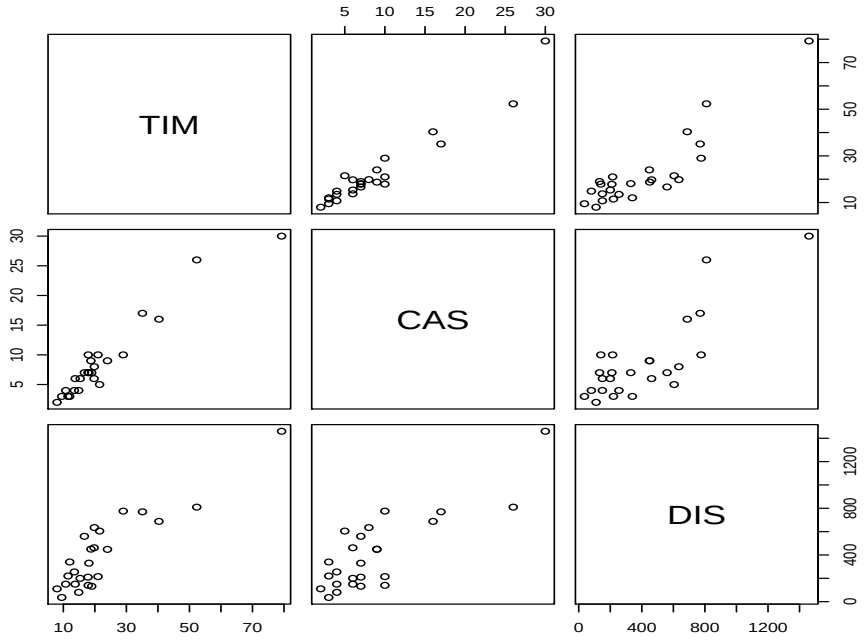
```
> #read in the data from a csv file
> dat1=read.table("Data Sets/Chapter 3/Examples/ex-3-1.csv", header=T, sep=",")
> str(dat1)  #checking data format

'data.frame':        25 obs. of  4 variables:
 $ Observation        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Delivery.Time..y   : num  16.7 11.5 12.0 14.9 13.8 ...
 $ Number.of.Cases..x1: int  7 3 3 4 6 7 2 7 30 5 ...
 $ Distance..x2..ft.  : int  560 220 340 80 150 330 110 210 1460 605 ...

> TIM=dat1[,2] #assign the second column in dat1 to be variable TIM
> CAS=dat1[,3] #assign the third column in dat1 to be variable CAS
> DIS=dat1[,4] #assign the forth column to be variable DIS
> EX3=data.frame(TIM, CAS, DIS)
> head(EX3)  #printout the first few row in dat1

    TIM CAS DIS
1 16.68   7 560
2 11.50   3 220
3 12.03   3 340
4 14.88   4  80
5 13.75   6 150
6 18.11   7 330

> cor(EX3)

          TIM       CAS      DIS
TIM 1.0000000 0.9646146 0.891670
CAS 0.9646146 1.0000000 0.824215
DIS 0.8916701 0.8242150 1.000000

> pairs(EX3)
```



```
> library(xtable)
> g=lm(TIM~CAS+DIS, data=EX3)
> xtable(anova(g))
```

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| CAS | 1 | 5382.41 | 5382.41 | 506.62 | 0.0000 |
| DIS | 1 | 168.40 | 168.40 | 15.85 | 0.0006 |
| Residuals | 22 | 233.73 | 10.62 | | |

```
> xtable(summary(g))
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 2.3412 | 1.0967 | 2.13 | 0.0442 |
| CAS | 1.6159 | 0.1707 | 9.46 | 0.0000 |
| DIS | 0.0144 | 0.0036 | 3.98 | 0.0006 |

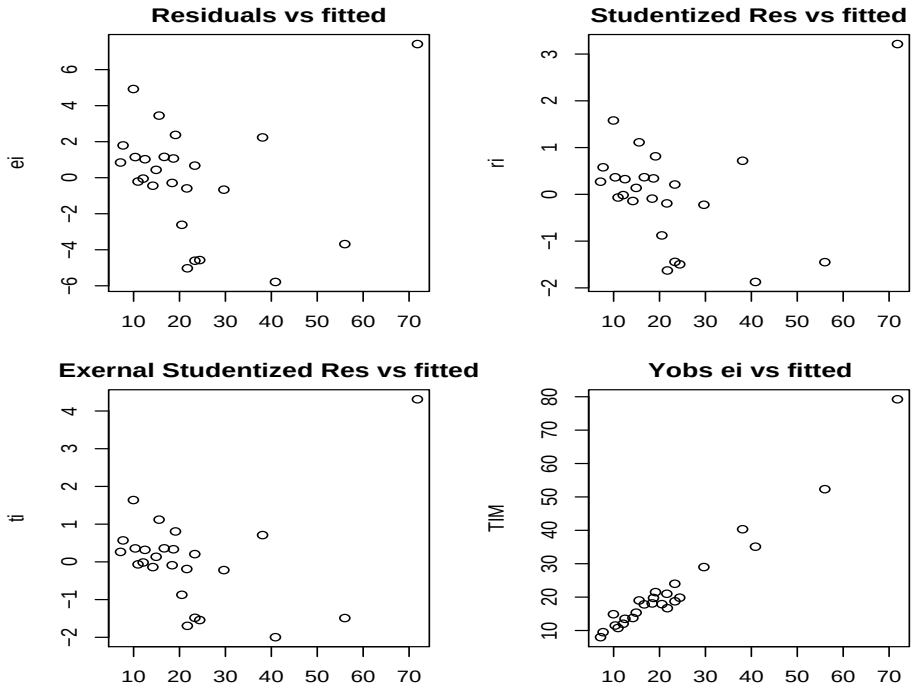
```
> yhat=fitted(g)  #fitted values
> ei=resid(g)     #residuals
> ri=rstandard(g) #studentized residuals: ei/(s*sqrt(1-hii))
> ti=rstudent(g)  #external studentized residuals: ei/(s(i)*sqrt(1-hii))
> hii=hatvalues(g)    #hii: diagnoal of H
> out=cbind(TIM, yhat, ei, ri, ti, hii)
> xtable(out)  #two decimal places for printout
```

1. (Scaled) Residuals vs fitted values

   outliner, constant variance, any patterns. transformation of y or xi weighted least squared estimator

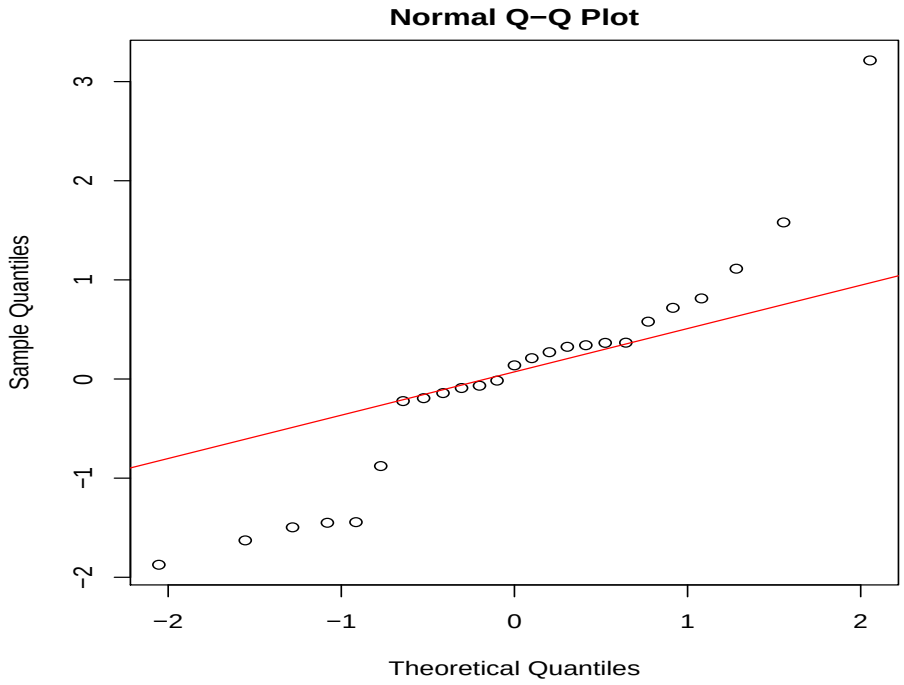
```
> par(mfrow=c(2,2),mar=c(3,4,2,2))
> plot(yhat, ei, main="Residuals vs fitted")
> plot(yhat, ri, main="Studentized Res vs fitted")
> plot(yhat, ti, main="Exernal Studentized Res vs fitted")
> plot(yhat, TIM, main="Yobs ei vs fitted")
```

| | TIM | yhat | ei | ri | ti | hii |
|---|---|---|---|---|---|---|
| 1 | 16.68 | 21.71 | -5.03 | -1.63 | -1.70 | 0.10 |
| 2 | 11.50 | 10.35 | 1.15 | 0.36 | 0.36 | 0.07 |
| 3 | 12.03 | 12.08 | -0.05 | -0.02 | -0.02 | 0.10 |
| 4 | 14.88 | 9.96 | 4.92 | 1.58 | 1.64 | 0.09 |
| 5 | 13.75 | 14.19 | -0.44 | -0.14 | -0.14 | 0.08 |
| 6 | 18.11 | 18.40 | -0.29 | -0.09 | -0.09 | 0.04 |
| 7 | 8.00 | 7.16 | 0.84 | 0.27 | 0.26 | 0.08 |
| 8 | 17.83 | 16.67 | 1.16 | 0.37 | 0.36 | 0.06 |
| 9 | 79.24 | 71.82 | 7.42 | 3.21 | 4.31 | 0.50 |
| 10 | 21.50 | 19.12 | 2.38 | 0.81 | 0.81 | 0.20 |
| 11 | 40.33 | 38.09 | 2.24 | 0.72 | 0.71 | 0.09 |
| 12 | 21.00 | 21.59 | -0.59 | -0.19 | -0.19 | 0.11 |
| 13 | 13.50 | 12.47 | 1.03 | 0.33 | 0.32 | 0.06 |
| 14 | 19.75 | 18.68 | 1.07 | 0.34 | 0.33 | 0.08 |
| 15 | 24.00 | 23.33 | 0.67 | 0.21 | 0.21 | 0.04 |
| 16 | 29.00 | 29.66 | -0.66 | -0.22 | -0.22 | 0.17 |
| 17 | 15.35 | 14.91 | 0.44 | 0.14 | 0.13 | 0.06 |
| 18 | 19.00 | 15.55 | 3.45 | 1.11 | 1.12 | 0.10 |
| 19 | 9.50 | 7.71 | 1.79 | 0.58 | 0.57 | 0.10 |
| 20 | 35.10 | 40.89 | -5.79 | -1.87 | -2.00 | 0.10 |
| 21 | 17.90 | 20.51 | -2.61 | -0.88 | -0.87 | 0.17 |
| 22 | 52.32 | 56.01 | -3.69 | -1.45 | -1.49 | 0.39 |
| 23 | 18.75 | 23.36 | -4.61 | -1.44 | -1.48 | 0.04 |
| 24 | 19.83 | 24.40 | -4.57 | -1.50 | -1.54 | 0.12 |
| 25 | 10.75 | 10.96 | -0.21 | -0.07 | -0.07 | 0.07 |

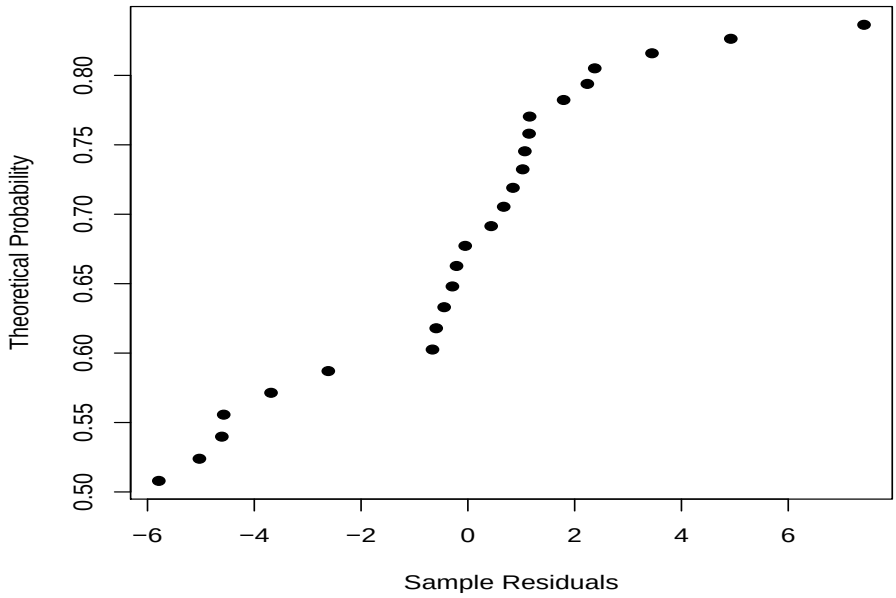



2. Normal Probability Plots: checking if errors follow a normal distribution
   Differences between $\varepsilon_i$ and $e_i$?

```
> par(mfrow=c(1,1),mar=c(5,4,2,2))
> qqnorm(ri);qqline(ri,col=2)
```




```
> my.ppnorm=function(r) {
+   n=length(r)
+   order.res=sort(r)
+   i=1:n
+   pi=pnorm((i-1/2)/n)
+   plot(order.res, pi, xlab = "Sample Residuals", ylab = "Theoretical Probability", pch=16)
+ }
> my.ppnorm(ei)
```




**example**

Some normal probability plots:

```
> par(mfrow=c(1,1))
> curve(dnorm(x), -4,4)
> curve(dcauchy(x), -4,4, add=TRUE, col=2)
> curve(dt(x,2), -4,4, add=TRUE ,col=3)
> curve(dt(x,4), -4,4, add=TRUE ,col=3, lty=3)
> curve(dexp(x),0,4, add=TRUE,col=4)
> legend("topright", legend=c("Normal","Cauchy", "t","Exp"), lty=1, col=c(1:4))
```

```
> par(mfrow=c(3,3), mar=c(2,2,1,1))
> for (i in 1:9) { e=3+rnorm(50); qqnorm(e, main="");qqline(e,col=2) }
```

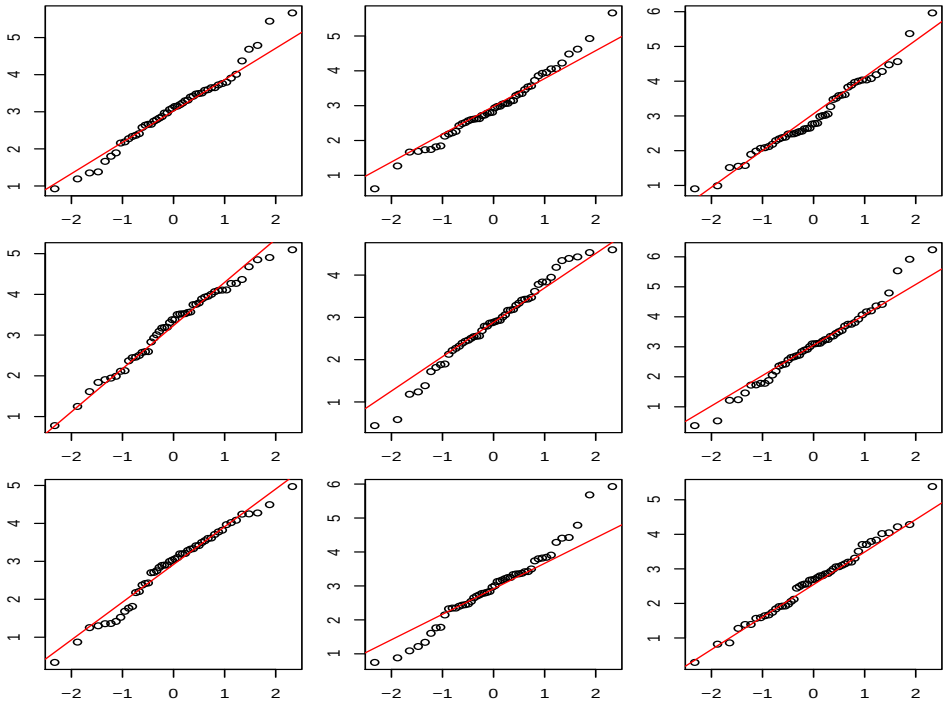
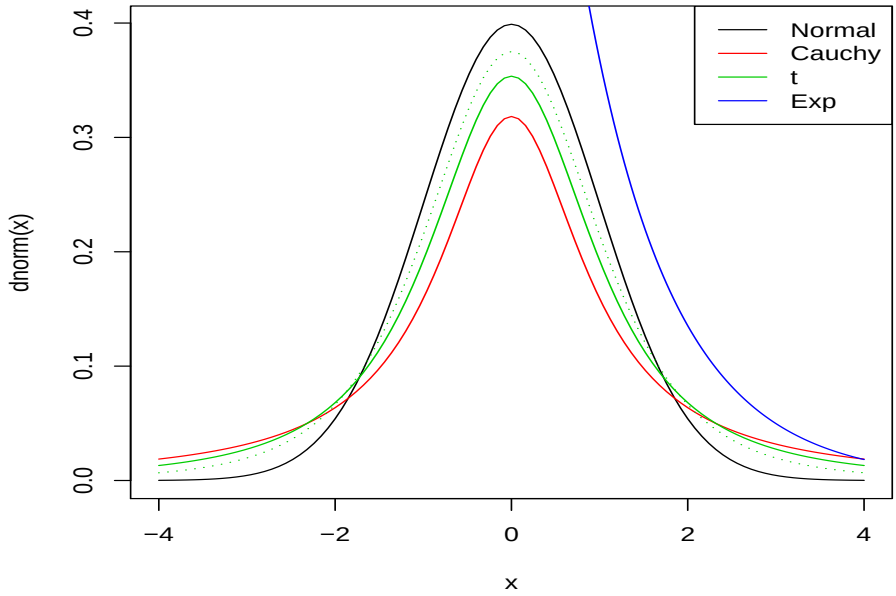


Figure 1: Normal($\mu = 3$,sd=1)




3. Plot of Residuals in Time sequence: autocorrelations

```
> par(mfrow=c(3,3), mar=c(2,2,1,1))
> for (i in 1:9) { e=exp(rnorm(50)); qqnorm(e, main="");qqline(e,col=2) }
```

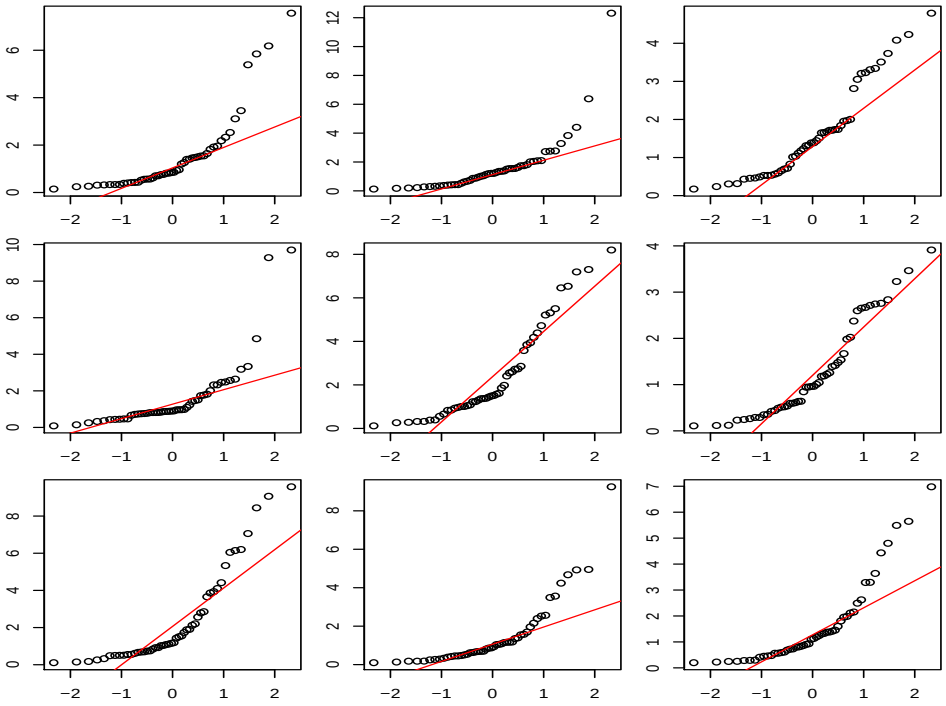


Figure 2: Exponetial: right skew

```
> par(mfrow=c(3,3), mar=c(2,2,1,1))
> for (i in 1:9) { e=3+rcauchy(50); qqnorm(e, main="");qqline(e,col=2) }
```

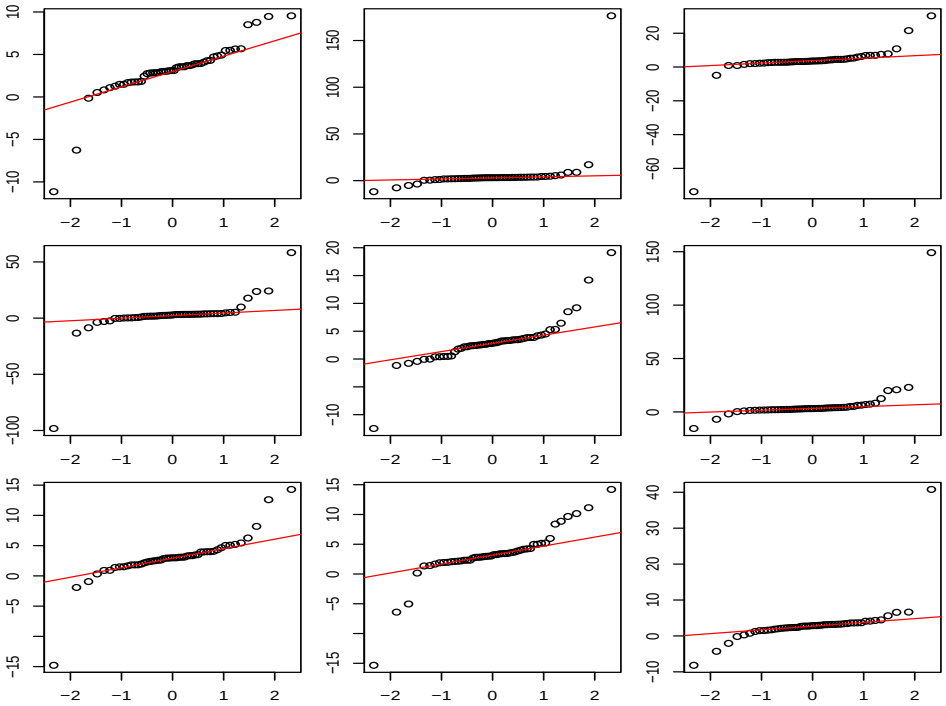


Figure 3: Cauchy: heavy tailed

```
> par(mfrow=c(3,3), mar=c(2,2,1,1))
> for (i in 1:9) { e=runif(50,-1,1); qqnorm(e, main="");qqline(e,col=2) }
```

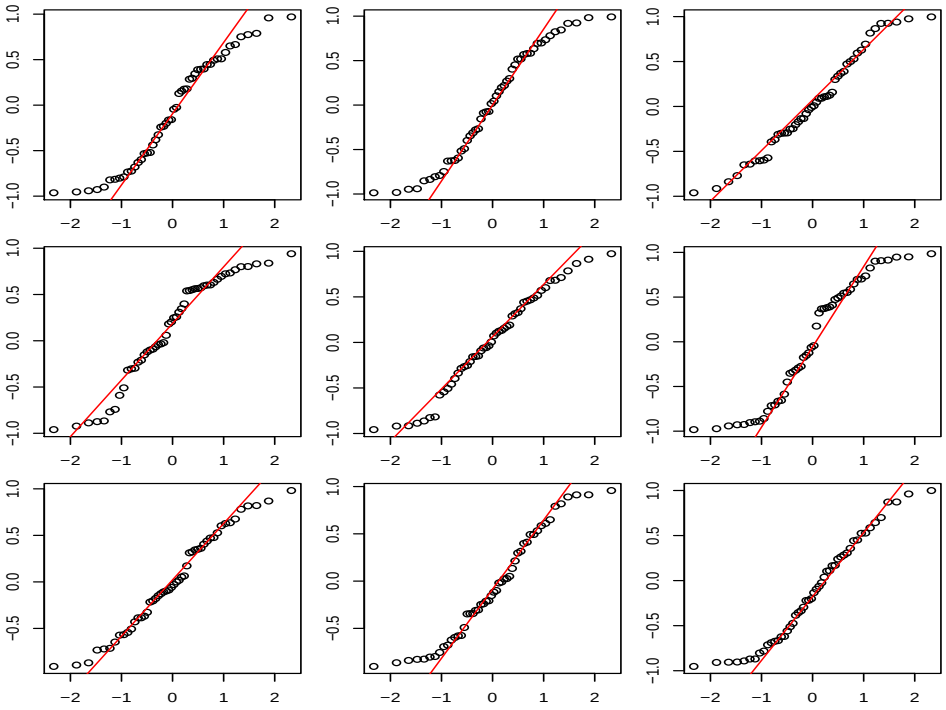


Figure 4: Unifor

4. Partial regression and partial residual plots We can look at plots of $e_i$ against $x_i$. The drawback is that the other regressors are also in the model, and they may not show the correct marginal effect of a regressor.

   (a) partial regression or added-variable plots can help to isolate the effect of $x_i$ on $y$.
   The relationshop between the slope of the partial regresion line and the (partial) regression coefficient of $x_i$ in the multiple regression model.

   ```
   > coef(lm(e~d))

    (Intercept)            d
   9.981487e-18 1.615907e+00

   > coef(g)

   (Intercept)         CAS         DIS
    2.34123115  1.61590721  0.01438483
   ```

   (b) partial residual plots: $e + \hat{\beta}_j x_j$ against $x_j$

   ```
   > cr.plot(g, CAS)
   ```

```
> e=resid(lm(TIM~DIS))
> d=resid(lm(CAS~DIS))
> plot(d, e, xlab="CAS", ylab="TIM Residuals")
> abline(0, coef(g)['CAS'])
```

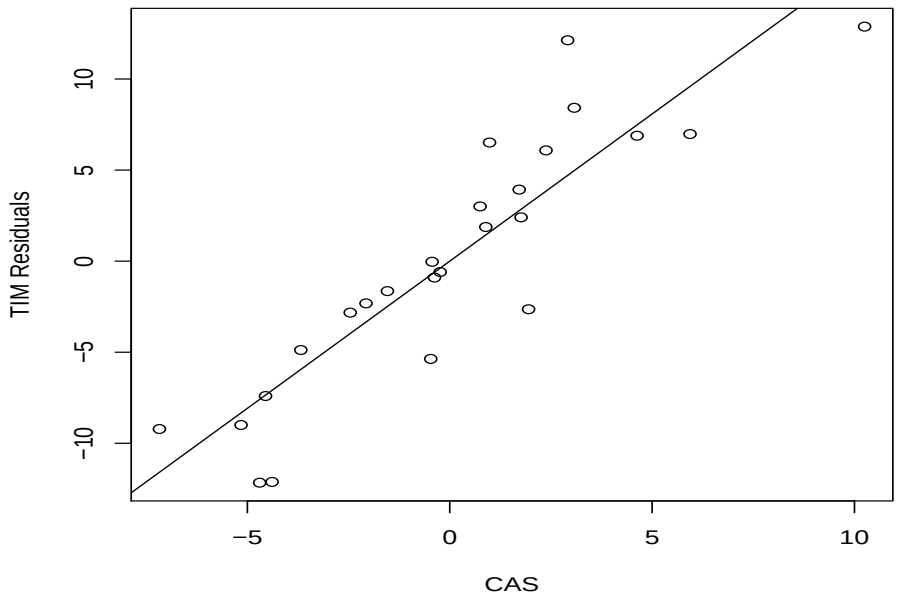
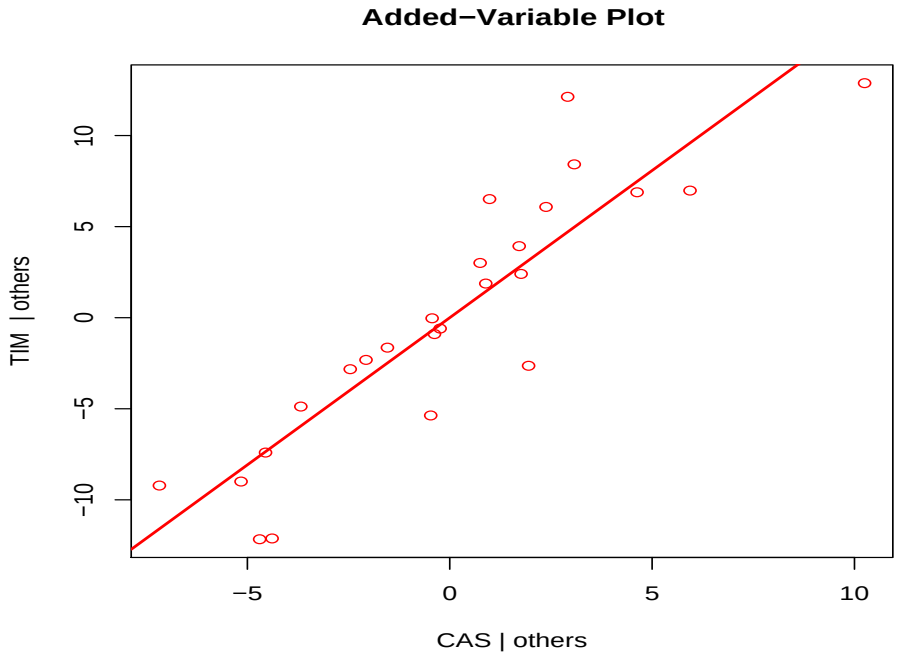



Figure 5: 4.7 Partial regression plot

```
> library(car)    #include car library
> av.plot(g, CAS, identify.points=F)
```



```
> b=coef(g)['CAS']
> plot(CAS, resid(g)+b*CAS, xlab="CAS", ylab="RES+COMPONENT")
```

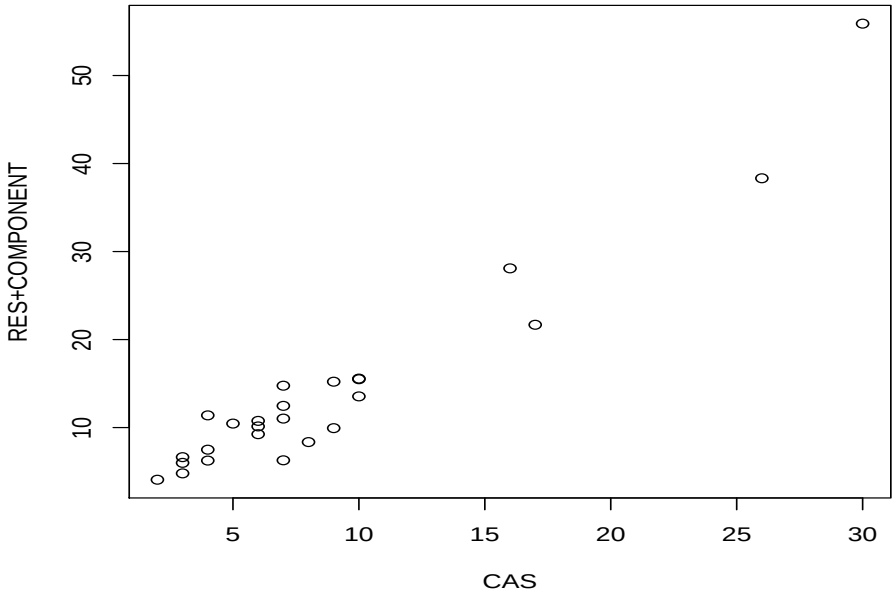
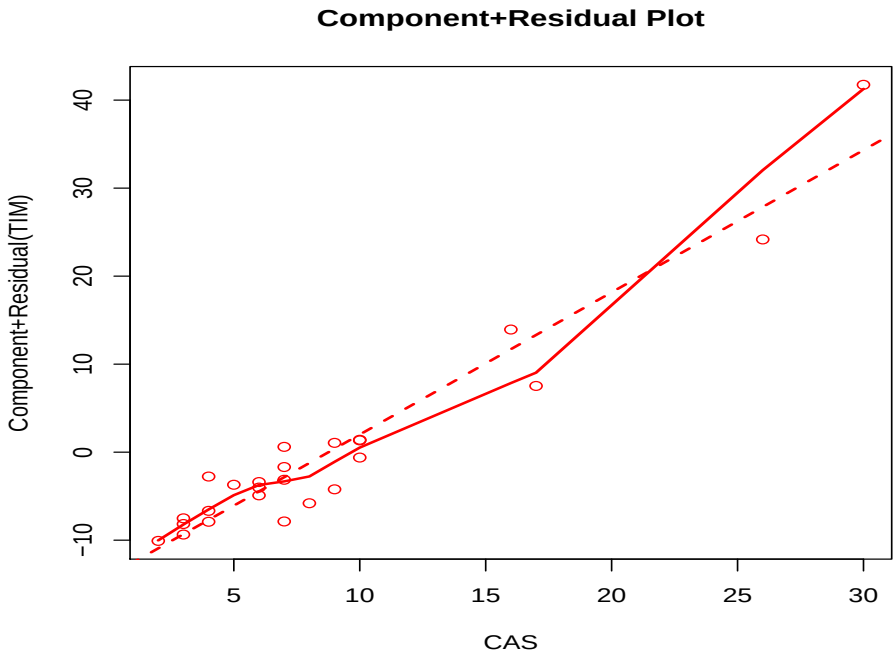



Figure 6: partial residual plot




**example:** Assume the true mdoel is $y = 2+??x1^2+??x2+??x3+\varepsilon$ where $\varepsilon \sim N(0, sd = 0.1)$ runif(20, -1, 1) : draw 20 random sample for a uniform distribution within (-1, 1)

rnorm(20, sd=0.1) : draw 20 random sample for a normal distribution with mean 0 and standard deviation 0.1.

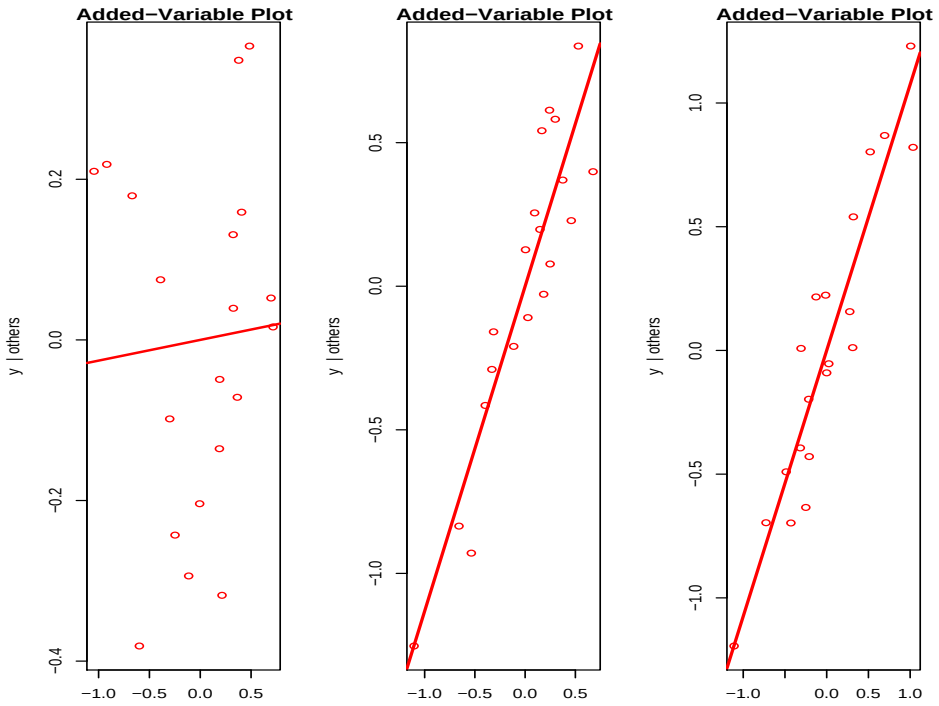
```
> x1=runif(20,-1,1)
> x2=runif(20,-1,1)
> x3=runif(20,-1,1)
> y=2+x2+x3+I(x1^2)+rnorm(20, sd=0.1)
> cor(cbind(y, x1, x2, x3))

           y         x1         x2          x3
y  1.0000000 0.32486084  0.5331992  0.61703192
x1 0.3248608 1.00000000  0.3617582  0.04026376
x2 0.5331992 0.36175820  1.0000000 -0.27535328
x3 0.6170319 0.04026376 -0.2753533  1.00000000
```

Fit a linear model $lm(y \sim x1 + x2 + x3)$.

```
> g=lm(y~x1+x2+x3)
> summary(g)

Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
     Min       1Q   Median       3Q      Max
-0.36572 -0.15625  0.01433  0.16058  0.35339

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.31108    0.06083  37.993  < 2e-16
x1           0.02580    0.10524   0.245     0.81
x2           1.13142    0.12292   9.205 8.59e-08
x3           1.07440    0.09957  10.790 9.44e-09

Residual standard error: 0.2368 on 16 degrees of freedom
Multiple R-squared: 0.916,        Adjusted R-squared: 0.9002
F-statistic: 58.12 on 3 and 16 DF,  p-value: 7.997e-09

> anova(g)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x1         1 1.1270  1.1270  20.090 0.0003772
x2         1 2.1230  2.1230  37.846 1.392e-05
x3         1 6.5313  6.5313 116.430 9.438e-09
Residuals 16 0.8975  0.0561
```

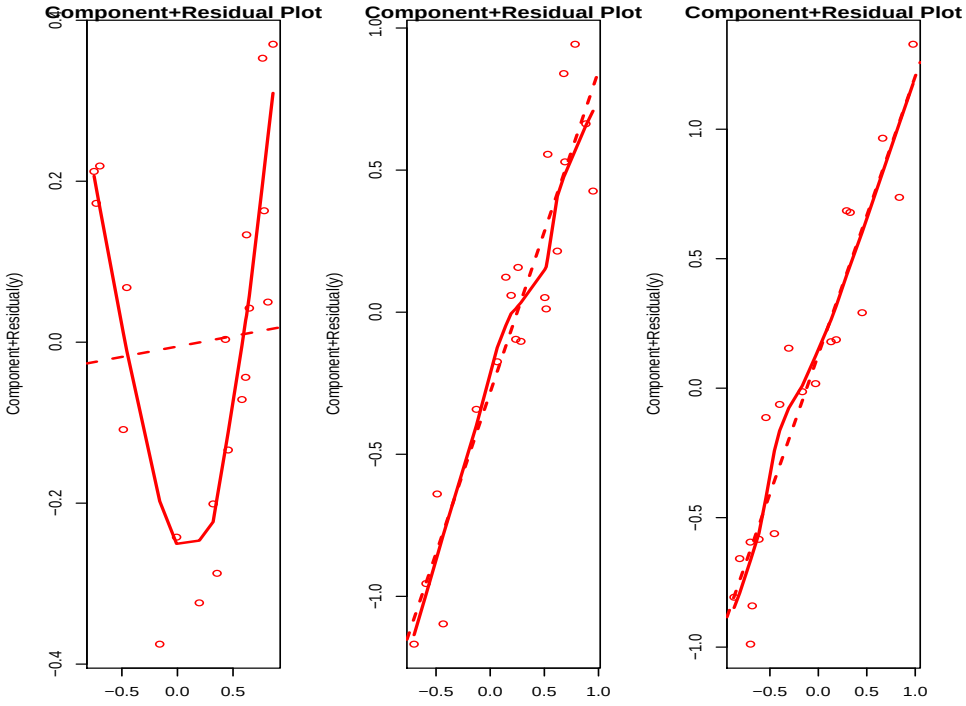
added-variable plots (partial regression plots) are better for outliner detection

```
> par(mfrow=c(1,3),mar=c(3,4,1,2))
> av.plot(g, x1, identify.points=FALSE)
> av.plot(g, x2, identify.points=FALSE)
> av.plot(g, x3, identify.points=FALSE)
```




partial residual plots are better for nonlinearity detection

```
> par(mfrow=c(1,3),mar=c(3,4,1,2))
> cr.plot(g, x1) #partial regression plot
> cr.plot(g, x2)
> cr.plot(g, x3)
```

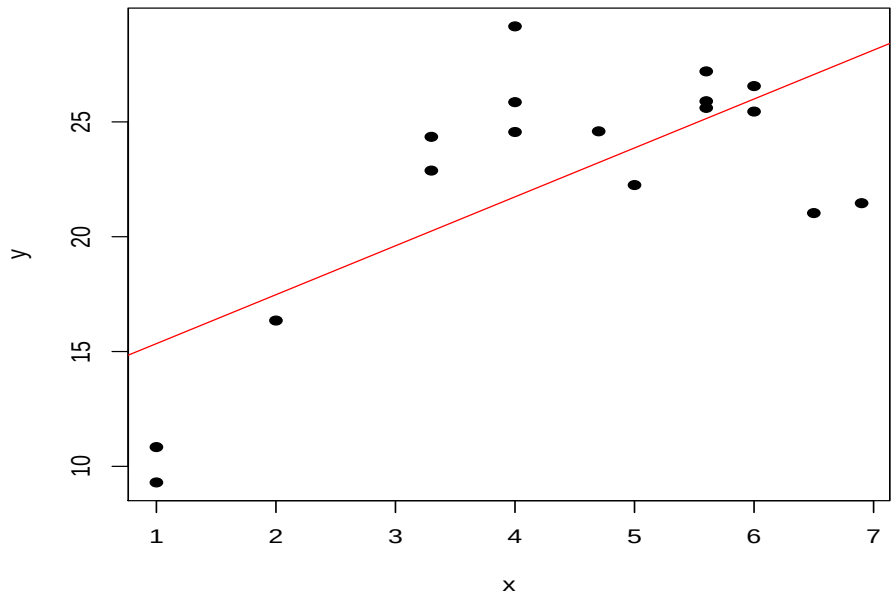



5. PRESS statistic

```
> sum(ei^2)   #Residual Sum of squares: SSres

[1] 233.7317

> press.ei=ei/(1-hii)
> sum(press.ei^2) #PRESS

[1] 459.0393

> n=length(TIM)  #number of obs
> (SSTO=(n-1)*var(TIM))  #SSTO=Syy

[1] 5784.543

> 1-sum(press.ei^2)/SSTO  #prediction R2

[1] 0.9206438
```

**Example 4.8 testing for lack of fit**

```
> par(mfrow=c(1,1))
> x=c(1,1,2, 3.3, 3.3, 4,4,4,4.7,5,5.6,5.6,5.6,6,6,6.5,6.9)
> y=c(10.84,9.3,16.35, 22.88, 24.35, 24.56, 25.86, 29.16,24.59,22.25,25.9, 27.2,
+ 25.61,25.45,26.56, 21.03, 21.46)
> print(rbind(x,y))
   [,1] [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12]
x  1.00  1.0  2.00  3.30  3.30  4.00  4.00  4.00  4.70  5.00   5.6   5.6
y 10.84  9.3 16.35 22.88 24.35 24.56 25.86 29.16 24.59 22.25  25.9  27.2
  [,13] [,14] [,15] [,16] [,17]
x  5.60  6.00  6.00  6.50  6.90
y 25.61 25.45 26.56 21.03 21.46

> plot(x,y, pch=16)
> g=lm(y~x)
> abline(coef(g), col=2)
```



```
> xtable(summary(g)) #test for the significance of regression?
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 13.2139 | 2.6649 | 4.96 | 0.0002 |
| x | 2.1304 | 0.5645 | 3.77 | 0.0018 |

```
> xtable(anova(g))
```

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| x | 1 | 237.48 | 237.48 | 14.24 | 0.0018 |
| Residuals | 15 | 250.13 | 16.68 | | |

Use the function factor to encode a vector $x$ as a factor $X$.

```
> X=factor(x)  #re-assigne x as a factor X
> levels(X)  #level for the factor X
```

[1] "1" "2" "3.3" "4" "4.7" "5" "5.6" "6" "6.5" "6.9"

```
> length(levels(X))
```

[1] 10

```
> f=lm(y~X)
> xtable(anova(f))
```

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| X | 9 | 472.05 | 52.45 | 23.59 | 0.0002 |
| Residuals | 7 | 15.56 | 2.22 | | |

Test for lack of fit

```
> anova(g,f)

Analysis of Variance Table

Model 1: y ~ x
Model 2: y ~ X
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1     15 250.134
2      7  15.563  8   234.571 13.188 0.001389
```