Formative Assessment 5

PATAYON, SPIKE LEE-ROY V

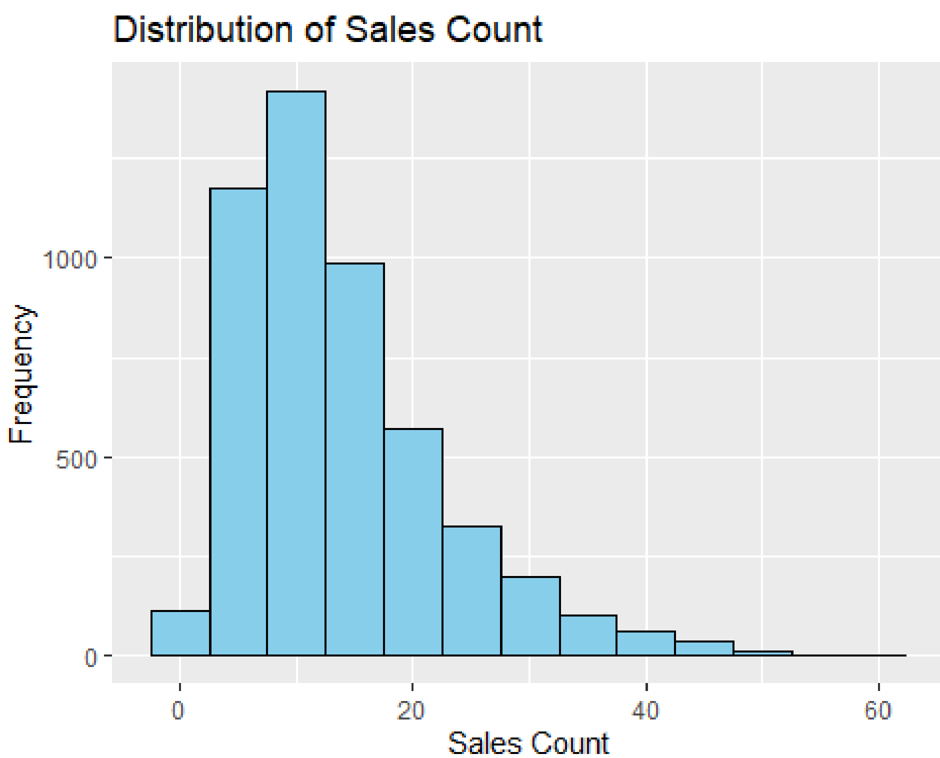
2025-05-02

Step 1:

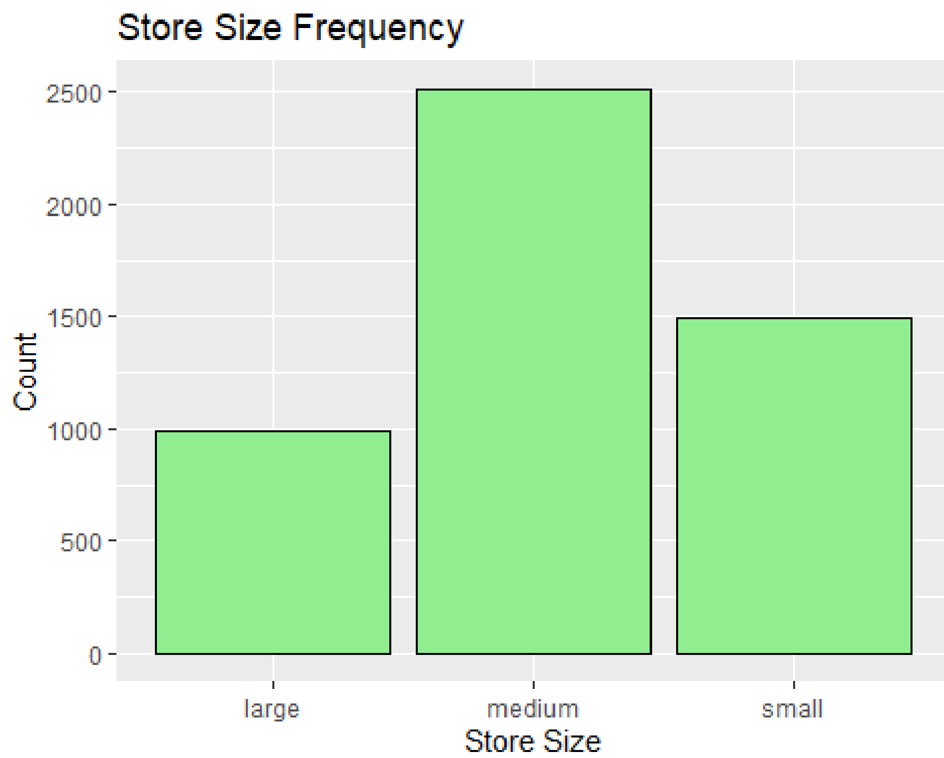
```
# Load libraries
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

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

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

sales_data <- read.csv("C:\\Users\\spike\\Downloads\\store_sales_data.csv")

head(sales_data)

##   day_of_week promo holiday store_size sales_count
## 1           6     0       0     medium          18
## 2           3     0       0     medium          13
## 3           4     0       0      large          24
## 4           6     1       0      small          16
## 5           2     0       0     medium          11
## 6           4     0       1     medium          13

str(sales_data)

## 'data.frame':    5000 obs. of  5 variables:
##  $ day_of_week: int  6 3 4 6 2 4 4 6 1 2 ...
##  $ promo      : int  0 0 0 1 0 0 0 1 1 1 ...
##  $ holiday    : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ store_size : chr  "medium" "medium" "large" "small" ...
##  $ sales_count: int  18 13 24 16 11 13 12 34 19 8 ...

summary(sales_data)

##   day_of_week        promo           holiday        store_size
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Length:5000
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :3.000   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :2.985   Mean   :0.3012   Mean   :0.0956
```

```
##   3rd Qu.:5.000    3rd Qu.:1.0000    3rd Qu.:0.0000
##   Max.   :6.000    Max.   :1.0000    Max.   :1.0000
##    sales_count
##   Min.   : 0.00
##   1st Qu.: 7.00
##   Median :12.00
##   Mean   :13.73
##   3rd Qu.:18.00
##   Max.   :61.00
```

```
# Histogram of sales count
ggplot(sales_data, aes(x = sales_count)) +
  geom_histogram(binwidth = 5, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Sales Count", x = "Sales Count", y =
"Frequency")
```




```
# Bar plot: Frequency of each store_size
ggplot(sales_data, aes(x = store_size)) +
  geom_bar(fill = "lightgreen", color = "black") +
  labs(title = "Store Size Frequency", x = "Store Size", y = "Count")
```




```
# Proportion of days with promo
table(sales_data$promo)

##
##    0    1
## 3494 1506

prop.table(table(sales_data$promo))

##
##      0      1
## 0.6988 0.3012

# Proportion of days with holiday
table(sales_data$holiday)

##
##    0    1
## 4522  478

prop.table(table(sales_data$holiday))

##
##      0      1
## 0.9044 0.0956
```

Step 2:

```r
# Fit Poisson regression model
model_poisson <- glm(sales_count ~ day_of_week + promo + holiday +
store_size,
                     data = sales_data,
                     family = poisson())

# Show model summary
summary(model_poisson)

##
## Call:
## glm(formula = sales_count ~ day_of_week + promo + holiday + store_size,
##     family = poisson(), data = sales_data)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.994849   0.009422  317.86   <2e-16 ***
## day_of_week       0.051115   0.001918   26.65   <2e-16 ***
## promo             0.410843   0.007817   52.55   <2e-16 ***
## holiday          -0.330938   0.014935  -22.16   <2e-16 ***
## store_sizemedium -0.697088   0.008296  -84.03   <2e-16 ***
## store_sizesmall  -1.395564   0.011868 -117.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 25307.2  on 4999  degrees of freedom
## Residual deviance:  5142.7  on 4994  degrees of freedom
## AIC: 26507
##
## Number of Fisher Scoring iterations: 4
```

Intepretation:

Intercept (Estimate = 2.995)

This is the expected log of sales count when all predictors are at their baseline:

Day of week = 0 (usually reference or Monday depending on encoding)

promo = 0 (no promotion)

holiday = 0 (not a holiday)

store_size = baseline category (likely large, since medium and small are listed)

Expected sales:

exp(2.995)≈19.97

So, a large store on the baseline day without promo or holiday is expected to have ~20 sales.

day_of_week (Estimate = 0.051)

Each additional day (assuming encoded as numeric: 0 = Monday, 6 = Sunday) increases log sales by 0.051.

This means sales tend to increase slightly later in the week.

exp(0.051)≈1.052

5.2% increase in sales per day progression through the week.

promo (Estimate = 0.411) Promotions increase expected log sales by 0.411.

exp(0.411)≈1.509 Sales increase by about 51% when a promotion is active. this is the strongest positive predictor in the model.

holiday (Estimate = -0.331)

Holidays reduce expected log sales by 0.331.

exp(−0.331)≈0.718 Sales decrease by about 28% on holidays.

store_size Effects (Reference: Large)

Medium Store:

exp(−0.697)≈0.498 Medium stores have ~50% of the sales of large stores, all else equal.

Small Store:

exp(−1.396)≈0.248 Small stores have only ~25% of the sales of large stores. ## Step 3:

```r
# Check for overdispersion
deviance(model_poisson) / df.residual(model_poisson)
```

```
## [1] 1.029785
```

```r
# Fit quasi-Poisson model
model_quasi <- glm(sales_count ~ day_of_week + promo + holiday + store_size,
                   data = sales_data,
                   family = quasipoisson())

# Fit negative binomial model (if MASS is available)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
model_nb <- glm.nb(sales_count ~ day_of_week + promo + holiday + store_size,
                   data = sales_data)
```

```
## Warning in glm.nb(sales_count ~ day_of_week + promo + holiday +
store_size, :
## alternation limit reached
```

```
# Compare models using AIC
AIC(model_poisson, model_nb)
```

```
##               df      AIC
## model_poisson  6 26506.91
## model_nb       7 26508.11
```

Step 4

```
# Create new data frame for prediction
new_data <- data.frame(
  day_of_week = c(0, 6),
  promo = c(1, 0),
  holiday = c(0, 1),
  store_size = c("medium", "large")
)

# Predict expected sales
predict(model_poisson, newdata = new_data, type = "response")
```

```
##        1        2
## 15.00832 19.50371
```

Intepretation: Medium store, Monday (day_of_week = 0), with promotion, no holiday We got a predicted sale of 15.01 or 15 sales with that date. Meaning with a promotion running, a medium-sized store should sell roughly 15 goods on a typical weekday.

Large store, Sunday (day_of_week = 6), no promotion, holiday We got a predicted sale of 19.50 or 20 sales with that dat. Meaning even on a holiday, a large business should sell about 19.5 goods, however the lack of promotion could negatively impact sales.

Some other insights: Sales are obviously increased by the promotion (Scenario 1 has a promotion, but Scenario 2 does not).

Store size matters: even on a holiday, the larger store (Scenario 2) does well.

Sunday may naturally have better sales because of more foot traffic, even if it is a holiday and there is no marketing.

Step 5:

The Poisson regression model provided reasonable insights into factors influencing store sales. The model fit was decent, though a slight overdispersion was detected, suggesting a quasi-Poisson or negative binomial model might be more appropriate. Among all predictors, promotion had the strongest impact, significantly increasing expected sales. Store size also showed a notable influence, with larger stores experiencing higher sales counts. One limitation of this model is its assumption that the mean equals the variance (in the Poisson model), which often doesn’t hold in real-world sales data due to variability caused by external events or local factors. Future models should also consider interaction effects and time-based trends for improved accuracy.