

DATA 605 HW 3

Kevin Havis

April 10, 2025

```
library(ggplot2)
library(tidyverse)
library(glue)
library(pracma)
library(plotly)
```

Problem 1: Transportation Safety

Scenario

You are a data analyst at a transportation safety organization. Your task is to analyze the relationship between the speed of cars and their stopping distance using the built-in R dataset `cars`. This analysis will help in understanding how speed affects the stopping distance, which is crucial for improving road safety regulations.

Tasks

We can load the built-in `cars` data and plot it using `ggplot`, drawing a regression line through the data points.

Data Visualization

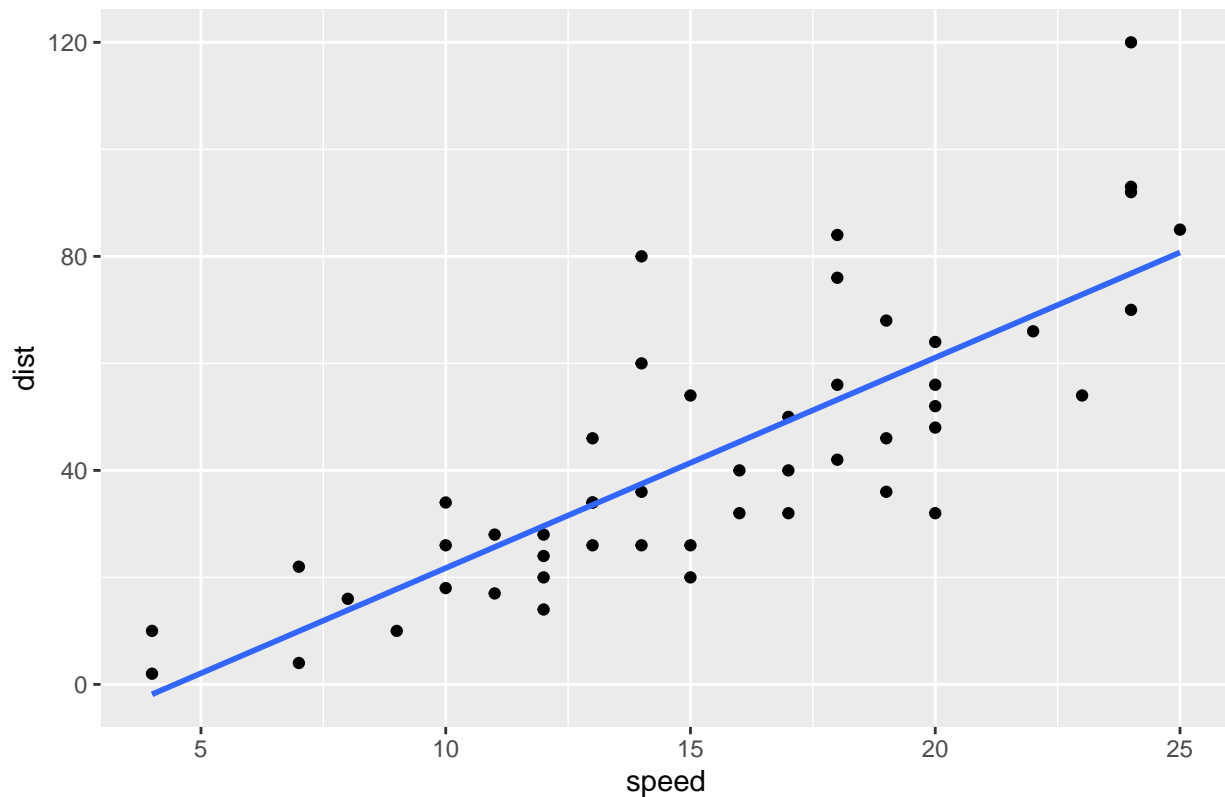
```
# Load the data
data(cars)

# Create a scatter plot of stopping distance (dist) as a function of speed (speed)

plot <- cars |>
  ggplot(aes(speed, dist)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE) + # Add a regression line to the plot
  ggtitle("Stopping Distance vs Speed")
print(plot)

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

Stopping Distance vs Speed



Build a Linear Model

Now that we know our data is roughly linear, we can create a regression model to predict `dist` given `speed`.

```
# Construct a simple linear regression model
model <- lm(dist ~ speed, data=cars)
```

```
# Summarize the model
summary(model)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601  0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
```

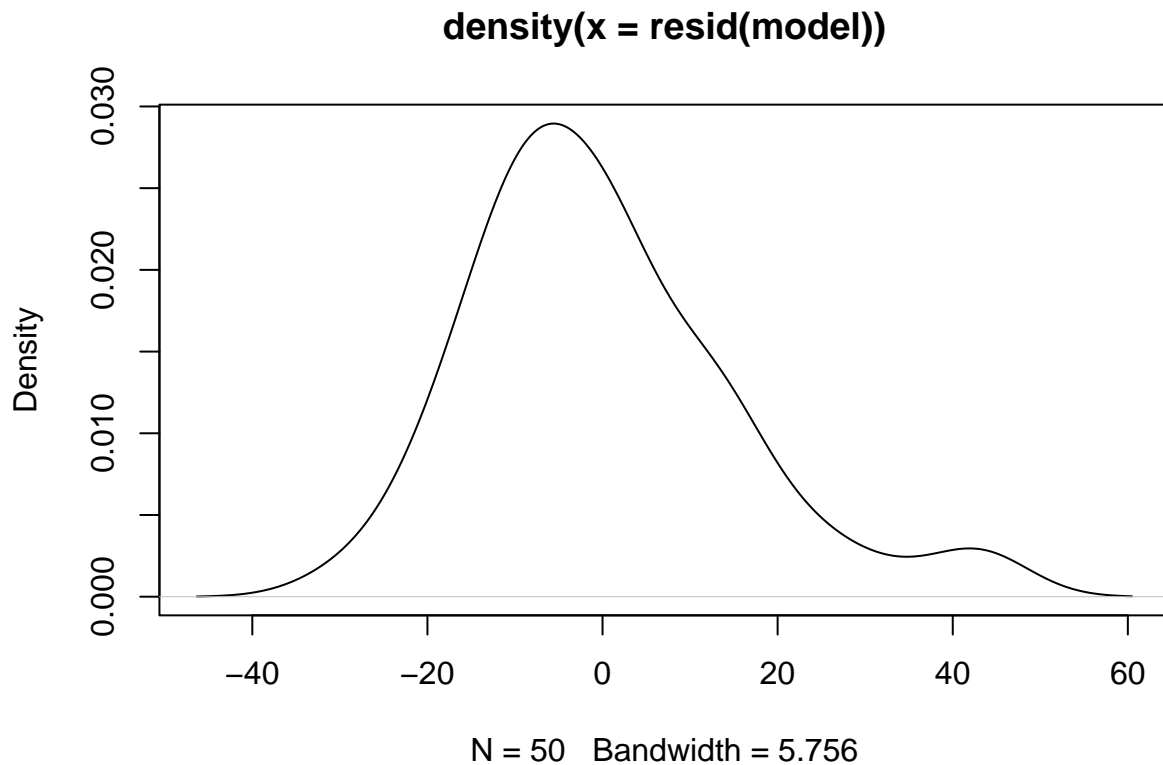
```
## F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12
```

Model Quality Evaluation

```
# Calculate and interpret the R-squared value  
summary(model)$r.squared
```

```
## [1] 0.6510794
```

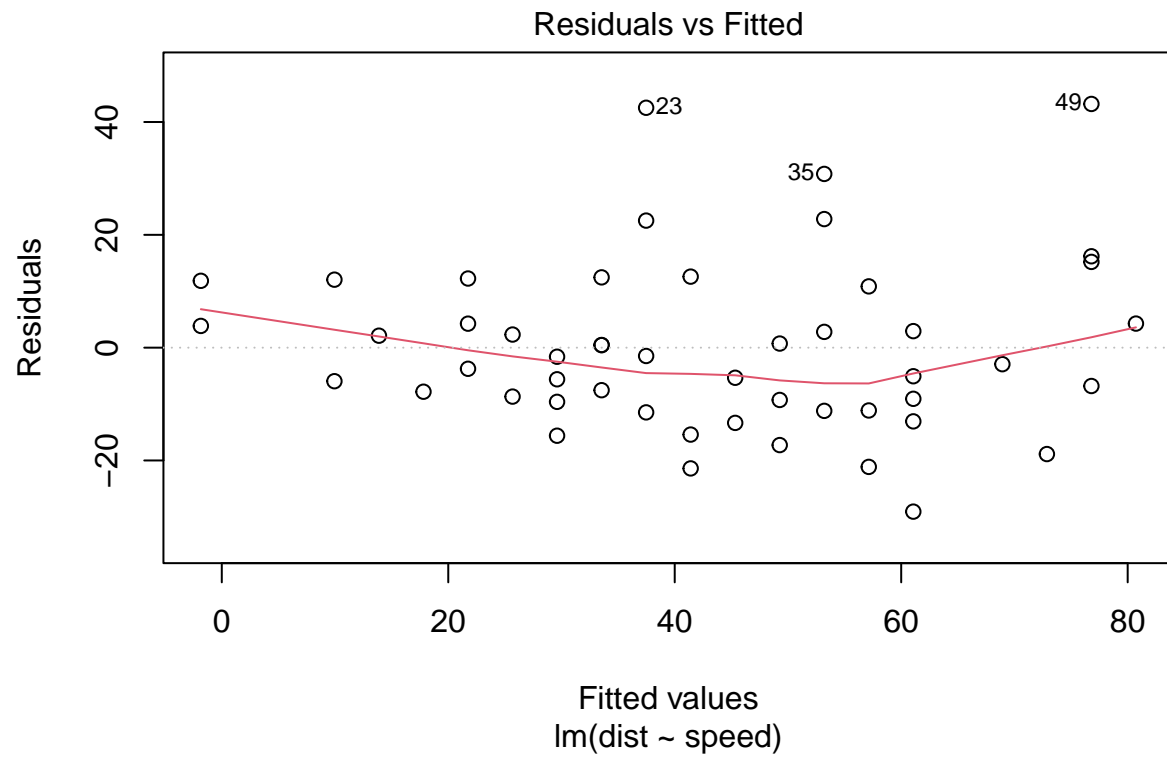
```
# Perform a residual analysis  
plot(density(resid(model)))
```



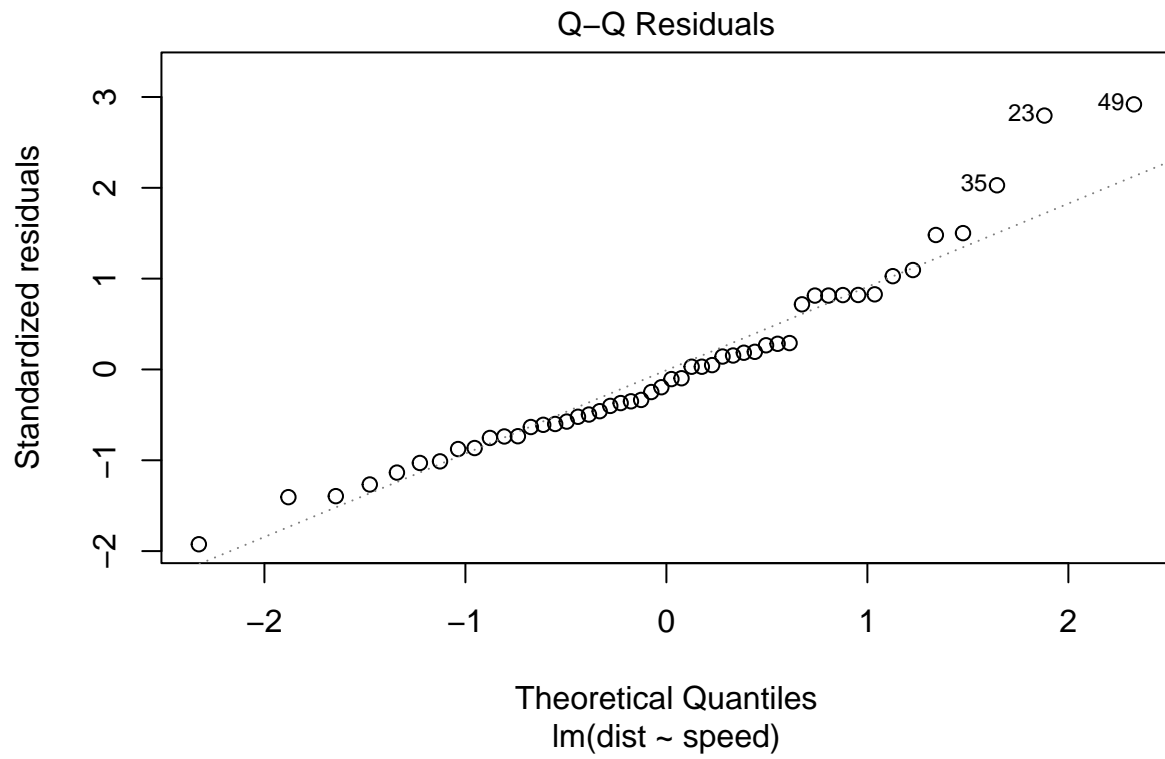
Good news! Our R^2 is 65.1%, which means our model has identified a strong correlation between **speed** and **dist**. Our residuals are normally distributed, meaning our simple model accounts for much of the variability between points. The linear model seems to be a good fit for the data.

Residual Analysis

```
# Plot residuals versus fitted values  
plot(model, 1)
```



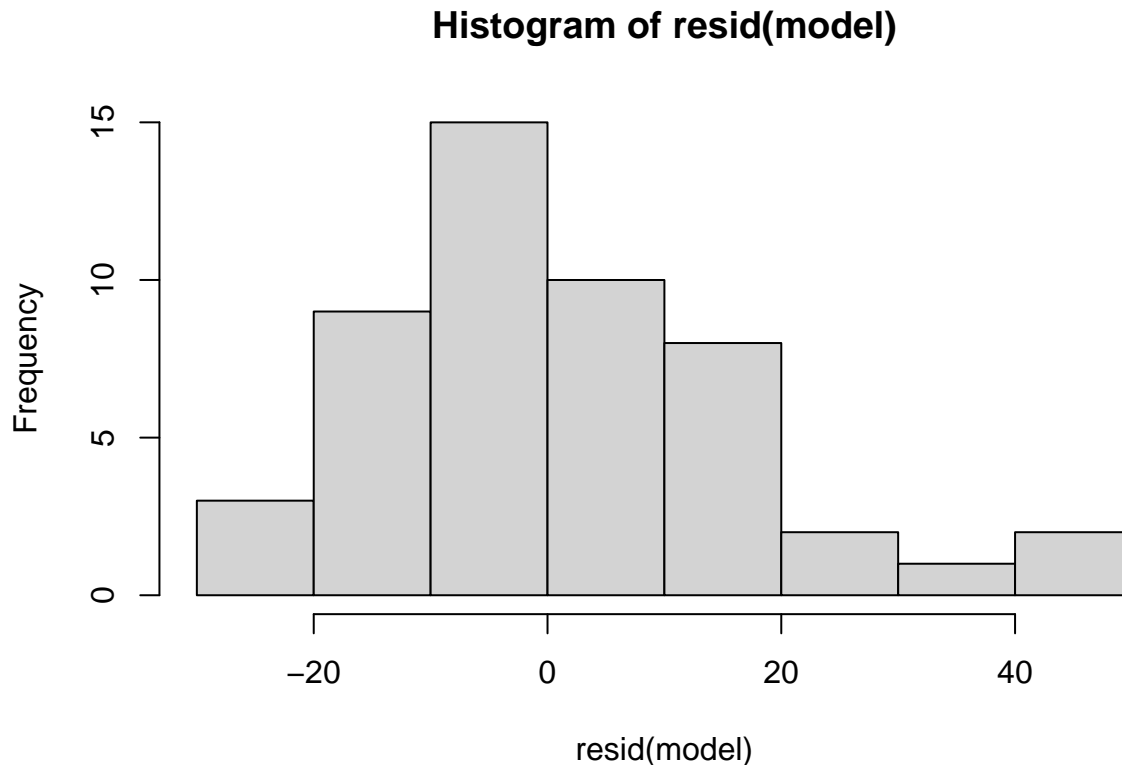
```
# Create Q-Q plot of residuals  
plot(model, 2)
```



```
# Perform Shapiro Wilkes test  
shapiro.test(model$residuals)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  model$residuals  
## W = 0.94509, p-value = 0.02152
```

```
# Plot histogram of residuals  
plot(hist(resid(model)))
```



These plots give us additional views of the residuals, which is important when analyzing linear models.

We should note that our Shapiro-Wilkes test results show a high W statistic, indicating high normality, however the p-value is below 0.05, indicating that there is statistically significant deviation from normality. Given we can visually confirm that the residuals are nearly normal, I would feel comfortable with these results in spite of the p-value, but it should be noted and disclosed.

If we decided we were uncomfortable with the results of the Test, we could examine the skew for potential outliers, however we should be very careful in choosing to eliminate them - this would be a classic example of p-hacking.

Conclusion

Write your conclusion here discussing model appropriateness, assumptions, and potential improvements.

Our model shows about 65% of the variance in stopping distance is explained by increases in speed, which is a very good indicator of a positive linear relationship.

Additionally, when reviewing our residuals (both in the QQ plot and histogram), we can see that they are well distributed about zero. This is also a good indicator of a high performing linear model; we expect this type of random variance to have a mean of zero.

Regarding improvements, there are a few outliers near zero I can see that may be skewing the regression. To improve generalized performance, it may be worth removing those and running the model again.

Problem 2: Health Policy Analysis

Scenario

As a health policy analyst for an international organization, you are tasked with analyzing data from the World Health Organization (WHO) to inform global health policies. The dataset provided (who.csv) contains crucial health indicators for various countries from the year 2008.

Question 1: Health Policy Analyst

```
# Import the WHO dataset
who <- read_csv("who.csv")

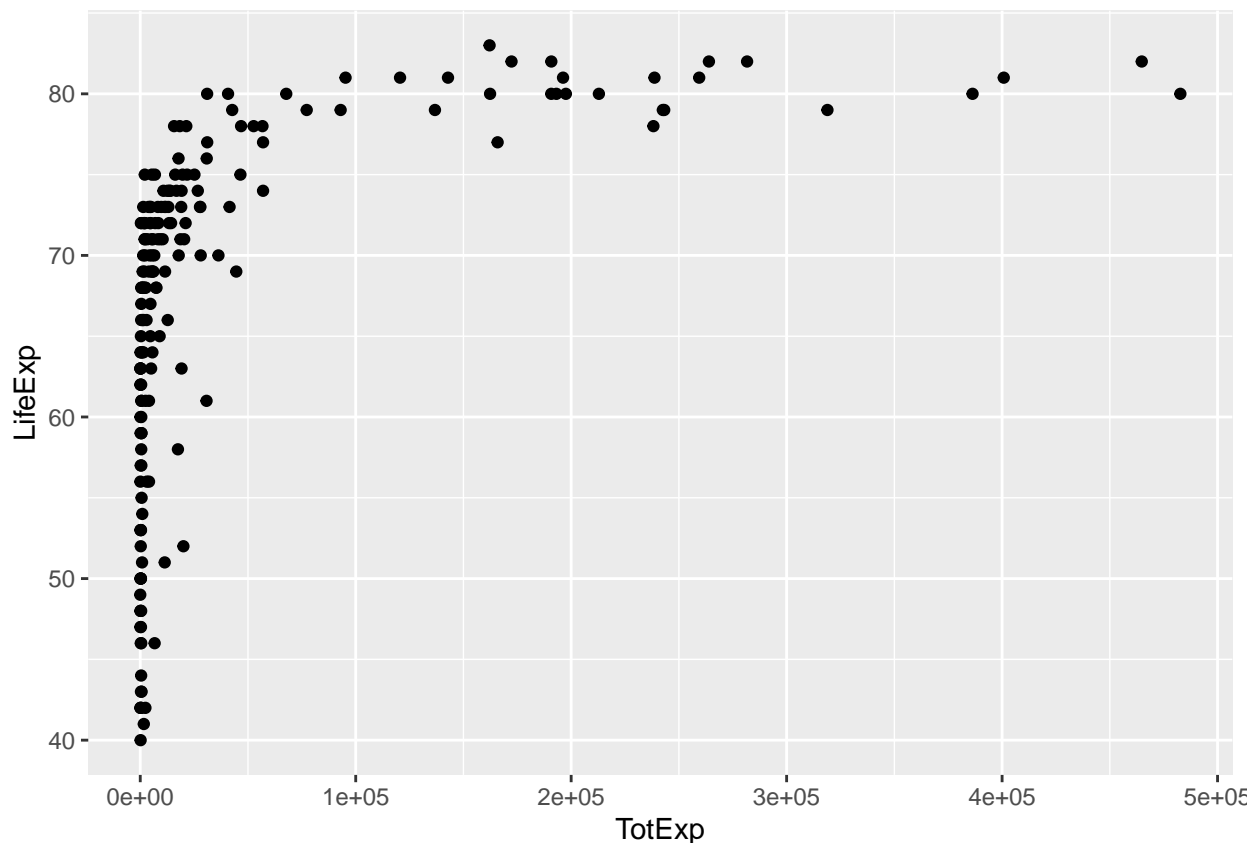
## New names:
## Rows: 190 Columns: 12
## -- Column specification
## ----- Delimiter: "," chr
## (1): Country dbl (10): LifeExp...2, InfantSurvival, Under5Survival, TBFree,
## PropMD, PropR... lgl (1): ...10
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `LifeExp` -> `LifeExp...2`
## * `` -> `...10`
## * `LifeExp` -> `LifeExp...12`

# Remove empty column
who <- who |>
  select(!c("...10", "LifeExp...12")) |>
  rename(LifeExp = LifeExp...2)

# Create a scatterplot of LifeExp vs. TotExp

plot <- who |>
  ggplot(aes(TotExp, LifeExp)) +
  geom_point()

print(plot)
```



Visualizing our data, we can see that it is not linearly distributed. This will be important to consider in a bit.

```
# Run a simple linear regression model
```

```
who_model <- lm(LifeExp ~ TotExp, data=who)
```

```
summary(who_model)
```

```
##
```

```
## Call:
```

```
## lm(formula = LifeExp ~ TotExp, data = who)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -24.764  -4.778   3.154   7.116  13.292
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 6.475e+01  7.535e-01  85.933  < 2e-16 ***
```

```
## TotExp      6.297e-05  7.795e-06   8.079  7.71e-14 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 9.371 on 188 degrees of freedom
```

```
## Multiple R-squared:  0.2577, Adjusted R-squared:  0.2537
```

```
## F-statistic: 65.26 on 1 and 188 DF, p-value: 7.714e-14
```

Interpret F-statistic, R-squared value, standard error, and p-values here.

- F-statistic: The F-statistic measures the ratio of variance, which we expect to be close to 1 if there is no

relationship. Since we see a high F-statistic, and its corresponding p-value is very low, we can say that there is some relationship between **LifeExp** and **TotExp**

- R^2 : The R^2 is positive and around 25%, which indicates that the linear model is showing some correlation, but it does not well explain the data. This intuitively makes sense given the scatterplot we observed before.
- p-value: We do see a statistical significance to the correlation between the two variables; even though the model explains a small amount of the variance, it likely a real relationship.

Discuss whether regression assumptions are met.

The data clearly that the relationship is not linear in the data's current state. We also are unlikely to see homoscedasticity (normally distributed residuals) were we to plot them given the shape of the data.

Discuss implications for health policy.

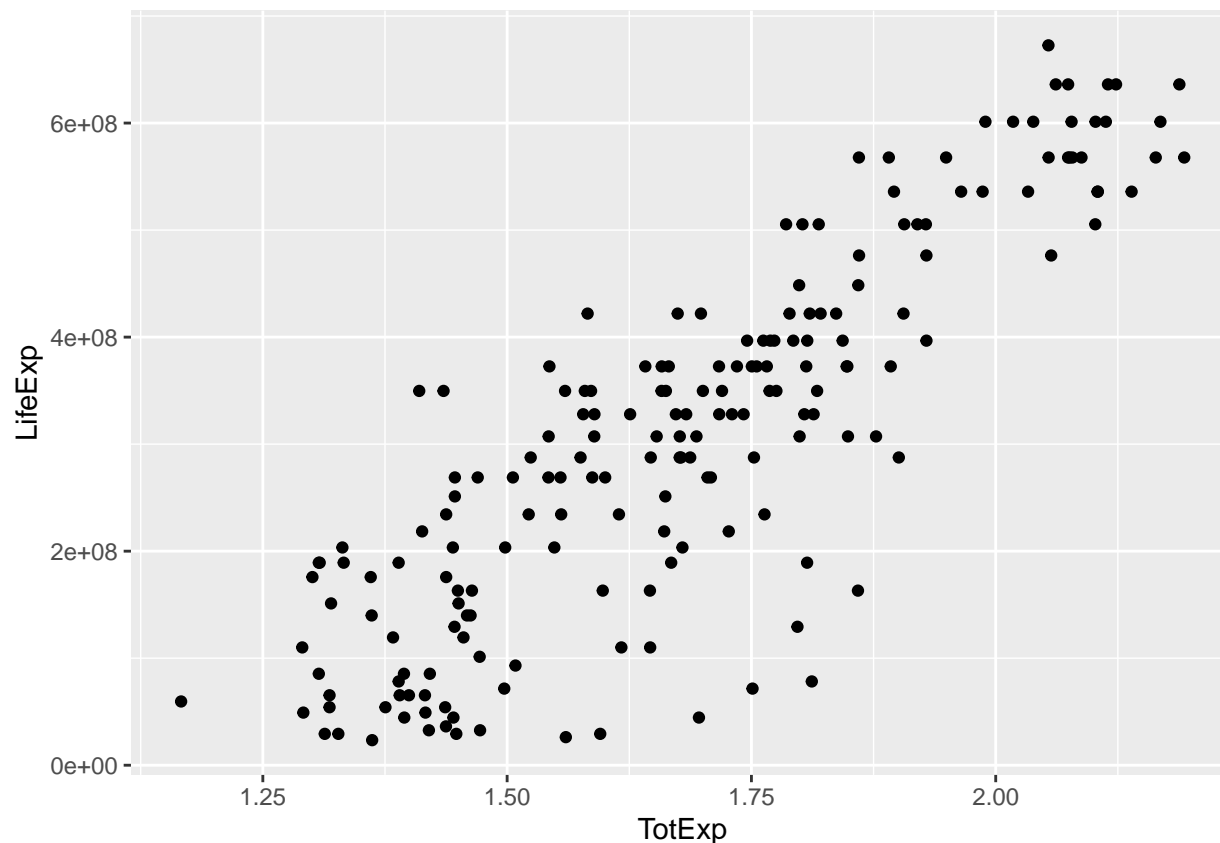
We can confirm that there is a positive relationship between increasing life expectancy and increased expenditures, however our model is not robust enough to accurately estimate expenditure costs. We will need to improve the model with some data preprocessing.

Question 2: Transforming Variables for a Better Fit

A simple exponential transformation should drastically improve our model.

```
# Transform variables: LifeExp^4.6 and TotExp^0.06
who_2 <- who |>
  mutate(TotExp = TotExp^0.06) |>
  mutate(LifeExp = LifeExp^4.6)

# Create a scatterplot with the transformed variables
ggplot(who_2, aes(TotExp, LifeExp)) +
  geom_point()
```



```
# Run regression with transformed variables
who_model_2 <- lm(LifeExp ~ TotExp, data=who_2)
summary(who_model_2)
```

```
##
## Call:
## lm(formula = LifeExp ~ TotExp, data = who_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -308616089  -53978977   13697187   59139231  211951764
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -736527910   46817945  -15.73  <2e-16 ***
## TotExp       620060216   27518940   22.53  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 90490000 on 188 degrees of freedom
## Multiple R-squared:  0.7298, Adjusted R-squared:  0.7283
## F-statistic: 507.7 on 1 and 188 DF, p-value: < 2.2e-16
```

Compare models and discuss which provides better fit.

We can see that after our transformations, our linear model is a far better fit for the data. For one metric, our R^2 value has increased from 25% to 73%!

Discuss impact of transformations on interpretation.

By transforming our data, we've been able to keep our simple linear regression model, as well as drastically improve its performance. For such a simple transformation, this is well worth it, although we should remember to invert the operation when communicating our prediction targets.

Question 3: Forecasting Life Expectancy Based on Transformed Expenditures

```
# Forecast life expectancy for specified transformed values
```

```
X_test <- tibble(TotExp = c(1.5, 2.5))
y_test <- predict(who_model_2, X_test)
y_test <- y_test ^ (1/4.6)
print(y_test)
```

```
##           1           2
## 63.31153 86.50645
```

Discuss implications of these forecasts.

With this improvement to our model, we are now in a much better position to inform the WHO predicted life expectancy at different budgets. It seems additional funding has the most impact after the age of 50, up to a plateau of 80.

Question 4: Interaction Effects in Multiple Regression

```
# Build multiple regression model with interaction
```

```
who_model_3 <- lm(LifeExp ~ PropMD + TotExp + (PropMD*TotExp), who_2)
summary(who_model_3)
```

```
##
## Call:
## lm(formula = LifeExp ~ PropMD + TotExp + (PropMD * TotExp), data = who_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -296470018 -47729263  12183210  60285515 212311883
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.244e+08  5.083e+07 -14.253  <2e-16 ***
## PropMD       4.727e+10  2.258e+10   2.094   0.0376 *
## TotExp       6.048e+08  3.023e+07  20.005  <2e-16 ***
## PropMD:TotExp -2.121e+10  1.131e+10  -1.876   0.0622 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 88520000 on 186 degrees of freedom
## Multiple R-squared:  0.7441, Adjusted R-squared:  0.74
## F-statistic: 180.3 on 3 and 186 DF, p-value: < 2.2e-16
```

Interpret F-statistic, R-squared value, standard error, and p-values.

These additional features slightly improve our R^2 metric by approximately 2%

Evaluate the interaction term.

The interaction term describes explained variance in the presence of both **PropMD** and **TotExp**, *without regard for the variance they explain individually*. In some cases, the sum of two features is greater than the whole; this is what the interaction effect describes.

In this case, we see no statistical significance in the interaction effect between the proportion of medical doctors and total healthcare expenses, so we should not include this in our model.

Discuss policy recommendations.

By itself, however, **PropMD** does have a positive relationship with **LifeExp** and has improved our R^2 , so we should keep it in the model.

Regarding policy, we can advise the WHO that more doctors can have a positive effect on average life expectancy.

Question 5: Forecasting Life Expectancy with Interaction Terms

```
# Forecast life expectancy using the multiple regression model
X_test <- tibble(TotExp = c(14), PropMD = c(0.03)) # Not transformed
y_test <- predict(who_model_3, X_test)
y_test <- y_test ^ (1/4.6)
print(y_test)
```

```
##          1
## 66.97703
```

Discuss whether the forecast seems realistic and limitations of the model.

Since **TotEXP** = 14 is a parameter far beyond our data (and likely an unrealistic expenditure amount to begin with), we must interpret the results with caution. Extrapolated inputs, that is data that is orders of magnitude beyond our training set, is unlikely to be accurately predicted by a simple linear model.

Problem 3: Retail Company Analysis

Question 1: Inventory Cost

Scenario

A retail company is planning its inventory strategy for the upcoming year. They expect to sell 110 units of a high-demand product. The storage cost is \$3.75 per unit per year, and there is a fixed ordering cost of \$8.25 per order.

$$C(Q) = \frac{D}{Q} \times S + \frac{Q}{2} \times H$$

Task

Setting the derivative of this function equal to zero will tell us the “equilibrium” state of the lot size. Then we can divide the annual demand by the lot size to determine how many orders we should place throughout the year.

First, the derivative of $C(Q)$;

$$C'(q) = \frac{1}{2} * 3.75 - \frac{110}{q^2} * 8.25 = 1.875 - \frac{907.5}{q^2}$$

Now, setting the derivative equal to zero to get q , our ideal lot size / order quantity;

$$1.875 - \frac{875}{q^2} = 0 \rightarrow \sqrt{q^2} = \sqrt{\frac{907.5}{1.875}} \rightarrow q = 22$$

Then our number of orders per year is simply the annual demand divided by lot size; $110/22 = 5$.

I've also solved this programmatically and graphically below.

```
# Calculate optimal lot size and number of orders

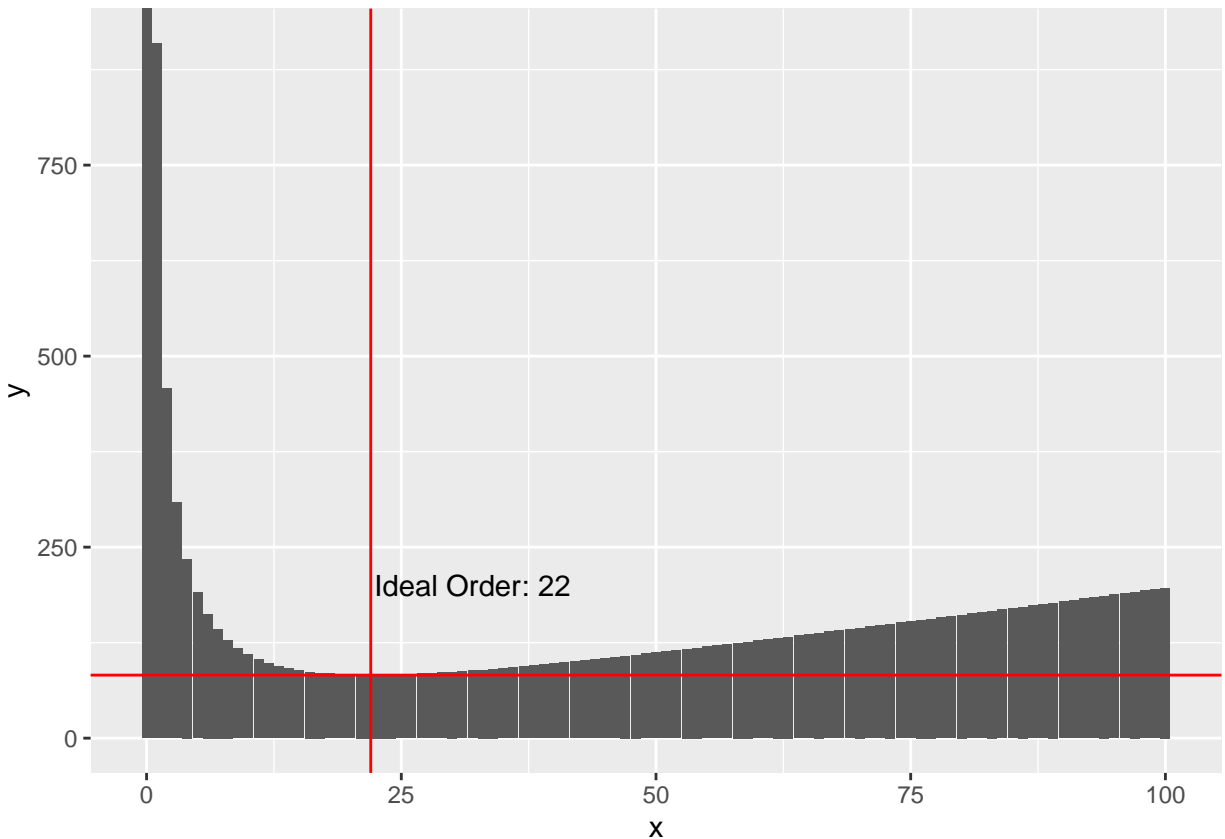
# Our cost function
inventory_cost <- function(d=110, s=8.25, h=3.75, q) {
  y <- (d / q) * s + (q / 2) * h
  return(y)
}

# Set up a number of potential orders
q_vec <- seq(0,100, by=1)

# Set up mock data
inventory_data <- tibble(
  x = q_vec,
  y = inventory_cost(q=q_vec)
)

# Get minimum value
ideal_order <- inventory_data |>
  slice_min(y, n=1) |>
  pull(x)

# Plot to help visualize the function
ggplot(inventory_data, aes(x=x, y=y)) +
  geom_histogram(stat='identity') +
  geom_hline(yintercept = min(inventory_data$y), color='red') +
  geom_vline(xintercept = (ideal_order), color='red') +
  annotate("text", x=ideal_order + 10, y = 200, label=glue("Ideal Order: {ideal_order}"))
```



```
print(ideal_order)
```

```
## [1] 22
```

Question 2: Revenue Maximization

Scenario

A company is running an online advertising campaign with revenue function:

$$R(t) = -3150t^{-4} - 220t + 6530$$

Task

Similarly to the last question, we can solve this by evaluating where the derivative equal to zero.

```
# Symbolic
# Must initialize values for R to work with them symbolically
# Values don't matter
t <- 1

fr <- expression(-3150*(t^-4) - 220*t + 6530)
fr_deriv <- D(fr, "t")
print(fr_deriv)
```

```
## 3150 * (t^-(4 + 1) * 4) - 220
```

$$R'(t) = 3150 * (t^{-(5)} * 4) - 220$$

$$0 = 3150 * 4t^{-(5)} - 220 \rightarrow t = 2.47$$

```
# Find time at which revenue is maximized

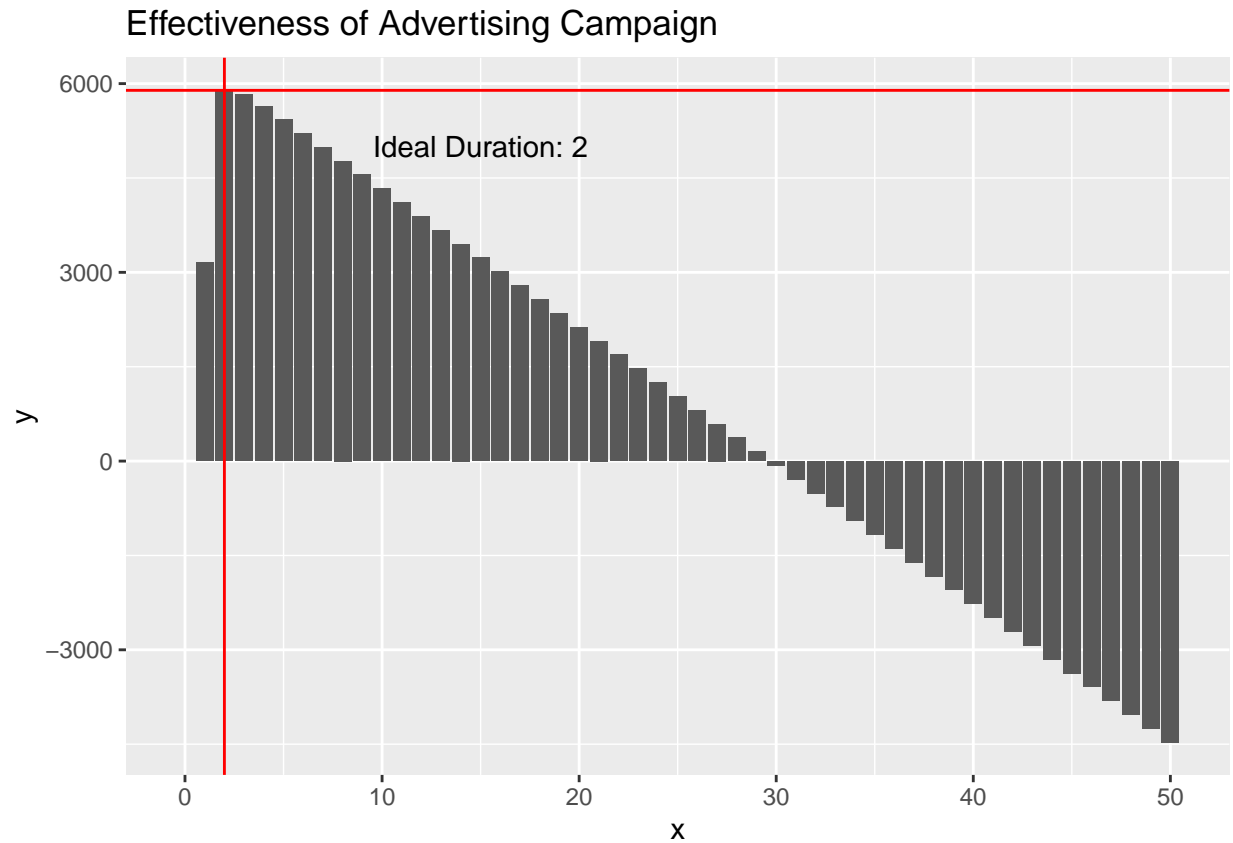
advert_revenue <- function(t) {
  y <- -3150*(t^-4) - 220*t + 6530
  return(y)
}

t_vec <- seq(0, 50, by=1)

# Set up mock data
revenue_data <- tibble(
  x = t_vec,
  y = advert_revenue(t=t_vec)
)

# Get minimum value
ideal_campaign_length <- revenue_data |>
  slice_max(y, n=1) |>
  pull(x)

# Plot to help visualize the function
ggplot(revenue_data, aes(x=x, y=y)) +
  geom_histogram(stat='identity') +
  geom_hline(yintercept = max(revenue_data$y), color='red') +
  geom_vline(xintercept = (ideal_campaign_length), color='red') +
  labs(title = "Effectiveness of Advertising Campaign") +
  annotate("text", x=15, y = 5000, label=glue("Ideal Duration: {ideal_campaign_length}"))
```



Question 3: Demand Area Under Curve

Scenario

A company sells a product with demand function:

$$P(q) = 2x - 9.3$$

Task

$$\int_2^5 (30 - 2q) dx$$

Calculate area under the demand curve

```
x1 <- 2
```

```
x2 <- 5
```

```
demand_func <- function(x) {
```

```
  y <- 2*x - 9.3
```

```
  return(y)
```

```
}
```

```
px1 <- demand_func(x1)
```

```
px2 <- demand_func(x2)
```

```
revenue_data <- tibble(
```

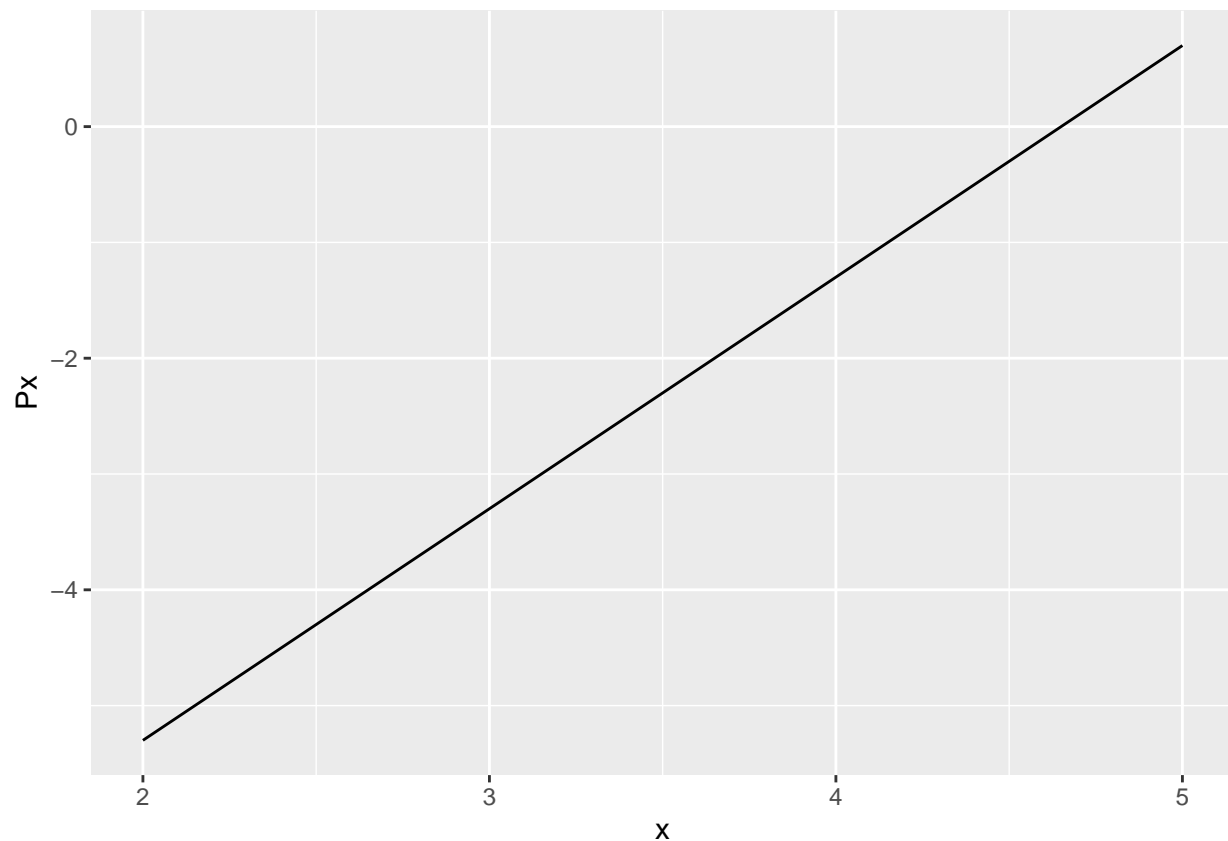


```
x = c(x1, x2),
Px = c(px1, px2)
)
```

```
print(revenue_data)
```

```
## # A tibble: 2 x 2
##       x      Px
##   <dbl> <dbl>
## 1     2 -5.3
## 2     5  0.700
```

```
ggplot(revenue_data, aes(x=x, y=Px)) +
  geom_line()
```



```
integrate(demand_func, lower=x1, upper = x2)
```

```
## -6.9 with absolute error < 7.9e-14
```

Producing between \int_2^5 shows us that for this product, we would actually lose money, approximately -7. We should instead plan to produce greater than 5 units, should the demand allow for it, which would give us a better opportunity to recognize real revenues.

Question 4: Profit Optimization

Scenario

A beauty supply store sells flat irons with profit function:

$$\Pi(x) = x \ln(9x) - \frac{x^6}{6}$$

Task

Find the value of x that maximizes profit.

To find the value that maximizes profit, we must take the derivative of the function and find the value at which it is zero.

We find the derivative to be $\Pi'(x) = \ln(9x) + 1 - x^5$. Setting this to zero, $0 = \ln(9x) + 1 - x^5$, we must solve for x .

Given the algebraic complexity of this expression, we can use Newton's method to solve, by stating that $x_{n+1} = x_n - \Pi'(x_n)/\Pi''(x_n)$. We can iterate through this series to find the value to which x converges.

To do this, we calculate the second derivative; $\Pi''(x) = \frac{1}{x} - 5x^4$

```
newton <- function(x) {  
  
  pi_1 <- function(x) log(9*x) + 1 - x^5  
  pi_2 <- function(x) 1/x - 5*(x^4)  
  
  x_0 <- x  
  
  for (i in 1:10) {  
    pi_1x <- pi_1(x)  
    pi_2x <- pi_2(x)  
  
    x_new <- x - pi_1x / pi_2x  
    print(x_new)  
  
    x <- x_new  
  }  
}
```

```
newton(1)
```

```
## [1] 1.549306  
## [1] 1.361415  
## [1] 1.290189  
## [1] 1.280792  
## [1] 1.280641  
## [1] 1.280641  
## [1] 1.280641  
## [1] 1.280641  
## [1] 1.280641  
## [1] 1.280641  
## [1] 1.280641
```

We can see that x converges to 1.2806 very quickly, which is the optimal value for profit maximization. To find the actual expected profit, we can use this x value in the original function.

```
Pi_x <- function(x) x*log(9*x) - (x^6)/6
print(Pi_x(1.280641))
```

```
## [1] 2.395423
```

In conclusion, the ideal profit is expected to be approximately \$2.40 for $x = 1.2806$

Question 5: Spending Behavior

Scenario

A market research firm is analyzing spending behavior with probability density function:

$$f(x) = \frac{1}{6x}$$

Task

Determine whether this function is a valid probability density function over the interval $[1, e^6]$. If it is, calculate the probability that a customer spends between \$1 and \$ e^6 .

A key property of a probability distribution is that its area is equal to one. With this property in mind, we can validate if this is a proper probability distribution by integrating the function between one and e^6 and asserting it is approximately equal to one.

If this is true, then we can also state that probability a customer spends between one and e^6 is also one, since this is the range of the entire distribution.

```
# Determine if valid PDF and calculate probability
```

```
# Check if integral = 1
```

```
fx <- function (x) {
```

```
  1 / (6 * x)
```

```
}
```

```
# Valid probability distribution
```

```
integrate(fx, 1, exp(6))
```

```
## 1 with absolute error < 9.3e-05
```

We can see that the integral of the function is 1, so it is indeed a valid probability distribution.

Question 6: Market Share Estimation

Scenario

An electronics company has market penetration rate:

$$\frac{dN}{dt} = \frac{500}{t^4 + 10}$$

Task

Integrate this function to find the cumulative market share $N(t)$ after t days, given that the initial market share $N(1) = 6530$. What will the market share be after 10 days?

We can integrate the derivative to find the market share value between 1 and 10 days. Given we know $N(1)$, we can then simply add $N(1) + \int_1^{10} \frac{dn}{dt}$ to find the market share after 10 days.

```

# Integrate to find cumulative market share
dN <- function (t) {
  500 / (t^4 + 10)
}

# Limit between 0, 1 to account for constant of integration
val <- integrate(dN, 1, 10)$value

print(val + 6530)

```

```
## [1] 6579.54
```

As a result, we can see that expected market share after 10 days is approximately 6579

Problem 4: Business Optimization

Scenario

As a data scientist at a consultancy firm, you are tasked with optimizing various business functions using Taylor Series expansions.

```

# Claude generated function to plot Taylor series approximations
# using pracma taylor() and polyval()

plot_with_taylor <- function(functions, labels = NULL, x_range = c(-2, 2),
                             center = 0, degree = 3,
                             colors = c("red", "blue", "green"),
                             xlab = "x", ylab = "y", main = "Functions with Taylor Series Approximations") {

  # Create x values for plotting
  x <- seq(x_range[1], x_range[2], length.out = 100)

  # Create a data frame for ggplot
  plot_data <- data.frame(x = numeric(0),
                          y = numeric(0),
                          type = character(0),
                          function_name = character(0))

  # Track y values for dynamic scaling
  all_y_values <- c()

  for (i in 1:length(functions)) {
    f <- functions[[i]]
    function_name <- labels[i]

    # Get y values for original function with error handling
    y <- sapply(x, function(x_val) {
      tryCatch(f(x_val), error = function(e) NA, warning = function(w) NA)
    })

    # Track y values for scaling
    all_y_values <- c(all_y_values, y)

    # Add original function to plot data

```

```

function_data <- data.frame(
  x = x,
  y = y,
  type = "Original",
  function_name = function_name
)
plot_data <- rbind(plot_data, function_data)

# Add Taylor approximation with error handling
tryCatch({
  taylor_coef <- taylor(f, center, degree)

  # Calculate Taylor approximation for each x
  y_taylor <- sapply(x, function(x_val) {
    tryCatch(
      polyval(taylor_coef, x_val - center),
      error = function(e) NA,
      warning = function(w) NA
    )
  })

  # Track y values for scaling
  all_y_values <- c(all_y_values, y_taylor)

  # Add Taylor approximation to plot data
  taylor_data <- data.frame(
    x = x,
    y = y_taylor,
    type = paste("Taylor degree", degree),
    function_name = function_name
  )
  plot_data <- rbind(plot_data, taylor_data)
}, error = function(e) {
  warning(paste("Could not compute Taylor series for function", i,
    "- Error:", e$message))
}, warning = function(w) {
  # Suppress warnings
})
}

# Create the named vectors for scale functions properly
linetype_values <- c("solid", "dashed")
names(linetype_values) <- c("Original", paste("Taylor degree", degree))

# Create ggplot with combined legend for line type and width
p <- ggplot(plot_data, aes(x = x, y = y, color = function_name)) +
  geom_line(aes(linetype = type)) +
  scale_linetype_manual(values = linetype_values) +
  scale_color_manual(values = colors[1:length(functions)]) +
  labs(
    title = main,

```

```

    x = xlab,
    y = ylab,
    color = "Function",
    linetype = "Type"
  ) +
  # Combine the legends for linetype and linewidth
  guides(
    linewidth = "none", # Hide the linewidth legend
    linetype = guide_legend(title = "Type") # Keep only linetype legend with title "Type"
  ) +
  theme_minimal() +
  geom_vline(xintercept = 0, linetype = "dotted", color = "gray50") +
  geom_hline(yintercept = 0, linetype = "dotted", color = "gray50")

  return(p)
}

```

Question 1: Revenue and Cost

Scenario

A company's revenue is approximated by $R(x) = e^x$ and cost by $C(x) = \ln(1 + x)$. Profit is defined as $\Pi(x) = R(x) - C(x)$

Task

```

Rx <- function (x) {
  x^2 * exp(1)^(-0.1 * x)
}

Cx <- function (x) {
  sqrt((x^3) + 1)
}

Px <- function (x) {
  Rx(x) - Cx(x)
}

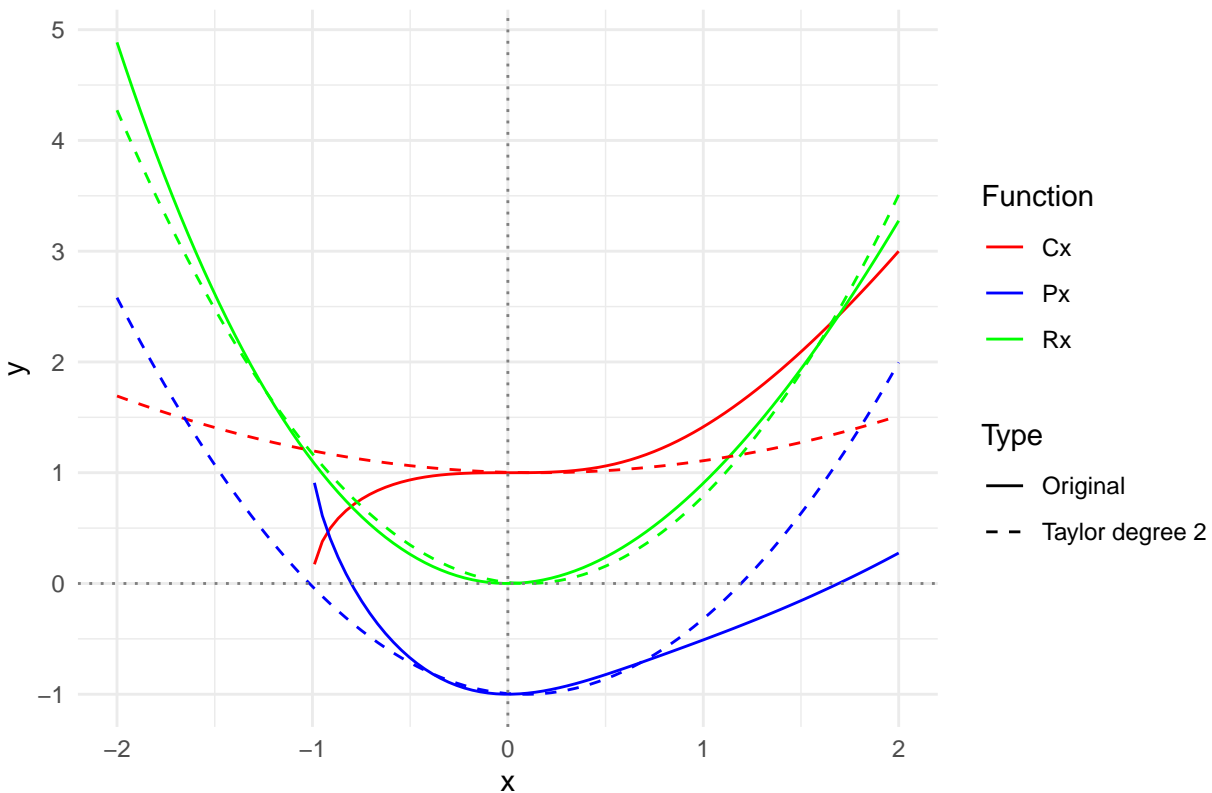
# Approximate Revenue Function using Taylor Series to second degree
# Approximate Cost Function using Taylor Series
# Approximate Profit function using Taylor Series

plot_with_taylor(functions = list(Rx, Cx, Px), labels = c("Rx", "Cx", "Px"), center = 0.1, degree = 2, n

```

Warning: Removed 50 rows containing missing values or values outside the scale range
(`geom_line()`).

Cost, Profit, and Revenue Functions



```
# Compare optimization results linear vs original
```

Taylor Series expansions are useful to approximate functions by leveraging the nature of derivatives of exponential functions.

Taylor series allow us to get simpler approximations around a point of a function. Real world functions may be polynomials of very high degrees, which can be difficult to compute. Using Taylor series, we can take a simpler linear or quadratic (1st and 2nd degree polynomials) approximation of the original function.

From our plot of $P(x)$, our approximation would suggest producing at least 1.5x units, and the actual function suggests greater than 1.75x units in order to recognized a profit.

Question 2: Financial Modeling

Scenario

A financial analyst is modeling risk as $f(x) = \sqrt{x}$.

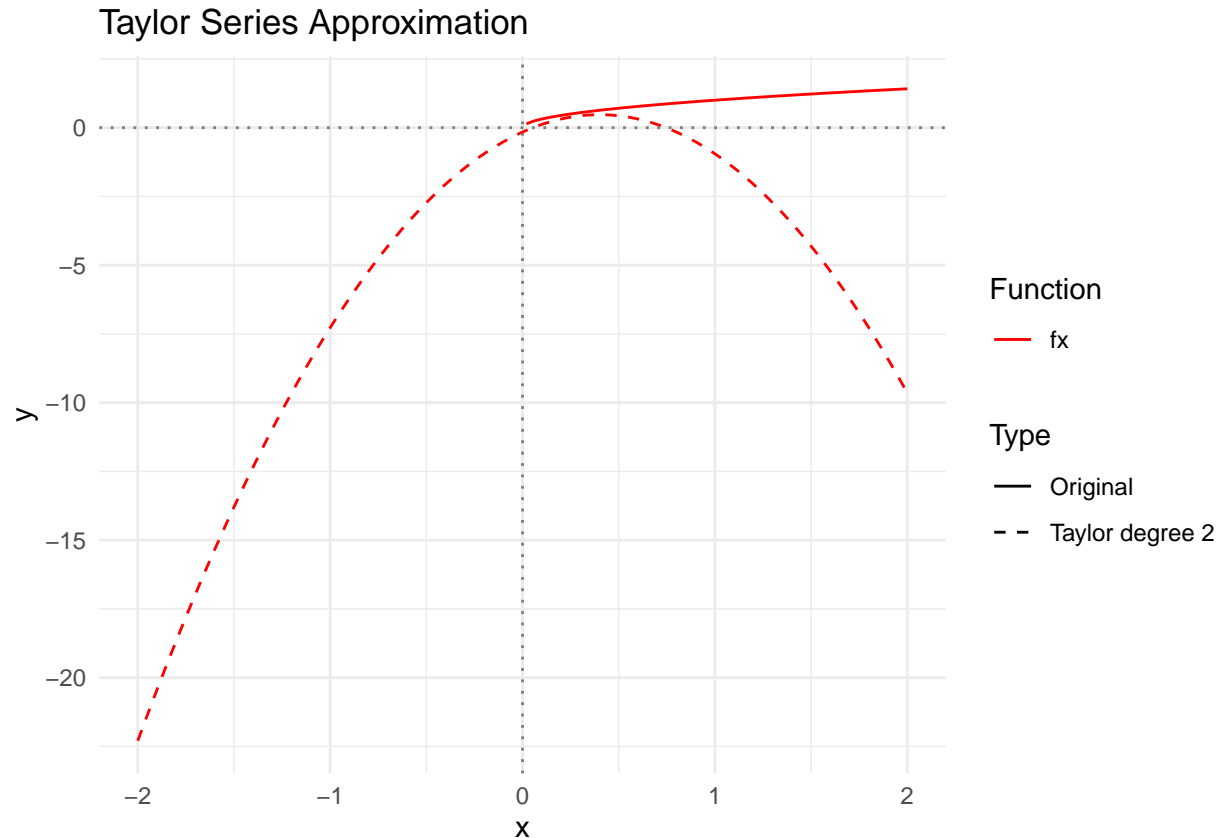
Task

```
# Derive Taylor Series expansion around 0 to 2nd degree
```

```
fx <- function(x){
  x^(1/2)
}
```

```
# Compare approximated risk with actual function
plot_with_taylor(functions = list(fx), labels = c("fx"), center = 0.1, degree = 2, main = "Taylor Series")

## Warning: Removed 50 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



```
# Find optimal investment amount that balances risk and return
```

In this scenario, our approximation is still useful for small investments ($0 < x < 0.5$). Outside of this range, the Taylor series does not converge well enough for us to reliably predict the risk of the investment.

An optimally balanced portfolio would depend on the risk preference of the investor, as $f(x)$ will continue to increase in risk as x increases. The derivative function should then be set to equal the alpha rate, which would provide a value of x that matches the investor's risk appetite.

Question 3: Inventory Management

Scenario

In a manufacturing process, demand decreases as price increases: $D(p) = 1 - p$, with cost $C(p) = e^p$.

Task

```
# Expand cost function using Taylor Series around 0 to 2nd degree

Cp <- function (p) {
  exp(p)
}
```



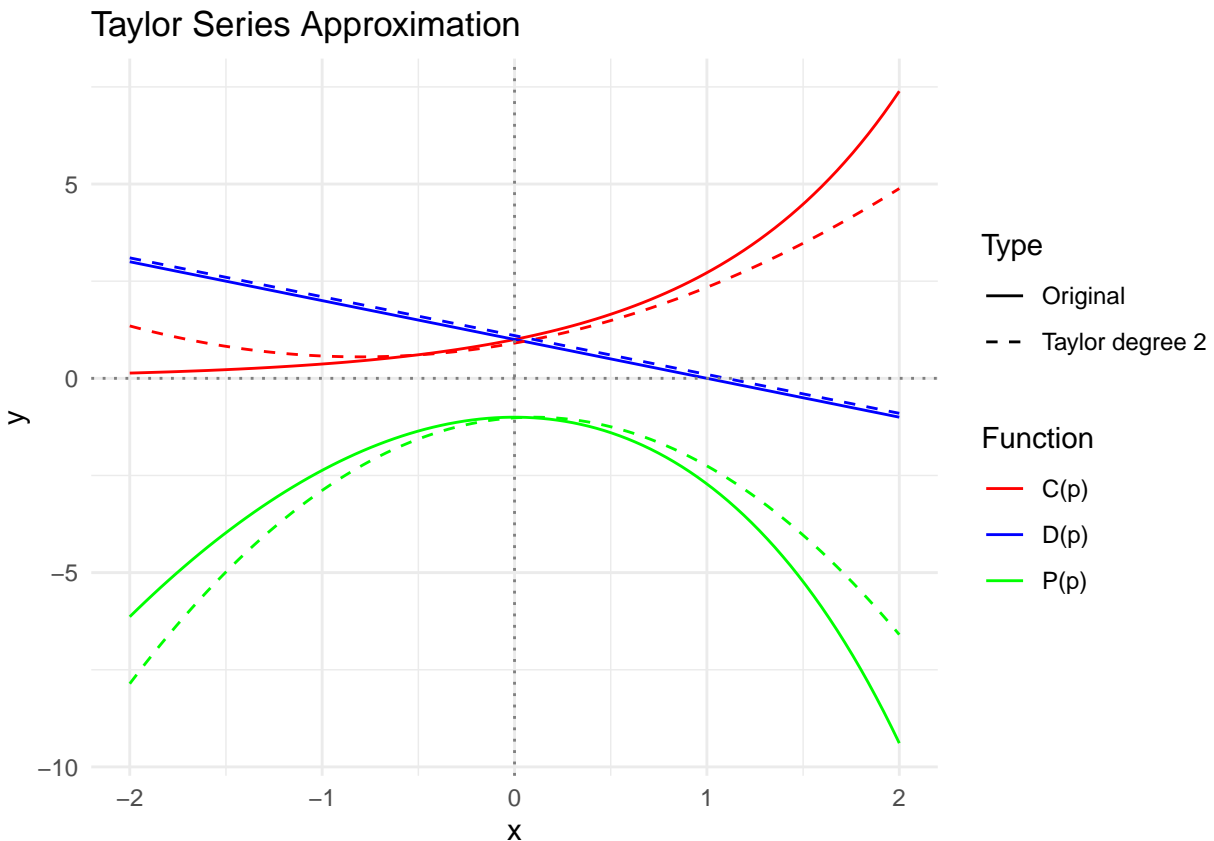
```

Dp <- function(p) {
  1 - p
}

# Approximate profit function
Pp <- function(p) {
  p * Dp(p) - Cp(p)
}

plot_with_taylor(functions = list(Cp, Dp, Pp), labels = c("C(p)", "D(p)", "P(p)"), center = 0.1, degree

```



With the simpler functions of $C(p)$, $D(p)$, $P(p)$, we can see that a 2 degree Taylor series expansion approximates the original functions quite well.

However, we must highlight that given these cost, demand, and profit functions, there is no price p that could be set to achieve a profit. There is simply not enough demand in this market to justify the cost of production.

Question 4: Economic Forecasting

Scenario

Economic growth is modeled by $G(x) = \ln(1 + x)$.

Task

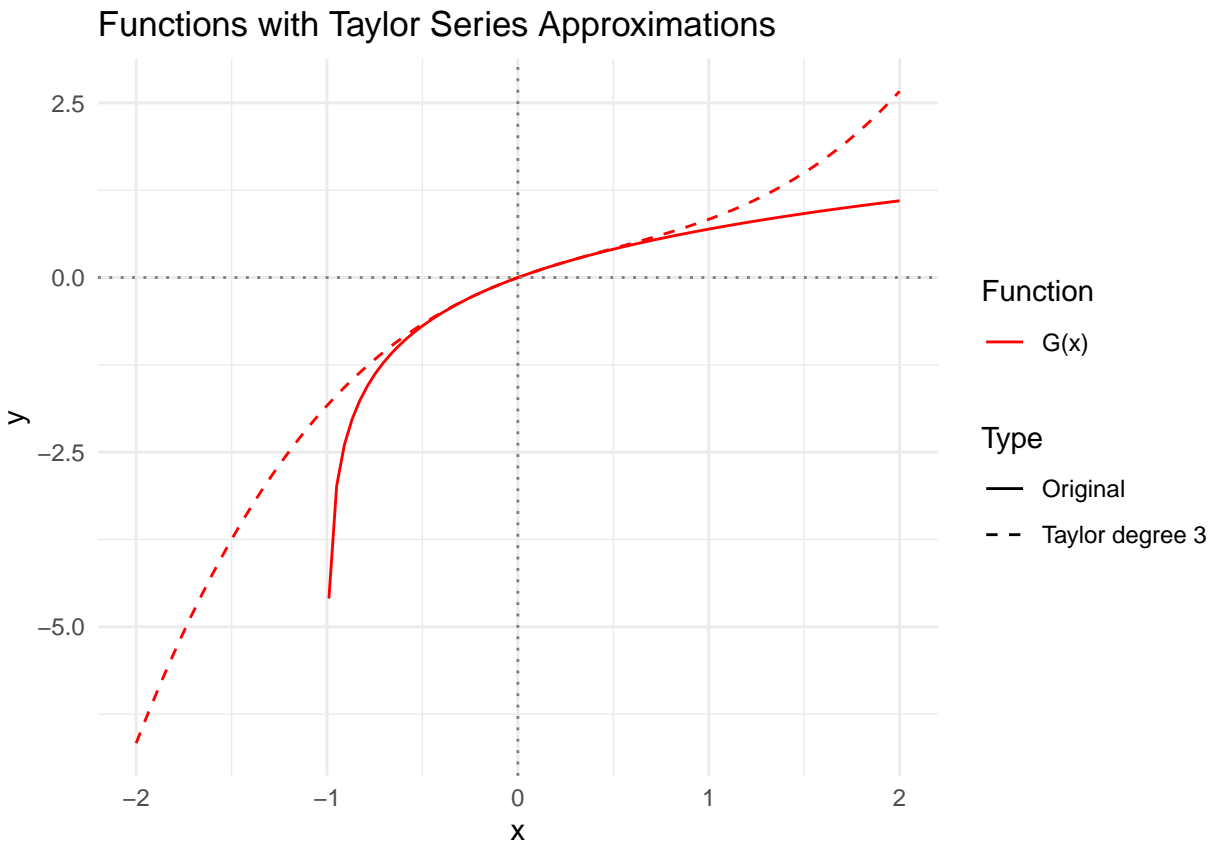
```

# Derive Maclaurin Series expansion

```

```
Gx <- function (x) {
  log(1+x)
}
# Approximate growth for small investments

plot_with_taylor(c(Gx), labels=c("G(x)"), center = 0) # 0 for Maclaurin
```



```
# Recommend investment level
```

In economic forecasting, we are frequently updating our forecast models with the last current events and information based on today's information at $t = 0$, which we could also describe as a steady state or $x = 0$. These models can become extremely complex, so being able to update our Taylor Series approximations, around the steady state $x = 0$, gives us a much better way of reducing the complexity of the forecasting model.

From the plot, we can see that this particular function is very well approximated between $x = [-0.5, 0.75]$. Within this range, we could advise policy the expected growth rate expected by our approximation with confidence. However, it would be important to update our data regularly; as we observed, the approximation will fall off as the steady state changes.

Problem 5: Profit, Cost, & Pricing

Question 1: Profit Maximization

Scenario

A company produces two products with profit function:

$$\Pi(x, y) = 30x - 2x^2 - 3xy + 24y - 4y^2$$

Task

```
# Find local maxima, minima, and saddle points

Px <- function(x, y) {
  30*x - 2*(x^2) - 3*x*y + 24*y - 4*(y^2)
}

# Partial derivatives
partial_Px <- function(vars) {
  x <- vars[1]
  y <- vars[2]

  # Partial derivatives
  dfdx <- 30 - 4*x - 3*y
  dfdy <- -3*x + 24 - 8*y

  return(c(dfdx, dfdy))
}

# Find critical point by setting to zero
cp <- fsolve(partial_Px, c(0,0))

# Second partial derivative
fxx <- -4
fyy <- -8
fxy <- -3

# Find Discriminant

D <- fxx * fyy - (fxy^2)

# Classify the critical point
if (D > 0 && fxx < 0) {
  cat("Critical point is a local maximum \n")
} else if (D > 0 && fxx > 0) {
  cat("Critical point is a local minimum\n")
} else if (D < 0) {
  cat("Critical point is a saddle point\n")
} else {
  cat("Further analysis required\n")
}

## Critical point is a local maximum
```

```
print(cp[[1]][1])
```

```
## [1] 7.304348
```

```
print(cp[[1]][2])
```

```
## [1] 0.2608696
```

```
Px(cp[[1]][1], cp[[1]][2])
```

```
## [1] 112.6957
```

The maximum profit is expected to be 112.70 at price points 7.30 and 0.26 for x, y respectively.

Question 2: Pricing Strategy

Scenario

A supermarket sells two competing brands with demand functions:

$$D_X(x, y) = 120 - 15x + 10y$$

$$D_Y(x, y) = 80 + 5x - 20y$$

Task

The revenue function, $R_{XY}(x, y)$ is the sum of demand functions $x * D_X(x, y) + y * D_Y(x, y)$.

Expanding and combining like terms, we get $R_{XY}(x, y) = 120x - 15x^2 + 15xy + 80y - 20y^2$.

We can take the partial derivatives with regards to x and y and set them to zero to find the critical point.

We find $f_x = 120 - 15x + 15y$, $f_y = 15x + 80 - 40y$. We can now solve the system of equations via substitution.

Finding that $x = 0.5y + 40$, we can solve for y such that $0 = 15(0.5y + 40) + 80 - 40y$ to find that $y = 4.31$. Similarly, we find $x = 0.5(4.31) + 40$ to find $x = 6.155$. Our critical point then is $P_C = (6.155, 4.31)$.

We take the second derivatives to classify our critical point. We find $f_{xx} = -15$, $f_{yy} = -40$, $f_{xy} = 15$.

We use these to calculate the discriminant $D = f_{xx} * f_{yy} - f_{xy}^2$, which evaluates to $D = 375$.

Since the $D > 0$ and $F_{xx} < 0$, there exists a local maximum for the revenue function. We simply evaluate $R(x, y)$ at our critical point to find the maximum revenue; $R(6.155, 4.31) = 541.3$

Question 3: Cost Minimization

Scenario

A manufacturing company operates two plants with total cost:

$$C(x, y) = \frac{1}{8}x^2 + \frac{1}{10}y^2 + 12x + 18y + 1500$$

Task

Since we need to produce 200 units, we must first establish $x + y = 200$. We can then take the usual steps to find the minimum of this cost function.

We can then express the cost function purely in terms of x as $y = 200 - x$ to get $C(x) = \frac{1}{8}x^2 + \frac{1}{10}(200 - x)^2 + 12x + 18(200 - x) + 1500$, which reduces to $C(x) = 0.225x^2 - 46x + 9100$

Reducing, we find the partial derivative with respect to x to be $f_x = 0.225x - 46$, and setting it equal to zero, $x = 102.22$

We then find $y = 200 - x = 200 - 102.22 = 97.78$, which gives us our critical point (minimum) to be $P_C(102.22, 97.78)$. This means we should produce approximately 102 units in New York, and 98 units in Chicago.

Since $C(x)$ is a quadratic, it has one extreme and no saddle points.

Question 4: Marketing Mix

Scenario

A company's marketing effectiveness is modeled by:

$$E(x, y) = 500x + 700y - 5x^2 - 10xy - 8y^2$$

Task

The partial derivatives with respect to x and y are $f_x = 500 - 10x - 10y$, $f_y = 700 - 16y - 10x$. We solve the system of equations to find $x = 16.7$ and $y = 33.3$, giving us our critical point $P_C = (16.7, 33.3)$.

We classify the critical point by calculating the discriminant $D = f_{xx} * f_{yy} - f_{xy}^2$. We find $f_{xx} = -10$, $f_{yy} = -16$, and $f_{xy} = -10$. We then calculate $D = -10 * -16 - (-10)^2 = 60$.

As $D = 60 > 0$ and $f_{xx} = -10 < 0$, we can confirm the critical point is the maximum of the function. Thus we can say that to maximize customer reach in our marketing campaign, we should invest approximately 17 units for product x and 33 units for product y .