SD2 - uge 1, tirsdag

Anne Petersen

1.2

Loader data og ser på det:

```
setwd("C:/Users/Anne/Dropbox/Arbejde/STATforLIFE2/uge1")
potter <- read.table("potter.txt", header=T)
potter
```

```
##    Treat Nitrogen
## 1      A     19.4
## 2      A     32.6
## 3      A     27.0
## 4      A     32.1
## 5      A     33.0
## 6      B     17.7
## 7      B     24.8
## 8      B     27.9
## 9      B     25.2
## 10     B     24.3
## 11     C     17.0
## 12     C     19.4
## 13     C      9.1
## 14     C     11.9
## 15     C     15.8
## 16     D     20.7
## 17     D     21.0
## 18     D     20.5
## 19     D     18.8
## 20     D     18.6
## 21     E     14.3
## 22     E     14.4
## 23     E     11.8
## 24     E     11.6
## 25     E     14.2
## 26     F     17.3
## 27     F     19.3
## 28     F     19.1
## 29     F     16.9
## 30     F     20.8
```

```
names(potter)
```

```
## [1] "Treat"    "Nitrogen"
```

```
dim(potter)
```

```
## [1] 30  2
```

```
potter$Nitrogen
```

```
##  [1] 19.4 32.6 27.0 32.1 33.0 17.7 24.8 27.9 25.2 24.3 17.0 19.4  9.1 11.9
## [15] 15.8 20.7 21.0 20.5 18.8 18.6 14.3 14.4 11.8 11.6 14.2 17.3 19.3 19.1
## [29] 16.9 20.8
```

```
#Attacher data:
attach(potter)
#og nu kan vi bare skrive variabelnavnene uden $:
Nitrogen
```

```
##  [1] 19.4 32.6 27.0 32.1 33.0 17.7 24.8 27.9 25.2 24.3 17.0 19.4  9.1 11.9
## [15] 15.8 20.7 21.0 20.5 18.8 18.6 14.3 14.4 11.8 11.6 14.2 17.3 19.3 19.1
## [29] 16.9 20.8
```

Bruger forskellige kommandoer til at få et overblik over variablen `Nitrogen`:

```
mean(Nitrogen) #middelværdi
```

```
## [1] 19.88333
```

```
var(Nitrogen) #varians
```

```
## [1] 38.96833
```

```
median(Nitrogen) #median
```

```
## [1] 19.2
```

```
sd(Nitrogen) #spredning
```

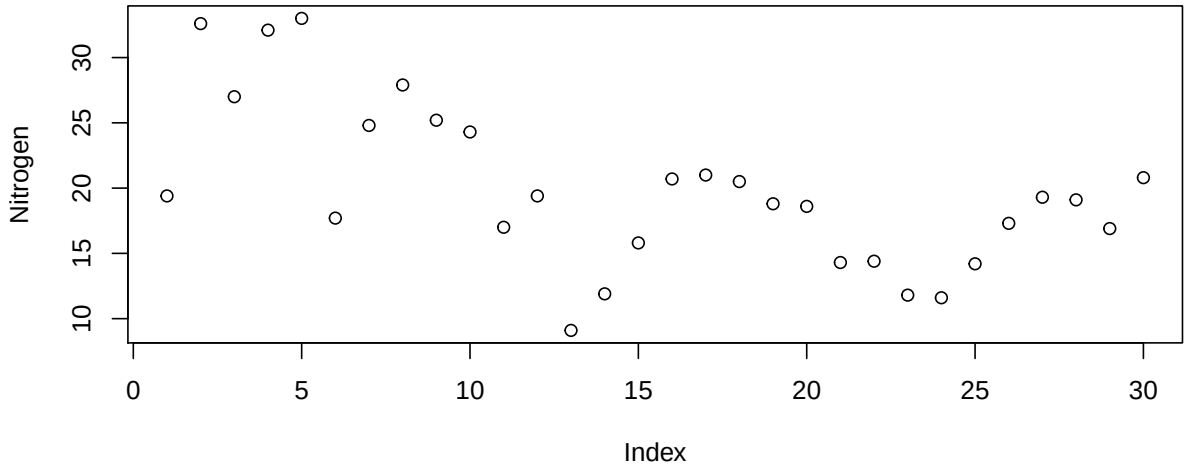
```
## [1] 6.242462
```

```
summary(Nitrogen) #kvartiler, mindste og største observation
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.10   16.08   19.20   19.88   23.48   33.00
```

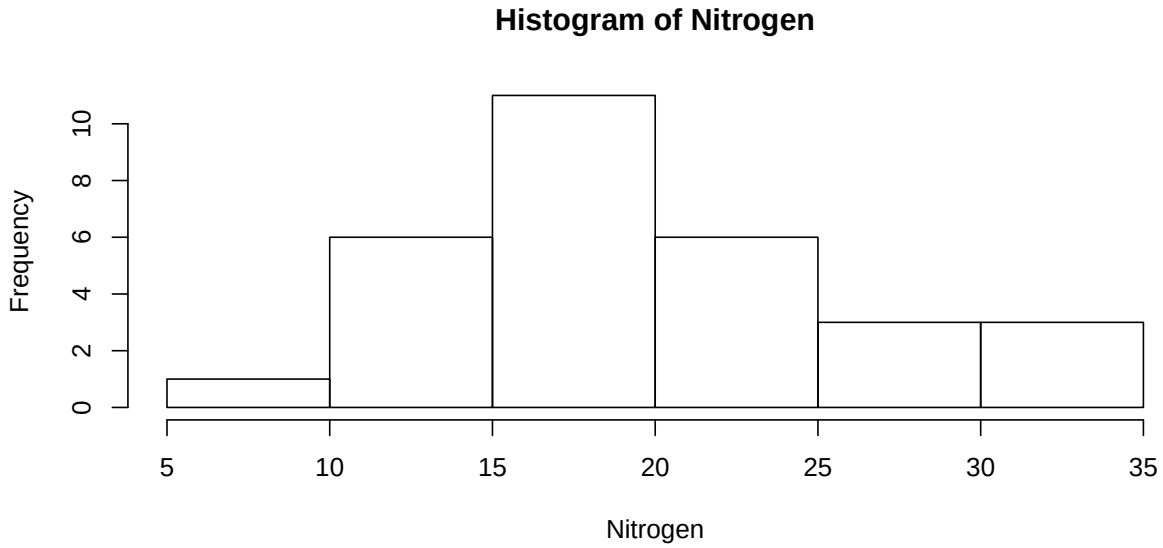
Vi laver et scatterplot med indeks (observationsnummer) på x-aksen:

```
plot(Nitrogen)
```



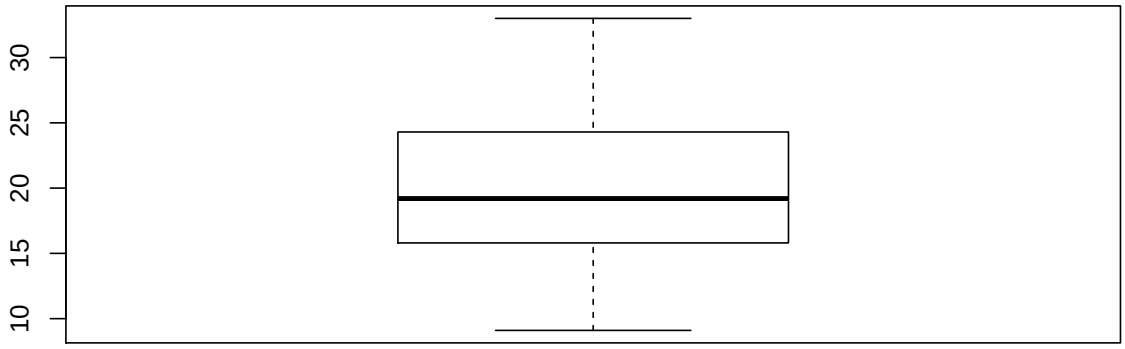
Det er ret meningsløst - vi har ikke nogen grund til at tro at observationernes rækkefølge skulle være interessant. Vi prøver at lave et histogram i stedet:

```
hist(Nitrogen)
```



Nu kan vi se fordelingen af variablen. Vi kan også lave et boxplot for at få et mere præcist indtryk af skævheder i fordelingen, dens bredde m.m.:

```
boxplot(Nitrogen)
```



Laver `Nitrogen2` som indeholder elementerne 1, 6, 11, 16, 21 og 26 fra Nitrogen:

```
Nitrogen2 <- c(Nitrogen[1], Nitrogen[6], Nitrogen[11], Nitrogen[16],
               Nitrogen[21], Nitrogen[26])
#Alternativ (og smartere) metode:
Nitrogen2 <- Nitrogen[c(1,6,11,16,21,26)]
Nitrogen2
```

```
## [1] 19.4 17.7 17.0 20.7 14.3 17.3
```

Gemmer elementer fra `Nitrogen` som er større end 25 i en ny vektor:

```
Nitrogen3 <- Nitrogen[Nitrogen > 25]
Nitrogen3
```

```
## [1] 32.6 27.0 32.1 33.0 27.9 25.2
```

Opstiller ensidet variansanalysemodel med `Treat` som faktor og `Nitrogen` som afhængig variabel (ingen referencegruppe):

```
model <- lm(Nitrogen ~ Treat-1)
```

Ser hvilke værdier af `Nitrogen` hver observation har ifølge modellen:

```
predict(model)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## 28.82 28.82 28.82 28.82 28.82 23.98 23.98 23.98 23.98 23.98 14.64 14.64
##    13    14    15    16    17    18    19    20    21    22    23    24
## 14.64 14.64 14.64 19.92 19.92 19.92 19.92 19.92 13.26 13.26 13.26 13.26
##    25    26    27    28    29    30
## 13.26 18.68 18.68 18.68 18.68 18.68
```

Ser modellens estimater for `Nitrogen` for hver gruppe (dvs. modellens parameterestimater):

```
coef(model)
```

```
## TreatA TreatB TreatC TreatD TreatE TreatF
##  28.82  23.98  14.64  19.92  13.26  18.68
```

Ser residualerne for hver observation, dvs. hvor meget observationen afviger fra det, modellen prædikterer:

```
resid(model)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## -9.42  3.78 -1.82  3.28  4.18 -6.28  0.82  3.92  1.22  0.32  2.36  4.76
##    13    14    15    16    17    18    19    20    21    22    23    24
## -5.54 -2.74  1.16  0.78  1.08  0.58 -1.12 -1.32  1.04  1.14 -1.46 -1.66
##    25    26    27    28    29    30
##  0.94 -1.38  0.62  0.42 -1.78  2.12
```

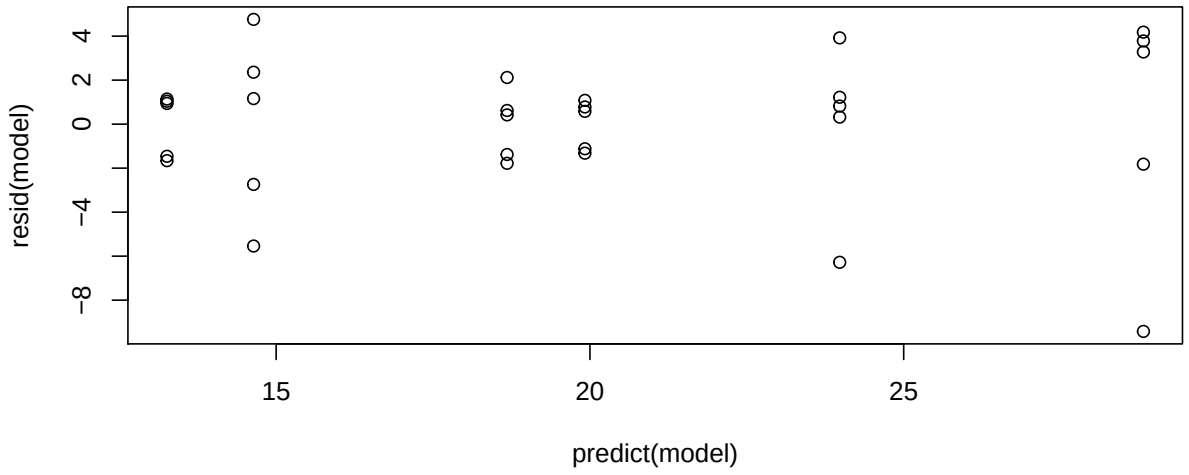
Ser 95%-konfidensintervaller for hver af modellens parameterestimater:

```
confint(model)
```

```
##           2.5 %   97.5 %
## TreatA 25.65164 31.98836
## TreatB 20.81164 27.14836
## TreatC 11.47164 17.80836
## TreatD 16.75164 23.08836
## TreatE 10.09164 16.42836
## TreatF 15.51164 21.84836
```

Laver et residualplot for modellen:

```
plot(predict(model), resid(model))
```



Laver en tom model, dvs. model hvor faktoren `Treat` ingen effekt har:

```
model2 <- lm(Nitrogen ~1)
```

Tester om `model2` er lige så god som `model`, dvs. om der er en effekt af `Treat`:

```
anova(model2, model)
```

```
## Analysis of Variance Table
##
## Model 1: Nitrogen ~ 1
## Model 2: Nitrogen ~ Treat - 1
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     29 1130.1
## 2     24  282.8  5    847.29 14.381 1.475e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vi ser at $p < 0.05$, altså forkastes hypotesen om ingen effekt af `Treat` - dvs. der er en signifikant effekt af `Treat`

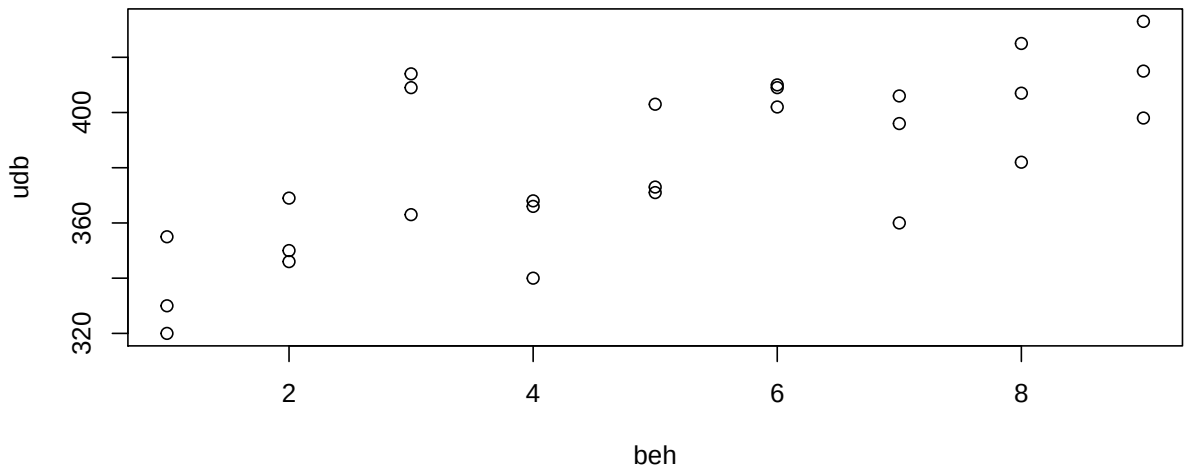
Opgave 1.3

Indlæser data:

```
beh <- rep(c(1,2,3,4,5,6,7,8,9),each=3)
udb <- c(330,320,355,346,350,369,409,363,414,368,340,366,371,
         373,403,409,410,402,360,396,406,382,407,425,398,415,433)
```

Plotter udbytte mod behandling:

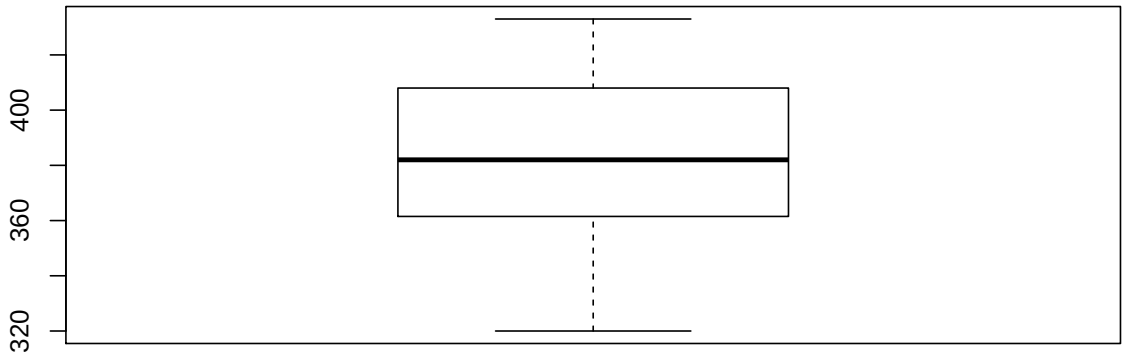
```
plot(beh, udb)
```



Det er ikke specielt meningsfuld. Vi har jo ikke nogen information, som gør det rimeligt at antage, at behandlingerne meningsfuldt kan forstås som tal - de kunne lige så godt hedde "A", "B", "C" osv. ud fra det vi ved.

Vi laver et boxplot for variablen `udb`:

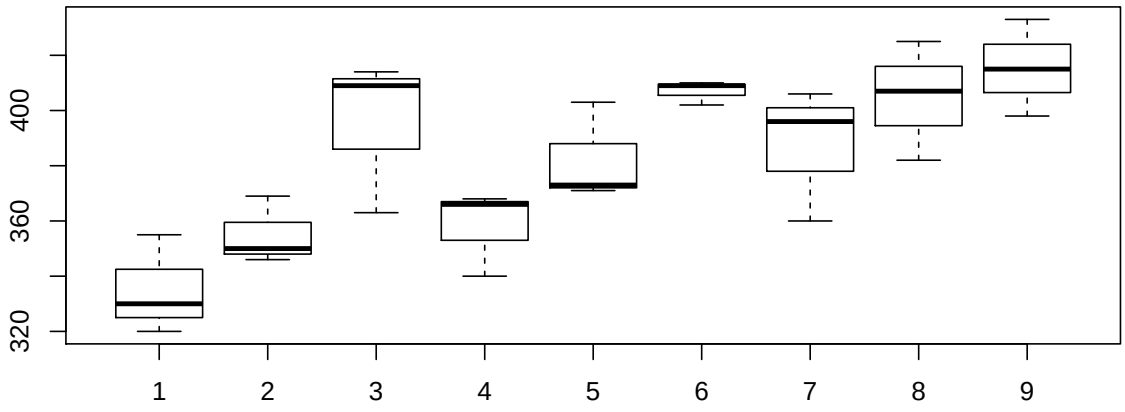
```
boxplot(udb)
```



Dette fortæller os noget om fordelingen af udbytte. Men det siger intet om sammenhængen mellem udbytte og behandling.

Vi laver nu en faktorversion af behandling og plotter udbytte mod denne nye behandlingsvariabel:

```
behfac <- factor(beh)
plot(behfac, udb)
```



Og vi ser at R nu laver ni boxplots i stedet for et scatterplot - et for hver behandlingstype. Det gør det en del lettere at sige noget om forskellen på de forskellige behandlinger.

Vi fitter en etsidet variansanalysemodel (ingen referencekategori):

```
fosformodel <- lm(udb ~behfac-1)
```

og vi kigger på parameterestimaterne:

```
summary(fosformodel)
```

```
##
## Call:
## lm(formula = udb ~ behfac - 1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32.33 -10.33   2.00  13.83  20.67
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## behfac1    335.0       10.9   30.73   <2e-16 ***
## behfac2    355.0       10.9   32.56   <2e-16 ***
## behfac3    395.3       10.9   36.27   <2e-16 ***
## behfac4    358.0       10.9   32.84   <2e-16 ***
## behfac5    382.3       10.9   35.07   <2e-16 ***
## behfac6    407.0       10.9   37.34   <2e-16 ***
## behfac7    387.3       10.9   35.53   <2e-16 ***
## behfac8    404.7       10.9   37.12   <2e-16 ***
## behfac9    415.3       10.9   38.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.88 on 18 degrees of freedom
## Multiple R-squared:  0.9984, Adjusted R-squared:  0.9976
## F-statistic:  1235 on 9 and 18 DF,  p-value: < 2.2e-16
```

og bemærker at det er de samme tal, der kommer ud som prædikterede værdier, blot gentaget alt efter behandlingsgruppen:

```
predict(fosformodel)
```

```
##        1        2        3        4        5        6        7        8
## 335.0000 335.0000 335.0000 355.0000 355.0000 355.0000 395.3333 395.3333
##        9       10       11       12       13       14       15       16
## 395.3333 358.0000 358.0000 358.0000 382.3333 382.3333 382.3333 407.0000
##       17       18       19       20       21       22       23       24
## 407.0000 407.0000 387.3333 387.3333 387.3333 404.6667 404.6667 404.6667
##       25       26       27
## 415.3333 415.3333 415.3333
```

Vi undersøger om der en effekt af behandling:

```
anova(lm(udb ~1), fosformodel)
```

```
## Analysis of Variance Table
##
## Model 1: udb ~ 1
## Model 2: udb ~ behfac - 1
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     26 24326.7
## 2     18  6417.3  8     17909 6.2792 0.0006054 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

og vi ser at $p < 0.05$, dvs. der er en signifikant effekt af behandling.

Vi indlæser nu det udvidede fosfor-datasæt:

```
fosfor <- read.table("fosfor.txt", header=T)
fosfor
```

```
##    fosfor p81 p82 blok udbytte
## 1       1   0   0    1     330
## 2       1   0   0    2     320
## 3       1   0   0    3     355
## 4       2   0  20    1     346
## 5       2   0  20    2     350
## 6       2   0  20    3     369
## 7       3   0  40    1     409
## 8       3   0  40    2     363
## 9       3   0  40    3     414
## 10      4  30   0    1     368
## 11      4  30   0    2     340
## 12      4  30   0    3     366
## 13      5  30  20    1     371
## 14      5  30  20    2     373
## 15      5  30  20    3     403
## 16      6  30  40    1     409
## 17      6  30  40    2     410
## 18      6  30  40    3     402
## 19      7  60   0    1     360
## 20      7  60   0    2     396
## 21      7  60   0    3     406
## 22      8  60  20    1     382
## 23      8  60  20    2     407
## 24      8  60  20    3     425
## 25      9  60  40    1     398
## 26      9  60  40    2     415
## 27      9  60  40    3     433
```

Bemærk at variablen `fosfor` svarer til `beh` fra ovenstående og at variablen `p82` er grovere end `fosfor`: Hvis vi kender `fosfor` for en observation, kender vi også dens værdi af `p82`, men det gælder ikke den anden vej rundt.

Vi fitter ny model hvor kun `p82` antages at have en effekt:

```
model82 <- lm(udbytte ~ factor(p82)-1, data=fosfor)
```

Vi tester en model med `fosfor` som forklarende variabel (svarende til `fosformodel` ovenfor) mod en model, som bruger `p82` som forklarende variabel:

```
anova(lm(udbytte ~factor(fosfor)-1, data=fosfor), model82)
```

```
## Analysis of Variance Table
##
## Model 1: udbytte ~ factor(fosfor) - 1
## Model 2: udbytte ~ factor(p82) - 1
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1     18  6417.3
## 2     24 14863.8 -6   -8446.4 3.9486 0.01076 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Vi finder $p < 0.05$, og altså er der en signifikant forskel på modellerne. Da `p82` er en forsimpling af `fosfor`, er denne form for test meningsfuld og vi konkluderer at udbyttet ikke kun afhænger af fosfortilførslen i 1982, men at vi også skal bruge variablen `fosfor` for at få den bedst mulige model.

1.9

Vi tæller antal observationer med hver behandling:

```
table(fosfor$fosfor)
```

```
##
## 1 2 3 4 5 6 7 8 9
## 3 3 3 3 3 3 3 3 3
```

Vi tæller antal observationer for her kombination af p81 og p82:

```
table(fosfor$p81, fosfor$p82)
```

```
##
##      0 20 40
##   0  3  3  3
##   30 3  3  3
##   60 3  3  3
```

Vi udtrækker den del af datasættet som havde p82=0:

```
fos0 <- subset(fosfor, p82==0)
```

Vi beregner middelværdi og spredning for udbyttet for de observationer, som har p82=0 og tegner et boxplot over deres udbytter:

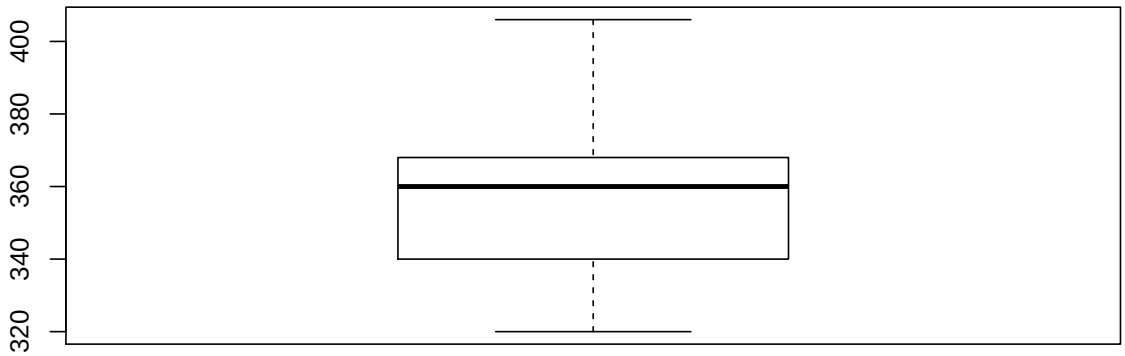
```r
mean(fos0$udbytte) #middelværdi
```

```
## [1] 360.1111
```

```r
sd(fos0$udbytte) #spredning
```

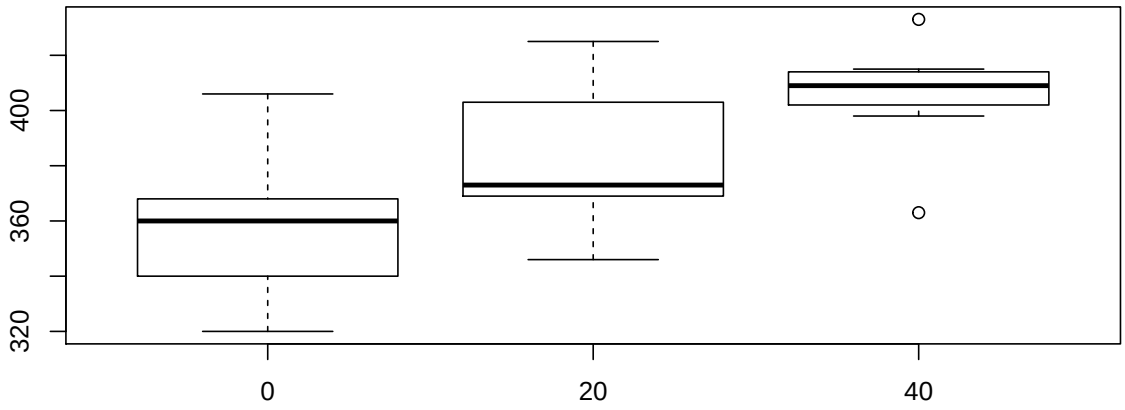
```
## [1] 28.36567
```

```r
boxplot(fos0$udbytte)
```



Vi tegner boxplots over udbytte alt efter værdien af p82:

```r
boxplot(udbytte~p82, data=fosfor)
```



Vi udskriver de første 10 linier af datasættet:

```
head(fosfor, 10)
```

```
##    fosfor p81 p82 blok udbytte
## 1       1   0   0    1     330
## 2       1   0   0    2     320
## 3       1   0   0    3     355
## 4       2   0  20    1     346
## 5       2   0  20    2     350
## 6       2   0  20    3     369
## 7       3   0  40    1     409
## 8       3   0  40    2     363
## 9       3   0  40    3     414
## 10      4  30   0    1     368
```

Vi udskriver rækkenumrene for de observationer, som har p82=0:

```
rownames(fos0)
```

```
## [1] "1"  "2"  "3"  "10" "11" "12" "19" "20" "21"
```

```
#Alternativ metode
which(fosfor$p82==0)
```

```
## [1]  1  2  3 10 11 12 19 20 21
```

Vi udskriver rækkenumrene for de observationer, som har p81=p82=0:

```
rownames(subset(fosfor, p82==0 & p81==0))
```

```
## [1] "1" "2" "3"
```

```
#Alternativ metode:
which(fosfor$p81==0 & fosfor$p82==0)
```

```
## [1] 1 2 3
```

Vi laver en ny dataframe uden observation nummer 8:

```
fosno8 <- fosfor[-8,]
fosno8
```

```
##    fosfor p81 p82 blok udbytte
## 1       1   0   0    1     330
## 2       1   0   0    2     320
## 3       1   0   0    3     355
## 4       2   0  20    1     346
## 5       2   0  20    2     350
## 6       2   0  20    3     369
```

```
## 7       3   0  40    1     409
## 9       3   0  40    3     414
## 10      4  30   0    1     368
## 11      4  30   0    2     340
## 12      4  30   0    3     366
## 13      5  30  20    1     371
## 14      5  30  20    2     373
## 15      5  30  20    3     403
## 16      6  30  40    1     409
## 17      6  30  40    2     410
## 18      6  30  40    3     402
## 19      7  60   0    1     360
## 20      7  60   0    2     396
## 21      7  60   0    3     406
## 22      8  60  20    1     382
## 23      8  60  20    2     407
## 24      8  60  20    3     425
## 25      9  60  40    1     398
## 26      9  60  40    2     415
## 27      9  60  40    3     433
```