Analisis de patrones conductuales vinculados al rendimiento deportivo

Rosalia

2025-08-27

```
paquetes_necesarios <- c(
  "flexdashboard", "tidyverse", "readxl", "readr", "dplyr", "ggplot2",
  "ggrepel", "countrycode", "scales", "maps", "DT", "lorem", "png", "grid"
)

instalar_si_falta <- function(pkg) {
  if (!requireNamespace(pkg, quietly = TRUE)) {
    install.packages(pkg)
  }
}
message("Todos los paquetes necesarios están instalados.")
```

```
## Todos los paquetes necesarios están instalados.
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'tibble' was built under R version 4.4.3
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
## Warning: package 'readr' was built under R version 4.4.3
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
## Warning: package 'stringr' was built under R version 4.4.3
```

```
## Warning: package 'forcats' was built under R version 4.4.3
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readxl)
library(readr)
library(dplyr)
library(ggplot2)
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.4.3
```

```
library(countrycode)
```

```
## Warning: package 'countrycode' was built under R version 4.4.3
```

```
library(scales)
```

```
## Warning: package 'scales' was built under R version 4.4.3
```

```
##
## Adjuntando el paquete: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 4.4.3
```

```
##
## Adjuntando el paquete: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```r
library(DT)
```

```
## Warning: package 'DT' was built under R version 4.4.3
```

```r
library(lorem)
```

```
## Warning: package 'lorem' was built under R version 4.4.3
```

```r
library(png)
library(grid)
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.4.3
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.2
```

```
## corrplot 0.95 loaded
```

```r
library(here)
```

```
## Warning: package 'here' was built under R version 4.4.3
```

```
## here() starts at C:/Users/Rosalía Carballo/OneDrive/Documentos/Prueba_Trabajo_final_maestria
```

```r
library(stringr)
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.4.3
```

```
##
## Adjuntando el paquete: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(lubridate)
library(purrr)
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.4.3
```

```r
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.4.3
```

```r
getwd()
```

```
## [1] "C:/Users/Rosalía Carballo/OneDrive/Documentos/Prueba_Trabajo_final_maestria"
```

```r
Datos_APILOL <- read_csv ("C:/Users/Rosalía Carballo/OneDrive/Documentos/Prueba_Trabajo_final_maestria/
```

```
## Rows: 15000 Columns: 42
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (7): puuid, gameName, tagLine, tier, match_id, championName, individua...
## dbl  (33): leaguePoints, gameDuration, champExperience, champLevel, goldEarn...
## dttm  (1): datetime
## date  (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
Datos <- Datos_APILOL
```

```r
#Depuración de datos

glimpse(Datos)
```

```
## Rows: 15,000
## Columns: 42
## $ puuid                          <chr> "-9yJwHGHfPMq-whpYDPlbN1FwdIGP0ONpwccJS~
## $ gameName                       <chr> "Creepy Nuts", "Creepy Nuts", "Creepy N~
## $ date                           <date> 2025-07-14, 2025-07-12, 2025-07-11, 20~
## $ datetime                       <dttm> 2025-07-14 16:08:07, 2025-07-12 17:50:~
## $ tagLine                        <chr> "042", "042", "042", "042", "042", "042~
## $ tier                           <chr> "MASTER", "MASTER", "MASTER", "MASTER",~
## $ leaguePoints                   <dbl> 198, 198, 198, 198, 198, 198, 198, 198,~
## $ match_id                       <chr> "EUW1_7461274213", "EUW1_7459385550", "~
## $ gameDuration                   <dbl> 1730, 2219, 1729, 1119, 1703, 1775, 188~
## $ championName                   <chr> "Syndra", "Sivir", "Syndra", "Syndra", ~
## $ individualPosition             <chr> "MIDDLE", "BOTTOM", "MIDDLE", "MIDDLE",~
## $ champExperience                <dbl> 14944, 18534, 17525, 10737, 17553, 1576~
## $ champLevel                     <dbl> 16, 18, 17, 13, 17, 16, 18, 10, 11, 17,~
## $ goldEarned                     <dbl> 13622, 17354, 15139, 8732, 11757, 12036~
## $ totalMinionsKilled             <dbl> 198, 324, 255, 139, 193, 180, 260, 109,~
## $ kills                          <dbl> 8, 8, 8, 6, 8, 6, 7, 0, 0, 7, 1, 14, 6,~
## $ deaths                         <dbl> 4, 5, 1, 4, 5, 2, 4, 6, 4, 3, 2, 2, 1, ~
## $ assists                        <dbl> 9, 2, 13, 7, 10, 14, 9, 1, 3, 5, 2, 5, ~
## $ totalDamageDealt               <dbl> 144540, 296885, 175650, 83039, 120850, ~
## $ totalDamageTaken               <dbl> 16053, 22086, 19526, 8683, 40922, 14968~
## $ damageDealtToBuildings         <dbl> 2329, 4228, 2136, 184, 870, 2532, 4979,~
## $ damageDealtToObjectives        <dbl> 8262, 15721, 3822, 184, 870, 9558, 1087~
## $ damageDealtToTurrets           <dbl> 2329, 4228, 2136, 184, 870, 2532, 4979,~
## $ visionScore                    <dbl> 16, 18, 15, 10, 15, 13, 5, 3, 8, 10, 5,~
## $ baron_kills                    <dbl> 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, ~
## $ dragon_kills                   <dbl> 3, 0, 0, 2, 4, 1, 2, 1, 1, 1, 2, 3, 4, ~
```

```
## $ tower_kills                    <dbl> 9, 6, 7, 3, 11, 8, 11, 0, 0, 11, 1, 8, ~
## $ herald_kills                   <dbl> 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, ~
## $ inhibitor_kills                <dbl> 1, 1, 1, 0, 3, 1, 3, 0, 0, 3, 0, 1, 1, ~
## $ champion_kills                 <dbl> 30, 24, 35, 23, 43, 29, 42, 4, 8, 29, 1~
## $ win                            <dbl> 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, ~
## $ longestTimeSpentLiving         <dbl> 805, 804, 1241, 274, 576, 924, 682, 370~
## $ timeCCingOthers                <dbl> 55, 3, 33, 26, 37, 28, 60, 20, 18, 31, ~
## $ timePlayed                     <dbl> 1730, 2219, 1729, 1119, 1703, 1775, 188~
## $ totalTimeCCDealt               <dbl> 427, 58, 421, 288, 400, 375, 527, 207, ~
## $ totalTimeSpentDead             <dbl> 155, 122, 44, 66, 141, 78, 167, 108, 73~
## $ totalAllyJungleMinionsKilled   <dbl> 0, 0, 0, 0, 0, 4, 0, 2, 0, 0, 0, 0, 0, ~
## $ totalEnemyJungleMinionsKilled  <dbl> 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, ~
## $ totalHeal                      <dbl> 2501, 5139, 4431, 916, 15214, 13096, 23~
## $ totalHealsOnTeammates          <dbl> 0, 0, 0, 0, 0, 5491, 0, 0, 0, 0, 0, 0, ~
## $ totalUnitsHealed               <dbl> 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, ~
## $ totalDamageShieldedOnTeammates <dbl> 0, 0, 0, 0, 0, 1703, 0, 0, 0, 0, 0, 0, ~
```

```
summary(Datos)
```

```
##     puuid              gameName              date
##  Length:15000       Length:15000       Min.   :2025-02-16
##  Class :character   Class :character   1st Qu.:2025-06-26
##  Mode  :character   Mode  :character   Median :2025-07-06
##                                        Mean   :2025-07-02
##                                        3rd Qu.:2025-07-12
##                                        Max.   :2025-07-17
##     datetime                       tagLine              tier
##  Min.   :2025-02-16 11:58:28.00   Length:15000       Length:15000
##  1st Qu.:2025-06-26 19:01:45.50   Class :character   Class :character
##  Median :2025-07-06 20:23:37.00   Mode  :character   Mode  :character
##  Mean   :2025-07-03 03:53:05.44
##  3rd Qu.:2025-07-12 13:27:38.00
##  Max.   :2025-07-17 08:24:00.00
##   leaguePoints      match_id          gameDuration  championName
##  Min.   :  17.0   Length:15000       Min.   : 100   Length:15000
##  1st Qu.: 191.0   Class :character   1st Qu.:1413   Class :character
##  Median : 405.0   Mode  :character   Median :1652   Mode  :character
##  Mean   : 592.4                      Mean   :1616
##  3rd Qu.: 951.2                      3rd Qu.:1837
##  Max.   :2157.0                      Max.   :3539
##  individualPosition champExperience   champLevel      goldEarned
##  Length:15000       Min.   :    0   Min.   : 1.00   Min.   :  504
##  Class :character   1st Qu.:10102   1st Qu.:13.00   1st Qu.: 8302
##  Mode  :character   Median :13267   Median :15.00   Median :10838
##                     Mean   :13186   Mean   :14.24   Mean   :10915
##                     3rd Qu.:16286   3rd Qu.:16.00   3rd Qu.:13355
##                     Max.   :35782   Max.   :18.00   Max.   :29048
##  totalMinionsKilled     kills            deaths          assists
##  Min.   :  0.0      Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 26.0      1st Qu.: 2.000   1st Qu.: 3.000   1st Qu.: 4.000
##  Median :140.5      Median : 4.000   Median : 5.000   Median : 7.000
##  Mean   :123.8      Mean   : 5.216   Mean   : 5.025   Mean   : 8.228
##  3rd Qu.:206.0      3rd Qu.: 8.000   3rd Qu.: 7.000   3rd Qu.:11.000
##  Max.   :439.0      Max.   :31.000   Max.   :18.000   Max.   :43.000
```

```
##  totalDamageDealt totalDamageTaken damageDealtToBuildings
##  Min.   :     0   Min.   :     0   Min.   :    0.0
##  1st Qu.: 64109   1st Qu.: 16306   1st Qu.:  462.8
##  Median :135921   Median : 23192   Median : 2096.5
##  Mean   :140444   Mean   : 25397   Mean   : 3209.3
##  3rd Qu.:197724   3rd Qu.: 32375   3rd Qu.: 4863.0
##  Max.   :646165   Max.   :171024   Max.   :36101.0
##  damageDealtToObjectives damageDealtToTurrets  visionScore      baron_kills
##  Min.   :    0           Min.   :    0.0      Min.   :  0.00   Min.   :0.0000
##  1st Qu.: 2118           1st Qu.:  462.8      1st Qu.: 13.00   1st Qu.:0.0000
##  Median : 6357           Median : 2096.5      Median : 20.00   Median :0.0000
##  Mean   :11933           Mean   : 3209.3      Mean   : 28.78   Mean   :0.3432
##  3rd Qu.:16027           3rd Qu.: 4863.0      3rd Qu.: 32.00   3rd Qu.:1.0000
##  Max.   :95367           Max.   :36101.0      Max.   :209.00   Max.   :4.0000
##   dragon_kills    tower_kills      herald_kills    inhibitor_kills
##  Min.   :0.000   Min.   : 0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.: 3.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :2.000   Median : 5.000   Median :0.0000   Median :0.0000
##  Mean   :1.822   Mean   : 5.669   Mean   :0.4685   Mean   :0.7706
##  3rd Qu.:3.000   3rd Qu.: 9.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :7.000   Max.   :15.000   Max.   :1.0000   Max.   :9.0000
##  champion_kills       win         longestTimeSpentLiving timeCCingOthers
##  Min.   : 0.00   Min.   :0.0000   Min.   :   0.0         Min.   :  0.00
##  1st Qu.:18.00   1st Qu.:0.0000   1st Qu.: 393.0         1st Qu.: 11.00
##  Median :27.00   Median :1.0000   Median : 539.0         Median : 21.00
##  Mean   :26.58   Mean   :0.5368   Mean   : 580.7         Mean   : 24.77
##  3rd Qu.:35.00   3rd Qu.:1.0000   3rd Qu.: 729.0         3rd Qu.: 34.00
##  Max.   :77.00   Max.   :1.0000   Max.   :2386.0         Max.   :318.00
##    timePlayed   totalTimeCCDealt totalTimeSpentDead
##  Min.   : 100   Min.   :   0     Min.   :  0.0
##  1st Qu.:1413   1st Qu.: 105     1st Qu.: 69.0
##  Median :1652   Median : 191     Median :130.0
##  Mean   :1616   Mean   : 255     Mean   :142.3
##  3rd Qu.:1837   3rd Qu.: 326     3rd Qu.:201.0
##  Max.   :3539   Max.   :3311     Max.   :699.0
##  totalAllyJungleMinionsKilled totalEnemyJungleMinionsKilled   totalHeal
##  Min.   :  0.00               Min.   : 0.000                Min.   :     0
##  1st Qu.:  0.00               1st Qu.: 0.000                1st Qu.:  2839
##  Median :  0.00               Median : 0.000                Median :  5774
##  Mean   : 24.27               Mean   : 4.107                Mean   :  9154
##  3rd Qu.:  8.00               3rd Qu.: 4.000                3rd Qu.: 12568
##  Max.   :204.00               Max.   :79.000                Max.   :119194
##  totalHealsOnTeammates totalUnitsHealed totalDamageShieldedOnTeammates
##  Min.   :    0.0       Min.   : 0.000   Min.   :    0
##  1st Qu.:    0.0       1st Qu.: 1.000   1st Qu.:    0
##  Median :    0.0       Median : 1.000   Median :    0
##  Mean   :  660.1       Mean   : 1.852   Mean   :  636
##  3rd Qu.:    0.0       3rd Qu.: 1.000   3rd Qu.:    0
##  Max.   :37545.0       Max.   :29.000   Max.   :38498
```

```
cols_key   <- c("match_id","puuid")
cols_num   <- c("leaguePoints","gameDuration","champExperience","champLevel",
                "goldEarned","totalMinionsKilled","kills","deaths","assists",
                "totalDamageDealt","totalDamageTaken","damageDealtToBuildings",
```

```
                "damageDealtToObjectives","damageDealtToTurrets","visionScore",
                "baron_kills","dragon_kills","tower_kills","herald_kills",
                "inhibitor_kills","champion_kills","longestTimeSpentLiving",
                "timeCCingOthers","timePlayed","totalTimeCCDealt",
                "totalTimeSpentDead","totalAllyJungleMinionsKilled",
                "totalEnemyJungleMinionsKilled","totalHeal","totalHealsOnTeammates",
                "totalUnitsHealed","totalDamageShieldedOnTeammates")
```

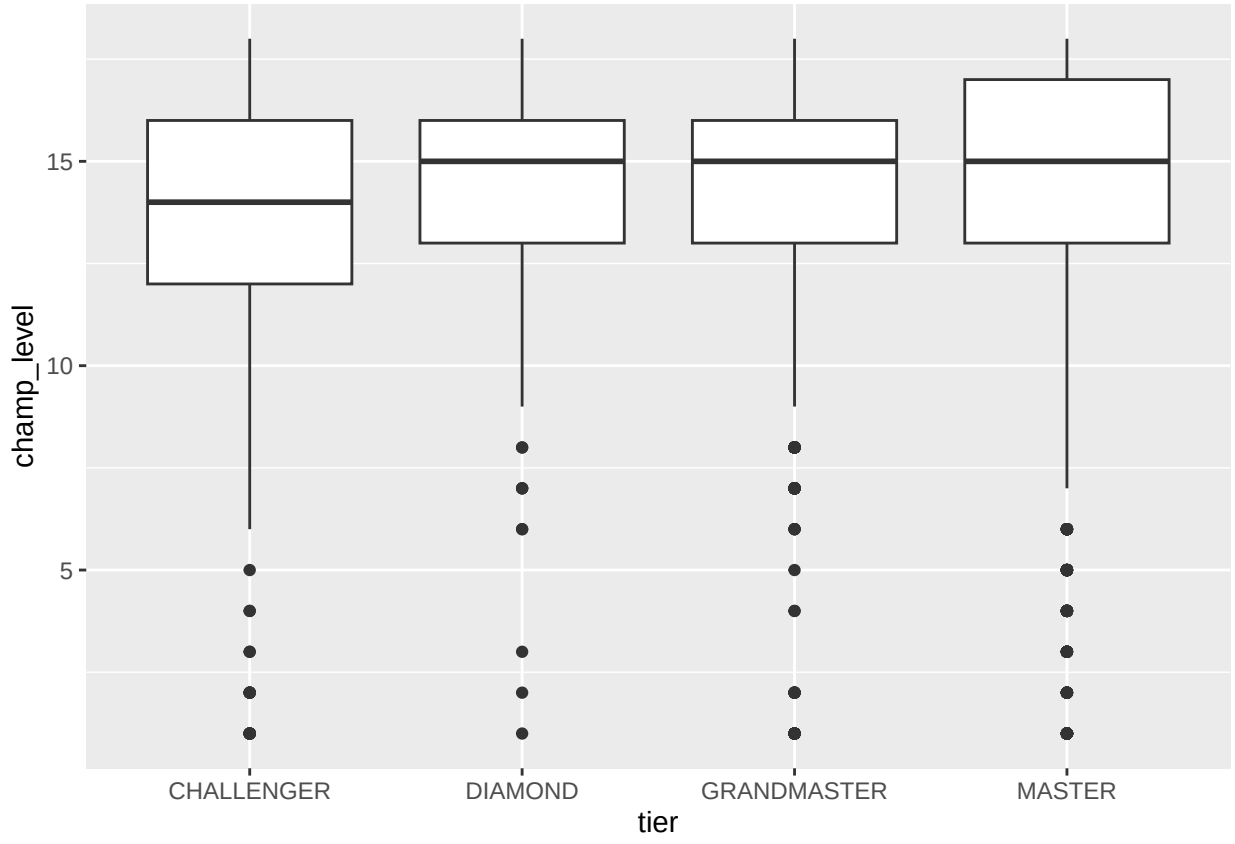
```
cols_chr   <- c("gameName","tagLine","tier","individualPosition","championName")
cols_date  <- c("date")       # YYYY-MM-DD ?
cols_dttm  <- c("datetime")   # ISO 8601 ?

# Limpieza rápida (espacios y tipos básicos)
dfc <- Datos %>%
  mutate(across(all_of(cols_chr), ~str_squish(as.character(.)))) %>%
  mutate(across(all_of(cols_num), ~suppressWarnings(as.numeric(.)))) %>%
  mutate(across(all_of(cols_date), ~suppressWarnings(as.Date(.)))) %>%
  mutate(across(all_of(cols_dttm), ~suppressWarnings(lubridate::ymd_hms(.)))) %>%
  clean_names()
```

```
colSums(is.na(dfc))
```

```
##                             puuid                         game_name
##                                 0                                 0
##                              date                          datetime
##                                 0                                 0
##                          tag_line                              tier
##                                 0                                 0
##                      league_points                          match_id
##                                 0                                 0
##                      game_duration                     champion_name
##                                 0                                 0
##                 individual_position                  champ_experience
##                                 0                                 0
##                        champ_level                       gold_earned
##                                 0                                 0
##                total_minions_killed                             kills
##                                 0                                 0
##                              deaths                           assists
##                                 0                                 0
##                  total_damage_dealt                 total_damage_taken
##                                 0                                 0
##           damage_dealt_to_buildings         damage_dealt_to_objectives
##                                 0                                 0
##             damage_dealt_to_turrets                      vision_score
##                                 0                                 0
##                         baron_kills                      dragon_kills
##                                 0                                 0
##                         tower_kills                      herald_kills
##                                 0                                 0
##                     inhibitor_kills                    champion_kills
##                                 0                                 0
##                                 win           longest_time_spent_living
```

```
##                                0                                0
##                time_c_cing_others                      time_played
##                                0                                0
##               total_time_cc_dealt             total_time_spent_dead
##                                0                                0
##  total_ally_jungle_minions_killed total_enemy_jungle_minions_killed
##                                0                                0
##                        total_heal          total_heals_on_teammates
##                                0                                0
##                total_units_healed total_damage_shielded_on_teammates
##                                0                                0
```

```
dfc |>
  ggplot(aes(x = tier, y = champ_level)) +
  geom_boxplot()
```




```
dfc |>
  summarise(across(where(is.character), ~ n_distinct(.x)))
```

```
## # A tibble: 1 x 7
##   puuid game_name tag_line  tier match_id champion_name individual_position
##   <int>     <int>    <int> <int>    <int>         <int>                <int>
## 1   277       291      205     4    12058           171                    6
```

```r
unique(dfc$tier)
```

```
## [1] "MASTER"      "DIAMOND"     "GRANDMASTER" "CHALLENGER"
```

```r
unique(dfc$champion_name)
```

```
##   [1] "Syndra"       "Sivir"        "Sylas"        "Senna"        "Kennen"
##   [6] "Lucian"       "Kayle"        "Tristana"     "Cassiopeia"   "Jayce"
##  [11] "Malphite"     "Yasuo"        "Corki"        "Fizz"         "Ekko"
##  [16] "Rell"         "Karma"        "Milio"        "Nami"         "Seraphine"
##  [21] "Nautilus"     "Alistar"      "Lulu"         "Renata"       "Sona"
##  [26] "LeeSin"       "Zed"          "Diana"        "Thresh"       "Janna"
##  [31] "Leona"        "Poppy"        "Rakan"        "Braum"        "Lillia"
##  [36] "Sejuani"      "Teemo"        "MonkeyKing"   "Neeko"        "FiddleSticks"
##  [41] "Elise"        "JarvanIV"     "Naafiri"      "Kassadin"     "Vladimir"
##  [46] "KSante"       "Ambessa"      "Jax"          "Viego"        "Skarner"
##  [51] "Vayne"        "Viktor"       "Annie"        "Aurora"       "Ahri"
##  [56] "Galio"        "Taliyah"      "Maokai"       "Yuumi"        "Blitzcrank"
##  [61] "Pantheon"     "Shen"         "Brand"        "Soraka"       "Bard"
##  [66] "TahmKench"    "Morgana"      "Zilean"       "Mel"          "Pyke"
##  [71] "Urgot"        "Samira"       "Jhin"         "Smolder"      "Ezreal"
##  [76] "Kaisa"        "Rengar"       "Gwen"         "Graves"       "Kindred"
##  [81] "Vi"           "Trundle"      "Talon"        "XinZhao"      "Volibear"
##  [86] "Nunu"         "Gragas"       "DrMundo"      "Nasus"        "Zeri"
##  [91] "Ziggs"        "Xayah"        "Jinx"         "Gnar"         "Aphelios"
##  [96] "Caitlyn"      "MissFortune"  "Varus"        "Twitch"       "Ornn"
## [101] "Quinn"        "Karthus"      "Kalista"      "Orianna"      "Hwei"
## [106] "Belveth"      "Aatrox"       "Yone"         "Renekton"     "Sett"
## [111] "Lissandra"    "Zyra"         "Malzahar"     "Vex"          "Ivern"
## [116] "Shaco"        "Camille"      "Ryze"         "Swain"        "Azir"
## [121] "AurelionSol"  "Mordekaiser"  "Shyvana"      "Fiora"        "Olaf"
## [126] "Veigar"       "TwistedFate"  "Akali"        "Irelia"       "Qiyana"
## [131] "Ashe"         "Draven"       "Sion"         "Anivia"       "Leblanc"
## [136] "RekSai"       "Chogath"      "Lux"          "MasterYi"     "Evelynn"
## [141] "Kayn"         "Zac"          "Zoe"          "Briar"        "Yorick"
## [146] "Rumble"       "KogMaw"       "Nocturne"     "Velkoz"       "Taric"
## [151] "Nidalee"      "Hecarim"      "Singed"       "Warwick"      "Katarina"
## [156] "Riven"        "Darius"       "Garen"        "Gangplank"    "Tryndamere"
## [161] "Udyr"         "Khazix"       "Xerath"       "Illaoi"       "Amumu"
## [166] "Nilah"        "Heimerdinger" "Yunara"       "Kled"         "Rammus"
## [171] "Akshan"
```

Objetivo 1:

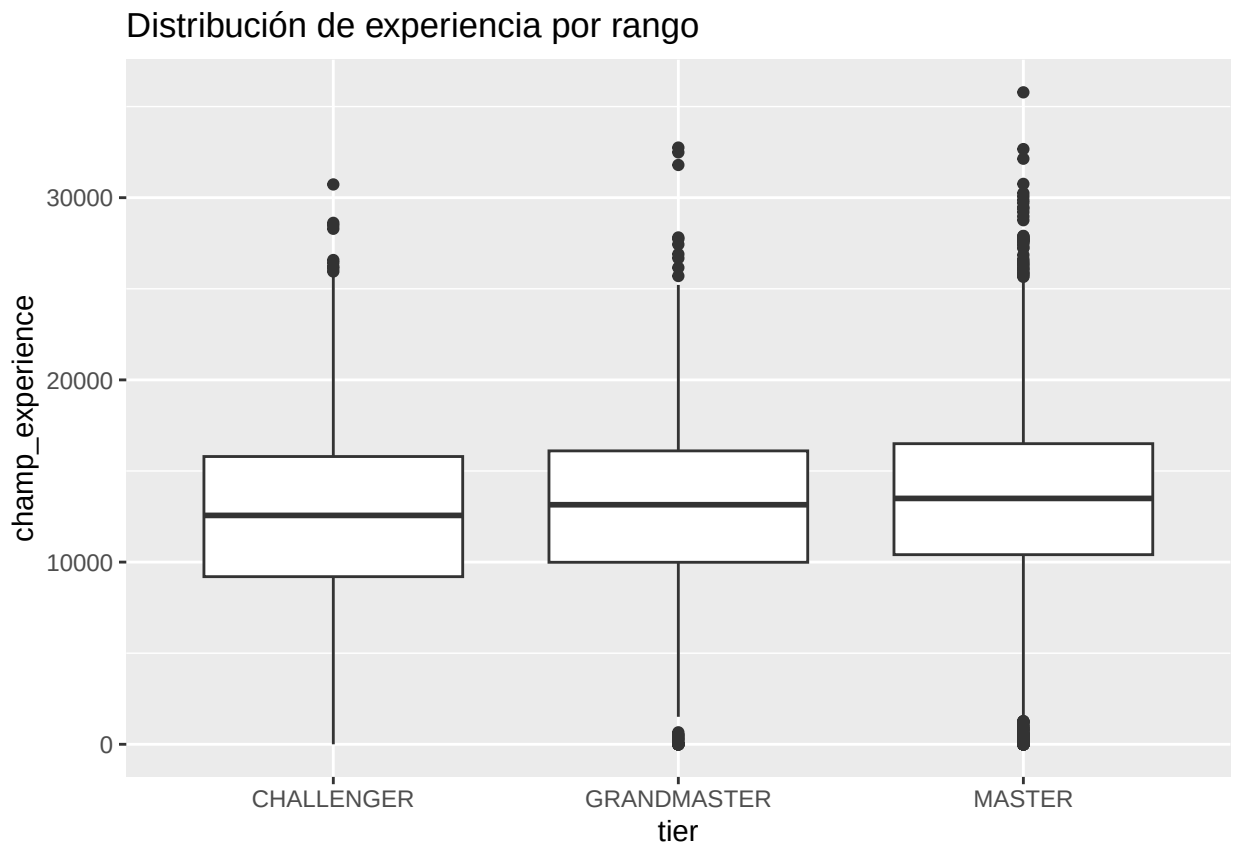
Definir el perfil y las características de los jugadores de rango Maestro, Gran Maestro y Challenger en League of Legends, considerando variables como experiencia, frecuencia de juego, rol preferido y el tipo de rol desempeñado dentro de las partidas, con el fin de caracterizar los estilos de juego asociados a cada categoría.

```r
#filtrarlos según Tier(rango alto)

datos_rango_alto <- dfc |>
```

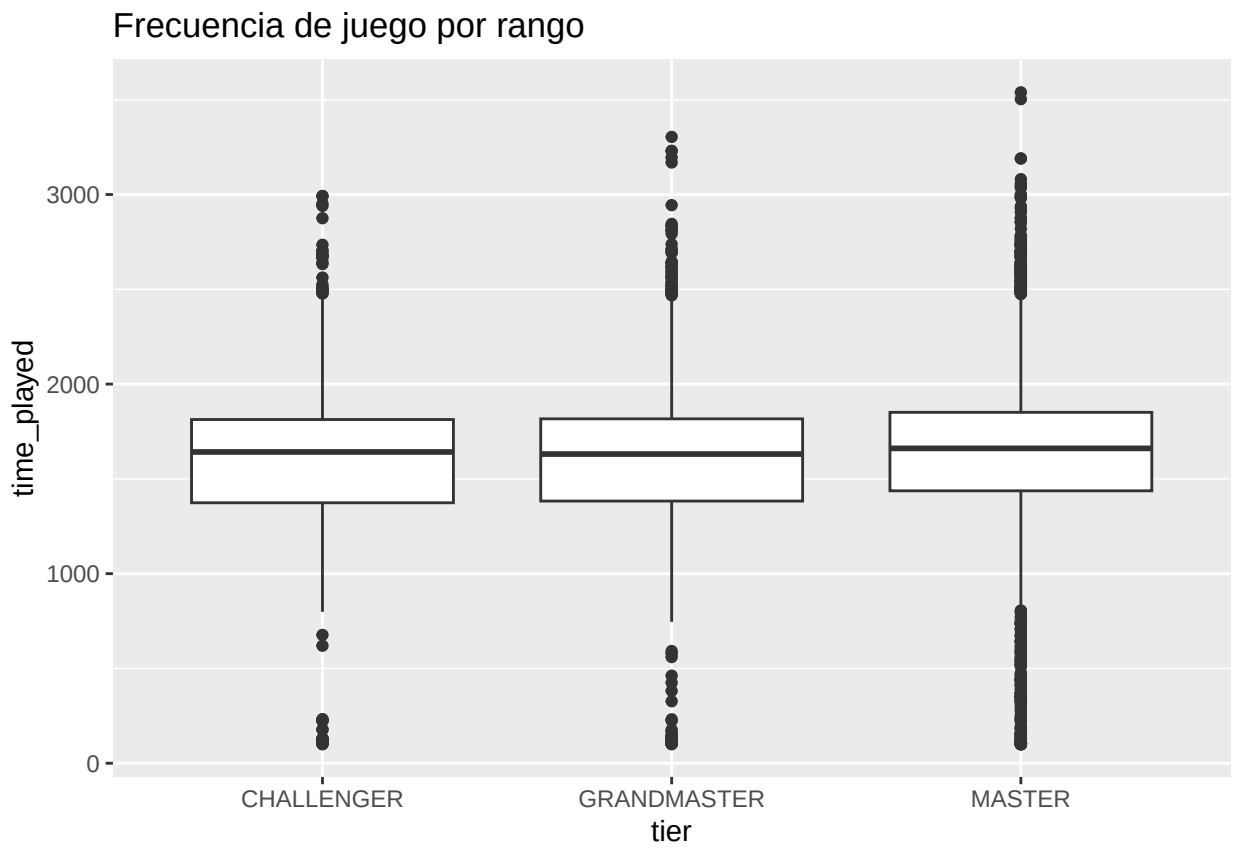
```r
  filter(tier %in% c("MASTER", "GRANDMASTER", "CHALLENGER"))

#Resumen Experiencia y frecuencia del juego

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    promedio_experiencia = mean(champ_experience, na.rm = TRUE),
    sd_experiencia = sd(champ_experience, na.rm = TRUE),
    promedio_Tiempojugado = mean(time_played, na.rm = TRUE),
    sd_Tiempojugado = sd(time_played, na.rm = TRUE)
  )
```

```
## # A tibble: 3 x 5
##   tier        promedio_experiencia sd_experiencia promedio_Tiempojugado
##   <chr>                      <dbl>          <dbl>                 <dbl>
## 1 CHALLENGER                12568.          4516.                 1595.
## 2 GRANDMASTER               13028.          4424.                 1593.
## 3 MASTER                    13429.          4563.                 1631.
## # i 1 more variable: sd_Tiempojugado <dbl>
```

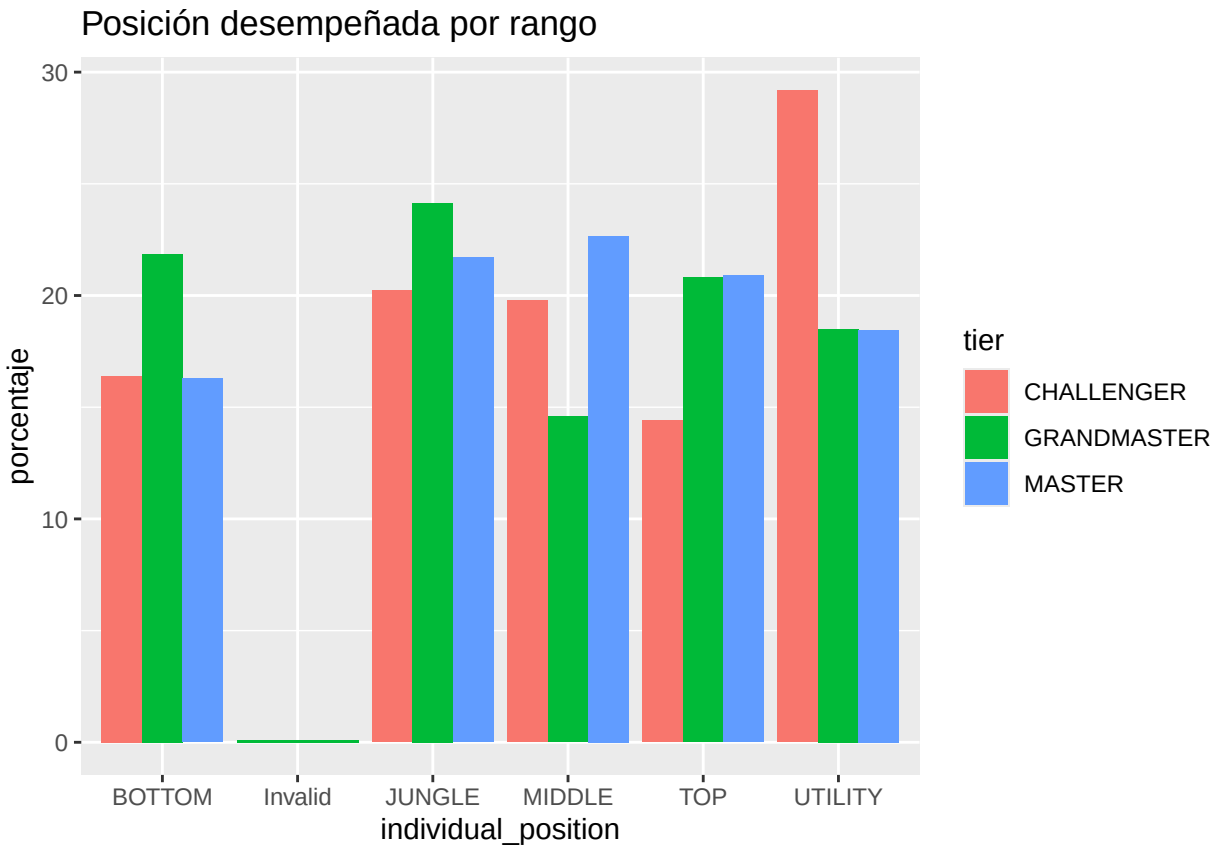
```r
#Visualización de Experiencia y frecuencia del Juego

datos_rango_alto |>
  ggplot(aes(x = tier, y = champ_experience)) +
  geom_boxplot() +
  labs(title = "Distribución de experiencia por rango")
```




```
datos_rango_alto |>
  ggplot(aes(x = tier, y = time_played)) +
  geom_boxplot() +
  labs(title = "Frecuencia de juego por rango")
```




```
#Resumen Posición desempeñada en partida
datos_rango_alto |>
  count(tier, individual_position) |>
  group_by(tier) |>
  mutate(porcentaje = n / sum(n) * 100) |>
  ggplot(aes(x = individual_position, y = porcentaje, fill = tier)) +
  geom_col(position = "dodge") +
  labs(title = "Posición desempeñada por rango")
```



Conclusiónes generales del objetivo 1.

Los Masters tienen un promedio de 13428.55 de experiencia de campeon, con una desviación estandar de 4562.527 y un tiempo jugado de 1630.551 con desviación estandar de 400.9034.Las posiciónes preferidas son Middle y Jungle.

Los Grandmaster tienen un promedio de 13027.64 de experiencia de campeon, con una desviación estandar de 4424.238 y un tiempo jugado de 1592.610 con desviación estandar de 389.5518. Las posiciónes preferidas son Jungle y Bottom.

Los Challenger tienen un promedio de 12567.57 de experiencia de campeon, con una desviación estandar de 4516.364 y un tiempo jugado de 1594.505 con desviación estandar de 381.2901. Las posiciónes preferidas son Utility y Jungle.

Objetivo 2

Definir y operacionalizar indicadores de rendimiento deportivo en e-sports sostenido, Tower Percentage, número de inhibidores destruidos, tiempo jugado, tasa de victorias (winrate), oro ganado por minuto (GPM), KDA ajustado y control de objetivos, considerando además el componente temporal (por minuto o por fase de partida).

```
# Winrate por rango

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    winrate = mean(win, na.rm = TRUE),
    sd_winrate = sd(win, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 3
##   tier        winrate sd_winrate
##   <chr>         <dbl>      <dbl>
## 1 CHALLENGER    0.561      0.496
## 2 GRANDMASTER   0.548      0.498
## 3 MASTER        0.531      0.499
```

```
#Oro ganado por duración de partida y KDA

datos_rango_alto <- datos_rango_alto |>
  mutate(
    gpm = gold_earned/ (game_duration / 60),
    kda_ajustado = (kills + assists) / pmax(1, deaths),
    )

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    GPM = mean(gpm, na.rm = TRUE),
    sd_gpm = sd(gpm, na.rm = TRUE),
    KDA = mean(kda_ajustado, na.rm = TRUE),
    sd_Kda = sd(kda_ajustado, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 5
##   tier          GPM sd_gpm   KDA sd_Kda
##   <chr>       <dbl>  <dbl> <dbl>  <dbl>
## 1 CHALLENGER   397.   92.4  4.53   4.55
## 2 GRANDMASTER  409.   88.4  4.32   4.52
## 3 MASTER       402.   86.0  3.87   4.11
```

```
#Torres
datos_rango_alto <- datos_rango_alto |>
  mutate(tower_dpm = damage_dealt_to_turrets / (game_duration / 60))

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    DDTurrets = mean(tower_dpm, na.rm = TRUE),
    sd_DDTurrets = sd(tower_dpm, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 3
##   tier        DDTurrets sd_DDTurrets
##   <chr>           <dbl>        <dbl>
## 1 CHALLENGER       104.         121.
## 2 GRANDMASTER      123.         130.
## 3 MASTER           119.         129.
```

```
datos_rango_alto |>
  group_by(tier) |>
  summarise(
    avg_tower_kills = mean(tower_kills, na.rm = TRUE)
  )
```

```
## # A tibble: 3 x 2
##   tier        avg_tower_kills
##   <chr>                 <dbl>
## 1 CHALLENGER             5.47
## 2 GRANDMASTER            5.59
## 3 MASTER                 5.78
```

```
#Inhibidores

datos_rango_alto <- datos_rango_alto |>
  mutate(IPM = inhibitor_kills / (game_duration / 60))

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    Inhibidores = mean(IPM, na.rm = TRUE),
    sd_IPM = sd(IPM, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 3
##   tier        Inhibidores sd_IPM
##   <chr>             <dbl>  <dbl>
## 1 CHALLENGER       0.0227 0.0331
## 2 GRANDMASTER      0.0249 0.0343
## 3 MASTER           0.0282 0.0364
```

```
#Control de objetivos
datos_rango_alto <- datos_rango_alto |>
  mutate(
    total_objetivos = dragon_kills + baron_kills + herald_kills + tower_kills
  )

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    Objetivos = mean(total_objetivos, na.rm = TRUE),
    sd_Obj = sd(total_objetivos, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 3
##   tier        Objetivos sd_Obj
##   <chr>           <dbl>  <dbl>
## 1 CHALLENGER       8.10   4.84
## 2 GRANDMASTER      8.16   4.96
## 3 MASTER           8.45   5.21
```

```
datos_rango_alto <- datos_rango_alto |>
  mutate(
    opm = total_objetivos / (game_duration / 60)
  )

datos_rango_alto |>
  group_by(tier) |>
```

```
  summarise(
    ObjetivosxM = mean(opm, na.rm = TRUE),
    sd_Opm = sd(opm, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 3
##   tier        ObjetivosxM sd_Opm
##   <chr>             <dbl>  <dbl>
## 1 CHALLENGER        0.295  0.164
## 2 GRANDMASTER       0.298  0.173
## 3 MASTER            0.300  0.179
```

```
datos_rango_alto |>
  group_by(tier) |>
  summarise(
    avg_opm = mean(opm, na.rm = TRUE),
    avg_dragons = mean(dragon_kills),
    avg_barons = mean(baron_kills),
    avg_heralds = mean(herald_kills)
  )
```

```
## # A tibble: 3 x 5
##   tier        avg_opm avg_dragons avg_barons avg_heralds
##   <chr>         <dbl>       <dbl>      <dbl>       <dbl>
## 1 CHALLENGER    0.295        1.83      0.331       0.473
## 2 GRANDMASTER   0.298        1.77      0.325       0.473
## 3 MASTER        0.300        1.85      0.356       0.468
```

```
#Tiempo vivo

datos_rango_alto <- datos_rango_alto |>
  mutate(TV = longest_time_spent_living / (game_duration / 60))

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    Tiempovivo = mean(TV, na.rm = TRUE),
    sd_TV = sd(TV, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 3
##   tier        Tiempovivo sd_TV
##   <chr>            <dbl> <dbl>
## 1 CHALLENGER        22.6  11.2
## 2 GRANDMASTER       22.4  11.1
## 3 MASTER            21.3  10.7
```

```
#Tiempo muerto

datos_rango_alto <- datos_rango_alto |>
  mutate(TM = total_time_spent_dead / (game_duration / 60))
```

```r
datos_rango_alto |>
  group_by(tier) |>
  summarise(
    Tiempomuerto = mean(TM, na.rm = TRUE),
    sd_TM = sd(TM, na.rm = TRUE),
  )
```

```
## # A tibble: 3 x 3
##   tier        Tiempomuerto sd_TM
##   <chr>              <dbl> <dbl>
## 1 CHALLENGER          4.57  2.73
## 2 GRANDMASTER         4.73  2.82
## 3 MASTER              5.18  2.93
```

```r
#Tiempo controlando

datos_rango_alto <- datos_rango_alto |>
  mutate(TC = total_time_cc_dealt / (game_duration / 60))

datos_rango_alto |>
  group_by(tier) |>
  summarise(
    Tiempocontrol = mean(TC, na.rm = TRUE),
    sd_TC = sd(TC, na.rm = TRUE),
  )
```

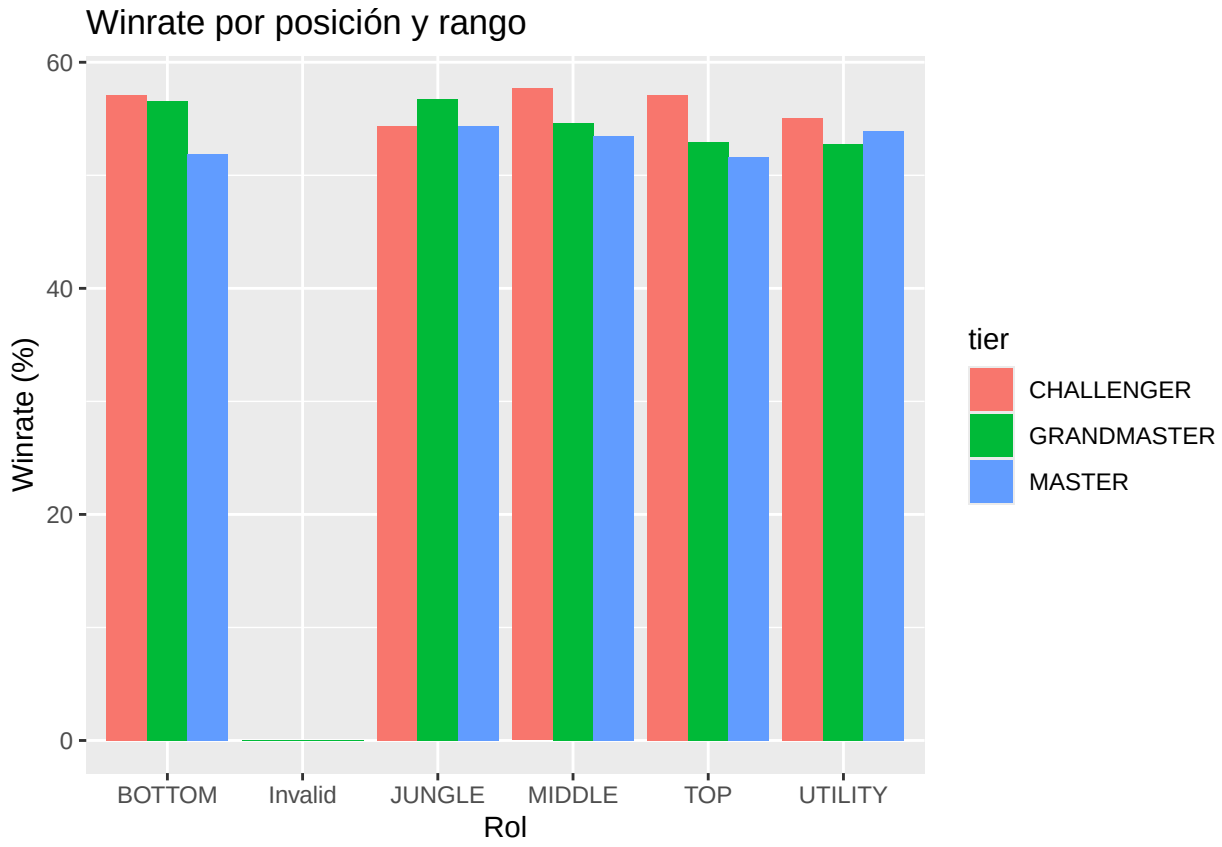
```
## # A tibble: 3 x 3
##   tier        Tiempocontrol sd_TC
##   <chr>               <dbl> <dbl>
## 1 CHALLENGER           9.39  8.57
## 2 GRANDMASTER          9.33  9.15
## 3 MASTER               9.34  8.51
```

ConclusiOnes generales del objetivo 2.

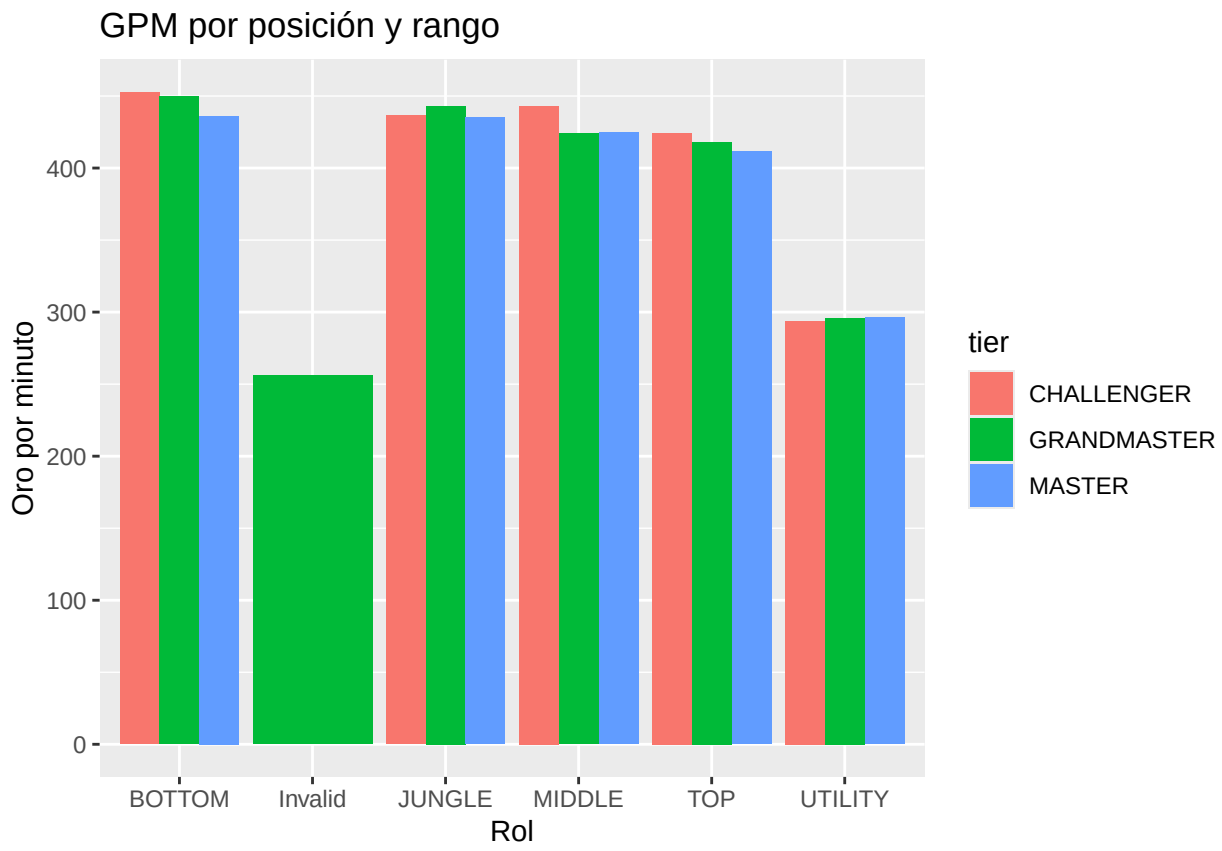
Por indicadores de resultado: Para los Challenger el

```r
resumen_rango_rol <- datos_rango_alto |>
  group_by(tier, individual_position) |>
  summarise(
    partidas = n(),
    winrate = mean(win, na.rm = TRUE) * 100,
    gpm = mean(gold_earned / (game_duration / 60), na.rm = TRUE),
    kda = mean((kills + assists) / pmax(1, deaths), na.rm = TRUE),
    objetivos = mean(dragon_kills + baron_kills + herald_kills, na.rm = TRUE),
    cc_pm = mean(total_time_cc_dealt / game_duration * 60, na.rm = TRUE),
    .groups = "drop"
  )

resumen_rango_rol |>
  ggplot(aes(x = individual_position, y = winrate, fill = tier)) +
  geom_col(position = "dodge") +
  labs(
```

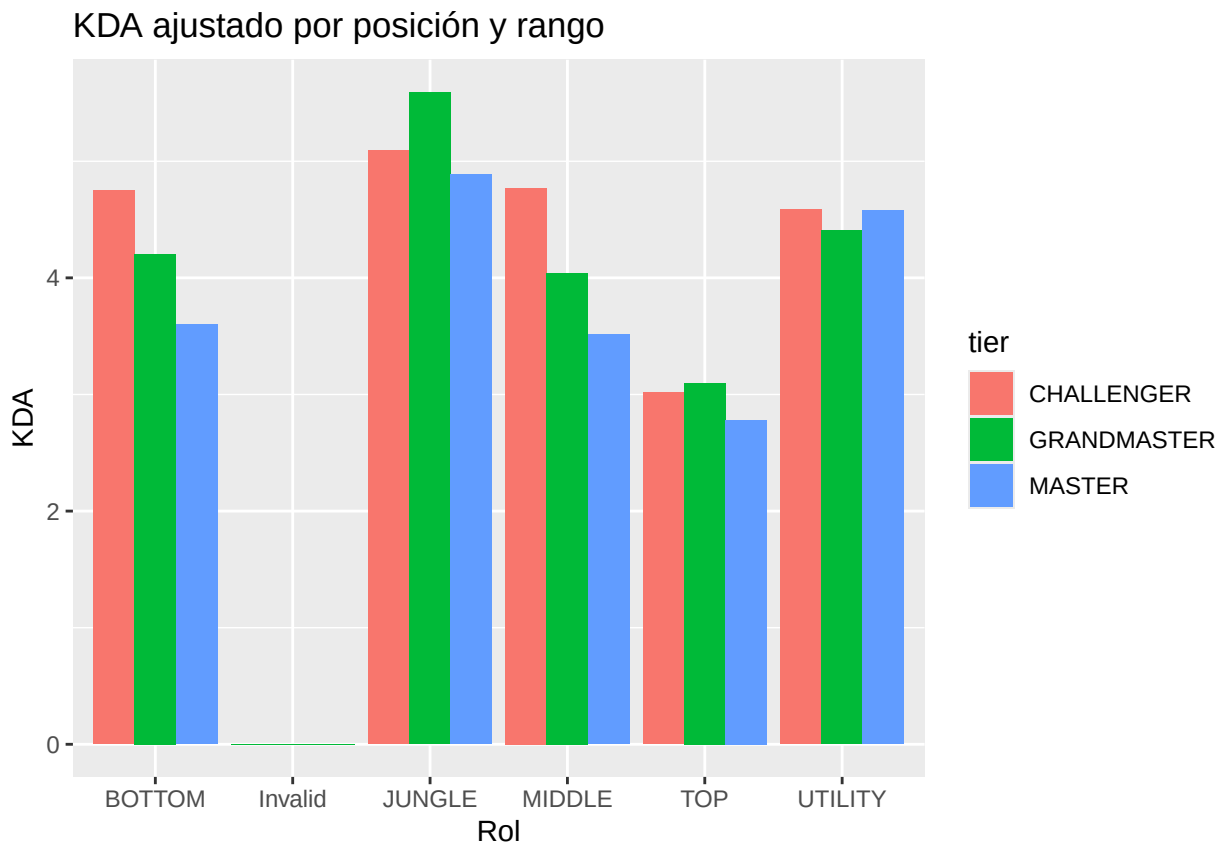
```
    title = "Winrate por posición y rango",
    x = "Rol",
    y = "Winrate (%)"
  )
```



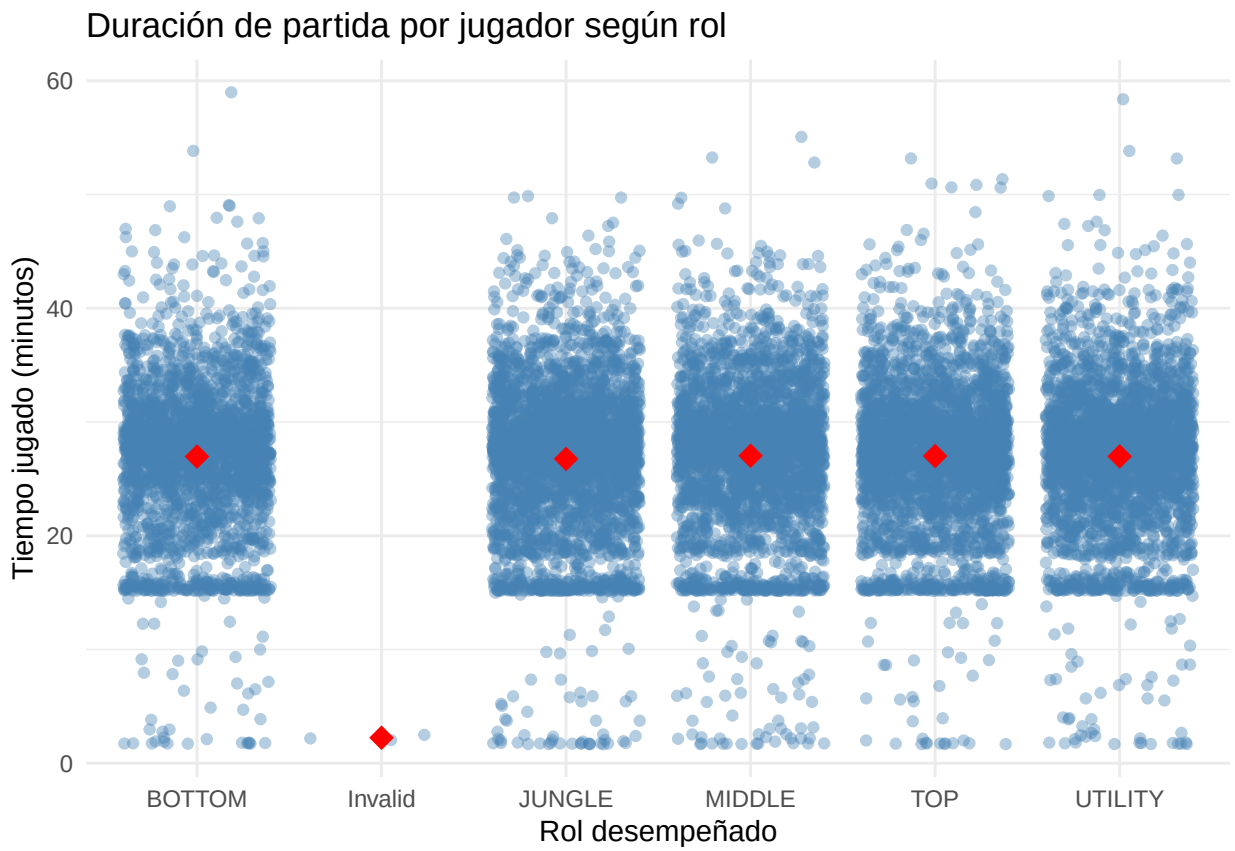

```
#Oro por minuto visualización

resumen_rango_rol |>
  ggplot(aes(x = individual_position, y = gpm, fill = tier)) +
  geom_col(position = "dodge") +
  labs(
    title = "GPM por posición y rango",
    x = "Rol",
    y = "Oro por minuto"
  )
```



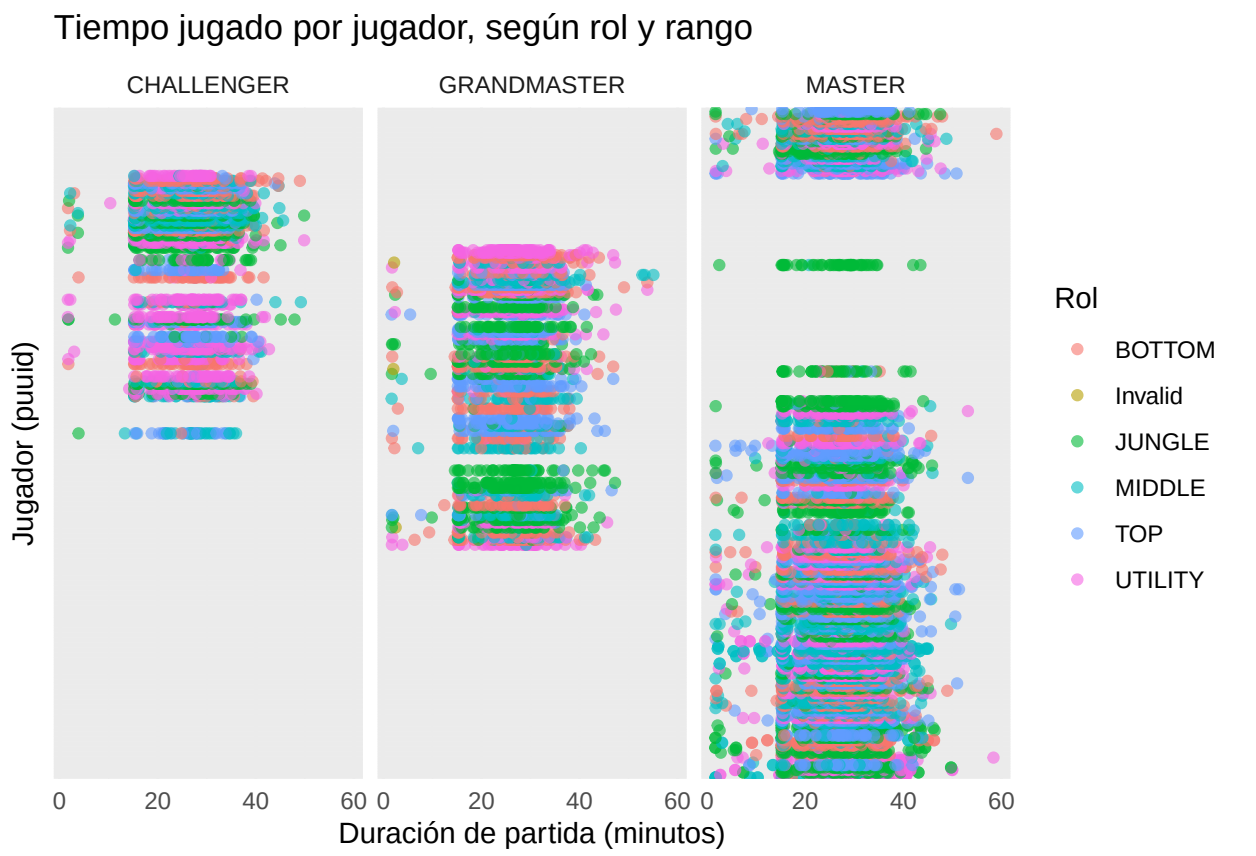

```
resumen_rango_rol |>
  ggplot(aes(x = individual_position, y = kda, fill = tier)) +
  geom_col(position = "dodge") +
  labs(
    title = "KDA ajustado por posición y rango",
    x = "Rol",
    y = "KDA"
  )
```



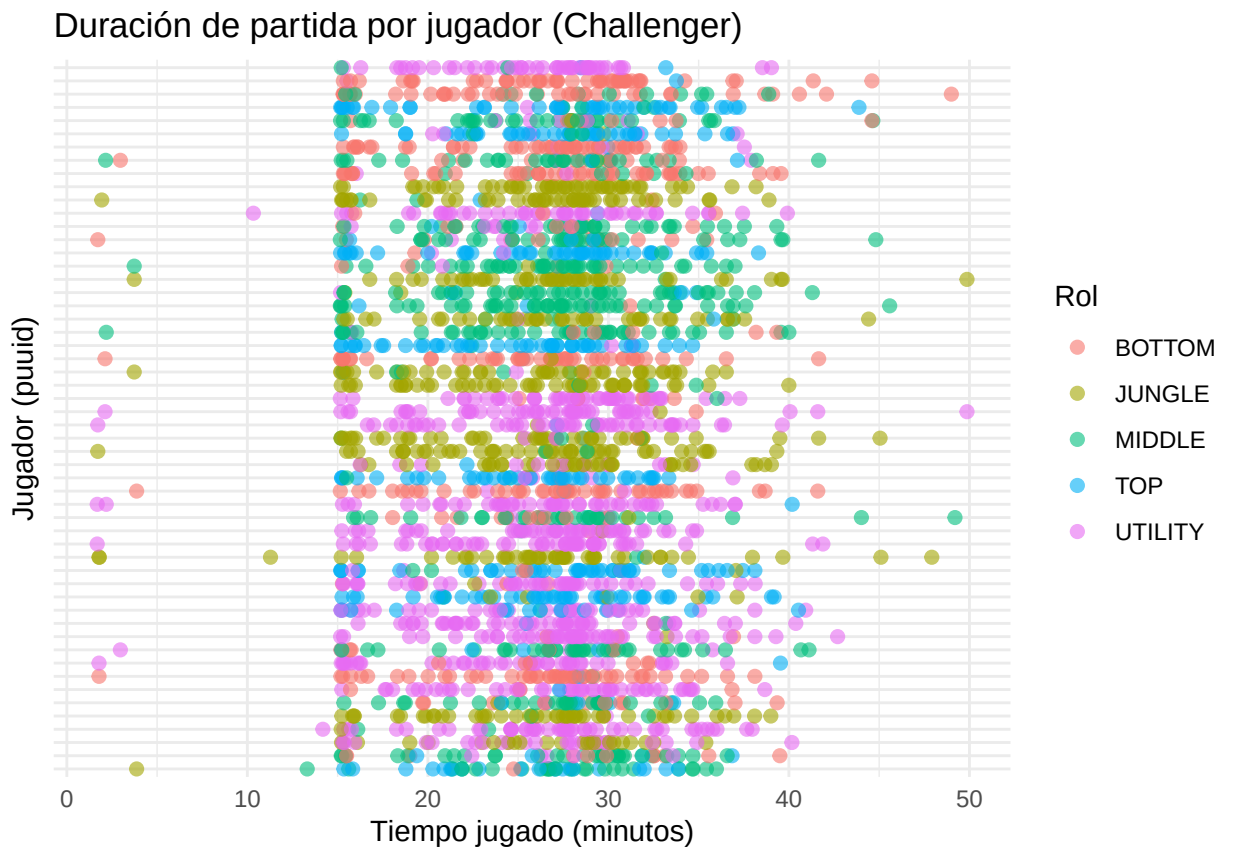

```
datos_rango_alto |>
  ggplot(aes(x = individual_position, y = game_duration / 60)) +  # convertimos a minutos
  geom_jitter(alpha = 0.4, color = "steelblue") +
  stat_summary(fun = mean, geom = "point", shape = 18, size = 4, color = "red") +
  labs(
    title = "Duración de partida por jugador según rol",
    x = "Rol desempeñado",
    y = "Tiempo jugado (minutos)"
  ) +
  theme_minimal()
```




```
datos_rango_alto |>
  ggplot(aes(x = game_duration / 60, y = fct_inorder(puuid), color = individual_position)) +
  geom_point(alpha = 0.6) +
  labs(
    title = "Tiempo jugado por jugador, según rol y rango",
    x = "Duración de partida (minutos)",
    y = "Jugador (puuid)",
    color = "Rol"
  ) +
  facet_wrap(~ tier) +
  theme_minimal() +
  theme(
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank()
  )
```

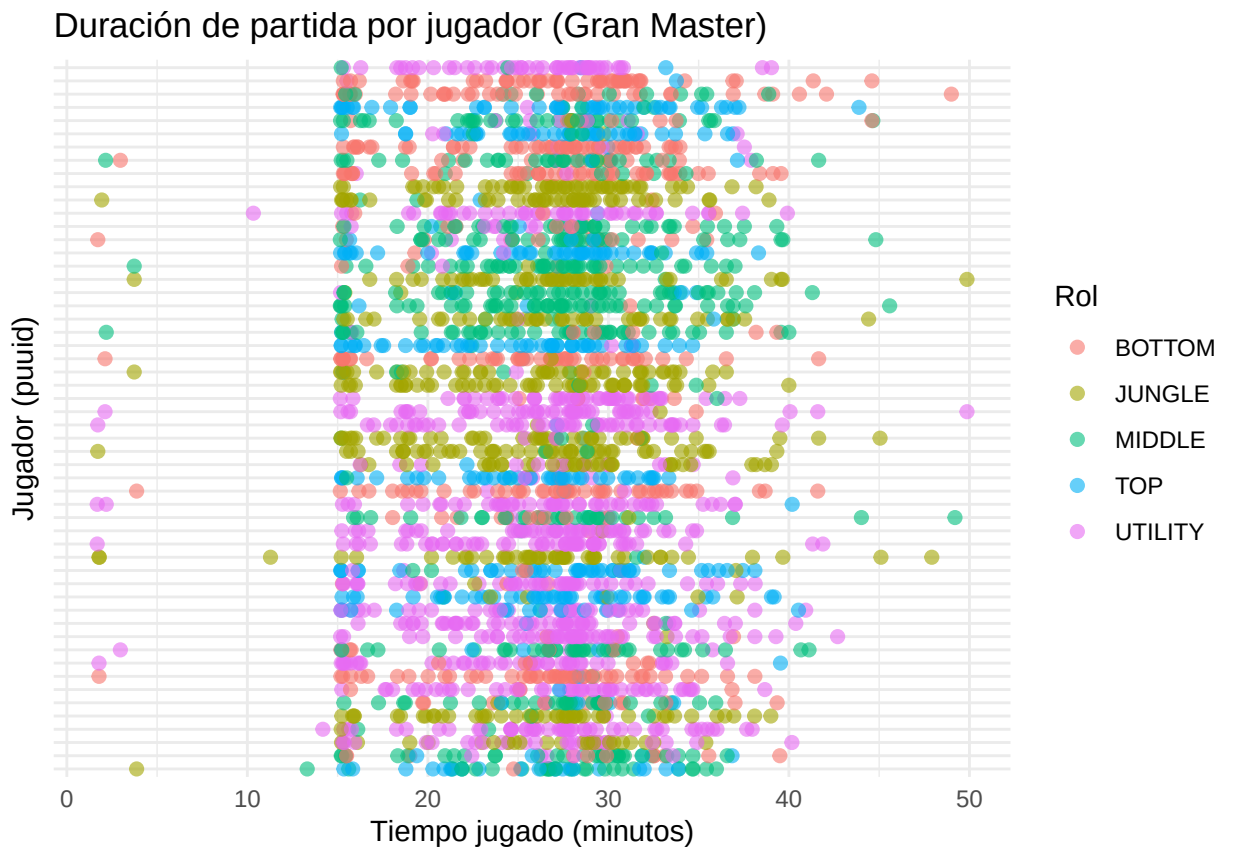



```r
datos_challenger <- datos_rango_alto |>
  filter(tier == "CHALLENGER")


library(ggplot2)
library(forcats)

datos_challenger |>
  ggplot(aes(x = game_duration / 60, y = fct_inorder(puuid), color = individual_position)) +
  geom_point(alpha = 0.6, size = 2) +
  labs(
    title = "Duración de partida por jugador (Challenger)",
    x = "Tiempo jugado (minutos)",
    y = "Jugador (puuid)",
    color = "Rol"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank()
  )
```




```
datos_master <- datos_rango_alto |>
  filter(tier == "MASTER")


library(ggplot2)
library(forcats)

datos_challenger |>
  ggplot(aes(x = game_duration / 60, y = fct_inorder(puuid), color = individual_position)) +
  geom_point(alpha = 0.6, size = 2) +
  labs(
    title = "Duración de partida por jugador (Master)",
    x = "Tiempo jugado (minutos)",
    y = "Jugador (puuid)",
    color = "Rol"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank()
  )
```


```
datos_granmaster <- datos_rango_alto |>
  filter(tier == "GRANDMASTER")


library(ggplot2)
library(forcats)

datos_challenger |>
  ggplot(aes(x = game_duration / 60, y = fct_inorder(puuid), color = individual_position)) +
  geom_point(alpha = 0.6, size = 2) +
  labs(
    title = "Duración de partida por jugador (Gran Master)",
    x = "Tiempo jugado (minutos)",
    y = "Jugador (puuid)",
    color = "Rol"
  ) +
  theme_minimal() +
  theme(
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank()
  )
```

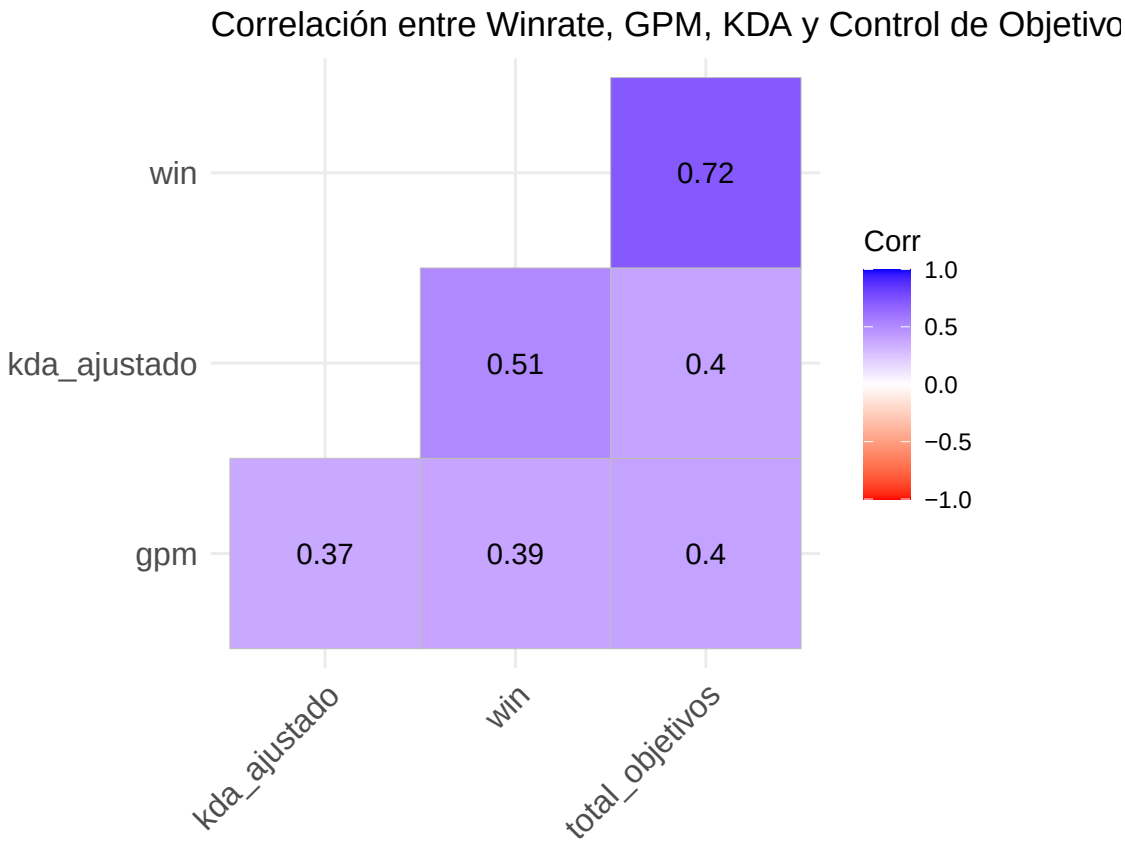



```
#Correlaciones

vars_cor <- datos_rango_alto %>%
  select(win, gpm, kda_ajustado, total_objetivos)

# Calcular matriz de correlaciones
cor_matrix <- cor(vars_cor, use = "complete.obs")

cor_matrix
```

```
##                       win       gpm kda_ajustado total_objetivos
## win             1.0000000 0.3913041    0.5075707       0.7150500
## gpm             0.3913041 1.0000000    0.3697646       0.3956404
## kda_ajustado    0.5075707 0.3697646    1.0000000       0.3960337
## total_objetivos 0.7150500 0.3956404    0.3960337       1.0000000
```

```
#Hipotesis planteada:
#Existe una correlación positiva entre los principales indicadores de rendimiento competitivo, como la


ggcorrplot(cor_matrix,
           hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 4,
```

```
          colors = c("red", "white", "blue"),
          title = "Correlación entre Winrate, GPM, KDA y Control de Objetivos",
          ggtheme = theme_minimal())
```




```
#Hipotesis planteada:
#La tasa de victorias (winrate) se correlaciona positivamente con el porcentaje de torres destruidas (T

library(scales)

MAX_TOWERS <- 11

df_side <- datos_rango_alto %>%
  group_by(match_id, win) %>%                  # dos grupos por partida: win=1 (ganador), win=0 (perdedo
  summarise(
    tower_kills     = sum(tower_kills, na.rm = TRUE),
    inhibitor_kills = sum(inhibitor_kills, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(
    tower_pct = pmin(tower_kills / MAX_TOWERS, 1),
    winrate   = as.numeric(win)                # 0 o 1
  )

# 2) Correlaciones
cor_win_tower <- cor.test(df_side$winrate, df_side$tower_pct,     use = "complete.obs")
```

```r
cor_tower_inh <- cor.test(df_side$tower_pct, df_side$inhibitor_kills, use = "complete.obs")

cor_win_tower
```

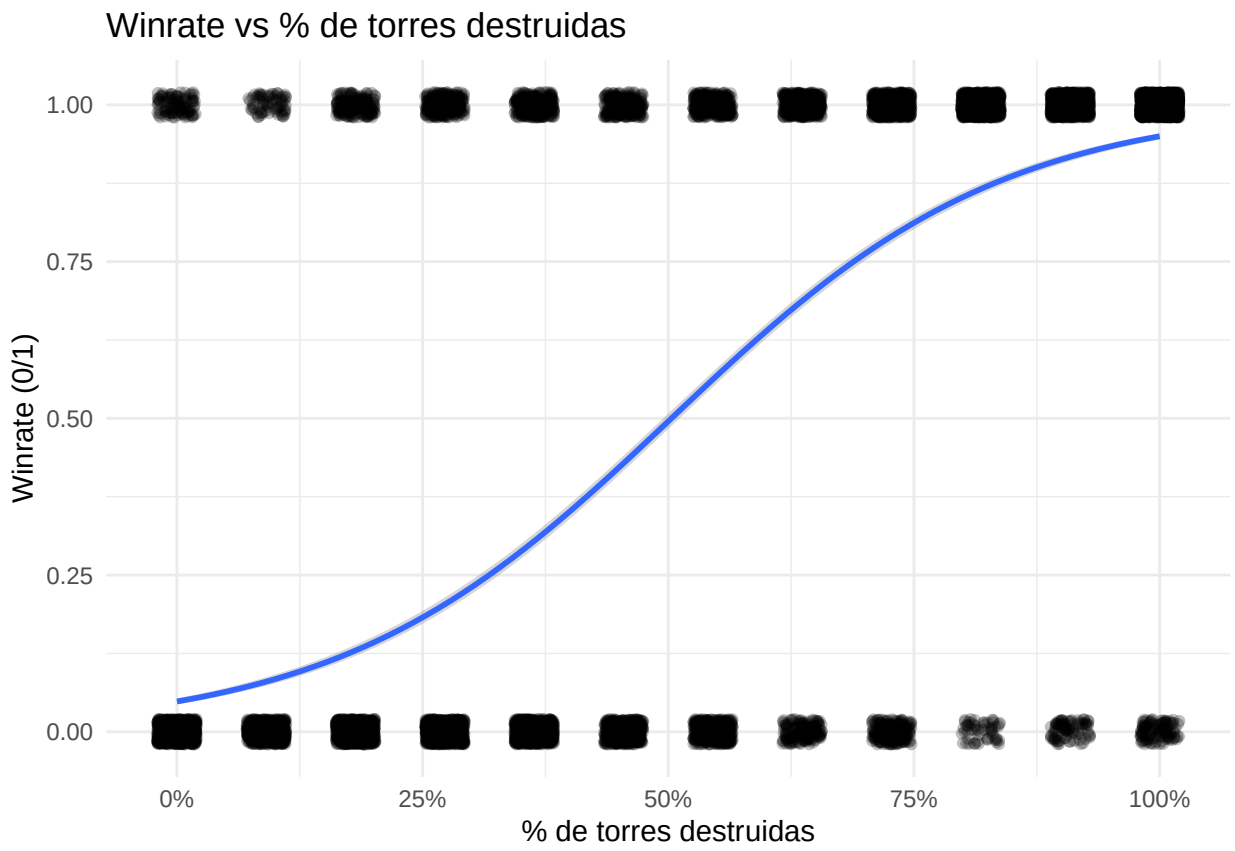
```
##
##  Pearson's product-moment correlation
##
## data:  df_side$winrate and df_side$tower_pct
## t = 105.81, df = 12655, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6757879 0.6942777
## sample estimates:
##       cor
## 0.6851431
```

```r
cor_tower_inh
```

```
##
##  Pearson's product-moment correlation
##
## data:  df_side$tower_pct and df_side$inhibitor_kills
## t = 121.83, df = 12655, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7265750 0.7426133
## sample estimates:
##       cor
## 0.7346968
```

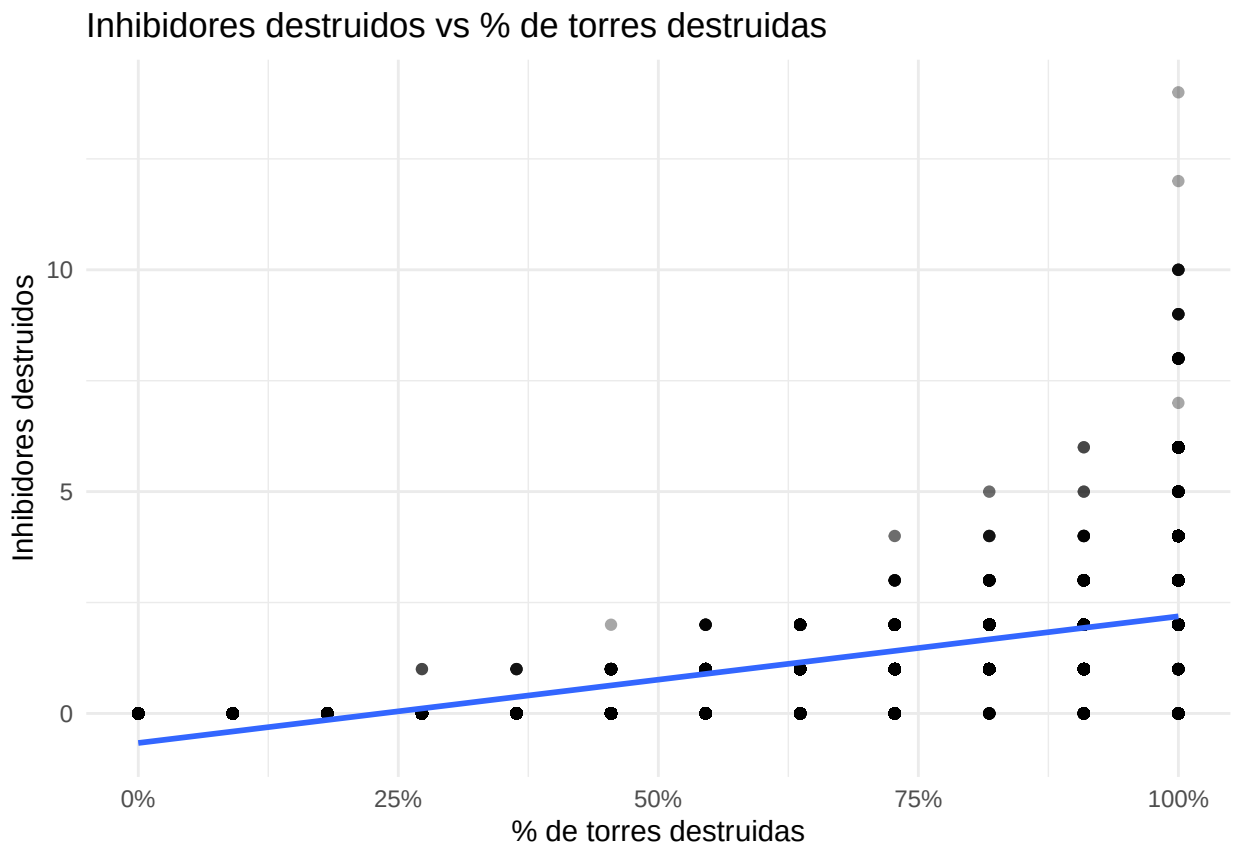
```r
# 3) Gráficos de apoyo
# Winrate (0/1) vs % torres (con curva logística)
ggplot(df_side, aes(x = tower_pct, y = winrate)) +
  geom_jitter(width = 0.02, height = 0.02, alpha = 0.25) +
  geom_smooth(method = "glm", method.args = list(family = "binomial"), se = TRUE) +
  scale_x_continuous(labels = percent_format(accuracy = 1)) +
  labs(title = "Winrate vs % de torres destruidas",
       x = "% de torres destruidas", y = "Winrate (0/1)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Inhibidores vs % torres
ggplot(df_side, aes(x = tower_pct, y = inhibitor_kills)) +
  geom_point(alpha = 0.35) +
  geom_smooth(method = "lm", se = TRUE) +
  scale_x_continuous(labels = percent_format(accuracy = 1)) +
  labs(title = "Inhibidores destruidos vs % de torres destruidas",
       x = "% de torres destruidas", y = "Inhibidores destruidos") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



#Que significan los valores

Winrate vs torres destruidas

cor = 0.685 → Correlación positiva y fuerte: cuando un lado destruye más torres, tiene mucha más probabilidad de ganar la partida.

p-value < 2.2e-16 → Esta relación no es por azar, es estadísticamente significativa.

IC 95% (0.676–0.694) → El valor verdadero de la correlación poblacional está en ese rango; siempre positivo y bastante alto.

Torres destruidas vs inhibidores destruidos

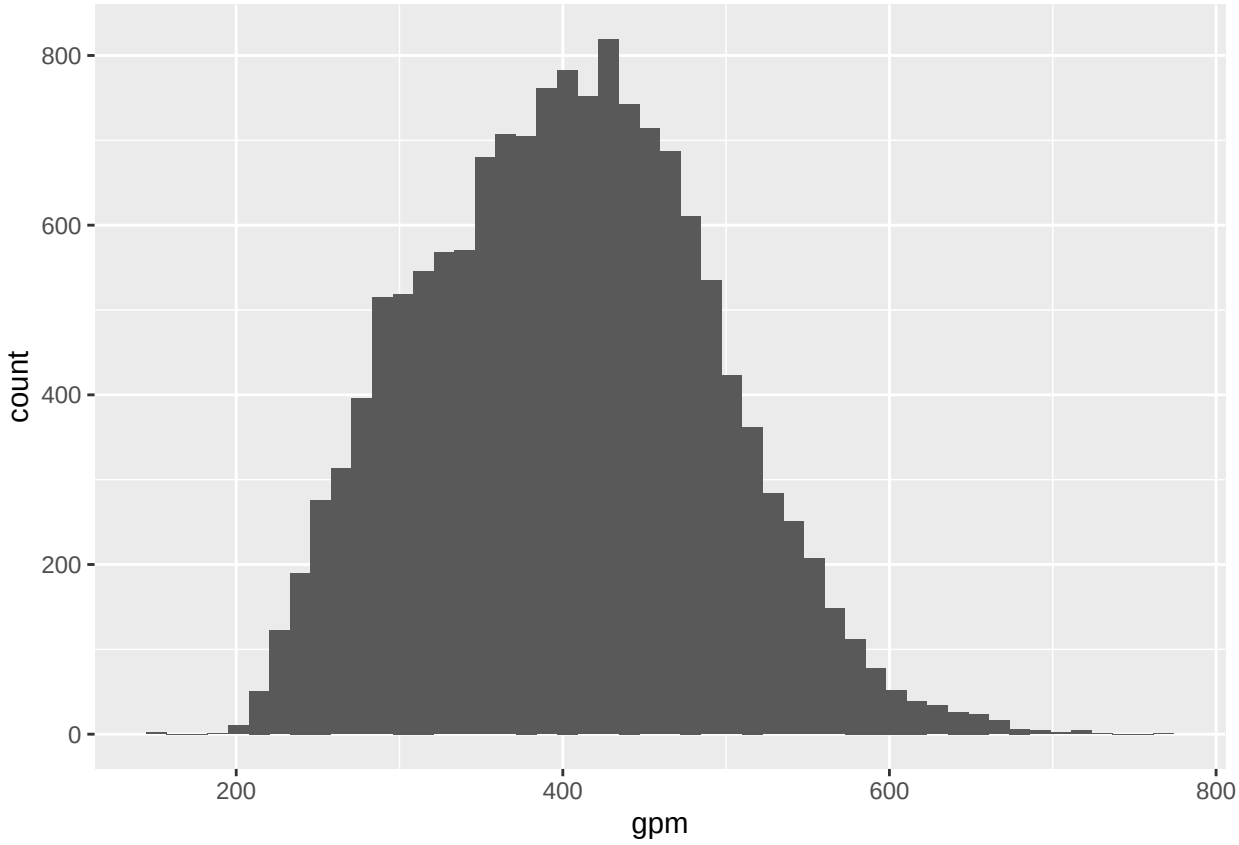
cor = 0.735 → Correlación positiva todavía más fuerte: cuantos más porcentajes de torres caen, más inhibidores se destruyen.

p-value < 2.2e-16 → Extremadamente significativo.

IC 95% (0.727–0.743) → La relación real casi seguro está en este rango alto.

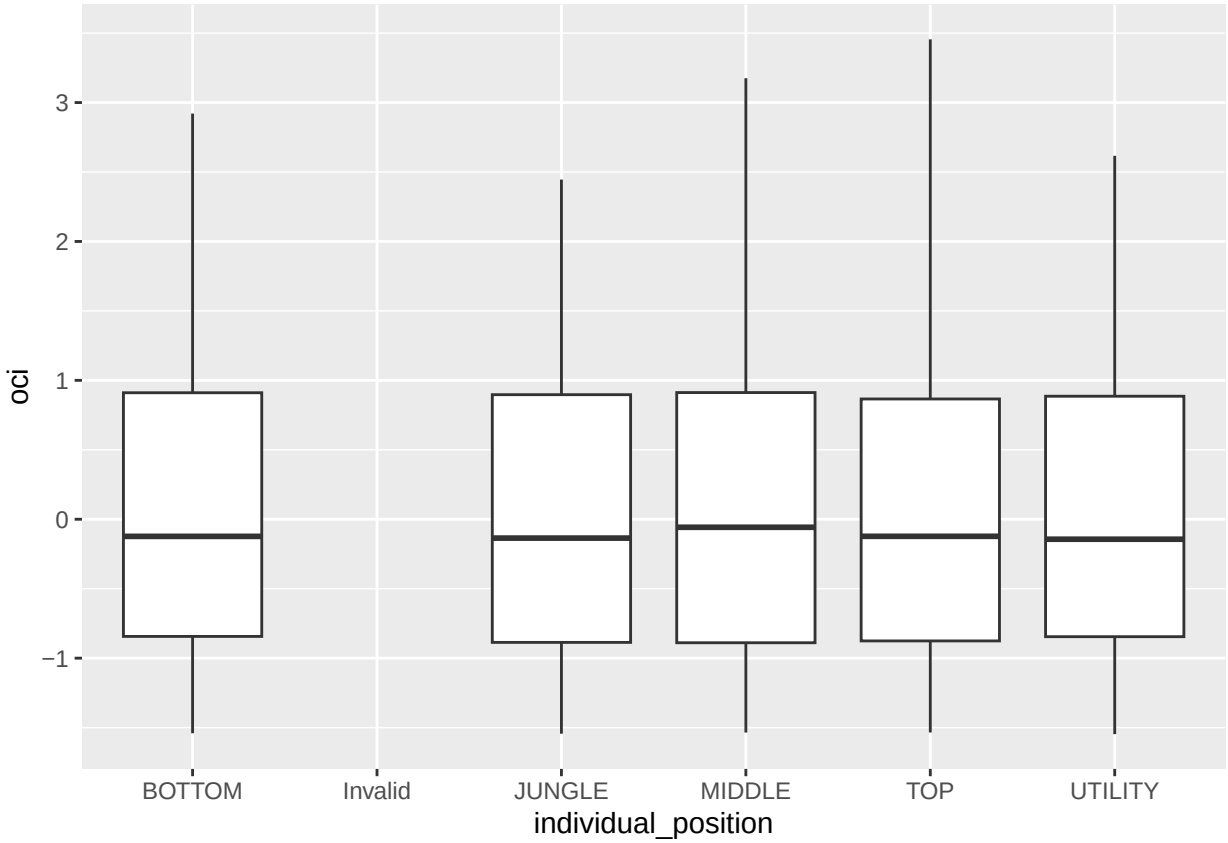
```
#Normalizar por duración

df <- datos_rango_alto %>%
  mutate(
    minutes = pmax(time_played/60, 1),
    gpm          = gold_earned / minutes,
    kda_adj      = (kills + assists) / pmax(deaths, 1),
    cs_min       = total_minions_killed / minutes,
    dmg_dealt_min= total_damage_dealt / minutes,
    dmg_taken_min= total_damage_taken / minutes,
```

```
    vision_min   = vision_score / minutes,
    cc_min       = (time_c_cing_others + total_time_cc_dealt) / (2*minutes)
  )

ggplot(df, aes(x = gpm)) + geom_histogram(bins = 50)
```




```
#OCI: indice de control de objetivos
df <- df %>%
  mutate(
    obj_epicos = dragon_kills + herald_kills + baron_kills,
    oci_raw    = 0.4*tower_kills + 0.3*inhibitor_kills + 0.3*obj_epicos
  ) %>%
  group_by(individual_position) %>%            # opcional: normaliza por rol
  mutate(oci = scale(oci_raw)[,1]) %>%
  ungroup()

ggplot(df, aes(x = individual_position, y = oci)) + geom_boxplot()
```

```
## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



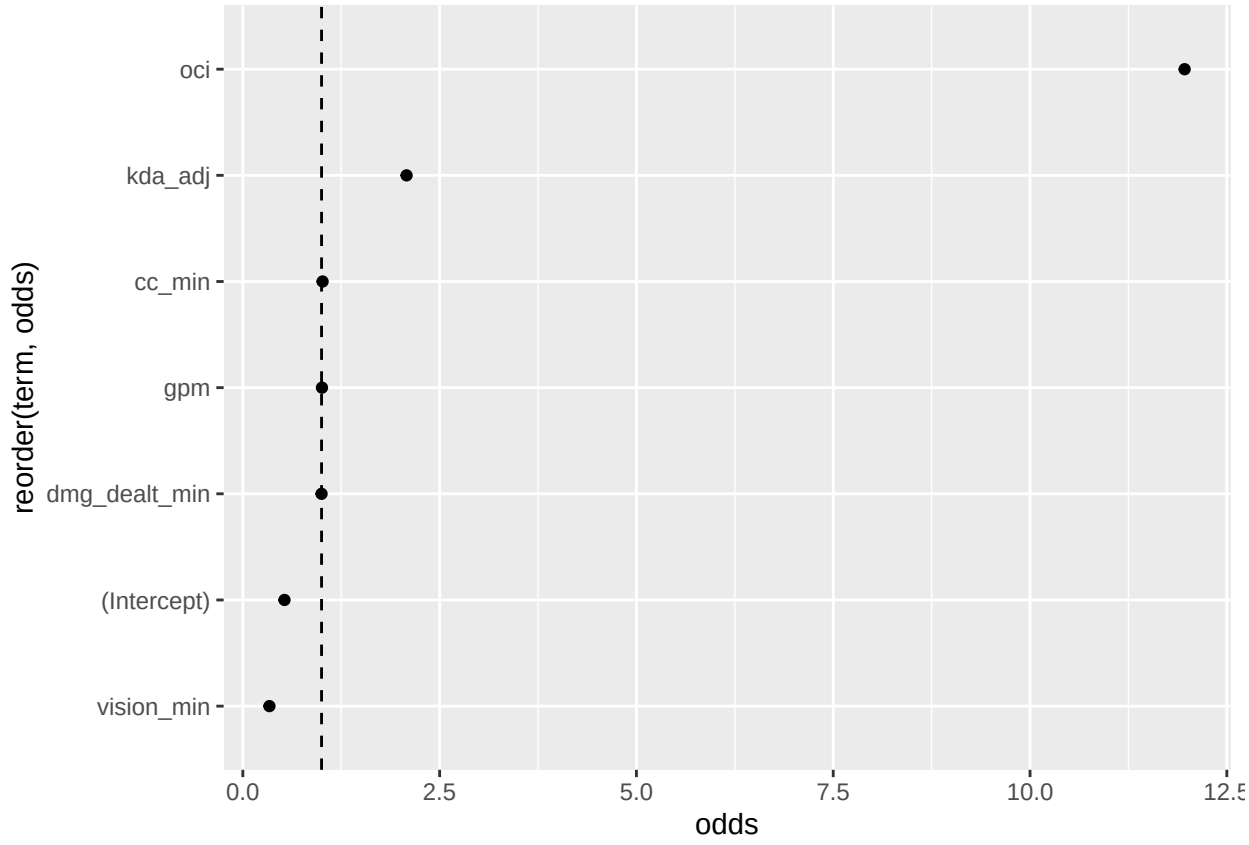

```
#Predición de victoria
m1 <- glm(win ~ gpm + kda_adj + oci + vision_min + dmg_dealt_min + cc_min,
          data = df, family = binomial())
broom::tidy(m1) |>
  mutate(odds_ratio = exp(estimate)) |>
  select(term, estimate, odds_ratio, p.value)
```

```
## # A tibble: 7 x 4
##   term           estimate odds_ratio   p.value
##   <chr>             <dbl>      <dbl>     <dbl>
## 1 (Intercept)   -0.637         0.529 1.17e-  2
## 2 gpm            0.00508       1.01  1.12e- 14
## 3 kda_adj        0.732         2.08  3.84e-278
## 4 oci            2.48         12.0   0
## 5 vision_min    -1.09          0.337 2.67e- 83
## 6 dmg_dealt_min -0.000442      1.00  2.18e-121
## 7 cc_min         0.0130        1.01  7.72e-  2
```

```
library(broom)

m1 <- glm(win ~ gpm + kda_adj + oci + vision_min + dmg_dealt_min + cc_min,
          data = df, family = binomial())
broom::tidy(m1) |>
  mutate(odds_ratio = exp(estimate)) |>
  select(term, estimate, odds_ratio, p.value)
```

```
## # A tibble: 7 x 4
##   term           estimate odds_ratio   p.value
##   <chr>             <dbl>      <dbl>     <dbl>
## 1 (Intercept)   -0.637         0.529 1.17e-  2
## 2 gpm            0.00508       1.01  1.12e- 14
## 3 kda_adj        0.732         2.08  3.84e-278
## 4 oci            2.48         12.0   0
## 5 vision_min    -1.09          0.337 2.67e- 83
## 6 dmg_dealt_min -0.000442      1.00  2.18e-121
## 7 cc_min         0.0130        1.01  7.72e-  2
```

```
coefs <- broom::tidy(m1) |> mutate(odds = exp(estimate))
ggplot(coefs, aes(x = reorder(term, odds), y = odds)) +
  geom_point() + coord_flip() + geom_hline(yintercept=1, linetype=2)
```




```
#importancias no llineales(random Forest)
library(ranger)
```

```
## Warning: package 'ranger' was built under R version 4.4.3
```

```
rf <- ranger(
  win ~ gpm + kda_adj + oci + vision_min + dmg_dealt_min + dmg_taken_min + cs_min + cc_min,
  data = df, probability = TRUE, importance = "impurity"
)
```

```r
sort(rf$variable.importance, decreasing = TRUE)
```

```
##           oci       kda_adj           gpm dmg_taken_min dmg_dealt_min
##     2935.6111     1856.1072      590.2135      375.8143      327.5572
##        cs_min    vision_min        cc_min
##      272.0233      228.8865      178.1355
```

```r
#Correlaciones y correlacion parcial (controlando duración)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.4.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
df |> select(win, gpm, kda_adj, oci, vision_min, dmg_dealt_min, cc_min) |> GGally::ggpairs()
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 3 rows containing missing values
```

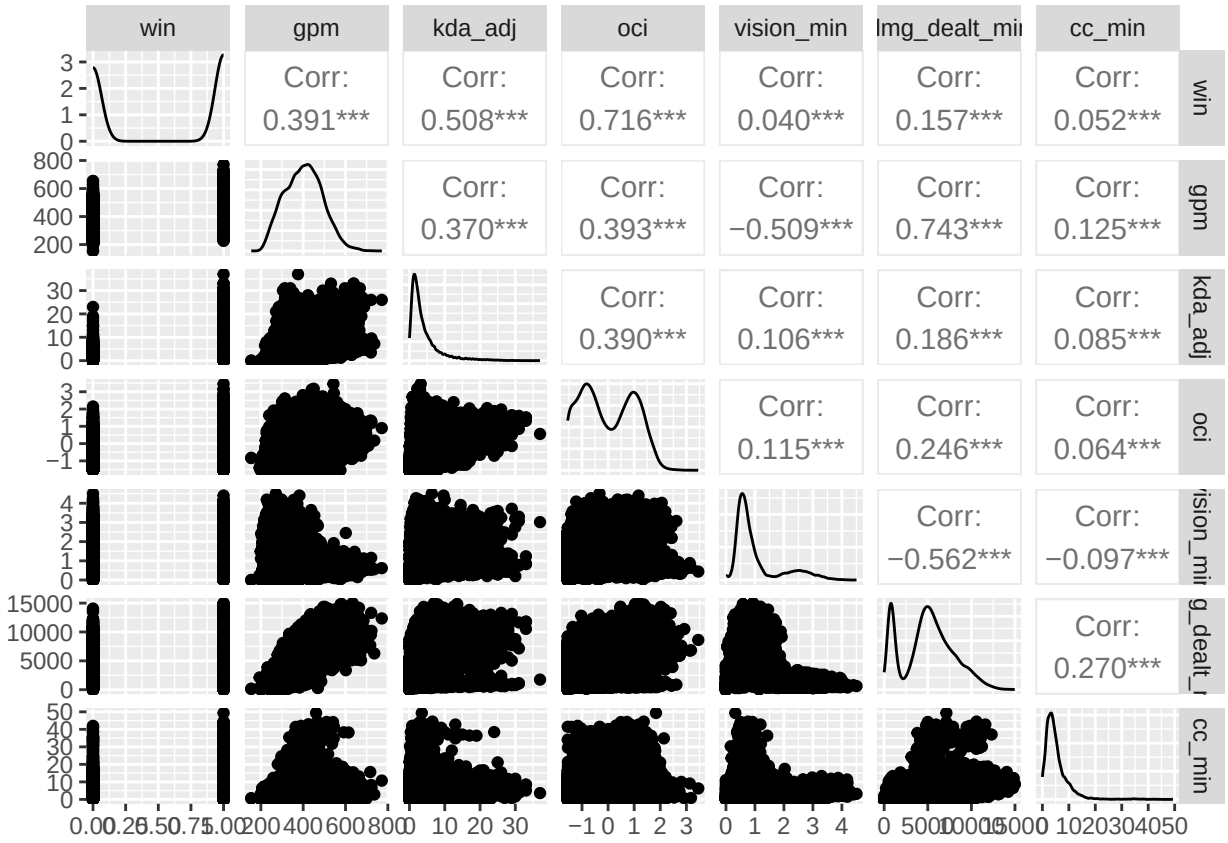
```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 3 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 3 rows containing missing values
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_density()`).
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 3 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 3 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 3 rows containing missing values
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 3 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# install.packages("ppcor")
library(ppcor)
```

```
## Warning: package 'ppcor' was built under R version 4.4.3
```

```
## Cargando paquete requerido: MASS
```

```
##
## Adjuntando el paquete: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
pcor.test(df$win, df$oci, df$time_played)    # correlación parcial win-oci | duración
```

```
## Error in if (det(cvx) < .Machine$double.eps) {: valor ausente donde TRUE/FALSE es necesario
```

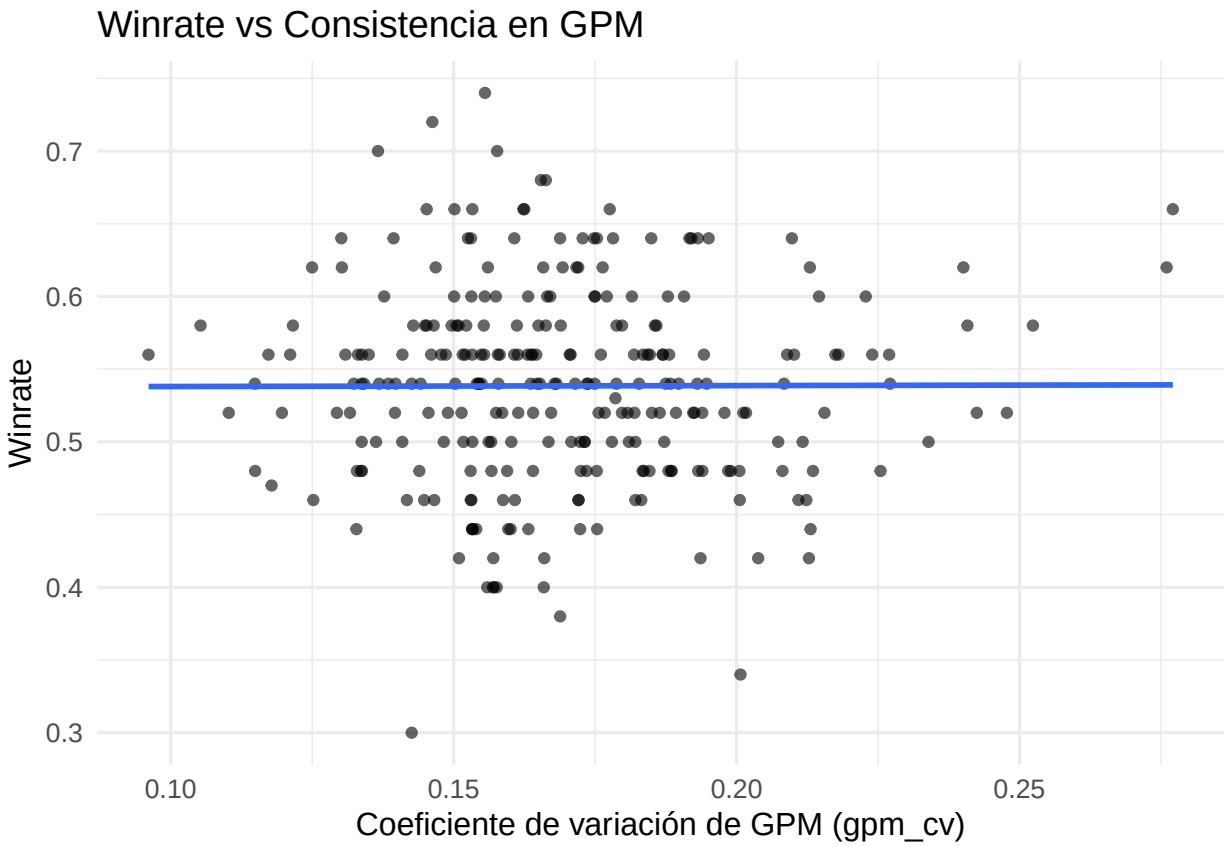
```
#Medir si algún rol pesa más que otro
m2 <- glm(win ~ gpm + kda_adj + oci*individual_position + vision_min,
          data = df, family = binomial())
broom::tidy(m2)
```

```
## # A tibble: 13 x 5
##    term                            estimate std.error statistic   p.value
##    <chr>                              <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                     -0.918     0.261      -3.52   4.28e-  4
##  2 gpm                              0.000594  0.000593    1.00   3.16e-  1
##  3 kda_adj                          0.704     0.0204     34.6    6.51e-262
##  4 oci                              2.13      0.0980     21.8    5.28e-105
##  5 individual_positionJUNGLE       -0.00308   0.100      -0.0307 9.76e-  1
##  6 individual_positionMIDDLE        0.148     0.0956      1.55   1.22e-  1
##  7 individual_positionTOP           0.430     0.0968      4.44   9.01e-  6
##  8 individual_positionUTILITY       3.61      0.234      15.4    1.83e- 53
##  9 vision_min                      -1.91      0.0989    -19.3    7.81e- 83
## 10 oci:individual_positionJUNGLE    0.00314   0.131       0.0239 9.81e-  1
## 11 oci:individual_positionMIDDLE    0.00942   0.128       0.0734 9.41e-  1
## 12 oci:individual_positionTOP       0.164     0.133       1.24   2.16e-  1
## 13 oci:individual_positionUTILITY   0.953     0.151       6.31   2.78e- 10
```

```
#Consistencia por jugador
consistencia <- df %>%
  group_by(puuid) %>%
  summarise(
    n_matches = n(),
    gpm_mean  = mean(gpm, na.rm=TRUE),
    gpm_sd    = sd(gpm, na.rm=TRUE),
    kda_mean  = mean(kda_adj, na.rm=TRUE),
    kda_sd    = sd(kda_adj, na.rm=TRUE),
    oci_mean  = mean(oci, na.rm=TRUE),
    oci_sd    = sd(oci, na.rm=TRUE),
    winrate   = mean(win, na.rm=TRUE)
  ) %>%
  filter(n_matches >= 10) %>%
  mutate(
    gpm_cv = gpm_sd/pmax(gpm_mean, 1e-6),
    kda_cv = kda_sd/pmax(kda_mean, 1e-6),
    oci_cv = oci_sd/pmax(oci_mean, 1e-6)
  )
```

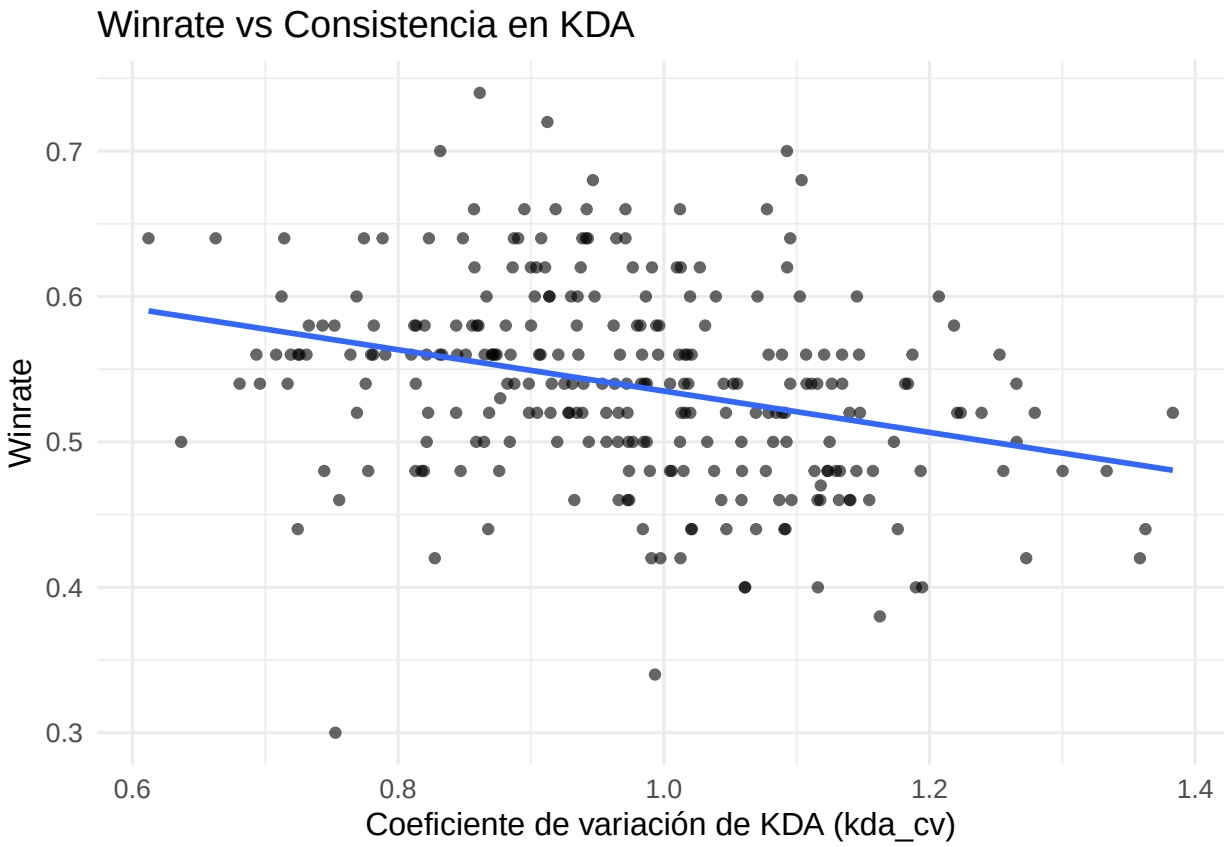
```
# Paquetes
library(ggplot2)
library(dplyr)

# 1) Winrate ~ gpm_cv
p_gpm <- ggplot(consistencia, aes(x = gpm_cv, y = winrate)) +
  geom_point(alpha = 0.6) +
  stat_smooth(method = "lm", se = FALSE) +
  labs(title = "Winrate vs Consistencia en GPM",
       x = "Coeficiente de variación de GPM (gpm_cv)",
       y = "Winrate") +
  theme_minimal(base_size = 12)
p_gpm
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



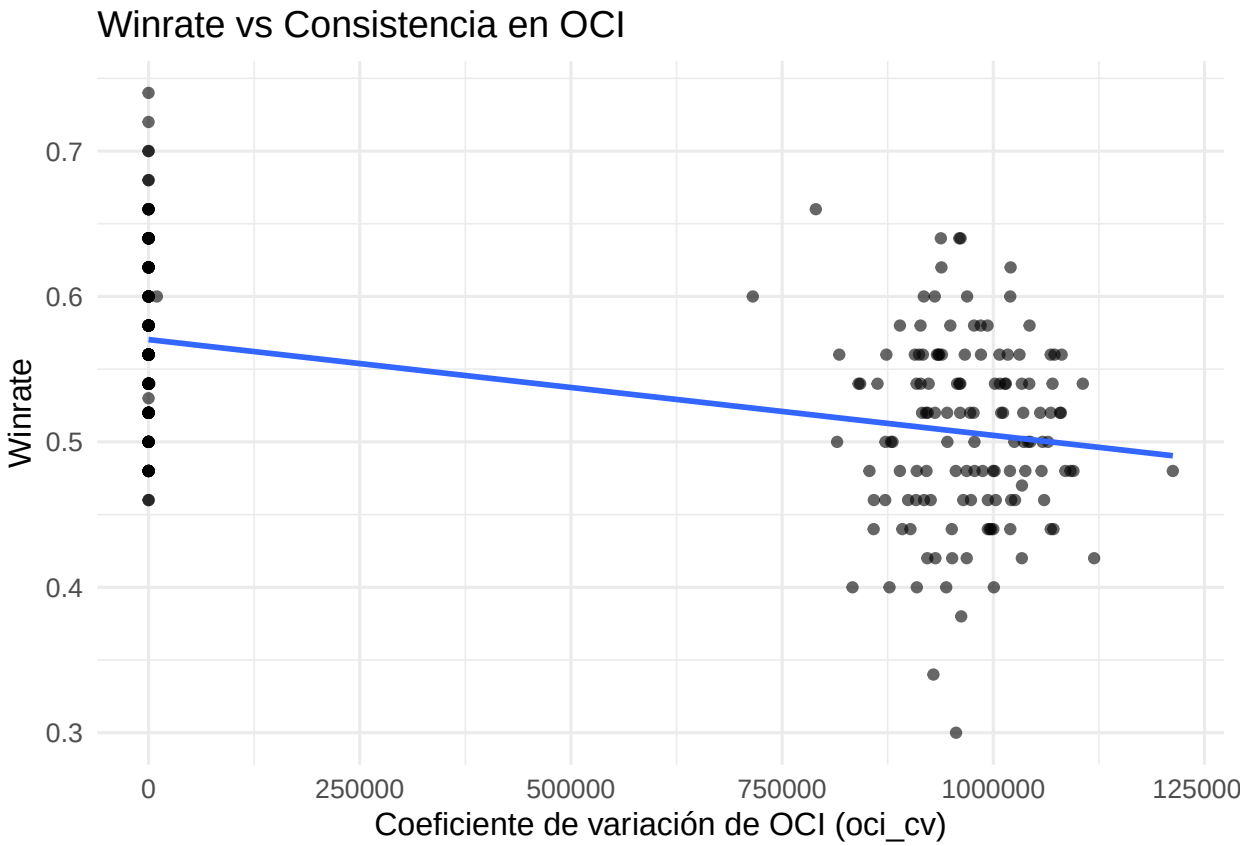

```
# 2) Winrate ~ kda_cv
p_kda <- ggplot(consistencia, aes(x = kda_cv, y = winrate)) +
  geom_point(alpha = 0.6) +
  stat_smooth(method = "lm", se = FALSE) +
  labs(title = "Winrate vs Consistencia en KDA",
       x = "Coeficiente de variación de KDA (kda_cv)",
       y = "Winrate") +
  theme_minimal(base_size = 12)
p_kda
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# 3) Winrate ~ oci_cv
p_oci <- ggplot(consistencia, aes(x = oci_cv, y = winrate)) +
  geom_point(alpha = 0.6) +
  stat_smooth(method = "lm", se = FALSE) +
  labs(title = "Winrate vs Consistencia en OCI",
       x = "Coeficiente de variación de OCI (oci_cv)",
       y = "Winrate") +
  theme_minimal(base_size = 12)
p_oci
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#Perfiles de estilo (PCA +k means)
library(dplyr)
library (FactoMineR); library(factoextra)
```

```
## Warning: package 'FactoMineR' was built under R version 4.4.3
```

```
## Warning: package 'factoextra' was built under R version 4.4.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
X <- df %>%
  dplyr::select(gpm, kda_adj, oci, vision_min, dmg_dealt_min, dmg_taken_min, cs_min, cc_min) %>% scale()
pca <- PCA(X, graph = FALSE)
```

```
## Warning in PCA(X, graph = FALSE): Missing values are imputed by the mean of the
## variable: you should use the imputePCA function of the missMDA package
```

```
set.seed(123)
km  <- kmeans(pca$ind$coord[,1:3], centers = 4, nstart = 25)
df$cluster <- factor(km$cluster)

df %>%
  group_by(cluster) %>%
  summarise(winrate = mean(win),
            across(where(is.numeric), mean, .names="mean_{.col}"))
```

```
## # A tibble: 4 x 55
##   cluster winrate mean_league_points mean_game_duration mean_champ_experience
##   <fct>     <dbl>              <dbl>              <dbl>                 <dbl>
## 1 1         0.919               590.              1705.                16147.
## 2 2         0.203               559.              1483.                11649.
## 3 3         0.539               647.              1631.                 9871.
## 4 4         0.606               651.              1698.                15154.
## # i 50 more variables: mean_champ_level <dbl>, mean_gold_earned <dbl>,
## #   mean_total_minions_killed <dbl>, mean_kills <dbl>, mean_deaths <dbl>,
## #   mean_assists <dbl>, mean_total_damage_dealt <dbl>,
## #   mean_total_damage_taken <dbl>, mean_damage_dealt_to_buildings <dbl>,
## #   mean_damage_dealt_to_objectives <dbl>, mean_damage_dealt_to_turrets <dbl>,
## #   mean_vision_score <dbl>, mean_baron_kills <dbl>, mean_dragon_kills <dbl>,
## #   mean_tower_kills <dbl>, mean_herald_kills <dbl>, ...
```

```
#Curvas de respuesta
# install.packages("mgcv")
library(mgcv)
```

```
## Cargando paquete requerido: nlme
```

```
##
## Adjuntando el paquete: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

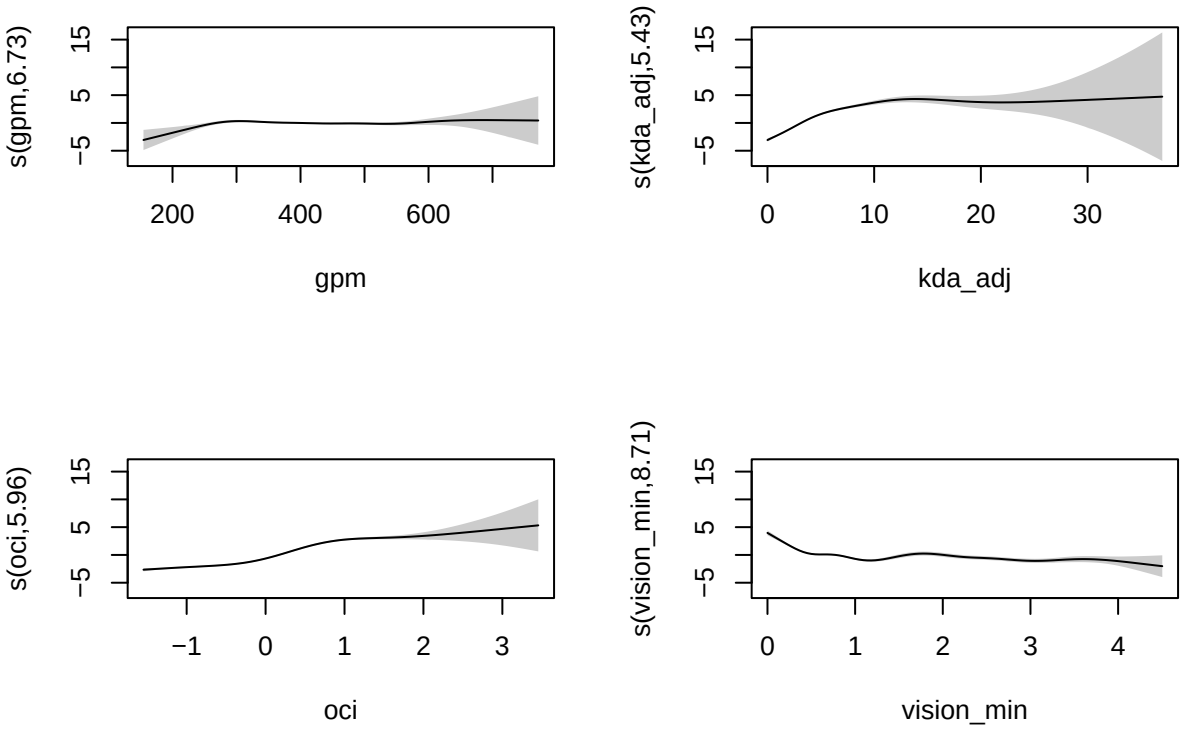
```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
gam1 <- gam(win ~ s(gpm) + s(kda_adj) + s(oci) + s(vision_min),
            data = df, family = binomial())
summary(gam1)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## win ~ s(gpm) + s(kda_adj) + s(oci) + s(vision_min)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.7822     0.0453   17.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df Chi.sq p-value
## s(gpm)       6.728  7.850   28.8 0.00041 ***
## s(kda_adj)   5.426  6.378 1709.8 < 2e-16 ***
## s(oci)       5.964  6.870 2630.9 < 2e-16 ***
```

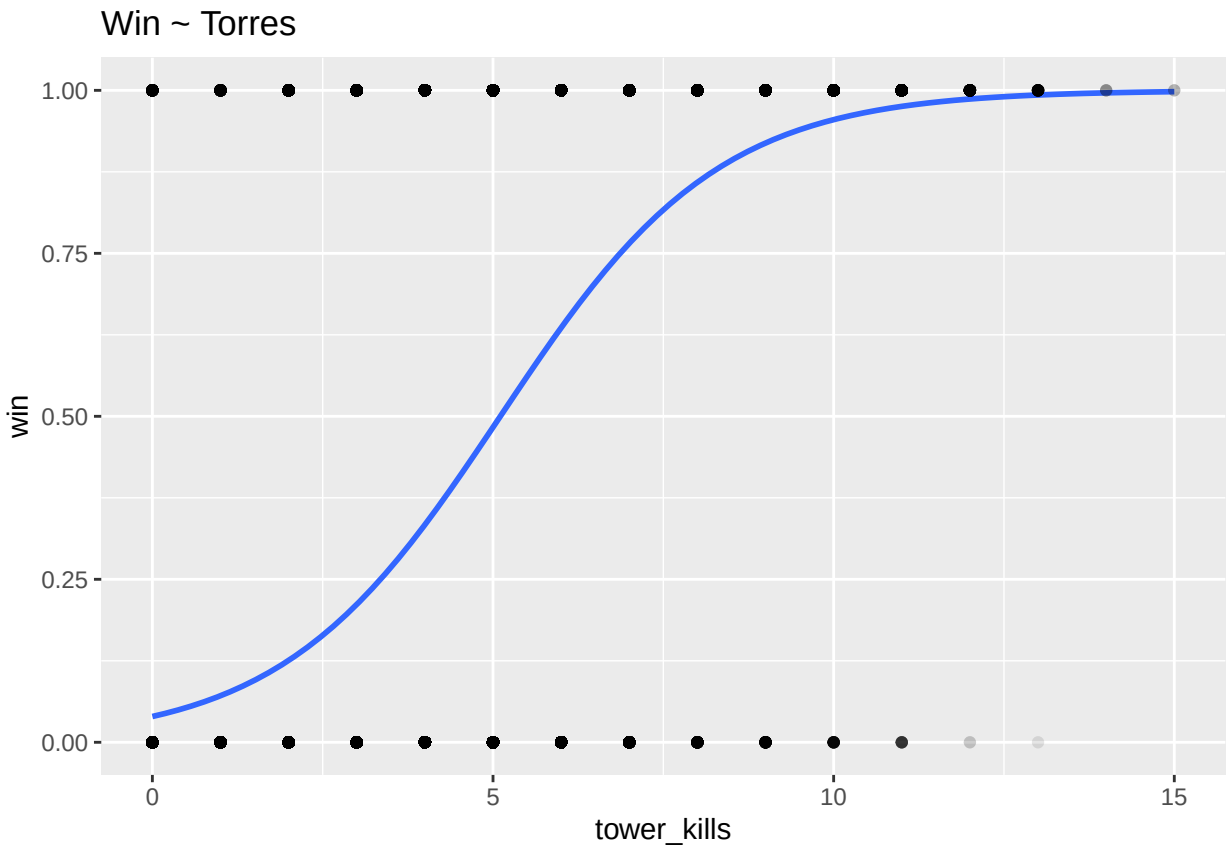
```
## s(vision_min) 8.714  8.975  447.6 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.723   Deviance explained = 65.8%
## UBRE = -0.52487  Scale est. = 1         n = 14647
```

```
plot(gam1, pages = 1, shade = TRUE)    # ver formas de las curvas
```



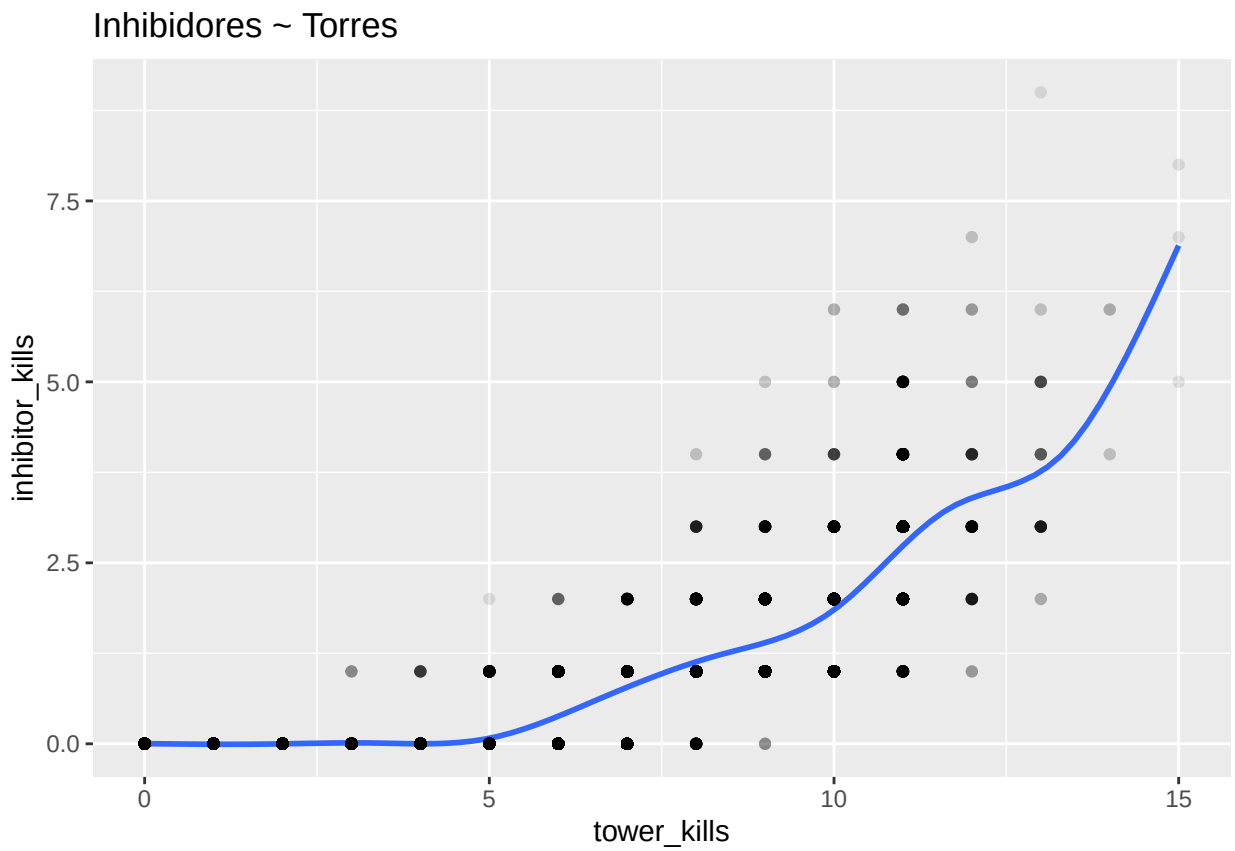

```
library(ggplot2)
p1 <- ggplot(df, aes(tower_kills, win)) +
  geom_smooth(method="glm", method.args=list(family=binomial()), se=FALSE) +
  geom_point(alpha=.1) + labs(title="Win ~ Torres")

p2 <- ggplot(df, aes(tower_kills, inhibitor_kills)) +
  geom_smooth(se=FALSE) + geom_point(alpha=.1) + labs(title="Inhibidores ~ Torres")

p3 <- ggplot(df, aes(inhibitor_kills, win)) +
  geom_smooth(method="glm", method.args=list(family=binomial()), se=FALSE) +
  geom_point(alpha=.1) + labs(title="Win ~ Inhibidores")
print(p1)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



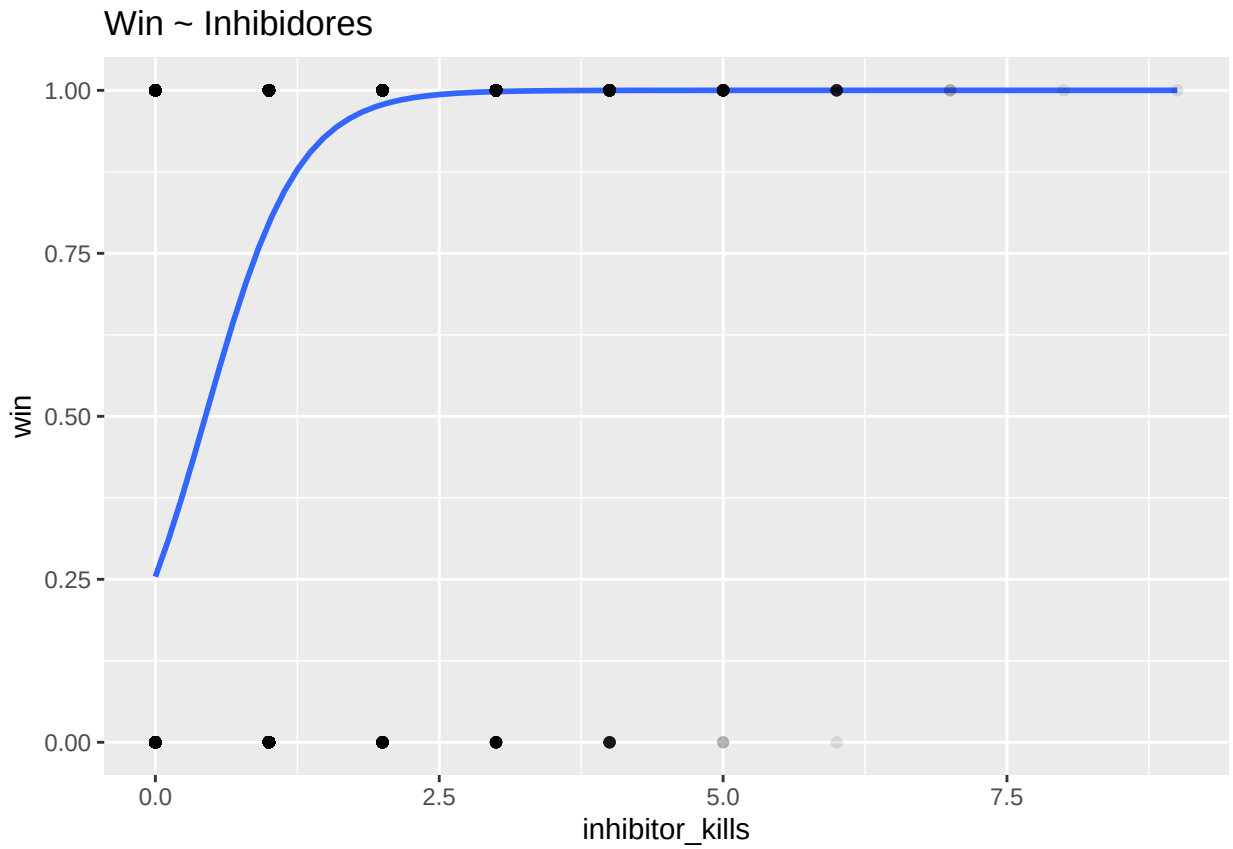

```
print(p2)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
print(p3)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



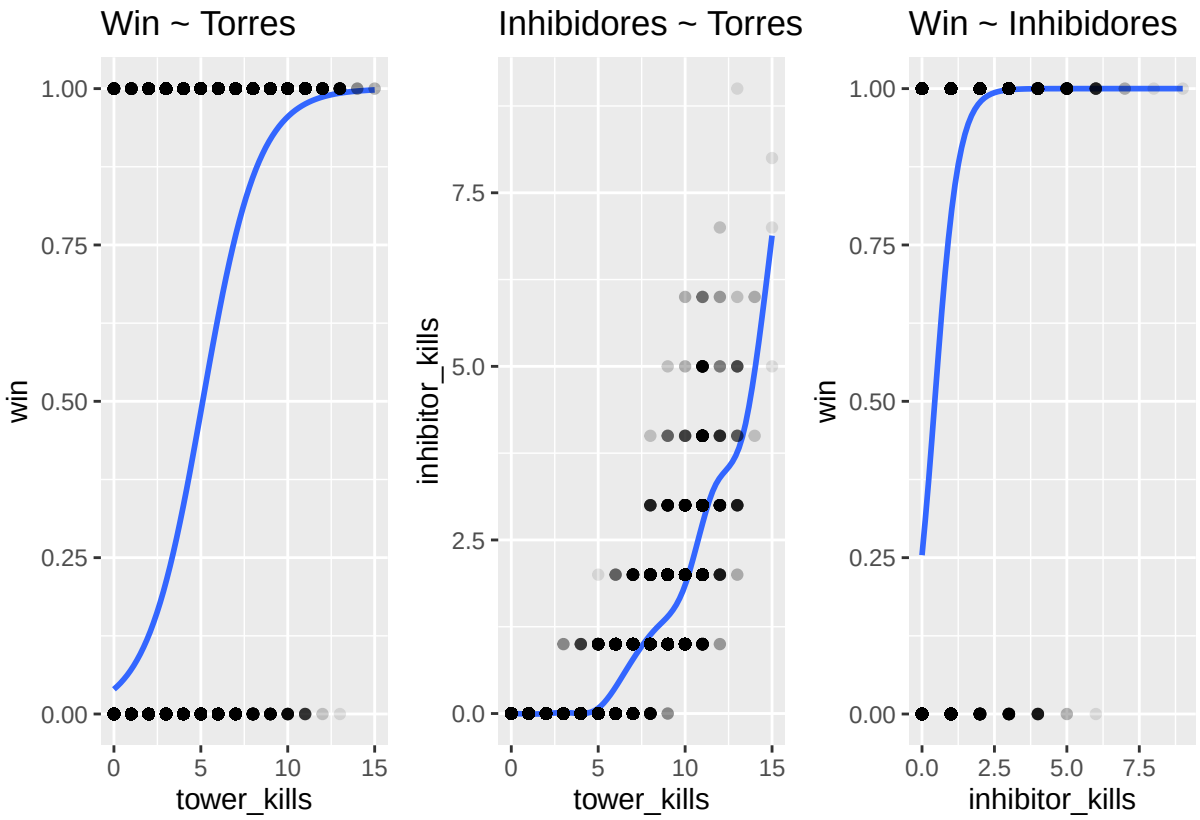

```
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.4.2
```

```
##
## Adjuntando el paquete: 'patchwork'
##
## The following object is masked from 'package:MASS':
##
##     area
```

```
p1 + p2 + p3
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using formula = 'y ~ x'
```


Win ~ Torres
win
1.00
0.75
0.50
0.25
0.00
0
5
10
15
tower_kills
Inhibidores ~ Torres
inhibitor_kills
7.5
5.0
2.5
0.0
0
5
10
15
tower_kills
Win ~ Inhibidores
win
1.00
0.75
0.50
0.25
0.00
0.0
2.5
5.0
7.5
inhibitor_kills