Hotels

Christina Liang

```
library(tidyverse)
library(infer)
library(leaps)
library(MASS)
```

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

Data cleaning

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)
```

Then, I did the same for is_repeated_guest, which should also be a categorical variable rather than a numerical variable.

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

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)
```

I also created a new variable for the season during the arrival at the hotel, assigning the months to season.

```
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"))
```

I also created another variable called "kids", which would classify whether or not the guests brought kids. Kids meant either bringing children or bringing babies–only when there were neither children nor babies would the guests be considered having "no kids".

```
hotel_bookings <- hotel_bookings %>%
  mutate(kids = case_when(children > 0 & babies == 0 ~ "Have kids",
                          children == 0 & babies == 0 ~ "No kids",
                          babies > 0 & children == 0 ~ "Have kids"))
```

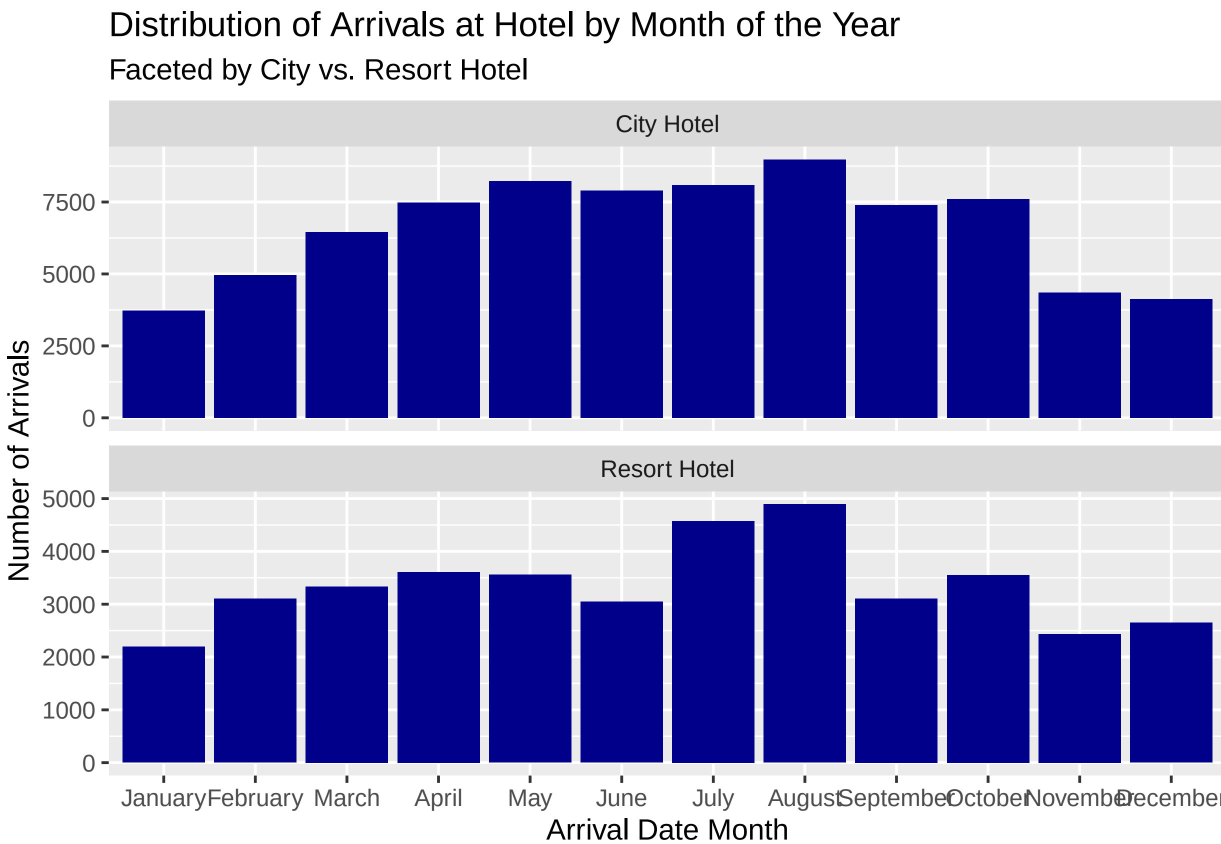
Lastly, I created a variable that accounted for the percentage of previous bookings cancelled as a proportion of total previous bookings (previous bookings cancelled/(previous bookings canceled + previous bookings not cancelled))

```
hotel_bookings <- hotel_bookings %>%
  mutate(prop_cancelled = previous_cancellations/sum(previous_cancellations
                                                     + previous_bookings_not_canceled) * 100)
```

Visualizations

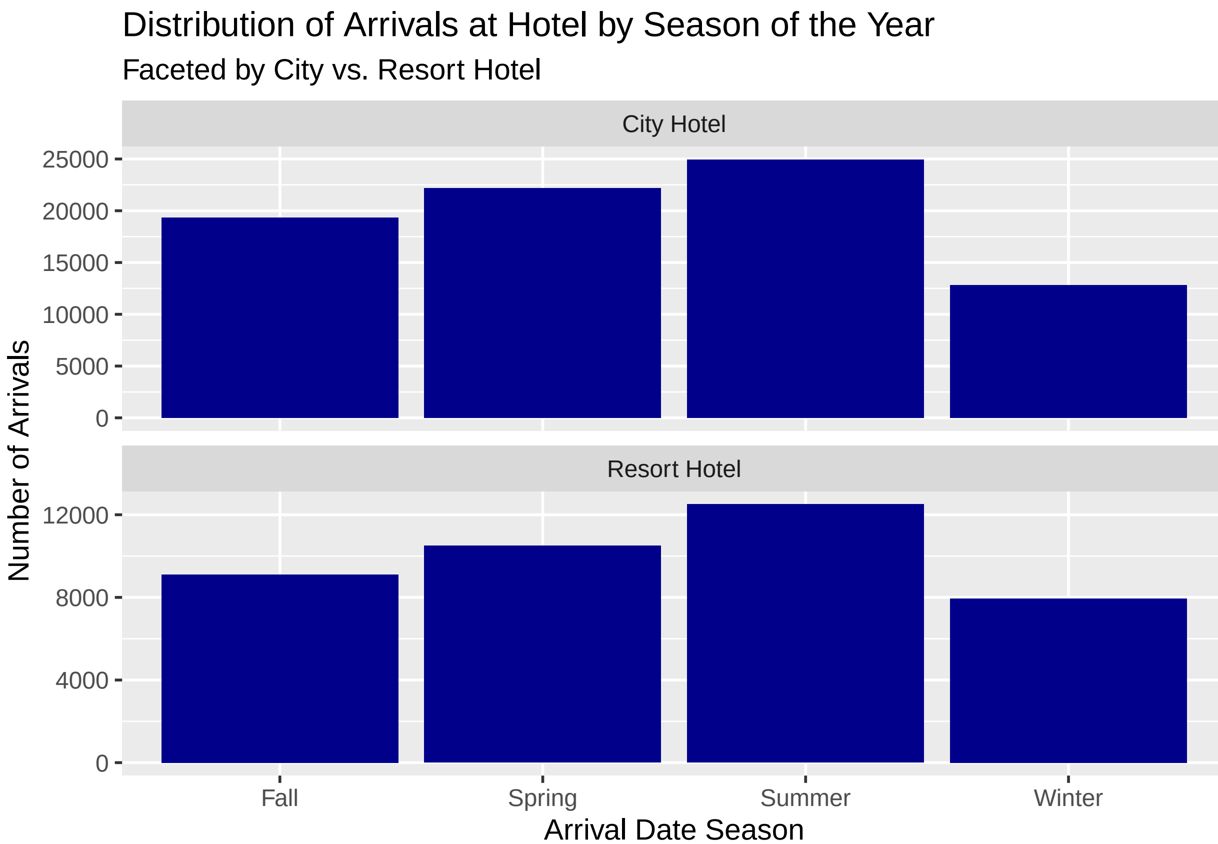
Next, I visualized the distribution of visits to the hotels based on month of the year, to find that there was an increase in volume of arrivals in the warmer months.

```
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")
```



Likewise, I visualized the distribution of arrivals at the hotels during the different seasons.

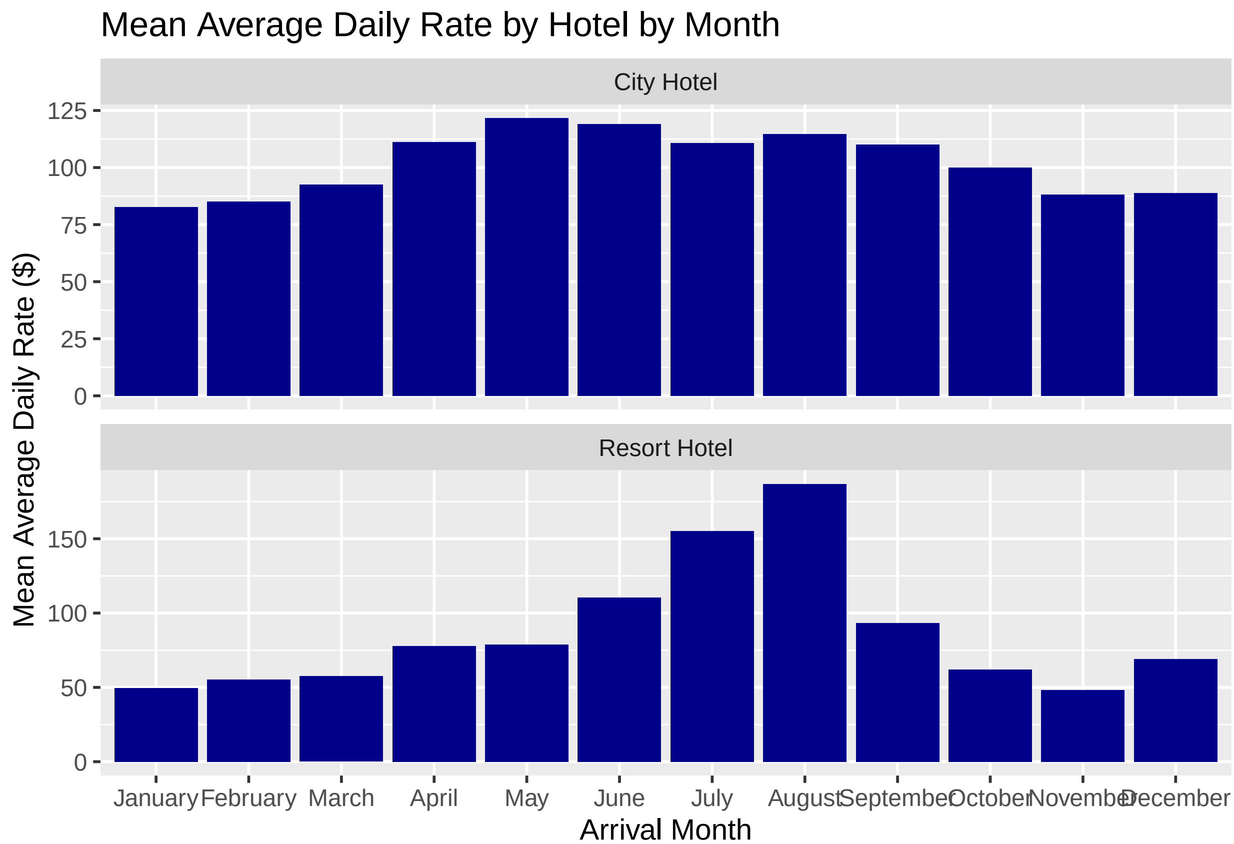
```
hotel_bookings %>%
  group_by(hotel, arrival_season) %>%
  ggplot(aes(x = arrival_season)) +
  geom_bar(fill = "darkblue") +
  facet_wrap(~ hotel,
             nrow = 2,
             scales = "free_y") +
  labs(title = "Distribution of Arrivals at Hotel by Season of the Year",
       subtitle = "Faceted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



Looking at average daily rate next, I visualized the distribution of average daily rate depending on the month of arrival at the hotels.

```
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)
```

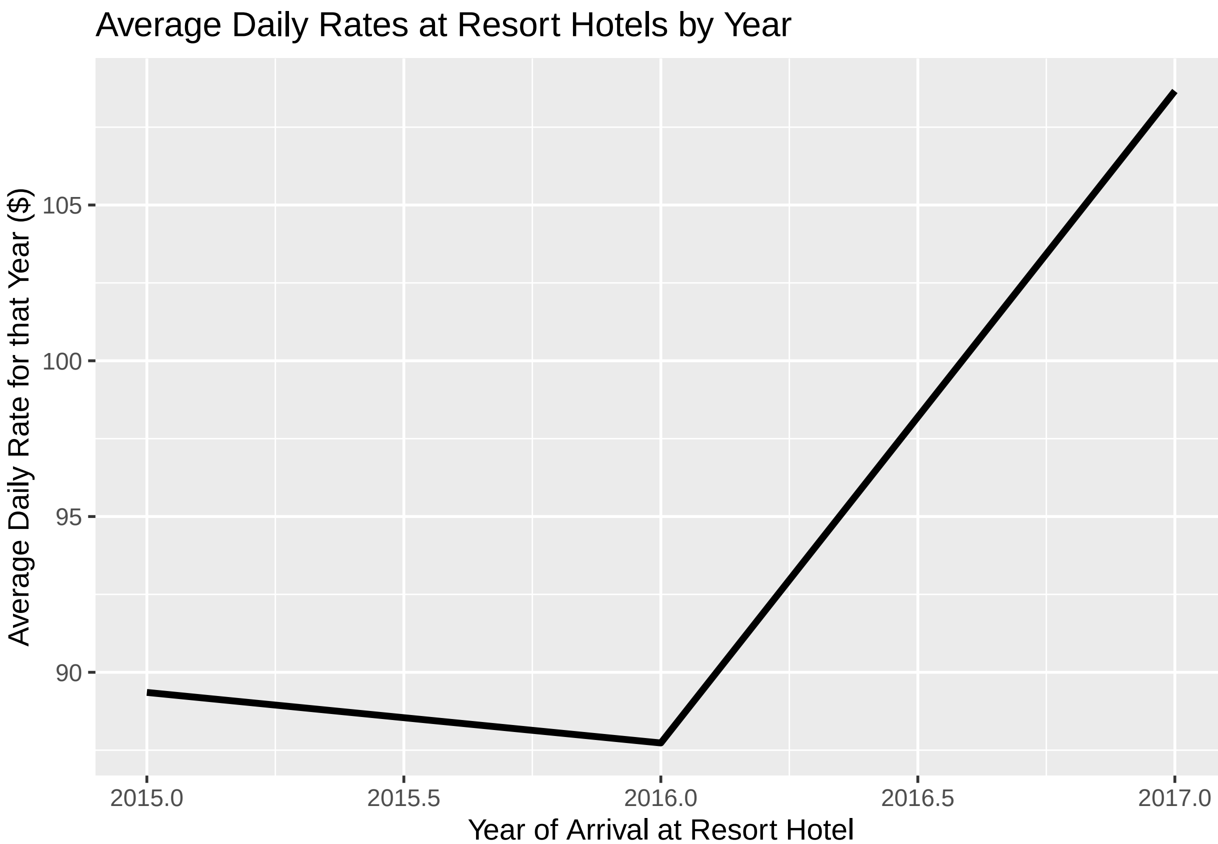


It seems that city hotels are pretty expensive year-round, whereas resort hotels are significantly cheaper in the colder months than in the warmer months.

I also want to see how the average daily rate at the hotels have changed over time.

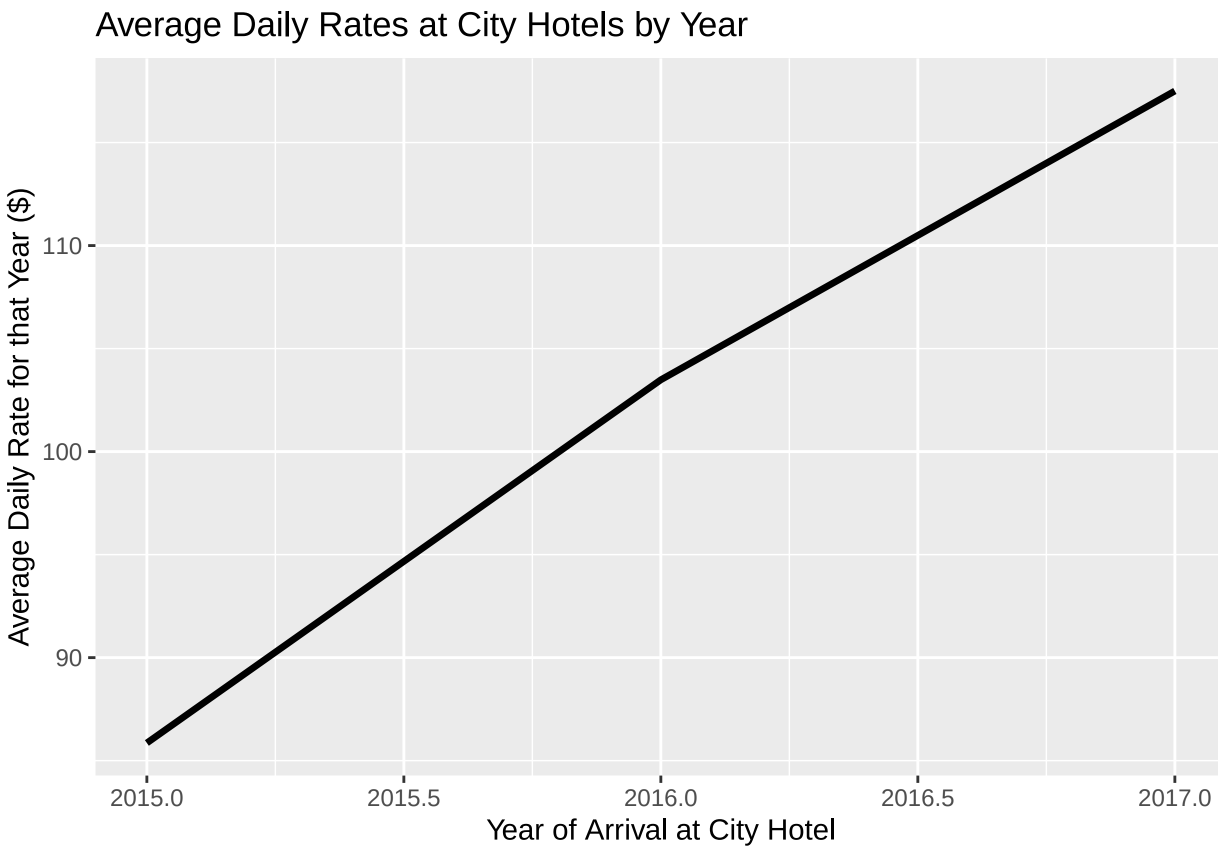
```
hotel_bookings %>%
  filter(hotel == "Resort Hotel") %>%
  group_by(arrival_date_year) %>%
  summarize(meanadryear = mean(adr)) %>%
  ggplot(mapping = aes(x = arrival_date_year, y = meanadryear)) +
  geom_line(size = 1.2) +
  labs(x = "Year of Arrival at Resort Hotel", y = "Average Daily Rate for that Year ($)",
       title = "Average Daily Rates at Resort Hotels by Year")
```

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

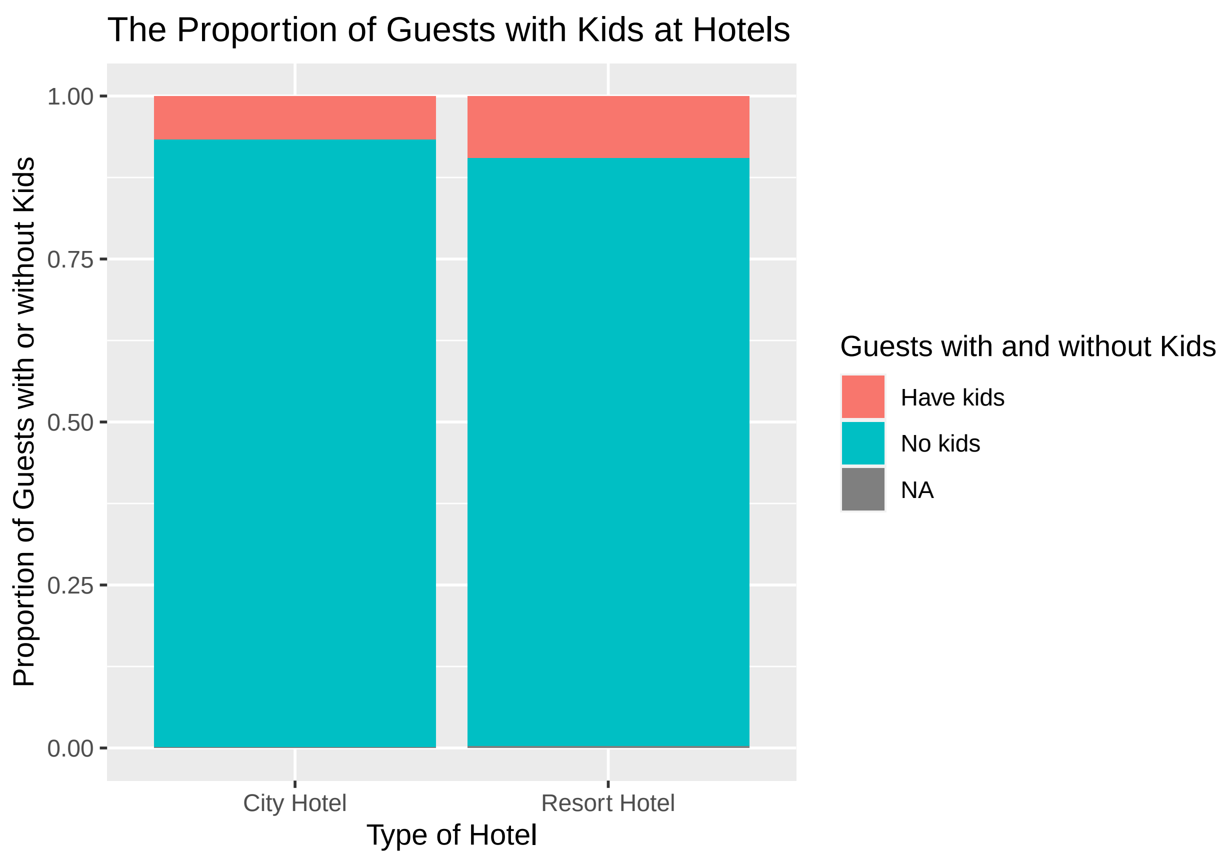



```r
hotel_bookings %>%
  filter(hotel == "City Hotel") %>%
  group_by(arrival_date_year) %>%
  summarize(meanadryear = mean(adr)) %>%
  ggplot(mapping = aes(x = arrival_date_year, y = meanadryear)) +
  geom_line(size = 1.2) +
  labs(x = "Year of Arrival at City Hotel", y = "Average Daily Rate for that Year ($)",
       title = "Average Daily Rates at City Hotels by Year")
```

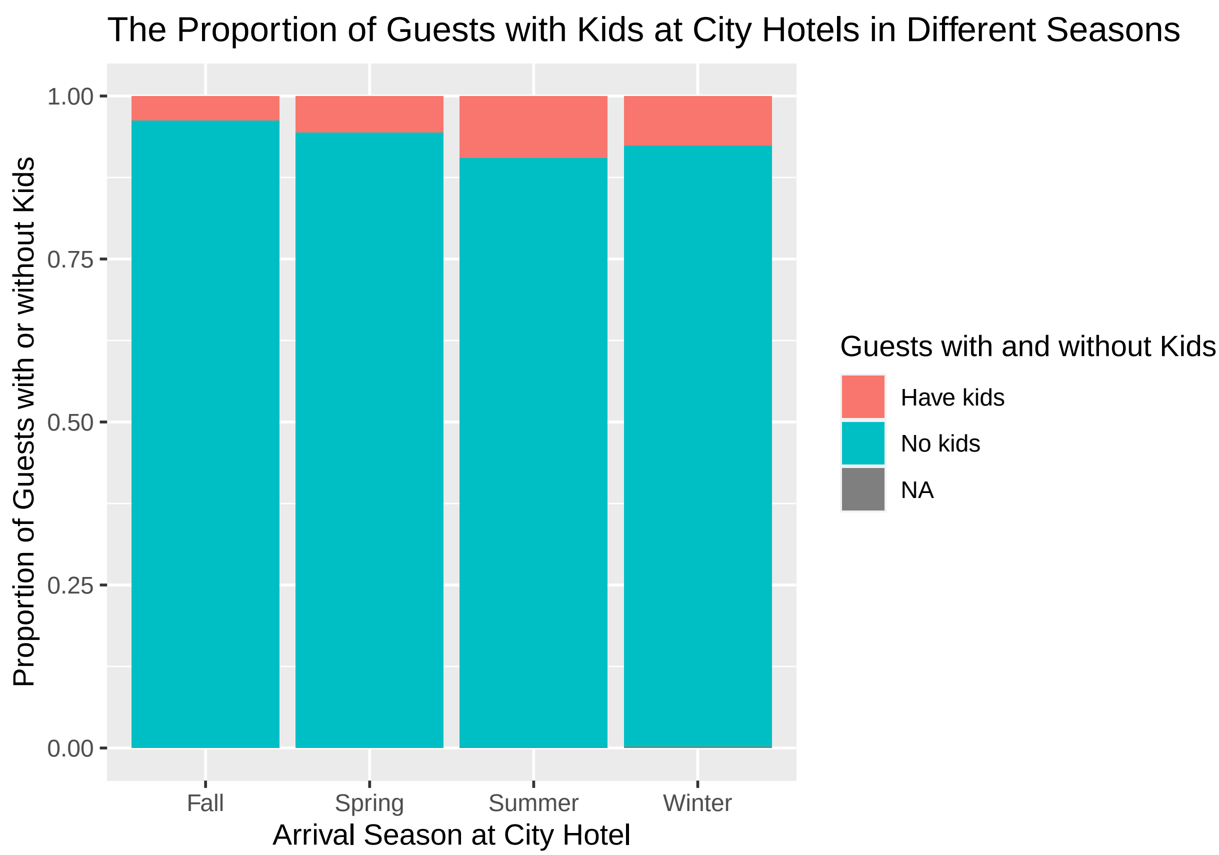
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



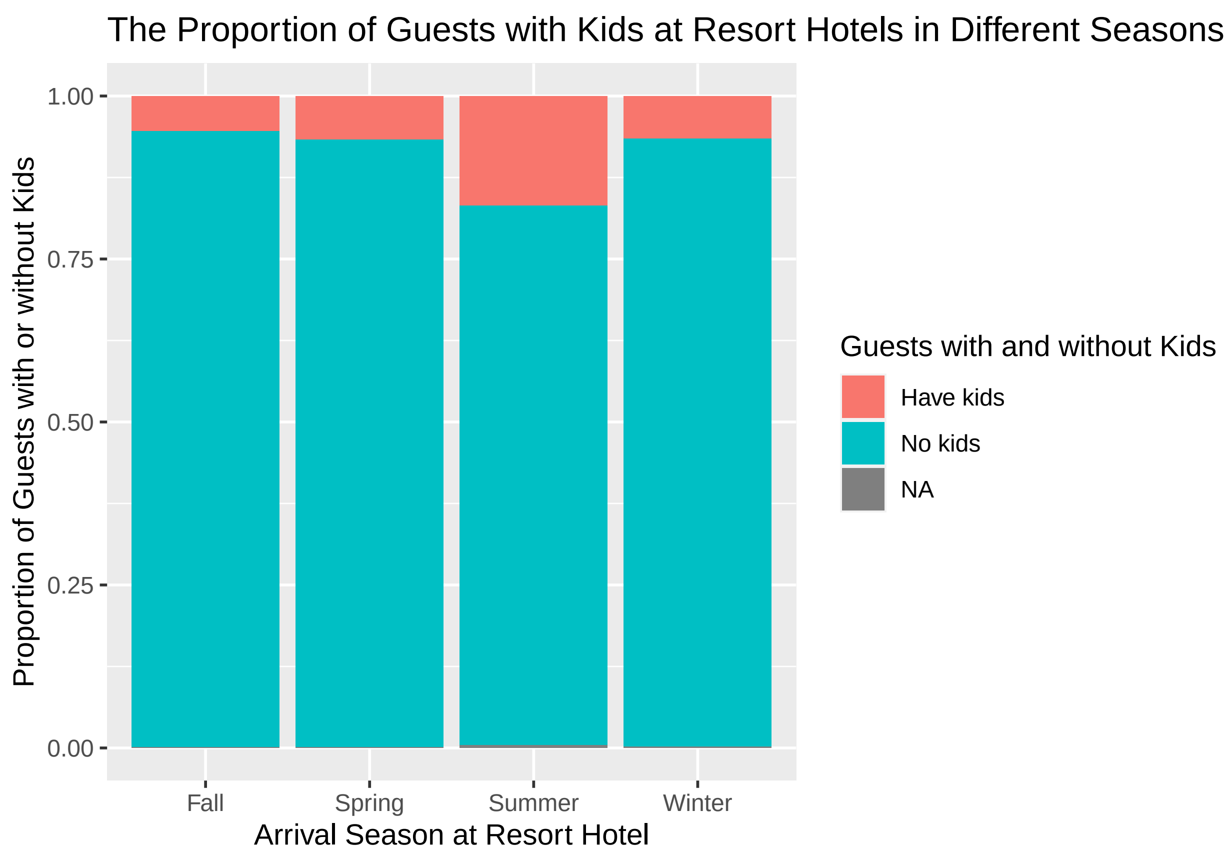

```
hotel_bookings %>%
  group_by(hotel) %>%
  ggplot(mapping = aes(x = hotel,
                       fill = kids)) +
  geom_bar(position = "fill") +
  labs(x = "Type of Hotel",
       y = "Proportion of Guests with or without Kids", fill = "Guests with and without Kids",
       title = "The Proportion of Guests with Kids at Hotels")
```




```r
hotel_bookings %>%
  filter(hotel == "City Hotel") %>%
  group_by(arrival_season) %>%
  ggplot(mapping = aes(x = arrival_season,
                       fill = kids)) +
  geom_bar(position = "fill") +
  labs(x = "Arrival Season at City Hotel",
       y = "Proportion of Guests with or without Kids", fill = "Guests with and without Kids",
       title = "The Proportion of Guests with Kids at City Hotels in Different Seasons")
```




```r
hotel_bookings %>%
  filter(hotel == "Resort Hotel") %>%
  group_by(arrival_season) %>%
  ggplot(mapping = aes(x = arrival_season,
                       fill = kids)) +
  geom_bar(position = "fill") +
  labs(x = "Arrival Season at Resort Hotel",
       y = "Proportion of Guests with or without Kids", fill = "Guests with and without Kids",
       title = "The Proportion of Guests with Kids at Resort Hotels in Different Seasons")
```



Resort Hotels

For this data challenge, I'll mainly be focusing on Resort Hotels, so I filtered the "City Hotels" out of my dataset. Resort Hotels piqued my interests because of the vacation- and family-oriented aspect. Additionally, the huge disparity in amount of arrivals and cost of a resort hotel between cold weather months and warm weather months I think is worth investigating. Practically, that disparity makes sense because families tend to take resort-type vacations in the summer.

```
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, because these variables are the most practical ones of the included variables when considering the price of a hotel during the booking stage. I'll build the model manually at first, and then use a stepwise backward and forward elimination to eliminate unnecessary predictors from the model. Afterwards, the model should follow the laws of Occam's Razor (the simplest model that explains the most).

There seems to be a sizeable disparity between mean average daily rates by season for the resort hotels, alluding to the practical relevance of season/month on daily rates at the resort hotel.

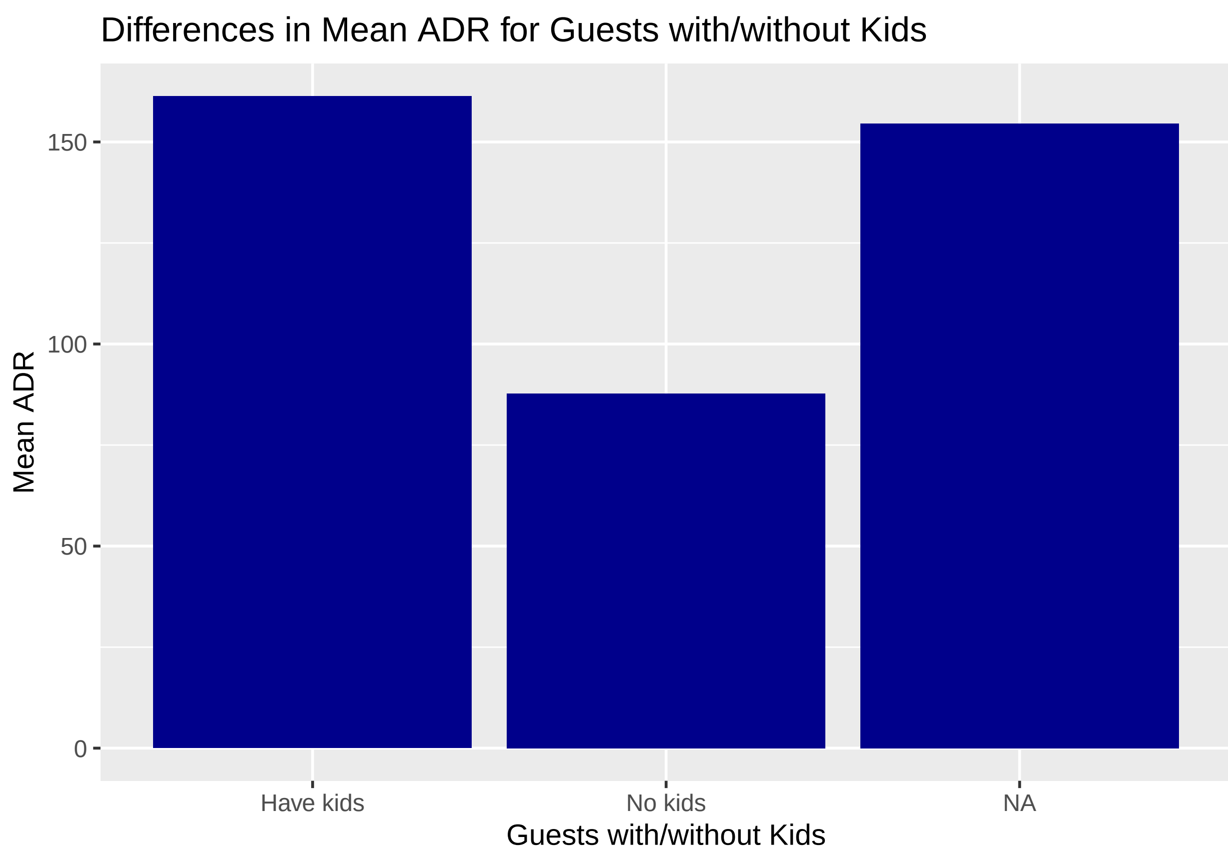
```
resort_bookings %>%
  group_by(arrival_season) %>%
  summarise(meanadr = mean(adr))
```

```
## `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
```

```
resort_bookings %>%
  group_by(kids) %>%
  summarise(meanadr = mean(adr)) %>%
  ggplot(mapping = aes(x = kids,
                       y = meanadr)) +
  geom_col(fill = "darkblue") +
  labs(title = "Differences in Mean ADR for Guests with/without Kids",
       x = "Guests with/without Kids",
       y = "Mean ADR")
```

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



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>
```

According to the r-squared values, arrival month explains more of the differences in average daily rate. Unfortunately, that means there will be twelve levels of that variable, rather than four levels.

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>
```

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:

(January is the reference level for the arrival_date_month variable)

```
m_1 <- lm(adr ~ arrival_date_month + adults,
          data = resort_bookings)

glance(m_1)
```

```
## # 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     0.575         0.575  40.0     4520.       0    12 -2.05e5 4.09e5 4.09e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
tidy(m_1)
```

```
## # A tibble: 13 x 5
##    term                        estimate std.error statistic   p.value
##    <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                    35.5      0.982     36.1  9.07e-281
##  2 arrival_date_monthFebruary      4.50     1.12       4.03 5.67e-  5
##  3 arrival_date_monthMarch         7.37     1.10       6.70 2.16e- 11
##  4 arrival_date_monthApril        27.2      1.09      25.1  1.97e-137
##  5 arrival_date_monthMay          27.7      1.09      25.5  5.90e-142
##  6 arrival_date_monthJune         58.8      1.12      52.4  0.
##  7 arrival_date_monthJuly        103.       1.05      98.4  0.
##  8 arrival_date_monthAugust      134.       1.03     130.   0.
##  9 arrival_date_monthSeptember    41.2      1.12      36.8  1.39e-291
## 10 arrival_date_monthOctober      11.0      1.09      10.1  7.80e- 24
## 11 arrival_date_monthNovember     -1.56     1.18      -1.32 1.87e-  1
## 12 arrival_date_monthDecember     18.3      1.16      15.9  1.96e- 56
## 13 adults                          8.42     0.291     29.0  1.54e-182
```

Slight increase –> 0.575 in adj. r. squared with adults, without kids

```r
m_2 <- lm(adr ~ arrival_date_month + adults + children,
          data = resort_bookings)

glance(m_2)
```

```
## # 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     0.629         0.629  37.4     5232.       0    13 -2.02e5 4.04e5 4.04e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
tidy(m_2)
```

```
## # A tibble: 14 x 5
##    term                        estimate std.error statistic   p.value
##    <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                   35.2       0.918    38.3   5.43e-316
##  2 arrival_date_monthFebruary     3.67      1.04      3.51  4.49e-  4
##  3 arrival_date_monthMarch        7.03      1.03      6.83  8.55e- 12
##  4 arrival_date_monthApril       26.2       1.01     25.9   3.66e-146
##  5 arrival_date_monthMay         26.7       1.02     26.3   5.20e-151
##  6 arrival_date_monthJune        56.0       1.05     53.3   0.
##  7 arrival_date_monthJuly        97.2       0.979    99.3   0.
##  8 arrival_date_monthAugust     128.        0.970   132.    0.
##  9 arrival_date_monthSeptember   41.0       1.05     39.2   0.
## 10 arrival_date_monthOctober     10.8       1.02     10.6   3.24e- 26
## 11 arrival_date_monthNovember    -0.873     1.10     -0.793 4.28e-  1
## 12 arrival_date_monthDecember    17.6       1.08     16.3   1.25e- 59
## 13 adults                         7.34      0.272    27.0   5.44e-159
## 14 children                      32.6       0.426    76.5   0.
```

Significant increase in r-squared –> 0.629.

```r
m_3 <- lm(adr ~ arrival_date_month + adults + children + babies,
          data = resort_bookings)
```

```r
glance(m_3)
```

```
## # 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     0.630         0.629  37.4     4861.       0    14 -2.02e5 4.04e5 4.04e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
tidy(m_3)
```

```
## # A tibble: 15 x 5
##    term                        estimate std.error statistic   p.value
##    <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                   35.2       0.918    38.3   1.03e-315
##  2 arrival_date_monthFebruary     3.66      1.04      3.51  4.51e-  4
##  3 arrival_date_monthMarch        7.03      1.03      6.84  8.23e- 12
##  4 arrival_date_monthApril       26.2       1.01     25.9   2.92e-146
##  5 arrival_date_monthMay         26.7       1.02     26.2   1.36e-150
##  6 arrival_date_monthJune        55.9       1.05     53.2   0.
##  7 arrival_date_monthJuly        97.1       0.979    99.2   0.
##  8 arrival_date_monthAugust     128.        0.970   132.    0.
##  9 arrival_date_monthSeptember   41.0       1.05     39.2   0.
## 10 arrival_date_monthOctober     10.8       1.02     10.6   2.99e- 26
## 11 arrival_date_monthNovember    -0.880     1.10     -0.799 4.24e-  1
## 12 arrival_date_monthDecember    17.6       1.08     16.3   2.21e- 59
## 13 adults                         7.32      0.272    26.9   2.69e-158
## 14 children                      32.6       0.426    76.5   0.
## 15 babies                         6.22      1.57      3.95  7.70e-  5
```

Very insignificant increase in r-squared with babies.

```r
m_4 <- lm(adr ~ arrival_date_month + adults + children + babies + meal,
          data = resort_bookings)

glance(m_4)
```

```
## # 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     0.655         0.655  36.1     4232.       0    18 -2.00e5 4.01e5 4.01e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
tidy(m_4)
```

```
## # A tibble: 19 x 5
##    term                        estimate std.error statistic   p.value
##    <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                   32.8       0.888    37.0   3.19e-294
##  2 arrival_date_monthFebruary     1.95      1.01      1.94  5.25e-  2
##  3 arrival_date_monthMarch        5.95      0.992     5.99  2.07e-  9
##  4 arrival_date_monthApril       23.4       0.980    23.9   5.38e-125
##  5 arrival_date_monthMay         26.1       0.983    26.5   6.87e-154
##  6 arrival_date_monthJune        55.1       1.01     54.3   0.
##  7 arrival_date_monthJuly        95.7       0.947   101.    0.
##  8 arrival_date_monthAugust     126.        0.940   134.    0.
##  9 arrival_date_monthSeptember   40.2       1.01     39.7   0.
```

```
## 10 arrival_date_monthOctober     11.0       0.983    11.2   6.93e- 29
## 11 arrival_date_monthNovember    -0.859     1.06     -0.808 4.19e-  1
## 12 arrival_date_monthDecember    14.5       1.05     13.9   1.57e- 43
## 13 adults                         6.51      0.263    24.8   2.48e-134
## 14 children                      32.8       0.411    79.7   0.
## 15 babies                         4.42      1.52      2.91  3.59e-  3
## 16 mealFB                        20.5       1.34     15.3   6.16e- 53
## 17 mealHB                        20.7       0.457    45.3   0.
## 18 mealSC                       -71.8       3.90    -18.4   1.89e- 75
## 19 mealUndefined                 26.9       1.09     24.6   8.59e-133
```

Tiny increase in r-squared with meal.

```r
m_5 <- lm(adr ~ arrival_date_month + adults + children + babies + meal + total_nights,
          data = resort_bookings)

glance(m_5)
```

```
## # 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     0.659         0.659  35.9     4080.       0    19 -2.00e5 4.01e5 4.01e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
tidy(m_5)
```

```
## # A tibble: 20 x 5
##    term                         estimate std.error statistic   p.value
##    <chr>                           <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                   35.4       0.891    39.8    0.
##  2 arrival_date_monthFebruary     2.05      1.00      2.04   4.10e-  2
##  3 arrival_date_monthMarch        7.29      0.989     7.38   1.66e- 13
##  4 arrival_date_monthApril       24.4       0.976    25.1    2.00e-137
##  5 arrival_date_monthMay         27.6       0.980    28.1    2.50e-172
##  6 arrival_date_monthJune        57.9       1.02     56.9    0.
##  7 arrival_date_monthJuly        98.3       0.949   104.     0.
##  8 arrival_date_monthAugust     128.        0.941   136.     0.
##  9 arrival_date_monthSeptember   42.5       1.01     42.0    0.
## 10 arrival_date_monthOctober     12.1       0.979    12.4    5.03e- 35
## 11 arrival_date_monthNovember    -0.0759    1.06     -0.0718 9.43e-  1
## 12 arrival_date_monthDecember    14.7       1.04     14.1    2.87e- 45
## 13 adults                         6.88      0.262    26.3    8.27e-151
## 14 children                      32.8       0.409    80.2    0.
## 15 babies                         4.51      1.51      2.99   2.80e-  3
## 16 mealFB                        20.7       1.33     15.6    2.00e- 54
## 17 mealHB                        22.4       0.461    48.5    0.
## 18 mealSC                       -68.2       3.88    -17.6    6.87e- 69
## 19 mealUndefined                 28.0       1.09     25.7    1.97e-144
## 20 total_nights                  -1.20      0.0557  -21.5    1.34e-101
```

Very small in r-squared with total_nights. The coefficient for total_nights is negative, indicating that holding all other factors constant, for each additional night of the stay, we expect a slightly over $1 discount in the average daily rate. This decrease in average daily rate makes sense, because usually a longer stay warrants an additional stay discount.

```r
m_6 <- lm(adr ~ arrival_date_month + adults + children + babies + meal + total_nights +
            total_of_special_requests,
```

```
         data = resort_bookings)

glance(m_6)
```

```
## # 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     0.665         0.665  35.6     3974.       0    20 -2.00e5 4.00e5 4.00e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
tidy(m_6)
```

```
## # A tibble: 21 x 5
##    term                        estimate std.error statistic   p.value
##    <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                    32.6      0.890     36.7  1.48e-289
##  2 arrival_date_monthFebruary      2.23     0.994      2.25 2.47e-  2
##  3 arrival_date_monthMarch         8.17     0.981      8.32 8.70e- 17
##  4 arrival_date_monthApril        25.1      0.968     25.9  1.17e-146
##  5 arrival_date_monthMay          28.1      0.972     28.9  2.02e-181
##  6 arrival_date_monthJune         57.7      1.01      57.2  0.
##  7 arrival_date_monthJuly         97.5      0.942    103.   0.
##  8 arrival_date_monthAugust      126.       0.935    135.   0.
##  9 arrival_date_monthSeptember    43.0      1.00      42.8  0.
## 10 arrival_date_monthOctober      12.9      0.972     13.3  2.43e- 40
## # ... with 11 more rows
```

Slightest increase in r-squared with number of special requests.

```
m_7 <- lm(adr ~ arrival_date_month + adults + children + babies + meal + total_nights +
            total_of_special_requests + required_car_parking_spaces,
          data = resort_bookings)

glance(m_7)
```

```
## # 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     0.673         0.672  35.2     3915.       0    21 -1.99e5 3.99e5 3.99e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
tidy(m_7)
```

```
## # A tibble: 22 x 5
##    term                        estimate std.error statistic   p.value
##    <chr>                          <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                    30.1      0.884     34.0  5.05e-250
##  2 arrival_date_monthFebruary      2.76     0.983      2.81 5.03e-  3
##  3 arrival_date_monthMarch         8.40     0.970      8.66 4.93e- 18
##  4 arrival_date_monthApril        25.3      0.957     26.4  1.94e-152
##  5 arrival_date_monthMay          28.3      0.961     29.4  3.11e-188
##  6 arrival_date_monthJune         57.6      0.997     57.7  0.
##  7 arrival_date_monthJuly         97.6      0.932    105.   0.
##  8 arrival_date_monthAugust      127.       0.925    137.   0.
##  9 arrival_date_monthSeptember    43.1      0.993     43.4  0.
## 10 arrival_date_monthOctober      13.0      0.961     13.6  9.40e- 42
```

```
## # ... with 12 more rows
```

Also a slight tiny increase in r-squared when car parking spaces are considered.

Because no coefficient in the model changes drastically when another is added, I can assume that there is not too much multicollinearity between the predictors and move forward without too much care for interaction variables.

I'm going to do backwards and forwards (both directions) elimination with multivariate regression to see which predictors most influences average daily rate. This stepwise elimination will remove excess variables from the model.

```r
step.model <- stepAIC(m_7, direction = "both",
                      trace = FALSE)
summary(step.model)
```

```
##
## Call:
## lm(formula = adr ~ arrival_date_month + adults + children + meal +
##     total_nights + total_of_special_requests + required_car_parking_spaces,
##     data = resort_bookings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -412.62  -17.20   -2.39   15.66  353.40
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  30.09395    0.88428  34.032  < 2e-16 ***
## arrival_date_monthFebruary    2.75751    0.98323   2.805  0.00504 **
## arrival_date_monthMarch       8.39711    0.97009   8.656  < 2e-16 ***
## arrival_date_monthApril      25.27508    0.95693  26.413  < 2e-16 ***
## arrival_date_monthMay        28.26713    0.96121  29.408  < 2e-16 ***
## arrival_date_monthJune       57.56562    0.99735  57.719  < 2e-16 ***
## arrival_date_monthJuly       97.58704    0.93150 104.764  < 2e-16 ***
## arrival_date_monthAugust    126.87494    0.92480 137.191  < 2e-16 ***
## arrival_date_monthSeptember  43.09500    0.99266  43.414  < 2e-16 ***
## arrival_date_monthOctober    13.01692    0.96063  13.550  < 2e-16 ***
## arrival_date_monthNovember    0.21174    1.03691   0.204  0.83820
## arrival_date_monthDecember   14.50766    1.01936  14.232  < 2e-16 ***
## adults                        6.33478    0.25728  24.622  < 2e-16 ***
## children                     32.29981    0.40144  80.460  < 2e-16 ***
## mealFB                       24.39468    1.30634  18.674  < 2e-16 ***
## mealHB                       23.26121    0.45249  51.407  < 2e-16 ***
## mealSC                      -67.19850    3.80299 -17.670  < 2e-16 ***
## mealUndefined                32.03642    1.07260  29.868  < 2e-16 ***
## total_nights                 -1.08184    0.05503 -19.660  < 2e-16 ***
## total_of_special_requests     5.35931    0.22206  24.135  < 2e-16 ***
## required_car_parking_spaces  15.32734    0.50688  30.239  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.17 on 40039 degrees of freedom
## Multiple R-squared:  0.6725, Adjusted R-squared:  0.6723
## F-statistic:  4111 on 20 and 40039 DF,  p-value: < 2.2e-16
```

The model kicked out babies, but kept all other predictors. The model has an adjusted r-squared of 0.6723, which is a pretty good r-squared value, signifying that approximately 67% of the variability in average daily rate at resort hotels can be explained by the model with the above predictors. Holding all other factors constant, the model predicts that the average daily rate of a resort hotel will be $126.87 more expensive in August than in January.

Cancellations at Resort Hotels

For resort hotels, I'd also like to investigate the likelihood of a booking being cancelled, specially based on the guest's previous behavior and caprice rather circumstantial or financial factors (i.e. price of hotel, how many kids the guests are bringing). So, I'll be looking at whether or not the guest is a repeated guest, if they've previously cancelled bookings, how many booking changes he or she made, and how many days he or she had to stay in the waiting list, which are variabled pertaining to my emphasis on behavior/loyalty/emotion. To build the desired model, I'll be following the laws of Occam's Razor and trying to find the best model with the lowest AIC and BIC values.

I started building my logistic regression model manually with the binary outcome variable "is_canceled":

```
log1 <- glm(is_canceled ~ is_repeated_guest,
                 data = resort_bookings, family = "binomial")

log1
```

```
##
## Call:  glm(formula = is_canceled ~ is_repeated_guest, family = "binomial",
##     data = resort_bookings)
##
## Coefficients:
##        (Intercept)  is_repeated_guest1
##            -0.9069             -1.8023
##
## Degrees of Freedom: 40059 Total (i.e. Null);  40058 Residual
## Null Deviance:       47330
## Residual Deviance: 46770     AIC: 46770
```

```
AIC(log1)
```

```
## [1] 46774.21
```

```
BIC(log1)
```

```
## [1] 46791.41
```

```
log2 <- glm(is_canceled ~ is_repeated_guest + previous_bookings_not_canceled,
                 data = resort_bookings, family = "binomial")

log2
```

```
##
## Call:  glm(formula = is_canceled ~ is_repeated_guest + previous_bookings_not_canceled,
##     family = "binomial", data = resort_bookings)
##
## Coefficients:
##                    (Intercept)              is_repeated_guest1
##                        -0.8951                         -1.2385
## previous_bookings_not_canceled
##                        -0.4168
##
```

```
## Degrees of Freedom: 40059 Total (i.e. Null);  40057 Residual
## Null Deviance:       47330
## Residual Deviance: 46620     AIC: 46630
```

```r
AIC(log2)
```

```
## [1] 46626.57
```

```r
BIC(log2)
```

```
## [1] 46652.36
```

The AIC and BIC values decreased when previous_bookings_not_canceled was added to the logistic regression model, so I can proceed to add more variables.

```r
log3 <- glm(is_canceled ~ is_repeated_guest + previous_bookings_not_canceled +
              booking_changes,
                data = resort_bookings, family = "binomial")

log3
```

```
##
## Call:  glm(formula = is_canceled ~ is_repeated_guest + previous_bookings_not_canceled +
##     booking_changes, family = "binomial", data = resort_bookings)
##
## Coefficients:
##                    (Intercept)              is_repeated_guest1
##                        -0.7749                         -1.2553
## previous_bookings_not_canceled                 booking_changes
##                        -0.4147                         -0.5223
##
## Degrees of Freedom: 40059 Total (i.e. Null);  40056 Residual
## Null Deviance:       47330
## Residual Deviance: 45940     AIC: 45950
```

```r
AIC(log3)
```

```
## [1] 45951.85
```

```r
BIC(log3)
```

```
## [1] 45986.25
```

The AIC and BIC values decreased when booking_changes was added to the logistic regression model, so I can proceed to add more variables.

```r
log4 <- glm(is_canceled ~ is_repeated_guest + previous_bookings_not_canceled +
              booking_changes + days_in_waiting_list,
                data = resort_bookings, family = "binomial")

tidy(log4)
```

```
## # A tibble: 5 x 5
##   term                           estimate std.error statistic   p.value
##   <chr>                             <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)                      -0.769    0.0122     -63.3  0.
## 2 is_repeated_guest1               -1.26     0.110      -11.4  3.05e- 30
## 3 previous_bookings_not_canceled   -0.416    0.0479      -8.69 3.67e- 18
## 4 booking_changes                  -0.521    0.0230     -22.7  1.06e-113
```

```
## 5 days_in_waiting_list            -0.0218   0.00361     -6.04 1.51e-   9
```

```r
AIC(log4)
```

```
## [1] 45878.74
```

```r
BIC(log4)
```

```
## [1] 45921.73
```

The AIC and BIC values decreased very slightly when days_in_waiting_list was added to the logistic regression model, so I'll keep it in the model and call this model the final model.

I've also created a model with the variable previous_cancellations included; however, this variable yields perfect separation, which is not desired. Practically, I believe that with previous_bookings_not_canceled in the model, this model that includes previous_cancellations is not necessary. Regardless, previous_cancellations reduces the AIC and BIC values significantly. This model is an alternative model to the previous model.

```r
logit_mod <- glm(is_canceled ~ is_repeated_guest + previous_bookings_not_canceled + previous_cancellati
                   booking_changes + days_in_waiting_list,
                  data = resort_bookings, family = "binomial", maxit = 100)
```

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

```r
tidy(logit_mod)
```

```
## # A tibble: 6 x 5
##   term                           estimate std.error statistic   p.value
##   <chr>                             <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)                     -0.847    0.0125     -67.8  0.
## 2 is_repeated_guest1              -1.90     0.148      -12.8  9.74e- 38
## 3 previous_bookings_not_canceled  -0.947    0.0588     -16.1  2.62e- 58
## 4 previous_cancellations           3.87     0.141       27.5  8.45e-167
## 5 booking_changes                 -0.486    0.0231     -21.0  2.84e- 98
## 6 days_in_waiting_list            -0.0209   0.00358     -5.84 5.28e-   9
```

```r
AIC(logit_mod)
```

```
## [1] 43736.14
```

```r
BIC(logit_mod)
```

```
## [1] 43787.73
```