

# Means and Standard Deviations of T-statistics in All Null Setting

*David Gerard*

## Abstract

Under the all-null setting, I calculate t-statistics from different versions of RUV4. I plot the standard deviations and means of the t-statistics.

## Simulations

I take subsets of the GTEX muscle data with  $p = 1000$ , randomly assign half of the observations to be “treatments” and the other half to be “controls”. I then run RUV4, giving it half of the observations as negative controls and obtain its t-statistics. I do this in four ways:

1. The t-statistics output from RUV4, which uses OLS to estimate the confounders.
2. RUV4 using GLS to estimate the confounders.
3. RUV4 with variance inflation assuming a t-likelihood but no ad-hoc multiplication by  $n/(n - p - q)$ .
4. RUV4 with variance inflation assuming a t-likelihood AND ad-hoc multiplication by  $n/(n - p - q)$ .

I then calculate the means and standard deviations of the t-statistics.

## Results

Above, method 3 results in standard deviations near the horizontal red lines. The no-inflation t-statistics standard deviations have a mean close to the average theoretical standard deviations (red horizontal lines), but have much larger variances. I calculated these average theoretical standard deviations by taking

$$df = n - k - q \tag{1}$$

$$sd = \sqrt{df/(df - 2)}, \tag{2}$$

then averaging these standard deviations. Here,  $k = 2$  (number of covariates),  $n$  is the sample size, and  $q$  is the estimated number of hidden confounders calculated using `num.sv`.

```
library(ggplot2)
library(reshape2)

itermax <- 200
Nsamp_seq <- c(10, 20, 40)
par_vals <- expand.grid(list(1:itermax, Nsamp_seq))

nullmat <- read.csv("null_mat.csv")
names(nullmat) <- c("inp_mean", "inp_sd", "i_mean", "i_sd", "s_mean",
```

```

      "s_sd", "scale", "adhoc_mult")
nullmat$sgls_mean <- nullmat$inp_mean * sqrt(nullmat$scale)
nullmat$sgls_sd <- nullmat$inp_sd * sqrt(nullmat$scale)

nullmat$Nsamp <- par_vals$Var2

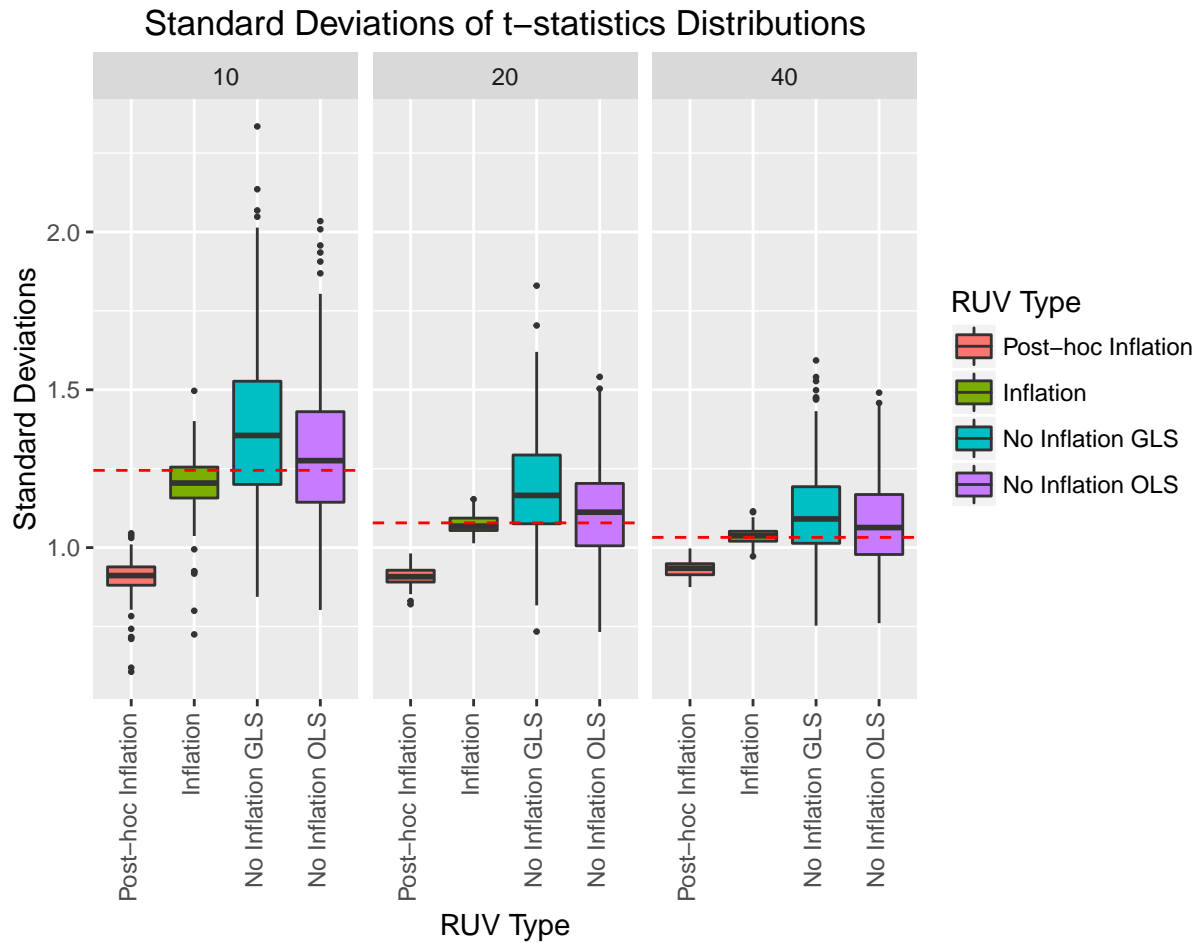
numsv <- nullmat$Nsamp - nullmat$Nsamp / nullmat$adhoc_mult^2 - 2
nu <- nullmat$Nsamp - numsv - 2
nullmat$tsd <- sqrt(nu / (nu - 2))
dummy_df <- aggregate(tsd ~ Nsamp, data = nullmat, FUN = mean)
names(dummy_df) <- c("Nsamp", "mean_sd")

sd_mat <- melt(nullmat, measure.vars = c("inp_sd", "i_sd", "sgls_sd", "s_sd"),
              id.vars = "Nsamp")

ggplot(data = sd_mat, mapping = aes(y = value,
                                     x = factor(variable,
                                                labels = c("Post-hoc Inflation",
                                                            "Inflation",
                                                            "No Inflation GLS",
                                                            "No Inflation OLS")),
                                     fill = factor(variable,
                                                labels = c("Post-hoc Inflation",
                                                            "Inflation",
                                                            "No Inflation GLS",
                                                            "No Inflation OLS")))) +

  facet_grid(.~Nsamp) +
  geom_boxplot(outlier.size = 0.5) +
  ggtitle("Standard Deviations of t-statistics Distributions") +
  guides(fill = guide_legend(title = "RUV Type")) +
  geom_hline(data = dummy_df, aes(yintercept = mean_sd), col = 2, lty = 2) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.3)) +
  xlab("RUV Type") + ylab("Standard Deviations")

```



```
mean_mat <- melt(nullmat, measure.vars = c("inp_mean", "i_mean", "sgls_mean", "s_mean"),
  id.vars = "Nsamp")
ggplot(data = mean_mat, mapping = aes(y = value,
  x = factor(variable,
    labels = c("Post-hoc Inflation",
      "Inflation",
      "No Inflation GLS",
      "No Inflation OLS")),
  fill = factor(variable,
    labels = c("Post-hoc Inflation",
      "Inflation",
      "No Inflation GLS",
      "No Inflation OLS")))) +
  facet_grid(.~Nsamp) +
  geom_boxplot(outlier.size = 0.5) +
  ggtitle("Means of t-statistics Distributions") +
  guides(fill = guide_legend(title = "RUV Type")) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.3)) +
  xlab("RUV Type") + ylab("Means")
```

