

APS Estatística I - Analisando os times do Brasil

Victor Alves

03/09/2021

Importando os dados

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --  
## v ggplot2 3.3.3    v purrr 0.3.4  
## v tibble 3.1.0     v dplyr 1.0.5  
## v tidyr 1.1.3      v stringr 1.4.0  
## v readr 1.4.0      v forcats 0.5.1  
  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()
```

```
library(janitor)
```

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

```
library(gridExtra)
```

```
##  
## Attaching package: 'gridExtra'  
  
## The following object is masked from 'package:dplyr':  
##  
##   combine
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(RColorBrewer)
```

```
theme_set(theme_minimal())
```

```
dados <- readxl::read_excel("C:/Users/alves/OneDrive - Insper - Instituto de Ensino e Pesquisa/Estudos/APS  
                           col_types = c("text", "text", "numeric",  
                                           "date", "text", "text", "text", "numeric",  
                                           "numeric", "numeric", "text", "numeric",  
                                           "numeric", "numeric")) %>% clean_names() %>%  
mutate('somagols' = golcasa+golvisitante)
```

Manipulando dados

Considerando a contagem de resultados por time e temporadas.

Contagem de cada resultado por temporada

```
resultado_ano <- dados %>% group_by(res, temporada) %>%  
  summarise(n = n())
```

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

Computando a quantidade de resultados do time mandante

```
resultado_ano_casa <- dados %>% group_by(res, temporada, casa) %>%  
  summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

`summarise()` has grouped output by 'res', 'temporada'. You can override using the `.groups` argument.

Computando a quantidade de resultados do time visitante

```
resultado_ano_visitante <- dados %>% group_by(res, temporada, visitante) %>%  
  summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

`summarise()` has grouped output by 'res', 'temporada'. You can override using the `.groups` argument.

Calculando algumas medidas de posição e dispersão da contagem dos resultados por time e por ano

```
medidas_res_c <- as.tibble(resultado_ano_casa %>% filter(res == 'C') %>%  
  summarise('media' = mean(n),  
            'mediana' = median(n),  
            'desvpad' = sd(n)))
```

Warning: `as.tibble()` was deprecated in tibble 2.0.0.

Please use `as_tibble()` instead.

The signature and semantics have changed, see `?as_tibble`.

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

```
medidas_res_v <- as.tibble(resultado_ano_casa %>% filter(res == 'V') %>%  
  summarise('media' = mean(n),  
            'mediana' = median(n),  
            'desvpad' = sd(n)))
```

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

```
medidas_res_e <- as.tibble(resultado_ano_casa %>% filter(res == 'E') %>%  
  summarise('media' = mean(n),  
            'mediana' = median(n),  
            'desvpad' = sd(n)))
```

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

```
medidas_res_casa <- as.tibble(resultado_ano_casa %>% group_by(casa, res, temporada) %>%  
  summarise('media' = mean(n),  
            'mediana' = median(n),  
            'desvpad' = sd(n)))
```

`summarise()` has grouped output by 'casa', 'res'. You can override using the `.groups` argument.

Trabalhando com os climas e estações do ano

Criando vetores com as estações de cada ano

```

inverno_2012 <- c(dados %>% select(data) %>% filter(data > '2012-03-20' & data < '2012-09-22'))
verao_2012 <- dados %>% select(data) %>% filter(data > '2012-09-22' & data < '2013-03-20')

inverno_2013 <- dados %>% select(data) %>% filter(data > '2013-03-20' & data < '2013-09-22')
verao_2013 <- dados %>% select(data) %>% filter(data > '2013-09-22' & data < '2014-03-20')

inverno_2014 <- dados %>% select(data) %>% filter(data > '2014-03-20' & data < '2014-09-22')
verao_2014 <- dados %>% select(data) %>% filter(data > '2014-09-22' & data < '2015-03-20')

inverno_2015 <- dados %>% select(data) %>% filter(data > '2015-03-20' & data < '2015-09-22')
verao_2015 <- dados %>% select(data) %>% filter(data > '2015-09-22' & data < '2016-03-20')

inverno_2016 <- dados %>% select(data) %>% filter(data > '2016-03-20' & data < '2016-09-22')
verao_2016 <- dados %>% select(data) %>% filter(data > '2016-09-22' & data < '2017-03-20')

inverno_2017 <- dados %>% select(data) %>% filter(data > '2017-03-20' & data < '2017-09-22')
verao_2017 <- dados %>% select(data) %>% filter(data > '2017-09-22' & data < '2018-03-20')

inverno_2018 <- dados %>% select(data) %>% filter(data > '2018-03-20' & data < '2018-09-22')
verao_2018 <- dados %>% select(data) %>% filter(data > '2018-09-22' & data < '2019-03-20')

inverno_2019 <- dados %>% select(data) %>% filter(data > '2019-03-20' & data < '2019-09-22')
verao_2019 <- dados %>% select(data) %>% filter(data > '2019-09-22' & data < '2020-03-20')

inverno_2020 <- dados %>% select(data) %>% filter(data > '2020-03-20' & data < '2020-09-22')
verao_2020 <- dados %>% select(data) %>% filter(data > '2020-09-22' & data < '2021-03-20')

```

Vetor com os estados mais quentes da amostra

```
estados_quentes <- c('Alagoas', 'Bahia', 'Ceara', 'Pernambuco', 'Goias', 'Rio de Janeiro')
```

Dataset contendo a contagem de resultados por data a cada período do dia nos estados mais quentes

```
climao_quente <- dados %>% group_by(res, data, periodo) %>% filter(estado %in% estados_quentes) %>%
  summarise(n = n())
```

`summarise()` has grouped output by 'res', 'data'. You can override using the `.groups` argument.

Dataset contendo a contagem de resultados por data a cada período do dia nos estados menos quentes

```
climao_gelado <- dados %>% group_by(res, data, periodo) %>% filter(!estado %in% estados_quentes) %>%
  summarise(n = n())
```

`summarise()` has grouped output by 'res', 'data'. You can override using the `.groups` argument.

Computando medidas da quantidade de resultados a cada estação por ano de cada estado mais quente

```
medidas_frio_quentes_2014 <- as.tibble(climao_quente %>%
  filter(data %in% inverno_2014$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_quentes_2014 <- as.tibble(climao_quente %>%
  filter(data %in% verao_2014$data) %>%
  summarise('media' = mean(n),
```

```

                                'mediana' = median(n),
                                'desvpad' = sd(n)))

## `summarise()` has grouped output by 'res'. You can override using the `.groups` argument.
medidas_frio_quentes_2015 <- as.tibble(climao_quente %>%
  filter(data %in% inverno_2015$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))

## `summarise()` has grouped output by 'res'. You can override using the `.groups` argument.
medidas_calor_quentes_2015 <- as.tibble(climao_quente %>%
  filter(data %in% verao_2015$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))

## `summarise()` has grouped output by 'res'. You can override using the `.groups` argument.
medidas_frio_quentes_2016 <- as.tibble(climao_quente %>%
  filter(data %in% inverno_2016$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))

## `summarise()` has grouped output by 'res'. You can override using the `.groups` argument.
medidas_calor_quentes_2016 <- as.tibble(climao_quente %>%
  filter(data %in% verao_2016$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))

## `summarise()` has grouped output by 'res'. You can override using the `.groups` argument.
medidas_frio_quentes_2017 <- as.tibble(climao_quente %>%
  filter(data %in% inverno_2017$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))

## `summarise()` has grouped output by 'res'. You can override using the `.groups` argument.
medidas_calor_quentes_2017 <- as.tibble(climao_quente %>%
  filter(data %in% verao_2017$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))

## `summarise()` has grouped output by 'res'. You can override using the `.groups` argument.
medidas_frio_quentes_2018 <- as.tibble(climao_quente %>%
  filter(data %in% inverno_2018$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))

```

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

```
medidas_calor_quentes_2018 <- as.tibble(climao_quente %>%
  filter(data %in% verao_2018$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_frio_quentes_2019 <- as.tibble(climao_quente %>%
  filter(data %in% inverno_2019$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_quentes_2019 <- as.tibble(climao_quente %>%
  filter(data %in% verao_2019$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_frio_quentes_2020 <- as.tibble(climao_quente %>%
  filter(data %in% inverno_2020$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_quentes_2020 <- as.tibble(climao_quente %>%
  filter(data %in% verao_2020$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

Computando medidas da quantidade de resultados a cada estação por ano de cada estado menos quente

```
medidas_frio_gelado_2014 <- as.tibble(climao_gelado %>%
  filter(data %in% inverno_2014$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_gelado_2014 <- as.tibble(climao_gelado %>%
  filter(data %in% verao_2014$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_frio_gelado_2015 <- as.tibble(climao_gelado %>%
  filter(data %in% inverno_2015$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_gelado_2015 <- as.tibble(climao_gelado %>%
  filter(data %in% verao_2015$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_frio_gelado_2016 <- as.tibble(climao_gelado %>%
  filter(data %in% inverno_2016$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_gelado_2016 <- as.tibble(climao_gelado %>%
  filter(data %in% verao_2016$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_frio_gelado_2017 <- as.tibble(climao_gelado %>%
  filter(data %in% inverno_2017$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_gelado_2017 <- as.tibble(climao_gelado %>%
  filter(data %in% verao_2017$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_frio_gelado_2018 <- as.tibble(climao_gelado %>%
  filter(data %in% inverno_2018$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_gelado_2018 <- as.tibble(climao_gelado %>%
  filter(data %in% verao_2018$data) %>%
```

```
summarise('media' = mean(n),
          'mediana' = median(n),
          'desvpad' = sd(n))
```

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

```
medidas_frio_gelado_2019 <- as.tibble(climao_gelado %>%
  filter(data %in% inverno_2019$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_gelado_2019 <- as.tibble(climao_gelado %>%
  filter(data %in% verao_2019$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_frio_gelado_2020 <- as.tibble(climao_gelado %>%
  filter(data %in% inverno_2020$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

```
medidas_calor_gelado_2020 <- as.tibble(climao_gelado %>%
  filter(data %in% verao_2020$data) %>%
  summarise('media' = mean(n),
            'mediana' = median(n),
            'desvpad' = sd(n)))
```

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

Medidas da soma de gols em cada estado (desconsiderando qual mais quente) a cada ano por estação do ano

```
medidas_sg_inverno_2014 <- as.tibble(dados %>% group_by(estado, periodo) %>%
  filter(data %in% inverno_2014$data) %>%
  summarise('media' = mean(somagols),
            'mediana' = median(somagols),
            'desvpad' = sd(somagols)))
```

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

```
medidas_sg_verao_2014 <- as.tibble(dados %>% group_by(estado, periodo) %>%
  filter(data %in% verao_2014$data) %>%
  summarise('media' = mean(somagols),
            'mediana' = median(somagols),
            'desvpad' = sd(somagols)))
```

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

```
medidas_sg_inverno_2015 <- as.tibble(dados %>% group_by(estado, periodo) %>%
  filter(data %in% inverno_2015$data) %>%
  summarise('media' = mean(somagols),
```

```

                                'mediana' = median(somagols),
                                'desvpad' = sd(somagols)))

## `summarise()` has grouped output by 'estado'. You can override using the `.groups` argument.
medidas_sg_verao_2015 <- as.tibble(dados %>% group_by(estado, periodo) %>%
                                filter(data %in% verao_2015$data) %>%
                                summarise('media' = mean(somagols),
                                          'mediana' = median(somagols),
                                          'desvpad' = sd(somagols)))

## `summarise()` has grouped output by 'estado'. You can override using the `.groups` argument.
medidas_sg_inverno_2016 <- as.tibble(dados %>% group_by(estado, periodo) %>%
                                filter(data %in% inverno_2016$data) %>%
                                summarise('media' = mean(somagols),
                                          'mediana' = median(somagols),
                                          'desvpad' = sd(somagols)))

## `summarise()` has grouped output by 'estado'. You can override using the `.groups` argument.
medidas_sg_verao_2016 <- as.tibble(dados %>% group_by(estado, periodo) %>%
                                filter(data %in% verao_2016$data) %>%
                                summarise('media' = mean(somagols),
                                          'mediana' = median(somagols),
                                          'desvpad' = sd(somagols)))

## `summarise()` has grouped output by 'estado'. You can override using the `.groups` argument.
medidas_sg_inverno_2017 <- as.tibble(dados %>% group_by(estado, periodo) %>%
                                filter(data %in% inverno_2017$data) %>%
                                summarise('media' = mean(somagols),
                                          'mediana' = median(somagols),
                                          'desvpad' = sd(somagols)))

## `summarise()` has grouped output by 'estado'. You can override using the `.groups` argument.
medidas_sg_verao_2017 <- as.tibble(dados %>% group_by(estado, periodo) %>%
                                filter(data %in% verao_2017$data) %>%
                                summarise('media' = mean(somagols),
                                          'mediana' = median(somagols),
                                          'desvpad' = sd(somagols)))

## `summarise()` has grouped output by 'estado'. You can override using the `.groups` argument.
medidas_sg_inverno_2018 <- as.tibble(dados %>% group_by(estado, periodo) %>%
                                filter(data %in% inverno_2018$data) %>%
                                summarise('media' = mean(somagols),
                                          'mediana' = median(somagols),
                                          'desvpad' = sd(somagols)))

## `summarise()` has grouped output by 'estado'. You can override using the `.groups` argument.
medidas_sg_verao_2018 <- as.tibble(dados %>% group_by(estado, periodo) %>%
                                filter(data %in% verao_2018$data) %>%
                                summarise('media' = mean(somagols),
                                          'mediana' = median(somagols),
                                          'desvpad' = sd(somagols)))

```


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

```
medidas_sg_inverno_2019 <- as.tibble(dados %>% group_by(estado, periodo) %>%  
  filter(data %in% inverno_2019$data) %>%  
  summarise('media' = mean(somagols),  
            'mediana' = median(somagols),  
            'desvpad' = sd(somagols)))
```

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

```
medidas_sg_verao_2019 <- as.tibble(dados %>% group_by(estado, periodo) %>%  
  filter(data %in% vero_2019$data) %>%  
  summarise('media' = mean(somagols),  
            'mediana' = median(somagols),  
            'desvpad' = sd(somagols)))
```

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

```
medidas_sg_inverno_2020 <- as.tibble(dados %>% group_by(estado, periodo) %>%  
  filter(data %in% inverno_2020$data) %>%  
  summarise('media' = mean(somagols),  
            'mediana' = median(somagols),  
            'desvpad' = sd(somagols)))
```

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

```
medidas_sg_verao_2020 <- as.tibble(dados %>% group_by(estado, periodo) %>%  
  filter(data %in% vero_2020$data) %>%  
  summarise('media' = mean(somagols),  
            'mediana' = median(somagols),  
            'desvpad' = sd(somagols)))
```

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

Trabalhando com a soma dos gols por período do dia

Dataset contendo a soma dos gols por estado e por período

```
somagols_estado_periodo <- dados %>% group_by(somagols, estado, periodo)
```

Medidas das soma de gols por período e por estado

```
medidas_gols_estado_periodo <- as.tibble(somagols_estado_periodo %>% group_by(estado, periodo) %>%  
  summarise('media' = mean(somagols),  
            'mediana' = median(somagols),  
            'desvpad' = sd(somagols)))
```

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

Medidas da soma de gols por estado

```
medidas_gols_estado <- as.tibble(somagols_estado_periodo %>% group_by(estado) %>%  
  summarise('media' = mean(somagols),  
            'mediana' = median(somagols),  
            'desvpad' = sd(somagols)))
```

Trabalhando com os gols da casa quando ele ganha

Calculando medidas dos gols da casa e visitante, por ano

```
medidas_gols_ano <- as.tibble(dados %>% group_by(temporada)%>%
  summarise('media casa' = mean(golcasa),
            'mediana casa' = median(golcasa),
            'desvpad casa' = sd(golcasa),
            'media visitante' = mean(golvisitante),
            'mediana visitante' = median(golvisitante),
            'desvpad visitante' = sd(golvisitante)))
```

Gerando frequências

Obtendo a frequência de jogos por período do dia

```
dados %>% group_by(periodo) %>% summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

```
## # A tibble: 2 x 3
##   periodo      n freq
##   <chr>    <int> <dbl>
## 1 Noite    1953  57.1
## 2 Tarde    1467  42.9
```

Obtendo a frequência de jogos por estado

```
dados %>% group_by(estados) %>% summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

```
## # A tibble: 11 x 3
##   estado      n freq
##   <chr>    <int> <dbl>
## 1 Alagoas      19  0.56
## 2 Bahia      228  6.67
## 3 Ceara       95  2.78
## 4 Goias      152  4.44
## 5 Minas Gerais 361 10.6
## 6 Parana      304  8.89
## 7 Pernambuco  190  5.56
## 8 Rio de Janeiro 627 18.3
## 9 Rio Grande do Sul 323  9.44
## 10 Santa Catarina 304  8.89
## 11 Sao Paulo   817 23.9
```

Obtendo a frequência de jogos por temporada por período

```
dados %>% group_by(periodo, temporada) %>% summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

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

```
## # A tibble: 18 x 4
## # Groups:   periodo [2]
##   periodo temporada      n freq
##   <chr>      <dbl> <int> <dbl>
## 1 Noite      2012    236 12.1
## 2 Noite      2013    231 11.8
## 3 Noite      2014    225 11.5
## 4 Noite      2015    191  9.78
## 5 Noite      2016    183  9.37
## 6 Noite      2017    192  9.83
## 7 Noite      2018    200 10.2
## 8 Noite      2019    231 11.8
```

```
## 9 Noite      2020    264 13.5
## 10 Tarde     2012    144  9.82
## 11 Tarde     2013    149 10.2
## 12 Tarde     2014    155 10.6
## 13 Tarde     2015    189 12.9
## 14 Tarde     2016    197 13.4
## 15 Tarde     2017    188 12.8
## 16 Tarde     2018    180 12.3
## 17 Tarde     2019    149 10.2
## 18 Tarde     2020    116  7.91
```

Obtendo a frequência de jogos por temporada por estado

```
dados %>% group_by(periodo, estado) %>% summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

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

```
## # A tibble: 22 x 4
## # Groups:   periodo [2]
##   periodo estado      n freq
##   <chr>   <chr>   <int> <dbl>
## 1 Noite   Alagoas      14  0.72
## 2 Noite   Bahia      122  6.25
## 3 Noite   Ceara       68  3.48
## 4 Noite   Goias      100  5.12
## 5 Noite   Minas Gerais 206 10.6
## 6 Noite   Parana      159  8.14
## 7 Noite   Pernambuco  118  6.04
## 8 Noite   Rio de Janeiro 368 18.8
## 9 Noite   Rio Grande do Sul 166  8.5
## 10 Noite  Santa Catarina 174  8.91
## # ... with 12 more rows
```

Obtendo a frequência de gols totais por estado

```
dados %>% group_by(somagols) %>% summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

```
## # A tibble: 10 x 3
##   somagols      n freq
##   <dbl> <int> <dbl>
## 1      0   308  9.01
## 2      1   782 22.9
## 3      2   868 25.4
## 4      3   748 21.9
## 5      4   406 11.9
## 6      5   183  5.35
## 7      6    78  2.28
## 8      7    32  0.94
## 9      8    13  0.38
## 10     9     2  0.06
```

Obtendo a frequência de gols totais por estado

```
dados %>% group_by(somagols, estado) %>% summarise(n = n()) %>% mutate(freq = round((n / sum(n))*100,2))
```

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

```
## # A tibble: 92 x 4
## # Groups:   somagols [10]
```

```
##      somagols estado          n freq
##      <dbl> <chr>          <int> <dbl>
## 1         0 Alagoas           2 0.65
## 2         0 Bahia            22 7.14
## 3         0 Ceara            13 4.22
## 4         0 Goias            12 3.9
## 5         0 Minas Gerais     31 10.1
## 6         0 Parana           27 8.77
## 7         0 Pernambuco       17 5.52
## 8         0 Rio de Janeiro   46 14.9
## 9         0 Rio Grande do Sul 31 10.1
## 10        0 Santa Catarina   38 12.3
## # ... with 82 more rows
```

Plots

Plot dos resultados ao longo dos anos

Distribuição da quantidade dos resultados a cada temporada

```
dist_res_2020 <- resultado_ano %>%
  group_by(temporada) %>%
  ggplot(aes(temporada, n, colour = res)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Quantidade de vitórias do mandante por temporada') +
  xlab('Temporada') + ylab('Quantidade') + labs(col = 'Resultado') +
  ggeasy::easy_center_title()
```

Distribuição da quantidade de vitórias da casa por temporada

```
dist_c_casa_ano <- resultado_ano_casa %>% filter(res == 'C') %>%
  mutate(casa = fct_reorder(casa, n)) %>%
  group_by(casa) %>%
  ggplot(aes(temporada, casa, colour = n)) + geom_point() +
  scale_color_viridis(discrete = FALSE) +
  ggtitle('Quantidade de vitórias da casa por ano') +
  xlab('Temporada') + ylab('Time') + labs(col = 'Quantidade') +
  ggeasy::easy_center_title()
```

Distribuição da quantidade de derrotas da casa por temporada

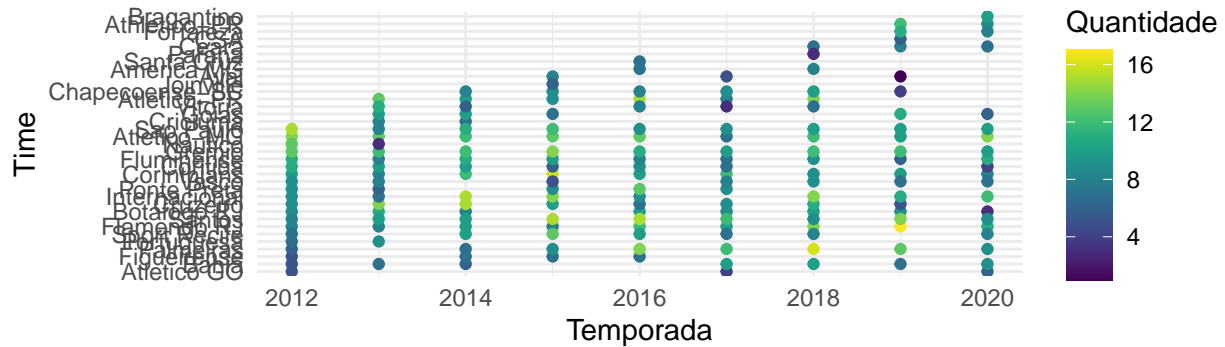
```
dist_v_casa_ano <- resultado_ano_casa %>% filter(res == 'V') %>%
  mutate(casa = fct_reorder(casa, n)) %>%
  group_by(casa) %>%
  ggplot(aes(temporada, casa, colour = n)) + geom_point() +
  scale_color_viridis(discrete = FALSE) +
  ggtitle('Quantidade de derrotas da casa por ano') +
  xlab('Temporada') + ylab('Time') + labs(col = 'Quantidade') +
  ggeasy::easy_center_title()
```

Unindo a distribuição das vitórias e derrotas do time mandante

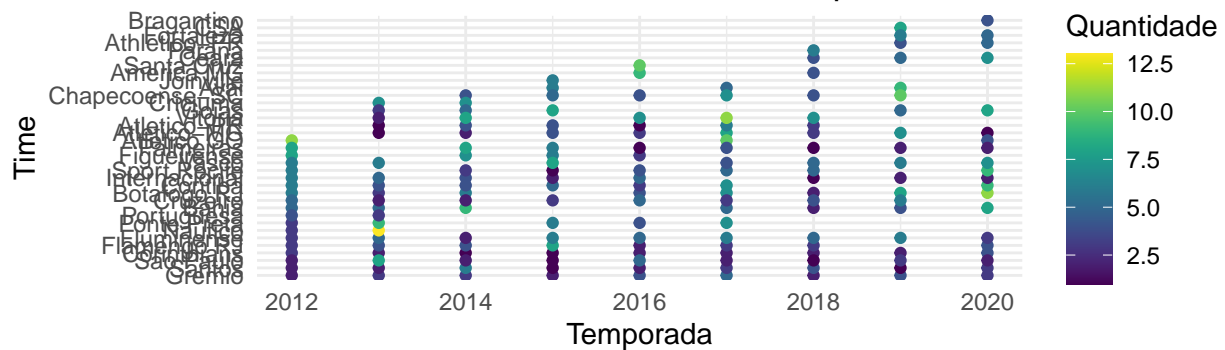
```
grid.arrange(arrangeGrob(dist_c_casa_ano, dist_v_casa_ano),
              top = 'Distribuição da quantidade de resultados do time mandante')
```

Distribuição da quantidade de resultados do time mandante

Quantidade de vitórias da casa por ano



Quantidade de derrotas da casa por ano



Plot de gols

Distribuição da média de gols por estado e período

```
dist_med_gols_perodo <- medidas_gols_estado_perodo %>%
  mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols a tarde por estado') +
  xlab('Média de gols') + ylab('Estado') +
  ggeasy::easy_center_title()
```

Distribuição da média de gols por estado

```
dist_med_gols_estado <- medidas_gols_estado %>%
  mutate(estado = fct_reorder(estado, media)) %>%
  ggplot(aes(media, estado, colour = media)) + geom_point() +
  scale_color_viridis(discrete = FALSE) +
  ggtitle('Media de gols da casa por estado') +
  xlab('Ano') + ylab('Media de gols') +
  theme(legend.position="none")
ggeasy::easy_center_title()
```

```
## List of 1
## $ plot.title:List of 11
## ..$ family : NULL
```

```
## ..$ face          : NULL
## ..$ colour        : NULL
## ..$ size          : NULL
## ..$ hjust         : num 0.5
## ..$ vjust         : NULL
## ..$ angle         : NULL
## ..$ lineheight    : NULL
## ..$ margin        : NULL
## ..$ debug         : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi FALSE
## - attr(*, "validate")= logi TRUE
```

Plots medias gol por estado e média gol por temporada

Distribuição dos gols do mandante por ano

```
medias_gols_casa_ano <- medidas_gols_ano %>%
  ggplot(aes(temporada, `media casa`, colour = `media casa`)) + geom_point() +
  scale_color_viridis(discrete = FALSE) +
  ggtitle('Media de gols da casa por ano') +
  xlab('Ano') + ylab('Media de gols') +
  theme(legend.position="none")
  ggeasy::easy_center_title()
```

```
## List of 1
## $ plot.title:List of 11
## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : num 0.5
## ..$ vjust       : NULL
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : NULL
## ..$ debug       : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi FALSE
## - attr(*, "validate")= logi TRUE
```

Distribuição dos gols do visitante por ano

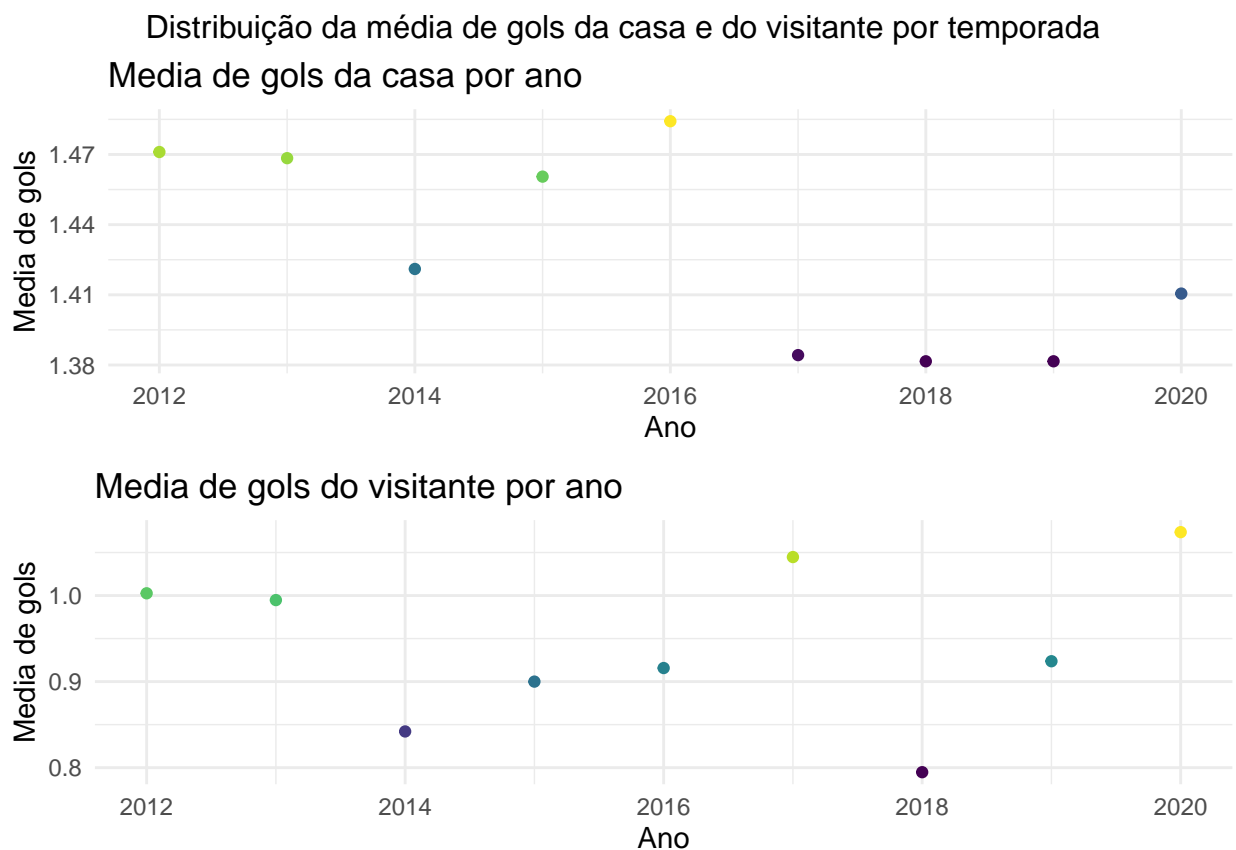
```
medias_gols_visitante_ano <- medidas_gols_ano %>%
  ggplot(aes(temporada, `media visitante`, colour = `media visitante`)) + geom_point() +
  scale_color_viridis(discrete = FALSE) +
  ggtitle('Media de gols do visitante por ano') +
  xlab('Ano') + ylab('Media de gols') +
  theme(legend.position="none")
  ggeasy::easy_center_title()
```

```
## List of 1
```

```
## $ plot.title:List of 11
## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : num 0.5
## ..$ vjust       : NULL
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : NULL
## ..$ debug       : NULL
## ..$ inherit.blank: logi FALSE
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi FALSE
## - attr(*, "validate")= logi TRUE
```

Unindo os plots da média de gols da casa e do visitante

```
grid.arrange(arrangeGrob(medias_gols_casa_ano, medias_gols_visitante_ano),
              top = 'Distribuição da média de gols da casa e do visitante por temporada')
```



Distribuição da soma de gols por estado

```
dist_sg_al <- dados %>% filter(estado == 'Alagoas') %>%
  ggplot(aes(somagols, y = stat(density), fill = periodo)) +
  geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0, 4) +
  scale_fill_viridis(discrete = TRUE) +
```

```

ggtitle('Distribuição da soma dos gols por período em Alagoas') +
xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
ggeasy::easy_center_title()

dist_sg_ba <- dados %>% filter(estado == 'Bahia') %>%
  ggplot(aes(somagols, y = stat(density), fill = periodo)) +
  geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0,8) +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição da soma dos gols por período na Bahia') +
  xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
  ggeasy::easy_center_title()

dist_sg_ce <- dados %>% filter(estado == 'Ceara') %>%
  ggplot(aes(somagols, y = stat(density), fill = periodo)) +
  geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0,5) +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição da soma dos gols por período no Ceara') +
  xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
  ggeasy::easy_center_title()

dist_sg_go <- dados %>% filter(estado == 'Goias') %>%
  ggplot(aes(somagols, y = stat(density), fill = periodo)) +
  geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0,8) +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição da soma dos gols por período em Goias') +
  xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
  ggeasy::easy_center_title()

dist_sg_mg <- dados %>% filter(estado == 'Minas Gerais') %>%
  ggplot(aes(somagols, y = stat(density), fill = periodo)) +
  geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0,8) +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição da soma dos gols por período em Minas Gerais') +
  xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
  ggeasy::easy_center_title()

dist_sg_pr <- dados %>% filter(estado == 'Parana') %>%
  ggplot(aes(somagols, y = stat(density), fill = periodo)) +
  geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0, 8) +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição da soma dos gols por período no Parana') +
  xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
  ggeasy::easy_center_title()

dist_sg_pb <- dados %>% filter(estado == 'Pernambuco') %>%
  ggplot(aes(somagols, y = stat(density), fill = periodo)) +
  geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0, 8) +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição da soma dos gols por período no Pernambuco') +
  xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
  ggeasy::easy_center_title()

dist_sg_rs <- dados %>% filter(estado == 'Rio Grande do Sul') %>%

```



```

ggplot(aes(somagols, y = stat(density), fill = periodo)) +
geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0, 8) +
scale_fill_viridis(discrete = TRUE) +
ggtitle('Distribuição da soma dos gols por período no Rio Grande do Sul') +
xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
ggeasy::easy_center_title()

dist_sg_rj <- dados %>% filter(estado == 'Rio de Janeiro') %>%
ggplot(aes(somagols, y = stat(density), fill = periodo)) +
geom_histogram(color = 'black', breaks = seq(0,10, 1)) + xlim(0,8) +
scale_fill_viridis(discrete = TRUE) +
ggtitle('Distribuição da soma dos gols por período no Rio de Janeiro') +
xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
ggeasy::easy_center_title()

dist_sg_sc <- dados %>% filter(estado == 'Santa Catarina') %>%
ggplot(aes(somagols, y = stat(density), fill = periodo)) +
geom_histogram(color = 'black', breaks = seq(0,10, 1)) +
scale_x_continuous(breaks = seq(0, 9, 1)) +
scale_fill_viridis(discrete = TRUE) +
ggtitle('Distribuição da soma dos gols por período em Santa Catarina') +
xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
ggeasy::easy_center_title()

dist_sg_sp <- dados %>% filter(estado == 'Sao Paulo') %>%
ggplot(aes(somagols, y = stat(density), fill = periodo)) +
geom_histogram(color = 'black', breaks = seq(0,10, 1)) +
scale_fill_viridis(discrete = TRUE) + xlim(0,8) +
ggtitle('Distribuição da soma dos gols por período em São Paulo') +
xlab('Soma dos gols por jogos') + ylab('Densidade') + labs(fill = 'Período') +
ggeasy::easy_center_title()

```

Unindo os plots da soma de gols por estado

```

dist_sg1 <- grid.arrange(arrangeGrob(dist_sg_al, dist_sg_ba, dist_sg_ce,
                                     dist_sg_go, dist_sg_mg, dist_sg_pb),
                        top = 'Distribuição da quantidade total de gols por estado e período (1)')

```

```

## Warning: Removed 12 rows containing missing values (geom_bar).
## Warning: Removed 4 rows containing missing values (geom_bar).
## Warning: Removed 10 rows containing missing values (geom_bar).
## Warning: Removed 4 rows containing missing values (geom_bar).

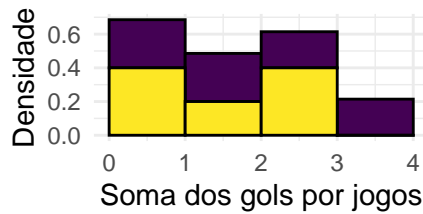
## Warning: Removed 4 rows containing missing values (geom_bar).

## Warning: Removed 4 rows containing missing values (geom_bar).

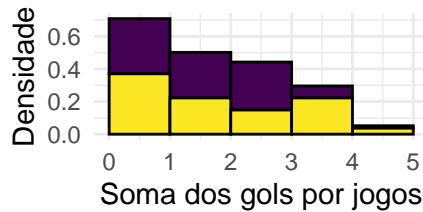
```

Distribuição da quantidade total de gols por estado e período (1)

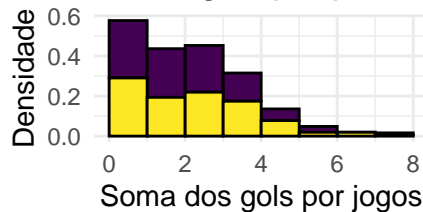
o da soma dos gols por período Distribuição da soma dos gols por período na Bar



ão da soma dos gols por período Distribuição da soma dos gols por período em Go



da soma dos gols por período Distribuição da soma dos gols por período no Perna



```
dist_sg2 <- grid.arrange(arrangeGrob(dist_sg_pr, dist_sg_rj, dist_sg_rs,
                                     dist_sg_sc, dist_sg_sp,
                                     top = 'Distribuição da quantidade total de gols por estado e período (2)'))
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```

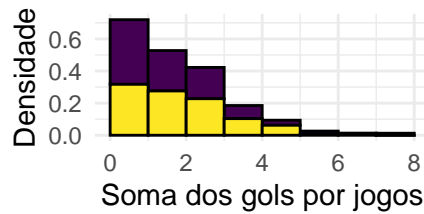
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```

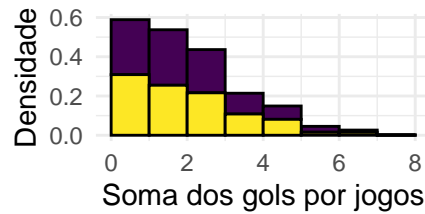
```
## Warning: Removed 4 rows containing missing values (geom_bar).
```

Distribuição da quantidade total de gols por estado e período (2)

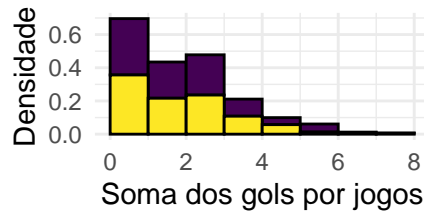
Dist. Rio de Janeiro



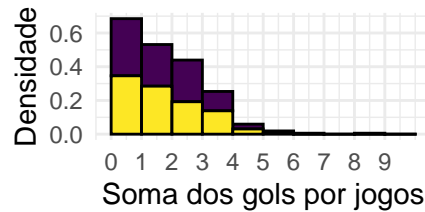
Dist. Rio Grande do Sul



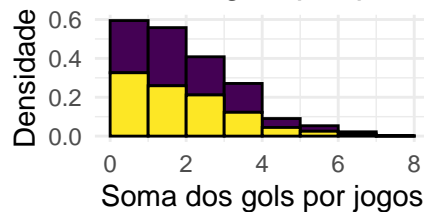
Dist. Santa Catarina



Dist. São Paulo



Dist. São Paulo



Plots dos resultados por estação do ano

Distribuição da quantidade de resultados nos estados “não quentes”

```
dist_frio_gelados2012 <- climao_gelado %>%
  filter(data %in% inverno_2012$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2012') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()
```

```
dist_calor_gelados2012 <- climao_gelado %>%
  filter(data %in% verao_2012$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2012') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()
```

```
dist_frio_gelados2013 <- climao_gelado %>%
  filter(data %in% inverno_2013$data) %>%
  ggplot(aes(res, fill = res)) +
```

```

geom_bar(alpha = .6,color = 'black') +
scale_fill_viridis(discrete = TRUE) +
ggtitle('Distribuição de resultado nos estados frios no inverno de 2013') +
xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
ggeasy::easy_center_title()

dist_calor_gelados2013 <- climao_gelado %>%
  filter(data %in% verao_2013$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2013') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_gelados2014 <- climao_gelado %>%
  filter(data %in% inverno_2014$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2014') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_gelados2014 <- climao_gelado %>%
  filter(data %in% verao_2014$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2014') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_gelados2015 <- climao_gelado %>%
  filter(data %in% inverno_2015$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2015') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_gelados2015 <- climao_gelado %>%
  filter(data %in% verao_2015$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2015') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

```

```

dist_frio_gelados2016 <- climao_gelado %>%
  filter(data %in% inverno_2016$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2016') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_gelados2016 <- climao_gelado %>%
  filter(data %in% verao_2016$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2016') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_gelados2017 <- climao_gelado %>%
  filter(data %in% inverno_2017$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2017') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_gelados2017 <- climao_gelado %>%
  filter(data %in% verao_2017$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2017') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_gelados2018 <- climao_gelado %>%
  filter(data %in% inverno_2018$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2018') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_gelados2018 <- climao_gelado %>%
  filter(data %in% verao_2018$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2018') +

```

```

xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
ggeasy::easy_center_title()

dist_frio_gelados2019 <- climao_gelado %>%
  filter(data %in% inverno_2019$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2019') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_gelados2019 <- climao_gelado %>%
  filter(data %in% verao_2019$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2019') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_gelados2020 <- climao_gelado %>%
  filter(data %in% inverno_2020$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no inverno de 2020') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_gelados2020 <- climao_gelado %>%
  filter(data %in% verao_2020$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados frios no verão de 2020') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

```

Distribuição da quantidade de resultados nos estados mais quentes

```

dist_frio_quentes2012 <- climao_quente %>%
  filter(data %in% inverno_2012$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2012') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2012 <- climao_quente %>%
  filter(data %in% verao_2012$data) %>%

```

```

ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2012') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2013 <- climao_quente %>%
  filter(data %in% inverno_2013$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2013') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2013 <- climao_quente %>%
  filter(data %in% verao_2013$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2013') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2014 <- climao_quente %>%
  filter(data %in% inverno_2014$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2014') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2014 <- climao_quente %>%
  filter(data %in% verao_2014$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2014') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2015 <- climao_quente %>%
  filter(data %in% inverno_2015$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2015') +

```

```

xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
ggeasy::easy_center_title()

dist_calor_quentes2015 <- climao_quente %>%
  filter(data %in% verao_2015$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2015') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2016 <- climao_quente %>%
  filter(data %in% inverno_2016$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2016') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2016 <- climao_quente %>%
  filter(data %in% verao_2016$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2016') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2017 <- climao_quente %>%
  filter(data %in% inverno_2017$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2017') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2017 <- climao_quente %>%
  filter(data %in% verao_2017$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6, color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2017') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2018 <- climao_quente %>%
  filter(data %in% inverno_2018$data) %>%

```



```

ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2018') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2018 <- climao_quente %>%
  filter(data %in% verao_2018$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2018') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2019 <- climao_quente %>%
  filter(data %in% inverno_2019$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2019') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2019 <- climao_quente %>%
  filter(data %in% verao_2019$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2019') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_frio_quentes2020 <- climao_quente %>%
  filter(data %in% inverno_2020$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no inverno de 2020') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

dist_calor_quentes2020 <- climao_quente %>%
  filter(data %in% verao_2020$data) %>%
  ggplot(aes(res, fill = res)) +
  geom_bar(alpha = .6,color = 'black') +
  scale_fill_viridis(discrete = TRUE) +
  ggtitle('Distribuição de resultado nos estados quentes no verão de 2020') +
  xlab('Quantidade de resultados') + ylab('Ocorrências') + labs(fill = 'Resultado') +
  ggeasy::easy_center_title()

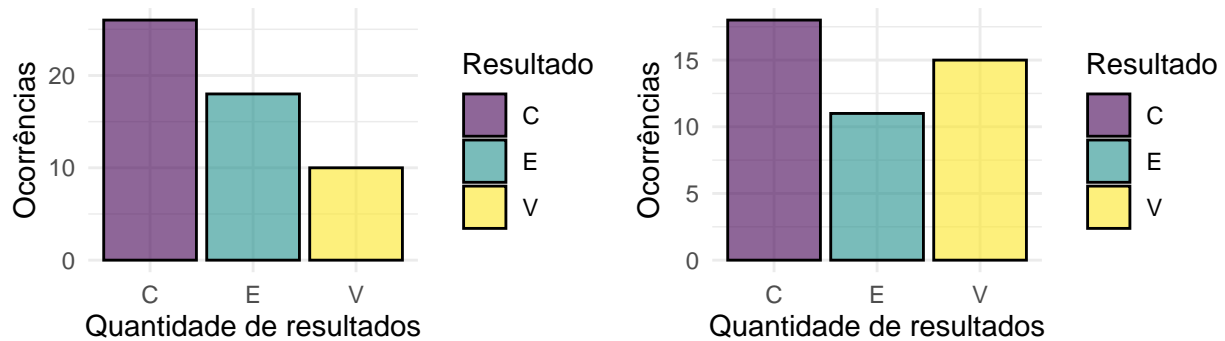
```

Unindo os plots para comparação de resultados ano-año clima-clima

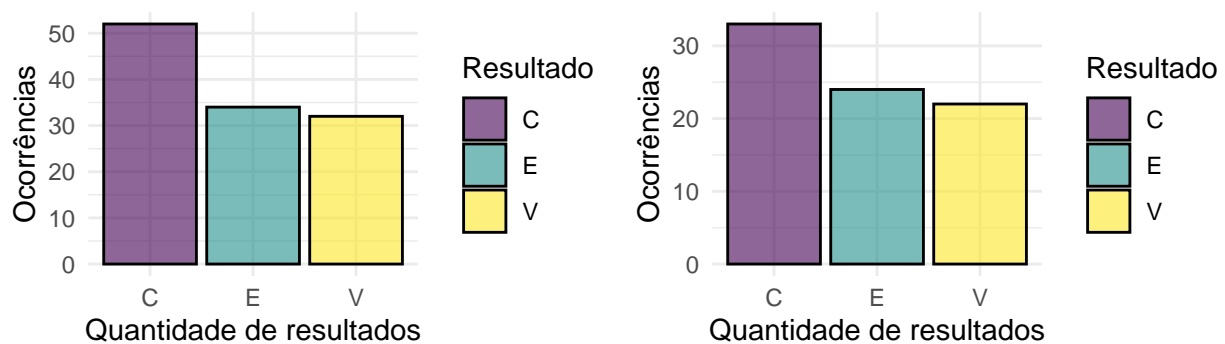
```
dist_res2012 <- grid.arrange(arrangeGrob(dist_calor_gelados2012, dist_calor_quentes2012,
  dist_frio_gelados2012, dist_frio_quentes2012),
  top = 'Distribuição dos resultados em 2012')
```

Distribuição dos resultados em 2012

e resultado nos estados frios no verão e resultado nos estados quentes no verão



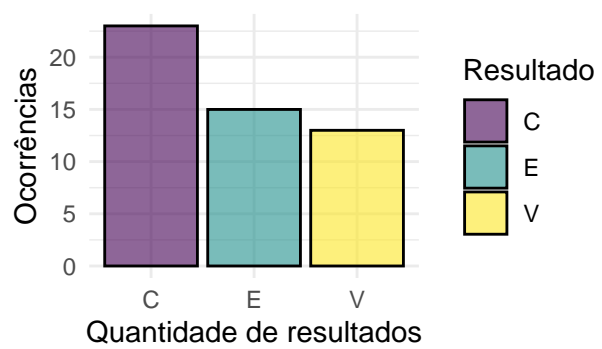
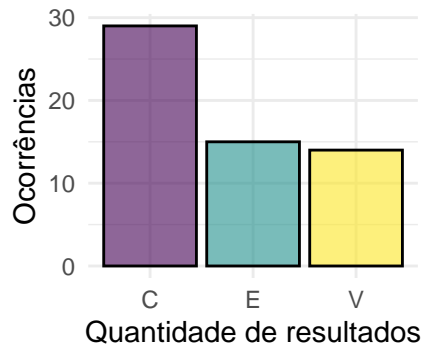
resultado nos estados frios no inverno e resultado nos estados quentes no inverno



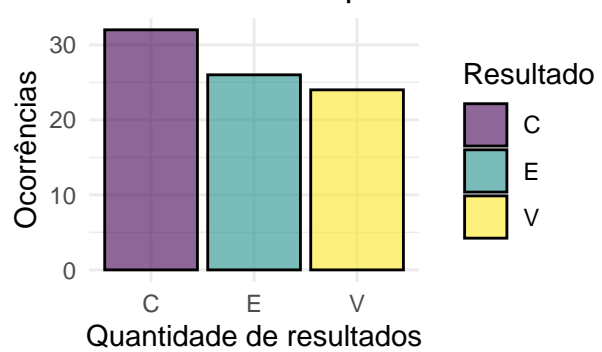
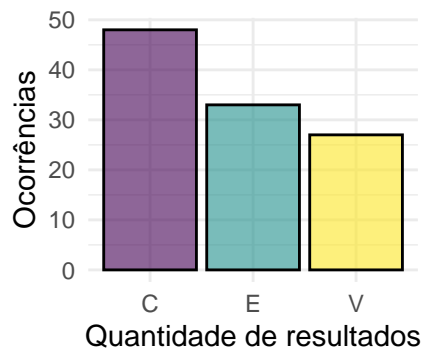
```
dist_res2013 <- grid.arrange(arrangeGrob(dist_calor_gelados2013, dist_calor_quentes2013,
  dist_frio_gelados2013, dist_frio_quentes2013),
  top = 'Distribuição dos resultados em 2013')
```

Distribuição dos resultados em 2013

e resultado nos estados frios Distribuição de resultado nos estados quentes no verão

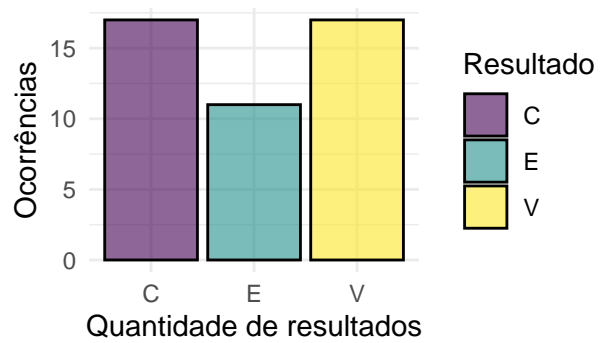
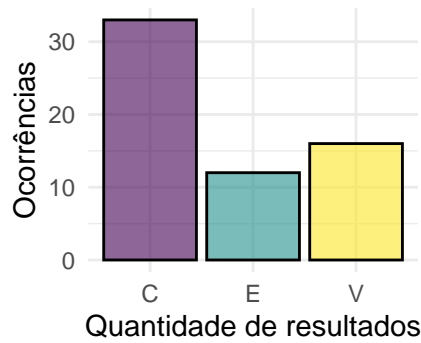


e resultado nos estados frios Distribuição de resultado nos estados quentes no inverno

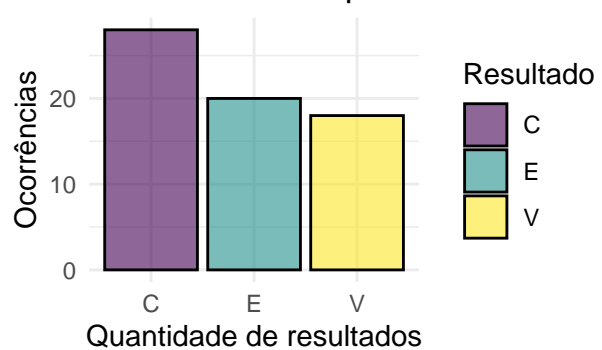
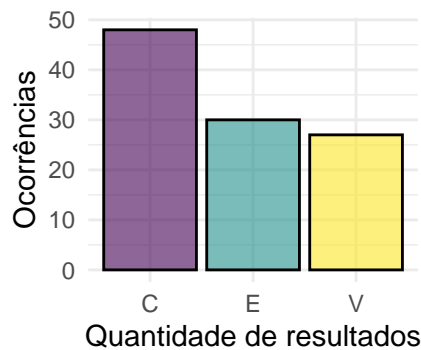


```
dist_res2014 <- grid.arrange(arrangeGrob(dist_calor_gelados2014, dist_calor_quentes2014,
  dist_frio_gelados2014, dist_frio_quentes2014),
  top = 'Distribuição dos resultados em 2014')
```

Distribuição dos resultados em 2014
e resultado nos estados frios Distribuição de resultado nos estados quentes no verão

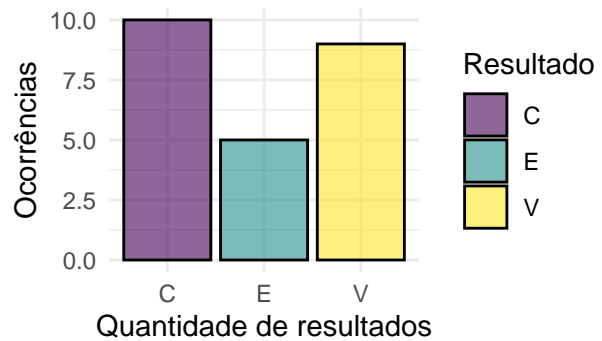
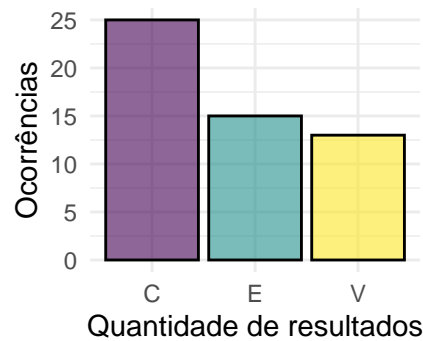


resultado nos estados frios Distribuição de resultado nos estados quentes no inverno

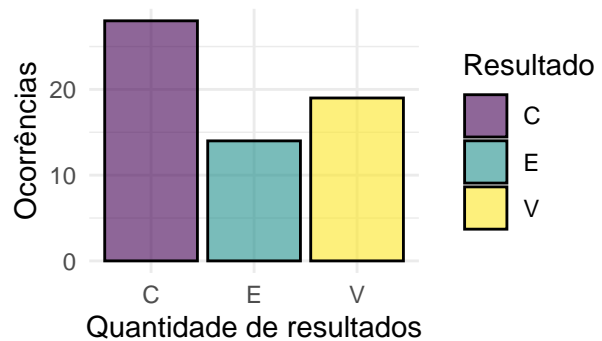
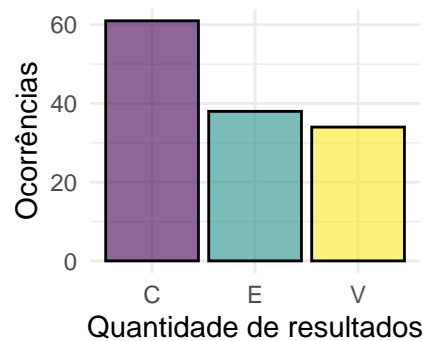


```
dist_res2015 <- grid.arrange(arrangeGrob(dist_calor_gelados2015, dist_calor_quentes2015,
dist_frio_gelados2015, dist_frio_quentes2015),
top = 'Distribuição dos resultados em 2015')
```

Distribuição dos resultados em 2015
e resultado nos estados frios Distribuição de 2015 e resultado nos estados quentes no verão

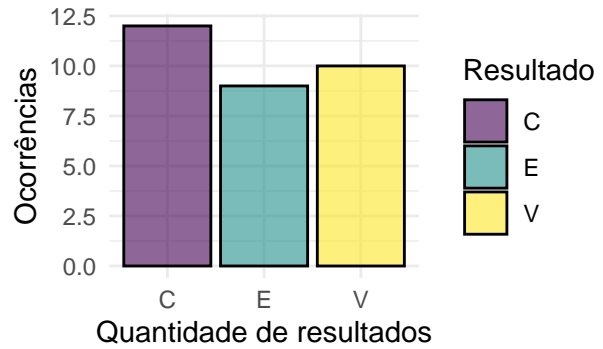
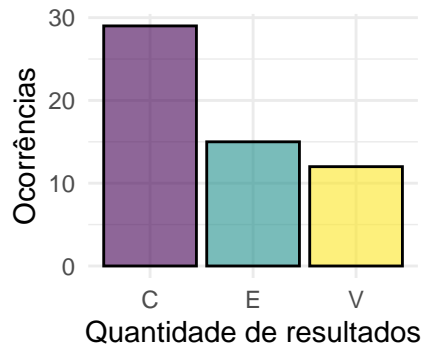


o resultado nos estados frios Distribuição de 2015 e resultado nos estados quentes no inverno

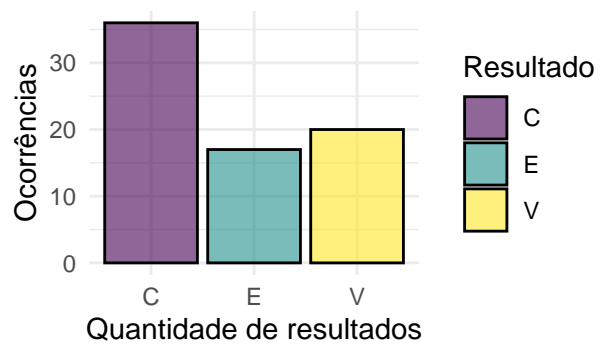
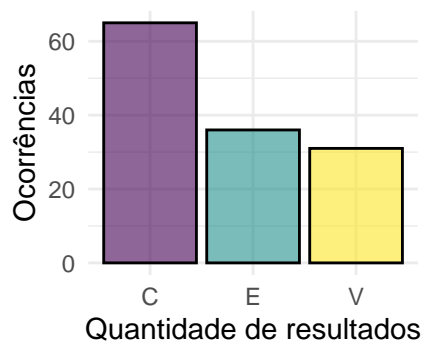


```
dist_res2016 <- grid.arrange(arrangeGrob(dist_calor_gelados2016, dist_calor_quentes2016,
dist_frio_gelados2016, dist_frio_quentes2016),
top = 'Distribuição dos resultados em 2016')
```

Distribuição dos resultados em 2016
e resultado nos estados frios Distribuição de 2016 no verão

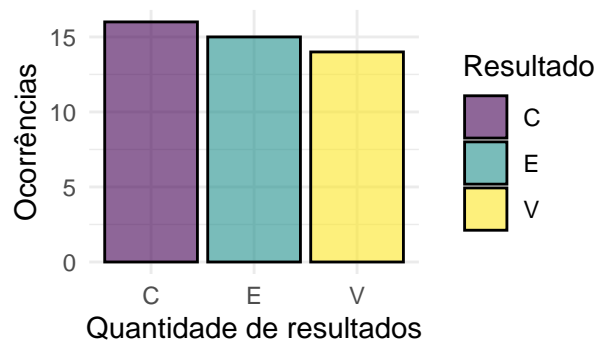
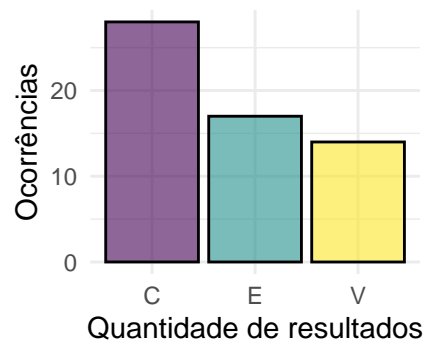


o resultado nos estados frios Distribuição de 2016 no inverno

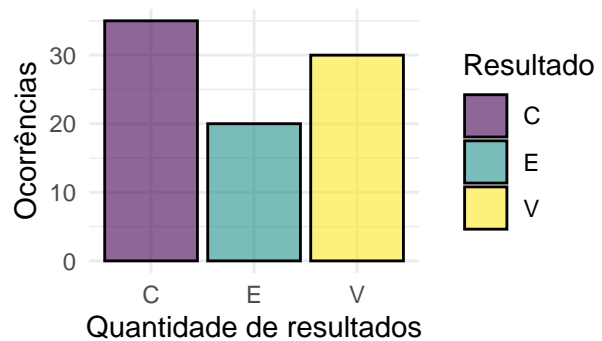
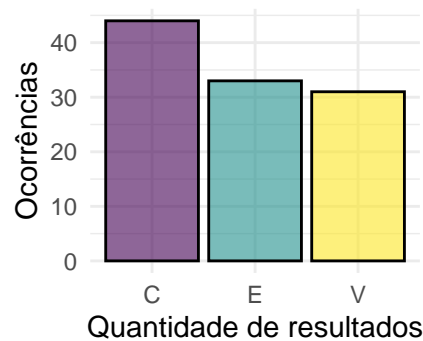


```
dist_res2017 <- grid.arrange(arrangeGrob(dist_calor_gelados2017, dist_calor_quentes2017,
  dist_frio_gelados2017, dist_frio_quentes2017),
  top = 'Distribuição dos resultados em 2017')
```

Distribuição dos resultados em 2017
 e resultado nos estados frios Distribuição de resultado nos estados quentes no verão



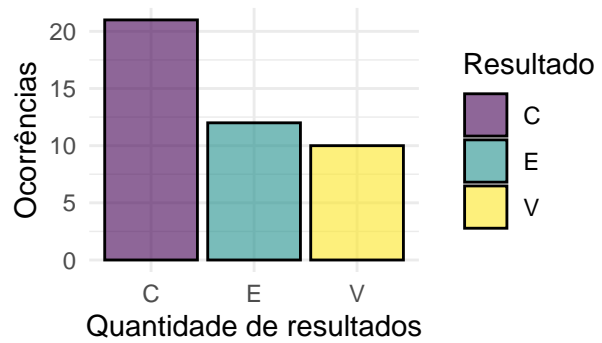
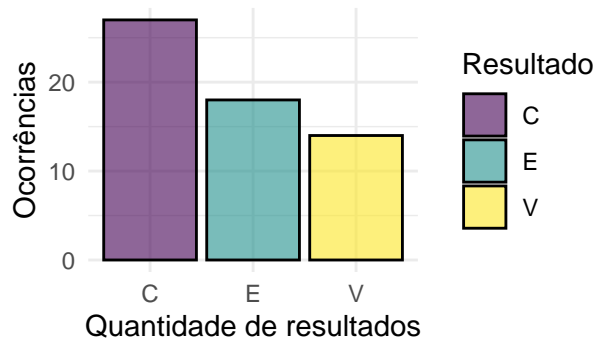
o resultado nos estados frios Distribuição de resultado nos estados quentes no inverno



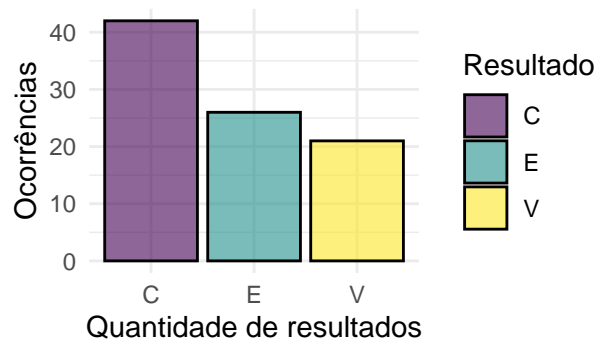
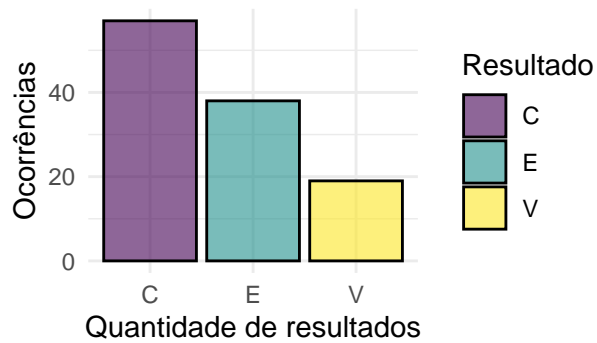
```
dist_res2018 <- grid.arrange(arrangeGrob(dist_calor_gelados2018, dist_calor_quentes2018,
  dist_frio_gelados2018, dist_frio_quentes2018),
  top = 'Distribuição dos resultados em 2018')
```

Distribuição dos resultados em 2018

e resultado nos estados frios no verão

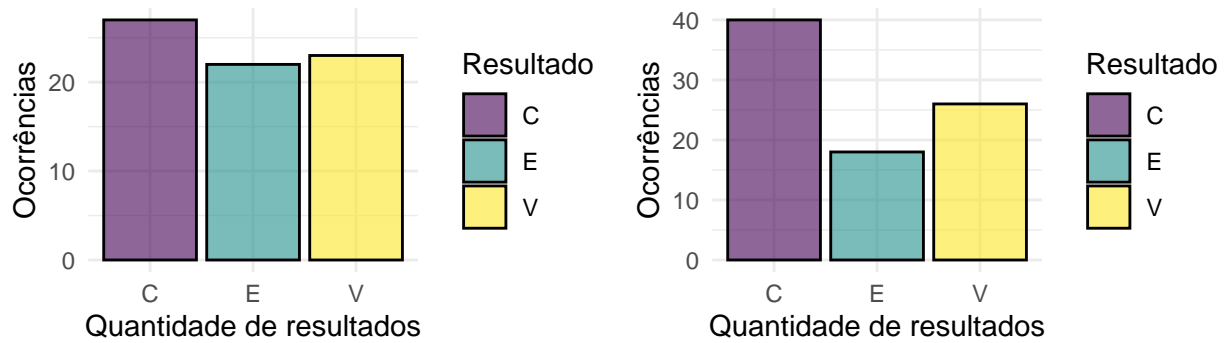


e resultado nos estados frios no inverno

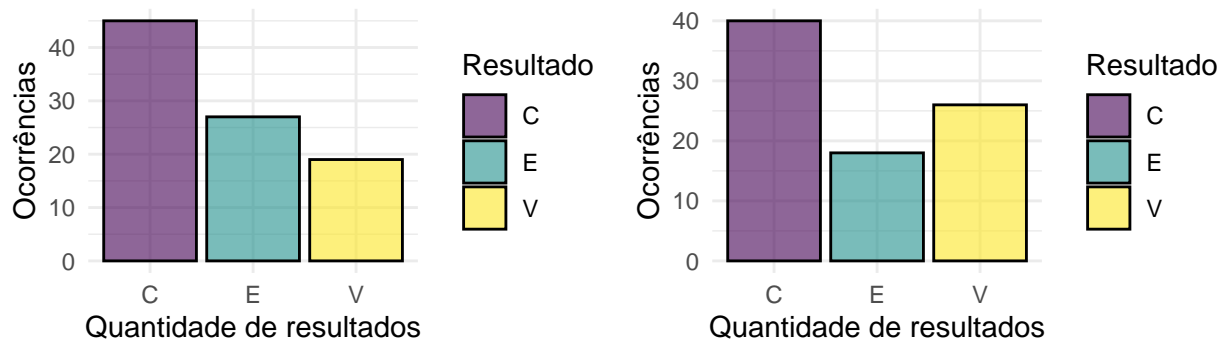


```
dist_res2019 <- grid.arrange(arrangeGrob(dist_calor_gelados2019, dist_frio_quentes2019,
  dist_frio_gelados2019, dist_frio_quentes2019),
  top = 'Distribuição dos resultados em 2019')
```


Distribuição dos resultados em 2019
e resultado nos estados Distribuição de 2019 e resultado nos estados quentes no inverno



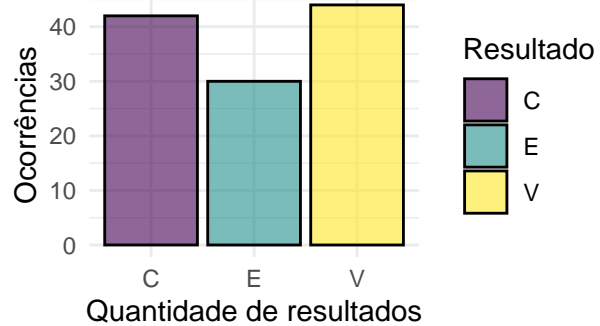
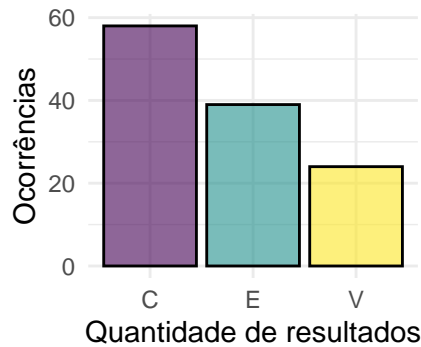
resultado nos estados Distribuição de 2019 e resultado nos estados quentes no inverno



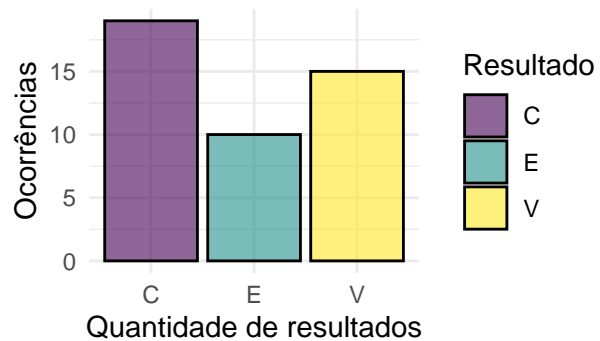
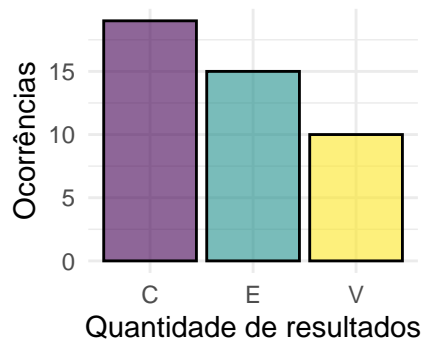
```
dist_res2020 <- grid.arrange(arrangeGrob(dist_calor_gelados2020, dist_calor_quentes2020,
dist_frio_gelados2020, dist_frio_quentes2020),
top = 'Distribuição dos resultados em 2020')
```

Distribuição dos resultados em 2020

e resultado nos estados frios Distribuição de resultado nos estados quentes no verão



e resultado nos estados frios Distribuição de resultado nos estados quentes no inverno



Plot da soma de gols por estação

```
somagols_inverno2014 <- medidas_sg_inverno_2014 %>% mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols no inverno por período por estado em 2014') +
  xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
  ggeasy::easy_center_title()

somagols_verao2014 <- medidas_sg_verao_2014 %>% mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols no inverno por período por estado em 2014') +
  xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
  ggeasy::easy_center_title()

somagols_inverno2015 <- medidas_sg_inverno_2015 %>% mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols no inverno por período por estado em 2015') +
  xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
  ggeasy::easy_center_title()

somagols_verao2015 <- medidas_sg_verao_2015 %>% mutate(estado = fct_reorder(estado, media)) %>%
```

```

group_by(estados) %>%
ggplot(aes(media, estado, colour = periodo)) + geom_point() +
scale_color_viridis(discrete = TRUE) +
ggtitle('Média de gols no inverno por período por estado em 2015') +
xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
ggeasy::easy_center_title()

somagols_inverno2016 <- medidas_sg_inverno_2016 %>% mutate(estado = fct_reorder(estado, media)) %>%
group_by(estados) %>%
ggplot(aes(media, estado, colour = periodo)) + geom_point() +
scale_color_viridis(discrete = TRUE) +
ggtitle('Média de gols no inverno por período por estado em 2016') +
xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
ggeasy::easy_center_title()

somagols_verao2016 <- medidas_sg_verao_2016 %>% mutate(estado = fct_reorder(estado, media)) %>%
group_by(estados) %>%
ggplot(aes(media, estado, colour = periodo)) + geom_point() +
scale_color_viridis(discrete = TRUE) +
ggtitle('Média de gols no inverno por período por estado em 2016') +
xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
ggeasy::easy_center_title()

somagols_inverno2017 <- medidas_sg_inverno_2017 %>% mutate(estado = fct_reorder(estado, media)) %>%
group_by(estados) %>%
ggplot(aes(media, estado, colour = periodo)) + geom_point() +
scale_color_viridis(discrete = TRUE) +
ggtitle('Média de gols no inverno por período por estado em 2017') +
xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
ggeasy::easy_center_title()

somagols_verao2017 <- medidas_sg_verao_2017 %>% mutate(estado = fct_reorder(estado, media)) %>%
group_by(estados) %>%
ggplot(aes(media, estado, colour = periodo)) + geom_point() +
scale_color_viridis(discrete = TRUE) +
ggtitle('Média de gols no inverno por período por estado em 2017') +
xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
ggeasy::easy_center_title()

somagols_inverno2018 <- medidas_sg_inverno_2018 %>% mutate(estado = fct_reorder(estado, media)) %>%
group_by(estados) %>%
ggplot(aes(media, estado, colour = periodo)) + geom_point() +
scale_color_viridis(discrete = TRUE) +
ggtitle('Média de gols no inverno por período por estado em 2018') +
xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
ggeasy::easy_center_title()

somagols_verao2018 <- medidas_sg_verao_2018 %>% mutate(estado = fct_reorder(estado, media)) %>%
group_by(estados) %>%
ggplot(aes(media, estado, colour = periodo)) + geom_point() +
scale_color_viridis(discrete = TRUE) +

```

```

ggtitle('Média de gols no inverno por período por estado em 2018') +
xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
ggeasy::easy_center_title()

somagols_inverno2019 <- medidas_sg_inverno_2019 %>% mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols no inverno por período por estado em 2014') +
  xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
  ggeasy::easy_center_title()

somagols_verao2019 <- medidas_sg_verao_2019 %>% mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols no inverno por período por estado em 2019') +
  xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
  ggeasy::easy_center_title()

somagols_inverno2020 <- medidas_sg_inverno_2020 %>% mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols no inverno por período por estado em 2020') +
  xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
  ggeasy::easy_center_title()

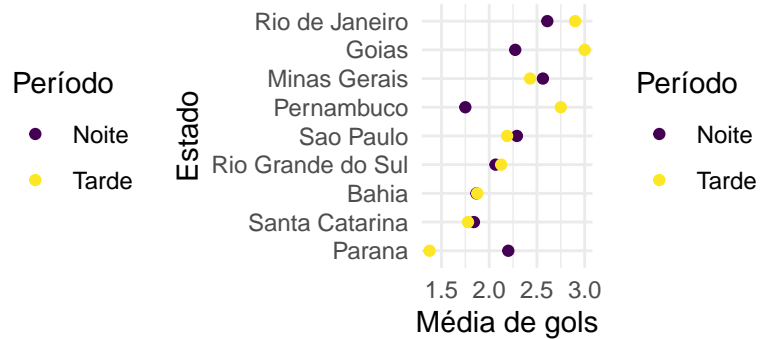
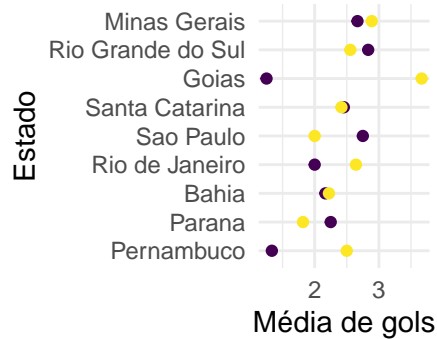
somagols_verao2020 <- medidas_sg_verao_2020 %>% mutate(estado = fct_reorder(estado, media)) %>%
  group_by(estado) %>%
  ggplot(aes(media, estado, colour = periodo)) + geom_point() +
  scale_color_viridis(discrete = TRUE) +
  ggtitle('Média de gols no inverno por período por estado em 2020') +
  xlab('Média de gols') + ylab('Estado') + labs(col = 'Período') +
  ggeasy::easy_center_title()

gols0 <- grid.arrange(arrangeGrob(somagols_verao2014, somagols_inverno2014,
                                  somagols_verao2015, somagols_inverno2015),
                      top = 'Distribuição da soma dos gols por estação (1)')

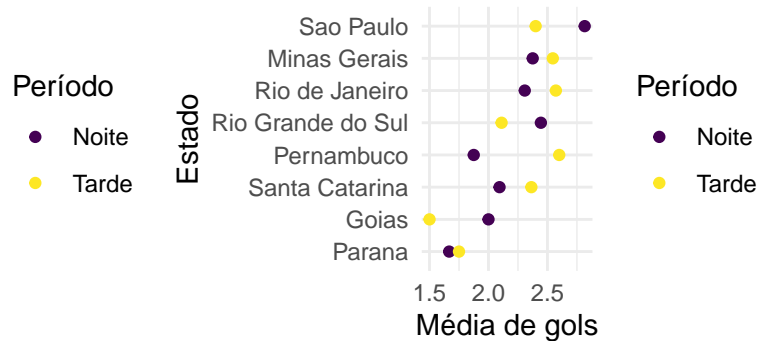
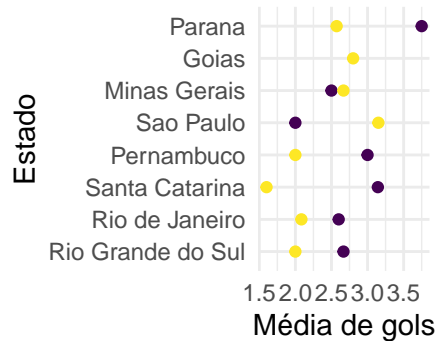
```

Distribuição da soma dos gols por estação (1)

de gols no inverno por período por estado de 2014



de gols no inverno por período por estado de 2015



```
gols1 <- grid.arrange(arrangeGrob(somagols_verao2016, somagols_inverno2016,
  somagols_verao2017, somagols_inverno2017),
  top = 'Distribuição da soma dos gols por estação (2)')
```

Distribuição da soma dos gols por estação (2)

de gols no inverno por período por estado de 2016



de gols no inverno por período por estado de 2017



```
gols2 <- grid.arrange(arrangeGrob(somagols_verao2018, somagols_inverno2018,
                                   somagols_verao2019, somagols_inverno2019),
                      top = 'Distribuição da soma dos gols por estação (3)')
```

Distribuição da soma dos gols por estação (3)

de gols no inverno por período por estado de 2016

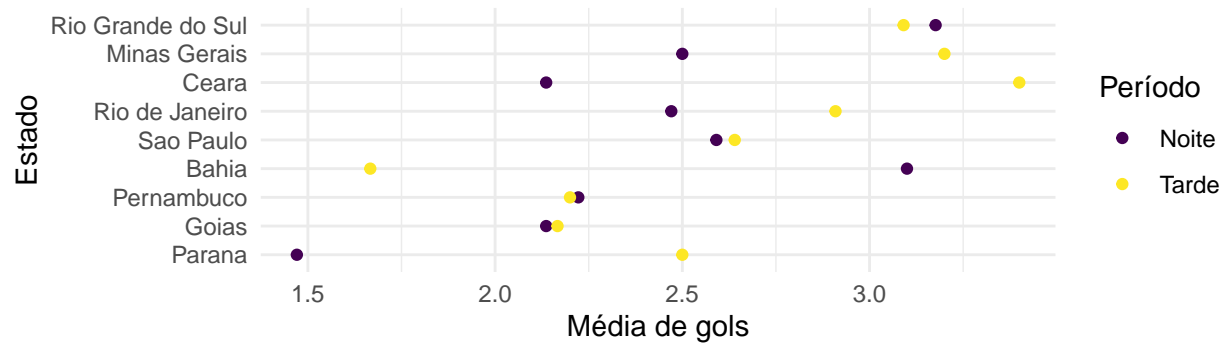


de gols no inverno por período por estado de 2019



```
gols3 <- grid.arrange(arrangeGrob(somagols_verao2020, somagols_inverno2020),
  top = 'Distribuição da soma dos gols por estação (4)')
```

Distribuição da soma dos gols por estação (4)
Média de gols no inverno por período por estado em 2020



Média de gols no inverno por período por estado em 2020

