Imperial College London

Ridit splines with applications to propensity weighting

Roger B. Newson
r.newson@imperial.ac.uk
http://www.imperial.ac.uk/nhli/r.newson/

Department of Primary Care and Public Health, Imperial College London

23rd UK Stata Users' Group Meeting, 7–8 September, 2017
Downloadable from the conference website at
http://ideas.repec.org/s/boc/usug17.html

What are ridits?

- The distribution of a random variable X can be specified by its **Bross ridit function**[2] $R_X(\cdot)$, defined by the formula

$$R_X(x) \;=\; \Pr(X < x) \;+\; \frac{1}{2}\Pr(X = x).$$

- *So*, ridits are like ranks, but expressed on a scale from 0 (below the bottom–ranking value) to 1 (above the top–ranking value).
- The word was chosen to be like logit and probit, as the prefix stands for "with respect to an identified distribution".
- The **Brockett–Levene ridit function**[1] $R^*_X(\cdot)$ is defined (on a scale from -1 to 1) as a *difference* between probabilities,

$$R^*_X(x) \;=\; \Pr(X < x) \;-\; \Pr(X > x) \,,$$

and should *always* be used to calculate the Bross ridit function

$$R_X(x) \;=\; \frac{1}{2}\left[R^*_X(x) + 1\right] \,,$$

avoiding the precision problems of adding tiny half–probabilities to huge probabilities.

What are ridits?

- The distribution of a random variable X can be specified by its **Bross ridit function**[2] $R_X(\cdot)$, defined by the formula

$$R_X(x) \;=\; \Pr(X < x) \;+\; \frac{1}{2}\Pr(X = x).$$

- *So*, ridits are like ranks, but expressed on a scale from 0 (below the bottom–ranking value) to 1 (above the top–ranking value).
- The word was chosen to be like logit and probit, as the prefix stands for "with respect to an identified distribution".
- The **Brockett–Levene ridit function**[1] $R^*_X(\cdot)$ is defined (on a scale from -1 to 1) as a *difference* between probabilities,

$$R^*_X(x) \;=\; \Pr(X < x) \;-\; \Pr(X > x)\,,$$

and should *always* be used to calculate the Bross ridit function

$$R_X(x) \;=\; \frac{1}{2}\left[R^*_X(x) + 1\right]\,,$$

avoiding the precision problems of adding tiny half–probabilities to huge probabilities.

What are ridits?

- The distribution of a random variable X can be specified by its **Bross ridit function**[2] $R_X(\cdot)$, defined by the formula

$$R_X(x) \;=\; \Pr(X < x) \;+\; \frac{1}{2}\Pr(X = x).$$

- *So*, ridits are like ranks, but expressed on a scale from 0 (below the bottom–ranking value) to 1 (above the top–ranking value).
- The word was chosen to be like logit and probit, as the prefix stands for "with respect to an identified distribution".
- The **Brockett–Levene ridit function**[1] $R^*_X(\cdot)$ is defined (on a scale from -1 to 1) as a *difference* between probabilities,

$$R^*_X(x) \;=\; \Pr(X < x) \;-\; \Pr(X > x)\,,$$

and should *always* be used to calculate the Bross ridit function

$$R_X(x) \;=\; \frac{1}{2}\left[R^*_X(x) + 1\right]\,,$$

avoiding the precision problems of adding tiny half–probabilities to huge probabilities.

What are ridits?

- The distribution of a random variable X can be specified by its **Bross ridit function**[2] $R_X(\cdot)$, defined by the formula

$$R_X(x) \;=\; \Pr(X < x) \;+\; \frac{1}{2}\Pr(X = x).$$

- *So*, ridits are like ranks, but expressed on a scale from 0 (below the bottom–ranking value) to 1 (above the top–ranking value).
- The word was chosen to be like logit and probit, as the prefix stands for "with <u>r</u>espect to an <u>i</u>dentified <u>d</u>istribution".
- The **Brockett–Levene ridit function**[1] $R^*_X(\cdot)$ is defined (on a scale from -1 to 1) as a *difference* between probabilities,

$$R^*_X(x) \;=\; \Pr(X < x) \;-\; \Pr(X > x)\,,$$

and should *always* be used to calculate the Bross ridit function

$$R_X(x) \;=\; \frac{1}{2}\left[R^*_X(x) + 1\right]\,,$$

avoiding the precision problems of adding tiny half–probabilities to huge probabilities.

What are ridits?

- The distribution of a random variable X can be specified by its **Bross ridit function**[2] $R_X(\cdot)$, defined by the formula

$$R_X(x) \ = \ \Pr(X < x) \ + \ \frac{1}{2}\Pr(X = x).$$

- *So*, ridits are like ranks, but expressed on a scale from 0 (below the bottom–ranking value) to 1 (above the top–ranking value).
- The word was chosen to be like logit and probit, as the prefix stands for "with <u>r</u>espect to an <u>i</u>dentified <u>d</u>istribution".
- The **Brockett–Levene ridit function**[1] $R_X^*(\cdot)$ is defined (on a scale from -1 to 1) as a *difference* between probabilities,

$$R_X^*(x) \ = \ \Pr(X < x) \ - \Pr(X > x)\,,$$

  and should *always* be used to calculate the Bross ridit function

$$R_X(x) \ = \ \frac{1}{2}\left[R_X^*(x) + 1\right]\,,$$

  avoiding the precision problems of adding tiny half–probabilities to huge probabilities.

Computing ridits using the `wridit` package

- The SSC package `wridit` computes "folded" Brockett–Levene ridits or "unfolded" Bross ridits for a numeric Stata variable.
- These ridits may be on a reverse scale (using the `reverse` option) and/or on a percentage scale (using the `percent` option), as with the `ridit` module of Nick Cox's `egenmore`.
- *However*, `wridit` also allows weights, so the ridits can be with respect to the distribution of the variable in a **target population**.
- In particular, zero weights are allowed, so the user can define ridits for the zero–weighted observations with respect to the distribution of the variable in the nonzero–weighted observations.
- *For instance*, in the `auto` data, we can define ridits of `length` with respect to the length distribution in US cars by zero–weighting non–US cars, or *vice versa*.
- On the default Bross scale, these ridits may be 0 in the former case for non–US cars, or 1 in the latter case for US cars.

Computing ridits using the `wridit` package

- The SSC package `wridit` computes "folded" Brockett–Levene ridits or "unfolded" Bross ridits for a numeric Stata variable.
- These ridits may be on a reverse scale (using the `reverse` option) and/or on a percentage scale (using the `percent` option), as with the `ridit` module of Nick Cox's `egenmore`.
- *However*, `wridit` also allows weights, so the ridits can be with respect to the distribution of the variable in a **target population**.
- In particular, zero weights are allowed, so the user can define ridits for the zero–weighted observations with respect to the distribution of the variable in the nonzero–weighted observations.
- *For instance*, in the `auto` data, we can define ridits of `length` with respect to the length distribution in US cars by zero–weighting non–US cars, or *vice versa*.
- On the default Bross scale, these ridits may be 0 in the former case for non–US cars, or 1 in the latter case for US cars.

Computing ridits using the `wridit` package

- The SSC package `wridit` computes "folded" Brockett–Levene ridits or "unfolded" Bross ridits for a numeric Stata variable.
- These ridits may be on a reverse scale (using the `reverse` option) and/or on a percentage scale (using the `percent` option), as with the `ridit` module of Nick Cox's `egenmore`.
- *However*, `wridit` also allows weights, so the ridits can be with respect to the distribution of the variable in a **target population**.
- In particular, zero weights are allowed, so the user can define ridits for the zero–weighted observations with respect to the distribution of the variable in the nonzero–weighted observations.
- *For instance*, in the `auto` data, we can define ridits of `length` with respect to the length distribution in US cars by zero–weighting non–US cars, or *vice versa*.
- On the default Bross scale, these ridits may be 0 in the former case for non–US cars, or 1 in the latter case for US cars.

Computing ridits using the `wridit` package

- The SSC package `wridit` computes "folded" Brockett–Levene ridits or "unfolded" Bross ridits for a numeric Stata variable.
- These ridits may be on a reverse scale (using the `reverse` option) and/or on a percentage scale (using the `percent` option), as with the `ridit` module of Nick Cox's `egenmore`.
- *However*, `wridit` also allows weights, so the ridits can be with respect to the distribution of the variable in a **target population**.
- In particular, zero weights are allowed, so the user can define ridits for the zero–weighted observations with respect to the distribution of the variable in the nonzero–weighted observations.
- *For instance*, in the `auto` data, we can define ridits of `length` with respect to the length distribution in US cars by zero–weighting non–US cars, or *vice versa*.
- On the default Bross scale, these ridits may be 0 in the former case for non–US cars, or 1 in the latter case for US cars.

Computing ridits using the `wridit` package

- The SSC package `wridit` computes "folded" Brockett–Levene ridits or "unfolded" Bross ridits for a numeric Stata variable.
- These ridits may be on a reverse scale (using the `reverse` option) and/or on a percentage scale (using the `percent` option), as with the `ridit` module of Nick Cox's `egenmore`.
- *However*, `wridit` also allows weights, so the ridits can be with respect to the distribution of the variable in a **target population**.
- In particular, zero weights are allowed, so the user can define ridits for the zero–weighted observations with respect to the distribution of the variable in the nonzero–weighted observations.
- *For instance*, in the `auto` data, we can define ridits of `length` with respect to the length distribution in US cars by zero–weighting non–US cars, or *vice versa*.
- On the default Bross scale, these ridits may be 0 in the former case for non–US cars, or 1 in the latter case for US cars.

Computing ridits using the `wridit` package

- The SSC package `wridit` computes "folded" Brockett–Levene ridits or "unfolded" Bross ridits for a numeric Stata variable.
- These ridits may be on a reverse scale (using the `reverse` option) and/or on a percentage scale (using the `percent` option), as with the `ridit` module of Nick Cox's `egenmore`.
- *However*, `wridit` also allows weights, so the ridits can be with respect to the distribution of the variable in a **target population**.
- In particular, zero weights are allowed, so the user can define ridits for the zero–weighted observations with respect to the distribution of the variable in the nonzero–weighted observations.
- *For instance*, in the `auto` data, we can define ridits of `length` with respect to the length distribution in US cars by zero–weighting non–US cars, or *vice versa*.
- On the default Bross scale, these ridits may be 0 in the former case for non–US cars, or 1 in the latter case for US cars.

Computing ridits using the `wridit` package

- The SSC package `wridit` computes "folded" Brockett–Levene ridits or "unfolded" Bross ridits for a numeric Stata variable.
- These ridits may be on a reverse scale (using the `reverse` option) and/or on a percentage scale (using the `percent` option), as with the `ridit` module of Nick Cox's `egenmore`.
- *However*, `wridit` also allows weights, so the ridits can be with respect to the distribution of the variable in a **target population**.
- In particular, zero weights are allowed, so the user can define ridits for the zero–weighted observations with respect to the distribution of the variable in the nonzero–weighted observations.
- *For instance*, in the `auto` data, we can define ridits of `length` with respect to the length distribution in US cars by zero–weighting non–US cars, or *vice versa*.
- On the default Bross scale, these ridits may be 0 in the former case for non–US cars, or 1 in the latter case for US cars.

What are ridit splines?

- A **ridit spline** in a variable X is a spline in the ridit–transformed variable $R_X(X)$.
- If the user has installed the SSC packages `bspline`[3] and `polyspline`[4] as well as `wridit`, then the user can compute an unrestricted **reference–spline basis** in the ridit of an X–variable.
- This spline basis will have the advantage that the corresponding parameters of a fitted model will be values of the ridit spline at a list of values on the ridit scale, ranging from 0 to 1 (such as 0, 0.25, 0.50, 0.75 and 1).
- These fitted parameters will be mean values of the outcome variable, corresponding to X–values equal to percentiles of X (such as the minimum, median, maximum, and 25th and 75th percentiles).
- This is because percentiles are defined as inverse ridits.

What are ridit splines?

- A **ridit spline** in a variable X is a spline in the ridit–transformed variable $R_X(X)$.
- If the user has installed the SSC packages `bspline`[3] and `polyspline`[4] as well as `wridit`, then the user can compute an unrestricted **reference–spline basis** in the ridit of an X–variable.
- This spline basis will have the advantage that the corresponding parameters of a fitted model will be values of the ridit spline at a list of values on the ridit scale, ranging from 0 to 1 (such as 0, 0.25, 0.50, 0.75 and 1).
- These fitted parameters will be mean values of the outcome variable, corresponding to X–values equal to percentiles of X (such as the minimum, median, maximum, and 25th and 75th percentiles).
- This is because percentiles are defined as inverse ridits.

What are ridit splines?

- A **ridit spline** in a variable X is a spline in the ridit–transformed variable $R_X(X)$.
- If the user has installed the SSC packages `bspline`[3] and `polyspline`[4] as well as `wridit`, then the user can compute an unrestricted **reference–spline basis** in the ridit of an X–variable.
- This spline basis will have the advantage that the corresponding parameters of a fitted model will be values of the ridit spline at a list of values on the ridit scale, ranging from 0 to 1 (such as 0, 0.25, 0.50, 0.75 and 1).
- These fitted parameters will be mean values of the outcome variable, corresponding to X–values equal to percentiles of X (such as the minimum, median, maximum, and 25th and 75th percentiles).
- This is because percentiles are defined as inverse ridits.

What are ridit splines?

- A **ridit spline** in a variable X is a spline in the ridit–transformed variable $R_X(X)$.
- If the user has installed the SSC packages `bspline`[3] and `polyspline`[4] as well as `wridit`, then the user can compute an unrestricted **reference–spline basis** in the ridit of an X–variable.
- This spline basis will have the advantage that the corresponding parameters of a fitted model will be values of the ridit spline at a list of values on the ridit scale, ranging from 0 to 1 (such as 0, 0.25, 0.50, 0.75 and 1).
- These fitted parameters will be mean values of the outcome variable, corresponding to X–values equal to percentiles of X (such as the minimum, median, maximum, and 25th and 75th percentiles).
- This is because percentiles are defined as inverse ridits.

What are ridit splines?

- A **ridit spline** in a variable X is a spline in the ridit–transformed variable $R_X(X)$.
- If the user has installed the SSC packages `bspline`[3] and `polyspline`[4] as well as `wridit`, then the user can compute an unrestricted **reference–spline basis** in the ridit of an X–variable.
- This spline basis will have the advantage that the corresponding parameters of a fitted model will be values of the ridit spline at a list of values on the ridit scale, ranging from 0 to 1 (such as 0, 0.25, 0.50, 0.75 and 1).
- These fitted parameters will be mean values of the outcome variable, corresponding to X–values equal to percentiles of X (such as the minimum, median, maximum, and 25th and 75th percentiles).
- This is because percentiles are defined as inverse ridits.

What are ridit splines?

- A **ridit spline** in a variable X is a spline in the ridit–transformed variable $R_X(X)$.
- If the user has installed the SSC packages `bspline`[3] and `polyspline`[4] as well as `wridit`, then the user can compute an unrestricted **reference–spline basis** in the ridit of an X–variable.
- This spline basis will have the advantage that the corresponding parameters of a fitted model will be values of the ridit spline at a list of values on the ridit scale, ranging from 0 to 1 (such as 0, 0.25, 0.50, 0.75 and 1).
- These fitted parameters will be mean values of the outcome variable, corresponding to X–values equal to percentiles of X (such as the minimum, median, maximum, and 25th and 75th percentiles).
- This is because percentiles are defined as inverse ridits.

Example: Mileage and car length in the `auto` data

- We will demonstrate our methods in the `auto` data, with 1 observation for each of 74 car models.
- We will regress fuel efficiency in US/Imperial miles per gallon with respect to a ridit spline in car length in US/Imperial inches.
- We will use `wridit` to define the ridits of car length, and `polyspline`[4] to define an unrestricted cubic reference–spline basis in the ridits.
- We will then use `rcentile`[4] to estimate the percentiles corresponding to the reference ridits.
- We will then fit the regression model for fuel efficiency with respect to car length, with 1 parameter for each of 5 length percentiles (0, 25, 50, 75 and 100).
- Finally, we will plot the results.

Example: Mileage and car length in the `auto` data

- We will demonstrate our methods in the `auto` data, with 1 observation for each of 74 car models.
- We will regress fuel efficiency in US/Imperial miles per gallon with respect to a ridit spline in car length in US/Imperial inches.
- We will use `wridit` to define the ridits of car length, and `polyspline`[4] to define an unrestricted cubic reference–spline basis in the ridits.
- We will then use `rcentile`[4] to estimate the percentiles corresponding to the reference ridits.
- We will then fit the regression model for fuel efficiency with respect to car length, with 1 parameter for each of 5 length percentiles (0, 25, 50, 75 and 100).
- Finally, we will plot the results.

Example: Mileage and car length in the `auto` data

- We will demonstrate our methods in the `auto` data, with 1 observation for each of 74 car models.
- We will regress fuel efficiency in US/Imperial miles per gallon with respect to a ridit spline in car length in US/Imperial inches.
- We will use `wridit` to define the ridits of car length, and `polyspline`[4] to define an unrestricted cubic reference–spline basis in the ridits.
- We will then use `rcentile`[4] to estimate the percentiles corresponding to the reference ridits.
- We will then fit the regression model for fuel efficiency with respect to car length, with 1 parameter for each of 5 length percentiles (0, 25, 50, 75 and 100).
- Finally, we will plot the results.

Example: Mileage and car length in the `auto` data

- We will demonstrate our methods in the `auto` data, with 1 observation for each of 74 car models.
- We will regress fuel efficiency in US/Imperial miles per gallon with respect to a ridit spline in car length in US/Imperial inches.
- We will use `wridit` to define the ridits of car length, and `polyspline`[4] to define an unrestricted cubic reference–spline basis in the ridits.
- We will then use `rcentile`[4] to estimate the percentiles corresponding to the reference ridits.
- We will then fit the regression model for fuel efficiency with respect to car length, with 1 parameter for each of 5 length percentiles (0, 25, 50, 75 and 100).
- Finally, we will plot the results.

Example: Mileage and car length in the auto data

- We will demonstrate our methods in the auto data, with 1 observation for each of 74 car models.
- We will regress fuel efficiency in US/Imperial miles per gallon with respect to a ridit spline in car length in US/Imperial inches.
- We will use wridit to define the ridits of car length, and polyspline[4] to define an unrestricted cubic reference–spline basis in the ridits.
- We will then use rcentile[4] to estimate the percentiles corresponding to the reference ridits.
- We will then fit the regression model for fuel efficiency with respect to car length, with 1 parameter for each of 5 length percentiles (0, 25, 50, 75 and 100).
- Finally, we will plot the results.

Example: Mileage and car length in the auto data

- We will demonstrate our methods in the auto data, with 1 observation for each of 74 car models.
- We will regress fuel efficiency in US/Imperial miles per gallon with respect to a ridit spline in car length in US/Imperial inches.
- We will use wridit to define the ridits of car length, and polyspline[4] to define an unrestricted cubic reference–spline basis in the ridits.
- We will then use rcentile[4] to estimate the percentiles corresponding to the reference ridits.
- We will then fit the regression model for fuel efficiency with respect to car length, with 1 parameter for each of 5 length percentiles (0, 25, 50, 75 and 100).
- Finally, we will plot the results.

Example: Mileage and car length in the `auto` data

- We will demonstrate our methods in the `auto` data, with 1 observation for each of 74 car models.
- We will regress fuel efficiency in US/Imperial miles per gallon with respect to a ridit spline in car length in US/Imperial inches.
- We will use `wridit` to define the ridits of car length, and `polyspline`[4] to define an unrestricted cubic reference–spline basis in the ridits.
- We will then use `rcentile`[4] to estimate the percentiles corresponding to the reference ridits.
- We will then fit the regression model for fuel efficiency with respect to car length, with 1 parameter for each of 5 length percentiles (0, 25, 50, 75 and 100).
- Finally, we will plot the results.

Computing ridits using `wridit`

After loading the `auto` data, we use `wridit` to generate a new variable `lengthridit`, containing ridits (on a percentage scale) for the variable `length`:

```
. wridit length, percent generate(lengthridit);

. lab var lengthridit "Ridit (%) of Length (in.)";

. desc lengthridit, fu;

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------------------------------
lengthridit     double  %10.0g                Ridit (%) of Length (in.)

. summ lengthridit;

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
 lengthridit |         74          50    29.04986   .6756757   99.32432
```

Note that the Bross ridits (on a percentage scale) are *strictly* bounded between 0 and 100 percent, and have a mean of *exactly* 50 percent.

Computing a cubic ridit spine basis in `length`

We use the SSC package `polyspline`[4] to generate a basis of 5 cubic reference splines `rs_1` to `rs_5` in the ridit variable, corresponding to percentages of 0, 25, 50, 75 and 100, respectively:

```
. polyspline lengthridit, power(3) refpts(0(25)100) gene(rs_) labprefix(Percent@);
5 reference splines generated of degree: 3

. desc rs_*, fu;

              storage   display    value
variable name   type    format     label      variable label
------------------------------------------------------------------------------------
rs_1            float   %8.4f                 Percent@0
rs_2            float   %8.4f                 Percent@25
rs_3            float   %8.4f                 Percent@50
rs_4            float   %8.4f                 Percent@75
rs_5            float   %8.4f                 Percent@100
```

Note that we have labelled them using the `labprefix()` option of `polyspline`.

Percentiles corresponding to the 5 reference percentage ridits

To estimate the inverse ridits (also known as percentiles) corresponding to our 5 reference percentage ridits, we use the SSC package `rcentile`[4] to compute percentile car lengths in inches:

```
. rcentile length, centile(0(25)100) transf(asin);
Percentile(s) for variable: length
Mean sign transformation: Daniels' arcsine
Valid observations: 74
95% confidence interval(s) for percentile(s)
  Percent    Centile    Minimum    Maximum
        0        142  -9.0e+307        142
       25        170        164        174
       50      192.5        179        198
       75        204        200        212
      100        233        233   9.0e+307
```

Percentiles 0 and 100 are estimated as the minimum and maximum lengths, respectively, with lower and upper confidence limits (respectively) equal to minus and plus infinity (respectively). *However*, we are not really interested in confidence limits here, because. . .

Mean mileages corresponding to the 5 reference percentage ridits

... `length` is the *X*–variable, and we are really interested in the *conditional* means of the *Y*–variable `mpg`, corresponding to our 5 *sample* percentile lengths. We estimate these using `regress`:

```
. regress mpg rs_*, noconst vce(robust);

Linear regression                               Number of obs     =         74
                                                F(5, 69)          =     757.73
                                                Prob > F          =     0.0000
                                                R-squared         =     0.9778
                                                Root MSE          =     3.4072

------------------------------------------------------------------------------
             |               Robust
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        rs_1 |    29.2563    2.17573    13.45   0.000     24.91584    33.59677
        rs_2 |   25.66597   .9778877    26.25   0.000     23.71514     27.6168
        rs_3 |   19.43958   .6659589    29.19   0.000     18.11103    20.76813
        rs_4 |   18.01778   .5218036    34.53   0.000     16.97681    19.05875
        rs_5 |   12.68334   1.043106    12.16   0.000      10.6024    14.76427
------------------------------------------------------------------------------
```

These estimates and confidence limits are expressed in miles per gallon, and in an alien–looking format. *However* . . .

Percentile lengths and mean mileages corresponding to the 5 reference percentage ridits

. . . if we collect the percentiles in an output dataset (or resultsset) using `xsvmat`, and collect the estimated mean mileages in a second resultsset using `parmest`, and reconstruct the `Percent` variable in the second resultsset using `factext`, and merge the 2 resultssets by `Percent` to form a single resultsset in memory, then we can list the percents, percentile lengths, and conditional mean mileages as follows:

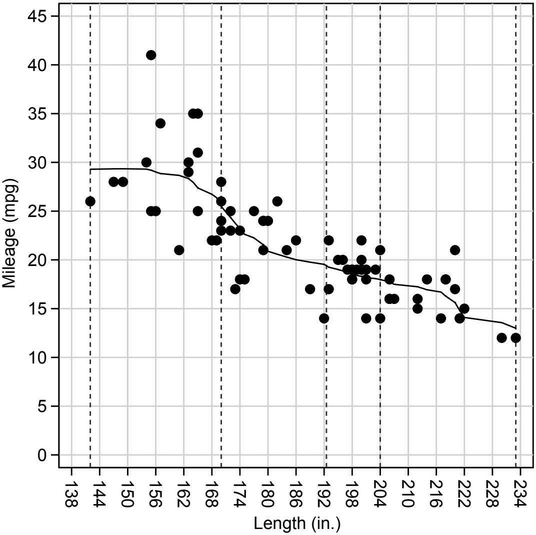
```
. list Percent Centile parm estimate min* max*, abbr(32);

     +------------------------------------------------------+
     | Percent   Centile   parm   estimate   min95   max95 |
     |------------------------------------------------------|
  1. |       0       142   rs_1      29.26   24.92   33.60 |
  2. |      25       170   rs_2      25.67   23.72   27.62 |
  3. |      50     192.5   rs_3      19.44   18.11   20.77 |
  4. |      75       204   rs_4      18.02   16.98   19.06 |
  5. |     100       233   rs_5      12.68   10.60   14.76 |
     +------------------------------------------------------+
```

This format is easier to understand. *However* . . .

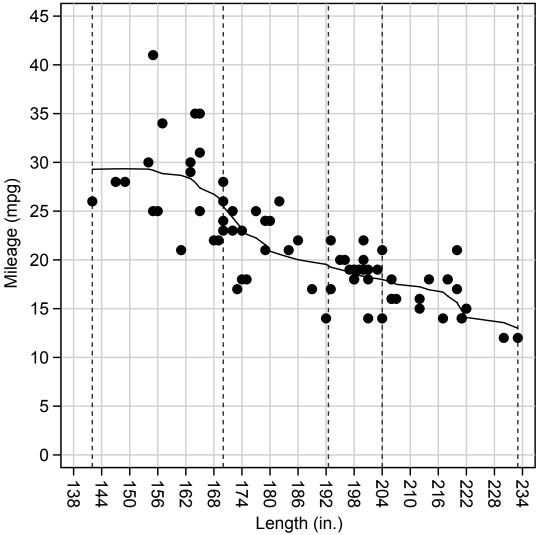
Plot of fitted and observed car mileages against car length

- ... we can be even more informative if we `append` the resultsset to the original dataset and create some graphics.
- Here, we have scatter–plotted the observed mileages, and line–plotted the fitted mileages, against car length.
- The horizontal–axis reference lines show the positions of car length percentiles 0, 25, 50, 75 and 100.



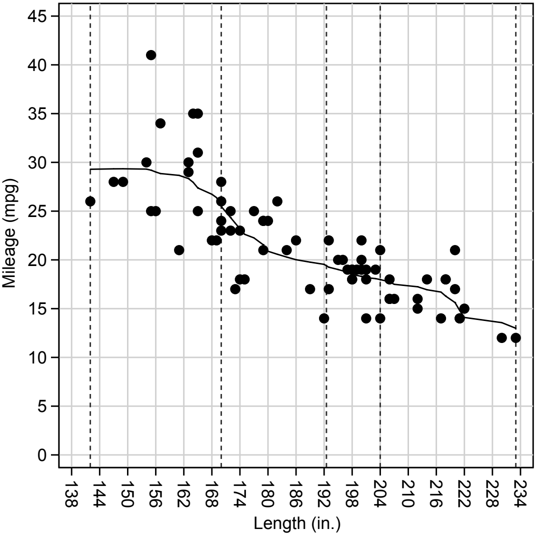
Plot of fitted and observed car mileages against car length

- . . . we can be even more informative if we `append` the resultsset to the original dataset and create some graphics.
- Here, we have scatter–plotted the observed mileages, and line–plotted the fitted mileages, against car length.
- The horizontal–axis reference lines show the positions of car length percentiles 0, 25, 50, 75 and 100.



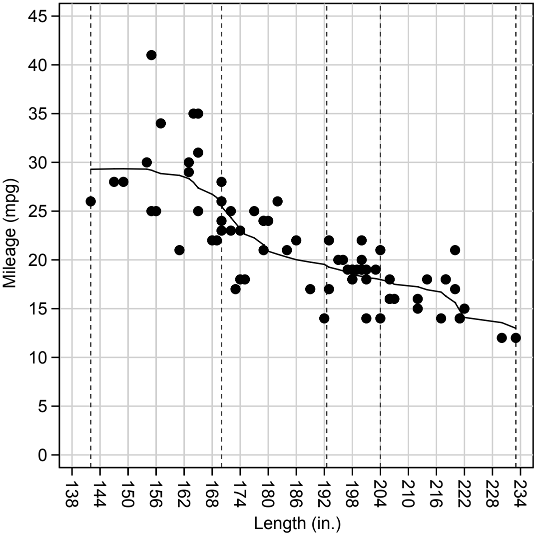
Plot of fitted and observed car mileages against car length

- ... we can be even more informative if we `append` the resultsset to the original dataset and create some graphics.
- Here, we have scatter–plotted the observed mileages, and line–plotted the fitted mileages, against car length.
- The horizontal–axis reference lines show the positions of car length percentiles 0, 25, 50, 75 and 100.



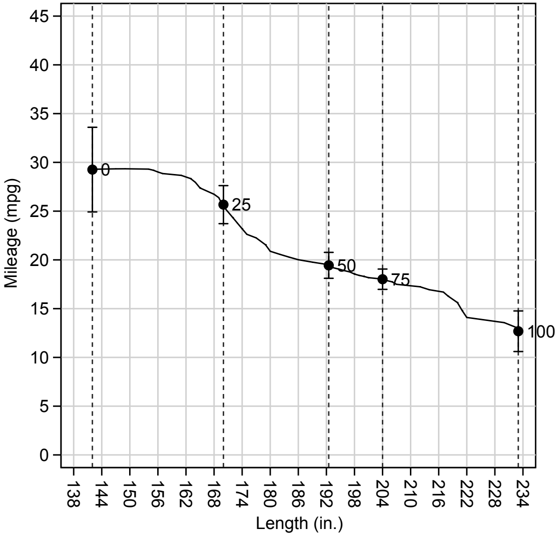
Plot of fitted and observed car mileages against car length

- . . . we can be even more informative if we `append` the resultsset to the original dataset and create some graphics.
- Here, we have scatter–plotted the observed mileages, and line–plotted the fitted mileages, against car length.
- The horizontal–axis reference lines show the positions of car length percentiles 0, 25, 50, 75 and 100.



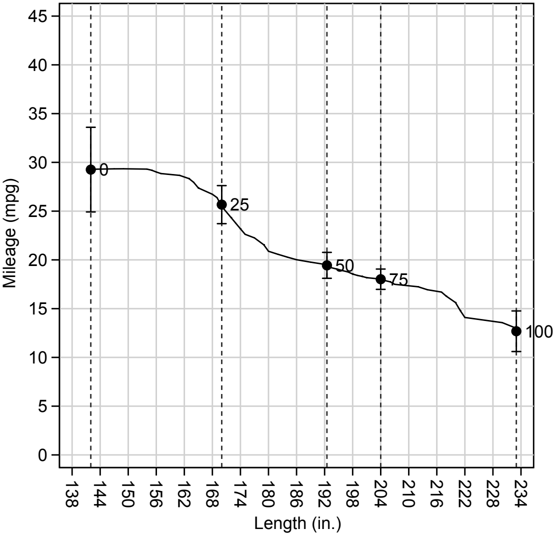
Plot of fitted and length–percentile mean car mileages against car length

- Alternatively, we can leave out the observed values, and show confidence intervals for the fitted values at the 5 car length percentiles, labelled with their percents.
- These are the fitted parameters of the ridit–spline model for mileage.
- Note that a ridit spline is less smooth than a spline, as a *sample* ridit function is non–smooth.



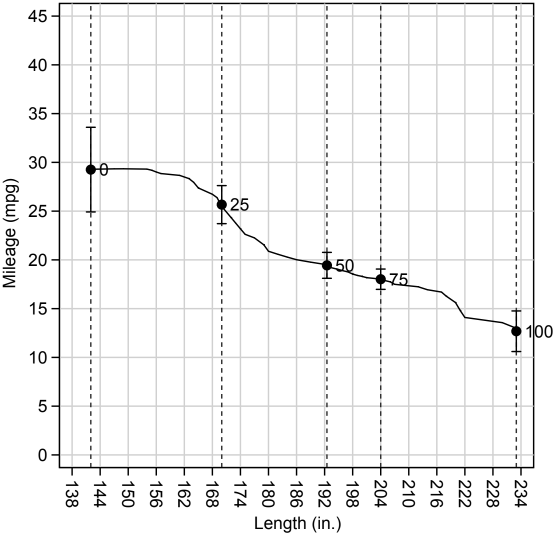
Plot of fitted and length–percentile mean car mileages against car length

- Alternatively, we can leave out the observed values, and show confidence intervals for the fitted values at the 5 car length percentiles, labelled with their percents.
- These are the fitted parameters of the ridit–spline model for mileage.
- Note that a ridit spline is less smooth than a spline, as a *sample* ridit function is non–smooth.



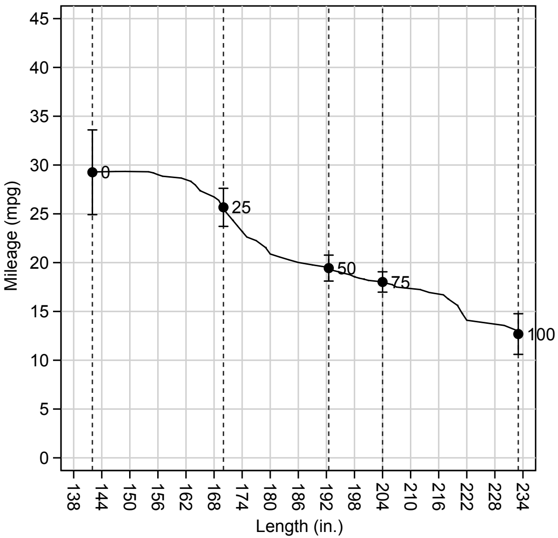
Plot of fitted and length–percentile mean car mileages against car length

- Alternatively, we can leave out the observed values, and show confidence intervals for the fitted values at the 5 car length percentiles, labelled with their percents.
- These are the fitted parameters of the ridit–spline model for mileage.
- Note that a ridit spline is less smooth than a spline, as a *sample* ridit function is non–smooth.



Plot of fitted and length–percentile mean car mileages against car length

- Alternatively, we can leave out the observed values, and show confidence intervals for the fitted values at the 5 car length percentiles, labelled with their percents.
- These are the fitted parameters of the ridit–spline model for mileage.
- Note that a ridit spline is less smooth than a spline, as a *sample* ridit function is non–smooth.



Application: Propensity weighting

- In an observational study, a **propensity score** typically measures the odds of a subject being allocated to Treatment A instead of to Treatment B.
- It is typically computed using a logit regression model of treatment allocation with respect to a list of **confounders**.
- The propensity score can then be used to calculate **propensity weights**.
- These are used to standardize directly from the sampled population to a fantasy **target population**, with a real–world distribution of confounders (and therefore of the propensity score), but with no treatment–confounder association.
- The **causal effect** of treatment allocation on an outcome is then estimated as the difference, in that fantasy target population, between the mean outcome for subjects on Treatment A and the mean outcome for subjects on Treatment B.

Application: Propensity weighting

- In an observational study, a **propensity score** typically measures the odds of a subject being allocated to Treatment A instead of to Treatment B.
- It is typically computed using a logit regression model of treatment allocation with respect to a list of **confounders**.
- The propensity score can then be used to calculate **propensity weights**.
- These are used to standardize directly from the sampled population to a fantasy **target population**, with a real–world distribution of confounders (and therefore of the propensity score), but with no treatment–confounder association.
- The **causal effect** of treatment allocation on an outcome is then estimated as the difference, in that fantasy target population, between the mean outcome for subjects on Treatment A and the mean outcome for subjects on Treatment B.

Application: Propensity weighting

- In an observational study, a **propensity score** typically measures the odds of a subject being allocated to Treatment *A* instead of to Treatment *B*.
- It is typically computed using a logit regression model of treatment allocation with respect to a list of **confounders**.
- The propensity score can then be used to calculate **propensity weights**.
- These are used to standardize directly from the sampled population to a fantasy **target population**, with a real–world distribution of confounders (and therefore of the propensity score), but with no treatment–confounder association.
- The **causal effect** of treatment allocation on an outcome is then estimated as the difference, in that fantasy target population, between the mean outcome for subjects on Treatment *A* and the mean outcome for subjects on Treatment *B*.

Application: Propensity weighting

- In an observational study, a **propensity score** typically measures the odds of a subject being allocated to Treatment A instead of to Treatment B.
- It is typically computed using a logit regression model of treatment allocation with respect to a list of **confounders**.
- The propensity score can then be used to calculate **propensity weights**.
- These are used to standardize directly from the sampled population to a fantasy **target population**, with a real–world distribution of confounders (and therefore of the propensity score), but with no treatment–confounder association.
- The **causal effect** of treatment allocation on an outcome is then estimated as the difference, in that fantasy target population, between the mean outcome for subjects on Treatment A and the mean outcome for subjects on Treatment B.

Application: Propensity weighting

- In an observational study, a **propensity score** typically measures the odds of a subject being allocated to Treatment *A* instead of to Treatment *B*.
- It is typically computed using a logit regression model of treatment allocation with respect to a list of **confounders**.
- The propensity score can then be used to calculate **propensity weights**.
- These are used to standardize directly from the sampled population to a fantasy **target population**, with a real–world distribution of confounders (and therefore of the propensity score), but with no treatment–confounder association.
- The **causal effect** of treatment allocation on an outcome is then estimated as the difference, in that fantasy target population, between the mean outcome for subjects on Treatment *A* and the mean outcome for subjects on Treatment *B*.

Application: Propensity weighting

- In an observational study, a **propensity score** typically measures the odds of a subject being allocated to Treatment A instead of to Treatment B.
- It is typically computed using a logit regression model of treatment allocation with respect to a list of **confounders**.
- The propensity score can then be used to calculate **propensity weights**.
- These are used to standardize directly from the sampled population to a fantasy **target population**, with a real–world distribution of confounders (and therefore of the propensity score), but with no treatment–confounder association.
- The **causal effect** of treatment allocation on an outcome is then estimated as the difference, in that fantasy target population, between the mean outcome for subjects on Treatment A and the mean outcome for subjects on Treatment B.

Problem: Outlying propensity weights

- *Unfortunately*, once the propensity weights are calculated from the model, we may find that some of these weights are *extremely* large.
- These weights belong to subjects with an *extremely* atypical confounder profile for the treatment group (A or B) to which they were allocated in the real world.
- Such outlying weights may imply that the propensity weights do not do a very good job of balancing out the confounders, and/or that the variance of the estimated causal effect is inflated.
- A possible solution is to compute a **secondary propensity score** (and a secondary propensity weight) from a second logit model, regressing treatment allocation with respect to a ridit spline in the primary propensity score.
- This secondary model might be less likely to generate outlying propensity weights than the primary model, as the ridit function is strictly bounded between 0 and 1.

Problem: Outlying propensity weights

- *Unfortunately*, once the propensity weights are calculated from the model, we may find that some of these weights are *extremely* large.
- These weights belong to subjects with an *extremely* atypical confounder profile for the treatment group (A or B) to which they were allocated in the real world.
- Such outlying weights may imply that the propensity weights do not do a very good job of balancing out the confounders, and/or that the variance of the estimated causal effect is inflated.
- A possible solution is to compute a **secondary propensity score** (and a secondary propensity weight) from a second logit model, regressing treatment allocation with respect to a ridit spline in the primary propensity score.
- This secondary model might be less likely to generate outlying propensity weights than the primary model, as the ridit function is strictly bounded between 0 and 1.

Problem: Outlying propensity weights

- *Unfortunately*, once the propensity weights are calculated from the model, we may find that some of these weights are *extremely* large.
- These weights belong to subjects with an *extremely* atypical confounder profile for the treatment group (A or B) to which they were allocated in the real world.
- Such outlying weights may imply that the propensity weights do not do a very good job of balancing out the confounders, and/or that the variance of the estimated causal effect is inflated.
- A possible solution is to compute a **secondary propensity score** (and a secondary propensity weight) from a second logit model, regressing treatment allocation with respect to a ridit spline in the primary propensity score.
- This secondary model might be less likely to generate outlying propensity weights than the primary model, as the ridit function is strictly bounded between 0 and 1.

Problem: Outlying propensity weights

- *Unfortunately*, once the propensity weights are calculated from the model, we may find that some of these weights are *extremely* large.
- These weights belong to subjects with an *extremely* atypical confounder profile for the treatment group (A or B) to which they were allocated in the real world.
- Such outlying weights may imply that the propensity weights do not do a very good job of balancing out the confounders, and/or that the variance of the estimated causal effect is inflated.
- A possible solution is to compute a **secondary propensity score** (and a secondary propensity weight) from a second logit model, regressing treatment allocation with respect to a ridit spline in the primary propensity score.
- This secondary model might be less likely to generate outlying propensity weights than the primary model, as the ridit function is strictly bounded between 0 and 1.

Problem: Outlying propensity weights

- *Unfortunately*, once the propensity weights are calculated from the model, we may find that some of these weights are *extremely* large.
- These weights belong to subjects with an *extremely* atypical confounder profile for the treatment group (A or B) to which they were allocated in the real world.
- Such outlying weights may imply that the propensity weights do not do a very good job of balancing out the confounders, and/or that the variance of the estimated causal effect is inflated.
- A possible solution is to compute a **secondary propensity score** (and a secondary propensity weight) from a second logit model, regressing treatment allocation with respect to a ridit spline in the primary propensity score.
- This secondary model might be less likely to generate outlying propensity weights than the primary model, as the ridit function is strictly bounded between 0 and 1.

Problem: Outlying propensity weights

- *Unfortunately*, once the propensity weights are calculated from the model, we may find that some of these weights are *extremely* large.
- These weights belong to subjects with an *extremely* atypical confounder profile for the treatment group (A or B) to which they were allocated in the real world.
- Such outlying weights may imply that the propensity weights do not do a very good job of balancing out the confounders, and/or that the variance of the estimated causal effect is inflated.
- A possible solution is to compute a **secondary propensity score** (and a secondary propensity weight) from a second logit model, regressing treatment allocation with respect to a ridit spline in the primary propensity score.
- This secondary model might be less likely to generate outlying propensity weights than the primary model, as the ridit function is strictly bounded between 0 and 1.

Example: Treatment effects on adverse event rates in Type 2 diabetics

- This example uses data from 2 British National Health Service databases, the Central Practice Research Datalink (CPRD) and the Hospital Episodes System (HES).
- We followed up 190,137 Type 2 diabetics in 490 English general practices, computing adverse event counts and 15 binary treatment indicators (9 prescribed drugs and 6 target achievements) for each of 10,135,062 patient–months.
- The aim was to assess the **average treatment effect in the treated (ATET)**, defined as a treated–untreated difference in adverse event counts per 1,000 patient–years.
- We used a list of patient–month–specific confounders to define a primary propensity score and propensity weight for each of the 15 treatment indicators, and also a secondary propensity score and propensity weight, using a logit model of the treatment with respect to a ridit spline in the primary propensity score.

Example: Treatment effects on adverse event rates in Type 2 diabetics

- This example uses data from 2 British National Health Service databases, the Central Practice Research Datalink (CPRD) and the Hospital Episodes System (HES).
- We followed up 190,137 Type 2 diabetics in 490 English general practices, computing adverse event counts and 15 binary treatment indicators (9 prescribed drugs and 6 target achievements) for each of 10,135,062 patient–months.
- The aim was to assess the **average treatment effect in the treated (ATET)**, defined as a treated–untreated difference in adverse event counts per 1,000 patient–years.
- We used a list of patient–month–specific confounders to define a primary propensity score and propensity weight for each of the 15 treatment indicators, and also a secondary propensity score and propensity weight, using a logit model of the treatment with respect to a ridit spline in the primary propensity score.

Example: Treatment effects on adverse event rates in Type 2 diabetics

- This example uses data from 2 British National Health Service databases, the Central Practice Research Datalink (CPRD) and the Hospital Episodes System (HES).
- We followed up 190,137 Type 2 diabetics in 490 English general practices, computing adverse event counts and 15 binary treatment indicators (9 prescribed drugs and 6 target achievements) for each of 10,135,062 patient–months.
- The aim was to assess the **average treatment effect in the treated (ATET)**, defined as a treated–untreated difference in adverse event counts per 1,000 patient–years.
- We used a list of patient–month–specific confounders to define a primary propensity score and propensity weight for each of the 15 treatment indicators, and also a secondary propensity score and propensity weight, using a logit model of the treatment with respect to a ridit spline in the primary propensity score.

Example: Treatment effects on adverse event rates in Type 2 diabetics

- This example uses data from 2 British National Health Service databases, the Central Practice Research Datalink (CPRD) and the Hospital Episodes System (HES).
- We followed up 190,137 Type 2 diabetics in 490 English general practices, computing adverse event counts and 15 binary treatment indicators (9 prescribed drugs and 6 target achievements) for each of 10,135,062 patient–months.
- The aim was to assess the **average treatment effect in the treated (ATET)**, defined as a treated–untreated difference in adverse event counts per 1,000 patient–years.
- We used a list of patient–month–specific confounders to define a primary propensity score and propensity weight for each of the 15 treatment indicators, and also a secondary propensity score and propensity weight, using a logit model of the treatment with respect to a ridit spline in the primary propensity score.

Example: Treatment effects on adverse event rates in Type 2 diabetics

- This example uses data from 2 British National Health Service databases, the Central Practice Research Datalink (CPRD) and the Hospital Episodes System (HES).
- We followed up 190,137 Type 2 diabetics in 490 English general practices, computing adverse event counts and 15 binary treatment indicators (9 prescribed drugs and 6 target achievements) for each of 10,135,062 patient–months.
- The aim was to assess the **average treatment effect in the treated (ATET)**, defined as a treated–untreated difference in adverse event counts per 1,000 patient–years.
- We used a list of patient–month–specific confounders to define a primary propensity score and propensity weight for each of the 15 treatment indicators, and also a secondary propensity score and propensity weight, using a logit model of the treatment with respect to a ridit spline in the primary propensity score.

Predictive power, balancing power and variance inflation checks

- To choose a propensity score for use in the final analysis, we used the methods of Newson (2016)[5].
- Predictive power was measured using the unweighted Somers' D of the propensity score with respect to the treatment indicator.
- Balancing power was measured using Somers' D of the propensity score with respect to the treatment indicator, weighted using the appropriate propensity weight.
- Costs of propensity weights were measured using variance and standard error (SE) inflation factors for the average treatment effect.

Predictive power, balancing power and variance inflation checks

- To choose a propensity score for use in the final analysis, we used the methods of Newson (2016)[5].
- Predictive power was measured using the unweighted Somers' D of the propensity score with respect to the treatment indicator.
- Balancing power was measured using Somers' D of the propensity score with respect to the treatment indicator, weighted using the appropriate propensity weight.
- Costs of propensity weights were measured using variance and standard error (SE) inflation factors for the average treatment effect.

Predictive power, balancing power and variance inflation checks

- To choose a propensity score for use in the final analysis, we used the methods of Newson (2016)[5].
- Predictive power was measured using the unweighted Somers' D of the propensity score with respect to the treatment indicator.
- Balancing power was measured using Somers' D of the propensity score with respect to the treatment indicator, weighted using the appropriate propensity weight.
- Costs of propensity weights were measured using variance and standard error (SE) inflation factors for the average treatment effect.

Predictive power, balancing power and variance inflation checks

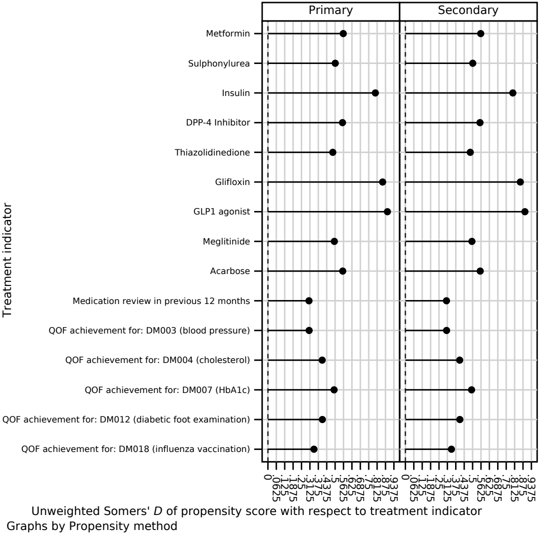
- To choose a propensity score for use in the final analysis, we used the methods of Newson (2016)[5].
- Predictive power was measured using the unweighted Somers' D of the propensity score with respect to the treatment indicator.
- Balancing power was measured using Somers' D of the propensity score with respect to the treatment indicator, weighted using the appropriate propensity weight.
- Costs of propensity weights were measured using variance and standard error (SE) inflation factors for the average treatment effect.

Predictive power, balancing power and variance inflation checks

- To choose a propensity score for use in the final analysis, we used the methods of Newson (2016)[5].
- Predictive power was measured using the unweighted Somers' *D* of the propensity score with respect to the treatment indicator.
- Balancing power was measured using Somers' *D* of the propensity score with respect to the treatment indicator, weighted using the appropriate propensity weight.
- Costs of propensity weights were measured using variance and standard error (SE) inflation factors for the average treatment effect.

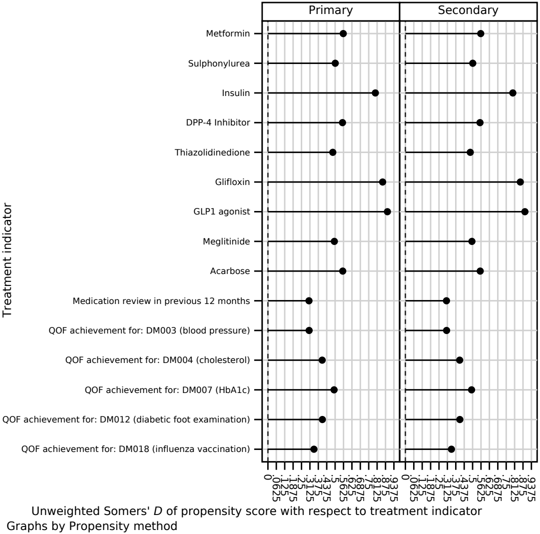
Unweighted Somers' *D* of propensity scores with respect to treatments

- The unweighted Somers' *D* values measure the power of propensity scores to predict the 15 treatments.
- The left and right panels show them for primary and secondary propensity scores, respectively.
- Values for the same treatment are practically identical between the two propensity methods.



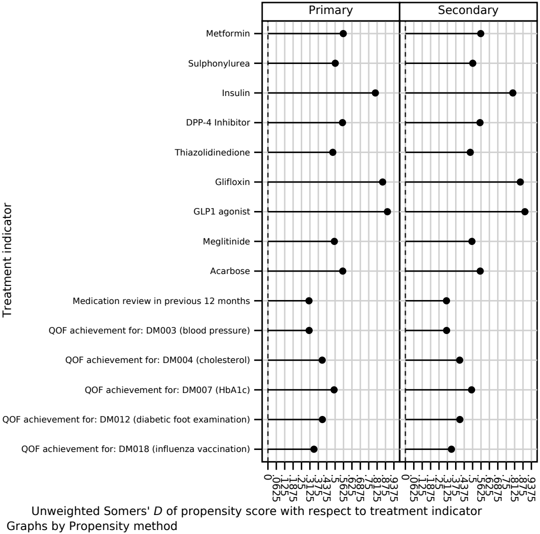
Unweighted Somers' *D* of propensity scores with respect to treatments

- The unweighted Somers' *D* values measure the power of propensity scores to predict the 15 treatments.
- The left and right panels show them for primary and secondary propensity scores, respectively.
- Values for the same treatment are practically identical between the two propensity methods.



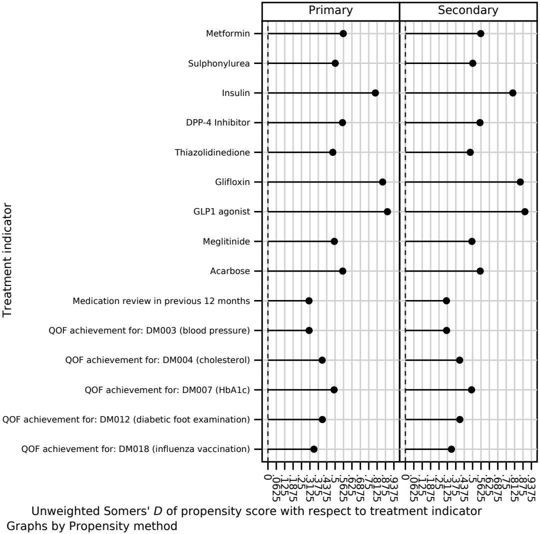
Unweighted Somers' D of propensity scores with respect to treatments

- The unweighted Somers' D values measure the power of propensity scores to predict the 15 treatments.
- The left and right panels show them for primary and secondary propensity scores, respectively.
- Values for the same treatment are practically identical between the two propensity methods.



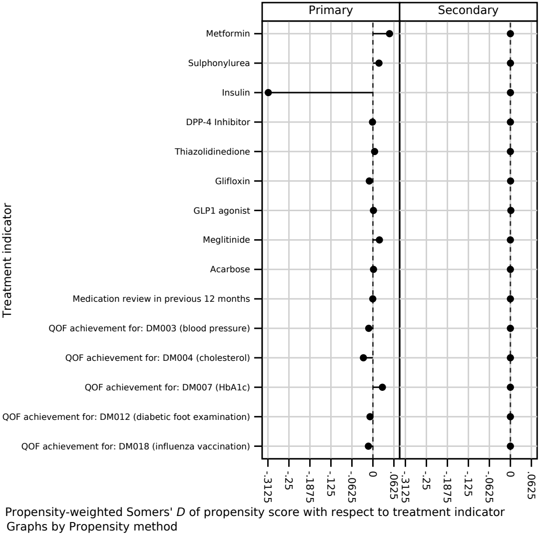
Unweighted Somers' *D* of propensity scores with respect to treatments

- The unweighted Somers' *D* values measure the power of propensity scores to predict the 15 treatments.
- The left and right panels show them for primary and secondary propensity scores, respectively.
- Values for the same treatment are practically identical between the two propensity methods.



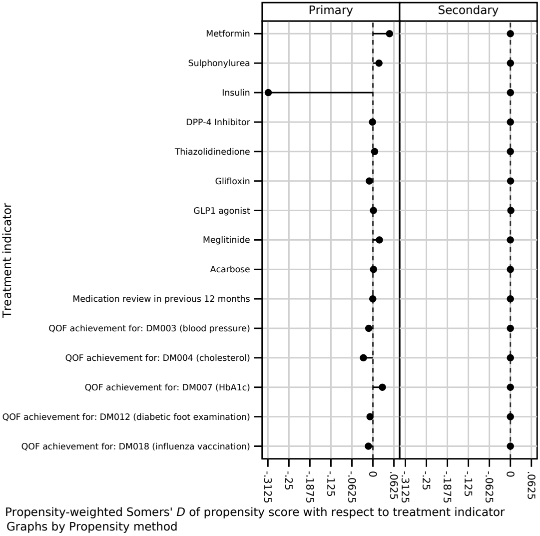
Propensity–weighted Somers' *D* of propensity scores with respect to treatments

- The propensity–weighted Somers' *D* values *should* be zero, *if* the weights standardize out the propensity–treatment association.
- The values for primary propensity scores are near zero for most treatments, but spectacularly nonzero for a few treatments.
- *However*, the values for secondary propensity scores are very nearly zero for all treatments.



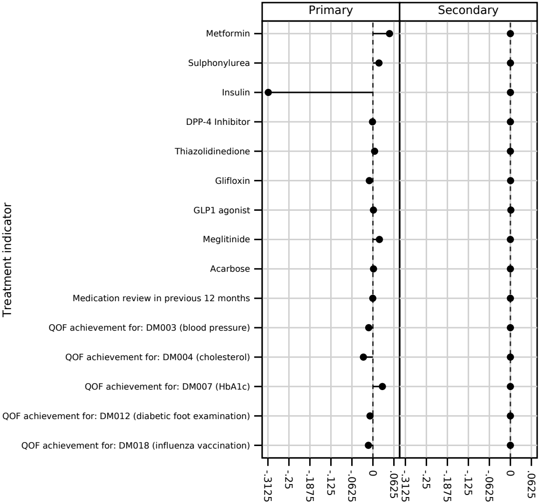
Propensity–weighted Somers' *D* of propensity scores with respect to treatments

- The propensity–weighted Somers' *D* values *should* be zero, *if* the weights standardize out the propensity–treatment association.
- The values for primary propensity scores are near zero for most treatments, but spectacularly nonzero for a few treatments.
- *However*, the values for secondary propensity scores are very nearly zero for all treatments.



Propensity–weighted Somers' *D* of propensity scores with respect to treatments

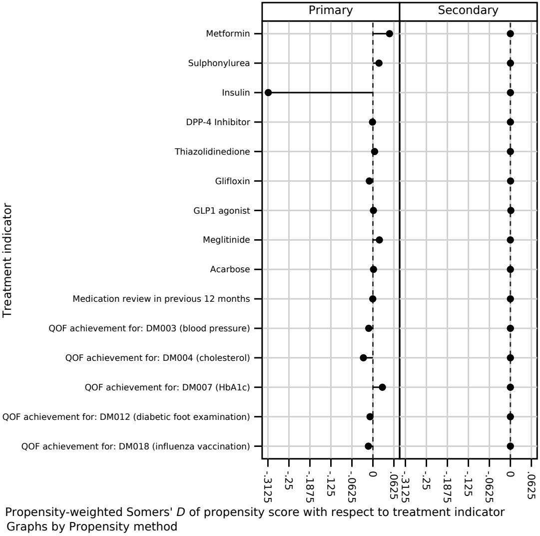
- The propensity–weighted Somers' *D* values *should* be zero, *if* the weights standardize out the propensity–treatment association.
- The values for primary propensity scores are near zero for most treatments, but spectacularly nonzero for a few treatments.
- *However*, the values for secondary propensity scores are very nearly zero for all treatments.



Propensity-weighted Somers' *D* of propensity score with respect to treatment indicator
Graphs by Propensity method

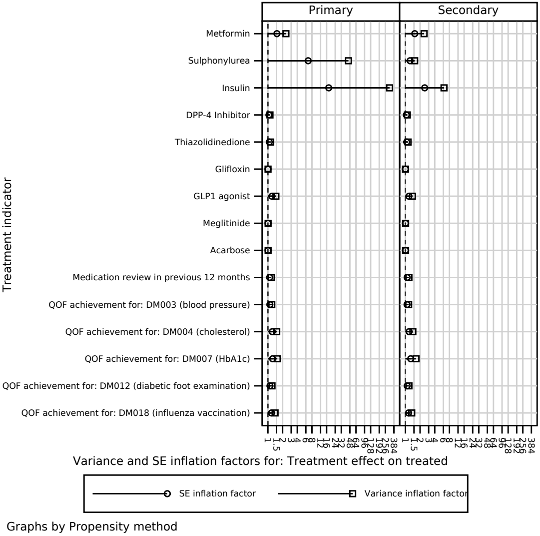
Propensity–weighted Somers' *D* of propensity scores with respect to treatments

- The propensity–weighted Somers' *D* values *should* be zero, *if* the weights standardize out the propensity–treatment association.
- The values for primary propensity scores are near zero for most treatments, but spectacularly nonzero for a few treatments.
- *However*, the values for secondary propensity scores are very nearly zero for all treatments.



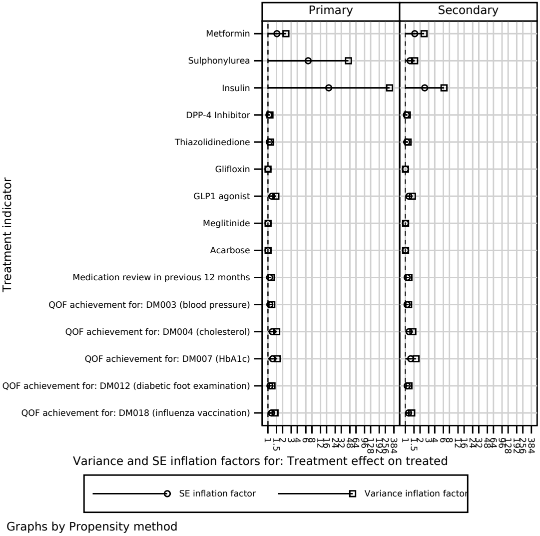
Variance and SE inflation factors for the average treatment effect on the treated (ATET)

- Variance and standard error inflation factors for the ATET are shown on a binary log scale.
- Both types of propensity weights may inflate the variance.
- *However*, the primary propensity weights (unlike the secondary propensity weights) may inflate it by orders of magnitude for some treatments.



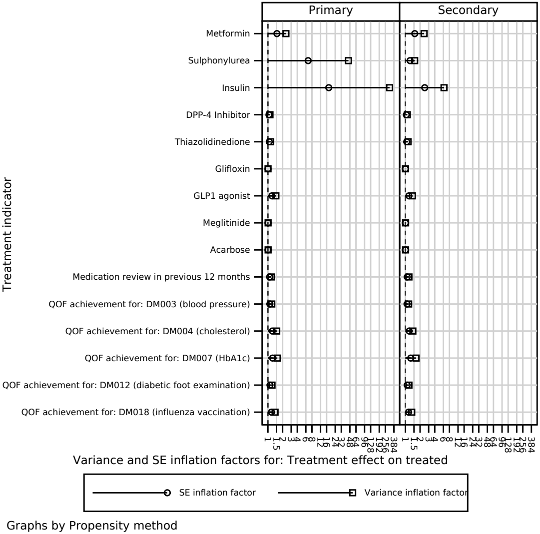
Variance and SE inflation factors for the average treatment effect on the treated (ATET)

- Variance and standard error inflation factors for the ATET are shown on a binary log scale.
- Both types of propensity weights may inflate the variance.
- *However*, the primary propensity weights (unlike the secondary propensity weights) may inflate it by orders of magnitude for some treatments.



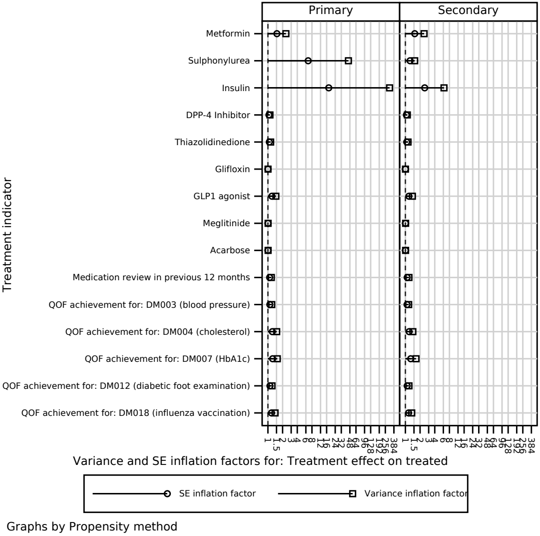
Variance and SE inflation factors for the average treatment effect on the treated (ATET)

- Variance and standard error inflation factors for the ATET are shown on a binary log scale.
- Both types of propensity weights may inflate the variance.
- *However*, the primary propensity weights (unlike the secondary propensity weights) may inflate it by orders of magnitude for some treatments.



Variance and SE inflation factors for the average treatment effect on the treated (ATET)

- Variance and standard error inflation factors for the ATET are shown on a binary log scale.
- Both types of propensity weights may inflate the variance.
- *However*, the primary propensity weights (unlike the secondary propensity weights) may inflate it by orders of magnitude for some treatments.



Summary: Costs and benefits of propensity scores and weights

- The secondary propensity scores (computed using a ridit spline) lost no predictive power, compared to the primary propensity scores.
- *However*, the secondary propensity weights were more reliable than the primary propensity weights for standardizing out the treatment–propensity associations.
- *And*, they *sometimes* caused *much* less variance inflation.
- *So*, the ridit spline seemed to be a good tool for stabilizing propensity weights, and was used in the final analysis to estimate the treatment effects.

Summary: Costs and benefits of propensity scores and weights

- The secondary propensity scores (computed using a ridit spline) lost no predictive power, compared to the primary propensity scores.
- *However*, the secondary propensity weights were more reliable than the primary propensity weights for standardizing out the treatment–propensity associations.
- *And*, they *sometimes* caused *much* less variance inflation.
- *So*, the ridit spline seemed to be a good tool for stabilizing propensity weights, and was used in the final analysis to estimate the treatment effects.

Summary: Costs and benefits of propensity scores and weights

- The secondary propensity scores (computed using a ridit spline) lost no predictive power, compared to the primary propensity scores.
- *However*, the secondary propensity weights were more reliable than the primary propensity weights for standardizing out the treatment–propensity associations.
- *And*, they *sometimes* caused *much* less variance inflation.
- *So*, the ridit spline seemed to be a good tool for stabilizing propensity weights, and was used in the final analysis to estimate the treatment effects.

Summary: Costs and benefits of propensity scores and weights

- The secondary propensity scores (computed using a ridit spline) lost no predictive power, compared to the primary propensity scores.
- *However*, the secondary propensity weights were more reliable than the primary propensity weights for standardizing out the treatment–propensity associations.
- *And*, they *sometimes* caused *much* less variance inflation.
- *So*, the ridit spline seemed to be a good tool for stabilizing propensity weights, and was used in the final analysis to estimate the treatment effects.

Summary: Costs and benefits of propensity scores and weights

- The secondary propensity scores (computed using a ridit spline) lost no predictive power, compared to the primary propensity scores.
- *However*, the secondary propensity weights were more reliable than the primary propensity weights for standardizing out the treatment–propensity associations.
- *And*, they *sometimes* caused *much* less variance inflation.
- *So*, the ridit spline seemed to be a good tool for stabilizing propensity weights, and was used in the final analysis to estimate the treatment effects.

References

[1] Brockett, P. L., and Levene, A. 1977. On a characterization of ridits. *The Annals of Statistics* **5(6)**: 1245–1248.

[2] Bross, I. D. J. 1958. How to use ridit analysis. *Biometrics* **14(1)**: 18–38.

[3] Newson, R. B. 2012. Sensible parameters for univariate and multivariate splines. *The Stata Journal* **12(3)**: 479–504.

[4] Newson, R. B. 2014. Easy–to–use packages for estimating rank and spline parameters. Presented at the *20th UK Stata User Meeting*, 11–12 September, 2014. Downloadable from the conference website at *http://ideas.repec.org/p/boc/usug14/01.html*

[5] Newson, R. B. 2016. The role of Somers' *D* in propensity modelling. Presented at the *22nd UK Stata User Meeting*, 08–09 September, 2016. Downloadable from the conference website at *http://ideas.repec.org/p/boc/usug16/01.html*

This presentation, and the do–file producing the `auto` data examples, can be downloaded from the conference website at *http://ideas.repec.org/s/boc/usug17.html*

The packages used in this presentation can be downloaded from SSC, using the `ssc` command.