DA Competition

Christopher Daniel Hitijahubessy

2025-11-11

```
# Datasets
rm(list=ls()) # Clear the console
graphics.off() # Clear the graphics window
options(scipen=20) # Turn off scientific notation
if (!require("tidyverse")) install.packages("tidyverse")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("readr")) install.packages("readr")
if (!require("lubridate")) install.packages("lubridate")
if (!require("sqldf")) install.packages("sqldf")
```

```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

```
library(sqldf)
library(lubridate)
library(tidyverse)
library(knitr)
library(sqldf)
Constructors <- read_csv("DA Competition/Constructors.csv")

DriverPrice <- read_csv("DA Competition/DriverPrice.csv")

Drivers <- read_csv("DA Competition/Drivers.csv")

laps <- read_csv("DA Competition/laps.csv")

QualificationResult <- read_csv("DA Competition/QualificationResult.csv")

RaceResult <- read_csv("DA Competition/RaceResult.csv")

RaceSchedule <- read_csv("DA Competition/RaceSchedule.csv")
```

```
# Converting status in RaceResult
RaceResult <- RaceResult %>%
  mutate(status_conv = case_when(
    status %in% c("+2 Laps",
                  "+1 Lap",
                  "+5 Laps",
                  "+3 Laps",
                  "Finished",
                  "Lapped",
                  "+6 Laps") ~ "Finished",
    TRUE ~ "Not Finished"
  ))
```

```r
## Convert to points based on position and status in RaceResult dataset
racepoints <- RaceResult %>%
  mutate(points_conv = case_when(
    (position == 1 & status_conv == "Finished") ~ 25,
    (position == 2 & status_conv == "Finished") ~ 18,
    (position == 3 & status_conv == "Finished") ~ 15,
    (position == 4 & status_conv == "Finished")~ 12,
    (position == 5 & status_conv == "Finished") ~ 10,
    (position == 6 & status_conv == "Finished") ~ 8,
    (position == 7 & status_conv == "Finished") ~ 6,
    (position == 8 & status_conv == "Finished") ~ 4,
    (position == 9 & status_conv == "Finished") ~ 1,
    (position == 10 & status_conv == "Finished") ~ 0,
    (position %in% c(11, 12, 13, 14, 15, 16, 17, 18, 19, 20) &
       status_conv == "Finished") ~ 0,
    TRUE ~ -20
  ),
  point_gain_loss = ifelse( # Calculate points from position gained and lost
    is.na(grid_position) | is.na(position),
    0,
    grid_position - position
  ))

# Sum up points
racepoints <- racepoints %>%
  mutate(points_real = points_conv + point_gain_loss)

# Combine DriverPrice and racepoints
DriverPriceResult <- left_join(DriverPrice, racepoints,
                               by = c("Driver Code"= "driver_code"))

# Combining points per driver
countPoints <- DriverPriceResult %>%
  group_by(`Driver Code`, `Driver`)%>%
  summarise(
    `Price` = max(Price),
    `Total Points` = sum(points_real, na.rm = TRUE),
    .groups = "drop") %>%
  arrange(desc(`Total Points`))
kable(countPoints)
```

Driver Code	Driver	Price	Total Points
VER	Max Verstappen	29.1	1710
HAM	Lewis Hamiliton	22.7	1270
NOR	Lando Norris	30.5	861
LEC	Charles Leclerc	23.0	589
PIA	Oscar Piastri	26.3	463
SAI	Carlos Sainz	6.9	350
RUS	George Russell	23.3	329
BEA	Oliver Bearman	7.7	48
OCO	Esteban Ocon	7.1	-28
ALO	Fernando Alonso	6.1	-29

Driver Code	Driver	Price	Total Points
ANT	Kimi Antonelli	15.7	-29
HAD	Isack Hadjar	6.5	-33
BOR	Gabriel Bortoleto	6.7	-45
COL	Franco Colapinto	4.5	-60
STR	Lance Stroll	8.7	-97
HUL	Nico Hulkenberg	7.4	-98
LAW	Liam Lawson	7.1	-126
ALB	Alexander Albon	13.0	-128
GAS	Pierre Gasly	4.5	-182
TSU	Yuki Tsunoda	10.2	-400

```r
### Miscellaneous - checking NA's in position and grid_position column
RR_Miscellaneous <- racepoints %>%
  filter(is.na(position) | is.na(grid_position))
```

Qualification Result Table

```r
# Converting positions to points
QualificationResult <- QualificationResult %>%
  mutate(points = case_when(
  position == 1 ~ 10,
  position == 2 ~ 9,
  position == 3 ~ 8,
  position == 4 ~ 7,
  position == 5 ~ 6,
  position == 6 ~ 5,
  position == 7 ~ 4,
  position == 8 ~ 3,
  position == 9 ~ 2,
  position == 10 ~ 1,
  TRUE ~ 0
))

#Sum all the points
qualifypoints <- QualificationResult %>%
  group_by(driver_code) %>%
  summarise(total_points_qualify = sum(points, na.rm = TRUE))

kable(qualifypoints)
```

driver_code	total_points_qualify
AIT	0
ALB	132
ALO	243
ANT	74
BEA	10
BOR	11
BOT	351
COL	2
DEV	0

driver_code	total_points_qualify
DOO	0
FIT	0
GAS	210
GIO	8
GRO	0
HAD	38
HAM	695
HUL	52
KUB	0
KVY	13
LAT	1
LAW	29
LEC	671
MAG	60
MAZ	0
MSC	13
NOR	588
OCO	148
PER	420
PIA	303
RAI	3
RIC	137
RUS	462
SAI	522
SAR	5
STR	116
TSU	79
VER	941
VET	34
ZHO	10

```r
# Combining countPoints and qualifypoints (excluding fastest lap)
union_data <- left_join(countPoints, qualifypoints,
                        by = c("Driver Code" = "driver_code"))
union_data <- union_data %>%
  mutate(subtotal = `Total Points` + total_points_qualify) %>%
  arrange(desc(subtotal))
kable(union_data)
```

Driver Code	Driver	Price	Total Points	total_points_qualify	subtotal
VER	Max Verstappen	29.1	1710	941	2651
HAM	Lewis Hamiliton	22.7	1270	695	1965
NOR	Lando Norris	30.5	861	588	1449
LEC	Charles Leclerc	23.0	589	671	1260
SAI	Carlos Sainz	6.9	350	522	872
RUS	George Russell	23.3	329	462	791
PIA	Oscar Piastri	26.3	463	303	766
ALO	Fernando Alonso	6.1	-29	243	214
OCO	Esteban Ocon	7.1	-28	148	120
BEA	Oliver Bearman	7.7	48	10	58
ANT	Kimi Antonelli	15.7	-29	74	45

Driver Code	Driver	Price	Total Points	total_points_qualify	subtotal
GAS	Pierre Gasly	4.5	-182	210	28
STR	Lance Stroll	8.7	-97	116	19
HAD	Isack Hadjar	6.5	-33	38	5
ALB	Alexander Albon	13.0	-128	132	4
BOR	Gabriel Bortoleto	6.7	-45	11	-34
HUL	Nico Hulkenberg	7.4	-98	52	-46
COL	Franco Colapinto	4.5	-60	2	-58
LAW	Liam Lawson	7.1	-126	29	-97
TSU	Yuki Tsunoda	10.2	-400	79	-321

```
laps_modified <- laps %>% mutate(LapTimeInSeconds = (as.numeric(str_extract(LapTime, "\\d{2}(?=:\\d{2}\

# add a new id column in RaceResult and laps_modified that assigns a unique id to each event
RaceResult$event_id <- paste(RaceResult$year,RaceResult$event_name)
laps_modified$event_id <- paste(laps_modified$year,laps_modified$event_name)

# construct a table that displays the driver and the minimum lap time per event. Note laps_modified con
LapTimePerDriver <- sqldf("SELECT Driver, MIN(LapTimeInSeconds), event_id FROM laps_modified WHERE LapT


LapTimePerDriver <- LapTimePerDriver %>%
  mutate(points_add = 10) %>%
  rename(driver_code = Driver)

fastest_lap_points_acc <- LapTimePerDriver %>%
  group_by(driver_code) %>%
  summarise(point_fastest = sum(points_add))

union_data <- left_join(union_data, fastest_lap_points_acc, by = c("Driver Code" = "driver_code"))

union_data <- union_data %>%
  mutate(point_fastest_conv = case_when(
    !is.na(point_fastest) ~ point_fastest,
    TRUE ~ 0))

union_data <- union_data %>%
  mutate(points_acc = subtotal + point_fastest_conv) %>%
  arrange(desc(points_acc))
kable(union_data)
```

Driver Code	Driver	Price	Total Points	total_points_qualify	subtotal	point_fastest	point_fastest_conv	points_acc
VER	Max Verstappen	29.1	1710	941	2651	260	260	2911
HAM	Lewis Hamiliton	22.7	1270	695	1965	200	200	2165
NOR	Lando Norris	30.5	861	588	1449	160	160	1609
LEC	Charles Leclerc	23.0	589	671	1260	40	40	1300
SAI	Carlos Sainz	6.9	350	522	872	40	40	912

Driver Code	Driver	Price	Total Points	total_points_qualify	total	point_fastest	point_fastest_c	points_acc
RUS	George Russell	23.3	329	462	791	100	100	891
PIA	Oscar Piastri	26.3	463	303	766	60	60	826
ALO	Fernando Alonso	6.1	-29	243	214	20	20	234
OCO	Esteban Ocon	7.1	-28	148	120	10	10	130
ANT	Kimi Antonelli	15.7	-29	74	45	30	30	75
BEA	Oliver Bearman	7.7	48	10	58	NA	0	58
GAS	Pierre Gasly	4.5	-182	210	28	10	10	38
STR	Lance Stroll	8.7	-97	116	19	NA	0	19
HAD	Isack Hadjar	6.5	-33	38	5	NA	0	5
ALB	Alexander Albon	13.0	-128	132	4	NA	0	4
BOR	Gabriel Bortoleto	6.7	-45	11	-34	NA	0	-34
HUL	Nico Hulkenberg	7.4	-98	52	-46	NA	0	-46
COL	Franco Colapinto	4.5	-60	2	-58	NA	0	-58
LAW	Liam Lawson	7.1	-126	29	-97	NA	0	-97
TSU	Yuki Tsunoda	10.2	-400	79	-321	10	10	-311

```r
racepoints_year = racepoints %>%
  group_by(year, driver_code) %>%
  summarise(point_year = sum(points_real), .groups = "drop")
countPoints_year <- racepoints_year %>%
  group_by(driver_code, year)%>%
  summarise(
    `Total Points per Year` = sum(point_year, na.rm = TRUE),
    .groups = "drop") %>%
  arrange(desc(`Total Points per Year`), desc(year))
head(countPoints_year)
```

```
## # A tibble: 6 x 3
##   driver_code  year `Total Points per Year`
##   <chr>       <dbl>                   <dbl>
## 1 VER          2023                     493
## 2 VER          2022                     386
## 3 HAM          2021                     360
## 4 HAM          2020                     341
## 5 VER          2021                     307
## 6 PIA          2025                     301
```

```r
qualifypoints_year <- QualificationResult %>%
  group_by(driver_code, year) %>%
  summarise(total_points_qualify = sum(points, na.rm = TRUE))
```

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

```r
head(qualifypoints)
```

```
## # A tibble: 6 x 2
##   driver_code total_points_qualify
##   <chr>                      <dbl>
## 1 AIT                            0
## 2 ALB                          132
## 3 ALO                          243
## 4 ANT                           74
## 5 BEA                           10
## 6 BOR                           11
```

```r
# Combining countPoints and qualifypoints (excluding fastest lap)
union_data_year <- left_join(countPoints_year, qualifypoints_year,
                        by = c("driver_code", "year"))
union_data_year <- union_data_year %>%
  mutate(subtotal = `Total Points per Year` + total_points_qualify) %>%
  select(driver_code, year, subtotal) %>%
  arrange(desc(subtotal))
head(union_data_year)
```

```
## # A tibble: 6 x 3
##   driver_code  year subtotal
##   <chr>       <dbl>    <dbl>
## 1 VER          2023      663
## 2 VER          2022      571
## 3 HAM          2021      555
## 4 VER          2021      496
## 5 HAM          2020      490
## 6 PIA          2025      461
```

```r
LapTimePerDriver_year <- laps_modified %>%
  filter(
    !is.na(LapTimeInSeconds) & LapTimeInSeconds > 0,
    IsAccurate == TRUE,
    !Deleted
  ) %>%
  group_by(event_id) %>%
  filter(LapTimeInSeconds == min(LapTimeInSeconds, na.rm = TRUE)) %>%
  ungroup() %>%
  distinct(event_id, .keep_all = TRUE) %>%
  select(driver_code = Driver,
         fastest_lap_time = LapTimeInSeconds,
         event_id,
         year)

LapTimePerDriver_year <- LapTimePerDriver_year %>% mutate(
  points_add = 10)
```

```r
fastest_lap_points_year <- LapTimePerDriver_year %>%
  group_by(driver_code, year) %>%
  summarise(point_fastest = sum(points_add))
```

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

```r
union_data_year <- left_join(union_data_year, fastest_lap_points_year,
                        by = c("driver_code", "year"))

union_data_year <- union_data_year %>%
  mutate(point_fastest_conv = case_when(
    !is.na(point_fastest) ~ point_fastest,
    TRUE ~ 0))

union_data_year <- union_data_year %>%
  mutate(points_year = subtotal + point_fastest_conv) %>%
  select(driver_code, year, points_year) %>%
  arrange(desc(points_year))
head(union_data_year)
```

```
## # A tibble: 6 x 3
##   driver_code  year points_year
##   <chr>       <dbl>       <dbl>
## 1 VER          2023         753
## 2 VER          2022         621
## 3 HAM          2021         615
## 4 VER          2021         556
## 5 HAM          2020         550
## 6 PIA          2025         511
```

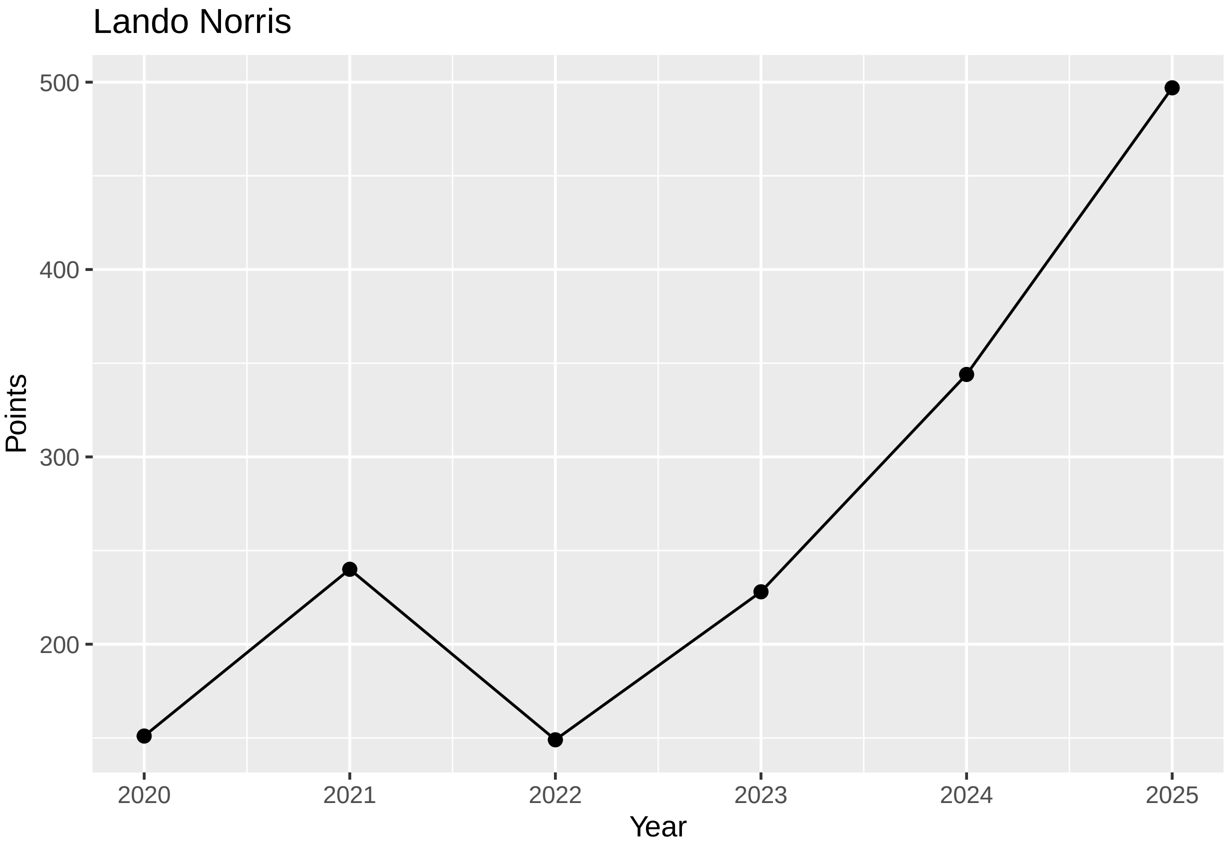
```r
points_year <- left_join(DriverPrice, union_data_year, by = c("Driver Code" = "driver_code"))
head(points_year)
```

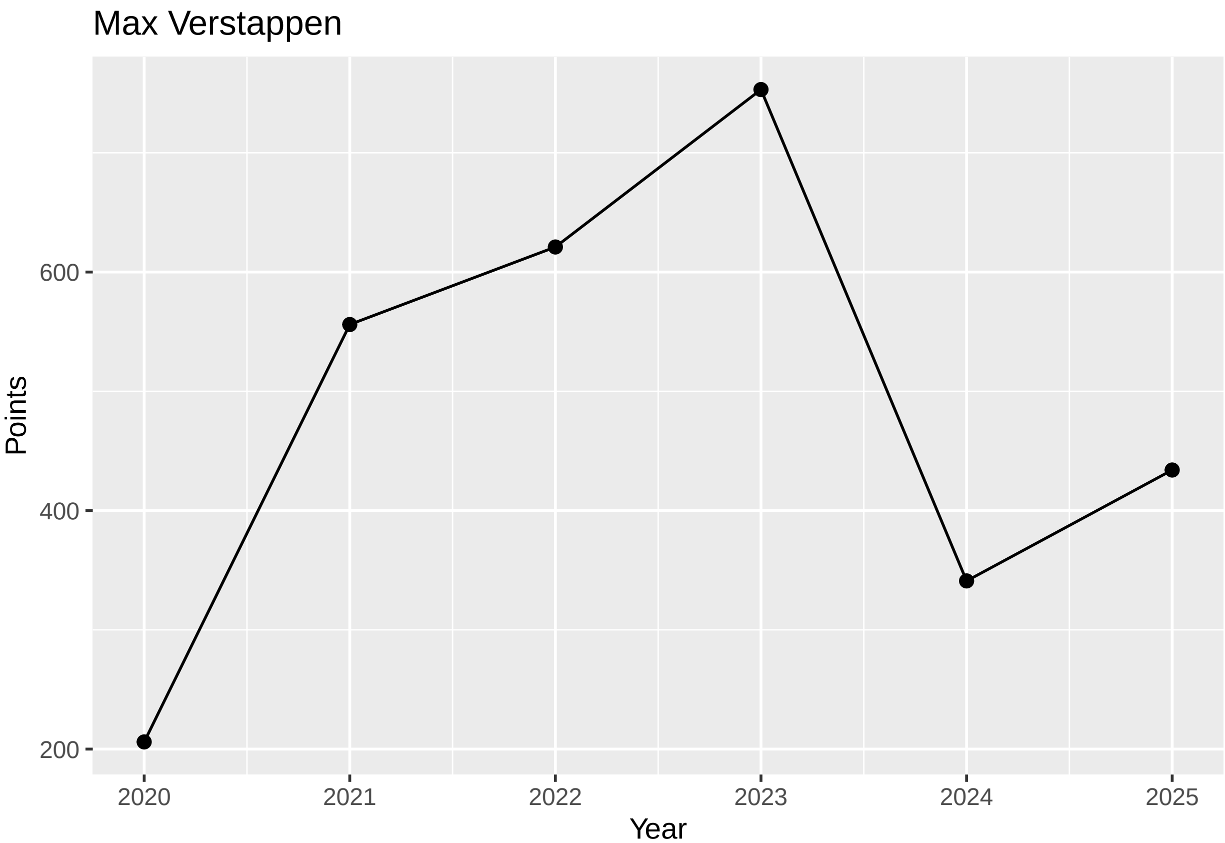
```
## # A tibble: 6 x 5
##   `Driver Code` Driver       Price  year points_year
##   <chr>         <chr>        <dbl> <dbl>       <dbl>
## 1 NOR           Lando Norris  30.5  2025         497
## 2 NOR           Lando Norris  30.5  2024         344
## 3 NOR           Lando Norris  30.5  2021         240
## 4 NOR           Lando Norris  30.5  2023         228
## 5 NOR           Lando Norris  30.5  2020         151
## 6 NOR           Lando Norris  30.5  2022         149
```

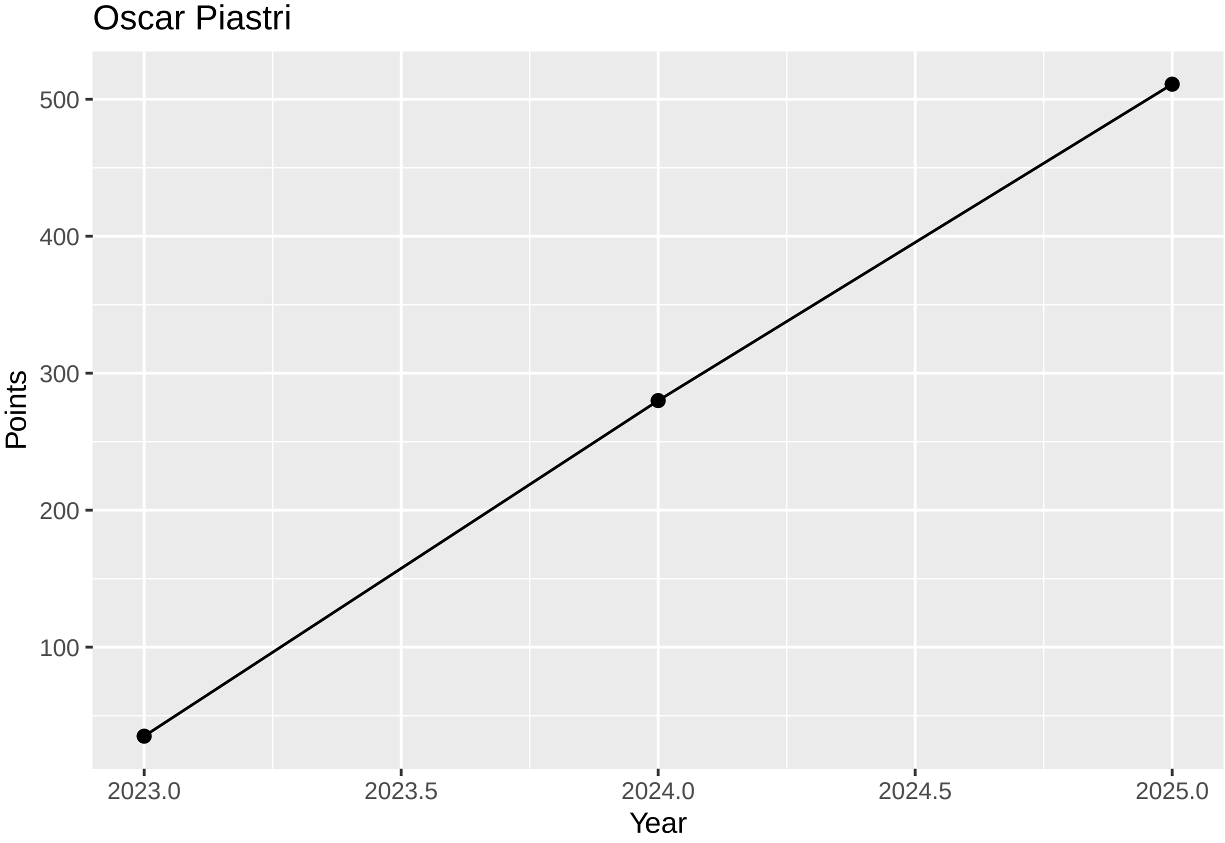
```r
points_by_year <- points_year %>% arrange(year) %>% pivot_wider(
  names_from = year,
  values_from = points_year,
  id_cols = c(`Driver Code`, Driver),
  values_fill = 0
) %>%
  arrange(desc(`2025`))
head(points_by_year)
```

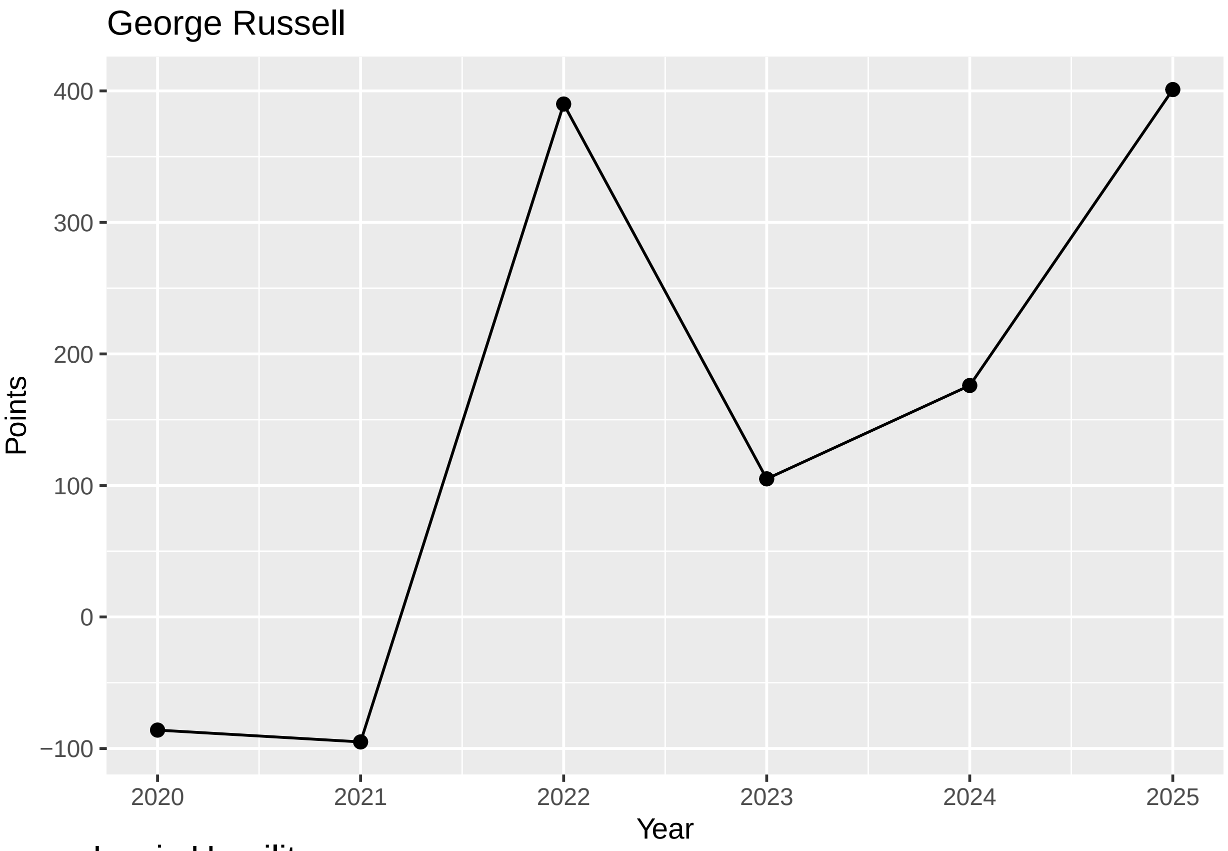
```
## # A tibble: 6 x 8
##   `Driver Code` Driver          `2020` `2021` `2022` `2023` `2024` `2025`
##   <chr>         <chr>            <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 PIA           Oscar Piastri        0      0      0     35    280    511
## 2 NOR           Lando Norris       151    240    149    228    344    497
## 3 VER           Max Verstappen     206    556    621    753    341    434
## 4 RUS           George Russell     -86    -95    390    105    176    401
## 5 LEC           Charles Leclerc     47    195    403    110    297    248
## 6 HAM           Lewis Hamiliton    550    615    327    275    211    187
```

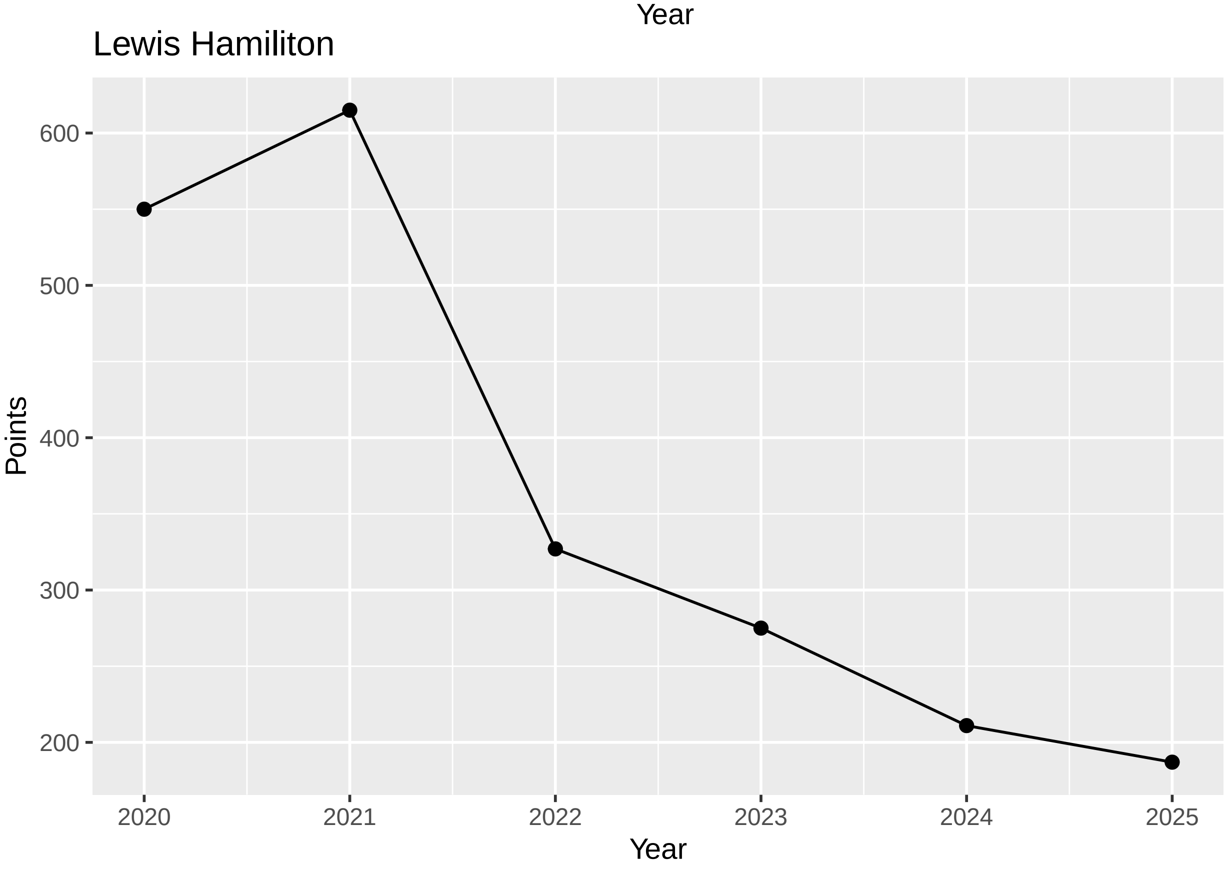
```
all_drivers <- unique(points_year$Driver)

for (current_driver in all_drivers){
  driver_data <- points_year %>%
    filter(Driver == current_driver)
  driver_plot <- ggplot(driver_data, aes(x = year, y = points_year)) +
    geom_line() +
    geom_point(size = 2) +
    labs(x = "Year", y = "Points",
         title = current_driver)
  print(driver_plot)
}
```

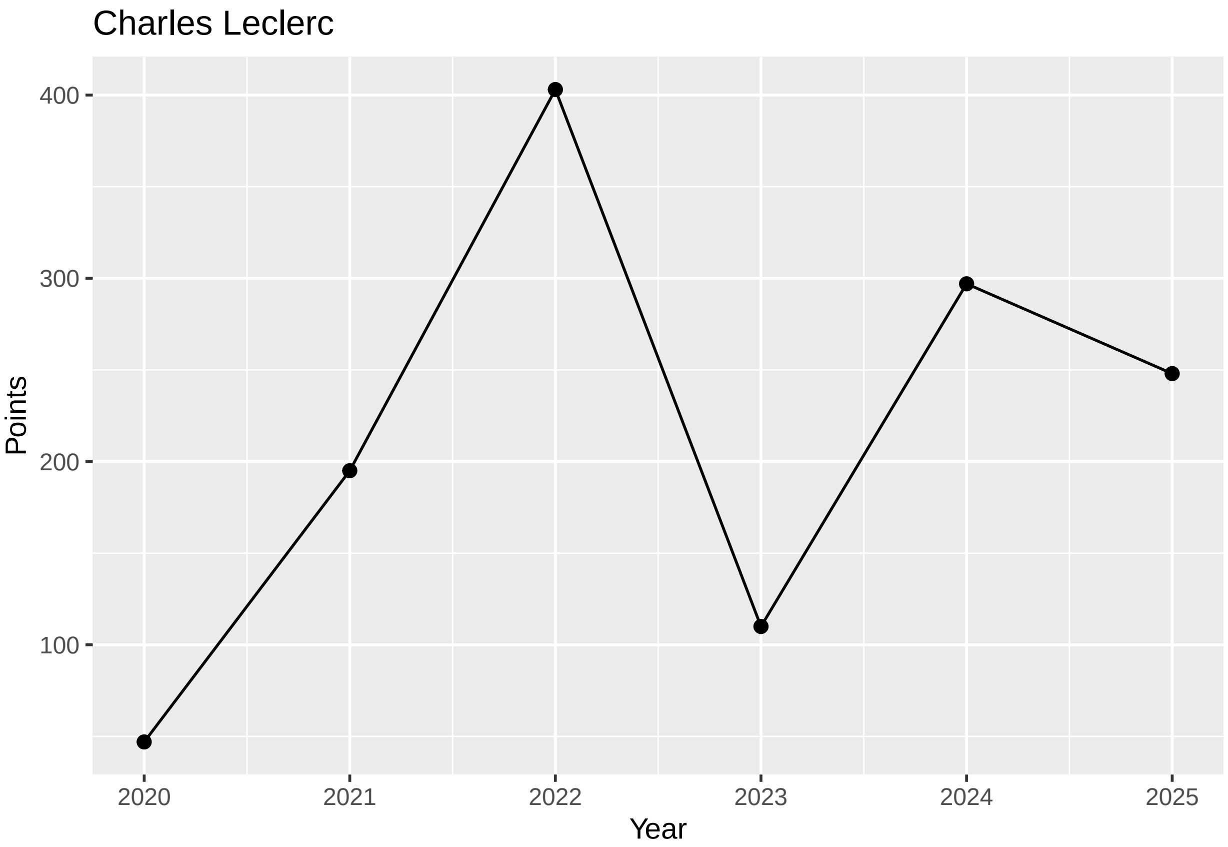



Max Verstappen
Points
600
400
200
2020
2021
2022
2023
2024
2025
Year

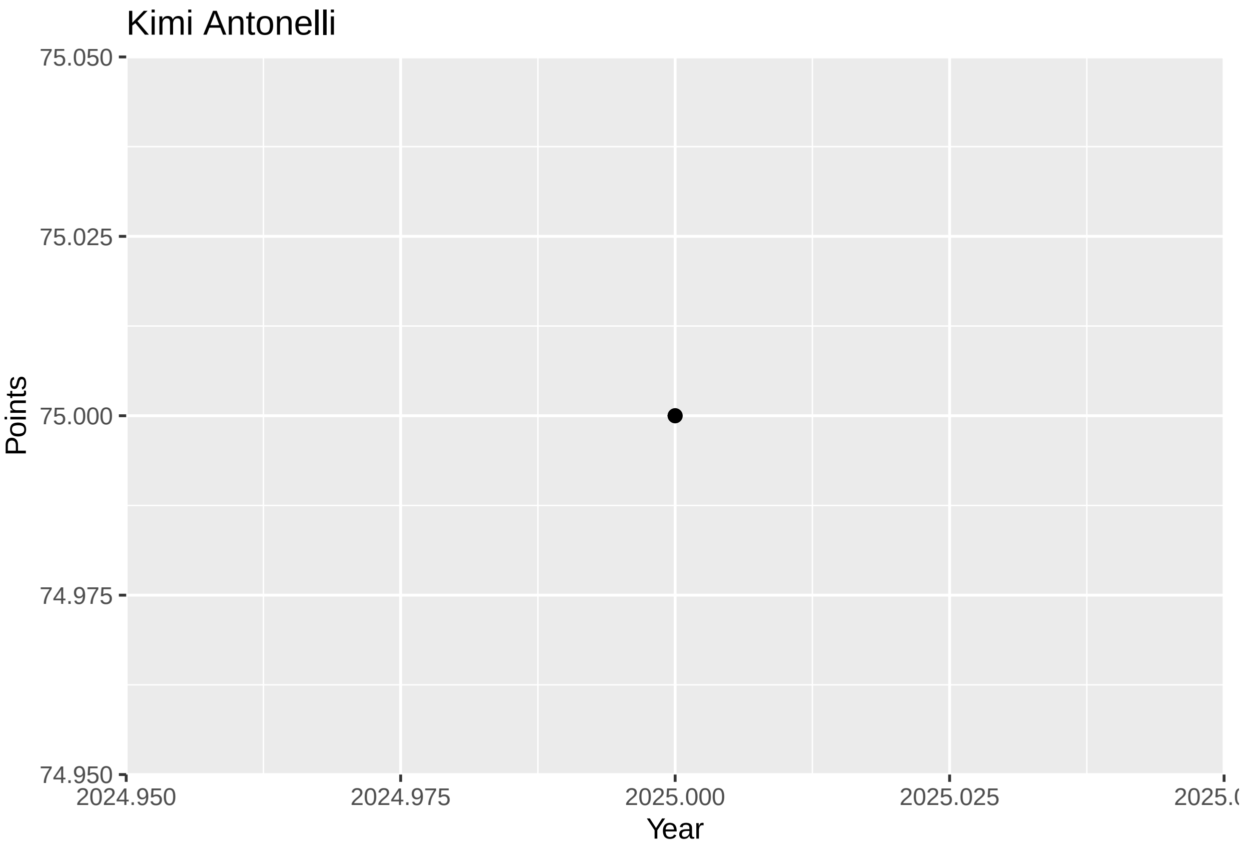

Oscar Piastri
Points
500
400
300
200
100
2023.0
2023.5
2024.0
2024.5
2025.0
Year

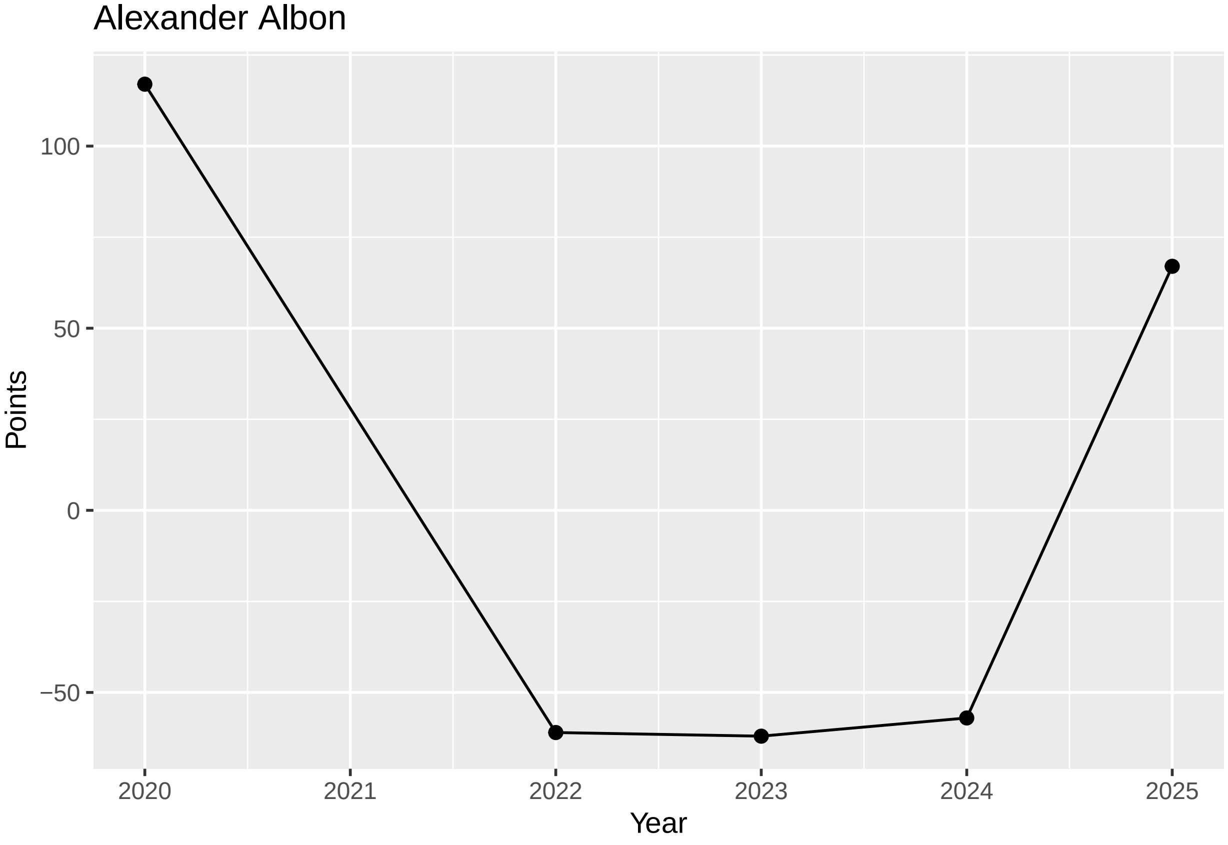

George Russell
Points
Year
400
300
200
100
0
−100
2020
2021
2022
2023
2024
2025

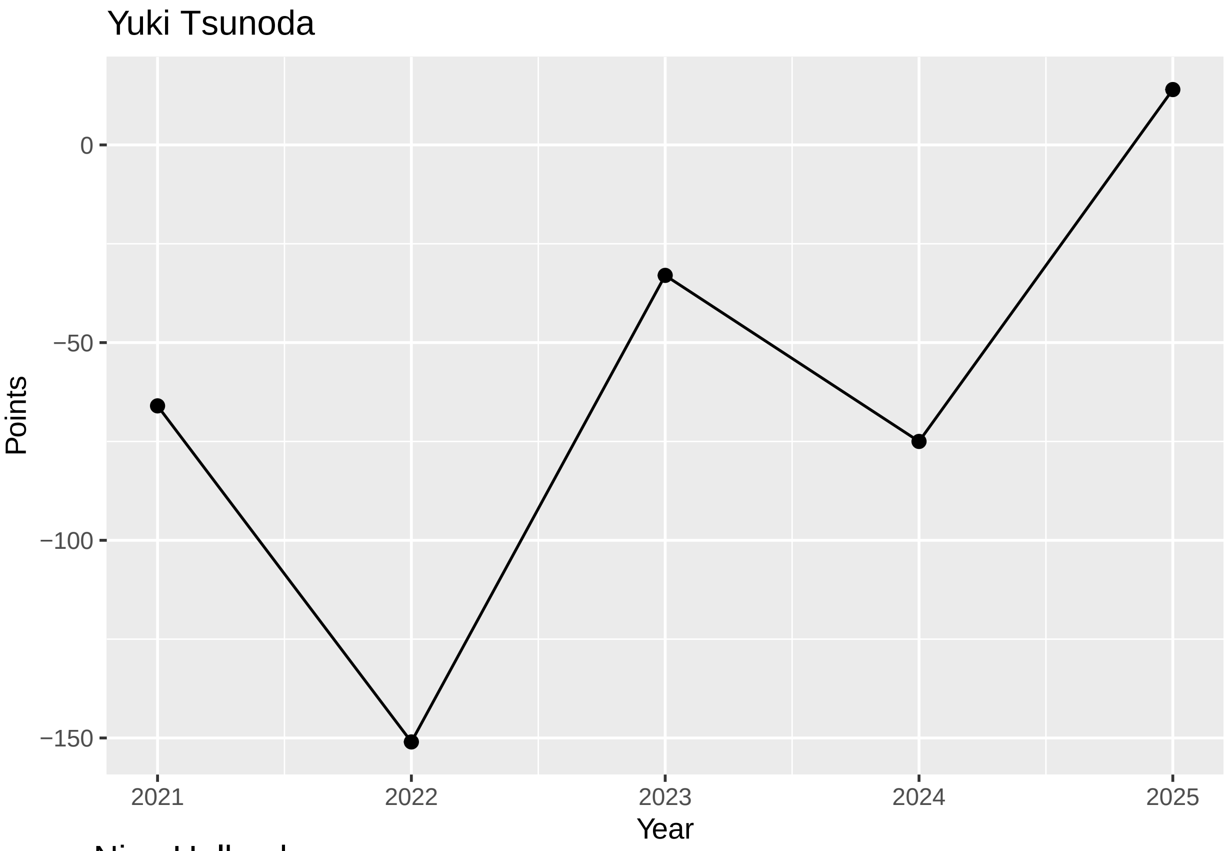

Lewis Hamiliton
Points
Year
600
500
400
300
200
2020
2021
2022
2023
2024
2025

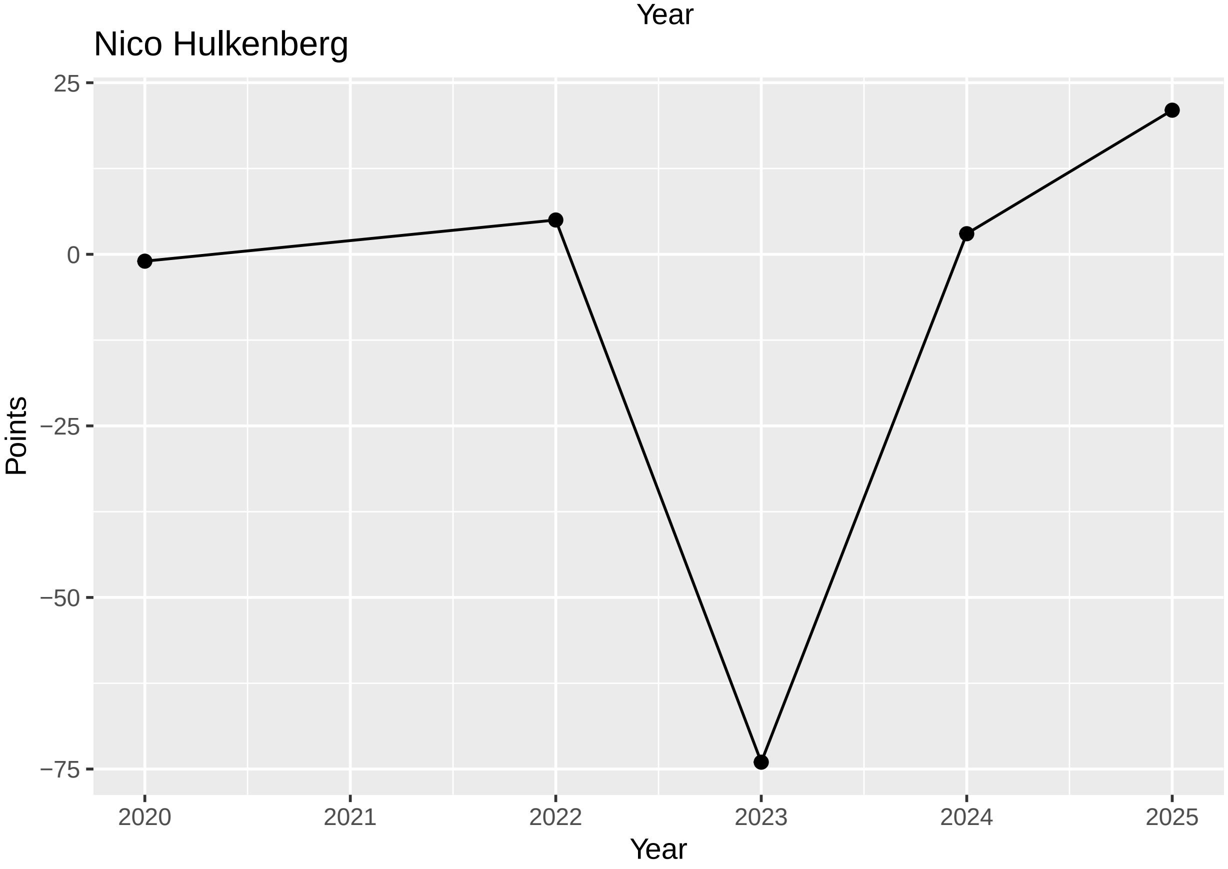


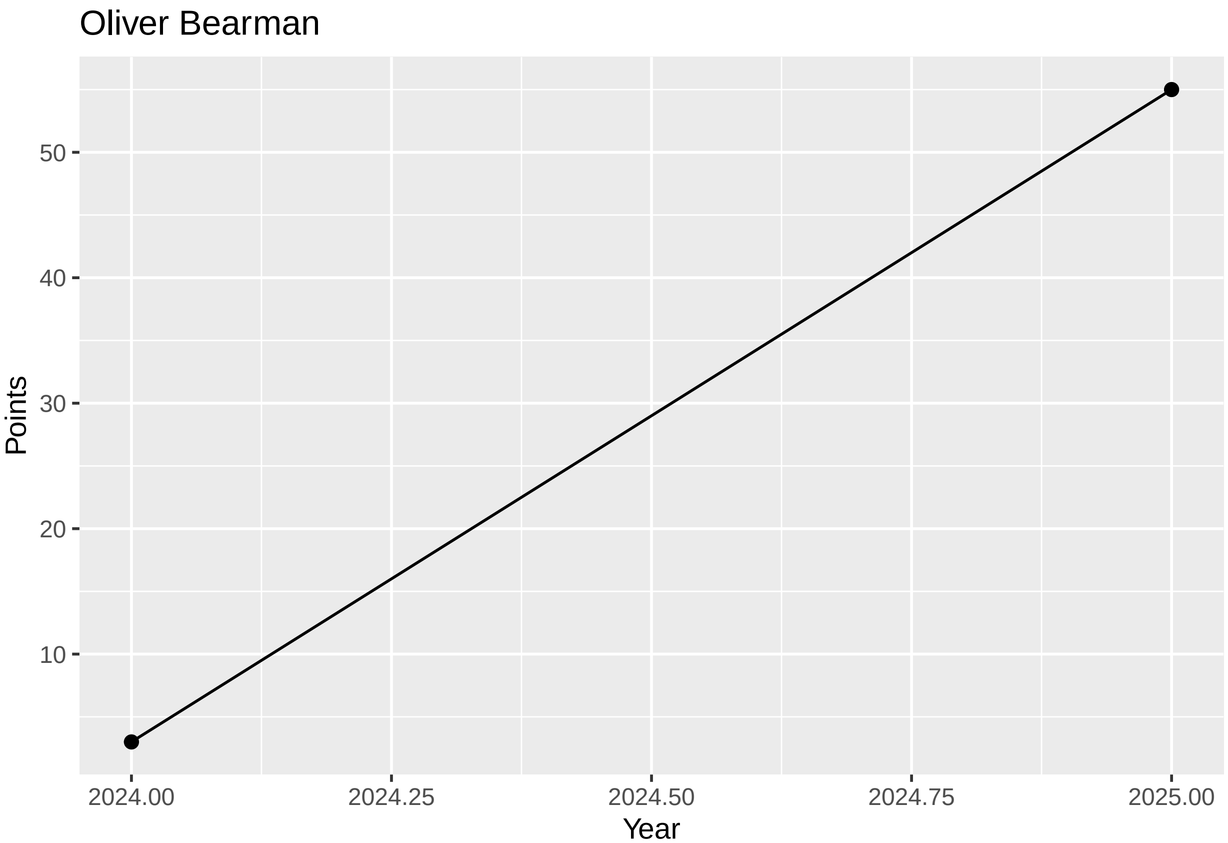
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

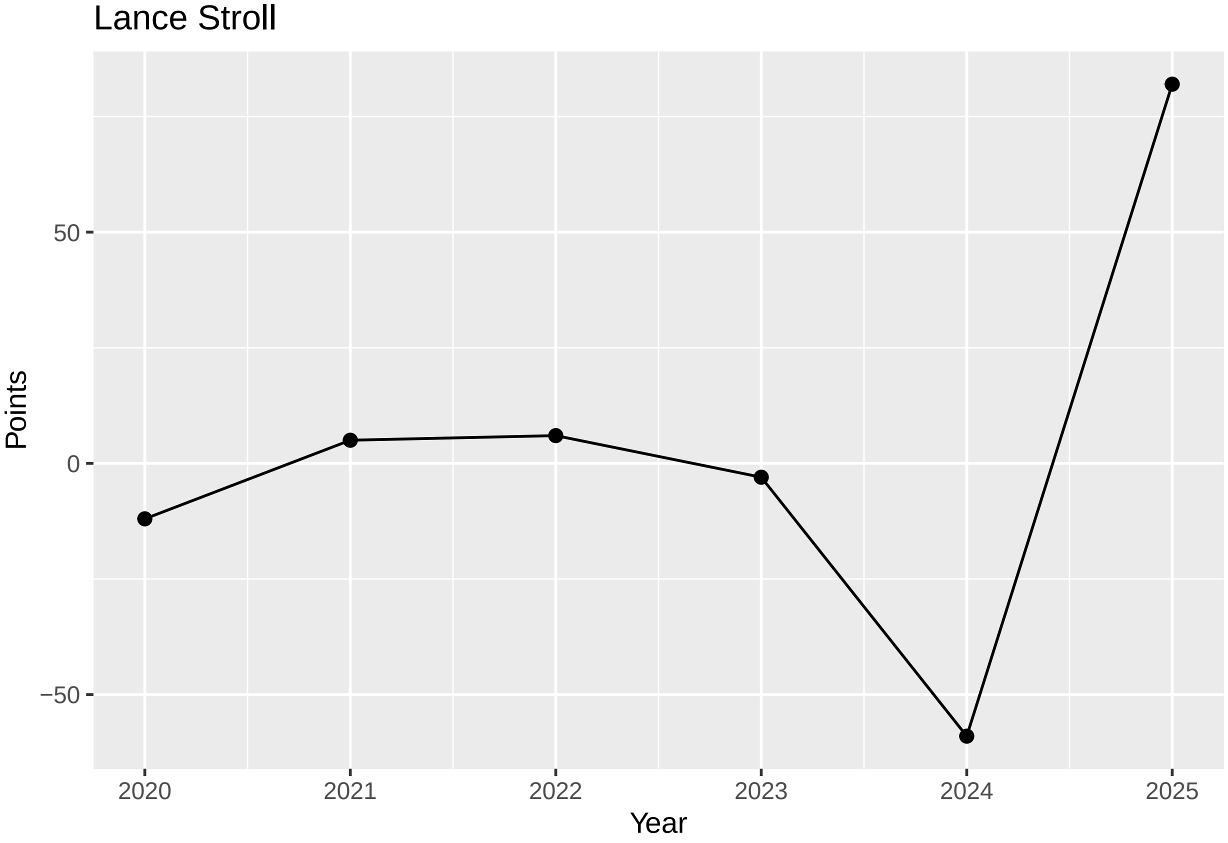

Kimi Antonelli
Points
75.050
75.025
75.000
74.975
74.950
2024.950
2024.975
2025.000
2025.025
2025.0
Year

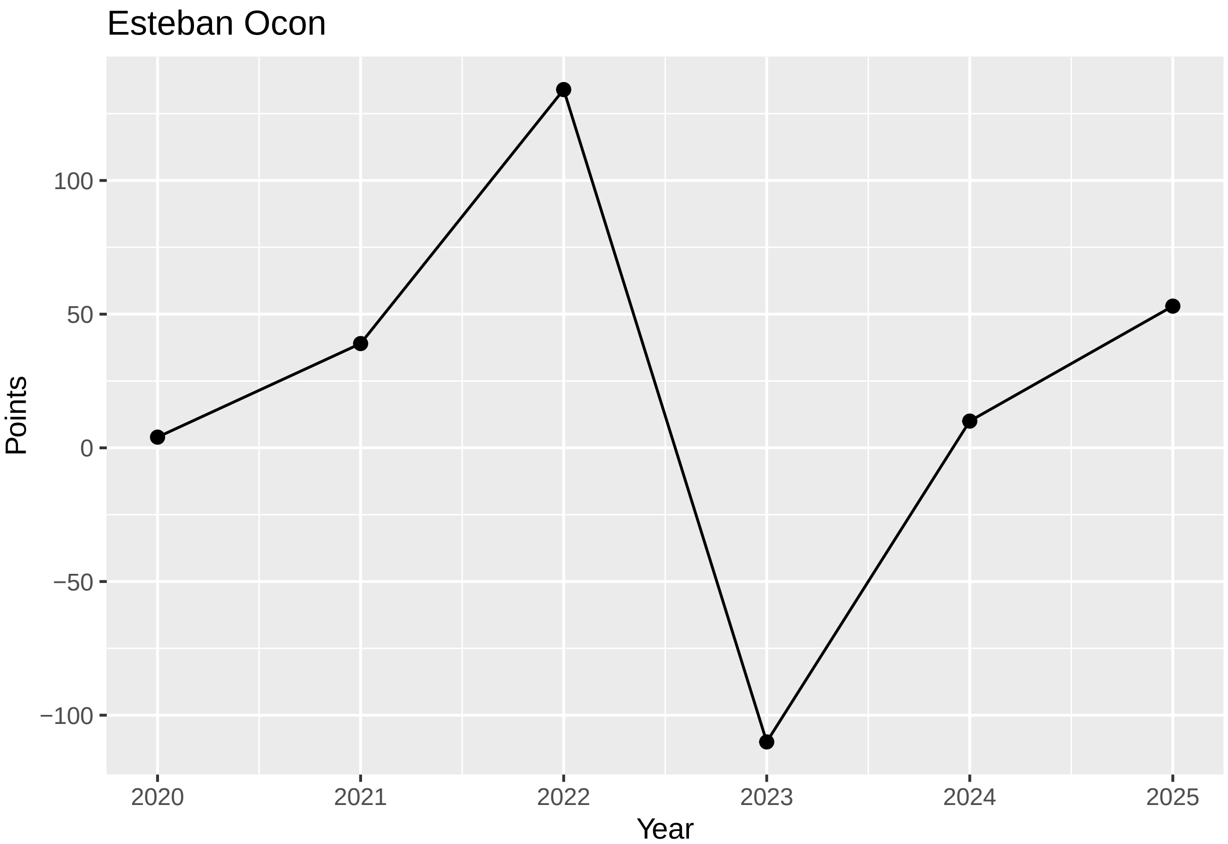

Alexander Albon
Points
100
50
0
−50
2020
2021
2022
2023
2024
2025
Year

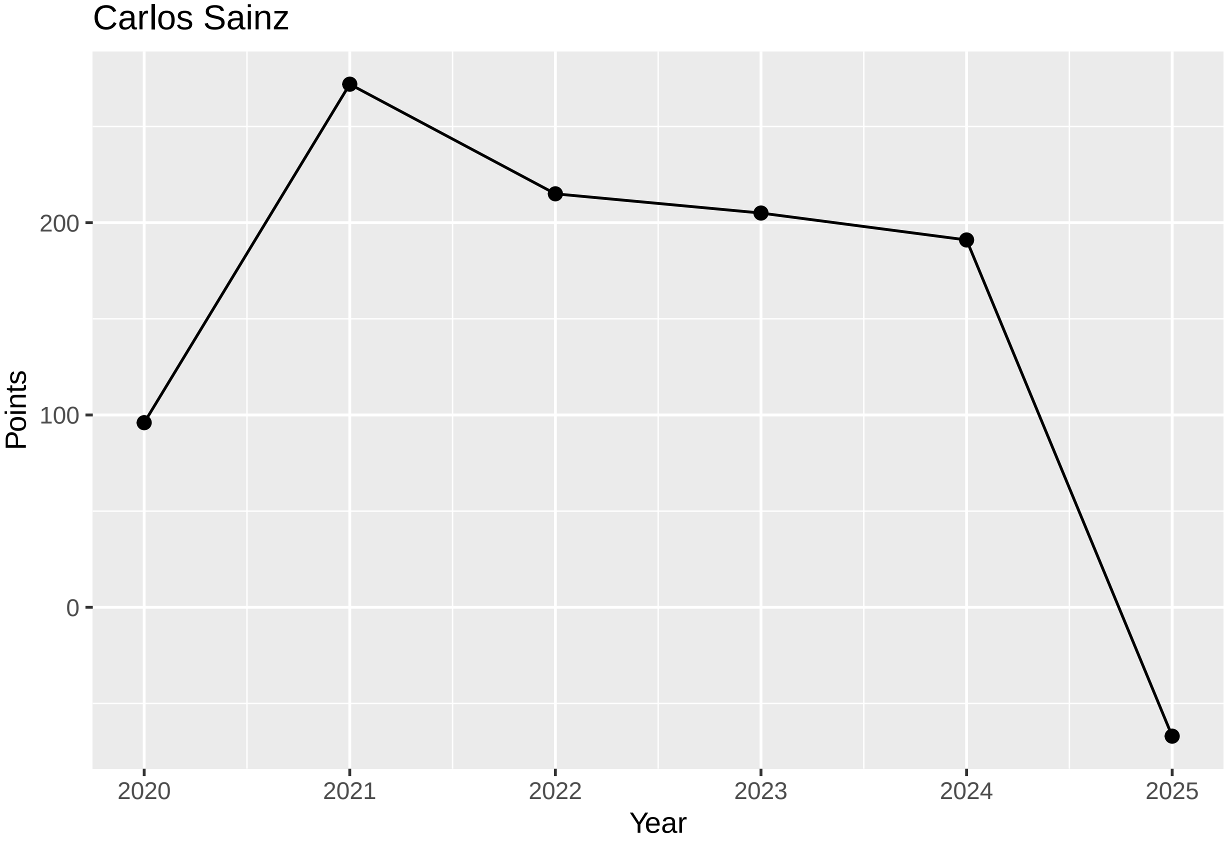

Yuki Tsunoda
Points
0
−50
−100
−150
2021
2022
2023
2024
2025
Year


Nico Hulkenberg
Points
25
0
−25
−50
−75
2020
2021
2022
2023
2024
2025
Year

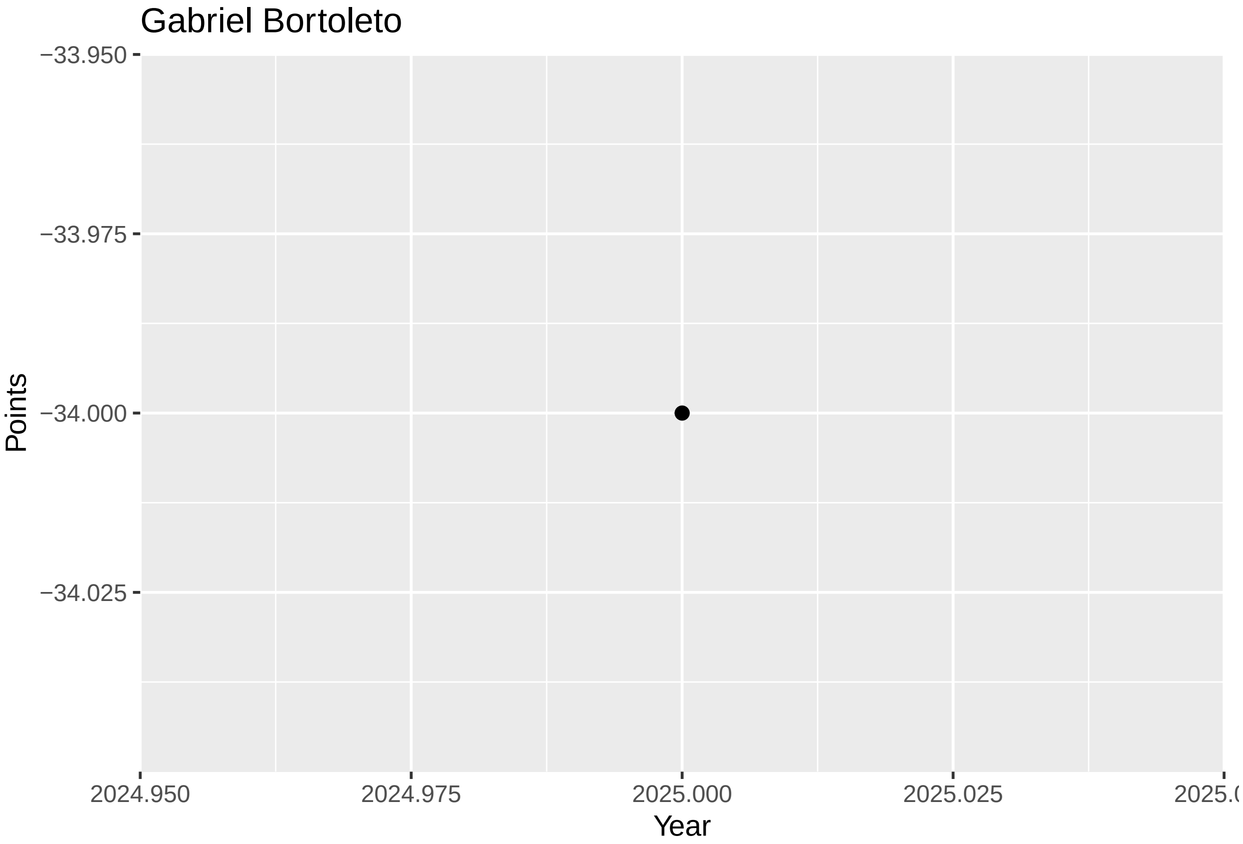

Oliver Bearman
Points
50
40
30
20
10
2024.00
2024.25
2024.50
2024.75
2025.00
Year


Lance Stroll
Points
50
0
−50
2020
2021
2022
2023
2024
2025
Year

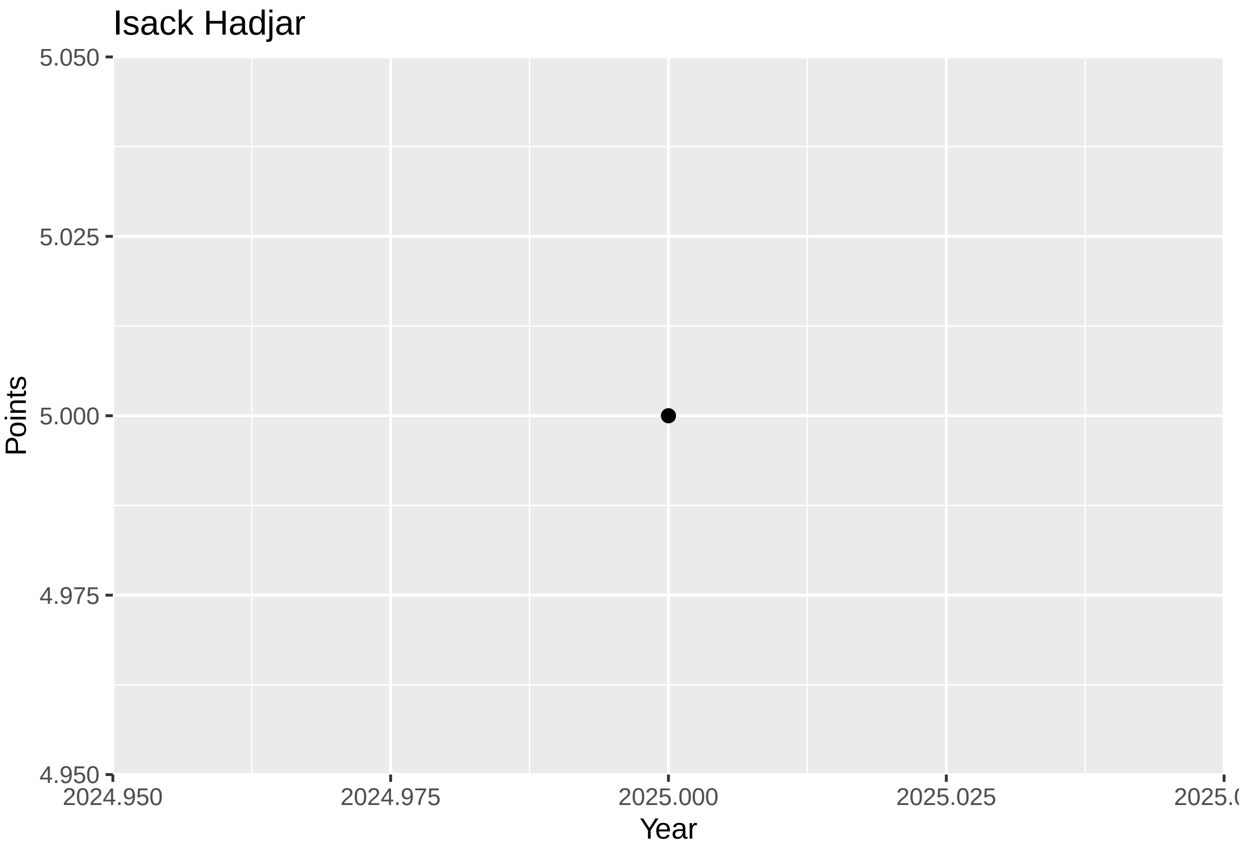


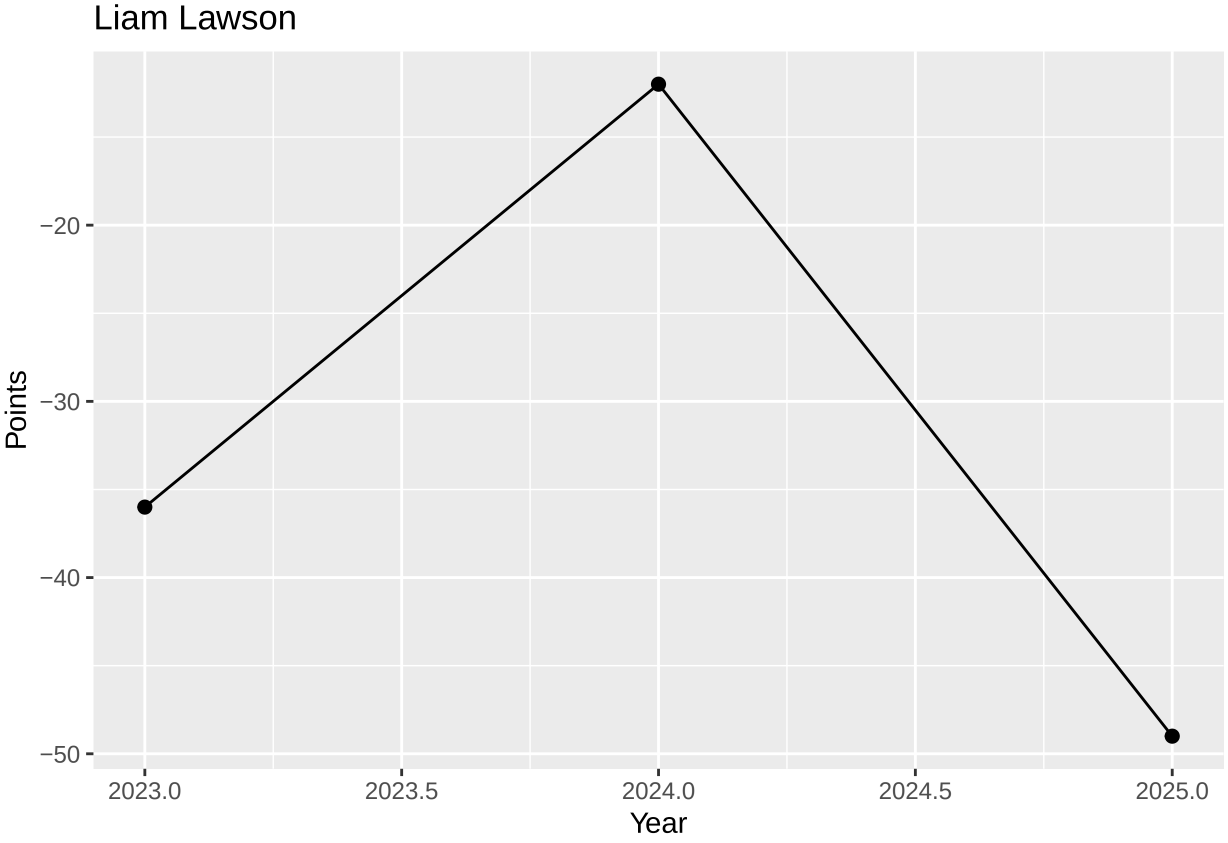


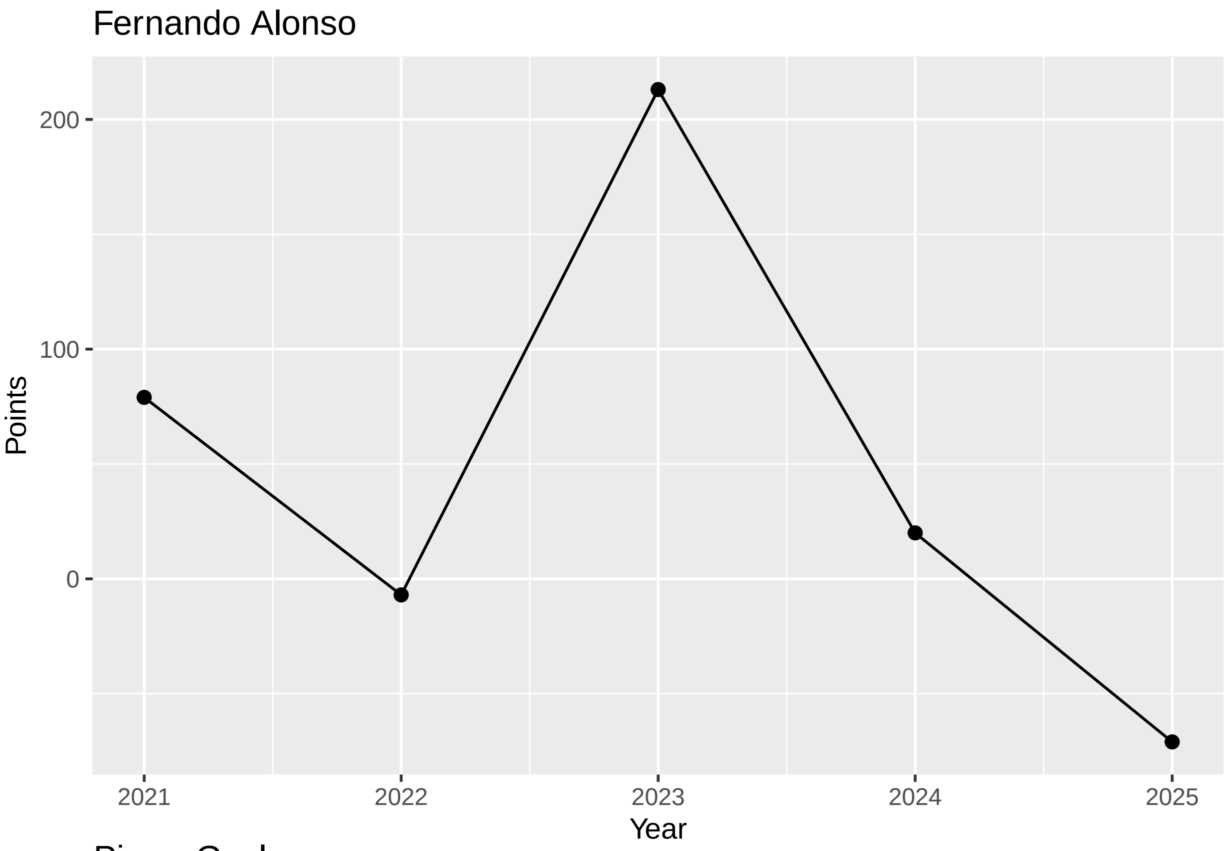
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

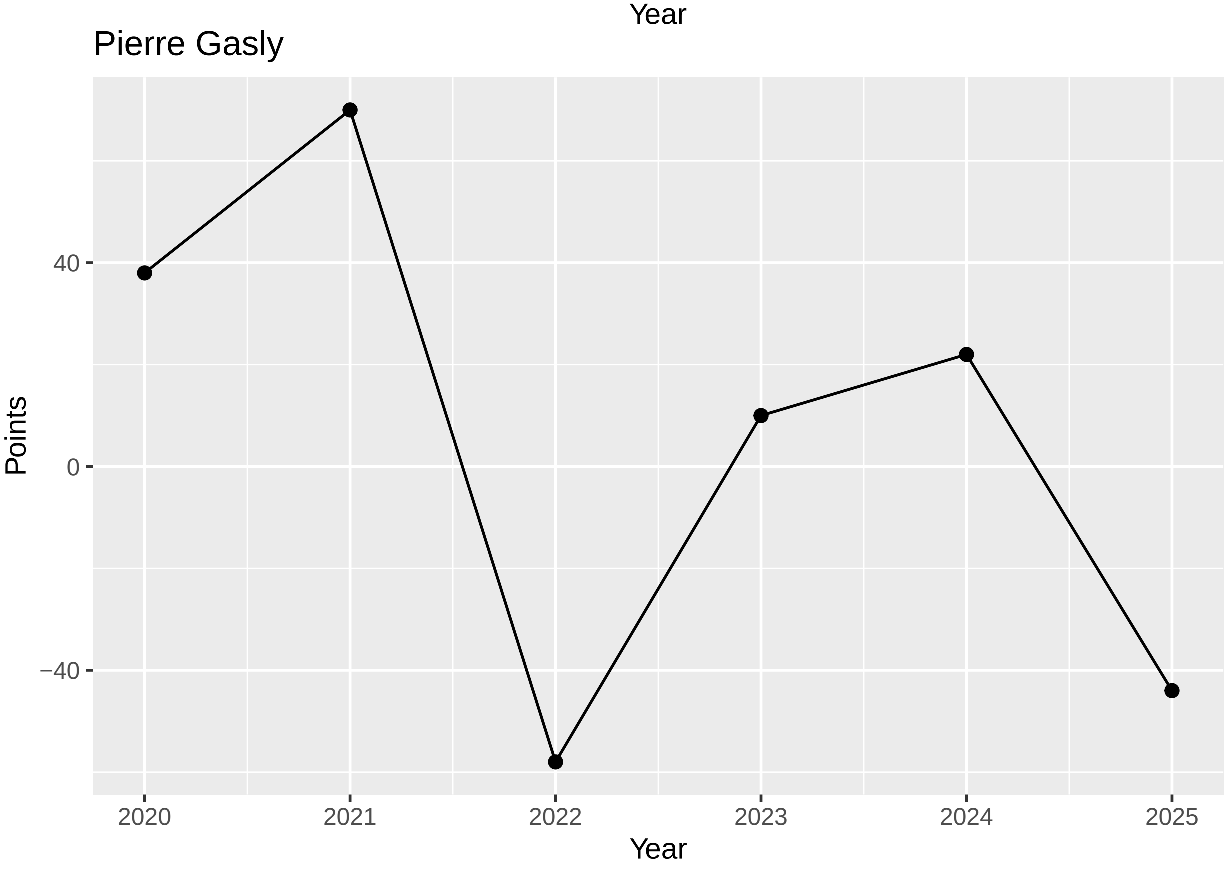


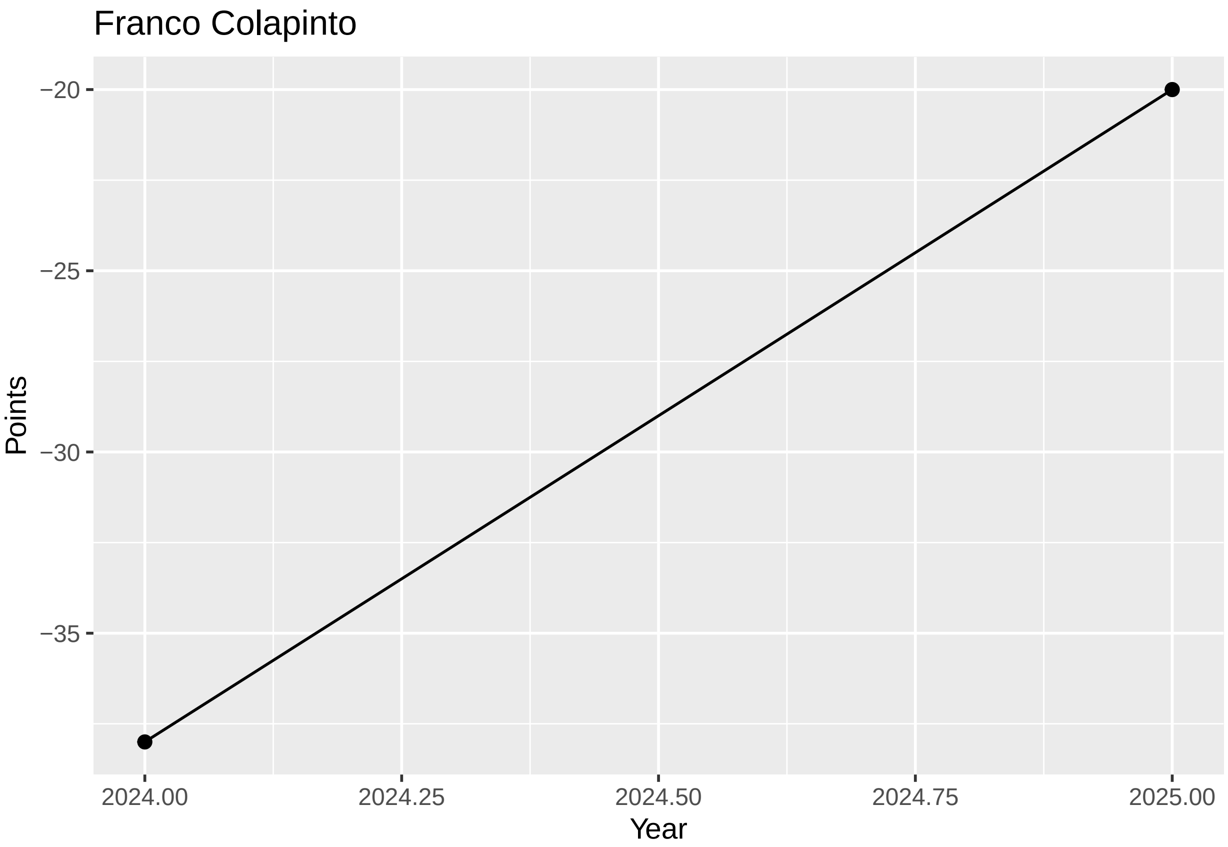
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```


Isack Hadjar
Points
5.050
5.025
5.000
4.975
4.950
2024.950
2024.975
2025.000
2025.025
2025.0
Year


Liam Lawson
Points
−20
−30
−40
−50
2023.0
2023.5
2024.0
2024.5
2025.0
Year


Fernando Alonso
Points
200
100
0
2021
2022
2023
2024
2025
Year


Pierre Gasly
Points
40
0
−40
2020
2021
2022
2023
2024
2025
Year



```
final_driver_summary <- union_data %>%
  select(`Driver Code`, Driver, Price, points_acc) %>%
  mutate(point_per_cost = points_acc / Price) %>%
  arrange(desc(point_per_cost))

model_1 <- lm(points_acc ~ Price, data = final_driver_summary)
summary(model_1)
```

```
##
## Call:
## lm(formula = points_acc ~ Price, data = final_driver_summary)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -726.72 -312.47  -34.59  132.31 1137.49
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -521.16     213.19  -2.445    0.025 *
## Price          78.85      13.38   5.894 0.000014 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 520.6 on 18 degrees of freedom
## Multiple R-squared:  0.6587, Adjusted R-squared:  0.6397
## F-statistic: 34.74 on 1 and 18 DF,  p-value: 0.00001401
```

```r
racepoints$event_id <- paste(racepoints$year,racepoints$event_name)
racepoints_events = racepoints %>%
  group_by(event_id, driver_code) %>%
  summarise(point_event = sum(points_real), .groups = "drop")
countPoints_events <- racepoints_events %>%
  group_by(driver_code, event_id)%>%
  summarise(
    `Total Points per Event` = sum(point_event, na.rm = TRUE),
    .groups = "drop") %>%
  arrange(desc(`Total Points per Event`), event_id)
head(countPoints_events)
```

```
## # A tibble: 6 x 3
##   driver_code event_id                  `Total Points per Event`
##   <chr>       <chr>                                        <dbl>
## 1 VER         2024 São Paulo Grand Prix                       41
## 2 VER         2022 Belgian Grand Prix                         38
## 3 VER         2021 Russian Grand Prix                         36
## 4 GAS         2020 Italian Grand Prix                         34
## 5 HAM         2021 São Paulo Grand Prix                       34
## 6 VER         2022 Hungarian Grand Prix                       34
```

```r
QualificationResult$event_id <- paste(QualificationResult$year, QualificationResult$event_name)
qualifypoints_events <- QualificationResult %>%
  group_by(driver_code, event_id) %>%
  summarise(total_points_qualify = sum(points, na.rm = TRUE))
```

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

```r
union_data_events <- left_join(countPoints_events, qualifypoints_events,
                        by = c("driver_code", "event_id"))
union_data_events <- union_data_events %>%
  mutate(subtotal = `Total Points per Event` + total_points_qualify) %>%
  select(driver_code, event_id, subtotal) %>%
  arrange(desc(subtotal))
head(union_data_events)
```

```
## # A tibble: 6 x 3
##   driver_code event_id                  subtotal
##   <chr>       <chr>                        <dbl>
## 1 VER         2022 Belgian Grand Prix         48
## 2 HAM         2021 São Paulo Grand Prix       44
## 3 VER         2024 São Paulo Grand Prix       41
## 4 BOT         2021 Italian Grand Prix         41
## 5 VER         2022 Italian Grand Prix         40
## 6 VER         2023 Hungarian Grand Prix       40
```

```r
LapTimePerDriver_events <- laps_modified %>%
  filter(
    !is.na(LapTimeInSeconds) & LapTimeInSeconds > 0,
    IsAccurate == TRUE,
    !Deleted
  ) %>%
  group_by(event_id) %>%
  filter(LapTimeInSeconds == min(LapTimeInSeconds, na.rm = TRUE)) %>%
  ungroup() %>%
  distinct(event_id, .keep_all = TRUE) %>%
  select(driver_code = Driver,
         fastest_lap_time = LapTimeInSeconds,
         event_id,
         year)

LapTimePerDriver_events <- LapTimePerDriver_events %>% mutate(
  points_add = 10)

fastest_lap_points_events <- LapTimePerDriver_events %>%
  group_by(driver_code, event_id) %>%
  summarise(point_fastest = sum(points_add))
```

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

```r
union_data_events <- left_join(union_data_events, fastest_lap_points_events,
                               by = c("driver_code", "event_id"))

union_data_events <- union_data_events %>%
  mutate(point_fastest_conv = case_when(
    !is.na(point_fastest) ~ point_fastest,
    TRUE ~ 0))

union_data_events <- union_data_events %>%
  mutate(points_events = subtotal + point_fastest_conv) %>%
  select(driver_code, event_id, points_events) %>%
  arrange(desc(points_events))
head(union_data_events)
```

```
## # A tibble: 6 x 3
##   driver_code event_id                  points_events
##   <chr>       <chr>                             <dbl>
## 1 VER         2022 Belgian Grand Prix              58
## 2 VER         2024 São Paulo Grand Prix            51
## 3 VER         2023 Hungarian Grand Prix            50
## 4 VER         2023 Miami Grand Prix                45
## 5 VER         2023 Qatar Grand Prix                45
## 6 BOT         2020 Russian Grand Prix              45
```

```r
points_events <- left_join(DriverPrice, union_data_events, by = c("Driver Code" = "driver_code"))
head(points_events)
```

```
## # A tibble: 6 x 5
##   `Driver Code` Driver       Price event_id                   points_events
##   <chr>         <chr>        <dbl> <chr>                              <dbl>
## 1 NOR           Lando Norris  30.5 2024 Dutch Grand Prix                 45
## 2 NOR           Lando Norris  30.5 2025 Australian Grand Prix            45
## 3 NOR           Lando Norris  30.5 2025 Monaco Grand Prix                45
## 4 NOR           Lando Norris  30.5 2025 Italian Grand Prix               37
## 5 NOR           Lando Norris  30.5 2025 Miami Grand Prix                 37
## 6 NOR           Lando Norris  30.5 2024 Spanish Grand Prix               37
```

```r
points_by_events <- points_events %>% arrange(event_id) %>% pivot_wider(
  names_from = event_id,
  values_from = points_events,
  id_cols = c(`Driver Code`, Driver),
  values_fill = 0
)
```

```r
racepoints_2 <- racepoints %>%
  select(driver_code,year, event_id, event_name, points_real)
head(racepoints_2)
```

```
## # A tibble: 6 x 5
##   driver_code  year event_id                 event_name          points_real
##   <chr>       <dbl> <chr>                    <chr>                     <dbl>
## 1 BOT          2020 2020 Austrian Grand Prix Austrian Grand Prix          25
## 2 LEC          2020 2020 Austrian Grand Prix Austrian Grand Prix          23
## 3 NOR          2020 2020 Austrian Grand Prix Austrian Grand Prix          15
## 4 HAM          2020 2020 Austrian Grand Prix Austrian Grand Prix          13
## 5 SAI          2020 2020 Austrian Grand Prix Austrian Grand Prix          13
## 6 PER          2020 2020 Austrian Grand Prix Austrian Grand Prix           8
```

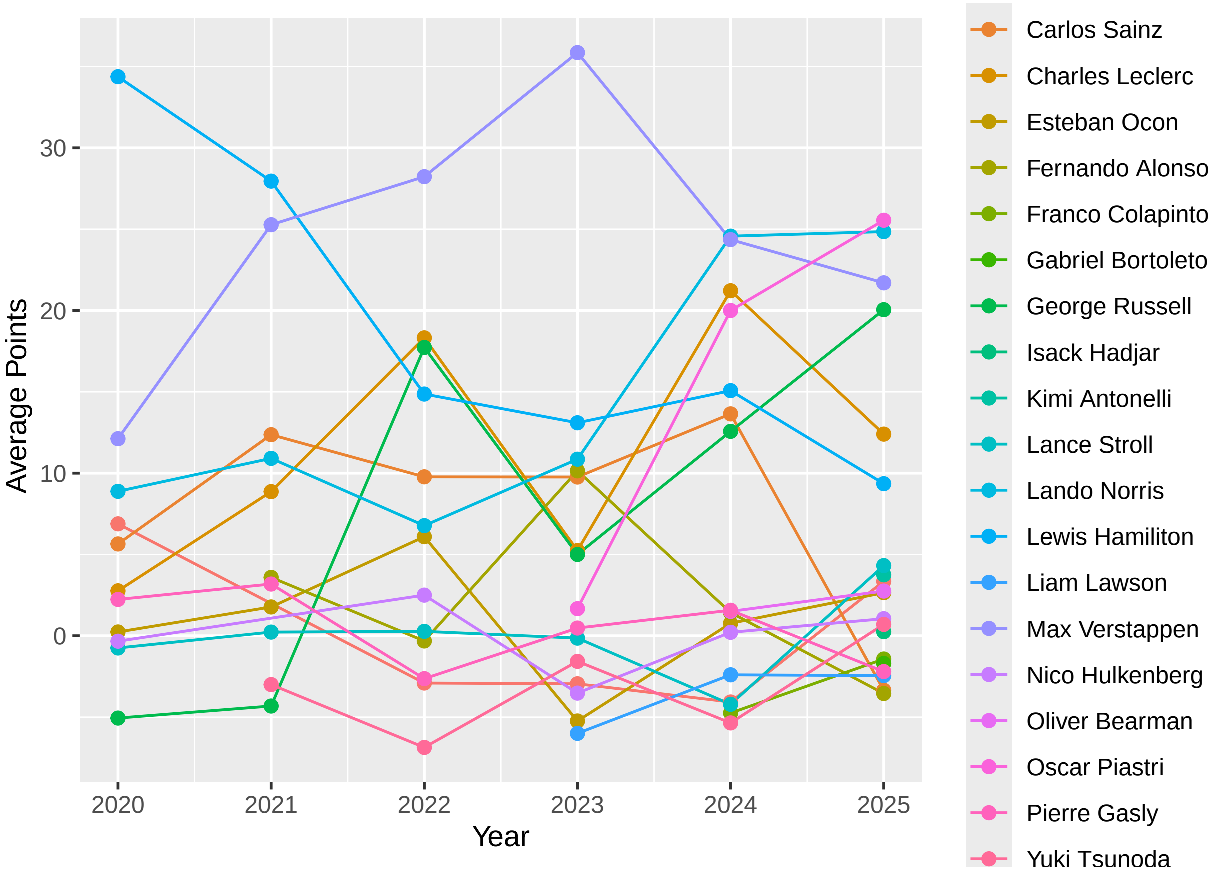
```r
qualifypoints_2 <- QualificationResult %>%
  group_by(driver_code, event_id) %>%
  summarise(points_sum = sum(points, na.rm = TRUE), .groups = "drop")

fastest_lap_2 <- laps_modified %>%
  filter(
    !is.na(LapTimeInSeconds) & LapTimeInSeconds > 0,
    IsAccurate == TRUE,
    !Deleted
  ) %>%
  group_by(event_id) %>%
  filter(LapTimeInSeconds == min(LapTimeInSeconds, na.rm = TRUE)) %>%
  ungroup() %>%
  distinct(event_id, .keep_all = TRUE) %>%
  select(driver_code = Driver, event_id) %>%
  mutate(points_lap = 10)

summary_union_data <- racepoints_2 %>%
  left_join(qualifypoints_2, by = c("driver_code", "event_id")) %>%
  left_join(fastest_lap_2, by = c("driver_code", "event_id")) %>%
  mutate(points_sum = replace_na(points_sum, 0),
         points_lap = replace_na(points_lap, 0),
         grand_total = points_real + points_sum + points_lap) %>%
  inner_join(DriverPrice, by=c("driver_code" = "Driver Code")) %>%
  select(Year = year, Event = event_name, Driver, grand_total) %>%
  arrange(Year, Event, Driver)

avg_union_data <- summary_union_data %>%
  group_by(Driver, Year) %>%
  summarise(number_events = n(),
            total_points_per_year = sum(grand_total),
            avg_points = total_points_per_year/number_events,
            .groups = "drop") %>%
  arrange(Driver, Year)
head(avg_union_data)
```

```
## # A tibble: 6 x 5
##   Driver            Year number_events total_points_per_year avg_points
##   <chr>            <dbl>         <int>                 <dbl>      <dbl>
## 1 Alexander Albon  2020            17                   117       6.88
## 2 Alexander Albon  2022            21                   -61      -2.90
## 3 Alexander Albon  2023            21                   -62      -2.95
## 4 Alexander Albon  2024            14                   -57      -4.07
## 5 Alexander Albon  2025            20                    67       3.35
## 6 Carlos Sainz     2020            17                    96       5.65
```

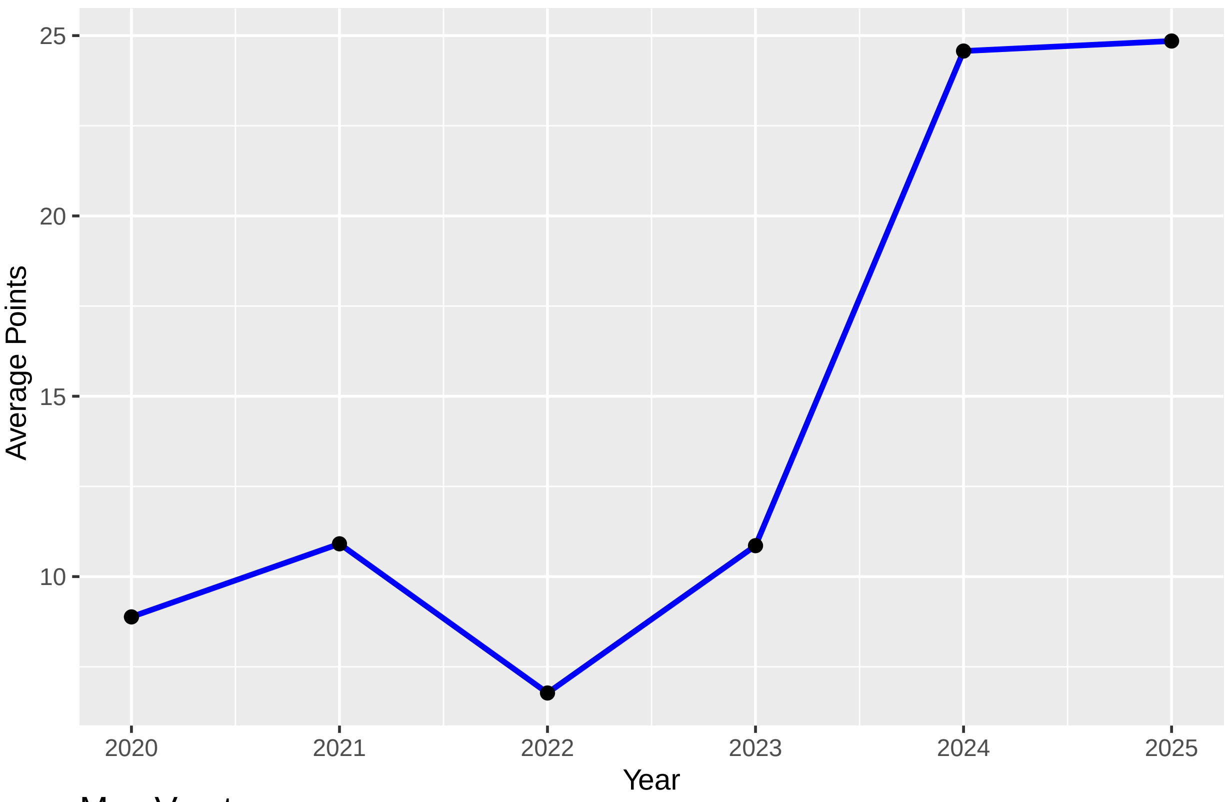
```
points_by_year_2 <- avg_union_data %>% arrange(Year) %>% pivot_wider(
  names_from = Year,
  values_from = avg_points,
  id_cols = Driver,
  values_fill = 0
) %>%
  arrange(desc(`2025`))
kable(points_by_year_2)
```

Driver	2020	2021	2022	2023	2024	2025
Oscar Piastri	0.0000000	0.0000000	0.0000000	1.6666667	20.0000000	25.550000
Lando Norris	8.8823529	10.9090909	6.7727273	10.8571429	24.5714286	24.850000
Max Verstappen	12.1176471	25.2727273	28.2272727	35.8571429	24.3571429	21.700000
George Russell	-5.0588235	-4.3181818	17.7272727	5.0000000	12.5714286	20.050000
Charles Leclerc	2.7647059	8.8636364	18.3181818	5.2380952	21.2142857	12.400000
Lewis Hamilton	34.3750000	27.9545455	14.8636364	13.0952381	15.0714286	9.350000
Lance Stroll	-0.7500000	0.2272727	0.2727273	-0.1428571	-4.2142857	4.315790
Kimi Antonelli	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	3.750000
Alexander Albon	6.8823529	0.0000000	-2.9047619	-2.9523810	-4.0714286	3.350000
Oliver Bearman	0.0000000	0.0000000	0.0000000	0.0000000	1.5000000	2.750000
Esteban Ocon	0.2352941	1.7727273	6.0909091	-5.2380952	0.7692308	2.650000
Nico Hulkenberg	-0.3333333	0.0000000	2.5000000	-3.5238095	0.2142857	1.050000
Yuki Tsunoda	0.0000000	-3.0000000	-6.8636364	-1.5714286	-5.3571429	0.700000
Isack Hadjar	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.250000
Franco Colapinto	0.0000000	0.0000000	0.0000000	0.0000000	-4.7500000	-1.428571
Gabriel Bortoleto	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	-1.700000
Pierre Gasly	2.2352941	3.1818182	-2.6363636	0.4761905	1.5714286	-2.200000
Liam Lawson	0.0000000	0.0000000	0.0000000	-6.0000000	-2.4000000	-2.450000
Carlos Sainz	5.6470588	12.3636364	9.7727273	9.7619048	13.6428571	-3.350000
Fernando Alonso	0.0000000	3.5909091	-0.3181818	10.1428571	1.4285714	-3.550000

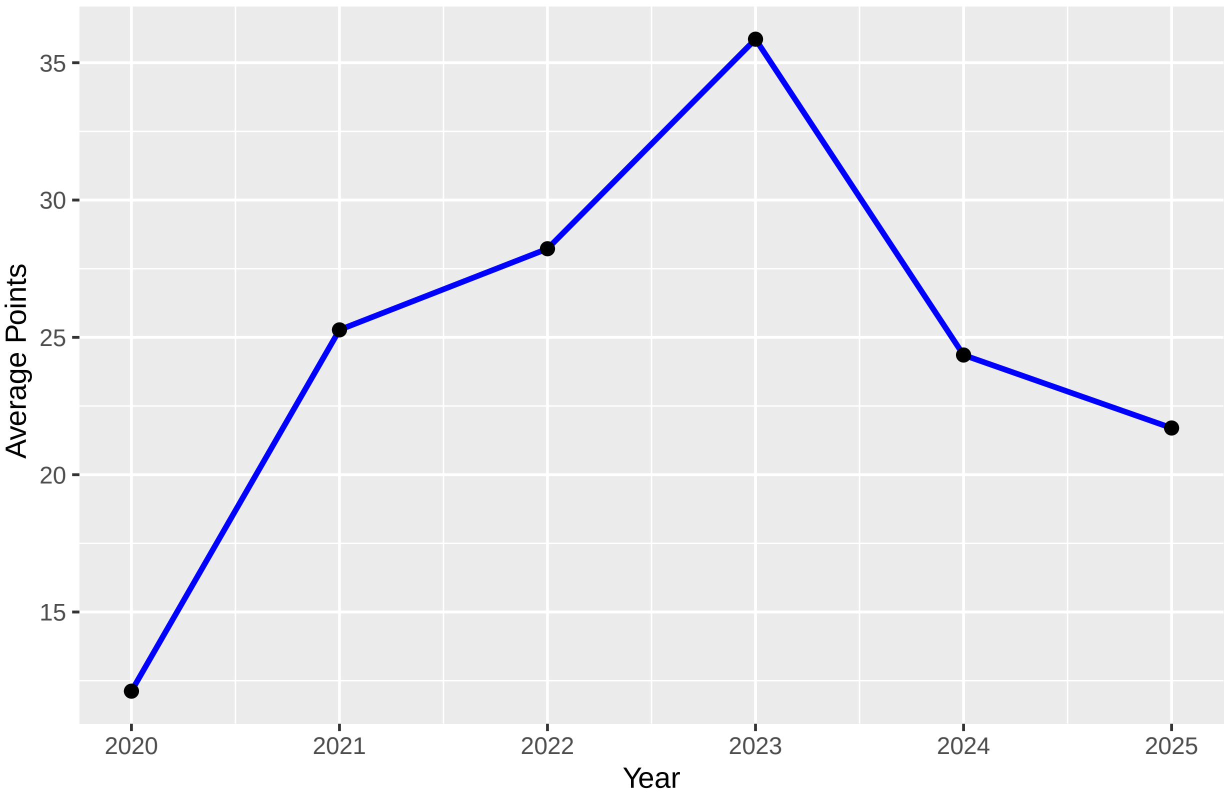
```
ggplot(avg_union_data, aes(x = Year, y = avg_points, col = Driver)) +
  geom_line() +
  geom_point(size = 2) +
  labs(x = "Year", y = " Average Points", col = "Drivers")
```



```
for (current_driver in all_drivers){
  driver_data_2 <- avg_union_data %>%
    filter(Driver == current_driver)
  driver_plot_2 <- ggplot(driver_data_2, aes(x = Year, y = avg_points)) +
    geom_line(aes(group = 1), color = "blue", linewidth = 1) +
    geom_point(size = 2) +
    scale_x_continuous(breaks = unique(driver_data_2$Year)) +
    labs(x = "Year", y = " Average Points",
         title = current_driver)
  print(driver_plot_2)
}
```

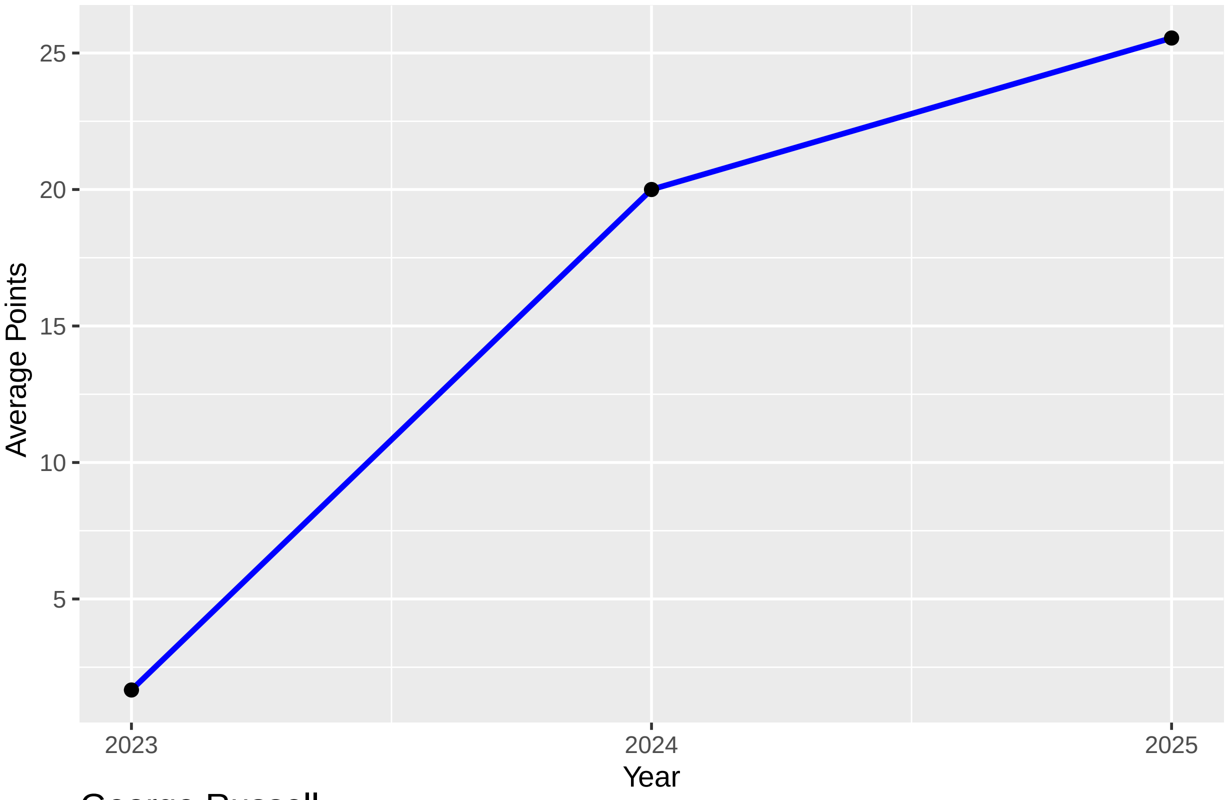


Lando Norris

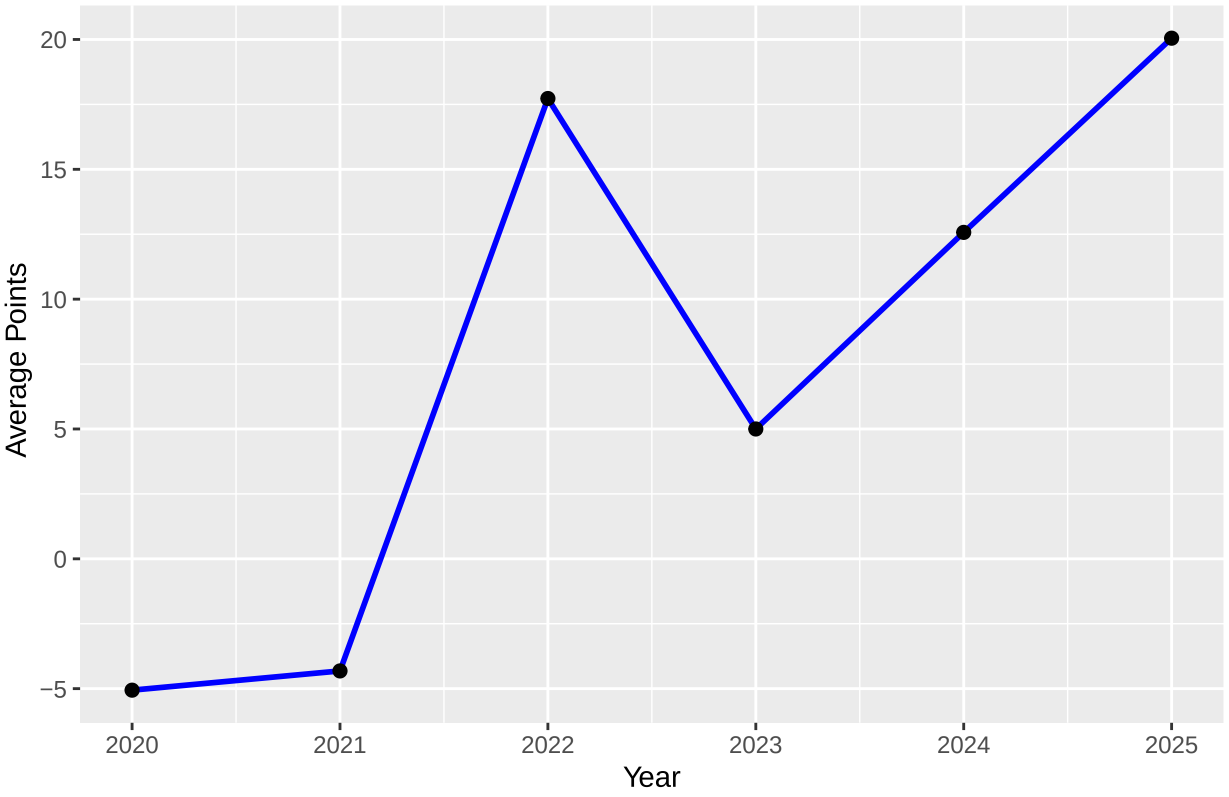


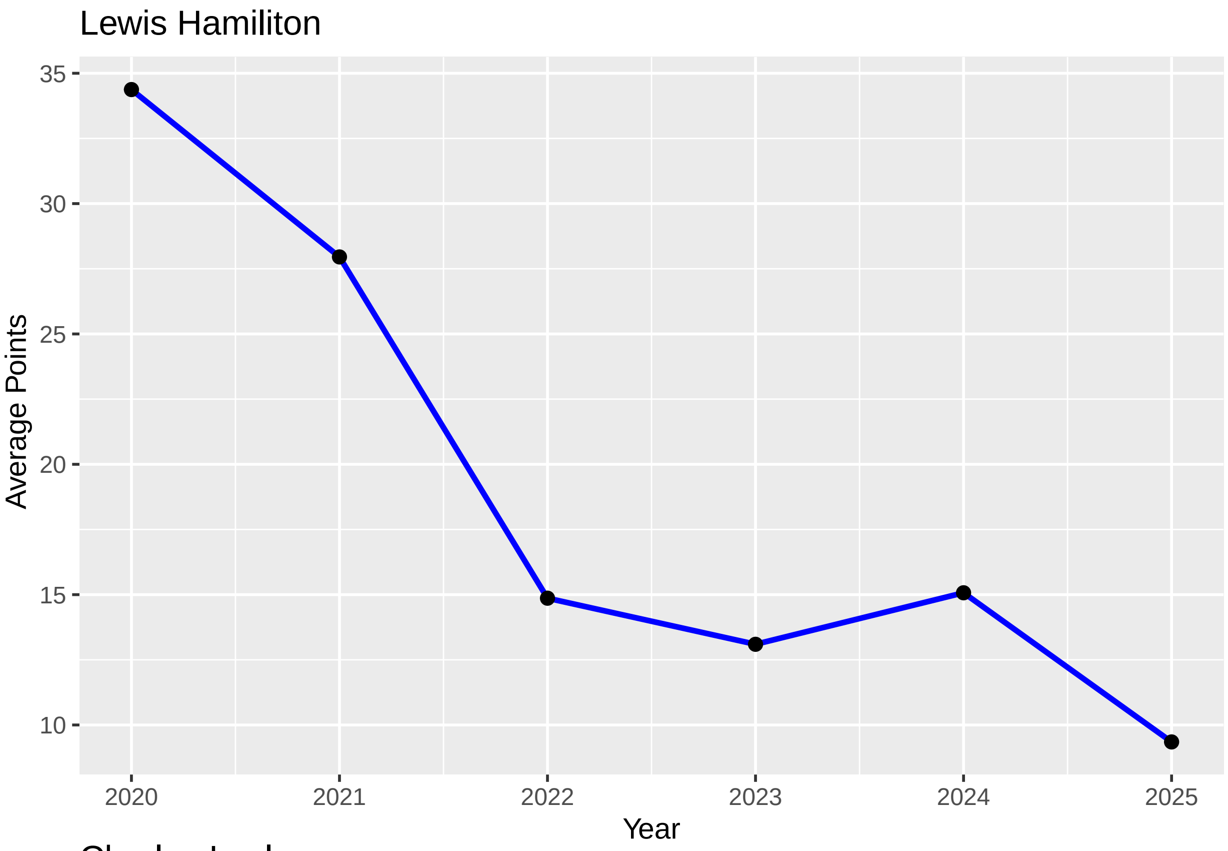
Max Verstappen

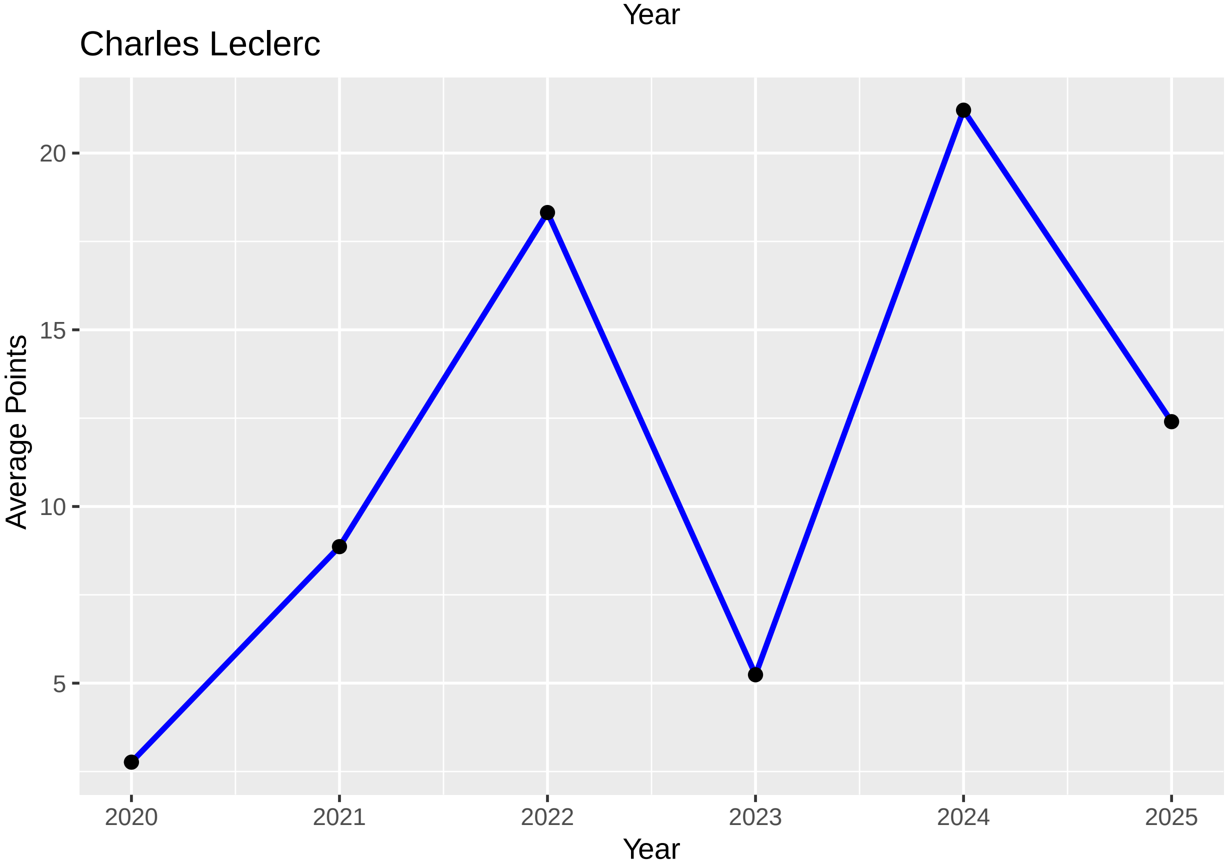
Oscar Piastri



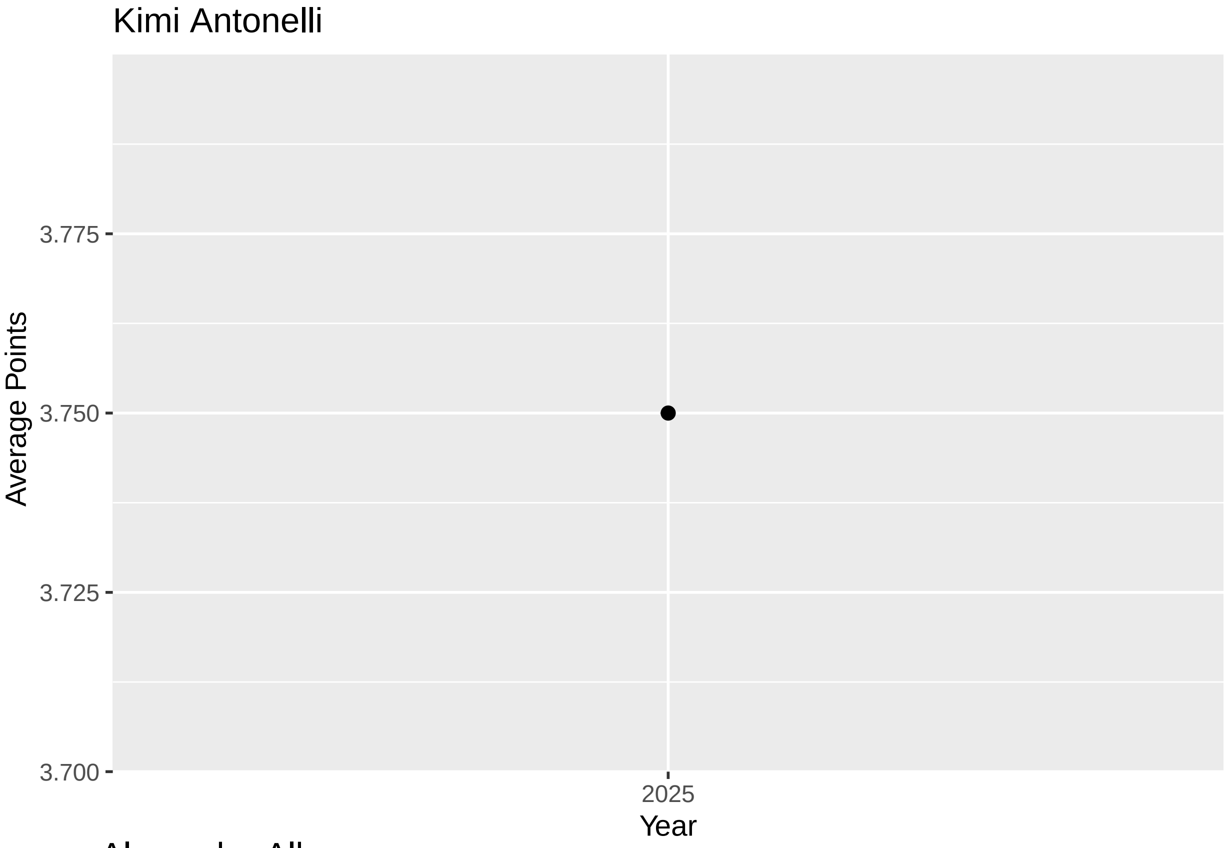
George Russell

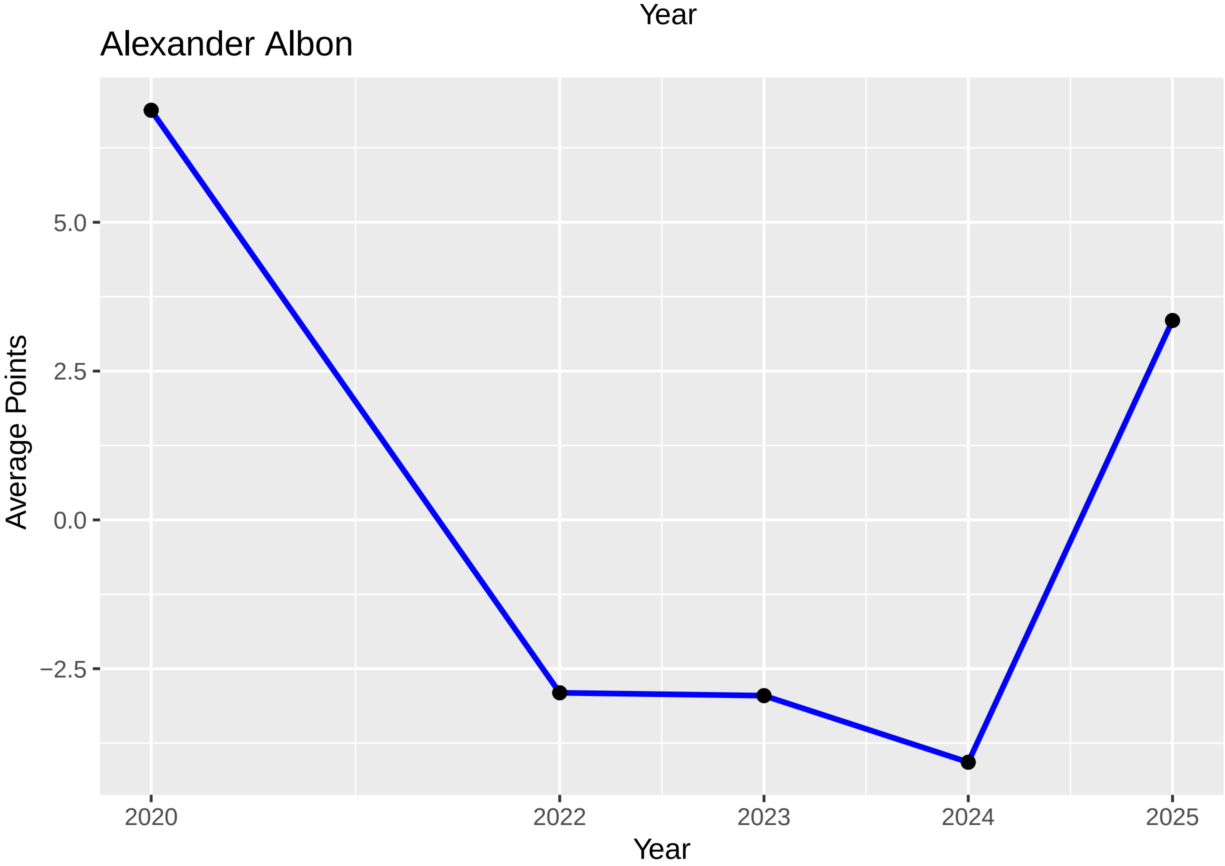


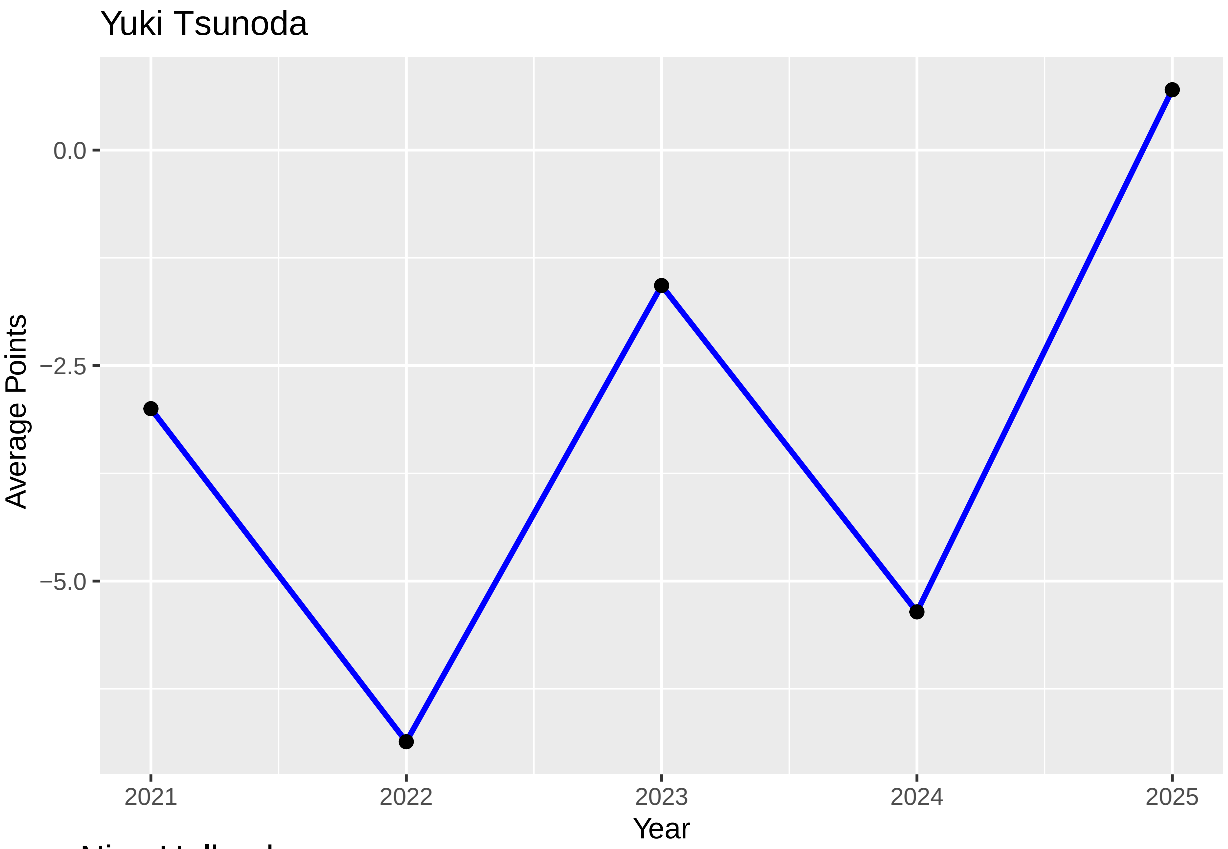




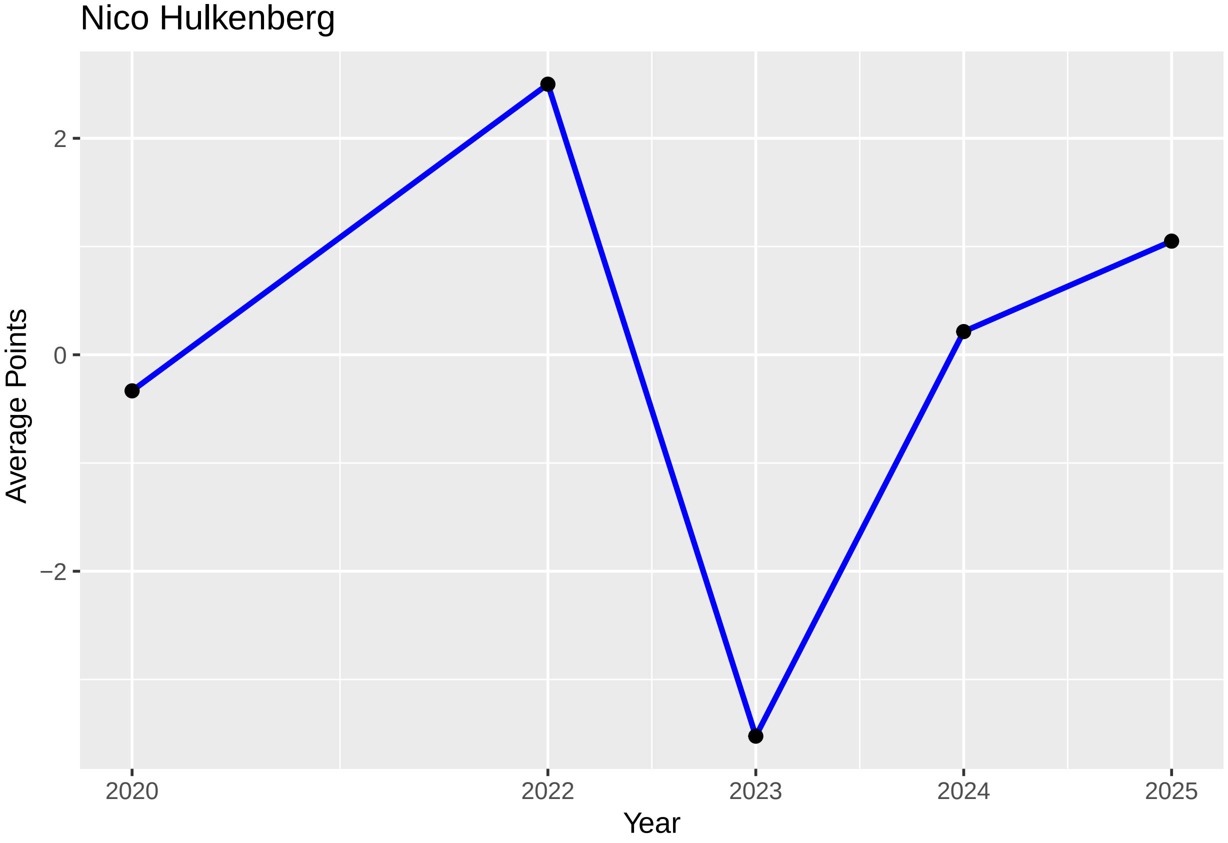
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



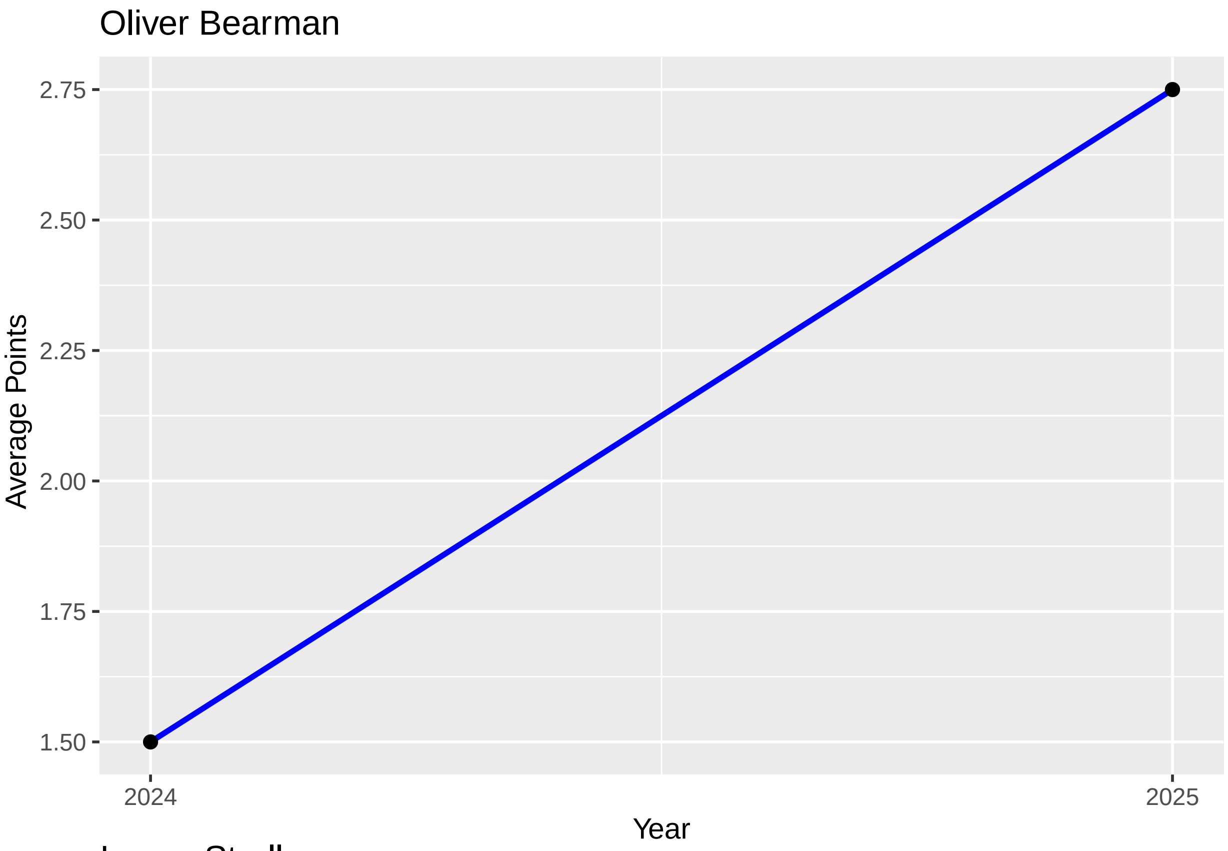


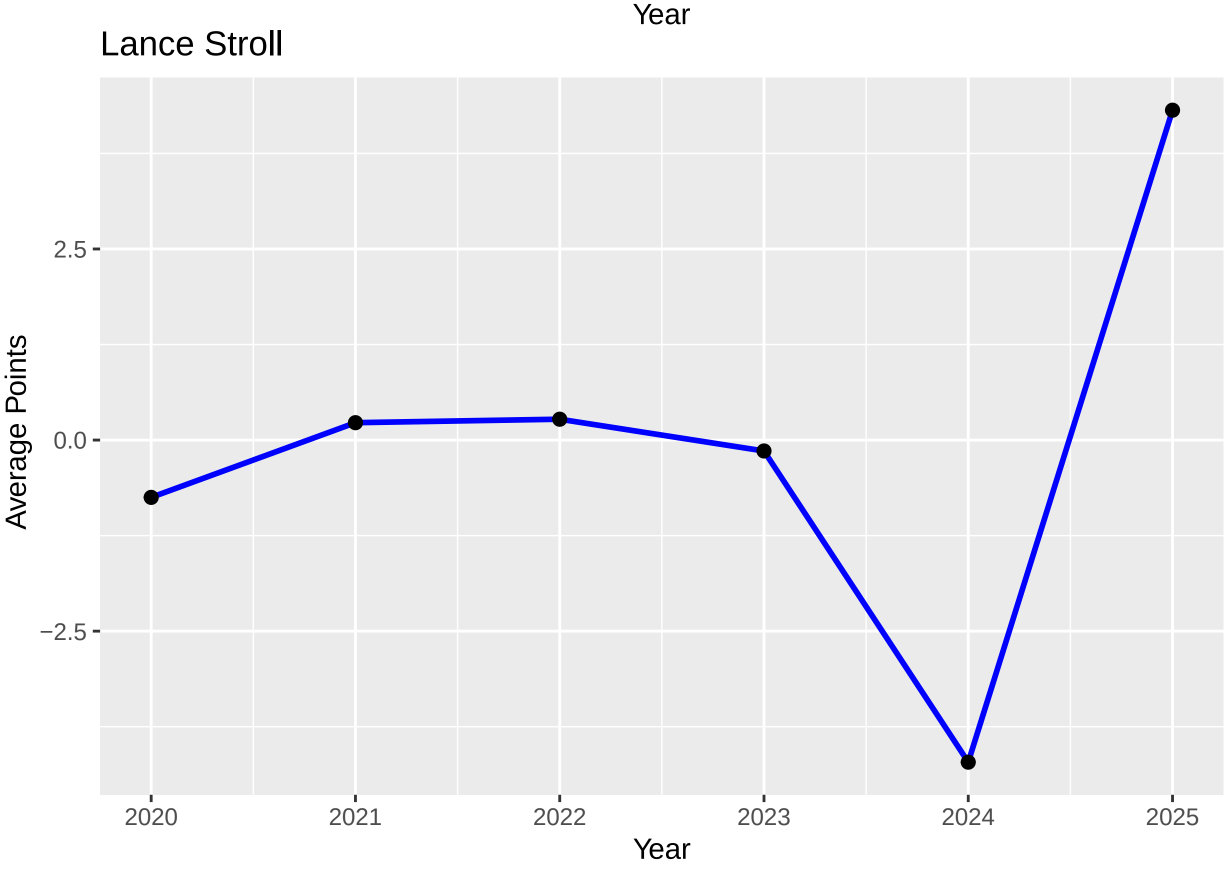


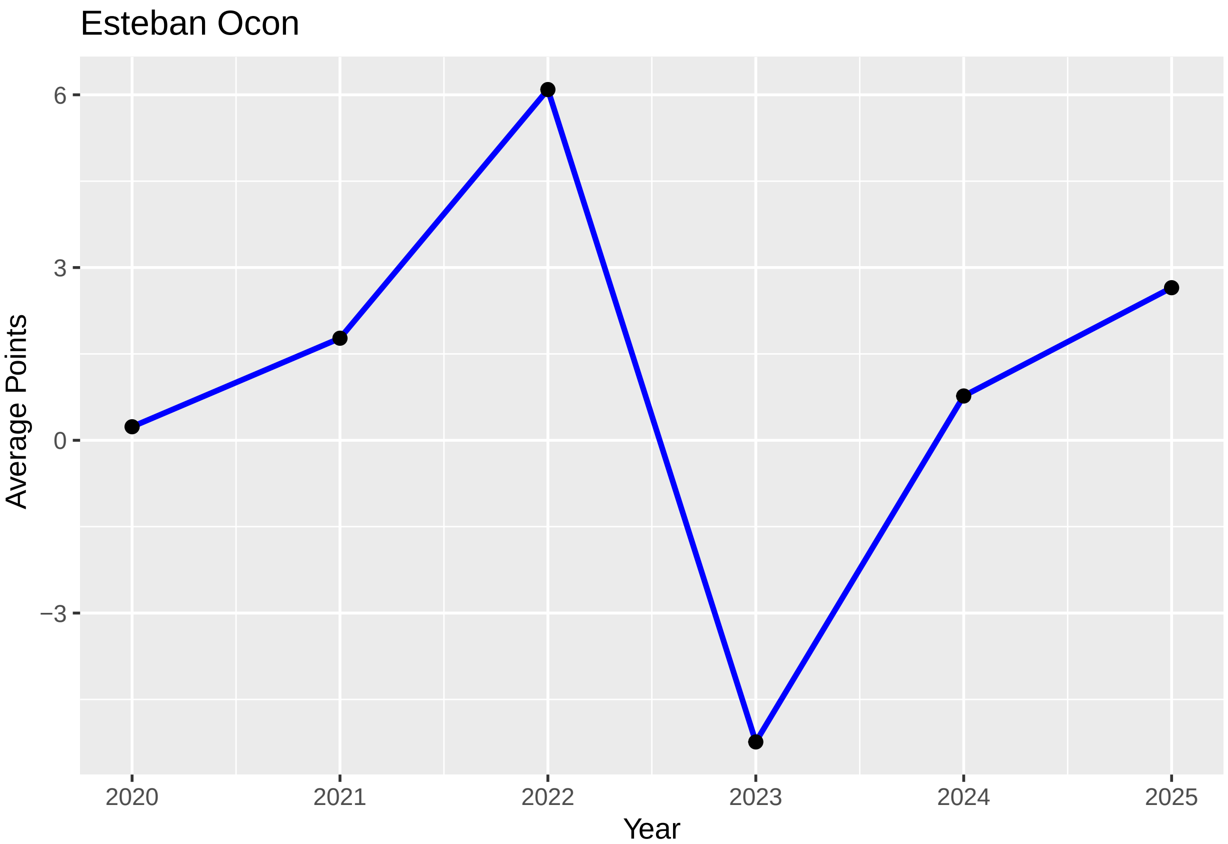
Yuki Tsunoda

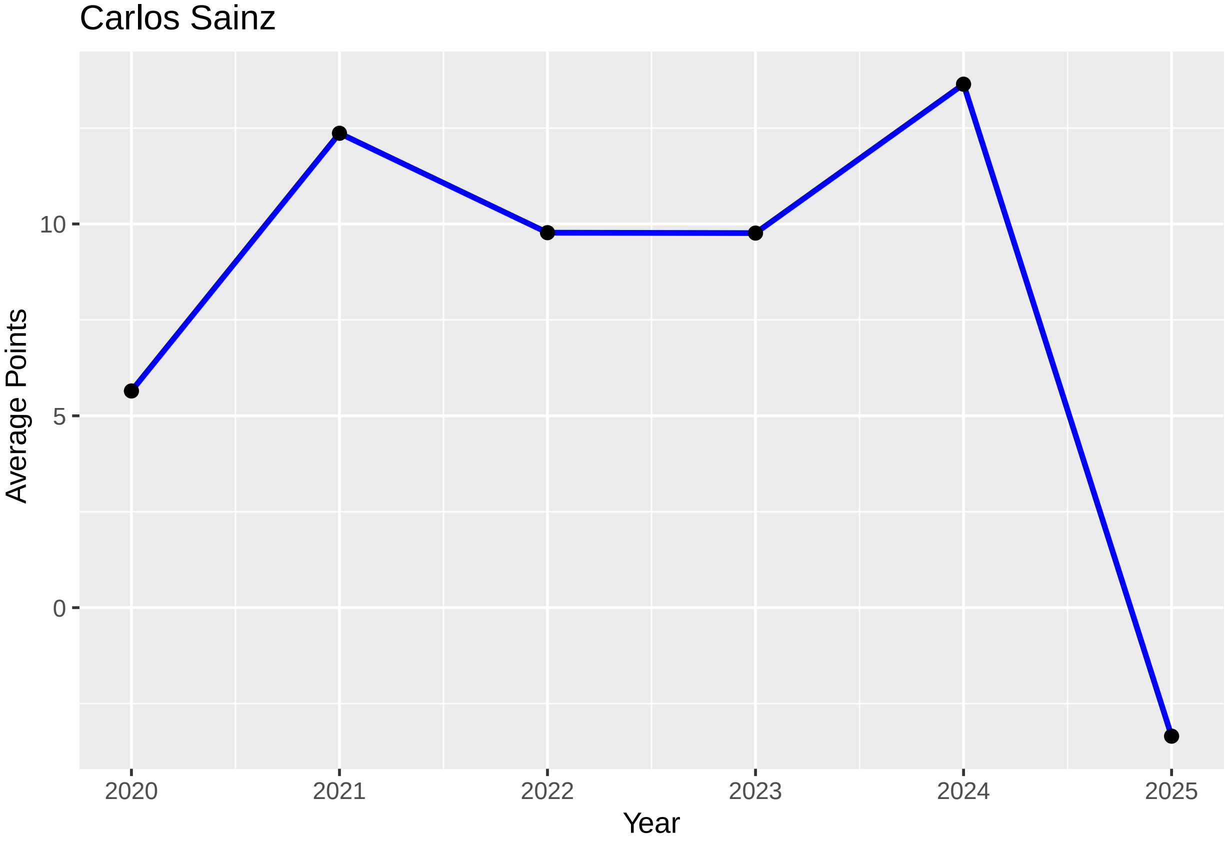


Nico Hulkenberg

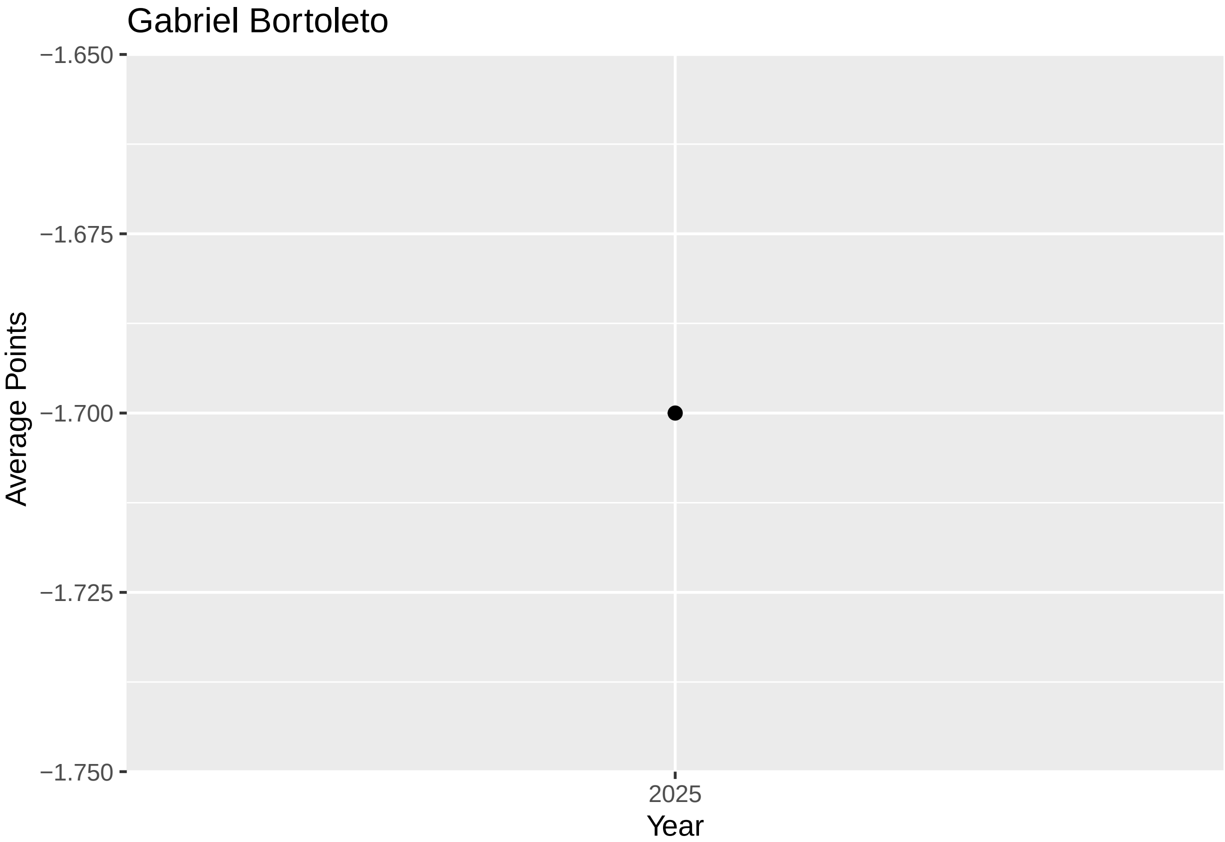

Oliver Bearman
Average Points
2.75
2.50
2.25
2.00
1.75
1.50
2024
2025
Year


Lance Stroll
Average Points
2.5
0.0
−2.5
2020
2021
2022
2023
2024
2025
Year

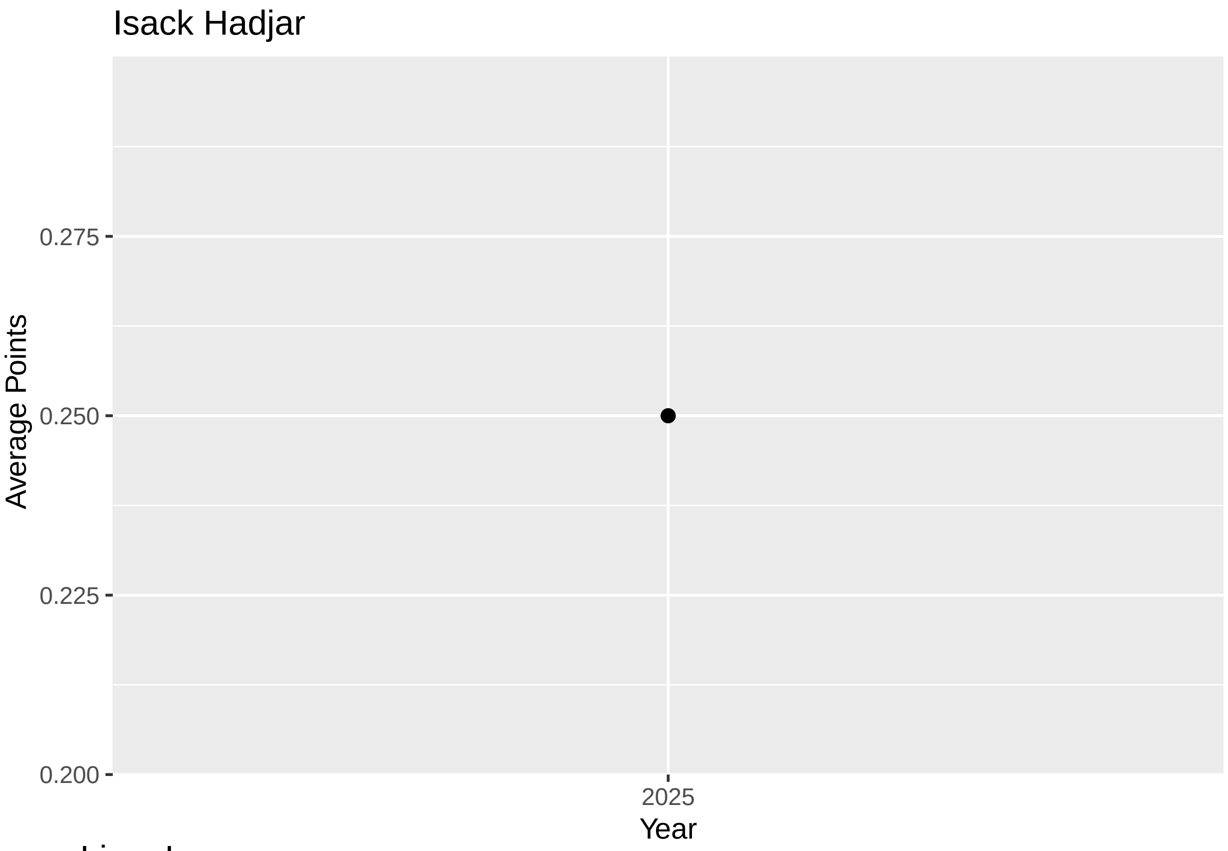


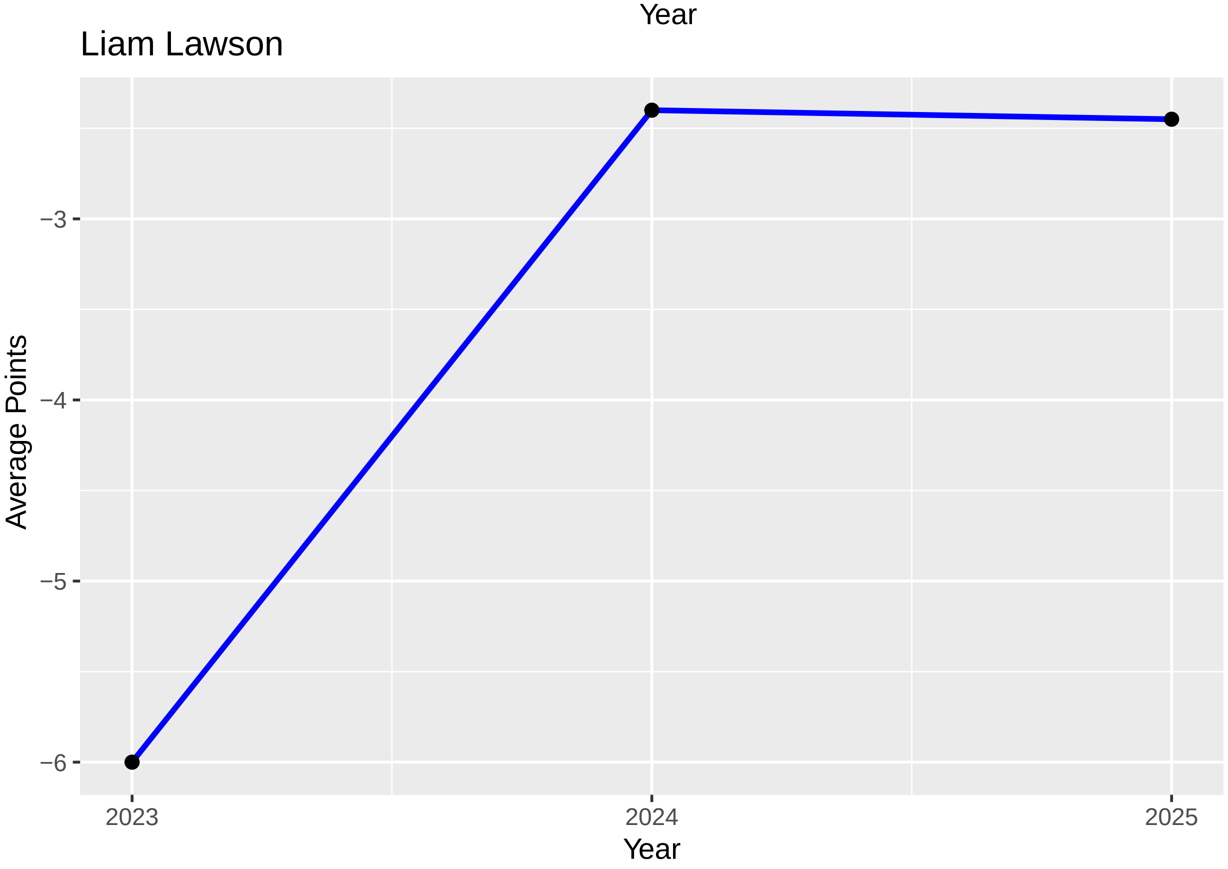


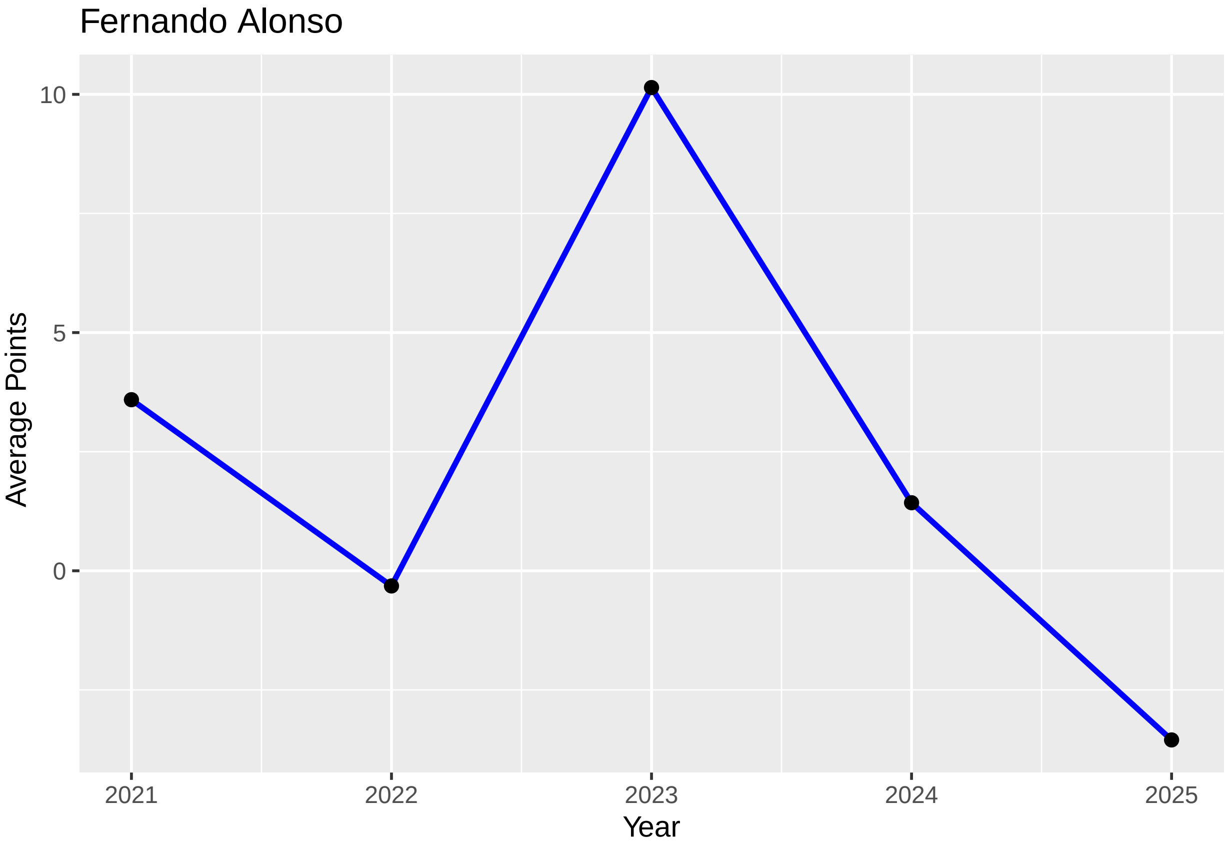
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

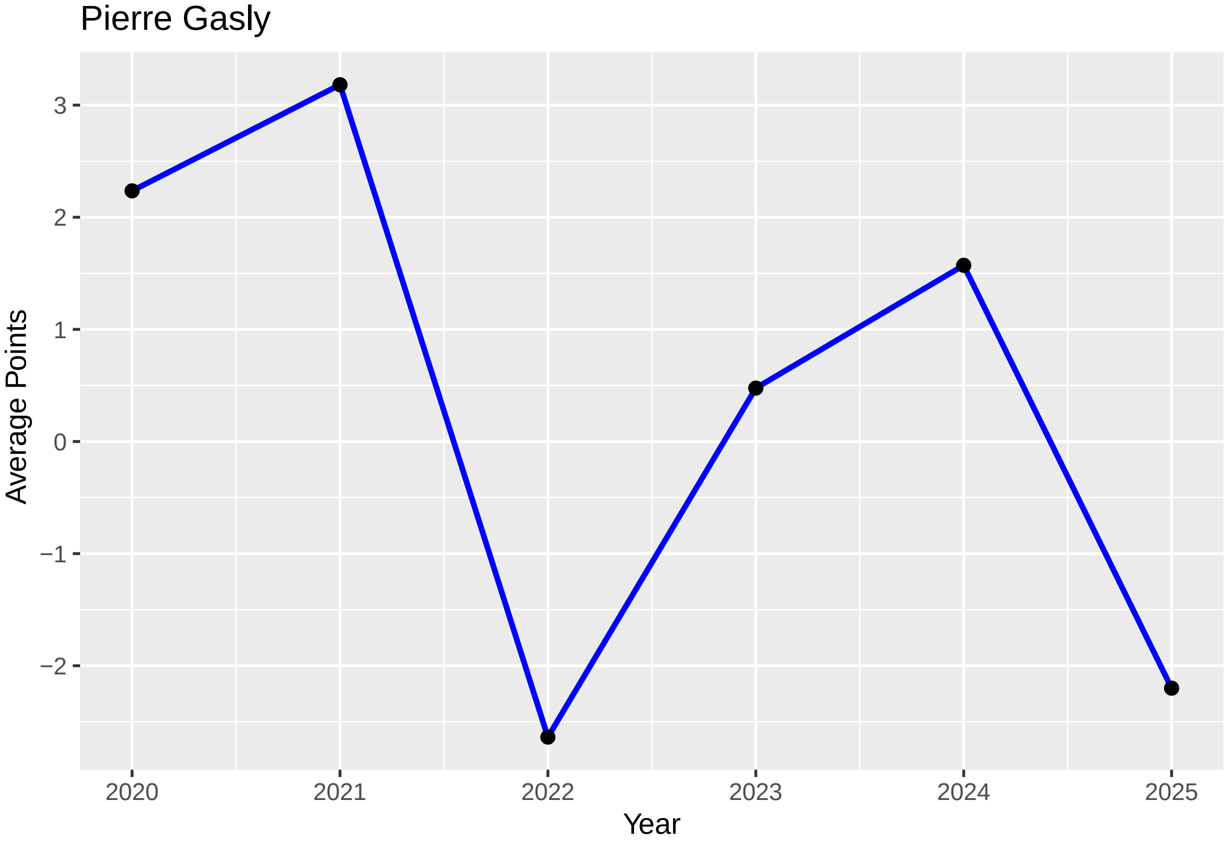


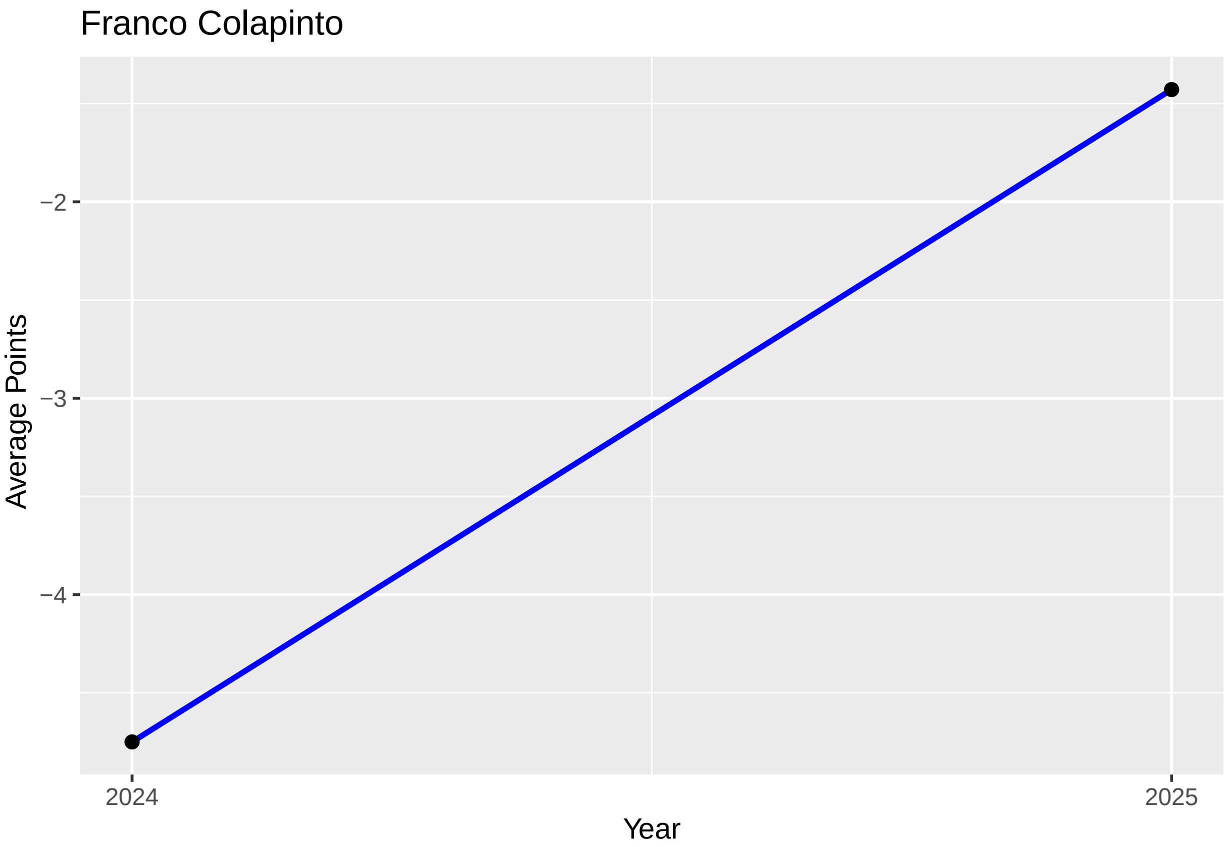
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```


Isack Hadjar
Average Points
0.275
0.250
0.225
0.200
2025
Year


Liam Lawson
Average Points
−3
−4
−5
−6
2023
2024
2025
Year


Fernando Alonso
Average Points
10
5
0
2021
2022
2023
2024
2025
Year


Pierre Gasly
Average Points
3
2
1
0
−1
−2
2020
2021
2022
2023
2024
2025
Year



```r
# Download the dataset in .csv file
write_csv(final_driver_summary, "driver_summary.csv")
write_csv(avg_union_data, "driver_trends.csv")
```

```r
points_2025 <- points_by_year %>%
  select(`Driver Code`,Driver, `2025`) %>%
  rename(pts = `2025`)
points_2025 <- points_2025 %>%
  left_join(DriverPrice, by = "Driver Code") %>%
  mutate(rate = pts/Price) %>%
  select(-Driver.y) %>%
  rename(Driver = Driver.x)
```

```r
# --- COMPREHENSIVE SCENARIO ANALYSIS (Sizes 2 to 10) WITH NAMES ---

# 1. Setup: Filter for positive drivers only
positive_drivers <- points_2025 %>% filter(pts > 0)
driver_codes <- positive_drivers$`Driver Code`

# Create vectors for fast math
prices <- setNames(positive_drivers$Price, driver_codes)
points <- setNames(positive_drivers$pts, driver_codes)

# --- NEW: Create a Name Lookup Map ---
names_map <- setNames(positive_drivers$Driver, driver_codes)

# 2. Create a list to store ALL valid teams
all_valid_teams_list <- list()
```

```r
# 3. Loop from Size 2 to Size 10
for (k in 2:10) {

  # A. Generate Combinations (Matrix where each col is a team)
  combos <- combn(driver_codes, k)

  # B. Calculate Cost & Points (Vectorized Speed)
  team_costs <- colSums(matrix(prices[combos], nrow = k))
  team_points <- colSums(matrix(points[combos], nrow = k))

  # C. Create Table
  temp_df <- data.frame(
    Team_Size = k,
    Total_Cost = team_costs,
    Total_Points = team_points,
    Rate_Points_Cost = team_points/team_costs,

    # --- CHANGED: Convert Codes to Names before pasting ---
    Drivers = apply(combos, 2, function(col) {
      # Look up the full names
      full_names <- names_map[col]
      # Paste them together with a comma
      paste(full_names, collapse = ", ")
    })
  )

  # D. Filter for Budget
  # We keep ALL teams that fit the budget
  valid_teams <- temp_df %>%
    filter(Total_Cost <= 100) %>%
    arrange(desc(Total_Points))

  # E. Save the results if any valid teams exist
  if (nrow(valid_teams) > 0) {
    all_valid_teams_list[[k]] <- valid_teams
  } else {
    message(paste("No teams fit the budget at size:", k))
  }
}
```

```
## No teams fit the budget at size: 10
```

```r
# 4. Combine into one Big Dataframe
all_possible_combinations <- bind_rows(all_valid_teams_list)

# 5. Show the Best Team for EACH Size
best_per_size <- all_possible_combinations %>%
  group_by(Team_Size) %>%
  slice_max(Total_Points, n = 1) %>%
  arrange(Team_Size)

kable(best_per_size, caption = "Best Feasible Team by Size (2-10 Drivers)")
```

Table 6: Best Feasible Team by Size (2-10 Drivers)

Team_Size	Total_Cost	Total_Points	Rate_Points_Cost	Drivers
2	56.8	1008	17.746479	Oscar Piastri, Lando Norris
3	85.9	1442	16.786962	Oscar Piastri, Lando Norris, Max Verstappen
4	94.6	1524	16.109937	Oscar Piastri, Lando Norris, Max Verstappen, Lance Stroll
5	96.5	1546	16.020725	Oscar Piastri, Lando Norris, George Russell, Lance Stroll, Oliver Bearman
6	100.0	1459	14.590000	Oscar Piastri, Max Verstappen, George Russell, Oliver Bearman, Esteban Ocon, Isack Hadjar
7	99.8	1270	12.725451	Oscar Piastri, Lando Norris, Lance Stroll, Alexander Albon, Oliver Bearman, Esteban Ocon, Isack Hadjar
8	100.0	1195	11.950000	Oscar Piastri, George Russell, Lance Stroll, Alexander Albon, Oliver Bearman, Esteban Ocon, Nico Hulkenberg, Isack Hadjar
9	99.6	773	7.761044	George Russell, Lance Stroll, Kimi Antonelli, Alexander Albon, Oliver Bearman, Esteban Ocon, Nico Hulkenberg, Yuki Tsunoda, Isack Hadjar

```r
# 6. (Optional) View the Top 10 Best Teams Overall (Any Size)
kable(head(all_possible_combinations %>% arrange(desc(Total_Points)), 10),
      caption = "Top 10 Best Teams Overall (Any Size)")
```

Table 7: Top 10 Best Teams Overall (Any Size)

Team_Size	Total_Cost	Total_Points	Rate_Points_Cost	Drivers
5	96.5	1546	16.02073	Oscar Piastri, Lando Norris, George Russell, Lance Stroll, Oliver Bearman
5	95.9	1544	16.10010	Oscar Piastri, Lando Norris, George Russell, Lance Stroll, Esteban Ocon
4	94.6	1524	16.10994	Oscar Piastri, Lando Norris, Max Verstappen, Lance Stroll
5	94.9	1517	15.98525	Oscar Piastri, Lando Norris, George Russell, Oliver Bearman, Esteban Ocon
5	96.2	1512	15.71726	Oscar Piastri, Lando Norris, George Russell, Lance Stroll, Nico Hulkenberg
4	98.9	1509	15.25784	Oscar Piastri, Lando Norris, Max Verstappen, Alexander Albon
5	99.0	1505	15.20202	Oscar Piastri, Lando Norris, George Russell, Lance Stroll, Yuki Tsunoda
5	99.5	1500	15.07538	Oscar Piastri, Lando Norris, Max Verstappen, Esteban Ocon, Isack Hadjar
4	93.6	1497	15.99359	Oscar Piastri, Lando Norris, Max Verstappen, Oliver Bearman
5	95.3	1496	15.69780	Oscar Piastri, Lando Norris, George Russell, Lance Stroll, Isack Hadjar