Step 1: Loading in the Data

```r
# Packages used in tutorials
library(MASS)      # boxcox
library(car)       # qqPlot
```

```
## Loading required package: carData
```

```r
library(randtests) # runs.test
```

```
## Warning: package 'randtests' was built under R version 4.3.3
```

```r
# library(forecast) # OPTIONAL if you want auto.arima, not required

bike <- read.csv("trips_per_day.csv")
bike$trip_date <- as.Date(bike$trip_date)

str(bike)
```

```
## 'data.frame':    2969 obs. of  2 variables:
##  $ trip_date: Date, format: "2016-01-10" "2016-01-11" ...
##  $ n_trips  : int  2273 3623 2535 2966 2970 2636 4122 3104 1642 4834 ...
```

```r
head(bike)
```

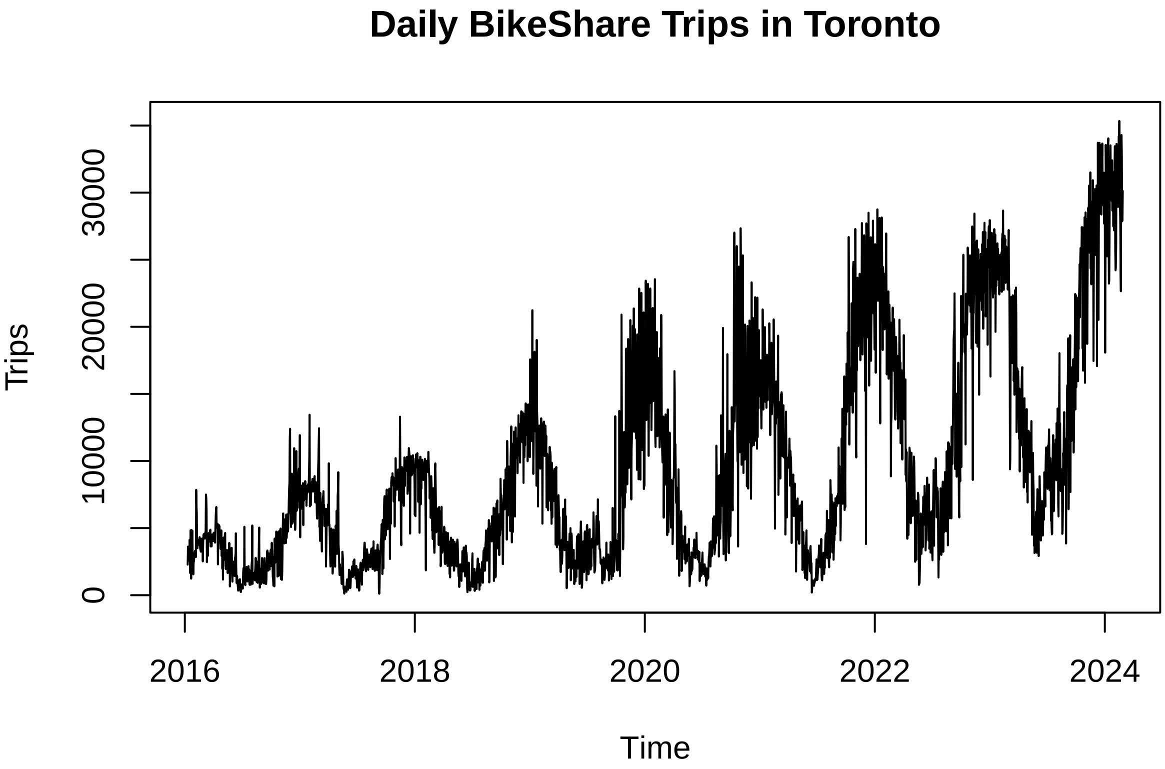
```
##    trip_date n_trips
## 1 2016-01-10    2273
## 2 2016-01-11    3623
## 3 2016-01-12    2535
## 4 2016-02-10    2966
## 5 2016-02-11    2970
## 6 2016-02-12    2636
```

```r
range(bike$trip_date)
```

```
## [1] "2016-01-10" "2024-09-30"
```

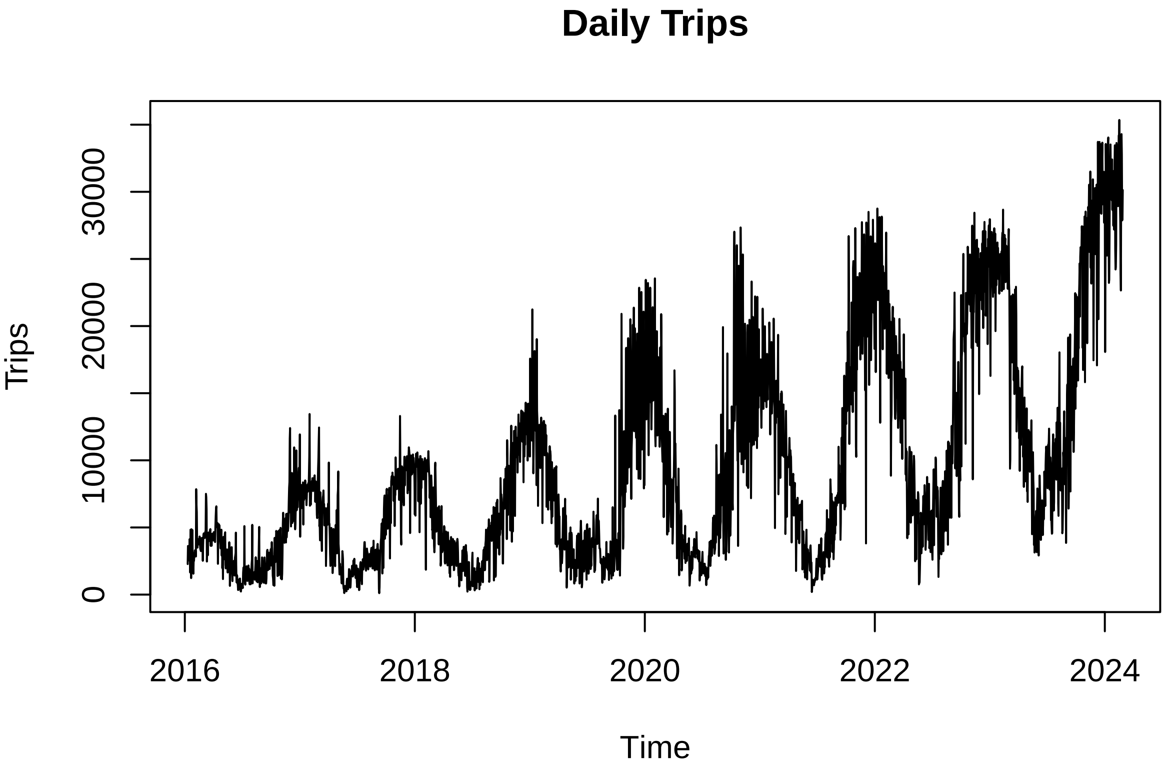
Initial Plotting for Time Series

```r
# Sort just in case
bike <- bike[order(bike$trip_date), ]

# Extract response as a vector
y <- bike$n_trips

# Daily frequency with yearly seasonality (approx 365)
bike_ts <- ts(
  y,
  start = c(as.numeric(format(min(bike$trip_date), "%Y")),
            as.numeric(format(min(bike$trip_date), "%j"))),
  frequency = 365
)

plot(bike_ts, main = "Daily BikeShare Trips in Toronto", ylab = "Trips")
```

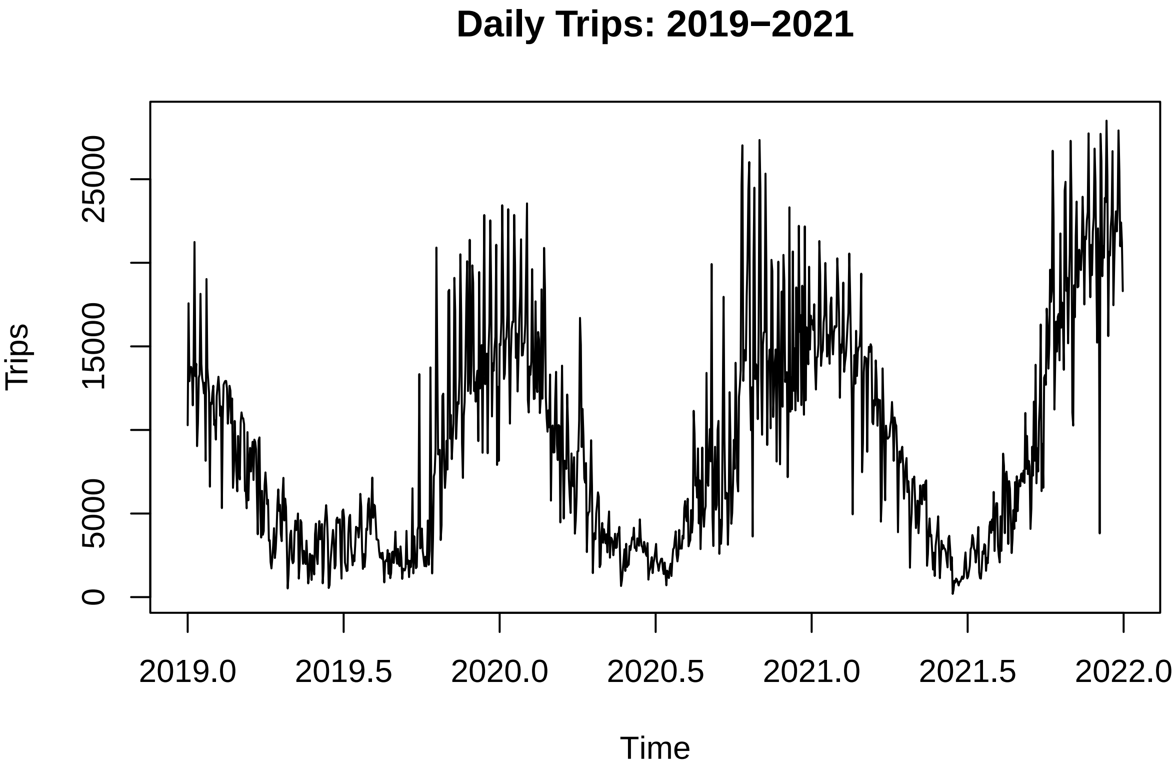


Step 2: EDA

```r
par(mfrow = c(1, 1))
plot(bike_ts, main = "Daily Trips", ylab = "Trips")
```

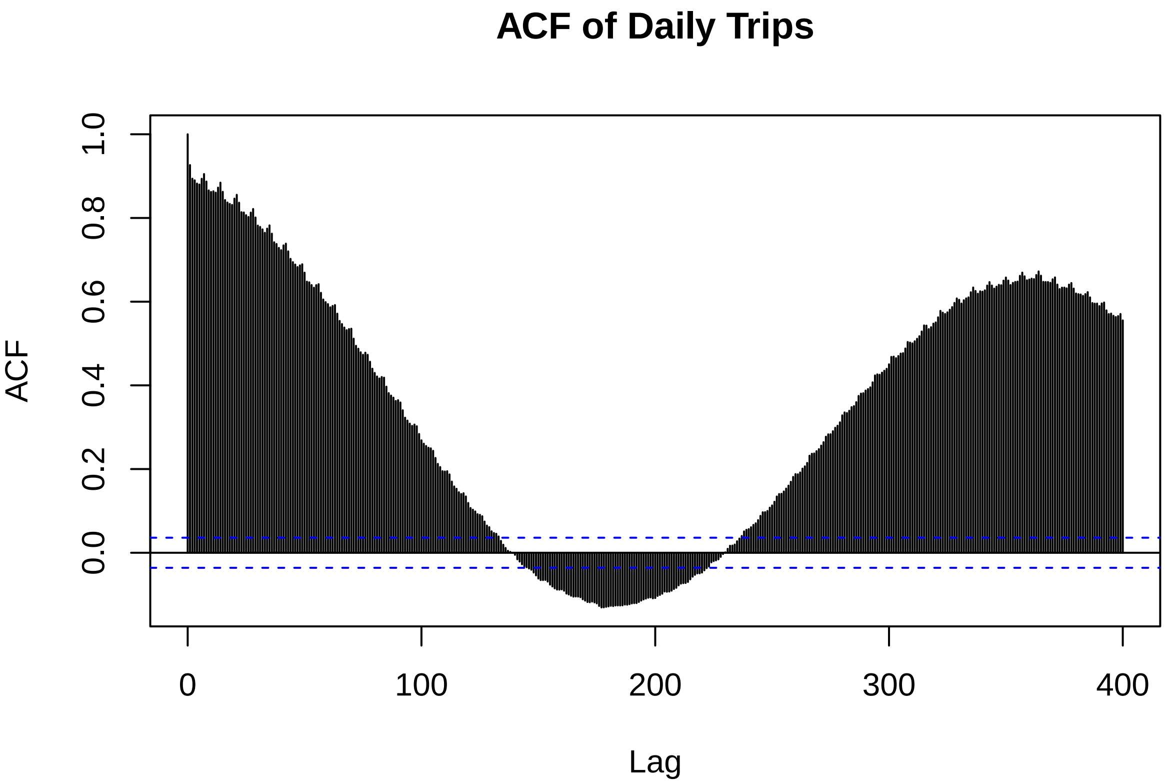


```r
# maybe a zoom on a couple of years
plot(window(bike_ts, start = c(2019, 1), end = c(2021, 365)),
     main = "Daily Trips: 2019-2021", ylab = "Trips")
```

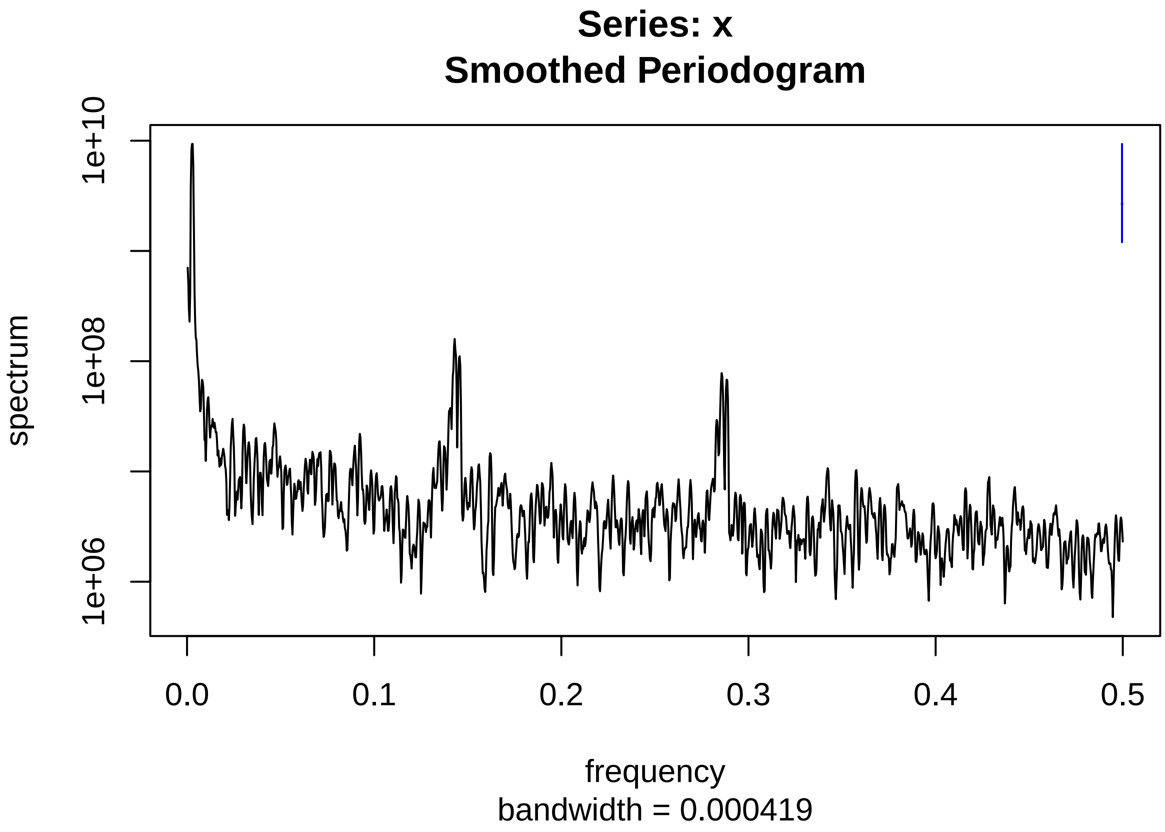


Confirming Seasonality with ACF and Spectrum

```r
acf(as.vector(bike_ts), lag.max = 400,
    main = "ACF of Daily Trips")
```



```r
spec_bike <- spectrum(as.vector(bike_ts), spans = 5)
```

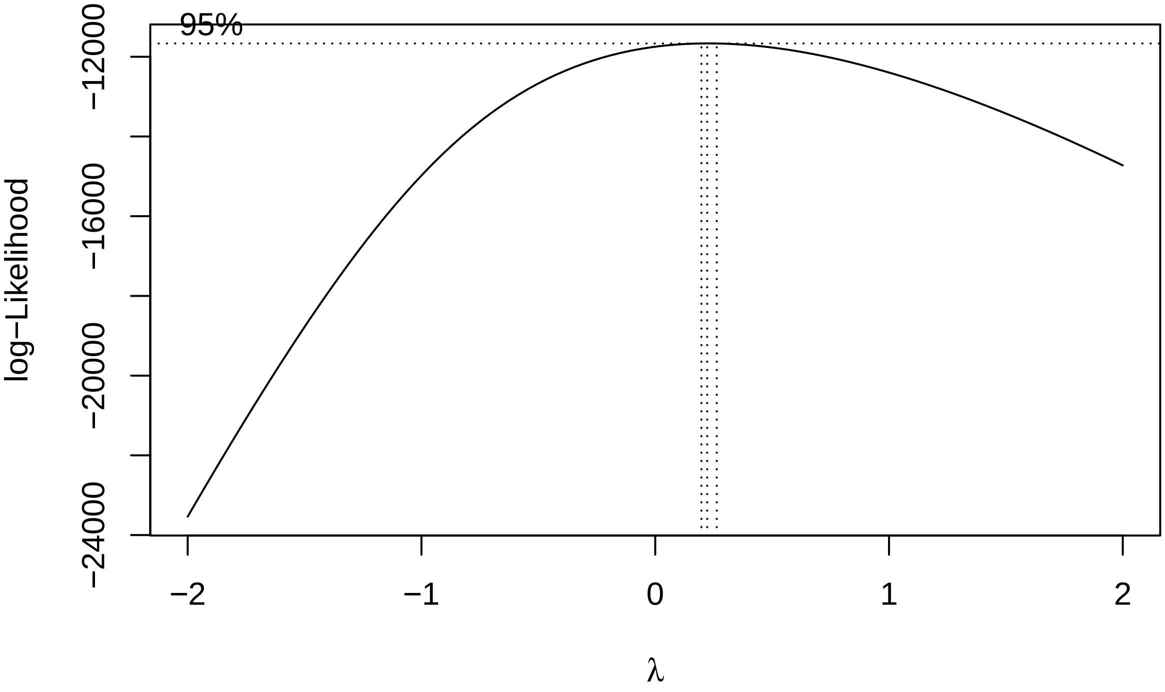


```r
1 / spec_bike$freq[which.max(spec_bike$spec)]  # estimated period
```

```
## [1] 333.3333
```

Step 3: Box-Cox transformation

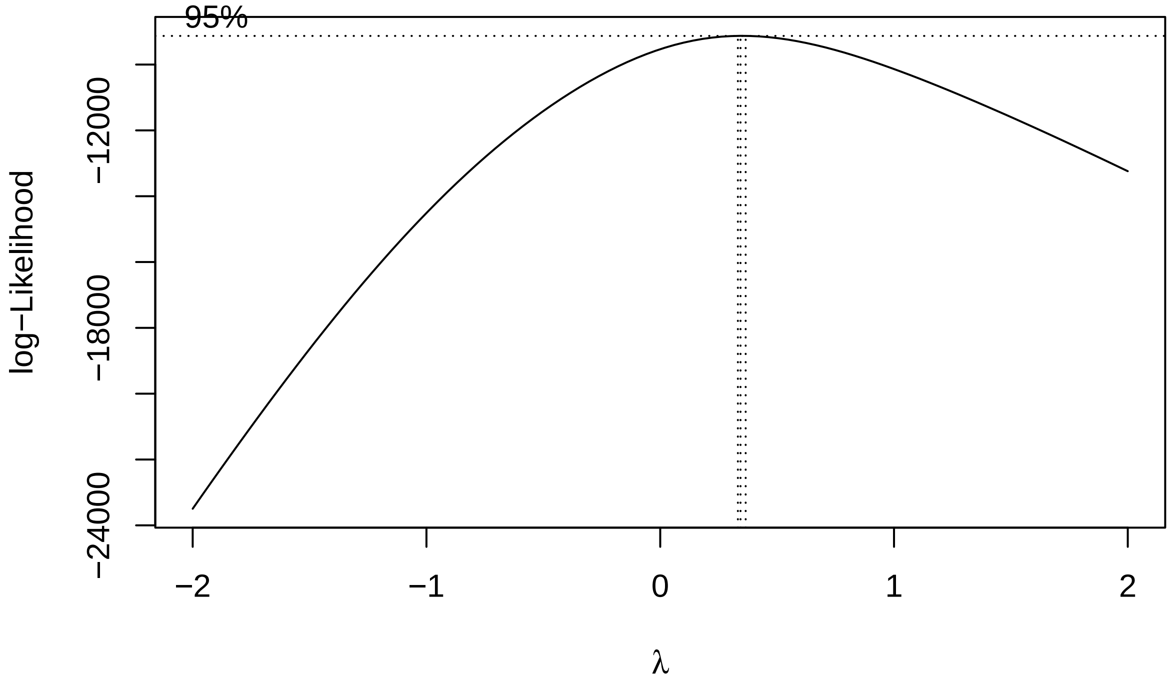
```r
# Simple intercept-only model (like in tutorial)
bc_model_raw <- lm(bike_ts ~ 1)
boxcox_raw   <- MASS::boxcox(bc_model_raw, lambda = seq(-2, 2, 0.1))
```



```r
(lambda_opt_raw <- boxcox_raw$x[which.max(boxcox_raw$y)])
```

```
## [1] 0.2222222
```

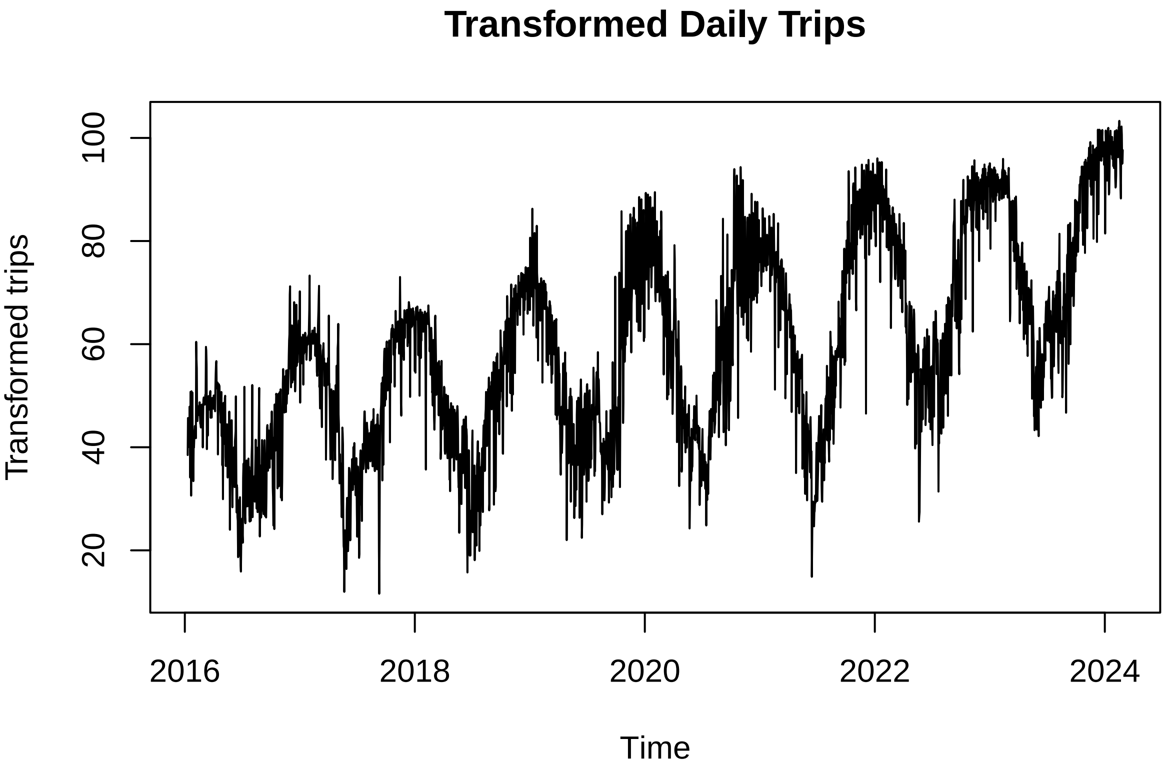
```r
tim <- time(bike_ts)  # continuous time index

# Season: year and day-of-year or month; simplest is month
# Build a monthly factor from dates (instead of cycle, since this is daily)
month <- factor(format(bike$trip_date, "%m"))

reg_for_bc <- lm(bike_ts ~ tim + month)
boxcox_mod <- MASS::boxcox(reg_for_bc, lambda = seq(-2, 2, 0.1))
```



```
(lambda_opt_mod <- boxcox_mod$x[which.max(boxcox_mod$y)])
```

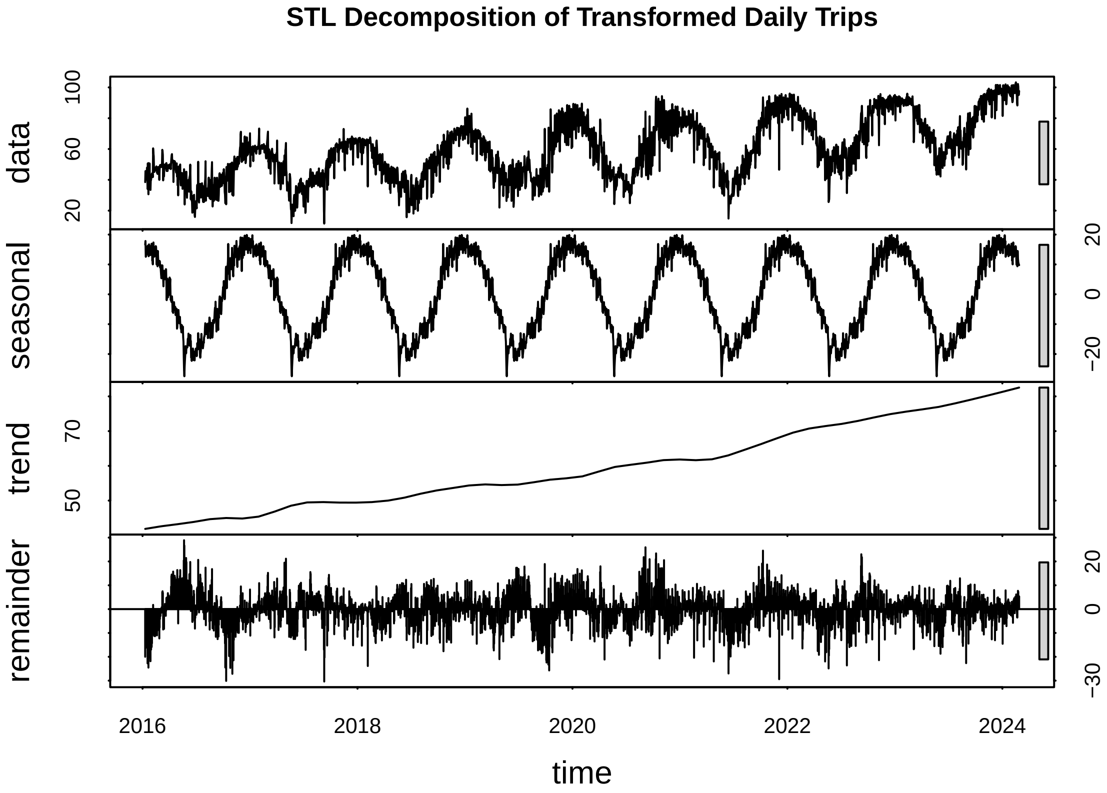
```
## [1] 0.3434343
```

```
lam <- lambda_opt_mod   # keep this for later

if (lam == 0) {
  y_trans <- log(bike_ts)
} else if (lam > 0) {
  y_trans <- (bike_ts^lam - 1) / lam
} else {
  # negative lambda → use minus sign trick like in lectures
  y_trans <- -(bike_ts^lam)
}

plot(y_trans, main = "Transformed Daily Trips", ylab = "Transformed trips")
```



Classical Decomposition

```
Decomp_bike <- stl(y_trans, s.window = "periodic")
plot(Decomp_bike, main = "STL Decomposition of Transformed Daily Trips")
```



```r
# Extract components if needed
bike_seasonal <- Decomp_bike$time.series[, "seasonal"]
bike_trend    <- Decomp_bike$time.series[, "trend"]
bike_remainder <- Decomp_bike$time.series[, "remainder"]
```

```r
tim   <- time(y_trans)
month <- factor(format(bike$trip_date, "%m"))
dow   <- factor(weekdays(bike$trip_date))  # optional, but nice

# Simple model: linear trend + monthly seasonality
reg1 <- lm(y_trans ~ tim + month)
summary(reg1)
```

```
##
## Call:
## lm(formula = y_trans ~ tim + month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.930  -4.421   0.841   5.135  33.503
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.010e+04  1.288e+02 -78.419  < 2e-16 ***
## tim          5.022e+00  6.378e-02  78.738  < 2e-16 ***
## month02      6.976e-01  7.537e-01   0.926 0.354735
## month03      6.327e+00  7.358e-01   8.599  < 2e-16 ***
## month04      1.183e+01  7.421e-01  15.935  < 2e-16 ***
## month05      2.364e+01  7.360e-01  32.113  < 2e-16 ***
## month06      3.001e+01  7.424e-01  40.431  < 2e-16 ***
## month07      3.270e+01  7.130e-01  45.862  < 2e-16 ***
## month08      3.376e+01  7.130e-01  47.344  < 2e-16 ***
## month09      3.207e+01  7.186e-01  44.635  < 2e-16 ***
## month10      2.345e+01  7.404e-01  31.666  < 2e-16 ***
## month11      1.429e+01  7.468e-01  19.132  < 2e-16 ***
## month12      2.715e+00  7.403e-01   3.667 0.000249 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.143 on 2956 degrees of freedom
## Multiple R-squared:  0.8212, Adjusted R-squared:  0.8205
## F-statistic:  1131 on 12 and 2956 DF,  p-value: < 2.2e-16
```

```r
reg2 <- lm(y_trans ~ poly(tim, 2, raw = FALSE) + month)
summary(reg2)
```

```
##
## Call:
## lm(formula = y_trans ~ poly(tim, 2, raw = FALSE) + month)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.363  -4.468   0.913   5.123  33.023
##
## Coefficients:
```

```
##                             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  41.6116     0.5175  80.412  < 2e-16 ***
## poly(tim, 2, raw = FALSE)1 642.5618     8.1150  79.182  < 2e-16 ***
## poly(tim, 2, raw = FALSE)2  47.9377     8.1680   5.869 4.87e-09 ***
## month02                      0.6949     0.7495   0.927 0.353935    
## month03                      6.3211     0.7317   8.639  < 2e-16 ***
## month04                     11.8129     0.7379  16.008  < 2e-16 ***
## month05                     23.6142     0.7319  32.264  < 2e-16 ***
## month06                     29.9821     0.7382  40.614  < 2e-16 ***
## month07                     32.4508     0.7102  45.690  < 2e-16 ***
## month08                     33.5098     0.7102  47.183  < 2e-16 ***
## month09                     31.8278     0.7158  44.468  < 2e-16 ***
## month10                     23.4850     0.7363  31.897  < 2e-16 ***
## month11                     14.3325     0.7427  19.299  < 2e-16 ***
## month12                      2.7584     0.7362   3.747 0.000183 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.097 on 2955 degrees of freedom
## Multiple R-squared:  0.8233, Adjusted R-squared:  0.8225 
## F-statistic:  1059 on 13 and 2955 DF,  p-value: < 2.2e-16
```

```r
reg3 <- lm(y_trans ~ poly(tim, 2, raw = FALSE) + month + dow)
summary(reg3)
```

```
## 
## Call:
## lm(formula = y_trans ~ poly(tim, 2, raw = FALSE) + month + dow)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -37.086  -4.311   0.836   4.901  31.512 
## 
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  42.3325     0.6270  67.516  < 2e-16 ***
## poly(tim, 2, raw = FALSE)1 642.5699     8.0310  80.011  < 2e-16 ***
## poly(tim, 2, raw = FALSE)2  48.1134     8.0835   5.952 2.96e-09 ***
## month02                      0.6592     0.7418   0.889 0.374229    
## month03                      6.2757     0.7242   8.666  < 2e-16 ***
## month04                     11.8192     0.7303  16.184  < 2e-16 ***
## month05                     23.5691     0.7244  32.538  < 2e-16 ***
## month06                     29.9529     0.7307  40.995  < 2e-16 ***
## month07                     32.4546     0.7029  46.171  < 2e-16 ***
## month08                     33.4604     0.7029  47.603  < 2e-16 ***
## month09                     31.8053     0.7084  44.897  < 2e-16 ***
## month10                     23.4927     0.7287  32.240  < 2e-16 ***
## month11                     14.2872     0.7350  19.438  < 2e-16 ***
## month12                      2.7472     0.7287   3.770 0.000166 ***
## dowMonday                   -1.9523     0.5495  -3.553 0.000387 ***
## dowSaturday                 -1.0741     0.5501  -1.953 0.050951 .  
## dowSunday                   -2.8167     0.5501  -5.120 3.25e-07 ***
## dowThursday                  0.3757     0.5501   0.683 0.494652    
## dowTuesday                  -0.2132     0.5501  -0.388 0.698380    
## dowWednesday                 0.8004     0.5508   1.453 0.146250    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.014 on 2949 degrees of freedom
## Multiple R-squared:  0.8273, Adjusted R-squared:  0.8262
## F-statistic: 743.3 on 19 and 2949 DF,  p-value: < 2.2e-16
```