Hotels

Christina Liang

```
library(tidyverse)
library(infer)
```

```
hotel_bookings <- read.csv("~/R/DIIG/hotel_bookings.csv")
```

First, I made some new variables and did some data cleaning:

New variable for total amount of nights stayed:

```
hotel_bookings <- hotel_bookings %>%
  mutate(total_nights = stays_in_week_nights + stays_in_weekend_nights)
```

Changing the month of arrival into chronologically-ordered levels:

```
hotel_bookings <- hotel_bookings %>%
  mutate(arrival_date_month = factor(arrival_date_month,
                                     levels = c("January", "February", "March", "April", "May",
                                                "June", "July", "August", "September",
                                                "October", "November", "December")))
```

I also changed the is_canceled variable from numeric to categorical, as 0 and 1 represent a booking being cancelled or not.

```
hotel_bookings$is_canceled <- as.factor(hotel_bookings$is_canceled)
```

I created a variable for the total number of guests during the duration of the stay:

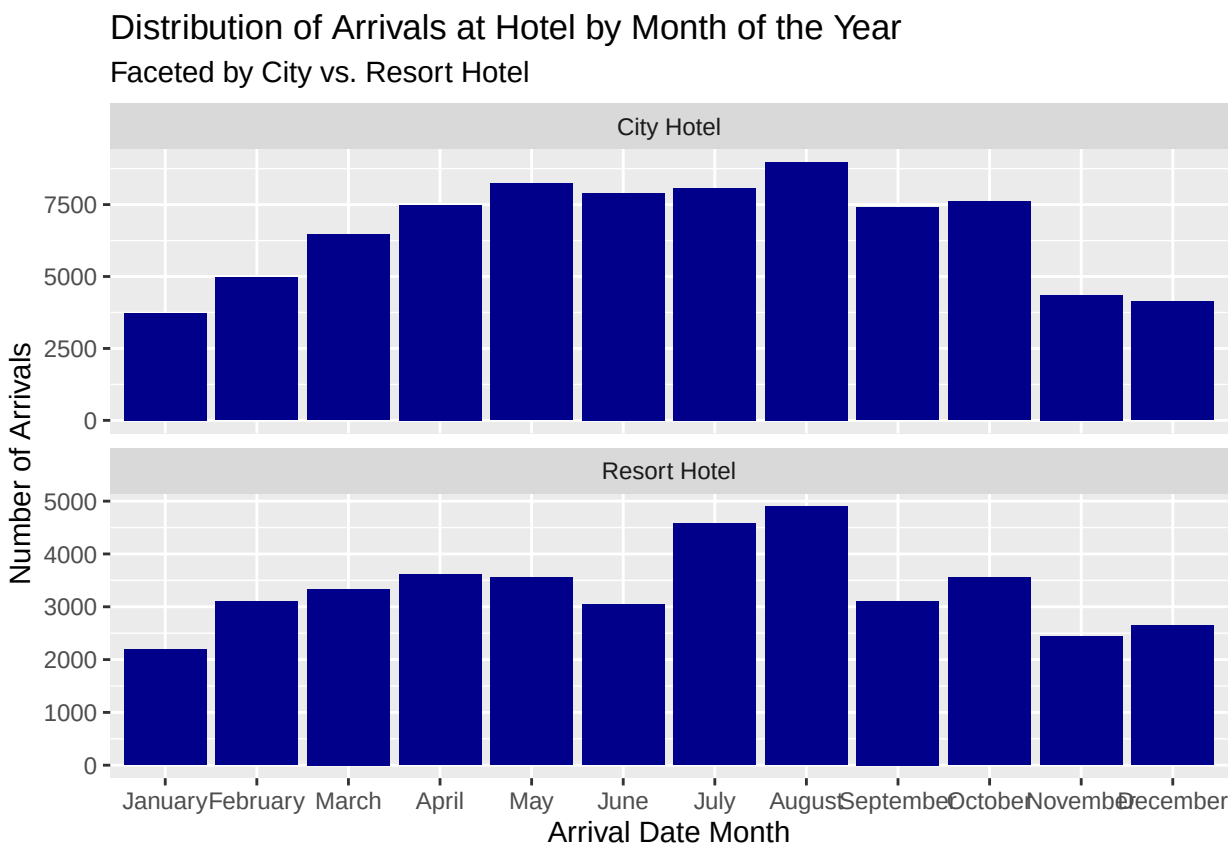
```
hotel_bookings <- hotel_bookings %>%
  mutate(total_guests = adults + children + babies)
```

```
hotel_bookings <- hotel_bookings %>%
  mutate(arrival_season = case_when(arrival_date_month == "December" ~ "Winter",
                                    arrival_date_month == "January" ~ "Winter",
                                    arrival_date_month == "February" ~ "Winter",
                                    arrival_date_month == "September" ~ "Fall",
                                    arrival_date_month == "October" ~ "Fall",
                                    arrival_date_month == "November" ~ "Fall",
                                    arrival_date_month == "March" ~ "Spring",
                                    arrival_date_month == "April" ~ "Spring",
                                    arrival_date_month == "May" ~ "Spring",
                                    arrival_date_month == "June" ~ "Summer",
                                    arrival_date_month =="July" ~ "Summer",
                                    arrival_date_month =="August" ~ "Summer"))
```

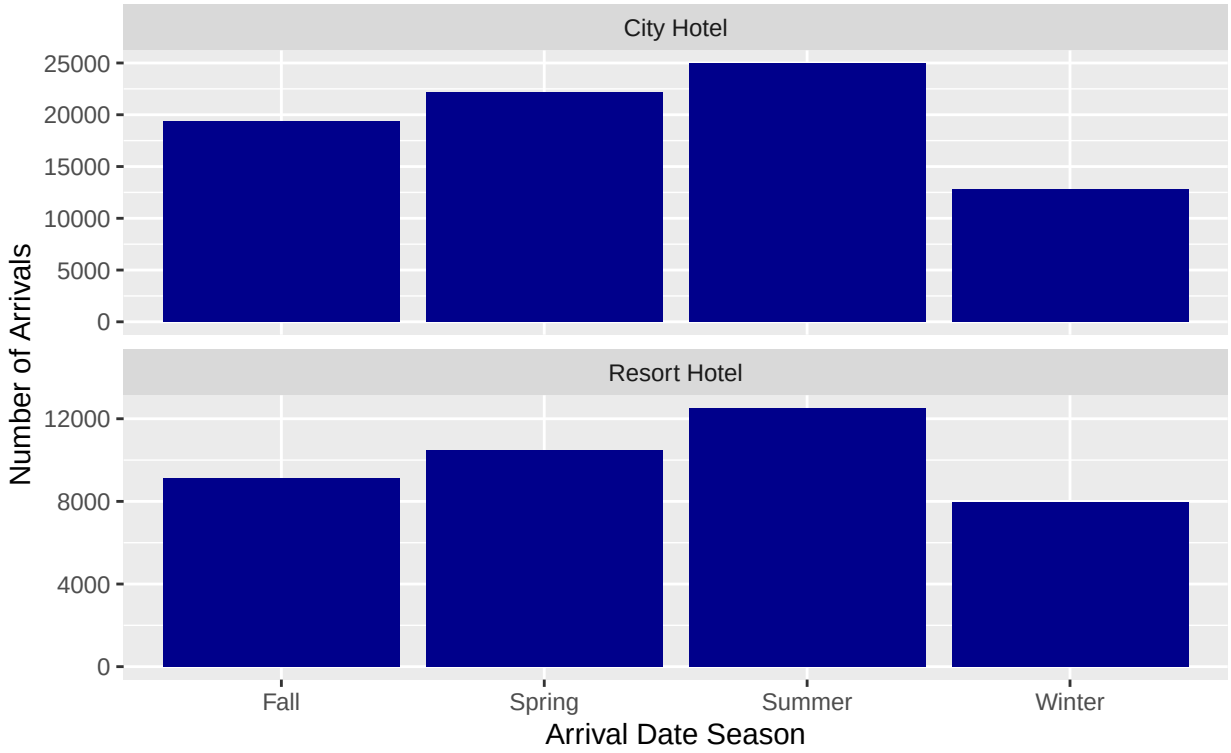
Next, I visualized the distribution of visits to the hotels based on month of the year, to find that there was an increase in volume of arrivals in the warmer months.

```
hotel_bookings %>%
  group_by(hotel, arrival_date_month) %>%
```

```
  ggplot(aes(x = arrival_date_month)) +
  geom_bar(fill = "darkblue") +
  facet_wrap(~ hotel,
             nrow = 2,
             scales = "free_y") +
  labs(title = "Distribution of Arrivals at Hotel by Month of the Year",
       subtitle = "Faceted by City vs. Resort Hotel",
       x = "Arrival Date Month",
       y = "Number of Arrivals")
```

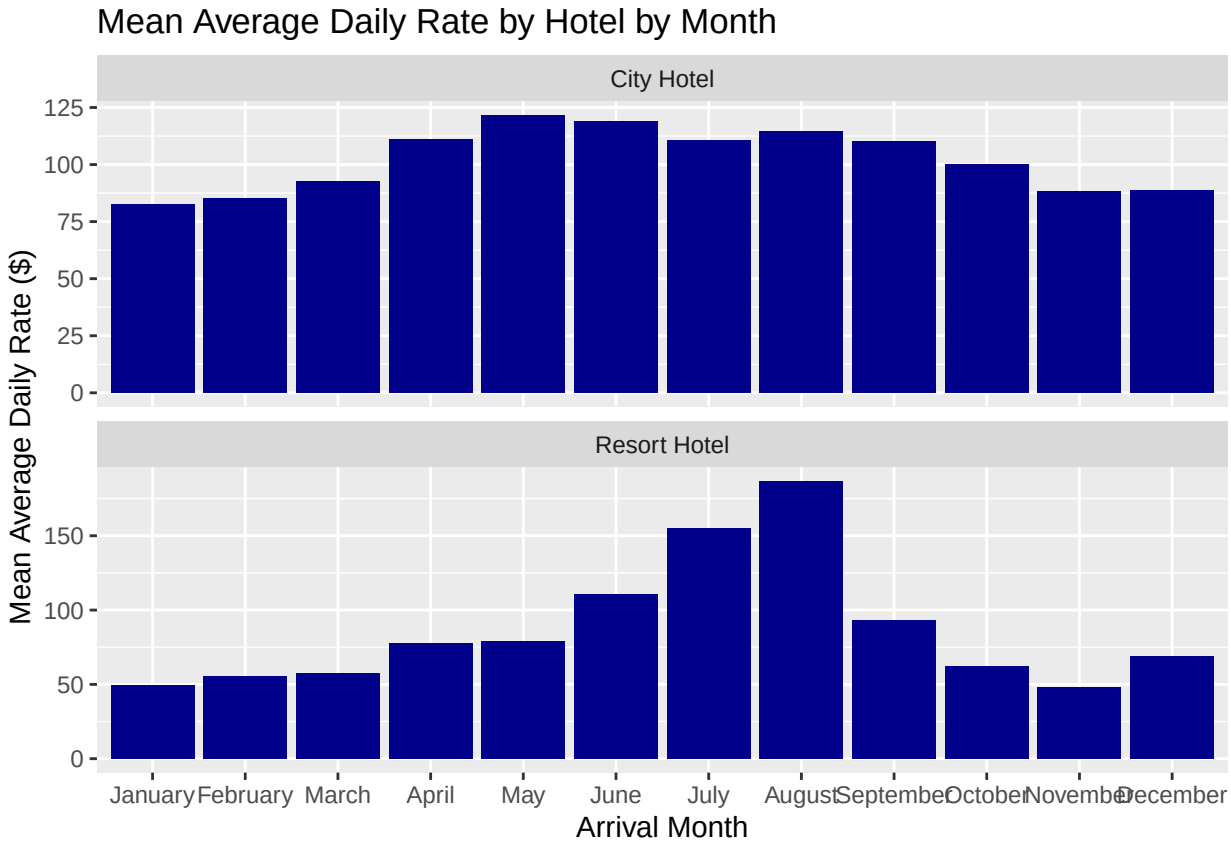



```
hotel_bookings %>%
  group_by(hotel, arrival_season) %>%
  ggplot(aes(x = arrival_season)) +
  geom_bar(fill = "darkblue") +
  facet_wrap(~ hotel,
             nrow = 2,
             scales = "free_y") +
  labs(title = "Distribution of Arrivals at Hotel by Season of the Year",
       subtitle = "Faceted by City vs. Resort Hotel",
       x = "Arrival Date Season",
       y = "Number of Arrivals")
```




```
hotel_bookings %>%
  group_by(hotel, arrival_date_month) %>%
  summarise(meanadr = mean(adr)) %>%
  ggplot(aes(x = arrival_date_month, y = meanadr)) +
  geom_col(fill = "darkblue") +
  facet_wrap(~ hotel, nrow = 2, scales = "free_y") +
  labs(title = "Mean Average Daily Rate by Hotel by Month",
       x = "Arrival Month",
       y = "Mean Average Daily Rate ($)")
```

```
## `summarise()` regrouping output by 'hotel' (override with `.groups` argument)
```



It seems that city hotels are pretty expensive year-round, whereas resort hotels are significantly cheaper in the colder months than in the warmer months.

For this data challenge, I'll mainly be focusing on Resort Hotels, so I filtered the "City Hotels" out of my dataset.

```
resort_bookings <- hotel_bookings %>%
  filter(hotel == "Resort Hotel")
```

Question: What influences the average daily rate at resort hotels?

I'll be looking at the number of adults, children, and babies, the arrival month, the total number of nights stayed, the meal plan, the number of special requests, and the number of purchased car parkings. I'll build the model manually.

```
resort_bookings %>%
  group_by(arrival_season) %>%
  summarise(meanadr = mean(adr))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 2
##   arrival_season meanadr
##   <chr>            <dbl>
## 1 Fall              69.0
## 2 Spring            71.7
## 3 Summer           157.
## 4 Winter            58.2
```

First, I need to figure out whether it is better to use month or season:

```
m_rate_month <- lm(adr ~ arrival_date_month,
                   data = resort_bookings)

glance(m_rate_month)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.566         0.566  40.5     4755.       0    11 -2.05e5 4.10e5 4.10e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
m_rate_season <- lm(adr ~ arrival_season,
                    data = resort_bookings)

glance(m_rate_season)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.464         0.464  45.0    11556.       0     3 -2.09e5 4.19e5 4.19e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Okay apparently it's month but both of them have pretty low adjusted r-squared values and that makes me sad.

I'll also need to figure out whether I want to use total number of guests or the individual number of adults, children, and babies.

```
m_rate_totalguests <- lm(adr ~ total_guests, data = resort_bookings)

glance(m_rate_totalguests)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.125         0.125  57.5     5709.       0     1 -2.19e5 4.38e5 4.38e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
m_rate_indguests <- lm(adr ~ adults + children + babies,
                       data = resort_bookings)

glance(m_rate_indguests)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.160         0.160  56.3     2536.       0     3 -2.18e5 4.37e5 4.37e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Anyways, using the individual guests instead of the overall number of guests is better due to a slightly higher adjusted r-squared value.

Now, I'll start building the bigger model manually:

```
logit_mod1 <- glm(is_canceled ~ arrival_date_month + total_nights,
                  data = resort_bookings, family = "binomial", maxit = 100)
```

```r
logit_mod2 <- glm(is_canceled ~ adults + children + babies + meal,
                  data = resort_bookings, family = "binomial", maxit = 100)

logit_mod2
```

```
##
## Call:  glm(formula = is_canceled ~ adults + children + babies + meal,
##     family = "binomial", data = resort_bookings, maxit = 100)
##
## Coefficients:
##   (Intercept)         adults       children         babies         mealFB
##      -1.85054        0.41463        0.32996       -0.64143        1.36302
##        mealHB         mealSC  mealUndefined
##       0.22745       -2.19245       -0.05317
##
## Degrees of Freedom: 40059 Total (i.e. Null);  40052 Residual
## Null Deviance:       47330
## Residual Deviance: 46310     AIC: 46330
```

```r
logit_mod3 <- glm(is_canceled ~ adr + required_car_parking_spaces + total_of_special_requests,
                  data = resort_bookings, family = "binomial", maxit = 100)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```