DATA 605: Computational Mathematics

Gabriel Campos

Last edited December 12, 2024

Contents

Library **2**

Instructions: **2**

Problem 1: **3**

Data Load 3

Part 1: 3

1. 3

X ~ Sales 4

Initial analysis 6

fitdstr Sales Solution 7

Mean and Variance (X~Sales) Solution 11

Answer 12

Y ~ Inventory Levels 12

Initial analysis 14

fitdistr Inventory Levels 15

Mean and Variance (Y~Inventory Levels) 16

Answer 17

Z ~ Lead Time 17

Mean and Variance Lead Time Solution 19

Answer 20

2. 20

Answer 20

Part 2: **21**

1. 21

i. 21

ii. 22

iii. 22

Answers 22

2. 22

Contingency, Joint and Marginal Table 23

Fisher and Chi Square 24

Problem 2 **25**

Part 1 25

1. 25

Univariate Descriptive Statistics 25

Scatterplots and Price 30

Correlation Plot 36

Hypothesis Test 37

2. 39
Discussion: 39
Part 2 39
1. 39
Linear Regression Model 39
Inverted Correlation Matrix 40
Precision Matrix Diagonals 41
LU Decomposition 41
Part 3: 42
1. 43
Right-skewed | Exponential Distr. 43
Samples 43
5th and 95th Percentile 44
95 % Confidence Interval 44
2. 45
Discussion: 45
Part 4 45
1. 45
2. 48

Referenced sites. 49

Library

```
library(corrplot)
library(dplyr)
library(e1071)
library(ggcorrplot)
library(ggplot2)
library(Hmisc)
library(kableExtra)
library(MASS)
library(Matrix)
library(matrixcalc)
library(readr)
```

Final Examination: Business Analytics and Data Science

Instructions:

You are required to complete this take-home final examination by the end of the last week of class. Your solutions should be uploaded in **pdf** format as a knitted document (with graphs, content, commentary, etc. in the pdf). This project will showcase your ability to apply the concepts learned throughout the course.

The dataset you will use for this examination is provided as retail data.csv, which contains the following variables:

- Product_ID: Unique identifier for each product.
- Sales: Simulated sales numbers (in dollars).
- Inventory_Levels: Inventory levels for each product.
- Lead_Time_Days: The lead time in days for each product.
- Price: The price of each product.
- Seasonality_Index: An index representing seasonality.

Problem 1:

Business Risk and Revenue Modeling

Context: You are a data scientist working for a retail chain that models sales, inventory levels, and the impact of pricing and seasonality on revenue. Your task is to analyze various distributions that can describe sales variability and forecast potential revenue.

Data Load

```
retail_df <- read_csv("synthetic_retail_data.csv")
```

Part 1:

Empirical and Theoretical Analysis of Distributions (5 Points)

Task:

1.

Generate and Analyze Distributions:

- **X ~ Sales:** Consider the Sales variable from the dataset. Assume it follows a Gamma distribution and estimate its shape and scale parameters using the fitdistr function from the MASS package.
- **Y ~ Inventory Levels:** Assume that the sum of inventory levels across similar products follows a Lognormal distribution. Estimate the parameters for this distribution.
- **Z ~ Lead Time:** Assume that Lead_Time_Days follows a Normal distribution. Estimate the mean and standard deviation. Calculate Empirical Expected Value and Variance:

```
head(retail_df)
```

```
## # A tibble: 6 x 6
##   Product_ID Sales Inventory_Levels Lead_Time_Days Price Seasonality_Index
##        <dbl> <dbl>            <dbl>          <dbl> <dbl>             <dbl>
## 1          1  158.             367.           6.31  18.8             1.18
## 2          2  279.             427.           5.80  26.1             0.857
## 3          3  699.             408.           3.07  22.4             0.699
## 4          4 1832.             392.           3.53  27.1             0.698
## 5          5  460.             448.          10.8   18.3             0.841
## 6          6 1693.             547.          10.1   23.5             1.13
```

```
glimpse(retail_df)
```

```
## Rows: 200
## Columns: 6
## $ Product_ID        <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ Sales             <dbl> 158.43952, 278.99020, 698.85868, 1832.39467, 459.703~
## $ Inventory_Levels  <dbl> 367.4421, 426.6512, 407.6394, 392.3912, 448.3120, 54~
## $ Lead_Time_Days    <dbl> 6.314587, 5.800673, 3.071936, 3.534253, 10.802241, 1~
## $ Price             <dbl> 18.795197, 26.089636, 22.399985, 27.092013, 18.30782~
## $ Seasonality_Index <dbl> 1.1839497, 0.8573051, 0.6986774, 0.6975404, 0.840725~
```

```
summary(retail_df)
```

```
##    Product_ID         Sales         Inventory_Levels Lead_Time_Days
##  Min.   :  1.00   Min.   :  25.57   Min.   : 67.35   Min.   : 0.491
##  1st Qu.: 50.75   1st Qu.: 284.42   1st Qu.:376.51   1st Qu.: 5.291
##  Median :100.50   Median : 533.54   Median :483.72   Median : 6.765
##  Mean   :100.50   Mean   : 636.92   Mean   :488.55   Mean   : 6.834
##  3rd Qu.:150.25   3rd Qu.: 867.58   3rd Qu.:600.42   3rd Qu.: 8.212
##  Max.   :200.00   Max.   :2447.49   Max.   :858.79   Max.   :12.722
##      Price        Seasonality_Index
##  Min.   : 5.053   Min.   :0.3305
##  1st Qu.:16.554   1st Qu.:0.8475
##  Median :19.977   Median :0.9762
##  Mean   :19.560   Mean   :0.9829
##  3rd Qu.:22.924   3rd Qu.:1.1205
##  Max.   :29.404   Max.   :1.5958
```

$X \sim$ Sales **$X \sim$ Sales:** Consider the Sales variable from the dataset. Assume it follows a Gamma distribution and estimate its shape and scale parameters using the fitdistr function from the MASS package.

```r
# Isolate Sales data
sales_retail_df <- retail_df$Sales
summary(sales_retail_df)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   25.57  284.42  533.54  636.92  867.58 2447.49
```

```r
sum(sales_retail_df<0)
```

```
## [1] 0
```

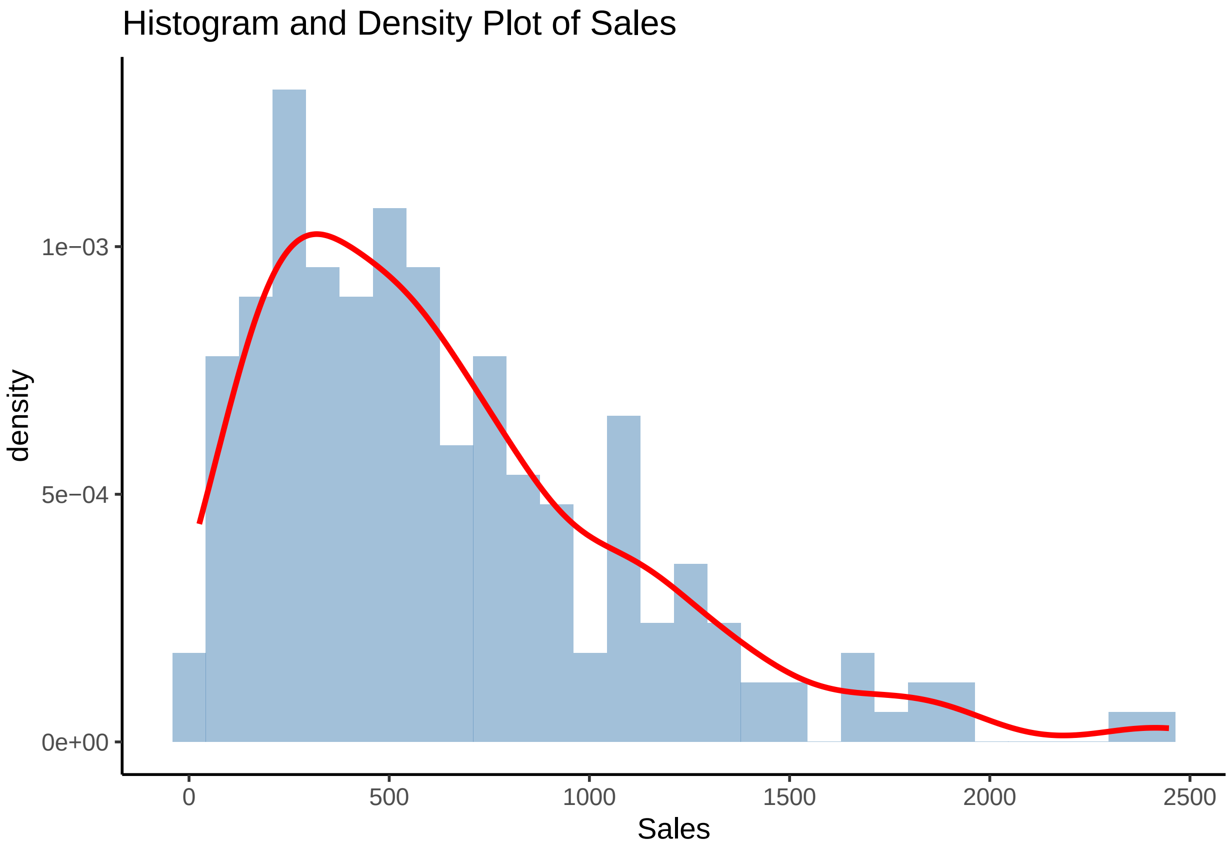
```r
sum(is.na(sales_retail_df))
```

```
## [1] 0
```

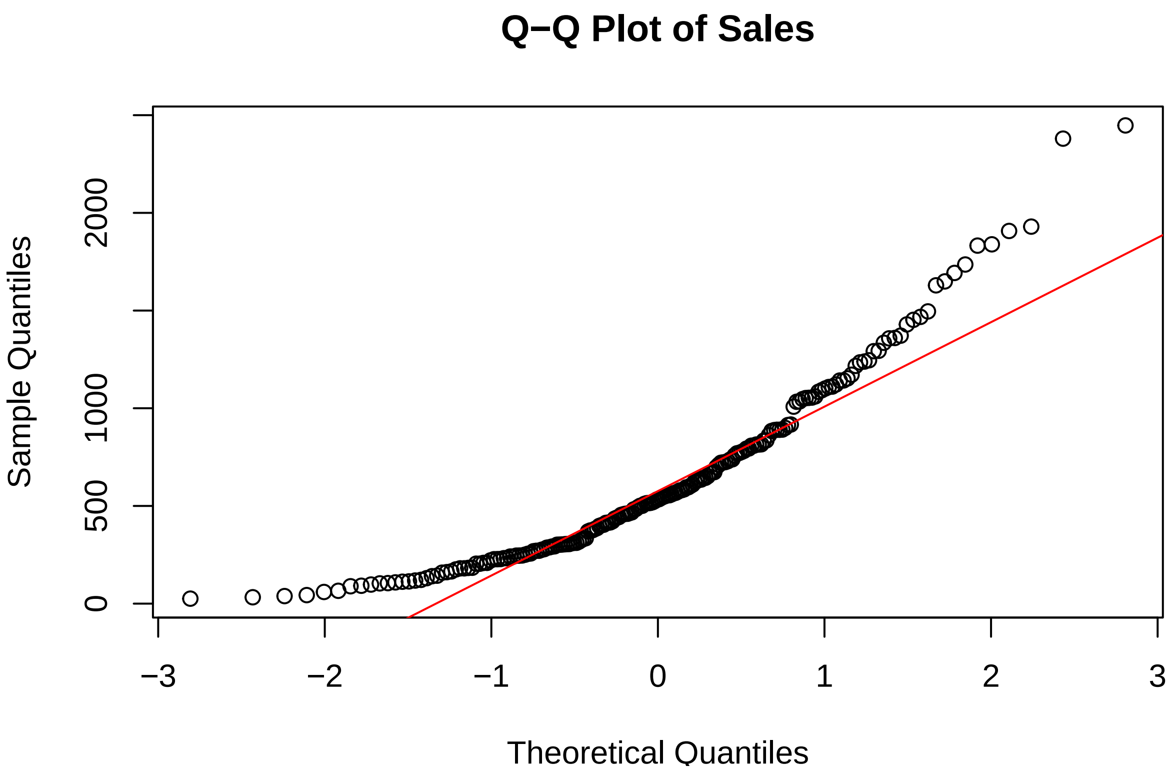
```r
shapiro.test(sales_retail_df)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sales_retail_df
## W = 0.90377, p-value = 4.397e-10
```

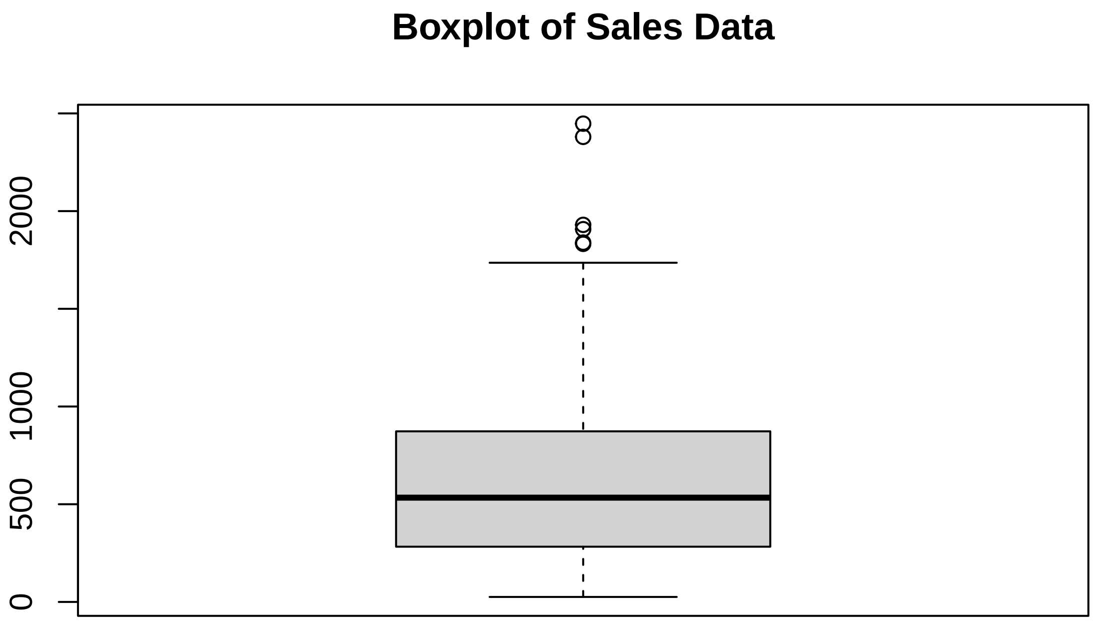
```r
ggplot(retail_df, aes(x = Sales)) +
  geom_histogram(aes(y = ..density..),
                 bins = 30, fill = "steelblue",
                 alpha = 0.5) +
  geom_density(color = "red", size = 1) +
  ggtitle("Histogram and Density Plot of Sales") +
  theme_classic()
```



```
qqnorm(sales_retail_df,
       main = "Q-Q Plot of Sales")
qqline(sales_retail_df,
       col = "red")
```



```
boxplot(sales_retail_df,
        main = "Boxplot of Sales Data")
```



Initial analysis

- For our *Sales* data our $Mean > Median$ (636.92 > 533.54) which indicates that our data is right skewed and not normalized. This is supported by our Histogram, our Q-Q plot and the Shapiro test's $p - value$ of less than 0.05.
- No NAs are noted with the *Sales* data
- Our range for the values within *Sales* is 25.57 to 2447.49, encompassing a wide range.

- Our Box plot indicates that there are outliers, primarily for values > 1000

fitdstr Sales Solution Assume $X \sim Gamma(\alpha, \beta)$ the parameter *"gamma"* will be used with *fitdistr()*.

```
sales_gamma_fit <- fitdistr(sales_retail_df, "gamma")
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```
print(sales_gamma_fit)
```

```
##       shape           rate
##   1.8349640762   0.0028810166
##  (0.1511756159) (0.0002556985)
```

Considering no NAs or negative values were noted in our original data set, the *NaNs produced* warning, is likely a result of the right-skewed data or from our outliers. I will remove the outliers to see if it removes the warning. Regardless, dealing with these outliers should improve precision.

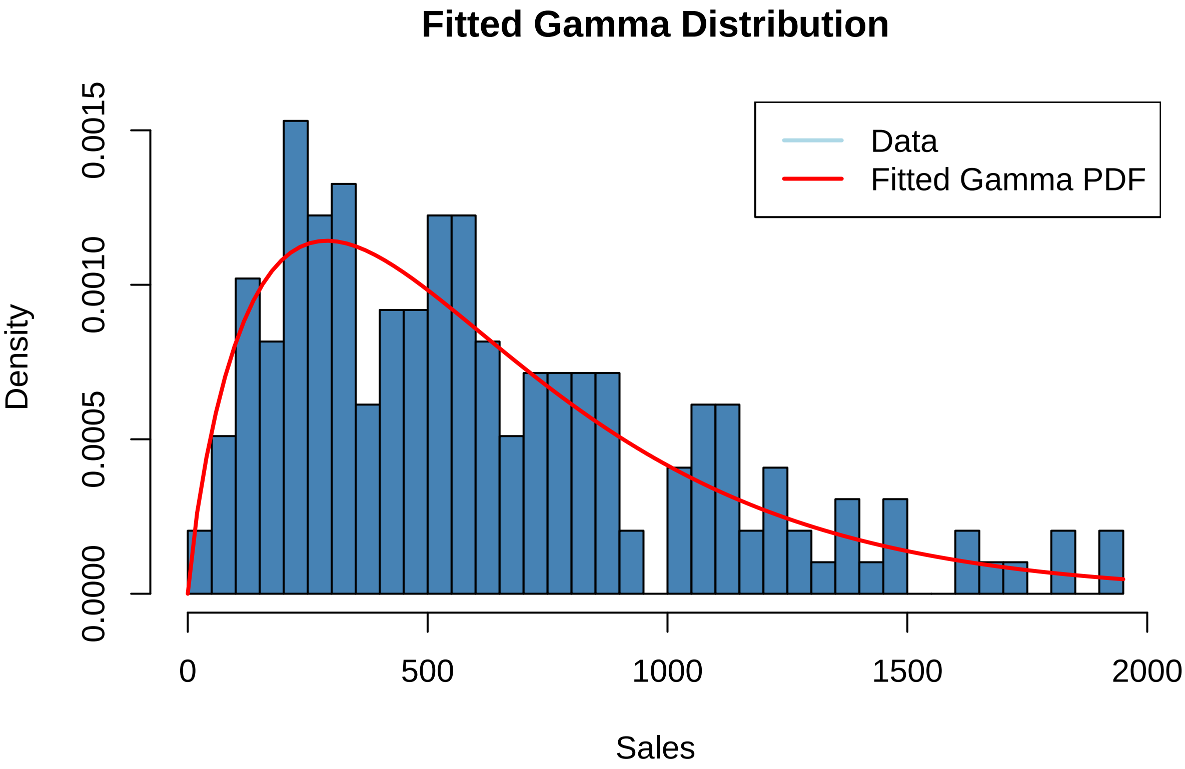
The below steps should remove values above our 99% quantile or below the 1%

```
# compute quantiles at 1% and and 99%
sales_retail_quantiles <-
  quantile(sales_retail_df, probs =c(0.01,0.99))
# remove outliers below the 1% and above 99%
sales_retail_df_clean<- sales_retail_df[
  sales_retail_df >= sales_retail_quantiles[1] &
    sales_retail_df <= sales_retail_quantiles[2]
]
sales_gamma_fit_clean <- fitdistr(sales_retail_df_clean, "gamma")
print(sales_gamma_fit_clean)
```

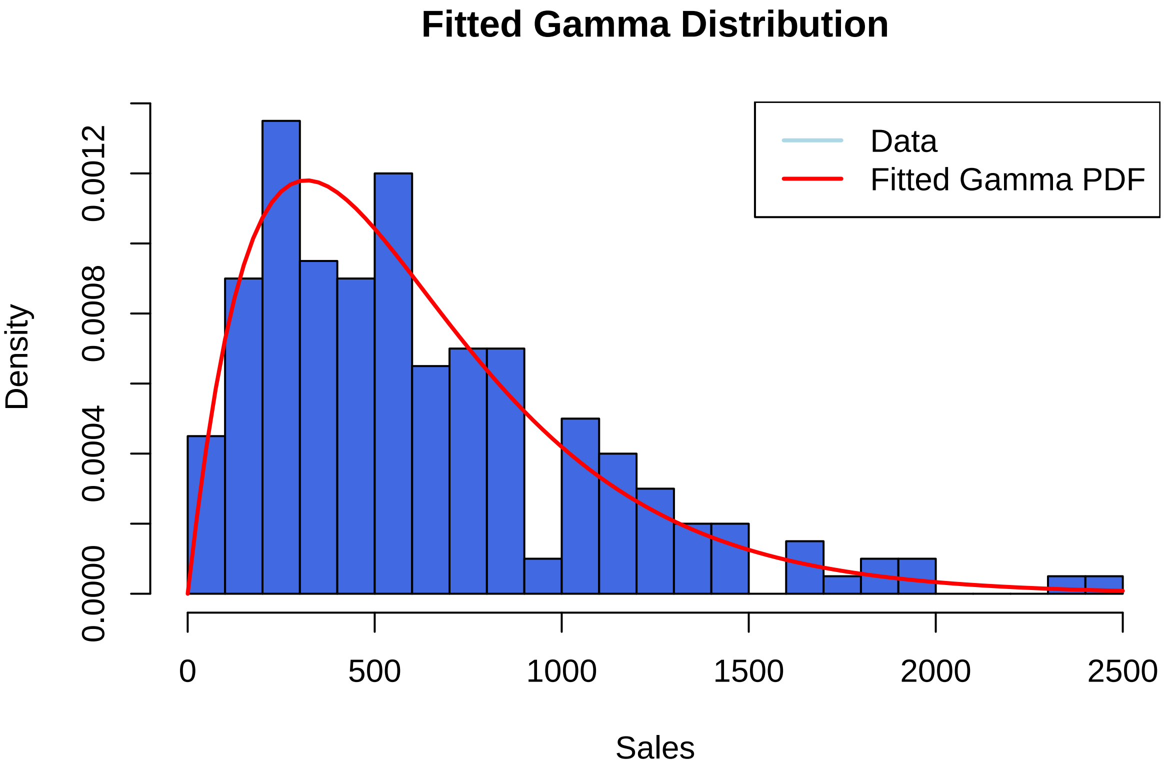
```
##       shape           rate
##   2.0323543224   0.0032518379
##  (0.1724396139) (0.0002975715)
```

The cleaned model still creates an error therefore I would like to see visually how well the values fit.

```
hist(sales_retail_df_clean,
     breaks = 30,
     probability = TRUE,
     main = "Fitted Gamma Distribution",
     xlab = "Sales",
     col = "steelblue")

curve(dgamma(x,
             shape = 1.8349640762,
             rate = 0.0028810166),
      col = "red",
      lwd = 2,
      add = TRUE)

legend("topright",
       legend = c("Data",
                  "Fitted Gamma PDF"),
       col = c("lightblue",
               "red"), lwd = 2)
```



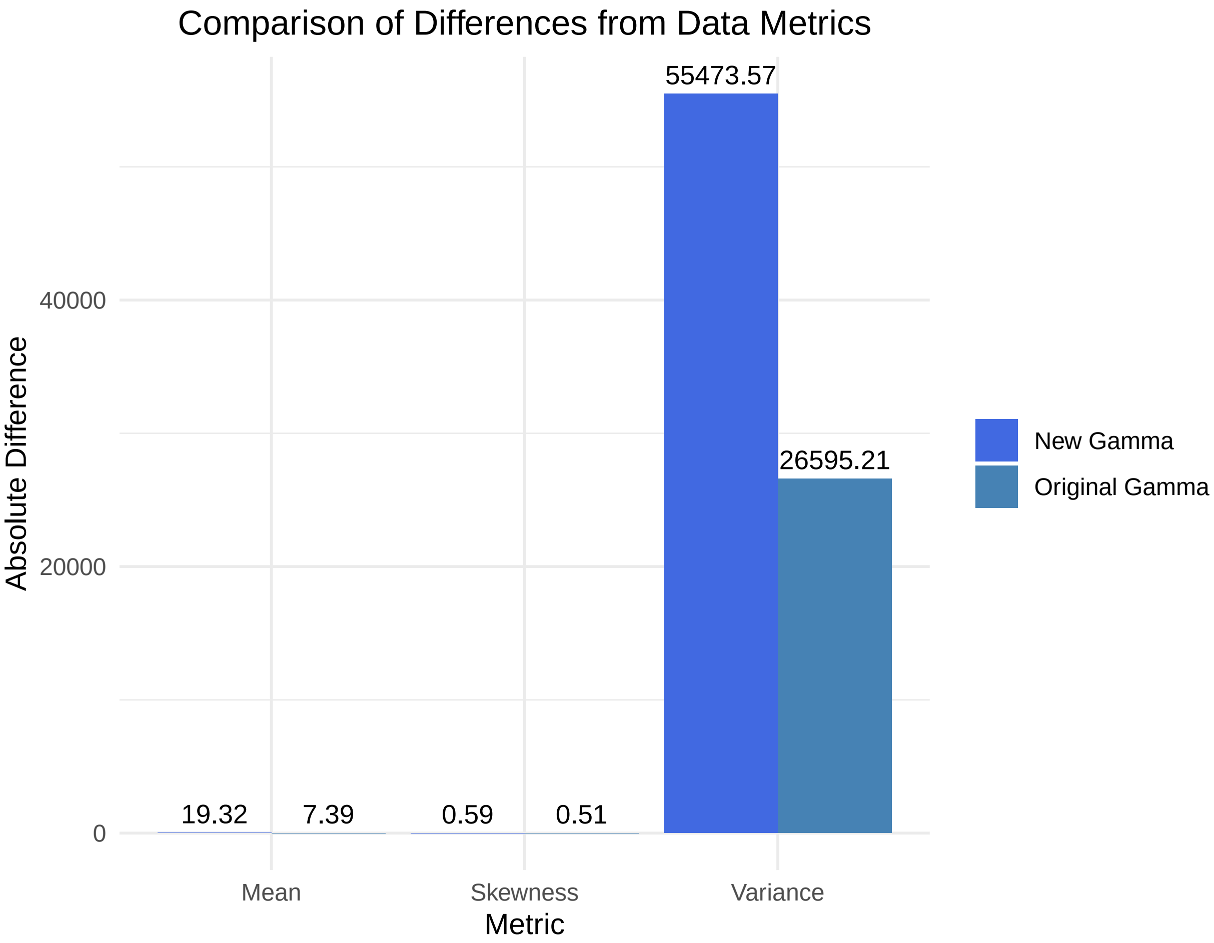
```r
hist(sales_retail_df,
     breaks = 30,
     probability = TRUE,
     main = "Fitted Gamma Distribution",
     xlab = "Sales",
     col = "royalblue")

curve(dgamma(x,
             shape = 2.0323543224,
             rate = 0.0032518379 ),
      col = "red",
      lwd = 2,
      add = TRUE)

legend("topright",
       legend = c("Data",
                  "Fitted Gamma PDF"),
       col = c("lightblue",
               "red"), lwd = 2)
```



Visually, both values seem to match the *Sales* behavior. By calculating the absolute difference in mean, skewness and variance, I might get a better indication on which gamma distribution and gamma parameters, better emulates the datas behavior.

```
cmp_metrics <- c("Mean", "Variance", "Skewness")

cmp_data_values <- c(617.595, 165599.6, 0.8891198)

# Original Gamma differences
cmp_shape1 <- 2.0323543224

cmp_rate1 <- 0.0032518379

cmp_original_gamma_values <-
  c(cmp_shape1 / cmp_rate1,
    cmp_shape1 / (cmp_rate1^2),
    2 / sqrt(cmp_shape1))

cmp_original_differences <-
  abs(cmp_original_gamma_values - cmp_data_values)

# New Gamma differences
cmp_shape2 <- 1.8349640762
cmp_rate2 <- 0.0028810166

cmp_new_gamma_values <-
  c(cmp_shape2 / cmp_rate2,
    cmp_shape2 / (cmp_rate2^2),
    2 / sqrt(cmp_shape2))
cmp_new_differences <-
```

```
  abs(cmp_new_gamma_values - cmp_data_values)

# Prepare data for ggplot
plot_data <- data.frame(
  Metric = rep(cmp_metrics,
               times = 2),
  Difference =
    c(cmp_original_differences,
                 cmp_new_differences),
  Gamma = rep(c("Original Gamma",
               "New Gamma"),
              each = length(cmp_metrics))
)

# Create the clustered bar plot
ggplot(plot_data, aes(x = Metric, y = Difference, fill = Gamma)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  geom_text(aes(label = round(Difference, 2)),
            position = position_dodge(width = 0.9),
            vjust = -0.5, size = 3.5) +
  labs(
    title = "Comparison of Differences from Data Metrics",
    x = "Metric",
    y = "Absolute Difference"
  ) +
  scale_fill_manual(values = c("Original Gamma" = "steelblue", "New Gamma" = "royalblue")) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_blank()
  )
```



Based on these results the original gamma parameters of 1.8349640762 and 0.0028810166 are the more accurate fit.

Mean and Variance (X~Sales) Solution Our Empirical mean and variance is just

```
sales_empirical_mean<-mean(sales_retail_df)
sales_empirical_var<-var(sales_retail_df)
```

and our theoretical mean and variance is calculate as

$\mu_{gamma} = \frac{\alpha}{\beta}$

$\sigma^2_{gamma} = \frac{\alpha}{\beta^2}$ or $\frac{shape}{rate^2}$

```
sales_gamma_shape <- sales_gamma_fit$estimate["shape"]
sales_gamma_rate <- sales_gamma_fit$estimate["rate"]

sales_theoretical_gamma_mean <- sales_gamma_shape / sales_gamma_rate
sales_theoretical_gamma_var <- sales_gamma_shape/ (sales_gamma_rate^2)
```

```
comparison_Sales <- data.frame(
  Metric = c("Mean", "Variance"),
  Empirical = c(sales_empirical_mean, sales_empirical_var),
  Theoretical = c(sales_theoretical_gamma_mean, sales_theoretical_gamma_var)
)
```

Answer The shape parameter of 1.8349640762 and the rate parameter of 0.0028810166 define the best-fit Gamma distribution for the data.

Y ~ Inventory Levels Y ~ Inventory Levels: Assume that the sum of inventory levels across similar products follows a Lognormal distribution. Estimate the parameters for this distribution.

```r
inv_retail_df <- retail_df$Inventory_Levels
summary(inv_retail_df)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   67.35  376.51  483.72  488.55  600.42  858.79
```

```r
sum(inv_retail_df<0)
```

```
## [1] 0
```

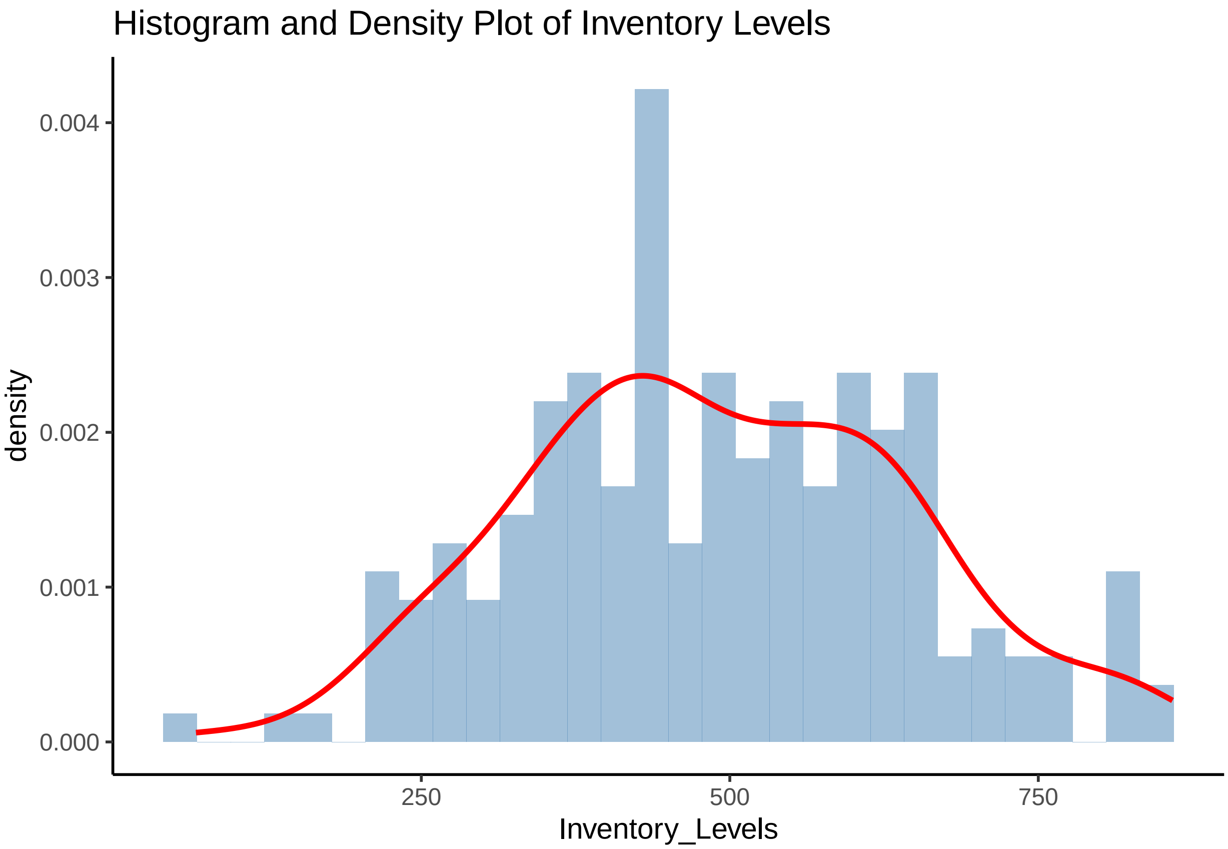
```r
sum(is.na(inv_retail_df))
```

```
## [1] 0
```

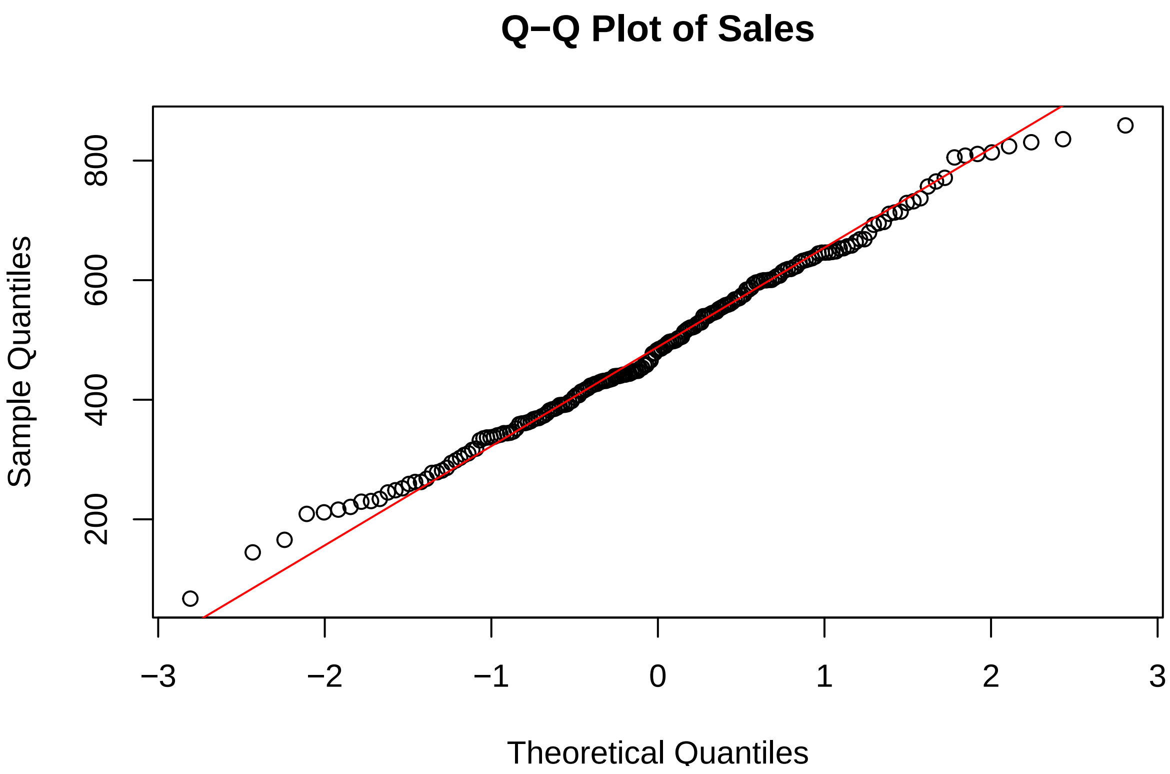
```r
shapiro.test(inv_retail_df)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  inv_retail_df
## W = 0.99303, p-value = 0.4646
```

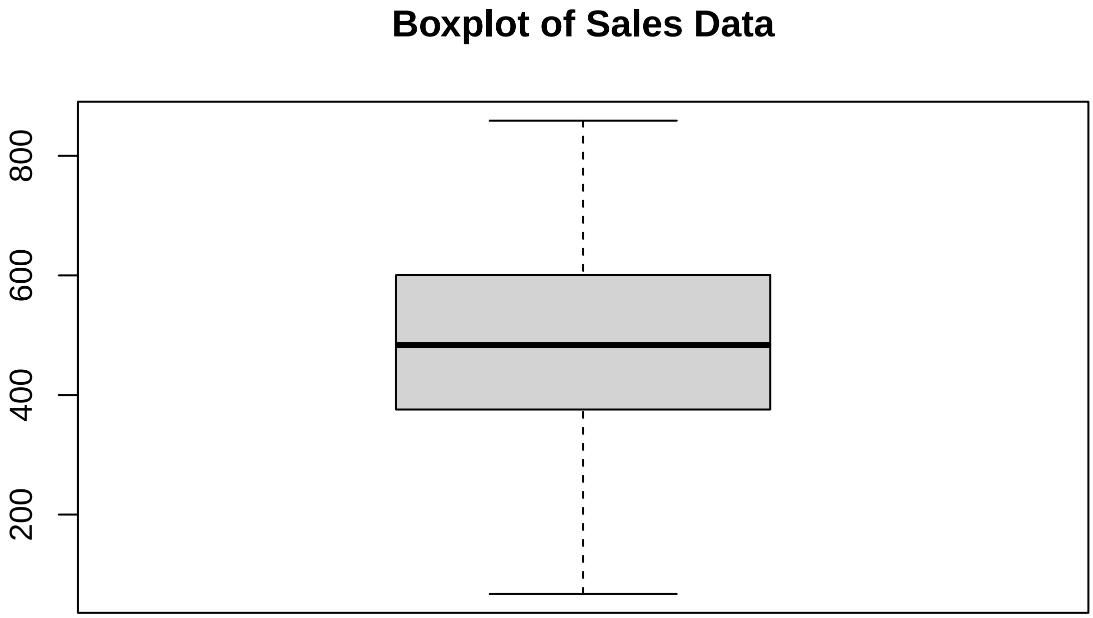
```r
ggplot(retail_df, aes(x = Inventory_Levels )) +
  geom_histogram(aes(y = ..density..),
                 bins = 30, fill = "steelblue",
                 alpha = 0.5) +
  geom_density(color = "red", size = 1) +
  ggtitle("Histogram and Density Plot of Inventory Levels") +
  theme_classic()
```



```
qqnorm(inv_retail_df,
       main = "Q-Q Plot of Sales")
qqline(inv_retail_df,
       col = "red")
```



```
boxplot(inv_retail_df,
        main = "Boxplot of Sales Data")
```



Initial analysis

- No NAs found in *Inventory_Levels*
- No values below 0 for the *Inventory_Levels* values.
- 488.55 $(Mean) > 483.72$ $(Median)$ suggests the data may be right skewed.
- Shapiro test had a $p-value = 0.4646$. This is below 0.5, suggesting it is not normalized, however it is relatively close to being normal.

- Q-Q plot and, Histogram and Density plot, show the data as near normal.

fitdistr Inventory Levels Since we are assuming the Inventory Levels across products follows a Lognormal distribution ($Y \sim Lognormal(\mu, \sigma^2)$), the parameter for *fitdistr()* we use is *lognormal*. Since we are explicitly looking for the sum of inventory levels across similar products, we will *sum* can consider finding the values for the individual *Product_ID*'s before evaluating the distribution.

```
retail_df %>%
  group_by(Inventory_Levels) %>%
  summarise(Count = n()) %>%
  filter(Count > 1)
```

```
## # A tibble: 0 x 2
## # i 2 variables: Inventory_Levels <dbl>, Count <int>
```

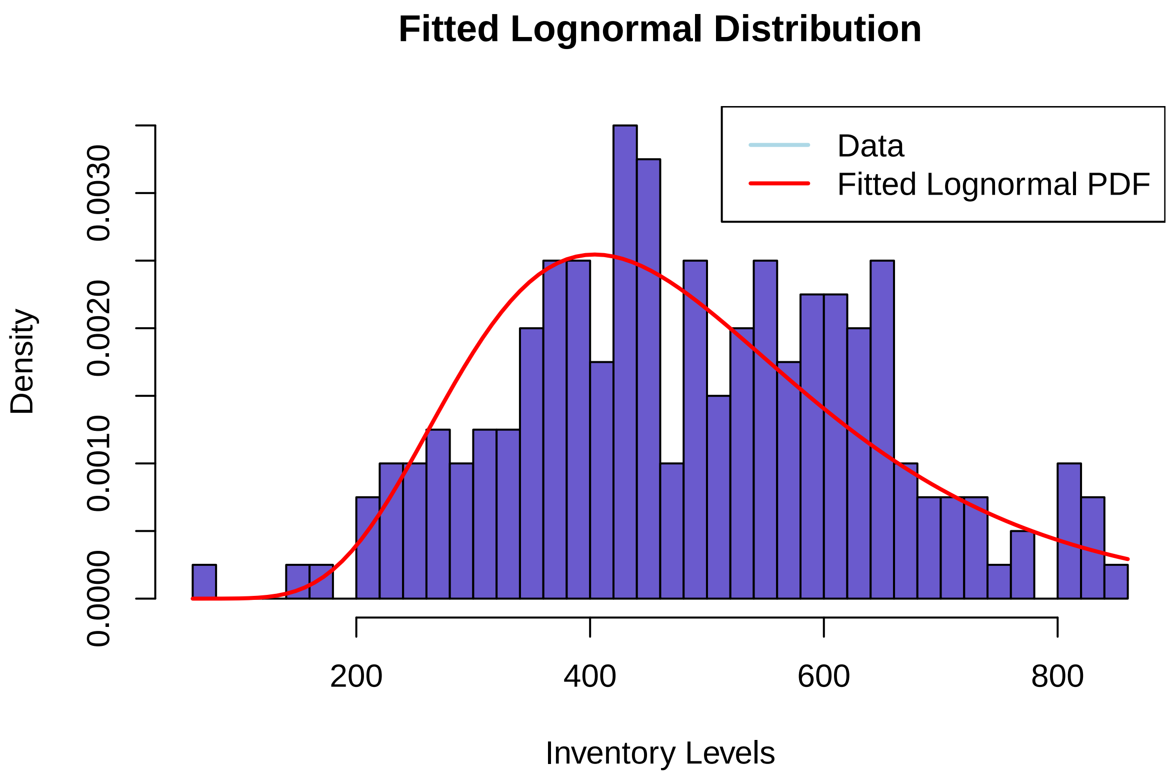
Since it appears that all *Product_ID*'s are unique I will refrain from using the sum.

The parameters of this distribution will ultimately be μ_{log} and σ_{log}

```
inv_lognormal <-
  fitdistr(inv_retail_df,"lognormal")

inv_log_mean <- inv_lognormal$estimate["meanlog"]
inv_log_var <- inv_lognormal$estimate["sdlog"]

print(inv_lognormal)
```

```
##     meanlog       sdlog
##   6.13303680   0.36332727
##  (0.02569112) (0.01816636)
```

```
# Histogram for Inventory Levels
hist(inv_retail_df,
     breaks = 30,
     probability = TRUE,
     main = "Fitted Lognormal Distribution",
     xlab = "Inventory Levels",
     col = "slateblue")

# Overlay the fitted lognormal curve
curve(dlnorm(x,
             meanlog = inv_log_mean,
             sdlog = inv_log_var),
      col = "red",
      lwd = 2,
      add = TRUE)

# Add legend
legend("topright",
       legend = c("Data", "Fitted Lognormal PDF"),
       col = c("lightblue", "red"),
       lwd = 2)
```



Mean and Variance (Y~Inventory Levels) *Theoretical Mean* or $E[x]$ of a random variable $X \sim Lognormal(\mu, \sigma^2)$ is equal to $e^{\mu+\frac{\sigma^2}{2}}$ and its theoretical variance is $(e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$

```
# Empirical statistics
inv_empirical_mean <- mean(retail_df$Inventory_Levels)
inv_empirical_variance <- var(retail_df$Inventory_Levels)

# Theoretical mean and variance for lognormal distribution
inv_theoretical_mean <-
  exp(inv_log_mean +
        (inv_log_var^2) / 2)

inv_theoretical_variance <-
  (exp(inv_log_var^2) - 1) *
  exp(2 * inv_log_mean + inv_log_var^2)

# # Print results
# cat("Empirical Mean:", empirical_mean, "\n")
# cat("Theoretical Mean:", theoretical_mean, "\n")
# cat("Empirical Variance:", empirical_variance, "\n")
# cat("Theoretical Variance:", theoretical_variance, "\n")

comparison_Inventory_Level <- data.frame(
  Metric = c("Mean", "Variance"),
  Empirical = c(inv_empirical_mean, inv_empirical_variance),
  Theoretical = c(inv_theoretical_mean, inv_theoretical_variance)
)
```

Answer The parameters for the Inventory Level distribution, $(X \sim Lognormal(\mu, \sigma^2))$ are $\mu_{log} = 6.13303680$ and $\sigma^2_{log} = 0.3633273$

Z ~ Lead Time **Z ~ Lead Time:** Assume that Lead_Time_Days follows a Normal distribution. Estimate the mean and standard deviation.

Calculate Empirical Expected Value and Variance:

```
ltd_retail_df <- retail_df$Lead_Time_Days
summary(ltd_retail_df)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.491   5.291   6.765   6.834   8.212  12.722
```

```
sum(ltd_retail_df<0)
```

```
## [1] 0
```

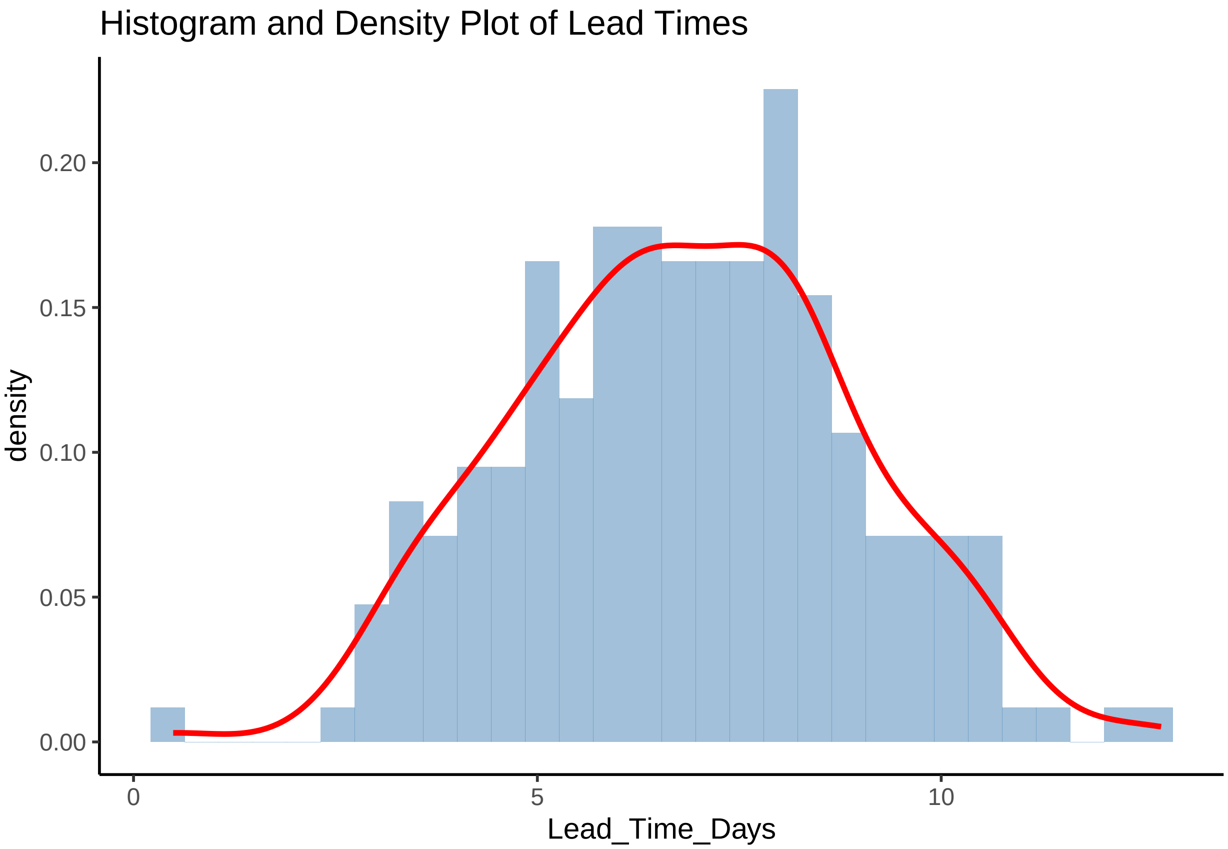
```
sum(is.na(ltd_retail_df))
```

```
## [1] 0
```

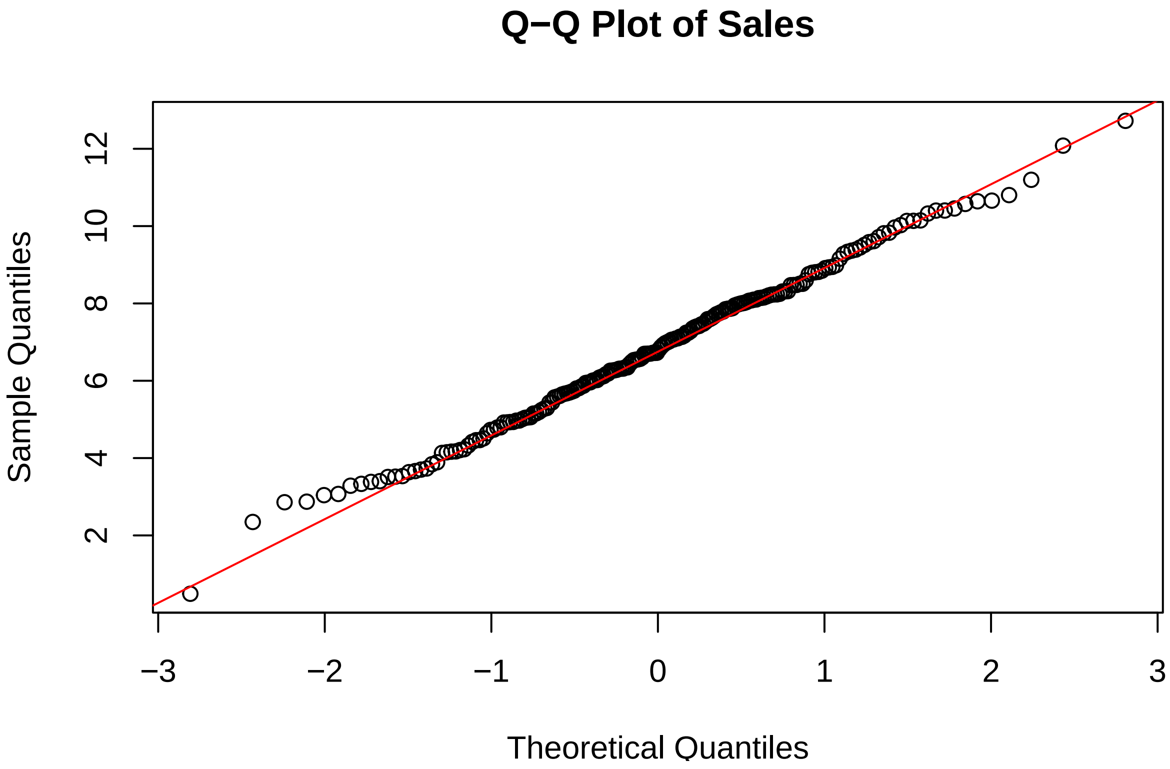
```
shapiro.test(ltd_retail_df)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ltd_retail_df
## W = 0.99618, p-value = 0.9026
```

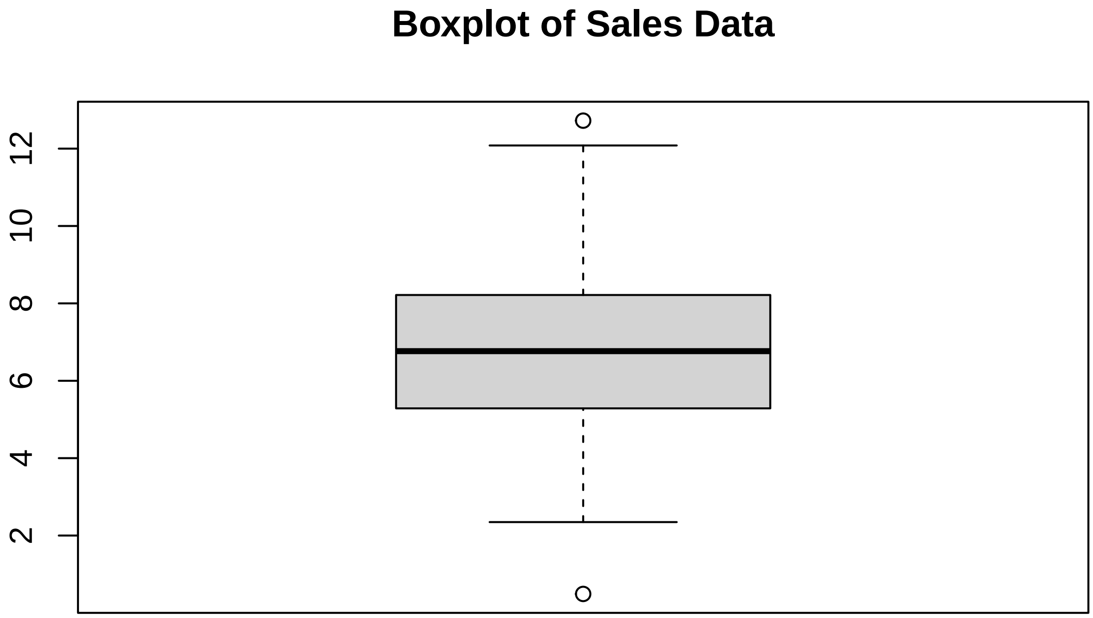
```
ggplot(retail_df, aes(x = Lead_Time_Days )) +
  geom_histogram(aes(y = ..density..),
                 bins = 30, fill = "steelblue",
                 alpha = 0.5) +
  geom_density(color = "red", size = 1) +
  ggtitle("Histogram and Density Plot of Lead Times") +
  theme_classic()
```



```
qqnorm(ltd_retail_df,
       main = "Q-Q Plot of Sales")
qqline(ltd_retail_df,
       col = "red")
```



```r
boxplot(ltd_retail_df,
        main = "Boxplot of Sales Data")
```



This data is basically normalized, with a slight right skewed and outliers that do not deviate much from the mean.

Mean and Variance Lead Time Solution We are using the same data with our theoretical Mean and Variance. Considering this, finding the mean and variance for our Lead Times should be the same. Another way of deriving our varian would be to square our standard deviation or σ. I've applied a few methods to show this.

1. Finding the standard deviation and squaring it
2. Using the *var()* function directly
3. Using the *fitdistr()* function to conclude our mean and standard are the same (alternatively I called on the *estimate* attribute for my *fitdistr* which is a redundant call to this value)

```
ltd_empirical_mean <- mean(retail_df$Lead_Time_Days)
sd(retail_df$Lead_Time_Days)
```

```
## [1] 2.088441
```

```
sd(retail_df$Lead_Time_Days)^2
```

```
## [1] 4.361587
```

```
var(retail_df$Lead_Time_Days)
```

```
## [1] 4.361587
```

```
ltd_fitdistr<-fitdistr(ltd_retail_df,"normal")
print(ltd_fitdistr)
```

```
##      mean          sd
##   6.8342981   2.0832137
##  (0.1473055) (0.1041607)
```

```
ltd_fitdistr$estimate
```

```
##     mean       sd
## 6.834298 2.083214
```

Answer The estimated mean is 6.834298 and standard deviation is 2.083214 Calculated Empirical Expected Value is 6.834298 and Variance is 4.361587

2.

Calculate the empirical mean and variance for all three variables. Compare these empirical values with the theoretical values derived from the estimated distribution parameters.

Answer

Our Empirical and Theoretical, Variance and Mean are as follow:

For **X~Sales**

```
##     Metric   Empirical Theoretical
## 1     Mean    636.9162    636.9155
## 2 Variance 214831.7509 221073.1799
```

For **Y~Inventory**

```
print(comparison_Inventory_Level)
```

```
##           Metric  Empirical Theoretical
## meanlog     Mean   488.5472    492.2763
## sdlog   Variance 24039.4464  34197.4741
```

For **Z~Lead Times** the Empirical and Theoretical Mean is 6.834298 and the Empirical and Theoretical Variance is 4.361587

Part 2:

Probability Analysis and Independence Testing (5 Points)

Task:

1.

Empirical Probabilities: For the Lead_Time_Days variable (assumed to be normally distributed), calculate the following empirical probabilities:

- $P(Z > \mu | Z > \mu - \sigma)$
- $P(Z > \mu + \sigma | Z > \mu)$
- $P(Z > \mu + 2\sigma | Z > \mu)$

Notes

- We assume the **standard normal distribution in context to these probabilities, because they do not specify a random variable X with its own mean μ and standard deviation σ
- The standard normal distribution also known as the z distribution, can have the notation $N(\mu, \sigma)$ where N signifies the distribution is normal, while μ, σ and σ^2 retains its known definition as the mean, standard deviation and variance of the distribution. Please note **Reference** *vii* for a more in depth explanation.

i.

- $P(Z > \mu | Z > \mu - \sigma) = \frac{P(Z>\mu \ \cap \ Z>\mu-\sigma)}{Z>\mu-\sigma}$
- Our numerator can be simplified to $P(Z > \mu)$. Stated plainly, $P(Z > \mu - \sigma)$ is encompassed with $P(Z > \mu)$, but both conditions cannot be met if we only satisfy $P(Z > \mu - \sigma)$ therefore we only need to satisfy the second condition $P(Z > \mu)$.
- This simplifies our conditional probability $P(Z > \mu | Z > \mu - \sigma) = \frac{P(Z>\mu \ \cap \ Z>\mu-\sigma)}{Z>\mu-\sigma}$ to $\frac{Z>\mu}{Z>\mu-\sigma}$
- $N(\mu, \sigma)$ for the standard normal distribution is $Z' \sim N(0, 1)$, since the mean of a standard normal distribution is $\mu = 0$ with a standard deviation of $\sigma = 1$ (Reference *vii*).
- $P(Z > \mu)$ represents 50% of the probability as it is the mean $\therefore P(Z > \mu) = 0.5$
- We can use the standardization formula $F_X(x) = P(X \leq x) = F_z(\frac{x-\mu}{\sigma})$ or just $Z = \frac{x-\mu}{\sigma}$ to substitute based on our second conditions probability (Grinstead and Snell's Introduction to Probability pg. 214):
  - random variable $X = \mu - \sigma$
  - $X \sim N(\mu, \sigma^2) \rightarrow Z = \frac{(\mu-\sigma)-\mu}{\sigma} = -\frac{\sigma}{\sigma} = -1$
  - This transforms our formula from $Z > \mu - \sigma$ to $Z > -1$

we can find use *pnorm()* to get the value for $Z = -1$

```
pnorm(-1)
```

```
## [1] 0.1586553
```

Since $P(Z > -1) = 1 - P(Z \leq -1)$ we can solve by subtracting these values.

```
1-pnorm(-1)
```

```
## [1] 0.8413447
```

Again we are trying to solve for $\frac{Z>\mu}{Z>\mu-\sigma}$ and by substituting we get $\frac{0.5}{0.8413447}$ which is

Conclusion

```
0.5/(1-pnorm(-1))
```

```
## [1] 0.5942867
```

ii.

- For $P(Z > \mu + \sigma | Z > \mu)$ we can use a similar conditional probability as above ($P(A|B) = \frac{P(A \cap B)}{P(B)}$) to get $\frac{P((Z>\mu+\sigma)\ \cap\ (Z>\mu))}{P(Z>\mu)}$
- In this probability the $P(Z > \mu + \sigma)$ the condition encompasses $P(Z > \mu)$, since for $P((Z > \mu + \sigma) \cap (Z > \mu)$ to be true, $P(Z > \mu + \sigma)$ would be required.
- Therefore $\frac{P((Z>\mu+\sigma)\ \cap\ (Z>\mu))}{P(Z>\mu)}$ can be written as $\frac{P(Z>\mu+\sigma)}{P(Z>\mu)}$
- $P(Z > \mu) = 0.5$ as per our last probabilities conclusions.
- Again we can use the standardization formula $F_X(x) = P(X \leq x) = F_z(\frac{x-\mu}{\sigma})$ or just $Z = \frac{x-\mu}{\sigma}$ to substitute based on our second conditions probability (Grinstead and Snell's Introduction to Probability pg. 214):
- random variable $X = \mu + \sigma$
  - $X \sim N(\mu, \sigma^2) \rightarrow Z = \frac{(\mu+\sigma)-\mu}{\sigma} = \frac{\sigma}{\sigma} = 1$
  - This transforms our formula from $Z > \mu + \sigma$ to $Z > 1$

Again we solve for $Z = 1$ this using *pnorm()* and we understand $P(Z > 1) = 1 - P(Z < 1)$ because we are computing for the tail of this probability

```
1-pnorm(1)
```

```
## [1] 0.1586553
```

Which we will use substitute and solve for $\frac{P(Z>\mu+\sigma)}{P(Z>\mu)}$

Conclusion

```
(1-pnorm(1))/0.5
```

```
## [1] 0.3173105
```

iii.

$P(Z > \mu + 2\sigma | Z > \mu) = \frac{P(Z>\mu+2\sigma\ \cap\ Z>\mu)}{Z>\mu}$ where $P(Z > \mu + 2\sigma\ \cap\ Z > \mu)$ implies $Z > \mu + 2\sigma$ for our numerator and $Z > \mu = 0.5$ based on previous work. - Substituting for the standardization formula we get $X \sim N(\mu, \sigma^2) \rightarrow Z = \frac{(\mu+2\sigma)-\mu}{\sigma} = \frac{2\sigma}{\sigma} = 2$ - $P(Z > 2) = 1 - P(Z < 2) = \frac{1-P(Z<2)}{0.5}$ which is

Conclusion

```
(1-pnorm(2))/0.5
```

```
## [1] 0.04550026
```

Answers

- $P(Z > \mu | Z > \mu - \sigma) \approx 0.594$
- $P(Z > \mu + \sigma | Z > \mu) \approx 0.317$
- $P(Z > \mu + 2\sigma | Z > \mu) \approx 0.0455$

2.

Correlation and Independence:

- Investigate the correlation between Sales and Price. Create a contingency table using quartiles of Sales and Price, and then evaluate the marginal and joint probabilities.
- Use Fisher's Exact Test and the Chi-Square Test to check for independence between Sales and Price. Discuss which test is most appropriate and why.

Contingency, Joint and Marginal Table

Contingency

```
sales_qrtl <- cut(retail_df$Sales,
                        breaks = quantile(retail_df$Sales,
                                          probs = seq(0, 1, 0.25)),
                        include.lowest = TRUE,
                        labels = c("Q1", "Q2", "Q3", "Q4"))


price_qrtl <- cut(retail_df$Price,
                        breaks = quantile(retail_df$Price,
                                          probs = seq(0, 1, 0.25)),
                        include.lowest = TRUE,
                        labels = c("Q1", "Q2", "Q3", "Q4"))
cont_tbl <- table(sales_qrtl, price_qrtl)

cont_tbl %>%
    kable()
```

	Q1	Q2	Q3	Q4
Q1	11	16	12	11
Q2	13	10	15	12
Q3	15	10	13	12
Q4	11	14	10	15

Joint

```
jnt_tbl<-prop.table(cont_tbl)

jnt_tbl %>%
  kable() %>%
  kable_paper()
```

	Q1	Q2	Q3	Q4
Q1	0.055	0.08	0.060	0.055
Q2	0.065	0.05	0.075	0.060
Q3	0.075	0.05	0.065	0.060
Q4	0.055	0.07	0.050	0.075

Marginal

```
# Marginal probabilities for Sales
marginal_sales <- margin.table(jnt_tbl, 1)

# Marginal probabilities for Price
marginal_price <- margin.table(jnt_tbl, 2)

marginal_sales%>%
  kable()%>%
  kable_classic_2()
```

sales_qrtl	Freq
Q1	0.25
Q2	0.25
Q3	0.25
Q4	0.25

```r
marginal_price %>%
  kable()%>%
  kable_minimal()
```

price_qrtl	Freq
Q1	0.25
Q2	0.25
Q3	0.25
Q4	0.25

The results seem balanced. Equal likelihood of the values for *Sales* and *Price* to end up in any quartile. So not much of a takeaway.

Fisher and Chi Square

Use Fisher's Exact Test and the Chi-Square Test to check for independence between Sales and Price. Discuss which test is most appropriate and why.

Fisher Exact Test

```r
fisher.test(cont_tbl, workspace = 2e7)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  cont_tbl
## p-value = 0.8637
## alternative hypothesis: two.sided
```

```r
chisq.test(cont_tbl)
```

```
##
##  Pearson's Chi-squared test
##
## data:  cont_tbl
## X-squared = 4.8, df = 9, p-value = 0.8514
```

If we can identify if the data is sparse we can decide which test to use. Sparse meaning the frequency output is all ≥ 5. We can also base it off of if it theres a lot of data, but checking the frequency output is a more practical approach.

```r
chisq.test(cont_tbl)$expected
```

```
##           price_qrtl
## sales_qrtl   Q1   Q2   Q3   Q4
##         Q1 12.5 12.5 12.5 12.5
##         Q2 12.5 12.5 12.5 12.5
##         Q3 12.5 12.5 12.5 12.5
##         Q4 12.5 12.5 12.5 12.5
```

chi test seems most appropriate. regardless the $p-value$ here looks very similar. and since both are > 0.05 there is not enough evidence to support a significant relationship.

Problem 2

Advanced Forecasting and Optimization (Calculus) in Retail

Context: You are working for a large retail chain that wants to optimize pricing, inventory management, and sales forecasting using data-driven strategies. Your task is to use regression, statistical modeling, and calculus-based methods to make informed decisions.

Part 1

Descriptive and Inferential Statistics for Inventory Data (5 Points)

Task:

1.

Inventory Data Analysis:

- Generate univariate descriptive statistics for the Inventory_Levels and Sales variables.
- Create appropriate visualizations such as histograms and scatterplots for Inventory_Levels, Sales, and Price.
- Compute a correlation matrix for Sales, Price, and Inventory_Levels.
- Test the hypotheses that the correlations between the variables are zero and provide a 95% confidence interval.

Univariate Descriptive Statistics

Generate univariate descriptive statistics for the Inventory_Levels and Sales variables.

I surprisingly did this for the first part, but will repeat this for completion sake

```
summary(sales_retail_df)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   25.57  284.42  533.54  636.92  867.58 2447.49
```

```
sum(sales_retail_df < 0)
```

```
## [1] 0
```

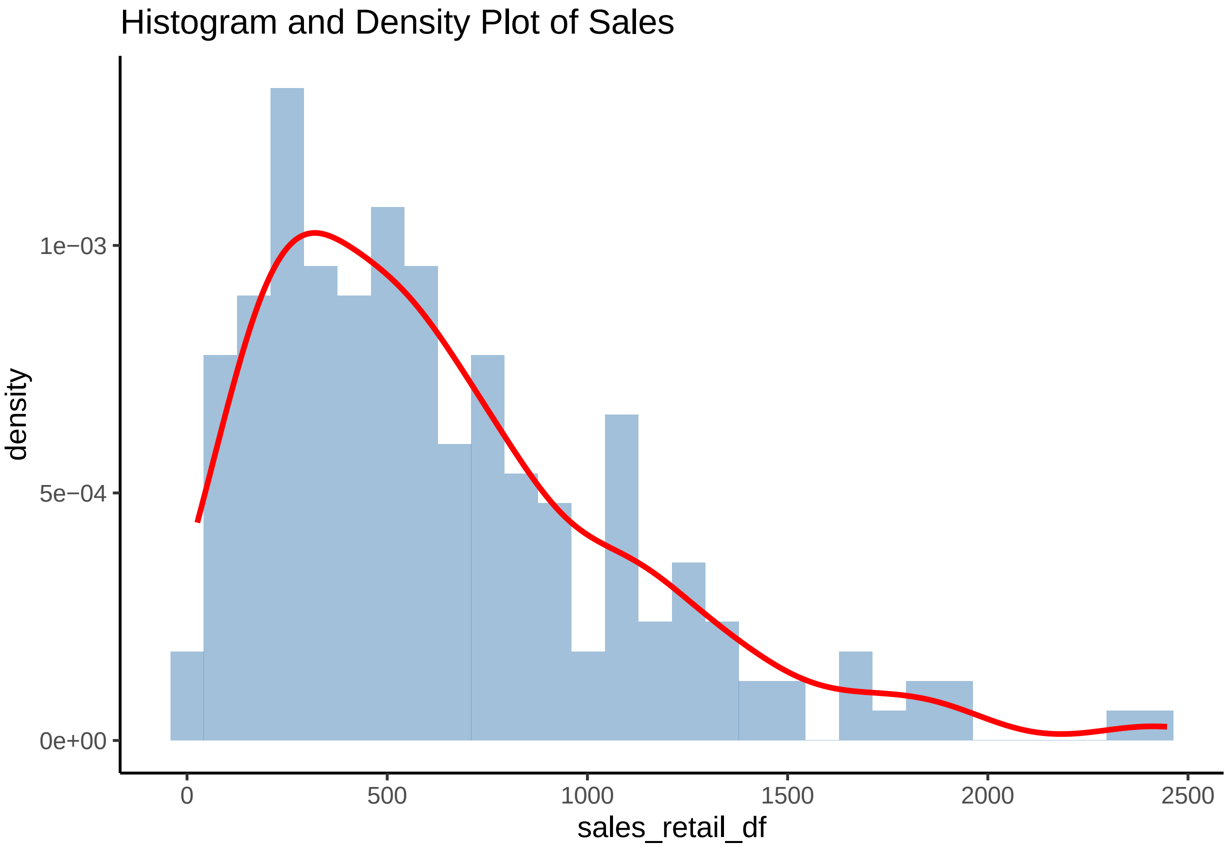
```
sum(is.na(sales_retail_df))
```

```
## [1] 0
```

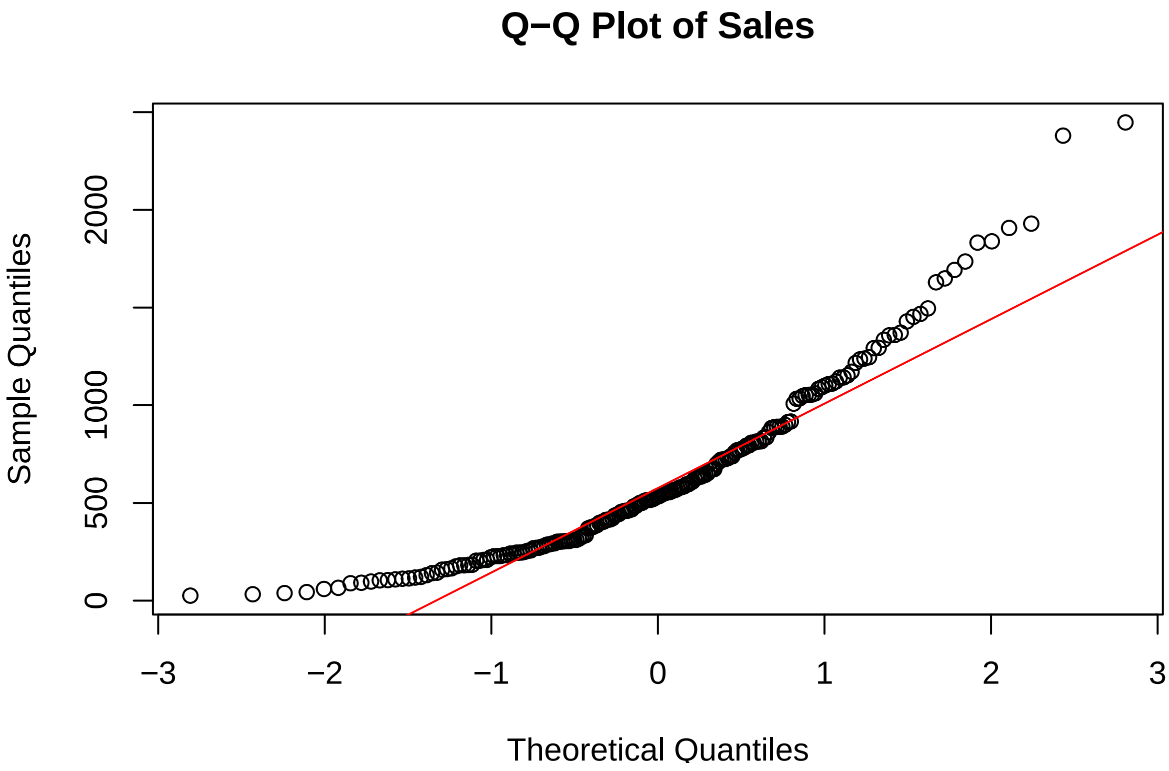
```
shapiro.test(sales_retail_df)
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  sales_retail_df
## W = 0.90377, p-value = 4.397e-10
```

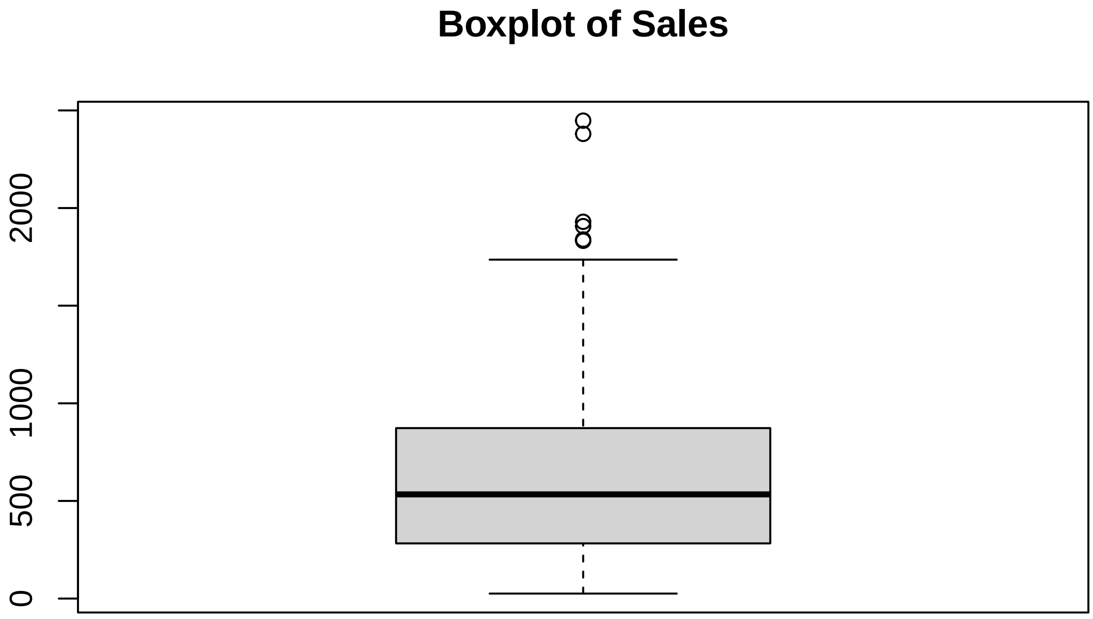
```
ggplot(retail_df, aes(x = sales_retail_df)) +
  geom_histogram(aes(y = ..density..),
                 bins = 30, fill = "steelblue",
                 alpha = 0.5) +
  geom_density(color = "red", size = 1) +
  ggtitle("Histogram and Density Plot of Sales") +
  theme_classic()
```



```
qqnorm(sales_retail_df, main = "Q-Q Plot of Sales")
qqline(sales_retail_df, col = "red")
```



```r
boxplot(sales_retail_df, main = "Boxplot of Sales")
```



```r
summary(inv_retail_df)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   67.35  376.51  483.72  488.55  600.42  858.79
```

```r
sum(inv_retail_df < 0)
```

```
## [1] 0
```

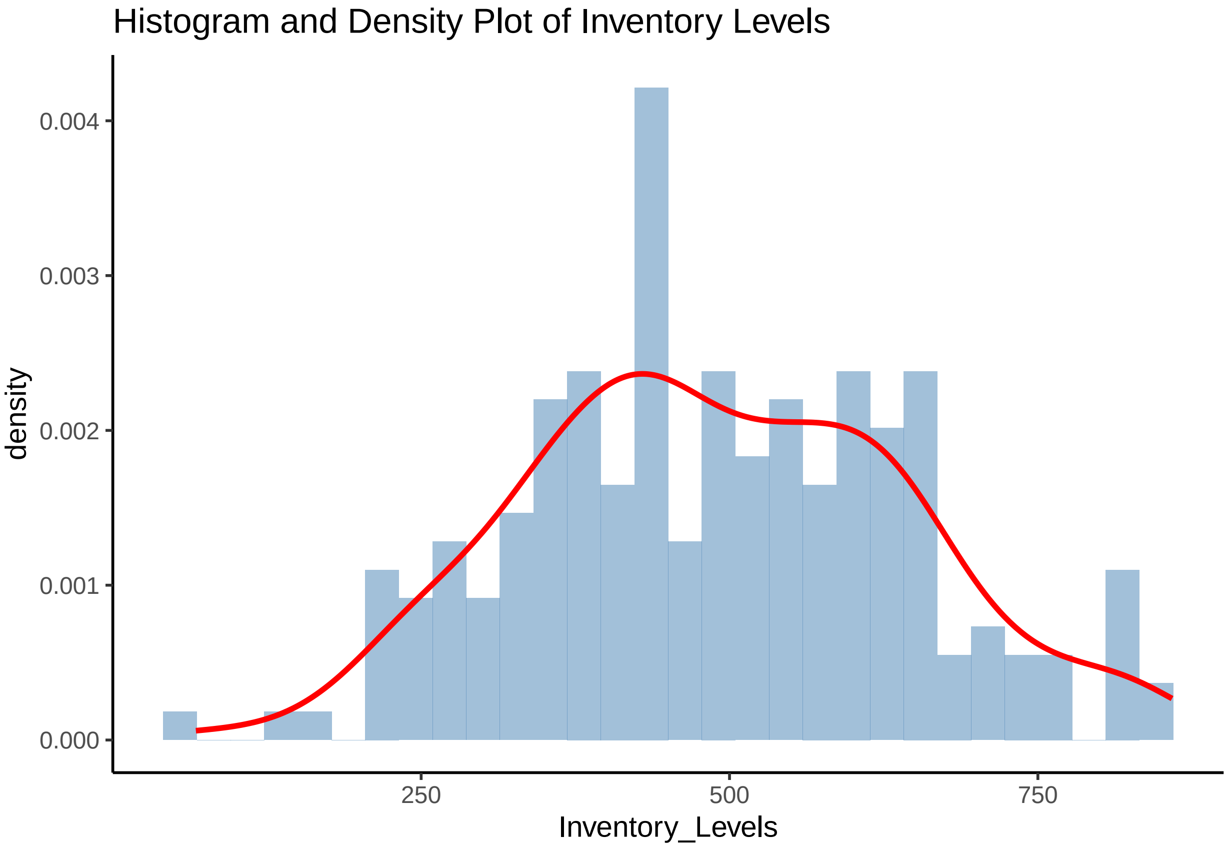
```
sum(is.na(inv_retail_df))
```

```
## [1] 0
```

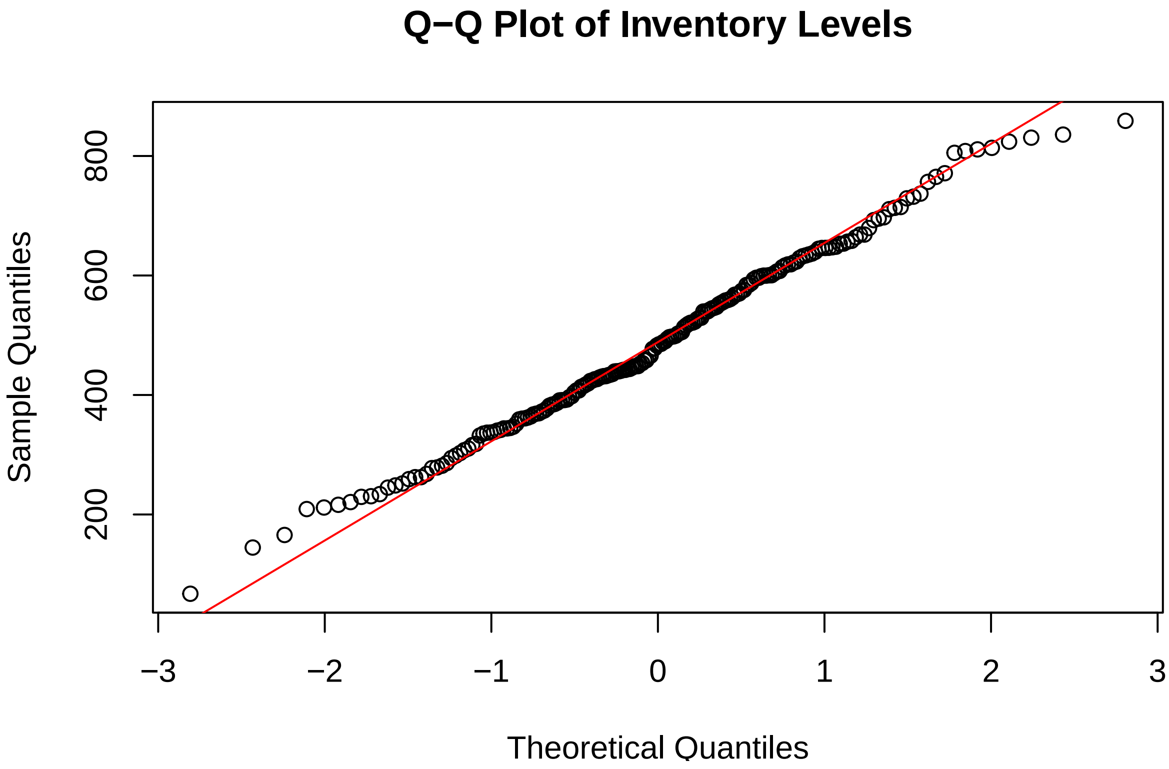
```
shapiro.test(inv_retail_df)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  inv_retail_df
## W = 0.99303, p-value = 0.4646
```

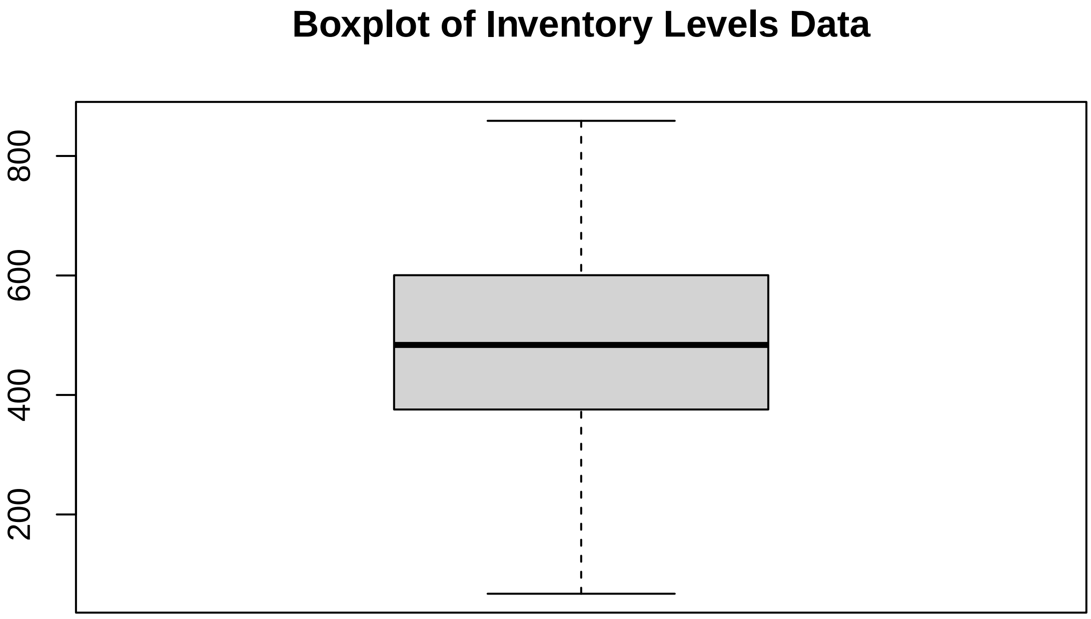
```
ggplot(retail_df, aes(x = Inventory_Levels)) +
  geom_histogram(aes(y = ..density..),
                 bins = 30, fill = "steelblue",
                 alpha = 0.5) +
  geom_density(color = "red", size = 1) +
  ggtitle("Histogram and Density Plot of Inventory Levels") +
  theme_classic()
```



```
qqnorm(inv_retail_df, main = "Q-Q Plot of Inventory Levels")
qqline(inv_retail_df, col = "red")
```



```r
boxplot(inv_retail_df, main = "Boxplot of Inventory Levels Data")
```



Repeating the conclusions from my initial analysis

Sales Analysis

- For our *Sales* data our $Mean > Median$ (636.92 > 533.54) which indicates that our data is right skewed and not normalized. This is supported by our Histogram, our Q-Q plot and the Shapiro test's $p - value$ of less than 0.05.
- No NAs are noted with the *Sales* data
- Our range for the values within *Sales* is 25.57 to 2447.49, encompassing a wide range.

- Our Box plot indicates that there are outliers, primarily for values > 1000

Inventory Levels Analysis

- No NAs found in *Inventory_Levels*
- No values below 0 for the *Inventory_Levels* values.
- 488.55 $(Mean) > 483.72$ $(Median)$ suggests the data may be right skewed.
- Shapiro test had a $p - value = 0.4646$. This is below 0.5, suggesting it is not normalized, however it is relatively close to being normal.
- Q-Q plot and, Histogram and Density plot, show the data as near normal.

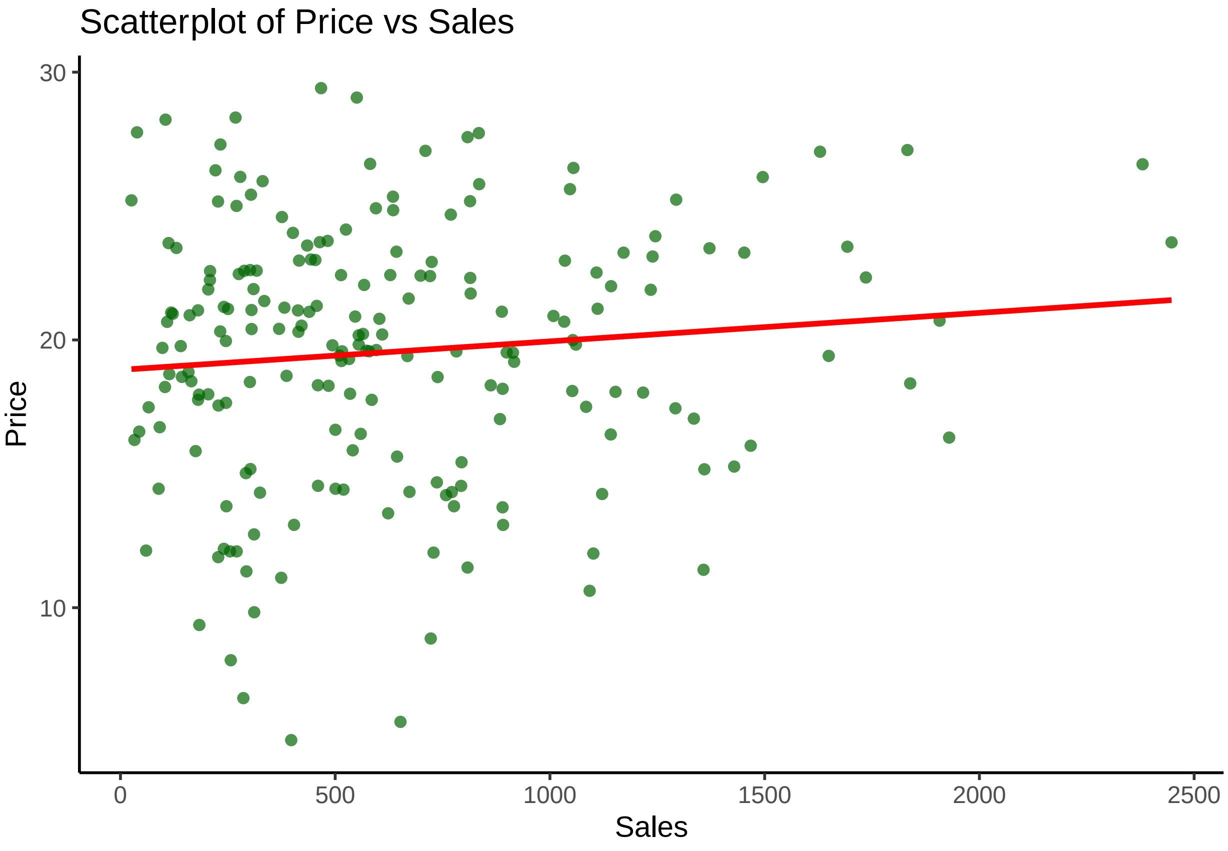
Scatterplots and Price

Create appropriate visualizations such as histograms and scatterplots for Inventory_Levels, Sales, and Price.

Considering I already generated a histogram for *Inventory_Levels* and *Sales* I will only create a histogram for *Price* to visualize the relationships

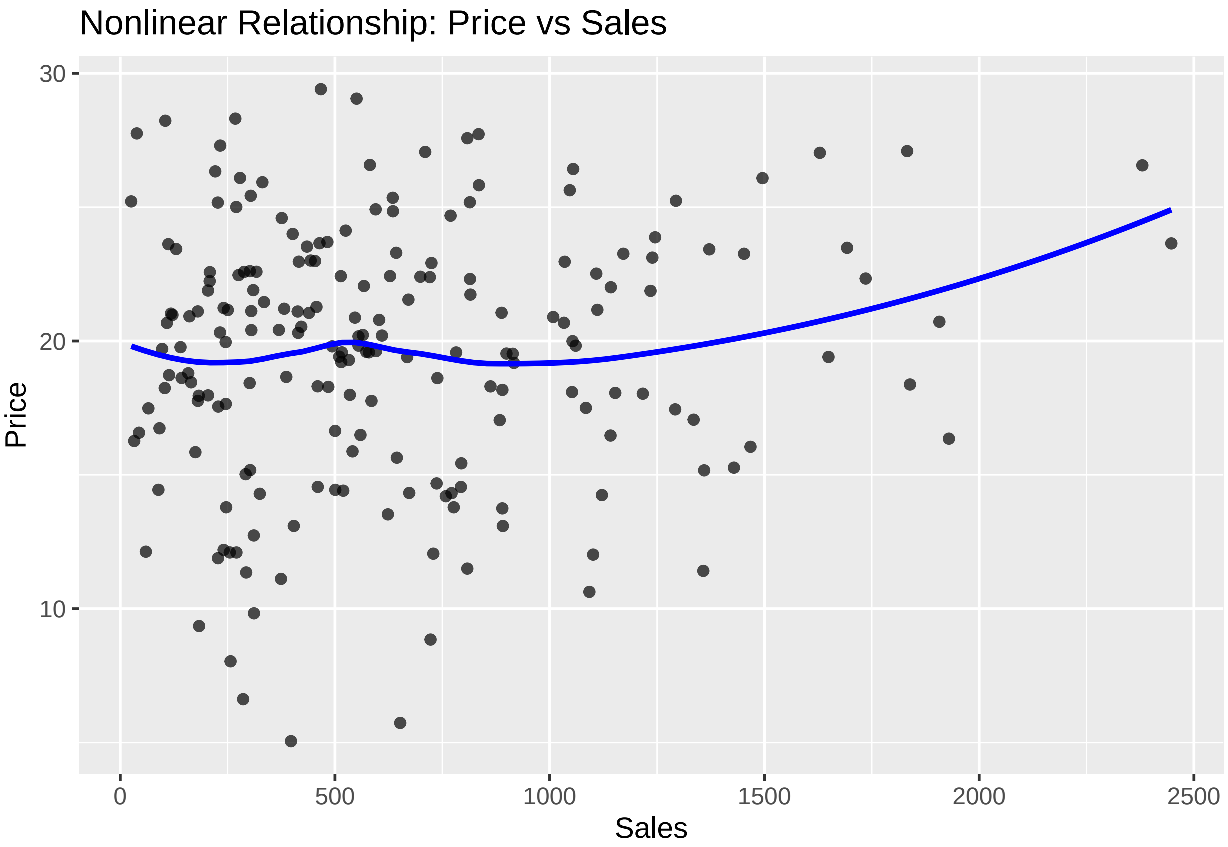
```r
# Scatter plot: Price vs Sales
ggplot(retail_df, aes(x = Sales, y = Price)) +
  geom_point(alpha = 0.7, color = "darkgreen") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  ggtitle("Scatterplot of Price vs Sales") +
  xlab("Sales") +
  ylab("Price") +
  theme_classic()
```

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



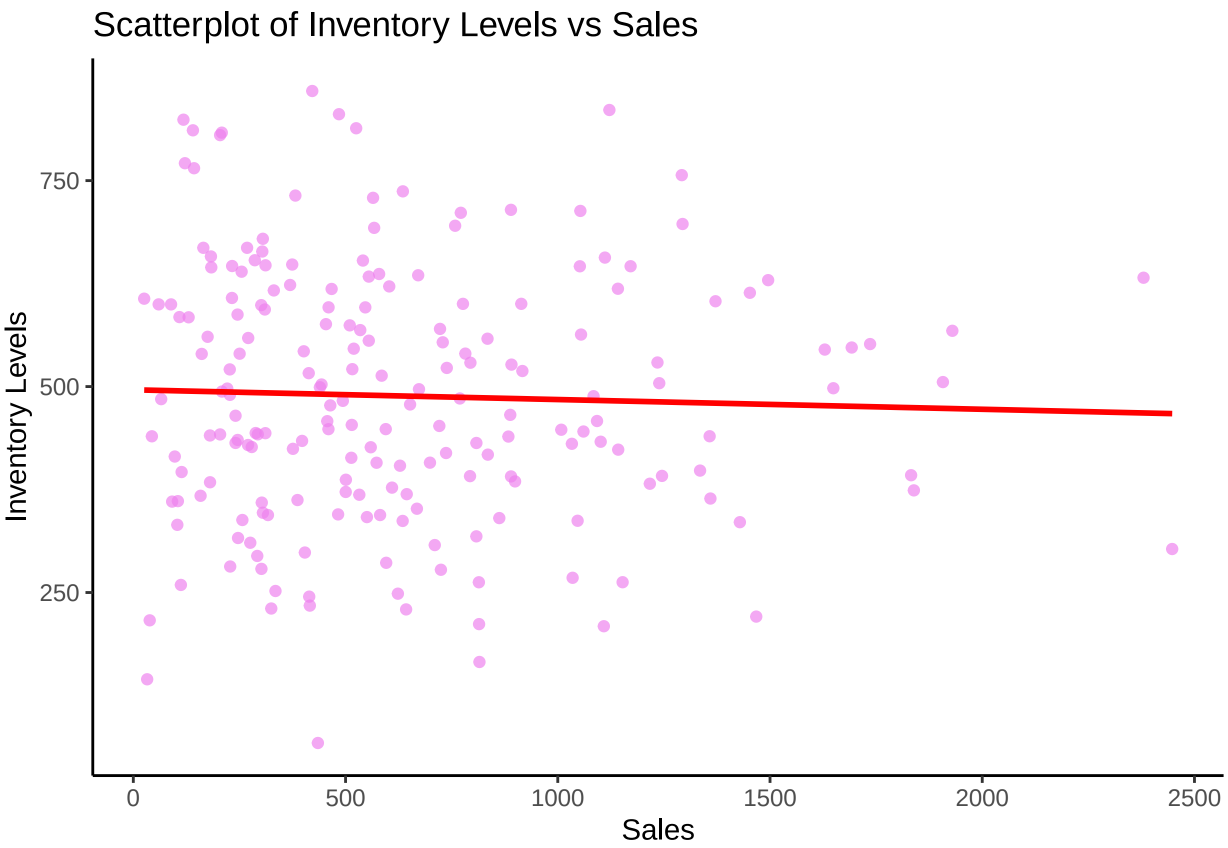
```r
ggplot(retail_df, aes(x = Sales, y = Price)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  ggtitle("Nonlinear Relationship: Price vs Sales") +
  theme_gray()
```

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



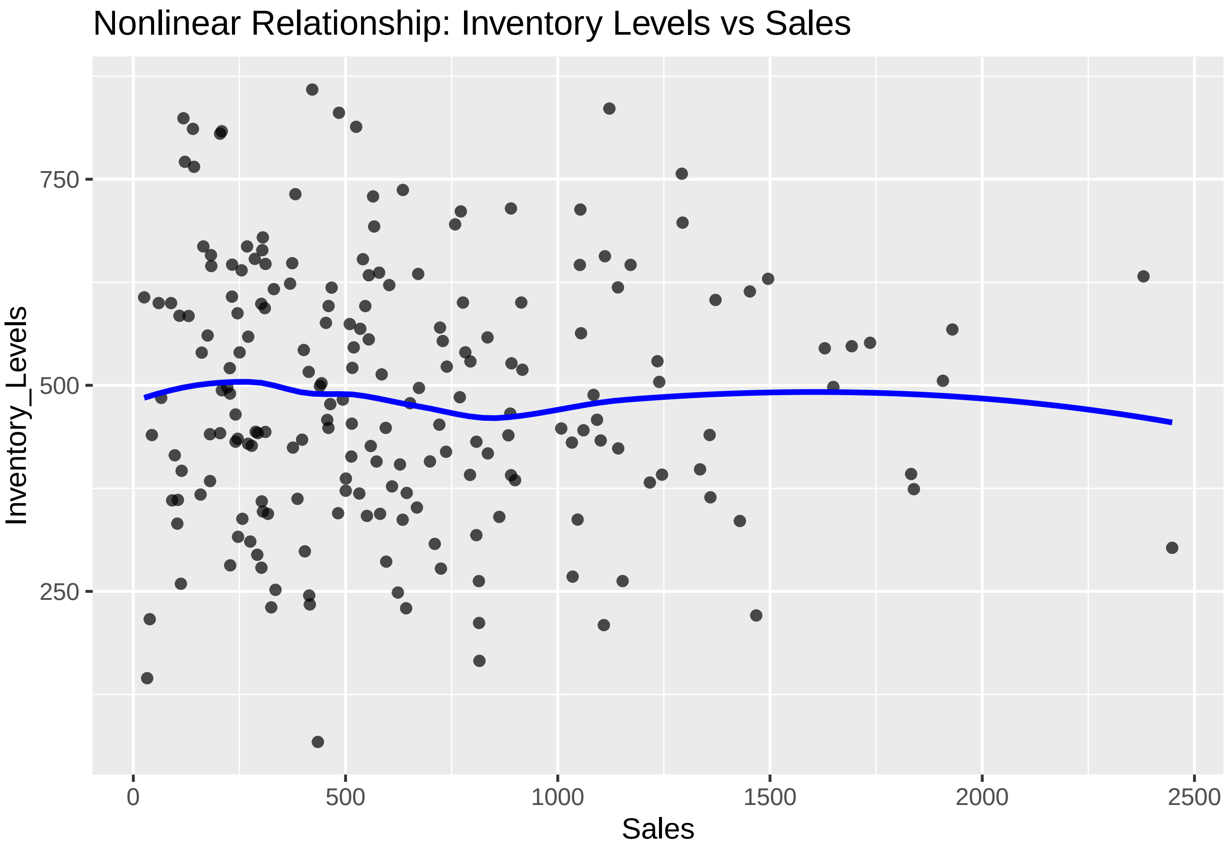
```r
# Scatter plot: Inventory Levels vs Sales
ggplot(retail_df, aes(x = Sales, y = Inventory_Levels)) +
  geom_point(alpha = 0.7, color = "violet") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  ggtitle("Scatterplot of Inventory Levels vs Sales") +
  xlab("Sales") +
  ylab("Inventory Levels") +
  theme_classic()
```

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



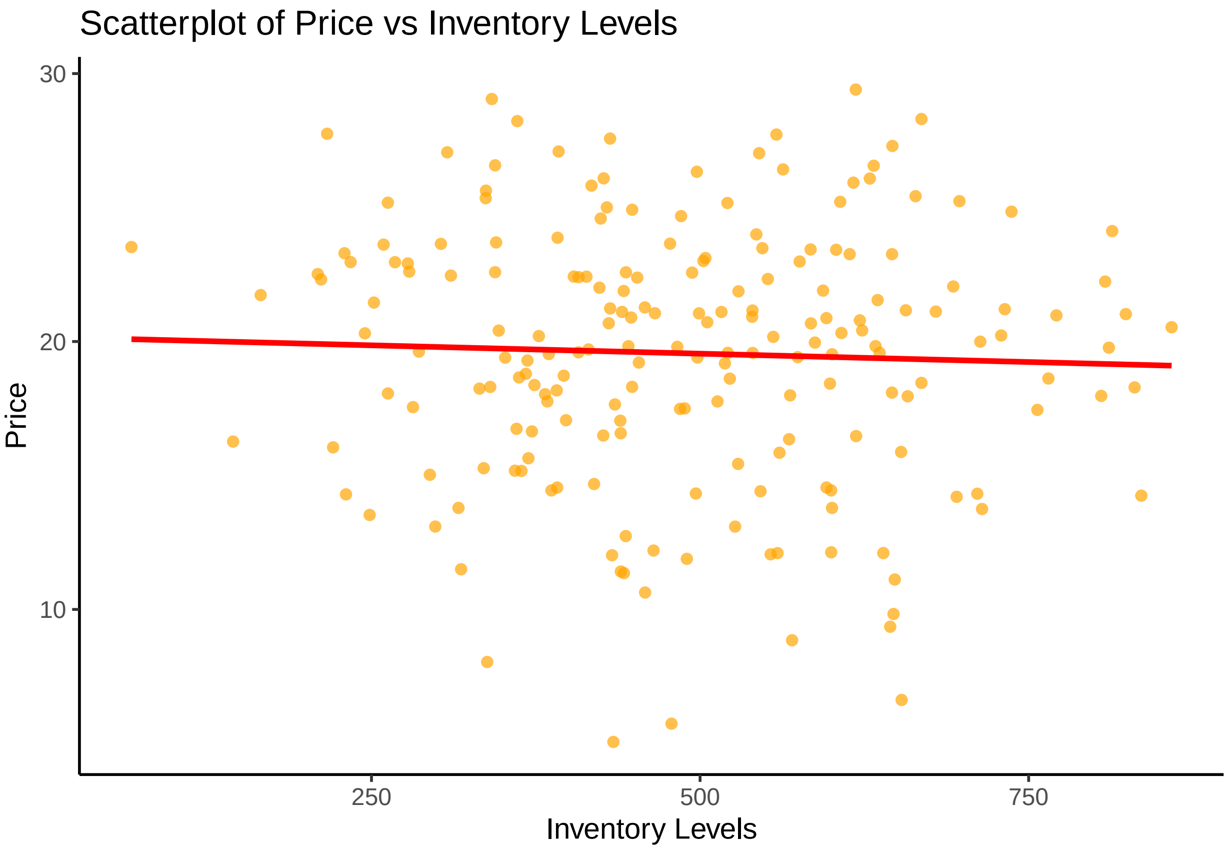
```r
ggplot(retail_df, aes(x = Sales, y = Inventory_Levels)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  ggtitle("Nonlinear Relationship: Inventory Levels vs Sales") +
  theme_gray()
```

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



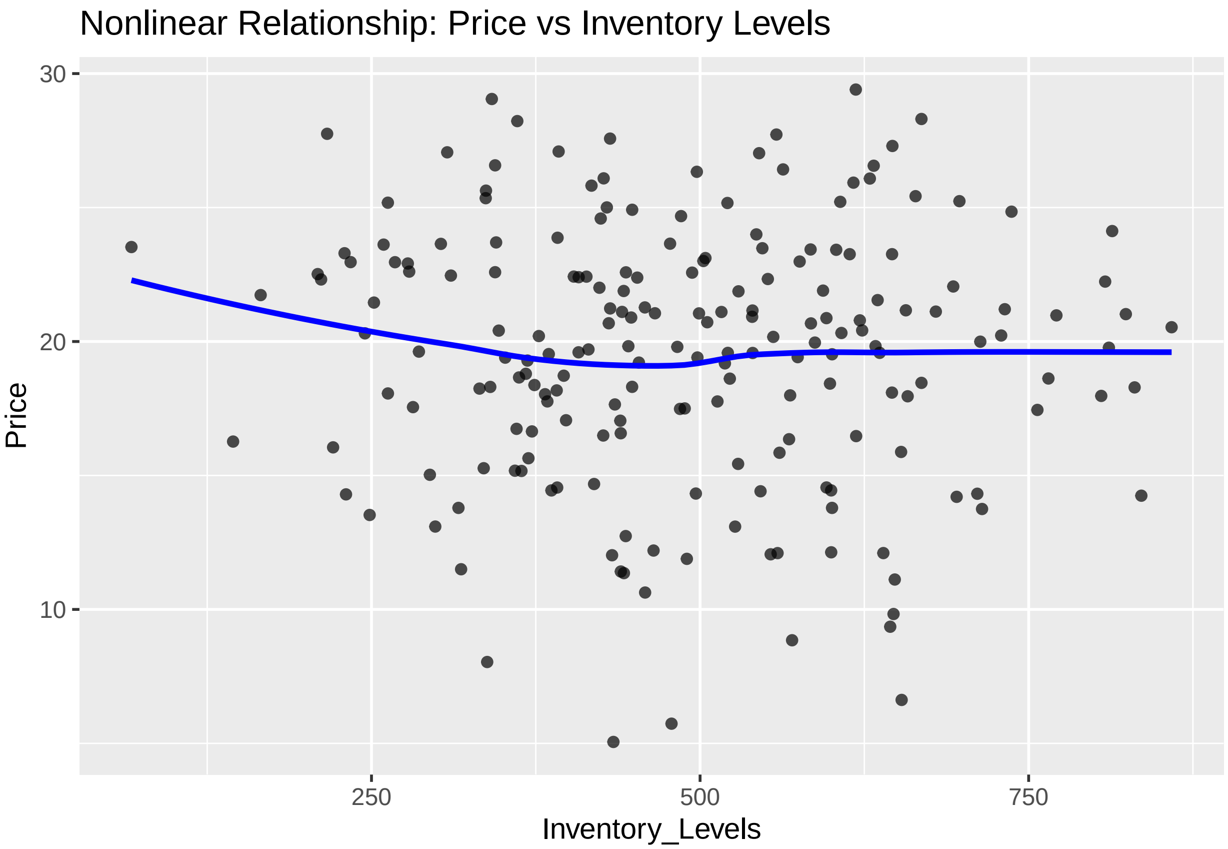
```r
# Scatter plot: Price vs Inventory Levels
ggplot(retail_df, aes(x = Inventory_Levels, y = Price)) +
  geom_point(alpha = 0.7, color = "orange") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  ggtitle("Scatterplot of Price vs Inventory Levels") +
  xlab("Inventory Levels") +
  ylab("Price") +
  theme_classic()
```

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

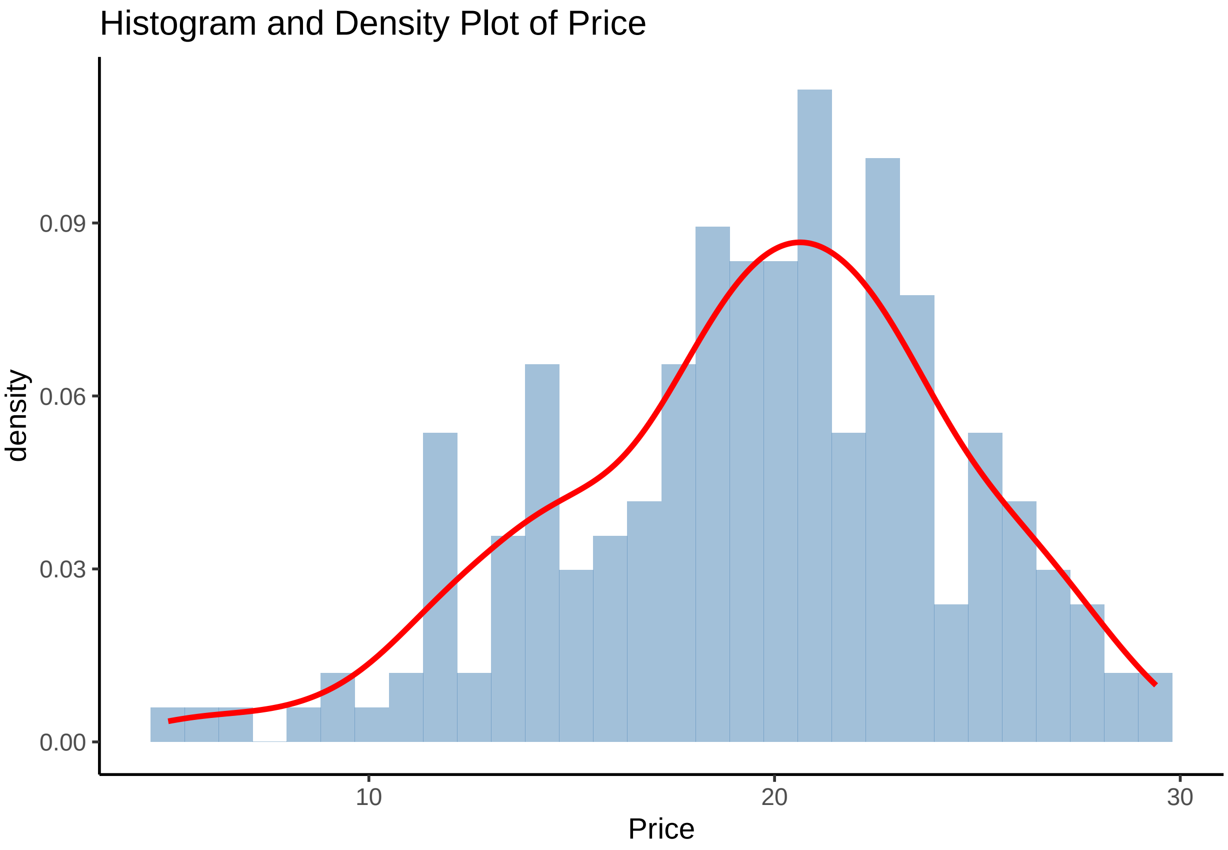


```
ggplot(retail_df, aes(x = Inventory_Levels, y = Price)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  ggtitle("Nonlinear Relationship: Price vs Inventory Levels") +
  theme_gray()
```

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



```r
ggplot(retail_df, aes(x = Price)) +
  geom_histogram(aes(y = ..density..),
                 bins = 30, fill = "steelblue",
                 alpha = 0.5) +
  geom_density(color = "red", size = 1) +
  ggtitle("Histogram and Density Plot of Price") +
  theme_classic()
```



Histogram and Density Plot of Price

- Scatter plots of There does not seem to be a strong relationship between these variables, as the Price hardly changes based on Inventory_Level or Sales, and the same can be said regarding Sales' impact on inventory.
- Regardless there is a trend which is somewhat expected:
  - As Sales increases so does price, and Inventory Levels slightly diminishes.
  - As Inventory Levels increase there is a drop in price
- This follows for the most part the understood relationship of supply and demand.

Price Histogram

The data here is clearly left skewed, and interpreting this for a business took some reading. My understanding is, because it still is somewhat normalized, the $20 mark (our mode) is likely a preferred price of this businesses customers. The skew is representative of either a lack of demand or lack of supply of cheaper items.

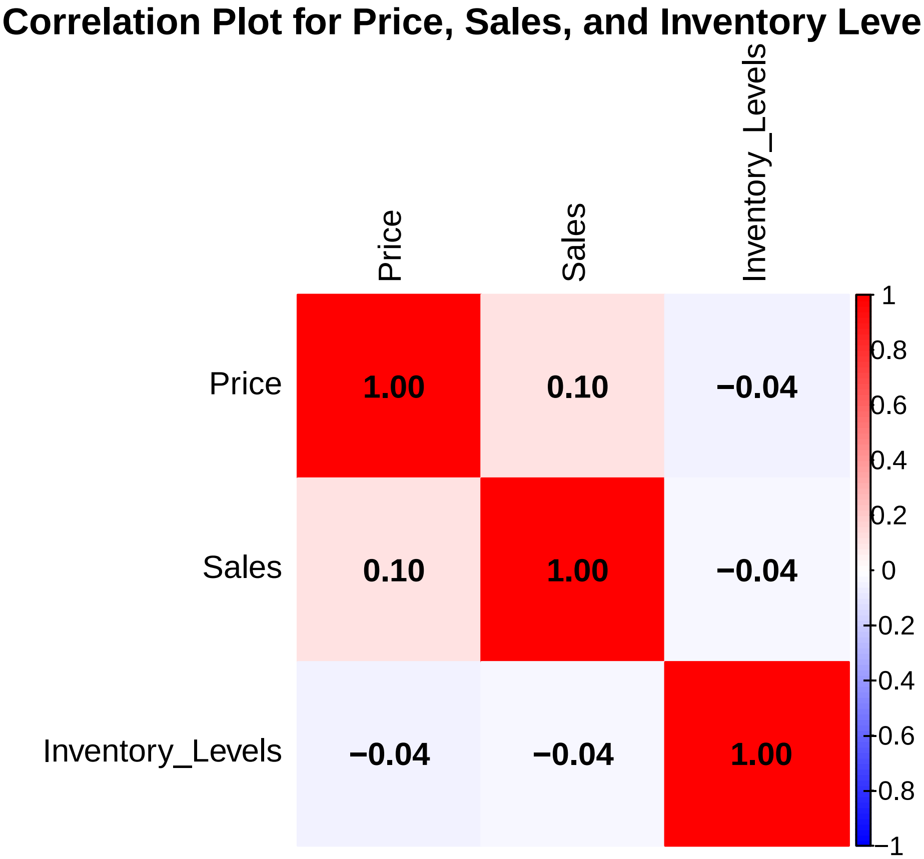
Correlation Plot

Compute a correlation matrix for Sales, Price, and Inventory_Levels.

```r
cor_matrix_retail <-
  cor(retail_df[,c("Price",
                   "Sales",
                   "Inventory_Levels")],
      use = "complete.obs")

print(cor_matrix_retail)
```

```
##                        Price      Sales Inventory_Levels
## Price             1.00000000  0.10272730      -0.04025941
## Sales             0.10272730  1.00000000      -0.03529619
## Inventory_Levels -0.04025941 -0.03529619       1.00000000
```

```r
# Correlation plot
corrplot(cor_matrix_retail, method = "color",
         col = colorRampPalette(c("blue", "white", "red"))(100),
         addCoef.col = "black", # Add coefficients
         tl.col = "black",      # Color for labels
         title = "Correlation Plot for Price, Sales, and Inventory Levels",
         mar = c(0, 0, 1, 0))   # Margin adjustment
```



The correlation plots are created using *corrplot()* and shows:

- The positive relationship with Sales and Price; as Sales increases so does Price, vice-versa.
- The negative relationship between Sales and Inventory Levels; as Sales increases Inventory Levels decreases, and the increase in Inventory Levels is reflecting a decrease in Sales.
- The negative relationship between Price and Inventory Levels; greater inventory of an item results in a decrease in Price, while scarcity or lower Inventory Levels of an item leads to a price hike.

Hypothesis Test

Test the hypotheses that the correlations between the variables are zero and provide a 95% confidence interval.

Price and Sales:

- Null Hypothesis ($H_{0_{Price\ and\ Sales}}$): The correlation between the two variables is zero ($r = 0$).
- Alternative Hypothesis ($H_{1_{Price\ and\ Sales}}$): The correlation between the two variables is not zero ($r \neq 0$)

Price and Inventory Levels:

- Null Hypothesis ($H_{0_{Price\ and\ Inventory\ Levels}}$): The correlation between the two variables is zero ($r = 0$).
- Alternative Hypothesis ($H_{1_{Price\ and\ Inventory\ Levels}}$): The correlation between the two variables is not zero ($r \neq 0$)

Sales and Inventory Levels:

- Null Hypothesis ($H_{0_{Sales\ and\ Inventory\ Levels}}$): The correlation between the two variables is zero ($r = 0$).
- Alternative Hypothesis ($H_{1_{Sales\ and\ Inventory\ Levels}}$): The correlation between the two variables is not zero ($r \neq 0$)

Below is a method to run the correlation test *cor.test* for all variable pairings and make a it into a data frame.

```r
correlation_data_hyp <- retail_df[, c("Price", "Sales", "Inventory_Levels")]

variable_pairs <-
  combn(names(correlation_data_hyp),
        2,
        simplify = FALSE)

cor_test_results <-
  lapply(variable_pairs,
         function(pair) {
           test <-
             cor.test(correlation_data_hyp[[pair[1]]],
                      correlation_data_hyp[[pair[2]]],
                      use = "complete.obs")
  data.frame(
    Variable1 = pair[1],
    Variable2 = pair[2],
    Correlation = test$estimate,
    p_value = test$p.value,
    CI_lower = test$conf.int[1],
    CI_upper = test$conf.int[2]
  )
})

# Combine results into a single data frame
cor_test_summary <-
  do.call(rbind, cor_test_results)

colnames(cor_test_summary) <-
  c("Variable 1",
    "Variable 2",
    "Correlation (r)",
    "p-value",
    "CI Lower",
    "CI Upper")
```

The results are as follows.

```r
cor_test_summary %>%
  kable() %>%
  kable_classic()
```

	Variable 1	Variable 2	Correlation (r)	p-value	CI Lower	CI Upper
cor	Price	Sales	0.1027273	0.1477542	-0.0365344	0.2380752
cor1	Price	Inventory_Levels	-0.0402594	0.5713837	-0.1780061	0.0990348
cor2	Sales	Inventory_Levels	-0.0352962	0.6197608	-0.1731891	0.1039539

The *p-value* for these *cor.test*'s $p - value \geq 0.05$ shows the correlations are not statistically significant which means we fail to reject $H_{0_{(Price\ and\ Sales),\ (Price\ and\ Inventory\ Levels),\ (Sales\ and\ Inventory\ Levels)}}$

2.

Discussion:

- Explain the meaning of your findings and discuss the implications of the correlations for inventory management. Would you be concerned about multicollinearity in a potential regression model? Why or why not?

With Sales, Price and Inventory Levels not being strongly correlated or statistically significant, we basically conclude there's a lack of linear association between them. This means the variables are not directly influenced by each other and do not have a predictable impact that the business can go off of. More variables could be considered through data collection, such as the *Seasonal_Index* or a new driver can be introduced to the dataset. Regardless further investigation would definitely be needed. I would

Part 2

Linear Algebra and Pricing Strategy (5 Points)

Task:

1.

Price Elasticity of Demand:

- Use linear regression to model the relationship between Sales and Price (assuming Sales as the dependent variable).
- Invert the correlation matrix from your model, and calculate the precision matrix.
- Discuss the implications of the diagonal elements of the precision matrix (which are variance inflation factors).
- Perform LU decomposition on the correlation matrix and interpret the results in the context of price elasticity.

Linear Regression Model

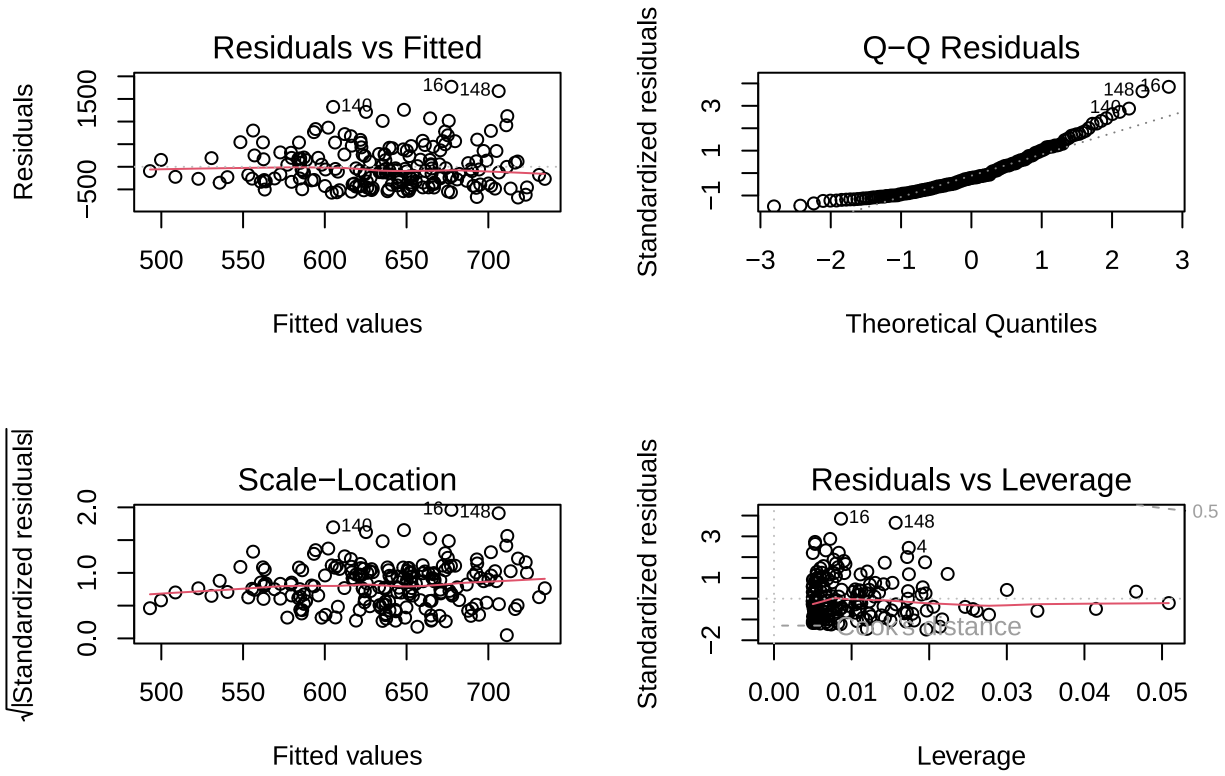
Use linear regression to model the relationship between Sales and Price (assuming Sales as the dependent variable).

```r
sales_price_model <- lm(Sales ~ Price, data = retail_df)
summary(sales_price_model)
```

```
##
## Call:
## lm(formula = Sales ~ Price, data = retail_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -679.54 -347.85  -98.63  241.12 1770.08
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  442.951    137.419   3.223  0.00148 **
## Price          9.916      6.824   1.453  0.14775
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 462.2 on 198 degrees of freedom
## Multiple R-squared:  0.01055,    Adjusted R-squared:  0.005556
## F-statistic: 2.112 on 1 and 198 DF,  p-value: 0.1478
```

```r
par(mfrow = c(2, 2)) # aesthetics for the plots
plot(sales_price_model)
```



With the model we see more of the same:

- $p - value = 0.1478 > 0.005$ therefore not statistically significant.
- $F - statistic = 2.112$ supports this.
- *Residual vs Fitted* and *Scaled-Location** plots graphically show scattered data points with no clear pattern.
- *Q-Q Plot* deviates from the diagonal line, which is an indication that it is not normalized. Again not surprising based on the analysis.

What's somewhat telling is the $R{-}squared = 0.01055$ showing that this does in fact capture 1.05% variability, making Price a weak predictor of Sale. So we can assume other variables may better explain what impacts this behavior.

Inverted Correlation Matrix

Invert the correlation matrix from your model, and calculate the precision matrix

In order to get the inverse matrix I can utilize the *solve()* function on my existing correlation matrix.

Our existing correlation matrix is

```
print(cor_matrix_retail)
```

```
##                        Price       Sales Inventory_Levels
## Price             1.00000000  0.10272730      -0.04025941
## Sales             0.10272730  1.00000000      -0.03529619
## Inventory_Levels -0.04025941 -0.03529619       1.00000000
```

while the inversion or precision matrix would result in

```
precision_matrix <-solve(cor_matrix_retail)
print(precision_matrix)
```

```
##                        Price       Sales Inventory_Levels
## Price             1.01203982 -0.10265390       0.03712083
## Sales            -0.10265390  1.01165983       0.03157495
## Inventory_Levels  0.03712083  0.03157495       1.00260894
```

Precision Matrix Diagonals

Discuss the implications of the diagonal elements of the precision matrix (which are variance inflation factors).

The Variance Inflation Factor (VIF) can be computed or found within the diagonal value of a precision matrix:

- $VIF_i - Diagonal\ Element\ of\ Precision\ Matrix\ for\ X_i$
- $Price = 1.01203982$
- $Sales = 1.01165983$
- $Inventory\_Levels = 1.00260894$

Our VIF can be written as $VIF_1 = \frac{1}{1-R_i^2}$ we can rewrite this as $R_i^2 = 1 - \frac{1}{VIF_i}$. Through substitution we can see the R^2 values for these VIFs are small.

```
1-1/1.01203982
```

```
## [1] 0.01189659
```

```
1-1/1.01165983
```

```
## [1] 0.01152545
```

```
1-1/1.00260894
```

```
## [1] 0.002602151
```

which would indicate $> 1.2\%$ of the variance is explained in X_i or minimal multicollinearity.

LU Decomposition

Perform LU decomposition on the correlation matrix and interpret the results in the context of price elasticity.

I'm leveraging the *Matrix* package with the function *lu()*. Quick understanding of the Lower and Upper and Lower Triangular interactions.

- LU decomposition is breaking of matrix M to $L \cdot U$
- U represents the amount of variance that remains independently.
- Basically a purely independent variable would contain a 1.00 or 100% of its own variance.

- The lower shows a measurement of the residual dependencies or the direct influence one variable has to another.

```
retail_lu_decomp <- lu(cor_matrix_retail)
retail_expand_decomp <- expand(retail_lu_decomp)
retail_expand_decomp$L
```

```
## 3 x 3 Matrix of class "dtrMatrix" (unitriangular)
##      [,1]        [,2]        [,3]
## [1,]  1.00000000           .           .
## [2,]  0.10272730  1.00000000           .
## [3,] -0.04025941 -0.03149279  1.00000000
```

Our L shows the dependency of *Sales* and *Price*.

- 0.10272730 is showing that ≈ 10.3% of variability regarding *Price* is explained by *Sales*
- The third row -0.04025941 and -0.03149279 explains this for *Price* and *Sales* as it relates to their impact on inventory. The negative indicates a negative relationship but the value close to 0 indicates explain minimal variability. So we're probably not looking at a linear model and we would likely have to search for a variable that is more significant than the ones we are analyzing.

```
retail_expand_decomp$U
```

```
## 3 x 3 Matrix of class "dtrMatrix"
##      [,1]        [,2]        [,3]
## [1,]  1.00000000  0.10272730 -0.04025941
## [2,]           .  0.98944710 -0.03116045
## [3,]           .           .  0.99739785
```

Our values for the diagonal of U 0.98944710 and 0.99739785 Shows these values maintain a large portion of there variance or are close to independent.

Just to verify the process worked I am comparing a reconstructed matrix with the original

```
retail_expand_decomp$L %*% retail_expand_decomp$U
```

```
## 3 x 3 Matrix of class "dgeMatrix"
##             [,1]        [,2]        [,3]
## [1,]  1.00000000  0.10272730 -0.04025941
## [2,]  0.10272730  1.00000000 -0.03529619
## [3,] -0.04025941 -0.03529619  1.00000000
```

```
cor_matrix_retail
```

```
##                        Price       Sales Inventory_Levels
## Price             1.00000000  0.10272730      -0.04025941
## Sales             0.10272730  1.00000000      -0.03529619
## Inventory_Levels -0.04025941 -0.03529619       1.00000000
```

The Upper triangular interaction really clarifies a lack of multicollinearity and *Price*'s 0.10272730 value makes it the best predictor as it relates to *Sales* which is good news for forecasting. *Price* and *Sale* however do not appear currently to be a good predictor of inventory.

Part 3:

Calculus-Based Probability & Statistics for Sales Forecasting (5 Points)

Task:

1.

Sales Forecasting Using Exponential Distribution:

- Identify a variable in the dataset that is skewed to the right (e.g., Sales or Price) and fit an exponential distribution to this data using the fitdistr function.
- Generate 1,000 samples from the fitted exponential distribution and compare a histogram of these samples with the original data's histogram.
- Calculate the 5th and 95th percentiles using the cumulative distribution function (CDF) of the exponential distribution.
- Compute a 95% confidence interval for the original data assuming normality and compare it with the empirical percentiles.

Right-skewed | Exponential Distr.

Identify a variable in the dataset that is skewed to the right (e.g., Sales or Price) and fit an exponential distribution to this data using the fitdistr function

I know from earlier that this is Sales, but Ill test the skewness of both for completeness.

```
skewness(retail_df$Sales, na.rm = TRUE)
```

```
## [1] 1.217147
```

```
skewness(retail_df$Price, na.rm = TRUE)
```

```
## [1] -0.4534494
```

Now Ill run the fitdistr.

```
sale_exp_fit <- fitdistr(sales_retail_df, "exponential")
sale_exp_fit
```

```
##        rate
##   0.0015700652
##  (0.0001110204)
```

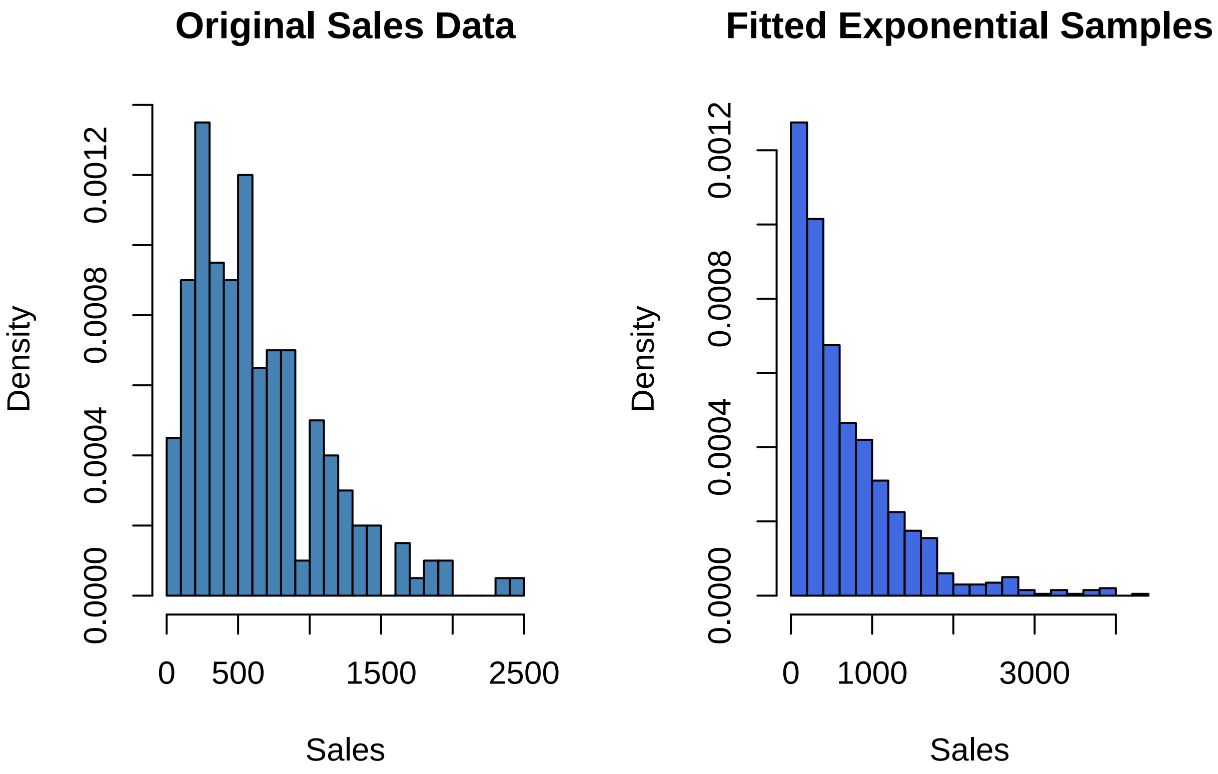
Samples

Generate 1,000 samples from the fitted exponential distribution and compare a histogram of these samples with the original data's histogram.

Using this lambda I'll generate my 100 samples

```
exp_fit_samples <- rexp(1000, rate = sale_exp_fit$estimate)
```

```
# Setting up panels like earlier
par(mfrow = c(1, 2))

# Plot the histogram of the original Sales data
hist(retail_df$Sales, breaks = 30, main = "Original Sales Data",
     xlab = "Sales", col = "steelblue", freq = FALSE)

# Plot the histogram of the generated samples
hist(exp_fit_samples, breaks = 30, main = "Fitted Exponential Samples",
     xlab = "Sales", col = "royalblue", freq = FALSE)
```



5th and 95th Percentile

Calculate the 5th and 95th percentiles using the cumulative distribution function (CDF) of the exponential distribution.

So our CDF of an Exponential distribution given a probability p is

$x = -\frac{ln(1-p)}{\lambda}$

where our p is either 0.05 or 0.95 respectively

So our 5th percentile is

```
-log(1 - 0.05) / sale_exp_fit$estimate
```

```
##     rate
## 32.66953
```

and our 95th percentile is

```
-log(1 - 0.95) / sale_exp_fit$estimate
```

```
##    rate
## 1908.03
```

95 % Confidence Interval

Compute a 95% confidence interval for the original data assuming normality and compare it with the empirical percentiles.

Formula for confidence interval is $CI = \bar{x} \pm z \cdot \frac{\sigma}{\sqrt{n}}$ but Ill be leveraging the *HMISC* package to calculate this.

```
list(
  CI_Normality = smean.cl.normal(retail_df$Sales, conf.int = 0.95),
  Empirical_Percentiles =
    c(quantile(retail_df$Sales, 0.05),
      quantile(retail_df$Sales, 0.95))
)
```

```
## $CI_Normality
##     Mean    Lower    Upper
## 636.9162 572.2866 701.5458
##
## $Empirical_Percentiles
##        5%       95%
##  104.9028 1502.2498
```

2.

Discussion:

- Discuss how well the exponential distribution models the data and what this implies for forecasting future sales or pricing. Consider whether a different distribution might be more appropriate.

I felt the Gamma distribution earlier better captured the behavior of *Sales* personally. Sure the exponential distribution shows the skewness, but it makes the sales at lower values appear more dense while removing the second peak noted within this values from the original data. The 5% percentile indicates that 5% of our sales are below ≈ 32.67 but again as the exponential is more dense at lower values, it could be possible that this is overstating or over approximating.

The same can be said about overstating with the higher values, as with the original data, values at 2500 appeared to be more like outliers and the exponential distribution stretches more to ≈ 4000

For normality the values centralize around the mean. I is definitely ideal to have a normalized data set for forecasting purposes, but the transformation does not capture the skewness and hence will likely be inaccurate, as it will not capture extreme outcomes like large sales. Overall Gamma was had the most precision, and lognormal was also a great choice for matching the skewed behavior.

Part 4

Regression Modeling for Inventory Optimization (10 Points)

Task:

1.

Multiple Regression Model:

- Build a multiple regression model to predict Inventory_Levels based on Sales, Lead_Time_Days, and Price.
- Provide a full summary of your model, including coefficients, R-squared value, and residual analysis.

```
inv_model <- lm(Inventory_Levels ~ Sales + Lead_Time_Days + Price, data = retail_df)
summary(inv_model)
```

```
##
## Call:
## lm(formula = Inventory_Levels ~ Sales + Lead_Time_Days + Price,
##     data = retail_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.54 -118.07   -7.68  111.81  372.56
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     464.792662  61.321852   7.580 1.35e-12 ***
## Sales            -0.007809   0.023955  -0.326    0.745
## Lead_Time_Days    7.316793   5.293049   1.382    0.168
## Price            -1.087778   2.305846  -0.472    0.638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 155.3 on 196 degrees of freedom
## Multiple R-squared:  0.01223,    Adjusted R-squared:  -0.002887
## F-statistic: 0.8091 on 3 and 196 DF,  p-value: 0.4902
```

Coefficients

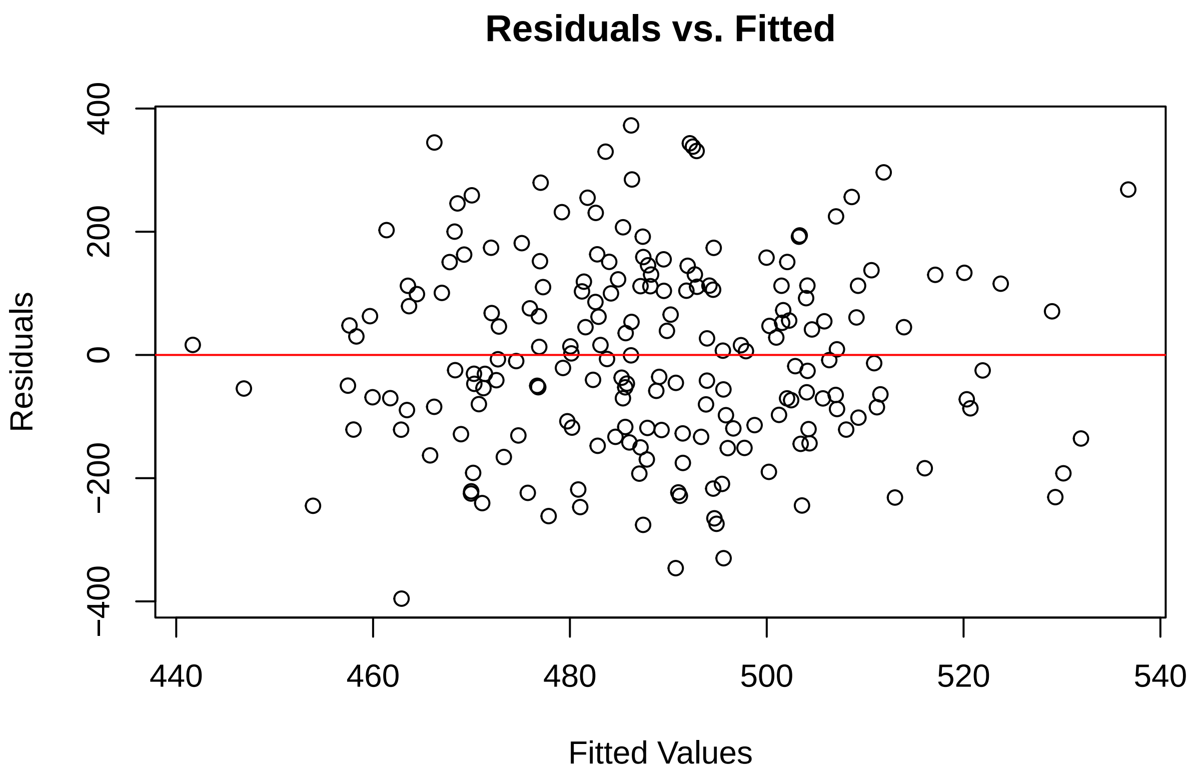
- Sales: -0.007809, for every 1 unit of increase of Sales, Inventory decreases by 0.007809. With a $p - value = 0.745$ its not very statistically significant b/c theres a 74.5% chance that the observed relationship happened by chance.
- Lead_Times_Days: Basically means time in days to complete a process. SO for every day it takes to complete, inventory increases by 7.316793, with a $p - value = 0.168$ the chance this observation was purely by chance is 16, which is still technically considered statistically not significant but still the highest performing.
- Price: For every 1 unit increase in price Inventory decreases 1.087778. again $p - value = 0.638$ so not statistically significant.

R-Squared

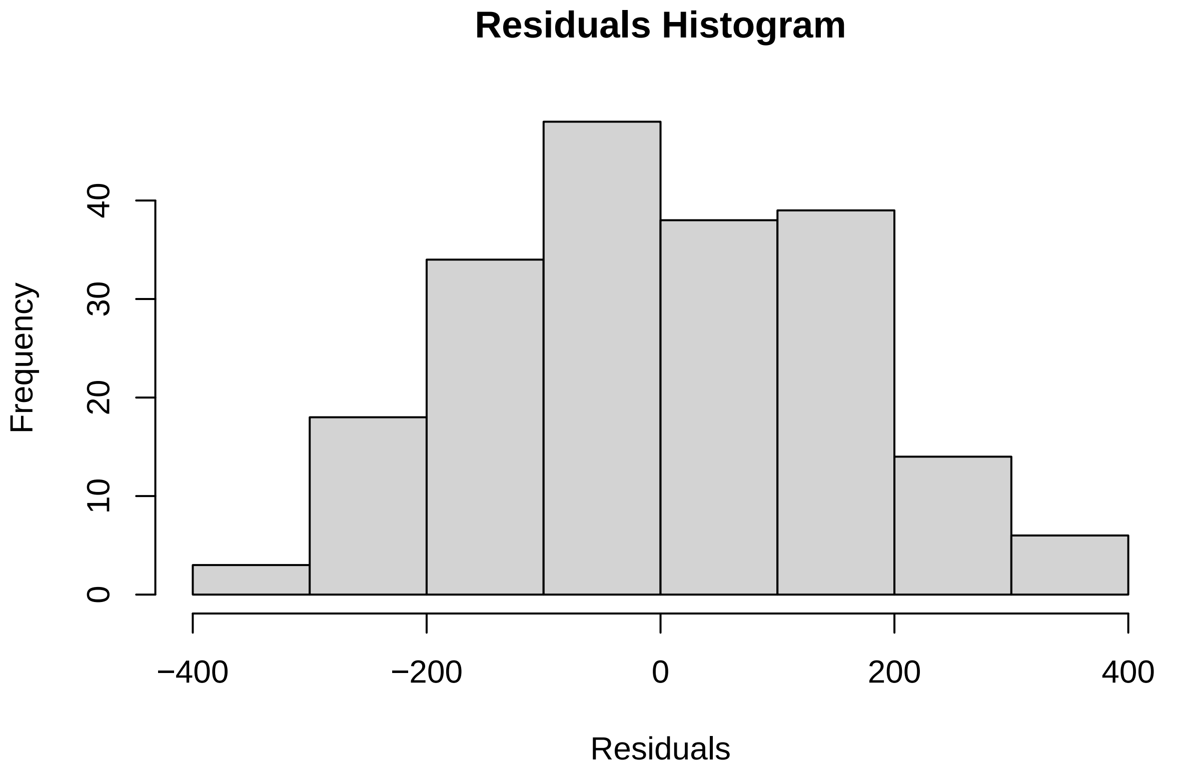
R-squared - supports the lack of variance captured at $\approx 1.12\%$ so this doesn't do a good job of explaining inventory behavior, the adjusted of -0.002887 just shows that it is even less captured by these variables.

Residual Analysis

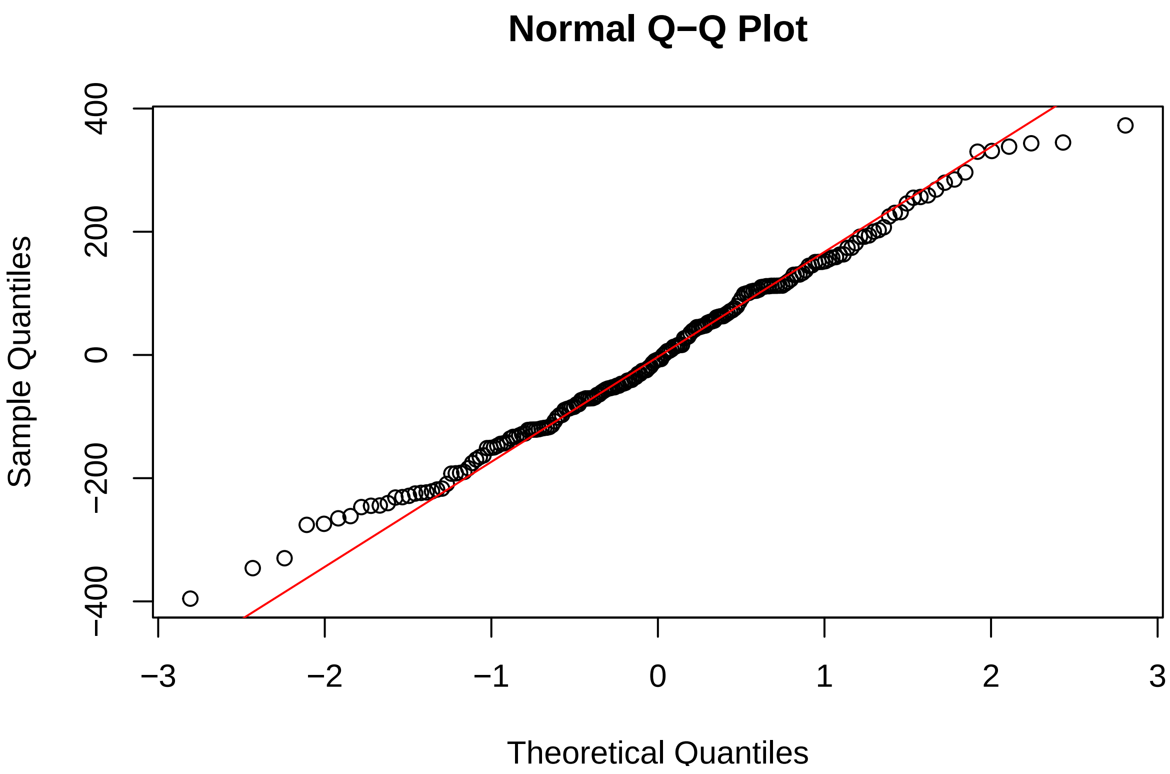
```r
# Residuals vs. Fitted values
plot(inv_model$fitted.values, inv_model$residuals,
     main = "Residuals vs. Fitted", xlab = "Fitted Values", ylab = "Residuals")
abline(h = 0, col = "red")
```



```
# Histogram of residuals
hist(inv_model$residuals, main = "Residuals Histogram", xlab = "Residuals")
```



```
# QQ plot of residuals
qqnorm(inv_model$residuals)
qqline(inv_model$residuals, col = "red")
```



The QQ Plot and the histogram indicate normality, but Residual vs. Fitted supports that these values have a an issue with heteroscadicity. Which I believe supports the evidence of a lack of relationship between these variables.

2.

Optimization:

- Use your model to optimize inventory levels for a peak sales season, balancing minimizing stockouts with minimizing overstock.

Assuming where not going to use the Seasonality Metric in the data set, our goal is to ensure we have enough in, or do not run out of stock.

Since I'm not really using variables that are good predictors, assuming our historical data would be a strong predictor for seasonal spikes makes no sense. My approach to optimization is straigh forward, lets calculate for 10%, 20% and 50% spikes. Business can decide the approach they want to use for the coming year. We can estimate if the percent chosen supported the inventory. This can be done simply by noting if there was still any surplus. From there we can adjust abritrarily or, if enough data has be stored, with the right variables having been researched and implemented, adjust the model altogether. The 95% confidence level is a $Z = 1.645$ we can use this to predict the safety stock needed by each percentage.

```
# Define sales increase scenarios (10%, 30%, and 50%)
sales_scenarios <- c(1.1, 1.3, 1.5)  # Multipliers for increase

inventory_scenarios <- lapply(sales_scenarios, function(increase) {
  peak_sales <- mean(retail_df$Sales) * increase
  peak_inputs <- data.frame(
    Sales = peak_sales,
    Lead_Time_Days = mean(retail_df$Lead_Time_Days),
    Price = mean(retail_df$Price)
  )
  predicted_inventory <- predict(inv_model, newdata = peak_inputs)
  safety_stock <- 1.645 * sd(retail_df$Sales) * sqrt(mean(retail_df$Lead_Time_Days))
  optimal_inventory <- predicted_inventory + safety_stock
  return(optimal_inventory)
})

# Name the scenarios for easy reference
names(inventory_scenarios) <- c("10% Increase", "30% Increase", "50% Increase")
inventory_scenarios
```

```
## $`10% Increase`
##        1
## 2481.301
##
## $`30% Increase`
##        1
## 2480.307
##
## $`50% Increase`
##        1
## 2479.312
```

As a result of reviewing the inventory results at a spike of these levels, we see that the predicted inventory units needed are very close. Although not intentional, it does support having a stock close to 2480 units, for expected peak times.

Referenced sites.

i. Statology fitdistr-r

ii. rdocumentation MASS package fitdistr

iii. Statology - fit gamma distribution to dataset in r

iv. Wiki Gamma Distribution

v. rdocumentation qualtile

vi. statlect lognormal distribution

vii. Penn State University Online Stat 200 book

viii. Geeks for Geeks: simple linear regression using r

ix. Yes I reference Reddit please do not judge

x. LU Decomposition of Square Matrix

xi. Three simple matrix decompositions

xii. LU Decomposition Method for Solving Simultaneous Linear Equations.

xiii. Geeks for Geeks LU decomposition linear equations