

Rescaling Bounded Parameter to be Unbounded and Positive and Negative Exponents with Different Bases

Fan Wang

2021-06-29

Contents

1	Log with Different Bases and Exponents	1
1.1	Log of Bases that Are not 10, 2 and e	1
1.2	Rescale Bounded Model Parameters to Unconstrained with Exponentiation	2
1.3	Positive Exponents	2
1.4	Negative Exponents	5
1.5	Inequality and Exponents	7

1 Log with Different Bases and Exponents

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools Package](#), [R Code Examples](#) Repository ([bookdown site](#)), or [Intro Stats with R](#) Repository ([bookdown site](#)).

1.1 Log of Bases that Are not 10, 2 and e

What is y below, with arbitrary base x ? It is $y = \frac{\log(z)}{\log(x)}$, because:

$$\begin{aligned}x^y &= z \\x^{\frac{\log(z)}{\log(x)}} &= z \\ \log \left(x^{\frac{\log(z)}{\log(x)}} \right) &= \log(z) \\ \frac{\log(z)}{\log(x)} \log(x) &= \log(z) \\ \frac{\log(z)}{\log(x)} &= \frac{\log(z)}{\log(x)}\end{aligned}$$

Given these, we can compute the exponents, y , for non-standard bases, x , given the value for z .

```
# base 1.1
x <- 1.1
y <- 5.5
z <- x^y
# given z and knowing x, and what is y?
y_solved <- log(z) / log(x)
# display
print(paste0("y_solved=", y_solved, ", y=", y))

## [1] "y_solved=5.5, y=5.5"
```

1.2 Rescale Bounded Model Parameters to Unconstrained with Exponentiation

We have a parameter to be estimated, the parameter's values can range between positive 1 and negative infinity. We want to use an estimator that is unconstrained. Use exponentiation to rescale the parameter so that it becomes unconstrained, use different bases so that the speed at which the parameter value approaches its bounds can be controlled.

While y is not bounded, $f(y; x)$ is bounded:

$$f(y; x) = 1 - x^y$$

where $x > 1$ and $-\infty < y < \infty$
then, $1 > f(y; x) > -\infty$

With $x > 1$, as y increases $f(y; x)$ decreases:

$$\frac{df(y; x)}{dy} = -x^y \log(x) < 0 \text{ when } x > 1$$

x controls the speed at which $f(y)$ approaches its bounds. In the simulation below, we try a number of different bases, at higher bases (2, e=2.71, 10), as y value changes $f(y)$ shifts too quickly to the bounds. But a base value of $x = 1.03$ or $x = 1.04$ would work well in an unbounded estimation routine that still generates parameters within bounds, which is below 1 in the case here.

```
# Vector of unbounded values, high and low
ar_y_vals <- sort(rnorm(20, 0, 20))
# Different base values
ar_bases <- c(1.01, 1.02, 1.03, 1.04, 1.1, 2, 2.71, 10)
# Transform back to f(y) scale with different bases
mt_f_of_y_vary_x <- matrix(NA,
  nrow = length(ar_y_vals),
  ncol = 1 + length(ar_bases)
)
ar_st_varnames <- c("yvalidx", "y_vals", paste0("base", ar_bases))
mt_f_of_y_vary_x[, 1] <- ar_y_vals
for (it_base in seq(1, length(ar_bases))) {
  fl_base <- ar_bases[it_base]
  ar_f_y <- 1 - fl_base^ar_y_vals
  mt_f_of_y_vary_x[, 1 + it_base] <- ar_f_y
}
# To tibble
tb_f_of_y_vary_x <- as_tibble(mt_f_of_y_vary_x) %>%
  rowid_to_column(var = "id") %>%
  rename_all(~ c(ar_st_varnames))
# Print
kable(tb_f_of_y_vary_x) %>% kable_styling_fc_wide()
```

1.3 Positive Exponents

Define exponents to consider and x-values to consider.

```
# positive value exponents
ar_exponents_posv <- c(0.05, 0.5, 1, 1.5)
# positive and negative values of the base
ar_baseval_pos <- seq(1e-10, 1.5, length.out = 1000)
# base to power
```

yvalidx	y_vals	base1.01	base1.02	base1.03	base1.04	base1.1	base2	base2.71	base10
1	-38.5153709	0.3183512	0.5335972	0.6796907	0.7792226	0.9745466	1.000000e+00	1.000000e+00	1.000000e+00
2	-31.5508872	0.2694384	0.4646264	0.6064733	0.7098764	0.9505662	1.000000e+00	1.000000e+00	1.000000e+00
3	-20.3611320	0.1833952	0.3318242	0.4522031	0.5500317	0.8563856	9.999993e-01	1.000000e+00	1.000000e+00
4	-17.6413216	0.1609936	0.2948498	0.4063446	0.4993786	0.8138863	9.999951e-01	1.000000e+00	1.000000e+00
5	-15.3604745	0.1417345	0.2622703	0.3649409	0.4525306	0.7686930	9.999762e-01	9.999998e-01	1.000000e+00
6	-7.0736023	0.0679648	0.1307078	0.1886755	0.2422727	0.4904291	9.925761e-01	9.991343e-01	9.999999e-01
7	-6.8987381	0.0663417	0.1276924	0.1844711	0.2370581	0.4818653	9.916194e-01	9.989695e-01	9.999999e-01
8	-0.8772047	0.0086905	0.0172210	0.0255958	0.0338195	0.0802069	4.555787e-01	5.829412e-01	8.673231e-01
9	6.0851203	-0.0624196	-0.1280623	-0.1970604	-0.2695503	-0.7859918	-6.688968e+01	-4.301915e+02	-1.216522e+06
10	6.4217741	-0.0659845	-0.1356078	-0.2090319	-0.2864244	-0.8442274	-8.473272e+01	-6.021578e+02	-2.641033e+06
11	9.8179763	-0.1026232	-0.2146084	-0.3367050	-0.4697143	-1.5491325	-9.016209e+02	-1.781796e+04	-6.576220e+09
12	12.7710549	-0.1355036	-0.2877551	-0.4586292	-0.6501891	-2.3777558	-6.988894e+03	-3.384335e+05	-5.902757e+12
13	14.4496274	-0.1546284	-0.3312796	-0.5328269	-0.7624849	-2.9637739	-2.237442e+04	-1.804034e+06	-2.815966e+14
14	16.0241060	-0.1728599	-0.3734412	-0.6058503	-0.8747529	-3.6055423	-6.663924e+04	-8.668536e+06	-1.057075e+16
15	20.1352792	-0.2218336	-0.4899334	-0.8133478	-1.2027796	-5.8148024	-1.151656e+06	-5.223451e+08	-1.365461e+20
16	21.0799447	-0.2333727	-0.5180677	-0.8646958	-1.2859243	-6.4568518	-2.216641e+06	-1.339580e+09	-1.202111e+21
17	30.4425692	-0.3537975	-0.8273062	-1.4592241	-2.3001875	-17.2011858	-1.459239e+09	-1.515994e+13	-2.770571e+30
18	35.7178433	-0.4267575	-1.0285214	-1.8742066	-3.0587672	-29.0924427	-5.651220e+10	-2.915609e+15	-5.222077e+35
19	36.4115292	-0.4366396	-1.0565790	-1.9337492	-3.1707094	-31.1492588	-9.140343e+10	-5.822024e+15	-2.579463e+36
20	49.9387703	-0.6436301	-1.6883264	-3.3759789	-6.0896373	-115.7077766	-1.079115e+15	-4.187492e+21	-8.685009e+49

```
mt_x2a_val <- matrix(data = NA, nrow = length(ar_exponents_posv), ncol = length(ar_baseval_pos))
# Generate values
it_row_ctr <- 0
for (fl_exponents_posv in ar_exponents_posv) {
  it_row_ctr <- it_row_ctr + 1
  mt_x2a_val[it_row_ctr, ] <- ar_baseval_pos^fl_exponents_posv
}
```

Note that the smaller exponents functions are higher when $x < 1$, but lower when $x > 1$.

if $b > a > 0$, then, $(x^a - x^b) > 0$, for all $1 > x > 0$

if $b > a > 0$, then, $(x^a - x^b) < 0$, for all $x > 1$

Note we also have: $\lim_{a \rightarrow 0} x^a = 1$ and $\lim_{a \rightarrow 1} x^a = x$ bounds. When $a > 1$, function becomes convex.

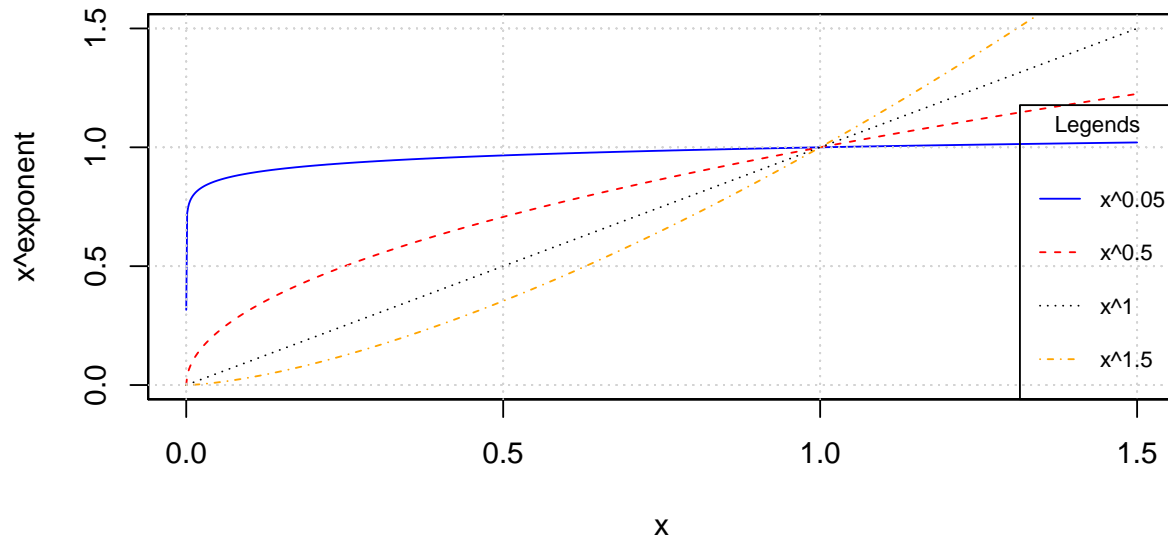
```
# x and bounds
ar_xlim <- c(min(ar_baseval_pos), max(ar_baseval_pos))
ar_ylim <- c(0, 1.5)
# function line
st_line_1_y_legend <- paste0("x^", ar_exponents_posv[1])
st_line_2_y_legend <- paste0("x^", ar_exponents_posv[2])
st_line_3_y_legend <- paste0("x^", ar_exponents_posv[3])
st_line_4_y_legend <- paste0("x^", ar_exponents_posv[4])
# Color and line
st_point_1_pch <- 10
st_point_1_cex <- 2
ar_colors <- c("blue", "red", "black", "orange")
ar_ltys <- c("solid", "dashed", "dotted", "dotdash")
# Graph and combine
for (it_graph in c(1, 2, 3, 4)) {
  if (it_graph != 1) {
    par(new = T)
  }
  ar_y_current <- mt_x2a_val[it_graph, ]
  plot(ar_baseval_pos, ar_y_current,
       type = "l",
       col = ar_colors[it_graph], lty = ar_ltys[it_graph],
```

```

    pch = 10, cex = 2, xlim = ar_xlim, ylim = ar_ylim, panel.first = grid(),
    ylab = "", xlab = "", yaxt = "n", xaxt = "n", ann = FALSE
  )
  plot_line <- recordPlot()
}
# CEX sizing Contorl Titling and Legend Sizes
fl_ces_fig_reg <- 1
fl_ces_fig_small <- 0.75
# R Legend
st_title <- paste0("Positive Exponential Graphing")
st_subtitle <- paste0(
  "https://fanwangecon.github.io/",
  "R4Econ/math/solutions/htmlpdf/fs_inequality.html"
)
st_x_label <- "x"
st_y_label <- "x^exponent"
title(
  main = st_title, sub = st_subtitle, xlab = st_x_label, ylab = st_y_label,
  cex.lab = fl_ces_fig_reg,
  cex.main = fl_ces_fig_reg,
  cex.sub = fl_ces_fig_small
)
axis(1, cex.axis = fl_ces_fig_reg)
axis(2, cex.axis = fl_ces_fig_reg)
grid()
# Legend sizing CEX
legend("bottomright",
  inset = c(0, 0),
  xpd = TRUE,
  c(st_line_1_y_legend, st_line_2_y_legend, st_line_3_y_legend, st_line_4_y_legend),
  col = c(ar_colors[1], ar_colors[2], ar_colors[3], ar_colors[4]),
  cex = fl_ces_fig_small,
  lty = c(ar_ltys[1], ar_ltys[2], ar_ltys[3], ar_ltys[4]),
  title = "Legends",
  y.intersp = 2
)

```

Positive Exponential Graphing



https://fanwangecon.github.io/R4Econ/math/solutions/htmlpdf/fs_inequality.html

1.4 Negative Exponents

Similar to above, but now with negative exponents.

```
# positive value exponents
ar_exponents_posv <- -c(0.05, 0.5, 1, 1.5)
# positive and negative values of the base
ar_baseval_pos <- seq(1e-10, 1.5, length.out = 1000)
# base to power
mt_x2a_val <- matrix(data = NA, nrow = length(ar_exponents_posv), ncol = length(ar_baseval_pos))
# Generate values
it_row_ctr <- 0
for (fl_exponents_posv in ar_exponents_posv) {
  it_row_ctr <- it_row_ctr + 1
  mt_x2a_val[it_row_ctr, ] <- ar_baseval_pos^fl_exponents_posv
}
```

For positive exponents, when $x < 1$, $x^a < 1$, when $x > 1$, $x^a > 1$. For negative exponents, when $x < 1$, $x^a > 1$, and when $x > 1$, $x^a < 1$. Large positive exponents generate small values when $x < 1$, and large negative exponents generate very large values when $x < 1$.

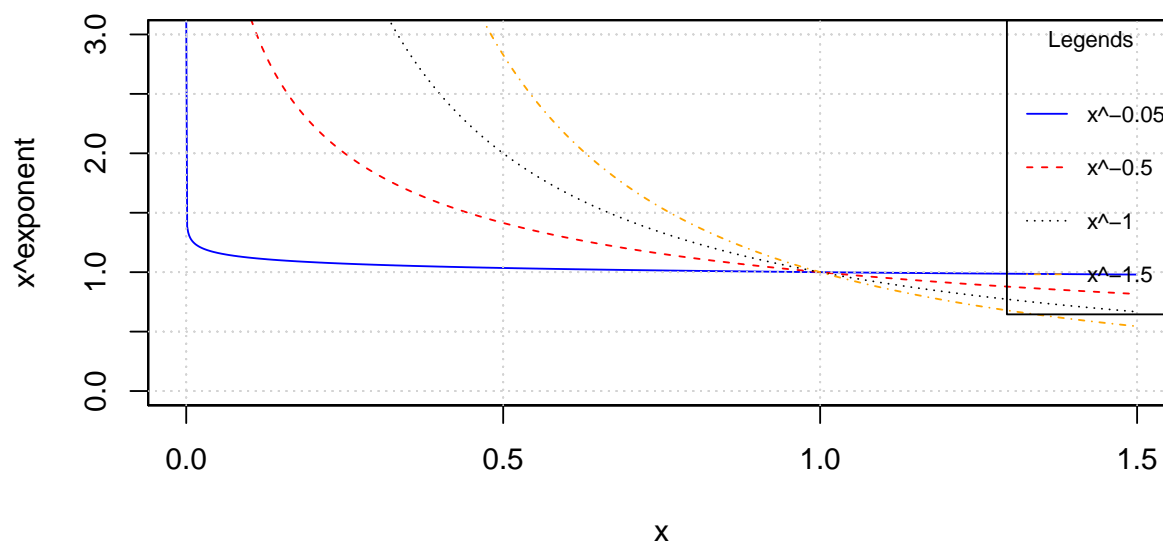
```
# x and bounds
ar_xlim <- c(min(ar_baseval_pos), max(ar_baseval_pos))
ar_ylim <- c(0, 3)
# function line
st_line_1_y_legend <- paste0("x^", ar_exponents_posv[1])
st_line_2_y_legend <- paste0("x^", ar_exponents_posv[2])
st_line_3_y_legend <- paste0("x^", ar_exponents_posv[3])
st_line_4_y_legend <- paste0("x^", ar_exponents_posv[4])
# Color and line
st_point_1_pch <- 10
st_point_1_cex <- 2
```

```

ar_colors <- c("blue", "red", "black", "orange")
ar_ltys <- c("solid", "dashed", "dotted", "dotdash")
# Graph and combine
for (it_graph in c(1, 2, 3, 4)) {
  if (it_graph != 1) {
    par(new = T)
  }
  ar_y_current <- mt_x2a_val[it_graph, ]
  plot(ar_baseval_pos, ar_y_current,
       type = "l",
       col = ar_colors[it_graph], lty = ar_ltys[it_graph],
       pch = 10, cex = 2, xlim = ar_xlim, ylim = ar_ylim, panel.first = grid(),
       ylab = "", xlab = "", yaxt = "n", xaxt = "n", ann = FALSE
  )
  plot_line <- recordPlot()
}
# CEX sizing Contorl Titling and Legend Sizes
fl_ces_fig_reg <- 1
fl_ces_fig_small <- 0.75
# R Legend
st_title <- paste0("Negative Exponential Graphing")
st_subtitle <- paste0(
  "https://fanwangecon.github.io/",
  "R4Econ/math/solutions/htmlpdf/fs_inequality.html"
)
st_x_label <- "x"
st_y_label <- "x^exponent"
title(
  main = st_title, sub = st_subtitle, xlab = st_x_label, ylab = st_y_label,
  cex.lab = fl_ces_fig_reg,
  cex.main = fl_ces_fig_reg,
  cex.sub = fl_ces_fig_small
)
axis(1, cex.axis = fl_ces_fig_reg)
axis(2, cex.axis = fl_ces_fig_reg)
grid()
# Legend sizing CEX
legend("topright",
      inset = c(0, 0),
      xpd = TRUE,
      c(st_line_1_y_legend, st_line_2_y_legend, st_line_3_y_legend, st_line_4_y_legend),
      col = c(ar_colors[1], ar_colors[2], ar_colors[3], ar_colors[4]),
      cex = fl_ces_fig_small,
      lty = c(ar_ltys[1], ar_ltys[2], ar_ltys[3], ar_ltys[4]),
      title = "Legends",
      y.intersp = 2
)

```

Negative Exponential Graphing



https://fanwangecon.github.io/R4Econ/math/solutions/htmlpdf/fs_inequality.html

1.5 Inequality and Exponents

Suppose we have the inequality $0 < a < b$, if we apply positive exponents to them, the direction of the inequality will stay the same: If $0 < a < b$, then $0 < a^{|\alpha|} < b^{|\alpha|}$ if $\alpha < 0$. Think about the graphs above, think of a and b as points along the x-axis, note that positive exponents are strictly increasing (although some concavely and some convexly) along the x-axis. Comparing x^α at $0 < b < a$ anywhere along the x-axis has still has $b^\alpha < a^\alpha$.

In contrast, if $0 < a < b$, then $a^{-|\alpha|} > b^{-|\alpha|} > 0$ if $\alpha < 0$. Sign flips. Visually from above, the sign-flipping happens because negative exponential is strictly decreasing along $x > 0$.