Data Analytics 344

Tutorial Test 2
Department of Engineering
Stellenbosch University
2 August 2023

Instructions

Welcome to the first tutorial test! This tutorial and future tutorials will be auto-graded using test cases. To ensure your submission can be graded, please follow these guidelines:

1. Adding your student number: In the first code block, please store your student number in the variable student_number. For instance if your student number is 1234, the code block should contain the code student_number <- 1234
2. Package Usage: Only use the packages specified in the setup code block. If you need additional packages, install them using the R Console.
3. Hands-Off Code Blocks: Do not modify code blocks that start with the comment #DO NOT EDIT. These blocks are for loading packages or illustrating programming concepts and should remain unchanged.
4. Completing Code Blocks: Your main task is to complete the incomplete code provided. There's no need to add or modify any of the code blocks or questions; simply fill in the necessary answers.
5. Error-Free Submission: Before submitting your final answer, make sure your R Markdown document runs without errors by pressing the knit button. Only variables declared in this document will be accessible for grading.
6. Case Sensitivity: Remember that variables in R are case-sensitive. Use the exact variable names specified in the questions (e.g., Q1, not q1) and do not overwrite your answers in subsequent code blocks.
7. Submission: Your final submission should consist of a single .Rmd file. Name your file as ???????.Rmd where the question marks should be replaced with your student number.

Please adhere to these guidelines, as submissions that do not follow them cannot be graded. Following the above principles will also help you to learn how to produce reproducible code.

Introduction

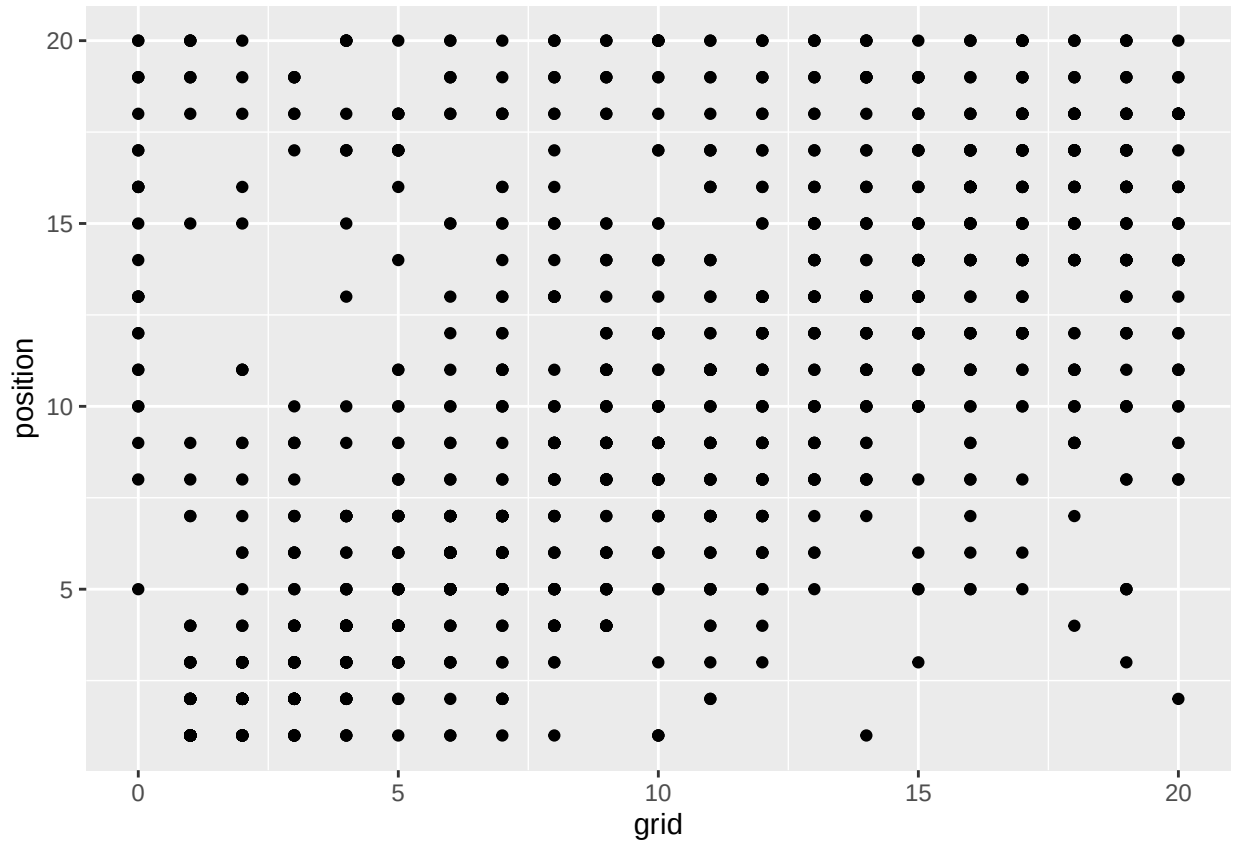
The Formula 1 team has approached you to assist them in predicting their performance for the year 2022. As a highly competitive motorsport, Formula 1 heavily relies on data-driven insights to optimize various aspects of the team's strategy and decision-making. By leveraging historical race data, we aim to build a predictive model that can forecast a driver's finishing position based on several critical factors.

This tutorial test centers around data exploration, a crucial preliminary step in the predictive modeling process. Our primary focus is to analyze the historical data, specifically the driver's finishing position, and identify the key features that significantly influence this outcome. By delving into the data, we seek to understand the relationships between various performance indicators such as the starting grid position, previous race positions, car reliability, driver reliability, team affiliation, and other pertinent variables.

Question 1: Grid position

To establish if there is a relationship between a driver's starting position i.e. `grid` and their final position i.e. `position` we will create a scatter plot. Using ggplot, create a scatter plot of the descriptive feature `grid` versus the target feature `position`. Store your answer in the variable `q1`

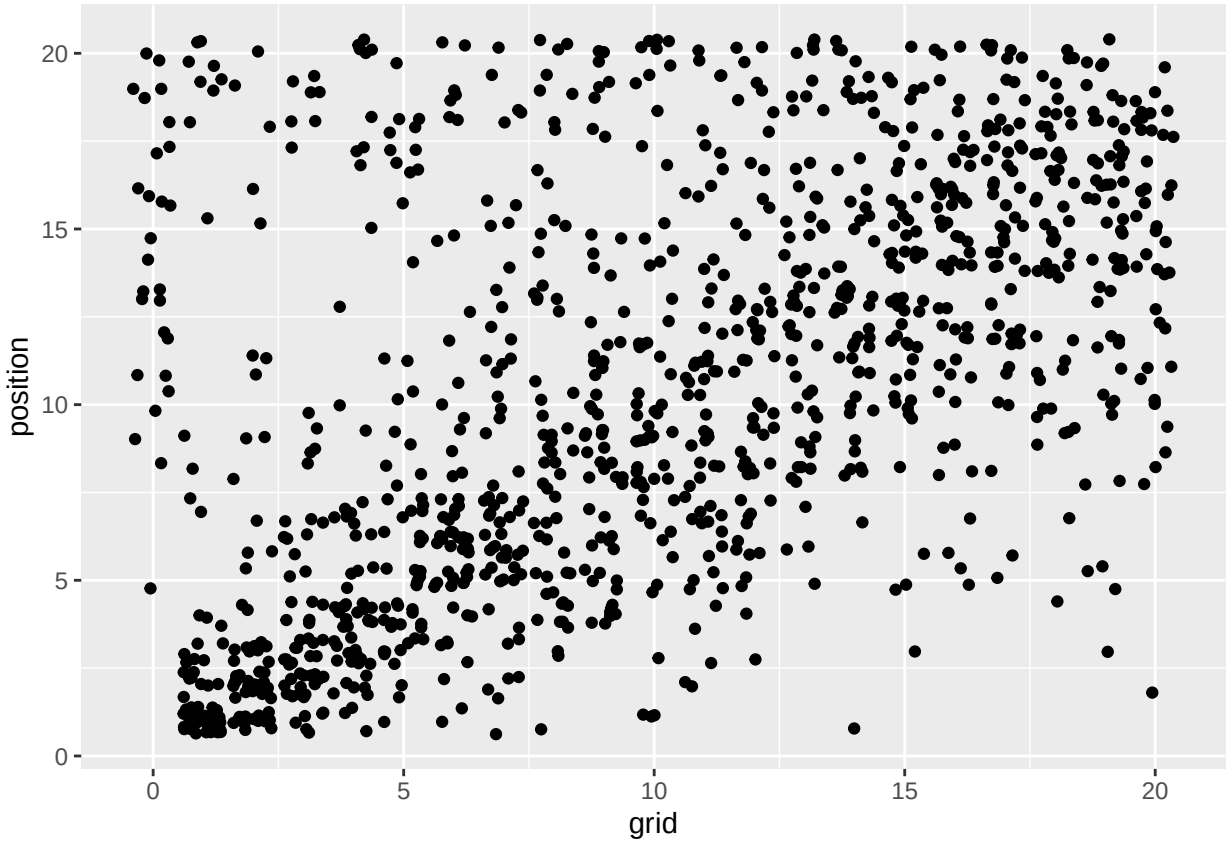
```
# Your answer here
q1 <- ggplot(data = data, aes(x = grid, y = position)) +
  geom_point()
q1
```



Question 2: Adding jitter

Since various data instances overlap we cannot visually establish whether there is a relationship between `grid` and `position`. To make it easier to establish if there is a relationship we can add a small amount of noise (`jitter`) to the points. Recreate the graph obtained in question 1, by adding `jitter` using the `position` argument of the geom `geom_point`. Store your answer in the variable `q2`

```
# Your answer here
q2 <- ggplot(data = data, aes(x = grid, y = position)) +
  geom_point(position = "jitter")
q2
```

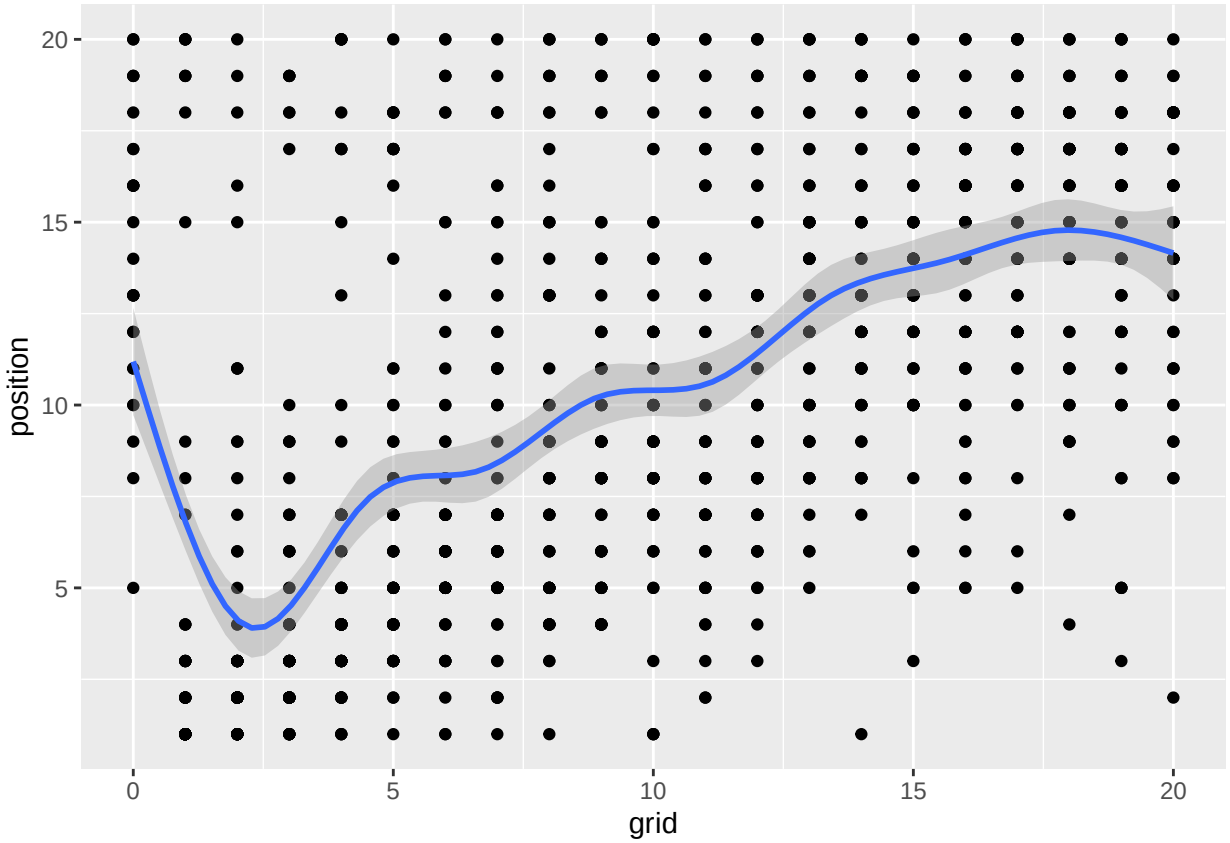


Question 3: Improved grid position

We can also highlight the relationship between `grid` and `position` by adding a `geom_smooth` layer. Add a geom_smooth layer to the plot produced in Question 1. Do not change any of the arguments. Store your answer in the variable `q3`.

```
# Your answer here
q3 <- ggplot(data = data, aes(x = grid, y = position)) +
  geom_point() +
  geom_smooth()
q3
```

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



We initially anticipated a linear relationship, but we have observed an interesting trend where drivers starting from a grid position of zero tend to finish at relatively higher positions. In Formula 1, a grid position of zero typically indicates that the driver did not qualify for the race or was unable to set a timed lap during the qualifying session. As a result, they start from the pit lane or the back of the grid, rather than from a specific starting grid position.

In the following weeks, we will explore how we can modify our dataset to facilitate an improved understanding of this relationship. One potential approach is to include a new position value i.e. 21 to represent the starting position for drivers who begin from the pit lane. This addition might provide our model with valuable insights to better capture and learn the intricacies of this particular scenario

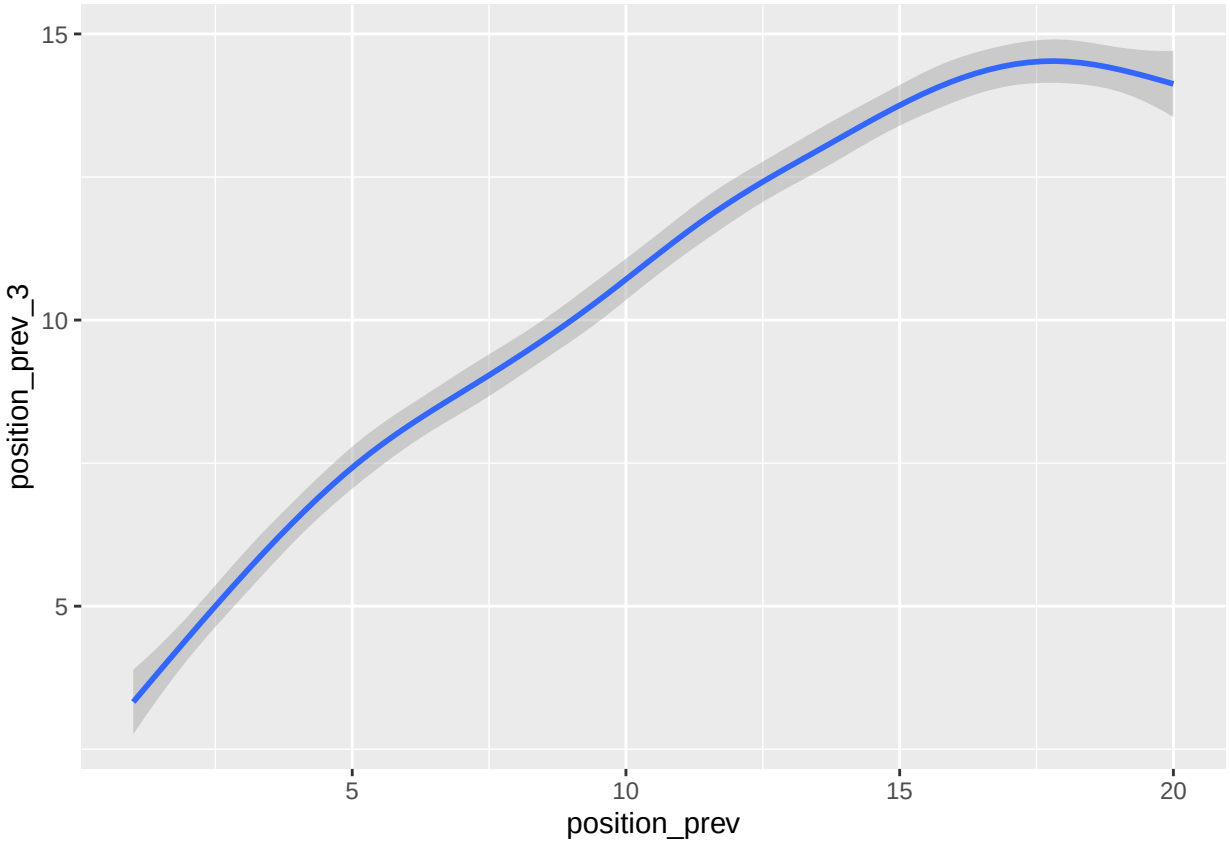
Question 4: MAXimum performance

Annoyingly some drivers manage to start at the back of the grid and end the race in a much better position. Perhaps we should account for the driver and the car. One way we can account for the driver and the car using a single descriptive feature is to analyse the relationship between a driver's previous result `position_prev` and their final position. Similarly, we also have the feature `position_prev_3` available that captures the average final position of a driver in the past three races. To analyse whether the feature `position_prev_3` and `position_prev` captures the same relationship create a plot of your choice using ggplot.

Is there a strong, weak or no linear relationship between these features? Uncomment the correct answer.

```
# Your answer here
ggplot(data = data, aes(x = position_prev, y = position_prev_3)) +
  geom_smooth()
```

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

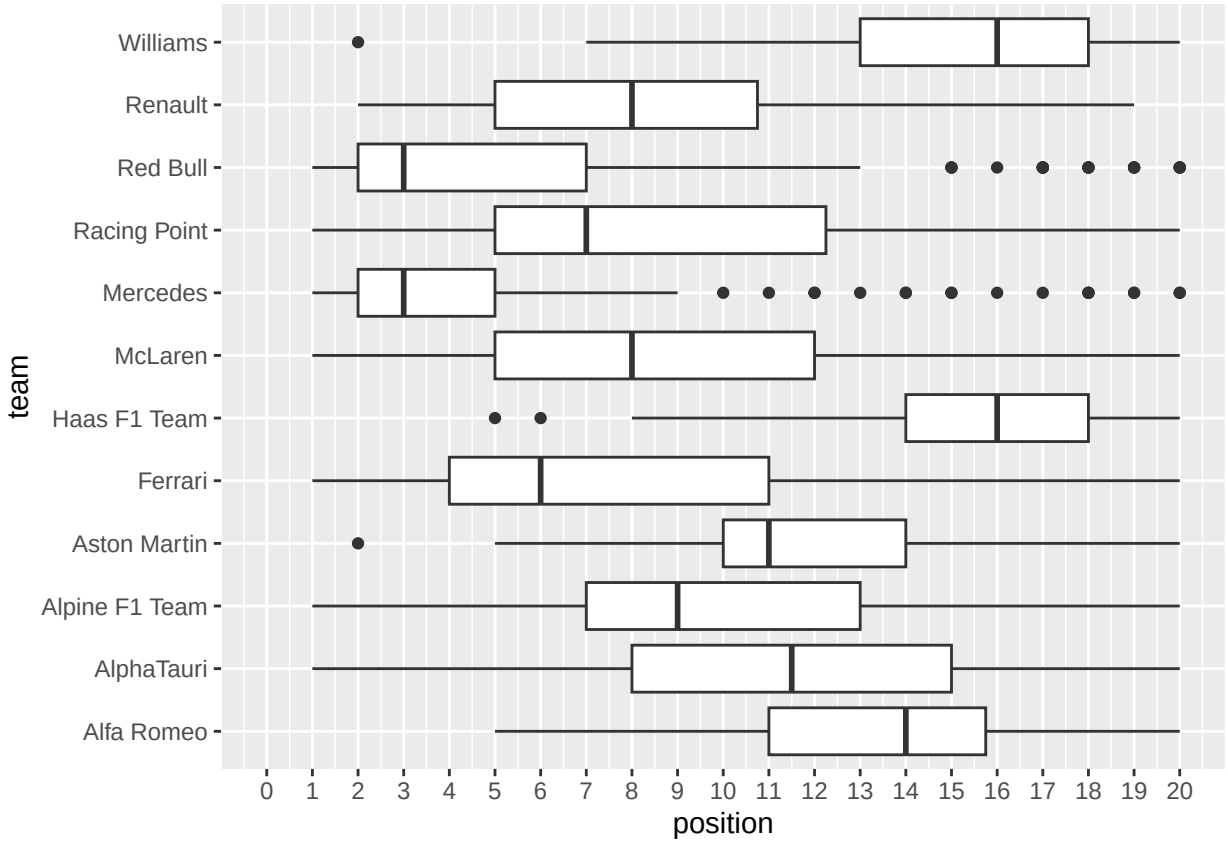



```
q4 <- "strong"
q4 <- "weak"
q4 <- "no"
```

Question 5: Betting on a team

Perhaps an easier way to predict performance is to simply bet on a team with good performance. To establish if there is a relationship between a team and performance we can create multiple boxplots of `position` one for each `team`. Using gglot create a boxplot with the team names on the vertical axis. In addition, specify the lower limit of the `position` axis to 0 and the upper limit of the position `axis` to 20. To make it easier to read the team's mean position add major breaks at intervals of 1. Store your answer in the variable `q5`.

```
# Your answer here
q5 <- ggplot(data, aes(y = team, x = position)) +
  geom_boxplot() +
  scale_x_continuous(limits = c(0,20), breaks = 0:20)
q5
```



Question 6: Reading a boxplot

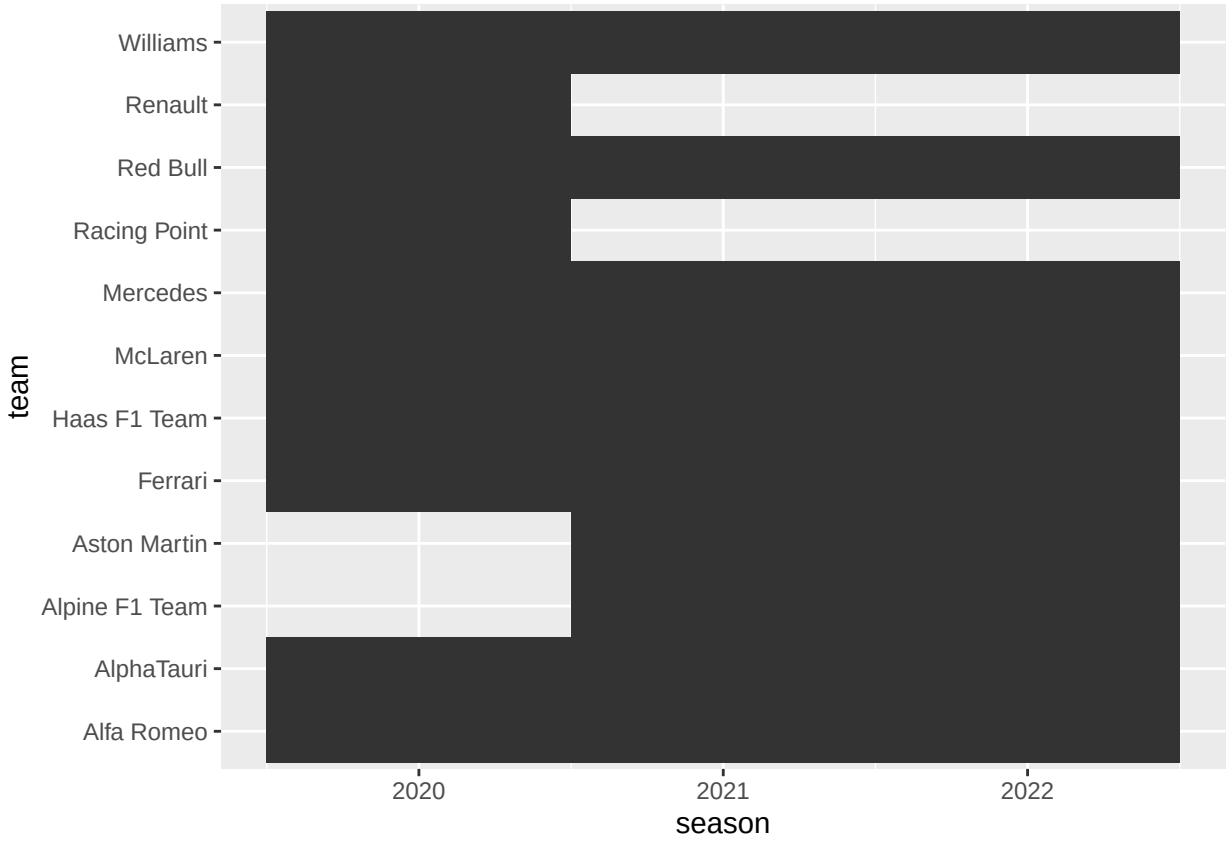
Based on the answer provided in question 5, what was the median position for Alfa Romeo? Store your answer in the variable `q6`.

```
# Your answer here
q6 <- 14
```

Question 7: Are we confident?

In class, we discussed that any relationship might simply be due to random chance. The chance of a relationship being present due to random chance increases as the amount of data that we based our findings on decreases. Although fairly constant, new teams can enter the Formula 1 grid. Using `geom_tile` create a heatmap that maps `season` on the x-axis to `teams` on the y-axis. Store your answer in the variable `q7`.

```
# Your answer here
q7 <- ggplot(data, aes(x = season, y = team)) +
  geom_tile()
q7
```



From the heatmap notice that teams can change over the years. If we build a model that uses `team` as a feature, our model should have a way to incorporate new teams.

Question 8: Interactions?

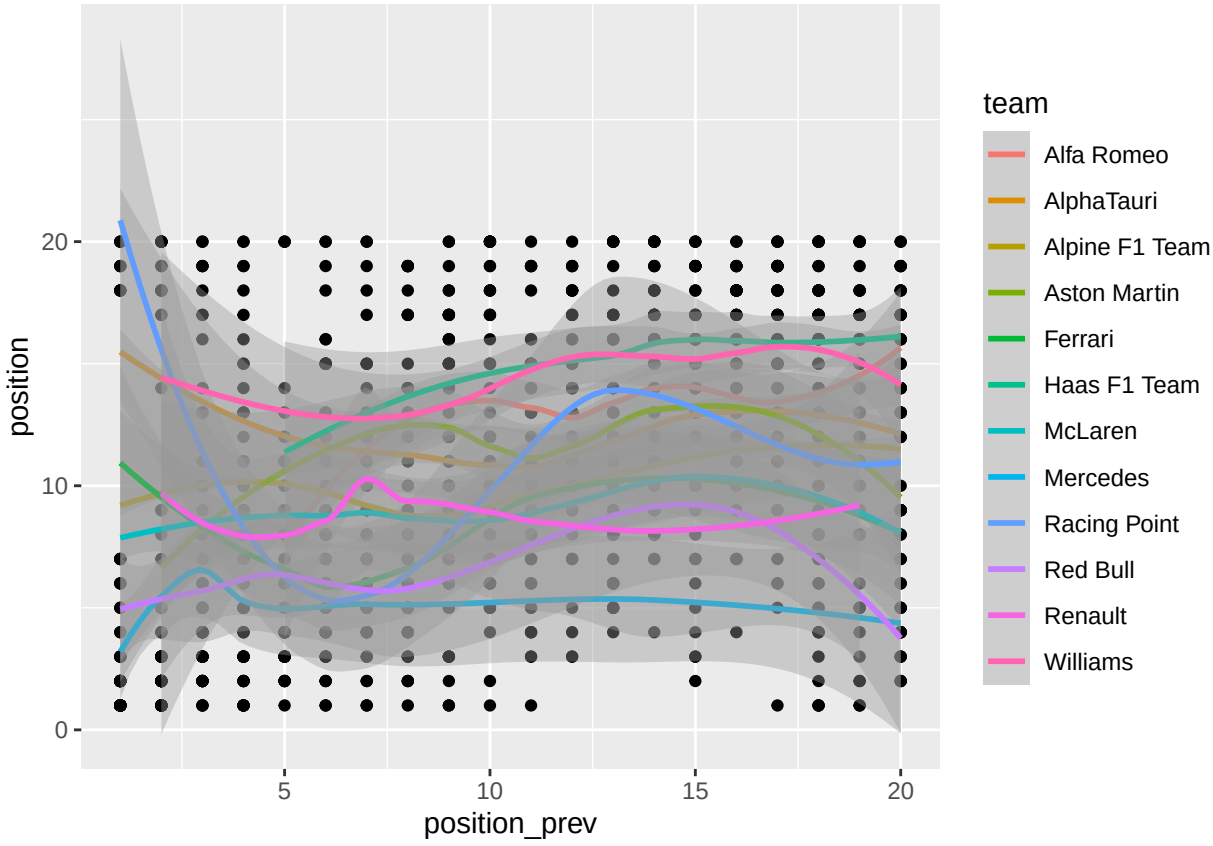
In most predictive problems, the majority of variation in target features can be explained by considering the cumulative effect of important individual predictors. For example, considering both the `previous_pos` and the `team` of a driver might reveal more complex relationships. Store your answer in the variable `q8`.

Using ggplot:

- add a geom_point layer that maps the previous grid position `position_prev` to the target feature `position` and
- add a geom_smooth layer that displays the relationship between `position_prev` and `position` for each team using the colour aesthetic.

```
# Your answer here
q8 <- ggplot(data, aes(x = position_prev, y = position, group = team)) +
  geom_point() +
  geom_smooth(aes(colour = team))
q8
```

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


position
20
10
0
5
10
15
20
position_prev
team
Alfa Romeo
AlphaTauri
Alpine F1 Team
Aston Martin
Ferrari
Haas F1 Team
McLaren
Mercedes
Racing Point
Red Bull
Renault
Williams