

DATA_605_Assignment_3_Fox

Amanda Fox

2025-05-18

Preparation

Load libraries:

```
# load libraries
library(tidyverse)
library(ggplot2)
library(scales)
library(Deriv)
library(rootSolve)
library(tidyr)
```

Problem 1: Transportation Safety

1. Data Visualization:

Create a scatter plot of stopping distance (dist) as a function of speed (speed). Add a regression line to the plot to visually assess the relationship.

Stopping distance increases with speed in a roughly linear pattern, with a few possible outliers and more variability at higher speeds.

```
# Load data
data("cars")
glimpse(cars)
```

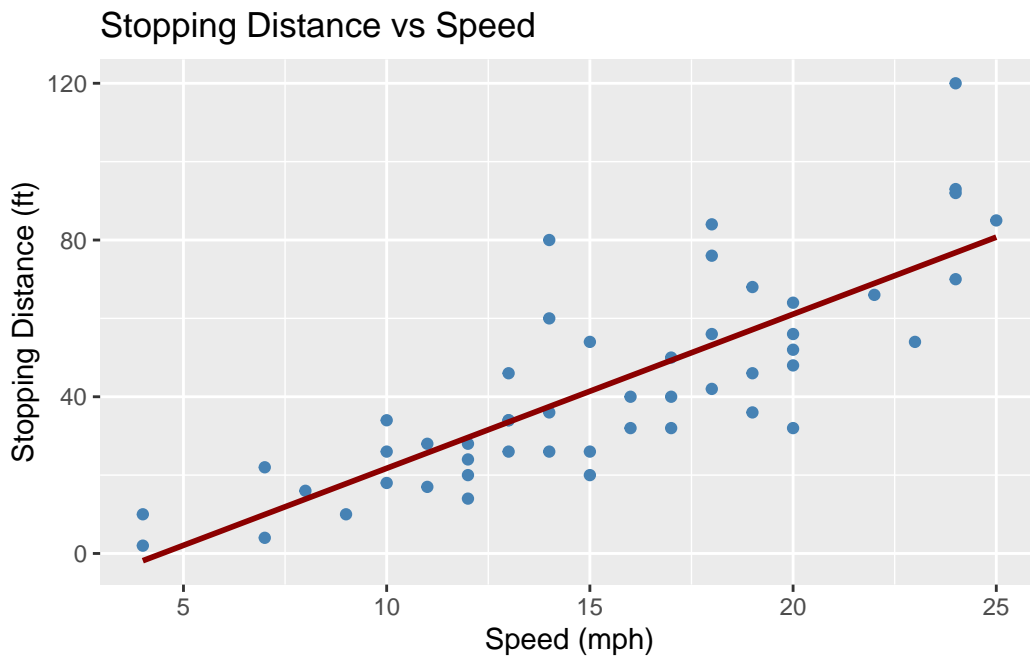
Rows: 50

Columns: 2

```
$ speed <dbl> 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 12, 13, 13, 13~
$ dist  <dbl> 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28, 14, 20, 24, 28, 26, 34~
```

```
plt1 <- cars %>%
  ggplot(aes(x = speed, y = dist)) +
    geom_point(color = "steelblue") +
    geom_smooth(method = "lm",
                se = FALSE,
                color = "darkred") +
    labs(title = "Stopping Distance vs Speed",
         x = "Speed (mph)", y = "Stopping Distance (ft)")

plt1
```



2. Build a Linear Model:

Construct a simple linear regression model where stopping distance (*dist*) is the dependent variable and speed (*speed*) is the independent variable. Summarize the model to evaluate its coefficients, R-squared value, and p-value

Stopping Distance = $3.932409 \times \text{speed} - 17.58$

R-squared = 0.6511 which indicates that speed explains 65.11% of variation in stopping distance.

p-value = 1.49×10^{-12} which indicates that this relationship is very unlikely to be due to chance.

```
# Model
mod_dist <- lm(dist ~ speed, data = cars)

# Summarize
summary(mod_dist)
```

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

```
# Add predicted values (verified new plot matches above)
df_cars_model <- cars %>%
  mutate(predicted_dist = predict(mod_dist))

glimpse(df_cars_model)
```

Rows: 50

Columns: 3

```
$ speed      <dbl> 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 12, 1~
$ dist       <dbl> 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28, 14, 20, 24, 2~
$ predicted_dist <dbl> -1.849460, -1.849460, 9.947766, 9.947766, 13.880175, 17~
```

3 & 4: Model Quality Evaluation & Residual Analysis

Calculate and interpret the R-squared value to assess the proportion of variance in stopping distance explained by speed.

R-squared is 0.6511 so 65.11% of the variation in stopping distance is explained by speed.

Perform a residual analysis to check the assumptions of the linear regression model, including linearity, homoscedasticity, independence, and normality of residuals. Plot the residuals versus fitted values to check for any patterns. Create a Q-Q plot of the residuals to assess normality. Perform a Shapiro-Wilk test for normality of residuals. Plot a histogram of residuals to further check for normality.

Assumptions are generally well met. The relationship is roughly linear and residual tests below show a reasonable fit with roughly normal distribution of residuals and some heteroscedasticity.

In the context of the small dataset size and visual examination of the scatterplots above, these results are acceptable. Note that independence is assumed since the behavior of one car should not impact another.

1. Residuals vs Fitted: Fan shape suggests some heteroscedasticity with increasing variance at higher speeds
2. Histogram of Residuals: Broadly normally distributed with a slight right skew
3. Q-Q: Follows the line generally but deviation at both ends, particularly the upper end, suggests some non-normality/outliers
4. Shapiro-Wilkes test: $p < 0.05$ suggests non-normality (null hypothesis = data is normal, which is rejected).

```
# Add to dataframe
df_cars_model <- df_cars_model %>%
  mutate(
    .fitted = fitted(mod_dist),
    .resid = resid(mod_dist),
    .std_resid = rstandard(mod_dist)
  )

glimpse(df_cars_model)
```

Rows: 50

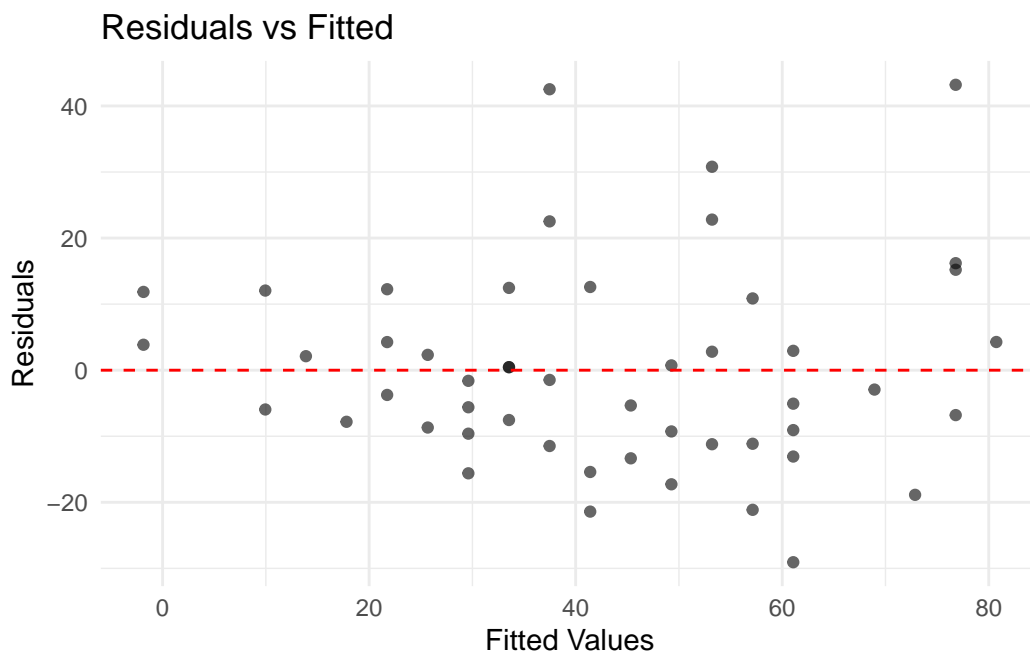
Columns: 6

```
$ speed      <dbl> 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 12, 1~
$ dist       <dbl> 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28, 14, 20, 24, 2~
$ predicted_dist <dbl> -1.849460, -1.849460, 9.947766, 9.947766, 13.880175, 17~
$ .fitted    <dbl> -1.849460, -1.849460, 9.947766, 9.947766, 13.880175, 17~
$ .resid     <dbl> 3.849460, 11.849460, -5.947766, 12.052234, 2.119825, -7~
$ .std_resid <dbl> 0.26604155, 0.81893273, -0.40134618, 0.81326629, 0.1421~
```

```
#-----
# Diagnostic Plots**
#-----

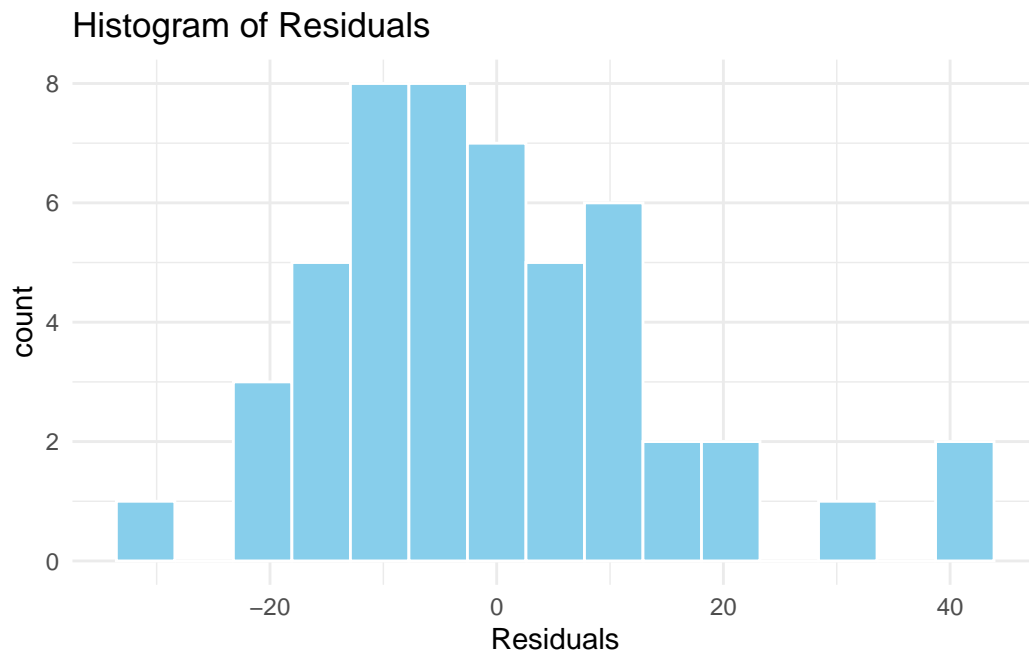
# Residuals vs Fitted
plot_resid <- df_cars_model %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Residuals vs Fitted",
    x = "Fitted Values",
    y = "Residuals"
  ) +
  theme_minimal()

plot_resid
```

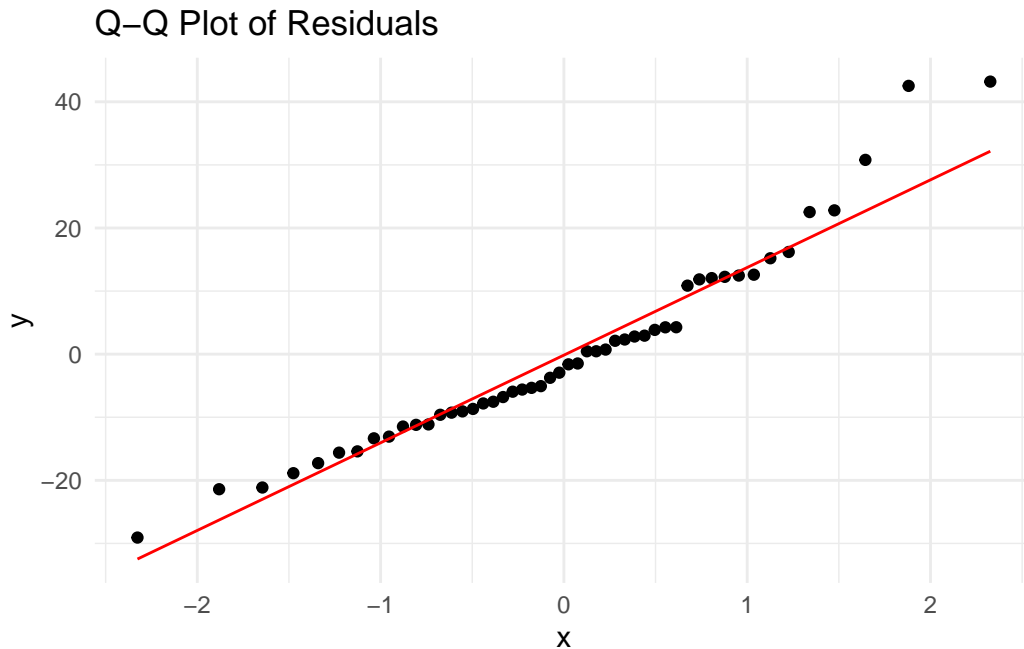


```
# Histogram of Residuals
plot_resid_hist <- df_cars_model %>%
  ggplot(aes(x = .resid)) +
  geom_histogram(bins = 15, fill = "skyblue", color = "white") +
```

```
labs(title = "Histogram of Residuals", x = "Residuals") +  
theme_minimal()  
  
plot_resid_hist
```



```
# Q-Q  
plot_qq <- df_cars_model %>%  
  ggplot(aes(sample = .resid)) +  
  stat_qq() +  
  stat_qq_line(color = "red") +  
  labs(title = "Q-Q Plot of Residuals") +  
  theme_minimal()  
  
plot_qq
```



```
# shapiro-wilkes test
df_cars_model$.resid %>%
  shapiro.test()
```

Shapiro-Wilk normality test

```
data: .
W = 0.94509, p-value = 0.02152
```

5. Conclusion

Based on the model summary and residual analysis, determine whether the linear model is appropriate for this data. Discuss any potential violations of model assumptions and suggest improvements if necessary.

Based on the model summary and residual analysis, the linear model is acceptable. The R-squared value is meaningful but not strong; model fit might improve by adding more predictors such as weight, tire specs, etc.

Linearity, normality, and constant variance are generally met, with acceptable slight heteroscedasticity and non-normality in residuals. Independence was not formally tested but reasonable to assume since each car's stopping distance is unrelated to others.

1. Initial Assessment of Healthcare Expenditures and Life Expectancy

```
#-----  
# Load data  
#-----  
  
df_who_raw <- read_csv("https://raw.githubusercontent.com/AmandaSFox/DATA605_Math/main/Assignment1/WHO_2019_2020.csv")  
  
glimpse(df_who_raw)
```

```

Rows: 190
Columns: 12
$ Country      <chr> "Afghanistan", "Albania", "Algeria", "Andorra", "Angola~
$ LifeExp...2  <dbl> 42, 71, 71, 82, 41, 73, 75, 69, 82, 80, 64, 74, 75, 63,~
$ InfantSurvival <dbl> 0.835, 0.985, 0.967, 0.997, 0.846, 0.990, 0.986, 0.979,~
$ Under5Survival <dbl> 0.743, 0.983, 0.962, 0.996, 0.740, 0.989, 0.983, 0.976,~
$ TBFree      <dbl> 0.99769, 0.99974, 0.99944, 0.99983, 0.99656, 0.99991, 0~
$ PropMD      <dbl> 0.000228841, 0.001143127, 0.001060478, 0.003297297, 0.0~
$ PropRN      <dbl> 0.000572294, 0.004614439, 0.002091362, 0.003500000, 0.0~
$ PersExp     <dbl> 20, 169, 108, 2589, 36, 503, 484, 88, 3181, 3788, 62, 1~
$ GovtExp     <dbl> 92, 3128, 5184, 169725, 1620, 12543, 19170, 1856, 18761~
$ ...10       <lg1> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ TotExp      <dbl> 112, 3297, 5292, 172314, 1656, 13046, 19654, 1944, 1907~
$ LifeExp...12 <dbl> 42, 71, 71, 82, 41, 73, 75, 69, 82, 80, 64, 74, 75, 63,~

```

```
#-----  
# Clean data  
#-----  
  
# check for NAs  
df_who_raw %>%  
  summarise(across(everything(), ~sum(is.na(.))))
```

8


```

1      0      0      0      0      0      0      0      0
# i 4 more variables: GovtExp <int>, ...10 <int>, TotExp <int>,
#   LifeExp...12 <int>

```

```

# check for duplicate rows
nrow(df_who_raw) - nrow(distinct(df_who_raw))

```

```
[1] 0
```

```

# count unique values in each column
df_who_raw %>%
  summarise(across(everything(), n_distinct))

```

```

# A tibble: 1 x 12
  Country LifeExp...2 InfantSurvival Under5Survival TBFree PropMD PropRN PersExp
    <int>      <int>      <int>      <int> <int> <int> <int> <int>
1    190      43      83      99    140    188    189    161
# i 4 more variables: GovtExp <int>, ...10 <int>, TotExp <int>,
#   LifeExp...12 <int>

```

```

# verify col 10 includes only NA values
unique(df_who_raw$...10)

```

```
[1] NA
```

```

# verify both LifeExp columns are identical
all(df_who_raw$`LifeExp...2` == df_who_raw$`LifeExp...12`)

```

```
[1] TRUE
```

```

# drop second LifeExp and NA columns, rename LifeExp
df_who_clean <- df_who_raw %>%
  select(- LifeExp...12, - ...10) %>%
  rename(LifeExp = `LifeExp...2`)

glimpse(df_who_clean)

```

Rows: 190

Columns: 10

```
$ Country      <chr> "Afghanistan", "Albania", "Algeria", "Andorra", "Angola~
$ LifeExp      <dbl> 42, 71, 71, 82, 41, 73, 75, 69, 82, 80, 64, 74, 75, 63,~
$ InfantSurvival <dbl> 0.835, 0.985, 0.967, 0.997, 0.846, 0.990, 0.986, 0.979,~
$ Under5Survival <dbl> 0.743, 0.983, 0.962, 0.996, 0.740, 0.989, 0.983, 0.976,~
$ TBFree       <dbl> 0.99769, 0.99974, 0.99944, 0.99983, 0.99656, 0.99991, 0~
$ PropMD       <dbl> 0.000228841, 0.001143127, 0.001060478, 0.003297297, 0.0~
$ PropRN       <dbl> 0.000572294, 0.004614439, 0.002091362, 0.003500000, 0.0~
$ PersExp      <dbl> 20, 169, 108, 2589, 36, 503, 484, 88, 3181, 3788, 62, 1~
$ GovtExp      <dbl> 92, 3128, 5184, 169725, 1620, 12543, 19170, 1856, 18761~
$ TotExp       <dbl> 112, 3297, 5292, 172314, 1656, 13046, 19654, 1944, 1907~
```

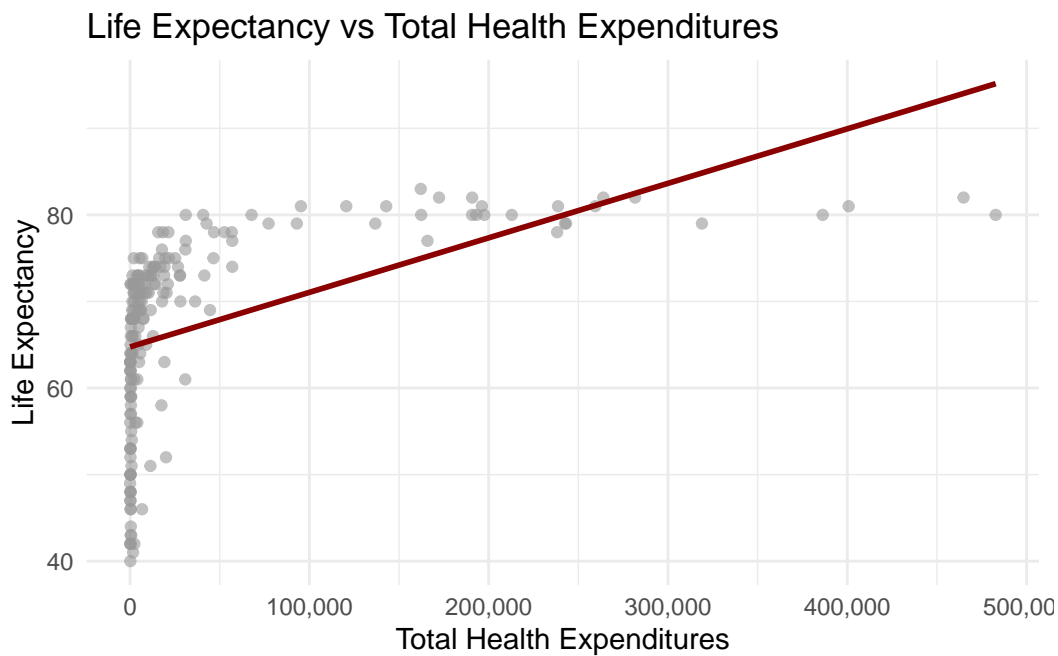
```
#-----
# Summary and Plot
#-----

# summarize
summary(df_who_clean)
```

Country	LifeExp	InfantSurvival	Under5Survival
Length:190	Min. :40.00	Min. :0.8350	Min. :0.7310
Class :character	1st Qu.:61.25	1st Qu.:0.9433	1st Qu.:0.9253
Mode :character	Median :70.00	Median :0.9785	Median :0.9745
	Mean :67.38	Mean :0.9624	Mean :0.9459
	3rd Qu.:75.00	3rd Qu.:0.9910	3rd Qu.:0.9900
	Max. :83.00	Max. :0.9980	Max. :0.9970
TBFree	PropMD	PropRN	PersExp
Min. :0.9870	Min. :0.0000196	Min. :0.0000883	Min. : 3.00
1st Qu.:0.9969	1st Qu.:0.0002444	1st Qu.:0.0008455	1st Qu.: 36.25
Median :0.9992	Median :0.0010474	Median :0.0027584	Median : 199.50
Mean :0.9980	Mean :0.0017954	Mean :0.0041336	Mean : 742.00
3rd Qu.:0.9998	3rd Qu.:0.0024584	3rd Qu.:0.0057164	3rd Qu.: 515.25
Max. :1.0000	Max. :0.0351290	Max. :0.0708387	Max. :6350.00
GovtExp	TotExp		
Min. : 10.0	Min. : 13		
1st Qu.: 559.5	1st Qu.: 584		
Median : 5385.0	Median : 5541		
Mean : 40953.5	Mean : 41696		
3rd Qu.: 25680.2	3rd Qu.: 26331		
Max. :476420.0	Max. :482750		

```
# Scatterplot with linear regression line
plt3 <- df_who_clean %>%
  ggplot(aes(x = TotExp, y = LifeExp)) +
  geom_point(alpha = 0.6, color = "gray60") +
  geom_smooth(method = "lm", se = FALSE, color = "darkred") +
  scale_x_continuous(labels = scales::comma)+
  labs(
    title = "Life Expectancy vs Total Health Expenditures",
    x = "Total Health Expenditures",
    y = "Life Expectancy"
  ) +
  theme_minimal()

plt3
```



*Run a simple linear regression with *LifeExp* as the dependent variable and *TotExp* as the independent variable (without transforming the variables).*

$$\text{LifeExp} = 64.75 + 0.00006297 \times \text{TotExp}$$

For every additional \$1 in total healthcare spending, life expectancy increases by about 0.000063 years or about 0.023 days above a “baseline” of 64.75 years at \$0 spending.

*Provide and interpret the *F*-statistic, *R*-squared value, standard error, and *p*-values.*

- F-statistic and p value: 65.26 on 1 and 188 DF with p of 7.7e-14 indicates the relationship between total expenditure and life expectancy is extremely unlikely to be due to chance
- R-squared: 0.2577 indicates that total healthcare expenditure explains only 25.77% of the variance in life expectancy, which is weak
- Standard error: 9.37 indicates that average error is 9.37 years which is significant in this case as the interquartile range is 13.75 years.

Discuss whether the assumptions of simple linear regression (linearity, independence, homoscedasticity, and normality of residuals) are met in this analysis.

Overall the model violates all assumptions except independence (countries are independent observations so independence is assumed).

- Linearity: The relationship is clearly non-linear in the scatterplot
- Homoscedasticity: The residual plot shows an uneven distribution of residuals around the line
- Normality: The histogram of residuals is skewed left and the qq plot varies widely around the line. The Shapiro-Wilkes small p value also indicates non normality.

Discussion: Consider the implications of your findings for health policy. Are higher healthcare expenditures generally associated with longer life expectancy? What do the assumptions of the regression model suggest about the reliability of this relationship?

Higher healthcare expenditures are positively correlated to life expectancy but the relationship is not linear. While the relationship is significant and not likely due to chance (low p value), the fit of this model is poor with a low R-squared and high standard error.

To draw conclusions on which to base policy, additional work is needed such as transformations or a non-linear model.

```
#-----
# Model
#-----
mod_life <- lm(LifeExp ~ TotExp, data = df_who_clean)

summary(mod_life)
```

Call:

```
lm(formula = LifeExp ~ TotExp, data = df_who_clean)
```

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

```
#-----  
# Diagnostics  
#-----  
  
# add predicted and residuals to df  
df_who_model <- df_who_clean %>%  
  mutate(predicted_life = predict(mod_life),  
         .fitted = fitted(mod_life),  
         .resid = resid(mod_life),  
         .std_resid = rstandard(mod_life))  
  
glimpse(df_who_model)
```

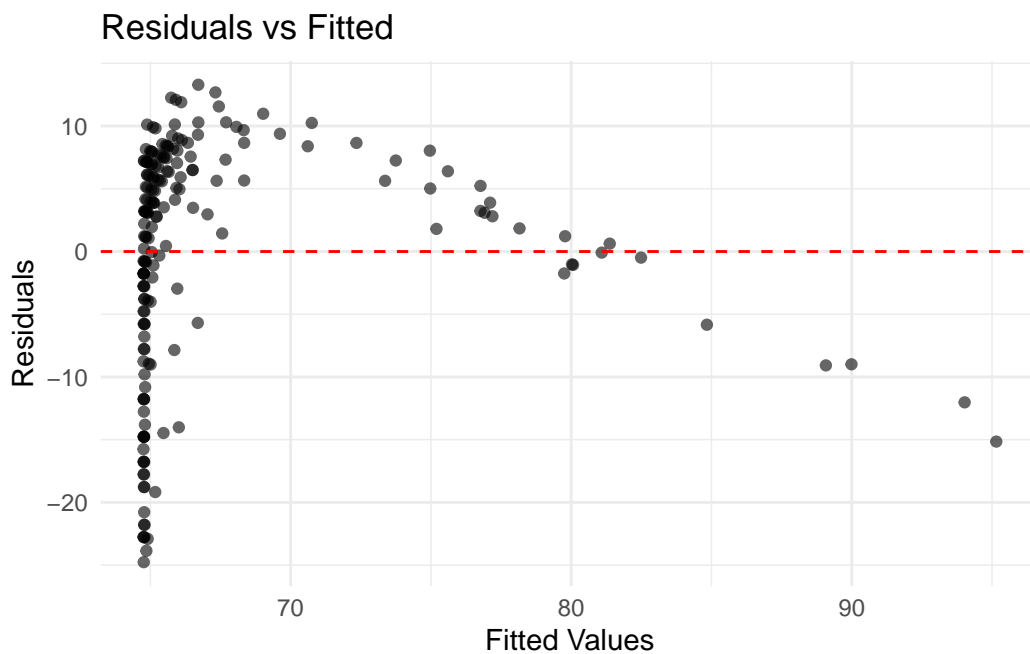
Rows: 190

Columns: 14

\$ Country	<chr>	"Afghanistan", "Albania", "Algeria", "Andorra", "Angola~
\$ LifeExp	<dbl>	42, 71, 71, 82, 41, 73, 75, 69, 82, 80, 64, 74, 75, 63,~
\$ InfantSurvival	<dbl>	0.835, 0.985, 0.967, 0.997, 0.846, 0.990, 0.986, 0.979,~
\$ Under5Survival	<dbl>	0.743, 0.983, 0.962, 0.996, 0.740, 0.989, 0.983, 0.976,~
\$ TBFree	<dbl>	0.99769, 0.99974, 0.99944, 0.99983, 0.99656, 0.99991, 0~
\$ PropMD	<dbl>	0.000228841, 0.001143127, 0.001060478, 0.003297297, 0.0~
\$ PropRN	<dbl>	0.000572294, 0.004614439, 0.002091362, 0.003500000, 0.0~
\$ PersExp	<dbl>	20, 169, 108, 2589, 36, 503, 484, 88, 3181, 3788, 62, 1~
\$ GovtExp	<dbl>	92, 3128, 5184, 169725, 1620, 12543, 19170, 1856, 18761~
\$ TotExp	<dbl>	112, 3297, 5292, 172314, 1656, 13046, 19654, 1944, 1907~
\$ predicted_life	<dbl>	64.76043, 64.96099, 65.08661, 75.60402, 64.85765, 65.57~
\$.fitted	<dbl>	64.76043, 64.96099, 65.08661, 75.60402, 64.85765, 65.57~
\$.resid	<dbl>	-22.7604272, 6.0390128, 5.9133872, 6.3959804, -23.85765~
\$.std_resid	<dbl>	-2.43668924, 0.64646820, 0.63298727, 0.68842673, -2.554~

```
# Residuals vs Fitted
plot_resid_who <- df_who_model %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Residuals vs Fitted", x = "Fitted Values", y = "Residuals") +
  theme_minimal()

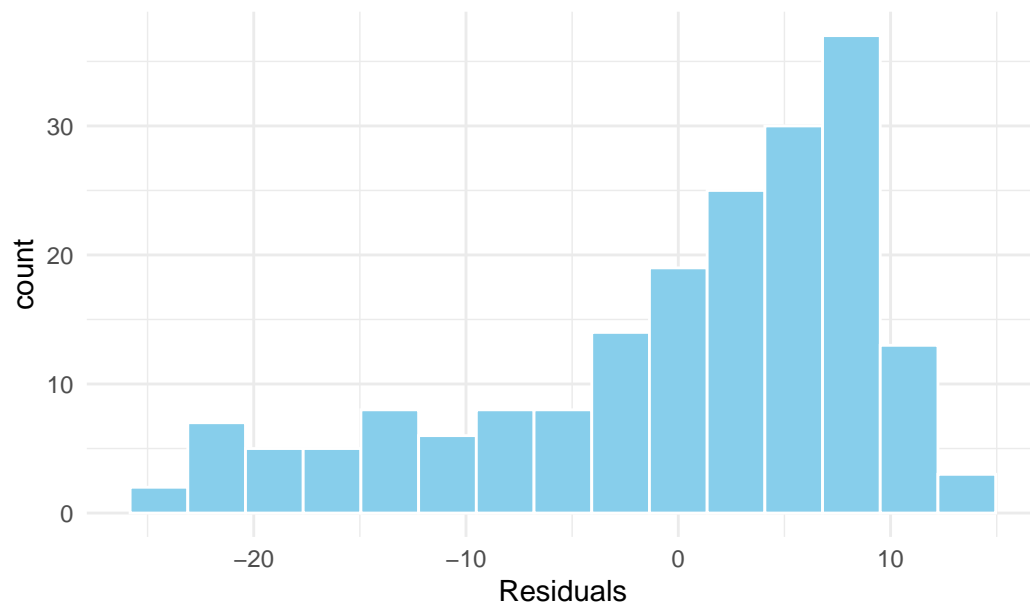
plot_resid_who
```



```
# Histogram of Residuals
plot_hist_who <- df_who_model %>%
  ggplot(aes(x = .resid)) +
  geom_histogram(bins = 15, fill = "skyblue", color = "white") +
  labs(title = "Histogram of Residuals", x = "Residuals") +
  theme_minimal()

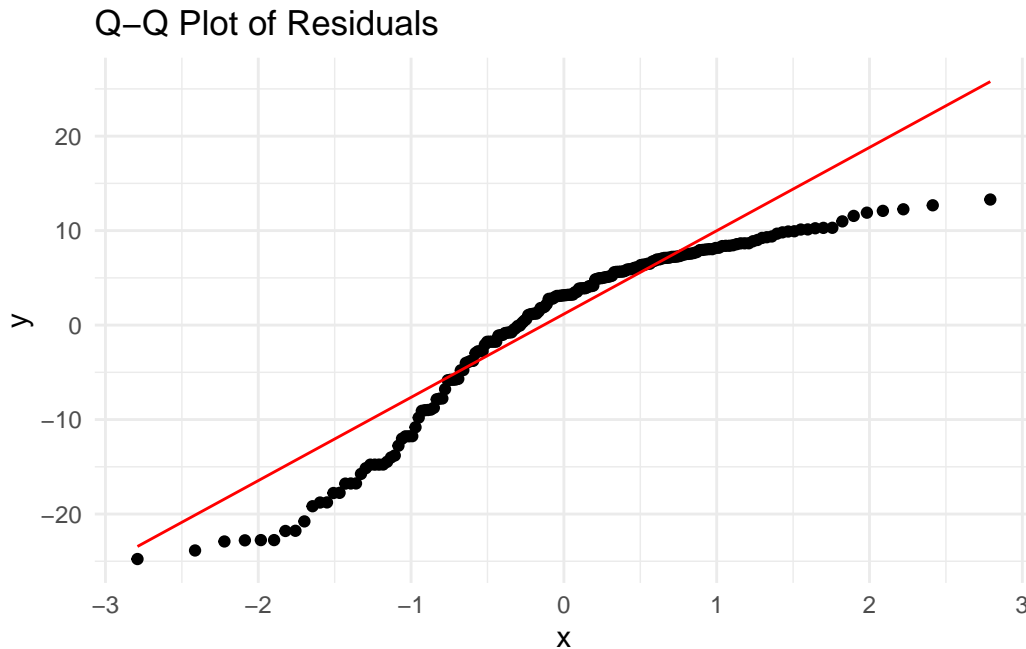
plot_hist_who
```

Histogram of Residuals



```
# QQ plot
plot_qq_who <- df_who_model %>%
  ggplot(aes(sample = .resid)) +
  stat_qq() +
  stat_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals") +
  theme_minimal()

plot_qq_who
```



```
# Shapiro Wilkes
shapiro.test(df_who_model$.resid)
```

Shapiro-Wilk normality test

```
data: df_who_model$.resid
W = 0.89146, p-value = 1.609e-10
```

2. Transforming Variables for a Better Fit

Task: Recognizing potential non-linear relationships, transform the variables as follows: Raise life expectancy to the 4.6 power ($LifeExp^{4.6}$). Raise total expenditures to the 0.06 power ($TotExp^{0.06}$), which is nearly a logarithmic transformation. Create a new scatterplot with the transformed variables and re-run the simple linear regression model.

The transformed data now has a visually linear relationship on the scatterplot with a more even distribution around the linear regression line.

Provide and interpret the F-statistic, R-squared value, standard error, and p-values for the transformed model.

- F-statistic and p value: 507.7 on 1 and 188 DF with p of 2.2e-16 indicates the relationship between total expenditure and life expectancy is extremely unlikely to be due to chance
- R-squared: 0.7298 indicates that total healthcare expenditure now explains 72.98% of the variance in life expectancy, which is good
- Standard error: 90,490,000 is not readily interpretable but it is still pretty significant compared to the transformed life expectancy values (e.g. it's about 21.4% of $75^{4.6}$).

The model has improved in fit with a higher R-square, but the standard error is still significant compared to the predicted values, so there is still room for improvement.

Compare this model to the original model (from Question 1). Which model provides a better fit, and why?

- F-statistic and p value: Already very good, they improved even more: the relationship between total expenditure and life expectancy is extremely unlikely to be due to chance
- R-squared: Significant improvement from 0.2577 (weak fit) to 0.7298 (strong)
- Standard error: Proportionately, the standard error increased compared to typical life spans from approximately 12.5% of a typical 75 year life span to 21.4%.

The transformed model provides a significantly better fit but can still be improved.

Discussion: How do the transformations impact the interpretation of the relationship between healthcare spending and life expectancy? Why might the transformed model be more appropriate for policy recommendations?

The transformations reveal a clear linear relationship and greatly improve the fit of the model, reducing unexplained variance. Prediction error however remains large, and to be interpretable, the data should be backtransformed. However, it is more appropriate for policy recommendations than the first model.

```
#-----
# Transform and Plot
#-----

df_who_transform <- df_who_clean %>%
  mutate(LifeExp_46 = LifeExp^4.6,
         TotExp_006 = TotExp^0.06)

glimpse(df_who_transform)
```

Rows: 190

Columns: 12

```
$ Country      <chr> "Afghanistan", "Albania", "Algeria", "Andorra", "Angola~
$ LifeExp      <dbl> 42, 71, 71, 82, 41, 73, 75, 69, 82, 80, 64, 74, 75, 63,~
```

```

$ InfantSurvival <dbl> 0.835, 0.985, 0.967, 0.997, 0.846, 0.990, 0.986, 0.979,~
$ Under5Survival <dbl> 0.743, 0.983, 0.962, 0.996, 0.740, 0.989, 0.983, 0.976,~
$ TBFree          <dbl> 0.99769, 0.99974, 0.99944, 0.99983, 0.99656, 0.99991, 0~
$ PropMD          <dbl> 0.000228841, 0.001143127, 0.001060478, 0.003297297, 0.0~
$ PropRN          <dbl> 0.000572294, 0.004614439, 0.002091362, 0.003500000, 0.0~
$ PersExp         <dbl> 20, 169, 108, 2589, 36, 503, 484, 88, 3181, 3788, 62, 1~
$ GovtExp         <dbl> 92, 3128, 5184, 169725, 1620, 12543, 19170, 1856, 18761~
$ TotExp          <dbl> 112, 3297, 5292, 172314, 1656, 13046, 19654, 1944, 1907~
$ LifeExp_46      <dbl> 29305338, 327935478, 327935478, 636126841, 26230450, 37~
$ TotExp_006      <dbl> 1.327251, 1.625875, 1.672697, 2.061481, 1.560068, 1.765~

```

```

# Scatterplot with linear regression line
plt4 <- df_who_transform %>%
  ggplot(aes(x = TotExp_006, y = LifeExp_46)) +
  geom_point(alpha = 0.6, color = "gray60") +
  geom_smooth(method = "lm", se = FALSE, color = "darkred") +
  scale_x_continuous(labels = scales::comma)+
  labs(
    title = "Transformed: Life Expectancy vs Total Health Expenditures",
    x = "Total Health Expenditures ^ 0.06",
    y = "Life Expectancy ^ 4.6"
  ) +
  theme_minimal()

plt4

```

Transformed: Life Expectancy vs Total Health Expenditures



```
#-----
# Model
#-----

mod_life_transform <- lm(LifeExp_46 ~ TotExp_006, data = df_who_transform)

summary(mod_life_transform)
```

Call:

```
lm(formula = LifeExp_46 ~ TotExp_006, data = df_who_transform)
```

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_006	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

3. Forecasting Life Expectancy Based on Transformed Expenditures

Task: Using the results from the transformed model in Question 2, forecast the life expectancy for countries with the following transformed total expenditures ($TotExp^{0.06}$):

- When $TotExp^{0.06} = 1.5$ life expectancy is predicted to be **63.3** years
- When $TotExp^{0.06} = 2.5$ life expectancy is predicted to be **86.5** years

Discussion: Discuss the implications of these forecasts for countries with different levels of healthcare spending. What do these predictions suggest about the potential impact of increasing healthcare expenditures on life expectancy?

Increasing health expenditures is expected to result in meaningfully increased life expectancy.

```
#-----  
# Predict  
#-----  
  
# create table with new total expense data  
df_who_new_data <- tibble(TotExp_006 = c(1.5, 2.5))  
  
# add predicted life expectancy and backtransform it  
df_who_new_data <- df_who_new_data %>%  
  mutate(predicted_life_exp_46 = predict(mod_life_transform,  
                                          newdata = df_who_new_data),  
          predicted_life_exp_backtransformed = predicted_life_exp_46^(1/4.6))  
  
glimpse(df_who_new_data)
```

```
Rows: 2  
Columns: 3  
$ TotExp_006          <dbl> 1.5, 2.5  
$ predicted_life_exp_46 <dbl> 193562414, 813622630  
$ predicted_life_exp_backtransformed <dbl> 63.31153, 86.50645
```

4: Interaction Effects in Multiple Regression

Task: Build a multiple regression model to investigate the combined effect of the proportion of MDs and total healthcare expenditures on life expectancy. Specifically, use the model:

$$\text{LifeExp} = b_0 + b_1 \times \text{PropMD} + b_2 \times \text{TotExp} + b_3 \times (\text{PropMD} \times \text{TotExp})$$

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

- F-statistic and p value: 34.49 on 3 and 186 DF with p of 2.2e-16 indicates the relationship between total expenditure and life expectancy is extremely unlikely to be due to chance
- R-squared: 0.3574 indicates that total healthcare expenditure now explains 35.74% of the variance in life expectancy, better than the original untransformed linear model (0.26) but worse than the transformed model (0.73). Note that I also ran this model using the transformed data which resulted in an even worse R-squared of about 0.19, so the untransformed version was used.
- Standard error: 8.765 is slightly better than the other two models but still significant.

Evaluate the interaction term (PropMD x TotExp). What does this interaction tell us about the relationship between the number of MDs, healthcare spending, and life expectancy?

The interaction coefficient is highly significant and negative, while the PropMD and TotExp separately are also highly significant and positive. So increasing either has a positive impact, but oddly combining them has a smaller effect.

This may mean that in a country that has a lot of docs or spending already, increasing the other variable has less effect than it otherwise would. For example, if a country already has lots of doctors, extra spending may go toward nicer clinics or equipment and have less impact on life expectancy.

```
# new linear model with PropMD, TotExp, and the interaction
mod_life_interaction <- lm(LifeExp ~ PropMD + TotExp + PropMD:TotExp,
                           data = df_who_clean)

summary(mod_life_interaction)
```

Call:

```
lm(formula = LifeExp ~ PropMD + TotExp + PropMD:TotExp, data = df_who_clean)
```

Residuals:

Min	1Q	Median	3Q	Max
-27.320	-4.132	2.098	6.540	13.074

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.277e+01	7.956e-01	78.899	< 2e-16	***
PropMD	1.497e+03	2.788e+02	5.371	2.32e-07	***
TotExp	7.233e-05	8.982e-06	8.053	9.39e-14	***
PropMD:TotExp	-6.026e-03	1.472e-03	-4.093	6.35e-05	***

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

Residual standard error: 8.765 on 186 degrees of freedom

Multiple R-squared: 0.3574, Adjusted R-squared: 0.3471

F-statistic: 34.49 on 3 and 186 DF, p-value: < 2.2e-16

5: Forecasting Life Expectancy with Interaction Terms

Task: Using the multiple regression model from Question 4, forecast the life expectancy for a country where:

The proportion of MDs is 0.03 (PropMD = 0.03). The total healthcare expenditure is 14 (TotExp = 14).

The predicted life expectancy in the above scenario is **107.7**

Discussion: Does this forecast seem realistic? Why or why not? Consider both the potential strengths and limitations of using this model for forecasting in real-world policy settings.

The forecast is not realistic: the average life expectancy of a population in the real world is extremely unlikely to be nearly 108 years. Nearly all individuals in a real-world population expire before 108.

However, the scenario itself is very extreme and unrealistic: the country would have nearly the lowest expenditure in the world but also nearly the highest proportion of physicians. Such a poor country could not train and support so many physicians with that level of expenditure.

In a real world setting, models can extrapolate beyond what is realistic or even possible in the real world, and so care should be taken especially at extremes like this.

```
#-----  
# Predict  
#-----  
  
# create table with new total expense data  
df_who_new_data2 <- tibble(TotExp = 14,  
                           PropMD = 0.03)
```

```
# add predicted life expectancy and backtransform it
df_who_new_data2 <- df_who_new_data2 %>%
  mutate(predicted_life_exp = predict(mod_life_interaction,
                                     newdata = df_who_new_data2))

glimpse(df_who_new_data2)
```

```
Rows: 1
Columns: 3
$ TotExp          <dbl> 14
$ PropMD          <dbl> 0.03
$ predicted_life_exp <dbl> 107.696
```

```
# original dataset for context
# summarize
df_who_clean %>%
  select(c(PropMD, TotExp)) %>%
  summary()
```

	PropMD	TotExp
Min.	:0.0000196	Min. : 13
1st Qu.	:0.0002444	1st Qu.: 584
Median	:0.0010474	Median : 5541
Mean	:0.0017954	Mean : 41696
3rd Qu.	:0.0024584	3rd Qu.: 26331
Max.	:0.0351290	Max. : 482750

Problem 3-Retail Company Analyst

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. The company wants to minimize its total inventory cost.

Task: Using calculus, determine the optimal lot size (the number of units to order each time) and the number of orders the company should place per year to minimize total inventory costs. Assume that the total cost function is given by:

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

D is the total demand (110 units).

Q is the order quantity.

S is the fixed ordering cost per order (\$8.25).

H is the holding cost per unit per year (\$3.75).

To minimize cost, take the derivative of $C(Q)$ (cost with respect to quantity), set it equal to zero, and solve for Q .

Optimal lot size: 22

Optimal orders/year: 5

```
D <- 110 # demand
S <- 8.25 # cost/order
H <- 3.75 # holding cost/year

# cost function
C <- function(Q) {(D/Q) * S + (Q/2) * H}

# derivative of C(Q) with respect to Q using Deriv package
C_prime <- Deriv(C, "Q")

# solve C'(Q) = 0 for optimal Q using rootSolve package
optimal_Q <- uniroot.all(C_prime, c(0.01, 1000)) # Avoid zero division
optimal_Q
```

```
[1] 22.00002
```

```
# orders per year
optimal_order <- D/optimal_Q
optimal_order
```

```
[1] 4.999996
```

```
# minimized cost
C(optimal_Q)
```

```
[1] 82.5
```


2. Revenue Maximization

Scenario: A company is running an online advertising campaign. The effectiveness of the campaign, in terms of revenue generated per day, is modeled by the function:

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

R(t) represents the revenue in dollars after t days of the campaign.

Task: Determine the time t at which the revenue is maximized by finding the critical points of the revenue function and determining which point provides the maximum value. What is the maximum revenue the company can expect from this campaign?

To find the maximum revenue, I took the derivative of R(t) and set it equal to zero to find the critical points, then found the second derivative and checked its sign at that critical point. It was negative, so that point was a maximum.

Maximum revenue = \$5,912.09

```
R <- function(t) {-3150 * t^(-4) - 220 * t + 6530}

# Derivative
R_prime <- Deriv(R, "t")

# set R' = 0 and solve using rootSolve package
t_critical <- uniroot.all(R_prime, c(0.5, 10)) # Avoid t=0 to prevent division by zero

# Second derivative
R_double_prime <- Deriv(R_prime, "t")

# Evaluate second derivative at the critical point
R_double_prime(t_critical) # negative, so this is a maximum
```

```
[1] -489.3732
```

```
# Max revenue
max_revenue <- R(t_critical)
max_revenue
```

```
[1] 5912.094
```

3. Demand Area Under Curve

Scenario: A company sells a product at a price that decreases over time according to the linear demand function:

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

Where: $P(x)$ is the price in dollars, and x is the quantity sold.

Task: The company is interested in calculating the total revenue generated by this product between two quantity levels, $x_1 = 2$ and $x_2 = 5$, where the price still generates sales. Compute the area under the demand curve between these two points, representing the total revenue generated over this range.

The area under the demand curve is **-6.90** and represents the total value or willingness to pay for the range of quantities. The function implies the company must pay customers to take the product for most of this range ($x \leq 4.65$). For example for $P(5) = 0.7$ but $P(2) = -5.3$.

It is possible that the signs in the function are transposed. In a demand curve, as price goes up, demand should go down, or as x goes up, $P(x)$ should go down.

Transposing the signs gives us that negative slope:

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

The area under the curve (total value or willingness to pay) is now **6.90** which aligns with positive consumer value for the range of quantities.

Now $P(2) = 5.30$, $P(4.65) = 0$, and $P(5) = -.70$. Theoretically when price reaches 0, demand should be infinite, but this is a limitation of a linear demand function.

To find total revenue between $x = 2$ and $x = 4$ (avoiding a zero or negative price), the derivative of $P(x) \cdot x$ can be taken.

```
# Price function
P <- function(x) {2 * x - 9.3}

# AUC using base R
area <- integrate(P, lower = 2, upper = 5)$value

# Print result
area
```

```
[1] -6.9
```

```
# Revised price function
P2 <- function(x) {-2 * x + 9.3}

# AUC using base R
area2 <- integrate(P2, lower = 2, upper = 5)$value

# Print result
area2
```

```
[1] 6.9
```

4. Profit Optimization

Scenario: A beauty supply store sells flat irons, and the profit function associated with selling x flat irons is given by:

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

Task: Use calculus to find the value of x that maximizes profit. Calculate the maximum profit that can be achieved and determine if this optimal sales level is feasible given market conditions.

The flat irons are not a feasible item to sell in current conditions. The maximum profit is achieved at $x = 1.28$ (one or two flat irons) and the **maximum profit is only \$2.40**.

The term $\frac{-x^6}{6}$ increases rapidly with increases in quantity and quickly offsets any additional revenue.

```
# Profit function
Pi <- function(x) {x * log(9 * x) - (x^6) / 6}

# First derivative
Pi_prime <- Deriv(Pi, "x")

# Set Pi' = 0 and solve using rootSolve package
optimal_x <- uniroot.all(Pi_prime, c(0.1, 5)) # Avoid x = 0 due to log
optimal_x
```

```
[1] 1.280677
```

```
# Second derivative
Pi_double_prime <- Deriv(Pi_prime, "x")

# Evaluate second derivative at the critical point
Pi_double_prime(optimal_x) # Should be negative maximum
```

```
[1] -12.66936
```

```
# Max profit
max_profit <- Pi(optimal_x)
max_profit
```

```
[1] 2.395423
```

5. Spending Behavior

Scenario: A market research firm is analyzing the spending behavior of customers in a retail store. The spending behavior is modeled by the probability density function:

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

Where x represents spending in dollars.

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 dollars.

To determine if the function is a valid probability density function (PDF) over the interval, it must be non-negative and integrate to 1.

It is non-negative over the range because the whole set $[1, e^6]$ is positive and $\frac{1}{6x}$ must be positive when x is positive.

It also integrates to 1 over this range, so **it is a valid PDF**.

The probability that a customer spends in that range, the entire PDF, is also **1**.

```
# PDF
pdf <- function(x) { 1 / (6 * x) }

# Define integration limits
pdf_a <- 1
pdf_b <- exp(6)
```

```
# Integrate to check validity
pdf_area <- integrate(pdf, lower = pdf_a, upper = pdf_b)
pdf_area$value # Should be 1
```

```
[1] 1
```

6. Market Share Estimation

Scenario: An electronics company is analyzing its market share over a certain period. The rate of market penetration is given by:

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

Where $N(t)$ is the cumulative market share at time t .

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?

After ten days, total cumulative market penetration will be 6597.54 or approximately **6580 customers**. The rate of change slows dramatically as t increases due to the t^4 in the denominator.

```
# define derivative dN/dt
dN_dt <- function(t) {500/(t^4 +10)}

# integrate from t= 1 to t = 10 (take value!)
mkt_integrated_area <- integrate(dN_dt, lower = 1, upper = 10)$value

# add initial market share at t = 1
N_1 <- 6530
N_10 <- N_1 + mkt_integrated_area
N_10
```

```
[1] 6579.54
```

Problem 4: Business Optimization

1. Revenue and Cost

Scenario: A company's revenue from a product can be approximated by the function $R(x) = e^x$ where x is the number of units sold. The cost of production is given by $C(x) = \ln(1+x)$. The company wants to maximize its profit, defined as $\Pi(x) = R(x) - C(x)$

1. Approximate the Revenue Function: Use the Taylor Series expansion around $x=0$ (Maclaurin series) to approximate the revenue function $R(x) = e^x$ up to the second degree. Explain why this approximation might be useful in a business context.

The Maclaurin series for e^x at the second degree is $e^x \approx 1 + x + \frac{x^2}{2}$.

This approximation is accurate for small values of x less than about 0.5 (see plot below). It would be easy to use for estimation especially when logs are impractical due to lack of skill/comfort level, lack of access to a computer/scientific calculator (which was more likely in the past), or to simplify complex optimization problems.

```
# define exact revenue function
bus_R_exact <- function(x) {exp(x)}

# define maclaurin series approximation function
bus_R_approx <- function(x) {1 + x + (x^2)/2}

# compare
bus_R_x_vals <- seq(0, 1, by = 0.1)

df_bus_R_compare <- data.frame(
  x = bus_R_x_vals,
  R_exact = bus_R_exact(bus_R_x_vals),
  R_approx = bus_R_approx(bus_R_x_vals),
  R_error = bus_R_approx(bus_R_x_vals) - bus_R_exact(bus_R_x_vals))

head(df_bus_R_compare, 11)
```

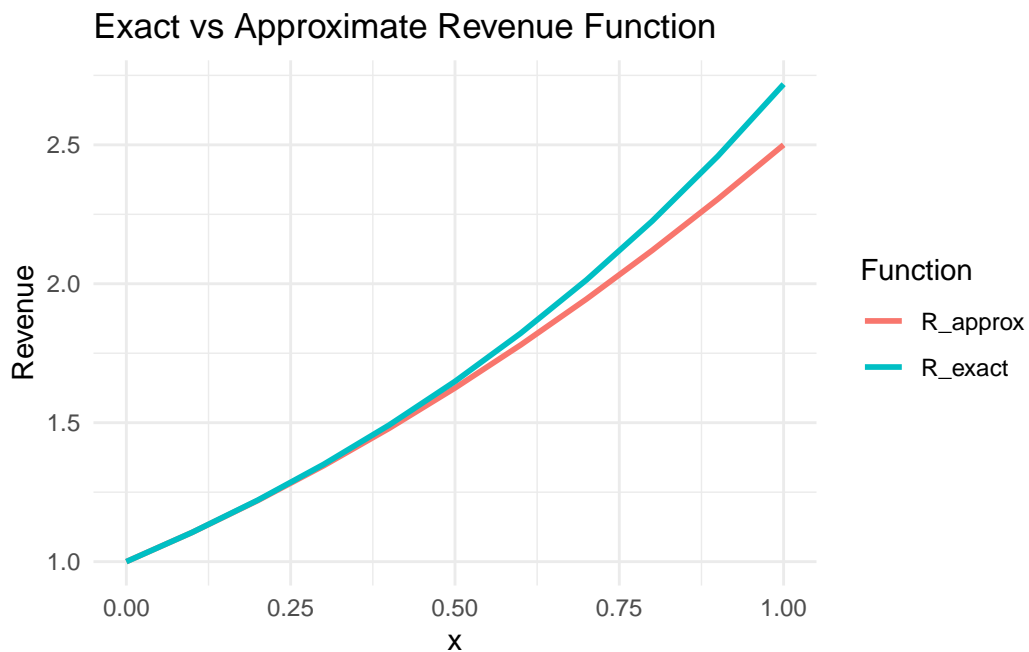
	x	R_exact	R_approx	R_error
1	0.0	1.000000	1.000	0.0000000000
2	0.1	1.105171	1.105	-0.0001709181
3	0.2	1.221403	1.220	-0.0014027582
4	0.3	1.349859	1.345	-0.0048588076
5	0.4	1.491825	1.480	-0.0118246976
6	0.5	1.648721	1.625	-0.0237212707

7	0.6	1.822119	1.780	-0.0421188004
8	0.7	2.013753	1.945	-0.0687527075
9	0.8	2.225541	2.120	-0.1055409285
10	0.9	2.459603	2.305	-0.1546031112
11	1.0	2.718282	2.500	-0.2182818285

```
#plot
df_bus_R_compare_long <- pivot_longer(df_bus_R_compare,
                                       cols = c("R_exact", "R_approx"),
                                       names_to = "type",
                                       values_to = "revenue")

plt_bus_R <- df_bus_R_compare_long %>%
  ggplot(aes(x = x, y = revenue, color = type)) +
  geom_line(size = 1) +
  labs(title = "Exact vs Approximate Revenue Function",
       x = "x",
       y = "Revenue",
       color = "Function") +
  theme_minimal()

plt_bus_R
```



2. *Approximate the Cost Function: Similarly, approximate the cost function $C(x) = \ln(1 + x)$ using its Maclaurin series expansion up to the second degree. Discuss the implications of this approximation for decision-making in production.*

The Maclaurin series for $\ln(1 + x)$ at the second degree is $\ln(1 + x) \approx x - \frac{x^2}{2}$.

This approximation is accurate for small values of $x < 0.25$ (see plot below) but using it in production would greatly underestimate costs for values of $x > 0.25$ and should be avoided to avoid budget surprises.

Note that the cost estimation function underestimates to a greater degree than the revenue estimation, so profit would also be underestimated for larger values of x .

```
# define exact cost function
bus_C_exact <- function(x) {log(1+x)}

# define maclaurin series approximation function
bus_C_approx <- function(x) {x - (x^2)/2}

# compare from 0 to 1
bus_C_x_vals <- seq(0, 1, by = 0.1)

df_bus_C_compare <- data.frame(
  x = bus_C_x_vals,
  C_exact = bus_C_exact(bus_C_x_vals),
  C_approx = bus_C_approx(bus_C_x_vals),
  C_error = bus_C_approx(bus_C_x_vals) - bus_C_exact(bus_C_x_vals))

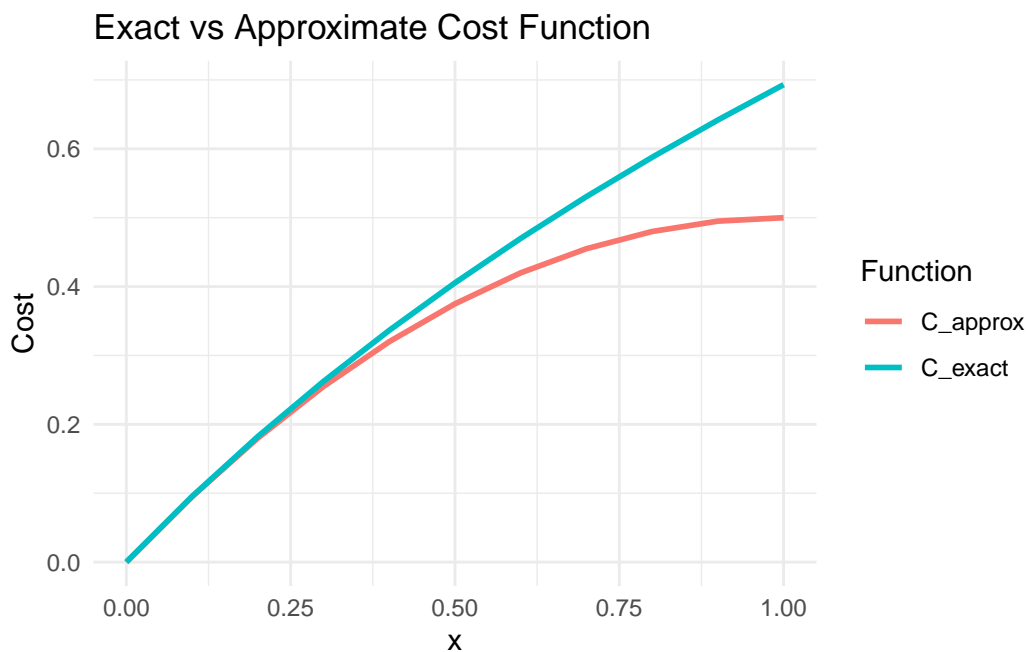
head(df_bus_C_compare, 11)
```

	x	C_exact	C_approx	C_error
1	0.0	0.00000000	0.000	0.0000000000
2	0.1	0.09531018	0.095	-0.0003101798
3	0.2	0.18232156	0.180	-0.0023215568
4	0.3	0.26236426	0.255	-0.0073642645
5	0.4	0.33647224	0.320	-0.0164722366
6	0.5	0.40546511	0.375	-0.0304651081
7	0.6	0.47000363	0.420	-0.0500036292
8	0.7	0.53062825	0.455	-0.0756282511
9	0.8	0.58778666	0.480	-0.1077866649
10	0.9	0.64185389	0.495	-0.1468538862
11	1.0	0.69314718	0.500	-0.1931471806


```
#plot
df_bus_C_compare_long <- pivot_longer(df_bus_C_compare,
                                       cols = c("C_exact", "C_approx"),
                                       names_to = "type",
                                       values_to = "cost")

plt_bus_C <- df_bus_C_compare_long %>%
  ggplot(aes(x = x, y = cost, color = type)) +
  geom_line(size = 1) +
  labs(title = "Exact vs Approximate Cost Function",
       x = "x",
       y = "Cost",
       color = "Function") +
  theme_minimal()

plt_bus_C
```



3. *Linear vs. Nonlinear Optimization: Using the Taylor Series expansions, approximate the profit function $\Pi(x) = R(x) - C(x)$. Compare the optimization results when using the linear approximations versus the original nonlinear functions. What are the differences, and when might it be more appropriate to use the approximation?*

The profit estimation is accurate for most values of x and begins to underestimate around x

~ 0.9 (note that underestimating profit is conservative and not usually a bad thing). It would be appropriate to use for estimations overall, and in production or sensitive applications for values of $x < 0.9$.

```
# combine rev and cost df
df_bus_P_compare <- merge(df_bus_C_compare,
                          df_bus_R_compare,
                          by = "x")

# calculate P
df_bus_P_compare <- df_bus_P_compare %>%
  mutate(P_exact = R_exact - C_exact,
         P_approx = R_approx - C_approx,
         P_error = P_approx - P_exact)

head(df_bus_P_compare, 10)
```

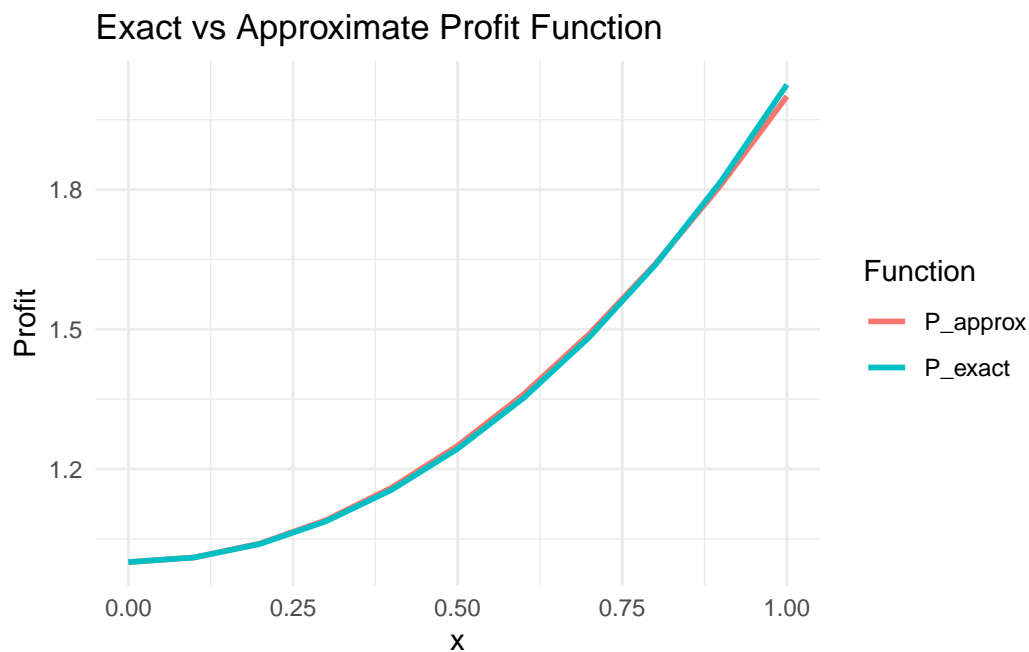
	x	C_exact	C_approx	C_error	R_exact	R_approx	R_error
1	0.0	0.00000000	0.000	0.0000000000	1.000000	1.000	0.0000000000
2	0.1	0.09531018	0.095	-0.0003101798	1.105171	1.105	-0.0001709181
3	0.2	0.18232156	0.180	-0.0023215568	1.221403	1.220	-0.0014027582
4	0.3	0.26236426	0.255	-0.0073642645	1.349859	1.345	-0.0048588076
5	0.4	0.33647224	0.320	-0.0164722366	1.491825	1.480	-0.0118246976
6	0.5	0.40546511	0.375	-0.0304651081	1.648721	1.625	-0.0237212707
7	0.6	0.47000363	0.420	-0.0500036292	1.822119	1.780	-0.0421188004
8	0.7	0.53062825	0.455	-0.0756282511	2.013753	1.945	-0.0687527075
9	0.8	0.58778666	0.480	-0.1077866649	2.225541	2.120	-0.1055409285
10	0.9	0.64185389	0.495	-0.1468538862	2.459603	2.305	-0.1546031112

	P_exact	P_approx	P_error
1	1.000000	1.00	0.0000000000
2	1.009861	1.01	0.0001392617
3	1.039081	1.04	0.0009187986
4	1.087495	1.09	0.0025054569
5	1.155352	1.16	0.0046475390
6	1.243256	1.25	0.0067438374
7	1.352115	1.36	0.0078848289
8	1.483124	1.49	0.0068755436
9	1.637754	1.64	0.0022457364
10	1.817749	1.81	-0.0077492250

```
#plot
df_bus_P_compare_long <- pivot_longer(df_bus_P_compare,
                                       cols = c("P_exact", "P_approx"),
                                       names_to = "type",
                                       values_to = "profit")

plt_bus_P <- df_bus_P_compare_long %>%
  ggplot(aes(x = x, y = profit, color = type)) +
  geom_line(size = 1) +
  labs(title = "Exact vs Approximate Profit Function",
       x = "x",
       y = "Profit",
       color = "Function") +
  theme_minimal()

plt_bus_P
```



2. Financial Modeling

Scenario: A financial analyst is modeling the risk associated with a new investment. The risk is proportional to the square root of the invested amount, modeled as $f(x) = \sqrt{x}$, where x

is the amount invested. However, to simplify calculations, the analyst wants to use a Taylor Series expansion to approximate this function for small investments.

1. *Maclaurin Series Expansion:* Derive the Taylor Series expansion around $x=0$ up to the second degree.

$x=0$ is undefined for \sqrt{x} because the derivative is undefined, so a Taylor series around $x = 1$ is used instead: $\sqrt{x} \approx 1 + \frac{x-1}{2}$

2. *Practical Application:* Use the derived series to approximate the risk for small investment amounts (e.g., when x is small). Compare the approximated risk with the actual function values for small and moderate investments. Discuss when this approximation might be useful in financial modeling.

The approximation is very close for most values of x between 0.5 and 1.5 with some deviation at the extremes as shown below. This would be very useful when simplifying calculations particularly around a typical normalized ($x=1$) investment amount.

3. *Optimization Scenario:* Suppose the goal is to minimize risk while maintaining a certain level of investment return. Using the Taylor Series approximation, suggest an optimal investment amount x that balances risk and return.

No return function is given so we can't optimize for risk and return. To minimize risk alone, x would have to equal zero, because all levels of investment have risk.

```
# define exact risk function
bus_risk_exact <- function(x) {sqrt(x)}

# define taylor series approximation function
bus_risk_approx <- function(x) {1 + 0.5 * (x-1) - (1/8) * (x-1)^2}

# compare from 0.5 to 1.5
bus_risk_x_vals <- seq(0.5, 1.5, by = 0.1)

df_bus_risk_compare <- data.frame(
  x = bus_risk_x_vals,
  risk_exact = bus_risk_exact(bus_risk_x_vals),
  risk_approx = bus_risk_approx(bus_risk_x_vals),
  risk_error = bus_risk_approx(bus_risk_x_vals) -
    bus_risk_exact(bus_risk_x_vals))

head(df_bus_risk_compare, 11)
```

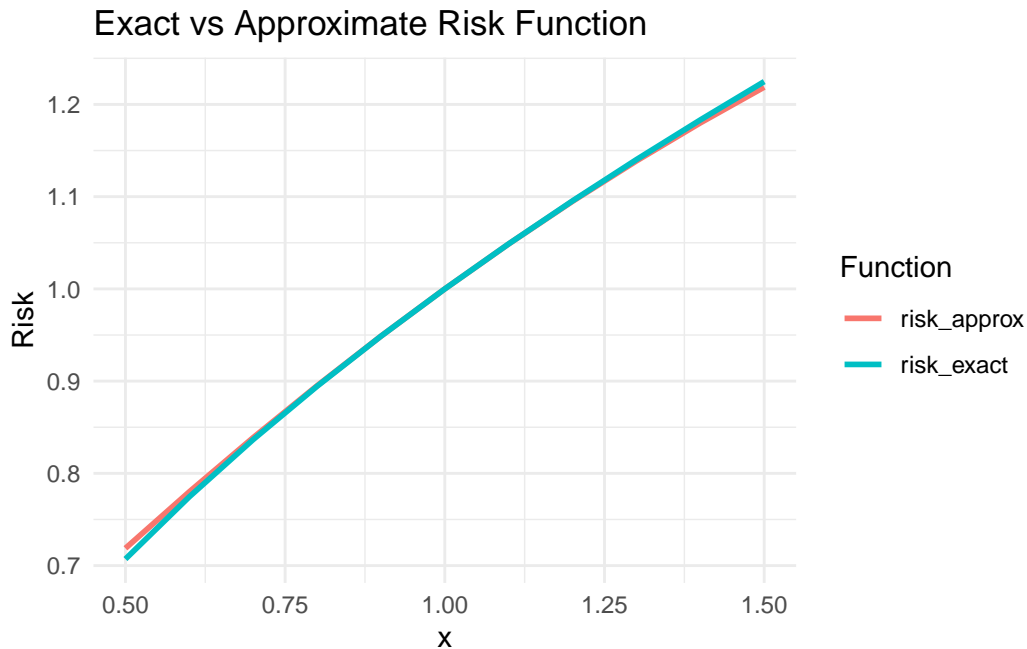
	x	risk_exact	risk_approx	risk_error
1	0.5	0.7071068	0.71875	1.164322e-02

2	0.6	0.7745967	0.78000	5.403331e-03
3	0.7	0.8366600	0.83875	2.089973e-03
4	0.8	0.8944272	0.89500	5.728090e-04
5	0.9	0.9486833	0.94875	6.670195e-05
6	1.0	1.0000000	1.00000	0.000000e+00
7	1.1	1.0488088	1.04875	-5.884817e-05
8	1.2	1.0954451	1.09500	-4.451150e-04
9	1.3	1.1401754	1.13875	-1.425425e-03
10	1.4	1.1832160	1.18000	-3.215957e-03
11	1.5	1.2247449	1.21875	-5.994871e-03

```
#plot
df_bus_risk_compare_long <- pivot_longer(df_bus_risk_compare,
                                          cols = c("risk_exact",
                                                    "risk_approx"),
                                          names_to = "type",
                                          values_to = "risk")

plt_bus_risk <- df_bus_risk_compare_long %>%
  ggplot(aes(x = x, y = risk, color = type)) +
  geom_line(size = 1) +
  labs(title = "Exact vs Approximate Risk Function",
       x = "x",
       y = "Risk",
       color = "Function") +
  theme_minimal()

plt_bus_risk
```



3. Inventory Management

Scenario: In a manufacturing process, the demand for a product decreases as the price increases, modeled by $D(p) = 1 - p$, where p is the price. The cost associated with producing and selling the product is modeled as $C(p) = e^p$. The company wants to maximize its profit, which is the difference between revenue and cost.

1. *Taylor Series Expansion: Expand the cost function $C(p) = e^p$ into a Taylor Series around $p=0$ up to the second degree. Discuss why approximating the cost function might be useful in a pricing strategy.*

The Maclaurin series for e^p at the second degree is $e^p \approx 1 + p + \frac{p^2}{2}$.

This approximation is accurate for values of $p < 0.5$ (see plot below).

It might still be useful in a pricing strategy: since covering cost is only one component of pricing (with significant buffers for desired margin, strategic and market factors, and uncertainty built in), a good estimate like this can suffice and avoiding logs would simplify the optimization of the full pricing model.

```
# define exact cost function
inv_C_exact <- function(p) {exp(p)}

# define maclaurin series approximation function
```

```

inv_C_approx <- function(p) {1 + p + (p^2)/2}

# compare
inv_C_p_vals <- seq(0, 1, by = 0.1)

df_inv_C_compare <- data.frame(
  p = inv_C_p_vals,
  C_exact = inv_C_exact(inv_C_p_vals),
  C_approx = inv_C_approx(inv_C_p_vals),
  C_error = inv_C_approx(inv_C_p_vals) - inv_C_exact(inv_C_p_vals))

head(df_inv_C_compare, 11)

```

	p	C_exact	C_approx	C_error
1	0.0	1.000000	1.000	0.0000000000
2	0.1	1.105171	1.105	-0.0001709181
3	0.2	1.221403	1.220	-0.0014027582
4	0.3	1.349859	1.345	-0.0048588076
5	0.4	1.491825	1.480	-0.0118246976
6	0.5	1.648721	1.625	-0.0237212707
7	0.6	1.822119	1.780	-0.0421188004
8	0.7	2.013753	1.945	-0.0687527075
9	0.8	2.225541	2.120	-0.1055409285
10	0.9	2.459603	2.305	-0.1546031112
11	1.0	2.718282	2.500	-0.2182818285

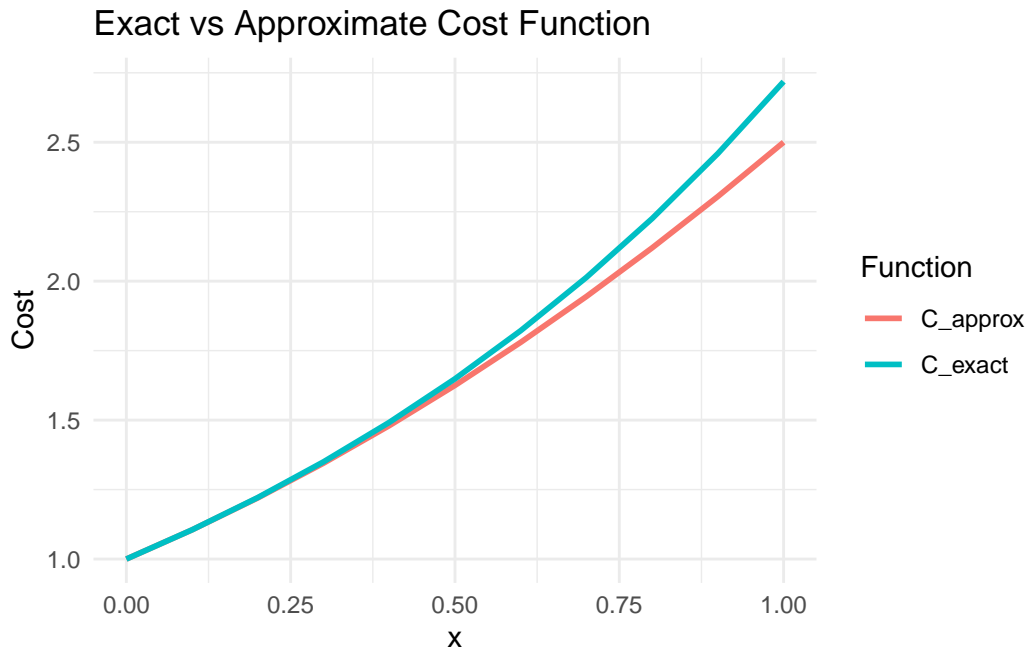
```

#plot
df_inv_C_compare_long <- pivot_longer(df_inv_C_compare,
                                       cols = c("C_exact", "C_approx"),
                                       names_to = "type",
                                       values_to = "cost")

plt_inv_C <- df_inv_C_compare_long %>%
  ggplot(aes(x = p, y = cost, color = type)) +
  geom_line(size = 1) +
  labs(title = "Exact vs Approximate Cost Function",
       x = "x",
       y = "Cost",
       color = "Function") +
  theme_minimal()

plt_inv_C

```



2. *Approximating Profit: Using the Taylor Series expansion, approximate the profit function $\Pi(p) = pD(p) - C(p)$. Compare the results when using the original nonlinear cost function versus the approximated cost function. What differences do you observe, and when might the approximation be sufficient?*

The approximated profit is largely accurate for small values of $p < 0.5$ but overestimates significantly above that.

Overestimating profit is a bigger problem than underestimating, but it could still be useful in planning with an adjustment for error.

```
glimpse(df_inv_C_compare)
```

```
Rows: 11
Columns: 4
$ p      <dbl> 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
$ C_exact <dbl> 1.000000, 1.105171, 1.221403, 1.349859, 1.491825, 1.648721, 1~
$ C_approx <dbl> 1.000, 1.105, 1.220, 1.345, 1.480, 1.625, 1.780, 1.945, 2.120~
$ C_error  <dbl> 0.0000000000, -0.0001709181, -0.0014027582, -0.0048588076, -0~
```

```
# calculate D and P_exact, P_est
df_inv_P_compare <- df_inv_C_compare %>%
  mutate(D = 1-p,
```



```

    Rev = p * D,
    P_exact = Rev - C_exact,
    P_approx = Rev - C_approx,
    P_error = P_approx - P_exact)

```

```
head(df_inv_P_compare, 11)
```

	p	C_exact	C_approx	C_error	D	Rev	P_exact	P_approx	P_error
1	0.0	1.000000	1.000	0.00000000000	1.0	0.00	-1.000000	-1.000	0.00000000000
2	0.1	1.105171	1.105	-0.0001709181	0.9	0.09	-1.015171	-1.015	0.0001709181
3	0.2	1.221403	1.220	-0.0014027582	0.8	0.16	-1.061403	-1.060	0.0014027582
4	0.3	1.349859	1.345	-0.0048588076	0.7	0.21	-1.139859	-1.135	0.0048588076
5	0.4	1.491825	1.480	-0.0118246976	0.6	0.24	-1.251825	-1.240	0.0118246976
6	0.5	1.648721	1.625	-0.0237212707	0.5	0.25	-1.398721	-1.375	0.0237212707
7	0.6	1.822119	1.780	-0.0421188004	0.4	0.24	-1.582119	-1.540	0.0421188004
8	0.7	2.013753	1.945	-0.0687527075	0.3	0.21	-1.803753	-1.735	0.0687527075
9	0.8	2.225541	2.120	-0.1055409285	0.2	0.16	-2.065541	-1.960	0.1055409285
10	0.9	2.459603	2.305	-0.1546031112	0.1	0.09	-2.369603	-2.215	0.1546031112
11	1.0	2.718282	2.500	-0.2182818285	0.0	0.00	-2.718282	-2.500	0.2182818285

```

#plot
df_inv_P_compare_long <- pivot_longer(df_inv_P_compare,
                                       cols = c("P_exact", "P_approx"),
                                       names_to = "type",
                                       values_to = "profit")

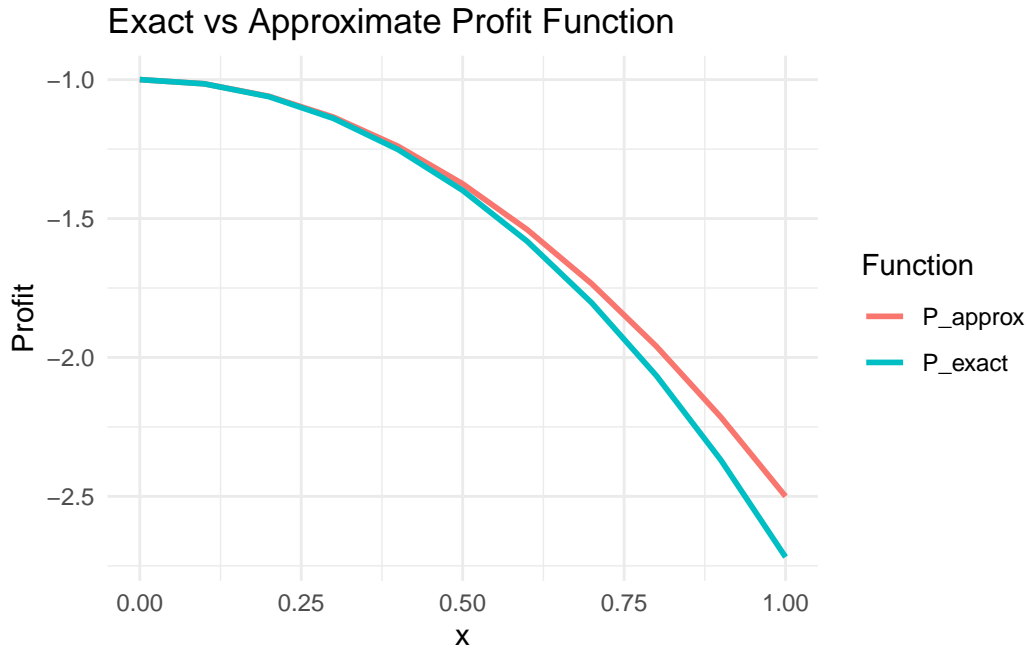
```

```

plt_inv_P <- df_inv_P_compare_long %>%
  ggplot(aes(x = p, y = profit, color = type)) +
  geom_line(size = 1) +
  labs(title = "Exact vs Approximate Profit Function",
       x = "x",
       y = "Profit",
       color = "Function") +
  theme_minimal()

```

```
plt_inv_P
```



3. *Pricing Strategy: Based on the Taylor Series approximation, suggest a pricing strategy that could maximize profit. Explain how the Taylor Series approximation helps in making this decision.*

Based on this model, I would not sell this product because it is unprofitable at any price. Revenue is maximized at $p = 0.5$ in the table below, but cost grows exponentially since $C = e^p$ and outpaces revenue $R = p * (1 - p)$.

```
head(df_inv_P_compare, 11)
```

	p	C_exact	C_approx	C_error	D	Rev	P_exact	P_approx	P_error
1	0.0	1.000000	1.000	0.00000000000	1.0	0.00	-1.000000	-1.000	0.00000000000
2	0.1	1.105171	1.105	-0.0001709181	0.9	0.09	-1.015171	-1.015	0.0001709181
3	0.2	1.221403	1.220	-0.0014027582	0.8	0.16	-1.061403	-1.060	0.0014027582
4	0.3	1.349859	1.345	-0.0048588076	0.7	0.21	-1.139859	-1.135	0.0048588076
5	0.4	1.491825	1.480	-0.0118246976	0.6	0.24	-1.251825	-1.240	0.0118246976
6	0.5	1.648721	1.625	-0.0237212707	0.5	0.25	-1.398721	-1.375	0.0237212707
7	0.6	1.822119	1.780	-0.0421188004	0.4	0.24	-1.582119	-1.540	0.0421188004
8	0.7	2.013753	1.945	-0.0687527075	0.3	0.21	-1.803753	-1.735	0.0687527075
9	0.8	2.225541	2.120	-0.1055409285	0.2	0.16	-2.065541	-1.960	0.1055409285
10	0.9	2.459603	2.305	-0.1546031112	0.1	0.09	-2.369603	-2.215	0.1546031112
11	1.0	2.718282	2.500	-0.2182818285	0.0	0.00	-2.718282	-2.500	0.2182818285

4. Economic Forecasting

Scenario: An economist is forecasting economic growth, which can be modeled by the logarithmic function $G(x) = \ln(1 + x)$ where x represents investment in infrastructure. The government wants to predict growth under different levels of investment.

1. Maclaurin Series Expansion: Derive the Maclaurin Series expansion of up to the second degree. Explain the significance of using this approximation for small values of x in economic forecasting.

The Maclaurin series would be $\ln(1 + x) \approx x - \frac{x^2}{2}$

It would be useful to simplify modeling growth estimates for small values of x .

2. Approximation of Growth: Use the Taylor Series to approximate the growth for small investments. Compare this approximation with the actual growth function. Discuss the accuracy of the approximation for different ranges of x .

This approximation is accurate for small values of $x < 0.25$ and would be useful for modeling the impact of small amounts of investment. It does increasingly underestimate growth for larger values of x , which might be conservative and acceptable for planning.

```
# define exact growth function
econ_growth_exact <- function(x) {log(1+x)}

# define taylor series approximation function
econ_growth_approx <- function(x) {x - (x^2)/2}

# compare from 0 to 1
econ_growth_x_vals <- seq(0, 1, by = 0.1)

df_econ_growth_compare <- data.frame(
  x = econ_growth_x_vals,
  growth_exact = econ_growth_exact(econ_growth_x_vals),
  growth_approx = econ_growth_approx(econ_growth_x_vals),
  growth_error = econ_growth_approx(econ_growth_x_vals) -
    econ_growth_exact(econ_growth_x_vals))

head(df_econ_growth_compare, 11)
```

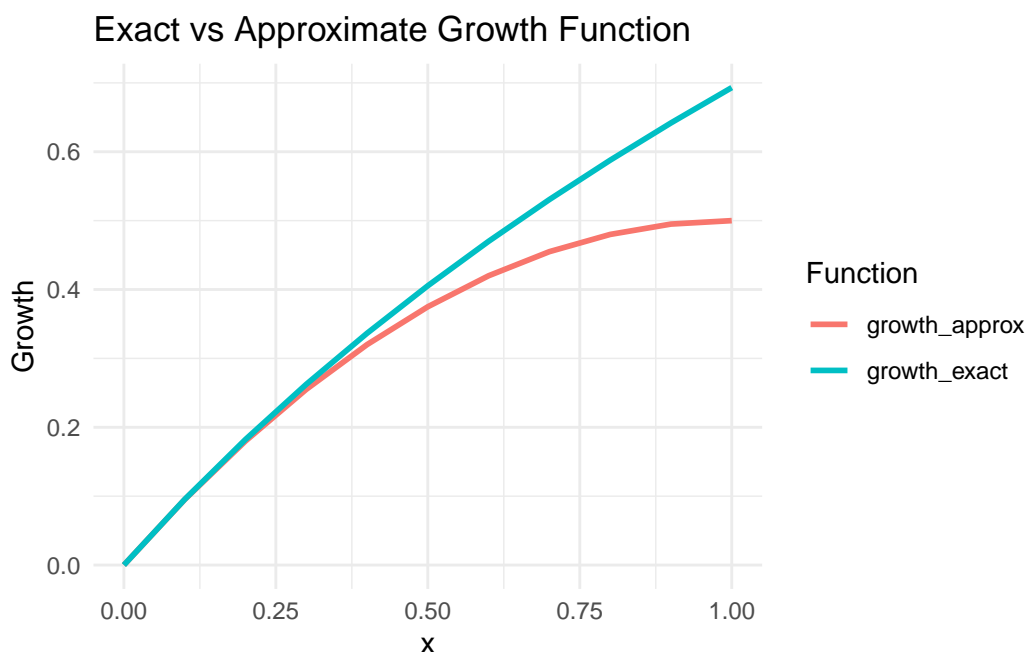
	x	growth_exact	growth_approx	growth_error
1	0.0	0.00000000	0.000	0.0000000000
2	0.1	0.09531018	0.095	-0.0003101798
3	0.2	0.18232156	0.180	-0.0023215568

4	0.3	0.26236426	0.255	-0.0073642645
5	0.4	0.33647224	0.320	-0.0164722366
6	0.5	0.40546511	0.375	-0.0304651081
7	0.6	0.47000363	0.420	-0.0500036292
8	0.7	0.53062825	0.455	-0.0756282511
9	0.8	0.58778666	0.480	-0.1077866649
10	0.9	0.64185389	0.495	-0.1468538862
11	1.0	0.69314718	0.500	-0.1931471806

```
#plot
df_econ_growth_compare_long <- pivot_longer(df_econ_growth_compare,
                                             cols = c("growth_exact",
                                                    "growth_approx"),
                                             names_to = "type",
                                             values_to = "growth")

plt_econ_growth <- df_econ_growth_compare_long %>%
  ggplot(aes(x = x, y = growth, color = type)) +
  geom_line(size = 1) +
  labs(title = "Exact vs Approximate Growth Function",
       x = "x",
       y = "Growth",
       color = "Function") +
  theme_minimal()

plt_econ_growth
```



3. *Policy Recommendation: Using the approximation, recommend a level of investment that could achieve a target growth rate. Discuss the limitations of using Taylor Series approximations for such policy recommendations.*

Using the table below, we can make recommendations to achieve targeted growth rates. For example, for a targeted growth rate of 25%, we would recommend a normalized investment amount of ~0.3.

The limitations of Taylor series approximations are that they are most accurate near the expansion point (here $x = 0$). In this case, higher values of x would overestimate the investment needed to achieve a targeted growth rate.

```
head(select(df_econ_growth_compare, x, growth_approx), 11)
```

	x	growth_approx
1	0.0	0.000
2	0.1	0.095
3	0.2	0.180
4	0.3	0.255
5	0.4	0.320
6	0.5	0.375
7	0.6	0.420
8	0.7	0.455
9	0.8	0.480

10	0.9	0.495
11	1.0	0.500

5. Profit, Cost, & Pricing

1. Profit Maximization

Scenario: A company produces two products, A and B. The profit function for the two products is given by: $\pi(x, y)$

Where: x is the quantity of Product A produced and sold, y is the quantity of Product B produced and sold, π is the profit in dollars

Find all local maxima, local minima, and saddle points for the profit function $\pi(x, y)$. Write your answer(s) in the form (x, y) . Separate multiple points with a comma. Discussion: Discuss the implications of the results for the company's production strategy. Which production levels maximize profit, and what risks are associated with the saddle points?