Covariance, Correlation, and Regression Tutorial

Christopher J. Schmank, PhD

2026-02-04

Load Libraries

```
#install.packages(tidyverse, dependencies = TRUE)
#install.packages(mice, dependencies = TRUE)
#install.packages(psych, dependencies = TRUE)
#install.packages(jmv, dependencies = TRUE)
#install.packages(apaTables, dependencies = TRUE)

library(tidyverse)
library(mice)
library(psych)
library(jmv)
library(apaTables)
```

Variance and Covariance Assessment: Complete Data Examples

```
var(attitude)
```

```
##               rating complaints privileges  learning    raises critical
## rating     148.17126  133.77931   63.46437  89.10460  74.68851 18.84253
## complaints 133.77931  177.28276   90.95172  93.25517  92.64138 24.73103
## privileges  63.46437   90.95172  149.70575  70.84598  56.67126 17.82529
## learning    89.10460   93.25517   70.84598 137.75747  78.13908 13.46782
## raises      74.68851   92.64138   56.67126  78.13908 108.10230 38.77356
## critical    18.84253   24.73103   17.82529  13.46782  38.77356 97.90920
## advance     19.42299   30.76552   43.21609  64.19770  61.42299 28.84598
##              advance
## rating      19.42299
## complaints  30.76552
## privileges  43.21609
## learning    64.19770
## raises      61.42299
## critical    28.84598
## advance    105.85747
```

```
cov(attitude)
```

```
##               rating complaints privileges  learning    raises critical
## rating     148.17126  133.77931   63.46437  89.10460  74.68851 18.84253
## complaints 133.77931  177.28276   90.95172  93.25517  92.64138 24.73103
## privileges  63.46437   90.95172  149.70575  70.84598  56.67126 17.82529
## learning    89.10460   93.25517   70.84598 137.75747  78.13908 13.46782
## raises      74.68851   92.64138   56.67126  78.13908 108.10230 38.77356
## critical    18.84253   24.73103   17.82529  13.46782  38.77356 97.90920
## advance     19.42299   30.76552   43.21609  64.19770  61.42299 28.84598
##              advance
## rating      19.42299
## complaints  30.76552
## privileges  43.21609
## learning    64.19770
## raises      61.42299
## critical    28.84598
## advance    105.85747
```

- These functions will work interchangeably by default IF AND ONLY IF there is no missing data!

Variance and Covariance Assessment: Incomplete Data Examples

```
var(nhanes)
```

```
##      age bmi hyp chl
## age 0.69  NA  NA  NA
## bmi   NA  NA  NA  NA
## hyp   NA  NA  NA  NA
## chl   NA  NA  NA  NA
```

```
cov(nhanes) # By default this function sets `use = "everything"`
```

```
##      age bmi hyp chl
## age 0.69  NA  NA  NA
## bmi   NA  NA  NA  NA
## hyp   NA  NA  NA  NA
## chl   NA  NA  NA  NA
```

- When missing data is present and default `var()` and `cov()` are used they result in same variance-covariance matrices —unusable in this form!!

```
var(nhanes, na.rm=TRUE) #All cases with NA removed
```

```
##            age        bmi       hyp         chl
## age  0.5641026 -1.1621795 0.1602564   21.935897
## bmi -1.1621795 21.1158974 0.2153846   83.687179
## hyp  0.1602564  0.2153846 0.1923077    9.173077
## chl 21.9358974 83.6871795 9.1730769 2378.064103
```

```
cov(nhanes, use = "complete.obs") #All cases with NA removed
```

```
##            age        bmi       hyp         chl
## age  0.5641026 -1.1621795 0.1602564   21.935897
## bmi -1.1621795 21.1158974 0.2153846   83.687179
## hyp  0.1602564  0.2153846 0.1923077    9.173077
## chl 21.9358974 83.6871795 9.1730769 2378.064103
```

```
cov(nhanes, use = "pairwise.complete.obs") #Pairwise deletion --- different N's
```

```
##            age         bmi        hyp         chl
## age  0.6900000 -1.30750000 0.18382353   18.328571
## bmi -1.3075000 17.76783333 0.09666667   83.687179
## hyp  0.1838235  0.09666667 0.19117647    8.554945
## chl 18.3285714 83.68717949 8.55494505 2044.400000
```

Calculating Correlation Matrix

Correlations can be generated with several different functions in base R but get familiar with the `psych` and `JMV` package function calls for correlation matrices

1. If all you want is a correlation matrix without any additional information: Use `cor()`

```
cor(attitude)
```

```
##               rating complaints privileges  learning    raises  critical
## rating     1.0000000  0.8254176  0.4261169 0.6236782 0.5901390 0.1564392
## complaints 0.8254176  1.0000000  0.5582882 0.5967358 0.6691975 0.1877143
## privileges 0.4261169  0.5582882  1.0000000 0.4933310 0.4454779 0.1472331
## learning   0.6236782  0.5967358  0.4933310 1.0000000 0.6403144 0.1159652
## raises     0.5901390  0.6691975  0.4454779 0.6403144 1.0000000 0.3768830
## critical   0.1564392  0.1877143  0.1472331 0.1159652 0.3768830 1.0000000
## advance    0.1550863  0.2245796  0.3432934 0.5316198 0.5741862 0.2833432
##              advance
## rating     0.1550863
## complaints 0.2245796
## privileges 0.3432934
## learning   0.5316198
## raises     0.5741862
## critical   0.2833432
## advance    1.0000000
```

```
cor(attitude[1:4]) #Can use subsetting to only pull certain variables into matrix
```

```
##               rating complaints privileges  learning
## rating     1.0000000  0.8254176  0.4261169 0.6236782
## complaints 0.8254176  1.0000000  0.5582882 0.5967358
## privileges 0.4261169  0.5582882  1.0000000 0.4933310
## learning   0.6236782  0.5967358  0.4933310 1.0000000
```

```
cor(nhanes) #If missing data is present you must remove it or set `use =` argument
```

```
##     age bmi hyp chl
## age   1  NA  NA  NA
## bmi  NA   1  NA  NA
## hyp  NA  NA   1  NA
## chl  NA  NA  NA   1
```

2. If you want to get correlation on ONLY 2 variables at once: Use `cor.test()`

```
cor.test(~rating + learning, data = attitude)
```

```
##
##  Pearson's product-moment correlation
##
## data:  rating and learning
## t = 4.2219, df = 28, p-value = 0.0002311
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3397475 0.8034243
## sample estimates:
##       cor
## 0.6236782
```

```
cor.test(x=attitude$rating, y=attitude$learning)
```

```
##
##  Pearson's product-moment correlation
##
## data:  attitude$rating and attitude$learning
## t = 4.2219, df = 28, p-value = 0.0002311
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3397475 0.8034243
## sample estimates:
##       cor
## 0.6236782
```

- Provides *t*-test, degrees of freedom, and *p*-value to assess if correlation is statistically significantly different from zero
- Provides correlation coefficient and 95% confidence interval
- Provides type of correlation coefficient calculated (i.e., Pearson's product-moment correlation)

3. For more detail and details needed for publication: Use `corr.test()` or `corrMatrix`

```
corr.test(attitude)
```

```
## Call:corr.test(x = attitude)
## Correlation matrix
##            rating complaints privileges learning raises critical advance
## rating       1.00       0.83       0.43     0.62   0.59     0.16    0.16
## complaints   0.83       1.00       0.56     0.60   0.67     0.19    0.22
## privileges   0.43       0.56       1.00     0.49   0.45     0.15    0.34
## learning     0.62       0.60       0.49     1.00   0.64     0.12    0.53
## raises       0.59       0.67       0.45     0.64   1.00     0.38    0.57
## critical     0.16       0.19       0.15     0.12   0.38     1.00    0.28
## advance      0.16       0.22       0.34     0.53   0.57     0.28    1.00
## Sample Size
## [1] 30
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##            rating complaints privileges learning raises critical advance
## rating       0.00       0.00       0.19     0.00   0.01     1.00    1.00
## complaints   0.00       0.00       0.02     0.01   0.00     1.00    1.00
## privileges   0.02       0.00       0.00     0.07   0.15     1.00    0.51
## learning     0.00       0.00       0.01     0.00   0.00     1.00    0.03
## raises       0.00       0.00       0.01     0.00   0.00     0.36    0.01
## critical     0.41       0.32       0.44     0.54   0.04     0.00    0.90
## advance      0.41       0.23       0.06     0.00   0.00     0.13    0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

```
corrMatrix(attitude,
           vars = vars(rating, complaints, privileges,learning))
```

```
##
##  CORRELATION MATRIX
##
##  Correlation Matrix
##  ─────────────────────────────────────────────────────────────────────────────────────
##                                rating       complaints    privileges    learning
##  ─────────────────────────────────────────────────────────────────────────────────────
##    rating        Pearson's r            —
##                  df                     —
##                  p-value                —
##
##    complaints    Pearson's r    0.8254176             —
##                  df                    28             —
##                  p-value       < .0000001             —
##
##    privileges    Pearson's r    0.4261169     0.5582882             —
##                  df                    28            28             —
##                  p-value        0.0188770     0.0013455             —
##
##    learning      Pearson's r    0.6236782     0.5967358     0.4933310             —
##                  df                    28            28            28             —
##                  p-value        0.0002311     0.0005000     0.0056024             —
##  ─────────────────────────────────────────────────────────────────────────────────────
```

4. If you need to save a correlation matrix for later use: Use `corr.test()` or `cor()`

```
cor_dat <- psych::corr.test(attitude[1:4])

cor_dat$r # Calls for ONLY the correlation matrix portion of corr.test() output
```

```
##               rating complaints privileges  learning
## rating     1.0000000  0.8254176  0.4261169 0.6236782
## complaints 0.8254176  1.0000000  0.5582882 0.5967358
## privileges 0.4261169  0.5582882  1.0000000 0.4933310
## learning   0.6236782  0.5967358  0.4933310 1.0000000
```

```
cor_dat_r <- cor_dat$r # May also want to make an R object for the correlations
```

```
cor_dat2 <- cor(attitude[1:4]) # This code creates a similar object as `cor_dat_r` above
cor_dat2
```

```
##               rating complaints privileges  learning
## rating     1.0000000  0.8254176  0.4261169 0.6236782
## complaints 0.8254176  1.0000000  0.5582882 0.5967358
## privileges 0.4261169  0.5582882  1.0000000 0.4933310
## learning   0.6236782  0.5967358  0.4933310 1.0000000
```

```
# Can even use these objects to do rudimentary matrix algebra

cor_dat_r - cor_dat2
```

```
##                   rating complaints privileges      learning
## rating      0.000000e+00          0          0 -1.110223e-16
## complaints  0.000000e+00          0          0  0.000000e+00
## privileges  0.000000e+00          0          0  0.000000e+00
## learning   -1.110223e-16          0          0  0.000000e+00
```

Creating Correlation Table Document

The `apaTables` package can be used to create quick correlation matrices and linear regression tables—these can be helpful but are VERY difficult to reformat or make changes to.

Make sure that you know how to create these tables using other programs like Excel and Word as these allow for much more customization!

```
apa.cor.table(data = attitude[1:4],
              # filename = "attitude-correlation-matrix.doc", # Can uncomment this line to create word doc
              table.number = 1)
```

```
##
##
## Table 1
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##   Variable      M     SD    1          2          3
##   1. rating     64.63 12.17
##
##   2. complaints 66.60 13.31 .83**
##                             [.66, .91]
##
##   3. privileges 53.13 12.24 .43*       .56**
##                             [.08, .68] [.25, .76]
##
##   4. learning   56.37 11.74 .62**      .60**      .49**
##                             [.34, .80] [.30, .79] [.16, .72]
##
##
## Note. M and SD are used to represent mean and standard deviation, respectively.
## Values in square brackets indicate the 95% confidence interval.
## The confidence interval is a plausible range of population correlations
## that could have caused the sample correlation (Cumming, 2014).
##  * indicates p < .05. ** indicates p < .01.
##
```

Conducting Bivariate Linear Regression

```
# Easy coding method requires linear model
# Followed by `anova()` and `summary()` calls for description of model
# Output is less than ideal but coding is efficient
lm <- lm(rating ~ privileges, data = attitude)
anova(lm)
```

```
## Analysis of Variance Table
##
## Response: rating
##            Df Sum Sq Mean Sq F value  Pr(>F)
## privileges  1  780.2  780.22  6.2121 0.01888 *
## Residuals  28 3516.7  125.60
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm)
```

```
##
## Call:
## lm(formula = rating ~ privileges, data = attitude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.9357  -5.7397  -0.1691   5.6026  23.3582
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.1087     9.2661   4.544 9.63e-05 ***
## privileges    0.4239     0.1701   2.492   0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.21 on 28 degrees of freedom
## Multiple R-squared:  0.1816, Adjusted R-squared:  0.1523
## F-statistic: 6.212 on 1 and 28 DF,  p-value: 0.01888
```

```
# JMV coding method requires several argument calls as shown below
# Output is clean but coding gets more difficult with more predictors
linReg(attitude,
       dep = rating,
       covs = ("privileges"),
       blocks = list(
         list("privileges")),
       modelTest = TRUE,
       ci = TRUE,
       stdEst = TRUE,
       ciStdEst = TRUE)
```

```
##
##  LINEAR REGRESSION
##
##  Model Fit Measures
##  ─────────────────────────────────────────────────────────────────────────────
##    Model    R            R²           F           df1    df2    p
##  ─────────────────────────────────────────────────────────────────────────────
##        1    0.4261169    0.1815756    6.212078      1     28    0.0188770
##  ─────────────────────────────────────────────────────────────────────────────
##    Note. Models estimated using sample size of N=30
##
##
##  MODEL SPECIFIC RESULTS
##
##  MODEL 1
##
##  Model Coefficients - rating
##  ──────────────────────────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────
##    Predictor    Estimate     SE           Lower          Upper         t            p            Stand.
Estimate    Lower         Upper
##  ──────────────────────────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────
##    Intercept    42.1086576    9.2660624    23.12798913    61.0893261    4.544396    0.0000963
##    privileges     0.4239274    0.1700877     0.07551844     0.7723364    2.492404    0.0188770
0.4261169    0.07590847    0.7763253
##  ──────────────────────────────────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────
```

Creating Regression Table Document

```
# Create Word Document with APA style Regression Table

apa.reg.table(lm,
              # filename = "regression_table.doc", # Can uncomment for word doc
              table.number = 2)
```

```
##
##
## Table 2
##
## Regression results using rating as the criterion
##
##
##    Predictor       b       b_95%_CI beta  beta_95%_CI sr2 sr2_95%_CI    r
##  (Intercept) 42.11** [23.13, 61.09]
##   privileges   0.42*   [0.08, 0.77] 0.43 [0.08, 0.78] .18 [.00, .41] .43*
##
##
##
##              Fit
##
##
##       R2 = .182*
##  95% CI[.00,.41]
##
##
## Note. A significant b-weight indicates the beta-weight and semi-partial correlation are also significan
t.
## b represents unstandardized regression weights. beta indicates the standardized regression weights.
## sr2 represents the semi-partial correlation squared. r represents the zero-order correlation.
## Square brackets are used to enclose the lower and upper limits of a confidence interval.
## * indicates p < .05. ** indicates p < .01.
##
```

```
set.seed(13)
apa.reg.boot.table(lm,
                   # filename = "regression_boot.doc", # Can uncomment for word doc
                   table.number = 2,
                   number.samples = 5000)
```

```
##
##
## apa.reg.boot.table is a beta version.
## Block 1: Generating 5000 bootstrap samples
```

```
## Bootstrap for Delta RSQ in progress
```

```
##
##
## Table 2
##
## Regression results using rating as the criterion
##
##
##    Predictor       b       b_95%_CI beta  beta_95%_CI sr2 sr2_95%_CI    r
##  (Intercept) 42.11** [23.04, 61.45]
##   privileges   0.42*   [0.08, 0.78] 0.43 [0.09, 0.69] .18 [.01, .48] .43*
##
##
##
##              Fit
##
##
##       R2 = .182*
##  95% CI[.01,.48]
##
##
## Note. A significant b-weight indicates the beta-weight and semi-partial correlation are also significan
t.
## b represents unstandardized regression weights. beta indicates the standardized regression weights.
## sr2 represents the semi-partial correlation squared. r represents the zero-order correlation.
## Square brackets are used to enclose the lower and upper limits of a confidence interval.
## * indicates p < .05. ** indicates p < .01.
##
```