Artigo ANPET - Velocidades Inseguras

1 Importação e correção dos dados naturalísticos

Os dados foram salvos previamente no formato `.parquet` e podem ser carregados com auxílio da biblioteca `arrow`.

```
ndsbr_path <- here::here("data/ndsbr_sample.parquet")
ndsbr_sample <- arrow::open_dataset(ndsbr_path)
```

Em seguida, seleciona-se as variáveis de interesse para o estudo de velocidade e transforma-se a base para `data.frame`.

```
selected_cols <- c("driver", "trip", "long", "lat", "date", "time", "spd_kmh")
speed_sample <- as.data.frame(ndsbr_sample[selected_cols], row.names = NULL)
knitr::kable(head(speed_sample))
```

driver	trip	long	lat	date	time	spd_kmh
A	2	-49.2341	-25.43476	2019-08-24	13H 7M 25S	0.1609
A	2	-49.2341	-25.43476	2019-08-24	13H 7M 26S	0.0000
A	2	-49.2341	-25.43476	2019-08-24	13H 7M 27S	0.1609
A	2	-49.2341	-25.43476	2019-08-24	13H 7M 28S	0.1609
A	2	-49.2341	-25.43476	2019-08-24	13H 7M 29S	0.1609
A	2	-49.2341	-25.43476	2019-08-24	13H 7M 30S	0.0000

Próximo passo é filtrar algumas inconsistências na amostra original, como pontos que não tem coordenadas e não tem dados de velocidades.

```r
filtered_speed_sample <- subset(
  speed_sample,
  !is.na(speed_sample$spd_kmh) &
  !is.na(speed_sample$long) &
  !is.na(speed_sample$lat)
)
```

Por fim, com auxílio do `sf`, é feita a conversão dos dados para pontos geográficos.

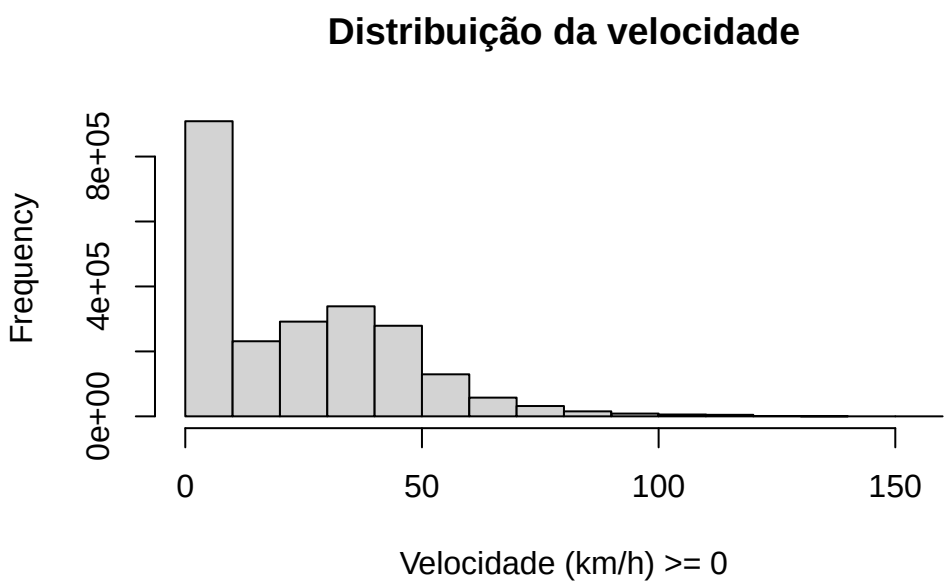
```r
speed_points <- sf::st_as_sf(
  filtered_speed_sample,
  coords = c("long", "lat"),
  crs = "4674"
)

knitr::kable(head(speed_points))
```

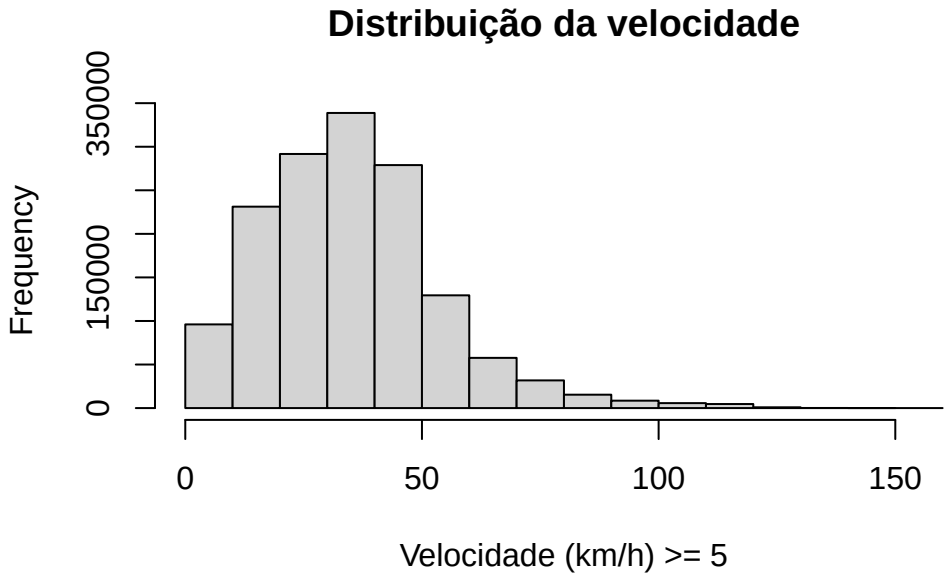
driver	trip	date	time	spd_kmh	geometry
A	2	2019-08-24	13H 7M 25S	0.1609	POINT (-49.2341 -25.43476)
A	2	2019-08-24	13H 7M 26S	0.0000	POINT (-49.2341 -25.43476)
A	2	2019-08-24	13H 7M 27S	0.1609	POINT (-49.2341 -25.43476)
A	2	2019-08-24	13H 7M 28S	0.1609	POINT (-49.2341 -25.43476)
A	2	2019-08-24	13H 7M 29S	0.1609	POINT (-49.2341 -25.43476)
A	2	2019-08-24	13H 7M 30S	0.0000	POINT (-49.2341 -25.43476)

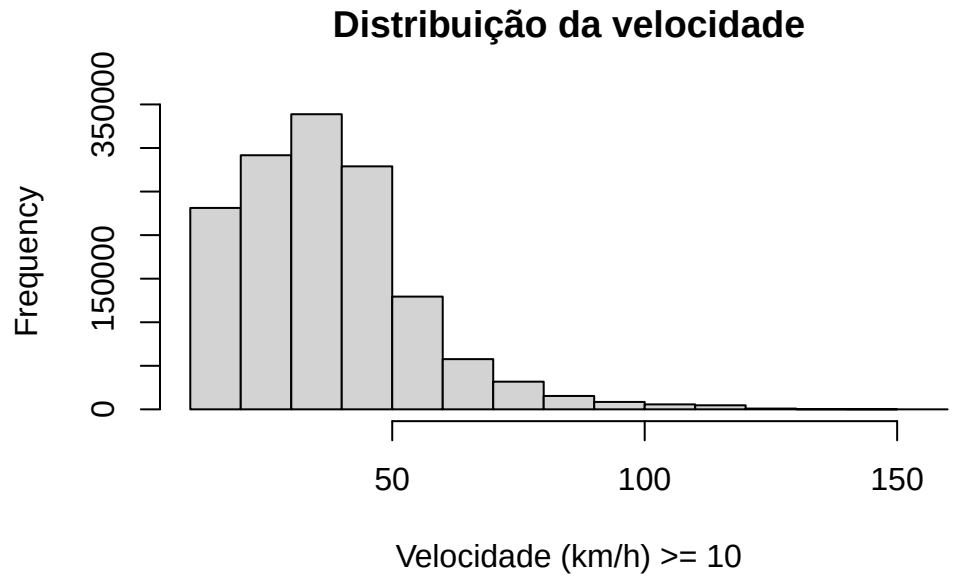
2 Entendendo a velocidade

Antes de prosseguir com as análises espaciais, faz-se uma análise de distribuição dos dados de velocidade. Apenas com o vetor das velocidades, é elaborado um histograma com a amostra completa e também com partes da amostra acima de 5 km/h e acima de 10 km/h.

```r
sapply(
  c(0, 5, 10),
  \(x) hist(
    speed_points$spd_kmh[speed_points$spd_kmh >= x],
    xlab = paste0("Velocidade (km/h) >= ", x),
    main = "Distribuição da velocidade",
    breaks = 20
  )
)
```


Distribuição da velocidade
Frequency
0e+00
4e+05
8e+05
0
50
100
150
Velocidade (km/h) >= 0


Distribuição da velocidade
Frequency
0
150000
350000
0
50
100
150
Velocidade (km/h) >= 5




```
         [,1]
breaks   numeric,17
counts   integer,16
density  numeric,16
mids     numeric,16
xname    "speed_points$spd_kmh[speed_points$spd_kmh >= x]"
equidist TRUE
         [,2]
breaks   numeric,17
counts   integer,16
density  numeric,16
mids     numeric,16
xname    "speed_points$spd_kmh[speed_points$spd_kmh >= x]"
equidist TRUE
         [,3]
breaks   integer,16
counts   integer,15
density  numeric,15
mids     numeric,15
xname    "speed_points$spd_kmh[speed_points$spd_kmh >= x]"
equidist TRUE
```

A partir dos histogramas, percebe-se a necessidade de remover os pontos com velocidades

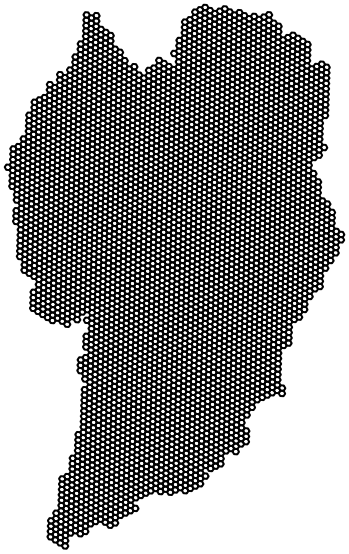
iguais a zero, para desconsiderar o tempo parado no trânsito. Dentre as duas opções de filtro (acima de 5 ou acima de 10), as análises foram baseadas nas velocidades iguais ou acima de 5.

```
speed_points_filtered <- speed_points[speed_points$spd_kmh > 5, ]
```

3 Grid H3

A análise espacial é realizada com base no grid H3, desenvolvido pelo Uber. O projeto "Acesso a Oportunidades", do IPEA, disponibiliza esse grid pronto para Curitiba, através do pacote `aopdata`.

```
grid_cwb <- aopdata::read_grid(city = "cur")
plot(grid_cwb["id_hex"], col = NA, main = NA)
```



Com spatial join é possível associar os pontos do nds-br com as informações do grid.

```
speed_points_grid <- sf::st_join(
  sf::st_set_crs(speed_points_filtered, 4326),
  grid_cwb
)
```

```
knitr::kable(head(speed_points_grid))
```

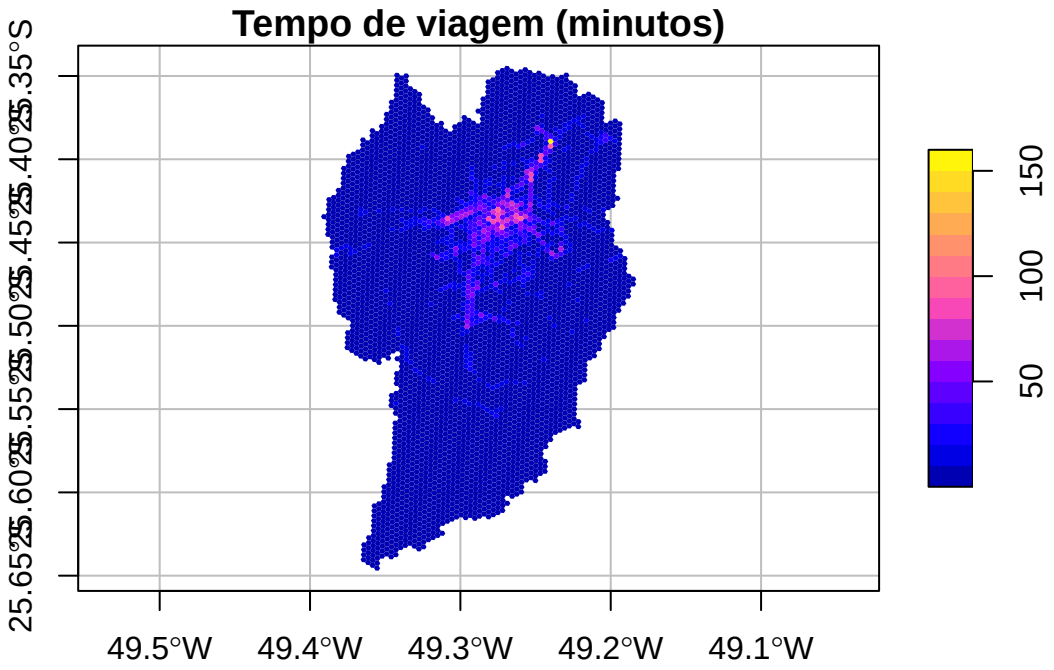
	driver	trip	date	time	spd_kmh	id_hex	abbrev_name	name_mun	code_mun	geometry
73	A	2	2019-08-24	13H 8M 37S	6.9202	89a8313443bffff	Curitiba	4106902	POINT	(-49.2341 -25.43477)
74	A	2	2019-08-24	13H 8M 38S	5.9546	89a8313443bffff	Curitiba	4106902	POINT	(-49.23411 -25.43478)
75	A	2	2019-08-24	13H 8M 39S	6.1155	89a8313443bffff	Curitiba	4106902	POINT	(-49.23413 -25.43478)
76	A	2	2019-08-24	13H 8M 40S	7.0811	89a8313443bffff	Curitiba	4106902	POINT	(-49.23414 -25.43479)
77	A	2	2019-08-24	13H 8M 41S	7.5639	89a8313443bffff	Curitiba	4106902	POINT	(-49.23415 -25.4348)
78	A	2	2019-08-24	13H 8M 42S	7.7248	89a8313443bffff	Curitiba	4106902	POINT	(-49.23415 -25.43482)

4 Análise da amostra do nds-br

Aqui o primeiro passo é entender a distribuição espacial da amostra, ou seja, quanto tempo de viagem cada célula do grid tem de amostra.

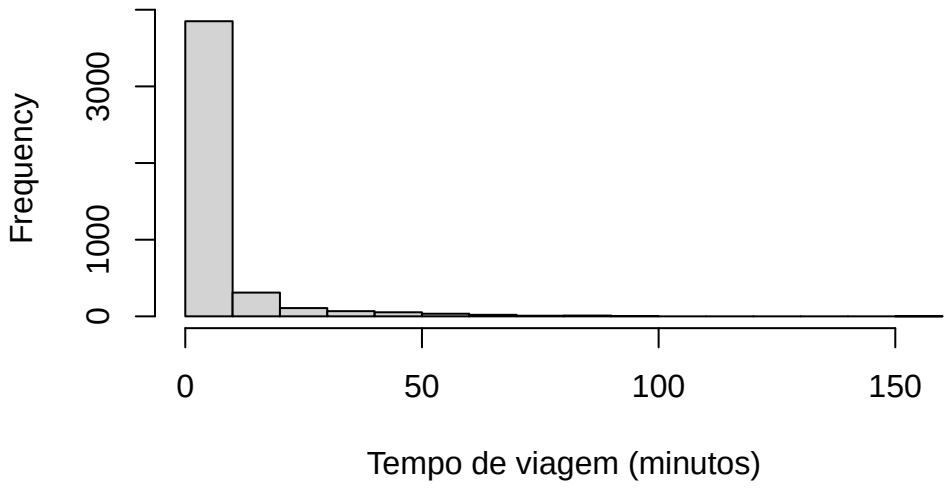
```
travel_time_minutes <- table(speed_points_grid$id_hex) / 60
travel_df <- as.data.frame(travel_time_minutes)
names(travel_df) <- c("id_hex", "travel_time_minutes")
grid_travel <- merge(grid_cwb, travel_df, by = "id_hex", all.x = TRUE)
grid_travel$travel_time_minutes[is.na(grid_travel$travel_time_minutes)] <- 0
plot(
  grid_travel["travel_time_minutes"],
  main = "Tempo de viagem (minutos)",
  graticule = TRUE,
  axes = TRUE,
  border = NA,
```

```
  breaks = c(seq(0, 160, 10))
)
```



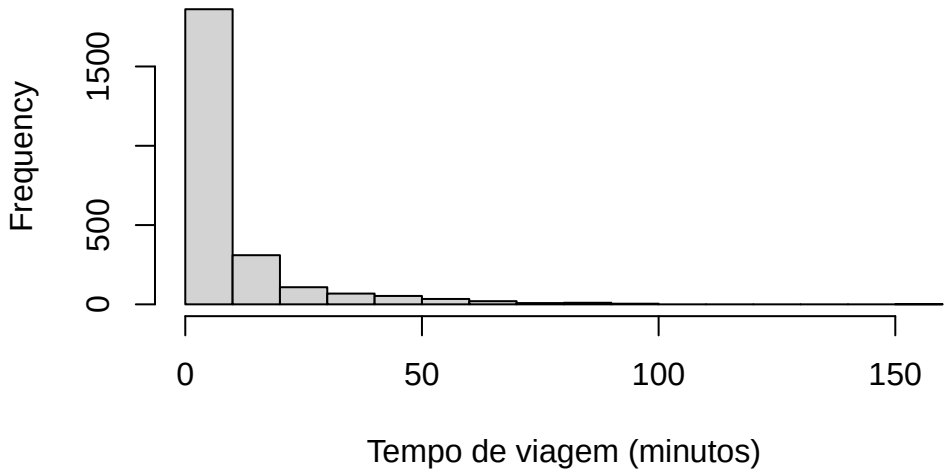
Com o histrograma fica mais fácil de observar o comportamento da distribuição

```
hist(
  grid_travel$travel_time_minutes,
  xlab = "Tempo de viagem (minutos)",
  main = ""
)
```



Grande parte da amostra se encontra na faixa entre 0 e 5 minutos de tempo de viagem. Assim, decidiu-se fazer outro histrograma só com os tempos acima de 0

```
hist(
  grid_travel$travel_time_minutes[grid_travel$travel_time_minutes > 0],
  xlab = "Tempo de viagem (minutos)",
  main = ""
)
```



Ainda assim, boa parte das células da amostra apresentam tempos de viagem até 5 minutos. A seguir está o cálculo exato de quantas celulas contêm / não contêm tempos de viagem.

```
total_celulas <- nrow(grid_cwb)
celulas_travel <- sum(grid_travel$travel_time_minutes > 0)
celulas_notravel <- total_celulas - celulas_travel
```

O território de Curitiba possui um total de 4466 células do grid H3. Deste total, 2477 possuem amostra do nds-br passando em sua área e 1989 sem amostra.

5 Análise espacial da velocidade

Com os dados de velocidade associados aos grids, é possível fazer uma análise exploratória da velocidade insegura no território de Curitiba com base nos seguintes indicadores:

- (V1) Velocidade média
- (V2) Velocidade mediana
- (V3) Desvio padrão da velocidade
- (V4) 85º quantil da velocidade

```r
results_v1_v3 <- sapply(
  list(mean, median, sd),
  \(x) tapply(speed_points_grid$spd_kmh, speed_points_grid$id_hex, x)
)

v4 <- tapply(
  speed_points_grid$spd_kmh,
  speed_points_grid$id_hex,
  quantile,
  p = 0.85
)

results_df <-
  data.frame(
    results_v1_v3,
    V4 = v4,
    id_hex = row.names(results_v1_v3),
    row.names = NULL
  )

names(results_df) <- c("V1", "V2", "V3", "V4", "id_hex")
knitr::kable(head(results_df))
```

V1	V2	V3	V4	id_hex
61.55811	60.804	2.8466185	64.0692	89a804cb003ffff
70.61400	70.704	0.8349363	71.2260	89a804cb00bffff
67.69029	68.184	1.0581205	68.4756	89a804cb00fffff
62.48160	62.334	0.5136783	62.9928	89a804cb013ffff
65.65886	65.520	1.7849886	67.0212	89a804cb017ffff
79.99425	79.794	4.0769942	84.1500	89a804cb063ffff

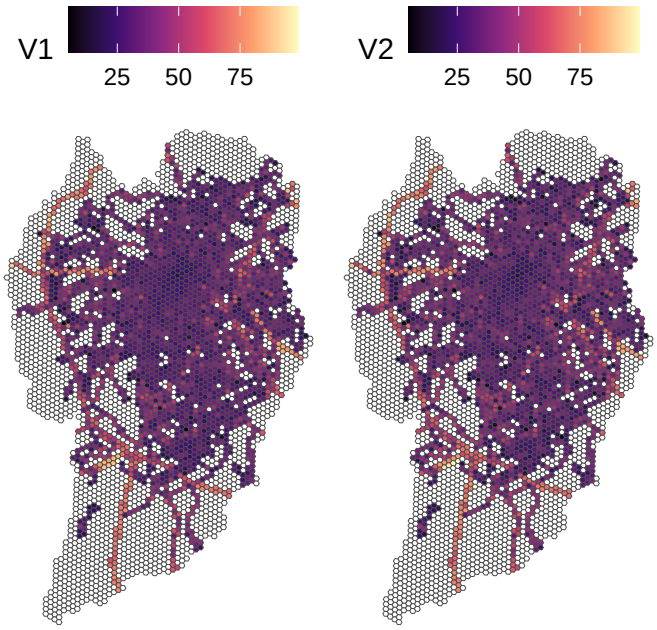
Com os indicadores resultantes, é possível uni-las com o grid através da chave única.

```r
grid_cwb_indicadores <-
  merge(grid_cwb, results_df, by = "id_hex", all.x = TRUE)

knitr::kable(head(grid_cwb_indicadores))
```

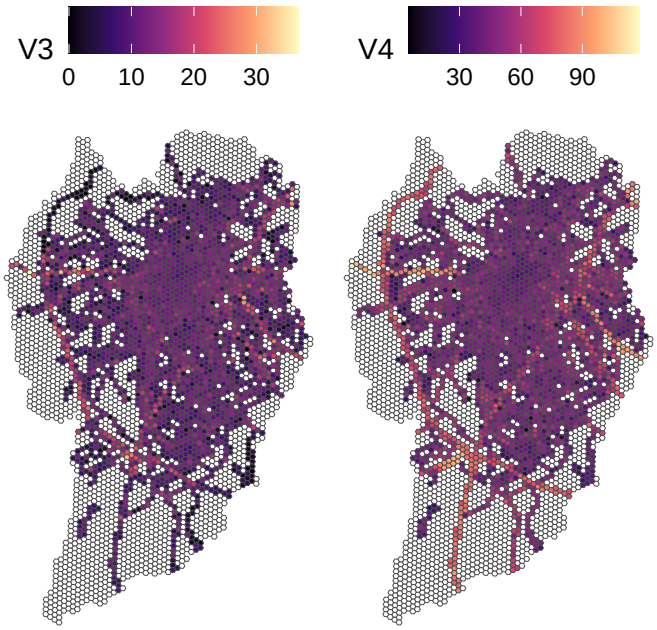
id_hex	abbrev_muni	name_muni	code_muni	V1	V2	V3	V4	geometry
89a804ca643ffff	cur	Curitiba	4106902	NA	NA	NA	NA	POLYGON ((-49.37018 -25.398...
89a804ca64bffff	cur	Curitiba	4106902	NA	NA	NA	NA	POLYGON ((-49.36709 -25.397...
89a804ca653ffff	cur	Curitiba	4106902	NA	NA	NA	NA	POLYGON ((-49.37028 -25.401...
89a804ca657ffff	cur	Curitiba	4106902	NA	NA	NA	NA	POLYGON ((-49.37326 -25.399...
89a804ca65bffff	cur	Curitiba	4106902	NA	NA	NA	NA	POLYGON ((-49.36719 -25.400...
89a804ca6c3ffff	cur	Curitiba	4106902	NA	NA	NA	NA	POLYGON ((-49.37347 -25.405...

6 Mapeando os resultados

```
library(ggplot2)
library(patchwork)

plot_indicadores <- function(ind) {
  ggplot() +
    geom_sf(
      data = grid_cwb_indicadores,
      aes(fill = {{ ind }}),
      color = "grey30",
      lwd = 0.1
    ) +
    scale_fill_viridis_c(na.value = "white", option = "A", direction = 1) +
    theme_void() +
    theme(legend.position = "top")
}

plot_indicadores(V1) + plot_indicadores(V2)
```



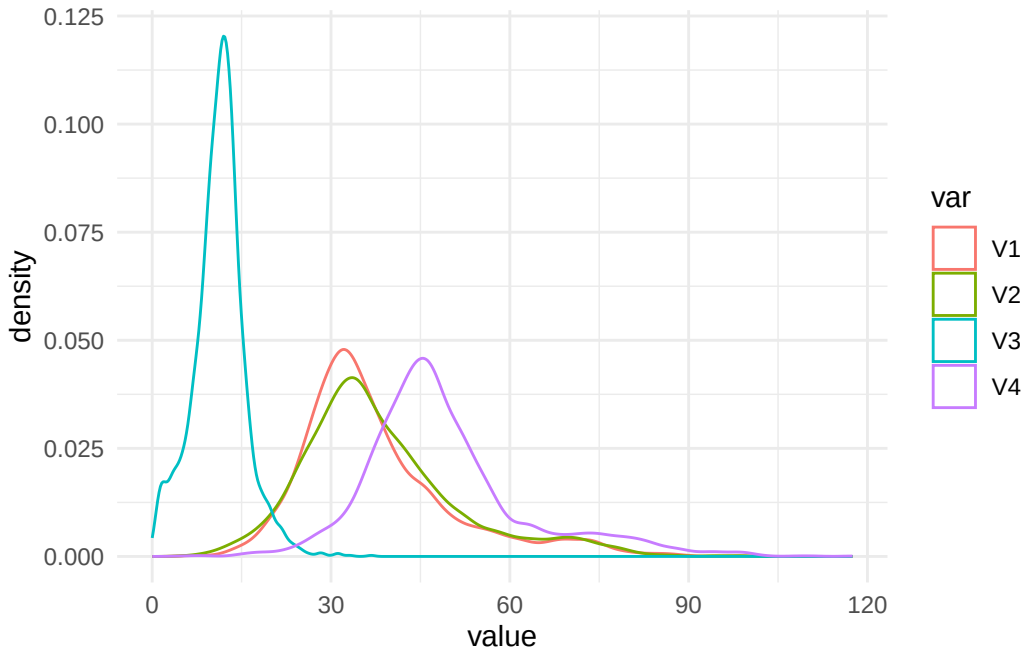

```
plot_indicadores(V3) + plot_indicadores(V4)
```



7 Distribuição dos indicadores

Plotando as densidades para avaliar a distribuição dos indicadores

```
table_vars_long <- grid_cwb_indicadores |>
  sf::st_drop_geometry() |>
  dplyr::select(V1:V4) |>
  tidyr::pivot_longer(
    dplyr::everything(),
    values_to = "value",
    names_to = "var"
  ) |>
  tidyr::drop_na()

ggplot(table_vars_long, aes(x = value, color = var)) +
  geom_density() +
  theme_minimal()
```



A média (V1) e a mediana (V2) possuem uma distribuição similar. Desvio padrão tem um valor máximo que se aproxima em 25 km/h, e o 85º quantil possui uma distribuição de formato similar a V1 e V2, porém, com valores maiores.

8 Correlação entre indicadores de velocidade

Aqui é analisada a correlação entre os indicadores, para detectar colinearidade dentro da amostra

```
var_cols <- c("V1", "V2", "V3", "V4")
vars_df <- sf::st_drop_geometry(grid_cwb_indicadores[var_cols])
vars_df <- subset(vars_df, !is.na(V1) & !is.na(V3))

cor_spearman <- psych::corr.test(vars_df)
cor_spearman$r
```

```
           V1         V2         V3        V4
V1 1.00000000 0.98135852 0.04214203 0.9276570
V2 0.98135852 1.00000000 0.05885401 0.9008611
V3 0.04214203 0.05885401 1.00000000 0.3783408
V4 0.92765697 0.90086111 0.37834076 1.0000000
```

```
cor_spearman$p
```

```
           V1          V2           V3           V4
V1 0.00000000 0.000000000 3.615859e-02 0.000000e+00
V2 0.00000000 0.000000000 6.839868e-03 0.000000e+00
V3 0.03615859 0.003419934 0.000000e+00 1.763388e-84
V4 0.00000000 0.000000000 5.877960e-85 0.000000e+00
```

V1, V2 e V4 possuem uma correlação alta (> 0.9). Assim, não faz sentido considerar esses três indicadores para analisar velocidade insegura. V3 possui uma correlação um pouco maior com V4 (> 0.4). Na análise da velocidade insegura, talvez faça mais sentido em utilizar apenas um indicador de frequência (V1, V2 ou V4) e um indicador de dispersão (V3). Todos os resultados são estatisticamente significativos no nível de 95% da confiabilidade (p-valor $< 0{,}05$).

9 Autocorrelação global e local

Dois passos importantes para analisar o comportamento espacial da amostra consiste em analisar a presença de autocorrelação global e local

9.1 Moran's I

Para a analise global aplica-se o método do I de Moran, considerando a configuração "queen" para analisar os lags espaciais, em que cada vizinho possui um peso igual (`style = "W"`).

```
library(spdep)
```

```
Loading required package: spData
```

```
To access larger datasets in this package, install the spDataLarge
package with: `install.packages('spDataLarge',
repos='https://nowosad.github.io/drat/', type='source')`
```

```
Loading required package: sf
```

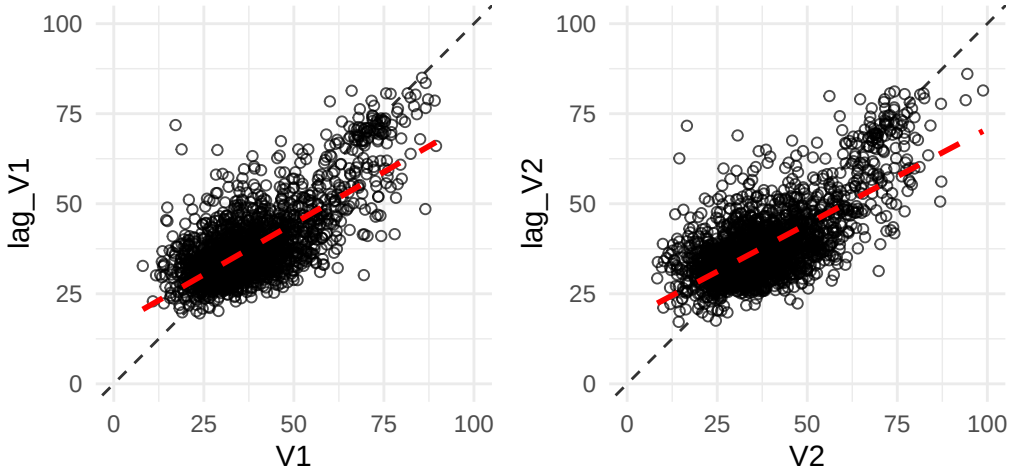
```
Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
## Removendo os grids sem dados de indicadores (NA)

grid_ind <-
  subset(
    grid_cwb_indicadores,
    !is.na(V1) & !is.na(V2) & !is.na(V3) & !is.na(V4)
  )

## Processeguindo com o método

nb <- poly2nb(grid_ind, queen = TRUE)
lw <- nb2listw(nb, style = "W", zero.policy = TRUE)
```

Plotando o lag espacial para os 4 indicadores.

```
ind_lags <- sapply(
  list(grid_ind$V1, grid_ind$V2, grid_ind$V3, grid_ind$V4),
  lag.listw,
  x = lw
)

ind_lags_df <- as.data.frame(ind_lags)
names(ind_lags_df) <- c("lag_V1", "lag_V2", "lag_V3", "lag_V4")
```

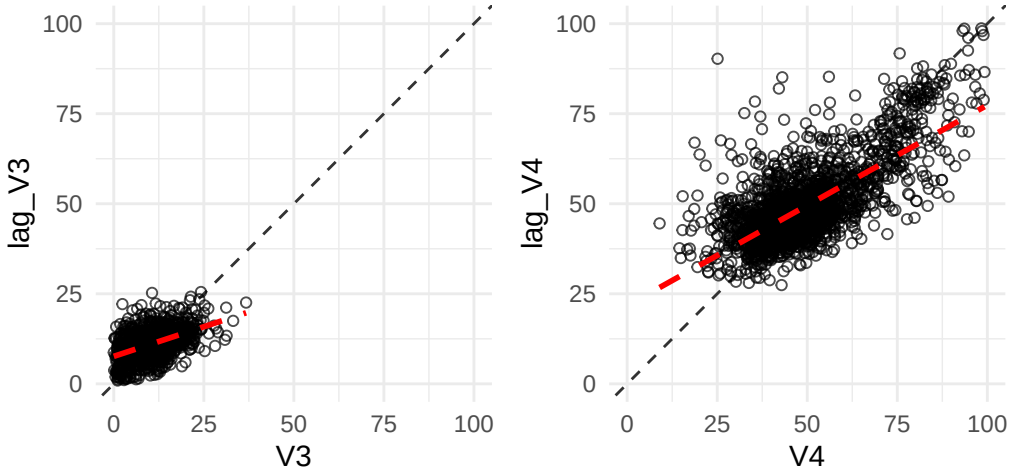
```r
df_lags <- cbind(ind_lags_df, sf::st_drop_geometry(grid_ind))

plot_lag <- function(ind, lag_ind) {
  # lmax <- ifelse({{ ind }} == V3, 40, 100)

  ggplot(df_lags, aes(x = {{ ind }}, y = {{ lag_ind }})) +
    geom_abline(
      intercept = 0,
      slope = 1,
      color = "grey20",
      lty = "dashed",
      lwd = 0.5
    ) +
    geom_point(pch = 21, alpha = 0.7) +
    geom_smooth(
      method = "lm",
      se = FALSE,
      color = "red",
      lty = "dashed",
      lwd = 1
    ) +
    coord_equal() +
    scale_x_continuous(limits = c(0, 100)) +
    scale_y_continuous(limits = c(0, 100)) +
    theme_minimal()
}

plot_lag(V1, lag_V1) + plot_lag(V2, lag_V2)
```




```
plot_lag(V3, lag_V3) + plot_lag(V4, lag_V4)
```

```
Warning: Removed 8 rows containing non-finite values (`stat_smooth()`).
```

```
Warning: Removed 8 rows containing missing values (`geom_point()`).
```



A linha pontilhada em vermelho indica a reta da regressão linear com base nos valores reais e lags espaciais desses valores. Uma análise visual previa já indica uma presença que uma autocorrelação global em um comportamento clusterizado. Porém, para ter certeza, é necessário calcular o resultado. O calculo do I de Moran foi realizado com base no método de Monte Carlo, com mil simulações.

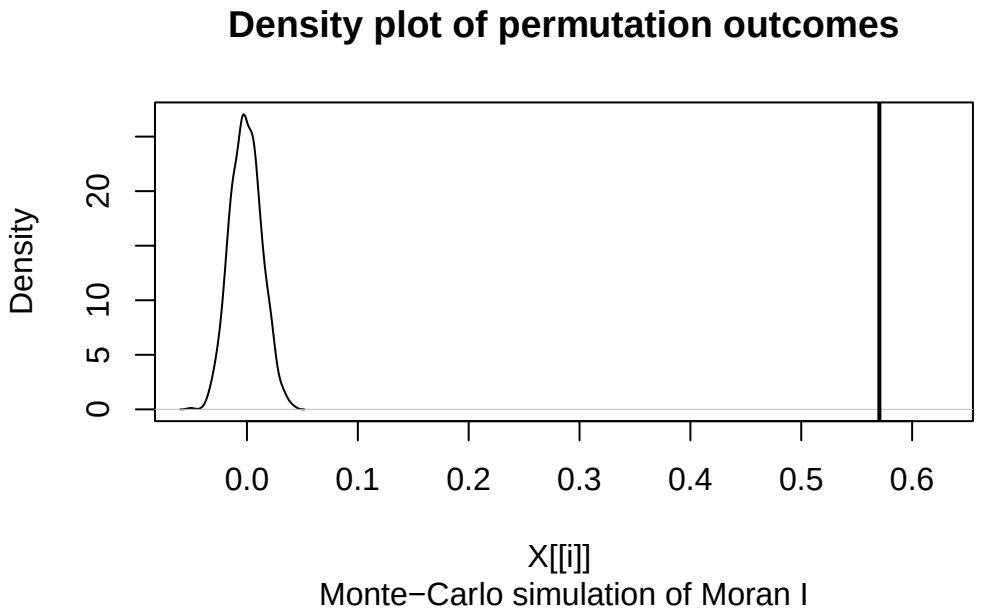
```
set.seed(42)

moran_results <- lapply(
  list(grid_ind$V1, grid_ind$V2, grid_ind$V3, grid_ind$V4),
  moran.mc,
  listw = lw,
  alternative = "greater",
  nsim = 999
)

extract_moran_results <- function(results) {
  moran_stat <- vector()
  moran_pvalue <- vector()
  for (i in 1:length(results)) {
    moran_stat[i] <- moran_results[[i]]$statistic
    moran_pvalue[i] <- moran_results[[i]]$p.value
  }
```

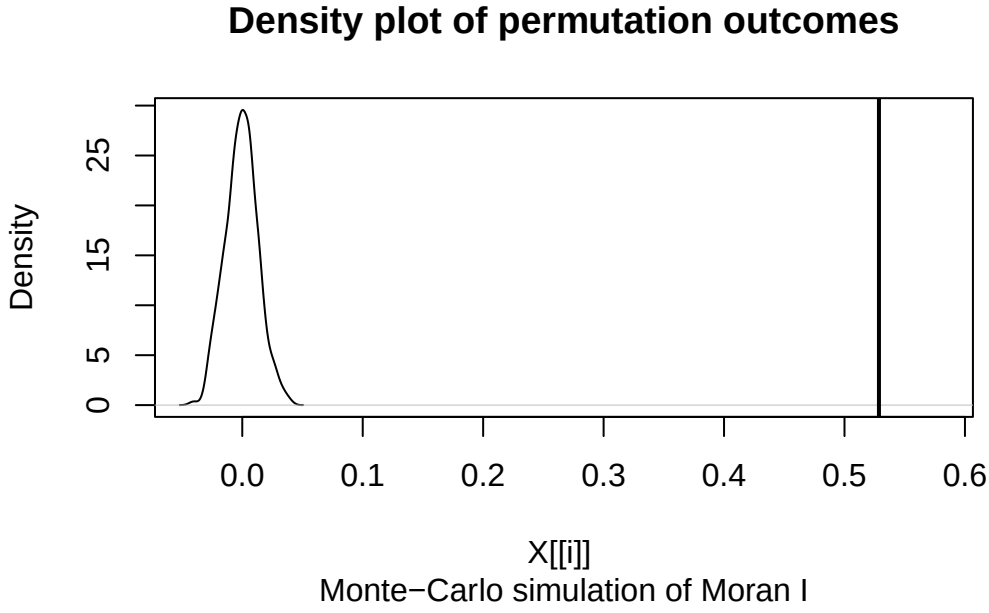
```
  df <- data.frame(
    indicadores = c("V1", "V2", "V3", "V4"),
    moran_stat = moran_stat,
    moran_pvalue = moran_pvalue
  )
  return(df)
}

tbl_moran <- extract_moran_results(moran_results)
knitr::kable(tbl_moran)
```

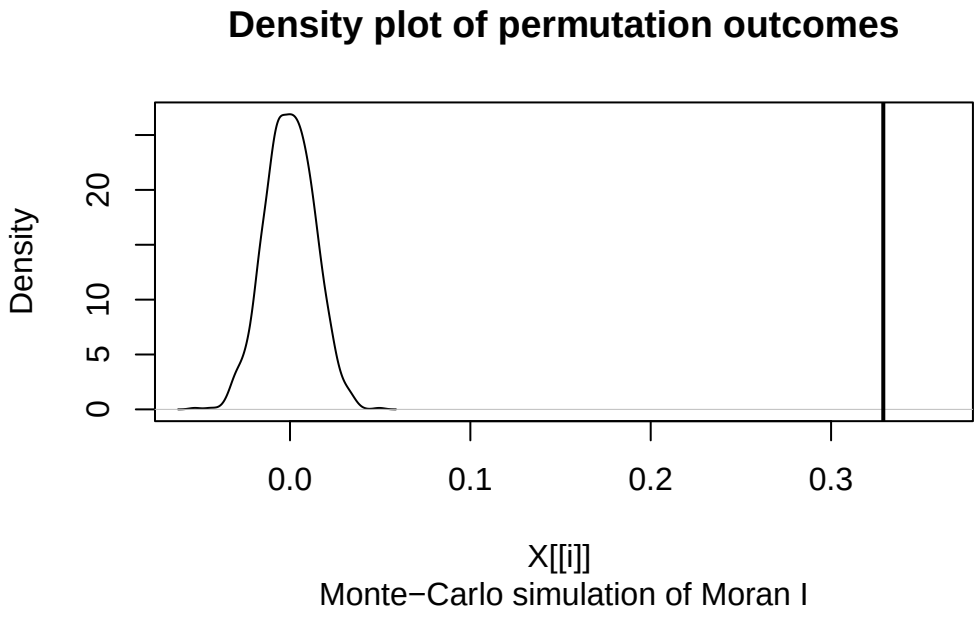
indicadores	moran_stat	moran_pvalue
V1	0.5704824	0.001
V2	0.5286144	0.001
V3	0.3289622	0.001
V4	0.5694284	0.001

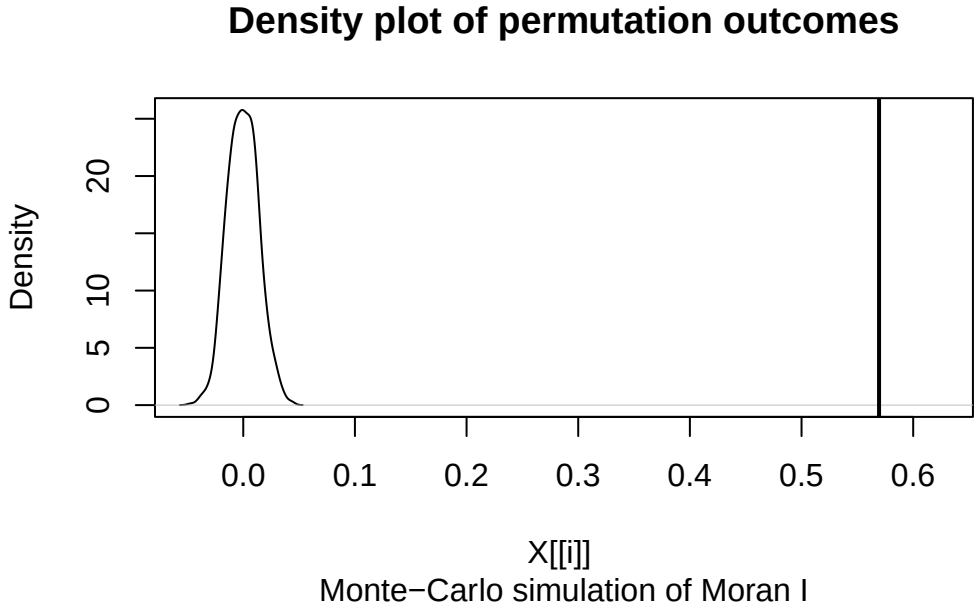
Todos os indicadores apresentaram um I de Moran maior que zero, com resultados estatisticamente significativos. Também é possível plotar a distribuição dos valores resultantes da simulação de Monte Carlo.

```
sapply(moran_results, plot)
```


Density plot of permutation outcomes
Density
0
5
10
20
0.0
0.1
0.2
0.3
0.4
0.5
0.6
X[[i]]
Monte−Carlo simulation of Moran I


Density plot of permutation outcomes
Density
0
5
15
25
0.0
0.1
0.2
0.3
0.4
0.5
0.6
X[[i]]
Monte−Carlo simulation of Moran I


Density plot of permutation outcomes
Density
0
5
10
20
0.0
0.1
0.2
0.3
X[[i]]
Monte−Carlo simulation of Moran I


Density plot of permutation outcomes
Density
0
5
10
20
0.0
0.1
0.2
0.3
0.4
0.5
0.6
X[[i]]
Monte−Carlo simulation of Moran I

```
[[1]]
NULL
```

```
[[2]]
NULL

[[3]]
NULL

[[4]]
NULL
```

9.2 Moran Local

Com o calculo da autocorrelação global elaborado, segue-se para o calculo da autocorrelação local através do Moran Local, possibilitando o mapeamento desse padrão espacial.

10 Correlação com indicadores socioeconômicos

11 Correlação com sinistros fatais