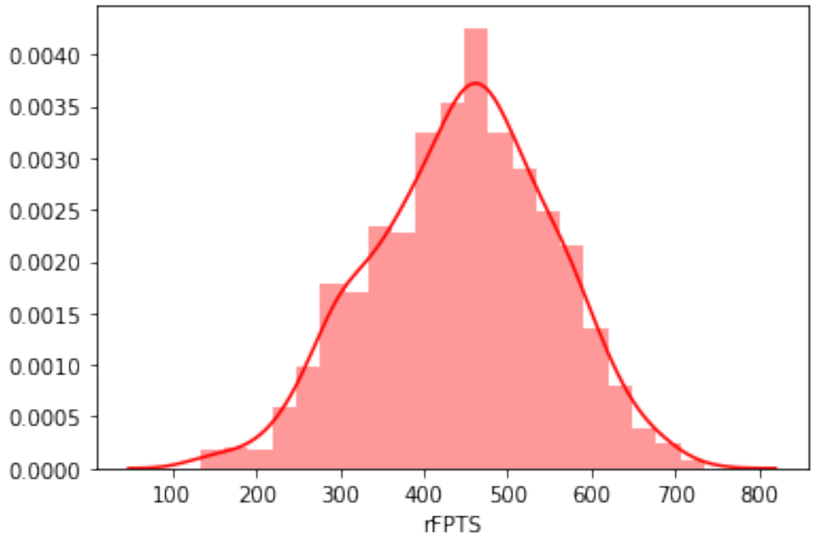
Back Testing the Model

Using Projections

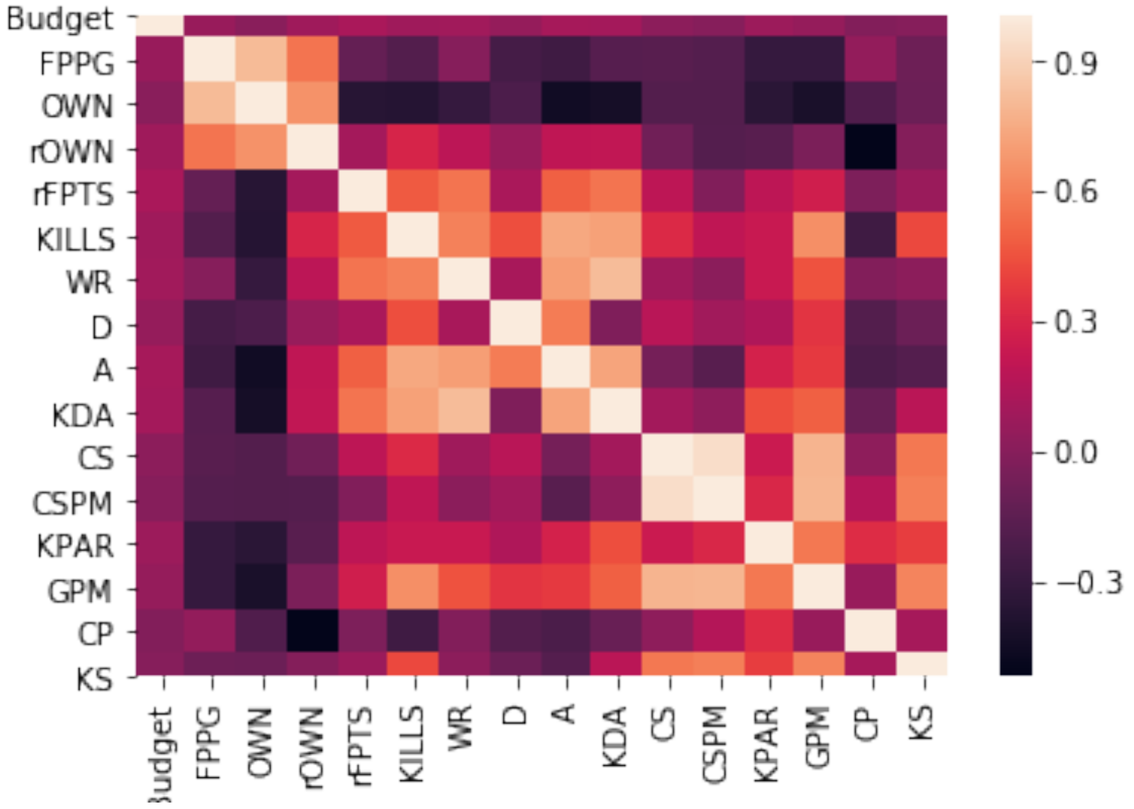
April 11 2020

Lineups built from fantasy points

- Optimizer Settings:
- 4- 3 stack….



Lineup Distribution



Lineup correlation heat map

Optimizer Settings

```
[ ]:
```

```
[3]:
optimizer = get_optimizer(Site.DRAFTKINGS_CAPTAIN_MODE,Sport.LEAGUE_OF_LEGENDS)
```

```
[ ]:
```

```
[5]:
optimizer.load_players_from_csv('{date}/lolclean{date}.csv'.format(date=timestr))
optimizer.set_deviation(0.05, 0.1)
#optimizer.set_max_repeating_players(4)
optimizer.add_stack(TeamStack(4, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP', 'TEAM'],
                              max_exposure=0.35))
optimizer.add_stack(TeamStack(3, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP', 'TEAM']))
#optimizer.add_stack(TeamStack(3, max_exposure=0.5, max_exposure_per_team={'MIA': 0.6}))  # stack 3 players from sa
optimizer.set_min_salary_cap(49400)
#optimizer.restrict_positions_for_opposing_team(['TEAM'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
#optimizer.restrict_positions_for_opposing_team(['CPT'], ['TEAM', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
#optimizer.restrict_positions_for_opposing_team(['TOP'], ['CPT', 'TEAM', 'JNG', 'MID', 'ADC','SUP'])
#optimizer.restrict_positions_for_opposing_team(['JNG'], ['CPT', 'TOP', 'TEAM', 'MID', 'ADC','SUP'])
#optimizer.restrict_positions_for_opposing_team(['MID'], ['CPT', 'TOP', 'JNG', 'TEAM', 'ADC','SUP'])
#optimizer.restrict_positions_for_opposing_team(['ADC'], ['CPT', 'TOP', 'JNG', 'MID', 'TEAM','SUP'])
#optimizer.restrict_positions_for_opposing_team(['SUP'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','TEAM'])
exporter = CSVLineupExporter(optimizer.optimize(1000, randomness = True))
```

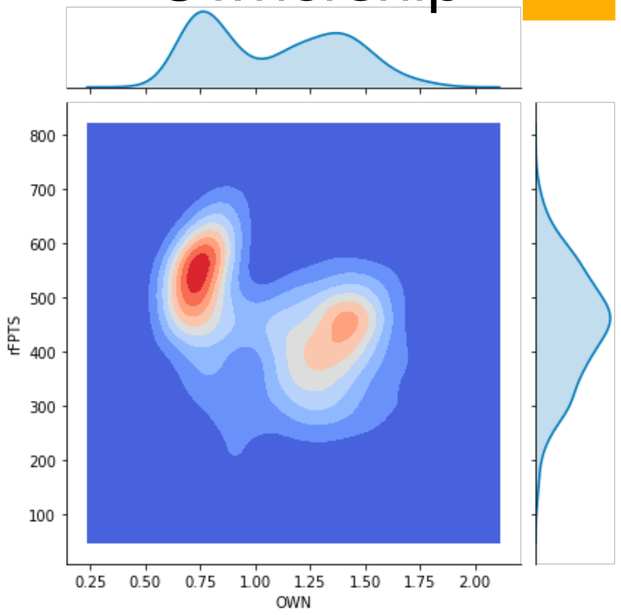
```
[ ]:
```

```
[6]:
exporter.export('{date}/lolpoop{date}.csv'.format(date=timestr))
```

LoL Statistics relationship to lineup fantasy points

Ownership

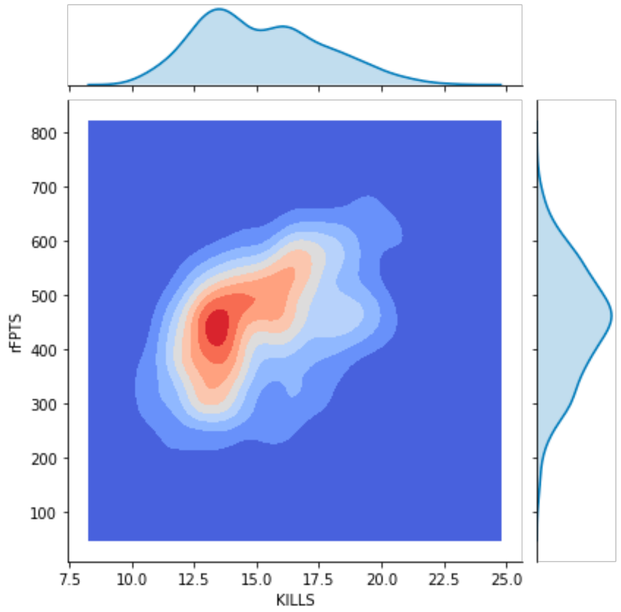
-0.36



There seems to be linear relationship with projected ownership and rFPTS. The two blobs show concentration of lineups. The slope of each blob is the same and the lines can be considered a family of curves

Kills

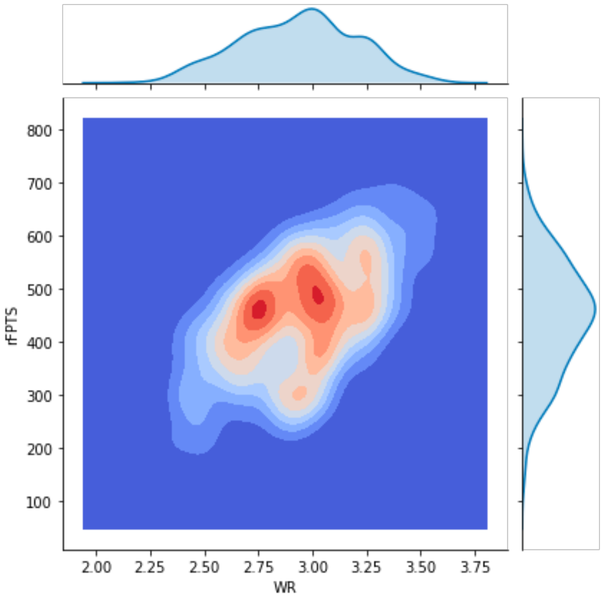
0.48



Lineups with higher Kills will score more fantasy points

Win Rate

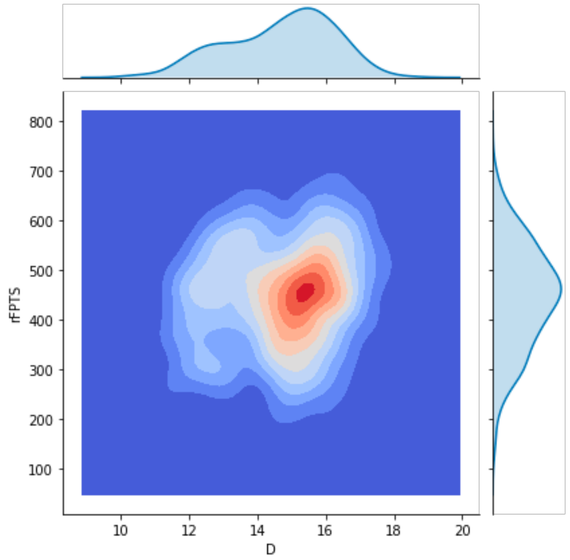
0.56



Lineups with higher WR over the set will have score more fantasy points. The local minimums will vary within +/- 0.25

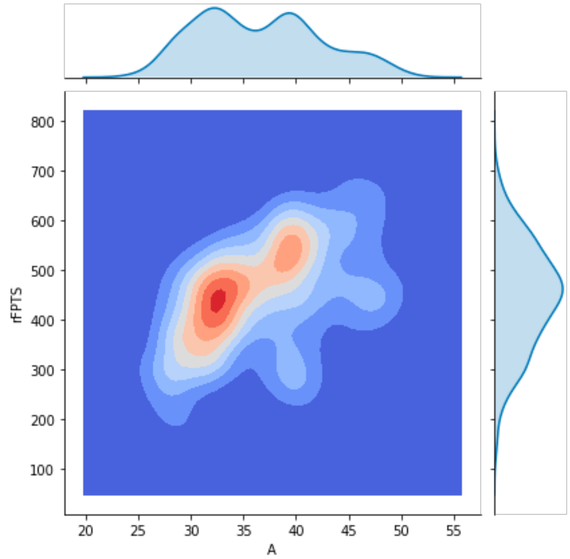
LoL Statistics relationship to lineup fantasy points

Deaths 0.12



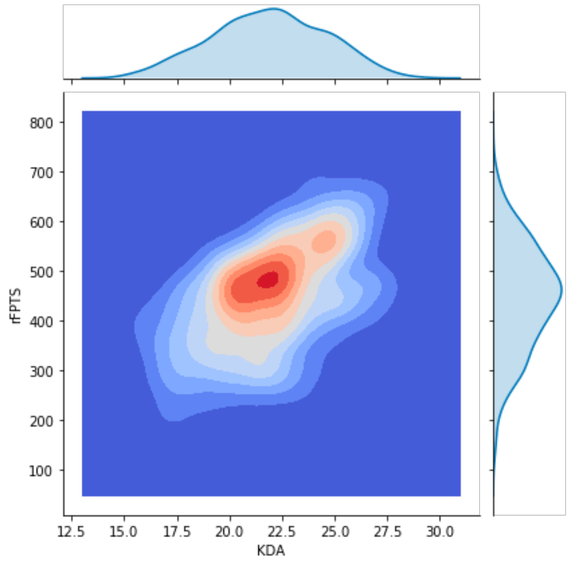
There seems to be no correlation between the number of deaths and fantasy point scored within a lineup

Assists 0.49



There appears to be correlation between the assists and fantasy points scored. You get points for kills and assists and right now I'm stacking all my lineups. The assists could show which teams are better to stack

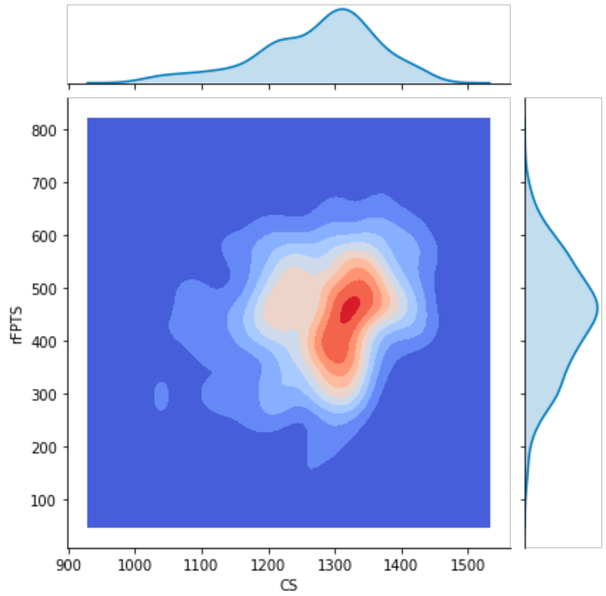
KDA 0.56



There appears to be correlation between KDA and fantasy points. Since kills and assists are involved in scoring this could give us a holistic view of scoring

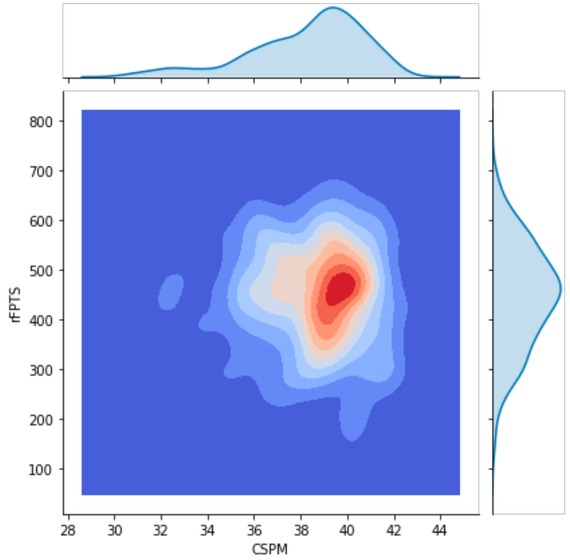
LoL Statistics relationship to lineup fantasy points

Creep Score 0.19



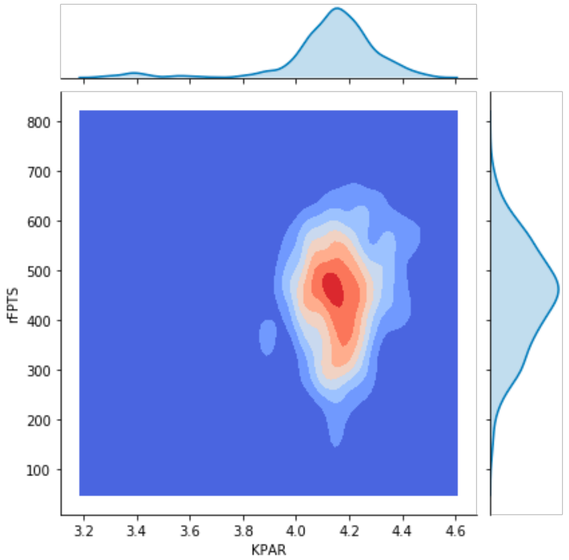
There appears to be no correlation between creep score and fantasy points

Creep Score/ Minute -0.18



There appears to be no correlation between creep score per minute and fantasy points

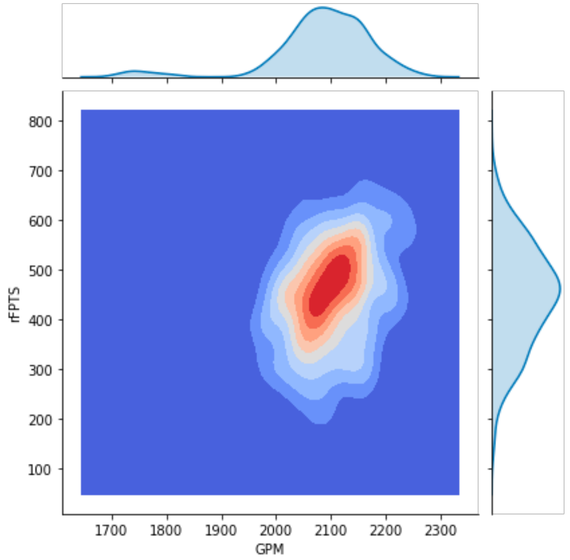
Kill Participation -.18



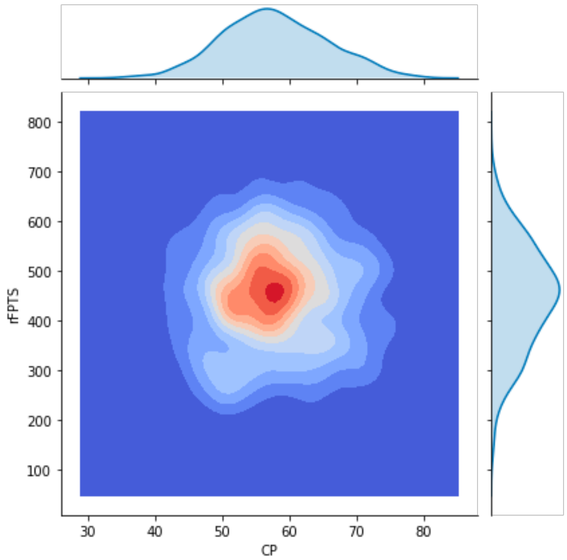
There appears to be no correlation between kill participation and fantasy points

LoL Statistics relationship to lineup fantasy points

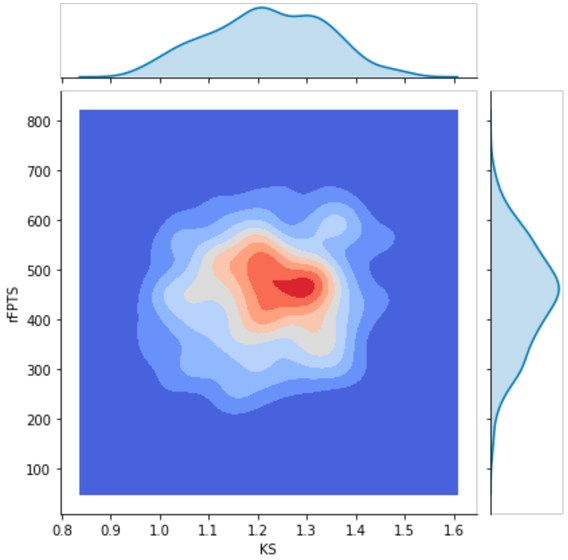
Gold Per Minute **0.26**



Champion Points **-0.04**



Kill Streak **0.06**



At first glance their isn't an immediate relationship between fantasy points and stat of interest. If we conduct a coordinate transformation- switching from cartesian to polar coordinates- we can observe a cyclical behavior within lineups. This could be useful as a filter after our machine learning algorithm predicts a variable of interest

Optimizer Setting 2

In [ ]:
```
1
```

In [3]:
```
1 optimizer = get_optimizer(Site.DRAFTKINGS_CAPTAIN_MODE,Sport.LEAGUE_OF_LEGENDS)
```

In [ ]:
```
1
```

In [5]:
```
1  optimizer.load_players_from_csv('{date}/lolclean{date}.csv'.format(date=timestr))
2  optimizer.set_deviation(0.05, 0.1)
3  #optimizer.set_max_repeating_players(4)
4  optimizer.add_stack(TeamStack(4, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'],
5                                max_exposure=0.35))
6  optimizer.add_stack(TeamStack(2, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP']))
7  #optimizer.add_stack(TeamStack(3, max_exposure=0.5, max_exposure_per_team={'MIA': 0.6}))  # stack 3 players from sa
8  optimizer.set_min_salary_cap(49400)
9  #optimizer.restrict_positions_for_opposing_team(['TEAM'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
10 #optimizer.restrict_positions_for_opposing_team(['CPT'], ['TEAM', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
11 #optimizer.restrict_positions_for_opposing_team(['TOP'], ['CPT', 'TEAM', 'JNG', 'MID', 'ADC','SUP'])
12 #optimizer.restrict_positions_for_opposing_team(['JNG'], ['CPT', 'TOP', 'TEAM', 'MID', 'ADC','SUP'])
13 #optimizer.restrict_positions_for_opposing_team(['MID'], ['CPT', 'TOP', 'JNG', 'TEAM', 'ADC','SUP'])
14 #optimizer.restrict_positions_for_opposing_team(['ADC'], ['CPT', 'TOP', 'JNG', 'MID', 'TEAM','SUP'])
15 #optimizer.restrict_positions_for_opposing_team(['SUP'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','TEAM'])
16 exporter = CSVLineupExporter(optimizer.optimize(1000, randomness = True))
```

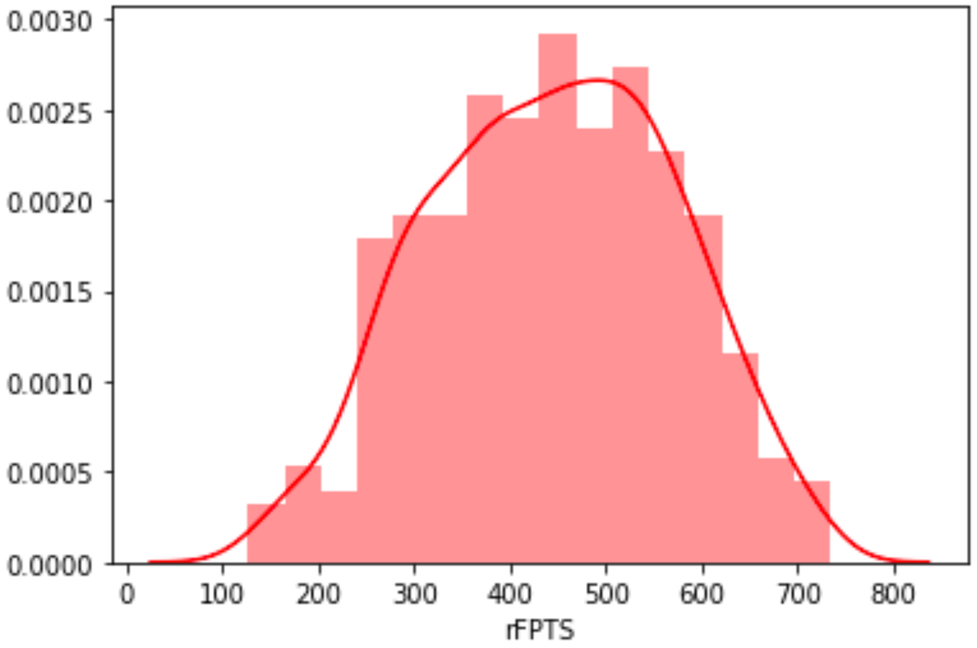
In [ ]:
```
1
```

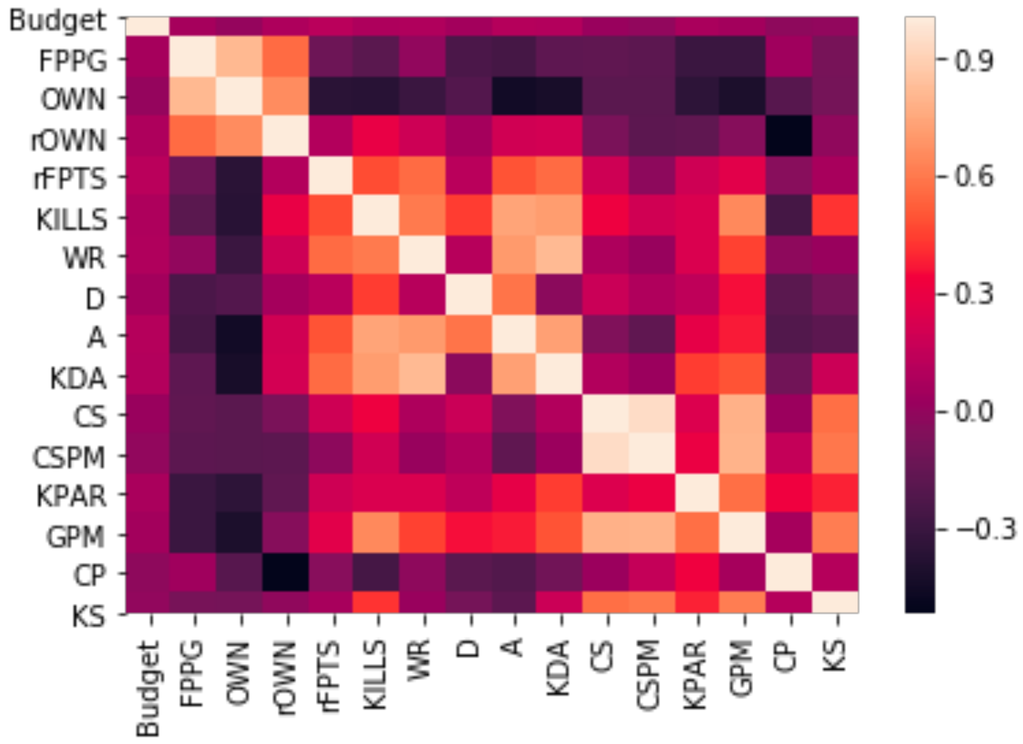
In [6]:
```
1 exporter.export('{date}/lolpoop{date}.csv'.format(date=timestr))
```

April 11 2020

Lineups built from fantasy points

- Optimizer Settings:
- 4- 2 stack….

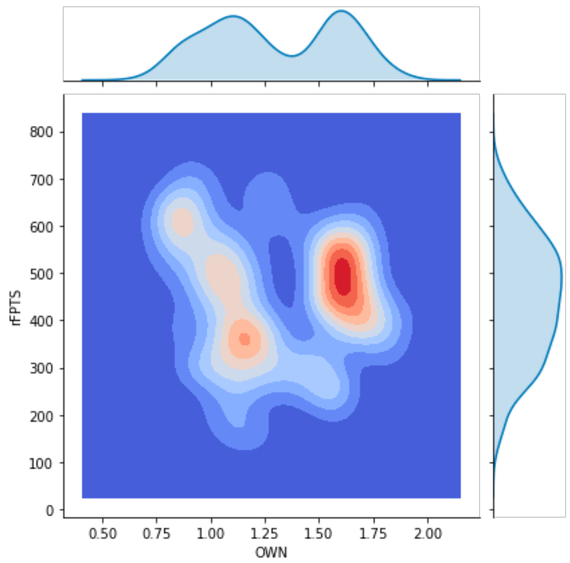




LoL Statistics relationship to lineup fantasy points

Ownership

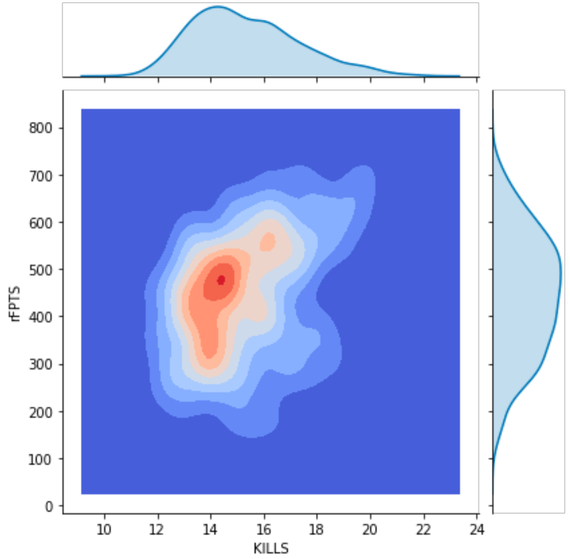
-0.14



There seems to be linear relationship with projected ownership and rFPTS. The two blobs show concentration of lineups. The slope of each blob is the same and the lines can be considered a family of curves

Kills

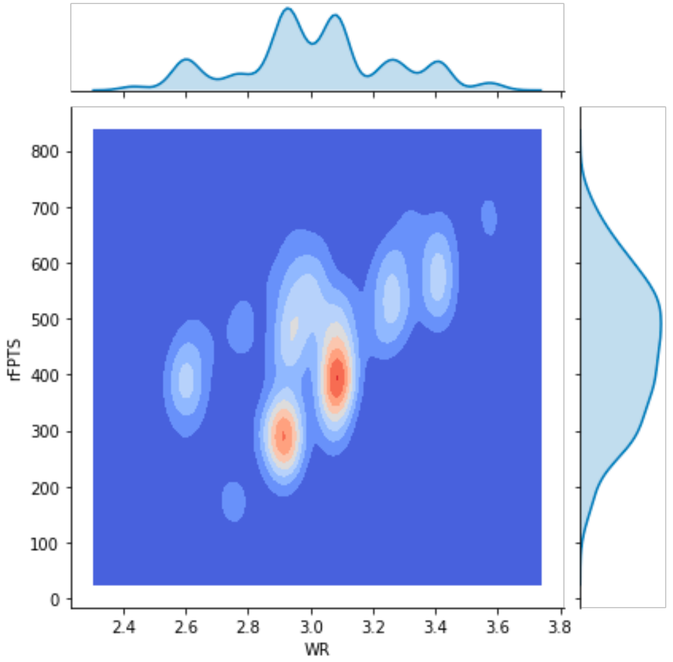
0.38



Lineups with higher Kills will score more fantasy points

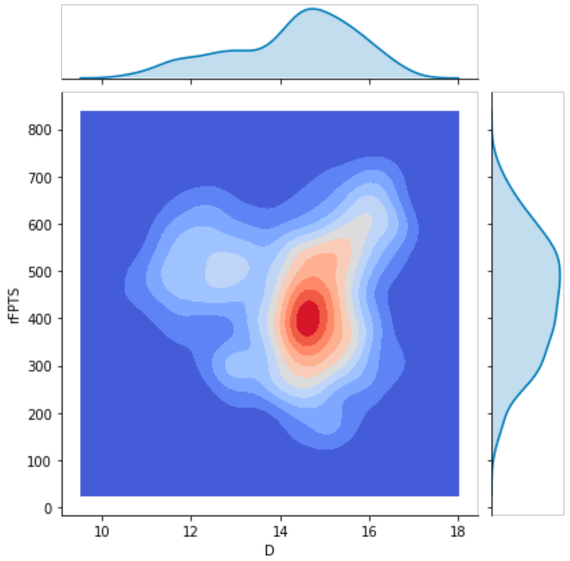
Win Rate

0.58



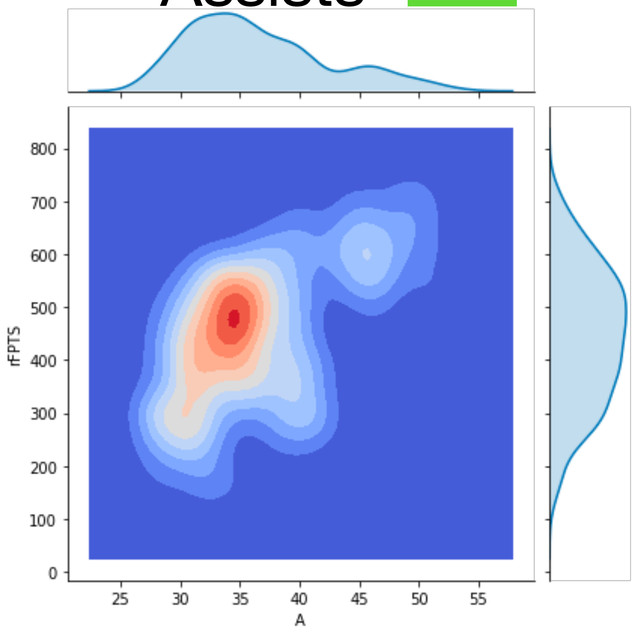
Lineups with higher WR over the set will have score more fantasy points. The local minimums will vary within +/- 0.25

Deaths 0



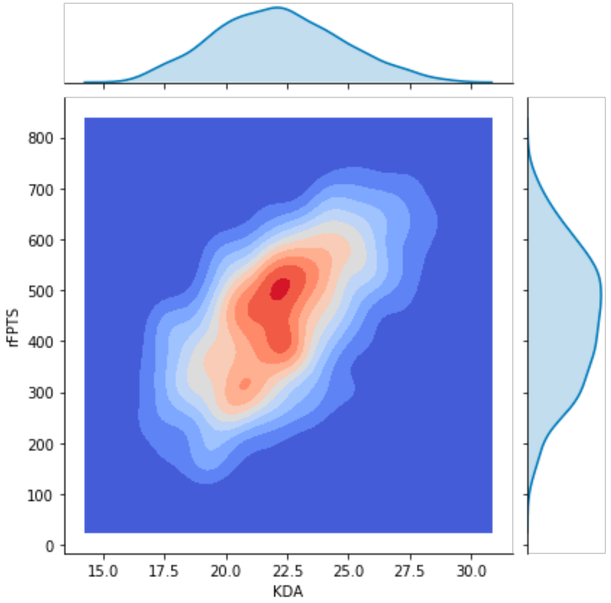
There seems to be no correlation between the number of deaths and fantasy point scored within a lineup

Assists 0.55



There appears to be correlation between the assists and fantasy points scored. You get points for kills and assists and right now I'm stacking all my lineups. The assists could show which teams are better to stack

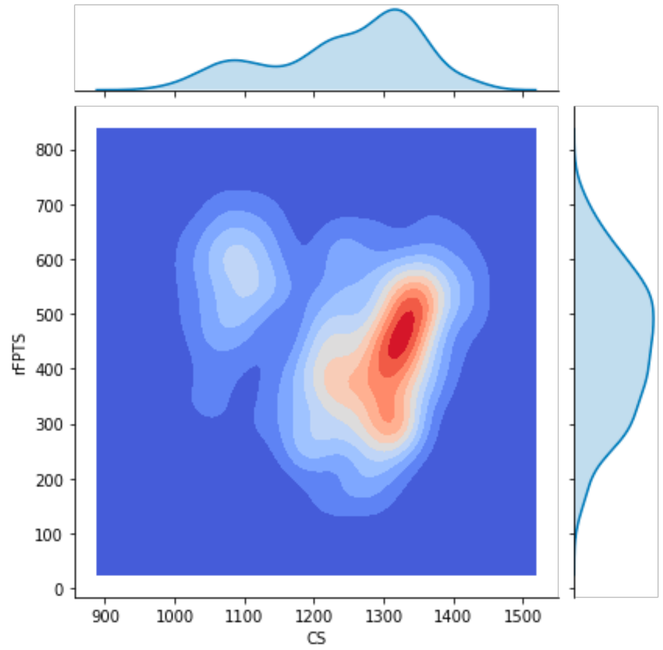
KDA 0.62



There appears to be correlation between KDA and fantasy points. Since kills and assists are involved in scoring this could give us a holistic view of scoring

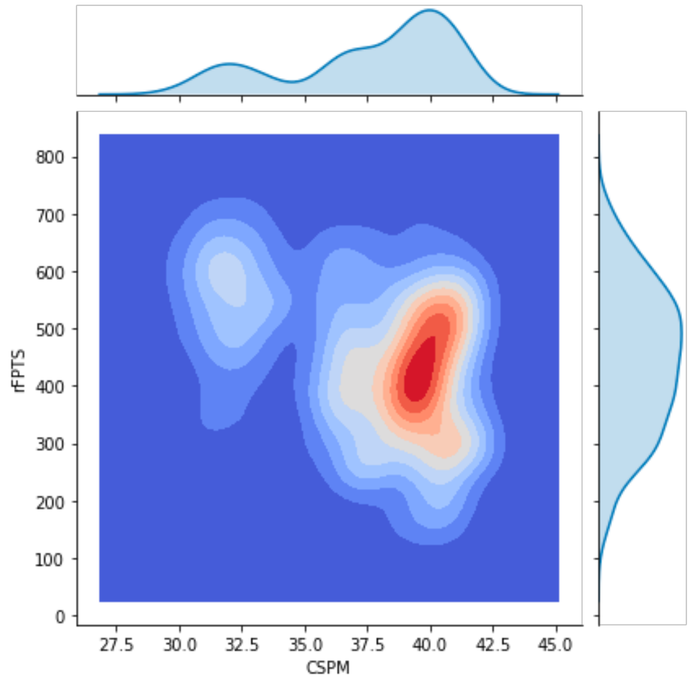
LoL Statistics relationship to lineup fantasy points

Creep Score 0.13



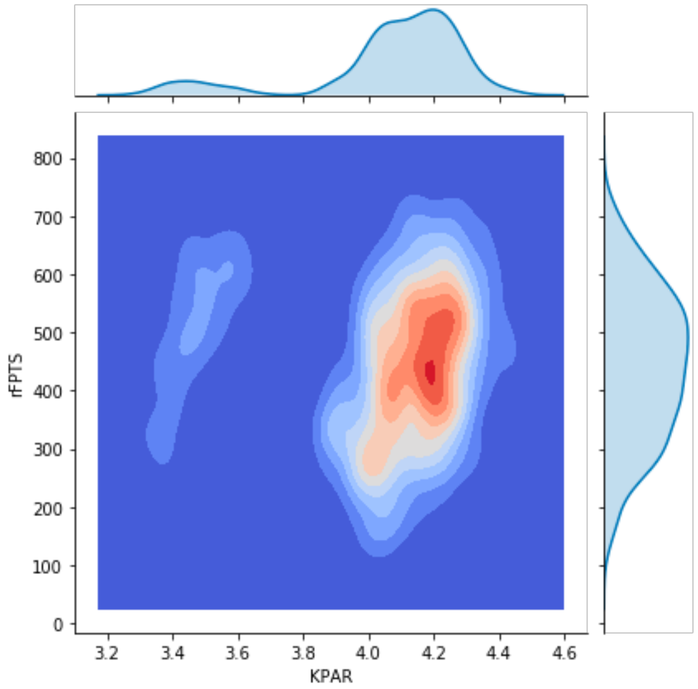
There appears to be no correlation between creep score and fantasy points

Creep Score/ Minute -0.36



There appears to be no correlation between creep score per minute and fantasy points

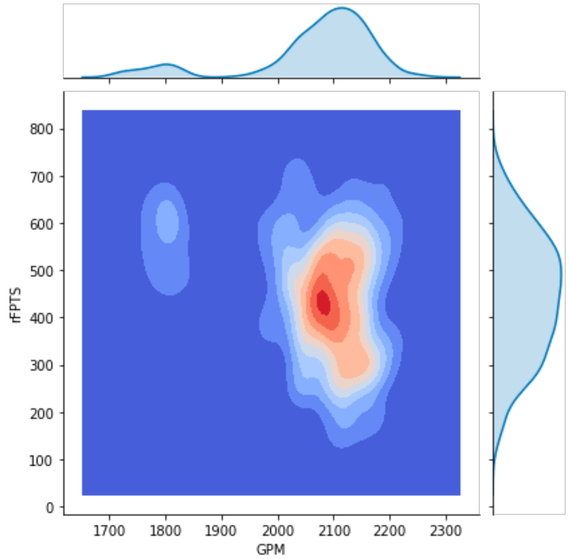
Kill Participation -0.03



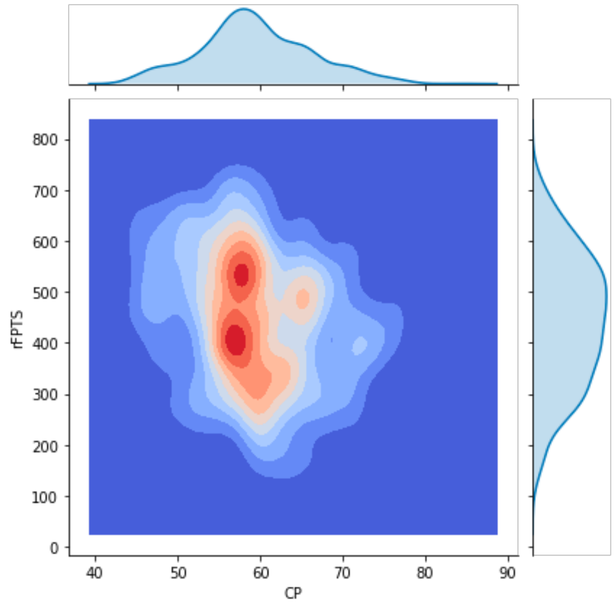
There appears to be no correlation between kill participation and fantasy points

LoL Statistics relationship to lineup fantasy points

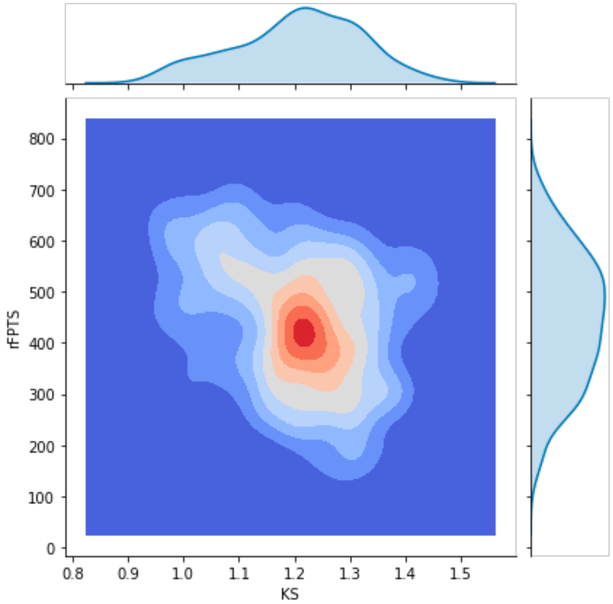
Gold Per Minute -0.18



Champion Points -0.24



Kill Streak -0.27

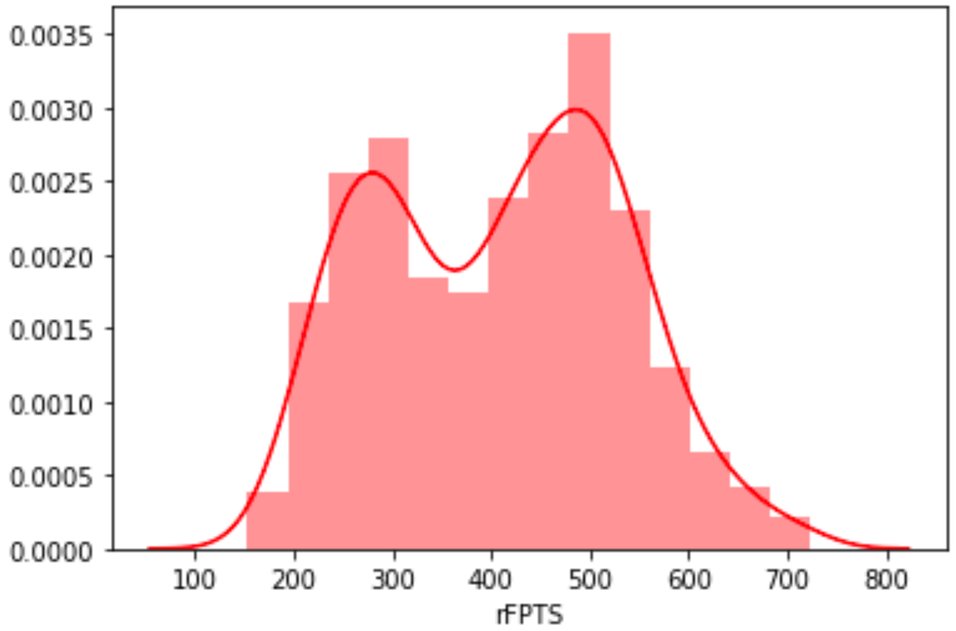


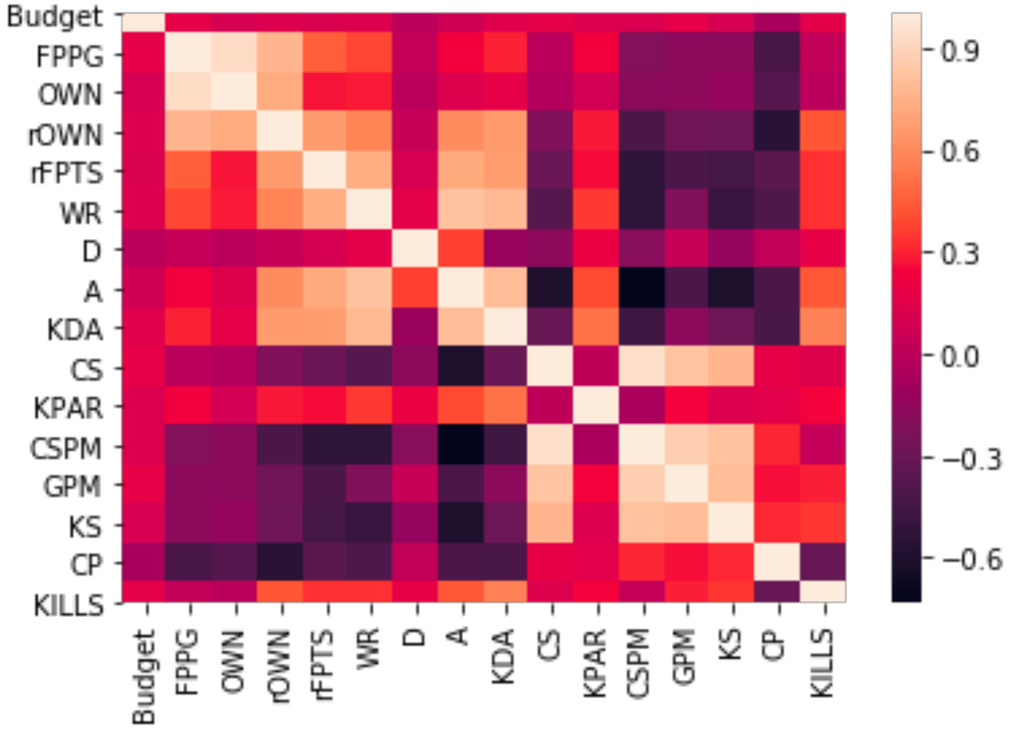
At first glance their isn't an immediate relationship between fantasy points and stat of interest. If we conduct a coordinate transformation- switching from cartesian to polar coordinates- we can observe a cyclical behavior within lineups. This could be useful as a filter after our machine learning algorithm predicts a variable of interest

April 11 2020

Lineups built from fantasy points

- Optimizer Settings:
- 4stack….





Optimizer Settings

In [ ]:
```
1
```

In [93]:
```
1 optimizer = get_optimizer(Site.DRAFTKINGS_CAPTAIN_MODE,Sport.LEAGUE_OF_LEGENDS)
```

In [ ]:
```
1
```

In [94]:
```
1  optimizer.load_players_from_csv('{date}/lolclean{date}.csv'.format(date=timestr))
2  optimizer.set_deviation(0.05, 0.1)
3  #optimizer.set_max_repeating_players(4)
4  optimizer.add_stack(TeamStack(4, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'],
5                                max_exposure=0.5))
6  #optimizer.add_stack(TeamStack(2, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP']))
7  #optimizer.add_stack(TeamStack(3, max_exposure=0.5, max_exposure_per_team={'MIA': 0.6}))  # stack 3 players from sa
8  optimizer.set_min_salary_cap(49400)
9  #optimizer.restrict_positions_for_opposing_team(['TEAM'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
10 #optimizer.restrict_positions_for_opposing_team(['CPT'], ['TEAM', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
11 #optimizer.restrict_positions_for_opposing_team(['TOP'], ['CPT', 'TEAM', 'JNG', 'MID', 'ADC','SUP'])
12 #optimizer.restrict_positions_for_opposing_team(['JNG'], ['CPT', 'TOP', 'TEAM', 'MID', 'ADC','SUP'])
13 #optimizer.restrict_positions_for_opposing_team(['MID'], ['CPT', 'TOP', 'JNG', 'TEAM', 'ADC','SUP'])
14 #optimizer.restrict_positions_for_opposing_team(['ADC'], ['CPT', 'TOP', 'JNG', 'MID', 'TEAM','SUP'])
15 #optimizer.restrict_positions_for_opposing_team(['SUP'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','TEAM'])
16 exporter = CSVLineupExporter(optimizer.optimize(1000, randomness = True))
```

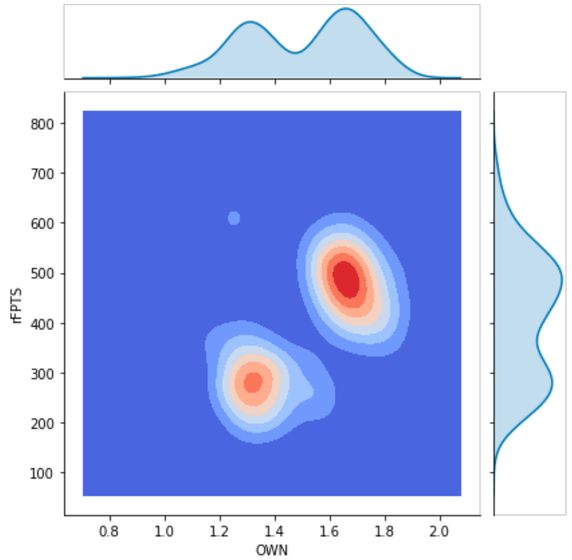
In [ ]:
```
1
```

In [95]:
```
1 exporter.export('{date}/lolpoop{date}.csv'.format(date=timestr))
```

Eliminated the 3 man stack for a single 4 man stack with a maximum exposure set at 50%

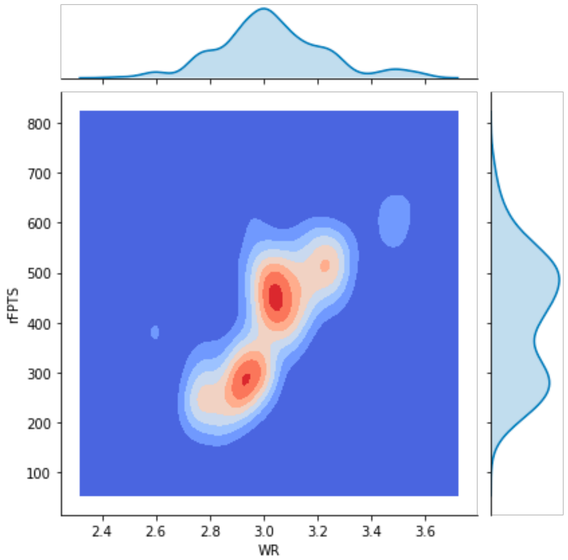
LoL Statistics relationship to lineup fantasy points

Ownership 0.27



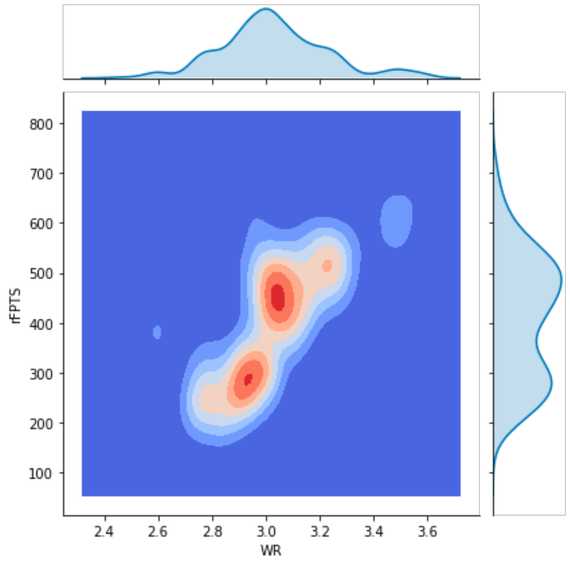
There seems to be linear relationship with projected ownership and rFPTS. The two blobs show concentration of lineups. The slope of each blob is the same and the lines can be considered a family of curves

Kills 0.33



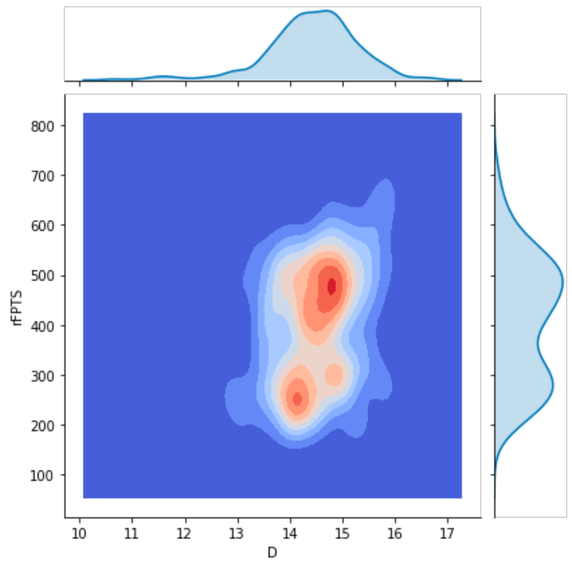
Lineups with higher Kills will score more fantasy points

Win Rate 0.73



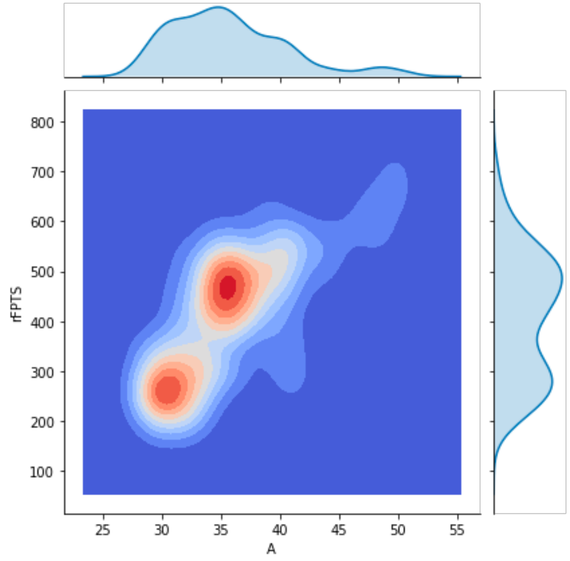
Lineups with high win rates will score more fantasy points.

Deaths 0.1



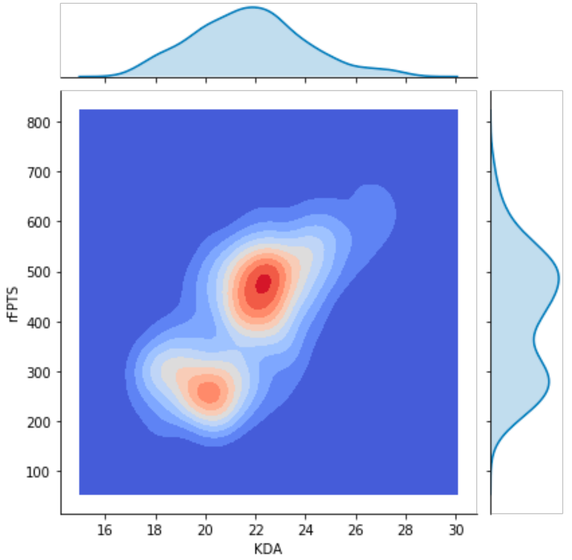
There seems to be no correlation between the number of deaths and fantasy point scored within a lineup

Assists 0.72



There appears to be correlation between the assists and fantasy points scored. You get points for kills and assists and right now I'm stacking all my lineups. The assists could show which teams are better to stack

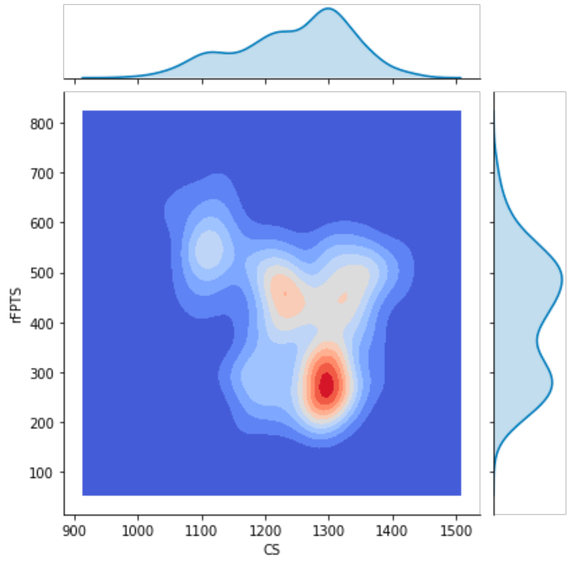
KDA 0.67



There appears to be correlation between KDA and fantasy points. Since kills and assists are involved in scoring this could give us a holistic view of scoring

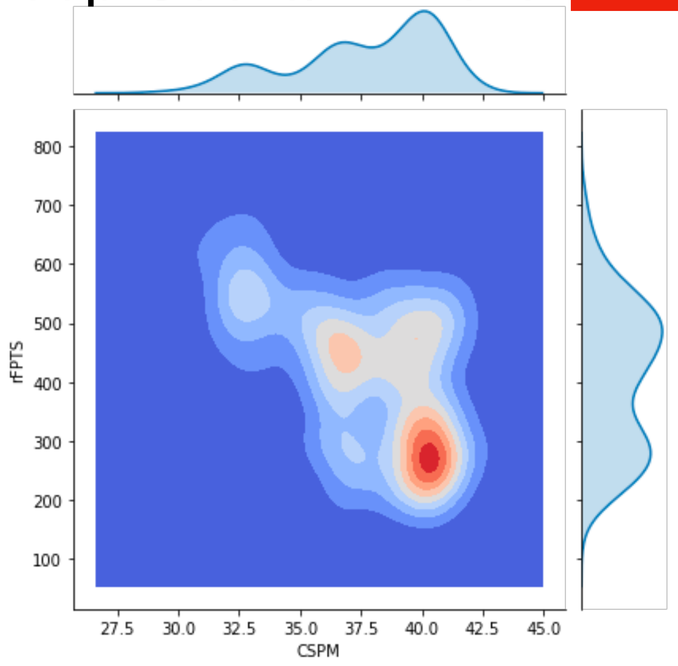
LoL Statistics relationship to lineup fantasy points

Creep Score -0.29



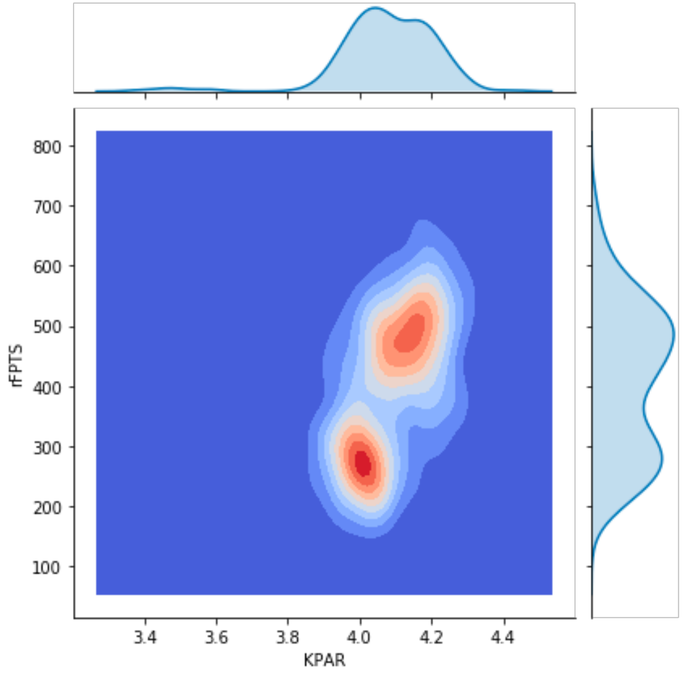
There appears to be no correlation between creep score and fantasy points

Creep Score/ Minute -0.54



There appears to be no correlation between creep score per minute and fantasy points

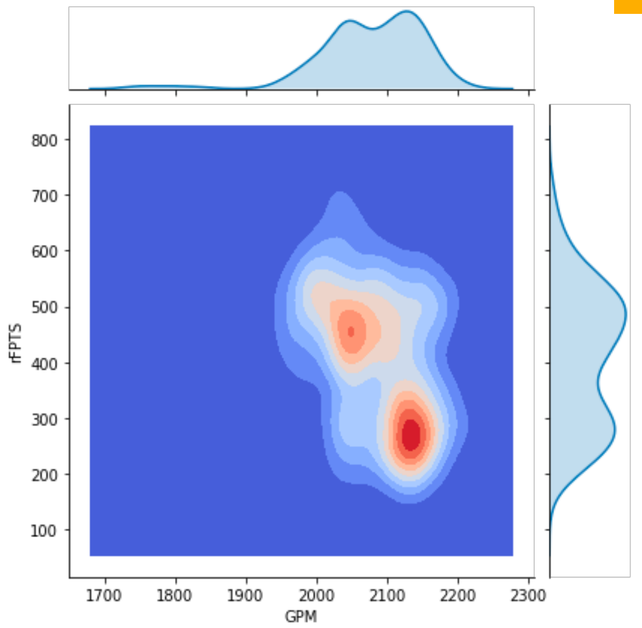
Kill Participation 0.26



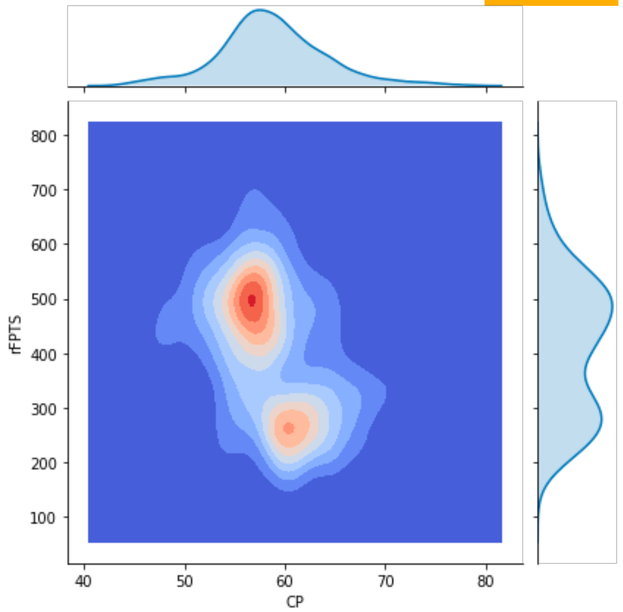
There appears to be no correlation between kill participation and fantasy points

LoL Statistics relationship to lineup fantasy points

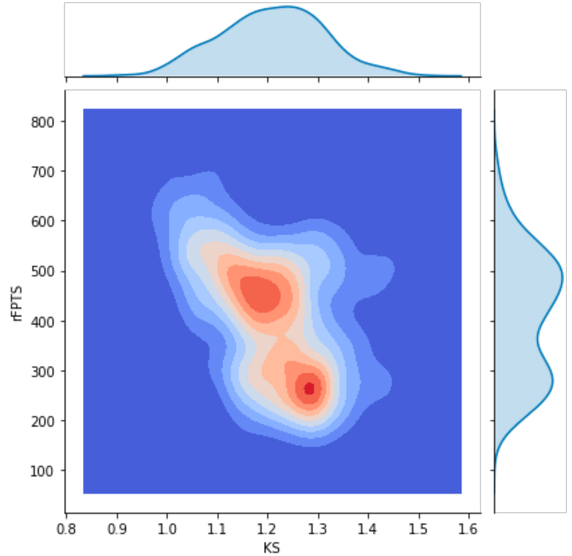
Gold Per Minute -0.42



Champion Points -0.36



Kill Streak -0.45



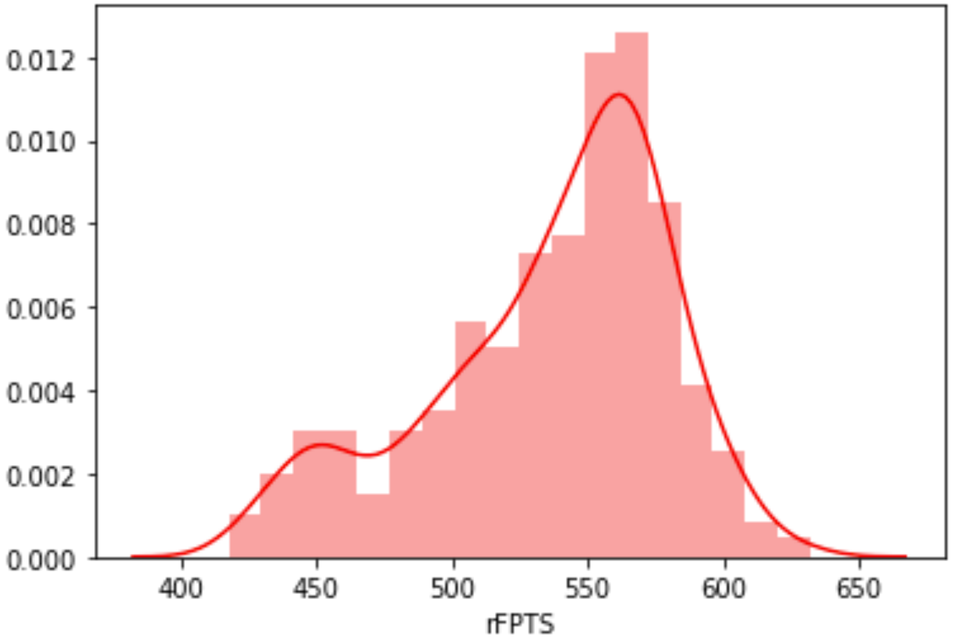
At first glance their isn't an immediate relationship between fantasy points and stat of interest. If we conduct a coordinate transformation- switching from cartesian to polar coordinates- we can observe a cyclical behavior within lineups. This could be useful as a filter after our machine learning algorithm predicts a variable of interest

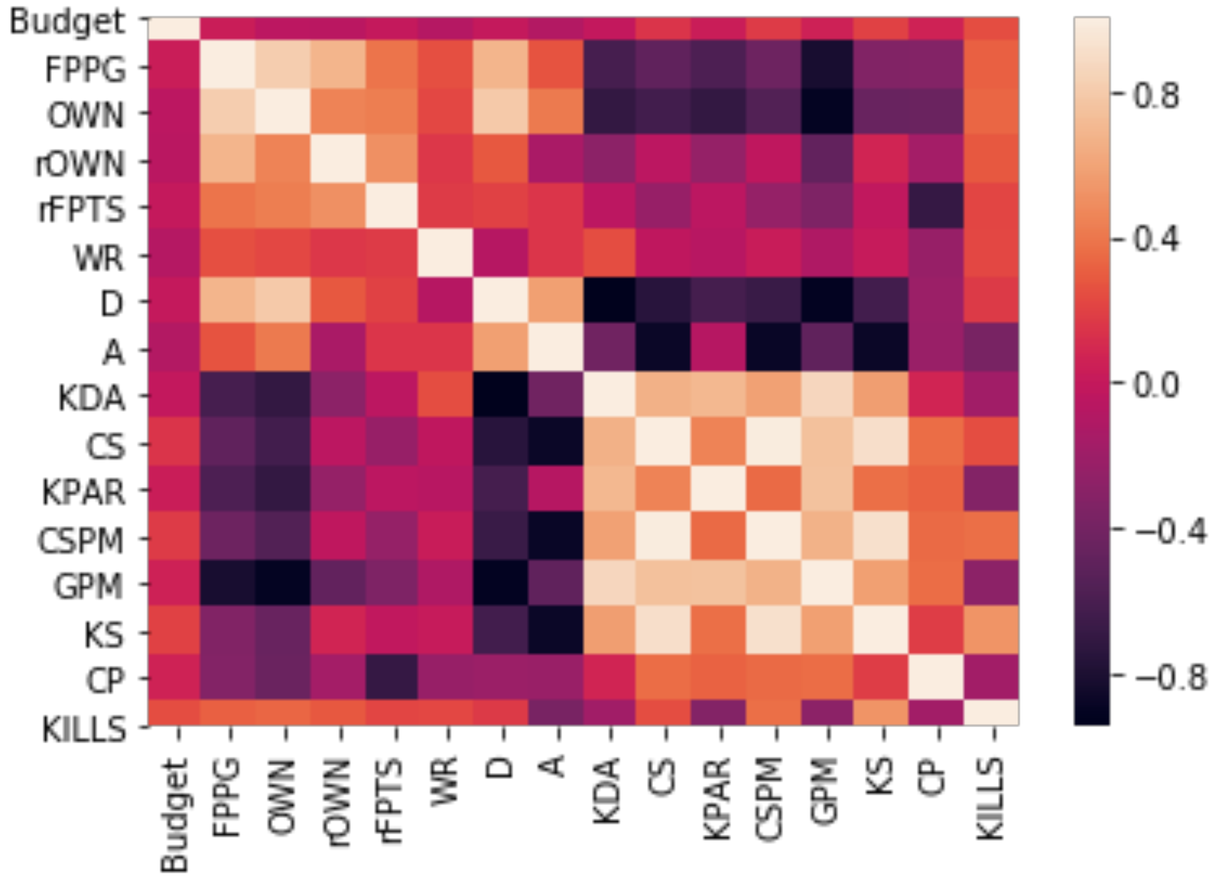
4/ 12 / 20

April 12 2020

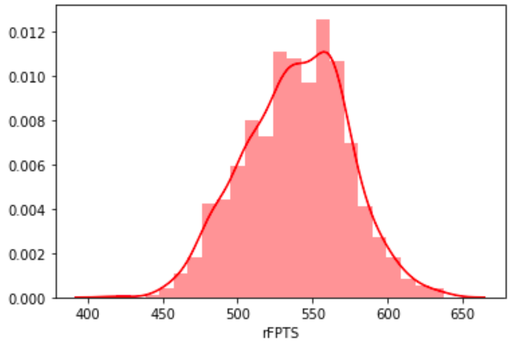
Lineups built from fantasy points

- Optimizer Settings:
- 4 man stack, $45100 min salary

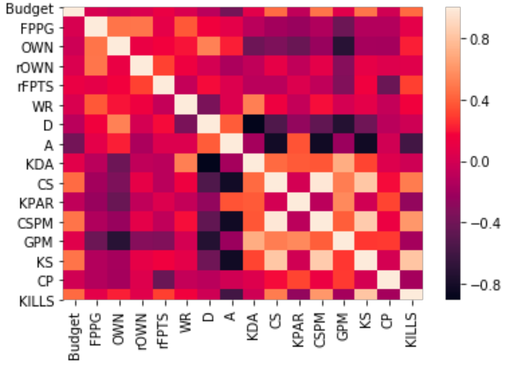


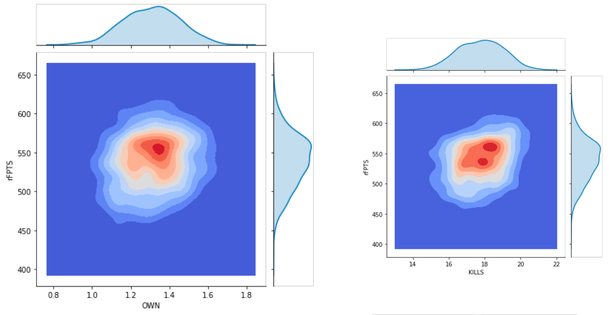


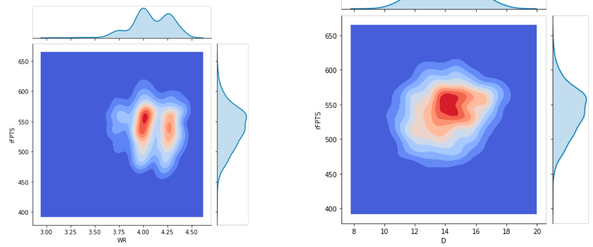
This was for an eSports LoL contest

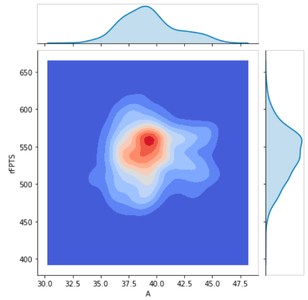


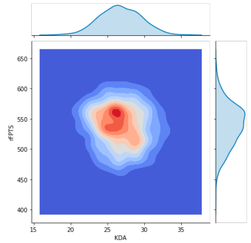

```
CPT       huanfeng (14480604)
TOP            bin (14480477)
JNG       Xiaopeng (14480511)
MID           xiye (14480491)
ADC       huanfeng (14480516)
SUP          XinMo (14480505)
TEAM       Suning (14480530)
Budget          50000
FPPG           491.01
OWN              1.73
rOWN             1.55
rFPTS           637.1
WR              4.459
D               18.61
A               46.32
KDA             35.59
CS            1463.34
KPAR            4.247
CSPM            44.28
GPM              2278
KS              1.435
CP                 83
KILLS           21.03
dtype: object
```

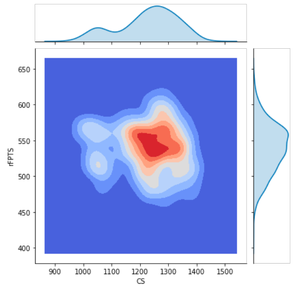


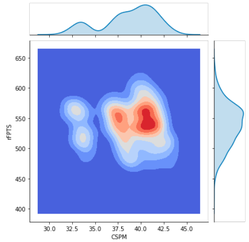


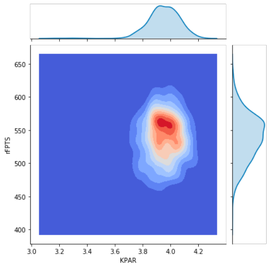


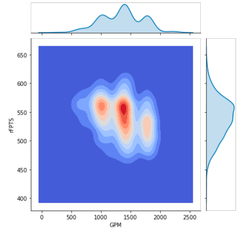


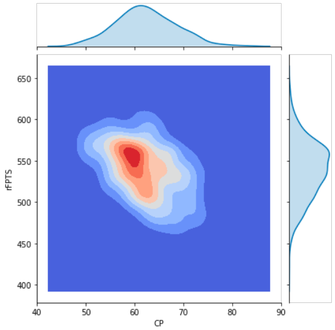


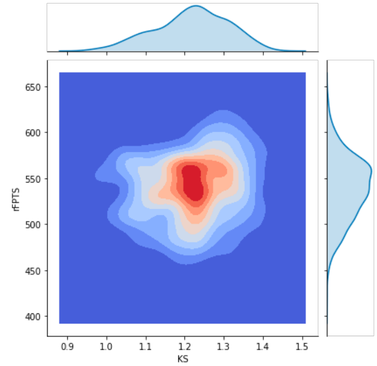












	Budget	FPPG	OWN	rOWN	rFPTS	WR	D	A	KDA	CS	KPAR
Budget	1.000000	0.018722	-0.020723	0.034541	0.114467	0.031353	-0.101480	-0.389336	0.073924	0.445814	-0.075167
FPPG	0.018722	1.000000	0.476559	0.484377	0.090921	0.377823	0.160042	0.083705	-0.102798	-0.193792	-0.235344
OWN	-0.020723	0.476559	1.000000	0.117878	0.167575	0.198741	0.531091	0.219168	-0.407277	-0.340428	-0.432280
rOWN	0.034541	0.484377	0.117878	1.000000	0.315137	0.143219	0.003134	-0.160727	-0.075567	0.094586	-0.074866
rFPTS	0.114467	0.090921	0.167575	0.315137	1.000000	-0.056961	0.185014	0.044345	-0.104959	-0.106472	0.045936
WR	0.031353	0.377823	0.198741	0.143219	-0.056961	1.000000	-0.360576	0.041800	0.523208	0.141141	-0.061765
D	-0.101480	0.160042	0.531091	0.003134	0.185014	-0.360576	1.000000	0.394520	-0.905019	-0.536329	-0.254656
A	-0.389336	0.083705	0.219168	-0.160727	0.044345	0.041800	0.394520	1.000000	-0.189726	-0.816358	0.360296
KDA	0.073924	-0.102798	-0.407277	-0.075567	-0.104959	0.523208	-0.905019	-0.189726	1.000000	0.438980	0.386493
CS	0.445814	-0.193792	-0.340428	0.094586	-0.106472	0.141141	-0.536329	-0.816358	0.438980	1.000000	0.001023
KPAR	-0.075167	-0.235344	-0.432280	-0.074866	0.045936	-0.061765	-0.254656	0.360296	0.386493	0.001023	1.000000
CSPM	0.474595	-0.145554	-0.242223	0.083685	-0.078620	0.190983	-0.459660	-0.827046	0.369028	0.983669	-0.098776
GPM	0.064644	-0.415559	-0.713064	-0.304919	-0.331133	0.007990	-0.735616	-0.226472	0.709373	0.510774	0.557986
KS	0.477768	-0.126460	-0.180089	0.093384	0.154515	0.083591	-0.400783	-0.827009	0.320931	0.823950	-0.010867
CP	-0.005836	-0.126792	-0.189550	0.048627	-0.422451	-0.057917	-0.103291	-0.007512	0.054108	0.188275	0.321959
KILLS	0.440451	0.078200	0.225249	0.069545	0.305584	0.159350	-0.016931	-0.595926	-0.019459	0.504991	-0.250658

Optimizer Settings

In [ ]:
```
1
```

In [93]:
```
1 optimizer = get_optimizer(Site.DRAFTKINGS_CAPTAIN_MODE,Sport.LEAGUE_OF_LEGENDS)
```

In [ ]:
```
1
```

In [94]:
```
1  optimizer.load_players_from_csv('{date}/lolclean{date}.csv'.format(date=timestr))
2  optimizer.set_deviation(0.05, 0.1)
3  #optimizer.set_max_repeating_players(4)
4  optimizer.add_stack(TeamStack(4, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'],
5                                max_exposure=0.5))
6  #optimizer.add_stack(TeamStack(2, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP']))
7  #optimizer.add_stack(TeamStack(3, max_exposure=0.5, max_exposure_per_team={'MIA': 0.6}))  # stack 3 players from sa
8  optimizer.set_min_salary_cap(49400)
9  #optimizer.restrict_positions_for_opposing_team(['TEAM'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
10 #optimizer.restrict_positions_for_opposing_team(['CPT'], ['TEAM', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
11 #optimizer.restrict_positions_for_opposing_team(['TOP'], ['CPT', 'TEAM', 'JNG', 'MID', 'ADC','SUP'])
12 #optimizer.restrict_positions_for_opposing_team(['JNG'], ['CPT', 'TOP', 'TEAM', 'MID', 'ADC','SUP'])
13 #optimizer.restrict_positions_for_opposing_team(['MID'], ['CPT', 'TOP', 'JNG', 'TEAM', 'ADC','SUP'])
14 #optimizer.restrict_positions_for_opposing_team(['ADC'], ['CPT', 'TOP', 'JNG', 'MID', 'TEAM','SUP'])
15 #optimizer.restrict_positions_for_opposing_team(['SUP'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','TEAM'])
16 exporter = CSVLineupExporter(optimizer.optimize(1000, randomness = True))
```

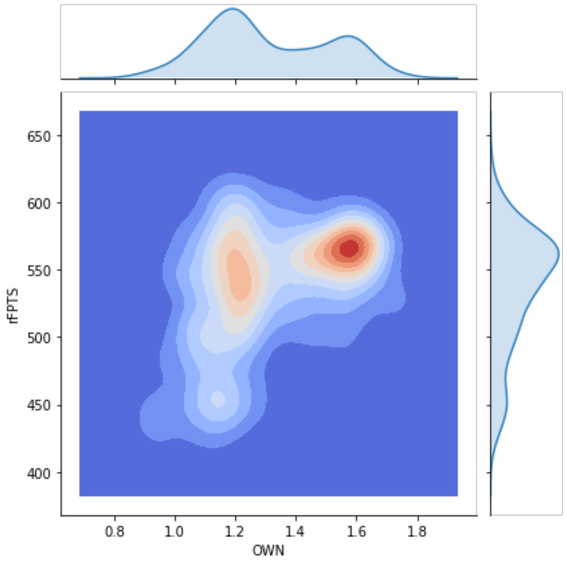
In [ ]:
```
1
```

In [95]:
```
1 exporter.export('{date}/lolpoop{date}.csv'.format(date=timestr))
```

Eliminated the 3 man stack for a single 4 man stack with a maximum exposure set at 50%

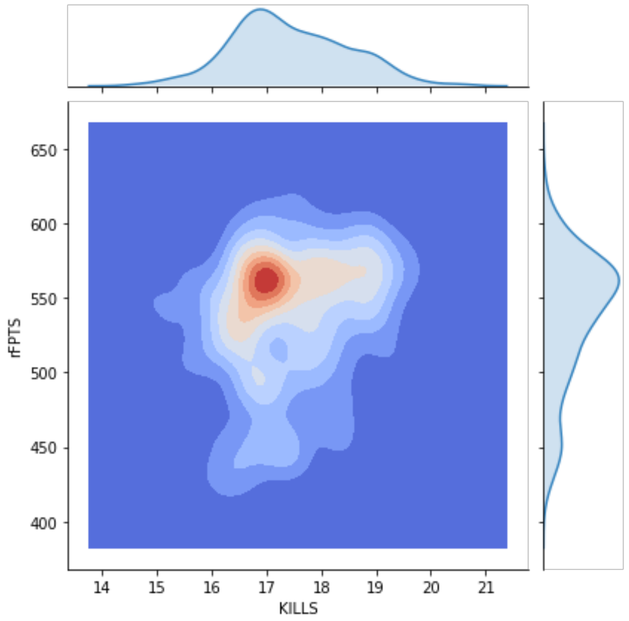
LoL Statistics relationship to lineup fantasy points

Ownership 0.27



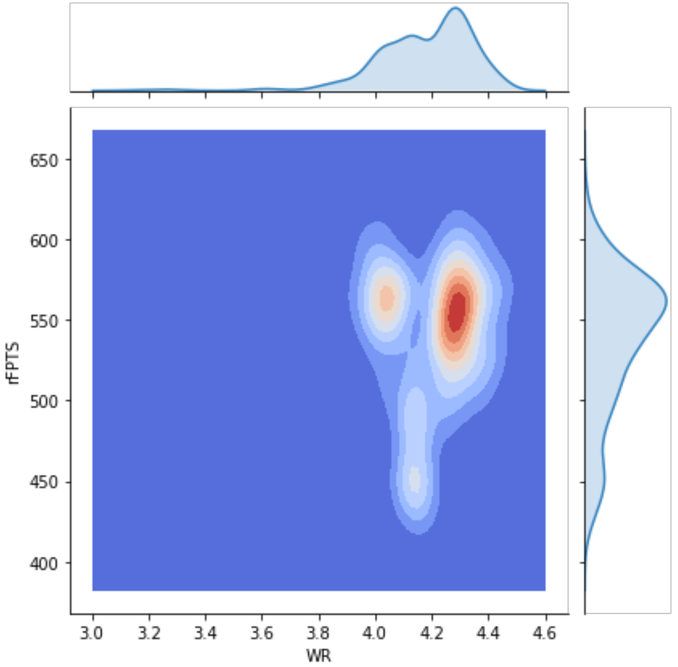
There seems to be linear relationship with projected ownership and rFPTS. The two blobs show concentration of lineups. The slope of each blob is the same and the lines can be considered a family of curves

Kills 0.33



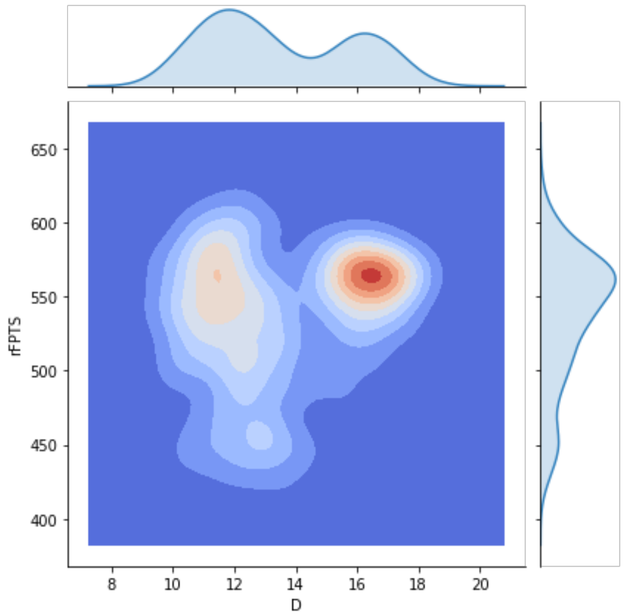
Lineups with higher Kills will score more fantasy points

Win Rate 0.73



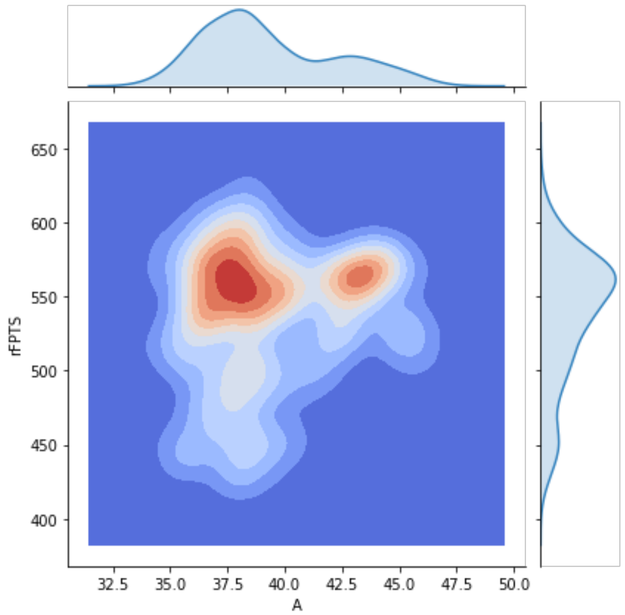
Lineups with higher WR over the set will have score more fantasy points. The local minimums will vary within +/- 0.25

Deaths 0.1



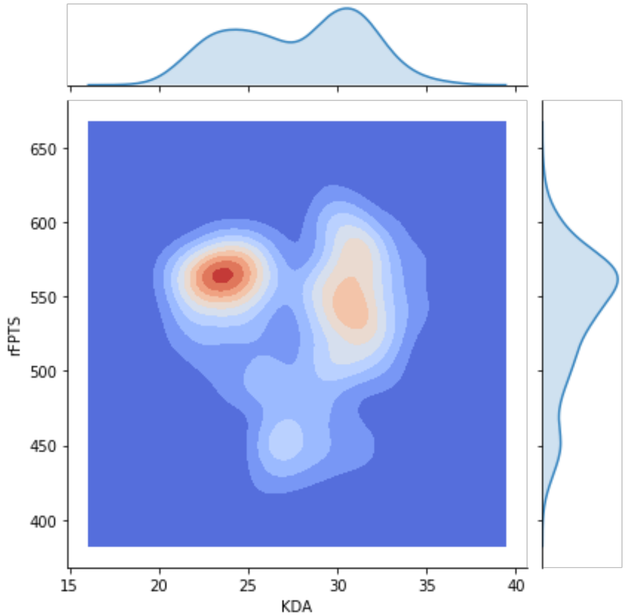
There seems to be no correlation between the number of deaths and fantasy point scored within a lineup

Assists 0.72



There appears to be correlation between the assists and fantasy points scored. You get points for kills and assists and right now I'm stacking all my lineups. The assists could show which teams are better to stack

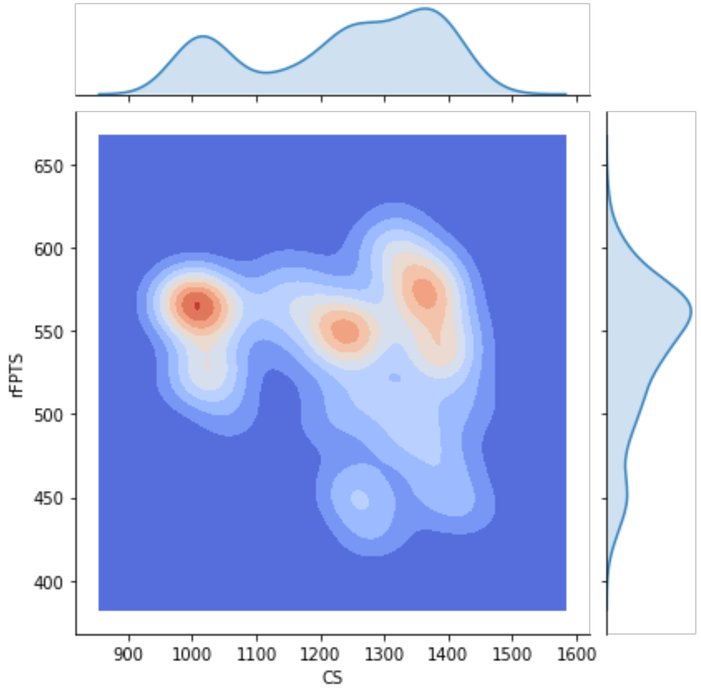
KDA 0.67



There appears to be correlation between KDA and fantasy points.
Since kills and assists are involved in scoring this could give us a holistic view of scoring

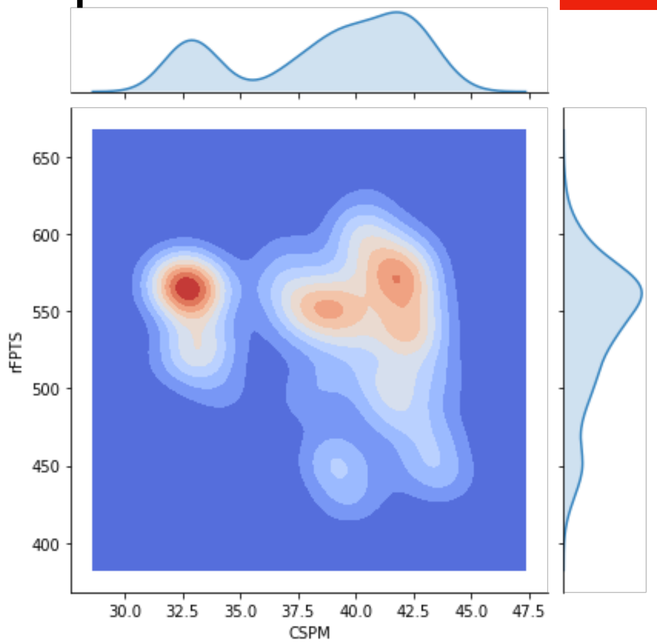
LoL Statistics relationship to lineup fantasy points

Creep Score -0.29



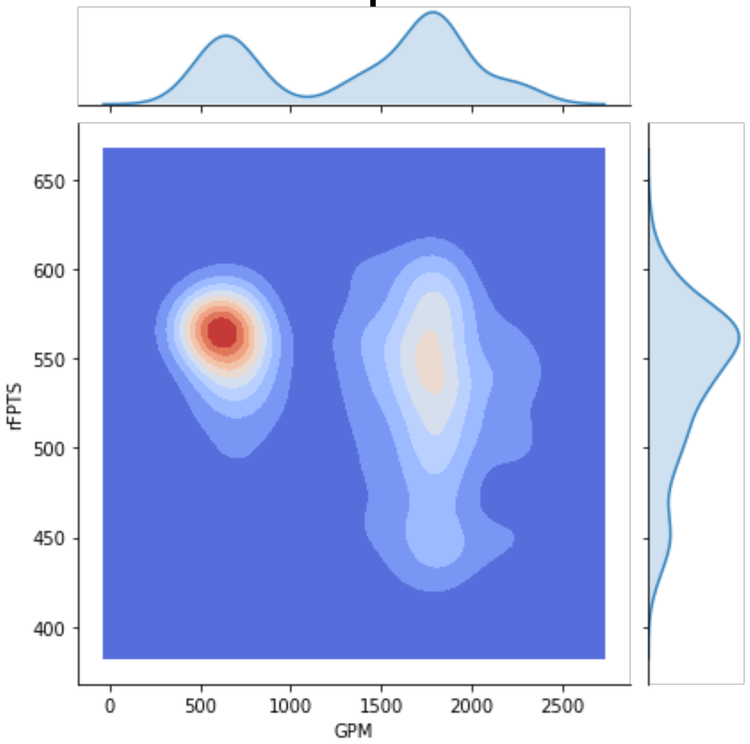
There appears to be no correlation between creep score and fantasy points

Creep Score/ Minute -0.54



There appears to be no correlation between creep score per minute and fantasy points

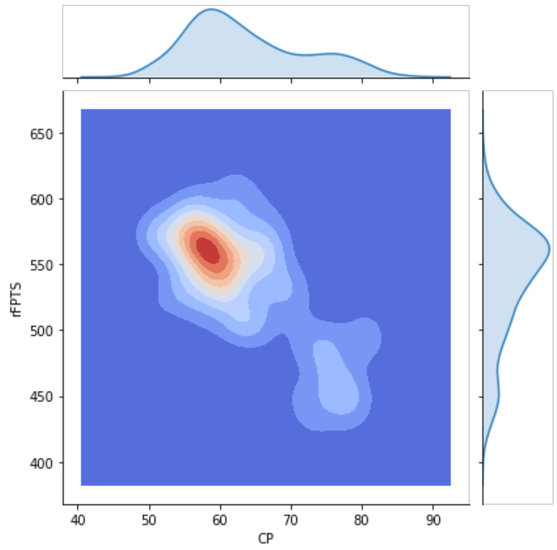
Kill Participation 0.26



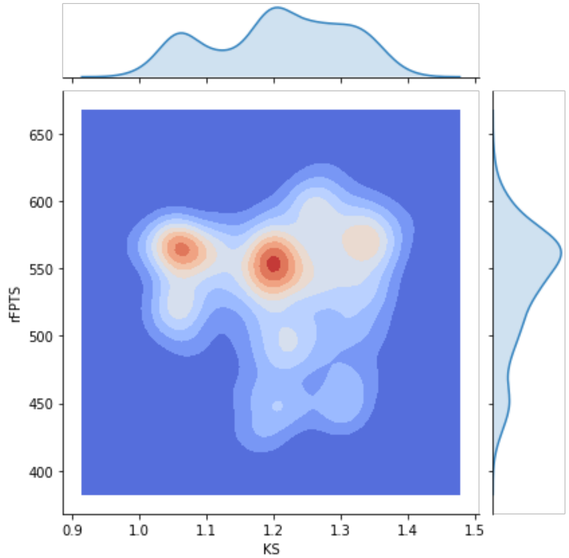
There appears to be no correlation between kill participation and fantasy points

LoL Statistics relationship to lineup fantasy points

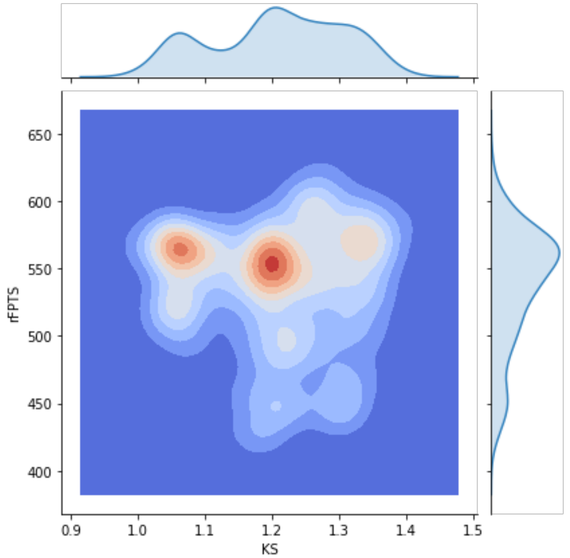
Gold Per Minute -0.42



Champion Points -0.36

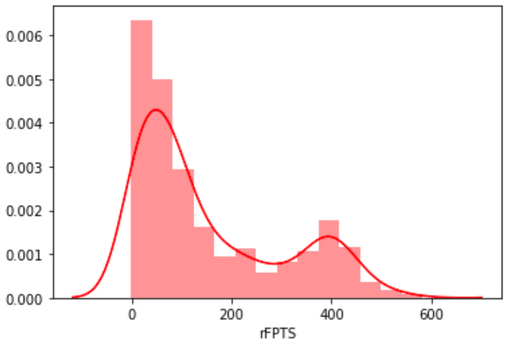


Kill Streak -0.45

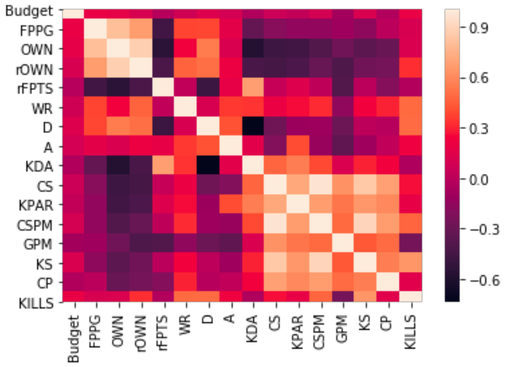


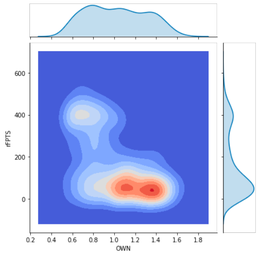
At first glance their isn't an immediate relationship between fantasy points and stat of interest. If we conduct a coordinate transformation- switching from cartesian to polar coordinates- we can observe a cyclical behavior within lineups. This could be useful as a filter after our machine learning algorithm predicts a variable of interest

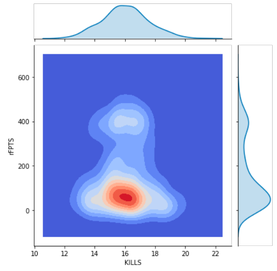
4/ 15 / 20

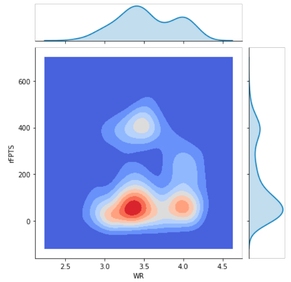


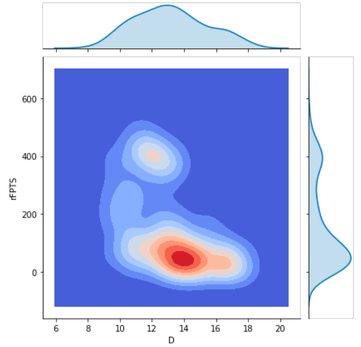

```
CPT        yuekai (14483373) (LOCKED)
TOP      TheShy (14483265) (LOCKED)
JNG        bonO (14483340) (LOCKED)
MID        xiye (14483293) (LOCKED)
ADC      Teddy (14483346) (LOCKED)
SUP       cold (14483319) (LOCKED)
TEAM    Team WE (14483370) (LOCKED)
Budget                        50000
FPPG                         523.26
OWN                           1.673
rOWN                         1.9438
rFPTS                        583.06
WR                            4.326
D                              18.7
A                             45.25
KDA                           38.08
CS                          1499.57
KPAR                          4.356
CSPM                          44.69
GPM                            2280
KS                             1.51
CP                               83
KILLS                         21.29
dtype: object
```

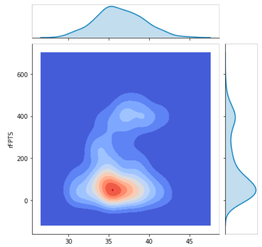


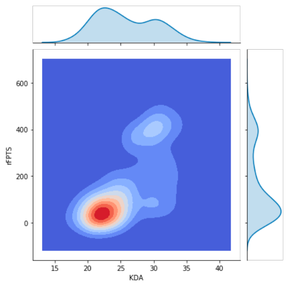


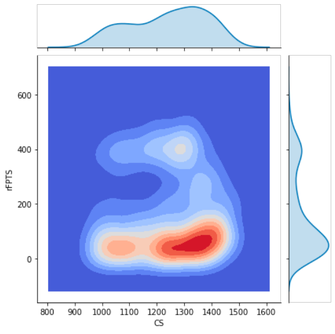


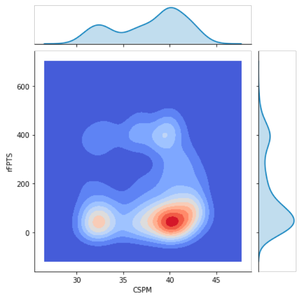


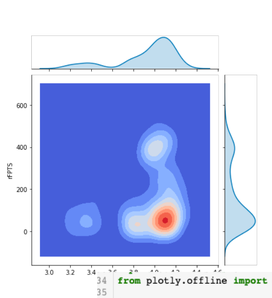


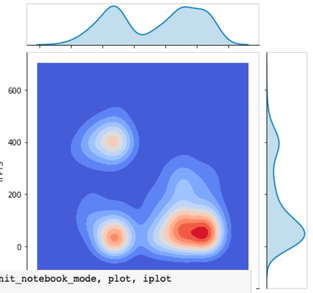


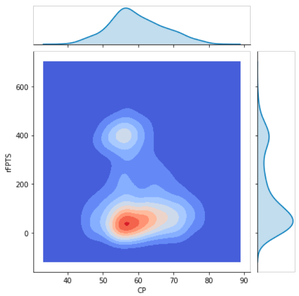


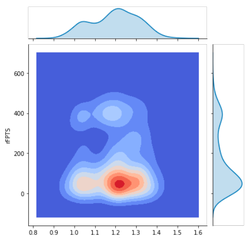













```
34 from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
35
36 init_notebook_mode(connected=True)
37 cf.go_offline()
```

	Budget	FPPG	OWN	rOWN	rFPTS	WR	D	A	KDA	CS	KPAR
Budget	1.000000	0.187622	0.175543	0.112532	-0.009191	0.077585	0.137160	0.090425	-0.040301	0.064060	0.029460
FPPG	0.187622	1.000000	0.809556	0.673368	-0.451470	0.377234	0.385774	0.166318	-0.315612	-0.180623	-0.150421
OWN	0.175543	0.809556	1.000000	0.876933	-0.550577	0.247555	0.546553	0.118966	-0.583020	-0.475234	-0.449945
rOWN	0.112532	0.673368	0.876933	1.000000	-0.420802	0.467285	0.510733	0.209422	-0.424720	-0.448699	-0.409372
rFPTS	-0.009191	-0.451470	-0.550577	-0.420802	1.000000	0.022864	-0.488549	0.184678	0.677021	0.040741	0.139784
WR	0.077585	0.377234	0.247555	0.467285	0.022864	1.000000	0.106775	0.351286	0.337786	0.203131	0.255254
D	0.137160	0.385774	0.546553	0.510733	-0.488549	0.106775	1.000000	0.415281	-0.733471	-0.277237	-0.113479
A	0.090425	0.166318	0.118966	0.209422	0.184678	0.351286	0.415281	1.000000	0.154844	-0.202854	0.398036
KDA	-0.040301	-0.315612	-0.583020	-0.424720	0.677021	0.337786	-0.733471	0.154844	1.000000	0.472679	0.559603
CS	0.064060	-0.180623	-0.475234	-0.448699	0.040741	0.203131	-0.277237	-0.202854	0.472679	1.000000	0.710679
KPAR	0.029460	-0.150421	-0.449945	-0.409372	0.139784	0.255254	-0.113479	0.398036	0.559603	0.710679	1.000000
CSPM	0.095614	-0.139666	-0.392090	-0.312368	0.013799	0.321911	-0.108959	-0.127950	0.395178	0.966620	0.668784
GPM	-0.087076	-0.091280	-0.271927	-0.396274	-0.394770	-0.155591	-0.286960	-0.334954	0.123003	0.631788	0.532522
KS	0.116926	-0.162287	-0.348979	-0.267436	-0.006342	0.247086	0.006252	-0.097842	0.300465	0.847103	0.650305
CP	-0.037998	-0.000155	-0.304616	-0.257682	-0.192821	0.298388	-0.013582	0.021877	0.239786	0.682169	0.595753
KILLS	0.199776	0.110115	0.125781	0.323394	-0.034133	0.486123	0.495170	0.223015	-0.048176	0.267423	0.184041

```
1
```

```
1 optimizer = get_optimizer(Site.DRAFTKINGS_CAPTAIN_MODE,Sport.LEAGUE_OF_LEGENDS)
```

```
1
```

```
1 optimizer.load_players_from_csv('{date}/lolclean{date}.csv'.format(date=timestr))
2 optimizer.set_deviation(0.05, 0.1)
3 optimizer.set_max_repeating_players(4)
4 optimizer.add_stack(TeamStack(4, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'],
5                               max_exposure=0.35))
6 optimizer.set_min_salary_cap(49400)
7 exporter = CSVLineupExporter(optimizer.optimize(1000, randomness = True))
```

```
1
```

April 15 2020

Lineups built from fantasy points

- Optimizer Settings:
- 4 stack….

Optimizer Settings

In [ ]:
```
1
```

In [93]:
```
1 optimizer = get_optimizer(Site.DRAFTKINGS_CAPTAIN_MODE,Sport.LEAGUE_OF_LEGENDS)
```

In [ ]:
```
1
```

In [94]:
```
1  optimizer.load_players_from_csv('{date}/lolclean{date}.csv'.format(date=timestr))
2  optimizer.set_deviation(0.05, 0.1)
3  #optimizer.set_max_repeating_players(4)
4  optimizer.add_stack(TeamStack(4, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'],
5                                max_exposure=0.5))
6  #optimizer.add_stack(TeamStack(2, for_positions= ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP']))
7  #optimizer.add_stack(TeamStack(3, max_exposure=0.5, max_exposure_per_team={'MIA': 0.6}))  # stack 3 players from sa
8  optimizer.set_min_salary_cap(49400)
9  #optimizer.restrict_positions_for_opposing_team(['TEAM'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
10 #optimizer.restrict_positions_for_opposing_team(['CPT'], ['TEAM', 'TOP', 'JNG', 'MID', 'ADC','SUP'])
11 #optimizer.restrict_positions_for_opposing_team(['TOP'], ['CPT', 'TEAM', 'JNG', 'MID', 'ADC','SUP'])
12 #optimizer.restrict_positions_for_opposing_team(['JNG'], ['CPT', 'TOP', 'TEAM', 'MID', 'ADC','SUP'])
13 #optimizer.restrict_positions_for_opposing_team(['MID'], ['CPT', 'TOP', 'JNG', 'TEAM', 'ADC','SUP'])
14 #optimizer.restrict_positions_for_opposing_team(['ADC'], ['CPT', 'TOP', 'JNG', 'MID', 'TEAM','SUP'])
15 #optimizer.restrict_positions_for_opposing_team(['SUP'], ['CPT', 'TOP', 'JNG', 'MID', 'ADC','TEAM'])
16 exporter = CSVLineupExporter(optimizer.optimize(1000, randomness = True))
```

In [ ]:
```
1
```

In [95]:
```
1 exporter.export('{date}/lolpoop{date}.csv'.format(date=timestr))
```

Eliminated the 3 man stack for a single 4 man stack with a maximum exposure set at 50%

LoL Statistics relationship to lineup fantasy points

Ownership 0.27

There seems to be linear relationship with projected ownership and rFPTS. The two blobs show concentration of lineups. The slope of each blob is the same and the lines can be considered a family of curves

Kills 0.33

Lineups with higher Kills will score more fantasy points

Win Rate 0.73

Lineups with higher WR over the set will have score more fantasy points. The local minimums will vary within +/- 0.25

Deaths 0.1

There seems to be no correlation between the number of deaths and fantasy point scored within a lineup

Assists 0.72

There appears to be correlation between the assists and fantasy points scored. You get points for kills and assists and right now I'm stacking all my lineups. The assists could show which teams are better to stack

KDA 0.67

There appears to be correlation between KDA and fantasy points. Since kills and assists are involved in scoring this could give us a holistic view of scoring

LoL Statistics relationship to lineup fantasy points

Creep Score **-0.29**

There appears to be no correlation between creep score and fantasy points

Creep Score/ Minute **-0.54**

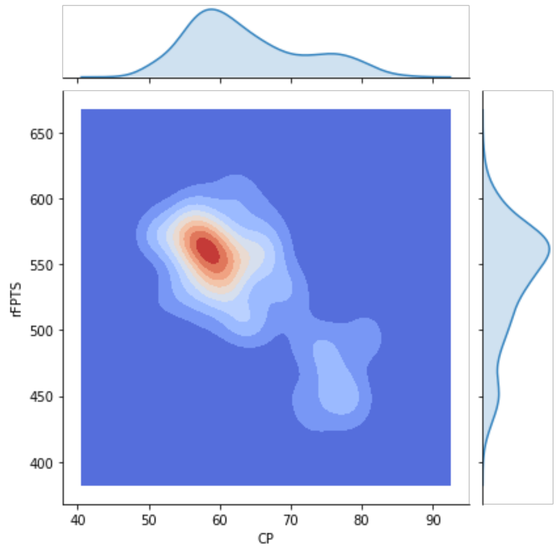
There appears to be no correlation between creep score per minute and fantasy points

Kill Participation **0.26**

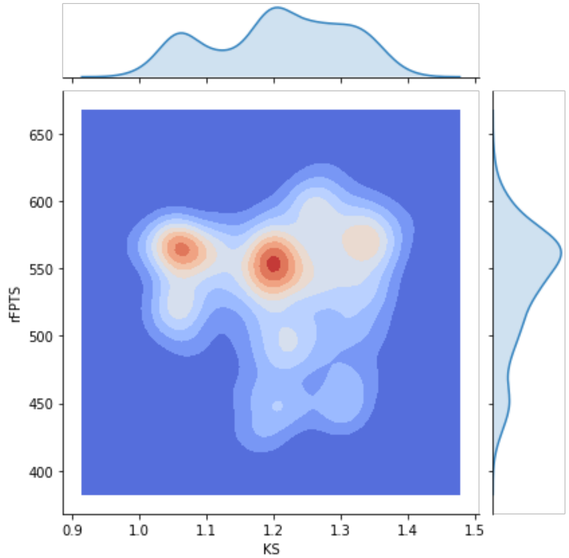
There appears to be no correlation between kill participation and fantasy points

LoL Statistics relationship to lineup fantasy points

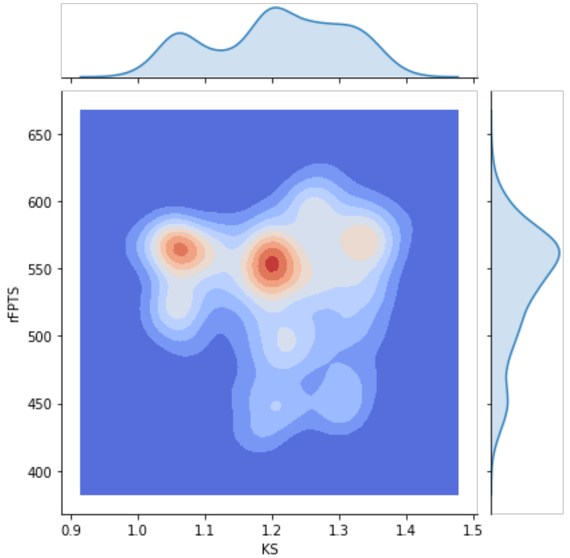
Gold Per Minute -0.42



Champion Points -0.36



Kill Streak -0.45

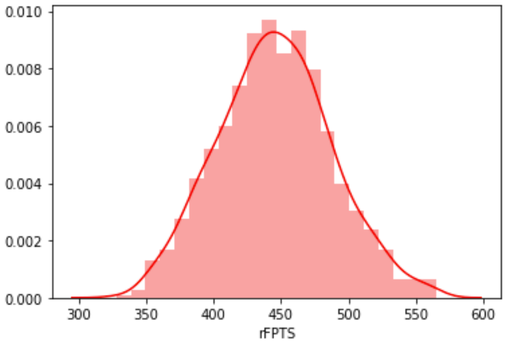


At first glance their isn't an immediate relationship between fantasy points and stat of interest. If we conduct a coordinate transformation- switching from cartesian to polar coordinates- we can observe a cyclical behavior within lineups. This could be useful as a filter after our machine learning algorithm predicts a variable of interest

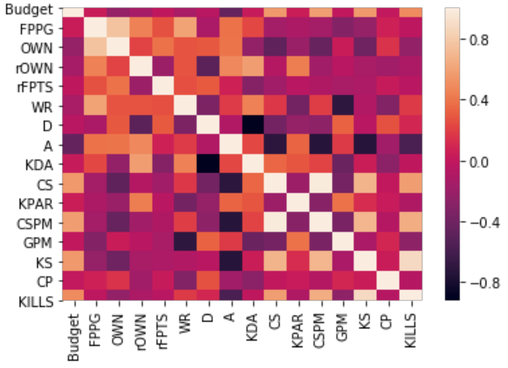
Table 1

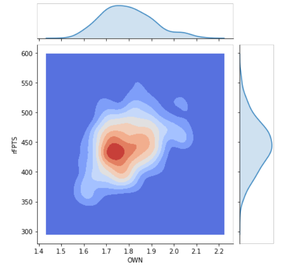
Date	Score	Salary	Winner	Min Score
110420	828.57	49500	Nty9problems	611.99
120420	637.68	50000	Awesemo-tie	561
150420	609.02	50000	Roma315	480
170420	578.2	48700	Unouno14	484.83
180420	651.26	50000	Supavash-tie	574.39
190420	705.52	50000	Roclark13	592.59
200420	696.37	45100	Etphonehome	508.36

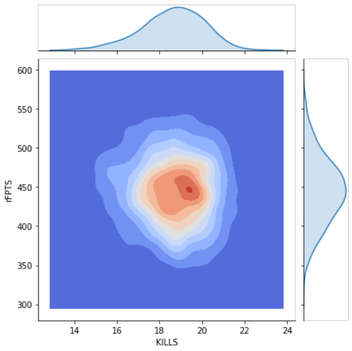
4/17/20

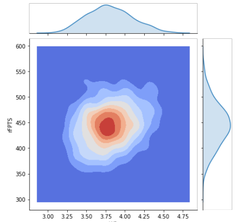


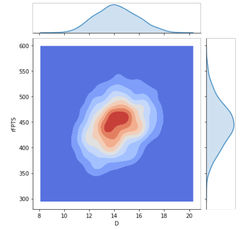

```
CPT          beishang (14485917) (LOCKED)
TOP        chenlun17 (14485849) (LOCKED)
JNG          beishang (14485863) (LOCKED)
MID             xiye (14485856) (LOCKED)
ADC            Puff (14485871) (LOCKED)
SUP       Southwind (14485859) (LOCKED)
TEAM        Team WE (14485876) (LOCKED)
Budget                                50000
FPPG                                499.45
OWN                                  2.134
rOWN                                 2.3792
rFPTS                               565.41
WR                                   4.606
D                                    19.02
A                                    47.4
KDA                                  39.79
CS                                 1402.18
KPAR                                 4.443
CSPM                                 42.98
GPM                                   1531
KS                                   1.548
CP                                      68
KILLS                                22.67
dtype: object
```

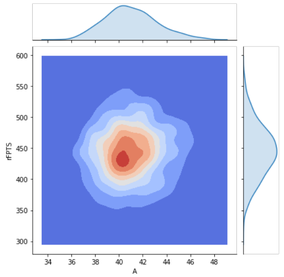


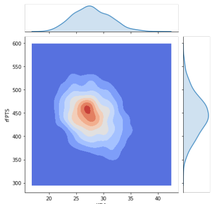


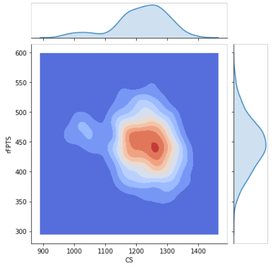


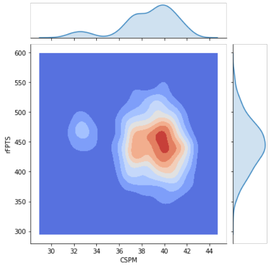


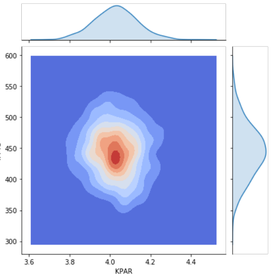


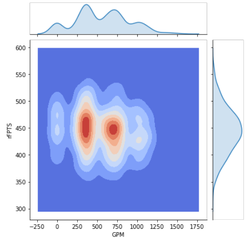


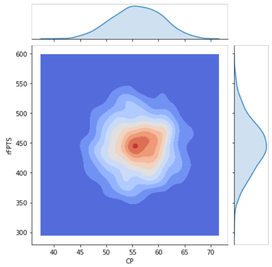


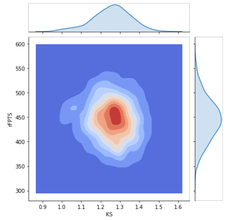






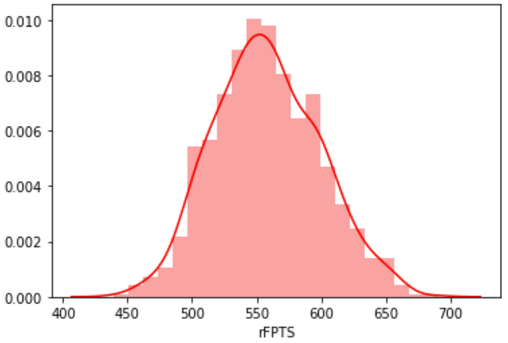




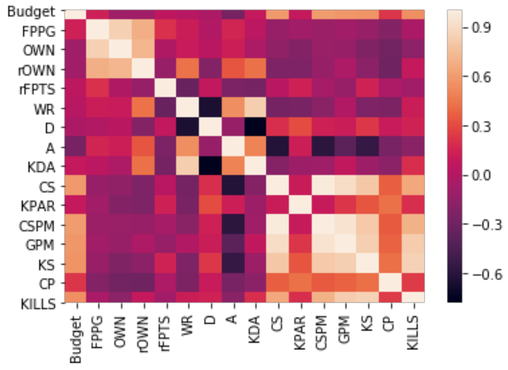


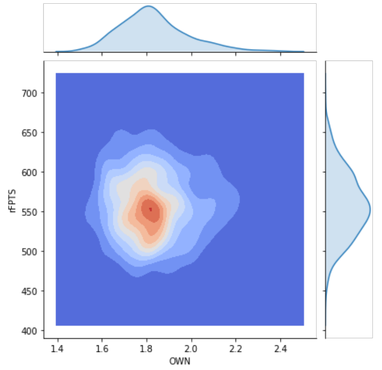
	Budget	FPPG	OWN	rOWN	rFPTS	WR	D	A
Budget	1.000000	0.033781	-0.223498	-0.135846	-0.015109	-0.124477	-0.046123	-0.451852
FPPG	0.033781	1.000000	0.766254	0.448785	0.269118	0.598925	-0.124162	0.390125
OWN	-0.223498	0.766254	1.000000	0.217228	0.385646	0.278004	0.295089	0.393610
rOWN	-0.135846	0.448785	0.217228	1.000000	-0.195817	0.277646	-0.495526	0.476587
rFPTS	-0.015109	0.269118	0.385646	-0.195817	1.000000	0.262099	0.300538	0.068700
WR	-0.124477	0.598925	0.278004	0.277646	0.262099	1.000000	-0.339729	0.189120
D	-0.046123	-0.124162	0.295089	-0.495526	0.300538	-0.339729	1.000000	-0.091264
A	-0.451852	0.390125	0.393610	0.476587	0.068700	0.189120	-0.091264	1.000000
KDA	0.024642	0.234038	-0.231279	0.577788	-0.299890	0.445891	-0.918893	0.229149
CS	0.551162	-0.148110	-0.480826	-0.069004	-0.166980	0.168047	-0.379284	-0.710572
KPAR	0.041243	-0.111881	-0.202556	0.435340	-0.043623	-0.384539	-0.223488	0.336257
CSPM	0.580061	-0.147274	-0.441371	-0.155384	-0.095638	0.192267	-0.252159	-0.733495
GPM	-0.001819	-0.309064	0.038223	-0.044895	-0.128023	-0.705295	0.330934	0.182598
KS	0.552209	-0.227194	-0.402211	-0.125931	-0.104830	-0.086215	-0.044825	-0.735726
CP	-0.005954	0.058311	0.151747	-0.167406	0.024075	-0.315021	0.271908	-0.167586
KILLS	0.495926	-0.024719	-0.236008	-0.112701	-0.049258	0.178688	0.095202	-0.580981

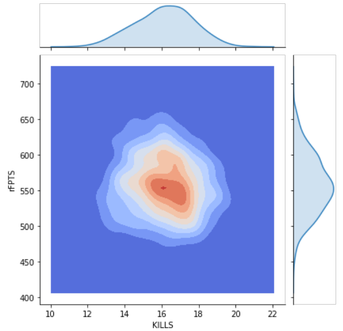
4/18/20

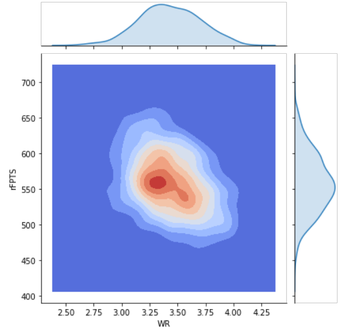


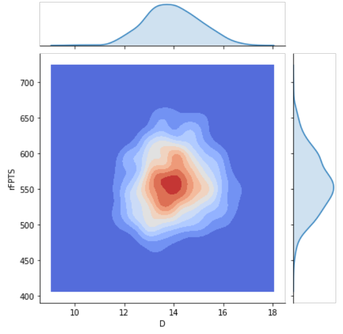

```
CPT              iBoy (14486367)
TOP               bin (14486267)
JNG           xiaohan (14486292)
MID               clx (14486276)
ADC                y4 (14486300)
SUP          TusiN (14486280)
TEAM    Victory Five (14486308)
Budget           50000
FPPG           569.375
OWN                2.4
rOWN               2.29
rFPTS            690.84
WR                4.159
D                 17.18
A                 46.3
KDA               31.05
CS              1417.25
KPAR              4.422
CSPM              43.35
GPM                2269
KS                1.532
CP                   80
KILLS             20.82
dtype: object
```

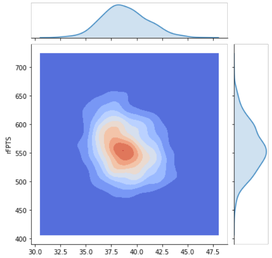


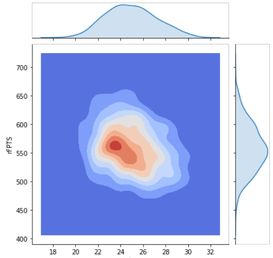


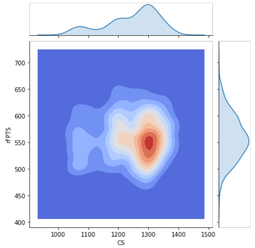


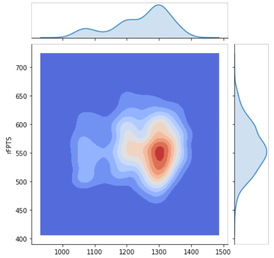


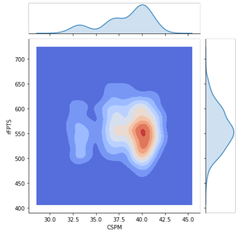


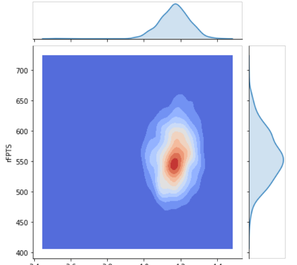


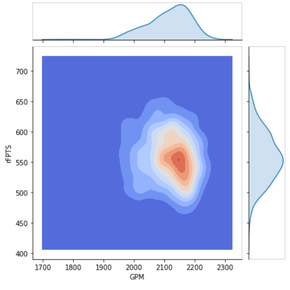


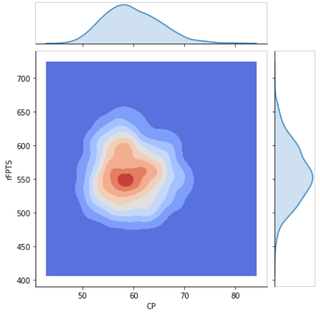


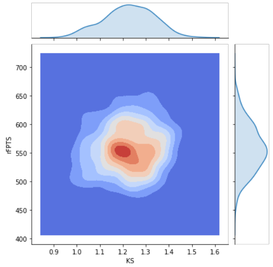








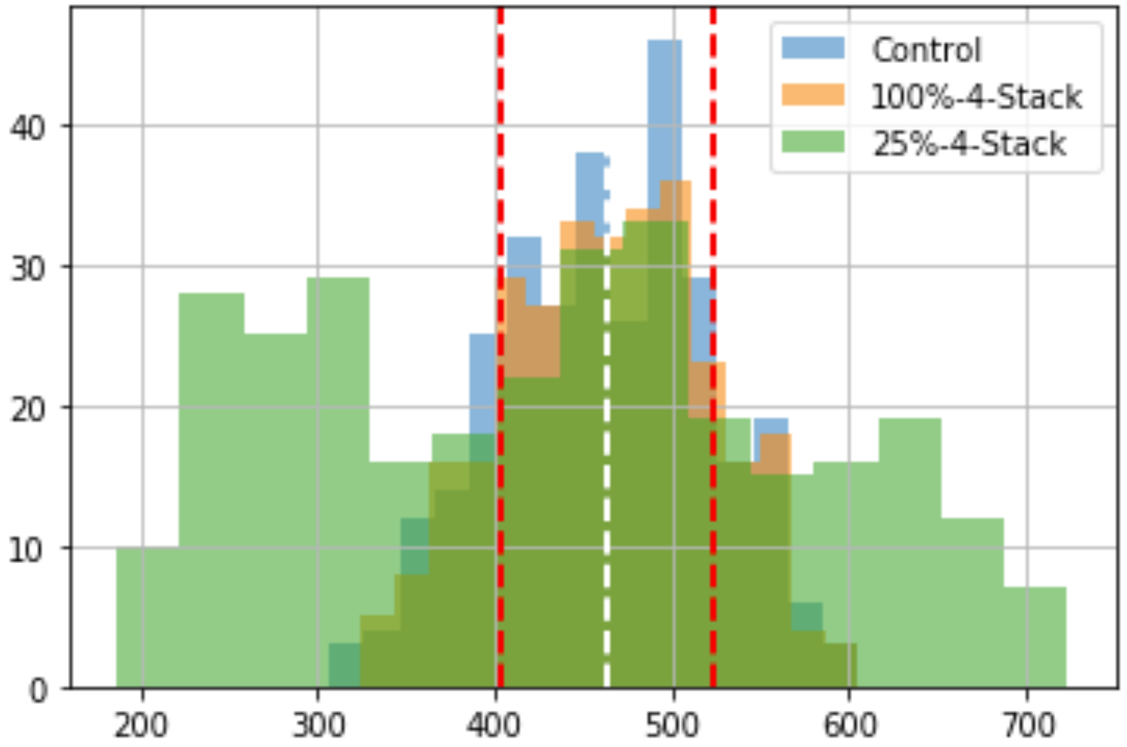




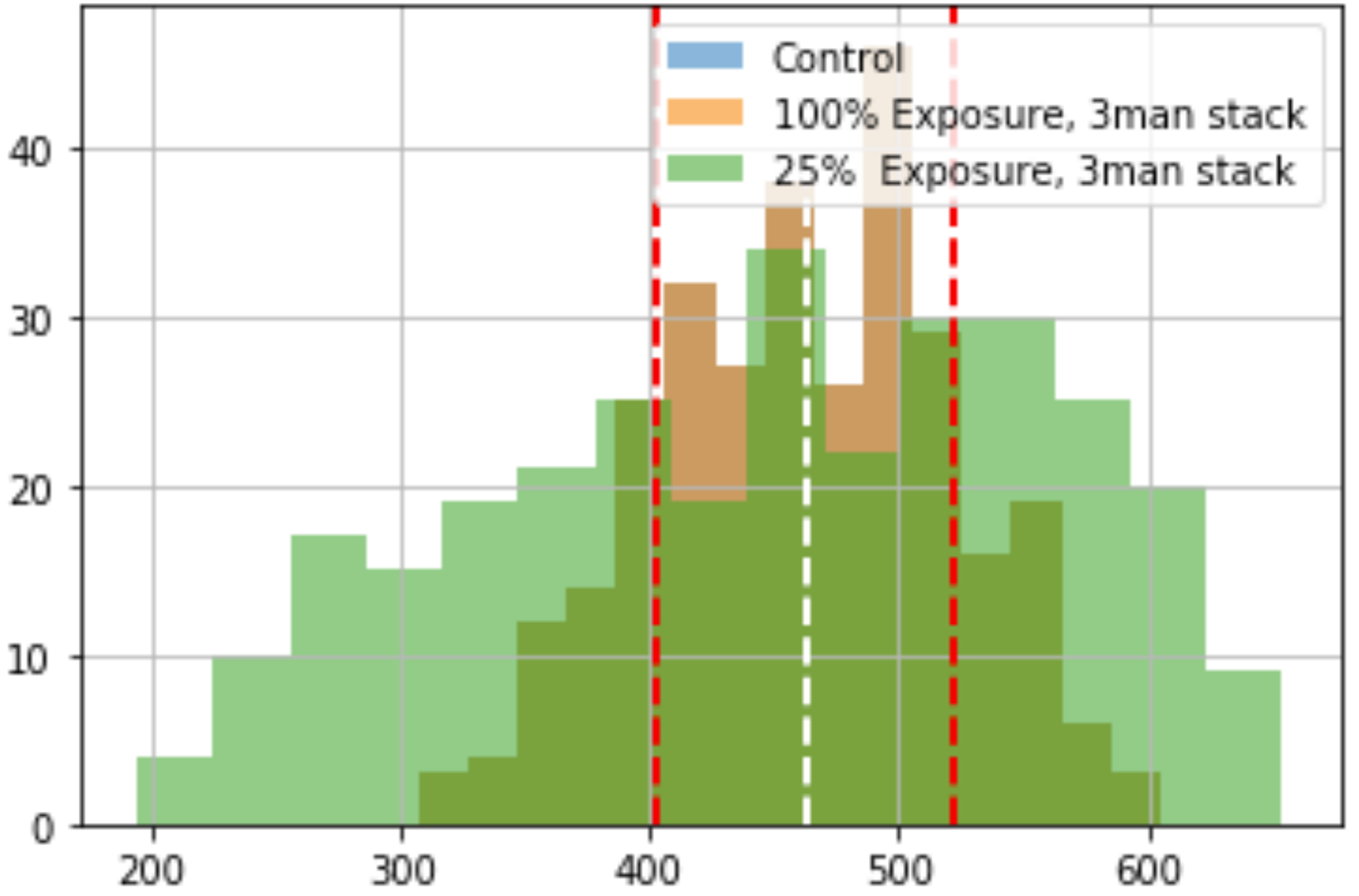
	Budget	FPPG	OWN	rOWN	rFPTS	WR	D	A
Budget	1.000000	0.124088	-0.091791	-0.087299	0.050307	0.024609	-0.030348	-0.269708
FPPG	0.124088	1.000000	0.840818	0.681601	0.205608	0.115760	-0.001935	0.157217
OWN	-0.091791	0.840818	1.000000	0.717417	-0.015878	0.098922	0.040627	0.123299
rOWN	-0.087299	0.681601	0.717417	1.000000	-0.131836	0.436389	-0.213726	0.338764
rFPTS	0.050307	0.205608	-0.015878	-0.131836	1.000000	-0.333779	0.065774	-0.249652
WR	0.024609	0.115760	0.098922	0.436389	-0.333779	1.000000	-0.650541	0.535669
D	-0.030348	-0.001935	0.040627	-0.213726	0.065774	-0.650541	1.000000	-0.125018
A	-0.269708	0.157217	0.123299	0.338764	-0.249652	0.535669	-0.125018	1.000000
KDA	0.088618	0.047504	-0.022308	0.426483	-0.272668	0.831769	-0.779652	0.493364
CS	0.583755	-0.130815	-0.157044	-0.236012	0.027610	-0.278465	0.193648	-0.626496
KPAR	0.086605	-0.080868	-0.213108	-0.236497	0.132619	-0.260632	0.298315	0.123240
CSPM	0.603364	-0.105861	-0.119078	-0.125770	-0.053867	-0.183742	0.131580	-0.585560
GPM	0.582448	-0.081615	-0.124115	-0.026281	-0.131683	-0.011952	0.112446	-0.385172
KS	0.562511	-0.128233	-0.227808	-0.178056	0.142805	-0.232684	0.234652	-0.551022
CP	0.223038	-0.215751	-0.286992	-0.302693	-0.028380	-0.240552	0.103155	-0.260177
KILLS	0.544556	-0.041035	-0.172080	0.084761	-0.074719	0.112377	0.138748	-0.185448

4/19/20

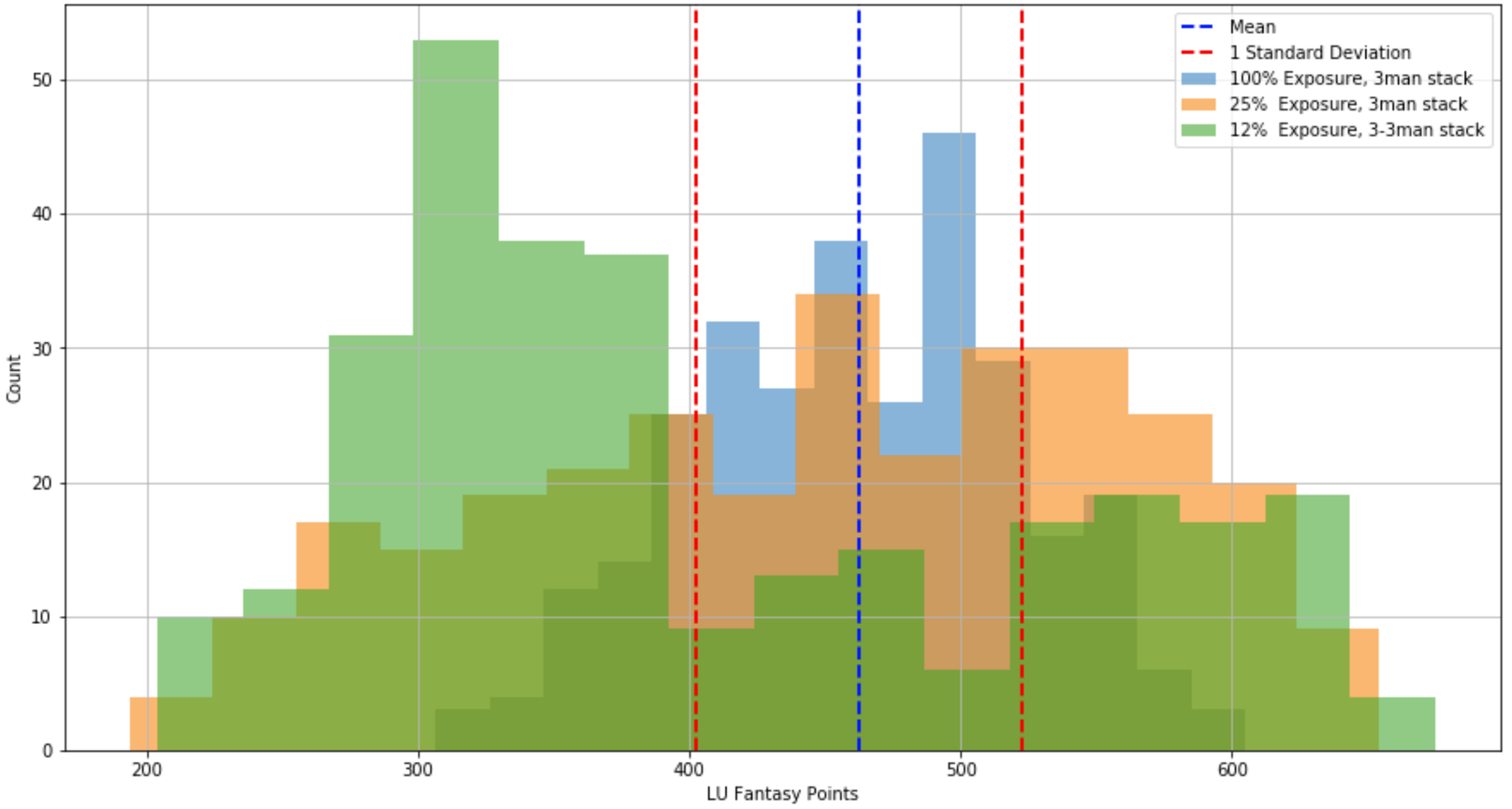
04/11/20

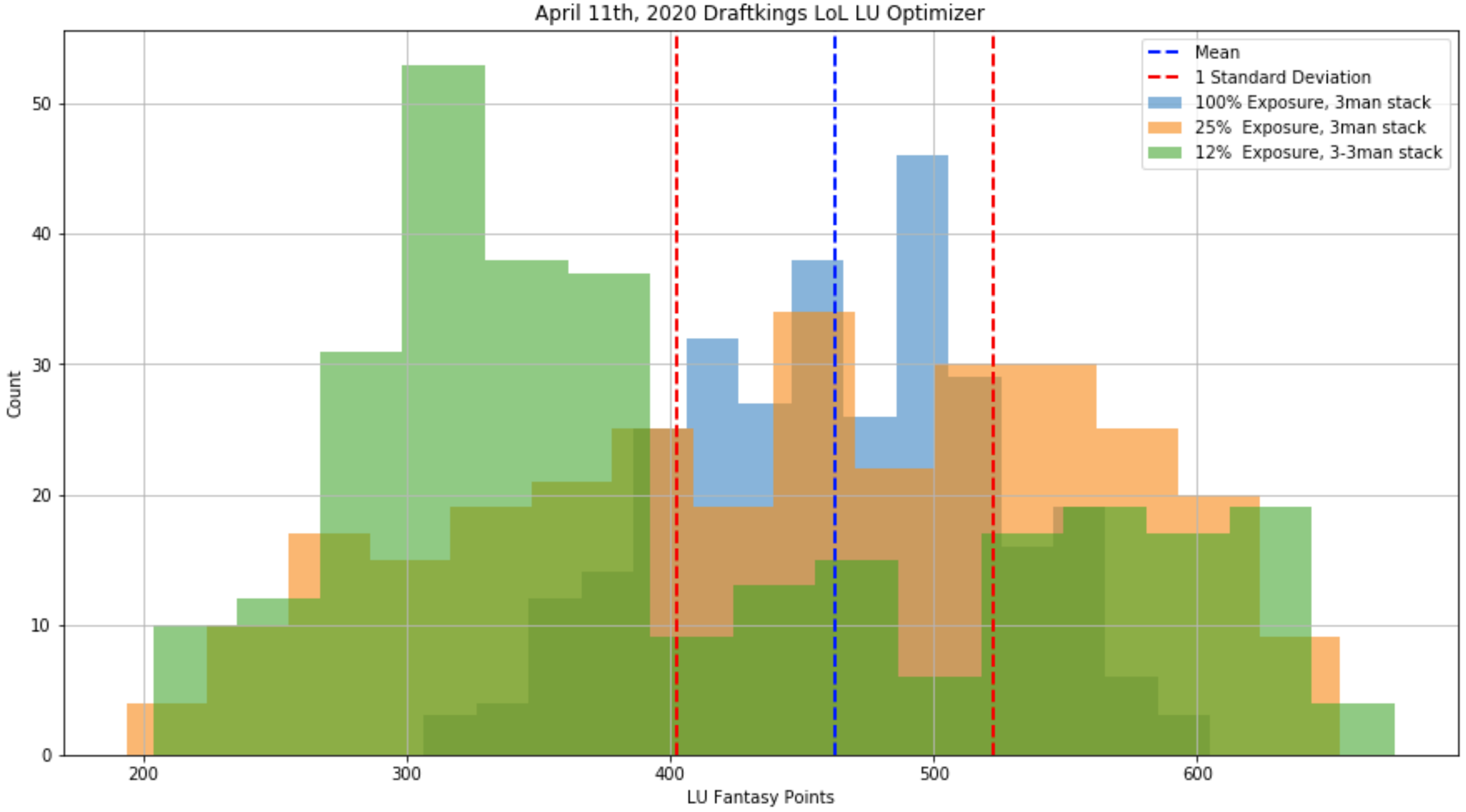


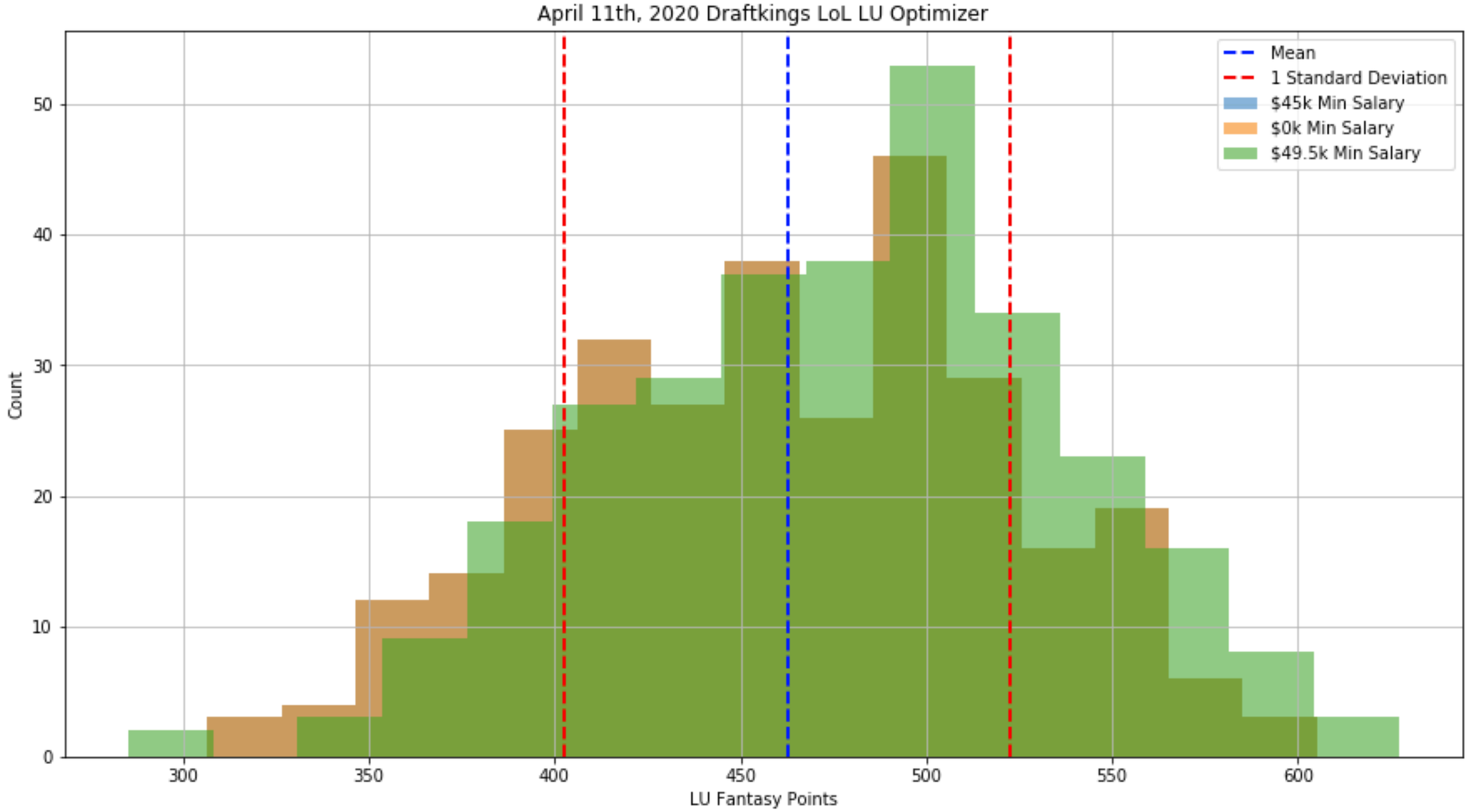
	Name	Kurtosis
0	100-4-stack	-0.611812
1	Control	-0.554862
2	25%-4-Stack	-1.025616



	Name	Kurtosis
0	Control	-0.554862
1	100% Exposure, 3man stack	-0.554862
2	25% Exposure, 3man stack	-0.911300


Mean
1 Standard Deviation
100% Exposure, 3man stack
25% Exposure, 3man stack
12% Exposure, 3-3man stack
Count
0
10
20
30
40
50
200
300
400
500
600
LU Fantasy Points


April 11th, 2020 Draftkings LoL LU Optimizer
Mean
1 Standard Deviation
100% Exposure, 3man stack
25% Exposure, 3man stack
12% Exposure, 3-3man stack
Count
0
10
20
30
40
50
200
300
400
500
600
LU Fantasy Points



Changing the minimum salary will cause a