



SportsCode Statistics Commands

Introduction

Sportscode and StudioCode have advanced script methods available in statistical windows. Simply open a statistical window, right click and select "Edit script". Using basic commands you can change the color of the cells, count labels from timelines, statistical analysis etc...

To execute your script click on "Execute" at the top of the statistics window. See the documentation below for details of the available commands.

Alphabetical list of commands

Command	Description
\$COLUMN	returns the column the cell is in
\$ROW	returns the row the cell is in
ABS	Returns the absolute value of a number
AND	if values are numbers: returns 1 if both values are not zero, otherwise returns 0. If using text then it is assumed that you are referring to labels and the return result will be instances from the timeline that contain both labels. "AND" is commonly used inside other commands (see IF, COUNT...) and can be combined with "OR" and "NOT"
CEILING	Returns a number which has been rounded up to the specified number of digits from the decimal point. If you need to display a certain number of digits after the decimal point use the decimal command
CELL	Grabs the output from another cell in the window
CELL COLOR	changes the color of the cell
COUNT	Will count the number of labels in the timeline (including multiple ones in the same instance)
DECIMAL	Returns a string which has the number to the specified number of digits from the decimal point. The number is always rounded down
END	Will give the latest end time of the labels or instances in the timeline in seconds. returns -1 if no instances
END TIME	Will return all instances that have the correct end time condition
FLOOR	Returns a number which has been rounded down to the specified number of digits from the decimal point. This is similar to the decimal command excepts that it returns a number and not a string. If you need to display a certain number of digits after the decimal point use the decimal command
IF	Executes the statement based on a condition
INSTANCES	Will return all the instances in the timeline.
INSTANCE[x]	Will return the 'x'th instance in the timeline based on start times.
LABELS	Will return all the labels in the timeline. If an instance has 3 labels it will return the instance 3 times
LENGTH	Will give the length of labels in the timeline (including multiple ones in the same instance) in seconds
LIMIT	will limit instances based on conditions
NOT	if value is number: returns 1 if value is 0, otherwise returns 0. If using text then it is assumed that you are referring to labels and the return result will be instances from the timeline that do not contain the label. "NOT" is commonly used inside other commands (see IF, COUNT...) and can be combined with "AND" and "OR"
OR	if values are numbers: returns 1 if either value is not zero, otherwise returns 0. If using text then it is assumed that you are referring to labels and the return result will be instances from the timeline that contain either label. "OR" is commonly used inside other commands (see IF, COUNT...) and can be combined with "AND" and "NOT"
OVERLAP	Will return instances from the given arguments that overlap for any part of the timeline. Opposite of the unique command
ROW COLOR	changes the color of the row
ROUND	Returns a number which has been rounded to the specified number of digits from the decimal point. It will round up or down depending on which is the closest digit. If

	you need to display a certain number of digits after the decimal point use the decimal command
SHOW	Outputs the number or text to the cell
START	Will give the earliest start time of the labels or instances in the timeline in seconds. returns -1 if no instances
START TIME	will return all instances that have the correct start time condition
SUM	adds all the values in the cell group
TIME	will return all instances that have the instance length condition
UNIQUE	Will return instances from the given arguments that do not overlap for any part of the timeline. Opposite of the overlap command
WHERE	will select certain instances based on conditions

Display commands

CELL_COLOR

changes the color of the cell

Format: CELL_COLOR (red , green , blue)

Arguments:

red: a number 0 to 100 green: a number 0 to 100 blue: a number 0 to 100

Example:

`cell_color (100,0,0)` turns the background color of the cell red

ROW_COLOR

changes the color of the row

Format: ROW_COLOR (red , green , blue)

Arguments:

red: a number 0 to 100 green: a number 0 to 100 blue: a number 0 to 100

Example:

`row_color (0,0,100)` turns the row background color blue

SHOW

Outputs the number or text to the cell

Format: SHOW (value)

Arguments:

value: numerical expression or quoted text

Example:

`show "Hello"` displays Hello in the cell

Logical commands

AND

if values are numbers: returns 1 if both values are not zero, otherwise returns 0. If using text then it is assumed that you are referring to labels and the return result will be instances from the timeline that contain both labels. "AND" is commonly used inside other commands (see IF, COUNT...) and can be combined with "OR" and "NOT"

Format: labels AND labels number AND number

Arguments:

labels: quoted text representing labels in the timeline number: numerical expression

Example:

`$a="block" and "spike" show count $a` shows the number of instances with labels "block" and "spike" in them

`show count not ("a" or "b" and "c")` counts the number of instances that do not have "a" or have "b" and "c" as labels in them. AND will be calculated first before the OR as it has higher priority. Use brackets to force logic.

IF

Executes the statement based on a condition

Format: IF (condition, true statement, false statement), IF (condition,true statement)

Arguments:

condition: if this numeric value is 0 it is considered false and will execute the false statement otherwise any nonzero result is considered true and will execute the true statement.

You can use any relational operators <,<=,>,>=,=,! = on numbers or quoted text and logical operators AND, OR, NOT on numbers

statement:

any normal command including another "IF"

Example:

`if (5<6, show "true", show "false")` shows true. You can change the result by changing the condition to `6<5`

`$a=cell(1,0) if ($a<0, show "cell is negative", show "cell is positive")` shows whether cell in column 1, row 0 is positive or negative number

`$a=cell(1,0) if ($a!=0, show 5/$a, show "N/A")` shows 5 divided by the contents of cell in column 1, row 0. If that cell is 0 then show "N/A"

`if ("tom" < "tot" and 4<5, show "true", show "false")` shows true

`if ("tom" < "tot" and 6<5, show "true")` will do nothing as the condition is false and there is no false statement

NOT

if value is number: returns 1 if value is 0, otherwise returns 0. If using text then it is assumed that you are referring to labels and the return result will be instances from the timeline that do not contain the label. "NOT" is commonly used inside other commands (see IF, COUNT...) and can be combined with "AND" and "OR"

Format: NOT labels NOT number

Arguments:

labels: quoted text representing instances labels in the timeline

number: numerical expression

Example:

`$a=not "spike" show count $a` shows the number of instances with so not have "spike" in them

`show count not ("a" or "b" and "c")` counts the number of instances that do not have "a" or have "b" and "c" as labels in them. AND will be calculated first before the OR as it has higher priority. Use brackets to force logic.

OR

if values are numbers: returns 1 if either value is not zero, otherwise returns 0. If using text then it is assumed that you are referring to labels and the return result will be instances from the timeline that contain either label. "OR" is commonly used inside other commands (see IF, COUNT...) and can be combined with "AND" and "NOT"

Format: labels OR labels number OR number

Arguments:

labels: quoted text representing labels in the timeline, number: numerical expression

Example:

`$a="block" or "spike" show count $a` shows the number of instances with either "block" or "spike" in them

`show count not ("a" or "b" and "c")` counts the number of instances that do not have "a" or have "b" and "c" as labels in them. AND will be calculated first before the OR as it has higher priority. Use brackets to force logic.

Numeric commands

These commands perform numeric calculations. All standard mathematical functions +, -, *, /, (), ^ can be used. eg. `show 7+8`

ABS

Returns the absolute value of a number

Format: ABS (number)

Arguments:

number: numerical expression

Example:

`show abs(-5*2)` returns 10

CEILING

Returns a number which has the number rounded up to the specified number of digits from the decimal point. If you need to display a certain number of digits after the decimal point use the decimal command

Format: CEILING (number ,number of digits)

Arguments:

number: numerical expression
number of digits: if positive the number of digits to show at the right of the decimal point. If negative the number of digits to the left of the decimal point starting at tens. If 0 no decimal placing will be left

Example:

show CEILING (34.23001, 2) returns 34.24

show CEILING (3423.456, -2) shows 3500

show decimal (CEILING(0.1,0) ,2) shows 1.00

DECIMAL

Returns a string which has the number to the specified number of digits from the decimal point. The number is always rounded down

Format: DECIMAL (number ,number of digits)

Arguments:

number: numerical expression

number of digits: if positive the number of digits to show at the right of the decimal point. If negative the number of digits to the left of the decimal point starting at tens. If 0 no decimal placing will be left

Example:

show DECIMAL (34.235, 2) shows 34.23

show DECIMAL (3423.456, -2) shows 3400

show DECIMAL (round(0.499,0) ,2) shows 0.00

FLOOR

Returns a number which has the number rounded down to the specified number of digits from the decimal point. This is similar to the deimal command excepts that it returns a number and not a string. If you need to display a certain number of digits after the decimal point use the decimal command

Format: FLOOR (number ,number of digits)

Arguments:

number: numerical expression
number of digits: if positive the number of digits to show at the right of the decimal point. If negative the number of digits to the left of the decimal point starting at tens. If 0 no decimal placing will be left

Example:

show FLOOR (34.235, 2) returns 34.23

show FLOOR (3423.456, -2) shows 3400

show decimal (FLOOR(0.999,0) ,2) shows 0.00

ROUND

Returns a number which has the number rounded to the specified number of digits from the decimal point. It will round up or down depending on which is the closest digit. If you need to display a certain number of digits after the decimal point use the decimal command.

Format: ROUND (number ,number of digits)

Arguments:

number: numerical expression
number of digits: if positive the number of digits to show at the right of the decimal point. If negative the number of digits to the left of the decimal point starting at tens. If 0 no decimal placing will be left

Example:

show ROUND (34.235, 2) returns 34.24

show ROUND (3423.456, -2) returns 3400

show decimal (ROUND(0.499,0) ,2) shows 0.00

Statistical commands

These commands perform statistical calculations performed on cell ranges

SUM

adds all the values in the cell group

Format: SUM (column1 , row1 .. column2 , row2)

Arguments:

column: column number, row: row number

Example:

`sum (2,1..4,3)` returns the sum of all cells between column2, row1 and column4, row3

Text commands

These commands manipulate text. To join text you simply use the + operator. eg. Show "Hello." + "How " + "are you?"

Variables commands

Variables are specified using the dollar sign and a name. eg. \$message = "hello". In that example \$message holds the text "hello" which can be used later on in the script. Variables only hold their value for the cell in the statistical window. A variable can hold a number, text or even instances in the timeline

\$COLUMN

returns the column the cell is in

Format: \$COLUMN

Arguments:

Example:

`show "My current column is "+$column` will show a message what the column is for that cell in the statistical window

\$ROW

returns the row the cell is in

Format: \$ROW

Arguments:

Example:

`show "My current row is "+$row` will show a message what the row is for that cell in the statistical window

Information commands

Information commands return information about timelines, the spreadsheet etc... If you wish to find the current row variables \$row and \$column contain these values.

CELL

Grabs the output from another cell in the window

Format: CELL (reference)

Arguments:

reference: column number , row number *or* quoted text representing a cell title

Example:

`show cell (2,1)` returns the output from the cell in column 2, row 1

`show cell ("mytitle")` returns the output from the cell which has a title "mytitle"

COUNT

Will count the number of labels in the timeline (including multiple ones in the same instance)

Format: COUNT labels

Arguments:

labels: quoted text specifying label you wish to count in the front timeline or instances(*which can be grouped using OR, AND, NOT*) optional WHERE conditions , conditions: ROW = quoted text *or* ROW != quoted text (*which can be grouped using OR, AND, NOT*)

Example:

`count "score"` returns a count of all the "score" labels in the front timeline

`count "score" and "fouls" where row="smith"` returns the total number of "score" and "foul" labels in the row "smith"

`count "score" where row!="smith"` returns the total number of "score" labels which are not in row "smith"

`count "score" where row!="smith" or row!="fred"` returns the total number of score labels in the front timeline which are not in row "smith" or row "fred"

`$a = not ("shot" or "foul") show count $a` shows the total number of labels in the front timeline which do not have "shot" or "foul" labels in them

END

Will give the latest end time of the labels or instances in the timeline in seconds. returns -1 if no instances

Format: END instances

Arguments:

instances: instances or quoted text specifying label you wish to get the latest end time

Example:

`END "score"` returns the end time of the latest instance which has the label "score" in seconds

`END "score" and "fouls" where row="smith"` returns the latest end time of "score" and "foul" labels in the row "smith"

`END instances` returns the end time of the last instance. i.e. The latest overall end time

`show END instance[-1]` shows the end time of the last instance (based on start time). This might not be the latest overall end time

END TIME

will return all instances that have the correct end time condition

Format: instances WHERE END TIME operator value

Arguments:

instances: quoted text specifying label you wish to count in the front timeline or instances(*which can be grouped using OR, AND, NOT*) operator: < <= > >= value: the end time you wish to test (in seconds)

Example:

`count instances where end time > 20` will count the number of instances that have an end time after 20 seconds

`count "a" or "b" where row="row1" and end time > 20` will count the instances that have "a" or "b" in them in row "row1" ending after the first 20 seconds

`count instances where (start time <4 or end time >20) and time<4` will count all the instances which are smaller than 4 seconds, and have a start time in the first 4 seconds or an end time after the last 20 seconds.

INSTANCES

Will return all the instances in the timeline.

Format: INSTANCES

Arguments:

Example:

`length instances` returns the total length of the instances in the timeline.

`count instances` returns the total number of instances in the timeline. Use in conjunction with WHERE to select particular rows

INSTANCE[x]

Will return the 'x'th instance in the timeline based on start times.

Format: INSTANCE[index]

Arguments:

index: a number less than or greater than 0. if index is greater than 0 then it will count from the beginning. eg. 1 represents the first instance in the timeline, 2 the second instance etc. If index is less than 0 it will count from the end. eg. -1 represents the instance with the latest start time, -2 the second latest start time etc.

Example:

`length instance[2]` returns the length of the second instance in the timeline.

`length instance[-3]` returns the length of the third last instance in the timeline. If you wish to have more control then the LIMIT command should be used.

LABELS

Will return all the labels in the timeline. If an instance has 3 labels it will return the instance 3 times.

Format: LABELS

Arguments:

Example:

`count labels` returns the total number of labels in the timeline. Use in conjunction with WHERE to select particular rows

LENGTH

Will give the length of labels in the timeline (including multiple ones in the same instance) in seconds

Format: LENGTH labels

Arguments:

labels: quoted text specifying label you wish to get the length of in the front timeline or instances(*which can be grouped using OR, AND, NOT*) optional WHERE conditions, conditions: ROW = quoted text or ROW != quoted text (*which can be grouped using OR, AND, NOT*)

Example:

`length "score"` returns the total length in seconds of all the "score" labels in the front timeline

`length "score" and "fouls" where row="smith"` returns the total length of "score" and "foul" labels in the row "smith"

`length "score" where row!="smith"` returns the total length of "score" labels which are not in row "smith"

`length "score" where row!="smith" or row!="fred"` returns the total length of score labels in the front timeline which are not in row "smith" or row "fred"

`$a = not ("shot" or "foul") show length $a` shows the total length of labels in the front timeline which do not have "shot" or "foul" labels in them

`show length instance[1]` shows the length in seconds of the first instance in the timeline

LIMIT

will limit instances based on conditions

Format: instances LIMIT offset, select instances LIMIT select

Arguments:

instances: quoted text specifying label you wish to count in the front timeline or instances(*which can be grouped using OR, AND, NOT*) offset: the offset to start counting. eg. 1 = will skip the first instance etc. If offset < 0 then will count from end. eg. -1=last instance. Can be left out if offset is 0. select: how many instances to select. If select < 0 it will select up to the end value (-1=last, -2=2nd last etc...)

Example:

instances limit 2 will grab the first 2 instances

instances limit 4,2 will offset 4 instances and select the next 2. That makes it the 5th and 6th instance.

instances limit 4,-1 will grab the 5th to the last instance

instances limit -3,-2 will grab the 3rd and 2nd last instance

length "a" or "b" where row="row 1" or row="row 2" limit 3,2 returns the total length of the 4th and 5th labels in the timeline with a or b in them and the row = 'row 1' or 'row 2'

OVERLAP

Will return instances from the given arguments that overlap for any part of the timeline. Opposite of the unique command

Format: OVERLAP (instances , instances)

Arguments:

instances: labels or instances

Example:

$\$a=OVERLAP$ ("label 1", "labels 2") returns all instances which have label 1 in them and overlap with any instance containing label2 in them. It will also return all instances which have label 2 in them and overlap with any instance containing label1. It will also return instances with both labels in them but only once.

$\$a=OVERLAP$ (instances,instances) will return all the instances as every instance will overlap itself

$\$a=OVERLAP$ (instance[1], instance[2]) returns both the first and second instance in the timeline provided they overlap. Otherwise it will return nothing

show start OVERLAP ("label 1" where row="row 1" ,"label 2" where row="row 2") will show the start time of the first instance that overlaps where label 1 is in row 1 and label 2 is in row 2

START

Will give the earliest start time of the labels or instances in the timeline in seconds. returns -1 if no instances

Format: START instances

Arguments:

instances: instances or quoted text specifying label you wish to get the earliest start time

Example:

START "score" returns the start time of the earliest instance which has the label "score" in seconds

START "score" and "fouls" where row="smith" returns the earliest start time of "score" and "foul" labels in the row "smith"

START instances returns the start time of the first instance

show START instance[-1] shows the start time of the last instance (based on start time)

START TIME

will return all instances that have the correct start time condition

Format: instances WHERE START TIME operator value

Arguments:

instances: quoted text specifying label you wish to count in the front timeline or instances(*which can be grouped using OR, AND, NOT*) operator: < <= > >= = value: the start time you wish to test (in seconds)

Example:

count instances where start time < 20 will count the number of instances that have a start time in the first 20 seconds

count "a" or "b" where row="row1" and start time < 20 will count the instances that have "a" or "b" in them in row "row1" starting in the first 20 seconds

count instances where (start time <4 or end time >20) and time<4 will count all the instances which are smaller than 4 seconds, and have a start time in the first 4 seconds or an end time after the last 20 seconds.

TIME

will return all instances that have the instance length condition

Format: instances WHERE TIME operator value

Arguments:

instances: quoted text specifying label you wish to count in the front timeline or instances(*which can be grouped using OR, AND, NOT*) operator: < <= > >= = value: the instance length you wish to test (in seconds)

Example:

count instances where time < 20 will count the number of instances that are shorter than 20 seconds in length

count "a" or "b" where row="row1" and time < 20 will count the instances that have "a" or "b" in them in row "row1" and are shorter than 20 seconds

count instances where (start time <4 or end time >20) and time<4 will count all the instances which are smaller than 4 seconds, and have a start time in the first 4 seconds or an end time after the last 20 seconds.

WHERE

will select certain instances based on conditions

Format: labels WHERE conditions

Arguments:

labels: quoted text specifying label you wish to count in the front timeline or instances(*which can be grouped using OR, AND, NOT*) conditions: ROW = quoted text *or* ROW != quoted text (*which can be grouped using OR, AND, NOT*)

Example:

count "a" or "b" where row="row 1" or row="row 2" returns the total number of labels in the timeline with a or b in them and the row = 'row 1' or 'row 2'

count instances where row!="row 1" returns the total number of instances in the timeline not counting those in 'row 1'

UNIQUE

Will return instances from the given arguments that do not overlap for any part of the timeline.

Opposite of the overlap command

Format: UNIQUE (instances , instances)

Arguments:

instances: labels or instances

Example:

\$a=UNIQUE ("label 1", "label 2") returns all instances which have label1 in them and do not overlap with any instance containing label2 in them. It will also return all instances which have label2 in them and do not overlap with any instance containing label1

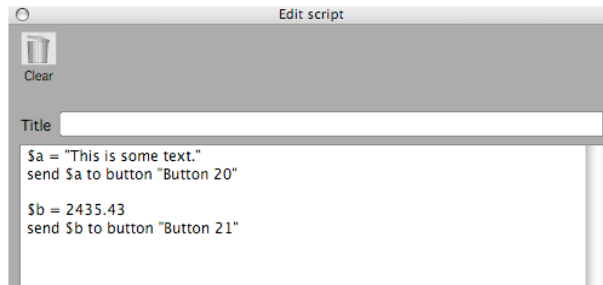
\$a=UNIQUE (instances,instances) will do nothing as there will be no unique instance as it will overlap with itself

\$a=UNIQUE (instance[1], instance[2]) returns both the first and second instance in the timeline provided they do not overlap. Otherwise it will return nothing

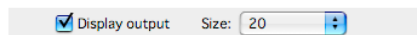
show start UNIQUE ("label 1" where row="row 1" , "label 2" where row="row 2") will show the start time of the first instance that does not overlap where label 1 is in row 1 and label 2 is in row 2

Send Output to a Button in the Code Window

Code buttons can now show output from statistical windows. The output can be strings or numbers. For now numbers show 2 decimal places, but this will change soon.



The following has been added to the button characteristics edit window for code input buttons.



For now the output is not saved with the code window, but it will soon.
For now the output is only sent to the front code window.
Refinements will mean it will be possible to send output to a specific code window, or all code windows.

A button can be

- code input only
- both code input and output
- code output only (Title with output)
- none of the above (Title only)

The Edit Script window has displays contextual errors and usage information.

