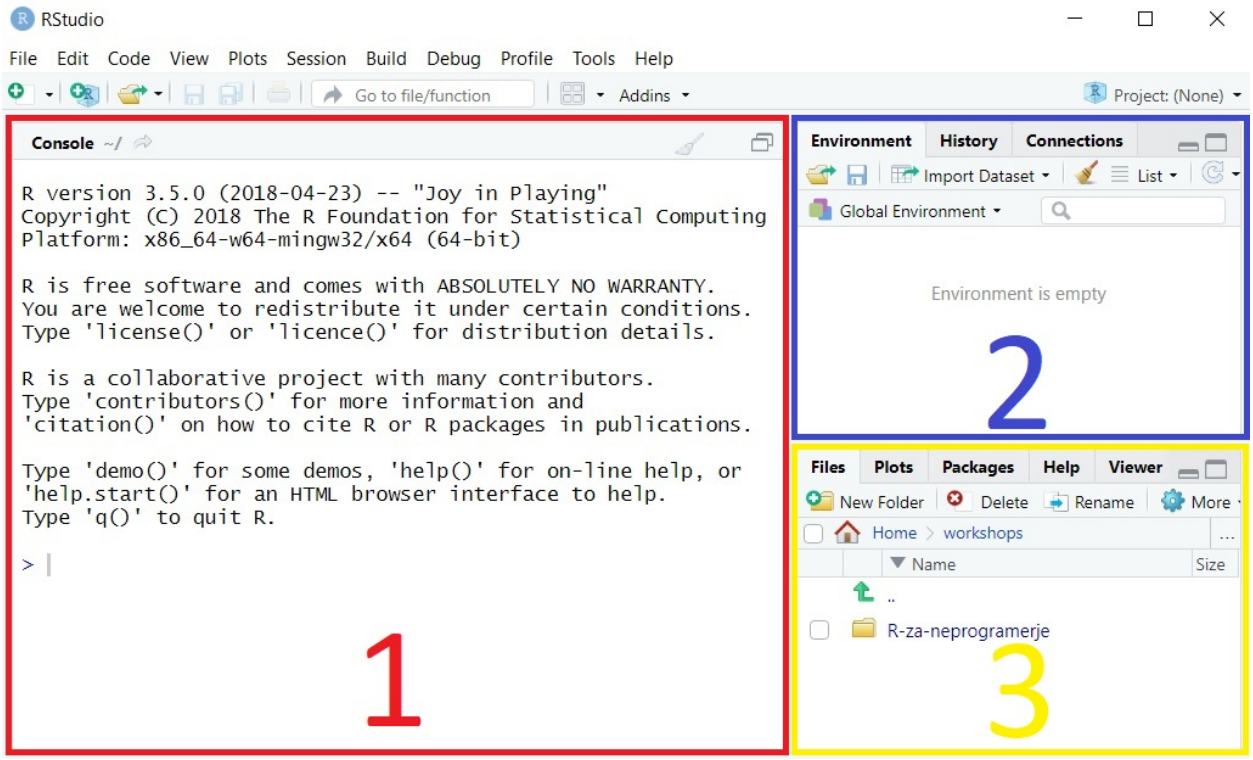
Predavanje 02 – RStudio, skripte, branje in shranjevanje podatkov

RStudio

RStudio je integrirano razvojno okolje (integrated development environment – IDE) za razvoj programov in aplikacij s programskim jezikom R. Bolj splošno, RStudio vsebuje vse potrebno za pisanje programov v R. Predstavlja nadgradnjo preproste konzole R, predvsem pa uporabniku olajša delo. V tem delu bomo predstavili pomembnejše funkcije RStudia.

Ko odpremo RStudio opazimo da je razdeljen na tri okna:



Poglejmo si vsakega izmed oken nekoliko bolj podrobno.

1: Konzola

Konzolo v rdečem okencu smo že spoznali na prvem predavanju. Tukaj se bodo izvajali vsi naši ukazi. Ta del je ekvivalenten konzoli, ki jo premore osnovna različica uporabniškega vmesnika, ki ga dobimo z osnovno namestitvijo programskega jezika R.

2: Okolje in zgodovina

V modrem okencu sta za nas pomembna zavihka **Environment** (okolje) in **History** (zgodovina). Prvi prikazuje spremenljivke, ki jih imamo v trenutni seji R. Spoznali smo že ukaz `ls()`, ki vrne imena vseh

spremenljivk, ki so trenutno shranjene. S pregledom okolja tega klica ne potrebujemo, saj imamo ta seznam ves čas na voljo. Poleg tega pa nam okolje vrne še nekaj podrobnosti o vsaki spremenljivki. Zaenkrat opazimo, da je naše okolje še prazno, saj še nismo definirali nobene spremenljivke. Zgodovina nam hrani vse klice v konzoli, ki smo jih izvedli tekom trenutne seje R.

3: Datoteke, grafi, paketi in pomoč

V rumeno označenem okencu so za nas pomembni zavihki **Files**, **Plots**, **Packages** in **Help**.

1) **Files**. Tukaj lahko brskamo po računalniku, odpiramo datoteke v RStudiu in ustvarjamo nove mape.
2) **Plots**. Tukaj se nam bodo prikazovali grafi, ki jih bomo ustvarili. Vizualizacijo bomo spoznali v prihodnjih predavanjih.
3) **Packages**. Zavihek namenjen namestitvi in nalaganju paketov (knjižnic). Te bomo spoznali kasneje v tem poglavju.
4) **Help**. Vsaka funkcija v R ima t. i. help file, kjer so navedene podrobnosti funkcije, katere argumente lahko vnesemo in primer uporabe ali dva. Uporabo tega bomo spoznali v prihodnjih predavanjih.

Delovni direktorij (working directory)

Preden začnemo z delom v R je dobro določiti mapo, ki bo predstavljala naš delovni direktorij. To nam bo omogočilo lažji dostop do datotek v tej mapi (na primer podatkov) in lažje shranjevanje rezultatov v to mapo. Da določimo delovni direktorij uporabimo ukaz `setwd(<pot-do-mape>)`. Na primer, recimo da želimo delovni direktorij v mapi *C:\Users\Gregor\Documents\R-za-neprogramerje-marec-2021\Predavanje_02*. Potem uporabimo

```
setwd("C:\Users\Gregor\Documents\R-za-neprogramerje-marec-2021\Predavanje_02")
```

```
## Error: '\U' used without hex digits in character string starting ""C:\U"
```

Opazimo, da je R vrnil napako. Ampak, kot smo že rekli, napak se ne bojimo. Poizkusimo ugotoviti, kaj nam R želi s svojim sporočilom povedati. V kolikor nam sporočilo ni razumljivo je najbolje da ga kar pogooglamo. V večini primerov je že nekdo naletel na isto težavo in je postavil vprašanje na enem izmed za to namenjenih portalov (na primer https://stackoverflow.com/). V zgornjem primeru bi hitro ugotovili da je problem v uporabi leve poševnice (backslash). R ima backslash namreč rezerviran za posebne funkcije, tako da ga je treba zamenjati ali z dvojno levo poševnico

```
setwd("C:\\Users\\Gregor\\Documents\\R-za-neprogramerje-marec-2021\\Predavanje_02")
```

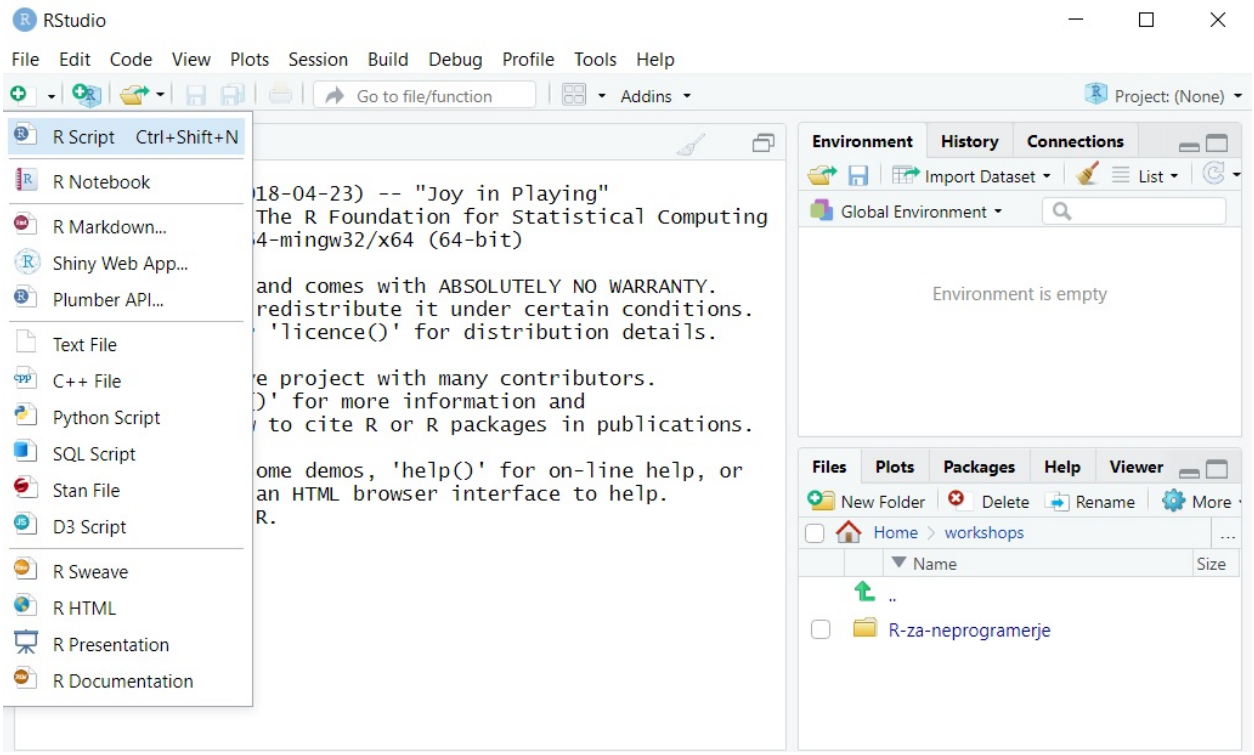
ali s poševnico

```
setwd("C:/Users/Gregor/Documents/R-za-neprogramerje-marec-2021/Predavanje_02")
```

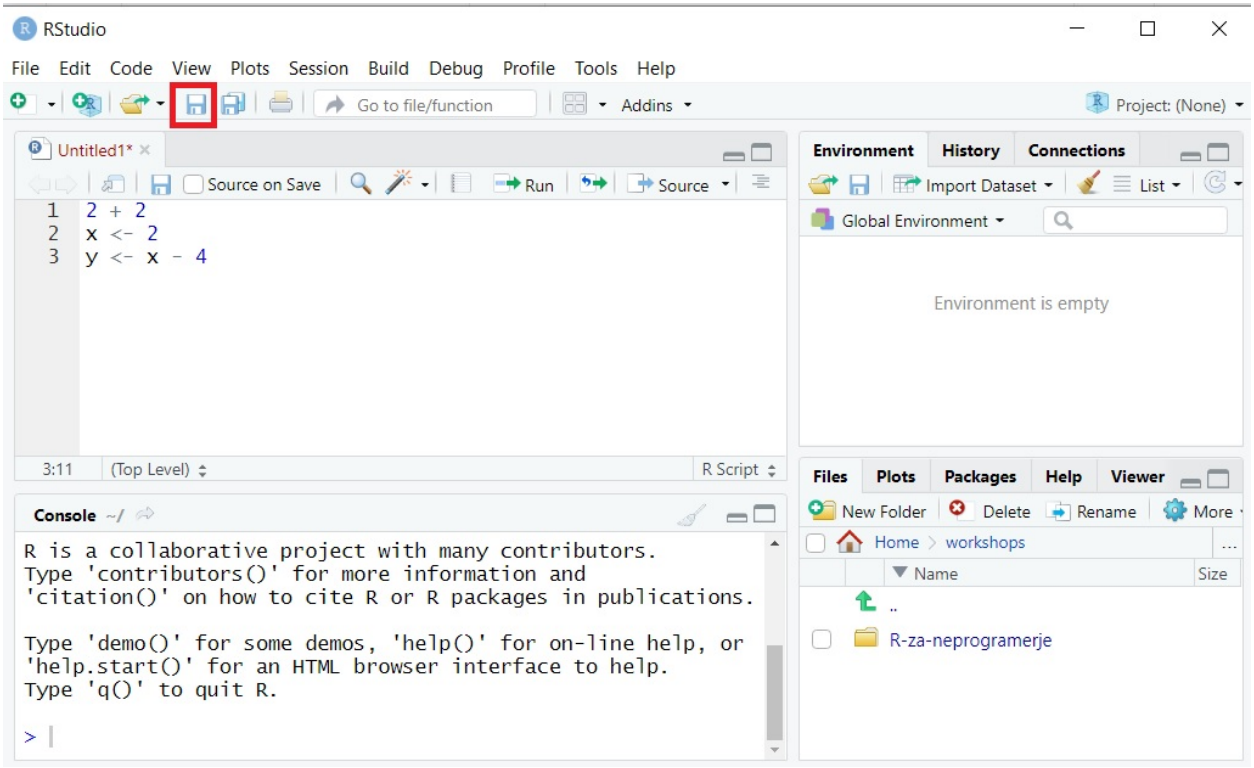
V kolikor kdaj nismo prepričani, katera mapa predstavlja naše trenutno delovno okolje, uporabimo ukaz `getwd()`.

Skripte

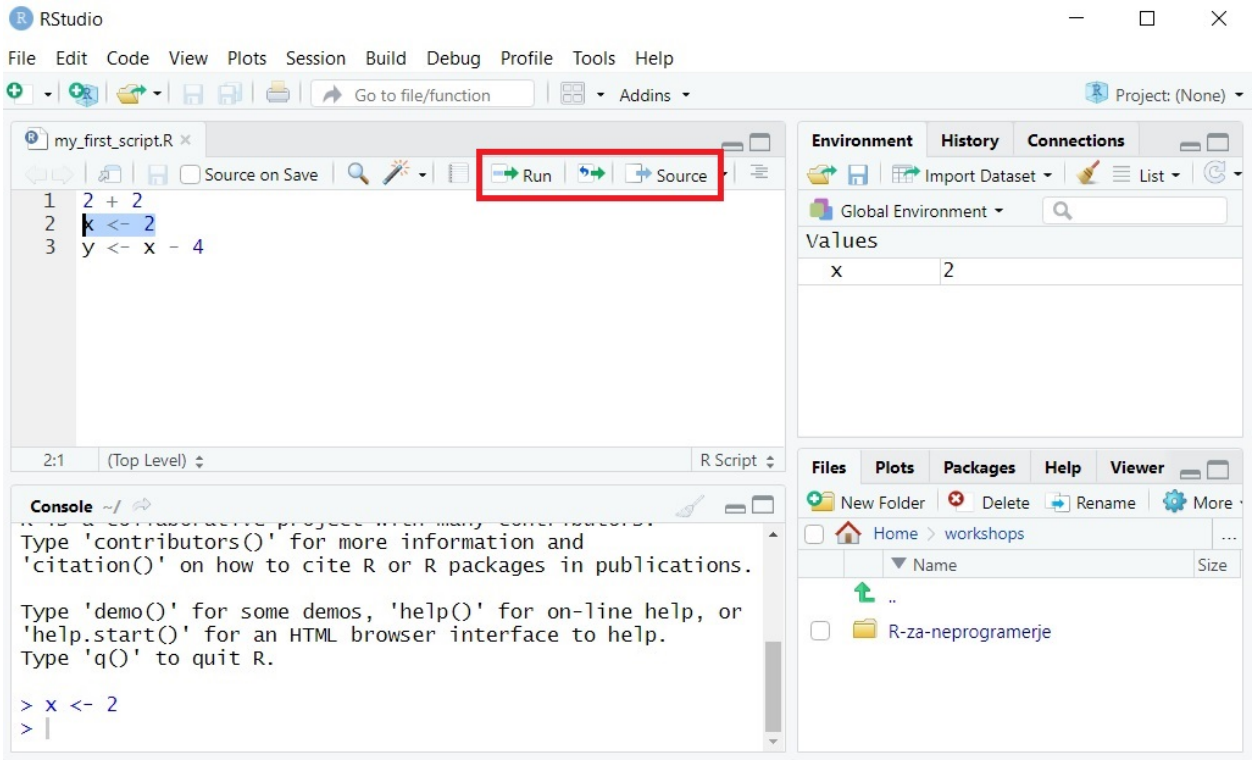
Skripte so glavni sestavni deli programov v R. So preproste datoteke, v katere zaporedno pišemo ukaze, ki jih nato pošiljamo R konzoli. Prednost uporabe skript v primerjavi s pisanjem ukazov direktno v konzolo je v tem, da si ukaze v skripti lahko shranimo. Naredimo sedaj našo prvo skripto. Za to imamo v RStudio na voljo ikono, ki se nahaja pod zavihkom **File** v levem zgornjem kotu.



Ko naredimo novo skripto, se nam odpre novo okno v RStudiu, poleg treh oken, ki smo jih že spoznali. To je okno, ki je v veliki meri namenjeno skriptam. Zapišimo nekaj že znanih ukazov v to novo skripto in jo shranimo. Za shranjevanje skripte lahko uporabimo ikono diskete označeno na spodnji sliki, ali ukaz **Ctrl + S**.



Kako sedaj poženemo te ukaze? Če želimo pognati samo del ukazov v skripti, to naredimo tako da z miško označimo samo tiste ukaze in kliknemo na ukaz **Run** (označeno z rdečo na spodnji sliki) ali uporabimo bljižnico na tipkovnici **Ctrl + Enter**. V kolikor želimo pognati celotno skripto lahko uporabimo ukaz **Source**. Poženimo sedaj ukaz `x <- 2`.



Kaj se je zgodilo? Ukaz **Run** je označen del pognal v konzoli, enako kot če bi mi zapisali ta ukaz v konzolo sami. Opazimo tudi, da sedaj naše okolje več ni prazno – sedaj se v njem nahaja spremenljivka x, ki vsebuje vrednost 2.

V začetku je uporaba ene skripte popolnoma zadovoljiva in lahko celoten program oziroma postopek analize shranimo v samo eno skripto. Ko naše analize in manipualcije postajajo bolj kompleksne, se običajno poslužujemo večih skript.

Paketi

Osnovna različica R nam nudi veliko možnosti za delo s podatki. Vendar včasih potrebujemo kakšne specifične funkcije, ki jih ne najdemo v osnovni različici. Prav tako je osnova različica lahko relativno toga in bi želeli uporabniku prijaznejše funkcije. Za to so na voljo t.i. paketi ali knjižnice. Za R obstajajo številni paketi, saj je R odprtokodni programski jezik in lahko vsak uporabnik kreira svoj paket in ga naloži na internet. Nekaj njihovih funkcionalnosti vključuje:

1) Lepše vizualizacije.
2) Lažje čiščenje in urejanje podatkov.
3) Metode strojnega učenja.
4) Povezave med R in drugimi orodji (kot na primer Microsoft Excel).
5) Avtomatsko generiranje poročil.

V sklopu te delavnice bomo spoznali pakete **openxlsx** za delo z Excelovimi datotekami, **tidyr** za specifične operacije nad razpredelnicami in **ggplot2** za vizualizacijo.

Pakete lahko namestimo s funkcijo `install.packages("<ime-paketa>")` ali s klikom na zavihek **Packages** in nato **install**, kjer nato poiščemo ustrezen paket.

Parametri funkcij

Spoznali smo že nekaj funkcij, na primer `sqrt()`, `sin()`, `mean()`. Nekatere med njimi so prejele tudi več kot 1 argument, na primer `rep()`. Pri teh funkcijah lahko argumente podamo na dva načina:

1) V vrstnem redu. Funkcije imajo vnaprej določen vrstni red argumentov, ki ga moramo poznati. Lahko si tudi pomagamo z `?<ime-funkcije>`.
2) Uporabimo vgrajena imena parametrov. V tem primeru ne rabimo poznati pravega vrstnega reda, ampak samo imena posameznih argumentov. Takšna programska koda je tudi bolj robustna in načeloma bolj pregledna. Imena argumentov lahko prav tako najdemo s klicem `?<ime-funkcije>`.

Poglejmo si različne pristope k podajanju argumentov na primeru funkcije `rep()`. Recimo, da želimo 10 krat ponoviti število 2:

```
rep(2, 10)
```

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

Pri tem je pomemben vrsti red argumentov. Na primer

```
rep(10, 2)
```

```
## [1] 10 10
```

2 krat ponovi število 10. Namesto izbire argumentov glede na vrstni red lahko posamezne argumente poimenujemo:

```
rep(x = 2, times = 10)
```

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

```
rep(times = 10, x = 2)
```

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

V tem primeru je bralcu kode bolj jasno, kaj smo želeli računalniku sporočiti. Poleg tega je takšna koda neodvisna od vrstnega reda in zmanjša možnosti za napake. Prav tako lahko nekatere funkcije prejmejo veliko število parametrov, mi pa želimo nastaviti le nekatere in pri ostalih uporabiti privzeto vrednost (vrednost, ki je v funkciji nastavljena, če uporabnik ne poda druge vrednosti eksplicitno). Na primer, če bi želeli pri neki funkciji nastaviti samo osmi parameter, bi s prvim pristopom (vrstni red parametrov) morali vnesti vseh prvih 8 parametrov. Z drugim pristopom pa bi lahko enostavno podali samo osmega.

Branje podatkov

V kolikor želimo R uporabljati za delo s podatki, se moramo najprej naučiti, kako prenesti podatke iz drugih virov v R. V tem poglavju bomo spoznali branje podatkov iz tekstovnih in excelovih datotek.

Dva pogosta tipa podatkovnih datotek

Podatkovne datoteke lahko srečamo v veliko različnih formatih. Dva najbolj pogosta sta:

1) Excelova datoteka. Večina podjetij za velik del dela s podatki uporablja Microsoft Excel, tako da so Excelove datoteke pogost vir podatkov.
2) Tekstovna datoteka (končnice txt, csv). Podatki so zapisani v tekstovno datoteko. Da ločimo posamezne elemente med seboj, se običajno uporabi podpičje, vejica, ali tabulator (tab).

Branje iz tekstovnih datotek

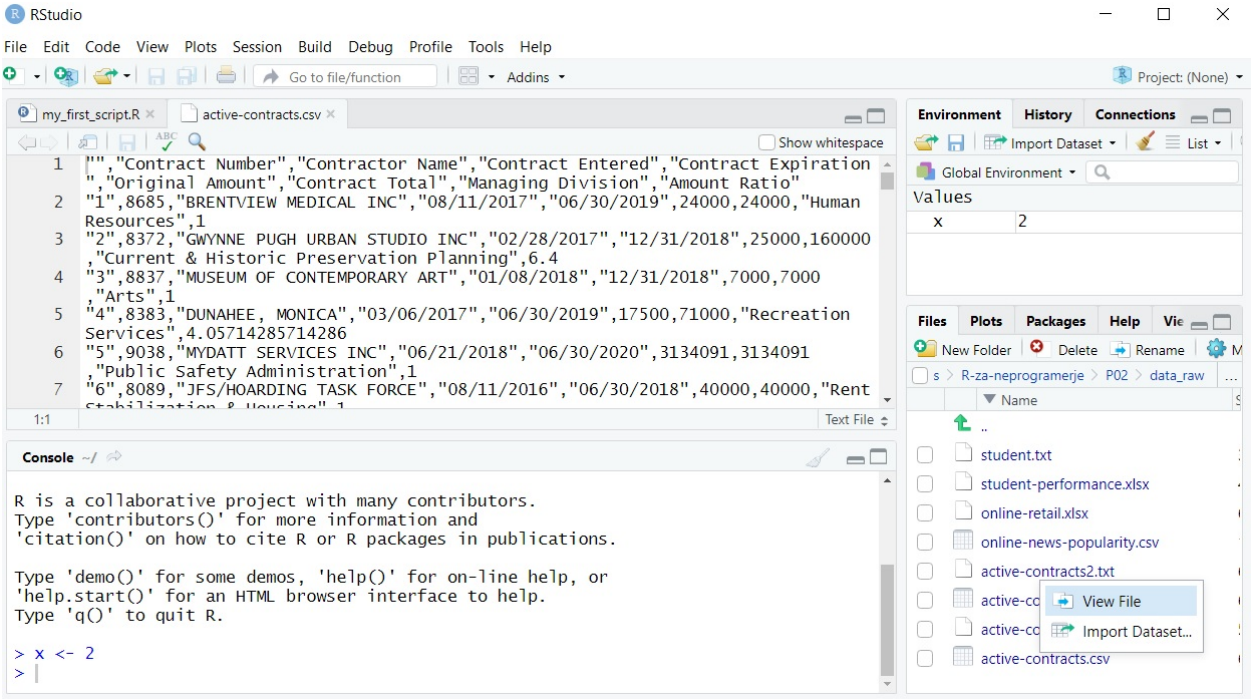
Za branje podatkov iz standardnih txt ali csv zapisov imamo v R na voljo funkcijo `read.table()`. Ta funkcija lahko prejme veliko število parametrov. Za nas so pomembni sledeči:

1) `file`. Pot do datoteke, katero želimo naložiti v trenutno sejo R.
2) `sep`. Določa simbol, ki loči posamezne elemente.
3) `dec`. Določa decimalni simbol.
4) `header`. Ali imajo podatki podana imena stolpcev. Vnesemo lahko vrednosti `TRUE` če je to res in `FALSE` če ni.
5) `quote`. Kateri simbol je uporabljen za narekovaje, ki označujejo besede v tekstovni datoteki.

Kadar so elementi ločeni z vejico ali podpičjem imamo običajno opravka s csv (comma separated value) končnico, čeprav bi lahko te podatke shranili tudi v txt formatu. Kadar imamo elemente ločene kako drugače se običajno uporabi txt končnica.

Naložimo sedaj podatke iz tekstovne datoteke *active-contracts.csv*, ki je na voljo v mapi *data_raw*, v našo trenutno sejo R. Podatki predstavljajo tekoče pogodbe mesta City of West Hollywood. Originalni podatki so na voljo na https://data.world/weho/atdr-sk64. Zaenkrat ne bomo posvečali preveč pozornosti samim podatkom, naš cilj je samo da se jih naučimo naložiti in shraniti.

Preden se lotimo pisanja programske kode, moramo preveriti, kako so podatki shranjeni. Tekstovno datoteko lahko odpremo kar z RStudiem. V zavihku **Files** se pomakenemo do vsebine mape *data_raw*, kliknemo na željeno datoteko in izberemo **View file**.



Opazimo, da imamo decimalno piko, elemente ločene z vejico, prva vrstica predstavlja glavo podatkov in besede so označene z ". Da lahko preberemo podatke, moramo ustrezno nastaviti argumente funkcije `read.table()`. Kjer nastavljamo znake, oziroma besede, moramo okoli njih uporabiti narekovaje, saj jih R drugače ne prepozna:

```
act_contracts <- read.table(file = "./data_raw/active-contracts.csv",
                            dec = ".",
                            sep = ",",
                            header = TRUE,
                            quote = "\"")
```

Pri tem smo pri parametru `quote` znotraj narekovajev uporabili kombinacijo znakov \". Sam znak " ima posebno mesto v R-ju – nakazuje besede – in ga ne moremo enostavno uporabiti kot del besede. Da lahko naredimo to, moramo uporabiti to kombinacijo z levo poševnico.

Potrebno je omeniti, da pika v zgornji poti predstavlja trenutni direktorij. Celotno besedilo znotraj narekovajev torej kaže na datoteko, relativno glede na naš delovni direktorij. R avtomatsko shrani prebrane podatke v **podatkovno razpredelnico** oziroma `data.frame`. Ta objekt si lahko predstavljamo kot Excelovo tabelo, ki jo R hrani v obliki spremenljivke, v našem primeru `act_contracts`. Za ogled celotne razpredelnice lahko uporabimo funkcijo `View()`.

```
View(act_contracts)
```

Enako lahko dosežemo s klikom na to spremenljivko v zavihku **Environment**. Velikokrat želimo videti samo strukturo razpredelnice. Za ta namen lahko pogledamo samo prvih 6 vrednosti v razpredelnici s klicem `head()`.

```
head(act_contracts)
```

```
##   Contract.Number              Contractor.Name Contract.Entered
## 1            8685        BRENTVIEW MEDICAL INC       08/11/2017
## 2            8372 GWYNNE PUGH URBAN STUDIO INC       02/28/2017
## 3            8837   MUSEUM OF CONTEMPORARY ART       01/08/2018
## 4            8383              DUNAHEE, MONICA       03/06/2017
## 5            9038          MYDATT SERVICES INC       06/21/2018
## 6            8089      JFS/HOARDING TASK FORCE       08/11/2016
##   Contract.Expiration Original.Amount Contract.Total
## 1          06/30/2019           24000          24000
## 2          12/31/2018           25000         160000
## 3          12/31/2018            7000           7000
## 4          06/30/2019           17500          71000
## 5          06/30/2020         3134091        3134091
## 6          06/30/2018           40000          40000
##                          Managing.Division Amount.Ratio
## 1                          Human Resources     1.000000
## 2 Current & Historic Preservation Planning     6.400000
## 3                                    Arts     1.000000
## 4                     Recreation Services     4.057143
## 5             Public Safety Administration     1.000000
## 6             Rent Stabilization & Housing     1.000000
```

Ko imamo razpredelnico shranjeno kot spremenljivko v R, lahko nad njo izvajamo operacije in manipulacije, ki jih bomo spoznali v prihodnjih predavanjih. Dobra stran R-ja je v tem, da to razpredelnico spreminjamo lokalno znotraj R in originalna datoteka, iz katere smo prebrali podatke, ostane nespremenjena.

Za vajo naložimo še podatke *active-contracts.txt*, *active-contracts2.csv* in *active-contracts2.txt*. Pri vseh datotekah gre za iste podatke shranjene v različnih formatih.

Opazimo da ima datoteka *active-contracts.txt* elemente ločene s tabulatorjem in decimalno piko. Prav tako niso besede označene z narekovaji. Če želimo to sporočiti funkciji `read.table()` uporabimo prazno besedo `""`. Tudi za tabulator imamo posebno kombinacijo simbolov in sicer `\t`. Uporabimo vse zapisano, da preberemo te podatke:

```
act_contracts <- read.table(file = "./data_raw/active-contracts.txt",
                            dec = ".",
                            sep = "\t",
                            header = TRUE,
                            quote = "")
```

Datoteka *active-contracts2.csv* ima elemente ločene s podpičjem in decimalno vejico.

```
act_contracts <- read.table(file = "./data_raw/active-contracts2.csv",
                            dec = ",",
                            sep = ";",
```

```
                              header = TRUE,
                              quote = "\"")
```

Datoteka *active-contracts2.txt* ima elemente ločene z dvopičjem in decimalno vejico.

```
act_contracts <- read.table(file = "./data_raw/active-contracts2.txt",
                              dec = ",",
                              sep = ":",
                              header = TRUE,
                              quote = "\"")
```

V kolikor kljub pravilnim nastavitvam argumentov R razpredelnice ne prebere pravilno, moramo to nadalje raziskati. Velikokrat gre v tem primeru za kakšne posebne znake v besedilnih stolpcih in moramo zadevo reševati od primera do primera.

Variante `read.table()`

Obstajajo standardni formati, v katerih največkrat najdemo podatke. Da ne potrebujemo vsakič ročno nastavljati vseh argumentov v funkciji `read.table()` imamo na voljo 4 funkcije, ki že imajo ustrezno podane privzete vrednosti argumentov in jim moramo podati samo pot do datoteke:

1) `read.csv()`. Kadar so elementi ločeni z vejico in imamo decimalno piko.
2) `read.csv2()`. Kadar so elementi ločeni s podpičjem in imamo decimalno vejico.
3) `read.delim()`. Kadar so elementi ločeni s tabulatorjem in imamo decimalno piko.
4) `read.delim2()`. Kadar so elementi ločeni s tabulatorjem in imamo decimalno vejico.

Na primer

```
act_contracts <- read.csv("./data_raw/active-contracts.csv")
```

Branje iz excelovih datotek

V tem poglavju se bomo lotili branja podatkov iz Excelovih datotek. Za to potrebujemo paket **openxlsx**. Najprej moramo namestiti ta paket.

```
install.packages("openxlsx")
```

Ter ga naložiti v našo trenutno sejo R.

```
library("openxlsx")
```

```
## Warning: package 'openxlsx' was built under R version 4.0.4
```

Za branje podatkov iz Exelovih datotek uporabimo funckijo `read.xlsx()`. Funkcija zahteva vsaj dva argumenta:

1) `file`. Pot do datoteke, iz katere želimo prebrati podatke.
2) `sheet`. Indeks ali ime Excelovega lista, ki ga želimo odpreti. Običajno se v eni excelovi datoteki nahaja več listov. S tem argumentom R-ju sporočimo, kateri list naj naloži.

Poglejmo si sedaj kako naložiti podatke iz excelove datoteke. V mapi *data_raw* imamo datoteko *student-performance.xlsx*, kjer so shranjeni podatki o uspešnosti dijakov pri testih iz matematike in portugalščine. V datoteki *student.txt* se nahajajo razlage vrednosti posameznih spremenljivk. Podatki so bili uporabljeni v raziskavi Cortez and Silva (2008) in so na voljo na https://archive.ics.uci.edu/ml/datasets/Student+Performance. Preberimo podatke testov iz matematike in portugalščine in shranimo vsake v svojo spremenljivko.

```
student_math <- read.xlsx("./data_raw/student-performance.xlsx",
                          sheet = "Math scores")
student_port <- read.xlsx("./data_raw/student-performance.xlsx",
                          sheet = "Portuguese scores")
```

Enako dosežemo z

```
student_math <- read.xlsx("./data_raw/student-performance.xlsx", sheet = 2)
student_port <- read.xlsx("./data_raw/student-performance.xlsx", sheet = 1)
head(student_math)
```

```
##   school sex famsize Pstatus traveltime studytime internet absences G1 G2 G3
## 1     GP   F     GT3       A          2         2       no        6  5  6  6
## 2     GP   F     GT3       T          1         2      yes        4  5  5  6
## 3     GP   F     LE3       T          1         2      yes       10  7  8 10
## 4     GP   F     GT3       T          1         3      yes        2 15 14 15
## 5     GP   F     GT3       T          1         2       no        4  6 10 10
## 6     GP   M     LE3       T          1         2      yes       10 15 15 15
##   averageScore
## 1     5.666667
## 2     5.333333
## 3     8.333333
## 4    14.666667
## 5     8.666667
## 6    15.000000
```

```
head(student_port)
```

```
##   school sex famsize Pstatus traveltime studytime internet absences G1 G2 G3
## 1     GP   F     GT3       A          2         2       no        4  0 11 11
## 2     GP   F     GT3       T          1         2      yes        2  9 11 11
## 3     GP   F     LE3       T          1         2      yes        6 12 13 12
## 4     GP   F     GT3       T          1         3      yes        0 14 14 14
## 5     GP   F     GT3       T          1         2       no        0 11 13 13
## 6     GP   M     LE3       T          1         2      yes        6 12 12 13
##   averageScore
## 1     7.333333
## 2    10.333333
## 3    12.333333
## 4    14.000000
## 5    12.333333
## 6    12.333333
```

Shranjevanje podatkov

Ena glavnih prednosti R-ja je, da lahko originalne podatke shranimo in tam ostanejo nespremenjeni. Ko jih naložimo v R, se znotraj R ustvari kopija originalnih podatkov. Nato so vse manipulacije nad njimi shranjene v R skripti. Potem lahko spremenjene podatke shranimo kot novo datoteko. Dokaj običajen pojav pri delu s podatki je, da čiščenje in urejanje podatkov vzame več časa, kot ga dejanska analiza. Zato je ta korak pri delu s podatki zelo pomemben.

Dodatna prednost uporabe R skript je v tem, da lahko podatke, ki se spreminjajo periodično, uredimo z isto skripto. Na primer, če bi morali vsak mesec prenesti podatke iz podatkovne baze in jih nato ročno očistiti, bi nam to vzelo precej časa. Z uporabo R lahko proces čiščenja vsaj do neke mere avtomatiziramo in prihranimo dragocen čas za preostale naloge.

V prejšnjem poglavju smo se naučili prebrati podatke iz različnih virov. V tem poglavju bomo pogledali kako podatke shraniti.

Shranjevanje v tekstovne datoteke

Za shranjevanje v tekstovne datoteke lahko uporabimo funkcijo `write.table()`, ki prav tako prejme veliko število parametrov. Za nas so pomembni:

1) `x`. Spremenljivka (razpredelnica), ki jo želimo shraniti.
2) `file`. Pot do datoteke, v katero želimo shraniti podatke.
3) `sep`. Določa simbol, ki loči posamezne elemente.
4) `dec`. Določa decimalni simbol.

Privzeto ta funkcija shrani tudi imena stolpcev in uporabi dvojne narekovaje za neštevilske celice. Predlagamo, da teh privzetih vrednosti ne spreminjamo.

Shranimo podatke o uspešnosti dijakov pri matematiki v tekstovno datoteko s podpičjem in decimalno vejico.

```
write.table(x = student_math,
            file = "./data_saved/student-math-scores.csv",
            dec = ",",
            sep = ";")
```

Recimo, da želimo kot separator tabulator, ki ga v R označimo z `\t`. Shranimo sedaj podatke o uspešnosti dijakov pri portugalščini.

```
write.table(x = student_port,
            file = "./data_saved/student-portuguese-scores.txt",
            sep = "\t",
            dec = ".")
```

Shranjevanje v excelove datoteke

Za shranjevanje xlsx datotek obstaja funkcija `write.xlsx`. Najbolj osnovna uporaba zahteva samo dva parametra – razpredelnico, ki jo želimo shraniti in ime datoteke, kamor bomo to razpredelnico shranili. Shranimo sedaj podatke o pogodbah v Excelovo datoteko.

```
write.xlsx(act_contracts, "./data_saved/active-contracts.xlsx")
```

Ta klic bo avtomatsko ustvaril novo Excelovo datoteko z enim listom, v katerem bo ta razpredelnica.

Shranjevanje v več listov iste datoteke

V kolikor želimo v Excelovo datoteko shraniti več listov, se moramo poslužiti sledečega zaporedja ukazov.

```
wb <- createWorkbook()
```

Ta klic ustvari prazno Excelovo datoteko v trenutni seji R. Nato moramo v to spremenljivko dodati list s funkcijo `addWorksheet`:

```
addWorksheet(wb, sheetName = "<ime-lista>")
```

Sedaj imamo Excelovo datoteko, ki ima en prazen list. Ta datoteka je shranjena samo v trenutni seji R. Sedaj ko imamo prazen list, lahko v njega zapišemo podatke:

```
writeData(wb, sheet = "<ime-lista>", <ime-razpredelnice-v-R>)
```

Na koncu je potrebno to Excelovo datoteko shraniti:

```
saveWorkbook(wb, file = "<pot-do-shranjene-datoteke>")
```

Recimo, da želimo shraniti vse 3 razpredelnice, ki smo jih naložili v R, v eno samo Excelovo datoteko. Potem bi naš program izgledal tako:

```
wb <- createWorkbook()
addWorksheet(wb, sheetName = "active contracts")
addWorksheet(wb, sheetName = "student math scores")
addWorksheet(wb, sheetName = "student Portuguese scores")
writeData(wb, sheet = "active contracts", act_contracts)
writeData(wb, sheet = "student math scores", student_math)
writeData(wb, sheet = "student Portuguese scores", student_port)
saveWorkbook(wb, file = "./data_saved/several-sheets-example.xlsx")
```

Domača naloga

1) V mapi *data_raw* se nahajajo podatki o popularnosti posameznih novic v tekstovni datoteki. Podatki so bili del znanstvene raziskave Fernandes, Vinagre, and Cortez (2015) in so dostopni na https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity. Vaša naloga je, da pripravite skripto, ki bo nastavila pot do delovnega direktorija, naložila ustrezne pakete, naložila te podatke v spremenljivko v R. Nato shranite te podatke v xlsx datoteko z imenom *online-news-popularity.xlsx*.

V kolikor smo podatke pravilno prebrali v R, bi s klicem `head(<ime-spremenljivke>)` morali dobiti:

```
head(news_pop)
```

```
##                                                    url num_imgs
## 1   http://mashable.com/2013/01/07/amazon-instant-video-browser/        1
## 2    http://mashable.com/2013/01/07/ap-samsung-sponsored-tweets/        1
## 3 http://mashable.com/2013/01/07/apple-40-billion-app-downloads/        1
## 4       http://mashable.com/2013/01/07/astronaut-notre-dame-bcs/        1
```

```
## 5               http://mashable.com/2013/01/07/att-u-verse-apps/       20
## 6               http://mashable.com/2013/01/07/beewi-smart-toys/        0
##   num_videos num_keywords is_weekend global_rate_positive_words
## 1          0            5          0                 0.04566210
## 2          0            4          0                 0.04313725
## 3          0            6          0                 0.05687204
## 4          0            7          0                 0.04143126
## 5          0            7          0                 0.07462687
## 6          0            9          0                 0.02972973
##   global_rate_negative_words shares
## 1                0.013698630    593
## 2                0.015686275    711
## 3                0.009478673   1500
## 4                0.020715631   1200
## 5                0.012126866    505
## 6                0.027027027    855
```

Ali smo podatke shranili pravilno v xlsx datoteko preverite tako, da jo odprete in pogledate, ali tabela izgleda enako kot ta v R.

2) V mapi *data_raw* se nahajajo podatki o spletnih nakupih darilnih in zabavnih izdelkov v Excelovi datoteki. Podatki so bili del znanstvene raziskave Chen, Sain, and Guo (2012) in so dostopni na https://archive.ics.uci.edu/ml/datasets/Online+Retail. Datoteka ima 3 liste, v vsakem se nahajajo podatki o nakupih iz ene izmed treh držav. Vaša naloga je, da pripravite skripto, ki bo nastavila pot do delovnega direktorija, naložila ustrezne pakete in prebrala vsak list posebej ter ga shranila kot spremenljivko z ustreznim imenom. Nato shranite razpredelnico za Avstrijo v tekstovno datoteko s separatorjem vejico in decimalno piko, za Italijo s separatorjem podpičjem in decimalno vejico in za Grčijo v tekstovno datoteko s separatorjem dvopičjem in decimalno vejico.

V kolikor smo podatke pravilno prebrali v R, bi s klicem `head(<ime-spremenljivke>)` morali dobiti:

```
head(retail_aus)
```

```
##   InvoiceNo StockCode                         Description Quantity UnitPrice
## 1   C538971     22153     ANGEL DECORATION STARS ON DRESS      -48      0.42
## 2    539330     37449  CERAMIC CAKE STAND + HANGING CAKES        8      8.50
## 3    539330     37446  MINI CAKE STAND WITH HANGING CAKES        8      1.45
## 4    539330     22962               JAM JAR WITH PINK LID       12      0.85
## 5    539330     21428 SET3 BOOK BOX GREEN GINGHAM FLOWER         4      4.25
## 6    539330     22113         GREY HEART HOT WATER BOTTLE        4      3.75
##   Country
## 1 Austria
## 2 Austria
## 3 Austria
## 4 Austria
## 5 Austria
## 6 Austria
```

```
head(retail_ita)
```

```
##   InvoiceNo StockCode                         Description Quantity UnitPrice
## 1    537022     22791        T-LIGHT GLASS FLUTED ANTIQUE       12      1.25
```

```
## 2    537022     21287       SCENTED VELVET LOUNGE CANDLE        12      1.25
## 3    537022     79337       BLUE FLOCK GLASS CANDLEHOLDER        6      1.65
## 4    537022     85111 SILVER GLITTER FLOWER VOTIVE HOLDER       12      1.25
## 5    537022     85038     6 CHOCOLATE LOVE HEART T-LIGHTS        6      2.10
## 6    537022     22809             SET OF 6 T-LIGHTS SANTA        6      2.95
##   Country
## 1   Italy
## 2   Italy
## 3   Italy
## 4   Italy
## 5   Italy
## 6   Italy
```

```
head(retail_gre)
```

```
##   InvoiceNo StockCode                         Description Quantity UnitPrice
## 1    541932     22699    ROSES REGENCY TEACUP AND SAUCER        24      2.55
## 2    541932     22697     GREEN REGENCY TEACUP AND SAUCER       24      2.55
## 3    541932     22957 SET 3 PAPER VINTAGE CHICK PAPER EGG       24      2.95
## 4    541932     22720   SET OF 3 CAKE TINS PANTRY DESIGN        24      4.25
## 5    541932    72760B   VINTAGE CREAM 3 BASKET CAKE STAND       16      8.49
## 6    541932     22763             KEY CABINET MA CAMPAGNE       12      8.50
##   Country
## 1  Greece
## 2  Greece
## 3  Greece
## 4  Greece
## 5  Greece
## 6  Greece
```

Ali smo podatke shranili pravilno v tekstovne datoteke preverite tako, da jih odprete in pogledate, ali so shranjene v pravem formatu.

Reference

Chen, Daqing, Sai Laing Sain, and Kun Guo. 2012. "Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining." *Journal of Database Marketing & Customer Strategy Management* 19 (3): 197–208.

Cortez, Paulo, and Alice Maria Gonçalves Silva. 2008. "Using data mining to predict secondary school student performance."

Fernandes, Kelwin, Pedro Vinagre, and Paulo Cortez. 2015. "A proactive intelligent decision support system for predicting the popularity of online news." In *Portuguese Conference on Artificial Intelligence*, 535–46. Springer.