Weekl 2

Contents

1 Week 2

1.1 Reading from MySQL

1.1.1 mySQL

Free and widely used open source database software Widely used in internet based applications Data are structured in - Databases - Tables within databases - Fields within tables Each row is called a record

1.1.2 Connecting and listing databases

```
library(RMySQL)
```

```
## Loading required package: DBI
```

```
ucscDb <-
  dbConnect(MySQL(), user = "genome", host = "genome-mysql.cse.ucsc.edu")
result <- dbGetQuery(ucscDb, "show databases;")

dbDisconnect(ucscDb)
```

```
## [1] TRUE
```

```
result
```

Database
acaChl1
ailMel1
allMis1
allSin1
amaVit1
anaPla1
ancCey1
angJap1
anoCar1
anoCar2
anoGam1
anoGam3
apaSpi1
apaVit1
apiMel1
apiMel2
aplCal1
aptFor1
aptMan1
aquChr2
araMac1
ascSuu1
balAcu1
balPav1
bisBis1
bosTau2
bosTau3
bosTau4
bosTau5
bosTau6
bosTau7
bosTau8
bosTau9
bosTauMd3
braFlo1
bruMal2
bucRhi1
burXyl1

Database
caeAng2
caeJap1
caeJap4
caePb1
caePb2
caePb3
caeRem2
caeRem3
caeRem4
caeSp111
caeSp51
calAnn1
calJac1
calJac3
calJac4
calMil1
canFam1
canFam2
canFam3
canFam4
canFam5
canFam6
capCar1
carCri1
cavPor3
cb1
cb3
cb4
ce10
ce11
ce2
ce4
ce6
cerSim1
chaVoc2
cheMyd1
chlSab2
chlUnd1
choHof1
chrPic1
chrPic2
ci1
ci2
ci3
colLiv1
colStr1
corBra1
corCor1
cotJap2
criGri1
criGriChoV1
criGriChoV2

Database
cucCan1
danRer1
danRer10
danRer11
danRer2
danRer3
danRer4
danRer5
danRer6
danRer7
dasNov3
dipOrd1
dirImm1
dm1
dm2
dm3
dm6
dp2
dp3
droAna1
droAna2
droEre1
droGri1
droMoj1
droMoj2
droPer1
droSec1
droSim1
droSim2
droVir1
droVir2
droYak1
droYak2
eboVir3
echTel1
echTel2
egrGar1
enhLutNer1
equCab1
equCab2
equCab3
eriEur1
eriEur2
eurHel1
falChe1
falPer1
felCat3
felCat4
felCat5
felCat8
felCat9
ficAlb2

Database
fr1
fr2
fr3
fulGla1
gadMor1
galGal2
galGal3
galGal4
galGal5
galGal6
galVar1
gasAcu1
gavSte1
gbMeta
geoFor1
go
go080130
go140213
go150121
go180426
gorGor3
gorGor4
gorGor5
gorGor6
haeCon2
halAlb1
halLeu1
hetBac1
hetGla1
hetGla2
hg16
hg17
hg18
hg19
hg19Patch10
hg19Patch13
hg38
hg38Patch11
hgFixed
hgcentral
hs1
information_schema
latCha1
lepDis1
letCam1
loaLoa1
loxAfr3
macEug1
macEug2
macFas5
manPen1
melGal1

Database
melGal5
melHap1
melInc2
melUnd1
merNub1
mesUni1
micMur1
micMur2
mm10
mm39
mm5
mm6
mm7
mm8
mm9
monDom1
monDom4
monDom5
mpxvRivers
musFur1
myoLuc2
nanPar1
nasLar1
necAme1
neoSch1
nipNip1
nomLeu1
nomLeu2
nomLeu3
ochPri2
ochPri3
oncVol1
opiHoa1
oreNil1
oreNil2
oreNil3
ornAna1
ornAna2
oryCun2
oryLat2
otoGar3
oviAri1
oviAri3
oviAri4
panPan1
panPan2
panPan3
panRed1
panTro1
panTro2
panTro3
panTro4

Database
panTro5
panTro6
papAnu2
papAnu4
papHam1
pelCri1
pelSin1
performance_schema
petMar1
petMar2
petMar3
phaCar1
phaLep1
phoRub1
picPub1
ponAbe2
ponAbe3
priExs1
priPac1
priPac3
proCap1
proteins120806
proteins121210
proteins140122
proteins150225
proteins160229
proteins180404
proteome
pteGut1
pteVam1
pygAde1
pytBiv1
rheMac1
rheMac10
rheMac2
rheMac3
rheMac8
rhiRox1
rn3
rn4
rn5
rn6
rn7
sacCer1
sacCer2
sacCer3
saiBol1
sarHar1
serCan1
sorAra1
sorAra2
sp120323

Database
sp121210
sp140122
sp150225
sp160229
sp180404
speTri2
strCam1
strPur1
strPur2
strRat2
susScr11
susScr2
susScr3
sys
taeGut1
taeGut2
tarSyr1
tarSyr2
tauEry1
tetNig1
tetNig2
thaSir1
tinGut2
triMan1
triSpi1
triSui1
tupBel1
turTru2
tytAlb1
uniProt
vicPac1
vicPac2
visiGene
wuhCor1
xenLae2
xenTro1
xenTro10
xenTro2
xenTro3
xenTro7
xenTro9
zonAlb1

1.1.3 Connecting to hg19 and listing tables

```
hg19 <- dbConnect(MySQL(),user="genome", db="hg19",host="genome-mysql.cse.ucsc.edu")
allTables <- dbListTables(hg19)
length(allTables)
```

```
## [1] 12619
```

```
allTables[1:5]
```

```
## [1] "HInv"         "HInvGeneMrna" "acembly"      "acemblyClass" "acemblyPep"
```

1.1.4 Get dimensions of a specific table

```
dbListFields(hg19,"affyU133Plus2")
```

```
##  [1] "bin"         "matches"     "misMatches"  "repMatches"  "nCount"
##  [6] "qNumInsert"  "qBaseInsert" "tNumInsert"  "tBaseInsert" "strand"
## [11] "qName"       "qSize"       "qStart"      "qEnd"        "tName"
## [16] "tSize"       "tStart"      "tEnd"        "blockCount"  "blockSizes"
## [21] "qStarts"     "tStarts"
```

```
dbGetQuery(hg19, "select count(*) from affyU133Plus2")
```

count(*)
58463

1.1.5 Read from the table

```
affyData <- dbReadTable(hg19, "affyU133Plus2")
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 0 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 1 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 2 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 3 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 4 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 5 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 6 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 7 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 8 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 11 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 12 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 13 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 15 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 16 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 17 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 18 imported as
## numeric
```

```
head(affyData)
```

bin	matches	misMatches	repMatches	nCount	qNumInsert	qBaseInsert	tNumInsert	tBaseInsert	strand	qName	qSize	qStart	qEnd	tName	tSize	tStart	tEnd	blockCount	blockSizes	qStarts	tStarts
585	530	4	0	23	3	41	3	898	-	225	9[illegible]5	x_60	3	chr1	249	[illegible]	[illegible]	6	93,144,229	3[illegible]	[illegible],14454,14599,14968,15795
585	335	17	0	1099	67	9	1162	21	225	3[illegible]	5	x_35	48	chr1	249	[illegible]	[illegible]	37	73,375,71,	[illegible]	[illegible]
585	415	6	4	83	16	18	2	93	-	226	3[illegible]	18	x_42	74	chr1	249	[illegible]	[illegible]	690,10,32,	3[illegible]	[illegible]
585	466	7	0	68	21	42	3	5743	-	155	7[illegible]	348	s_48	34	chr1	249	[illegible]	[illegible]	99,352,286	[illegible]	[illegible]
585	518	14	0	167	10	38	1	29	-	231	8[illegible]	190	at_53	99	chr1	249	[illegible]	[illegible]	131,26,130	0,63[illegible]	[illegible]
585	468	5	0	14	0	0	0	0	-	236	8[illegible]	870	at_48	7	chr1	249	[illegible]	[illegible]	487,	0,	27542,

1.1.6 Select a specific subset

```
query<- dbSendQuery(hg19, "select * from affyU133Plus2 where misMatches between 1 and 3")
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 0 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 1 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 2 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 3 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 4 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 5 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 6 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 7 imported as
## numeric

## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 8 imported as
## numeric
```

```
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 11 imported as
## numeric
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 12 imported as
## numeric
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 13 imported as
## numeric
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 15 imported as
## numeric
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 16 imported as
## numeric
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 17 imported as
## numeric
## Warning in .local(conn, statement, ...): Unsigned INTEGER in col 18 imported as
## numeric
```

```
affyMis <- fetch(query); quantile(affyMis$rnisMatches)
```

```
##   0%  25%  50%  75% 100%
##   NA   NA   NA   NA   NA
```

```
affyMisSmall <- fetch(query, n = 10)
```

```
## Error in h(simpleError(msg, call)): error in evaluating the argument 'n' in selecting a method for fu
```

```
dbClearResult(query)
```

```
## [1] TRUE
```

```
dirn(affyMisSmall)
```

```
## Error in dirn(affyMisSmall): could not find function "dirn"
```

1.1.7 Don't forget to close the connection

```
dbDisconnect(hg19)
```

```
## [1] TRUE
```

1.2 Reading from HDF5

1.2.1 HDFS

Used for storing large data sets Supports storing a range of data types Heirarchical data format groups containing zero or more data sets and metadata - Have a group header with group name and list of attributes - Have a group symbol table with a list of objects in group datasets multidmensional array of data elements with metadata - Have a headerwith name, datatype, dataspace, and storage layout - Have a data array with the data lhttp://www.hdfgroup.org/

1.2.2 R HDF5 package

```
library(rhdf5)
created <- h5createFile("example.h5")
```

```
## file '/Volumes/LocalData/Developer/JHU-Data-Science/3. Getting and Cleaning Data/week2/example.h5' al
```

```
created
```

```
## [1] FALSE
```

1.2.3 Create groups

```
created <- h5createGroup("example.h5", "foo")
```

```
## Can not create group. Object with name 'foo' already exists.
```

```
created <- h5createGroup("example.h5", "baa")
```

```
## Can not create group. Object with name 'baa' already exists.
```

```
created <- h5createGroup("example.h5", "foo/foobaa")
```

```
## Can not create group. Object with name 'foo/foobaa' already exists.
```

```
h5ls("example.h5")
```

	group	name	otype	dclass	dim
0	/	baa	H5I_GROUP		
1	/	df	H5I_DATASET	COMPOUND	5
2	/	foo	H5I_GROUP		
3	/foo	A	H5I_DATASET	INTEGER	5 x 2
4	/foo	foobaa	H5I_GROUP		
5	/foo/foobaa	B	H5I_DATASET	FLOAT	5 x 2 x 2

1.2.4 Write to groups

```
A <- matrix(1:10,nr=5,nc=2)
h5write(A,"example.h5","foo/A")
B <- array(seq(0.1,2.0,by=0.1),dim = c(5,2,2))
attr(B,"scale") <- "liter"
```

```
h5write(B,"example.h5","foo/foobaa/B")
```

```
h5ls("example.h5")
```

	group	name	otype	dclass	dim
0	/	baa	H5I_GROUP		
1	/	df	H5I_DATASET	COMPOUND	5
2	/	foo	H5I_GROUP		
3	/foo	A	H5I_DATASET	INTEGER	5 x 2
4	/foo	foobaa	H5I_GROUP		
5	/foo/foobaa	B	H5I_DATASET	FLOAT	5 x 2 x 2

1.2.5 Write a data set

```
df <- data.frame(1L:5L, seq(0,1,length.out=5),c("ab","cde","fghi","a","s"),stringsAsFactors = FALSE)
```

```
h5write(df,"example.h5","df")
```

```
## Error in h5writeDataset.data.frame(obj, loc$H5Identifier, name, ...): Cannot write data.frame. Objec
```

```
h5ls("example.h5")
```

	group	name	otype	dclass	dim
0	/	baa	H5I_GROUP		
1	/	df	H5I_DATASET	COMPOUND	5
2	/	foo	H5I_GROUP		
3	/foo	A	H5I_DATASET	INTEGER	5 x 2
4	/foo	foobaa	H5I_GROUP		
5	/foo/foobaa	B	H5I_DATASET	FLOAT	5 x 2 x 2

1.2.6 Reading data

```
readA <- h5read("example.h5","foo/A")
readB <- h5read("example.h5","foo/foobaa/B")
readdf <- h5read("example.h5","df")
```

```
readA
```

```
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    8
## [4,]    4    9
## [5,]    5   10
```

```
readB
```

```
## , , 1
##
##      [,1] [,2]
## [1,]  0.1  0.6
## [2,]  0.2  0.7
## [3,]  0.3  0.8
## [4,]  0.4  0.9
## [5,]  0.5  1.0
##
## , , 2
##
##      [,1] [,2]
## [1,]  1.1  1.6
## [2,]  1.2  1.7
## [3,]  1.3  1.8
## [4,]  1.4  1.9
## [5,]  1.5  2.0
```

```
readdf
```

X1L.5L	seq.0..1..length.out...5.	c..ab....cde....fghi....a....s..
1	0.00	ab
2	0.25	cde
3	0.50	fghi
4	0.75	a
5	1.00	s

1.2.7 Writing and reading chunks

```
h5write(c(12,13,14),"example.h5","foo/A",index=list(1:3,1))
```

```
h5read("example.h5","foo/A")
```

```
##      [,1] [,2]
## [1,]   12    6
## [2,]   13    7
## [3,]   14    8
## [4,]    4    9
## [5,]    5   10
```

1.3 Reading from The Web

1.3.1 Webscraping

Webscraping: Programatically extracting data from the HTML code of websites. It can be a great way to get data How Netflix reverse engineered Hollywood Many websites have information you may want to programaticaly read In some cases this is against the terms of service for the website Attempting to read too many pages too quickly can get your IP address blocked

1.3.2 Getting data off webpages - readlinesO

```
con <- url("https://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en")
htmlCode <- readLines(con)
```

```
## Warning in readLines(con): incomplete final line found on
## 'https://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en'
```

```
close(con)
htmlCode
```

```
##  [1] "<!doctype html><html><head><title>Jeff Leek - Google Scholar</title><meta http-equiv=\"Content-
##  [2] ""
##  [3] " Copyright The Closure Library Authors."
##  [4] " SPDX-License-Identifier: Apache-2.0"
##  [5] "*/"
##  [6] "var ba=\"function\"==typeof Object.create?Object.create:function(a){var b=function(){};b.proto
##  [7] "},ea=da(this),g=function(a,b){if(b)a:{var c=ea;a=a.split(\".\");for(var d=0;d<a.length-1;d++){
##  [8] "var ka=fa,la=function(a,b){a.prototype=ba(b.prototype);a.prototype.constructor=a;if(ka)ka(a,b)
##  [9] "var ma=function(a){a=Math.trunc(a)||0;0>a&&(a+=this.length);if(!(0>a||a>=this.length))return t
## [10] "g(\"String.prototype.at\",function(a){return a?a:ma});var na=function(a,b){var c=Array.prototy
## [11] "ua=va?0<+va[1]:x(\"Android\")?!0:window.matchMedia&&window.matchMedia(\"(pointer)\").matches?w
## [12] "function Fa(a){var b=[];a=a.elements;for(var c=a.length,d=0;d<c;d++){var e=a[d],f=encodeURICom
## [13] "function Ka(a){if(!A(\"gs_hats\")){var b=document.createElement(Aa.aa);b.id=\"gs_hats\";b.src=
## [14] "var Pa=function(a){var b=a.o,c=b.length;a=a.I;for(var d=0,e=0;e<c;e++){var f=b[e];f&&(b[d]=f,a
## [15] "function Za(a){return(a.ctrlKey?1:0)|(a.altKey?2:0)|(a.metaKey?4:0)|(a.shiftKey?8:0)}function
## [16] "var Ua,bb=!!document.attachEvent,cb=document.readyState;if(bb?\"complete\"!=cb:\"loading\"==cb
## [17] "function gb(a,b,c,d,e,f){f=void 0===f?\"\":f;var h=Ca(a),k=hb(h);if(!k||k<hb(f)){var m=documen
## [18] "function kb(a){ib((hb(a)||1E6)-1,!1)}function lb(a){a=void 0===a?!1:a;J.pop()(a)}function ib(a
## [19] "C(document,\"focus\",function(a){var b=J.length;if(b)for(var c=jb(a.target);c<b;){var d=\"\",e
## [20] "function wb(a){var b={};a=a.split(\"&\");for(var c=0;c<a.length;c++){var d=a[c],e=d.indexOf(\"=
## [21] "function tb(a){var b=xb(window.location.hash);zb(b,a);return b}function sb(a){var b=L||K(windo
## [22] "if(\"undefined\"==typeof GSP)Eb=!1;else{var Fb=.001*Date.now(),Gb=GSP.eventId,Hb=!1,M,Ib=ob;if
## [23] "var Sb=Eb;\"onpageshow\"in window?C(window,\"pageshow\",Ab):H(Ab);C(window,vb?\"popstate\":\"ha
```

```
## [24] "function bc(a){a&&(a.onreadystatechange=function(){},a.abort())};var cc=function(a,b,c){this.ty
## [25] "function Q(a,b,c){var d=mc;\"string\"===typeof b&&(nc[0]=b,b=nc);var e=b.length;a=gc(a);for(var
## [26] "function mc(a,b,c,d){var e=pc[c];e||(\"touchstart\"!=c&&\"mouseover\"!=c&&\"mouseout\"!=c&&C(do
## [27] "function lc(a){for(var b=a.target;b&&b!=document&&!b.disabled&&!u(b,\"gs_dis\");){a.currentTarg
## [28] "function kc(a,b,c){a:{for(var d=c.currentTarget;d&&d!=document;){var e=sc(a,d);if(e){a=tc(e,d)
## [29] "function xc(a,b,c,d){c=(d=(c=hc[c])&&c[d])?d.length:0;for(var e=0;e<c;e++){var f=d[e];!(f===a|
## [30] "function Dc(a,b,c,d,e){for(var f;b&&a;){if(c(b)){if(e)return b}else for(f=Ac(b,d);f;f=Bc(f,d))
## [31] "var Qc=function(a){for(var b=U;b.j>a;)kb(b.top().i)},Rc=function(a,b){for(var c=0;c<a.l.length&
## [32] "function V(a,b,c,d){b=void 0===b?\"\":b;c=void 0===c?\"\":c;d=void 0===d?\"\":d;var e=U.top();
## [33] "function Zc(){var a=A(\"gs_top\"),b=document.documentElement;a=a.scrollHeight>b.clientHeight;fo
## [34] "function Uc(){function a(){var w=c.clientHeight,y=+E.getAttribute(\"data-h\");y||(h.style.maxHe
## [35] "\"\",E=Yc(f),T=window.pageYOffset,Wc=D&&\"#\"!=D[0]&&!l,Xc=0<U.j?U.at(U.j-1).i:\"\",ta=!!I[e];W
## [36] "(Kc.removeListener(a),E&&r(E,\"gs_vis\"),r(f,\"gs_vis\"),r(f,\"gs_abt\"));for(var w=U.top()?U.
## [37] "Xc);Wc&&(bc(b.m),b.m=null,Oc(b,ac(D,t,function(w,y,z){b.m=null;z=(w=200==w&&z.match(/^text\\/ht
## [38] "function ad(a,b,c,d){v(b,\"gs_md_ldg\",!1);for(var e=b.querySelectorAll(\"[data-duid]\"),f=e.le
## [39] "function cd(a){if(a=document.querySelector(\"#\"+a+\">.gs_md_bdy\"))a.scrollTop=a.scrollLeft=0
## [40] "function qb(){var a=U.top();return{d:a&&a.i||void 0,u:a&&a.F||void 0,p:a&&a.D?\"1\":void 0,t:a&
## [41] "Cb.add(function(){var a=xb(window.location.hash),b=a.d||\"\",c=b?A(b):null;++dd;if(c){var d=a.u
## [42] "e=Date.now()),c!=!!h&&(d=h=\"\",f=gd()),b=new Nc(b,d,h,f,a,e),c=U,e=b.C,a=Rc(c,e),e=a<c.l.leng
## [43] "jd.prototype.pa=function(a){var b=this;N(a);if((a=this.J)&&!this.m){var c=\"json=&\"+Fa(a);kd(
## [44] "f.elements;k=k.E;\"object\"==typeof k||(k=Object.create(null));for(var l in b.V){var t=h[l],D=
## [45] "var kd=function(a,b){a=a.J;var c=a.getAttribute(\"data-bsel\");a=c?document.querySelectorAll(c)
## [46] "c.appendChild(b);Tb();v(a,\"gs_fm_s\",!0);Vb()}};ec(jd,[new O(\".gs_ajax_frm\",{submit:jd.proto
## [47] "C(window,\"pageshow\",function(a){a.persisted&&(od=\"&bn=1\",sd())});"
## [48] "C(document,nd,function(a){if(!(\"click\"==a.type&&a.button||\"mouseup\"==a.type&&1!=a.button))
## [49] "!1,!0);Q(\".gs_fm_s\",\"click\",function(a){a=a.currentTarget.getAttribute(\"data-fm\")||\"\";
## [50] "W.prototype.da=function(a){var b=a.g.keyCode;if(38==b||40==b)N(a),this.open(38==b)};W.prototype
## [51] "var ud=function(a,b){var c=b.currentTarget,d=A(a.ra),e=a.T();c!=e&&(d.value=c.getAttribute(\"da
## [52] "ec(wd,[new O(\".gs_md_ulr\",{}),new O(\".gs_md_li\",{keydown:wd.prototype.U})]);Q(\"#gs_hdr_mnu
## [53] "Q(\"#gs_hdr_tsi\",[\"focus\",\"blur\"],function(a){function b(){var h=d.getBoundingClientRect()
## [54] "Q(\"#gs_hdr_tsc\",\"mousedown\",function(a){N(a);var b=A(\"gs_hdr_tsi\");b.value=\"\";b.focus()
## [55] "var Ad=function(a){a=a.g.keyCode;return 32==a||13==a},zd=function(a){R(\"gs-press\",a.currentTa
## [56] "\"mouseup\",\"click\"],Gd=x(\"Android\")&&!x(\"Chrome\"),Hd=0,Id=0,Jd=[\"touchstart\",\"mousedo
## [57] "function Ud(){Od&&Md(Sd.getBoundingClientRect())}var Qd,Td,Sd,Nd,Od=!1,Pd;H(function(){if(Td=A
## [58] "function Wd(a){var b=a();b.triggerId&&Ka(function(){return Vd(b.triggerId,b.sa)})}function Xd(a
## [59] "function ne(){ke();me(!1);for(var a=A(\"gsc_cods_res\").querySelectorAll(\".gsc_ccb_ck\"),b=a.l
## [60] "function se(a){(a=a.currentTarget.getAttribute(\"data-a\"))&&he(a)}"
## [61] "function me(a){var b=A(\"gsc_cods_frm\");if(b){b=b.elements;var c=b[1];b[0].disabled=c.disable
## [62] "b||\"\")}a&&(te=+a.getAttribute(\"data-max\")||0,b=A(\"gsc_cods_save\"),Ga(b,\"xsrf\").value=a
## [63] "function oe(){var a=0<Y.A||0<Z.A,b=de(),c=0>b;b=0>=b;A(\"gsc_cod_done\").disabled=!a||c;v(A(\"g
## [64] "function xe(){var a=A(\"gsc_cods_save\");Ga(a,\"colleague_add\").value=pa(Y.v).join(\",\");Ga(a
## [65] "var Ge=[];function He(a){N(a);var b=a.currentTarget;a=b.href;var c=b.getAttribute(\"data-eid\")
## [66] "function Je(a){var b=A(\"gsc_a_tr0\"),c=A(\"gsc_a_trh\");b=b.querySelector(\".gsc_a_t\");c.quer
## [67] "e.innerHTML=f;b.disabled=!h.N}else Ce(2)})}var Me=\"\",Ne=0,Le=0;function Oe(){var a=window.loc
## [68] "function Se(a){a.g&&a.g.preventDefault();a=A(\"gsc_fol_f\");var b=A(\"gsc_fol_inp\");b.innerHTM
## [69] "function Ye(){P([\"#gsc_coauth_opn\",\".gsc_rsb_btne\",\".gsc_rsb_btnv\"],\"click\",Ve);P(\".gs
## [70] "function gf(a){(\"keydown\"!=a.type||13==a.g.keyCode&&a.g&&!Za(a.g))&&ff()}function hf(){A(\"gs
## [71] "\"gs-md-ldin\",ye);Q(\"#gsc_md_cbyd\",\"gs-md-lded\",ze);P(\"#gsc_md_cbym\",[\"gs-md-ldin\",\"g
## [72] "}({\"customAC\":0,\"eventId\":\"qp-aZK_JK6zGsQK5wLfQBw\"});</script></head><body><div id=\"gs_f
```

1.3.3 Parsing with XML

```
library(XML)
```

```
html <- htmlTreeParse(htmlCode, useInternalNodes = T)

xpathSApply(html,"//title",xmlValue)
```

```
## [1] "Jeff Leek - Google Scholar"
```

```
xpathSApply(html,"//td[@id='col-citedby']",xmlValue)
```

```
## list()
```

1.3.4 GET from the httr package

```
library(httr)
```

```
html2 = GET(url)
```

```
## Error in as.character(url): cannot coerce type 'closure' to vector of type 'character'
```

```
content2 <- content(html2,as="text")
```

```
## Error in eval(expr, envir, enclos): object 'html2' not found
```

```
parsedHTML <- htmlParse(content2,asText = TRUE)
```

```
## Error in eval(expr, envir, enclos): object 'content2' not found
```

```
xpathSApply(parsedHTML,"//title",xmlValue)
```

```
## Error in eval(expr, envir, enclos): object 'parsedHTML' not found
```

1.3.5 Accessing websites with passwords

```
pg1 = GET("http://httpbin.org/basic-auth/user/passwd")
pg1
```

```
## Response [http://httpbin.org/basic-auth/user/passwd]
##   Date: 2023-06-27 08:37
##   Status: 401
##   Content-Type: <unknown>
## <EMPTY BODY>
```

```
pg2 = GET("http://httpbin.org/basic-auth/user/passwd",
authenticate( "user", "passwd"))
pg2
```

```
## Response [http://httpbin.org/basic-auth/user/passwd]
##   Date: 2023-06-27 08:37
##   Status: 200
##   Content-Type: application/json
##   Size: 47 B
## {
##   "authenticated": true,
##   "user": "user"
## }
```

1.3.6 Using handles

```
google = handle("http://google.com")
pgl = GET(handle=google,path="/")
pg2 = GET(handle=google,path="search")
```

1.4 Reading From APIs

In general look at the documentation httr allows GET, POST, PUT, DELETE requests if you are authorized You can authenticate with a user name or a password Most modern APIs use something like oauth httr works well with Facebook, Google, Twitter, Githb, etc.

Authenticate -> Connect -> retrive -> convert JSON -> consume

1.5 Reading From Other Sources

1.5.1 There is a package for that

Roger has a nice video on how there are R packages for most things that you will want to access. Here l1m going to briefly review a few useful packages In general the best way to find out if the R package exists is to Google 11data storage mechanism R package11 - For example: 11MySQL R package11

1.5.2 Interactingm ore directlyw ith files

file - open a connection to a text file url - open a connection to a url gzfile - open a connection to a .gz file bzfile - open a connection to a .bz2 file ?connections for more information Remember to close connections

1.5.3 foreign package

Loads data from Minitab, S, SAS, SPSS, Stata,Systat Basic functions read.too - read.arff (Weka) - read.dta (Stata) - read.mtp (Minitab) - read.octave (Octave) - read.spss (SPSS) - read.xport (SAS) See the help page for more details lhttp://cran.r-project.org/web/packages/foreign/foreign.pdf

1.5.4 Examples of other database packages

RPostresSQL provides a DBI-compliant database Tutorial-https://code.google.com/p/rpostgre~, help project.org/web/packages/RPostgreSQURPostgreSQL.pdf connection from R. file-[nttp://cra[B RODBC provides interfaces to multiple databases including PostgreQL, MySQL, Microsoft Access and SQLite. Tutorial - ffi!p://cran.r-project.org/web/packages/RODBC/vignettes/RODBC.pdf, help file - http://cran.r-project.org/web/packages/RODBC/RODBC.pcflj RM on go lhttp://cran.r-project.org/web/packages/RMongo/RMongo.pd~ (example of Rmongo http://www.r-bloggers.com/r-and-mongodb/) and rmongodbl provide interfaces to MongoDb.

1.5.5 Reading images

jpeg - ffi!p://cran.r-project.org/web/packages/jpeg/index.html readbitmap - lhttp://cran.r-project.org/web/packages/readbit
png - http://cran.r-project.org/web/packages/png/i ndex. htm I EBlmage (Bioconductor) - http://www.bioconductor.org/packages/2.13/bioc/html/EBlmage.htmll

1.5.6 Reading GIS data

rd gal - http://cran.r-project.org/web/packages/rgdal/index. hTm] rgeos - lhttp://cran.r-project.org/web/packages/rgeos/index.htmll raster - [nttp://cran.r-project.org/web/packages/raster/index.html

1.5.7 Reading music data

tu neR - lfittp: // cran.r -project.org/web/packages/tun ~ seewave - http://rug.mnhn.fr/seewave