Taxi Analysis

In this project, I worked with r’s taxi dataset, one of its many preloaded datasets. My goal with the project are as follows: a) to create a model capable of accurately predicting if a passenger will tip or not and b) to understand if tipping varies by taxi company

Downloading Data and Packages

Our first step, as always, is to install the data and other required packages.

```
#Setting a CRAN mirror
options(repos = c(CRAN = "https://cran.r-project.org"))

library(modeldata)
library(tidyverse)
library(dplyr)
library(ggplot2)

data(taxi)
```

Exploratory Data Analysis

Next, we conduct some exporatory data analysis. From our basic examination of the dataset, we see that it includes 7 variables and 10000 observations. The variables are as follows.

- Tip: A binary variable recording if a passenger tipped or not
- Distance: A double recording the distance of the trip
- Company: A factor recording which company provided the ride
- Local: A binary variable recording if a passenger is a local
- Dow: A factor recording the day of the week
- Month: A factor recording what month of the year it is
- Hour: A integer recroding what hour of the day it is, operating on a single 24 hour cycle rather than a AM/PM cycle.

```
dim(taxi)
```

```
[1] 10000     7
```

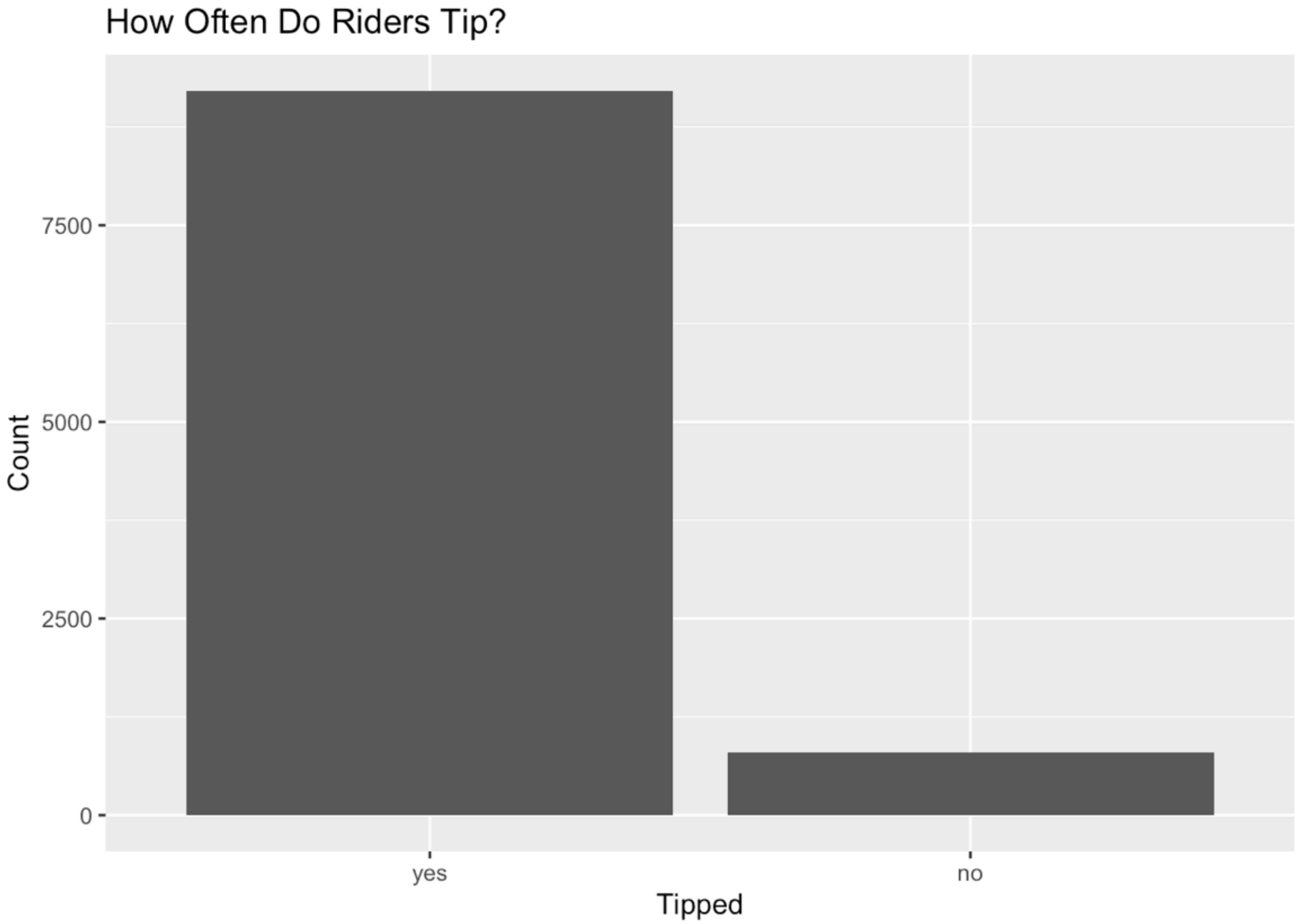
```r
glimpse(taxi)
```

```
Rows: 10,000
Columns: 7
$ tip      <fct> yes, yes, yes, yes, yes, yes, yes, yes, yes, yes, yes, yes, y…
$ distance <dbl> 17.19, 0.88, 18.11, 20.70, 12.23, 0.94, 17.47, 17.67, 1.85, 1…
$ company  <fct> Chicago Independents, City Service, other, Chicago Independen…
$ local    <fct> no, yes, no, no, no, yes, no, no, no, no, no, no, no, yes, no…
$ dow      <fct> Thu, Thu, Mon, Mon, Sun, Sat, Fri, Sun, Fri, Tue, Tue, Sun, W…
$ month    <fct> Feb, Mar, Feb, Apr, Mar, Apr, Mar, Jan, Apr, Mar, Mar, Apr, A…
$ hour     <int> 16, 8, 18, 8, 21, 23, 12, 6, 12, 14, 18, 11, 12, 19, 17, 13, …
```

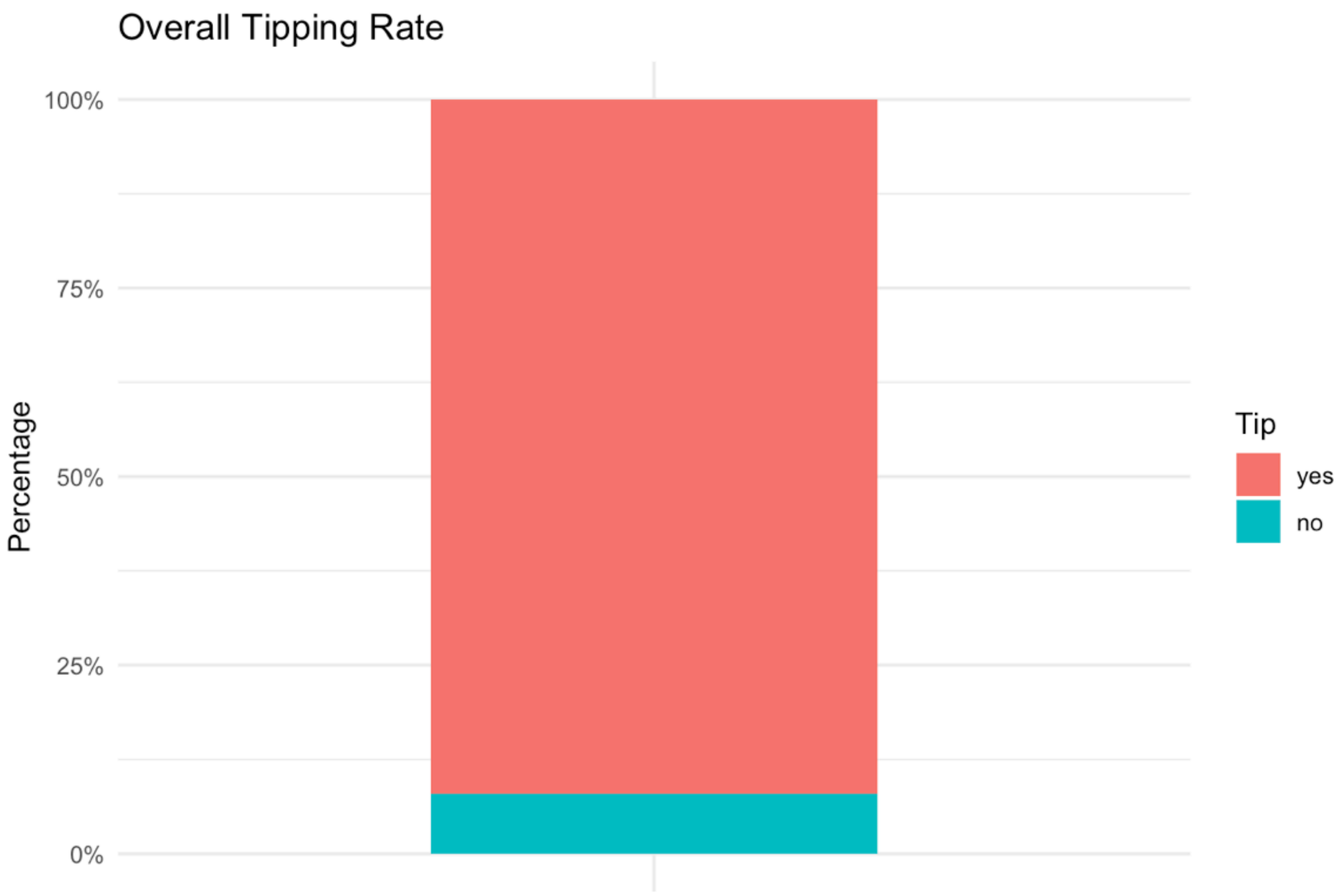
Understanding our binary variables

It is clear from our basic exploration that the vast majority (92%) of riders do indeed tip. As a result, it may be of more valuable to predict when they do not.

```r
#Absolute Number
ggplot(taxi, aes(x = tip)) +
  geom_bar() +
  labs(title = "How Often Do Riders Tip?", x = "Tipped", y = "Count")
```



```
#Percentage
ggplot(taxi, aes(x = "", fill = tip)) +
  geom_bar(position = "fill", width = 0.5) +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Overall Tipping Rate",
       x = "",
       y = "Percentage",
       fill = "Tip") +
  theme_minimal()
```

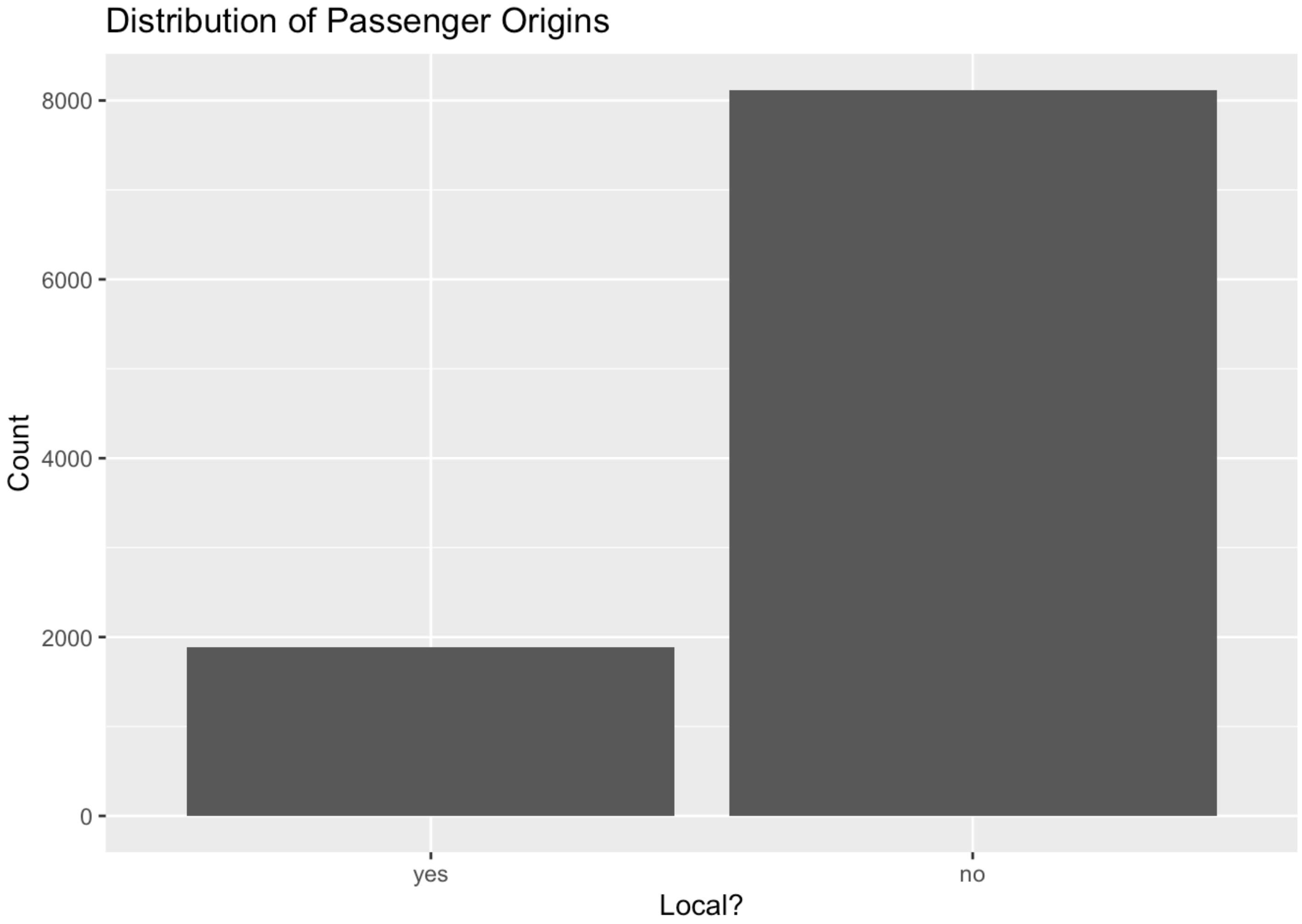


```
#Generating numerical results
taxi |>
  summarise(
    total_rides = n(),
    tipped_rides = sum(tip == "yes", na.rm = TRUE),
    tip_rate_percent = mean(tip == "yes", na.rm = TRUE) * 100
  )
```

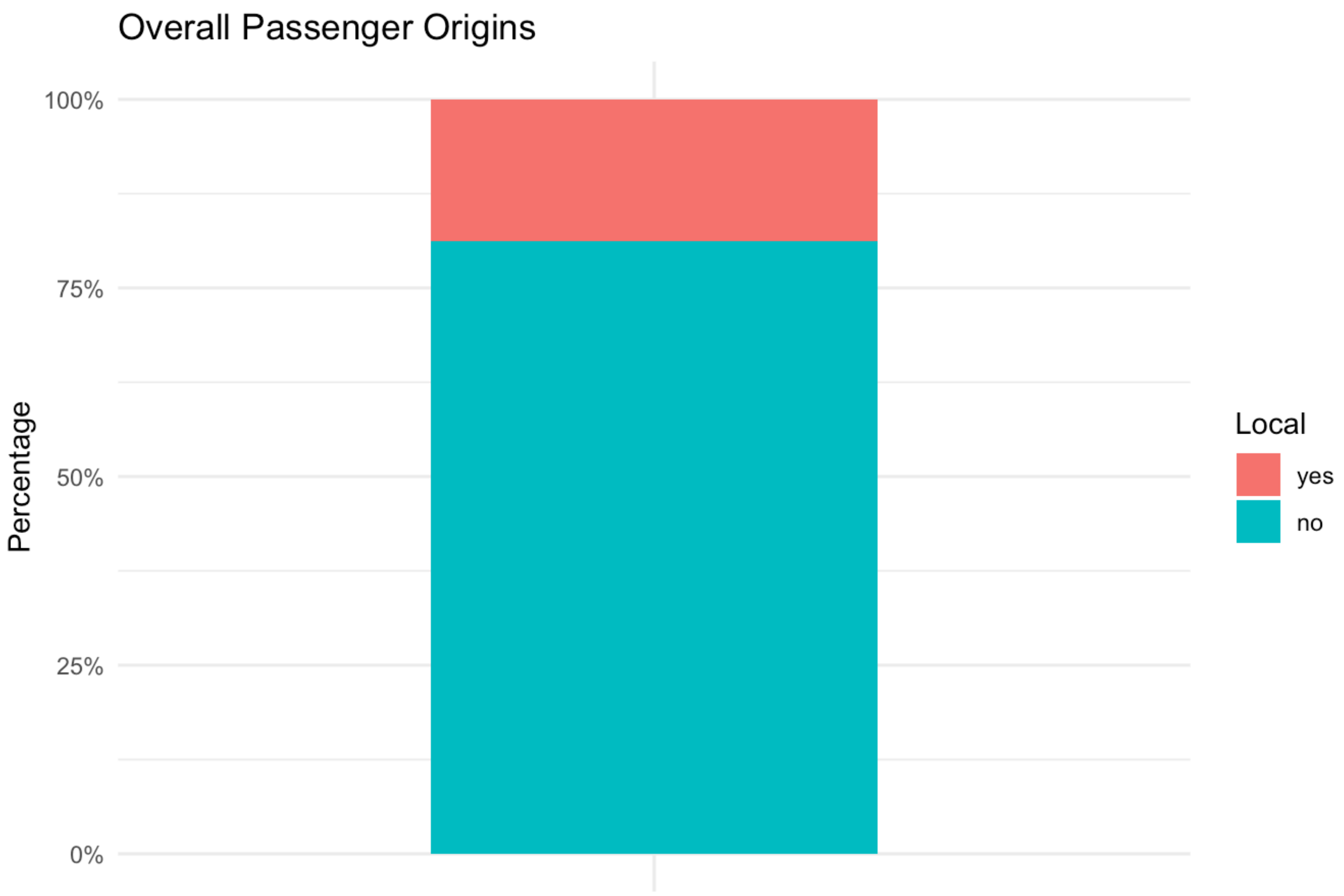
```
# A tibble: 1 × 3
  total_rides tipped_rides tip_rate_percent
        <int>        <int>            <dbl>
1       10000         9209             92.1
```

It also becomes clear that the majority of passengers (81.17%) are non-locals. This makes sense, as locals likely already have some method of transportation secured that is not a taxi. However, locals still take a notable number of rides (18.83%), likely in emergencies/other exceptional situations.

```
#Absolute Number
ggplot(taxi, aes(x = local)) +
  geom_bar() +
  labs(title = "Distribution of Passenger Origins", x = "Local?", y = "Count")
```



```
#Percentage
ggplot(taxi, aes(x = "", fill = local)) +
  geom_bar(position = "fill", width = 0.5) +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Overall Passenger Origins",
       x = "",
       y = "Percentage",
       fill = "Local") +
  theme_minimal()
```



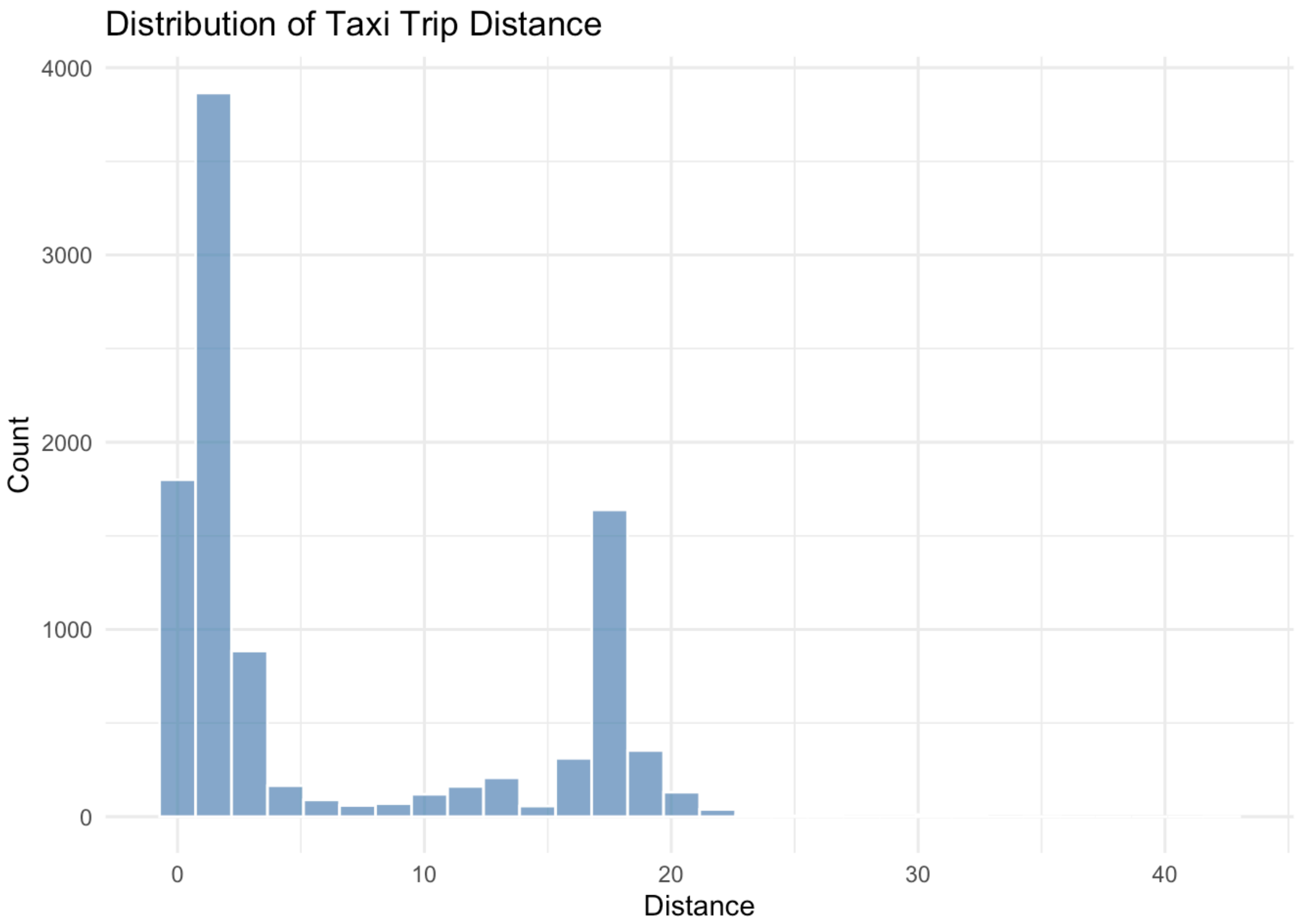
```
#Generating numerical results
taxi |>
  summarise(
    total_rides = n(),
    local_rides = sum(local == "yes", na.rm = TRUE),
    local_ride_percent = mean(local == "yes", na.rm = TRUE) * 100
  )
```

```
# A tibble: 1 × 3
  total_rides local_rides local_ride_percent
        <int>       <int>              <dbl>
1       10000        1883               18.8
```

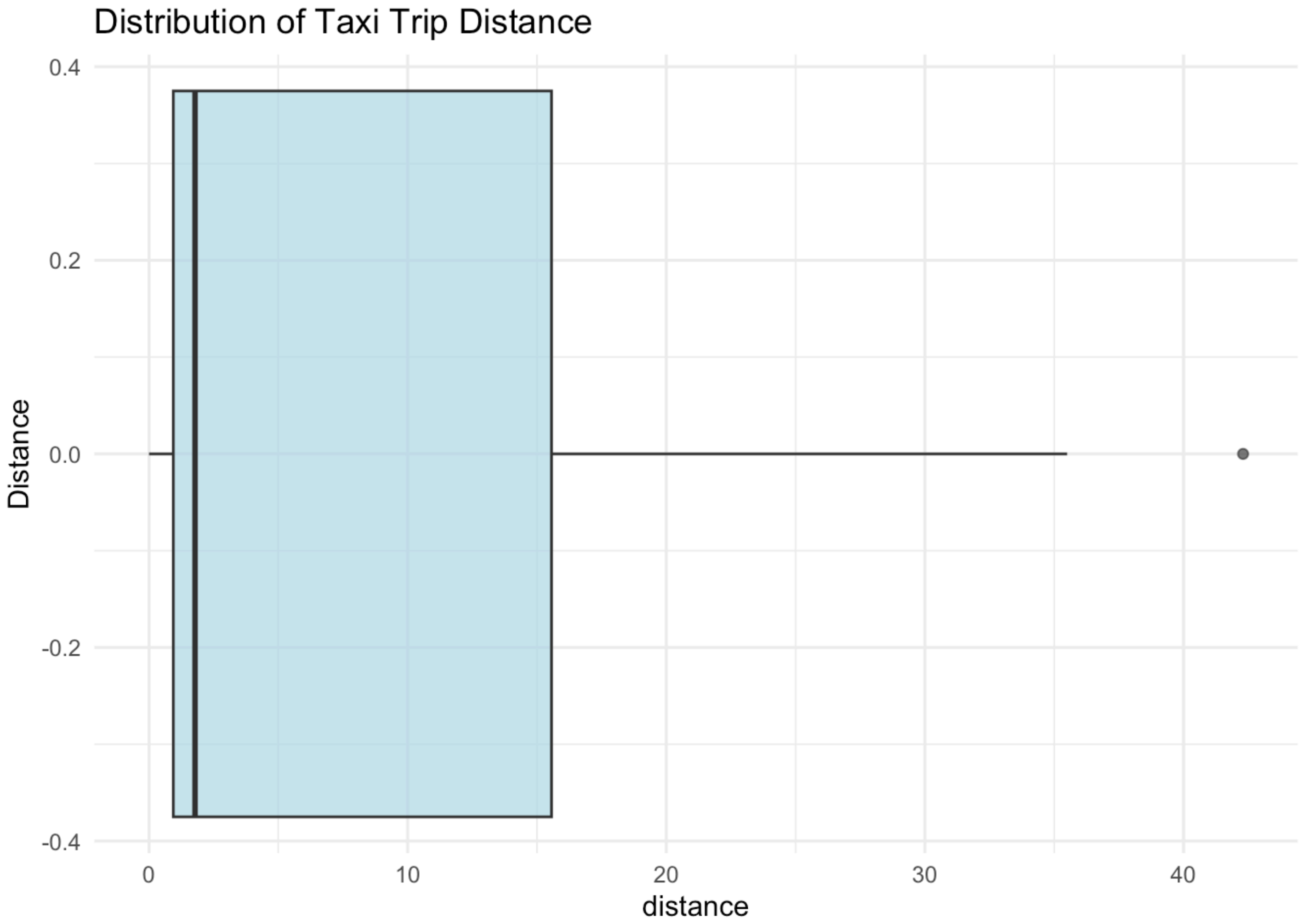
Understanding our non-binary variables

From our basic visualizations, it becomes clear that distance is highly bipolar and right-tailed, with most rides incredibly short (under five miles) but with a second cluster nearly the 16 mile market. Distance has a median of 1.78 miles, mean of 6.22 miles and standard deviation of 7.38 miles.

```r
ggplot(taxi, aes(x = distance)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white", alpha = 0.7) +
  labs(title = "Distribution of Taxi Trip Distance",
       x = "Distance",
       y = "Count") +
  theme_minimal()
```



```r
ggplot(taxi, aes(x = distance)) +
  geom_boxplot(fill = "lightblue", alpha = 0.7) +
  labs(title = "Distribution of Taxi Trip Distance",
       y = "Distance") +
  theme_minimal()
```



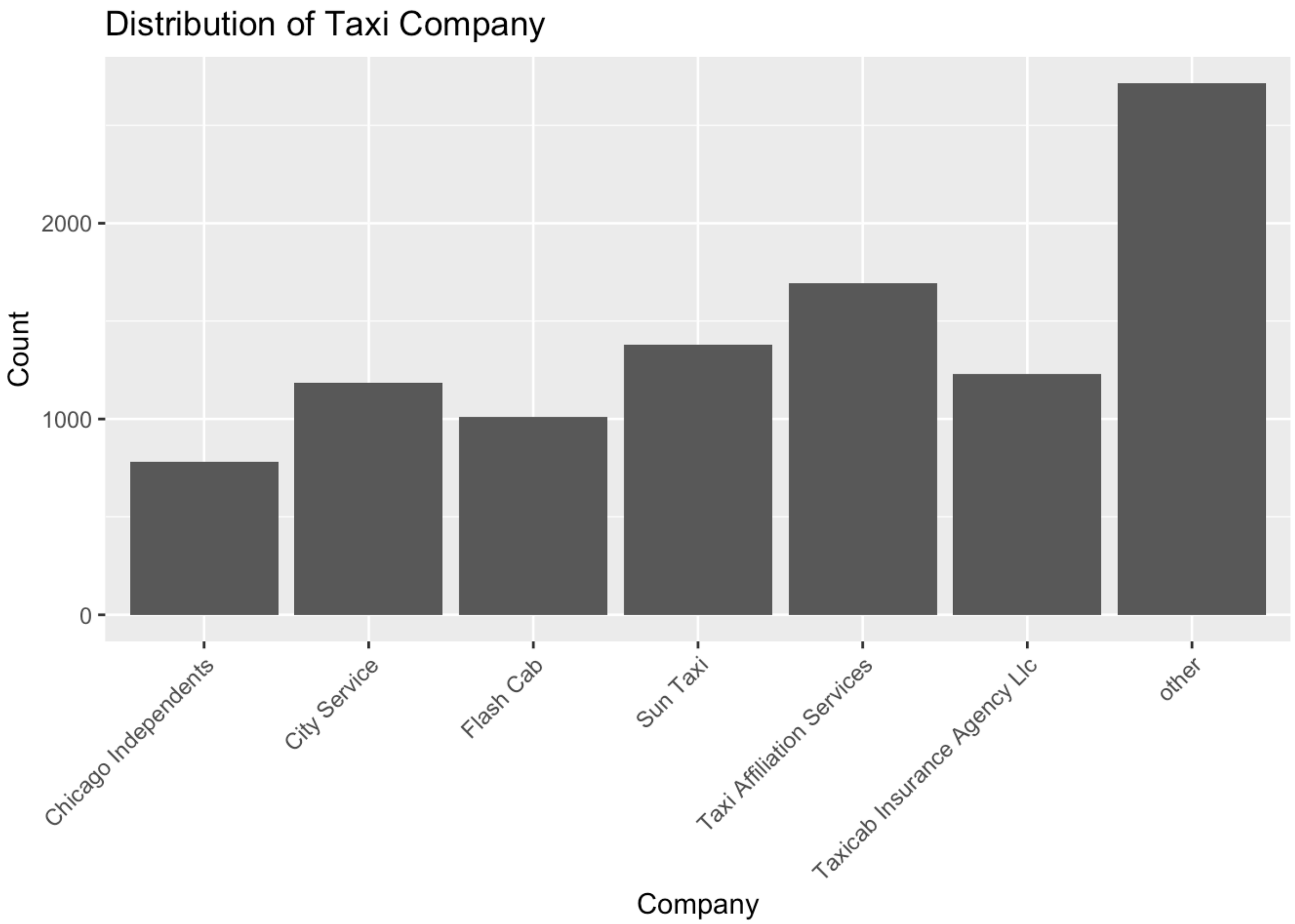
```
taxi |>
  summarise(
    count = n(),
    min_distance = min(distance, na.rm = TRUE),
    q1 = quantile(distance, 0.25, na.rm = TRUE),
    median = median(distance, na.rm = TRUE),
    mean = round(mean(distance, na.rm = TRUE), 2),
    q3 = quantile(distance, 0.75, na.rm = TRUE),
    max_distance = max(distance, na.rm = TRUE),
    std_dev = round(sd(distance, na.rm = TRUE), 2),
    missing_values = sum(is.na(distance))
  )
```

```
# A tibble: 1 × 9
  count min_distance    q1 median  mean    q3 max_distance std_dev
  <int>        <dbl> <dbl>  <dbl> <dbl> <dbl>        <dbl>   <dbl>
1 10000            0  0.94   1.78  6.22  15.6         42.3    7.38
```

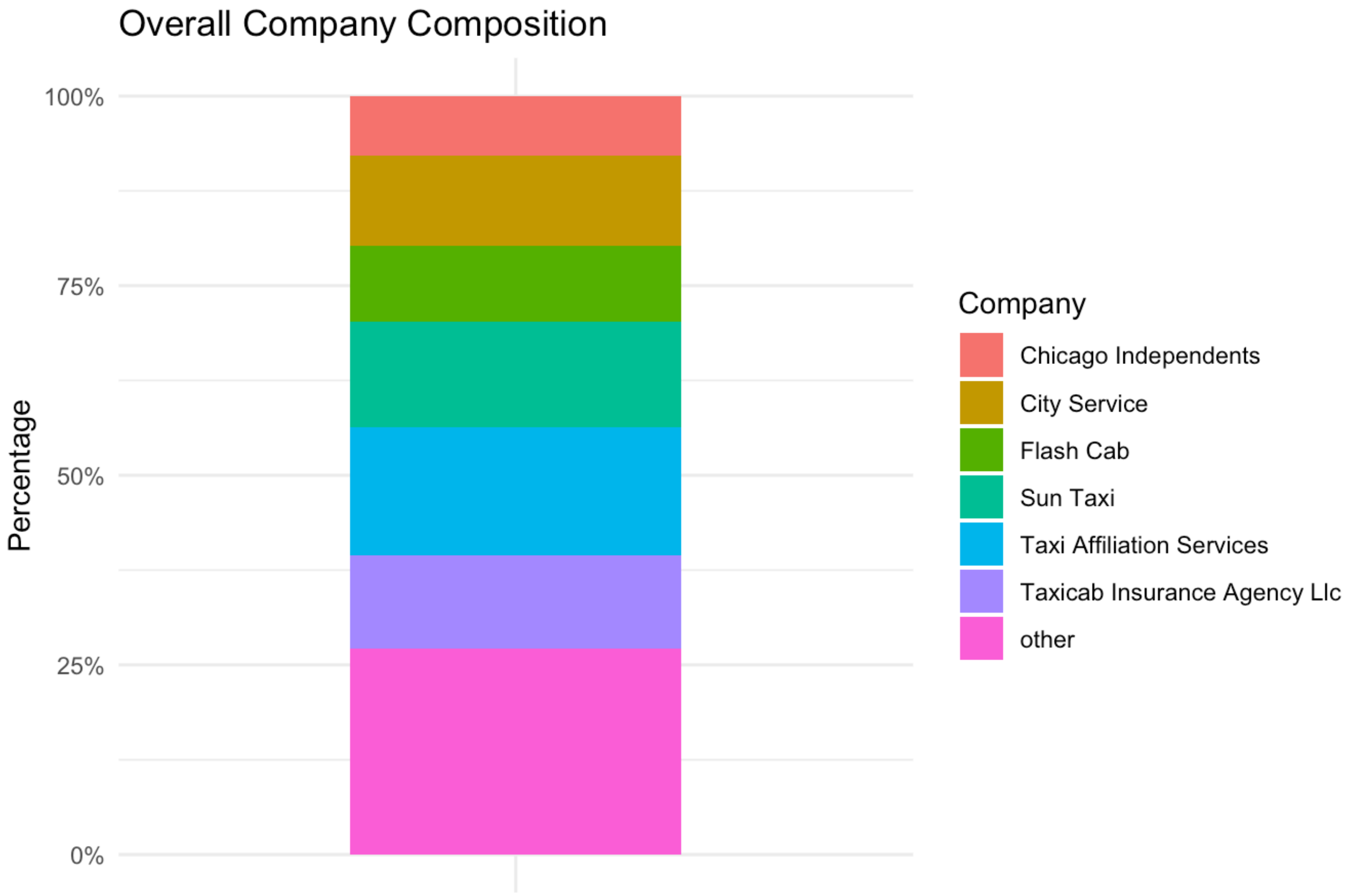
```
# ℹ 1 more variable: missing_values <int>
```

We have a total of six taxi companies in our data set (Chicago Independents, City Service, Sun Taxi, Flash Cab, Taxicab Insurance Agency Llc, and Taxi Affiliation Services) as well as a generic catch-all for other companies/non-company rides. Our data set is largely uniform with the exception of the “other” category, which dominates with ~27% of the business.

```
#Absolute Number
ggplot(taxi, aes(x = company)) +
  geom_bar() +
  labs(title = "Distribution of Taxi Company", x = "Company", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
#Percentage
ggplot(taxi, aes(x = "", fill = company)) +
  geom_bar(position = "fill", width = 0.5) +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Overall Company Composition",
       x = "",
       y = "Percentage",
       fill = "Company") +
  theme_minimal()
```

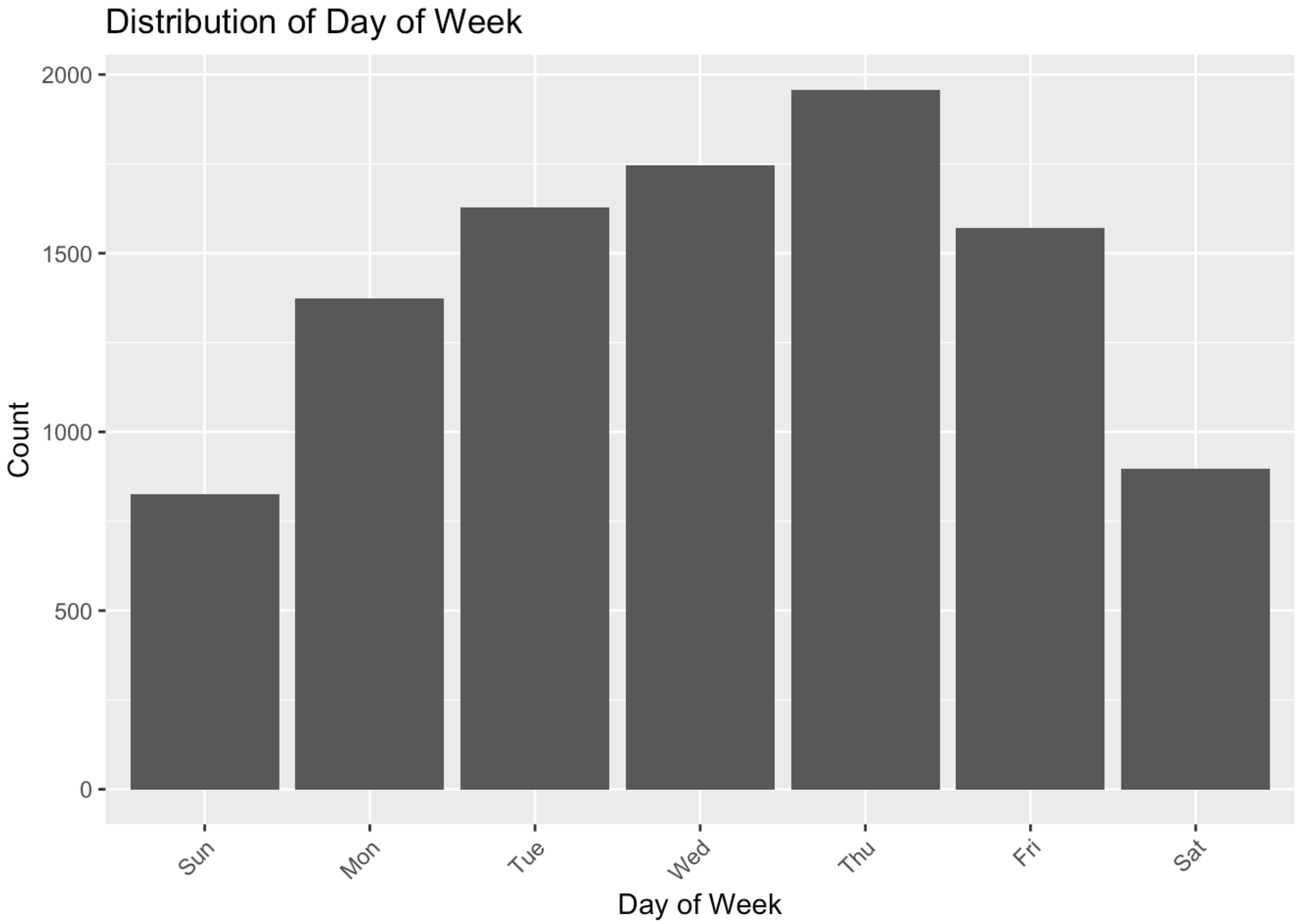


```
#Generating numerical results
taxi |>
  group_by(company) |>
  summarise(
    total_rides = n(),
  ) |>
  mutate(
    percentage = total_rides/sum(total_rides)
  ) |>
  select(company, percentage)
```

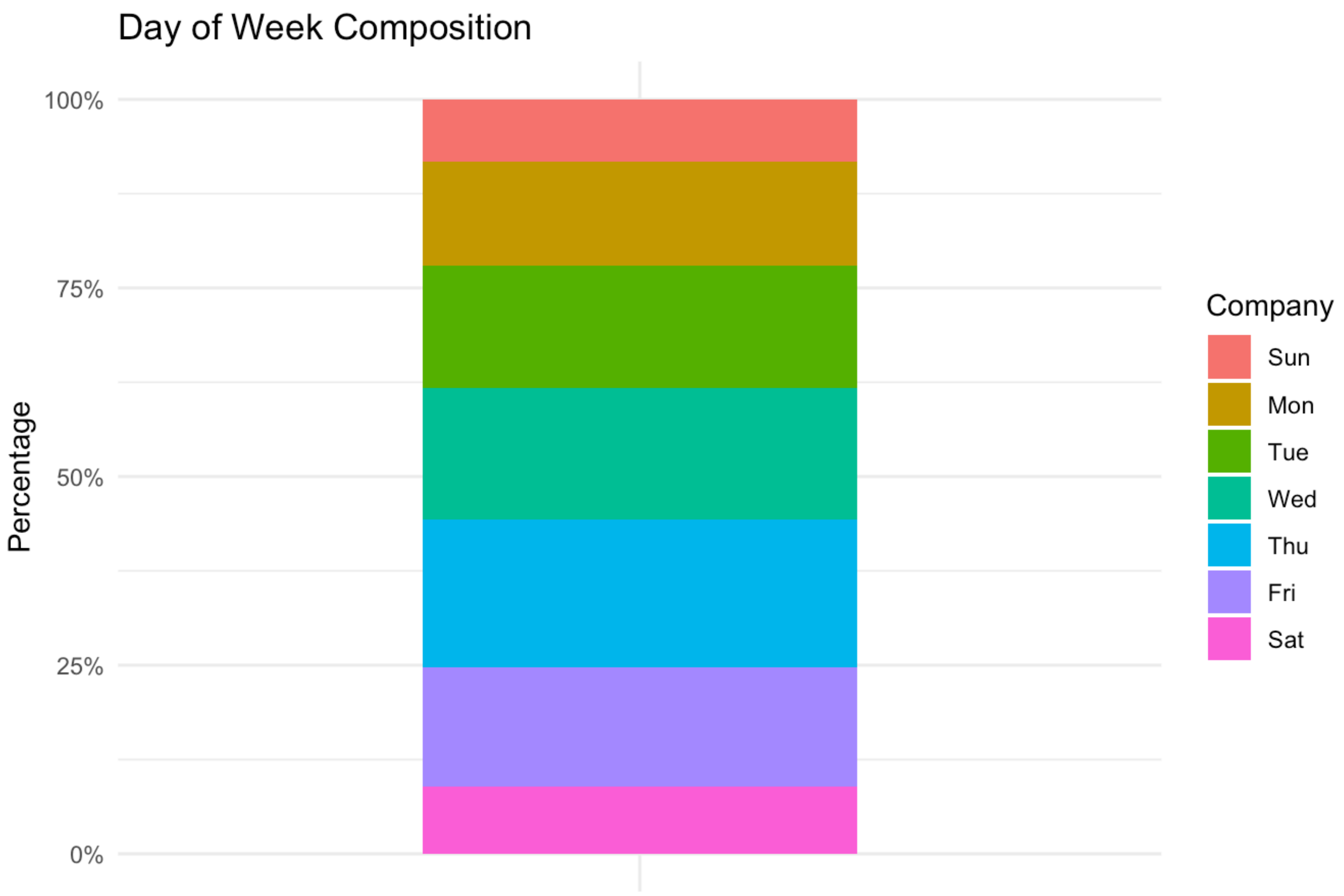
```
# A tibble: 7 × 2
  company                      percentage
  <fct>                             <dbl>
1 Chicago Independents             0.0781
2 City Service                     0.119
3 Flash Cab                        0.101
4 Sun Taxi                         0.138
5 Taxi Affiliation Services        0.169
6 Taxicab Insurance Agency Llc     0.123
7 other                            0.272
```

Our dataset includes all seven days of the week, albeit not in a uniform distribution. Instead, Taxi usage peaks on Thursdays (which are responsible for ~20% of total traffic) and is at its lowest on the weekends.

```
#Absolute Number
ggplot(taxi, aes(x = dow)) +
  geom_bar() +
  labs(title = "Distribution of Day of Week", x = "Day of Week", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
#Percentage
ggplot(taxi, aes(x = "", fill = dow)) +
  geom_bar(position = "fill", width = 0.5) +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Day of Week Composition",
       x = "",
       y = "Percentage",
       fill = "Company") +
  theme_minimal()
```



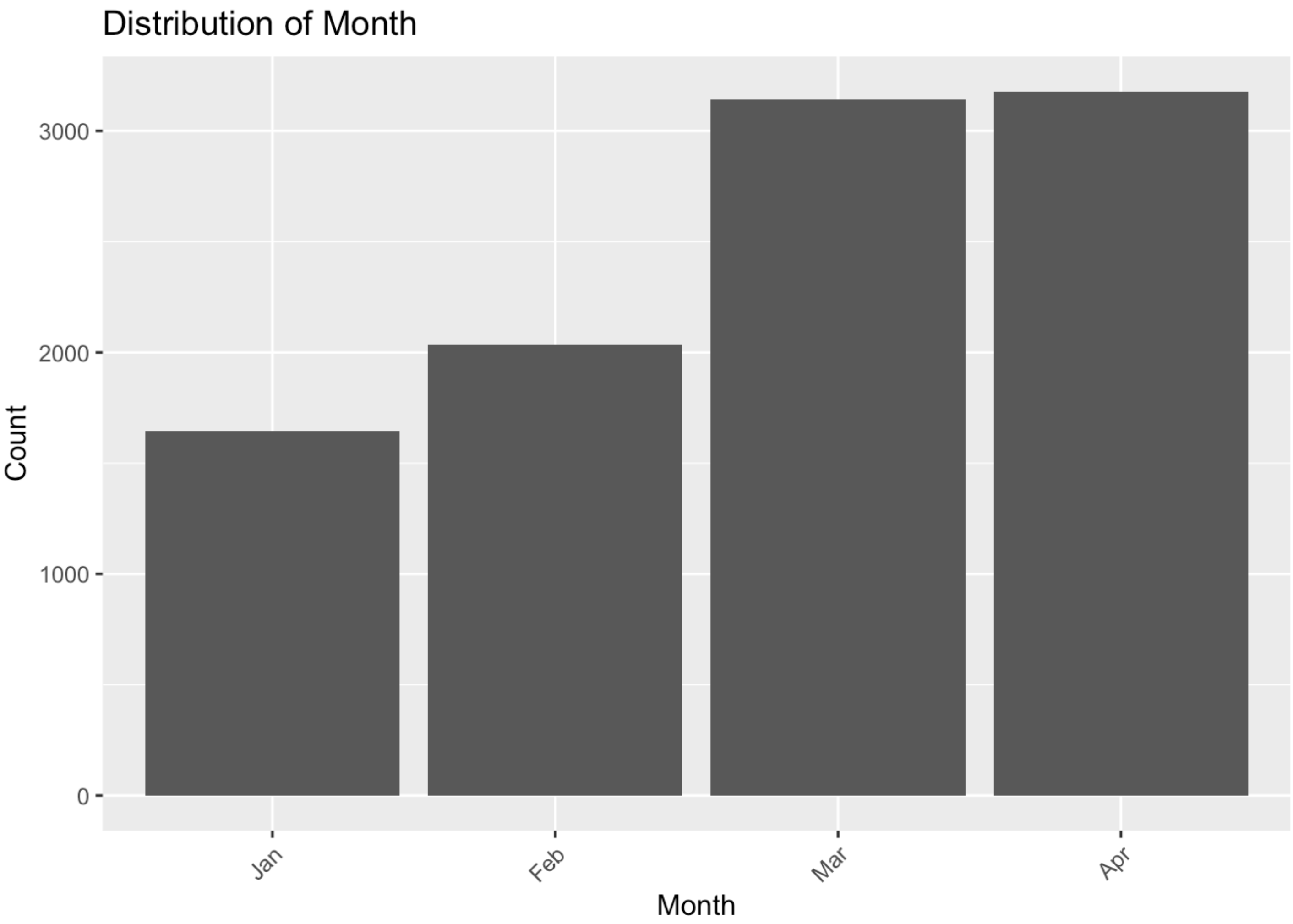
```
#Generating numerical results
taxi |>
  group_by(dow) |>
  summarise(
    total_days = n(),
  ) |>
  mutate(
    percentage = total_days/sum(total_days)
  ) |>
  select(dow, percentage)
```

```
# A tibble: 7 × 2
  dow   percentage
  <fct>      <dbl>
1 Sun       0.0827
2 Mon       0.137
3 Tue       0.163
```

```
4 Wed       0.175
5 Thu       0.196
6 Fri       0.157
7 Sat       0.0896
```

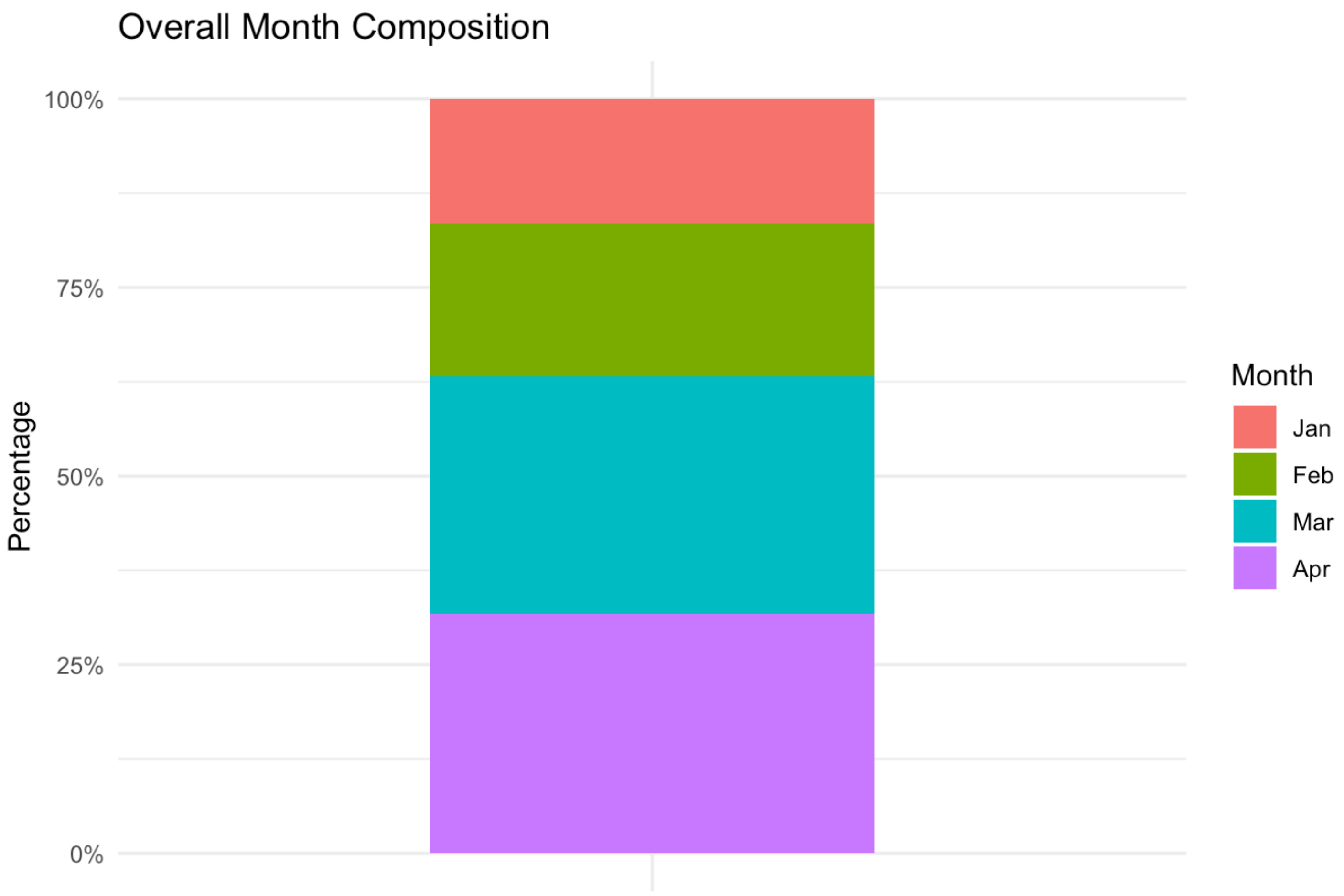
We only have the first four days of the month in our dataset, and it is clear that the later months are busier than the earlier ones. March and April are both at ~30% of total rides each, versus 16% for January and 20% for February.

```
#Absolute Number
ggplot(taxi, aes(x = month)) +
  geom_bar() +
  labs(title = "Distribution of Month", x = "Month", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
#Percentage
```

```
ggplot(taxi, aes(x = "", fill = month)) +
  geom_bar(position = "fill", width = 0.5) +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Overall Month Composition",
       x = "",
       y = "Percentage",
       fill = "Month") +
  theme_minimal()
```

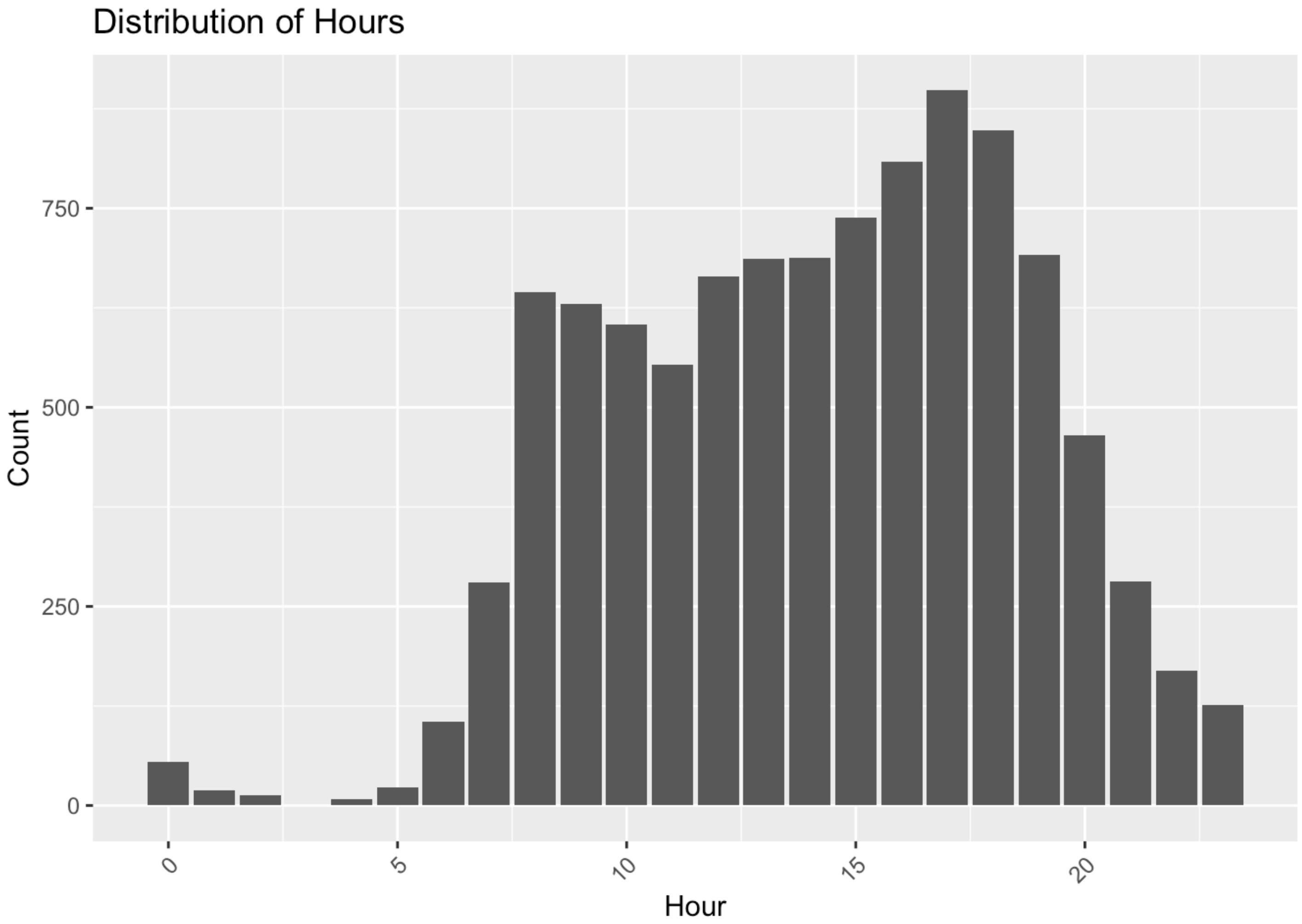


```
#Generating numerical results
taxi |>
  group_by(month) |>
  summarise(
    total_rides = n(),
  ) |>
  mutate(
    percentage = total_rides/sum(total_rides)
  ) |>
  select(month, percentage)
```

```
# A tibble: 4 × 2
  month percentage
  <fct>      <dbl>
1 Jan        0.164
2 Feb        0.204
3 Mar        0.314
4 Apr        0.318
```

It is clear that the majority of flights happen during what I would label “normal daytime hours.” Indeed, the mean and median times are 14.18 and 15 hours respectively, with a standard deviation of 4.36 hours.

```
#Absolute Number
ggplot(taxi, aes(x = hour)) +
  geom_bar() +
  labs(title = "Distribution of Hours", x = "Hour", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
#Generating numerical results
taxi |>
  group_by(hour) |>
  summarise(
    total_rides = n(),
  ) |>
  mutate(
    percentage = total_rides/sum(total_rides)
  ) |>
  select(hour, percentage)
```

```
# A tibble: 23 × 2
    hour percentage
   <int>      <dbl>
 1     0     0.0055
 2     1     0.0019
 3     2     0.0013
 4     4     0.0008
 5     5     0.0023
 6     6     0.0105
 7     7     0.028 
 8     8     0.0645
 9     9     0.063 
10    10     0.0604
# ℹ 13 more rows
```

```
taxi |>
  summarise(
    count = n(),
    min_distance = min(hour, na.rm = TRUE),
    q1 = quantile(hour, 0.25, na.rm = TRUE),
    median = median(hour, na.rm = TRUE),
    mean = round(mean(hour, na.rm = TRUE), 2),
    q3 = quantile(hour, 0.75, na.rm = TRUE),
    max_distance = max(hour, na.rm = TRUE),
    std_dev = round(sd(hour, na.rm = TRUE), 2),
    missing_values = sum(is.na(hour))
  )
```

```
# A tibble: 1 × 9
  count min_distance    q1 median  mean    q3 max_distance std_dev
  <int>        <int> <dbl>  <dbl> <dbl> <dbl>        <int>   <dbl>
1 10000            0    11     15  14.2    18           23    4.36
# ℹ 1 more variable: missing_values <int>
```

Predicting Tip

What factor matters most?

First, it is valuable to understand the extent to which each variable can help us predict the changes in tip/the likelihood of a tip. Since we're dealing with a binomial problem rather than a simple numerical prediction, we can't rely on R^2. Instead, I'm choosing to rely on a metric called Likelihood Ratio Chi-Square, which captures how much each variable reduces model deviance (with higher values equaling higher predictive power). From the results, it is clear that distance and company are the most powerful predictors of tip chance, followed by local. Day of week and month have some predictive power, while hour has little.

```
#Downloading a new package to get pseudo R^2
```

```r
install.packages("pscl")
```

```
The downloaded binary packages are in
    /var/folders/jx/19wzy4r974zgwcx8kgshk9400000gn/T//RtmpeybPZg/downloaded_packages
```

```r
library(pscl)
```

```
Classes and Methods for R originally developed in the
Political Science Computational Laboratory
Department of Political Science
Stanford University (2002-2015),
by and under the direction of Simon Jackman.
hurdle and zeroinfl functions by Achim Zeileis.
```

```r
#Fitting models
model1 <- glm(tip ~ distance, data = taxi, family = binomial)
model2 <- glm(tip ~ company, data = taxi, family = binomial)
model3 <- glm(tip ~ local, data = taxi, family = binomial)
model4 <- glm(tip ~ hour, data = taxi, family = binomial)
model5 <- glm(tip ~ dow, data = taxi, family = binomial)
model6 <- glm(tip ~ month, data = taxi, family = binomial)

null_model <- glm(tip ~ 1, data = taxi, family = binomial)

lr_results <- data.frame(
  variable = c("distance", "company", "local", "hour", "dow", "month"),
  lr_chisq = c(
    null_model$deviance - model1$deviance,  # distance
    null_model$deviance - model2$deviance,  # company
    null_model$deviance - model3$deviance,  # local
    null_model$deviance - model4$deviance,  # hour
    null_model$deviance - model5$deviance,  # dow
    null_model$deviance - model6$deviance   # month
  )
)

print(lr_results)
```

```
  variable  lr_chisq
1 distance 46.635505
2  company 55.536151
3    local 23.510580
```

```
4     hour  0.511610
5      dow  8.625229
6    month  7.398870
```

What’s the best model we can get?

Since 92% of the rides result in a tip, our job is a little trickier. After all, a model that always guesses “yes” will be 92% accurate, a pretty good outcome. However, building a predictive model is still possible and important.

Lasso, Ridge, and Traditional

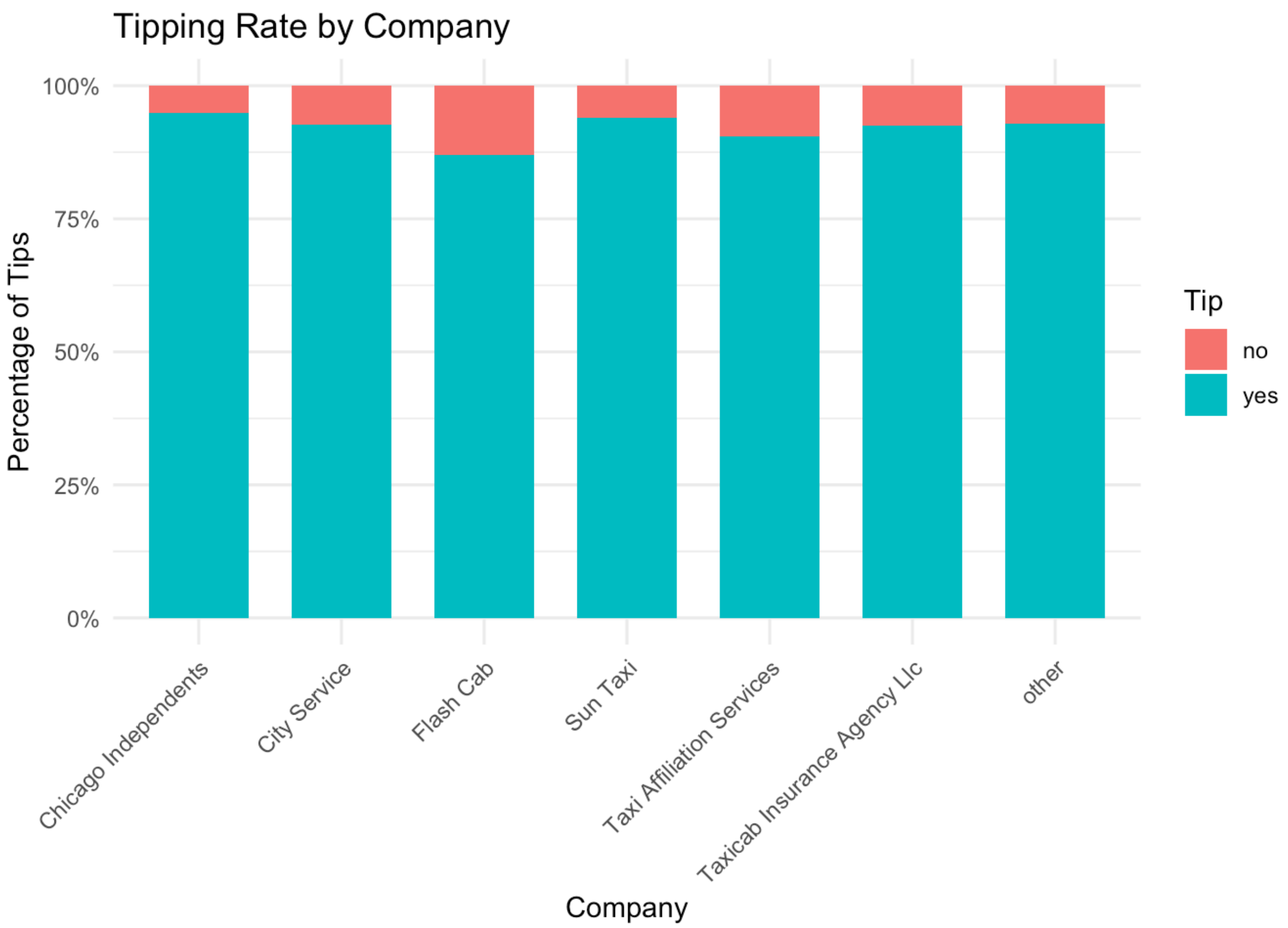
First, we build a traditional logistic regression using three versions of optimization: Lasso, Ridge, and Traditional. These are all remarkably similar in error when compared using an AUC, but a Likelihood Ratio Chi-Square shows that the full OLS seems to win out. The best model uses every single variable to make its predictions.

```
Loading required package: Matrix


Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack

Loaded glmnet 4.1-10

Type 'citation("pROC")' for a citation.


Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var

Setting levels: control = yes, case = no

Setting direction: controls < cases

Setting levels: control = yes, case = no
Setting direction: controls < cases

Setting levels: control = yes, case = no

Setting direction: controls < cases

Warning in Ops.factor(data_clean$tip, log(lasso_preds)): '*' not meaningful for
factors

Warning in Ops.factor(1, data_clean$tip): '-' not meaningful for factors

Warning in Ops.factor(data_clean$tip, log(ridge_preds)): '*' not meaningful for
factors

Warning in Ops.factor(1, data_clean$tip): '-' not meaningful for factors
```

How Do Tips Vary By Route?

Based on a just a single visualization, it becomes clear that from our sample alone that Flash Cab has the highest no tip rates and that tip rates do vary by company. Indeed, adding some simple calculations lets us see that Flash Cab has a 13% no tip rate, compared to 9% for Taxi Affiliation Services (the next highest) and a general level of about 7% no tips. The best performing (defined as the best chance of getting tips) is Chicago Independent.

```r
taxi$tip <- factor(taxi$tip, levels = c("no", "yes"))

ggplot(taxi, aes(x = company, fill = tip)) +
  geom_bar(position = "fill", width = 0.7) +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Tipping Rate by Company",
       x = "Company",
       y = "Percentage of Tips",
       fill = "Tip") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
tip_stats <- taxi |>
  group_by(company) |>
  summarise(
    n_total = n(),
    n_no_tip = sum(tip == "no"),
    non_tip_percentage = n_no_tip / n_total,
    se = sqrt((non_tip_percentage * (1 - non_tip_percentage)) / n_total),
    .groups = "drop"
  )

tip_stats
```

```
# A tibble: 7 × 5
  company                      n_total n_no_tip non_tip_percentage      se
  <fct>                          <int>    <int>              <dbl>   <dbl>
1 Chicago Independents             781       40             0.0512 0.00789
2 City Service                    1187       87             0.0733 0.00756
```

```
3 Flash Cab                       1010      132             0.131  0.0106
4 Sun Taxi                        1382       84             0.0608 0.00643
5 Taxi Affiliation Services       1694      160             0.0945 0.00711
6 Taxicab Insurance Agency Llc    1231       92             0.0747 0.00749
7 other                           2715      196             0.0722 0.00497
```