responseRateAnalysis - Overview

Daniel Heimgartner

June 27, 2024

The default data can be loaded with `default_data()`. This applies the logistic transformation to the response rate, divides the response burden score by 1000 and defines the weights to be `sqrt(sample_size)`.

```
> dat <- default_data()
> head(dat)
#> # A tibble: 6 x 13
#>        y     x weight  year authors  survey_id response_rate
#>    <dbl> <dbl>  <dbl> <dbl> <chr>        <dbl>         <dbl>
#> 1  0.741 0.12    39.5  2004 Vrtic a~         1          67.7
#> 2  0.889 0.12    35.1  2006 Vrtic a~         2          70.9
#> 3  0.110 0.303   48.1  2007 Axhause~         3          52.7
#> 4 -0.584 0.44    22.4  2004 Locatel~         4          35.8
#> 5 -0.778 0.526   43.6  2006 Jäggle           5          31.5
#> 6 -0.572 0.231   96.6  2005 Waldner~         6          36.1
#> # i 6 more variables: response_burden_score <dbl>,
#> #   sample_size <dbl>, yes_yes <dbl>, yes_no <dbl>,
#> #   no_no <dbl>, no_yes <dbl>
```

The model is a simple weighted linear regression model with a logistic transformation of the response. We can add clustered standard errors and p-values with the `add_clustered()`. The `summary()` generic has a method for class "clustered" (using `texreg::screenreg()` under the hood).

```
> fit <- lm(y ~ 0 + x + yes_yes + yes_no + no_no + no_yes,
+           data = dat, weights = weight)
> m1 <- add_clustered(fit, cluster = dat$survey_id, type = "CR2")
> summary(m1)  # texreg::screenreg(fit_)
#>
#> ======================
#>            Model 1
#> ----------------------
#> x            -1.01 ***
#>              (0.19)
#> yes_yes       1.89 ***
#>              (0.25)
#> yes_no        0.63
#>              (0.33)
#> no_no        -0.64
```

```
#>              (0.34)
#> no_yes       -0.73
#>              (0.57)
#> ---------------------
#> R^2           0.81
#> Adj. R^2      0.80
#> Num. obs.    80
#> LL         -106.09
#> AIC         224.19
#> BIC         238.48
#> =====================
#> *** p < 0.001; ** p < 0.01; * p < 0.05
```

Let's estimate separate models for the different categories (recruitment x incentive).

```
> dat_yes_yes <- subset(dat, yes_yes == 1)
> fit <- lm(formula = y ~ x,
+           data = dat_yes_yes, weights = weight)
> m2 <- add_clustered(fit, cluster = dat_yes_yes$survey_id, type = "CR2")
>
> ## helper
> estimator <- function(dat, subset) {
+   dat_ <- subset(dat, subset = subset)
+   fit <- lm(formula = y ~ x,
+             data = dat_, weights = weight)
+   m <- add_clustered(fit, cluster = dat_$survey_id, type = "CR2")
+   m
+ }
>
> m3 <- estimator(dat, subset = (dat$yes_no == 1))
> m4 <- estimator(dat, subset = (dat$no_no == 1))
> m5 <- estimator(dat, subset = (dat$no_yes == 1))
```

Let's compare the results to the findings from the last publication. We essentially repeat the above steps but for a subset of the data.

```
> dat_last <- subset(dat[1:67, ], sample_size >= 10)
> fit <- lm(y ~ 0 + x + yes_yes + yes_no + no_no + no_yes,
+           data = dat_last, weights = weight)
> m1_last <- add_clustered(fit, cluster = dat_last$survey_id, type = "CR2")
> m2_last <- estimator(dat_last, subset = (dat_last$yes_yes == 1))
> m3_last <- estimator(dat_last, subset = (dat_last$yes_no == 1))
> m4_last <- estimator(dat_last, subset = (dat_last$no_no == 1))
> # m5_last (no no_yes combinations at that time)
```

We can easily create a regression table with **texreg**.

```
> m_last <- list(Pooled = m1_last,
+                `Yes, yes` = m2_last,
+                `Yes, no`= m3_last,
```

	Pooled	Yes, yes	Yes, no	No, no
x	-0.58^{***}	-0.32^{**}	-1.55	-1.25^{***}
	(0.17)	(0.11)	(1.16)	(0.31)
yes_yes	1.53^{***}			
	(0.20)			
yes_no	0.84^{***}			
	(0.15)			
no_no	-1.11^{***}			
	(0.20)			
(Intercept)		1.28^{***}	1.13^{**}	-0.81^{***}
		(0.18)	(0.32)	(0.20)
R^2	0.79	0.13	0.13	0.22
Adj. R^2	0.78	0.07	0.07	0.19
Num. obs.	66	18	18	30
LL	−69.99	−20.08	−13.94	−31.10
AIC	149.98	46.16	33.89	68.20
BIC	160.93	48.83	36.56	72.40

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 1: Old models

```
+                 `No, no` = m4_last)
> texreg::texreg(m_last, caption = "Old models")
```

```
> m <- list(Pooled = m1,
+           `Yes, yes` = m2,
+           `Yes, no` = m3,
+           `No, no` = m4,
+           `No, yes` = m5)
> texreg::texreg(m, caption = "New models")
```

Finally, we want to visualize our sample along with the model implied response rate curves.

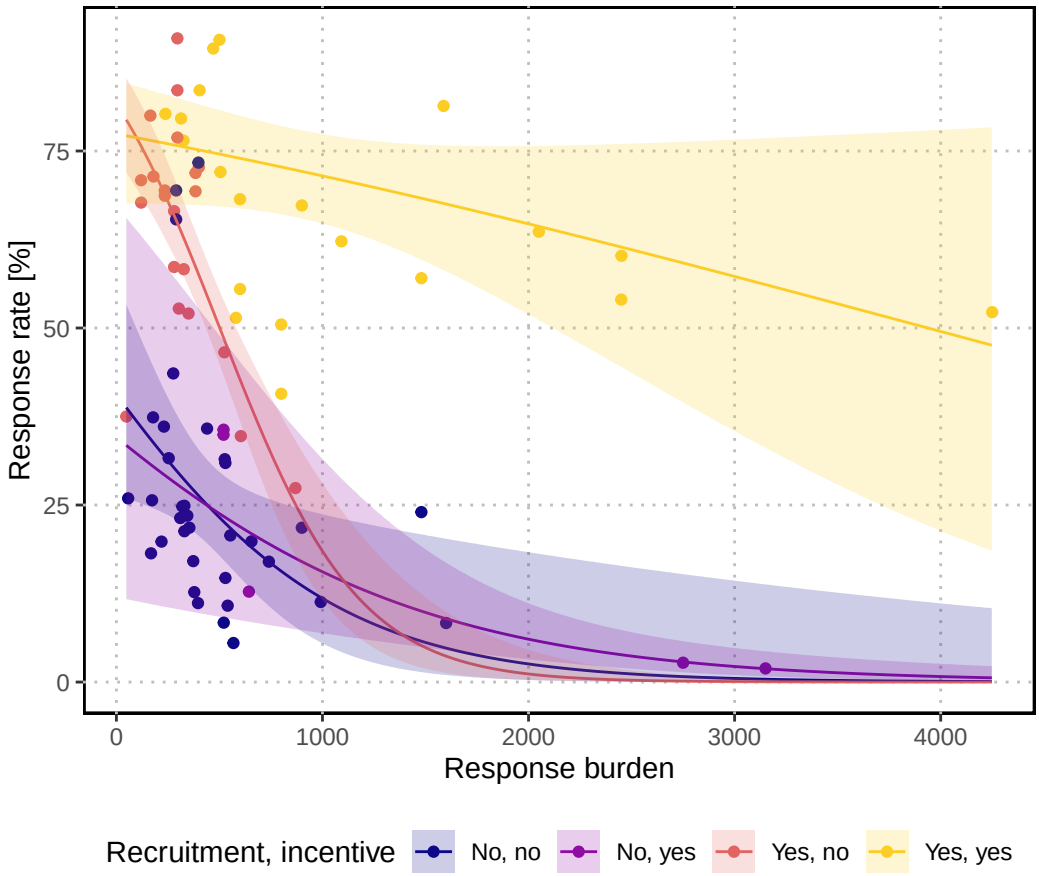
```
> new_x <- seq(min(dat$x), max(dat$x), by = 0.01)
> newdata <- data.frame(x = new_x)
>
> m_ <- m[2:length(m)]
> p <-
+   map2(m_, names(m_), function(x, y) {
+     p <- as.data.frame(predict(x, newdata = newdata, interval = "confidence"))
+     p$name <- y
+     p
+   })
> df <- reduce(p, rbind)
>
> ## backtransform (undo the logit)
> df <-
+   df %>%
+   mutate(across(c(fit, lwr, upr), function(x) backtransform(x)),
```

	Pooled	Yes, yes	Yes, no	No, no	No, yes
x	-1.01^{***}	-0.31^{**}	-2.98^{***}	-1.63^{**}	-1.05
	(0.19)	(0.11)	(0.71)	(0.47)	(0.40)
yes_yes	1.89^{***}				
	(0.25)				
yes_no	0.63				
	(0.33)				
no_no	-0.64				
	(0.34)				
no_yes	-0.73				
	(0.57)				
(Intercept)		1.23^{***}	1.49^{***}	-0.38	-0.64
		(0.16)	(0.25)	(0.41)	(0.74)
R^2	0.81	0.13	0.72	0.16	0.92
Adj. R^2	0.80	0.08	0.70	0.13	0.90
Num. obs.	80	20	21	34	5
LL	-106.09	-20.97	-17.01	-47.70	-2.48
AIC	224.19	47.94	40.02	101.39	10.97
BIC	238.48	50.92	43.16	105.97	9.80

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 2: New models

```
+          x = rep(1000 * newdata$x, 4))
>
> survey_data <-
+   dat %>%
+   mutate(yb = backtransform(y),
+          xb = x * 1000) %>%
+   pivot_longer(c(yes_yes, yes_no, no_no, no_yes)) %>%
+   mutate(name = case_when(name == "yes_yes" ~ "Yes, yes",
+                           name == "yes_no" ~ "Yes, no",
+                           name == "no_no" ~ "No, no",
+                           name == "no_yes" ~ "No, yes")) %>%
+   filter(value == 1)
>
> option <- "C"
> df %>%
+   ggplot(aes(x = x, group = name)) +
+   geom_point(aes(x = xb, y = yb, col = name), data = survey_data) +
+   geom_line(aes(y = fit, col = name)) +
+   geom_ribbon(aes(ymin = lwr, ymax = upr, fill = name), alpha = 0.2) +
+   scale_fill_viridis_d(option = option, end = 0.9) +
+   scale_color_viridis_d(option = option, end = 0.9) +
+   labs(x = "Response burden", y = "Response rate [%]",
+        col = "Recruitment, incentive", fill = "Recruitment, incentive") +
+   Heimisc::my_theme() +
+   Heimisc::add_grid() +
+   theme(legend.position = "bottom")
```



Despite drawing a line through such a point cloud is maybe a little critical and the sparsity of surveys with high response burdens (the ones we have are essentially outliers and influence the curve dramatically!), there is some indication, that any survey beyond 2000 points is just to burdensome for respondents. However, if they were recruited (i.e. agreed to participate) it looks different: Interestingly, there, the incentive seems absolutely essential. Further, the incentive flattens the curve - it will be interesting to compare the slopes of the incentive groups, once we have more data for surveys without a recruitment but with an incentive. However, anecdotal evidence from the TimeUse+ study suggests, that the incentive was important (see pre-test Caro; TRB24?).