

# Package ‘EVchargcost’

January 20, 2025

**Type** Package

**Title** Computes and Plot the Optimal Charging Strategy for Electric Vehicles

**Version** 0.1.0

**Description** The purpose of this library is to compute the optimal charging cost function for a electric vehicle (EV). It is well known that the charging function of a EV is a concave function that can be approximated by a piece-wise linear function, so bigger the state of charge, slower the charging process is. Moreover, the other important function is the one that gives the electricity price. This function is usually step-wise, since depending on the time of the day, the price of the electricity is different. Then, the problem of charging an EV to a certain state of charge is not trivial. This library implements an algorithm to compute the optimal charging cost function, that is, it plots for a given state of charge  $r$  (between 0 and 1) the minimum cost we need to pay in order to charge the EV to that state of charge  $r$ . The details of the algorithm are described in González-Rodríguez et al (2023) <<https://inria.hal.science/hal-04362876v1>>.

**License** GPL-3

**Language** en-US

**Imports** ggplot2, cowplot

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Brais Gonzalez-Rodriguez [aut, cre]

**Maintainer** Brais Gonzalez-Rodriguez <[braisgonzalez.rodriguez@usc.es](mailto:braisgonzalez.rodriguez@usc.es)>

**Repository** CRAN

**Date/Publication** 2024-01-10 13:43:07 UTC

## Contents

minimum_cost . . . . .	2
plot_functions . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

`minimum_cost`*Computes the minimum cost function*

---

**Description**

Function that computes the minimal cost function for a given charging function (given by `a`, `alpha` and `beta`), an electricity price function (given by `delta` and `gamma`), a consumption `tau` and a range `R`

**Usage**

```
minimum_cost(a, alpha, beta, delta, gamma, tau, R)
```

**Arguments**

<code>a</code>	vector with the breaking points of charging function in the x-axis
<code>alpha</code>	vector with the slopes of the charging function on each segment
<code>beta</code>	vector with the y-intercepts of the charging function on each segment
<code>delta</code>	vector with the times duration of each segment of electricity price function
<code>gamma</code>	vector with the prices of the electricity on each segment of electricity price function
<code>tau</code>	consumption of the vehicle (numerical value)
<code>R</code>	range of the vehicle (numerical value)

**Value**

list with the x-values and y-values of the minimum cost function

**Examples**

```
a <- c(0,3.3,6.6,10)
alpha <- c(0.1757576, 0.07272727, 0.05294118)
beta <- c(0, 0.34, 0.4705882)
delta <- c(4, 3, 5)
gamma <- c(0.45, 0.25, 0.5)
tau <- 0.15
R <- 250
opt_cost_function = minimum_cost(a, alpha, beta, delta, gamma, tau, R)
print(opt_cost_function)
```

---

plot_functions	<i>Plots the charging function, the electricity price function and the optimal cost function</i>
----------------	--

---

**Description**

Function that plots the charging function, the electricity price function and the optimal cost function

**Usage**

```
plot_functions(a, alpha, beta, delta, gamma, tau, R, x_values, y_values)
```

**Arguments**

a	vector with the breaking points of charging function in the x-axis
alpha	vector with the slopes of the charging function on each segment
beta	vector with the y-intercepts of the charging function on each segment
delta	vector with the times duration of each segment of electricity price function
gamma	vector with the prices of the electricity on each segment of electricity price function
tau	consumption of the vehicle (numerical value)
R	range of the vehicle (numerical value)
x_values	vector with the x-values of the breaking points of the charging cost function
y_values	vector with the y-values of the breaking points of the charging cost function

**Value**

A plot with the charging function, the electricity price function and the optimal cost function

**Examples**

```
a <- c(0,3.3,6.6,10)
alpha <- c(0.1757576, 0.07272727, 0.05294118)
beta <- c(0, 0.34, 0.4705882)
delta <- c(4, 3, 5)
gamma <- c(0.45, 0.25, 0.5)
tau <- 0.15
R <- 250
opt_cost_function = minimum_cost(a, alpha, beta, delta, gamma, tau, R)
xvalues <- opt_cost_function[["xvalues"]]
yvalues <- opt_cost_function[["yvalues"]]
plot_functions(a, alpha, beta, delta, gamma, tau, R, xvalues, yvalues)
```

# Index

`minimum_cost`, [2](#)

`plot_functions`, [3](#)