

Championship Series CS105 Final Project

David Ryan, Kendrew Christanto, Nicholas Chang, Jonathan Thai

What is Rocket League?

- An online multiplayer game in which up to 8 players play 5-min soccer in rocket-powered cars
- Players control cars to score goals on the opposing team's goal area.
- Players can jump to hit the ball while being mid-air
- Players can utilize <u>speed boost</u> to speed across the field, add momentum towards hitting the ball, or <u>demolish</u> opposing team's cars
- Rocket League Championship Series is an <u>esport</u> tournament based on Rocket League

The Goal of this Project



- To draw correlations between winning matches alongside various game statistics of players that perform in the Rocket League Championship Series 2021-2022.
- We hope to accomplish by relating winning matches to data points like <u>goals</u> scored, speed boost, movement, positioning, steering sensitivity, and mvp <u>status</u> with respect to <u>wins</u>.
- In order to find correlation, we plan to do:
 - Linear Regression analysis
 - K-Means Clustering
 - Chi-Square Analysis

The Data

- We needed data for player and team game stats for goals, assists, boost amount, movement speed and distance, offensive and defensive positioning, and steering sensitivity.
 - Since this Esport event is annual, we took the data from 2021-2022.
- Dataset: https://www.kaggle.com/datasets/dylanmonfret/rlcs-202122
 - Because there's so much data, we looked into specifically games_by_players.csv and games_by_teams.csv.
- The data is cleaned to only include relevant data points of interest

Dataset (games_by_players.csv)

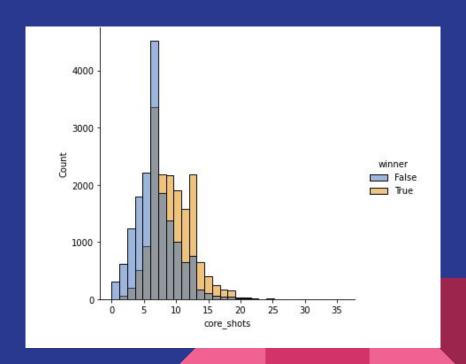
# core_goals = Goals.	# core_assists = Assists (last pass before a goal).	# core_score = Game score.	# boost_avg = Average amount of boost.	# boost_tim = Time spent with full (100) boost.	# movement = Average speed.	# movement = Total distance traveled in the field.	# movement = Part of the time spent on the ground.	## movement = Part of the time spent low in the air.	# movement = Part of the time spent high in the air.	# positionin = Average distance to the ball.	# positionin = Time spent on defensive third.	# positionin = Time spent in defensive half.	# positionin = Part of the time spent in the offensive third.	# positionin = Part of the time spent in the offensive / opponent's half.	# advanced F Goals participation rate.
0 9	0 8	0 1642	17.9 69.4	0 210	546 2.02k	21.8k 1.84m	36.8 91.8	8.2 58	0 25.2	2k 4.71k	12.4 667	19.1 858	0 64.8	0 81	0 100
1	0	348	45.92	55.88	1446.0	460974.0	51.139492	38.491447	10.369063	2848.0	193.98	242.43	15.740048	27.496487	50.0
0	1	343	51.26	27.82	1619.0	518367.0	50.295986	44.90553	4.798493	2829.0	155.44	205.38	21.179775	38.59539	50.0
1	0	230	47.91	36.14	1462.0	454065.0	58.292614	37.654037	4.053348	3047.0	164.31	212.25	20.283932	34.7766	50.0
0	0	469	53.41	28.71	1588.0	507457.0	61.898357	31.521673	6.57997	3097.0	158.44	220.21	18.32885	34.16938	0.0
0	0	353	48.43	51.28	1616.0	515947.0	51.726818	40.271503	8.001675	3039.0	156.7	220.85	17.844755	33.962265	0.0
1	0	246	50.55	33.45	1632.0	515994.0	52.498642	42.872234	4.6291265	2757.0	141.9	192.33	28.076529	41.96089	100.0
2	0	499	53.65	54.26	1597.0	530662.0	54.894745	37.164364	7.9408894	3045.0	165.85	219.81	25.950813	38.711838	50.0
1	2	414	55.37	37.48	1630.0	546844.0	48.111977	42.991505	8.896511	2733.0	165.03	213.75	25.23168	40.869736	75.0
1	1	370	50.24	28.15	1507.0	503930.0	60.863667	34.56886	4.567474	3144.0	177.07	226.18	22.322489	37.38962	50.0
2	0	455	49.89	18.12	1552.0	509929.0	53.22835	41.414513	5.357143	3128.0	179.97	233.76	16.541058	34.263218	100.0
0	0	294	46.19	35.09	1703.0	561274.0	48.897697	44.97497	6.1273265	3515.0	188.1	251.75	16.717283	29.206154	0.0
0	1	274	45.99	31.3	1476.0	490058.0	59.224464	37.368977	3.4065566	3234.0	193.44	276.7	8.142614	22.86678	50.0
2	1	477	49.29	40.35	1503.0	527154.0	59.463436	38.21789	2.3186698	2888.0	174.33	232.98	21.54868	37.62082	100.0
1	2	452	53.17	46.44	1651.0	588968.0	44.736355	44.81277	10.450868	2874.0	147.25	196.36	28.778097	48.258236	100.0
0	0	272	47.53	44.39	1591.0	563734.0	55.422424	38.876198	5.7013817	3231.0	155.08	218.85	23.7407	41.857063	0.0
1	1	439	45.89	50.4	1571.0	561050.0	52.756466	41.762238	5.4812937	2879.0	207.66	265.33	14.59344	30.04377	100.0
1	1	362	48.22	37.19	1507.0	538689.0	60.8911	33.976234	5.132664	3055.0	200.34	282.07	13.471926	25.679129	100.0
0	0	247	53.84	27.46	1609.0	574370.0	55.826393	39.40655	4.7670493	2642.0	194.53	266.56	15.491845	29.75651	0.0
5	2	823	50.49	18.96	1674.0	572067.0	56.023792	37.252712	6.7234926	3079.0	112.78	177.0	37.10401	53.00053	87.5
1	4	567	53.12	36.08	1648.0	571707.0	51.91847	40.194347	7.8871827	2833.0	137.91	211.16	24.354145	44.392067	62.5
2	2	548	53.9	39.3	1587.0	543074.0	53.629536	37.203068	9.167397	3162.0	145.15	212.53	26.1585	43.582596	50.0
0	0	257	47.79	30.73	1607.0	549803.0	56.606827	40.02868	3.3644907	2773.0	257.78	311.81	7.7672734	17.197334	0.0
0	0	247	48.07	29.37	1524.0	526598.0	58.65048	34.85475	6.494772	2760.0	229.96	298.52	9.112697	21.377966	0.0
0	0	126	39.68	18.26	1447.0	490634.0	58.418995	35.92997	5.6510334	3010.0	257.81	306.49	9.163187	17.954277	0.0
3	3	694	49.98	24.49	1597.0	571943.0	47.87979	45.605503	6.514708	3217.0	128.5	193.18	30.946133	50.80097	100.0
2	2	529	50.23	38.67	1581.0	573110.0	54.22384	41.596542	4.1796174	2816.0	139.15	219.84	24.620323	44.44697	66.666666666 6666
1	1	344	53.88	56.96	1641.0	596951.0	55.94996	36.84039	7.2096457	3301.0	138.11	213.98	26.863874	46.357483	33.333333333 3333
1	1	469	38.51	28.82	1616.0	570134.0	56.330425	39.222782	4.4467983	2911.0	224.98	285.05	13.206037	26.347477	100.0
1	0	445	39.33	20.39	1499.0	534578.0	54.60796	40.70386	4.6881766	2705.0	244.87	303.4	11.202221	22.738037	50.0
0	0	124	50.45	40.21	1578.0	575205.0	63.101807	31.02307	5.8751254	3121.0	221.58	294.73	11.48696	26.095787	0.0
2	3	875	55.41	31.98	1593.0	557275.0	48.501556	42.451103	9.047337	2865.0	203.48	273.81	15.227027	29.443142	71.4285714285 7143
4	1	802	49.06	26.09	1611.0	556288.0	55.738304	40.003143	4.258551	3011.0	157.26	224.16	24.639883	41.293243	71.4285714285 7143

Dataset (games_by_teams.csv)

# ball_possession_t =			# core_shooting_pe =	# boost_avg_amount =	# movement_total =	# movement_time_l =	# movement_time =	# positioning_time =	# positioning_time =	# positioning_time_
Time spent with ball possession.	Shots on target.	Goals.	Goals / shots ratio.	Average amount of boost.	Total distance traveled in the field.	Time spent low in the air.	Time spent high in the air.	Time spent in neutral third / midfield.	Time spent in offensive third.	Time spent in defen half.
			h							
12.3 587	0 36	0 18	0 150	102 190	188k 5.36m	39.7 1.23k	1.9 269	33.2 1.21k	21.9 890	63.4
136.54	13	2	15.384615384615385	145.09	1433406.0	401.4299999999995	63.91	291.06	189.480000000000002	660.06
153.6	4	1	25.0	152.39	1539398.0	382.19	64.11	329.25	214.03000000000000	633.39
155.79	10	4	40.0	159.26	1581436.0	413.58	77.14	308.51	264.91999999999996	659.74
126.81	6	2	33.333333333333	142.07	1561261.0	441.260000000000005	53.06	360.960000000000004	147.480000000000002	762.21
165.76	11	3	27.272727272727	149.99	1679856.0	459.13	69.78	373.65999999999997	279.05	648.19
141.17	5	2	40.0	147.95	1674109.0	436.89	58.36	370.5	165.27	813.96
140.93	14	8	57.14285714285714	157.51	1686848.0	433.089999999999	89.81	406.49	330.77	600.68999999999
126.65	3	0	0.0	135.54	1567035.0	417.300000000000007	58.44	286.2	98.08	916.81999999999
142.42	14	6	42.857142857142854	154.09	1742004.0	490.64	70.88	455.41999999999996	326.1	627.0
151.1	5	2	40.0	128.290000000000002	1679917.0	435.36	59.0500000000000004	346.1699999999999	140.91	883.180000000000
130.82	18	7	38.8888888888889	157.97	1669160.0	462.01	77.59	385.25	235.890000000000001	744.96
139.44	8	2	25.0	135.26	1646360.0	457.86	56.98	369.59000000000000	195.3	800.170000000000
148.71	14	2	14.285714285714285	167.74	1461339.0	337.09	69.34	367.22	284.62	505.14
132.09	3	1	33.3333333333333	155.01	1397388.0	363.92	67.14	284.25	82.2	782.579999999999
142.62	11	7	63.63636363636363	155.42000000000000	1618589.0	454.74	64.94	375.28	241.7	765.94
129.13	6	2	33.3333333333333	149.62	1631433.0	439.64	50.48999999999995	370.63	152.53	857.78
238.29	10	0	0.0	166.9599999999998	2113964.0	525.19	88.4499999999999	493.09000000000000	269.67	953.75
223.13	9	1	11.1111111111111	160.95	2210247.0	526.88	70.83	527.59	297.770000000000004	880.95
132.03	11	3	27.272727272727	156.72	1486993.0	375.4199999999996	68.12	351.31	202.95	675.83
159.91	6	2	33.333333333333	160.77	1484975.0	368.86	58.730000000000004	319.93	137.10999999999999	787.0

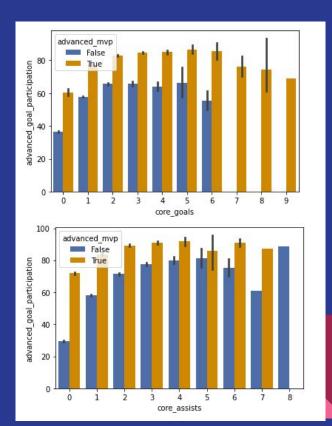
EDA: Shots based on winning teams

 Winning teams tend to shoot towards the goal more often than losing teams because it leads to higher chances of scoring.



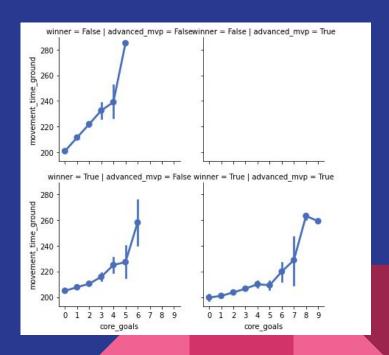
EDA: MVP players v. Regular players on winning teams

- When comparing MVP players to regular players on winning teams, we find that MVP players tend to score goals more often than regular players, leading to wins
- They also happen to assist more often than regular players
- This means you can win matches in RLCS if you have an MVP player who can shoot goals and assist



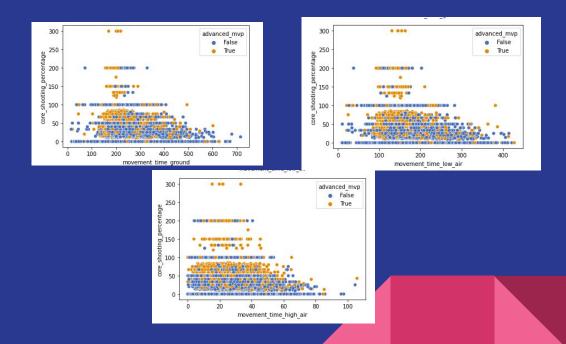
EDA: Time spent in the ground or air

- Those who are winning players, regardless of MVP status tend to stay on the ground less when scoring early goals.
- MVP players, in fact, score early goals on the ground much less that regular winning players
- This leads to the fact that MVP players score more often in the air



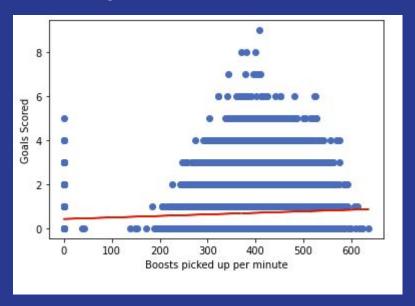
EDA: Time spent in the ground or air

- This is backed up by these 3 scatter plots here.
- This shows that MVPs with a greater shooting percentage tend to spend more time high in the air, which is most likely where they take their shots from.

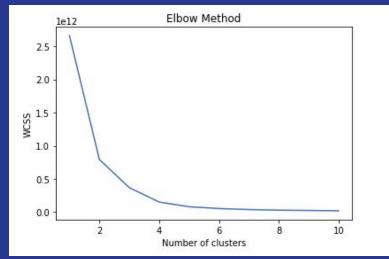


Linear Regression

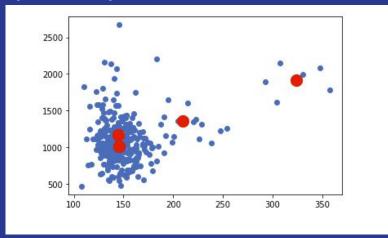
- For predicting goals scored based on boost pickups per minute.
- Not a good predictor of goals.



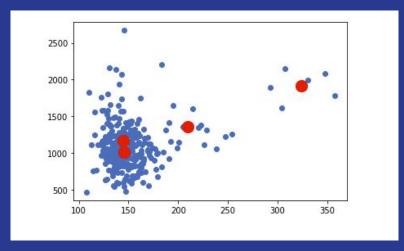
- First K-Means used to find clustering for ball possession time and the core score.
- We first find the optimal number of clusters using the elbow method
- We pick the number at the elbow which is 4 in this case.



- Using the optimal number of clusters we found we plug them
- The points put into the graph which are called centroids
- We calculate the distance of each point to the centroid and find the average
- Those steps are repeated until clusters are found.

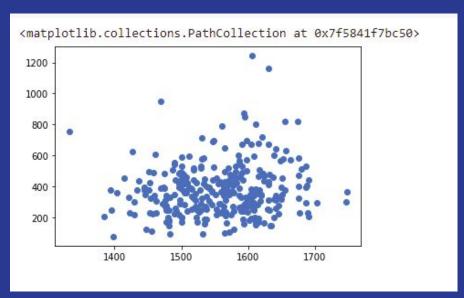


- The groups are formed where the clusters are.
- This means that there are clusters of data when looking at a ball possession time and core score graph

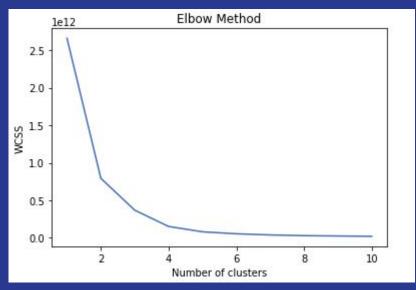


The same is done for the other one for average movement speed and core

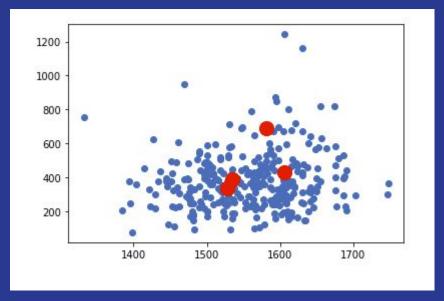
score



We find the optimal number of clusters using the elbow method which is 4

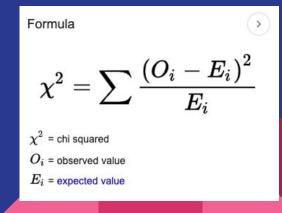


And we find the groups using the clusters.



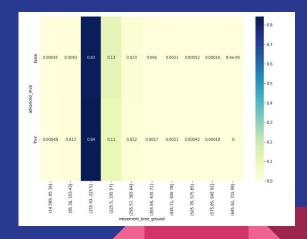
Chi Squared Test [Correlation Analysis]

- Mathematical test to determine whether there is a statistically significant difference between observed and expected frequencies at the intersection of multiple categorical variables with respect to one variable.
- Can be used to compute a p-value



Findings [MVP Status and Ground Movement Time]

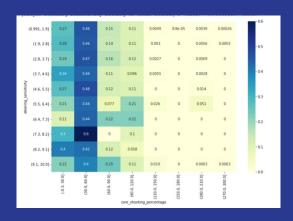
• Produced a contingency table and ran chi squared test. The resulting p-value was 2.4394338384981347e-08

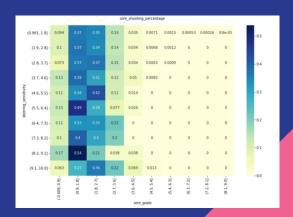


movement_time_ground	(14.589, 85.36)	(85.36, 155.43)	(155.43, 225.5)	(225.5, 295.57)	(295.57, 365.64]	(365.64, 435.71]	(435.71, 505.78]	(505.78, 575.85)	(575.85, 645.92]	(645.92, 715.99]
advanced_mvp										
False	37	763	68944	10632	1951	500	175	43	13	7
True	8	221	13998	1884	364	94	35	7	3	0

MVP Findings [Steering Sensitivity and Performance]

- [Player Steering Sensitivity Setting to Shooting Percentage] P-Value: 0.000178140588134638
- [Player Steering Sensitivity Setting to Goals Scored] P-Value: 0.7884866377610897





Question 1:

What statistics are most important for determining an MVP?

Question 2:

What irregularity can be found when comparing offensive vs defensive positioning statistics?

Question 3:

Are there any correlations between diasporic metrics and MVP status?