

Análise de Trajetos de Aeronaves com R e Leaflet

Análise de Código - Laboratório Especial

8 de setembro de 2025

Conteúdo

1	Introdução	2
2	Bibliotecas Utilizadas	2
3	Carregamento dos Dados	2
4	Função Principal: analisar_trajeto_completo	2
4.1	Filtração e Preparação dos Dados	2
4.2	Validação dos Dados	3
4.3	Criação da Tabela Tidy	3
4.4	Adição de Coordenadas Geográficas	4
5	Visualização com Leaflet	4
5.1	Inicialização do Mapa	4
5.2	Adição de Trajetos Sequenciais	4
5.3	Marcadores de Aeroportos	5
5.4	Painel de Estatísticas	5
6	Retorno da Função	6
7	Execução do Código	6

1 Introdução

Este documento apresenta uma análise detalhada do código R desenvolvido para analisar trajetos completos de aeronaves. O código utiliza bibliotecas especializadas para manipulação de dados geoespaciais e visualização interativa, criando mapas dinâmicos que mostram as rotas de voo de uma aeronave específica.

2 Bibliotecas Utilizadas

O código faz uso de várias bibliotecas essenciais do R:

- **tidyverse**: Conjunto de pacotes para manipulação e análise de dados
- **leaflet**: Criação de mapas interativos
- **lubridate**: Manipulação de datas e horários
- **ggmap**: Ferramentas para mapas estáticos
- **geosphere**: Cálculos geoespaciais

3 Carregamento dos Dados

```
1 flights <- read_csv(unz("flights.csv.zip", "flights.csv"))
2 airports <- read_csv('airports.csv')
```

Listing 1: Carregamento dos dados de voos e aeroportos

O código carrega dois conjuntos de dados fundamentais:

1. **flights**: Dados de voos em formato ZIP comprimido
2. **airports**: Informações sobre aeroportos, incluindo coordenadas geográficas

4 Função Principal: `analisar_trajeto_completo`

A função principal recebe dois parâmetros:

- **tail_number**: Número de identificação da aeronave
- **airports_df**: DataFrame com dados dos aeroportos

4.1 Filtração e Preparação dos Dados

```
1 dados_aeronave <- flights %>%
2   filter(TAIL_NUMBER == tail_number) %>%
3   arrange(YEAR, MONTH, DAY, DEPARTURE_TIME) %>%
4   mutate(
5     DEPARTURE_TIME_NUM = as.numeric(DEPARTURE_TIME),
6     DATA = make_date(YEAR, MONTH, DAY),
7     DATA_HORA = make_datetime(YEAR, MONTH, DAY,
```

```

8         hour = ifelse(is.na(DEPARTURE_TIME_NUM), 0,
9                       DEPARTURE_TIME_NUM %% 100),
10        min = ifelse(is.na(DEPARTURE_TIME_NUM), 0,
11                     DEPARTURE_TIME_NUM %% 100)),
12    VELOCIDADE_MEDIA = ifelse(AIR_TIME > 0 & !is.na(AIR_TIME),
13                             DISTANCE / (AIR_TIME/60), NA),
14    ESPESSURA_LINHA = scales::rescale(VELOCIDADE_MEDIA, to = c(1, 10),
15                                       na.rm = TRUE)
16  )

```

Listing 2: Filtração e preparação dos dados da aeronave

Esta seção realiza as seguintes operações:

1. **Filtração:** Seleciona apenas os voos da aeronave específica
2. **Ordenação:** Organiza os dados cronologicamente
3. **Criação de variáveis derivadas:**
 - DEPARTURE_TIME_NUM: Converte horário de partida para numérico
 - DATA: Cria objeto de data usando `make_date()`
 - DATA_HORA: Combina data e horário em formato datetime
 - VELOCIDADE_MEDIA: Calcula velocidade média ($\frac{\text{Distância}}{\text{Tempo de voo}}$)
 - ESPESSURA_LINHA: Normaliza velocidade para espessura visual (1-10)

4.2 Validação dos Dados

```

1 if (nrow(dados_aeronave) == 0) {
2   stop(paste("Aeronave", tail_number, "n o encontrada nos dados."))
3 }

```

Listing 3: Verificação da existência da aeronave

O código implementa uma verificação de segurança para garantir que a aeronave especificada existe no conjunto de dados.

4.3 Criação da Tabela Tidy

```

1 tabela_tidy <- dados_aeronave %>%
2   select(TAIL_NUMBER, DATA, DATA_HORA,
3          ORIGIN_AIRPORT, DESTINATION_AIRPORT,
4          AIRLINE, FLIGHT_NUMBER,
5          DEPARTURE_TIME, ARRIVAL_TIME,
6          AIR_TIME, DISTANCE, VELOCIDADE_MEDIA) %>%
7   arrange(DATA_HORA)

```

Listing 4: Geração da tabela tidy ordenada

Cria uma tabela limpa e organizada contendo apenas as variáveis relevantes para a análise.

4.4 Adição de Coordenadas Geográficas

```
1 dados_com_coords <- dados_aeronave %>%
2   left_join(airports_df %>%
3     select(IATA_CODE, LATITUDE, LONGITUDE) %>%
4     rename(ORIGIN_AIRPORT = IATA_CODE,
5           ORIGIN_LAT = LATITUDE,
6           ORIGIN_LON = LONGITUDE),
7   by = "ORIGIN_AIRPORT") %>%
8   left_join(airports_df %>%
9     select(IATA_CODE, LATITUDE, LONGITUDE) %>%
10    rename(DESTINATION_AIRPORT = IATA_CODE,
11          DEST_LAT = LATITUDE,
12          DEST_LON = LONGITUDE),
13    by = "DESTINATION_AIRPORT")
```

Listing 5: Junção com dados de coordenadas dos aeroportos

Realiza duas junções (*left joins*) para adicionar as coordenadas geográficas:

1. Coordenadas do aeroporto de origem
2. Coordenadas do aeroporto de destino

5 Visualização com Leaflet

5.1 Inicialização do Mapa

```
1 mapa <- leaflet() %>%
2   addTiles() %>%
3   addProviderTiles(providers$CartoDB.Positron)
```

Listing 6: Configuração inicial do mapa Leaflet

Cria um mapa Leaflet básico com tiles do CartoDB Positron, que oferece um estilo limpo e minimalista.

5.2 Adição de Trajetos Sequenciais

```
1 for (i in 1:nrow(dados_com_coords)) {
2   if (!is.na(dados_com_coords$ORIGIN_LAT[i]) && !is.na(dados_com_coords$
3     DEST_LAT[i])) {
4     popup_info <- paste(
5       "<b>Voo:</b>", dados_com_coords$FLIGHT_NUMBER[i],
6       "<br><b>Data:</b>", dados_com_coords$DATA[i],
7       "<br><b>Rota:</b>", dados_com_coords$ORIGIN_AIRPORT[i], " ",
8       dados_com_coords$DESTINATION_AIRPORT[i],
9       "<br><b>Velocidade:</b>", round(dados_com_coords$VELOCIDADE_MEDIA[
10         i], 1), "mph",
11       "<br><b>Dist ncia:</b>", dados_com_coords$DISTANCE[i], "milhas",
12       "<br><b>Tempo de voo:</b>", dados_com_coords$AIR_TIME[i], "min"
13     )
14   }
15   mapa <- mapa %>%
```

```

14     addPolylines(
15         lng = c(dados_com_coords$ORIGIN_LON[i], dados_com_coords$DEST_
16             LON[i]),
17         lat = c(dados_com_coords$ORIGIN_LAT[i], dados_com_coords$DEST_
18             LAT[i]),
19         weight = dados_com_coords$ESPESSURA_LINHA[i],
20         color = "#2c7bb6",
21         opacity = 0.8,
22         popup = popup_info
23     )
24 }
25 }

```

Listing 7: Loop para adicionar trajetos ao mapa

Este loop adiciona linhas ao mapa representando cada voo, onde:

- A espessura da linha é proporcional à velocidade média
- Cada linha contém informações detalhadas em popup
- As coordenadas são verificadas antes da adição

5.3 Marcadores de Aeroportos

```

1 aeroportos_visitados <- unique(c(dados_com_coords$ORIGIN_AIRPORT, dados_
2   com_coords$DESTINATION_AIRPORT))
3 coords_aeroportos <- airports_df %>%
4   filter(IATA_CODE %in% aeroportos_visitados)
5
6 mapa <- mapa %>%
7   addCircleMarkers(
8     data = coords_aeroportos,
9     lng = ~LONGITUDE,
10    lat = ~LATITUDE,
11    radius = 5,
12    color = "#d73027",
13    fillColor = "#fc8d59",
14    fillOpacity = 0.8,
15    popup = ~paste("<b>Aeroporto:</b>", IATA_CODE, "<br><b>Nome:</b>",
16        AIRPORT)
17 )

```

Listing 8: Adição de marcadores para aeroportos visitados

Adiciona marcadores circulares para todos os aeroportos únicos visitados pela aeronave.

5.4 Painel de Estatísticas

```

1 estatisticas <- paste(
2   "<h4> Estat sticas da Aeronave", tail_number, "</h4>",
3   "<b>Total de voos:</b>", nrow(dados_aeronave), "<br>",
4   "<b>Velocidade m dia:</b>", round(mean(dados_com_coords$VELOCIDADE_
5     MEDIA, na.rm = TRUE), 1), "mph<br>",
6   "<b>Dist ncia total:</b>", sum(dados_com_coords$DISTANCE, na.rm =
7     TRUE), "milhas<br>",

```

```

6   "<b>Aeroportos visitados:</b>", length(aeroportos_visitados), "<br>",
7   "<b>Per odo:</b>", min(dados_com_coords$DATA), "a", max(dados_com_
      coords$DATA)
8 )
9
10 mapa <- mapa %>%
11   addControl(
12     html = estatisticas,
13     position = "topright"
14   )

```

Listing 9: Criação do painel de controle com estatísticas

Adiciona um painel informativo no canto superior direito com estatísticas resumidas da aeronave.

6 Retorno da Função

A função retorna uma lista contendo:

1. **tabela_tidy**: DataFrame organizado com todos os voos
2. **mapa**: Objeto Leaflet interativo
3. **estatisticas**: Lista com métricas calculadas
4. **dados_completos**: Dataset completo com coordenadas

7 Execução do Código

```

1 resultado <- analisar_trajeto_completo("N407AS", airports)

```

Listing 10: Chamada da função para aeronave específica

O código é executado para a aeronave com identificação "N407AS", demonstrando a funcionalidade completa da análise.



Mapa de trajetos da aeronave N407AS mostrando 1234 voos realizados entre 2015-01-01 e 2015-12-31. As linhas azuis conectam aeroportos de origem e destino, com espessura variando conforme a velocidade

(258-581.1 mph). Os marcadores circulares vermelhos representam os 54 aeroportos visitados, concentrados principalmente na região oeste dos Estados Unidos. O painel superior direito exibe estatísticas operacionais: velocidade média de 435.2 mph, distância total percorrida de 178011 milhas. A visualização revela um padrão de operação regional intenso, com rotas frequentes entre aeroportos da Costa Oeste americana.