Correlations and Reshaping Data

Para checar correlação, pedi ajuda ao Chat GTP para gerar um CSV de informações numéricas metereológicas. Abaixo o código sugerido com pequenas adaptações para forçar correlações para efeito de demonstração:

```r
# Carrega a biblioteca random
library(random)

# Define as colunas
colunas <- c("Data", "Temperatura (C)", "Pressão Atmosférica (hPa)",
             "Umidade (%)", "Velocidade do Vento (km/h)", "Direção do Vento",
             "Precipitação (mm)", "Nebulosidade (%)", "Índice UV", "Tempestade")

# Gera os dados aleatórios
dados <- data.frame(matrix(ncol = 10, nrow = 1000))
colnames(dados) <- colunas

for(i in 1:1000) {
  datas <- seq.Date(as.Date("2022-01-01"), as.Date("2022-12-31"), by = 1)
  dados[i, "Data"] <- as.character(sample(datas, 1))
  dados[i, "Temperatura (C)"] <- runif(1, -10, 40)
  dados[i, "Pressão Atmosférica (hPa)"] <- runif(1, 900, 1100)
  dados[i, "Umidade (%)"] <- runif(1, 0, 100)
  dados[i, "Velocidade do Vento (km/h)"] <- runif(1, 0, 100)
  direcoes_vento <- c("N", "NE", "E", "SE", "S", "SW", "W", "NW")
  dados[i, "Direção do Vento"] <- sample(direcoes_vento, 1)
  dados[i, "Precipitação (mm)"] <- runif(1, 0, 50)
  dados[i, "Nebulosidade (%)"] <- runif(1, 0, 100)
  dados[i, "Índice UV"] <- runif(1, 0, 10)
  tempestades <- c("Sim", "Não")
  dados[i, "Tempestade"] <- sample(tempestades, 1)
}

amostra1 <- dados %>% filter(`Temperatura (C)`>30,`Umidade (%)` <20)
amostra2 <- dados %>% filter(`Temperatura (C)`<0,`Umidade (%)` >70)
```

```r
dados <- rbind(dados %>% slice_sample(n=10),
               amostra1 %>% slice(rep(1:n(), each=10)),
               amostra2 %>% slice(rep(1:n(), each=10)))
# Salva os dados em um arquivo CSV
write.csv(dados, "dados-meteorologicos.csv", row.names = FALSE)

dados %>% head()
```

```
##         Data Temperatura (C) Pressão Atmosférica (hPa) Umidade (%)
## 1 2022-07-21        8.672447                  989.7860    93.79863
## 2 2022-09-15       38.267171                 1064.3880    58.37699
## 3 2022-06-27       20.602561                  933.2506    49.83508
## 4 2022-02-04       34.802825                 1014.1801    32.52858
## 5 2022-03-19       14.571144                  925.3798    38.56407
## 6 2022-02-20       15.589813                  917.6721    51.06920
##   Velocidade do Vento (km/h) Direção do Vento Precipitação (mm)
## 1                   39.76243                E         44.400254
## 2                   61.96181               NW         12.857980
## 3                   76.38719               SW         41.341544
## 4                   86.10556                E          2.674158
## 5                   86.45434               SW          9.731735
## 6                   79.75742               NW         36.963813
##   Nebulosidade (%) Índice UV Tempestade
## 1        23.225197 1.1081520        Sim
## 2         4.044583 2.8790634        Não
## 3        70.332874 0.6044239        Sim
## 4        30.883673 8.6794094        Não
## 5        57.340055 2.7718012        Não
## 6        29.261653 9.8237043        Sim
```

Simplificando o nome das colunas

```r
colunas <- c("data", "temp", "press_atmo", "umid", "veloc_vento", "dir_vento",
             "precipit", "nebulos", "uv", "tempestade")
colnames(dados) <- colunas
```

Correlação

Uma maneira eficiente de enxergar a correlação entre duas informações é calculando o coeficiente de correlação de Pearson. O cálculo gera um resultado que fica num intervalo de 1.00 a -1.00.

- 1.00 indica correlação positiva, ou seja, quando “a” aumenta “b” também aumenta
- -1.00 indica correlação negativa, ou seja, quando “a” aumenta “b” diminui
- entre -0.25 e 0.25 consideramos baixa ou nenhuma correlação
- valores acima de 0.25 ou abaixo de -0.25 existe alguma correlação
- valores acima de 0.75 ou abaixo de -0.75 existe forte correlação

Para o cálculo é necessário filtrar appenas colunas numéricas.

```
matriz_correlacao <- dados %>%
  select(where(is.numeric)) %>%
  cor(use="pairwise.complete.obs")

matriz_correlacao
```

```
##                   temp  press_atmo        umid veloc_vento    precipit
## temp         1.00000000 -0.08427619 -0.95639340 -0.02442944 -0.07944685
## press_atmo  -0.08427619  1.00000000  0.02774809 -0.12339840  0.02962176
## umid        -0.95639340  0.02774809  1.00000000  0.04403701  0.10696029
## veloc_vento -0.02442944 -0.12339840  0.04403701  1.00000000 -0.13980419
## precipit    -0.07944685  0.02962176  0.10696029 -0.13980419  1.00000000
## nebulos     -0.06206250 -0.01181750  0.08341229  0.04408318 -0.01402840
## uv          -0.12481583 -0.01711916  0.08114893  0.08931071  0.09230838
##                 nebulos          uv
## temp        -0.06206250 -0.12481583
## press_atmo  -0.01181750 -0.01711916
## umid         0.08341229  0.08114893
## veloc_vento  0.04408318  0.08931071
## precipit    -0.01402840  0.09230838
## nebulos      1.00000000  0.17725699
## uv           0.17725699  1.00000000
```

É possível filtrar apenas uma das informações para buscar entender como ela se relaciona com as demais. Nesse caso selecionamos a coluna de temperatura que demonstrou ter forte correlação com a umidade.

```
df_correlacao <- matriz_correlacao %>%
  as_tibble(rownames = "variable")

df_correlacao %>% select(variable,temp)
```

```
## # A tibble: 7 x 2
```

```
##   variable       temp
##   <chr>         <dbl>
## 1 temp          1
## 2 press_atmo   -0.0843
## 3 umid         -0.956
## 4 veloc_vento  -0.0244
## 5 precipit     -0.0794
## 6 nebulos      -0.0621
## 7 uv           -0.125
```

Pivot

Às vezes para plotar um gráfico se faz necessário remodelar o dataset. Por exemplo aqui queremos comparar o comportamento das métricas em relação à temperatura, para isso o ideal é ter uma coluna com as informações e outra para os valores. Assim se formos plotar em um gráfico, para diferenciar pela cor com legenda.

O Pivot Longer serve justamente para "alongar" o dataset, tranformar colunas em linhas. (O Pivot Wider faz o movimento inverso)

```
pivot_df <- dados %>%
  pivot_longer(cols= c(umid,veloc_vento,precipit,nebulos),
               names_to  = "tipo_metrica",
               values_to = "valor_metrica")

pivot_df %>% head()
```

```
## # A tibble: 6 x 8
##   data     temp press_atmo dir_vento    uv tempestade tipo_metrica valor_metrica
##   <chr>   <dbl>      <dbl> <chr>     <dbl> <chr>      <chr>                <dbl>
## 1 2022-0~  8.67       990. E          1.11 Sim        umid                  93.8
## 2 2022-0~  8.67       990. E          1.11 Sim        veloc_vento           39.8
## 3 2022-0~  8.67       990. E          1.11 Sim        precipit              44.4
## 4 2022-0~  8.67       990. E          1.11 Sim        nebulos               23.2
## 5 2022-0~ 38.3       1064. NW         2.88 Não        umid                  58.4
## 6 2022-0~ 38.3       1064. NW         2.88 Não        veloc_vento           62.0
```

Com o gráfico plotado fica evidente a correlação entre a temperatura e a umidade. Quando a temperatura está alta a umidade fica muito baixa, e vice-versa, o que faz muito sentido comparado com a métrica que vimos acima quase atingindo -1 no coeficiente de correlação.

```
pivot_df %>%
  ggplot(aes(x=valor_metrica,y=temp,color=tipo_metrica))+
  geom_point()
```