

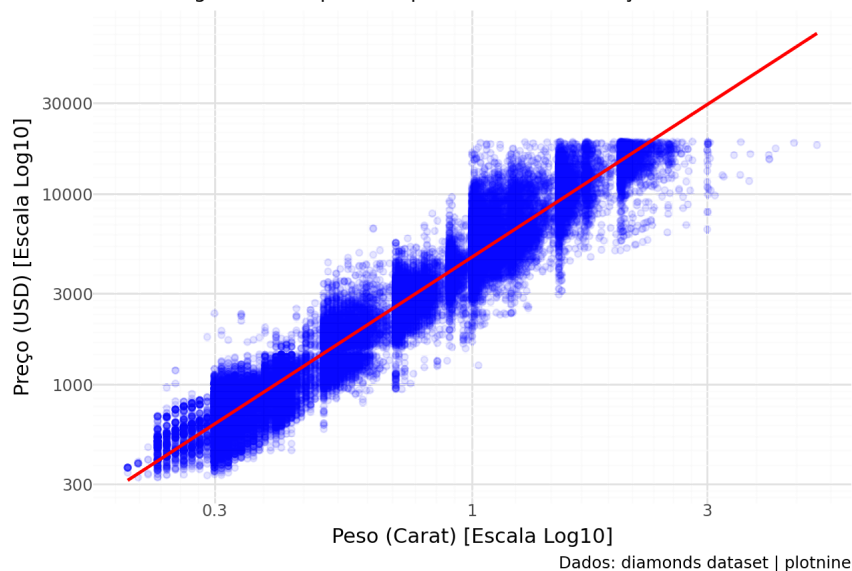
```
# Instalar bibliotecas necessárias (se ainda não estiverem instaladas)
# !pip install plotnine pandas

import pandas as pd
from plotnine import (
    ggplot,
    aes,
    geom_point,
    geom_smooth,
    facet_wrap,
    labs,
    scale_x_log10,
    scale_y_log10,
    theme_minimal,
    theme,
    element_text,
    element_rect,
    position_jitter
)
from plotnine.data import diamonds
```

```
# Gráfico de dispersão de Price vs. Carat com escalas logarítmicas
plot_carat_price = (
    ggplot(diamonds, aes(x='carat', y='price'))
    + geom_point(alpha=0.1, color='blue') # alpha reduz a opacidade para mitigar overplotting
    + geom_smooth(method='lm', color='red', fill='red', alpha=0.2, se=True) # Linha de regressão linear com IC
    + scale_x_log10(name='Peso (Carat)')
    + scale_y_log10(name='Preço (USD)')
    + labs(
        title='Associação entre Peso do Diamante (Carat) e Preço (USD)',
        subtitle='Escalas logarítmicas aplicadas para linearizar a relação.',
        caption='Dados: diamonds dataset | plotnine'
    )
    + theme_minimal()
)

plot_carat_price
```

Associação entre Peso do Diamante (Carat) e Preço (USD)  
Escalas logarítmicas aplicadas para linearizar a relação.



```
# Gráfico de dispersão de Price vs. Carat, facetado por Cut
plot_carat_price_by_cut = (
    ggplot(diamonds, aes(x='carat', y='price'))
    + geom_point(aes(color='cut'), alpha=0.2, size=1.5) # CORRETO # Cor pelo tipo de corte
    + geom_smooth(method='lm', color='black', linetype='dashed', se=False) # Linha de tendência para CADA facet
    + scale_x_log10(name='Peso (Carat) [Escala Log10]')
    + scale_y_log10(name='Preço (USD) [Escala Log10]')
    + facet_wrap('~ cut', nrow=2) # Facetamento por qualidade de corte (cut)
    + labs(
        title='Relação Price vs. Carat FACetada pela Qualidade de Corte (Cut)',
        subtitle='A tendência Price-Carat é similar em todos os cortes.',
        caption='Dados: diamonds dataset | plotnine'
    )
    + theme_minimal()
)
```

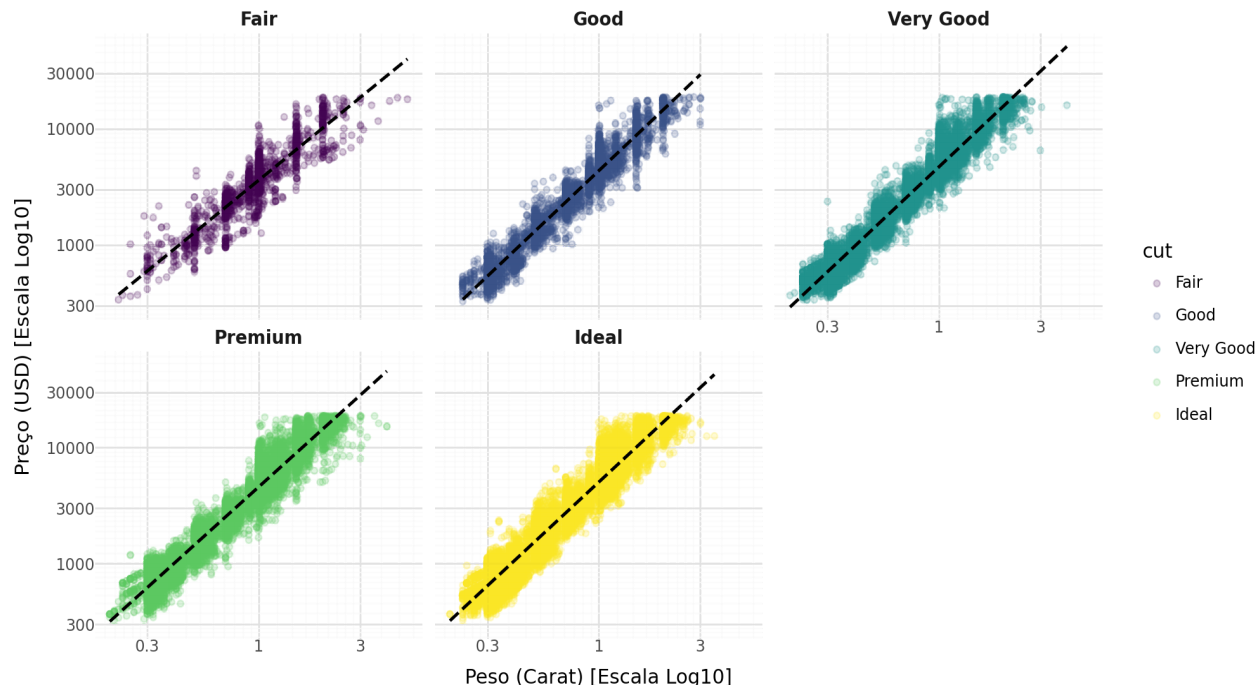
```
+ theme(
  subplots_adjust={'wspace': 0.3, 'hspace': 0.3},
  strip_text=element_text(size=10, weight='bold'),
  figure_size=(10, 6) # Ajuste de tamanho para caber todas as facetas
)
)

plot_carat_price_by_cut
```

/usr/local/lib/python3.12/dist-packages/plotnine/themes/themeable.py:2438: FutureWarning: You no longer need to use subplots

### Relação Price vs. Carat FACETADA pela Qualidade de Corte (Cut)

A tendência Price-Carat é similar em todos os cortes.



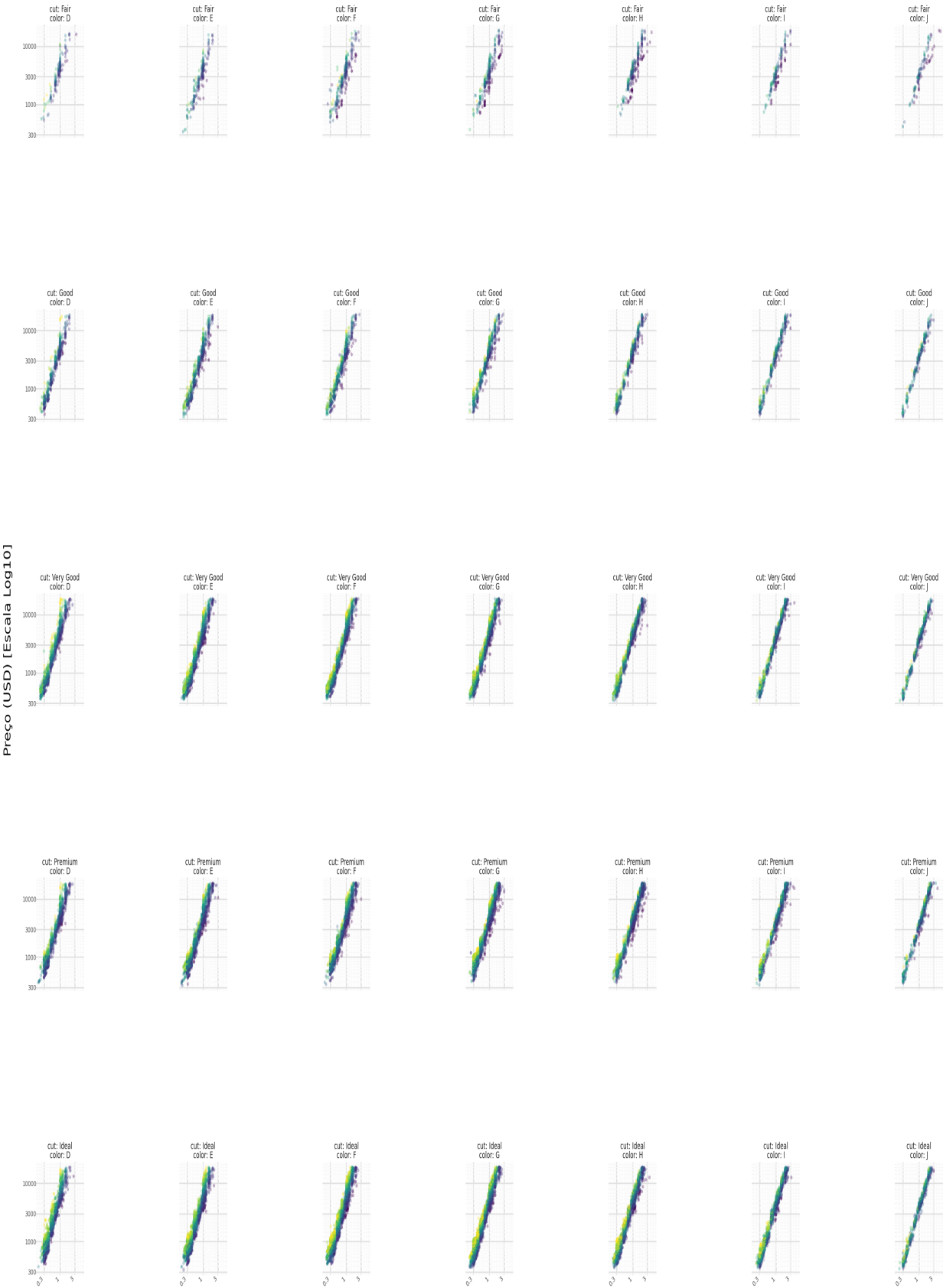
```
# Importações necessárias
from plotnine import (
    ggplot, aes, geom_point, facet_wrap, labs,
    scale_x_log10, scale_y_log10, theme_minimal, theme, element_text, element_blank
)
from plotnine.data import diamonds

# Gráfico de dispersão de Price vs. Carat, facetado por Cut e Color, com cor por Clarity
plot_complex_3vars = (
    ggplot(diamonds, aes(x='carat', y='price', color='clarity')) # Clarity mapeada para cor
    + geom_point(alpha=0.2, size=0.5) # Pontos menores e mais transparentes
    + scale_x_log10(name='Peso (Carat) [Escala Log10]')
    + scale_y_log10(name='Preço (USD) [Escala Log10]')
    + facet_wrap(('cut', 'color'), labeller='label_both', nrow=5) # Facetamento duplo: Cut e Color
    + labs(
        title='Relação Price vs. Carat por Qualidade (Cut, Color e Clarity)',
        subtitle='Cada faceta é uma combinação de Corte (Cut) e Cor (Color). Cor dos pontos = Clareza (Clarity).',
        color='Clareza',
        caption='Dados: diamonds dataset | plotnine'
    )
    + theme_minimal()
    + theme(
        # Ajuste o tamanho da figura para tentar acomodar 35 painéis
        figure_size=(15, 15),
        # Reduzir o tamanho da fonte para os rótulos das facetas e eixos
        strip_text_x=element_text(size=6),
        strip_text_y=element_text(size=6),
        axis_text_x=element_text(size=5, rotation=45, hjust=1),
        axis_text_y=element_text(size=5),
        legend_position='bottom',
        panel_spacing_x=0.1, # Espaçamento mínimo entre painéis
        panel_spacing_y=0.1
    )
)

plot_complex_3vars
```

Relação Price vs. Carat por Qualidade (Cut, Color e Clarity)

Cada faceta é uma combinação de Corte (Cut) e Cor (Color). Cor dos pontos = Clareza (Clarity).



Clareza

- I1
- SI1
- VS1
- VS2
- SI2
- VS2
- WS2
- IF