HW4

Dan Fanelli

July 5, 2016

Outline:

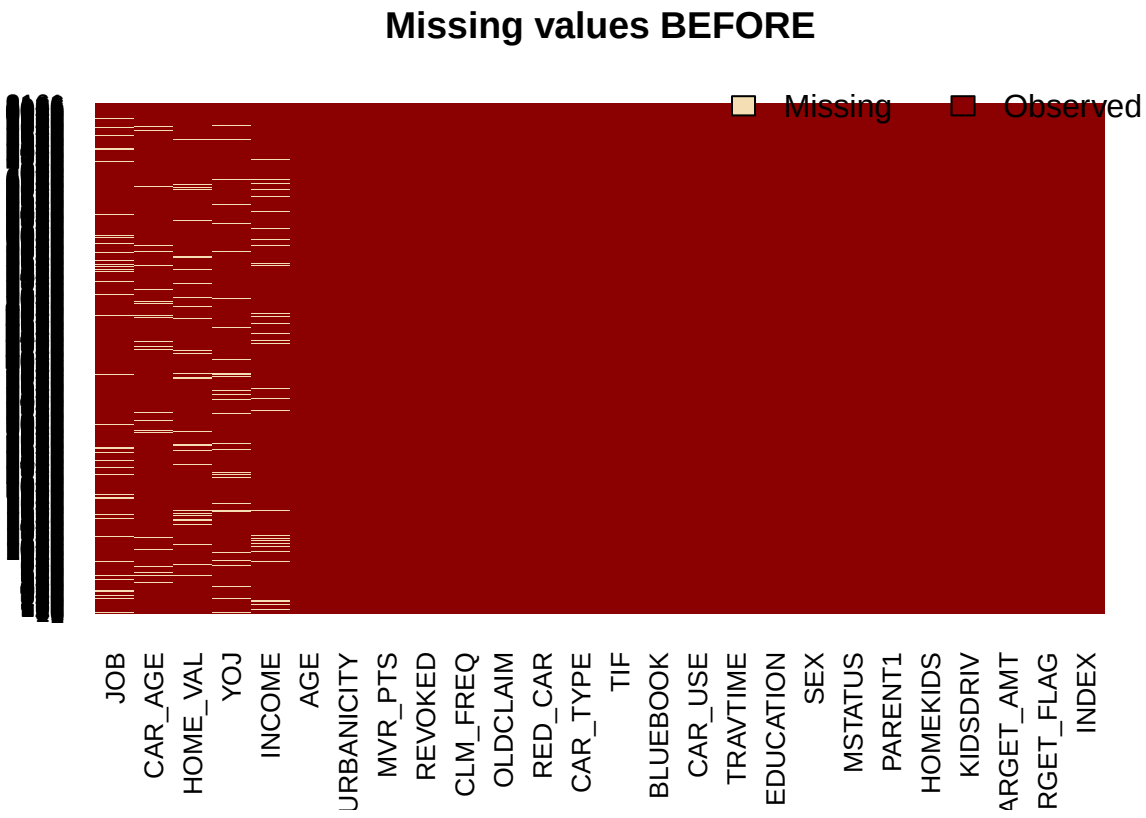
- This data set has ***HIGH CORRELATIONS*** between dependent variables.
- Regular "Step" selection does is not having much of an impact (AUC stays the same before and after attribute selection).
- So calculate and rank all combinations of categorical variables by brute force. (after removing repeat highly correlated data)

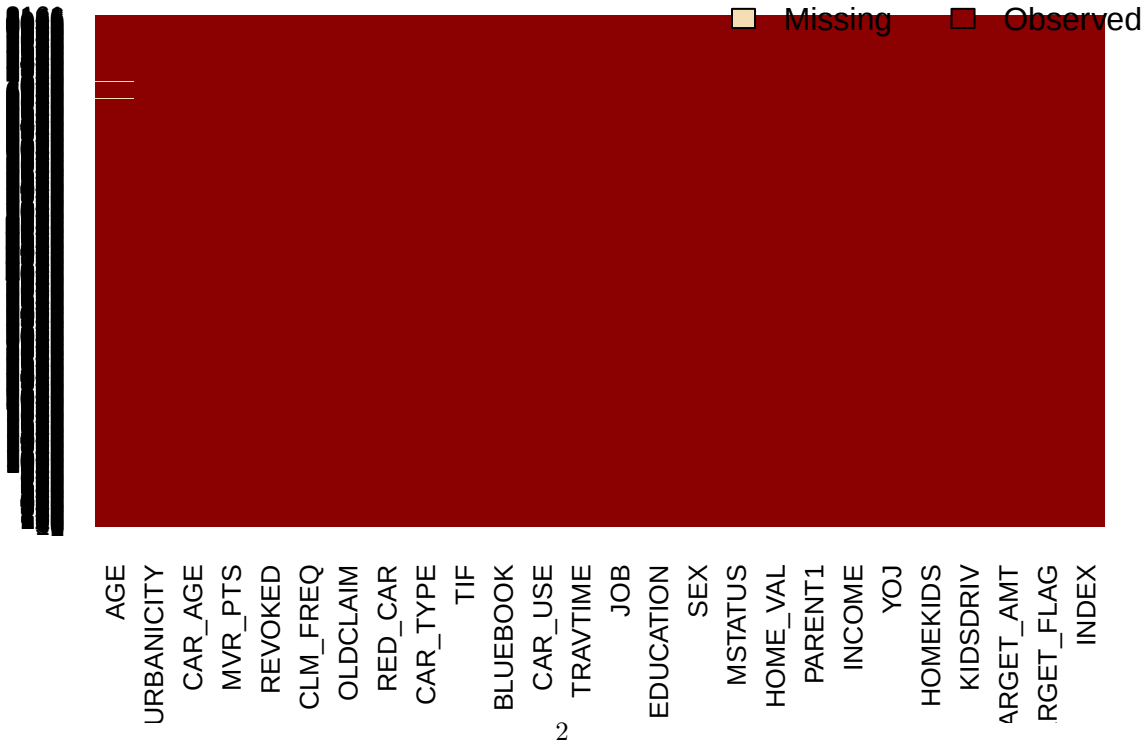
Load Data:

Converting the money character fields to numeric.

INDEX	TARGET_FLAG	TARGET_AMT	KIDSDRIV	AGE	HOMEKIDS	YOJ	INCOME	PARENT1	HO
1	0	0	0	60	0	11	67349	No	
2	0	0	0	43	0	11	91449	No	
4	0	0	0	35	1	10	16039	No	
5	0	0	0	51	0	14	NA	No	
6	0	0	0	50	0	NA	114986	No	

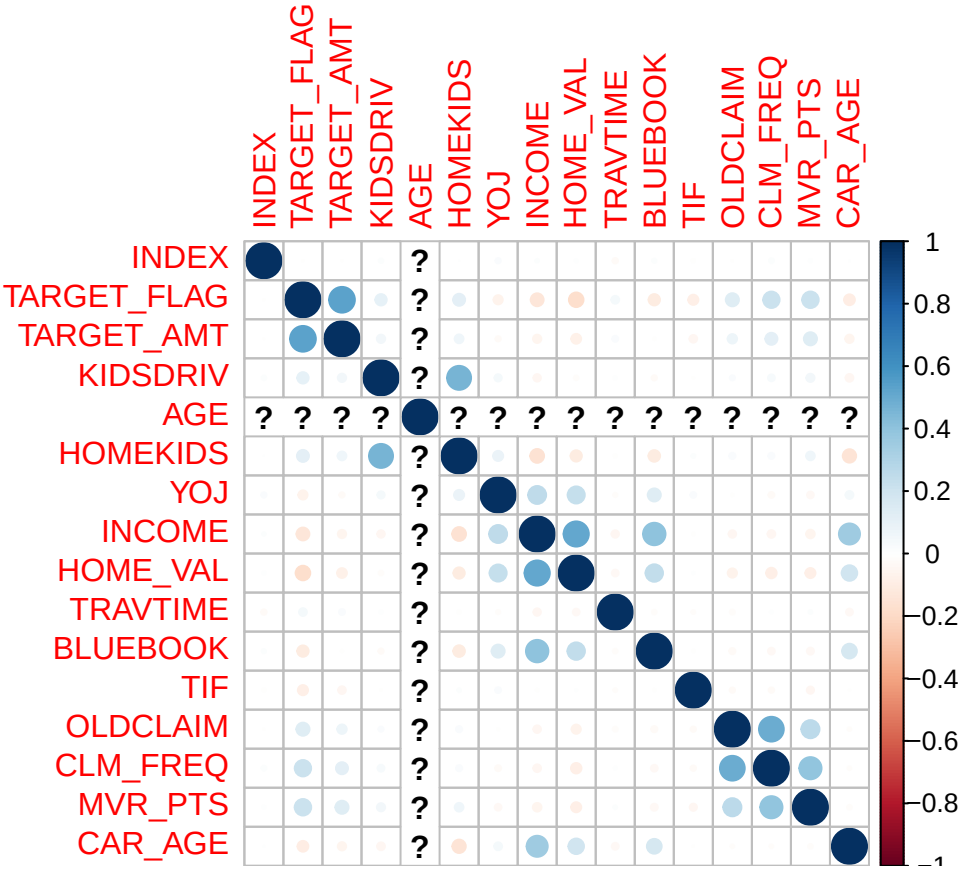
Plot and fill in the Nulls.





Data Exploration

The cor plot shows ***VERY LITTLE CORRELATION*** between TARGET_FLAG/AMT and other fields (a "Weak" model?):



Work:

Build multiple linear regression and binary logistic regression models to: * predict the probability that a person will crash their car and also * the amount of money it will cost if the person does crash their car

TARGET_FLAG: Was Car in a crash? 1=YES 0=NO

```
cutoff <- nrow(ins)*.75
ins.train <- ins[1:cutoff,]
ins.test <- ins[(cutoff+1):nrow(ins),]
```

Calculate the prediction ACCURACY based on JUST the Categorical Variables:

col_names	accuracy
CAR_TYPE	0.7331699
CAR_USE	0.7331699
EDUCATION	0.7331699
JOB	0.7331699

col_names	accuracy
MSTATUS	0.7331699
PARENT1	0.7331699
RED_CAR	0.7331699
REVOKED	0.7331699
SEX	0.7331699
URBANICITY	0.7331699

Those categorical variables have the exact same calculated accuracties.

Could this be coincidence? Seems those columns are duplicates, run Chi-Square correlation tests to confirm:

all_chi_sq_labels	all_chi_sq_results
car_type_to_car_use	0.000
car_type_to_education	0.000
car_type_to_job	0.000
car_type_to_mstatus	0.315
car_type_to_parent1	0.000
car_type_to_red_car	0.000
car_type_to_revoked	0.035
car_type_to_sex	0.000
car_type_to_urbancity	0.000

Chi-Squared says some columns are exact duplicates: Remove Them! (We'll keep CARTYPE):

```
drops <- c("CAR_USE","EDUCATION","JOB","PARENT1","RED_CAR","SEX","URBANICITY")
ins.train <- ins.train[ , !(names(ins.train) %in% drops)]
ins.test <- ins.test[ , !(names(ins.test) %in% drops)]
```

Also, ***car_type_to_mstatus*** and ***car_type_to_revoked*** were close, lets compare mstatus to revoked:

```
mstatus_to_revoked <- chisq.test(table(ins.train$MSTATUS, ins.train$REVOKED))
round(mstatus_to_revoked$p.value, digits = 3)
```

```
## [1] 0.001
```

So those are ALSO ***DUPLICATES***, remove REVOKED

```
drops <- c("REVOKED")
ins.train <- ins.train[ , !(names(ins.train) %in% drops)]
ins.test <- ins.test[ , !(names(ins.test) %in% drops)]
```

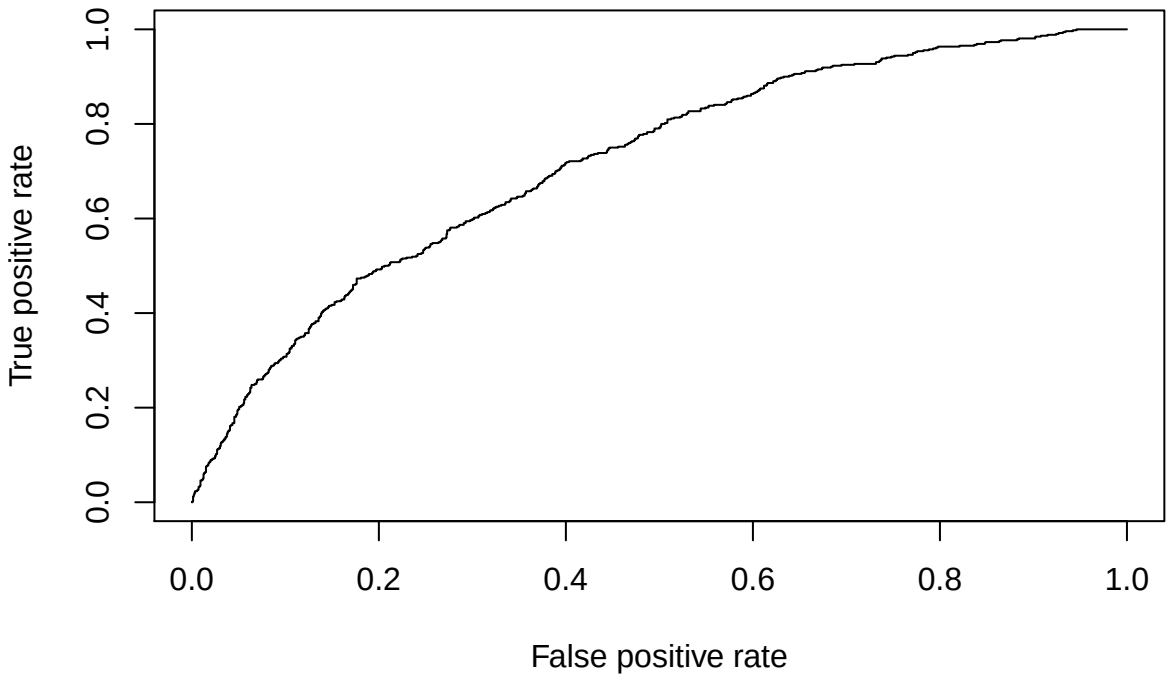
Look at the data after those drops:

```
kable(head(ins.train))
```

INDEX	TARGET_FLAG	TARGET_AMT	KIDSDRIV	AGE	HOMEKIDS	YOJ	INCOME	HOME_VAL
1	0	0	0	60	0	11	67349	0
2	0	0	0	43	0	11	91449	257252
4	0	0	0	35	1	10	16039	124191
5	0	0	0	51	0	14	79213	306251
6	0	0	0	50	0	12	114986	243925
7	1	2946	0	34	1	12	125301	0

Now, try some basic Step Selection:

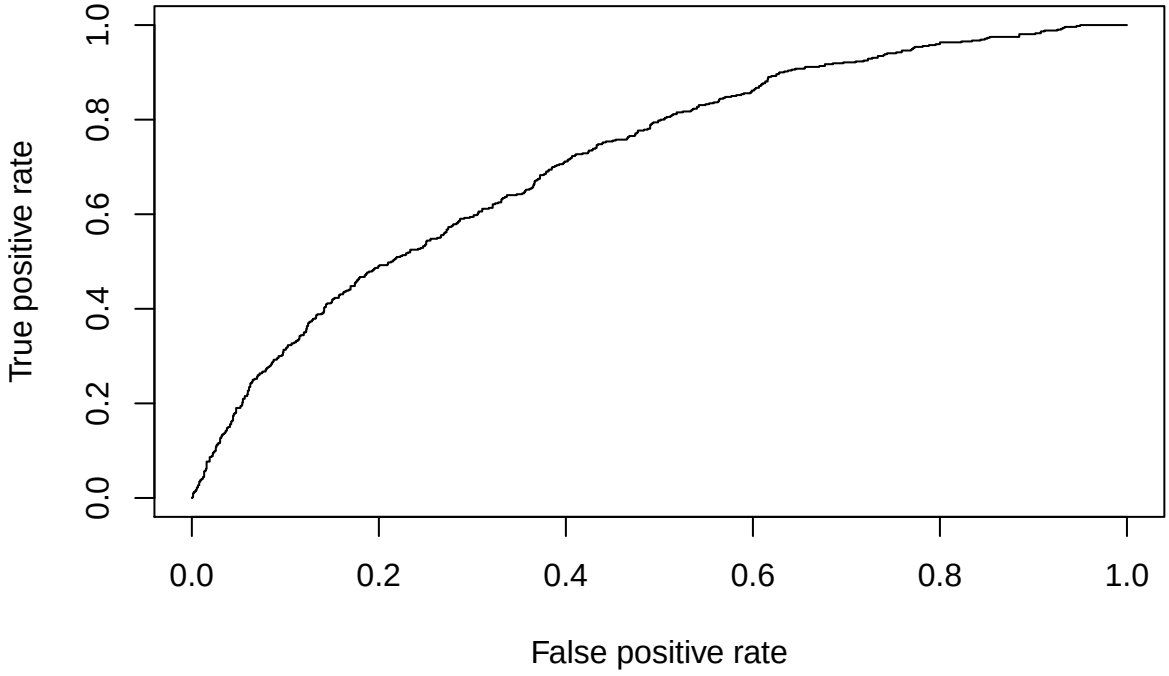
```
## ACCURACY (BEFORE STEPS):

## [1] 0.7544118

## ACCURACY (AFTER STEPS):

## [1] 0.7578431

## AUC (BEFORE STEPS):
```



```
## NULL
```

```
## [1] 0.7176164
```

```
## AUC (AFTER STEPS):
```




```
## NULL
```

```
## [1] 0.7172988
```

The Point: Step Selection (STILL) not giving much of an improvement.

So what can we do when no columns are showing real strength above the others?

Grind out all combinations with some Tools: MuMIn DREDGE

```
library(MuMIn)
core.glm <- glm(TARGET_FLAG ~  . -TARGET_AMT, data = ins.train)
```

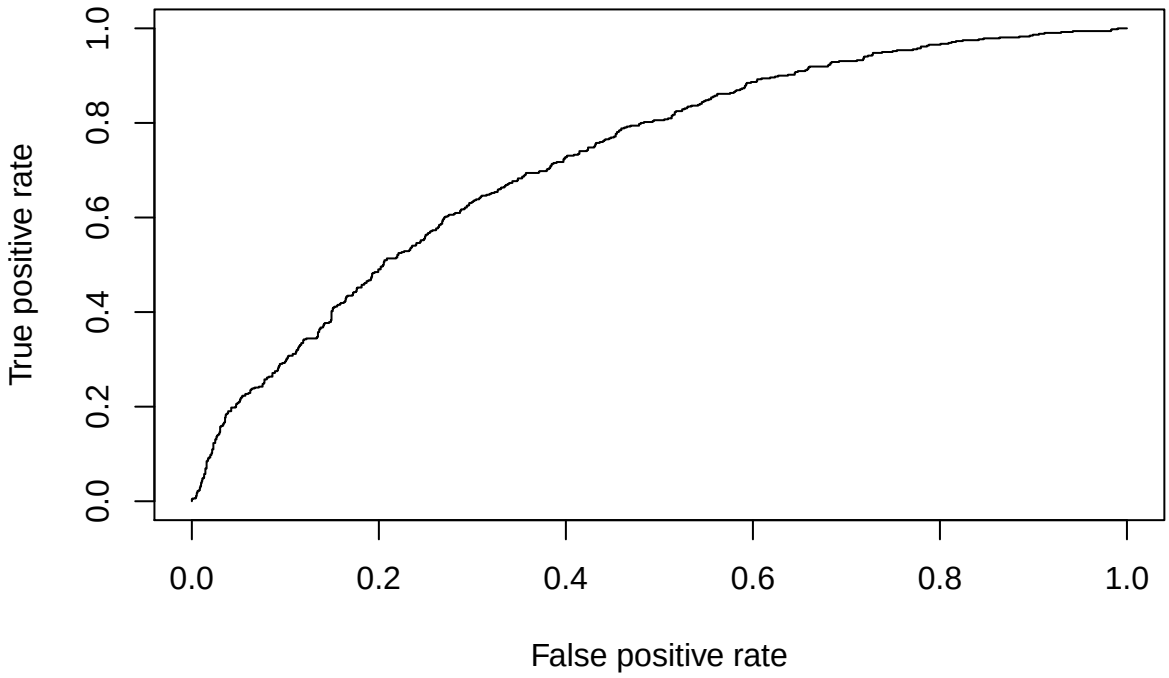
```
options(na.action = "na.fail")
#dredge.result <- dredge(core.glm, rank="AIC", extra=c("R^2", "adjR^2"))
dredge.result <- dredge(core.glm, extra=c("R^2", "adjR^2"))
kable(head(dredge.result, n=20))
```

ed term is "(Intercept)"

(Intercept)	AGE	BLUEBOOK	CAR_AGE	CAR_TYPE	CLM_FREQ	HOME_VAL	HOMEKIDS	INCOME	INDEX	KIDSDRIV	MSTATUS	MVR_PTS	OLDCLAIM	TIF	TRAVTIME	YOJ	R^2	adjR^2	df	logLik	AICc	delta	weight
0.1654218	NA	-3.8e-06	-0.0031407	+	0.0471399	-2e-07	0.0230821	-4e-07	NA	0.0525913	+	0.0295669	NA	-0.0079085	0.0012764	NA	0.1407629	0.2008849	18	-3227.216	6490.544	0.0000000	0.1954422
0.1644285	NA	-3.8e-06	-0.0031395	+	0.0442270	-2e-07	0.0229330	-4e-07	NA	0.0528252	+	0.0293054	8e-07	-0.0078922	0.0012901	NA	0.1409646	0.2011728	19	-3226.497	6491.119	0.5749944	0.1466087
0.1774636	NA	-3.8e-06	-0.0031665	+	0.0471655	-2e-07	0.0237294	-4e-07	NA	0.0524009	+	0.0295398	NA	-0.0078817	0.0012753	-0.0013654	0.1409067	0.2010901	19	-3226.704	6491.532	0.9878500	0.1192640
0.1565010	NA	-3.8e-06	-0.0031228	+	0.0471269	-2e-07	0.0231646	-4e-07	2.2e-06	0.0523743	+	0.0295434	NA	-0.0078966	0.0012820	NA	0.1408909	0.2010676	19	-3226.760	6491.644	1.1004981	0.1127322
0.1766875	NA	-3.8e-06	-0.0031658	+	0.0442131	-2e-07	0.0235907	-4e-07	NA	0.0526343	+	0.0292742	8e-07	-0.0078648	0.0012891	-0.0013916	0.1411140	0.2013860	20	-3225.965	6492.068	1.5238568	0.0912256
0.1552147	NA	-3.8e-06	-0.0031211	+	0.0441564	-2e-07	0.0230151	-4e-07	2.3e-06	0.0526061	+	0.0292760	8e-07	-0.0078797	0.0012961	NA	0.1411006	0.2013668	20	-3226.013	6492.163	1.6197132	0.0869565
0.1685750	NA	-3.8e-06	-0.0031488	+	0.0471527	-2e-07	0.0238236	-4e-07	2.3e-06	0.0521770	+	0.0295155	NA	-0.0078692	0.0012809	-0.0013870	0.1410393	0.2012793	20	-3226.231	6492.600	2.0564177	0.0698993
0.1675090	NA	-3.8e-06	-0.0031475	+	0.0441409	-2e-07	0.0236850	-4e-07	2.4e-06	0.0524082	+	0.0292437	9e-07	-0.0078516	0.0012952	-0.0014144	0.1412548	0.2015870	21	-3225.463	6493.078	2.5341826	0.0550462
0.1842441	NA	-4.2e-06	-0.0037819	+	0.0471916	-2e-07	0.0244862	NA	NA	0.0518297	+	0.0296445	NA	-0.0078078	0.0012849	-0.0019021	0.1401207	0.1999684	18	-3229.502	6495.116	4.5722445	0.0198686
0.1676800	NA	-4.3e-06	-0.0038083	+	0.0471576	-3e-07	0.0236362	NA	NA	0.0520438	+	0.0296941	NA	-0.0078384	0.0012876	NA	0.1398334	0.1995584	17	-3230.524	6495.148	4.6047179	0.0195486
0.1834021	NA	-4.2e-06	-0.0037769	+	0.0441647	-2e-07	0.0243388	NA	NA	0.0520730	+	0.0293714	9e-07	-0.0077909	0.0012991	-0.0019253	0.1403386	0.2002794	19	-3228.726	6495.577	5.0334421	0.0157768
0.1666533	NA	-4.3e-06	-0.0038037	+	0.0441795	-3e-07	0.0234811	NA	NA	0.0522857	+	0.0294261	9e-07	-0.0078222	0.0013015	NA	0.1400444	0.1998595	18	-3229.774	6495.659	5.1154511	0.0151430
0.1753697	NA	-4.2e-06	-0.0037643	+	0.0471788	-2e-07	0.0245804	NA	2.3e-06	0.0516060	+	0.0296202	NA	-0.0077953	0.0012906	-0.0019238	0.1402529	0.2001571	19	-3229.031	6496.187	5.6436745	0.0116281
0.1588439	NA	-4.3e-06	-0.0037914	+	0.0471448	-3e-07	0.0237186	NA	2.2e-06	0.0518281	+	0.0296709	NA	-0.0078266	0.0012931	NA	0.1399591	0.1997378	18	-3230.077	6496.266	5.7221741	0.0111806
0.1742300	NA	-4.2e-06	-0.0037587	+	0.0440926	-2e-07	0.0244331	NA	2.4e-06	0.0518469	+	0.0293410	9e-07	-0.0077777	0.0013052	-0.0019481	0.1404793	0.2004801	20	-3228.226	6496.589	6.0452536	0.0095128
0.1575178	NA	-4.3e-06	-0.0037863	+	0.0441093	-3e-07	0.0235631	NA	2.3e-06	0.0520678	+	0.0293970	9e-07	-0.0078097	0.0013075	NA	0.1401781	0.2000503	19	-3229.298	6496.720	6.1763559	0.0089092
0.1470002	NA	-3.9e-06	NA	+	0.0471102	-2e-07	0.0246553	-5e-07	NA	0.0525257	+	0.0295755	NA	-0.0079810	0.0012953	NA	0.1393673	0.1988933	17	-3232.182	6498.464	7.9198383	0.0037260
0.1460114	NA	-3.9e-06	NA	+	0.0441896	-2e-07	0.0245053	-5e-07	NA	0.0527602	+	0.0293133	8e-07	-0.0079647	0.0013091	NA	0.1395702	0.1991828	18	-3231.460	6499.033	8.4889284	0.0028033

So evaluate the top 3 models in the chart:

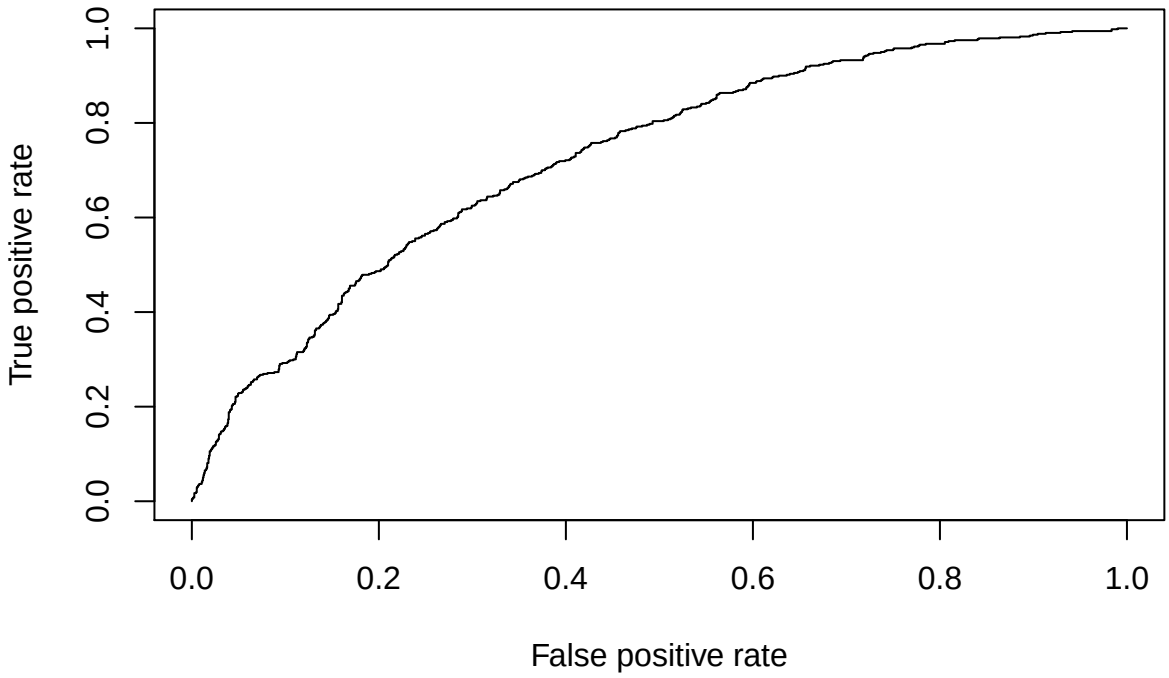
```
model_num_1 <- glm(TARGET_FLAG ~ BLUEBOOK + CAR_AGE + CAR_TYPE + CLM_FREQ + HOME_VAL + HOMEKIDS + INCOM
do_auc(model_num_1, ins.test)
```




```
## NULL
```

```
## [1] 0.7264233
```

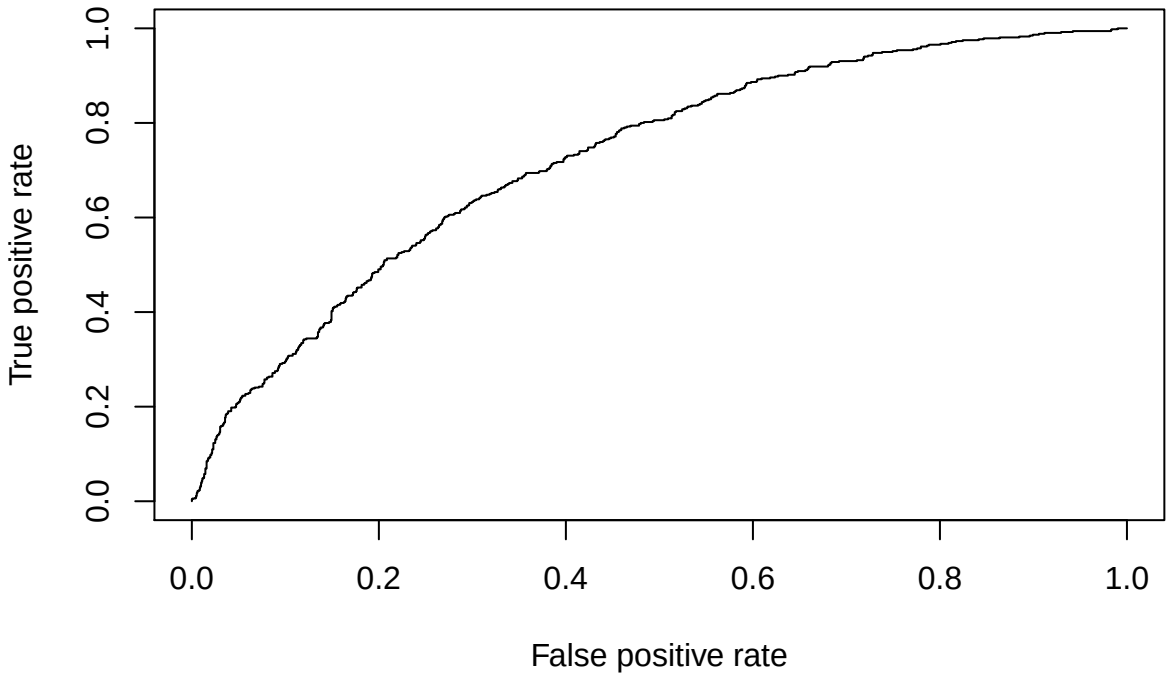
```
# model 2 is same as 1, only with OLDCLAIM added
model_num_2 <- glm(TARGET_FLAG ~ BLUEBOOK + CAR_AGE + CAR_TYPE + CLM_FREQ + HOME_VAL + HOMEKIDS + INCOM
do_auc(model_num_2, ins.test)
```




```
## NULL
```

```
## [1] 0.7266587
```

```
# model 3 is same as 1, only with YOJ added
model_num_3 <- glm(TARGET_FLAG ~ BLUEBOOK + CAR_AGE + CAR_TYPE + CLM_FREQ + HOME_VAL + HOMEKIDS + INCOM
do_auc(model_num_3, ins.test)
```




```
## NULL
```

```
## [1] 0.7264233
```

Model Comparisons:

Table 5: Confusion Matrix

	Model1	Model2	Model3	ModelFull	ModelSteps
Accuracy	0.7612745	0.7583333	0.7612745	0.7593137	0.7514706
Kappa	0.1705794	0.1575473	0.1705794	0.1595570	0.0713952
AccuracyLower	0.7421624	0.7391480	0.7421624	0.7401527	0.7321190
AccuracyUpper	0.7796328	0.7767734	0.7796328	0.7777267	0.7700969
AccuracyNull	0.7450980	0.7450980	0.7450980	0.7450980	0.7450980
AccuracyPValue	0.0486021	0.0885420	0.0486021	0.0731648	0.2635146
McnemarPValue	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Sensitivity	0.1634615	0.1538462	0.1634615	0.1538462	0.0634615
Specificity	0.9657895	0.9651316	0.9657895	0.9664474	0.9868421
Pos Pred Value	0.6204380	0.6015038	0.6204380	0.6106870	0.6226415
Neg Pred Value	0.7714136	0.7692711	0.7714136	0.7695128	0.7549069
Prevalence	0.2549020	0.2549020	0.2549020	0.2549020	0.2549020
Detection Rate	0.0416667	0.0392157	0.0416667	0.0392157	0.0161765
Detection Prevalence	0.0671569	0.0651961	0.0671569	0.0642157	0.0259804
Balanced Accuracy	0.5646255	0.5594889	0.5646255	0.5601468	0.5251518

Conclustion:

The models generated by the tool do not have a lower AU, but some of their other specs seem better, so seems the tool has picked them over the base Steps model.

Basically, I think this data set is a weak model since the AUC can not really be improved upon much by attribute selection.

NOT DONE AT ALL: TARGET_AMT calulation...