exploration1-questions

Duu Renn

August 29, 2016

Questions

As people ask questions, I'll post responses here and push them to the shared folder. You can email me these questions or create an "issue" in the 'uips-stat-share' repository. Note that this is different than the 'explorations' repository – we should probably try to keep that relatively clutter-free.

Windsorize (Fabian)

Q: "Hi. I'm working with Hye Soo, and we get stuck on figuring out where the highest value from winsorize(hlp) (3.965) comes. We read about it, and we tried to calculate it but we couldn't understand how it was calculated."

First off, it's good that you read about the winsorize() command. Did you look in the documentation of robustHD or somewhere else? Remember, if you use online resources please put them in your response.

Anyways, let me first look the result.

```
winsorize(hlp)[1:10]
```

```
 hlphrs1  hlphrs2  hlphrs3  hlphrs4  hlphrs5  hlphrs6  hlphrs7  hlphrs8  hlphrs9 hlphrs10
   0.000    3.965    0.000    0.000    3.965    3.965    3.000    3.965    0.000    3.965
```

You report one number as a result, but clearly that isn't the case here. I include only the first 10 of what would have been 797 results. I think you meant this:

```
mean(winsorize(hlp))
```

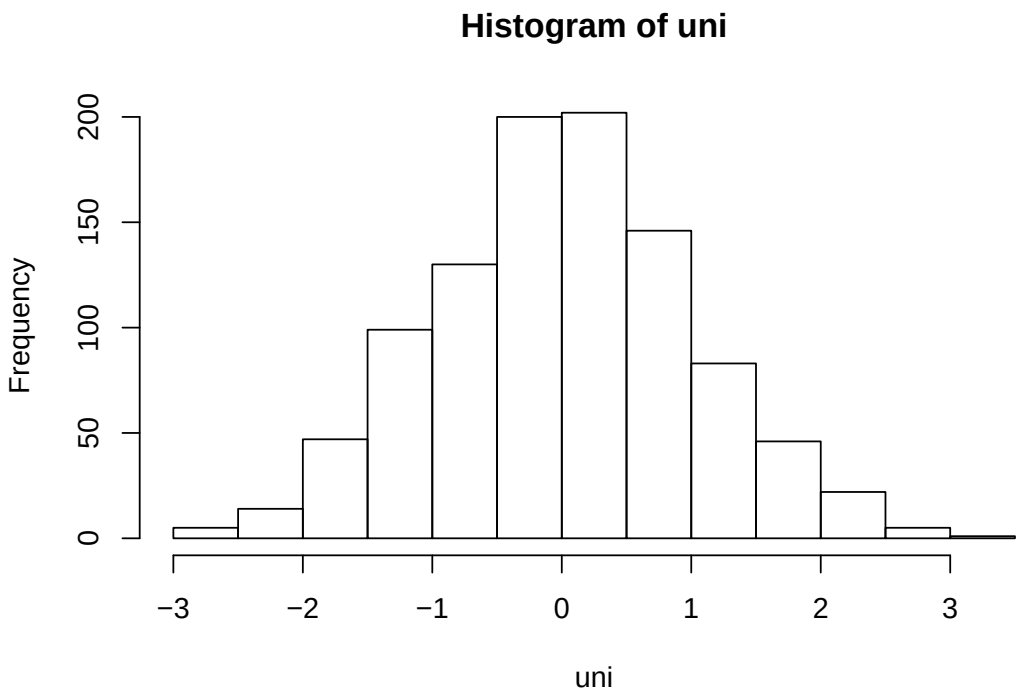
```
[1] 1.802
```

But even here, I receive a different result. Why is that?

Anyways, I suggest looking at the documentation and running your code again (all of it, probably) to see if your result (3.9) is actually what you're supposed to get. Maybe it's an issue with my version of the code and data – who knows. (You can know if you want, as my code is all online now.)

When you find out if the result is accurate, consider the following. Remember, this is a computational stats approach, not purely mathematical. So you may want to try to simulate things. Maybe helpful, maybe not.

```
set.seed(123)
uni <- rnorm(1000)
hist(uni)
```




```
sd(uni)
```

```
[1] 0.9917
```

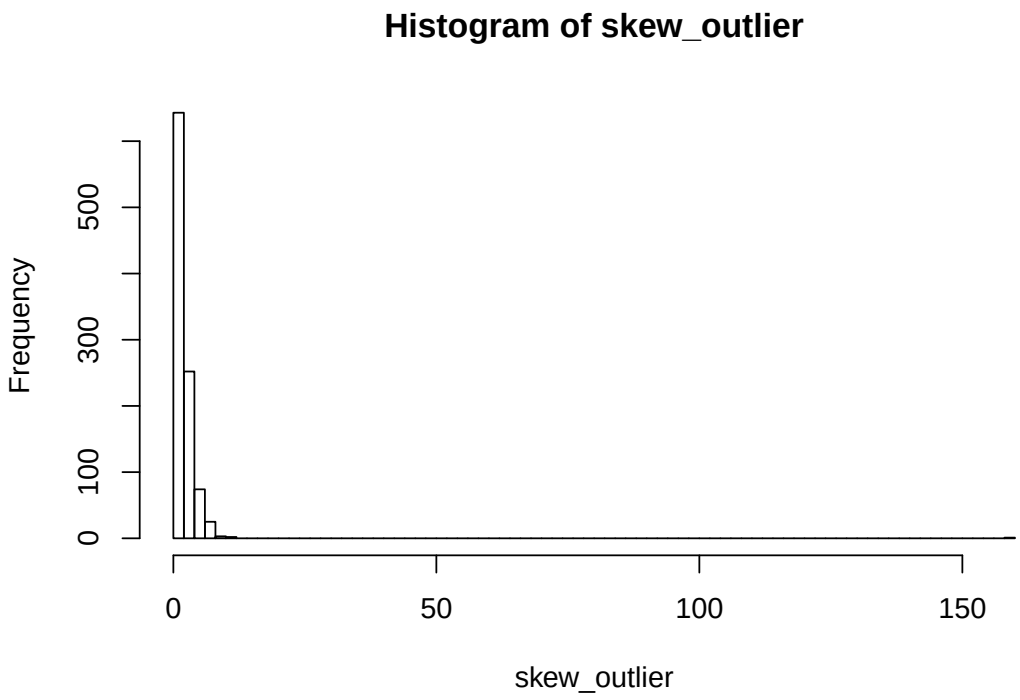
```
mean(uni)
```

```
[1] 0.01613
```

```
mean(uni)
```

```
[1] 0.01613
```

```
set.seed(123)
skew <- rnbinom(999, 5, 0.7)
skew_outlier <- append(160, skew)
hist(skew_outlier, breaks = 100)
```




```
sd(skew_outlier)
```

```
[1] 5.306
```

```
mean(skew_outlier)
```

```
[1] 2.369
```

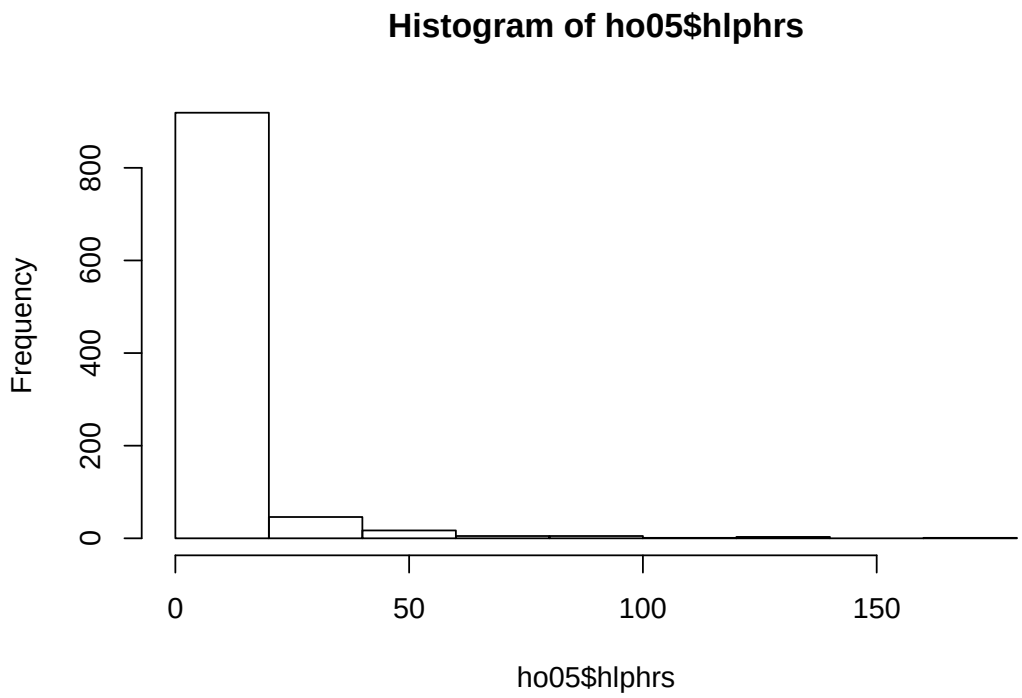
```
mean(winsorize(skew_outlier))
```

```
[1] 2.102
```

How to visualize

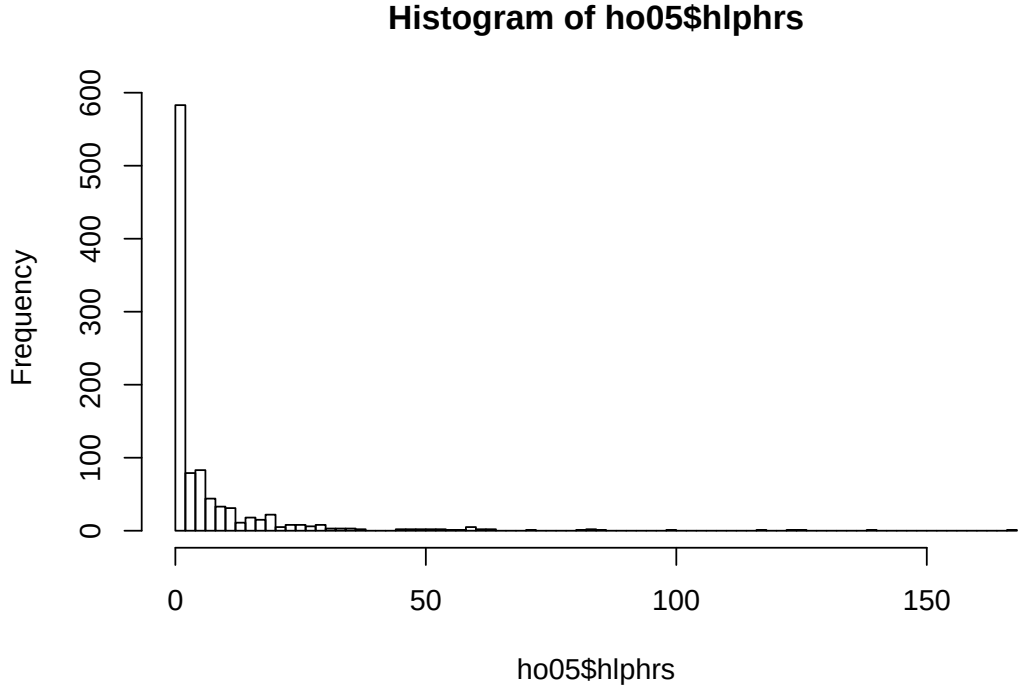
Well, you could use R's hist() function, like we did in the chunk above with the made up date.

```
hist(ho05$hlphrs)
```



But the amount of information conveyed by a plot really matters about design decisions. That is, visualization of data requires that you make decisions (or accept the default options).

```
hist(ho05$hlphrs, breaks = 100)
```

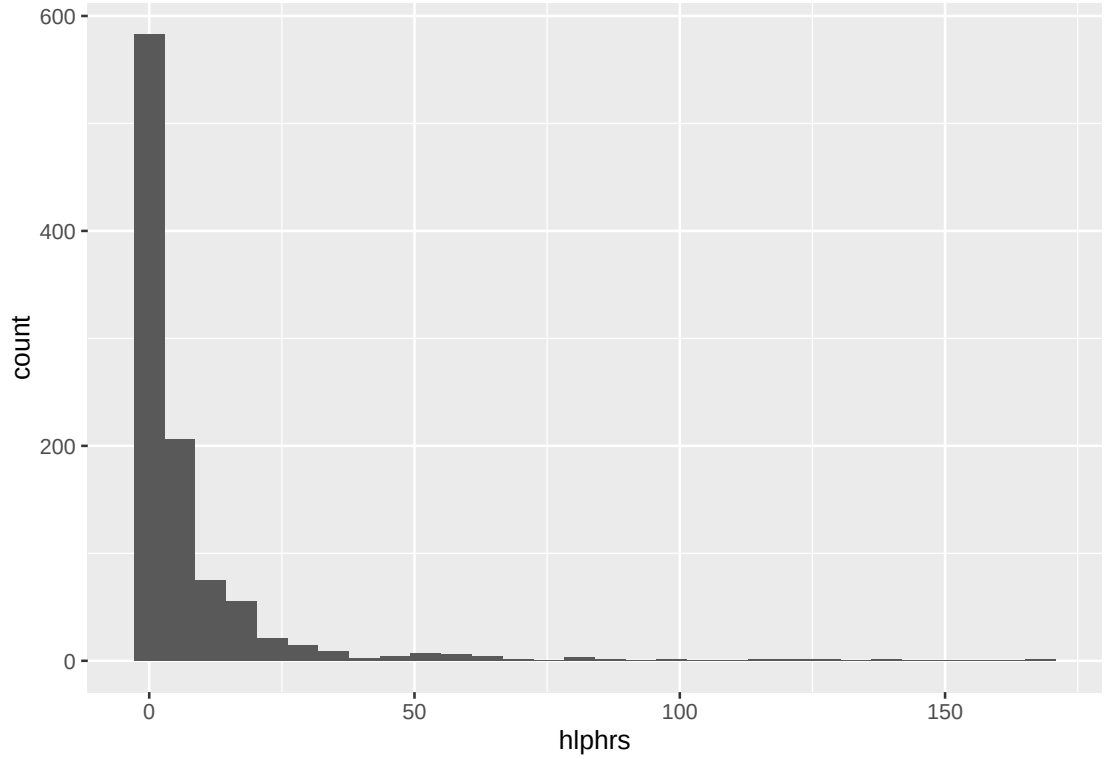


And of course, we could do some simple things like add color:

```
require(ggplot2)
ggplot(ho05, aes(x = hlphrs)) + geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

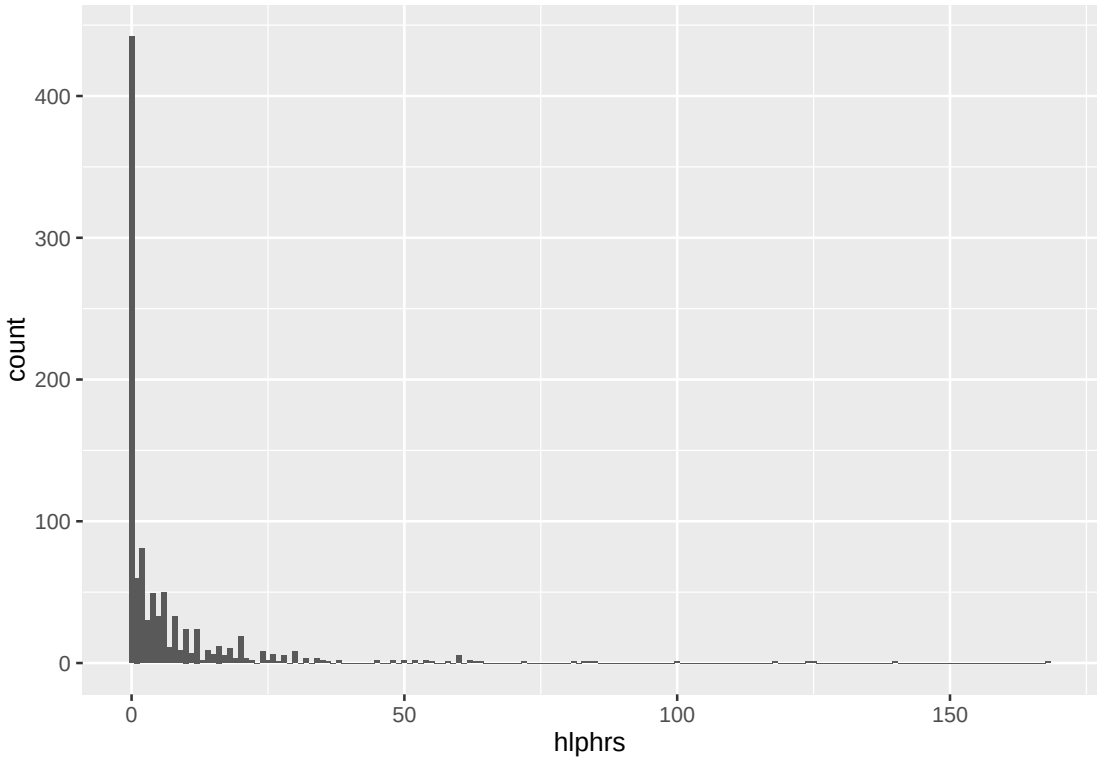
```
Warning: Removed 3 rows containing non-finite values (stat_bin).
```



Notice that the function actually suggests that you pick a better bin width, which is a little different (though not too difficult) in ggplot.

```
ggplot(ho05, aes(x = hlphrs)) + geom_histogram(binwidth = 1)
```

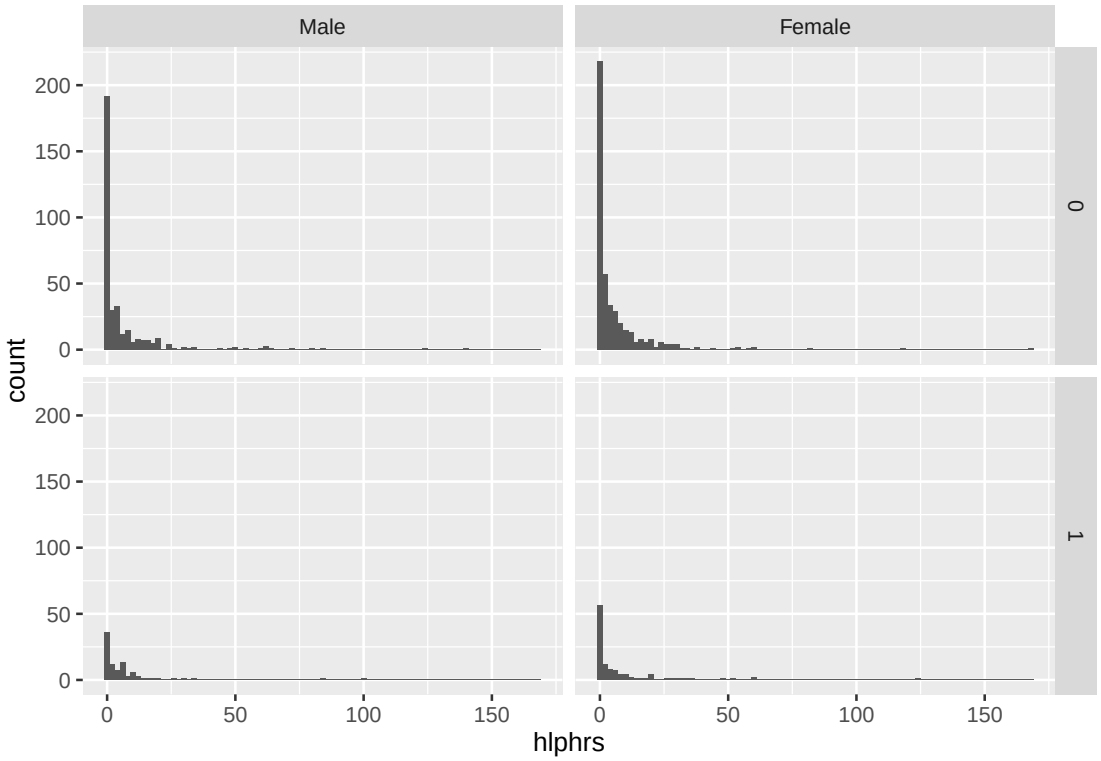
```
Warning: Removed 3 rows containing non-finite values (stat_bin).
```



In substantive terms, what is the bin width mean for our interpretation of the histogram?

Also, we can preview some other things too:

```
ggplot(subset(ho05, !is.na(hlphrs)), aes(x = hlphrs)) + geom_histogram(binwidth = 2) +
    facet_grid(postbomb ~ Rsex)
```


Male
Female
0
1
count
hlphrs
0
50
100
150
200