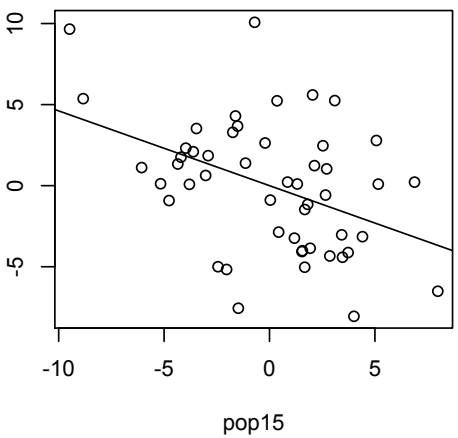
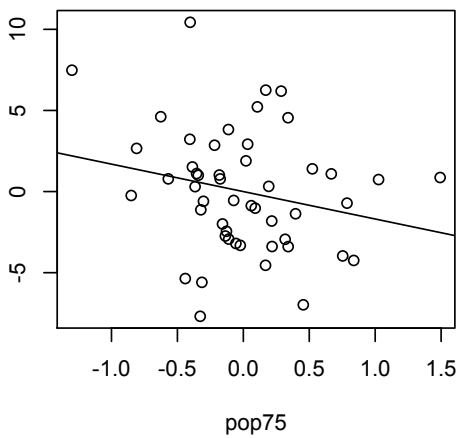
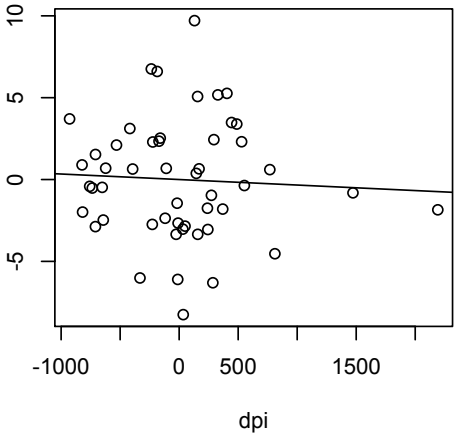
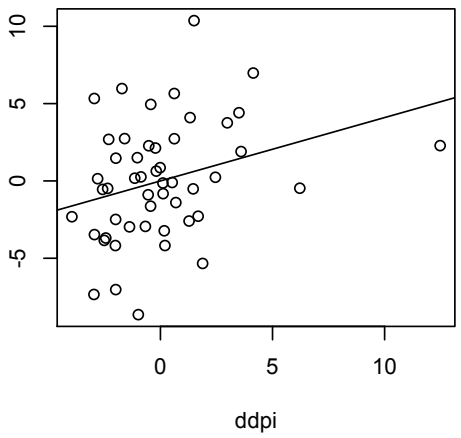
```
> attach(savings)
> n=50; p=5;
> g = lm(sr ~., data=savings);

> library(lmtest)
> bptest(g)

     studentized Breusch-Pagan test

data:  g
BP = 4.9852, df = 4, p-value = 0.2888

> tmp.fit = lm(g$res^2 ~ pop15 + pop75 + dpi + ddpi)
> summary(tmp.fit)

Residual standard error: 18.7 on 45 degrees of freedom
Multiple R-squared:  0.0997,  Adjusted R-squared:  0.01968

> summary(tmp.fit)$r.sq*50
[1] 4.985161
```

```
> g = lm(sr ~., data=savings);
> summary(g)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
pop15       -0.4611931  0.1446422  -3.189 0.002603 **
pop75       -1.6914977  1.0835989  -1.561 0.125530
dpi         -0.0003369  0.0009311  -0.362 0.719173
ddpi         0.4096949  0.1961971   2.088 0.042471 *

Residual standard error: 3.803 on 45 degrees of freedom
Multiple R-squared: 0.3385,   Adjusted R-squared: 0.2797
F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904

> r.y = update(g, ~. - pop15)$res;
> r.pop15 = lm(pop15 ~., data=savings[,-1])$res;
> tmp=lm(r.y ~ r.pop15); summary(tmp)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.908e-17  5.207e-01   0.000  1.00000
r.pop15     -4.612e-01  1.400e-01  -3.293  0.00187 **

> plot(r.pop15, r.y, xlab="pop15", ylab=""); abline(tmp)
```


10
5
0
-5
-10
-5
0
5
pop15

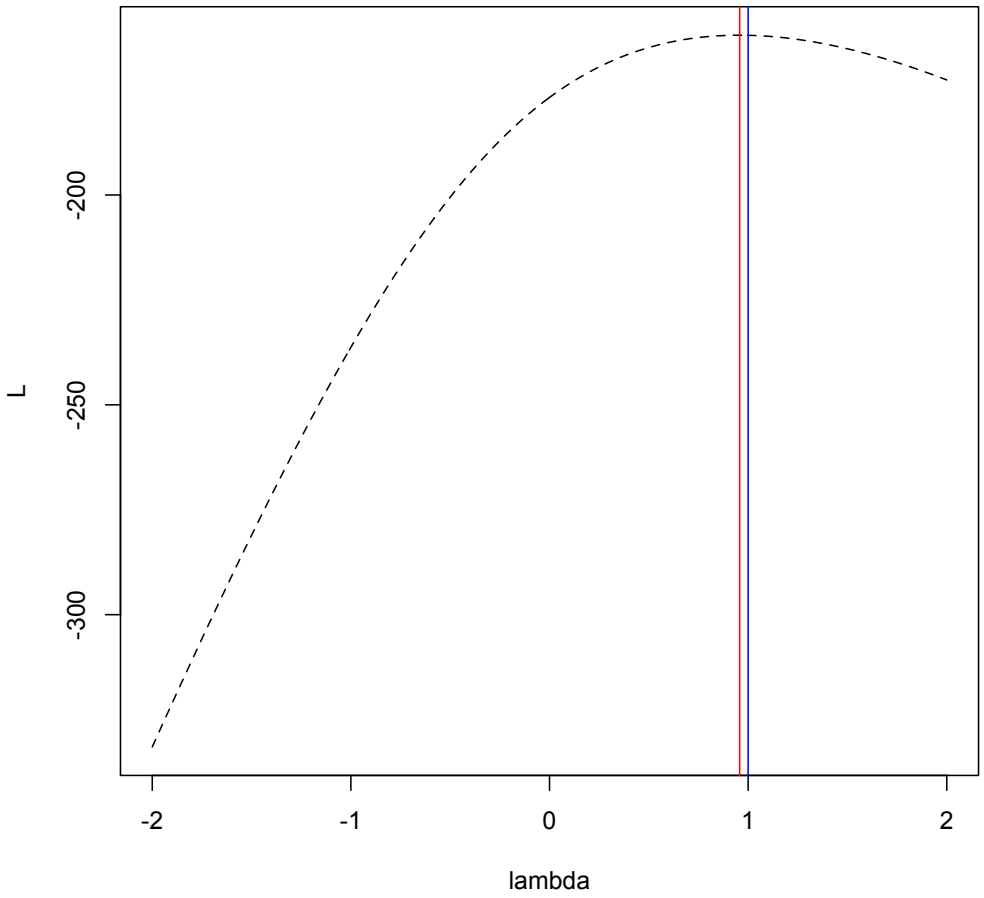

10
5
0
-5
-1.0
-0.5
0.0
0.5
1.0
1.5
pop75


10
5
0
-5
-1000
0
500
1500
dpi

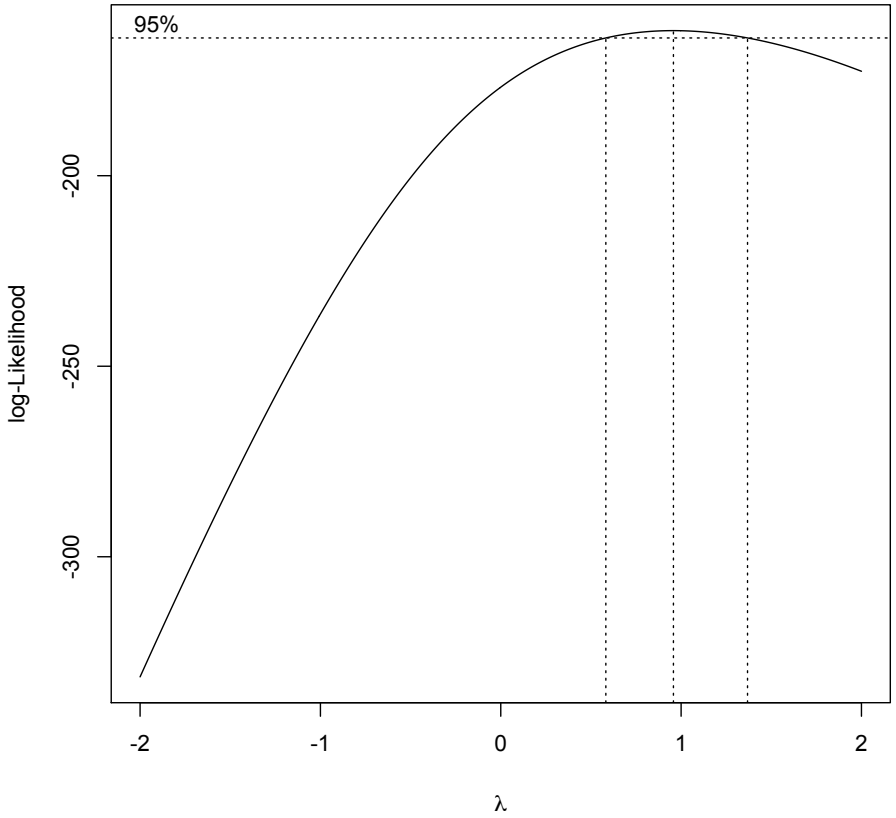

10
5
0
-5
0
5
10
ddpi

```
> g = lm(sr ~  pop15+pop75+dpi+ddpi, savings)

# We will do a grid search for the maximum of lambda

> lambda = seq(-2, 2, length=400); # it doesn't contain zero.
> L = NULL;
> n =nrow(savings)

# Now loop over the 400 values of lambda and
# record the log-likelihood function

for(i in 1:400){
    y=(sr^lambda[i]-1)/lambda[i];
    rss = sum(lm(y~pop15+pop75+dpi+ddpi)$res^2);
    L = c(L, -(n/2)*log(rss) + (lambda[i]-1)*sum(log(sr)));
}

# Now select lambda.hat that maximizes L
> lambda.hat = lambda[L==max(L)]
> lambda.hat

[1] 0.9573935
```

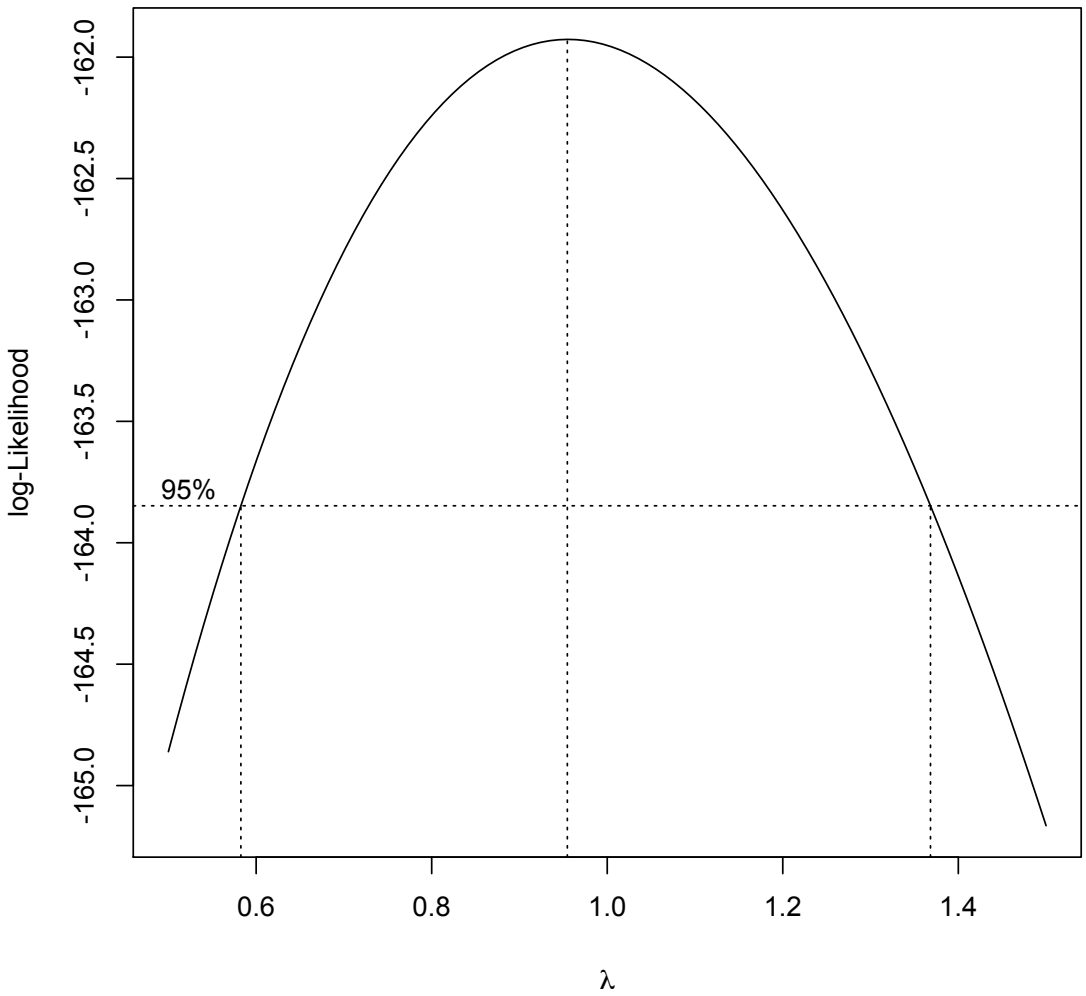
```
> plot(lambda, L, type="n"); lines(lambda, L, lty=2);
> abline(v=lambda.hat, col="red"); abline(v=1, col="blue")
```



```
### Use the build-in function in the "MASS" package
> library(MASS)
> sr.trans=boxcox(g, lambda=seq(-2, 2, length=400))
```



```
> sr.trans
$x
  [1] -2.000000000 -1.989974937 -1.979949875 -1.969924812 -1.959899749
  [6] -1.949874687 -1.939849624 -1.929824561 -1.919799499 -1.909774436

......
$y
  [1] -331.5194 -330.4686 -329.4191 -328.3711 -327.3244 -326.2792 -325.2353
  [8] -324.1930 -323.1520 -322.1125 -321.0745 -320.0380 -319.0030 -317.9695

......

> sr.trans$x[sr.trans$y == max(sr.trans$y)]    # lambda.hat
[1] 0.9573935

> tmp=sr.trans$x[sr.trans$y > max(sr.trans$y) - qchisq(0.95, 1)/2];
> range(tmp)    # 95% CI.
[1] 0.5864662 1.3583960

> boxcox(g,plotit=T)  # plotit=T is the default setting
> boxcox(g,plotit=T,lambda=seq(0.5,1.5,by=0.1))  # zoom-in
```


log-Likelihood
-162.0
-162.5
-163.0
-163.5
-164.0
-164.5
-165.0
95%
0.6
0.8
1.0
1.2
1.4
λ