Voorbeeld script opmerking analyser

Deze notebook zal een voorbeeld geven hoe je kunt omgaan met de package Deliflor.opmerking.analyser. De eerste stap die wij doen is het inladen van de bestanden. Het inladen van sample_woordenboek is geen verplichting, deze wordt door de functies die een woordenboek nodig heeft automatisch ingeladen.

```
sample_kloon_met_hun_opmerkingen <- Deliflor.opmerking.analyser::kloon_met_hun_opmerkingen
sample_kloon_met_objectieve_waarden <- Deliflor.opmerking.analyser::kloon_met_objectieve_waarden
sample_kloon_met_ouders <- Deliflor.opmerking.analyser::kloon_met_ouders
sample.woordenboek <- Deliflor.opmerking.analyser::sample_woordenboek
```

Als eerste voor data manipulatie gaan wij de objectieve waarden veranderen. Als we eerst gaan we kijken naar een kloonnummer:

```
head(sample_kloon_met_objectieve_waarden[sample_kloon_met_objectieve_waarden$Kloonnummer==29393,], 10)
```

```
##       Kloonnummer Reactietijd Takgewicht Lengte Aantal.bloemen Bladlengte
## 21871       29393          53          0      0              0          0
## 22580       29393          52          0      0              0          0
## 22970       29393          57          0      0              0          0
## 23287       29393          56          0      0              0          0
## 24716       29393          50          0      0              0          0
## 25713       29393          51          0      0              0          0
## 27187       29393          54          0      0              0          0
## 27339       29393          53          0      0              0          0
## 28316       29393          50          0      0              0          0
## 29093       29393          58          0      0              0          0
```

Zie dat voor deze kloonnummer verschillende waarden zijn in de kolom “Reactietijd”. Dit komt doordat deze in verschillende bloeiproeven op een andere dag in bloei stond.

Om te zorgen dat wij een enkele waarden hebben van deze kloon gebruiken wij eerst de functie objectieve_waarden_gem. Deze functie heeft twee inputs nodig. De eerste is de data.frame sample_kloon_met_objectieve_waarden. De tweede is een getal. Dit getal staat voor de kolom die wij willen onderzoeken. In ons geval 2 voor de Reactietijd.

```
objectief <- Deliflor.opmerking.analyser::objectieve_waarden_gem(sample_kloon_met_objectieve_waarden, 2)
objectief[objectief$k.nummer==29393,]
```

```
##      k.nummer objectief
## 9497    29393     54.92
```

Hierbij is de dataset voor de objectieve data klaar. We zullen nu onze focus leggen op de manipulatie van de opmerkingen. Hiervoor gebruiken wij de dataset sample_kloon_met_hun_opmerkingen met de focus op Opmerking.

```
sample_kloon_met_hun_opmerkingen[1830:1836,]
```

```
##      Kloonnummer                                                    Opmerking
## 1830          25                                                  A9-mei'17,
## 1831          25                                                   W-nov'17,
## 1832          58                                Jul-08, gaatjes in hrt, 33003
## 1833          58                              Aug-08, KRO, mat gr, wat open hrt,
## 1834          58                                          CChMVd-aug'11, nk?
## 1835          58                           traag, rom hrt, open hrt, hier groei 2
## 1836          58 Pl-apr-10, openhart, hardesteel, goedblad, blmvorm te klein,
##      Jaar Locatie Bloeiproefnummer Veld Beoordeling  PNH Gebruik.NL Doelgroep
## 1830 2017     KKD            17015  808           1 ....          #         Y
## 1831 2017     KKD            17047  859           1 ....          #         Y
## 1832 2008     KKD             7493  223           1   PH         Pl        PO
## 1833 2008     KKD             7494   80           1 ..N.         Pl        PO
## 1834 2011     KKD             5028   86           1 ....         Pl        PO
## 1835 2005     NWW              426  328           1   HH         Pl        PO
## 1836 2010     KKD             9488  217           1 HNN.         Pl        PO
##      Bloemtype
## 1830         #
## 1831         #
## 1832         p
## 1833         p
## 1834         p
## 1835         p
## 1836         p
```

Het kan voorkomen dat in de opmerking-data diakritische tekens voorkomen. Dit zijn onderandere de aanhalingstekens boven de letters. Hier weet R niet altijd even goed mee om te gaan. Daarvoor is de volgende functie gemaakt indien het nodig is deze te verwijderen. Ook hierbij geven we de kolom aan waar onze data staat.

```
opmerking <- Deliflor.opmerking.analyser::diakritische_tekens_verwijderen(sample_kloon_met_objectieve_wa
```

Vervolgens kunnen we gaan beginnen met het manipuleren van onze opmerking data. Momenteel is het nog een brij met allemaal woorden aan elkaar. Dit gaan we met de functie opmerking_split van elkaar los trekken. De functie doet dat door te zoeken naar de leestekens ‘,’-komma, ‘.’-punt en ‘|’-sluisteken. Iedere keer als de functie deze tekens vindt maakt hij twee indivuele regels van. Ook hier moeten wij nog aangeven om welke kolom het gaat.

```
opmerking_los <- Deliflor.opmerking.analyser::opmerking_split(sample_kloon_met_hun_opmerkingen, 2, "onb
head(opmerking_los)
```

```
##    opmerking
## 1 Col-dec'19
## 2 J&A-juni16
## 3  TL-okt'14
## 4  TL-jul'14
## 5  TL-jul'16
## 6  TL-okt'16
```

Wat misschien opvalt is dat naast de sample data en het getal ook het woord 'onbegrensd' is meegenomen. Dit is omdat deze functie op twee manieren kan werken. De eerste is onbegrensd. Dit houdt in dat alleen het woord behouden blijft en de rest van de informatie wordt weg gegooid. Dit is puur voor het bekijken van de woorden die in de dataset voorkomen. De tweede is per_key. Met per_key onthoudt hij welk kloonnummer in relatie stond met de opmerking. Hiermee kun je minder accuraat de woorden bekijken, maar is juist bedoelt om het aantal woorden per k.nummer in kaart te brengen.

```
opmerking_per_key <- Deliflor.opmerking.analyser::opmerking_split(sample_kloon_met_hun_opmerkingen, 2,
head(opmerking_per_key)
```

```
##    k.nummer  opmerking
## 1         1 Col-dec'19
## 3         1 J&A-juni16
## 5         1  TL-okt'14
## 7         1  TL-jul'14
## 9         1  TL-jul'16
## 11        1  TL-okt'16
```

Dit effect wordt duidelijker zodra we de opmerkingen gaan optellen. Afhankelijk van de eerdere keuze zal ook de volgende functie op twee verschillende manieren werken.

```
opmerking_los_tel <- Deliflor.opmerking.analyser::tel_aantal_per_woord(opmerking_los)
opmerking_per_key_tel <- Deliflor.opmerking.analyser::tel_aantal_per_woord(opmerking_per_key)
head(opmerking_los_tel)
```

```
##   opmerking  Freq
## 1     traag 20167
## 2     mager 14832
## 3     zwaar 10639
## 4      kort  9191
## 5    stevig  7766
## 6    reflex  6888
```

```
tail(opmerking_per_key_tel[opmerking_per_key_tel$k.nummer==29393,], 10)
```

```
##        k.nummer          opmerking Freq
## 152009    29393              traag    2
## 152010    29393 tripsschade linten    1
## 152011    29393         tussenmaat    1
## 152012    29393        vold lengte    1
## 152013    29393          wat ongel    1
## 152014    29393        wat rom blm    1
## 152015    29393       wat stakerig    1
## 152016    29393            wat stm    1
## 152017    29393          wat traag    1
## 152018    29393              zwaar    1
```

Zoals te zien is zal de functie met de optie 'onbegrensd' alle woorden die hetzelfde zijn bij elkaar optellen. Het woord 'traag' komt in de gehele dataset 20167 keer voor. Als de optie 'per_key' was gekozen telt hij ook de woorden bij elkaar op, alleen zal dit gedaan worden binnen een kloonnummer, in ons geval staat het woord traag voor kloonnummer 29393 maar 2 keer voor.

Zoals eerder was gesteld werkt de optie onbegrensd beter om de gebruikte woorden in kaart te brengen. Dit is te doen met de functies uitslag_zuiverheid_vs_woordenboek en uitslag_beste_ratio. Met de functie uitslag_zuiverheid_vs_woordenboek wordt er gekeken hoeveel van de opmerkingen in de dataset overeenkomen met het woordenboek. Indien geen woordenboek meegegeven wordt zal de default-woordenboek van de package gebruikt worden. Met top_woorden kun je zelf bepalen hoeveel van de woorden er bekeken kunnen worden.Met ratio is het mogelijk aan te geven hoeveel de woorden op elkaar moeten lijken.

Deze ratio is te bepalen met de functie uitslag_beste_ratio. Ook hier kunnen we weer een eigen woordenboek meegeven en bepalen hoeveel van de woorden we gaan bekijken.

```
uitslag <- Deliflor.opmerking.analyser::uitslag_zuiverheid_vs_woordenboek(opmerking_los_tel, top_woorde
```

```
## Registered S3 methods overwritten by 'ffbase':
##   method   from
##   [.ff     ff
##   [.ffdf   ff
##   [<-.ff   ff
##   [<-.ffdf ff
```

```
tail(uitslag[[1]])
```

```
##                       opmerking Freq      V3
## 495                   Pl-okt'19  370    <NA>
## 496                   Sa-mei'18  370    <NA>
## 497                  viezekleur  370 In WB 1
## 498                   Pl-jan'15  369    <NA>
## 499                   A9-feb'11  367    <NA>
## 500 ref hitte traag Tr-juli'17  367    <NA>
```

```
uitslag[[2]]
```

```
##       Gevonden woord    Lijkt op     ratio
## 2377     mooie kleur  mooiekleur 0.9090909
## 2743      kleine blm   kleineblm 0.9000000
## 21457     weinigeblm   weinigblm 0.9000000
## 25630     weinigeblm  weinig blm 0.9000000
## 47613     slecht bld slecht blad 0.9090909
## 50502     slecht bld   slechtbld 0.9000000
```

uitslag_zuiverheid_vs_woordenboek geeft twee outputs. De eerste output is een data.frame. Hierin zal in kolom een de top_woorden te zien zijn, gevolgd door 'In WB 2', 'In WB 1', 'Boven ratio' of NA. Als het woord gevolgd wordt door 'In WB 1' of 'In WB 2' betekent dit, dat het woord al in uw woordenboek staat en aangepast gaat worden. Als het woord gevolgd wordt door 'Boven ratio' is het woord vergelijkbaar met een woord uit uw woordenboek en zal ook dez veranderd worden. Als het woord gevolgd wordt door NA betekent dat het woord niet te vinden is en verwijderd zal worden.

De tweede output zal een data.frame zijn met drie kolommen. De eerste kolom met het woord dat gevonden is uit uw dataset die niet voorkomt in het woordenboek, de tweede kolom het woord dat wel in het woordenboek staat en waar deze op lijkt. En als laatste de derde kolom wat de ratio laat zien hoeveel deze woorden op elkaar lijken. Deze woorden zou je kunnen toevoegen aan het woordenboek voor een betere kwaliteit.

Beide outputs worden gebaseerd op een ratio. Deze ratio is te bepalen met de functie uitslag_beste_ratio. Deze functie zal alle woorden die zijn meegegeven met elkaar vergelijken en bepalen hoeveel deze op elkaar lijken. Vervolgens controleert hij beide woorden met het woordenboek. Als beide woorden in het woordenboek gelijk zijn zal hij deze ook als gelijkwaardig zien. Dit zal een output geven.

```
Deliflor.opmerking.analyser::uitslag_beste_ratio(opmerking_los_tel, top_woorden = 500, woordenboek = sa
```

```
##      perc false_pos false_neg true_pos true_neg totaal_fal totaal_tru
## 0.7  0.70         5        79       61    14904         84      14965
## 0.71 0.71         5        84       56    14904         89      14960
## 0.72 0.72         3        85       55    14906         88      14961
## 0.73 0.73         2        87       53    14907         89      14960
## 0.74 0.74         2        87       53    14907         89      14960
## 0.75 0.75         2        87       53    14907         89      14960
## 0.76 0.76         2        94       46    14907         96      14953
## 0.77 0.77         2        95       45    14907         97      14952
## 0.78 0.78         2        99       41    14907        101      14948
## 0.79 0.79         2        99       41    14907        101      14948
## 0.8  0.80         2        99       41    14907        101      14948
## 0.81 0.81         2       105       35    14907        107      14942
## 0.82 0.82         2       109       31    14907        111      14938
## 0.83 0.83         2       109       31    14907        111      14938
## 0.84 0.84         2       111       29    14907        113      14936
## 0.85 0.85         2       111       29    14907        113      14936
## 0.86 0.86         2       114       26    14907        116      14933
## 0.87 0.87         2       115       25    14907        117      14932
## 0.88 0.88         2       119       21    14907        121      14928
## 0.89 0.89         2       129       11    14907        131      14918
## 0.9  0.90         2       129       11    14907        131      14918
## 0.91 0.91         1       135        5    14908        136      14913
## 0.92 0.92         1       136        4    14908        137      14912
## 0.93 0.93         0       139        1    14909        139      14910
## 0.94 0.94         0       140        0    14909        140      14909
## 0.95 0.95         0       140        0    14909        140      14909
##      false positive rate false negative rate sensitivity specificity
## 0.7         3.353679e-04           0.5642857 0.435714286   0.9996646
## 0.71        3.353679e-04           0.6000000 0.400000000   0.9996646
## 0.72        2.012207e-04           0.6071429 0.392857143   0.9997988
## 0.73        1.341472e-04           0.6214286 0.378571429   0.9998659
## 0.74        1.341472e-04           0.6214286 0.378571429   0.9998659
## 0.75        1.341472e-04           0.6214286 0.378571429   0.9998659
## 0.76        1.341472e-04           0.6714286 0.328571429   0.9998659
## 0.77        1.341472e-04           0.6785714 0.321428571   0.9998659
## 0.78        1.341472e-04           0.7071429 0.292857143   0.9998659
## 0.79        1.341472e-04           0.7071429 0.292857143   0.9998659
## 0.8         1.341472e-04           0.7071429 0.292857143   0.9998659
## 0.81        1.341472e-04           0.7500000 0.250000000   0.9998659
## 0.82        1.341472e-04           0.7785714 0.221428571   0.9998659
## 0.83        1.341472e-04           0.7785714 0.221428571   0.9998659
## 0.84        1.341472e-04           0.7928571 0.207142857   0.9998659
## 0.85        1.341472e-04           0.7928571 0.207142857   0.9998659
## 0.86        1.341472e-04           0.8142857 0.185714286   0.9998659
## 0.87        1.341472e-04           0.8214286 0.178571429   0.9998659
## 0.88        1.341472e-04           0.8500000 0.150000000   0.9998659
## 0.89        1.341472e-04           0.9214286 0.078571429   0.9998659
## 0.9         1.341472e-04           0.9214286 0.078571429   0.9998659
## 0.91        6.707358e-05           0.9642857 0.035714286   0.9999329
## 0.92        6.707358e-05           0.9714286 0.028571429   0.9999329
```

```
## 0.93        0.000000e+00           0.9928571 0.007142857   1.0000000
## 0.94        0.000000e+00           1.0000000 0.000000000   1.0000000
## 0.95        0.000000e+00           1.0000000 0.000000000   1.0000000
```

Ik adviseer een combinatie te zoeken met een zo klein mogelijk false_pos en totaal_fal.

Deze ratio wordt ook gebruikt in de functie zuiver_maken_met_woordenboek. Deze functie vergelijkt alle woorden in de dataset met alle woorden in het woordenboek en gebruik daarmee veel rekenkracht. Hiervoor is er mogelijkheid gemaakt om aan te geven hoeveel processoren er gebruikt mogen worden. De functie zal alle opmerkingen gelijk maken aan de tweede kolom van het woordenboek. Woorden die te veel verschillen met het woordenboek zullen niet worden meegenomen in de output. Het is mogelijk zowel de onbegrensd als per_key opties te gebruiken.

```
opmerking_los_tel_zuiver <- Deliflor.opmerking.analyser::zuiver_maken_met_woordenboek(opmerking_los_tel
```

```
## Grove filter, woorden vervangen met woordenboek.
```

```
## Vergelijkbare woorden vervangen gebaseerd op ratio.
```

```
## Vervangen woorden samenvoegen.
```

```
opmerking_per_key_tel_zuiver <- Deliflor.opmerking.analyser::zuiver_maken_met_woordenboek(opmerking_per
```

```
## Grove filter, woorden vervangen met woordenboek.
```

```
## Vergelijkbare woorden vervangen gebaseerd op ratio.
```

```
## Vervangen woorden samenvoegen.
```

Na het zuiveren van de data waarbij de optie onbegrend was gebruikt (opmerking_los_tel_zuiver) kan je nogmaals de functies uitslag_zuiverheid_vs_woordenboek en uitslag_beste_ratio uit voeren. Dit is tevens de laatste functie die je hier op kan uitvoeren.

Met de per_key optie worden de functies hier beneden uitgevoerd. Zo zal de functie samenvoegen_opmerking_objectief, de output opmerking_per_key_tel_zuiver samenvoegen met de eerder gemaakte output objectief. In beide outputs komen k.nummers voor en zullen worden gekoppeld.

```
kloon_opm_obj <- Deliflor.opmerking.analyser::samenvoegen_opmerking_objectief(opmerking_per_key_tel_zui
```

Om echt goed te kunnen kijken naar de data doen we nog een laatste filter. Deze is om onderscheid te maken tussen woorden als groepen. De functie zoekt alle woorden met de bijbehorende gegevens. Tevens voegt deze functie een nieuw woord toe, namelijk het woord 'geen'. Deze presenteerd de k.nummers die geen van de ingevoerde woorden heeft.

Als voorbeeld kijken we naar de woorden (traag en snel) in combinatie met een k.nummer met de woorden 'groenhart', 'leuk' en 'mooi', maar zonder de woorden 'traag' en/of 'snel' dan zal deze het woord 'geen' krijgen.

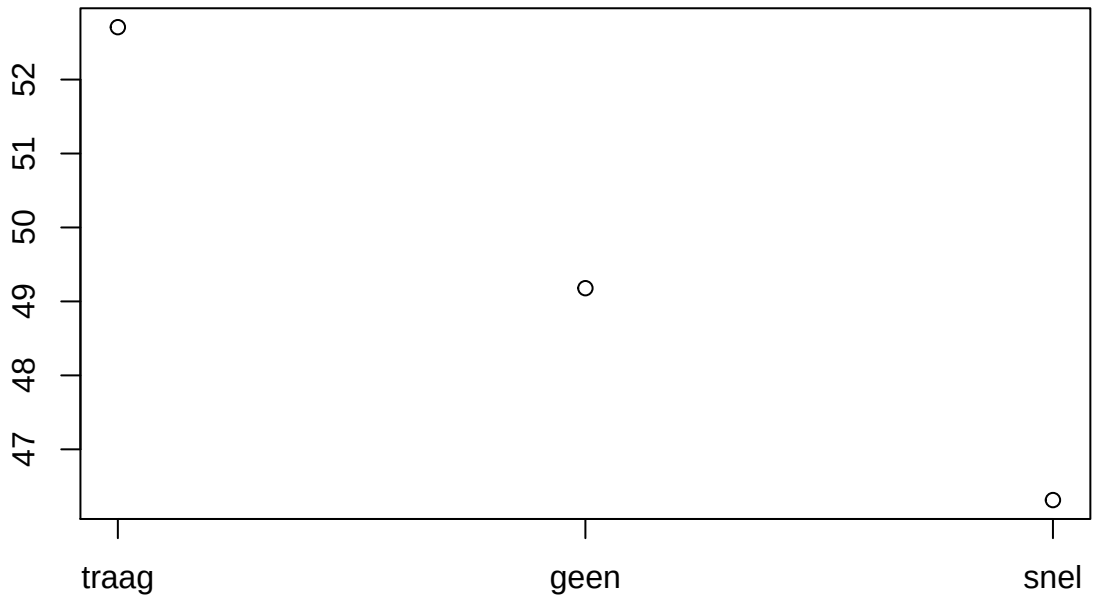
```
uitslag_data1 <- Deliflor.opmerking.analyser::keuze_woorden_voor_analyse(kloon_opm_obj, c("traag", "snel
```

Nu is de data correct om uitslagen te geven. De functie Uitslag_verschillende_woorden geeft het aantal k.nummers en de gemiddelde objectieve waarden per woord weer. Ook zet hij de gemiddelden uit in een grafiek en doet een correlatie toets.

De functie Uitslag_per_los_woord heeft als input een enkel woord nodig naast de dataset. De functie zet twee grafieken uit. De eerst zal de spreiding van de objectieve data zijn, uitgezet tegen het aantal keer voorkomen van het woord. Als tweede een box-diagram met een trendlijn die alleen gebaseerd is op het aantal keer voorkomen van het woord.

Als laatste voert de functie uitslag_regressie een linear regression toets uit waarbij een relatie wordt getrokken tussen de woorden en het aantal keer dat deze woorden voorkwamen.

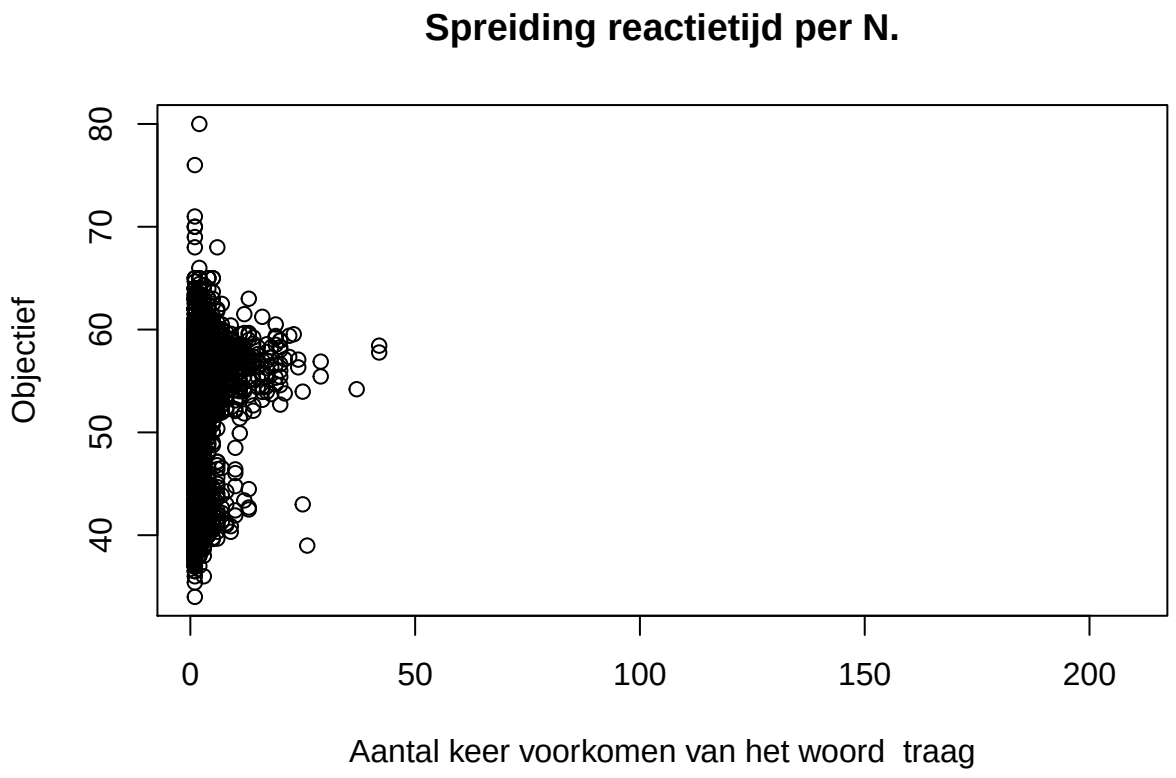
```
Deliflor.opmerking.analyser::Uitslag_verschillende_woorden(uitslag_data1)
```

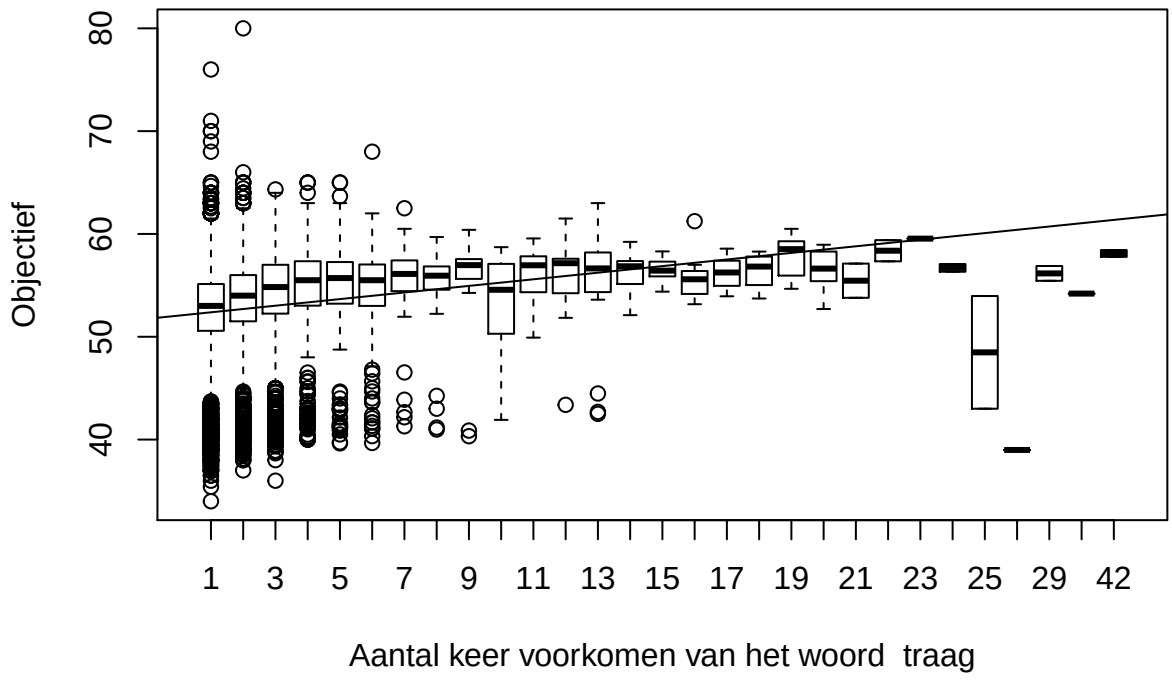



```
## [[1]]
##   woord      gem_reactie aantal
## 2 traag 52.7082834774931   9180
## 3  geen 49.1789875938147  27500
## 4  snel 46.3138647371814   3281
##
## $cor_test
##
##  Pearson's product-moment correlation
##
## data:  tabel[tabel$dummie %in% c(dummie_find), 4] and tabel[tabel$dummie %in% c(dummie_find), "dummi
## t = -68.778, df = 39959, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3340844 -0.3165505
## sample estimates:
##        cor
## -0.3253454
```

```
Deliflor.opmerking.analyser::Uitslag_per_los_woord(uitslag_data1, woord = "traag")
```






```
Deliflor.opmerking.analyser::uitslag_regressie(uitslag_data1)
```

```
##
## Call:
## lm(formula = objectief ~ opmerking:Freq, data = tabel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.422  -2.219   1.115   3.527  29.862
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          50.59149    0.06609  765.44   <2e-16 ***
## opmerkingsnel:Freq  -1.12930    0.03982  -28.36   <2e-16 ***
## opmerkingtraag:Freq  0.64683    0.02391   27.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.459 on 12458 degrees of freedom
##   (31659 observations deleted due to missingness)
## Multiple R-squared:  0.143,  Adjusted R-squared:  0.1428
## F-statistic:  1039 on 2 and 12458 DF,  p-value: < 2.2e-16
```

Voor de laatste uitslag is het nodig om de ouders van het kloonnummer te hebben. We passen de data van de sample_kloon_met_ouders hiervoor aan. We gebruiken de kolommen kloonnummer, kloon.pa en kloon.Ma

en zorgen ervoor dat de kolom kloonnummer veranderd naar de naam k.nummer. Hierna koppelen we de dataset met de output kloon_opm_obj die uit de functie samenvoegen_opmerking_objectief kwam. Hierna kiezen we opnieuw de woorden traag en snel in de functie keuze_woorden_voor_analyse.

```
sample_kloon_met_ouders <- sample_kloon_met_ouders[,c(1,4,5)]
names(sample_kloon_met_ouders) <- c("k.nummer", "Kloon.Pa", "Kloon.Ma")
kloon_opm_obj_ouder <- Deliflor.opmerking.analyser::samenvoegen_onderzoeksdata_ouders(kloon_opm_obj, sa
uitslag_data2 <- Deliflor.opmerking.analyser::keuze_woorden_voor_analyse(kloon_opm_obj_ouder, c("traag"
```

Met deze output is het mogelijk om de laatste functie genaamt uitslag_overerving te gebruiken. Het blijft echter ook mogelijk om de uitslag functies Uitslag_verschillende_woorden, Uitslag_per_los_woord en uitslag_regressie uit te voeren.

De functie uitslag_overerving geeft een list met verschillende outputs. Als eerste een linear regression toets waarbij een voorspelling gedaan wordt over de objectieve waarden van het k.nummer met de woorden van de ouders en hun objectieve waarden. Als tweede een tabel waarbij het gemiddelde als ook de standaardafwijking van de objectieve waarden van de kloonnummers per unieke woorden-combinatie te zien zijn van beide ouders. Als laatste worden de gemiddelde en standaardafwijking uitgezet in een plot.

```
Deliflor.opmerking.analyser::uitslag_overerving(uitslag_data2)
```

```
## [[1]]
##
## Call:
## lm(formula = k.objectief ~ pa.woord:pa.objectief:ma.woord:ma.objectief,
##     data = tabel)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4586  -2.5053   0.0375   2.5112  21.0634
##
## Coefficients:
##                                                                  Estimate
## (Intercept)                                                     2.385e+01
## pa.woordgeen:pa.objectief:ma.woordgeen:ma.objectief             1.023e-02
## pa.woordsnel:pa.objectief:ma.woordgeen:ma.objectief             1.072e-02
## pa.woordtraag:pa.objectief:ma.woordgeen:ma.objectief            9.943e-03
## pa.woordtraag:snel:pa.objectief:ma.woordgeen:ma.objectief       1.028e-02
## pa.woordgeen:pa.objectief:ma.woordsnel:ma.objectief             1.033e-02
## pa.woordsnel:pa.objectief:ma.woordsnel:ma.objectief             1.076e-02
## pa.woordtraag:pa.objectief:ma.woordsnel:ma.objectief            1.009e-02
## pa.woordtraag:snel:pa.objectief:ma.woordsnel:ma.objectief       1.045e-02
## pa.woordgeen:pa.objectief:ma.woordtraag:ma.objectief            9.955e-03
## pa.woordsnel:pa.objectief:ma.woordtraag:ma.objectief            1.038e-02
## pa.woordtraag:pa.objectief:ma.woordtraag:ma.objectief           9.676e-03
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:ma.objectief      9.920e-03
## pa.woordgeen:pa.objectief:ma.woordtraag:snel:ma.objectief       1.009e-02
## pa.woordsnel:pa.objectief:ma.woordtraag:snel:ma.objectief       1.042e-02
## pa.woordtraag:pa.objectief:ma.woordtraag:snel:ma.objectief      9.789e-03
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:snel:ma.objectief 1.030e-02
##                                                                 Std. Error
## (Intercept)                                                      1.662e-01
## pa.woordgeen:pa.objectief:ma.woordgeen:ma.objectief              6.703e-05
## pa.woordsnel:pa.objectief:ma.woordgeen:ma.objectief              7.909e-05
```

```
## pa.woordtraag:pa.objectief:ma.woordgeen:ma.objectief             6.566e-05
## pa.woordtraag:snel:pa.objectief:ma.woordgeen:ma.objectief        8.860e-05
## pa.woordgeen:pa.objectief:ma.woordsnel:ma.objectief              7.827e-05
## pa.woordsnel:pa.objectief:ma.woordsnel:ma.objectief              8.891e-05
## pa.woordtraag:pa.objectief:ma.woordsnel:ma.objectief             7.424e-05
## pa.woordtraag:snel:pa.objectief:ma.woordsnel:ma.objectief        9.635e-05
## pa.woordgeen:pa.objectief:ma.woordtraag:ma.objectief             6.516e-05
## pa.woordsnel:pa.objectief:ma.woordtraag:ma.objectief             7.749e-05
## pa.woordtraag:pa.objectief:ma.woordtraag:ma.objectief            6.422e-05
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:ma.objectief       8.980e-05
## pa.woordgeen:pa.objectief:ma.woordtraag:snel:ma.objectief        8.038e-05
## pa.woordsnel:pa.objectief:ma.woordtraag:snel:ma.objectief        1.009e-04
## pa.woordtraag:pa.objectief:ma.woordtraag:snel:ma.objectief       7.992e-05
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:snel:ma.objectief  1.748e-04
##                                                                 t value
## (Intercept)                                                      143.52
## pa.woordgeen:pa.objectief:ma.woordgeen:ma.objectief              152.57
## pa.woordsnel:pa.objectief:ma.woordgeen:ma.objectief              135.50
## pa.woordtraag:pa.objectief:ma.woordgeen:ma.objectief             151.43
## pa.woordtraag:snel:pa.objectief:ma.woordgeen:ma.objectief        116.01
## pa.woordgeen:pa.objectief:ma.woordsnel:ma.objectief              131.92
## pa.woordsnel:pa.objectief:ma.woordsnel:ma.objectief              120.99
## pa.woordtraag:pa.objectief:ma.woordsnel:ma.objectief             135.89
## pa.woordtraag:snel:pa.objectief:ma.woordsnel:ma.objectief        108.50
## pa.woordgeen:pa.objectief:ma.woordtraag:ma.objectief             152.78
## pa.woordsnel:pa.objectief:ma.woordtraag:ma.objectief             134.03
## pa.woordtraag:pa.objectief:ma.woordtraag:ma.objectief            150.66
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:ma.objectief       110.47
## pa.woordgeen:pa.objectief:ma.woordtraag:snel:ma.objectief        125.47
## pa.woordsnel:pa.objectief:ma.woordtraag:snel:ma.objectief        103.34
## pa.woordtraag:pa.objectief:ma.woordtraag:snel:ma.objectief       122.49
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:snel:ma.objectief   58.91
##                                                                 Pr(>|t|)
## (Intercept)                                                       <2e-16 ***
## pa.woordgeen:pa.objectief:ma.woordgeen:ma.objectief               <2e-16 ***
## pa.woordsnel:pa.objectief:ma.woordgeen:ma.objectief               <2e-16 ***
## pa.woordtraag:pa.objectief:ma.woordgeen:ma.objectief              <2e-16 ***
## pa.woordtraag:snel:pa.objectief:ma.woordgeen:ma.objectief         <2e-16 ***
## pa.woordgeen:pa.objectief:ma.woordsnel:ma.objectief               <2e-16 ***
## pa.woordsnel:pa.objectief:ma.woordsnel:ma.objectief               <2e-16 ***
## pa.woordtraag:pa.objectief:ma.woordsnel:ma.objectief              <2e-16 ***
## pa.woordtraag:snel:pa.objectief:ma.woordsnel:ma.objectief         <2e-16 ***
## pa.woordgeen:pa.objectief:ma.woordtraag:ma.objectief              <2e-16 ***
## pa.woordsnel:pa.objectief:ma.woordtraag:ma.objectief              <2e-16 ***
## pa.woordtraag:pa.objectief:ma.woordtraag:ma.objectief             <2e-16 ***
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:ma.objectief        <2e-16 ***
## pa.woordgeen:pa.objectief:ma.woordtraag:snel:ma.objectief         <2e-16 ***
## pa.woordsnel:pa.objectief:ma.woordtraag:snel:ma.objectief         <2e-16 ***
## pa.woordtraag:pa.objectief:ma.woordtraag:snel:ma.objectief        <2e-16 ***
## pa.woordtraag:snel:pa.objectief:ma.woordtraag:snel:ma.objectief   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.076 on 29650 degrees of freedom
```

```
##   (2316 observations deleted due to missingness)
## Multiple R-squared:  0.4761, Adjusted R-squared:  0.4759
## F-statistic:  1684 on 16 and 29650 DF,  p-value: < 2.2e-16
##
##
## [[2]]
##      pa.woord   ma.woord gemiddelde standaardafwijking
## 1        geen       geen   50.56022           4.656787
## 2        snel       geen   49.15943           5.600521
## 3       traag       geen   50.53929           5.575669
## 4  traag:snel       geen   50.80499           5.250536
## 5        geen       snel   47.87837           5.760375
## 6        snel       snel   46.89145           5.677589
## 7       traag       snel   48.72947           5.867117
## 8  traag:snel       snel   48.39443           5.416821
## 9        geen      traag   50.62484           5.177597
## 10       snel      traag   48.83680           5.437052
## 11      traag      traag   50.88606           5.421210
## 12 traag:snel      traag   49.07493           6.451845
## 13       geen traag:snel   50.17922           5.887385
## 14       snel traag:snel   46.67292           6.635121
## 15      traag traag:snel   49.74349           6.193265
## 16 traag:snel traag:snel   48.48958           7.853169
##
## [[3]]
```