

Numeric Variable Graphs

Justin Post

Recap: ggplot2

A great plotting system!

To create plots:

- Create base object
- Add geom or stat layers
- Use `aes()` to map variables to attributes of the plot
- Add other layers to modify things

ggplot2 Smoothed Histogram

Numeric variables - generally, describe distribution via a histogram or boxplot!

- For a single numeric variable, describe the distribution via
 - Shape: Histogram, Density plot, ...
 - Comparing across a categorical variable: Boxplot
- For two numeric variables, describe the distribution via
 - Shape: Scatter plot

NC STATE UNIVERSITY

ndix_Diameter <dbl>, Migratory_Pain <chr>,
, Contralateral_Rebound_Tenderness <chr>,
a <chr>, Loss_of_Appetite <chr>,

```
## # A tibble: 782 x 61
##   Age      BMI Sex      Height Weight Length_of_Stay Management      Severity
##   <dbl> <dbl> <chr>    <dbl> <dbl>         <dbl> <chr>         <chr>
## 1  12.7   16.9 female    148    37             3 conservative uncomplicated
## 2  14.1   31.9 male      147   69.5           2 conservative uncomplicated
## 3  14.1   23.3 female    163    62             4 conservative uncomplicated
## 4  16.4   20.6 female    165    56             3 conservative uncomplicated
## 5  11.1   16.9 female    163    45             3 conservative uncomplicated
## # i 777 more rows
```

4 / 34

Density Plot

- **Kernel Smoother** - Smoothed version of a histogram
- Common aes values (from **cheat sheet**):

```
c + geom_density(kernel = "gaussian")
```

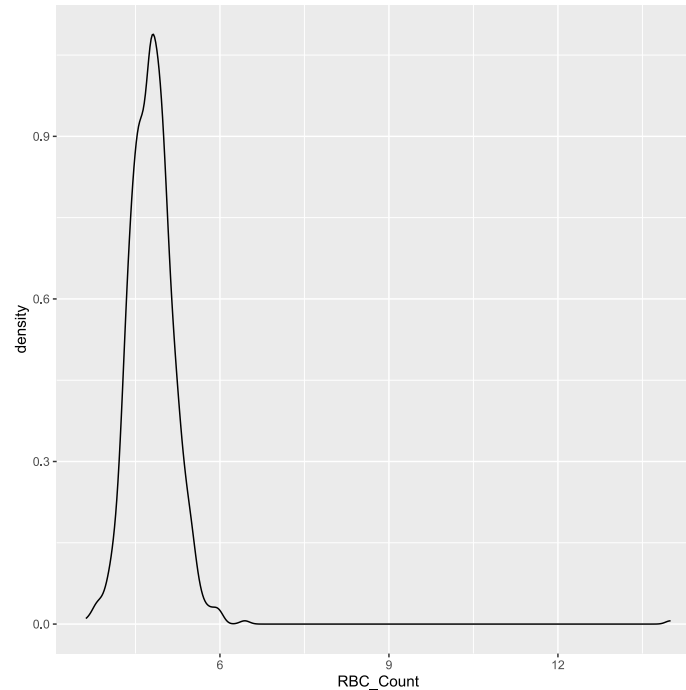
```
x, y, alpha, color, fill, group, linetype, size, weight
```

- Only `x` is really needed

ggplot2 Smoothed Histogram

- **Kernel Smoother** - Smoothed version of a histogram

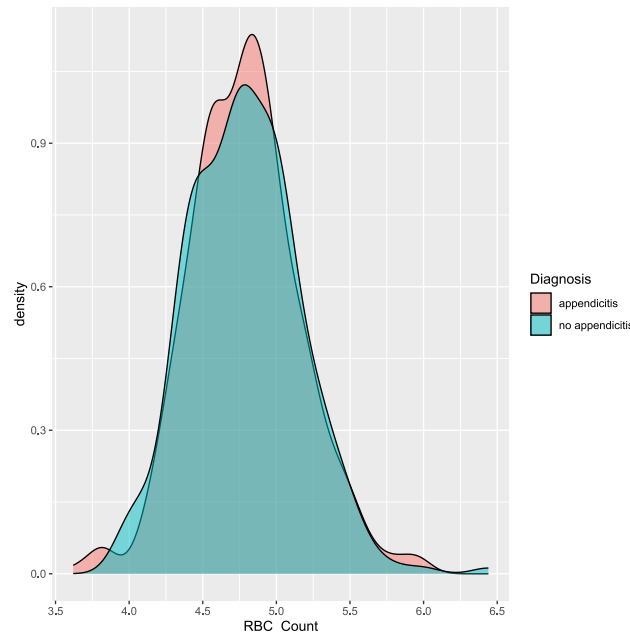
```
g <- ggplot(app_data |> drop_na(RBC_Count), aes(x = RBC_Count))  
g + geom_density()
```



ggplot2 Smoothed Histogram

- **Kernel Smoother** - Smoothed version of a histogram
- Remove really large value and use the fill aesthetic to compare groups!

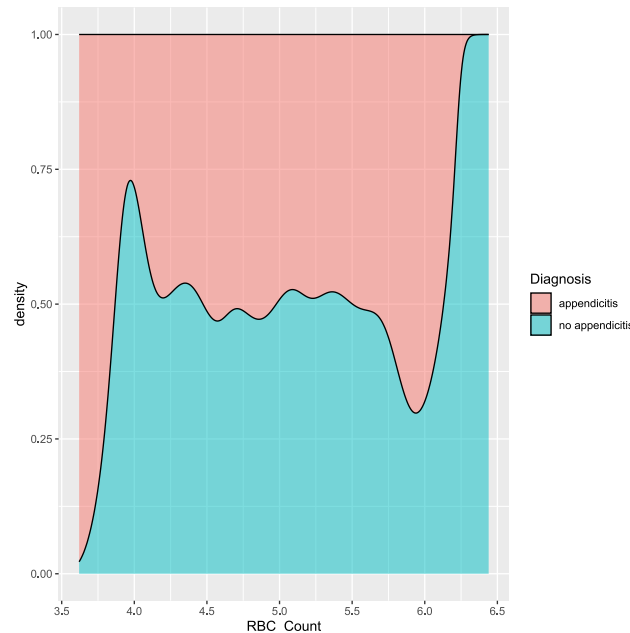
```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis) |> filter(RBC_Count < 8), aes(x = RBC_Count))  
g + geom_density(alpha = 0.5, aes(fill = Diagnosis))
```



ggplot2 Smoothed Histogram

- **Kernel Smoother** - Smoothed version of a histogram
- Recall position choices of dodge, jitter, fill, and stack

```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis) |> filter(RBC_Count < 8), aes(x = RBC_Count))  
g + geom_density(alpha = 0.5, position = "fill", aes(fill = Diagnosis))
```



ggplot2 **Boxplots**

- **Boxplot** - Provides the five number summary in a graph
- Common aes values (from cheat sheet):

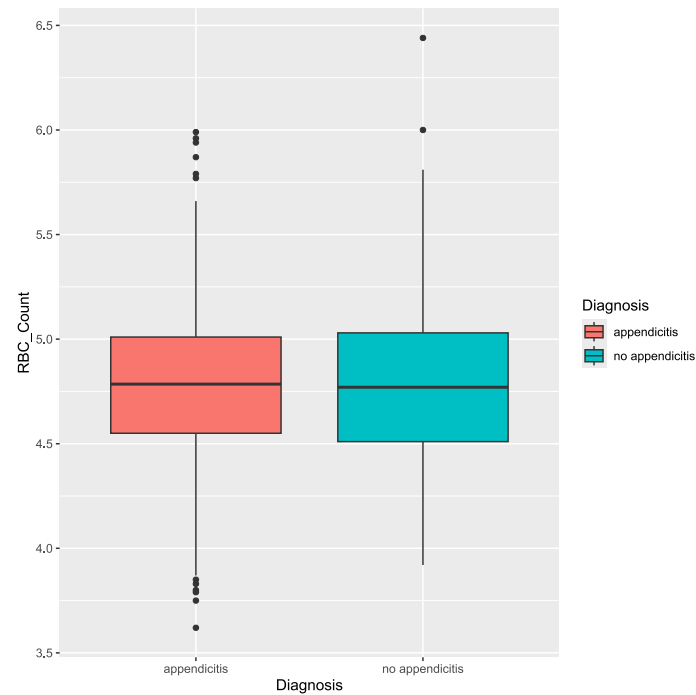
```
f + geom_boxplot()
```

x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

- Only x =, y = are really needed

ggplot2 Boxplots

```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis) |> filter(RBC_Count < 8))  
g + geom_boxplot(aes(x = Diagnosis, y = RBC_Count, fill = Diagnosis))
```



ggplot2 Boxplots with Points

- Can add data points (jittered) to see shape of data better (or use violin plot)

```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis) |> filter(RBC_Count < 8))  
g + geom_boxplot(aes(x = Diagnosis, y = RBC_Count, fill = Diagnosis)) +  
  geom_jitter(width = 0.1, alpha = 0.3)
```

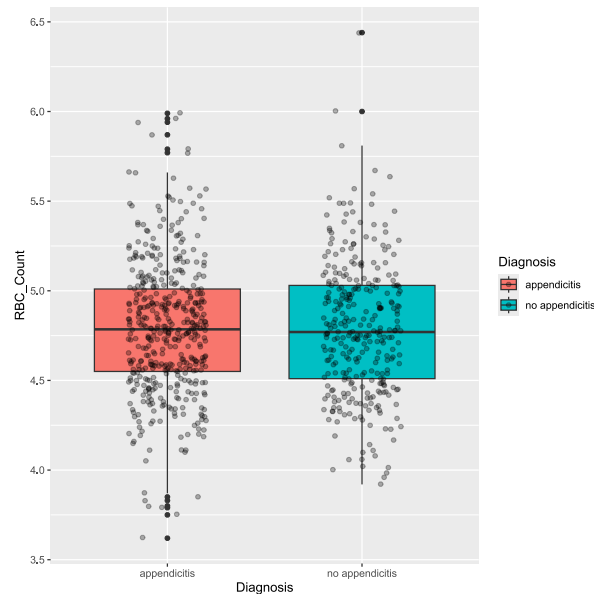
```
## Error in `geom_jitter()`:  
## ! Problem while setting up geom.  
## i Error occurred in the 2nd layer.  
## Caused by error in `compute_geom_1()`:  
## ! `geom_point()` requires the following missing aesthetics: x and y.
```

- Oh, global vs local aes()!

ggplot2 Boxplots with Points

- Can add data points (jittered) to see shape of data better (or use violin plot)

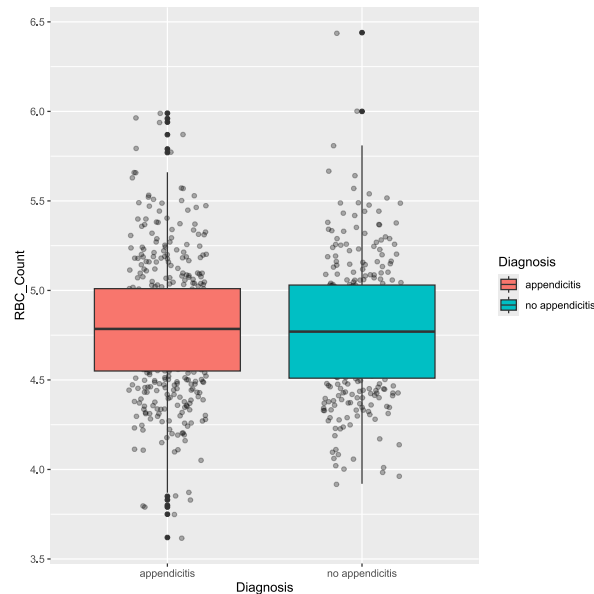
```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis) |> filter(RBC_Count < 8),  
  aes(x = Diagnosis, y = RBC_Count, fill = Diagnosis))  
g + geom_boxplot() +  
  geom_jitter(width = 0.2, alpha = 0.3)
```



ggplot2 Boxplots with Points

- Order of layers important!

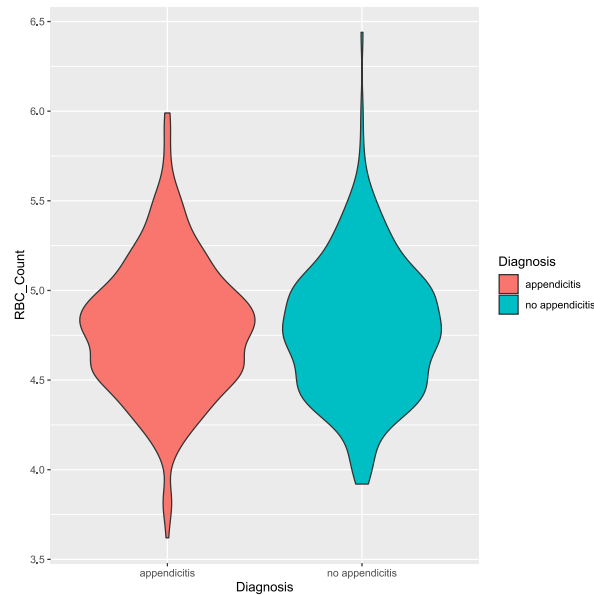
```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis) |> filter(RBC_Count < 8),  
  aes(x = Diagnosis, y = RBC_Count, fill = Diagnosis))  
g + geom_jitter(width = 0.2, alpha = 0.3) +  
  geom_boxplot()
```



ggplot2 Violin Plots

- Violin plot similar to boxplot

```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis) |> filter(RBC_Count < 8),  
            aes(x = Diagnosis, y = RBC_Count, fill = Diagnosis))  
g + geom_violin()
```



ggplot2 Scatter Plots

Two numerical variables

- **Scatter Plot** - graphs points corresponding to each observation
- Common aes values (from cheat sheet):

```
e + geom_point()
```

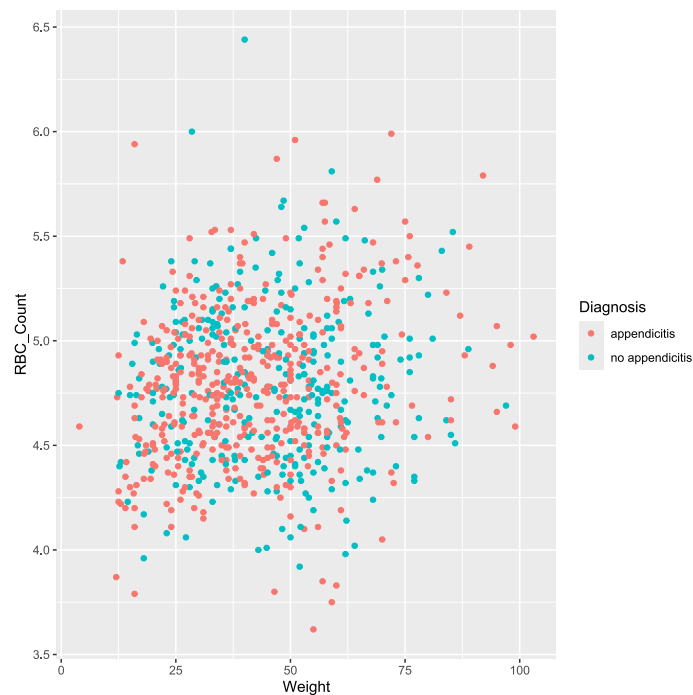
```
x, y, alpha, color, fill, shape, size, stroke
```

- Only `x =`, `y =` are really needed

ggplot2 Scatter Plots

- **Scatter Plot** - graphs points corresponding to each observation

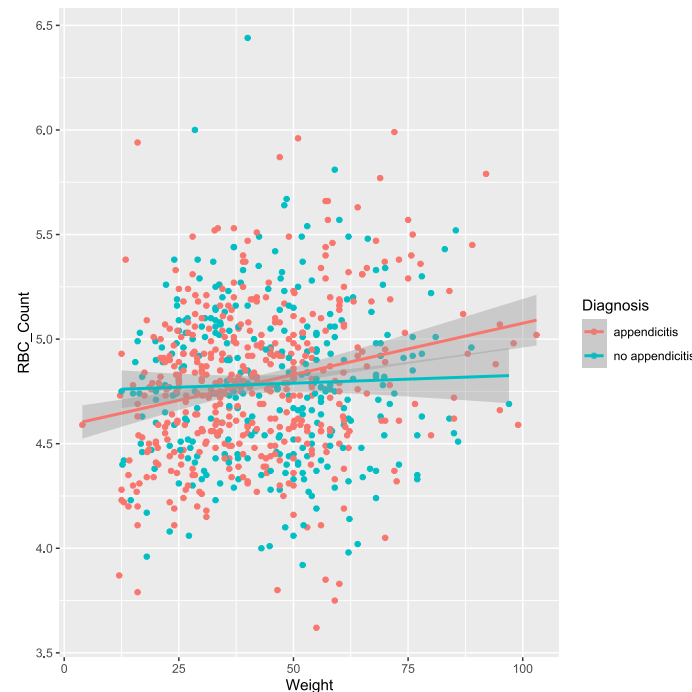
```
g <- ggplot(app_data |> drop_na(RBC_Count, Weight, Diagnosis) |> filter(RBC_Count < 8),  
            aes(x = Weight, y = RBC_Count, color = Diagnosis))  
g + geom_point()
```



ggplot2 Scatter Plots with Trend Line

- Add trend lines easily with `geom_smooth()`

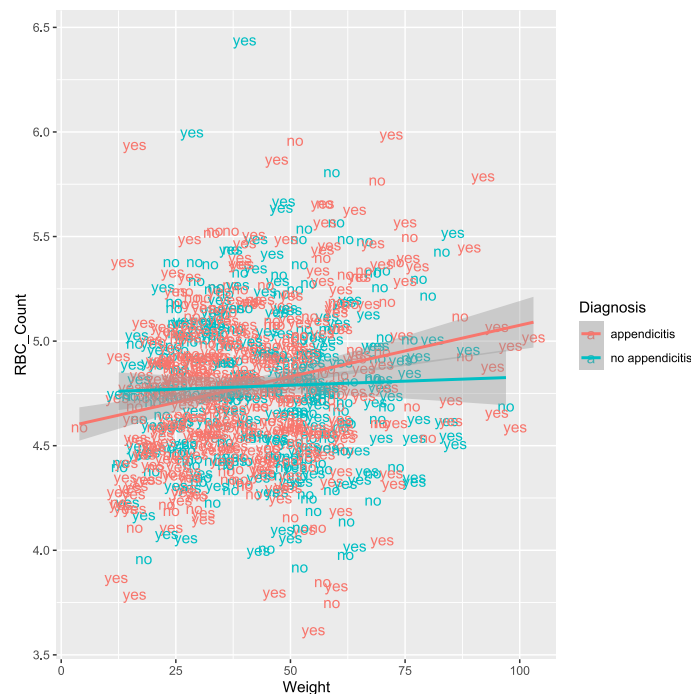
```
g <- ggplot(app_data |> drop_na(RBC_Count, Weight, Diagnosis) |> filter(RBC_Count < 8),  
  aes(x = Weight, y = RBC_Count, color = Diagnosis))  
g + geom_point() +  
  geom_smooth(method = lm)
```



ggplot2 Scatter Plots with Text Points

- Text for points with `geom_text()`

```
g <- ggplot(app_data |> drop_na(RBC_Count, Weight, Diagnosis) |> filter(RBC_Count < 8),  
  aes(x = Weight, y = RBC_Count, color = Diagnosis))  
g + geom_text(aes(label = Nausea)) +  
  geom_smooth(method = lm)
```



ggplot2 Scatter Plots with Text

- Can add a note to the plot with `geom_text()` too

```
correlation <- app_data |>
  drop_na(RBC_Count, Weight, Diagnosis) |>
  filter(RBC_Count < 8) |>
  group_by(Diagnosis) |>
  select(RBC_Count, Weight, Diagnosis) |>
  summarize(cor_vals = cor(RBC_Count, Weight, use = "complete.obs"))
correlation
```

```
## # A tibble: 2 x 2
##   Diagnosis      cor_vals
##   <chr>          <dbl>
## 1 appendicitis    0.233
## 2 no appendicitis 0.0353
```

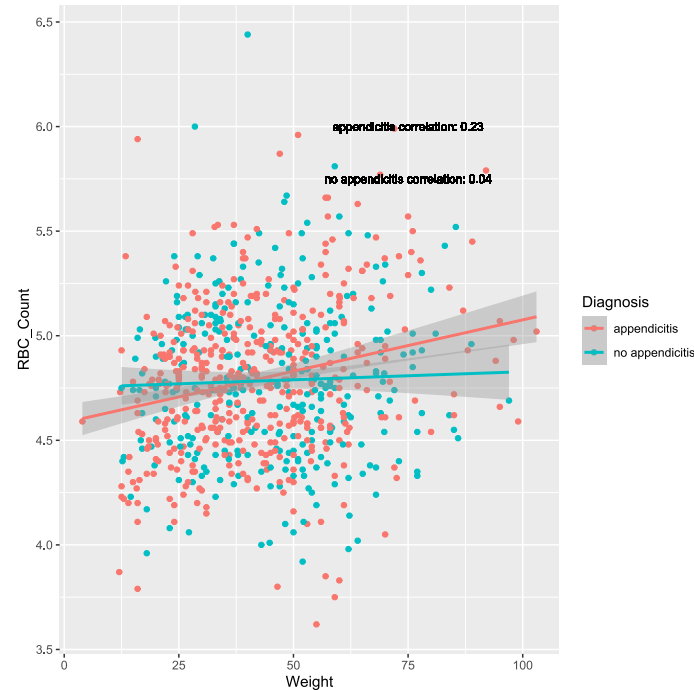
ggplot2 Scatter Plots with Text

- Can add a note to the plot with `geom_text()` too

```
g <- ggplot(app_data |> drop_na(RBC_Count, Weight, Diagnosis) |> filter(RBC_Count < 8))
g + geom_point(aes(x = Weight, y = RBC_Count, color = Diagnosis)) +
  geom_smooth(method = lm, aes(x = Weight, y = RBC_Count, color = Diagnosis)) +
  geom_text(x = 75, y = 6, size = 3,
            label = paste0(correlation[1,1], " correlation: ", round(correlation[1,2], 2))) +
  geom_text(x = 75, y = 5.75, size = 3,
            label = paste0(correlation[2,1], " correlation: ", round(correlation[2,2], 2)))
```

ggplot2 Scatter Plots with Text

- Can add a note to the plot with `geom_text()` too

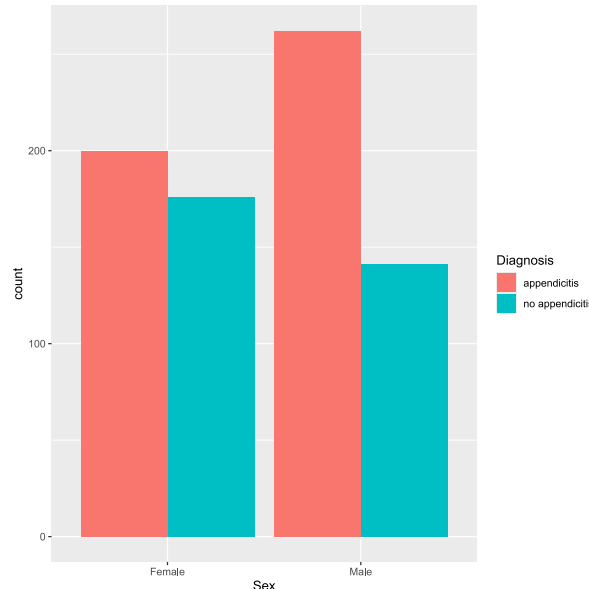


ggplot2 Faceting

Suppose we want to take one of our plots and produce similar plots across another variable!

How to create this plot across each Management category? Use **faceting**!

```
ggplot(data = app_data |> drop_na(SexF, DiagnosisF), aes(x = SexF, fill = DiagnosisF)) +  
  geom_bar(position = "dodge") +  
  labs(x = "Sex")+  
  scale_fill_discrete("Diagnosis")
```



ggplot2 Faceting

`facet_wrap(~ var)` - creates a plot for each setting of `var`

- Can specify `nrow` and `ncol` or let R figure it out

ggplot2 Faceting

`facet_wrap(~ var)` - creates a plot for each setting of `var`

- Can specify `nrow` and `ncol` or let R figure it out

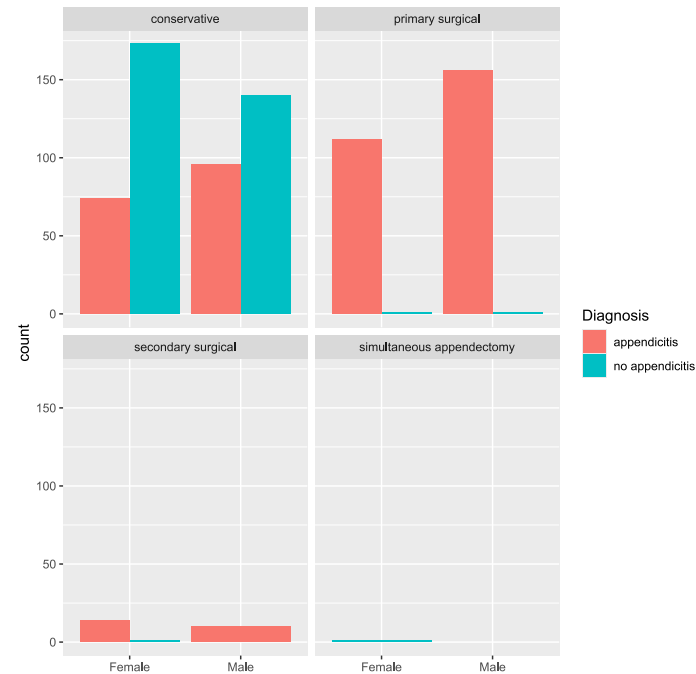
`facet_grid(var1 ~ var2)` - creates a plot for each combination of `var1` and `var2`

- `var1` values across rows
- `var2` values across columns
- Use `. ~ var2` or `var1 ~ .` to have only one row or column

ggplot2 Faceting

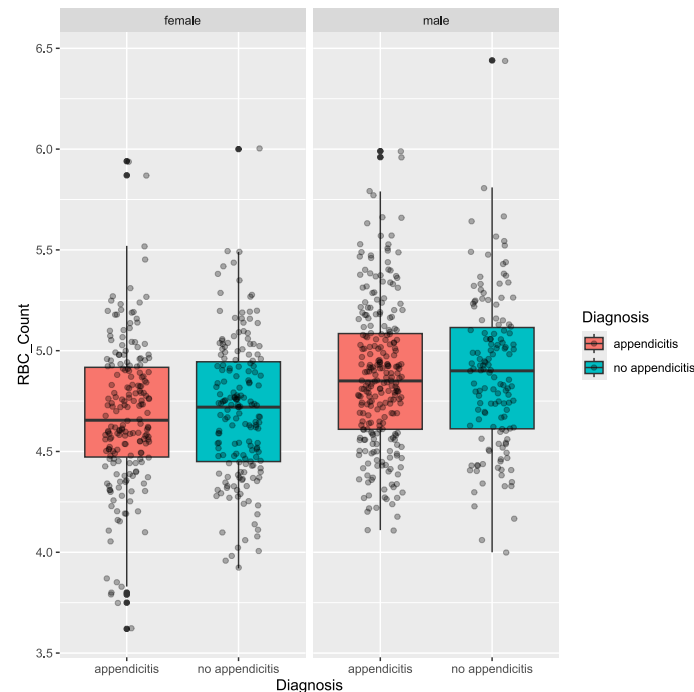
- `facet_wrap(~ var)` - creates a plot for each setting of var

```
ggplot(data = app_data |> drop_na(SexF, DiagnosisF, Management), aes(x = SexF, fill = DiagnosisF)) +  
  geom_bar(position = "dodge") +  
  labs(x = "Sex") +  
  scale_fill_discrete("Diagnosis") +  
  facet_wrap(~ Management)
```



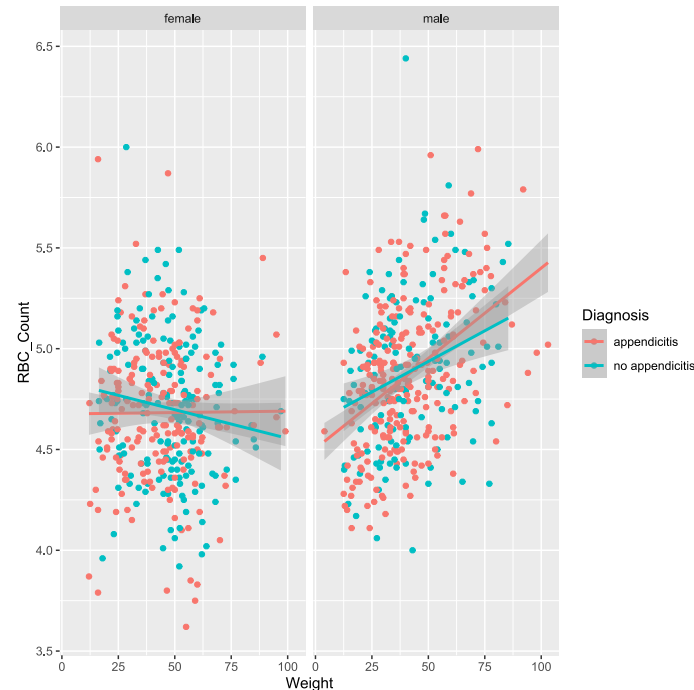
ggplot2 Faceting Can be Used with Any ggplot

```
g <- ggplot(app_data |> drop_na(RBC_Count, Diagnosis, Sex) |> filter(RBC_Count < 8),  
            aes(x = Diagnosis, y = RBC_Count, fill = Diagnosis))  
g + geom_boxplot() +  
  geom_jitter(width = 0.2, alpha = 0.3) +  
  facet_wrap(~ Sex)
```



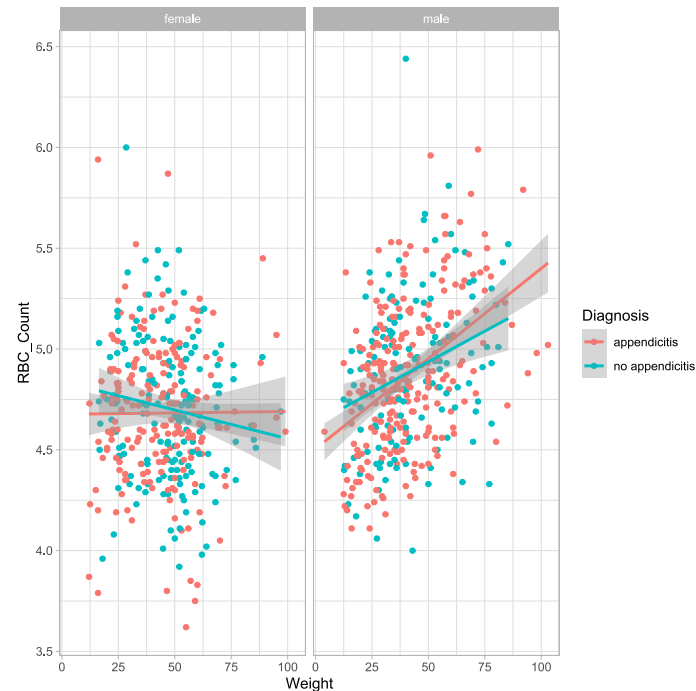
ggplot2 Faceting Can be Used with Any ggplot

```
g <- ggplot(app_data |> drop_na(RBC_Count, Weight, Diagnosis, Sex) |> filter(RBC_Count < 8),  
            aes(x = Weight, y = RBC_Count, color = Diagnosis))  
g + geom_point() +  
  geom_smooth(method = lm) +  
  facet_wrap(~ Sex)
```



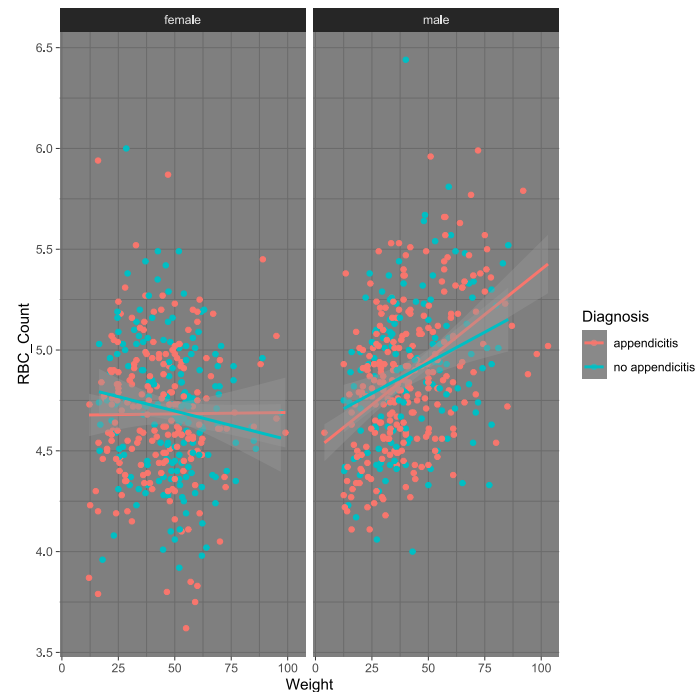
ggplot2 Themes

```
g <- ggplot(app_data |> drop_na(RBC_Count, Weight, Diagnosis, Sex) |> filter(RBC_Count < 8),  
            aes(x = Weight, y = RBC_Count, color = Diagnosis))  
g + geom_point() +  
  geom_smooth(method = lm) +  
  facet_wrap(~ Sex) +  
  theme_light()
```



ggplot2 Themes

```
g <- ggplot(app_data |> drop_na(RBC_Count, Weight, Diagnosis, Sex) |> filter(RBC_Count < 8),  
           aes(x = Weight, y = RBC_Count, color = Diagnosis))  
g + geom_point() +  
  geom_smooth(method = lm) +  
  facet_wrap(~ Sex) +  
  theme_dark()
```

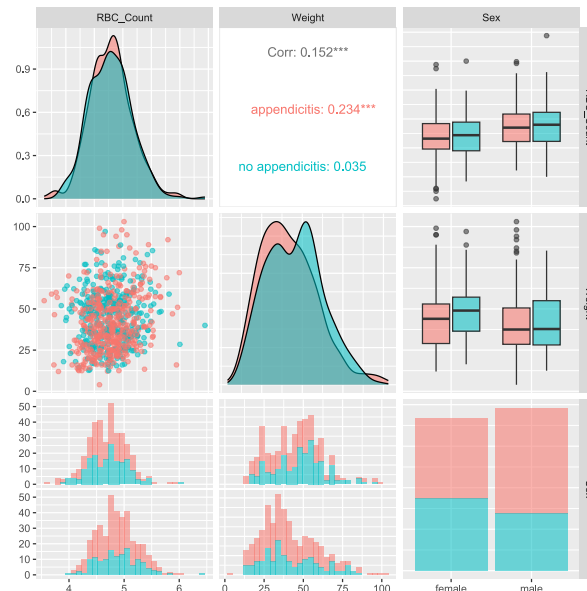


ggplot2 Extensions

Many extension packages that do nice things!

- GGally package has the `ggpairs()` function

```
library(GGally) #install if needed
ggpairs(app_data |> drop_na(RBC_Count, Weight, Diagnosis, Sex) |> filter(RBC_Count < 8),
        aes(colour = Diagnosis, alpha = 0.4), columns = c("RBC_Count", "Weight", "Sex"))
```



ggplot2 Extensions

Over 100 registered extensions at <https://exts.ggplot2.tidyverse.org/>!

- gganimate package allows for the creation of gifs

```
#install each if needed
```

```
library(gapminder)
```

```
library(gganimate)
```

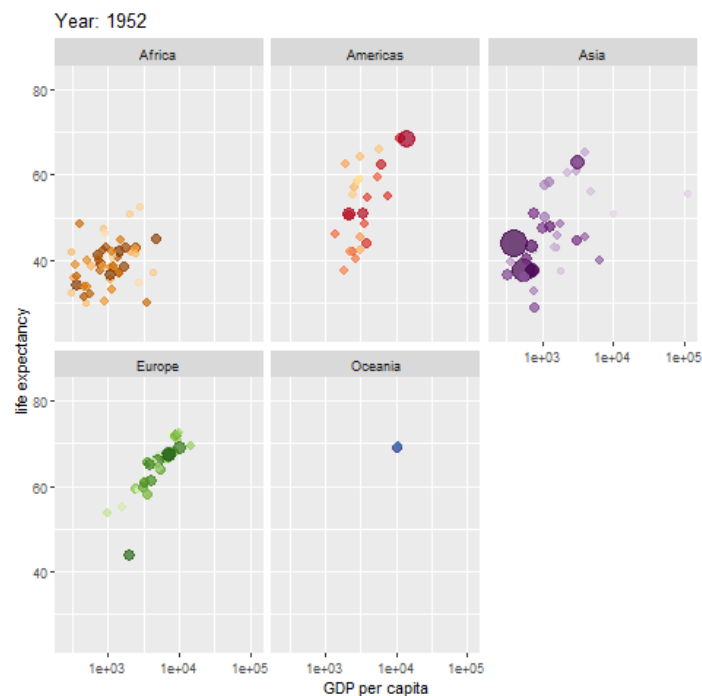
```
library(gifski)
```

```
gif <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +  
  geom_point(alpha = 0.7, show.legend = FALSE) +  
  scale_colour_manual(values = country_colors) +  
  scale_size(range = c(2, 12)) +  
  scale_x_log10() +  
  facet_wrap(~continent) +  
  # Here comes the gganimate specific bits  
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +  
  transition_time(year) +  
  ease_aes('linear')  
anim_save(filename = "img/myGif.gif", animation = gif, renderer = gifski_renderer())
```

ggplot2 Extensions

Over 100 registered extensions at <https://exts.ggplot2.tidyverse.org/>!

- gganimate package allows for the creation of gifs



Recap!

General ggplot2 things:

- Can set local or global `aes()`
 - Generally, only need `aes()` if setting a mapping value that is dependent on the data
- Modify titles/labels by adding more layers
- Use either `stat` or `geom` layer
- Faceting (multiple plots) via `facet_grid()` or `facet_wrap()`
- `esquisse` is a **great package for exploring ggplot2!**

Big Recap!

Goal: Understand types of data and their distributions

- Numerical summaries (across subgroups)
 - Contingency Tables
 - Mean/Median
 - Standard Deviation/Variance/IQR
 - Quantiles/Percentiles
- Graphical summaries (across subgroups)
 - Bar plots
 - Histograms
 - Box plots
 - Scatter plots