Reading: CheatSheet - Creating Database Objects and Querying Data from R

Command	Syntax	Description	Example
sqlQuery()	`sqlQuery(conn, _QUERY_, errors=FALSE)`	`sqlQuery` function submits an SQL query that is passed as the second parameter to an ODBC database connection, and retrieve the results. conn - connection _QUERY_ - The query you want to execute on the database errors - logical: if true halt and display error, else return -1.	**Code** `sqlQuery(conn, "SELECT COUNT(B_ID) FROM BOARD;")` **Output** `sqlQuery(conn, "SELECT COUNT(B_ID) FROM BOARD;")` A data.frame: 1 × 1 1 <int> 1 76
sqlTables()	`sqlTables(conn, schema = schemaname)`	`sqlTables` function list the table-like objects accessible from an ODBC connection.	**Code** `sqlTables(conn)` **Output** `sqlTables(conn)` A data.frame: 476 × 5 TABLE_CAT TABLE_SCHEM TABLE_NAME TABLE_TYPE REMARKS <chr> <chr> <chr> <chr> <chr> 1 SYSPUBLIC DUAL ALIAS 2 SYSIBM SYSATTRIBUTES SYSTEM TABLE 3 SYSIBM SYSAUDITEXCEPTIONS SYSTEM TABLE 4 SYSIBM SYSAUDITPOLICIES SYSTEM TABLE 5 SYSIBM SYSAUDITUSE SYSTEM TABLE 6 SYSIBM SYSBUFFERPOOLNODES SYSTEM TABLE 7 SYSIBM SYSBUFFERPOOLS SYSTEM TABLE 8 SYSIBM SYSCHECKS SYSTEM TABLE 9 SYSIBM SYSCODEPROPERTIES SYSTEM TABLE
sqlTypeInfo()	**Code** `sqlTypeInfo(conn)`	`sqlTypeInfo` function request information about column types in an ODBC database.	**Code** `sqlTypeInfo(conn)` **Output** `sqlTypeInfo(conn)` TYPE_NAME DATA_TYPE COLUMN_SIZE LITERAL_PREFIX LITERAL_SUFFIX CREATE_PARAMS NULLABI <chr> <int> <int> <chr> <chr> <chr> <int> 1 XML -370 0 2 DECFLOAT -360 34 PRECISION 3 CLOB -99 2147483647 ' ' LENGTH 4 BLOB -98 2147483647 BLOB(X' ') LENGTH 5 LONG VARCHAR FOR BIT DATA -4 32700 ' ' LENGTH 6 VARBINARY -3 32672 VARBINARY(X' ') LENGTH 7 BINARY -2 255 BINARY(X' ') LENGTH
odbcGetInfo()	**Code** `odbcGetInfo(conn)`	`odbcGetInfo` function requests information on an ODBC connection.	**Code** `odbcGetInfo(conn)` **Output** `odbcGetInfo(conn)` **DBMS_Name:** 'DB2/LINUXX8664' **DBMS_Ver:** '11.01.0404' **Driver_ODBC_Ver:** '03.51' **Data_Source_Name:** '' **Driver_Name:** 'libdb2.a' **Driver_Ver:** '11.05.0600' **ODBC_Ver:** '03.52' **Server_Name:** 'DB2'

nrow	`nrow(x)`	`nrow` returns the number of rows present in x.	**Code**
			`ma <- matrix(1:12, 3, 4)`
			`nrow(ma)`
			Output
			`ma <- matrix(1:12, 3, 4)`
			`nrow(ma)`
			3

ncol	`ncol(x)`	`ncol` returns the number of columns present in x.	**Code**
			`ma <- matrix(1:12, 3, 4)`
			`ncol(ma)`
			Output
			`ma <- matrix(1:12, 3, 4)`
			`ncol(ma)`
			4

paste	`paste(..., sep = " ", collapse = NULL)`	`paste` concatenates vectors after converting to them character.	**Code**
			`secquery <- paste("select s.enrollment as ENROLLMENT from school s, board b where b.b_name = 'Toronto DSB' and b.b_id=s.b_id and s.level = 'Secondary' order by enrollment desc")`
			Output
			`secquery <- paste("select s.enrollment as ENROLLMENT from school s, board b where b.b_name = 'Toronto DSB' and b.b_id=s.b_id and s.level = 'Secondary' order by enrollment desc")`
			`secdf <- sqlQuery(conn, secquery)`
			`head(secdf)`
			A data.frame: 6 × 1
			ENROLLMENT
			<int>
			1 NA
			2 NA
			3 0
			4 0
			5 2209
			6 1993

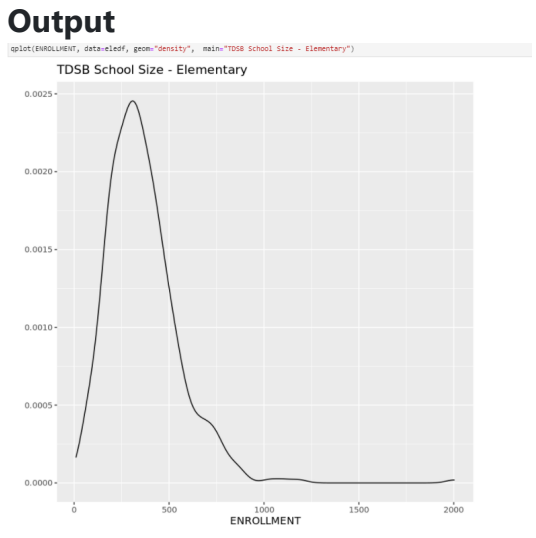
cat	`cat(...,file = “”, sep = ” “, fill = FALSE, labels = NULL, append = FALSE)`	`cat` Returns the concatenated objects passed as parameters	**Code**
			`cat('one',2,'three',4,'five') #one 2 three 4 five`
			Output
			`cat('one',2,'three',4,'five')`
			one 2 three 4 five

odbcGetErrMsg()	`odbcGetErrMsg(conn)`	`odbcGetErrMsg` returns a (possibly zero-length) character vector of pending messages.	

iconv()	`iconv(x, from = "", to = "", sub = NA, mark = TRUE, toRaw = FALSE)`	`iconv` uses system facilities to convert a character vector between encodings: the 'i' stands for internationalization.	**Code**
			`x <- "fa\xE7ile"`
			`iconv(x, "latin1", "ASCII", "?")`
			Output
			`odbcGetErrMsg(conn)`
			`x <- "fa\xE7ile"`
			`iconv(x, "latin1", "ASCII", "?")`
			'fa?ile'

sqlSave()	`sqlSave(conn, dat, tablename = NULL, append = FALSE,rownames = TRUE, colnames= FALSE, verbose = FALSE,safer = TRUE, addPK = FALSE, typeInfo, varTypes,fast = TRUE, test = FALSE, nastring = NULL)`	`sqlSave` writes or updates a table in an ODBC database.	**Code** `sqlSave(conn, boarddf, "BOARD", append=TRUE, fast=FALSE, rownames=FALSE, colnames=FALSE, verbose=FALSE)` **Output** 1 B28010 Algoma DSB Public English 2 B67202 Algonquin and Lakeshore CDSB Roman Catholic English 3 B66010 Avon Maitland DSB Public English 4 B66001 Bluewater DSB Public English 5 B67164 Brant Haldimand Norfolk CDSB Roman Catholic English 6 B67008 Bruce-Grey CDSB Roman Catholic English 7 B67172 CDSB of Eastern Ontario Roman Catholic English 8 B66303 CS Viamonde Public French 9 B67300 CSD des coles catholiques du Sud-Ouest Roman Catholic French 10 B28118 CSD du Grand Nord de l'Ontario Public French 11 B28100 CSD du Nord-Est de l'Ontario Public French 12 B67318 CSDC Centre-Sud Roman Catholic French 13 B29114 CSDC Franco-Nord Roman Catholic French 14 B67326 CSDC de l'Est ontarien Roman Catholic French

sqlFetch()	`sqlFetch(conn, sqtable, ..., colnames = FALSE, rownames = TRUE)`	`sqlFetch` Reads some or all of a table from an ODBC database into a dataframe.	**Code**
			`sqlFetch(conn, "BOARD")`
			Output
			`boarddb <- sqlFetch(conn, "BOARD")`
			`head(boarddb)`
			A data.frame: 6 × 4
			B_ID B_NAME TYPE LANGUAGE
			<fct> <fct> <fct> <fct>
			1 B28010 Algoma DSB Public English
			2 B67202 Algonquin and Lakeshore CDSB Roman Catholic English
			3 B66010 Avon Maitland DSB Public English
			4 B66001 Bluewater DSB Public English
			5 B67164 Brant Haldimand Norfolk CDSB Roman Catholic English
			6 B67008 Bruce-Grey CDSB Roman Catholic English

qplot()	`qplot(x,y,...,data,facets = NULL,margins = FALSE,geom = "auto", xlim = c(NA, NA),ylim = c(NA, NA),log = "",main = NULL,xlab = NULL,ylab = NULL,asp = NA, stat = NULL, position = NULL)`	`qplot()` is a shortcut designed to be familiar if you're used to base plot(). It's a convenient wrapper for creating a number of different types of plots using a consistent calling scheme.	**Code** `qplot(ENROLLMENT, data=eledf, geom="density", main="TDSB School Size - Elementary")` **Output** 

Author(s)

Malika Singla

Contributor(s)

DM Naidu

Changelog

Date	Version	Changed by	Change Description
2020-08-12	1.1	DM Naidu	Added output images
2020-07-31	1.0	Malika Singla	Initial Version