INVESTIGATE YOUR DATA
4 POWERFUL SAS LANGUAGES

Vancouver SAS Users Group
Vancouver, British Columbia
26 November 2014

Charu Shankar
Technical Training Specialist
SAS Institute Inc. Canada

Copyright © 2013, SAS Institute Inc. All rights reserved.

SITUATION

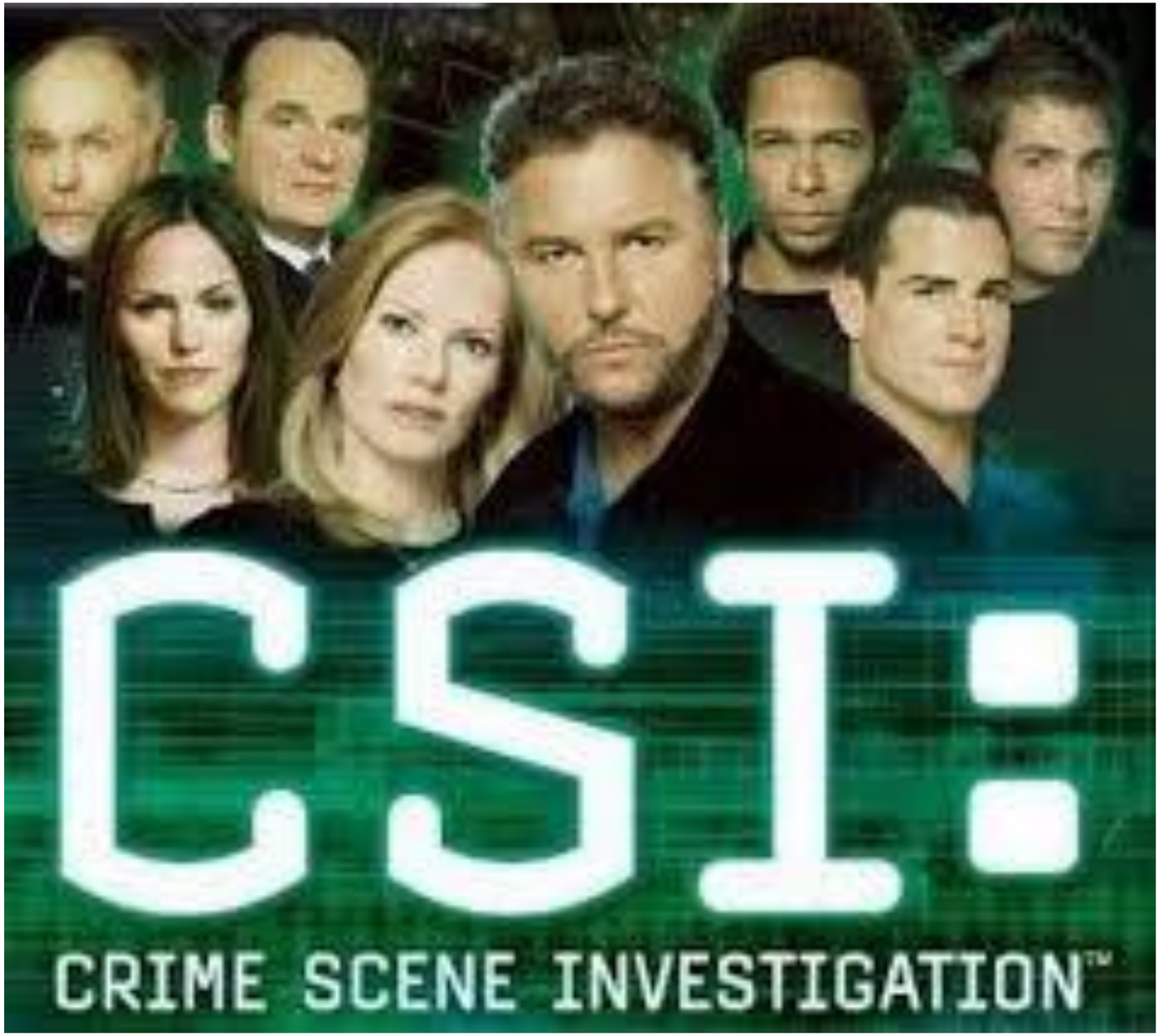
Crime is down overall in Henrico County, but the two districts in the county's east end continue to experience the highest amount of violent crime by a large margin, according to new crime statistics released by the Henrico Police Department.

Eastern districts Varina and Fairfield hold just 40 percent of the county's population, yet account for 66 percent of violent crimes in the county. Those are numbers the county's police chief is working to change.

Crime numbers actually increased in Varina in 2013 compared to 2012.

Burglary, motor vehicle thefts, aggravated assault, and rape offenses are all up from 2012.

Copyright © 2013, SAS Institute Inc. All rights reserved.



Copyright © 2013, SAS Institute Inc. All rights reserved.

Copyright © 2013, SAS Institute Inc. All rights reserved.

WHAT SAS TOOLS CAN YOU USE TO INVESTIGATE DATA

Copyright © 2013, SAS Institute Inc. All rights reserved.

AGENDA

- Tool 1 - The SAS Data step with special operators
- Tool 2 - PERL Natural language processing
- Tool 3 - Interfacing PROC SQL with the macro language
- Tool 4 - SQL the Boolean operator
- Close
- Q&A

Copyright © 2013, SAS Institute Inc. All rights reserved.

PHONE TIP

An anonymous caller left a phone tip saying they heard one suspect call the other one Alfred.

But the police officer says it was a heavily accented voice, so they're not sure that they got the name right

Copyright © 2013, SAS Institute Inc. All rights reserved.

PART 1

THE SAS DATA STEP

The master manipulator

Here is where manipulative is a good Word ☺

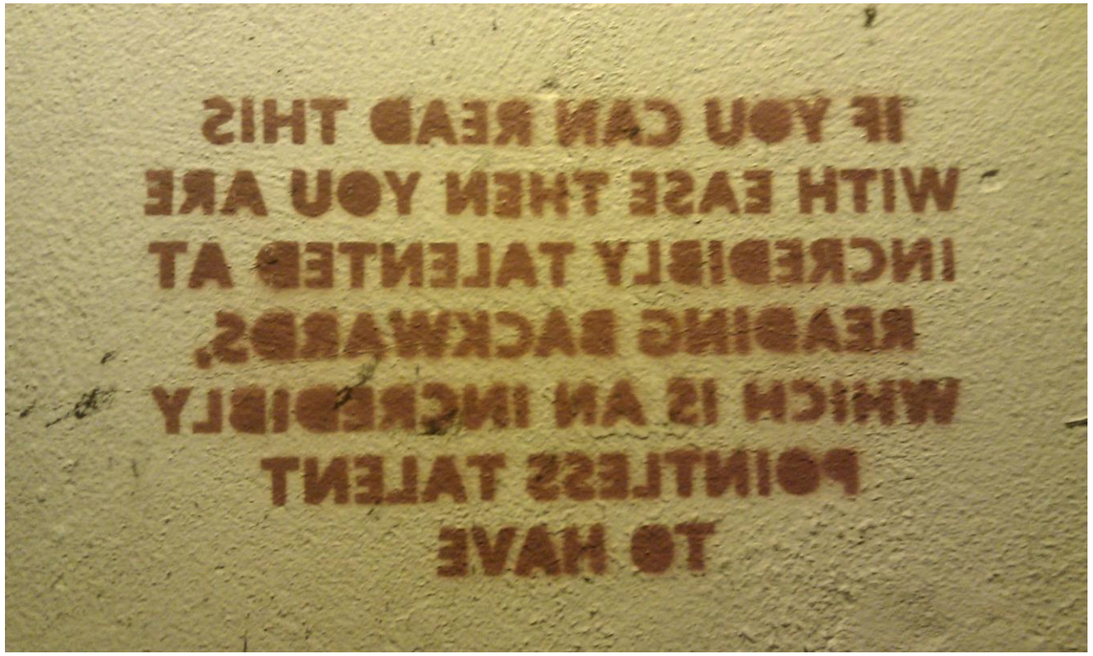
Data step - anytime you want to manipulate data, anytime you want to build.

Copyright © 2013, SAS Institute Inc. All rights reserved.

PART 1
INVESTIGATING JUMBLED UP DATA

7H15 M3554G3
53RV35 7O PR0V3
H0W 0UR M1ND5 C4N
D0 4M4Z1NG 7H1NG5!
1MPR3551V3 7H1NG3!
1N 7H3 B3G1NN1NG
17 WA5 H4RD BU7
Y0UR M1ND 1S
R34D1NG 17
4U70M471C4LLY
W17H 0U7 3V3N
7H1NK1NG 4B0U7 17,
B3 PROUD! 0NLY
C3R741N P39PL3 C4N
R3AD 7H15.
PL3453 F0RW4RD 1F
U C4N R34D 7H15.

Fr scr nd svn yrs ....
-brhm Lncln



Copyright © 2013, SAS Institute Inc. All rights reserved.

LOOKING FOR ALFRED ??

- sounds-like operator =* is very useful when fuzzy matching of character values is needed.
- They match character strings based on their phonetic values.

Copyright © 2013, SAS Institute Inc. All rights reserved.

I LIVE IN THE CITY OF MISSISSAUGA SO THOUGHT I'D GIVE THIS MUCH MISSPELLED CITY A TRY WITH THE SOUNDS LIKE OPERATOR

```
data SoundsLike;
length StudentID $4. Name $7. city $20;
input StudentId $ Name $ city $;
datalines;
1111 aaaa mississauga
2222 bbbbb misisuga
3333 cccccc misisaga
4444 ddddddd missisaga
5555 eee missisuga
6666 ff misisssaga
;
run;
proc print data=SoundsLike noobs;
where city =* 'Mississauga';
run;
```

The SAS System

Obs	city
1	mississauga
2	misisuga
3	misisaga
4	missisaga
5	missisuga
6	misisssaga

Copyright © 2013, SAS Institute Inc. All rights reserved.

HOW DOES THE SOUNDEX ALGORITHM WORK

The steps used by soundex to derive the phonetic equivalent of a character string are as follows:

a) Retain the first letter of the character string

b) Discard the letters A E H I O U W Y

c) Assign a numeric value to the following consonants:

   1. B F P V
   2. C G J K Q S Z
   3. D T
   4. L
   5. M N
   6. R

d) Discard all duplicate classification values if they are adjacent (that is DT will result in a single value of 3, NN will result in a single value of 5).

Copyright © 2013, SAS Institute Inc. All rights reserved.

BUT BEFORE WE GET TOO COMPLACENT

Copyright © 2013, SAS Institute Inc. All rights reserved.

LET'S PUT BACK OUR INVESTIGATOR CAP ON

Does the soundex algorithm work in every possible situation..

- demo

Copyright © 2013, SAS Institute Inc. All rights reserved.

TAKE THE HELP OF ANOTHER OPERATOR, CONTAINS

Copyright © 2013, SAS Institute Inc. All rights reserved.

THIS IS ALL VERY GOOD BUT YAWN HENRICO COUNTY COPS ARE NOT SUITABLY IMPRESSED YET

Copyright © 2013, SAS Institute Inc. All rights reserved.

THAT'S WHEN I REMEMBERED A CUSTOMER DATA CHALLENGE

"Help! How can I find data that matches a specific pattern"? My HS10_ column has a series of any 10 or 6 digit numbers. An additional challenge- this series never appears in the same position".

Sticks or profile shapes of subheading 3916.10

Reproduction proofs for the production of printing plates, rolls, of tariff item No. 8442.50.20

Microcopies of tariff item No. 4903.00.10, 4905.91.00, 4911.10.10 or 4911.10.20

Copyright © 2013, SAS Institute Inc. All rights reserved.

PERL CODE

I never meta character I didn't like. Will Rogers

```
DATA jan201;
 set statcan.'data$'n;
 where prxmatch(("/\d{4}\./"),HS10_TSCHED_EDESC) > 0;
 run;
```

```
NOTE: There were 52 observations read from the data set STATCAN.'data$'n.
      WHERE PRXMATCH('/\d{4}\./', HS10_TSCHED_EDESC)>0;
NOTE: The data set WORK.JAN201 has 52 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time           0.54 seconds
      cpu time            0.21 seconds
```

Of the original 12685 rows, 52 rows match your pattern rows,

/ I used forward slashes as default Perl delimiters
\d matches a digit 0 to 9
{n} matches the previous expression n times
\d{4} matches any 4 digits
\. Is the pattern to match a period

Copyright © 2013, SAS Institute Inc. All rights reserved.

A CLOSER LOOK

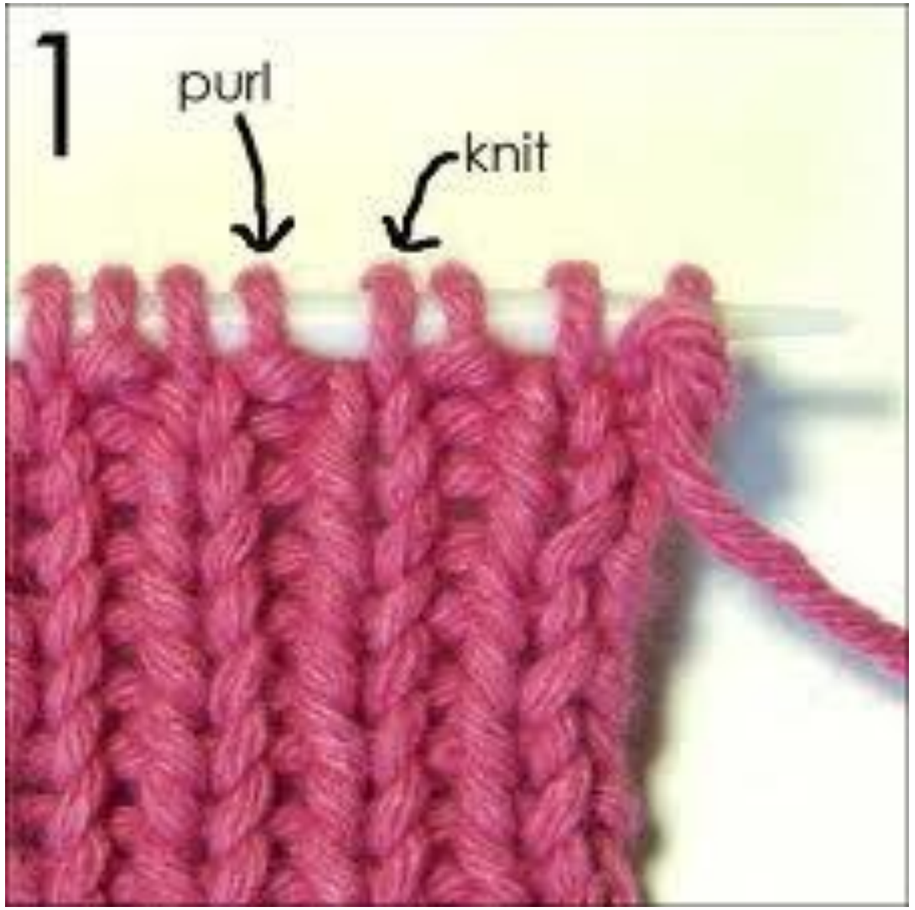
We are now interested in all statutes that have ( ) and can have any type of text values upto 3 characters

Becoz this is leading us closer to the crime !!

	Arrest# - Seq#	ICR #	Specific Crime	Statute Description	Arrest Date/Time
30	14022706402 - C02	140227064	619 - Larceny-Over 200-Other	18.2-95(ii) - Grand Larceny: >=$200 Not From A Person	17MAY14:16:32:00
31	14022706402 - C03	140227064	1011 - Forgery-Check	18.2-172 - Other Forgery Writing:Not In 18.2-168 & 18.2-170	17MAY14:16:32:00
32	14022706402 - C04	140227064	1012 - Forgery-Check-Uttering	18.2-172 - Other Forgery Writing:Not In 18.2-168 & 18.2-170	17MAY14:16:32:00
33	14022706402 - C05	140227064	1012 - Forgery-Check-Uttering	18.2-172 - Other Forgery Writing:Not In 18.2-168 & 18.2-170	17MAY14:16:32:00
34	14022706402 - C06	140227064	1011 - Forgery-Check	18.2-172 - Other Forgery Writing:Not In 18.2-168 & 18.2-170	17MAY14:16:32:00
35	14030131701 - C01	.	1821 - Drug Viol-Marijuana-Possess	18.2-250.1 - Drugs: Possess Marijuana, 1St Off	13MAY14:11:02:00
36	14031133901 - C01	.	2607 - Other Offenses-Contempt Of Court	16.1-278.16 - Order: Not Comply W/ J&Dr Court Order	21MAY14:19:24:00
37	14031929101 - C01	.	1850 - Drug Viol - Poss Of Drug Paraphernalia	54.1-3466 - Possess Or Distribute Controlled Paraphernalia	03JUN14:10:02:00
38	14032322201 - C01	.	1821 - Drug Viol-Marijuana-Possess	18.2-250.1 - Drugs: Possess Marijuana, 1St Off	29MAY14:10:46:00
39	14040205601 - C01	140402056	1200 - Embezzlement	18.2-111 - Embezzlement: >=$200	13MAY14:20:28:00
40	14040323101 - C01	.	2618 - Other Offenses-Probation Violation	19.2-306 - Good Behavior: Violation On Felony Offense	23MAY14:01:24:00
41	14040416901 - C01	140404169	1152 - Fraud-Credit Card	18.2-192(1)(a) - Credit Card Larceny: Take/Obtain No.	14MAY14:17:32:00
42	14040416901 - C02	140404169	1152 - Fraud-Credit Card	18.2-195(1) - Credit Card Fraud: By Person, <$200 In 6M	14MAY14:17:32:00
43	14040416901 - C03	140404169	1152 - Fraud-Credit Card	18.2-195(1) - Credit Card Fraud: By Person, <$200 In 6M	14MAY14:17:32:00
44	14040416901 - C04	140404169	1152 - Fraud-Credit Card	18.2-195(1) - Credit Card Fraud: By Person, <$200 In 6M	14MAY14:17:32:00
45	14041111501 - C01	140411115	450 - Assault-Non-Aggravated	18.2-57.2(A) - Assault & Battery - Family Member	14MAY14:15:57:00
46	14042013401 - C01	140420134	1821 - Drug Viol-Marijuana-Possess	18.2-250.1 - Drugs: Possess Marijuana, 1St Off	21MAY14:23:44:00

Copyright © 2013, SAS Institute Inc. All rights reserved.

TOOL 2 PERL NATURAL LANGUAGE PROCESSING



Copyright © 2013, SAS Institute Inc. All rights reserved.

WE LIKE PERL FOR ITS DENSITY

Perl is a very high-level language.

That means that the code is quite dense

A Perl program may be around 30% to 70% as long as the corresponding program in C.

Copyright © 2013, SAS Institute Inc. All rights reserved.

MANY WORDS TO DESCRIBE PERL, SOME OF THEM MAYBE OFFENSIVE

It is nicknamed "the Swiss Army chainsaw of scripting languages"

Copyright © 2013, SAS Institute Inc. All rights reserved.

PERL IN SAS

Perl regular expressions were added to SAS in Version 9.

SAS regular expressions (similar to Perl regular expressions but using a different syntax to indicate text patterns) have actually been around since version 6.12,

but many SAS users are unfamiliar with either SAS or Perl regular expressions.

Copyright © 2013, SAS Institute Inc. All rights reserved.

WHY PERL

Since SAS already has such a powerful set of string functions, you may wonder why you need regular expressions. .

Copyright © 2013, SAS Institute Inc. All rights reserved.

BACK TO INVESTIGATING OUR DATA

```
42    data vansug.perl;
43    set vansug.criminal;
44    where prxmatch(("/\(\D{1,3}\)/"), statute_description) > 0 ;
45    run;

NOTE: There were 430 observations read from the data set VANSUG.CRIMINAL.
      WHERE PRXMATCH('/\(\D{1,3}\)/', statute_description)>0;
NOTE: The data set VANSUG.PERL has 430 observations and 16 variables.
NOTE: DATA statement used (Total process time):
      real time                0.01 seconds
      cpu time                 0.01 seconds
```

/ delimiters
\(matches an open paranthesis
\Dmatches a non-digit
{n,m} Matches the previous subexpression n or more times, but no more than m
\) matches a closed paranthesis

Copyright © 2013, SAS Institute Inc. All rights reserved.

PERL DEMO

Copyright © 2013, SAS Institute Inc. All rights reserved.

NOW THE COUNTY SHERIFF IS SITTING UP. HE'S BEGINNING TO SEE THE LIGHT OF DAY. HE ASKS

" We want the city in Henrico county which has the count of top crimes". But we have a memory problem can't remember stuff. Too much going on..

Is there a way to store these names so we can reuse over and over again. We want to feed that data to another report."

Copyright © 2013, SAS Institute Inc. All rights reserved.

TOOL 3 THE MACRO LANGUAGE

- We want the city in Henrico county which has the count of top crimes.
- Let’s store it in a macro so we can reuse in another report
- Reduce reuse recycle of course

Copyright © 2013, SAS Institute Inc. All rights reserved.

PROC SQL AND MACRO VARIABLES

The ANSI specification requires SQL to provide a mechanism for passing data values returned by a query to the host system. In PROC SQL, the host (SAS) receives data from a query as macro variable values.

PROC SQL creates or updates macro variables using an INTO clause.

The INTO clause has three syntaxes, and each produces a different result.

Copyright © 2013, SAS Institute Inc. All rights reserved.
sas THE POWER TO KNOW.

PROC SQL AND MACRO VARIABLES

Syntax places values from the **first row** returned by an SQL query into macro variable(s).

Data from additional rows returned by the query is ignored.

General form of the SELECT statement with an INTO clause:

```
SELECT column-1<, …column-n>
      INTO :macvar_1<, ... :macvar_n>
      FROM table|view …
```

The value from the first column in the SELECT list is placed in the first macro variable listed in the INTO clause, and so on.

Copyright © 2013, SAS Institute Inc. All rights reserved.
sas THE POWER TO KNOW.

PROC SQL AND MACRO VARIABLES

Example: Create a single macro variable containing the average salary for the entire company, and use the INTO clause.

```
proc sql noprint;
   select avg(Salary)
      into :MeanSalary
      from orion.Employee_payroll;
%put The average salary is &MeanSalary;
```

Macro variable names are preceded by a colon (:).

- Partial SAS Log

```
The average salary is 38041.51
```


Copyright © 2013, SAS Institute Inc. All rights reserved.

- DEMO LET’S GET THE TOP COUNT OF CRIME

Copyright © 2013, SAS Institute Inc. All rights reserved.

TOOL 4 SQL AND THE BOOLEAN OPERATOR

"Give me powerful enough SQL and I can analyze the world – ARCHIMIDES CIRCA 270 BCE"

Copyright © 2013, SAS Institute Inc. All rights reserved.

PROC SQL AND THE BOOLEAN IS :

The efficient way, the fast way, the only way

clean and elegant way

To

Getting the haves and the have nots

Copyright © 2013, SAS Institute Inc. All rights reserved.

BUSINESS SCENARIO

- Create a report that lists the following for each department:
  - total number of managers
  - total number of non-manager employees
  - manager-to-employee (M/E) ratio

Department	Managers	Employees	M/E Ratio
Accounts	1	5	20%
Administration	2	20	10%

Copyright © 2013, SAS Institute Inc. All rights reserved.

BUSINESS DATA

- Determine if an employee is a manager or a non-manager.
- The **Job_Title** column contains the information about each employee.

Department	Job_Title
Administration	Administration Manager
Administration	Secretary I
Administration	Office Assistant II

Copyright © 2013, SAS Institute Inc. All rights reserved.

COUNTING ROWS THAT MEET A SPECIFIED CRITERION

How do you determine the rows that ***do*** have *Manager* in **Job_Title**, as well as rows that ***do not***?

Question : Can you use the WHERE clause?

Copyright © 2013, SAS Institute Inc. All rights reserved.

COUNTING ROWS THAT MEET A SPECIFIED CRITERION

How do you determine the rows that ***do*** have *Manager* in **Job_Title**, as well as rows that ***do not***?

Answer : You cannot use a WHERE clause to exclude either group.

Department	Job_Title
Administration	Administration Manager
Administration	Secretary I
Administration	Office Assistant II

- Use the FIND function in a Boolean expression to identify rows that contain *Manager* in the **Job_Title** column.

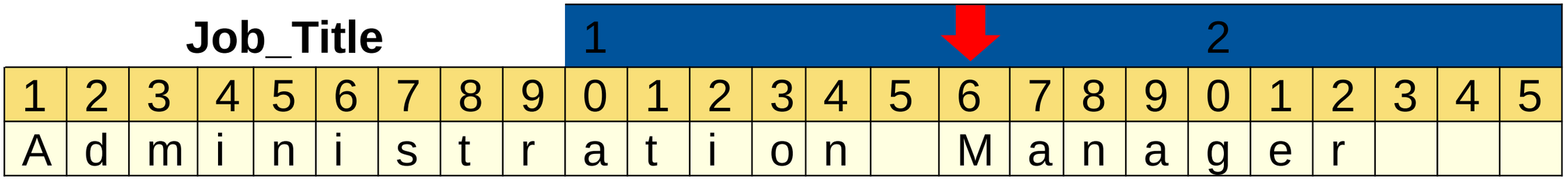
Copyright © 2013, SAS Institute Inc. All rights reserved.

FIND FUNCTION

The *FIND function* returns the starting position of the first occurrence of a substring within a string (character value).

Find the starting position of the substring *Manager* in the character variable **Job_Title**.

```
find(Job_Title,"manager","i")
```



FIND(*string*, *substring*<,*modifier(s)*><,*startpos*>)

The value returned by the FIND function is 16.

Copyright © 2013, SAS Institute Inc. All rights reserved.

USING BOOLEAN EXPRESSIONS

Part 1: Use a Boolean expression to determine if an employee is a manager.

```
proc sql;
select Department, Job_Title,
       (find(Job_Title,"manager","i")>0)
       "Manager"
   from orion.employee_information;
quit;
```

- Boolean expressions evaluate to true (1) or false (0).

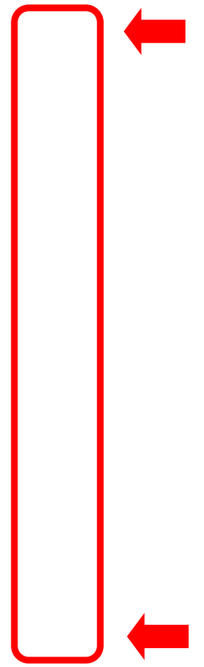
  - If **Job_Title** contains *Manager*, the value is 1.
  - If **Job_Title** does not contain *Manager*, the value is 0.

Copyright © 2013, SAS Institute Inc. All rights reserved.

VIEWING THE OUTPUT

Partial PROC SQL Output

Department	Job_Title	Manager
Administration	Administration Manager	1
Administration	Secretary I	0
Administration	Office Assistant II	0
Administration	Office Assistant III	0
Administration	Warehouse Assistant II	0
Administration	Warehouse Assistant I	0
Administration	Warehouse Assistant III	0
Administration	Security Guard II	0
Administration	Security Guard I	0
Administration	Security Guard II	0
Administration	Security Manager	1



Copyright © 2013, SAS Institute Inc. All rights reserved.

USING BOOLEAN EXPRESSIONS

```
proc sql;
title "Manager-to-Employee Ratios";
select Department,
       sum((find(Job_Title,"manager","i")>0))
         as Managers,
       sum((find(Job_Title,"manager","i")=0))
         as Employees,
       calculated Managers/calculated Employees
          "M/E Ratio" format=percent8.1
   from orion.employee_information
   group by Department;
quit;
```


Copyright © 2013, SAS Institute Inc. All rights reserved.

VIEWING THE OUTPUT

PROC SQL Output

How cool is that?

Manager-to-Employee Ratios

Department	Managers	Employees	M/E Ratio
Accounts	3	14	21.4%
Accounts Management	1	8	12.5%
Administration	5	29	17.2%
Concession Management	1	10	10.0%
Engineering	1	8	12.5%
Executives	0	4	0.0%
Group Financials	0	3	0.0%
Group HR Management	3	15	20.0%
IS	2	23	8.7%
Logistics Management	6	8	75.0%
Marketing	6	14	42.9%
Purchasing	3	15	20.0%
Sales	0	201	0.0%
Sales Management	5	6	83.3%
Secretary of the Board	0	2	0.0%
Stock & Shipping	5	21	23.8%
Strategy	0	2	0.0%

Copyright © 2013, SAS Institute Inc. All rights reserved.

EXTEND THE BOOLEAN TO OTHER INDUSTRIES – E.G HEALTH

high risk individuals 10:37 Monday, May 7, 2012

Diabetes	bmi < 18.5 & 21-30 yrs & Glucose 140-199	bmi < 18.5 & 31-40 yrs & Glucose 140-199	bmi 18.5-25 & 21-30 yrs & Glucose 140-199	bmi 18.5-25 & 31-40 yrs& Glucose 140-199	bmi 26-30 & 21-30 yrs & Glucose 140-199	bmi 26-30 & 31-40 yrs& Glucose 140-199	bmi 30-40 & 21-30 yrs & Glucose 140-199	bmi 30-40 & 31-40 yrs& Glucose 140-199	bmi > 40 & 21-30 yrs & Glucose 140-199	bmi > 40 & 31-40 yrs & Glucose 140-199
0	0	0	3	1	5	6	10	6	8	1
1	0	0	0	2	4	3	22	23	17	5

Copyright © 2013, SAS Institute Inc. All rights reserved.

- DEMO SQL and the Boolean

Copyright © 2013, SAS Institute Inc. All rights reserved.

FINALLY: ABOUT DATA

- For this presentation I shared criminal records data.
- You can just as easily use these 4 powerful SAS languages to investigate your data

Copyright © 2013, SAS Institute Inc. All rights reserved.

GREAT REFERENCES

Data for this presentation was pulled from The Henrico County Virginia database publicly available online http://randolph.co.henrico.va.us/public-data-access/searchicr.aspx

Topic	Web link
Charu Shankar	#1 SAS programming tip for 2012
Charu Shankar	Find your data pattern with PERL
SAS Support	Matching data using Sounds-Like Operators and SAS Compare Functions
phonetics	Phonetic matching of character data
PERL	http://docstore.mik.ua/orelly/perl3/lperl/ch01_02.htm

Copyright © 2013, SAS Institute Inc. All rights reserved.

THANKS FOR ATTENDING
QUESTIONS???

Charu Shankar, SAS institute Inc.

BLOG http://blogs.sas.com/content/sastraining/author/charushankar/

LINKEDIN http://ca.linkedin.com/in/charushankar

TWITTER https://twitter.com/CharuSAS

EMAIL Charu.Shankar@sas.com

Copyright © 2013, SAS Institute Inc. All rights reserved.