

Lab 2

Lab Overview

The entire lab will be worth 20 points.

1. R Style and debugging (10 points)

The code and output below are designed to create a figure from the titanic dataset. Update the code - so it works- follow the style guide and comment your code.

Please include a bullet point summary of what you changed and why. Note this first code chunk has the option `eval = F` which means it is not currently evaluated.

```
THEDATA=read_csv(http://math.montana.edu/ahoegh/teaching/stat408/datasets/titanic.csv)
```

```
titanic
```

```
%>% filter(!is.na(Age)) %>% # removed passengers without age
mutate(Pclass = factor(Pclass)) %>% # changed class to factor
ggplot(y = Age, x = Pclass) %>%
  geom_boxplot(outlier.shape = NA) %>%
  geom_jitter(color = Sex) +
  theme_bw() +
  xlab( Passenger Class) +
  ggtitle('Passenger age by class and gender on Titanic') +
  facet_wrap(Sex~.)
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
```

```

v lubridate 1.9.3      v tidyr      1.3.1
v purrr      1.0.2
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become

```

```

titanic <- read_csv('http://math.montana.edu/ahoegh/teaching/stat408/datasets/titanic.csv')

```

```

Rows: 891 Columns: 12

```

```

-- Column specification -----

```

```

Delimiter: ","

```

```

chr (5): Name, Sex, Ticket, Cabin, Embarked

```

```

dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare

```

```

i Use `spec()` to retrieve the full column specification for this data.

```

```

i Specify the column types or set `show_col_types = FALSE` to quiet this message.

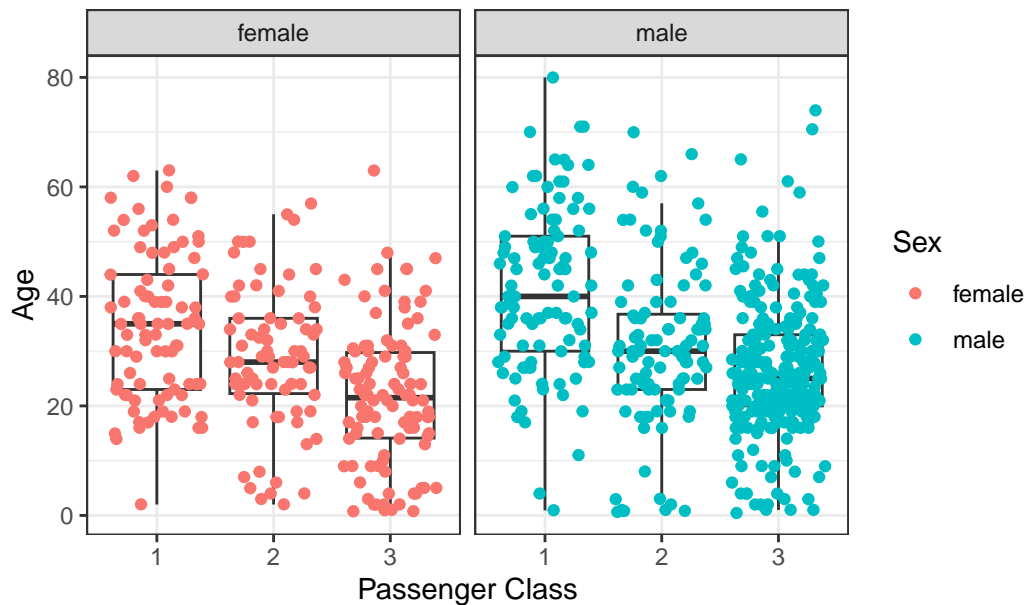
```

```

titanic %>%
  filter(!is.na(Age)) %>% # removed passengers without age
  mutate(Pclass = factor(Pclass)) %>% # changed class to factor
  ggplot(aes(y = Age, x = Pclass)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Sex)) +
  theme_bw() +
  xlab("Passenger Class") +
  ggtitle('Passenger age by class and gender on Titanic') +
  facet_wrap(Sex~.)

```

Passenger age by class and gender on Titanic



- include library statement for tidyverse
- meaningful data name: titanic
- use arrow for assignment
- spacing around operators
- piping operators on correct line
- character strings in label name

2. Data Structures / Subsetting (5 points)

Use in line R code to answer these questions with complete sentences. As an example you can call `r` variables using backticks, then `r`, then your `r` command. For example to print the current time, we can use `Sys.time()` so that it is currently 2024-09-04 09:14:32.619409.

a.

How old is the 747th passenger in the dataset?

```
titanic$Age[747]
```

```
[1] 16
```

b.

How old is the oldest passenger in the dataset?

```
max(titanic$Age, na.rm = T)
```

```
[1] 80
```

c.

Who is the oldest passenger in the dataset?

```
titanic %>% filter(Age == 80) %>% select(Name)
```

```
# A tibble: 1 x 1
  Name
  <chr>
1 Barkworth, Mr. Algernon Henry Wilson
```

d.

What percentage of 3rd class males survived (Leonardo DiCaprio as Jack)?

```
titanic %>% filter(Pclass == 3, Sex == 'male') %>% summarize(mean(Survived))
```

```
# A tibble: 1 x 1
  `mean(Survived)`
  <dbl>
1           0.135
```

e.

What percentage of 1st class females survived (Kate Winslet as Rose)?

```
titanic %>% filter(Pclass == 1, Sex == 'female') %>% summarize(mean(Survived))
```

```
# A tibble: 1 x 1
  `mean(Survived)`
  <dbl>
1           0.968
```

3. Writing data out of R (5 points)

We've seen how to use the `tibble()` function to create a dataframe. Another option is `tribble()` which allows us to specify the rows, rather than columns. See the documentation for `tribble()` <https://tibble.tidyverse.org/reference/tribble.html> and create a tibble with three columns and as many rows as you have group members. The first column contains the names of the group members, the second contains the group member's hometown, and the final contains the group member's favorite season. Print out this data frame.

```
andy_info <- tribble(
  ~name, ~hometown, ~season,
  "Andy 24", 'Bozeman, MT', 'Summer',
  "Andy 23", 'Sunshine Beach, QLD, Australia', 'Spring',
  "Andy 14", 'Blacksburg, VA', 'Fall',
  "Andy 08", 'Golden, CO', 'Winter')
```