

Project 3: Data Science Skills Trend Analysis

Candace Grant

2025-10-17

Understanding the Data Science Skills Landscape

This analysis examines Google search trends for three essential data science skills—Python, SQL, and Tableau—over a five-year period from October 2020 to October 2025. By analyzing search interest patterns, we can identify which skills are gaining traction, which are declining, and what this means for data professionals and employers.

Key Questions We'll Answer:

- Which skill dominates the data science landscape?
- How have these skills evolved over time?
- What does the future hold for Python demand?

```
# Core packages
library(DBI)
library(RSQLite)
library(dplyr)
library(ggplot2)
library(lubridate)
library(tidyr)
library(readr)
library(knitr)
library(scales)
library(zoo)
```

```
# Load the CSV
trends_viz <- read_csv("data/trends_long.csv", show_col_types = FALSE)

# Convert date column from numeric to Date
trends_viz <- trends_viz %>%
  mutate(date = as.Date(date, origin = "1970-01-01"))

# Verify the conversion
cat("Date range:", as.character(min(trends_viz$date)), "to",
    as.character(max(trends_viz$date)), "\n")
```

Converting date column to date type to prep for data visualizations and predictive analysis

```
## Date range: 2020-10-11 to 2025-10-12
```

```
cat("Number of weeks:", n_distinct(trends_viz$date), "\n")
```

```
## Number of weeks: 262
```

```
# Preview
#head(trends_viz, 10)
```

```
# Summary by skill
trends_viz %>%
  group_by(skill_name) %>%
  summarise(
    observations = n(),
    avg_interest = round(mean(interest), 2),
    min_interest = min(interest),
    max_interest = max(interest)
  ) %>%
  knitr::kable(caption = "Summary Statistics by Skill")
```

Summarize the dataset by skill

Table 1: Summary Statistics by Skill

skill_name	observations	avg_interest	min_interest	max_interest
python	262	65.18	30.0	100
sql	262	16.50	6.0	22
tableau	262	1.20	0.5	2

Initial Observations

The summary statistics reveal a striking disparity in search interest across the three skills. **Python dominates with an average interest score of 65.18**—nearly four times higher than SQL (16.50) and over 50 times higher than Tableau (1.20). Python also shows the widest range of interest (30-100), suggesting significant fluctuations over time, while SQL and Tableau remain relatively stable at lower levels.

This initial snapshot suggests Python has become the clear focal point for data science skill development, but we need to examine the trends over time to understand the full story.

```
#Which skill has been most consistently popular over time?
ggplot(trends_viz, aes(x = date, y = interest, color = skill_name)) +
  geom_line(linewidth = 1) +
  labs(
    title = "Google Search Interest: Data Science Skills Over Time",
    subtitle = paste("Weekly trends from", min(trends_viz$date), "to", max(trends_viz$date)),
    x = "Date",
    y = "Search Interest Score (0-100)",
```

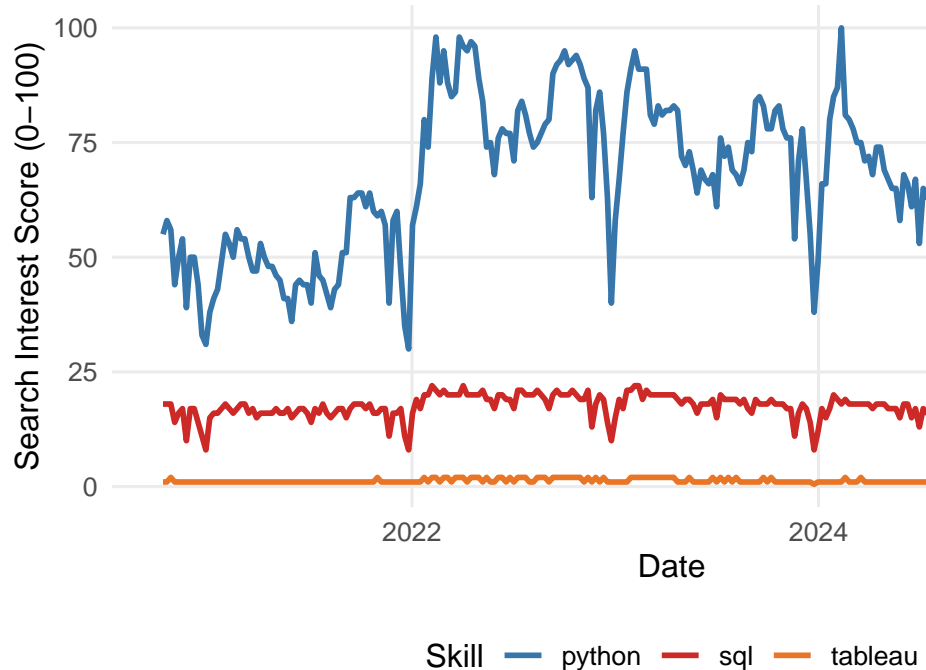
```

color = "Skill",
caption = "Data source: Google Trends"
) +
scale_color_manual(values = c("python" = "#3776AB",
                              "sql" = "#CC2927",
                              "tableau" = "#E97627")) +
theme_minimal(base_size = 12) +
theme(
  legend.position = "bottom",
  plot.title = element_text(face = "bold", size = 16),
  panel.grid.minor = element_blank()
)

```

Google Search Interest: Data Science Skills

Weekly trends from 2020-10-11 to 2025-10-12



Visualizing Search Interest Over Time

The Big Picture: Five Years of Search Trends

This visualization tells a compelling story of Python's dominance in the data science ecosystem. Python consistently maintains the highest search interest throughout the entire period, though it shows considerable volatility with several dramatic peaks and valleys. The high volatility in Python searches suggests it's an actively evolving skill with fluctuating demand, possibly tied to academic cycles, new releases, or market trends.

Key Insights:

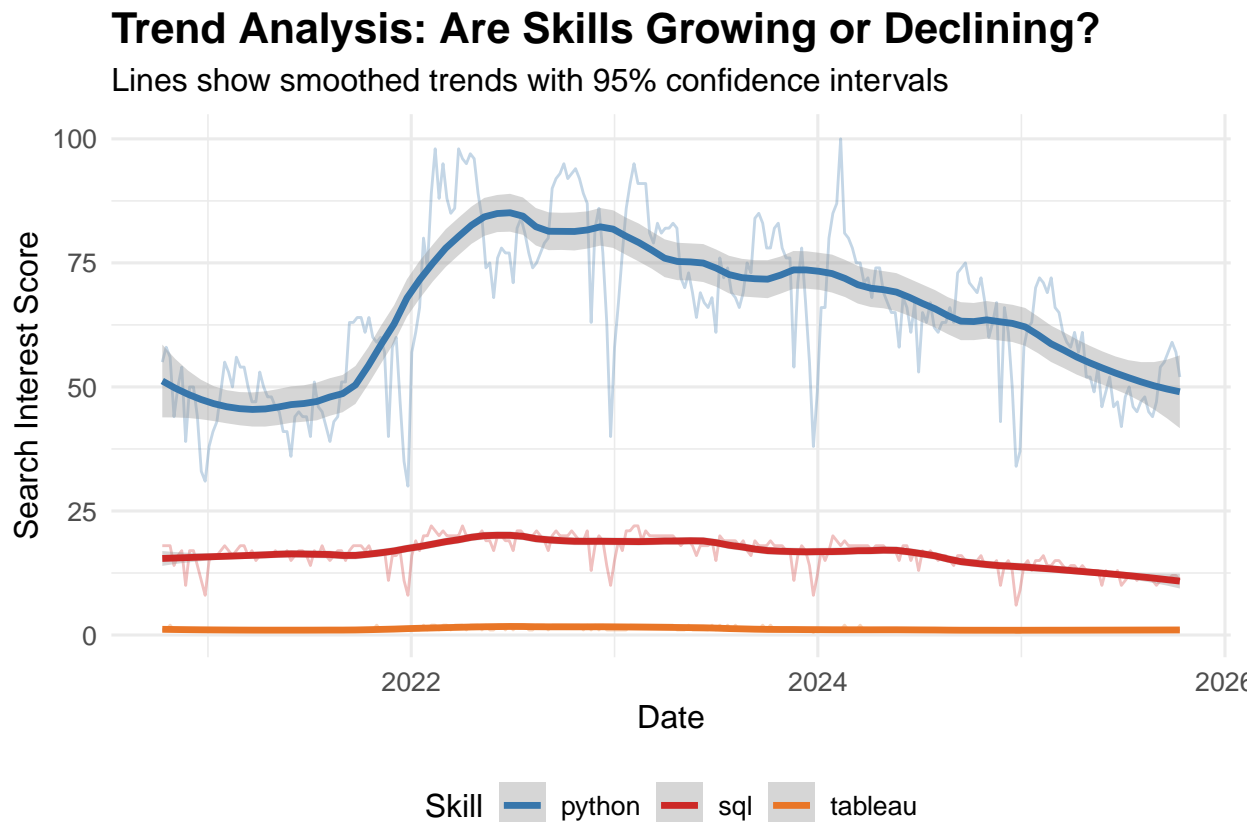
- Python leads decisively: It never drops below the other two skills, maintaining its position as the most sought-after data science skill
- SQL shows stability: Red line remains relatively flat, indicating steady but modest interest over time

- Tableau barely registers: Orange line hugs the bottom, showing minimal search activity

```
ggplot(trends_viz, aes(x = date, y = interest, color = skill_name)) +
  geom_line(alpha = 0.3, linewidth = 0.5) + # Actual data (faded)
  geom_smooth(method = "loess", se = TRUE, linewidth = 1.2, span = 0.3) + # Smoothed trend
  labs(
    title = "Trend Analysis: Are Skills Growing or Declining?",
    subtitle = "Lines show smoothed trends with 95% confidence intervals",
    x = "Date",
    y = "Search Interest Score",
    color = "Skill"
  ) +
  scale_color_manual(values = c("python" = "#3776AB",
                                "sql" = "#CC2927",
                                "tableau" = "#E97627")) +
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "bottom",
    plot.title = element_text(face = "bold", size = 16)
  )
```

Smoothed Trend Lines: Filtering Noise to Reveal Underlying Patterns

'geom_smooth()' using formula = 'y ~ x'



Revealing the Underlying Patterns

By applying smoothing to the noisy weekly data, we can see the true directional trends more clearly. This analysis reveals a pattern:

- The smoothed lines show that Python interest has been declining since around 2022, despite remaining the dominant skill.
- The trend line shows a clear peak in early 2022 followed by a steady downward trajectory.
- SQL shows a similar but less pronounced decline, while Tableau's minimal interest has remained essentially flat.

While Python remains the most popular skill, its peak popularity may have passed. It may indicate market maturation as Python proficiency becomes a baseline expectation rather than a differentiator.

```
growth_stats <- trends_viz %>%
  group_by(skill_name) %>%
  arrange(date) %>%
  summarise(
    first_year = year(min(date)),
    last_year = year(max(date)),
    first_year_avg = mean(interest[date <= min(date) + years(1)], na.rm = TRUE),
    last_year_avg = mean(interest[date >= max(date) - years(1)], na.rm = TRUE),
    overall_change = last_year_avg - first_year_avg,
    percent_change = round((last_year_avg - first_year_avg) / first_year_avg * 100, 1)
  )

print(growth_stats)
```

Calculate Search Trend Rates

```
## # A tibble: 3 x 7
##   skill_name first_year last_year first_year_avg last_year_avg overall_change
##   <chr>         <dbl>    <dbl>         <dbl>         <dbl>         <dbl>
## 1 python         2020      2025          48.2           55.8           7.62
## 2 sql            2020      2025          16.1           12.8          -3.36
## 3 tableau        2020      2025           1.02           0.981         -0.0377
## # i 1 more variable: percent_change <dbl>
```

```
# Data formatted as a table
growth_stats %>%
  knitr::kable(
    col.names = c("Skill", "First Year", "Last Year",
                  "First Year Avg", "Last Year Avg",
                  "Change", "% Change"),
    digits = 2,
    caption = "Growth Statistics: Comparing First and Last Year of Data Collection"
  )
```

Table 2: Growth Statistics: Comparing First and Last Year of Data Collection

Skill	First Year	Last Year	First Year Avg	Last Year Avg	Change	% Change
python	2020	2025	48.17	55.79	7.62	15.8
sql	2020	2025	16.11	12.75	-3.36	-20.8
tableau	2020	2025	1.02	0.98	-0.04	-3.7

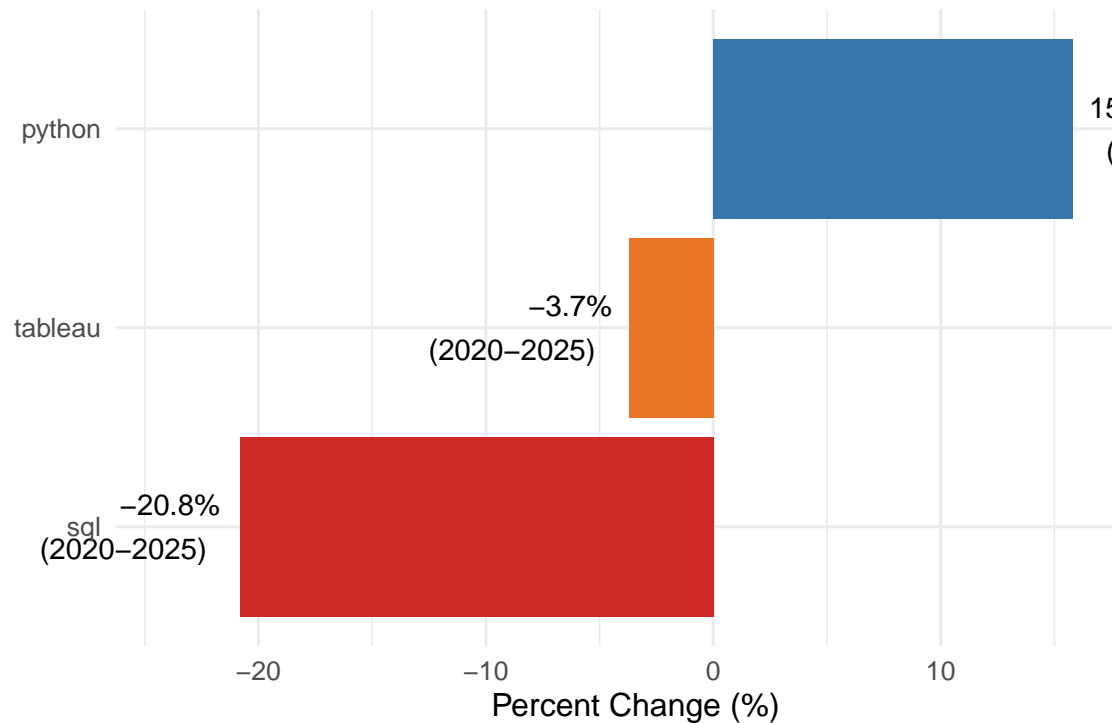
Search Parameter Growth Analysis: Winners and Losers

Over the 5-year period from 2020 to 2025, Python was the only skill showing positive growth (+15.8%), while SQL experienced the sharpest decline (-20.8%) and Tableau remained stagnant with minimal change (-3.7%).

```
ggplot(growth_stats, aes(x = reorder(skill_name, percent_change), y = percent_change, fill = skill_name)) +
  geom_col(show.legend = FALSE) +
  geom_text(aes(label = paste0(percent_change, "%\n(", first_year, "-", last_year, ")")),
    hjust = ifelse(growth_stats$percent_change > 0, -0.2, 1.2),
    size = 4) +
  coord_flip(clip = "off") + # Turn off clipping
  labs(
    title = "Overall Trend in Data Science Search Terms",
    subtitle = "Comparing first year average to last year average",
    x = NULL,
    y = "Percent Change (%)"
  ) +
  scale_fill_manual(values = c("python" = "#3776AB", "sql" = "#CC2927", "tableau" = "#E97627")) +
  scale_y_continuous(expand = expansion(mult = c(0.15, 0.1))) + # Add padding
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(face = "bold"),
    plot.margin = margin(10, 30, 10, 10) # Add right margin
  )
```

Overall Trend in Data Science Search Terms

Comparing first year average to last year average



Visualize Search Trends

Recent Trends in Search Terms

The growing volume of Python-related searches reflects its status as the skill people most actively pursue and research. Meanwhile, declining search interest in SQL and Tableau suggests these tools have either become less central to data science learning paths or are increasingly bundled into broader Python-focused educational searches.

```
# Filter for last 24 months
recent_24mo <- trends_viz %>%
  filter(date >= max(date) - months(24))

# Plot last 24 months
ggplot(recent_24mo, aes(x = date, y = interest, color = skill_name)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 1, alpha = 0.5) +
  labs(
    title = "Recent Trends: Last 24 Months",
    subtitle = paste("Focus period:", min(recent_24mo$date), "to", max(recent_24mo$date)),
    x = "Date",
    y = "Search Interest Score",
    color = "Skill"
  ) +
  scale_color_manual(values = c("python" = "#3776AB",
                                "sql" = "#CC2927",
```

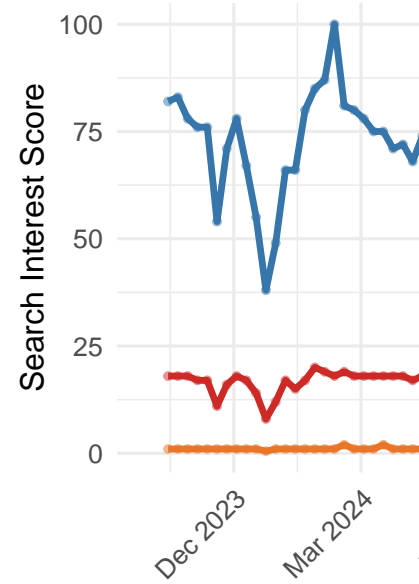
```

        "tableau" = "#E97627")) +
scale_x_date(date_breaks = "3 months", date_labels = "%b %Y") +
theme_minimal(base_size = 12) +
theme(
  legend.position = "bottom",
  plot.title = element_text(face = "bold", size = 16),
  axis.text.x = element_text(angle = 45, hjust = 1)
)

```

Recent Trends:

Focus period: 2023–1



Focused Google Search Results over the Last 24 and 12 Months of the Dataset

```

# Filter for last 12 months
recent_12mo <- trends_viz %>%
  filter(date >= max(date) - months(12))

# Plot last 12 months
ggplot(recent_12mo, aes(x = date, y = interest, color = skill_name)) +
  geom_line(linewidth = 1.0) +
  geom_point(size = 1.5) +
  labs(
    title = "Most Recent Trends: Last 12 Months",
    subtitle = paste("Focus period:", min(recent_12mo$date), "to", max(recent_12mo$date)),
    x = "Date",
    y = "Search Interest Score",
    color = "Skill"
  ) +
  scale_color_manual(values = c("python" = "#3776AB",
                                "sql" = "#CC2927",
                                "tableau" = "#E97627")) +

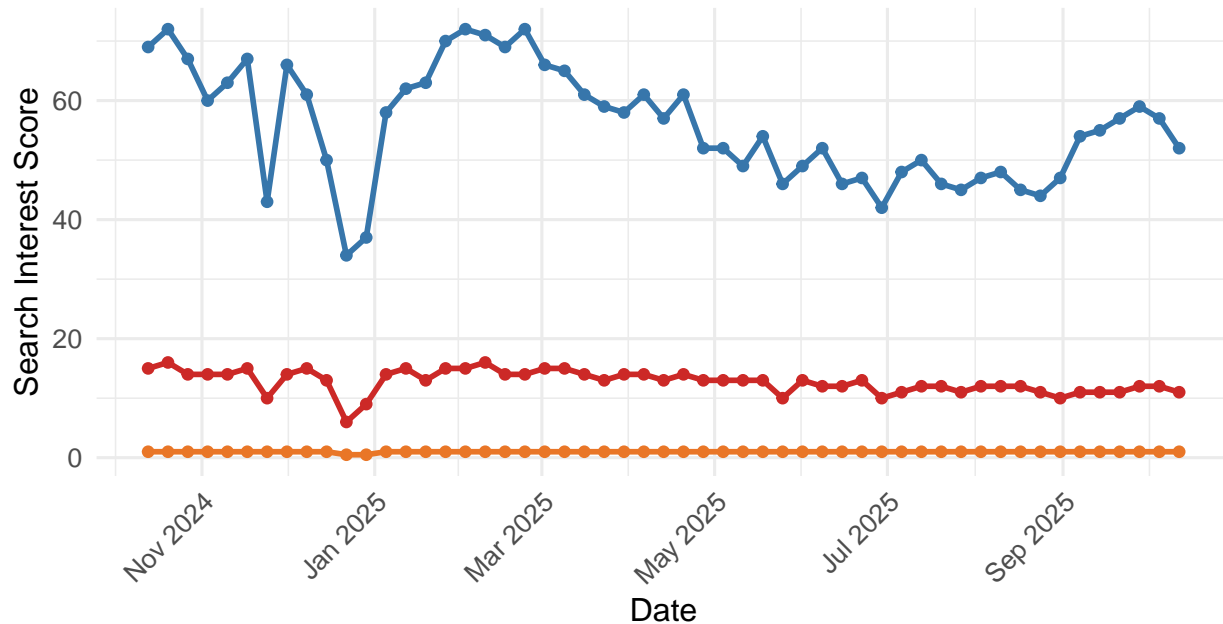
```



```
scale_x_date(date_breaks = "2 months", date_labels = "%b %Y") +
theme_minimal(base_size = 12) +
theme(
  legend.position = "bottom",
  plot.title = element_text(face = "bold", size = 16),
  axis.text.x = element_text(angle = 45, hjust = 1)
)
```

Most Recent Trends: Last 12 Months

Focus period: 2024-10-13 to 2025-10-12



Skill — python — sql — tableau

Volatility Analysis: Which Skill Shows the Most Fluctuation?

Key Insight Across Both Graphs: Python dominates but has transitioned from a volatile growth phase to a more stable maturity phase, while SQL and Tableau show no meaningful variation—they've essentially flat-lined at their respective low levels.

```
# Calculate volatility metrics
volatility_stats <- trends_viz %>%
  group_by(skill_name) %>%
  summarise(
    mean_interest = mean(interest),
    sd_interest = sd(interest),
    cv = sd_interest / mean_interest, # Coefficient of Variation
    min_interest = min(interest),
    max_interest = max(interest),
    range = max_interest - min_interest
  ) %>%
```

```
arrange(cv)

print(volatility_stats)
```

Volatility Metrics

```
## # A tibble: 3 x 7
##   skill_name mean_interest sd_interest   cv min_interest max_interest range
##   <chr>          <dbl>         <dbl> <dbl>         <dbl>         <dbl> <dbl>
## 1 sql             16.5           3.08  0.187           6           22    16
## 2 python          65.2          16.1  0.247          30          100   70
## 3 tableau         1.20           0.414 0.344           0.5           2     1.5
```

Visualizing Volatility

SQL - Most Stable Searches:

CV = 0.187 - The lowest coefficient of variation indicates people search for SQL in the most consistent, predictable patterns. With a mean of 16.5 and standard deviation of 3.1, SQL searches fluctuate modestly but stay within a narrow range (6-22).

Python - Moderately Stable Searches:

CV = 0.247 - Despite appearing volatile in absolute numbers, Python's search fluctuations are proportionate to its much higher average volume (65.2). Standard deviation of 16.1 means search activity varies significantly, but this is expected given Python's popularity. The wide range (30-100) shows dramatic peaks and valleys in how often people look up Python information.

Tableau - Least Stable Searches:

CV = 0.344 - The highest coefficient of variation reveals the most unpredictable search patterns relative to its size. With a tiny mean of 1.2, even small absolute changes (SD = 0.41) represent large percentage swings in search activity. Range of 0.5-2 shows search volume can double or halve frequently.

```
# Plot 1: Line chart with standard deviation bands
trends_with_sd <- trends_viz %>%
  group_by(skill_name) %>%
  mutate(
    rolling_mean = rollmean(interest, k = 4, fill = NA, align = "right"),
    rolling_sd = rollapply(interest, width = 4, FUN = sd, fill = NA, align = "right"),
    upper_band = rolling_mean + rolling_sd,
    lower_band = rolling_mean - rolling_sd
  ) %>%
  ungroup()

ggplot(trends_with_sd, aes(x = date, y = interest, color = skill_name, fill = skill_name)) +
  geom_line(aes(y = rolling_mean), linewidth = 1.2) +
  geom_ribbon(aes(ymin = lower_band, ymax = upper_band), alpha = 0.2, color = NA) +
  facet_wrap(~skill_name, ncol = 1, scales = "free_y") +
  labs(
    title = "Volatility Analysis: 4-Week Rolling Average with ±1 SD Bands",
    subtitle = "Wider bands indicate more volatility",
```

```

x = "Date",
y = "Search Interest Score",
fill = "Skill",
color = "Skill"
) +
scale_color_manual(values = c("python" = "#3776AB",
                              "sql" = "#CC2927",
                              "tableau" = "#E97627")) +
scale_fill_manual(values = c("python" = "#3776AB",
                              "sql" = "#CC2927",
                              "tableau" = "#E97627")) +
theme_minimal(base_size = 12) +
theme(
  legend.position = "none",
  plot.title = element_text(face = "bold", size = 14),
  strip.text = element_text(face = "bold", size = 12)
)

```

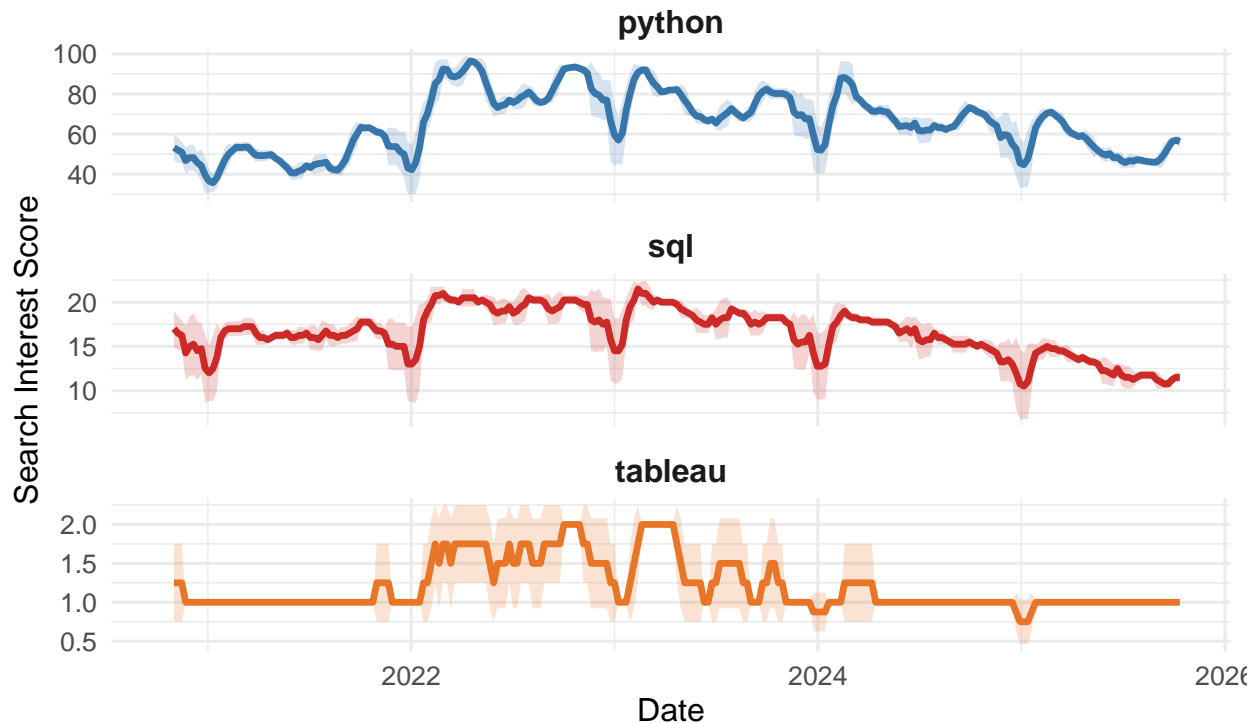
Standard Deviation Bands

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## ('geom_line()').
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## ('geom_ribbon()').
```

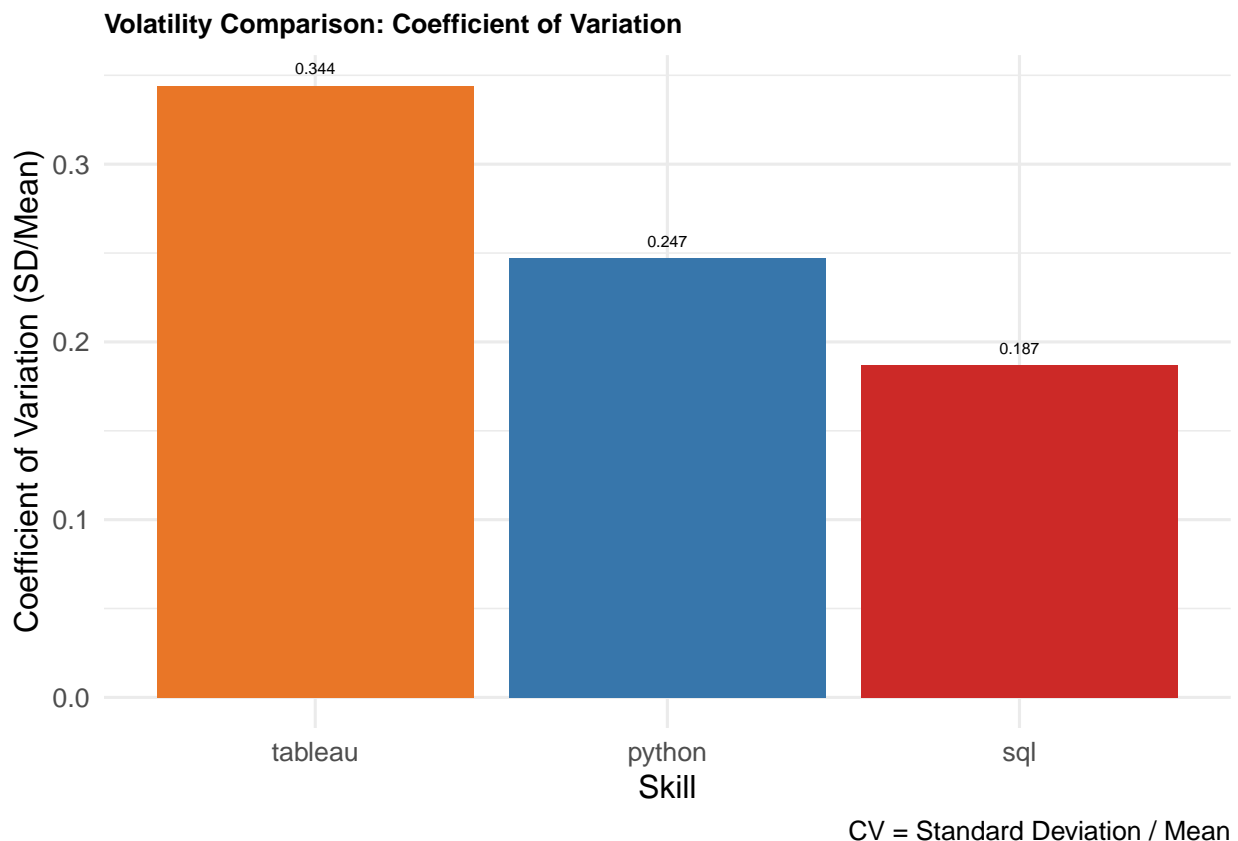
Volatility Analysis: 4–Week Rolling Average with ± 1 SD Bands

Wider bands indicate more volatility



```
#### Coefficient of Variation and Box Plots
```

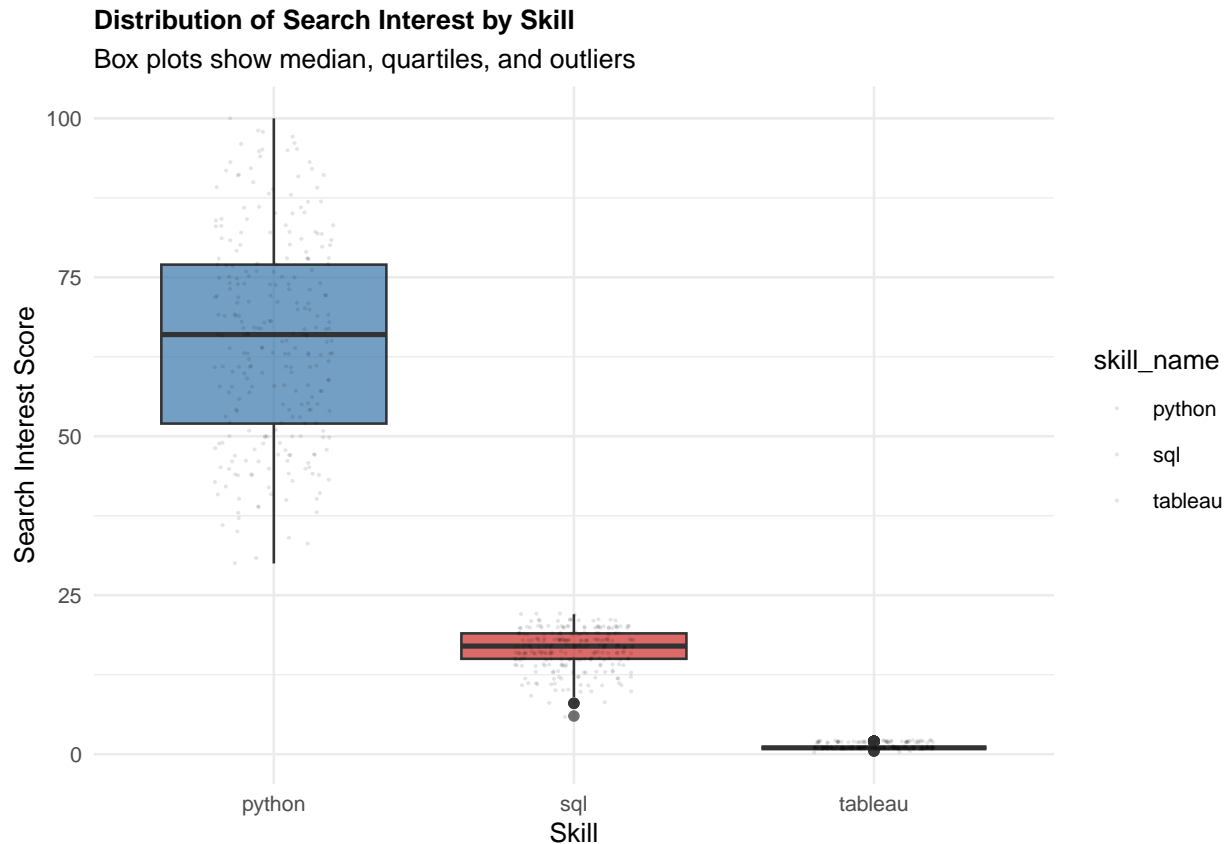
```
# Coefficient of Variation comparison
ggplot(volatility_stats, aes(x = reorder(skill_name, -cv), y = cv, fill = skill_name)) +
  geom_col(show.legend = FALSE) +
  geom_text(aes(label = round(cv, 3)), vjust = -1, size = 2) + # Change from -0.5 to -1 +
  labs(
    title = "Volatility Comparison: Coefficient of Variation",
    x = "Skill",
    y = "Coefficient of Variation (SD/Mean)",
    caption = "CV = Standard Deviation / Mean"
  ) +
  scale_fill_manual(values = c("python" = "#3776AB",
                                "sql" = "#CC2927",
                                "tableau" = "#E97627")) +
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold", size = 10))
```



```
# Box plots showing distribution
ggplot(trends_viz, aes(x = skill_name, y = interest, fill = skill_name)) +
  geom_boxplot(show.legend = FALSE, alpha = 0.7) +
  geom_jitter(width = 0.2, alpha = 0.1, size = 0.1) +
  labs(
    title = "Distribution of Search Interest by Skill",
    subtitle = "Box plots show median, quartiles, and outliers",
    x = "Skill",
    y = "Search Interest Score"
  ) +
```

```
scale_fill_manual(values = c("python" = "#3776AB",
                             "sql" = "#CC2927",
                             "tableau" = "#E97627")) +

theme_minimal(base_size = 10) +
theme(plot.title = element_text(face = "bold", size = 10))
```



Interpreting Volatility Findings

Tableau exhibits the highest search volatility ($CV = 0.344$), indicating the most erratic and unpredictable search patterns. Python shows moderate volatility ($CV = 0.247$) despite high absolute search volumes, reflecting dynamic but proportionally stable interest. SQL demonstrates the lowest volatility ($CV = 0.187$), confirming it has the most consistent, predictable search behavior among all three skills.

```
volatility_stats %>%
  mutate(
    stability_rank = rank(cv),
    interpretation = case_when(
      cv < 0.3 ~ "Very Stable",
      cv < 0.5 ~ "Moderately Stable",
      cv < 0.7 ~ "Moderate Volatility",
      TRUE ~ "High Volatility"
    )
  ) %>%
  select(skill_name, mean_interest, sd_interest, cv, interpretation, stability_rank) %>%
```

```

arrange(stability_rank) %>%
knitr::kable(
  col.names = c("Skill", "Mean Interest", "Std Dev", "CV", "Interpretation", "Rank"),
  digits = 2,
  caption = "Volatility Rankings (1 = Most Stable)"
)

```

Volatility States summary table

Table 3: Volatility Rankings (1 = Most Stable)

Skill	Mean Interest	Std Dev	CV	Interpretation	Rank
sql	16.50	3.08	0.19	Very Stable	1
python	65.18	16.10	0.25	Very Stable	2
tableau	1.20	0.41	0.34	Moderately Stable	3

```

# Prepare Python data
python_data <- trends_viz %>%
  filter(skill_name == "python") %>%
  arrange(date)

# Fit linear model on FULL data
python_data$time_index <- 1:nrow(python_data)
lm_model <- lm(interest ~ time_index, data = python_data)

# Print model summary
cat("\n=== Linear Model Summary ===\n")

```

Python's search interest 6 month forecast

```

##
## === Linear Model Summary ===

```

```
summary(lm_model)
```

```

##
## Call:
## lm(formula = interest ~ time_index, data = python_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -34.063 -13.122  -0.042  12.223  34.095
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  63.00184   1.99317   31.609  <2e-16 ***
## time_index    0.01659   0.01314    1.263    0.208
## ---

```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.08 on 260 degrees of freedom
## Multiple R-squared:  0.006093,    Adjusted R-squared:  0.002271
## F-statistic: 1.594 on 1 and 260 DF,  p-value: 0.2079
```

```
# Extract key stats
slope <- coef(lm_model)[2]
r_squared <- summary(lm_model)$r.squared

cat("\nKey Findings:\n")
```

```
##
## Key Findings:
```

```
cat("- Growth rate:", round(slope, 3), "points per week\n")
```

```
## - Growth rate: 0.017 points per week
```

```
cat("- Model explains", round(r_squared * 100, 1), "% of variance (R2)\n")
```

```
## - Model explains 0.6 % of variance (R2)
```

```
# Forecast 26 weeks ahead
future_time <- (nrow(pyhton_data) + 1):(nrow(pyhton_data) + 26)
forecast_lm <- predict(lm_model,
                      newdata = data.frame(time_index = future_time),
                      interval = "prediction", level = 0.95)

forecast_df <- data.frame(
  date = seq(max(pyhton_data$date) + 7, by = "week", length.out = 26),
  forecast = forecast_lm[, "fit"],
  lower_95 = forecast_lm[, "lwr"],
  upper_95 = forecast_lm[, "upr"]
)
```

```
# Show only last 18-24 months + forecast for clarity
recent_pyhton <- pyhton_data %>%
  filter(date >= max(date) - months(18))
```

```
# Plot
ggplot() +
  geom_line(data = recent_pyhton, aes(x = date, y = interest),
            color = "gray50", linewidth = 1) +
  geom_point(data = recent_pyhton, aes(x = date, y = interest),
             color = "gray50", size = 1, alpha = 0.5) +
  geom_line(data = forecast_df, aes(x = date, y = forecast),
            color = "#3776AB", linewidth = 1.5, linetype = "solid") +
  geom_ribbon(data = forecast_df,
             aes(x = date, ymin = lower_95, ymax = upper_95),
             alpha = 0.3, fill = "#3776AB") +
```

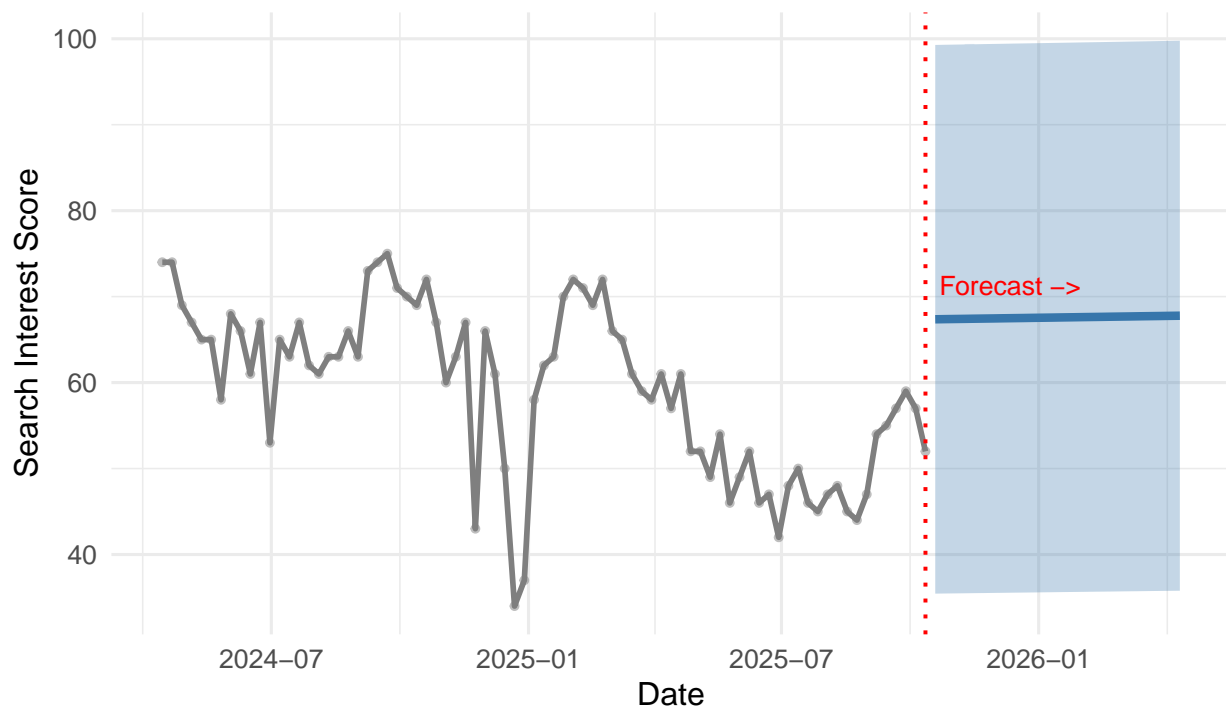
```

geom_vline(xintercept = max(python_data$date),
           linetype = "dotted", color = "red", linewidth = 0.7) +
annotate("text", x = max(python_data$date), y = max(recent_python$interest) * 0.95,
         label = "Forecast ->", hjust = -0.1, size = 3.5, color = "red") +
labs(
  title = "Python Search Interest: 6-Month Forecast",
  subtitle = sprintf("Based on %.1f years of data | Growth: %.3f pts/week | R² = %.1f%%",
                    as.numeric(max(python_data$date) - min(python_data$date))/365.25,
                    slope,
                    r_squared * 100),
  x = "Date",
  y = "Search Interest Score",
  caption = "Shaded area represents 95% prediction interval. Dotted line marks forecast start."
) +
theme_minimal(base_size = 12) +
theme(
  plot.title = element_text(face = "bold", size = 14),
  plot.subtitle = element_text(size = 10, color = "gray30")
)

```

Python Search Interest: 6-Month Forecast

Based on 5.0 years of data | Growth: 0.017 pts/week | R² = 0.6%



Shaded area represents 95% prediction interval. Dotted line marks forecast start.

```

# Print forecast summary
cat("\n=== 6-Month Forecast Summary ===\n")

```

```

##
## === 6-Month Forecast Summary ===

```



```
cat("Current interest (last obs):", round(tail(python_data$interest, 1), 1), "\n")
```

```
## Current interest (last obs): 52
```

```
cat("Forecast in 6 months:", round(tail(forecast_df$forecast, 1), 1), "\n")
```

```
## Forecast in 6 months: 67.8
```

```
cat("95% Prediction Interval: [",  
    round(tail(forecast_df$lower_95, 1), 1), ",",  
    round(tail(forecast_df$upper_95, 1), 1), "]\n")
```

```
## 95% Prediction Interval: [ 35.8 , 99.8 ]
```

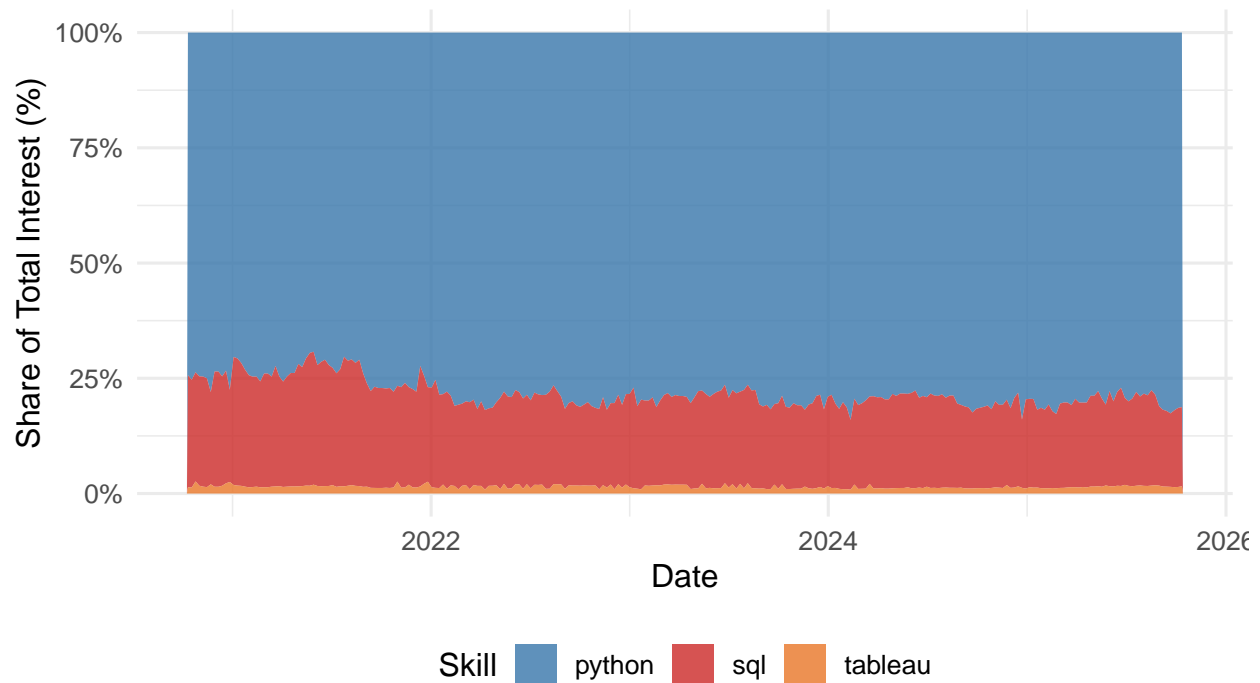
We're 95% confident the true value will be somewhere in this shaded area. The middle line is the most likely forecast for the next 26 weeks.

Analysis of Python's Dominance as a Percentage of Total Interest.

```
# The Growing Interest of Python as a Data Science Skill  
  
# Calculate trends  
market_share <- trends_viz %>%  
  group_by(date) %>%  
  mutate(  
    total_interest = sum(interest),  
    market_share = interest / total_interest * 100  
  ) %>%  
  ungroup()  
  
# Stacked area chart  
ggplot(market_share, aes(x = date, y = market_share, fill = skill_name)) +  
  geom_area(alpha = 0.8) +  
  labs(  
    title = "Python's Dominance as a percentage of Total Search Interest",  
    subtitle = "Percentage of total search interest across all three skills",  
    x = "Date",  
    y = "Share of Total Interest (%)",  
    fill = "Skill",  
    caption = "Python now represents over 80% of total search interest"  
  ) +  
  scale_fill_manual(values = c("python" = "#3776AB",  
                                "sql" = "#CC2927",  
                                "tableau" = "#E97627")) +  
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +  
  theme_minimal(base_size = 12) +  
  theme(  
    legend.position = "bottom",  
    plot.title = element_text(face = "bold", size = 14)  
  )
```

Python's Dominance as a percentage of Total Search Interest

Percentage of total search interest across all three skills



Python now represents over 80% of total search interest

```
# Summary table
market_share_summary <- market_share %>%
  mutate(year = year(date)) %>%
  group_by(skill_name, year) %>%
  summarise(avg_market_share = mean(market_share), .groups = "drop") %>%
  pivot_wider(names_from = year, values_from = avg_market_share)

market_share_summary %>%
  knitr::kable(
    digits = 1,
    caption = "Average Market Share by Year (%)"
  )
```

Table 4: Average Market Share by Year (%)

skill_name	2020	2021	2022	2023	2024	2025
python	74.8	73.9	79.5	79.3	79.9	80.0
sql	23.4	24.5	18.9	19.3	18.9	18.5
tableau	1.8	1.6	1.6	1.4	1.2	1.5

Peak Search History Comparison

```
# Peak Performance: Distance from All-Time Highs

# Find peaks and current values
```

```

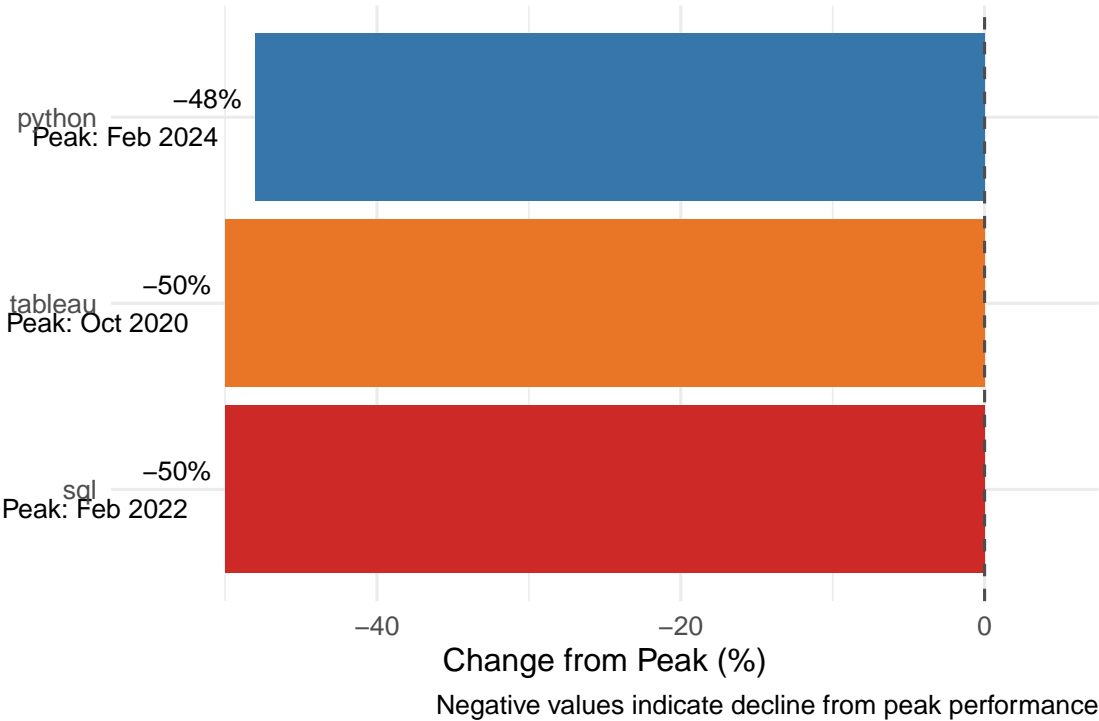
peak_comparison <- trends_viz %>%
  group_by(skill_name) %>%
  summarise(
    peak_interest = max(interest),
    peak_date = date[which.max(interest)],
    current_interest = last(interest),
    current_date = last(date),
    decline_from_peak = current_interest - peak_interest,
    pct_from_peak = (current_interest - peak_interest) / peak_interest * 100
  )

# Visualization
ggplot(peak_comparison, aes(x = reorder(skill_name, pct_from_peak),
                             y = pct_from_peak, fill = skill_name)) +
  geom_col(show.legend = FALSE) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray30") +
  geom_text(aes(label = paste0(round(pct_from_peak, 1), "%\n",
                                "Peak: ", format(peak_date, "%b %Y"))),
             hjust = ifelse(peak_comparison$pct_from_peak > 0, -0.2, 1.2),
             size = 3.5) +
  coord_flip(clip = "off") +
  labs(
    title = "Current Position Relative to Historical Peak",
    subtitle = "How does today's interest compare to all-time highs?",
    x = NULL,
    y = "Change from Peak (%)",
    caption = "Negative values indicate decline from peak performance"
  ) +
  scale_fill_manual(values = c("python" = "#3776AB",
                                "sql" = "#CC2927",
                                "tableau" = "#E97627")) +
  scale_y_continuous(expand = expansion(mult = c(0.15, 0.15))) +
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    plot.margin = margin(10, 50, 10, 10)
  )

```

Current Position Relative to Historical Peak

How does today's interest compare to all-time highs?



```
# Summary table
peak_comparison %>%
  knitr::kable(
    col.names = c("Skill", "Peak Interest", "Peak Date", "Current Interest",
                  "Current Date", "Change", "% from Peak"),
    digits = 1,
    caption = "Peak vs. Current Performance Metrics"
  )
```

Table 5: Peak vs. Current Performance Metrics

Skill	Peak Interest	Peak Date	Current Interest	Current Date	Change	% from Peak
python	100	2024-02-11	52	2025-10-12	-48	-48
sql	22	2022-02-06	11	2025-10-12	-11	-50
tableau	2	2020-10-25	1	2025-10-12	-1	-50

Current Search Interest vs. All-Time Highs

- All three data science skills are currently well below their historical peak search interest levels.
- Python has declined 48% from its February 2024 peak, when search interest reached 100, settling now at around 52.
- SQL and Tableau have both dropped approximately 50% from their respective peaks—SQL from its February 2022 high and Tableau from October 2020.
- This across-the-board decline from historical peaks suggests the market is undergoing a correction or normalization period, possibly reflecting a return to more sustainable baseline interest levels after the exceptional surge in online learning and tech skill development during 2020-2024.

- While these declines appear dramatic, they likely indicate market maturation rather than obsolescence—these skills remain relevant but are no longer experiencing the heightened search activity of earlier pandemic and post-pandemic periods.

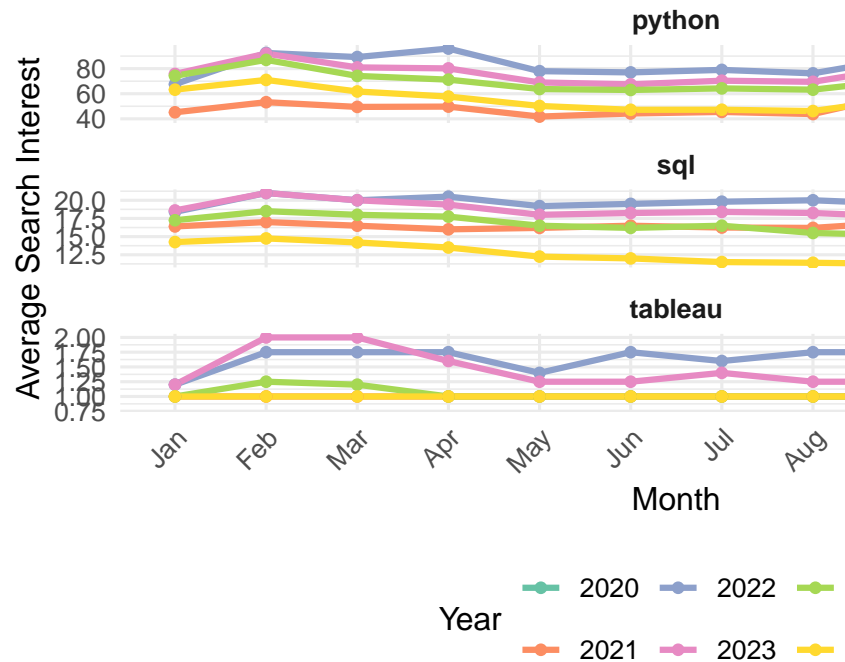
```
# Seasonal Patterns: Do Skills Show Predictable Cycles?

# Year-over-Year comparison
trends_yoy <- trends_viz %>%
  mutate(
    year = year(date),
    month = month(date, label = TRUE, abbr = TRUE)
  ) %>%
  group_by(skill_name, year, month) %>%
  summarise(avg_interest = mean(interest), .groups = "drop")

# Faceted line plot
ggplot(trends_yoy, aes(x = month, y = avg_interest,
                      color = as.factor(year), group = year)) +
  geom_line(linewidth = 1.1) +
  geom_point(size = 1.5) +
  facet_wrap(~skill_name, ncol = 1, scales = "free_y") +
  labs(
    title = "Seasonal Patterns: Month-by-Month Comparison Across Years",
    subtitle = "Do skills show consistent seasonal trends?",
    x = "Month",
    y = "Average Search Interest",
    color = "Year",
    caption = "Each line represents one calendar year"
  ) +
  scale_color_brewer(palette = "Set2") +
  theme_minimal(base_size = 12) +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    legend.position = "bottom",
    axis.text.x = element_text(angle = 45, hjust = 1),
    strip.text = element_text(face = "bold", size = 10)
  )
)
```

Seasonal Patterns: Month-by-Month Comparison

Do skills show consistent seasonal trends?



Seasonal Patterns Analysis for Search Terms

```
# Calculate seasonality index (coefficient of variation by month)
seasonality_stats <- trends_viz %>%
  mutate(month = month(date, label = TRUE)) %>%
  group_by(skill_name, month) %>%
  summarise(
    mean_interest = mean(interest),
    sd_interest = sd(interest),
    cv = sd_interest / mean_interest,
    .groups = "drop"
  ) %>%
  group_by(skill_name) %>%
  summarise(
    avg_monthly_cv = mean(cv),
    seasonality = ifelse(avg_monthly_cv > 0.3, "High",
                        ifelse(avg_monthly_cv > 0.15, "Moderate", "Low")),
    .groups = "drop"
  )

seasonality_stats %>%
  knitr::kable(
    col.names = c("Skill", "Avg Monthly CV", "Seasonality Level"),
    digits = 3,
    caption = "Seasonality Assessment by Skill"
  )
```

Table 6: Seasonality Assessment by Skill

Skill	Avg Monthly CV	Seasonality Level
python	0.226	Moderate
sql	0.175	Moderate
tableau	0.328	High

Seasonal Trends: Predictable Cycles or Random Fluctuations?

- The month-by-month comparison across years reveals no consistent seasonal patterns for any of the three skills. None of the skills show strong, consistent seasonal patterns that repeat predictably year after year. The dominant pattern is year-over-year decline rather than seasonal cyclicalities, indicating people’s search timing is not heavily influenced by academic calendars, hiring cycles, or other seasonal factors—instead, the overall market interest is simply declining over time.