Package 'learningCurve'

March 30, 2018

Type Package

Title An Implementation of Crawford's and Wright's Learning Curve Production Functions

Version 1.1.2

Description Implements common learning curve production functions. It incorporates Crawford's and Wright's learning curve functions to compute unit and cumulative block estimates for time (or cost) of units along with an aggregate learning curve. It also provides delta and error functions and some basic learning curve plotting functions.along with functions to compute aggregated learning curves, error rates, and to visualize learning curves.

Date 2018-03-30

URL https://github.com/AFIT-R/learningCurve

BugReports https://github.com/AFIT-R/learningCurve/issues

License GPL (>= 2) | file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports ggplot2, scales, gridExtra, tibble

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Bradley Boehmke [aut, cre], Jason Freels [aut]

Maintainer Bradley Boehmke <bradleyboehmke@gmail.com>

Repository CRAN

Date/Publication 2018-03-30 17:35:52 UTC

R topics documented:

agg_curve . . . 2
ca_block . . . 3
ca_unit . . . 4
cum_error . . . 4
delta . . . 5
lc_rate . . . 6
lc_rate_est . . . 7
natural_slope . . . 7
natural_slope_est . . . 8
plot_block_summary . . . 8
plot_delta . . . 9
plot_unit_curve . . . 10
unit_block_summary . . . 11
unit_cum_appx . . . 12
unit_cum_exact . . . 13
unit_curve . . . 13
unit_midpoint . . . 14

Index **16**

`agg_curve` *Aggregate Learning Curve*

Description

Computes the approximate aggregate cumulative learning curve formula by calculating the sum of all contributing hours from all departments for all production units 1 through n.

Usage

```
agg_curve(t, r, n, na.rm = FALSE)
```

Arguments

`t`	vector of hours (or costs) for the first unit from departments 1 through m
`r`	vector of historical learning rates for departments 1 through m
`n`	total units to be produced across all departments
`na.rm`	Should `NA` values be removed?

Examples

```
## Not run:
# A project is expected to get underway soon to produce 300
# widgets. Three departments will be involved. Historically,
# these departments have had learning curves of 85%, 87%, and
# 80% respectively. The first unit hours for these departments
# have been estimated at 70, 45, and 25 respectively. What is
# the total predicted hours required for the entire effort?

t <- c(70, 45, 25)
r <- c(.85, .87, .8)

agg_curve(t = t, r = r, n = 300)
## [1] 11000.96

## End(Not run)
```

`ca_block` *Wright's Cumulative Average Learning Curve Function*

Description

Computes cumulative time or cost for units m through n in a production block using Wright's cumulative average model. Assumes the block begins at unit m and ends at unit n.

Usage

```
ca_block(t, n, r, m = 1, na.rm = FALSE)
```

Arguments

`t`	time (or cost) required for the mth unit of production
`n`	last unit of the production block of concern
`r`	learning curve rate
`m`	first unit of the production block of concern (default: m = 1)
`na.rm`	Should `NA` values be removed?

Examples

```
# Production of the first 200 units of a product is nearing its
# end. Your customer said he is willing to buy an additional 50
# units. There will be no break in production or in learning. The
# first unit required 75 hours and the first 200 units had an 85%
# learning curve. How many hours will the second block of 50 units
# require?
```

```
ca_block(t = 75, m = 201, n = 250, r = .85)
## [1] 806.772
```

`ca_unit` *Wright's Cumulative Average Unit Learning Curve Function*

Description

Computes the time (or cost) required for a specific unit using Wright's cumulative average model.

Usage

```
ca_unit(t, n, r, m = 1, na.rm = FALSE)
```

Arguments

`t`	time (or cost) required for the mth unit of production
`n`	nth unit you wish to predict the time (or cost) for
`r`	learning curve rate
`m`	mth unit for which you have time (or cost) information (default is m = 1)
`na.rm`	Should `NA` values be removed?

Examples

```
# An estimator want to know the unit hours for unit 2,200 using
# when the hours for unit 1 were 110 and the learning rate was
# 88.5%.

ca_unit(t = 110, m = 1, n = 2200, r = .885)
## [1] 23.34001
```

`cum_error` *Approximate Prediction Error*

Description

Computes approximate percent error in cumulative time (or cost) due to an incorrect choice of learning curve rate. The output provides the measure of error when learning curve r1 is erroneously chosen when r2 should have been chosen. It is the ratio of the actual cumulative results based on the realized learning curve to the predicted cumulative results based on the erroneously used learning rate.

Usage

```
cum_error(n, r1, r2)
```

Arguments

n	cummulative units in the production quantity
r1	original learning curve rate (aka erroneously used learning curve rate)
r2	learning curve rate to compare to r1 (aka realized learning curve rate)

Examples

```
# An estimator is predicting hours for a block of 250 units. Historically,
# the organization has had a learning rate between 85-87%. What is the
# potential error in the prediction by using one of these two learning
# rates (85% vs. 87%)? If you go with a learning rate of 85% and the
# organization performs at a learning rate of 87% then the error would
# be 20%.

cum_error(n = 250, r1 = .85, r2 = .87)
## [1] 0.2035303
```

---

delta *Crawford vs. Wright Unit Difference*

Description

Computes the difference between the unit or cumulative prediction estimates provided by the Crawford and Wright models.

Usage

```
delta(t, m, n, r, level = "u")
```

Arguments

t	time (or cost) required to produce the first unit
m	mth unit for which you have time (or cost) information (default is m = 1)
n	the nth unit you wish to predict the time (or cost) for when comparing unit predictions or the last unit in the block when comparing cumulative time (or costs)
r	learning curve rate
level	calculate unit ("u") versus cumulative ("c") differences (default = "u")

Examples

```
# The first unit of production is expected to require 50 hours and
# the learning rate is expected to be 88.5%. However, the estimator
# is not sure whether the learning rate is based on the unit model
# or cumulative average model and wants to understand the difference
# between potential outcomes for each unit.

# differences between per unit time requirements
delta(t = 50, m = 1, n = 25, r = .885)
## [1] 0.000000 5.750000 6.103821 6.110519 6.041146 5.953271 5.863560 5.777401 5.696436
## [10] 5.620942 5.550687 5.485263 5.424223 5.367136 5.313606 5.263280 5.215844 5.171025
## [19] 5.128579 5.088293 5.049980 5.013473 4.978624 4.945304 4.913395

# differences between cumulative unit time requirements
delta(t = 50, m = 1, n = 25, r = .885, level = "c")
## [1]   0.00000   5.75000  11.85382  17.96434  24.00549  29.95876  35.82232  41.59972
## [9]  47.29615  52.91710  58.46778  63.95305  69.37727  74.74440  80.05801  85.32129
## [17]  90.53713  95.70816 100.83674 105.92503 110.97501 115.98848 120.96711 125.91241
## [25] 130.82581
```

---

lc_rate *Learning Rate Converter*

Description

Computes the learning rate for given natural slopes

Usage

```
lc_rate(b, na.rm = FALSE)
```

Arguments

b	natural slope
na.rm	Should NA values be removed?

Examples

```
# Calculate the learning rates for natural slopes -.19, -.22, -.25
lc_rate(b = c(-.19, -.22, -.25))
## [1] 0.8766057 0.8585654 0.8408964
```

lc_rate_est *Learning Rate Estimate*

Description

Computes the learning rate based on total time (cost) to produce the first n units, time (cost) required for the first unit and total units produced.

Usage

```
lc_rate_est(T, t, n)
```

Arguments

T	total time (or cost) required to produce the first n units
t	time (or cost) required to produce the first unit
n	total n units produced

Examples

```
# Estimate the learning curve rate for 250 units when the time
# for unit one took 80 hours and the total time for all 250
# units took 8,250 hours.

lc_rate_est(T = 8250, t = 80, n = 250)
## [1] 0.8947908
```

natural_slope *Natural Slope Rate Converter*

Description

Computes the natural slope rate for given learning rates

Usage

```
natural_slope(r, na.rm = FALSE)
```

Arguments

r	learning curve rate
na.rm	Should NA values be removed?

Examples

```
# Calculate the natural slope for learning rates of 80%, 85%, 90%

natural_slope(r = c(.80, .85, .90))
## [1] -0.3219281 -0.2344653 -0.1520031
```

`natural_slope_est` *Natural Slope Estimate*

Description

Computes the natural slope based on total time (cost) to produce the first n units, time (cost) required for the first unit and total units produced.

Usage

```
natural_slope_est(T, t, n)
```

Arguments

`T`	total time (or cost) required to produce the first n units
`t`	time (or cost) required to produce the first unit
`n`	total n units produced

Examples

```
# Estimate the natural slope for 250 units when the time for unit
# one took 80 hours and the total time for all 250 units took
# 8,250 hours.

natural_slope_est(T = 8250, t = 80, n = 250)
## [1] -0.1603777
```

`plot_block_summary` *Block Summary Plot*

Description

Plots the Crawford unit learning curve for the production block containing units m through n (inclusive) while highlighting midpoint values.

Usage

```
plot_block_summary(t, m, n, r)
```

Arguments

t	time (or cost) required for the mth unit of production
m	mth unit for which you have time (or cost) information (default is m = 1)
n	nth (last) unit of production in the production block of concern (n > m)
r	learning curve rate

Examples

```
# A production block runs from unit 201 to unit 500 inclusive.
# The 201st unit had a required time of 125 hours with a 75%
# learning curve. Plot the block summary?

plot_block_summary(t = 125, m = 201, n = 500, r = .75)
```

---

plot_delta *Crawford vs. Wright Delta Plot*

Description

Plots the delta of hours (or cost) per unit between Crawford's unit model and Wright's cumulative average model.

Usage

```
plot_delta(t, m, n, r, level = "u")
```

Arguments

t	time (or cost) required to produce the mth unit
m	mth unit for which you have time (or cost) information (default is m = 1)
n	the nth unit you wish to predict the time (or cost) for when comparing unit predictions or the last unit in the block when comparing cumulative time (or costs)
r	learning curve rate
level	plot the delta between the Crawford and Wright models at the unit ("u") or cumulative ("c") level

Examples

```
# The first unit of production is expected to require 50 hours and
# the learning rate is expected to be 88.5%. However, the estimator
# is not sure whether the learning rate is based on the unit model
# or cumulative average model and wants to understand the difference
# between potential outcomes for each unit.

# Plot the differences between per unit time requirements
plot_delta(t = 50, m = 1, n = 25, r = .885)

# Plot the differences between cumulative time requirements
plot_delta(t = 50, m = 1, n = 25, r = .885, level = "c")
```

---

plot_unit_curve *Learning Curve Plot*

Description

Plots the learning curve for units m through n. Allows you to choose between the Crawford and Wright models and also between a unit level plot or a cumulative level plot.

Usage

```
plot_unit_curve(t, m, n, r, model = "u", level = "u")
```

Arguments

t	time (or cost) required for the mth unit of production
m	mth unit for which you have time (or cost) information (default is m = 1)
n	nth unit of production you wish to plot the learning curve through (n > m)
r	learning curve rate
model	choose between the Crawford ("u") or Wright ("ca") models or plot both models with "both"
level	plot the learning curve at the unit ("u") or cumulative ("c") level

Examples

```
# library(learningCurve)
# An estimator wants to plot the learning curve for for units
# one through 125 where the first unit requires 100 hours and
# the learning rate is 85%.

# plot the time (or cost) per unit based on Crawford's Unit
# Learning Curve Function
plot_unit_curve(t = 100, m = 1, n = 125, r = .85)
```

```
# plot the cumulative time (or cost) per unit based on Crawford's
# Unit Learning Curve Function
plot_unit_curve(t = 100, m = 1, n = 125, r = .85, level = "c")

# plot the time (or cost) per unit based on Wright's Cumulative
# Average Learning Curve Function
plot_unit_curve(t = 100, m = 1, n = 125, r = .85, model = "ca")

# plot the cumulative time (or cost) per unit based on Wrights's
# Cumulative Average Learning Curve Function
plot_unit_curve(t = 100, m = 1, n = 125, r = .85, model = "ca", level = "c")
```

`unit_block_summary` *Block Summary Function*

Description

Provides summary information for the block containing units m through n (where n > m). Based on Crawford's unit learning curve model.

Usage

```
unit_block_summary(t, m, n, r, na.rm = FALSE)
```

Arguments

`t`	time for the mth unit
`m`	lower bound unit of production block
`n`	upper bound unit of production block
`r`	learning curve rate
`na.rm`	Should `NA` values be removed?

Examples

```
# A production block runs from unit 201 to unit 500 inclusive.
# The 201st unit had a required time of 125 hours with a 75%
# learning curve, what is the block summary?

unit_block_summary(t = 125, m = 201, n = 500, r = .75)
## $`block units`
## [1] 300

## $`block hours`
## [1] 30350.48

## $`midpoint unit`
## [1] 334.6103
```

```
## $`midpoint hours`
## [1] 101.1683
```

`unit_cum_appx` *Approximate Cumulative Unit Learning Curve Function*

Description

Provides the approximate cumulative time or cost required for units m through n (inclusive) using the Crawford unit model. Provides nearly the exact output as unit_cum_exact(), usually only off by 1-2 units but reduces computational time drastically if trying to calculate cumulative hours (costs) for over a million units.

Usage

```
unit_cum_appx(t, n, r, m = 1, na.rm = FALSE)
```

Arguments

`t`	time (or cost) required for the mth unit of production
`n`	The unit you wish to predict the cumulative time (or cost) to
`r`	learning curve rate
`m`	mth unit of production (default set to 1st production unit)
`na.rm`	Should `NA` values be removed?

Examples

```
library(learningCurve)
# An estimator believes that the first unit of a product will
# require 100 labor hours. How many total hours will be required
# for 125 units given the organization has historically experienced
# an 85% learning curve?

unit_cum_appx(t = 100, n = 125, r = .85)
## [1] 5202.998

# Computational difference between unit_cum_exact() and unit_cum_appx()
# for 1 million units

system.time(unit_cum_exact(t = 100, n = 1000000, r = .85))
##  user  system elapsed
## 0.105   0.004   0.109

system.time(unit_cum_appx(t = 100, n = 1000000, r = .85))
## user  system elapsed
##  0       0       0
```

unit_cum_exact *Exact Cumulative Unit Learning Curve Function*

Description

Provides the exact cumulative time or cost required for units m through n (inclusive) using the Crawford unit model

Usage

```
unit_cum_exact(t, n, r, m = 1, na.rm = FALSE)
```

Arguments

t	time (or cost) required for the mth unit of production
n	The unit you wish to predict the cumulative time (or cost) to
r	learning curve rate
m	mth unit of production (default set to 1st production unit)
na.rm	Should NA values be removed?

Examples

```
library(learningCurve)
# An estimator believes that the first unit of a product will
# require 100 labor hours. How many total hours will be required
# for 125 units given the organization has historically experienced
# an 85% learning curve?

unit_cum_exact(t = 100, n = 125, r = .85)
## [1] 5201.085
```

unit_curve *Crawford's Unit Learning Curve Function*

Description

Predicts the time or cost of the nth unit given the time of the mth unit and the learning rate

Usage

```
unit_curve(t, n, r, m = 1, na.rm = FALSE)
```

Arguments

t	time (or cost) required for the mth unit of production
n	nth unit you wish to predict the time (or cost) for
r	learning curve rate
m	mth unit of production (default set to 1st production unit)
na.rm	Should NA values be removed?

Examples

```
library(learningCurve)
# An estimator believes that the first unit of a product will
# require 100 labor hours. How many hours will the 125th unit
# require given the organization has historically experienced
# an 85% learning curve?

unit_curve(t = 100, m = 1, n = 125, r = .85)
## [1] 32.23647

# If the estimator wants to assess the hours required for the
# 125 unit given multiple learning curve rates

r <- c(.8, .85, .9, .95)
unit_curve(t = 100, m = 1, n = 125, r = r)
## [1] 21.13225 32.23647 48.00243 69.95640

# If the estimator has the time required for the 100th unit
unit_curve(t = 100, m = 100, n = 125, r = .85)
## [1] 94.90257
```

---

unit_midpoint *Midpoint Unit Function*

Description

Provides the so-called "midpoint" or average unit between units m and n (where n > m). Based on Crawford's unit learning curve model.

Usage

```
unit_midpoint(m, n, r, na.rm = FALSE)
```

Arguments

m	lower bound unit of production
n	upper bound unit of production
r	learning curve rate
na.rm	Should NA values be removed?

Examples

```
# If a production block runs from unit 201 to unit 500 inclusive
# with a 75% learning curve, what is the midpoint unit?

unit_midpoint(m = 201, n = 500, r = .75)
## [1] 334.6103
```

Index

agg_curve, 2

ca_block, 3
ca_unit, 4
cum_error, 4

delta, 5

lc_rate, 6
lc_rate_est, 7

natural_slope, 7
natural_slope_est, 8

plot_block_summary, 8
plot_delta, 9
plot_unit_curve, 10

unit_block_summary, 11
unit_cum_appx, 12
unit_cum_exact, 13
unit_curve, 13
unit_midpoint, 14