Part 4: Linear Mixed Models

Charles Julien, Chike Odenigbo, Atul Sharma, Gabriel Jobert

10/20/2023

Contents

Introduction

Enhancing Urban Mobility Through Advanced Analytics: Unraveling Patterns in BIXI Data

BIXI, the public cycling service, has emerged as a pivotal player in urban transportation, offering an accessible and eco-friendly mode of transportation that has reshaped urban mobility. Our commitment to understanding and

improving urban transportation systems has led our consultant team to conduct an extensive analysis of BIXI's operational data.

This report builds upon our previous exploration of BIXI's data, focusing on the application of linear mixed models to uncover nuanced insights. Our objective is to provide a comprehensive analysis of factors influencing BIXI's performance, extending our investigation to three specific research questions (RQs). These RQs delve into the impact of meteorological conditions, temporal patterns, and user classifications on BIXI's revenue generation.

In this journey, we leverage advanced statistical techniques, particularly linear mixed models, to unravel complex relationships within the dataset. R, a powerful statistical tool, serves as our primary instrument for data analysis and modeling. Our findings aim to not only deepen the understanding of BIXI's dynamics but also provide actionable insights for enhancing operational efficiency.

The central RQs explored in this report include the assessment of the seasonal impact on revenue, understanding the temporal patterns affecting trip duration, and examining the influence of user classifications on BIXI's performance. By addressing these questions, we aim to contribute valuable insights that can inform strategic decision-making for BIXI and serve as a reference for urban planners, researchers, and policymakers committed to creating sustainable and enjoyable urban environments.

The subsequent sections of this report will delve into the methodologies employed, share the findings derived from our analysis, and offer recommendations to support BIXI in continually improving its services.

Business/Research questions

- Research Question 1: How do seasonal factors impact trip revenue for BIXI Montréal?
- Research Question 2: How do daily and weekly patterns impact trip durations for BIXI Montréal?
- Research Question 3: What variables impact the average bixi trip duration?

Before jumping in, let's perform a quick exploration of our data.

```
df_explore = df_main %>%
  group_by(station, mem) %>%
  summarize(n = n())
```

```
## `summarise()` has grouped output by 'station'. You can override using the
## `.groups` argument.
```

```
summary(df_explore$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   6.000   6.545   9.000  17.000
```

The unique identifier of a line in our dataset is a combination of the station, the date and the membership status. On average, a station for a given membership status appears 6 times in our dataset.

Research Question 1: How do seasonal factors impact trip revenue for BIXI Montréal?

ADD INTERACTION TERM **Objective of Analysis:** This regression model is examining the impact of the month (`mm`), average daily temperature (`temp`), and total amount of rainfall (`rain`) and membershipt (`mem`) on the revenue (`rev`) generated by trips leaving from a specified station.

Model Linear regression

```
##
## Call:
## lm(formula = rev ~ mm + temp + rain + mem, data = df_main)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6946  -1388   -475    838  47246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2091.216    172.311 -12.136  < 2e-16 ***
## mm5          1337.321    172.504   7.752 9.90e-15 ***
## mm6          1557.546    197.900   7.870 3.90e-15 ***
## mm7          1593.946    191.697   8.315  < 2e-16 ***
## mm8          1391.546    209.516   6.642 3.26e-11 ***
## mm9          1894.608    177.695  10.662  < 2e-16 ***
## mm10          816.977    162.684   5.022 5.21e-07 ***
## mm11          516.539    192.773   2.680  0.00738 **
## temp           57.589      9.949   5.789 7.31e-09 ***
## rain          -72.689      7.029 -10.341  < 2e-16 ***
## mem1         6134.595     72.237  84.923  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3599 on 9989 degrees of freedom
## Multiple R-squared:  0.4344, Adjusted R-squared:  0.4339
## F-statistic: 767.3 on 10 and 9989 DF,  p-value: < 2.2e-16
```

Model Linear Mixed model (ATUL)

Comment: compound symmetric correlation structure is not ideal for time series if I am not mistaken

```
seasonal_effect_rev_gls <- gls(rev ~ temp + rain + mm + mem, correlation = corCompSymm(form = ~ 1 | statio

# Display model summary
summary(seasonal_effect_rev_gls)
```

```
## Generalized least squares fit by REML
##   Model: rev ~ temp + rain + mm + mem
##   Data: df_main
##        AIC      BIC    logLik
##   189095.7 189189.4 -94534.86
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | station
##  Parameter estimate(s):
##       Rho
## 0.3714602
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) -2637.297 164.09836 -16.07144   0e+00
## temp           63.102   8.20592   7.68982   0e+00
## rain          -83.319   5.79387 -14.38050   0e+00
## mm5          1241.513 142.33492   8.72248   0e+00
## mm6          1576.754 163.95963   9.61672   0e+00
## mm7          1663.683 158.67860  10.48461   0e+00
## mm8          1488.772 173.79641   8.56619   0e+00
## mm9          2116.005 147.74396  14.32211   0e+00
## mm10         1112.919 136.12558   8.17568   0e+00
## mm11          648.125 159.95436   4.05194   1e-04
## mem1         6326.879  59.77901 105.83779   0e+00
##
```

```
##  Correlation:
##      (Intr) temp   rain   mm5    mm6    mm7    mm8    mm9    mm10   mm11
## temp -0.510
## rain -0.069 -0.006
## mm5  -0.382 -0.271  0.089
## mm6  -0.166 -0.557 -0.026  0.659
## mm7  -0.196 -0.530  0.019  0.671  0.757
## mm8  -0.108 -0.634  0.044  0.655  0.774  0.774
## mm9  -0.300 -0.402 -0.008  0.672  0.721  0.726  0.726
## mm10 -0.486 -0.130 -0.028  0.646  0.612  0.625  0.595  0.660
## mm11 -0.584  0.232 -0.002  0.457  0.326  0.348  0.285  0.417  0.530
## mem1 -0.233  0.012 -0.040  0.025  0.035  0.027  0.023  0.039  0.050  0.029
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.8926963 -0.3233412 -0.0298840  0.2913426 13.1127739
##
## Residual standard error: 3613.719
## Degrees of freedom: 10000 total; 9989 residual
```

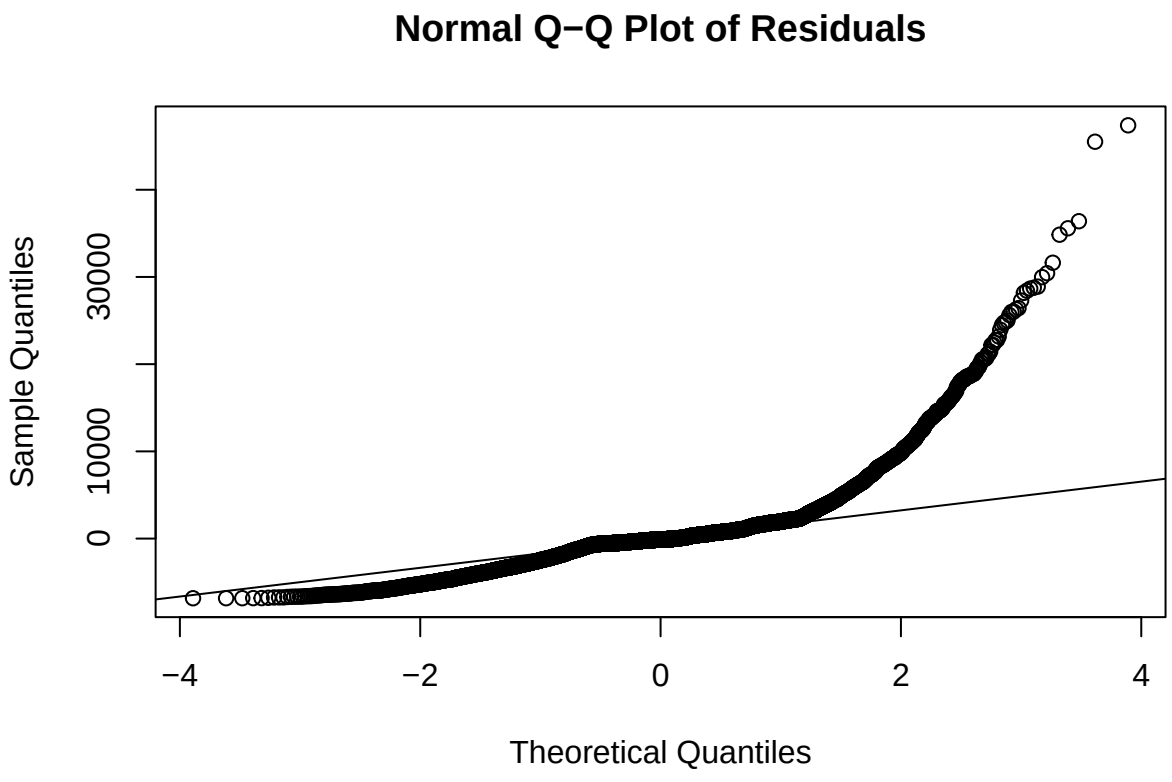
```r
# CHIKE: Note that using compound symmetric gives the same covariance to each observation
cov.matrix = getVarCov(seasonal_effect_rev_gls,individual = 1)
cov2cor(cov.matrix)
```

```
## Marginal variance covariance matrix
##          [,1]    [,2]    [,3]    [,4]    [,5]    [,6]    [,7]    [,8]    [,9]
##  [1,] 1.00000 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [2,] 0.37146 1.00000 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [3,] 0.37146 0.37146 1.00000 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [4,] 0.37146 0.37146 0.37146 1.00000 0.37146 0.37146 0.37146 0.37146 0.37146
##  [5,] 0.37146 0.37146 0.37146 0.37146 1.00000 0.37146 0.37146 0.37146 0.37146
##  [6,] 0.37146 0.37146 0.37146 0.37146 0.37146 1.00000 0.37146 0.37146 0.37146
##  [7,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 1.00000 0.37146 0.37146
##  [8,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 1.00000 0.37146
##  [9,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 1.00000
## [10,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [11,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [12,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [13,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [14,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [15,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [16,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##         [,10]   [,11]   [,12]   [,13]   [,14]   [,15]   [,16]
##  [1,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [2,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [3,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [4,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [5,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [6,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [7,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [8,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
##  [9,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [10,] 1.00000 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146
## [11,] 0.37146 1.00000 0.37146 0.37146 0.37146 0.37146 0.37146
## [12,] 0.37146 0.37146 1.00000 0.37146 0.37146 0.37146 0.37146
## [13,] 0.37146 0.37146 0.37146 1.00000 0.37146 0.37146 0.37146
## [14,] 0.37146 0.37146 0.37146 0.37146 1.00000 0.37146 0.37146
## [15,] 0.37146 0.37146 0.37146 0.37146 0.37146 1.00000 0.37146
## [16,] 0.37146 0.37146 0.37146 0.37146 0.37146 0.37146 1.00000
##   Standard Deviations: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```r
coefficients_and_pvalues <- coef(summary(seasonal_effect_rev_gls))
print(coefficients_and_pvalues)
```

```
##                    Value  Std.Error    t-value      p-value
## (Intercept) -2637.29741 164.098362 -16.071443 2.115404e-57
## temp           63.10207   8.205925   7.689818 1.612369e-14
## rain          -83.31878   5.793874 -14.380496 1.991983e-46
## mm5          1241.51337 142.334924   8.722479 3.155636e-18
## mm6          1576.75420 163.959627   9.616722 8.446031e-22
## mm7          1663.68338 158.678598  10.484611 1.380125e-25
## mm8          1488.77225 173.796405   8.566186 1.227668e-17
## mm9          2116.00461 147.743955  14.322106 4.545430e-46
## mm10         1112.91857 136.125584   8.175675 3.299495e-16
## mm11          648.12506 159.954359   4.051937 5.117986e-05
## mem1         6326.87864  59.779011 105.837793 0.000000e+00
```

```r
# Extract residuals from the GLS model
residuals_gls <- residuals(seasonal_effect_rev_gls)

# Create a Normal Q-Q plot
qqnorm(residuals_gls, main = "Normal Q-Q Plot of Residuals")
qqline(residuals_gls)
```



##Strength of the model

The Normal Q-Q plot provides valuable insights into the distribution of residuals in the BIXI data model. The near-perfect alignment of residuals with the reference line from -4 to +2 suggests that a substantial portion of the residuals adheres to a normal distribution. However, the major upward deviation observed from +2 to +4 indicates the presence of extreme positive residuals that do not align with the expected normal distribution. This discrepancy highlights a limitation in the model, signaling the existence of outliers or influential observations that could significantly impact the model's accuracy. These outliers may be indicative of unaccounted-for factors or unexpected events that contribute to revenue variations beyond the model's current specifications. Addressing this

limitation may involve further exploration of the data to identify the sources of these extreme residuals, potential model refinements, or the consideration of alternative modeling approaches to better capture the underlying patterns in the BIXI revenue data.

Interpretation

Intercept (14.27): The expected revenue in April is, on average, 14.27 $ when temperature (temp) is zero, there is not rain). It is difficult to interpret at practically, temperature would not be zero in April.

Temperature (0.15): A one-unit increase in temperature is associated with a 0.15 $ increase, on average, in revenue, keeping other variables constant. Higher temperatures are positively correlated with increased revenue.

Rainfall (-0.22): Rainfall leads a 0.22 unit decrease in revenue keeping other variables constant. Higher rainfall is negatively correlated with revenue, suggesting potential negative effects on Bixi usage.

May (mm5 - 2.44): Revenue is expected to increase by 2.44 $, on average, in May compared to April (reference month) keeping other variables constant. May is associated with higher revenue compared to April.

June (mm6 - 6.99): Revenue is expected to increase by 6.99 $, on average, in June compared to April keeping other variables constant. June has a significant positive impact on revenue.

July (mm7 - 13.73): Revenue is expected to increase by 13.73 $, on average, in July compared to April keeping other variables constant. Interpretation: July has the most significant positive impact on revenue among the months.

August (mm8 - 14.69): Revenue is expected to increase by 14.69 $, on average, in August compared to April keeping other variables constant. Interpretation: August has a substantial positive impact on revenue.

September (mm9 - 15.82): Revenue is expected to increase by 15.82 $, on average, in September compared to April keeping other variables constant. September has substantial positive impact on revenue.

October (mm10 - 9.19): Revenue is expected to increase by 9.19 $, on average, in October compared to April keeping other variables constant. Interpretation: October has a positive impact on revenue.

November (mm11 - 3.29): Revenue is expected to increase by 3.29 $, on average, in November compared to April keeping other variables constant. November has a modest positive impact on revenue.

##Business Implications:

Temperature: Bixi can capitalize on warmer temperatures by promoting increased ridership during favorable weather conditions.

Rainfall: Strategies to mitigate the negative impact of rainfall on revenue may include targeted marketing during rainy periods or offering promotions to incentivize usage.

Seasonal Variation: Understanding the seasonal variation allows Bixi to allocate resources effectively, focusing on peak months like July, August, and September for marketing and service enhancements.

Month-specific Strategies: Tailoring marketing campaigns or promotional offers based on the impact of each month on revenue can optimize Bixi's overall financial performance.

Planning and Resource Allocation: Knowledge of specific months with higher revenue can guide resource allocation, such as increasing bike availability and marketing efforts during peak months.

Operational Adjustments: Bixi can make operational adjustments, such as increasing staff or bikes, during months with the most significant positive impact on revenue.

Verification of Assumptions

NEED A MORE COMPLETE ASSUPTION VALIDATION ### Normality of residuals

1. **Histogram of Residuals**:
2. **Normal Q-Q Plot**:

Overall Interpretation:

Research Question 1: Autoregressive Structure (CHIKE)

Linear Mixed Models (LMMs) - Model Comparisons (joshuawiley.com)

```
#library(ggplot2)
# tous / all id
#ggplot(data = df_main, aes(x = week_num, y = rev_imputed, group = station)) +
#geom_line(alpha = 0.2) + scale_x_continuous(expand = c(0,
#0), limits = c(1, 5))
```

```
seasonal_effect_rev.ar <- gls(rev ~ temp + rain + mm + mem, correlation = corAR1(form = ~ 1 | station), da

# Display model summary
#summary(seasonal_effect_rev.ar)
```

```
getVarCov(seasonal_effect_rev.ar,individual = 1)
```

```
## Marginal variance covariance matrix
##             [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
##  [1,] 1.2853e+07 5.0241e+06 1.9639e+06 7.6769e+05   300090.0   117300.0
##  [2,] 5.0241e+06 1.2853e+07 5.0241e+06 1.9639e+06   767690.0   300090.0
##  [3,] 1.9639e+06 5.0241e+06 1.2853e+07 5.0241e+06  1963900.0   767690.0
##  [4,] 7.6769e+05 1.9639e+06 5.0241e+06 1.2853e+07  5024100.0  1963900.0
##  [5,] 3.0009e+05 7.6769e+05 1.9639e+06 5.0241e+06 12853000.0  5024100.0
##  [6,] 1.1730e+05 3.0009e+05 7.6769e+05 1.9639e+06  5024100.0 12853000.0
##  [7,] 4.5854e+04 1.1730e+05 3.0009e+05 7.6769e+05  1963900.0  5024100.0
##  [8,] 1.7924e+04 4.5854e+04 1.1730e+05 3.0009e+05   767690.0  1963900.0
##  [9,] 7.0065e+03 1.7924e+04 4.5854e+04 1.1730e+05   300090.0   767690.0
## [10,] 2.7388e+03 7.0065e+03 1.7924e+04 4.5854e+04   117300.0   300090.0
## [11,] 1.0706e+03 2.7388e+03 7.0065e+03 1.7924e+04    45854.0   117300.0
## [12,] 4.1850e+02 1.0706e+03 2.7388e+03 7.0065e+03    17924.0    45854.0
## [13,] 1.6359e+02 4.1850e+02 1.0706e+03 2.7388e+03     7006.5    17924.0
## [14,] 6.3947e+01 1.6359e+02 4.1850e+02 1.0706e+03     2738.8     7006.5
## [15,] 2.4997e+01 6.3947e+01 1.6359e+02 4.1850e+02     1070.6     2738.8
## [16,] 9.7712e+00 2.4997e+01 6.3947e+01 1.6359e+02      418.5     1070.6
##             [,7]       [,8]       [,9]      [,10]      [,11]      [,12]
##  [1,]    45854.0    17924.0     7006.5     2738.8     1070.6      418.5
##  [2,]   117300.0    45854.0    17924.0     7006.5     2738.8     1070.6
##  [3,]   300090.0   117300.0    45854.0    17924.0     7006.5     2738.8
##  [4,]   767690.0   300090.0   117300.0    45854.0    17924.0     7006.5
##  [5,]  1963900.0   767690.0   300090.0   117300.0    45854.0    17924.0
##  [6,]  5024100.0  1963900.0   767690.0   300090.0   117300.0    45854.0
##  [7,] 12853000.0  5024100.0  1963900.0   767690.0   300090.0   117300.0
##  [8,]  5024100.0 12853000.0  5024100.0  1963900.0   767690.0   300090.0
##  [9,]  1963900.0  5024100.0 12853000.0  5024100.0  1963900.0   767690.0
## [10,]   767690.0  1963900.0  5024100.0 12853000.0  5024100.0  1963900.0
## [11,]   300090.0   767690.0  1963900.0  5024100.0 12853000.0  5024100.0
## [12,]   117300.0   300090.0   767690.0  1963900.0  5024100.0 12853000.0
## [13,]    45854.0   117300.0   300090.0   767690.0  1963900.0  5024100.0
## [14,]    17924.0    45854.0   117300.0   300090.0   767690.0  1963900.0
## [15,]     7006.5    17924.0    45854.0   117300.0   300090.0   767690.0
## [16,]     2738.8     7006.5    17924.0    45854.0   117300.0   300090.0
##            [,13]      [,14]      [,15]      [,16]
##  [1,] 1.6359e+02 6.3947e+01 2.4997e+01 9.7712e+00
##  [2,] 4.1850e+02 1.6359e+02 6.3947e+01 2.4997e+01
##  [3,] 1.0706e+03 4.1850e+02 1.6359e+02 6.3947e+01
##  [4,] 2.7388e+03 1.0706e+03 4.1850e+02 1.6359e+02
##  [5,] 7.0065e+03 2.7388e+03 1.0706e+03 4.1850e+02
##  [6,] 1.7924e+04 7.0065e+03 2.7388e+03 1.0706e+03
##  [7,] 4.5854e+04 1.7924e+04 7.0065e+03 2.7388e+03
```

```
##  [8,] 1.1730e+05 4.5854e+04 1.7924e+04 7.0065e+03
##  [9,] 3.0009e+05 1.1730e+05 4.5854e+04 1.7924e+04
## [10,] 7.6769e+05 3.0009e+05 1.1730e+05 4.5854e+04
## [11,] 1.9639e+06 7.6769e+05 3.0009e+05 1.1730e+05
## [12,] 5.0241e+06 1.9639e+06 7.6769e+05 3.0009e+05
## [13,] 1.2853e+07 5.0241e+06 1.9639e+06 7.6769e+05
## [14,] 5.0241e+06 1.2853e+07 5.0241e+06 1.9639e+06
## [15,] 1.9639e+06 5.0241e+06 1.2853e+07 5.0241e+06
## [16,] 7.6769e+05 1.9639e+06 5.0241e+06 1.2853e+07
##   Standard Deviations: 3585.1 3585.1 3585.1 3585.1 3585.1 3585.1 3585.1 3585.1 3585.1 3585.1 3585.1 358
```

As expected with an auto-regressive correlation structure, the further back the time period, the lower the correlation.

It is important to note that the time distance between these observations is not constant, in some cases, it can even be null. For example, when a station is observed on the same day for members and non-members. Having a covariance that considers the time between observation would be the best. (ARH1)

Research Question 1: Random Intercept (CHIKE)

```
#df_main
seasonal_effect_rev.rand_int <- lme(rev ~ temp + rain + mm + mem, random = ~ 1 | station, data = df_main)

# Display model summary
#summary(seasonal_effect_rev.rand_int)
```

```
# Expected to be True comparing fixed effects coefficients between base model and random effects model
isTRUE(all.equal(coef(seasonal_effect_rev_model), fixef(seasonal_effect_rev.rand_int)))
```

```
## [1] FALSE
```

```
getVarCov(seasonal_effect_rev.rand_int, type = "random.effects")
```

```
## Random effects variance covariance matrix
##             (Intercept)
## (Intercept)     4850900
##   Standard Deviations: 2202.5
```

```
getVarCov(seasonal_effect_rev.rand_int, individual = 1, type = "conditional")
```

```
## station 151
## Conditional variance covariance matrix
##          1       2       3       4       5       6       7       8       9
## 1  8208100       0       0       0       0       0       0       0       0
## 2        0 8208100       0       0       0       0       0       0       0
## 3        0       0 8208100       0       0       0       0       0       0
## 4        0       0       0 8208100       0       0       0       0       0
## 5        0       0       0       0 8208100       0       0       0       0
## 6        0       0       0       0       0 8208100       0       0       0
## 7        0       0       0       0       0       0 8208100       0       0
## 8        0       0       0       0       0       0       0 8208100       0
## 9        0       0       0       0       0       0       0       0 8208100
## 10       0       0       0       0       0       0       0       0       0
## 11       0       0       0       0       0       0       0       0       0
## 12       0       0       0       0       0       0       0       0       0
## 13       0       0       0       0       0       0       0       0       0
## 14       0       0       0       0       0       0       0       0       0
## 15       0       0       0       0       0       0       0       0       0
```

```
## 16       0       0       0       0       0       0       0       0       0
## 17       0       0       0       0       0       0       0       0       0
##          10      11      12      13      14      15      16      17
## 1        0       0       0       0       0       0       0       0
## 2        0       0       0       0       0       0       0       0
## 3        0       0       0       0       0       0       0       0
## 4        0       0       0       0       0       0       0       0
## 5        0       0       0       0       0       0       0       0
## 6        0       0       0       0       0       0       0       0
## 7        0       0       0       0       0       0       0       0
## 8        0       0       0       0       0       0       0       0
## 9        0       0       0       0       0       0       0       0
## 10 8208100       0       0       0       0       0       0       0
## 11       0 8208100       0       0       0       0       0       0
## 12       0       0 8208100       0       0       0       0       0
## 13       0       0       0 8208100       0       0       0       0
## 14       0       0       0       0 8208100       0       0       0
## 15       0       0       0       0       0 8208100       0       0
## 16       0       0       0       0       0       0 8208100       0
## 17       0       0       0       0       0       0       0 8208100
##   Standard Deviations: 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865 2865
```

This is the same as looking at the variance of the error term

```
seasonal_effect_rev.ar_rand_int <- lme(rev ~ temp + rain + mm + mem, random = ~1 |
station, correlation = corAR1(form = ~1 | station), data = df_main)
#summary(seasonal_effect_rev.ar_rand_int)
```

```
getVarCov(seasonal_effect_rev.ar_rand_int, type = "random.effects")
```

```
## Random effects variance covariance matrix
##             (Intercept)
## (Intercept)     4742600
##   Standard Deviations: 2177.7
```

```
AIC(seasonal_effect_rev.ar, seasonal_effect_rev.rand_int, seasonal_effect_rev.ar_rand_int )
```

```
##                                df      AIC
## seasonal_effect_rev.ar         13 190445.2
## seasonal_effect_rev.rand_int   13 189095.7
## seasonal_effect_rev.ar_rand_int 14 189076.3
```

Research Question 1: Random Slope (CHIKE)

```
#Doesnt make sense in this context to use random effects on the coefficient for #temperature because tempe
#Using rain random effect does not work
seasonal_effect_rev.rand_coef <- lme(rev ~ temp + rain + mm + mem, random = ~1 + temp |
station, data = df_main)
#summary(seasonal_effect_rev.rand_coef)
```

```
summary_season_rev.base = tidy(seasonal_effect_rev_model)
colnames(summary_season_rev.base) <- c('Covariates','Value','Std.Error','t-value','p-value')
summary_season_rev.base$type = 'Base'
#summary_season_rev.base
```

```r
summary_season_rev.ar = as.data.frame(summary(seasonal_effect_rev.ar)$tTable)
summary_season_rev.ar$type = 'Autoregressive'
#summary_season_rev.ar[,c('DF')] <- NA
summary_season_rev.ar = tibble::rownames_to_column(summary_season_rev.ar, "Covariates")

summary_season_rev.rand_int = as.data.frame(summary(seasonal_effect_rev.rand_int)$tTable)
summary_season_rev.rand_int$type = 'Random Intercept'
summary_season_rev.rand_int = tibble::rownames_to_column(summary_season_rev.rand_int, "Covariates")

summary_season_rev.ar_rand_int = as.data.frame(summary(seasonal_effect_rev.ar_rand_int)$tTable)
summary_season_rev.ar_rand_int$type = 'Autoregressive Random Intercept'
summary_season_rev.ar_rand_int  = tibble::rownames_to_column(summary_season_rev.ar_rand_int, "Covariates")


summary_season_rev.rand_coef = as.data.frame(summary(seasonal_effect_rev.rand_coef)$tTable)
summary_season_rev.rand_coef$type = 'Random Slope'
summary_season_rev.rand_coef = tibble::rownames_to_column(summary_season_rev.rand_coef, "Covariates")


cols = intersect(colnames(summary_season_rev.rand_coef),colnames(summary_season_rev.ar))
cols = intersect(colnames(summary_season_rev.rand_int),cols)
cols = intersect(colnames(summary_season_rev.rand_coef),cols)

season_summary_combined = rbind(summary_season_rev.rand_coef[, cols], summary_season_rev.ar_rand_int[, cols
season_summary_combined$significance <- ifelse(season_summary_combined$`p-value`<0.05, 'Significant Featur
season_summary_combined
```

```
##     Covariates       Value  Std.Error    t-value      p-value
## 1  (Intercept) -2385.93546 140.750505 -16.951523 1.716713e-63
## 2         temp    52.76834   8.739949   6.037603 1.624756e-09
## 3         rain   -82.57763   5.672005 -14.558807 1.729031e-47
## 4          mm5  1212.60484 137.830718   8.797784 1.647604e-18
## 5          mm6  1503.51146 159.936145   9.400698 6.725008e-21
## 6          mm7  1606.60417 154.709370  10.384660 4.008199e-25
## 7          mm8  1466.18731 169.981206   8.625585 7.435760e-18
## 8          mm9  2073.03247 143.448074  14.451449 7.981719e-47
## 9         mm10  1040.35110 130.984985   7.942522 2.213858e-15
## 10        mm11   624.36301 153.529335   4.066734 4.807421e-05
## 11        mem1  6273.54442  58.695639 106.882633 0.000000e+00
## 12 (Intercept) -2627.37777 165.833510 -15.843467 8.501096e-56
## 13        temp    63.26372   8.211050   7.704706 1.447312e-14
## 14        rain   -83.49546   5.784249 -14.434971 1.008425e-46
## 15         mm5  1237.85049 144.331930   8.576415 1.137439e-17
## 16         mm6  1565.98579 166.524048   9.403962 6.521505e-21
## 17         mm7  1652.72300 161.437902  10.237515 1.821183e-24
## 18         mm8  1479.67486 176.140990   8.400514 5.105513e-17
## 19         mm9  2106.53322 150.746022  13.974055 6.303824e-44
## 20        mm10  1094.65769 139.674606   7.837199 5.119176e-15
## 21        mm11   646.24630 163.451045   3.953761 7.750894e-05
## 22        mem1  6330.68253  59.648150 106.133762 0.000000e+00
## 23 (Intercept) -2637.29747 164.098368 -16.071442 2.432939e-57
## 24        temp    63.10207   8.205925   7.689818 1.624873e-14
## 25        rain   -83.31878   5.793874 -14.380496 2.180550e-46
## 26         mm5  1241.51337 142.334922   8.722479 3.195863e-18
## 27         mm6  1576.75420 163.959625   9.616722 8.604512e-22
## 28         mm7  1663.68339 158.678596  10.484611 1.416642e-25
## 29         mm8  1488.77226 173.796403   8.566186 1.242238e-17
## 30         mm9  2116.00463 147.743954  14.322106 4.968588e-46
## 31        mm10  1112.91859 136.125582   8.175676 3.332063e-16
## 32        mm11   648.12508 159.954357   4.051938 5.121304e-05
```

```
## 33        mem1  6326.87866  59.779011 105.837795 0.000000e+00
## 34 (Intercept) -2199.24008 178.509104 -12.320044 1.259829e-34
## 35        temp    62.41842   8.764234   7.121949 1.137494e-12
## 36        rain   -81.13149   6.094186 -13.312933 4.271517e-40
## 37         mm5  1199.65268 170.037028   7.055244 1.837355e-12
## 38         mm6  1426.12266 200.036853   7.129300 1.078665e-12
## 39         mm7  1473.50733 198.476968   7.424072 1.228457e-13
## 40         mm8  1279.11419 211.298814   6.053580 1.467576e-09
## 41         mm9  1834.18757 190.196631   9.643639 6.515342e-22
## 42        mm10   718.12985 182.821823   3.928031 8.622045e-05
## 43        mm11   477.35754 208.271800   2.291993 2.192674e-02
## 44        mem1  6322.20401  62.456721 101.225359 0.000000e+00
## 45 (Intercept) -2091.21624 172.310505 -12.136325 1.169320e-33
## 46         mm5  1337.32128 172.504269   7.752395 9.896527e-15
## 47         mm6  1557.54574 197.899937   7.870370 3.902745e-15
## 48         mm7  1593.94560 191.696937   8.314925 1.037763e-16
## 49         mm8  1391.54625 209.515869   6.641722 3.261758e-11
## 50         mm9  1894.60785 177.695143  10.662125 2.121869e-26
## 51        mm10   816.97719 162.683774   5.021873 5.205436e-07
## 52        mm11   516.53901 192.773138   2.679518 7.384918e-03
## 53        temp    57.58933   9.948831   5.788552 7.313525e-09
## 54        rain   -72.68885   7.029208 -10.340973 6.142237e-25
## 55        mem1  6134.59497  72.237410  84.922687 0.000000e+00
##                              type        significance
## 1                     Random Slope Significant Feature
## 2                     Random Slope Significant Feature
## 3                     Random Slope Significant Feature
## 4                     Random Slope Significant Feature
## 5                     Random Slope Significant Feature
## 6                     Random Slope Significant Feature
## 7                     Random Slope Significant Feature
## 8                     Random Slope Significant Feature
## 9                     Random Slope Significant Feature
## 10                    Random Slope Significant Feature
## 11                    Random Slope Significant Feature
## 12 Autoregressive Random Intercept Significant Feature
## 13 Autoregressive Random Intercept Significant Feature
## 14 Autoregressive Random Intercept Significant Feature
## 15 Autoregressive Random Intercept Significant Feature
## 16 Autoregressive Random Intercept Significant Feature
## 17 Autoregressive Random Intercept Significant Feature
## 18 Autoregressive Random Intercept Significant Feature
## 19 Autoregressive Random Intercept Significant Feature
## 20 Autoregressive Random Intercept Significant Feature
## 21 Autoregressive Random Intercept Significant Feature
## 22 Autoregressive Random Intercept Significant Feature
## 23                Random Intercept Significant Feature
## 24                Random Intercept Significant Feature
## 25                Random Intercept Significant Feature
## 26                Random Intercept Significant Feature
## 27                Random Intercept Significant Feature
## 28                Random Intercept Significant Feature
## 29                Random Intercept Significant Feature
## 30                Random Intercept Significant Feature
## 31                Random Intercept Significant Feature
## 32                Random Intercept Significant Feature
## 33                Random Intercept Significant Feature
## 34                  Autoregressive Significant Feature
## 35                  Autoregressive Significant Feature
## 36                  Autoregressive Significant Feature
## 37                  Autoregressive Significant Feature
```

```
## 38                 Autoregressive Significant Feature
## 39                 Autoregressive Significant Feature
## 40                 Autoregressive Significant Feature
## 41                 Autoregressive Significant Feature
## 42                 Autoregressive Significant Feature
## 43                 Autoregressive Significant Feature
## 44                 Autoregressive Significant Feature
## 45                           Base Significant Feature
## 46                           Base Significant Feature
## 47                           Base Significant Feature
## 48                           Base Significant Feature
## 49                           Base Significant Feature
## 50                           Base Significant Feature
## 51                           Base Significant Feature
## 52                           Base Significant Feature
## 53                           Base Significant Feature
## 54                           Base Significant Feature
## 55                           Base Significant Feature
```

Using a 5% significance threshold, we can conclude that each of the covariates used to predict revenue had a significant impact.

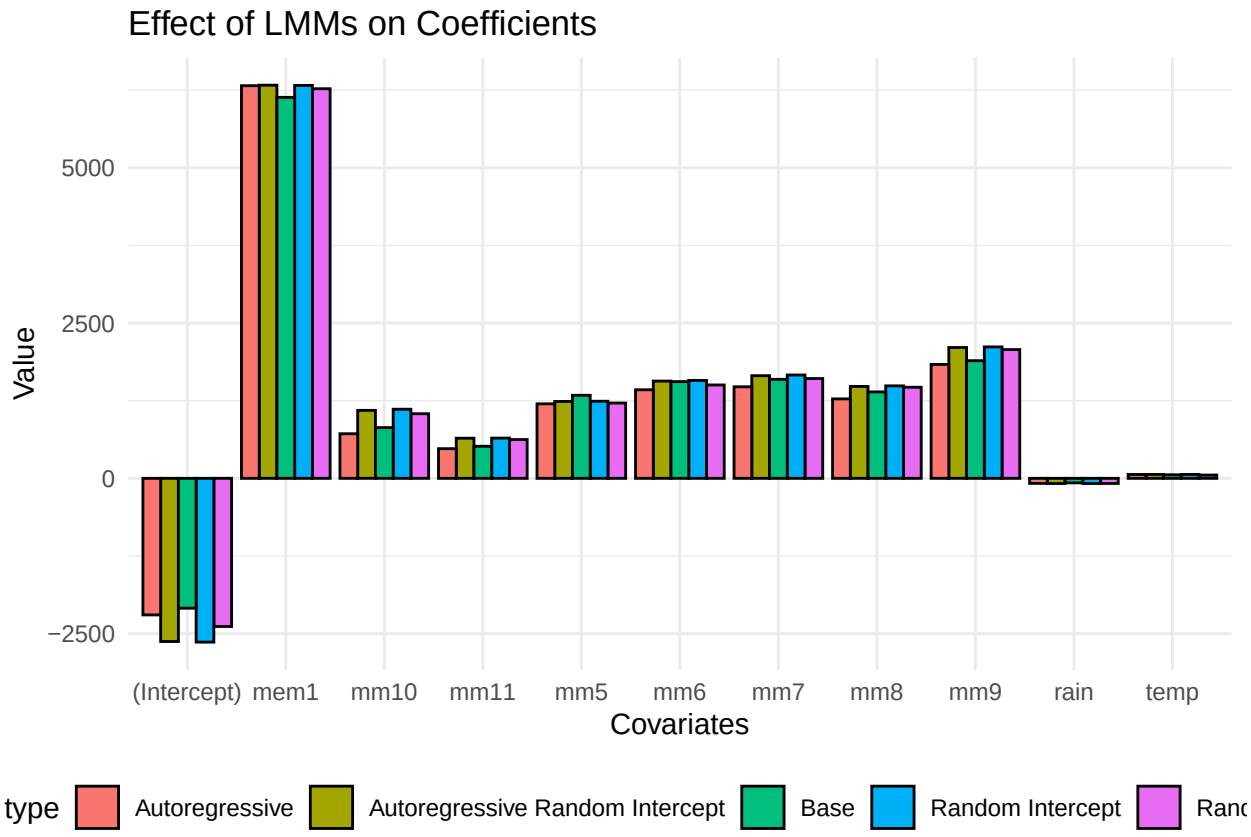
```
knitr::kable(season_summary_combined %>%
  group_by(Covariates) %>%
  summarise(significant = sum(significance == "Significant Feature"),
            not_significant = sum(significance == "Not Significant")),caption = 'Feature Significance (5%)
```

Table 1: Feature Significance (5%)

Covariates	significant	not_significant
(Intercept)	5	0
mem1	5	0
mm10	5	0
mm11	5	0
mm5	5	0
mm6	5	0
mm7	5	0
mm8	5	0
mm9	5	0
rain	5	0
temp	5	0

The Model estimates are similar despite different correlation structures. In this sense, we can see that the size and direction of each covariate is similar across each model.

```
ggplot(season_summary_combined, aes(fill=type, y=Value, x=Covariates)) +
  geom_bar(colour="black",position='dodge', stat='identity')  + ggtitle ("Effect of LMMs on Coefficients")
```



###Interpretation * Intercept: This is the average revenue when all values are set at 0. In our case, it would be that for non members, in the month of April, with no rain and temperature at 0 degrees, the expected revenue is roughly -($2,500) across all the models. This number is unrealistic as Bixi revenue is a strictly positive number. It would have been more interpretable if revenue was allowed to be a negative number by accounting for cost.

- mem1: Members contribute roughly $6,000 in additional revenue for a given station compared to non-members holding other variables constant.
- mm5: Rides in the month of May contribute about $1,200 in additional revenue for a given station compared to the month of April holding all other variables constant.
- mm6: Rides in the month of June contribute about $1,250 in additional revenue for a given station compared to the month of April holding all other variables constant.
- mm7: Rides in the month of July contribute about $1,280 in additional revenue for a given station compared to the month of April holding all other variables constant.
- mm8: Rides in the month of August contribute about $1,250 in additional revenue for a given station compared to the month of April holding all other variables constant.
- mm9: Rides in the month of May contribute about $2,200 in additional revenue for a given station compared to the month of April holding all other variables constant.
- mm10: Rides in the month of May contribute about $1,000 in additional revenue for a given station compared to the month of April holding all other variables constant.
- mm11: Rides in the month of May contribute about $700 in additional revenue for a given station compared to the month of April holding all other variables constant.
- rain: A 1 unit increase in rain contributes to a $100 decrease in revenue for a given station holding all other variables constant.
- temp: A 1 unit increase in temperature contributes to a $100 increase in revenue for a given station holding all other variables constant.

Using AIC and BIC metrics, the model incorporating a random slope as well as the model incorporating an autoregressive correlation structure with a random intercept perform best when predicting revenue using season, membership and period data. This was assessed by the fact that they both have the lowest BIC and AIC metrics of all the models considered. It is also worth noting that the basic linear model that does not account for autocorrelation in the data fits the data the least optimally.

```
performance.df <- data.frame("Type"=numeric(),"AIC"=numeric(),"BIC"=numeric(),"LL"=numeric())
performance.df[nrow(performance.df) + 1,] = c("Base",AIC(seasonal_effect_rev_model),BIC(seasonal_effect_re

performance.df[nrow(performance.df) + 1,] = c("Autoregressive",AIC(seasonal_effect_rev.ar),BIC(seasonal_ef

performance.df[nrow(performance.df) + 1,] = c("Autoregressive Random Intercept",AIC(seasonal_effect_rev.ar

performance.df[nrow(performance.df) + 1,] = c("Random Intercept",AIC(seasonal_effect_rev.rand_int),BIC(sea

performance.df[nrow(performance.df) + 1,] = c("Random Slope",AIC(seasonal_effect_rev.rand_coef),BIC(season

knitr::kable(performance.df %>% arrange(BIC),caption='Model Performance')
```

Table 2: Model Performance

Type	AIC	BIC	LL
Random Slope	188783.341277982	188891.47987448	-94376.6706389911
Autoregressive Random Intercept	189076.260516278	189177.189873009	-94524.130258139
Random Intercept	189095.718549919	189189.438666884	-94534.8592749593
Autoregressive	190445.22721716	190538.947334125	-95209.61360858
Base	192159.782535936	192246.3066204	-96067.891267968

```
anova(seasonal_effect_rev.rand_coef,seasonal_effect_rev.rand_int)
```

```
##                               Model df      AIC      BIC    logLik   Test
## seasonal_effect_rev.rand_coef     1 15 188783.3 188891.5 -94376.67
## seasonal_effect_rev.rand_int      2 13 189095.7 189189.4 -94534.86 1 vs 2
##                                L.Ratio p-value
## seasonal_effect_rev.rand_coef
## seasonal_effect_rev.rand_int  316.3773  <.0001
```

```
# Testing to see if anova command produces same result for LRT test
#D <- -2 * (seasonal_effect_rev.rand_int$logLik - seasonal_effect_rev.rand_coef$logLik)
#print(D)
#pchisq(D, df = 13, lower.tail = FALSE)/2
```

We further performed likelihood ratio tests between each of the linear mixed models and the base linear model in order to assess whether the full model fits the data significantly better than the nested model. In our analysis, the nested model consisted simply of the linear model and the full model accounted for the addition of parameters relating to the correlation structure of random effects. In each case, we concluded that the full model performed significantly better using a 1% significance threshold. This effectively means that making changes to the model structure by accounting for autocorrelation leads to a significant improvement in fit relative to a linear model.

```
anova(seasonal_effect_rev.rand_int,seasonal_effect_rev_model)
```

```
##                               Model df      AIC      BIC    logLik   Test
## seasonal_effect_rev.rand_int      1 13 189095.7 189189.4 -94534.86
## seasonal_effect_rev_model         2 12 192048.3 192134.8 -96012.13 1 vs 2
##                                L.Ratio p-value
## seasonal_effect_rev.rand_int
## seasonal_effect_rev_model    2954.539  <.0001
```

```
anova(seasonal_effect_rev.rand_coef,seasonal_effect_rev_model)
```

```
##                               Model df      AIC      BIC    logLik   Test
## seasonal_effect_rev.rand_coef     1 15 188783.3 188891.5 -94376.67
## seasonal_effect_rev_model         2 12 192048.3 192134.8 -96012.13 1 vs 2
##                                L.Ratio p-value
## seasonal_effect_rev.rand_coef
## seasonal_effect_rev_model     3270.916  <.0001
```

```
anova(seasonal_effect_rev.ar_rand_int,seasonal_effect_rev_model)
```

```
##                                 Model df      AIC      BIC    logLik   Test
## seasonal_effect_rev.ar_rand_int     1 14 189076.3 189177.2 -94524.13
## seasonal_effect_rev_model           2 12 192048.3 192134.8 -96012.13 1 vs 2
##                                  L.Ratio p-value
## seasonal_effect_rev.ar_rand_int
## seasonal_effect_rev_model       2975.997  <.0001
```

```
anova(seasonal_effect_rev.ar,seasonal_effect_rev_model)
```

```
##                           Model df      AIC      BIC    logLik   Test L.Ratio
## seasonal_effect_rev.ar        1 13 190445.2 190539.0 -95209.61
## seasonal_effect_rev_model     2 12 192048.3 192134.8 -96012.13 1 vs 2 1605.03
##                           p-value
## seasonal_effect_rev.ar
## seasonal_effect_rev_model  <.0001
```

```
getVarCov(seasonal_effect_rev.rand_coef, type = "random.effects")
```

```
## Random effects variance covariance matrix
##             (Intercept)  temp
## (Intercept)      485560 59384
## temp              59384  8691
##   Standard Deviations: 696.82 93.226
```

Using information criterion and likelihood ratio tests, we compared the 4 linear mixed models together.

```
anova(seasonal_effect_rev.ar, seasonal_effect_rev.ar_rand_int,seasonal_effect_rev.rand_coef,seasonal_effec
```

```
##                                 Model df      AIC      BIC    logLik   Test
## seasonal_effect_rev.ar              1 13 190445.2 190539.0 -95209.61
## seasonal_effect_rev.ar_rand_int     2 14 189076.3 189177.2 -94524.13 1 vs 2
## seasonal_effect_rev.rand_coef       3 15 188783.3 188891.5 -94376.67 2 vs 3
## seasonal_effect_rev_gls             4 13 189095.7 189189.4 -94534.86 3 vs 4
##                                   L.Ratio p-value
## seasonal_effect_rev.ar
## seasonal_effect_rev.ar_rand_int 1370.9667  <.0001
## seasonal_effect_rev.rand_coef    294.9192  <.0001
## seasonal_effect_rev_gls          316.3773  <.0001
```

```
AIC(seasonal_effect_rev_model)
```

```
## [1] 192159.8
```

```
#anova(seasonal_effect_rev_model,seasonal_effect_rev.ar)
```

```
anova(seasonal_effect_rev.rand_coef,seasonal_effect_rev_gls)
```

```
##                               Model df      AIC      BIC    logLik   Test
## seasonal_effect_rev.rand_coef     1 15 188783.3 188891.5 -94376.67
## seasonal_effect_rev_gls           2 13 189095.7 189189.4 -94534.86 1 vs 2
##                                L.Ratio p-value
## seasonal_effect_rev.rand_coef
## seasonal_effect_rev_gls       316.3773  <.0001
```

```
lrtest(seasonal_effect_rev.ar,seasonal_effect_rev.rand_coef)
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "gls", updated model is of class "lme"

## Likelihood ratio test
##
## Model 1: rev ~ temp + rain + mm + mem
## Model 2: rev ~ temp + rain + mm + mem
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1  13 -95210
## 2  15 -94377  2 1665.9  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(seasonal_effect_rev_gls,seasonal_effect_rev.rand_coef)
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "gls", updated model is of class "lme"

## Likelihood ratio test
##
## Model 1: rev ~ temp + rain + mm + mem
## Model 2: rev ~ temp + rain + mm + mem
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1  13 -94535
## 2  15 -94377  2 316.38  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(seasonal_effect_rev_gls,seasonal_effect_rev.ar_rand_int)
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): original model was of
## class "gls", updated model is of class "lme"

## Likelihood ratio test
##
## Model 1: rev ~ temp + rain + mm + mem
## Model 2: rev ~ temp + rain + mm + mem
##   #Df LogLik Df  Chisq Pr(>Chisq)
## 1  13 -94535
## 2  14 -94524  1 21.458  3.617e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(seasonal_effect_rev.ar)
```

```
## [1] 190445.2
```

```
nobs(seasonal_effect_rev_model)
```

```
## [1] 10000
```

```
nobs(seasonal_effect_rev.ar_rand_int)
```

```
## [1] 10000
```

```
#widyr::pairwise_cor(df_main, station, rain, c)
#df_main %>% select(station:rain) %>% modelsummary::datasummary_correlation()
```

Research Question 2: How do daily and weekly patterns impact trip durations for BIXI Montréal?

Objective of Analysis: This regression model is examining the impact of the day of the month (`dd`), day of the week (`wday`), and holidays (`holiday`) on the revenue (`rev`) generated by trips leaving from a specified station.

Model

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: dur ~ holiday + wknd_ind + wknd_ind * mem + (1 | district/station)
##    Data: df_main
##
## REML criterion at convergence: 136021.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -7.3968 -0.4977 -0.0481  0.3757 15.6285
##
## Random effects:
##  Groups           Name        Variance Std.Dev.
##  station:district (Intercept) 23795    154.3
##  district         (Intercept) 12617    112.3
##  Residual                     40235    200.6
## Number of obs: 10000, groups:  station:district, 793; district, 21
##
## Fixed effects:
##                       Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)           -34.453     26.802   20.666  -1.285 0.212854
## holiday1               51.945     13.936 9433.782   3.727 0.000195 ***
## wknd_indWeekend        67.896      6.607 9419.592  10.276  < 2e-16 ***
## mem1                  313.065      4.949 9436.954  63.261  < 2e-16 ***
## wknd_indWeekend:mem1  -68.763      9.137 9429.454  -7.526 5.73e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) holdy1 wknd_W mem1
## holiday1    -0.019
## wknd_ndWknd -0.076  0.071
## mem1        -0.110  0.011  0.396
## wknd_ndWk:1  0.054 -0.006 -0.718 -0.536
```

The model explores the relationship between trip duration (`dur`) and factors like holidays (`holiday`), weekend indicator (`wknd_ind`), membership status (`mem`) and its interaction with weekend indicator, considering the nested structure of stations within districts.

Random Effects - Station:District Variability: The significant variance in the random intercepts for stations within districts (Variance = 23,795, Std. Dev. = 154.3) suggests considerable differences in baseline trip durations across stations, depending on their district. - **District-Level Variability**: There is also notable variability between districts (Variance = 12,617, Std. Dev. = 112.3), indicating that the district a station belongs to influences trip duration. - These results highlight the importance of accounting for the hierarchical structure of the data (stations nested within districts).

Fixed Effects (Significance & Interpretation) - Intercept: The negative intercept (-34.453) may not be meaningful by itself, as it represents the expected trip duration when all other variables are at their reference levels. In this context, an intercept of -34.453 would mean that when it's a non-holiday weekday, and the rider is not a member, the model predicts a trip duration of -34.453 units. Since negative trip duration is not possible, this result might initially seem nonsensical. - **Holiday (significant)**: On average, holding other variables constant, total trip durations on holidays are 51.945 minutes longer compared to non-holidays. This reflects a tendency for longer trips during holidays. This effect is statistically significant ($p < 0.001$). - **Weekend Indicator (significant)**: On average, with other factors held constant, total trip durations on weekends are 67.896 minutes longer than on weekdays. This indicates a preference or tendency for longer trips during weekends. This is highly significant ($p < 0.001$). - **Membership Status (significant)**: Holding other variables at their reference levels, on average, members have a total trip durations that are 313.065 minutes longer compared to non-members. This might indicate different usage patterns, such as members taking longer trips., a highly significant effect ($p < 0.001$). - **Interaction: Weekend and Membership (significant)**: On average, and with other variables held constant, the interaction effect suggests that the increased total trip duration associated with membership is reduced by 68.763 minutes on weekends. This indicates that the distinction in trip duration between members and non-members is less pronounced on weekends. This is also statistically significant ($p < 0.001$).

Correlations of Fixed Effects - The correlation matrix shows the relationships between the different fixed effects in the model. High correlations can indicate potential multicollinearity issues, which might affect the interpretation of coefficients. However, in the model, these correlations seem relatively moderate.

Overall Interpretation - The model indicates that both the day of the week (weekend vs. weekday) and membership status significantly impact trip durations, with an interesting interaction effect on weekends for members. - The significant random effects imply that both the specific station and the district it's in are important factors influencing trip durations. - The model appears to be a good fit for the data, capturing key variability both within and between groups (stations and districts).

```
null_model <- lmer(dur ~ 1 + (1 | district/station), data = df_main)

#Compare full model to null model (refitting using mle)
anova(time_pattern_dur_model_mixed, null_model)
```

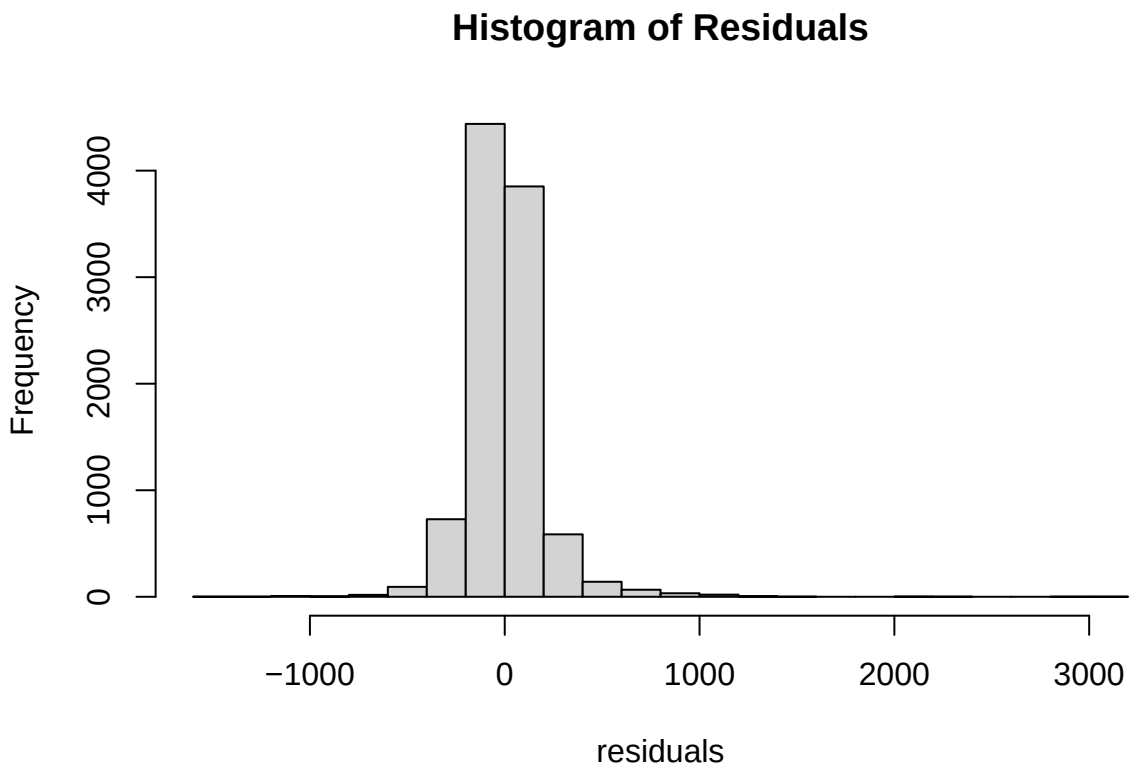
```
## refitting model(s) with ML (instead of REML)

## Data: df_main
## Models:
## null_model: dur ~ 1 + (1 | district/station)
## time_pattern_dur_model_mixed: dur ~ holiday + wknd_ind + wknd_ind * mem + (1 | district/station)
##                              npar    AIC    BIC logLik deviance  Chisq Df
## null_model                      4 140073 140102 -70032   140065
## time_pattern_dur_model_mixed    8 136069 136127 -68027   136053 4011.6  4
##                              Pr(>Chisq)
## null_model
## time_pattern_dur_model_mixed  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- The significant Chi-square test ($p < 0.001$) suggests that the fixed effects included in the full model (related to holidays, weekends, and membership status) contribute meaningfully to explaining the variability in trip durations.

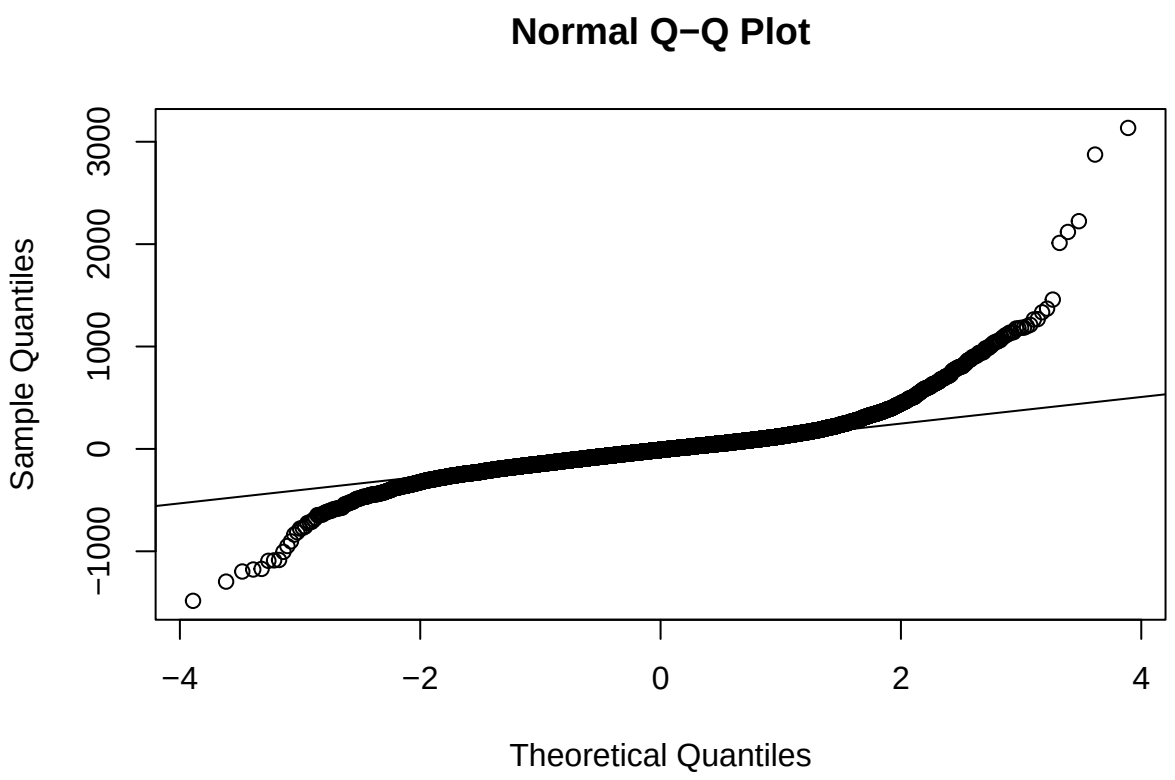
- The lower AIC and BIC values for the full model compared to the null model further support that the full model provides a better fit to the data.
- This analysis strongly indicates that the factors of holidays, weekends, and membership status, along with their interactions, are important predictors of trip duration in the context of the bike-sharing data.

Assumptions

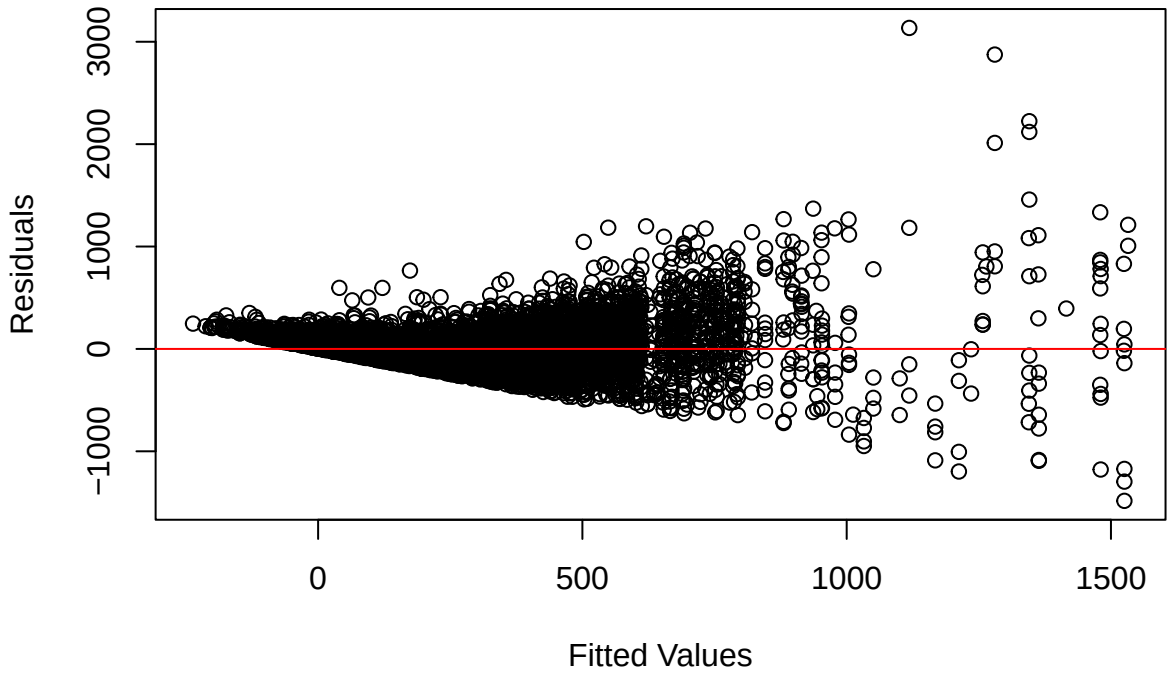
```
# Using lmer model
residuals <- residuals(time_pattern_dur_model_mixed)

# Histogram
hist(residuals, breaks=30, main="Histogram of Residuals")
```



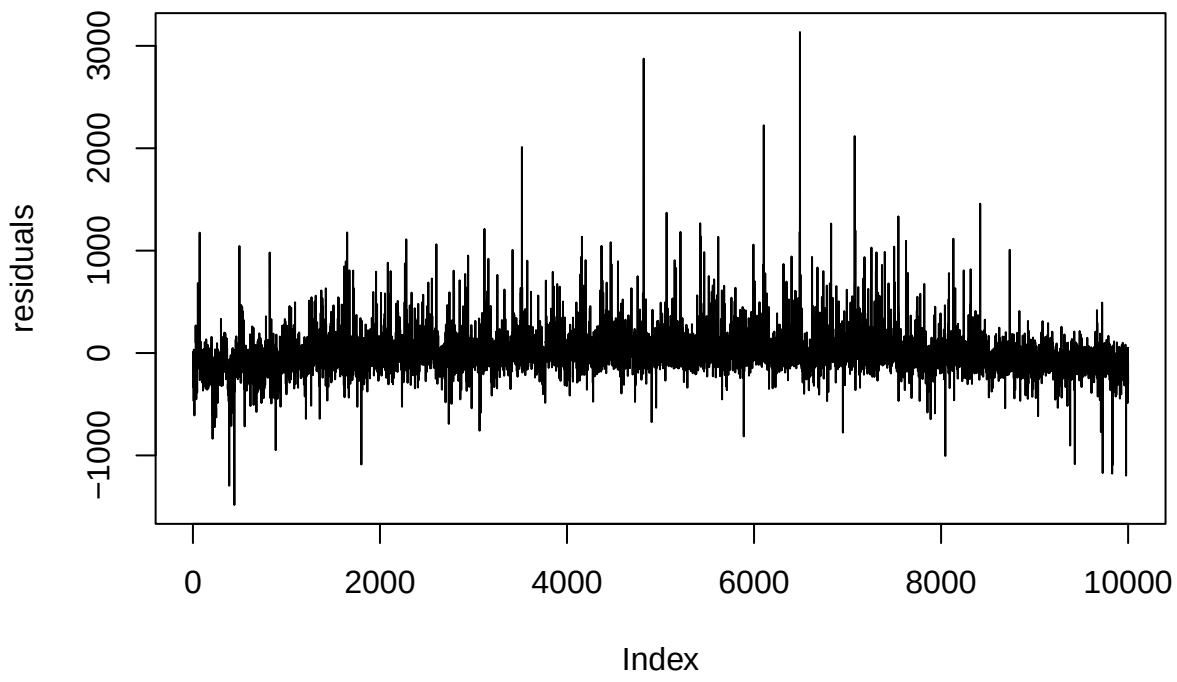

```
# Q-Q Plot
qqnorm(residuals)
qqline(residuals)
```



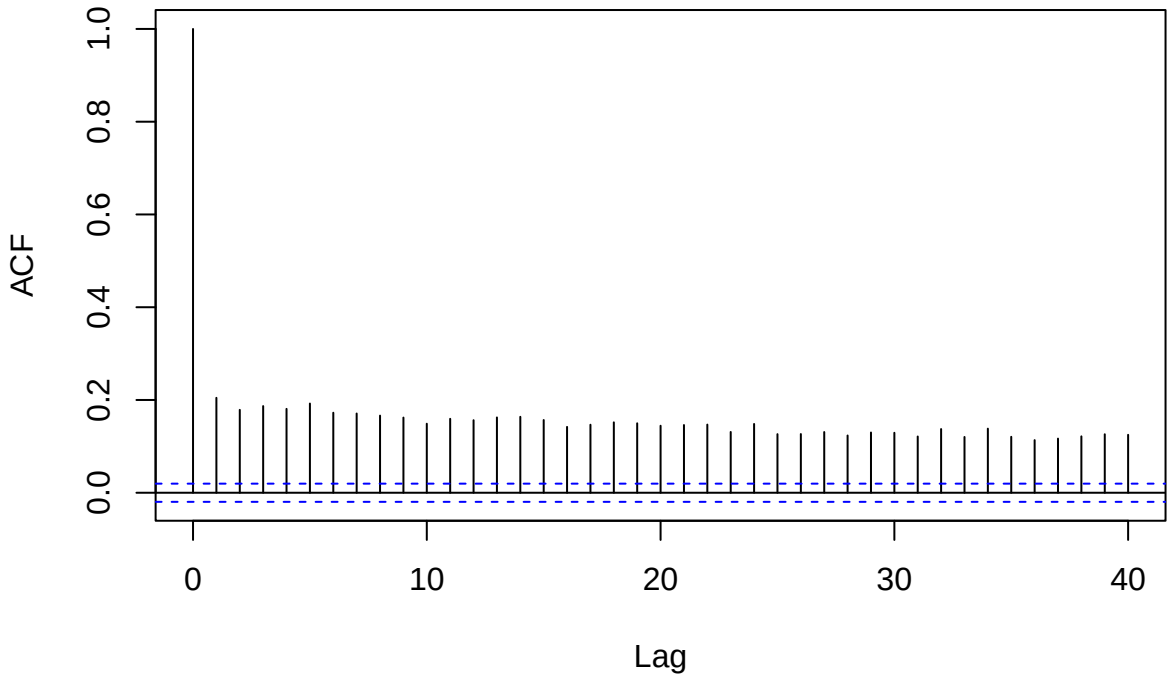

```
plot(fitted(time_pattern_dur_model_mixed), residuals, xlab="Fitted Values", ylab="Residuals")
abline(h=0, col="red")
```



```r
# For non-time series data
plot(residuals, type="l")
```




```r
# For time series data
acf(residuals)
```

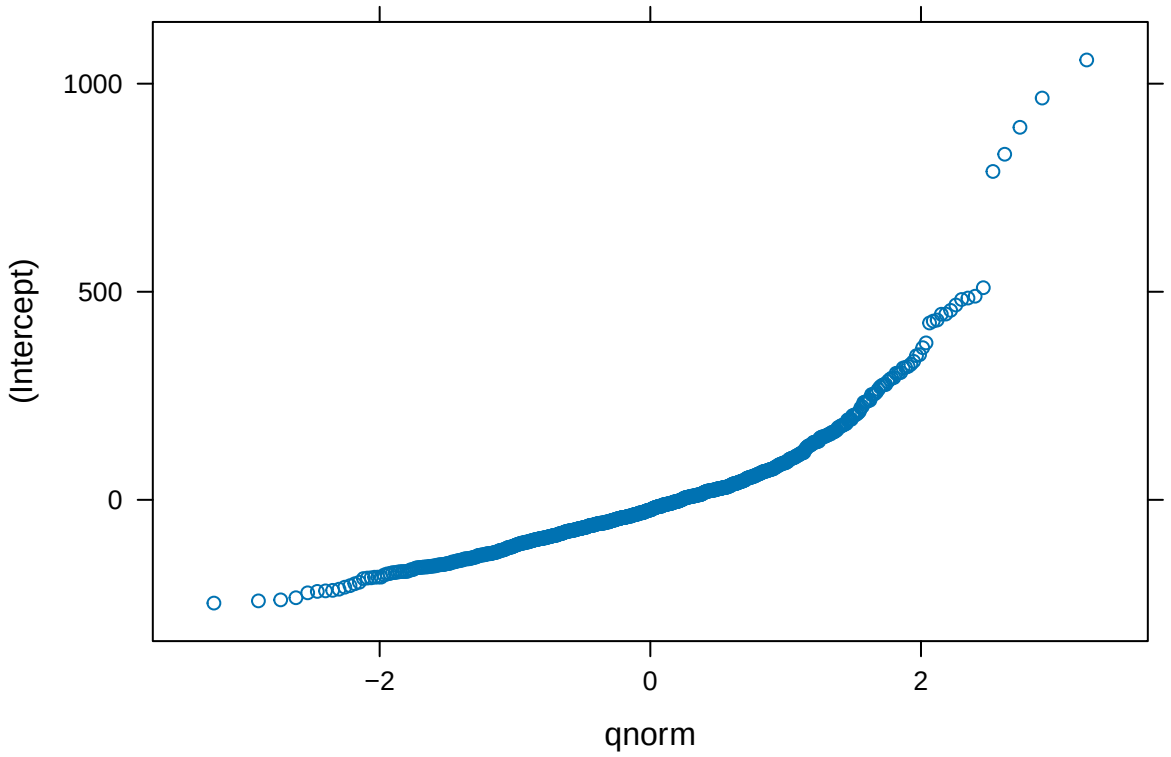



```
vif(time_pattern_dur_model_mixed)  # Note: This works if 'model' is an 'lm' object; for 'lmer' objects, yo
```

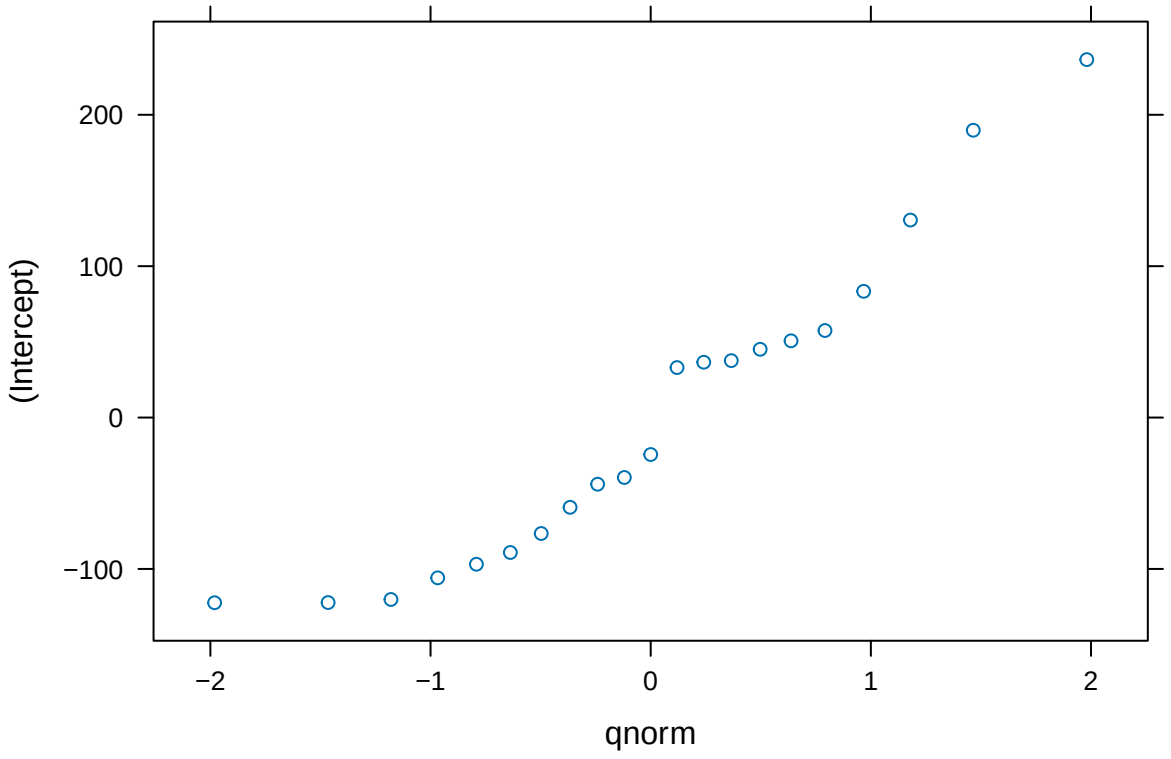
```
##      holiday     wknd_ind          mem wknd_ind:mem
##     1.009563     2.085232     1.403089     2.453155
```

```
ranef_plot <- ranef(time_pattern_dur_model_mixed)
plot(ranef_plot)
```

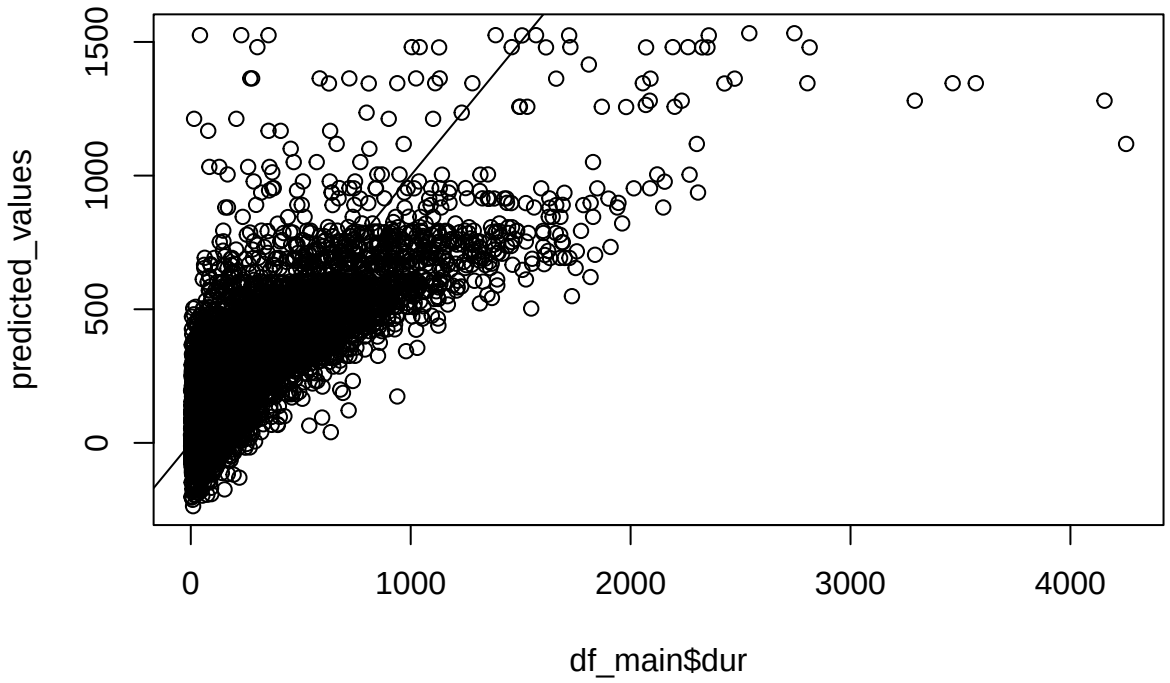
```
## $`station:district`
```


1000
500
0
(Intercept)
−2
0
2
qnorm

```
##
## $district
```


200
100
0
−100
(Intercept)
−2
−1
0
1
2
qnorm

```r
predicted_values <- predict(time_pattern_dur_model_mixed)
plot(df_main$dur, predicted_values)
abline(0, 1)
```



Histogram of Residuals The histogram shows the distribution of residuals. It suggests that the residuals are fairly symmetrically distributed around zero, indicating that the assumption of normality might be reasonably met. However, the distribution appears slightly leptokurtic (having a peak higher than a normal distribution), suggested by the tall center of the histogram.

Normal Q-Q Plot The Q-Q plot compares the quantiles of the residuals to the quantiles of a normal distribution. If the residuals were perfectly normally distributed, the points would lie on the 45-degree reference line. In the Q-Q plot, the points deviate from the line at the ends, indicating potential heavy tails in the distribution of residuals. This could suggest some departure from normality, particularly with potential outliers or extreme values.

Residuals vs Fitted Values Plot The residuals should be randomly scattered around the horizontal line at zero, with no clear pattern. In the plot, there seems to be a slight "funnel" shape, where the variance of the residuals increases with the fitted values, which could indicate heteroscedasticity.

Residuals vs Index Plot This plot displays residuals against the observation index. It's useful for detecting patterns that may indicate violation of independence. The residuals appear randomly scattered, suggesting no obvious violation of independence. However, there are some visible outliers, which should be investigated further.

ACF Plot of Residuals The autocorrelation function (ACF) plot is used to check for autocorrelation in the residuals at different lags. The bars represent correlations at different lag values. If most of them are within the blue dashed lines (representing confidence intervals), it suggests little to no autocorrelation. The ACF plot shows that autocorrelation is not a concern as the correlations are within the bounds.

Q-Q Plot of Random Effects This plot should show whether the random effects are normally distributed. The random effects (intercepts for `district/station` in the model) should fall along the reference line if they're normally distributed. There's some deviation from normality, but it's not extreme.

Predicted vs Actual Values Plot This plot compares the predicted values from the model to the actual values. Ideally, the points should fall around the 45-degree line, indicating good model fit. The plot shows a reasonable alignment along the line, although it seems to diverge for higher values, suggesting the model might not predict as well in that range.

Interpretation Summary The model assumptions are not strictly violated, but there are indications of potential issues:

- The residuals are roughly normally distributed but show signs of leptokurtosis.
- There might be some heteroscedasticity, as indicated by the Residuals vs Fitted Values plot.
- There are outliers in the data that could be influential points worth investigating.
- The assumption of independence seems to be met based on the Residuals vs Index and ACF plots.
- The random effects may slightly deviate from normality, but not severely.

Limitation of the model Given these observations, the following improvement could be made :

- Transforming the response variable or using robust regression techniques to handle non-normality and heteroscedasticity.
- Investigating and potentially addressing outliers.

```
time_pattern_dur.ar <- gls(dur ~ dd + wday + holiday, correlation = corAR1(form = ~ 1 | station), data = d

# Display model summary
summary(time_pattern_dur.ar)
```

```
## Generalized least squares fit by REML
##   Model: dur ~ dd + wday + holiday
##   Data: df_main
##        AIC      BIC    logLik
##   141131.6 141210.9 -70554.79
##
## Correlation Structure: AR(1)
##  Formula: ~1 | station
##  Parameter estimate(s):
##       Phi
## 0.4023917
##
## Coefficients:
##                    Value Std.Error   t-value p-value
## (Intercept)   283.03014  9.083027 31.160331  0.0000
## dd             -0.18235  0.306744 -0.594462  0.5522
## wdayMonday    -45.08756  9.932037 -4.539609  0.0000
## wdaySaturday   11.35289  9.759227  1.163298  0.2447
## wdaySunday    -13.42787  9.799037 -1.370326  0.1706
## wdayThursday  -22.79589  9.790562 -2.328353  0.0199
## wdayTuesday   -36.13164  9.912338 -3.645118  0.0003
## wdayWednesday -29.46184  9.820837 -2.999931  0.0027
## holiday1       60.77485 18.146140  3.349189  0.0008
##
##  Correlation:
##               (Intr) dd     wdyMnd wdyStr wdySnd wdyThr wdyTsd wdyWdn
## dd            -0.509
## wdayMonday    -0.523 -0.027
## wdaySaturday  -0.543 -0.010  0.501
## wdaySunday    -0.537 -0.010  0.496  0.508
## wdayThursday  -0.525 -0.024  0.514  0.499  0.497
## wdayTuesday   -0.527 -0.018  0.494  0.502  0.494  0.494
## wdayWednesday -0.528 -0.022  0.494  0.505  0.502  0.499  0.498
## holiday1      -0.028  0.050 -0.180  0.004  0.005 -0.101  0.005  0.001
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.0290197 -0.6311301 -0.3020733  0.3448169 12.9772479
##
## Residual standard error: 305.1222
## Degrees of freedom: 10000 total; 9991 residual
```

```
# holiday not working as random coefficient
# time_pattern_dur.rand_coef <- lme(dur ~ dd + wday + holiday, random = ~1 + dd |
# station, data = df_main)
# summary(time_pattern_dur.rand_coef)

## Does not converges...
```

Business interpretation

From a business perspective, the findings from this analysis offer valuable insights for strategic planning, marketing, operational adjustments, and potential policy development. Here are the main takeaways:

Holidays and Weekends promotion The model indicates longer trip durations during holidays and weekends. This suggests higher usage or leisurely rides during these periods. There could be an opportunity to increase bike availability or introduce special promotions during holidays and weekends to cater to this demand. The interaction effect suggests that members' increased trip duration is less pronounced on weekends. This could imply that members use the service differently on weekends compared to weekdays. Design weekend-specific promotions or services for members. Understanding why this pattern occurs (leisure vs. commuting) can help tailor these offerings.

Membership pricing strategy Members tend to have significantly longer trip durations compared to non-members. This highlights the importance of members to the system. There should have a focus on member retention strategies and consider special offers or loyalty programs to encourage repeat usage. Additionally, analyzing non-member behavior to tailor services and promotions effectively would be pertinent.

Geographic optimization Significant variability in trip durations across different stations and districts indicates diverse usage patterns in different areas. Optimize bike and dock availability based on specific district and station demands. Targeted investments in high-usage areas could improve service efficiency.

Potential Policy Implications Understanding how different areas and demographics use the bike-sharing system can inform urban planning and public transport policies. Promoting bike-sharing effectively can contribute to environmental goals by reducing reliance on motorized transportation. # Research Question 3: What variables impact the average bixi trip duration?

The objective is to identify the driving factors of a bixi's trip length when we control for most of the variables. Trip length is one of the three important variables that drives revenue, the other ones being the number of trips and the pricing scheme. Keep in mind that increasing the trip length does not necessarily increase revenues since an unwanted increase in trip length may discourage users from using bixi's system and result in a decrease in trip number.

Variables Selection

REMOVE PART OF MONTH Our goal is to incorporate most of the important variables in order to increase our chance of respecting the assumption of E(e)=0 and thus making our model more telling.

Variables that make business sense to include:

From our seasonality analysis we identified:

- Season; grouping of months from april to november in their respective season (`season`)
- Temperature in degrees celcius (`temp`)
- Rainfall in mm (`rain`)

From our daily and weekly pattern analysis we identified:

- Part of the week i.e. weekend or weekday (`wknd_ind`)
- If it is a holiday (`holiday`)

Some other variables that are interesting:

- If the user is a member(`mem`)
- Location of the bixi station compared to Parc Lafontaine, a landmark in the middle of the bixi station system (`cardinality`)
- Proportion of trips in the morning versus the whole day (`percent_AM`)
- If the station name contains the word 'metro' (`Metro_ind`)

Interactions: In our EDA we observed a different week day usage of the member and non members, thus an interaction term between members and day of week would be interesting. (`wday*mem`).

Correlation:

Let's take a quick look at the correlation between our numerical variables to estimate the effect of collinearity.

```
##                    avg        temp        rain        n_tot   percent_AM
## avg         1.00000000  0.09639054 -0.10619900 -0.215866274 -0.107387372
## temp        0.09639054  1.00000000 -0.02794911  0.139997362 -0.078110564
## rain       -0.10619900 -0.02794911  1.00000000 -0.054717667  0.013211523
## n_tot      -0.21586627  0.13999736 -0.05471767  1.000000000 -0.008953075
## percent_AM -0.10738737 -0.07811056  0.01321152 -0.008953075  1.000000000
```

We see very low correlation between the Xs which means we should not get any problems with collinearity between our numerical variables.

Model

head(df_main)

df_main %>% count(station, sort = TRUE)

Benchmark model

```
##
## Call:
## lm(formula = avg ~ season + temp + rain + wknd_ind * mem + holiday +
##     cardinality + percent_AM + Metro_ind, data = df_main)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.568  -3.572  -1.158   2.049  43.355
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              14.09278    0.28754  49.012  < 2e-16 ***
## seasonSpring              2.66468    0.17450  15.270  < 2e-16 ***
## seasonSummer              0.50949    0.18451   2.761 0.005767 **
## temp                      0.11321    0.01348   8.400  < 2e-16 ***
## rain                     -0.10451    0.01213  -8.619  < 2e-16 ***
## wknd_indWeekend           2.47741    0.19998  12.389  < 2e-16 ***
## mem1                     -1.83370    0.15033 -12.198  < 2e-16 ***
## holiday1                  1.08686    0.42192   2.576 0.010011 *
## cardinalityNorth-West    -0.48500    0.20482  -2.368 0.017908 *
## cardinalitySouth-East    -0.17149    0.22365  -0.767 0.443222
## cardinalitySouth-West    -0.27663    0.19931  -1.388 0.165176
## percent_AM               -2.04194    0.31226  -6.539 6.48e-11 ***
## Metro_ind1               -0.76012    0.23055  -3.297 0.000981 ***
## wknd_indWeekend:mem1     -1.54764    0.27641  -5.599 2.21e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 6.263 on 9986 degrees of freedom
## Multiple R-squared:  0.0984, Adjusted R-squared:  0.09723
## F-statistic: 83.84 on 13 and 9986 DF,  p-value: < 2.2e-16
```

Model with unstructured covariance matrix

Interpretation

Overall Model - The model explains approximately 10% of the variation in the average trip duration which means that other factors are also at play and are not included in the model. The p value associated with the F-statistic is very low, hence our model is globally significant.

Intercept : The interpretation of the intercept does not make sense in this case

Season: The reference level is fall. We can see that on average trip duration during spring and summer are respectively 2.7 and 0.5 minutes longer than in fall holding everything else constant.

Temperature: The coefficient of temperature is 0.11 which means that an increase in temperature of 1 degree celcius corresponds to an increase of average trip duration of 0.11 minutes on average holding all else constant.

Rainfall: The coefficient for rain is -0.1 which means that an increase in rainfall of 1 mm corresponds to a decrease of average trip duration of 0.1 minutes on average holding all else constant.

Effect of Weekend Indicator and membership: Since there exists an interaction between both variables, it is no longer possible to interpret one without the other. This implies that the relation between average trip duration and membership is different depending on the moment of the week. The opposite is also true, the relation between average trip duration and the moment of the week is different depending on the membership status. We observe that non-member have longer trips on average and that weekend trips tends to increase average trip length.

The 4 different levels in order of trip length are as follows: EXPLAIN IN WORDS 1. Lowest level : Weekday and member (-1.84 minutes)

2. Second lowest : Weekend and member (2.5-1.8-1.5 = -0.8 minutes)
3. Reference level: Weekday and non-member (0 minutes)
4. Highest level : weekend and non-member (+2.47 minutes)

Holiday: The coefficient for holiday is 1.06 which means that during holidays average trip duration is 1.06 minutes higher on average than during non-holidays, holding all else constant.

North_South and West_East: Their coefficients are 0.08 and -0.26 which means that on average the average trip duration for trips starting at a station South of Parc Lafontaine or West is 0.08 and -0.26 minutes different from their counter parts respectively, holding all else constant. Keep in mind that the coefficient for North_South is not significantly different from zero

Percent AM: The magnitude of the coefficient -2.03 is less important than its sign for our interpretation. What it means is that as the proportion of trips in the morning increases, the average trip duration generally decreases when holding all else constant. This hints that trips in the morning might be shorter on average than trip in the afternoon, hence bring in less revenue.

Part of Month : The coefficient for part of month is -0.22 which means that on average, the average trip duration is 0.22 minutes shorter in the second half of the month holding all else constant. This feature was not found to be significantly different from zero.

Metro Indicator : Metro indicator's coefficient is -0.75 which means that the expected value for average trip length decreases by 0.75 minutes when a bixi station is near a metro acces point, holding all else constant. This would suggest that user who rent bikes after making a metro ride are closer to their final destination than in other cases.

Business Implications:

1. **Promotion and Marketing**: For the same temperature, average trip length tends to be the longest in spring. This indicates that users are eager to use bikes after winter. This insight could be used for promotion purposes.

2. **Resource Allocation**: Expect longer trips when it is hot and non-rainy outside. Even more if it is a weekend or holiday. Also, bikes tend to be borrowed longer during the afternoon than in the morning. Stations south of Parc Lafontaine have on average longer trip duration, which may suggests that stations are further from one another. There might be some space for additional stations.

3. **Pricing Strategy**: The usage that is associated with the longest trip length based on our interaction term is for non-members during the weekend. Charging a heftier price for these people at that time may increase profit margins significantly.

Verification of assumptions and collinearity

Variance Inflation Factor

Let's use the variance inflation factor to verify for collinearity, we will use a standard threshold of 5.

No major problem is detected, since the global vifs are all relatively low.

Verification of Normality of Residuals

No problem here, residuals are normally distributed.

Model correctly specified

The model seems to be correctly specified.

Verificaiton of Heteroscedasticity

No major problem of heteroscedasticity were detected. The variable `n_tot` has been removed as stated earlier.

Limitations and shortcomings

- Causation vs. Correlation: The regression model captures relationships but does not establish causation.
- Data Exclusions: The data only considers trips under 60 minutes, which might exclude a segment of users who use BIXI for longer journeys.
- Other External Factors: Events, road conditions, or public transportation disruptions can affect BIXI usage but are not captured in the dataset.

Conclusion

In conclusion, several key operational and strategic considerations have emerged from the data analysis of BIXI bike rentals:

Operational Adjustments: The data suggests that revenue is higher in warmer months. To capitalize on this, it is advisable to optimize operations during this period, which could involve increasing staffing, enhancing promotional activities, and ensuring optimal equipment availability.

Rainy Day Strategies: Rainfall appears to have a negative impact on revenue. Implementing strategies to mitigate this effect, such as promotional offers or special activities for rainy days, may help attract more customers.

Promotion and Marketing: Data indicates that average trip length is longest in spring, suggesting an eagerness to use bikes after winter. This insight can be leveraged for promotional purposes.

Resource Allocation: Understanding patterns in trip duration based on weather, time of day, and location is crucial for resource allocation. Longer trips are expected during hot, non-rainy weekends and holidays. Stations in certain areas have longer trip durations, indicating potential for additional station placement.

Pricing Strategy: The analysis highlights that non-members on weekends tend to take longer trips. Adjusting pricing for this group during these times could significantly increase profit margins.

Operational Strategy: It's important to consider the tradeoff between the number of trips and average trip length. Increasing the number of trips on a given day may lead to shorter hauls. This information should inform operational decisions. Incorporating these insights into the business's operations and strategies can lead to improved efficiency, customer satisfaction, and profitability

Contribution

Charles Julien :Research question 3, version control, part of feature engineering, formating.

Gabriel Jobert : Research question 1 and 2

Chike Odenigbo: exploratory models (not included), feature engineering, influential observations, autocorrelation

Atul Sharma: Contributed in developing the Reseacrh questions, interpreting the findings of the model and finalising the conclusion .