Data 621 Homework 4: Code Appendix

Jeff Nieman, Scott Karr, James Topor, Armenoush Aslanian-Persico

Contents

This Appendix contains all of the source R code and associated relevant output from our final writeup and our model building efforts. The R code is organized to match up to the relevant sections of the Writeup document.

However, we begin here by providing the full ouput of our Evaluation data set predictions as indicated in Part 4 of the final writeup document.

Full Results of Evaluation Data Set Predictions

The full set of Evaluation data set predictions listed in order of their 'INDEX' identifier is as follows:

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
3	0.221	0	0
9	0.455	0	0
10	0.126	0	0
18	0.181	0	0
21	0.266	0	0
30	0.163	0	0
31	0.343	0	0
37	0.320	0	0
39	0.021	0	0
47	0.174	0	0
60	0.026	0	0
62	0.563	1	4088
63	0.830	1	3619
64	0.118	0	0
68	0.033	0	0
75	0.583	1	4005

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
76	0.703	1	3262
83	0.144	0	0
87	0.515	1	3855
92	0.376	0	0
98	0.180	0	0
106	0.453	0	0
107	0.104	0	0
113	0.323	0	0
120	0.364	0	0
123	0.413	0	0
125	0.426	0	0
126	0.456	0	0
128	0.135	0	0
129	0.177	0	0
131	0.194	0	0
135	0.450	0	0
141	0.060	0	0
147	0.198	0	0
148	0.108	0	0
151	0.039	0	0
156	0.177	0	0
157	0.091	0	0
174	0.041	0	0
186	0.601	1	4149
193	0.276	0	0
195	0.476	0	0
212	0.020	0	0
213	0.497	0	0
217	0.005	0	0
223	0.214	0	0
226	0.149	0	0
228	0.490	0	0
230	0.015	0	0
241	0.517	1	3535
243	0.177	0	0
249	0.321	0	0
281	0.791	1	4160
288	0.108	0	0
294	0.491	0	0
295	0.194	0	0
300	0.402	0	0
302	0.359	0	0
303	0.076	0	0
308	0.601	1	3652
319	0.010	0	0
320	0.080	0	0
324	0.365	0	0
331	0.230	0	0
343	0.042	0	0
347	0.535	1	3206
348	0.750	1	3842
350	0.473	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
357	0.150	0	0
358	0.054	0	0
360	0.037	0	0
366	0.231	0	0
367	0.624	1	3865
368	0.300	0	0
376	0.711	1	4046
380	0.365	0	0
388	0.384	0	0
396	0.250	0	0
398	0.176	0	0
403	0.057	0	0
410	0.557	1	3512
412	0.389	0	0
420	0.422	0	0
434	0.049	0	0
440	0.520	1	4046
450	0.557	1	4386
453	0.241	0	0
464	0.349	0	0
465	0.036	0	0
466	0.761	1	4251
473	0.091	0	0
476	0.074	0	0
478	0.073	0	0
479	0.215	0	0
493	0.051	0	0
497	0.329	0	0
503	0.006	0	0
504	0.336	0	0
505	0.407	0	0
507	0.257	0	0
513	0.539	1	3322
519	0.454	0	0
521	0.558	1	4127
522	0.770	1	4261
545	0.123	0	0
549	0.051	0	0
551	0.163	0	0
556	0.082	0	0
557	0.425	0	0
559	0.399	0	0
560	0.738	1	3896
566	0.035	0	0
569	0.141	0	0
573	0.442	0	0
578	0.649	1	3935
579	0.037	0	0
582	0.020	0	0
596	0.770	1	4000
598	0.643	1	3489
599	0.253	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
602	0.340	0	0
605	0.899	1	4022
617	0.708	1	3717
619	0.530	1	3790
630	0.189	0	0
634	0.592	1	3737
643	0.363	0	0
645	0.198	0	0
647	0.383	0	0
649	0.075	0	0
656	0.155	0	0
657	0.150	0	0
658	0.174	0	0
667	0.046	0	0
692	0.255	0	0
693	0.265	0	0
698	0.730	1	4187
699	0.604	1	4300
700	0.051	0	0
704	0.142	0	0
707	0.077	0	0
708	0.716	1	3863
709	0.092	0	0
713	0.125	0	0
714	0.049	0	0
716	0.675	1	3262
718	0.122	0	0
722	0.099	0	0
729	0.458	0	0
731	0.029	0	0
733	0.470	0	0
746	0.227	0	0
747	0.562	1	4123
748	0.459	0	0
753	0.280	0	0
757	0.418	0	0
763	0.032	0	0
767	0.140	0	0
774	0.487	0	0
776	0.491	0	0
788	0.192	0	0
794	0.377	0	0
799	0.226	0	0
803	0.182	0	0
806	0.309	0	0
807	0.207	0	0
811	0.024	0	0
816	0.159	0	0
818	0.327	0	0
819	0.127	0	0
831	0.117	0	0
835	0.498	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
837	0.210	0	0
841	0.843	1	4544
846	0.285	0	0
856	0.402	0	0
861	0.502	1	4676
862	0.456	0	0
863	0.777	1	4507
865	0.718	1	3885
871	0.515	1	3886
879	0.188	0	0
880	0.099	0	0
881	0.252	0	0
885	0.312	0	0
887	0.262	0	0
892	0.051	0	0
898	0.059	0	0
900	0.161	0	0
904	0.289	0	0
906	0.689	1	3618
910	0.702	1	3924
912	0.291	0	0
913	0.379	0	0
919	0.060	0	0
924	0.619	1	3453
925	0.404	0	0
930	0.172	0	0
940	0.261	0	0
941	0.144	0	0
946	0.197	0	0
949	0.309	0	0
951	0.106	0	0
962	0.032	0	0
966	0.068	0	0
967	0.013	0	0
971	0.843	1	3713
981	0.017	0	0
982	0.054	0	0
983	0.106	0	0
984	0.045	0	0
989	0.172	0	0
990	0.602	1	4025
992	0.475	0	0
995	0.120	0	0
996	0.393	0	0
998	0.542	1	3830
1001	0.048	0	0
1007	0.129	0	0
1008	0.086	0	0
1016	0.047	0	0
1022	0.059	0	0
1027	0.392	0	0
1032	0.394	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
1033	0.235	0	0
1041	0.231	0	0
1065	0.596	1	3930
1074	0.379	0	0
1075	0.314	0	0
1081	0.133	0	0
1094	0.052	0	0
1099	0.080	0	0
1105	0.358	0	0
1123	0.072	0	0
1135	0.008	0	0
1142	0.216	0	0
1155	0.088	0	0
1169	0.017	0	0
1176	0.035	0	0
1178	0.615	1	4470
1180	0.027	0	0
1184	0.121	0	0
1185	0.506	1	3961
1193	0.221	0	0
1196	0.156	0	0
1199	0.461	0	0
1203	0.311	0	0
1205	0.444	0	0
1207	0.034	0	0
1208	0.499	0	0
1212	0.569	1	3815
1213	0.667	1	3762
1222	0.087	0	0
1223	0.220	0	0
1226	0.399	0	0
1227	0.220	0	0
1229	0.084	0	0
1230	0.279	0	0
1231	0.700	1	3833
1241	0.063	0	0
1243	0.051	0	0
1244	0.166	0	0
1246	0.160	0	0
1248	0.168	0	0
1249	0.148	0	0
1252	0.056	0	0
1261	0.110	0	0
1275	0.091	0	0
1281	0.857	1	4411
1285	0.427	0	0
1288	0.546	1	3965
1290	0.064	0	0
1291	0.260	0	0
1304	0.642	1	3973
1305	0.100	0	0
1323	0.228	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
1342	0.459	0	0
1348	0.291	0	0
1353	0.261	0	0
1363	0.162	0	0
1371	0.346	0	0
1372	0.200	0	0
1378	0.242	0	0
1381	0.254	0	0
1382	0.426	0	0
1393	0.638	1	3499
1394	0.015	0	0
1398	0.248	0	0
1404	0.471	0	0
1405	0.723	1	3895
1419	0.344	0	0
1421	0.129	0	0
1426	0.149	0	0
1431	0.446	0	0
1435	0.031	0	0
1437	0.407	0	0
1438	0.182	0	0
1442	0.518	1	4255
1464	0.087	0	0
1471	0.396	0	0
1473	0.195	0	0
1476	0.073	0	0
1478	0.254	0	0
1479	0.573	1	4615
1487	0.534	1	3405
1492	0.350	0	0
1496	0.102	0	0
1497	0.698	1	3697
1515	0.018	0	0
1519	0.152	0	0
1522	0.588	1	4230
1526	0.497	0	0
1537	0.036	0	0
1538	0.868	1	3811
1540	0.257	0	0
1543	0.130	0	0
1548	0.208	0	0
1549	0.094	0	0
1556	0.595	1	4074
1564	0.014	0	0
1570	0.172	0	0
1577	0.634	1	3569
1585	0.249	0	0
1590	0.210	0	0
1592	0.617	1	4218
1594	0.330	0	0
1596	0.492	0	0
1598	0.236	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
1603	0.384	0	0
1607	0.360	0	0
1612	0.059	0	0
1627	0.074	0	0
1629	0.722	1	4332
1630	0.105	0	0
1640	0.279	0	0
1641	0.297	0	0
1646	0.150	0	0
1662	0.535	1	3863
1668	0.123	0	0
1671	0.042	0	0
1672	0.657	1	3539
1673	0.403	0	0
1686	0.277	0	0
1688	0.691	1	3946
1696	0.022	0	0
1701	0.028	0	0
1707	0.102	0	0
1708	0.111	0	0
1713	0.058	0	0
1715	0.171	0	0
1717	0.035	0	0
1721	0.172	0	0
1724	0.875	1	4577
1725	0.714	1	4163
1730	0.176	0	0
1731	0.704	1	3192
1734	0.317	0	0
1740	0.112	0	0
1748	0.030	0	0
1749	0.061	0	0
1750	0.620	1	3614
1763	0.178	0	0
1768	0.062	0	0
1773	0.473	0	0
1777	0.206	0	0
1778	0.441	0	0
1780	0.107	0	0
1782	0.362	0	0
1784	0.054	0	0
1786	0.130	0	0
1787	0.147	0	0
1792	0.211	0	0
1800	0.549	1	4650
1801	0.184	0	0
1803	0.108	0	0
1804	0.624	1	3605
1807	0.028	0	0
1818	0.235	0	0
1821	0.026	0	0
1822	0.079	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
1828	0.078	0	0
1833	0.205	0	0
1844	0.335	0	0
1847	0.528	1	3589
1850	0.070	0	0
1854	0.310	0	0
1858	0.376	0	0
1864	0.126	0	0
1867	0.051	0	0
1876	0.774	1	3423
1880	0.204	0	0
1881	0.151	0	0
1891	0.259	0	0
1894	0.142	0	0
1895	0.063	0	0
1901	0.386	0	0
1905	0.061	0	0
1912	0.385	0	0
1918	0.320	0	0
1921	0.398	0	0
1923	0.251	0	0
1924	0.185	0	0
1931	0.113	0	0
1941	0.056	0	0
1950	0.079	0	0
1951	0.139	0	0
1954	0.009	0	0
1961	0.323	0	0
1966	0.010	0	0
1979	0.082	0	0
1982	0.050	0	0
1987	0.783	1	3732
1997	0.328	0	0
2004	0.034	0	0
2011	0.517	1	3766
2015	0.454	0	0
2025	0.013	0	0
2033	0.363	0	0
2034	0.008	0	0
2035	0.220	0	0
2036	0.612	1	3642
2053	0.620	1	4382
2059	0.653	1	4280
2060	0.030	0	0
2073	0.305	0	0
2084	0.401	0	0
2089	0.074	0	0
2092	0.275	0	0
2109	0.398	0	0
2129	0.354	0	0
2134	0.401	0	0
2135	0.025	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
2148	0.016	0	0
2149	0.029	0	0
2150	0.253	0	0
2165	0.784	1	4407
2166	0.035	0	0
2168	0.071	0	0
2170	0.292	0	0
2171	0.124	0	0
2172	0.040	0	0
2176	0.104	0	0
2182	0.055	0	0
2189	0.237	0	0
2191	0.071	0	0
2197	0.025	0	0
2202	0.172	0	0
2203	0.205	0	0
2204	0.616	1	4083
2206	0.713	1	4141
2218	0.071	0	0
2219	0.117	0	0
2221	0.419	0	0
2226	0.165	0	0
2228	0.339	0	0
2232	0.517	1	4284
2236	0.261	0	0
2241	0.745	1	4507
2245	0.056	0	0
2251	0.482	0	0
2255	0.036	0	0
2256	0.010	0	0
2259	0.010	0	0
2263	0.300	0	0
2264	0.037	0	0
2267	0.092	0	0
2273	0.765	1	4145
2277	0.708	1	3689
2287	0.154	0	0
2289	0.579	1	4482
2291	0.063	0	0
2296	0.885	1	3579
2299	0.097	0	0
2306	0.047	0	0
2314	0.390	0	0
2317	0.014	0	0
2318	0.556	1	4126
2321	0.850	1	3755
2324	0.028	0	0
2340	0.421	0	0
2343	0.028	0	0
2349	0.351	0	0
2352	0.327	0	0
2353	0.301	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
2365	0.892	1	3274
2370	0.676	1	3897
2378	0.288	0	0
2390	0.395	0	0
2399	0.254	0	0
2402	0.553	1	4302
2403	0.512	1	3920
2404	0.073	0	0
2414	0.092	0	0
2422	0.217	0	0
2424	0.176	0	0
2430	0.539	1	4130
2435	0.297	0	0
2439	0.046	0	0
2442	0.471	0	0
2445	0.294	0	0
2449	0.274	0	0
2451	0.360	0	0
2461	0.744	1	3712
2464	0.076	0	0
2465	0.445	0	0
2472	0.073	0	0
2476	0.304	0	0
2482	0.166	0	0
2487	0.303	0	0
2498	0.254	0	0
2501	0.095	0	0
2504	0.413	0	0
2511	0.276	0	0
2518	0.021	0	0
2521	0.177	0	0
2530	0.123	0	0
2543	0.558	1	4515
2545	0.365	0	0
2561	0.340	0	0
2566	0.546	1	4019
2572	0.195	0	0
2577	0.112	0	0
2578	0.194	0	0
2580	0.232	0	0
2581	0.231	0	0
2582	0.084	0	0
2584	0.033	0	0
2590	0.024	0	0
2598	0.007	0	0
2602	0.176	0	0
2605	0.008	0	0
2616	0.208	0	0
2618	0.176	0	0
2619	0.299	0	0
2624	0.032	0	0
2632	0.192	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
2640	0.208	0	0
2646	0.014	0	0
2651	0.098	0	0
2660	0.052	0	0
2661	0.047	0	0
2668	0.094	0	0
2670	0.362	0	0
2680	0.332	0	0
2681	0.021	0	0
2689	0.258	0	0
2694	0.074	0	0
2695	0.774	1	4431
2696	0.410	0	0
2702	0.043	0	0
2704	0.092	0	0
2708	0.036	0	0
2709	0.052	0	0
2714	0.416	0	0
2716	0.108	0	0
2723	0.065	0	0
2725	0.326	0	0
2738	0.088	0	0
2750	0.511	1	4030
2756	0.332	0	0
2758	0.069	0	0
2766	0.332	0	0
2767	0.322	0	0
2771	0.208	0	0
2775	0.315	0	0
2776	0.199	0	0
2779	0.928	1	4263
2780	0.376	0	0
2781	0.354	0	0
2782	0.472	0	0
2783	0.142	0	0
2796	0.315	0	0
2798	0.298	0	0
2800	0.069	0	0
2803	0.162	0	0
2806	0.003	0	0
2813	0.135	0	0
2818	0.131	0	0
2821	0.440	0	0
2825	0.340	0	0
2829	0.042	0	0
2830	0.621	1	4255
2833	0.075	0	0
2839	0.859	1	3315
2843	0.111	0	0
2846	0.049	0	0
2847	0.082	0	0
2848	0.132	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
2856	0.812	1	3536
2863	0.386	0	0
2867	0.291	0	0
2869	0.291	0	0
2873	0.006	0	0
2874	0.414	0	0
2875	0.514	1	3737
2880	0.618	1	3760
2886	0.568	1	4167
2887	0.342	0	0
2888	0.211	0	0
2889	0.758	1	3666
2890	0.439	0	0
2892	0.352	0	0
2901	0.141	0	0
2902	0.157	0	0
2905	0.239	0	0
2917	0.294	0	0
2922	0.519	1	4550
2924	0.109	0	0
2930	0.339	0	0
2931	0.077	0	0
2946	0.097	0	0
2955	0.347	0	0
2962	0.010	0	0
2964	0.025	0	0
2965	0.470	0	0
2967	0.018	0	0
2970	0.069	0	0
2973	0.532	1	3676
2974	0.213	0	0
2976	0.621	1	4628
2977	0.372	0	0
2978	0.214	0	0
2986	0.103	0	0
2988	0.260	0	0
2989	0.147	0	0
2995	0.665	1	4611
3005	0.500	0	0
3011	0.127	0	0
3013	0.047	0	0
3019	0.352	0	0
3021	0.033	0	0
3022	0.432	0	0
3029	0.211	0	0
3037	0.215	0	0
3042	0.256	0	0
3043	0.171	0	0
3049	0.246	0	0
3050	0.671	1	4450
3053	0.173	0	0
3058	0.349	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
3062	0.165	0	0
3063	0.265	0	0
3065	0.056	0	0
3080	0.125	0	0
3088	0.218	0	0
3093	0.472	0	0
3096	0.346	0	0
3101	0.318	0	0
3103	0.320	0	0
3107	0.268	0	0
3109	0.078	0	0
3111	0.061	0	0
3113	0.843	1	3489
3116	0.007	0	0
3132	0.168	0	0
3141	0.198	0	0
3153	0.299	0	0
3154	0.058	0	0
3160	0.135	0	0
3167	0.070	0	0
3170	0.396	0	0
3173	0.483	0	0
3174	0.306	0	0
3177	0.181	0	0
3179	0.188	0	0
3184	0.423	0	0
3190	0.207	0	0
3193	0.049	0	0
3199	0.311	0	0
3201	0.086	0	0
3202	0.158	0	0
3203	0.513	1	3239
3206	0.618	1	3299
3209	0.048	0	0
3210	0.462	0	0
3217	0.281	0	0
3220	0.074	0	0
3228	0.357	0	0
3232	0.024	0	0
3239	0.126	0	0
3243	0.526	1	4647
3245	0.116	0	0
3246	0.427	0	0
3251	0.060	0	0
3253	0.290	0	0
3257	0.086	0	0
3260	0.011	0	0
3261	0.186	0	0
3263	0.368	0	0
3278	0.210	0	0
3281	0.181	0	0
3283	0.077	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
3290	0.024	0	0
3297	0.296	0	0
3304	0.069	0	0
3305	0.453	0	0
3307	0.052	0	0
3308	0.419	0	0
3313	0.278	0	0
3314	0.186	0	0
3317	0.126	0	0
3348	0.108	0	0
3350	0.287	0	0
3359	0.028	0	0
3367	0.052	0	0
3376	0.095	0	0
3378	0.313	0	0
3384	0.776	1	4012
3386	0.134	0	0
3387	0.136	0	0
3388	0.105	0	0
3390	0.037	0	0
3391	0.423	0	0
3396	0.339	0	0
3398	0.022	0	0
3404	0.035	0	0
3406	0.027	0	0
3407	0.048	0	0
3414	0.048	0	0
3419	0.099	0	0
3423	0.590	1	3693
3427	0.041	0	0
3432	0.047	0	0
3434	0.049	0	0
3438	0.066	0	0
3442	0.227	0	0
3443	0.049	0	0
3448	0.067	0	0
3456	0.114	0	0
3464	0.104	0	0
3470	0.711	1	3107
3475	0.472	0	0
3477	0.376	0	0
3490	0.088	0	0
3493	0.285	0	0
3502	0.656	1	3480
3508	0.030	0	0
3516	0.095	0	0
3517	0.290	0	0
3525	0.217	0	0
3532	0.802	1	3660
3535	0.391	0	0
3536	0.719	1	3771
3540	0.089	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
3547	0.367	0	0
3550	0.502	1	4670
3557	0.682	1	3892
3562	0.186	0	0
3563	0.095	0	0
3564	0.288	0	0
3570	0.108	0	0
3573	0.461	0	0
3577	0.573	1	3820
3579	0.545	1	4023
3581	0.064	0	0
3587	0.356	0	0
3602	0.348	0	0
3609	0.500	1	4161
3612	0.097	0	0
3621	0.389	0	0
3642	0.184	0	0
3647	0.770	1	3837
3649	0.489	0	0
3654	0.442	0	0
3660	0.504	1	3802
3665	0.553	1	4054
3669	0.415	0	0
3673	0.324	0	0
3675	0.458	0	0
3678	0.132	0	0
3680	0.411	0	0
3686	0.625	1	4142
3693	0.237	0	0
3710	0.510	1	4074
3713	0.032	0	0
3718	0.316	0	0
3725	0.072	0	0
3726	0.286	0	0
3747	0.140	0	0
3753	0.023	0	0
3754	0.278	0	0
3760	0.811	1	4297
3763	0.043	0	0
3765	0.342	0	0
3769	0.178	0	0
3771	0.668	1	4293
3784	0.104	0	0
3787	0.177	0	0
3794	0.257	0	0
3796	0.056	0	0
3798	0.043	0	0
3809	0.125	0	0
3812	0.307	0	0
3819	0.269	0	0
3828	0.188	0	0
3831	0.235	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
3833	0.084	0	0
3837	0.832	1	3262
3839	0.820	1	3937
3843	0.378	0	0
3846	0.145	0	0
3854	0.036	0	0
3861	0.117	0	0
3864	0.331	0	0
3868	0.084	0	0
3869	0.145	0	0
3870	0.174	0	0
3883	0.140	0	0
3886	0.054	0	0
3889	0.332	0	0
3894	0.354	0	0
3907	0.034	0	0
3910	0.101	0	0
3913	0.020	0	0
3914	0.318	0	0
3921	0.262	0	0
3923	0.017	0	0
3929	0.453	0	0
3931	0.501	1	4612
3932	0.249	0	0
3937	0.625	1	3651
3943	0.308	0	0
3956	0.420	0	0
3957	0.524	1	3445
3961	0.530	1	3706
3971	0.216	0	0
4004	0.271	0	0
4005	0.056	0	0
4006	0.010	0	0
4011	0.119	0	0
4013	0.193	0	0
4014	0.121	0	0
4016	0.421	0	0
4017	0.016	0	0
4020	0.081	0	0
4022	0.111	0	0
4026	0.117	0	0
4032	0.123	0	0
4043	0.110	0	0
4045	0.378	0	0
4048	0.084	0	0
4051	0.082	0	0
4052	0.281	0	0
4056	0.041	0	0
4059	0.050	0	0
4069	0.044	0	0
4074	0.415	0	0
4076	0.279	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
4077	0.682	1	4351
4079	0.769	1	4032
4081	0.581	1	4447
4088	0.117	0	0
4105	0.141	0	0
4125	0.187	0	0
4134	0.583	1	3879
4139	0.004	0	0
4146	0.053	0	0
4149	0.083	0	0
4151	0.805	1	4144
4155	0.134	0	0
4157	0.101	0	0
4168	0.557	1	3788
4170	0.150	0	0
4174	0.069	0	0
4179	0.407	0	0
4185	0.126	0	0
4199	0.510	1	4219
4205	0.090	0	0
4208	0.022	0	0
4211	0.616	1	3885
4212	0.055	0	0
4215	0.627	1	4401
4217	0.113	0	0
4219	0.827	1	3963
4226	0.540	1	3283
4227	0.421	0	0
4229	0.031	0	0
4231	0.120	0	0
4233	0.008	0	0
4237	0.385	0	0
4243	0.422	0	0
4248	0.172	0	0
4255	0.238	0	0
4262	0.052	0	0
4266	0.698	1	3732
4268	0.275	0	0
4270	0.750	1	4392
4273	0.083	0	0
4276	0.096	0	0
4277	0.120	0	0
4279	0.219	0	0
4299	0.141	0	0
4313	0.051	0	0
4322	0.060	0	0
4324	0.067	0	0
4328	0.341	0	0
4331	0.351	0	0
4335	0.036	0	0
4337	0.473	0	0
4338	0.382	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
4343	0.062	0	0
4347	0.100	0	0
4355	0.702	1	4286
4357	0.005	0	0
4359	0.056	0	0
4362	0.148	0	0
4368	0.547	1	4160
4374	0.073	0	0
4375	0.507	1	3774
4378	0.457	0	0
4381	0.656	1	4499
4387	0.103	0	0
4400	0.014	0	0
4423	0.127	0	0
4424	0.058	0	0
4428	0.475	0	0
4433	0.635	1	4688
4436	0.469	0	0
4437	0.184	0	0
4439	0.368	0	0
4449	0.246	0	0
4456	0.085	0	0
4463	0.097	0	0
4467	0.088	0	0
4468	0.087	0	0
4469	0.066	0	0
4472	0.320	0	0
4473	0.021	0	0
4476	0.798	1	3695
4500	0.046	0	0
4509	0.176	0	0
4513	0.879	1	4008
4521	0.038	0	0
4527	0.444	0	0
4530	0.343	0	0
4532	0.511	1	4119
4533	0.111	0	0
4535	0.383	0	0
4536	0.432	0	0
4542	0.292	0	0
4551	0.710	1	4087
4554	0.061	0	0
4555	0.557	1	3252
4564	0.292	0	0
4572	0.437	0	0
4573	0.150	0	0
4577	0.302	0	0
4579	0.379	0	0
4583	0.081	0	0
4584	0.433	0	0
4596	0.049	0	0
4599	0.238	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
4607	0.344	0	0
4609	0.524	1	3925
4610	0.071	0	0
4616	0.390	0	0
4617	0.304	0	0
4633	0.237	0	0
4638	0.273	0	0
4641	0.043	0	0
4653	0.640	1	3966
4655	0.303	0	0
4659	0.386	0	0
4669	0.061	0	0
4678	0.085	0	0
4685	0.638	1	4287
4686	0.155	0	0
4691	0.247	0	0
4695	0.151	0	0
4698	0.129	0	0
4700	0.604	1	4346
4711	0.087	0	0
4722	0.029	0	0
4727	0.525	1	3620
4756	0.010	0	0
4762	0.267	0	0
4763	0.326	0	0
4766	0.071	0	0
4770	0.107	0	0
4784	0.312	0	0
4791	0.092	0	0
4795	0.034	0	0
4799	0.703	1	3884
4802	0.471	0	0
4805	0.642	1	3365
4814	0.569	1	3607
4816	0.324	0	0
4817	0.059	0	0
4822	0.234	0	0
4827	0.454	0	0
4833	0.115	0	0
4836	0.013	0	0
4842	0.178	0	0
4844	0.090	0	0
4845	0.327	0	0
4849	0.289	0	0
4850	0.276	0	0
4860	0.032	0	0
4863	0.252	0	0
4871	0.165	0	0
4878	0.442	0	0
4881	0.404	0	0
4888	0.465	0	0
4900	0.160	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
4906	0.353	0	0
4909	0.016	0	0
4916	0.096	0	0
4918	0.335	0	0
4926	0.364	0	0
4928	0.144	0	0
4941	0.410	0	0
4946	0.144	0	0
4949	0.046	0	0
4956	0.068	0	0
4966	0.037	0	0
4969	0.405	0	0
4973	0.121	0	0
4978	0.422	0	0
4982	0.354	0	0
4985	0.050	0	0
4991	0.114	0	0
4998	0.038	0	0
5000	0.504	1	4621
5004	0.358	0	0
5005	0.558	1	4229
5011	0.720	1	3595
5016	0.467	0	0
5018	0.058	0	0
5034	0.155	0	0
5038	0.024	0	0
5042	0.068	0	0
5046	0.086	0	0
5051	0.106	0	0
5054	0.206	0	0
5057	0.354	0	0
5062	0.061	0	0
5063	0.047	0	0
5065	0.068	0	0
5066	0.113	0	0
5076	0.225	0	0
5089	0.203	0	0
5092	0.641	1	3739
5093	0.692	1	3374
5094	0.032	0	0
5098	0.754	1	3433
5102	0.031	0	0
5112	0.267	0	0
5117	0.460	0	0
5127	0.531	1	3628
5130	0.288	0	0
5131	0.479	0	0
5132	0.512	1	3930
5135	0.782	1	3788
5136	0.022	0	0
5147	0.436	0	0
5157	0.078	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
5160	0.310	0	0
5165	0.012	0	0
5166	0.394	0	0
5172	0.575	1	3803
5173	0.183	0	0
5179	0.893	1	3213
5184	0.497	0	0
5187	0.051	0	0
5191	0.103	0	0
5193	0.147	0	0
5194	0.173	0	0
5199	0.178	0	0
5212	0.028	0	0
5213	0.498	0	0
5224	0.370	0	0
5226	0.107	0	0
5239	0.243	0	0
5252	0.722	1	4477
5264	0.203	0	0
5266	0.017	0	0
5271	0.019	0	0
5273	0.035	0	0
5276	0.651	1	3865
5278	0.060	0	0
5281	0.660	1	4428
5283	0.638	1	4311
5291	0.120	0	0
5294	0.300	0	0
5296	0.555	1	3444
5297	0.868	1	4533
5313	0.029	0	0
5314	0.403	0	0
5321	0.232	0	0
5325	0.009	0	0
5326	0.189	0	0
5328	0.014	0	0
5334	0.201	0	0
5338	0.461	0	0
5344	0.312	0	0
5348	0.258	0	0
5352	0.168	0	0
5353	0.063	0	0
5354	0.466	0	0
5361	0.876	1	3213
5364	0.028	0	0
5365	0.079	0	0
5367	0.512	1	3610
5379	0.417	0	0
5382	0.409	0	0
5386	0.269	0	0
5395	0.112	0	0
5410	0.340	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
5411	0.106	0	0
5416	0.398	0	0
5424	0.716	1	3717
5426	0.284	0	0
5428	0.149	0	0
5430	0.379	0	0
5433	0.173	0	0
5437	0.006	0	0
5440	0.282	0	0
5442	0.890	1	4110
5445	0.437	0	0
5449	0.138	0	0
5452	0.550	1	3498
5460	0.627	1	3439
5461	0.078	0	0
5465	0.169	0	0
5467	0.234	0	0
5471	0.326	0	0
5474	0.500	0	0
5475	0.024	0	0
5480	0.073	0	0
5481	0.178	0	0
5484	0.130	0	0
5494	0.104	0	0
5495	0.767	1	4059
5497	0.074	0	0
5499	0.406	0	0
5507	0.031	0	0
5510	0.114	0	0
5515	0.162	0	0
5516	0.039	0	0
5517	0.185	0	0
5524	0.056	0	0
5530	0.136	0	0
5534	0.331	0	0
5543	0.317	0	0
5545	0.571	1	3795
5558	0.181	0	0
5562	0.171	0	0
5573	0.728	1	4451
5581	0.188	0	0
5583	0.430	0	0
5587	0.540	1	3026
5589	0.835	1	4259
5591	0.192	0	0
5596	0.264	0	0
5606	0.834	1	3923
5608	0.176	0	0
5611	0.076	0	0
5612	0.205	0	0
5614	0.248	0	0
5620	0.050	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
5623	0.124	0	0
5624	0.131	0	0
5626	0.305	0	0
5633	0.146	0	0
5635	0.107	0	0
5640	0.444	0	0
5643	0.106	0	0
5644	0.417	0	0
5653	0.418	0	0
5663	0.021	0	0
5664	0.661	1	3432
5667	0.503	1	4362
5671	0.409	0	0
5673	0.590	1	4343
5676	0.106	0	0
5678	0.064	0	0
5698	0.243	0	0
5700	0.054	0	0
5705	0.305	0	0
5706	0.795	1	3880
5711	0.057	0	0
5712	0.875	1	3745
5716	0.409	0	0
5719	0.284	0	0
5725	0.888	1	3453
5728	0.122	0	0
5734	0.061	0	0
5735	0.082	0	0
5743	0.218	0	0
5754	0.158	0	0
5755	0.341	0	0
5756	0.082	0	0
5766	0.040	0	0
5770	0.590	1	3861
5774	0.147	0	0
5775	0.020	0	0
5776	0.208	0	0
5778	0.032	0	0
5786	0.644	1	3851
5787	0.331	0	0
5791	0.192	0	0
5794	0.181	0	0
5803	0.173	0	0
5804	0.209	0	0
5808	0.163	0	0
5810	0.027	0	0
5813	0.601	1	3605
5828	0.112	0	0
5839	0.348	0	0
5842	0.399	0	0
5843	0.039	0	0
5844	0.178	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
5847	0.556	1	4646
5851	0.017	0	0
5854	0.065	0	0
5857	0.018	0	0
5866	0.485	0	0
5874	0.383	0	0
5886	0.065	0	0
5895	0.064	0	0
5897	0.025	0	0
5898	0.206	0	0
5900	0.542	1	4511
5902	0.454	0	0
5908	0.647	1	3865
5909	0.021	0	0
5912	0.025	0	0
5913	0.104	0	0
5917	0.327	0	0
5918	0.572	1	4395
5921	0.177	0	0
5931	0.324	0	0
5942	0.485	0	0
5943	0.676	1	4188
5950	0.028	0	0
5954	0.005	0	0
5983	0.023	0	0
5995	0.669	1	4216
6002	0.096	0	0
6005	0.033	0	0
6009	0.208	0	0
6011	0.004	0	0
6012	0.013	0	0
6019	0.256	0	0
6021	0.376	0	0
6029	0.687	1	4515
6036	0.383	0	0
6037	0.006	0	0
6038	0.051	0	0
6043	0.038	0	0
6045	0.133	0	0
6047	0.731	1	3853
6048	0.034	0	0
6061	0.356	0	0
6063	0.198	0	0
6064	0.069	0	0
6068	0.661	1	4195
6069	0.063	0	0
6070	0.405	0	0
6071	0.174	0	0
6074	0.392	0	0
6079	0.371	0	0
6082	0.045	0	0
6088	0.793	1	3375

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
6094	0.147	0	0
6095	0.273	0	0
6098	0.468	0	0
6102	0.017	0	0
6105	0.443	0	0
6113	0.110	0	0
6116	0.252	0	0
6120	0.560	1	4282
6121	0.276	0	0
6126	0.218	0	0
6144	0.095	0	0
6145	0.035	0	0
6153	0.160	0	0
6156	0.173	0	0
6159	0.274	0	0
6162	0.052	0	0
6184	0.686	1	4236
6188	0.486	0	0
6189	0.334	0	0
6191	0.375	0	0
6211	0.471	0	0
6216	0.170	0	0
6218	0.580	1	3845
6222	0.186	0	0
6235	0.233	0	0
6245	0.169	0	0
6248	0.656	1	4483
6253	0.225	0	0
6256	0.005	0	0
6257	0.394	0	0
6259	0.309	0	0
6266	0.105	0	0
6268	0.291	0	0
6275	0.244	0	0
6280	0.632	1	4008
6283	0.361	0	0
6288	0.046	0	0
6289	0.112	0	0
6301	0.060	0	0
6308	0.244	0	0
6314	0.044	0	0
6315	0.151	0	0
6316	0.498	0	0
6317	0.495	0	0
6318	0.044	0	0
6323	0.741	1	3584
6329	0.599	1	3675
6336	0.161	0	0
6341	0.884	1	4102
6348	0.168	0	0
6349	0.029	0	0
6365	0.041	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
6372	0.251	0	0
6376	0.025	0	0
6378	0.068	0	0
6379	0.830	1	4082
6382	0.139	0	0
6383	0.396	0	0
6389	0.824	1	2981
6390	0.069	0	0
6392	0.062	0	0
6394	0.589	1	4291
6402	0.088	0	0
6404	0.346	0	0
6405	0.039	0	0
6406	0.281	0	0
6409	0.155	0	0
6410	0.155	0	0
6411	0.149	0	0
6421	0.061	0	0
6428	0.276	0	0
6429	0.345	0	0
6432	0.088	0	0
6436	0.042	0	0
6437	0.254	0	0
6438	0.210	0	0
6445	0.120	0	0
6447	0.564	1	3837
6450	0.050	0	0
6462	0.208	0	0
6467	0.671	1	4672
6478	0.040	0	0
6484	0.207	0	0
6492	0.417	0	0
6497	0.054	0	0
6504	0.254	0	0
6505	0.119	0	0
6513	0.612	1	3617
6525	0.212	0	0
6526	0.397	0	0
6528	0.054	0	0
6540	0.009	0	0
6542	0.110	0	0
6544	0.424	0	0
6548	0.133	0	0
6552	0.238	0	0
6558	0.007	0	0
6567	0.072	0	0
6569	0.534	1	4421
6572	0.127	0	0
6577	0.080	0	0
6581	0.308	0	0
6588	0.508	1	4305
6591	0.699	1	3356

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
6594	0.351	0	0
6600	0.520	1	4415
6602	0.208	0	0
6604	0.134	0	0
6605	0.035	0	0
6614	0.262	0	0
6616	0.405	0	0
6621	0.348	0	0
6640	0.362	0	0
6641	0.353	0	0
6643	0.048	0	0
6644	0.171	0	0
6649	0.432	0	0
6650	0.723	1	3872
6655	0.459	0	0
6661	0.019	0	0
6672	0.255	0	0
6677	0.051	0	0
6688	0.132	0	0
6689	0.081	0	0
6691	0.067	0	0
6692	0.347	0	0
6694	0.769	1	3563
6702	0.607	1	4150
6714	0.044	0	0
6716	0.507	1	3414
6724	0.100	0	0
6725	0.090	0	0
6730	0.165	0	0
6735	0.481	0	0
6738	0.342	0	0
6739	0.150	0	0
6743	0.242	0	0
6747	0.148	0	0
6750	0.714	1	4242
6751	0.564	1	3432
6753	0.559	1	3584
6754	0.316	0	0
6755	0.116	0	0
6762	0.177	0	0
6764	0.075	0	0
6772	0.651	1	3707
6774	0.093	0	0
6787	0.153	0	0
6789	0.027	0	0
6793	0.092	0	0
6798	0.009	0	0
6799	0.020	0	0
6800	0.102	0	0
6802	0.024	0	0
6808	0.279	0	0
6809	0.052	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
6812	0.013	0	0
6814	0.843	1	4198
6816	0.607	1	3628
6822	0.037	0	0
6829	0.287	0	0
6834	0.818	1	4075
6836	0.027	0	0
6839	0.139	0	0
6840	0.520	1	2981
6843	0.070	0	0
6846	0.626	1	4018
6848	0.013	0	0
6852	0.063	0	0
6856	0.177	0	0
6860	0.077	0	0
6866	0.327	0	0
6870	0.471	0	0
6878	0.624	1	3479
6880	0.128	0	0
6885	0.026	0	0
6897	0.053	0	0
6902	0.740	1	3598
6904	0.402	0	0
6907	0.070	0	0
6909	0.164	0	0
6914	0.540	1	3395
6915	0.455	0	0
6922	0.344	0	0
6924	0.168	0	0
6933	0.084	0	0
6934	0.087	0	0
6941	0.289	0	0
6957	0.215	0	0
6960	0.091	0	0
6969	0.070	0	0
6975	0.087	0	0
6980	0.762	1	3628
6983	0.094	0	0
6987	0.113	0	0
6994	0.044	0	0
6997	0.005	0	0
7002	0.108	0	0
7010	0.025	0	0
7015	0.569	1	3757
7019	0.305	0	0
7022	0.198	0	0
7025	0.037	0	0
7029	0.074	0	0
7031	0.205	0	0
7037	0.444	0	0
7038	0.137	0	0
7043	0.139	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
7049	0.050	0	0
7052	0.190	0	0
7053	0.334	0	0
7056	0.006	0	0
7057	0.626	1	3712
7080	0.190	0	0
7086	0.194	0	0
7087	0.105	0	0
7105	0.535	1	3732
7108	0.007	0	0
7121	0.544	1	2981
7122	0.263	0	0
7125	0.488	0	0
7132	0.304	0	0
7134	0.136	0	0
7151	0.244	0	0
7152	0.685	1	3867
7157	0.178	0	0
7159	0.191	0	0
7166	0.718	1	3679
7167	0.076	0	0
7177	0.038	0	0
7179	0.720	1	4291
7181	0.267	0	0
7183	0.155	0	0
7186	0.027	0	0
7193	0.016	0	0
7205	0.047	0	0
7207	0.024	0	0
7209	0.383	0	0
7216	0.267	0	0
7232	0.852	1	3606
7235	0.108	0	0
7238	0.391	0	0
7240	0.578	1	3874
7243	0.309	0	0
7252	0.296	0	0
7269	0.140	0	0
7275	0.023	0	0
7281	0.128	0	0
7283	0.059	0	0
7287	0.204	0	0
7289	0.338	0	0
7291	0.420	0	0
7294	0.024	0	0
7304	0.552	1	3347
7308	0.231	0	0
7313	0.044	0	0
7319	0.428	0	0
7325	0.108	0	0
7326	0.111	0	0
7330	0.317	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
7332	0.027	0	0
7337	0.429	0	0
7341	0.378	0	0
7346	0.571	1	4503
7353	0.634	1	3928
7354	0.751	1	4025
7361	0.416	0	0
7366	0.476	0	0
7368	0.033	0	0
7372	0.043	0	0
7375	0.491	0	0
7377	0.495	0	0
7380	0.103	0	0
7382	0.380	0	0
7385	0.769	1	3976
7392	0.661	1	4335
7395	0.090	0	0
7397	0.247	0	0
7403	0.070	0	0
7406	0.558	1	4565
7409	0.716	1	3667
7410	0.188	0	0
7412	0.067	0	0
7419	0.233	0	0
7425	0.133	0	0
7435	0.212	0	0
7438	0.275	0	0
7440	0.152	0	0
7447	0.101	0	0
7449	0.636	1	3852
7456	0.239	0	0
7464	0.131	0	0
7478	0.123	0	0
7480	0.046	0	0
7481	0.452	0	0
7483	0.198	0	0
7484	0.273	0	0
7491	0.561	1	4518
7494	0.465	0	0
7501	0.469	0	0
7503	0.783	1	4388
7509	0.270	0	0
7517	0.098	0	0
7518	0.168	0	0
7519	0.417	0	0
7521	0.650	1	3730
7522	0.490	0	0
7536	0.074	0	0
7539	0.019	0	0
7547	0.537	1	4301
7549	0.050	0	0
7552	0.414	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
7554	0.236	0	0
7556	0.059	0	0
7564	0.099	0	0
7566	0.237	0	0
7570	0.269	0	0
7571	0.018	0	0
7572	0.222	0	0
7575	0.129	0	0
7586	0.100	0	0
7589	0.041	0	0
7590	0.070	0	0
7597	0.268	0	0
7602	0.030	0	0
7604	0.316	0	0
7605	0.325	0	0
7612	0.788	1	3771
7615	0.076	0	0
7617	0.128	0	0
7624	0.109	0	0
7632	0.106	0	0
7639	0.356	0	0
7642	0.194	0	0
7643	0.217	0	0
7649	0.428	0	0
7650	0.425	0	0
7653	0.246	0	0
7654	0.293	0	0
7657	0.598	1	4243
7662	0.192	0	0
7669	0.797	1	4426
7671	0.013	0	0
7675	0.043	0	0
7678	0.188	0	0
7682	0.761	1	3700
7688	0.547	1	3874
7689	0.194	0	0
7690	0.149	0	0
7692	0.444	0	0
7699	0.294	0	0
7705	0.521	1	3887
7712	0.249	0	0
7726	0.643	1	3990
7728	0.106	0	0
7735	0.322	0	0
7737	0.731	1	3806
7739	0.053	0	0
7743	0.603	1	4237
7744	0.179	0	0
7746	0.241	0	0
7749	0.391	0	0
7750	0.299	0	0
7752	0.046	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
7755	0.139	0	0
7756	0.740	1	3782
7762	0.130	0	0
7764	0.683	1	3913
7769	0.092	0	0
7770	0.462	0	0
7776	0.113	0	0
7778	0.263	0	0
7784	0.464	0	0
7786	0.308	0	0
7789	0.201	0	0
7793	0.293	0	0
7794	0.083	0	0
7804	0.279	0	0
7811	0.187	0	0
7813	0.106	0	0
7815	0.272	0	0
7817	0.014	0	0
7818	0.232	0	0
7821	0.167	0	0
7825	0.038	0	0
7830	0.656	1	3609
7832	0.111	0	0
7835	0.011	0	0
7839	0.130	0	0
7842	0.045	0	0
7849	0.436	0	0
7856	0.333	0	0
7857	0.005	0	0
7863	0.064	0	0
7866	0.135	0	0
7871	0.064	0	0
7875	0.494	0	0
7882	0.760	1	4599
7887	0.460	0	0
7888	0.538	1	4119
7891	0.788	1	4566
7895	0.020	0	0
7901	0.307	0	0
7906	0.264	0	0
7908	0.831	1	4076
7917	0.231	0	0
7924	0.697	1	3928
7948	0.326	0	0
7950	0.736	1	4047
7955	0.187	0	0
7957	0.073	0	0
7959	0.161	0	0
7967	0.058	0	0
7969	0.042	0	0
7971	0.204	0	0
7974	0.318	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
7976	0.068	0	0
7986	0.848	1	3184
7987	0.688	1	3869
7993	0.295	0	0
7996	0.445	0	0
7998	0.336	0	0
8018	0.083	0	0
8019	0.227	0	0
8027	0.014	0	0
8036	0.110	0	0
8040	0.077	0	0
8044	0.097	0	0
8050	0.035	0	0
8052	0.529	1	4135
8054	0.391	0	0
8057	0.762	1	3213
8058	0.268	0	0
8059	0.511	1	4138
8066	0.841	1	3929
8070	0.053	0	0
8072	0.345	0	0
8078	0.025	0	0
8079	0.069	0	0
8080	0.530	1	2860
8081	0.152	0	0
8088	0.097	0	0
8091	0.666	1	4044
8094	0.135	0	0
8095	0.634	1	4281
8099	0.124	0	0
8101	0.252	0	0
8102	0.012	0	0
8116	0.465	0	0
8125	0.379	0	0
8134	0.190	0	0
8139	0.028	0	0
8141	0.060	0	0
8147	0.057	0	0
8158	0.353	0	0
8160	0.122	0	0
8165	0.427	0	0
8187	0.328	0	0
8205	0.341	0	0
8209	0.293	0	0
8211	0.201	0	0
8232	0.003	0	0
8236	0.072	0	0
8237	0.158	0	0
8238	0.772	1	3966
8245	0.366	0	0
8256	0.405	0	0
8268	0.053	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
8269	0.024	0	0
8270	0.334	0	0
8286	0.077	0	0
8289	0.054	0	0
8301	0.457	0	0
8305	0.149	0	0
8310	0.162	0	0
8312	0.032	0	0
8318	0.856	1	3948
8321	0.214	0	0
8328	0.072	0	0
8331	0.035	0	0
8334	0.421	0	0
8344	0.308	0	0
8345	0.228	0	0
8352	0.283	0	0
8358	0.452	0	0
8359	0.227	0	0
8360	0.144	0	0
8365	0.308	0	0
8366	0.085	0	0
8369	0.655	1	3677
8373	0.040	0	0
8378	0.123	0	0
8392	0.122	0	0
8397	0.479	0	0
8399	0.223	0	0
8400	0.087	0	0
8405	0.530	1	4270
8406	0.058	0	0
8410	0.141	0	0
8413	0.051	0	0
8414	0.216	0	0
8416	0.733	1	3704
8426	0.058	0	0
8434	0.313	0	0
8439	0.208	0	0
8440	0.262	0	0
8475	0.015	0	0
8480	0.148	0	0
8497	0.241	0	0
8499	0.712	1	4516
8500	0.329	0	0
8501	0.070	0	0
8502	0.519	1	3754
8518	0.454	0	0
8520	0.510	1	3502
8523	0.261	0	0
8525	0.136	0	0
8532	0.131	0	0
8535	0.264	0	0
8543	0.340	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
8554	0.249	0	0
8560	0.095	0	0
8561	0.255	0	0
8563	0.019	0	0
8566	0.875	1	3763
8570	0.295	0	0
8572	0.068	0	0
8582	0.108	0	0
8583	0.240	0	0
8587	0.218	0	0
8592	0.221	0	0
8593	0.442	0	0
8607	0.025	0	0
8609	0.168	0	0
8610	0.066	0	0
8614	0.244	0	0
8616	0.399	0	0
8622	0.158	0	0
8623	0.073	0	0
8624	0.193	0	0
8633	0.205	0	0
8641	0.338	0	0
8644	0.617	1	3824
8649	0.602	1	3816
8653	0.149	0	0
8657	0.165	0	0
8658	0.143	0	0
8663	0.129	0	0
8672	0.680	1	4314
8680	0.570	1	3778
8684	0.547	1	2860
8687	0.144	0	0
8688	0.116	0	0
8690	0.227	0	0
8712	0.251	0	0
8717	0.280	0	0
8730	0.048	0	0
8739	0.128	0	0
8744	0.020	0	0
8747	0.249	0	0
8748	0.340	0	0
8751	0.772	1	3982
8758	0.206	0	0
8761	0.358	0	0
8763	0.007	0	0
8764	0.307	0	0
8765	0.173	0	0
8773	0.104	0	0
8780	0.113	0	0
8781	0.247	0	0
8782	0.475	0	0
8785	0.137	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
8786	0.168	0	0
8797	0.808	1	4429
8799	0.064	0	0
8807	0.750	1	3963
8816	0.054	0	0
8817	0.077	0	0
8826	0.355	0	0
8833	0.182	0	0
8834	0.068	0	0
8835	0.127	0	0
8840	0.108	0	0
8843	0.088	0	0
8849	0.283	0	0
8855	0.117	0	0
8861	0.269	0	0
8862	0.169	0	0
8865	0.246	0	0
8868	0.008	0	0
8870	0.040	0	0
8880	0.298	0	0
8885	0.052	0	0
8894	0.219	0	0
8895	0.161	0	0
8899	0.031	0	0
8912	0.254	0	0
8922	0.011	0	0
8924	0.130	0	0
8928	0.223	0	0
8932	0.306	0	0
8943	0.130	0	0
8945	0.146	0	0
8946	0.039	0	0
8954	0.419	0	0
8958	0.463	0	0
8960	0.703	1	3830
8965	0.228	0	0
8966	0.087	0	0
8967	0.055	0	0
8969	0.302	0	0
8980	0.158	0	0
8984	0.032	0	0
8985	0.753	1	3746
8988	0.298	0	0
8989	0.342	0	0
8995	0.084	0	0
9004	0.053	0	0
9010	0.044	0	0
9012	0.235	0	0
9018	0.664	1	3645
9036	0.247	0	0
9037	0.245	0	0
9040	0.071	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
9041	0.341	0	0
9044	0.447	0	0
9045	0.095	0	0
9047	0.530	1	4149
9049	0.018	0	0
9061	0.012	0	0
9062	0.422	0	0
9076	0.239	0	0
9079	0.280	0	0
9081	0.298	0	0
9082	0.139	0	0
9089	0.632	1	3804
9092	0.188	0	0
9094	0.236	0	0
9115	0.028	0	0
9117	0.404	0	0
9118	0.287	0	0
9120	0.089	0	0
9124	0.015	0	0
9128	0.146	0	0
9135	0.411	0	0
9136	0.707	1	4087
9138	0.299	0	0
9157	0.452	0	0
9176	0.056	0	0
9183	0.275	0	0
9187	0.506	1	3863
9188	0.225	0	0
9190	0.173	0	0
9197	0.024	0	0
9200	0.038	0	0
9201	0.183	0	0
9203	0.017	0	0
9212	0.374	0	0
9213	0.064	0	0
9214	0.151	0	0
9217	0.296	0	0
9219	0.007	0	0
9220	0.146	0	0
9221	0.091	0	0
9237	0.011	0	0
9240	0.092	0	0
9241	0.006	0	0
9248	0.224	0	0
9253	0.457	0	0
9259	0.549	1	4424
9267	0.111	0	0
9271	0.303	0	0
9273	0.210	0	0
9285	0.041	0	0
9290	0.298	0	0
9291	0.079	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
9293	0.091	0	0
9294	0.083	0	0
9301	0.188	0	0
9302	0.063	0	0
9312	0.018	0	0
9316	0.386	0	0
9319	0.585	1	3107
9328	0.069	0	0
9331	0.895	1	2981
9338	0.033	0	0
9350	0.299	0	0
9356	0.113	0	0
9359	0.449	0	0
9362	0.472	0	0
9364	0.135	0	0
9370	0.408	0	0
9380	0.134	0	0
9386	0.142	0	0
9394	0.433	0	0
9407	0.373	0	0
9411	0.602	1	4424
9422	0.274	0	0
9423	0.264	0	0
9429	0.261	0	0
9433	0.155	0	0
9439	0.024	0	0
9451	0.211	0	0
9452	0.370	0	0
9453	0.013	0	0
9460	0.015	0	0
9465	0.026	0	0
9470	0.050	0	0
9476	0.493	0	0
9485	0.604	1	3808
9486	0.065	0	0
9488	0.265	0	0
9507	0.009	0	0
9508	0.373	0	0
9517	0.254	0	0
9521	0.146	0	0
9528	0.108	0	0
9532	0.488	0	0
9536	0.146	0	0
9540	0.148	0	0
9542	0.100	0	0
9546	0.244	0	0
9548	0.179	0	0
9549	0.131	0	0
9554	0.139	0	0
9555	0.358	0	0
9558	0.026	0	0
9573	0.745	1	3980

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
9575	0.555	1	4173
9584	0.715	1	4237
9586	0.099	0	0
9588	0.113	0	0
9591	0.432	0	0
9592	0.644	1	4499
9597	0.539	1	4211
9600	0.080	0	0
9603	0.617	1	4097
9605	0.360	0	0
9614	0.686	1	4150
9616	0.020	0	0
9622	0.613	1	3708
9624	0.150	0	0
9629	0.399	0	0
9633	0.090	0	0
9640	0.236	0	0
9644	0.409	0	0
9645	0.501	1	3658
9646	0.110	0	0
9648	0.945	1	3262
9649	0.054	0	0
9660	0.207	0	0
9664	0.485	0	0
9675	0.058	0	0
9679	0.779	1	3751
9680	0.366	0	0
9682	0.068	0	0
9697	0.015	0	0
9701	0.217	0	0
9704	0.375	0	0
9705	0.211	0	0
9707	0.321	0	0
9714	0.107	0	0
9718	0.092	0	0
9722	0.139	0	0
9739	0.127	0	0
9747	0.719	1	3637
9751	0.237	0	0
9757	0.141	0	0
9759	0.018	0	0
9760	0.089	0	0
9764	0.596	1	4350
9776	0.315	0	0
9778	0.179	0	0
9786	0.070	0	0
9803	0.644	1	3986
9804	0.067	0	0
9815	0.126	0	0
9824	0.016	0	0
9825	0.148	0	0
9826	0.279	0	0

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
9827	0.024	0	0
9833	0.084	0	0
9835	0.089	0	0
9860	0.350	0	0
9865	0.190	0	0
9871	0.211	0	0
9874	0.293	0	0
9880	0.211	0	0
9882	0.440	0	0
9885	0.072	0	0
9888	0.640	1	3838
9892	0.051	0	0
9893	0.286	0	0
9896	0.252	0	0
9902	0.102	0	0
9906	0.085	0	0
9910	0.482	0	0
9914	0.317	0	0
9918	0.463	0	0
9920	0.251	0	0
9926	0.243	0	0
9931	0.071	0	0
9935	0.348	0	0
9945	0.916	1	3672
9953	0.234	0	0
9957	0.007	0	0
9963	0.128	0	0
9972	0.242	0	0
9976	0.370	0	0
9979	0.297	0	0
9980	0.017	0	0
9982	0.138	0	0
9991	0.702	1	4109
10000	0.242	0	0
10003	0.175	0	0
10005	0.877	1	3890
10014	0.019	0	0
10032	0.288	0	0
10034	0.302	0	0
10041	0.008	0	0
10042	0.026	0	0
10044	0.075	0	0
10045	0.289	0	0
10054	0.553	1	4067
10061	0.187	0	0
10062	0.498	0	0
10073	0.217	0	0
10081	0.048	0	0
10084	0.431	0	0
10086	0.166	0	0
10093	0.369	0	0
10101	0.505	1	4208

INDEX	TARGET_FLAG_PROB	TARGET_FLAG	TARGET_AMT
10105	0.377	0	0
10110	0.432	0	0
10113	0.621	1	4726
10115	0.641	1	3851
10119	0.492	0	0
10121	0.494	0	0
10124	0.789	1	3883
10126	0.191	0	0
10127	0.026	0	0
10145	0.117	0	0
10147	0.541	1	3547
10148	0.009	0	0
10162	0.305	0	0
10163	0.026	0	0
10166	0.779	1	3814
10172	0.056	0	0
10173	0.395	0	0
10175	0.034	0	0
10180	0.158	0	0
10186	0.044	0	0
10192	0.312	0	0
10199	0.277	0	0
10209	0.929	1	3650
10210	0.145	0	0
10214	0.047	0	0
10215	0.204	0	0
10216	0.668	1	4189
10232	0.230	0	0
10239	0.258	0	0
10249	0.046	0	0
10253	0.413	0	0
10255	0.104	0	0
10262	0.075	0	0
10264	0.033	0	0
10266	0.156	0	0
10268	0.209	0	0
10271	0.140	0	0
10272	0.149	0	0
10276	0.481	0	0
10277	0.034	0	0
10279	0.515	1	3107
10281	0.017	0	0
10285	0.008	0	0
10294	0.317	0	0
10300	0.107	0	0

Part 1. Data Exploration

```
library(alr3)
library(car)
library(ggplot2)
library(dplyr)
library(knitr)
library(lmtest)
library(plyr)
library(psych)
```

Data Summary

```
# original data set
url <- "https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/insurance_training_data.csv"
hw4 <- read.csv(url, stringsAsFactors = FALSE)

# remove $ signs from variables
hw4$INCOME <- as.numeric(gsub("[,$]", "", hw4$INCOME))
hw4$HOME_VAL <- as.numeric(gsub("[,$]", "", hw4$HOME_VAL))
hw4$BLUEBOOK <- as.numeric(gsub("[,$]", "", hw4$BLUEBOOK))
hw4$OLDCLAIM <- as.numeric(gsub("[,$]", "", hw4$OLDCLAIM))

hw4$INCOME <- as.numeric(as.character(hw4$INCOME))
hw4$HOME_VAL <- as.numeric(as.character(hw4$HOME_VAL))
hw4$BLUEBOOk <- as.numeric(as.character(hw4$BLUEBOOK))
hw4$OLDCLAIM <- as.numeric(as.character(hw4$OLDCLAIM))


# transformed data
url <- "https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/621-HW4-XFORMED-DATA.csv"
hw4.t <- read.csv(url, stringsAsFactors = FALSE)
```

NUMERIC VARIABLES table

```
# get descripitive statistics of original data set

hw4.c <- hw4[,-c(1,2)]
hw4.n <- hw4.c[sapply(hw4.c, is.numeric)]
d <- describe(hw4.n)
d$mean <- round(d$mean,0)
d$sd <- round(d$sd,0)
d$min <- round(d$min,0)
d$max <- round(d$max,0)
d$range <- round(d$range,0)
d$skew <- round(d$skew,0)
d$kurtosis <- round(d$kurtosis,0)
d <- d[, -c(1,6,7)]
kable(d,digits=0)
```

Box + barplots

```
# box plots of each predictor variable relative to the response
# See Figure 8.8 on page 286

# ---------------------------------------------------------------------------

# Boxplots and barplots for each numeric and categorical variable

attach(hw4.t)

par(mfrow=c(3,3), oma=c(1,1,1,1), mar=c(3,3,5,3))

# get number of TARGET_FLAG == 1/0 responses
TF_POS <- sum(TARGET_FLAG == 1)
TF_NEG <- sum(TARGET_FLAG == 0)

# box plots of each predictor variable relative to the response

# See Figure 8.8 on page 286
#
# ---------------------------------------------------------------------
# CATEGORICAL: KIDSDRIV
# how many KIDSDRIV == TRUE??
s.KD <- sum(KIDSDRIV > 0)
# there are 981

KD.NO.TPOS <-  nrow(subset(hw4.t, KIDSDRIV == 0 & TARGET_FLAG == 1))
KD.YES.TPOS <- nrow(subset(hw4.t, KIDSDRIV > 0 & TARGET_FLAG == 1))


#KIDSDRIV == TRUE is 0.387 correlated with TARGET_FLAG
KDY.InAcc <- KD.YES.TPOS / s.KD

# now get proportion of KIDSDRIV == FALSE involved in accidents
KDN.InAcc <- KD.NO.TPOS / (nrow(hw4.t) - s.KD)

rel_percs <- c(KDN.InAcc, KDY.InAcc )

mp <- barplot(rel_percs, names.arg = c('KIDSDRIV = 0', 'KIDSDRIV = 1'),
        main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
        xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75 )

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
        pos = 3, cex = .75)


# ------------------------------------------------------------------------

boxplot(AGE ~ TARGET_FLAG, ylab="AGE",
       main = "Was In a Car Crash? (1 = YES, 0 = NO): AGE", col = "yellow",
       xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

# ---------------------------------------------------------------------
```

```r
# CATEGORICAL: HOMEKIDS
# how many HOMEKIDS == TRUE??
s.HK <- sum(HOMEKIDS > 0)
# there are 2872

HK.NO.TPOS <-  nrow(subset(hw4.t, HOMEKIDS == 0 & TARGET_FLAG == 1))
HK.YES.TPOS <- nrow(subset(hw4.t, HOMEKIDS > 0 & TARGET_FLAG == 1))


# HOMEKIDS == TRUE is 0.4419 correlated with TARGET_FLAG
HKY.InAcc <- HK.YES.TPOS / s.HK

# now get proportion of non-single parents involved in accidents
HKN.InAcc <- HK.NO.TPOS / (nrow(hw4.t) - s.HK)

rel_percs <- c(HKN.InAcc, HKY.InAcc )


mp <- barplot(rel_percs, names.arg = c('HOMEKIDS = 0', 'HOMEKIDS = 1'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
          pos = 3, cex = .75)

# ------------------------------------------------------------------------

boxplot(YOJ ~ TARGET_FLAG, ylab="YOJ",
        main="Was In a Car Crash? (1 = YES, 0 = NO): YOJ", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

boxplot(INCOME ~ TARGET_FLAG, ylab="INCOME",
        main="Was In a Car Crash? (1 = YES, 0 = NO): INCOME", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

# -----------------------------------------

# -----------------------------------------
# CATEGORICAL: PARENT1

# how many PARENT1 == Yes??
s.Par1 <- sum(PARENT1 == 'Yes')
# there are 1077

PARENT1.NO.TPOS <-  nrow(subset(hw4.t, PARENT1 == 'No' & TARGET_FLAG == 1))
PARENT1.YES.TPOS <- nrow(subset(hw4.t, PARENT1 == 'Yes' & TARGET_FLAG == 1))

# get ratios of PARENT1.NO.POS / s.Par1
# THIS IS the proportion of single parents that had car accidents ->
# PARENT1 == YES is 0.4419 correlated with TARGET_FLAG
Par1Y.InAcc <- PARENT1.YES.TPOS / s.Par1
```

```r
# now get proportion of non-single parents involved in accidents
Par1N.InAcc <- PARENT1.NO.TPOS / (nrow(hw4.t) - s.Par1)

rel_percs <- c(Par1N.InAcc, Par1Y.InAcc )


mp <- barplot(rel_percs, names.arg = c('Parent1 = No', 'Parent1 = Yes'),
         main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
         xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)

# -----------------------------------------------------------------------

boxplot(HOME_VAL ~ TARGET_FLAG, ylab="HOME_VAL",
        main="Was In a Car Crash? (1 = YES, 0 = NO): HOME_VAL", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

# -----------------------------------------
# CATEGORICAL: MSTATUS

# how many MSTATUS == Yes??
s.MStat <- sum(MSTATUS == 'Yes')
# there are 4894

MSTATUS.NO.TPOS <-  nrow(subset(hw4.t, MSTATUS == 'z_No' & TARGET_FLAG == 1))
MSTATUS.YES.TPOS <- nrow(subset(hw4.t, MSTATUS == 'Yes' & TARGET_FLAG == 1))

# get ratios of MSTATUS.YES.TPOS / s.MStat
# THIS IS the proportion of married people that had car accidents ->
# MSTATUS of YES is 0.2151 correlated with TARGET_FLAG
MstatY.InAcc <- MSTATUS.YES.TPOS / s.MStat

# now get proportion of non-single parents involved in accidents
MstatN.InAcc <- MSTATUS.NO.TPOS / (nrow(hw4.t) - s.MStat)

rel_percs <- c(MstatN.InAcc, MstatY.InAcc )

mp <- barplot(rel_percs, names.arg = c('MSTATUS = No', 'MSTATUS = Yes'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)

# ----------------------------------------------------------------
# CATEGORICAL; SEX

# how many SEX == M??
s.SEX <- sum(SEX == 'M')
```

```r
# there are 3786 M's

SEX.F.TPOS <-  nrow(subset(hw4.t, SEX == 'z_F' & TARGET_FLAG == 1))
SEX.M.TPOS <- nrow(subset(hw4.t, SEX == 'M' & TARGET_FLAG == 1))

# get ratios
# THIS IS the proportion of males that had car accidents ->
# SEX.M  is 0.2538 correlated with TARGET_FLAG
SEX.M.Y.InAcc <- SEX.M.TPOS / s.SEX

# now get proportion of females involved in accidents
SEX.F.Y.InAcc <- SEX.F.TPOS / (nrow(hw4.t) - s.SEX)

rel_percs <- c(SEX.M.Y.InAcc, SEX.F.Y.InAcc )

mp <- barplot(rel_percs, names.arg = c('SEX = M', 'SEX = F'),
         main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
         xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)
```

```r
par(mfrow=c(3,3), oma=c(1,1,1,1), mar=c(3,3,3,3))
# ------------------------------------------------------------------------
# CATEGORICAL: EDUCATION

# how many EDUCATION for each category?
s.EDU.LHS <- sum(EDUCATION == '<High School')
# there are 1203 <High School

s.EDU.HS <- sum(EDUCATION == 'z_High School')
# there are 2330 z_High School

s.EDU.B <- sum(EDUCATION == 'Bachelors')
# there are 2242 Bachelors

s.EDU.M <- sum(EDUCATION == 'Masters')
# there are 1658 Masters

s.EDU.P <- sum(EDUCATION == 'PhD')
# there are 728 PhD

# now get counts of past accidents for each category
EDU.LHS.TPOS <- nrow(subset(hw4.t, EDUCATION == '<High School' & TARGET_FLAG == 1))
EDU.HS.TPOS <- nrow(subset(hw4.t, EDUCATION == 'z_High School' & TARGET_FLAG == 1))
EDU.B.TPOS <- nrow(subset(hw4.t, EDUCATION == 'Bachelors' & TARGET_FLAG == 1))
EDU.M.TPOS <- nrow(subset(hw4.t, EDUCATION == 'Masters' & TARGET_FLAG == 1))
EDU.P.TPOS <- nrow(subset(hw4.t, EDUCATION == 'PhD' & TARGET_FLAG == 1))

# get ratios for each category
EDU.LHS.InAcc <- EDU.LHS.TPOS / s.EDU.LHS
EDU.HS.InAcc <- EDU.HS.TPOS / s.EDU.HS
```

```
EDU.B.InAcc <- EDU.B.TPOS / s.EDU.B
EDU.M.InAcc <- EDU.M.TPOS / s.EDU.M
EDU.P.InAcc <- EDU.P.TPOS / s.EDU.P

rel_percs <- c(EDU.LHS.InAcc, EDU.HS.InAcc, EDU.B.InAcc, EDU.M.InAcc, EDU.P.InAcc)

mp <- barplot(rel_percs, names.arg = c('< HS', 'HS', 'Bachelors', 'Masters', 'PhD'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,6),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75, las=2)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)


# --------------------------------------------------------------------------
# CATEGORICAL: JOB

# how many JOB for each category?
s.J.BC <- sum(JOB == 'z_Blue Collar')
# there are 1825 Blue Collar

s.J.PRO <- sum(JOB == 'Professional')
# there are 1117 Professionals

s.J.MGR <- sum(JOB == 'Manager')
# there are 988 Managers

s.J.HM <- sum(JOB == 'Home Maker')
# there are 641 Home Makers

s.J.CLR <- sum(JOB == 'Clerical')
# there are 1271 Clerical

s.J.DOC <- sum(JOB == 'Doctor')
# there are 246 Doctor

s.J.LAW <- sum(JOB == 'Lawyer')
# there are 835 Lawyer

s.J.STU <- sum(JOB == 'Student')
# there are 712 Students


# now get counts of past accidents for each category
J.BC.TPOS <- nrow(subset(hw4.t, JOB == 'z_Blue Collar' & TARGET_FLAG == 1))
J.PRO.TPOS <- nrow(subset(hw4.t, JOB == 'Professional' & TARGET_FLAG == 1))
J.MGR.TPOS <- nrow(subset(hw4.t, JOB == 'Manager' & TARGET_FLAG == 1))
J.HM.TPOS <- nrow(subset(hw4.t, JOB == 'Home Maker' & TARGET_FLAG == 1))
J.CLR.TPOS <- nrow(subset(hw4.t, JOB == 'Clerical' & TARGET_FLAG == 1))
J.DOC.TPOS <- nrow(subset(hw4.t, JOB == 'Doctor' & TARGET_FLAG == 1))
J.LAW.TPOS <- nrow(subset(hw4.t, JOB == 'Lawyer' & TARGET_FLAG == 1))
J.STU.TPOS <- nrow(subset(hw4.t, JOB == 'Student' & TARGET_FLAG == 1))
```

```r
# get ratios for each category
J.BC.InAcc <- J.BC.TPOS / s.J.BC
J.PRO.InAcc <- J.PRO.TPOS / s.J.PRO
J.MGR.InAcc <- J.MGR.TPOS / s.J.MGR
J.HM.InAcc <- J.HM.TPOS / s.J.HM
J.CLR.InAcc <- J.CLR.TPOS / s.J.CLR
J.DOC.InAcc <- J.DOC.TPOS / s.J.DOC
J.LAW.InAcc <- J.LAW.TPOS / s.J.LAW
J.STU.InAcc <- J.STU.TPOS / s.J.STU

rel_percs <- c(J.BC.InAcc, J.PRO.InAcc, J.MGR.InAcc, J.HM.InAcc, J.CLR.InAcc, J.DOC.InAcc,
               J.LAW.InAcc, J.STU.InAcc)

mp <- barplot(rel_percs, names.arg = c('Blue Collar.', 'Professional', 'Manager', 'Home Maker', 'Clerica
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow', las=2,
          xlim=c(0,8),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75, las=2)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)


# ---------------------------------------------------------------------------

boxplot(TRAVTIME ~ TARGET_FLAG, ylab="TRAVTIME",
        main="Was In a Car Crash? (1 = YES, 0 = NO): TRAVTIME", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

# ----------------------------------------------------------------------
# CATEGORICAL: CAR_USE

# how many CAR_USE == Private??
s.CAR_USE <- sum(CAR_USE == 'Private')
# there are 5132 Private cars

CAR_USE.P.TPOS <-  nrow(subset(hw4.t, CAR_USE == 'Private' & TARGET_FLAG == 1))
CAR_USE.C.TPOS <- nrow(subset(hw4.t, CAR_USE == 'Commercial' & TARGET_FLAG == 1))

# get ratios
# THIS IS the proportion of private cars that had car accidents ->
# Private car  is 0.2155 correlated with TARGET_FLAG
CU.P.InAcc <- CAR_USE.P.TPOS / s.CAR_USE

# now get proportion of commercial vehicles involved in accidents
# 0.3456
CU.C.InAcc <- CAR_USE.C.TPOS / (nrow(hw4.t) - s.CAR_USE)

rel_percs <- c(CU.P.InAcc, CU.C.InAcc )

mp <- barplot(rel_percs, names.arg = c('CAR_USE = Private', 'CAR_USE = Commercial'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)
```

```
# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)

# -----------------------------------------------



boxplot(BLUEBOOK ~ TARGET_FLAG, ylab="BLUEBOOK",
        main="Was In a Car Crash? (1 = YES, 0 = NO): BLUEBOOK", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

boxplot(TIF ~ TARGET_FLAG, ylab="TIF",
        main="Was In a Car Crash? (1 = YES, 0 = NO): TIF", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )


# ------------------------------------------------------------------------
# CATEGORICAL: CAR_TYPE

# how many CAR_TYPE for each category?
s.CT.Mini <- sum(CAR_TYPE == 'Minivan')
# there are 2145 Minivans

s.CT.SUV <- sum(CAR_TYPE == 'z_SUV')
# there are 2294 SUV's

s.CT.SC <- sum(CAR_TYPE == 'Sports Car')
# there are 907 Sports Cars

s.CT.Van <- sum(CAR_TYPE == 'Van')
# there are 750 Vans

s.CT.PT <- sum(CAR_TYPE == 'Panel Truck')
# there are 676 Panel Trucks

s.CT.PU <- sum(CAR_TYPE == 'Pickup')
# there are 1389 Pickups


# now get counts of past accidents for each category
CT.Mini.TPOS <- nrow(subset(hw4.t, CAR_TYPE == 'Minivan' & TARGET_FLAG == 1))
CT.SUV.TPOS <- nrow(subset(hw4.t, CAR_TYPE == 'z_SUV' & TARGET_FLAG == 1))
CT.SC.TPOS <- nrow(subset(hw4.t, CAR_TYPE == 'Sports Car' & TARGET_FLAG == 1))
CT.Van.TPOS <- nrow(subset(hw4.t, CAR_TYPE == 'Van' & TARGET_FLAG == 1))
CT.PT.TPOS <- nrow(subset(hw4.t, CAR_TYPE == 'Panel Truck' & TARGET_FLAG == 1))
CT.PU.TPOS <- nrow(subset(hw4.t, CAR_TYPE == 'Pickup' & TARGET_FLAG == 1))

# get ratios for each category
CT.Mini.InAcc <- CT.Mini.TPOS / s.CT.Mini
CT.SUV.InAcc <- CT.SUV.TPOS / s.CT.SUV
CT.SC.InAcc <- CT.SC.TPOS / s.CT.SC
```

```r
CT.Van.InAcc <- CT.Van.TPOS / s.CT.Van
CT.PT.InAcc <- CT.PT.TPOS / s.CT.PT
CT.PU.InAcc <- CT.PU.TPOS / s.CT.PU

rel_percs <- c(CT.Mini.InAcc, CT.SUV.InAcc, CT.SC.InAcc, CT.Van.InAcc, CT.PT.InAcc, CT.PU.InAcc)

mp <- barplot(rel_percs, names.arg = c('Minivan', 'SUV', 'Sports Car', 'Van', 'Panel Truck', 'Pickup'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,8),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75, las=2)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)


# ---------------------------------------------------------------------------
# CATEGORICAL: RED_CAR

# how many RED_CAR == yes??
s.RED_CAR <- sum(RED_CAR == 'yes')
# there are 2378 red cars

RED_CAR.Y.TPOS <-  nrow(subset(hw4.t, RED_CAR == 'yes' & TARGET_FLAG == 1))
RED_CAR.N.TPOS <- nrow(subset(hw4.t, RED_CAR == 'no' & TARGET_FLAG == 1))

# get ratios
# THIS IS the proportion of red cars that had car accidents ->
# red car  is 0.259 correlated with TARGET_FLAG
RC.Y.InAcc <- RED_CAR.Y.TPOS / s.RED_CAR

# now get proportion of non-red cars involved in accidents
# 0.26.5779
RC.N.InAcc <- RED_CAR.N.TPOS / (nrow(hw4.t) - s.RED_CAR)

rel_percs <- c(RC.N.InAcc, RC.Y.InAcc )

mp <- barplot(rel_percs, names.arg = c('RED_CAR = no', 'RED_CAR = yes'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)
```

```r
par(mfrow=c(3,3), oma=c(1,1,1,1), mar=c(3,3,3,3))

# ----------------------------------------------------------------------------

boxplot(OLDCLAIM ~ TARGET_FLAG, ylab="OLDCLAIM",
        main="Was In a Car Crash? (1 = YES, 0 = NO): OLDCLAIM", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

boxplot(CLM_FREQ ~ TARGET_FLAG, ylab="CLM_FREQ",
```

```
        main="Was In a Car Crash? (1 = YES, 0 = NO): CLAIM_FREQ", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

# ------------------------------------------------------------------------
# CATEGORICAL: REVOKED

# how many REVOKED == yes??
s.REV <- sum(REVOKED == 'Yes')
# there are 1000 revoked's

REV.Y.TPOS <-  nrow(subset(hw4.t, REVOKED == 'Yes' & TARGET_FLAG == 1))
REV.N.TPOS <- nrow(subset(hw4.t, REVOKED == 'No' & TARGET_FLAG == 1))

# get ratios
# THIS IS the proportion of REVOKED's that had car accidents ->
# REVOKED  is 0.443 correlated with TARGET_FLAG
RV.Y.InAcc <- REV.Y.TPOS / s.REV

# now get proportion of non-REVOKED's involved in accidents
# 0.23879
RV.N.InAcc <- REV.N.TPOS / (nrow(hw4.t) - s.REV)

rel_percs <- c(RV.N.InAcc, RV.Y.InAcc )

mp <- barplot(rel_percs, names.arg = c('REVOKED = No', 'REVOKED = Yes'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)


boxplot(MVR_PTS ~ TARGET_FLAG, ylab="MVR_PTS",
        main="Was In a Car Crash? (1 = YES, 0 = NO): MVR_PTS", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

boxplot(CAR_AGE ~ TARGET_FLAG, ylab="CAR_AGE",
        main="Was In a Car Crash? (1 = YES, 0 = NO): CAR_AGE", col = "yellow",
        xlim=c(0,3),  width=c(1,1), cex.main=.75, cex.lab=.75, cex.axis=0.75 )

# ----------------------------------------------------------------------------------------
# CATEGORICAL: URBANICITY

# how many URBANICITY == Highly Urban/ Urban??
s.URB <- sum(URBANICITY == 'Highly Urban/ Urban')
# there are 6492 urbans

URB.Y.TPOS <-  nrow(subset(hw4.t, URBANICITY == 'Highly Urban/ Urban' & TARGET_FLAG == 1))
URB.N.TPOS <- nrow(subset(hw4.t, URBANICITY != 'Highly Urban/ Urban' & TARGET_FLAG == 1))

# get ratios
# THIS IS the proportion of URBANICITY's that had car accidents ->
```

```r
# URBANICITY  is 0.3139 correlated with TARGET_FLAG
URB.Y.InAcc <- URB.Y.TPOS / s.URB

# now get proportion of rural URBANICITY's involved in accidents
# 0.0689
URB.N.InAcc <- URB.N.TPOS / (nrow(hw4.t) - s.URB)

rel_percs <- c(URB.N.InAcc, URB.Y.InAcc )

mp <- barplot(rel_percs, names.arg = c('URBANICITY = Rural', 'URBANICITY = Urban'),
          main = ('Proportions w/ Past Accidents'), ylim = c(0, 1), col = 'yellow',
          xlim=c(0,4),  width=c(2,2), cex.main=.75, cex.lab=.75, cex.axis=0.75, cex.names=.75)

# write the percentage values above the individual bars in the plot
text(mp, rel_percs, labels = format(round(rel_percs, 3), 4),
     pos = 3, cex = .6)

# ------------------------------------------------------------------------
```

Histograms

```r
attach(hw4.t)

par(mfrow=c(4,5), oma=c(2,2,2,2), mar=c(2,2,2,2))

# Make small histograms for each variable

df <- hw4.t[sapply(hw4.t, is.numeric)]
#colnames(df)
hist(df$TARGET_FLAG, main="TARGET_FLAG",col="yellow",cex.main=.75)
hist(df$TARGET_AMT, main="TARGET_AMT",col="yellow",cex.main=.75, breaks = 20)
hist(df$KIDSDRIV, main="KIDSDRIV",col="yellow",cex.main=.75)
hist(df$AGE, main="AGE",col="yellow",cex.main=.75)
hist(df$HOMEKIDS, main="HOMEKIDS",col="yellow",cex.main=.75)
hist(df$YOJ, main="YOJ",col="yellow",cex.main=.75)
hist(df$INCOME, main="INCOME",col="yellow",cex.main=.75, breaks = 20)
hist(df$HOME_VAL, main="HOME_VAL",col="yellow",cex.main=.75, breaks = 20)
hist(df$TRAVTIME, main="TRAVTIME",col="yellow",cex.main=.75)
hist(df$BLUEBOOK, main="BLUEBOOK",col="yellow",cex.main=.75, breaks = 20)
hist(df$TIF, main="TIF",col="yellow",cex.main=.75)
hist(df$OLDCLAIM, main="OLDCLAIM",col="yellow",cex.main=.75, breaks = 20)
hist(df$CLM_FREQ, main="CLM_FREQ",col="yellow",cex.main=.75)
hist(df$MVR_PTS, main="MVR_PTS",col="yellow",cex.main=.75)
hist(df$CAR_AGE, main="CAR_AGE",col="yellow",cex.main=.75)
hist(df$NEW_CAR, main="NEW_CAR",col="yellow",cex.main=.75)
hist(df$H_RENTER, main="H_RENTER",col="yellow",cex.main=.75)
```

Part 2. Data Preparation

```
library(plyr)
library(pander)
library(knitr)

hw4 <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/insurance_training_
```

```
hw4$PARENT1 <- factor(hw4$PARENT1)
hw4$MSTATUS <- factor(hw4$MSTATUS)
hw4$SEX <- factor(hw4$SEX)
hw4$EDUCATION <- factor(hw4$EDUCATION)
hw4$JOB <- factor(hw4$JOB)
hw4$CAR_USE <- factor(hw4$CAR_USE)
hw4$CAR_TYPE <- factor(hw4$CAR_TYPE)
hw4$RED_CAR <- factor(hw4$RED_CAR)
hw4$REVOKED <- factor(hw4$REVOKED)
hw4$URBANICITY <- factor(hw4$URBANICITY)
```

convert INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM to integers

```
# remove $ signs from variables
hw4$INCOME <- as.numeric(gsub("[,$]", "", hw4$INCOME))
hw4$HOME_VAL <- as.numeric(gsub("[,$]", "", hw4$HOME_VAL))
hw4$BLUEBOOK <- as.numeric(gsub("[,$]", "", hw4$BLUEBOOK))
hw4$OLDCLAIM <- as.numeric(gsub("[,$]", "", hw4$OLDCLAIM))

hw4$INCOME <- as.numeric(as.character(hw4$INCOME))
hw4$HOME_VAL <- as.numeric(as.character(hw4$HOME_VAL))
hw4$BLUEBOOK <- as.numeric(as.character(hw4$BLUEBOOK))
hw4$OLDCLAIM <- as.numeric(as.character(hw4$OLDCLAIM))
```

Impute function

```
impute <- function (a, a.impute){
  ifelse (is.na(a), a.impute,a)
}

hw4.1 <- hw4[,-c(1:3)]
```

Step 1: Imputation for age

Impute age with median since there are only 6 missing

```
hw4.1$AGE[is.na(hw4.1$AGE)] <- median(hw4.1$AGE, na.rm=TRUE)
```

Step 2: Imputation for home value

For the home values with NA it makes sense to impute them as 0 and assume they are renters since there are already a large percentage of renters.

```
hw4.1$HOME_VAL[is.na(hw4.1$HOME_VAL)] <- 0
```

Step 3: Imputation for Job

For the job types that are blank we will create a new value of "None Specified"

```
hw4.1$JOB <- as.character(hw4.1$JOB)
d <- ifelse(nchar(hw4.1$JOB)==0, "None Specified", hw4.1$JOB)
hw4.1$JOB <- as.factor(d)
#summary(hw4.1$JOB)
```

Step 4: Imputation for car age

Build a linear model to impute car age using subtraction. Also take absolute value of the negative value assuming typo (only 1 instance)

```
car.age <- lm(data=hw4.1, CAR_AGE~.)
#summary(car.age)

#remove single parent, old claim, claim freq and urban
car.age1 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY)
#summary(car.age1)

#remove job
car.age2 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB)
#summary(car.age2)

#remove car use
car.age3 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE)
#summary(car.age3)

#remove kids at home
car.age4 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HOM
# summary(car.age4)

#remove travel time
car.age5 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HOM
#summary(car.age5)

#remove marital status
car.age6 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HOM
#summary(car.age6)

#remove Years on JOb
car.age7 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HOM
#summary(car.age7)

#remove kids driving
car.age8 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HOM
#summary(car.age8)
```

```
#remove car type
car.age9 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HOM
#summary(car.age9)

#remove blue book
car.age10 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age10)

#remove age
car.age11 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age11)

#remove revoked
car.age12 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age12)

#remove MVR points
car.age13 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age13)

#remove red car
car.age14 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age14)

#remove sex
car.age15 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age15)

#remove TIF
car.age16 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age16)

#remove Income
car.age17 <- lm(data=hw4.1, CAR_AGE~. - PARENT1 - OLDCLAIM - CLM_FREQ - URBANICITY - JOB - CAR_USE - HO
#summary(car.age17)

# plot(car.age17$residuals)
pred.carage <- round(predict(car.age17, hw4.1))

carage.Imp <- impute(hw4$CAR_AGE, pred.carage)

hw4.1$CAR_AGE <- carage.Imp
 #assume negative number is a typo so take absolute value
hw4.1$CAR_AGE <- abs(hw4.1$CAR_AGE)
```

Step 5: Impute for Income (Part 1)

Impute missing values for Income except for the 29 values that have both YOJ and income missing. THis process is justified in that these values do show YOJ values.

```
Inc <- lm(data=hw4.1, INCOME~.)
#summary(Inc)
```

```r
#eliminate car type
Inc1 <- lm(data=hw4.1, INCOME~.-CAR_TYPE)
#summary(Inc1)

#eliminate revoked
Inc2 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED)
#summary(Inc2)

#eliminate red car
Inc3 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR)
#summary(Inc3)

#eliminate old claim
Inc4 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM)
#summary(Inc4)

#eliminate car use
Inc5 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE)
#summary(Inc5)

#eliminate home kids
Inc6 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS)
#summary(Inc6)

#eliminate claim freq
Inc7 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS - CLM_FR
#summary(Inc7)

#eliminate TIF
Inc8 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS - CLM_FR
#summary(Inc8)

#eliminate travel time
Inc9 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS - CLM_FR
#summary(Inc9)

#eliminate urban city
Inc10 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS - CLM_F
#summary(Inc10)

#eliminate sex
Inc11 <- lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS - CLM_F
#summary(Inc11)

#eliminate kids drive
Inc12 <-  lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS - CLM_
#summary(Inc12)

#eliminate car age
Inc13 <-  lm(data=hw4.1, INCOME~. - CAR_TYPE - REVOKED - RED_CAR - OLDCLAIM - CAR_USE - HOMEKIDS - CLM_
#summary(Inc13)

#All p values look good
```

```
# plot(Inc13$residuals)
pred.inc <- round(predict(Inc13, hw4.1))

Inc.Imp <- impute(hw4$INCOME, pred.inc)

hw4.1$INCOME <- Inc.Imp

#fix negative income imputed

hw4.1$INCOME[which(hw4.1$INCOME<0)] <- 0
```

Step 6: Impute YOJ (Part 1)

Impute values for all values except for the 29 that have blanks for Income

```
yoj <- lm(data=hw4.1, YOJ~.)
#summary(yoj)

#eliminate car type, car age, red car
yoj1 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE)
#summary(yoj1)

#eliminate TIF
yoj2 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF)
#summary(yoj2)

#eliminate home value
yoj3 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL)
#summary(yoj3)

#eliminate travel time
yoj4 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME)
#summary(yoj4)

#eliminate revoked
yoj5 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED)
#summary(yoj5)

#eliminate claim freq
yoj6 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED - CLM_F
#summary(yoj6)

#eliminate urban
yoj7 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED - CLM_F
#summary(yoj7)

#eliminate bluebook
yoj8 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED - CLM_F
#summary(yoj8)

#eliminate old claim
yoj9 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED - CLM_F
```

```r
#summary(yoj9)

#eliminate single parent
yoj10 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED - CLM_
#summary(yoj10)

#eliminate sex
yoj11 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED - CLM_
#summary(yoj11)

#eliminate car use
yoj12 <- lm(data=hw4.1, YOJ~.- CAR_TYPE - RED_CAR - CAR_AGE - TIF - HOME_VAL - TRAVTIME - REVOKED - CLM_
#summary(yoj12)

# plot(yoj12$residuals)

pred.yoj <- round(predict(yoj12, hw4.1))

yoj.imp <- impute(hw4$YOJ, pred.yoj)

hw4.1$YOJ <- yoj.imp
```

Step 7: Impute for YOJ and Income when both are missing.

Since both YOJ and Income are blank it is reasonable to assume that the 29 rows with both blank have no income. Fix remaining 29 rows with NA for both Income and YOJ with 0

```r
hw4.1$INCOME[is.na(hw4.1$INCOME)] <- 0


hw4.1$YOJ[is.na(hw4.1$YOJ)] <- 0

# hw4.1 is missing first 3 columns of info
hw4.t <- cbind(hw4[,c(1:3)], hw4.1)

hw4 <- hw4.t
```

Create NEW_CAR variable

```r
# build NEW_CAR column
# first check count of CAR_AGE <= 1 entries: there are 1937
sum(hw4$CAR_AGE <= 1)

hw4.s$NEW_CAR <- 0
hw4.s$NEW_CAR[hw4$CAR_AGE <= 1] <- 1

# make sure new column matches original
sum(hw4.s$NEW_CAR == 1)
```

Create H_RENTER variable

```r
# build H_RENTER column
# first check count of HOME_VAL == 0 entries: there are 2758
sum(hw4$HOME_VAL == 0)

hw4.s$H_RENTER <- 0
hw4.s$H_RENTER[hw4$HOME_VAL == 0] <- 1

# make sure new column matches original count
sum(hw4.s$H_RENTER == 1)
```

Convert HOMEKIDS + KIDSDRIV to binary

```r
hw4.s$HOMEKIDS[hw4$HOMEKIDS > 0] <- 1

hw4.s$KIDSDRIV[hw4$KIDSDRIV > 0] <- 1
```

Add a JOB_COLOR variable to the data set: white =

```r
b<- as.character(hw4.s$JOB)
b[which(b=="Doctor")] <- "White"
b[which(b=="Clerical")] <- "White"
b[which(b=="Lawyer")] <- "White"
b[which(b=="Manager")] <- "White"
b[which(b=="Professional")] <- "White"
b[which(b=="None Specified")] <- "White"
b[which(b=="Student")] <- "Blue"
b[which(b=="Home Maker")] <- "Blue"
b[which(b=="z_Blue Collar")] <- "Blue"
b <- as.factor(b)
hw4.s$JOB_COLOR <- b
```

Now write updated data set to a file

```r
write.csv(hw4.s, file = "C:/SQLData/621-HW4-XFORMED-DATA.csv", row.names = FALSE)
```

Part 3. Build Models

Binary Model 1

```r
# library(bestglm)
library(alr3)
library(car)
library(pROC)
options(scipen=999)

hw4 <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/621-HW4-XFORMED-DAT
```

Convert categoricals with strings to factors

```
hw4$PARENT1 <- factor(hw4$PARENT1)
hw4$MSTATUS <- factor(hw4$MSTATUS)
hw4$SEX <- factor(hw4$SEX)
hw4$EDUCATION <- factor(hw4$EDUCATION)
hw4$JOB <- factor(hw4$JOB)
hw4$CAR_USE <- factor(hw4$CAR_USE)
hw4$CAR_TYPE <- factor(hw4$CAR_TYPE)
hw4$RED_CAR <- factor(hw4$RED_CAR)
hw4$REVOKED <- factor(hw4$REVOKED)
hw4$URBANICITY <- factor(hw4$URBANICITY)
hw4$KIDSDRIV <- factor(hw4$KIDSDRIV)
hw4$HOMEKIDS <- factor(hw4$HOMEKIDS)
hw4$H_RENTER <- factor(hw4$H_RENTER)
hw4$NEW_CAR <- factor(hw4$NEW_CAR)
hw4$JOB_COLOR <- factor(hw4$JOB_COLOR)
```

Try using step() function

```
bm1.init <- glm(TARGET_FLAG ~ . - INDEX - TARGET_AMT - HOME_VAL - CAR_AGE - JOB, family = binomial(link


summary(bm1.init)

# use STEP function to find best BIC model
bm1 <- step(bm1.init, trace=0, k=log(nrow(hw4)))

summary(bm1)


# now just recreate output of step function to ensure it matches:
m1.bic <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + INCOME + MSTATUS +
    EDUCATION + TRAVTIME + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
    OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS + URBANICITY + H_RENTER,
    family = binomial(link = "logit"), data = hw4)

summary(m1.bic)
```

Now check the marginal model plots

```
# par(mfrow=c(2,4))

mmps(m1.bic,layout=c(2,4),key=TRUE)
```

Skew shown for INCOME, TRAVTIME, BLUEBOOK, CLM_FREQ, MVR_PTS. Take log of all and refit

```
# add logs of skewed vars
m2.bic <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + INCOME + log(INCOME + 1) + MSTATUS +
    EDUCATION + TRAVTIME + log(TRAVTIME) + CAR_USE + BLUEBOOK + log(BLUEBOOK) + TIF + CAR_TYPE +
    OLDCLAIM + CLM_FREQ + log(CLM_FREQ + 1) + REVOKED + MVR_PTS + log(MVR_PTS + 1) + URBANICITY + H_REN
    family = binomial(link = "logit"), data = hw4)

summary(m2.bic)
```

Check results for statistical significance and refit - results say remove BLUEBOOK

```
m3.bic <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + INCOME + log(INCOME + 1) + MSTATUS +
    EDUCATION + TRAVTIME + log(TRAVTIME) + CAR_USE + log(BLUEBOOK) + TIF + CAR_TYPE +
    OLDCLAIM + CLM_FREQ + log(CLM_FREQ + 1) + REVOKED + MVR_PTS + log(MVR_PTS + 1) + URBANICITY + H_REN
    family = binomial(link = "logit"), data = hw4)

summary(m3.bic)
```

Remove log(MVR_PTS)

```
m4.bic <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + INCOME + log(INCOME + 1) + MSTATUS +
    EDUCATION + TRAVTIME + log(TRAVTIME) + CAR_USE + log(BLUEBOOK) + TIF + CAR_TYPE +
    OLDCLAIM + CLM_FREQ + log(CLM_FREQ + 1) + REVOKED + MVR_PTS + URBANICITY + H_RENTER,
    family = binomial(link = "logit"), data = hw4)

summary(m4.bic)
```

remove TRAVTIME

```
m5.bic <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + INCOME + log(INCOME + 1) + MSTATUS +
    EDUCATION + log(TRAVTIME) + CAR_USE + log(BLUEBOOK) + TIF + CAR_TYPE +
    OLDCLAIM + CLM_FREQ + log(CLM_FREQ + 1) + REVOKED + MVR_PTS + URBANICITY + H_RENTER,
    family = binomial(link = "logit"), data = hw4)

summary(m5.bic)
```

STOP

Check marginal model plots

```
# par(mfrow=c(2,4))

mmps(m5.bic,layout=c(3,4),key=TRUE)
```

Add back log(MVR_PTS), remove MVR_PTS and INCOME

```
m6.bic <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + log(INCOME + 1) + MSTATUS +
    EDUCATION + log(TRAVTIME) + CAR_USE + log(BLUEBOOK) + TIF + CAR_TYPE +
    OLDCLAIM + CLM_FREQ + log(CLM_FREQ + 1) + REVOKED + log(MVR_PTS + 1) + URBANICITY + H_RENTER,
    family = binomial(link = "logit"), data = hw4)

summary(m6.bic)
```

Try removing CLM_FREQ to check effect on AIC: slight decrease in AIC but mmps deteriorate so DON'T DO THIS!!!

```
m7.bic <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + log(INCOME + 1) + MSTATUS +
    EDUCATION + log(TRAVTIME) + CAR_USE + log(BLUEBOOK) + TIF + CAR_TYPE +
    OLDCLAIM + log(CLM_FREQ + 1) + REVOKED + log(MVR_PTS + 1) + URBANICITY + H_RENTER,
    family = binomial(link = "logit"), data = hw4)

summary(m7.bic)
```

Check marginal model plots

```r
# par(mfrow=c(2,4))

mmps(m6.bic,layout=c(3,4),key=TRUE)
```

STOP.

Copy m6.bic to m1 for outlier tests

```r
m1 <- m6.bic
```

Now test model: First load functions for metrics

```r
# Load R functions for model statistics

accuracy <- function(actual, predicted){

  # Equation to be modeled: (TP + TN) / (TP + FP + TN + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( (TP + TN) / (TP + FP + TN + FN) )
}
```

```r
classif.err.rate <- function(actual, predicted) {

  # Equation to be modeled: (FP + FN) / (TP + FP + TN + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( (FP + FN) / (TP + FP + TN + FN) )
}
```

```r
precision <- function(actual, predicted) {

  # Precision : the proportion of positive cases that were correctly identified.
```

```r
  # Equation to be modeled: TP / (TP + FP)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TP / (TP + FP) )
}
```

```r
sensitivity <- function(actual, predicted) {

  # Equation to be modeled: TP / (TP + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TP / (TP + FN) )
}
```

```r
specificity <- function(actual, predicted) {

  # Equation to be modeled: TN / (TN + FP)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TN / (TN + FP) )
}
```

```r
F1.Score <- function(actual, predicted) {

  # Equation to be modeled: ( 2 * precision * sensitivity) / (precision + sensitivity)
```

```
  # now calculate the required metric
  return( ( 2 * precision(actual, predicted) * sensitivity(actual, predicted))
          / (precision(actual, predicted) + sensitivity(actual, predicted)) )
}
```

Check for outliers: This MUST be done by hand - the identify function requires that you click on points that are of interest to you so that it can label them. Does not seem possible to use this in a writeup.

```
#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m1)$hat
stanresDeviance <- residuals(m1)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '18' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 18 / nrow(hw4),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.t)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw3.names, cex=0.75)
```

Now run metrics

```
# Coefficient Interpretation

# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar <- mean(dlogis(predict(m1, type = "link")))
LogitScalar * coef(m1)

# Logit model predicted probabilities - yields likelihood that each eval item is '+'
#
predprob.crash<- round(predict(m1, type="response"), 2)
summary(predprob.crash)

# Percent correctly predicted values
# NOTE: Need to create variable 'Y' for this to work - set it to response variable
Y <- hw4$TARGET_FLAG

pred.crash <- round(fitted(m1))

table(true = Y, pred = pred.crash)


# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, pred.crash)
classif.err.rate(Y, pred.crash)
```

```
precision(Y, pred.crash)
sensitivity(Y, pred.crash)
specificity(Y, pred.crash)
F1.Score(Y, pred.crash)

# get AUC
rocCurve <- roc(response= Y, predictor= pred.crash)
auc(rocCurve)
```

Binary Model 2

```
library(car)
library(pROC)
hw4t <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/621-HW4-XFORMED-DA

hw4t$PARENT1 <- as.factor(hw4t$PARENT1)
hw4t$MSTATUS <- as.factor(hw4t$MSTATUS)
hw4t$SEX <- as.factor(hw4t$SEX)
hw4t$EDUCATION <- as.factor(hw4t$EDUCATION)
hw4t$JOB <- as.factor(hw4t$JOB)
hw4t$CAR_USE <- as.factor(hw4t$CAR_USE)
hw4t$CAR_TYPE <- as.factor(hw4t$CAR_TYPE)
hw4t$RED_CAR <- as.factor(hw4t$RED_CAR)
hw4t$REVOKED <- as.factor(hw4t$REVOKED)
hw4t$URBANICITY <- as.factor(hw4t$URBANICITY)
hw4t$JOB_COLOR <- as.factor(hw4t$JOB_COLOR)
hw4t$KIDSDRIV <- as.factor(hw4t$KIDSDRIV)
hw4t$H_RENTER <- as.factor(hw4t$H_RENTER)
hw4t$NEW_CAR <- as.factor(hw4t$NEW_CAR)
hw4t$HOMEKIDS <- as.factor(hw4t$HOMEKIDS)

###eliminate variables we agreed to leave out.
hw4t.1 <- hw4t

#step 1:  start to look at all options
mod <- glm(data=hw4t.1, TARGET_FLAG~.- INDEX - TARGET_AMT - YOJ- HOME_VAL - SEX - TRAVTIME - JOB - RED_
summary(mod)

#step 2:  eliminate CAR_AGE
mod1 <- glm(data=hw4t.1, TARGET_FLAG~. - INDEX - TARGET_AMT - YOJ- HOME_VAL - SEX - TRAVTIME - JOB - RE
summary(mod1)

#step 3:  eliminate AGE
mod2 <- glm(data=hw4t.1, TARGET_FLAG~. - INDEX - TARGET_AMT - YOJ- HOME_VAL - SEX - TRAVTIME - JOB - RE
summary(mod2)

#step 4:  eliminate new_car
mod3 <- glm(data=hw4t.1, TARGET_FLAG~. - INDEX - TARGET_AMT - YOJ- HOME_VAL - SEX - TRAVTIME - JOB - RE
summary(mod3)
```

```
#step5: eliminate HS education except for Bachelors, Masters and PhD
mod4 <- glm(data=hw4t.1, TARGET_FLAG~KIDSDRIV+ INCOME + PARENT1 + MSTATUS  + CAR_USE + BLUEBOOK + TIF +
summary(mod4)

#step6: eliminate old claim
mod5 <- glm(data=hw4t.1, TARGET_FLAG~KIDSDRIV+ INCOME + PARENT1 + MSTATUS  + CAR_USE + BLUEBOOK + TIF +
summary(mod5)

#step7 : eliminate single parent
mod6 <- glm(data=hw4t.1, TARGET_FLAG~KIDSDRIV+ INCOME + MSTATUS  + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
summary(mod6)

#step8 : eliminate job color
mod7 <- glm(data=hw4t.1, TARGET_FLAG~KIDSDRIV+ INCOME + MSTATUS  + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
summary(mod7)

#Step9:  create college variable to simplify
b<- as.character(hw4t.1$EDUCATION)
b[which(b=="<High School")] <- "Not College"
b[which(b=="z_High School")] <- "Not College"
b[which(b=="Bachelors")] <- "College"
b[which(b=="Masters")] <- "College"
b[which(b=="PhD")] <- "College"
b <- as.factor(b)
hw4t.1$ED_LEVEL <- b

#Step 10: eliminate individual college values in place of ED_LEVEL
mod8 <- glm(data=hw4t.1, TARGET_FLAG~KIDSDRIV+ INCOME + MSTATUS  + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
summary(mod8)

#look at mmps to see if there are data issues
mmps(mod8,layout=c(4,4),key=TRUE)

#mmps shows issues with Income, Bluebook, MVR_PTS

#Step 11:  Box Cox Transformations
library(car)
library(MASS)
summary(powerTransform(BLUEBOOK~TARGET_FLAG, hw4t.1, family="bcPower"))
boxcox(hw4t.1$BLUEBOOK~hw4t.1$TARGET_FLAG)
#use sqrt for BLUEBOOK

summary(powerTransform(INCOME+1~TARGET_FLAG, hw4t.1, family="bcPower"))
boxcox(hw4t.1$INCOME+1~hw4t.1$TARGET_FLAG)
#use sqrt for INCOME

summary(powerTransform(MVR_PTS +1~TARGET_FLAG, hw4t.1, family="bcPower"))
boxcox(hw4t.1$MVR_PTS + 1 ~hw4t.1$TARGET_FLAG)
#use 1/sqrt for MVR_PTS

#Step12:  New model with transformed data
mod9 <- glm(data=hw4t.1, TARGET_FLAG~KIDSDRIV+ sqrt(INCOME) + MSTATUS  + CAR_USE + sqrt(BLUEBOOK) + TIF
summary(mod9)
```

```r
mmps(mod9,layout=c(4,3),key=TRUE)

#MVR_PTS still looks off, try log instead

#Step13:  Log of MVR_PTS
mod10 <- glm(data=hw4t.1, TARGET_FLAG~KIDSDRIV+ sqrt(INCOME) + MSTATUS  + CAR_USE + sqrt(BLUEBOOK) + TI
summary(mod10)



mmps(mod10,layout=c(4,3),key=TRUE)
```

Load functions

```r
accuracy <- function(actual, predicted){

  # Equation to be modeled: (TP + TN) / (TP + FP + TN + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( (TP + TN) / (TP + FP + TN + FN) )
}
```

```r
classif.err.rate <- function(actual, predicted) {

  # Equation to be modeled: (FP + FN) / (TP + FP + TN + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( (FP + FN) / (TP + FP + TN + FN) )
}
```

```r
precision <- function(actual, predicted) {

  # Precision : the proportion of positive cases that were correctly identified.

```

```
  # Equation to be modeled: TP / (TP + FP)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TP / (TP + FP) )
}
```

```
sensitivity <- function(actual, predicted) {

  # Equation to be modeled: TP / (TP + FN)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TP / (TP + FN) )
}
```

```
specificity <- function(actual, predicted) {

  # Equation to be modeled: TN / (TN + FP)

  # derive confusion matrix cell values
  c.mat <- data.frame(table(actual, predicted))

  # extract all four confusion matrix values from the data frame
  TN <- as.numeric(as.character(c.mat[1,3]))
  FN <- as.numeric(as.character(c.mat[2,3]))
  FP <- as.numeric(as.character(c.mat[3,3]))
  TP <- as.numeric(as.character(c.mat[4,3]))

  # now calculate the required metric
  return( TN / (TN + FP) )
}
```

```
F1.Score <- function(actual, predicted) {

  # Equation to be modeled: ( 2 * precision * sensitivity) / (precision + sensitivity)

```

```
  # now calculate the required metric
  return( ( 2 * precision(actual, predicted) * sensitivity(actual, predicted))
          / (precision(actual, predicted) + sensitivity(actual, predicted)) )
}
```

Check for outliers: This MUST be done by hand - the identify function requires that you click on points that are of interest to you so that it can label them. Does not seem possible to use this in a writeup.

```
#Figure 8.13 on page 291
m1 <- mod10
par(mfrow=c(1,1))
hvalues <- influence(m1)$hat
stanresDeviance <- residuals(m1)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '17' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2* 17 / nrow(hw4),lty=2)

hw3.names <- as.character(seq(1:nrow(hw3.t)))

# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw4t.names, cex=0.75)
```

no highly leveraged outliers

Now run metrics

```
# Coefficient Interpretation
m1 <- mod10
# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar <- mean(dlogis(predict(m1, type = "link")))
LogitScalar * coef(m1)

# Logit model predicted probabilities - yields likelihood that each eval item is '+'
#
predprob.crash<- round(predict(m1, type="response"), 2)
summary(predprob.crash)

# Percent correctly predicted values
# NOTE: Need to create variable 'Y' for this to work - set it to response variable
Y <- hw4t$TARGET_FLAG

pred.crash <- round(fitted(m1))

table(true = Y, pred = pred.crash)


# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r
```

```
# now use functions built in HW 2 to get required statistics
accuracy(Y, pred.crash)
classif.err.rate(Y, pred.crash)
precision(Y, pred.crash)
sensitivity(Y, pred.crash)
specificity(Y, pred.crash)
F1.Score(Y, pred.crash)

# get AUC
rocCurve <- roc(response= Y, predictor= pred.crash)
auc(rocCurve)
```

Binary Model 3

Load Training Data

```
hw4.t <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/621-HW4-XFORMED-DA

hw4.t0 <-hw4.t

attach(hw4.t)

hw4.t$PARENT1 <- as.factor(hw4.t$PARENT1)
hw4.t$MSTATUS <- as.factor(hw4.t$MSTATUS)
hw4.t$SEX <- as.factor(hw4.t$SEX)
hw4.t$EDUCATION <- as.factor(hw4.t$EDUCATION)
hw4.t$JOB <- as.factor(hw4.t$JOB)
hw4.t$CAR_USE <- as.factor(hw4.t$CAR_USE)
hw4.t$CAR_TYPE <- as.factor(hw4.t$CAR_TYPE)
hw4.t$RED_CAR <- as.factor(hw4.t$RED_CAR)
hw4.t$REVOKED <- as.factor(hw4.t$REVOKED)
hw4.t$URBANICITY <- as.factor(hw4.t$URBANICITY)
hw4.t$HOMEKIDS <- as.factor(hw4.t$HOMEKIDS)
hw4.t$KIDSDRIV <- as.factor(hw4.t$KIDSDRIV)
hw4.t$H_RENTER <- as.factor(hw4.t$H_RENTER)
hw4.t$NEW_CAR <- as.factor(hw4.t$NEW_CAR)

###eliminate variables we agreed to leave out.  Eliminated older_car and h.renter because the opposite v
###added back in BLUEBOOK & TRAVTIME bc their log transforms help the model
hw4.t <- hw4.t[,-c(1,3,7,10,12,20,27,29)]
#INDEX, TARGET_AMT, YOJ, HOME_VAL, SEX, RED_CAR, NEW_CAR, JOB_COLOR
```

```
# Use forward selection strategy to find model with lowest AIC using PREPPED data set (prepped as above
# iterate through predictors in descending order of correlation with target
# avoid highly collinear predictors with each iteration

m1 <- glm(data = hw4.t, TARGET_FLAG ~ MVR_PTS, family = binomial(link = "logit"))
summary(m1)

# added log(MVR_PTS + 1) due to high deviance of MVR_PTS
```

```
m2 <- glm(data = hw4.t, TARGET_FLAG ~ log(MVR_PTS + 1) + MVR_PTS, family = binomial(link = "logit"))
summary(m2)

# threw out log(MVR_PTS + 1) due to high p-value, threw out MVR_PTS due to high deviance
m3 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ, family = binomial(link = "logit"))
summary(m3)

# added log(CLM_FREQ + 1) due to high deviance of MVR_PTS
m4 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1), family = binomial(link = "logit"))
summary(m4)

# added OLDCLAIM
m5 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + OLDCLAIM, family = binomial(link =
summary(m5)

#removed OLDCLAIM due to high p-value, added INCOME
m6 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME, family = binomial(link = "
summary(m6)

# added log(INCOME + 1) due to high deviance of INCOME
m7 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1), family =
summary(m7)

# added AGE
m8 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE, fa
summary(m8)

# added log(AGE + 1) due to high deviance of AGE
m9 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE + l
summary(m9)

# added BLUEBOOK
m10 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE +
BLUEBOOK, family = binomial(link = "logit"))
summary(m10)

# added log(BLUBOOK) due to high deviance of BLUEBOOK
m11 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE +
summary(m11)

# added CAR_AGE, removed BLUEBOOK due to high p-value
m11 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE +
summary(m11)

# added CAR_AGE, removed BLUEBOOK due to high p-value
m12 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE +
summary(m12)

# added TIF
m13 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE +
summary(m13)

# added TRAVTIME
```

```r
m14 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE + l
summary(m14)

# added log(TRAVTIME) due to high deviation of TRAVTIME
m15 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE + l
summary(m15)

# remove TRAVTIME due to high p-value
m16 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE + l
summary(m16)

# remove log(INCOME + 1) due to high deviation of log(INCOME + 1)
m17 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m17)

m18 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m18)

m19 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m19)

m20 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m20)

m21 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m21)

m22 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m22)

# don't add HOMEKIDS due to high p-value
m23 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m23)

# remove JOB_COLOR due to high p-value
m24 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m24)

m25 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m25)

m26 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m26)

# don't add EDUCATION or JOB due to high p-value
m <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log(B
summary(m)
```

plot(hw4.t$TARGET_AMT)

```r
# Find outliers using ~ twice the average leverage
# Avg leverage is first dotted line ~.015
# Cutoff leverage is second dotted line ~.030
```

```r
# Note, the strategy in this model is forward selection and minimizing AIC
# while maintaining all predictor p-values within .05 significance levels.

# AIC minimization drove selection of outliers first, removing as many as plausible
# while staying within customary cutoff threshold

#Figure 8.13 on page 291
par(mfrow=c(1,1))
hvalues <- influence(m)$hat
stanresDeviance <- residuals(m)/sqrt(1-hvalues)

plot(hvalues,stanresDeviance,ylab="Standardized Deviance Residuals",
     xlab="Leverage Values",ylim=c(-3,3),xlim=c(-0.05,0.7))

# NOTE: the '7' indicated here is found by adding 1 to the number of predictor variables
# used in the final model
abline(v=2 * 13 / nrow(hw4.t),lty=2)
#.015

# Find outliers using ~ twice the average leverage
abline(v=2 * 26 / nrow(hw4.t),lty=2)
# .030

hw4.t\\.names <- as.character(seq(1:nrow(hw4.t)))



# need to click on potential outliers using the mouse and then click "finish" in the plot window
identify(hvalues, stanresDeviance, labels = hw4.names, cex=0.75)
```

Results say no rows to be removed

```r
hw4.re <- hw4.t

# now rebuild

#remove predictors with excessive deviation plots
m.re.1 <- glm(data = hw4.t, TARGET_FLAG ~ MVR_PTS, family = binomial(link = "logit"))
summary(m.re.1)

m.re.2 <- glm(data = hw4.t, TARGET_FLAG ~ MVR_PTS + log(MVR_PTS + 1), family = binomial(link = "logit"))
summary(m.re.2)

m.re.3 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ, family = binomial(link = "logit"))
summary(m.re.3)

m.re.4 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1), family = binomial(link = "logit
summary(m.re.4)

m.re.5 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME, family = binomial(link
summary(m.re.5)

m.re.6 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1), famil
```

```
summary(m.re.6)

m.re.7 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE
summary(m.re.7)

m.re.8 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE
summary(m.re.8)

m.re.9 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AGE
summary(m.re.9)

m.re.10 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AG
summary(m.re.10)

m.re.11 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AG
summary(m.re.11)

m.re.12 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AG
summary(m.re.12)

m.re.13 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AG
summary(m.re.13)

m.re.14 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AG
summary(m.re.14)

m.re.15 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + log(INCOME + 1) + AG
summary(m.re.15)

# add the binary predictors and see if we've introduced deviance that wasn't in the prior model
# we are ok
m.re.16 <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) +
summary(m.re.16)

# now do the same for categorical predictors
# we are ok
m.re <- glm(data = hw4.t, TARGET_FLAG ~ CLM_FREQ + log(CLM_FREQ + 1) + INCOME + AGE + log(AGE + 1) + log
summary(m.re)
```

marginal model plots

```
par(mar=c(1,1,1,1))
par(mfrow=c(1,1))
mmps(m.re,layout=c(4,3),key=TRUE)
dev.off()
```

Results say no rows to be removed

STOP

Now run metrics

```
# Coefficient Interpretation

# Logit model average marginal effects - use it to generate interpretable versions of coefficients
LogitScalar <- mean(dlogis(predict(m.re, type = "link")))
LogitScalar * coef(m.re)

# Logit model predicted probabilities - yields likelihood that each eval item is '+'
predprob.crash <- round(predict(m.re, type="response"), 2)
summary(predprob.crash)

# Percent correctly predicted values
# NOTE: Need to create variable 'Y' for this to work - set it to response variable
Y <- hw4.re[,1]

pred.crash <- round(fitted(m.re))

table(true = Y, pred = pred.crash)


# t.r <- data.frame(table(true = Y, pred = pred.crime))
# t.r

# now use functions built in HW 2 to get required statistics
accuracy(Y, pred.crash)
classif.err.rate(Y, pred.crash)
precision(Y, pred.crash)
sensitivity(Y, pred.crash)
specificity(Y, pred.crash)
F1.Score(Y, pred.crash)

# get AUC
rocCurve <- roc(response= Y, predictor= pred.crash)
auc(rocCurve)
```

Linear Model 1

```
library(alr3)
library(car)
library(pROC)
# library(MASS)


hw4 <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/621-HW4-XFORMED-DAT
```

Convert categoricals with strings to factors

```
hw4$PARENT1 <- factor(hw4$PARENT1)
hw4$MSTATUS <- factor(hw4$MSTATUS)
hw4$SEX <- factor(hw4$SEX)
```

```r
hw4$EDUCATION <- factor(hw4$EDUCATION)
hw4$JOB <- factor(hw4$JOB)
hw4$CAR_USE <- factor(hw4$CAR_USE)
hw4$CAR_TYPE <- factor(hw4$CAR_TYPE)
hw4$RED_CAR <- factor(hw4$RED_CAR)
hw4$REVOKED <- factor(hw4$REVOKED)
hw4$URBANICITY <- factor(hw4$URBANICITY)
hw4$KIDSDRIV <- factor(hw4$KIDSDRIV)
hw4$HOMEKIDS <- factor(hw4$HOMEKIDS)
hw4$H_RENTER <- factor(hw4$H_RENTER)
hw4$NEW_CAR <- factor(hw4$NEW_CAR)
hw4$JOB_COLOR <- factor(hw4$JOB_COLOR)
```

Transform INCOME, BLUEBOOK, CLM_FREQ, MVR_PTS, TRAVTIME using log(x)

```r
hw4$INCOME <- log(hw4$INCOME + 1)
hw4$BLUEBOOK <- log(hw4$BLUEBOOK)
hw4$CLM_FREQ <- log(hw4$CLM_FREQ + 1)
hw4$MVR_PTS <- log(hw4$MVR_PTS + 1)
hw4$TRAVTIME <- log(hw4$TRAVTIME)

# hw4$TARGET_AMT <- log(hw4$TARGET_AMT + 1)
```

```r
# Try removing all TARGET_FLAG == 0

hw4.t <- hw4[which(hw4$TARGET_FLAG == 1),]
hw4.safe <- hw4
hw4 <- hw4.t
```

Try using step() function

```r
m1.init <- lm(log(TARGET_AMT + 1) ~ . - INDEX - TARGET_FLAG - H_RENTER - CAR_AGE - JOB_COLOR, data=hw4)
# m1.init <- lm(TARGET_AMT ~ . - INDEX - H_RENTER - CAR_AGE - JOB_COLOR, data=hw4)
summary(m1.init)

# use STEP function to find best BIC model
m1 <- step(m1.init, trace=0)
summary(m1)

# now just recreate output of step function to ensure it matches:
m1.gen <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + SEX + BLUEBOOK +
    OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS, data = hw4)

summary(m1.gen)
vif(m1.gen)
```

Remove SEX

```r
m2 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK +
    OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS, data = hw4)
summary(m2)
```

Remove REVOKED

```
m3 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK +
    OLDCLAIM + CLM_FREQ + MVR_PTS, data = hw4)
summary(m3)
```

Remove OLDCLAIM

```
m4 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK + CLM_FREQ + MVR_PTS, data = hw4)
summary(m4)
```

Remove CLM_FREQ

```
m5 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK + MVR_PTS, data = hw4)
summary(m5)
```

STOP

```
par(mfrow=c(2,2))
plot(m5)
```

Diagnostics

AV plots show outliers so remove them and refit

```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(m5, id.n = 8)
```

REMOVE OUTLIERS AND REFIT

Per Cooks Distance, remove items 7691, 5389, 3599, 1592, 3577, 6606, 5190, 3595, 7072

```
############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
hw4_rem <- hw4[-c(7691, 5389, 3599, 1592, 3577, 6606, 5190, 3595, 7072),]

# renumber rows
rownames(hw4_rem) <- 1:nrow(hw4_rem)
```

Now refit first model from above: all variables

```
outr.init <- lm(log(TARGET_AMT + 1) ~ . - INDEX - TARGET_FLAG - H_RENTER - CAR_AGE - JOB_COLOR, data=hw4
summary(outr.init)

# use STEP function to find best BIC model
outr <- step(outr.init, trace=0)

summary(outr)


# now just recreate output of step function to ensure it matches:
outr.gen <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + SEX + BLUEBOOK +
```

```
    OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS, data = hw4_rem)

summary(outr.gen)
vif(outr.gen)
```

Remove SEX

```
outr.2 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK +
    OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS, data = hw4_rem)
summary(outr.2)
```

Remove REVOKED

```
outr.3 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK +
    OLDCLAIM + CLM_FREQ + MVR_PTS, data = hw4_rem)
summary(outr.3)
```

Remove OLDCLAIM

```
outr.4 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK + CLM_FREQ + MVR_PTS, data = hw4_rem)
summary(outr.4)
```

Remove CLM_FREQ

```
outr.5 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK + MVR_PTS, data = hw4_rem)
summary(outr.5)
```

STOP -

Diagnostics

```
par(mfrow=c(2,2))
plot(outr.5)
```

AV plots show outliers so remove them and refit

```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(outr.5, id.n = 8)
```

More outliers: remove and refit

Per Cooks Distance, remove items 1748, 1377, 944, 419, 947, 2037, 939, 1857, 1430

```
############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
hw4_rem <- hw4_rem[-c(1748, 1377, 944, 419, 947, 2037, 939, 1857, 1430),]

# renumber rows
rownames(hw4_rem) <- 1:nrow(hw4_rem)
```

Now refit first model from above: all variables

```
outr.init <- lm(log(TARGET_AMT + 1) ~ . - INDEX - TARGET_FLAG - H_RENTER - CAR_AGE - JOB_COLOR, data=hw4
summary(outr.init)

# use STEP function to find best BIC model
outr <- step(outr.init, trace=0)

summary(outr)

# now just recreate output of step function to ensure it matches:
outr.gen <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + SEX + BLUEBOOK +
    OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS, data = hw4_rem)

summary(outr.gen)
vif(outr.gen)
```

Remove REVOKED

```
outr.2 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + SEX + BLUEBOOK +
    OLDCLAIM + CLM_FREQ + MVR_PTS, data = hw4_rem)
summary(outr.2)
```

Remove OLDCLAIM

```
outr.3 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + SEX + BLUEBOOK + CLM_FREQ + MVR_PTS, data = hw4_r
summary(outr.3)
```

Remove CLM_FREQ

```
outr.4 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + SEX + BLUEBOOK + MVR_PTS, data = hw4_rem)
summary(outr.4)
```

Remove SEX

```
outr.5 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK + MVR_PTS, data = hw4_rem)
summary(outr.5)
```

STOP

Diagnostics

```
par(mfrow=c(2,2))
plot(outr.5)
```

AV plots show outliers so remove them and refit

```
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(outr.5, id.n = 8)
```

Outliers so remove and refit

REMOVE OUTLIERS AND REFIT

Per Cooks Distance, remove items 1727, 1942, 384, 2053, 639, 1546, 1824, 1353, 1709, 2065, 1612, 718

```
############ FIRST SET OF OUTLIERS ######################
# drop outlier records from data set
hw4_rem <- hw4_rem[-c(1727, 1942, 384, 2053, 639, 1546, 1824, 1353, 1709, 2065, 1612, 718),]

# renumber rows
rownames(hw4_rem) <- 1:nrow(hw4_rem)
```

Now refit first model from above: all variables

```
outr.init <- lm(log(TARGET_AMT + 1) ~ . - INDEX - TARGET_FLAG - H_RENTER - CAR_AGE - JOB_COLOR, data=hw4
summary(outr.init)

# use STEP function to find best BIC model
outr <- step(outr.init, trace=0)

summary(outr)


# now just recreate output of step function to ensure it matches:
outr.gen <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + EDUCATION + BLUEBOOK +
    RED_CAR + MVR_PTS, data = hw4_rem)

summary(outr.gen)
vif(outr.gen)
```

Remove EDUCATION

```
outr.2 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK + RED_CAR + MVR_PTS, data = hw4_rem)
summary(outr.2)
```

Remove RED_CAR

```
outr.3 <- lm(formula = log(TARGET_AMT + 1) ~ MSTATUS + BLUEBOOK + MVR_PTS, data = hw4_rem)
summary(outr.3)
```

Remove MSTATUS

```
outr.4 <- lm(formula = log(TARGET_AMT + 1) ~ BLUEBOOK + MVR_PTS, data = hw4_rem)
summary(outr.4)
```

Get Mean Squared Error

```
anova(outr.4)
```

STOP

SUMMARY MODEL DIAGNOSTIC PLOTS

```
par(mfrow=c(2,2))
plot(outr.4)
```

AV Plots

```r
# CREATE ADDED VARIABLE PLOTS TO ASSESS predictor vs response
avPlots(outr.4, id.n = 8)
```

PLOT STANDARDIZED RESIDUALS AGAINST EACH PREDICTOR

```r
StanRes1 <- rstandard(outr.4)
par(mfrow=c(2,2))

plot(hw4_rem$BLUEBOOK, StanRes1, ylab="Standardized Residuals")
plot(hw4_rem$MVR_PTS, StanRes1, ylab="Standardized Residuals")
```

PLOT Y AGAINST FITTED VALUES

```r
fit1 <- outr.4$fitted.values
summary(outr.4$fitted.values)

fit2 = round(exp(fit1) - 1)
summary(fit2)

# fit3 <- fit2[which(fit2 > 0)]
# summary(fit3)

Payout <- hw4_rem$TARGET_AMT
summary(Payout)

par(mfrow = c(1,1))
plot(fit2, Payout, xlab="Fitted Values")
abline(lsfit(fit2, Payout),lty=2)
```

Linear Model 2

```r
#Step 0:  Subset data for only those with payouts (TARGET_FLAG=1) and get rid of extra variables:
hw4t.p <- hw4t[which(hw4t$TARGET_FLAG == 1),]

#Step 2:
#Step 1: take out new fields
pay <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR)
summary(pay)

#Step 2:  take out kids drive
pay1 <- lm(data=hw4t.p, TARGET_AMT~-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR - KIDSDRIV)
summary(pay1)

#Step 3: take out job
pay2 <- lm(data=hw4t.p, TARGET_AMT~. -INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- KIDSDRIV - 
summary(pay2)

#Step 4: take out car type
```

```
pay3 <- lm(data=hw4t.p, TARGET_AMT~. -INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- KIDSDRIV - J
summary(pay3)

#Step 5: take out urbanicity
pay4 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDRI
summary(pay4)

#Step 6: take out travel time
pay5 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR-JOB - KIDSDRIV
summary(pay5)

#Step 7: take out single parent
pay6 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR-JOB - KIDSDRIV
summary(pay6)

#Step 8: take out TIF
pay7 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDRI
summary(pay7)

#Step 9:  take out red car
pay8 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDRI
summary(pay8)

#Step 10:  take out education
pay9 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDRI
summary(pay9)

#Step 11:  take out car use
pay10 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDR
summary(pay10)

#Step 12:  take out YOJ
pay11 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB  - KIDSD
summary(pay11)

#Step 13:  take out CLM_FREQ
pay12 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDR
summary(pay12)

#Step 14:  take out old claim
pay13 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR-JOB - KIDSDRI
summary(pay13)

#Step 15:  take out income
pay14 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDR
summary(pay14)

#Step 16:  take out home value
pay15 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDR
summary(pay15)

#Step 17:  take out home kids
pay16 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDR
```

```
summary(pay16)

#Step 18:  take out age
pay17 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB  - KIDSD
summary(pay17)

#Step 19:  take out car age
pay18 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR- JOB - KIDSDR
summary(pay18)

#Step 20:  take out marital status
pay19 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR-JOB - KIDSDRI
summary(pay19)

#Step 21:  take out revoked
pay20 <- lm(data=hw4t.p, TARGET_AMT~.-INDEX - TARGET_FLAG - NEW_CAR - H_RENTER - JOB_COLOR-JOB - KIDSDRI
summary(pay20)



#Look at mmps
mmps(pay20,layout=c(2,2),key=TRUE)

#Step22:  look at transformations
#POssible transformations
summary(powerTransform(BLUEBOOK~TARGET_AMT, hw4t.p, family="bcPower"))
boxcox(hw4t.p$BLUEBOOK~hw4t.p$TARGET_AMT)
#use sqroot for BLUEBOOK

summary(powerTransform(MVR_PTS+1~TARGET_AMT, hw4t.p, family="bcPower"))
boxcox(hw4t.p$MVR_PTS +1~hw4t.p$TARGET_AMT)
#use log for MVR_PTS

#Step 23:  Make new model with transformations
pay21 <- lm(data=hw4t.p, TARGET_AMT~sqrt(BLUEBOOK) + log(MVR_PTS +1) + SEX )
summary(pay21)
vif(pay21)

#Step 24:  remove sex
pay22 <- lm(data=hw4t.p, TARGET_AMT~sqrt(BLUEBOOK) + log(MVR_PTS +1))
summary(pay22)
vif(pay22)

plot(pay22$residuals~hw4t.p$TARGET_AMT)
avPlots(pay22, id.n = 8)


#this data has so much variance it is hard to say what should be removed or not - so leaving it all in.
mmps(pay22, layout=c(2,2), key=T)
```

Part 4. Select Models

R code for the required 2-stage prediction process

First stage:

1) Load training data
2) Perform any necessary transforms on data
3) build selected binary regression model
4) Load eval data
5) Perform any necessary transforms on eval data
6) use **predict** function to get required probabilities
7) Save both the probabilities and their rounded 0/1 values to the eval data set

```
# load training set so that binary model can be built

hw4 <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/621-HW4-XFORMED-DAT

# convert categoricals to factors
hw4$PARENT1 <- factor(hw4$PARENT1)
hw4$MSTATUS <- factor(hw4$MSTATUS)
hw4$SEX <- factor(hw4$SEX)
hw4$EDUCATION <- factor(hw4$EDUCATION)
hw4$JOB <- factor(hw4$JOB)
hw4$CAR_USE <- factor(hw4$CAR_USE)
hw4$CAR_TYPE <- factor(hw4$CAR_TYPE)
hw4$RED_CAR <- factor(hw4$RED_CAR)
hw4$REVOKED <- factor(hw4$REVOKED)
hw4$URBANICITY <- factor(hw4$URBANICITY)
hw4$KIDSDRIV <- factor(hw4$KIDSDRIV)
hw4$HOMEKIDS <- factor(hw4$HOMEKIDS)
hw4$H_RENTER <- factor(hw4$H_RENTER)
hw4$NEW_CAR <- factor(hw4$NEW_CAR)
hw4$JOB_COLOR <- factor(hw4$JOB_COLOR)

# save a copy of TARGET_AMT and TARGET_FLAG for stats at end
Target.amt <- hw4$TARGET_AMT
Target.f <- hw4$TARGET_FLAG

m1 <- glm(formula = TARGET_FLAG ~ KIDSDRIV + HOMEKIDS + log(INCOME + 1) + MSTATUS +
    EDUCATION + log(TRAVTIME) + CAR_USE + log(BLUEBOOK) + TIF + CAR_TYPE +
    OLDCLAIM + CLM_FREQ + log(CLM_FREQ + 1) + REVOKED + log(MVR_PTS + 1) + URBANICITY + H_RENTER,
    family = binomial(link = "logit"), data = hw4)

# ---------------------------------------

# now that model is built, load eval data set

# load EVAL data set
hw4 <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/HW4-XFORMED-EVAL-DAT

# convert categoricals to factors
hw4$PARENT1 <- factor(hw4$PARENT1)
hw4$MSTATUS <- factor(hw4$MSTATUS)
```

```
hw4$SEX <- factor(hw4$SEX)
hw4$EDUCATION <- factor(hw4$EDUCATION)
hw4$JOB <- factor(hw4$JOB)
hw4$CAR_USE <- factor(hw4$CAR_USE)
hw4$CAR_TYPE <- factor(hw4$CAR_TYPE)
hw4$RED_CAR <- factor(hw4$RED_CAR)
hw4$REVOKED <- factor(hw4$REVOKED)
hw4$URBANICITY <- factor(hw4$URBANICITY)
hw4$KIDSDRIV <- factor(hw4$KIDSDRIV)
hw4$HOMEKIDS <- factor(hw4$HOMEKIDS)
hw4$H_RENTER <- factor(hw4$H_RENTER)
hw4$NEW_CAR <- factor(hw4$NEW_CAR)
hw4$JOB_COLOR <- factor(hw4$JOB_COLOR)

# make a copy of the original data
eval.out <- hw4

# now predict TARGET_FLAG using model
pred.CR <- predict(m1, newdata=eval.out, type="response")

# Save predicted probability and rounded value to eval data set
eval.out$TARGET_FLAG_PROB <- round(pred.CR, 3)
eval.out$TARGET_FLAG <- round(pred.CR)
```

Second stage:

1) Load training data set again
2) Perform any necessary transforms on data
3) build selected linear regression model
4) Extract only TARGET_FLAG = 1 rows from eval data
5) Perform any necessary transforms on copy of eval data
6) use **predict** function to get required payout amounts
7) Save predicted payout amount to eval data set

```
# load training set so that linear model can be built

hw4 <- read.csv("https://raw.githubusercontent.com/jtopor/CUNY-MSDA-621/master/HW-4/621-HW4-XFORMED-DAT

# convert categoricals to factors
hw4$PARENT1 <- factor(hw4$PARENT1)
hw4$MSTATUS <- factor(hw4$MSTATUS)
hw4$SEX <- factor(hw4$SEX)
hw4$EDUCATION <- factor(hw4$EDUCATION)
hw4$JOB <- factor(hw4$JOB)
hw4$CAR_USE <- factor(hw4$CAR_USE)
hw4$CAR_TYPE <- factor(hw4$CAR_TYPE)
hw4$RED_CAR <- factor(hw4$RED_CAR)
hw4$REVOKED <- factor(hw4$REVOKED)
hw4$URBANICITY <- factor(hw4$URBANICITY)
hw4$KIDSDRIV <- factor(hw4$KIDSDRIV)
hw4$HOMEKIDS <- factor(hw4$HOMEKIDS)
hw4$H_RENTER <- factor(hw4$H_RENTER)
hw4$NEW_CAR <- factor(hw4$NEW_CAR)
```

```r
hw4$JOB_COLOR <- factor(hw4$JOB_COLOR)

# read in data set was done above
hw4.L1 <- hw4[which(hw4$TARGET_FLAG == 1),]

# tranform variables as needed
# hw4.L1$INCOME <- log(hw4.L1$INCOME + 1)
# hw4.L1$BLUEBOOK <- log(hw4.L1$BLUEBOOK)
# hw4.L1$CLM_FREQ <- log(hw4.L1$CLM_FREQ + 1)
# hw4.L1$MVR_PTS <- log(hw4.L1$MVR_PTS + 1)
# hw4.L1$TRAVTIME <- log(hw4.L1$TRAVTIME)

# remove all outliers

# set 1
hw4_rem <- hw4.L1[-c(7691, 5389, 3599, 1592, 3577, 6606, 5190, 3595, 7072),]

# renumber rows
rownames(hw4_rem) <- 1:nrow(hw4_rem)

# --------------------------
# set 2
hw4_rem <- hw4_rem[-c(1748, 1377, 944, 419, 947, 2037, 939, 1857, 1430),]

# renumber rows
rownames(hw4_rem) <- 1:nrow(hw4_rem)

# -------------------------
# set 3
hw4_rem <- hw4_rem[-c(1727, 1942, 384, 2053, 639, 1546, 1824, 1353, 1709, 2065, 1612, 718),]

# renumber rows
rownames(hw4_rem) <- 1:nrow(hw4_rem)

# now fit the model
lm1 <- lm(formula = log(TARGET_AMT + 1) ~ log(BLUEBOOK) + log(MVR_PTS + 1), data = hw4_rem)


# ---------------------------------

# extract only transform variables as needed

eval.lm <- eval.out[which(eval.out$TARGET_FLAG == 1),]

# tranform variables as needed
# eval.lm$INCOME <- log(eval.lm$INCOME + 1)
# eval.lm$BLUEBOOK <- log(eval.lm$BLUEBOOK)
# eval.lm$CLM_FREQ <- log(eval.lm$CLM_FREQ + 1)
# eval.lm$MVR_PTS <- log(eval.lm$MVR_PTS + 1)
# eval.lm$TRAVTIME <- log(eval.lm$TRAVTIME)

# now predict TARGET_AMT using model
pred.CR <- predict(lm1, newdata=eval.lm, type="response")
```

```r
# back transform predicted values if response was log transformed
preds = round(exp(pred.CR) - 1)

# now add predicted TARGET_AMT to TARGET_FLAG = 1 restricted eval data set
eval.lm$TARGET_AMT <- round(preds, 2)

# ------------------------------------------------------------------
# now combine prediction results with eval rows that didn't require predictions

eval.out <- rbind(eval.lm, eval.out[which(eval.out$TARGET_FLAG == 0),])

# re-sort eval data by iNDEX
library(plyr)

eval.out <- arrange(eval.out, INDEX)
```

```r
# write full model EVAL data to a CSV file
write.csv(eval.out, file = "C:/SQLData/621/HW4-PRED-EVAL-ALL-DATA.csv", row.names = FALSE)

# write only pertinent columns to a CSV file
# now write just INDEX and TARGET_WINS to a separate file
eval.s <- eval.out[,c(1,30,2,3)]

write.csv(eval.s, file = "C:/SQLData/621/HW4-PRED-EVAL-COLS-ONLY.csv", row.names = FALSE)

library(psych)
describe(eval.out$TARGET_FLAG)
describe(Target.f)

describe(eval.out$TARGET_AMT)
describe(Target.amt)
```