

# Package ‘PH1XBAR’

June 19, 2025

**Type** Package

**Title** Phase I Shewhart X-Bar Chart

**Version** 0.11.3

**Maintainer** Yuhui Yao <yyao17@ua.edu>

**Description** The purpose of 'PH1XBAR' is to build a Phase I Shewhart control chart for the basic Shewhart, the variance components and the ARMA models in R for subgrouped and individual data. More details can be found: Yao and Chakraborti (2020) <[doi:10.1002/qre.2793](https://doi.org/10.1002/qre.2793)>, Yao and Chakraborti (2021) <[doi:10.1080/08982112.2021.1911111](https://doi.org/10.1080/08982112.2021.1911111)>

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5.0)

**Imports** forecast, mvtnorm, pracma, VGAM

**URL** <https://github.com/bolus123/PH1XBAR>

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Yuhui Yao [aut, cre],  
Subha Chakraborti [ctb],  
Tyler Thomas [ctb],  
Jason Parton [ctb],  
Xin Yang [ctb]

**Repository** CRAN

**Date/Publication** 2025-06-19 16:30:02 UTC

## Contents

PH1XBAR-package . . . . .	2
bore_diameter_data . . . . .	3
getCC.ARMA . . . . .	4
getCC.XBAR . . . . .	5

grinder_data . . . . .	6
PH1ARMA . . . . .	7
PH1XBAR . . . . .	10
pistonring_data . . . . .	12
preston_data . . . . .	12
semiconductor_data . . . . .	13
snowfall_data . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

PH1XBAR-package	<i>PH1XBAR: Phase I Shewhart X-Bar Chart</i>
-----------------	--

---

## Description

The purpose of 'PH1XBAR' is to build a Phase I Shewhart control chart for the basic Shewhart, the variance components and the ARMA models in R for subgrouped and individual data. More details can be found: Yao and Chakraborti (2020) doi: [10.1002/qre.2793](https://doi.org/10.1002/qre.2793), Yao and Chakraborti (2021) doi: [10.1080/08982112.2021.1878220](https://doi.org/10.1080/08982112.2021.1878220), and Yao et al. (2023) doi: [10.1080/00224065.2022.2139783](https://doi.org/10.1080/00224065.2022.2139783).

The utility of this package is in building a Shewhart-type control chart based on new methods for subgrouped and individual data. The Phase I chart is based on the multivariate normal/t or ARMA process.

## Author(s)

**Maintainer:** Yuhui Yao <yyao17@ua.edu>

Other contributors:

- Subha Chakraborti <schakrab@ua.edu> [contributor]
- Tyler Thomas <tjthomas7@crimson.ua.edu> [contributor]
- Jason Parton <jmparton@ua.edu> [contributor]
- Xin Yang <xyang15@cba.ua.edu> [contributor]

## References

- Champ, C.W., and Jones, L.A. (2004) Designing Phase I X-bar charts with small sample sizes. *Quality and Reliability Engineering International*. 20(5), 497-510
- Yao, Y., Hilton, C.W., and Chakraborti, S. (2017) Designing Phase I Shewhart X-bar charts: Extended tables and software. *Quality and Reliability Engineering International*. 33(8), 2667-2672.
- Yao, Y., and Chakraborti, S. (2021). Phase I monitoring of individual normal data: Design and implementation. *Quality Engineering*, 33(3), 443-456.
- Yao, Y., and Chakraborti, S. (2021). Phase I process monitoring: The case of the balanced one-way random effects model. *Quality and Reliability Engineering International*, 37(3), 1244-1265.
- Yao, Y., Chakraborti, S., Yang, X., Parton, J., Lewis Jr, D., and Hudnall, M. (2023). Phase I control chart for individual autocorrelated data: application to prescription opioid monitoring. *Journal of Quality Technology*, 55(3), 302-317.

## See Also

Useful links:

- <https://github.com/bolus123/PH1XBAR>

## Examples

```
#Build a Phase I basic Shewhart control chart
data(grinder_data)
PH1XBAR(grinder_data, nsim=10)

# Build a Phase I individual control chart with an ARMA model
data(pleston_data)
PH1ARMA(pleston_data, nsim.process=10, nsim.coefs=10)
```

---

bore_diameter_data	<i>Bore diameter in manufacturing automotive driver gears</i>
--------------------	---

---

## Description

A dataset containing bore diameter measurements in mm

## Usage

```
bore_diameter_data
```

## Format

A matrix with 20 rows and 5 variables:

- X1** Diameter measurement at Position 1
- X2** Diameter measurement at Position 2
- X3** Diameter measurement at Position 3
- X4** Diameter measurement at Position 4
- X5** Diameter measurement at Position 5

## References

Wooluru, Yerriswamy, D. R. Swamy, and P. Nagesh. "THE PROCESS CAPABILITY ANALYSIS-A TOOL FOR PROCESS PERFORMANCE MEASURES AND METRICS-A CASE STUDY." International Journal for Quality Research 8.3 (2014).

getCC.ARMA

*get Phase I corrected charting constant with an ARMA model***Description**

get Phase I corrected charting constant with an ARMA model

**Usage**

```
getCC.ARMA(
  fap0 = 0.05,
  interval = c(1, 4),
  m = 50,
  order = c(1, 0),
  phi.vec = 0.5,
  theta.vec = NULL,
  case = "U",
  method = "MLE+MOM",
  nsim.coefs = 100,
  nsim.process = 1000,
  burn.in = 50,
  sim.type = "Recursive",
  verbose = FALSE
)
```

**Arguments**

fap0	nominal false Alarm Probabilty in Phase 1
interval	searching range of charting constants for the exact method
m	number of observations
order	order for ARMA(p, q) model
phi.vec	a vector of length p containing autoregressive coefficient(s). When case = 'K', the vector must have a length equal to the first value in the order. If no autoregressive coefficient presents, set phi.vec = NULL
theta.vec	a vector of length q containing moving-average coefficient(s). When case = 'K', the vector must have a length equal to the first value in the order. If no moving-average coefficient presents, set theta.vec = NULL
case	known or unknown case. When case = 'U', the parameters are estimated, when case = 'K', the parameters need to be input
method	estimation method for the control chart. When method = 'Method 3' is maximum likelihood estimations plus method of moments. Other options are 'Method 1' which is pure MLE and 'Method 2' which is pure CSS.
nsim.coefs	number of simulation for coefficients. It is functional when double.sim = TRUE.
nsim.process	number of simulation for ARMA processes

burn.in	number of burn-ins. When burn.in = 0, the ECM gets involved. When burn.in is large enough, the ACM gets involved.
sim.type	type of simulation. When sim.type = 'Matrix', the simulation is generated using matrix computation. When sim.type = 'Recursive', the simulation is based on a recursion.
verbose	print diagnostic information about fap0 and the charting constant during the simulations for the exact method

### Value

Object type double. The corrected charting constant.

### Examples

```
# load the data in the package as an example
set.seed(12345)

# Calculate the charting constant using fap0 of 0.05, and 50 observations
getCC.ARMA(fap0=0.05, m=50, nsim.coefs=10, nsim.process=10)
```

---

getCC.XBAR

*Random Flexible Level Shift Model*


---

### Description

get Phase I corrected charting constant

### Usage

```
getCC.XBAR(
  m,
  fap0 = 0.05,
  var.est = c("S", "MR"),
  ub.cons = 1,
  method = c("exact", "BA"),
  interval = c(1, 4),
  nsim = 10000,
  nu = m - 1,
  lambda = 1,
  verbose = FALSE
)
```

**Arguments**

<code>m</code>	number of subgroups when the data are subgrouped or number of observations when the data are individual.
<code>fap0</code>	nominal False Alarm Probabilty in Phase 1
<code>var.est</code>	'S' - use mean-square-based estimator, 'MR' - use moving-range-based estimator
<code>ub.cons</code>	unbiasing constant
<code>method</code>	'exact' - calculate results using the exact method, 'BA' - calculate results using the Bonfferoni approximation
<code>interval</code>	searching range of charting constants for the exact method
<code>nsim</code>	number of simulation for the exact method
<code>nu</code>	degrees of freedom; When <code>var.est = 'S'</code> , the degrees of freedom is that of the chi-squared distribution itself for the variance estimator. When <code>var.est = 'MR'</code> , the degrees of freedom is that of the chi-squared distribution approximating to the actual distribution.
<code>lambda</code>	unbiasing constant for the chi-squared distribution approximation. When <code>var.est = 'S'</code> , there is no need to do the unbiasing. When <code>var.est = 'MR'</code> , the unbiasing constant needs to be used.
<code>verbose</code>	print diagnostic information about <code>fap0</code> and the charting constant during the simulations for the exact method

**Value**

Object type double. The corrected charting constant.

**Examples**

```
set.seed(12345)

# Calculate the charting constant using 10 simulations and mean-square-based estimator
getCC.XBAR(fap0=0.05, m=20, nsim=10, var.est='S', verbose = TRUE)

# Calculate the charting constant using 10 simulations and moving-range-based estimator
getCC.XBAR(fap0=0.05, m=20, nsim=10, var.est='MR', verbose = TRUE)
```

---

grinder\_data

*Thickness measurement of silicon wafer*


---

**Description**

A dataset containing the thickness measurements in nm at different positions on the silicon wafer

**Usage**

```
grinder_data
```

**Format**

A matrix with 30 rows and 5 variables:

- pos1** Thickness measurement at Position 1 (outer circle)
- pos2** Thickness measurement at Position 2 (outer circle)
- pos3** Thickness measurement at Position 3 (middle circle)
- pos4** Thickness measurement at Position 4 (middle circle)
- pos5** Thickness measurement at Position 5 (inner circle)

**References**

Roes, Kit CB, and Ronald JMM Does. "Shewhart-type charts in nonstandard situations." *Technometrics* 37.1 (1995): 15-24

---

 PH1ARMA

---

*Phase I individual control chart with an ARMA model*


---

**Description**

Build a Phase I individual control chart for the ARMA models. The charting constant is corrected by this approach.

**Usage**

```
PH1ARMA(
  X,
  cc = NULL,
  fap0 = 0.05,
  order = c(1, 0),
  plot.option = TRUE,
  interval = c(1, 4),
  case = "U",
  phi.vec = NULL,
  theta.vec = NULL,
  mu0 = NULL,
  sigma0 = NULL,
  method = "MLE+MOM",
  nsim.coefs = 100,
  nsim.process = 1000,
  burn.in = 50,
  sim.type = "Recursive",
  transform = "none",
```

```

    lambda = 1,
    standardize = FALSE,
    verbose = FALSE
)

```

### Arguments

<code>x</code>	input and it must be a vector (m by 1)
<code>cc</code>	nominal Phase I charting constant. If this is given, the function will not re-compute the charting constant.
<code>fap0</code>	nominal false Alarm Probabilty in Phase I
<code>order</code>	order for ARMA(p, q) model
<code>plot.option</code>	- draw a plot for the process; TRUE - Draw a plot for the process, FALSE - Not draw a plot for the process
<code>interval</code>	searching range of charting constants for the exact method
<code>case</code>	known or unknown case. When case = 'U', the parameters are estimated, when case = 'K', the parameters need to be input
<code>phi.vec</code>	a vector of length p containing autoregressive coefficient(s). When case = 'K', the vector must have a length equal to the first value in the order. If no autoregressive coefficient presents, set phi.vec = NULL
<code>theta.vec</code>	a vector of length q containing moving-average coefficient(s). When case = 'K', the vector must have a length equal to the first value in the order. If no moving-average coefficient presents, set theta.vec = NULL
<code>mu0</code>	value of the IC process mean. When case = 'K', the value needs to be provided.
<code>sigma0</code>	value of the IC process standard deviation. When case = 'K', the value needs to be provided.
<code>method</code>	estimation method for the control chart. When method = 'MLE+MOM' is maximum likelihood estimations plus method of moments. Other options are 'MLE' which is pure MLE and 'CSS' which is pure CSS.
<code>nsim.coefs</code>	number of simulation for coefficients.
<code>nsim.process</code>	number of simulation for ARMA processes
<code>burn.in</code>	number of burn-ins. When burn.in = 0, the simulated process is assumed to be in the initial stage. When burn.in is sufficiently large (e.g., the default value of 50), the simulated process is assumed to have reached a stable state.
<code>sim.type</code>	type of simulation. When sim.type = 'Recursive', the simulation is generated recursively, as in the ARMA model. When sim.type = 'Matrix', the simulation is generated using the covariance matrix among observations, derived from the relationship between the ARMA coefficient(s) and the partial autocorrelation(s). Note that sim.type = 'Matrix' is primarily used as a proof of concept and is not recommended for practical use due to its high computational cost.
<code>transform</code>	type of transformation. When transform = 'none', no transformation is performed. When transform = 'boxcox', the Box-Cox transformation is used. When transform = 'yeojohnson', the Yeo-Johnson transformation is used.
<code>lambda</code>	parameter used in the Box-Cox or Yeo-Johnson transformation.



standardize	Output standardized charting statistics instead of raw ones. When standardize = TRUE, the standardization is used. When standardize = FALSE, the standardization is not performed.
verbose	print diagnostic information about fap0 and the charting constant during the simulations for the exact method

### Value

CL Object type double - central line  
 gamma Object type double - process variance estimate  
 cc Object type double - charting constant  
 order Object type integer - order for ARMA model  
 phi.vec Object type integer - values of autoregressors  
 theta.vec Object type integer - values of moving averages  
 LCL Object type double - lower charting limit  
 UCL Object type double - upper charting limit  
 CS Object type double - charting statistic

### References

Yao, Y., Chakraborti, S., Yang, X., Parton, J., Lewis Jr, D., and Hudnall, M. (2023). Phase I control chart for individual autocorrelated data: application to prescription opioid monitoring. *Journal of Quality Technology*, 55(3), 302-317.

### Examples

```
# load the data in the package as an example
data(preston_data)

# set number of simulations
nsim.process <- 10
nsim.coefs <- 10

# An example using the default setting whose fap0 = 0.1
PH1ARMA(preston_data, nsim.process = nsim.process, nsim.coefs = nsim.coefs)

# When users get an error message about the size of matrix,
# the function needs to use the alternative simulation type as follows
PH1ARMA(preston_data, fap0 = 0.05,
nsim.process = nsim.process, nsim.coefs = nsim.coefs, sim.type = 'Recursive')
```

PH1XBAR

*Phase I X-bar control chart with a corrected charting constant***Description**

Build a Phase I Shewhart control chart for the variance components model if the data are subgrouped or for the basic Shewhart model if the data are individual. The charting constant is corrected by this approach.

**Usage**

```
PH1XBAR(
  X,
  cc = NULL,
  fap0 = 0.05,
  var.est = c("S", "MR"),
  ub.option = TRUE,
  method = c("exact", "BA"),
  plot.option = TRUE,
  interval = c(1, 4),
  nsim = 10000,
  transform = "none",
  lambda = 1,
  standardize = FALSE,
  verbose = FALSE
)
```

**Arguments**

X	input and it must be a matrix (m by n) or a vector (m by 1)
cc	nominal Phase I charting constant. If this is given, the function will not recompute the charting constant.
fap0	nominal False Alarm Probability in Phase 1
var.est	'S' - use mean-square-based estimator, 'MR' - use moving-range-based estimator
ub.option	TRUE - the standard deviation estimator corrected by a unbiasing constant. For S, it is c4 and for MR, it is d2. FALSE - no unbiasing constant
method	'exact' - calculate results using the exact method, 'BA' - calculate results using the Bonferroni approximation
plot.option	- draw a plot for the process; TRUE - Draw a plot for the process, FALSE - Not draw a plot for the process
interval	searching range of charting constants for the exact method
nsim	number of simulation for the exact method

transform	type of transformation. When transform = 'none', no transformation is performed. When transform = 'boxcox', the Box-Cox transformation is used. When transform = 'yeojohnson', the Yeo-Johnson transformation is used.
lambda	parameter used in the Box-Cox or Yeo-Johnson transformation.
standardize	Output standardized charting statistics instead of raw ones. When standardize = TRUE, the standardization is used. When standardize = FALSE, the standardization is not performed.
verbose	print diagnostic information about fap0 and the charting constant during the simulations for the exact method

### Value

CL Object type double - central line  
var.est Object type double - variance estimate  
ub.cons Object type double - unbiasing constant  
cc Object type double - charting constant  
m Object type integer - number of subgroups when X is a matrix or number of observations when X is a vector  
nu Object type integer - degrees of freedom; When var.est = 'S', the degrees of freedom is that of the chi-squared distribution itself for the variance estimator. When var.est = 'MR', the degrees of freedom is that of the chi-squared distribution approximating to the actual distribution.  
lambda Object type integer - chi-squared unbiasing constant for the chi-squared distribution approximation  
LCL Object type double - lower charting limit  
UCL Object type double - upper charting limit  
CS Object type double - charting statistic

### References

Champ, C.W., and Jones, L.A. (2004) Designing Phase I X-bar charts with small sample sizes. *Quality and Reliability Engineering International*. 20(5), 497-510  
Yao, Y., Hilton, C.W., and Chakraborti, S. (2017) Designing Phase I Shewhart X-bar charts: Extended tables and software. *Quality and Reliability Engineering International*. 33(8), 2667-2672.  
Yao, Y., and Chakraborti, S. (2021). Phase I monitoring of individual normal data: Design and implementation. *Quality Engineering*, 33(3), 443-456.  
Yao, Y., and Chakraborti, S. (2021). Phase I process monitoring: The case of the balanced one-way random effects model. *Quality and Reliability Engineering International*, 37(3), 1244-1265.

### Examples

```
set.seed(12345)

# load the data in the package as an example
data(grinder_data)
```

```
# An example using a false alarm probability of 0.05, and 10 simulations
PH1XBAR(grinder_data, fap0 = 0.05, nsim=10, verbose=TRUE)
```

---

pistonring\_data      *Pistonring data*

---

### Description

A dataset containing piston ring data

### Usage

```
pistonring_data
```

### Format

A data frame with 25 rows and 5 variables:

**X1** Observation 1 in subgroups

**X2** Observation 2 in subgroups

**X3** Observation 3 in subgroups

**X4** Observation 4 in subgroups

**X5** Observation 5 in subgroups

### References

Montgomery, Douglas C. 2005. Introduction to Statistical Quality Control. John Wiley & Sons.

---

preston\_data      *Prescription fentanyl consumption in Preston county, WV*

---

### Description

A dataset containing prescription fentanyl consumption in Preston county, WV, measured using MME percapita. This is a subset from Rich et al. <doi: 10.21105/joss.02450>

### Usage

```
preston_data
```

### Format

A vector with 60 elements

## References

Rich, S., Tran, A. B., Williams, A., Holt, J., Sauer, J., & Oshan, T. M. (2020). arcos and arcosp: R and Python packages for accessing the DEA ARCOS database from 2006-2014. *Journal of Open Source Software*, 5(53), 2450.

---

semiconductor_data	<i>Semiconductor data</i>
--------------------	---------------------------

---

## Description

A dataset containing the 151st feature in SECOM dataset

## Usage

```
semiconductor_data
```

## Format

A vector with 50 observations:

**obs** the 151st feature

## References

McCann, Michael, and Adrian Johnston. 2008. "SECOM." UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C54305>.

---

snowfall_data	<i>Seasonal snowfall in inches in Minneapolis/St. Paul, MN</i>
---------------	--

---

## Description

A dataset containing snowfalls measured in inches in Minneapolis/St. Paul, MN.

## Usage

```
snowfall_data
```

## Format

A data frame with 82 rows and 4 variables:

**Year** year of the snowfalls

**jan** snowfalls in January

**feb** snowfalls in February

**mar** snowfalls in March

**References**

Mukherjee, P. S. (2016). On phase II monitoring of the probability distributions of univariate continuous processes. *Statistical Papers*, 57(2), 539-562.

# Index

## \* **datasets**

- bore\_diameter\_data, [3](#)
- grinder\_data, [6](#)
- pistonring\_data, [12](#)
- preston\_data, [12](#)
- semiconductor\_data, [13](#)
- snowfall\_data, [13](#)

## \* **package**

- PH1XBAR-package, [2](#)

bore\_diameter\_data, [3](#)

getCC (getCC.XBAR), [5](#)

getCC.ARMA, [4](#)

getCC.XBAR, [5](#)

grinder\_data, [6](#)

PH1ARMA, [7](#)

PH1XBAR, [10](#)

PH1XBAR-package, [2](#)

pistonring\_data, [12](#)

preston\_data, [12](#)

semiconductor\_data, [13](#)

snowfall\_data, [13](#)