Yonsei Internship Sample 2021

Sung Bum (Simon) Ahn

3/10/2021

```
# Load the necessary packages
library(tidyverse)
library(pdxTrees)
library(infer)
library(broom)
library(maps)
library(mapdata)
```

We are going to use data from the `pdxTrees` package. In particular, we will use the dataset called `four_parks` that I created below.

Make sure to run the following R chunk.

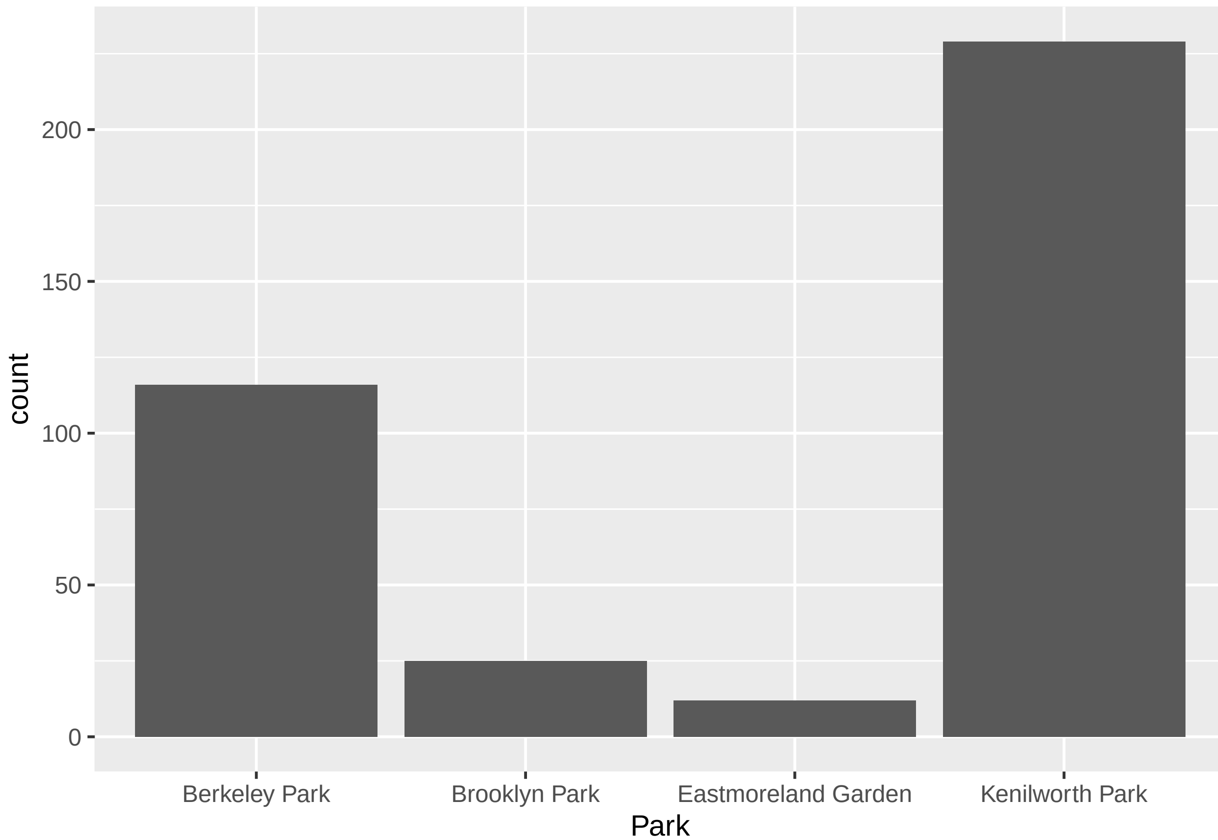
```
# Grab trees near Reed
four_parks <- get_pdxTrees_parks(park =
                                  c("Brooklyn Park",
                                    "Kenilworth Park",
                                    "Eastmoreland Garden",
                                    "Berkeley Park"))

#Remove trees with no functional type
four_parks <- drop_na(four_parks, Functional_Type)
```

A. Low difficulty questions

   a. Create a bar plot of `park`.

Answer.

```
# Bar plot
ggplot(data = four_parks, mapping = aes(x = Park)) +
  geom_bar()
```



b. Using the four parks dataset, select Genus and Common_Name to create a new dataset called "tree_name". Print "tree_name" to get a grade.

```
tree_name <- four_parks %>%
  select(c(Genus, Common_Name))
tree_name
```

```
## # A tibble: 382 x 2
##    Genus       Common_Name
##    <chr>       <chr>
##  1 Acer        Norway Maple
##  2 Acer        Norway Maple
##  3 Pseudotsuga Douglas-Fir
##  4 Pseudotsuga Douglas-Fir
##  5 Acer        Norway Maple
##  6 Acer        Norway Maple
##  7 Acer        Norway Maple
##  8 Acer        Norway Maple
##  9 Acer        Norway Maple
## 10 Acer        Norway Maple
## # ... with 372 more rows
```

c. Tally the condition of trees in these four parks. Tally them accordingly and print only Park, Condition and n on a table. Print "tree_name" to get a grade.

```
four_parks %>%
  group_by(Park, Condition) %>%
  tally()
```

```
## # A tibble: 10 x 3
## # Groups:   Park [4]
##    Park                Condition     n
##    <chr>               <chr>     <int>
##  1 Berkeley Park       Fair         93
##  2 Berkeley Park       Good         10
##  3 Berkeley Park       Poor         13
##  4 Brooklyn Park       Fair         15
##  5 Brooklyn Park       Poor         10
##  6 Eastmoreland Garden Fair         11
##  7 Eastmoreland Garden Good          1
##  8 Kenilworth Park     Fair        218
##  9 Kenilworth Park     Good          7
## 10 Kenilworth Park     Poor          4
```

```
four_parks
```

```
## # A tibble: 382 x 34
##    Longitude Latitude UserID Genus    Family     DBH Inventory_Date      Species
##        <dbl>    <dbl> <chr>  <chr>    <chr>    <dbl> <dttm>              <chr>
##  1     -123.     45.5 21     Acer     Sapinda~  12.2 2017-05-19 00:00:00 ACPL
##  2     -123.     45.5 22     Acer     Sapinda~  29   2017-05-19 00:00:00 ACPL
##  3     -123.     45.5 23     Pseudot~ Pinaceae  54.7 2017-05-19 00:00:00 PSME
##  4     -123.     45.5 24     Pseudot~ Pinaceae  39.6 2017-05-19 00:00:00 PSME
##  5     -123.     45.5 25     Acer     Sapinda~  13.5 2017-05-19 00:00:00 ACPL
##  6     -123.     45.5 26     Acer     Sapinda~  28   2017-05-19 00:00:00 ACPL
##  7     -123.     45.5 27     Acer     Sapinda~  28.1 2017-05-19 00:00:00 ACPL
##  8     -123.     45.5 28     Acer     Sapinda~  29   2017-05-19 00:00:00 ACPL
##  9     -123.     45.5 29     Acer     Sapinda~  27.7 2017-05-19 00:00:00 ACPL
## 10     -123.     45.5 30     Acer     Sapinda~  24.6 2017-05-19 00:00:00 ACPL
## # ... with 372 more rows, and 26 more variables: Common_Name <chr>,
## #   Condition <chr>, Tree_Height <dbl>, Crown_Width_NS <dbl>,
## #   Crown_Width_EW <dbl>, Crown_Base_Height <dbl>, Collected_By <chr>,
## #   Park <chr>, Scientific_Name <chr>, Functional_Type <chr>,
## #   Mature_Size <fct>, Native <chr>, Edible <chr>, Nuisance <chr>,
## #   Structural_Value <dbl>, Carbon_Storage_lb <dbl>,
## #   Carbon_Storage_value <dbl>, Carbon_Sequestration_lb <dbl>,
## #   Carbon_Sequestration_value <dbl>, Stormwater_ft <dbl>,
## #   Stormwater_value <dbl>, Pollution_Removal_value <dbl>,
## #   Pollution_Removal_oz <dbl>, Total_Annual_Services <dbl>, Origin <chr>,
## #   Species_Factoid <chr>
```

B. Medium difficulty questions

a. Find the tallest tree within the park, and give its tree height, common name, date of data collected (Inventory Date), and park name.

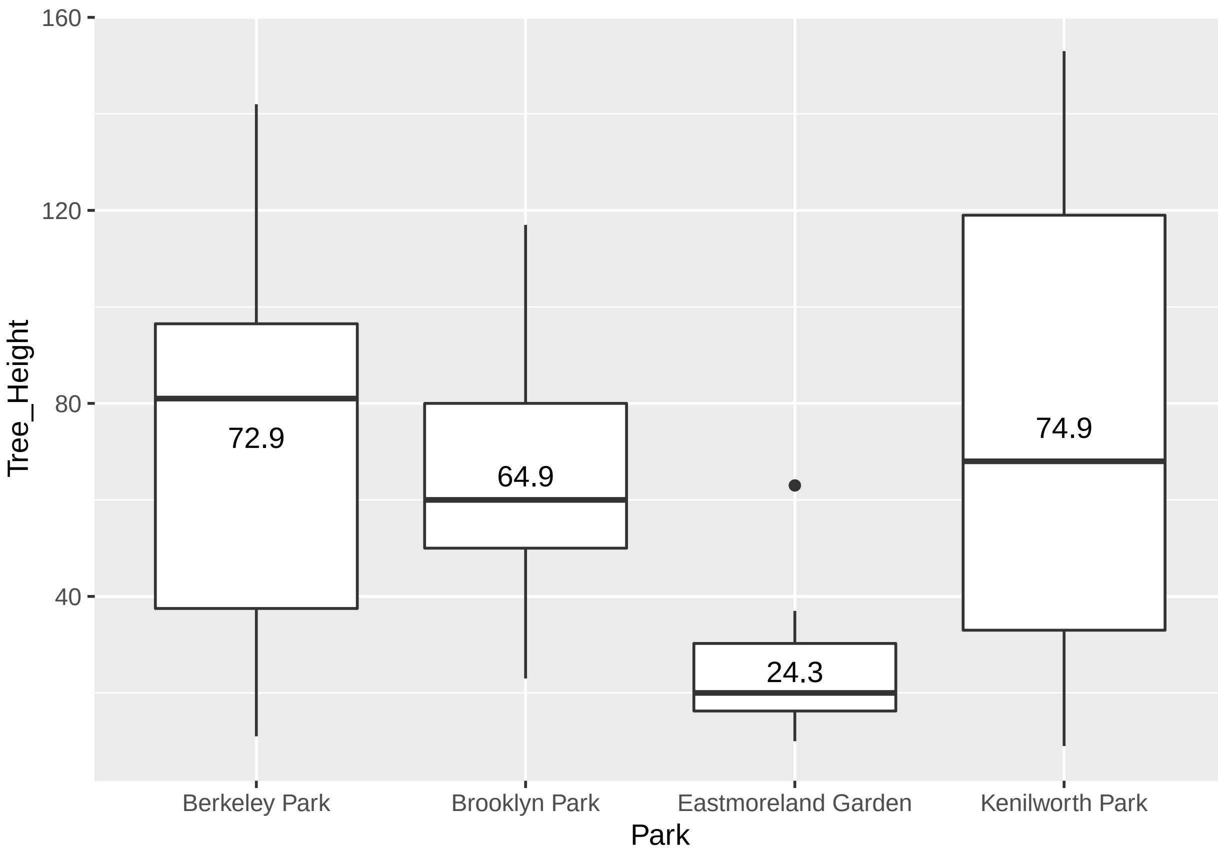
```
# Tallest tree data frame
tallest <- four_parks %>%
  filter(Tree_Height == max(Tree_Height)) %>%
  select(Tree_Height, Common_Name, Inventory_Date, Park)
# Print wrangled data frame
tallest
```

```
## # A tibble: 1 x 4
##   Tree_Height Common_Name Inventory_Date      Park
##         <dbl> <chr>       <dttm>              <chr>
```

```
## 1         153 Douglas-Fir 2018-07-26 00:00:00 Kenilworth Park
```

b. Generate a boxplot that shows the tree heights by park. Make sure to calculate and include the mean of the tree height rounded to the tenth place on the graph.

```
means <- aggregate(Tree_Height ~  Park, four_parks, mean)

ggplot(data = four_parks, mapping = aes(x = Park, y= Tree_Height)) +
  geom_boxplot()  +
  geom_text(data = means, aes(label = round(Tree_Height, 1), y = Tree_Height + 0.1))
```



c.Generate a bootstrap distribution with Tree_Height and DBH (Diameter at Breast Height). Drop any NA values. Have repetitions of bootstrapping at 1000.

```
bootstrap_dist <- four_parks %>%
  drop_na(Tree_Height, DBH) %>%
  specify(Tree_Height ~ DBH) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "correlation")
bootstrap_dist
```

```
## # A tibble: 1,000 x 2
##    replicate  stat
##        <int> <dbl>
##  1         1 0.862
##  2         2 0.856
##  3         3 0.847
##  4         4 0.867
##  5         5 0.871
```

```
## 6           6 0.836
## 7           7 0.850
## 8           8 0.849
## 9           9 0.859
## 10         10 0.858
## # ... with 990 more rows
```

C. Hard questions a. Using the 'four__parks' dataset, create a dataset called 'top2__park__op' with the two parks with the most "Douglas-Fir" trees. After this, change the name "Douglas-Fir" to "Oregon Pine". Do not remove n. You will be assessed on the accuracy of "top2__park__op" data set.

```
top2_park_op <- four_parks %>%
  group_by(Park) %>%
  count(Common_Name) %>%
  filter(Common_Name == "Douglas-Fir") %>%
  arrange(desc(n)) %>%
  ungroup() %>%
  slice_head(n = 2) %>%
  mutate(Common_Name = case_when(Common_Name == "Douglas-Fir" ~ "Oregon Pine",
                                 TRUE ~ as.character(Common_Name)))
top2_park_op
```

```
## # A tibble: 2 x 3
##   Park            Common_Name     n
##   <chr>           <chr>       <int>
## 1 Kenilworth Park Oregon Pine    82
## 2 Berkeley Park   Oregon Pine    18
```

b. Compute the ANOVA test statistic (F_o) for Tree__Height by Park in four__parks. and also produce an ANOVA table using 'aov()'. Now generate a simulation-based null distribution of the test statistics for Tree Heights, round up the xintercept to the nearest tenth place.

```
th_four_parks <- four_parks %>%
  specify(Tree_Height ~ Park) %>%
  calculate(stat = "F")
th_four_parks
```

```
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  6.29
```

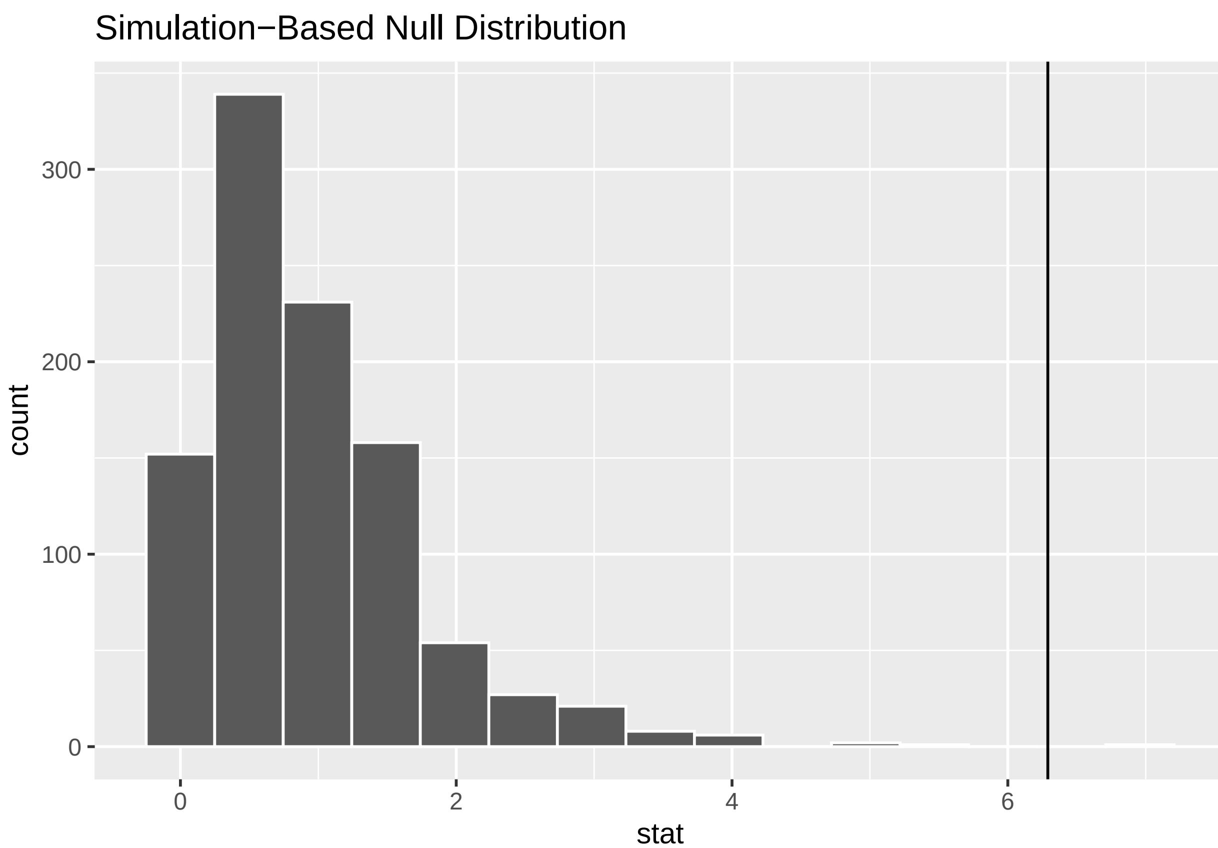
```
mod <- aov(Tree_Height ~ Park, data = four_parks)
tidy(mod)
```

```
## # A tibble: 2 x 6
##   term         df   sumsq meansq statistic   p.value
##   <chr>     <dbl>   <dbl>  <dbl>     <dbl>     <dbl>
## 1 Park          3  30582. 10194.      6.29  0.000358
## 2 Residuals   378 612699.  1621.     NA    NA
```

```
null_dist_four_parks <- four_parks %>%
  specify(Tree_Height ~ Park) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "F")
null_dist_four_parks
```

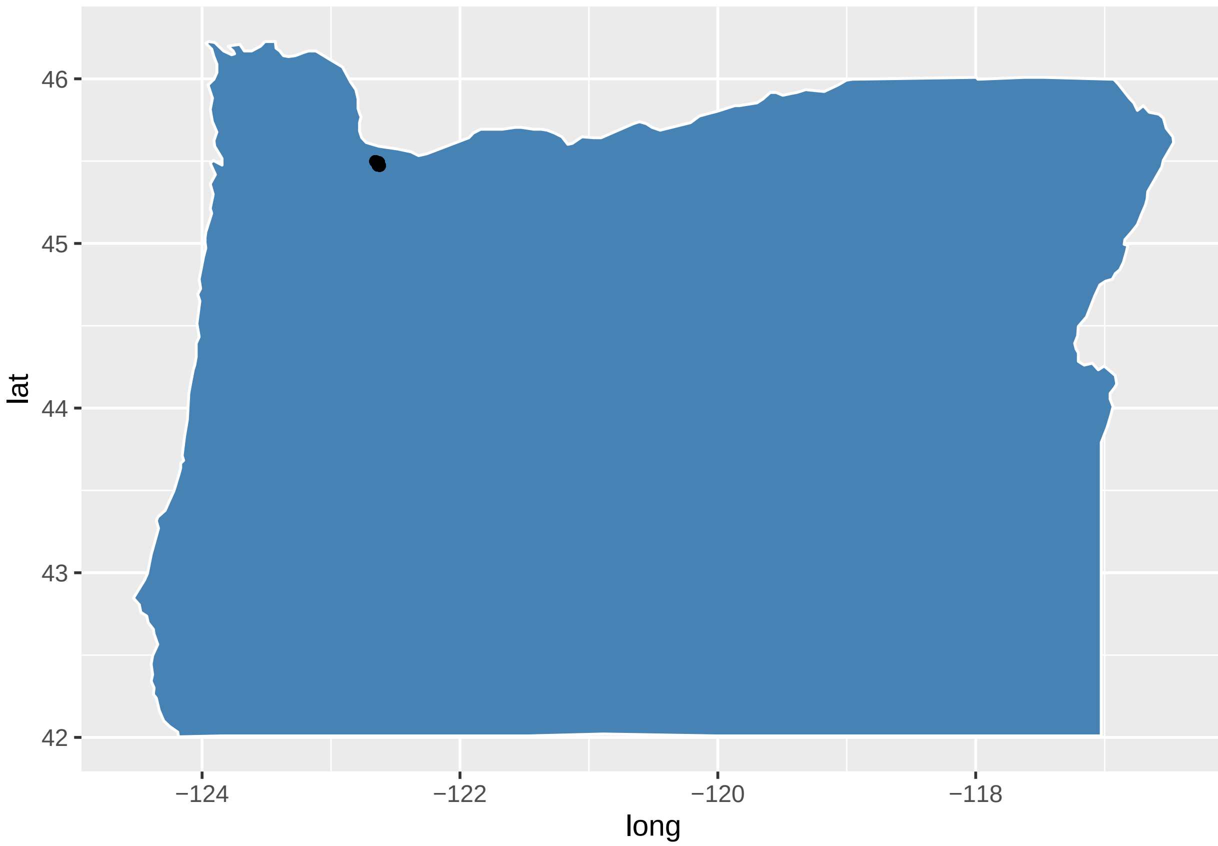
```
## # A tibble: 1,000 x 2
##    replicate   stat
##        <int>  <dbl>
##  1         1 0.0685
##  2         2 0.422
##  3         3 0.295
##  4         4 0.825
##  5         5 1.59
##  6         6 1.12
##  7         7 0.807
##  8         8 0.793
##  9         9 0.270
## 10        10 0.245
## # ... with 990 more rows
```

```
visualize(null_dist_four_parks) +
  geom_vline(xintercept= 6.29)
```



c. Try to create a map of Oregon with coordinate of trees in four_parks. Make the fill of the map "steelblue", and color = "white".

```
us_states <- map_data("state")
oregon <- subset(us_states, region == "oregon")

ggplot(data = oregon) +
  geom_polygon(aes(x = long, y = lat, group = group), fill = "steelblue", color = "white") +
  geom_point(data = four_parks, (aes(x = Longitude ,y = Latitude)))
```


46
45
44
43
42
lat
−124
−122
−120
−118
long