Efficacy of Vitamin-C Delivery Methods on Tooth Growth

Darren Bishop (mail@darrenbishop.com)

```
library(dplyr)
library(tidyr)
library(ggplot2)
library(knitr)
library(UsingR)
library(gridExtra)

set.seed(221181)

data(ToothGrowth)

opts_chunk$set(message=FALSE, fig.path="figures/", fig.width=10, fig.align = "center")
```

Overview

This study analyses the ToothGrowth dataset, which looks at the affect of Vitaming-C dietary supplementation on tooth-growth in Guinea Pigs.

Through permutation testing I attempt to assess the efficacy of Orange Juice over Ascorbic Acid as a delivery method of Vitamin-C.

The Null-Hypothesis

> There is no difference in the efficacy of Orange Juice over Ascorbic Acid as a develivery method for Vitamin-C

Assumptions

Here I assume that Vitamin-C does indeed affect tooth-growth and that the difference in tooth-growth, if any, is attributed to the delivery method.

With respect to the data, I assume that the variance in the population sampled is equal and the distributions generated are Normal Gaussian.

Data Processing

First a brief summary of the tooth-growth data

```
tg.df = tbl_df(ToothGrowth) %>%
    rename(Supplement = supp, Dose = dose, Length = len) %>%
    mutate(Dose = factor(sprintf("%.1f", Dose))) %>%
    group_by(Supplement, Dose)

tg.summary = tg.df %>%
    summarise(Mean = mean(Length), Median = median(Length), SD = sd(Length))

kable(tg.summary,
      caption = "Simple summary statistics for tooth-growth, grouped by Supplement and Dose")
```

Table 1: Simple summary statistics for tooth-growth, grouped by Supplement and Dose

Supplement	Dose	Mean	Median	SD
OJ	0.5	13.23	12.25	4.459708
OJ	1.0	22.70	23.45	3.910953
OJ	2.0	26.06	25.95	2.655058
VC	0.5	7.98	7.15	2.746634
VC	1.0	16.77	16.50	2.515309
VC	2.0	26.14	25.95	4.797731

```
ggplot(data = inner_join(tg.df, tg.summary), aes(x = Dose, y = Length, fill = Supplement)) +
    geom_violin(alpha = 0.5) +
    geom_point(aes(y = Mean, group = Supplement, color = Supplement), shape = 5, size = 2) +
    geom_line(aes(y = Mean, group = Supplement, color = Supplement)) +
    ylab(label = "Tooth Growth Length") +
    xlab(label = "Dose (mg/day)")
```

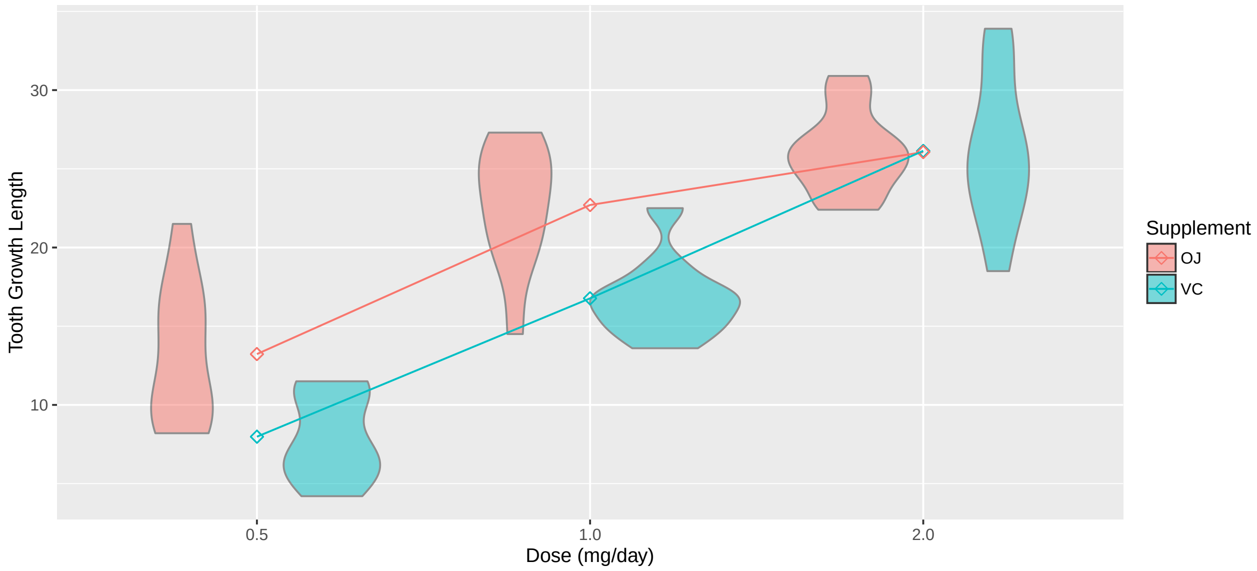


Figure 1: A violin-plot of Tooth Growth Length for each Dose, split by Supplement; the hollow point represents the mean

Taking this rudimentary anaylsis as an early indicator, it appears that Orange Juice maybe more effective at delivering Vitamnin-C to stimulate tooth-growth, but that becomes unclear at higher dosages.

Permutation Tests

To carry out the tests, I separate the observations by dose and establish the efficacy of Orange Juice versus Ascorbic Acid by permuting the delivery method against the fixed tooth-growth length.

By doing this I am stating my belief that there is no difference between delivery method of Vitamin-C within a given dosage.

```r
simulations = 10000

permute <- function(dose) {
    tg.subdata = ToothGrowth %>%
        filter(dose == dose) %>%
        mutate(dose = sprintf("%.1f", dose)) %>%
        unite(treatment, supp, dose)

    y = tg.subdata[,1]
    group = as.character(tg.subdata[,2])
    n = nrow(tg.subdata)

    oj = sprintf("OJ_%.1f", dose)
    vc = sprintf("VC_%.1f", dose)

    print(t.test(y[group == oj] - y[group == vc], paired = F, alternative = "t"))

    testStat <- function(w, g) mean(w[g == oj]) - mean(w[g == vc])
    observedStat <- testStat(y, group)
    permutations <- sapply(1:simulations, function(i) testStat(y, sample(group)))

    p.mean = mean(permutations)
    p.value = mean(permutations > observedStat)
    p.value.adjusted = p.adjust(mean(permutations > observedStat), method = "bonferroni")
    print(paste("P-value =", p.value, "\\nbonferroni(P-Value) =", p.value.adjusted))
    p.sd = sd(permutations)

    p.ci = p.mean + c(-1,1) * qt(.95, n - 2) * p.sd / sqrt(n)

    plot = ggplot(data = data.frame(permutations = permutations), aes(x = permutations)) +
        geom_histogram(color = "black", alpha = 0.5, fill = "lightblue") +
        geom_vline(aes(xintercept = observedStat), color = "black") +
        xlab(label = sprintf("Permutations for %.1f mg/day", dose))

    list(dose = dose, stat = observedStat, permutations = permutations,
         p.value = p.value, p.sd = p.sd, p.ci = p.ci, plot = plot)
}
```

```r
p0.5 = permute(0.5)
```

```
##
##  One Sample t-test
##
## data:  y[group == oj] - y[group == vc]
## t = 2.9791, df = 9, p-value = 0.01547
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.263458 9.236542
## sample estimates:
## mean of x
##      5.25
##
## [1] "P-value = 0.061 \\nbonferroni(P-Value) = 0.061"
```

```r
p1.0 = permute(1.0)
```

```
##
##  One Sample t-test
##
## data:  y[group == oj] - y[group == vc]
## t = 3.3721, df = 9, p-value = 0.008229
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1.951911 9.908089
## sample estimates:
## mean of x
##      5.93
##
## [1] "P-value = 0.0421 \\nbonferroni(P-Value) = 0.0421"
```

```r
p2.0 = permute(2.0)
```

```
##
##  One Sample t-test
##
## data:  y[group == oj] - y[group == vc]
## t = -0.042592, df = 9, p-value = 0.967
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -4.328976  4.168976
## sample estimates:
## mean of x
##     -0.08
##
## [1] "P-value = 0.5136 \\nbonferroni(P-Value) = 0.5136"
```

```r
grid.arrange(p0.5$plot, p1.0$plot, p2.0$plot, ncol=3)
```

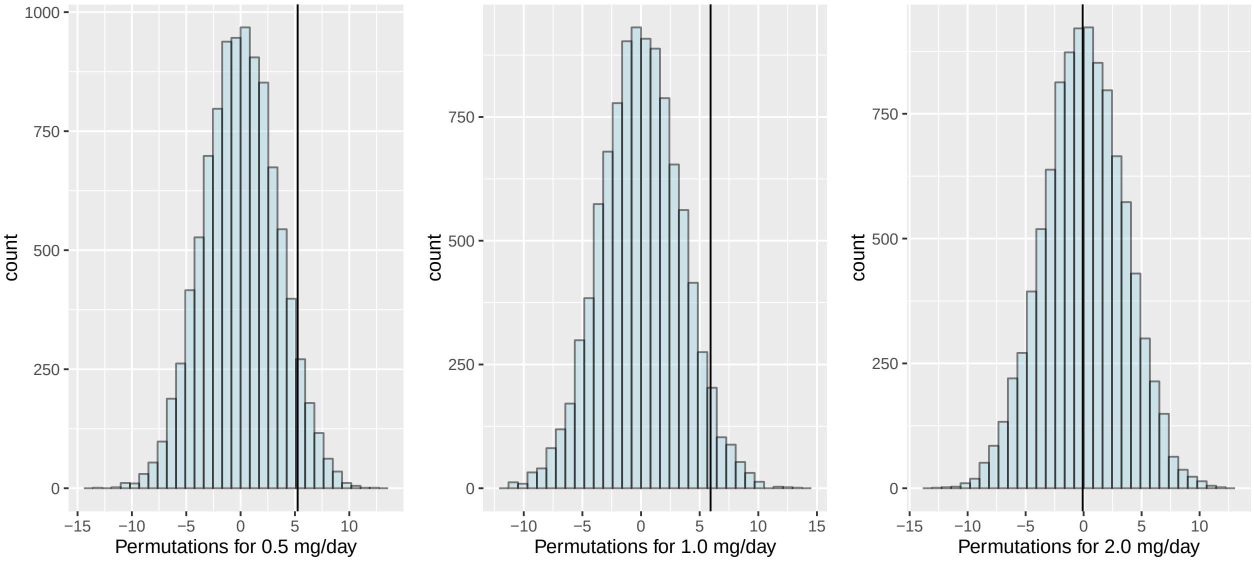


Figure 2: Histograms for the Null-distribution of permutation tests

Figure 2. shows the distribution of the 3 permutation tests, each with the corresponding observed test statistic plotted as the black vertical line.

None of the observed values look very extreme, particularly for the 2.0 mg/day test, where the observed test statistic falls at the centre of the distribution.

```r
p.summary = data.frame() %>% rbind(
    with(p0.5, c(sprintf("%.1f", dose), p.value, paste(p.ci, collapse = " to "), stat)),
    with(p1.0, c(sprintf("%.1f", dose), p.value, paste(p.ci, collapse = " to "), stat)),
    with(p2.0, c(sprintf("%.1f", dose), p.value, paste(p.ci, collapse = " to "), stat))
)

kable(p.summary, digits = 4, col.names = c("Dose", "P-value", "Confidence Interval", "Observed Statisti
```

Table 2: Summary of table containing P-values, Confidence Intervals and Observed Statistics for each Dose

Dose	P-value	Confidence Interval	Observed Statistic
0.5	0.061	-0.716211103106713 to 0.746805103106713	5.25
1.0	0.0421	-0.727093101787345 to 0.747703101787345	5.93
2.0	0.5136	-0.706620787368929 to 0.789790787368929	-0.0800000000000018

Table 2. shows P-values favouring the the Null-hypothesis, where I would accept the Null-hypothesis in 2 out 3 cases if I were to set the Type-I error rate $\alpha = 0.05$ and accept the Null-hypothesis in all cases where $\alpha = 0.01$.

Table 2. also shows the 95% Confidence Intervals i.e. Type-I error rate $\alpha = 0.05$; all intervals contain 0, supporting the posibilty that the delivery methods are equivalent, that is, one is not more effective than the other, thus further supporting the Null-hypothesis.

Conclusion

Based on this exploratory analysis and findings, I conclude that the Null-hypothesis is true.

P-Value Adjustment Example

```
require(graphics)

set.seed(123)
x <- rnorm(50, mean = c(rep(0, 25), rep(3, 25)))
p <- 2*pnorm(sort(-abs(x)))

round(p, 3)
```

```
##  [1] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001
## [12] 0.002 0.003 0.004 0.005 0.007 0.007 0.009 0.009 0.011 0.021 0.049
## [23] 0.061 0.063 0.074 0.083 0.086 0.119 0.189 0.206 0.221 0.286 0.305
## [34] 0.466 0.483 0.492 0.532 0.575 0.578 0.619 0.636 0.645 0.656 0.689
## [45] 0.719 0.818 0.827 0.897 0.912 0.944
```

```
round(p.adjust(p), 3)
```

```
##  [1] 0.000 0.001 0.001 0.005 0.005 0.006 0.006 0.007 0.009 0.016 0.024
## [12] 0.063 0.125 0.131 0.189 0.239 0.240 0.291 0.301 0.350 0.635 1.000
## [23] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
## [34] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
## [45] 1.000 1.000 1.000 1.000 1.000 1.000
```

```
round(p.adjust(p, "BH"), 3)
```

```
##  [1] 0.000 0.000 0.000 0.001 0.001 0.001 0.001 0.001 0.001 0.002 0.003
## [12] 0.007 0.013 0.013 0.017 0.021 0.021 0.024 0.025 0.028 0.050 0.112
## [23] 0.130 0.130 0.148 0.159 0.160 0.213 0.326 0.343 0.356 0.446 0.462
## [34] 0.684 0.684 0.684 0.719 0.741 0.741 0.763 0.763 0.763 0.763 0.782
## [45] 0.799 0.880 0.880 0.930 0.930 0.944
```

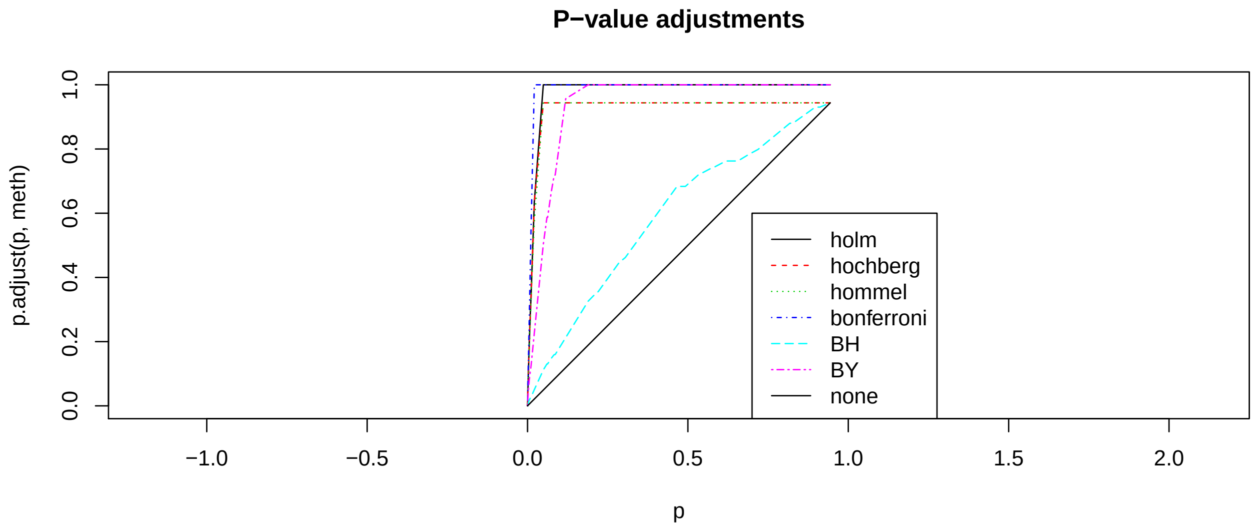
```
## or all of them at once (dropping the "fdr" alias):
p.adjust.M <- p.adjust.methods[p.adjust.methods != "fdr"]
p.adj    <- sapply(p.adjust.M, function(meth) p.adjust(p, meth))
p.adj.60 <- sapply(p.adjust.M, function(meth) p.adjust(p, meth, n = 60))
stopifnot(identical(p.adj[,"none"], p), p.adj <= p.adj.60)
round(p.adj, 3)
```

```
##        holm hochberg hommel bonferroni    BH    BY  none
##  [1,] 0.000    0.000  0.000      0.000 0.000 0.000 0.000
##  [2,] 0.001    0.001  0.001      0.001 0.000 0.002 0.000
##  [3,] 0.001    0.001  0.001      0.001 0.000 0.002 0.000
##  [4,] 0.005    0.005  0.004      0.005 0.001 0.004 0.000
##  [5,] 0.005    0.005  0.005      0.005 0.001 0.004 0.000
##  [6,] 0.006    0.006  0.005      0.006 0.001 0.004 0.000
##  [7,] 0.006    0.006  0.006      0.007 0.001 0.004 0.000
##  [8,] 0.007    0.007  0.007      0.008 0.001 0.004 0.000
##  [9,] 0.009    0.009  0.009      0.011 0.001 0.006 0.000
## [10,] 0.016    0.016  0.015      0.019 0.002 0.009 0.000
## [11,] 0.024    0.024  0.024      0.031 0.003 0.013 0.001
## [12,] 0.063    0.063  0.058      0.081 0.007 0.030 0.002
## [13,] 0.125    0.125  0.109      0.165 0.013 0.057 0.003
## [14,] 0.131    0.131  0.117      0.177 0.013 0.057 0.004
## [15,] 0.189    0.189  0.168      0.262 0.017 0.079 0.005
## [16,] 0.239    0.239  0.212      0.342 0.021 0.093 0.007
## [17,] 0.240    0.240  0.219      0.353 0.021 0.093 0.007
## [18,] 0.291    0.291  0.273      0.440 0.024 0.110 0.009
## [19,] 0.301    0.301  0.291      0.470 0.025 0.111 0.009
## [20,] 0.350    0.350  0.339      0.565 0.028 0.127 0.011
## [21,] 0.635    0.635  0.571      1.000 0.050 0.227 0.021
## [22,] 1.000    0.944  0.944      1.000 0.112 0.503 0.049
## [23,] 1.000    0.944  0.944      1.000 0.130 0.587 0.061
## [24,] 1.000    0.944  0.944      1.000 0.130 0.587 0.063
## [25,] 1.000    0.944  0.944      1.000 0.148 0.665 0.074
## [26,] 1.000    0.944  0.944      1.000 0.159 0.717 0.083
## [27,] 1.000    0.944  0.944      1.000 0.160 0.719 0.086
## [28,] 1.000    0.944  0.944      1.000 0.213 0.957 0.119
## [29,] 1.000    0.944  0.944      1.000 0.326 1.000 0.189
## [30,] 1.000    0.944  0.944      1.000 0.343 1.000 0.206
## [31,] 1.000    0.944  0.944      1.000 0.356 1.000 0.221
## [32,] 1.000    0.944  0.944      1.000 0.446 1.000 0.286
## [33,] 1.000    0.944  0.944      1.000 0.462 1.000 0.305
## [34,] 1.000    0.944  0.944      1.000 0.684 1.000 0.466
## [35,] 1.000    0.944  0.944      1.000 0.684 1.000 0.483
## [36,] 1.000    0.944  0.944      1.000 0.684 1.000 0.492
## [37,] 1.000    0.944  0.944      1.000 0.719 1.000 0.532
## [38,] 1.000    0.944  0.944      1.000 0.741 1.000 0.575
## [39,] 1.000    0.944  0.944      1.000 0.741 1.000 0.578
## [40,] 1.000    0.944  0.944      1.000 0.763 1.000 0.619
## [41,] 1.000    0.944  0.944      1.000 0.763 1.000 0.636
## [42,] 1.000    0.944  0.944      1.000 0.763 1.000 0.645
## [43,] 1.000    0.944  0.944      1.000 0.763 1.000 0.656
## [44,] 1.000    0.944  0.944      1.000 0.782 1.000 0.689
## [45,] 1.000    0.944  0.944      1.000 0.799 1.000 0.719
## [46,] 1.000    0.944  0.944      1.000 0.880 1.000 0.818
## [47,] 1.000    0.944  0.944      1.000 0.880 1.000 0.827
## [48,] 1.000    0.944  0.944      1.000 0.930 1.000 0.897
## [49,] 1.000    0.944  0.944      1.000 0.930 1.000 0.912
## [50,] 1.000    0.944  0.944      1.000 0.944 1.000 0.944
```

```
## or a bit nicer:
noquote(apply(p.adj, 2, format.pval, digits = 3))
```

```
##       holm  hochberg hommel bonferroni BH    BY    none
##  [1,] 0     0        0      0          0     0     0
##  [2,] 0.001 0.001    0.001  0.001      0.000 0.002 0
##  [3,] 0.001 0.001    0.001  0.001      0.000 0.002 0
##  [4,] 0.005 0.005    0.004  0.005      0.001 0.004 0
##  [5,] 0.005 0.005    0.005  0.005      0.001 0.004 0.000
##  [6,] 0.006 0.006    0.005  0.006      0.001 0.004 0.000
##  [7,] 0.006 0.006    0.006  0.007      0.001 0.004 0.000
##  [8,] 0.007 0.007    0.007  0.008      0.001 0.004 0.000
##  [9,] 0.009 0.009    0.009  0.011      0.001 0.006 0.000
## [10,] 0.016 0.016    0.015  0.019      0.002 0.009 0.000
## [11,] 0.024 0.024    0.024  0.031      0.003 0.013 0.001
## [12,] 0.063 0.063    0.058  0.081      0.007 0.030 0.002
## [13,] 0.125 0.125    0.109  0.165      0.013 0.057 0.003
## [14,] 0.131 0.131    0.117  0.177      0.013 0.057 0.004
## [15,] 0.189 0.189    0.168  0.262      0.017 0.079 0.005
## [16,] 0.239 0.239    0.212  0.342      0.021 0.093 0.007
## [17,] 0.240 0.240    0.219  0.353      0.021 0.093 0.007
## [18,] 0.291 0.291    0.273  0.440      0.024 0.110 0.009
## [19,] 0.301 0.301    0.291  0.470      0.025 0.111 0.009
## [20,] 0.350 0.350    0.339  0.565      0.028 0.127 0.011
## [21,] 0.635 0.635    0.571  1.000      0.050 0.227 0.021
## [22,] 1.000 0.944    0.944  1.000      0.112 0.503 0.049
## [23,] 1.000 0.944    0.944  1.000      0.130 0.587 0.061
## [24,] 1.000 0.944    0.944  1.000      0.130 0.587 0.063
## [25,] 1.000 0.944    0.944  1.000      0.148 0.665 0.074
## [26,] 1.000 0.944    0.944  1.000      0.159 0.717 0.083
## [27,] 1.000 0.944    0.944  1.000      0.160 0.719 0.086
## [28,] 1.000 0.944    0.944  1.000      0.213 0.957 0.119
## [29,] 1.000 0.944    0.944  1.000      0.326 1.000 0.189
## [30,] 1.000 0.944    0.944  1.000      0.343 1.000 0.206
## [31,] 1.000 0.944    0.944  1.000      0.356 1.000 0.221
## [32,] 1.000 0.944    0.944  1.000      0.446 1.000 0.286
## [33,] 1.000 0.944    0.944  1.000      0.462 1.000 0.305
## [34,] 1.000 0.944    0.944  1.000      0.684 1.000 0.466
## [35,] 1.000 0.944    0.944  1.000      0.684 1.000 0.483
## [36,] 1.000 0.944    0.944  1.000      0.684 1.000 0.492
## [37,] 1.000 0.944    0.944  1.000      0.719 1.000 0.532
## [38,] 1.000 0.944    0.944  1.000      0.741 1.000 0.575
## [39,] 1.000 0.944    0.944  1.000      0.741 1.000 0.578
## [40,] 1.000 0.944    0.944  1.000      0.763 1.000 0.619
## [41,] 1.000 0.944    0.944  1.000      0.763 1.000 0.636
## [42,] 1.000 0.944    0.944  1.000      0.763 1.000 0.645
## [43,] 1.000 0.944    0.944  1.000      0.763 1.000 0.656
## [44,] 1.000 0.944    0.944  1.000      0.782 1.000 0.689
## [45,] 1.000 0.944    0.944  1.000      0.799 1.000 0.719
## [46,] 1.000 0.944    0.944  1.000      0.880 1.000 0.818
## [47,] 1.000 0.944    0.944  1.000      0.880 1.000 0.827
## [48,] 1.000 0.944    0.944  1.000      0.930 1.000 0.897
## [49,] 1.000 0.944    0.944  1.000      0.930 1.000 0.912
## [50,] 1.000 0.944    0.944  1.000      0.944 1.000 0.944
```

```r
## and a graphic:
matplot(p, p.adj, ylab="p.adjust(p, meth)", type = "l", asp = 1, lty = 1:6,
        main = "P-value adjustments")
legend(0.7, 0.6, p.adjust.M, col = 1:6, lty = 1:6)
```



```
## Can work with NA's:
pN <- p; iN <- c(46, 47); pN[iN] <- NA
pN.a <- sapply(p.adjust.M, function(meth) p.adjust(pN, meth))
## The smallest 20 P-values all affected by the NA's :
round((pN.a / p.adj)[1:20, ] , 4)
```

```
##          holm hochberg hommel bonferroni   BH     BY none
##  [1,] 0.9600   0.9600 0.9600       0.96 0.96 0.9514    1
##  [2,] 0.9592   0.9592 0.9583       0.96 0.96 0.9514    1
##  [3,] 0.9583   0.9583 0.9583       0.96 0.96 0.9514    1
##  [4,] 0.9574   0.9574 0.9535       0.96 0.96 0.9514    1
##  [5,] 0.9565   0.9565 0.9535       0.96 0.96 0.9514    1
##  [6,] 0.9556   0.9556 0.9524       0.96 0.96 0.9514    1
##  [7,] 0.9545   0.9545 0.9524       0.96 0.96 0.9514    1
##  [8,] 0.9535   0.9535 0.9524       0.96 0.96 0.9514    1
##  [9,] 0.9524   0.9524 0.9512       0.96 0.96 0.9514    1
## [10,] 0.9512   0.9512 0.9500       0.96 0.96 0.9514    1
## [11,] 0.9500   0.9500 0.9500       0.96 0.96 0.9514    1
## [12,] 0.9487   0.9487 0.9444       0.96 0.96 0.9514    1
## [13,] 0.9474   0.9474 0.9394       0.96 0.96 0.9514    1
## [14,] 0.9459   0.9459 0.9394       0.96 0.96 0.9514    1
## [15,] 0.9444   0.9444 0.9375       0.96 0.96 0.9514    1
## [16,] 0.9429   0.9429 0.9355       0.96 0.96 0.9514    1
## [17,] 0.9412   0.9412 0.9355       0.96 0.96 0.9514    1
## [18,] 0.9394   0.9394 0.9355       0.96 0.96 0.9514    1
## [19,] 0.9375   0.9375 0.9355       0.96 0.96 0.9514    1
## [20,] 0.9355   0.9355 0.9333       0.96 0.96 0.9514    1
```