

Question 1A

The Question:

How did the October 1, 2020 reinstatement of fares affect ridership on King County Metro?

Key Details:

- reinstatement on October 1 may have been staggered or irregularly implemented (source: correspondence)
- ridership is “stop-level passenger boardings... and alightings.”
- UA for this view – per-day per-route per-direction (e.g. 100 northbound riders on bus 1 on September 25th 2020) as the unit of analysis

Questions for discussion:

1. Measures.

- Should the UA be different? (“stop level”?)
- I am assuming that SD units for daily ridership counts makes sense – what do folks think? There are either rarely-ridden routes or ones with very low data. What shall we do (if anything)?

2. Time/Seasonality.

- How shall we handle weekend vs weekday?
- Overall picture has very weak evidence of change. Should we hunt for routes that saw change?
- Identifying the treatment is a bit arbitrary. Is it worth doing changepoint analysis for the ITS?

```
#Set some early assumptions, do some data cleaning
threshold = 3 # how many days, plus or minus, should we treat as an implementation period?

#some have both N S E W and NULL dirs
routeDF <- subset(routeDF, routeDF$TRIP_COMPASS_DIR_CD != 'NULL') #so let's drop nulls

#add in a boolean for before/after treatment
routeDF <- routeDF %>% mutate(treated = case_when(OPERATION_DATE <= policyChange ~ TRUE,
                                                OPERATION_DATE > policyChange ~ FALSE))

# strong seasonality in the data
routeDF.weekdays <- subset(routeDF, (wday(routeDF$OPERATION_DATE) != 7 &
                                     wday(routeDF$OPERATION_DATE) != 1)) #remove sat/sun
routeDF.sunday <- subset(routeDF, wday(routeDF$OPERATION_DATE) == 1)
routeDF.saturday <- subset(routeDF, wday(routeDF$OPERATION_DATE) == 7)

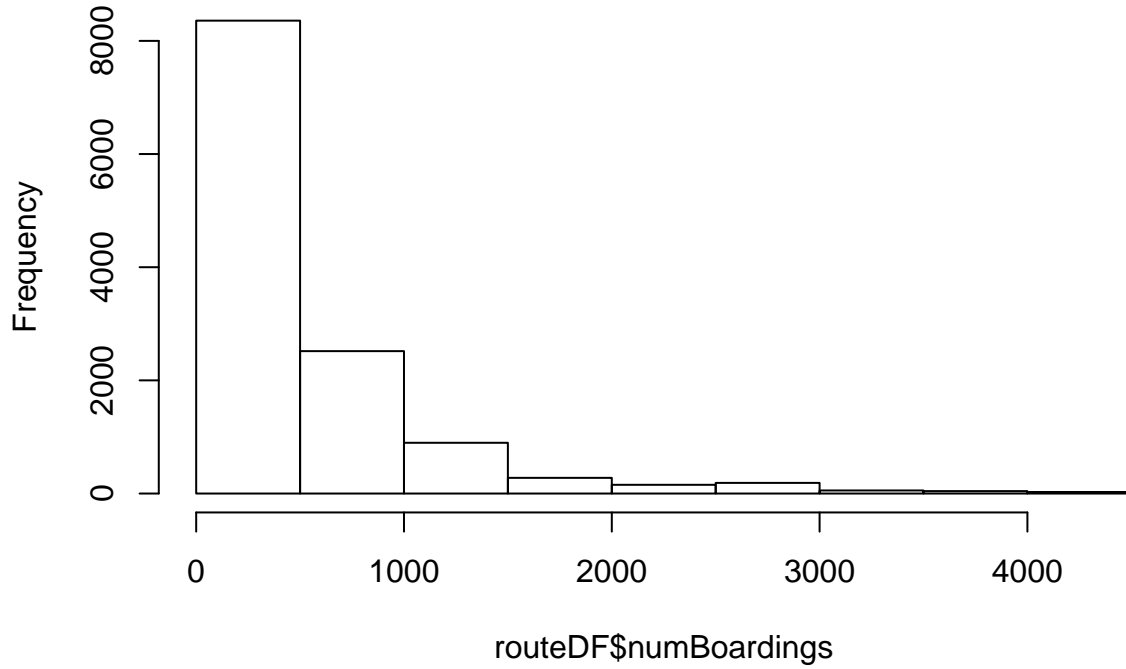
routeDF.weekdays <- routeDF.weekdays %>%
  group_by(SERVICE_RTE_LIST, TRIP_COMPASS_DIR_CD) %>%
  mutate(numBoardings.sd = scale(numBoardings))

# for ITS, add boolean for before/after treatment. drops dates within threshold
routeDF.filt <- subset(routeDF, (routeDF$OPERATION_DATE > policyChange + threshold) |
                      (routeDF$OPERATION_DATE < policyChange - threshold))
routeDF.weekdays.filt <- subset(routeDF.weekdays,
                                (routeDF.weekdays$OPERATION_DATE > policyChange + threshold) |
                                (routeDF.weekdays$OPERATION_DATE < policyChange - threshold))
```

Distributions of Number of Boardings

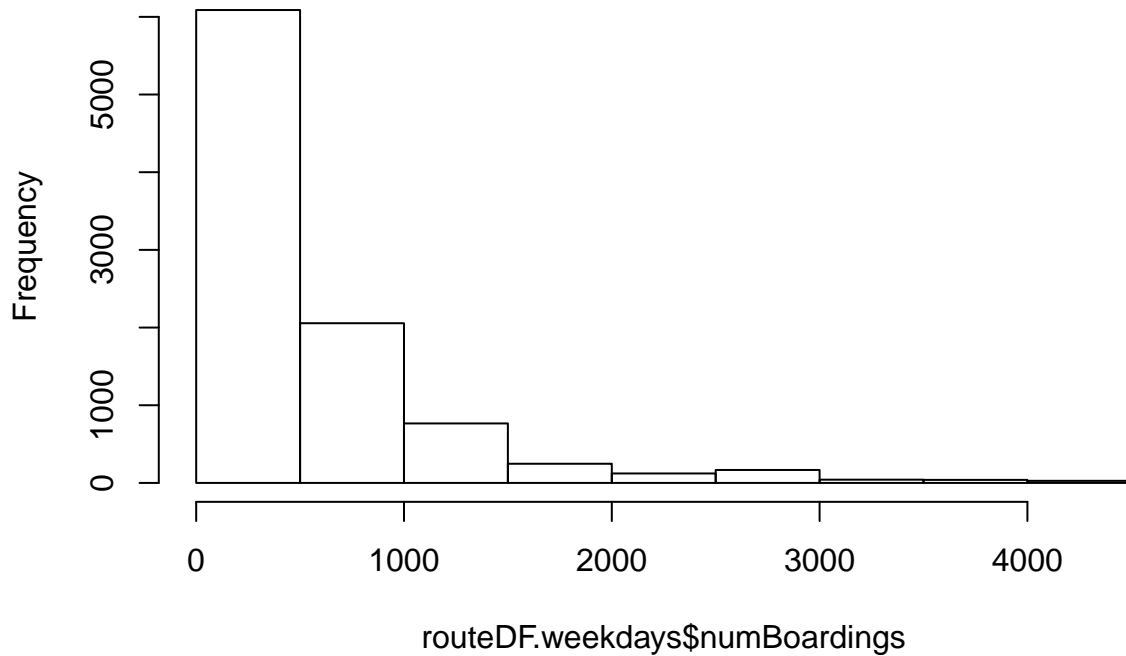
```
hist(routeDF$numBoardings)
```

Histogram of routeDF\$numBoardings



```
hist(routeDF.weekdays$numBoardings)
```

Histogram of routeDF.weekdays\$numBoardings



```
summary(routeDF$numBoardings)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0   105.0   282.0   493.2   626.5   4410.0
```

```
summary(routeDF.weekdays$numBoardings)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0   102.0   295.0   524.2   683.8   4410.0
```

```
summary(routeDF.sunday$numBoardings)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0    98.0   233.0   355.7   419.0   3140.0
```

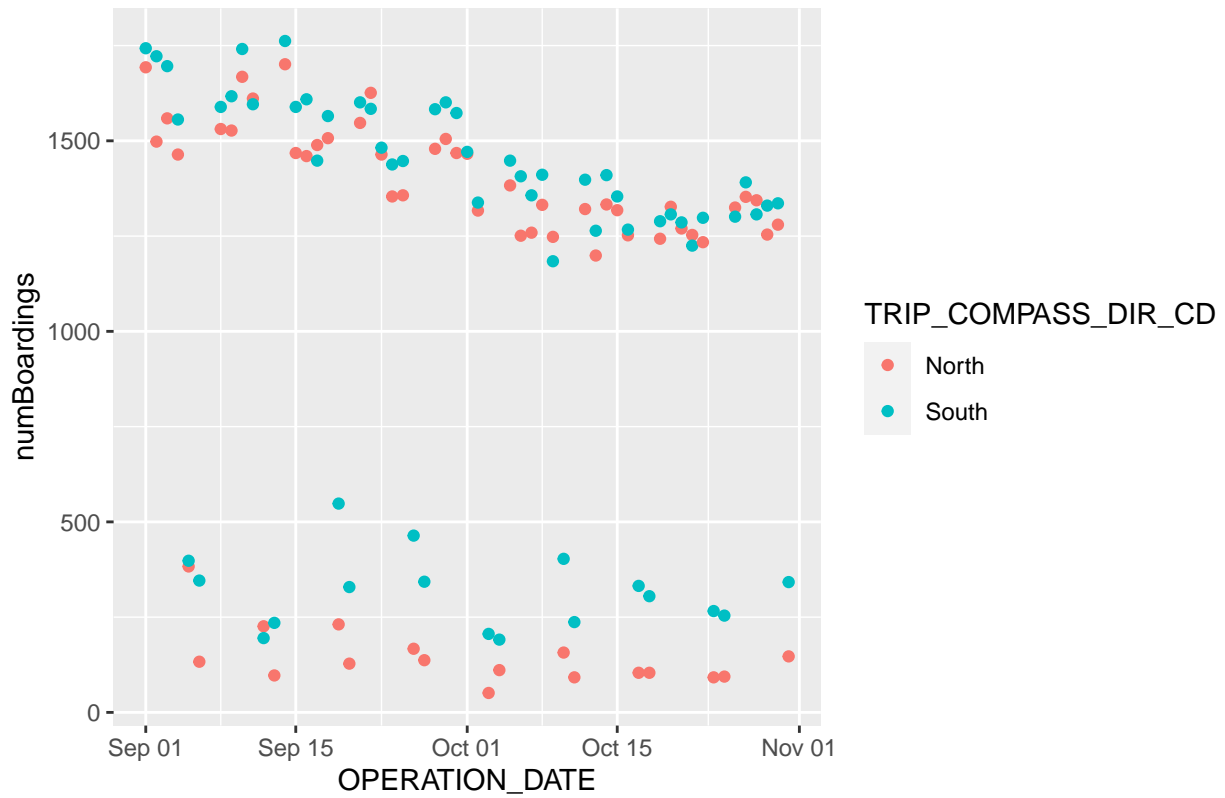
```
summary(routeDF.saturday$numBoardings)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.0   122.5   279.0   424.8   523.0   3662.0
```

Examining Single-Route Data – Route 3

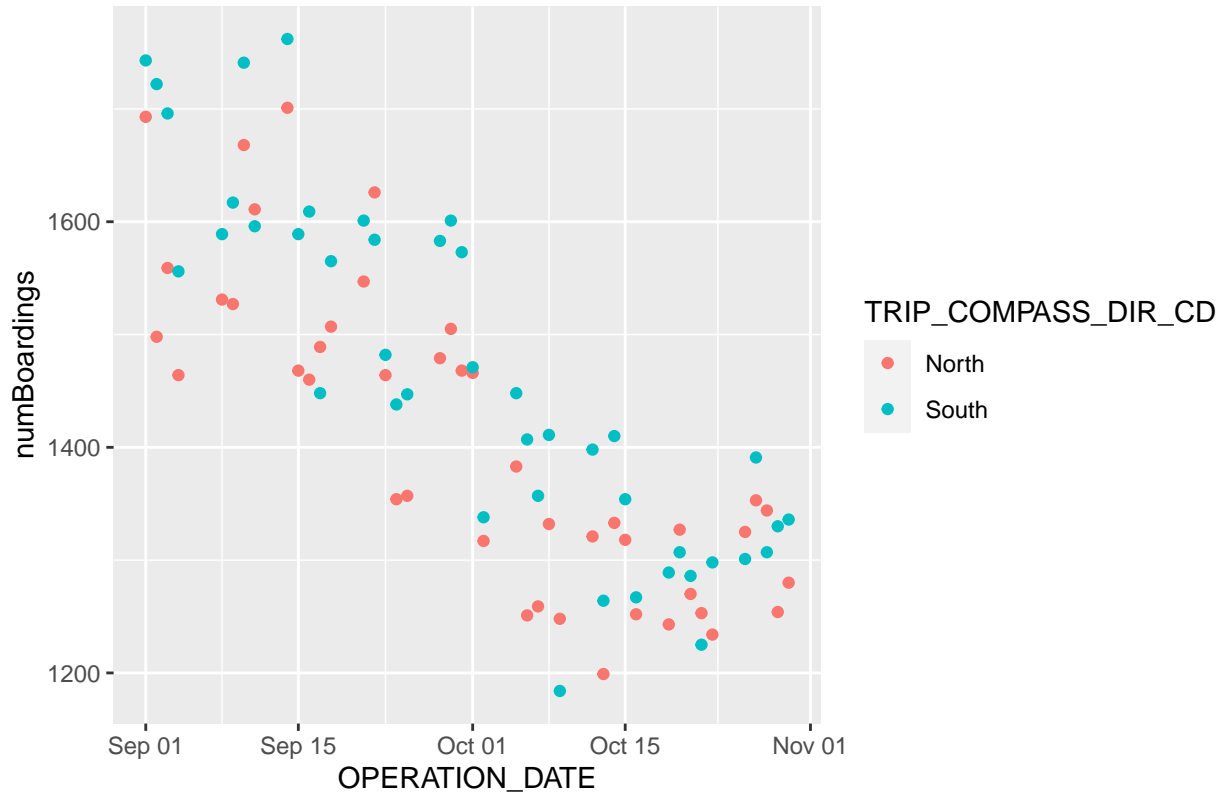
```
make_route_scatter(3, routeDF)
```

Bus Route #3



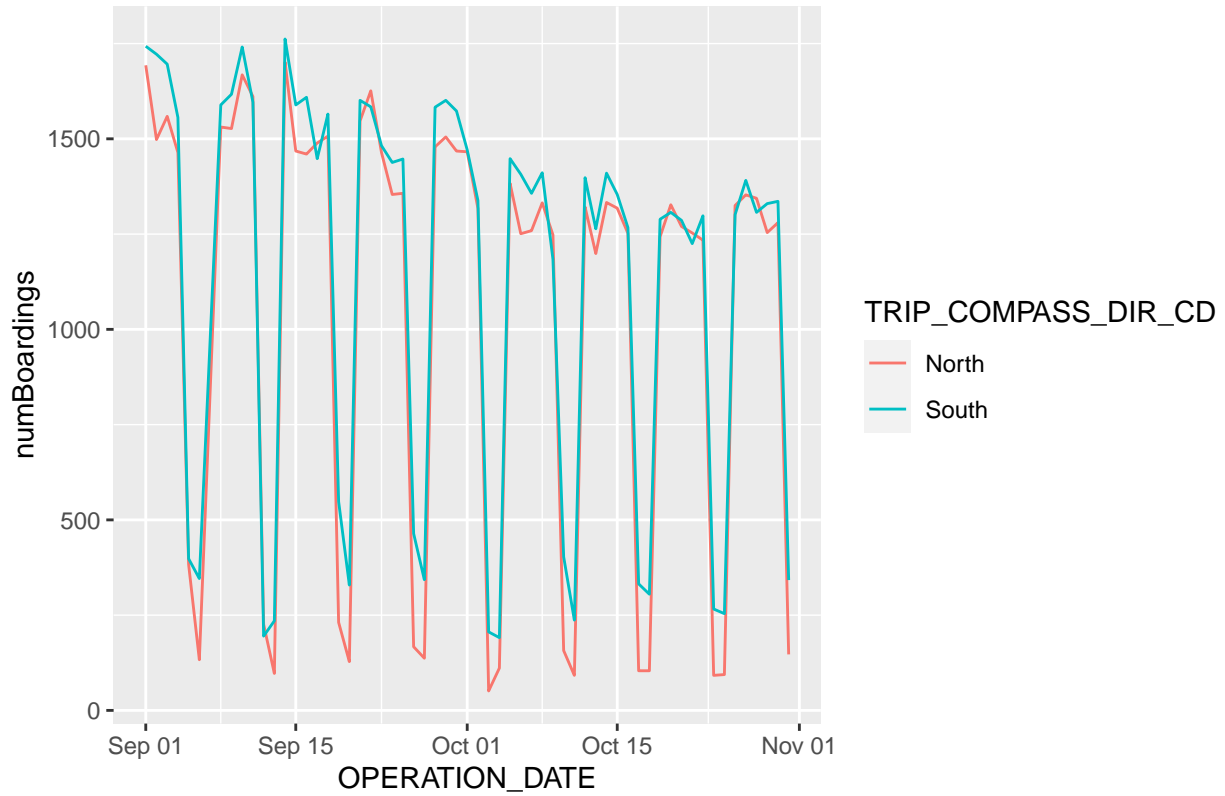
```
make_route_scatter(3, routeDF.weekdays)
```

Bus Route #3



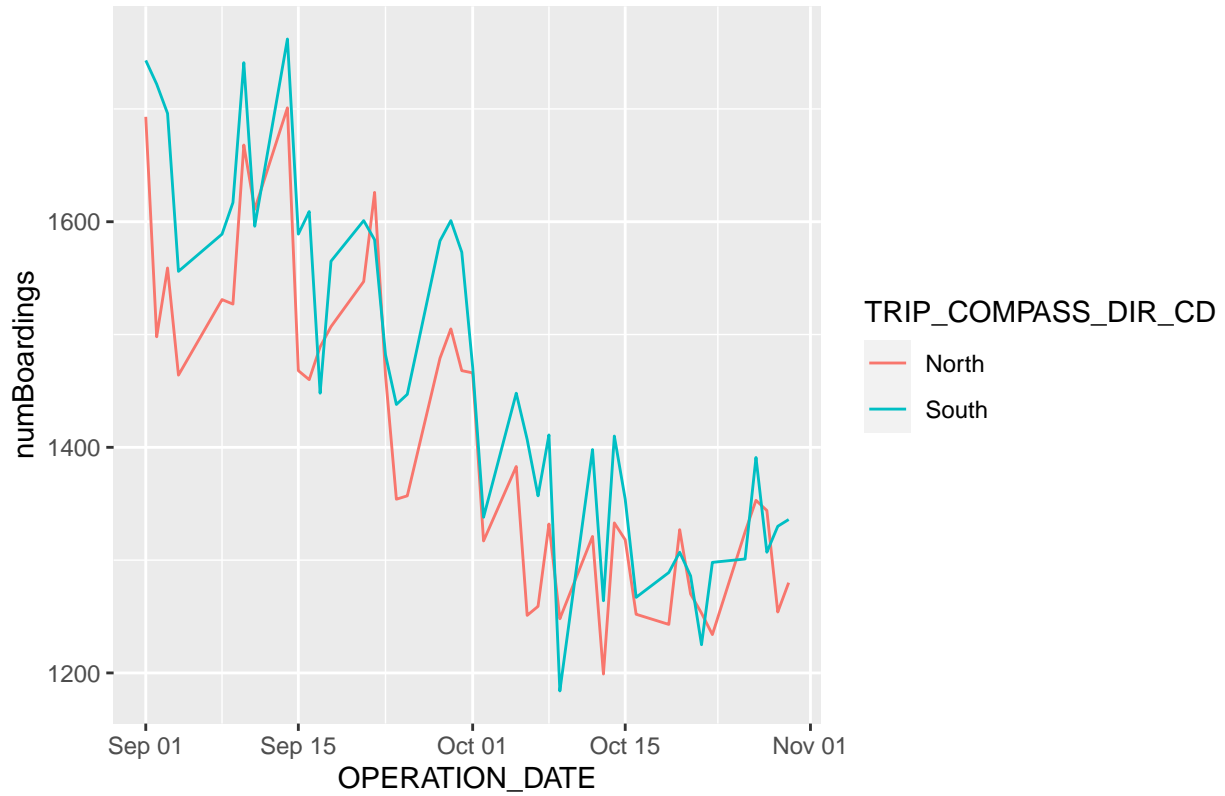
```
make_route_line(3, routeDF)
```

Bus Route #3



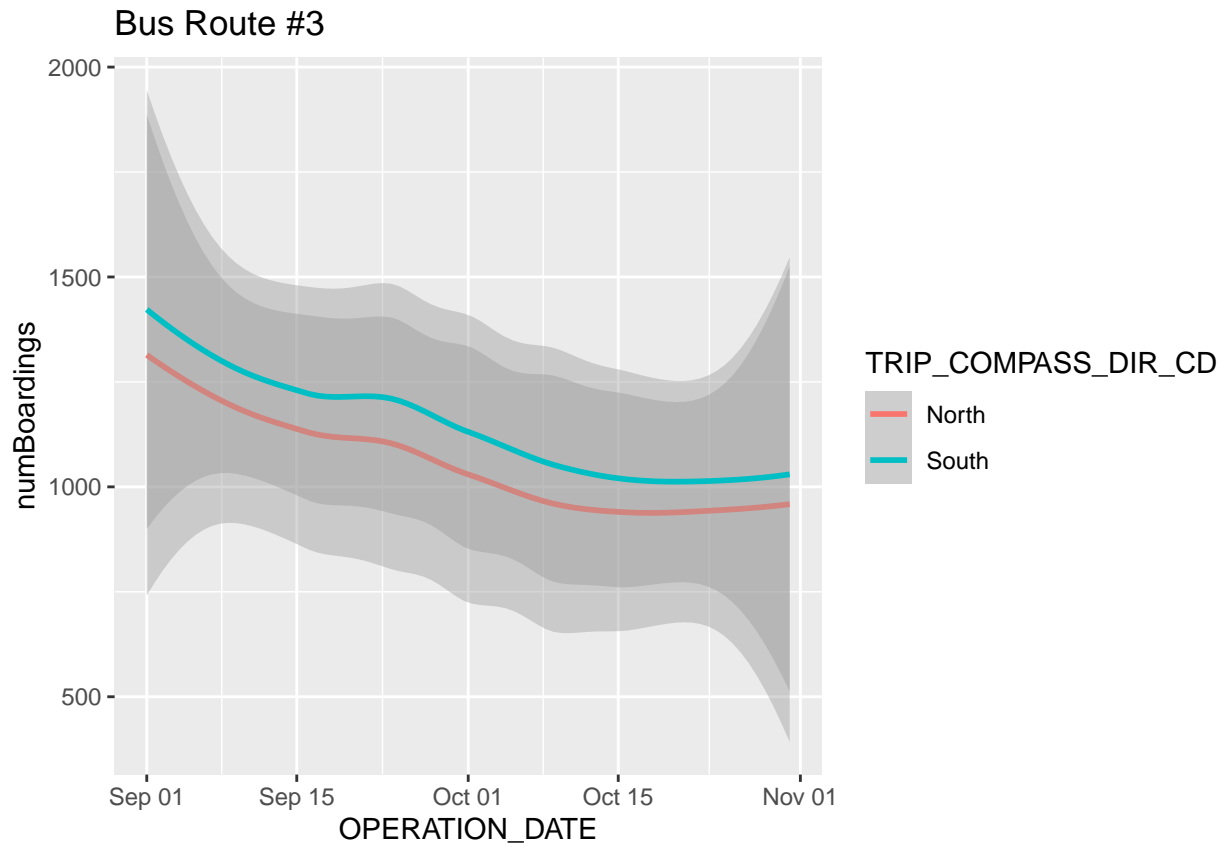
```
make_route_line(3, routeDF.weekdays)
```

Bus Route #3



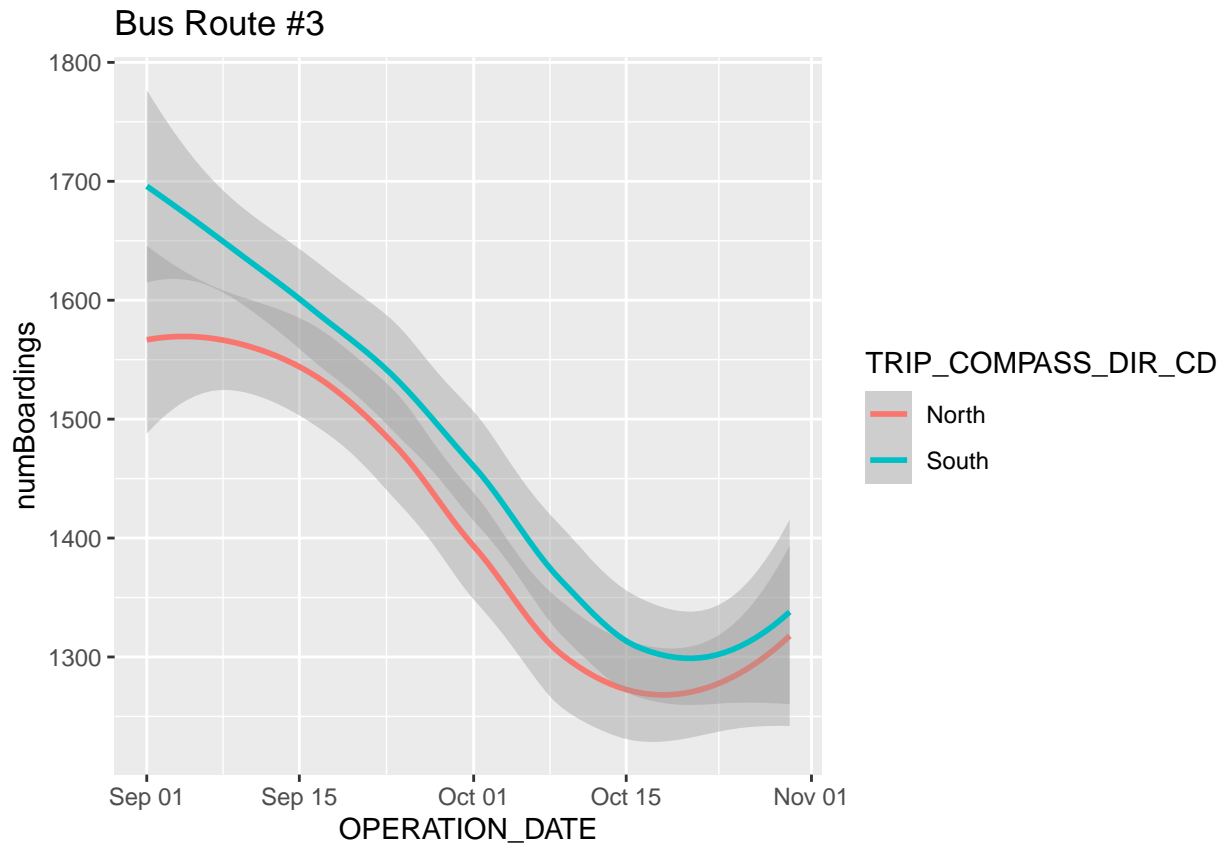
```
make_route_smooth(3, routeDF)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



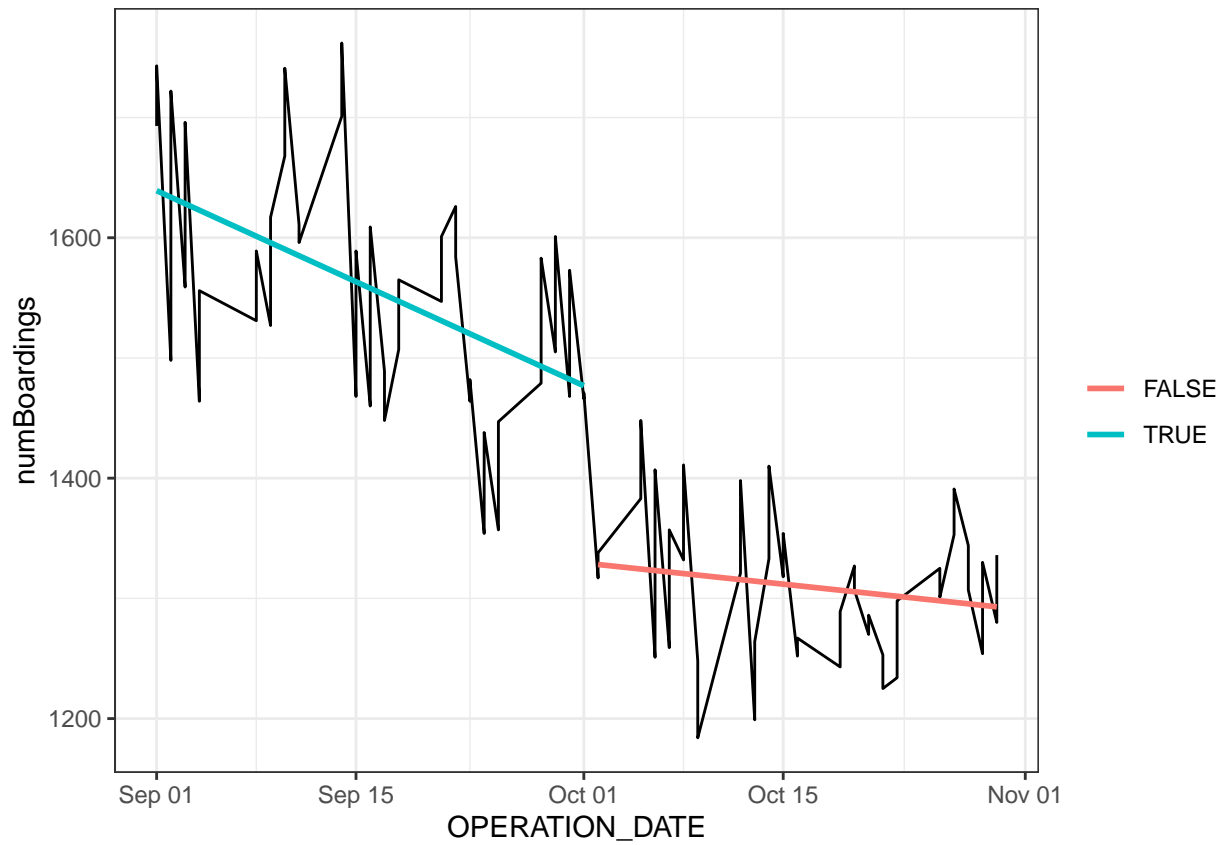
```
make_route_smooth(3, routeDF.weekdays)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
make_route_ts(3, routeDF.weekdays) #interrupts a decline then recovery
```

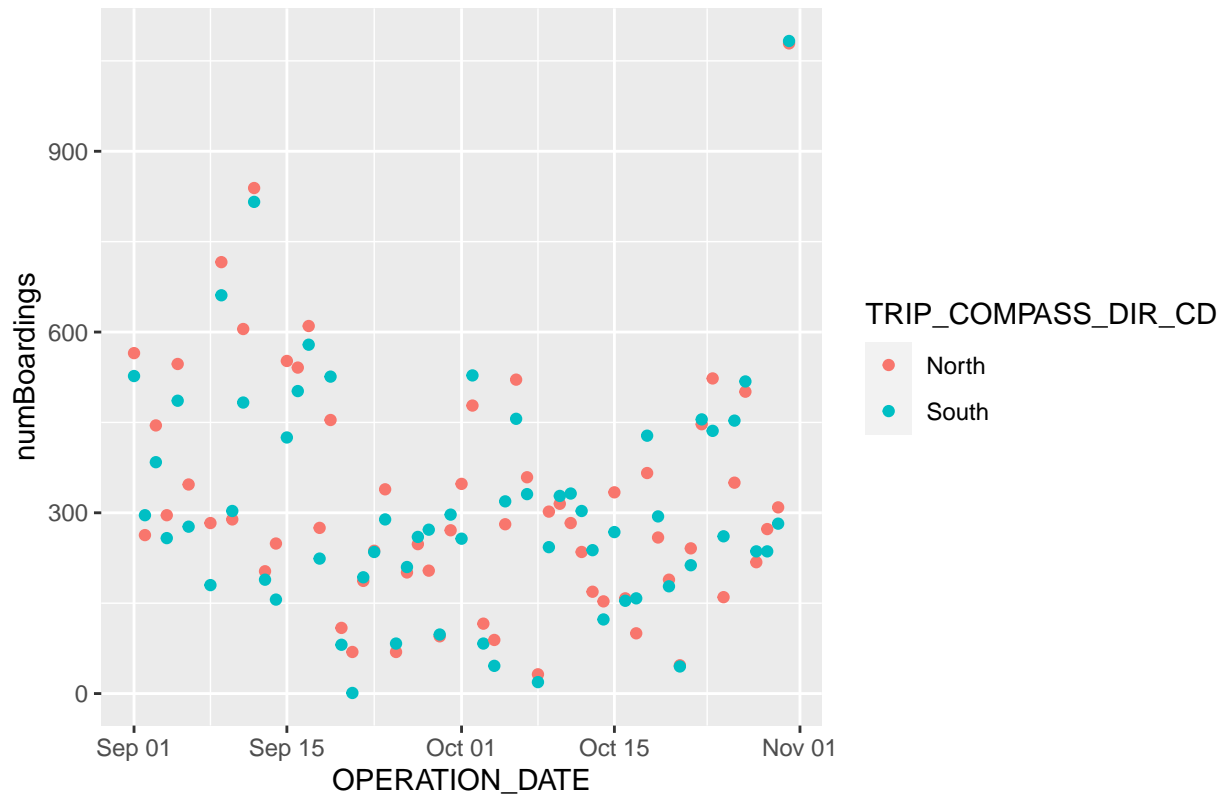
```
## `geom_smooth()` using formula 'y ~ x'
```



Examining Single-Route Data – Route 120

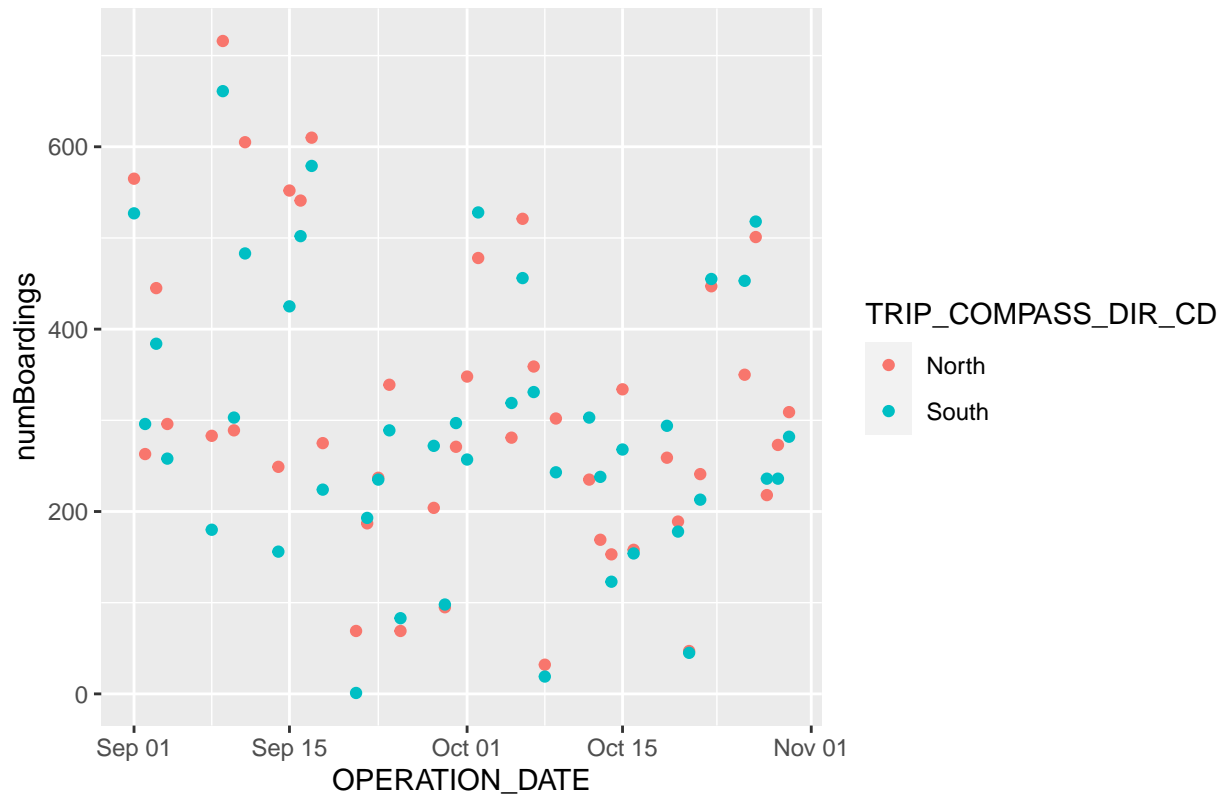
```
make_route_scatter(120, routeDF)
```

Bus Route #120



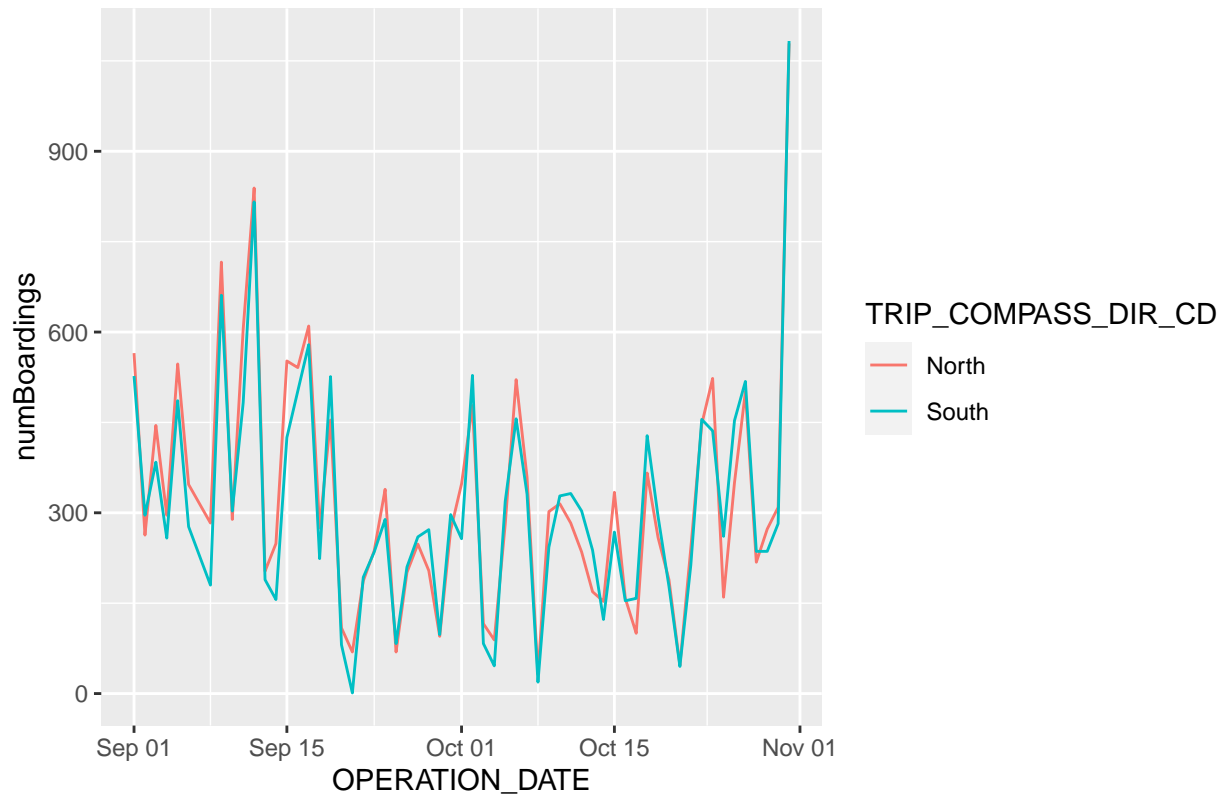
```
make_route_scatter(120, routeDF.weekdays)
```

Bus Route #120



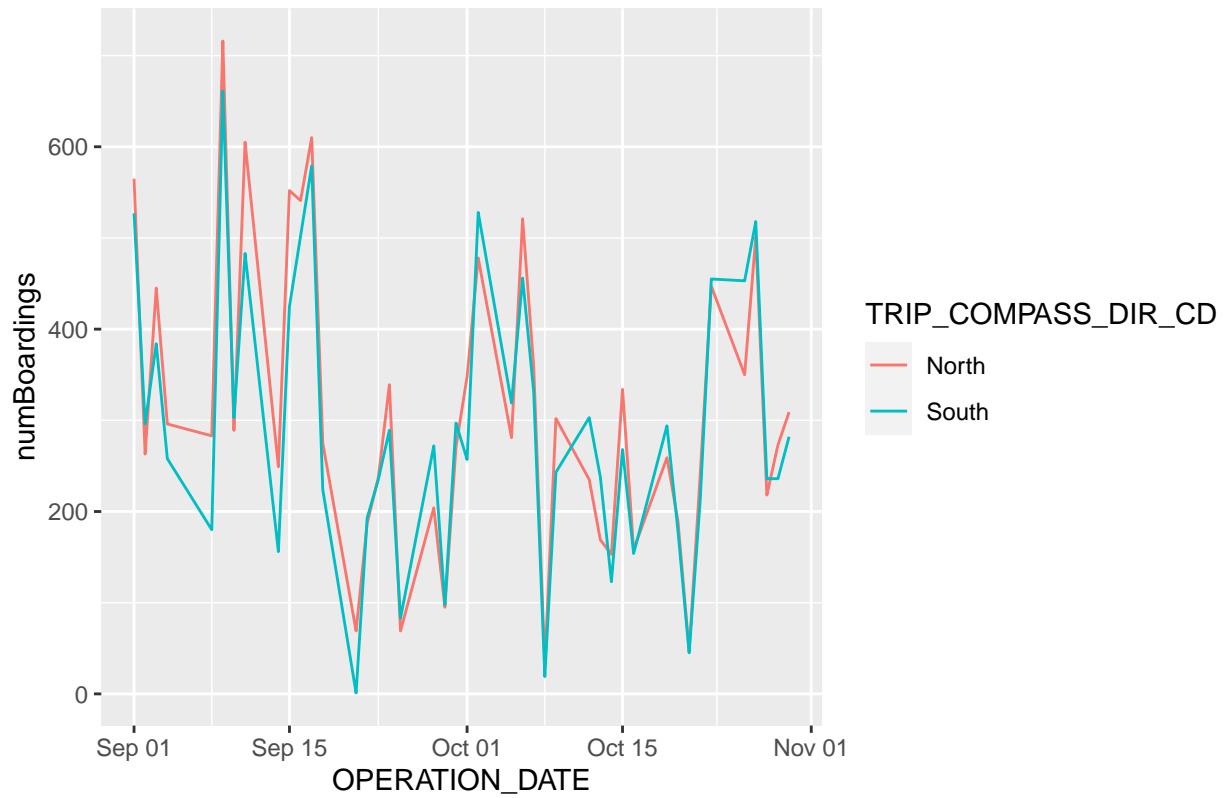
```
make_route_line(120, routeDF)
```

Bus Route #120



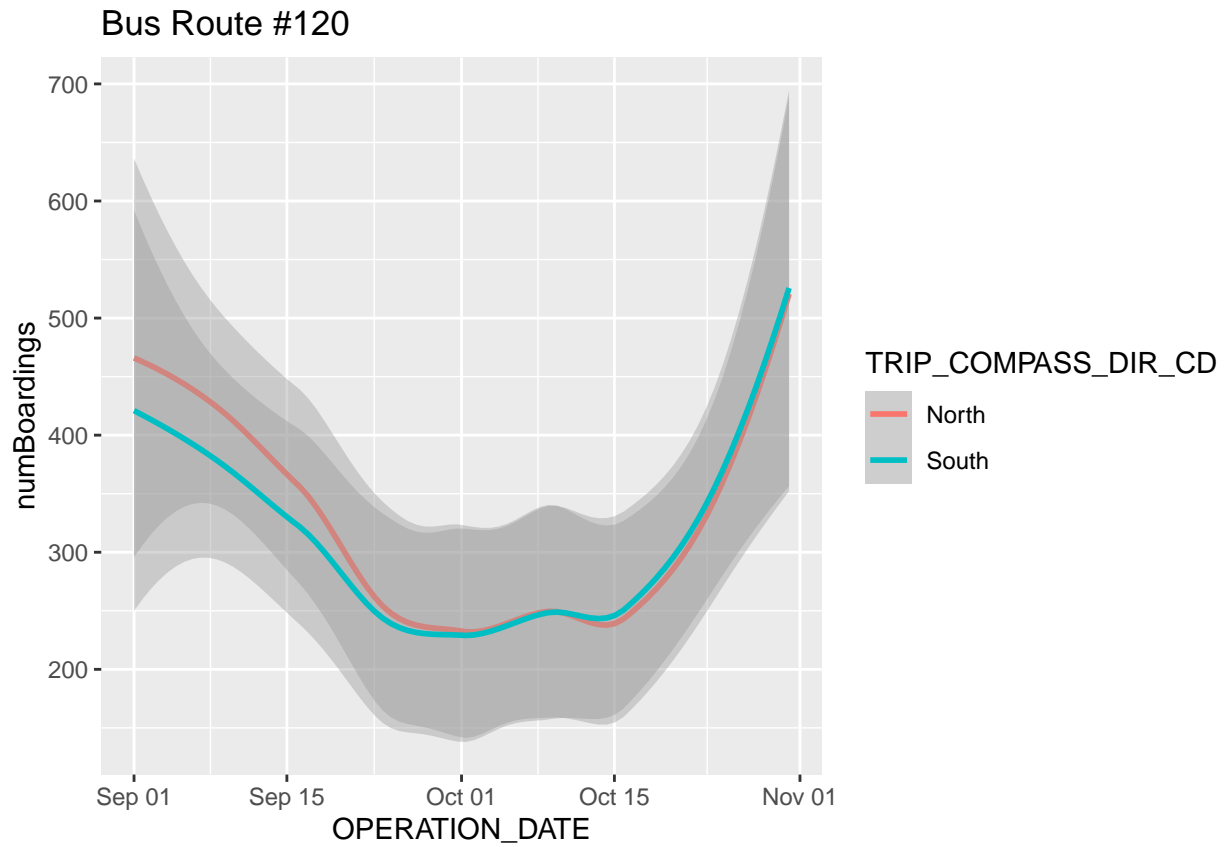
```
make_route_line(120, routeDF.weekdays)
```

Bus Route #120



```
make_route_smooth(120, routeDF)
```

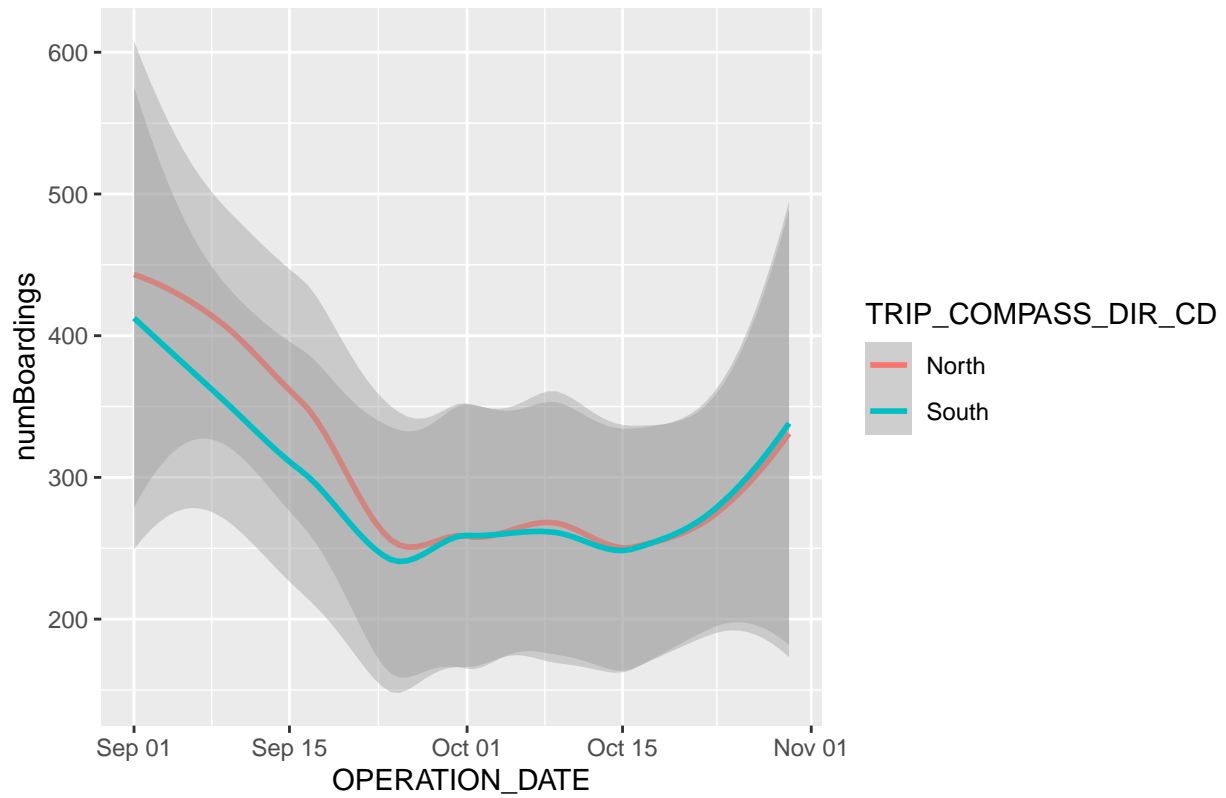
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
make_route_smooth(120, routeDF.weekdays)
```

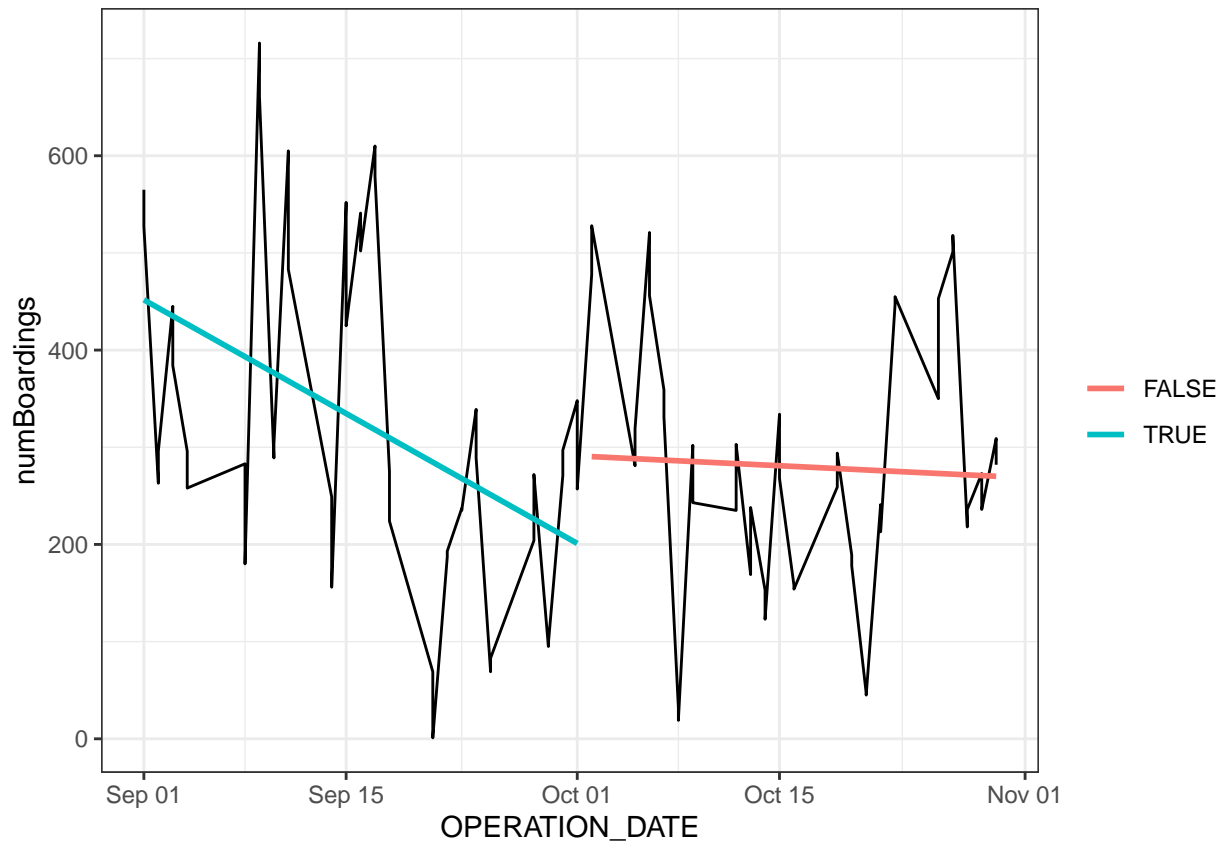
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Bus Route #120



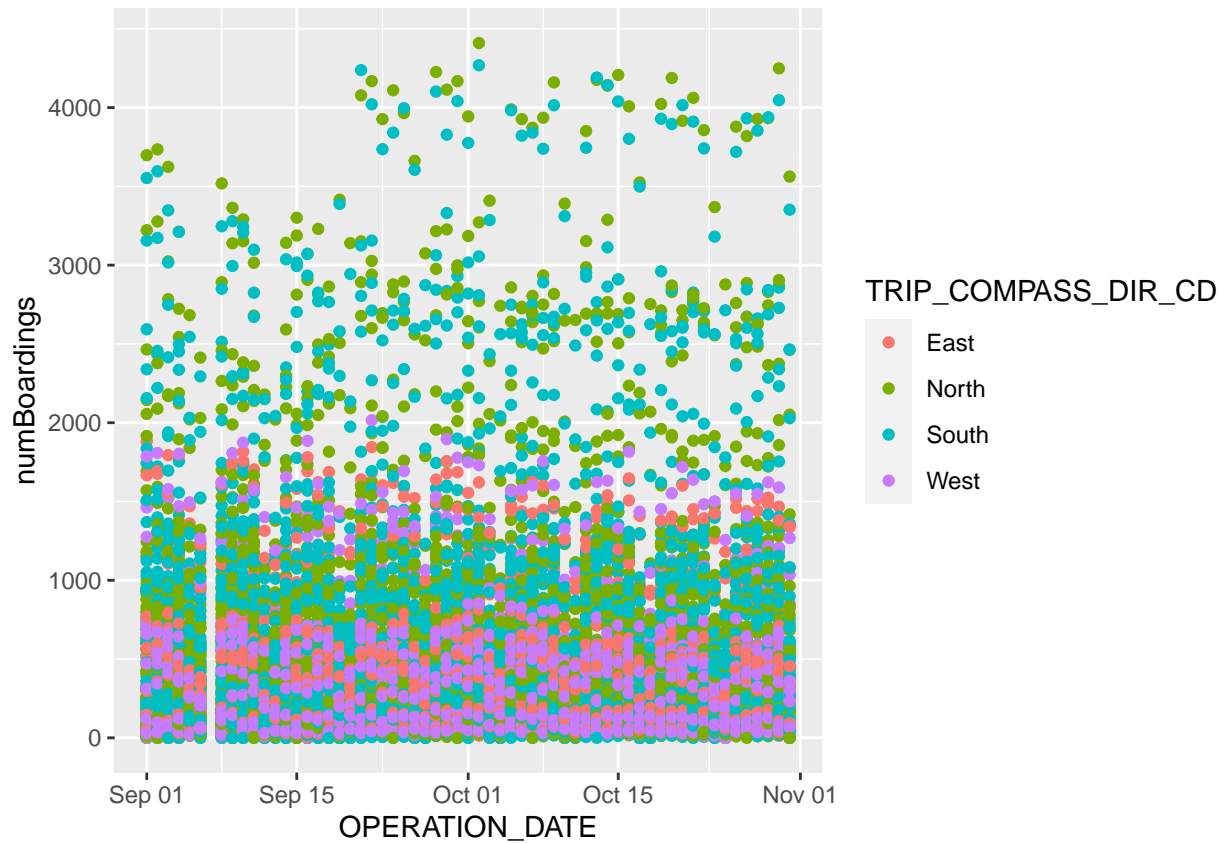
```
make_route_ts(120, routeDF.weekdays)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

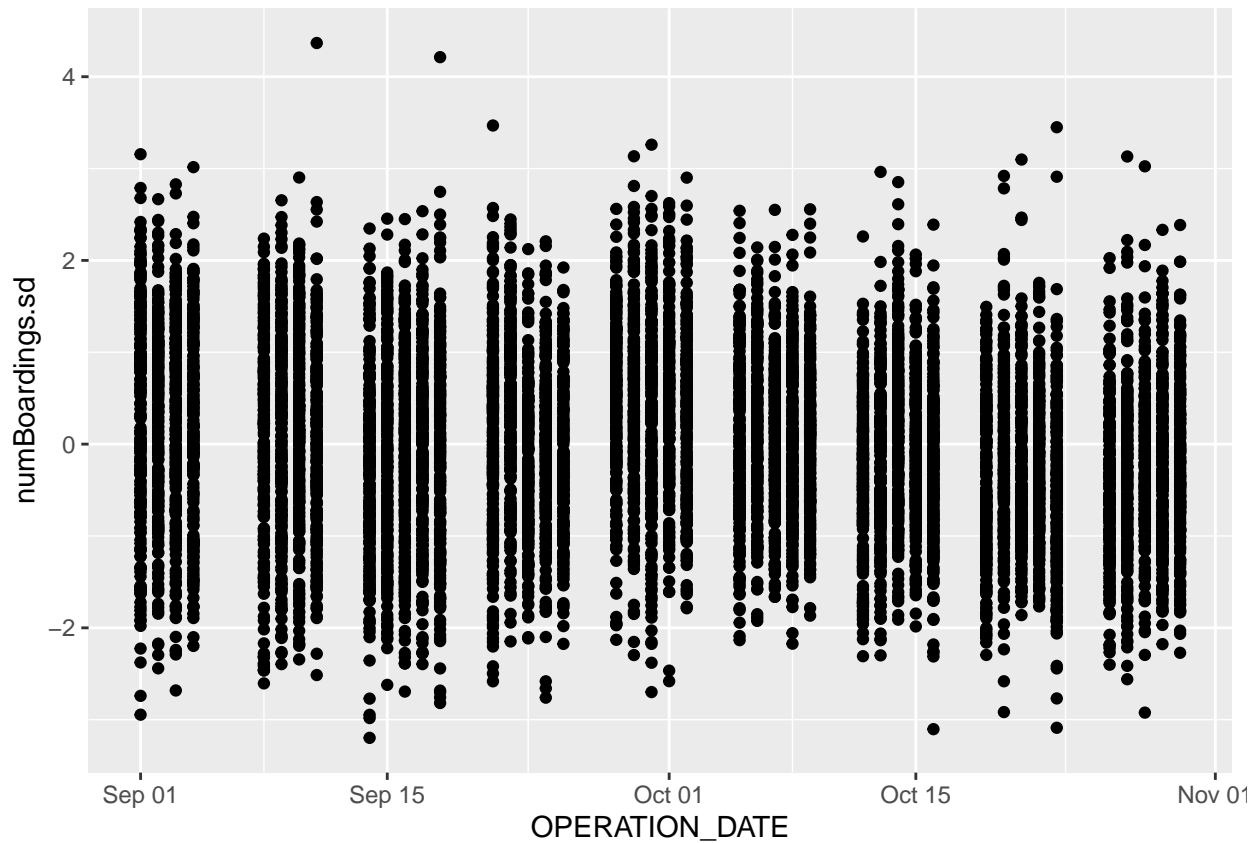
Examining System-wide data

```
g <- ggplot(routeDF, aes(x=OPERATION_DATE, y=numBoardings, color=TRIP_COMPASS_DIR_CD)) +  
  geom_point()  
g
```



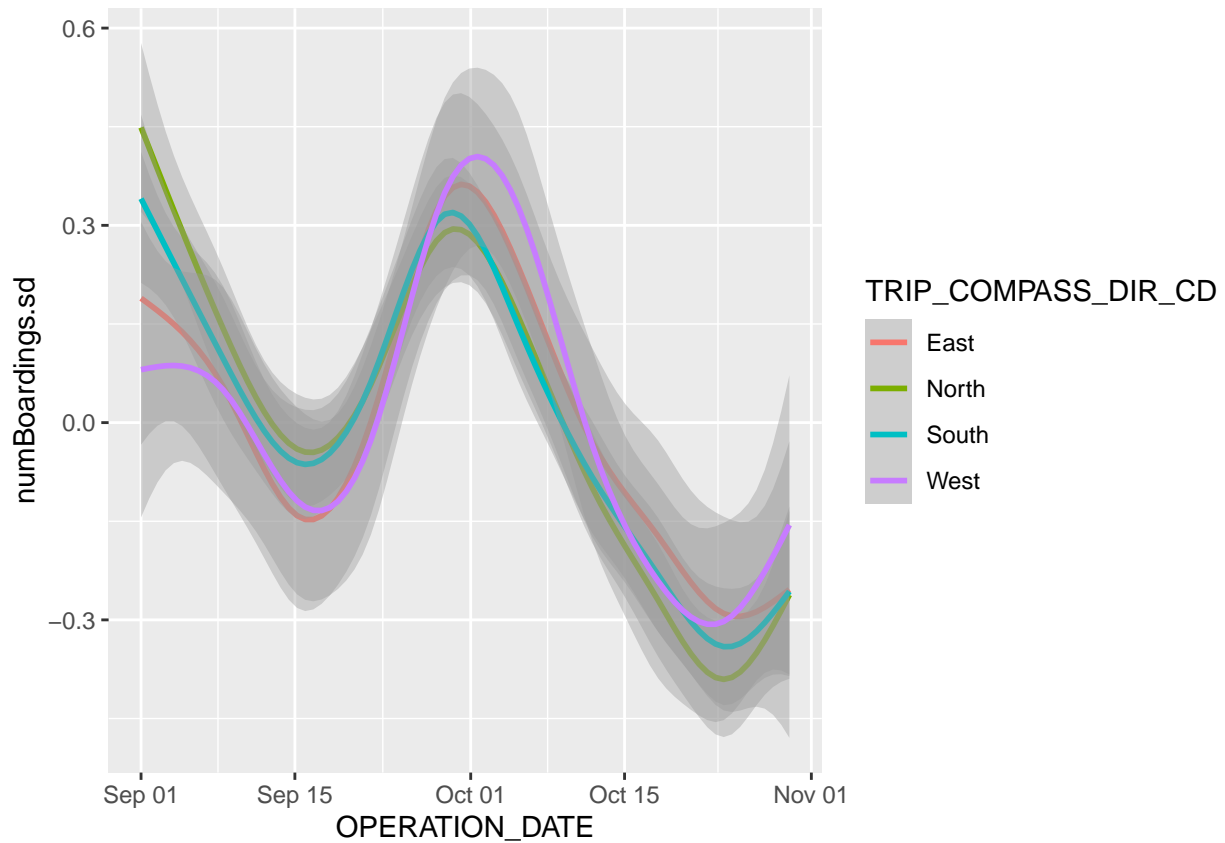
```
#presumably what we really want is sd units
g <- ggplot(routedDF.weekdays, aes(x=OPERATION_DATE, y=numBoardings.sd)) +
  geom_point()
g
```

Warning: Removed 25 rows containing missing values (geom_point).



```
g <- ggplot(routeDF.weekdays, aes(x=OPERATION_DATE, y=numBoardings.sd, color=TRIP_COMPASS_DIR_CD)) +
  geom_smooth()
g
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## Warning: Removed 25 rows containing non-finite values (stat_smooth).
```



Taking some early peeks at the time series data

```
g <- ggplot(routedF.weekdays.filt, aes(x = OPERATION_DATE, y = numBoardings.sd)) +
  geom_line() +
  geom_smooth(method="lm", se=FALSE, aes(colour=treated)) +
  theme_bw() +
  labs(colour="")
```

g

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 24 rows containing non-finite values (stat_smooth).
```

