Regression via Ordinary Least Squares (OLS)

Gustavo de los Campos

Consider a linear model of the form

,

where indexes subjects in the sample.

In matrix form, we can write the above equation as follows

,

where and .

Stacking the n-data-equations into a system we get

where is a "response" vector, is a incidence matrix for the vector of effects .

##### Example

The [Gout data set](https://github.com/gdlc/STAT_COMP/blob/master/goutData.txt) contains data on serum urate, gout, sex, ethnicity, and age.

DATA=read.table('https://raw.githubusercontent.com/gdlc/STAT\_COMP/master/goutData.txt',header=TRUE,sep='')   
 head(DATA)

## sex race age su gout  
## 1 M W 67 8.3 N  
## 2 F W 72 8.6 N  
## 3 F W 70 7.3 N  
## 4 F W 63 6.2 N  
## 5 F W 55 4.3 N  
## 6 M W 63 7.0 N

table(DATA$sex)

##   
## F M   
## 225 175

table(DATA$race)

##   
## B W   
## 92 308

#### The incidence matrix of effects

Consider a model for serum urate (su) as a function of sex, race and age (su~sex+race+age). The variables sex and race, are categorical, we introduce these variables in the model using dummy variables (as many as the number of levels of the factor minus one). The model.matrix() function in R creates incidence matrices for effects from a formula. This is illustrated in the following example.

X=model.matrix(~sex+race+age,data=DATA)  
 y=DATA$su  
   
 head(DATA)

## sex race age su gout  
## 1 M W 67 8.3 N  
## 2 F W 72 8.6 N  
## 3 F W 70 7.3 N  
## 4 F W 63 6.2 N  
## 5 F W 55 4.3 N  
## 6 M W 63 7.0 N

head(X)

## (Intercept) sexM raceW age  
## 1 1 1 1 67  
## 2 1 0 1 72  
## 3 1 0 1 70  
## 4 1 0 1 63  
## 5 1 0 1 55  
## 6 1 1 1 63

head(y)

## [1] 8.3 8.6 7.3 6.2 4.3 7.0

dim(DATA)

## [1] 400 5

dim(X)

## [1] 400 4

length(y)

## [1] 400

## Ordinary least squares

Ordinary Least Squares (OLS) estimates are obtained by minimizing the Residual Sum of Squares (RSS),

.

Differentiating the with respect to leads to

,

setting this equal to zero, leads to the following first-order conditions (FOCs, aka "normal equations"):

Thus, when is a full-column-rank matrix, that is if exist,

##### Example 1: using lm() to obtain OLS estimates

fm=lm(su~sex+race+age,data=DATA)  
  
 coef(fm)

## (Intercept) sexM raceW age   
## 4.31975213 1.52852797 -0.78211876 0.02673734

**Note**: model.matrix() chooses the first level of each factor as the reference group, if you wish to choose a different reference, you can change the order of the levels. This is illustrated in the following example:

x=factor(c('a','a','b','c','c'))  
 levels(x)

## [1] "a" "b" "c"

model.matrix(~x)

## (Intercept) xb xc  
## 1 1 0 0  
## 2 1 0 0  
## 3 1 1 0  
## 4 1 0 1  
## 5 1 0 1  
## attr(,"assign")  
## [1] 0 1 1  
## attr(,"contrasts")  
## attr(,"contrasts")$x  
## [1] "contr.treatment"

# now let's use c as refernece  
 levels(x)=c('c','b','a')  
   
 model.matrix(~x)

## (Intercept) xb xa  
## 1 1 0 0  
## 2 1 0 0  
## 3 1 1 0  
## 4 1 0 1  
## 5 1 0 1  
## attr(,"assign")  
## [1] 0 1 1  
## attr(,"contrasts")  
## attr(,"contrasts")$x  
## [1] "contr.treatment"

### Interpretation of regression coefficients

If a predictor is quantiative (e.g., age, weight), then the corresponding regression coefficient is interpreted as an slope, that is the expected rate of change in y, per unit change in x.

For categorcial predictors, when we use dummy coding (i.e., one dummy variable per level, without including one for the reference level) the corresponding coefficients are interpreted as mean differences. For example, if is a dummy variable for male ( for male, 0 for female) and we have a linear model of the form: , then is interpreted as the "male minus female"" difference in the expected value of , holding age constant.

## Inference

The previous code shows how to obtain a point-estimate (an OLS estimate) from a sample. Our goal is to make inference about the population parameters (that is the regression coefficients in the population from which the sample was drawn). Frequentist inference studies the distribution of estimates over conceptual repeated sampling.

**Sampling distribution**: We can think of estimators (e.g., least-squares) as a function that maps from data (e.g., and ) onto a point-estimate (e.g., $\mathbf{\hat{\beta}$). For every possible sample from the population, the function returns a point estimate. Clearly, the estimator is random because it is a function of the data which we assume is randomly sampled from the population. The sampling distribution of an estimator describes how the estimator is expected to vary over conceptual repeated sampling. Important features of this distribution include the expected value and the variance of the estimator.

**The expected value of the OLS estimator** is

Therefore, if either or , then

,

thus, implying that OLS estimates are unbiased.

**The sampling (co)variance matrix of the OLS estimator** is:

Assuming that the error terms are independent, and have homogeneous variance, , then

Clearly, to obtain an estimate of the (co)variance matrix of the estimator we need an estimate of the error variance

**An unbiased estimate of the error variance**: The expected value of the is

, in the full-rank case ;

therefore, with the notation used here, a method-of moment, unbiased estimator of the error variance is:

.

**Large-sample distribution**

Clearly, the estimator, , is a weighted sum of the data ().

According to the [Central Limit Theorem](https://en.wikipedia.org/wiki/Central_limit_theorem) the asymptotic distribution of the estimator is [Multivariate Normal](https://en.wikipedia.org/wiki/Multivariate_normal_distribution), the mean is the true parameter () because the estimator is unbiased (see previous results), and the (co)variance matrix is

;

therefore,

~MVN(,)

**Note**: The CLT does not require any assumptions about the distribution of the error terms (not even independence or homskedasticity are required!). However, if the error terms are IID Normal, then the above result also holds in small samples.

**Standard Errors**: The SE of each of the coefficients is simply the square root of the corresponding diagonal value of the sampling (co)variance matrix, that is

**z-statistic**: According to the CLT,

~;

therefore,

~.

**Hypothesis testing**: A p-value for the (two-sided) test Vs can be obtained from the standard normal distribution

pValue=pnorm(q=abs(z),lower.tail=FALSE)\*2

Now, to compute the SE we replace with an estimate (, see above); therefore, instead of using the standard normal distribution, we should use the t-distribution with degrees of freedom equal to the residual degree of freedom (n-p-1), thus

pValue=pt(q=abs(z),df=nrow(X)-ncol(X),lower.tail=FALSE)\*2

For large sample size (e.g., n-p-1>30) the t converges to the standard normal distribution and the difference in the p-value computed from one or the other would be small.

##### Example 2: using lm() to obtain OLS estimates, SE, and p-values

**Fitting the model and examining estimates, SEs, and p-values**

fm=lm(su~sex+race+age,data=DATA)  
 class(fm)

## [1] "lm"

summary(fm)

##   
## Call:  
## lm(formula = su ~ sex + race + age, data = DATA)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.4843 -0.9717 -0.1829 0.8276 5.4296   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.31975 0.81533 5.298 1.95e-07 \*\*\*  
## sexM 1.52853 0.14306 10.684 < 2e-16 \*\*\*  
## raceW -0.78212 0.16932 -4.619 5.22e-06 \*\*\*  
## age 0.02674 0.01299 2.058 0.0402 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.413 on 396 degrees of freedom  
## Multiple R-squared: 0.2504, Adjusted R-squared: 0.2447   
## F-statistic: 44.09 on 3 and 396 DF, p-value: < 2.2e-16

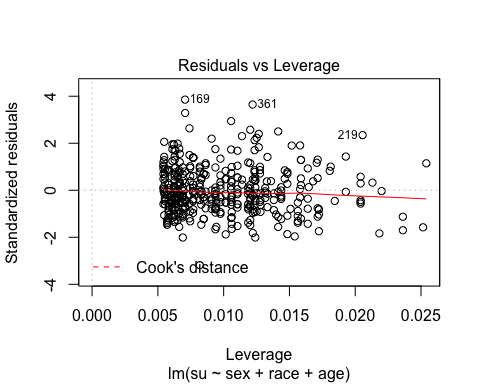
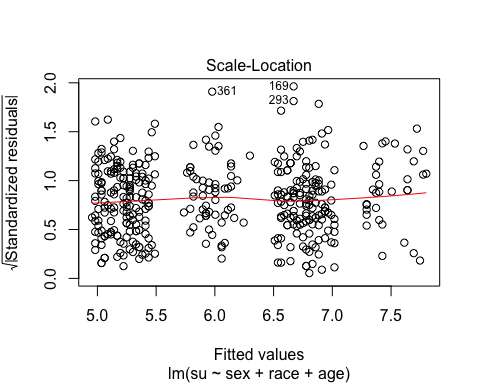
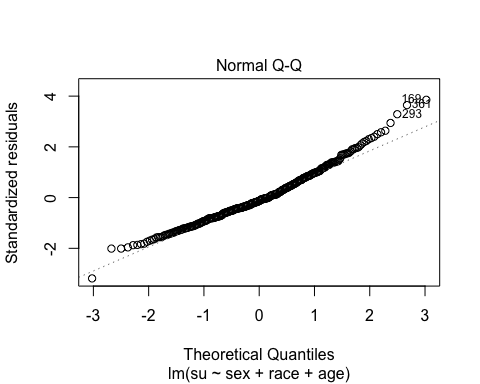
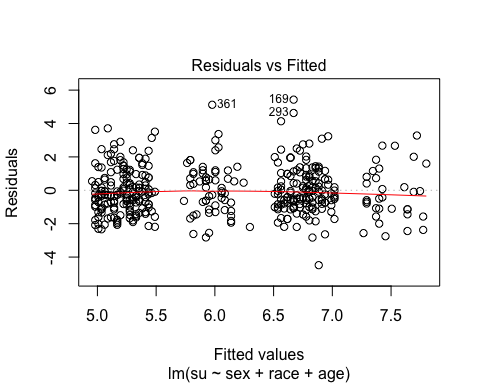
**Residuals and predictions**

# Predictions and residuals for the training data  
 eHat=residuals(fm) # try help(residuals) for options about different type of residuals, more on this below...  
 yHat=predict(fm)  
   
 # You can also derive predicitons for new data  
 tmp=data.frame(age=c(50,55),sex=c('F','F'),race=c('W','B'))  
 predict(fm,newdata=tmp)

## 1 2   
## 4.874501 5.790306

**Diagnostic plots**

plot(fm)



**Retrieving the variance co-variance matrix of estimates**

vcov(fm)

## (Intercept) sexM raceW age  
## (Intercept) 0.664764024 -0.0038349056 -0.0081685585 -0.0103951575  
## sexM -0.003834906 0.0204667219 -0.0020227609 -0.0000568041  
## raceW -0.008168559 -0.0020227609 0.0286690772 -0.0002076739  
## age -0.010395158 -0.0000568041 -0.0002076739 0.0001687320

## Correlation of estimates  
 cov2cor(vcov(fm))

## (Intercept) sexM raceW age  
## (Intercept) 1.00000000 -0.03287735 -0.05917044 -0.98151874  
## sexM -0.03287735 1.00000000 -0.08350521 -0.03056728  
## raceW -0.05917044 -0.08350521 1.00000000 -0.09442268  
## age -0.98151874 -0.03056728 -0.09442268 1.00000000

#### The Hat matrix

Predictions in a linear model take the form

,

replacing with the OLS estimate, we get

where,

is the 'Hat matrix'.

This Hat matrix is symmetric (i.e., ), positive semi-definite (i.e., ) and idempotent (implying that ).

Using , model residuals residuals can be represented as follows:

.

#### Studentized residuals

The (co)varinance matrix of model residuals is ,

assuming independent and homoscedasticity residuals, we get,

;

therefore, the variance of the ith predicted residual is

.

Studentized residuals are standarized to unit variance; thus they are defined as:

.

eStd=rstudent(fm)  
 head(eStd)