

### ANA 610 Homework #3

Continuing our analysis of the Donor dataset (s\_pml\_donor\_hw\_v2), refer to “Data Dictionary – Donor.pdf” for data field definitions and your data audit performed for Homework #1.

#### Task #1 (68 pts):

Using SAS, check the variable LIFETIME\_GIFT\_RANGE for extreme values using the following three techniques: top/bottom X%, interquartile range, and clustering.

1. Summarize your findings from all 3 methods using the following table template. Use the shown Benchmark Parameters.

Technique	Benchmark Parameter(s)	Extremes (cutoff value)	Extremes (count)	Extremes (% of file)
Top/Bottom X%	1%	$\geq 48$	196	1.01%
IQR	3*IQR	$\geq 45$	237	1.22%
Clustering	pmin=.006	$\geq 110$	36 extremes total	0.19%
			13 extremes with count of 1	0.07%

2. In a written discussion, discuss (a) whether there are any extreme values (i.e. what single cutoff value do you recommend why); and (b) if so, recommend what should be done about them.
  - a. There are potential extreme values for lifetime\_gift\_range ranging from  $\geq 45$  to  $\geq 110$ . Lifetime\_gift\_range is defined as the maximum gift amount minus the minimum gift amount between 1976 (first gift date) - 1998 (last gift date), which is 22 years. Clustering has a cutoff value of  $\geq 110$ , which is much higher cutoff value and counts 36 extreme values. 1-99% of all the values range between 0-48, and the Top/Bottom approach demonstrates this with a cutoff value of  $\geq 48$ .

However, this dataset spans for 22 years, so it's reasonable to say that a donor's lifetime donation could range higher, since the timeline for this dataset is large. According LIFETIME\_GIFT\_COUNT, donor's donate, on average, 8 times in this dataset.

Analysis Variable : LIFETIME_GIFT_COUNT LIFETIME_GIFT_COUNT				
N	N Miss	Minimum	Median	Maximum
19372	0	1.0000000	8.0000000	95.0000000

According to LIFETIME\_AVG\_GIFT\_AMT, donor's donate, on average, 11.2 every donation in this dataset.

Analysis Variable : LIFETIME_AVG_GIFT_AMT LIFETIME_AVG_GIFT_AMT				
N	N Miss	Minimum	Median	Maximum
19372	0	1.3600000	11.2000000	450.0000000

So if donor's donate, on average, 8 times multiplied by 11.2 (how much they donate on average), it's possible that 89.6 ( $8 \times 11.2 = 89.6$ ) may not be that extreme since this dataset spans 22 years. Therefore, in order to model this sample to become more representative of the data, I recommend to have a cutoff value of  $\geq 90$ .

- b. If we have a cutoff value at  $\geq 90$ , I would recommend to implement a capping rule. I would create a new field called "outlier\_lifetime\_gift\_range" and flag values with a "1" if it exceeds  $\geq 90$ .

```

457 data work.tmp; set sasdata.s_pml_donor_hw_v2;
458   if lifetime_gift_range ge 90 and not missing(control_number notin) then
459     outlier_lifetime_gift_range = 1; else outlier_lifetime_gift_range = 0;
460 run;

```

Based on that analysis, I can exclude or include suspected extreme values to see its effect.

**NOTE:** After applying the 3 techniques, you will find extremes values (we can always find extreme values by varying the benchmark parameter). But, considering all 3 techniques jointly using the assigned benchmark parameters, together with how the variable is defined, what is your recommended cutoff value for an extreme value of the variable (and why).

- Using Tableau, repeat the clustering approach for identifying extreme values for LIFETIME\_GIFT\_RANGE. Do your findings using Tableau confirm your findings using SAS? Explain. Support your explanation with charts and/or summary tables from Tableau.

Yes, my findings in Tableau confirms my findings in SAS; however Tableau was able to get the same cutoff value with a smaller amount of clusters. Tableau created the clusters and identified the same cutoff value (110) (see yellow highlight in tables below) with 50 clusters (like our SAS code). It was also able to produce the same 110 cutoff value with a minimum of 35 clusters too (see table below).

### Inputs for Clustering

Variables: Sum of Lifetime Gift Range

Level of Detail: Control Number

Scaling: Normalized

### Summary Diagnostics

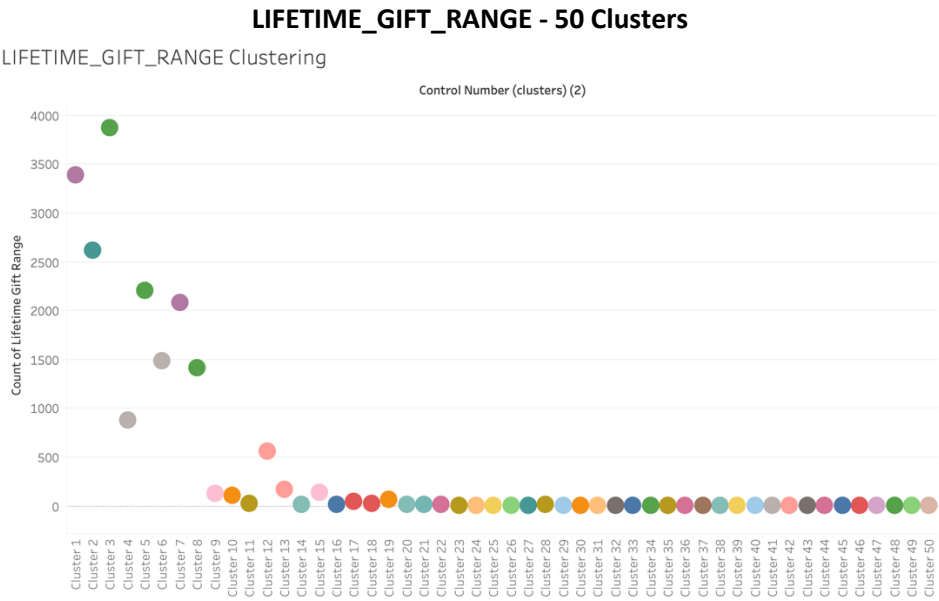
Number of Clusters: 50  
 Number of Points: 19372  
 Between-group Sum of Squares: 4.4444  
 Within-group Sum of Squares: 0.0089706  
 Total Sum of Squares: 4.4533

### Centers

Clusters	Number of Items	Sum of Lifetime Gift Range
----------	-----------------	----------------------------

Cluster 1	3394	4.8881
Cluster 2	2623	0.28219
Cluster 3	3879	9.9852
Cluster 4	877	16.988
Cluster 5	2206	14.883
Cluster 6	1490	20.4
Cluster 7	2081	7.3855
Cluster 8	1415	12.313
Cluster 9	124	39.943
Cluster 10	106	35.412
Cluster 11	23	75.087
Cluster 12	563	24.82
Cluster 13	172	29.569
Cluster 14	17	97.647
Cluster 15	140	44.91
Cluster 16	11	55.0
Cluster 17	45	47.233
Cluster 18	26	50.038
Cluster 19	71	31.76
Cluster 20	18	95.0
Cluster 21	11	80.091
Cluster 22	11	90.0
Cluster 23	2	195.0
Cluster 24	7	60.133
Cluster 25	1	185.0
Cluster 26	3	123.62
Cluster 27	4	175.0
Cluster 28	12	70.375
Cluster 29	2	296.0
Cluster 30	1	137.0
Cluster 31	7	64.857
Cluster 32	1	595.5
Cluster 33	3	57.0
Cluster 34	1	245.0
Cluster 35	1	191.0
Cluster 36	4	85.75
Cluster 37	3	129.33
Cluster 38	3	145.0
Cluster 39	2	199.0
Cluster 40	2	215.0
Cluster 41	1	160.0
Cluster 42	1	225.0
Cluster 43	1	110.0
Cluster 44	1	240.0
Cluster 45	1	490.0
Cluster 46	1	115.0
Cluster 47	1	997.0
Cluster 48	1	248.0

Cluster 49	1	165.0
Cluster 50	1	150.0
Not Clustered	0	



Inputs for Clustering

Variables:	Sum of Lifetime Gift Range
Level of Detail:	Control Number
Scaling:	Normalized

Summary Diagnostics

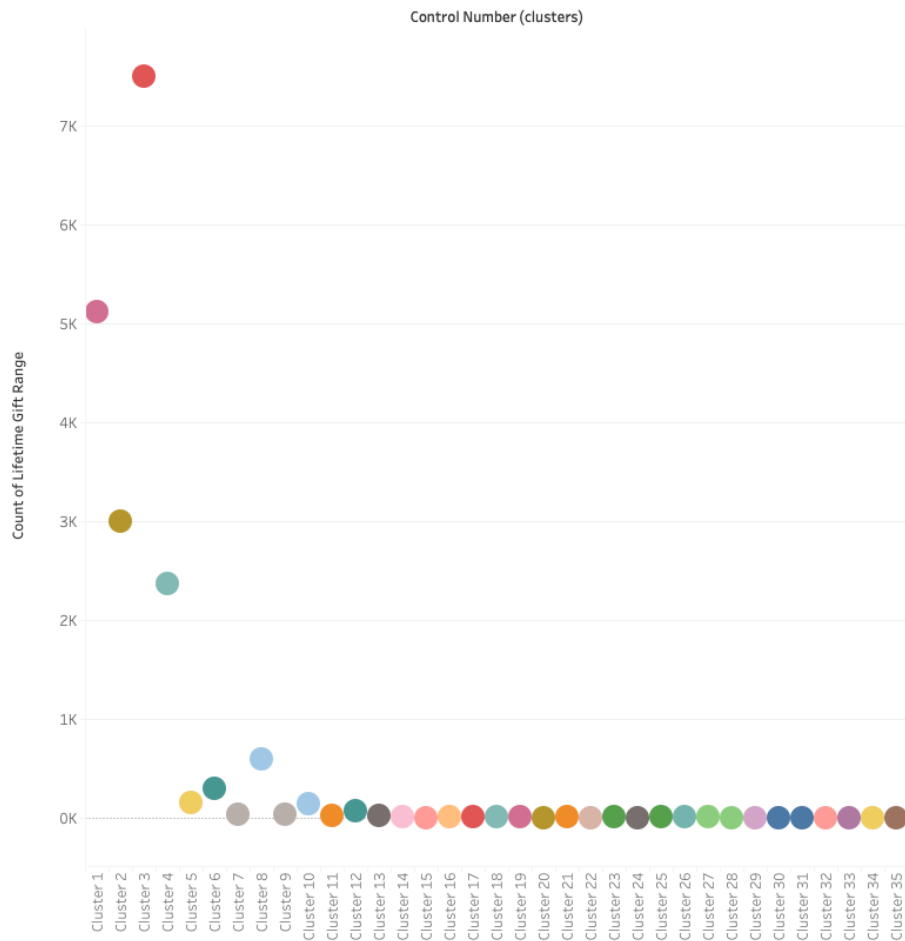
Number of Clusters:	35
Number of Points:	19372
Between-group Sum of Squares:	4.3942
Within-group Sum of Squares:	0.059143
Total Sum of Squares:	4.4533

Centers		
Clusters	Number of Items	Sum of Lifetime Gift Range
Cluster 1	5109	6.0474
Cluster 2	2995	0.61976
Cluster 3	7489	11.865
Cluster 4	2370	19.126
Cluster 5	155	39.301
Cluster 6	291	31.615
Cluster 7	34	76.706

Cluster 8	592	24.957
Cluster 9	35	96.286
Cluster 10	140	44.91
Cluster 11	20	56.746
Cluster 12	71	48.261
Cluster 13	15	88.867
Cluster 14	4	197.0
Cluster 15	1	185.0
Cluster 16	3	123.62
Cluster 17	4	175.0
Cluster 18	12	70.375
Cluster 19	2	296.0
Cluster 20	1	137.0
Cluster 21	8	64.5
Cluster 22	1	595.5
Cluster 23	2	246.5
Cluster 24	1	191.0
Cluster 25	3	129.33
Cluster 26	4	146.25
Cluster 27	2	215.0
Cluster 28	1	160.0
Cluster 29	1	225.0
Cluster 30	1	110.0
Cluster 31	1	240.0
Cluster 32	1	490.0
Cluster 33	1	115.0
Cluster 34	1	997.0
Cluster 35	1	165.0
Not Clustered	0	

#### LIFETIME\_GIFT\_RANGE - 35 Clusters

## LIFETIME\_GIFT\_RANGE Clustering



### Task #2 (26 pts):

- Calculate the cardinality ratio for DONOR\_GENDER, URBANICITY and REGENCY\_STATUS\_96NK.  
**DONOR\_GENDER: 0.02%**  
**URBANICITY: 0.03%**  
**REGENCY\_STATUS\_96NK: 0.03%**
- Using non-macro code, show your SAS code for how you would recode DONOR\_GENDER, URBANICITY, and REGENCY\_STATUS\_96NK into “dummy” variables.

```

180 /* Task 2, Question 2 */
181 /* Import */
182 libname sasdata '/home/u50066252/my_shared_file_links/kevinduffy-denol/Homework 3';
183 data work.dummy1; set sasdata.s_pml_donor_hw_v2;
184 /* Levels */
185 proc freq data=work.dummy1; table donor_gender urbanicity recency_status_96nk; run;
186
187 /* DONOR GENDER */
188 data work.dummy1; set work.dummy1;
189   dgF_dum = (donor_gender = "F");
190   dgM_dum = (donor_gender = "M");
191   dgU_dum = (donor_gender = "U");
192   dgA_dum = (donor_gender = "A");
193 run;
194 %let donor_gender_dums = dgF_dum dgM_dum dgU_dum dgA_dum;
195 proc means data=work.dummy1 nmiss min mean max sum; var &donor_gender_dums; run;
196
197 /* URBANICITY */
198 data work.dummy1; set work.dummy1;
199   urbQ_dum = (urbanicity = "Q");
200   urbC_dum = (urbanicity = "C");
201   urbR_dum = (urbanicity = "R");
202   urbS_dum = (urbanicity = "S");
203   urbT_dum = (urbanicity = "T");
204   urbU_dum = (urbanicity = "U");
205 run;
206 %let urb_dums = urbQ_dum urbC_dum urbR_dum urbS_dum urbT_dum urbU_dum;
207 proc means data=work.dummy1 nmiss min mean max sum; var &urb_dums; run;
208
209 /* RECENCY STATUS_96NK */
210 data work.dummy1; set work.dummy1;
211   rsA_dum = (recency_status_96nk = "A");
212   rsE_dum = (recency_status_96nk = "E");
213   rsF_dum = (recency_status_96nk = "F");
214   rsL_dum = (recency_status_96nk = "L");
215   rsN_dum = (recency_status_96nk = "N");
216   rsS_dum = (recency_status_96nk = "S");
217 run;
218 %let rs_dums = rsA_dum rsE_dum rsF_dum rsL_dum rsN_dum rsS_dum;
219 proc means data=work.dummy1 nmiss min mean max sum; var &rs_dums; run;

```

- Run PROC MEANS, show the output table including NMISS, MIN, MEAN, MAX and SUM.

#### DONOR\_GENDER

Variable	N Miss	Minimum	Mean	Maximum	Sum
dgF_dum	0	0	0.5369089	1.0000000	10401.00
dgM_dum	0	0	0.4105410	1.0000000	7953.00
dgU_dum	0	0	0.0524985	1.0000000	1017.00
dgA_dum	0	0	0.000051621	1.0000000	1.0000000

#### URBANICITY

Variable	N Miss	Minimum	Mean	Maximum	Sum
urbQ_dum	0	0	0.0234359	1.0000000	454.0000000
urbC_dum	0	0	0.2076192	1.0000000	4022.00
urbR_dum	0	0	0.2067417	1.0000000	4005.00
urbS_dum	0	0	0.2318294	1.0000000	4491.00
urbT_dum	0	0	0.2035928	1.0000000	3944.00
urbU_dum	0	0	0.1267809	1.0000000	2456.00

#### RECENCY\_STATUS\_96NK

Variable	N Miss	Minimum	Mean	Maximum	Sum
rsA_dum	0	0	0.6152178	1.0000000	11918.00
rsE_dum	0	0	0.0220421	1.0000000	427.0000000
rsF_dum	0	0	0.0785154	1.0000000	1521.00
rsL_dum	0	0	0.0048007	1.0000000	93.0000000
rsN_dum	0	0	0.0615321	1.0000000	1192.00
rsS_dum	0	0	0.2178918	1.0000000	4221.00

- Does the SUM for each of your dummy variables equals what you found in your data audit report? Discuss and support by including the relevant tables (or portions of) from your audit report.

**Yes, the sum of the dummy variables match what I found in the data audit using PROC FREQ (see tables below). PROC FREQ shows us the frequency of each category in our**

DONOR\_GENDER, URBANICITY, and REGENCY\_STATUS\_96NK variables. The SUM for each of my dummy variables match the Frequency generated using PROC FREQ.

The FREQ Procedure				
DONOR_GENDER				
DONOR_GENDER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
A	1	0.01	1	0.01
F	10401	53.69	10402	53.70
M	7953	41.05	18355	94.75
U	1017	5.25	19372	100.00

URBANICITY				
URBANICITY	Frequency	Percent	Cumulative Frequency	Cumulative Percent
?	454	2.34	454	2.34
C	4022	20.76	4476	23.11
R	4005	20.67	8481	43.78
S	4491	23.18	12972	66.96
T	3944	20.36	16916	87.32
U	2456	12.68	19372	100.00

REGENCY_STATUS_96NK				
REGENCY_STATUS_96NK	Frequency	Percent	Cumulative Frequency	Cumulative Percent
A	11918	61.52	11918	61.52
E	427	2.20	12345	63.73
F	1521	7.85	13866	71.58
L	93	0.48	13959	72.06
N	1192	6.15	15151	78.21
S	4221	21.79	19372	100.00

DONOR_GENDER					
Variable	N Miss	Minimum	Mean	Maximum	Sum
dgF_dum	0	0	0.5369089	1.0000000	10401.00
dgM_dum	0	0	0.4105410	1.0000000	7953.00
dgU_dum	0	0	0.0524985	1.0000000	1017.00
dgA_dum	0	0	0.000051621	1.0000000	1.0000000

URBANICITY					
Variable	N Miss	Minimum	Mean	Maximum	Sum
urbQ_dum	0	0	0.0234359	1.0000000	454.0000000
urbC_dum	0	0	0.2076192	1.0000000	4022.00
urbR_dum	0	0	0.2067417	1.0000000	4005.00
urbS_dum	0	0	0.2318294	1.0000000	4491.00
urbT_dum	0	0	0.2035928	1.0000000	3944.00
urbU_dum	0	0	0.1267809	1.0000000	2456.00

REGENCY_STATUS_96NK					
Variable	N Miss	Minimum	Mean	Maximum	Sum
rsA_dum	0	0	0.6152178	1.0000000	11918.00
rsE_dum	0	0	0.0220421	1.0000000	427.0000000
rsF_dum	0	0	0.0785154	1.0000000	1521.00
rsL_dum	0	0	0.0048007	1.0000000	93.0000000
rsN_dum	0	0	0.0615321	1.0000000	1192.00
rsS_dum	0	0	0.2178918	1.0000000	4221.00

- If all we are after is SUM, why is it useful to also show and review NMISS, MIN, MEAN and MAX in PROC MEANS? Explain what each statistic can tell you specifically about your dummy variables.

It is useful to show and review the NMISS, MIN, MEAN, and MAX in PROC MEANS because it ultimately helps us determine if the dummy variable is correct. NMISS value of 0 shows that there are no missing values and is an indicator that SAS created our dummy variable. MIN value of 0 and a MAX value of 1 makes sense because our dummy variables are either coded with a "0" or a "1" to create a flag for us. A "1" is flagged if there is an instance of a specific category, and a "0" is flagged if there is not. The MEAN shows us the percentage of that category within the variable. For example, for DONOR\_GENDER, our mean for Females is "0.53", which is correlated to 53% of the DONOR\_GENDER values are Female.

### Task #3 (65 pts):

- To avoid overfitting issues, threshold coding can result in fewer dummy variables.
  - Show your SAS code for how you would recode RECENT\_STAR\_STATUS into dummy variables considering a threshold value of 30.



```

234 /* Task 3, Question 1 */
235 /* Import */
236 libname sasdata '/home/u50066252/my_shared_file_links/kevinduffy-deno1/Homework 3';
237 data work.donor3; set sasdata.s_pml_donor_hw v2;
238 recent_star_status_num = RECENT_STAR_STATUS + 1; run;
239 %let anal_var = recent_star_status_num;
240 proc sort data=work.donor3; by &anal_var; run;
241 proc freq data=work.donor3 order=freq;
242 tables &anal_var / out=work.freq (drop = percent); run;
243
244 proc sort data=work.freq; by &anal_var; run;
245 proc sort data=work.donor3; by &anal_var; run;
246
247 /* Create our dummy variables while applying the check for min_count by segment level */
248 %let min_count = 30;
249 data work.dummies; merge work.donor3 work.freq; by &anal_var;
250 array red(23) rstar_dum1-rstar_dum23;
251 do i = 1 to dim(red); if (&anal_var = i and count ge &min_count)
252 then red(i) = 1; else red(i) = 0;
253 end;
254 if sum(of rstar_dum1-rstar_dum23) = 0 then rstar_oth = 1; else rstar_oth = 0; run;
255
256 *====> check sums;
257 proc means data=work.dummies nmiss min mean max sum; var rstar_dum1-rstar_dum23 rstar_oth;
258 output out=work.tmp_sum (drop = _TYPE_ _FREQ_)
259 sum = rstar_dum1-rstar_dum23 rstar_oth; run;
260
261 proc transpose data=work.tmp_sum out=work.tmp_sum_t; run;
262 proc print data=work.tmp_sum_t; run;

```

- b. Execute your code, run PROC MEANS and show the resulting output table including NMISS, MIN, MEAN, MAX and SUM. Check and show that SUM indeed is at least 30 for each dummy variable that passes the threshold test.

The resulting output table successfully shows that the SUM for each dummy variable is at least 30. The “Before” table shows the frequency before the 30 minimum of recent\_star\_status. The “After” table shows after when I applied the Threshold dummy variable coding of a minimum of 30. Dummy variables 3, 10, 16-23 did not meet the 30 minimum requirement, and were separated to “Other” (rstar\_oth).

Before:

The FREQ Procedure				
recent_star_status_num	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	13239	68.34	13239	68.34
2	4289	22.14	17528	90.48
4	346	1.79	17874	92.27
5	320	1.65	18194	93.92
6	205	1.06	18399	94.98
12	183	0.94	18582	95.92
8	140	0.72	18722	96.64
13	122	0.63	18844	97.27
7	88	0.45	18932	97.73
11	86	0.44	19018	98.17
14	84	0.43	19102	98.61
15	61	0.31	19163	98.92
9	54	0.28	19217	99.20
20	27	0.14	19244	99.34
3	26	0.13	19270	99.47
16	26	0.13	19296	99.61
17	18	0.09	19314	99.70
10	15	0.08	19329	99.78
18	15	0.08	19344	99.86
19	12	0.06	19356	99.92
22	10	0.05	19366	99.97
21	4	0.02	19370	99.99
23	2	0.01	19372	100.00

After:

Variable	N Miss	Minimum	Mean	Maximum	Sum
rstar_dum1	0	0	0.6834090	1.0000000	13239.00
rstar_dum2	0	0	0.2214020	1.0000000	4289.00
rstar_dum3	0	0	0	0	0
rstar_dum4	0	0	0.0178608	1.0000000	346.0000000
rstar_dum5	0	0	0.0165187	1.0000000	320.0000000
rstar_dum6	0	0	0.0105823	1.0000000	205.0000000
rstar_dum7	0	0	0.0045426	1.0000000	88.0000000
rstar_dum8	0	0	0.0072269	1.0000000	140.0000000
rstar_dum9	0	0	0.0027875	1.0000000	54.0000000
rstar_dum10	0	0	0	0	0
rstar_dum11	0	0	0.0044394	1.0000000	86.0000000
rstar_dum12	0	0	0.0094466	1.0000000	183.0000000
rstar_dum13	0	0	0.0062977	1.0000000	122.0000000
rstar_dum14	0	0	0.0043362	1.0000000	84.0000000
rstar_dum15	0	0	0.0031489	1.0000000	61.0000000
rstar_dum16	0	0	0	0	0
rstar_dum17	0	0	0	0	0
rstar_dum18	0	0	0	0	0
rstar_dum19	0	0	0	0	0
rstar_dum20	0	0	0	0	0
rstar_dum21	0	0	0	0	0
rstar_dum22	0	0	0	0	0
rstar_dum23	0	0	0	0	0
rstar_oth	0	0	0.0080012	1.0000000	155.0000000

Obs	_NAME_	COL1
1	rstar_dum1	13239
2	rstar_dum2	4289
3	rstar_dum3	0
4	rstar_dum4	346
5	rstar_dum5	320
6	rstar_dum6	205
7	rstar_dum7	88
8	rstar_dum8	140
9	rstar_dum9	54
10	rstar_dum10	0
11	rstar_dum11	86
12	rstar_dum12	183
13	rstar_dum13	122
14	rstar_dum14	84
15	rstar_dum15	61
16	rstar_dum16	0
17	rstar_dum17	0
18	rstar_dum18	0
19	rstar_dum19	0
20	rstar_dum20	0
21	rstar_dum21	0
22	rstar_dum22	0
23	rstar_dum23	0
24	rstar_oth	155

2. Now, the sum for each dummy variable you created in part (1) should be at least 30 for each segment in your analysis as well.
  - a. Check and show using SAS whether the sum for each dummy variable you created in part (1) is 30 for each segment of TARGET\_B, the target variable for a predictive model of likelihood to donate.

The table generated below shows the SUM for each dummy variable created in Part 1 (with a minimum of 30) and how it explains each segment of TARGET\_B (“1” showing who is likely to donate during campaign and “0” showing who is not likely to donate).

The MEANS Procedure

TARGET_B	N Obs	Variable	N Miss	Minimum	Mean	Maximum	Sum
0	14529	rstar_dum1	0	0	0.7041090	1.0000000	10230.00
		rstar_dum2	0	0	0.1982931	1.0000000	2881.00
		rstar_dum3	0	0	0	0	0
		rstar_dum4	0	0	0.0186524	1.0000000	271.0000000
		rstar_dum5	0	0	0.0165187	1.0000000	240.0000000
		rstar_dum6	0	0	0.0106683	1.0000000	155.0000000
		rstar_dum7	0	0	0	0	0
		rstar_dum8	0	0	0.0067451	1.0000000	98.0000000
		rstar_dum9	0	0	0	0	0
		rstar_dum10	0	0	0	0	0
		rstar_dum11	0	0	0	0	0
		rstar_dum12	0	0	0.0099112	1.0000000	144.0000000
		rstar_dum13	0	0	0	0	0
		rstar_dum14	0	0	0	0	0
		rstar_dum15	0	0	0	0	0
		rstar_dum16	0	0	0	0	0
		rstar_dum17	0	0	0	0	0
		rstar_dum18	0	0	0	0	0
		rstar_dum19	0	0	0	0	0
		rstar_dum20	0	0	0	0	0
		rstar_dum21	0	0	0	0	0
		rstar_dum22	0	0	0	0	0
		rstar_dum23	0	0	0	0	0
		rstar_oth	0	0	0.0351022	1.0000000	510.0000000
1	4843	rstar_dum1	0	0	0.6213091	1.0000000	3009.00
		rstar_dum2	0	0	0.2907289	1.0000000	1408.00
		rstar_dum3	0	0	0	0	0
		rstar_dum4	0	0	0.0154863	1.0000000	75.0000000
		rstar_dum5	0	0	0.0165187	1.0000000	80.0000000
		rstar_dum6	0	0	0.0103242	1.0000000	50.0000000
		rstar_dum7	0	0	0	0	0
		rstar_dum8	0	0	0.0086723	1.0000000	42.0000000
		rstar_dum9	0	0	0	0	0
		rstar_dum10	0	0	0	0	0
		rstar_dum11	0	0	0	0	0
		rstar_dum12	0	0	0.0080529	1.0000000	39.0000000
		rstar_dum13	0	0	0	0	0
		rstar_dum14	0	0	0	0	0
		rstar_dum15	0	0	0	0	0
		rstar_dum16	0	0	0	0	0
		rstar_dum17	0	0	0	0	0
		rstar_dum18	0	0	0	0	0
		rstar_dum19	0	0	0	0	0
		rstar_dum20	0	0	0	0	0
		rstar_dum21	0	0	0	0	0
		rstar_dum22	0	0	0	0	0
		rstar_dum23	0	0	0	0	0
		rstar_oth	0	0	0.0289077	1.0000000	140.0000000

- b. Now, show your SAS code for how you would then create dummy variables only if the sum is at least 30 in both TARGET\_B segments.

```

281 /* Task 3, 2b */
282 /* Find freq by segment level and merge onto master */
283 libname sasdata '/home/u50066252/my_shared_file_links/kevinduffy-denol/Homework 3';
284 data work.donor3; set sasdata.s_pml_donor_hw_v2;
285 recent_star_status_num = RECENT_STAR_STATUS + 1; run;
286 %let segment = target_b;
287 %let anal_var = recent_star_status_num;
288 proc sort data=work.donor3; by &segment; run;
289 proc freq data=work.donor3 order=freq; table &anal_var / out=work.freq (drop = percent);
290 by &segment;
291 where not missing(&segment); run;
292 data work.seg_1 work.seg_0; set work.freq;
293 if &segment = 1 then output work.seg_1;
294 if &segment = 0 then output work.seg_0; run;
295 data work.seg_1; set work.seg_1; count_seg_1 = count; keep &anal_var count_seg_1; run;
296 data work.seg_0; set work.seg_0; count_seg_0 = count; keep &anal_var count_seg_0; run;
297 proc sort data=work.seg_1; by &anal_var; run;
298 proc sort data=work.seg_0; by &anal_var; run;
299 proc sort data=work.donor3; by &anal_var; run;
300 /* Create our dummy variables while applying the check for min_count by segment level */
301 %let min_count = 30;
302 data work.dummies; merge work.donor3 work.seg_1 work.seg_0; by &anal_var;
303 array red(23) rstar_dum1-rstar_dum23;
304 do i = 1 to dim(red); if (&anal_var = i and count_seg_0 ge &min_count and count_seg_1 ge &min_count)
305 then red(i) = 1; else red(i) = 0;
306 end;
307 if sum(of rstar_dum1-rstar_dum23) = 0 then rstar_oth = 1; else rstar_oth = 0; run;
308 *====> check sums by segment;
309 proc sort data=work.dummies; by &segment; run;
310 proc means data=work.dummies sum; var rstar_dum1-rstar_dum23 rstar_oth;
311 output out=work.tmp_sum (drop = _TYPE_ _FREQ_)
312 sum = rstar_dum1-rstar_dum23 rstar_oth; by &segment;
313 where not missing(&segment); run;
314 proc transpose data=work.tmp_sum out=work.tmp_sum_t; run;
315 proc print data=work.tmp_sum_t; run;

```

- c. Check and show that SUM of your dummies created in 2(b) is indeed at least 30 in each TARGET\_B segment.

The resulting output table successfully shows that the SUM for each dummy variable is at least 30 for both segments of TARGET\_B (“0” and “1”). The “Before” table shows the frequency of recent\_star\_status in each segment of TARGET\_B before the 30 minimum threshold. The “After” table shows after when I applied the Threshold dummy variable coding of a minimum of 30. Dummy variables 3, 7, 9-11, 13-23 did not meet the 30 minimum requirement for each TARGET\_B segment, and were separated to “Other” (rstar\_oth).

Before:

TARGET_B=0					TARGET_B=1				
recent_star_status_num	Frequency	Percent	Cumulative Frequency	Cumulative Percent	recent_star_status_num	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	10230	70.41	10230	70.41	1	3009	62.13	3009	62.13
2	2881	19.83	13111	90.24	2	1408	29.07	4417	91.20
3	19	0.13	13130	90.37	3	7	0.14	4424	91.35
4	271	1.87	13401	92.24	4	75	1.55	4499	92.90
5	240	1.65	13641	93.89	5	80	1.65	4579	94.55
6	155	1.07	13796	94.95	6	50	1.03	4629	95.58
7	64	0.44	13860	95.40	7	24	0.50	4653	96.08
8	98	0.67	13958	96.07	8	42	0.87	4695	96.94
9	39	0.27	13997	96.34	9	15	0.31	4710	97.25
10	14	0.10	14011	96.43	10	1	0.02	4711	97.27
11	67	0.46	14078	96.90	11	19	0.39	4730	97.67
12	144	0.99	14222	97.89	12	39	0.81	4769	98.47
13	96	0.66	14318	98.55	13	26	0.54	4795	99.01
14	71	0.49	14389	99.04	14	13	0.27	4808	99.28
15	50	0.34	14439	99.38	15	11	0.23	4819	99.50
16	24	0.17	14463	99.55	16	2	0.04	4821	99.55
17	13	0.09	14476	99.64	17	5	0.10	4826	99.65
18	12	0.08	14488	99.72	18	3	0.06	4829	99.71
19	9	0.06	14497	99.78	19	3	0.06	4832	99.77
20	24	0.17	14521	99.94	20	3	0.06	4835	99.83
21	3	0.02	14524	99.97	21	1	0.02	4836	99.86
22	4	0.03	14528	99.99	22	6	0.12	4842	99.98
23	1	0.01	14529	100.00	23	1	0.02	4843	100.00

After:

TARGET_B=0		TARGET_B=1		Obs	_NAME_	_LABEL_	COL1	COL2
Variable	Sum	Variable	Sum	1	TARGET_B	TARGET_B	0	1
rstar_dum1	10230.00	rstar_dum1	3009.00	2	rstar_dum1		10230	3009
rstar_dum2	2881.00	rstar_dum2	1408.00	3	rstar_dum2		2881	1408
rstar_dum3	0	rstar_dum3	0	4	rstar_dum3		0	0
rstar_dum4	271.0000000	rstar_dum4	75.0000000	5	rstar_dum4		271	75
rstar_dum5	240.0000000	rstar_dum5	80.0000000	6	rstar_dum5		240	80
rstar_dum6	155.0000000	rstar_dum6	50.0000000	7	rstar_dum6		155	50
rstar_dum7	0	rstar_dum7	0	8	rstar_dum7		0	0
rstar_dum8	98.0000000	rstar_dum8	42.0000000	9	rstar_dum8		98	42
rstar_dum9	0	rstar_dum9	0	10	rstar_dum9		0	0
rstar_dum10	0	rstar_dum10	0	11	rstar_dum10		0	0
rstar_dum11	0	rstar_dum11	0	12	rstar_dum11		0	0
rstar_dum12	144.0000000	rstar_dum12	39.0000000	13	rstar_dum12		144	39
rstar_dum13	0	rstar_dum13	0	14	rstar_dum13		0	0
rstar_dum14	0	rstar_dum14	0	15	rstar_dum14		0	0
rstar_dum15	0	rstar_dum15	0	16	rstar_dum15		0	0
rstar_dum16	0	rstar_dum16	0	17	rstar_dum16		0	0
rstar_dum17	0	rstar_dum17	0	18	rstar_dum17		0	0
rstar_dum18	0	rstar_dum18	0	19	rstar_dum18		0	0
rstar_dum19	0	rstar_dum19	0	20	rstar_dum19		0	0
rstar_dum20	0	rstar_dum20	0	21	rstar_dum20		0	0
rstar_dum21	0	rstar_dum21	0	22	rstar_dum21		0	0
rstar_dum22	0	rstar_dum22	0	23	rstar_dum22		0	0
rstar_dum23	0	rstar_dum23	0	24	rstar_dum23		0	0
rstar_oth	510.0000000	rstar_oth	140.0000000	25	rstar_oth		510	140

NOTE: SAS arrays should start with "1". Since RECENT\_STAR\_STATUS starts with 0 and is a numeric, create RECENT\_STAR\_STATUS\_NUM = RECENT\_STAR\_STATUS + 1 and use it as your analysis variable.

Task #4 (61 pts):

Another remedy for sparseness in the levels of categorical variables is to collapse the levels. Ideally, subject-matter considerations should be used to collapse levels. This is not always practical in predictive modeling.

1. Using PROC CLUSTER and the process presented in lecture, determine the optimal number of clusters for RECENT\_STAR\_STATUS and create dummy variables for these clusters.
  - a. Create and show the table with the RECENT\_STAR\_STATUS level proportions (PROP) that will be used in the clustering.

Obs	RECENT_STAR_STATUS	_TYPE_	_FREQ_	prop
1	21	1	10	60.00%
2	22	1	2	50.00%
3	1	1	4289	32.83%
4	7	1	140	30.00%
5	8	1	54	27.78%
6	16	1	18	27.78%
7	6	1	88	27.27%
8	2	1	26	26.92%
9	4	1	320	25.00%
10	18	1	12	25.00%
11	20	1	4	25.00%
12	5	1	205	24.39%
13	0	1	13239	22.73%
14	10	1	86	22.09%
15	3	1	346	21.68%
16	11	1	183	21.31%
17	12	1	122	21.31%
18	17	1	15	20.00%
19	14	1	61	18.03%
20	13	1	84	15.48%
21	19	1	27	11.11%
22	15	1	26	7.69%
23	9	1	15	6.67%

- b. Using your table from (1) and RECENT\_STAR\_STATUS "0" as an example, explain where the PROP value comes from for this level. Support with relevant tables and include numbers and calculations in your answer.

**In my previous table, for RECENT\_STAR\_STATUS "0", this shows that out of 13239 counts where RECENT\_STAR\_STATUS is "0", 22.73% of the 13239 are associated with our TARGET\_B variable.  $(3009/13239)*100 = 22.73\%$**

**We can code this by:**

```

344 | *FREQ of RECENT_STAR_STATUS;
345 | proc freq data=&input_data; table &anal_var; run;          /* are 13239 accts at "0" recent_star_status */
346 | *FREQ of branch where target_b = 1;
347 | proc freq data=&input_data; table &anal_var; where &target_var = 1; run; /* are 3009 "1" of target_b */

```

**Our output tables are:**

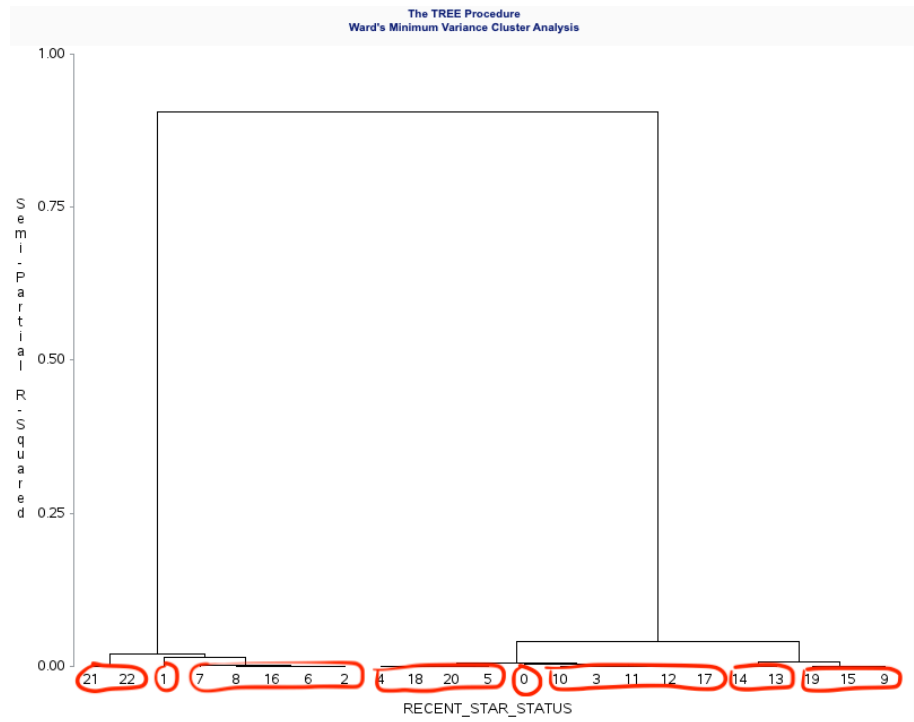
RECENT_STAR_STATUS				
RECENT_STAR_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	13239	68.34	13239	68.34
1	4289	22.14	17528	90.48
2	26	0.13	17554	90.62
3	346	1.79	17900	92.40
4	320	1.65	18220	94.05
5	205	1.06	18425	95.11
6	88	0.45	18513	95.57
7	140	0.72	18653	96.29
8	54	0.28	18707	96.57
9	15	0.08	18722	96.64
10	86	0.44	18808	97.09
11	183	0.94	18991	98.03
12	122	0.63	19113	98.66
13	84	0.43	19197	99.10
14	61	0.31	19258	99.41
15	26	0.13	19284	99.55
16	18	0.09	19302	99.64
17	15	0.08	19317	99.72
18	12	0.06	19329	99.78
19	27	0.14	19356	99.92
20	4	0.02	19360	99.94
21	10	0.05	19370	99.99
22	2	0.01	19372	100.00

RECENT_STAR_STATUS				
RECENT_STAR_STATUS	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	3009	62.13	3009	62.13
1	1408	29.07	4417	91.20
2	7	0.14	4424	91.35
3	75	1.55	4499	92.90
4	80	1.65	4579	94.55
5	50	1.03	4629	95.58
6	24	0.50	4653	96.08
7	42	0.87	4695	96.94
8	15	0.31	4710	97.25
9	1	0.02	4711	97.27
10	19	0.39	4730	97.67
11	39	0.81	4769	98.47
12	26	0.54	4795	99.01
13	13	0.27	4808	99.28
14	11	0.23	4819	99.50
15	2	0.04	4821	99.55
16	5	0.10	4826	99.65
17	3	0.06	4829	99.71
18	3	0.06	4832	99.77
19	3	0.06	4835	99.83
20	1	0.02	4836	99.86
21	6	0.12	4842	99.98
22	1	0.02	4843	100.00

To create this percentage, SAS counts the total of RECENT\_STAR\_STATUS (13239 observations) and then counts where TARGET\_B is "1" (3009 observations). After we divide these two numbers and multiply by 100, we create our PROP value:  $(3009 / 13239) * 100 = 22.73\%$

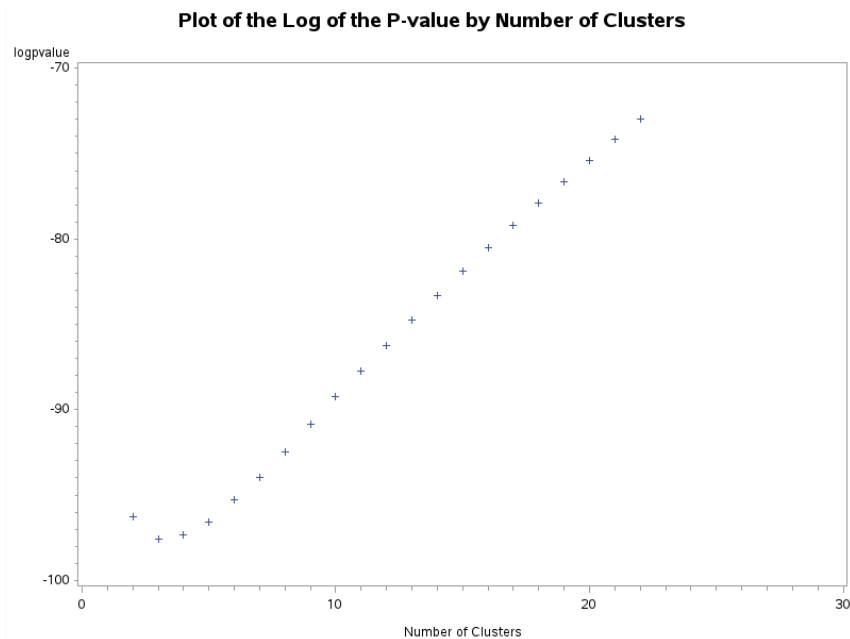
13	0	1	13239	22.73%
----	---	---	-------	--------

- c. Show the tree chart (dendrogram) and indicate your guess at the number of clusters. Circle your guesses on the tree chart. Explain your guess.  
**I guess 8 clusters because I counted 8 groups (circled in red below) at the bottom of the tree chart (dendrogram). Typically, clusters are at the ends of the long tree branches. Visually, it looks like there are 8 clusters of RECENT\_STAR\_STATUS branches.**



- d. Determine the optimal number of clusters (i.e. what do the statistics say?) and show the plot of the log p-value.

**The optimal number of clusters for RECENT\_STAR\_STATUS is 3. This is calculated by the minimum log p-value and can be visually represented on the plot at the tip of the “elbow”.**





Number of Clusters
3

- e. Show a table with the cluster assignments of the levels for RECENT\_STAR\_STATUS. Show and discuss why your guess of the number of clusters did or did not differ from the optimal number.

The output table below identifies 3 clusters, which is different my guess of 8 clusters based off the tree chart. In my tree chart, I was visually looking at the very ends of the branches, which is why I guessed 8 clusters. SAS grouped 8, 16, 6, 2, 21, 7, 1 as one cluster (CL3), while I guessed based off the tree chart that it was three separate clusters. SAS grouped 4, 18, 20, 11, 12, 10, 3, 17, 5, 0 as a cluster (CL6), while I guessed it was also three separate clusters. SAS grouped 15, 9, 14, 13, 19 as a cluster (CL6), while I guessed it was two separate clusters.

SAS identified that the RECENT\_STAR\_STATUS category proportions are similar, which is why SAS grouped them as a cluster. By clustering these categorical variable levels, it minimizes the reduction of the overall chi-square and it helps preserve the difference in proportions across the clustered levels.

Levels of Categorical Variable by Cluster

CLUSNAME	RECENT_STAR_STATUS	CLUSTER
CL3	8	1
	16	1
	6	1
	2	1
	21	1
	22	1
	7	1
	1	1

CLUSNAME	RECENT_STAR_STATUS	CLUSTER
CL5	15	3
	9	3
	14	3
	13	3
	19	3

CLUSNAME	RECENT_STAR_STATUS	CLUSTER
CL6	4	2
	18	2
	20	2
	11	2
	12	2
	10	2
	3	2
	17	2
	5	2
	0	2

- f. Show your SAS code for how you would create dummy variables for these cluster assignments.

```

406 data work.clus2; set work.clus; drop clusname; run;
407 proc sort data=work.clus2; by &anal_var; run;
408 proc sort data=&input_data; by &anal_var; run;
409 data work.scored; merge &input_data work.clus2; by &anal_var;
410 rstar_clus1=(cluster=1);
411 rstar_clus2=(cluster=2);
412 rstar_clus3=(cluster=3); run;
413 %let dum_vars = rstar_clus1 rstar_clus2 rstar_clus3;
414 proc means data=work.scored sum; var &dum_vars; run;

```

- g. Show a table with the frequencies of the new cluster dummy variables.  
The output table below shows the frequencies for each of the 3 cluster dummy variables created. The “Sum” is the number of occurrences SAS found for each cluster.

#### Levels of Categorical Variable by Cluster

The MEANS Procedure

Variable	Sum
rstar_clus1	4627.00
rstar_clus2	14532.00
rstar_clus3	213.0000000

- h. Check and show that for each cluster dummy variable, SUM is at least 30 in each TARGET\_B segment.  
The output table below shows that the SUM for each TARGET\_B segment (“0” and “1”) is at least 30. The SUM for each cluster is broken down if TARGET\_B was met (“1”), or not met (“0”). For TARGET\_B = “1”, rstar\_clus3, the 30 minimum was just barely met.

#### Levels of Categorical Variable by Cluster

The MEANS Procedure

TARGET\_B=0

Variable	Sum
rstar_clus1	3119.00
rstar_clus2	11227.00
rstar_clus3	183.0000000

TARGET\_B=1

Variable	Sum
rstar_clus1	1508.00
rstar_clus2	3305.00
rstar_clus3	30.0000000

#### Levels of Categorical Variable by Cluster

Obs	_NAME_	_LABEL_	COL1	COL2
1	TARGET_B	TARGET_B	0	1
2	rstar_clus1		3119	1508
3	rstar_clus2		11227	3305
4	rstar_clus3		183	30

2. Using Tableau,
- Determine the optimal number of clusters for RECENT\_STAR\_STATUS, allowing Tableau to set the number automatically. Explain what Tableau finds, supporting your discussion with Tableau’s “Describe Clusters” data as well as a visual of the cluster assignment.  
According to Tableau, the optimal number of clusters for RECENT\_STAR\_STATUS is 3. Tableau groups RECENT\_STAR\_STATUS values of 0-8, 10-12, 16-18, and 20 as Cluster 1 (blue dots). Values 9, 13-15, and 19 are grouped as Cluster 2 (orange dots). Values 21 and 22 are grouped as Cluster 3 (red dots).

Tableau’s “Describe Clusters” shows us the numbers of values in each Cluster (For example, Cluster 1 has 16 items) and also shows us the Centers for my Target B Prop

variable, which is the proportion for each cluster ratio. Tableau’s “Analysis of Variance” generated a F-statistic of 8.889 and a p-value of 0.001729. The higher the F-statistic, the higher the statistical significance. A low p-value (< 0.05) also shows statistical significance. This indicates that our F-statistic (8.889) and p-value (0.001729) is statistically significant.

Inputs for Clustering

Variables: Target B Prop  
Level of Detail: Recent Star Status  
Scaling: Normalized

Summary Diagnostics

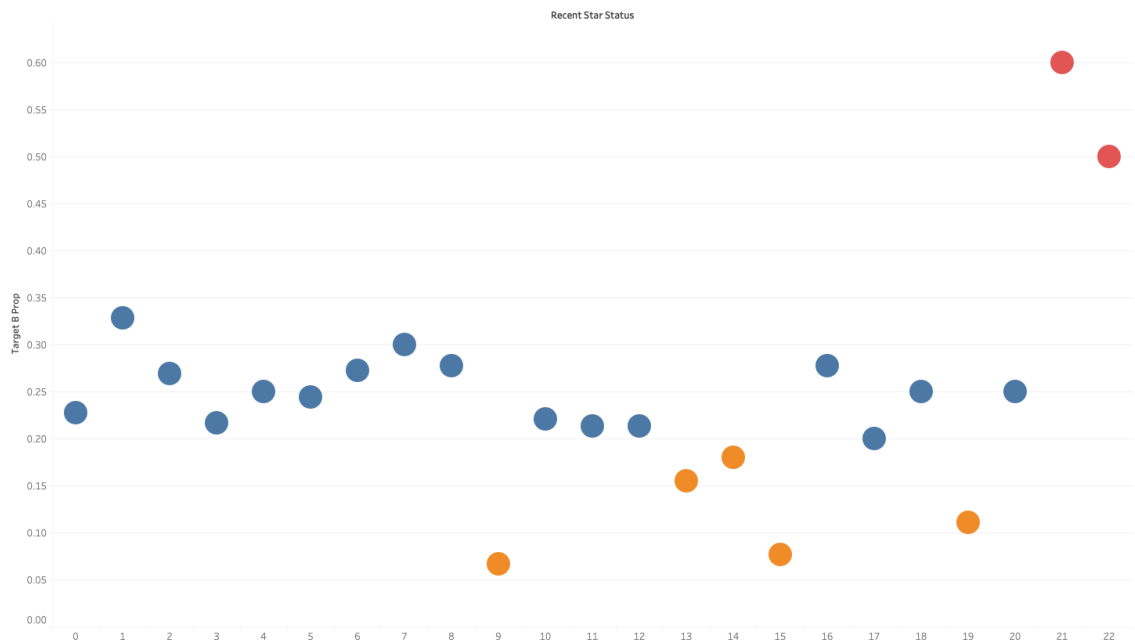
Number of Clusters: 3  
Number of Points: 23  
Between-group Sum of Squares: 0.93894  
Within-group Sum of Squares: 0.11735  
Total Sum of Squares: 1.0563

Clusters	Number of Items	Centers
		Target B Prop
Cluster 1	16	0.25068
Cluster 2	5	0.11796
Cluster 3	2	0.55
Not Clustered	0	

Analysis of Variance:

Variable	F-statistic	p-value	Model		Error	
			Sum of Squares	DF	Sum of Squares	DF
Target B Prop	8.889	0.001729	0.9389	2	1.056	20

RECENT\_STAR\_STATUS Clustering



- b. Do the number of clusters differ from what you found above using SAS? Discuss and include an analysis of the differences in cluster membership between Tableau and SAS. The number of clusters generated from Tableau (3 clusters) is the same amount of Clusters generated from SAS (3 Clusters); however the grouping of each cluster is different.

In SAS, “CL5” is grouped exactly the same as Tableau (15, 9, 14, 13, 19).

In SAS, “CL3” is grouped as: 8, 16, 6, 2, 21, 22, 7, 1. In Tableau, groups this as: 0-8, 10-12, 16-18, and 20.

In SAS “CL6” is grouped as: 4, 18, 20, 11, 12, 10, 3, 17, 5, 0. In Tableau, groups this as: 21 and 22.

Tableau's method of using k-means (partitive clustering) to partition the data into k clusters is what likely caused the cluster membership to be different. It starts with k clusters, then splits and reassigns. Tableau finds the k, which maximizes the highest ratio of the between-cluster variance to the within-cluster variance.

SAS uses an agglomerative hierarchy, where all the data start in their own cluster, the nearest clusters are joined, and the clusters are hierarchically nested within clusters at earlier iterations.

Levels of Categorical Variable by Cluster

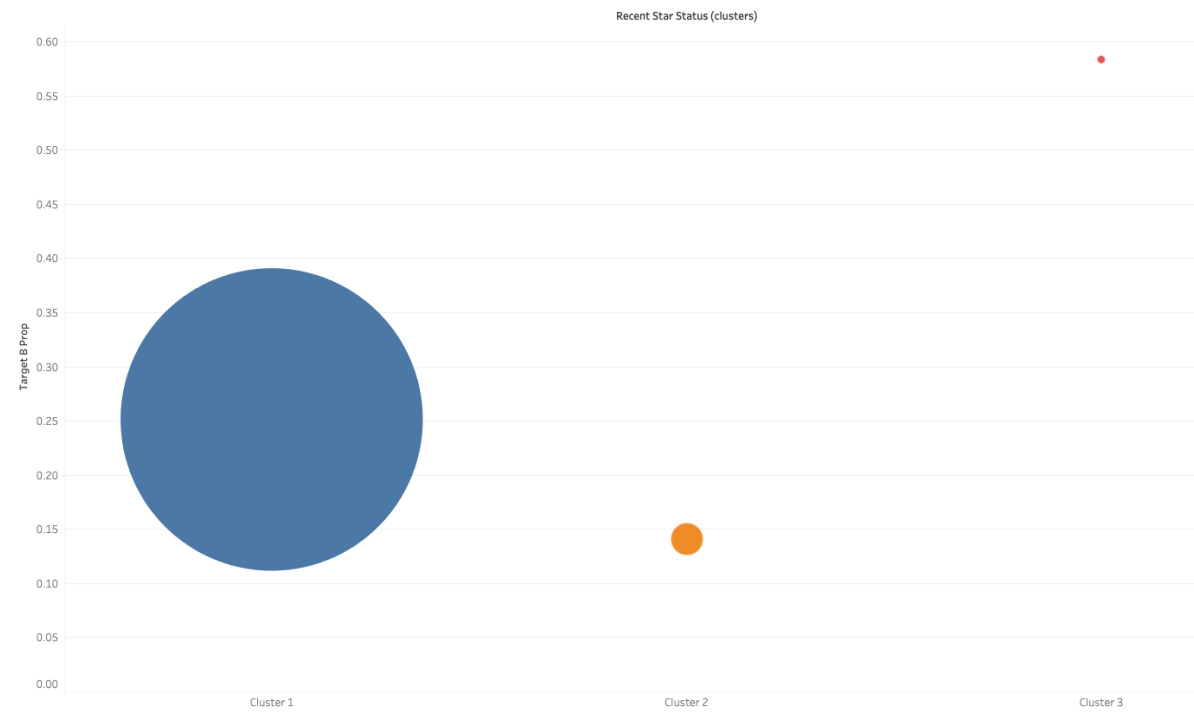
CLUSNAME	RECENT_STAR_STATUS	CLUSTER
CL3	8	1
	16	1
	6	1
	2	1
	21	1
	22	1
	7	1
	1	1

CLUSNAME	RECENT_STAR_STATUS	CLUSTER
CL5	15	3
	9	3
	14	3
	13	3
	19	3

CLUSNAME	RECENT_STAR_STATUS	CLUSTER
CL6	4	2
	18	2
	20	2
	11	2
	12	2
	10	2
	3	2
	17	2
	5	2
	0	2

- c. Create a visual showing cluster 1,2,...N on the horizontal axis, PROP on the vertical axis and the number of records in each cluster indicated by the size of the chart mark.

# RECENT\_STAR\_STATUS with TARGET\_B Prop Clustering



## Extra Credit (20 pts):

1. We can create another variable from our categorical variables by enumerating them, using the Weight of Evidence (WOE).
  - a. Using the SAS code provided in lecture, calculate the WOE and information value (IV) for each level of RECENT\_STAR\_STATUS and summarize in a frequency table. Your table should have 8 columns, just like the one from lecture: RECENT\_STAR\_STATUS, \_FREQ\_, events, non-events, pct\_events, pct\_non\_events, WOE and the IV.

Obs	RECENT_STAR_STATUS	_FREQ_	events	non_events	pct_events	pct_non_events	woe	iv
1	0	13239	3009	10230	0.62131	0.70411	-0.12510	0.010359
2	1	4289	1408	2881	0.29073	0.19829	0.38265	0.035370
3	2	26	7	19	0.00145	0.00131	0.10008	0.000014
4	3	346	75	271	0.01549	0.01865	-0.18602	0.000589
5	4	320	80	240	0.01652	0.01652	0.00000	0.000000
6	5	205	50	155	0.01032	0.01067	-0.03279	0.000011
7	6	88	24	64	0.00496	0.00440	0.11778	0.000065
8	7	140	42	98	0.00867	0.00675	0.25131	0.000484
9	8	54	15	39	0.00310	0.00268	0.14310	0.000059
10	9	15	1	14	0.00021	0.00096	-1.54045	0.001166
11	10	86	19	67	0.00392	0.00461	-0.16164	0.000111
12	11	183	39	144	0.00805	0.00991	-0.20764	0.000386
13	12	122	26	96	0.00537	0.00661	-0.20764	0.000257
14	13	84	13	71	0.00268	0.00489	-0.59912	0.001320
15	14	61	11	50	0.00227	0.00344	-0.41552	0.000486
16	15	26	2	24	0.00041	0.00165	-1.38629	0.001717
17	16	18	5	13	0.00103	0.00089	0.14310	0.000020
18	17	15	3	12	0.00062	0.00083	-0.28768	0.000059
19	18	12	3	9	0.00062	0.00062	0.00000	0.000000
20	19	27	3	24	0.00062	0.00165	-0.98083	0.001013
21	20	4	1	3	0.00021	0.00021	0.00000	0.000000
22	21	10	6	4	0.00124	0.00028	1.50408	0.001449
23	22	2	1	1	0.00021	0.00007	1.09861	0.000151
		19372	4843	14529	1.00000	1.00000	-2.39000	0.055087

- b. Your table should show that there are 346 occurrences of RECENT\_STAR\_STATUS = 3. Explain, using numbers and calculations, how the values in each of the following columns are derived for that row in your table:

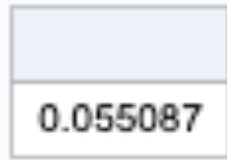
- 1) "Events"
  - i. "Events" are the number of events where TARGET\_B = 1 when RECENT\_STAR\_STAUS = 3. Out of 346 occurrences of when RECENT\_STAR\_STATUS = 3, here were 75 events where TARGET\_B = 1.
- 2) "Non-events"
  - i. "Non-events" are the number of events where TARGET\_B = 0 when RECENT\_STAR\_STAUS = 3. Out of 346 occurrences of when RECENT\_STAR\_STATUS = 3, here were 271 events where TARGET\_B = 1. We can also check this by subtracting the total count (346) minus the "Events" (75), which equals our "Non-events" (271).
- 3) "Pct\_events"
  - i. "Percent of events" is calculated by the number of "Events" divided by the total sum of "Events". When RECENT\_STAR\_STAUS = 3, this is calculated by 75 (Events) divided by 4843 (total sum of "Events"), which equals 0.01549.
- 4) "Pct\_non\_events"
  - i. "Percent of non-events" is calculated by the number of "Non-events" divided by the total sum of "Non-Events". When RECENT\_STAR\_STAUS = 3, this is calculated by 271 (Events) divided by 14529 (total sum of "Non-events"), which equals 0.01865.
- 5) WOE
  - i. "Weight of evidence" (WOE) is the measure of the separation of TARGET\_B= 1 from TARGET\_B=0 in a particular group (RECENT\_STAR\_STATUS).

"Weight of evidence" is calculated by:  $\ln((\text{events}/\text{total events})/(\text{non-events}/\text{total non-events}))$

When RECENT\_STAR\_STAUS = 3, WOE is calculated by:  
 $\ln((75/4843)/(271/14529)) = -0.18602$
- 6) IV
  - i. "Information value" (IV) is calculated by:  $(\% \text{ events} - \% \text{non-events}) * \text{WOE}$ .

When RECENT\_STAR\_STAUS = 3, IV is calculated by:  
 $(0.01549 - 0.01865) * -0.18602 = 0.000589$

- c. Also calculate the variable-level Information Value (IV) and interpret as to the likely predictive power of this variable in a model explaining TARGET\_B.  
 The variable-level IV can be calculated by taking the sum of all the RECENT\_STAR\_STATUS levels. The calculated variable-level IV is 0.055087, which is considered a weak predictive power (< 0.1). This shows that RECENT\_STAR\_STATUS is unlikely to be predictive of whether donors fit TARGET\_B.



0.055087

**Homework deliverables:**

- Word doc with your analysis and discussion, including all tables and charts
- A SAS program with all code used for this assignment