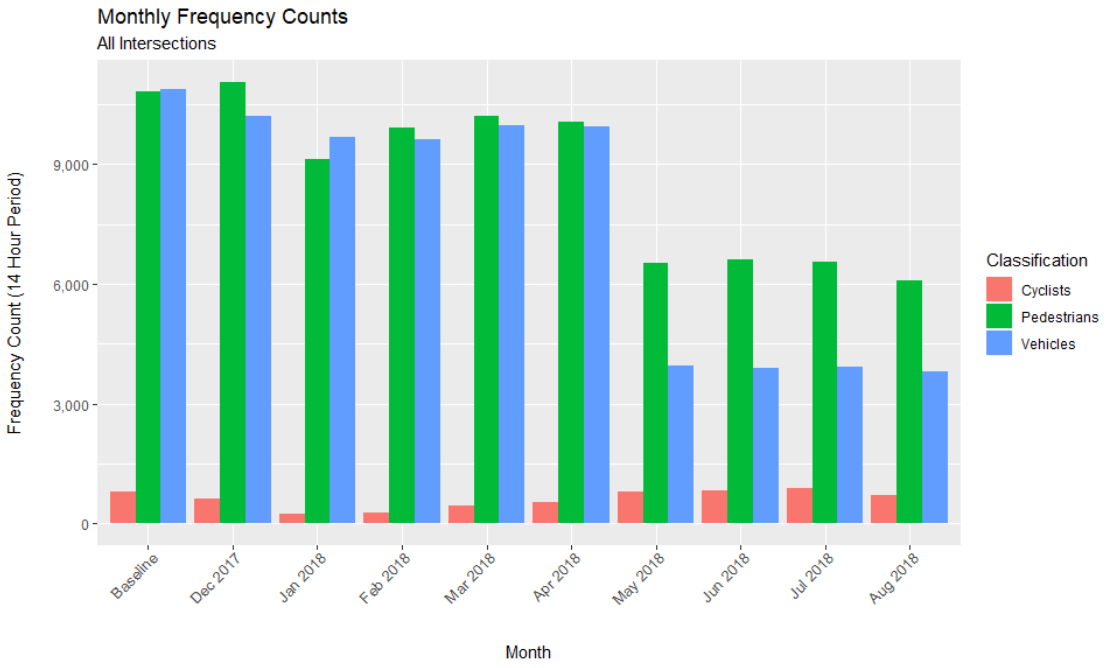
Source Code: King_St_Pilot_Volume

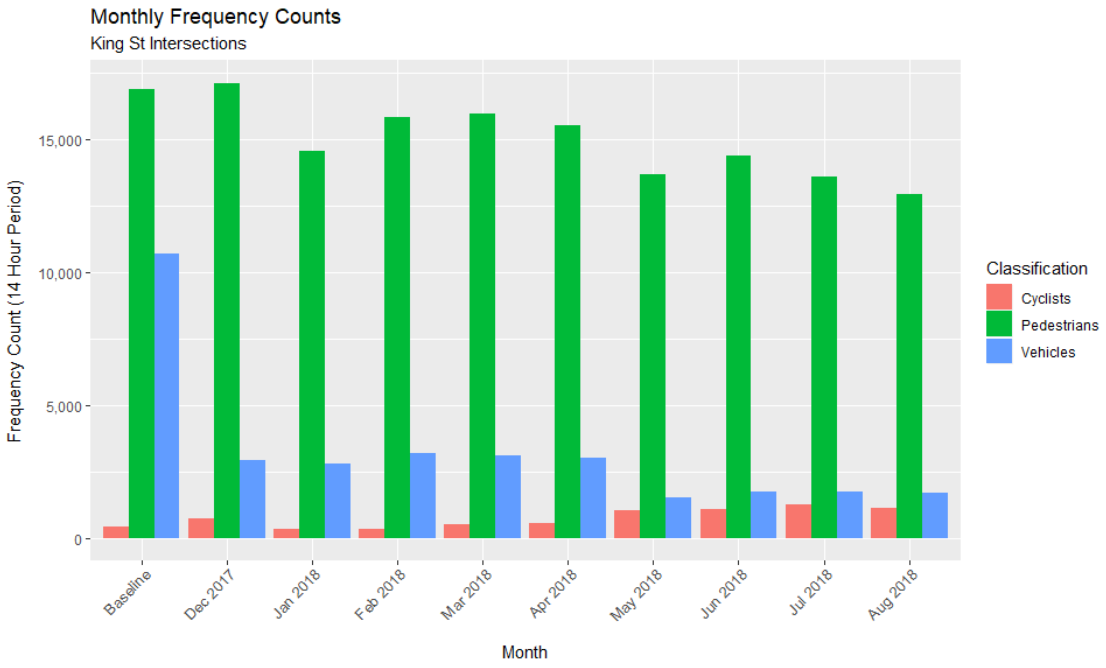
```
knitr::opts_chunk$set(fig.fullwidth = TRUE, fig.width=10, fig.height=6)

library("dplyr")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library("reshape2")
library("ggplot2")
library("plyr")

## ------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first,
then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

#Average monthly count for all intersections
count = read.csv(file ="count_orig.csv")

#Filter by 14 hour period to avoid duplicate counts
count_monthly = filter(count, period_name %in% c("Morning (08:00-12:00)",
"Afternoon (12:00-17:00)", "Evening (17:00-22:00)"))

#To sort data by month, first separate baseline data
#Daily count for each classification at each intersection
temp1 = xtabs(formula =
volume~classification+aggregation_period+intersection_name, data =
count_monthly)
```

```r
temp1 = as.data.frame(temp1)

#Average daily count for each classification for all intersections by month
temp2 = xtabs(formula = Freq~classification+aggregation_period,
aggregate(Freq~classification+aggregation_period, temp1, mean))

#Save Baseline data
Baseline = temp2[,'Baseline']

#Sort data by month
count_monthly$aggregation_period <- factor(count_monthly$aggregation_period,
levels=paste(month.abb, sort(rep(2017:2018, each=12))))

#Daily count for each classification at each intersection
temp1 = xtabs(formula =
volume~classification+aggregation_period+intersection_name, data =
count_monthly)
temp1 = as.data.frame(temp1)

#Average daily count for each classification for all intersections by month
temp2 = xtabs(formula = Freq~classification+aggregation_period,
aggregate(Freq~classification+aggregation_period, temp1, mean))

#Add Baseline back into the data
vol_sum = cbind(Baseline, temp2[,12:20])

#Change to dataframe in order to plot
df = as.data.frame(vol_sum)
df <- melt(df)

## No id variables; using all as measure variables

df$classification <- c('Cyclists', 'Pedestrians', 'Vehicles')

#Classification Frequency Count per Month - All Intersections
ggplot(df, aes(x = variable, y = round(value, digits = 0), fill =
classification)) + geom_col(stat = 'identity', position = 'dodge') + labs(x =
'Month', y = 'Frequency Count (14 Hour Period)', title = "Monthly Frequency
Counts", subtitle = "All Intersections") + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 20, b = 0, l = 0)), axis.text.x =
element_text(size = 10, angle = 45, hjust = 1, vjust = 1, margin = margin(t =
0, r = 0, b = 20, l = 0))) + scale_y_continuous(labels = scales::comma) +
guides(fill = guide_legend(title = "Classification"))

## Warning: Ignoring unknown parameters: stat
```

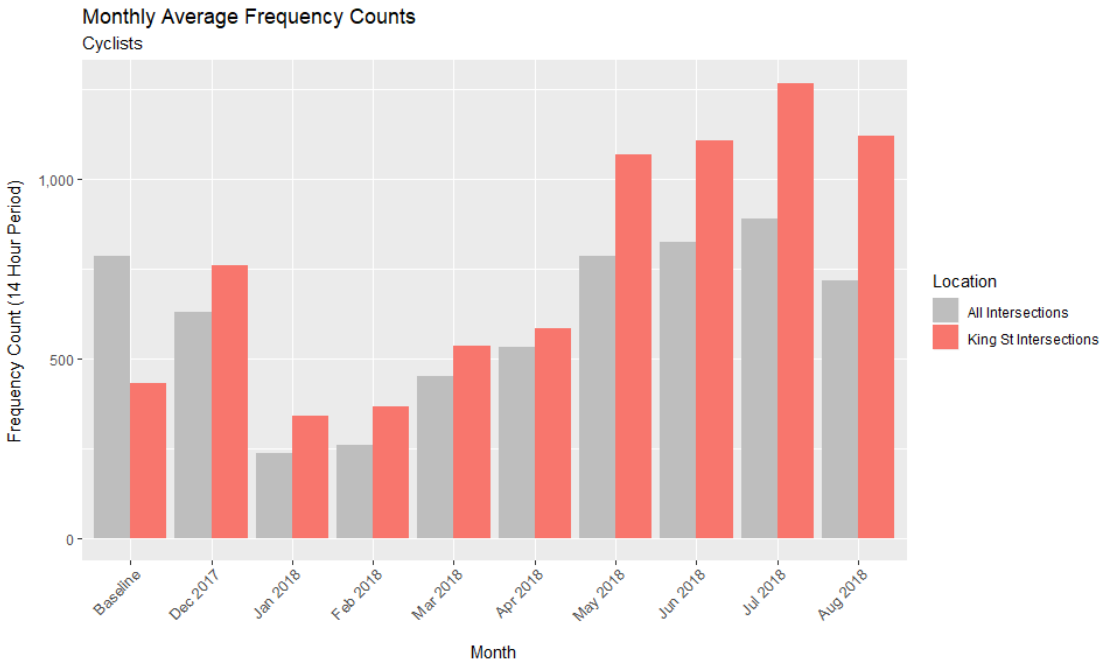



```
#Average Monthly Count - King St
count = read.csv(file ="count_orig.csv")

#Filter for King St intersections
count_king = filter(count, intersection_name %in% c("Bay St / King St W",
"Spadina Ave / King St W", "Bathurst St / King St W"))

#ilter for 14 hour period to avoid duplicate counts
count_monthly = filter(count_king, period_name %in% c("Morning (08:00-
12:00)", "Afternoon (12:00-17:00)", "Evening (17:00-22:00)"))

#Daily count for each classification at each intersection
temp1 = xtabs(formula =
volume~classification+aggregation_period+intersection_name, data =
count_monthly)
temp1 = as.data.frame(temp1)
temp1 = temp1[temp1$Freq != 0,]

#Average daily count for each classification for all King intersections
temp2 = xtabs(formula = Freq~classification+aggregation_period,
aggregate(Freq~classification+aggregation_period, temp1, mean))

#Save Baseline data
Baseline = temp2[,'Baseline']

#Sort data by month
count_monthly$aggregation_period <- factor(count_monthly$aggregation_period,
levels=paste(month.abb, sort(rep(2017:2018, each=12))))
```

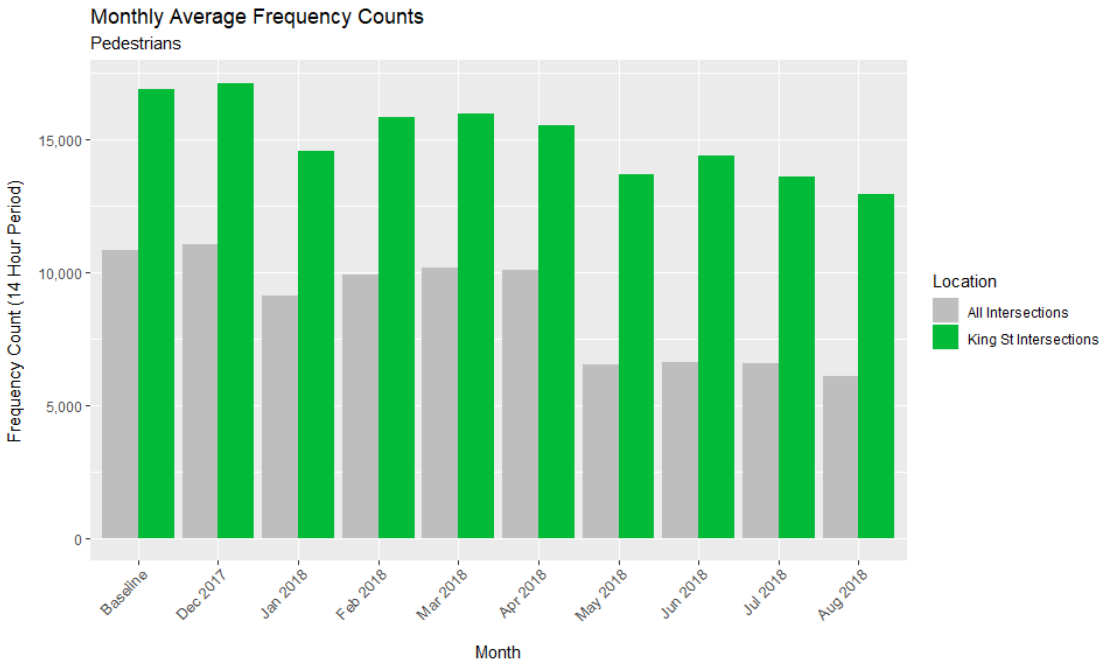
```r
#Daily count for each classification at each intersection
temp1 = xtabs(formula =
volume~classification+aggregation_period+intersection_name, data =
count_monthly)
temp1 = as.data.frame(temp1)
temp1 = temp1[temp1$Freq != 0,]

#Average daily count for each classification for all King intersections
temp2 = xtabs(formula = Freq~classification+aggregation_period,
aggregate(Freq~classification+aggregation_period, temp1, mean))

#Add Baseline back into the data
vol_sum = cbind(Baseline, temp2[,12:20])

#Change to dataframe in order to plot
df_king = as.data.frame(vol_sum)
df_king <- melt(df_king)

## No id variables; using all as measure variables

df_king$classification <- c('Cyclists', 'Pedestrians', 'Vehicles')
#Rename value to identify as value for King St
colnames(df_king)[colnames(df_king) == "value"] <- "king_value"

#Classification Frequency Count per Month - King St
ggplot(df_king, aes(x = variable, y = round(king_value, digits = 0), fill =
classification)) + geom_col(stat = 'identity', position = 'dodge') + labs(x =
'Month', y = 'Frequency Count (14 Hour Period)', title = "Monthly Frequency
Counts", subtitle = "King St Intersections") + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
element_text(size = 10, angle = 45, hjust = 1, vjust = 1, margin = margin(t =
0, r = 0, b = 10, l = 0))) + scale_y_continuous(labels = scales::comma) +
guides(fill = guide_legend(title = "Classification"))

## Warning: Ignoring unknown parameters: stat
```



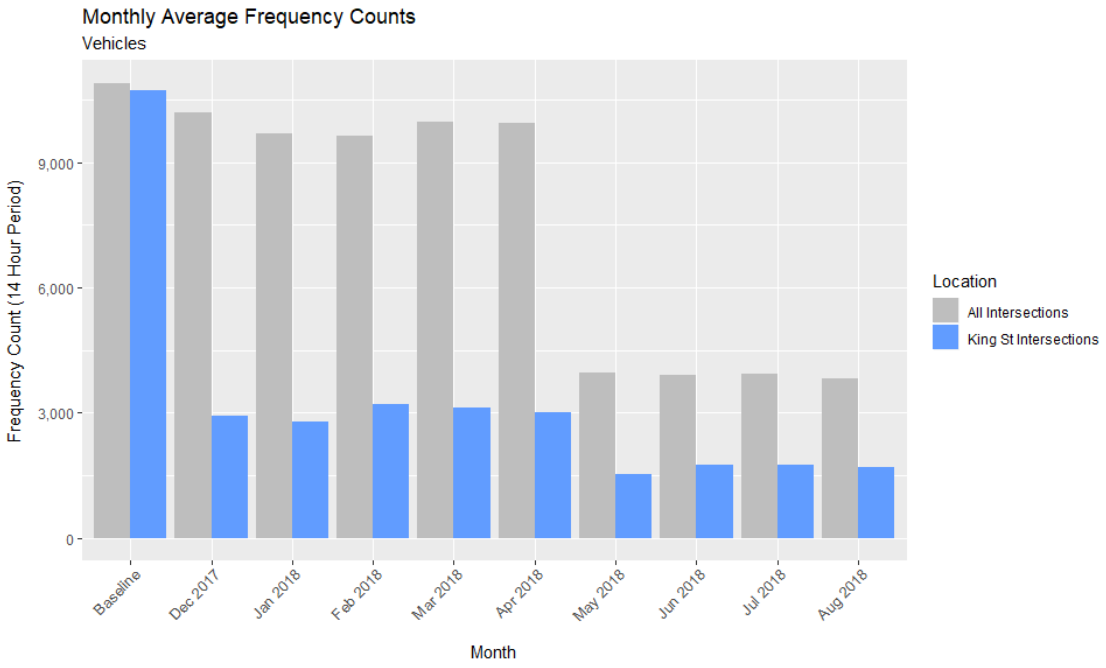

```
#Closer look at each classification
comparison = left_join(df, df_king[,c("variable", "king_value",
"classification")], by = c("variable", "classification"))

#Cyclists
comparison_cycl = comparison[comparison$classification == 'Cyclists',]

df_cycl = melt(comparison_cycl[, c('variable','value','king_value')], id.vars
= 1)
colnames(df_cycl)[1] <- "period"

#Classification Frequency Count per Month - King St
ggplot(df_cycl, aes(x = period, y = round(value, digits = 0), fill =
factor(variable, labels = c("All Intersections", "King St Intersections"))))
+ geom_col(stat = 'identity', position = 'dodge') + labs(color = "Location",
x = 'Month', y = 'Frequency Count (14 Hour Period)', title = "Monthly Average
Frequency Counts", subtitle ="Cyclists") + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
element_text(size = 10, angle = 45, hjust = 1, vjust = 1, margin = margin(t =
0, r = 0, b = 10, l = 0))) + scale_y_continuous(labels = scales::comma) +
scale_fill_manual(values = c('grey', '#F8766D')) + guides(fill =
guide_legend(title = "Location"))

## Warning: Ignoring unknown parameters: stat
```




```
#Pedestrians
comparison_ped = comparison[comparison$classification == 'Pedestrians',]

df_ped = melt(comparison_ped[, c('variable','value','king_value')], id.vars =
1)
colnames(df_ped)[1] <- "period"

#Classification Frequency Count per Month - King St
ggplot(df_ped, aes(x = period, y = round(value, digits = 0), fill =
factor(variable, labels = c("All Intersections", "King St Intersections"))))
+ geom_col(stat = 'identity', position = 'dodge') + labs(color = "Location",
x = 'Month', y = 'Frequency Count (14 Hour Period)', title = "Monthly Average
Frequency Counts", subtitle = "Pedestrians") + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
element_text(size = 10, angle = 45, hjust = 1, vjust = 1, margin = margin(t =
0, r = 0, b = 10, l = 0))) + scale_y_continuous(labels = scales::comma) +
scale_fill_manual(values = c('grey', '#00BA38')) + guides(fill =
guide_legend(title = "Location"))

## Warning: Ignoring unknown parameters: stat
```

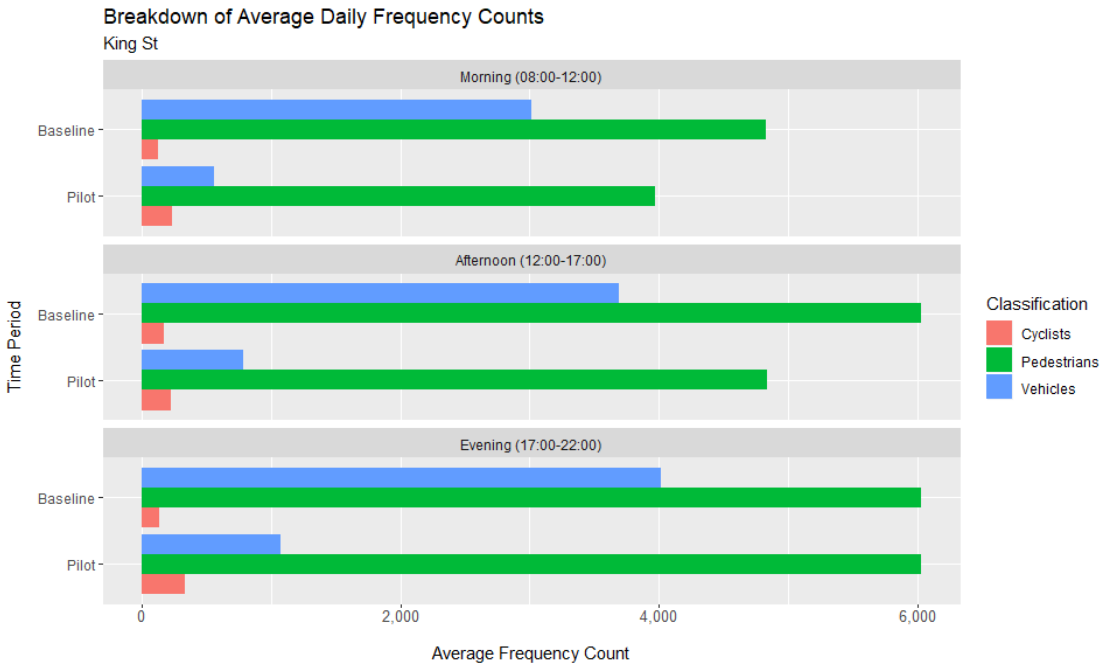



```
#Vehicles
comparison_veh = comparison[comparison$classification == 'Vehicles',]

df_veh = melt(comparison_veh[, c('variable','value','king_value')], id.vars =
1)
colnames(df_veh)[1] <- "period"

#Classification Frequency Count per Month - King St
ggplot(df_veh, aes(x = period, y = round(value, digits = 0), fill =
factor(variable, labels = c("All Intersections", "King St Intersections"))))
+ geom_col(stat = 'identity', position = 'dodge') + labs(color = "Location",
x = 'Month', y = 'Frequency Count (14 Hour Period)', title = "Monthly Average
Frequency Counts", subtitle ="Vehicles") + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
element_text(size = 10, angle = 45, hjust = 1, vjust = 1, margin = margin(t =
0, r = 0, b = 10, l = 0))) + scale_y_continuous(labels = scales::comma) +
scale_fill_manual(values = c('grey', '#619CFF')) + guides(fill =
guide_legend(title = "Location"))

## Warning: Ignoring unknown parameters: stat
```



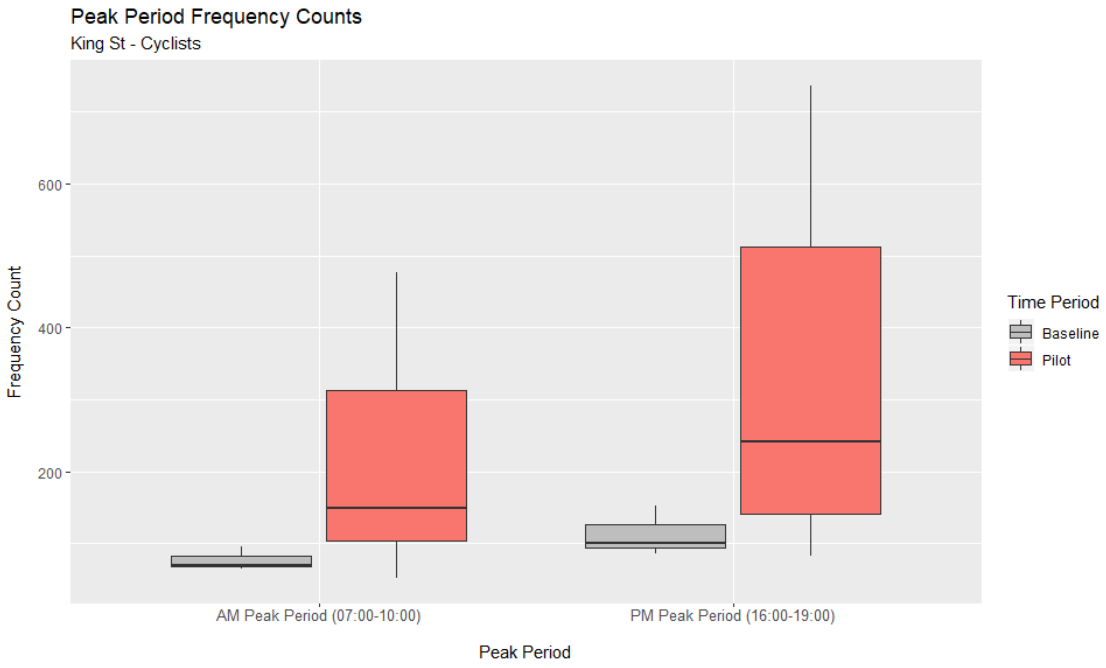

```
#King St Daily Summary
count = read.csv(file ="count_orig.csv")
count_king = filter(count, intersection_name %in% c("Bay St / King St W",
"Spadina Ave / King St W", "Bathurst St / King St W"))
count_monthly = filter(count_king, period_name %in% c("Morning (08:00-
12:00)", "Afternoon (12:00-17:00)", "Evening (17:00-22:00)"))

#Daily count for each classification at each intersection on King
temp1 = xtabs(formula =
volume~classification+aggregation_period+intersection_name+period_name, data
= count_monthly)
temp1 = as.data.frame(temp1)
temp1 = temp1[temp1$Freq != 0,]

#Average daily count for each classification for all King intersections
temp2 = xtabs(formula = Freq~classification+aggregation_period+period_name,
aggregate(Freq~classification+aggregation_period+period_name, temp1, mean))
temp2 = as.data.frame(temp2)
temp2 = temp2[temp2$Freq != 0,]

#Save Baseline data
Baseline = temp2[temp2$aggregation_period == 'Baseline',]
colnames(Baseline)[colnames(Baseline) == "aggregation_period"] <- "period"

#Remove Baseline data from data set to isolate Pilot time period
temp3 = temp2[temp2$aggregation_period != 'Baseline',]
#Average frequency count during Pilot
temp3 = xtabs(formula = Freq~classification+period_name,
aggregate(Freq~classification+period_name, temp3, mean))
```

```r
temp3 = as.data.frame(temp3)
temp3 = temp3[temp3$Freq != 0,]
temp3 = cbind(temp3, period = rep('Pilot', period = length(temp3$Freq)))
#Add baseline data
king_daily = rbind(Baseline, temp3)
king_daily$period = factor(king_daily$period, levels=c('Pilot', 'Baseline'))
king_daily$period_name = factor(king_daily$period_name, levels=c('Morning
(08:00-12:00)', 'Afternoon (12:00-17:00)', 'Evening (17:00-22:00)'))

#plot
ggplot(king_daily, aes(x = period, y = Freq, fill = classification)) +
  geom_bar(stat = 'identity', position = 'dodge') +
facet_wrap(vars(period_name), nrow = 3) + coord_flip() + labs(x = 'Time
Period', y = 'Average Frequency Count', title = "Breakdown of Average Daily
Frequency Counts", subtitle = "King St") + guides(fill = guide_legend(title =
"Classification")) + theme(axis.title.y = element_text(margin = margin(t = 0,
r = 10, b = 0, l = 0)), axis.text.x = element_text(size = 10, margin =
margin(t = 0, r = 0, b = 10, l = 0))) + scale_y_continuous(labels =
scales::comma)
```

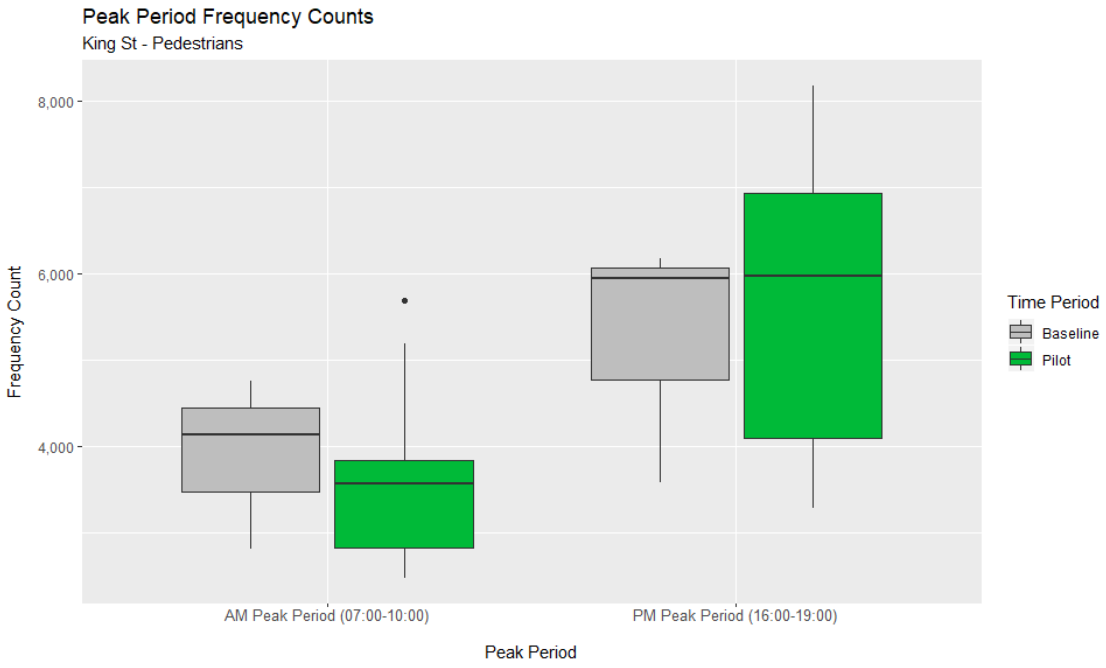



```r
#King St Summary - Peak Periods
count = read.csv(file ="count_orig.csv")
count_king = filter(count, intersection_name %in% c("Bay St / King St W",
"Spadina Ave / King St W", "Bathurst St / King St W"))


#Filter by Peak Period
count_monthly = filter(count_king, period_name %in% c("AM Peak Period (07:00-
10:00)", "PM Peak Period (16:00-19:00)"))
```

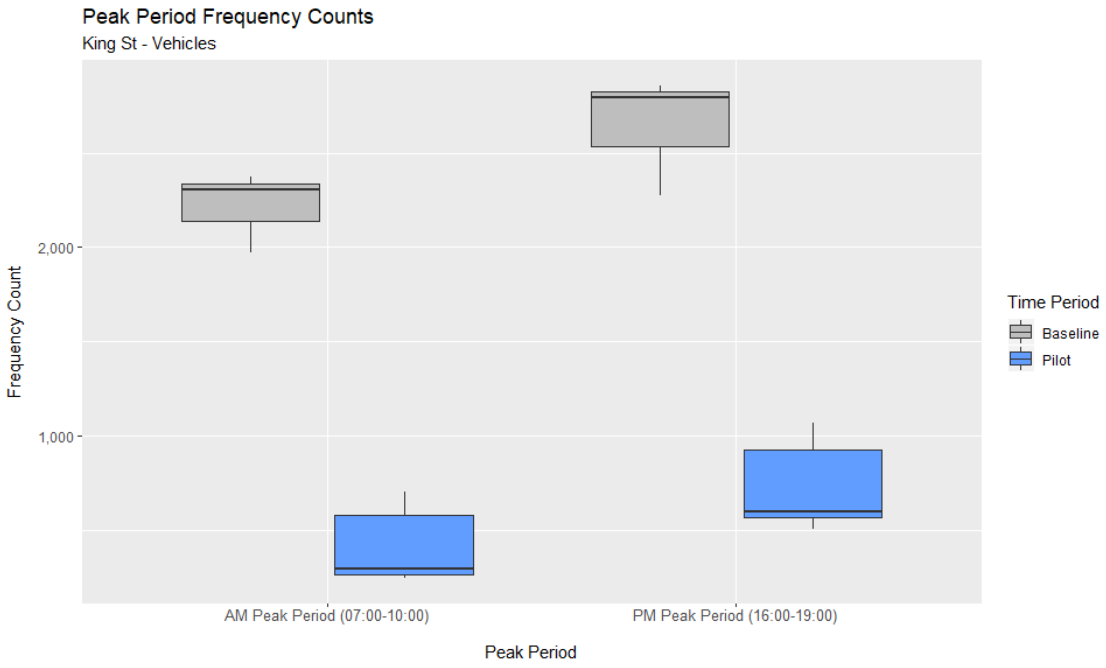
```r
#Daily count for each classification at each intersection
temp1 = xtabs(formula =
volume~classification+aggregation_period+intersection_name+period_name, data
= count_monthly)
temp1 = as.data.frame(temp1)
temp1 = temp1[temp1$Freq != 0,]

#Save Baseline data
Baseline = temp1[temp1$aggregation_period == 'Baseline',]
Baseline = cbind(Baseline, period = rep('Baseline', length(Baseline$Freq)))

#save pilot data
temp2 = temp1[temp1$aggregation_period != 'Baseline',]
temp2 = cbind(temp2, period = rep('Pilot', length(temp2$Freq)))

#Combine baseline and pilot data
king_daily = rbind(Baseline, temp2)

#Cyclists
comparison_cycl = king_daily[king_daily$classification == 'Cyclists',]

#Classification Frequency Count during Peak Period
ggplot(comparison_cycl, aes(x = period_name, y = Freq, fill = period)) +
geom_boxplot() + labs(x = 'Peak Period', y = 'Frequency Count', title = "Peak
Period Frequency Counts", subtitle = "King St - Cyclists") +
theme(axis.title.y = element_text(margin = margin(t = 0, r = 10, b = 0, l =
0)), axis.text.x = element_text(size = 10, margin = margin(t = 0, r = 0, b =
10, l = 0))) + scale_y_continuous(labels = scales::comma) +
scale_fill_manual(values = c('grey', '#F8766D')) + guides(fill =
guide_legend(title = "Time Period"))
```




```
#Pedestrians
comparison_ped = king_daily[king_daily$classification == 'Pedestrians',]

#Classification Frequency Count during Peak Period
ggplot(comparison_ped, aes(x = period_name, y = Freq, fill = period)) +
geom_boxplot() + labs(x = 'Peak Period', y = 'Frequency Count', title = "Peak
Period Frequency Counts", subtitle = "King St - Pedestrians") +
theme(axis.title.y = element_text(margin = margin(t = 0, r = 10, b = 0, l =
0)), axis.text.x = element_text(size = 10, margin = margin(t = 0, r = 0, b =
10, l = 0))) + scale_y_continuous(labels = scales::comma) +
scale_fill_manual(values = c('grey', '#00BA38')) + guides(fill =
guide_legend(title = "Time Period"))
```

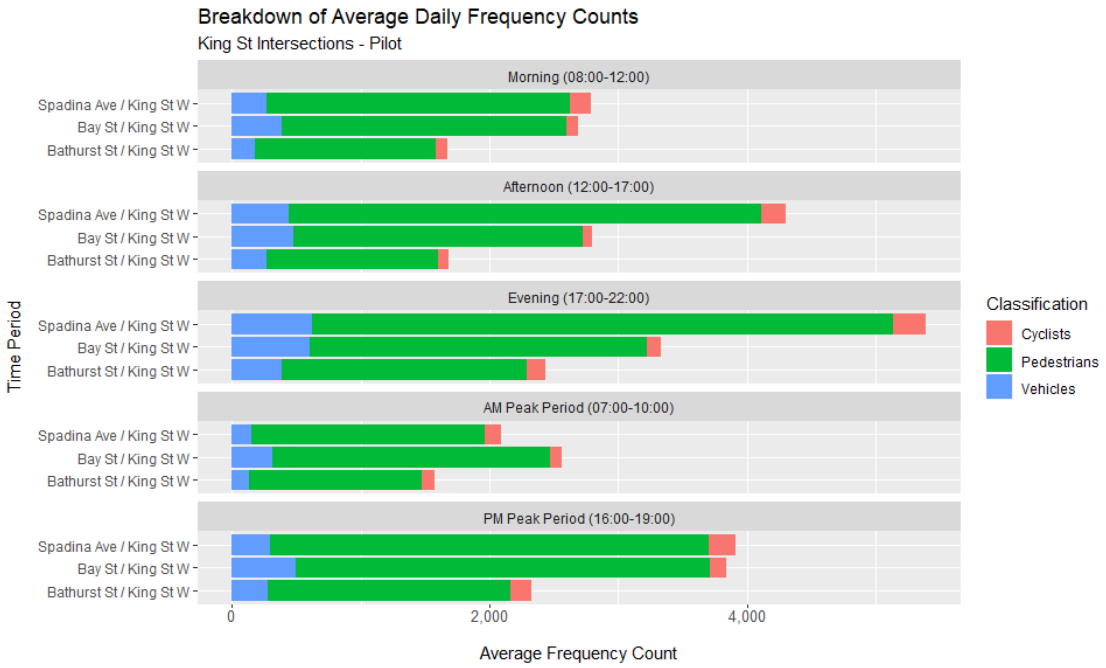



```r
#Vehicles
comparison_veh = king_daily[king_daily$classification == 'Vehicles',]

#Classification Frequency Count during Peak Period
ggplot(comparison_veh , aes(x = period_name, y = Freq, fill = period)) +
geom_boxplot() + labs(x = 'Peak Period', y = 'Frequency Count', title = "Peak
Period Frequency Counts", subtitle = "King St - Vehicles") +
theme(axis.title.y = element_text(margin = margin(t = 0, r = 10, b = 0, l =
0)), axis.text.x = element_text(size = 10, margin = margin(t = 0, r = 0, b =
10, l = 0))) + scale_y_continuous(labels = scales::comma) +
scale_fill_manual(values = c('grey', '#619CFF')) + guides(fill =
guide_legend(title = "Time Period"))
```



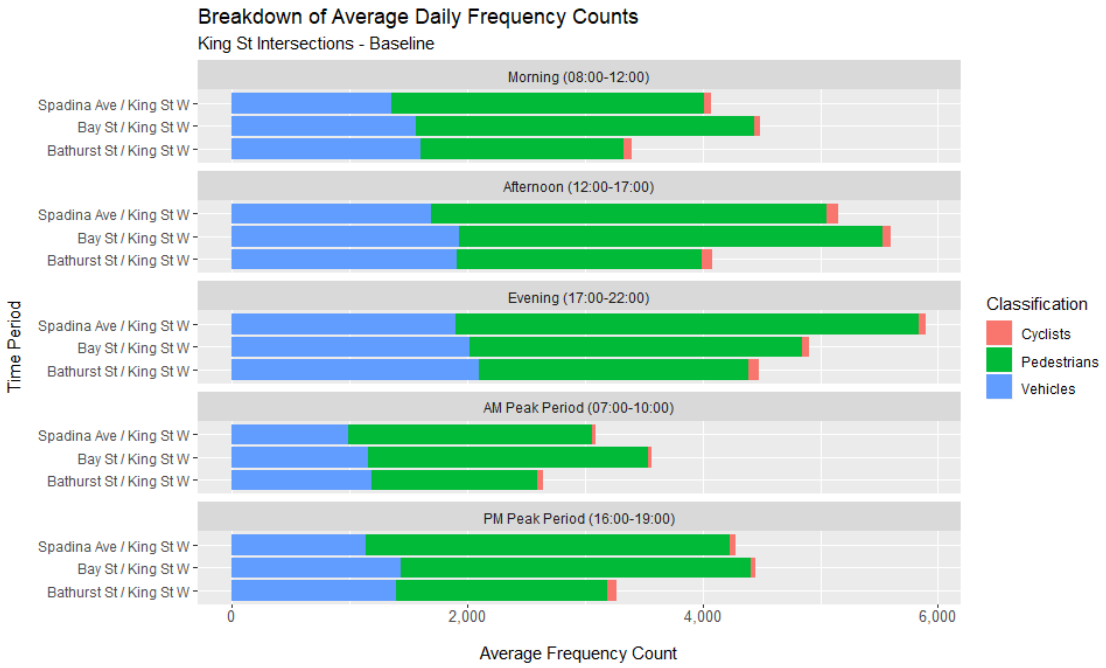

```r
#King St Intersection Breakdown
count = read.csv(file ="count_orig.csv")
count_king = filter(count, intersection_name %in% c("Bay St / King St W",
"Spadina Ave / King St W", "Bathurst St / King St W"))
count_monthly = filter(count_king, period_name %in% c("Morning (08:00-
12:00)", "Afternoon (12:00-17:00)", "Evening (17:00-22:00)","AM Peak Period
(07:00-10:00)", "PM Peak Period (16:00-19:00)"))

#Daily count for each classification at each intersection
temp1 = xtabs(formula =
volume~classification+aggregation_period+intersection_name+period_name,
aggregate(volume~classification+aggregation_period+intersection_name+period_n
ame, count_monthly, mean))
temp1 = as.data.frame(temp1)
temp1 = temp1[temp1$Freq != 0,]

#Save Baseline data
king_intersections_baseline = temp1[temp1$aggregation_period == 'Baseline',]
king_intersections_baseline$period_name =
factor(king_intersections_baseline$period_name, levels=c('Morning (08:00-
12:00)', 'Afternoon (12:00-17:00)', 'Evening (17:00-22:00)', "AM Peak Period
(07:00-10:00)", "PM Peak Period (16:00-19:00)"))

#Isolate Pilot data
king_intersections = temp1[temp1$aggregation_period != 'Baseline',]
king_intersections = xtabs(formula =
Freq~classification+intersection_name+period_name,
aggregate(Freq~classification+intersection_name+period_name,
king_intersections, mean))
```

```
king_intersections = as.data.frame(king_intersections)
king_intersections = king_intersections[king_intersections$Freq != 0,]
king_intersections = cbind(king_intersections, aggregation_period =
rep('Pilot', length(king_intersections$Freq)))
king_intersections$period_name = factor(king_intersections$period_name,
levels=c('Morning (08:00-12:00)', 'Afternoon (12:00-17:00)', 'Evening (17:00-
22:00)', "AM Peak Period (07:00-10:00)", "PM Peak Period (16:00-19:00)"))

#Pilot Data
ggplot(king_intersections, aes(x = intersection_name, y = Freq, fill =
classification)) +
  geom_bar(stat = 'identity', position = 'stack') +
facet_wrap(vars(period_name), nrow = 5) + coord_flip() + labs(x = 'Time
Period', y = 'Average Frequency Count', title = "Breakdown of Average Daily
Frequency Counts", subtitle = "King St Intersections - Pilot") + guides(fill
= guide_legend(title = "Classification")) + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
element_text(size = 10, margin = margin(t = 0, r = 0, b = 10, l = 0))) +
scale_y_continuous(labels = scales::comma)
```



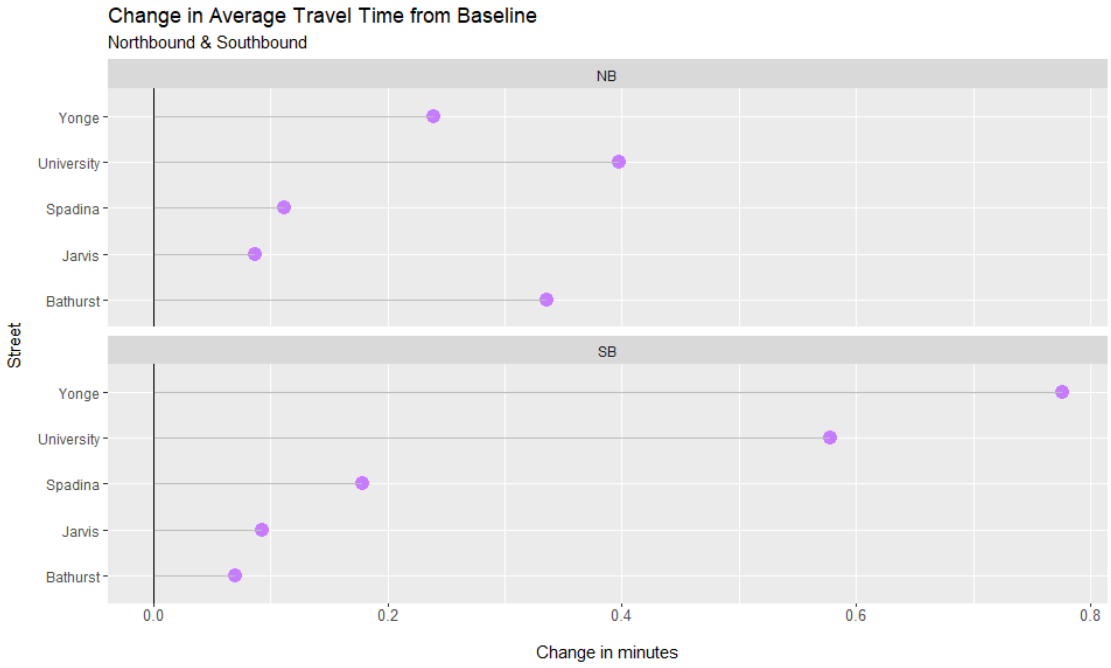

```
#Baseline Data
ggplot(king_intersections_baseline, aes(x = intersection_name, y = Freq, fill
= classification)) +
  geom_bar(stat = 'identity', position = 'stack') +
facet_wrap(vars(period_name), nrow = 5) + coord_flip() + labs(x = 'Time
Period', y = 'Average Frequency Count', title = "Breakdown of Average Daily
Frequency Counts", subtitle = "King St Intersections - Baseline") +
guides(fill = guide_legend(title = "Classification")) + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
```

```
element_text(size = 10, margin = margin(t = 0, r = 0, b = 10, l = 0))) +
scale_y_continuous(labels = scales::comma)
```

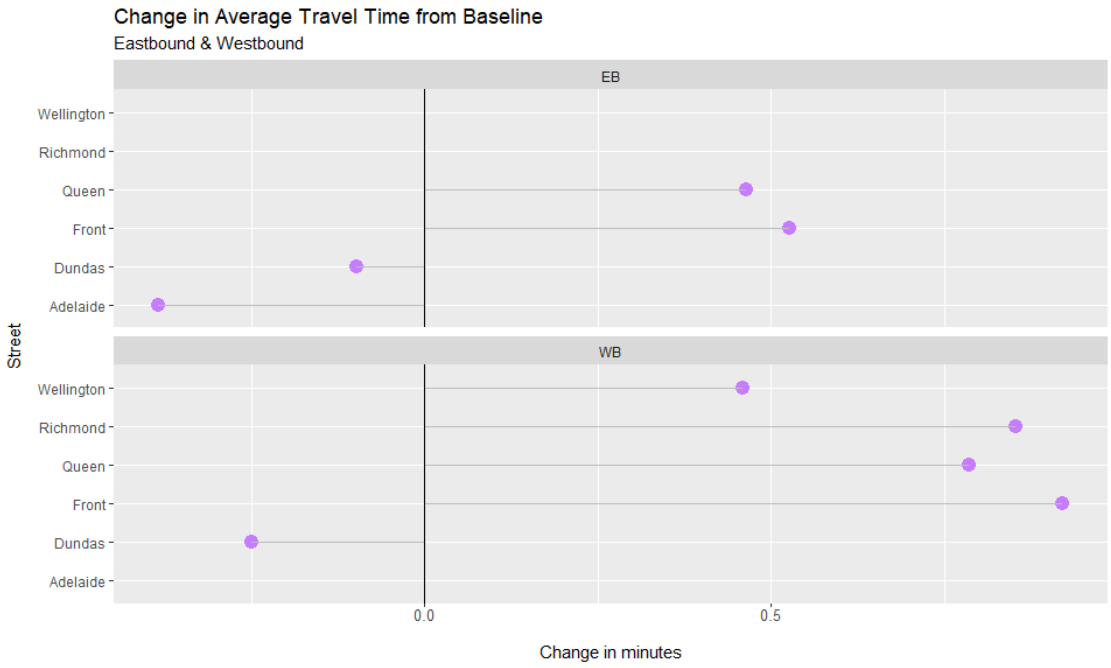



```
time = read.csv("time.csv")

#Find Baseline average
temp1 = xtabs(formula = baseline_travel_time ~ street+direction,
aggregate(baseline_travel_time~street+direction, time, mean))
temp1 = as.data.frame(temp1)
temp1 = temp1[temp1$Freq != 0,]
colnames(temp1)[colnames(temp1) == "Freq"] <- "baseline_travel"

#Find pilot average
temp2 = xtabs(formula = average_travel_time ~ street+direction,
aggregate(average_travel_time~street+direction, time, mean))
temp2 = as.data.frame(temp2)
temp2 = temp2[temp2$Freq != 0,]
colnames(temp2)[colnames(temp2) == "Freq"] <- "average_travel"


#combine baseline and pilot averages into data
time <- join(temp1, temp2)

## Joining by: street, direction

#caculate change
diff = time$average_travel - time$baseline_travel

#add column for change
time = cbind(time, Difference = diff)
```

```
#Filter by EB WB & NB SB
time_NS = filter(time, direction %in% c("NB", "SB"))
time_EW = filter(time, direction %in% c("EB", "WB"))

#Lollipop Chart
ggplot(time_NS) +
  geom_point( aes(x = street, y= Difference), color = "#C77CFF", size=4) +
geom_segment(aes(x = street, xend = street, y = 0, yend= Difference), color =
"grey") + labs(title = "Change in Average Travel Time from Baseline",
subtitle = "Northbound & Southbound", y = "Change in minutes", x = "Street")
+ facet_wrap(~direction, ncol = 1) + coord_flip() + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
element_text(size = 10, margin = margin(t = 0, r = 0, b = 10, l = 0))) +
geom_hline(yintercept=0, color = "black")
```




```
ggplot(time_EW) +
  geom_point( aes(x = street, y= Difference), color = "#C77CFF", size=4) +
geom_segment(aes(x = street, xend = street, y = 0, yend= Difference), color =
"grey") + labs(title = "Change in Average Travel Time from Baseline",
subtitle = "Eastbound & Westbound", y = "Change in minutes", x = "Street") +
facet_wrap(~direction, ncol = 1) + coord_flip() + theme(axis.title.y =
element_text(margin = margin(t = 0, r = 10, b = 0, l = 0)), axis.text.x =
element_text(size = 10, margin = margin(t = 0, r = 0, b = 10, l = 0))) +
geom_hline(yintercept=0, color = "black")
```


Change in Average Travel Time from Baseline
Eastbound & Westbound
EB
WB
Wellington
Richmond
Queen
Front
Dundas
Adelaide
Street
0.0
0.5
Change in minutes