the result. Nonetheless, the most important parameters and the quantities for typical nutrients are given mostly, making extensive search requests and statistical predictions still possible. For this, the Swiss Feed Database is modelled as seen in Figure 2.2, where the *fact_table* lists all measurements with the keys to the corresponding feed, nutrient, time and origin as explained above.

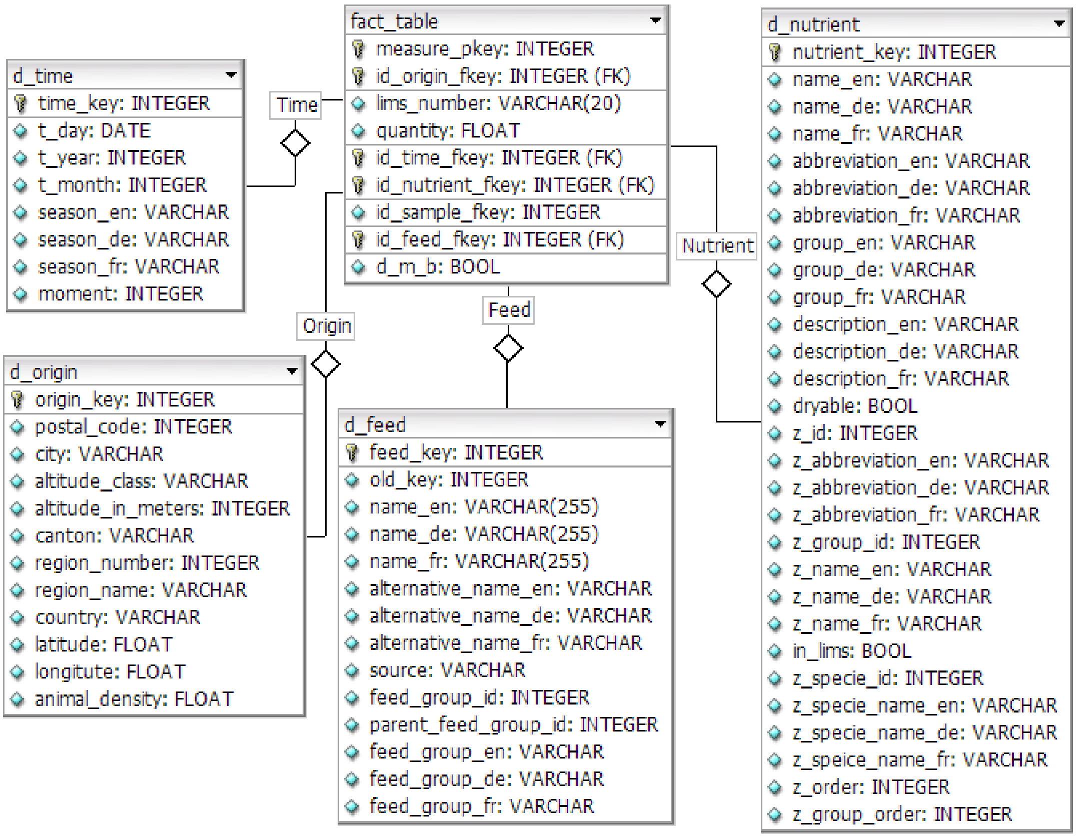


Figure 2.2: Architecture of the Swiss Feed Database

# 3 Region Comparison

Consider the map in Figure 3.1, where nutrient measurements are presented with a flag at their corresponding locations. Two circles with a radius of 10km indicate two regions. All nutrient measurements within those regions are then assigned to the nutrient measurement set $X_1$ for region 1 and $X_2$ for region 2 respectively (cf. Table 1).

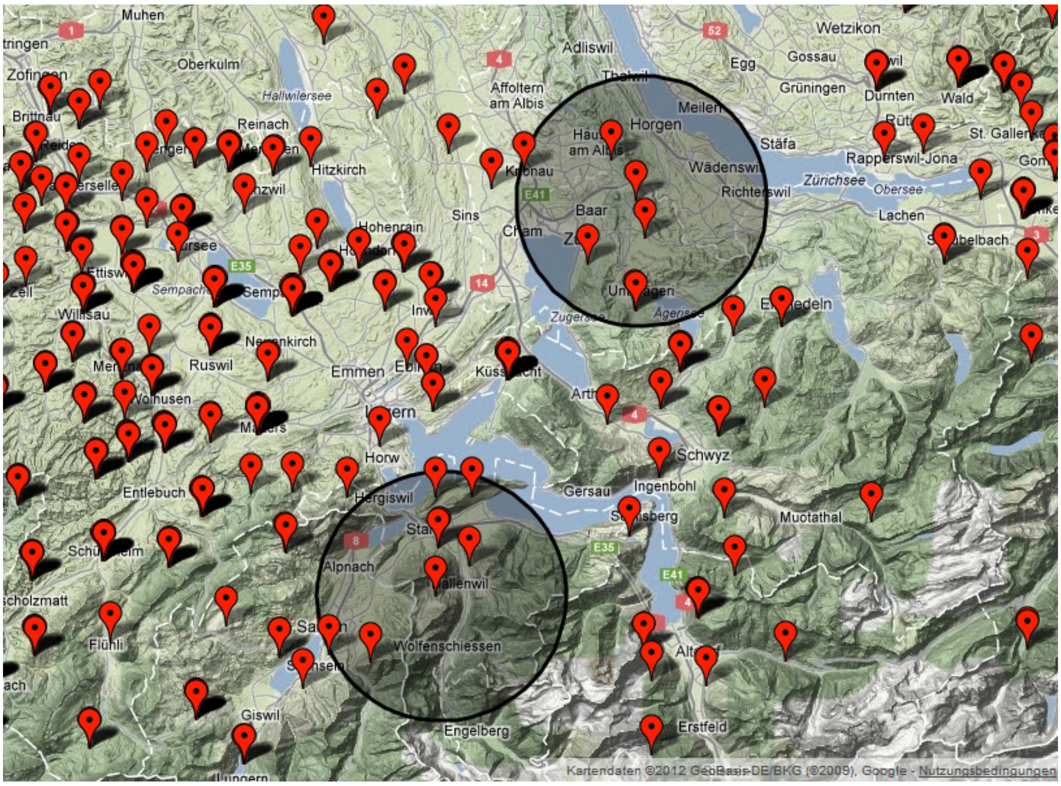


Figure 3.1: Map with nutrient measurements and two regions

Table 1: Nutrient Measurement sets $X_1$ and $X_2$ for some nutrient in region 1 and 2 respectively

| **Region** $X_1$ | 4 | 8 | 3 | 5 | 5 | 7 | 9 |
|---|---|---|---|---|---|---|---|
| **Region** $X_2$ | 6 | 6 | 4 | 5 | 9 | 7 | - |

Given this two nutrient measurement sets $X_1$ and $X_2$, we are now able to calculate the similarity between them. Schematically the steps are illustrated in Figure 3.2.

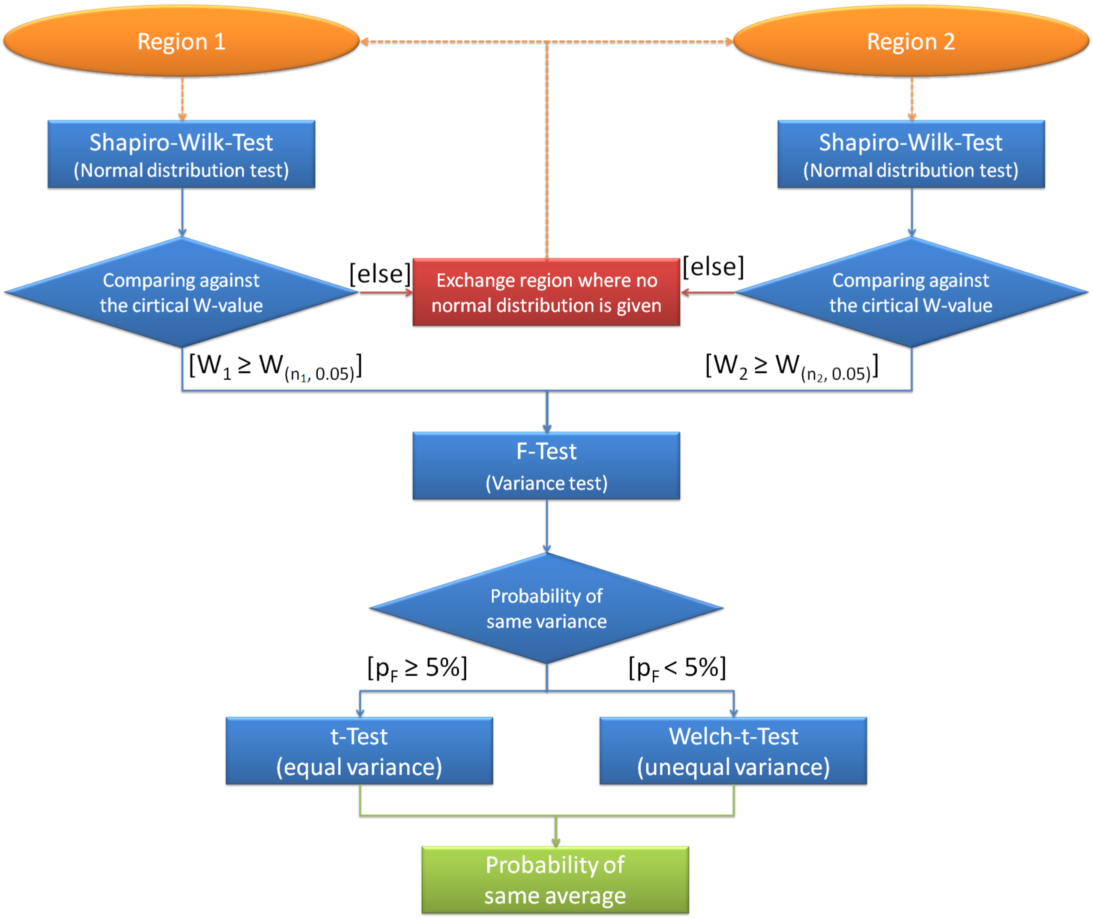


Figure 3.2: Algorithm how to calculate the similarity probability of two regions

First, we verify that both nutrient measurement sets $X_1$ and $X_2$ have normally distributed measurements using the Shapiro-Wilk-Test. For this, we calculate for each region individually the Shapiro-Wilk value, which is $W_1 = 0.949$ for region 1 and $W_2 = 0.933$ for region 2 respectively. For normal distribution, this value must be greater or equal than the critical Shapiro-Wilk value, which is $W_{(6,0.05)} = 0.803$ for the nutrient measurement set $X_1$ and $W_{(7,0.05)} = 0.788$ for $X_2$ respectively.

Given that both measurement sets are normal distributed, we compute the probability of how much the variances $s_1^2$ of the nutrient measurement set $X_1$ and $s_2^2$ of $X_2$ can be considered equal. For this we apply the F-Test which results in a probability of $p_F = 0.287$.

Finally, using the t-Test, we compute the probability of how much the averages $\overline{x_1}$ of the nutrient measurement set $X_1$ and $\overline{x_2}$ of $X_2$ can be considered similar. Since $p_F = 0.287 \geq 0.05$ we assume equal variance, which implies that we have to apply the two sample t-Test for equal variance. If this would not be the case, the Welch-t-Test should be applied. Both t-Tests returns a T value, which can then be transformed to a probability, which for our example is $p_T = 0.82$. This is also the probability that those two regions are similar.

Reconsidering the introduced example, the most important results are shown in Table 2.

Table 2: Results of example nutrient measurement sets $X_1$ and $X_2$

| | $n_i$ | $\overline{x_i}$ | $s_i^2$ | $W_i$ | $W_{(n_i,0.05)}$ | $p_F$ | $p_T$ |
|---|---|---|---|---|---|---|---|
| **Region** $X_1$ | 5.86 | 4.81 | 7 | 0.949 | 0.803 | 0.287 | 0.82 |
| **Region** $X_2$ | 6.17 | 2.97 | 6 | 0.933 | 0.788 | | |

## 3.1 Notation and preliminary definitions

The following table summarizes the notation used in this thesis:

| Notation | Meaning |
|---|---|
| $a_{(n_i,k')}$ | $k'^{th}$ constant for Shapiro-Wilk-Test and nutrient measurement set of size $n_i$ |
| $B(\alpha, \beta)$ | Complete beta function |
| $B_x(\alpha, \beta)$ | Incomplete beta function |
| $F$ | value of F-Test |
| $FGF_i$ | $i^{th}$ degree of freedom of F-Test |
| $FGT$ | degree of freedom of t-Test |
| $I_x(\alpha, \beta)$ | Regularized incomplete beta function |
| $n_i$ | number of measurements in region $i$ |
| $p_F$ | probability that the variances are equal |
| $p_T$ | probability that the averages are equal |
| $s_i^2$ | variance of measurements in region $i$ |
| $T$ | value of t-Test |
| $W_i$ | value of Shapiro-Wilk-Test in region $i$ |
| $W_{(n_i,0.05)}$ | Critical Shapiro & Wilk value |
| $\overline{x_i}$ | average in region $i$ |
| $x_{ij}$ | $j^{th}$ measurement value in region $i$ |
| $x_{i(z')}$ | $z'^{th}$ lowest measurement value in measurement set of region $i$ |

Moreover, the following definitions completes the statistics presented in the subsections afterwards.

We use $x_{ij}$ to denote an individual nutrient measurement in a nutrient measurement set $X_i = \{x_{i1}, x_{i2}, x_{i3}, ..., x_{in}\}$ of region $i$.

Example: For our two regions 1 and 2 we have the following nutrient measurement sets $X_1$ and $X_2$ with their individual nutrient measurements $x_{ij}$:

$$X_1 = \{4, 8, 3, 5, 5, 7, 9\}$$
$$\text{with } x_{11} = 4, x_{12} = 8, x_{13} = 3, x_{14} = 5, x_{15} = 5, x_{16} = 7, x_{17} = 9$$

$$X_2 = \{6, 6, 4, 5, 9, 7\}$$
$$\text{with } x_{21} = 6, x_{22} = 6, x_{23} = 4, x_{24} = 5, x_{25} = 9, x_{26} = 7$$

Definition: Let $x_{ij}$ be the $j^{th}$ nutrient measurement the nutrient measurement set $X_i$ and $n_i$ be the number of measurements in this set. Then, the average $\overline{x_i}$ of the nutrient measurement set $X_1$ is:

$$\overline{x_i} = \frac{\sum_{j=1}^{n_i} x_{ij}}{n_i} \tag{1}$$

Example: For our two measurement sets $X_1$ and $X_2$ we receive the following averages:

$$\begin{aligned} \overline{x_1} &= \tfrac{4+8+3+5+5+7+9}{7} = 5.86 \\ \overline{x_2} &= \tfrac{6+6+4+5+9+7}{6} = 6.17 \end{aligned}$$

Definition: Let $x_{ij}$ be the $j^{th}$ nutrient measurement, $\overline{x_i}$ the average and $n_i$ be the number of measurements in the nutrient measurement set $X_1$. Then, the variance $s_i^2$ of the nutrient measurement set $X_i$ is:

$$s_i^2 = \frac{\sum_{j=1}^{n_i} (x_{ij} - \overline{x_i})^2}{n_i - 1} \tag{2}$$

Example: For our two measurement sets $X_1$ and $X_2$ we get the following variances:

$$\begin{aligned} s_1^2 &= \tfrac{(4-5.86)^2+(8-5.86)^2+(3-5.86)^2+(5-5.86)^2+(5-5.86)^2+(7-5.86)^2+(9-5.86)^2}{7-1} = 4.81 \\ s_2^2 &= \tfrac{(6-6.17)^2+(6-6.17)^2+(4-6.17)^2+(5-6.17)^2+(9-6.17)^2+(7-6.17)^2}{6-1} = 2.97 \end{aligned}$$

Definition: Let $\gamma \in \mathbb{R}$, $x \in [0, 1]$ and $\alpha, \beta > 0$. Then the beta functions are as follows:

- Complete beta function

$$B(\alpha, \beta) = \int_0^1 \gamma^{\alpha-1} * (1-\gamma)^{\beta-1} \mathrm{d}\gamma \tag{3}$$

- Incomplete beta function

$$B_x(\alpha, \beta) = \int_0^x \gamma^{\alpha-1} * (1-\gamma)^{\beta-1} \mathrm{d}\gamma \tag{4}$$

- Regularized incomplete beta function

$$I_x(\alpha, \beta) = \frac{B_x(\alpha, \beta)}{B(\alpha, \beta)} \tag{5}$$

Example: For the complete beta function, its values for some $\alpha \in [-2, 2]$ and $\beta \in [-2, 2]$ can be seen in Figure 3.3.

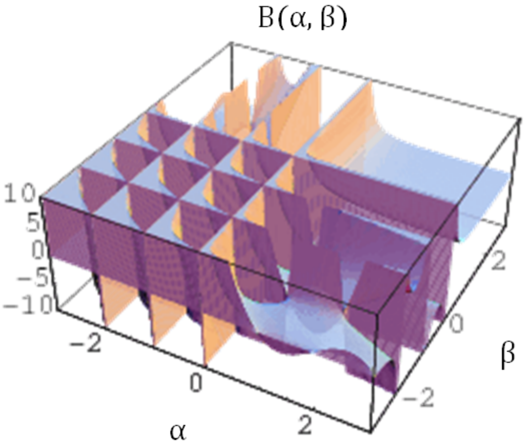


Figure 3.3: Values of the Complete Beta function for $\alpha \in [-2, 2]$ and $\beta \in [-2, 2]$

Explanation: The beta function is common for defining probability density and cumulative distribution functions, especially for the F-Test and t-Test. $\alpha$ and $\beta$ refer to the degrees of freedom of the first and second nutrient measurement set respectively. The $\gamma$ will omit when computing the integral. We will further explain this concept when we use this functions.

In this thesis, whenever we speak of a level of significance, we mean a probability up to which extent we might consider something as unequal although it would be similar. We will always use a level of significance of 5% as this is one of the most often used probabilities. It is not possible to set this probability to zero, as this would imply that we consider everything as similar. Of course, this is not the case and would imply that our result is more likely to be wrong.

## 3.2 Shapiro-Wilk-Test

This section describes the Shapiro-Wilk-Test which verifies if a nutrient measurement set can be considered normally distributed or not. According [2], the Shapiro-Wilk value $W_i$ must be calculated first, before it can be compared against the critical Shapiro-Wilk value $W_{(n_i,0.05)}$ for the considered measurement set $X_i$.

Definition: Let $x_{i(z')}$ be the $z'^{th}$ lowest value of the nutrient measurement set $X_i$ and $a_{(n_i,k')}$ the $k'^{th}$ Shapiro-Wilk constant for a nutrient measurement set $X_i$ of size $n_i$ as listed in the Appendix 1. Then the Shapiro-Wilk value is:

$$W_i = \begin{cases} \frac{\left(\sum_{k'=1}^{\frac{n_i}{2}} \left(a_{(n_i,k')} * (x_{i(n_i-k'+1)} - x_{i(k')})\right)\right)^2}{(n_i-1)*s_i^2}, & \text{if } n_i \text{ is even} \\ \frac{\left(\sum_{k'=1}^{\frac{n_i-1}{2}} \left(a_{(n_i,k')} * (x_{i(n_i-k'+1)} - x_{i(k')})\right)\right)^2}{(n_i-1)*s_i^2}, & \text{if } n_i \text{ is uneven} \end{cases} \tag{6}$$

Explanation: In every sum sequence, the two most different remaining measurements in the corresponding nutrient measurement set $X_i$ are multiplied with the $k'^{th}$ Shapiro-Wilk constant $a_{(n_i,k')}$. The higher the difference of this two measurements, the higher the numerator of the Shapiro-Wilk-Test. But as the variance increases faster, the lower the Shapiro-Wilk value and the less likely normal distribution occurs. The Shapiro-Wilk-Test returns $W_i \in [0, 1]$.

Example: For our nutrient measurement sets $X_1$ and $X_2$, this results in the following Shapiro-Wilk values:

$$W_1 = \frac{(0.6233*(9-3)+0.3031*(8-4)+0.1401*(7-5))^2}{(7-1)*4.81} = 0.949$$
$$W_2 = \frac{(0.6233*(9-4)+0.3031*(7-5)+0.1401*(6-6))^2}{(6-1)*2.97} = 0.933$$

Definition: Let $W_i$ be the Shapiro-Wilk value and $W_{(n_i,0.05)}$ be the critical Shapiro-Wilk value to a level of significance of 5% and a nutrient measurement set $X_i$ of size $n_i$ as listed in the Appendix 2. Then, normal distribution can be considered if:

$$W_i \geq W_{(n_i,0.05)} \tag{7}$$

Explanation: If the equation is satisfied, the nutrient measurement set $X_i$ can be considered as normally distributed. The higher the level of significance and the number of measurements $n_i$ in the considered measurement set $X_i$ the higher the critical Shapiro-Wilk value and the less likely normal distribution can be considered.

Example: For our nutrient measurement sets $X_1$ and $X_2$, both do not violate the equation and can therefore be considered normally distributed to a level of significance of 5%:

$$W_1 = 0.949 \geq W_{(7,0.05)=0.803}$$
$$W_2 = 0.933 \geq W_{(6,0.05)=0.788}$$

If a nutrient measurement set cannot be considered normally distributed, further calculations are not possible for this nutrient measurement set $X_i$ with the algorithms developed in this thesis.

## 3.3 F-Test

This section describes the F-Test which verifies if the variances of two normally distributed measurement sets can be considered equal or not. According [2] and [3], we first have to calculate the F value, before we can derive the probability to which extent they are similar.

Definition: Let $s_1^2$ and $s_2^2$ be the variance of the nutrient measurement sets $X_1$ and $X_2$ respectively. Then the F-Test is:

$$F = \begin{cases} \frac{s_1^2}{s_2^2}, & \text{if } s_1^2 \geq s_2^2 \\ \frac{s_2^2}{s_1^2}, & \text{else} \end{cases} \tag{8}$$

Explanation: The F-Test divides the larger variance through the smaller variance of the two nutrient measurement sets. The result is a F value, a positive number greater or equal than 1.

Example: Since the variance of the nutrient measurement set $X_1$ of Region 1 is higher than this of the nutrient measurement set $X_2$ of Region 2, the F value is:

$$F = \frac{4.81}{2.97} = 1.62$$

Given the F value, we can derive the probability how much the variances of the two nutrient measurement sets $X_1$ and $X_2$ can be considered equal. For this we introduce the probability density function for the F-Distribution as seen in Figure 3.4. It is a function, whose area below is always 1 between 0 and infinity. We now cut the function at the position of the obtained F value, whereupon the tail correlates to the probability how similar the two nutrient measurement sets $X_1$ and $X_2$ are according their variance. Reconsidering our example, we cut at $F = 1.62$, whereupon the blue shaded tail corresponds to 0.287, the searched probability.

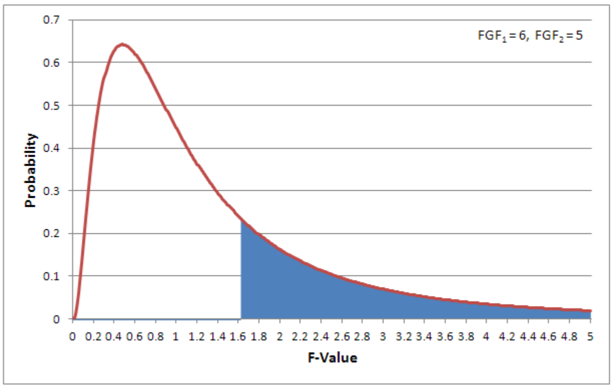


Figure 3.4: probability density function of F-Distribution for some degrees of freedom $FGF_1 = 6$ and $FGF_2 = 5$

Definition: Let $n_1$ and $n_2$ be the number of measurements and $s_1^2$ and $s_2^2$ the variance in the nutrient measurement set $X_1$ and $X_2$ respectively. Then the degrees of freedom for the F-Test are as follows:

$$FGF_1 = \begin{cases} n_1 - 1, & \text{if } s_1^2 \geq s_2^2 \\ n_2 - 1, & \text{else} \end{cases} \tag{9}$$

$$FGF_2 = \begin{cases} n_2 - 1, & \text{if } s_1^2 \geq s_2^2 \\ n_1 - 1, & \text{else} \end{cases} \tag{10}$$

Explanation: The degrees of freedom influence the shape of the probability density function of the F-Distribution. For example, in Figure 3.4, lower values of the degree of freedom would flatten the tail of the density function.

Example: Since the variance of the nutrient measurement set $X_1$ is higher than this of $X_2$, the degrees of freedom for the F-Test are:

$$FGF_1 = 7 - 1 = 6$$
$$FGF_2 = 6 - 1 = 5$$

We can now compute the corresponding probability, how much the two nutrient measurement sets $X_1$ and $X_2$ can be considered equal according their variances. As was explained earlier, we need to evaluate the area enclosed by the tail of the probability density function. In other words, the probability we search is 1 minus the probability received from the cumulative distribution function.

Definition: Given the F value denoted as $F$, and the degrees of freedom $FGF_1$ and $FGF_2$, the probability to which extent two measurement sets $X_1$ and $X_2$ can be considered equal according their variances is:

$$p_F = I_k\left(\frac{FGF_2}{2}, \frac{FGF_1}{2}\right) \tag{11}$$

where $k = \frac{FGF_2}{FGF_2 + FGF_1 * F}$.

Explanation: The higher the received probability the more likely the variances are similar. We will consider the variances of the two nutrient measurement sets $X_1$ and $X_2$ as equal, if the probability is at least as high as the level of significance, which is 5%.

Example: Given the nutrient measurement sets $X_1$ and $X_2$ of region 1 and 2 respectively, the probability that the two measurement sets can be considered equal according their variances is:

$$p_F = I_{0.35}(3, 3.5)) = 0.287$$

Since $p_F = 0.287$ is greater than 5%, the level of significance, we consider the variances of the two nutrient measurement sets $X_1$ and $X_2$ as equal.

## 3.4 t-Test

This section describes the t-Test which calculates the probability of how much two nutrient measurement sets can be considered equal according their averages. This will also be the probability we use to describe to which extent two regions are similar. According [2] and [3], we first have to calculate the T value, before we can derive the probability to which extent the two nutrient measurement sets, respectively regions are similar.

Depending if the variances of the two nutrient measurement sets $X_1$ and $X_2$ can be considered equal ($p_F \geq 0.05$) or not ($p_F < 0.05$), either the two sample t-Test for equal variance, or the Welch-t-Test must be applied. According our example, the two sample t-Test for equal variance must be applied.

### 3.4.1 Two sample t-Test for equal variance

Definition: Let $s_i$ be the variance, $\overline{x_i}$ be the average and $n_i$ the number of measurements in the nutrient measurement set $X_i$. Then the two sample t-Test is:

$$T = \begin{cases} \frac{|\overline{x_1} - \overline{x_2}|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, & \text{if } n_1 = n_2 \\ \frac{|\overline{x_1} - \overline{x_2}|}{\sqrt{\frac{s_1^2 * (n_1 - 1) + s_2^2 * (n_2 - 1)}{n_1 + n_2 - 2}}} * \sqrt{\frac{n_1 * n_2}{n_1 + n_2}}, & \text{else} \end{cases} \tag{12}$$

Explanation: The T value is obtained using the average, variance and the number of measurements in both nutrient measurement sets. The higher the T value, the less likely the two nutrient measurement sets $X_1$ and $X_2$ can be considered equal according their averages.

Example: Given the variances $s_1^2$ and $s_2^2$ of the two nutrient measurement sets $X_1$ and $X_2$ respectively, the T value is:

$$T = \frac{|5.86 - 6.17|}{\sqrt{\frac{4.81 * (7-1) + 2.97 * (6-1)}{7+6-2}}} * \sqrt{\frac{7 * 6}{7+6}} = 0.14$$

To get the probability how much the averages of the two nutrient measurement sets $X_1$ and $X_2$ can be considered equal, we also have to define the degree of freedom, as they influence the probability density function of the t-Distribution as explained in Section 3.4.3.

Definition: Let $n_1$ and $n_2$ be the number of measurements in the nutrient measurement set $X_1$ and $X_2$ respectively. Then, the degree of freedom for the t-Test is:

$$FGT = n_1 + n_2 - 2 \tag{13}$$

Explanation: The degree of freedom has an influence on the probability density function of the t-Distribution.

Example: Given the nutrient measurement sets $X_1$ and $X_2$, the degree of freedom for the two sample t-Test for equal variance is:

$$FGT = 7 + 6 - 2 = 11$$

### 3.4.2 Welch-t-Test

Definition: Let $s_i$ be the variance, $\overline{x_i}$ the average and $n_i$ the number of measurements in the nutrient measurement set $X_i$. Then the Welch-t-Test is:

$$T = \frac{|\overline{x_1} - \overline{x_2}|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{14}$$

Explanation: The T value is obtained using the average, variance and the number of measurements in both nutrient measurement sets. Again, the higher the T value, the less likely the two nutrient measurement sets can be considered equal according their averages.

Example: Assuming that the level of significance would have been 50% for the F-Test, implying that the variances would be considered dissimilar, the T value would have been:

$$T = \frac{|5.86 - 6.17|}{\sqrt{\frac{4.81}{7} + \frac{2.97}{6}}} = 0.26$$

To get the probability how much the averages of the two nutrient measurement sets $X_1$ and $X_2$ can be considered equal, we also have to define the degree of freedom, as they influence the probability density function of the t-Distribution as explained in Section 3.4.3.

Definition: Let $s_i$ be the variance and $n_i$ the number of measurements in the measurement set $X_i$. Then the degree of freedom for the t-Test is:

$$FGT = \begin{cases} (n_1 - 1) + \frac{(n_1+n_2)-2}{\frac{s_1^2}{s_2^2} + \frac{s_2^2}{s_1^2}}, & \text{if } n_1 = n_2 \\ \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}}, & \text{else} \end{cases} \tag{15}$$

Explanation: The degree of freedom has an influence on the probability density function of the t-Distribution.

Example: Given the nutrient measurement sets $X_1$ and $X_2$, the degree of freedom for the Welch-t-Test would be:

$$FGT = \frac{\left(\frac{4.81}{7}+\frac{2.97}{6}\right)^2}{\frac{\left(\frac{4.81}{7}\right)^2}{7-1}+\frac{\left(\frac{2.97}{6}\right)^2}{6-1}} = 10.94$$

### 3.4.3 Similarity probability

Given the T value, we can derive the probability how much the averages of the two nutrient measurement sets $X_1$ and $X_2$ can be considered equal according [3]. For this we introduce the probability density function for the t-Distribution as seen in Figure 3.5. It is a function, whose area below is always 1 between minus infinity and infinity. We now cut the function at the position of the obtained T value, whereupom the tail correlates to the half probability how similar the two nutrient measurement sets $X_1$ and $X_2$ are according their average. As the probability density function is symmetric, we just have to multiply this probability by two, or we alternatively add the blue shaded area between minus infinity and minus the T value. Reconsidering our example, we cut at $T = 0.14$, whereupon both blue shaded areas corresponds to 0.445, resulting in a total of 0.89, the searched probability.

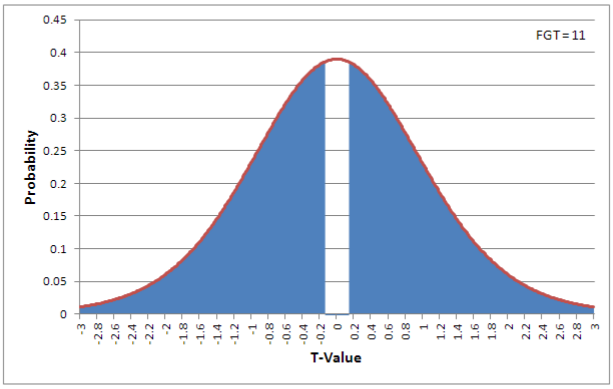


Figure 3.5: probability density function of t-Distribution for some degree of freedom $FGT = 11$

Definition: Let $T$ be the T value, and $\lfloor FGT \rfloor$ the to the next integer rounded degree of freedom not greater than $FGT$. Then, the probability to which extent two measurement sets $X_1$ and $X_2$ can be considered equal according their averages is:

$$p_T = 1 - 2 * \int_0^T \frac{\left(1+\frac{T^2}{\lfloor FGT \rfloor}\right)^{\frac{-(\lfloor FGT \rfloor+1)}{2}}}{B(0.5, 0.5 * \lfloor FGT \rfloor) * \sqrt{\lfloor FGT \rfloor}} dT \qquad (16)$$

Explanation: The higher the received probability the more likely the averages of the two nutrient measurement sets $X_1$ and $X_2$ and herewith the regions are similar.

Example: Given the nutrient measurement sets $X_1$ and $X_2$, the probability that the two nutrient measurement sets can be considered equal according their averages is:

$$p_T = 1 - 2 * \int\limits_0^T \frac{\left(1+\frac{T^2}{11}\right)^{\frac{-(11+1)}{2}}}{B(0.5,0.5*11)*\sqrt{11}} dT|_{T=0.14} = 0.89$$

This means that the regions 1 and 2 can be considered to be 89% similar.

# 4 Top-k similar regions

This section describes the algorithm how to find the most similar regions to a user selected region.

The approach for finding the top-k similar regions to a user defined region is based on the calculation of the two region comparison. Basically, it is a region comparison of every possible region with the user defined region for some nutrient, which is then ordered according the similarity probability. However in some cases regions can be pruned. These constraints are now explained in detail:

- A nutrient measurement set $X_i$ of a region $i$ must contain at least five measurements:

  $$n_i \geq 5 \tag{17}$$

  Explanation: If a region contains less than five measurements, the calculated probability may be very inaccurate. A single outlier could imply that a region would be considered as dissimilar to the user defined region although it would be similar if more measurements would have been taken from this region.

- A nutrient measurement set $X_i$ of a region $i$ may contain at most fifty measurements:

  $$n_i \leq 50 \tag{18}$$

  Explanation: If a region contains more than fifty measurements, the Shapiro-Wilk value cannot be computed, as Shapiro & Wilk listed the $a_{(n_i,k')}$ weights only for measurement sets of up to fifty measurements.

- For every region comparison, no region shall intersect the other region.

  Explanation: It is very likely that the most similar region is a region intersecting the user defined region as most of the measurements are taken from the same locations and are therefore identically. However, we are not interested in those regions as we want to find other regions not intersecting the user defined region.

- If a region has no normal distributed nutrient measurement set, the region must be ignored.

  Explanation: For not normal distributed measurement sets, the F-Test and the t-Test cannot be applied.

Given this constraints, the similarity probability can be computed for every remaining region pair. The descended ordering of this probabilities results in the most similar regions to the user defined regions.

## 4.1 Views

With the PostgreSQL, the programming language on the relational database, we design four views to calculate those similarity probabilities. They return the following tables if executed individually:

- top_k_similiar_regions: This view lists the similarity probability for every region, nutrient pair, whereupon the *origin_key* of the centre of each region is given. The similarity probability is always computed between the user defined region (not listed) and the presented region.

| id | origin_key | nutrient_key | similarity_probability |
|---|---|---|---|
| 1 | 1227 | 283 | 0.994 |
| 2 | 2039 | 283 | 0.991 |

- region_summary_data: This view lists the regions, nutrients, the number, average and variance of the corresponding nutrient measurement set. Furthermore, the Shapiro-Wilk value is listed if the nutrient measurement set can be considered normally distributed. In the other case, the displayed value is negative.

| id | origin_key | nutrient_key | count | avg | variance | shapiro_wilk |
|---|---|---|---|---|---|---|
| 1 | 1044 | 283 | 34 | 132.86 | 621.83 | 0.95 |
| 2 | 1146 | 283 | 48 | 132.19 | 181.55 | 0.98 |

- region: This view lists for every location all other locations which are located within some user defined distance including itself in multiple rows.

| id | origin_key | neighbour |
|---|---|---|
| 1 | 1038 | 1038 |
| 2 | 1038 | 1062 |
| 3 | 1043 | 1044 |

- aggregated_data: This view lists for every location, nutrient pair, respectively feed sample, the aggregated or rather average measurement quantity.

| id | nutrient_key | origin_key | aggregated_quantity |
|---|---|---|---|
| 1 | 283 | 1077 | 125.84 |
| 2 | 283 | 1102 | 111.06 |

These views are composed as seen in Figure 4.1. It lists all the needed views and the new developed aggregate user defined functions. We will explain its parts in detail, using a top to bottom approach.

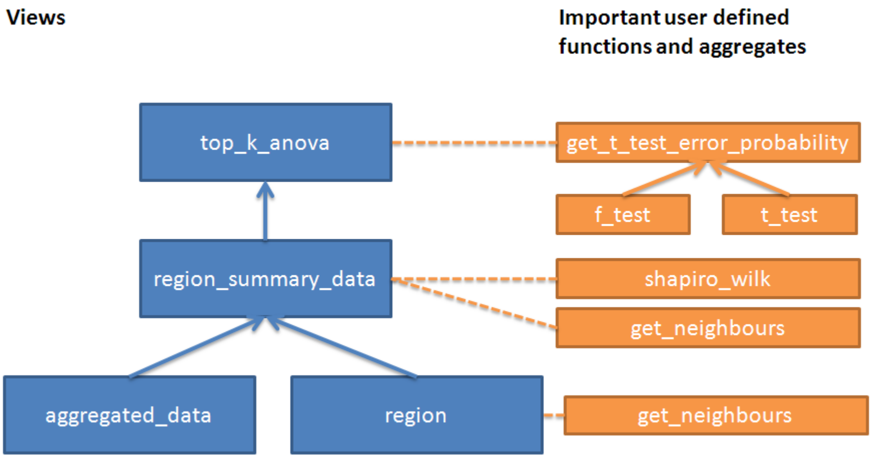


Figure 4.1: Introduced views and important user defined functions and aggregates for top-k similar regions algorithm

### 4.1.1 top_k_similar_regions

The *top_k_similar_regions* view is based on the *region_summary_data* view and uses the *get_similarity_probability* function. It has the following PostgreSQL-Query:

```sql
CREATE VIEW
    top_k_similar_regions

AS SELECT
    neighbours.origin_key,
    neighbours.nutrient_key,
    get_similarity_probability(
        neighbours.count,
        neighbours.avg,
        neighbours.variance,
        selected_region.count,
        selected_region.avg,
        selected_region.variance) AS similarity_probability

FROM
    region_summary_data AS selected_region
        INNER JOIN region_summary_data AS neighbours
        ON selected_region.nutrient_key = neighbours.nutrient_key

WHERE
    selected_region.origin_key = 2094 /* origin_key of user defined region */
    AND neighbours.origin_key <> 2094 /* origin_key of user defined region */
```

```
23      AND selected_region.shapiro_wilk >= 0
24      AND neighbours.shapiro_wilk >= 0
25
26  ORDER BY
27      similarity_probability DESC
```

The query joins two *region_summary_data* views on their *nutrient_key* as seen on lines 16 to 18. This is essential, as for every region comparison two regions, respectively two nutrient measurement sets are needed. From the as *selected_region* denoted *region_summary_data* view, the number of measurements, average and variance of the nutrient measurement set of the user defined region is taken if it is normally distributed for the considered nutrient, as seen on lines 21 and 23. This results in at most one entry per nutrient. In contrast, the as *neighbours* denoted *region_summary_data* view contains all those values for all other regions. At this position we have to remark, that all regions which intersect the user defined region were already dropped in the *region_summary_data* view. This makes it possible to compute the similarity probabilities as seen on lines 7 to 13 instantly. Finally, those similarity probabilities are ordered descending, whereupon the most similar regions for some nutrient are listed on top.

### get_similarity_probability

For calculating the similarity probability, the *get_similarity_probability* function is used. It has the following PostgreSQL-Query:

```
CREATE FUNCTION
    get\_similarity\_probability(
        count double precision,
        avg double precision,
        var double precision,
        _count double precision,
        _avg double precision,
        _var double precision)

RETURNS double precision AS
$$

DECLARE
    t_test_p    double precision;
    f_test_p    double precision;

BEGIN

    f_test_p   := 1 − ftest(var, count, _var, _count);

    IF f_test_p >= 0.05 THEN
        /* equal variance */
        t_test_p   := ttest(count, avg, var, _count, _avg, _var, 1, 2);

    ELSE
        /* unequal variance */
```

```
27          t_test_p  := ttest(count, avg, var, _count, _avg, _var, 1, 3);
28
29      END IF;
30
31      RETURN t_test_p;
32
33  END;
34
35  $$
36  LANGUAGE plpgsql VOLATILE
```

It takes as input the average, variance and the number of measurements of the two nutrient measurement sets being considered when this function is called. We then apply the F-Test and t-Test as explained in Section 3.3 and 3.4 respectively. This means, that we first calculate the probability to which extent the variances of the two nutrient measurement sets can be considered equal as seen on lines 19. Then, depending if the probability is greater or equal than the level of significance of 5% as seen on line 22, the two sample t-Test or the Welch-t-Test behind the function on lines 24 and 28 respectively is applied. We will go into the details of those functions in a moment. Irrespective which t-Test is applied, both functions will return a probability to which extent the two nutrient measurement sets can be considered equal according their averages.

### ftest

This function returns the probability to which extent the two nutrient measurement sets can be considered equal according their variances. It has the following PostgreSQL-Query:

```
CREATE FUNCTION ftest (
    V1  double precision,
    N1  double precision,
    V2  double precision,
    N2  double precision)

RETURNS double precision AS
$$

DECLARE
    p   double precision;

BEGIN
    /* avoid division by zero */
    IF V1 = 0 THEN
        var1 := 0.000001;
    END IF;

    IF V2 = 0 THEN
        var2 := 0.000001;
    END IF;

    IF V1 > V2 THEN
```

```
24           p   := pgnumerics.fcdf (var1/var2, N1-1, N2-1);
25       ELSE
26           p   := pgnumerics.fcdf (var2/var1, N2-1, N1-1);
27       END IF;
28 
29       RETURN p;
30 END;
31 
32 $$
33 LANGUAGE 'plpgsql';
```

It takes as input the variance and the number of measurements of the two nutrient measurement sets being considered when this function is called as seen on lines 2 to 5.

We then assure that the variances are non zero as seen on lines 15 to 21, to avoid division by zero when we calculate the F-Test on lines 24 or 26. If this would be the case, we slightly add some negligible millionth to the variance.

Given non zero variances, we apply the F-Test as described in the equation 8, before we compute the tail of the probability density function of the F-Distribution as seen on lines 24 and 26. For this, we also have to compute the two degree of freedoms, $FGF_1$ and $FGF_2$ as proposed in equation 9 and 10 respectively.

The problem which arises now is that we have to calculate the integral over the tail the probability density function. As this is not possible without further ado, we use the *pgnumerics* library. This library has statistical functions, like the *pgnumerics.fcdf* function. It uses constants for approximating the integral of the probability density function of the F-Distribution.

### ttest

This function returns the probability to which extent the two nutrient measurement sets can be considered equal according their averages. It has the following PostgreSQL-Query:

```
CREATE FUNCTION ttest (
    count1 double precision,
    avg1 double precision,
    var1 double precision,
    count2 double precision,
    avg2 double precision,
    var2 double precision,
    tail double precision,
    T     double precision
) RETURNS double precision

AS $$
DECLARE
    df double precision;
    tx double precision;
    sig double precision;
    pv  double precision;
```

```
18 BEGIN
19     -- two sample equal variance
20     IF T=2 THEN
21         pv := ((count1 -1.0)*var1 + (count2 -1.0)*var2) / (count1+count2 -2.0);
22         tx := (avg1-avg2) / sqrt(pv * (1/count1 + 1/count2));
23         df := (count1+count2 -2.0);
24         sig:= tail * pgnumerics.tcdf(tx, df);
25 
26     -- two sample unequal variance
27     ELSIF T=3 THEN
28         df := (((var1/count1)+(var2/count2)) * ((var1/count1)+(var2/count2)))
               / ((var1*var1)/(count1*count1*(count1 -1.0))+(var2*var2)/(count2*
             count2*(count2 -1.0)));
29         tx := (avg1-avg2) / sqrt(var1/count1 + var2/count2);
30         sig:= tail * pgnumerics.tcdf(tx, df);
31     END IF;
32 
33     RETURN sig;
34 
35 END;
36 
37 $$ LANGUAGE 'plpgsql';
```

This function is based on the *pgnumerics.ttest* function and was slightly modified in order that it takes aggregates like the number of measurements, average and the variance of a nutrient measurement set instead of the whole nutrient measurement set.

It takes as input the average, variance and the number of measurements of the two nutrient measurement sets being considered when this function is called as seen on lines 2 to 7. Furthermore it requests the parameters *tail* and *T* as seen on lines 8 and 9. If *tail* equals 1, twice the tail of the cumulative distribution function of the t-Test is returned, the probability we search. On the other hand, the *T* is a parameter which defines which t-Test must be applied. $T = 2$ stands for the two sample equal variance t-Test as seen on line 20, whereupon $T = 3$ stands for the Welch-t-Test on line 27.

On lines 21 and 22, the T value for the two sample t-Test for equal variance is calculated in an equivalent way as in equation 12. It does not distinguish between $n_1 \neq n_2$ and $n_1 = n_2$ as the formula for $n_1 \neq n_2$ also holds for $n_1 = n_2$. Afterwards, the degree of freedom is calculated on line 23, like in equation 13, before the similarity probability is being calculated on line 24.

In case that the variances of the two nutrient measurement sets cannot be considered equal, the T value is calculated on line 29 like in equation 14. The degree of freedom are calculated on line 28, equivalently to equation 15. Again, the formula for $n_1 \neq n_2$ can be also applied for $n_1 = n_2$.

## 4.2 region_summary_data

The *region_summary_data* view is based on the *region* and *aggregated_data* view and uses the shapiro_wilk-function. It has the following PostgreSQL-Query:

```
 1 CREATE VIEW
 2     region_summary_data
 3
 4 AS SELECT
 5     region.origin_key,
 6     nutrient_key,
 7     count(aggregated_quantity),
 8     avg(aggregated_quantity),
 9     variance(aggregated_quantity),
10     shapiro_wilk(aggregated_quantity)
11
12 FROM
13     region INNER JOIN aggregated_data
14         ON region.neighbour = aggregated_data.origin_key
15
16 WHERE
17     region.origin_key NOT IN
18         (SELECT * FROM get_neighbours(47.20811000, 9.18687000, 20))
19         /* longitude and latitude of user defined region */
20         /* twice the user defined distance / radius */
21     OR region.origin_key = 2094 /* origin_key of user defined region */
22
23 GROUP BY
24     region.origin_key, nutrient_key;
```

The query assembles the corresponding entries from the *region* and the *aggregated_data* view through an INNER JOIN as seen on line 13 and 14. At this step, we still have individual locations with their aggregated measurement quantities. In fact, there is an entry for every two locations which are within the user defined distance, whereupon *region.origin_key* will later become the centre of a region and *region.neighbour* will be the locations within this region. On line 17 and 18, we now drop those locations which would become the centre of a region which are closer than twice the user defined distance. Without this statement a region may intersect the user defined region.

The *get_neighbours* function itself returns a set of origins respectively locations which are within the user defined distance to the commited coordinates. However, the development of this function was not part of this thesis.

As we still need the user defined region in the view, we have to explicitly formulate the *origin_key* of its centre, which is 2094 here as seen on line 21.

Given all this entries, the GROUP BY clause on lines 23 and 24 assembles all neighbour locations (*region.neighbour*) to locations (*region.origin_key*) and forms herewith regions for every nutrient. Given the SELECT statement on lines 4 to 10, this means that the average, variance, the number of entries and the Shapiro-Wilk value of all aggregated measurement which correspond to the same region is being calculated for every region, nutrient pair.

### shapiro_wilk

This aggregate returns the Shapiro-Wilk value if the nutrient measurement is normally distributed. Otherwise, some negative value is returned. It has the following PostgreSQL-Query:

```sql
CREATE AGGREGATE shapiro_wilk(double precision) (
    SFUNC=array_append,
    STYPE=double precision[],
    FINALFUNC=array_shapiro_wilk
);

CREATE FUNCTION array_shapiro_wilk(measurements double precision[])
RETURNS double precision AS
$$

DECLARE
    shapiro_wilk_a double precision[][];
    shapiro_wilk_critical double precision[];
    ordered_measurements double precision[];
    N integer;
    N_half integer;
    i double precision;
    numerator double precision;
    denominator double precision;
    y double precision;
    y_square double precision;
    z integer;
    W double precision;

BEGIN
    SELECT ARRAY(select unnest(measurements) as measurement order by
        measurement asc) INTO ordered_measurements ;

    shapiro_wilk_a = ARRAY[...];

    shapiro_wilk_critical := ARRAY[...];

    N := array_upper(ordered_measurements,1) - array_lower(
        ordered_measurements,1) + 1;
    N_half := floor(N/2);
    numerator = 0;
    y = 0;
    y_square = 0;

    /* to few nutrient measurements in nutrient measurement set */
    IF n < 5 THEN
        RETURN -1;
    /* to many nutrient measurements in nutrient measurement set */
    ELSIF n > 50 THEN
        RETURN -2;
    END IF;
```

```
45
46      FOR i IN 1..N_half LOOP
47          numerator := (shapiro_wilk_a[i][N] * (ordered_measurements[n-i+1] -
                ordered_measurements[i])) + numerator;
48          y := ordered_measurements[n-i+1] + ordered_measurements[i] + y;
49          y_square := pow(ordered_measurements[n-i+1],2) + pow(
                ordered_measurements[i],2) + y_square;
50      END LOOP;
51
52      IF 2 * N_half < N THEN
53          y := ordered_measurements[N_half+1] + y;
54          y_square := pow(ordered_measurements[N_half+1],2) + y_square;
55      END IF;
56
57      denominator := y_square - pow(y,2) / N;
58
59
60      /* normal distribution is given if the denominator zero */
61      IF denominator = 0 THEN
62          RETURN 1;
63      END IF;
64
65      W := pow(numerator,2) / denominator;
66
67      /* nutrient measurements are not normally distributed */
68      IF W < shapiro_wilk_critical[N] THEN
69          RETURN -3;
70      END IF;
71
72      RETURN W;
73
74  END;
75
76  $$
77  LANGUAGE 'plpgsql';
```

As we need individual access on the measurements for calculating the Shapiro-Wilk value, we first build an array containing all aggregated measurement quantities for some nutrient which belong to the same region. Keep in mind, that this will happen due to the construction of our SELECT and GROUP BY clause of our *region_summary_data* view. After the array is built on line 4, the *array_shapiro_wilk* function is executed. We then apply the Shapiro-Wilk-Test as explained in Section with reserve of the constrains above. In case that a region contains less than five measurements for some nutrient, we let the Shapiro-Wilk-Test fail on lines 39 and 40 irrespective what the result would actually be, due to error-proneness. Furthermore, as no $a_{(n_i,k')}$ Shapiro-Wilk constants are available for nutrient measurement sets with more than fifty measurements, also those regions are excluded for the considered nutrient as seen on lines 42 and 43.

From line 46 on we calculate the Shapiro-Wilk value as explained in 6. In case of an uneven number of measurements in a nutrient measurement set, the addition on lines 51 to 55 ensures that the denominator gets correctly calculated, as the median is not considered in the numerator.

On line 61 we assure that the denominator is non zero. If this would not be the case, the variance of the nutrient measurement set would be zero, implying that it is normally distributed anyway.

Finally, on line 65, the $W_i$ value is computed, before it is compared against the critical $W_{(n_i,0.05)}$ value on line 68 like in equation 7. If the nutrient measurement set can be considered normally distributed, the Shapiro-Wilk value is returned, otherwise a negative integer.

## 4.3 region

The *region* view is based on the *d_origin* table and uses the *get_neighbours* function. It has the following PostgreSQL-Query:

```
1 CREATE VIEW region AS
2
3 SELECT
4     origin_key, get_neighbours(latitude, longitude, 10) AS neighbour
5
6 FROM
7     d_origin
```

This view aggregates to every location (*origin_key*) a set of locations containing all locations which are within a user defined distance to the corresponding location. For this, the function *get_neighbours* is used as seen on line 4 which takes as parameter the coordinates of the corresponding location and the selectable radius of a region.

## 4.4 aggregated_data

The aggregated_data view is based on the *fact_table*, *d_time*, *d_nutrient*, *d_origin* and *d_feed* table. It has the following PostgreSQL-Query:

```
CREATE VIEW
    aggregated_data AS

SELECT
    nutrient_key, origin_key, avg(quantity) AS aggregated_quantity

FROM
    fact_table INNER JOIN d_time ON id_time_fkey = time_key
               INNER JOIN d_nutrient ON id_nutrient_fkey = nutrient_key
               INNER JOIN d_origin ON id_origin_fkey = origin_key
               INNER JOIN d_feed ON id_feed_fkey = feed_key

```

```
13 WHERE
14     /* user selection */
15     z_abbreviation_de = 'Ca'
16
17 GROUP BY
18     lims_number, nutrient_key, origin_key;
```

This view connects for every measurement the corresponding entries in the time, nutrient, origin and feed table together. Those measurements which do not fulfil the user selection, here the nutrient must be calcium as seen on line 15, are then dropped. Furthermore, this view aggregates multiple measurements of one nutrient from one feed sample together. This can be seen on lines 5, 17 and 18 with the aggregate with calculates the average over the feed samples.

# 5 Implementation

This section describes optimizations made when the top k similar regions algorithm was elaborated.

The goal was to minimize the execution time, by minimizing the number of tuples in each view. For this, the statistics used to calculate the similarity probabilities was analysed in detail, to investigate whether a region could be summarized with few aggregates instead of storing the whole nutrient measurement set. It reveals, that the t-Test and its probability $p_F$ can be computed given the variance, average and the number of measurements in each nutrient measurement set $X_i$. Further, the F-Test can be computed given those aggregates as well, it does not even depend on the average of a nutrient measurement set. As those aggregates are easily to be retrieved in PostgreSQL, the numerator of the Shapiro-Wilk-Test is more complex as it depends on different $a_{(n_i,k')}$ Shapiro-Wilk constants and the ascending ordered nutrient measurement set. However, defining a user defined aggregate with a final function as explained in Section 4.2, solves also this problem. Moreover, it does not only calculate the Shapiro-Wilk value, but verifies also if the nutrient measurement set is normally distributed and if it has the right size. If not, a negative value indicates this, whereupon this region, nutrient pair will be dropped in the following *top_k_similar_regions* view.

Given the fact that it is sufficient to have those four aggregates per region, we now have only one entry in the *region_summary_data* view per region, nutrient pair instead of several tuples each containing a single measurement. The difference can be seen between Tables 3 and 4.

Table 3: Optimized *region_summary_data* view

| **id** | **origin_key** | **nutrient_key** | **count** | **avg** | **variance** | **shapiro_wilk** |
|---|---|---|---|---|---|---|
| 1 | 1040 | 283 | 8 | 147.45 | 2454.77 | 0.852 |

Table 4: Not optimized *region_summary_data* view

| **id** | **origin_key** | **nutrient_key** | **aggregated_quantity** |
|---|---|---|---|
| 1 | 1040 | 283 | 97.89 |
| 2 | 1040 | 283 | 120.27 |
| 3 | 1040 | 283 | 188.66 |
| 4 | 1040 | 283 | 102.08 |
| 5 | 1040 | 283 | 207.53 |
| 6 | 1040 | 283 | 137.01 |
| 7 | 1040 | 283 | 217.97 |
| 8 | 1040 | 283 | 108.20 |

Another point was whether to first assemble all measurements to every tuple in the *aggregated_data* view which belongs to the same region, or to list for every location a set of neighbour locations. An advantage of assembling all measurements which belong to the same region to every tuple in *aggregated_data* view is that in case the user has a hard selection criteria involving measurements only from some locations, that no unnecessary neighbours would be searched for regions which will be dropped later on. In contrast, if two or more measurements from the same nutrient are from the same location, the assembling would be done more than once, creating unnecessary duplicates. Assuming regular search request, usually all or nearly all of the 1510 different locations will remain in the result, leading to the fact that it is faster to obtain the regions, respectively the set of locations to every location first. Afterwards, the quantities in the *aggregated_data* view are assigned to their corresponding region, nutrient pairs as described in Section 4.2.

# 6 Evaluation

This section describes the evaluation of the top-k similar regions algorithm.

Given the Swiss Feed Database, we evaluate the performance of our algorithm depending on the number of:

- measurements
- distinct nutrients
- distinct regions

Furthermore we investigate the query plan.

## 6.1 Impact on the number of measurements

Given the Swiss Feed Database with its nearly four million nutrient measurements, 87'702 measurements were taken on the nutrient calcium. The following diagram shows the execution time in dependence of the number of all measurements over all nutrients.

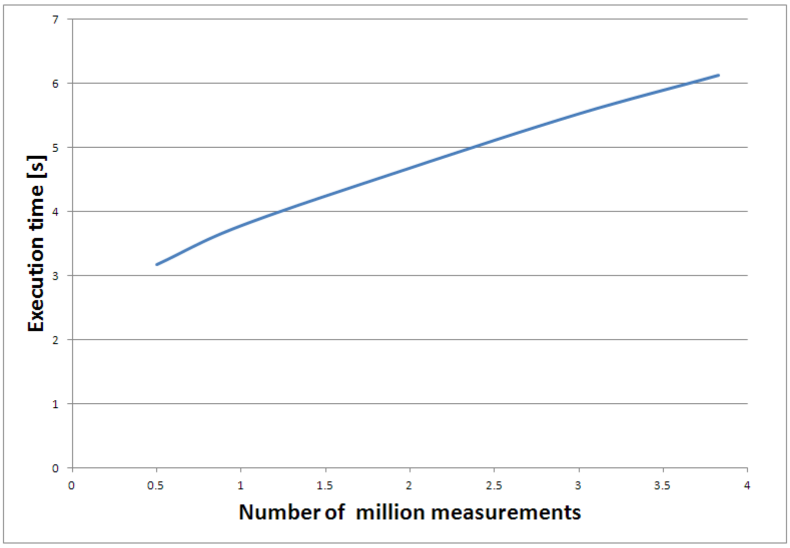


Figure 6.1: Execution time in dependence of the number of measurements over all nutrients

According Figure 6.1, we can clearly see that the execution time increases approximately linear in the number of measurements over all nutrients. However, in case the user defined region does not pass the Shapiro-Wilk-Test, or if the region has to few or many measurements, the execution time is only about 156 ms. The user can encounter to small or large regions by adjusting the radius of the region. This may also help in case the region is not normally distributed.

## 6.2 Impact on the number of distinct nutrients

The top-k similar regions algorithm delivers for every region, nutrient pair a similarity probability to the user defined region. This implies that a region might have different similarity probabilities on different nutrients to the user defined region. Depending on the number of measured nutrients the user selects in the user defined region, the execution time on nearly four million measurements is as follows.

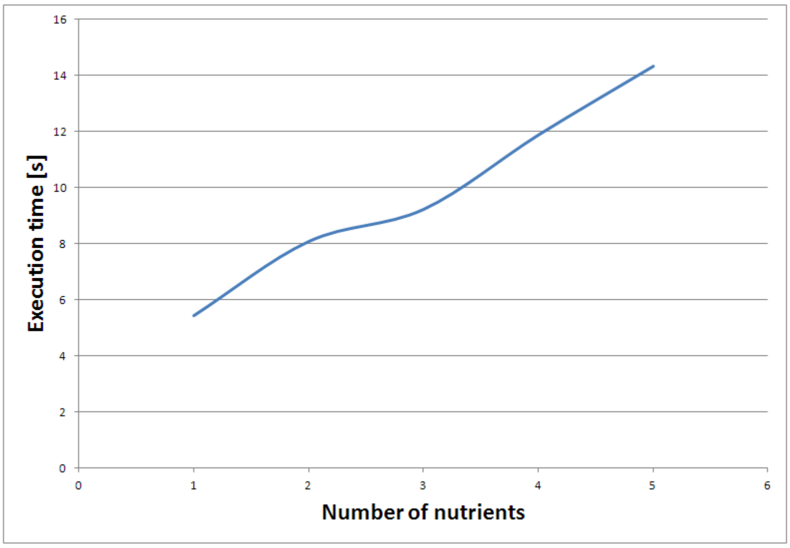


Figure 6.2: Execution time in dependence of the number of distinct nutrients measured in the user defined region

According Figure 6.2, we can make out a slight S-curve. This can be explained with the fact, that for some nutrients many measurements were taken whereas for others nearly none. In this case here, only twenty thousand measurements were taken on the third nutrient, whereas sixty to ninety thousand measurements were taken on the other four nutrients.

## 6.3 Impact on the number of distinct regions

As discussed in the subsection before, we get a similarity probability for every nutrient, region pair to the user defined region. Assuming that the user only selects the nutrient calcium, the execution time in dependence of the number of distinct regions is as follows:

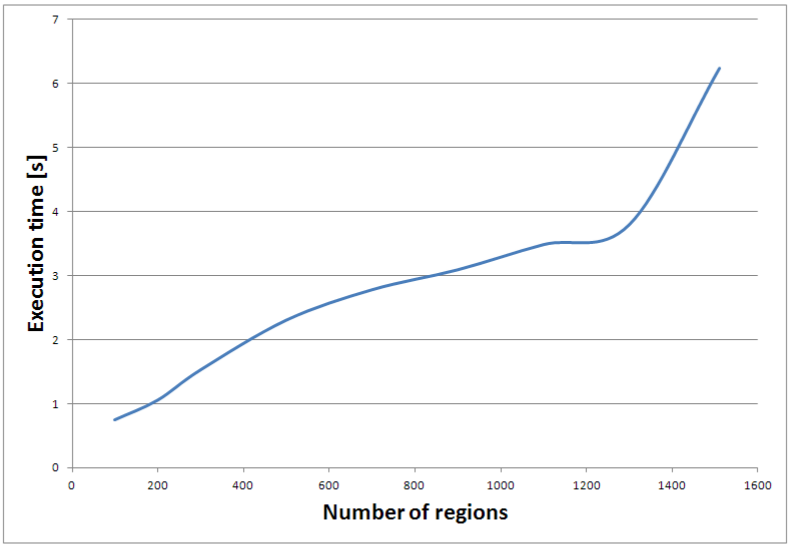


Figure 6.3: Execution time in dependence of the number of distinct regions for nutrient calcium

According Figure 6.3, we can make out a linear increase in execution time in dependence of the number of the number of distinct regions up to 1300 regions. For more than 1300 regions, the execution time seems exploding, but this is only as most measurements are part of the last two hundred regions in this example. The outcome is not unexpected, as the region view lists for every location its corresponding neighbours, before each entry is assembled with the set of nutrient measurements of the *aggregated_data* view in the *region_summary_data* view. Therefore, the number of regions have a direct and nearly a linear impact on the number of entries in the *region_summary_data* view if all measurements would be uniformly distributed over all regions.

## 6.4 Query execution plan

The query execution plan for finding the top-k similar regions can be seen in Figure 6.4. As already discussed in Section 4, the algorithm is based on four views, whereas the *top_k_similar_regions* view is red bordered and based on the composition of two *region_summary_data* views, which itself are orange bordered and depend on the *aggregated_data* and *region* view. Both green shaded areas shows the composition of the *aggregated_data* view, whereas the blue shaded areas displays the composition of the *region* view.

The complexity of computing the views depends on the number of nutrients the user selected. This can be seen in Table 5.

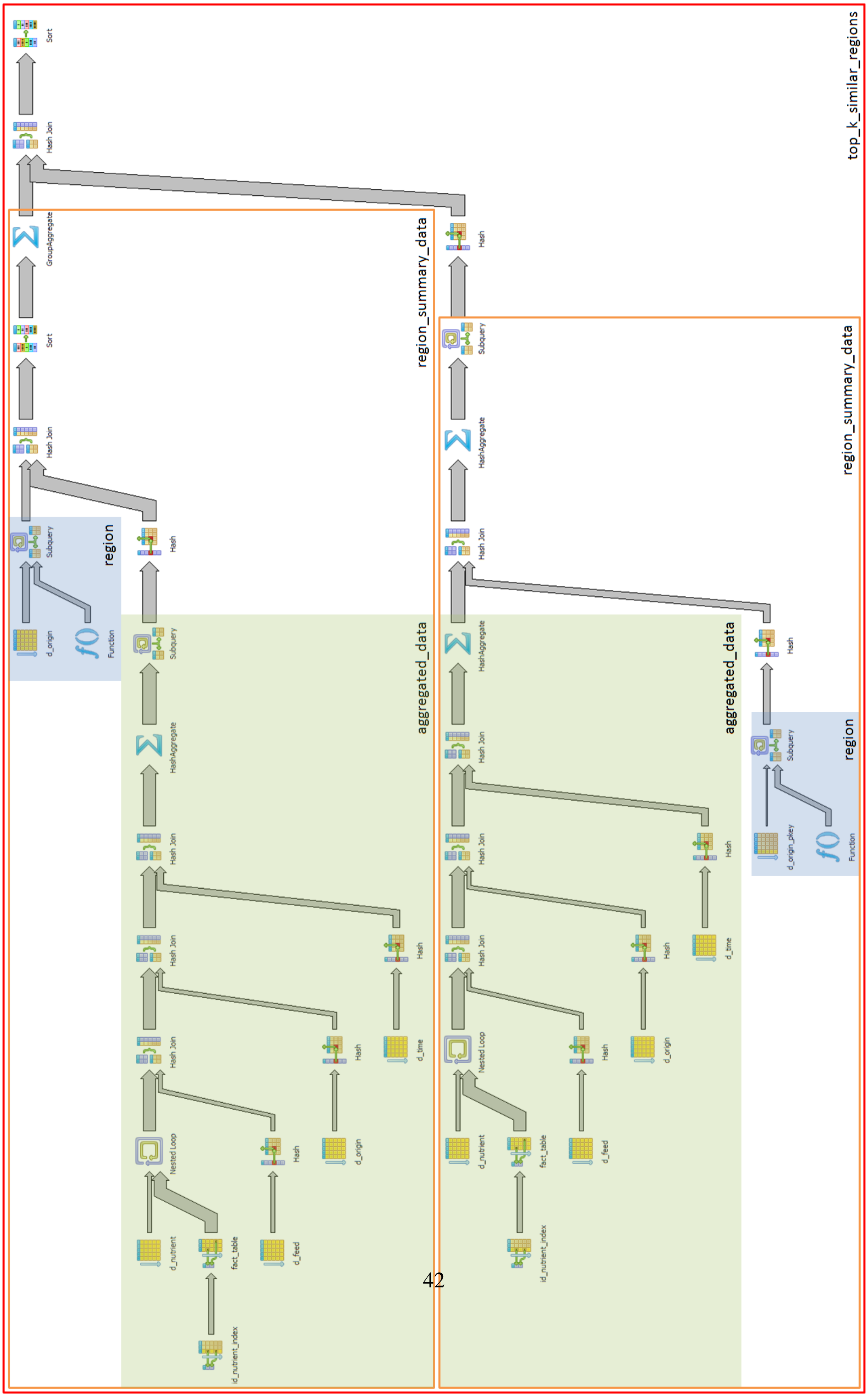


Figure 6.4: Query execution plan for finding the top-k similar regions for the nutrient calcium

Table 5: Execution time for different views

| Nutrient<br>View | Calcium | All |
|---|---|---|
| **aggregated_data** | 3.93% | 20.83% |
| **region** | 30.67% | 1.21% |
| **region_summary_data (neighbours)** | 49.06% | 48.50% |
| **region_summary_data (user defined region)** | 4.20% | 11.99% |
| **top_k_similar_regions** | 12.14% | 17.48% |

If the top-k similar regions for the nutrient calcium shall be found, it is hardest to compute the *region_summary_data* view containing all the entries of the other regions. It takes about 12.5 times as long as the computation of the *aggregated_data* view. This might be due to the complex Shapiro-Wilk aggregate beside the fact that the same *get_neighbours* function is used here like in the region view to drop regions intersecting the user defined region. On the other hand, the *region_summary_data* view is quite as fast as the *aggregated_data view*, as it lists only the entry for the user defined region. A further complex view is the *region_view* which computes all the neighbour locations to a location. The query needs 6.8 times longer to build the view in comparison to the *aggregated_data* view.

In case all nutrients were selected, the *region_summary_data* view containing all the entries of the other regions is still the slowest, but the *region* view is now the fastest as its number of entries remain unchanged. On the other hand, there are now more entries in the *aggregated_data* view, making it more complex, which results in the fact that it already needs half as long as the *region_summary_data* view. Furthermore, also the *region_summary_data* view containing all the entries of the user defined region is now much slower, as it has to list an entry for every value. Fortunately, the complexity of the *top_k_similar_regions* view is now about 1.2 times faster than the *aggregated_data* view.

We have to acknowledge that some views, like the *aggregated_data* and *region* view can be computed in parallel. If those two views are at least partially computed, also the two *region_summary_data* views, which depend on this two views, can be computed in parallel, reducing the total execution time.

## 6.5 Top k (dis)similar regions

Given the nutrient calcium with Nesslau as the centre of the user defined region, the most similar region is around Vignon near Illanz in Grisons as seen in Figure 6.5. The two regions can be considered 99.9% similar. The next eight most similar regions are all within the same area around Illanz. This is not very surprising, as most of their measurements are part of some nearby region as well. With 97.9% the next completely different region around Grandvillard follows up.

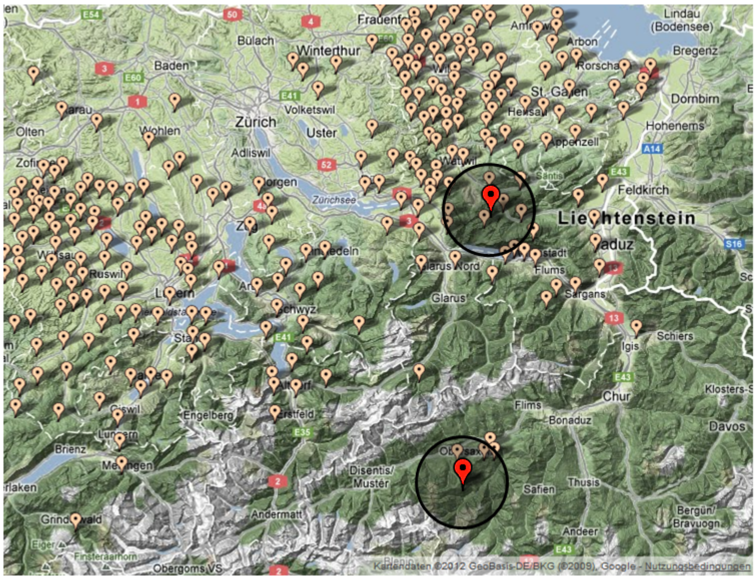


Figure 6.5: Most similar region to Nesslau for nutrient calcium

On the other hand, the most dissimilar region can be found around Rohrbach near Burgdorf as seen in Figure 6.6. Its similarity probability is nearly 0.0%. Moreover, nearby regions are as dissimilar as well, as most measurements are similar in both regions.

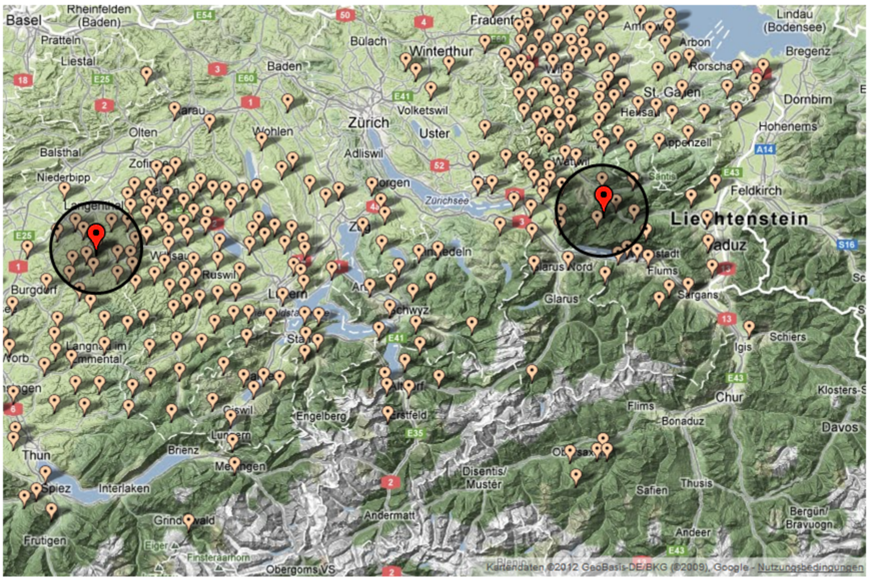


Figure 6.6: Most dissimilar region to Nesslau for nutrient calcium

# 7 Conclusions and Future Work

The developed algorithm for finding the top-k similar regions, contributes to the work of Agroscope. Researches are now able to identify similar regions, which allows them to make further studies. They can now make decision whether measurements shall be further taken from some region or if they can analyse new regions due to similarities. Furthermore, researchers can compare two regions individually on their similarity probability for some nutrient.

The evaluation revealed that the computation of the top-k similar regions can be done in reasonable time for few nutrients. Furthermore, there indeed exist very similar and dissimilar regions, at least for the considered example. Moreover, the execution time increases linearly in the number of measurements, distinct nutrients and regions if the measurements are uniformly distributed among them. However, the algorithm returns a similarity probability for every region, nutrient pair to the user defined region. The use of a multivariate approach would encounter this problem and give more meaningful results about which regions are most similar and to which degree irrespective of the nutrient. Furthermore, a work around for nutrient measurement sets which are not normally distributed should be invented using other statistical tests.

# References

[1] Google Maps. (2012). [Switzerland] [Terrain]. Retrieved August 15, 2012, from http://maps.google.ch

[2] Lozán, J. L. & Kausch, H. (2007). *Angewandte Statistik für Naturwissenschaftler* (4th ed.). Hamburg: Wissenschaftliche Auswertungen.

[3] NIST/SEMATECH. (2012). *e-Handbook of Statistical Methods*. Retrieved August 15, 2012, from http://www.itl.nist.gov/div898/handbook/

[4] Pearson, E. S. & Hartley, H. O. (1972). *Biometrika tables for statisticians: Volume II*. Cambridge: The University Press.

[5] The PostgreSQL Global Development Group. (2012). *PostgreSQL*. Retrieved August 15, 2012, from http://www.postgresql.org/

# Appendix 1

According [4] the $a_{(n_i,k')}$ Shapiro-Wilk constants are as follows:

| n<br>i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.0000 | 0.7071 | 0.7071 | 0.6872 | 0.6646 | 0.6431 | 0.6233 | 0.6052 | 0.5888 | 0.5739 |
| **2** | - | - | 0.0000 | 0.1667 | 0.2413 | 0.2806 | 0.3031 | 0.3164 | 0.3244 | 0.3291 |
| **3** | - | - | - | - | 0.0000 | 0.0875 | 0.1401 | 0.1743 | 0.1976 | 0.2141 |
| **4** | - | - | - | - | - | - | 0.0000 | 0.0561 | 0.0947 | 0.1224 |
| **5** | - | - | - | - | - | - | - | - | 0.0000 | 0.0399 |

| n<br>i | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.5601 | 0.5475 | 0.5359 | 0.5251 | 0.5150 | 0.5056 | 0.4968 | 0.4886 | 0.4808 | 0.4734 |
| **2** | 0.3315 | 0.3325 | 0.3325 | 0.3318 | 0.3306 | 0.3290 | 0.3273 | 0.3253 | 0.3232 | 0.3211 |
| **3** | 0.2260 | 0.2347 | 0.2412 | 0.2460 | 0.2495 | 0.2521 | 0.2540 | 0.2553 | 0.2561 | 0.2565 |
| **4** | 0.1429 | 0.1586 | 0.1707 | 0.1802 | 0.1878 | 0.1939 | 0.1988 | 0.2027 | 0.2059 | 0.2085 |
| **5** | 0.0695 | 0.0922 | 0.1099 | 0.1240 | 0.1353 | 0.1447 | 0.1524 | 0.1587 | 0.1641 | 0.1686 |
| **6** | 0.0000 | 0.0303 | 0.0539 | 0.0727 | 0.0880 | 0.1005 | 0.1109 | 0.1197 | 0.1271 | 0.1334 |
| **7** | - | - | 0.0000 | 0.0240 | 0.0433 | 0.0593 | 0.0725 | 0.0837 | 0.0932 | 0.1013 |
| **8** | - | - | - | - | 0.0000 | 0.0196 | 0.0359 | 0.0496 | 0.0612 | 0.0711 |
| **9** | - | - | - | - | - | -0 | 0.0000 | 0.0163 | 0.0303 | 0.0422 |
| **10** | - | - | - | - | - | - | - | - | 0.0000 | 0.0140 |

| n<br>i | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.4643 | 0.4590 | 0.4542 | 0.4493 | 0.4450 | 0.4407 | 0.4366 | 0.4328 | 0.4291 | 0.4254 |
| **2** | 0.3185 | 0.3156 | 0.3126 | 0.3098 | 0.3069 | 0.3043 | 0.3018 | 0.2992 | 0.2968 | 0.2944 |
| **3** | 0.2578 | 0.2571 | 0.2563 | 0.2554 | 0.2543 | 0.2533 | 0.2522 | 0.2510 | 0.2499 | 0.2487 |
| **4** | 0.2119 | 0.2131 | 0.2139 | 0.2145 | 0.2148 | 0.2151 | 0.2152 | 0.2151 | 0.2150 | 0.2148 |
| **5** | 0.1736 | 0.1764 | 0.1787 | 0.1807 | 0.1822 | 0.1836 | 0.1848 | 0.1857 | 0.1864 | 0.1870 |
| **6** | 0.1399 | 0.1443 | 0.1480 | 0.1512 | 0.1539 | 0.1563 | 0.1584 | 0.1601 | 0.1616 | 0.1630 |
| **7** | 0.1092 | 0.1150 | 0.1201 | 0.1245 | 0.1283 | 0.1316 | 0.1346 | 0.1372 | 0.1395 | 0.1415 |
| **8** | 0.0804 | 0.0878 | 0.0941 | 0.0997 | 0.1046 | 0.1089 | 0.1128 | 0.1162 | 0.1192 | 0.1219 |
| **9** | 0.0530 | 0.0618 | 0.0696 | 0.0764 | 0.0823 | 0.0876 | 0.0923 | 0.0965 | 0.1002 | 0.1036 |
| **10** | 0.0263 | 0.0368 | 0.0459 | 0.0539 | 0.0610 | 0.0672 | 0.0728 | 0.0778 | 0.0822 | 0.0862 |
| **11** | 0.0000 | 0.0122 | 0.0228 | 0.0321 | 0.0403 | 0.0476 | 0.0540 | 0.0598 | 0.0650 | 0.0697 |
| **12** | - | - | 0.0000 | 0.0107 | 0.0200 | 0.0284 | 0.0358 | 0.0424 | 0.0483 | 0.0537 |
| **13** | - | - | - | - | 0.0000 | 0.0094 | 0.0178 | 0.0253 | 0.0320 | 0.0381 |
| **14** | - | - | - | - | - | - | 0.0000 | 0.0084 | 0.0159 | 0.0227 |
| **15** | - | - | - | - | - | - | - | - | 0.0000 | 0.0076 |

| n<br>i | **31** | **32** | **3** | **34** | **35** | **36** | **37** | **38** | **39** | **40** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.4220 | 0.4188 | 0.4156 | 0.4127 | 0.4096 | 0.4068 | 0.4040 | 0.4015 | 0.3989 | 0.3964 |
| **2** | 0.2921 | 0.2898 | 0.2876 | 0.2854 | 0.2834 | 0.2813 | 0.2794 | 0.2774 | 0.2755 | 0.2737 |
| **3** | 0.2475 | 0.2463 | 0.2451 | 0.2439 | 0.2427 | 0.2415 | 0.2403 | 0.2391 | 0.2380 | 0.2368 |
| **4** | 0.2145 | 0.2141 | 0.2137 | 0.2132 | 0.2127 | 0.2121 | 0.2116 | 0.2110 | 0.2104 | 0.2098 |
| **5** | 0.1874 | 0.1878 | 0.1880 | 0.1882 | 0.1883 | 0.1883 | 0.1883 | 0.1881 | 0.1880 | 0.1878 |
| **6** | 0.1641 | 0.1651 | 0.1660 | 0.1667 | 0.1673 | 0.1678 | 0.1683 | 0.1686 | 0.1689 | 0.1691 |
| **7** | 0.1433 | 0.1449 | 0.1463 | 0.1475 | 0.1487 | 0.1496 | 0.1505 | 0.1513 | 0.1520 | 0.1526 |
| **8** | 0.1243 | 0.1265 | 0.1284 | 0.1301 | 0.1317 | 0.1331 | 0.1344 | 0.1356 | 0.1366 | 0.1376 |
| **9** | 0.1066 | 0.1093 | 0.1118 | 0.1140 | 0.1160 | 0.1179 | 0.1196 | 0.1211 | 0.1225 | 0.1237 |
| **10** | 0.0899 | 0.0931 | 0.0961 | 0.0988 | 0.1013 | 0.1036 | 0.1056 | 0.1075 | 0.1092 | 0.1108 |
| **11** | 0.0739 | 0.0777 | 0.0812 | 0.0844 | 0.0873 | 0.0900 | 0.0924 | 0.0947 | 0.0967 | 0.0986 |
| **12** | 0.0585 | 0.0629 | 0.0669 | 0.0706 | 0.0739 | 0.0770 | 0.0798 | 0.0824 | 0.0848 | 0.0870 |
| **13** | 0.0435 | 0.0485 | 0.0530 | 0.0572 | 0.0610 | 0.0645 | 0.0677 | 0.0706 | 0.0733 | 0.0759 |
| **14** | 0.0289 | 0.0344 | 0.0395 | 0.0441 | 0.0484 | 0.0523 | 0.0559 | 0.0592 | 0.0622 | 0.0651 |
| **15** | 0.0144 | 0.0206 | 0.0262 | 0.0314 | 0.0361 | 0.0404 | 0.0444 | 0.0481 | 0.0515 | 0.0546 |
| **16** | 0.0000 | 0.0068 | 0.0131 | 0.0187 | 0.0239 | 0.0287 | 0.0331 | 0.0372 | 0.0409 | 0.0444 |
| **17** | - | - | 0.0000 | 0.0062 | 0.0119 | 0.0172 | 0.0220 | 0.0264 | 0.0305 | 0.0343 |
| **18** | - | - | - | - | 0.0000 | 0.0057 | 0.0110 | 0.0158 | 0.0203 | 0.0244 |
| **19** | - | - | - | - | - | - | 0.0000 | 0.0053 | 0.0101 | 0.0146 |
| **20** | - | - | - | - | - | - | - | - | 0.0000 | 0.0049 |

| n<br>i | **41** | **42** | **43** | **44** | **45** | **46** | **47** | **48** | **49** | **50** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0.3940 | 0.3917 | 0.3894 | 0.3872 | 0.3850 | 0.3830 | 0.3808 | 0.3789 | 0.3770 | 0.3751 |
| **2** | 0.2719 | 0.2701 | 0.2684 | 0.2667 | 0.2651 | 0.2635 | 0.2620 | 0.2604 | 0.2589 | 0.2574 |
| **3** | 0.2357 | 0.2345 | 0.2334 | 0.2323 | 0.2313 | 0.2302 | 0.2291 | 0.2281 | 0.2271 | 0.2260 |
| **4** | 0.2091 | 0.2085 | 0.2078 | 0.2072 | 0.2065 | 0.2058 | 0.2052 | 0.2045 | 0.2038 | 0.2032 |
| **5** | 0.1876 | 0.1874 | 0.1871 | 0.1868 | 0.1865 | 0.1862 | 0.1859 | 0.1855 | 0.1851 | 0.1847 |
| **6** | 0.1693 | 0.1694 | 0.1695 | 0.1695 | 0.1695 | 0.1695 | 0.1695 | 0.1693 | 0.1692 | 0.1691 |
| **7** | 0.1531 | 0.1535 | 0.1539 | 0.1542 | 0.1545 | 0.1548 | 0.1550 | 0.1551 | 0.1553 | 0.1554 |
| **8** | 0.1384 | 0.1392 | 0.1398 | 0.1405 | 0.1410 | 0.1415 | 0.1420 | 0.1423 | 0.1427 | 0.1430 |
| **9** | 0.1249 | 0.1259 | 0.1269 | 0.1278 | 0.1286 | 0.1293 | 0.1300 | 0.1306 | 0.1312 | 0.1317 |
| **10** | 0.1123 | 0.1136 | 0.1149 | 0.1160 | 0.1170 | 0.1180 | 0.1189 | 0.1197 | 0.1205 | 0.1212 |
| **11** | 0.1004 | 0.1020 | 0.1035 | 0.1049 | 0.1062 | 0.1073 | 0.1085 | 0.1095 | 0.1105 | 0.1113 |
| **12** | 0.0891 | 0.0909 | 0.0927 | 0.0943 | 0.0959 | 0.0972 | 0.0986 | 0.0998 | 0.1010 | 0.1020 |
| **13** | 0.0782 | 0.0804 | 0.0824 | 0.0842 | 0.0860 | 0.0876 | 0.0892 | 0.0906 | 0.0919 | 0.0932 |
| **14** | 0.0677 | 0.0701 | 0.0724 | 0.0745 | 0.0765 | 0.0783 | 0.0801 | 0.0817 | 0.0832 | 0.0846 |
| **15** | 0.0575 | 0.0602 | 0.0628 | 0.0651 | 0.0673 | 0.0694 | 0.0713 | 0.0731 | 0.0748 | 0.0764 |
| **16** | 0.0476 | 0.0506 | 0.0534 | 0.0560 | 0.0584 | 0.0607 | 0.0628 | 0.0648 | 0.0667 | 0.0685 |
| **17** | 0.0379 | 0.0411 | 0.0442 | 0.0471 | 0.0497 | 0.0522 | 0.0546 | 0.0568 | 0.0588 | 0.0608 |
| **18** | 0.0283 | 0.0318 | 0.0352 | 0.0383 | 0.0412 | 0.0439 | 0.0465 | 0.0489 | 0.0511 | 0.0532 |
| **19** | 0.0188 | 0.0227 | 0.0263 | 0.0296 | 0.0328 | 0.0357 | 0.0385 | 0.0411 | 0.0436 | 0.0459 |
| **20** | 0.0094 | 0.0136 | 0.0175 | 0.0211 | 0.0245 | 0.0277 | 0.0307 | 0.0335 | 0.0361 | 0.0386 |
| **21** | 0.0000 | 0.0045 | 0.0087 | 0.0126 | 0.0163 | 0.0197 | 0.0229 | 0.0259 | 0.0288 | 0.0314 |
| **22** | - | - | 0.0000 | 0.0042 | 0.0081 | 0.0118 | 0.0153 | 0.0185 | 0.0215 | 0.0244 |
| **23** | - | - | - | - | 0.0000 | 0.0039 | 0.0076 | 0.0111 | 0.0143 | 0.0174 |
| **24** | - | - | - | - | - | - | 0.0000 | 0.0037 | 0.0071 | 0.0104 |
| **25** | - | - | - | - | - | - | - | - | 0.0000 | 0.0035 |

# Appendix 2

According [4] the critical Shapiro-Wilk values $W_{(n_i,0.05)}$ are as follows:

| level of significance<br>n | 5% |
|---|---|
| 3 | 0.767 |
| 4 | 0.748 |
| 5 | 0.762 |
| 6 | 0.788 |
| 7 | 0.803 |
| 8 | 0.818 |
| 9 | 0.829 |
| 10 | 0.842 |
| 11 | 0.850 |
| 12 | 0.859 |
| 13 | 0.866 |
| 14 | 0.874 |
| 15 | 0.881 |
| 16 | 0.887 |
| 17 | 0.892 |
| 18 | 0.897 |
| 19 | 0.901 |
| 20 | 0.905 |
| 21 | 0.908 |
| 22 | 0.911 |
| 23 | 0.914 |
| 24 | 0.916 |
| 25 | 0.918 |
| 26 | 0.920 |
| 27 | 0.923 |
| 28 | 0.924 |
| 29 | 0.926 |
| 30 | 0.927 |
| 31 | 0.939 |
| 32 | 0.930 |
| 33 | 0.931 |
| 34 | 0.933 |
| 35 | 0.934 |
| 36 | 0.935 |
| 37 | 0.936 |
| 38 | 0.938 |
| 39 | 0.939 |
| 40 | 0.940 |
| 41 | 0.941 |
| 42 | 0.942 |
| 43 | 0.943 |
| 44 | 0.944 |
| 45 | 0.945 |
| 46 | 0.945 |
| 47 | 0.946 |
| 48 | 0.947 |
| 49 | 0.947 |
| 50 | 0.947 |