```
> ph = read.table("ph.dat", header=T)
> attach(ph)

> anova(lm(phWert~Beregnung+Kalkung))
Analysis of Variance Table

Response: phWert
          Df   Sum Sq  Mean Sq    F value      Pr(>F)
Beregnung  2    3.175    1.588     8.4727   0.0004194 ***
Kalkung    1  196.282  196.282  1047.4749  < 2.2e-16 ***
Residuals 92   17.239    0.187
```

```
> ph = read.table("ph.dat", header=T)
> attach(ph)

> anova(lm(phWert~Beregnung*Kalkung))
Analysis of Variance Table

Response: phWert
                   Df   Sum Sq  Mean Sq    F value      Pr(>F)
Beregnung           2    3.175    1.588     9.4909   0.0001820 ***
Kalkung             1  196.282  196.282  1173.3557  < 2.2e-16 ***
Beregnung:Kalkung   2    2.184    1.092     6.5281   0.0022521 **
Residuals          90   15.055    0.167
```

```
> goats= read.table("goats.data", header=T)
> attach(goats)

> anova(lm(Gain~Treatment))
Analysis of Variance Table

Response: Gain
          Df Sum Sq Mean Sq F value  Pr(>F)
Treatment  1   16.9 16.9000  4.1299 0.04916 *
Residuals 38  155.5  4.0921


> anova(lm(Gain~Treatment+Weight))
Analysis of Variance Table

Response: Gain
          Df Sum Sq Mean Sq F value    Pr(>F)
Treatment  1 16.900  16.900  6.4559   0.01538 *
Weight     1 58.643  58.643 22.4022 3.208e-05 ***
Residuals 37 96.857   2.618
```

```
> cor.test(body.weight, metabolic.rate)

          Pearson's product-moment correlation

data:  body.weight and metabolic.rate
t = 7.2213, df = 42, p-value = 7.025e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5742343 0.8527119
sample estimates:
      cor
0.7442379

> cor.test(body.weight, metabolic.rate, method="spearman")

         Spearman's rank correlation rho

data:  body.weight and metabolic.rate
S = 3694.151, p-value = 9.701e-09
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.7396652

Warnmeldung:
In cor.test.default(body.weight, metabolic.rate, method = "spearma
  Kann exakte p-Werte bei Bindungen nicht berechnen
```

	abstraktion	koordination	alter
1	9	8	6
2	11	12	8
3	13	14	9
4	13	13	9
5	14	14	10
6	9	8	7
7	10	9	8
8	11	12	9
9	10	8	8
10	8	9	7
11	13	14	10
12	7	7	6
13	9	10	10
14	13	12	10
15	14	12	6

```
> Blutdruck
  rauchen  uebergewicht  schnarchen  blutdruck.normal  blutdruck.hoch
1   nein       nein          nein             55                5
2    ja        nein          nein             15                2
3   nein        ja           nein              7                1
4    ja         ja           nein              2                0
5   nein       nein           ja             152               35
6    ja        nein           ja              72               13
7   nein        ja            ja              36               15
8    ja         ja            ja              15                8
```

```
> glm1 = glm(cbind(blutdruck.hoch, blutdruck.normal)~ rauchen+uebergewicht
    +schnarchen, family = binomial)
> summary(glm1)

Call:
glm(formula = cbind(blutdruck.hoch, blutdruck.normal) ~ rauchen +
    uebergewicht + schnarchen, family = binomial)

Deviance Residuals:
        1          2          3          4          5          6          7
-0.04344    0.54145   -0.25476   -0.80051    0.19759   -0.46602   -0.21262
        8
 0.56231

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -2.37766    0.38018  -6.254    4e-10 ***
rauchenja       -0.06777    0.27812  -0.244   0.8075
uebergewichtja   0.69531    0.28509   2.439   0.0147 *
schnarchenja     0.87194    0.39757   2.193   0.0283 *
___
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 14.1259  on 7  degrees of freedom
Residual deviance:  1.6184  on 4  degrees of freedom
AIC: 34.537

Number of Fisher Scoring iterations: 4
```

```
> glm2 = glm(cbind(blutdruck.hoch,blutdruck.normal)~ uebergewicht+schnarchen,
> summary(glm2)

Call:
glm(formula = cbind(blutdruck.hoch, blutdruck.normal) ~ uebergewicht +
    schnarchen, family = binomial)

Deviance Residuals:
        1         2         3         4         5         6         7
-0.01247   0.47756  -0.24050  -0.82050   0.30794  -0.62742  -0.14449
        8
 0.45770

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      -2.3921     0.3757  -6.366 1.94e-10 ***
uebergewichtja    0.6954     0.2851   2.440   0.0147 *
schnarchenja      0.8655     0.3967   2.182   0.0291 *
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 14.1259  on 7  degrees of freedom
Residual deviance:  1.6781  on 5  degrees of freedom
AIC: 32.597

Number of Fisher Scoring iterations: 4
```

```
> read.csv(url("http://www.ats.ucla.edu/stat/r/dae/binary.csv"))
> attach(admission)
> plot(gre, admit, xlab="",ylab="")
> glm1 = glm(admit~gre, family= binomial)

> summary(glm1)

Call:
glm(formula = admit ~ gre, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1623  -0.9053  -0.7547   1.3486   1.9879

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.901344   0.606038  -4.787 1.69e-06 ***
gre          0.003582   0.000986   3.633  0.00028 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 486.06  on 398  degrees of freedom
AIC: 490.06

Number of Fisher Scoring iterations: 4
```

```
> read.csv(url("http://www.ats.ucla.edu/stat/r/dae/binary.csv"))
> attach(admission)
> plot(gre, admit, xlab="",ylab="")
> glm2 = glm(admit~gre, family= binomial(link="probit"))
> summary(glm2)

Call:
glm(formula = admit ~ gre, family = binomial(link = "probit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1583  -0.9072  -0.7551   1.3483   2.0114

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.7681857  0.3579384  -4.940 7.82e-07 ***
gre          0.0021750  0.0005882   3.698 0.000217 ***
———
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 485.99  on 398  degrees of freedom
AIC: 489.99

Number of Fisher Scoring iterations: 4
```

```
> gewicht = read.table("http://www.jku.at/ifas/content/e108280/
  ⌴⌴e108502/e109088/e109094/Gewicht.txt", header=T)$Gewicht
> mean(gewicht)
 [1] 79.90434

> var(gewicht)/length(gewicht)
 [1] 0.2504675

> btstrmean = numeric(0)
> for(j in 1:5000) {btstrmean = c(btstrmean, mean(sample(gewicht,
   length(gewicht), replace=T)))}

> 4999/5000*var(btstrmean)
 [1] 0.2616644

> btstrmean = numeric(0)
> for(j in 1:20000) {btstrmean = c(btstrmean, mean(sample(gewicht,
   length(gewicht), replace=T)))}
> 19999/20000*var(btstrmean)
 [1] 0.2519285
```

```
> LSAT= law$LSAT
> GPA = law$GPA
> cor(LSAT,GPA)
[1] 0.7763745

> bootstrapcor = numeric(0)
> for(j in 1:5000) {y = sample(1:15, size=15, replace=T);
    bootstrapcor = c(bootstrapcor, cor(LSAT[y],GPA[y]))}
> hist(bootstrapcor, breaks=40, probability=T, xlim=c(-0.3,1), ylim=c(0,3.5),
    xlab="Bootstrap_Samples", ylab="",main="")


> cor(law82$LSAT, law82$GPA)
[1] 0.7599979

> samplecor = numeric(0)
> for(j in 1:20000) {y = sample(1:82, size=15, replace=T);
        samplecor = c(samplecor, cor(law82$LSAT[y],law82$GPA[y]))}
> hist(samplecor, breaks=40, probability=T, xlim=c(-0.3,1), ylim=c(0,3.5),
    xlab="Random_Samples", ylab="",main="")
```