Final Project Math 17

Christopher Kingston

2023-05-25

-Introduction-

This is data analysis of ranked play data from Overwatch 2 that uses statistical analysis to evaluate not just player performance, but choice performance, trends between league ranks from Bronze to Platinum using hero statistics like season wide average KTD (Kills To Deaths) ratios and seasonal changes. The game is part of the e-sports "meta" and the developer is generally opaque with releasing data, however uniquely the developer chose to publish a site called Overbuff with seasonal statistics. The data is available on Kaggle: "https://www.kaggle.com/datasets/mykhailokachan/overwatch-2-statistics (https://www.kaggle.com/datasets/mykhailokachan/overwatch-2-statistics)". Here's a general outline of how such an analysis could be conducted:

Data Collection: The first step was to gather data on ranked play from Overwatch 2. This has been accomplished already via Kaggle. I did attempt a more recent BeautifulSoup grab for complete season 3 data, but since the Kaggle Data was pulled and implemented at the beginning of Season 4, Blizzard Entertainment that owns and manages Overwatch 2's API updated their profile policy to auto opt out of sharing data. The result is any updated data was radially incomplete.

Data Processing: Once the data had been collected, it needs to be clean and prepare for analysis. This involved removing missing values, outliers, and duplicates, as well as transforming the data into a format that is suitable for analysis. Many of the statistics are choice specific and require care and game knowledge to know which to remove.

Analysis: Overwatch 2 is a competitive first-person shooter that places players with Hero selections in a shared Player-vs-Player (PvP) environment. Player's pick from 36 Heroes across three roles: Tank, Support, and Damage. The main objective of playing the game is to win and the choices players make before entering the game can play a significant factor in outcome and advancement within the league. I used exploratory and statistically analysis to reveal the dominant choices that would assist in victory and advancement. Game play statistics were organized and means tested to find the dominant Role, play style, and Hero

Season and Data Structure Analysis: Overwatch 2 has had two complete seasons in the data sets and one incomplete Season 3 stretched from February 7th to April 3 and the data was acquired March 12th. 33 of the 55 days or 60% of teh season was captured by the data. Season 1 was held October 4, 2022 to December 6, 2022 for 63 days. Season 2 was held December 6, 2022 to February 7, 2023 also for 63 days.

Season 1 and 2 were 63 days long and Season 3 was scheduled for only 55 days. Be aware that though season 3 data is incomplete, the season is shorter than previous seasons so captured more fo the percent of the season than had they been the previous season length.

```
# Overwatch 2 has had two complete seasons in the data sets and one incomplete
# Season 3 stretched from February 7th to April 3 and the data was acquired
# March 12th.

season3start = as.Date('2023-2-7')
season3end = as.Date('2023-4-3')
season3length <- length(seq(from=season3start, to=season3end, by='day')) - 1
datapulldate = as.Date('2023-3-12')
length(seq(from=season3start, to=datapulldate, by='day')) - 1
datacompletion = 33/55
datacompletion


# Season 1 October 4, 2022 to December 6, 2022
season1start = as.Date('2022-10-4')
season1end = as.Date('2022-12-6')
season1length <- length(seq(from=season1start, to=season1end, by='day')) - 1
# Season 2 December 6, 2022 to February 7, 2023
season1start = as.Date('2022-12-6')
season1end = as.Date('2023-2-7')
season2length <- length(seq(from=season1start, to=season1end, by='day')) - 1

# 60% of season 3 was complete
# Season 1 and 2 were 63 days long and Season 3 was scheduled for only 55 days.
# Be aware that though season 3 data is incomplete, the season is shorter than
# previous seasons so captured more fo the percent of the season than had they
# been the previous season length
```

Role Choice Analysis (Pick and win rate)

Win Rates: Comparing Season One Support to Damage than Support to Tank and Damage win rates were positive, showing Support had a distinct win advantage over other roles in Season One. Support Mean - 46.82844 Tank Mean 46.80025 Damage Mean 46.19618

Comparing Season Two Support to Damage than Support to Tank and Damage win rates were positive and the trend continued form Season One. Support Mean - 46.8925 Tank Mean - 46.47125 Damage Mean - 46.17471

Comparing Season Three Support to Damage than Support to Tank and Damage win rates were positive the trend continued. Support Mean - 48.98531 Tank Mean - 46.92761 Damage Mean - 47.98544

The conclusion for win rate was that between all three season Support roles have greater mean win rates than Tank and Damage roles.

Pick Rates: Support had a higher average pick rate in all three seasons. Despite having the highest pick, the roles had different pool amounts of characters that I was concerned was distorting the pick rate. Out of the 36 hero selections: 17 are Damage, 11 Tank, and 8 Support. I created a PKI to multiple the Role average pick rates by the percentage of their representation to correctly reflect preference. The Support pick rate fell fromt eh highest almsot thriple another to the lowest.

Conclusion of this first section is that a player selecting to play Support over the other two roles will experience more success and less competition.

```
# Seasons Grouped by Role
Season1Role <- Season1Cleaned |>
  group_by(Role) |>
  summarise(Average_win_rate_S1 = mean(Win.Rate...),
                                    Average_pick_rate_S1 = mean(Pick.Rate...))
Season2Role <- Season2Cleaned |>
  group_by(Role) |>
  summarise(Average_win_rate_S2 = mean(Win.Rate...),
            Average_pick_rate_S2 = mean(Pick.Rate...))
Season3Role <- Season3Cleaned |>
  group_by(Role) |>
  summarise(Average_win_rate_S3 = mean(Win.Rate...),
            Average_pick_rate_S3 = mean(Pick.Rate...))

# Join Season Roles
SeasonalRoles <- full_join(Season1Role, Season2Role,by='Role')
SeasonalRoles <- full_join(SeasonalRoles, Season3Role,by='Role')

# Wins data frame
SeasonalWinRole <- SeasonalRoles |>
  select(Role, Average_win_rate_S1, Average_win_rate_S2, Average_win_rate_S3)

# Isolating win rates by role
# Season 1
S1WinS <- Season1Cleaned |>
  filter(Role == 'Support') |>
  select(Win.Rate...)
#S1WinS
S1WinT <- Season1Cleaned |>
  filter(Role == 'Tank') |>
  select(Win.Rate...)
#S1WinT
S1WinD <- Season1Cleaned |>
  filter(Role == 'Damage') |>
  select(Win.Rate...)
#S1WinT
# Season 2
S2WinS <- Season2Cleaned |>
  filter(Role == 'Support') |>
  select(Win.Rate...)
#S2WinS
S2WinT <- Season2Cleaned |>
  filter(Role == 'Tank') |>
  select(Win.Rate...)
#S2WinT
S2WinD <- Season2Cleaned |>
  filter(Role == 'Damage') |>
  select(Win.Rate...)
#S2WinT
# Season 3
S3WinS <- Season3Cleaned |>
  filter(Role == 'Support') |>
```

```
  select(Win.Rate...)
#3S3WinS
S3WinT <- Season3Cleaned |>
  filter(Role == 'Tank') |>
  select(Win.Rate...)
#3S3WinT
S3WinD <- Season3Cleaned |>
  filter(Role == 'Damage') |>
  select(Win.Rate...)
#S3WinT

# Testing Win Means

# Setup
xmean <- mean(S1WinS$Win.Rate...)
ymean <- mean(S1WinD$Win.Rate...)
xmean
```

```
## [1] 46.82844
```

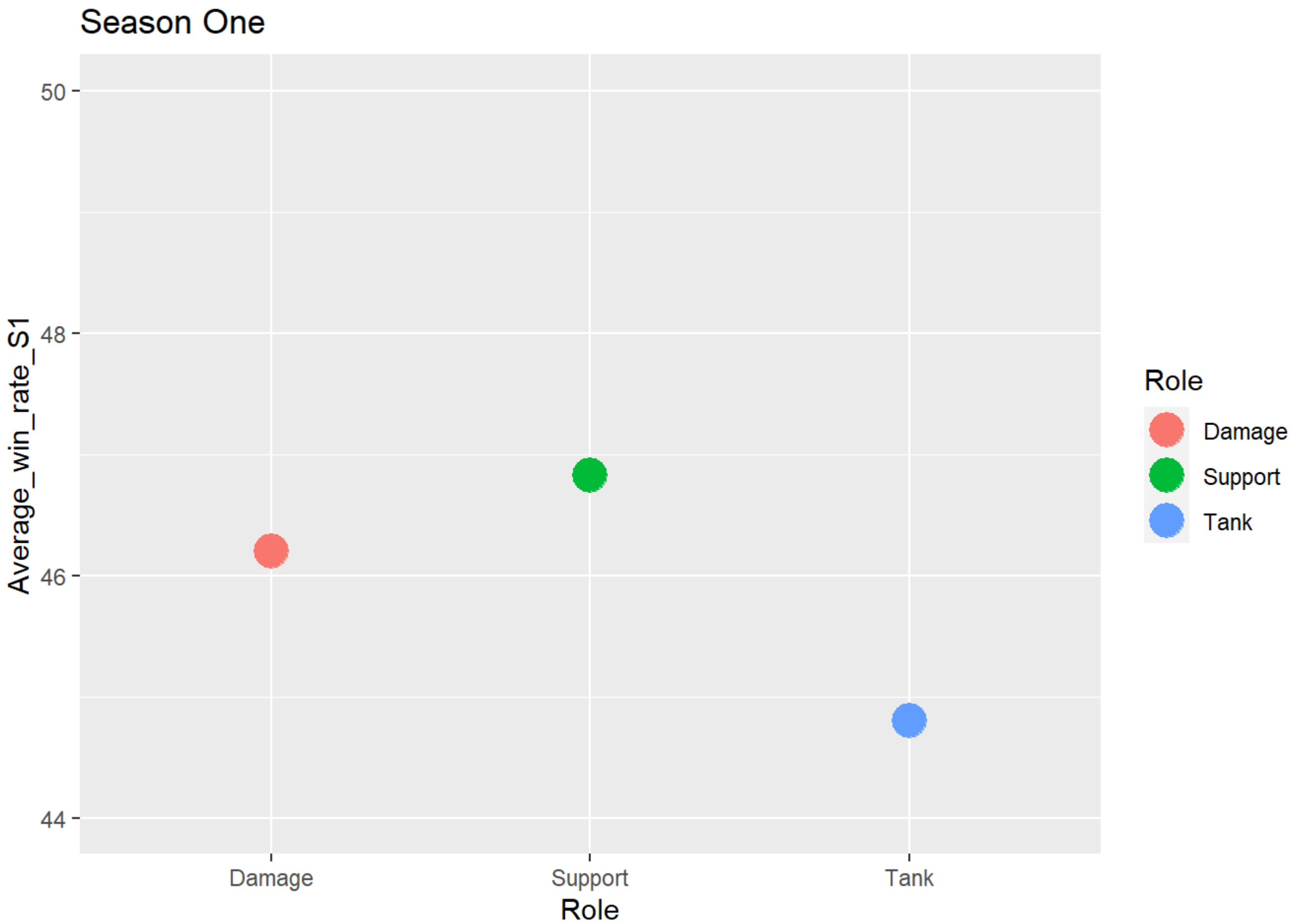
```
ymean
```

```
## [1] 46.19618
```

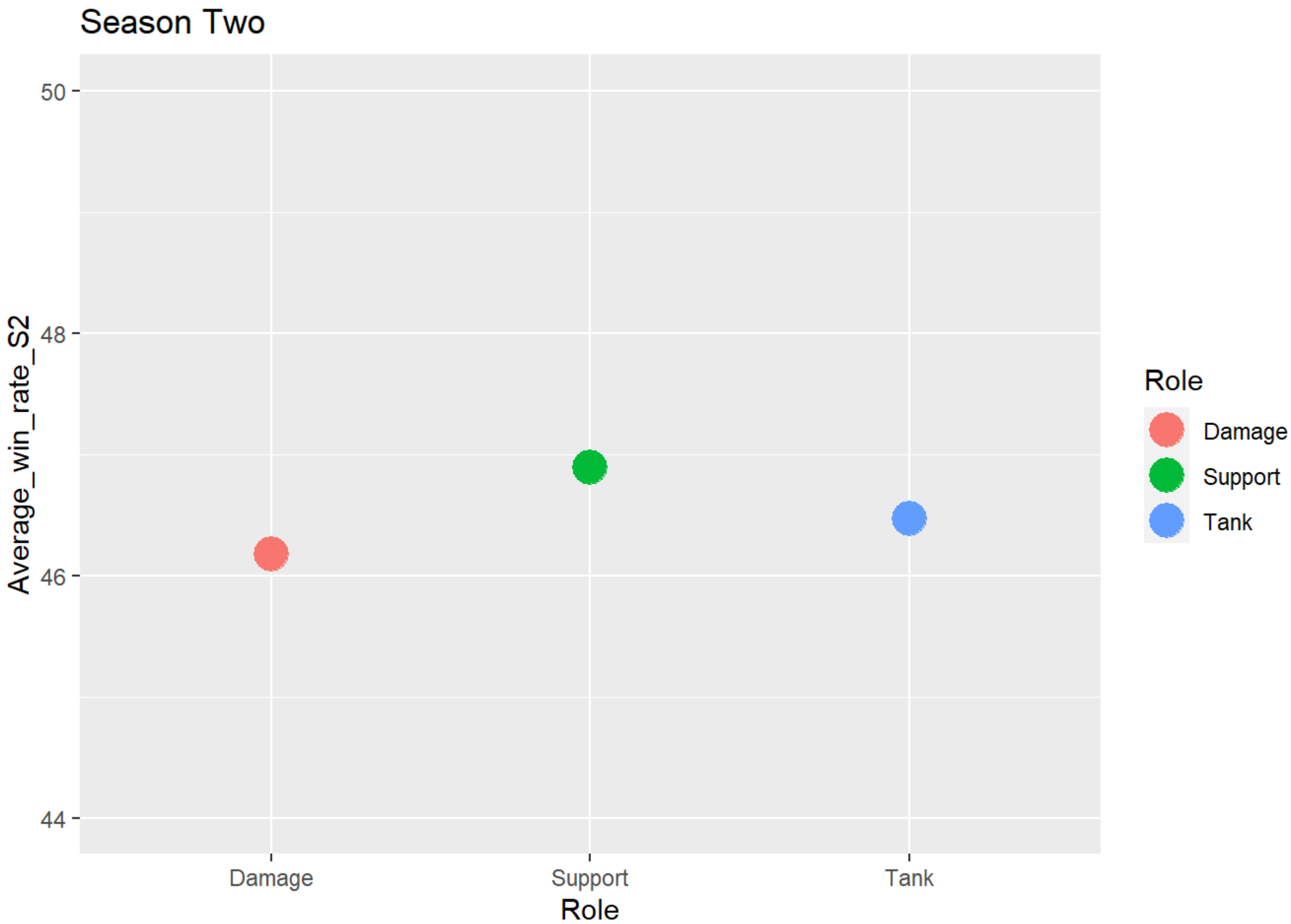
```
xmean <- mean(S1WinS$Win.Rate...)
ymean <- mean(S1WinT$Win.Rate...)

# Function to compare means
CompareMeans <- function(x,y){
  if (mean(x)==mean(y)) return('same')
  if (mean(x)>mean(y)) return('positive')
  if (mean(x)<mean(y))return('negative')
}
CompareMeans(xmean,ymean)
```

```
## [1] "positive"
```

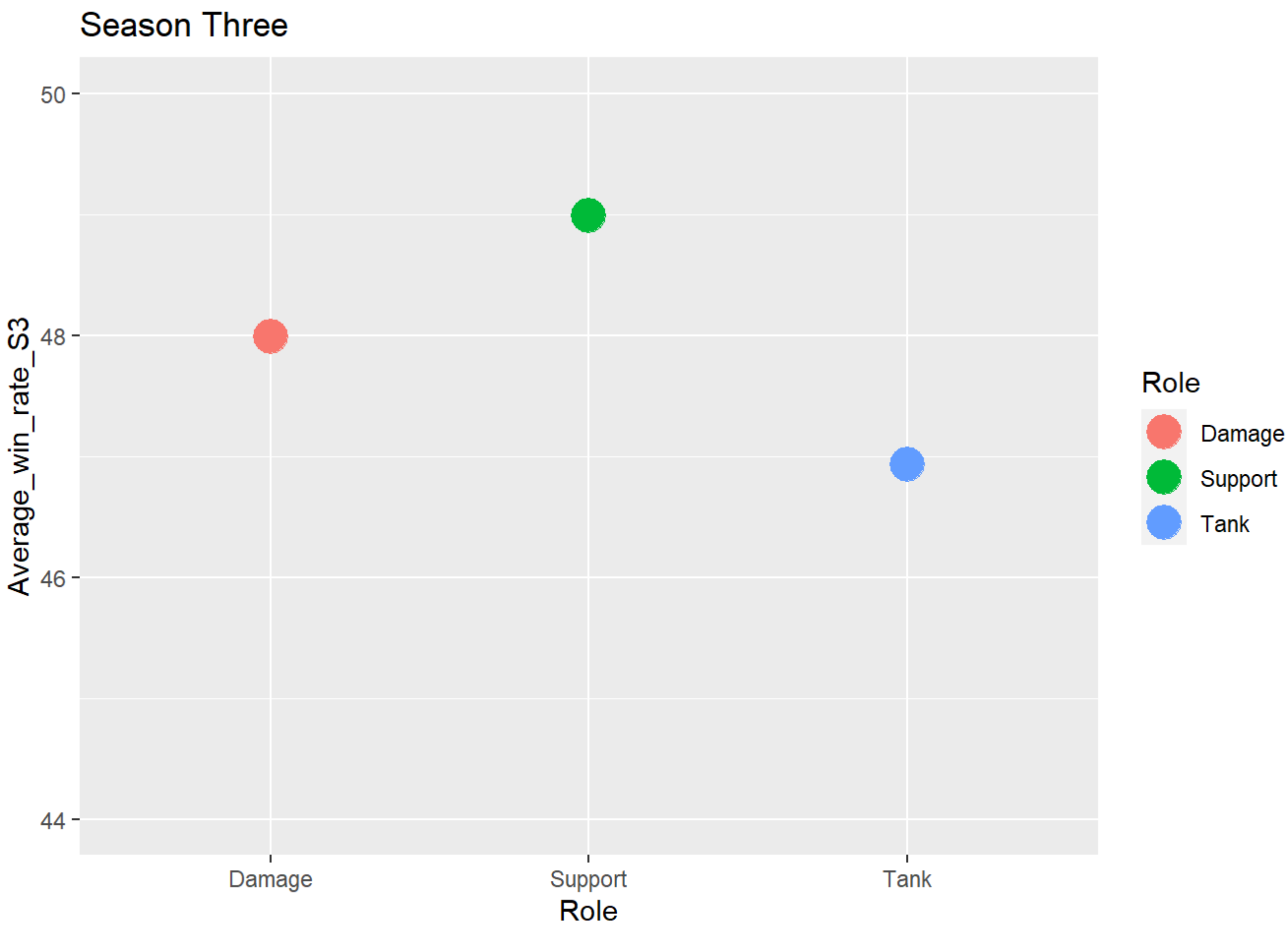
```
# Compare Season One Support to Damage than Support to Tank and Damage win rates were positive
# S1S - 46.82844
# S1T - 46.80025
# S1D - 46.19618

# Compare Season Two Support to Damage than Support to Tank and Damage win rates were positive
# S2S - 46.8925
# S2T - 46.47125
# S2D - 46.17471

# Compare Season Three Support to Damage than Support to Tank and Damage win rates were positive
# S3S - 48.98531
# S3T - 46.92761
# S3D - 47.98544

# Concludes that between all three season Support roles have greater mean win rates than Tank an
d Damage roles

#S1WinSplot <- ggplot() +
  #geom_hex(Season1Cleaned,
             #mapping=aes(x=Role, y=Win.Rate..., color=Role))
#S1WinSplot

S1 <- ggplot(SeasonalWinRole,
       aes(x=Role, y=Average_win_rate_S1, color=Role)) +
  geom_point(size=6) +
  ggtitle('Season One') +
  ylim(44,50)
S1
```



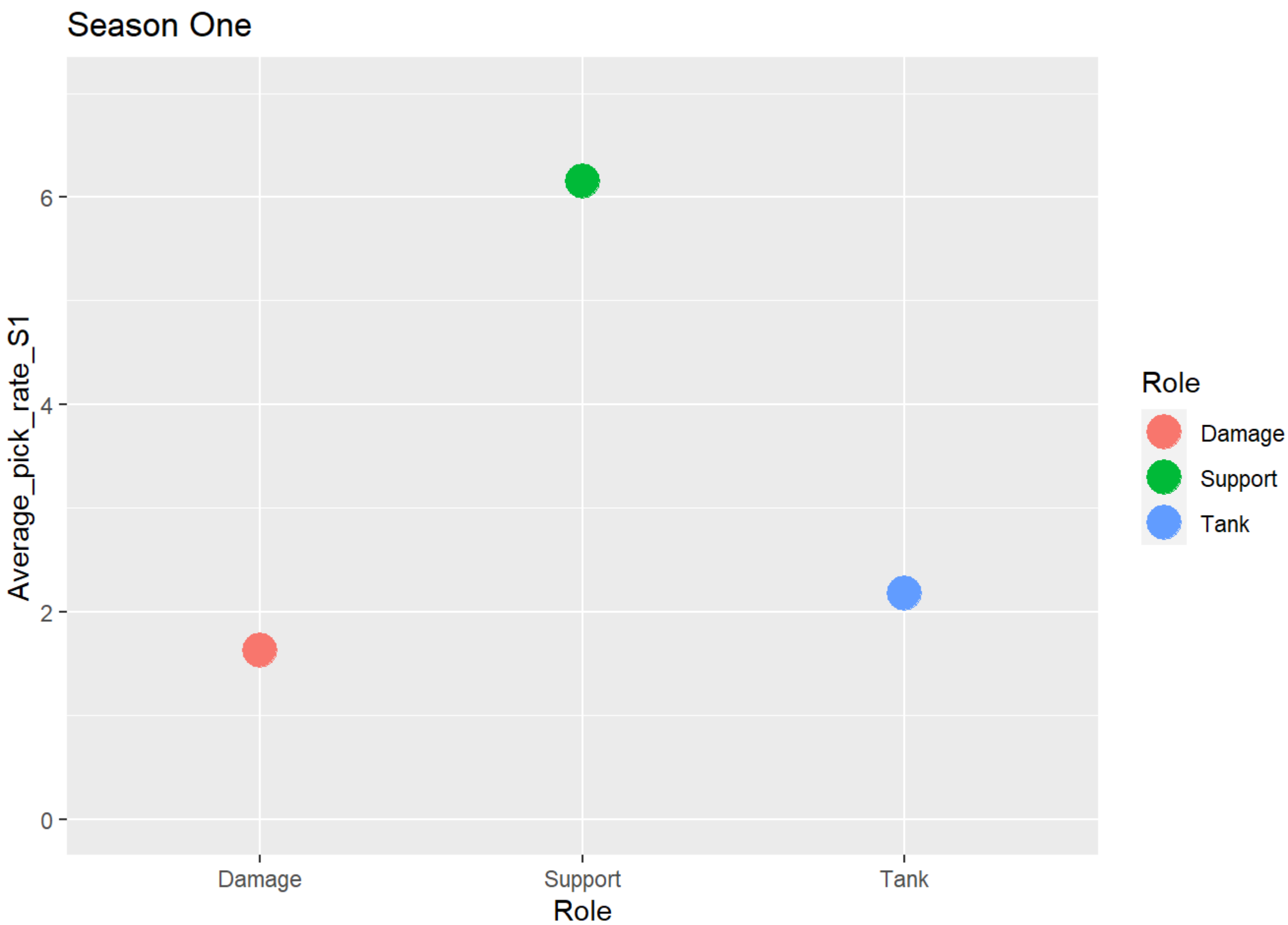

```
S2 <- ggplot(SeasonalWinRole,
             aes(x=Role, y=Average_win_rate_S2, color=Role)) +
  geom_point(size=6) +
  ggtitle('Season Two') +
  ylim(44,50)
S2
```



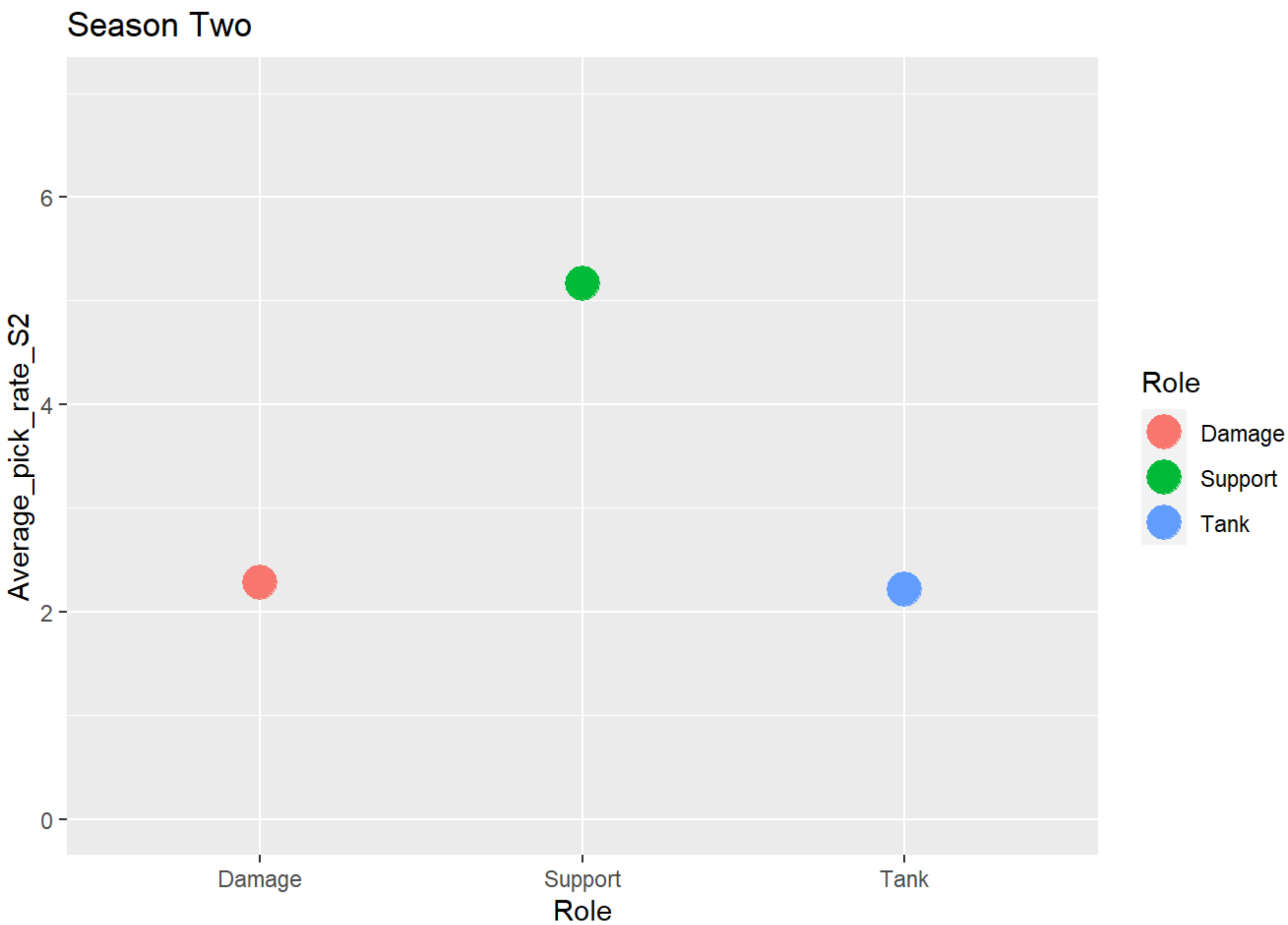

```
S3 <- ggplot(SeasonalWinRole,
             aes(x=Role, y=Average_win_rate_S3, color=Role)) +
  geom_point(size=6) +
  ggtitle('Season Three') +
  ylim(44,50)
S3
```




```
# Win Visual by Role
# WinRates <- grid.arrange(S1 + S2 + S3)
# WinRates
# Pick data fame
SeasonalPickRole <- SeasonalRoles |>
  select(Role, Average_pick_rate_S1, Average_pick_rate_S2, Average_pick_rate_S3)
View(SeasonalPickRole)

S1P <- ggplot(SeasonalPickRole,
             aes(x=Role, y=Average_pick_rate_S1, color=Role)) +
  geom_point(size=6) +
  ggtitle('Season One') +
  ylim(0,7)
S1P
```




```
S2P <- ggplot(SeasonalPickRole,
             aes(x=Role, y=Average_pick_rate_S2, color=Role)) +
  geom_point(size=6) +
  ggtitle('Season Two') +
  ylim(0,7)
S2P
```




```
S3P <- ggplot(SeasonalPickRole,
             aes(x=Role, y=Average_pick_rate_S3, color=Role)) +
  geom_point(size=6) +
  ggtitle('Season Three') +
  ylim(0,7)
# Pick Visual
# PickRates <- S1P + S2P + S3P
# PickRates
# Support has a higher average win rate and pick rate in all three seasons

# Count depreciation of pick rate
HeroCount <- Season3Cleaned |>
  group_by(Role) |>
  summarise(HeroCountByRole = unique(Hero))
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## ℹ Please use `reframe()` instead.
## ℹ When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'Role'. You can override using the
## `.groups` argument.
```

```
View(HeroCount)
# Out of the 36 hero selections: 17 are Damage, 11 Tank, and 8 Support
# Depreciate by pool size to show over inflated support pick rate
Role <- c("Damage","Tank","Support")
RoleNum <- c("17","11","8")
PickDeflator <- c("0.472","0.305","0.222")
Season_One <- c("0.769880616","1.876846170","0.4840155")
Season_Two <- c("1.077548152","1.574324295","0.49273904")
Season_Three <- c("1.4811081152","2.068948285","0.638123904")
Deflated_Pick <- data.frame(Role, PickDeflator, Season_One, Season_Two, Season_Three)
View(Deflated_Pick)
# Despite having the highest pick, when deflated for the pool in each role Support is picked the
least to a staggering degree.

# Conclusion of the first section is that a player selecting to play Support over the other two
roles will experience more success and less competition.
```

Play style and League Ranks

League Structure: A majority of the players moved up in the ranks each season, but interestingly the that lower ranks got thinner; perhaps player’s skill has grown, but fewer new players are participating.

Play style: Examining objective time/kill to win rate by role reveals the a play style where I expected Support to be near objectives. Interestingly Damage lost Objective kills to Tanks and Tanks also carried objective times.

Tank objective performance does, while having the lowest win and pick rates in all seasons, suggests objectives are not significant to overall wins and counter to my first impressions. This also means that the lower objective time of supports and damage means they are roaming outside the objectives. The latter might be due to roaming ahead and creating spaces for objective advance that are not recognized in the stats. Basically the players create a non-identified objective area around the effective objective area. Dueling is another option where players are seeking smaller engagements, but that would have the effect of creating barrier areas around objectives anyway.

Conclusion of this second section is that less new players are participating and in general the remaining players are improving. Additionally, support roles are not directly participating in objectives and objective participation does not directly increase win rates.

```
# Look at relationship between league ranks
# Rank ordering
RankOrdering <- data.frame(Skill.Tier = c('All','Bronze','Silver','Gold', 'Platinum','Diamon
d','Master','Grandmaster'),
                           Ranknum = c(1:8))
# Season1
Season1ST <- Season1Cleaned |>
  group_by(Skill.Tier) |>
  summarise(S1_Average_win_rate = mean(Win.Rate...),
            S1_Average_pick_rate = mean(Pick.Rate...),
            S1_Count_pick_rate = sum(Pick.Rate...))
View(Season1ST)
# Order by rank
Season1ST <- Season1ST |>
  left_join(RankOrdering,
            by = 'Skill.Tier')
Season1ST <- Season1ST[order(Season1ST$Ranknum),]
knitr::kable(Season1ST, format = "html")
```

Skill.Tier	S1_Average_win_rate	S1_Average_pick_rate	S1_Count_pick_rate	Ranknum
All	50.49600	2.826857	98.94	1
Bronze	43.47314	2.701143	94.54	2
Silver	46.61771	2.528000	88.48	3
Gold	48.48800	2.863714	100.23	4
Platinum	46.47486	2.864000	100.24	5
Diamond	45.55286	3.025429	105.89	6
Master	44.08886	2.977143	104.20	7
Grandmaster	42.34343	2.787429	97.56	8

```
# Season2
Season2ST <- Season2Cleaned |>
  group_by(Skill.Tier) |>
  summarise(S2_Average_win_rate = mean(Win.Rate...),
            S2_Average_pick_rate = mean(Pick.Rate...),
            S2_Count_pick_rate = sum(Pick.Rate...))
knitr::kable(Season2ST, format = "html")
```

Skill.Tier	S2_Average_win_rate	S2_Average_pick_rate	S2_Count_pick_rate
All	48.39028	2.762500	99.45
Bronze	41.99722	1.952500	70.29
Diamond	48.50222	3.738056	134.57
Gold	47.88361	2.916667	105.00
Grandmaster	43.36889	2.730556	98.30
Master	46.63722	3.420556	123.14
Platinum	49.16806	3.602778	129.70
Silver	45.45111	2.102500	75.69

```
# Order by rank
Season2ST <- Season2ST |>
  left_join(RankOrdering,
            by = 'Skill.Tier')
Season2ST <- Season2ST[order(Season2ST$Ranknum),]

# Season3
Season3ST <- Season3Cleaned |>
  group_by(Skill.Tier) |>
  summarise(S3_Average_win_rate = mean(Win.Rate...),
            S3_Average_pick_rate = mean(Pick.Rate...),
            S3_Count_pick_rate = sum(Pick.Rate...))
knitr::kable(Season3ST, format = "html")
```

Skill.Tier	S3_Average_win_rate	S3_Average_pick_rate	S3_Count_pick_rate
All	48.45528	2.755833	99.21
Bronze	43.85000	1.239722	44.63
Diamond	49.10250	5.770833	207.75
Gold	47.17417	1.938889	69.80
Grandmaster	50.70944	7.768333	279.66
Master	49.45250	6.604167	237.75
Platinum	48.34528	3.532500	127.17
Silver	45.98611	1.330000	47.88

```
# Order by rank
Season3ST <- Season3ST |>
  left_join(RankOrdering,
            by = 'Skill.Tier')
Season3ST <- Season3ST[order(Season3ST$Ranknum),]

# Bulk of the players move up in the ranks each season
# Unusual that lower ranks get thinner, perhaps player's skill has grown, but fewer new players
are participating
SeasonalRanks <- list(Season1ST, Season2ST, Season3ST)
SeasonalRanks |> reduce(full_join, by='Skill.Tier')
```

```
## # A tibble: 8 × 13
##   Skill.Tier  S1_Average_win_rate S1_Average_pick_rate S1_Count_pick_rate
##   <chr>                     <dbl>                <dbl>              <dbl>
## 1 All                        50.5                 2.83               98.9
## 2 Bronze                     43.5                 2.70               94.5
## 3 Silver                     46.6                 2.53               88.5
## 4 Gold                       48.5                 2.86              100. 
## 5 Platinum                   46.5                 2.86              100. 
## 6 Diamond                    45.6                 3.03              106. 
## 7 Master                     44.1                 2.98              104. 
## 8 Grandmaster                42.3                 2.79               97.6
## # ℹ 9 more variables: Ranknum.x <int>, S2_Average_win_rate <dbl>,
## #   S2_Average_pick_rate <dbl>, S2_Count_pick_rate <dbl>, Ranknum.y <int>,
## #   S3_Average_win_rate <dbl>, S3_Average_pick_rate <dbl>,
## #   S3_Count_pick_rate <dbl>, Ranknum <int>
```

```
SeasonalRanks <- data.frame(SeasonalRanks)
print(SeasonalRanks)
```

```
##    Skill.Tier S1_Average_win_rate S1_Average_pick_rate S1_Count_pick_rate
## 1         All            50.49600             2.826857              98.94
## 2      Bronze            43.47314             2.701143              94.54
## 3      Silver            46.61771             2.528000              88.48
## 4        Gold            48.48800             2.863714             100.23
## 5    Platinum            46.47486             2.864000             100.24
## 6     Diamond            45.55286             3.025429             105.89
## 7      Master            44.08886             2.977143             104.20
## 8 Grandmaster            42.34343             2.787429              97.56
##   Ranknum Skill.Tier.1 S2_Average_win_rate S2_Average_pick_rate
## 1       1          All            48.39028             2.762500
## 2       2       Bronze            41.99722             1.952500
## 3       3       Silver            45.45111             2.102500
## 4       4         Gold            47.88361             2.916667
## 5       5     Platinum            49.16806             3.602778
## 6       6      Diamond            48.50222             3.738056
## 7       7       Master            46.63722             3.420556
## 8       8  Grandmaster            43.36889             2.730556
##   S2_Count_pick_rate Ranknum.1 Skill.Tier.2 S3_Average_win_rate
## 1              99.45         1          All            48.45528
## 2              70.29         2       Bronze            43.85000
## 3              75.69         3       Silver            45.98611
## 4             105.00         4         Gold            47.17417
## 5             129.70         5     Platinum            48.34528
## 6             134.57         6      Diamond            49.10250
## 7             123.14         7       Master            49.45250
## 8              98.30         8  Grandmaster            50.70944
##   S3_Average_pick_rate S3_Count_pick_rate Ranknum.2
## 1             2.755833              99.21         1
## 2             1.239722              44.63         2
## 3             1.330000              47.88         3
## 4             1.938889              69.80         4
## 5             3.532500             127.17         5
## 6             5.770833             207.75         6
## 7             6.604167             237.75         7
## 8             7.768333             279.66         8
```

```
# look at objective time/kill to win rate by role
# Season1
Season1O <- Season1Cleaned |>
  group_by(Role) |>
  summarise(S1_Average_Objective_Time = mean(Objective.Time...10min),
            S1_Average_Objective_Kills = mean(Objective.Kills...10min))
Season2O <- Season2Cleaned |>
  group_by(Role) |>
  summarise(S2_Average_Objective_Time = mean(Objective.Time...10min),
            S2_Average_Objective_Kills = mean(Objective.Kills...10min))
Season3O <- Season3Cleaned |>
  group_by(Role) |>
  summarise(S3_Average_Objective_Time = mean(Objective.Time...10min),
            S3_Average_Objective_Kills = mean(Objective.Kills...10min))
# Interestingly Damage lost Objective kills to Tanks and Tanks also carried objective times
# Tank objective performance does, while having the lowest win and pick rates in all seasons, su
ggests objectives are not significant to overall wins
# This also means that the lower objective time of supports and damage means they are roaming ou
tside the objectives
# The latter might be due to roaming ahead and creating spaces for objective advance that are no
t recognized in the stats.
# Dueling is another option where players are seeking smaller engagements, but that would have t
he effect of creating barrier areas around objectives anyway.
```

Hero Selection:

Looking at top 10 performing heroes in all three seasons, Support was under represented in the earlier seasons.

Conclusion with the increased use of Support role heroes in following their performance suggests a slow recognition and adoption to incentive play styles. Within the top five support fo each season, three heroes repeated: Mercy, Lucio, Brigitte.

```
# dive into characters like pick and win rates top ten and their roles

# Season1
Top10HeroPicksS1 <- Season1Cleaned |>
  group_by(Hero) |>
  summarise(S1_Win_Rate = mean(Win.Rate...)) |>
  arrange(-S1_Win_Rate) |>
  head(10)
Top10HeroPicksS1$Role <- c('Damage', 'Damage','Damage','Damage','Tank','Support','Tank','Damag
e','Damage','Support')
knitr::kable(Top10HeroPicksS1, format = "html")
```

Hero	S1_Win_Rate	Role
Torbjorn	54.51125	Damage
Genji	54.08000	Damage
Junkrat	51.94625	Damage
Widowmaker	51.85750	Damage
D.Va	51.46625	Tank
Moira	51.34250	Support

Hero	S1_Win_Rate	Role
Reinhardt	50.62875	Tank
Ashe	48.20250	Damage
Sombra	48.08250	Damage
Lucio	48.04250	Support

```
# Not as strong a Support representation as I expected

# Season2
Top10HeroPicksS2 <- Season2Cleaned |>
  group_by(Hero) |>
  summarise(S2_Win_Rate = mean(Win.Rate...)) |>
  arrange(-S2_Win_Rate) |>
  head(10)
Top10HeroPicksS2$Role <- c('Damage','Damage','Tank','Tank','Damage','Support','Support','Tan
k','Support','Tank')
knitr::kable(Top10HeroPicksS2, format = "html")
```

Hero	S2_Win_Rate	Role
Torbjorn	52.22500	Damage
Symmetra	52.13000	Damage
Ramattra	50.22125	Tank
Reinhardt	49.64000	Tank
Pharah	49.33750	Damage
Ana	49.21625	Support
Zenyatta	48.76125	Support
Doomfist	48.59125	Tank
Lucio	48.25750	Support
Genji	47.87750	Tank

```
# Still an even distribution

# Season3
Top10HeroPicksS3 <- Season3Cleaned |>
  group_by(Hero) |>
  summarise(S3_Win_Rate = mean(Win.Rate...)) |>
  arrange(-S3_Win_Rate) |>
  head(10)
Top10HeroPicksS3$Role <- c('Suport','Tank','Support','Damage','Tank','Support','Tank','Suppor
t','Damage','Support')
knitr::kable(Top10HeroPicksS3, format = "html")
```

Hero	S3_Win_Rate	Role
Symmetra	53.64750	Suport
Reinhardt	53.43875	Tank
Brigitte	52.73375	Support
Torbjorn	52.64000	Damage
Wrecking Ball	51.42500	Tank
Zenyatta	51.01875	Support
Sigma	49.96250	Tank
Lucio	49.94875	Support

Hero	S3_Win_Rate	Role
Mei	49.68125	Damage
Ana	49.36750	Support

```
Season_One_role <- c("6","2","2")
Season_Two_role <- c("3","4","3")
Season_Three_role <- c("2","3","5")
Top_Ten_Trend <- data.frame(Role, Season_One_role, Season_Two_role, Season_Three_role)
knitr::kable(Top_Ten_Trend, format = "html")
```

Role	Season_One_role	Season_Two_role	Season_Three_role
Damage	6	3	2
Tank	2	4	3
Support	2	3	5

```
# Finally support influenced, but not as heavily as I expected
# This may be a delay in the player base to adapt to Support win rates

# Top Support Heroes
# Season1
Top5HeroPicksS1Support <- Season1Cleaned |>
  group_by(Hero) |>
  filter(Role == 'Support') |>
  summarise(S1_Win_Rate = mean(Win.Rate...)) |>
  arrange(-S1_Win_Rate) |>
  head(5)
knitr::kable(Top5HeroPicksS1Support, format = "html")
```

Hero	S1_Win_Rate
Moira	51.34250
Lucio	48.04250
Kiriko	47.87125
Mercy	47.33625
Brigitte	46.26375

```
# Season2
Top5HeroPicksS2Support <- Season2Cleaned |>
  group_by(Hero) |>
  filter(Role == 'Support') |>
  summarise(S2_Win_Rate = mean(Win.Rate...)) |>
  arrange(-S2_Win_Rate) |>
  head(5)
knitr::kable(Top5HeroPicksS2Support, format = "html")
```

Hero	S2_Win_Rate
Ana	49.21625
Zenyatta	48.76125
Lucio	48.25750
Mercy	47.30875
Brigitte	46.03250

```
# Season3
Top5HeroPicksS3Support <- Season3Cleaned |>
  group_by(Hero) |>
  filter(Role == 'Support') |>
  summarise(S3_Win_Rate = mean(Win.Rate...)) |>
  arrange(-S3_Win_Rate) |>
  head(5)
knitr::kable(Top5HeroPicksS3Support, format = "html")
```

Hero	S3_Win_Rate
Brigitte	52.73375
Zenyatta	51.01875
Lucio	49.94875
Ana	49.36750
Mercy	48.95875

```
# Within the top five support fo each season, three heroes repeated: Mercy, Lucio, Brigitte.
```

In final conclusion, to win more in Overwatch 2 and have a more successful tome climbing the league ranks: Play support Stay off the objectives Play specifically heroes Mercy, Lucio, Brigitte