2.3. pyspark.sql.functions

A collections of builtin functions

2.3.1. Functions

`abs`(col)	Computes the absolute value.
`acos`(col)	Computes the cosine inverse of the given value; the returnec
`add_months`(start, months)	Returns the date that is *months* months after *start*
`approxCountDistinct`(col[, rsd])	Returns a new `Column` for approximate distinct count of `col`
`array`(*cols)	Creates a new array column.
`array_contains`(col, value)	Collection function: returns True if the array contains the giv
`asc`(col)	Returns a sort expression based on the ascending order of th
`ascii`(col)	Computes the numeric value of the first character of the stri
`asin`(col)	Computes the sine inverse of the given value; the returned a
`atan`(col)	Computes the tangent inverse of the given value.
`atan2`(col1, col2)	Returns the angle theta from the conversion of rectangular c
`avg`(col)	Aggregate function: returns the average of the values in a gr
`base64`(col)	Computes the BASE64 encoding of a binary column and retu
`bin`(col)	Returns the string representation of the binary value of the g

`bitwiseNOT`(col)	Computes bitwise not.
`broadcast`(df)	Marks a DataFrame as small enough for use in broadcast join
`bround`(col[, scale])	Round the given value to *scale* decimal places using HALF_EV
`cbrt`(col)	Computes the cube-root of the given value.
`ceil`(col)	Computes the ceiling of the given value.
`coalesce`(*cols)	Returns the first column that is not null.
`col`(col)	Returns a `Column` based on the given column name.
`collect_list`(col)	Aggregate function: returns a list of objects with duplicates.
`collect_set`(col)	Aggregate function: returns a set of objects with duplicate el
`column`(col)	Returns a `Column` based on the given column name.
`concat`(*cols)	Concatenates multiple input string columns together into a s
`concat_ws`(sep, *cols)	Concatenates multiple input string columns together into a s
`conv`(col, fromBase, toBase)	Convert a number in a string column from one base to anoth
`corr`(col1, col2)	Returns a new `Column` for the Pearson Correlation Coefficie
`cos`(col)	Computes the cosine of the given value.
`cosh`(col)	Computes the hyperbolic cosine of the given value.
`count`(col)	Aggregate function: returns the number of items in a group.
`countDistinct`(col, *cols)	Returns a new `Column` for distinct count of `col` or `cols`.
`covar_pop`(col1, col2)	Returns a new `Column` for the population covariance of `col1`
`covar_samp`(col1, col2)	Returns a new `Column` for the sample covariance of `col1` an
`crc32`(col)	Calculates the cyclic redundancy check value (CRC32) of a bi
`create_map`(*cols)	Creates a new map column.

`cume_dist` ()	Window function: returns the cumulative distribution of valu
`current_date` ()	Returns the current date as a date column.
`current_timestamp` ()	Returns the current timestamp as a timestamp column.
`date_add` (start, days)	Returns the date that is *days* days after *start*
`date_format` (date, format)	Converts a date/timestamp/string to a value of string in the f
`date_sub` (start, days)	Returns the date that is *days* days before *start*
`datediff` (end, start)	Returns the number of days from *start* to *end*.
`dayofmonth` (col)	Extract the day of the month of a given date as integer.
`dayofyear` (col)	Extract the day of the year of a given date as integer.
`decode` (col, charset)	Computes the first argument into a string from a binary using
`dense_rank` ()	Window function: returns the rank of rows within a window
`desc` (col)	Returns a sort expression based on the descending order of t
`encode` (col, charset)	Computes the first argument into a binary from a string using
`exp` (col)	Computes the exponential of the given value.
`explode` (col)	Returns a new row for each element in the given array or ma
`expm1` (col)	Computes the exponential of the given value minus one.
`expr` (str)	Parses the expression string into the column that it represen
`factorial` (col)	Computes the factorial of the given value.
`first` (col[, ignorenulls])	Aggregate function: returns the first value in a group.
`floor` (col)	Computes the floor of the given value.
`format_number` (col, d)	Formats the number X to a format like '#,–#,–#.–', rounded to
`format_string` (format, *cols)	Formats the arguments in printf-style and returns the result

`from_unixtime` (timestamp[, format])	Converts the number of seconds from unix epoch (1970-01-
`from_utc_timestamp` (timestamp, tz)	Assumes given timestamp is UTC and converts to given time
`get_json_object` (col, path)	Extracts json object from a json string based on json path spe
`greatest` (*cols)	Returns the greatest value of the list of column names, skipp
`grouping` (col)	Aggregate function: indicates whether a specified column in
`grouping_id` (*cols)	Aggregate function: returns the level of grouping, equals to
`hash` (*cols)	Calculates the hash code of given columns, and returns the re
`hex` (col)	Computes hex value of the given column, which could be Stri
`hour` (col)	Extract the hours of a given date as integer.
`hypot` (col1, col2)	Computes *sqrt(a^2 + b^2)* without intermediate overflow or
`ignore_unicode_prefix` (f)	Ignore the 'u' prefix of string in doc tests, to make it works
`initcap` (col)	Translate the first letter of each word to upper case in the se
`input_file_name` ()	Creates a string column for the file name of the current Sparl
`instr` (str, substr)	Locate the position of the first occurrence of substr column i
`isnan` (col)	An expression that returns true iff the column is NaN.
`isnull` (col)	An expression that returns true iff the column is null.
`json_tuple` (col, *fields)	Creates a new row for a json column according to the given fi
`kurtosis` (col)	Aggregate function: returns the kurtosis of the values in a gr
`lag` (col[, count, default])	Window function: returns the value that is *offset* rows befor
`last` (col[, ignorenulls])	Aggregate function: returns the last value in a group.
`last_day` (date)	Returns the last day of the month which the given date belon
`lead` (col[, count, default])	Window function: returns the value that is *offset* rows after

`least` (*cols)	Returns the least value of the list of column names, skipping
`length` (col)	Calculates the length of a string or binary expression.
`levenshtein` (left, right)	Computes the Levenshtein distance of the two given strings.
`lit` (col)	Creates a `Column` of literal value.
`locate` (substr, str[, pos])	Locate the position of the first occurrence of substr in a strin
`log` (arg1[, arg2])	Returns the first argument-based logarithm of the second ar
`log10` (col)	Computes the logarithm of the given value in Base 10.
`log1p` (col)	Computes the natural logarithm of the given value plus one.
`log2` (col)	Returns the base-2 logarithm of the argument.
`lower` (col)	Converts a string column to lower case.
`lpad` (col, len, pad)	Left-pad the string column to width *len* with *pad*.
`ltrim` (col)	Trim the spaces from left end for the specified string value.
`max` (col)	Aggregate function: returns the maximum value of the expre
`md5` (col)	Calculates the MD5 digest and returns the value as a 32 char
`mean` (col)	Aggregate function: returns the average of the values in a gr
`min` (col)	Aggregate function: returns the minimum value of the expres
`minute` (col)	Extract the minutes of a given date as integer.
`monotonically_increasing_id` ()	A column that generates monotonically increasing 64-bit int
`month` (col)	Extract the month of a given date as integer.
`months_between` (date1, date2)	Returns the number of months between date1 and date2.
`nanvl` (col1, col2)	Returns col1 if it is not NaN, or col2 if col1 is NaN.
`next_day` (date, dayOfWeek)	Returns the first date which is later than the value of the dat

`ntile` (n)	Window function: returns the ntile group id (from 1 to *n* inclu
`percent_rank` ()	Window function: returns the relative rank (i.e.
`posexplode` (col)	Returns a new row for each element with position in the give
`pow` (col1, col2)	Returns the value of the first argument raised to the power o
`quarter` (col)	Extract the quarter of a given date as integer.
`rand` ([seed])	Generates a random column with i.i.d.
`randn` ([seed])	Generates a column with i.i.d.
`rank` ()	Window function: returns the rank of rows within a window
`regexp_extract` (str, pattern, idx)	Extract a specific(idx) group identified by a java regex, from t
`regexp_replace` (str, pattern, replacement)	Replace all substrings of the specified string value that matcl
`repeat` (col, n)	Repeats a string column n times, and returns it as a new strin
`reverse` (col)	Reverses the string column and returns it as a new string colu
`rint` (col)	Returns the double value that is closest in value to the argum
`round` (col[, scale])	Round the given value to *scale* decimal places using HALF_U
`row_number` ()	Window function: returns a sequential number starting at 1
`rpad` (col, len, pad)	Right-pad the string column to width *len* with *pad*.
`rtrim` (col)	Trim the spaces from right end for the specified string value.
`second` (col)	Extract the seconds of a given date as integer.
`sha1` (col)	Returns the hex string result of SHA-1.
`sha2` (col, numBits)	Returns the hex string result of SHA-2 family of hash functio
`shiftLeft` (col, numBits)	Shift the given value numBits left.
`shiftRight` (col, numBits)	Shift the given value numBits right.

`shiftRightUnsigned` (col, numBits)	Unsigned shift the given value numBits right.
`signum` (col)	Computes the signum of the given value.
`sin` (col)	Computes the sine of the given value.
`since` (version)	A decorator that annotates a function to append the version
`sinh` (col)	Computes the hyperbolic sine of the given value.
`size` (col)	Collection function: returns the length of the array or map st
`skewness` (col)	Aggregate function: returns the skewness of the values in a g
`sort_array` (col[, asc])	Collection function: sorts the input array for the given colum
`soundex` (col)	Returns the SoundEx encoding for a string
`spark_partition_id` ()	A column for partition ID of the Spark task.
`split` (str, pattern)	Splits str around pattern (pattern is a regular expression).
`sqrt` (col)	Computes the square root of the specified float value.
`stddev` (col)	Aggregate function: returns the unbiased sample standard d
`stddev_pop` (col)	Aggregate function: returns population standard deviation o
`stddev_samp` (col)	Aggregate function: returns the unbiased sample standard d
`struct` (*cols)	Creates a new struct column.
`substring` (str, pos, len)	Substring starts at *pos* and is of length *len* when str is String t
`substring_index` (str, delim, count)	Returns the substring from string str before count occurrenc
`sum` (col)	Aggregate function: returns the sum of all values in the expre
`sumDistinct` (col)	Aggregate function: returns the sum of distinct values in the
`tan` (col)	Computes the tangent of the given value.
`tanh` (col)	Computes the hyperbolic tangent of the given value.

`toDegrees` (col)	Converts an angle measured in radians to an approximately e
`toRadians` (col)	Converts an angle measured in degrees to an approximately
`to_date` (col)	Converts the column of StringType or TimestampType into D
`to_utc_timestamp` (timestamp, tz)	Assumes given timestamp is in given timezone and converts t
`translate` (srcCol, matching, replace)	A function translate any character in the *srcCol* by a characte
`trim` (col)	Trim the spaces from both ends for the specified string colum
`trunc` (date, format)	Returns date truncated to the unit specified by the format.
`udf` (f[, returnType])	Creates a `Column` expression representing a user defined fur
`unbase64` (col)	Decodes a BASE64 encoded string column and returns it as a
`unhex` (col)	Inverse of hex.
`unix_timestamp` ([timestamp, format])	Convert time string with given pattern ('yyyy-MM-dd HH:mm
`upper` (col)	Converts a string column to upper case.
`v` (name[, doc])	Create a binary mathfunction by name
`var_pop` (col)	Aggregate function: returns the population variance of the va
`var_samp` (col)	Aggregate function: returns the unbiased variance of the valu
`variance` (col)	Aggregate function: returns the population variance of the va
`weekofyear` (col)	Extract the week number of a given date as integer.
`when` (condition, value)	Evaluates a list of conditions and returns one of multiple pos
`window` (timeColumn, windowDuration[, ...])	Bucketize rows into one or more time windows given a times
`year` (col)	Extract the year of a given date as integer.

2.3.2. Classes

`AutoBatchedSerializer`(serializer[, bestSize])	Choose the size of batch automatically based on the siz
`Column`(jc)	A column in a DataFrame.
`DataFrame`(jdf, sql_ctx)	A distributed collection of data grouped into named col
`PickleSerializer`()	Serializes objects using Python's pickle serializer:
`SparkContext`([master, appName, sparkHome, ...])	Main entry point for Spark functionality.
`StringType`	String data type.
`UserDefinedFunction`(func, returnType[, name])	User defined function in Python