INTRODUCERE ÎN ȘTIINȚA DATELOR

Suport de curs pentru studenți

Semestrul iarnă, 2021

Cuprins

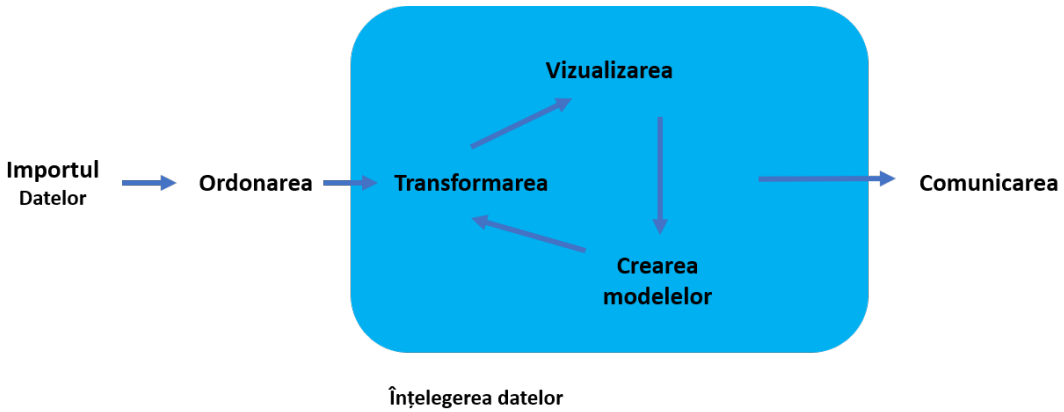
1. Introducere

Știința Datelor este disciplina care permite să transformăm datele brute în noi cunoștințe. Scopul acestui curs este să învățați cele mai importante ***instrumente R*** care vă vor permite să le aplicați în Știința Datelor. După efectuarea acestui curs veți putea manipula instrumentele necesare pentru a aborda o mare varietate de provocări din domeniul Științei Datelor, folosind cele mai bune părți a limbajului R.

1.1. Ce vom învăța?

Știința Datelor sau ***Data Science*** este un domeniu foarte larg și nu există nici o modalitate de a-l stăpâni citind doar o singură carte sau făcând un singur curs. Scopul acestui curs este de a oferi studenților o bază solidă în cele mai importante instrumente în R. În cadrul acestui curs,

modelul de instrumente necesare într-un proiect tipic de Știință a Datelor arată în felul următor:



Mai întâi trebuie să ***importăm*** datele în R. Acest lucru înseamnă că trebuie să luăm datele stocate într-un fișier, bază de date sau API web și să le încărcăm într-un cadru de date R.

După importul datelor, urmează etapa de ***ordonare*** *și* ***curățarea datelor*****.** Ordonarea datelor înseamnă stocarea datelor într-o formă consistentă, care se potrivește cu semantica setului de date, cu modul în care sunt stocate și scopul pe care îl urmați. Pe scurt, odată ce datele sunt ordonate, fiecare coloană este o variabilă și fiecare rând este o observație. Ordonarea datelor este o etapă importantă, deoarece structura consecventă vă permite să vă concentrați asupra întrebărilor privind setul de date, nu să pierdem timp pentru a obține date în forme potrivite pentru diferite funcții.

Odată ce am obținut date ordonate, următorul pas este ***transformarea*** acestora. Transformarea include restrângerea la observațiile de interes (cum ar fi toți studenții de la o singură facultate sau toate tranzacțiile dintr-un an), crearea de variabile noi în baza celor brute inițiale (cum ar fi calcularea vitezei din datele de distanță și timp) și obținerea unui set de statistici sumare (cum ar fi sumele și mediile).

Odată ce am obținut date ordonate, cu variabile de care avem nevoie, există două motoare principale de generare a cunoștințelor: vizualizarea și modelarea. Acestea au puncte forte și slabe complementare, astfel încât orice analiză reală va itera între ele de mai multe ori.

Vizualizarea este o activitate și proprietate umană fundamentală. O modalitate bună de vizualizare vă va arăta noi lucruri privind datele și cu siguranță va ridica noi întrebări despre date. O vizualizare corespunzătoare indică dacă punem greșit întrebările despre datele corespunzătoare sau avem nevoie să colectăm noi date. Vizualizările ne pot surprinde și acestea trebuie scalate corespunzător, deoarece vor fi interpretate de oameni.

Modelele sunt instrumente complementare vizualizărilor. După ce am formulat întrebări suficient de precise, putem utiliza un model pentru a răspunde la întrebări. Modele sunt instrumente fundamentale matematice sau statistice. Fiecare model face presupuneri și, prin natura lor, modelele nu pot pune la îndoială propriile ipoteze. Asta înseamnă că un model nu ne poate surprinde în mod fundamental.

Ultima etapă din știința datelor este ***comunicarea***, o parte critică a oricărui proiect de analiză a datelor. Nu contează cât de bune sunt modelele sau vizualizările pe care le-ați obținut, dacă nu puteți comunica rezultatele și altora.

Cadrul general care cuprinde toate acestea relatate mai sus este ***programarea***. Programarea este un instrument pe care îl utilizăm la fiecare etapă a proiectului. Nu este nevoi să fii un programator expert pentru fi un ***Data Scientist***, însă merită să afli mai multe despre programare, deoarece devenind un programator mai bun, poți să automatizezi sarcinile comune și să rezolvi probleme noi cu ușurință.

1.2. Cum este organizat cursul?

Descrierea anterioară a etapelor în analizele din cadrul Științei Datelor este organizată aproximativ în ordinea în care le vom utiliza într-o analiză. ***Din experiență, aceasta nu este cel mai bun mod de a învăța Data Science.***

- A începe cu curățarea și ordonarea datelor nu este foarte eficient, deoarece 80% din timp este o rutină plictisitoare, iar celelalte 20% din timp este bizară și frustrantă. Aceasta nu este o idee bună să începem de aici un subiect nou! Vom începe cu vizualizarea și ordonarea datelor deja curățate și importate. În acest fel, atunci când veți dori să ordonați și curățați propriile date, motivația va rămâne ridicată, deoarece veți ști că „durerea de cap" se merită.
- Unele subiecte sunt mai bine explicate și înțelese prin anumite instrumente. De exemplul, este mai ușor de înțeles cum funcționează modelele dacă știm deja despre vizualizarea, ordonarea, curățarea datelor și programarea în R.
- Instrumentele de programare nu sunt neapărat interesante în sine, însă vă permit să abordați probleme mult mai provocatoare. Aici vom oferi o colecție de instrumente de programare, și vom vedea cum programarea poate fi combinată cu Știința Datelor pentru a aborda probleme interesante de modelare.

În cadrul fiecărei secțiuni vom repeta un patern similar, vom începe cu câteva exemple motivante, astfel încât să vă permită să înțelegeți imaginea generală, apoi să vă concentrați pe detalii. Fiecare secțiune a cursului conține exerciții pentru a vă ajuta să exersați ceea ce învățați. Deși este tentant să omiteți exercițiile, nu există o modalitate mai bună de a învăța decât să practicați probleme reale.

1.3. În acest curs

Există câteva subiecte importante pe care acest curs nu le acoperă. Cu toate acestea, este important să rămâneți concentrați asupra elementelor esențiale, oferite de acest curs, astfel încât să puteți construi o bază de deprinderi.

1.3.1. Big data

În acest curs vom lucra cu seturi de date mici. Acestea sunt potrivite pentru a începe, deoarece nu puteți aborda datele mari decât dacă faceți experiență cu date mici. Instrumentele pe care le

vom învăța în acest curs permit gestionarea cu ușurință a sute de mega biți de date și, cu puțină grijă, putem folosi de obicei pentru a lucra cu 1-2 Gb de date. Dacă veți avea nevoie să lucrați cu date mari (10 - 100 Gb), ar trebui să aflați mai multe despre `data.table`.

Dacă datele utilizate sunt mari, luați în considerație cu atenție dacă problemele de date mari ar putea fi de fapt o mică problemă de date deghizate. Deși datele brute complete pot fi mari, de multe ori datele necesare pentru a răspunde la întrebări pot fi destul de mici. S-ar putea să găsiți un subset, un sub-eșantion sau un sumar de date care se potrivesc și care vă permit să răspundeți la întrebările care vă interesează. Provocare pe care v-o propunem este găsirea datelor mici potrivite, care deseori necesită iterații.

O altă posibilitate este că problema datelor mari este de fapt un număr mare de probleme mici. Fiecare problemă individuală s-ar putea încadra în memoria calculatorului, și ați putea să le tratați individual din setul de date. Ar fi simplu dacă ar fi 10 sau 100 de oameni, dar dacă avem un milion? Din fericire, fiecare problemă este independentă de celelalte, deci avem nevoie de un sistem de tipul **Handoop** sau **Spark** care ne-ar permite să trimitem seturi de date diferite la diferite calculatoare pentru procesare. După ce am aflat răspunsul cum răspundem la întrebări pentru un singur sub-set folosind inistrumentele descrise în acest curs, veți continua cu noi instrumente, precum ***sparklyr***, ***rhipe*** și ***ddr*** pentru a rezolva probleme cu seturi de date complete.

1.3.2. Alte limbaje în Data Science

În acest curs nu veți afla nimic despre Python, Julia sau orice alt limbaj de programare util pentru știința datelor. Aceasta nu înseamnă că credem că alte limbaje sunt mai rele sau mai bune. În practică, majoritatea echipelor de ***Data Science*** folosesc un amestec de limbaje de programare, în dependență de date și scopurile propuse, adesea cel puțin R și Python.

Cu toate acestea, credem cu tărie că cel mai bine este să stăpâniți câte un instrument la fiecare etapă. Vă veți îmbunătăți mai repede abilitățile și cunoștințele odată ce vă veți focusa pe un limbaj, decât să vă răspândiți superficial pe mai multe subiecte. Este bine să învățați lucruri noi de-a lungul carierei voastre, dar înainte de asta, asigurați-vă de o înțelegere solidă înainte de a trece la următorul lucru interesant.

Credem că R este o destinație minunată pentru a începe călătoria voastră în domeniu ***Data Science***, deoarece este un mediu conceput de la bază pentru a sprijini științele datelor. R nu este doar un limbaj de programare, ci și un mediu interactiv pentru a face ***Data Science***. Pentru a susține interacțiunea, R este conceput ca un limbaj mult mai flexibil față de alte limbaje. Această flexibilitate vine cu dezavantajele sale, însă marele avantaj este ușurința evolutivă a gramaticilor adaptate pentru anumite părți ale procesului de ***Data Science***. Aceste mini-limbaje vă ajută să vă gândiți la probleme ca un om de știință, susținând în același timp interacțiunea fluentă dintre creier și calculator.

1.3.3. Date ne-rectangulare

Acest curs se bazează exclusiv pe date rectangulare: colecții de valori care sunt asociate cu o variabilă și o observație. Există o mulțime de seturi de date care nu se încadrează în mod natural în această paradigmă: inclusiv imagini, sunete, copaci, text. Însă cadrele de date rectangulare sunt extrem de frecvente în știință, economie și industrie și credem că sunt un loc minunat pentru a începe călătoria în domeniul științei datelor.

1.3.4. Confirmarea ipotezelor

Putem împărți analiza datelor în două tabere: generarea de ipoteze și confirmare de ipoteze (analiza de confirmare). Acest curs se bazează fără îndoială pe generarea de ipoteze sau exploatarea datelor. Aici veți privi profund datele și, în combinație cu cunoștințele voastre despre subiect, veți genera multe ipoteze interesante pentru a explica de ce datele se comportă așa cum se comportă. Evaluați ipotezele în mod informal, folosindu-vă scepticismul pentru a contesta datele în mai multe moduri.

Complementul generării de ipoteze este confirmarea ipotezei. Confirmarea ipotezei este dificilă din două motive:

1. Aveți nevoie de un model matematic precis pentru a genera predicții. Acest lucru necesită adesea o sofisticare statistică considerabilă;

2. Puteți utiliza o observație, o singură dată pentru a confirma o ipoteză. De îndată ce o faceți de mai multe ori, reveniți la analize exploratorii. Astfel, pentru confirmarea ipoteze, trebuie să vă „înregistrați” (să scrieți în prealabil) planul de analiză și să nu vă abateți de la acesta chiar și atunci când ați văzut sau ați primit datele. Se va vorbi puțin despre câteva strategii pe care le puteți utiliza pentru a face acest lucru mai ușor în secțiune despre modelare.

Este obișnuit să ne gândim la modelare ca instrument de confirmare a ipotezelor și vizualizare ca instrument de generare a ipotezelor, însă aceasta este o dihotomă falsă: modelele sunt adesea folosite pentru explorare și, cu puțină grijă, puteți utiliza vizualizarea pentru confirmare. Diferența este cât de des evaluezi sau privești fiecare observație: dacă le privești o singură dată, este confirmare, dacă de mai multe ori, este o explorare.

1.4. Cerințe

Facem câteva presupuneri pentru a putea profita la maxim de acest curs. În general, ar trebui să fiți alfabetizați numeric și este foarte util dacă deja aveți o experiență de programare.

Există patru lucruri de care aveți nevoie pentru a rula codurile din acest curs: ***R***, ***RStudio***, o colecție de pachete R numite în continuare ***tidyverse*** și câteva altele. Pachetele sunt unitățile fundamentale ale codului R reproductibil. Aceste includ funcții reutilizabile, documentația care descrie modul de utilizare a acestora și mostre de date.

1.4.1. R

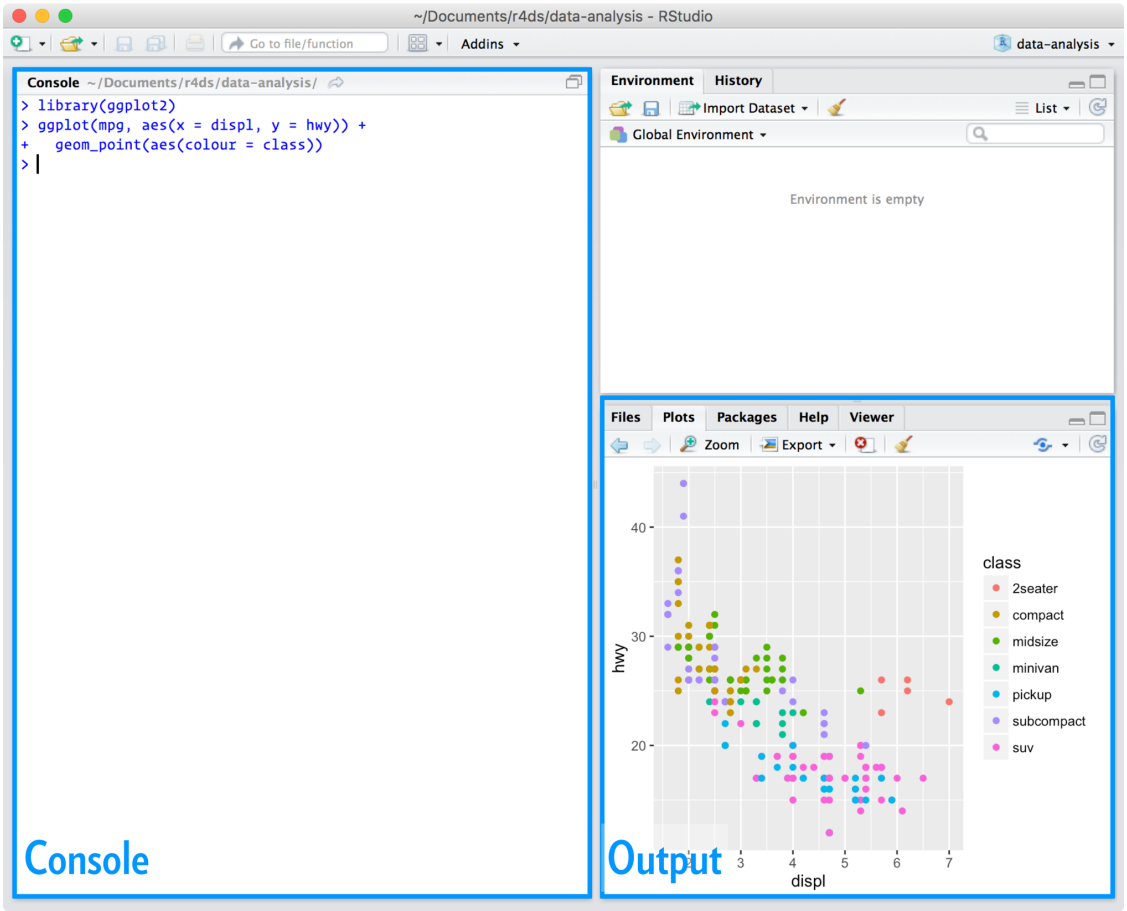
Pentru a descărca R, accesați CRAN, rețeaua completă de arhive R. CRAN este compus dintr-un set de servere oglindă distribuite în întreaga lume și este utilizat pentru distribuirea pachetelor R și a R-ului. Nu încercați să alegeți o oglindă care vă este apropiată - folosiți o oglindă cloud, *https://coud.r-project.org*, care o calculează automat pentru voi.

O nouă versiune majoră R apare o dată pe an și există câte 2-3 versiuni minore în fiecare an. Este o idee bună să actualizați în mod regulat. Actualizare poate fi puțin dificilă, în special pentru versiunile majore, care necesită reinstalarea tuturor pachetelor, însă amânarea nu face decât înrăutățirea situației.

1.4.2. RStudio

RStudio este un IDE pentru programarea în R. Descărcați și instalați Rstudio de pe *http://www.rstudio.com/download*. RStudio este actualizat de câteva ori pe an. Când este disponibilă o nouă versiune, RStudio vă va anunța. Este o idee bună să faceți upgrade într-un mod regulat, astfel încât să puteți profita la maxim de cele mai recente și mia bune funcții.

Când porniți RStudio, veți vedea două câmpuri cheie în interfață:



Pentru moment, totul ce trebuie să știți este că tastați codurile R în panoul consolei și apăsați ***Enter*** pentru a le rula. Veți afla mai multe pe măsură ce mergem mai departe.

1.4.3. tydiverse

De asemenea, va trebui să instalați câteva pachete R. Un pachet R este o colecție de funcții, date și documente care extind capacitățile bazei R. Utilizarea pachetelor este esențială pentru utilizarea cu succes a R-ului. Colecția de pachete ***tidyverse*** împărtășesc o filozofie comună a modelării datelor și sunt concepute pentru a lucra în mod natural împreună.

Puteți instala versiunea completă cu o singură linie de cod:

```
install.packages("tidyverse")
```

La propriul PC tastați această linie de cod în consolă, apoi apăsați Enter pentru a o rula. R va descărca pachetele din CRAN și le va instala pe calculator. Dacă apar probleme de instalare, asigurați-vă că aveți conexiune la internet și că *https://cloud.r-project.org/* nu este blocat de firewall sau Proxy.

Nu veți pute utiliza funcțiile, obiectele și fișierele de ajutor dintr-un pachet până nu îl încărcați în bibliotecă cu funcția *`library()`*. După ce ați instalat un pachet, îl puteți încărca prin funcția *`library ()`*:

```
library(tidyverse)

## -- Attaching packages ----------------------------------------------- tid
yverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts -------------------------------------------------- tidyverse
_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Aceste lucruri vă spun că **tidyverse** încarcă pachetele **ggplot2**, **tibble**, **readr** și **dplyr**. Acestea reprezintă nucleul pentru **tidyverse**, deoarece le veți folosi aproape în fiecare analiză.

Versiunile pachetelor din **tidyverse** se schimbă destul de frecvent. Puteți vedea dacă sunt disponibile actualizări și să le instalați opțional, rulând *`tidyverse_update()`*.

1.4.4. Alte pachete

Există multe alte pachete excelente care nu fac parte din **tidyverse**, care rezolvă probleme din alte domenii sau sunt proiectate cu un set diferit de principii subiacente. Acest lucru nu le face mai bune sau mai rele, ci diferite. Pe măsură ce veți aborda mai multe proiecte noi în ***Data Science*** prin R, veți învăța pachete noi și noi moduri de gândire despre date.

În acest curs vom folosi trei pachete de date din exteriorul **tidyverse**:

```
install.packages(c("nycflights13", "gapminder", "Lahman"))
```

Aceste pachete oferă date despre zborurile companiilor aeriene, dezvoltarea mondială și baseball pe care le vom folosi pentru a ilustra ideile în domeniul ***Data Science***.

1.5. Executarea codului R

```
1 + 2
#> [1] 3
#> [1] 3
```

Dacă rulați acest cod în consola locală, aceasta va arăta în felul următor:

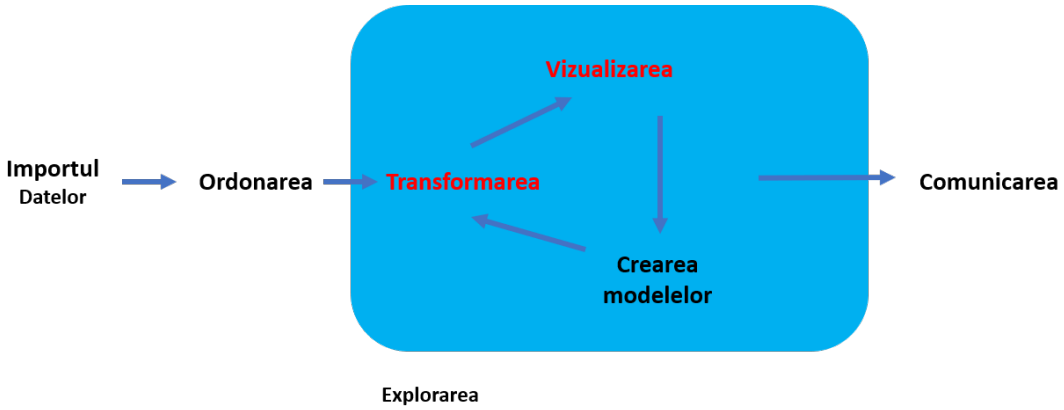
```
1+2

## [1] 3
```

Există două diferențe principale. În consola tastați după simbolul `>` , numit prompt; nu afișăm acest simbol în suportul de curs. În suportul de curs, ieșirile au în față `#>`; în consolă ieșirile apar imediat după cod. Aceste două diferențe înseamnă că, dacă lucrați cu o versiune electronică a suportului de curs, puteți cu ușurință copia codul în consolă și să-l rulați. De-a lungul cursului ne vom folosi de un set consistent de convenții pentru a ne referi la cod:

- Funcțiile sunt într-un font special și sunt urmate de paranteze, de ex. `sum ()` sau `mean ()`.
- Alte obiecte R (cum ar fi datele sau argumentele funcționale) sunt într-un font special, fără paranteze, de ex. `flights` sau `x`.
- Pentru a face claritate din ce pachet provine un obiect, vom folosi numele pachetului urmat de două puncte, de ex. `dplyr::mutate()` sau `nycflights13::flights`.

2. Explorarea datelor (intro)

Scopul acestei secțiuni este de a vă pune la curent cu instrumentele de bază pentru explorarea datelor. Explorarea datelor este arta de a privi datele în examinare, de a genera ipoteze, de a efectua testări rapide, apoi de a repeta totul din nou și din nou. Scopul explorării datelor este de a genera numeroase oportunități promițătoare pe care ulterior le puteți explora mai în profunzime.



În continuare vom studia câteva instrumente utile care:

- Vizualizarea este o modalitate bună pentru a începe programarea în R, deoarece răsplata cunoștințelor este clară: puteți face reprezentări elegante (ploturi) și informative care vă ajută să înțelegeți datele. În reprezentarea datelor vă veți aprofunda în vizualizarea, învățând structura de bază a unei reprezentări ggplot2 și tehnici puternice de transformare a datelor pentru reprezentarea în ploturi.
- Vizualizarea singură, de obicei, nu este suficientă, așa că în procesul de transformare a datelor vom învăța instrumentele principale care permit selectarea variabilelor importante, filtrarea observațiilor cheie, crearea de noi variabile și calcularea statisticilor sumare.
- În cele din urmă, în analiza exploratoare a datelor, vom combina vizualizarea și transformarea cu scepticismul pentru a răspunde la întrebări interesante despre date.

Modelarea este o parte importantă a procesului exploratoriu, însă la moment, încă nu aveți abilitățile necesare pentru a o aplica și învăța în mod eficient. Ne vom întoarce la modelare odată ce vom fi echipați cu tehnici și cunoștințe de programare.