

Simple Corpus Stats

Chris Bentz

10/01/2023

Load libraries

If the libraries are not installed yet, you need to install them using, for example, the command: `install.packages("ggplot2")`.

```
library(stringr)
library(ggplot2)
library(plyr)
```

List files

List all the files in the directory “corpus”.

```
file.list <- list.files(path = "~/Github/NaLaFi/data/",
                        recursive = T, full.names = T)
#print(file.list)
length(file.list)

## [1] 244
```

Count number of files in writing and non-writing subfolders

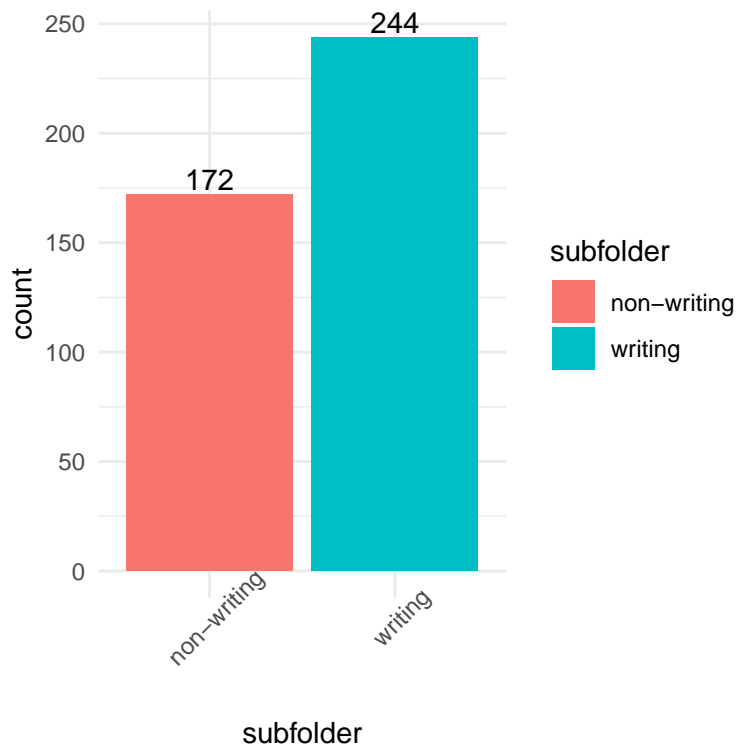
Count how many files are in each of the highest level subfolders of “data”, and create a dataframe with counts.

```
#number of "animal" files
writing.count <- length(file.list[grepl("writing", file.list)])
nonwriting.count <- length(file.list[grepl("non-writing", file.list)])

#create data frame
df <- data.frame(subfolder = c("writing", "non-writing"),
                  count = c(writing.count, nonwriting.count))
```

Create a bar plot with counts.

```
counts.plot <- ggplot(data = df, aes(x = subfolder, y = count, fill = subfolder)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = count), vjust = -0.2, color = "black", size = 4) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
counts.plot
```



Safe figure to file

```
ggsave("~/Github/NaLaFi/figures/simpleStats_writing.pdf", counts.plot, dpi = 300,  
        scale = 1, device = cairo_pdf)
```

```
## Saving 4 x 4 in image
```

Count number of files in subfolders

Count how many files are in each of the highest level subfolders of “data”, and create a dataframe with counts.

```
#number of "animal" files  
animal.count <- length(file.list[grepl("animal", file.list)])  
ancient.count <- length(file.list[grepl("ancient", file.list)])  
heraldics.count <- length(file.list[grepl("heraldics", file.list)])  
morse.count <- length(file.list[grepl("morse", file.list)])  
natural.count <- length(file.list[grepl("natural", file.list)])  
pycode.count <- length(file.list[grepl("pycode", file.list)])  
random.count <- length(file.list[grepl("random", file.list)])  
signlang.count <- length(file.list[grepl("signlang", file.list)])  
shuffled.count <- length(file.list[grepl("shuffled", file.list)])  
udhr.count <- length(file.list[grepl("udhr", file.list)])  
weather.count <- length(file.list[grepl("weather", file.list)])  
  
#create data frame  
df <- data.frame(subfolder = c("animal", "ancient", "heraldics", "morse",  
                              "natural", "pycode", "random", "signlang",  
                              "shuffled", "udhr", "weather"))
```

```

        "shuffled", "udhr", "weather"),
count = c(animal.count, ancient.count, heraldics.count,
          morse.count, natural.count, pycode.count,
          random.count, signlang.count, shuffled.count,
          udhr.count, weather.count))

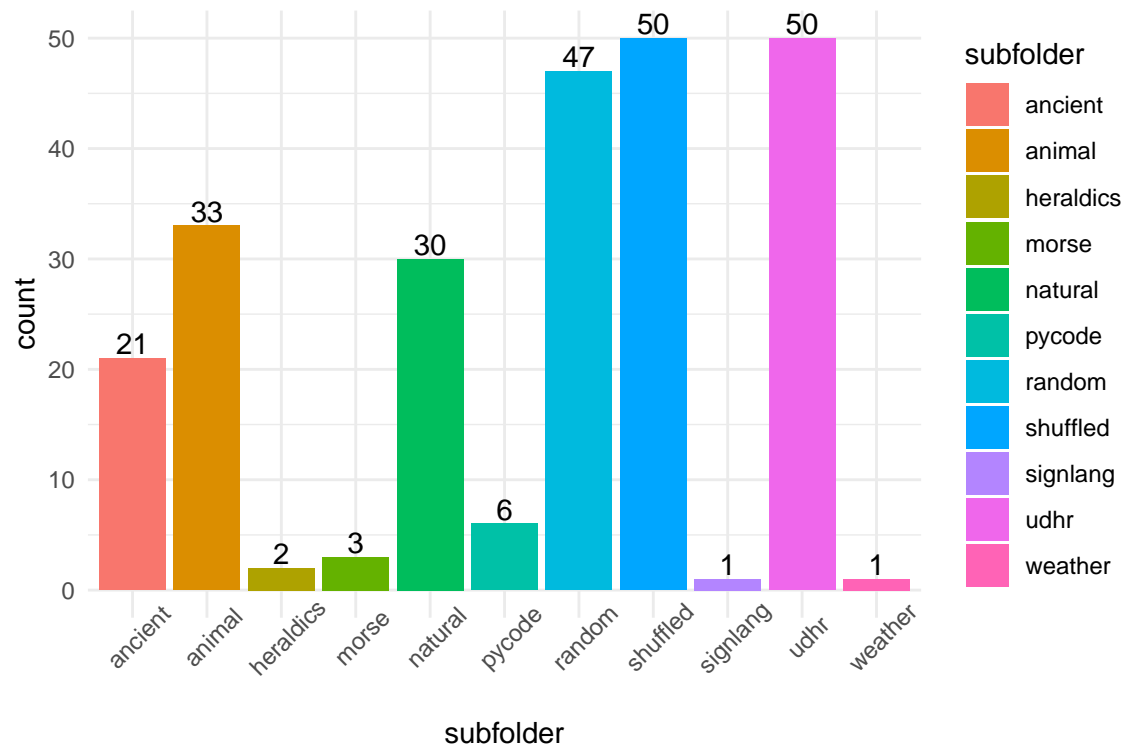
```

Create a bar plot with counts.

```

counts.plot <- ggplot(data = df, aes(x = subfolder, y = count, fill = subfolder)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = count), vjust = -0.2, color = "black", size = 4) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
counts.plot

```



Safe figure to file

```

ggsave("~/Github/NaLaFi/figures/simpleStats_counts.pdf", counts.plot, dpi = 300,
        scale = 1, device = cairo_pdf)

```

Saving 6 x 4.5 in image

Lengths of files in characters

Read files and count characters

```
# set counter
counter = 0
# initialize dataframe to append results to
simpleStats.df <- data.frame(filename = character(0), subcorpus = character(0),
                             num.lines = numeric(0), num.chars = numeric(0))

for (file in file.list)
{
  # loading textfile
  textfile <- scan(file, what = "char", quote = "", comment.char = "",
                   encoding = "UTF-8", sep = "\n", skip = 7)
  textfile <- gsub("\t","",textfile) # remove tabs
  textfile <- gsub("<.*>","",textfile) # remove annotations marked by '<>'
  #print(head(textfile))
  # get filename
  filename <- basename(file)
  #print(filename) # for visual inspection
  # get subcorpus category
  subcorpus <- sub("_.*", "", filename)
  #print(subcorpus) # for visual inspection
  # count number of lines in text file
  num.lines <- length(textfile)
  # count the number of utf-8 characters in text file (note that this includes white
  # spaces)
  num.chars <- sum(nchar(textfile, type = "chars"))
  #print(num.chars) # for visual inspection
  # append results to dataframe
  local.df <- data.frame(filename, subcorpus, num.lines, num.chars)
  simpleStats.df <- rbind(simpleStats.df, local.df)
  # counter
  counter <- counter + 1
  #print(counter)
}
simpleStats.df
```

##	filename	subcorpus	num.lines	num.chars
## 1	animal_bhg_0001.txt	animal	1	765
## 2	animal_bhg_0002.txt	animal	1	272
## 3	animal_bhg_0003.txt	animal	1	734
## 4	animal_bhg_0004.txt	animal	1	341
## 5	animal_bhg_0005.txt	animal	1	182
## 6	animal_bhg_0006.txt	animal	1	1598
## 7	animal_bhg_0007.txt	animal	1	1118
## 8	animal_bhg_0008.txt	animal	1	995
## 9	animal_bhg_0009.txt	animal	1	213
## 10	animal_bhg_0010.txt	animal	1	426
## 11	animal_cad_0001.txt	animal	6	36
## 12	animal_cad_0002.txt	animal	17	151
## 13	animal_cav_0001.txt	animal	1	407

## 14	animal_cav_0002.txt	animal	1	636
## 15	animal_cav_0003.txt	animal	1	176
## 16	animal_cav_0004.txt	animal	1	23
## 17	animal_cav_0005.txt	animal	1	167
## 18	animal_cav_0006.txt	animal	1	116
## 19	animal_cav_0007.txt	animal	1	32
## 20	animal_cav_0008.txt	animal	1	161
## 21	animal_cav_0009.txt	animal	1	116
## 22	animal_cth_0001.txt	animal	1	287
## 23	animal_cth_0002.txt	animal	1	871
## 24	animal_cth_0003.txt	animal	1	391
## 25	animal_cth_0004.txt	animal	1	779
## 26	animal_cth_0005.txt	animal	1	1533
## 27	animal_cth_0006.txt	animal	1	608
## 28	animal_cth_0007.txt	animal	1	560
## 29	animal_cth_0008.txt	animal	1	2288
## 30	animal_cth_0009.txt	animal	1	1440
## 31	animal_cth_0010.txt	animal	1	4492
## 32	animal_cth_0011.txt	animal	1	604
## 33	animal_zfi_0001.txt	animal	3	31
## 34	heraldics_bla_0001.txt	heraldics	503	44198
## 35	heraldics_bla_0002.txt	heraldics	207	13932
## 36	morse_moc_0001.txt	morse	1	1291
## 37	morse_moc_0002.txt	morse	3	2257
## 38	morse_moc_0003.txt	morse	1	175
## 39	natural_dna_0001.txt	natural	157	10958
## 40	natural_dna_0002.txt	natural	209	14621
## 41	natural_dna_0003.txt	natural	713	49910
## 42	natural_dna_0004.txt	natural	713	49910
## 43	natural_dna_0005.txt	natural	713	49910
## 44	natural_dna_0006.txt	natural	713	49910
## 45	natural_dna_0007.txt	natural	713	49910
## 46	natural_dna_0008.txt	natural	713	49910
## 47	natural_dna_0009.txt	natural	713	49910
## 48	natural_dna_0010.txt	natural	713	49910
## 49	natural_dna_0011.txt	natural	713	49910
## 50	natural_dna_0012.txt	natural	713	49910
## 51	natural_dna_0013.txt	natural	713	49910
## 52	natural_dna_0014.txt	natural	713	49910
## 53	natural_dna_0015.txt	natural	713	49910
## 54	natural_dna_0016.txt	natural	713	49910
## 55	natural_dna_0017.txt	natural	713	49910
## 56	natural_dna_0018.txt	natural	713	49910
## 57	natural_dna_0019.txt	natural	713	49910
## 58	natural_dna_0020.txt	natural	713	49910
## 59	natural_dna_0021.txt	natural	713	49910
## 60	natural_dna_0022.txt	natural	713	49910
## 61	natural_dna_0023.txt	natural	713	49910
## 62	natural_dna_0024.txt	natural	713	49910
## 63	natural_dna_0025.txt	natural	713	49910
## 64	natural_dna_0026.txt	natural	713	49910
## 65	natural_dna_0027.txt	natural	713	49910
## 66	natural_dna_0028.txt	natural	713	49910
## 67	natural_dna_0029.txt	natural	713	49910

## 68	natural_dna_0030.txt	natural	713	49910
## 69	pycode_pyc_0001.txt	pycode	83	2515
## 70	pycode_pyc_0002.txt	pycode	25	570
## 71	pycode_pyc_0003.txt	pycode	9	136
## 72	pycode_pyc_0004.txt	pycode	43	1064
## 73	pycode_pyc_0005.txt	pycode	67	1370
## 74	pycode_pyc_0006.txt	pycode	34	884
## 75	random_ran_10	random	1	1000
## 76	random_ran_11	random	1	1000
## 77	random_ran_12	random	1	1000
## 78	random_ran_13	random	1	1000
## 79	random_ran_14	random	1	1000
## 80	random_ran_15	random	1	1000
## 81	random_ran_16	random	1	1000
## 82	random_ran_17	random	1	1000
## 83	random_ran_18	random	1	1000
## 84	random_ran_19	random	1	1000
## 85	random_ran_2	random	1	1000
## 86	random_ran_20	random	1	1000
## 87	random_ran_21	random	1	1000
## 88	random_ran_22	random	1	1000
## 89	random_ran_23	random	1	1000
## 90	random_ran_24	random	1	1000
## 91	random_ran_25	random	1	1000
## 92	random_ran_26	random	1	1000
## 93	random_ran_27	random	1	1000
## 94	random_ran_28	random	1	1000
## 95	random_ran_29	random	1	1000
## 96	random_ran_3	random	1	1000
## 97	random_ran_30	random	1	1000
## 98	random_ran_31	random	1	1000
## 99	random_ran_32	random	1	1000
## 100	random_ran_33	random	1	1000
## 101	random_ran_34	random	1	1000
## 102	random_ran_35	random	1	1000
## 103	random_ran_36	random	1	1000
## 104	random_ran_37	random	1	1000
## 105	random_ran_38	random	1	1000
## 106	random_ran_39	random	1	1000
## 107	random_ran_4	random	1	1000
## 108	random_ran_40	random	1	1000
## 109	random_ran_41	random	1	1000
## 110	random_ran_42	random	1	1000
## 111	random_ran_43	random	1	1000
## 112	random_ran_44	random	1	1000
## 113	random_ran_45	random	1	1000
## 114	random_ran_46	random	1	1000
## 115	random_ran_47	random	1	1000
## 116	random_ran_48	random	1	1000
## 117	random_ran_5	random	1	1000
## 118	random_ran_6	random	1	1000
## 119	random_ran_7	random	1	1000
## 120	random_ran_8	random	1	1000
## 121	random_ran_9	random	1	1000

## 122	shuffled_aii_0001	shuffled	1	1000
## 123	shuffled_arb_0001	shuffled	1	1000
## 124	shuffled_azj_0001	shuffled	1	1000
## 125	shuffled_azj_0002	shuffled	1	1000
## 126	shuffled_ben_0001	shuffled	1	1000
## 127	shuffled_blt_0001	shuffled	1	1000
## 128	shuffled_bod_0001	shuffled	1	1000
## 129	shuffled_bos_0001	shuffled	1	1000
## 130	shuffled_bos_0002	shuffled	1	1000
## 131	shuffled_chr_0001	shuffled	1	1000
## 132	shuffled_cmn_0001	shuffled	1	1000
## 133	shuffled_cmn_0002	shuffled	1	1000
## 134	shuffled_csw_0001	shuffled	1	1000
## 135	shuffled_div_0001	shuffled	1	1000
## 136	shuffled_ell_0001	shuffled	1	1000
## 137	shuffled_eng_0001	shuffled	1	1000
## 138	shuffled_epo_0001	shuffled	1	1000
## 139	shuffled_eus_0001	shuffled	1	1000
## 140	shuffled_gaz_0001	shuffled	1	1000
## 141	shuffled_guj_0001	shuffled	1	1000
## 142	shuffled_heb_0001	shuffled	1	1000
## 143	shuffled_hin_0001	shuffled	1	1000
## 144	shuffled_hye_0001	shuffled	1	1000
## 145	shuffled_ibt_0001	shuffled	1	1000
## 146	shuffled_iii_0001	shuffled	1	1000
## 147	shuffled_ike_0001	shuffled	1	1000
## 148	shuffled_jav_0001	shuffled	1	1000
## 149	shuffled_jav_0002	shuffled	1	1000
## 150	shuffled_jpn_0001	shuffled	1	1000
## 151	shuffled_kal_0001	shuffled	1	1000
## 152	shuffled_kan_0001	shuffled	1	1000
## 153	shuffled_kat_0001	shuffled	1	1000
## 154	shuffled_khm_0001	shuffled	1	1000
## 155	shuffled_kkh_0001	shuffled	1	1000
## 156	shuffled_kor_0001	shuffled	1	1000
## 157	shuffled_lao_0001	shuffled	1	1000
## 158	shuffled_lug_0001	shuffled	1	1000
## 159	shuffled_mal_0001	shuffled	1	1000
## 160	shuffled_mya_0001	shuffled	1	1000
## 161	shuffled_pan_0001	shuffled	1	1000
## 162	shuffled_rus_0001	shuffled	1	1000
## 163	shuffled_sin_0001	shuffled	1	1000
## 164	shuffled_tam_0001	shuffled	1	1000
## 165	shuffled_tel_0001	shuffled	1	1000
## 166	shuffled_tgl_0001	shuffled	1	1000
## 167	shuffled_tha_0001	shuffled	1	1000
## 168	shuffled_tir_0001	shuffled	1	1000
## 169	shuffled_vai_0001	shuffled	1	1000
## 170	shuffled_zgh_0001	shuffled	1	1000
## 171	shuffled_zul_0001	shuffled	1	1000
## 172	weather_wsy_0001.txt	weather	26	3987
## 173	ancient_akk_0001.txt	ancient	10	2698
## 174	ancient_akk_0002.txt	ancient	8	1603
## 175	ancient_cre_0001.txt	ancient	19	270

## 176	ancient_cre_0002.txt	ancient	48	1099
## 177	ancient_ela_0001.txt	ancient	30	578
## 178	ancient_ela_0002.txt	ancient	59	692
## 179	ancient_pra_0001.txt	ancient	4	1101
## 180	ancient_prc_0001.txt	ancient	46	848
## 181	ancient_prc_0002.txt	ancient	24	448
## 182	ancient_prc_0003.txt	ancient	18	210
## 183	ancient_prc_0004.txt	ancient	16	217
## 184	ancient_sum_0001.txt	ancient	70	1560
## 185	ancient_sum_0002.txt	ancient	30	361
## 186	ancient_sum_0003.txt	ancient	9	171
## 187	ancient_sum_0004.txt	ancient	18	237
## 188	ancient_sum_0005.txt	ancient	9	275
## 189	ancient_sum_0006.txt	ancient	11	137
## 190	ancient_sum_0007.txt	ancient	10	227
## 191	ancient_sum_0008.txt	ancient	9	192
## 192	ancient_sum_0009.txt	ancient	19	579
## 193	ancient_sum_0010.txt	ancient	10	228
## 194	signlang_tsl_0001.txt	signlang	14	479
## 195	udhr_aii_0001.txt	udhr	88	6471
## 196	udhr_arb_0001.txt	udhr	90	7629
## 197	udhr_azj_0001.txt	udhr	90	10749
## 198	udhr_azj_0002.txt	udhr	90	10745
## 199	udhr_ben_0001.txt	udhr	94	9718
## 200	udhr_blt_0001.txt	udhr	89	8804
## 201	udhr_bod_0001.txt	udhr	90	12649
## 202	udhr_bos_0001.txt	udhr	90	9694
## 203	udhr_bos_0002.txt	udhr	90	9871
## 204	udhr_chr_0001.txt	udhr	89	8985
## 205	udhr_cmn_0001.txt	udhr	89	2960
## 206	udhr_cmn_0002.txt	udhr	90	2789
## 207	udhr_csw_0001.txt	udhr	67	6513
## 208	udhr_div_0001.txt	udhr	88	18129
## 209	udhr_ell_0001.txt	udhr	90	12392
## 210	udhr_eng_0001.txt	udhr	91	10606
## 211	udhr_epo_0001.txt	udhr	91	9884
## 212	udhr_eus_0001.txt	udhr	93	10907
## 213	udhr_gaz_0001.txt	udhr	92	10473
## 214	udhr_guj_0001.txt	udhr	91	9959
## 215	udhr_heb_0001.txt	udhr	89	7261
## 216	udhr_hin_0001.txt	udhr	90	10497
## 217	udhr_hye_0001.txt	udhr	92	11119
## 218	udhr_ibt_0001.txt	udhr	100	13205
## 219	udhr_iii_0001.txt	udhr	88	3272
## 220	udhr_ike_0001.txt	udhr	67	8541
## 221	udhr_jav_0001.txt	udhr	92	10816
## 222	udhr_jav_0002.txt	udhr	92	13651
## 223	udhr_jpn_0001.txt	udhr	89	4157
## 224	udhr_kal_0001.txt	udhr	90	16786
## 225	udhr_kan_0001.txt	udhr	89	10534
## 226	udhr_kat_0001.txt	udhr	90	11828
## 227	udhr_khm_0001.txt	udhr	90	10652
## 228	udhr_kkh_0001.txt	udhr	82	9938
## 229	udhr_kor_0001.txt	udhr	91	4709

## 230	udhr_lao_0001.txt	udhr	90	10502
## 231	udhr_lug_0001.txt	udhr	87	10354
## 232	udhr_mal_0001.txt	udhr	82	10557
## 233	udhr_mya_0001.txt	udhr	89	15131
## 234	udhr_pan_0001.txt	udhr	90	10681
## 235	udhr_rus_0001.txt	udhr	90	11684
## 236	udhr_sin_0001.txt	udhr	90	10543
## 237	udhr_tam_0001.txt	udhr	89	13133
## 238	udhr_tel_0001.txt	udhr	89	11075
## 239	udhr_tgl_0001.txt	udhr	94	12246
## 240	udhr_tha_0001.txt	udhr	89	9278
## 241	udhr_tir_0001.txt	udhr	89	6640
## 242	udhr_vai_0001.txt	udhr	91	8556
## 243	udhr_zgh_0001.txt	udhr	89	7770
## 244	udhr_zul_0001.txt	udhr	91	10217

Density plot

Density plot for numbers of characters per data file and subcorpus.

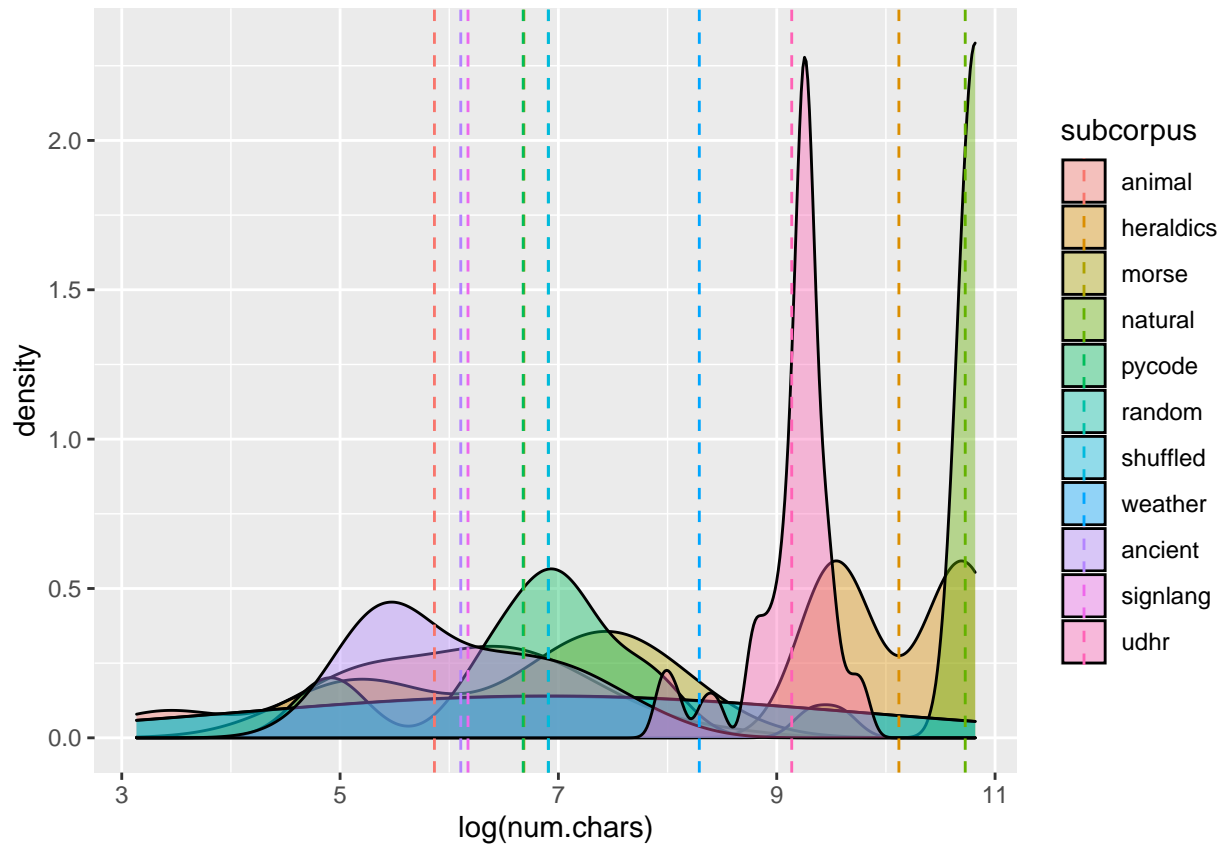
```
# select subcorpora (if applicable)
# selection <- c("writing", "shuffled")
# simpleStats.df <- simpleStats.df[simpleStats.df$subcorpus %in% selection, ]
# get means per subcorpus
mu <- ddply(simpleStats.df, "subcorpus", summarise, grp.mean = mean(log(num.chars)))
# plot densities with mean values as vertical lines
density.plot <- ggplot(simpleStats.df, aes(x = log(num.chars), fill = subcorpus)) +
  geom_density(alpha = 0.4) +
  geom_vline(data = mu, aes(xintercept = grp.mean, color = subcorpus),
    linetype = "dashed")
print(density.plot)
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```



Safe figure to file

```
ggsave("~/Github/NaLaFi/figures/simpleStats_density.pdf", density.plot, dpi = 300,
        scale = 1, device = cairo_pdf)
```

```
## Saving 8 x 4 in image
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```