

11-1-2016

Fitting Flexible Parametric Regression Models with GLDreg in R

Steve Su

Covance, allegro.su@gmail.com

Follow this and additional works at: <http://digitalcommons.wayne.edu/jmasm>

 Part of the [Applied Statistics Commons](#), [Social and Behavioral Sciences Commons](#), and the [Statistical Theory Commons](#)

Recommended Citation

Su, Steve (2016) "Fitting Flexible Parametric Regression Models with GLDreg in R," *Journal of Modern Applied Statistical Methods*: Vol. 15 : Iss. 2 , Article 46.

DOI: 10.22237/jmasm/1478004240

Available at: <http://digitalcommons.wayne.edu/jmasm/vol15/iss2/46>

This Statistical Software Applications and Review is brought to you for free and open access by the Open Access Journals at DigitalCommons@WayneState. It has been accepted for inclusion in Journal of Modern Applied Statistical Methods by an authorized editor of DigitalCommons@WayneState.

Fitting Flexible Parametric Regression Models with *GLDreg* in R

Steve Su
Covance
Sydney, Australia

This article outlines the functionality of the *GLDreg* package in R which fits parametric regression models using generalized lambda distributions via maximum likelihood estimation and L moment matching. The main advantage of *GLDreg* is the provision of robust regression lines and smooth regression quantiles beyond the capabilities of existing known methods.

Keywords: Regression model, quantile regression, generalized lambda distributions, GLDreg, GLDEX, R

Package *GLDreg* in R

The *GLDreg* package in R is designed to implement the Generalized Lambda Distribution (GLD) regression model outlined in Su (2015) with some extensions. Currently, it is possible to fit GLD regression to data using maximum likelihood estimation (MLE) (Su, 2007a; b) and L moment matching (Asquith, 2007; Karvanen & Nuutinen, 2008). Users may also chose initial values to start the model building process, or use the default searching algorithm using the ordinary least square regression model (Su, 2015). The *GLDreg* package also allows user to fit quantile regressions parametrically and non-parametrically by: 1) fixing the intercept, 2) fixing coefficients other than the intercept, and 3) allowing all coefficients to vary.

The *GLDreg* package requires *GLDEX* (Su, 2007a; 2010). The *GLDEX 2.0.0.1* package has a faster implementation of the GLD fitting algorithm compared to its predecessors. This is because a number of frequently used codes have been written in C. In addition, the *GLDEX 2.0.0.1* package has faster maximum likelihood fitting functions *fun.RMFMKL.ml.m*, *fun.RPRS.ml.m* for FKML

Steve Su is a Senior Statistician. Email him at: allegro.su@gmail.com.

(Freimer, Kollia, Mudholkar, & Lin, 1988) and RS (Ramberg & Schmeiser, 1974) GλDs.

Background

Traditionally, the Pearson and Johnson systems are considered to be the standard approaches to identifying and fitting different types of statistical distributions to data. However, these systems require different mathematical functions to cover a range of symmetric and asymmetric distributions. Tukey's lambda distribution contains a class of symmetric distributions which can approximate a number of common distributions such as Normal and Cauchy distributions, and provide an indication as to whether heavy tailed distributions are needed. Ramberg and Schmeiser (1974) then generalized Tukey's lambda distribution to include both symmetric and asymmetric distributions, and this became known as the GλD. Although defined as a single mathematical function, GλD can cover a broad range of statistical distributions which is much more efficient than the use of Pearson and Johnson systems involving several mathematical functions. Since then, the flexibility of GλD has attracted a number of researchers. Today, along with the increased computation power and the introduction of dedicated packages for GλD in R, it is now possible to fit GλD to data and extend the use of GλD in many areas of statistical analysis.

Generalized Lambda Distributions

The RS GλD (Ramberg & Schmeiser, 1974) is defined by its inverse distribution function:

$$F^{-1}(u) = \lambda_1 + \frac{u^{\lambda_3} - (1-u)^{\lambda_4}}{\lambda_2}, \quad 0 \leq u \leq 1 \quad (1)$$

From (1), $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are respectively the location, inverse scale, and shape parameters of generalized lambda distribution $G\lambda D(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$. Note that λ_3, λ_4 are both shape parameters. Karian, Dudewicz, and McDonald (1996) noted that GλD is defined only if

$$\frac{\lambda_2}{\lambda_4(1-u)^{\lambda_4-1} + \lambda_3 u^{\lambda_3-1}} \geq 0$$

where $0 \leq u \leq 1$.

Freimer et al. (1988) described another distribution known as FKML/FMKL G λ D. This distribution is slightly different to RS G λ D with respect to formulation. The FKML/FMKL G λ D is defined as:

$$F^{-1}(u) = \lambda_1 \frac{u^{\lambda_3} - 1}{\lambda_3} - \frac{(1-u)^{\lambda_4} - 1}{\lambda_4} \lambda_2, \quad 0 \leq u \leq 1 \quad (2)$$

Under (2), λ_1 , λ_2 , λ_3 , λ_4 are respectively the location, scale, and shape parameters of the generalized lambda distribution. Again, both λ_3 , λ_4 are shape parameters. Technically, the correct abbreviation for this distribution is FKML distribution, based on the correct ordering of authorship in the original paper. However, the use of the term FMKL distribution had been widespread in the literature and this package, along with *GLDEX 2.0.0.1*, will allow both “*fkml*” and “*finkl*” specifications in the implementation of FKML distribution.

The fundamental motivation for the development of FKML G λ D is that the distribution is defined over all λ_3 and λ_4 (Freimer et al., 1988). The only restriction on FKML G λ D is $\lambda_2 > 0$.

An extensive discussion of the shapes and properties of G λ Ds can be found in the original papers (Ramberg & Schmeiser, 1974; Freimer et al., 1988), as well as in subsequent works such as in Su (2015). Note the probability density functions of G λ Ds is obtained by observing $F^{-1}(u) = x$ and so

$$f(x) = \frac{1}{\frac{dF^{-1}(u)}{du}}$$

This is calculated using the Newton-Raphson method in *GLDEX*.

Sketch of the Regression Fitting Mechanism

The full fitting algorithm of G λ D regression models is provided in Su (2015). This section sketches the procedure in building these models. Consider the following regression model

$$Y = X\beta + \varepsilon \quad (3)$$

The goal is to find estimated coefficients for β under the condition $\varepsilon \sim \text{GLD}(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, with the property such that $E(\varepsilon) = 0$. Consequently, the average deviation of actual and estimated values should be zero.

When deriving the probability density function of GLDs, λ_1 will vanish, which means this parameter has no bearing on the overall shape of the distribution, only its location. The consequence is that it is possible to shift the GLDs to have exactly zero mean by calculating λ_1 after estimating $\lambda_2, \lambda_3, \lambda_4$.¹

The idea of fitting the GLD regression in Su (2015) is to find a set of $\hat{\beta}$ by modelling the residuals using GLD through maximum likelihood estimation or L moment matching. The zero mean residual line is achieved by only allowing $\lambda_2, \lambda_3, \lambda_4$ to vary in the optimization process, and the intercept of the line is adjusted to ensure the residuals add up to zero. Statistical properties of GLD regression coefficients are obtained by recreating actual values $y_k = \hat{y}_k + \hat{\varepsilon}_k$ for $k = 1, 2, 3, \dots, n$ observations by simulating $\hat{\varepsilon}_k \sim \text{GLD}(\hat{\lambda}_1, \hat{\lambda}_2, \hat{\lambda}_3, \hat{\lambda}_4)$ and refitting the entire model and repeating the process, say 1000 times. This can all be done using the *GLD.lm.full* function and the goodness of fit of the model is assessed using the Kolmogorov-Smirnoff test and QQ plots. To run a simple model without any simulations, the function *GLD.lm* can be used instead.

Once a reference GLD regression line is found, quantile regression can be obtained by 1) fixing the intercept, 2) fixing coefficients other than the intercept, and 3) allowing all coefficients to vary. Case 1 and 3 are designed to fit non parallel lines for heteroskedastic data, and case 2 is primarily designed to fit parallel lines for homoskedastic data.

GLDreg allows non-parametric and parametric fitting of quantile regression lines. The non-parametric approach uses the least squares approach to find a q -th quantile GLD line such that the percentage of observations below the line corresponds to the q -th quantile. The parametric approach uses the least squares approach to find a q -th quantile GLD line such that the percentage of observations below the line (under a GLD fit) corresponds to the q -th quantile. In the case of 1) and 3), the Nelder-Mead simplex algorithm from *optim* is used in the optimization process. In the case of 2), the Brent method from *optim* is used instead. The initial values for non-parametric quantile regression optimization are taken from the sample quantile of simulated regression coefficients obtained during the model building process. For the parametric approach, the initial values are taken from the non-parametric quantile regression. The estimates of both parametric and non-parametric GLD quantile regression can be obtained using a single wrapper function (*GLD.quantreg*).

Using the Package

GLDreg can be installed from CRAN² through the R interface. Once installed, the following command will load the package.

```
> library(GLDreg)
```

```
Loading required package: GLDEX
Loading required package: cluster
```

As usual, `?GLDreg` will take user to the main help menu. Table 1 shows the main functions available under *GLDreg*.

Table 1. List of main functions

	Purpose	Function
	Fit GAD Regression only	<code>GAD.lm</code>
Fit GAD Regression and obtain statistical properties through simulation		<code>GLD.lm.full</code>
	Fit GAD quantile regression	<code>GLD.quantreg</code>
Plot summary graphics of GAD regression		<code>summaryGraphics.GAD.lm</code>
	Plot quantile regression lines	<code>fun.plot.q</code>

Table 2. Fitting RS or FKML GAD regression using MLE or L moment matching under `GLD.lm` or `GLD.lm.full`

Type of GAD	Type of estimation	Param	fun
RS	MLE	"rs"	<code>fun.RPRS.ml.m</code> or <code>fun.RPRS.ml</code>
RS	L moment matching	"rs"	<code>fun.RPRS.lm</code>
FKML	MLE	"fmkl" or "fkml"	<code>fun.RMFMKL.ml.m</code> or <code>fun.RMFMKL.ml</code>
FKML	L moment matching	"fmkl" or "fkml"	<code>fun.RMFMKL.lm</code>

GLDreg currently implements MLE and L moment matching for GAD regression. The associated *param* and *fun* inputs for `GLD.lm` and `GLD.lm.full` are given in Table 2. The `fun.RPRS.ml.m` and `fun.RMFMKL.ml.m` functions are faster implementations of MLE than the previous `fun.RPRS.ml` and `fun.RMFMKL.ml` functions.

Engel Dataset

This example uses the well-known Engel dataset from the *quantreg* package in R. The following fits a full FKML GLD regression model using MLE, along with simulations (default is 1000 runs), and then fits a quantile regression for quantiles from 0.1 to 0.9 in 9 equal spacings with fixed intercept and varying slopes.

```
> library(quantreg)
> data(engel)
> set.seed(1000)

> engel.fit.all<-GLD.lmfull(foodexp~income,data=engel,
+ param="fmkl",fun=fun.RMFMKL.ml.m,summary.plot=F)

> result<-GLD.quantreg(seg(0.1,.9,length=9),
+engel.fit.all,intercept="fixed")
```

There are warnings associated with using the Nelder-Mead algorithm for single parameter optimization problems. To check whether optimization has indeed been achieved for quantile regression, the easiest way is to check whether the proportion of the fitted quantile line below the response variable corresponds to the level of quantile specified. The multiplication of 100 is to convert the quantiles into percentages, and the aim is to see how close these are to 10, 20, 30, 40, 50, 60, 70, 80 and 90.

```
> sapply(1:9,function(i) sum(engel$foodexp-
cbind(1,engel$income)%*(result[1:2,i])<0)/nrow(engel)*100)
```

```
[1] 8.085106 22.127660 30.638298 38.723404 51.489362 58.723404
67.659574 82.978723 90.638298
```

The results are reasonably close to the intended quantiles. Note differences occur because the optimization here is based on matching the quantile of fitted GLD rather than the empirical data, the idea being that if the fitted GLD is close to the actual distribution, the quantile lines obtained here will be more robust to changes in empirical data.

Alternatively, quantile lines could be obtained using the non-parametric approach using the following, and checked whether the desired quantile line has

been achieved despite warnings from the *optim* function. Obviously, the match is better, simply because optimization was done based on empirical data.

Once the quantile lines are obtained, it is possible to plot them; in this case, the parametric G λ D quantile lines are plotted in Figure 1.

```
> fun.plot.q(x=engel$income,y=engel$foodexp,fit=engel.fit.all[[1]],
result)
```

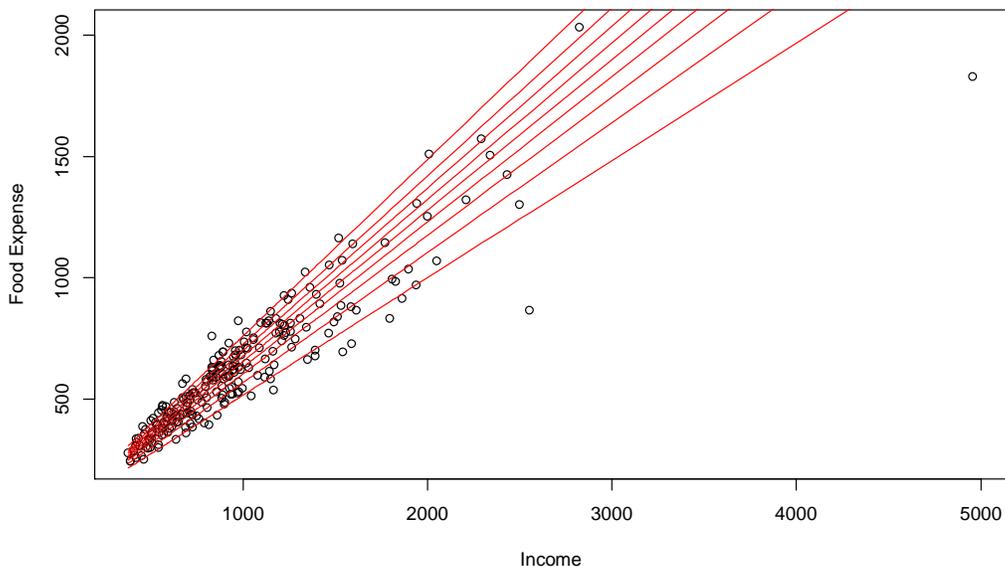


Figure 1. FKML G λ D quantile regression for Engel dataset

It is also possible to speed up *GLD.lm.full* by reducing the number of simulations to 100 by setting $n = 100$, if the primary purpose is to obtain quantile regression rather than looking at the statistical properties of the coefficients. *GLDreg* also provides a graphic summary of regression coefficients based on object obtained from *GLD.lm.full*, using the *summaryGraphics.gld.lm* function, and the graphics are shown in Figure 2 and Figure 3.

```
> summaryGraphics.gld.lm(engel.fit.all)
```

Shown in Figure 2 is the statistical distribution of coefficients obtained using simulation, and the 95% interval is obtained directly from simulated results. Shown in Figure 3 is how well FKML GAD fits the data; in this case the fit is quite good, with Kolmogorov-Smirnoff goodness of fit p -value exceeding 5% and the QQ plot indicating a close fit.

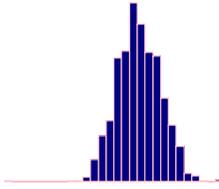
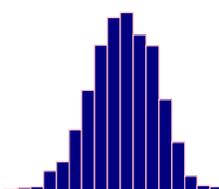
Parameter	Estimate	Density	Summary 95% interval:
(Intercept)	32.2		-20.5, 85.1
income	0.603		0.55, 0.649

Figure 2. FKML GAD quantile regression for Engel dataset coefficient summary plot

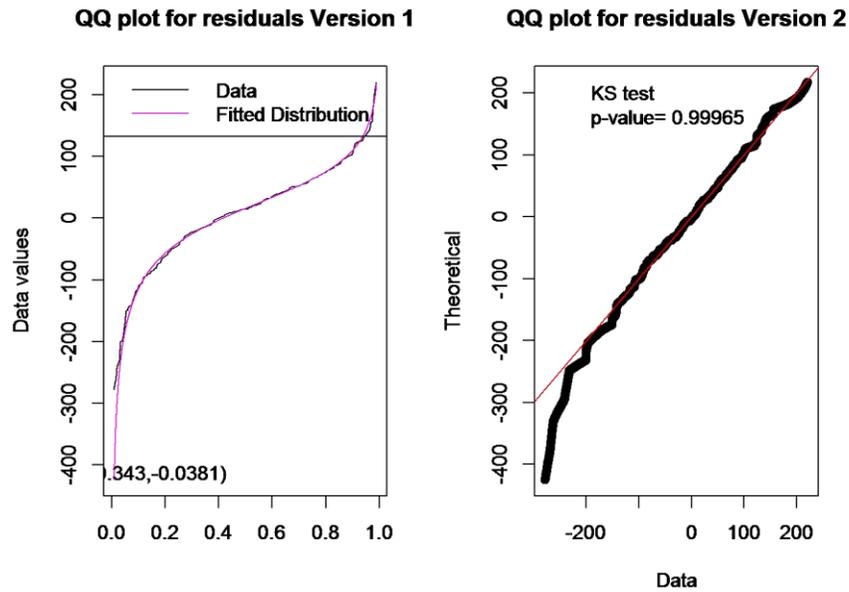


Figure 3. FKML GAD quantile regression for Engel dataset – QQ plots

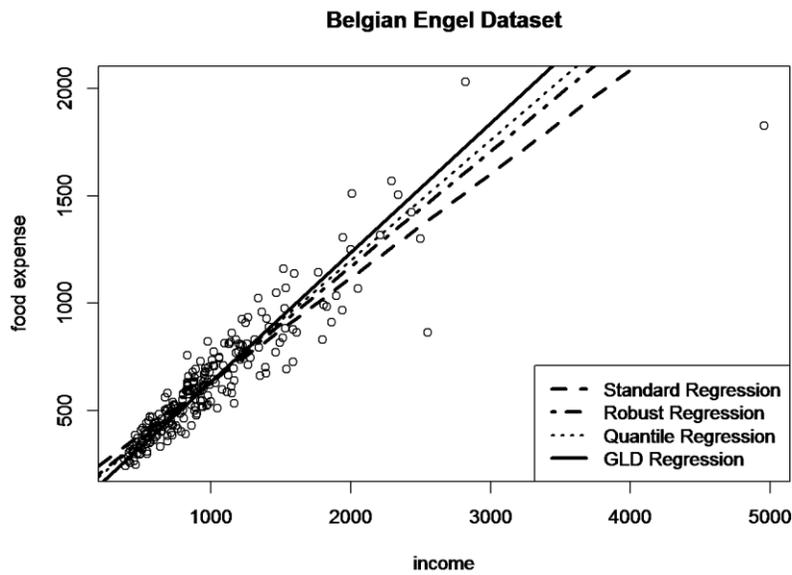


Figure 4. Comparison of different regression techniques for Engel dataset

Note the FKML G λ D regression line is the most robust to outliers compared to linear regression, robust regression, and quantile regression at median. This can be seen using the following code, which is displayed in [Figure 4](#).

```

> library(quantreg)
> library(MASS)
> par(mfrow=c(1,1))

> plot(foodexp~income,data=engel,xlab="income",ylab="food expense",
+ main="Belgian Engel Dataset")

> abline(lm(foodexp~income,data=engel),lty=2,lwd=3)
> abline(rlm(foodexp~income,data=engel),lty=4,lwd=3)
> abline(rq(foodexp~income,data=engel),lty=3,lwd=2)
> abline(engel.fit.all[[1]][[3]][1:2],lty=1,lwd=3)

> legend("bottomright",c("Standard Regression","Robust Regression",
+ "Quantile Regression","GLD Regression"),lty=c(2,4,3,1),lwd=c(3,3,2,3))

```

Simulated Motorcycle Accident Dataset

The *mcycle* dataset is a simulated motorcycle accident dataset from MASS library in R. It is possible to fit splines to this dataset, and a reasonable strategy is to fit time before 15 seconds separately as was done in Su (2015). In the same fashion as in Engel dataset analysis, the full G λ D regression is fitted first, followed by evaluation of quantile regression lines, for which parallel lines appear to be quite suitable for this dataset. The only difference is that the modelling is split into two parts. In this case, the first part (before 15 seconds) was fitted using splines with 8 degrees of freedom using RS G λ D via maximum likelihood estimation. The second part (greater or equal to 15 seconds) was fitted using splines with 15 degrees of freedom using FKML G λ D via maximum likelihood estimation. The code to produce [Figure 5](#) is given below.

```

> library(MASS)
> library(splines)

> cutoff<-15
> mcycle.part1<-mcycle[mcycle$times<cutoff,]

```

GLDreg IN R

```
> mcycle.part2<-mcycle[mcycle$times>=cutoff,]  
  
> mcycle.p1.fit<-GLD.lm.full(accel~bs(times,df=8),  
+ data=mcycle.part1,param="rs",fun=fun.RPRS.ml.m,summary.plot=F,n=100)  
> mcycle.p2.fit<-GLD.lm.full(accel~bs(times,df=15),  
+ data=mcycle.part2,param="fmkl",fun=fun.RMFMKL.ml.m,summary.plot=F,  
n=100)  
  
> mcycle.p1.fit.quant<-GLD.quantreg(seq(0.1,.9,length=9),  
+ mcycle.p1.fit,slope="fixed")  
> mcycle.p2.fit.quant<-GLD.quantreg(seq(0.1,.9,length=9),  
+ mcycle.p2.fit,slope="fixed")  
  
> plot(mcycle,ylim=c(-150,75))  
  
> sapply(1:9,function(i)  
+ lines(mcycle$times,c(  
+ cbind(1,bs(mcycle[which(mcycle$times<cutoff),]$times,  
+ df=8))%%mcycle.p1.fit.quant[1:9,i],  
+ cbind(1,bs(mcycle[which(mcycle$times>=cutoff),]$times,  
+ df=15))%%mcycle.p2.fit.quant[1:16,i])))
```

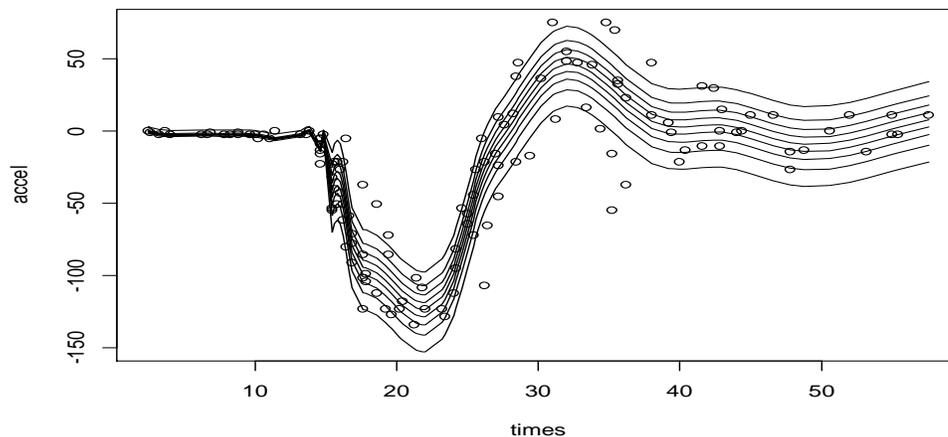


Figure 5. Simulated motorcycle accident dataset

Modified Crime Data

A fair exposition of G λ D regression needs to include a discussion of its weaknesses. Being a regression model involving numerical computations, the initial values play an important role in deriving the final model results. While all numerical optimization problems suffer from bad choice of initial values and it is a general limitation of the Nelder-Mead optimization method, it is possible to still attain the attractive robust property of G λ D regression by simply choosing a better set of initial values.

This example used the crime dataset from UCLA website³ and examined the relationship between crime rate and percentage of single family parents. To examine the behavior of regression models under extreme outliers, two data points were altered in this illustration. The coding involved in extracting and altering the dataset is given below.

```

> require(foreign)
> require(MASS)
> require(GLDreg)
> require(quantreg)

> cdata <- read.dta("http://www.ats.ucla.edu/stat/data/crime.dta")
> mcdata<-subset(cdata,select=c("crime","single"))

# Altering the data to create extreme outliers

> mcdata[51,1]<-10
> mcdata[25,2]<-22

```

If the modeler begins with RS G λ D using MLE using the default initial values, the resulting G λ D regression line is shown in panel A of [Figure 6](#), which is perhaps the worst model among all others (linear regression, robust regression, and quantile regression). This happens because the default initial values was based on linear regression and a less-than-optimal line was found using the Nelder-Mead simplex algorithm.

```

> par(mfrow=c(2,2))

> plot(crime~single,data=mcdata,yLab="Violent crimes per 100,000 people",

```

GLDreg IN R

```
+ xlab="Percentage of population that are single parents",
+ main="GLD Regression fails due to
+ improper initial value selection (A)")

> abline(rlm(crime~single,data=mcdata),lty=2,lwd=3)
> abline(lm(crime~single,data=mcdata),lty=4,lwd=3)
> abline(rq(crime~single,data=mcdata),lty=3,lwd=2)
> abline(GLD.lm(crime~single,data=mcdata,
+ param="rs",fun=fun.RPRS.ml.m,diagnostics=FALSE)[[3]][1:2],
+ lty=1,lwd=3)

> legend("topright",c("Standard Regression","Robust Regression",
+ "Quantile Regression","GLD Regression"),
+ lty=c(2,4,3,1),lwd=c(3,3,2,3),bg="white")
```

It is possible to improve the quality of fit by using different initial values. In the following, *init1* uses quantile regression coefficients, *init2* uses robust regression coefficients, and *init3* uses GLD regression coefficients obtained by removing the outliers at the bottom right corner of the graph.

```
> init1<-rq(crime~single,data=mcdata)$coeff
> init2<-rlm(crime~single,data=mcdata)$coeff
> init3<-GLD.lm(crime~single,
+ data=rbind(mcdata[1:24,],mcdata[26:50,]),
+ param="rs",fun=fun.RPRS.ml.m,diagnostics=FALSE)[[3]][1:2]
```

The modeler can then refit RS GLD regression model using maximum likelihood estimation by using these initial values using the following code.

```
> plot(crime~single,data=mcdata,
+ ylab="Violent crimes per 100,000 people",
+ xlab="Percentage of population that are single parents",
+ main="GLD Regression with quantile regression coefficients
+ as initial values (B)")

> abline(rlm(crime~single,data=mcdata),lty=2,lwd=3)
> abline(lm(crime~single,data=mcdata),lty=4,lwd=3)
> abline(rq(crime~single,data=mcdata),lty=3,lwd=2)
```

STEVE SU

```
> abline(GLD.lm(crime~single,data=mcdata,  
+ param="rs",fun=fun.RPRS.ml.m,diagnostics=FALSE,  
+ init=init1)[[3]][1:2],lty=1,lwd=3)  
  
> plot(crime~single,data=mcdata,  
+ ylab="Violent crimes per 100,000 people",  
+ xlab="Percentage of population that are single parents",  
+ main="GLD Regression with robust regression coefficients  
+ as initial values (C)")  
  
> abline(rlm(crime~single,data=mcdata),lty=2,lwd=3)  
> abline(lm(crime~single,data=mcdata),lty=4,lwd=3)  
> abline(rq(crime~single,data=mcdata),lty=3,lwd=2)  
> abline(GLD.lm(crime~single,data=mcdata,  
+ param="rs",fun=fun.RPRS.ml.m,diagnostics=FALSE,  
+ init=init2)[[3]][1:2],lty=1,lwd=3)  
  
plot(crime~single,data=mcdata,  
+ ylab="Violent crimes per 100,000 people",  
+ xlab="Percentage of population that are single parents",  
+ main=" GLD Regression using modified data  
fitted by GLD regression as initial values (D)")  
> abline(rlm(crime~single,data=mcdata),lty=2,lwd=3)  
> abline(lm(crime~single,data=mcdata),lty=4,lwd=3)  
> abline(rq(crime~single,data=mcdata),lty=3,lwd=2)  
> abline(GLD.lm(crime~single,data=mcdata,  
+ param="rs",fun=fun.RPRS.ml.m,diagnostics=FALSE,  
+ init=init3)[[3]][1:2],lty=1,lwd=3)
```

The results are shown in Panel B to D in [Figure 6](#). In these cases, the GLD regression line is now the most robust among all regression lines. The importance of checking the resulting fit and trying out different sets of initial values for optimization is highlighted in this example.

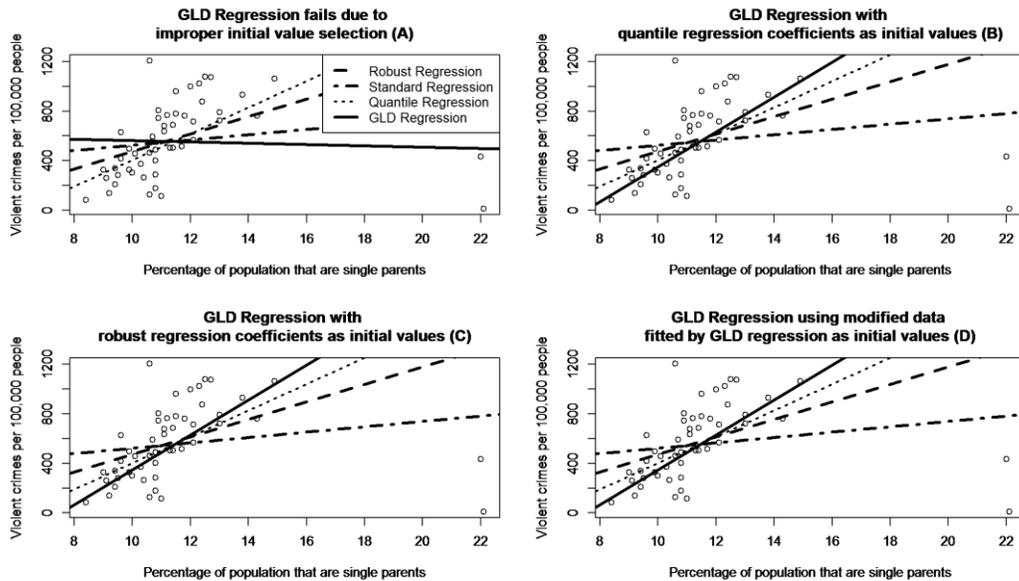


Figure 6. Modified crime data set

Occupation Dataset

The last example involves modelling beyond a single explanatory variable on the Duncan dataset from `car` in R. In this case, a FKML $G\lambda D$ regression using L moment estimation was fitted to the dataset with initial values taken from robust linear regression. It is worthwhile to plot the fitted values against actual values, and to see whether the observations correspond to a 45 degree line from the origin. Large deviations from a linear trend would suggest the model does not fit well with respect to the response variable. The coding to carry the above tasks is given below.

The resulting fit is similar to the robust regression result, and QQ plots and high p -value (Figure 7) suggest the $G\lambda D$ regression fit is a good one. Statistical properties obtained (Figure 8) suggest that the most important variable related to income is prestige. Figure 9 shows that, while the fitted values are not extremely accurate with respect to actual values, the general linear trend is still observed, indicating that the linear form of the model is appropriate.

As a final remark, it can be worthwhile to fit quantile regression lines for fitted values against actual values to give a range of likely actual values that could be obtained using the linear model.

```
> library(MASS)
```

```

> library(car)
> data(Duncan)

> job.fit.full<-GLD.lm.full(income~education+prestige,
+ data=Duncan,param="fkml",fun=fun.RFMKL.Lm,
+ init=rlm(income~education+prestige,data=Duncan)$coeff,
+ summary.plot=F)

> summaryGraphics.gld.lm(job.fit.full)

# Plot actual vs fitted observations:

> plot(job.fit.full[[1]]$Fitted,
+ job.fit.full[[1]]$y,xlab="Fitted",ylab="Actual")

> abline(0,1)

```

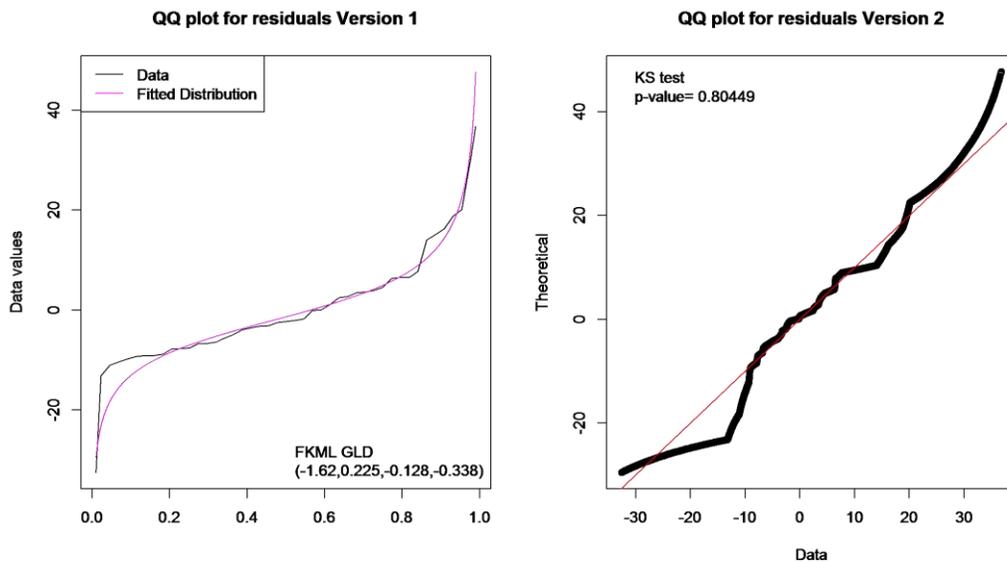


Figure 7. Occupation data set modelling – QQ plots

GLDreg IN R

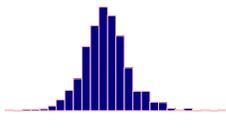
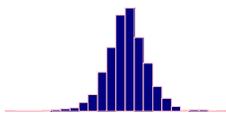
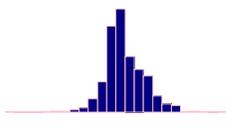
Parameter	Estimate	Density	Summary 95% interval:
(Intercept)	7.69		-3.07, 19.8
education	0.109		-0.13, 0.337
prestige	0.597		0.403, 0.841

Figure 8. Occupation data set modelling – summary plots

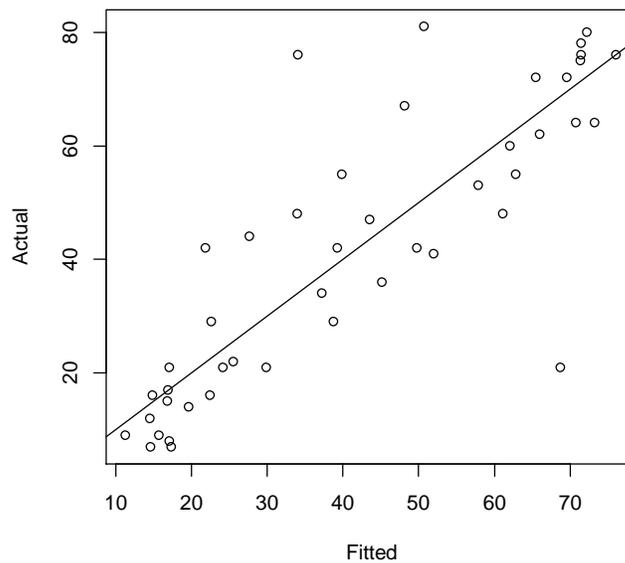


Figure 9. Occupation data set modelling – actual vs. fitted values

Conclusion

The flexibility of G λ D regression models was illustrated; rather than confining the regression model to only examining the mean or median as is the case of linear regression or classic quantile regression model, the G λ D regression models attempt to fit a line that represent a typical value for the dataset which may or may not correspond to standard measures such as the mean or median. Some extensions were also shown beyond the methodology described in Su (2015), and two different optimization schemes (L moment matching and maximum likelihood estimation) were implemented to increase the versatility of G λ D regression in different modelling situations. Potential shortcomings of G λ D regression in initial value selection were illustrated, as well as how different initial values could lead to better regression fits, which is a problem for many model fitting problems involving numerical computations. Since its inception, G λ Ds have been used to model a wide range of empirical data, and the flexibility and robustness of G λ D regression is particularly attractive either as a check for standard results or as a replacement. It is hoped that the introduction of this work would encourage future researchers to develop new methodological improvements to further enhance the usability of G λ Ds in practice.

References

- Asquith, W. (2007). L-moments and TL-moments of the generalized lambda distribution. *Computational Statistics and Data Analysis*, 51(9), 4484-4496. doi: 10.1016/j.csda.2006.07.016
- Freimer, M., Kollia, G., Mudholkar, G. S., & Lin, C. T. (1988). A Study of the generalized Tukey lambda family. *Communications in Statistics – Theory and Methods*, 17(10), 3547-3567. doi: 10.1080/03610928808829820
- Karian, Z., Dudewicz, E., & McDonald, P. (1996). The extended generalized lambda distribution systems for fitting distributions to data: history, completion of theory, tables, applications, the "final word" on moment fits. *Communications in Statistics – Computation and Simulation*, 25(3), 611-642. doi: 10.1080/03610919608813333
- Karvanen, J., & Nuutinen, A. (2008). Characterizing the generalized lambda distribution by L-moments. *Computational Statistics and Data Analysis*, 52(4), 1971-1983. doi: 10.1016/j.csda.2007.06.021

Ramberg, J., & Schmeiser, B. (1974). An approximate method for generating asymmetric random variables. *Communications of the Association for Computing Machinery*, 17(2), 78-82. doi: 10.1145/360827.360840

Su, S. (2007a). Fitting single and mixture of generalized lambda distributions to data via discretized and maximum likelihood methods: GLDEX in R. *Journal of Statistical Software*, 21(9), 1-17. doi: 10.18637/jss.v021.i09

Su, S. (2007b). Numerical maximum log likelihood estimation for generalized lambda distributions. *Computational Statistics and Data Analysis*, 51(8), 3983-3998. doi: 10.1016/j.csda.2006.06.008

Su, S. (2010). Chapter 15: Fitting GLD to data using the GLDEX 1.0.4 in R. In Z. Karian & E. Dudewicz (Eds.), *Handbook of distribution fitting methods with R* (585-608). Boca Raton: CRC Press/Taylor & Francis.

Su, S. (2015). Flexible parametric quantile regression model. *Statistics and Computing*, 25(3), 635-650. doi: 10.1007/s11222-014-9457-1

Footnotes

1. The theoretical mean for RS G λ D is

$$\lambda_1 + \frac{\frac{1}{\lambda_3 + 1} - \frac{1}{\lambda_4 + 1}}{\lambda_2}$$

and the theoretical mean for FKML G λ D is

$$\lambda_1 - \frac{\frac{1}{\lambda_3 + 1} - \frac{1}{\lambda_4 + 4}}{\lambda_2}$$

2. CRAN can be accessed at <http://cran.r-project.org/>
3. The UCLA crime dataset can be accessed at <http://www.ats.ucla.edu/stat/data/crime.dta>