

# Definitions of $\psi$ -Functions Available in Robustbase

Manuel Koller and Martin Mächler

June 13, 2019

## Contents

<b>1</b>	<b>Monotone <math>\psi</math>-Functions</b>	<b>2</b>
1.1	Huber . . . . .	3
<b>2</b>	<b>Redescenders</b>	<b>3</b>
2.1	Bisquare . . . . .	4
2.2	Hampel . . . . .	5
2.3	GGW . . . . .	6
2.4	LQQ . . . . .	7
2.5	Optimal . . . . .	8
2.6	Welsh . . . . .	9

## Preamble

Unless otherwise stated, the following definitions of functions are given by [Maronna et al. \(2006, p. 31\)](#), however our definitions differ sometimes slightly from theirs, as we prefer a different way of *standardizing* the functions. To avoid confusion, we first define  $\psi$ - and  $\rho$ -functions.

**Definition 1** A  $\psi$ -function is a piecewise continuous function  $\psi : \mathbb{R} \rightarrow \mathbb{R}$  such that

1.  $\psi$  is odd, i.e.,  $\psi(-x) = -\psi(x) \forall x$ ,
2.  $\psi(x) \geq 0$  for  $x \geq 0$ , and  $\psi(x) > 0$  for  $0 < x < x_r := \sup\{\tilde{x} : \psi(\tilde{x}) > 0\}$  ( $x_r > 0$ , possibly  $x_r = \infty$ ).
- 3\* Its slope is 1 at 0, i.e.,  $\psi'(0) = 1$ .

Note that ‘3\*’ is not strictly required mathematically, but we use it for standardization in those cases where  $\psi$  is continuous at 0. Then, it also follows (from 1.) that  $\psi(0) = 0$ , and we require  $\psi(0) = 0$  also for the case where  $\psi$  is discontinuous in 0, as it is, e.g., for the M-estimator defining the median.

**Definition 2** A  $\rho$ -function can be represented by the following integral of a  $\psi$ -function,

$$\rho(x) = \int_0^x \psi(u) du, \quad (1)$$

which entails that  $\rho(0) = 0$  and  $\rho$  is an even function.

A  $\psi$ -function is called *redescending* if  $\psi(x) = 0$  for all  $x \geq x_r$  for  $x_r < \infty$ , and  $x_r$  is often called *rejection point*. Corresponding to a redescending  $\psi$ -function, we define the function  $\tilde{\rho}$ , a version of  $\rho$  standardized such as to attain maximum value one. Formally,

$$\tilde{\rho}(x) = \rho(x)/\rho(\infty). \quad (2)$$

Note that  $\rho(\infty) = \rho(x_r) \equiv \rho(x) \forall |x| \geq x_r$ .  $\tilde{\rho}$  is a  $\rho$ -function as defined in [Maronna et al. \(2006\)](#) and has been called  $\chi$  function in other contexts. For example, in package `robustbase`, `Mchi(x, *)` computes  $\tilde{\rho}(x)$ , whereas `Mpsi(x, *, deriv=-1)` (“(-1)-st derivative” is the primitive or antiderivative) computes  $\rho(x)$ , both according to the above definitions.

**Note:** An alternative slightly more general definition of *redescending* would only require  $\rho(\infty) := \lim_{x \rightarrow \infty} \rho(x)$  to be finite. E.g., “Welsh” does *not* have a finite rejection point, but *does* have bounded  $\rho$ , and hence well defined  $\rho(\infty)$ , and we *can* use it in `lmrob()`.<sup>1</sup>

**Weakly redescending  $\psi$  functions.** Note that the above definition does require a finite rejection point  $x_r$ . Consequently, e.g., the score function  $s(x) = -f'(x)/f(x)$  for the Cauchy ( $= t_1$ ) distribution, which is  $s(x) = 2x/(1+x^2)$  and hence non-monotone and “re descends” to 0 for  $x \rightarrow \pm\infty$ , and  $\psi_C(x) := s(x)/2$  also fulfills  $\psi_C'(0) = 1$ , but it has  $x_r = \infty$  and hence  $\psi_C()$  is *not* a redescending  $\psi$ -function in our sense. As they appear e.g. in the MLE for  $t_\nu$ , we call  $\psi$ -functions fulfilling  $\lim_{x \rightarrow \infty} \psi(x) = 0$  *weakly redescending*. Note that they’d naturally fall into two sub categories, namely the one with a *finite*  $\rho$ -limit, i.e.  $\rho(\infty) := \lim_{x \rightarrow \infty} \rho(x)$ , and those, as e.g., the  $t_\nu$  score functions above, for which  $\rho(x)$  is unbounded even though  $\rho' = \psi$  tends to zero.

## 1 Monotone $\psi$ -Functions

Monotone  $\psi$ -functions lead to convex  $\rho$ -functions such that the corresponding M-estimators are defined uniquely.

Historically, the “Huber function” has been the first  $\psi$ -function, proposed by Peter Huber in [Huber \(1964\)](#).

---

<sup>1</sup>E-mail Oct. 18, 2014 to Manuel and Werner, proposing to change the definition of “redescending”.

## 1.1 Huber

The family of Huber functions is defined as,

$$\rho_k(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \leq k \\ k(|x| - \frac{k}{2}) & \text{if } |x| > k \end{cases},$$

$$\psi_k(x) = \begin{cases} x & \text{if } |x| \leq k \\ k \operatorname{sign}(x) & \text{if } |x| > k \end{cases}.$$

The constant  $k$  for 95% efficiency of the regression estimator is 1.345.

```
> plot(huberPsi, x., ylim=c(-1.4, 5), leg.loc="topright", main=FALSE)
```

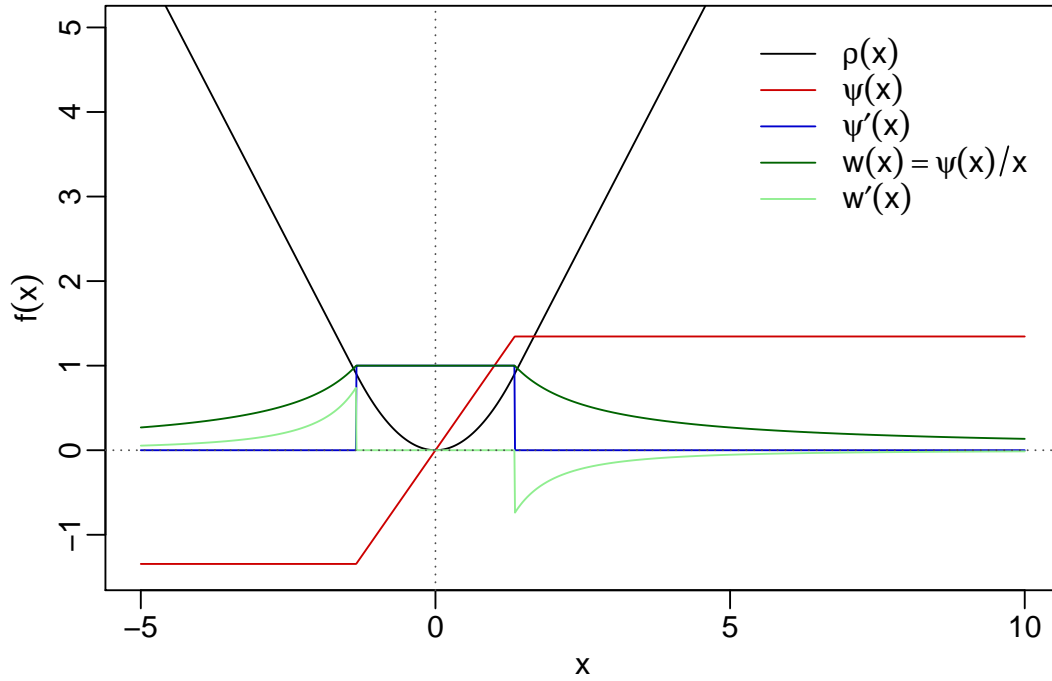


Figure 1: Huber family of functions using tuning parameter  $k = 1.345$ .

## 2 Redescenders

For the MM-estimators and their generalizations available via `lmrob()` (and for some methods of `nlrob()`), the  $\psi$ -functions are all redescending, i.e., with finite “rejection point”  $x_r = \sup\{t; \psi(t) > 0\} < \infty$ . From `lmrob`, the psi functions are available via `lmrob.control`, or more directly, `.Mpsi.tuning.defaults`,

```
> names(.Mpsi.tuning.defaults)
```

```
[1] "huber"      "bisquare"  "welsh"    "ggw"      "lqq"
[6] "optimal"   "hampel"
```

and their  $\psi$ ,  $\rho$ ,  $\psi'$ , and weight function  $w(x) := \psi(x)/x$ , are all computed efficiently via C code, and are defined and visualized in the following subsections.

## 2.1 Bisquare

Tukey's bisquare (aka "biweight") family of functions is defined as,

$$\tilde{\rho}_k(x) = \begin{cases} 1 - \left(1 - (x/k)^2\right)^3 & \text{if } |x| \leq k \\ 1 & \text{if } |x| > k \end{cases},$$

with derivative  $\tilde{\rho}'_k(x) = 6\psi_k(x)/k^2$  where,

$$\psi_k(x) = x \left(1 - \left(\frac{x}{k}\right)^2\right)^2 \cdot I_{\{|x| \leq k\}}.$$

The constant  $k$  for 95% efficiency of the regression estimator is 4.685 and the constant for a breakdown point of 0.5 of the S-estimator is 1.548. Note that the *exact* default tuning constants for M- and MM- estimation in **robustbase** are available via `.Mpsi.tuning.default()` and `.Mchi.tuning.default()`, respectively, e.g., here,

```
> print(c(k.M = .Mpsi.tuning.default("bisquare"),
+         k.S = .Mchi.tuning.default("bisquare")), digits = 10)
```

```
      k.M      k.S
4.685061 1.547640
```

and that the `p.psiFun(.)` utility is available via

```
> source(system.file("xtraR/plot-psiFun.R", package = "robustbase", mustWork=TRUE))
```

```
> p.psiFun(x., "biweight", par = 4.685)
```

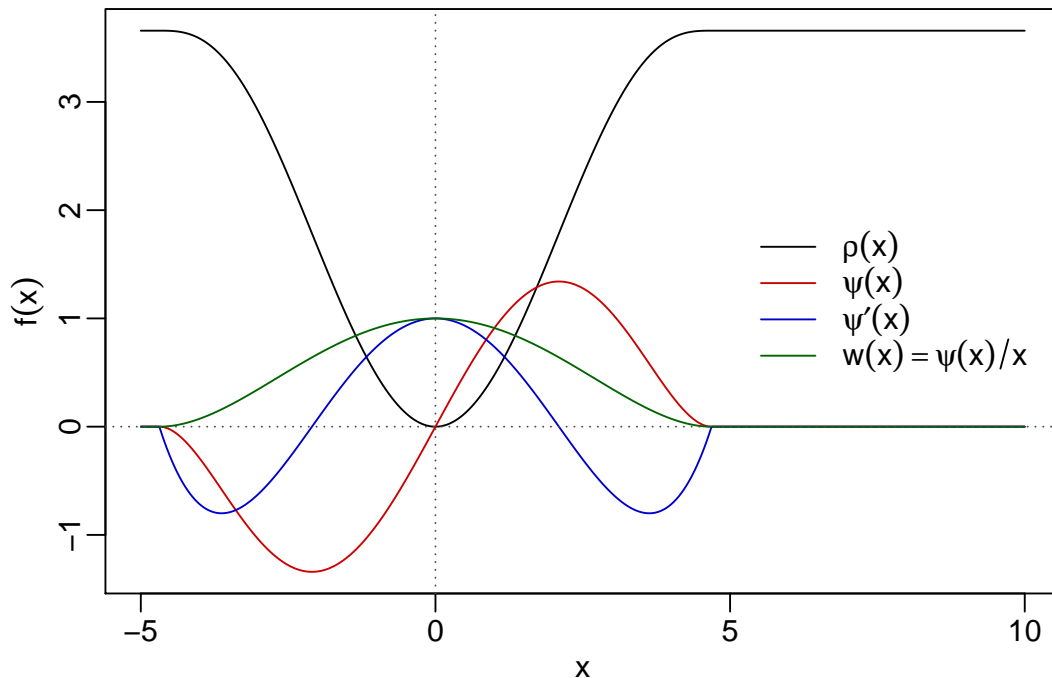


Figure 2: Bisquare family functions using tuning parameter  $k = 4.685$ .

## 2.2 Hampel

The Hampel family of functions (Hampel et al., 1986) is defined as,

$$\tilde{\rho}_{a,b,r}(x) = \begin{cases} \frac{1}{2}x^2/C & |x| \leq a \\ \left(\frac{1}{2}a^2 + a(|x| - a)\right)/C & a < |x| \leq b \\ \frac{a}{2} \left(2b - a + (|x| - b) \left(1 + \frac{r-|x|}{r-b}\right)\right)/C & b < |x| \leq r \\ 1 & r < |x| \end{cases},$$

$$\psi_{a,b,r}(x) = \begin{cases} x & |x| \leq a \\ a \operatorname{sign}(x) & a < |x| \leq b \\ a \operatorname{sign}(x) \frac{r-|x|}{r-b} & b < |x| \leq r \\ 0 & r < |x| \end{cases},$$

where  $C := \rho(\infty) = \rho(r) = \frac{a}{2}(2b - a + (r - b)) = \frac{a}{2}(b - a + r)$ .

As per our standardization,  $\psi$  has slope 1 in the center. The slope of the redescending part ( $x \in [b, r]$ ) is  $-a/(r - b)$ . If it is set to  $-\frac{1}{2}$ , as recommended sometimes, one has

$$r = 2a + b.$$

Here however, we restrict ourselves to  $a = 1.5k$ ,  $b = 3.5k$ , and  $r = 8k$ , hence a redescending slope of  $-\frac{1}{3}$ , and vary  $k$  to get the desired efficiency or breakdown point.

The constant  $k$  for 95% efficiency of the regression estimator is 0.902 (0.9016085, to be exact) and the one for a breakdown point of 0.5 of the S-estimator is 0.212 (i.e., 0.2119163).

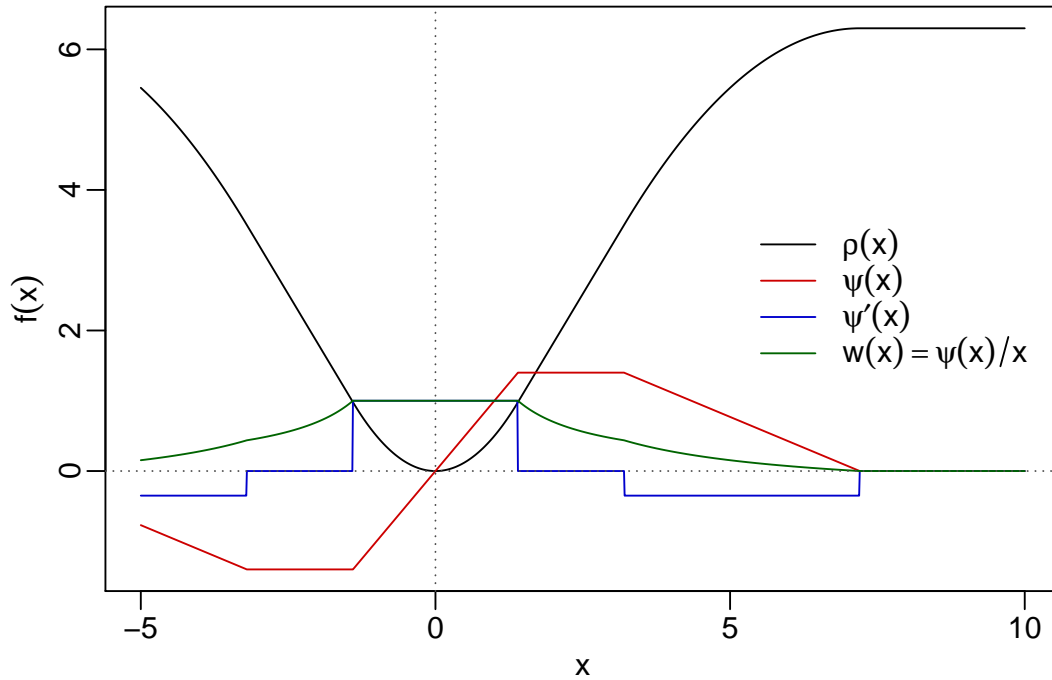


Figure 3: Hampel family of functions using tuning parameters  $0.902 \cdot (1.5, 3.5, 8)$ .

## 2.3 GGW

The Generalized Gauss-Weight function, or *ggw* for short, is a generalization of the Welsh  $\psi$ -function (subsection 2.6). In [Koller and Stahel \(2011\)](#) it is defined as,

$$\psi_{a,b,c}(x) = \begin{cases} x & |x| \leq c \\ \exp\left(-\frac{1}{2}\frac{(|x|-c)^b}{a}\right) x & |x| > c \end{cases}.$$

Our constants, fixing  $b = 1.5$ , and minimal slope at  $-\frac{1}{2}$ , for 95% efficiency of the regression estimator are  $a = 1.387$ ,  $b = 1.5$  and  $c = 1.063$ , and those for a breakdown point of 0.5 of the S-estimator are  $a = 0.204$ ,  $b = 1.5$  and  $c = 0.296$ :

```
> cT <- rbind(cc1 = .psi.ggw.findc(ms = -0.5, b = 1.5, eff = 0.95      ),
+             cc2 = .psi.ggw.findc(ms = -0.5, b = 1.5,                bp = 0.50)); cT

      [,1]      [,2] [,3]      [,4]      [,5]
cc1      0 1.3863620  1.5 1.0628199 4.7773893
cc2      0 0.2036739  1.5 0.2959131 0.3703396
```

Note that above,  $cc*[1] = 0$ ,  $cc*[5] = \rho(\infty)$ , and  $cc*[2:4] = (a, b, c)$ . To get this from  $(a, b, c)$ , you could use

```
> ipsi.ggw <- .psi2ipsi("GGW") # = 5
> ccc <- c(0, cT[1, 2:4], 1)
> integrate(.Mpsi, 0, Inf, ccc=ccc, ipsi=ipsi.ggw)$value # = rho(Inf)
```

```
[1] 4.777389
```

```
> p.psiFun(x., "GGW", par = c(-.5, 1, .95, NA))
```

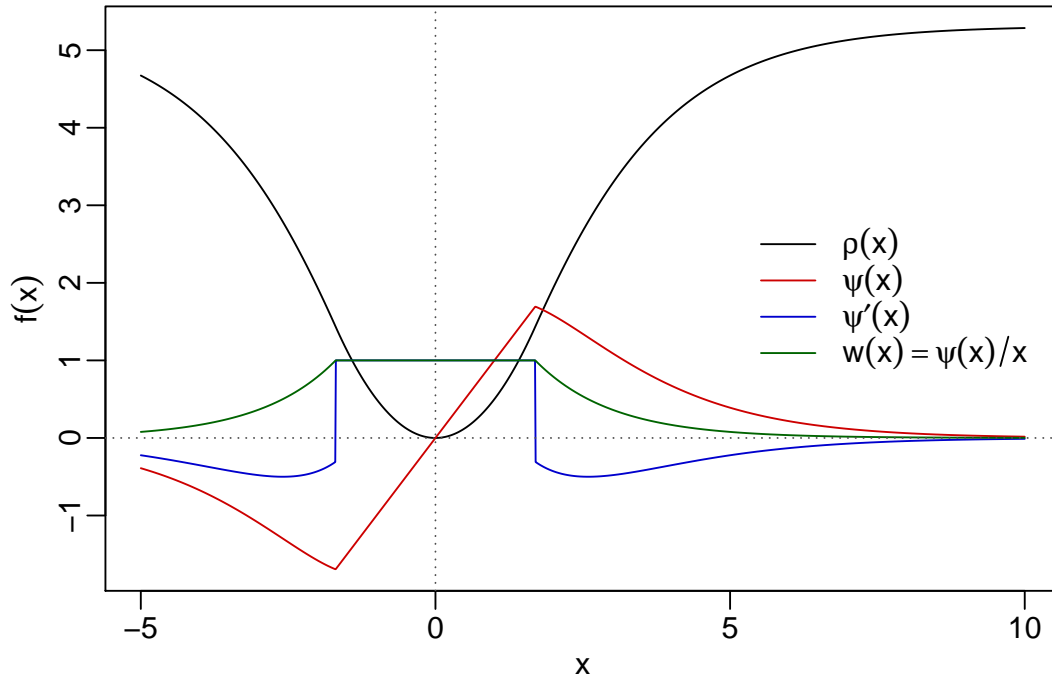


Figure 4: GGW family of functions using tuning parameters  $a = 1.387$ ,  $b = 1.5$  and  $c = 1.063$ .

## 2.4 LQQ

The “linear quadratic quadratic”  $\psi$ -function, or *lqq* for short, was proposed by [Koller and Stahel \(2011\)](#). It is defined as,

$$\psi_{b,c,s}(x) = \begin{cases} x & |x| \leq c \\ \text{sign}(x) \left( |x| - \frac{s}{2b} (|x| - c)^2 \right) & c < |x| \leq b + c \\ \text{sign}(x) \left( c + b - \frac{bs}{2} + \frac{s-1}{a} \left( \frac{1}{2}\tilde{x}^2 - a\tilde{x} \right) \right) & b + c < |x| \leq a + b + c \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\tilde{x} := |x| - b - c \quad \text{and} \quad a := (2c + 2b - bs)/(s - 1). \quad (3)$$

The parameter  $c$  determines the width of the central identity part. The sharpness of the bend is adjusted by  $b$  while the maximal rate of descent is controlled by  $s$  ( $s = 1 - \min_x \psi'(x) > 1$ ). From (3), the length  $a$  of the final descent to 0 is a function of  $b$ ,  $c$  and  $s$ .

```
> cT <- rbind(cc1 = .psi.lqq.findc(ms= -0.5, b.c = 1.5, eff=0.95, bp=NA ),
+             cc2 = .psi.lqq.findc(ms= -0.5, b.c = 1.5, eff=NA , bp=0.50))
> colnames(cT) <- c("b", "c", "s"); cT
```

	b	c	s
cc1	1.4734061	0.9822707	1.5
cc2	0.4015457	0.2676971	1.5

If the minimal slope is set to  $-\frac{1}{2}$ , i.e.,  $s = 1.5$ , and  $b/c = 3/2 = 1.5$ , the constants for 95% efficiency of the regression estimator are  $b = 1.473$ ,  $c = 0.982$  and  $s = 1.5$ , and those for a breakdown point of 0.5 of the S-estimator are  $b = 0.402$ ,  $c = 0.268$  and  $s = 1.5$ .

```
> p.psiFun(x., "LQQ", par = c(-.5, 1.5, .95, NA))
```

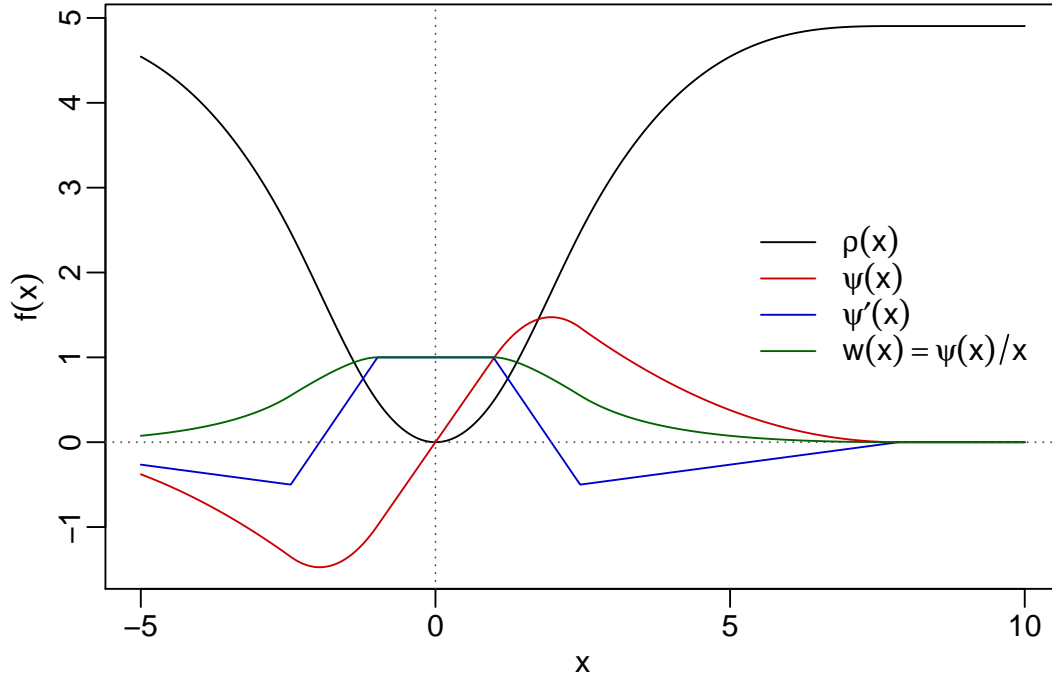


Figure 5: LQQ family of functions using tuning parameters  $b = 1.473$ ,  $c = 0.982$  and  $s = 1.5$ .

## 2.5 Optimal

The optimal  $\psi$  function as given by [Maronna et al. \(2006, Section 5.9.1\)](#),

$$\psi_c(x) = \text{sign}(x) \left( -\frac{\varphi'(|x|) + c}{\varphi(|x|)} \right)_+,$$

where  $\varphi$  is the standard normal density,  $c$  is a constant and  $t_+ := \max(t, 0)$  denotes the positive part of  $t$ .

Note that the **robustbase** implementation uses rational approximations originating from the **robust** package's implementation. That approximation also avoids an anomaly for small  $x$  and has a very different meaning of  $c$ .

The constant for 95% efficiency of the regression estimator is 1.060 and the constant for a breakdown point of 0.5 of the S-estimator is 0.405.

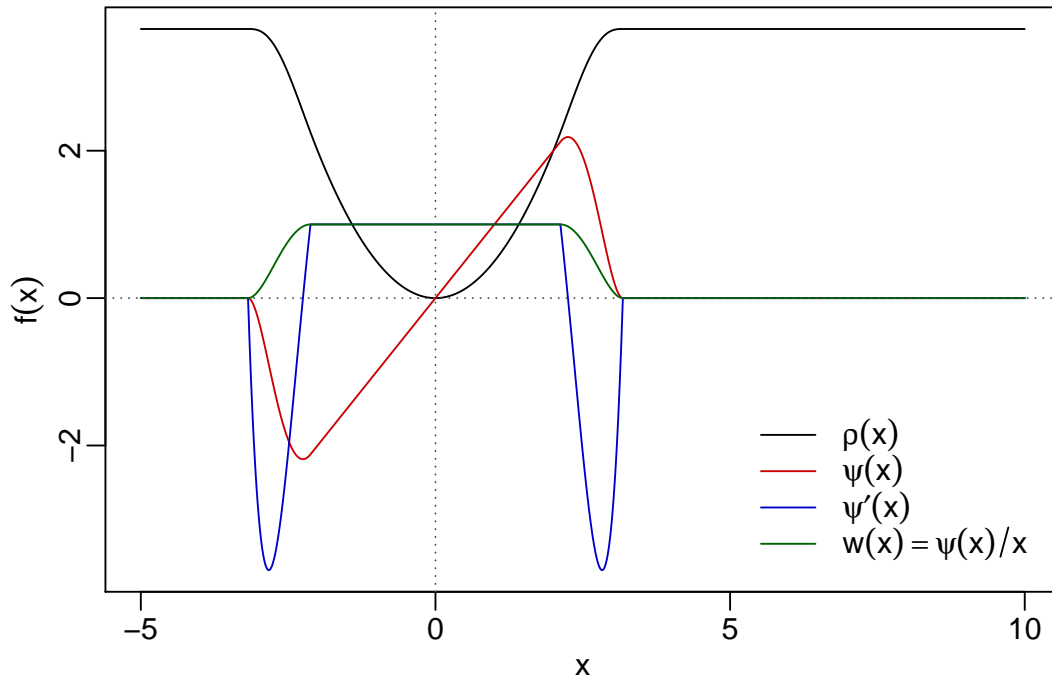


Figure 6: ‘Optimal’ family of functions using tuning parameter  $c = 1.06$ .



## 2.6 Welsh

The Welsh  $\psi$  function is defined as,

$$\begin{aligned}\tilde{\rho}_k(x) &= 1 - \exp(-(x/k)^2/2) \\ \psi_k(x) &= k^2 \tilde{\rho}'_k(x) = x \exp(-(x/k)^2/2) \\ \psi'_k(x) &= (1 - (x/k)^2) \exp(-(x/k)^2/2)\end{aligned}$$

The constant  $k$  for 95% efficiency of the regression estimator is 2.11 and the constant for a breakdown point of 0.5 of the S-estimator is 0.577.

Note that GGW (subsection 2.3) is a 3-parameter generalization of Welsh, matching for  $b = 2$ ,  $c = 0$ , and  $a = k^2$  (see R code there):

```
> ccc <- c(0, a = 2.11^2, b = 2, c = 0, 1)
> (ccc[5] <- integrate(.Mpsi, 0, Inf, ccc=ccc, ipsi = 5)$value) # = rho(Inf)
[1] 4.4521

> stopifnot(all.equal(Mpsi(x., ccc, "GGW"), ## psi[ GGW ](x; a=k^2, b=2, c=0) ==
+                      Mpsi(x., 2.11, "Welsh")))## psi[Welsh](x; k)
```

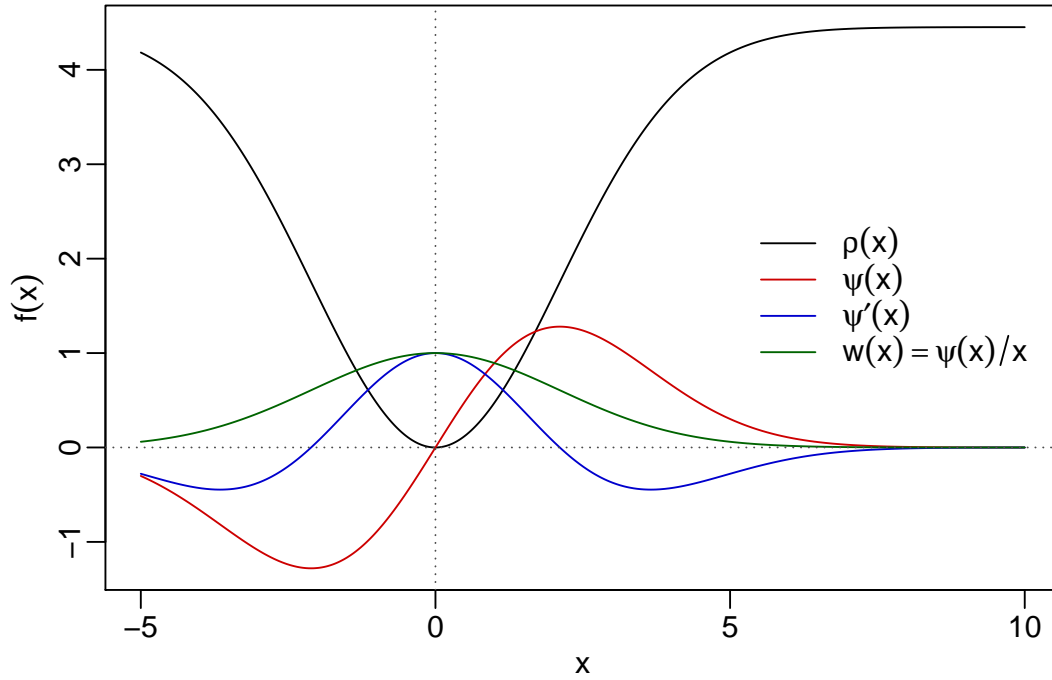


Figure 7: Welsh family of functions using tuning parameter  $k = 2.11$ .

## References

- Hampel, F., E. Ronchetti, P. Rousseeuw, and W. Stahel (1986). *Robust Statistics: The Approach Based on Influence Functions*. N.Y.: Wiley.
- Huber, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.* 35, 73–101.
- Koller, M. and W. A. Stahel (2011). Sharpening wald-type inference in robust regression for small samples. *Computational Statistics & Data Analysis* 55(8), 2504–2515.
- Maronna, R. A., R. D. Martin, and V. J. Yohai (2006). *Robust Statistics, Theory and Methods*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd.