In [9]:
```
# import the necessary packages
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from plotnine import *
from plotnine.data import mtcars

%mathplotlib inline
```

```
UsageError: Line magic function `%mathplotlib` not found.
```

1. (4 pts) Using the Palmer Penguin data
   ("https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/penguins.csv
   make a plot that looks at whether the correlation between **body mass and flipper length** is the same between male and female penguins. Start with a default graph and change one thing (or one "class" of things, like getting rid of all gridlines) at a time, similar to how I did in the Class 6 lecture video. Again, make sure you're thinking about these concepts:
   - what visual elements can I get rid of because they distract from my message?
   - what visual elements can I add to support my message?
   - how can I make this visualizations more accessible?
2. (3 pts) In words (type this answer into a new cell and change the cell type to Markdown), explain your thought process for each step.
3. (3 pts) Recreate the graph cereal.png (in the Assignmnets folder on GH) using the cereal data set
   ("https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/cereal.csv"

In [10]:
```
### Number 1 ###
penguins = "https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/penguins.csv"
penguin = pd.read_csv(penguins)

penguin.head()
```

Out[10]:

	Unnamed: 0	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
0	0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007
1	1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007
2	2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007
3	3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007
4	4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007

load the data set into the notebook and then look at the head of the data set

In [11]:
```
penguin.columns
penguin.describe()
```

Out[11]:

	Unnamed: 0	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
count	344.000000	342.000000	342.000000	342.000000	342.000000	344.000000
mean	62.151163	43.921930	17.151170	200.915205	4201.754386	2008.029070
std	40.430199	5.459584	1.974793	14.061714	801.954536	0.818356
min	0.000000	32.100000	13.100000	172.000000	2700.000000	2007.000000
25%	28.000000	39.225000	15.600000	190.000000	3550.000000	2007.000000
50%	57.000000	44.450000	17.300000	197.000000	4050.000000	2008.000000
75%	94.250000	48.500000	18.700000	213.000000	4750.000000	2009.000000
max	151.000000	59.600000	21.500000	231.000000	6300.000000	2009.000000

In [12]:
```
penguin["body_mass_g"]
```

Out[12]:
```
0      3750.0
```

```
1      3800.0
2      3250.0
3         NaN
4      3450.0
       ...
339    4000.0
340    3400.0
341    3775.0
342    4100.0
343    3775.0
Name: body_mass_g, Length: 344, dtype: float64
```

In [13]:
```
penguin["flipper_length_mm"]
```

Out[13]:
```
0      181.0
1      186.0
2      195.0
3        NaN
4      193.0
       ...
339    207.0
340    202.0
341    193.0
342    210.0
343    198.0
Name: flipper_length_mm, Length: 344, dtype: float64
```

List out the data that I want to isolate and how much data there is

In [14]:
```
penguin.set_index("sex")
```

Out[14]:

sex	Unnamed: 0	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
male	0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	2007
female	1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	2007
female	2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	2007
NaN	3	Adelie	Torgersen	NaN	NaN	NaN	NaN	2007
female	4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	2007
...	...	...	...	...	...	...	...	...
male	63	Chinstrap	Dream	55.8	19.8	207.0	4000.0	2009
female	64	Chinstrap	Dream	43.5	18.1	202.0	3400.0	2009
male	65	Chinstrap	Dream	49.6	18.2	193.0	3775.0	2009
male	66	Chinstrap	Dream	50.8	19.0	210.0	4100.0	2009
female	67	Chinstrap	Dream	50.2	18.7	198.0	3775.0	2009

344 rows × 8 columns

In [15]:
```
penguin["sex"].isnull().sum()
```

Out[15]:
```
11
```

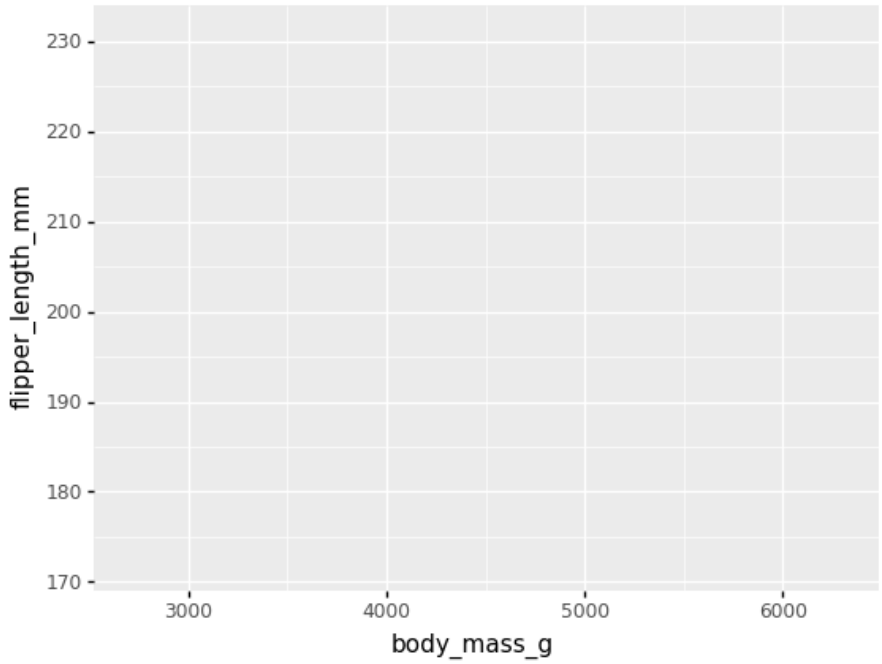
In [16]:
```
penguin = penguin[penguin["sex"].notnull()]
```

In [17]:
```
penguin["sex"].isnull().sum()
```

Out[17]:
```
0
```

Wanted to get rid of the NaN values to accurately show only male and female penguins

In [18]:
```
(ggplot(penguin, aes(x = "body_mass_g",
                     y = "flipper_length_mm",
                     color = "sex")))
```



Out[18]: <ggplot: (30267871)>

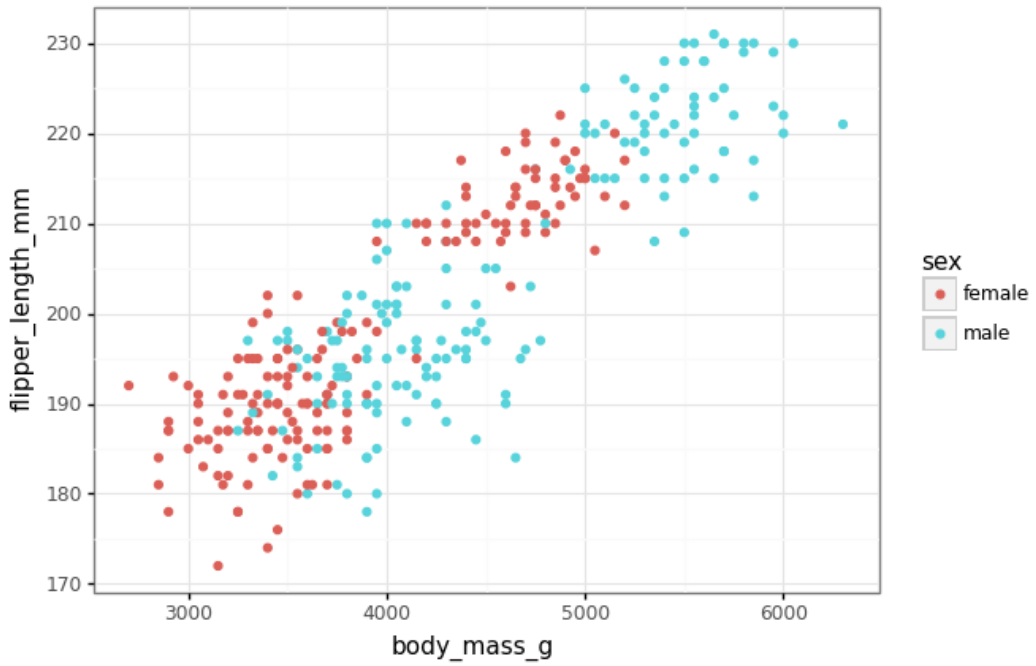
Set up a scatterplot showing body mass vs flipper length, and the correlation between

In [19]:
```
(ggplot(penguin, aes(x = "body_mass_g",
                     y = "flipper_length_mm",
                     color = "sex")) +
                     geom_point())
```

Out[19]: <ggplot: (31949704)>

Add in my data points

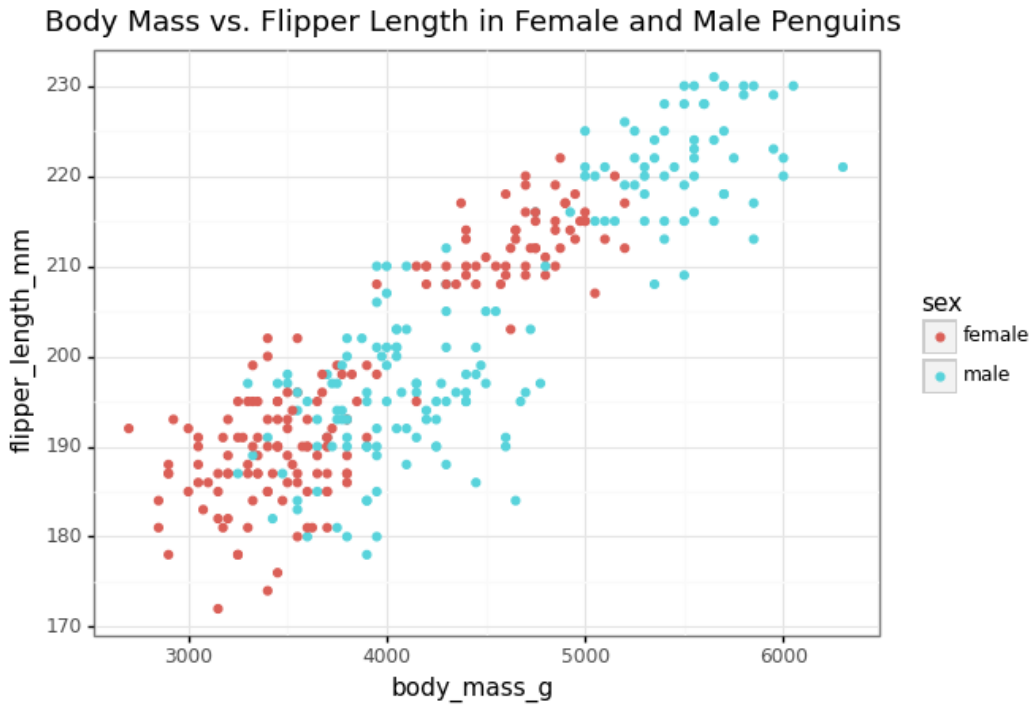
In [20]:
```
(ggplot(penguin, aes(x = "body_mass_g",
                     y = "flipper_length_mm",
                     color = "sex")) +
                     geom_point() +
                     theme_bw())
```



Out[20]: <ggplot: (30268067)>

Changed the background to black and white

In [21]:
```
(ggplot(penguin, aes(x = "body_mass_g",
                     y = "flipper_length_mm",
                     color = "sex")) +
                     geom_point() +
                     theme_bw() +
                     ggtitle("Body Mass vs. Flipper Length in Female and Male Penguins"))
```

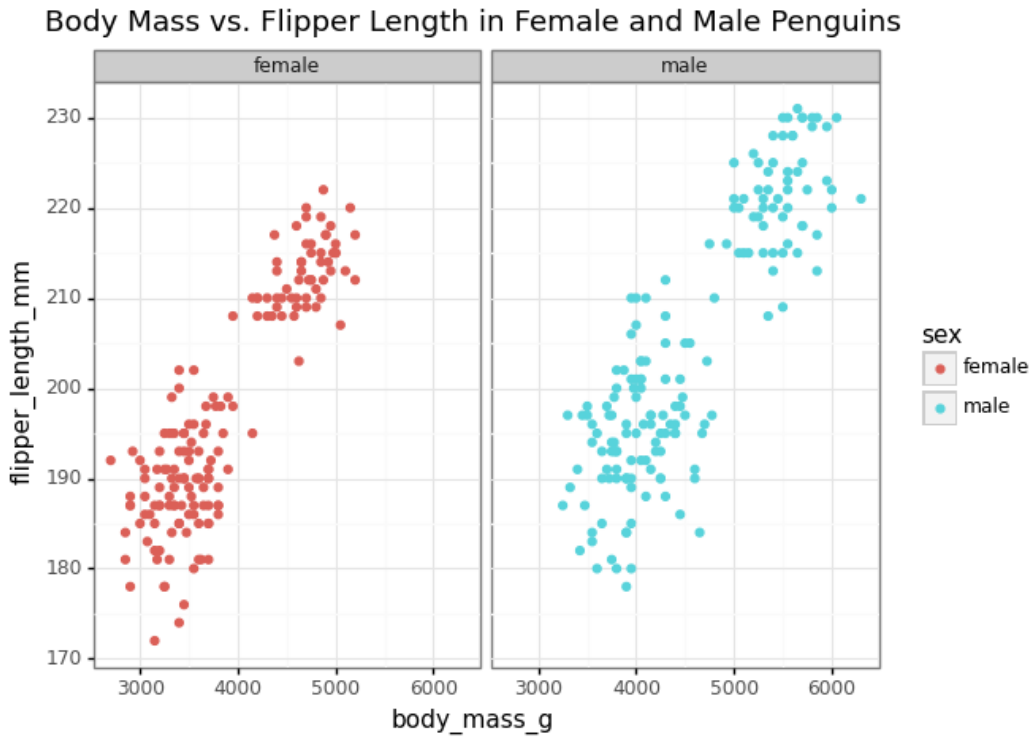


Out[21]: <ggplot: (-2114466676)>

Added a title to make the scatterplot clearer

In [22]:
```
(ggplot(penguin, aes(x = "body_mass_g",
                     y = "flipper_length_mm",
                     color = "sex")) +
                     geom_point() +
                     theme_bw() +
```

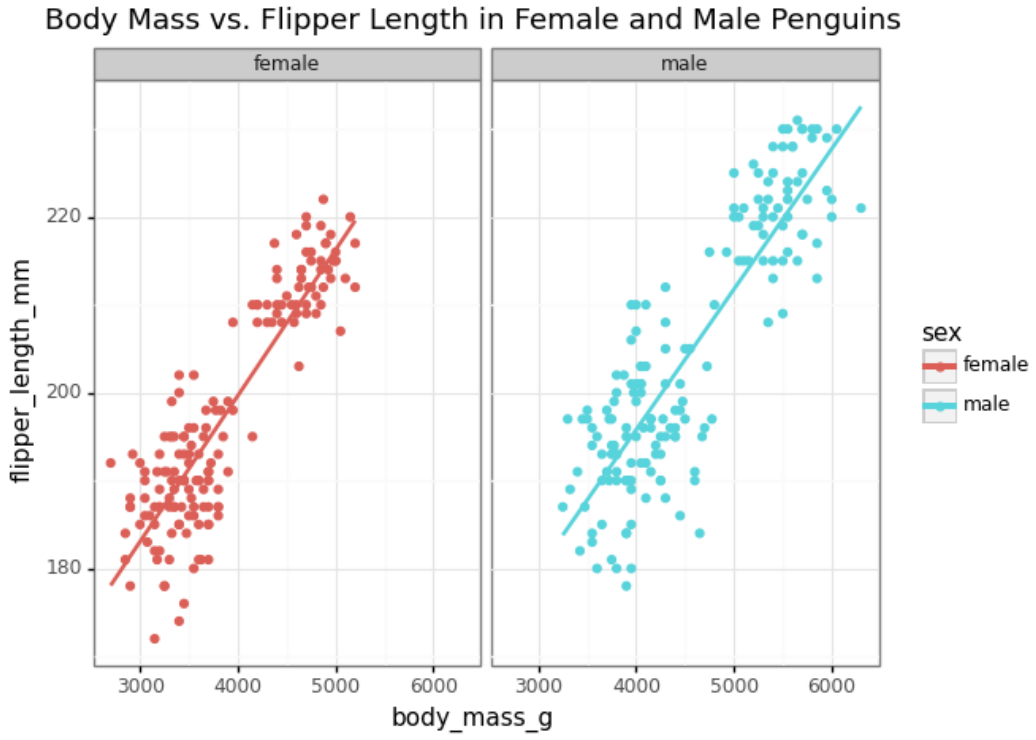
```
                ggtitle("Body Mass vs. Flipper Length in Female and Male Penguins") +
                facet_wrap("sex"))
```



Out[22]: <ggplot: (-2114462456)>

Facet_wrap to split the scatterplot up into plots based on each sex.

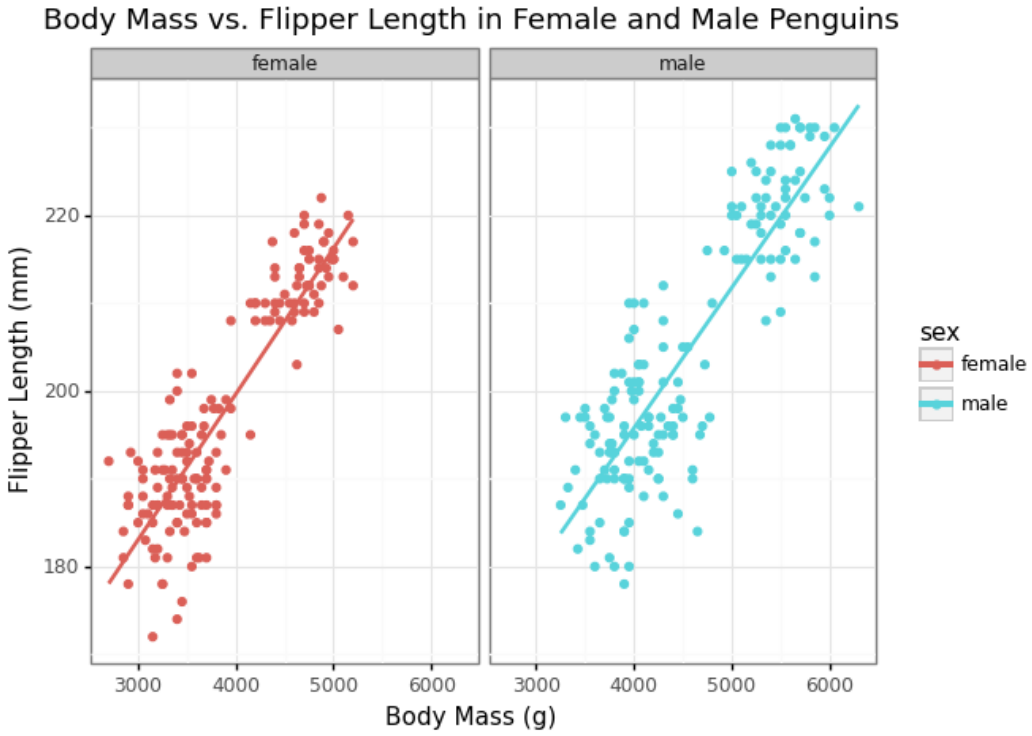
In [23]:
```
(ggplot(penguin, aes(x = "body_mass_g",
                     y = "flipper_length_mm",
                     color = "sex")) +
                     geom_point() +
                     theme_bw() +
                     ggtitle("Body Mass vs. Flipper Length in Female and Male Penguins") +
                     facet_wrap("sex") +
                     stat_smooth(method = "lm", se = False))
```



Out[23]: <ggplot: (31818015)>

Adde in a line of best fit and removed the grey around it to make the data more readable

In [24]:
```
(ggplot(penguin, aes(x = "body_mass_g",
                     y = "flipper_length_mm",
                     color = "sex")) +
                     geom_point() +
                     theme_bw() +
                     xlab('Body Mass (g)') +
                     ylab('Flipper Length (mm)') +
                     ggtitle("Body Mass vs. Flipper Length in Female and Male Penguins") +
                     facet_wrap("sex") +
                     stat_smooth(method = "lm", se = False))
```



Out[24]: <ggplot: (-2114463348)>

Renamed the x and y-axis for more clarity

In [ ]:

In [25]:
```
### Number 3: Recreate the graph cereal.png###
```

In [26]:
```
cereals = "https://raw.githubusercontent.com/cmparlettpelleriti/CPSC392ParlettPelleriti/master/Data/cereal.csv"
cereal = pd.read_csv(cereals)

cereal.head()
```

Out[26]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	ra
0	100%_Bran	N	C	70	4	1	130	10.0	5.0	6.0	280.0	25	3	1.0	0.33	68.40
1	100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0	8.0	135.0	0	3	1.0	1.00	33.98
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5.0	320.0	25	3	1.0	0.33	59.42
3	All-Bran_with_Extra_Fiber	K	C	50	4	0	140	14.0	8.0	0.0	330.0	25	3	1.0	0.50	93.70
4	Almond_Delight	R	C	110	2	2	200	1.0	14.0	8.0	NaN	25	3	1.0	0.75	34.38

In [27]:
```
cereal["mfr"].unique()
```

Out[27]:
```
array(['N', 'Q', 'K', 'R', 'G', 'P', 'A'], dtype=object)
```

Check how many unique mfr values there are

In [ ]:
```
```

In [28]:
```
cereal["protein"]
```

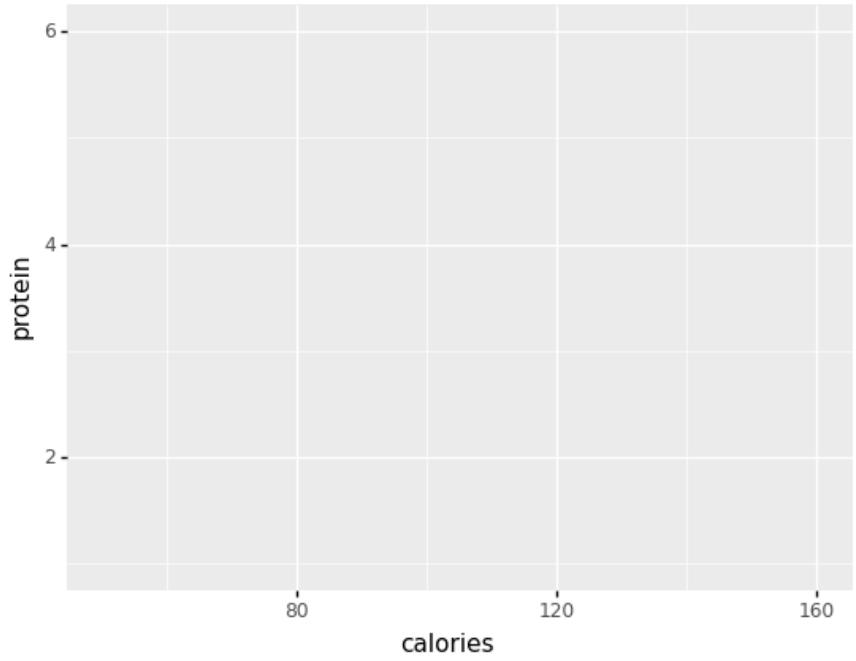
Out[28]:
```
0     4
1     3
2     4
3     4
4     2
     ..
72    2
73    1
74    3
75    3
76    2
Name: protein, Length: 77, dtype: int64
```

In [29]:
```
cereal.columns
cereal.describe()
```

Out[29]:

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight
count	77.000000	77.000000	77.000000	77.000000	77.000000	76.000000	76.000000	75.000000	77.000000	77.000000	77.000000
mean	106.883117	2.545455	1.012987	159.675325	2.151948	14.802632	7.026316	98.666667	28.246753	2.207792	1.029610
std	19.484119	1.094790	1.006473	83.832295	2.383364	3.907326	4.378656	70.410636	22.342523	0.832524	0.150477
min	50.000000	1.000000	0.000000	0.000000	0.000000	5.000000	0.000000	15.000000	0.000000	1.000000	0.500000
25%	100.000000	2.000000	0.000000	130.000000	1.000000	12.000000	3.000000	42.500000	25.000000	1.000000	1.000000
50%	110.000000	3.000000	1.000000	180.000000	2.000000	14.500000	7.000000	90.000000	25.000000	2.000000	1.000000
75%	110.000000	3.000000	2.000000	210.000000	3.000000	17.000000	11.000000	120.000000	25.000000	3.000000	1.000000
max	160.000000	6.000000	5.000000	320.000000	14.000000	23.000000	15.000000	330.000000	100.000000	3.000000	1.500000

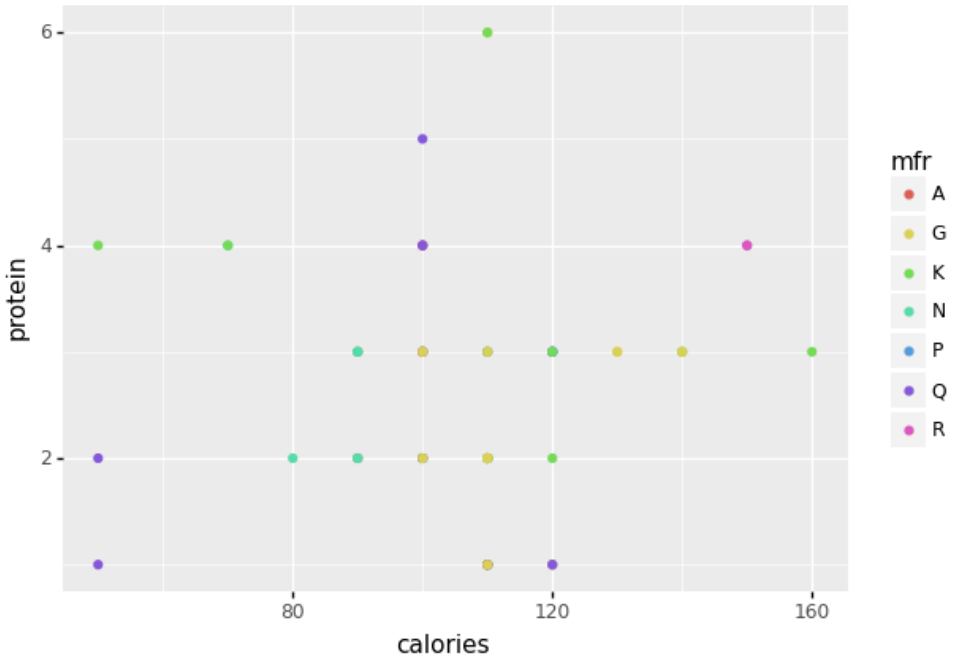
In [30]:
```
(ggplot(cereal, aes (x = "calories",
                     y = "protein",
                     color = "mfr")))
```



Out[30]:
```
<ggplot: (-2113910677)>
```

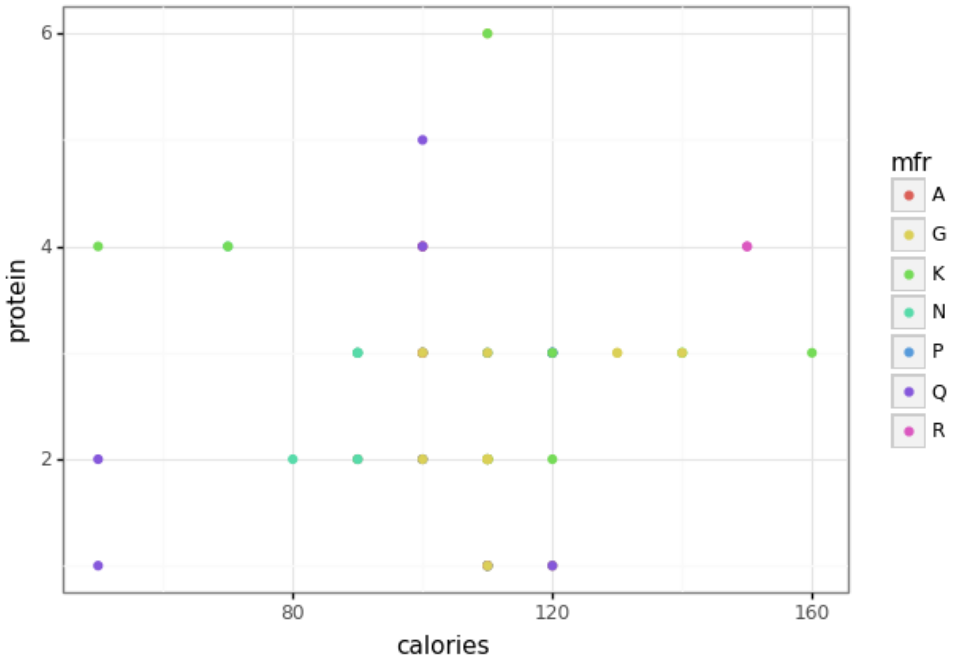
In [31]:
```
(ggplot(cereal, aes (x = "calories",
                     y = "protein",
```

```
                    color = "mfr")) +
                    geom_point())
```



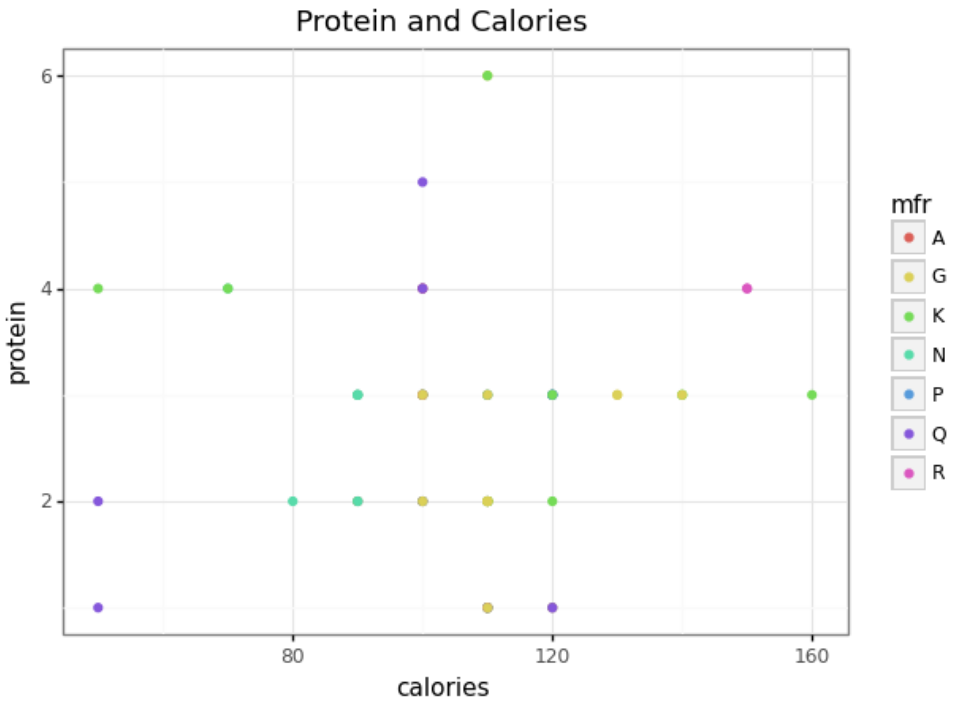
Out[31]: <ggplot: (-2114458418)>

In [32]:
```
(ggplot(cereal, aes (x = "calories",
                     y = "protein",
                     color = "mfr")) +
                     geom_point() +
                     theme_bw())
```



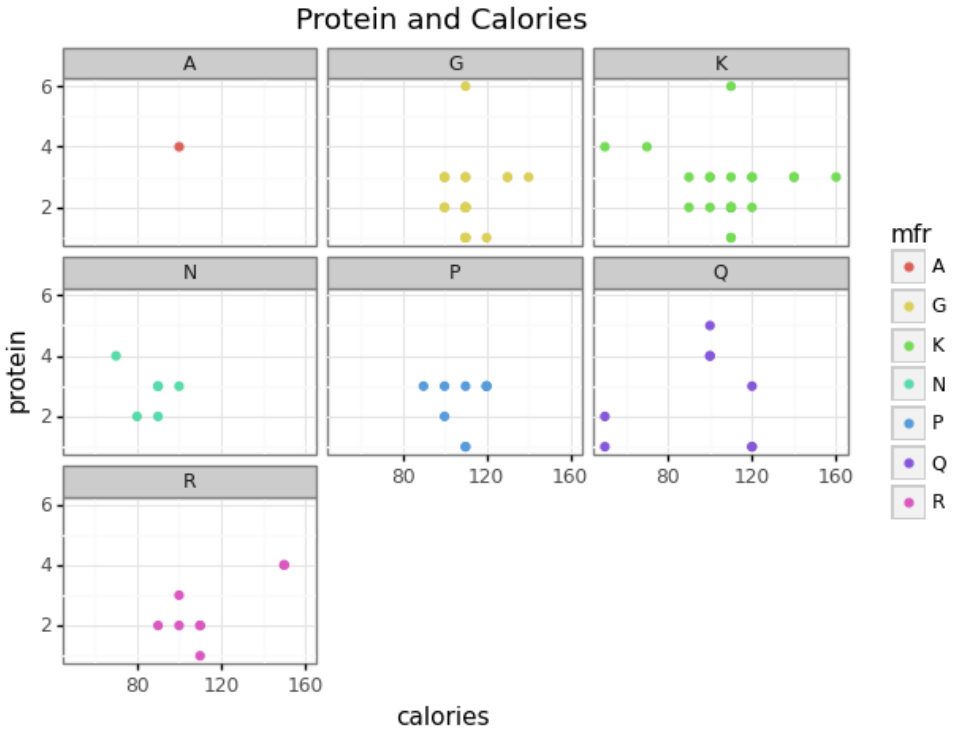
Out[32]: <ggplot: (-2114440459)>

In [33]:
```
(ggplot(cereal, aes (x = "calories",
                     y = "protein",
                     color = "mfr")) +
                     geom_point() +
                     theme_bw() +
                     ggtitle("Protein and Calories"))
```



Out[33]: <ggplot: (-2116758478)>

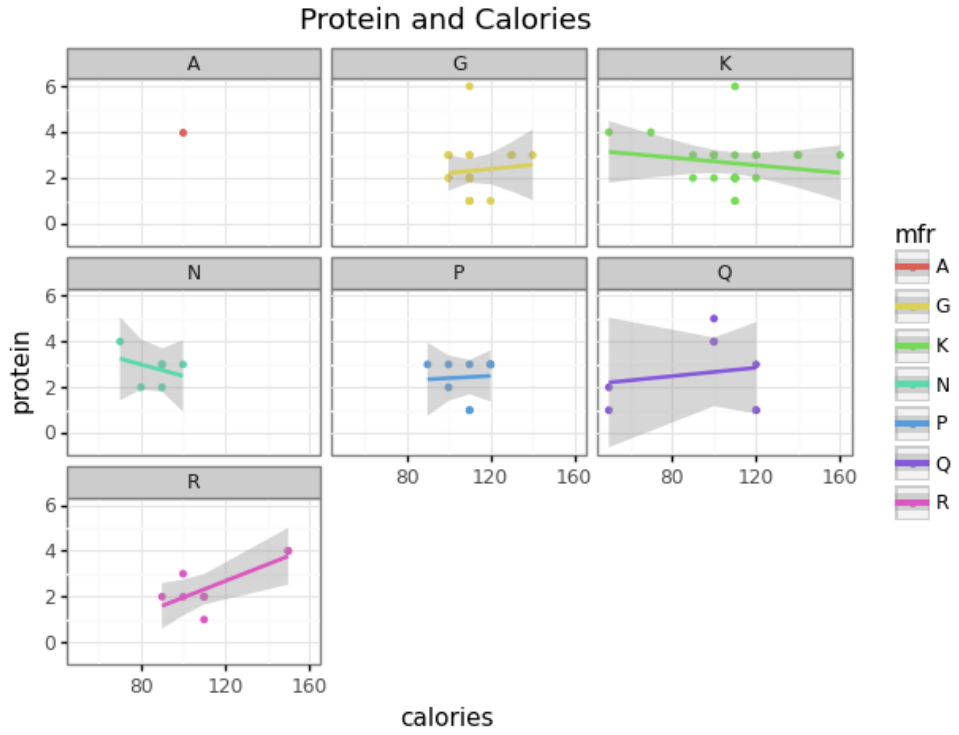
In [34]:
```
(ggplot(cereal, aes (x = "calories",
                     y = "protein",
                     color = "mfr")) +
                     geom_point() +
                     theme_bw() +
                     ggtitle("Protein and Calories") +
                     facet_wrap("mfr"))
```



Out[34]: <ggplot: (-2114444043)>

In [35]:
```
(ggplot(cereal, aes (x = "calories",
                     y = "protein",
                     color = "mfr")) +
                     geom_point(size = 1) +
                     theme_bw() +
```

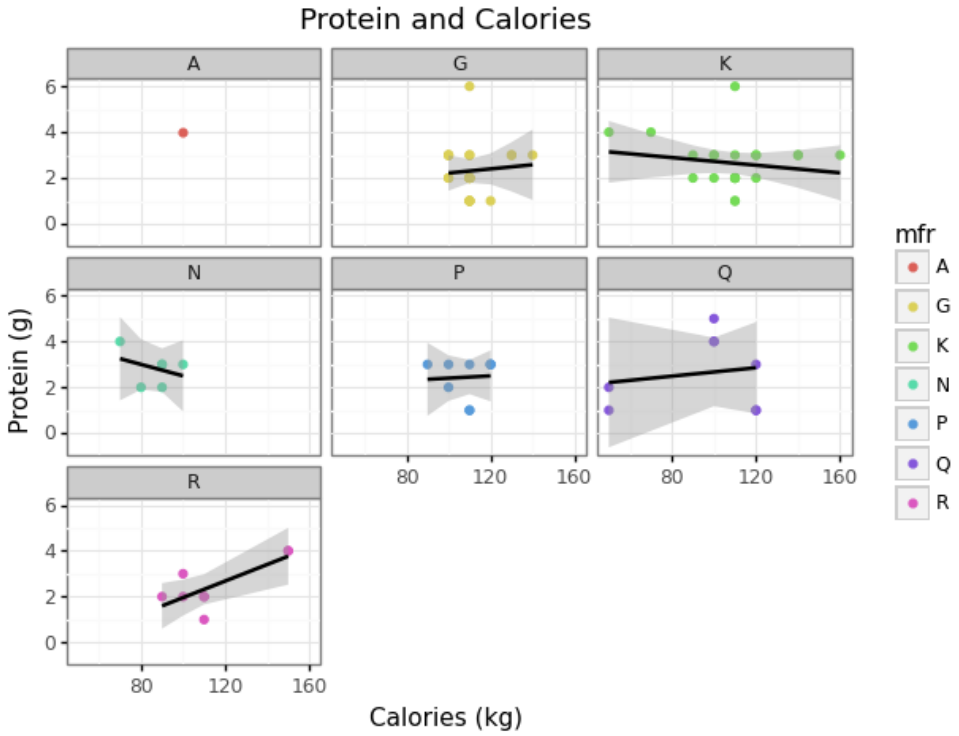
```
                    ggtitle("Protein and Calories") +
                    facet_wrap("~mfr") +
                    stat_smooth(method = "lm"))
```



Out[35]: <ggplot: (30267775)>

In [373…

```
(ggplot(cereal, aes (x = "calories",
                     y = "protein")) +
                     geom_point(aes(color = "mfr")) +
                     theme_bw() +
                     xlab('Calories (kg)') +
                     ylab('Protein (g)') +
                     ggtitle("Protein and Calories") +
                     facet_wrap("~mfr") +
                     stat_smooth(method = "lm"))
```



Out[373… <ggplot: (-2113768977)>

In [ ]: