Various Versatile Variances
An Object-Oriented Implementation of Clustered Covariances in R

Susanne Berger, Nathaniel Graham, Achim Zeileis

https://R-Forge.R-project.org/projects/sandwich/
Motivation

Goals:
• Object-oriented clustered covariance implementation in R.
• Monte Carlo study to assess the performance of clustered standard errors beyond `lm()` and `glm()`.

Strategies:
• Account for clustered dependencies: Random effects, GEEs.
• Quasi-ML: Assume a correctly specified score function but a potentially misspecified remaining likelihood.

Special cases: QML with sandwich covariances.
• Cross-section data: Heteroscedasticity consistent (HC).
• Time series data: Heteroskedasticity and autocorrelation consistent (HAC).
• Clustered/Panel data: Panel Newey-West, Beck & Katz, ...
Motivation

R package for sandwich covariances:

• *sandwich* (Zeileis, 2004, 2006).
• Object-oriented implementation: betareg, clm, crch, hurdle, lm, mlm, mlogit, nls, polr, survreg, zeroinfl, ...
• Cross-section: `vcovHC(x, ...)` (default: HC3) and `sandwich(x, ...)` (default: HC0).
• Time series: `vcovHAC(x, ...)`. Convenience functions `kernHAC(x, ...)` (Andrews’ kernel HAC), `NeweyWest(x, ...)` (Newey-West-style HAC).

Problem: No clustered/panel covariance *up to now*...
Motivation

**R packages for clustered sandwich covariances:**

- `multiwayvcov` for `lm/glm`(-like) objects.
- `plm` for `plm` objects.
- `geepack` for `geeglm` objects.
- `lfe` for `felm` objects.
- among others (`clusterSEs`, `clubSandwich`, ...)

**Problem:** No object-oriented implementation *up to now*...
R implementation

**Building blocks:** Provided by R package *sandwich*.

- `sandwich(x, ...)` calculates an estimator of the sandwich
  \[ S(\theta) = B(\theta) \cdot M(\theta) \cdot B(\theta). \]

- `bread(x, ...)` returns the bread \( B(\theta) \), typically an empirical version of the inverse Hessian.

- `estfun(x, ...)` extracts empirical estimating functions from a fitted model object \( x \) (typically \( \partial \ell / \partial \theta \)).

- `meat(x, ...)`, `meatHC(x, ...)`, `meatHAC(x, ...)` return different “flavors” of meat \( M(\theta) \) – relying on the model just through `estfun(x)`.

**Needed:** New `meat*()` functions only based on `estfun()`. Full sandwich covariance `vcov*()` couples the `meat*()` with the `bread()` estimate.
R implementation

**Clustered data:** One-, two-, and multi-way clustering.

**Idea:** Aggregate `estfun` within each cluster prior to computing an HC-type `meat`.

**New R function:**

```r
covCL(x, cluster = NULL,
       type = NULL, cadjust = TRUE, multi0 = FALSE, ...)
```

- **cluster:** One/more cluster variables (cross-section if none).
- **type:** Type of bias correction: HC0–HC3.
- **cadjjust:** Cluster adjustment: $\frac{G}{G-1}$.
- **multi0:** Multi-way clustering: HC0 for final adjustment.
R implementation

**Panel data:** Driscoll & Kraay and Panel Newey & West.

**Idea:** Sum up `estfun` across clusters within each time period prior to computing an HAC-type meat.

**New R function:**

```r
vcovPL(x, cluster = NULL, order.by = NULL, 
   kernel = "Bartlett", lag = "NW1987", ...)
```

- `cluster`: Cluster/group/id variable (or list of cluster/time).
- `order.by`: Time variable.
- `kernel`: Kernel functions: "Bartlett", "truncated", "Tukey-Hanning", "quadratic spectral".
- `lag`: Lag length: "NW1987", "NW1994", "max" (or "P2009").
R implementation

**Panel data:** Panel-corrected covariances (Beck & Katz).

**Idea:** Decompose \( \text{estfun} \) into regressor matrix and working residuals, and employ outer product in each cluster.

**New R function:**

\[
\text{vcovPC}(x, \text{cluster} = \text{NULL}, \text{order.by} = \text{NULL}, \\
\quad \text{subsample} = \text{FALSE}, \ldots)
\]

- **cluster:** Cluster/group/id variable (or list of cluster/time).
- **order.by:** Time variable.
- **subsample:** Pairwise balanced sample or balanced subset.
Innovation data

**Motivation:** Applicability of clustered covariances to models beyond `lm()` or `glm()`.

**Data:** Aghion *et al.* (2013) analyze the effect of (the share of stock owned by) institutions on innovation (cite-weighted patents).

**Original analysis by Aghion et al. (2013):** Quasi-Poisson with clustered standard errors.

**Reanalysis by Berger et al. (2017):** Clustered covariances were provided for negative binomial hurdle models.
Innovation data

Data and model fit:

R> data(InstInnovation, package = "sandwich")
R> library("countreg")
R> m <- hurdle(
+   cites ~ institutions + log(capital/employment) + log(sales),
+   data = InstInnovation,
+   dist = "negbin", zero.dist = "negbin", separate = FALSE)

Customizing covariances: “Standard”, basic and clustered covariances.

R> library("sandwich")
R> vc <- list(
+   "standard" = vcov(m),
+   "basic" = sandwich(m),
+   "CL-1" = vcovCL(m, cluster = InstInnovation$company)
+ )
Innovation data

R> sapply(vc, function(x) sqrt(diag(x)))

<table>
<thead>
<tr>
<th></th>
<th>standard</th>
<th>basic</th>
<th>CL-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>count_(Intercept)</td>
<td>0.224</td>
<td>0.611</td>
<td>0.900</td>
</tr>
<tr>
<td>count_institutions</td>
<td>0.002</td>
<td>0.002</td>
<td>0.004</td>
</tr>
<tr>
<td>count_log(capital/employment)</td>
<td>0.054</td>
<td>0.082</td>
<td>0.152</td>
</tr>
<tr>
<td>count_log(sales)</td>
<td>0.013</td>
<td>0.032</td>
<td>0.050</td>
</tr>
<tr>
<td>zero_(Intercept)</td>
<td>0.577</td>
<td>0.929</td>
<td>1.518</td>
</tr>
<tr>
<td>zero_institutions</td>
<td>0.005</td>
<td>0.008</td>
<td>0.012</td>
</tr>
<tr>
<td>zero_log(capital/employment)</td>
<td>0.127</td>
<td>0.178</td>
<td>0.301</td>
</tr>
<tr>
<td>zero_log(sales)</td>
<td>0.053</td>
<td>0.098</td>
<td>0.158</td>
</tr>
</tbody>
</table>

For clustered data, standard errors can overstate estimator precision.

Compared to basic, CL-1 standard errors are scaled up by factors between 1.47 and 2.
Simulation

**Response:** $y_{ig} \sim F(\mu_{ig})$. $F =$ Normal/Gaussian, binomial (logit), Poisson, zero-truncated Poisson, beta, zero-inflated Poisson.

**Regression:** $h(\mu_{ig}) = 0 + 0.85 \cdot x_{ig}$.

**Correlations:**
- Cluster correlation $\rho$ for response $y_{ig}$ (via Gaussian copula).
- Regressor $x_{ig}$ has within cluster correlation 0.25.

**Setup:** 10,000 replications, 500 observations (5 per cluster).

**Outcome:** Empirical coverage of 95% confidence intervals for slope.
Simulation

- Gaussian
- Binomial (logit)
- Poisson

Coverage plot with different models:
- CL-0
- PL
- PC
- Standard
- Basic
- Random
- GEE
Simulation

\begin{figure}
\centering
\begin{tikzpicture}
\begin{axis}[
    title={CL–0 standard},
    xlabel={$\rho$},
    ylabel={coverage},
    xmin=0, xmax=0.9,
    ymin=0.85, ymax=1.0,
    xtick={0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9},
    ytick={0.85,0.90,0.95},
    legend pos=north west,
    legend cell align={left},
    legend style={draw=none},
    cycle list name=mark list
]
\addplot coordinates {
(0,1)(0.1,0.97)(0.2,0.94)(0.3,0.92)(0.4,0.90)(0.5,0.88)(0.6,0.86)(0.7,0.84)(0.8,0.82)(0.9,0.80)
};
\addplot coordinates {
(0,1)(0.1,0.98)(0.2,0.96)(0.3,0.94)(0.4,0.92)(0.5,0.90)(0.6,0.88)(0.7,0.86)(0.8,0.84)(0.9,0.82)
};
\addplot coordinates {
(0,1)(0.1,0.99)(0.2,0.97)(0.3,0.95)(0.4,0.93)(0.5,0.91)(0.6,0.89)(0.7,0.87)(0.8,0.85)(0.9,0.83)
};
\legend{basic, standard}
\end{axis}
\end{tikzpicture}
\caption{Simulation results for CL–0 standard with different coverage levels.

- Basic: "● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● 
\end{figure}
References

URL: https://R-Forge.R-project.org/projects/sandwich/

New manuscript: vignette("sandwich-CL", package="sandwich")


Simulation

Copula with AR(1) correlation structure.
Simulation

\[ F = \text{Gaussian with correlated, clustered, and uncorrelated regressors.} \]
Simulation

Increase number of clusters (observations per cluster constant).