1.1 R Commands

This is an R Markdown Notebook, outlining the different commands asked in Task 5 of 2.11 of the course textbook

library() opens a window that lists the packages that are installed on your computer;

```r
library()
```

library() opens a window that lists the packages that are installed on your computer;

```r
search()
```

```
## [1] ".GlobalEnv"        "package:stats"     "package:graphics"
## [4] "package:grDevices" "package:utils"     "package:datasets"
## [7] "package:methods"   "Autoloads"         "package:base"
```

Install seqinr from the closest CRAN mirror and load it for this session.

```r
?install.packages

# Note: the parameter is a quoted string!
install.packages("seqinr",repos="https://cran.rstudio.com/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/rk/jp3gt8553b37kp5ms0rgb2_00000gn/T//RtmpKlysLJ/downloaded_packages
```

launch a new window with the seqinr package info

```r
library(seqinr)
```

list all the functions available in the seqinr package.

```r
ls("package:seqinr")
```

```
##   [1] "a"                       "aaa"
##   [3] "AAstat"                  "acnucclose"
##   [5] "acnucopen"               "al2bp"
##   [7] "alllistranks"            "alr"
##   [9] "amb"                     "as.alignment"
##  [11] "as.matrix.alignment"     "as.SeqAcnucWeb"
##  [13] "as.SeqFastaAA"           "as.SeqFastadna"
##  [15] "as.SeqFrag"              "autosocket"
##  [17] "baselineabif"            "bma"
##  [19] "c2s"                     "cai"
##  [21] "cfl"                     "choosebank"
##  [23] "circle"                  "clfcd"
##  [25] "clientid"                "closebank"
##  [27] "col2alpha"               "comp"
##  [29] "computePI"               "con"
##  [31] "consensus"               "count"
##  [33] "countfreelists"          "countsubseqs"
##  [35] "crelistfromclientdata"   "css"
##  [37] "dia.bactgensize"         "dia.db.growth"
##  [39] "dist.alignment"          "dotchart.uco"
##  [41] "dotPlot"                 "draw.oriloc"
##  [43] "draw.rearranged.oriloc"  "draw.recstat"
##  [45] "exseq"                   "extract.breakpoints"
##  [47] "extractseqs"             "fastacc"
##  [49] "gb2fasta"                "gbk2g2"
##  [51] "gbk2g2.euk"              "GC"
##  [53] "GC1"                     "GC2"
##  [55] "GC3"                     "GCpos"
##  [57] "get.db.growth"           "get.ncbi"
##  [59] "getAnnot"                "getAnnot.default"
##  [61] "getAnnot.list"           "getAnnot.logical"
##  [63] "getAnnot.qaw"            "getAnnot.SeqAcnucWeb"
##  [65] "getAnnot.SeqFastaAA"     "getAnnot.SeqFastadna"
##  [67] "getAttributsocket"       "getFrag"
##  [69] "getFrag.character"       "getFrag.default"
##  [71] "getFrag.list"            "getFrag.logical"
##  [73] "getFrag.qaw"             "getFrag.SeqAcnucWeb"
##  [75] "getFrag.SeqFastaAA"      "getFrag.SeqFastadna"
##  [77] "getFrag.SeqFrag"         "getKeyword"
##  [79] "getKeyword.default"      "getKeyword.list"
##  [81] "getKeyword.logical"      "getKeyword.qaw"
##  [83] "getKeyword.SeqAcnucWeb"  "getLength"
##  [85] "getLength.character"     "getLength.default"
##  [87] "getLength.list"          "getLength.logical"
##  [89] "getLength.qaw"           "getLength.SeqAcnucWeb"
##  [91] "getLength.SeqFastaAA"    "getLength.SeqFastadna"
##  [93] "getLength.SeqFrag"       "getlistrank"
##  [95] "getliststate"            "getLocation"
##  [97] "getLocation.default"     "getLocation.list"
##  [99] "getLocation.logical"     "getLocation.qaw"
## [101] "getLocation.SeqAcnucWeb" "getName"
## [103] "getName.default"         "getName.list"
## [105] "getName.logical"         "getName.qaw"
## [107] "getName.SeqAcnucWeb"     "getName.SeqFastaAA"
## [109] "getName.SeqFastadna"     "getName.SeqFrag"
## [111] "getNumber.socket"        "getSequence"
## [113] "getSequence.character"   "getSequence.default"
## [115] "getSequence.list"        "getSequence.logical"
## [117] "getSequence.qaw"         "getSequence.SeqAcnucWeb"
## [119] "getSequence.SeqFastaAA"  "getSequence.SeqFastadna"
## [121] "getSequence.SeqFrag"     "getTrans"
## [123] "getTrans.character"      "getTrans.default"
## [125] "getTrans.list"           "getTrans.logical"
## [127] "getTrans.qaw"            "getTrans.SeqAcnucWeb"
## [129] "getTrans.SeqFastadna"    "getTrans.SeqFrag"
## [131] "getType"                 "gfrag"
## [133] "ghelp"                   "gln"
## [135] "glr"                     "gls"
## [137] "is.SeqAcnucWeb"          "is.SeqFastaAA"
## [139] "is.SeqFastadna"          "is.SeqFrag"
## [141] "isenum"                  "isn"
## [143] "kaks"                    "kdb"
## [145] "knowndbs"                 "lseqinr"
## [147] "modifylist"               "move"
## [149] "mv"                       "n2s"
## [151] "ncbi.fna.url"             "ncbi.gbk.url"
## [153] "ncbi.ptt.url"             "ncbi.stats"
## [155] "oriloc"                   "parser.socket"
## [157] "peakabif"                 "permutation"
## [159] "pga"                      "plot.SeqAcnucWeb"
## [161] "plotabif"                 "plotladder"
## [163] "plotPanels"               "pmw"
## [165] "prepgetannots"            "prettyseq"
## [167] "print.qaw"                "print.SeqAcnucWeb"
## [169] "query"                    "quitacnuc"
## [171] "read.abif"                "read.alignment"
## [173] "read.fasta"               "readBins"
## [175] "readfirstrec"             "readPanels"
## [177] "readsmj"                  "rearranged.oriloc"
## [179] "recstat"                  "residuecount"
## [181] "reverse.align"            "rho"
## [183] "rot13"                    "s2c"
## [185] "s2n"                      "savelist"
## [187] "SEQINR.UTIL"              "setlistname"
## [189] "splitseq"                 "stresc"
## [191] "stutterabif"              "summary.SeqFastaAA"
## [193] "summary.SeqFastadna"      "swap"
## [195] "syncodons"                "synsequence"
## [197] "tablecode"                "test.co.recstat"
## [199] "test.li.recstat"          "translate"
## [201] "trimSpace"                "uco"
## [203] "ucoweight"                "where.is.this.acc"
## [205] "words"                    "words.pos"
## [207] "write.fasta"              "zscore"
```

run a sample function

```
seqinr::a("Tyr")
```

```
## [1] "Y"
```

```
seqinr::words(3, c("A", "G", "C", "U"))
```

```
##  [1] "AAA" "AAG" "AAC" "AAU" "AGA" "AGG" "AGC" "AGU" "ACA" "ACG" "ACC" "ACU"
## [13] "AUA" "AUG" "AUC" "AUU" "GAA" "GAG" "GAC" "GAU" "GGA" "GGG" "GGC" "GGU"
## [25] "GCA" "GCG" "GCC" "GCU" "GUA" "GUG" "GUC" "GUU" "CAA" "CAG" "CAC" "CAU"
## [37] "CGA" "CGG" "CGC" "CGU" "CCA" "CCG" "CCC" "CCU" "CUA" "CUG" "CUC" "CUU"
## [49] "UAA" "UAG" "UAC" "UAU" "UGA" "UGG" "UGC" "UGU" "UCA" "UCG" "UCC" "UCU"
## [61] "UUA" "UUG" "UUC" "UUU"
```

Testing of various functions ?data -> launches a help page about datasets data available for seqinr:

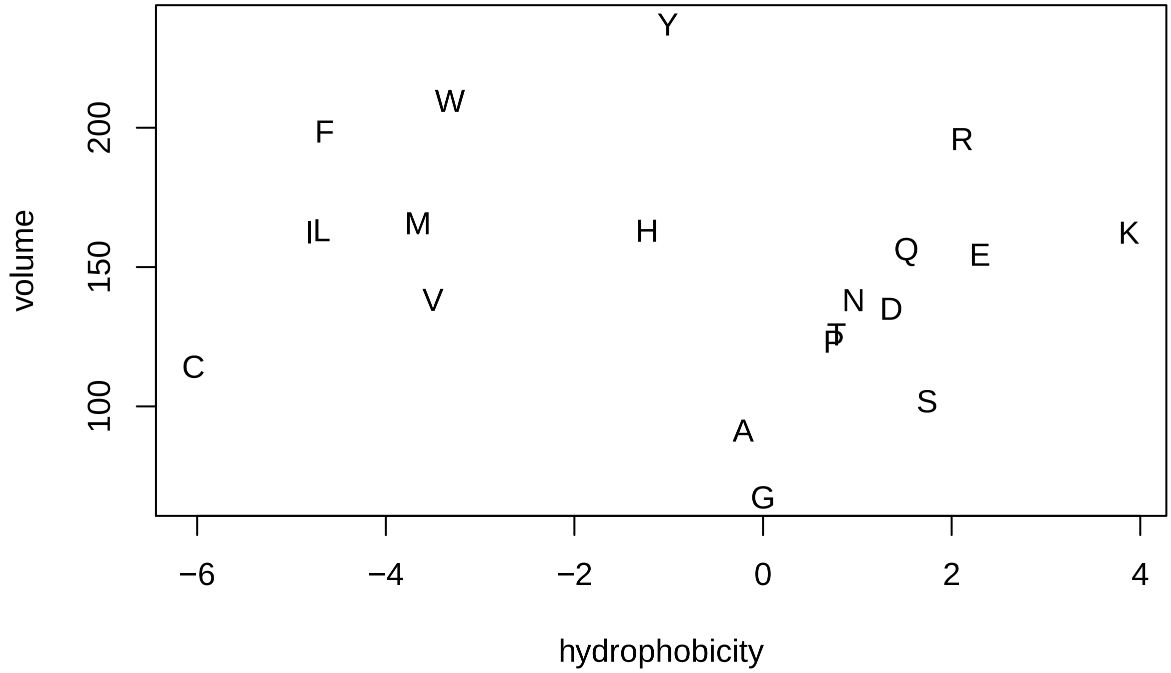
```
data(package="seqinr")
```

load aaindex

```
data(aaindex, package="seqinr")
```

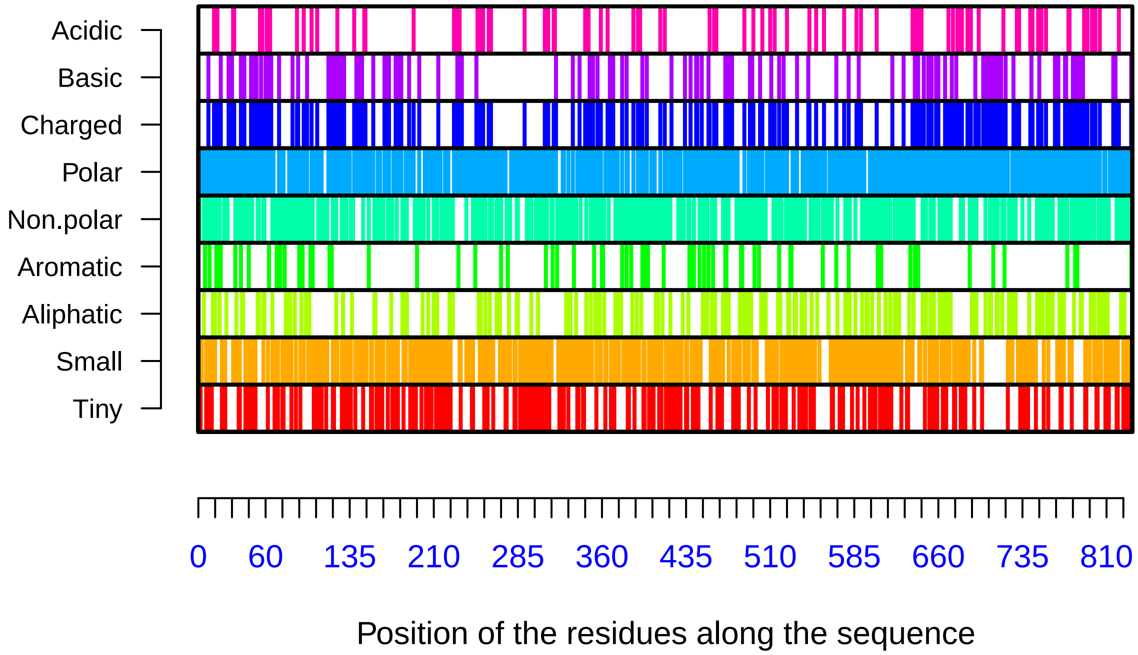
an example of data available is pK values for the side chain of charged amino acids from various sources

```
plot(aaindex$FASG890101$I,
     aaindex$PONJ960101$I,
     xlab="hydrophobicity", ylab="volume", type="n")
text(aaindex$FASG890101$I,
     aaindex$PONJ960101$I,
     labels=a(names(aaindex$FASG890101$I)))
```

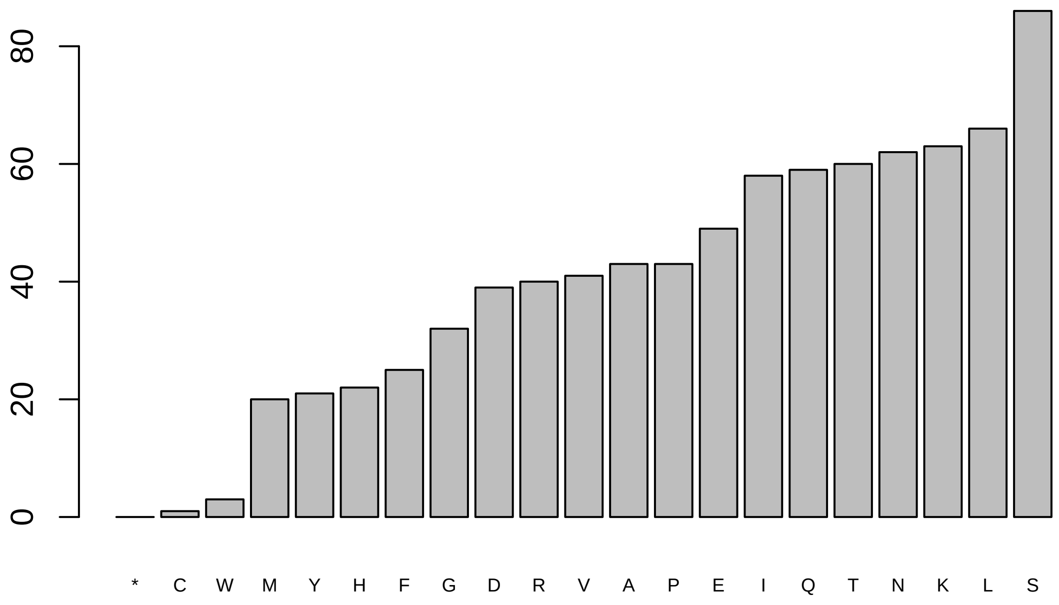


another graphing example:

```
seqinr::choosebank("swissprot")
mySeq <- seqinr::query("mySeq", "N=MBP1_YEAST")
mbp1 <- seqinr::getSequence(mySeq)
seqinr::closebank()
x <- seqinr::AAstat(mbp1[[1]])
```



```
barplot(sort(x$Compo), cex.names = 0.6)
```


0
20
40
60
80
*
C
W
M
Y
H
F
G
D
R
V
A
P
E
I
Q
T
N
K
L
S