Compare permutation and scalar transformation approaches on simulated and real data

Sayani Gupta

1 Simulated data

1.1 Data generation

Observations are generated from a Gamma(2,1) distribution for each combination of nx and $nfacet$ from the following sets: $nx = nfacet = \{2, 3, 5, 7, 14, 20, 31, 50\}$ to cover a wide range of levels from very low to moderately high. Each combination is being referred to as a *panel*. That is, data is being generated for each of the panels $\{nx = 2, nfacet = 2\}, \{nx = 2, nfacet = 3\}, \{nx = 2, nfacet = 5\}, \dots, \{nx = 50, nfacet = 31\}, \{nx = 50, nfacet = 50\}$. For each of the 64 panels, $ntimes = 500$ observations are drawn for each combination of the categories. That is, if we consider the panel $\{nx = 2, nfacet = 2\}$, 500 observations are generated for each of the combination of categories from the panel, namely, $\{(1, 1), (1, 2), (2, 1), (2, 2)\}$. The values of λ is set to 0.67 and values of raw wpd wpd_{raw} is obtained.

1.2 Scalar transformation approach to normalisation

A log-linear model is fitted to see how the values of wpd_{raw} changes with the values of nx and $nfacet$. The model is of the form

$$y = a + b * log(x) + e$$

, where $y = median(wpd_{raw})$ and $x = nx * nfacet$. wpd_norm is a transformation on wpd_{raw} which should be designed to remove the effect of $nx * nfacet$ on wpd_{raw} and thus is defined as follows: $wpd_{norm} = wpd_{raw} - b * log(nx * nfacet)$

1.3 Permutation approach to normalisation

The simulated data for each of the panels is permuted/shuffled $nperm = 200$ times and for each of those permutations wpd_{norm} is computed as follows: $wpd_{norm} = (wpd_{raw} - mean(wpd_{raw}))/sd(wpd_{raw})$. This is done so that the distribution of the normalised measure wpd_{norm} has the same mean and standard deviation across different nx and nfacet.

1.4 Comparing the two approaches

In Figure 1, we see that the distribution of $wpd_{norm2} = wpd_{raw} - b * log(nx * nfacet)$ is similar when normalised with true or approximate estimate of b. Morever, the distributions are pretty similar across different nx and nfacet.

```
##
## Call:
## lm(formula = actual ~ poly(log(`nx * nfacet`), 1, raw = TRUE),
##     data = G21_median)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.946e-03 -2.240e-04  5.135e-05  4.147e-04  1.014e-03
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             4.003e-02  4.171e-04   95.97   <2e-16
## poly(log(`nx * nfacet`), 1, raw = TRUE) 2.826e-03  8.796e-05   32.13   <2e-16
##
## (Intercept)                             ***
## poly(log(`nx * nfacet`), 1, raw = TRUE) ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0007881 on 32 degrees of freedom
## Multiple R-squared:  0.9699, Adjusted R-squared:  0.969
## F-statistic:  1032 on 1 and 32 DF,  p-value: < 2.2e-16
```

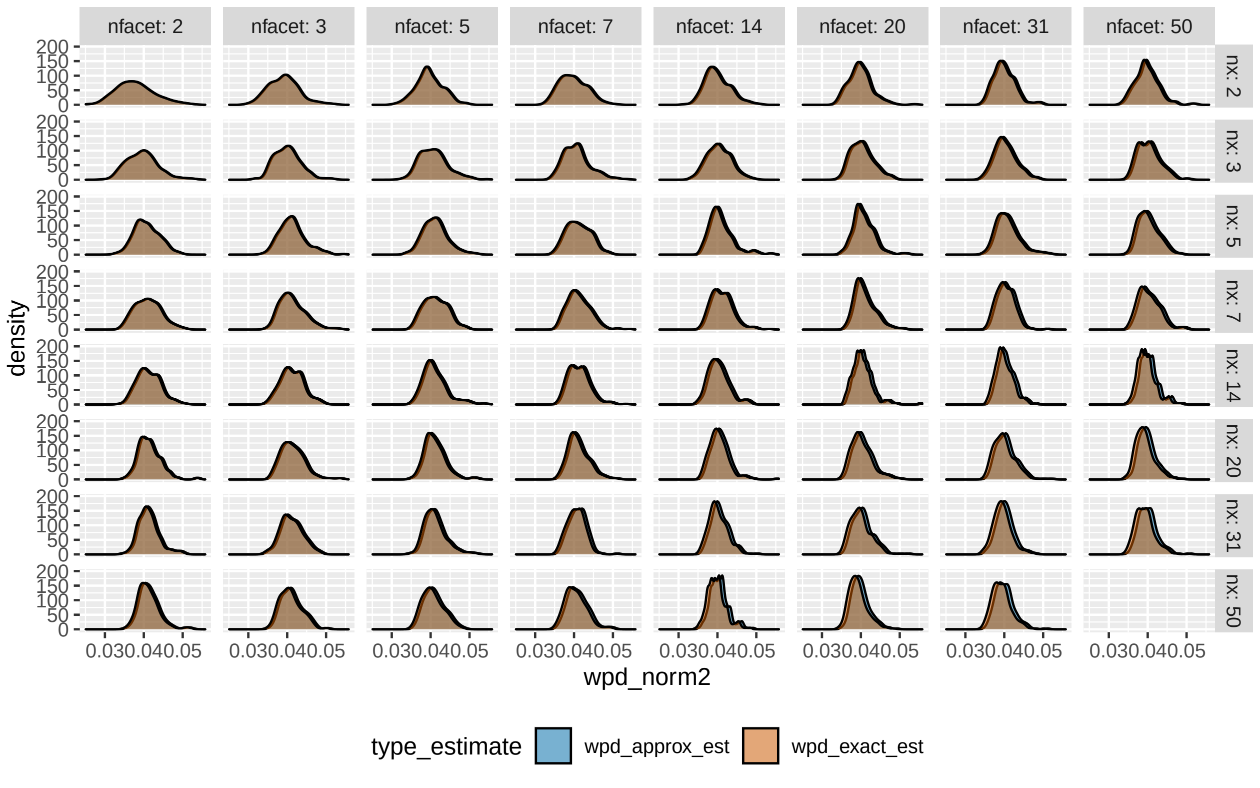

Figure 1: The distribution of wpd_{norm2} is plotted for approximate estimate of b. The distributions are similar when plotted against true and approxiomate estimate and b.

1.5 Ranking different designs

```
## # A tibble: 5,000 x 11
##    type     nx nfacet     w nperm ntimes perm_id  null vary_f vary_x right_order
```

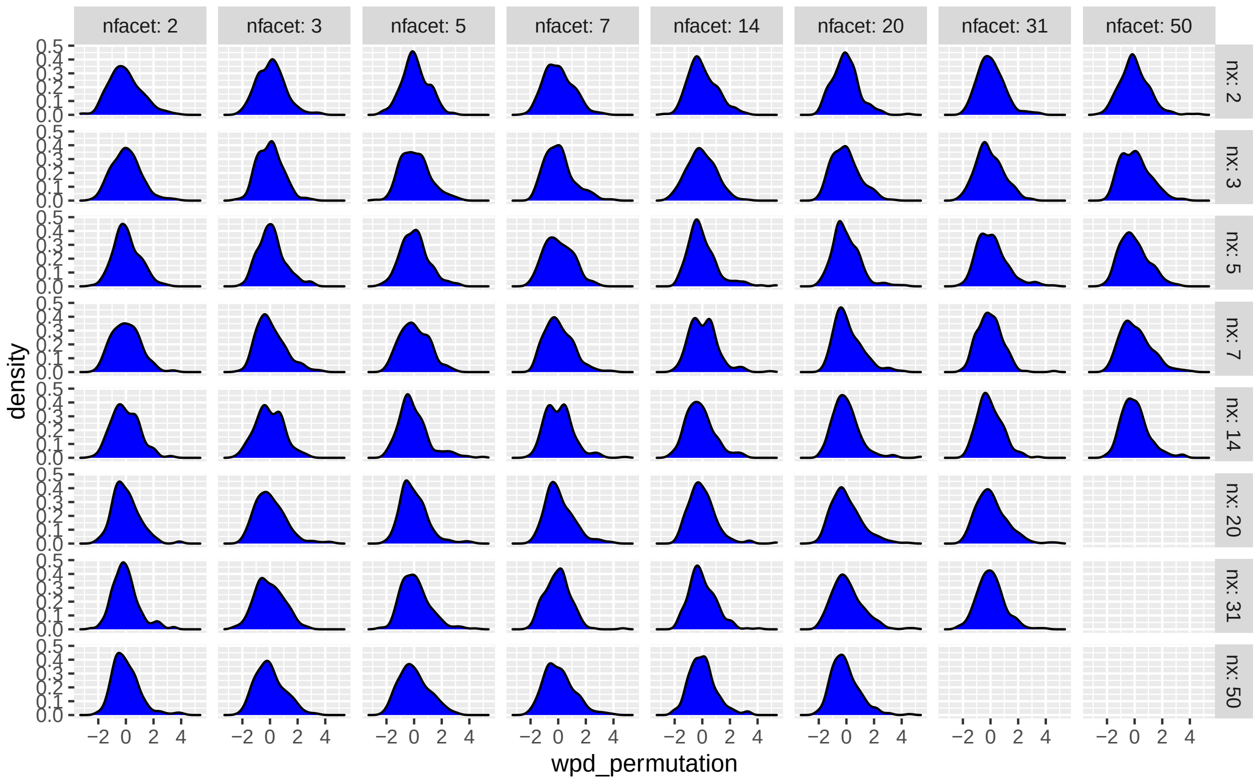

Figure 2: The distribution of $wpd_{permutation}$ is plotted. The distributions are more similar across different nx and nfacet (specially for small nx and nfacet) but this approach has the downside of more computational time.

```
##    <chr> <dbl>  <dbl> <dbl> <dbl>  <dbl>   <int> <dbl>  <dbl>  <dbl> <lgl>
##  1 raw      14     14     1    10    100       1 0.177  0.177  0.157 FALSE
##  2 raw      14     14     1    10    100       2 0.173  0.159  0.171 FALSE
##  3 raw      14     14     1    10    100       3 0.175  0.176  0.179 FALSE
##  4 raw      14     14     1    10    100       4 0.171  0.218  0.163 FALSE
##  5 raw      14     14     1    10    100       5 0.184  0.154  0.156 FALSE
##  6 raw      14     14     1    10    100       6 0.159  0.176  0.168 FALSE
##  7 raw      14     14     1    10    100       7 0.175  0.19   0.16  FALSE
##  8 raw      14     14     1    10    100       8 0.159  0.178  0.175 FALSE
##  9 raw      14     14     1    10    100       9 0.173  0.164  0.165 FALSE
## 10 raw      14     14     1    10    100      10 0.156  0.169  0.159 FALSE
## # ... with 4,990 more rows
```