

# **An Incomplete Review of Descriptive Statistics Commands in R**

**EDP 613**





**Week 7**

# Packages needed and a Note about Icons

Please load up the `tidyverse` package

```
library(tidyverse)
```

You may come across the following icons. The table below lists what each means.

Icon	Description
	Indicates that an example continues on the following slide.
	Indicates that a section using common syntax has ended.
	Indicates that there is an active hyperlink on the slide.
	Indicates that a section covering a concept has ended.

# Descriptives

We're going to use the Star Wars data set that's included in `dplyr`

```
data(starwars)
```

```
starwars
```

```
## # A tibble: 87 x 14
##   name      height  mass hair_color skin_color eye_color birth_year
##   <chr>    <int> <dbl> <chr>      <chr>    <chr>      <dbl>
## 1 Luke...    172    77 blond      fair      blue        19
## 2 C-3PO     167    75 <NA>      gold      yellow     112
## 3 R2-D2      96    32 <NA>      white, bl... red         33
## 4 Dart...   202   136 none      white     yellow     41.9
## 5 Leia...   150    49 brown     light     brown       19
## 6 Owen...   178   120 brown, gr... light     blue       52
## 7 Beru...   165    75 brown     light     blue       47
## 8 R5-D4      97    32 <NA>      white, red red        NA
## 9 Bigg...   183    84 black     light     brown       24
## 10 Obi-...   182    77 auburn, w... fair      blue-gray   57
## # ... with 77 more rows, and 7 more variables: sex <chr>,
## #   gender <chr>, homeworld <chr>, species <chr>, films <list>,
## #   vehicles <list>, starships <list>
```

# View a Portion of the Data Set

```
head(starwars)
```

```
## # A tibble: 6 x 14
##   name    height  mass hair_color skin_color eye_color birth_year
##   <chr>   <int> <dbl> <chr>      <chr>      <chr>      <dbl>
## 1 Luke...   172    77 blond      fair        blue        19
## 2 C-3PO     167    75 <NA>      gold        yellow      112
## 3 R2-D2      96    32 <NA>      white, bl... red         33
## 4 Dart...   202   136 none      white       yellow     41.9
## 5 Leia...   150    49 brown     light       brown       19
## 6 Owen...   178   120 brown, gr... light       blue       52
## # ... with 7 more variables: sex <chr>, gender <chr>,
## #   homeworld <chr>, species <chr>, films <list>, vehicles <list>,
## #   starships <list>
```

# Counts

## Total number of names

Tidy approach

```
nrow(starwars %>%  
  select(name))
```

```
## [1] 87
```

Base R approach

```
length(starwars$name)
```

```
## [1] 87
```



# Column Types

Using `str`

```
str(starwars)
```

```
## tibble [87 × 14] (S3: tbl_df/tbl/data.frame)
## $ name      : chr [1:87] "Luke Skywalker" "C-3P0" "R2-D2" "Darth Vader" ..
## $ height    : int [1:87] 172 167 96 202 150 178 165 97 183 182 ...
## $ mass      : num [1:87] 77 75 32 136 49 120 75 32 84 77 ...
## $ hair_color: chr [1:87] "blond" NA NA "none" ...
## $ skin_color: chr [1:87] "fair" "gold" "white, blue" "white" ...
## $ eye_color : chr [1:87] "blue" "yellow" "red" "yellow" ...
## $ birth_year: num [1:87] 19 112 33 41.9 19 52 47 NA 24 57 ...
## $ sex       : chr [1:87] "male" "none" "none" "male" ...
## $ gender    : chr [1:87] "masculine" "masculine" "masculine" "masculine" .
## $ homeworld : chr [1:87] "Tatooine" "Tatooine" "Naboo" "Tatooine" ...
## $ species   : chr [1:87] "Human" "Droid" "Droid" "Human" ...
## $ films     :List of 87
## ..$ : chr [1:5] "The Empire Strikes Back" "Revenge of the Sith" "Return o
## ..$ : chr [1:6] "The Empire Strikes Back" "Attack of the Clones" "The Pha
## ..$ : chr [1:7] "The Empire Strikes Back" "Attack of the Clones" "The Pha
## ..$ : chr [1:4] "The Empire Strikes Back" "Revenge of the Sith" "Return o
## ..$ : chr [1:5] "The Empire Strikes Back" "Revenge of the Sith" "Return o
## ..$ : chr [1:3] "Attack of the Clones" "Revenge of the Sith" "A New Hope"
## ..$ : chr [1:3] "Attack of the Clones" "Revenge of the Sith" "A New Hope"
## ..$ : chr "A New Hope"
```

## Using `glimpse`

```
glimpse(starwars)
```

```
## Rows: 87
## Columns: 14
## $ name      <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vad...
## $ height    <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182...
## $ mass      <dbl> 77.0, 75.0, 32.0, 136.0, 49.0, 120.0, 75.0, 32...
## $ hair_color <chr> "blond", NA, NA, "none", "brown", "brown, grey...
## $ skin_color <chr> "fair", "gold", "white, blue", "white", "light...
## $ eye_color  <chr> "blue", "yellow", "red", "yellow", "brown", "b...
## $ birth_year <dbl> 19.0, 112.0, 33.0, 41.9, 19.0, 52.0, 47.0, NA,...
## $ sex        <chr> "male", "none", "none", "male", "female", "mal...
## $ gender     <chr> "masculine", "masculine", "masculine", "mascul...
## $ homeworld  <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "...
## $ species    <chr> "Human", "Droid", "Droid", "Human", "Human", "...
## $ films      <list> [<"The Empire Strikes Back", "Revenge of the ...
## $ vehicles   <list> [<"Snowspeeder", "Imperial Speeder Bike">, <>...
## $ starships  <list> [<"X-wing", "Imperial shuttle">, <>, <>, "TIE...
```



# Frequencies

```
starwars %>%  
  count(sex)
```

```
## # A tibble: 5 x 2  
##   sex          n  
##   <chr>      <int>  
## 1 female      16  
## 2 hermaphroditic 1  
## 3 male       60  
## 4 none        6  
## 5 <NA>        4
```





# Measures of Central Tendency

## Mean

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(mean(birth_year)) %>%  
  rename(`mean age by species` = `mean(birth_year)`) %>%  
  ungroup()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 11 x 2  
##   species      `mean age by species`  
##   <chr>                <dbl>  
## 1 Cerean                92  
## 2 Ewok                   8  
## 3 Gungan                52  
## 4 Human                45.5  
## 5 Kel Dor               22  
## 6 Mirialan              49  
## 7 Mon Calamari          41  
## 8 Trandoshan            53  
## 9 Twi'lek               48  
## 10 Wookiee              200  
## 11 Zabrak               54
```

## Side Note: Using Base R vs. Tidy

Either is fine but think about the outcome and what you're going to do with it. Let's take the `mean` again with a fake data set from taste test ratings using two varieties of bananas: cavendish and ice cream.

```
banana_data <- tibble(  
  id = c(1,2,3,4,5),  
  cav_cat = c("Excellent", "Above Average", "Very Poor", "Average",  
              "Excellent"),  
  cav_code = c(5,4,1,3,5),  
  ic_cat = c("Excellent", "Excellent", "Above Average", "Excellent",  
             "Excellent"),  
  ic_code = c(5,5,4,5,5)  
)
```



If you are wondering, the blue java - aka the ice cream banana is real.

```
banana_data
```

```
## # A tibble: 5 x 5
##       id cav_cat      cav_code ic_cat      ic_code
##   <dbl> <chr>      <dbl> <chr>      <dbl>
## 1     1    Excellent      5 Excellent      5
## 2     2  Above Average      4 Excellent      5
## 3     3   Very Poor      1 Above Average      4
## 4     4    Average      3 Excellent      5
## 5     5    Excellent      5 Excellent      5
```



If we just wanted to find the means, then the Base R method is likely simpler

```
mean(banana_data$cav_code)
```

```
## [1] 3.6
```

```
mean(banana_data$ic_code)
```

```
## [1] 4.8
```

but if we wanted to pass that output along say to find the range of the means, the Tidy way is a simpler and more efficient approach

```
banana_data %>%  
  summarise(mean_cav = mean(cav_code),  
            mean_ic = mean(ic_code)) %>%  
  mutate(range_means = mean_ic - mean_cav)
```

```
## # A tibble: 1 x 3  
##   mean_cav mean_ic range_means  
##   <dbl>   <dbl>   <dbl>  
## 1     3.6     4.8     1.20
```

## Median

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(median(birth_year)) %>%  
  rename(`median age by species` = `median(birth_year)`) %>%  
  ungroup()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 11 x 2  
##   species      `median age by species`  
##   <chr>          <dbl>  
## 1 Cerean          92  
## 2 Ewok             8  
## 3 Gungan         52  
## 4 Human         41.9  
## 5 Kel Dor        22  
## 6 Mirialan       49  
## 7 Mon Calamari   41  
## 8 Trandoshan     53  
## 9 Twi'lek        48  
## 10 Wookiee      200  
## 11 Zabrak        54
```

## Mode

Remember that `mode` means something else in R. Instead first run the chunk below

```
Mode <- function(x) {  
  ux <- unique(x)  
  ux[which.max(tabulate(match(x, ux)))]  
}
```



Then you can find the **Mode**

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(Mode(birth_year)) %>%  
  rename(`mode age by species` = `Mode(birth_year)`) %>%  
  ungroup()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 11 x 2  
##   species      `mode age by species`  
##   <chr>          <dbl>  
## 1 Cerean          92  
## 2 Ewok             8  
## 3 Gungan         52  
## 4 Human          19  
## 5 Kel Dor        22  
## 6 Mirialan       58  
## 7 Mon Calamari   41  
## 8 Trandoshan     53  
## 9 Twi'lek        48  
## 10 Wookiee      200  
## 11 Zabrak        54
```

**Mean by Value** Sometimes you need to find a measure for a particular subgroup. In this example, we only want to find the birth year for the female classification of each species. One way to accomplish this is to use the `filter` command to get what we want.

```
starwars %>%
  filter(gender == "feminine") %>%
  group_by(species) %>%
  na.omit() %>%
  summarise(mean(birth_year)) %>%
  rename(`female mean age by species` = `mean(birth_year)`) %>%
  ungroup() %>%
  na.omit()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 2
##   species `female mean age by species`
##   <chr>          <dbl>
## 1 Human          37.3
## 2 Mirialan       49
## 3 Twi'lek        48
```



# Measures of Spread

## Sample Standard Deviation

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(sd(birth_year)) %>%  
  rename(`age standard deviation by species` = `sd(birth_year)`) %>%  
  ungroup() %>%  
  na.omit()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2  
##   species `age standard deviation by species`  
##   <chr>                                <dbl>  
## 1 Human                                23.1  
## 2 Mirialan                             12.7
```

## Population Standard Deviation

Reflecting common assumptions and practice, most descriptive statistics in R do not assume that you have an entire population. So you should *always assume that you have a sample unless the description explicitly says otherwise*. When you do come across with a population, run the following first

```
pop_sd <- function(x) sd(x) * (length(x)-1) / length(x)
```

and then calculate the population standard deviation

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(pop_sd(birth_year)) %>%  
  rename(`age standard deviation by species` = `pop_sd(birth_year)`) %>%  
  ungroup() %>%  
  na.omit()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2  
##   species `age standard deviation by species`  
##   <chr>                                <dbl>  
## 1 Human                                21.8  
## 2 Mirialan                             6.36
```

## Sample Variance

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(var(birth_year)) %>%  
  rename(`age variance by species` = `var(birth_year)`) %>%  
  ungroup() %>%  
  na.omit()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2  
##   species `age variance by species`  
##   <chr>          <dbl>  
## 1 Human          533.  
## 2 Mirialan       162
```

You can find a list of commands that may be used with [summarise](#) [here](#)

## Population Variance

Paralleling the argument given about the population standard deviation, when you have a known population and want to find the variance, first run

```
pop_var <- function(x) var(x) * (length(x)-1) / length(x)
```

and then you can calculate the population variance

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(pop_var(birth_year)) %>%  
  rename(`age variance by species` = `pop_var(birth_year)`) %>%  
  ungroup() %>%  
  na.omit()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2  
##   species `age variance by species`  
##   <chr>          <dbl>  
## 1 Human          504.  
## 2 Mirialan       81
```



## Side Note: More about Summarise

You can find a list of commands that may be used with `summarise` [here](#)



# Range

## Maximum

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(max(birth_year)) %>%  
  rename(`maximum age by species` = `max(birth_year)`) %>%  
  ungroup() %>%  
  na.omit()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 11 x 2  
##   species      `maximum age by species`  
##   <chr>                <dbl>  
## 1 Cerean                92  
## 2 Ewok                   8  
## 3 Gungan               52  
## 4 Human              102  
## 5 Kel Dor              22  
## 6 Mirialan             58  
## 7 Mon Calamari         41  
## 8 Trandoshan           53  
## 9 Twi'lek              48  
## 10 Wookiee            200
```

## Minimum

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(min(birth_year)) %>%  
  rename(`minimum age by species` = `min(birth_year)`) %>%  
  ungroup() %>%  
  na.omit()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 11 x 2  
##   species      `minimum age by species`  
##   <chr>          <dbl>  
## 1 Cerean          92  
## 2 Ewok             8  
## 3 Gungan         52  
## 4 Human          19  
## 5 Kel Dor        22  
## 6 Mirialan       40  
## 7 Mon Calamari   41  
## 8 Trandoshan     53  
## 9 Twi'lek        48  
## 10 Wookiee      200  
## 11 Zabrak        54
```

## Range

```
starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(max(birth_year), min(birth_year)) %>%  
  rename(`max` = `max(birth_year)`) %>%  
  rename(`min` = `min(birth_year)`) %>%  
  ungroup() %>%  
  na.omit() %>%  
  mutate(`age range by species` = max - min)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 11 x 4  
##   species      max    min `age range by species`  
##   <chr>      <dbl> <dbl>          <dbl>  
## 1 Cerean      92     92             0  
## 2 Ewok         8      8             0  
## 3 Gungan     52     52             0  
## 4 Human     102     19            83  
## 5 Kel Dor     22     22             0  
## 6 Mirialan    58     40            18  
## 7 Mon Calamari 41     41             0  
## 8 Trandoshan  53     53             0  
## 9 Twi'lek     48     48             0  
## 10 Wookiee   200    200             0  
## 11 Zabrak     54     54             0
```





# Plots

We'll use the sample mean age for all of the species

```
starwars_by_species <- starwars %>%  
  group_by(species) %>%  
  na.omit() %>%  
  summarise(mean(birth_year)) %>%  
  rename(`mean age by species` = `mean(birth_year)`) %>%  
  ungroup() %>%  
  arrange(`mean age by species`)
```

## `summarise()` ungrouping output (override with `.groups` argument)

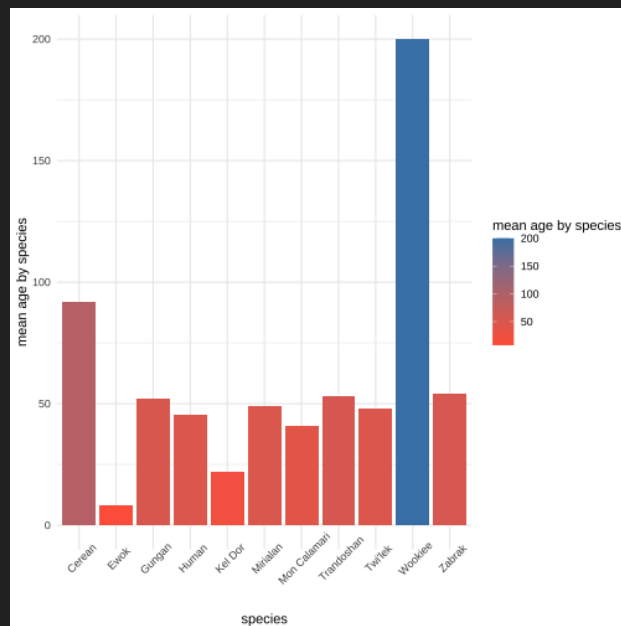
## Side Note: R Graph Gallery

Want some inspiration or just want to copy code? Good! Head over to [The R Graph Gallery](#) to see some examples of basic visualizations and plots you can do in R right now.



# Bar Plot

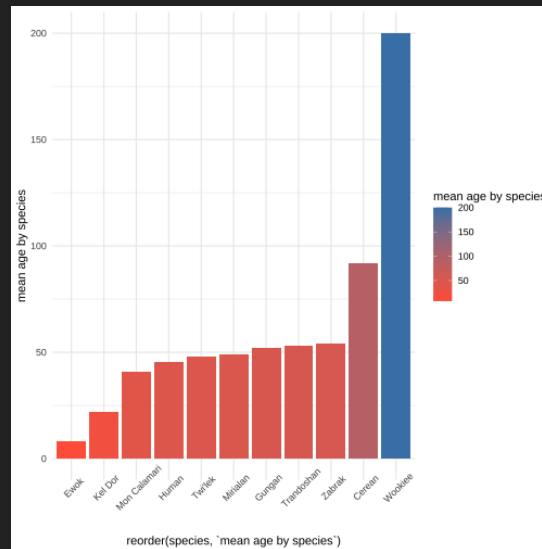
```
ggplot(starwars_by_species, aes(x = species,  
                                y = `mean age by species`,  
                                fill = `mean age by species`)) +  
  geom_bar(stat='identity') +  
  scale_fill_gradient(low = "tomato",  
                     high = "steelblue") +  
  theme_minimal() +  
  theme(axis.text.x = element_text(angle = 45))
```



# Ordered Bar Plot

Use `reorder` to order the bars from least to greatest.

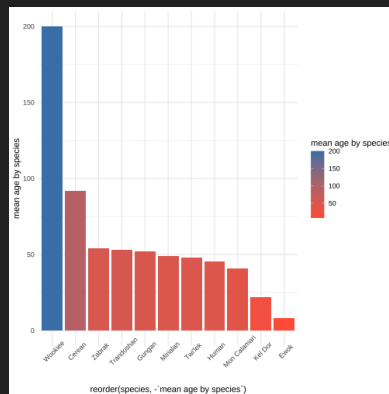
```
ggplot(starwars_by_species, aes(x = reorder(species,  
                                             `mean age by species`),  
                                y = `mean age by species`,  
                                fill = `mean age by species`)) +  
  geom_bar(stat='identity') +  
  scale_fill_gradient(low = "tomato",  
                     high = "steelblue") +  
  theme_minimal() +  
  theme(axis.text.x = element_text(angle = 45))
```



# Side Note: The Other Direction

Want greatest to least? Stick a - in front of `mean age by species` within `reorder()`.

```
ggplot(starwars_by_species, aes(x = reorder(species,  
                                             -`mean age by species`),  
                                y = `mean age by species`,  
                                fill = `mean age by species`)) +  
  geom_bar(stat='identity') +  
  scale_fill_gradient(low = "tomato",  
                     high = "steelblue") +  
  theme_minimal() +  
  theme(axis.text.x = element_text(angle = 45))
```



## Side Note: Using Hex Color Codes

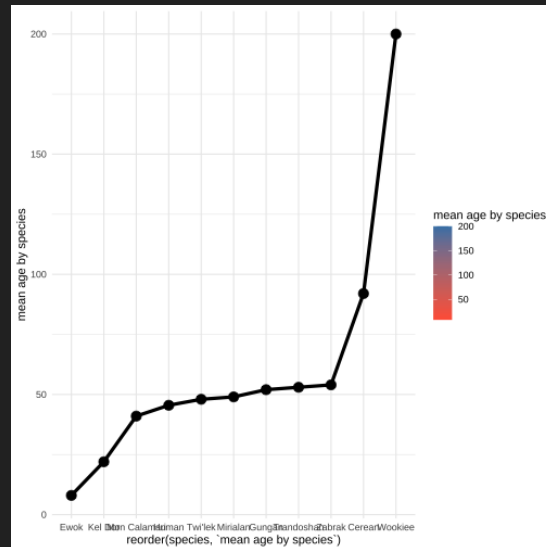
You could have used the hex color codes `#ff6347` for `tomato` and `#4682b4` for `steelblue`.

You can find more about hex colors using a web search for "hex color codes". For example, here is a `site` I use.



# Line Plot

```
ggplot(starwars_by_species, aes(x = reorder(species,  
                                             `mean age by species`),  
                                y = `mean age by species`,  
                                group = 1,  
                                fill = `mean age by species`)) +  
  geom_line(size = 1.5) +  
  geom_point(size = 4) +  
  scale_fill_gradient(low = "tomato",  
                     high = "steelblue") +  
  theme_minimal()
```



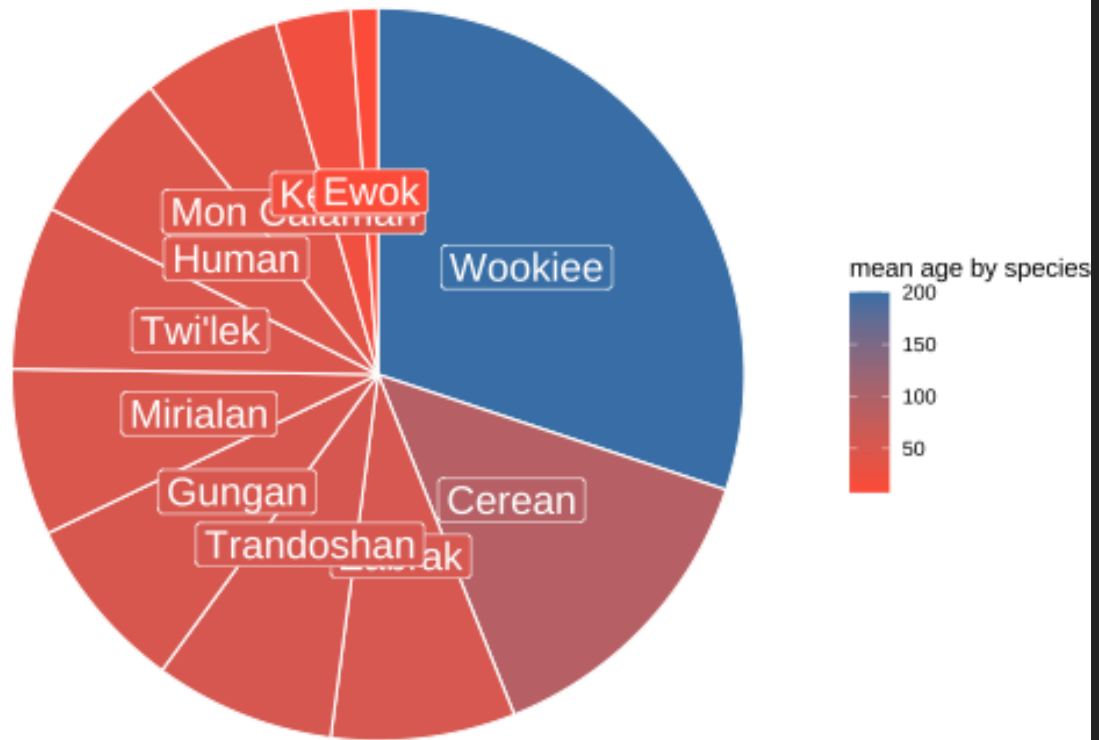
# Pie Chart

```
starwars_by_species_pc <- starwars_by_species %>%  
  arrange(desc(`mean age by species`)) %>%  
  mutate(prop = `mean age by species` /  
           sum(starwars_by_species$`mean age by species`) *100) %>%  
  mutate(ypos = cumsum(prop)- 0.5*prop )
```

```
ggplot(starwars_by_species_pc, aes(x = "",  
                                   y = prop,  
                                   fill = `mean age by species`)) +  
  geom_bar(stat="identity",  
           width=1,  
           color="white") +  
  coord_polar("y", start = 0) +  
  geom_label(aes(y = ypos,  
                 label = `species`),  
             color = "white",  
             size=6) +  
  scale_fill_gradient(low = "tomato",  
                     high = "steelblue") +  
  theme_void()
```







**Thats it!**