PA

Proportionally Less Difficult?: Reevaluating Keele's "Proportionally Difficult"

Shawna K. Metzger

James Madison College, Michigan State University, East Lansing, USA. Email: shawna@shawnakmetzger.com

Abstract

Keele (2010, *Political Analysis* 18:189–205) emphasizes that the incumbent test for detecting proportional hazard (PH) violations in Cox duration models can be adversely affected by misspecified covariate functional form(s). In this note, I reevaluate Keele's evidence by running a full set of Monte Carlo simulations using the original article's illustrative data-generating processes (DGPs). I make use of the updated PH test calculation available in R's survival package starting with v3.0-10. Importantly, I find the updated PH test calculation performs better for Keele's DGPs, suggesting its scope conditions are distinct and worth further investigating. I also uncover some evidence for the traditional calculation suggesting it, too, may have additional scope conditions that could impact practitioners' interpretation of Keele (2010). On the whole, while we should always be attentive to model misspecification, my results suggest we should also become more attentive to how frequently the PH test's performance is affected in practice, and that the answer may depend on the calculation's implementation.

Keywords: Duration models, Proportional hazards assumption, Model misspecification

1 Introduction

The proportional hazards (PH) assumption is synonymous with the Cox model. This assumption states that any covariate's effect is unconditional on t, the duration. The Cox model incorporates covariates by permitting them to have a multiplicative (rather than additive) effect on the baseline hazard, $h_0(t)$, which represents the underlying rate at which the event occurs as a function of t when all the substantive covariates are set to 0. As a consequence of this functional form, an assumption about x's effect being unconditional on t in a Cox model setting implies the covariate's effect must be *proportional* across time: a one-unit increase in x will always enlarge or shrink the baseline hazard by the same scaling factor, regardless of t value. As with any regression model, treating x's effect as unconditional when it is not is a form of misspecification bias. Estimates will be inefficient at best and biased at worst (Keele 2010, 194). Furthermore, because the Cox model employs a non-linear transformation of the covariates and their estimates, this inefficiency and/or bias has the potential to affect *any* estimate, not just the violating covariate (Keele 2008, 6).

Given the ramifications of violating the PH assumption, scholars have developed ways to test for violations. The most widely used test in political science comes from Grambsch and Therneau (1994). This test involves Schoenfeld residuals, a special type of covariate-specific residual unique to the Cox model. In effect, Grambsch and Therneau's test checks for a correlation between the values of (a scaled version of) a covariate's Schoenfeld residual and some function of time, $g(t)$. If a non-zero correlation exists, we interpret it as suggestive evidence of a PH violation for the residuals' corresponding covariate. The test is formally articulated as a variant of a score test. Traditionally, the R, Python, and Stata routines for the Schoenfeld-based test (hereafter, "PH test") have used an approximation of Grambsch and Therneau's formal score test for speed reasons (Therneau and Grambsch 2000, 132 [formal], 134 [approximation]).[1]

1 The approximation makes a simplifying assumption about the formal score test's variance-covariance matrix (Therneau 2021, lines 36–45; Therneau and Grambsch 2000, 133–34). See Supplementary Appendix A for details.

Political Analysis (2023)
vol. 31: 156–163
DOI: 10.1017/pan.2022.13

Published
20 June 2022

Corresponding author
Shawna K. Metzger

Edited by
Jeff Gill

Over the past decade or so, our awareness of the Schoenfeld-based test's properties has grown. In particular, Keele (2010) reemphasizes one of Therneau and Grambsch's points (2000, Sec. 6.6): the PH test checks for any evidence of nonproportionality, and time-varying covariate effects are not the only way nonproportionality can arise. The implication is the PH test can return false positives, in the sense of researchers incorrectly concluding a covariate's effect is time varying, in the presence of omitted relevant covariates (e.g., interactive effects) or misspecified covariate functional forms. Keele argues this point is underappreciated, and provides two simulated datasets to illustrate how severely the PH test's performance can be impacted in the presence of such model misspecification. His simulated datasets and applied examples give the overall impression that these false positives happen with some degree of regularity.

However, in the time since Keele's (2010) writing, how some programs compute the PH test has changed. Specifically, R's PH test routine no longer uses a quick approximation of Grambsch and Therneau's test, but calculates the actual test, in full, starting with survival 3.0-10 (8/2019[2]). Metzger and Jones' (2019) supplemental simulation results suggest there are performance differences between the approximation and the actual test calculations, in some contexts. It is possible, then, that these performance differences may extend to other PH-specification-related issues, including those highlighted by Keele.

This note evaluates the PH test's performance in practice by running proper Monte Carlo simulations using the new PH test routines on Keele's original data-generating processes (DGPs). I do find that the new routines outperform the old routines in a greater number of situations, suggesting they may be less sensitive to misspecified covariate functional forms or omitted covariates. However, for some small tweaks to Keele's original DGPs, I also find no evidence of the ubiquitous poor PH test performance generally associated with Keele (2010), for *both* the old and new test routines—a qualifier still relevant for Stata and Python users.

The major implications are twofold. First, my results imply the PH test, in general, may be less sensitive to returning false positives in the presence of model misspecification than some practitioners might think after reading Keele (2010). Second, they imply that we should more thoroughly investigate the extent to which the test's analytic, misspecification-related properties may affect PH-related inference in practice, while being mindful that these conclusions may be calculation specific. I proceed by describing the setup for the Monte Carlos, discussing the simulation results, and then conclude with some summarizing remarks.

2 Simulation Setup

I replicate Keele's two examples from his Section 2.1 as closely as possible (Table 1).[3,4] The true DGPs are exponential hazard functions with two uncorrelated covariates. One of the covariates is continuous; the other is binary. The first example's DGP involves a quadratic term for the continuous covariate, while the other example has an interaction between the binary and continuous covariate. Both of Keele's examples use the binary covariate as the PH violator, with $\ln(t)$ serving as time's functional form for the violation.[5]

I use the simsurv package to generate the simulated duration data (Brilleman *et al.* 2021), allowing me to exploit its routines for permitting covariates with time-varying effects. I run 10,000

2 First affected CRAN release: survival 3.1-6 (11/2019).
3 See Metzger (2022a, b) for replication materials.
4 I cannot replicate Keele's original simulation results if I run his original file (Supplementary Appendix D). Additionally, simsurv had trouble generating data with Keele's x_1-related parameter values. These problems disappeared when I reduced the magnitude of those parameters. Later, I investigate other ways related to x_1 of reducing the hazard's value (Section 3.3).
5 As an extra check, I vary how I induce right censoring (RC). My conclusions are unaffected. I report the most conservative results in text (*rc*%) and report the rest in Supplementary Appendix B.
6 The article states 25% RC, but the code in Supplementary Appendix D shows the two illustrative datasets have 0% RC, in truth. I reran the scenarios with 0% RC; none of the substantive conclusions change (Supplementary Appendix B).

Table 1. Simulated data: setup details.

	Keele (2010)	Metzger Rerun
n	100	100
$h_0(t)$		
Funct. Form.	exponential	exponential
Scale	0.15	0.15
Covariates		
x_1	Uniform integers, [22,90]	Uniform integers, [22,90]
x_2	Binomial with $p = 0.5$	Binomial with $p = 0.5$
True linear combo		
Sc. 1: Quadratic	$0.1\,x_1^2 + 1\,x_2 \ln(t)$	$0.001\,x_1^2 + 1\,x_2 \ln(t)$[4]
Sc. 2: Interactive	$0.1\,x_1 + 1\,x_2 \ln(t) + 0.4\,x_1 x_2$	$0.001\,x_1 + 1\,x_2 \ln(t) + 0.004\,x_1 x_2$
Right censoring		
% RC	25%[6]	25%
RC Type	(see Footnote 6)	{Random, Largest rc%}[5]

simulations per DGP because of my interest in the PH test's p-values. The reported results use successfully converged draws only, noted at the bottom of each graph.

3 Simulation Results

I report my simulation results as a series of binned scatterplots (Figure 1) for every scenario–PH calculation combination. Each individual bin contains up to 10,000 points, jittered horizontally for visibility, representing a covariate's PH test p-values from each simulation draw, of which there are 10,000, at most.[7] The vertical axis represents the PH test's p-value. The horizontal dashed lines denote the bin's mean p-value; the thinner horizontal solid lines, its 2.5th/97.5th percentiles. The dashed lines are comparable to Keele's output (2010, Table 1), which reports each covariate's PH test p-value from a single simulation draw.

Each covariate has a pair of bins, corresponding to the two model specifications I check. The specifications represent the first model we would estimate in an analysis—i.e., the model we would use to diagnose potential PH violations. Accordingly, these models do not include any PH-violation corrections. The pair's first bin (light gray bar) represents the incorrect base specification in which x_1's functional form is misspecified. Depending on the scenario, the base specification lacks either an x_1^2 term (Sc. 1) or an $x_1 x_2$ interaction term (Sc. 2). The second bin (dark gray bar) represents the correct base specification—it includes the appropriate additional x_1 term for the scenario in question. The final pair of bins in each scatterplot corresponds to the joint hypothesis test for *any* PH violations in the model (global test). Recall that PH test p-values below 0.05 are suggestive of a PH violation.

If Keele's findings from the approximation hold for the actual test, we should make erroneous conclusions about whether a covariate violates PH for the wrong base specification, but correct conclusions for the correct base specification (Table 2).

3.1. Scenario 1

Figure 1 provides information about Scenario 1 for the approximated PH test (panel (a))[8] and the actual PH test (panel (b)). Starting with the approximation (Figure 1a), the

7 I run all four standard $g(t)$ forms for the PH test, but all return fairly similar results. I discuss $g(t) = \ln(t)$ here, to match the DGP's true $g(t)$. I report the full output with all four forms in Supplementary Appendix B.

8 I also ran the simulations using survival 2.18, the version available when many of Keele's replication files are dated. They are identical to the reported approximation results.

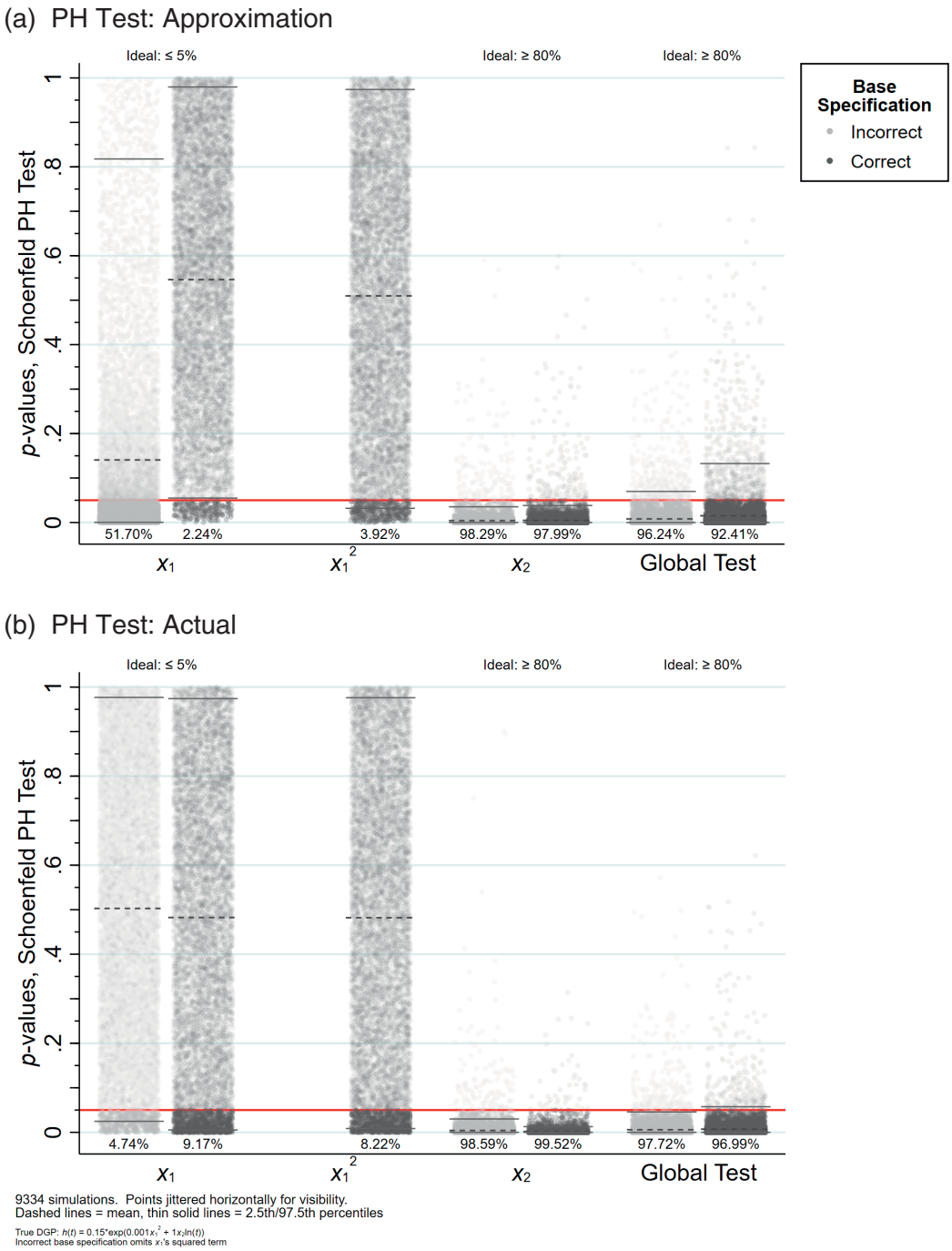

Figure 1. Scenario 1 simulations.

Table 2. Expectations based on Keele (2010).

	Wrong base specification		Correct base specification	
	Non-PH violators	PH violators	Non-PH violators	PH violators
	WRONG CONCLUSION	CORRECT CONCLUSION	CORRECT CONCLUSION	CORRECT CONCLUSION
Mean *p*-value	≤0.05	≤0.05	>0.05	≤0.05
% *p*-values <0.05	More than 5%	At least 80%	5% or lower	At least 80%

results are somewhat consistent with Keele's original findings. First, *counter* to Keele's illustrative example, the PH test never returns an on-average false positive for non-violator x_1 (average $\not\leq 0.05$; first set of bars, dashed line), regardless of specification (Keele's Table 1, top portion). However, x_1's false positive rate under the misspecified base model does surpass the usual 5% threshold (light gray bar; 51.7%)—we detect violations far more frequently than we should—consistent with the implications of Keele's results.

Once the model's base specification is correct (dark gray bar), the PH test's performance for x_1 improves drastically. x_1's average p-value is 41 percentage points higher (0.14 vs. 0.55), meaning fewer simulation draws return incorrect evidence of x_1 being a PH violator. Additionally, x_1's false positive rate drops below 5% (2.24%). The PH test's poor performance for non-PH violators in the misspecified base model, but vastly improved performance in the correctly specified base model, matches Keele's original findings.

Second, the PH test is more than adequately powered for PH violators.[9] We detect x_2's PH violation (third set of bars) ~98% of the time, exceeding our 80% rule of thumb. Notably, this is true regardless of base model specification. As before, these results comport with Keele's Table 1, where he reports the PH test has no issues detecting violations for true PH-violating covariates, regardless of specification.

However, these patterns change once we use the updated PH test procedure to compute Grambsch and Therneau's actual score test (Figure 1b), in a manner inconsistent with Keele. Specifically, the actual PH test's performance for x_1 differs dramatically.[10] Like before, the PH test never returns an on-average false positive for x_1. The difference between the incorrect vs. correct base specification's average p-value, though, is considerably smaller (0.50 [incorrect] vs. 0.48 [correct]), suggesting that misspecification has little to do with the PH test's performance for Sc. 1—a very different conclusion than Keele's original findings. This statement is further supported by x_1's false positive rate. The misspecified base model's rate is 4.7%, but the correctly specified base model's false positive rate is *higher* (9.2%), not lower.

3.2. Scenario 2

Figure 2 displays Scenario 2's simulation results, where the model misspecification is in the form of an omitted x_1x_2 interaction. Similar to Scenario 1, Scenario 2's results are starkly different than we would expect, based on Keele's illustrative example.

For the approximated PH test (Figure 2a), none of Keele's results replicate:

- The PH test's performance for non-PH-violating x_1 is surprisingly unaffected by misspecification, counter to Keele's findings. The average p-value is well above 0.05, regardless of the model's base specification. Additionally, the PH test's statistical size is >5%, but like the Scenario 1 actual PH test results, the size is *worse* for the correctly specified base model, not better (5.7% [incorrect] vs. 13.9% [correct]), suggesting misspecification is not at fault.
- The PH test's performance for PH-violator x_2 *is* affected by misspecification, also counter to Keele's findings. Exacerbating matters, the *correctly* specified base model performs far worse than the misspecified base model, with a far higher average p-value (0.34 [correct] vs. 0.02 [incorrect]) and a very underpowered test that falls well short of the 80% rule of thumb (21.3% [correct] vs. 90.1% [incorrect]).

The updated PH test procedure suffers from none of these issues (Figure 2b). We draw the correct conclusions about x_1, by and large, regardless of base model specification. The broad patterns for x_1's actual PH test average p-value and statistical size are identical to those from Scenario 1's actual PH test results. In addition, we draw the correct conclusion about x_2. Regardless of the base specification, x_2's average p-value is always below 0.05 and is always well powered.

9 I refer to "the PH test's power" as a shorthand, but the type of statistical test does not affect statistical power.
10 Its performance for x_2 is similar to the traditional approximation.

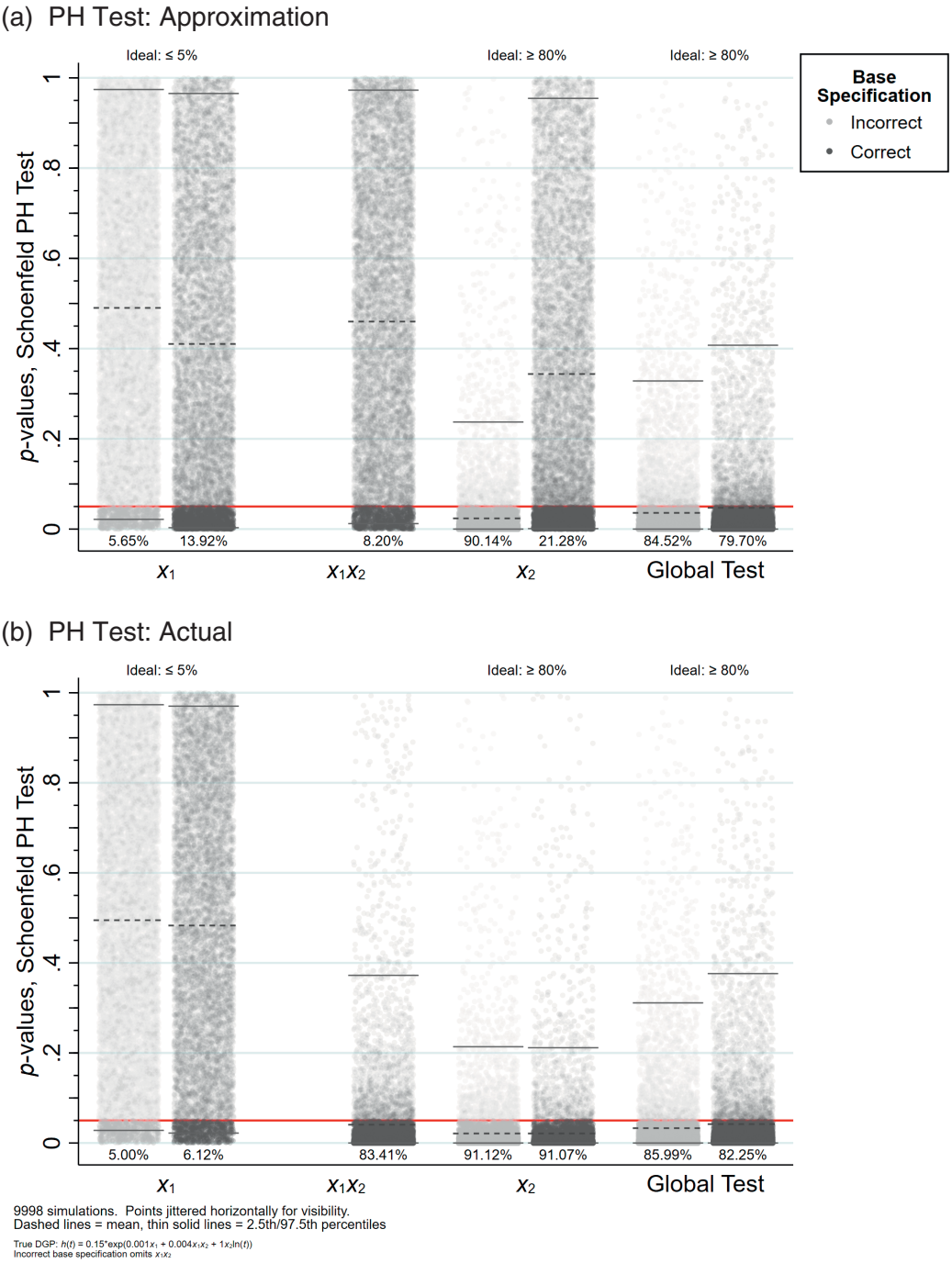

Figure 2. Scenario 2 simulations.

3.3. Encore: Original Parameter Values, Tweaked x_1 Distribution

I also investigated what occurs if I use Keele's original parameter values, but alter the range of x_1's values ($x_1 \sim \mathcal{U}[0,1]$) (fn. 4). The actual PH test continues to perform just as well as before, further supporting my previous simulation results.[11] However, the approximated PH test results are surprising and worth mentioning (Supplementary Appendix C, Figures 3a/4a).

For both scenarios, the Supplementary Appendix figures make clear that situations exist in which *the approximation has no performance issues*, even in the face of model misspecification. The simulated DGPs for Figures 3 and 4 are only slight modifications of Keele's originals,

11 Results reported in Supplementary Appendix C.I.

making the results all the more notable. The major implication is that model misspecification's deleterious effects on the PH test's performance may not be as widespread as Keele's simulations unintentionally suggest, but instead, may have scope conditions. Therneau and Grambsch hint at some possibilities (2000, Sec. 6.6). Another is the proportion of subjects failing in $t \in (0,1]$ (Supplementary Appendix C.II), and another still is the degree of correlation among the covariates. Probing any of these scope conditions more deeply is left to future research. The conditions would only affect practitioners using PH test routines that approximate Grambsch and Therneau's original test—Stata, lifelines (Python), or <survival 3.0-10 (R). All of my simulation results show that routines calculating the actual PH test (≥survival 3.0-10) are unaffected by model misspecification, at best, or more likely, governed by a different set of scope conditions that future research would need to investigate, at worst.

4 Conclusion

In this note, I used Monte Carlo simulations to reassess Keele's (2010) illustrative examples regarding the Schoenfeld-based test for PH violations in Cox duration models and its propensity to return false positives in the presence of omitted relevant covariates or misspecified covariate functional forms. I primarily use survival::cox.zph's recent rewrite, which calculates Grambsch and Therneau's actual Schoenfeld-based PH test instead of an approximation of it.

If the actual and approximated calculations perform similarly, Keele's discussion suggests we should (a) erroneously conclude that non-PH-violating covariates are violators, and (b) correctly conclude that PH-violating covariates are violators. My simulation results, which use the same DGPs as Keele, challenge both points. Using the updated PH test calculation, the results are strongly *not* supportive of (a) and only sometimes supportive of (b). I also run the simulations using the PH test's traditional approximation-based calculation, where I find evidence sometimes weakly consistent with Keele's, but sometimes not at all consistent, as discussed at the end of the previous section.

All and all, there are three key takeaways. First, practitioners should potentially consider using the updated PH test procedure if possible, as it is less sensitive than the traditional procedure to the forms of model misspecification and DGPs I check here. However, we know less about the updated PH test's performance, on the whole. Future work should use a more extensive set of DGPs to investigate the extent to which the updated test's performance advantage generalizes, allowing practitioners to make better-informed decisions about which procedure to use.

Second, Keele (2010) may have given us a false sense of how frequently the PH test returns false positives in the presence of model misspecification. Unintentionally, his illustrative simulations and three applications create the impression that false positives occur regularly. My full simulations using Keele's DGPs cast doubt on this point.

Finally, we know model misspecification *can* affect the PH test's performance, as Keele (2010) correctly notes. However, we need more information about whether and when it *does* affect the PH test's performance, in practice. My simulations also suggest these conditions may differ, depending on the calculation. These scope conditions merit further, future investigation.

Acknowledgments

I thank Janet Box-Steffensmeier, Justin Esarey, and Benjamin Jones for feedback on earlier drafts. Thanks also to Luke Keele for sharing additional replication files from his original analysis. Any mistakes are mine alone.

Data Availability

All analyses run using Stata 17.0 MP6 or R 4.1.0. This article's replication code has been published through Code Ocean and can be viewed interactively at https://doi.org/10.24433/CO.9885736.v1

(Metzger 2022a). A preservation copy of the same code and data can also be accessed via Dataverse at https://doi.org/10.7910/DVN/ZYOJF6 (Metzger 2022b).

Supplementary Materials

To view supplementary material for this article, please visit http://doi.org/10.1017/pan.2022.13.

Works Cited

Brilleman, S. L., R. Wolfe, M. Moreno-Betancur, and M. J. Crowther. 2021. "Simulating Survival Data Using the simsurv R Package." *Journal of Statistical Software* 97 (1): 1–27.

Grambsch, P. M., and T. M. Therneau. 1994. "Proportional Hazards Tests and Diagnostics Based on Weighted Residuals." *Biometrika* 81 (3): 515–526.

Keele, L. 2008. *Semiparametric Regression for the Social Sciences*. New York: Wiley.

Keele, L. 2010. "Proportionally Difficult: Testing for Nonproportional Hazards in Cox Models." *Political Analysis* 18 (2): 189–205.

Metzger, S. K. 2022a. "Replication Data for: Proportionally Less Difficult?: Reevaluating Keele's 'Proportionally Difficult'." Code Ocean. https://doi.org/10.24433/CO.9885736.v1.

Metzger, S. K. 2022b. "Replication Data for: Proportionally Less Difficult?: Reevaluating Keele's 'Proportionally Difficult'." Harvard Dataverse, V1. https://doi.org/10.7910/DVN/ZYOJF6.

Metzger, S. K., and B. T. Jones. 2019. "Be Kind, Please PH-Stratify: Stratified Hazards and Proportional Hazard Testing." Working Paper.

Therneau, T. M. 2021. "cox.zph: zph.rnw Documentation." https://github.com/therneau/survival/blob/f2567b77252ac7935eba0ead364665c654ef28d3/noweb/zph.Rnw.

Therneau, T. M., and P. M. Grambsch. 2000. *Modeling Survival Data: Extending the Cox Model*. New York: Springer.