Project Proposal

Benjamin Leidig, Monte Thomas, Harmony Pham

2025-03-26

Section: GR

Research Questions

- How can we forecast daily hotel total revenue using historical revenue data?
- Are there seasonal fluctuations in daily total hotel revenue?
- Do any external events (sport events, family weekends, etc.) effect daily hotel total revenue?

Potential Sources

- (https://medium.com/@chenycy/predict-hotel-demands-leveraging-time-series-forecasting-techniques-62e25606f273)
- (https://pure.psu.edu/en/publications/forecasting-hotel-occupancy-rates-with-time-series-models-an-empi)

Dataset Cleaning

```r
# Create a function to clean data
clean_hotel_data <- function(file_path, year){
  read.csv(file_path) %>%
    pivot_longer(cols = -X, names_to = "Date", values_to = "Value") %>%
    pivot_wider(names_from = "X", values_from = "Value") %>%
    mutate(
      Date = str_remove(Date, "^[A-Za-z]+") %>%
             paste0(".", year) %>%
             mdy(),
      Weekday = wday(Date, label = TRUE)
    ) %>%
    select(Date, Weekday, Occupancy, ADR, TOTAL_REVENUE)
}
```

```r
# Apply function to data1 (year 2024)
data1 <- clean_hotel_data("hoteldata24.csv", "2024")
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## incomplete final line found by readTableHeader on 'hoteldata24.csv'
```

```r
# Apply function to data2 (year 2025)
data2 <- clean_hotel_data("hoteldata25.csv", "2025")
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## incomplete final line found by readTableHeader on 'hoteldata25.csv'
```

```r
# Combine datasets
hotel_data <- bind_rows(data1, data2)
str(hotel_data)
```

```
## tibble [365 x 5] (S3: tbl_df/tbl/data.frame)
##  $ Date         : Date[1:365], format: "2024-03-28" "2024-03-29" ...
##  $ Weekday      : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 5 6 7 1 2 3 4 5 6 7 ...
##  $ Occupancy    : num [1:365] 0.98 0.9 0.98 0.58 0.81 0.92 0.97 0.8 0.93 0.98 ...
##  $ ADR          : num [1:365] 151 150 144 104 121 ...
##  $ TOTAL_REVENUE: num [1:365] 18259 16605 17403 7059 11055 ...
```

```r
head(hotel_data)
```

```
## # A tibble: 6 x 5
##   Date       Weekday Occupancy   ADR TOTAL_REVENUE
##   <date>     <ord>       <dbl> <dbl>         <dbl>
## 1 2024-03-28 Thu          0.98  151.        18259.
## 2 2024-03-29 Fri          0.9   150.        16605.
## 3 2024-03-30 Sat          0.98  144.        17403.
## 4 2024-03-31 Sun          0.58  104.         7059.
## 5 2024-04-01 Mon          0.81  121.        11055.
## 6 2024-04-02 Tue          0.92  147.        16073.
```

Visualizations

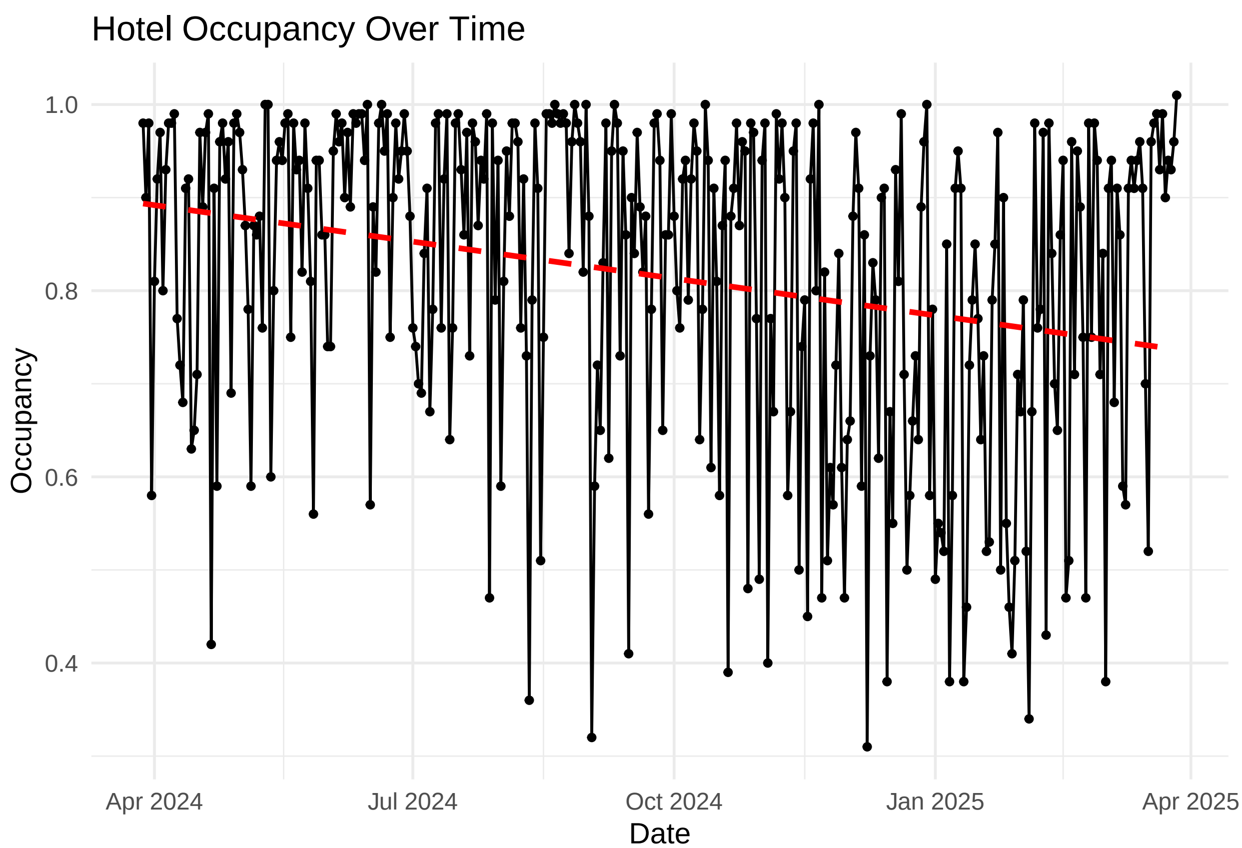
```r
summary(hotel_data)
```

```
##       Date              Weekday    Occupancy           ADR
##  Min.   :2024-03-28   Sun:52   Min.   :0.3100   Min.   : 78.03
##  1st Qu.:2024-06-27   Mon:52   1st Qu.:0.7100   1st Qu.:130.51
##  Median :2024-09-26   Tue:52   Median :0.8800   Median :156.80
##  Mean   :2024-09-26   Wed:52   Mean   :0.8152   Mean   :162.95
##  3rd Qu.:2024-12-26   Thu:53   3rd Qu.:0.9600   3rd Qu.:178.05
##  Max.   :2025-03-27   Fri:52   Max.   :1.0100   Max.   :441.33
##                       Sat:52
##  TOTAL_REVENUE
##  Min.   : 3364
##  1st Qu.:10080
##  Median :16688
##  Mean   :16466
##  3rd Qu.:20558
##  Max.   :52518
##
```

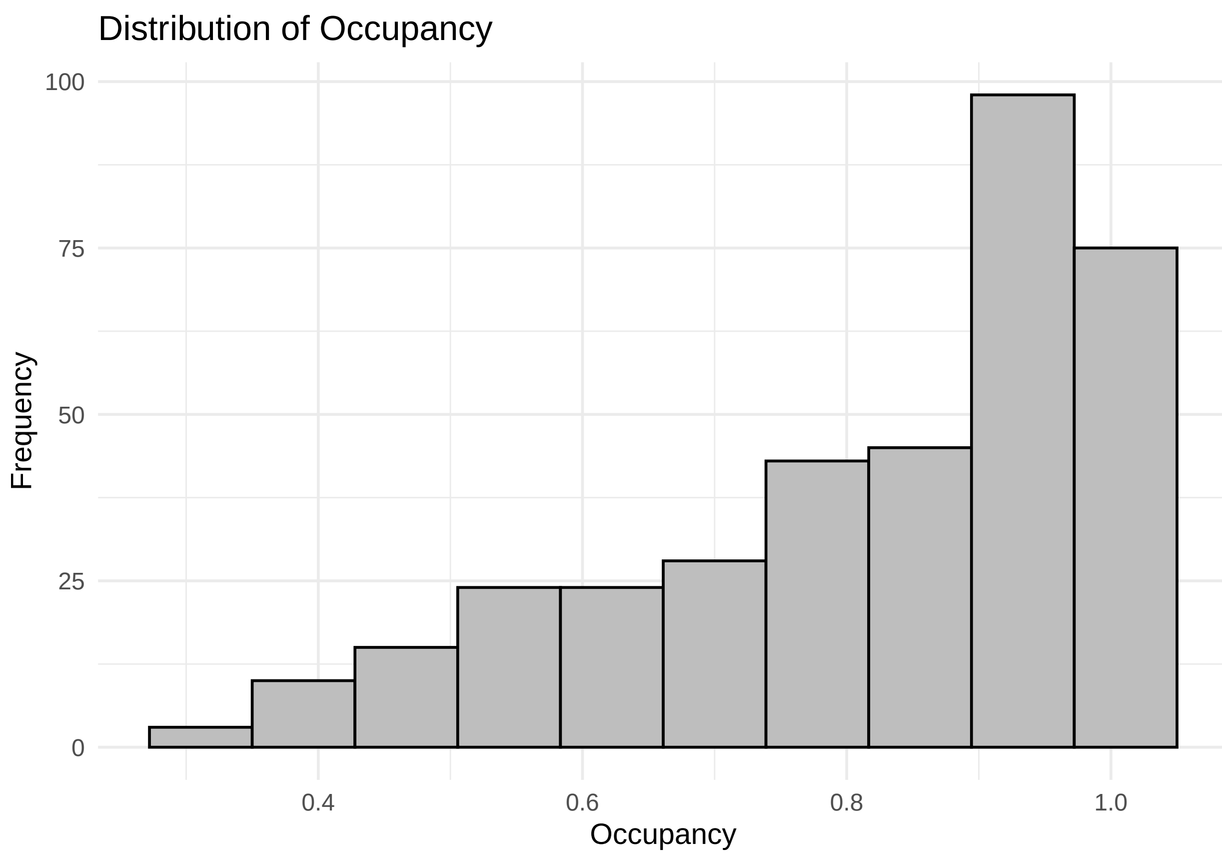
Occupancy Rate Visualizations

```r
# Plot Occupancy over Time
ggplot(hotel_data, aes(x = Date, y = Occupancy)) +
  geom_line(color = "black", linewidth = 0.5) +
  geom_point(color = "black", size = 1) +
  geom_smooth(method = "lm", color = "red", se = FALSE, linewidth = 1, linetype = 2) +
  labs(title = "Hotel Occupancy Over Time",
       x = "Date",
       y = "Occupancy") +
  theme_minimal()
```

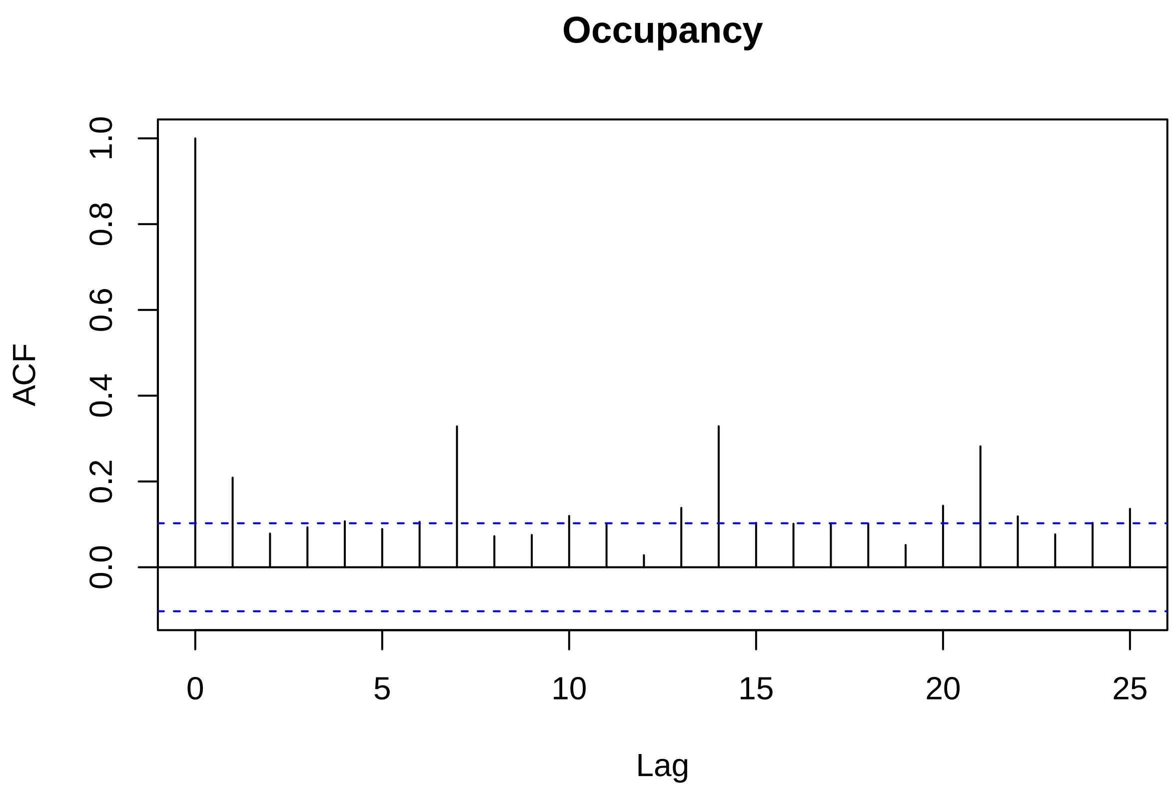
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Plot Occupancy distribution
ggplot(hotel_data, aes(x = Occupancy)) +
  geom_histogram(bins = 10, fill = "gray", color = "black") +
  labs(title = "Distribution of Occupancy",
       x = "Occupancy",
       y = "Frequency") +
  theme_minimal()
```




```
acf(hotel_data$Occupancy, main = 'Occupancy')
```




```r
adf.test(hotel_data$Occupancy); kpss.test(hotel_data$Occupancy, null = 'Trend')
```

```
## Warning in adf.test(hotel_data$Occupancy): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  hotel_data$Occupancy
## Dickey-Fuller = -4.7615, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

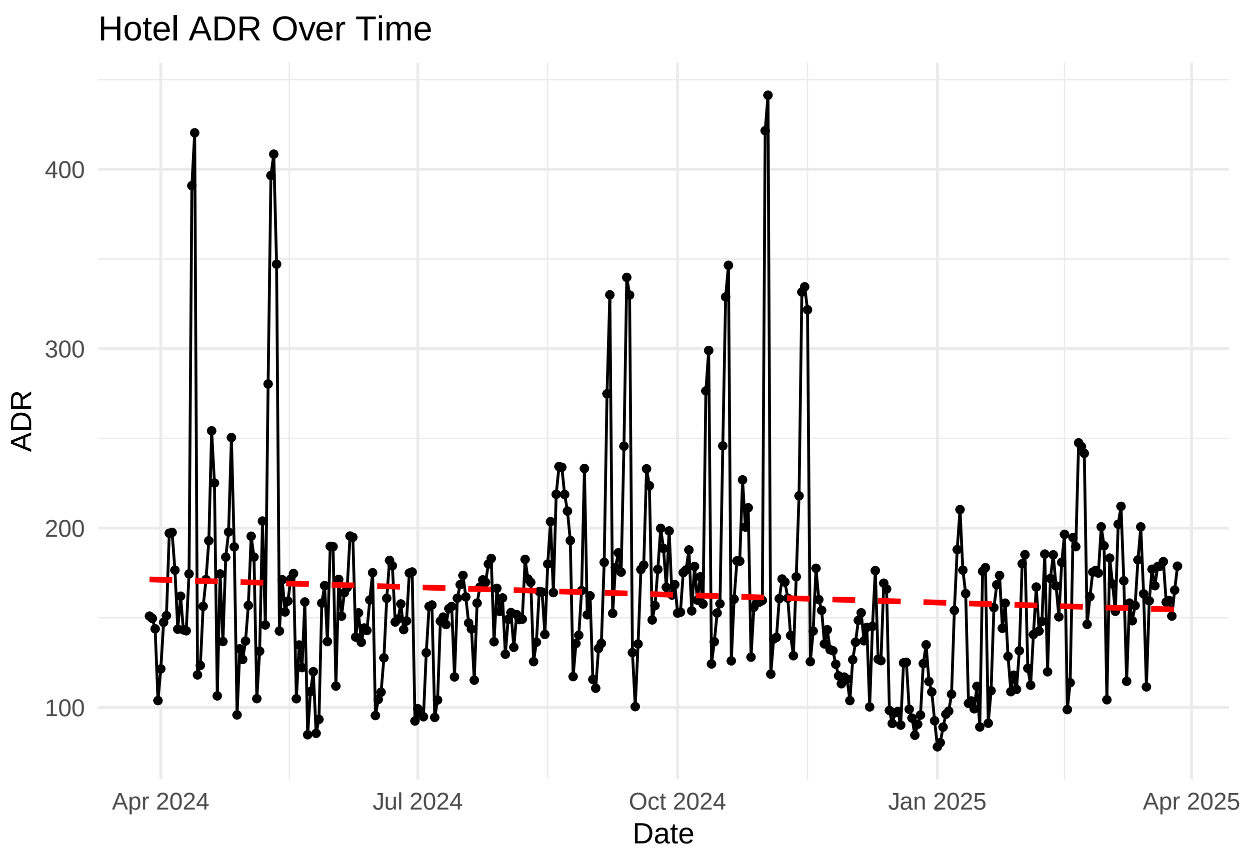
```
## Warning in kpss.test(hotel_data$Occupancy, null = "Trend"): p-value smaller
## than printed p-value
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  hotel_data$Occupancy
## KPSS Trend = 0.24554, Truncation lag parameter = 5, p-value = 0.01
```

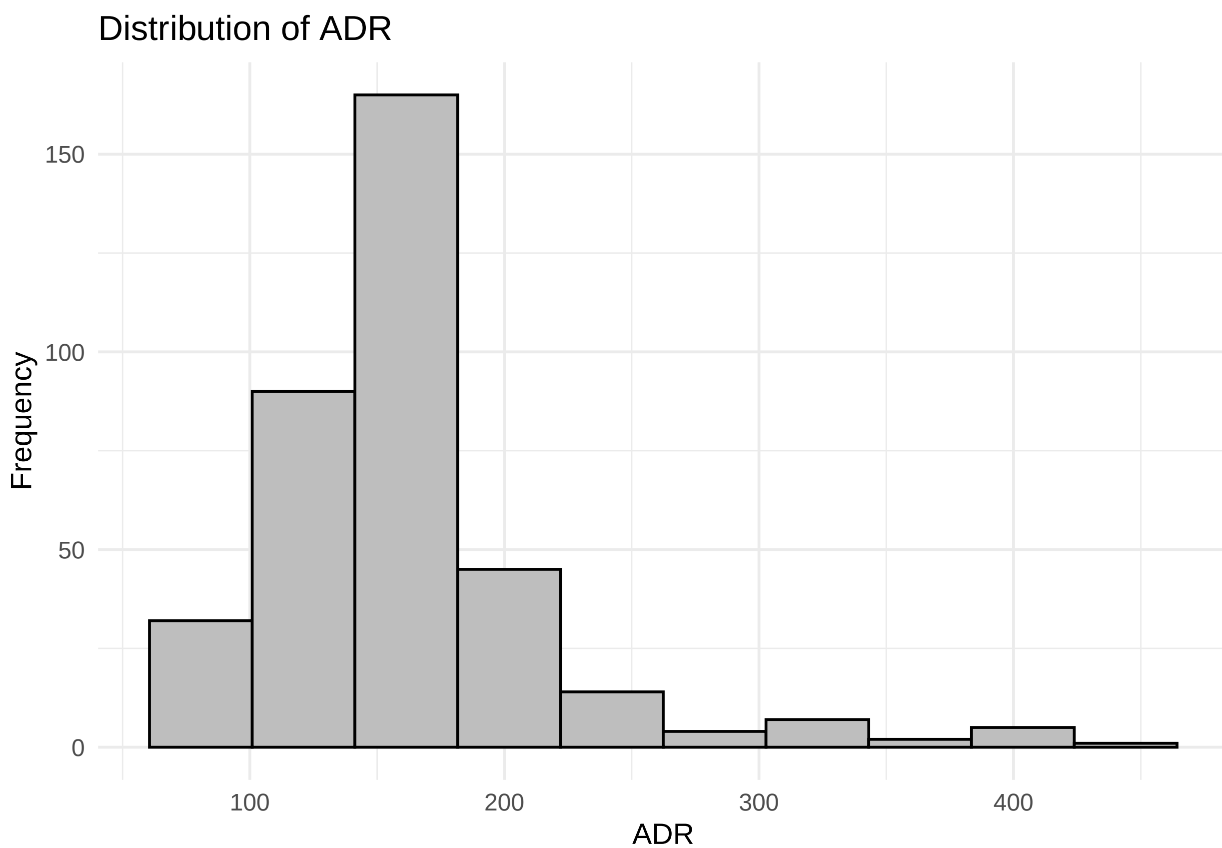
ADR Visualizations

```r
# Plot ADR over Time
ggplot(hotel_data, aes(x = Date, y = ADR)) +
  geom_line(color = "black", linewidth = 0.5) +
  geom_point(color = "black", size = 1) +
  geom_smooth(method = "lm", color = "red", se = FALSE, linewidth = 1, linetype = 2) +
  labs(title = "Hotel ADR Over Time",
       x = "Date",
       y = "ADR") +
  theme_minimal()
```

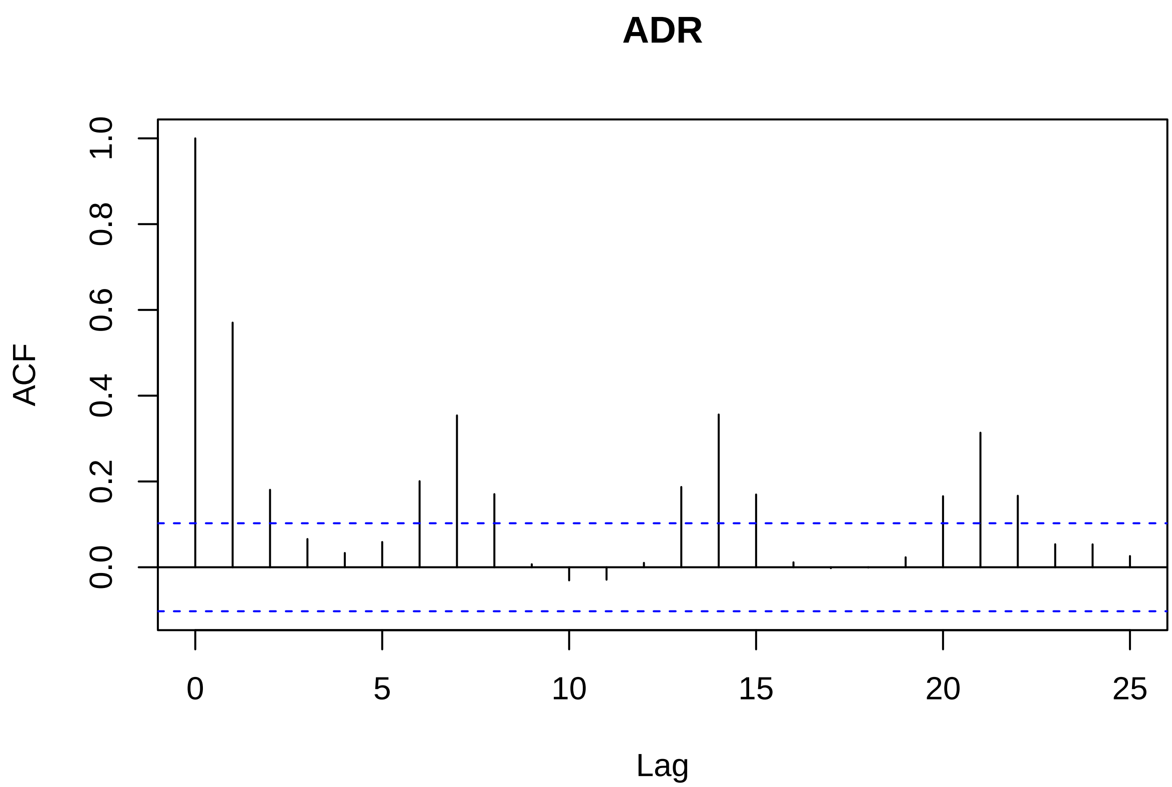
```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Plot ADR distribution
ggplot(hotel_data, aes(x = ADR)) +
  geom_histogram(bins = 10, fill = "grey", color = "black") +
  labs(title = "Distribution of ADR",
       x = "ADR",
       y = "Frequency") +
  theme_minimal()
```




```r
acf(hotel_data$ADR, main = 'ADR')
```




```r
adf.test(hotel_data$ADR); kpss.test(hotel_data$ADR, null = 'Trend')
```

```
## Warning in adf.test(hotel_data$ADR): p-value smaller than printed p-value
```

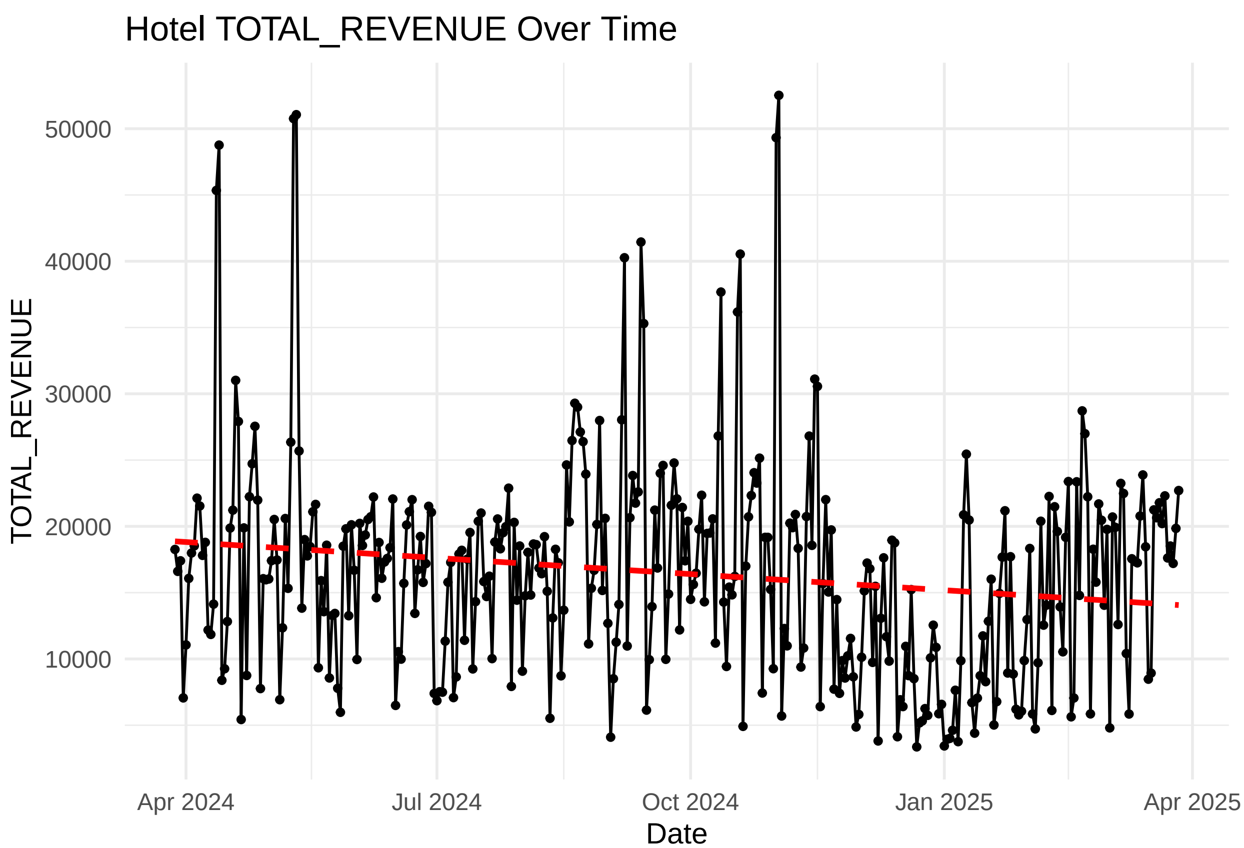
```
##
##  Augmented Dickey-Fuller Test
##
## data:  hotel_data$ADR
## Dickey-Fuller = -5.1263, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  hotel_data$ADR
## KPSS Trend = 0.1794, Truncation lag parameter = 5, p-value = 0.02373
```

Total Revenue Visualizations

```r
# Plot TOTAL_REVENUE over Time
ggplot(hotel_data, aes(x = Date, y = TOTAL_REVENUE)) +
  geom_line(color = "black", linewidth = 0.5) +
  geom_point(color = "black", size = 1) +
  geom_smooth(method = "lm", color = "red", se = FALSE, linewidth = 1, linetype = 2) +
  labs(title = "Hotel TOTAL_REVENUE Over Time",
       x = "Date",
       y = "TOTAL_REVENUE") +
  theme_minimal()
```

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

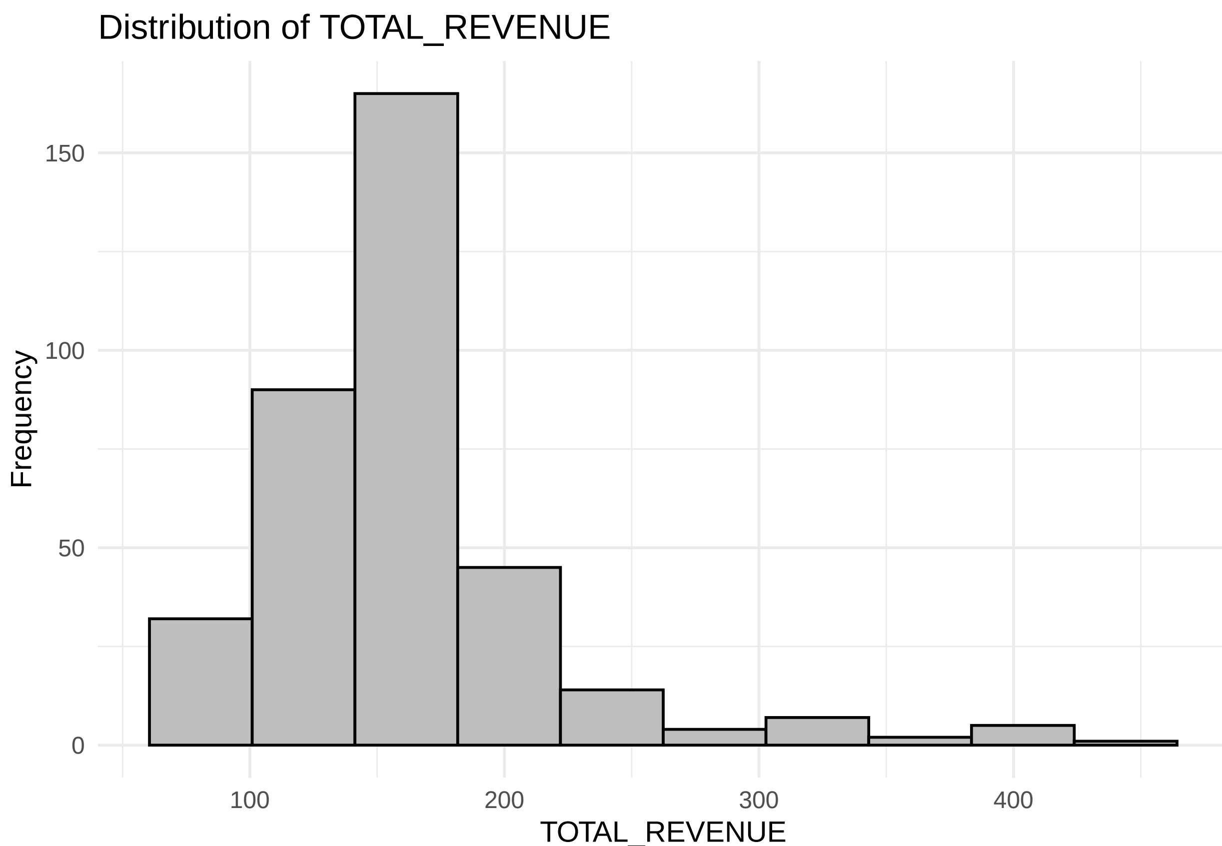



```r
adf.test(hotel_data$TOTAL_REVENUE); kpss.test(hotel_data$TOTAL_REVENUE, null = 'Trend')
```

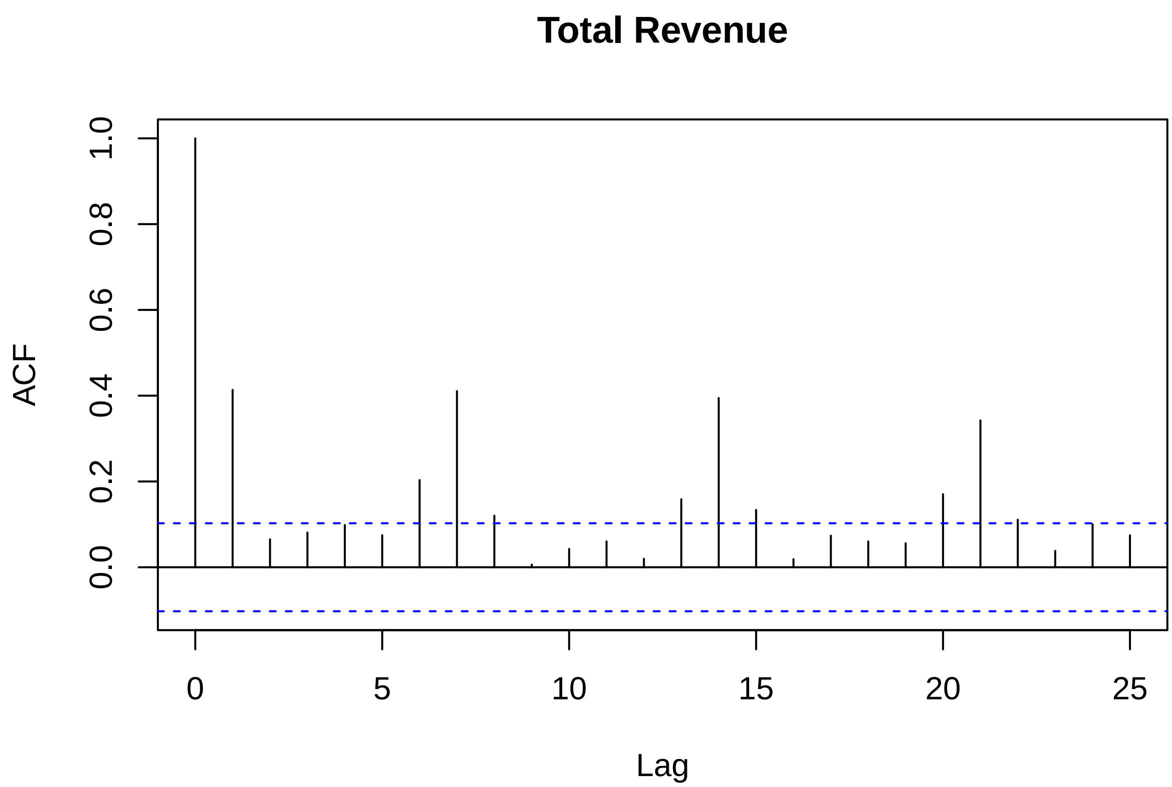
```
## Warning in adf.test(hotel_data$TOTAL_REVENUE): p-value smaller than printed
## p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  hotel_data$TOTAL_REVENUE
## Dickey-Fuller = -4.736, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  hotel_data$TOTAL_REVENUE
## KPSS Trend = 0.20772, Truncation lag parameter = 5, p-value = 0.01311
```

```r
# Plot TOTAL_REVENUE distribution
ggplot(hotel_data, aes(x = ADR)) +
  geom_histogram(bins = 10, fill = "grey", color = "black") +
  labs(title = "Distribution of TOTAL_REVENUE",
       x = "TOTAL_REVENUE",
       y = "Frequency") +
  theme_minimal()
```




```r
acf(hotel_data$TOTAL_REVENUE, main = 'Total Revenue')
```




```r
adf.test(hotel_data$TOTAL_REVENUE); kpss.test(hotel_data$TOTAL_REVENUE, null = 'Trend')
```

```
## Warning in adf.test(hotel_data$TOTAL_REVENUE): p-value smaller than printed
## p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  hotel_data$TOTAL_REVENUE
## Dickey-Fuller = -4.736, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  hotel_data$TOTAL_REVENUE
## KPSS Trend = 0.20772, Truncation lag parameter = 5, p-value = 0.01311
```

Dataset Description

For this project, we are using real data collected from the I Hotel & Illinois Conference Center. The dataset–stored in the dataframe object, `hotel_data`–consists of five variables: `Date` (YYYY-MM-DD), `Weekday` (Sun, Mon, Tue, Wed, Thu, Fri, Sat), `Occupancy` (a percentage represented as a decimal), `ADR` (Average Daily Rate; (room revenue)/(rooms sold); average daily revenue in USD earned per occupant), and `TOTAL_REVENUE` (total daily revenue in USD). Each observations represents an individual day, between 2024/03/28 and 2025/03/27.

Occupancy rates tends to be skewed to the left, with a mode of approximately 0.9. Looking at the time series plot of occupancy rates, there appears to be a non-constant mean function that is decreasing, although there is also a potential seasonal component as well. There appears to be a non-constant variance that increases with time. An ADF test yields a p-value of less than 0.01, meaning that the time series could be stationary. However, a KPSS test (with H_0 : trend stationarity) also yields a p-value of less than 0.01, which means the time series isn't trend-stationary, either. Although these test are useful, we conclude via visual inspection that the series is neither stationary nor trend-stationary.

ADR tends to be skewed to the right, with a mode of approximately 150. Looking at the time series plot of ADR, there appears to be a constant mean function, although there is also a strong seasonal component. There appears to be a non-constant variance, with variance increasing during peak seasons and decreasing during the depressions. An ADF test yields a p-value of less than 0.01, meaning that the time series could be stationary. However, a KPSS test (with H_0 : trend stationarity) yields a p-value of 0.02373, which, if using a significance level of 0.05, means the time series isn't trend stationary. Although these test are useful, we conclude via visual inspection that the series is neither stationary nor trend-stationary.

Total revenue tends to be skewed to the right, with a mode of approximately 150. Looking at the time series plot of total revenue, there appears to be a non-constant mean function. There is also not a prominent seasonal component. There appears to be a non-constant variance, with variance increasing through April to May and September to November. An ADF test yields a p-value of less than 0.01, meaning that the time series could be stationary. However, a KPSS test (with H_0 : trend stationarity) yields a p-value of 0.01311, which, if using a significance level of 0.05, means the time series isn't trend stationary. Although these test are useful, we conclude via visual inspection that the series is neither stationary nor trend-stationary.

According to the sample ACF plots, there is a strong seasonal component in all of the time series. In particular, starting at lag 0 ($h = 0$), the sample ACF peaks at increments of 7 (i.e. at $h = 7$, $h = 14$, $h = 21$, etc.). This implies that all three variables have an association with the day of the week.