Results_WordSim353

Fleur Petit

11 February 2019

Function to load the data

```
loadData <- function() {
  files <- list.files(file.path("./results"), full.names = TRUE)
  data <- do.call(rbind, lapply(files, read.csv)) %>%
    drop_na()
  as_tibble(data)
}
```

Table with all the results

```
df <- loadData()
kable(t(df))
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
tournament_avenue	0.0	5.1	2.6	1.0	6.1	0.7	1.2	0.9	5.0	0.0	0.0	2.1	0.8	0.0
rabbi_minister	0.0	8.2	0.5	4.9	3.0	7.9	7.7	7.4	3.5	0.0	8.0	0.0	2.2	0.9
doctor_scientist	4.0	8.6	8.9	10.0	6.0	7.0	9.0	6.0	8.6	8.3	6.0	7.0	7.7	3.0
smile_psychology	3.0	5.7	7.3	8.0	4.0	2.0	2.5	5.0	7.2	8.3	8.0	7.0	3.9	7.2
egg_peace	0.0	0.0	2.9	2.0	0.0	0.4	2.2	6.0	0.0	1.7	0.0	0.0	1.1	0.0
Mars_health	0.0	0.0	8.9	0.0	0.0	0.0	0.5	1.0	5.0	0.0	0.0	0.0	3.2	0.0
season_astronomer	5.0	5.0	7.3	1.0	0.0	0.0	0.4	7.0	7.0	5.0	5.0	1.0	6.9	0.0
hardware_nation	0.0	1.6	2.1	6.0	0.0	0.0	1.6	0.9	2.6	0.0	0.0	0.0	0.5	0.0
flight_departure	9.0	8.0	10.0	8.5	6.4	10.0	8.3	8.1	9.0	10.0	10.0	10.0	9.5	10.0
basketball_championship	9.0	8.6	10.0	8.2	3.8	6.9	8.7	9.1	9.0	8.6	8.0	8.0	9.4	2.0
forecast_concert	0.0	3.2	5.9	0.0	7.5	2.0	0.0	1.0	0.4	0.2	2.0	1.0	1.3	0.0
street_challenge	0.0	3.2	1.4	1.5	1.2	7.2	2.5	4.0	3.6	0.9	0.0	0.8	2.7	0.1
consumer_withdrawal	0.0	6.9	5.0	8.0	3.0	2.0	2.5	1.0	2.0	0.0	2.0	6.0	2.0	1.0
insurance_automobile	9.0	7.8	8.4	8.5	2.0	8.0	7.4	9.0	7.7	7.5	8.5	9.0	8.7	3.0
mile_keyboard	0.0	0.1	1.9	0.0	0.0	0.0	1.3	0.0	1.3	0.0	0.0	1.0	1.7	0.0
deployment_infrastructure	0.0	5.0	0.7	6.0	2.0	0.0	3.9	8.0	2.1	7.2	3.0	5.0	3.2	0.2
hardware_infrastructure	5.0	5.0	1.8	7.6	2.2	0.0	8.2	8.1	1.8	9.1	7.6	3.0	5.9	0.0
loss_recovery	7.0	7.3	9.1	8.0	6.0	2.0	8.9	8.0	9.1	5.0	9.5	8.0	7.2	6.5
health_fertility	5.0	8.5	9.2	7.0	7.9	3.1	9.0	9.2	9.0	7.8	9.2	10.0	7.4	2.9
landscape_fauna	9.0	9.2	6.6	8.0	5.7	6.4	4.1	9.0	9.1	6.9	9.0	9.0	8.0	10.0
card_credit	9.0	8.3	9.1	7.9	7.0	9.0	1.9	9.6	8.5	9.3	9.0	9.0	5.6	1.5
food_water	7.5	8.4	8.8	8.0	3.9	6.5	8.0	9.1	8.2	10.0	9.0	7.0	8.3	9.0
software_accommodation	0.0	2.7	3.0	3.0	5.5	0.3	0.0	2.0	3.1	0.0	0.0	1.0	0.4	1.5
competition_championship	9.0	7.5	8.7	10.0	8.0	9.0	8.7	9.1	9.0	8.9	9.0	10.0	8.5	10.0
love_glass	0.0	0.8	3.3	3.5	0.5	0.0	3.0	3.0	1.3	0.7	0.5	0.0	0.5	0.0
live_book	5.0	7.3	3.5	7.5	0.0	0.4	3.6	0.5	8.3	0.0	0.1	1.0	3.8	0.0
basketball_game	9.0	8.8	10.0	9.0	6.0	9.0	10.0	8.0	9.6	9.0	9.5	10.0	9.3	7.4
stupid_furnace	0.0	1.1	7.0	2.0	0.0	0.0	6.7	3.0	1.7	0.0	0.0	0.0	0.9	0.0
moon_planet	8.9	7.0	9.4	9.0	7.2	10.0	10.0	9.0	6.8	10.0	9.5	9.0	7.1	10.0
psychiatry_price	0.0	2.3	2.2	2.0	5.0	1.0	1.0	9.5	5.9	1.4	0.2	0.0	2.7	0.0

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

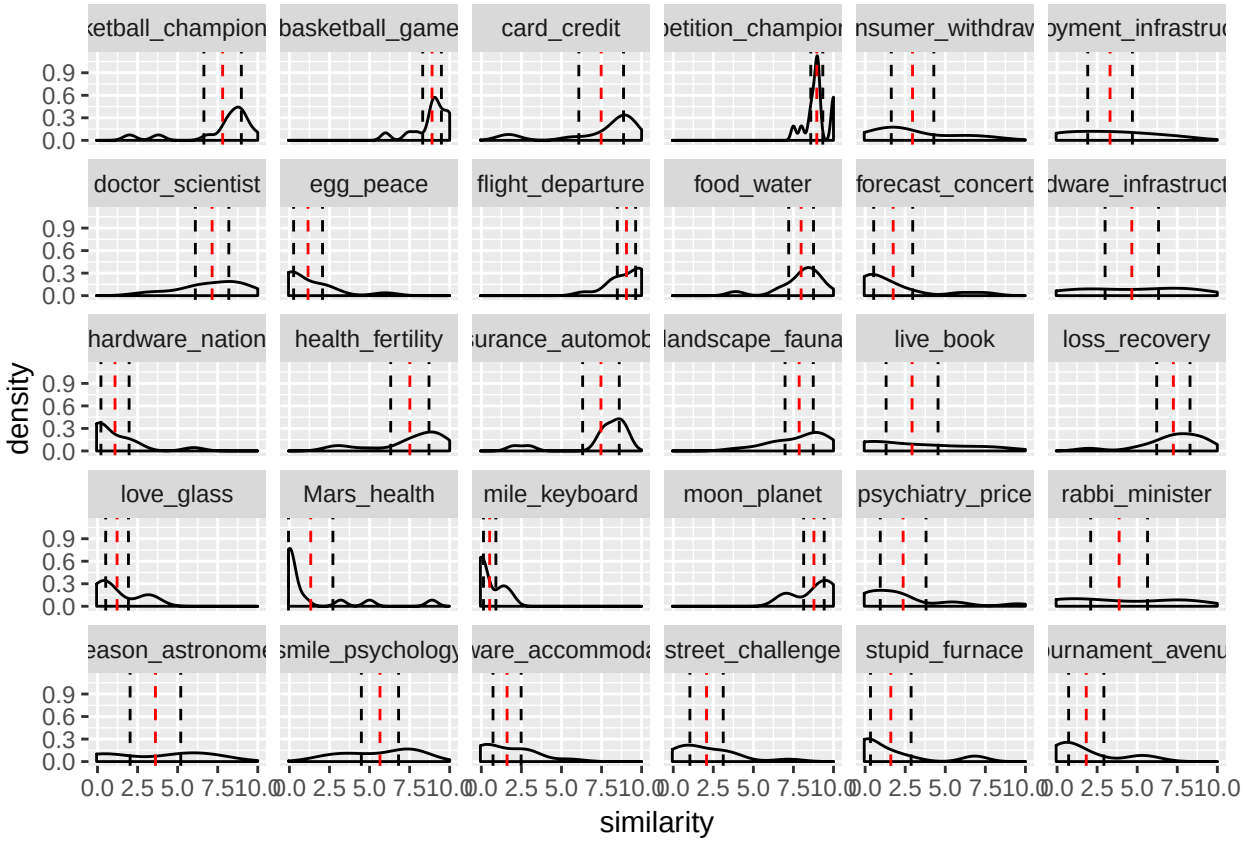
```
description <-
  df %>%
  gather(key = "word_pair", value = "similarity") %>%
  group_by(word_pair) %>%
  mutate(mean = mean(similarity),
             sd = sd(similarity),
             lower = ci(mean,sd,nrow(df))[["lower"]],
             upper = ci(mean,sd,nrow(df))[["upper"]],
             median = median(similarity)
             )

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

word_pair	mean	sd	lower	upper	median
basketball_championship	7.8071429	2.2317514	6.6381015	8.9761842	8.60
basketball_game	8.9000000	1.1149336	8.3159728	9.4840272	9.00
card_credit	7.4785714	2.6634791	6.0833814	8.8737615	8.75
competition_championship	8.9571429	0.7165240	8.5818115	9.3324742	9.00
consumer_withdrawal	2.9571429	2.5333815	1.6301008	4.2841849	2.00
deployment_infrastructure	3.3071429	2.6618283	1.9128176	4.7014681	3.10
doctor_scientist	7.1500000	1.9921963	6.1064429	8.1935571	7.35
egg_peace	1.1642857	1.7216303	0.2624572	2.0661143	0.20
flight_departure	9.0571429	1.0924315	8.4849028	9.6293830	9.25
food_water	7.9785714	1.4791778	7.2037449	8.7533979	8.25
forecast_concert	1.7500000	2.3256926	0.5317501	2.9682499	1.00
hardware_infrastructure	4.6642857	3.1832753	2.9968148	6.3317567	5.00
hardware_nation	1.0928571	1.6785819	0.2135783	1.9721360	0.25
health_fertility	7.5142857	2.2819887	6.3189289	8.7096425	8.20
insurance_automobile	7.4642857	2.1816039	6.3215127	8.6070587	8.20
landscape_fauna	7.8571429	1.6777798	6.9782842	8.7360015	8.50
live_book	2.9285714	3.0937725	1.3079840	4.5491588	2.25
loss_recovery	7.2571429	1.9832818	6.2182554	8.2960303	7.65
love_glass	1.2214286	1.3537291	0.5123149	1.9305422	0.60
Mars_health	1.3285714	2.6455852	-0.0572454	2.7143882	0.00
mile_keyboard	0.5214286	0.7381667	0.1347603	0.9080968	0.00
moon_planet	8.7785714	1.2223424	8.1382811	9.4188618	9.00
psychiatry_price	2.3714286	2.7280392	0.9424205	3.8004366	1.70
rabbi_minister	3.8714286	3.3858140	2.0978633	5.6449938	3.25
season_astronomer	3.6142857	3.0150538	2.0349330	5.1936385	5.00

word_pair	mean	sd	lower	upper	median
smile_psychology	5.6500000	2.2169799	4.4886962	6.8113038	6.35
software_accommodation	1.6071429	1.6753710	0.7295460	2.4847398	1.25
street_challenge	2.0785714	1.9846250	1.0389804	3.1181625	1.45
stupid_furnace	1.6000000	2.4172457	0.3337925	2.8662075	0.45
tournament_avenue	1.8214286	2.0998823	0.7214632	2.9213940	0.95

Density plots with mean and 95% confidence interval

```
ggplot(description, aes(similarity)) +
  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")
```



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
consumer_withdrawal	2.533381
Mars_health	2.645585
deployment_infrastructure	2.661828
card_credit	2.663479
psychiatry_price	2.728039
season_astronomer	3.015054
live_book	3.093773
hardware_infrastructure	3.183275
rabbi_minister	3.385814