`xtable.2`: `xtable`'s Hot Cousin

Casey Crisman-Cox

29 November 2012

1 Preliminaries

A few things to make set up before using the package. First in your LaTeXdocument make sure that you have the following in your preamble:

```
\usepackage{siunitx}
\sisetup{
  input-symbols=(),
table-align-text-post = false,
group-digits=false
}
```

Second, source the function

```
source("Tabling.R")
```

Now the functions are loaded and the necesssary libraries are loaded. They are automatically installed if R can't find them.[1]

2 Using the Function

Let's start by getting some data. Say we have some binary data

```
##Data that is shamelessly taken from the internet
data <- read.csv("http://www.ats.ucla.edu/stat/data/binary.csv")
head(data)

##   admit gre  gpa rank
## 1     0 380 3.61    3
## 2     1 660 3.67    3
## 3     1 800 4.00    1
## 4     1 640 3.19    4
## 5     0 520 2.93    4
## 6     1 760 3.00    2
```

Now let's say we want to run a probit on this, both using `glm` and `maxLik`, as well as a linear probability model. Where for some reason we decide to leave a regressor out of the linear model. First we'll set up our likelihood function and then we'll estimate the three models.

[1] It's only `stringr` and `xtable` so don't worry, I'm not downloading porn to your computer

```
library(maxLik)
Probit<-function(b, X, Y){
     if(!all(X[,1]==1)){
          X<-cbind(1, X)
     }
     Xb<-X %*% b
     ll<-ifelse(Y==1,
                pnorm(Xb, log.p=TRUE),
                pnorm(-Xb, log.p=TRUE))
     return(ll)
} ##Probit likelihood function

####Estimation#####
m1<-lm(admit~gre, data=data)
m2<-glm(admit~gre+gpa, data=data, family=binomial(link="probit"))

x.ml<-with(data, cbind(gre,
                       gpa))
y.ml<-data$admit
b<-runif(3)*0.01
names(b)<-c("(Intercept)", colnames(x.ml))
m3<-maxLik(Probit, start=b, X=x.ml, Y=y.ml, method="BFGS")
```

We now have three models, and we can do a couple of things with it.

2.1 Model Summaries

The first step is to generate the model summary. We do this by using the `mod.sum()` function. We have two options. The first is if we plan on using "classic" xtable style and the second is if we want apsrtable style. For reasons that will be clearer down the road, it is important to make sure that you give all your variables names that match across all the models before running the `mod.sum()` command.

```
info.1<-mod.sum(m1)
cat(info.1[[1]]) ##xtable style--Only handles 1 model

## \hline adj. R$^2$& \multicolumn{4}{c}{0.03159}\\
## $N$ & \multicolumn{4}{c}{400}\\


info.2<-mod.sum(m2)
cat(info.2[[1]]) ##xtable style

## \hline Log $L$ & \multicolumn{4}{c}{-240.1}\\
## $N$ & \multicolumn{4}{c}{400}\\


info.3<-mod.sum(m3)
cat(info.3[[1]]) ##xtable style

## \hline Log $L$ & \multicolumn{4}{c}{-240.1}\\
## $N$ & \multicolumn{4}{c}{400}\\


info.all<-mod.sum(m1, m2, m3, apsr=TRUE)
cat(info.all[[1]]) ##Apsrtable style handle n models
```

```
## \hline  Log $L$ &  &  -240.1 &  -240.1\\
## adj. R$^2$ &  0.03159 &  & \\
## $N$&  400 &  400 &  400\\


info.23<-mod.sum(m2, m3, apsr=TRUE)
cat(info.23[[1]]) ##Apsrtable style

## \hline  Log $L$ &  -240.1 &  -240.1\\
## $N$ &  400 &  400\\
```

As we can see it picks out the class of the model and reports appropriate information (Log-Likelihood for non-OLS models and adj R^2 for OLS). It also only adds in the R^2 row if there is an OLS model in the line-up. You'll notice that in each case I pulled off the first element of list. We'll come back to the second part of the list. We can now put these into a table.

2.2 Table 1: Basic xtable+

The first table style is the basic `xtable` plus the model info. So let's take just `m3` (the `maxLik` object).

```
xtable.2(list(m3),
         caption="xtable Plus",
         label="tab:xtp",
         digits=2,
         add.to.row=list(pos=list(info.3[[2]]),
                         command=info.3[[1]]),
         include.rownames=FALSE,
         caption.placement="top",
         table.placement="H")
```

Table 1: xtable Plus

	Estimate	Std. Error	Z Value	Pr(|z|)
(Intercept)	-2.97	0.63	-4.72	0.00
gre	0.00	0.00	2.66	0.01
gpa	0.44	0.19	2.32	0.02
Log L		-240.1		
N		400		

The results of this command are in table 1. Note that we used the 2nd element of the `mod.sum` output. This tells xtable where to put the model info. It is calculated based on the number of unique variable names. This is why it is important to make sure you have consistent names prior to running the summary command. You can name them whatever you want so long as you're consistent across models.

2.3 Table 2: APSR Style

Let's switch it up and do all three models in one table.

```
xtable.2(list(m1, m2, m3),
         caption="APSR Style",
         label="tab:apsr",
         digits=2,
         add.to.row=list(pos=list(info.all[[2]]), ##Model info placement
```

```
                          command=info.all[[1]]), ## Model info text
          include.rownames=FALSE, ##Needed to exclude row numbers
          caption.placement="top", ##Style choice, not necessary
          table.placement="H", ##Visual choice, not necessary
          apsr=TRUE, ##apsr style
          stars="default", ##default==only at 0.05, use "all" to get full gammit
          sanitize.text.function=function(x){x} ##needed to stop messy screw-ups
)

## Warning:  invalid factor level, NAs generated
## Warning:  Nonstandard alignments in align string
```

Table 2: APSR Style

	Model 1	Model 2	Model 3
(Intercept)	−0.12	−3.00*	−2.97*
	(0.12)	(0.63)	(0.63)
gre	0.00074*	0.00164*	0.00168*
	(0.00020)	(0.00063)	(0.00063)
gpa		0.45*	0.44*
		(0.19)	(0.19)
Log L		−240.1	−240.1
adj. R^2	0.03159		
N	400	400	400

To make the change to APSR style make sure you include all the arguements I mark as needed. The warning messages are normal, just ignore them.

2.4 Table 3: But wait, there's more...

Let's say that you doubt your standard errors and want to change them either with bootstrapping or an analytical correction. Let's do an example with bootstrapping. We'll bootstrap `m3` and have a table with just `m2` and `m3`

```
####bootstrap####
Nboots<-50
boot<-matrix(0,
             nrow=Nboots,
             ncol=ncol(x.ml)+1)
for(i in 1:Nboots){
  rows<-sample(nrow(data), replace=TRUE)
  x.sam<-x.ml[rows,]
  y.sam<-y.ml[rows]
  boot[i,]<-maxLik(Probit,
                   start=b,
                   X=x.sam,
                   Y=y.sam,
                   method="BFGS")$estimate
}
se.3<-apply(boot, 2, sd) ##bootstrapped standard errors
se.2<-sqrt(diag(vcov(m2))) ##the standard errors from glm
```

```
##option 1 use the original coef estimates and new standard errors
xtable.2(list(m2, m3),##report the coef from the models
         se=list(se.2, se.3), ##list of se
         caption="APSR Style + bootstrap",
         label="tab:apsr",
         digits=2,
         add.to.row=list(pos=list(info.23[[2]]), ##Model info placement
                         command=info.23[[1]]), ## Model info text
         include.rownames=FALSE, ##Needed to exclude row numbers
         caption.placement="top", ##Style choice, not necessary
         table.placement="H", ##Visual choice, not necessary
         apsr=TRUE, ##apsr style
         stars="all", ##switching to all
         sanitize.text.function=function(x){x} ##needed to stop messy screw-ups
)

## Warning:  invalid factor level, NAs generated
## Warning:  Nonstandard alignments in align string
```

Table 3: APSR Style + bootstrap

	Model 1	Model 2
(Intercept)	-3.00^{***}	-2.97^{***}
	(0.63)	(0.64)
gre	0.00164^{**}	0.00168^{*}
	(0.00063)	(0.00071)
gpa	0.45^{*}	0.44^{*}
	(0.19)	(0.19)
Log L	-240.1	-240.1
N	400	400

```
##option 2 use just a vector of coefs for m3
coef.m3<-apply(boot, 2, mean)
names(coef.m3)<-names(b) ##Always name things
xtable.2(list(coef(m2), coef.m3),##report the coefs, not models
         se=list(se.2, se.3), ##list of se
         coef=TRUE, ## Note this addition
         caption="APSR Style + bootstrap",
         label="tab:apsr",
         digits=2,
         model.names<-c("New", "Names"), ##Add a vector of model names if you want
         add.to.row=list(pos=list(info.23[[2]]), ##Model info placement
                         command=info.23[[1]]), ## Model info text
         include.rownames=FALSE, ##Needed to exclude row numbers
         caption.placement="top", ##Style choice, not necessary
         table.placement="H", ##Visual choice, not necessary
         apsr=TRUE, ##apsr style
         stars="default", ##default==only at 0.05, use "all" to get full gammit
         sanitize.text.function=function(x){x} ##needed to stop messy screw-ups
)

## Warning:  invalid factor level, NAs generated
```

```
## Warning:  Nonstandard alignments in align string
```

Table 4: APSR Style + bootstrap

	New	Names
(Intercept)	−3.00*	−2.89*
	(0.63)	(0.64)
gre	0.00164*	0.00166*
	(0.00063)	(0.00071)
gpa	0.45*	0.42*
	(0.19)	(0.19)
Log L	−240.1	−240.1
N	400	400