Results WordSim353 Assignment 1

Fleur Petit

11 February 2019

Load the data

```
df <- read_csv("results_csv.csv") %>%
  filter(profession != "test") %>%
  mutate(start_time = as.POSIXct(as.character(start_time), format = "%Y-%m-%d %H:%M:%S"),
         end_time = as.POSIXct(as.character(end_time), format = "%Y-%m-%d %H:%M:%S"),
         duration = end_time - start_time) %>%
  select(-c(start_time, end_time)) %>%
  rownames_to_column("id") %>%
  gather(key = word_pair, value = relatedness, -c(id, duration, level, study_type, profession)) %>%
  arrange(id)
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   level = col_character(),
##   study_type = col_character(),
##   profession = col_character(),
##   start_time = col_character(),
##   end_time = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
word_pairs <- read_csv("20190219_wordpairs.csv") %>%
  mutate(word_pair = str_replace(wordpairs, " vs. ", "_")) %>%
  rename(source = similarity) %>%
  select(-wordpairs)
```

```
## Parsed with column specification:
## cols(
##   wordpairs = col_character(),
##   similarity = col_character()
## )
```

```
df <- df %>%
  full_join(word_pairs)
```

```
## Joining, by = "word_pair"
```

Function to calculate the 95% confidence interval

```
ci <- function(mean, sd, n){
  error <- qnorm(0.975)*sd/sqrt(n)
  lower <- mean-error
  upper <- mean+error
  return(tibble(lower = lower,upper = upper))
}
```

Mean, standard deviation, lower and upper 95% confidence interval, median

```
n <- length(unique(df$id))

description <-
  df %>%
  group_by(word_pair) %>%
  mutate(mean = mean(relatedness),
             sd = sd(relatedness),
             lower = ci(mean, sd, n)[["lower"]],
             upper = ci(mean, sd, n)[["upper"]],
             median = median(relatedness)
             )

description %>%
  group_by(word_pair, mean, sd, lower, upper, median, source) %>%
  summarise %>%
  arrange(source) %>%
  kable()
```

word_pair	mean	sd	lower	upper	median	source
archive_bird	2.6052632	3.1325620	1.6092709	3.6012554	1.0	dissimilar
defeating_discipline	6.5000000	2.0500494	5.8481906	7.1518094	7.0	dissimilar
fertility_crane	3.1315789	2.5058963	2.3348339	3.9283240	3.0	dissimilar
fertility_hotel	8.7894737	0.9345586	8.4923325	9.0866148	9.0	dissimilar
governor_jazz	1.6052632	1.8677848	1.0114044	2.1991219	1.0	dissimilar
motto_Jackson	1.5526316	1.9823187	0.9223570	2.1829061	1.0	dissimilar
number_equipment	9.1842105	1.2048385	8.8011344	9.5672867	9.5	dissimilar
phone_artifact	8.6052632	1.6364014	8.0849724	9.1255539	9.0	dissimilar
possession_popcorn	5.2631579	2.4013984	4.4996377	6.0266780	6.0	dissimilar
professor_currency	8.6578947	1.4570669	8.1946230	9.1211665	9.0	dissimilar
round_Jerusalem	9.0263158	1.3653401	8.5922084	9.4604231	10.0	dissimilar
size_environment	3.4210526	2.4674265	2.6365390	4.2055663	3.0	dissimilar
travel_soap	1.8421053	1.9935886	1.2082475	2.4759631	1.0	dissimilar
video_problem	0.9473684	1.5412766	0.4573224	1.4374144	0.0	dissimilar
war_journal	4.2631579	2.5960907	3.4377357	5.0885801	5.0	dissimilar
artifact_object	0.5526316	0.8604640	0.2790486	0.8262145	0.0	similar
collection_memorabilia	9.5526316	1.0829731	9.2083023	9.8969609	10.0	similar
drug_alcohol	0.4736842	0.7965073	0.2204362	0.7269322	0.0	similar
flight_airport	4.4473684	2.2743858	3.7242317	5.1705052	4.0	similar
gem_jewel	1.2368421	1.4599928	0.7726401	1.7010441	1.0	similar
gin_alcohol	7.8157895	2.0382195	7.1677414	8.4638376	8.0	similar
investigation_operation	3.5000000	2.7188034	2.6355615	4.3644385	3.0	similar
minister_bishop	8.0263158	1.6355319	7.5063015	8.5463301	8.0	similar
ministry_government	2.4210526	2.3667925	1.6685353	3.1735699	2.0	similar
music_jazz	0.7368421	1.6054672	0.2263868	1.2472974	0.0	similar
practice_law	3.7368421	2.4680029	2.9521452	4.5215391	3.0	similar
radio_music	5.5789474	2.9921662	4.6275937	6.5303010	7.0	similar
science_scientist	5.3157895	2.3261761	4.5761861	6.0553928	6.0	similar
troops_personnel	9.0526316	0.9571175	8.7483178	9.3569453	9.0	similar
year_century	9.0789474	0.8504873	8.8085365	9.3493582	9.0	similar

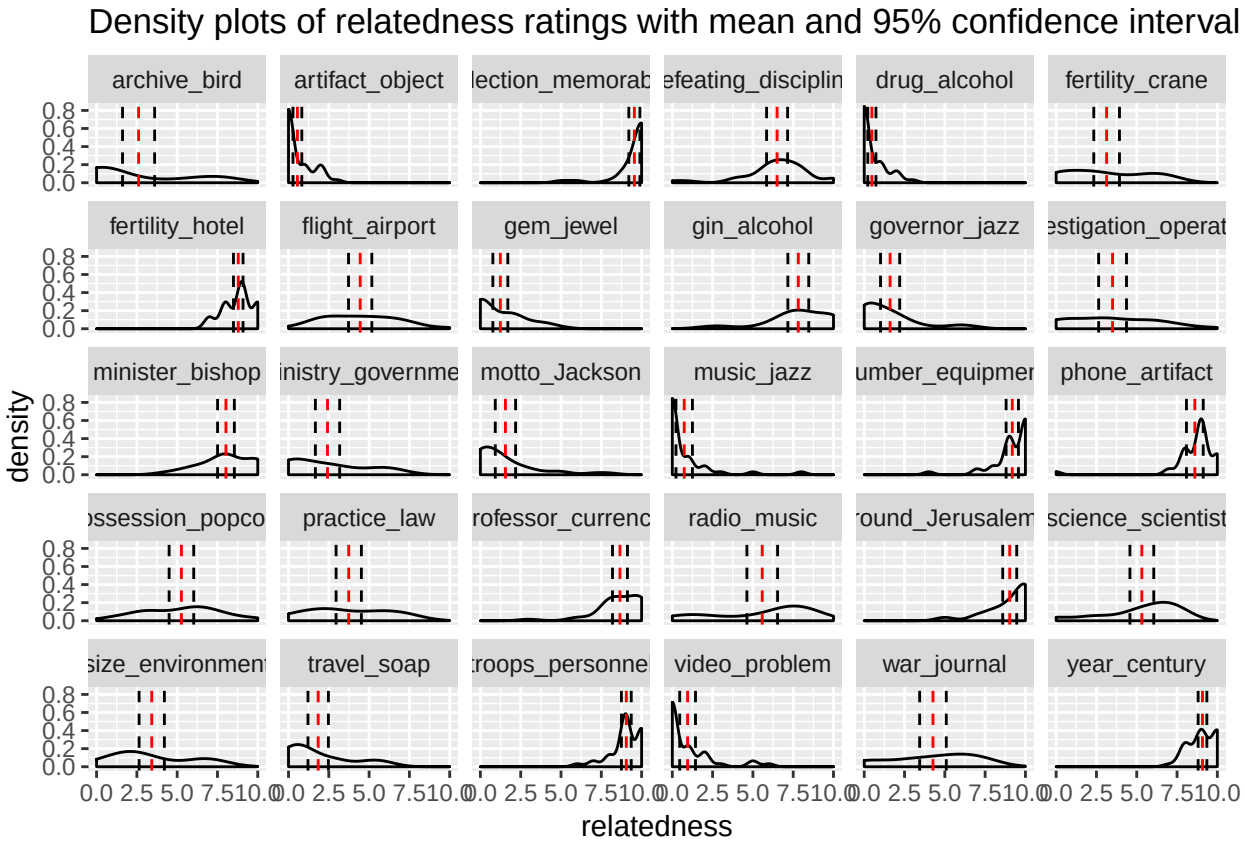
```r
df %>%
  group_by(source) %>%
  summarise(mean = mean(relatedness),
            sd = sd(relatedness),
            lower = ci(mean, sd, n)[["lower"]],
            upper = ci(mean, sd, n)[["upper"]],
            median = median(relatedness)
            ) %>%
  kable()
```

source	mean	sd	lower	upper	median
dissimilar	5.026316	3.644242	3.867636	6.184996	6
similar	4.768421	3.716545	3.586752	5.950090	5

The relatedness ratings are relatively similar for word pairs from the "similar" dataset and word pairs from the "dissimilar" dataset. This may indicate that word relatedness and word similarity are rated according to different standards.

Density plots of relatedness rateings per word pair with mean and 95% confidence interval

```r
ggplot(description, aes(relatedness)) +
  geom_density() +
  facet_wrap(~ word_pair) +
  geom_vline(aes(xintercept = mean), colour = "red", linetype = "dashed") +
  geom_vline(aes(xintercept = lower), linetype = "dashed") +
  geom_vline(aes(xintercept = upper), linetype = "dashed") +
  ggtitle("Density plots of relatedness ratings with mean and 95% confidence interval")
```



Pairs with a relatively large spread

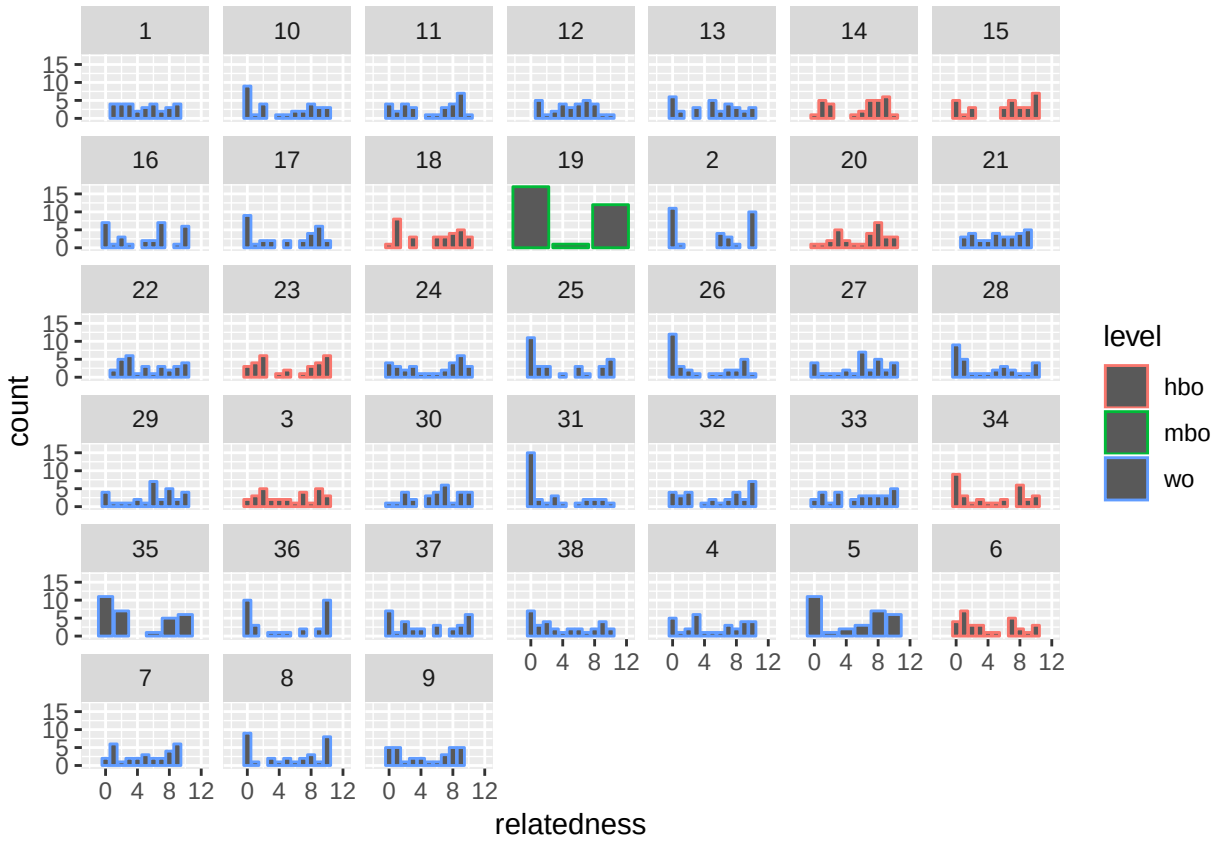
A sd > 2.5 has been chosen arbitrarily to indicate a relatively large spread.

```
description %>%
  group_by(word_pair, sd) %>%
  summarise() %>%
  filter(sd > 2.5) %>%
  arrange(sd) %>%
  kable()
```

word_pair	sd
fertility_crane	2.505896
war_journal	2.596091
investigation_operation	2.718803
radio_music	2.992166
archive_bird	3.132562

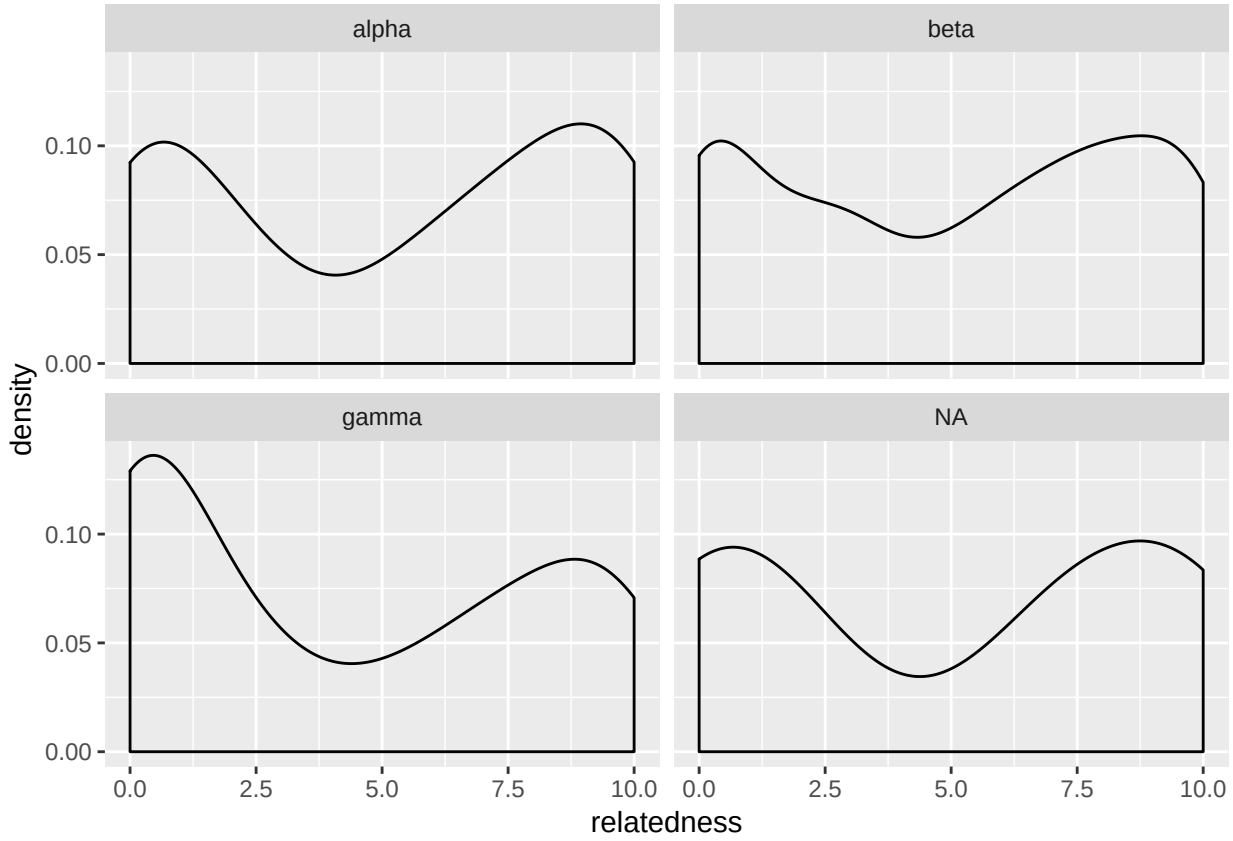
Relatedness rating frequencies per participant

```
ggplot(df, aes(x = relatedness, colour = level)) +
  geom_bar() +
  facet_wrap(~id)
```



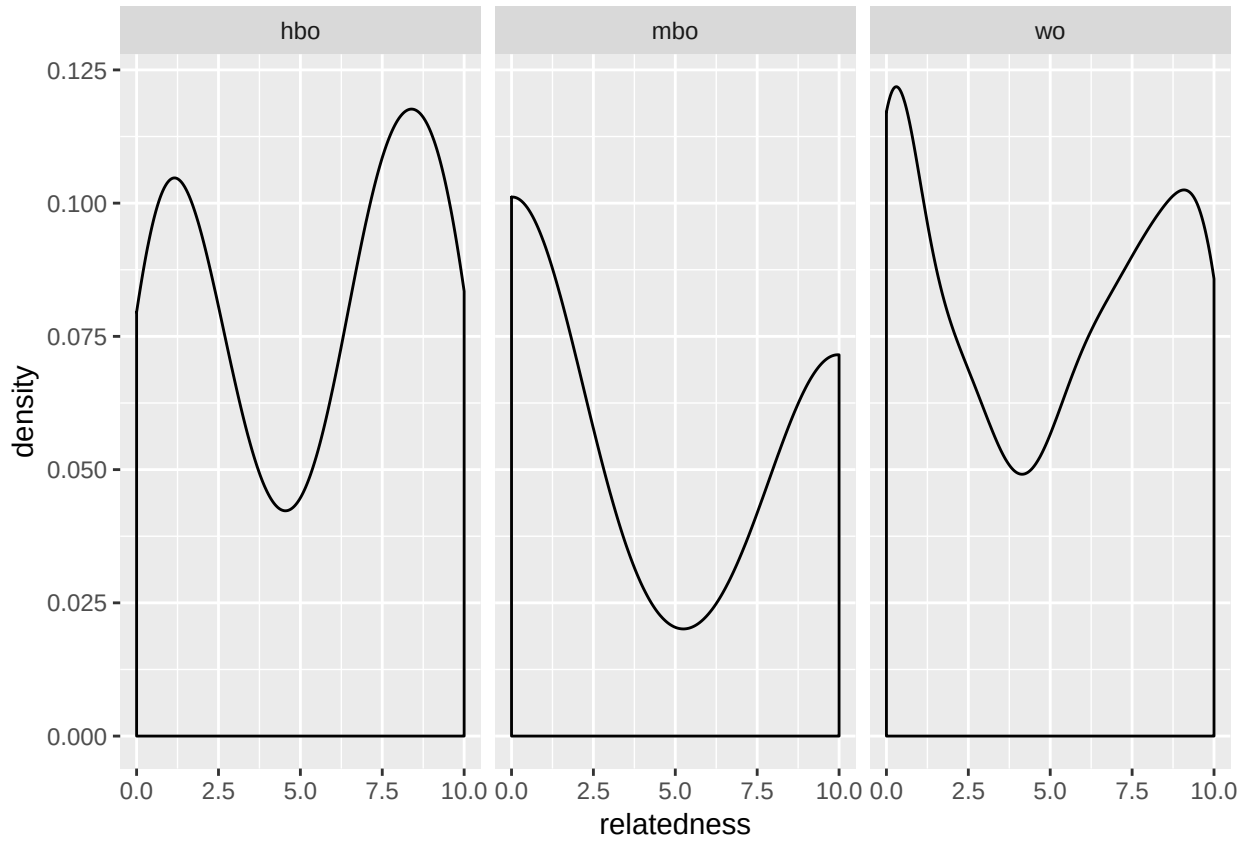
Relatedness distribution per `study_type`

```
ggplot(df, aes(x = relatedness)) +
  geom_density() +
  facet_wrap(~study_type)
```



Per `level`

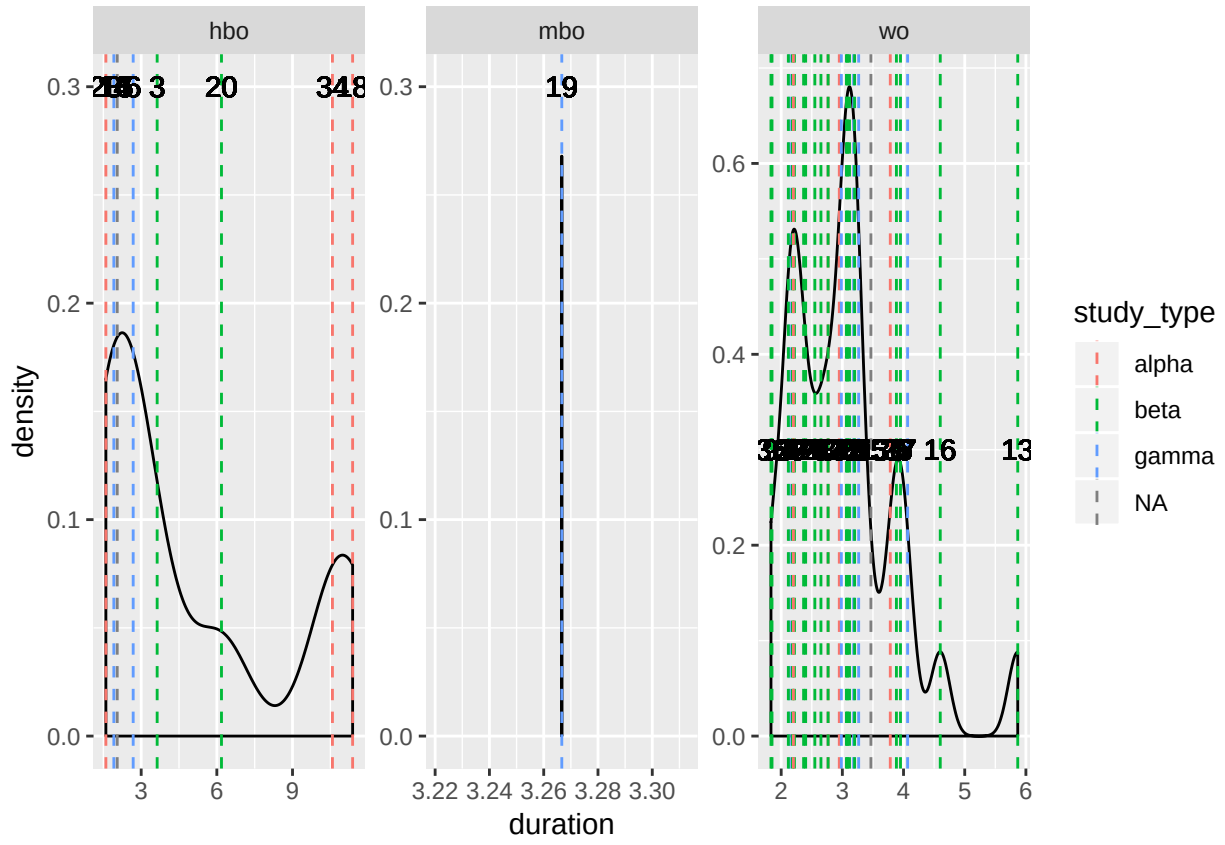
```
ggplot(df, aes(x = relatedness)) +
  geom_density() +
  facet_wrap(~level)
```



Duration denisty

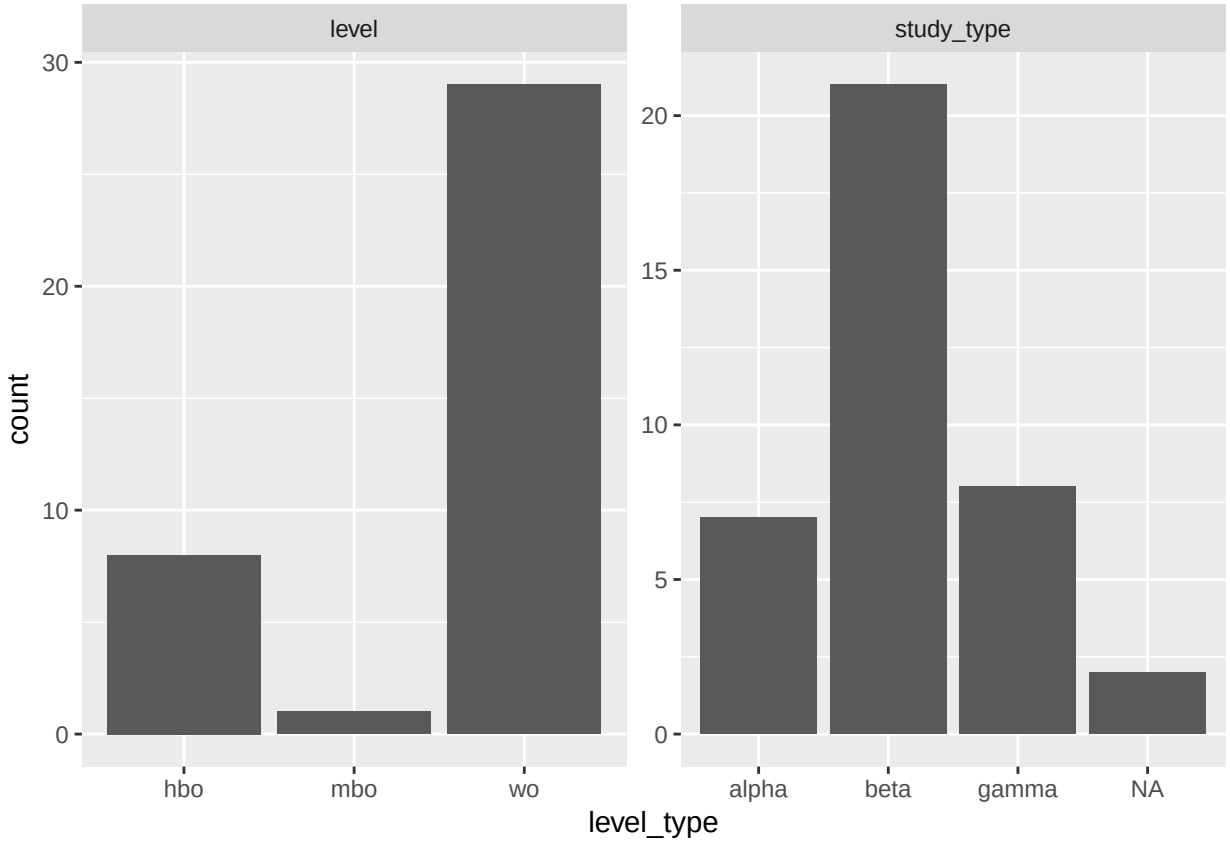
```
ggplot(df, aes(duration)) +
  geom_density() +
  geom_vline(aes(xintercept = duration, colour = study_type), linetype = "dashed") +
  geom_text(aes(y = .3, label = id)) +
  facet_wrap(~level, scales = "free")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



Distributions of participants among different groups

```
df %>%
  group_by(id, study_type, level) %>%
  summarise %>%
  gather(key = grouping_var, value = level_type, -id) %>%
  ggplot(aes(level_type)) +
  geom_bar() +
  facet_wrap(~grouping_var, scales = "free")
```



Does a model that includes `study_type` explain the variance better than one that does not?

We will use an `lmer` model from the `lme4` package, because we we are interested in the effects of `study_type`, and not the individual differences that we can not control for. The intercept of each individual (`id`) will be defined as a random effect. We will leave out the `NA`'s.

Furthermore, `word_pair` will be defined as a random effect. At this moment we are mainly interested in whether `study_type` explains differences in relatedness in general. We don't want to look at the effect of `study_type` on the relatedness ratings of each `word_pair`.

The null-model

The null-model only includes the intercept for each `word_pair` and the intercept of each individual as a random effects.

```
df_naomit <- df %>% na.omit

model0 <- lmer(relatedness ~ (1|word_pair) + (1|id), data = df_naomit, REML = F)
```

The alternative model (includes `study_type`)

In the table we can see that there is a relatively large difference between the ratings of people with a `gamma` background and those with an `alpha background`. `estimate` > `(Intercept)` gives us the relatedness intercept of relatedness of `people` with an `alpha` background. The `estimate` value at `study_typebeta` gives the difference with this intercept. The same holds for `study_typegamma`.

```
model1 <- lmer(relatedness ~ study_type + (1|word_pair) + (1|id), data = df_naomit, REML = F)

model1 %>%
  tidy() %>%
  rename("t-value" = statistic) %>%
  select(-group) %>%
  kable()
```

```
## Warning in bind_rows_(x, .id): binding factor and character vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

term	estimate	std.error	t-value
(Intercept)	5.1809524	0.6147807	8.4273179
study_typebeta	-0.1349206	0.2751886	-0.4902842
study_typegamma	-0.9517857	0.3263336	-2.9166038
sd_(Intercept).id	0.5345427	NA	NA
sd_(Intercept).word_pair	3.1039912	NA	NA
sd_Observation.Residual	1.8317224	NA	NA

Does the alternate model explain significantly more of the variance?

The p-value indicates that a model that includes `study_type` explains significantly more of the variance.

```
anova(model0, model1)
```

```
## Data: df_naomit
## Models:
## model0: relatedness ~ (1 | word_pair) + (1 | id)
## model1: relatedness ~ study_type + (1 | word_pair) + (1 | id)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## model0  4 4574.1 4594.1 -2283.1   4566.1
## model1  6 4568.1 4598.0 -2278.1   4556.1 9.9942      2   0.006758 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How about `level` and `duration` and combinations of those and `study_type`?

```
model2 <- lmer(relatedness ~ level + (1|word_pair) + (1|id), data = df_naomit, REML = F)
model3 <- lmer(relatedness ~ duration + (1|word_pair) + (1|id), data = df_naomit, REML = F)
```

`level`

```
anova(model0, model2)
```

```
## Data: df_naomit
## Models:
## model0: relatedness ~ (1 | word_pair) + (1 | id)
## model2: relatedness ~ level + (1 | word_pair) + (1 | id)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## model0  4 4574.1 4594.1 -2283.1   4566.1
## model2  6 4576.4 4606.3 -2282.2   4564.4 1.6782      2     0.4321
```

duration

```
anova(model0, model3)
```

```
## Data: df_naomit
## Models:
## model0: relatedness ~ (1 | word_pair) + (1 | id)
## model3: relatedness ~ duration + (1 | word_pair) + (1 | id)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## model0  4 4574.1 4594.1 -2283.1   4566.1
## model3  5 4576.1 4601.0 -2283.0   4566.1 0.0275      1     0.8683
```

Models including `level` xor `duration` do not explain significantly more of the variance than models that do not include these.

Correlations

Between participants

```
corr <- df %>%
  arrange(id) %>%
  select(id, relatedness, word_pair) %>%
  spread(key = id, value = relatedness) %>%
  select(-word_pair) %>%
  cor() %>%
  as.data.frame() %>%
  rownames_to_column("id") %>%
  gather(key = participant2, value = correlation, -id) %>%
  # Add duration of test, study type and level of participant indicated by id:
  left_join(df %>% select(id, study_type, level, duration)) %>%
  rename(study_type1 = study_type,
         level1 = level) %>%
  left_join(by = c("participant2" = "id"), df %>% select(id, study_type, level, duration)) %>%
  rename(study_type2 = study_type,
         level2 = level) %>%
  # Correlations between participant and himself are not interesting,
  # so filter them out:
  filter(id != participant2)
```

```
## Joining, by = "id"
```

Are there any correlations between participants higher than .9? And lower tha .5?

```
corr %>%
  filter(correlation > .9) %>%
  group_by(id, participant2, correlation, study_type1, level1, study_type2, level2) %>%
  summarise() %>%
  kable()
```

id	participant2	correlation	study_type1	level1	study_type2	level2
1	38	0.9105724	beta	wo	gamma	wo
1	9	0.9235667	beta	wo	beta	wo
10	7	0.9072691	gamma	wo	gamma	wo
2	36	0.9248204	alpha	wo	beta	wo
2	7	0.9004022	alpha	wo	gamma	wo
2	9	0.9380832	alpha	wo	beta	wo

id	participant2	correlation	study_type1	level1	study_type2	level2
20	33	0.9314128	beta	hbo	beta	wo
25	26	0.9105942	gamma	wo	beta	wo
25	31	0.9154552	gamma	wo	gamma	wo
26	25	0.9105942	beta	wo	gamma	wo
26	3	0.9029268	beta	wo	beta	hbo
26	31	0.9088292	beta	wo	gamma	wo
26	34	0.9201829	beta	wo	alpha	hbo
26	36	0.9281192	beta	wo	beta	wo
26	9	0.9066503	beta	wo	beta	wo
27	29	1.0000000	beta	wo	beta	wo
29	27	1.0000000	beta	wo	beta	wo
3	26	0.9029268	beta	hbo	beta	wo
3	34	0.9105397	beta	hbo	alpha	hbo
3	7	0.9068298	beta	hbo	gamma	wo
31	25	0.9154552	gamma	wo	gamma	wo
31	26	0.9088292	gamma	wo	beta	wo
31	38	0.9147887	gamma	wo	gamma	wo
33	20	0.9314128	beta	wo	beta	hbo
34	26	0.9201829	alpha	hbo	beta	wo
34	3	0.9105397	alpha	hbo	beta	hbo
34	9	0.9436587	alpha	hbo	beta	wo
36	2	0.9248204	beta	wo	alpha	wo
36	26	0.9281192	beta	wo	beta	wo
36	7	0.9070593	beta	wo	gamma	wo
36	9	0.9188863	beta	wo	beta	wo
37	38	0.9055737	beta	wo	gamma	wo
38	1	0.9105724	gamma	wo	beta	wo
38	31	0.9147887	gamma	wo	gamma	wo
38	37	0.9055737	gamma	wo	beta	wo
38	9	0.9116106	gamma	wo	beta	wo
7	10	0.9072691	gamma	wo	gamma	wo
7	2	0.9004022	gamma	wo	alpha	wo
7	3	0.9068298	gamma	wo	beta	hbo
7	36	0.9070593	gamma	wo	beta	wo
9	1	0.9235667	beta	wo	beta	wo
9	2	0.9380832	beta	wo	alpha	wo
9	26	0.9066503	beta	wo	beta	wo
9	34	0.9436587	beta	wo	alpha	hbo
9	36	0.9188863	beta	wo	beta	wo
9	38	0.9116106	beta	wo	gamma	wo

```
corr %>%
  filter(correlation < .5) %>%
  group_by(id, participant2, correlation, study_type1, level1, study_type2, level2) %>%
  summarise() %>%
  kable()
```

id	participant2	correlation	study_type1	level1	study_type2	level2
12	22	0.4552611	beta	wo	beta	wo
14	19	0.4886006	gamma	hbo	gamma	mbo
19	14	0.4886006	gamma	mbo	gamma	hbo

id	participant2	correlation	study_type1	level1	study_type2	level2
19	6	0.4429555	gamma	mbo	gamma	hbo
19	8	0.4936313	gamma	mbo	beta	wo
22	12	0.4552611	beta	wo	beta	wo
23	6	0.4880715	alpha	hbo	gamma	hbo
6	19	0.4429555	gamma	hbo	gamma	mbo
6	23	0.4880715	gamma	hbo	alpha	hbo
8	19	0.4936313	beta	wo	gamma	mbo

Hey comment hier: Is dit informatief? Volgens mij schieten we niet zo veel op met deze info:

Correlation of mean relatedness ratings between `study_type`

```
df %>%
  group_by(word_pair, study_type) %>%
  summarise(mean = mean(relatedness)) %>%
  spread(key = study_type, value = mean) %>%
  ungroup() %>%
  select(-word_pair) %>%
  cor() %>%
  kable()
```

	alpha	beta	gamma	
alpha	1.0000000	0.9757188	0.9629684	0.8973075
beta	0.9757188	1.0000000	0.9642089	0.9188923
gamma	0.9629684	0.9642089	1.0000000	0.8736004
	0.8973075	0.9188923	0.8736004	1.0000000

Correlation of mean relatedness ratings between `level`

```
df %>%
  group_by(word_pair, level) %>%
  summarise(mean = mean(relatedness)) %>%
  spread(key = level, value = mean) %>%
  ungroup() %>%
  select(-word_pair) %>%
  cor() %>%
  kable()
```

	hbo	mbo	wo
hbo	1.0000000	0.6937382	0.9806499
mbo	0.6937382	1.0000000	0.7405557
wo	0.9806499	0.7405557	1.0000000

```
#Bekijk correlatie/scores tussen groepen alpha, beta en gamma?
#Bekijk correlatie/scores tussen wo, hbo en work?
#Bekijk algemeen wat de gemiddelde scores zijn?
```

```
#Gebruik t.test om te kijken of er daadwerkelijk verschil is.
```