Project progress reports:
By Friday, Nov. 18, 5:00 PM, each group should submit a progress report in the dropbox folder of the person of contact.
- The progress report should detail any relevant progress made by the team, including contributions of each member to this progress.
- The report is expected to have sufficient information to cover at least 1.5 pages and show enough progress for over a month that will have passed since the project proposals were submitted.

<u>Project Progress Report - Fall 2016 CPSC 545</u>

Students: Chen Gu, Zhishan Gu, Zijun Tang
Project dataset: NBA Shots Log
Data source: Kaggle
Data Size: 16M

Preliminary Analysis (by Chen Gu)

- Download and clean the dataset so that the data set can be used for modelling
  - Remove rows with negative TOUCH_TIME
  - Filling missing SHOT_CLOCK with according GAME_CLOCK value

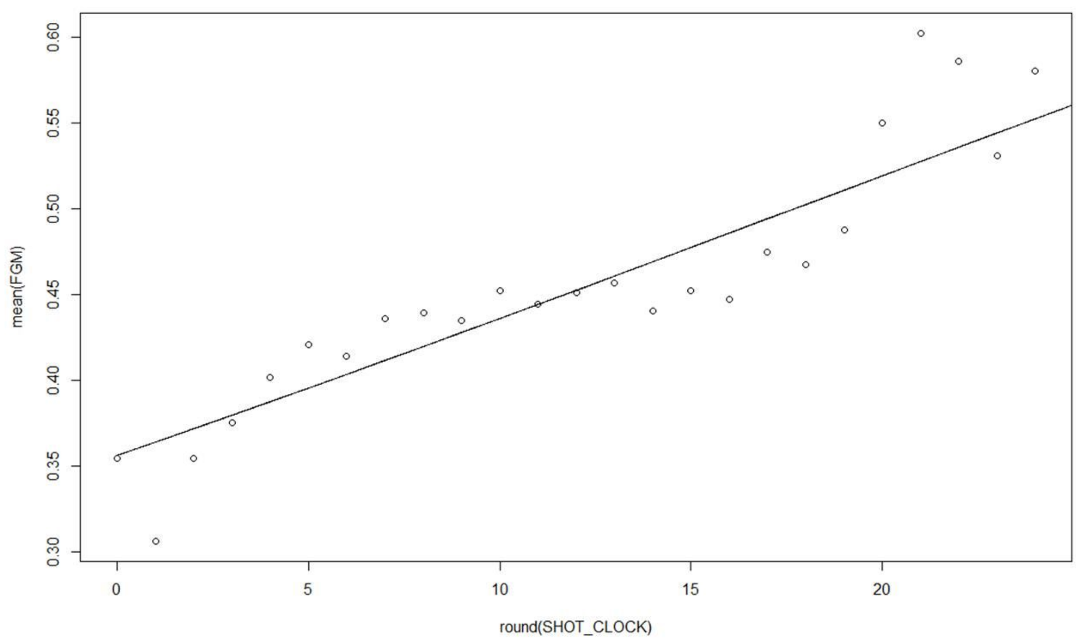
- Show/visualize the relationship between FGM and all other attributes (by Zhishan Gu)
  - FGM vs Location:

LOCATION	`mean(FGM)`
A	0.4484133
H	0.4565350

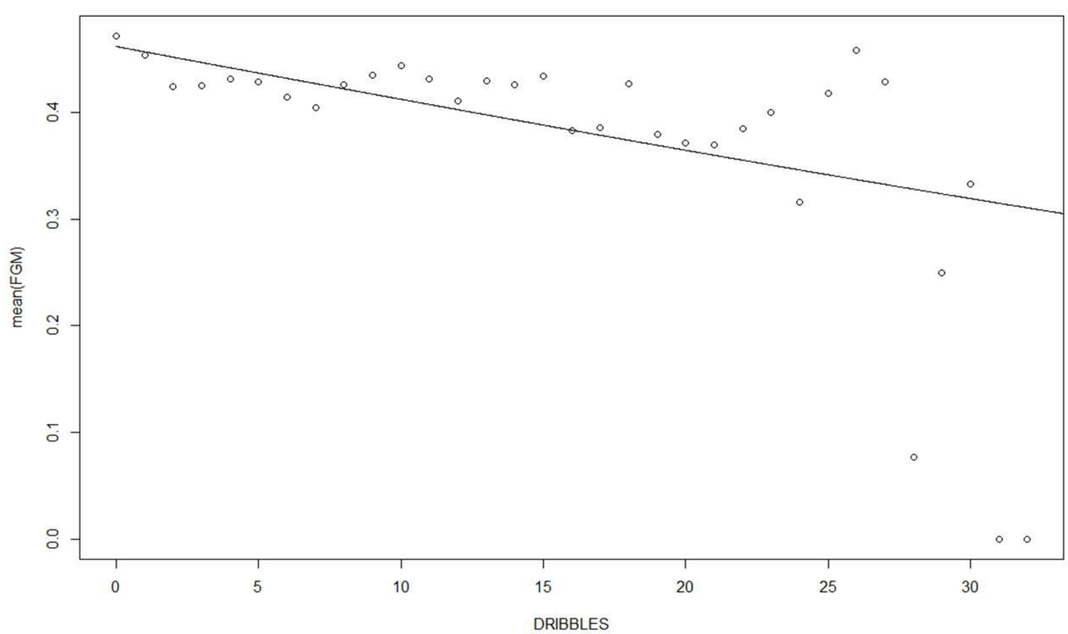
  - FGM vs Period:

PERIOD	`mean(FGM)`
1	0.4605283
2	0.4511074
3	0.4571420
4	0.4400989
5	0.3903509
6	0.4345238
7	0.3720930

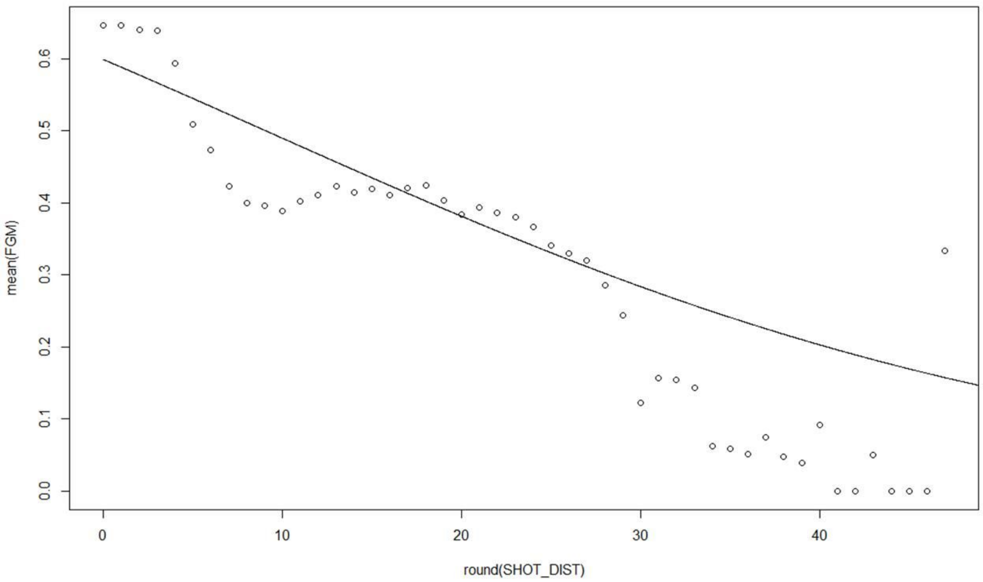
  - FGM vs Shot Clock:



- FGM vs Dribbles:



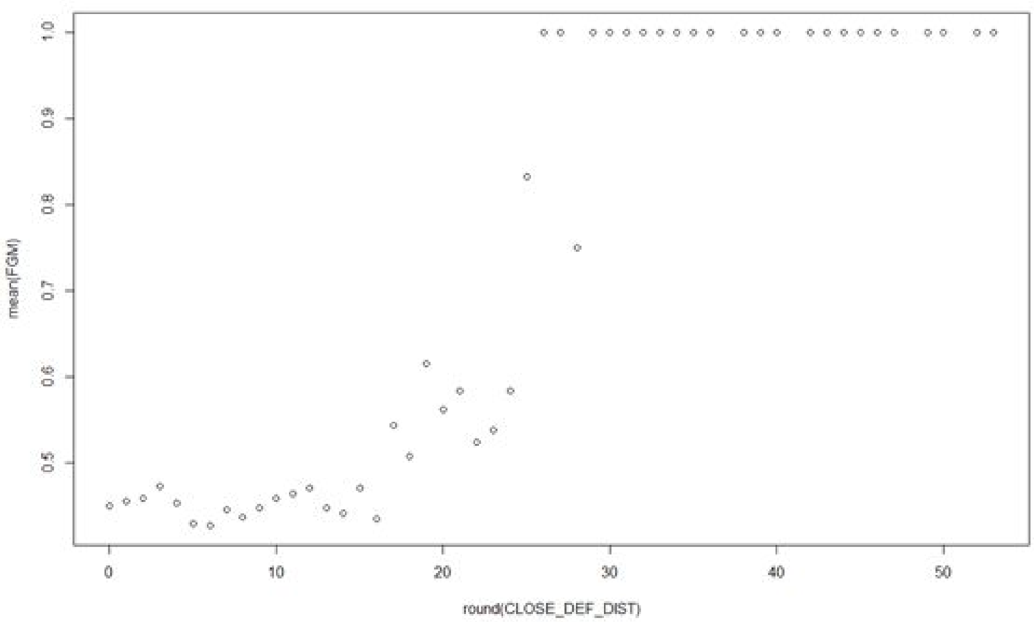
- FGM vs Shot Distance:



- FGM vs Shot Type:

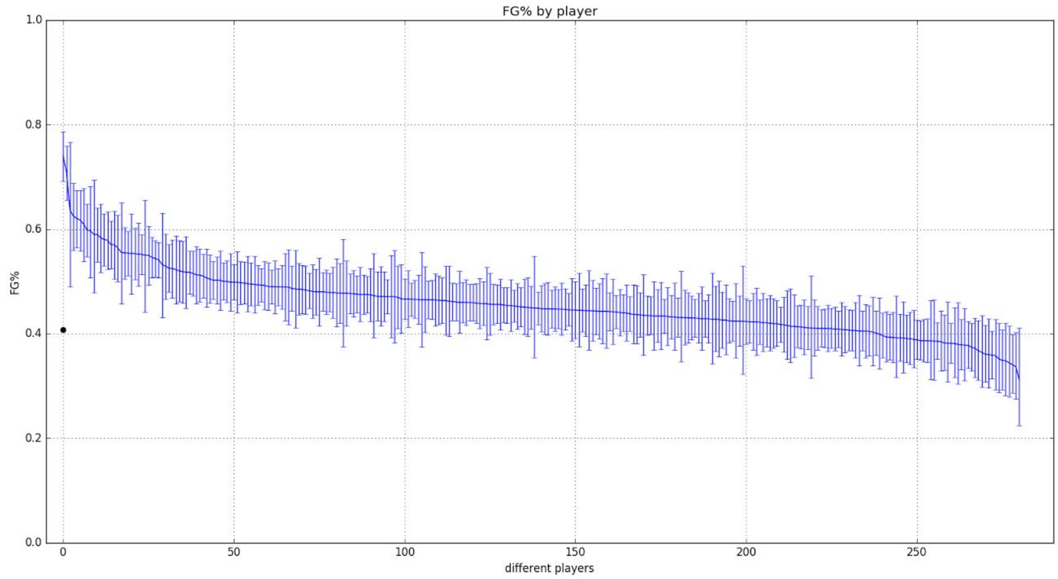
PTS_TYPE `mean(FGM)`
2 0.4888506
3 0.3515406

- FGM vs Closest Defender Distance



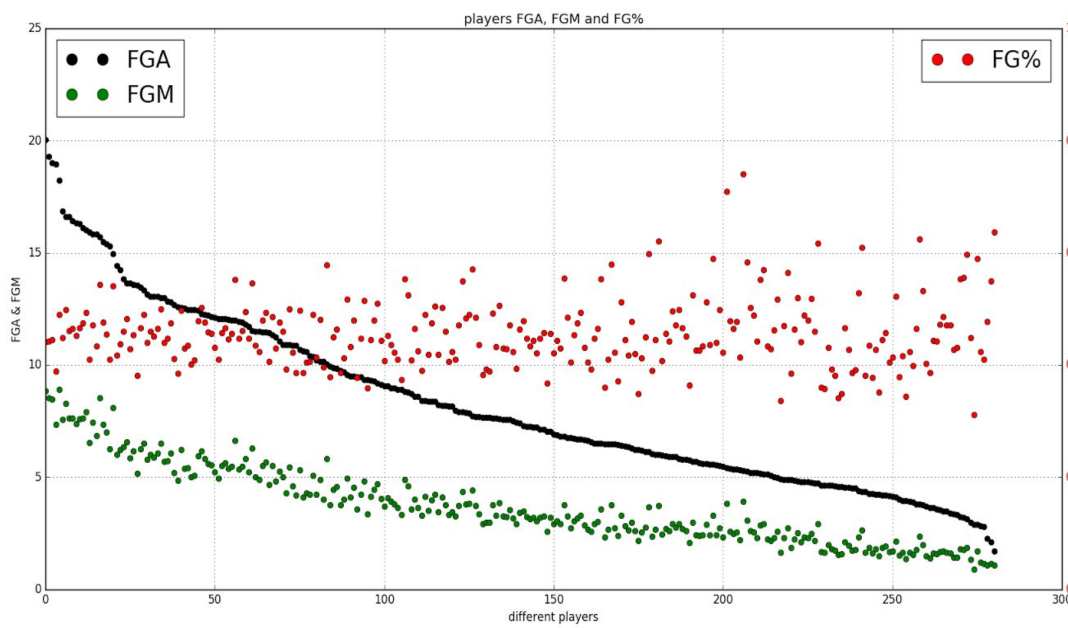
- Field Goal Ratio graph (by Chen Gu)

To have a rough understanding about the highest, lowest and average FG%, I draw a graph to illustrate these figures.



- Field Goal Made, Field Goal Attempted, Field Goal Ratio graph (by Chen Gu)

  I suspect that with FGA increases, FG% (efficiency) map drop. To test this guess, I draw a graph which indicates the relationship between FGM, FGA and FG%. Surprisingly, efficiency is not influced by Field Goal Attemps.



- Offensive player Ranking (by Chen Gu)

  For offensive player ranking, currently I user a pretty naive method. I sort players by Field Goal Ratio. Below is the ranking of players with FGA larger than or equal to

100. I will modify this method in the future, since it is not fair to compare a center and a guard by their FG%.

```
4.0 %
        PLAYER_ID             PLAYER    FGM    FGA  total_games       FG%  \
109092     201599     deandre jordan  243.0  328.0         62.0  0.740854
36622        2199     tyson chandler  207.0  292.0         54.0  0.708904
8408       203497        rudy gobert  135.0  216.0         57.0  0.625000
34368      201949      james johnson  183.0  295.0         49.0  0.620339
10523      202334           ed davis  166.0  269.0         57.0  0.617100
30838      201582      alexis ajinca  114.0  187.0         43.0  0.609626
124686     203486       mason plumlee 216.0  361.0         59.0  0.598338
95766        2365     chris andersen   71.0  119.0         38.0  0.596639
46812      202684   tristan thompson  204.0  346.0         62.0  0.589595
74776      203089        john henson  126.0  216.0         41.0  0.583333
```

- Defensive player Ranking (by Chen Gu)

For defensive player ranking, we the difference between a shooters' average FG% and FG% guarded by a certain defender. The larger the difference is, more effective the defense will be. We print top 10 defender who involved in at least 100 defense.

```
Run FGM_efficiency
4.0 %
       PLAYER_ID              PLAYER   DFGM   DFGA      DFG%      OFG%  \
7308        1889       Miller, Andre   48.0  154.0  0.311688  0.434543
108       204038  Galloway, Langston   59.0  175.0  0.337143  0.436138
1198      203120     Hamilton, Justin  39.0  102.0  0.382353  0.477245
4087        2748       Wright, Dorell  36.0  101.0  0.356436  0.447443
978       101106       Bogut, Andrew  162.0  412.0  0.393204  0.483025
10550     202407     Millsap, Elijah   45.0  128.0  0.351562  0.439633
440       203497        Gobert, Rudy  182.0  464.0  0.392241  0.477539
135       201579        Hibbert, Roy  188.0  471.0  0.399151  0.481228
5475      203093    Jones, Terrence   72.0  189.0  0.380952  0.461527
908         2754         Allen, Tony  107.0  295.0  0.362712  0.439788
```

Regression (by Zhishan Gu)

- Fitted a standard logistic regression and removed insignificant attributes from the model:

Coefficients:	Estimate	Std. Error	z value	P-value
(Intercept)	1.58E-01	2.53E-02	6.228	4.74E-10
LOCATIONH	2.75E-02	1.16E-02	2.376	0.0175
PERIOD2	-3.19E-02	1.62E-02	-1.972	0.0486
PERIOD3	7.78E-03	1.61E-02	0.483	0.6289
PERIOD4	-2.36E-02	1.66E-02	-1.419	0.1559
PERIOD5	-1.70E-01	7.14E-02	-2.382	0.0172
PERIOD6	1.11E-01	1.61E-01	0.689	0.4908
PERIOD7	-1.29E-01	3.23E-01	-0.4	0.6889
GAME_CLOCK	-2.00E-05	2.85E-05	-0.701	0.4834
SHOT_CLOCK	1.56E-02	9.90E-04	15.767	2E-16
DRIBBLES	3.19E-02	4.63E-03	6.894	5.41E-12
TOUCH_TIME	-6.31E-02	5.46E-03	-11.541	2.00E-16
SHOT_DIST	-6.40E-02	1.10E-03	-57.972	2.00E-16
PTS_TYPE3	8.39E-02	2.03E-02	4.138	3.51E-05
CLOSE_DEF_DIST	1.02E-01	2.73E-03	37.264	2.00E-16

- Using VIF (Variance inflation factor) test to remove attributes with collinearity

	VIF Test		
	GVIF	Df	GVIF^(1/(2*Df))
LOCATION	1.00031	1	1.000153
PERIOD	1.02034	6	1.001679
GAME_CLOCK	1.03724	1	1.018447
SHOT_CLOCK	1.09135	1	1.044675
DRIBBLES	7.48754	1	2.736337
TOUCH_TIME	7.61748	1	2.759978
SHOT_DIST	2.79769	1	1.672629
PTS_TYPE	2.30353	1	1.517737
CLOSE_DEF_DIST	1.56988	1	1.252947

- Refit the logistic model with the rest attributes:

Coefficients:					
	Estimate	Std. Error	z-value	Pr(>|z|)	
(Intercept)	0.0166469	0.0210611	0.790	0.4293	
LOCATIONH	0.0272171	0.0115509	2.356	0.0185	*
PERIOD2	-0.0280839	0.0161253	-1.742	0.0816	.
PERIOD3	0.0121570	0.0160497	0.757	0.4488	
PERIOD4	-0.0162074	0.0165541	-0.979	0.3276	
PERIOD5	-0.1575079	0.0710317	-2.217	0.0266	*
PERIOD6	0.1106633	0.1603299	0.690	0.4901	
PERIOD7	-0.1340025	0.3239996	-0.414	0.6792	
SHOT_CLOCK	0.0175909	0.0009664	18.203	<2e-16	***
DRIBBLES	-0.0183777	0.0017226	-10.669	<2e-16	***
SHOT_DIST	-0.0606711	0.0008297	-73.121	<2e-16	***
CLOSE_DEF_DIST	0.1036243	0.0027235	38.048	<2e-16	

Classifier (by Zijun Tang)

- SVM: using RBF kernel.
- Decision Tree
- Naive Bayes: The likelihood of the features is assumed to be Gaussian.

Validation (by Zijun Tang)

For all the three classifier, use 10 fold cross validation and stratified sampling.
Accuracy:

- SVM: 0.6036852 0.60103061 0.59806371 0.60134291 0.5950652 0.60580978 0.59815711 0.60135874 0.59472122 0.59401843
- Decision Tree: 0.54044347 0.53864772 0.53810119 0.54247345 0.5354884 0.545057 0.54052788 0.54021552 0.54044979 0.53506169
- Naive Bayes: 0.59775141 0.59423798 0.59353529 0.59814179 0.59420629 0.59831329 0.58488209 0.59003592 0.59237857 0.59503358

Future plan:

- Incorporate players' ranking into the regression formula
- Brake some of the numerical variables into couple categorical levels so that the regression curve can be smoother.
- Fit Lasso and Ridge regressions and compare the result with standard GLM (generalized linear model)
- Classifier analysis will be carried out.
- Random Forest Classifier will be adopted and compared with SVM, Decision Tree and Naive Bayes.