LOL computation: What can Roles do with more gold?

Siwei Hu

December 2, 2018

Abstract:

In League of Legend competition, there exists too much variables which can impact the result of one game. In this project, I want to research about how gold difference of each roles from two sides will affect the result of the game. I decide to use logistic regression model and interpret the coefficient of five roles' gold difference. I am also interested in building a multilevel logistic regression model using Team or League as level.Statistic transformation will be helpful in this project. After building the model, I will check it with binned residual plot and prediction.

Introduction:

I. Background:

LOL Game Map:

League of Legends (aka LOL) is a multiplayer online battle arena video game.

Area in Map: From the Map we can see there are two side of base and three main lanes. Except these lanes and base, map has a lot of jungle regions where is the home of monsters. (as shown Figure 1)

Roles: The circles on the Map shows different roles in each team. They are Attack Damage Carry(abb.ADC), Middle, Top, Jungle and Support. ADC and Support both called Bot. (as shown Figure 1)

Process: During the Competition, players will keep farming to earn gold to buy their weapon. Besides farming, some solo fights and team fights will happen to help team build gold difference.

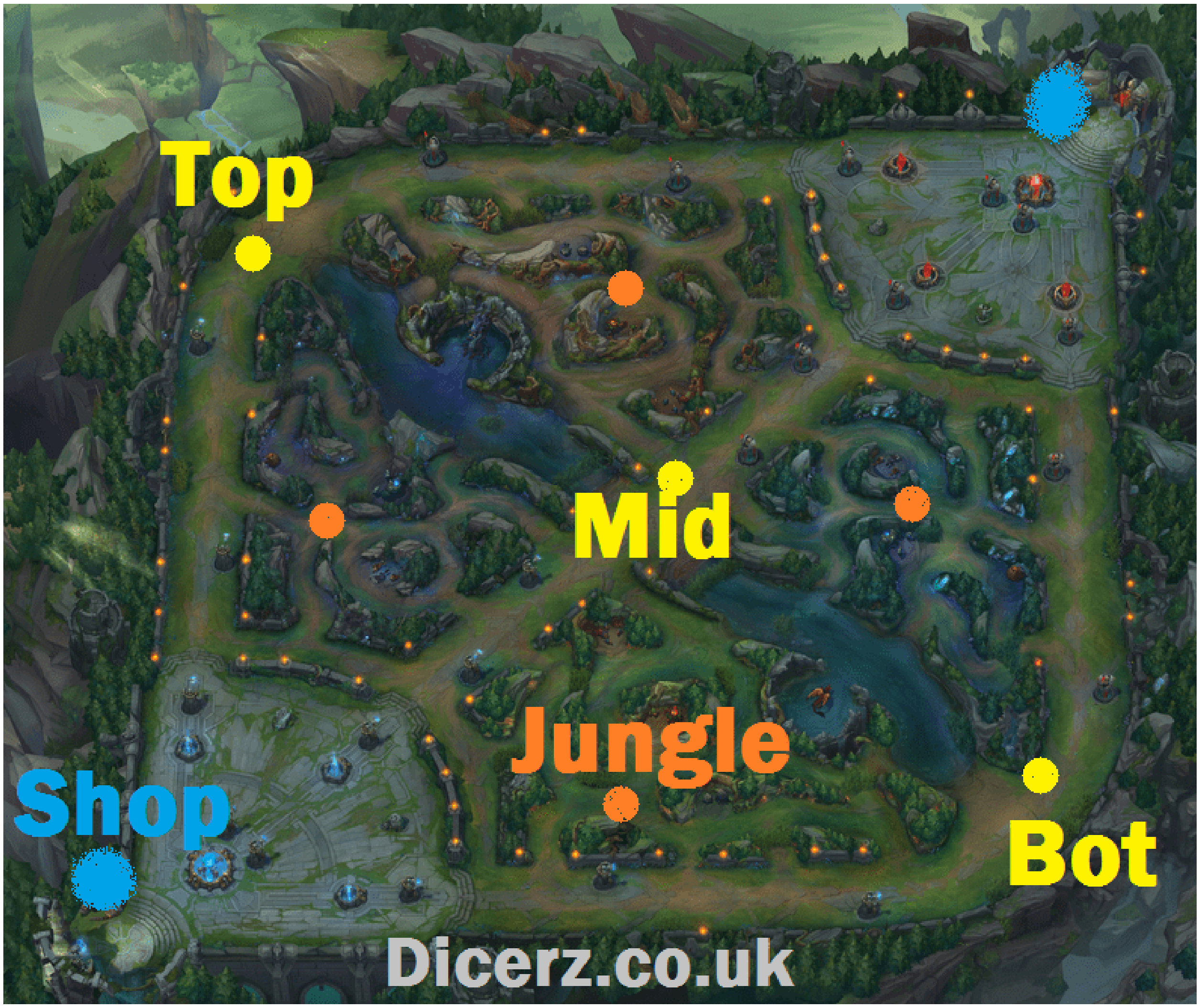


Figure 1: “GameMap”

II. Previous Work:

i. Data source

The raw data has 7620 competition from 242 game clubs and 15 Leagues. The raw data has 6 different datasets. Here are two dataset i used in my project:

LOL: the information of Teams,Year,Gamelength,Result ,Leagues and etc.

Gold: Each minutes gold change in each game. It has types includes

1.golddiff : gold difference between two team

2.goldADC : gold difference between two team's **ADC** role

3.golddSupport : gold difference between two team's **Support** role

4.golddMiddle : gold difference between two team's **Middle** role

5.golddTop : gold difference between two team's **Top** role

6.golddMiddle : gold difference between two team's **Middle** role

ii. Data cleaning

LOL: LOL dataset includes many background information. Select League,Year,Season,Type,blueTeamTa, bResult,rResult,redTeamTag,gamelength.

Gold part: Cleaned Gold data with deleting the NA cells and then rowsum each min to see the final gold. Calculated max gold difference between two teams and gold difference between each roles.

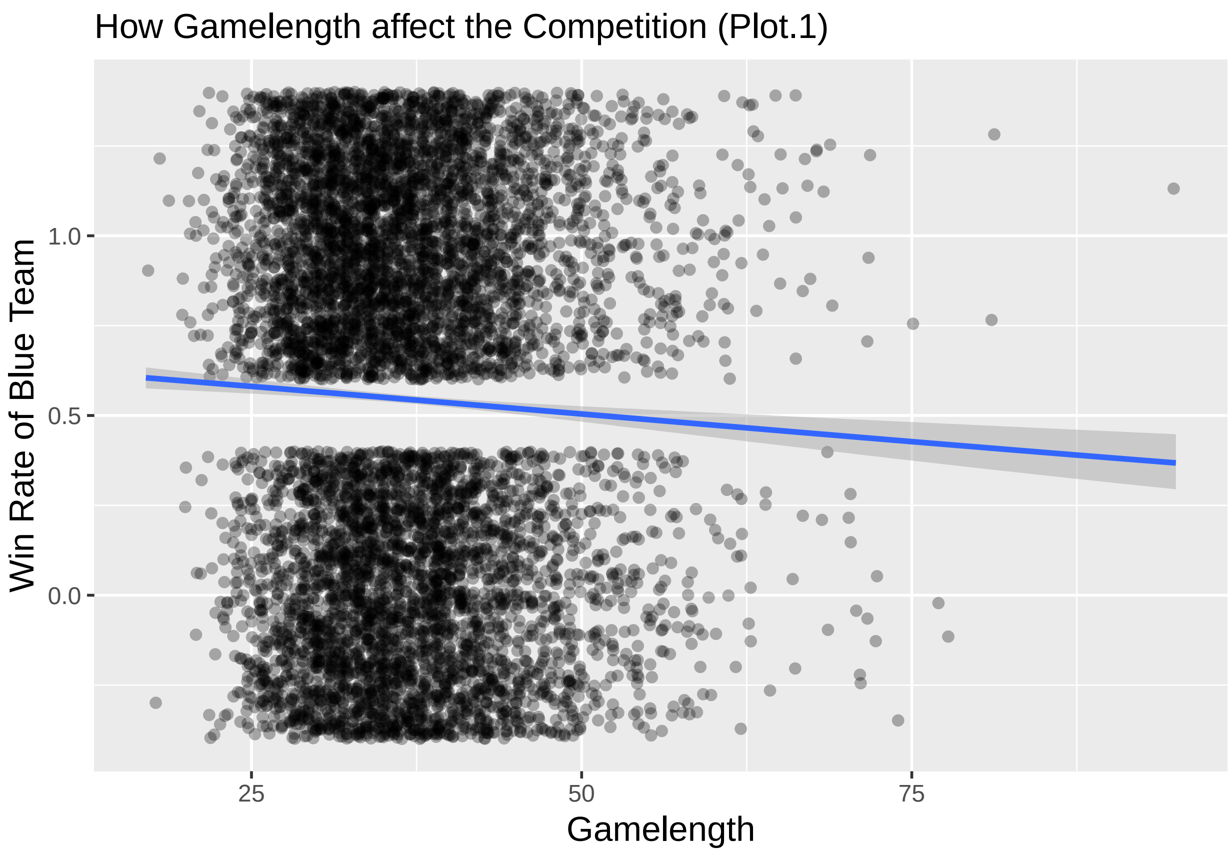
Monster part: Use number of blue team monster - red team monster to get monsterdiff. Sum different type dragons into Dragon column.

Structure part: Calculate how many different turrets blue team and red team get.

'data cleaning for LOL.R' includes data cleaning work. Write out 'LOL.all.csv' to include all variables i need.

iii. EDA:

Plot 1: Since there exist huge standard error at the beginning and end, we can only pay attention to middle part, like from 25 minutes to 45 minutes. According to this data set, the win rate becomes lower with longer gamelength. In conclusion, the coefficient of gamelength to Result should be very small.



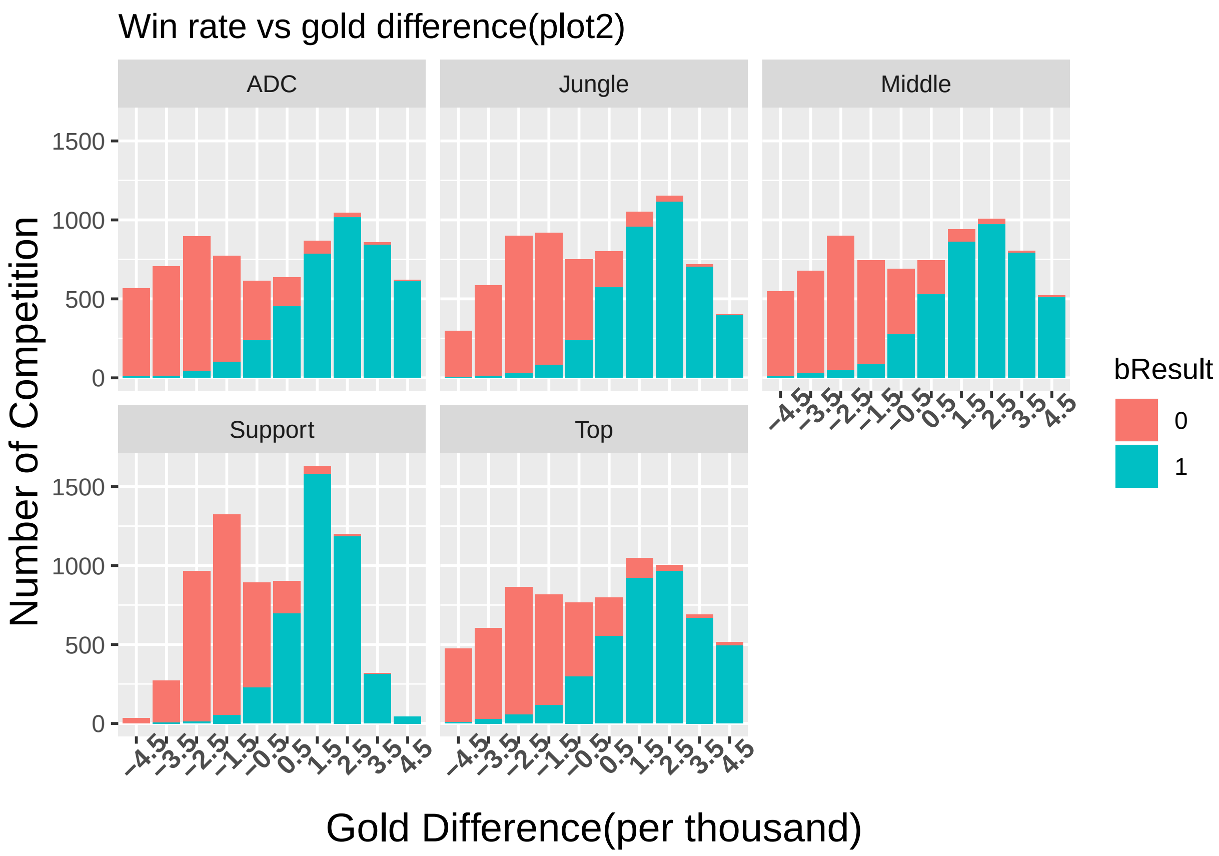
Plot2:

1. The most significant difference displayed in Support role. The gold difference of Support roles keep comparatively low. In most competition, it's from -3000 to 3000. The gold difference of Support should be high correlative to that of ADC because they are in same lane.

2. The distributions of gold difference for ADC,TOP and Middle are kind similar which make sense because each of them do farming in their own lane. So these three predictors have lower correlation between each other.

3. Jungle distribution is between Support and Other three. First, this role have jungle region resource but need to share with three main lanes. Second, Jungle always run to different lane to help their teammate, however, it's hard to say will win or not. So Jungle has lower farming ability than roles in main lanes. However, its gold difference may have correlation with other three lanes.

4. All plots show that higher gold difference have higher win rate. So their coefficient should all be positive.

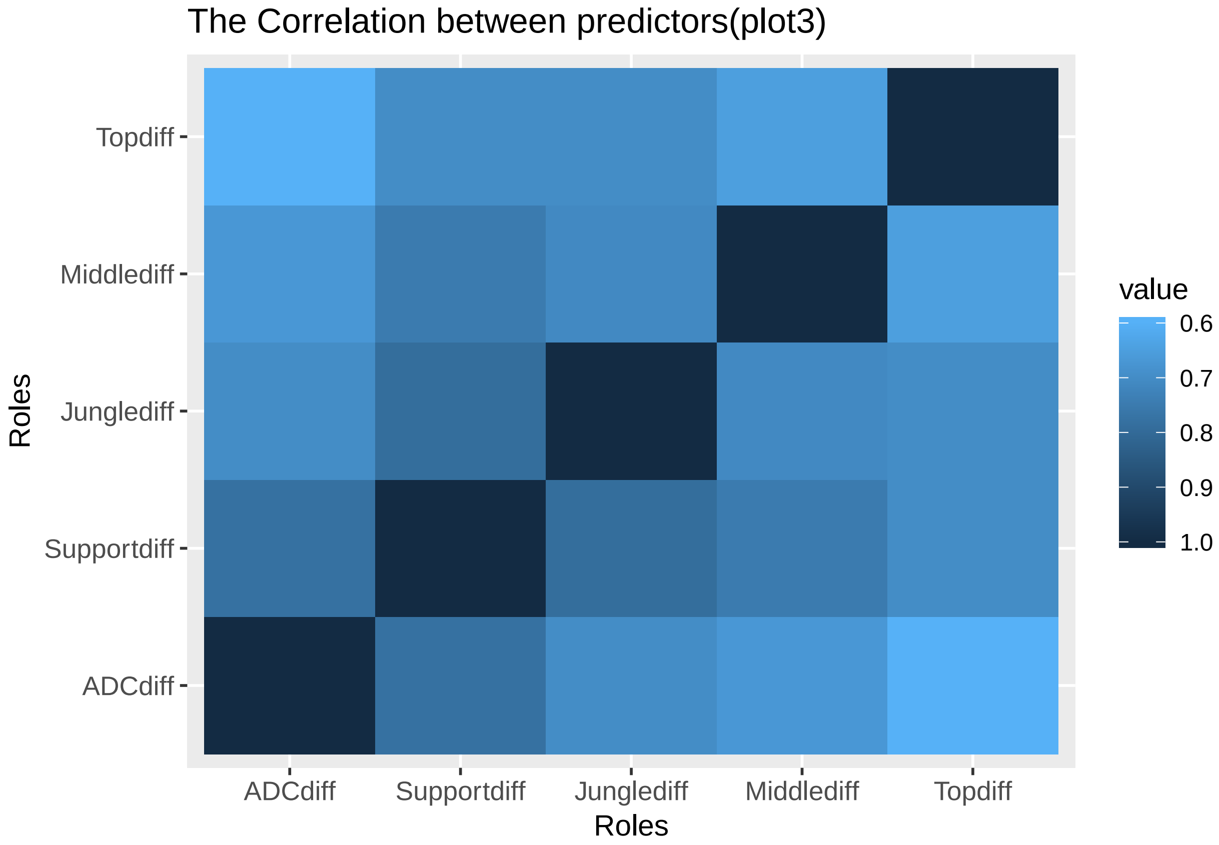


Plot3: From plot, we found Supportdiff and Junglediff both have high correlation to other three predictors. This make sense because these two Roles keep moving and try to gank three main lanes in the game.

For this project, I know the predictors in a game definitely have some correlations between each other. But I don't know how this affect my model and I will try to test it.

```
##             ADCdiff Supportdiff Junglediff Middlediff Topdiff
## ADCdiff        1.00        0.78       0.70       0.67    0.60
## Supportdiff    0.78        1.00       0.79       0.75    0.70
## Junglediff     0.70        0.79       1.00       0.71    0.70
## Middlediff     0.67        0.75       0.71       1.00    0.65
## Topdiff        0.60        0.70       0.70       0.65    1.00
```

```
##          Var1        Var2 value
## 1     ADCdiff     ADCdiff  1.00
## 2 Supportdiff     ADCdiff  0.78
## 3  Junglediff     ADCdiff  0.70
## 4  Middlediff     ADCdiff  0.67
## 5     Topdiff     ADCdiff  0.60
## 6     ADCdiff Supportdiff  0.78
```



Method:

I. Model used

I choose logistic regression model(binomial). The main reason is the outcome of project is win or not(1 or 0) which is binomial.

Then try no pooling and partial pooling to see how group: League affects the binomial logistic regression model.

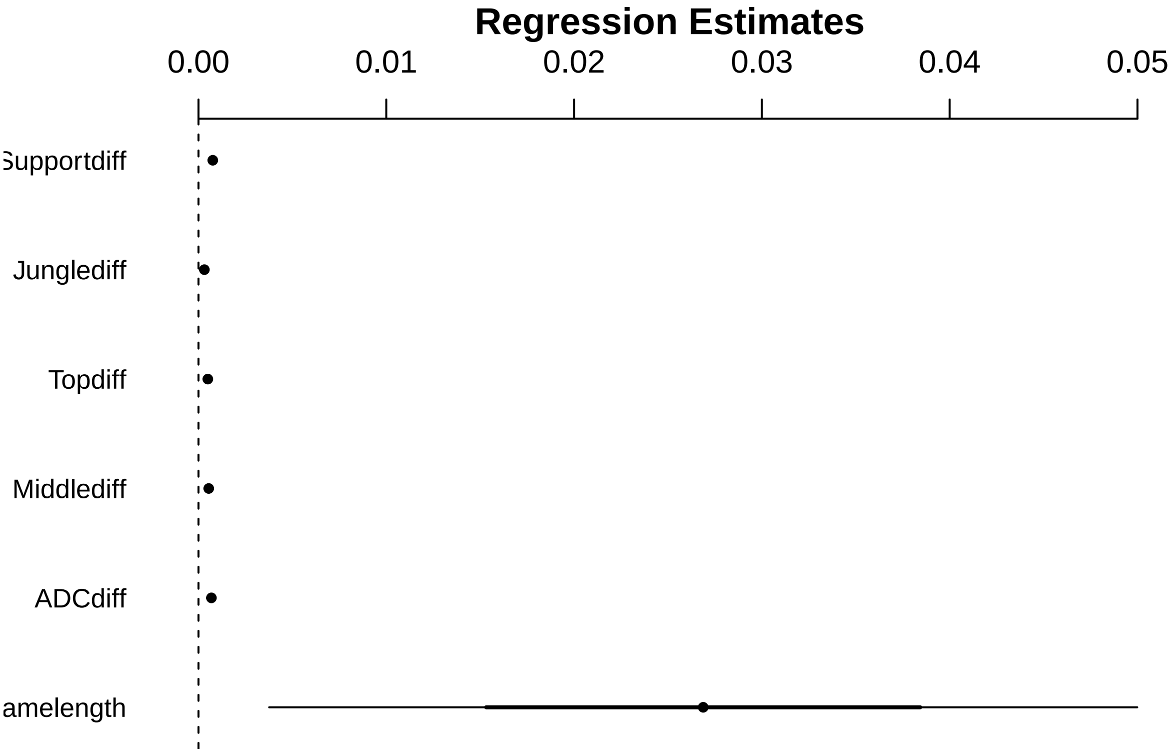
II. Modle Selection:

* complete pooling model:

Model1: The coefficients of model1 are all too small. Reason is that the predictors are large numbers. So Use statistic scale to solve this problem.

```
train=(LOL.all$Year < 2017)
LOL.after <- LOL.all[!train,]

model1 <- glm(bResult ~ gamelength + ADCdiff + Middlediff + Topdiff +
                        Junglediff + Supportdiff,
              family = binomial(link = "logit"),
              data = LOL.all, subset = train)
coefplot(model1)
```



* Scaling Model:

Model2: After doing scaling, now the coefficient can easier to understand. With each 1000 gold difference increase, we can see the influence on the game result from each role.

Coefficients in model2 are all significant and the Residual Deviance has reduced by 8963.4, a significant reduction in deviance, with a loss of six degrees of freedom. This mean add six predictors in the model increase goodness of fit of a generalized linear model.

```
model2 <- glm(bResult ~ gamelength + c.ADCdiff + c.Middlediff +
                c.Topdiff + c.Junglediff+ c.Supportdiff,
              family = binomial(link = "logit"), data = LOL.all)

summary(model2)
```

```
##
## Call:
## glm(formula = bResult ~ gamelength + c.ADCdiff + c.Middlediff +
##     c.Topdiff + c.Junglediff + c.Supportdiff, family = binomial(link = "logit"),
##     data = LOL.all)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7993  -0.0717   0.0161   0.0834   3.7266
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.494991   0.364606  -1.358   0.1746
## gamelength     0.017594   0.008263   2.129   0.0332 *
## c.ADCdiff      0.694596   0.037635  18.456  < 2e-16 ***
## c.Middlediff   0.553054   0.037131  14.895  < 2e-16 ***
## c.Topdiff      0.466936   0.033667  13.869  < 2e-16 ***
## c.Junglediff   0.334166   0.043887   7.614 2.65e-14 ***
## c.Supportdiff  0.666603   0.063313  10.529  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10451.2  on 7581  degrees of freedom
## Residual deviance:  1487.8  on 7575  degrees of freedom
## AIC: 1501.8
##
## Number of Fisher Scoring iterations: 8
```

* no pooling model:

Used League to do no pooling model, the coefficients of factors are all not significant. After comparing AIC of two model, i found model3 produces a probability distribution with the bigger discrepancy from the true distribution than model4.

```
model3 <- glm(bResult ~ gamelength + c.ADCdiff + c.Middlediff +
                c.Topdiff + c.Junglediff+ c.Supportdiff +
                factor(League)-1,
              family = binomial(link = "logit"), data = LOL.all)

AIC(model2,model3)
```

```
##        df      AIC
## model2  7 1501.765
## model3 21 1514.142
```

* partial pooling with random intercept:

Used League as group and 'glmer' function to build multilevel logistic regression model. From 'summary', the variance in random effect shows 0 which means there is no random effect. So model4 is same as a complete pooling model.

```
model4 <- glmer(bResult ~ gamelength + c.ADCdiff + c.Middlediff +
                  c.Topdiff + c.Junglediff+ c.Supportdiff +
                  (1|League),
                family = binomial(link = "logit"), data = LOL.all)
display(model4)
```

```
## glmer(formula = bResult ~ gamelength + c.ADCdiff + c.Middlediff +
##     c.Topdiff + c.Junglediff + c.Supportdiff + (1 | League),
##     data = LOL.all, family = binomial(link = "logit"))
##               coef.est coef.se
## (Intercept)   -0.49     0.36
## gamelength     0.02     0.01
## c.ADCdiff      0.69     0.04
## c.Middlediff   0.55     0.04
## c.Topdiff      0.47     0.03
## c.Junglediff   0.33     0.04
## c.Supportdiff  0.67     0.06
##
## Error terms:
##  Groups   Name        Std.Dev.
##  League   (Intercept) 0.00
##  Residual             1.00
## ---
## number of obs: 7582, groups: League, 15
## AIC = 1503.8, DIC = 1487.8
## deviance = 1487.8
```

* partial pooling with Random intercept and Random Slope:

Used League as group and 'glmer' function to build multilevel logistic regression model with varying slope and intercept. From 'summary', the variance in random effect shows number close to 0 which means there is nearly no random effect.

From Anova result of three multilevel logistic regression model,model5 shows same AIC number as model4 and 0 chi-sq value compare to model4. They are almost same.

model6 display a higher AIC and BIC compare to model4, which may produces a probability distribution with the bigger discrepancy from the true distribution than model4.

Actually, Multilevel model perform not well in this project. Although League or Team can be group level to do Mutilevel model,they are all top players of this game. So the influence from levels is not significant.

Finally, after Comparing their AIC and chi-square value, i think model2 is the model i want to select. Then I will check how well it is and interprete it.

```
model5 <- glmer(bResult ~ gamelength + c.ADCdiff + c.Middlediff +
                  c.Topdiff + c.Junglediff+ c.Supportdiff +
                  (1 + gamelength|League),
               family = binomial(link = "logit"), data = LOL.all)
```

```
## singular fit
```

```
model6 <- glmer(bResult ~ gamelength + c.ADCdiff + c.Middlediff +
                  c.Topdiff + c.Junglediff+ c.Supportdiff +
                  (1 + c.ADCdiff+ c.Middlediff + c.Topdiff +
                     c.Junglediff+ c.Supportdiff|League),
                family = binomial(link = "logit"), data = LOL.all)
```

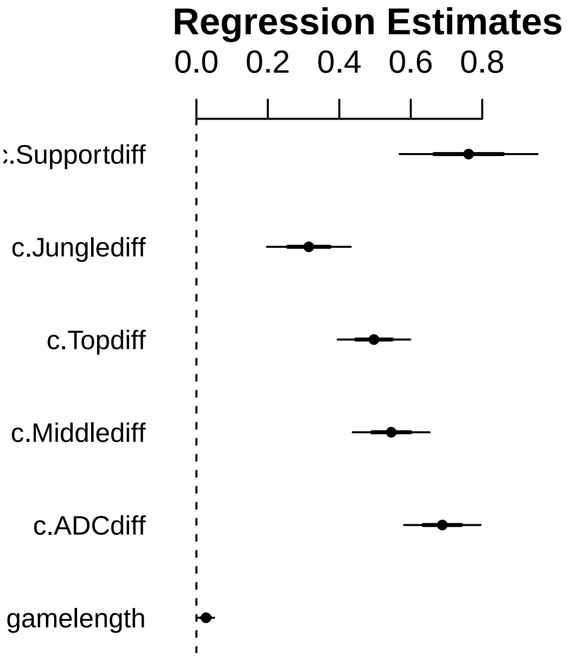
```
## singular fit
```

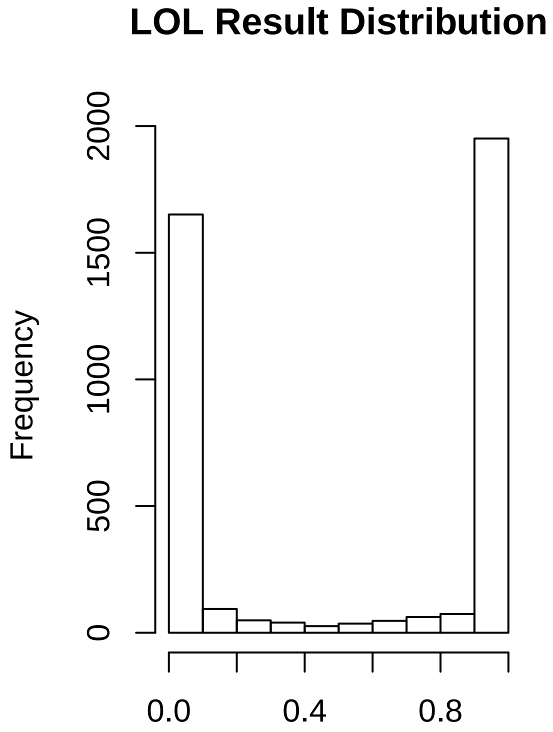
```
anova(model4,model5,model6,model2,model3)
```

```
## Data: LOL.all
## Models:
## model2: bResult ~ gamelength + c.ADCdiff + c.Middlediff + c.Topdiff + 
## model2:     c.Junglediff + c.Supportdiff
## model4: bResult ~ gamelength + c.ADCdiff + c.Middlediff + c.Topdiff + 
## model4:     c.Junglediff + c.Supportdiff + (1 | League)
## model5: bResult ~ gamelength + c.ADCdiff + c.Middlediff + c.Topdiff + 
## model5:     c.Junglediff + c.Supportdiff + (1 + gamelength | League)
## model3: bResult ~ gamelength + c.ADCdiff + c.Middlediff + c.Topdiff + 
## model3:     c.Junglediff + c.Supportdiff + factor(League) - 1
## model6: bResult ~ gamelength + c.ADCdiff + c.Middlediff + c.Topdiff + 
## model6:     c.Junglediff + c.Supportdiff + (1 + c.ADCdiff + c.Middlediff + 
## model6:     c.Topdiff + c.Junglediff + c.Supportdiff | League)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## model2  7 1501.8 1550.3 -743.88   1487.8                         
## model4  8 1503.8 1559.2 -743.88   1487.8  0.000      1     1.0000
## model5 10 1507.8 1577.1 -743.88   1487.8  0.000      2     1.0000
## model3 21 1514.1 1659.8 -736.07   1472.1 15.623     11     0.1557
## model6 28 1529.6 1723.7 -736.78   1473.6  0.000      7     1.0000
```

Result:

```
model <- glm(bResult ~ gamelength + c.ADCdiff + c.Middlediff +
               c.Topdiff + c.Junglediff+ c.Supportdiff,
             family = binomial(link = "logit"),
             data = LOL.all,subset = train)
par(mfrow=c(1,2),oma = c(1.5,0.5,1,0.5))
coefplot(model)
hist(fitted(model,type = "response"),main = "LOL Result Distribution",xlab = "Blue Team
```





I. Interprete:

The formula is $result = logit^{-1}(-0.85 + 0.027 * gamelength + 0.69 * c.ADCdiff + 0.55 * c.Middlediff + 0.5 * c.Topdiff + 0.31 * c.Junglediff + 0.76 * c.Supportdiff$

The coefficient of this formula:

Gamelength: With one more minutes,the blue team will increase nearly 0.53% win rate

ADCdiff : The blue team 's ADC earn 1000 gold more than the red team's ADC, the blue team will increase 17.3% win rate.

Supportdiff : The blue team 's Support earn 1000 gold more than the red team's Support, the blue team will increase 19% win rate.

Middlediff : The blue team 's Middle earn 1000 gold more than the red team's Middle, the blue team will increase 13.7% win rate.

Topdiff : The blue team 's Top earn 1000 gold more than the red team's Top, the blue team will increase 12.3% win rate.

Junglediff : The blue team 's Jungle earn 1000 gold more than the red team's Jungle, the blue team will increase 7.8% win rate.

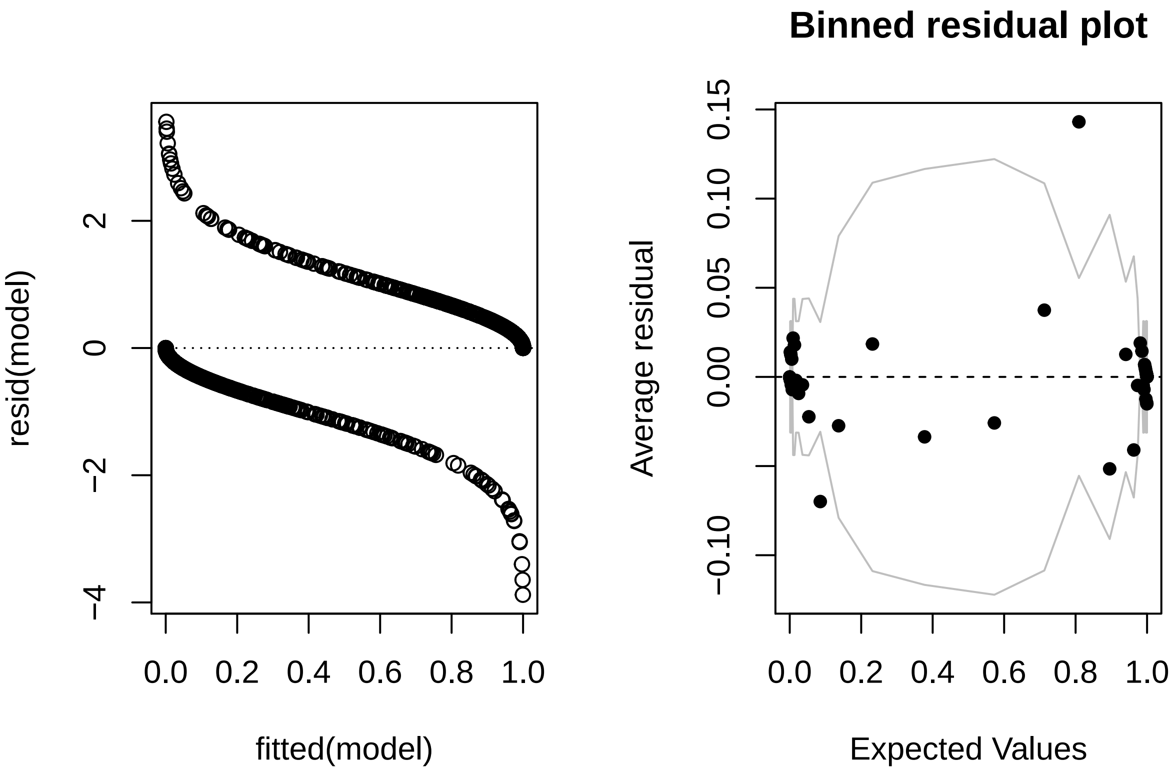
II. Model Check:

i. Fitted vs Residual plot and Binned residual plot.

The fitted vs residual plot show good result which upper part show lower residual, fitted value closer to 1 and down part show lower residual, fitted value closer to 0.

The binned plot is also good. Points are almost in the 2*se range of average residual. The reason most of them cluster at Expect value equal to 0 and 1 is the results of logistic regression are binary.

```
## glm(formula = bResult ~ gamelength + c.ADCdiff + c.Middlediff +
##     c.Topdiff + c.Junglediff + c.Supportdiff, family = binomial(link = "logit"),
##     data = LOL.all, subset = train)
##               coef.est coef.se
## (Intercept)   -0.85     0.51
## gamelength     0.03     0.01
## c.ADCdiff      0.69     0.05
## c.Middlediff   0.55     0.05
## c.Topdiff      0.50     0.05
## c.Junglediff   0.31     0.06
## c.Supportdiff  0.76     0.10
## ---
##   n = 4030, k = 7
##   residual deviance = 758.1, null deviance = 5565.3 (difference = 4807.2)
```



ii. Prediction using testset

i split data into train set and test set. Use test set to do prediction for checking model.

The result of accuracy is nearly 96% which is very high for prediction. This means model works well on inference.

```
##
## glm.pred    0    1
##        0 1509   63
##        1   78 1902
```

```
##   Predict.probs      RMSE        R2
## 1     0.9603041 0.1992384 0.8457731
```

iii. Variance Inflation Function

From plot3, I know there exist high correlation among five golddiff predictors. So i try to use variance inflation function to check the influence of their correlation to the model.

The result of 'VIF' function: all six predictions are closer to 1 which means the model are not affected by Multicollinearity problem.

```
##    gamelength     c.ADCdiff  c.Middlediff     c.Topdiff  c.Junglediff
##      1.031897      1.089392      1.021462      1.102427      1.167084
## c.Supportdiff
##      1.161564
```

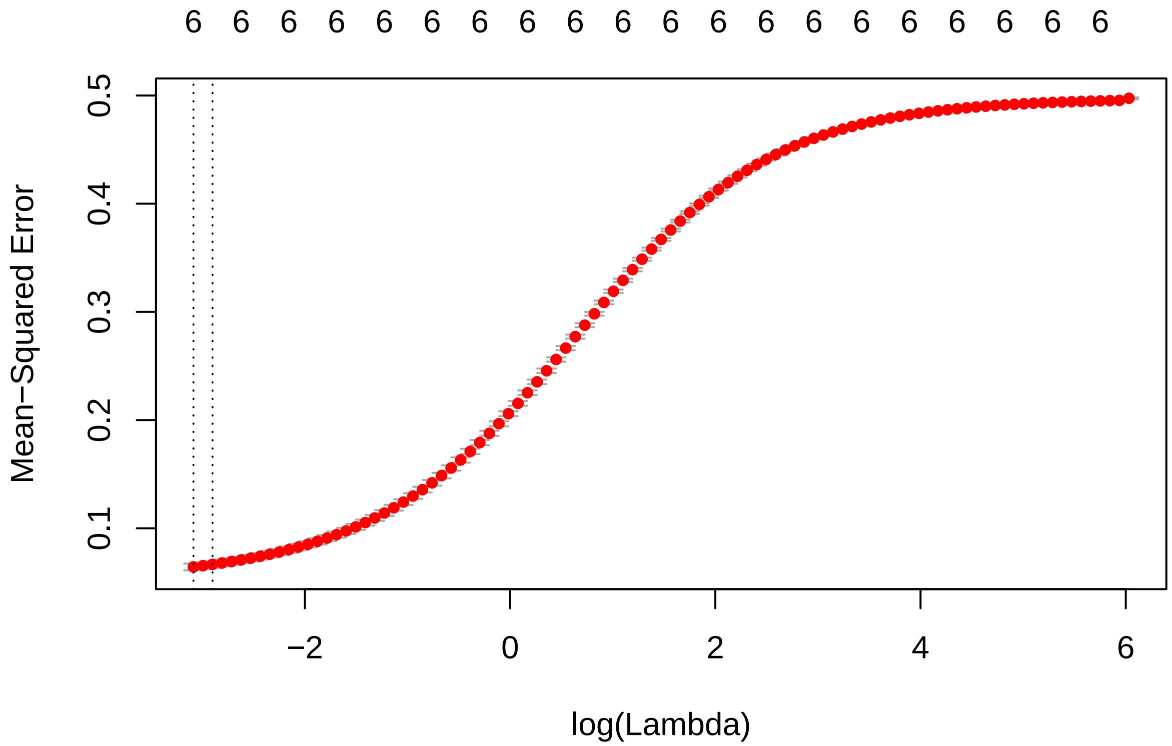
iv. Ridge regularisation using glmnet

Use 'cv.glmnet' function and traindata to do ridge regularization, function 'cv.glmnet'automatically select the optimal value of λ that minimises error called λ_{min}.(see plot cv.out)

But λ_{1se}, number within 1se of λ_{min}, is what we'll use in the rest of the computation. Get coefficient of predictors through 'coefficient' function to compare with coefficient of predictors in the logistic regression model. Then used testset and λ_{1se} to do prediction. Check the accuracy.

The coefficients of cv.out with λ_{1se} are much smaller than that of logistic regression, but Supportdiff still have highest influence on result and Jungle also keep lowest influence on result. After used testset to do the predictions, the accuracy is same as that of logistic regression.

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept)   0.042356985
## (Intercept)   .
## gamelength    0.002558043
## c.ADCdiff     0.259653228
## c.Supportdiff 0.370580380
## c.Junglediff  0.223688720
## c.Middlediff  0.241123526
## c.Topdiff     0.231877396
```



```
##
## ridge_pred    0    1
##          0 1509   63
##          1   78 1902
```

```
## [1] 0.9603041
```

v. Conclusion from Model check

The model performs good and can infer the result of competition through its six predictors. There is no multilinearity problem exists in logistic model.

Discussion:

I. Implication:

At first, I simply thought the coefficients of gold different predictors tell me how each role affects this game if they get 1000 more gold difference. In my opinion, some powerful roles like ADC and Middle should have high coefficients.

However, after I saw Supportdiff Coefficient, I thought there must be something wrong. How could Support affect this game so much if they get 1000 more golddiff?

After thinking, I know where's my mistake. From Plot2, we know in most of competitions, gold difference of Support are in range -3000~3000. However, gold difference of other roles have much equative possible to be in different intervals(each 1000 golddiff one interval) from -4000~4000.

In conclusion, generally, Support should be the role get least gold in each competition. If gold difference of Support increase 1000, the whole team gold difference highly possible to increase 5000+, this made higher possibility to "win". So higher coefficients of gold difference of each roles also represented more difficult to increase gold difference.

II. Limitation

The function of this model is doing a inference. If team know these predictors and put them in the model, the model will give a good estimation about result. But if team know about final gold difference, team also should know the result. It is not useful to one team.

III. Further direction

1. Improve model:

Collect data and add predictor which can evaluate the level of one team. The evaluation of result become fairer.

2. Time series model:

Collect one team's data as much as possible and build time series model to predicte what will happen next minues in game according to their perious one minute's behavior.

Reference:

1. Kaggle Leagle of Legend: https://www.kaggle.com/chuckephron/leagueoflegends
2. LOL Map: http://i1.wp.com/dicerz.co.uk/wp-content/uploads/2014/11/ Lol-season-5-map-beginners-guide1.png
3. Andrew Gelman Jennifer Hill, Data Analysis using Regression and Multilevel/Hierarchical Models, Cambridge

4.Model Comparisons , Frederick A.A. Kingdom, Nicolaas Prins, in Psychophysics (Second Edition), 2016

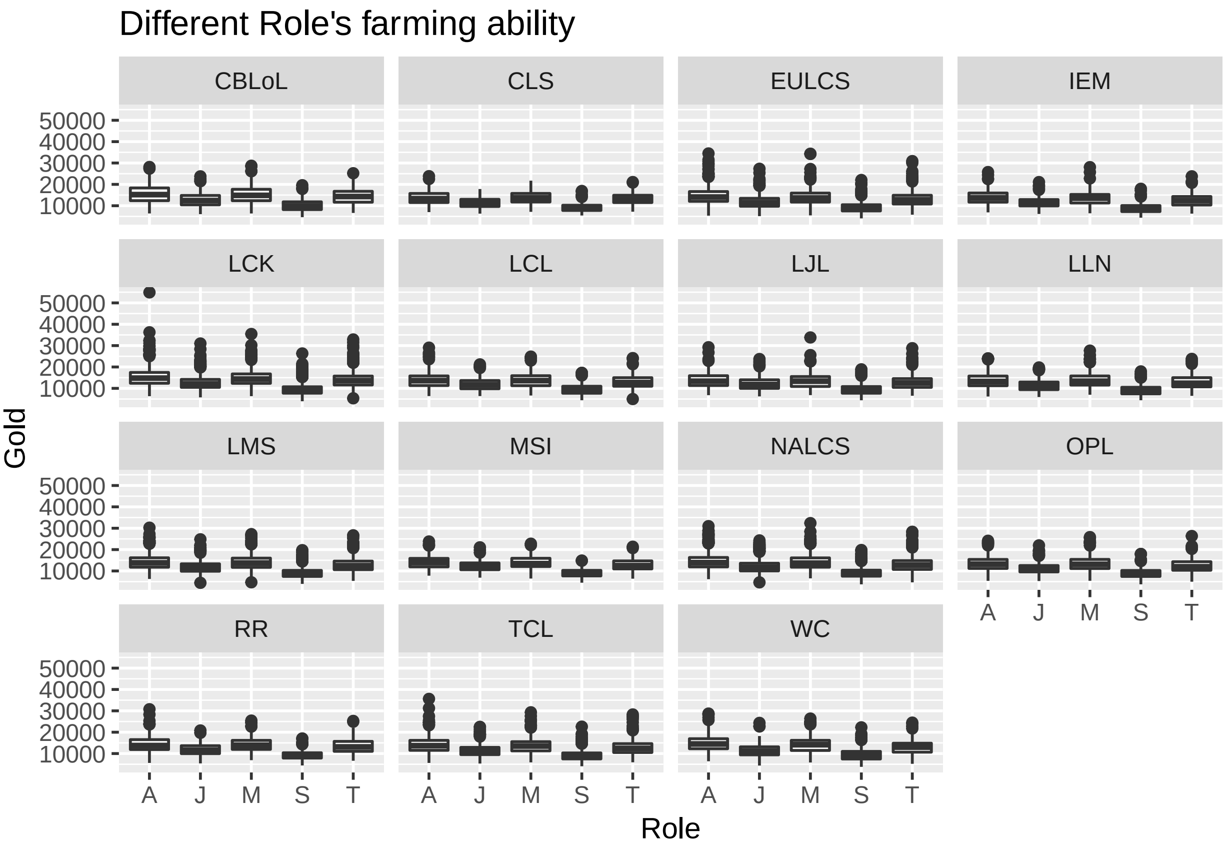
5.Regression Model Diagnostics,Multicollinearity Essentials and VIF in R, kassambara, 11/03/2018

6.Ridge regularisation using R: https://eight2late.wordpress.com/2017/07/11/a-gentle-introduction-to -logistic-regression-and-lasso-regularisation-using-r/

Appendix:

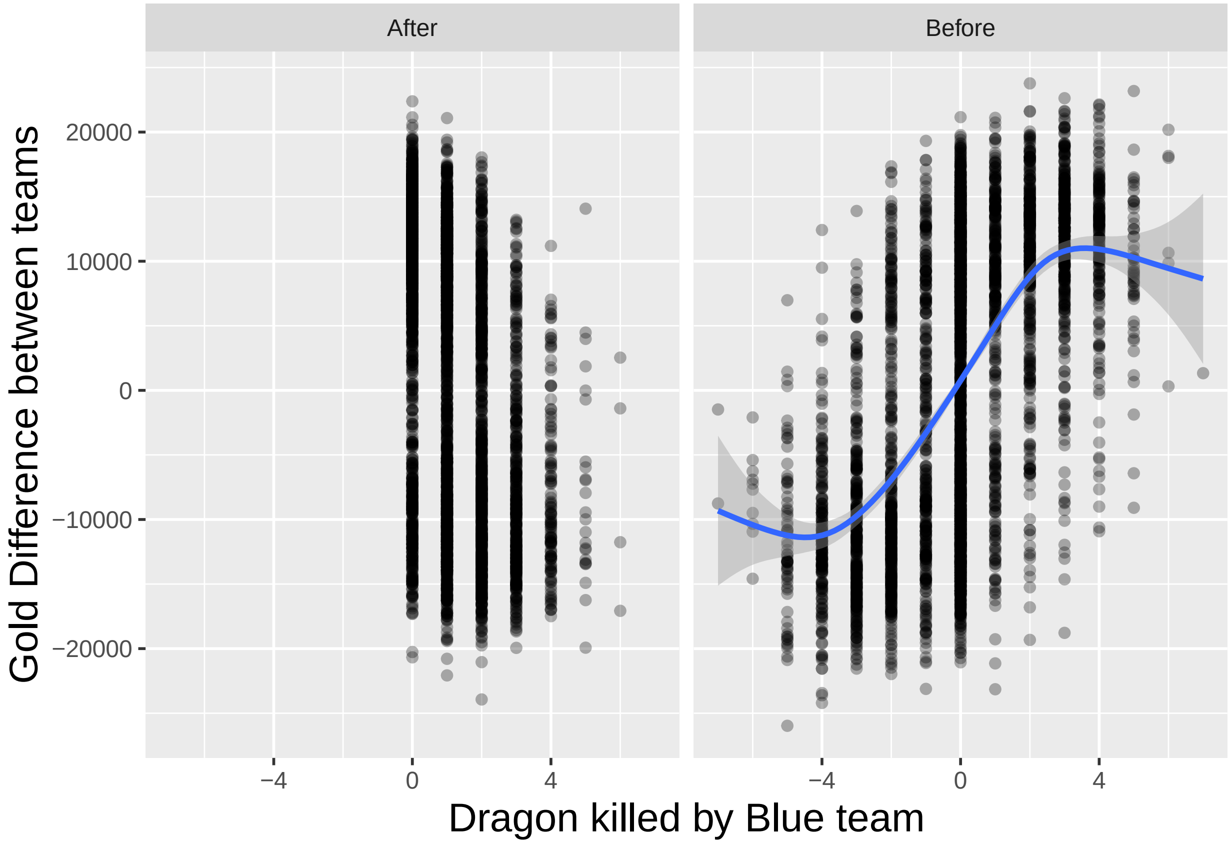
1. Role farming ability

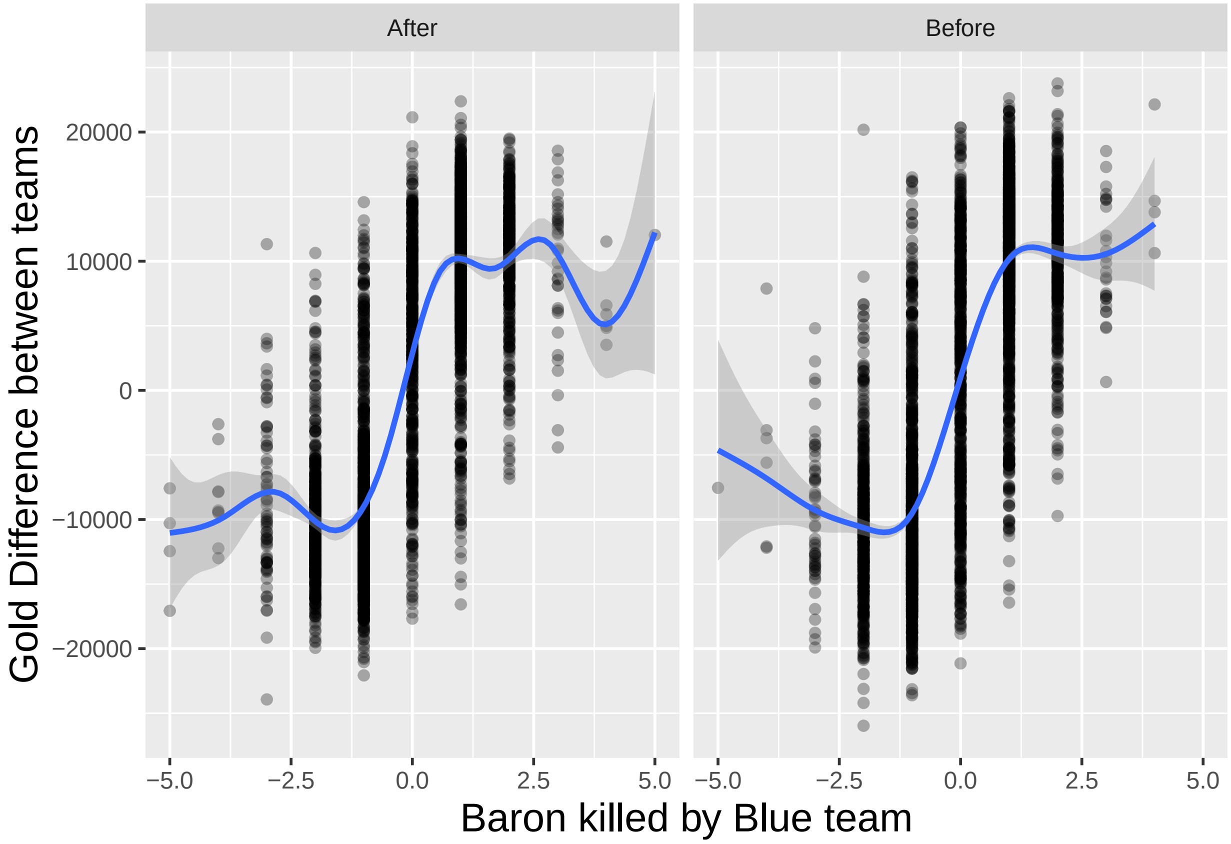
Build a facet.grid for each league. I show different Roles' farming ability in different League.



2. Monster Effect on Gold difference

Plot 'geom_smooth' and scatter plot to see the influence of dragon and baron before & after the change in jungle field (2017)


After
Before
Gold Difference between teams
20000
10000
0
−10000
−20000
−4
0
4
−4
0
4
Dragon killed by Blue team


After
Before
Gold Difference between teams
20000
10000
0
−10000
−20000
−5.0
−2.5
0.0
2.5
5.0
−5.0
−2.5
0.0
2.5
5.0
Baron killed by Blue team