

Handbuch zum Buddy Programm Matching Computerprogramm

erstellt für das International Office von Lisa Adams

Aktualisierte Version von Januar 2020

Inhaltsverzeichnis

1	Übersicht	2
2	Ausführen des Matchings	2
2.1	Download der Antworten des Google Formulars, Anpassen der Tabellen	2
2.2	Matching	2
2.3	Änderungen am Matching	3
3	Änderungen am Google Formular	3
3.1	Änderung der Reihenfolge der Fragen	3
3.2	Hinzufügen eines Studienfachs	4
3.3	Hinzufügen eines Hobbies	4
4	Änderung der Gewichtung der Matchingkriterien	5
5	Änderung am Output des Matching Programms	6
5.1	Änderung an ausführlicher Übersicht	6
5.2	Änderung an Kurzübersicht	6

1 Übersicht

Das folgende Handbuch soll einen Überblick über das automatisierte Matching der Buddys des Tübinger Buddy Programms geben. Zuerst wird beschrieben, wie das Matching ausgeführt wird. Anschließend werden mögliche Änderungen am Programm angesprochen.

2 Ausführen des Matchings

2.1 Download der Antworten des Google Formulars, Anpassen der Tabellen

Zuerst müssen die Antworten der Buddys als .csv Datei heruntergeladen werden. Dazu wird jeweils für die Tübinger und die Incomings das entsprechende Formular geöffnet, dann wird unter **Antworten** der grüne Button **Antworten in Google Tabellen ansehen** ausgewählt, und dort dann **Datei, Herunterladen als, Microsoft Excel (.xlsx)**. Diese finden sich anschließend im **Downloads** Ordner wieder. Nun kann die Tabelle nach Belieben angepasst werden, wobei jedoch die Spaltenzahl und die Spaltenreihenfolge gleich bleiben müssen.

Ist die Tabelle angepasst, so müssen zuerst alle Umlaute mittels der Funktion **Suchen und Ersetzen** in beiden Tabellen wie folgt ausgetauscht werden: aus Ö und ö wird oe, aus Ü und ü ue, und aus Ä und ä ae. Dann werden die Tabellen jeweils als .csv Datei gespeichert. Diese müssen dann jeweils im **TextEditor** geöffnet werden, und in der ersten Zeile muss am Ende ein Semikolon eingefügt werden. Für Schritt 2.2 brauchen wir nun die Dateipfade der beiden csv-Tabellen. (Es kann zum Beispiel mit gehaltener Shift-Taste ein Rechtsklick auf der Datei ausgeführt werden, und so die Option **Als Pfad kopieren** ausgewählt werden.)

2.2 Matching

Für das Matching wird nun R geöffnet. Jetzt müssen erforderliche sogenannte Packages geladen werden. (Falls diese noch nicht installiert sind, müssen sie unter demselben Menüpunkt zuerst installiert werden (**Pakete, Installieren**; dabei kann ein beliebiger CRAN Spiegel ausgewählt werden). Die benötigten Packages heißen **Rcpp** und **matchingR**. Diese werden geladen, indem im Menüpunkt **Pakete** auf **Pakete laden** geklickt wird, und dann die entsprechenden Pakete aus der angezeigten Liste ausgewählt werden.

Dann wird das Programm wie folgt ausgeführt: Zuerst wird es geladen, wofür den Dateipfad des Programmes **matching_algorithm.R** brauchen (der Dateipfad kann dabei kopiert werden, indem mit gehaltener Shift-Taste ein Rechtsklick auf der Datei im Explorer ausgeführt wird, und dann die Option **Als Pfad kopieren** ausgewählt wird.). **Dabei müssen im Folgenden alle \im Dateipfad doppelt geschrieben werden** (statt Ordner\matching_algorithm.R

also Ordner `\\matching_algorithm.R`). Damit wird dann folgender Command in der Kommandozeile eingegeben:

```
source('Dateipfad zu matching_algorithm.R')
```

Dann wird das Programm mit den beiden in Schritt 2.1 heruntergeladenen Tabellen ausgeführt, wobei wir auch hier wieder die Dateipfade der beiden Tabellen brauchen, auch wieder mit doppelten `\\`(siehe oben). Folgendes wird dann in R in der Kommandozeile eingegeben:

```
matching_programm("Dateipfad Tabelle Incomings", "Dateipfad Tabelle Tuebingen")
```

Dabei müssen die Dateipfade in Anführungszeichen stehen, und die **Tabelle der Incomings muss unbedingt als erstes angegeben werden!**

Das Programm erstellt dann zwei Tabellen, eine etwas übersichtlichere Kurzübersicht mit nur den wichtigsten Informationen der gematchten Buddys (`Kurzübersicht.csv`), und eine ausführliche mit allen Angaben der Buddys (`ausführliche_Übersicht.csv`).

Dabei werden jeweils in jeder Zeile zuerst die Informationen der Gruppe angegeben, die größer ist, d.h. je nachdem ob mehr Tübinger oder Incomings angemeldet waren stehen die Tübinger oder die Incomings vorne. Diese Dateipfade müssen dabei jedes Semester geändert werden, damit nicht die letztjährigen Tabellen überschrieben werden (siehe 5).

2.3 Änderungen am Matching

Nun können die vom Programm erstellten Tabellen geöffnet und beliebig verändert werden. Dabei ist wichtig, dass die Informationen wieder jeweils in den richtigen Spalten stehen, d.h. das Alter eines Buddys in der passenden Spalte zum Alter steht. Zudem muss die Tabelle anschließend unbedingt als `.csv` Datei gespeichert werden. Zum Öffnen der Tabelle bietet sich dazu `Notepad` an.

3 Änderungen am Google Formular

Bei Änderungen am Google Formular muss unter Umständen auch etwas im R-Programm geändert werden. Dazu wird die R-Datei `matching_algorithm.R` in R mit einem Klick auf **Öffne Skript** geöffnet, und kann dann bearbeitet werden.

3.1 Änderung der Reihenfolge der Fragen

Ändert sich etwas an der Reihenfolge der Fragen, oder wird eine neue Frage hinzugefügt, so ändern sich auch die Positionen der Antworten zu den Fragen in der entstehenden `.csv` Datei. Im R-Programm wird dabei die Spalte festgehalten, in der sich die jeweiligen Informationen befinden. So gibt beispielsweise die Variable `Spalte_Alter_Tuebingen` an, in welche Spalte der Tabelle der Daten der Tübinger Buddys das Alter angegeben wird. Diese Zahl steht jeweils an der Stelle x:

```
spalte_info_tabelle = c(x)
```

So muss bei Änderung der Reihenfolge der Fragen oder beim Einfügen einer neuen Frage in der .csv Tabelle durchgezählt werden, ob die Spaltennummern im Programm dann noch stimmen. Die Variablen befinden sich zum einen im Programm `matching_algorithm.R`, und zwar am Beginn der Funktion `matching_programm` und am Beginn der Funktion `punkte_algorithmus`, und zum anderen am Beginn des Programms `matching_email.R`. Letztere Spaltenangaben beziehen sich auf die große, von `mmatching_algorithm.R` erstellte Tabelle `ausführliche_Übersicht`, und nicht auf die vom Google Formular erstellten Tabellen. Da die Tabelle sich aber auf die vom Google Formular erstellten Tabellen stützt, müssen auch hier die Spaltenzahlen neu überprüft werden.

3.2 Hinzufügen eines Studienfachs

Kommt ein neues Studienfach auf die Auswahlliste im Google Formular, so muss dies auch im R-Programm `matching_algorithm.R` festgehalten werden. Dazu muss in der Funktion `studiengang_funktion` der Studiengang im Bereich der zugehörigen Fakultät und des richtigen Fachbereichs hinzugefügt werden. Er muss dann mit der Konstruktion `| studiengang == 'x'` wie im folgenden Beispielcode eingefügt werden, und zwar **mit der exakten Schreibweise wie im Formular**, d.h. auch alle Leerzeichen müssen gleich sein (am besten im Copy-Paste aus dem Formular kopieren und einfügen), und ebenfalls in Anführungszeichen.

```
(...)  
else if(studiengang = "schon eingetragene Studiengaenge" | studiengang == "x"){  
  return(c("hier steht der zugehoerige Fachbereich",F))  
}  
(...)
```

Das `|` Zeichen steht bei R für das logische Oder, das heißt hier wird für das eingetragene Studienfach `studiengang` des Buddys überprüft, ob es mit einem der aufgeführten Studienfächer übereinstimmt, und wenn dies der Fall ist, wird der zugehörige Fachbereich und die Fakultät zurückgegeben.

3.3 Hinzufügen eines Hobbies

Wenn ein neues Hobby hinzugefügt wird, so muss im R-Programm nur etwas geändert werden, falls eine neue "Kategorie"(wie z.B. `Sports`) hinzukommt. Dann muss in der Funktion `hobbykategorie_funktion` diese neue Kategorie hinzugefügt werden. Sie kann einfach im Vektor `Hobbies` mit einem Komma eingefügt werden. Auch hier muss wieder darauf geachtet werden, dass die Schreibweise dieselbe ist wie im Formular, und auch hier muss das Wort wieder in Anführungszeichen geschrieben werden.

```
Hobbies = c("Hobbykategorie 1", "Hobbykategorie 2", "neue Hobbykategorie")
```

Wenn im Formular mehr als 3 Hobbies angegeben werden sollen können, so muss für jede hinzugekommene Hobby-Auswahl-Frage Folgendes geändert werden, was ich hier am Beispiel einer vierten hinzukommenden Auswahlmöglichkeit beschreibe: Zuerst einmal muss am Beginn der Funktion `punkte_algorithmus` eine neue Variable für jede Tabelle hinzugefügt werden. Diese kann theoretisch beliebig benannt werden, im Folgenden wird sie einfachheitshalber `Spalte_Hobby4_Tuebingen` bzw. `Spalte_Hobby4_Incoming` genannt. Dann muss hinter der Variable mit `= c(x)` die Spaltennummer `x` des Hobbys festgehalten werden (vgl. 3.1). Dann muss im Bereich `HOBBSYS` derselben Funktion die Hobbyspalte zu den Vektoren `hobbies_tü` und `hobbies_in` hinzugefügt werden, und zwar indem jeweils mit einem Komma `unnamed(tuebingen[Spalte_Hobby4_Tuebingen]` bzw. `unnamed(tuebingen[Spalte_Hobby4_Incoming]` in die zum `c()` gehörige Klammer eingefügt wird. Anschließend muss in der darunter stehenden for-Schleife der Vektor von 3 auf 4 vergrößert werden, und das vierte Hobby muss abgefragt werden, d.h. die Schleife muss wie folgt aussehen:

```
for(h in 1:4){
  if(hobbies_in[[h]] == hobbies_tü[[1]] | hobbies_in[[h]] == hobbies_tü[[2]]
    | hobbies_in[[h]] == hobbies_tü[[3]] | hobbies_in[[h]] == hobbies_tü[[4]]){
    score <- score+Hobby_Gewichtung
  }
}
```

Dann muss nur noch die Funktion `hobbykategorie_funktion` erweitert werden. Hier muss in der `if` Abfrage auch das vierte Hobby überprüft werden, sodass sie dann wie folgt aussehen sollte:

```
if(startsWith(toString(hobby_in), hobby)
&((startsWith(toString(hobbies_tü[[1]]),hobby)&hobby_nicht_genutzt(h, hobbies_in, hobby))
| (startsWith(toString(hobbies_tü[[2]]),hobby)&hobby_nicht_genutzt(h, hobbies_in, hobby))
| (startsWith(toString(hobbies_tü[[3]]),hobby)&hobby_nicht_genutzt(h, hobbies_in, hobby))
| (startsWith(toString(hobbies_tü[[4]]),hobby)&hobby_nicht_genutzt(h, hobbies_in, hobby))))){
  return(TRUE)
}
```

4 Änderung der Gewichtung der Matchingkriterien

Soll die Gewichtung der Übereinstimmungen, also beispielsweise die Punktzahl die vergeben wird, wenn Tübinger und Incoming dasselbe studieren, vergeben wird, geändert werden, so muss dies in der R-Datei `matching_algorithm.R` geändert werden. Dies geschieht in der Funktion `punkte_algorithmus`. Hier sind am Anfang Variablen für die Gewichtungen definiert; dabei ist die Variable immer nach der Kategorie benannt, für die Punkte vergeben werden. So gibt zum Beispiel `Studienfach_Gewichtung = c(x)` an, dass für eine Übereinstimmung des Studienfachs `x` Punkte vergeben werden sollen. Möchte man die Punktzahl ändern, so wird einfach an die Stelle `x` die neue Punktzahl geschrieben.

5 Änderung am Output des Matching Programms

5.1 Änderung an ausführlicher Übersicht

Möchte man den Dateipfad zur Speicherung der ausführlichen Tabelle ändern, so geschieht dies im Programm `matching_algorithm.R`, am Ende der Funktion `matching_programm`. Dort findet sich folgender Befehl:

```
write.csv2(ausführliche_Übersicht, file='Dateipfad\\ausführliche_Übersicht.csv')
```

Dabei kann an der Stelle `Dateipfad` der gewünschte Dateipfad spezifiziert werden. Auch hier müssen im Dateipfad wieder alle \doppelt geschrieben werden. Anschließend wird noch der Dateiname der zu erstellenden Tabelle mit `ausführliche_Übersicht.csv` spezifiziert.

5.2 Änderung an Kurzübersicht

Möchte man die in der Kurzübersicht dargestellten Informationen verändern, so geschieht dies im Programm `matching_algorithm.R`, am Ende der Funktion `matching_programm`. Dort befinden sich die beiden Vektoren `auswahl_incoming` und `auswahl_tuebingen`. Hier können jeweils beliebig die Spalten aufgezählt werden, die in der Kurzübersicht dargestellt werden sollen. Die Spaltennamen kann man dabei ganz am Anfang der Funktion `matching_programm` nachlesen. Würde man die Vektoren wie im folgenden Beispiel befüllen, so würde die Kurzübersicht nur den Nachnamen und den Studiengang des Incomings, und nur das Alter und Geschlecht des Tübingers anzeigen:

```
auswahl_incoming <- (Spalte_Nachname_Incoming, Spalte_Uni_Incoming)
auswahl_tuebingen <- (Spalte_Alter_Tuebingen, Spalte_Geschlecht_Tuebingen)
```

Möchte man den Dateipfad zur Speicherung der Kurzübersicht ändern, so geschieht dies im Programm `matching_algorithm.R`, am Ende der Funktion `matching_programm`. Dort findet sich folgender Befehl:

```
write.csv2(Kurzübersicht, file='Dateipfad\\Kurzübersicht.csv')
```

Dabei kann an der Stelle `Dateipfad` der gewünschte Dateipfad spezifiziert werden. Auch hier müssen im Dateipfad wieder alle \doppelt geschrieben werden. Anschließend wird noch der Dateiname der zu erstellenden Tabelle mit `ausführliche_Übersicht.csv` spezifiziert.