# cvcqv: A Package for Estimation of Relative Variability

*by Maani Beigy*

**Abstract** Coefficient of variation (*cv*) and coefficient of quartile variation (*cqv*) are widely used measures of relative dispersion which play descriptive and inferential roles (*e.g.*, reliability analysis, quality control, inequality measurement, and anomaly detection) in various fields such as biological and medical sciences, economics, actuarial sciences, etc. Since *cv* and *cqv* are unit-free, they are useful for comparing data from different distributions, data from different scales, or widely different means. However, to avoid their common misuses, confidence intervals (*CI*) are required. The **cvcqv** package provides a home for such tools. To our knowledge, the new `R` package **cvcqv** is the first `R` implementation of **cqv** as a robust variability measure, with almost all available methods for *CI* of *cv* and *cqv*. This paper elucidates this versatile functionality using reproducible examples on real datasets. Also, the new insights that **cvcqv**, alongside other `R` packages, brings into data science will be discussed.

## Introduction

Researchers and practitioners in various fields use the coefficient of variation (*cv*) as a measure of relative variability (Panichkitkosolkul, 2013; Payton, 1996). *cv* is calculated as the ratio of the sample standard deviation (*sd*) to the sample mean ($\bar{x}$). However, *cv* is often misleading for variables with non-ratio scales (Payton, 1996), for homoscedastic data, and for variables without different magnitudes or units (Shechtman, 2013).

Robust statistical measurements such as coefficient of quartile variation (*cqv*) are better alternatives in non-normal distributions (Altunkaynak and Gamgam, 2018):

$$cqv = \left(\frac{q_3 - q_1}{q_3 + q_1}\right) \times 100$$

where $q_3$ and $q_1$ are the sample third quartile (*i.e.*, $75^{th}$ percentile) and first quartile (*i.e.*, $25^{th}$ percentile), respectively.

Almost always, we calculate *cv* and *cqv* from samples but the final objective is to generalize them as the populations' parameters (Albatineh et al., 2014). For example, one may be interested in comparing the variabilities of the time-varying measurements of a variable to detect anomalies such as extreme behaviors of customers or institutes (as in actuarial sciences). Or someone might inquire into whether a laboratory test or technique has sufficient inter-assay and intra-assay reliability (Panichkitkosolkul, 2013; Payton, 1996). In such scenarios, variabilities calculated from samples are often biased and misleading (Sørensen, 2002; Payton, 1996). Therefore, various confidence intervals (*CI*) have been introduced to correctly estimate the relative variability.

This paper sets out to demonstrate the versatility of **cvcqv** package (Beigy, 2019a) in a variety of data science tasks related to variability measurement. `R` (R Core Team, 2016) provides a strong asset for progress in this direction because it already contains functionality used in a variety of packages like **DescTools**, **MBESS**, **goeveg**, and **sjstats**. However, robust variability measures such as *cqv* has been missing in `R` for a long time. Moreover, the implementations of *CI* for *cv* have been limited to one or two methods. Lack of functions for the rigorous methods of calculation of *CI* for *cv* and *cqv*, though available in the statistical literature, was a major motivation to develop this package and explain its versatile functionality in this paper.

## Package structure and functionality

The package can be installed and loaded as follows (see the package's README for dependencies and access to development versions):

```
install.packages("cvcqv")

library(cvcqv)
```

**cvcqv** depends on **dplyr** (Wickham et al., 2019) for using `nth()` function and imports **R6** (Chang, 2019) for `"R6"` classes, **SciViews** (Grosjean, 2018) for `ln()` function, **boot** (Canty and Ripley, 2019) for bootstrapping methods, and **MBESS** (Kelley, 2018) for noncentral distributions.

### Core functions and classes

The functionality of the package is developed as both simple functions and "R6" classes, for sake of versatility, portability and efficiency:

- The R6 class "SampleQuantiles" to produce the sample quantiles corresponding to the given probabilities. It uses quantile function from the built-in R package **stats**, but provides an "R6" interface to be inherited for other classes.
- The R6 class "BootCoefVar" produces the bootstrap resampling for the *cv*. It uses boot.ci function from **boot**, but provides an "R6" interface to be inherited for child classes.
- The R6 class "BootCoefQuartVar" produces the bootstrap resampling for the *cqv*. It uses boot and boot.ci functions from **boot**, but provides an "R6" interface to be inherited for child classes.
- The R6 class "CoefVar" calculates the sample *cv*.
- The R6 class "CoefQuartVar" calculates the sample *cqv*.
- The R6 class "CoefVarCI" calculates *CI* for *cv*.
- The R6 class "CoefQuartVarCI" calculates *CI* for *cqv*.
- The function cv_versatile calculates *cv* and its various *CIs*.
- The function cqv_versatile calculates *cqv* and its various *CIs*.

### R6 Objects Tree

```
SampleQuantiles
        ├──── BootCoefVar
        │      └──────────── CoefVar
        │                     └──────────── CoefVarCI
        │
        └──── BootCoefQuartVar
               └──────────── CoefQuartVar
                              └──────────── CoefQuartVarCI
```

### Confidence Interval Methods

There are various methods for the calculation of *CI* for *cv* and *cqv*, which have been implemented in **cvcqv** package:

**Table 1:** Methods for calculation of *CI* for *cv* and *cqv*

| cv | cqv |
|---|---|
| "kelley" (2018; 2007) | "bonett" (2006) |
| "mckay" (1932) | "norm" (2018) |
| "miller" (1991) | "basic" (2018) |
| "vangel" (1996) | "perc" (2018) |
| "mahmoudvand_hassani" (2009) | "bca" (2018) |
| "equal_tailed" (2013) | |
| "shortest_length" (2013) | |
| "normal_approximation" (2013) | |
| "norm" (2019; 1997) | |
| "basic" (2019; 1997) | |
| "perc" (2019; 1997) | |
| "bca" (2019; 1997) | |

For more statistical details on these methods, read the vignettes provided for cv and cqv (Beigy, 2019a).

## Solutions for real-world problems

This section contains examples on real-world data science problems:

### Consistency or Reliability of Measurements

Instruments and measurements have to be not only valid but also reliable. Reliability is defined as the extent to which they measure the variables consistently (Shechtman, 2013). Reliability may also be called as consistency, repeatability, reproducibility, stability, and precision (Shechtman, 2013).

The *cv* and *cqv* can be used as indicators of reliability because they assesses the stability of measurements across repeated tests. An advantage of dimensionless measures such as *cv* and *cqv* is that they allow us to make direct comparisons between the measurements regardless of the scale or calibration. Hence, they enable us to compare reliability among instruments and assays (Shechtman, 2013; Hopkins, 2000).

In terms of assessing the reliability of measurements, the interesting questions How to measure consistency of measurement over time and How to measure the consistency of improvement on different conditions? on Cross Validated community statistics, properly address such problem (tach, 2017; Ida, 2019). Inspired by them, a sample `data.frame` named `wine.csv` for testing the quality of **five different type of wines** by **three experts** was created. A small chunk of the `data.frame` is:

```
     expert    measurement Wine_1 Wine_2 Wine_3 Wine_4 Wine_5
1    expert_a  2019-01-01   0.70   0.60   0.30   0.10   0.80
2    expert_a  2019-01-02   0.60   0.70   0.40   0.20   0.80
3    expert_a  2019-01-03   0.65   0.65   0.35   0.15   0.80
44   expert_b  2019-01-04   0.90   0.10   0.90   0.10   0.90
45   expert_b  2019-01-05   0.20   0.12   0.21   0.31   0.21
46   expert_b  2019-01-06   0.80   0.56   0.79   0.89   0.69
115  expert_c  2019-02-04   0.43   0.24   0.15   0.68   0.92
116  expert_c  2019-02-05   0.42   0.32   0.16   0.69   0.91
117  expert_c  2019-02-06   0.41   0.31   0.15   0.70   0.90
```

Then, we prepare the data using the **tidyverse** (Wickham, 2017) packages. We need the `wine` `data.frame` in the long format:

```
library(tidyverse)
wine_gather %>% gather(
  key = "wines",
  value = "score",
  Wine_1:Wine_5, -measurement
)
```

Then, we test the normality of `scores` variable:

```
shapiro.test(wine$score)
        Shapiro-Wilk normality test

data:  wine_gather$score
W = 0.89857, p-value < 2.2e-16
```

Because of the non-normal distribution of the `scores` variable, we calculate the *cqv* with *Bootstrap percentile 95% CI* using **cvcqv** `R6` class `CoefQuartVarCI` with `perc_ci` method:

```
library(cvcqv)
wine_gather %>% group_by(expert, wines) %>% summarise(
  cqv_est = cvcqv::CoefQuartVarCI$new(
    x = score, na.rm = TRUE, alpha = 0.05, R = 100, digits = 3,
  )$perc_ci()$statistics$est,
  cqv_lower = cvcqv::CoefQuartVarCI$new(
    x = score, na.rm = TRUE, alpha = 0.05, R = 100, digits = 3,
  )$perc_ci()$statistics$lower,
  cqv_upper = cvcqv::CoefQuartVarCI$new(
    x = score, na.rm = TRUE, alpha = 0.05, R = 100, digits = 3,
  )$perc_ci()$statistics$upper
)

  expert   wines  cqv_est cqv_lower cqv_upper
 1 expert_a Wine_1    5.58     3.33       6.15
 2 expert_a Wine_2    3.3      2.33       4.70
 3 expert_a Wine_3    6.02     4.22       8.01
 4 expert_a Wine_4   12.5      7.06      18.8
```

```
 5 expert_a Wine_5    1.38     0.621      2.5
 6 expert_b Wine_1   70.3     47.1       75.6
 7 expert_b Wine_2   66.0     52.9       69.1
 8 expert_b Wine_3   58       55.3       58.4
 9 expert_b Wine_4   45.8     31.2       70.8
10 expert_b Wine_5   49.9     13.3       53.6
11 expert_c Wine_1   30.1     18.3       53.7
12 expert_c Wine_2   49.6     10.7       52.3
13 expert_c Wine_3   70.9     39.2       72.4
14 expert_c Wine_4   14.5      4.74      15.9
15 expert_c Wine_5   70.7      9.61      76.0
```

As you see in figure 1, only the **expert_a** shows consistent measurements for various wines over time; because large measurements with *cqv* or *cv* values (here higher than 10%) are generally considered non-reliable (Beigy, 2019b):

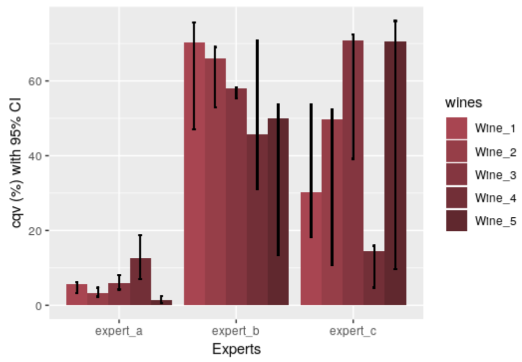


**Figure 1:** The consistency of experts' scores on the different wine types over time

### Detection of Outliers and Anomalies

The *cv* and *cqv* play direct and indirect roles to detect outliers (Zhou and Xu, 2014) and anomalies (Fathnia and Bayaz, 2018). Detecting variables or measurements with high *cv* or *cqv* migth be helpful before employing anomaly detection techniques. An example is provided here based on the question of Ida (2019), investigating the speed improvement of various cases (codes in `anomaly.R` file). Let us use the `data.frame` called `speed_tbl_df`:

```
head(speed_tbl_df)
        date case speed
1 2019-01-01    A   1.2
2 2019-01-02    A   1.3
3 2019-01-03    A   1.1
4 2019-01-04    A   1.1
5 2019-01-05    A   1.5
6 2019-01-01    B   1.2
```

We calculate *cv* and *cqv* with *Basic bootstrap 95% CI* for cases A, B, and C:

```
head(speed_tbl_df)
cqv_speed_A <- cvcqv::CoefQuartVarCI$new(
  x = subset(speed_tbl_df, case == "A")$speed,
  na.rm = TRUE,
  digits = 3,
  R = 1000,
  alpha = 0.05
  )$basic_ci()

cv_speed_A <- cvcqv::CoefVarCI$new(
  x = subset(speed_tbl_df, case == "A")$speed,
  na.rm = TRUE,
  digits = 3,
```

```
  R = 1000,
  alpha = 0.05
)$basic_ci()

# do the same for the remaining cases B and C.Then, collect them all in one df:
cvcqv_speed <- dplyr::bind_rows(list(
  cqv_speed_A$statistics,
  cqv_speed_B$statistics,
  cqv_speed_C$statistics,
  cv_speed_A$statistics,
  cv_speed_B$statistics,
  cv_speed_C$statistics
))
attr(cvcqv_speed, "row.names") <- c(
  "cqv_A", "cqv_B", "cqv_C", "cv_A", "cv_B", "cv_C"
  )
cvcqv_speed
          est  lower   upper
cqv_A   8.333  1.282  16.667
cqv_B  92.308 86.791 184.615
cqv_C  42.857 28.852  85.714
cv_A   13.495  9.627  22.996
cv_B  133.997 80.843 214.743
cv_C   54.696 36.741  87.258
```

As I explained in the question on Cross Validated (Beigy, 2019c), case A shows minimal variability (≈ 8%); case B shows severe variation (≈ 92%), and case C shows moderate variation (≈ 43%). In cases with severe variation, it is more probable to find anomalies. Here, **anomalize** (Dancho and Vaughan, 2018) package may be helpful:

```
anomalize::anomalize(speed_tbl_df, speed, method = "iqr", alpha = 0.05)
# A tibble: 15 x 6
   date       case  speed speed_l1 speed_l2 anomaly
   <date>     <fct> <dbl>    <dbl>    <dbl> <chr>
 1 2019-01-01 A       1.2    -5.90     10.5 No
 2 2019-01-02 A       1.3    -5.90     10.5 No
 3 2019-01-03 A       1.1    -5.90     10.5 No
 4 2019-01-04 A       1.1    -5.90     10.5 No
 5 2019-01-05 A       1.5    -5.90     10.5 No
 6 2019-01-01 B       1.2    -5.90     10.5 No
 7 2019-01-02 B       1.1    -5.90     10.5 No
 8 2019-01-03 B      20      -5.90     10.5 Yes
 9 2019-01-04 B      30      -5.90     10.5 Yes
10 2019-01-05 B     100      -5.90     10.5 Yes
11 2019-01-01 C       1.2    -5.90     10.5 No
12 2019-01-02 C       1.1    -5.90     10.5 No
13 2019-01-03 C       2      -5.90     10.5 No
14 2019-01-04 C       3      -5.90     10.5 No
15 2019-01-05 C       4      -5.90     10.5 No
```

As you can see in the anomaly column of the result, the speed improvements of "`20,30,100`" of case B (the one with severe variability based on *cqv*) are anomalies/outliers.

## Bibliography

A. N. Albatineh, B. M. Kibria, M. L. Wilcox, and B. Zogheib. Confidence interval estimation for the population coefficient of variation using ranked set sampling: A simulation study. 41(4):733–751, 2014. ISSN 02664763. URL `https://doi.org/10.1080/02664763.2013.847405`. [p1]

B. Altunkaynak and H. Gamgam. Bootstrap confidence intervals for the coefficient of quartile variation. 0(0):1–9, 2018. ISSN 15324141. doi: 10.1080/03610918.2018.1435800. URL `https://doi.org/10.1080/03610918.2018.1435800`. [p1, 2]

M. Beigy. *cvcqv: Coefficient of Variation (CV) with Confidence Intervals (CI)*, 2019a. URL `https://CRAN.R-project.org/package=cvcqv`. R package version 1.0.0. [p1, 2]

M. Beigy. How to measure consistency of measurement over time. Cross Validated, 2019b. URL https://stats.stackexchange.com/q/422666. URL:https://stats.stackexchange.com/q/422666 (version: 2019-08-18). [p4]

M. Beigy. How to measure the consistency of improvement on different conditions? Cross Validated, 2019c. URL https://stats.stackexchange.com/q/399073. URL:https://stats.stackexchange.com/q/399073 (version: 2019-08-18). [p5]

D. G. Bonett. Confidence interval for a coefficient of quartile variation. 50(11):2953–2957, 2006. ISSN 01679473. URL https://doi.org/10.1016/j.csda.2005.05.007. [p2]

A. Canty and B. D. Ripley. *boot: Bootstrap R (S-Plus) Functions*, 2019. R package version 1.3-22. [p1, 2]

W. Chang. *R6: Encapsulated Classes with Reference Semantics*, 2019. URL https://CRAN.R-project.org/package=R6. R package version 2.4.0. [p1]

M. Dancho and D. Vaughan. *anomalize: Tidy Anomaly Detection*, 2018. URL https://CRAN.R-project.org/package=anomalize. R package version 0.1.1. [p5]

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge, 1997. URL https://doi.org/10.1017/CBO9780511802843. ISBN 0-521-57391-2. [p2]

G. Edward Miller. Asymptotic test statistics for coefficients of variation. 20(10):3351–3363, 1991. ISSN 0361-0926. [p2]

F. Fathnia and M. H. J. D. Bayaz. Anomaly detection in smart grid with help of an improved optics using coefficient of variation. In *Electrical Engineering (ICEE), Iranian Conference on*, pages 1044–1050. IEEE, 2018. URL https://doi.org/10.1109/ICEE.2018.8472534. [p4]

P. Grosjean. *SciViews-R*. UMONS, MONS, Belgium, 2018. URL http://www.sciviews.org/SciViews-R. [p1]

W. G. Hopkins. Measures of reliability in sports medicine and science. *Sports medicine*, 30(1):1–15, 2000. URL https://doi.org/10.2165/00007256-200030010-00001. [p3]

Ida. How to measure the consistency of improvement on different conditions? Cross Validated, 2019. URL https://stats.stackexchange.com/q/398462. URL:https://stats.stackexchange.com/q/398462 (version: 2019-03-20). [p3, 4]

K. Kelley. Sample size planning for the coefficient of variation from the accuracy in parameter estimation approach. 39(4):755–766, 2007. ISSN 1554351X. URL https://doi.org/10.3758/BF03192966. [p2]

K. Kelley. *MBESS: The MBESS R Package*, 2018. URL https://CRAN.R-project.org/package=MBESS. R package version 4.4.3. [p1, 2]

R. Mahmoudvand and H. Hassani. Two new confidence intervals for the coefficient of variation in a normal distribution. 36(4):429–442, 2009. ISSN 0266-4763. URL https://doi.org/10.1080/02664760802474249. [p2]

A. T. McKay. Distribution of the Coefficient of Variation and the Extended" t" Distribution. 95(4): 695–698, 1932. ISSN 0952-8385. [p2]

W. Panichkitkosolkul. Confidence Intervals for the Coefficient of Variation in a Normal Distribution with a Known Population Mean. 2013:1–11, 2013. ISSN 1687-952X. URL https://doi.org/10.1155/2013/324940. [p1, 2]

M. E. Payton. Confidence Intervals for the Coefficient of Variation. pages 82–90, 1996. URL https://doi.org/10.4148/2475-7772.1320. [p1]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL https://www.R-project.org/. ISBN 3-900051-07-0. [p1]

O. Shechtman. *The Coefficient of Variation as an Index of Measurement Reliability*, pages 39–49. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-37131-8. URL https://doi.org/10.1007/978-3-642-37131-8_4. [p1, 3]

J. B. Sørensen. The use and misuse of the coefficient of variation in organizational demography research. *Sociological methods & research*, 30(4):475–491, 2002. URL https://doi.org/10.1177/0049124102030004001. [p1]

tach. How to measure consistency of measurement over time. Cross Validated, 2017. URL https://stats.stackexchange.com/q/318044. URL:https://stats.stackexchange.com/q/318044 (version: 2017-12-10). [p3]

M. G. Vangel. Confidence intervals for a normal coefficient of variation. 50(1):21–26, 1996. ISSN 0003-1305. [p2]

H. Wickham. *tidyverse: Easily Install and Load the 'Tidyverse'*, 2017. URL https://CRAN.R-project.org/package=tidyverse. R package version 1.2.1. [p3]

H. Wickham, R. François, L. Henry, and K. Müller. *dplyr: A Grammar of Data Manipulation*, 2019. URL https://CRAN.R-project.org/package=dplyr. R package version 0.8.3. [p1]

S. B. Zhou and W. X. Xu. Local outlier detection algorithm based on coefficient of variation. In *Advanced Design and Manufacturing Technology IV*, volume 635 of *Applied Mechanics and Materials*, pages 1723–1728. Trans Tech Publications Ltd, 11 2014. URL https://doi.org/10.4028/www.scientific.net/AMM.635-637.1723. [p4]

*Maani Beigy*
*Department of Epidemiology and Biostatistics*
*School of Public Health*
*Tehran University of Medical Sciences*
*Tehran*
*Iran*
*ORCiD: 0000-0003-2963-3533*
manibeygi@gmail.com
m-beigy@alumnus.tums.ac.ir