Update to `Love-boost.R`: `eda.ksam` instead of `eda.2sam`

Thomas E. Love, Ph.D.

2017-11-26

Contents

What is this?

The `eda.2sam` function within the `Love-boost.R` script isn't working perfectly. This document suggests that you instead use the new `eda.ksam` function which doesn't have as many problems, at least for the remaining work you do this semester. The `Love-boost.R` file as posted on 2017-11-26 now includes both `eda.2sam` and the new `eda.ksam` functions, and we suggest you use `eda.ksam`. We'll get rid of `eda.2sam` entirely at the start of 2018.

Setup for this work

```
library(gridExtra); library(tidyverse)

source("Love-boost.R")
```

A Data Set to Demonstrate the Problem

Consider the following toy data set. It contains observations on 3 variables for 32 subjects. Specifically, we have a `subject` identification number (1-32), a `type` (low, middle, or high) and a quantitative `result` in each row.

```
test3sam <- read_csv("test3sam.csv")
```

```
Parsed with column specification:
cols(
  subject = col_integer(),
  type = col_character(),
  result = col_integer()
)
```

```
test3sam
```

```
# A tibble: 32 x 3
   subject   type result
     <int>  <chr>  <int>
 1       1    low     13
 2       2    low     12
 3       3    low     19
 4       4    low     16
 5       5    low     14
 6       6    low     16
 7       7    low     12
 8       8    low     12
 9       9 middle     23
10      10 middle     23
# ... with 22 more rows
```

A numerical summary of the `test3sam` data

We have three independent samples here, and we can summarize them as follows:

```
test3sam %>%
  group_by(type) %>%
  summarize(n = n(), mean = mean(result), sd = sd(result),
            median = median(result), min = min(result), max = max(result))
```

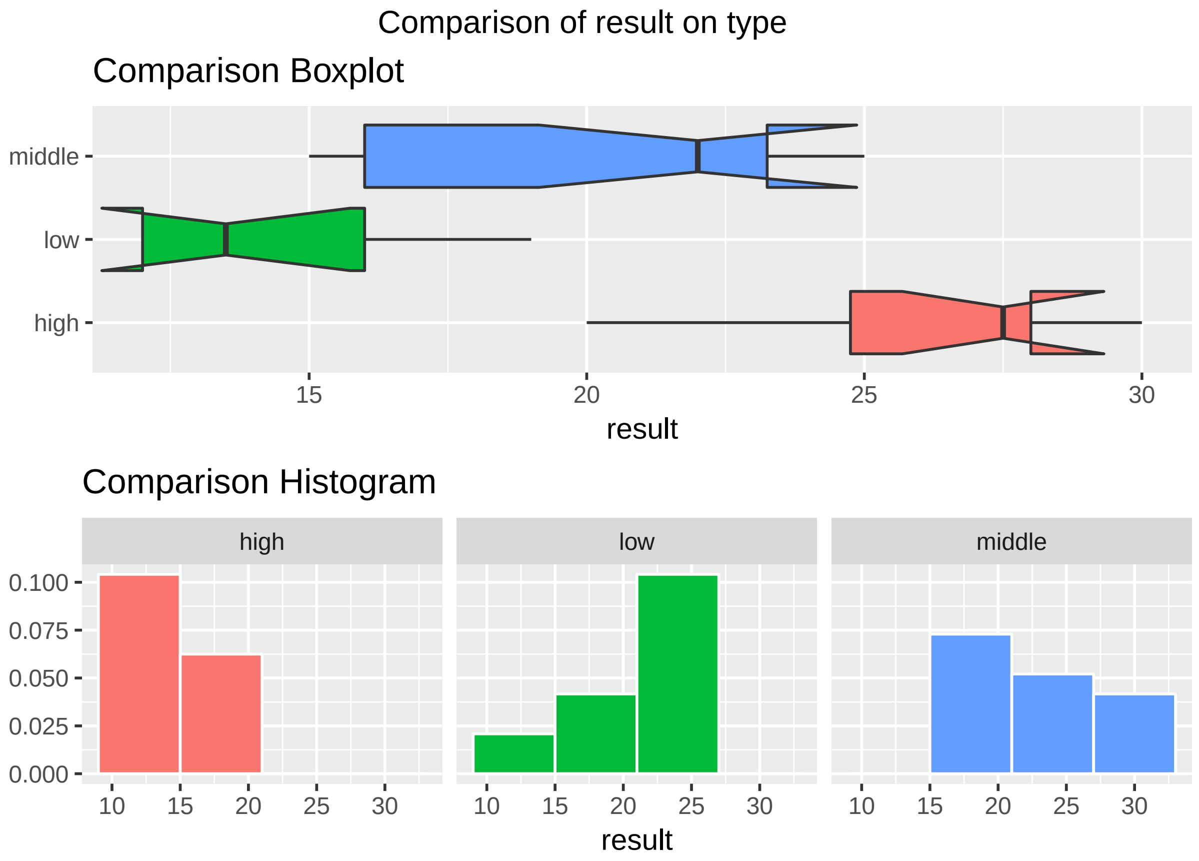
```
# A tibble: 3 x 7
    type     n   mean       sd median   min   max
   <chr> <int>  <dbl>    <dbl>  <dbl> <dbl> <dbl>
1   high     8 26.000 3.585686   27.5    20    30
2    low     8 14.250 2.549510   13.5    12    19
3 middle    16 20.375 3.703602   22.0    15    25
```

The Problem

The old `eda.2sam`, when applied to these data, incorrectly shows the data.

```
eda.2sam(outcome = test3sam$result,
         group = test3sam$type)
```

```
notch went outside hinges. Try setting notch=FALSE.
notch went outside hinges. Try setting notch=FALSE.
notch went outside hinges. Try setting notch=FALSE.
```



The data described as `high` in the Comparison Boxplot doesn't match the data described as `high` in the Comparison Histogram, and it looks like the boxplot may be correct, but the histogram certainly isn't, in terms of where the extreme values fall.

This is due to a problem with `eda.2sam` that becomes acute occasionally.

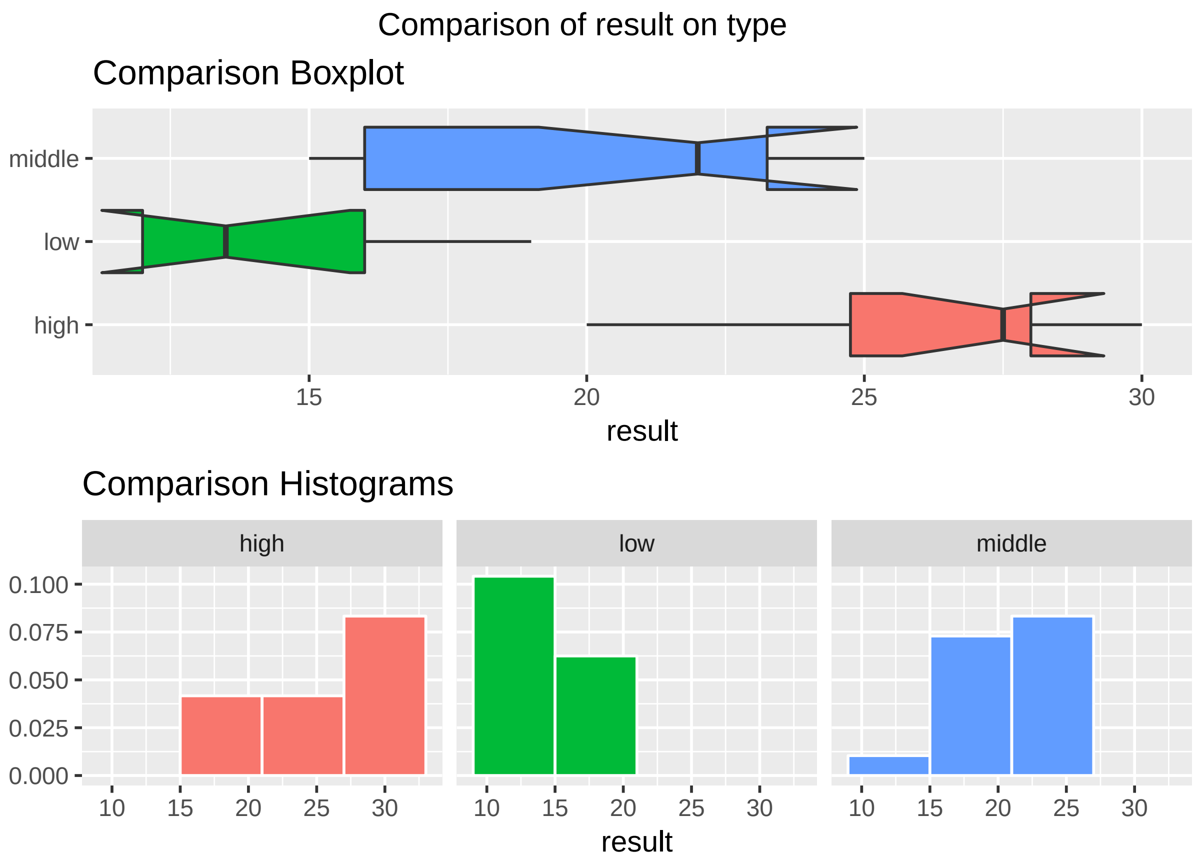
Also, you'll note the appropriate warnings for the `notch` because of the small number of observations in each group, but the suggestion (to drop the notching) is difficult to do, practically.

A Potential Solution: A new `eda.ksam` function

I've written a new function, called `eda.ksam`, now available in the `Love-boost.R` script (reposted on 2017-11-26), which seems to fix this problem. At any rate, we now get histograms and boxplots that match the data, and each other in this case.

```r
eda.ksam(outcome = test3sam$result,
         group = test3sam$type)
```

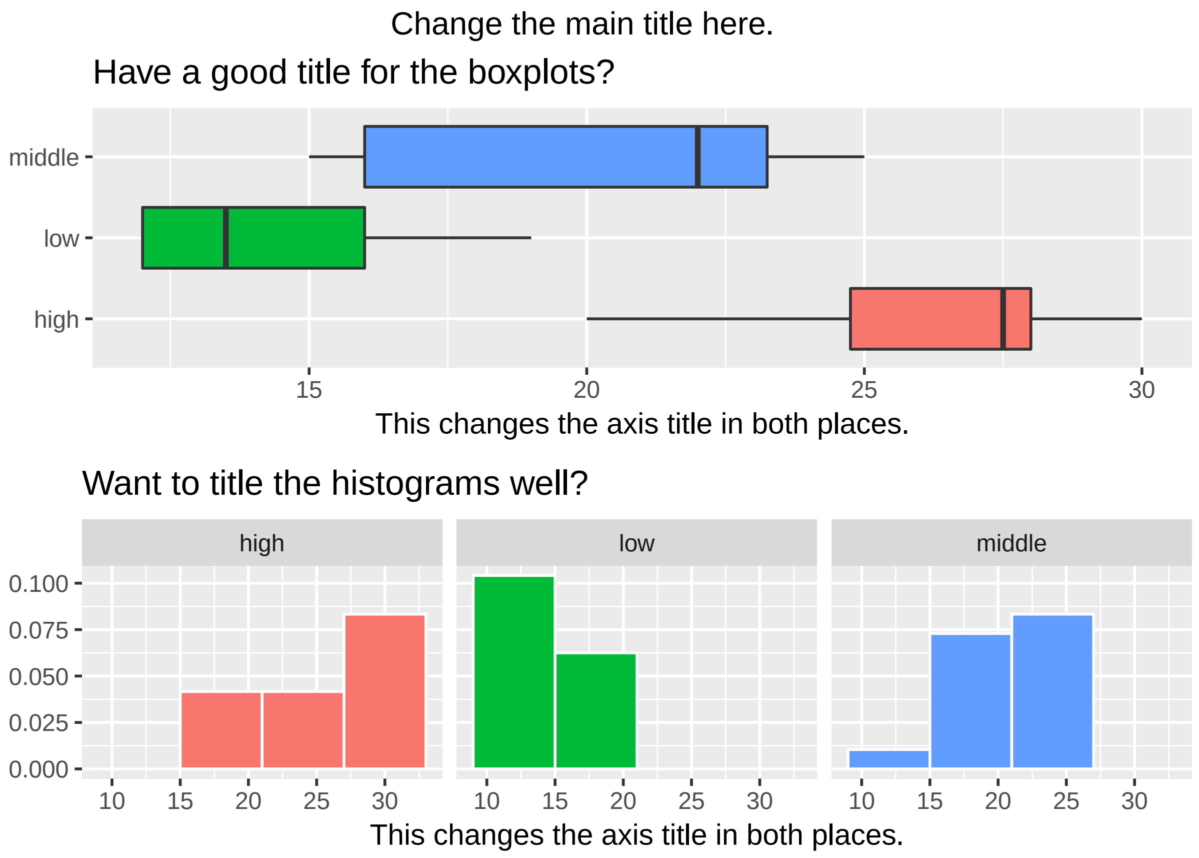
```
notch went outside hinges. Try setting notch=FALSE.
notch went outside hinges. Try setting notch=FALSE.
notch went outside hinges. Try setting notch=FALSE.
```



Other Reasons to Use `eda.ksam`

As a bonus, I've added the ability to change four different kinds of titles seamlessly, and to toggle off the notch behavior in the boxplots when we have small sample sizes. For example:

```r
eda.ksam(outcome = test3sam$result,
         group = test3sam$type,
         axis.title = "This changes the axis title in both places.",
         main.title = "Change the main title here.",
         boxplot.title = "Have a good title for the boxplots?",
         hist.title = "Want to title the histograms well?",
         notch = FALSE)
```



Revising the Levels of the type variable into a factor

It appears that if we clean up the `type` information by creating a factor (rather than a character variable) and reordering the levels from alphabetical order, all using the `fct_relevel` function, we get better results from `eda.2sam`.

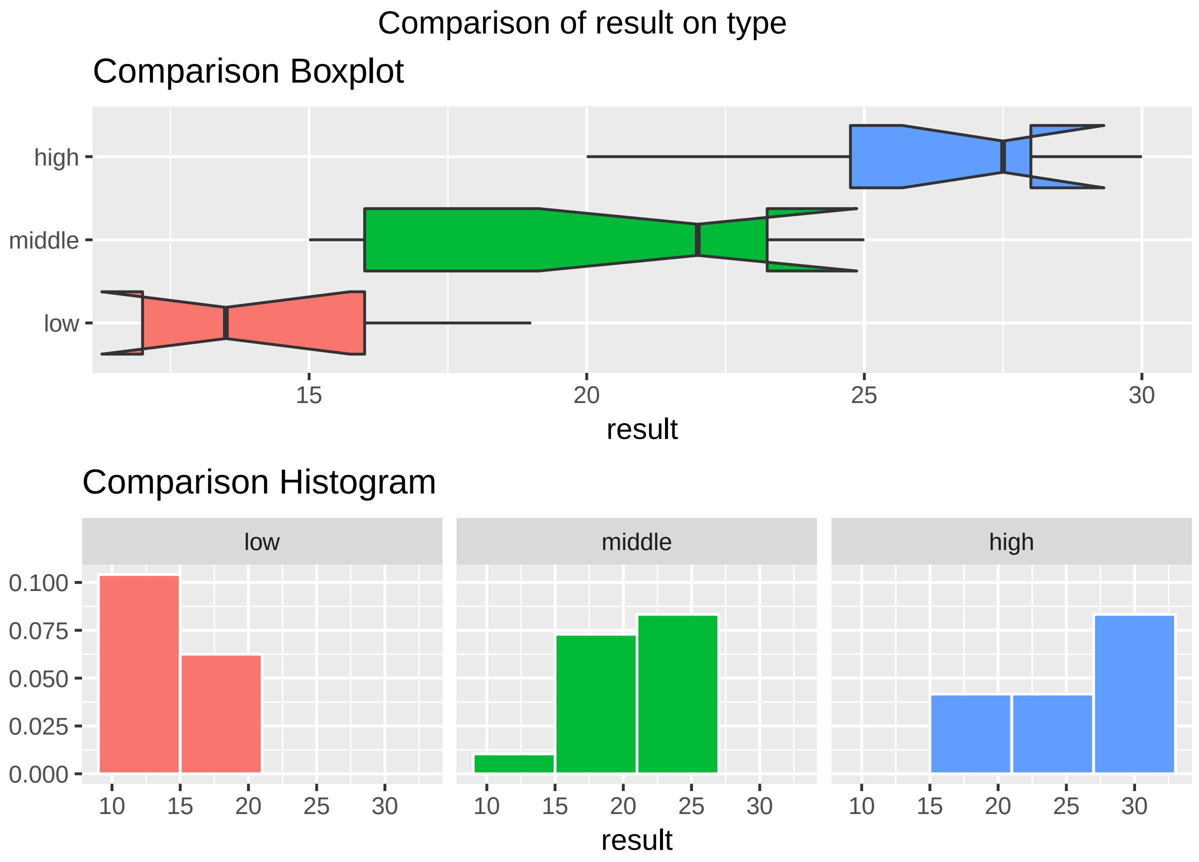
```
test3sam_repaired <- test3sam %>%
  mutate(type = fct_relevel(type, "low", "middle", "high"))
```

eda.2sam may work with factors better than characters

Here are the resulting `eda.2sam` graphs:

```
eda.2sam(test3sam_repaired$result, test3sam_repaired$type)
```

```
notch went outside hinges. Try setting notch=FALSE.
notch went outside hinges. Try setting notch=FALSE.
notch went outside hinges. Try setting notch=FALSE.
```

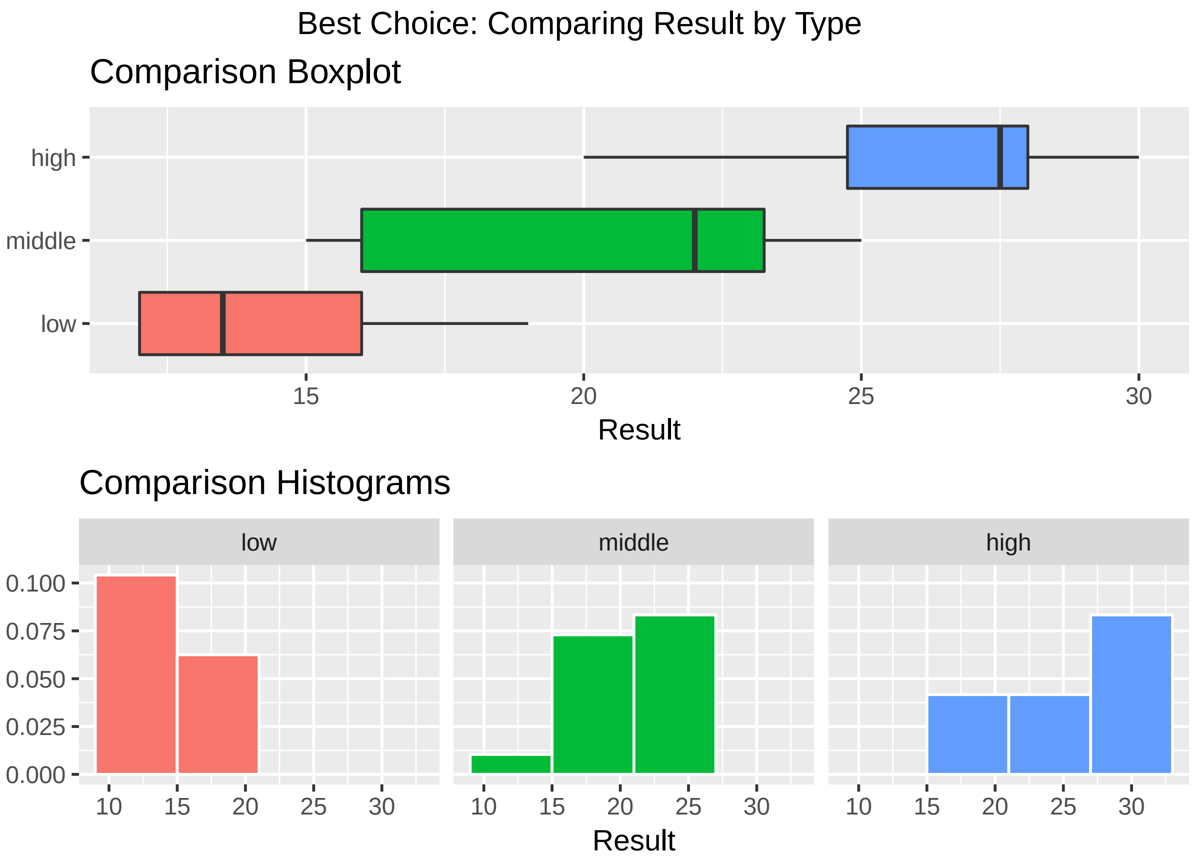


I am not certain that this solution (just building a factor and leveling it carefully) is sufficient to avoid the problem of misplaced data in all plots, though. So I'm still recommending the use of `eda.ksam`

Current Best Approach

The best solution I have now for this little problem is to use `eda.ksam` with the new, cleaner, presentation of the `type` factor, like this:

```r
eda.ksam(test3sam_repaired$result, test3sam_repaired$type,
         notch = FALSE, main.title = "Best Choice: Comparing Result by Type",
         axis.title = "Result")
```



Thank you for your attention.