

Föreläsning 2: Kort introduktion till R

Matias Quiroz¹

¹Statistiska institutionen, Stockholms universitet

VT 2023

- ▶ I tidernas begynnelse.
- ▶ Vad är R?
- ▶ Varför R?
- ▶ R in action.

Tillbaka till programmeringens stenålder

- ▶ En gång i tiden var `Fortran` det ledande programmeringspråket för numeriska beräkningar.
- ▶ `Fortran` skapades av IBM (lett av John Backus) 1957.
- ▶ `Fortran` har väldigt hög beräkningsprestanda och används än idag (i mycket mindre utsträckning) inom high-performance computing.
- ▶ Inom statistik så växte utforskande dataanalys, **exploratory data analysis** (EDA) på engelska, fram starkt på 70-talet.
- ▶ EDA utför informell analys av data, ofta via grafiska verktyg. John Tukey var en stor förespråkare.
- ▶ EDA tillämpades flitigt av statistiker såväl av folk utanför de matematiska vetenskaperna.
- ▶ EDA möjliggörs av tillgång till en dator och att kunna ge datorn instruktioner (programmera).

Tillbaka till programmeringens stenålder, forts.

- ▶ Enklaste exemplet på EDA är en plot mellan två variabler y och x .
- ▶ Exempel: $y = (0.5, 1.2, 0.9, 2.1)$ och $x = (1, 2, 3, 4)$.

```
program scatter_plot
  use gnuplot
  implicit none

  real, dimension(4) :: x, y
  y=(/0.5, 1.2, 0.9, 2.1/)
  x=(/1,2,3,4/)

  call gnuplot_start()
  call gnuplot_plot_xy(x,y,"Scatter Plot")
  call gnuplot_close()
end program scatter_plot
```

Figure 1: Fortran kod för enkel EDA.

- ▶ Man behövde vara programmeringsnörd för att göra enkla analyser.

Tillbaka till programmeringens stenålder, forts.

- Hatten av för Fortran som ändå får anses vara enkelt.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <gnuplot_c.h>

int main()
{
    double x[4] = {1, 2, 3, 4};
    double y[4] = {0.5, 1.2, 0.9, 2.1};
    gnuplot_ctrl *h;

    h = gnuplot_init();
    gnuplot_setstyle(h, "points");
    gnuplot_set_xlabel(h, "X");
    gnuplot_set_ylabel(h, "Y");
    gnuplot_plot_xy(h, x, y, 4, "Scatter Plot");
    gnuplot_close(h);

    return 0;
}
```

Figure 2: C kod för enkel EDA.

R förändrar spelplanen

- ▶ Motiverat av att kunna göra EDA enkelt, utvecklade Bell Laboratories ett statistiskt programmeringsspråk vid namn *S* år 1976.
- ▶ En kommersialiserad implementation av *S* som gick vid namnet *S-PLUS* utvecklades av Statistical Sciences, Inc år 1988.
- ▶ En implementation av *S* som gick vid namnet *R* utvecklades år 1993 av Ross Ihaka och Robert Gentleman för undervisning i University of Auckland.
- ▶ *R* blev en del av GNU projektet¹ 1997.
- ▶ Öppen källkod: `cran.r-project.org/`.
- ▶ The **Comprehensive R Archive Network** (CRAN) är *Rs* centrala kodförråd. År 2020 fanns det mer 16000 paket. Vem som helst kan bidra.
- ▶ *RStudio* är en integrerad utvecklingsmiljö för *R* som är gratis och enkel att använda.

¹gnu.org: GNU is an operating system that is free software—that is, it respects users' freedom

- ▶ Behöver inte vara programmeringsnörd för att göra enkel EDA.

```
y <- c(0.5, 1.2, 0.9, 2.1)
x <- c(1, 2, 3, 4)
plot(x, y)
```

- ▶ Många avancerade statistiska metoder finns implementerade i R-paket.
- ▶ Att använda R-paket kräver minimalt med programmeringskunskaper.
- ▶ Föreläsning 8 och 9 tillägnar jag ca 4 timmar på att förklara regression och prediktion.
- ▶ Tillämpning med hjälp av R-paket `lm`:

```
model <- lm(y ~ x)
predict(model)
```

- ▶ Trots sin enkelhet, kan R uppfattas som svårt, speciellt i början.
- ▶ Datorlaborationerna och inlämningsuppgiften kommer att bevisa min tes.
- ▶ När frustrationen tar över, meditera en stund på hur livet hade sett ut om R inte fanns och datorlabbarna var i `Fortran` eller `C`.
- ▶ Om det inte hjälper, ponera istället att datorlabbarna var i `Assembly` (ett programmeringsspråk på lågnivå, en nivå över maskinkod).

R förändrar spelplanen, forts.

- Program för att räkna ut en median av fyra värden.

```
section .data
y dd 0.5, 1.2, 0.9, 2.1

section .text
    global _start

_start:
    mov eax, [y]
    add eax, 4
    mov ebx, [eax]
    cmp ebx, [y]
    jg _swap
    add eax, 4
    mov ebx, [eax]
    cmp ebx, [y]
    jg _swap
    jmp _end

_swap:
    mov ecx, [y]
    mov [y], ebx
    mov [eax], ecx
    jmp _start
```

Figure 3: Assembly kod för median, Del 1.

► Inte slut än:

```
_end:
    mov ecx, [y]
    add ecx, 4
    mov edx, [ecx]
    cmp edx, [y]
    jg _result
    mov edx, [y]

_result:
    mov eax, edx
    mov [y], eax
    mov ebx, 0
    mov eax, 1
    int 0x80
```

Figure 4: Assembly kod för median, Del 2.

► Programkod som åstadkommer samma sak i R:

```
y <- c(0.5, 1.2, 0.9, 2.1)
median(y)
```

R förändrar spelplanen, forts.

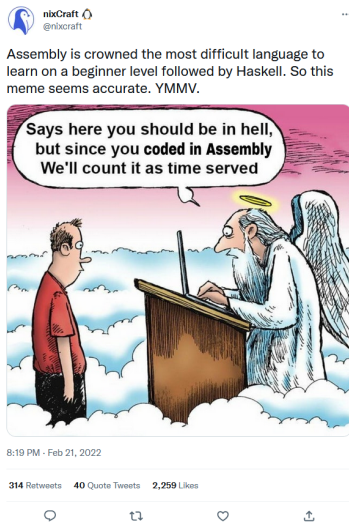


Figure 5: Från nixcrafts Twitter profil.

R förändrar spelplanen, forts.



Figure 6: Från rstatsmemes Twitter profil.

1. ChatGPT vet bäst:



Which programming languages should a statistician learn?



A statistician should be proficient in at least one programming language that is commonly used for statistical analysis and data manipulation, such as:



1. R: R is a popular open-source programming language that is widely used in the field of statistics. It has a large community of users and developers and it offers a wide range of libraries and packages for data analysis, visualization, and modeling.

Figure 7: Min konversation med ChatGPT 12 januari 2023.

2. Python och SAS kom på andra respektive tredje plats.

1. ChatGPT har inte förstått att Data Science och statistik är samma sak 😊:



Which programming languages should a data scientist learn?



A data scientist should have a strong foundation in at least one programming language, such as Python or R, as they are commonly used for data analysis and machine learning tasks. Additionally, some other languages that are also useful for data scientists to learn are:

1. Python: Python is a versatile programming language that is widely used for data science and machine learning tasks. It has a large number of libraries and frameworks for data manipulation, analysis, and visualization, such as NumPy, pandas, and Matplotlib.
2. R: R is a programming language that is specifically designed for statistical computing and data visualization. It has a wide range of libraries, such as ggplot2 and dplyr, that are useful for data manipulation and visualization.

Figure 8: Min konversation med ChatGPT 12 januari 2023.

Varför R?, forts.

1. Första träffen på Google-sökningen “10 reasons for learning R”:

Top 10 Reasons To Learn R Programming

You might have come across the term “R programming” quite a few times and numerous questions would have popped into your head. Is it just another programming language? How is it different from other programming languages? What is the future of it? How much will I earn if I know this language? All your queries will be answered in this blog, which gives you the top 10 reasons to learn R Programming.

These are the reasons, which make R Programming such an in-demand skill:

1. R Programming gets you High Paying Jobs
2. R is Lingua Franca for Statistical Analysis and Data Science
3. R is used by Top Companies
4. R is used to Create Interactive Web-Apps
5. R is used to Create Stunning Visualizations
6. R provides a Comprehensive Library
7. R has Huge Community
8. R is used by Multiple Domains and Industries
9. R is Cross-Platform Compatible
10. R is Open Source

Figure 9: Från www.edureka.co.

Varför R?, forts.

- ▶ Låt oss sammanfatta.
- ▶ R är ett (förhållandevis) enkelt programmeringsspråk.
- ▶ R är gratis och open source.
- ▶ R är mycket användbart.
- ▶ R ger (välbetalda!) jobb.
- ▶ När (eventuell) frustration lägger sig, är det kul att jobba med R!
- ▶ Roger Peng har gjort en mycket mer ingående YouTubevideo om Rs historia.

- ▶ Datorlaboration 1 ger en introduktion till R syntax.
- ▶ De övriga datorlaborationerna bygger på era R kunskaper.
- ▶ Syftet med denna föreläsning är inte att lära ut R syntax.
- ▶ R syntax lärs bäst i lugn och ro **när ni själva programmerar** (med hjälp av instruktionerna i datorlabbarna).
- ▶ Syftet med den här föreläsningen är att visa några exempel på analyser i R.
- ▶ Koden (`R_lecture_examples.R`) och dataseten vi går igenom finns på kurshemsidan under Föreläsning 2.

- ▶ Filen `SmartPhones.RData` innehåller data från en undersökning med 2500 personer².
- ▶ Undersökningen frågade vilken smartphone man föredrog. Man registrerade också åldersgrupp.
- ▶ Beror smartphonepreferens på åldersgrupp?

²Källa: [statista.com](https://www.statista.com). Två saknade värden borttagna.

- ▶ Datasetet `diamonds` från R-paketet `ggplot2` innehåller data för ca 54000 diamanter.
- ▶ Några variabler i datasetet:
 - ▶ Pris.
 - ▶ Vikt (enhet carat. 1 carat är 0.1 gram).
 - ▶ Färg.
 - ▶ Klarhetsmått.
 - ▶ Längd.
- ▶ Vad finns det för samband mellan pris och de andra variablerna?

Riskvärdering av tillgångar

- Vårt sista exempel är lite mer komplicerat, men något ni skulle kunna få som uppgift i arbetslivet.
- En finansiell tillgång, t.ex en Tesla aktie, har en avkastning (return på engelska) i period t som definieras som

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

där S_t är aktiepriset vid tidpunkten t .

- Om t är på månadsfrekvens får vi månadsavkastning.
- Exempel: Om tillgången var värderad till 105 vid $t - 1$ och 110 vid t ,

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

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

Risikvärdering av tillgångar, forts.

- ▶ Hur vet man om en tillgång riskfylld? Detta utvärderas i förhållande till marknaden.
- ▶ En metod för riskvärdering är **capital asset pricing model** (CAPM).
- ▶ CAPM studerar tillgångens “excess return” i förhållande till en riskfri tillgång (t.ex en statsobligation) och jämför mot densamma för marknadsportföljen.
- ▶ Tillgångens “excess return” i förhållande till en riskfri tillgång

$$y_t = r_t - r_t^{(f)},$$

där $r_t^{(f)}$ är avkastningen för den riskfria tillgången.

- ▶ Marknadsportföljen “excess return” i förhållande till den riskfria tillgången

$$x_t = r_t^{(m)} - r_t^{(f)},$$

där $r_t^{(m)}$ är avkastningen för marknadsportföljen.

Riskvärdering av tillgångar, forts.

- ▶ CAPM antar ett linjärt förhållande mellan y_t (excess return tillgång) och x_t (excess return marknad) med slumpmässiga avvikelser,

$$y_t = \alpha + \beta x_t + \text{en slump term.}$$

- ▶ **Parametrarna** α och β kan estimeras med hjälp av data.
- ▶ α fångar excess return i tillgången (y_t) som inte förklaras av marknadens excess return (x_t).
- ▶ β mäter **tillgångens känslighet för förändringar i marknadens excess return**.
 - ▶ $\beta > 1$ **amplifierar förändringar** i marknadens excess return.
Exempel: Om $x_t \uparrow 1\% \implies y_t \uparrow > 1\%$.
 - ▶ $\beta < 1$ **dämpar förändringar** i marknadens excess return.
Exempel: Om $x_t \uparrow 1\% \implies y_t \uparrow < 1\%$.
- ▶ En tillgång med $\beta > 1$ är mycket riskfylld.
- ▶ Om vi har tio tillgångar (med varsitt β) kan vi rangordna deras risk (ju större β desto större risk).

- ▶ Filen `CAPM_data.RData` innehåller finansiella data från USA³.
- ▶ 10 år av data, från januari 1978 till december 1987. 120 tidsobservationer.
- ▶ Vi analyserar IBM och TANDY aktierna.
- ▶ Ett exempel på **linjär regression** (Föreläsning 8 och 9).

³Skapad från rådata tillgänglig på github.com/BenjaminKay/berndt-econometrics.