Application of F-tests

Signif. codes - how significant a predictor is (more stars - more signif.)

6.1 - Data and model

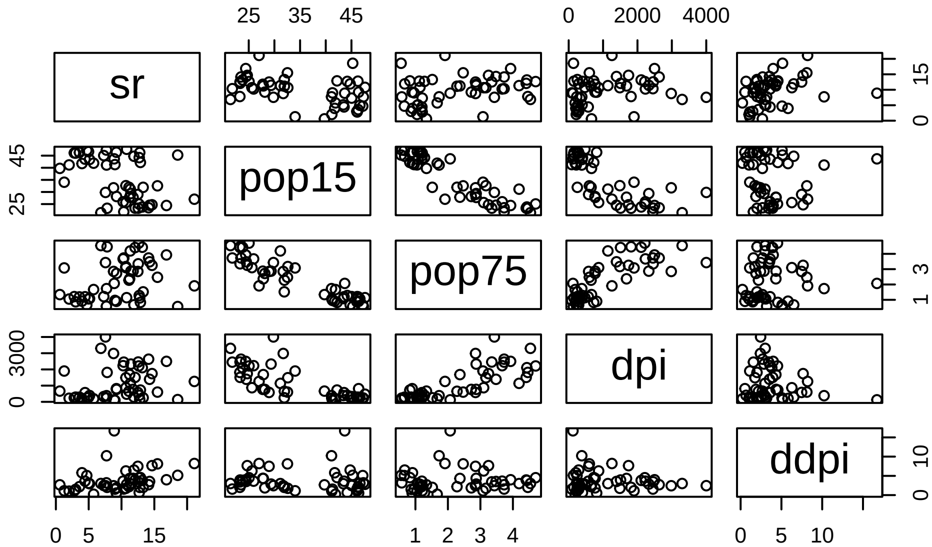
```
library(faraway)
data(savings)
attach(savings)
head(savings)
```

```
             sr pop15 pop75     dpi ddpi
Australia 11.43 29.35  2.87 2329.68 2.87
Austria   12.07 23.32  4.41 1507.99 3.93
Belgium   13.17 23.80  4.43 2108.47 3.82
Bolivia    5.75 41.89  1.67  189.13 0.22
Brazil    12.88 42.19  0.83  728.47 4.56
Canada     8.79 31.72  2.85 2982.88 2.43
```

```
nrow(savings)
```

```
[1] 50
```

```
pairs(savings)
```




```r
fit = lm(sr ~ pop15 + pop75 + dpi + ddpi, data = savings)
summary(fit)
```

```
Call:
lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = savings)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.2422 -2.6857 -0.2488  2.4280  9.7509 

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 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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
```

6.2 - Testing all the predictors

```
fit0 = lm(sr ~ 1, data = savings)

X = model.matrix(fit)
n = nrow(X)
p = ncol(X)

RSS = deviance(fit)
RSS0 = deviance(fit0)

F = ((RSS0 - RSS) / (p - 1)) / (RSS / (n - p))
F
```

```
[1] 5.755681
```

```
quantile = qf(0.95, p - 1, n - p)
quantile
```

```
[1] 2.578739
```

```
p_val = 1 - pf(F, p - 1, n - p)
p_val
```

```
[1] 0.0007903779
```

6.3 - F test for single predictor

```
fit1 = lm(sr ~ pop75 + dpi + ddpi)

p1 = 4
RSS1 = deviance(fit1)

F = ((RSS1 - RSS) / (p - p1)) / (RSS / (n - p))
F
```

```
[1] 10.16659
```

```r
p_val = 1 - pf(F, p - p1, n - p)
p_val
```

```
[1] 0.002603019
```

t - test

```r
X = model.matrix(fit)
beta = coefficients(fit)
c = c(0, 1, 0, 0, 0)

SEpsi = sigma(fit) * sqrt(t(c) %*% solve(t(X) %*% X) %*% c)

t = (t(c) %*% beta) / SEpsi
t
```

```
         [,1]
[1,] -3.18851
```

```r
p_val = 2 * (1 - pt(abs(t), n - p))
p_val
```

```
            [,1]
[1,] 0.002603019
```

anova

```r
anova(fit1, fit)
```

```
Analysis of Variance Table

Model 1: sr ~ pop75 + dpi + ddpi
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     46 797.72
2     45 650.71  1    147.01 10.167 0.002603 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6.4 - excluding pop75 and ddpi

```r
fit2 = lm(sr ~ pop15 + dpi, data = savings)
anova(fit2, fit)
```

```
Analysis of Variance Table

Model 1: sr ~ pop15 + dpi
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     47 744.12
2     45 650.71  2    93.411 3.2299 0.04889 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6.5 - same coefficients

```r
fit3 = lm(sr ~ I(pop15 + pop75) + dpi + ddpi, data = savings)
anova(fit3, fit)
```

```
Analysis of Variance Table

Model 1: sr ~ I(pop15 + pop75) + dpi + ddpi
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     46 673.63
2     45 650.71  1    22.915 1.5847 0.2146
```

6.6 - testing concrete value of coefficient

```r
fit4 = lm(sr ~ pop15 + pop75 + dpi + offset(0.5 * ddpi), data = savings)
anova(fit4, fit)
```

```
Analysis of Variance Table

Model 1: sr ~ pop15 + pop75 + dpi + offset(0.5 * ddpi)
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     46 653.78
2     45 650.71  1    3.0635 0.2119 0.6475
```

```r
X = model.matrix(fit)
beta = coefficients(fit)
c = c(0, 0, 0, 0, 1)
psi = 0.5
SEpsi = sigma(fit) * sqrt(t(c) %*% solve(t(X) %*% X) %*% c)

t = (t(c) %*% beta - psi) / SEpsi
t
```

```
           [,1]
[1,] -0.4602772
```

```r
t ^ 2
```

```
          [,1]
[1,] 0.2118551
```

```r
p_val = 2 * (1 - pt(abs(t), n - p))
p_val
```

```
          [,1]
[1,] 0.6475337
```