STAT 437 Homework 1

Christopher Mims

1/20/2022

```
library(knitr)
library(tidyr)
library(ggplot2)
library(dplyr)
library(nycflights13)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Problem 1

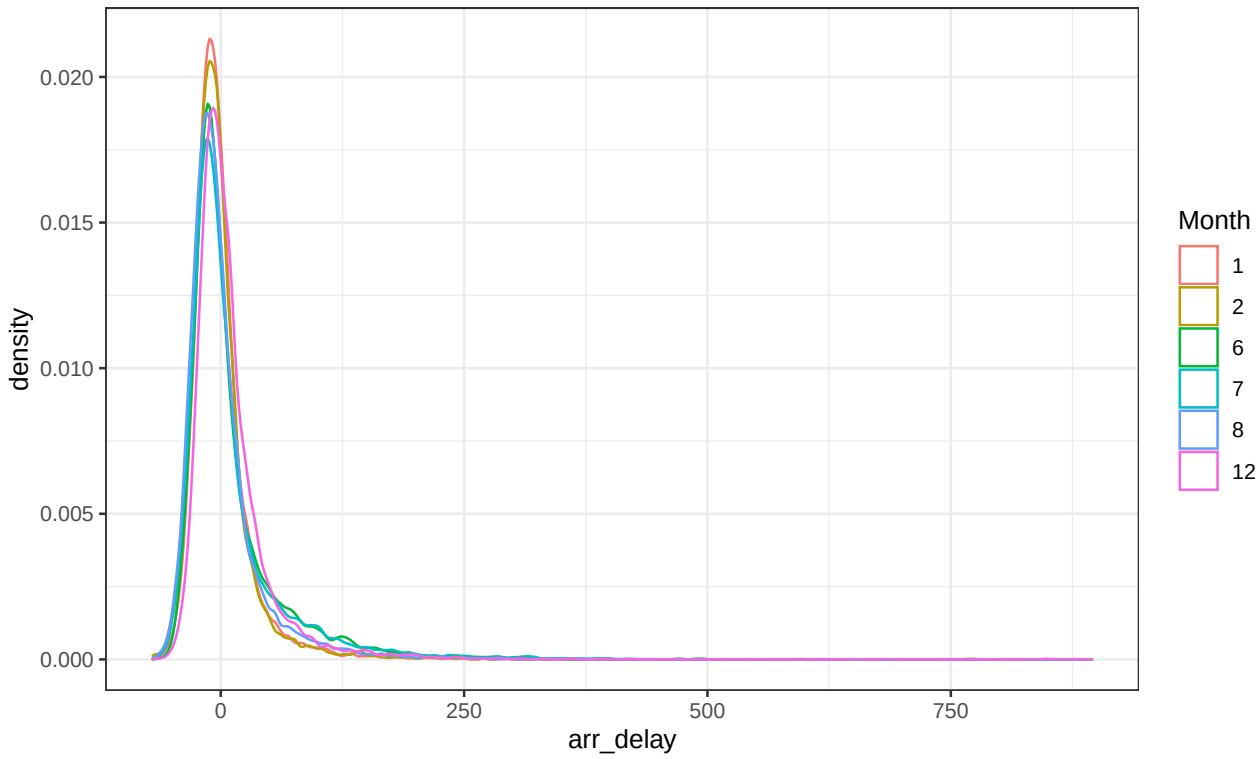
In order to make things easy, we will need to create a subset of the data where we filter out all but the information we will need to build our plots. We will use the `dplyr` package to obtain our subset. Once this has been created, then we can calculate the average `arr_delay` for each of the months.

```
p1 <- select(flights, month, arr_delay, carrier, distance) %>%
    filter(carrier %in% c("UA", "AA", "DL"), month %in% c(1:2,
        6:8, 12), distance > 700)
```

Part A

Below I will create a single plot containing the density plots for `arr_delay` for each of the 6 months, and using `month` to determine the color. In order to do this, I must first convert `month` to a factor.

```
Month <- as.factor(p1$month)
p1a <- ggplot(p1, aes(x = arr_delay, color = Month)) + geom_density() +
    theme_bw()
p1a
```

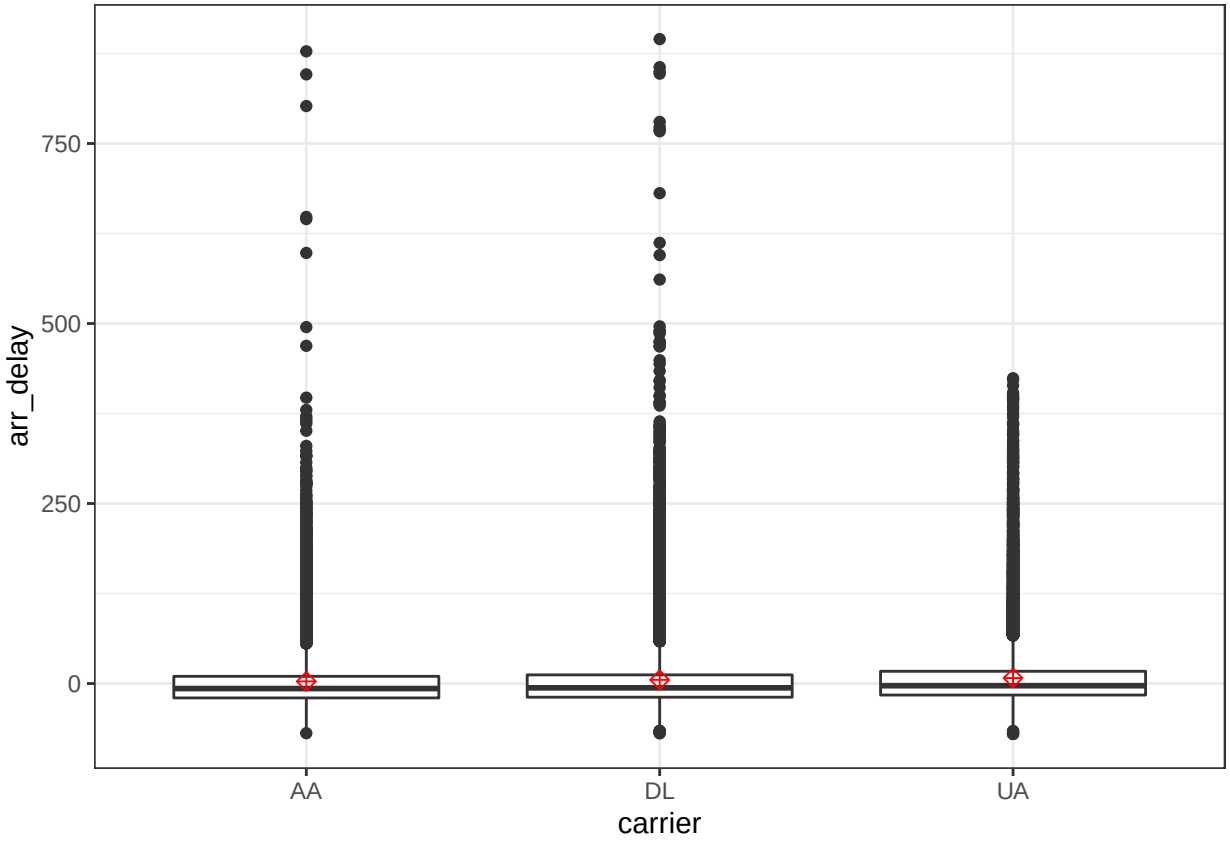


From this plot, it looks as though the average `arr_delay` across the 6 months are very similar and are near a value of zero.

Part B

Below I will create a single plot containing a box plot for `arr_delay` for each `carrier` (there are 3).

```
p1b <- ggplot(p1, aes(x = carrier, y = arr_delay)) + geom_boxplot() +
    stat_summary(fun = mean, geom = "point", shape = 9, size = 2,
        color = "red") + theme_bw()
p1b
```

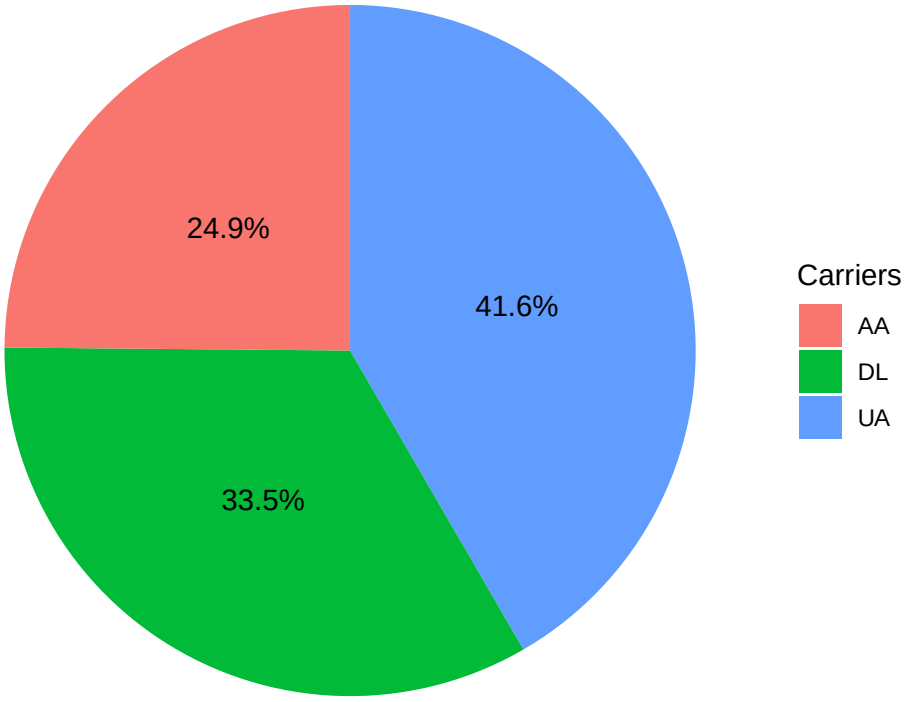


And in this plot we also see that the average `arr_delay` for each of the three carriers are very similar and that they lie near a value of zero. One other observation we can see from this plot is that 'UA' has fewer extreme outliers than 'AA' and 'DL'. To note, this plot is very compressed and any assumptions about the accuracy of the averages would not be precise.

Part C

Below I will create a pie chart showing the proportions of the chart representing each carriers share of the total number of observations. I will superimpose the values onto each section. Each section will also be distinguished by color.

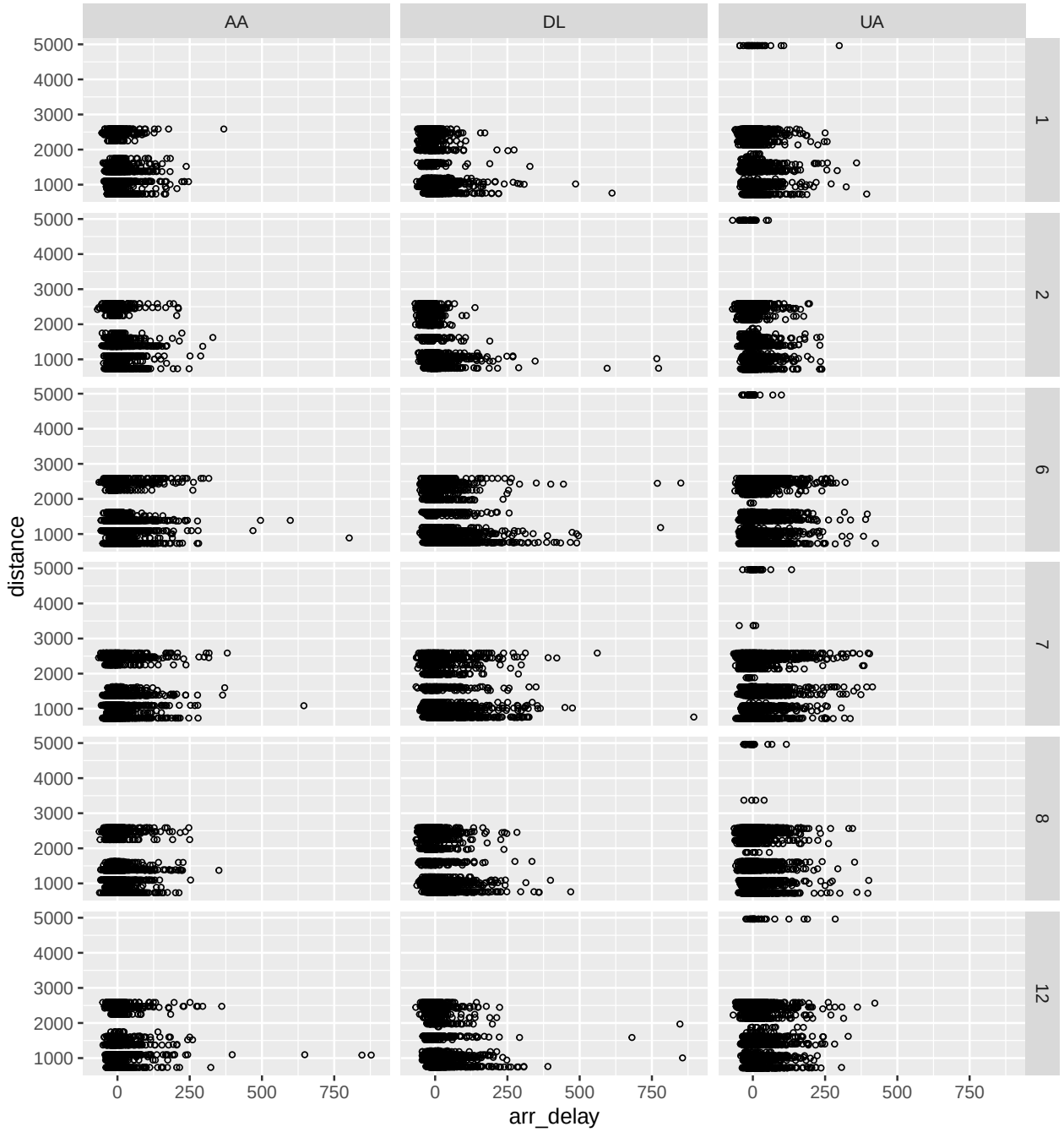
```
# First perform the calculations
p1c <- p1 %>%
    group_by(carrier) %>%
    count() %>%
    ungroup() %>%
    mutate(percentage = n/sum(n)) %>%
    arrange(desc(carrier))
p1c$labels <- scales::percent(p1c$percentage)

# Create the pie chart
Carriers <- as.factor(p1c$carrier)
pie <- ggplot(p1c) + geom_bar(aes(x = "", y = percentage, fill = Carriers),
    stat = "identity", width = 1) + coord_polar("y", start = 0) +
    theme_void() + geom_text(aes(x = 1, y = cumsum(percentage) -
    percentage/2, label = labels))
pie
```



Part D

Below I will plot `arr_delay` against `distance` in a grid using `facet_grid` designated by `month` and `carrier`.

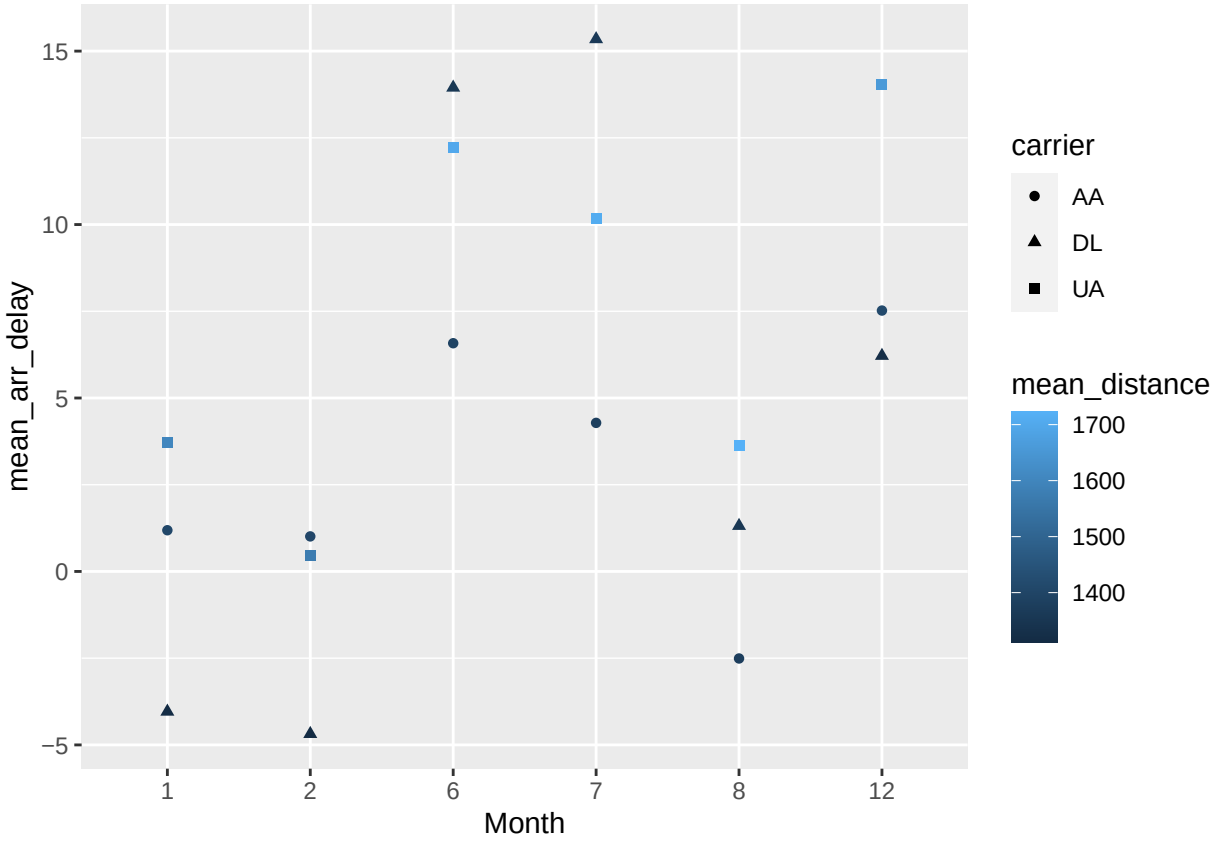
```
# Build a base layer
p1d <- ggplot(p1, aes(x = arr_delay, y = distance)) + geom_point(size = 1,
    shape = 1) + facet_grid(month ~ carrier)
p1d
```



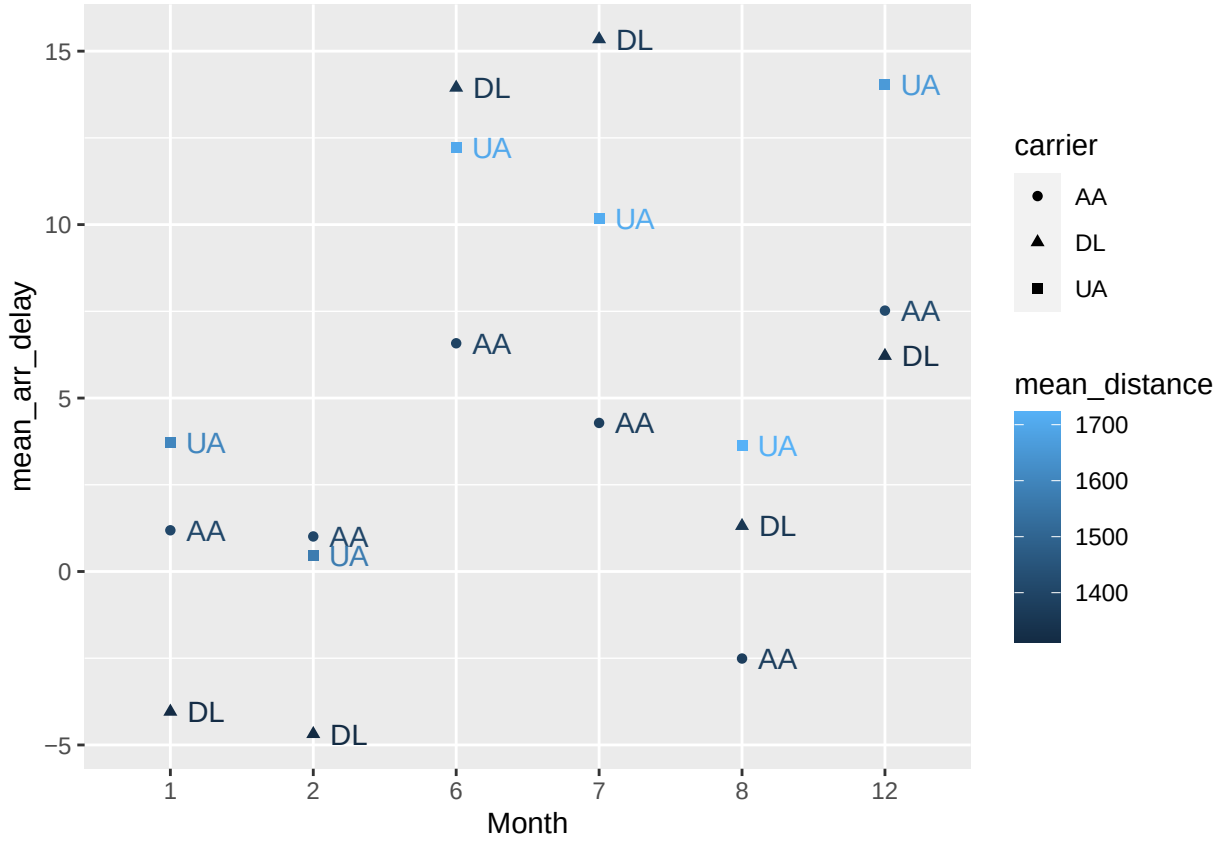
Part E

Below I will compute the sample average of `arr_delay`, as well as, the sample average of `distance`. These will be named `mean_arr_delay` and `mean_distance` respectively. Once these calculations are completed there will be two plots made. The first will plot `month` against `mean_arr_delay` with the shape based off of `carrier` and color based off of `mean_distance`. And the second plot is the same but adds on the annotation of the `carrier` label (which I colored the same as the point).

```
# Create data frame with calculated mean values
tmp <- na.omit(p1)
p1e <- tmp %>%
    group_by(month, carrier) %>%
    summarise(mean_arr_delay = mean(arr_delay), mean_distance = mean(distance)) %>%
    as.data.frame()

# Create first plot
Month <- as.factor(p1e$month)
plt1 <- ggplot(p1e, aes(x = Month, y = mean_arr_delay)) + geom_point(aes(shape = carrier,
    color = mean_distance))
plt1
```



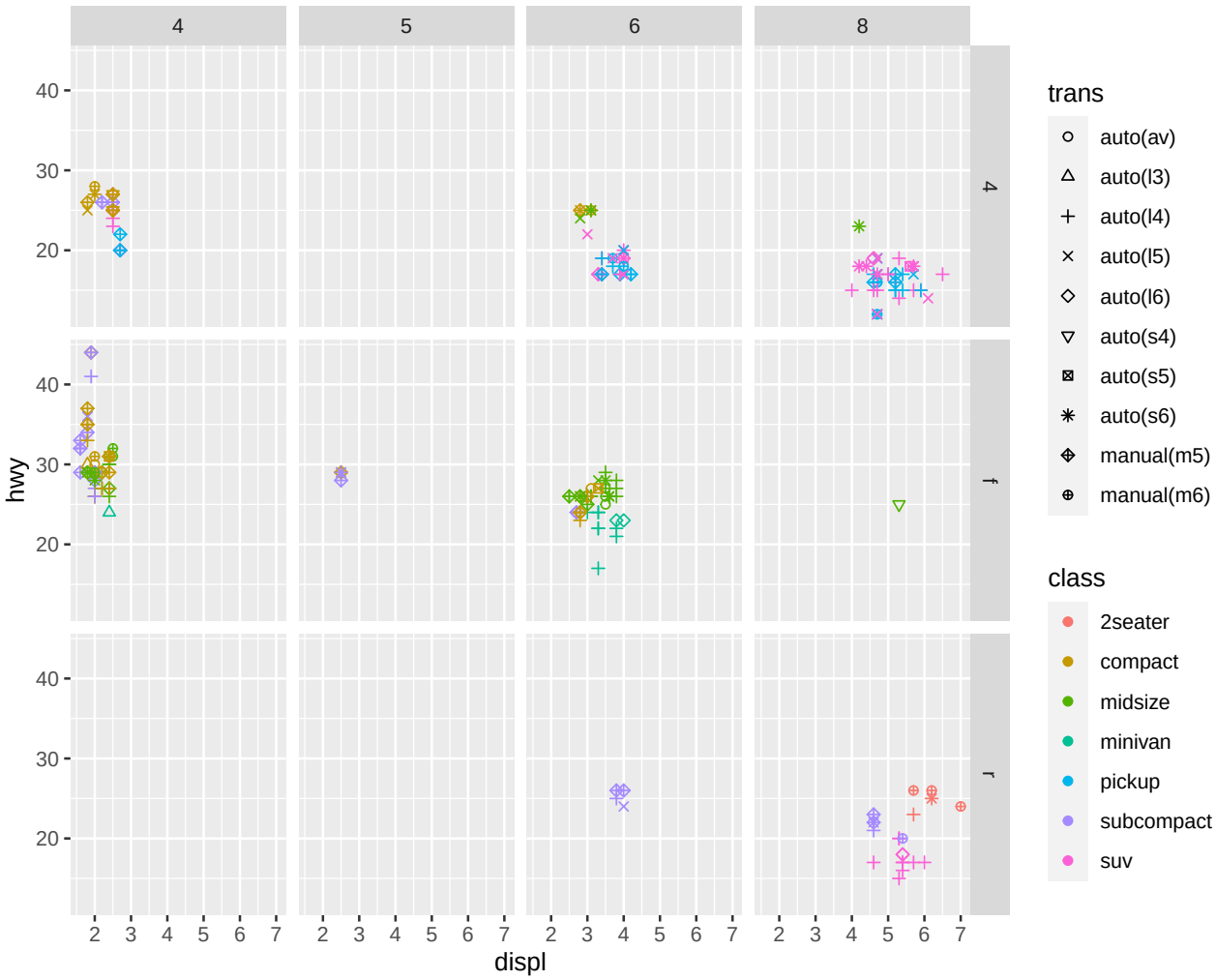
```
plt2 <- ggplot(p1e, aes(x = Month, y = mean_arr_delay)) + geom_point(aes(shape = carrier,
    color = mean_distance)) + geom_text(aes(label = carrier,
    color = mean_distance), nudge_x = 0.25)
plt2
```



Problem 2

In this problem I will use the `mpg` dataset within `ggplot2` to create a plot that plots `displ` against `hwy` and is faceted by `drv` and `cyl`. I will use color based off of `class` and shape based off of `trans`. First I need to select a subset of the data (not entirely mandatory but simplifies the process) and ensure `drv`, `cyl`, `class`, and `trans` have been converted into factors. This will allow for the plot to populate correctly.

```
# Create subset
p2 <- select(mpg, displ, hwy, drv, cyl, class, trans)

# Convert `drv`, `cyl`, `class`, and `trans` to factors
p2$drv <- as.factor(p2$drv)
p2$cyl <- as.factor(p2$cyl)
p2$class <- as.factor(p2$class)
p2$trans <- as.factor(p2$trans)

# Create plot
p2plt <- ggplot(p2, aes(x = displ, y = hwy)) + geom_point(aes(shape = trans,
    color = class)) + scale_shape_manual(values = 1:length(unique(p2$trans))) +
    facet_grid(drv ~ cyl)
p2plt
```




```
## Loading required package: usethis
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] devtools_2.4.3     usethis_2.1.5      nycflights13_1.0.2 dplyr_1.0.7
## [5] ggplot2_3.3.5      tidyr_1.1.4        knitr_1.34
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.26         remotes_2.4.2     purrr_0.3.4
```

```
##  [5] testthat_3.0.4    colorspace_2.0-2  vctrs_0.3.8       generics_0.1.0
##  [9] htmltools_0.5.2   yaml_2.2.1        utf8_1.2.2        rlang_0.4.11
## [13] pkgbuild_1.3.0    pillar_1.6.2      glue_1.4.2        withr_2.4.2
## [17] DBI_1.1.1         sessioninfo_1.2.2 lifecycle_1.0.0   stringr_1.4.0
## [21] munsell_0.5.0     gtable_0.3.0      evaluate_0.14     memoise_2.0.0
## [25] labeling_0.4.2    callr_3.7.0       fastmap_1.1.0     ps_1.6.0
## [29] fansi_0.5.0       highr_0.9         scales_1.1.1      formatR_1.11
## [33] cachem_1.0.6      desc_1.4.0        pkgload_1.2.2     farver_2.1.0
## [37] fs_1.5.0          digest_0.6.27     stringi_1.7.4     processx_3.5.2
## [41] rprojroot_2.0.2   grid_4.1.1        cli_3.1.0         tools_4.1.1
## [45] magrittr_2.0.1    tibble_3.1.4      crayon_1.4.1      pkgconfig_2.0.3
## [49] ellipsis_0.3.2    prettyunits_1.1.1 assertthat_0.2.1  rmarkdown_2.11
## [53] rstudioapi_0.13   R6_2.5.1          compiler_4.1.1
```