

Relatório Velocidades

Arthur Hideio Noguti

2024-07-26

Contents

1	Introdução	5
1.1	Partes	5
1.2	Observações	5
2	Indicadores radares/frota	7
2.1	Bibliotecas	7
2.2	Pré processamento	7
2.3	Cálculo dos indicadores e <i>Data Frame</i> final	8
3	Indicador radar/Km	13
3.1	Bibliotecas	13
3.2	Pré processamento	13
3.3	Cálculo do Indicador	14
4	Indicador Mortalidade	15
4.1	Bibliotecas	15
4.2	Pré processamento	15
4.3	Cálculo do Indicador	16
5	Quantidade de radares por limite de velocidade	19
5.1	Bibliotecas	19
5.2	Construção do gráfico	19

6	Correlações	21
6.1	Bibliotecas	21
6.2	Importando os dados e manipulando estruturas	21
6.3	Gráficos	22
6.4	Correlograma entre indicadores radar/frota e indicador infrações	24

Chapter 1

Introdução

Esse documento possui a finalidade de transmitir meus pensamentos sobre o que foi feito em R no relatório de velocidades e também serve como uma forma de treinamento para minhas habilidades em *markdown*

1.1 Partes

Esse documento está dividido em 5 partes, sendo elas:

- Indicadores radares/frota;
- Indicador radares/Km;
- Indicador mortalidade;
- Quantidade de radares por limite de velocidade;
- Correlações entre indicadores.

1.2 Observações

Algumas bases de dados serão importadas pelo *GitHub* do ONSV, entretanto a base de dados que representa a quantidade de radares com suas características e da extensão das rodovias em cada unidade da federação serão importadas diretamente do computador, pois para a utilização dessas, você deverá baixar o arquivo excel e mudar o diretório do script quando tiver algumas função do tipo `read_csv(SEU DIRETÓRIO)`

Chapter 2

Indicadores radares/frota

2.1 Bibliotecas

Para realizar o cálculo dos indicadores foram utilizados as seguintes bibliotecas:

```
library(tidyverse)
library(readxl)
# devtools::install_github("jotasaraiva/fleetbr")
library(fleetbr)
```

Note que tem uma parte do *script* que está depois de um #, caso você não tenha instalado em sua máquina a base de dados sobre a frota de veículos no Brasil basta retirar o # e rodar o *script*. Eu não vou entrar em detalhes técnicos nesse documento, como por exemplo o que significa cada indicador e sua função, pois isto é apresentado no relatório em si.

2.2 Pré processamento

Além dessa base de dados, será necessário também a base sobre os radares, que será importado pelo arquivo .csv, como é demonstrado pelo *script* abaixo:

```
radares <- read_xlsx("/home/silvana/Downloads/INDICADORES_RADARES_VELOCIDADE_UF (indicador veloci
```

Note que o parâmetro `sheet = 2` indica que queremos apenas os dados da segunda aba da planilha identificada. Posterior a isso é filtrado ambas bases de dados para apenas as tuplas que contenham a classe indicada pelo estudo. Na base dos Radares é filtrado apenas os radares que sejam do tipo fixo, e na

base da frota, a frota relativa ao mês de dezembro do ano de 2023 e o modal seja o total de todos. Também é feito um `rename` da coluna `uf` da base de dados da frota, para que as duas bases possuam uma coluna que contenha os mesmos valores (no caso as siglas de cada unidade da federação) para depois utilizar as funções de `join(união)`.

```
radares <- radares %>%
  filter(TipoMedidor == "Fixo")

frota_2023_TOTAL <- fleetbr %>%
  filter(ano == "2023",
         mes == "12",
         modal == "TOTAL")

frota_2023_TOTAL_renomeada <- rename(frota_2023_TOTAL, SiglaUf = uf)
```

2.3 Cálculo dos indicadores e *Data Frame* final

2.3.1 I_1 ao I_9

Todos os indicadores possuem um *script* semelhante entre si, mudando apenas os filtros que levam cada um, sendo apenas o I_4 que possuem um código diferente dentre eles, segue abaixo o *script* para a construção de todos indicadores:

```
# i1 (aprovados+reparados)/frota ####

aprov_repari1 <- radares %>%
  filter(UltimoResultado != "Reprovado") %>%
  group_by(SiglaUf) %>%
  summarise(quantidade = n())

i1 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari1, by = "SiglaUf")
i1 <- i1 %>%
  mutate(i1 = (quantidade/frota)*10^4)
# i2 (urb aprov + urb repar / frota) ####

aprov_repari2 <- radares %>%
  filter(UltimoResultado != "Reprovado",
         Tipo == "Via Urbana") %>%
  group_by(SiglaUf) %>%
  summarise(quantidade = n())
```



```
i2 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari2, by = "SiglaUf")
i2 <- i2 %>%
  mutate(i2 = (quantidade/frota)*10^4)

# i3 (rod aprov + rod repar / frota) ####

aprov_repari3 <- radares %>%
  filter(UltimoResultado != "Reprovado",
         Tipo == "Rodovia") %>%
  group_by(SiglaUf) %>%
  summarise(quantidade = n())

i3 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari3, by = "SiglaUf")
i3 <- i3 %>%
  mutate(i3 = (quantidade/frota)*10^4)

# i4 (aprovados/(aprovados+reparados)) ####

aprov <- radares %>%
  filter(UltimoResultado == "Aprovado") %>%
  group_by(SiglaUf) %>%
  summarise(quantidade_aprov = n())

i4 <- inner_join(aprov, aprov_repari1, by = "SiglaUf")
i4 <- i4 %>%
  mutate(i4 = quantidade_aprov/quantidade)

# i5 (aprovados+reparados)/frota com v = 40 km/h ####
aprov_repari5 <- radares %>%
  filter(UltimoResultado != "Reprovado",
         VelocidadeNominal == "40") %>%
  group_by(SiglaUf) %>%
  summarise(quantidade = n())

i5 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari5, by = "SiglaUf")
i5 <- i5 %>%
  mutate(i5 = (quantidade/frota)*10^4)

# i6 (aprovados+reparados)/frota com v = 50 km/h ####
aprov_repari6 <- radares %>%
  filter(UltimoResultado != "Reprovado",
         VelocidadeNominal == "50") %>%
```

```
group_by(SiglaUf) %>%
summarise(quantidade = n())

i6 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari6, by = "SiglaUf")
i6 <- i6 %>%
  mutate(i6 = (quantidade/frota)*10^4)

# i7 (aprovados+reparados)/frota com v = 60 km/h ####
aprov_repari7 <- radares %>%
  filter(UltimoResultado != "Reprovado",
         VelocidadeNominal == "60") %>%
  group_by(SiglaUf) %>%
  summarise(quantidade = n())

i7 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari7, by = "SiglaUf")
i7 <- i7 %>%
  mutate(i7 = (quantidade/frota)*10^4)

# i8 (aprovados+reparados)/frota com v = 70 km/h ####
aprov_repari8 <- radares %>%
  filter(UltimoResultado != "Reprovado",
         VelocidadeNominal == "70") %>%
  group_by(SiglaUf) %>%
  summarise(quantidade = n())

i8 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari8, by = "SiglaUf")
i8 <- i8 %>%
  mutate(i8 = (quantidade/frota)*10^4)

# i9 (aprovados+reparados)/frota com v >= 80 km/h ####
aprov_repari9 <- radares %>%
  filter(UltimoResultado != "Reprovado",
         VelocidadeNominal >= 80) %>%
  group_by(SiglaUf) %>%
  summarise(quantidade = n())

i9 <- right_join(frota_2023_TOTAL_renomeada, aprov_repari9, by = "SiglaUf")
i9 <- i9 %>%
  mutate(i9 = (quantidade/frota)*10^4)
```

2.3.2 Geração do *Data Frame*

Para fazer o *Data Frame* com todos os indicadores juntos, foi primeiramente feito um seleção de cada indicador, de forma a selecionar apenas as colunas que representavam a sigla da unidade da federação e o valor do indicador.

```
I1 <- i1[,c(1,7)]
I2 <- i2[,c(1,7)]
I3 <- i3[,c(1,7)]
I4 <- i4[,c(1,4)]
I5 <- i5[,c(1,7)]
I6 <- i6[,c(1,7)]
I7 <- i7[,c(1,7)]
I8 <- i8[,c(1,7)]
I9 <- i9[,c(1,7)]
```

Com isso foi feito uma lista de todos esses indicadores para que então seja utilizado a função `reduce()` de forma a não haver nove colunas com as siglas das UF e para que cada tupla represente apenas uma UF e todos os seus respectivos indicadores. E feito uma alteração para que onde tivesse NA seja trocado pelo valor 0.

```
ind_rad <- list(I1,I2,I3,I4,I5,I6,I7,I8,I9)
Indicador_radares <- reduce(ind_rad, full_join, by = "SiglaUf")
Indicador_radares[is.na(Indicador_radares)] <- 0
Indicador_radares
```

```
## # A tibble: 27 x 10
##   SiglaUf    i1    i2    i3    i4    i5    i6    i7    i8    i9
##   <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 AC      0.428 0.371 0.0571 1      0.228 0.171 0      0.0285 0
## 2 AL      1.08 0      1.08 0.941 0.0274 0.740 0.292 0      0.0183
## 3 AM      0.115 0      0.115 0.846 0.00885 0      0.0619 0      0.0442
## 4 AP      1.44 1.32 0.124 0.943 0.247 0      1.20 0      0
## 5 BA      1.73 0.773 0.953 0.963 0.494 0.393 0.580 0.0879 0.141
## 6 CE      3.75 1.39 2.36 0.863 0.567 0.682 2.30 0      0.200
## 7 DF      7.50 1.58 5.92 0.965 1.55 1.16 3.62 0.202 0.922
## 8 ES      1.69 0.110 1.58 0.947 0.246 0.0509 1.23 0.00424 0.157
## 9 GO      4.60 2.00 2.60 0.911 1.11 0.681 1.97 0.108 0.679
## 10 MA     0.844 0.502 0.342 0.839 0.159 0.206 0.309 0      0.0188
## # i 17 more rows
```


Chapter 3

Indicador radar/Km

3.1 Bibliotecas

Para o cálculo do indicador radar/Km foi utilizado apenas as seguintes bibliotecas:

```
library(tidyverse)
library(readxl)
```

3.2 Pré processamento

Neste indicador, as duas bases de dados foram importadas do computador, sendo uma base a dos radares (a mesma base utilizada no capítulo 2) e a outra a base de dados relativa a extensão das rodovias em cada unidade da federação. Abaixo é mostrado o *script* utilizado para importar ambas:

```
radares <- read_xlsx("/home/silvana/Downloads/INDICADORES_RADARES_VELOCIDADE_UF (indicador veloci
km <- read_xlsx('/home/silvana/Downloads/Rodovias Federais.xlsx')
```

Posterior a isso é feito um filtro na base de radares, para selecionarmos apenas os radares do tipo fixo, que estão em rodovias, e que sejam “Aprovados” ou “Reparados”.

```
radares <- radares %>%
  filter(TipoMedidor == "Fixo")
radares_rodovia <- radares %>%
  filter(Tipo == "Rodovia" &
         UltimoResultado != "Reprovado")
```

E então é utilizado a função `group_by()` seguida de um `summarise()` para que seja agrupado pela sigla da UF e somado a quantidade de radares.

```
radares_rodovia_quantidade <- radares_rodovia %>%
  group_by(SiglaUf) %>%
  summarise(Quantidade = n())
```

3.3 Cálculo do Indicador

Como o arquivo excel da extensão da vias já estava separado por UF e somado em cada, não foi necessário realizar esse passo para ele. Como os dois possuíam a mesma quantidade de linhas e uma coluna com valores iguais, foi feito um *full join* (união total da duas tabelas) utilizam tal coluna idêntica como chave primária.

```
Indicador_radar_km <- full_join(km, radares_rodovia_quantidade, by = c("sg_uf" = "SiglaUf"))
```

Esse *data frame* criado ainda não possui o valor do indicador, mas já possui todas as informações necessárias para realizar o cálculo do mesmo, então é utilizado a função `mutate()` para criar uma nova coluna com o valor do indicador para cada UF. E foi também renomeado as colunas para algo mais “interpretativo”.

```
Indicador_radar_km <- Indicador_radar_km %>%
  mutate(indicador = (Quantidade / `soma vl_extensa`) * 100) # Quantidade de radares a cada UF

Indicador_radar_km <- rename(Indicador_radar_km, "Extensão_das_rodovias" = `soma vl_extensa`,
  Quantidade_de_radares = Quantidade)
```

Para este caso foi exportado em formato excel, para simplesmente copiar e colar os dados no documento *Word*, e então para isso foi devido a utilização de outra biblioteca.

```
library(writexl)
writexl::write_xlsx(Indicador_radar_km, path = "/home/silvana/Downloads/Indicador_radar.xlsx")
```

Chapter 4

Indicador Mortalidade

Esse indicador foi feito para tentar encontrar alguma relação entre a mortalidade e a quantidade de radares.

4.1 Bibliotecas

Nesse indicador foi utilizado duas bases de dados disponíveis pelo Observatório, a de frota e a base de dados do DataSus, o *script* a seguir contém as bibliotecas utilizadas e a instalação dessas bases.

```
library(devtools)
library(tidyverse)
# devtools::install_github("pabsantos/roadtrafficdeaths")
# devtools::install_github("jotasaraiva/fleetbr")
library(roadtrafficdeaths)
library(fleetbr)
```

4.2 Pré processamento

Na base do DataSus foi feito um filtro para o ano de 2022 e depois agrupado pelo nome da UF e somado a quantidade de óbitos. Então foi retirado os casos com NA e criado uma nova coluna com a siglas das UF, pois a base de dados da frota só contém as Siglas, então para unir as duas foi devido tal construção. Essa nova coluna teve de ser feita manualmente, ou seja, foi feita olhando o nome de cada UF e colocando sua sigla de forma organizada.

```

mortes_2022 <- rtdeaths %>%
  filter(ano_ocorrencia == "2022")

mortes_2022_porUF <- mortes_2022 %>%
  group_by(nome_uf_ocor) %>%
  summarise(Quantidade_de_obitos = n())

mortes_2022_porUF <- na.omit(mortes_2022_porUF)

mortes_2022_porUF <- mortes_2022_porUF %>%
  mutate(uf = c("AC", "AL", "AP", "AM", "BA",
                "CE", "DF", "ES", "GO", "MA",
                "MT", "MS", "MG", "PR", "PB",
                "PA", "PE", "PI", "RN", "RS",
                "RJ", "RO", "RR", "SC", "SE",
                "SP", "TO"))

```

Já na base da frota, foi apenas filtrado para o mês de dezembro do ano de 2022 com o modal “TOTAL”.

```

frota_2022_TOTAL <- fleetbr %>%
  filter(ano == "2022",
         mes == "12",
         modal == "TOTAL")

```

4.3 Cálculo do Indicador

Foi feito a união de ambas as bases e gerado uma nova coluna que representaria o valor do indicador.

```

Indicador_mortal <- (full_join(mortes_2022_porUF, frota_2022_TOTAL, by = "uf" ))

Indicador_mortal <- Indicador_mortal %>%
  mutate(Indicador = (Quantidade_de_obitos/frota)*(10^4))

Indicador_mortal [,c(3,8)]

```

```

## # A tibble: 27 x 2
##   uf      Indicador
##   <chr>    <dbl>
## 1 AC        3.08
## 2 AL        5.58
## 3 AP        4.51

```



```
## 4 AM      4.09
## 5 BA      5.06
## 6 CE      3.87
## 7 DF      1.71
## 8 ES      3.66
## 9 GO      3.81
## 10 MA     6.59
## # i 17 more rows
```


Chapter 5

Quantidade de radares por limite de velocidade

Esse *script* foi criado apenas para gerar um gráfico incluído no relatório.

5.1 Bibliotecas

Foram utilizadas as seguintes bibliotecas:

```
library('readxl')
library('tidyverse')
library('onsvplot') # usar o tema de gráfico do onsv
```

5.2 Construção do gráfico

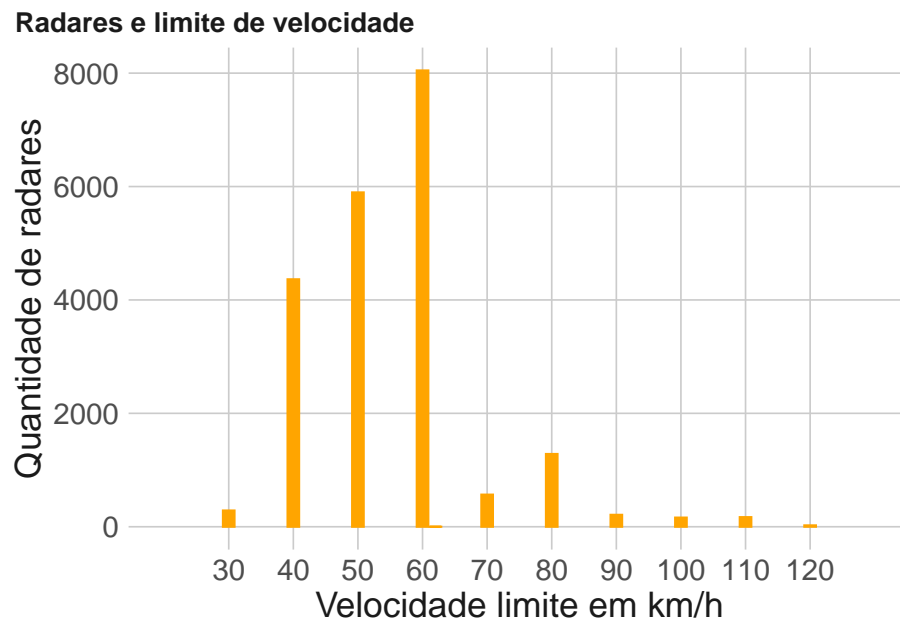
Primeiramente foi importado os dados sobre os radares.

```
radar_velo <- read_xlsx("/home/silvana/Downloads/INDICADORES_RADARES_VELOCIDADE_UF (indicador vel
```

E então foi feito o gráfico de barras para representar a quantidade de radares por cada limite de velocidade.

```
filter(radar_velo, VelocidadeNominal != 0 & TipoMedidor == "Fixo")%>%
  ggplot(aes(x = VelocidadeNominal))+
  geom_bar(color = "orange", fill = "orange")+
  scale_x_continuous(limits = c(20,130),
```

```
breaks = seq(30,120,10))+  
labs(x = "Velocidade limite em km/h",  
     y = "Quantidade de radares",  
     title = "Radares e limite de velocidade")+  
theme_onsv()+  
theme(axis.title = element_text(size = 18))+  
theme(axis.text.x = element_text(size = 15),  
      axis.text.y = element_text(size = 15))
```



Chapter 6

Correlações

6.1 Bibliotecas

Foram utilizadas diversas bibliotecas nessa seção, sendo elas:

```
library('readxl') #ler arquivo xlsx excel
library('tidyverse') # plotar gráficos e utilizar o pipe
library('patchwork') # plotar gráficos juntos
library('onsvplot') # usar o tema de gráfico do onsu
library('corrplot') # fazer o correlograma
# devtools::install_github("pabsantos/roadtrafficdeaths")
# devtools::install_github("jotasaraiva/fleetbr")
library('roadtrafficdeaths') # dados sobre mortalidade no trânsito
library('fleetbr') # dados sobre a frota de veículos
```

6.2 Importando os dados e manipulando estruturas

Foi importado os dados relativos as infrações e retirado a linha “BR”, além de renomear a coluna relativo ao indicador de infrações. Com isso foi feito uma união com o *Data Frame* criado no capítulo 2 e trocado onde havia NA por 0.

```
infrações <- read_xlsx("/home/silvana/Downloads/INDICADORES_RADARES_VELOCIDADE_UF (indicador velo
infrações <- infrações[-7,] #excluindo a linha "BR"
infrações <- rename(infrações, infrações = `infrações/10000 veículos`)

Indicador_radares <- rename(Indicador_radares, UF = SiglaUf) #Esse data frame vem do script "Ind
```

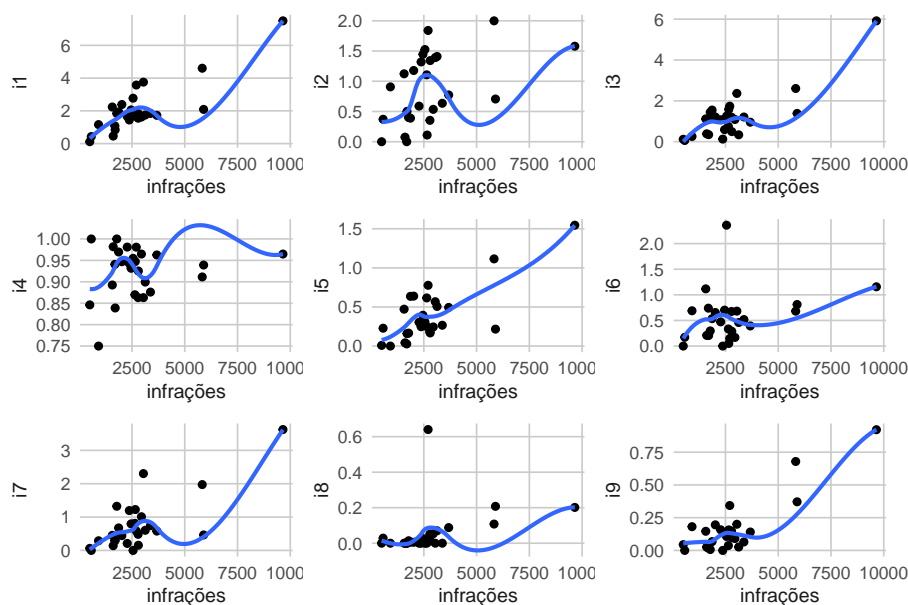
```
rad_inf <- full_join(infrações[,c(1,4)], Indicador_radares, by = "UF")

rad_inf <- rad_inf %>%
  mutate_all(replace_na, 0)
```

6.3 Gráficos

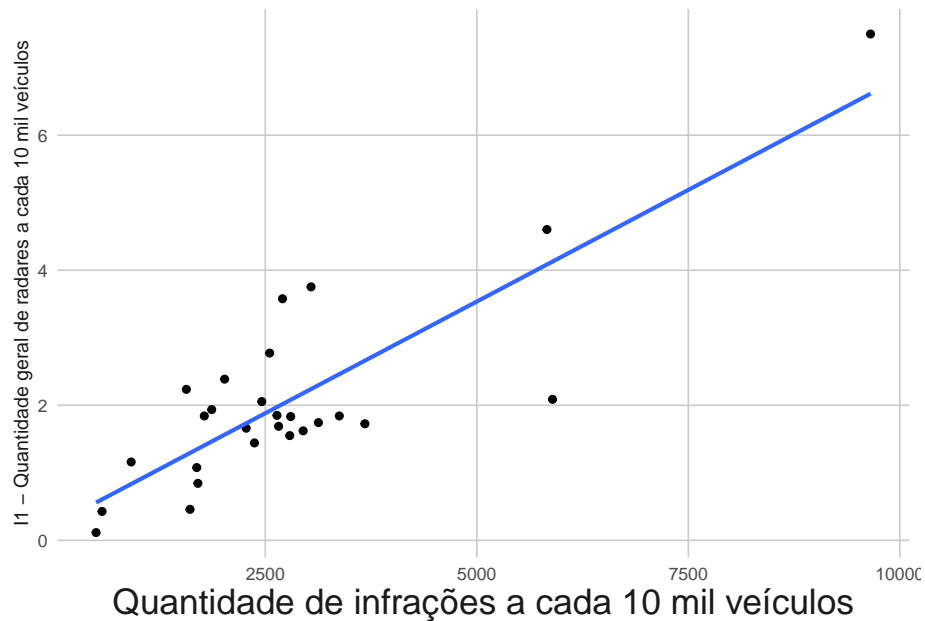
Foi feito então um gráfico geral para avaliar a correlação visual entre os indicadores radar/frota e o indicador de infrações.

```
ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i1))+
  geom_smooth(mapping = aes(x= infrações , y = i1), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i2))+
  geom_smooth(mapping = aes(x= infrações , y = i2), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i3))+
  geom_smooth(mapping = aes(x= infrações , y = i3), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i4))+
  geom_smooth(mapping = aes(x= infrações , y = i4), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i5))+
  geom_smooth(mapping = aes(x= infrações , y = i5), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i6))+
  geom_smooth(mapping = aes(x= infrações , y = i6), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i7))+
  geom_smooth(mapping = aes(x= infrações , y = i7), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i8))+
  geom_smooth(mapping = aes(x= infrações , y = i8), se = FALSE)+theme_onsv()+
  ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i9))+
  geom_smooth(mapping = aes(x= infrações , y = i9), se = FALSE)+theme_onsv()
```



Mas por fim foi adicionado no relatório apenas o gráfico de I_1 vs infrações.

```
ggplot(data = rad_inf)+
  geom_point(mapping = aes(x= infrações , y = i1))+
  geom_smooth(mapping = aes(x= infrações , y = i1), se = FALSE, method = "lm")+
  theme_onsv()+
  labs(x = "Quantidade de infrações a cada 10 mil veículos", y="I1 - Quantidade geral de radares")
  theme(axis.title.x = element_text(size =18),axis.title.y = element_text(size =10))
```



Nesse arquivo *bookdown* eu tive que diminuir o tamanho da legenda do eixo “y” para 10 pois não estava “cabendo” no gráfico, mas no relatório foi utilizado a fonte 18 para o eixo “y” (o mesmo tamanho da fonte do eixo “x”)

6.4 Correlograma entre indicadores radar/frota e indicador infrações

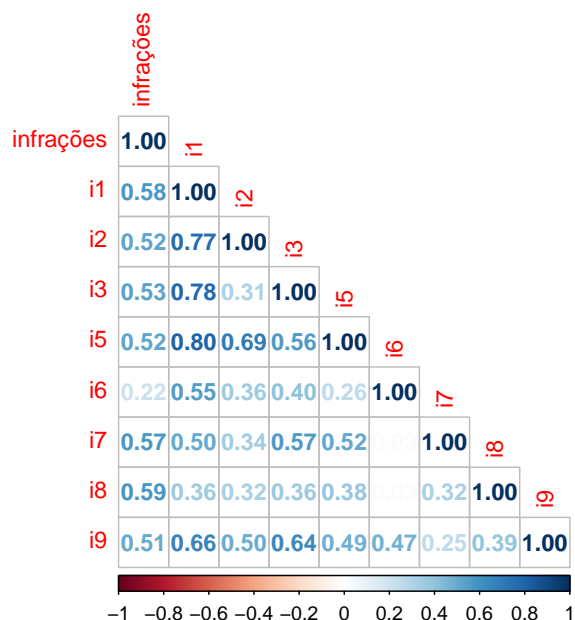
Foi feito um `cor.test()` em todos os casos para verificar se são estatisticamente correlacionados:

```
cor.test(rad_inf$infrações, rad_inf$i1, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i2, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i3, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i4, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i5, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i6, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i7, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i8, method = 'spearman')
cor.test(rad_inf$infrações, rad_inf$i9, method = 'spearman')
```

Com isso ficou nítido que o I_4 não possui correlação alguma com as infrações e o I_6 também poderia ser excluído se considerássemos um $p\text{-valor} < 0.05$. Após esse teste, foi feito o gráfico de correlação geral:

6.4. CORRELOGRAMA ENTRE INDICADORES RADAR/FROTA E INDICADOR INFRAÇÕES²⁵

```
graf_corr <- cor(rad_inf[,c(-1,-6)], method = "spearman")
corrplot(graf_corr, type = "lower", method = "number")
```



Correlação entre infrações e mortalidade

Para esse correlação foi feito um ajuste na base de dados do DataSus e teve de ser criado um `tibble` (um *Data Frame* evoluído) sobre as regiões das unidades da federação, classificando cada unidade da federação na sua respectiva região. Fazendo então a união de três bases, a base de mortalidade, a base de infrações e a base das regiões.

```
rtdeaths_2022 <- filter(rtdeaths, rtdeaths$ano_ocorrendia == 2022)
rtdeaths_2022 <- rtdeaths_2022 %>%
  group_by(rtdeaths_2022$nome_uf_ocor) %>%
  summarise(
    quantidade = n()
  )
rtdeaths_2022 <- rtdeaths_2022[-28,]
rtdeaths_2022 <- mutate(rtdeaths_2022, UF = c("AC", "AL", "AP", "AM", "BA", "CE", "DF", "ES", "GO", "MA", "MS", "MG", "PR", "PB", "PE", "PI", "RN", "RS", "RJ", "RO", "RR", "SC", "SE", "SP", "TO"))
rtdeaths_2022 <- rename(rtdeaths_2022, Estado = `rtdeaths_2022$nome_uf_ocor`)

fleet_2022 <- filter(fleetbr, mes == 12, ano == 2022, modal == "TOTAL")
```

```
rtdeaths_fleet <- inner_join(rtdeaths_2022, fleet_2022, by = c("UF" = "uf"))
rtdeaths_fleet <- mutate(rtdeaths_fleet, A_cada_10_mil = (quantidade*10^4)/frota)

rad_inf2 <- subset(rad_inf, select = infrações)
cor.test(rad_inf2$infrações, rtdeaths_fleet$A_cada_10_mil, method = "spearman")
```

```
##
## Spearman's rank correlation rho
##
## data: rad_inf2$infrações and rtdeaths_fleet$A_cada_10_mil
## S = 3108, p-value = 0.7992
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.05128205
```

```
cor.test(rad_inf$i1, rtdeaths_fleet$A_cada_10_mil, method = "spearman")
```

```
##
## Spearman's rank correlation rho
##
## data: rad_inf$i1 and rtdeaths_fleet$A_cada_10_mil
## S = 3796, p-value = 0.4274
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## -0.1587302
```

```
mort_ind <- rtdeaths_fleet %>%
  inner_join(rad_inf, by = "UF")

regioes <- tibble(UF = c("RS", "SC", "PR", "SP", "MG", "RJ", "ES", "MS", "MT", "GO", "DF", "BA", "PE",
                        "TO", "PA", "AP", "RR", "RO", "AM", "AC"),
                 Regiao = c("Sul", "Sul", "Sul", "Sudeste", "Sudeste", "Sudeste", "Sudeste",
                            "Centro Oeste", "Centro Oeste", "Nordeste", "Nordeste", "Nordeste",
                            "Nordeste", "Nordeste", "Nordeste", "Norte", "Norte", "Norte", "Norte", "Norte"))

mort_ind <- mort_ind %>%
  inner_join(regioes, by = "UF")
```

Note também que foi feito um teste de correlação, e ele já indicou que não há tal correlação, mas mesmo assim podemos tirar *insights* visuais do gráfico feito abaixo:

6.4. CORRELOGRAMA ENTRE INDICADORES RADAR/FROTA E INDICADOR INFRAÇÕES²⁷

```
ggplot(data = mort_ind)+
  geom_point(mapping = aes(x = i1, y = A_cada_10_mil, color = Regiao))+
  geom_label(aes(x = i1, y = A_cada_10_mil, label = UF), size = 3.5, hjust = -0.3)+
  theme_onsv()+
  labs(x = "I1 - Quantidade de radares a cada 10 mil veículos", y = "Mortes a cada 10 mil veículos")+
  theme(axis.title.x = element_text(size = 18),
        axis.title.y = element_text(size = 18),
        legend.text = element_text(size = 12))
```

