T-test vs Wilcoxon

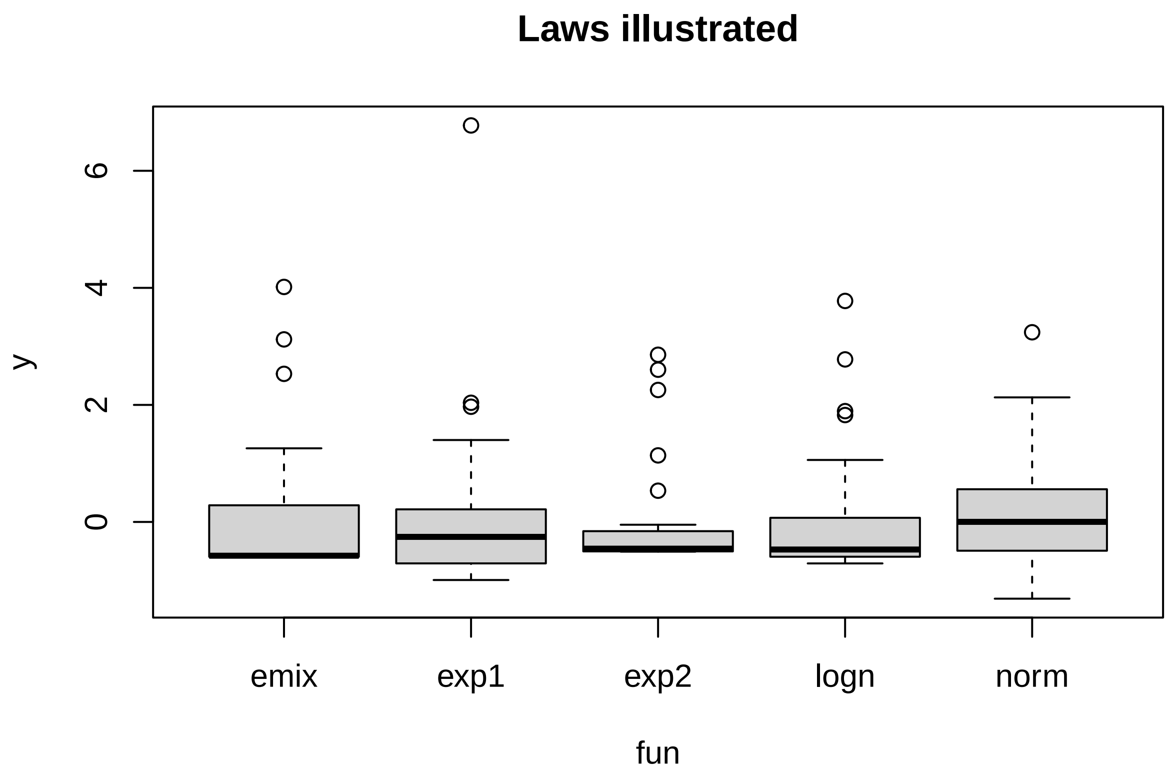
Lukas Graz

2023-09-20

Define a list of laws (i.e. distributions) with `mean` and `sd`.

```r
# functions
n <- 40 # per group
laws <- list(
  norm = function(mean=0, sd=1) mean + sd *  rnorm(n, 0, 1),
  logn = function(mean=0, sd=1) mean + sd * (rlnorm(n, sdlog=1) - 1.65) / 1.95,
  exp1 = function(mean=0, sd=1) mean + sd * (rexp(n) - 1),
  exp2 = function(mean=0, sd=1) mean + sd * (rexp(n)^2 - 2) / 3.94,
  emix = function(mean=0, sd=1) mean + sd * (c(rep(0, n/2), rexp(n/2)) - 0.5) / 0.84
)
```

```r
set.seed(123)
get_data_all_laws <- function(){
  dat <- lapply(names(laws), function(fname){
    f <- laws[[fname]]
    data.frame(y=f(), fun=fname)
  })
  dat <- do.call(rbind, dat)
  dat
}
boxplot(y ~ fun, get_data_all_laws(), main= "Laws illustrated")
```



verify mean and standard deviation

```
sapply(laws, function(f) mean(replicate(10000, mean(f(0, 1)))))
```

```
     norm      logn      exp1      exp2      emix 
 1.55e-03 -2.89e-03 -1.66e-03 -8.53e-05 -1.32e-03 
```

```
sapply(laws, function(f) mean(replicate(10000, mean(f(1, 2)))))
```

```
 norm  logn  exp1  exp2  emix 
1.003 1.002 0.998 0.994 0.997 
```

```
sapply(laws, function(f) mean(replicate(10000,   sd(f(0, 1)))))
```

```
 norm  logn  exp1  exp2  emix 
0.994 1.008 0.977 0.999 1.000 
```

```r
sapply(laws, function(f) mean(replicate(10000,   sd(f(1, 2)))))
```

```
norm logn exp1 exp2 emix
1.99 2.01 1.96 1.99 2.00
```

Define the tests used

```r
pval_t <- function(d) t.test(y ~ group,d)$p.value
pval_w <- function(d) coin::pvalue(coin::wilcox_test(y ~ group, d))
pval_m <- function(d) coin::pvalue(coin::median_test(y ~ group, d))
```

```r
# retuns data with f1() for group "A" and f2(mean2, sd2) for group "B"
get_data <- function(f1, f2, mean2=0, sd2=1){
  d <- rbind(
    data.frame(
      y=f1(),
      group="A"
    ),
    data.frame(
      y=f2(mean2, sd2),
      group="B"
    ))
  d$group <- as.factor(d$group)
  d
}

# get power of tests
get_power <- function(f1, f2, nsim=1000, mean2=0, sd2=1) {
  data_list <- replicate(nsim, get_data(f1, f2, mean2=mean2, sd2=sd2), simplify = FALSE)
  c(t = mean(sapply(data_list, pval_t) < 0.05),
    w = mean(sapply(data_list, pval_w) < 0.05),
    m = mean(sapply(data_list, pval_m) < 0.05))
}

# get all combinations of functions
fun_comb <- expand.grid(names(laws), names(laws)) |> as.matrix()
rnames <- apply(fun_comb, 1, paste0, collapse="_")

sim <- function(nsim=1000, mean2=0, sd2=1) {
  fun_comb_list <- split(fun_comb, row(fun_comb))
  coverage <- parallel::mclapply(fun_comb_list, function(f_names){
    f1 <- laws[[f_names[1]]]
    f2 <- laws[[f_names[2]]]
    get_power(f1, f2, nsim=nsim, mean2=mean2, sd2=sd2)
  })
  coverage <- do.call(rbind, coverage)
  rownames(coverage) <- rnames
  colnames(coverage) <- paste0(
    colnames(coverage), " ", as.character(mean2), " ", as.character(sd2))
  coverage |> as.data.frame()
}
```

Simulation

```
nsim <- 10
```

```
set.seed(4321)
null <- sim(nsim=nsim)
```

```
set.seed(4321)
s_diff <- sim(nsim=nsim, sd2=2)
```

```
set.seed(4321)
s_difff <- sim(nsim=nsim, sd2=5)
```

```
set.seed(4321)
mu_diff <- sim(nsim=nsim, mean2 = 0.1)
```

```
set.seed(4321)
mu_difff <- sim(nsim=nsim, mean2 = 0.2)
```

```
results <- cbind(
  null,
  s_diff,
  s_difff,
  mu_diff,
  mu_difff
)
results * 100
```

	t 0 1	w 0 1	m 0 1	t 0 2	w 0 2	m 0 2	t 0 5	w 0 5	m 0 5	t 0.1 1	w 0.1 1	m 0.1 1	t 0.2 1	w 0.2 1	m 0.2 1
norm_norm	10	10	0	0	10	0	10	10	10	0	0	0	20	20	10
logn_norm	0	20	40	0	0	20	0	30	30	0	20	40	10	20	60
exp1_norm	0	0	20	10	10	10	20	20	30	0	10	40	10	40	60
exp2_norm	0	20	40	10	10	20	0	10	20	10	20	100	50	100	100
emix_norm	0	10	40	10	10	30	0	0	10	10	30	70	0	10	80
norm_logn	0	20	50	20	40	70	0	70	80	0	0	10	0	0	20
logn_logn	10	0	10	10	60	20	10	80	70	10	0	0	10	60	40
exp1_logn	0	0	0	20	70	70	20	80	70	10	20	0	10	40	10
exp2_logn	10	0	0	10	70	30	0	70	60	10	0	20	40	70	70
emix_logn	0	20	40	0	50	0	10	90	20	0	50	70	20	100	60
norm_exp1	0	10	10	0	20	40	10	70	70	0	0	10	0	0	0
logn_exp1	0	0	0	0	30	10	20	50	40	10	0	0	10	20	10
exp1_exp1	20	0	0	0	40	10	0	60	40	0	10	0	20	40	10
exp2_exp1	0	0	0	0	40	10	10	30	30	20	10	20	20	30	60
emix_exp1	0	20	10	10	20	0	10	50	10	10	10	40	10	10	70
norm_exp2	10	10	40	20	80	90	30	90	90	0	20	50	0	0	20
logn_exp2	0	10	0	0	80	70	10	100	100	0	10	0	0	50	10
exp1_exp2	10	30	10	0	90	80	20	100	100	0	30	10	10	70	10
exp2_exp2	0	0	0	0	80	60	10	100	100	20	90	40	0	100	50
emix_exp2	0	80	20	10	70	10	10	100	40	0	80	30	0	100	70
norm_emix	10	10	30	0	60	70	10	100	100	0	0	30	0	0	20
logn_emix	0	0	10	0	100	90	0	100	90	10	30	0	10	30	0
exp1_emix	20	0	0	20	90	70	20	90	80	0	10	10	20	30	0
exp2_emix	10	60	20	0	80	60	0	70	70	40	0	0	10	80	0
emix_emix	0	0	0	0	80	10	0	100	30	0	80	0	10	90	0

Confidence intervals of ratios

```
prop <- function(ratio, nsim){
  confint_ <- prop.test(round(ratio*nsim), nsim)$conf.int[1:2]
  names(confint_) <- c("lower", "upper")
   c(
    ratio= ratio,
    confint_
)}
sapply(c(0:4/40, 3:10/20), prop, nsim) |> as.data.frame() * 100
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13
ratio	0.0	2.5	5.0	7.500	10.000	15.00	20.00	25.00	30.0	35.0	40.0	45.0	50.0
lower	0.0	0.0	0.0	0.524	0.524	3.54	3.54	3.54	8.1	13.7	13.7	13.7	23.7
upper	34.5	34.5	34.5	45.885	45.885	55.78	55.78	55.78	64.6	72.6	72.6	72.6	76.3

this gives an idea of the uncertainty of a ratio given 10 simulations

redo analysis but with groupwise equal medians

Define a list of laws (i.e. distributions) with `median` and `sd`

```
# functions
n <- 40 # per group
# mean == to keep notation consistent
laws <- list(
  norm = function(mean=0, sd=1) mean + sd *  rnorm(n, 0, 1)
  ,logn = function(mean=0, sd=1) mean + sd * (rlnorm(n, sdlog=1) - 1.02) / 1.95
  ,exp1 = function(mean=0, sd=1) mean + sd * (rexp(n) - 0.706)
  ,exp2 = function(mean=0, sd=1) mean + sd * (rexp(n)^2 - 0.523) / 3.94
  ,emix = function(mean=0, sd=1) mean + sd * (c(rep(0, n/2), rexp(n/2)) -0.025) / 0.84
)
```

verify **median** and standard deviation

```
sapply(laws, function(f) mean(replicate(10000, median(f(0, 1)))))
```

```
     norm      logn      exp1      exp2      emix
-0.000142 -0.000216  0.002144  0.000204  0.000246
```

```
sapply(laws, function(f) mean(replicate(10000, median(f(1, 2)))))
```

```
 norm  logn  exp1  exp2  emix
1.000 1.001 0.998 1.000 0.999
```

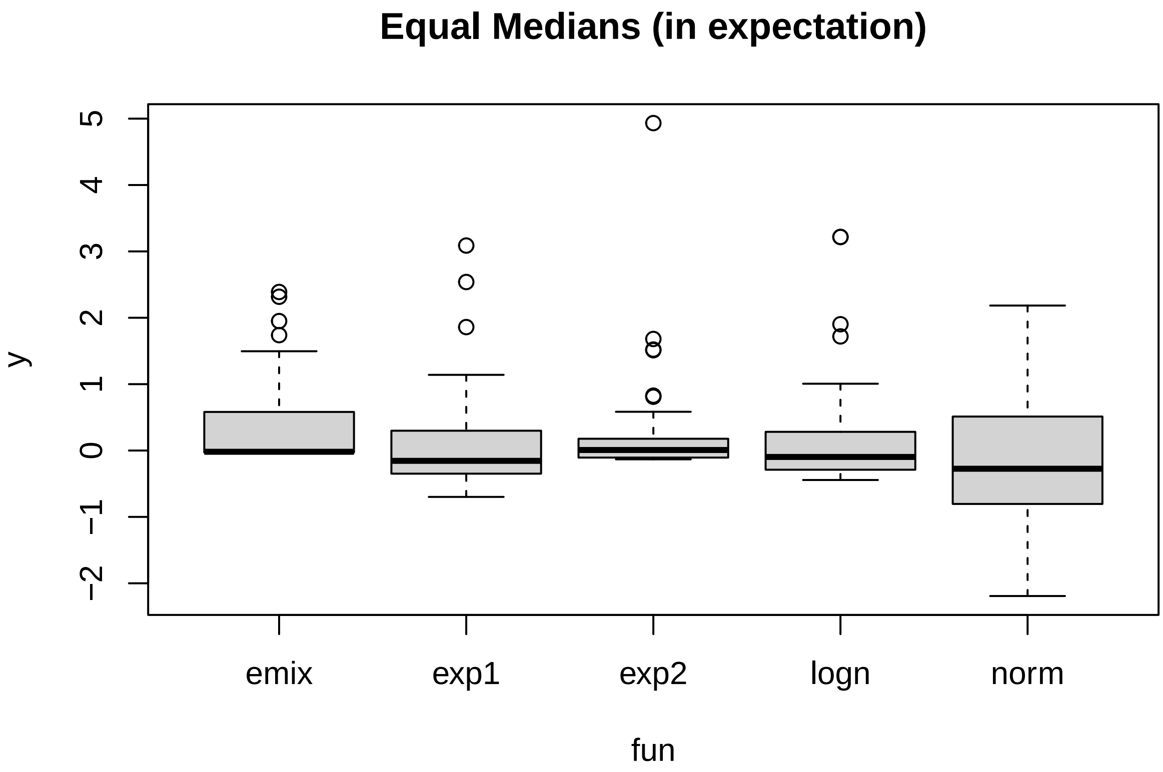
```
sapply(laws, function(f) mean(replicate(10000,     sd(f(0, 1)))))
```

```
 norm  logn  exp1  exp2  emix
0.996 1.006 0.978 1.010 1.003
```

```
sapply(laws, function(f) mean(replicate(10000,     sd(f(1, 2)))))
```

```
norm logn exp1 exp2 emix
1.99 2.00 1.95 1.99 2.01
```

```
boxplot(y ~ fun, get_data_all_laws(), main= "Equal Medians (in expectation)")
```



```
set.seed(4321)
null_median <- sim(nsim=nsim)
```

```
set.seed(4321)
s_diff_median <- sim(nsim=nsim, sd2=2)
```

```
set.seed(4321)
s_difff_median <- sim(nsim=nsim, sd2=5)
```

```
set.seed(4321)
mu_diff_median <- sim(nsim=nsim, mean2 = 0.1)
```

```
set.seed(4321)
mu_difff_median <- sim(nsim=nsim, mean2 = 0.2)
```

```r
results_median <- cbind(
  null_median,
  s_diff_median,
  s_difff_median,
  mu_diff_median,
  mu_difff_median
)
results_median * 100
```

	t 0	w 0	m 0	t 0	w 0	m 0	t 0	w 0	m 0	t 0.1	w	m	t 0.2	w	m
	1	1	1	2	2	2	5	5	5	1	0.1 1	0.1 1	1	0.2 1	0.2 1
norm__norm	0	0	0	0	10	0	0	0	0	10	20	10	0	0	0
logn__norm	10	10	0	10	10	0	0	0	10	20	0	10	0	10	10
exp1__norm	10	10	0	0	10	0	0	0	0	0	0	0	10	10	10
exp2__norm	20	20	20	20	0	10	0	10	10	10	10	20	0	0	0
emix__norm	60	60	0	40	40	0	0	10	20	40	40	10	10	0	10
norm__logn	20	10	0	40	10	0	40	10	0	60	30	10	60	40	10
logn__logn	0	0	0	0	10	0	20	10	10	0	20	10	0	60	30
exp1__logn	10	20	0	0	0	0	20	10	10	20	40	20	10	60	20
exp2__logn	10	20	20	0	20	10	40	0	10	10	10	0	0	20	50
emix__logn	30	70	0	0	30	0	0	30	0	20	0	0	20	0	0
norm__exp1	30	20	0	10	0	0	50	10	10	30	30	0	60	50	0
logn__exp1	10	0	0	10	10	20	20	0	10	20	10	10	0	0	0
exp1__exp1	0	0	10	10	0	10	30	10	10	10	10	10	10	50	10
exp2__exp1	20	30	0	0	0	0	40	10	0	0	0	10	20	20	20
emix__exp1	40	70	0	0	30	0	40	30	10	10	10	0	0	0	10
norm__exp2	60	20	0	60	40	0	80	30	0	60	60	20	80	50	30
logn__exp2	20	30	10	20	20	0	30	10	0	0	60	10	30	90	30
exp1__exp2	10	30	20	20	40	0	40	20	30	20	50	20	10	40	20
exp2__exp2	0	0	0	30	10	0	40	0	0	0	50	10	10	80	50
emix__exp2	0	60	0	10	30	20	20	20	0	0	0	0	0	30	0
norm__emix	70	70	0	100	90	0	100	90	0	100	70	10	100	100	0
logn__emix	30	50	0	60	90	0	100	80	0	40	100	0	50	90	30
exp1__emix	20	60	0	70	70	0	100	80	0	60	70	20	70	100	10
exp2__emix	0	80	0	60	70	0	100	0	0	30	80	10	60	100	30
emix__emix	0	0	0	20	0	0	100	0	0	0	70	0	0	80	0