Untitled

Elyse Adamic

10/10/2019

Exercise 1: here::here

Q: What is the value of the here::here package? A: This package essentially abstracts from specifying explicit paths, and allows your code to be more accesible between users, especially between Mac and Windows due to the "/". It makes it clear which sub-directories are within a project in an organized manner. It finds the project directory, and is robust to human error in writing file path names, as well as avoiding a mess if you rename or move directories. No one wants to waste time fidding with file paths on some code to make a code run locally on your computer, especially when going back and forth with collaborators. When you specify `here::here` it sets the top level of the project folder as "here" and you specify where files are relative to that top level. This is simply good a coding practice, similar to never setting a working directory or `rm(list = ls())`. Furthermore, you should just always set up a project, as R "refreshes" whenever you open a new project, and has a default working directory.

```
library(here)
```

```
## here() starts at /Users/elyseadamic/stat545-hw-elyseadamic
```

```
here()
```

```
## [1] "/Users/elyseadamic/stat545-hw-elyseadamic"
```

Exercise 2: Factors

Choose a dataset and a factor variable to explore by 1) Drop factor / levels and 2) Reorder levels based on knowledge from data.

Then explore the effects by 1) Comparing the results of arrange on the original and re-leveled factor and 2) Plotting a figure of before/after re-leveling the factor.

```
# Check to see variable is factor:
gapminder$continent %>%
  str()
```

```
##  Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```
nrow(gapminder)
```

```
## [1] 1704
```

```
nlevels(gapminder$continent) #5 continents
```

```
## [1] 5
```

```
levels(gapminder$continent)
```

```
## [1] "Africa"   "Americas" "Asia"     "Europe"   "Oceania"
```

```
# Drop Oceania
gap <- gapminder %>%
  filter(continent != "Oceania")

nlevels(gap$continent) #note there are still 5 levels!
```

```
## [1] 5
```

```
nrow(gap)
```

```
## [1] 1680
```

```
gap_dropped <- gap %>%
  droplevels()

nlevels(gap_dropped$continent) # now only 4 levels
```

```
## [1] 4
```

```
levels(gap_dropped$continent) #Oceania is gone
```

```
## [1] "Africa"   "Americas" "Asia"     "Europe"
```

First note without reordering, the variables are just in alphabetical order.

```
# Comparing results of arrange
# This is not as clear as figures - but here Asia comes before Americas
gap_dropped %>%
  filter(year == 2007) %>%
  arrange(fct_reorder(continent, gdpPercap, min))
```

```
## # A tibble: 140 x 6
##    country                  continent  year lifeExp      pop gdpPercap
##    <fct>                    <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Algeria                  Africa     2007    72.3 33333216     6223.
##  2 Angola                   Africa     2007    42.7 12420476     4797.
##  3 Benin                    Africa     2007    56.7  8078314     1441.
##  4 Botswana                 Africa     2007    50.7  1639131    12570.
##  5 Burkina Faso             Africa     2007    52.3 14326203     1217.
##  6 Burundi                  Africa     2007    49.6  8390505      430.
##  7 Cameroon                 Africa     2007    50.4 17696293     2042.
##  8 Central African Republic Africa     2007    44.7  4369038      706.
##  9 Chad                     Africa     2007    50.7 10238807     1704.
## 10 Comoros                  Africa     2007    65.2   710960      986.
## # ... with 130 more rows
```

```
# This more clearly shows the order has been flipped
gap_order <- gap_dropped %>%
  filter(year == 2007) %>%
  mutate(continent = fct_reorder(continent, gdpPercap, min))

gap_order$continent %>% levels()
```

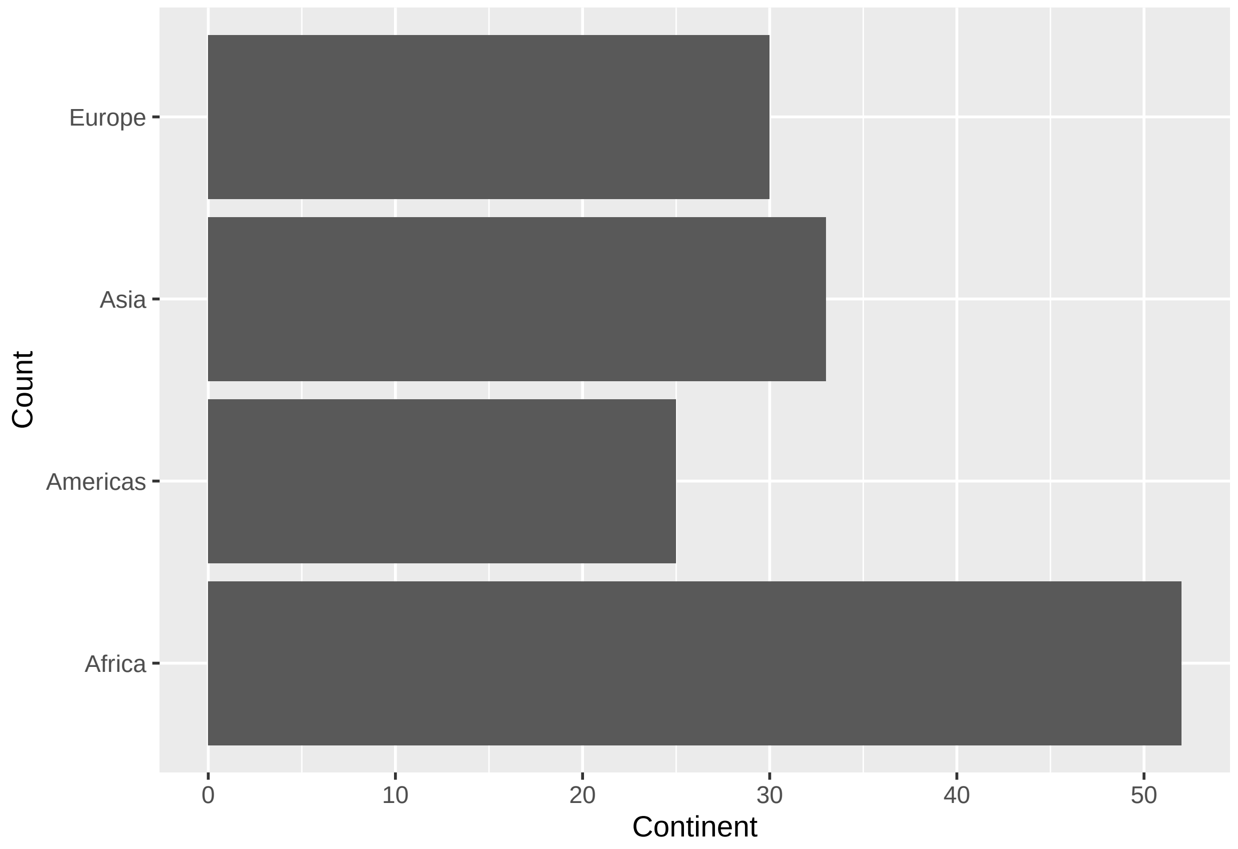
```
## [1] "Africa"   "Asia"     "Americas" "Europe"
```

```
gap_dropped$continent %>% levels()
```

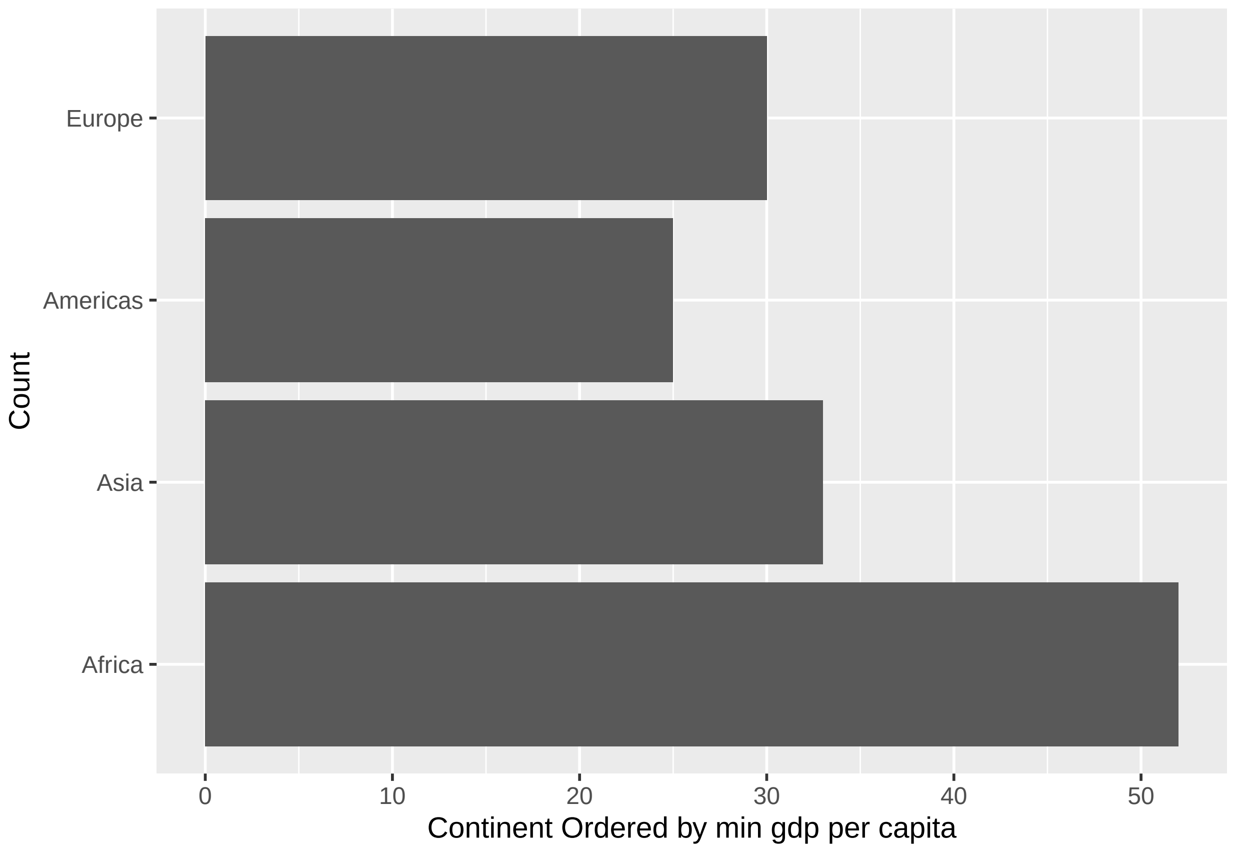
```
## [1] "Africa"   "Americas" "Asia"     "Europe"
```

```
# Unordered/ BEFORE
gap_dropped %>%
  filter(year == 2007) %>%
  ggplot(aes(x=continent)) +
  geom_bar() +
  coord_flip() +
```

```r
  xlab("Count") + ylab("Continent")
```



```r
# Ordered/ AFTER
gap_dropped %>%
  filter(year == 2007) %>%
  ggplot(aes(fct_reorder(continent, gdpPercap, min))) +
  geom_bar() +
  coord_flip()+
  xlab("Count") + ylab("Continent Ordered by min gdp per capita")
```



```
# Note there are many features to the forcats package:
gap_dropped %>%
  filter(year == 2007) %>%
  mutate(continent = fct_lump(continent, n = 2)) %>%
  count(continent)
```

```
## # A tibble: 3 x 2
##   continent     n
##   <fct>     <int>
## 1 Africa       52
## 2 Asia         33
## 3 Other        55
```

Exercise 3: File input/output

1) Export a grouped dataset to .csv

```
gapminder_group <- gapminder %>%
  group_by(country) %>%
  summarize(ave_lifeExp = mean(lifeExp), ave_gdpPercap = mean(gdpPercap))

write_csv(gapminder_group,here::here("hw05_EA","gapminder_group.csv"))
```

2) Read the dataset back in, it survived the round trip! Then play with factors again. Note saveRDS()/readRDS() is for the R data format .rds and dput()/dget is for ASCII.

```
gap_read <- read_csv(here::here("hw05_EA","gapminder_group.csv"))
```

```
## Parsed with column specification:
## cols(
##   country = col_character(),
##   ave_lifeExp = col_double(),
##   ave_gdpPercap = col_double()
## )
```

```
gap_read %>%
  arrange(fct_reorder(country, ave_lifeExp, min))
```

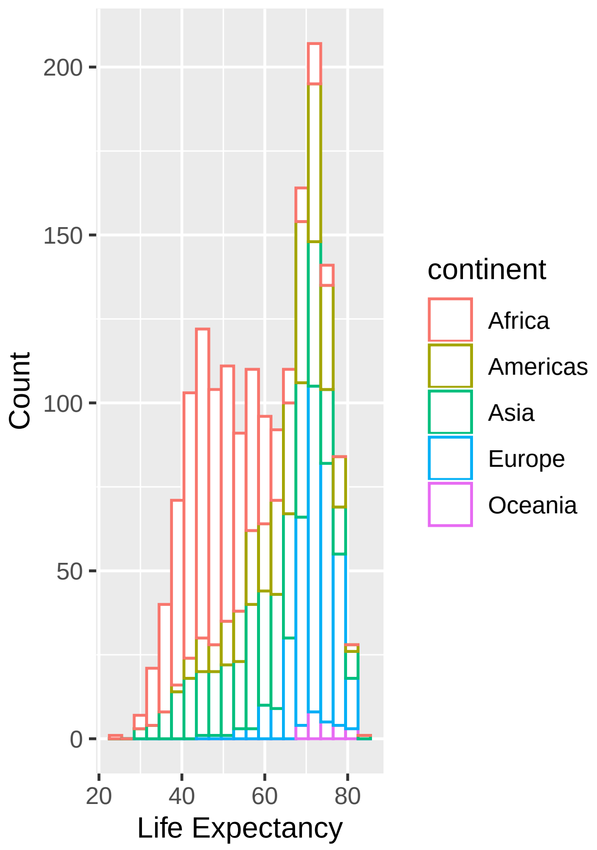
```
## # A tibble: 142 x 3
##    country            ave_lifeExp ave_gdpPercap
##    <chr>                    <dbl>         <dbl>
##  1 Sierra Leone              36.8         1073.
##  2 Afghanistan               37.5          803.
##  3 Angola                    37.9         3607.
##  4 Guinea-Bissau             39.2          652.
##  5 Mozambique                40.4          542.
##  6 Somalia                   41.0         1141.
##  7 Rwanda                    41.5          676.
##  8 Liberia                   42.5          605.
##  9 Equatorial Guinea         43.0         2469.
## 10 Guinea                    43.2          776.
## # ... with 132 more rows
```

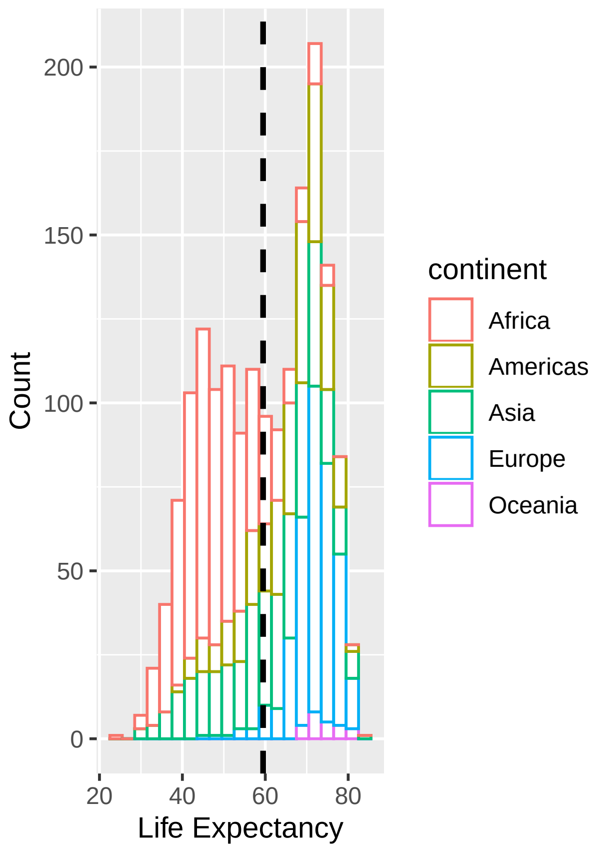
Exercise 4: Visualization Design

Juxtapose one of the first graphs you made with new skills.

```
hist <- ggplot (gapminder, aes(x = lifeExp, color = continent)) +
  geom_histogram(fill = "white", binwidth = 3) +
  xlab("Life Expectancy") +
  ylab("Count")

hist1 <- hist + geom_vline(aes(xintercept=mean(lifeExp)),
            color="black", linetype="dashed", size=1)
grid.arrange(hist,hist1,ncol = 2)
```





Exercise 5: Writing figures to files

Stat 545 [::big check mark::]