Dealing with Pathological Distributions

Steven Lowen

February 17, 2019

Problem

Some variables have distributions with many extreme values. Intuitive methods that work with well-behaved distributions fail. These types of variables require different statistical methods, and in some cases different statistics. Transformations can turn those variables into more well-behaved ones.

First, a well-behaved variable: Gaussian (normal)

Let's calculate the running average of a bunch of normally distributed randomly variables, with mean 10 and standard deviation 1.

```
npts <- 1e3
mu <- 10
stdev <- 1

set.seed(1234)
maindat <- data.frame(
        gauss = rnorm(npts, mean = 0, sd = stdev)) %>%
    mutate(
        shiftgauss = gauss + mu,
        n = row_number(),
        sgca = cummean(shiftgauss))
```

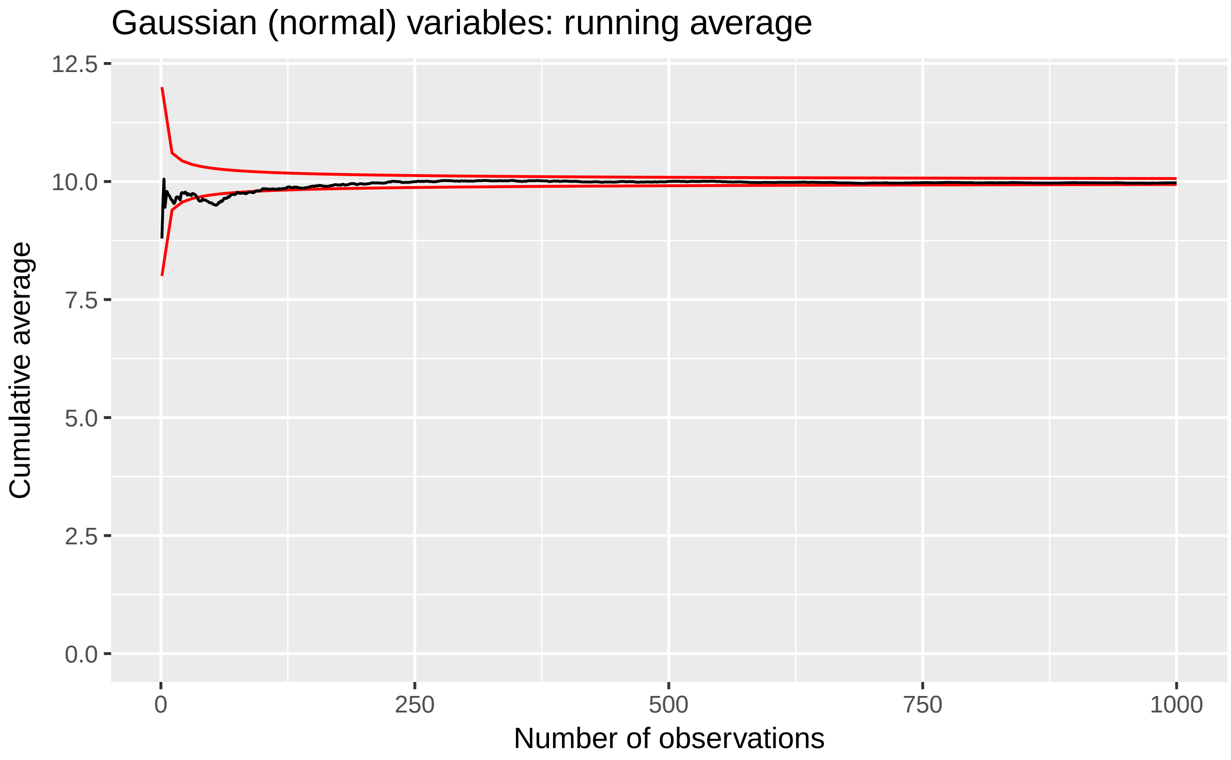
```
## Warning: `as_dictionary()` is soft-deprecated as of rlang 0.3.0.
## Please use `as_data_pronoun()` instead
## This warning is displayed once per session.
```

```
## Warning: `new_overscope()` is soft-deprecated as of rlang 0.2.0.
## Please use `new_data_mask()` instead
## This warning is displayed once per session.
```

```
## Warning: The `parent` argument of `new_data_mask()` is deprecated.
## The parent of the data mask is determined from either:
##
##   * The `env` argument of `eval_tidy()`
##   * Quosure environments when applicable
## This warning is displayed once per session.
```

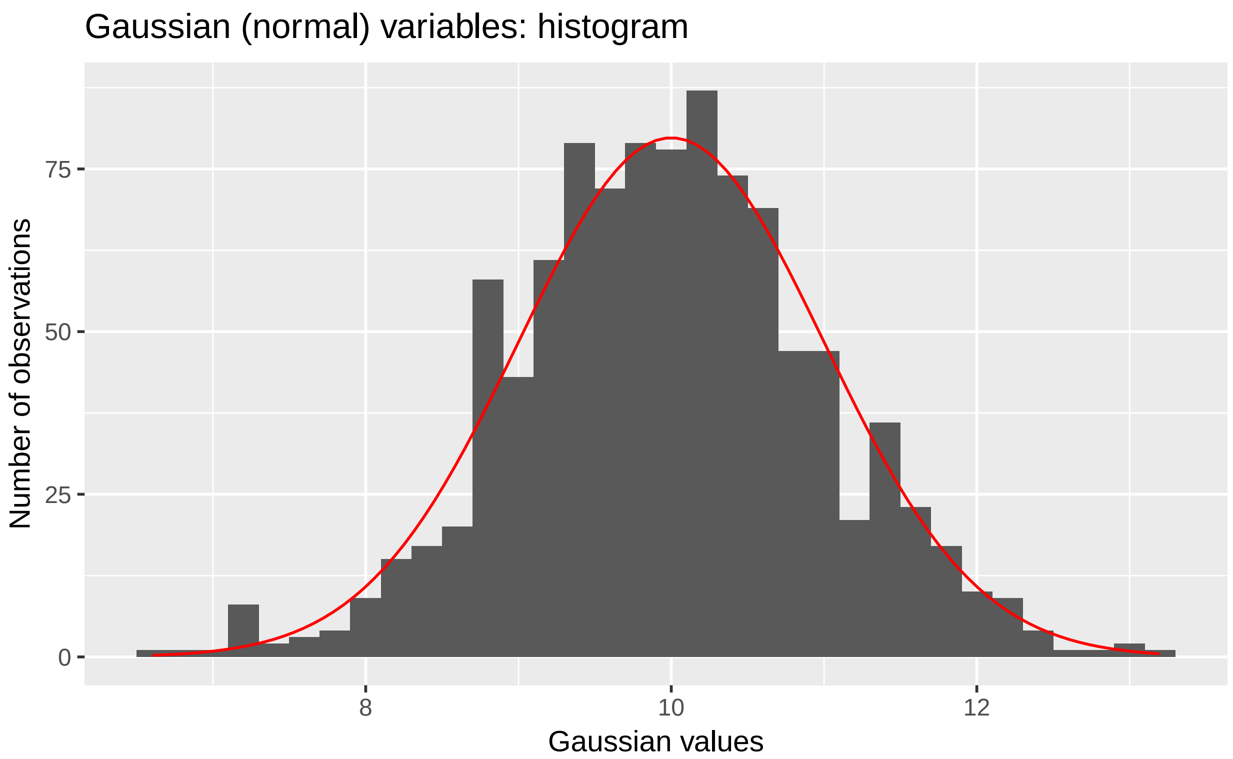
```
## Warning: `overscope_clean()` is soft-deprecated as of rlang 0.2.0.
## This warning is displayed once per session.
```

```
p1 <- maindat %>%
    ggplot +
    aes(x = n, y = sgca) +
    expand_limits(y = 0) +
    stat_function(fun = function(x) mu + 2.0 * stdev / sqrt(x),
        color = "red") +
    stat_function(fun = function(x) mu - 2.0 * stdev / sqrt(x),
        color = "red") +
    geom_line() +
    labs(x = "Number of observations", y = "Cumulative average",
        title = "Gaussian (normal) variables: running average")
print(p1)
```



We can look at the histogram, and see the familiar bell-shaped curve.

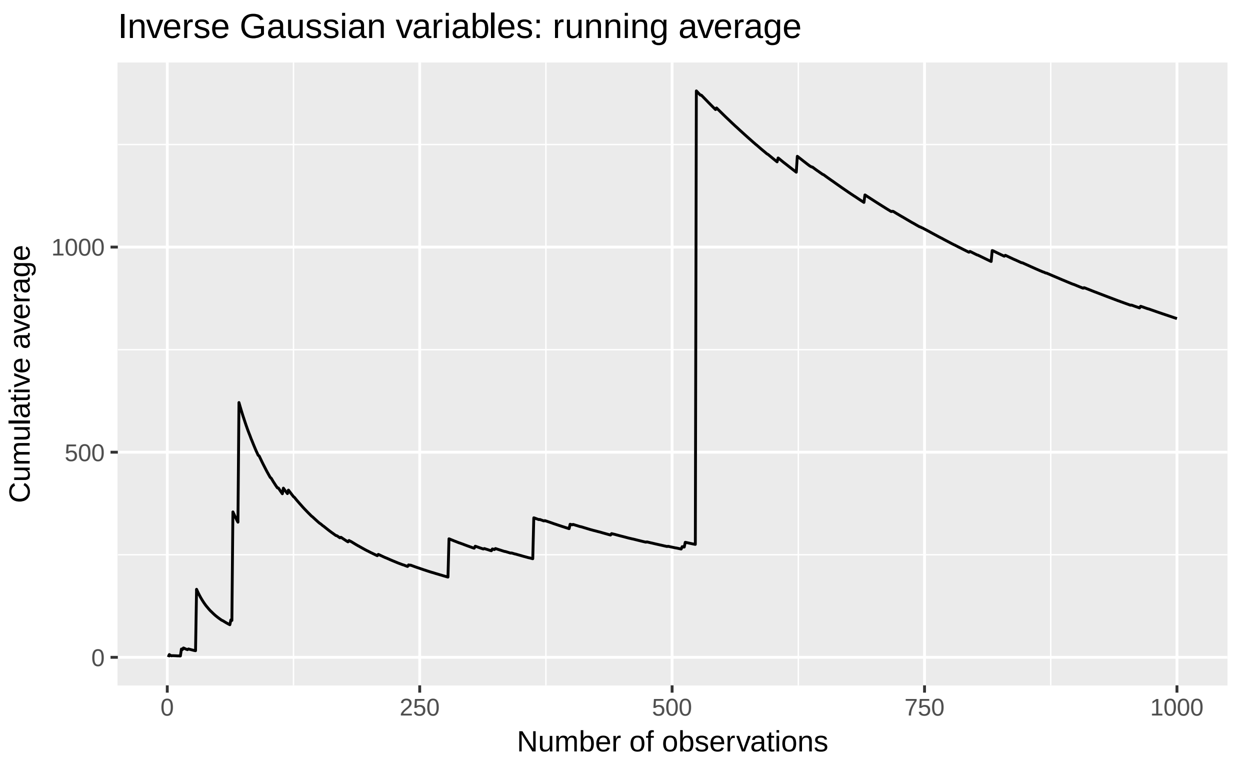
```
binwidth <- 0.2
gauhst <- function(x) dnorm(x, mean = mu, sd = stdev) * npts * binwidth

p2 <- ggplot(maindat) +
    aes(shiftgauss) +
    geom_histogram(binwidth = binwidth) +
    stat_function(fun = gauhst, color = "red") +
    labs(x = "Gaussian values", y = "Number of observations",
        title = "Gaussian (normal) variables: histogram")
print(p2)
```


Gaussian (normal) variables: histogram

Inverse Gaussian

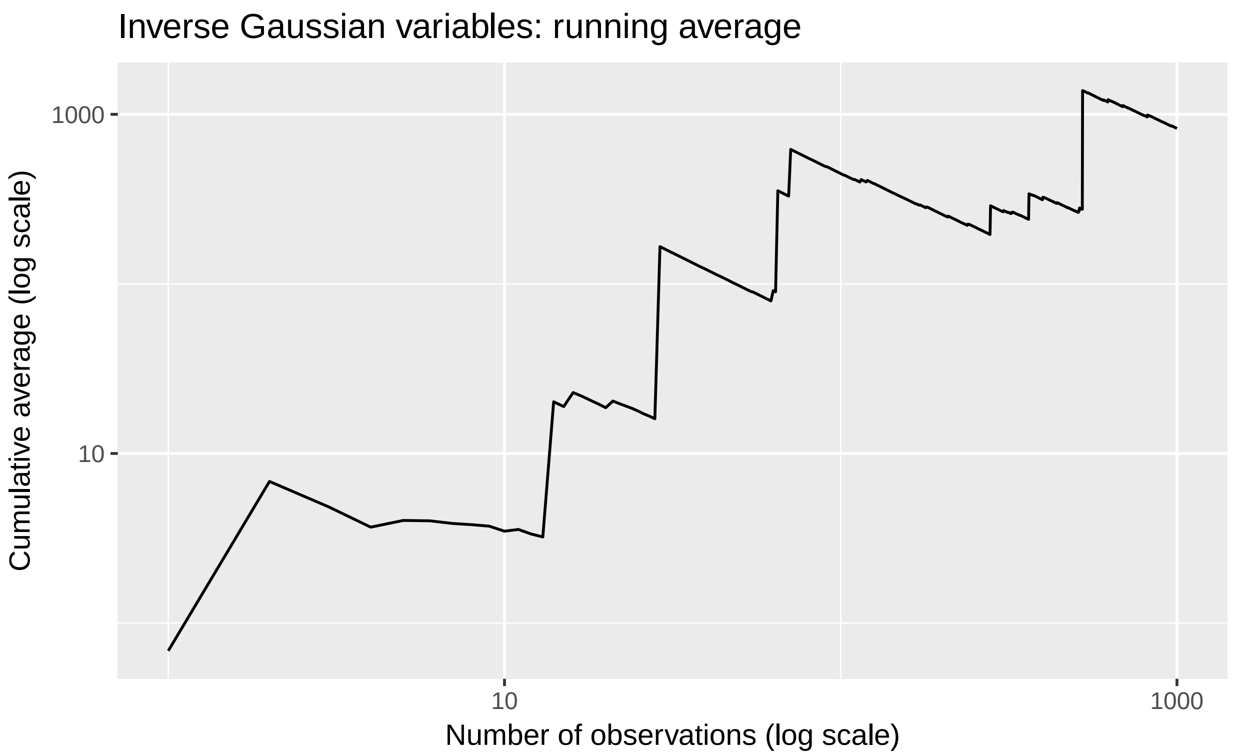
Now suppose we take zero mean Gaussian random variables, square them, take the inverse, and take the same running average. What happens then?

```
maindat <- maindat %>%
    mutate(
        invgs = 1.0 / gauss^2,
        igca = cummean(invgs))
p3 <- ggplot(maindat) +
    aes(x = n, y = igca) +
    geom_line() +
    labs(x = "Number of observations", y = "Cumulative average",
        title = "Inverse Gaussian variables: running average")
print(p3)
```



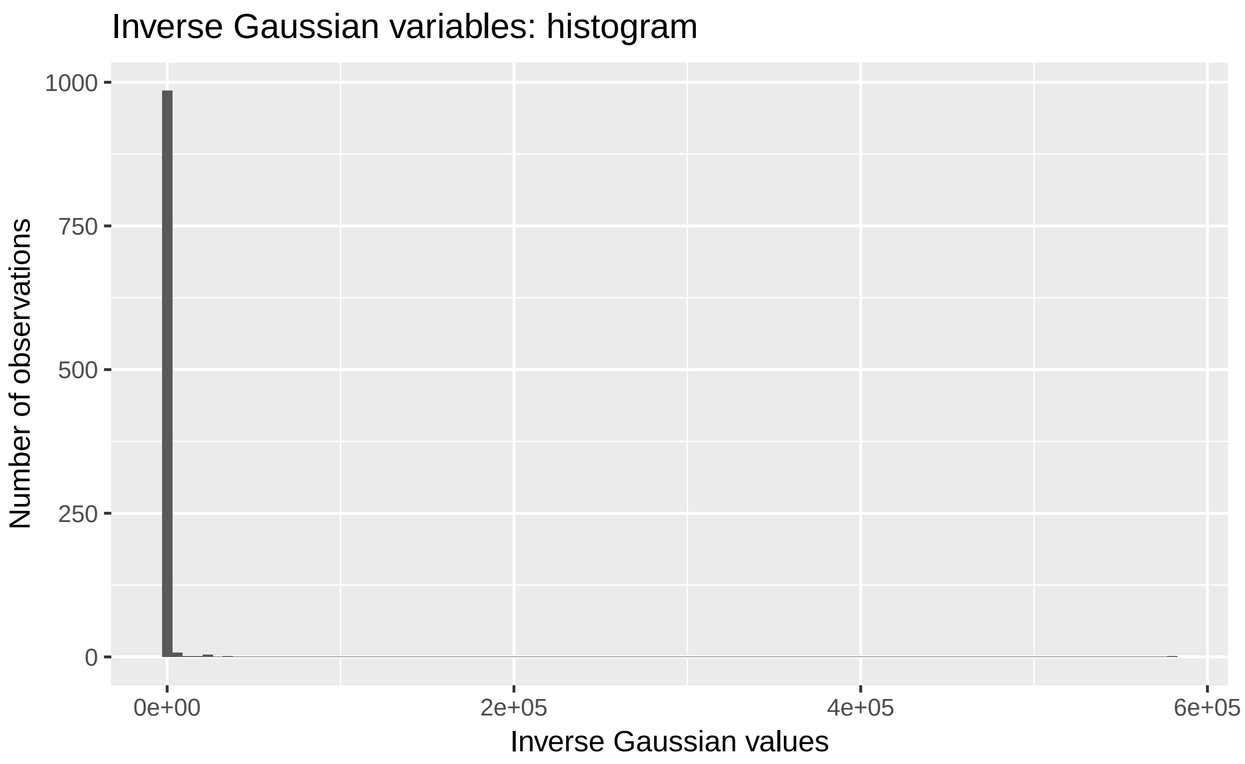
That doesn't look like it's settling down to an average value. How about on a log-log plot?

```
p4 <- p3 +
    scale_x_log10() +
    scale_y_log10() +
    labs(x = "Number of observations (log scale)",
        y = "Cumulative average (log scale)")
print(p4)
```



Now it looks like it's steadily increasing! How about the histogram?

```r
p5 <- ggplot(maindat) +
    aes(invgs) +
    geom_histogram(bins = 100) +
    labs(x = "Inverse Gaussian values", y = "Number of observations",
        title = "Inverse Gaussian variables: histogram")
print(p5)
```

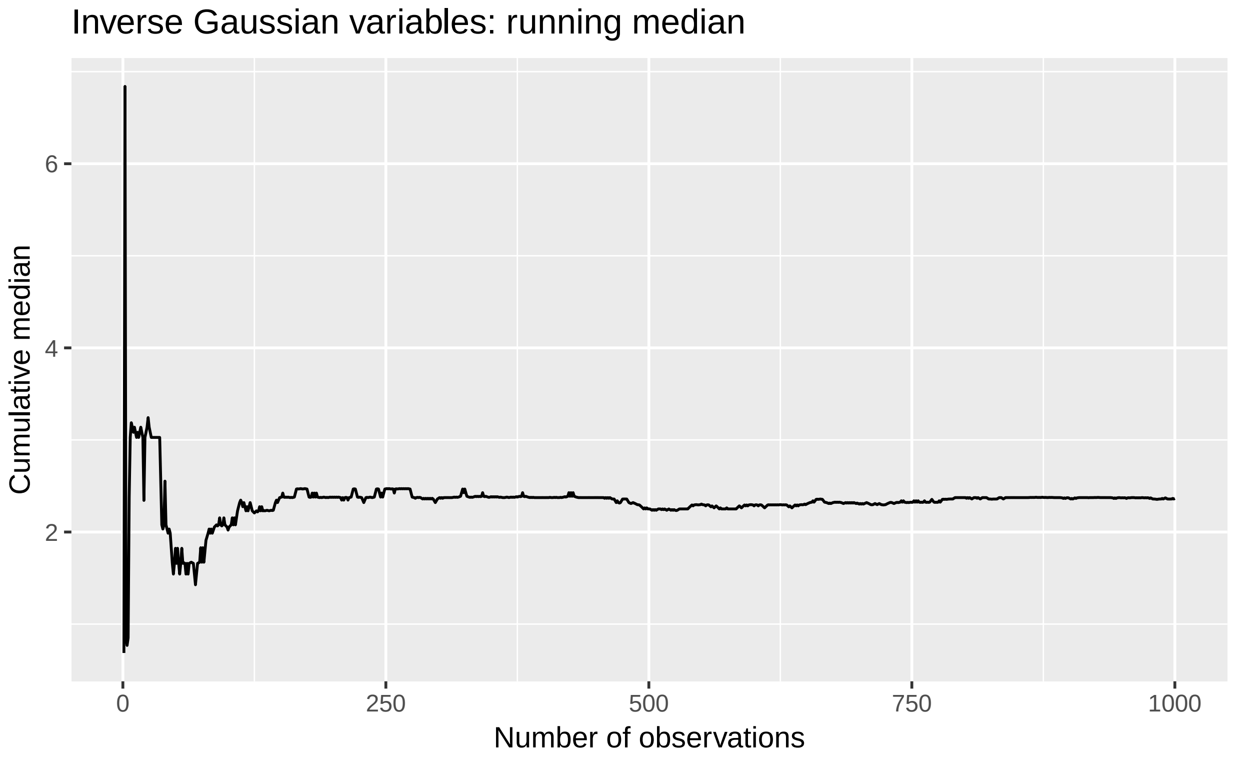


Now we know what the problem is: a distribution with long tails. In fact, for this random variable, the mean is infinite. It does not make sense to estimate something that is infinte.

Other statistics

Some statistics are well-behaved even when dealing with infinite means. The median is perhaps the best known. What would that look like?

```
maindat <- maindat %>%
    mutate(igcm = cummedian(invgs))

p6 <- ggplot(maindat) +
    aes(x = n, y = igcm) +
    geom_line() +
    labs(x = "Number of observations", y = "Cumulative median",
        title = "Inverse Gaussian variables: running median")
print(p6)
```



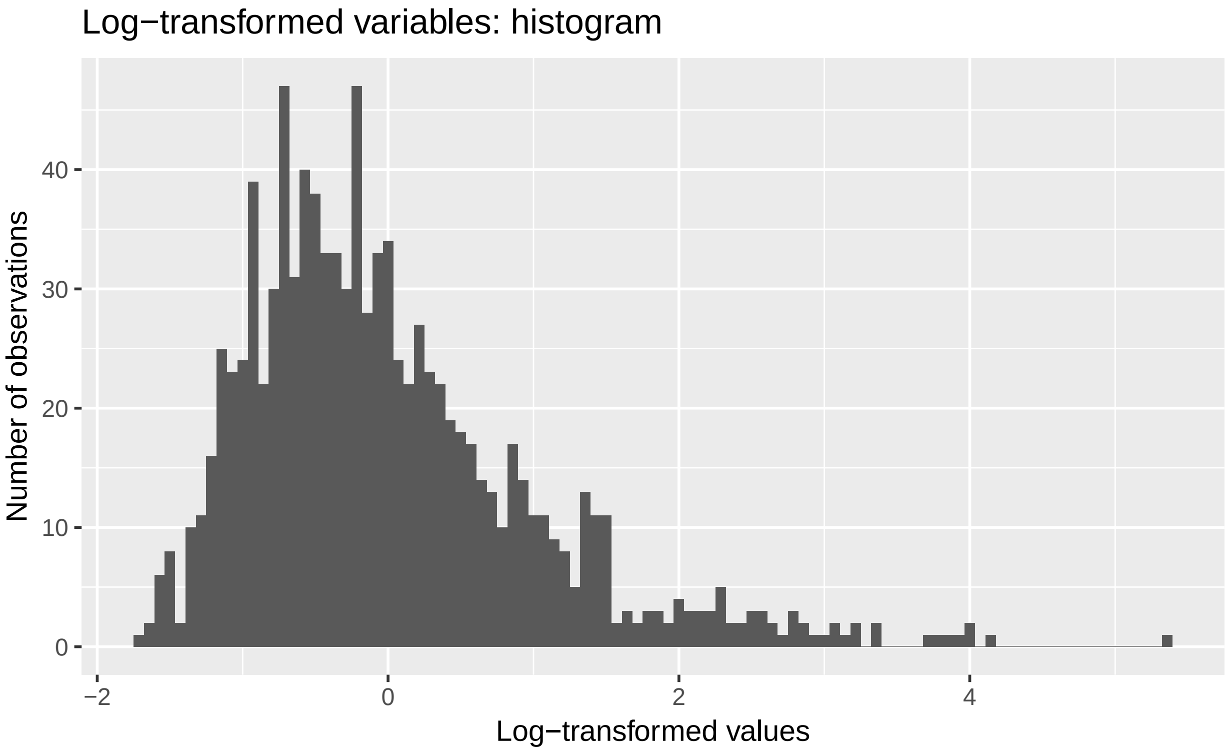
That settles down nicely. So one way to deal with random variabled that have extreme values is to use statistics that can deal with those values. Similarly, even when the mean is finite, the median is more robust

Transformations

Log transform

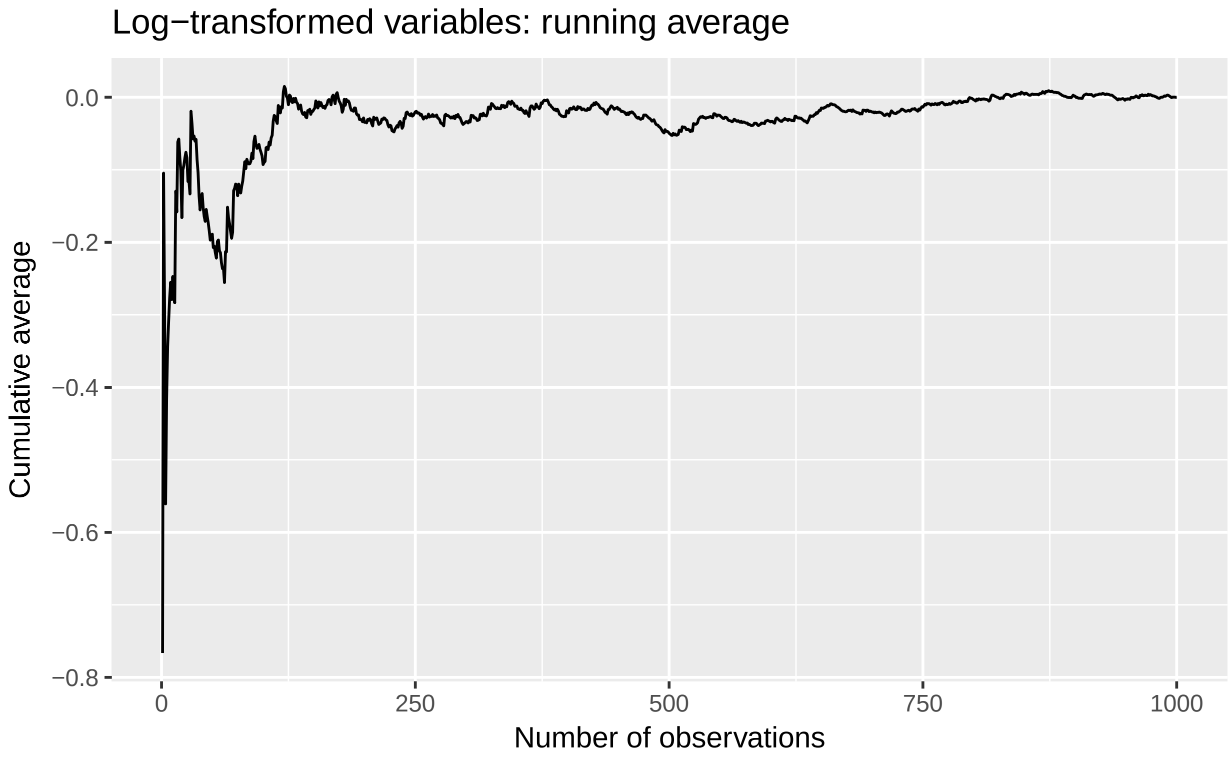
Let's pretend we don't know where the variable "invgs" came from, but we want to use it in a model that can't handle pathological variables. A standard trick is to take the logarithm of the variable.

```
maindat <- maindat %>%
    mutate(xfrm = log(invgs), xfrm = (xfrm - mean(xfrm)) / sd(xfrm))

p7 <- ggplot(maindat) +
        aes(xfrm) +
        geom_histogram(bins = 100) +
        labs(x = "Log-transformed values", y = "Number of observations",
                title = "Log-transformed variables: histogram")
print(p7)
```



That yields a histogram that doesn't match a Gaussian form perfectly, but it's much more well behaved. Because we scaled the output to have zero mean (and unit variance), the output converges towards zero. We can try a running average with the transformed variable.

```r
maindat <- maindat %>%
    mutate(xfca = cummean(xfrm))

p8 <- ggplot(maindat) +
    aes(x = n, y = xfca) +
    geom_line() +
    labs(x = "Number of observations", y = "Cumulative average",
        title = "Log-transformed variables: running average")
print(p8)
```



That looks reasonable.

Box-Cox transform

The log transform is a special case of a more general set that estimates the transform from the data. The Box-Cox transformation is popular. We use the caret package, which has a slightly different interface.

```
trans <- preProcess(maindat["invgs"], method=c("BoxCox", "center", "scale"))
maindat["boxcox"] <- predict(trans, maindat["invgs"])
maindat %>%
    mutate(dff = xfrm - boxcox) %>%
    summarize(low = min(dff), high = max(dff)) %>%
    print
```

```
##             low         high
## 1 -1.110223e-15 3.552714e-15
```

In this case, the results are identical to the log transform to within roundoff error, but in other cases the Box-Cox transformation would yield something closer to a Gaussian form than would the log transform.

Conclusion

Pathological variables can wreak havoc with analyses, some having annoying properties such as infinite means. Two methods can be used to tame them. First, we can use more robust statistics, and obtain different measures; here we look at central tendency (mean and median). Second, we can transform the variables so that the pathological properties are removed, at which point we can use standard measures such as the mean.