Investigation of Patterns Within Single Trip (Pay-as-you-ride) MetroBike Trips

Amanda Zhang, Ava Lim, Zoe Ly

SDS 322E Spring 2024 | Dr. Layla Guyot

Introduction

About the MetroBike System

- aims to strengthen the city's transportation methods and increase accessibility to bikes
- offers different payment plans, price ranges, and bike types

Motivations for Investigation

- understand user preferences (ie. peak usage hours or trip duration)
    - will help better allocate resources and optimize individual needs

Variables

- **Trip Duration**
- **Day Type** (either a weekday or weekend)
- **Checkout Time** (time after midnight in hours when bike was checked out)
- **Year**
- **Season** (Spring, Summer, Fall, and Winter)
- **Bike Type** (classic or electric).

Each unique row represents one bike trip.

EDA Research Questions:

1. *Is there a relationship between checkout time and the season a trip takes place?*
2. *Is there a relationship between the year a trip takes place and its duration?*
3. *Is there a relationship between the type of bike used and the day a trip takes place?*

Regression and Classification Research Questions:

1. *Can we predict a trip's duration given the year and checkout time?*
2. *Can we predict whether a classic or electric bike was used based on the trip duration and day type?*

Methods

*Original dataset: **35,320 rows** and **14 columns***

- *Subset: **32,551 rows** and **6 columns***

Original Variable Name	Example(s) of Original Variable	Mutated Variable Name	Example(s) of Mutated Variable	Other Modifications
Trip Duration Minutes	43	Duration	43	
Month	1	Season	Winter	changed into character format
Checkout Time	10:43:15	Time	104315	
Checkout Date	05/02/2023	DayType	Weekday	changed into character format
Bike Type	classic, electric	BikeType	0, 1	

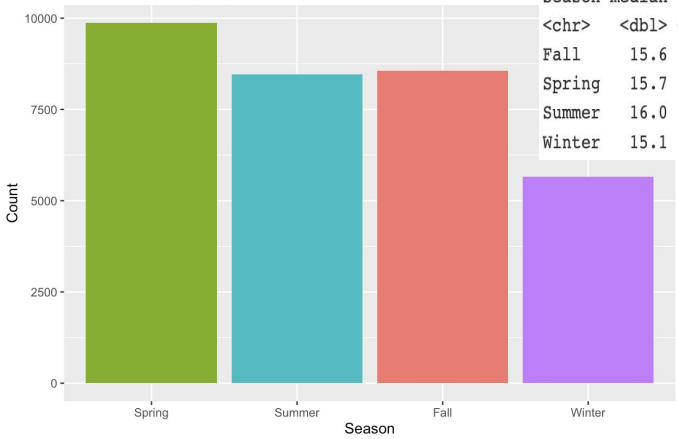
*We filtered the City of Austin MetroBike data for **Single Trip (Pay-as-you-ride) Memberships** that occurred **between 2020 and 2023** to obtain the data we worked with.*

Data is tidy

- *Each variable has its own column*
- *Each observation has its own row*
- *Each value has its own cell*

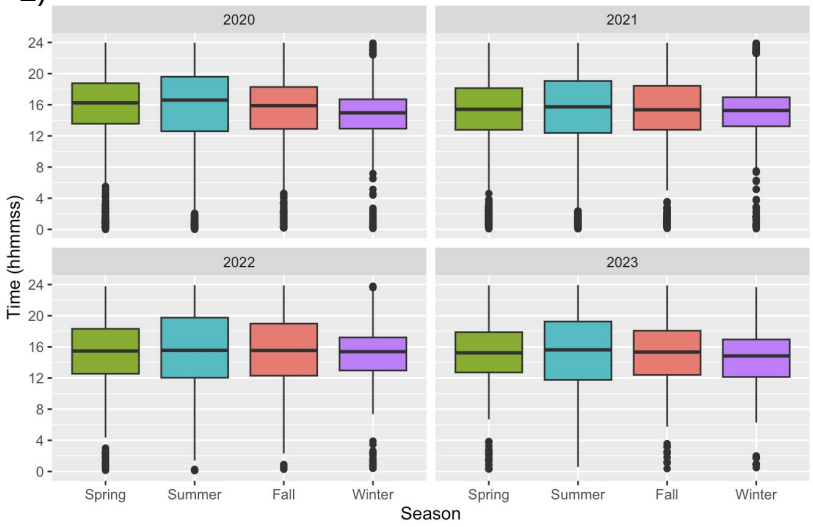
To predict...	***based on...***	***we will use the following model(s):***
Duration	Year + Checkout Time	Linear regression; kNN
BikeType	Duration + DayType	Logistic regression

1)



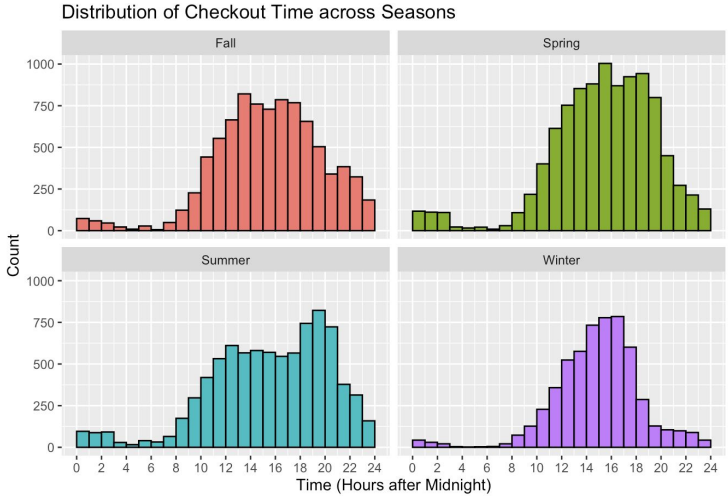
Season	median	IQR	mean
<chr>	<dbl>	<dbl>	<dbl>
Fall	15.6	5.58	15.4
Spring	15.7	5.45	15.3
Summer	16.0	6.95	15.5
Winter	15.1	3.94	14.9

2)



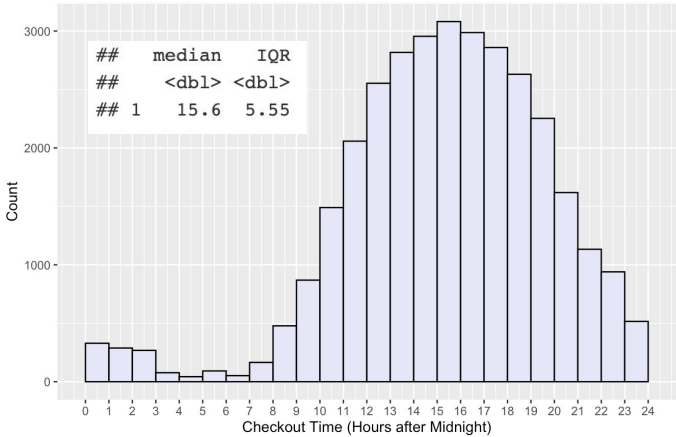
	Year	Season	median	IQR
	<dbl>	<chr>	<dbl>	<dbl>
1	2020	Fall	15.9	5.38
2	2020	Spring	16.2	5.2
3	2020	Summer	16.6	7
4	2020	Winter	15.0	3.77
5	2021	Fall	15.4	5.65
6	2021	Spring	15.4	5.35
7	2021	Summer	15.7	6.67
8	2021	Winter	15.3	3.73
9	2022	Fall	15.5	6.68
10	2022	Spring	15.5	5.77
11	2022	Summer	15.6	7.72
12	2022	Winter	15.4	4.24
13	2023	Fall	15.3	5.67
14	2023	Spring	15.2	5.17
15	2023	Summer	15.6	7.49
16	2023	Winter	14.8	4.82

3)



Season	n
<chr>	<int>
Fall	8558
Spring	9869
Summer	8461
Winter	5663

4)



EDA 1

Model 1 | knnreg(Duration ~ Year + Time)

```
fit_knn <- knnreg(Duration ~ Year + Time, data = MetroBike, k=5)

MetroBike |> mutate(predicted = predict(fit_knn, MetroBike)) |> select(Duration, predicted) |> head(n=10)
```

```
## # A tibble: 10 × 2
##    Duration predicted
##       <dbl>     <dbl>
##  1       43      61.9
##  2       14     306.
##  3        5     125.
##  4       18      42.1
##  5       21      42.2
##  6       14      35.9
##  7       77      42.2
##  8       73      39.4
##  9        7      35.4
## 10       76      35.4
```

```
sqrt(mean((MetroBike$Duration - predict(fit_knn, MetroBike))^2))
```

```
## [1] 304.3085
```

Model 1 | Cross-Validation

Average RMSE across 5 folds	346.3188 minutes	Poor performance
Standard Deviation	51.23808 minutes	Inconsistent performance

Creating the folds:

```
# Choose number of folds
k = 5

# To have the same random sample, use set.seed
set.seed(322)

# Randomly order rows in the dataset
data <- MetroBike[sample(nrow(MetroBike)), ]

# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)
```

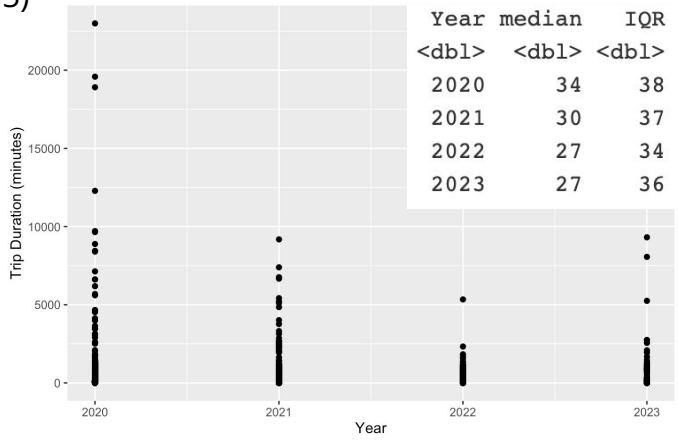
Performing the 5-fold cross validation and finding the mean performance and variatifion:

```
# Initialize a vector to keep track of the performance for each k-fold
perf_k <- NULL

# Use a for-loop to get performance for each k-fold
for(i in 1:k){
  # Split data into train and test data
  train_not_i <- data[folds != i, ] # train data = all observations except in fold i
  test_i <- data[folds == i, ]  # test data = observations in fold i

  # Train model on train data (all but fold i)
  train_model <- knnreg(Duration ~ Year + Time,
                        data = train_not_i,
                        k=5)

  # Performance listed for each test data = fold i
  perf_k[i] <- sqrt(mean((
    test_i$Duration - predict(train_model, newdata = test_i))^2,
    na.rm = TRUE))
}

# Performance for each fold
perf_k

#stats
mean(perf_k)
sd(perf_k)
```

```
[1] 328.4984 352.0900 278.8622 420.9967 351.1466
[1] 346.3188
[1] 51.23808
```

5)

Trip durations of rides (minutes) by year

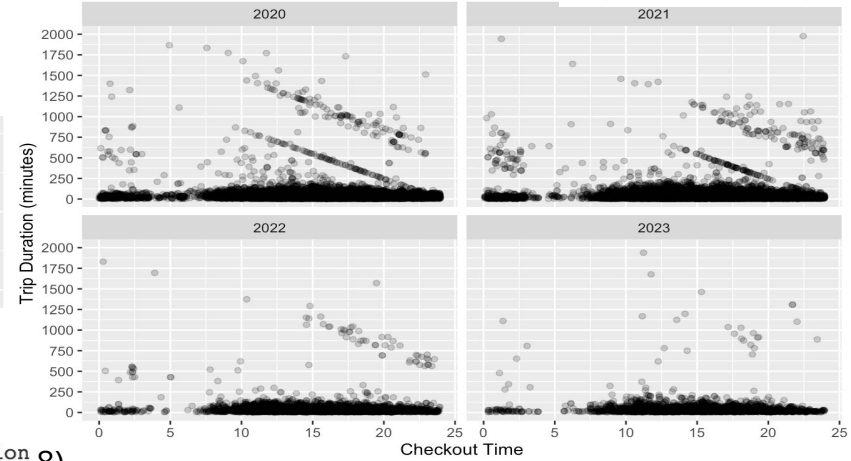


Year	median	IQR
<dbl>	<dbl>	<dbl>
2020	34	38
2021	30	37
2022	27	34
2023	27	36

6)

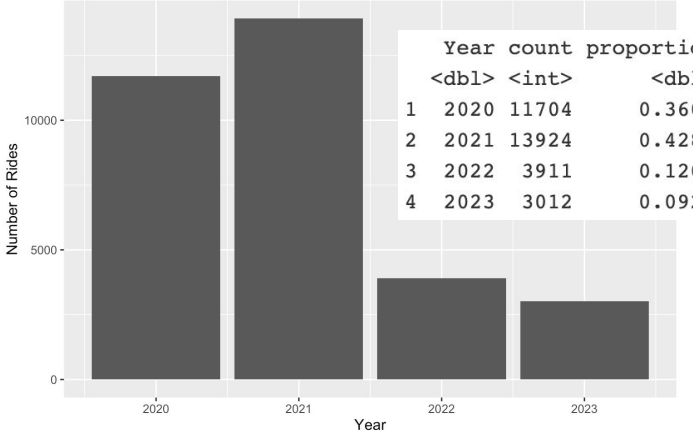
	Year	`Mean Checkout Time`	`SD Checkout Time`	`Mean Trip Duration`
	<dbl>	<dbl>	<dbl>	<dbl>
1	2020	15.4	4.40	75.3
2	2021	15.3	4.33	58.3
3	2022	15.3	4.22	52.2
4	2023	15.1	4.10	57.5

Trip Duration vs. Checkout Time, by Year



7)

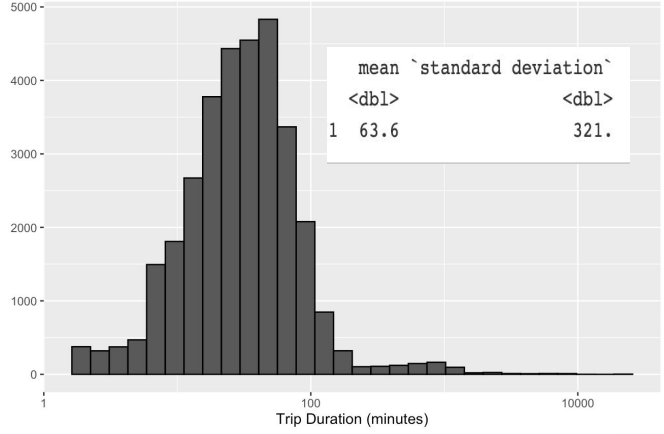
Number of recorded bike rides per year



	Year	count	proportion
	<dbl>	<int>	<dbl>
1	2020	11704	0.360
2	2021	13924	0.428
3	2022	3911	0.120
4	2023	3012	0.0925

8)

Distribution of trip duration (natural log fitted onto x axis)



mean	`standard deviation`
<dbl>	<dbl>
63.6	321.

EDA 2

Model 2 | lm(Duration ~ Year + Time)

```{r}
# a linear regression that uses year and time to predict trip duration
fit_lin_both <- lm(Duration ~ Year + Time, data = MetroBike)

# look at coeff.s and p-values for this reg
summary(fit_lin_both)
```

```
Call:
lm(formula = Duration ~ Year + Time, data = MetroBike)

Residuals:
    Min      1Q  Median      3Q     Max
  -99.4   -47.8   -32.3   -10.3 22910.8

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.610e+04  3.437e+03   4.685 2.80e-06 ***
Year        -7.923e+00  1.701e+00  -4.659 3.18e-06 ***
Time        -1.699e-04  4.052e-05  -4.193 2.76e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 328 on 35317 degrees of freedom
Multiple R-squared:  0.00108,   Adjusted R-squared:  0.001023
F-statistic: 19.09 on 2 and 35317 DF,  p-value: 5.16e-09
```

```{r}
# measures of fit: RMSE, R^2
## root mean square error
sqrt(mean(resid(fit_lin_both)^2))
## coefficient of determination
summary(fit_lin_both)$adj.r.squared
```

```
[1] 327.9545
[1] 0.00102348
```

Model 2 | Cross-Validation

```r
# Choose number of folds
k = 5
# To have the same random sample, use set.seed
set.seed(322)

# Randomly order rows in the dataset
data <- MetroBike[sample(nrow(MetroBike)), ]

# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Initialize a vector to keep track of the performance for each k-fold
perf_k <- NULL

# Use a for-loop to get performance for each k-fold
for(i in 1:k){
  # Split data into train and test data
  train_not_i <- data[folds != i, ] # train data = all observations except in fold i
  test_i <- data[folds == i, ]  # test data = observations in fold i

  # Train model on train data (all but fold i)
  train_model <- lm(Duration ~ Year + Time,
                    data = train_not_i)

  # Performance listed for each test data = fold i
  perf_k[i] <- sqrt(mean((
    test_i$Duration - predict(train_model, newdata = test_i))^2,
    na.rm = TRUE))
}

# Performance for each fold
perf_k
```

```
[1] 306.4002 328.5955 249.9060 410.8395 323.9941
```

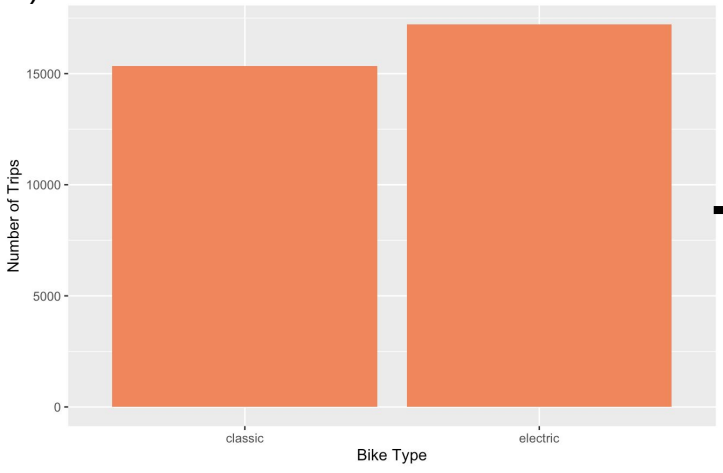
Average RMSE across 5 folds	323.9471 minutes	Poor performance
Standard Deviation	57.7968 minutes	Inconsistent performance

```r
# mean and std of RMSE
mean(perf_k)
sd(perf_k)
```

```
[1] 323.9471
[1] 57.79679
```

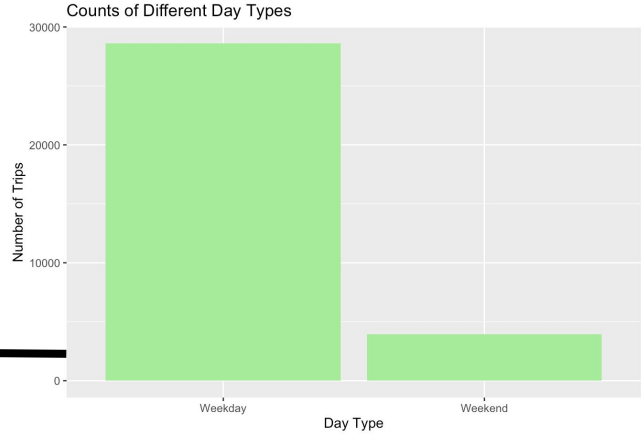
EDA 3

9)



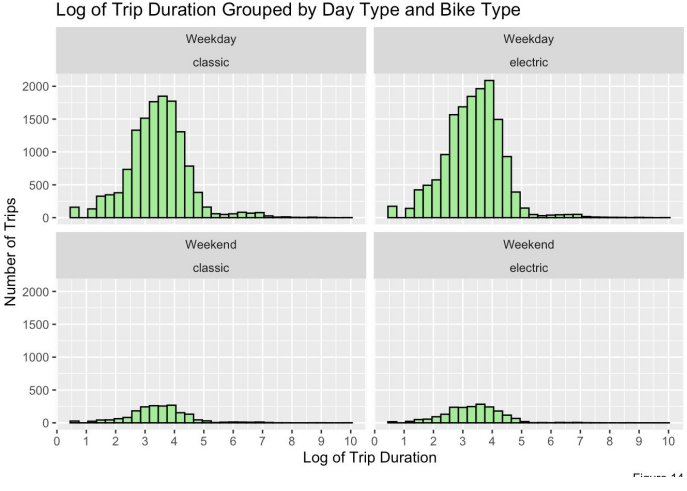
BikeType	count	proportion
<chr>	<int>	<dbl>
1 classic	15339	0.471
2 electric	17212	0.529

10)



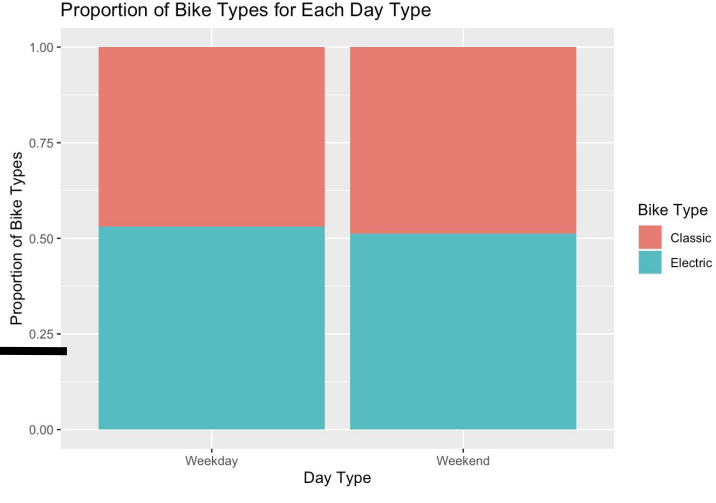
DayType	count	proportion
<chr>	<int>	<dbl>
1 Weekday	28603	0.879
2 Weekend	3948	0.121

11)



BikeType	DayType	median	IQR
<chr>	<chr>	<dbl>	<dbl>
1 classic	Weekday	3.47	1.17
2 classic	Weekend	3.43	1.10
3 electric	Weekday	3.40	1.20
4 electric	Weekend	3.37	1.22

12)



	Weekday	Weekend
classic	0.4689718	0.4875887
electric	0.5310282	0.5124113

Model 3 | glm(BikeType ~ DayType + Duration)

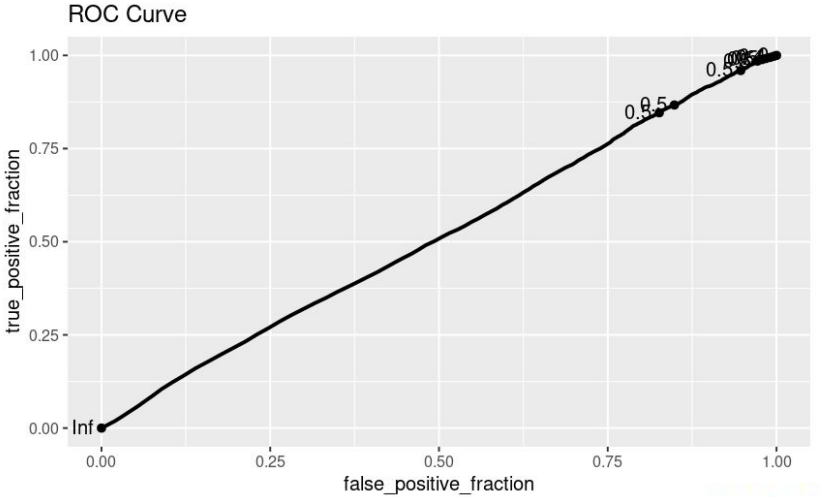
```r
# Overwrite existing dataset
MetroBike <- MetroBike |>
# Recode the outcome variable categories as '0' and '1' for logistic regression
mutate(BikeType = ifelse(BikeType == "classic", "0", "1")) |>
# Change `BikeType` variable to a numeric variable
mutate(BikeType = as.numeric(BikeType))

# Fit the model
fit_log <- glm(BikeType ~ DayType + Duration,
               data = MetroBike,
               family = "binomial")
# Look athe model summary
summary(fit_log)
```

```
Call:
glm(formula = BikeType ~ DayType + Duration, family = "binomial",
    data = MetroBike)

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.090e-02  1.171e-02  -2.639  0.00831 **
DayTypeWeekend -5.991e-02  3.270e-02  -1.832  0.06692 .
Duration       -1.907e-04  4.714e-05  -4.046 5.22e-05 ***
```

```r
BikeType_pred <- MetroBike |>
  # Create new variables for probability and predicted values
  mutate(probability = predict(fit_log, type = "response"),
         # if the probability is greater than 0.5, predictio = 1
         predicted = ifelse(probability > 0.5, 1, 0)) |>
          # select for relevant variables
  select(BikeType, DayType, Duration, probability, predicted)
# Take a look
head(BikeType_pred, 3)
```

BikeType <dbl>	DayType <chr>	Duration <dbl>	probability <dbl>	predicted <dbl>
0	Weekend	43	0.4752668	0
1	Weekday	14	0.4916075	0
1	Weekday	5	0.4920365	0

```r
# Make an ROC curve
ROC <- BikeType_pred |>
  # Calculate predictions
  mutate(probability = predict(fit_log, type = "response")) |>
  ggplot() +
  geom_roc(aes(d = BikeType, m = probability), n.cuts = 10)

# View ROC curve
ROC

# Calculate the area under the curve
calc_auc(ROC)$AUC
```




```
[1] 0.5132656
```

Model 3 | Cross-Validation

```r
# Select number of folds
k = 5

# Use the same random sample
set.seed(322)

# Randomly order rows in the dataset
data <- MetroBike[sample(nrow(MetroBike)), ]

# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Make a vector to track performances for each k-fold
perf_k <- NULL
# Use a for-loop to get performance for each k-fold
for(i in 1:k){
  # Split data into train and test data
  train_not_i <- data[folds != i, ] # train data = all observations except in fold i
  test_i <- data[folds == i, ]  # test data = observations in fold i

  # Train model on train data (all but fold i)
  train_model <- glm(BikeType ~ DayType + Duration,
                     data = train_not_i,
                     family = "binomial")

  # Performance listed for each test data = fold i
  perf_k[i] <- calc_auc(
  # Make a ROC curve
  ggplot(test_i) +
    geom_roc(aes(
      # Outcome is `BikeType`
      d = BikeType,
      # Probability of bike type based on the logistic model
      m = predict(train_model, newdata = test_i, type = "response")))
```

```r
# Performance for each fold
perf_k

# Average performance across k folds
mean(perf_k)

# Standard deviation of performance across k folds
sd(perf_k)
```

```
[1] 0.5173021 0.5010607 0.5164287 0.5170835 0.5145903
[1] 0.5132931
[1] 0.006920712
```

Average AUC across 5 folds	51.3293%	Poor performance
Standard Deviation	0.6921%	Consistent performance

Results

Across all three regression models we fitted, the results were **consistently poor,** with very high RMSEs for models with a numeric outcome and a low AUC for the logistic regression. This suggests that **the predictor variables were not reliable indicators of the outcome variable in each model.**

- The overall performance of the **KNN Regression Model** was an **RMSE of 304.3085 minutes**, and the average performance from the **cross validation had a mean of 346.3188 minutes and standard deviation of 51.23808 minutes**. The average performance was worse than the overall performance, and from the high standard deviation, it seems that the model overfits the new data.
- The **Linear Regression Model** for the same prediction had an **overall RMSE of 327.9545 minutes**, a **mean RMSE of 323.9471 minutes, and standard deviation of 57.79679 minutes** from the cross validation**.** As we can see by the model's large standard deviation as well, the model is overfitting the data.
- The linear regression model has a lower mean RMSE than the KNN model, suggesting that **the linear model is slightly better in predicting Trip Duration than the KNN model.**
- The high average RMSE values from both models suggests that the value of Trip Duration predicted by both models were **very far off from the true value**. Because of the very high RMSE values for both, we can conclude that these models **do not accurately predict new data**.
- The overall performance of the **Logistic Regression model** was poor as the **AUC was only 51.3266%**. Similarly, the average performance from the **cross validation** was also poor but consistent across the five folds. The **average AUC was 51.329%** with a **standard deviation of 0.6291%.** Therefore, the model **does not perform well, but it predicts new data as well as it can predict values in the dataset.**

From the EDA Graphs:

- Spring was the most popular time to travel, and the checkout time was fairly normally distributed. However, we have shown that there is not much of a relationship between the two variables.
- There seems to be a relationship between Year and Trip Duration, as the spread of Trip Duration is noticeably larger in the years 2020 and 2021.
- The distribution of bike types across day types is highly similar, suggesting the two variables are not strongly related. However, the sample of weekend bike trips is significantly smaller than that of the weekday trips, limiting the applicability of the findings.

Results

Across all three regression models we fitted, the results were **consistently poor,** with very high RMSEs for models with a numeric outcome. This suggests that **the predictor variables were not reliable indicators of the outcome variable in each model.**

- The overall performance of the **KNN Regression Model** was an **RMSE of 304.3085 minutes**, and the average performance from the **cross validation had a mean of 346.3188 minutes and standard deviation of 51.23808 minutes**. The average performance was worse than the overall performance, and from the high standard deviation, it seems that the model overfits the new data.
- The **Linear Regression Model** for the same prediction had an **overall RMSE of 327.9545 minutes**, a **mean RMSE of 323.9471 minutes, and standard deviation of 57.79679 minutes** from the cross validation**.** As we can see by the model's large standard deviation as well, the model is overfitting the data.
- The linear regression model has a lower mean RMSE than the KNN model, suggesting that **the linear model is slightly better in predicting Trip Duration than the KNN model.**
- The high average RMSE values from both models suggests that the value of Trip Duration predicted by both models were **very far off from the true value**. Because of the very high RMSE values for both, we can conclude that these models **do not accurately predict new data**.

Discussion

1. Our main takeaway from this analysis is that there is likely no relationship between any of the variables we explored, and that humans act very unpredictable in the MetroBike industry concerning single ride memberships.

2. Answers to research questions:
   a. We cannot accurately predict a trip's duration using a KNN regression model given the year and checkout time. The overall RMSE value of 304.3085 suggests that our predictions were on average 304 minutes off from the actual duration.
   b. A trip's year and checkout time are not good linear predictors of a trip's duration. We can conclude this because the RMSE of 323.9471 is extremely high, and the R^2 value reports that less than 1% of the variation in trip duration can be explained by year and checkout time.
   c. We cannot make accurate predictions about the type of bike used with trip duration and day type as predictors. The overall AUC of 51.3266% indicates that the logistic regression performs poorly.

3. The data did not match what we expected - although we could infer from the EDA project the regressions would not perform very well, we were surprised at how bad our predictions were. The high (300s) RMSE values and almost half AUC value were a lot worse than what we expected our models to perform at.
   We are curious about if the type of membership (as we filtered by only the Single Trip memberships) would have an effect on the relationship between variables.

4. One ethical concern in our findings is that they were derived from publicly sourced data. Though this is not inherently a concern, the people who "created" these data by riding MetroBikes may not know their data is being collected and published for free use electronically. Despite there being no identifying data connected to the MetroBike observations, nowhere does it say that riders are told their data is being collected and made easily accessible by anyone—e.g., three random college students. However, our results can positively affect the community in that they reflect the Austin community's consistent use—and thus demonstrated need—for the MetroBike system. Despite the decline of its use after 2021, the data shows a regular use of MetroBikes in Austin, even post-lockdown.

Discussion

1. Our main takeaway from this analysis is that there is likely no relationship between any of the variables we explored, and that humans act very unpredictable in the MetroBike industry concerning single ride memberships.

2. Answers to research questions:
   a. We cannot accurately predict a trip's duration using a KNN regression model given the year and checkout time. The overall RMSE value of 304.3085 suggests that our predictions were on average 304 minutes off from the actual duration.
   b. A trip's year and checkout time are not good linear predictors of a trip's duration. We can conclude this because the RMSE of 323.9471 is extremely high, and the R^2 value reports that less than 1% of the variation in trip duration can be explained by year and checkout time.
3. The data did not match what we expected - although we could infer from the EDA project the regressions would not perform very well, we were surprised at how bad our predictions were. The high (300s) RMSE values and almost half AUC value were a lot worse than what we expected our models to perform at.
   We are curious about if the type of membership (as we filtered by only the Single Trip memberships) would have an effect on the relationship between variables.

4. One ethical concern in our findings is that they were derived from publicly sourced data. Though this is not inherently a concern, the people who "created" these data by riding MetroBikes may not know their data is being collected and published for free use electronically. Despite there being no identifying data connected to the MetroBike observations, nowhere does it say that riders are told their data is being collected and made easily accessible by anyone—e.g., three random college students. However, our results can positively affect the community in that they reflect the Austin community's consistent use—and thus demonstrated need—for the MetroBike system. Despite the decline of its use after 2021, the data shows a regular use of MetroBikes in Austin, even post-lockdown.

Reflection, acknowledgements, and references.

1. The most challenging aspect of the project was cleaning the dataset so that we could work with the data we needed. Although we were lucky that we did not have to deal with NA values, we did have to select the variables we wanted to predict and logic out which of the other variables would best predict them. However, we found that our predictions were not very good. Additionally, we had to use the data given to us and manipulate them into fitting the guidelines for our project.

2. This process has taught us valuable lessons about finding patterns (using regression or knn models) and conducting cross validation tests between the nuance of large data sets and analyzing the results. The data manipulation process for just about every variable we used taught us how to troubleshoot and adjust given data to answer questions with nuance beyond the data at face value.

3. First and foremost, thank you to Professor Guyot for providing us the framework and toolkit to independently conduct analyses in R. Thank you to the City of Austin for this MetroBike data, as well as Ciara, Vaishnavi, and Dustyn for their clarifying notes on codes. Additionally, as a team, we worked together on all of the questions except the ones relating to our individual research question - for example, we worked on the introduction, methods, discussion, and reflection sections together as a group, but we worked on our question for our graphs and visualizations in the results section, which are clearly labeled by our names.

4. **Citations**:

    1. "The Bike Share You Love, with a New Name and New Features!" Metrobike, austin.bcycle.com/blog/2021/01/26/metrobike-the-bike-share-you-love-with-a-new-name-and-new-features! Accessed 28 Mar. 2024.
    2. City of Austin, Texas - data.austintexas.gov. "Austin MetroBike Trips: Open Data: City of Austin Texas." Data.AustinTexas.Gov - The Official City of Austin Open Data Portal, 12 Feb. 2024, data.austintexas.gov/Transportation-and-Mobility/Austin-MetroBike-Trips/tyfh-5r8s/about_data.
    3. GfG. "Convert Date to Day of Week in R." GeeksforGeeks, GeeksforGeeks, 23 May 2021, www.geeksforgeeks.org/convert-date-to-day-of-week-in-r/.
    4. "Remove Matched Patterns - Str_remove." - Str_remove • Stringr, stringr.tidyverse.org/reference/str_remove.html. Accessed 28 Mar. 2024.
    5. robk@statmethods.net, Robert Kabacoff -. "Operators." Quick-R: Operators, www.statmethods.net/management/operators.html#:~:text=How%20can%20I%20perform%20integer,yields%20a%20quotient%20of%202. Accessed 28 Mar. 2024.