Part 1

Load the libraries

```
library(readr)
library(ggplot2)
library(dplyr)
library(jhur)
```

Read Bike Lanes Dataset using `read_bike()` function from `jhur` package. Assign it to `bike` variable.

Then, use the provided code to compute a data frame `bike_agg` with aggregate summary of bike lanes: average length of lanes (`lane_avg_length`) for each year (`dateInstalled`).

```
bike <- read_bike()
```

```
## Rows: 1631 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): subType, name, block, type, project, route
## dbl (3): numLanes, length, dateInstalled
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
bike_agg <- bike %>%
  # filter data to keep only these observations for which year is non-0
  filter(dateInstalled != 0) %>%
  group_by(dateInstalled) %>%
  summarise(lane_avg_length = mean(length))

bike_agg
```

```
## # A tibble: 8 x 2
##   dateInstalled lane_avg_length
##           <dbl>           <dbl>
## 1          2006           1469.
## 2          2007            310.
## 3          2008            249.
## 4          2009            407.
## 5          2010            246.
## 6          2011            233.
## 7          2012            271.
## 8          2013            290.
```

1.1

Use `ggplot2` package make plot of average length of lanes (`lane_avg_length`; y-axis) for each year (`dateInstalled`; x-axis). You can use lines layer (`+ geom_line()`) or points layer (`+ geom_point()`), or both!

Assign the plot to variable `my_plot`. Type `my_plot` in the console to have it displayed.

```
# General format
ggplot(???, aes(x = ???, y = ???)) +
  ??? +
  ???
```

1.2

"Update" your plot by adding a title and changing the x and y axis titles.

1.3

Use the `scale_x_continuous()` function to plot the x axis with the following breaks `c(2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013)`.

```
# General format
my_plot<- my_plot +
  scale_x_continuous(?????)
```

1.4

Observe several different versions of the plot by displaying `my_plot` while adding a different "theme" to it.

```
# General format
my_plot + theme_bw()
```

Practice on Your Own!

P.1

Create a boxplot (with the `geom_boxplot()` function) using the `Orange` data, where `Tree` is plotted on the x axis and `circumference` is plotted on the y axis.

Notice how the trees are ordered. We will learn more about this soon!

Part 2

2.1

Use the provided code to compute a data frame `bike_agg_2` with aggregate summary of bike lanes: number of lanes (`lane_count`) – separately for each year (`dateInstalled`) and for each lane type.

```
bike_agg_2 <- bike %>%
  filter(dateInstalled != 0) %>%
  group_by(dateInstalled, type) %>%
  summarise(lane_count = n())
```

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

```
bike_agg_2
```

```
## # A tibble: 22 x 3
## # Groups:   dateInstalled [8]
##    dateInstalled type          lane_count
##            <dbl> <chr>              <int>
##  1          2006 BIKE LANE              2
##  2          2007 BIKE LANE            127
##  3          2007 SHARROW               95
##  4          2007 SIGNED ROUTE         146
##  5          2008 BIKE LANE             55
##  6          2008 SHARROW              148
##  7          2008 SIDEPATH               3
##  8          2009 BIKE LANE             46
##  9          2009 SHARED BUS BIKE         30
## 10          2009 SHARROW                 10
## # i 12 more rows
```

2.2

Use `ggplot2` package to make a plot showing trajectories of number of lanes (`lane_count`; y-axis) over year (`dateInstalled`; x-axis), where each bike line type has a different color (hint: use `color = type` in mapping).

```
# General format
ggplot(???, aes(
  x = ???,
  y = ???,
  color = ???
)) +
  geom_line() +
  geom_point()
```

2.3

Redo the above plot by adding a faceting (`+ facet_wrap( ~ type, ncol = 3)`) to have data for each bike line type in a separate plot panel.

(You may see `geom_path: Each group consists of only one observation. Do you need to adjust the group aesthetic?` warning as some bike lane types will have only 1 point plotted while trying to plot a line). Assign the new plot as an object called `facet_plot`.

2.4

Observe what happens when you remove either `geom_line()` OR `geom_point()` from one of your plots above.

Practice on Your Own!

P.2

Modify `facet_plot` to remove the legend (hint use `theme()` and the `legend.position` argument) and change the names of the axis titles to be "Number of bike lanes" for the y axis and "Date bike lane was installed" for the x axis.

P.3

Modify `facet_plot` one more time with a fun theme! Look into the ThemePark package It has lots of fun themes! Try one out! Remember you will need to install it using `remotes::install_github("MatthewBJane/ThemePark")`and load in the library.