Startup Profitability and Cost Analysis

📥 Load Packages & Data

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(tidyr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
setwd("C:/Users/LENOVO/Downloads")
data <- read_csv("food_delivery_startup_2024_fictional_data.csv") %>%
  janitor::clean_names()
```

```
## Rows: 12 Columns: 15
```

```
## ── Column specification
────────────────────────────────────────────────────────
```

```
## Delimiter: ","
## chr  (1): Month
## dbl (14): Number_of_Orders, Avg_Order_Value (PKR), Delivery_Fees_Collected
(...
##
```

```
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
month_levels <- c("Jan 2024", "Feb 2024", "Mar 2024", "Apr 2024",
                  "May 2024", "Jun 2024", "Jul 2024", "Aug 2024",
                  "Sep 2024", "Oct 2024", "Nov 2024", "Dec 2024")
data$month <- factor(data$month, levels = month_levels, ordered = TRUE)
```

Calculations

```
data <- data %>%
  mutate(
    total_revenue_calculated = subscription_revenue_pkr +
commission_revenue_pkr +
      delivery_fees_collected_pkr + in_app_ads_revenue_pkr,
    total_costs_calculated = fixed_costs_pkr + delivery_partner_payouts_pkr +
      marketing_spend_pkr + packaging_costs_pkr +
customer_support_refunds_pkr,
    profit_calculated = total_revenue_calculated - total_costs_calculated,
    profit_margin = round(profit_calculated / total_revenue_calculated, 3),
    cumulative_profit = cumsum(profit_calculated),
    revenue_growth_pct = round((total_revenue_calculated /
lag(total_revenue_calculated) - 1) * 100, 2)
  )
```

Summary Statistics

```
total_profit <- sum(data$profit_calculated)
max_row <- data[which.max(data$profit_calculated), ]
break_even <- data %>% filter(profit_calculated > 0) %>% slice(1)
avg_margin <- mean(data$profit_margin, na.rm = TRUE)
sd_profit <- sd(data$profit_calculated)

data.frame(
  `Total Annual Profit (PKR)` = total_profit,
  `Highest Profit Month` = max_row$month,
  `Highest Profit (PKR)` = max_row$profit_calculated,
  `Avg. Monthly Profit Margin (%)` = round(avg_margin * 100, 2),
  `Break-even Month` = break_even$month,
  `Profit SD (PKR)` = round(sd_profit, 2)
)
```

```
##   Total.Annual.Profit..PKR. Highest.Profit.Month Highest.Profit..PKR.
## 1                    899983             Jun 2024               280150
##   Avg..Monthly.Profit.Margin.... Break.even.Month Profit.SD..PKR.
## 1                           4.57         Jan 2024          145309
```

📈 Monthly Revenue Growth (%)

```
library(knitr)
kable(data %>% select(month, revenue_growth_pct), caption = "Revenue Growth % by Month")
```

Revenue Growth % by Month

month	revenue_growth_pct
Jan 2024	NA
Feb 2024	17.64
Mar 2024	-3.52
Apr 2024	-3.75
May 2024	10.62
Jun 2024	6.38
Jul 2024	-20.69
Aug 2024	2.15
Sep 2024	-15.07
Oct 2024	31.49
Nov 2024	0.14
Dec 2024	-21.28

💰 Cumulative Profit

```
kable(data %>% select(month, cumulative_profit), caption = "Cumulative Profit by Month")
```

Cumulative Profit by Month

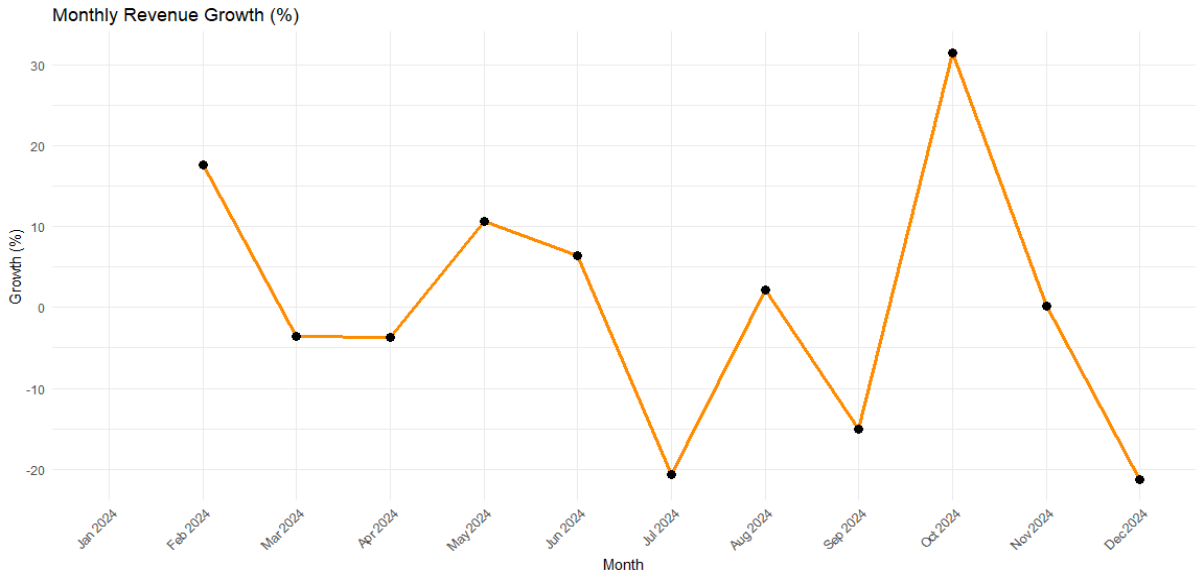
month	cumulative_profit
Jan 2024	47357
Feb 2024	275894
Mar 2024	416682
Apr 2024	582538
May 2024	758849
Jun 2024	1038999
Jul 2024	989106
Aug 2024	912155
Sep 2024	769445
Oct 2024	942833
Nov 2024	1041053

month	cumulative_profit
Dec 2024	899983

Visualizations

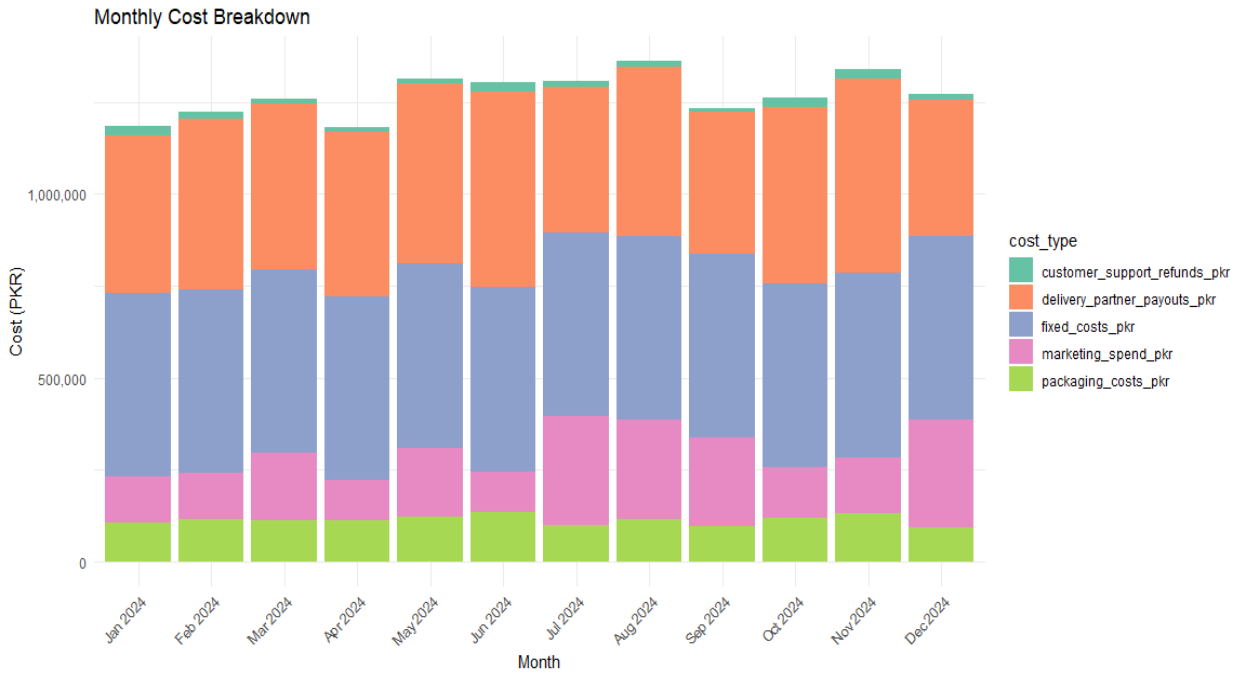
(a) Revenue Growth %

```
ggplot(data, aes(x = month, y = revenue_growth_pct, group = 1)) +
  geom_line(color = "darkorange", linewidth = 1.2) +
  geom_point(color = "black", size = 3) +
  labs(title = "Monthly Revenue Growth (%)", x = "Month", y = "Growth (%)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



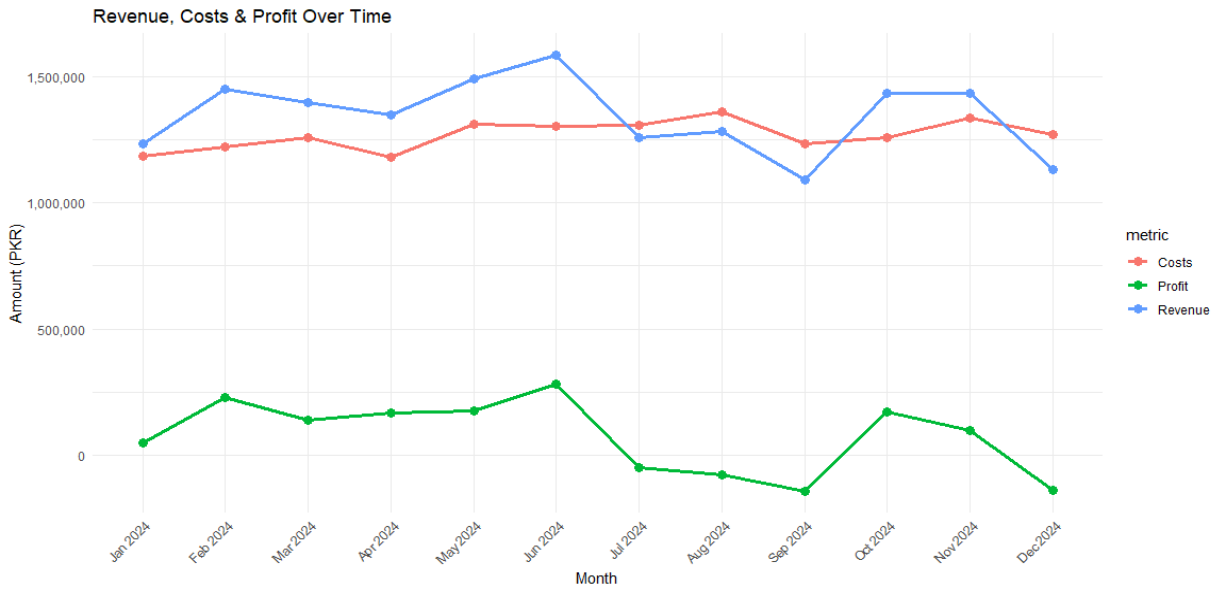
(b) Monthly Cost Breakdown

```
cost_data <- data %>%
  select(month, fixed_costs_pkr, delivery_partner_payouts_pkr,
         marketing_spend_pkr, packaging_costs_pkr,
         customer_support_refunds_pkr) %>%
  pivot_longer(-month, names_to = "cost_type", values_to = "amount")

cost_data$month <- factor(cost_data$month, levels = month_levels, ordered =
TRUE)

ggplot(cost_data, aes(x = month, y = amount, fill = cost_type)) +
  geom_bar(stat = "identity") +
  labs(title = "Monthly Cost Breakdown", x = "Month", y = "Cost (PKR)") +
  scale_y_continuous(labels = comma) +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



(c) Revenue, Costs, Profit Trend

```
line_data <- data %>%
  select(month, total_revenue_calculated, total_costs_calculated,
profit_calculated) %>%
  rename(Revenue = total_revenue_calculated,
         Costs = total_costs_calculated,
         Profit = profit_calculated) %>%
  pivot_longer(-month, names_to = "metric", values_to = "value")

ggplot(line_data, aes(x = month, y = value, color = metric, group = metric))
+
  geom_line(linewidth = 1.2) +
  geom_point(size = 3) +
  labs(title = "Revenue, Costs & Profit Over Time", x = "Month", y = "Amount
(PKR)") +
  scale_y_continuous(labels = comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



(d) Profit Trend + Break-even Annotation

```
ggplot(data, aes(x = month, y = profit_calculated, group = 1)) +
  geom_line(color = "blue", linewidth = 1.2) +
  geom_point(color = "darkgreen", size = 3) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_text(data = break_even, aes(label = "Break-even", y =
profit_calculated + 100000),
            color = "purple", size = 4, vjust = -1) +
  labs(title = "Monthly Profit Trend", x = "Month", y = "Profit (PKR)") +
  scale_y_continuous(labels = comma) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```