Journal of Statistical Software

August 2020, Volume 94, Issue 12. doi: 10.18637/jss.v094.i12

PResiduals: An R Package for Residual Analysis Using Probability-Scale Residuals

Qi Liu
Merck & CO., Inc.

Bryan Shepherd
Vanderbilt University

Chun Li
University of Southern California

Abstract

We present the R package **PResiduals** for residual analysis using the probability-scale residual. This residual is well defined for a wide variety of outcome types and models, including some settings where other popular residuals are not applicable. It can be used for model diagnostics, tests of conditional associations, and covariate-adjustment for Spearman's rank correlation. These tests and measures of conditional association are applicable to any orderable variable. They use order information but do not require assigning scores to ordered categorical variables or transforming continuous variables, and therefore, can achieve a good balance between robustness and efficiency. We illustrate the usage of the **PResiduals** package with a publicly available dataset.

Keywords: residual, diagnostics, correlation, association, covariate-adjustment, rank statistics.

1. Introduction

We recently proposed a new type of residual, the probability-scale residual (PSR), defined as $\mathsf{P}(Y^* < y) - \mathsf{P}(Y^* > y)$, where y is the observed value and Y^* is a random variable from the fitted distribution (Li and Shepherd 2012; Shepherd, Li, and Liu 2016). This residual is on the probability scale ranging from -1 to 1. It is well defined for a wide variety of outcome types and models, including some settings where other popular residuals are not applicable. Under properly-specified models, the PSR has expectation 0, and it can, therefore, be used for model diagnostics. In addition, PSRs can be used to test for conditional associations (Li and Shepherd 2010) and to construct covariate-adjusted Spearman's rank correlation (Liu, Li, Wanga, and Shepherd 2018). These methods are applicable to any orderable variable. They use order information but do not require assigning scores to ordered categorical variables or transforming continuous outcomes, and therefore, can achieve a good balance between

robustness and efficiency. The R (R Core Team 2020) package, **PResiduals** (Dupont, Horner, Li, Liu, and Shepherd 2020), has been developed to facilitate residual analyses using PSRs. **PResiduals** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=PResiduals`. The purpose of this paper is to provide an introduction to the **PResiduals** package. We organize this paper as follows. In Section 2, we provide a brief review of PSRs and related methods. In Section 3, we illustrate the main functions in **PResiduals** with examples. Section 4 contains a summary.

2. Review of methods

2.1. PSRs

A residual can be viewed as a contrast between the observed value and its fitted distribution. For example, the commonly used observed-minus-expected residual (OMER) can be written as $y - \hat{y} = \mathsf{E}(y - Y^*)$, where y is the observed value, Y^* is a random variable from the fitted distribution F^*, and the contrast function is the difference. The PSR can be written similarly with a more general contrast function $\text{sign}(y, Y^*)$, where $\text{sign}(a, b)$ is -1, 0, and 1 for $a < b$, $a = b$, and $a > b$. Specifically, $r(y, F^*) = \mathsf{E}[\text{sign}(y, Y^*)] = \mathsf{P}(Y^* < y) - \mathsf{P}(Y^* > y) = F^*(y-) + F^*(y) - 1$, where $F^*(y-) = \lim_{t \uparrow y} F^*(t)$. The PSR was originally proposed for ordered categorical variables where the difference between categories is not well defined (Li and Shepherd 2010, 2012). Later, it was extended to other types of orderable variables, including continuous, discrete, and censored outcomes (Shepherd *et al.* 2016).

With continuous outcomes, the PSR is $2F^*(y) - 1$. If the model is properly specified, as $n \to +\infty$, $F^* \to F$ where F is the true distribution of Y, then $r(Y, F^*) \to 2F(Y) - 1$. Note that $F(Y)$ is the probability integral transformation and it is uniformly distributed from 0 to 1. Therefore, if the PSR is from the properly-specified model, it is a re-scaling of the probability integral transformation and it is approximately uniformly distributed from -1 to 1 with expectation 0 and constant variance 1/3. A quantile-quantile (QQ) plot of PSRs versus the theoretical quantiles of the uniform distribution can be used to assess the overall model fit. In addition, PSRs can also be used in residual-by-predictor plots to detect lack of fit for specific predictors.

With discrete outcomes, the PSR is $2F^*(y) - f^*(y) - 1$, where f^* is the probability mass function of the fitted distribution. In the extreme case where Y is binary, the PSR reduces to $y - \mathsf{P}(Y^* = 1)$, which is the OMER or unscaled Pearson residual. Although the PSR still has expectation 0 under the properly-specified model, it is not uniformly distributed due to the discreteness. Therefore, residual-by-predictor plots still provide information for the fit of specific predictors, but QQ-plots with PSRs are generally not useful.

With right censored outcomes, we denote T as the time to event and C as the time to censoring. Rather than directly observing T we only observe $Y = \min(T, C)$ and $\Delta = I(T \leq C)$. The above formula for the PSR can only be applied to non-censored observations. If censored, the failure time is unknown but it occurs after the censoring time Y. Therefore, we define the PSR as its conditional expectation given that $T > Y$, i.e., $\mathsf{E}[r(T^*, F^*) \mid T^* > Y] = F^*(Y)$. Formally, the PSR for censored outcomes is defined in terms of y and δ, the observed values of Y and Δ, as $r(y, F^*, \delta) = F^*(y) - \delta[1 - F^*(y-)]$. Note that with this definition, the PSR for censored observations is always non-negative. But under the properly-specified

model and $T \perp C$, it still has expectation 0. Therefore, the PSR can be used for model diagnostics for censored outcomes.

The PSR has many attributes that make it useful in practice. Since the PSR does not require calculation of a fitted mean or full specification of a fitted distribution, it is especially useful for models where expectations cannot be computed or that are not fully parametric. For example, it is the natural residual for ordered categorical outcomes and cumulative probability models, where other commonly used residuals are not available or cannot be easily derived. As illustrated below, there are also benefits to having a residual that is well defined with a common scale for a wide variety of outcome types and models. In addition to those already mentioned, the PSR has some connections with other residuals. For example, the quantile residuals (Dunn and Smyth 1996) of continuous outcomes can be viewed as a normalized version of PSRs, and normalized PSRs for censored outcomes extend the concept of quantile residuals to time-to-event data (Shepherd *et al.* 2016). Residuals have been developed for discrete data that jitter the outcome, thereby making residuals behave more like those for continuous data (Dunn and Smyth 1996); a jittered residual for ordinal regression models based on an implied latent variable distribution was recently proposed and compared to the probability-scale residual, and shown to perform favorably for some diagnostics (Liu and Zhang 2018). For a more detailed discussion of comparisons and connections between PSRs and other commonly used residuals, such as observed-minus-expected residuals (OMER) for continuous outcomes, Pearson and deviance residuals for discrete outcomes, and martingale, Cox-Snell, and deviance residuals for censored outcomes, we refer readers to our earlier paper (Shepherd *et al.* 2016).

2.2. Test of residual correlation with PSRs

The PSR was initially independently proposed as a component of test statistics involving ordinal variables (Wang, Ye, and Zhang 2006; Li and Shepherd 2010). Specifically, the PSR was used for testing the conditional association between two ordered categorical variables X and Y while adjusting for covariates Z, referred to as COBOT (conditional ordinal by ordinal tests) in Li and Shepherd (2010). Traditional regression approaches treat the ordinal predictor as either categorical or numerical, whereas the former ignores the order information and the latter often makes linear assumptions. The basic idea of COBOT is to obtain conditional distributions of X and Y from models of X on Z and of Y on Z, and then to determine whether these conditional distributions are independent.

Three test statistics were proposed based on this idea. The first test statistic (T1) compares the observed joint distribution between X and Y with its expected distribution under the null of conditional independence. If X and Y are independent conditional on Z, their joint distribution given Z is expected to follow the product of the conditional distributions of X and Y given Z. Therefore, we test for conditional independence by computing Goodman and Kruskal's gamma for the observed and expected joint distributions and taking their difference. The second test statistic (T2) is based on PSRs. Specifically, T2 takes PSRs from models of X on Z and of Y on Z and tests the null of no residual correlation. The third test statistic (T3) evaluates the concordance-discordance of data drawn from the joint fitted distribution of X and Y under conditional independence with those drawn from the empirical joint distributions, which can be written as the covariance of PSRs. p values for all three test statistics are computed based on large sample theory using M-estimation procedures. More details of these test statistics are given in Li and Shepherd (2010).

2.3. Covariate-adjusted Spearman's rank correlation with PSRs

When there are no covariates, the PSR is a linear transformation of ranks, and the correlation of PSRs is simply Spearman's rank correlation (Li and Shepherd 2012; Shepherd *et al.* 2016). Formally, the population parameter of Spearman's rank correlation can be expressed as the correlation of PSRs. With covariates, the PSR can be viewed as a linear transformation of adjusted ranks. This motivates us to use PSRs to construct covariate-adjusted rank correlations (Liu *et al.* 2018).

There are generally two types of covariate-adjusted correlations. One is partial correlation, i.e., removing the effect of covariates and summarizing the relationship with a single number. The other is conditional correlation, i.e., assessing the correlation at specific levels of the covariates. We have proposed estimators for both partial and conditional Spearman's rank correlations: our partial estimator is the correlation of PSRs and our conditional estimator is the conditional correlation of PSRs (Liu *et al.* 2018).

To obtain those estimators, we first need to fit models of X on Z and of Y on Z, and then compute PSRs from both models. Although the PSR is well defined and can be easily computed from many parametric or nonparametric models, to maintain the spirit of Spearman's rank correlation and to achieve a good balance between robustness and efficiency we favor fitting rank-based semiparametric models of X on Z and Y on Z. Specifically, we advocate fitting cumulative probability models. This class of models was originally developed for discrete ordinal data (McCullagh 1980; Agresti 2010), but can be applied to continuous data (Sall 1991; Harrell 2015; Liu, Shepherd, Li, and Harrell 2017). Since the model fit only uses the order information of X and Y, using PSRs from this type of model preserves the rank-based nature of Spearman's rank correlation.

After obtaining PSRs from models of X on Z and of Y on Z, our partial Spearman's rank correlation estimator can be obtained simply as the correlation of PSRs. Note that this procedure is analogous to the partial Pearson's correlation, which is computed as the correlation of OMERs from linear regression models. M-estimation techniques can be used to obtain its standard error. Since the correlation coefficient is bounded between -1 and 1, Fisher's transformation can be used to obtain better coverage to its confidence interval. Technical details are found in Liu *et al.* (2018).

To obtain the conditional estimator for Spearman's rank correlation, we need to model the conditional correlation between PSRs. If Z is a categorical variable with sufficient numbers in each category, we can do a stratified analysis, i.e., compute the correlation of PSRs within each level of Z. If Z is continuous, smoothing is needed and can be achieved nonparametrically or parametrically. We have described a nonparametric approach based on kernel weighting and a parametric approach using linear regression in Liu *et al.* (2018).

3. Analysis with the PResiduals package

3.1. Wage data

Throughout this section, we use a publicly available dataset, the `Wage` data, to illustrate the usage of key functions in the **PResiduals** package. This dataset can be obtained from the R package **ISLR** (James, Witten, Hastie, and Tibshirani 2017). It contains annual wage (in

Variable	Description
`year`	Year that wage information was recorded
`age`	Age of worker
`maritl`	A factor with levels 1. Never Married 2. Married 3. Widowed 4. Divorced and 5. Separated indicating marital status
`race`	A factor with levels 1. White 2. Black 3. Asian and 4. Other
`education`	A factor with levels 1. < HS Grad 2. HS Grad 3. Some College 4. College Grad and 5. Advanced Degree indicating education level
`region`	Region of the country (mid-atlantic only)
`jobclass`	A factor with levels 1. Industrial and 2. Information indicating type of job
`health`	A factor with levels 1. $\leq$ Good and 2. $\geq$ Very Good indicating health level of worker
`health_ins`	A factor with levels 1. Yes and 2. No indicating whether worker has health insurance
`logwage`	Log of worker's wage
`wage`	Worker's raw wage

Table 1: Variables in `Wage` dataset.

thousands of dollars) and other information for 3,000 male workers in the mid-Atlantic region of the United States from 2003 to 2009. Table 1 summarizes the description of the dataset and its variables. With this dataset, we can build regression models for wage and study its relationship with other variables.

```
R> data("Wage", package = "ISLR")
```

3.2. Calculation of PSRs

We first illustrate how to obtain PSRs from various models. The function `presid()` is implemented to compute PSRs. Its usage is very similar to the function `residuals()` from the **stats** package (R Core Team 2020). Specifically, it takes a model object and returns a numerical vector containing PSRs in the order of original observations in the dataset. Currently supported model objects include those returned by `lm` and `glm` (Poisson, binomial, and gaussian families) in the **stats** package (R Core Team 2020); `polr` and `glm.nb` in the **MASS** package (Venables and Ripley 2002; Ripley 2020); `ols`, `Glm`, `lrm`, `orm`, `psm`, and `cph` in the **rms** package (Harrell Jr 2020); and `survreg` (Weibull, exponential, gaussian, logistic, and lognormal distributions) and `coxph` in the **survival** package (Therneau and Grambsch 2000; Therneau 2020). Hence, using the function `presid()`, we can easily obtain PSRs from proportional odds models (more generally cumulative probability models), linear regression models, generalized linear regression models (such as Poisson and negative binomial models), parametric survival models, and Cox proportional hazards models. We now illustrate the calculation of PSRs from some of these models and their application in model diagnostics with the `Wage` data.

We start with ordinal regression models for ordered categorical variables. The PSR is a natural residual for the ordered categorical outcome as it does not require assigning distance scores to categories (Li and Shepherd 2012). Specifically, we model the ordered categorical

variable `education`, which has 5 levels, with a proportional odds model. We include `age`, `race`, `jobclass`, `maritl` (marital status), `health` (health status), and `year` (calender year) as covariates; in addition, a transformation of `age` using restricted cubic splines was considered to account for a potential nonlinear relationship. Note that we use the `rcs()` function from the **rms** package to perform the restricted cubic splines transformation throughout the paper, but other alternative smoothing functions such as `ns()` and `bs()` from the **splines** package (R Core Team 2020) could perform similarly (examples not shown). PSRs can be obtained as functions of regression coefficients directly. In R, proportional odds models can be fitted using the function `polr()` from the **MASS** package or the function `orm()` from the **rms** package. The following chunk of code illustrates the usage of these two functions along with `presid()`. When using `orm()`, we need to set the arguments `x = TRUE` and `y = TRUE` so that the expanded design matrix and the values of the response variable are returned; this is a convention of the **rms** package. In this specific example, the PSRs obtained using these two functions are slightly different at the sixth digit after the decimal. This is because `polr()` and `orm()` use different fitting procedures and yield slightly different regression coefficients.

```
R> library("PResiduals")
R> library("MASS")
R> library("rms")
R> po.polr <- polr(education ~ rcs(age, 5) + race + jobclass + maritl +
+    health + year, data = Wage)
R> PSR.po.polr <- presid(po.polr)
R> po.orm <- orm(education ~ rcs(age, 5) + race + jobclass + maritl +
+    health + year, data = Wage, x = TRUE, y = TRUE)
R> PSR.po.orm <- presid(po.orm)
R> summary(cbind(PSR.po.polr, PSR.po.orm))

  PSR.po.polr           PSR.po.orm
 Min.   :-0.9882886   Min.   :-0.988289
 1st Qu.:-0.4510896   1st Qu.:-0.451093
 Median :-0.0072629   Median :-0.007264
 Mean   : 0.0000001   Mean   : 0.000000
 3rd Qu.: 0.5012186   3rd Qu.: 0.501223
 Max.   : 0.9716579   Max.   : 0.971659

R> max(abs(PSR.po.polr - PSR.po.orm))

[1] 1.33206e-05
```

Figure 1 shows the application of PSRs in residual-by-predictor plots. Specifically, in the left panel of Figure 1, we include both linear and nonlinear terms by transforming `age` using restricted cubic splines with 5 knots, whereas in the right panel, we only include the linear term. The smoothed curve shows a nonlinear relationship between PSRs and `age` when only including the linear term, suggesting a better fit when both linear and nonlinear terms are included. This is also supported by smaller AIC and BIC for the model that includes both linear and nonlinear terms. Next, we consider linear regression models (specifically, normal linear regression models and least squares models). For normal linear regression models,

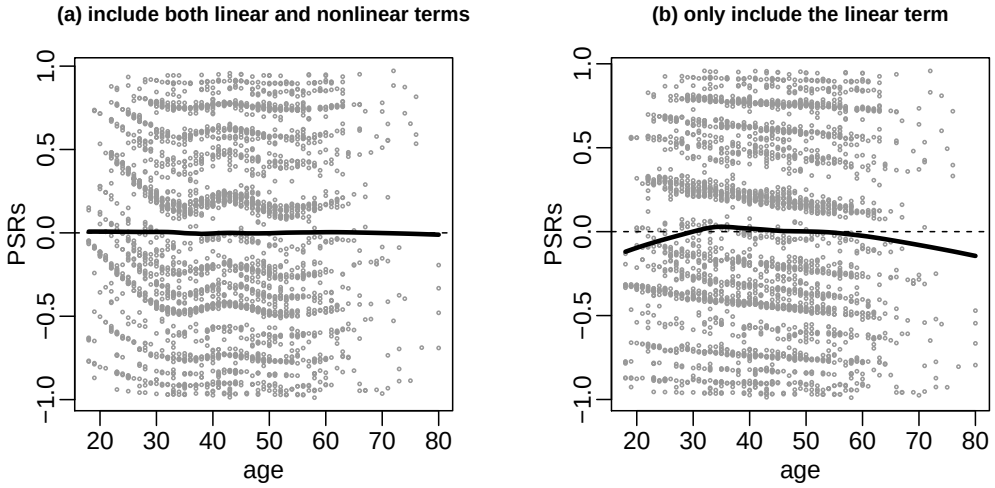

Figure 1: Residual-by-predictor plots with PSRs from proportional odds models. (a): PSRs are from the model including both linear and nonlinear terms. (b): PSRs are from the model only including the linear term.

PSRs can be obtained by assuming normality for the error distribution. For example, the PSR for observed value y_i can be computed as $2\Phi[(y_i - \hat{y}_i)/\hat{\sigma}] - 1$, where $\hat{y}_i$ is the fitted value, $\hat{\sigma}$ is the standard deviation of the observed-minus-expected residuals (OMERs), and $\Phi()$ is the cumulative distribution function (CDF) of the standard normal distribution. This is the default in `presid()` for linear model objects (returned by `lm`, `ols` and `Glm`). But the normality assumption may not be necessary since it is well known that least squares models are fairly robust to nonnormal errors as long as they are not highly skewed. In some application, we may be willing to only assume homoscedasticity instead of normality. In that case, PSRs can be obtained by empirically ranking OMERs. Specifically, if we denote the OMER for observation i as $\hat{\epsilon}_i = y_i - \hat{y}_i$, the corresponding empirical PSR would be $\sum_{j=1}^{n} I(\hat{\epsilon}_j < \hat{\epsilon}_i)/n - \sum_{j=1}^{n} I(\hat{\epsilon}_j > \hat{\epsilon}_i)/n$ (Shepherd *et al.* 2016). This can be obtained with `presid()` by setting the argument `emp = TRUE`. In the `Wage` example, consider a linear regression model of `logwage` (log transformed `wage`) on `education`, `age`, `race`, `jobclass`, `maritl`, `health`, and `year`, where we apply the log transformation to `wage` due to its skewed distribution. The following chunk of code illustrates how to use `presid()` to obtain PSRs from linear regression models under different assumptions.

```
R> lm.mod <- lm(logwage ~ education + rcs(age, 5) + race + jobclass +
+    maritl + health + year, data = Wage)
R> PSR.lm.normal <- presid(lm.mod)
R> PSR.lm.emp <- presid(lm.mod, emp = TRUE)
R> OMER.lm <- residuals(lm.mod)
R> pit.trans <- pnorm(Wage$logwage, mean = lm.mod$fitted.values,
+    sd = summary(lm.mod)$sigma)
R> qresid <- qnorm(pnorm(Wage$logwage, mean = lm.mod$fitted.values,
+    sd = summary(lm.mod)$sigma))
R> summary(cbind(OMER.lm, PSR.lm.normal, PSR.lm.emp, pit.trans, qresid))
```

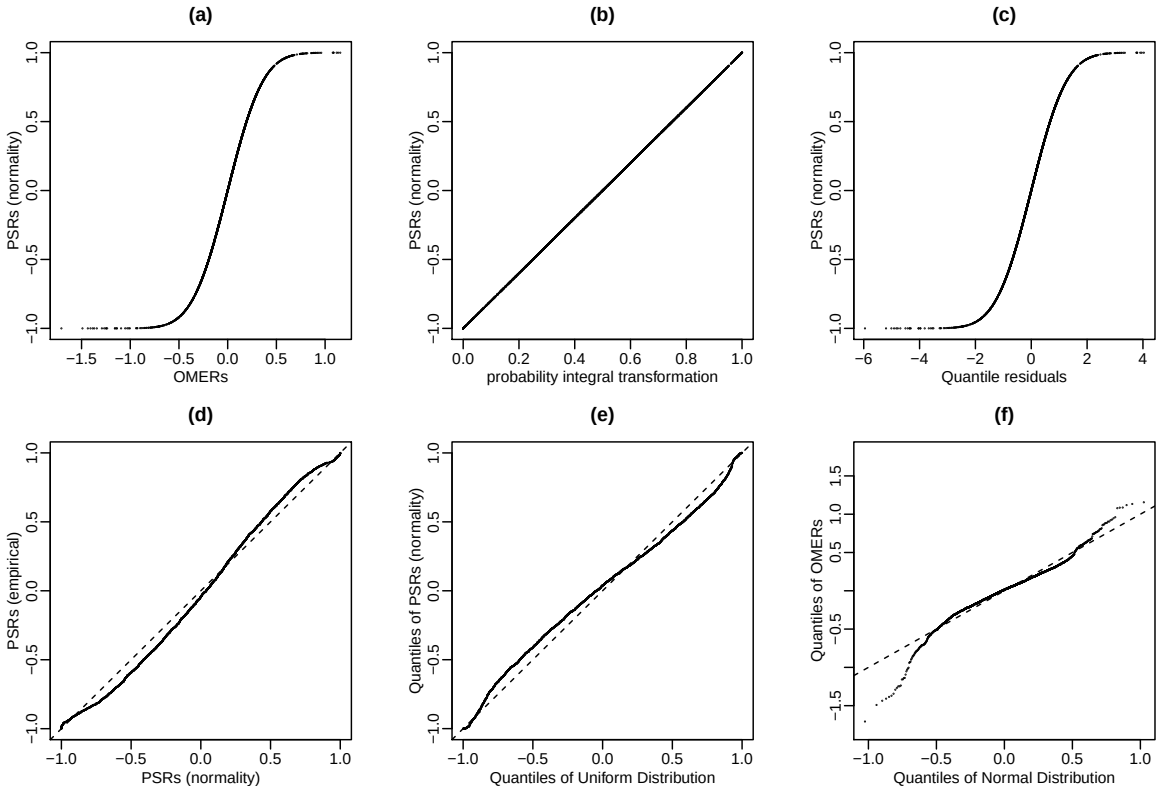

Figure 2: PSRs from linear regression models. (a): PSRs assuming normality are compared with OMERs. (b): PSRs assuming normality are compared with probability integral transformation. (c): PSRs assuming normality are compared with quantile residuals. (d): empirical PSRs are compared with PSRs assuming normality. (e): QQ-plot with PSRs under the assumption of normality. (f): QQ-plot with OMERs.

```
   OMER.lm         PSR.lm.normal        PSR.lm.emp        pit.trans
Min.   :-1.7070   Min.   :-1.00000   Min.   :-0.9997   Min.   :0.0000
1st Qu.:-0.1551   1st Qu.:-0.41148   1st Qu.:-0.4998   1st Qu.:0.2943
Median : 0.0138   Median : 0.03841   Median : 0.0000   Median :0.5192
Mean   : 0.0000   Mean   : 0.01264   Mean   : 0.0000   Mean   :0.5063
3rd Qu.: 0.1657   3rd Qu.: 0.43675   3rd Qu.: 0.4998   3rd Qu.:0.7184
Max.   : 1.1556   Max.   : 0.99994   Max.   : 0.9997   Max.   :1.0000
    qresid
Min.   :-5.95505
1st Qu.:-0.54097
Median : 0.04816
Mean   : 0.00000
3rd Qu.: 0.57802
Max.   : 4.03135
```

Figure 2 plots the PSRs from the log-transformed linear model of wages under different assumptions and their relationships with OMERs, the probability integral transformation, and quantile residuals. As shown in Figure 2 (a)–(c), for linear regression models, PSRs

(assuming normality) can be written as one-to-one functions of OMERs, the probability integral transformation, and quantile residuals. Since the PSRs of continuous responses are approximately uniformly distributed over $(-1, 1)$ under properly-specified models, the QQ-plot of the empirical quantiles of the PSRs versus theoretical quantiles of uniform$(-1, 1)$ can be used to assess the overall model fit (Shepherd *et al.* 2016). A QQ-plot of PSRs from linear regression assuming normality is also plotted in Figure 2, suggesting the normal linear assumption for `logwage` may not be ideal. A similar conclusion can be reached using OMERs. Note, the PSR under the assumption of homoscedasticity is obtained by empirically ranking the OMERs; therefore, it is uniformly distributed by construction and its QQ-plot does not provide useful information about the model fit. However, this empirical PSR can still be used in residual-by-predictor plots to detect lack of fit for specific predictors. For example, in Figure 3, we compare the residual-by-predictor plots using the empirical PSRs from linear regression models including both linear and nonlinear terms for `age` (transformed using restricted cubic splines) and not including nonlinear terms. Again, the smoothed curves show a clear nonlinear pattern, suggesting lack of fit when only including the linear term. A similar conclusion (although perhaps less striking in this example) can be obtained with OMERs.

Although the log transformation is commonly used for right-skewed data, it may not be optimal. Different transformations may give conflicting results. One option is to fit a semi-parametric transformation model. One such semiparametric transformation model, which can be viewed as a natural extension of ordinal cumulative probability models to continuous responses, can be fit using the `orm()` function in the **rms** package (Harrell Jr 2020). The PSR is the natural residual for this type of model, since conditional cumulative probabilities, instead of conditional means, are modeled (Sall 1991; Harrell 2015; Liu *et al.* 2017).We now illustrate its usage and the calculation of PSRs with `presid()` using the `Wage` data. Again, we need to set the arguments `x = TRUE` and `y = TRUE` when calling `orm()`.

```
R> cpm.probit <- orm(wage ~ education + rcs(age, 5) + race + jobclass +
+    maritl + health + year, data = Wage, x = TRUE, y = TRUE,
+    family = probit)
R> cpm.cloglog <- update(cpm.probit, family = cloglog)
R> PSR.cpm.probit <- presid(cpm.probit)
R> PSR.cpm.cloglog <- presid(cpm.cloglog)
R> summary(cbind(PSR.cpm.probit, PSR.cpm.cloglog))
```

```
 PSR.cpm.probit      PSR.cpm.cloglog
 Min.   :-0.99998   Min.   :-1.00000
 1st Qu.:-0.47360   1st Qu.:-0.42772
 Median : 0.03304   Median : 0.03212
 Mean   : 0.01123   Mean   : 0.00906
 3rd Qu.: 0.48780   3rd Qu.: 0.45051
 Max.   : 0.99994   Max.   : 0.99945
```

PSRs from cumulative probability models can also be used in QQ-plots and residual-by-predictor plots to assess model fit. Figure 4 shows QQ-plots of PSRs from cumulative probability models with the probit link and with the cloglog link, suggesting better model fit with

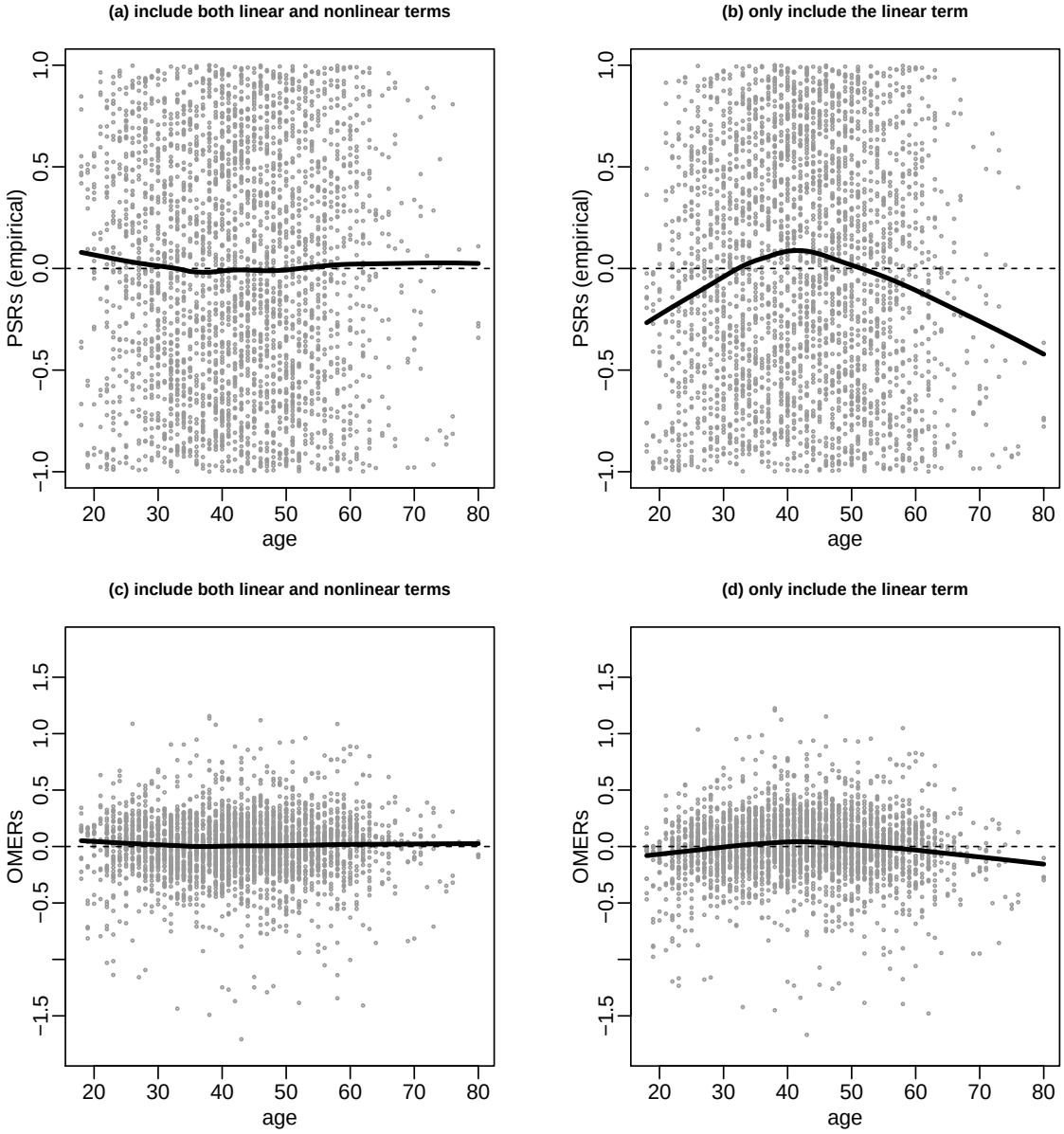

Figure 3: Residual-by-predictor plots using PSRs and OMERs from linear regression models. (a): PSRs are from the model including both linear and nonlinear terms. (b): PSRs are from the model only including the linear term. (c): OMERs are from the model including both linear and nonlinear terms. (d): OMERs are from the model only including the linear term.

the probit link. The residual-by-predictor plots in Figure 5 show a similar nonlinear relationship between wage and age as seen in the linear regression models. Similar conclusions can be obtained from QQ-plots and residual-by-predictor plots with OMERs (results not shown), although OMERs for cumulative probability models are not available for all observations and their computation is less straightforward. We refer readers to the paper of Liu *et al.* (2017) for the technical details.

To illustrate PSRs for censored outcomes with the `Wage` data, we artificially create a censoring

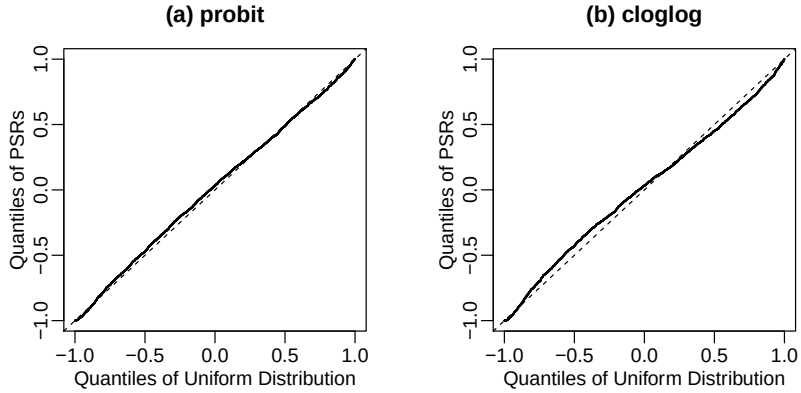

Figure 4: QQ-plots with PSRs from cumulative probability models with different link functions. (a) PSRs are from the model using the probit link. (b) PSRs are from the model using the cloglog link function.

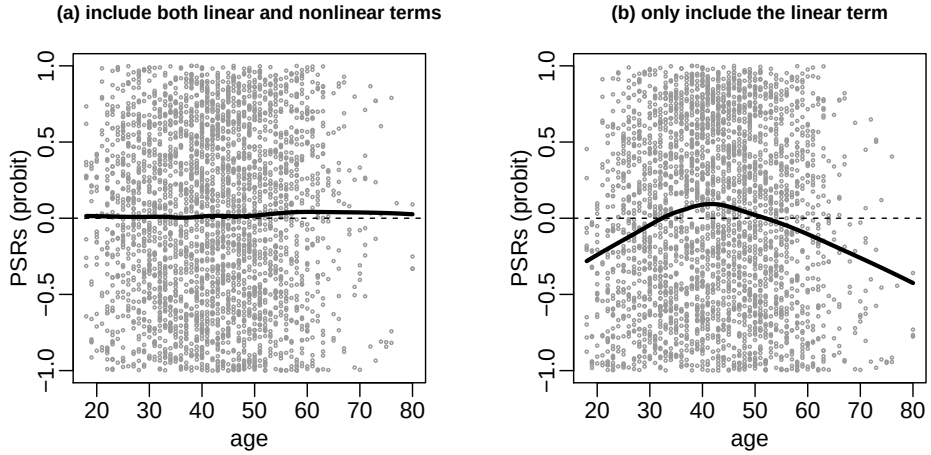

Figure 5: Residual-by-predictor plots using PSRs from cumulative probability models with the probit link function. (a): PSRs are from the model including both linear and nonlinear terms. (b): PSRs are from the model only including the linear term.

indicator δ with the probability of being censored equal to 0.2. If $\delta = 0$, we assume that the worker was not willing to share their exact wage and only reported a lower bound, i.e., the true wage is higher than the reported wage; whereas for workers with $\delta = 1$, we assume that they reported the exact value of their wages. We create this artificially censored dataset to illustrate the use of the PSR with right-censored data while maintaining the flow of the manuscript and allowing readers to easily compare and contrast PSRs with and without censoring. An additional illustration to a real dataset is provided in Appendix A.

```
R> set.seed(1)
R> Wage$delta <- sample(c(0, 1), size = dim(Wage)[1], replace = TRUE,
+    prob = c(0.2, 0.8))
```

Survival models can be used to model right censored data. We first illustrate how to obtain PSRs from parametric survival models. Specifically, we use the `survreg()` function in

the **survival** package (Therneau 2020) to fit three parametric survival models, assuming the response distribution is Weibull, logistic, or Gaussian.

```
R> library("survival")
R> psm.1 <- survreg(Surv(wage, delta) ~ education + rcs(age, 5) + race +
+    jobclass + maritl + health + year, dist = "weibull", data = Wage)
R> psm.2 <- update(psm.1, dist = "logistic")
R> psm.3 <- update(psm.1, dist = "gaussian")
R> PSR.psm.1 <- presid(psm.1)
R> PSR.psm.2 <- presid(psm.2)
R> PSR.psm.3 <- presid(psm.3)
R> summary(cbind(PSR.psm.1, PSR.psm.2, PSR.psm.3))
```

```
   PSR.psm.1   PSR.psm.2          PSR.psm.3
 Min.   :1   Min.   :-0.99570   Min.   :-0.99899
 1st Qu.:1   1st Qu.:-0.45755   1st Qu.:-0.44846
 Median :1   Median : 0.03148   Median :-0.04532
 Mean   :1   Mean   : 0.00000   Mean   :-0.03946
 3rd Qu.:1   3rd Qu.: 0.41572   3rd Qu.: 0.32982
 Max.   :1   Max.   : 0.99997   Max.   : 1.00000
```

PSRs for censored outcomes are generally not uniformly distributed even when the model is properly specified. To assess the overall model fit, we have considered a modified version of the PSR, referred to as a Cox-Snell-like PSR (Shepherd *et al.* 2016). This residual is simply the PSR evaluated at the observed value (ignoring censoring). It can be written as a one-to-one transformation of the Cox-Snell residual. Similar to the Cox-Snell residual which corresponds to a censored exponential(1) distribution, this modified PSR corresponds to a censored uniform distribution from -1 to 1 under the properly-specified model. By comparing its Kaplan–Meier estimate with the uniform distribution, we can assess the goodness of fit. The following chunk of code shows the calculation of Cox-Snell-like PSRs. Note that this modified version of the PSR generally does not have expectation 0. Figure 6 shows QQ-plots of Cox-Snell-like PSRs based on the Kaplan–Meier estimates, suggesting better model fit when assuming the censored outcomes follow a logistic distribution. For the purpose of comparison, QQ-plots with Cox-Snell residuals are also provided in Figure 6.

```
R> PSR.CS.psm.1 <- presid(psm.1, type = "Cox-Snell-like")
R> PSR.CS.psm.2 <- presid(psm.2, type = "Cox-Snell-like")
R> PSR.CS.psm.3 <- presid(psm.3, type = "Cox-Snell-like")
R> summary(cbind(PSR.CS.psm.1, PSR.CS.psm.2, PSR.CS.psm.3))
```

```
 PSR.CS.psm.1  PSR.CS.psm.2      PSR.CS.psm.3
 Min.   :1     Min.   :-0.9957   Min.   :-0.9990
 1st Qu.:1     1st Qu.:-0.5487   1st Qu.:-0.5258
 Median :1     Median :-0.1731   Median :-0.2106
 Mean   :1     Mean   :-0.1133   Mean   :-0.1572
 3rd Qu.:1     3rd Qu.: 0.2728   3rd Qu.: 0.1567
 Max.   :1     Max.   : 1.0000   Max.   : 1.0000
```

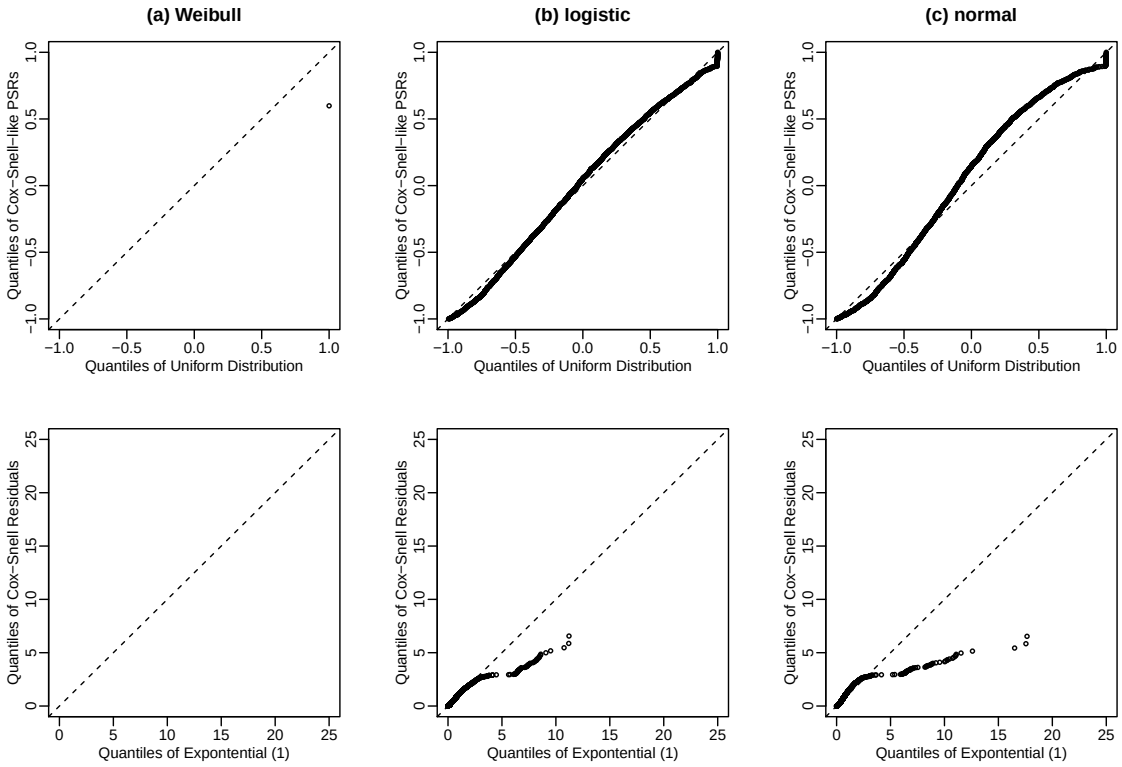

Figure 6: QQ-plots of Cox-Snell-like PSRs and the Cox-Snell residuals from parametric survival models with different distribution functions: (a) assuming Weilbull distribution, (b) assuming logistic distribution, and (c) assuming normal distribution.

The original PSR for censored data has expectation 0 under properly-specified models and independent censoring; therefore, it can be used in residual-by-predictor plots (Shepherd *et al.* 2016). Figure 7 plots PSRs from parametric survival models assuming the logistic distribution with and without the nonlinear terms for `age`, again suggesting a better fit when including the nonlinear terms. For the purpose of illustration, we highlight the PSRs of censored observations, showing that they are always non-negative. Similar conclusions can be obtained from the residual-by-predictor plots with martingale residuals (Figure 7), whose signs are opposite those of PSRs (Shepherd *et al.* 2016). However, due to their symmetric range, trends are often detected more easily with PSRs. This is not always the case, however; Appendix A contains an example where the martingale residuals are fairly symmetric and provide similar information to the PSR.

We can also fit semiparametric survival models, e.g., the widely used proportional hazards model, for the censored wage data. The PSR and the Cox-Snell-like PSR can be obtained using the following chunk of code. Figure 8 shows the QQ-plot of Cox-Snell-like PSRs and residual-by-predictor plots using PSRs from Cox proportional hazards models. For the purpose of comparison, the QQ-plot of Cox-Snell residuals and residual-by-predictor plots with martingale residuals are also provided in Figure 8. The results are generally similar to those in the parametric survival models.

```
R> coxph.1 <- coxph(Surv(wage, delta) ~ education + rcs(age, 5) + race +
+    jobclass + maritl + health + year, data = Wage)
```

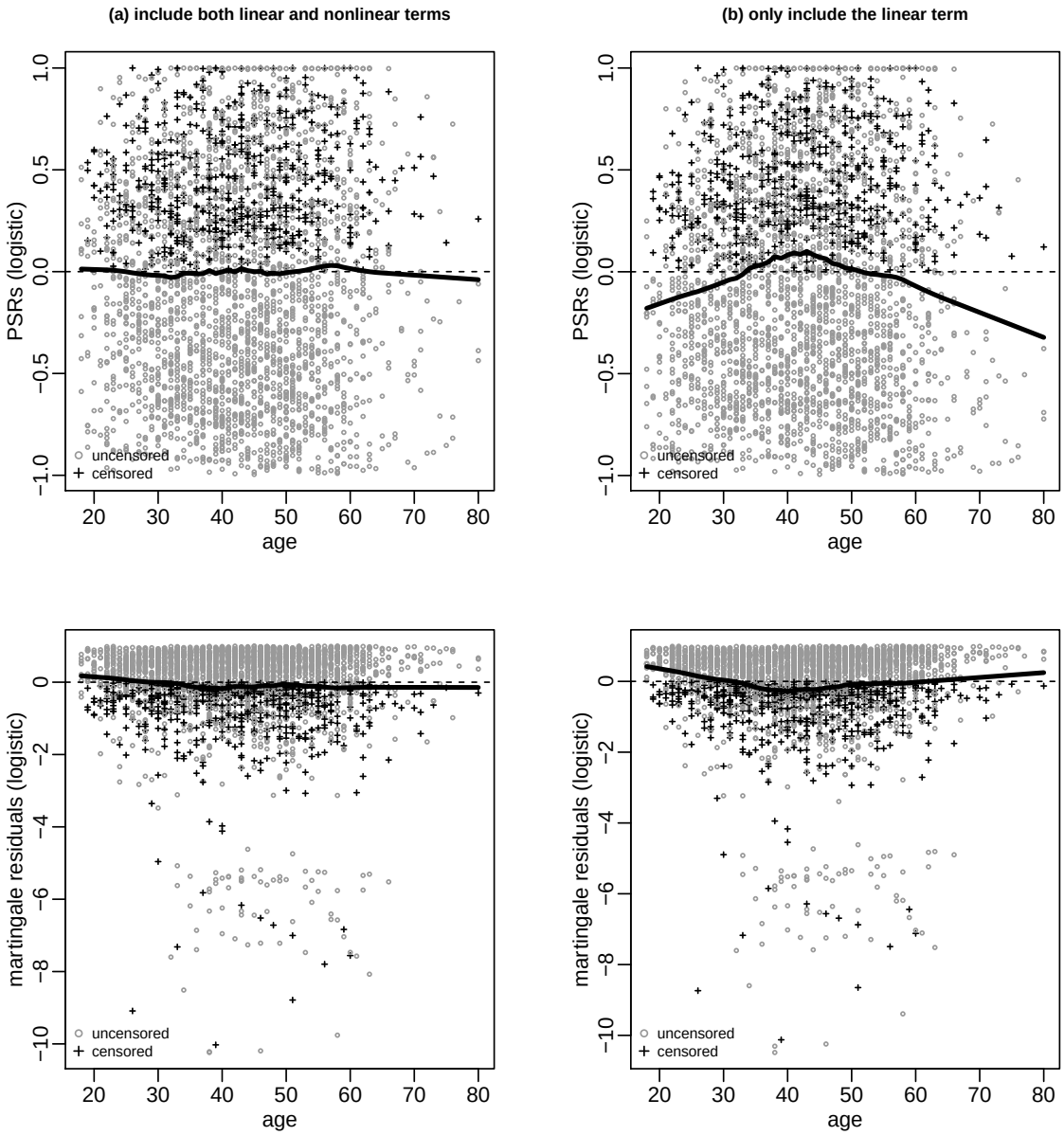

Figure 7: Residual-by-predictor plots using PSRs and martingale residuals from parametric survival models assuming the logistic distribution. (a): PSRs and martingale residuals are from the model including both linear and nonlinear terms. (b): PSRs and martingale residuals are from the model only including the linear term.

```
R> PSR.coxph <- presid(coxph.1)
R> PSR.CS.coxph <- presid(coxph.1, type = "Cox-Snell-like")
```

PSRs can also be computed for other types of data and models, for example, Poisson or negative binomial models for count data. The usage of the `presid()` function for these models is similar to what we described above. We refer readers to the manual and the help file of `presid()` for more details and examples.

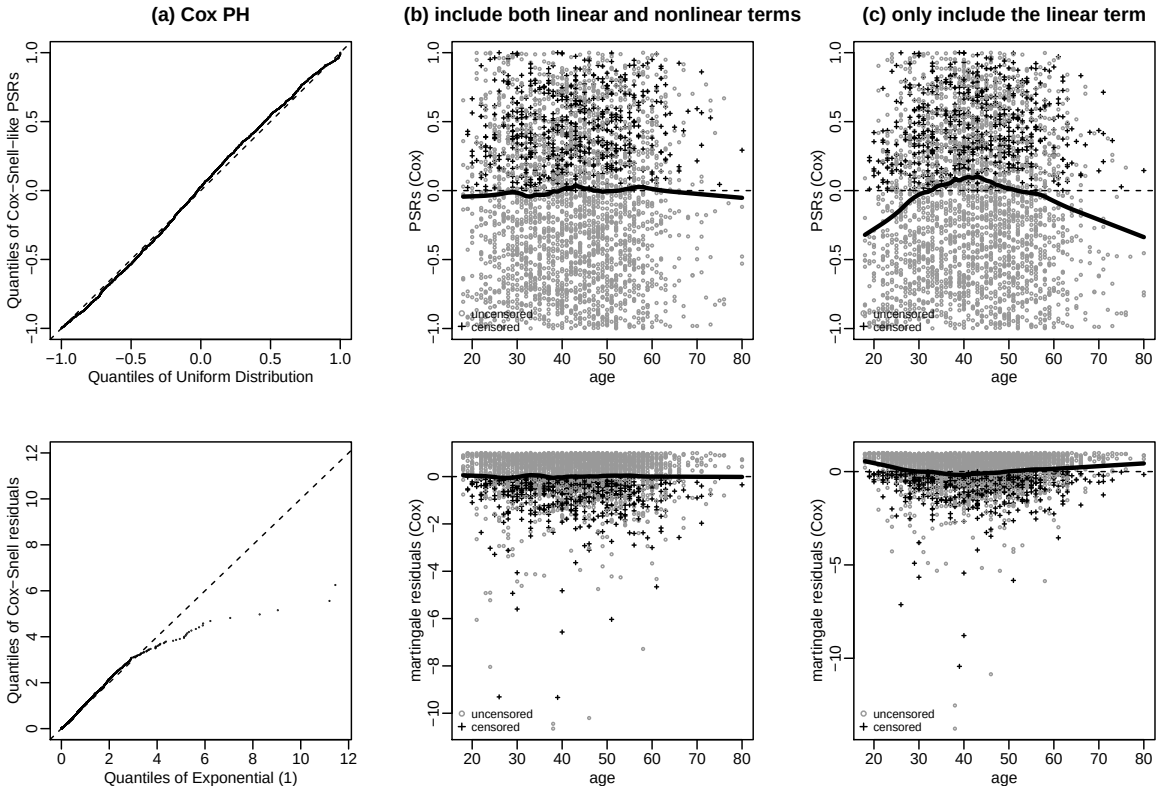

Figure 8: PSRs compared with Cox-Snell residuals and martingale residuals for Cox proportional hazards models. (a): QQ-plots using Cox-Snell-like PSRs and Cox-Snell residuals. (b): residual-by-predictor plots using PSRs and martingale residuals from the model including both linear and nonlinear terms. (c): residual-by-predictor plots using PSRs and martingale residuals from the model only including the linear term.

3.3. Tests of conditional association

In the previous section, we described the calculation of PSRs using the function `presid()` and illustrated their usage in model diagnostics. In this section, we focus on inference. Specifically, we describe how to use the **PResiduals** package (Dupont *et al.* 2020) to perform tests of conditional association.

Assume that we want to examine the association between `wage` and `education` while adjusting for a few potential confounders, such as `age`, `race`, `jobclass`, `maritl`, `health`, and `year`. One may consider fitting the linear regression or cumulative probability models we described earlier and then examining the regression coefficient for education. In this example, both the linear regression model and the cumulative probability model suggest significant associations between `wage` and `education` after adjusting for other covariates (results not shown).

However, in both regression models, the ordinal predictor `education` is coded as a categorical variable and the order information is ignored. To use the order information, we may consider assigning scores, e.g., the approximate years of education to different education levels, but that would force an assumption of linearity. COBOT provides a way to test for conditional associations while accounting for the ordinal nature of ordered categorical predictors. The COBOT approach has been implemented in the **PResiduals** package with the `cobot()` func-

tion. To illustrate its usage with the `Wage` data, we create an ordered categorical variable for wage, referred to as `wage.level`, by discretizing `wage` into five categories. Note, this is simply for the purpose of illustration and we do not recommend categorizing continuous variables in real data analyses (Royston, Altman, and Sauerbrei 2006); we demonstrate below how to do the analysis leaving wage as a continuous variable. The `cobot()` function takes a formula object in the form of $X \mid Y \sim Z$, where X and Y are the ordinal variables whose relationship we are interested in, and Z designates the covariates we want to adjust for. Note that Z could be multidimensional covariates with transformations. By default, `cobot()` fits proportional odds models for both X on Z and Y on Z. Cumulative probability models with other link functions can be specified with the arguments `link.x` and `link.y`. The `cobot()` function reports three test statistics proposed in Li and Shepherd (2010) and their standard errors, p values, and confidence intervals. The second statistic, T2, is the correlation of PSRs. Fisher's transformation is used by default to compute p values and confidence intervals for T2. In this example, we find a strong positive association between education and the discretized wage with highly significant p values and tight confidence intervals away from zero.

```
R> Wage$wage.level <- cut(Wage$wage,
+    breaks = c(0, quantile(Wage$wage, c(0.2, 0.4, 0.6, 0.8)), Inf))
R> summary(Wage$wage.level)

   (0,81.3] (81.3,97.5]  (97.5,114]   (114,135]   (135,Inf]
        661         548         635         558         598

R> cobot(wage.level | education ~ rcs(age, 5) + race + jobclass + maritl +
+    health + year, data = Wage)

                               est      stderr             p  lower CI
Gamma(Obs) - Gamma(Exp)  0.3873366 0.015024858 1.497331e-146 0.3574999
Correlation of Residuals 0.4455342 0.015584921 4.729171e-134 0.4144760
Covariance of Residuals  0.1367667 0.004861128 3.680273e-174 0.1272267
                          upper CI
Gamma(Obs) - Gamma(Exp)  0.4163833
Correlation of Residuals 0.4755558
Covariance of Residuals  0.1462814
Confidence Interval: 95%
Number of Observations: 3000
```

Since PSRs are well defined for a wide variety of outcomes, the COBOT approach based on PSRs can be extended to other types of X and Y as long as they are orderable. For example, in the **PResiduals** package, we have implemented `cocobot()` for an ordinal X and a continuous Y, `countbot()` for an ordinal X and a count variable Y, and a wrapper function `megabot()` for any orderable X and Y. The usage of `megabot()` is very similar to `cobot()` and is illustrated in the following chunk of code. Flexible modeling choices are available for both X on Z and Y on Z, and can be specified with the arguments `fit.x` and `fit.y`. Currently supported fitting procedures include `ordinal` (ordinal cumulative probability models fitted with `polr()`), `lm` (linear regression models assuming normality), `lm.emp` (linear regression models assuming homoscedasticity), `orm` (continuous or discrete cumulative probability models fitted

with `orm()`), `poisson` (Poisson models for count data), and `nb` (negative binomial models for count data). If cumulative probability models are used (with either `polr()` or `orm()`), the default link function is the logit function and other link functions can be specified with arguments `link.x` and `link.y`. We give a few examples, using PSRs for `education` obtained from ordinal cumulative probability models fitted with either `polr()` or `orm()` and PSRs for `wage` obtained from either linear regression models or cumulative probability models. Note that when cumulative probability models are used for both models of X on Z and of Y on Z, the test results only use rank information of X and Y and are therefore invariant to any monotonic transformations of X or Y. Results are very similar across different models.

```
R> megabot(logwage | education ~ rcs(age, 5) + race + jobclass + maritl +
+    health + year, data = Wage, fit.x = "lm.emp", fit.y = "ordinal")

               est     stderr             p  lower CI  upper CI
cor PSRs 0.4403039 0.01585574 1.412076e-127 0.4087066 0.4708471
Confidence Interval: 95%
Number of Observations: 3000
Fisher Transform: TRUE

R> megabot(logwage | education ~ rcs(age, 5) + race + jobclass + maritl +
+    health + year, data = Wage, fit.x = "lm.emp", fit.y = "ordinal",
+    link.y = "cloglog")

               est     stderr            p  lower CI  upper CI
cor PSRs 0.4409901 0.01562993 1.67254e-131 0.4098487 0.4711046
Confidence Interval: 95%
Number of Observations: 3000
Fisher Transform: TRUE

R> megabot(wage | education ~ rcs(age, 5) + race + jobclass + maritl +
+    health + year, data = Wage, fit.x = "orm", fit.y = "orm")

               est     stderr             p  lower CI  upper CI
cor PSRs 0.4428448 0.01564295 5.103498e-132 0.4116738 0.4729808
Confidence Interval: 95%
Number of Observations: 3000
Fisher Transform: TRUE
```

3.4. Covariate-adjusted Spearman's rank correlation with PSRs

As discussed in Section 2, PSRs can be used to construct partial and conditional Spearman's rank correlation adjusting for covariates (Liu *et al.* 2018). The test statistics in our tests for conditional association implemented in `megabot()` are actually partial Spearman's correlations.

We have implemented the function `partial_Spearman()` to obtain the partial Spearman's correlation where cumulative probability models are set as the default modeling method for both discrete and continuous ordinal X and Y (fitted with `orm()`). The following chunk of code illustrates its usage with different link functions in the example of `wage` and `education`.

```
R> partial_Spearman(wage | education ~ rcs(age, 5) + race + jobclass +
+    maritl + health + year, data = Wage, link.x = "logit",
+    link.y = "logit")

                      est     stderr             p  lower CI  upper CI
partial Spearman 0.4428448 0.01564295 5.103498e-132 0.4116738 0.4729808
Fisher Transform: TRUE
Confidence Interval: 95%
Number of Observations: 3000

R> partial_Spearman(wage | education ~ rcs(age, 5) + race + jobclass +
+    maritl + health + year, data = Wage, link.x = "probit",
+    link.y = "probit")

                      est    stderr            p  lower CI  upper CI
partial Spearman 0.4448799 0.0156437 8.34341e-133 0.4137038 0.4750138
Fisher Transform: TRUE
Confidence Interval: 95%
Number of Observations: 3000

R> partial_Spearman(wage | education ~ rcs(age, 5) + race + jobclass +
+    maritl + health + year, data = Wage, link.x = "cloglog",
+    link.y = "cloglog")

                      est     stderr             p  lower CI  upper CI
partial Spearman 0.4580628 0.01555897 2.233473e-139 0.4270347 0.4880135
Fisher Transform: TRUE
Confidence Interval: 95%
Number of Observations: 3000
```

The result with the logit link function shows that after adjusting for other covariates, the partial Spearman's rank correlation between `wage` and `education` is 0.44 with 95% confidence interval (CI) (0.41, 0.47). This is lower than the unadjusted Spearman's rank correlation 0.50 (95% CI: 0.47, 0.53), suggesting that part of the association between `wage` and `education` can be explained by their association with other covariates. Note that the point estimates and their confidence intervals are very similar with different link functions. (We did not report the result with the loglog link function because the cumulative probability model did not converge.) This is consistent with our simulations in Liu *et al.* (2018) where we found that the partial Spearman's rank correlation using PSRs from `orm()` is robust to link function misspecification.

It may be useful to examine the correlation of PSRs as a function of a single covariate. For example, we may be interested in whether Spearman's correlation varies for different job classes or ages while still adjusting for other covariates. The function `conditional_Spearman()` can be used to obtain the partial Spearman's correlation conditional on a specific covariate, denoted as Z_1. The usage of `conditional_Spearman()` is very similar to `megabot()` and `partial_Spearman()`. It takes a formula object in the form of $X \mid Y \sim Z$ to specify the

models of X on Z and of Y on Z. The fitting procedures can be specified with arguments `fit.x` and `fit.y` with the default as cumulative probability models with the logit link function. The covariate Z_1 is specified by the argument `conditional.by`. Different methods have been implemented to model the conditional correlation of PSRs and can be specified using the argument `conditional.method`. For categorical covariates such as `jobclass`, the conditional correlation of PSRs can be obtained by stratification, that is, we compute the correlation of PSRs within each category of `jobclass`. This can be achieved by setting `conditional.method = "stratification"`. For example,

```
R> conditional_Spearman(education | wage ~ rcs(age, 5) + race +
+    jobclass + maritl + health + year, conditional.by = "jobclass",
+    conditional.method = "stratification", data = Wage)

Partial Spearman's correlation conditional by: jobclass
Conditional method: stratification
Number of levels of jobclass : 2
        jobclass       est     stderr            p  lower.CI  upper.CI
1  1. Industrial 0.4079285 0.02287611 4.085476e-56 0.3621315 0.4517609
2 2. Information 0.4782682 0.02107400 5.666486e-81 0.4359197 0.5185035
Fisher Transform: TRUE
Confidence Interval: 95%
Number of Observations: 3000
```

If the stratification method is used, `conditional_Spearman()` reports the point estimates, standard error estimates, p value, and 95% confidence intervals for each category. In this example, after adjusting for other factors, Spearman's rank correlation between `wage` and `education` is higher in the information job class than that in the industrial class: 0.48 (95% CI: 0.44, 0.52) vs. 0.41 (95% CI: 0.36, 0.45).

For continuous variables such as `age`, two options are available for `conditional.method`: one is `lm`, which fits linear regression models for $X_{\text{res}}Y_{\text{res}}$ on Z_1, X^2_{res} on Z_1, and Y^2_{res} on Z_1 and then estimates the conditional correlation of PSRs using the fitted values, and the other is `kernel`, which estimates the conditional correlation of PSRs nonparametrically with kernel smoothing, allowing the user to input bandwidth parameters. Details are in Liu *et al.* (2018). These features are implemented in `conditional_Spearman()`. The results can be printed (showing the first few observations by default) and plotted directly using `plot()`.

```
R> conditional.lm <- conditional_Spearman(wage | education ~
+    rcs(age, 5) + race + jobclass + maritl + health + year,
+    conditional.by = "age", conditional.method = "lm",
+    conditional.formula = " ~ rcs(age, 5)", data = Wage)
R> conditional.lm

Partial Spearman's correlation conditional by: age
Conditional method: lm
Conditional Formula: ~ rcs(age, 5)
  age        est     stderr            p   lower.CI  upper.CI
1  18 -0.0595070 0.11620834 6.094477e-01 -0.2804321 0.1674055
```

```
2  24  0.2014775 0.05041056 1.012077e-04  0.1009438 0.2979382
3  45  0.4983515 0.02531541 2.447729e-59  0.4471233 0.5463211
4  43  0.4958329 0.02878779 4.819363e-46  0.4373494 0.5501400
5  50  0.4982091 0.02759435 3.273183e-50  0.4422148 0.5503349
6  54  0.4767692 0.03082396 1.144778e-38  0.4141486 0.5348978
...
Fisher Transform: TRUE
Confidence Interval: 95%
Number of Observations: 3000
```

```
R> conditional.kernel <- conditional_Spearman(wage | education ~
+    rcs(age, 5) + race + jobclass + maritl + health + year,
+    conditional.by = "age", conditional.method = "kernel",
+    kernel.bandwidth = "silverman", data = Wage)
R> conditional.kernel
```

```
Partial Spearman's correlation conditional by: age
Conditional method: kernel
kernel function: normal
kernel bandwidth: 2.467
     age        est
[1,]  18 0.01784734
[2,]  24 0.24475183
[3,]  45 0.50106153
[4,]  43 0.49954376
[5,]  50 0.50399806
[6,]  54 0.49989385
...
Fisher Transform: TRUE
Confidence Interval: 95%
Number of Observations: 3000
```

For the `lm` methods, `conditional_Spearman()` reports standard error estimates and pointwise confidence intervals, obtained by M-estimation methods with Fisher's transformation. For the `kernel` methods, only the point estimates are returned. Figure 9 shows that results from the two methods are similar, both suggesting that after adjusting for other factors, Spearman's rank correlation between wage and education is weaker among those who are younger (< 30 years).

4. Summary

The **PResiduals** package provides user-friendly functions for residual analysis with probability-scale residuals. The probability-scale residual is applicable across a wide variety of data types and models, and thus allows comparison of different models on the same scale. The PSR can also be used to construct conditional tests of association and to compute covariate-adjusted Spearman rank correlations. This paper illustrates its usage with examples. We hope users find it useful for model diagnostics and for assessing covariate-adjusted associations.

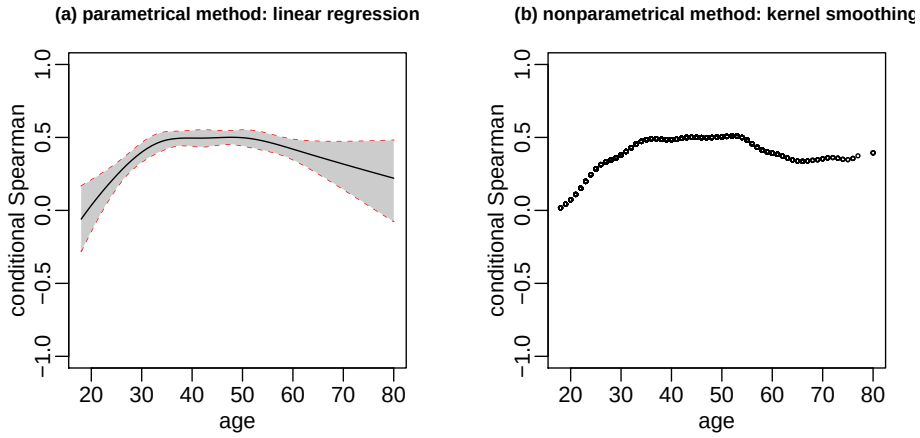

Figure 9: The age-specific conditional Spearman's rank correlation between wage and education: (a) modeled parametrically using linear regression models, (b) modeled nonparametrically using kernel smoothing.

References

Agresti A (2010). *Analysis of Ordinal Categorical Data*. 2nd edition. John Wiley & Sons, Hoboken. doi:10.1002/9780470594001.

Dunn PK, Smyth GK (1996). "Randomized Quantile Residuals." *Journal of Computational and Graphical Statistics*, **5**(3), 236–244. doi:10.1080/10618600.1996.10474708.

Dupont C, Horner J, Li C, Liu Q, Shepherd BE (2020). ***PResiduals**: Probability-Scale Residuals and Residual Correlations*. R package version 1.0-0, URL https://CRAN.R-project.org/package=PResiduals.

Harrell FE (2015). *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. 2nd edition. Springer-Verlag, New York. doi:10.1007/978-3-319-19425-7.

Harrell Jr FE (2020). ***rms**: Regression Modeling Strategies*. R package version 6.0-1, URL https://CRAN.R-project.org/package=rms.

Hothorn T (2019). ***TH.data**: TH's Data Archive*. R package version 1.0-10, URL https://CRAN.R-project.org/package=TH.data.

James G, Witten D, Hastie T, Tibshirani R (2017). ***ISLR**: Data for An Introduction to Statistical Learning with Applications in R*. R package version 1.2, URL https://CRAN.R-project.org/package=ISLR.

Li C, Shepherd BE (2010). "Test of Association between Two Ordinal Variables While Adjusting for Covariates." *Journal of the American Statistical Association*, **105**(490), 612–620. doi:10.1198/jasa.2010.tm09386.

Li C, Shepherd BE (2012). "A New Residual for Ordinal Outcomes." *Biometrika*, **99**(2), 473–480. doi:10.1093/biomet/asr073.

Liu D, Zhang H (2018). "Residuals and Diagnostics for Ordinal Regression Models: A Surrogate Approach." *Journal of the American Statistical Association*, **113**(522), 845–854. doi:10.1080/01621459.2017.1292915.

Liu Q, Li C, Wanga V, Shepherd BE (2018). "Covariate-Adjusted Spearman's Rank Correlation with Probability-Scale Residuals." *Biometrics*, **74**(2), 595–605. doi:10.1111/biom.12812.

Liu Q, Shepherd BE, Li C, Harrell FE (2017). "Modeling Continuous Response Variables Using Ordinal Regression." *Statistics in Medicine*, **36**(27), 4316–4335. doi:10.1002/sim.7433.

McCullagh P (1980). "Regression Models for Ordinal Data." *Journal of the Royal Statistical Society B*, **42**(2), 109–142. doi:10.1111/j.2517-6161.1980.tb01109.x.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ripley BD (2020). ***MASS**: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3-52, URL https://CRAN.R-project.org/package=MASS.

Royston P, Altman DG, Sauerbrei W (2006). "Dichotomizing Continuous Predictors in Multiple Regression: A Bad Idea." *Statistics in Medicine*, **25**(1), 127–141. doi:10.1002/sim.2331.

Sall J (1991). "A Monotone Regression Smoother Based on Ordinal Cumulative Logistic Regression." In *ASA Proceedings of Statistical Computing Section*, pp. 276–281.

Schumacher M, Bastert G, Bojar H, Hübner K, Olschewski M, Sauerbrei W, Schmoor C, Beyerle C, Neumann RL, Rauschecker HF (1994). "Randomized 2 × 2 Trial Evaluating Hormonal Treatment and the Duration of Chemotherapy in Node-Positive Breast Cancer Patients. German Breast Cancer Study Group." *Journal of Clinical Oncology*, **12**(10), 2086–2093. doi:10.1200/jco.1994.12.10.2086.

Shepherd BE, Li C, Liu Q (2016). "Probability-Scale Residuals for Continuous, Discrete, and Censored Data." *Canadian Journal of Statistics*, **44**(4), 463–479. doi:10.1002/cjs.11302.

Therneau TM (2020). ***survival**: A Package for Survival Analysis in R*. R package version 3.2-3, URL https://CRAN.R-project.org/package=survival.

Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag, New York. doi:10.1007/978-1-4757-3294-8.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. doi:10.1007/978-0-387-21706-2. URL https://www.stats.ox.ac.uk/pub/MASS4.

Wang X, Ye Y, Zhang H (2006). "Family-Based Association Tests for Ordinal Traits Adjusting for Covariates." *Genetic Epidemiology*, **30**(8), 728–736. doi:10.1002/gepi.20184.

A. Additional examples

We provide additional examples of residual analysis with PSRs for time-to-event outcomes using the `GBSG2` dataset in the R package **TH.data** (Hothorn 2019). The `GBSG2` dataset contains the recurrence-free survival time of 686 women from German Breast Cancer Study Group (GBSG) study, along with other covariate information summarized in Table 2 (Schumacher *et al.* 1994).

Similarly as in Section 3.2, we build parametric survival models (assuming the response distribution is Weibull, logistic, or Gaussian) and the semiparametric Cox proportional hazards models for the recurrence-free survival time. Specifically, we include `horTh`, `age`, `menostat`, `tsize`, `tgrade`, `pnodes`, `progrec` and `estrec` as covariates; in addition, with transformations of `age`, `tsize`, `pnodes`, and `progrec` using restricted cubic splines to account for potential nonlinear relationships. The following chunk of code shows the calculation of PSRs and Cox-Snell-like PSRs for parametric and semiparametric survival models with the `GBSG2` dataset.

```
R> library("PResiduals")
R> data("GBSG2", package = "TH.data")
R> psm.1 <- survreg(Surv(time, cens) ~ horTh + rcs(age) + menostat +
+    rcs(tsize) + tgrade + rcs(pnodes) + rcs(progrec) + estrec,
+    dist = "weibull", data = GBSG2)
R> psm.2 <- update(psm.1, dist = "logistic")
R> psm.3 <- update(psm.1, dist = "gaussian")
R> PSR.psm.1 <- presid(psm.1)
R> PSR.psm.2 <- presid(psm.2)
R> PSR.psm.3 <- presid(psm.3)
R> summary(cbind(PSR.psm.1, PSR.psm.2, PSR.psm.3))

   PSR.psm.1         PSR.psm.2          PSR.psm.3
 Min.   :0.4302   Min.   :-0.9700   Min.   :-0.98509
 1st Qu.:1.0000   1st Qu.:-0.4623   1st Qu.:-0.49117
 Median :1.0000   Median : 0.1002   Median : 0.09344
 Mean   :0.9992   Mean   : 0.0000   Mean   :-0.01329
 3rd Qu.:1.0000   3rd Qu.: 0.3732   3rd Qu.: 0.37257
 Max.   :1.0000   Max.   : 0.9359   Max.   : 0.93506

R> PSR.CS.psm.1 <- presid(psm.1, type = "Cox-Snell-like")
R> PSR.CS.psm.2 <- presid(psm.2, type = "Cox-Snell-like")
R> PSR.CS.psm.3 <- presid(psm.3, type = "Cox-Snell-like")
R> summary(cbind(PSR.CS.psm.1, PSR.CS.psm.2, PSR.CS.psm.3))

  PSR.CS.psm.1       PSR.CS.psm.2        PSR.CS.psm.3
 Min.   :-0.1396   Min.   :-0.99621   Min.   :-0.99956
 1st Qu.: 1.0000   1st Qu.:-0.76743   1st Qu.:-0.77002
 Median : 1.0000   Median :-0.49445   Median :-0.50150
 Mean   : 0.9983   Mean   :-0.37211   Mean   :-0.38799
 3rd Qu.: 1.0000   3rd Qu.:-0.05407   3rd Qu.:-0.07789
 Max.   : 1.0000   Max.   : 0.91641   Max.   : 0.92175
```

Variable	Description
`horTh`	Hormonal therapy, a factor at two levels no and yes
`age`	Age of the patients in years
`menostat`	Menopausal status, a factor at two levels pre and post
`tsize`	Tumor size (in mm)
`tgrade`	Tumor grade, a ordered factor at levels I < II < III
`pnodes`	Number of positive nodes
`progrec`	Progesterone receptor (in fmol)
`estrec`	Estrogen receptor (in fmol)
`time`	Recurrence free survival time (in days)
`cens`	censoring indicator (0: censored, 1: event)

Table 2: Variables in `GBSG2` dataset.

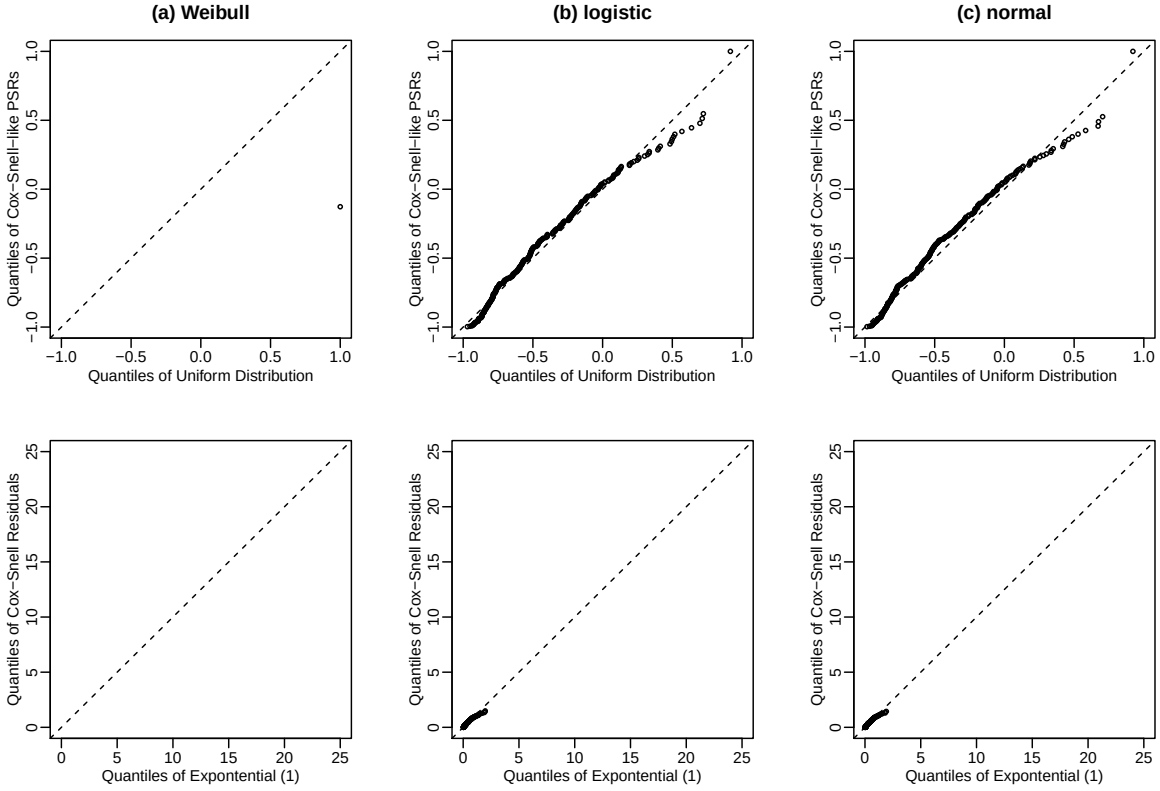

Figure 10: QQ-plots of Cox-Snell-like PSRs and the Cox-Snell residuals from parametric survival models with different distribution functions for the `GBSG2` dataset: (a) assuming Weibull distribution, (b) assuming logistic distribution, and (c) assuming normal distribution.

```
R> coxph.1 <- coxph(Surv(time, cens) ~ horTh + rcs(age) + menostat +
+    rcs(tsize) + tgrade + rcs(pnodes) + rcs(progrec) + estrec, data = GBSG2)
R> PSR.coxph <- presid(coxph.1)
R> PSR.CS.coxph <- presid(coxph.1, type = "Cox-Snell-like")
```

Figures 10 and 11 show QQ-plots of Cox-Snell-like PSRs from the parametric survival models

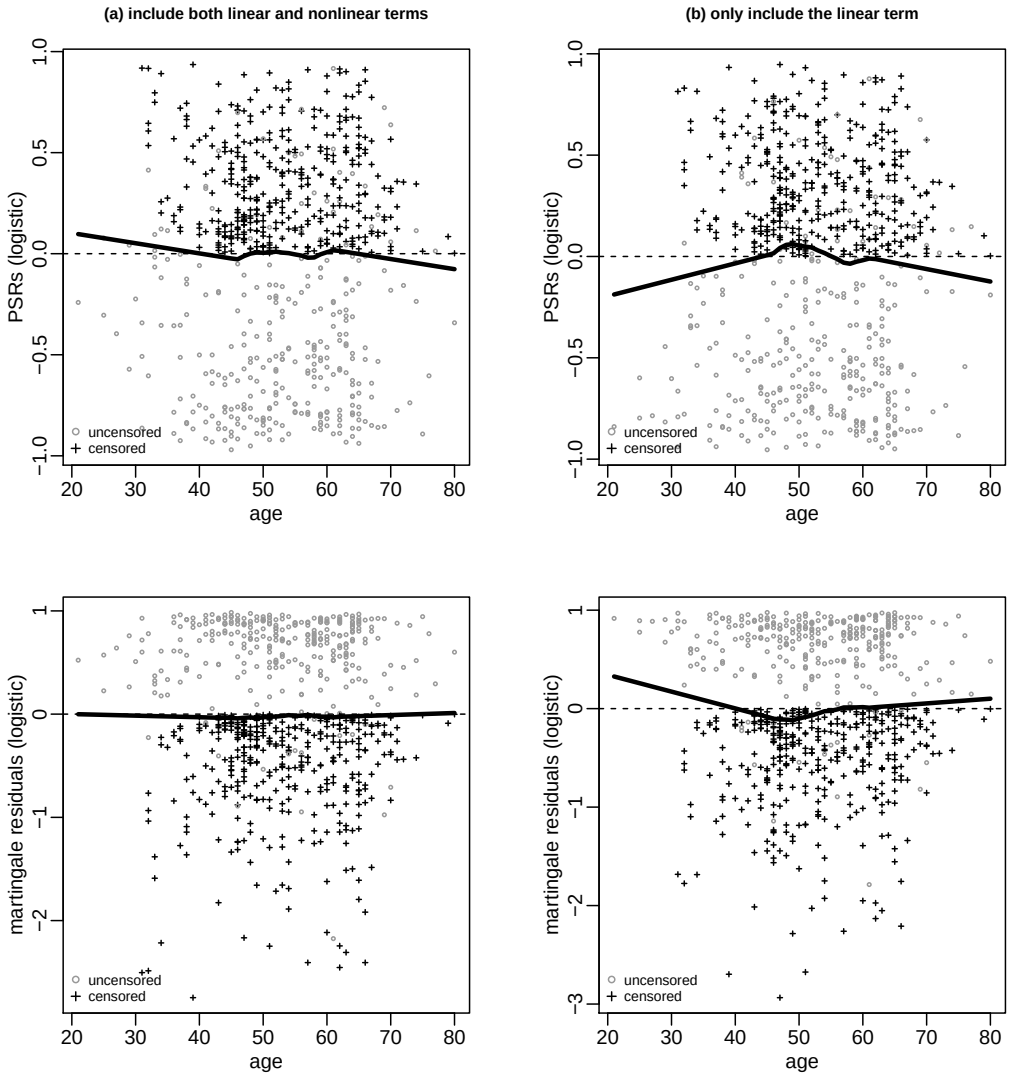

Figure 11: Residual-by-predictor plots using PSRs and martingale residuals from parametric survival models assuming the logistic distribution for the `GBSG2` dataset. (a): PSRs and martingale residuals are from the model including both linear and nonlinear terms. (b): PSRs and martingale residuals are from the model only including the linear term.

with different distribution functions and the residual-by-predictor plots using PSRs for the parametric survival models assuming the logistic distribution, respectively. Figure 12 shows the QQ-plot of Cox-Snell-like PSRs and residual-by-predictor plots using PSRs from Cox proportional hazards models. For the purpose of comparison, the QQ plots of Cox-Snell residuals and residual-by-predictor plots with martingale residuals are also provided. In this specific real survival data example, although the advantages of using PSRs are not as illustrative as in the `Wage` example in Section 3.2, e.g., non-linear effects are less apparent, the QQ-plots do not readily distinguish between parametric survival models, and the range of the martingale residual is more symmetric, the performance of PSRs is generally comparable with Cox-snell residuals and martingale residuals.

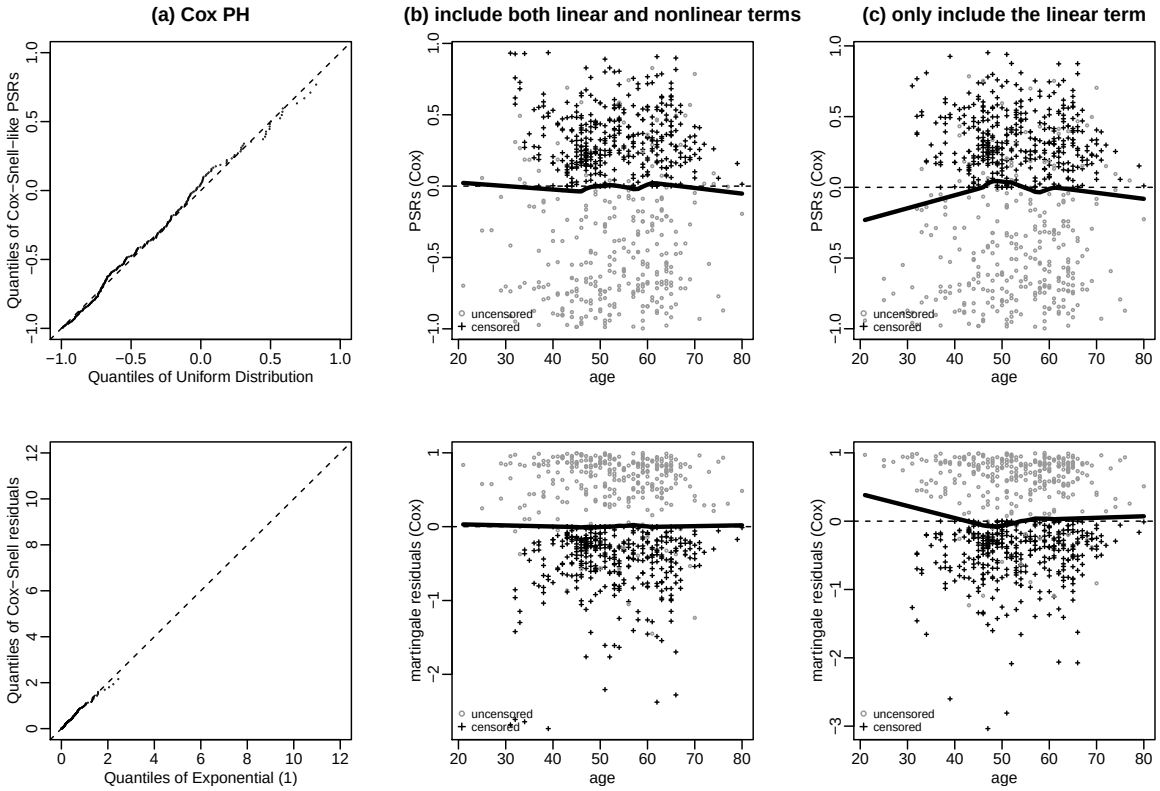

Figure 12: PSRs compared with Cox-Snell residuals and martingale residuals for Cox proportional hazards models for the `GBSG2` dataset. (a): QQ-plots using Cox-Snell-like PSRs and Cox-Snell residuals. (b): residual-by-predictor plots using PSRs and martingale residuals from the model including both linear and nonlinear terms. (c): residual-by-predictor plots using PSRs and martingale residuals from the model only including the linear term.

Affiliation:

Qi Liu
Biostatistics and Research Decision Sciences
Merck & CO., Inc.
126 E. Lincoln Avenue
Rahway, NJ 07065, United States of America
E-mail: qi.liu4@merck.com

Bryan Shepherd
Department of Biostatistics
Vanderbilt University School of Medicine
2525 West End, Suite 11000
Nashville, TN 37203, United States of America
E-mail: bryan.shepherd@vanderbilt.edu
URL: http://biostat.mc.vanderbilt.edu/wiki/Main/BryanShepherd/

Chun Li
Department of Preventive Medicine
University of Southern California
2001 N. Soto St.
Los Angeles, CA 90033, United States of America
E-mail: cli77199@usc.edu
URL: https://preventivemedicine.usc.edu/Chun-Li/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
August 2020, Volume 94, Issue 12 *Submitted:* 2017-11-21
doi:10.18637/jss.v094.i12 *Accepted:* 2019-04-09