

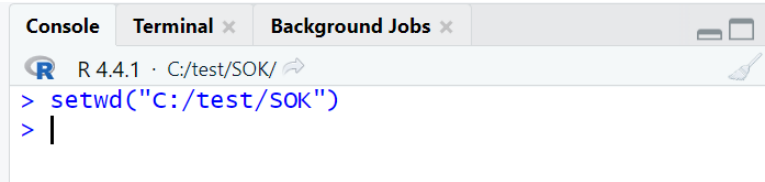
Datorlaboration 4, Statistisk översiktscurs (SÖK)

I den här datorlabben kommer vi arbeta med korrelation och regression

I appendix till labben (näst sista sidan) finns information om hur du får fram olika tecken på ditt tangentbord (bland annat tecknen ~ och |).

Innan själva uppgifterna börjar, gör följande (arbetskatalog, fil att spara i, ladda paket)

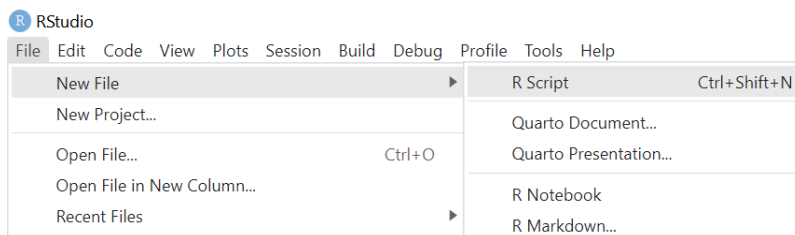
1. I Console i R, kör ditt **setwd**-kommando från labb 1 som sätter din arbetskatalog i R till din **SOK-mapp** på datorns hårddisk. Kommandot ser olika ut beroende på var på datorn din SOK-mapp finns.



```
R 4.4.1 · C:/test/SOK/
> setwd("C:/test/SOK")
> |
```

Du kan alternativt göra detta steg genom menyerna i R, som vi gjorde i kapitel 6 i laboration 1, dvs: Välj menyn **Session** i R, sedan **Set Working Directory**, sedan **Choose Directory...** och klicka dig fram till mappen **SOK**.

2. Skapa en textfil där du kommer spara dina labb4-kommandon (ett "R-script") genom att i menyn i R välja **File** och sedan **New File** och sedan **R script**.



3. Labb 4 förutsätter att följande paket finns installerade: **corrplot** och **mosaic**

Paket installeras via kommandot `install.packages('packagename')`, där `packagename` är paketnamnet. Kör kommandot:

`install.packages('corrplot')`

`mosaic`-paketet ska du redan ha installerat, om inte, installera det paketet också. Skriv sedan in följande `library`-kommandon i din textfil:


`library(mosaic)`

`library(corrplot)`

Kör sen de två `library`-kommandona, ställ dig på första raden med ett `library`-kommando och tryck på Run, två gånger.

4. På samma sätt som i tidigare labbar är det bra att som första kommando i textfilen ha kommandot som bestämmer arbetskatalogen i R. Du kan kopiera in kommandot du körde i steg 1 ovan, in i din textfil (kopiera från Console eller från History-fliken i övre högra delen av RStudio.)



5. Spara din textfil genom att trycka på sparasymbolen . Välj lämpligt namn. Spara din fil ofta.

1. Läs in CAPM_data

Det finns ett färdigt R-dataset som heter CAPM_data.RData som kan laddas ner [här](#) (högerklicka), från Athena (Datafiler), eller från labbhemsidan på Github. Datasetet består redan av en data frame, när du har data i din arbetskatalog kan du läsa in data med följande kommando (vilket är lite annorlunda mot tidigare):

```
load("CAPM_data.RData")
```

Bekräfta att "CAPM" dykt upp i din Environment-flik i R-studio. Notera att den data frame som laddas in heter CAPM och inte CAPM_data. Det finns ingen koppling mellan filnamnet och namnet på data framen i R. Namnet på data framen bestäms av vad den döptes till när den sparades, i det här fallet CAPM.

CAPM består av finansmarknadsdata (se mer nedan), **bekanta dig med data** genom att köra följande kommandon, som vi sett i tidigare labbar:

```
head(CAPM)
str(CAPM)
class(CAPM)
summary(CAPM)
```

Testa också dessa två kommandon:

```
is.na(CAPM)
which(is.na(CAPM))
```

Det första kommandot, som skriver ut massa "FALSE", går igenom alla data och testar logiskt om värdet = NA eller NaN (se labb 2, kapitel 7) förekommer (i så fall får vi "TRUE").

Det andra kommandot (med **which()**-kommandot "runt" det första kommandot) returnerar, om det finns NA eller NaN, en typ av "platsindex" för varje saknat värde, så att vi sen kan hitta positionerna med saknade data.

Uppgift 1.1. Finns det några saknade värden i CAPM?

Filen `CAPM_data.RData` innehåller tidsserier över månadsavkastningar för olika finansiella tillgångar samt makroekonomiska variabler såsom obligationsränta (`RKFREE`) och konsumentprisindex (`CPI`). Observationerna är ordnade i tid, där första observationen är från januari 1978 och sista observationen är från december 1987. Observationerna är på månadsfrekvens, så vi har totalt 120 observationer (12 per år under 10 år). Månadsavkastningen r_t för en finansiell tillgång som är värderad till S_t i period t definieras som

$$r_t = \frac{S_t - S_{t-1}}{S_{t-1}}.$$

Exempelvis, antag att tillgången var värderad till 105 vid tidpunkten $t - 1$, dvs $S_{t-1} = 105$ och 110 vid tidpunkten t , dvs $S_t = 110$. Då är månadsavkastningen i perioden t

$$r_t = \frac{110 - 105}{105} \approx 0.048,$$

dvs en uppgång på nästan 5% jämfört med föregående månad.

2. Samband mellan två numeriska variabler

Numeriska variabler är variabler vars utfall är numeriska värden som har betydelse (se föreläsning 2, F2). En kategorisk variabel kan visserligen vara kodad som ett numeriskt värde, men värdet är godtyckligt. Ett exempel är variabeln `landlock` som vi såg i `labb2`, där ett land som "inte har kust" kodas som 1 och ett land som "har kust" kodas som 0. Vi skulle lika gärna kunna ha kodat "inte har kust" som 0 och "har kust" som 1. Eller som -1 och 1, eller några helt andra värden. `landlock` är inte numerisk.

Ett vanligt sambandsmått mellan **två numeriska variabler** är korrelation (F8). Korrelation mäter det linjära sambandet mellan variablerna. Stickprovskorrelationen räknas enligt

$$\text{Corr}(x,y) = r = \frac{1}{n-1} \sum_{i=1}^n \frac{(x_i - \bar{x})}{s_x} \frac{(y_i - \bar{y})}{s_y}$$

(där i är index för observationer, n urvalsstorleken, \bar{x} medelvärdet för x , s_x standardavvikelsen för x , \bar{y} medelvärdet för y , s_y standardavvikelsen för y .)

Vi ska illustrera korrelationsbegreppet med hjälp av vårt dataset `CAPM`, men det är viktigt att förstå att **korrelation är ett allmänt mått för det linjära sambandet mellan två variabler**, oavsett om variablerna är tidsserier eller inte. Den enda förutsättningen för att beräkna korrelationen är att variablerna är numeriska.

Låt oss beräkna korrelationen mellan två tillgångar, till exempel `IBM` och `CITCRP` med hjälp av funktionen `cor()` i `mosaic`-paketet. Kommandot använder tildetecknet (`~`) som vi sett tidigare:

```
Console Terminal x Background Jobs x
R 4.4.1 · C:/test/SOK/
> cor(IBM~CITCRP, data=CAPM)
[1] 0.4237027
> |
```

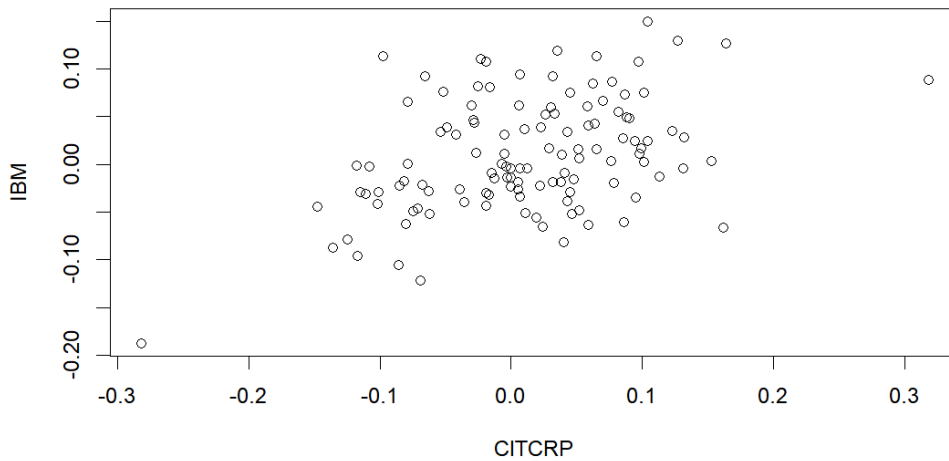
Vi ser att korrelationen är positiv och har ett värde runt 0.424.

Uppgift 2.1

Beräkna den omvända korrelationen, dvs korrelationen mellan `CITCRP` och `IBM`. Kan du förklara varför svaret är detsamma som ovan?

Eftersom korrelation endast beskriver linjära samband, antas det att variablerna förhåller sig approximativt linjärt till varandra för att korrelation ska anses vara ett lämpligt sambandsmått. Vi kan göra ett **spridningsdiagram** (F2, F8, mfl.) för att validera antagandet.

```
Console Terminal x Background Jobs x
R 4.4.1 · C:/test/SOK/
> plot(IBM~CITCRP, data=CAPM)
> |
```



Antag att vi vill beräkna den parvisa korrelationen mellan flera variabler. Vi skulle kunna repetera koden `cor(IBM ~ CITCRP, data = CAPM)`, där vi byter ut IBM och CITCRP mot alla parvisa kombinationer av de variabler vi är intresserade av. Mer elegant (och mindre tidskrävande) kan vi skapa en så kallad **korrelationsmatris**. För att illustrera korrelationsmatrisen, låt oss betrakta följande sju variabler: MARKET, RKFREE, BOISE, CITCRP, IBM, WEYER och CPI. De fyra variablerna förutom MARKET, RKFREE och CPI är månadsavkastning för stora börsnoterade företag. MARKET är marknadens månadsavkastning, RKFREE är avkastningen på en riskfri tillgång (statsobligationsränta) och CPI är konsumentprisindex.

Vi skapar en ny data frame där vi enbart behåller variablerna av intresse (vid behov: se indexering i labb 1, kap. 10), samt titta på de första raderna i vår nya data frame:

```

R 4.4.1 · C:/test/SOK/
> CAPM_7_variables <- CAPM[, c("MARKET", "RKFREE", "BOISE", "CITCRP", "IBM", "WEYER", "CPI")]
> head(CAPM_7_variables)
  MARKET RKFREE BOISE CITCRP IBM WEYER CPI
1 -0.045 0.00487 -0.079 -0.115 -0.029 -0.116 166.7
2  0.010 0.00494  0.013 -0.019 -0.043 -0.135 167.1
3  0.050 0.00526  0.070  0.059 -0.063  0.084 167.5
4  0.063 0.00491  0.120  0.127  0.130  0.144 168.2
5  0.067 0.00513  0.071  0.005 -0.018 -0.031 169.2
6  0.007 0.00527 -0.098  0.007 -0.004  0.005 170.1
> |

```

Vi kan nu skapa och spara korrelationsmatrisen i objektet `correlation_matrix_CAPM` (eller något annat lämpligt namn) med kommandot `cor()`. Notera att input till funktionen är vår nya data frame. I koden nedan skapar vi först objektet (korrelationsmatrisen), skriver sen ut objektet, och skriver sen ut objektet igen, avrundat till tre decimaler.

```

R 4.4.1 · C:/test/SOK/
> correlation_matrix_CAPM <- cor(CAPM_7_variables)
> correlation_matrix_CAPM
  MARKET RKFREE BOISE CITCRP IBM WEYER CPI
MARKET  1.00000000 -0.099660149  0.65247145  0.563709867  0.524628090  0.65637580 -0.059722724
RKFREE -0.09966015  1.000000000 -0.17574987  0.001121694 -0.107327855 -0.14202262 -0.455649631
BOISE  0.65247145 -0.175749868  1.000000000  0.589834599  0.453575798  0.75142308  0.059048864
CITCRP  0.56370987  0.001121694  0.58983460  1.000000000  0.423702674  0.53999649  0.089561571
IBM  0.52462809 -0.107327855  0.45357580  0.423702674  1.000000000  0.49181936 -0.001850249
WEYER  0.65637580 -0.142022621  0.75142308  0.539996486  0.491819361  1.00000000  0.067213865
CPI -0.05972272 -0.455649631  0.05904886  0.089561571 -0.001850249  0.06721387  1.000000000
> round(correlation_matrix_CAPM,3)
  MARKET RKFREE BOISE CITCRP IBM WEYER CPI
MARKET  1.000 -0.100  0.652  0.564  0.525  0.656 -0.060
RKFREE -0.100  1.000 -0.176  0.001 -0.107 -0.142 -0.456
BOISE  0.652 -0.176  1.000  0.590  0.454  0.751  0.059
CITCRP  0.564  0.001  0.590  1.000  0.424  0.540  0.090
IBM  0.525 -0.107  0.454  0.424  1.000  0.492 -0.002
WEYER  0.656 -0.142  0.751  0.540  0.492  1.000  0.067
CPI -0.060 -0.456  0.059  0.090 -0.002  0.067  1.000
> |

```

Uppgift 2.2

Vilken är den största (positiva eller negativa korrelationen? (bortse från 1:orna på diagonalen)

Uppgift 2.3

Ta fram ett spridningsdiagram för de variabler som har den största korrelationen. Ser sambandet linjärt ut?

Uppgift 2.4

Ta fram ett spridningsdiagram mellan IBM och RKFREE. Stämmer diagrammets utseende överens med vad du förväntade dig från korrelationen mellan de två variablerna (i tabellen ovan)?

Uppgift 2.5

Varför är det 1:or på diagonalen i korrelationsmatrisen?

Uppgift 2.6* (om du vill) (viss inspiration kan finnas i labb 1, kapitel 10)

Använd följande kommando för att spara / göra om din korrelationsmatris till en data frame:

```
correlation_dataframe_CAPM <- data.frame(correlation_matrix_CAPM)
```

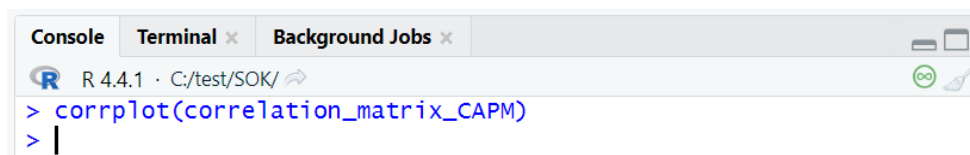
Nu har du en data frame där du kan använda \$-notationen för att komma åt enskilda variabler (kolumner).

Testa att använda kommandona `min()` och `max()`, exempelvis för en viss kolumn, för att ta fram minsta respektive största värden på korrelationskoefficienter.

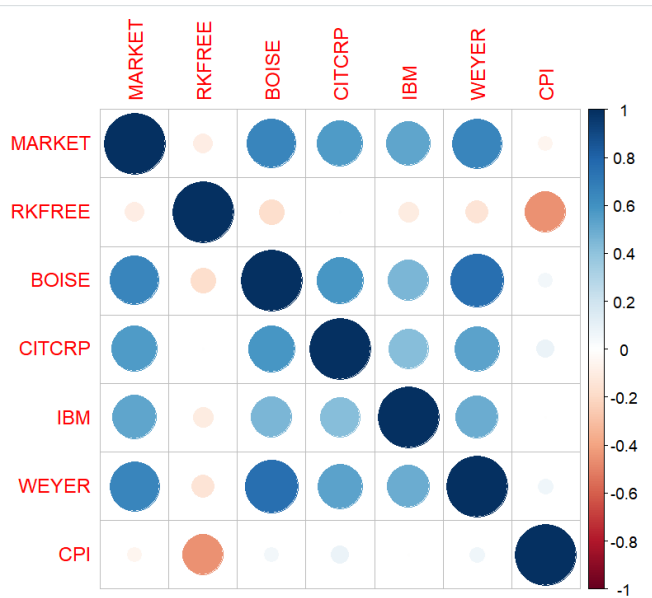
Uppgift 2.7

Använd `cor()` med formelsyntax (dvs. med tildetecknet) för att beräkna korrelationen mellan IBM och WEYER (ha mosaicpaketet laddat). Stäm av att resultatet är detsamma som korrelationsmatrisen visar (tänk på att vi avrundade, resultaten kommer därför inte stämma exakt).

Informationen i en korrelationsmatris kan vara svår att utläsa pga. alltför många siffror. En korrelationsplot som illustrerar korrelationerna med färgskalor kan vara enklare att utläsa. Funktionen **`corrplot()`** från `corrplot`-paketet tar en korrelationsmatris som argument för att skapa plotten. Vi skapade korrelationsmatrisen ovan (`correlation_matrix_CAPM`).



```
Console Terminal x Background Jobs x
R 4.4.1 · C:/test/SOK/
> corrplot(correlation_matrix_CAPM)
> |
```

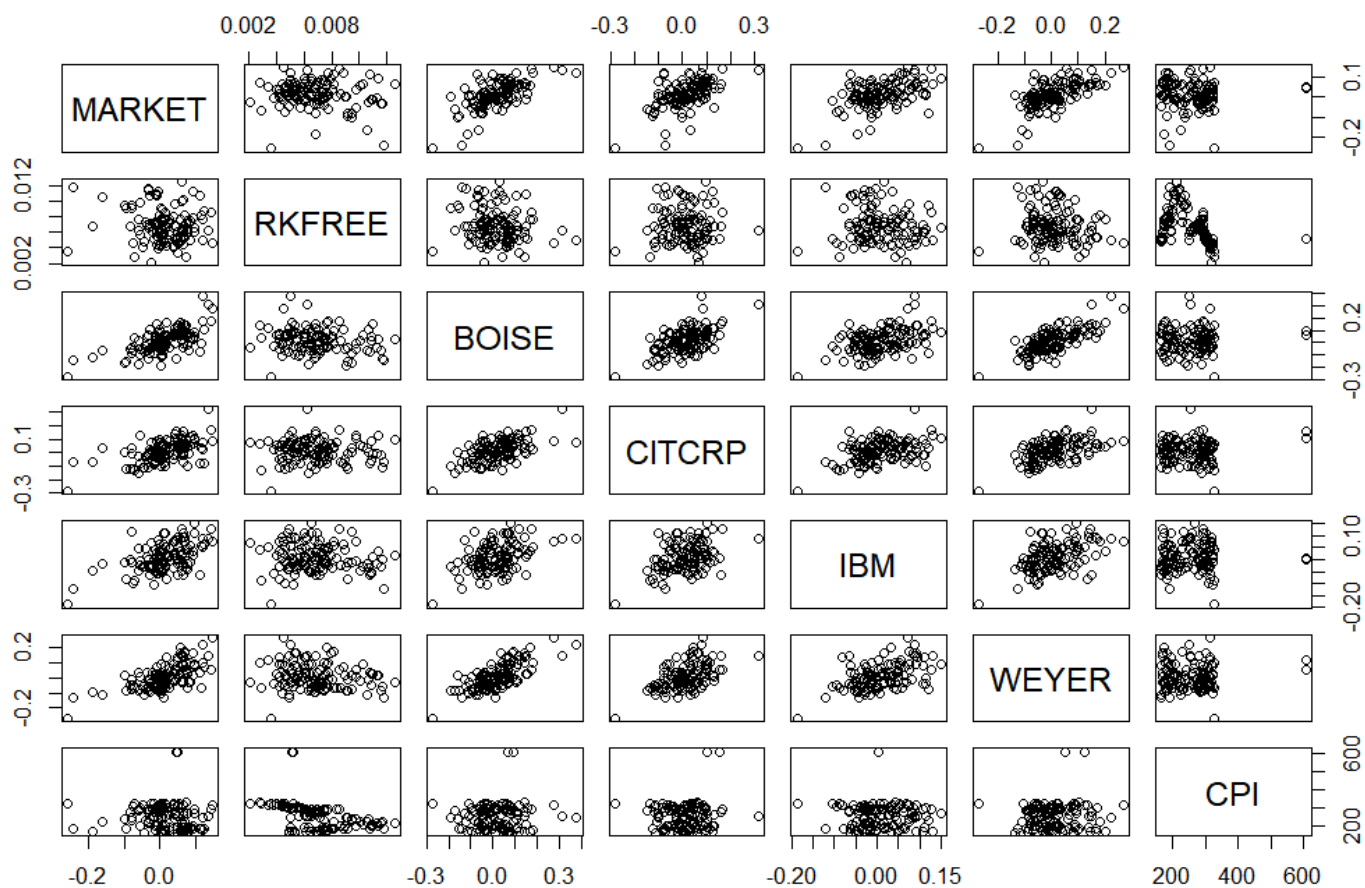


Ett intressant R-kommando är **pairs()**. Du kan i ett kommando ta fram spridningsdiagram för alla numeriska variabler plottade mot varandra, två och två. Kommandot kan vara hjälpsamt men också skapa grafer som är så små att de blir svåra att tolka. Via använder **pairs()** på vår skapade data frame **CAPM_7_variables**:

```

Console Terminal x Background Jobs x
R 4.4.1 · C:/test/SOK/
> pairs(CAPM_7_variables)
> |

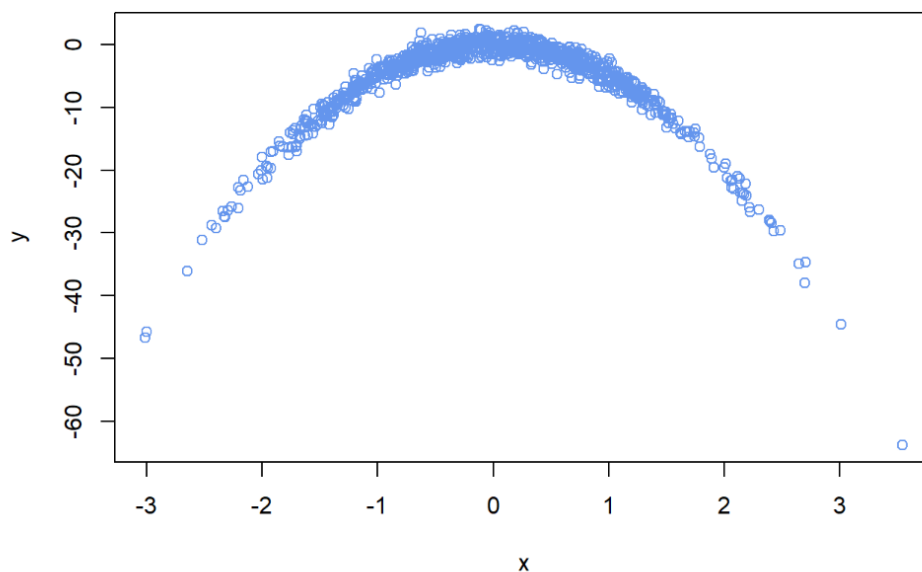
```



Korrelation är ett lämpligt sammanfattande mått när det finns linjära samband. Om sambandet mellan två variabler inte är linjärt är korrelation mindre lämpligt som mått.

Relaterat till om och när korrelationsmått är lämpligt eller inte har vi också det faktum att bara för att en korrelation mellan två variabler är nära noll betyder det inte att det inte finns något samband mellan variablerna. Låt oss illustrera detta med ett exempel där vi använder slumpstal för att skapa **två variabler x och y som har ett icke-linjärt samband**, men en korrelation som är nära noll. (Koden här är inte viktig för kursen.)

```
set.seed(10) # Same random numbers generated every run
x <- rnorm(n = 1000, sd = 1) # Simulate a normal variable with standard deviation 1
y <- -5*x^2 + rnorm(n = 1000)
plot(x, y, col = "cornflowerblue")
```



Det finns ett uppenbart kvadratisk samband mellan y och x . Låt oss beräkna korrelationen.

```
cor(x, y)
```

```
[1] -0.003297157
```

Korrelationen är nästan noll! En slutsats är att om vi bara hade fokuserat på korrelationen, utan att plotta y mot x , så hade vi alltså missat detta uppenbara samband.

En annan mycket viktig slutsats är att man alltid ska plotta sina variabler och det man ska analysera. Om vi ska analysera hur två variabler samvarierar ska vi ALLTID plotta variablerna mot varandra.

3. Läs in FevChildren

Det finns ett färdigt R-dataset som heter FevChildren.RData som kan laddas ner [här](#) (högerklicka), från Athena (Datafiler), eller från labbhemsidan på Github. Datasetet består redan av en data frame, när du har data i din arbetskatalog kan du läsa in data med följande kommando:

```
load("FevChildren.RData")
```

Du ska nu ha en data frame som heter FevChildren (dubbelkolla i din Environment-flik). Datasetet innehåller data över lungkapacitet (forcerad utandningsvolym) (forced expiratory volume, FEV, på engelska), hos 606 barn och

ungdomar. De numeriska variablerna i datasetet är forcerad utandningsvolym i liter (fev), längd i cm (height) och ålder i år (age). De kategoriska variablerna är rökare (smoking, ja kodat som 1 och nej kodat som 0), åldersgrupp (age.group) och kön (gender).

Inspektera data med följande kommandon

```
head(FevChildren)
str(FevChildren)
class(FevChildren)
summary(FevChildren)
is.na(FevChildren)
which(is.na(FevChildren))
```

Gör ett histogram över utandningsvolymen, för att studera variationen i data. Ta fram medelvärde och median.

```
histogram(FevChildren$fev)
mean(FevChildren$fev)
median(FevChildren$fev)
```

Uppgift 3.1

Beskriv fördelningen av variabeln fev. Är fördelningen centrerad eller har vi skevhet åt något håll? Säger oss skillnaden mellan medelvärde och median något vad gäller hur fördelningen ser ut? (Jämför exv. med fördelningarna på sid 43 och 48 i F2.)

4. Enkel linjär regression

En linjär regression är användbar för att beskriva det linjära sambandet mellan en responsvariabel y och en förklarande variabel x . Med en anpassad linjär regressionsmodell kan man, likt ovan, beräkna korrelationen mellan y och x . Men en linjär regressionsmodell erbjuder mer än så. Vi kan **kvantifiera hur en förändring i x är associerad med en förändring i y** och vi kan, **givet ett nytt x , prediktera det genomsnittliga värdet på y** för detta x . Vi kan också göra **inferens** (F8-F10).

Låt oss illustrera enkel linjär regression med hjälp av datasetet FevChildren.RData. Vi anpassar en enkel linjär regression med responsvariabel utandningsvolymen (fev) och förklaringsvariabel längd (height) med hjälp av funktionen **lm()** som står för **linear model** och sparar resultatet i ett objekt vi döper till `lm_fev_vs_height`.

** Vid intresse - Om du är nyfiken på **lm()***

*Funktionen `lm()` returnerar ett **objekt** som är en instans av **klass** `lm`, med vissa så kallade **attribut** (vad som lagras i objektet), och vilka funktioner som går att använda på objektet. Vi kan se klasstypen genom att anropa **class()**-funktionen.*

```
class(lm_fev_vs_height)
```

```
[1] "lm"
```

*Innehållet i objektet kan visas med anropet **str(lm_fev_vs_height)**, som visar objektets struktur på ett kompakt sätt.*

Använd `str()` enligt ovan för att få en översikt av innehållet i objektet `lm_fev_vs_height`. Känner du igen några av namnen (som följer efter `$`-tecknet) från kursen?

Funktionen `lm()` returnerar ett så kallat objekt (som vi sparar i variabeln `lm_fev_vs_height`) av klass `lm` som förklaras i extramaterialet ovan. Vi behöver inte veta detaljerna, men en viktig sak att veta är att det finns många användbara funktioner vi kan använda på vårt objekt. En sådan funktion är `summary()` som skriver ut resultaten från regressionen:

```
Console Terminal x Background Jobs x
R 4.4.1 · C:/test/SOK/
> lm_fev_vs_height <- lm(fev ~ height, data = FevChildren)
> summary(lm_fev_vs_height)

Call:
lm(formula = fev ~ height, data = FevChildren)

Residuals:
    Min       1Q   Median       3Q      Max
-1.76006 -0.25417  0.00064  0.23903  2.10393

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.741452   0.210370  -27.29  <2e-16 ***
height       0.053809   0.001337   40.25  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4327 on 604 degrees of freedom
Multiple R-squared:  0.7284,    Adjusted R-squared:  0.7279
F-statistic: 1620 on 1 and 604 DF,  p-value: < 2.2e-16
```

Från F9 kan vi känna igen de skattade **regressionskoefficienterna**, **R2**, **t-värde** och **p-värde**. Gå gärna tillbaka till föreläsningen och repetera.

Vi tänker oss att de data vi har är ett urval (med $n=606$) från en population. En populationsformulering av regressionen som vi skattar ser ut som på F9, sid. 3, med X =längd, Y = utandningsvolym. De skattade koefficienterna skriver vi som b_0 och b_1 (F9, sid. 4) och när vi predikterar sätter vi en hattsymbol på Y -variabeln (F9, sid. 4).

Låt oss tolka resultaten ovan. 72.84% av variationen i forcerad utandningsvolym förklaras av variabeln längd. Minsta kvadratanpassningen är

$$\widehat{fev} = b_0 + b_1 \text{height} = -5.741 + 0.054 \text{height}.$$

Tolkningen för $b_0 = -5.741$ är den predikterade genomsnittliga forcerade utandningsvolymen för barn och ungdomar som är 0 cm långa, vilket inte är meningsfullt. Vi kan inte göra en kausal tolkning för b_1 eftersom det inte är så att längden medför bättre eller sämre lugnkapacitet. Istället säger vi att barn och ungdomar som är 1 cm längre tenderar att i genomsnitt ha $b_1 = 0.054$ fler enheter forcerad utandningsvolym (än de som är 1 cm kortare). Vi kan också använda oss av b_1 för att till exempel säga att barn och ungdomar som är 10 cm längre tenderar att ha $10 \cdot b_1 = 0.54$ fler enheter forcerad utandningsvolym (än de som är 10 cm kortare).

Vårt dataset innehåller forcerad utandningsvolym och längd hos 606 individuella barn och ungdomar. Vår anpassade modell ger oss 606 prediktioner av de genomsnittliga forcerade utandningsvolymerna, dvs en prediktion \hat{y}_i (`fev`) för varje x_i (`height`) i datasetet. Dessa kan fås genom funktionen `predict()`. Låt oss plotta data tillsammans med de predikterade värden i samma figur med hjälp av funktionen `lines()` som beskrivs lite mer nedan. Vi använder också funktionen `abline()` som ritar den räta linjen (minsta kvadratanpassningen).

```
plot(fev ~ height, data = FevChildren, col = "cornflowerblue", ylim = c(0, 7))
y_hat <- predict(lm_fev_vs_height)
head(y_hat)
```

```
1      2      3      4      5      6
2.048981 3.484061 1.707295 1.502284 2.048981 2.595678
```

```
lines(FevChildren$height, y_hat, type = "p", col = "lightcoral")
abline(lm_fev_vs_height, col = "lightcoral")
```

lines() ovan är ett användbart kommando. Varje gång du använder kommandot plot() börjar R "på nytt", med en ny figur. Med lines() kan du sedan lägga till andra linjer och kurvor i samma graf. Exempelvis ovan: i första koddelen kördes plot(), som tar fram ett spridningsdiagram, i andra koddelen används sedan lines(), med det extra argumentet **type="p"** för att rita predikterade fev-värden (y_hat), som punkter, för olika height-värden i våra data. (Därefter ritas abline() ut själva linjen). Kör kommandona plot(), lines() och abline() ovan, ett i taget, och observera vad som händer.

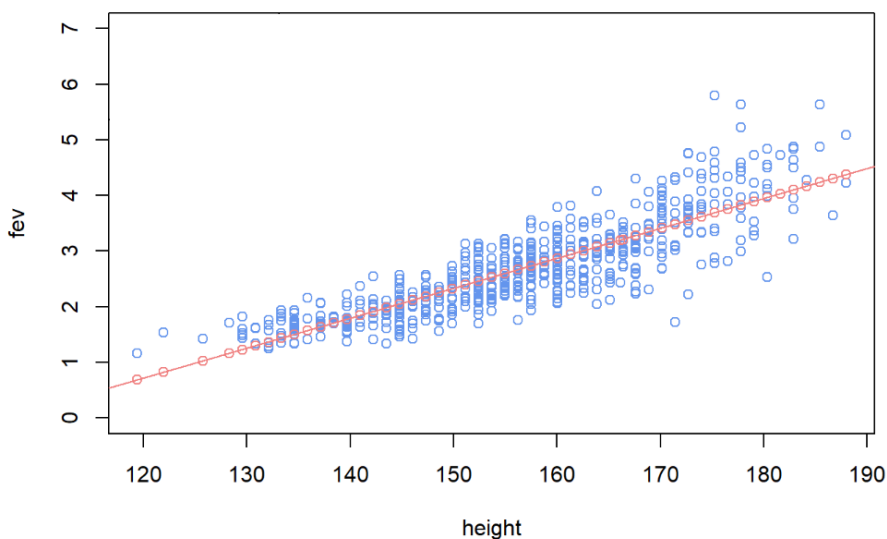
* Vid intresse

lines-argumentet ovan kan alternativt skrivas

```
lines(yhat ~ FevChildren$height, type = "p", col = "lightcoral")
```

formen här är $y \sim x$, medan i kommandot i rutan är formen x, y , vilket är ett alternativt sätt att skriva.

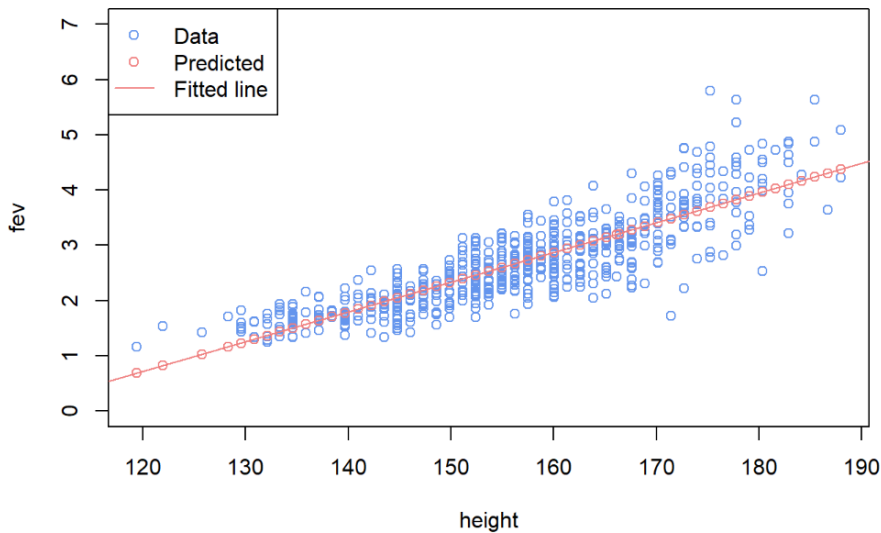
I abline är argumentet den skattade modellen



I labb 2, kapitel 5 (och i appendix), såg vi att **legend()** kan användas för att skapa en förklaringsruta / etikett, vi lägger till en förklaringsruta till grafen ovan:

Gör plottkommandot (första raden på denna sida) och sedan:

```
lines(FevChildren$height, y_hat, type = "p", col = "lightcoral")
abline(lm_fev_vs_height, col = "lightcoral")
legend(x = "topleft", pch = c(1, 1, NA), lty = c(NA, NA, 1), col = c("cornflowerblue", "lightcoral", "lightcoral"), legend=c("Data", "Predicted", "Fitted line"))
```



I legend-funktionen finns ett nytt argument, `pch=c(1, 1, NA)`, som anger att de två första ska vara cirkelsymboler i legendtexten och anger ingen cirkelsymbol för den sista. Argumentet `lty = c(NA, NA, 1)` anger en linje för den sista men ingen linje för de två första.

Antag att vi vill prediktera genomsnittliga forcerade utandningsvolymen för längder som inte finns med bland de 606 observationerna, exempelvis $x = 160$ och $x = 170$. Vi kan använda `predict()` funktionens argument `newdata` som är en dataframe med samma variabelnamn som vi använde när vi anpassade modellen (`height` i vårt fall). Följande kod skapar dataframen i en variabel vi väljer att kalla `new_x` och predikterar de genomsnittliga forcerade utandningsvolymerna för de nya x värden ovan.

```
new_x <- data.frame(height = c(160, 170))
predict(lm_fev_vs_height, newdata = new_x)
```

```
      1      2
2.867951 3.406038
```

Exempelvis ser vi att ett barn (eller en ungdom) på 170 cm har i genomsnitt ett `fev`-värde på ca 3.4.

Uppgift 4.1

Skriv manuellt upp den skattade regressionsekvationen (räta linjen vi tagit fram) och räkna ut, i R, eller med papper och penna, den predikterade volymen för individer som är 160 cm långa. Verifiera att ditt svar stämmer med svaret ovan.

Uppgift 4.2

Prediktera, ”manuellt” respektive med `predict()`-kommandot, utandningsvolymen för en individ av längd 180 cm. För att prediktera med `predict` måste du, liknande ovan, skapa en ny data frame, och det går bra att bara ta med ett värde (och då behövs ingen vektor (men det blir inte fel om du skapar en vektor med enbart ett tal)).

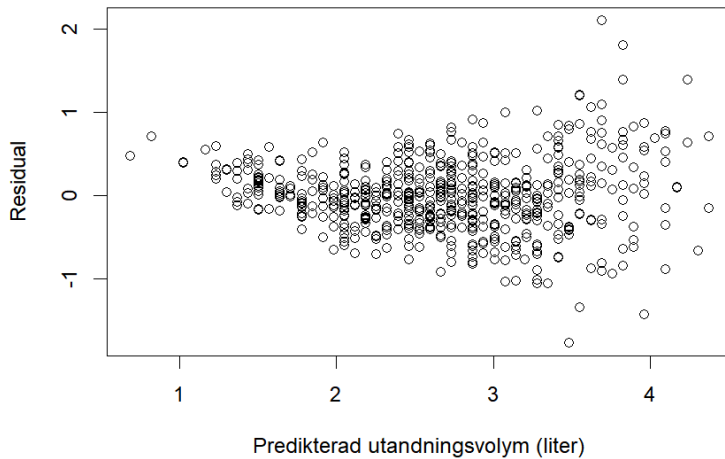
Uppgift 4.3

Förklara i ord vad R^2 -värdet innebär (se F9, F10)

Uppgift 4.4

Nedanstående figur visar residualerna för den modell vi precis skattat. På y-axeln har vi värdet på residualerna och på x-axeln våra predikterade värden (på samma sätt som i boken s. 110-111). Diskutera mönster i residualerna och försök relatera dessa till hur spridningsdiagrammet ovan, med regressionslinjen inritad, ser ut.

Residualer vs. predikterade värden



Uppgift 4.5

Tänk dig att du kom in i analysen med hypotesen att det inte finns något samband mellan längd och utandningsvolym i populationen. Ställ upp en nollhypotes och en alternativhypotes. Formulera dessa i ord respektive med den notation vi gick igenom på F9, sid 21-23, 25. Kan du förkasta din nollhypotes?

Uppgift 4.6

Ta fram ett 95%-igt konfidensintervall för skattningen av β_1 , dvs skattningen av förhållandet mellan längd och utandningsvolym i populationen. För ett 95%-igt konfidensintervall kan du använda ett t-värde=2 i följande formel, som vi såg i F9:



Confidence intervals for coefficients.

Confidence intervals for model coefficients (e.g., the intercept or the slope) can be computed using the t -distribution:

$$b_i \pm t_{df}^* \times SE_{b_i}$$

where t_{df}^* is the appropriate t^* cutoff corresponding to the confidence level with the model's degrees of freedom, $df = n - 2$.

Övriga värden du behöver finns i vår regressionstabell.

För att ta fram ett 95%-igt konfidensintervall för regressionskoefficientskattningarna kan du också använda följande kommando:

```
confint(lm_fev_vs_height, level=0.95)
```

Uppgift 4.7

Bekräfta att du får samma (eller väldigt liknande) svar med `confint()`-kommandot som du fick i 4.6.

5. Multipel linjär regression

Vi gör nu en multipel linjär regression, där vi till ovanstående regression lägger till dummyvariabeln för om en person är rökare eller inte:

```
Console Terminal Background Jobs
R 4.4.1 · C:/test/SOK/
> lm_fev_vs_height_smoking <- lm(fev ~ height+smoking, data = FevChildren)
> summary(lm_fev_vs_height_smoking)

Call:
lm(formula = fev ~ height + smoking, data = FevChildren)

Residuals:
    Min       1Q   Median       3Q      Max
-1.76487 -0.25513  0.00027  0.23445  2.09853

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.763158   0.216872  -26.574  <2e-16 ***
height       0.053963   0.001388   38.865  <2e-16 ***
smoking1    -0.025260   0.060666   -0.416   0.677
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.433 on 603 degrees of freedom
Multiple R-squared:  0.7285,    Adjusted R-squared:  0.7276
F-statistic: 808.9 on 2 and 603 DF,  p-value: < 2.2e-16

> |
```

För att få denna regression la vi alltså till ytterligare en x-variabel, med ett plustecken och sen namnet på variabeln. Vi skapade ett nytt regressionsobjekt och skrev ut regressionsresultaten med summary-kommandot.

Uppgift 5.1

Diskutera kort om något hänt med förklaringsgrad eller skattad lutningskoefficient för längdvariabeln.

Uppgift 5.2

Skriv ner ekvationen för den skattade regressionen.

Uppgift 5.3

Tolka koefficienterna för skattningarna på height respektive smoking. OBS – vi har multipel linjär regression – vad måste vi tänka på? (F10 och boken kapitel 8).

Uppgift 5.4

Tolka något av p-värdena i regressionstabellen.

6. Datasetet gapm

I inlämningsuppgiften, del 4, kommer vi använda datasetet gapm med länder och sex variabler från Gapminder¹. Vi har dessutom lagt till vår variabel "landlocked" från tidigare i kursen. Du kan ladda ner data [här](#) (högerklicka), eller från Githubsidan (labb 4) eller från Datafiler i Athena. Data är från 2022. Variablerna som finns i datasetet är²:

country – de länder som finns i Gapminderdata och för vilka det finns kompletta data

child_mort – antal barn som dör före fem års ålder, per 1000 barn födda

fertility – förväntat antal barn per kvinna

co2_cap – antal ton koldioxid som varje individ "konsumerar"

gdp_cap – BNP per capita i dollar (köpkraftsjusterat)

life_exp – förväntad medellivslängd

landlocked – indikator för om ett land har kust eller inte (1=har inte kust)

¹ Based on free material from GAPMINDER.ORG, CC-BY LICENSE.

² Mer exakta definitioner av vissa av variablerna finns på Gapminders hemsida men är inte viktiga för uppgiften.

Efter att ha läst in datasetet kan man bla. se att vi har 192 länder, något färre än i vårt tidigare dataset med landlocked. Några länder hade inte kompletta data och togs bort.

Om man tittar på `summary()` av data ser man att data kan anges på formen $9.990e+03$ (vetenskaplig notation), vilket betyder 9,99 gånger 10 upphöjt till tre, dvs 9,99 gånger 1000, dvs 9990. Det minst befolkade landet i datasetet har så många invånare. På samma sätt betyder $1e+06$ en miljon, osv.

Det kan vara intressant att titta på data med `summary()`, göra histogram för varje numerisk variabel, plotta spridningsdiagram, ta fram fördelningar betingade på `landlocked`, etc., för att få en känsla för hur data och eventuella samband ser ut. Exempelvis ser man att det är mycket stora skillnader mellan lägsta och högsta värde på `c02_cap` och `gdp_cap`.

Description of Shortcuts and Special Characters in R

Description	Windows/Linux	Mac
Run code	Ctrl + Enter	⌘ + Enter
Run code and stay on current line	Alt + Enter	Option + Enter
Select All	Ctrl + A	⌘ + A
Copy	Ctrl + C	⌘ + C
Paste	Ctrl + V	⌘ + V
Cut selected text	Ctrl + X	⌘ + X
Delete line	Ctrl + D	⌘ + D
Undo	Ctrl + Z	⌘ + Z
Redo	Ctrl + Shift + Z	⌘ + Shift + Z
Select multiple lines/area	Shift + arrow key	Shift + arrow key
Duplicate line/area	Ctrl + Shift + D	⌘ + Shift + D
Scroll up/down	Ctrl + up/down arrow key	⌘ + up/down arrow key
Go to the top	Ctrl + Home	⌘ + Home
Go to the bottom	Ctrl + End	⌘ + End
Go to line	Shift + Alt + G	⌘ + Shift + Option + G
Save	Ctrl + S	⌘ + S
Save all documents	Ctrl + Alt + S	⌘ + Option + S
Open document	Ctrl + O	⌘ + O
\$ symbol	Ctrl + Alt + 4	Option + 4
Vertical bar/pipe (logical OR):	Ctrl + Alt + < or Alt gr + <	Option + 7
Assignment (<-)	Alt + -	Option + -
Zoom out/in	Ctrl + -/+	⌘ + -/+
Interrupt running code	Esc	Esc
Clear Console	Ctrl + L	⌘ + Option + L
Left square bracket [Ctrl + Alt + 8 or Alt gr + 8	Option + [or Option + 8
Right square bracket]	Ctrl + Alt + 9 or Alt gr + 9	Option +] or Option + 9
Left curly brace {	Ctrl + Alt + 7 or Alt gr + 7	Option + Shift + 8
Right curly brace }	Ctrl + Alt + 0 or Alt gr + 0	Option + Shift + 9
Slash /	Shift + 7	Shift + 7
Backslash \	Alt gr + + or Ctrl + Alt + +	Option + Shift + 7
Tilde symbol ~	Alt gr + ~ (key near Å)	Option + ~ (key near Å)
Search expression	Ctrl + F	⌘ + F
Arrange indentation	Ctrl + I	⌘ + I
Show list of common shortcuts	Shift + Alt + K	Option + Shift + K

Denna version av dokumentet: 260413

Materialet i Statistisk översikt kurs har tagits fram av Ulf Högnäs och Anders Fredriksson, med inspiration och ibland direkt användande av material från andra kurser och personer, bland annat kurserna Statistik och dataanalys 1-3, med material av Michael Carlson, Ellinor Fackle Fornius, Jessica Franzén, Oskar Gustafsson, Oscar Oelrich, Mona Sfaxi, Karl Sigfrid, Mattias Villani, Valentin Zulj, med flera.