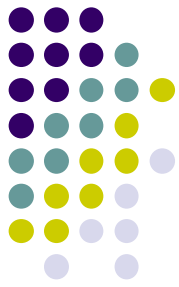


Using SAS to map petroleum distributions

Saskatoon SUCCESS SAS User Group

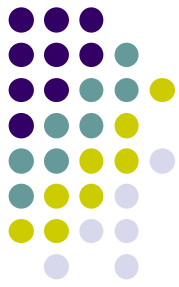
May 13th, 2008

FCL Background



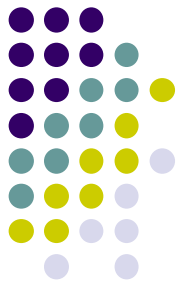
- Wholesaler for over 300 retail Co-operatives with over 1,000,000 retail members.
- Distribution warehouses in Winnipeg, Saskatoon, Calgary and Edmonton.
- 7 Feed plants in Saskatchewan and Alberta.
- 90,000 bbl/day Heavy Oil Upgrader and Refinery Complex in Regina
- Sawmill and Plywood Plant in Canoe, B.C.

SAS at FCL



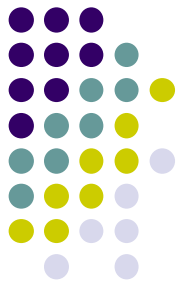
- FCL's has been using SAS since the early 1980's
- Currently use SAS for:
 - Reporting
 - Data manipulation and preparation
 - Data warehouse
 - Statistical analysis

Petroleum distribution planning



- FCL uses a linear program (LP) to optimize petroleum distributions.
- We have a number of ‘exchange agreements’ with different partner companies throughout western Canada.
- The goal of the LP is to find the most economical plan to meet required distribution volumes, given:
 - 1.exchange costs
 - 2.volume limits at each supply point.

Petroleum distribution planning (example)



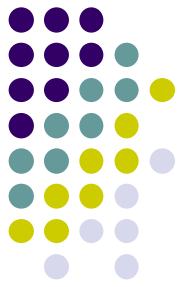
- Consider a petroleum site in Medicine Hat.
 - Costs 3.5 c/L to ship there from Regina.
 - Costs 1.5 c/L to ship there from Calgary, but Calgary supply source incurs a 1.0 c/L additional 'exchange differential' surcharge.
 - Total cost = 2.5 c/L from Calgary.
- Ideally, we would supply this account from Calgary (if there is supply available).

Petroleum Distribution Map



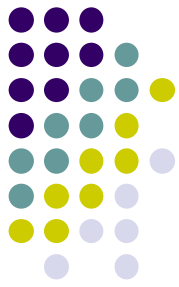
- Goal of the map is to visually display the distribution plan.
- Main benefits:
 - 1. Identify any potential mistakes in the planning system (ie. Supplying an Alberta account from Winnipeg)
 - 2. Visualize the edges of each supply point shipping 'orbit'.
 - 3. Visualize the effect that existing exchange contract limits are imposing on our distribution.

The Map



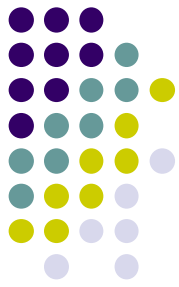
- Maps are generated automatically during each distribution LP run:
 - SAS program is called from Excel Visual Basic, via SAS Integration Technologies.
 - PDF file is automatically regenerated.

The Map



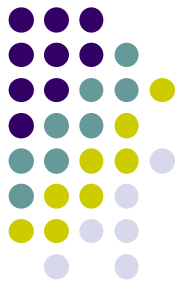
- Sorry, for confidentiality reasons, the distribution maps have been excluded from the public posting of this presentation.

Required data:



- Data is split into three logical groups:
 - SAS Maps – by province.
 - Supply point cities.
 - Plot points, color driven by source location.
- Other data considerations:
 - Split supply – which point to plot?

Proc Gmap



Proc GMap

Map=FinalMap **Data**=Fake ;

Where Province in ('99' '35' '46' '47' '48' '59' '60' '61') ;

Id province ;

Choro number / discrete

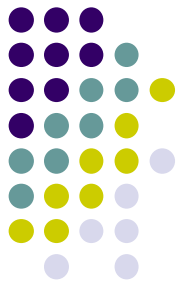
nolegend

Annotate = OptimalSrceAll

outline=black ;

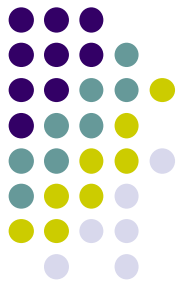
Run ;

SAS Mapping Elements



- Proc Gmap statement
 - Map=
 - Data=
 - ID=
 - Annotate=
- Other mapping considerations
 - Radians vs. Degrees
 - Proc Gproject

Proc Gmap



Proc GMap

Map=FinalMap Data=Fake ;

Where Province in ('99' '35' '46' '47' '48' '59' '60' '61') ;

Id province ;

Choro number / discrete

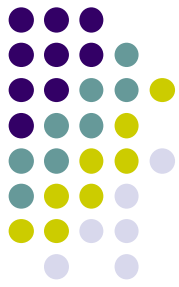
nolegend

Annotate = OptimalSrceAll

outline=black ;

Run ;

Map=



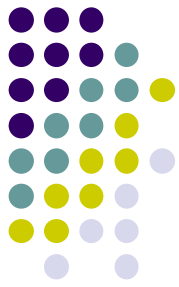
- Specifies the map dataset.
- Is the map outline only.
- Expected variable names:
 - 'X' and 'Y'
 - Variable defining different map areas (Province)

Map=



	Unprojected Longitude in Radians	Unprojected Latitude in Radians	DENSITY	RecType	Province Segment Number	Province Abbreviation
1	0.3247506742	-0.236995437	6	Maps	1	35
2	0.3240224279	-0.237880519	6	Maps	1	35
3	0.3234327266	-0.239372731	6	Maps	1	35
4	0.3227369732	-0.240130283	6	Maps	1	35
5	0.3212280573	-0.241079187	6	Maps	1	35
6	0.3200849593	-0.241532818	6	Maps	1	35
7	0.3193288476	-0.242313818	6	Maps	1	35
8	0.3186400357	-0.243269419	6	Maps	1	35
9	0.3171284966	-0.245213241	6	Maps	1	35
10	0.3165027393	-0.246158146	6	Maps	1	35
11	0.3159492298	-0.247048322	6	Maps	1	35
12	0.3149680926	-0.249847552	6	Maps	1	35
13	0.3148180478	-0.250554899	6	Maps	1	35
14	0.3141493424	-0.251536375	6	Maps	1	35
15	0.3131495334	-0.254181531	6	Maps	1	35
16	0.3127847511	-0.255317482	6	Maps	1	35
17	0.3125736323	-0.256169949	6	Maps	1	35
18	0.3122404094	-0.256797657	6	Maps	1	35

Proc Gmap



Proc GMap

Map=FinalMap **Data=Fake** ;

Where Province in ('99' '35' '46' '47' '48' '59' '60' '61') ;

Id province ;

Choro number / discrete

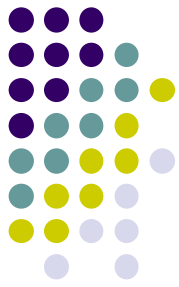
nolegend

Annotate = OptimalSrceAll

outline=black ;

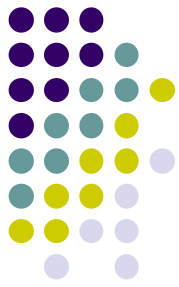
Run ;

Data=



- Specifies the ‘response’ data set
 - Data points correspond to the map objects included in the Map= dataset (in this case provinces).
 - Only the map areas with response values are shown on the map.
 - Contains any data that correlates directly with each map group

Data= ...continued



- My SAS code for the response dataset:
 - * Build fake data required for the response dataset (by Prov);

Data Fake ;

length province \$ 2 ;

input province \$char2. ;

number=1 ;

cards ;

35 46 47 48 59 60 61 ;

Run ;

Data= ...continued



	province	number
1	35	1
2	46	1
3	47	1
4	48	1
5	59	1
6	60	1
7	61	1

Proc Gmap



Proc GMap

```
Map=FinalMap Data=Fake ;
```

```
Where Province in ('99' '35' '46' '47' '48' '59' '60' '61') ;
```

```
Id province ;
```

```
Choro number / discrete
```

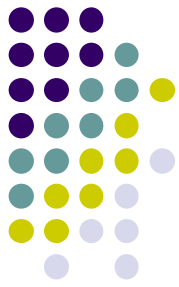
```
nolegend
```

```
Annotate = OptimalSrceAll
```

```
outline=black ;
```

```
Run ;
```

ID=



- Identifies the variable(s) that define map areas
- Every variable listed in the ID statement must appear in both the **Map=** and **Data=** data sets.

Proc Gmap



Proc GMap

Map=FinalMap Data=Fake ;

Where Province in ('99' '35' '46' '47' '48' '59' '60' '61') ;

Id province ;

Choro number / discrete

nolegend

Annotate = OptimalSrceAll

coutline=black ;

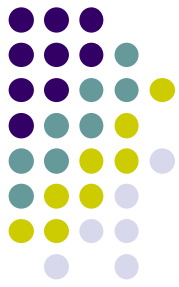
Run ;

Annotate=



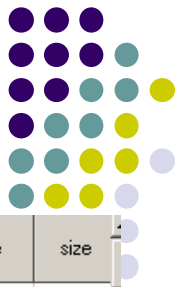
- This is the dataset that holds any additional information to be plotted on the map.
- In my case:
 - Cities
 - Distribution points
- This dataset contains information about how to plot each point:
 - Symbols
 - Colors
 - Labels / Label position

Annotate=continued.



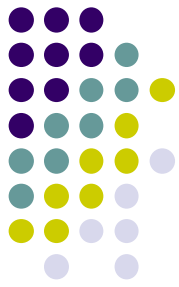
- In this case, really just plotting labels:
 - `function=label ;`
 - For distribution end-points, `text='*'`
 - For supply points, `text='City name'`
- Other variables:
 - `Position=` where to place the label relative to the x/y co-ordinate.
 - `Size=`
 - `Style=`
 - `Color=`

Annotate= ...continued.



	Fvw Supp Pt No	ShpQty3	Lat Degrees	Lon Degrees	ProvName	color	function	text	position	xsys	ysys	style	size
1	07.Nan	13723.792053	49.233	-124.817	BC	purple	label	*	c	2	2	simplex	1
2	07.Nan	4547.7122638	49.233	-124.833	BC	purple	label	*	c	2	2	simplex	1
3	07.Nan	3810.7940979	49.250	-124.800	BC	purple	label	*	c	2	2	simplex	1
4	02.Reg	579.87830409	55.406	-105.156	SK	dark_green	label	*	c	2	2	simplex	1
5	02.Reg	712.78667272	55.417	-104.550	SK	dark_green	label	*	c	2	2	simplex	1
6	05.Edm	4750.6837208	53.883	-112.333	AB	dark_blue	label	*	c	2	2	simplex	1
7	01.Wpg	4043.3003501	50.900	-97.217	MB	dark_yellow	label	*	c	2	2	simplex	1
8	01.Wpg	7028.7813092	50.906	-97.222	MB	dark_yellow	label	*	c	2	2	simplex	1
9	02.Reg	1511.50504	52.433	-103.867	SK	dark_green	label	*	c	2	2	simplex	1
10	02.Reg	310.92956216	52.300	-103.817	SK	dark_green	label	*	c	2	2	simplex	1
11	02.Reg	4482.2018829	49.639	-102.489	SK	dark_green	label	*	c	2	2	simplex	1
12	04.Calg	7949.6958208	50.450	-119.183	BC	red	label	*	c	2	2	simplex	1
13	04.Calg	7232.7719038	50.456	-119.189	BC	red	label	*	c	2	2	simplex	1
14	02.Reg	6703.5707834	50.733	-113.150	AB	dark_green	label	*	c	2	2	simplex	1
15	02.Reg	680.15958992	52.133	-107.217	SK	dark_green	label	*	c	2	2	simplex	1
16	02.Reg	6100.3074675	49.637	-105.987	SK	dark_green	label	*	c	2	2	simplex	1
17	02.Reg	3704.1838999	50.633	-104.117	SK	dark_green	label	*	c	2	2	simplex	1
18	02.Reg	1716.9664697	50.022	-105.072	SK	dark_green	label	*	c	2	2	simplex	1
19	05.Edm	8917.0205087	54.117	-114.400	AB	dark_blue	label	*	c	2	2	simplex	1
20	05.Edm	9611.8714628	54.122	-114.406	AB	dark_blue	label	*	c	2	2	simplex	1
21	01.Wpg	6847.7210018	50.067	-96.517	MB	dark_yellow	label	*	c	2	2	simplex	1
22	01.Wpg	14309.30812	50.072	-96.522	MB	dark_yellow	label	*	c	2	2	simplex	1
23	01.Wpg	4680.4510756	50.367	-95.900	MB	dark_yellow	label	*	c	2	2	simplex	1
24	05.Edm	18424.739298	53.767	-112.783	AB	dark_blue	label	*	c	2	2	simplex	1
25	02.Reg	828.4622161	52.853	-104.053	SK	dark_green	label	*	c	2	2	simplex	1
26	02.Reg	48.624507143	53.100	-103.650	SK	dark_green	label	*	c	2	2	simplex	1
27	02.Reg	4871.4236432	52.856	-104.056	SK	dark_green	label	*	c	2	2	simplex	1
28	02.Reg	517.94055358	52.858	-104.058	SK	dark_green	label	*	c	2	2	simplex	1
29	02.Reg	526.0202965	52.850	-104.050	SK	dark_green	label	*	c	2	2	simplex	1
30	02.Reg	952.4455487	53.100	-103.667	SK	dark_green	label	*	c	2	2	simplex	1
31	02.Reg	1851.3365187	50.717	-105.200	SK	dark_green	label	*	c	2	2	simplex	1
32	02.Reg	97.464730503	52.983	-105.433	SK	dark_green	label	*	c	2	2	simplex	1
33	02.Reg	2000.3780831	52.989	-105.439	SK	dark_green	label	*	c	2	2	simplex	1
34	02.Reg	1318.5828538	52.983	-105.417	SK	dark_green	label	*	c	2	2	simplex	1
35	02.Reg	3103.5900956	49.233	-100.050	MB	dark_green	label	*	c	2	2	simplex	1
36	05.Edm	4724.1852033	54.587	-112.804	AB	dark_blue	label	*	c	2	2	simplex	1
37	05.Edm	924.5000000	54.587	-112.804	AB	dark_blue	label	*	c	2	2	simplex	1

Other considerations



- Radians vs. Degrees
 - SAS map data sets are in radians
 - Must convert latitude/longitude from degrees to radians:

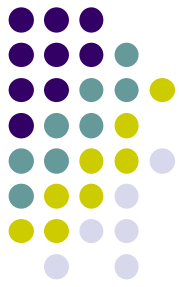
*Convert to radians ;

$$d2r = \text{atan}(1) / 45 ;$$

$$x = d2r * (\text{lond} + \text{lonm} / 60) ;$$

$$y = d2r * (\text{latd} + \text{latm} / 60) ;$$

Other considerations



- Proc Gproject
 - Converts ‘spherical’ coordinates onto a two dimensional plane so they can be properly plotted by Proc Gplot.
 - SAS map data sets and latitude/longitude coordinates must be projected – they will not plot properly in raw form:

```
Proc Gproject Data=Combined Out=CombinedProj ;  
  Id Province RecType ;  
Run ;
```

Questions?

