Documentação Técnica: Análise Visual de COFDI-ERP com R

Análise de Investimentos Chineses

1 Introdução Técnica

Este documento detalha a implementação técnica do pipeline de Análise Exploratória de Dados (EDA) e visualização, desenvolvido em R, para o projeto de Análise de Investimentos Chineses no Exterior (COFDI) e sua relação com a política ERP. O código emprega uma abordagem sistemática dividida em 6 etapas principais, utilizando o ecossistema 'Tidyverse' e pacotes de visualização estatística avançada.

2 Arquitetura de Pacotes (R) e Justificativas

2.1 Manipulação e Processamento de Dados (Tidyverse)

- **readr**: Carregamento eficiente de dados CSV (`read_csv`).
- **dplyr**: Gramática de manipulação de dados (verbos `mutate`, `filter`, `group_by`, `summarise`).
- **tidyr**: Reestruturação de dados, especialmente `pivot_longer` para facilitar a visualização.
- **stringr**: Funções de manipulação de strings e expressões regulares (`gsub`).
- **forcats**: Ferramentas para manipulação de variáveis categóricas (fatores), como `fct_na_value_to_level`.

2.2 Visualização de Dados

- **ggplot2**: Biblioteca base para a gramática dos gráficos, permitindo alta customização.
- **ggridges**: Criação de "Joy Plots"(gráficos de densidade sobrepostos) para visualização de distribuições.
- **ggthemes**: Temas visuais pré-definidos (`theme_tufte`, `theme_minimal`) para gráficos com qualidade de publicação.
- **scales**: Formatação de eixos e legendas (ex: `percent`, `comma`).
- **viridis**: Paletas de cores.

2.3 Visualização Estatística

- **ggstatsplot**: Pacote de alto nível que combina `ggplot2` com testes estatísticos. Gera gráficos (ex: boxplots, histogramas) com resultados estatísticos (p-valores, tamanho do efeito) embutidos.

3 Implementação do Pipeline de Análise em R

O script R é estruturado sequencialmente, desde a configuração até a geração final dos gráficos.

3.1 ETAPA 1: Configuração e Carregamento de Dados

Pacotes principais: readr, stringr

```
data_file <- "C:/Users/PC GAMER/Downloads/erp-ofdi-model/data/data.csv"
output_dir <- "C:/Users/PC GAMER/Downloads/erp-ofdi-model/results/R"

if (!dir.exists(output_dir)) {
  dir.create(output_dir, recursive = TRUE)
}

dados_base <- readr::read_csv(data_file, col_types = cols(.default = "c"))
names(dados_base) <- gsub("\\s+", ".", names(dados_base))
```

Objetivo: Definir caminhos, garantir a existência do diretório de saída, carregar os dados como texto (`col_types = "c"`) para prevenir erros de parsing, e padronizar os nomes das colunas (substituindo espaços por pontos).

3.2 ETAPA 2: Limpeza e Engenharia de Features

Pacotes: dplyr, stringr, tidyr, forcats

3.2.1 Limpeza Numérica e Tratamento de NAs

Uma função `parse_num` é definida para limpar colunas numéricas que contêm caracteres não numéricos (ex: vírgulas, símbolos monetários).

```
parse_num <- function(x) {
  x_clean <- as.character(x)
  x_clean <- gsub("[^0-9\\.,-]", "", x_clean) # Remove caracteres indesejados
  x_clean <- gsub(",", "", x_clean) # Remove separador de milhar
  suppressWarnings(as.numeric(x_clean))
}
dados$Quantity.in.Millions <- parse_num(dados$Quantity.in.Millions)
```

3.2.2 Criação de Variáveis de Política (Going Global)

Features binárias são criadas com base no ano para capturar as diferentes fases da política.

```
dados_clean <- dados_clean %>%
  mutate(
    Going_Global1.0 = as.numeric(Year %in% c(2005:2011)),
    Going_Global2.0 = as.numeric(Year %in% c(2012:2016)),
    Going_Global3.0 = as.numeric(Year %in% c(2017:2024))
  )
```

3.2.3 Criação de Variáveis Categóricas (BRI, Greenfield)

Colunas de texto com múltiplas entradas (ex: "y", "yes", "1") são padronizadas em categorias binárias.

```
dados_clean$bri_grp <- ifelse(grepl("^(1|bri|y|yes|sim|true)$", ...), "BRI", "Others")
dados_clean$GF <- ifelse(grepl("^(g|greenfield|1|y|yes|sim|true)$", ...), "Greenfield", "Other")
```

3.2.4 Binarização e Preparação de Fatores

A variável `Share.Size` é convertida de numérica para categórica (binned) usando a função `cut`. As demais colunas são transformadas em fatores, onde os valores ausentes (NAs) são tratados como uma categoria explícita, denominada ("Unknown").

```
dados_clean <- dados_clean %>%
  mutate(
    Share.Bin = cut(Share.Size.Numeric,
                    breaks = c(-Inf, 19, 39, 59, 79, Inf),
                    labels = c("0-19", "20-39", "40-59", "60-79", "80-99")
    )
  )
...
forcats::fct_na_value_to_level(as.factor(x), "Unknown")
```

3.3 ETAPA 3: Gráficos de Densidade (Joy Plots)

Pacotes: ggplot2, ggridges, viridis

Uma função `create_density_plot` é criada para automatizar a geração de gráficos de densidade (Joy Plots), que são ideais para comparar a distribuição de `Quantity.in.Millions` através de diferentes categorias (Setor, Região, etc.).

```
create_density_plot <- function(data, y_col_name, y_label, file_name, scale = 3) {
  p <- ggplot(data, aes(x = Quantity.in.Millions, y = .data[[y_col_name]], fill = ..x..)) +
    ggridges::geom_density_ridges_gradient(scale = scale, rel_min_height = 0.01) +
    scale_fill_viridis_c(option = "plasma", name = "Amount (Millions)") +
    labs(...) +
    theme_minimal()
  ggsave(file.path(output_dir, file_name), p, width = 10, height = 8, dpi = 150)
}
```

Característica: Esta função utiliza `geom_density_ridges_gradient` do `ggridges` para mostrar como a distribuição dos valores de investimento muda para cada nível da variável `y`.

3.4 ETAPA 4: Gráficos de Barras e Histogramas

Pacotes: ggplot2, ggstatsplot, ggthemes, dplyr

3.4.1 Histogramas com Estatísticas (ggstatsplot)

Para analisar a distribuição dos investimentos por fase da política "Going Global", utiliza-se `grouped_gghistostats`. Esta função plota histogramas para cada grupo, sobrepõe estatísticas descritivas e testes de comparação.

```
p_hist_gg <- grouped_gghistostats(
  data = df_gg_hist,
  x = Quantity.in.Millions,
  grouping.var = Going_Global_Phase,
  type = "p", # Paramétrico (usa média)
  ggtheme = ggthemes::theme_tufte(),
  messages = FALSE
)
```

3.4.2 Barras Empilhadas por Fase (Market Share)

Para comparar a composição do investimento (por `Share.Bin`) entre as fases "Going Global", um gráfico de barras facetado é utilizado.

```
p_gg_share <- ggplot(df_gg_summary, aes(x = Share.Bin, y = Amount_investment)) +
  geom_bar(stat = "identity", fill = "darkgrey") +
  geom_text(
    aes(label = scales::percent(Proportion, accuracy = 0.1)),
    vjust = -0.5, size = 3
  ) +
  facet_wrap(~Going_Global_Phase, scales = "free_y") +
  ...
  theme_minimal(base_size = 12)
```

Característica: `facet_wrap` é usado para criar sub-gráficos separados para cada `Going_Global_Phase`, permitindo uma comparação direta das proporções de investimento.

3.5 ETAPA 5: Análises Estatísticas Visuais (Boxplots)

Pacotes: ggstatsplot, ggthemes

Esta etapa foca na comparação de grupos usando `ggbetweenstats`, que cria boxplots (ou violin plots) e automaticamente executa os testes estatísticos apropriados (ex: ANOVA ou Kruskal-Wallis) e testes post-hoc (ex: Tukey) para comparar as médias/medianas entre os grupos.

```
p_boxplot_year <- ggbetweenstats(
  data = dados_clean,
  x = Year,
  y = Quantity.in.Millions,
  type = "p", # Paramétrico
  pairwise.display = "s", # "significant" - mostra apenas comparações significativas
  ggtheme = ggthemes::theme_tufte(),
  outlier.tagging = TRUE
)
```

Metodologia: Esta abordagem é aplicada para analisar o efeito de `Year`, `GF` (Greenfield), e `bri_grp` (BRI) sobre o valor do investimento (`Quantity.in.Millions`), integrando visualização e inferência estatística em um único gráfico.

4 Características Técnicas da Implementação em R

4.1 Pré-processamento de Dados

- **Pipeline 'Tidyverse'**: Uso intensivo do operador *pipe* (`%>%`) para criar um fluxo de transformação de dados lógico, legível e sequencial.
- **Limpeza Robusta**: A função `parse_num` utiliza `gsub` com expressões regulares para lidar com dados numéricos inconsistentes de forma flexível.
- **Tratamento de Fatores**: Uso de `forcats` para garantir que valores ausentes (NAs) sejam tratados como um nível categórico explícito ("Unknown"), evitando perda de dados em visualizações.

4.2 Visualização e Análise

- **Visualização Estatística Integrada**: O uso de `ggstatsplot` é central, pois automatiza a execução e apresentação de testes de hipóteses (ex: t-testes, ANOVAs) diretamente nos gráficos.
- **Automação de Gráficos**: A criação de funções wrapper (ex: `create_density_plot`) permite a geração padronizada e reproduzível de múltiplos gráficos complexos com uma única chamada.
- **Facetamento**: O uso de `facet_wrap` (visto na Etapa 4) é uma técnica chave para comparar distribuições e proporções através de subgrupos.

5 Conclusão

Esta implementação em R constitui um pipeline reproduzível para a Análise Exploratória de Dados (EDA) do projeto COFDI-ERP. Focando no ecossistema 'Tidyverse' e em pacotes de visualização estatística como `ggstatsplot`, o pipeline gera visualizações melhores em informação, que não apenas mostram os dados, mas também os resultados de testes estatísticos relevantes para a pequisa.