

R is widely used free software similar to *Splus*. (The link **Cran** gives *R* support.) *R* can be downloaded from (www.r-project.org/).

The computer lab is in **Neckers 258** (2 doors from my office), open MTWThF8-4:30. Certain hours are reserved, check at the door. The lab is not open on weekends, holidays. In the computer lab, click on the *Rgui* icon to get into *R*. Then typing *q()* gets you out of *R*.

Useful websites:

<http://www.stat.cmu.edu/~larry/=stat326.02/>

<https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf>

help(fn) and *args(fn)* give information about function *fn*, eg if *fn = glm*.

Most of the following commands can be copied and pasted from the (<http://parker.ad.siu.edu/Olive/M485SASRhw.txt>) link on the Math 485 webpage (near the top of the syllabus), or type the following commands for Poisson regression $Y|SP \sim \text{Poisson}(\exp(SP))$:

```
# Generates data for Poisson regression.
n <- 100
q <- 5
y <- 0 * 1:n
beta <- 0 * 1:q
beta[1:3] <- 1
alpha <- -2.5
x <- matrix(rnorm(n * q), nrow = n, ncol = q)
x <- 0.5*x + 1
SP <- alpha + x%*%beta
y <- rpois(n,lambda=exp(SP))
#fit the Poisson regression
out <- glm(y ~ x[, 1] + x[, 2] + x[, 3] + x[, 4] + x[,5],
family = poisson)
out
summary(out)
#make a response plot
ESP <- x %*% out$coef[-1] + out$coef[1]
Y <- y
plot(ESP,Y)
lines(lowess(ESP,Y))
```

Some regpack functions are at (www.parker.ad.edu/olive/regpack.txt). The following commands are useful for getting these functions and data into *R*.

```
source("http://parker.ad.siu.edu/Olive/regpack.txt")
```

```
source("http://parker.ad.siu.edu/Olive/regdata.txt")
```

From regpack, copy and paste *llrplot* into *R*. Then type "*llrplot(x,y)*".

The matrix command makes an *n* by *q* matrix *x* with $N(0,1)$ entries. The *SP* line makes the sufficient predictor $SP[i] = -2.5 + 1 * x[i, 1] + 1 * x[i, 2] + 1 * x[i, 3] + 0 *$

$x[i, 4] + \dots + 0 * x[i, q] = \alpha + \mathbf{x}_i^T \boldsymbol{\beta}$. The command `1:n` creates the vector $(1, 2, \dots, n)^T$, and the matrix multiplication operator is `% * %`. The function `glm` fits the generalized linear model for Poisson regression. Typing “out” and “summary(out)” give output for the Poisson regression. The first term in the plot command is always the horizontal axis while the 2nd is the vertical axis.

Now lets fit a generalized additive model $Y|AP \sim \text{Poisson}(\exp(AP))$

```
library(mgcv)
x1 <- x[,1]
x2 <- x[,2]
x3 <- x[,3]
out <- gam(y ~ s(x1) + s(x2) + s(x3),family = poisson)
out
summary(out)
plot(out)
#make a response plot
EAP<-predict.gam(out)
plot(EAP,Y)
lines(lowess(EAP,Y))
#make an EE plot
plot(EAP,ESP)
abline(0,1)
```

To put a graph in “Word”, hold down the *Ctrl* and *c* buttons simultaneously. Then in the *Word* Edit menu, select “paste.”

Type the following commands to make logistic regression binomial($m, \rho(SP)$) data where $\rho(SP) = \exp(SP)/(1 + \exp(SP))$.

```
mv <- 1 + 0 * y
pv <- exp(SP)/(1 + exp(SP))
for(i in 1:n)
y[i] <- rbinom(1, size = 1, prob = pv[i])
out <- glm(y ~ x[, 1] + x[, 2] + x[, 3] + x[, 4] + x[, 5],
family = binomial, weights = mv)
out
summary(out)
```

From regpack, copy and paste `lressp` into *R*. Then type “`lressp(x,y)`”.

Now lets fit a generalized additive model $Y|AP \sim \text{binomial}(1, \rho(AP))$

```
x1 <- x[,1]
x2 <- x[,2]
x3 <- x[,3]
#library(mgcv)
out <- gam(y ~ s(x1) + s(x2) + s(x3),family = binomial, weights = mv)
out
```

```
summary(out)
plot(out)
```

To enter data, open a data set in *Word* or *Notepad*. You need to know the number of rows and the number of columns and I am assuming that each case is entered in a row. For example, from the ARC data directory, open *cbrain.lsp* in *Notepad*. It has 267 rows and 13 columns. In *R*, write

```
cbrn <- matrix(scan(),nrow=267,ncol=13,byrow=T)
```

Hit “Enter” and then copy and paste the *cbrain* data (after “begin data”) into *R*. Hit *Enter* twice or until the cursor appears. The command *dim(cbrn)* will show if you have it correctly.

Enter the commands:

```
y <- cbrn[,12]
out <- glm(y~cbrn[,3]+cbrn[,8],family=binomial)
summary(out)
```

This will open an editor such as *Notepad* and allow you to make changes.

To save data or a function, when you exit, say *yes* to “save worksheet.” When you reenter *R*, type *ls()*. This will show you what is saved. You should hardly ever save anything. To remove unwanted items from the worksheet, eg *x*, type *rm(x)*.