Appendix 1: Simulation Study with Normal Distribution and Location Shift

Chris Bentz

March 09, 2021

Session Info

Give the session info (reduced).

```
## [1] "R version 3.6.3 (2020-02-29)"
```

```
## [1] "x86_64-pc-linux-gnu"
```

Load Packages

Load packages. If they are not installed yet on your local machine, use install.packages() to install them.

```
library(ggplot2)
library(plyr)
library(rstatix)
```

Give the package versions.

```
## rstatix    plyr ggplot2
## "0.6.0" "1.8.6" "3.3.3"
```

Simulate Vectors

Create vectors of random numbers from the standard normal distribution with mu = 0 and SD = 1. These are the pseudo-measurements of complexity for languages A, B, and C. Language A and B are generated from the same sampling procedure, while language C has a location shift of one standard deviation towards higher values.

```
n = 20 # choose number of pseudo-measurements
# set the seed for random number generation in order to get the same result when the code is re-run
set.seed(1)
# generate values for language A
langA.values <- rnorm(n)
# generate values for language B
langB.values <- rnorm(n)
# generate values for language C with location shift (1 standard deviation)
langC.values <- rnorm(n) + 1

# concatenate to get long format data frame
```

```
value <- c(langA.values, langB.values, langC.values)
measurement <- rep(c(1:n), times = 3)
language <- c(rep("Language A", times = n), rep("Language B", times = n),
              rep("Language C", times = n))
simulation.df <- data.frame(language, measurement, value)
head(simulation.df)
```

```
##     language measurement      value
## 1 Language A           1 -0.6264538
## 2 Language A           2  0.1836433
## 3 Language A           3 -0.8356286
## 4 Language A           4  1.5952808
## 5 Language A           5  0.3295078
## 6 Language A           6 -0.8204684
```

```
tail(simulation.df)
```

```
##      language measurement       value
## 55 Language C          15  2.43302370
## 56 Language C          16  2.98039990
## 57 Language C          17  0.63277852
## 58 Language C          18 -0.04413463
## 59 Language C          19  1.56971963
## 60 Language C          20  0.86494540
```

Visualization: Density Distributions

Plot density distributions of complexity pseudo-measurements by language. Individual values for each complexity pseudo-measurement are plotted as black dots. The central value (0) is indicated by a vertical dotted line for visual reference. The median and mean values of complexity pseudo-measurements per language might also be indicated.

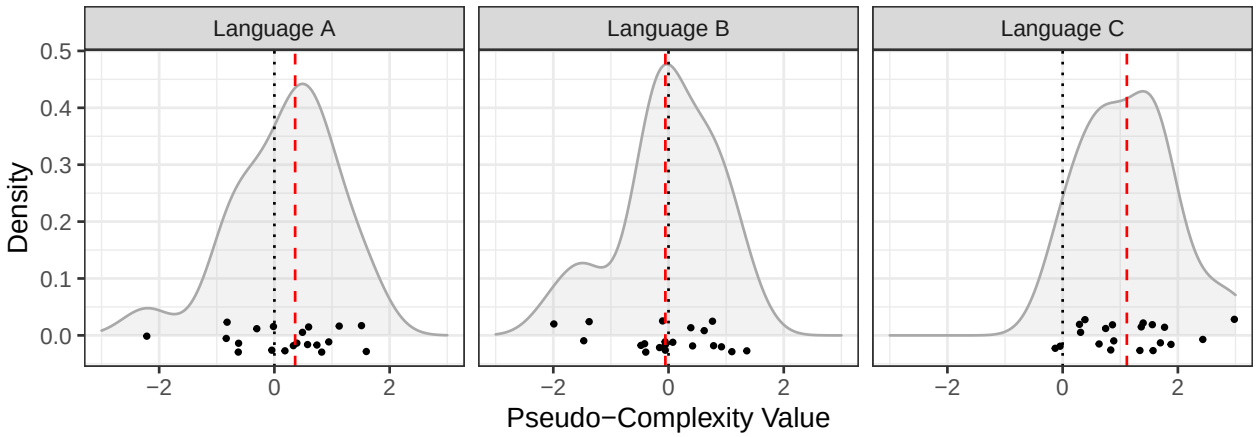
Get mean, median, and standard deviation values.

```
# get mean values for each language
mu <- ddply(simulation.df, "language", summarise, grp.mean = mean(value, na.rm = T))
# get median values for each language
med <- ddply(simulation.df, "language", summarise, grp.median = median(value, na.rm = T))
# get standard deviation values for each language
sdev <- ddply(simulation.df, "language", summarise, grp.sd = sd(value, na.rm = T))
```

Plot density distributions with indication of median (mean) values.

```
density.plot <- ggplot(simulation.df, aes(x = value)) +
 # histograms could be added as well here (or instead of the density distributions)
 # geom_histogram(aes(y = ..density..), colour = "white", fill = "light grey",
                  # binwidth = 0.1) +
 geom_density(alpha = .2, fill = "grey", color = "darkgrey") +
 geom_jitter(data = simulation.df, aes(x = value, y = 0),
             size = 0.7, height = 0.03, width = 0) + # add some jitter to prevent overplotting
 facet_wrap(~ language) +
 # geom_vline(data = mu, aes(xintercept=grp.mean),
 #            linetype = "dotted", color = "blue") +
 geom_vline(data = med, aes(xintercept = grp.median),
             linetype = "dashed", color = "red") +
```

```
 geom_vline(aes(xintercept = 0), linetype = "dotted") +
 labs(x = "Pseudo-Complexity Value", y = "Density") +
 xlim(-3, 3) +
 theme_bw()
print(density.plot)
```



Save figure to file.

```
ggsave("Figures/simulation_densities.pdf", density.plot, dpi = 300, scale = 1,
       device = cairo_pdf)
```

```
## Saving 7 x 2.5 in image
```

Descriptive statistics

Give an overview of mean, median, and standard deviation values (i.e. values reflecting the location of a distribution).

```
stats.df <- cbind(mu, med[, 2], sdev[, 2])
colnames(stats.df) <- c("language", "mu", "med", "sdev")
stats.df.sorted <- stats.df[order(-stats.df$med),]
# round values to two decimal places,  the "-1" excludes column 1
stats.df.sorted[, -1] <- round(stats.df.sorted[, -1], 2)
print(stats.df.sorted)
```

```
##     language    mu   med sdev
## 3 Language C  1.14  1.11 0.81
## 1 Language A  0.19  0.36 0.91
## 2 Language B -0.01 -0.05 0.87
```

Output data frame as csv file.

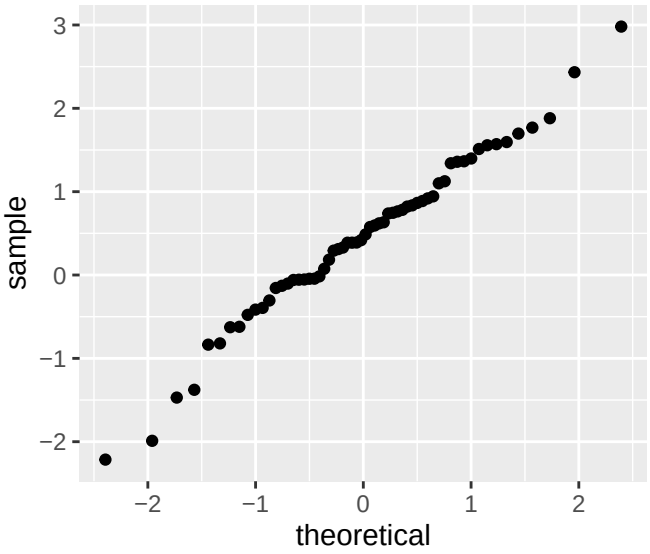
```
write.csv(stats.df.sorted, file = "Tables/simulation_descriptiveStats.csv", row.names = F)
```

Normality

The assumption that the tested data stems from a normally distributed population is often necessary for the mathematical proofs underlying standard statistical techniques. We might apply normality tests to check for this assumption (e.g. Baayen 2008, p. 73), but some statisticians advice against such pre-tests, since they are

often too sensitive (MacDonald 2014, p. 133-136, Rasch et al. (2020), p. 67). In fact, Rasch et al. (2020, p. xi) argue based on earlier simulation studies that almost all standard statistical tests are fairly robust against deviations from normality. In a similar vein, Lumley et al. (2009) argue that non-normality of the data is a negligible issue with the t-test, at least for larger sample sizes, e.g. >= 100. However, especially for smaller sample sizes, it is still advisable to check for gross deviations from normality in the data. One common way of doing this is quantile-quantile plots. The points should here roughly follow a straight line (Crawley 2007, p. 281).

```
ggplot(simulation.df, aes(sample = value)) + stat_qq()
```



Statistical Tests

Standard t-tests can be used to assess significant differences in the means of the pseudo-complexity distributions, if we assume that the underlying population distributions are normal. Wilcoxon tests are a non-parametric alternative, i.e. they do not make assumptions about the underlying population distribution (Crawley 2007, p. 283; Baayen 2008, p. 77). We here run pairwise t-tests for illustration purposes. If we supply two data samples, then by default the function pairwise.t.test() runs a Welch two sample t-test (for unpaired samples); with the argument "paired = T" a paired t-test is invoked. We here assume that our data consists of three samples which are linked via the same measurement procedure (here randomly generated), and we hence consider them "paired". A more general term is "related samples", which are defined as "two sets of data where a data point in one set has a pairwise relationship to a point in the other set of data" (Cahusac 2021, p. 56).

P-value adjustment for multiple comparisons: In case of multiple testing, we should account for the fact that the likelihood of finding a significant result by chance increases with the number of statistical tests. One of the most conservative methods to account for this is the so-called Bonferroni correction, i.e. multiplying the p-values with the number of tests. This method assumes that tests are independent of one another (MacDonald 2014, p. 254-260). However, since we here compare, for example, language A to B and language A to C, there is dependence between the test results. We therefore apply the so-called Holm-Bonferroni method, which is less conservative. It does not assume independence between tests (see the descriptions in the vignette invoked by the command "?p.adjust()").

```
p.values <- pairwise.t.test(simulation.df$value, simulation.df$language,
                    paired = T, p.adjust.method = "holm")
p.values
```

```
##
##  Pairwise comparisons using paired t tests
##
## data:  simulation.df$value and simulation.df$language
##
##            Language A Language B
## Language B 0.5345     -
## Language C 0.0017     0.0024
##
## P value adjustment method: holm
```

Effect sizes

Statistical significance is only one part of the story. For instance, a difference in complexity values might be statistically significant, but so small that it is negligible for any theorizing. In fact, it is sometimes argued that effect sizes – rather than p-values – should be the aim of statistical inquiry (Cahusac 2020, p. 12-15). An overview of effect size measures per statistical test is given in Patil (2020). In conjunction with the t-test we here use Cohen's d (i.e. function cohens_d() of the "rstatix" package). (Note: the d estimate given by this function can be negative. However, the sign is not relevant here, only the absolute value.)

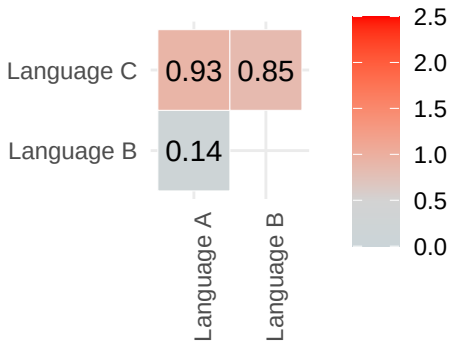
```
effect.sizes <- cohens_d(simulation.df, value ~ language, paired = T)
print(effect.sizes)
```

```
## # A tibble: 3 x 7
##   .y.   group1     group2     effsize    n1    n2 magnitude
## * <chr> <chr>      <chr>        <dbl> <int> <int> <ord>
## 1 value Language A Language B   0.141    20    20 negligible
## 2 value Language A Language C  -0.926    20    20 large
## 3 value Language B Language C  -0.850    20    20 large
```

Effect size heatmap

Plot a heatmap with effect sizes to get a better overview.

```
effect.sizes.plot <- ggplot(as.data.frame(effect.sizes), aes(group1, group2)) +
  geom_tile(aes(fill = abs(effsize)), color = "white") +
  scale_fill_gradient2(low = "light blue", mid = "light grey", high = "red",
                       midpoint = 0.5, limit = c(0, 2.5)) +
  geom_text(aes(label = round(abs(effsize), 2))) +
  labs(x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
effect.sizes.plot
```



Save figure to file.

```
ggsave("Figures/simulation_effectSizes.pdf", effect.sizes.plot, dpi = 300, scale = 1,
       device = cairo_pdf)
```

```
## Saving 3 x 2 in image
```

Interpretation

Note that the exact results of the tests above will differ every time the random vectors are re-sampled (which is avoided here by using the set.seed() function). Having said this, below is a more general interpretation of the likely outcomes.

Statistical significance

In case $p > 0.05$ for a given statistical test (after correction for multiple comparisons), we conclude that the null hypothesis (in our case that the difference in means of the two distributions is 0) cannot be rejected, i.e. the pseudo-complexity measurements likely derive from the same underlying distribution. In case $p < 0.05$, we conclude that the null hypothesis cannot be upheld, i.e. the pairwise differences in pseudo-complexity values are shifted away from 0. For the simulated data above, $p > 0.05$ is very likely the case for a comparison between language A and B, and $p < 0.05$ is very likely the case for comparisons between A and C as well as B and C.

We would thus conclude that language A and B have the same complexity, while C has a significantly higher complexity (also considering the median and mean values).

Effect size

If we have found a significant location shift in the distributions, the question is how strong this shift (i.e. "effect") is. In this particular example, we used Cohen's d as an effect size measure. The effect is typically considered "small" when $d < 0.2$, "medium" when $0.2 < d < 0.8$, and "large" when $d > 0.8$. Sometimes "very large" is attributed to $d > 1.3$ (Cahusac 2021, p. 14).

For the simulated data above, the effect between A and B is typically "negligible" or "small", and the effect between A and C as well as B and C "large". Note that the expected location shift of C compared to A and B is exactly one standard deviation.

Alternative statistical approaches

We here used so-called "frequentist" statistical tests. A common alternative are Bayesian statistics. For the t-test, for instance, there is a Bayesian alternative proposed in Kruschke (2012). Another, less widespread

alternative, are tests in the framework of "evidence-based" statistics which are based on likelihood ratios for competing hypotheses (see Cahusac, 2021, pp. 7 for a discussion). While different researchers might prefer different statistical approaches, Cahusac (2021, p. 8) states that: "If the collected data are not strongly influenced by prior considerations, it is somewhat reassuring that the three approaches usually reach the same conclusion."

References

Baayen, R. H. (2008). Analyzing linguistic data: A practical introduction using statistics in R. Cambridge University Press.

Cahusac, P. M. B. (2021). Evidence-based statistics. John Wiley & Sons.

Crawley, M. J. (2007). The R book. John Wiley & Sons Ltd.

Kruschke, J. K. (2012). Bayesian estimation supersedes the t test. Journal of Experimental Psychology.

Lumley et al. (2002). The importants of the normality assumption in large public health data sets. Annu. Rev. Public Health.

McDonald, J.H. (2014). Handbook of Biological Statistics (3rd ed.). Sparky House Publishing, Baltimore, Maryland. online at http://www.biostathandbook.com

Patil, I. (2020). Test and effect size details. online at https://cran.r-project.org/web/packages/statsExpressions/vignettes/stats_details.html.

Rasch, D., Verdooren, R., and Jürgen Pilz (2020). Applied statistics. Theory and problem solutions with R. John Wiley & Sons Ltd.