rockchalk package

Paul E. Johnson[1] [2]
<pauljohn@ku.edu>

[1]Department of Political Science

[2]Center for Research Methods and Data Analysis, University of Kansas

2013

Outline

Outline

Thanks for Joining

Thanks to Ray DiGiacomo, Jr & OC RUG for organizing

Downloads:

http://pj.freefaculty.org/guides {all my lectures on anything}

.../Rcourse/rockchalk-2013 {this lecture, source code, LyX doc, etc}

http://pj.freefaculty.org/R: Rtips, links to other R stuff

Why Make a Package?

- Avoid a riot after an influx of 40 MA-bound behavioral scientists into my regression class
  - Honestly, I'd rather teach R programming, but
  - I can understand the view that statistics exists apart from R
- Package has "convenience" functions for

  Me preparing lectures
  Them doing papers (with nice looking graphs!)
- I had distributed functions before, but never made a package

What do you expect in rockchalk?

- Functions for difficult/tedious/hard-to-teach chores
- Verbose documentation, (too) many examples
- vignettes
  - *"rockchalk"*: Dicussion & demonstration of package
  - *"Rchaeology"*: Deep insights into R programming I accumulate while working on the package
  - "*Rstyle*": The style manual I wish R Core would adopt
- Hidden value added: the examples folder in the package install directory includes some special educational R examples (look for `noWords-001.R` and `centeredRegression.R`)

Where is the hard work in version 1.8?

- `predictOMatic()`. *Flexible* way to demonstrate marginal effects of predictors. Goal: make it easy to understand regression as a translation of inputs into predicted values (and uncertainty)
- Scan fitted regressions, create newdata objects with possible predictor values ("divider" algorithms to create focal values for consideration).

Outline

Make a Presentable Table Describing The Data

- Assignment: create a summary table for your research article
- R's `summary()`
  - does not include diversity estimates
  - does not separate numeric from factor variables in the report
  - does not provide output in a usable format
- rockchalk `summarize()`
  - does

Example

```
(datsum <- summary(dat))
```

```
     income            educ            age            religion     gender
 Min.   :-56816   Min.   : 2.00   Min.   : 9.00   cath   :177   female:532
 1st Qu.: -2225   1st Qu.: 8.00   1st Qu.:19.00   jewish : 87   male  :468
 Median : 10565   Median :10.00   Median :22.00   muslem : 94
 Mean   : 10473   Mean   :10.02   Mean   :22.04   other  :294
 3rd Qu.: 23772   3rd Qu.:12.00   3rd Qu.:25.00   prot   :169
 Max.   : 77189   Max.   :21.00   Max.   :37.00   roman  :105
 NA's   :80       NA's   :40                      NA's   : 74
```

- Can you wrestle that into a paper? I can't! It has text and values combined

```
datsum[ ,1]
```

```
"Min.   :-56816  " "1st Qu.: -2225  " "Median : 10565  " "Mean   : 10473  " "3rd Qu.: 23772  " "Max.   : 77189  " "NA's   :80  "
```

- Default output from summarize separates numerics & factors, alphabetizes

Example ...

```
datsum2 <- summarize(dat)
```

The result object datsum2 is a list with 2 parts, a numeric matrix part and a factor variable display.

- The numerics are a matrix, easy to take rows or columns to put into a paper

```
datsum2$numerics
```

	age	educ	income
0%	9.000	2.000	−56820
25%	19.000	8.000	−2225
50%	22.000	10.000	10570
75%	25.000	12.000	23770
100%	37.000	21.000	77190
mean	22.040	10.020	10470
sd	4.556	3.056	19630
var	20.760	9.337	385400000
NA's	0.000	40.000	80
N	1000.000	1000.000	1000

Example ...

- The factors are a separate list

```
datsum2$factors
```

```
             gender                      religion
female          : 532.000   other            : 294.0000
male            : 468.000   cath             : 177.0000
NA's            :   0.000   prot             : 169.0000
entropy         :   0.997   roman            : 105.0000
normedEntropy:    0.997   (All Others)     : 181.0000
N              :1000.000   NA's             :  74.0000
                           entropy          :   2.4414
                           normedEntropy:    0.9445
                           N                :1000.0000
```

- Indicators of central tendency and dispersion are included in both displays
- Try `summarizeNumerics()` and `summarizeFactors()` to get just one or the other.

Sidenote: recoding a factor

- Note the religion variable has levels "cath" and "roman", which was a data entry error. Catholic and Roman Catholic represent the same idea
- Did you ever try to write R code to fix that (without killing yourself)?
- Try `rockchalk::combineLevels()`:

```
dat$religion2 <- combineLevels(dat$religion,
    c("cath", "roman"), "cath")
```

```
The original levels cath jewish muslem other
    prot roman
have been replaced by jewish muslem other prot
    cath
```

Sidenote: recoding a factor ...

```
table(dat$religion2, dat$religion, dnn = c("religion2", "
    religion"))
```

```
          religion
religion2 cath jewish muslem other prot roman
   jewish    0     87      0     0    0     0
   muslem    0      0     94     0    0     0
   other     0      0      0   294    0     0
   prot      0      0      0     0  169     0
   cath    177      0      0     0    0   105
```

Outline

Need a Nice Looking Regression Table?

- Each student should not invent a unique report format for regressions.
- MS Word users especially tempted to "finger paint" with fonts and formats.
- Solution: provide usable LaTeX tables (added benefit: bait to get them to use LaTeX)
- rockchalk-1.8 provides HTML backend as well (compromise with reality)

For many years, outreg was a function in search of a package

- Dave Armstrong (then at U. Maryland student) gave me the outreg idea 10 years ago
- I wrote up a function that more-or-less worked, distributed it, revised it as my R programming skills improved
- I didn't know there was "outreg" module for Stata. . ..

outreg example usage

I fit a regression using a subset of the American National Election Study 2004 (ICPSR), which I called "mydta1"

```
mod1age <- lm(th.bush.kerry ~ V043250, data =
    mydta1)
outreg(mod1age, tight = FALSE, modelLabels =
   c("Age as Predictor"))
```

Produces this LaTeX Markup

```
\begin{tabular}{*{3}{l}}
 \hline
                &\multicolumn{2}{c}{Age as Predictor}    \\
                &Estimate &(S.E.) \\
 \hline
 \hline
  (Intercept)    & -6.841  &   (4.596) \\
  V043250        &  0.184* &   (0.092) \\
 \hline
N                &1191     &       \\
RMSE             &53.885   &  \\
$R^2$            &0.003    &  \\
 \hline
 \hline

 \multicolumn{2}{l}{${*}  p \le 0.05$   }\\
 \end{tabular}
```

Which LaTeX Renders as

	Age as Predictor	
	Estimate	(S.E.)
(Intercept)	-6.841	(4.596)
V043250	0.184*	(0.092)
N	1191	
RMSE	53.885	
R^2	0.003	

$*p \leq 0.05$

My terminology:

tight = FALSE $\Rightarrow$ $\hat{\beta}$ and $std.err(\hat{\beta})$ are side by side

tight = TRUE $\Rightarrow$ $\hat{\beta}$ and $std.err(\hat{\beta})$ are vertically aligned.

Add Gender

```
## Run a new regression
mod2age <- lm(th.bush.kerry ~ V043250 +
    V041109A, data = mydta1)
## Put 2 regressions in same table
outreg(list(mod1age, mod2age), tight = TRUE,
    modelLabels = c("Age Only","Age With
    Gender"))
```

NB: tight = TRUE

Output To LaTEX

	Age Only Estimate (S.E.)	Age With Gender Estimate (S.E.)
(Intercept)	-6.841	4.628
	(4.596)	(6.527)
V043250	0.184*	0.191*
	(0.092)	(0.092)
V041109A	.	-7.713*
		(3.123)
N	1191	1191
RMSE	53.885	53.77
R^2	0.003	0.008
adj R^2	0.003	0.007

$*p \leq 0.05$

Alternative way to specify model labels (rockchalk 1.8)

```
outreg(list("Age Only" = mod1age, "Age With
    Gender" = mod2age), tight = FALSE)
```

Perhaps more coherent usage: keep labels with models in a list

Output To LaTEX

	Age Only		Age With Gender	
	Estimate	(S.E.)	Estimate	(S.E.)
(Intercept)	-6.841	(4.596)	4.628	(6.527)
V043250	0.184*	(0.092)	0.191*	(0.092)
V041109A	.		-7.713*	(3.123)
N	1191		1191	
RMSE	53.885		53.77	
R^2	0.003		0.008	
adj R^2	0.003		0.007	

$*p \leq 0.05$

Beautify Variable Labels

```
outreg(list("Age Only" = mod1age, "Age With
    Gender" = mod2age), tight = TRUE,
    varLabels = list("V043250" = "Age", "
    V041109A" = "Gender"))
```

Quotation marks optional before equal sign in list; this works too

```
outreg(list("Age Only" = mod1age, "Age With
    Gender" = mod2age), tight = FALSE,
    varLabels = list(V043250 = "Age",
    V041109A = "Gender"))
```

Not necessary to provide new labels for all variables

My Beautiful Table with Lovely Variable Labels

	Age Only		Age With Gender	
	Estimate	(S.E.)	Estimate	(S.E.)
(Intercept)	-6.841	(4.596)	4.628	(6.527)
Age	0.184*	(0.092)	0.191*	(0.092)
Gender	.		-7.713*	(3.123)
N	1191		1191	
RMSE	53.885		53.77	
R^2	0.003		0.008	
adj R^2	0.003		0.007	

$*p \leq 0.05$

Quick R style comment: My opinion

- Students often have urge to rename variables in the analysis itself, to create new dat$gender and dat$age
- I urge them to resist the temptation
- In a team setting, everybody has same input variables with names like V234234, cooperation is frustrated when everybody renames everything
- However, in output, no reader wants to see V234234

What is this Good For?

- Good-enough tables in lectures & term papers
- Possible to "script" together a lot of separate estimates for a lot of different datasets
- Especially when the students start to think they know everything, show *I'm still smarter than you*:
  - http://pj.freefaculty.org/R/gloating/test2
  - http://pj.freefaculty.org/guides/stat/Regression/Multicollinearity/Multicollinearity-1-lecture.pdf

Recent updates to outreg

- I get more emails about `outreg()` than any of the other functions. People want more and more features.
- Compromises so far allow customization of:
  - model "header" labels and variable names
  - the selection of "goodness of fit" indicators in the bottom of the table
  - choice of alpha levels (Previously, I first refused p-values, then insisted only 0.05).
  - HTML output (next slide)

outreg can create html file output

- This is a brand new feature in outreg 1.8 (June, 2013)
  - outreg2HTML() receives outreg results and converts into Web markup.
  - Wrestle that into Word however you like.
    - open the html document File -> Open
    - view the html document in a web browser, copy & paste manually into word (use paste Special HTML).
  - Not as nice looking or as automatic as LaTeX, but I may try harder in future

HTML output

```
or1 <- outreg(list(mod1age, mod2age), tight =
    TRUE, modelLabels = c("Age Only", "Age
    With Gender"))
outreg2HTML(or1, file = "or1-test.html")
```

That creates a file, "or1-test.html". See if your web browser can open it. See if Word can open that. I uploaded a copy you can inspect: http://pj.freefaculty.org/R/or1-test.html

Outline

I want it to be easy to make scatterplots with Predicted Values

192 6. Kernel Smoothing Methods

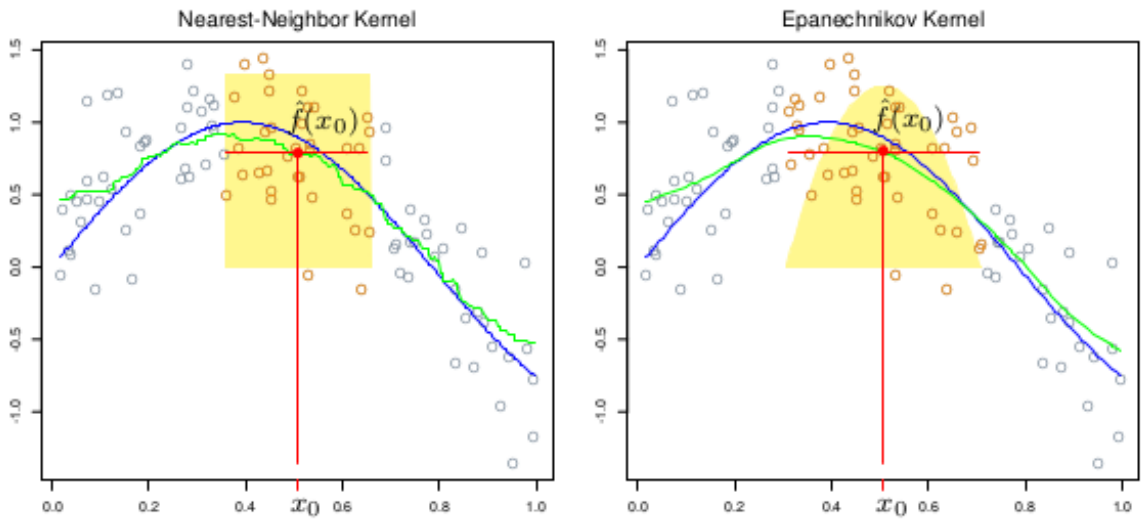


FIGURE 6.1. *In each panel* 100 *pairs* x_i, y_i *are generated at random from the blue curve with Gaussian errors:* $Y = \sin(4X) + \varepsilon$, $X \sim U[0,1]$, $\varepsilon \sim N(0, 1/3)$. *In the left panel the green curve is the result of a* 30*-nearest-neighbor running-mean*

T. Hastie, R. Tibshirani, J. Friedman, *Elements of Statistical Learning: Data Mining, Inference, And Prediction, 2ed* (Springer,

Especially in 3D

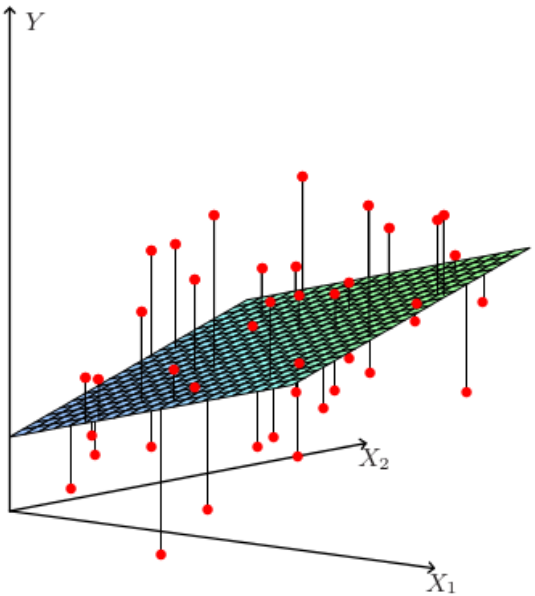
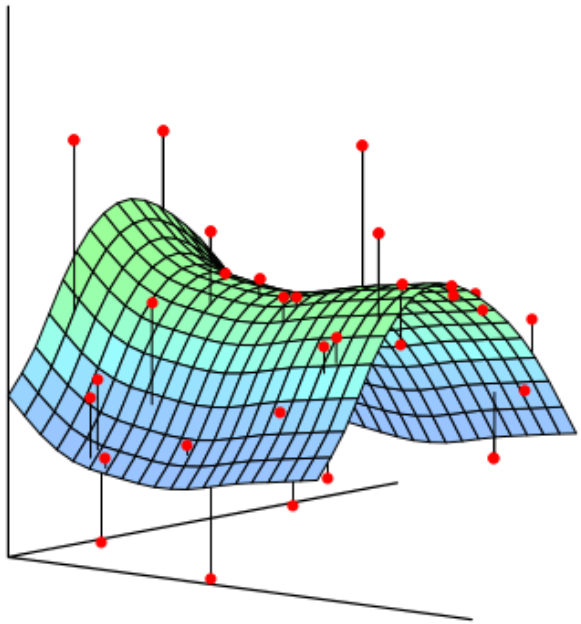


FIGURE 3.1. *Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y.*

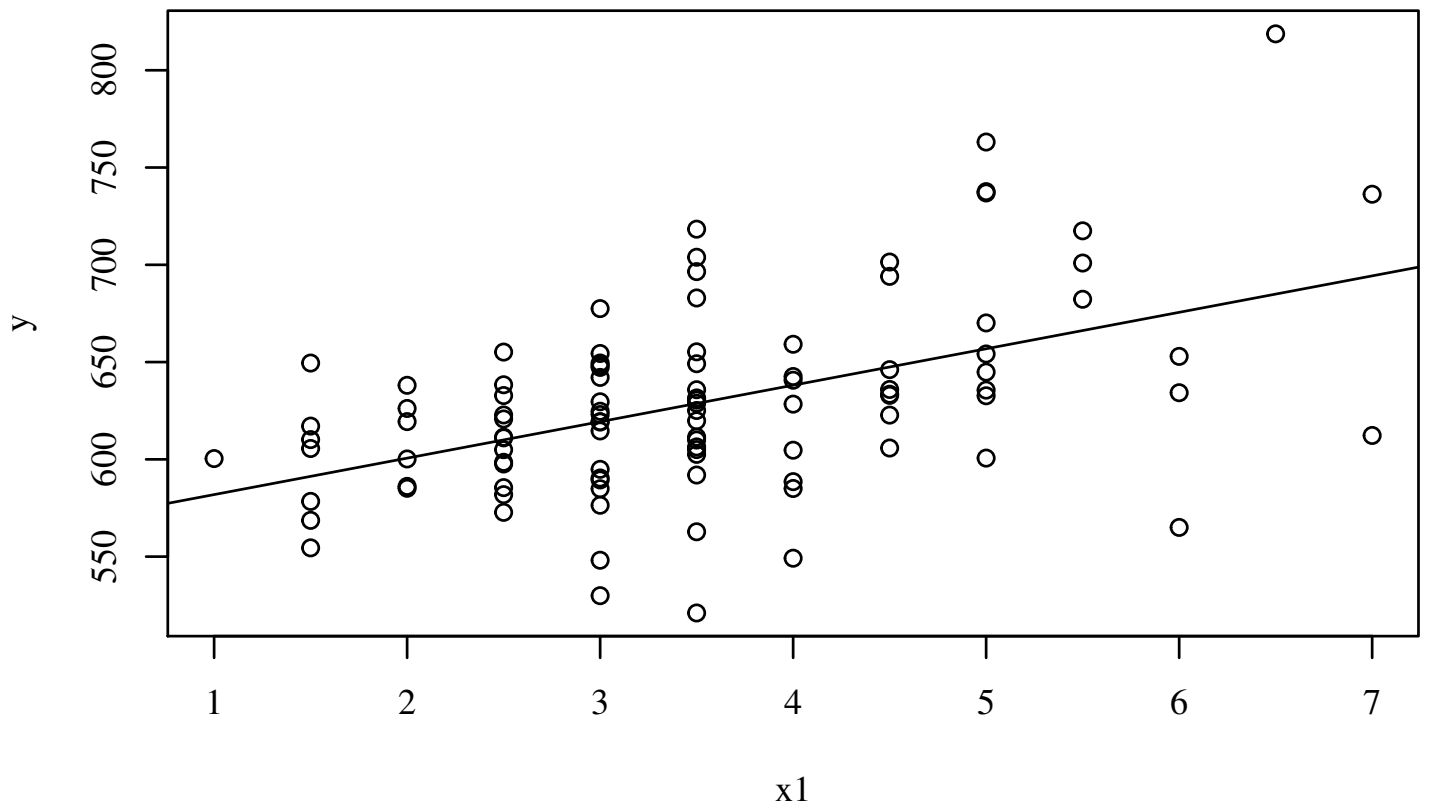


abline is an R staple

- Everybody has done this. (Right?)

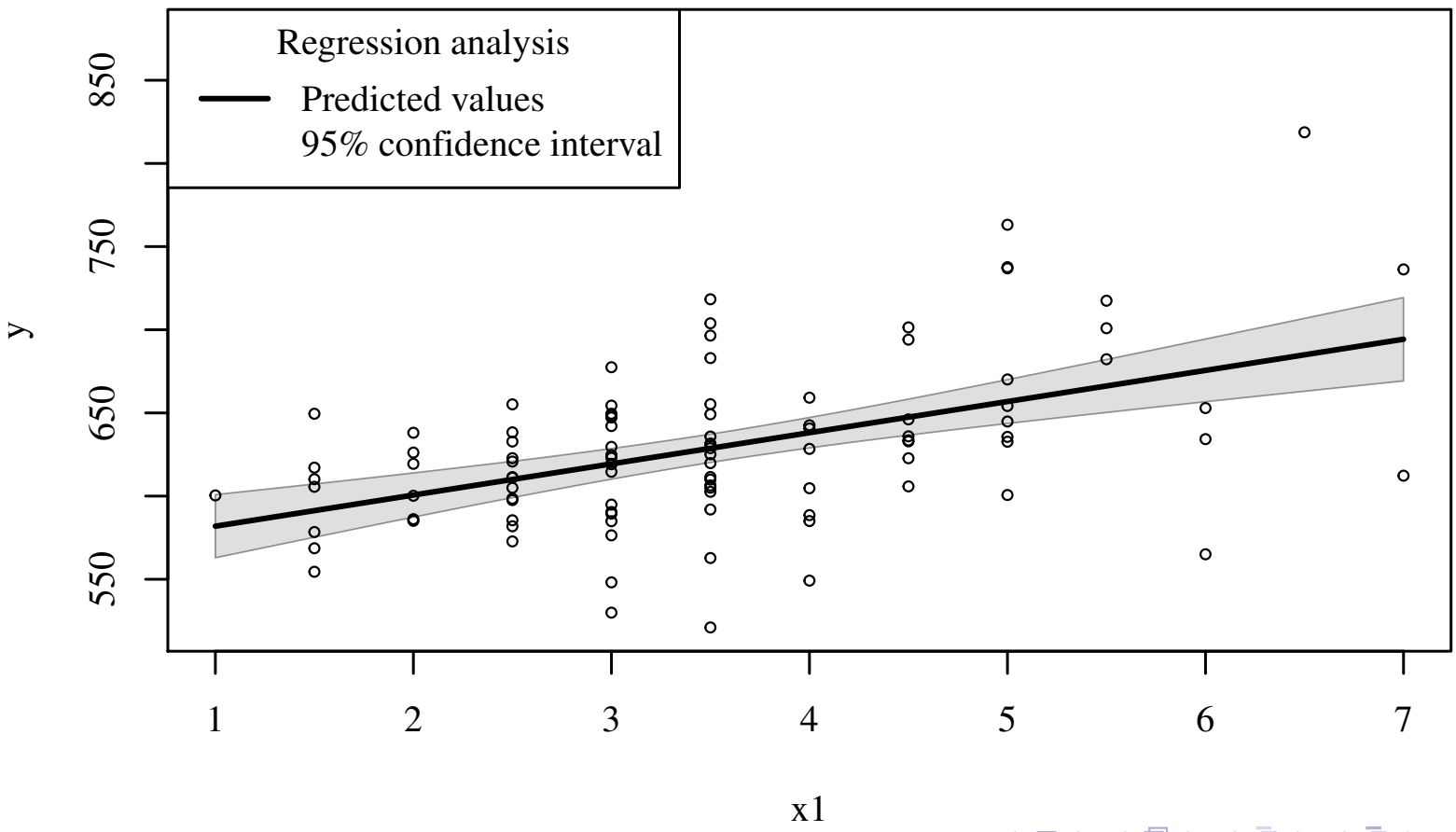
```
mod1 <- lm(y ~ x1, data = dat)
plot(y ~ x1, data = dat)
abline(mod1)
```

abline



I'd rather look at this plot

```
ps1 <- plotSlopes(mod1, plotx = "x1",
    interval = "confidence")
```



abline's fatal flaws

- Suppose the regression model is

```
mod4 <- lm(y ~ x1 + x2 + x3, data = dat)
```

```
mod2 <- lm(y ~ log(x1), data = dat)
```

```
mod2 <- lm(y ~ x1*x2, data = dat)
```

```
mod3 <- glm(y ~ x1, data = dat, family = "
    binomial")
```

- Common answer: abline is an epic fail.

I have taught this ""easy' 3 step procedure" many times

Step 1. Create a "newdata" data frame that has values of the x's for which we want to calculate predictions.

Step 2. Use that newdata object (say, ndat) with the regression model's predict method, with syntax like

```
p1 <- predict(mod1, newdata = ndat)
```

or, if confidence intervals are desired,

```
p2 <- predict(mod1, newdata = ndat,
    interval = "confidence")
```

Frustratingly, p1 and p2 are returned as different object types

Step 3. Wrestle those predicted values into a plot

A sophisticated R user should learn to do that

- I've taught that (look for notes in `http://pj.freefaculty.org/R/WorkingExamples`), but it is too difficult
- I needed to create plots and calculate correlations as described in *Applied Multiple Regression*, by Cohen, Cohen, West, and Aiken, (Routledge, 2002). Students needed lots of R help, some calculations not trivial.
- `plotSlopes()` is the "simple-slope" routine ala CCWA, it was improvised in an emergency, `plotCurves()` & `plotPlane()` used same terminology for consistency.

Syntax

- User fits `m1`, a multiple regression
- Then gives that to `plotSlopes()`, with arguments

plotx:	The name of the variable on the horizontal axis
modx:	The name of a "moderator" variable on which predicted values may depend.
modxVals:	Values of the moderator for which "simple slopes" are desired

- Other arguments to will be passed through to `plot()` and `predict()`
- See the rockchalk vignette.

Difference between plotSlopes and plotCurves

- `plotSlopes()`: for linear models
  - Allows interactions (unlike `termplot()`)
  - Output object can be passed to rockchalk function `testSlopes()`
- `plotCurves()`: for nonlinear models (`lm()` or `glm()`).
  - Complete drop-in replacement for `plotSlopes()`
  - Nonlinear formulae in the predictors (succeeds where termplot fails)
  - Does not create object suitable for `testSlopes()`

Example: moderator is an R factor

- x3 is a predictor with values "left" and "right"
- If there are more predictors, they will be set to their central values (mean or mode) for calculation of predicted values

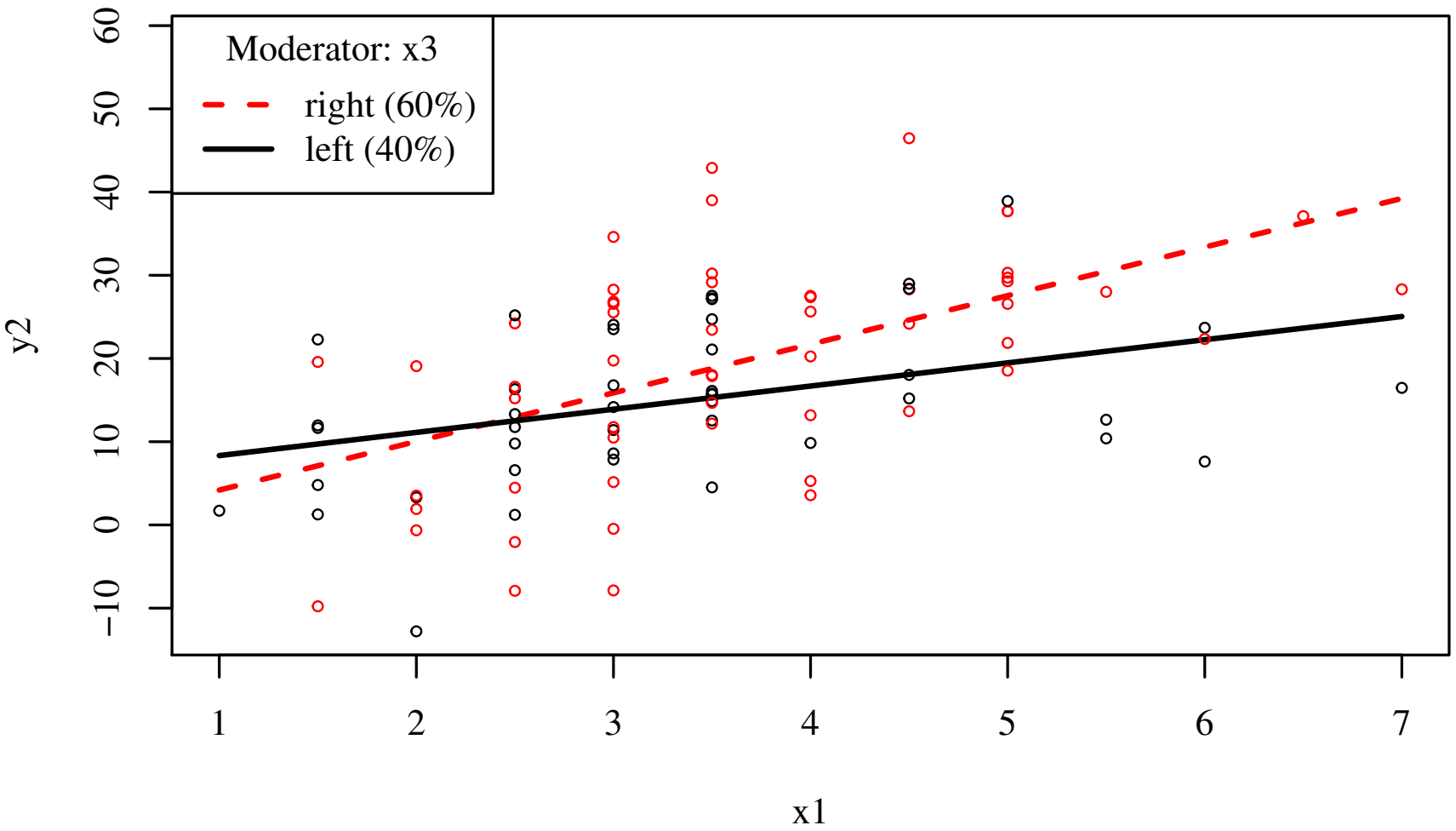
```
mod1 <- lm(y2 ~ x1*x3, data = dat)
ps1 <- plotSlopes(mod1, plotx = "x1", modx =
    "x3")
```

The estimated regression is

	Example Interaction	
	Estimate	(S.E.)
(Intercept)	5.549	(4.199)
x1	2.785*	(1.172)
x3right	-7.197	(6.104)
x1:x3right	3.055	(1.644)
N	100	
RMSE	10.312	
R^2	0.277	
adj R^2	0.254	

$*p \leq 0.05$

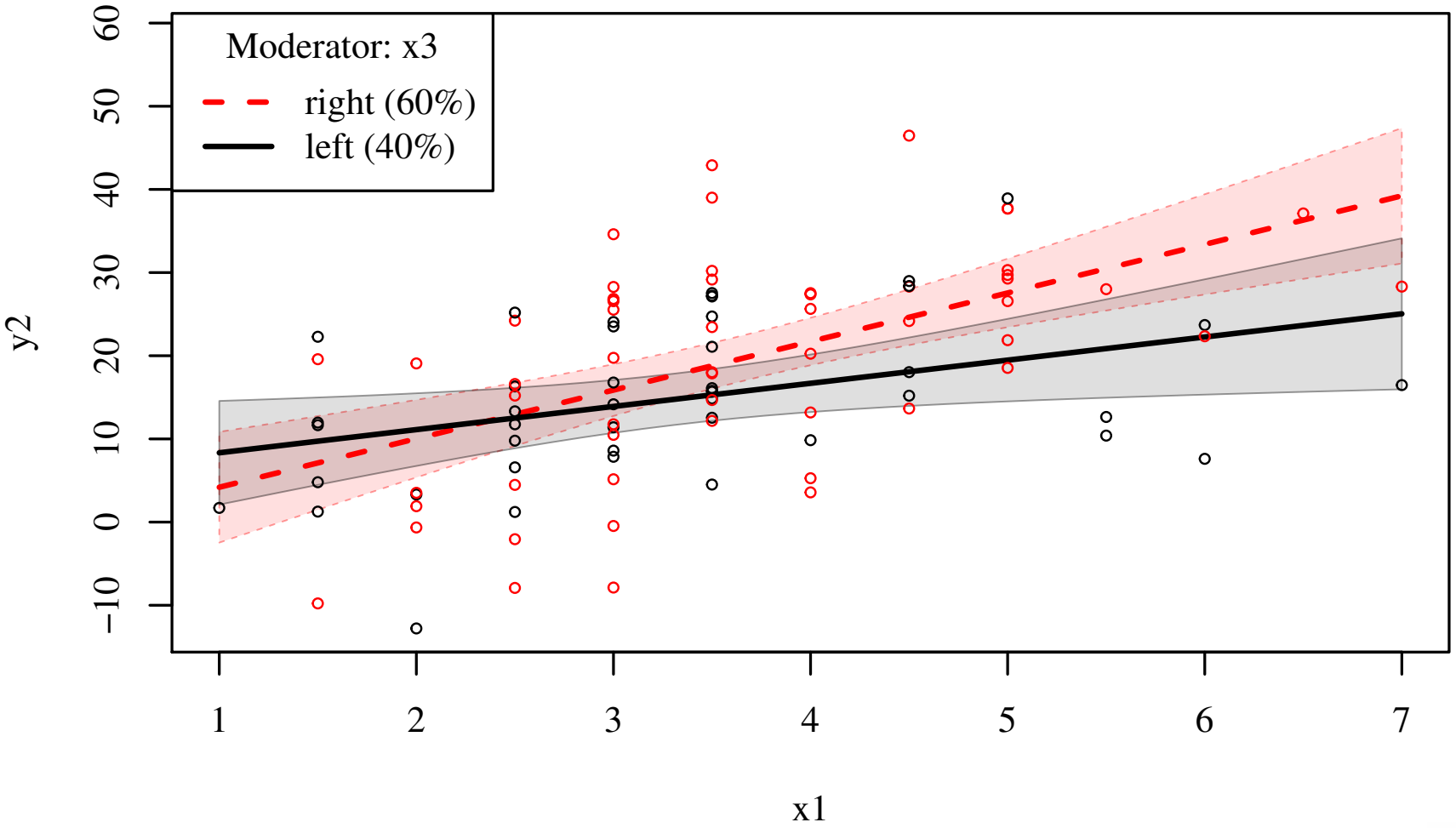
2 lines, one for each value of modx



Add confidence interval

```
ps2 <- plotSlopes(mod1, plotx = "x1", modx =
    "x3", interval = "confidence")
```

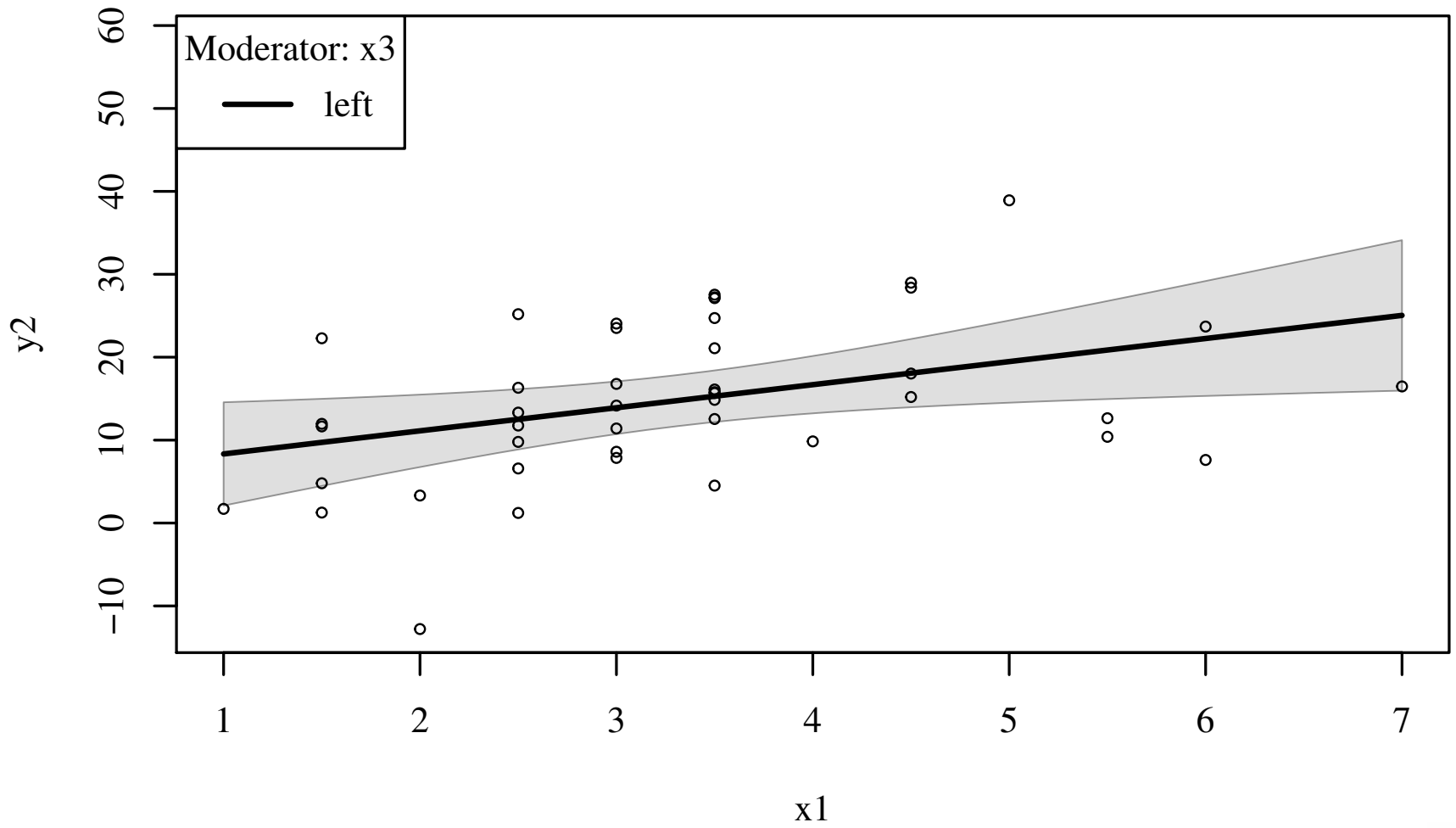
Add confidence interval



Plot a particular group

```
ps3a <- plotSlopes(mod1, plotx = "x1", modx =
    "x3", modxVals = c("left"), interval = "
   confidence")
```

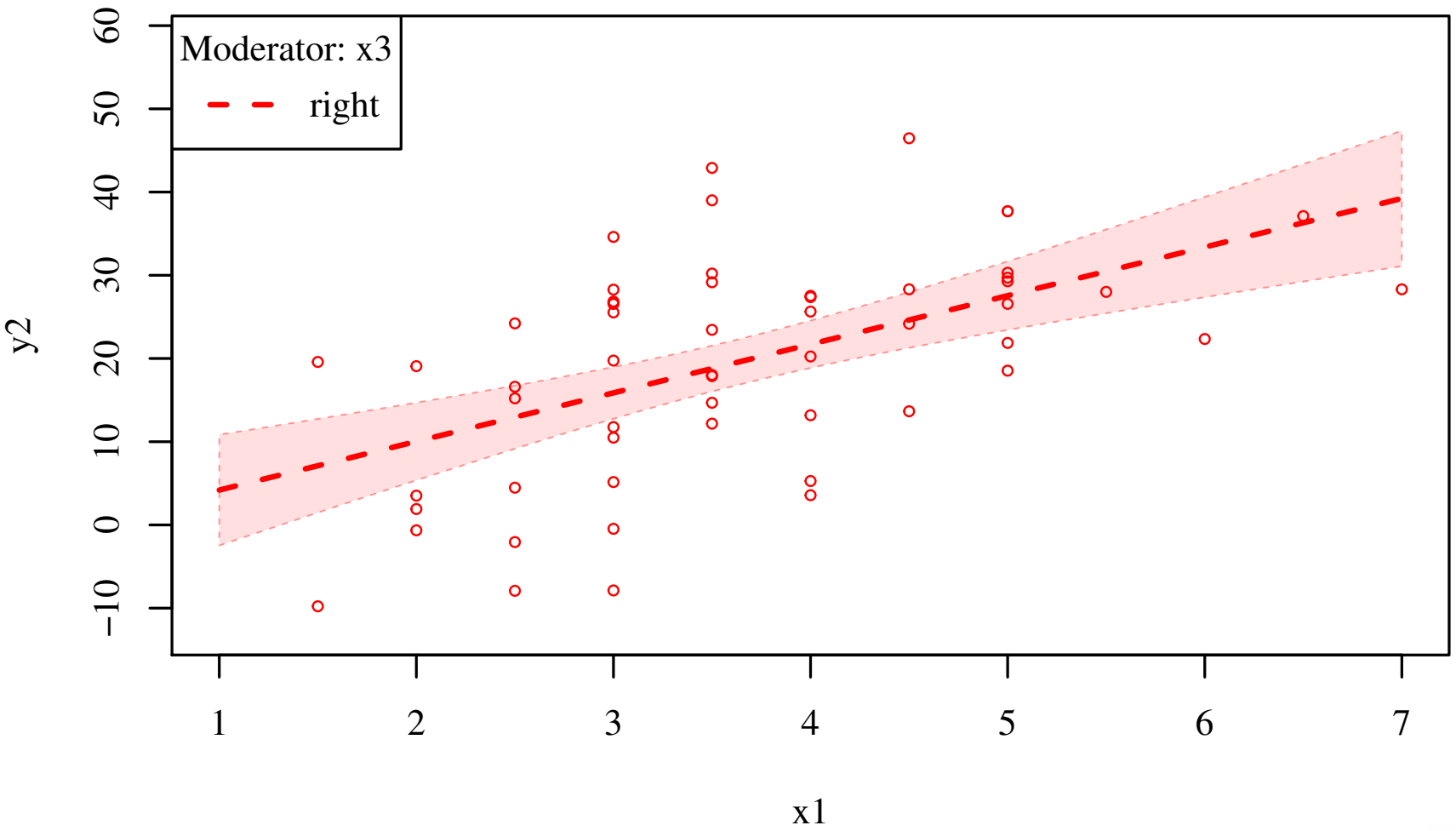
Plot of values for "left" group



Plot of values for "right" group

```
ps3b <- plotSlopes(mod1, plotx = "x1", modx =
    "x3", modxVals = c("right"), interval = "
    confidence")
```

Note my hard work to keep colors consistent



What if the modx variable is numeric?

- When modx is numeric, then particular values need to be chosen for plotting
- Originally, I thought users would explicitly specify values, modxVals
- Have received many user requests, rockchalk 1.8 offers a variety of selection methods.

What if the modx variable is numeric?

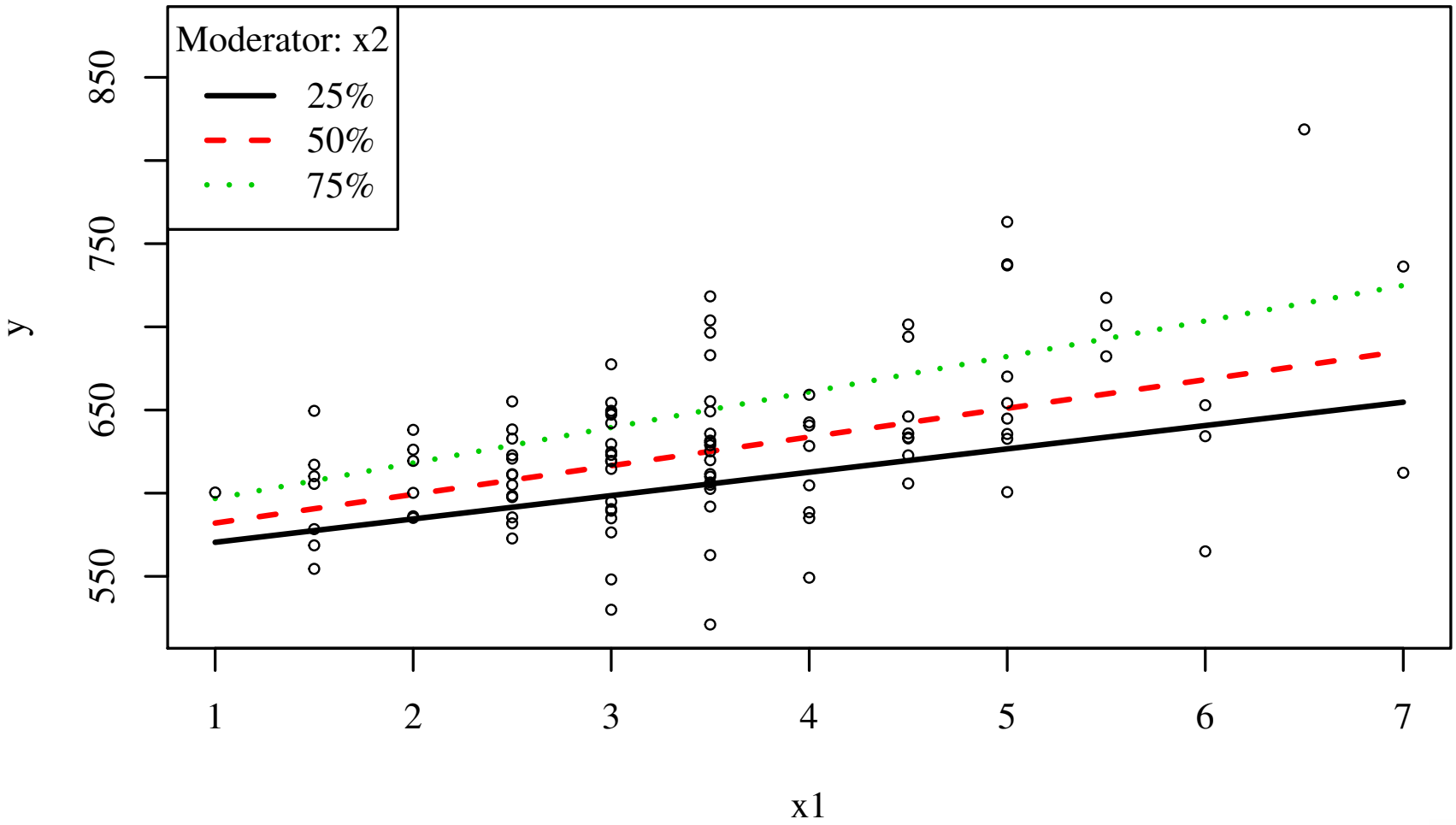
- psychologists generally prefer $mean - std.dev.$, $mean$, $mean + std.dev.$ (or more standard deviations)
- other fields prefer quantiles, such as the 25%, 50% and 75%
- User selects either particular values or a "divider algorithm" to get this done

Defaults

```
mod2 <- lm(y ~ x1*x2, data = dat)
ps5 <- plotSlopes(mod2, plotx = "x1", modx =
    "x2")
```

The default will select the 3 middle quartiles

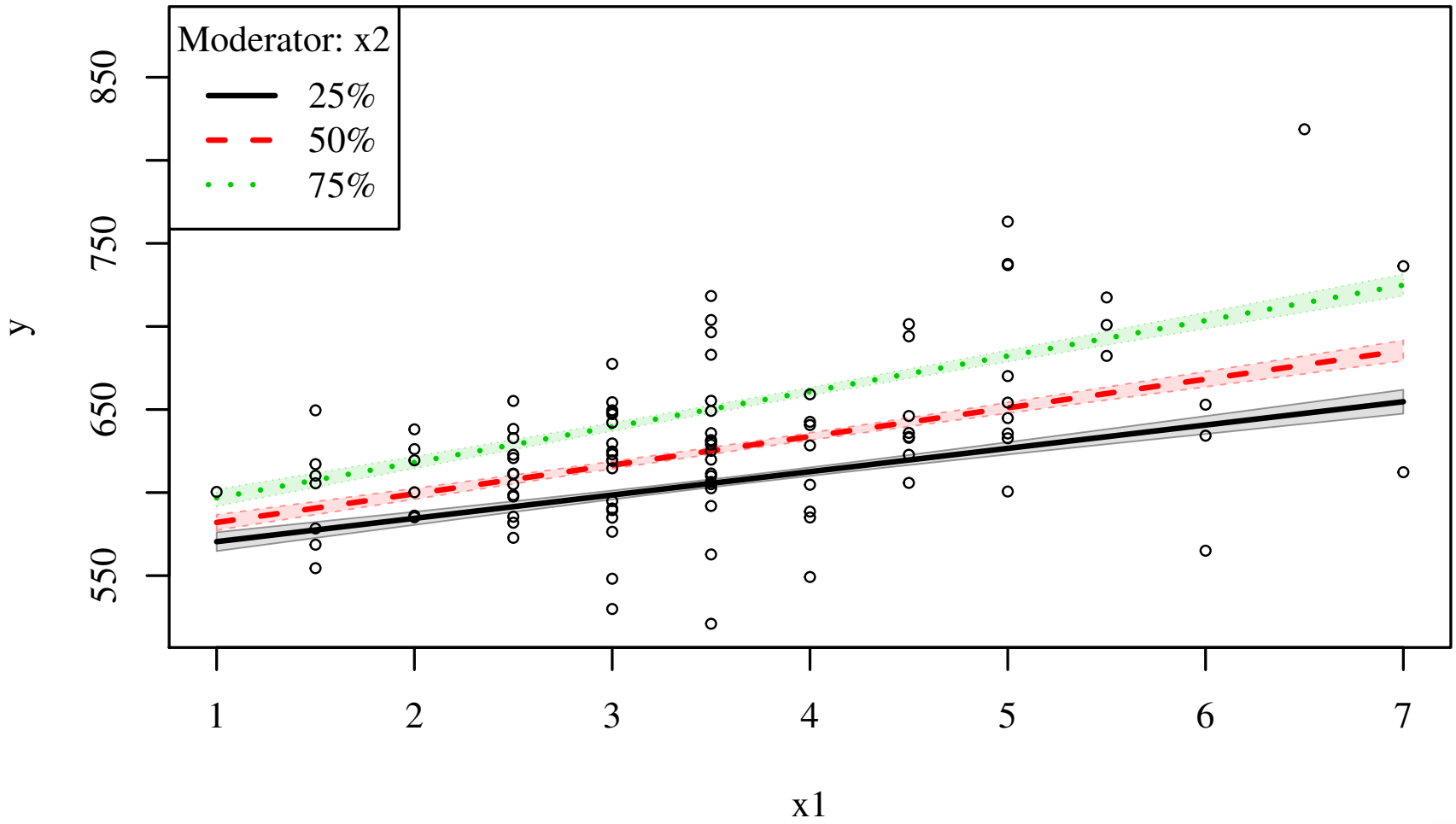
plotSlopes with numeric modx



Add confidence intervals

```
ps5 <- plotSlopes(mod2, plotx = "x1", modx =
    "x2", interval = "confidence")
```

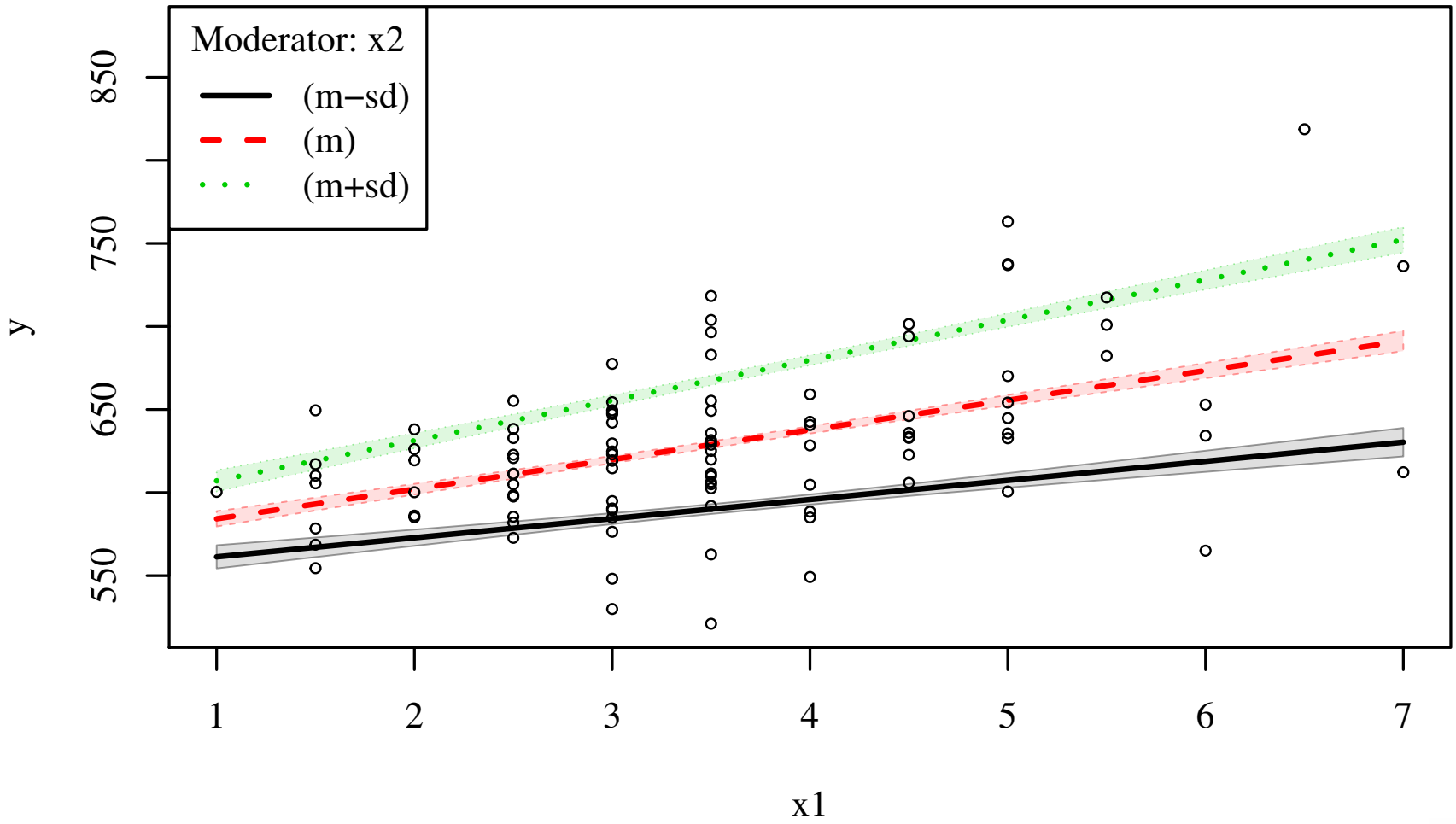
plotSlopes with confidence intervals



Change the algorithm to chose modx values

```
ps7 <- plotSlopes(mod2, plotx = "x1", modx =
    "x2", modxVals = "std.dev.", interval = "
    confidence")
```

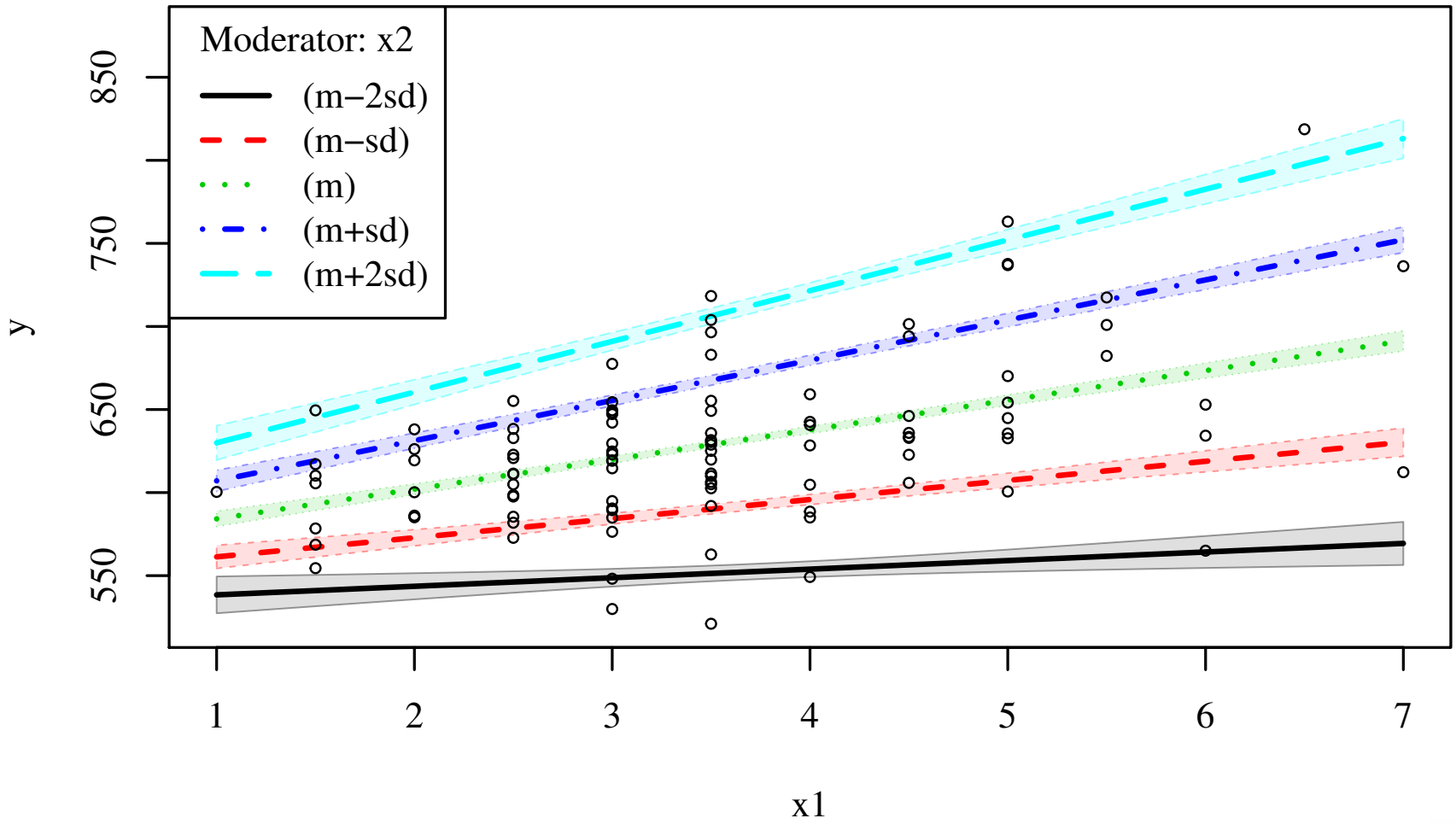
std.dev, +/-



Want a lot of lines? n = 5

```
ps8 <- plotSlopes(mod2, plotx = "x1", modx =
    "x2", modxVals = "std.dev.", n = 5,
    interval = "confidence")
```

5 lines



Conclusion about plotSlopes

- If you don't want a plot, but rather just the newdata matrix and the predicted values, please look at newdata() and predictOMatic().
- plotCurves() can do all of that stuff, and it works with nonlinear models and glm
- Have studied extension to other regression packages.
  - package writers are inconsistent, don't provide predict methods.
  - Conf. Intervals for glm objects controversial

Analyzing Interaction effects

Selway & Templeman (2012) "Myth of Consocionalism?"
Comparative Political Studies
Model has PR*EthnicFractionalization

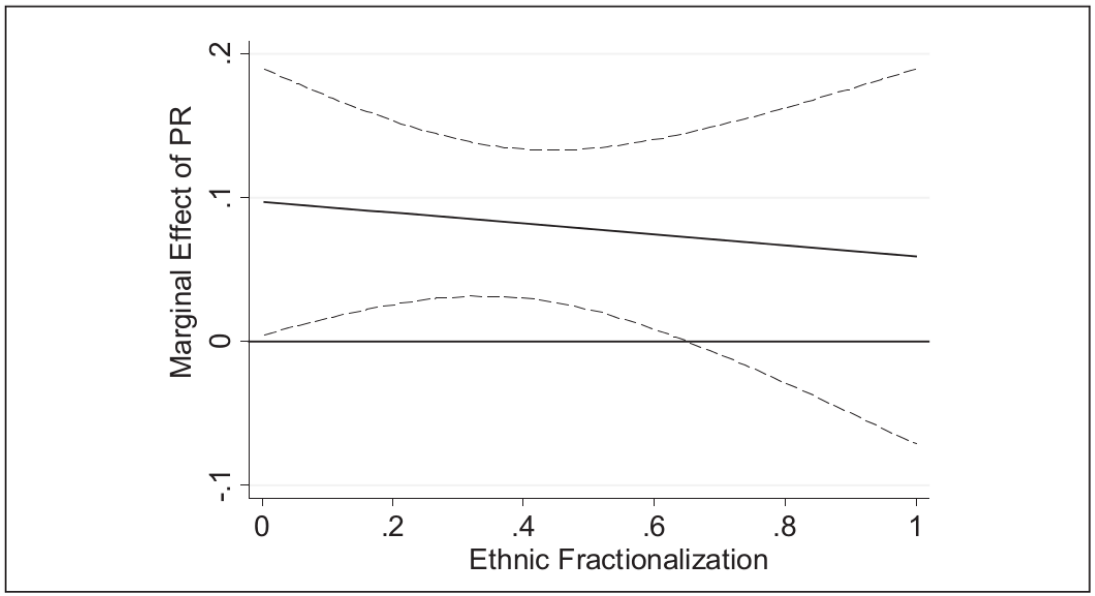


Figure 1. Marginal effect of PR on *Political_Deaths* across levels of ethnic fractionalization

The "marginal effect" is the slope $(\hat{\beta}_{PR} + \hat{\beta}_{EF \cdot PR} EF_i) PR_i$

testSlopes

- `plotSlopes()` creates an output object that allows a 'simple-slopes' analysis of statistical significance.
- If modx is

  categorical: simply calculates the slope of the relationship and tests whether it is different from 0

  numeric: calculates a Johnson-Neyman analysis: for which values of modx would the slope of plotx be different from 0?

  - J-N: if the fitted model is $\hat{y}_i = \hat{\beta}_0 + (\hat{\beta}_1 + \hat{\beta}_3 x2_i)x1_i$, for which values of $x2_i$ is $(\hat{\beta}_1 + \hat{\beta}_3 x2_i)$ statistically significantly different from 0?

testSlopes

```
ps5ts <- testSlopes(ps5)
```

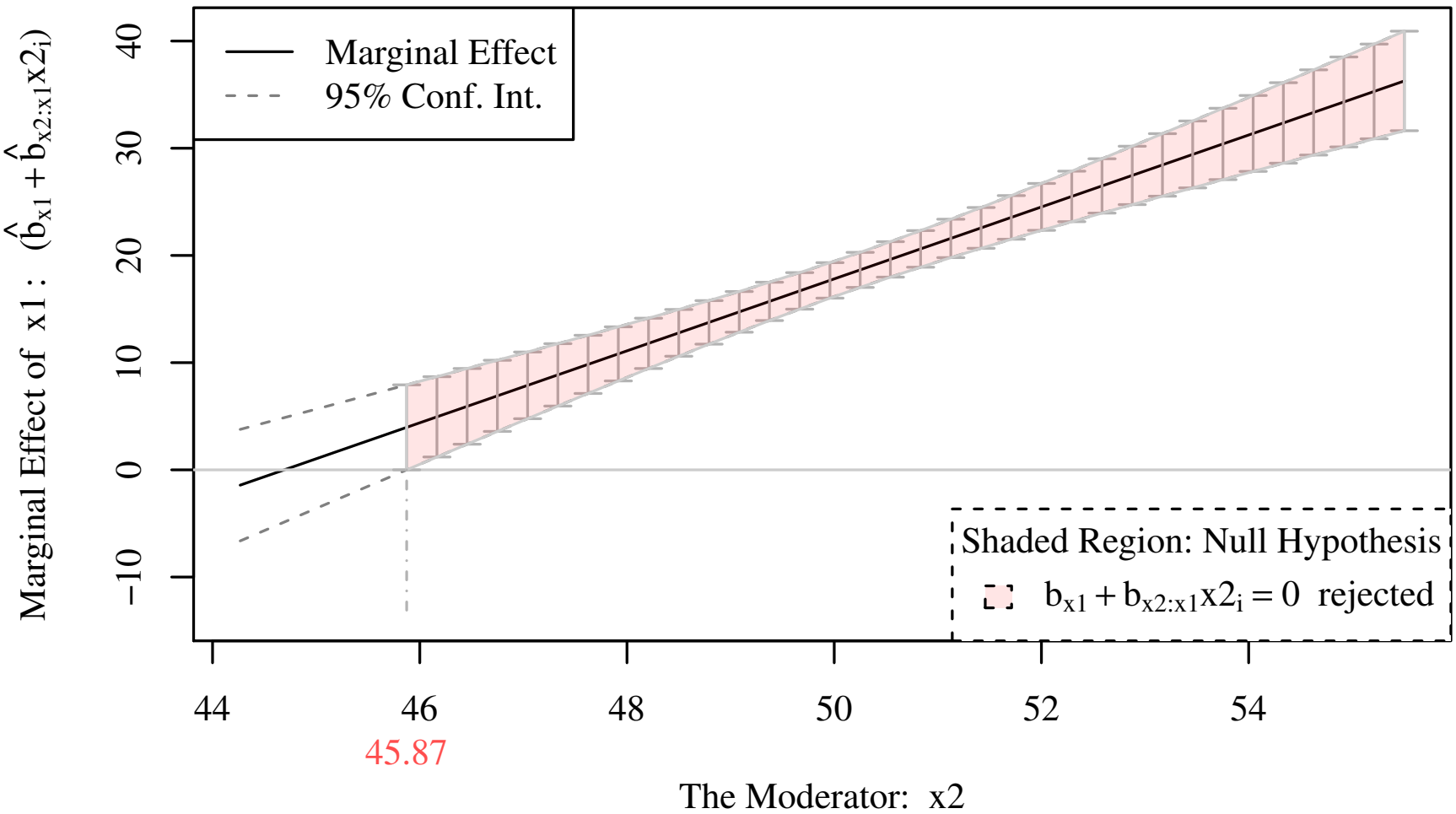
```
Values of x2 OUTSIDE this interval:
      lo       hi
42.79481 45.87360
cause the slope of (b1 + b2*x2)x1 to be
   statistically significant
```

A method for testSlopes objects (plot.testSlopes)

```
plot(ps5ts)
```

rockchalk is an S3 type R package.
If you are uncertain about the significance of S3 and the term "method", I strongly recommend you get a copy of Friedrich Leisch, "Creating R Packages: A Tutorial" (available in CRAN contributed documentation) which has many excellent insights!

plot of a testSlopes object



Note: intended verbosity of labels & legend

Outline

Mean-Center, Residual-Centered Regressions

- Start with
  lm (y ~ x1 * x2 + x3, data = dat)
- which implies
  lm (y ~ x1 + x2 + x1:x2 + x3, data = dat)
- Should it matter if we replace x1 with
  - mean centered values, x1c = (x1 - mean(x1)) by fitting
    lm(y ~ x1c + x2 + x1c:x2 + x3, data = dat)
- Or if we replace x1:x2 by with the
  - "residual centered" value of the interaction term, which is the residual from this regresision?
    lm((x1*x2) ~ x1 + x2, data = dat)

Several authorities say those changes may be important

- Cohen, Cohen, Aichen & West (2002) strongly endorse mean-centering
- Little, T. D., Bovaird, J. A., & Widaman, K. F. (2006). On the Merits of Orthogonalizing Powered and Product Terms: Implications for Modeling Interactions Among Latent Variables. *Structural Equation Modeling,* 13(4), 497-519.

3 ease of use functions in rockchalk

- `standardize()` calculates centered & scaled values of all variables and re-fits the model.
- `meanCenter()` adjusts predictors by subtracting observed means
- `residualCenter()` calculates one variant of orthogonal regression
- rockchalk supplies `print()`, `predict()` and `summary()` methods for these functions

Mean-Center

- Fit some big multiple regression

```
m1 <- lm(someDV ~ x1 + x2 + x3 * x4, data =
    dat)
```

- Center only the interactive predictors

```
m1 <- lm(someDV ~ x1 + x2 + x3c*x4c, data =
    dat)
m1mc <- meanCenter(m1)
```

ends up fitting

```
lm(someDV ~ x1 + x2 + x3c + x4c + x3c:x4c,
    data = dat)
```

Mean-Center

- Center all predictors

```
m1mc2 <- meanCenter(m1, centerOnlyInteractors
    = FALSE)
```

ends up fitting

```
lm(someDV ~ x1c + x2c + x3c + x4c + x3c:x4c,
    data = dat)
```

- Center also the DV

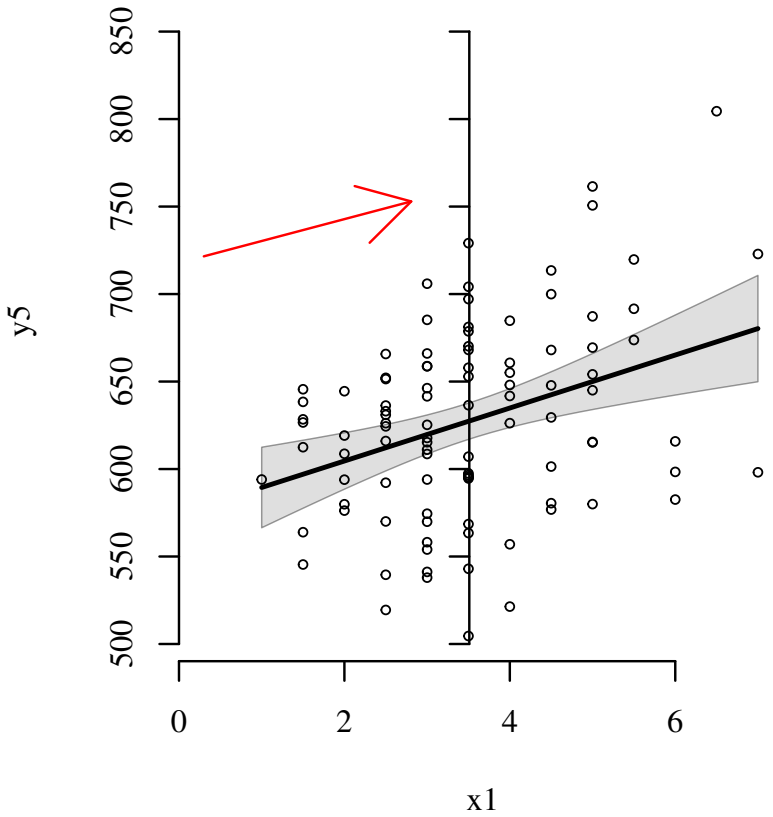
```
m1mc3 <- meanCenter(m1, centerDV= TRUE,
    centerOnlyInteractors = FALSE)
```

ends up fitting

```
lm(someDVc ~ x1c + x2c + x3c + x4c + x3c:x4c,
    data = dat)
```

Why this is fool's gold

- Changing a predictor column from X_i to $X_i - 5$ cannot improve statistical precision.
- It simply re-positions the Y axis.
  - Slope same, standard error of slope same
  - Intercept is "bigger"
  - Predicted value at Y axis is more precise, due to hour-glass shape of CI



I was not so sure about residual centering

- The residualCenter() function leaves the linear terms in the model unchanged, but re-constructs interactive variables, replacing `x3:x4` with the residual from $\mathrm{lm}(\mathrm{x3}^{*}\mathrm{x4} \sim \mathrm{x3}{+}\mathrm{x4})$, which I'm calling "x3.X.x4"

```
m1rc <- lm(someDV ~ x1 + x2 + x3 + x4 + x5 +
    x6 + x5.X.x6 , data = dat)
```

- This is one way to create truly orthogonal columns. Before introduction of QR decomposition, it might have actually been a good way to do so
- Requires some serious fancy coding to make interactions like `x3*x4*x5` work correctly (see also `predict.mcreg()`)

Alternatives seem better, but they are not actually different

- The predicted values are identical
- See the rockchalk vignette, which gives a full argument.
- In directory with this presentation, find the small example file curve-example-1.R

Outline

Other functions worth mentioning

- mcDiagnose: splattering of multicollinearity diagnostics
- getDeltaRsquare, getPartialCor: partial and semi-partial correlations
- See the rockchalk vignette, which gives a full argument.
- In directory with this presentation, find the small example file curve-example-1.R

Outline

What makes package building easier?

`roxygen2` (Hadley Wickham, Peter Danenberg, Manuel Eugster).

Usual R development: one writes R files, and documentation files in a separate directory. Very inconvenient to keep documents in sync with R code.

roxygen2 approach: put documentation in the R files, use functions to extract & format the documents.

Am I competing with "car", "rms", "memisc", "texreg", etc?

- No. "car" and "rms" are established industry leading packages that support widely sold textbooks. Those authors are "up there", I'm "down here."
- No.
  - I'm filling in perceived gaps to create convenience
- Yes. Perhaps I think their
  - jargon is difficult (tough for me $\Longrightarrow$impossible for students)
  - their functions are clumsy, or
  - I think their source code is not clear