HW 7

Due Tuesday March 7

YOUR NAME HERE

3/10/2022

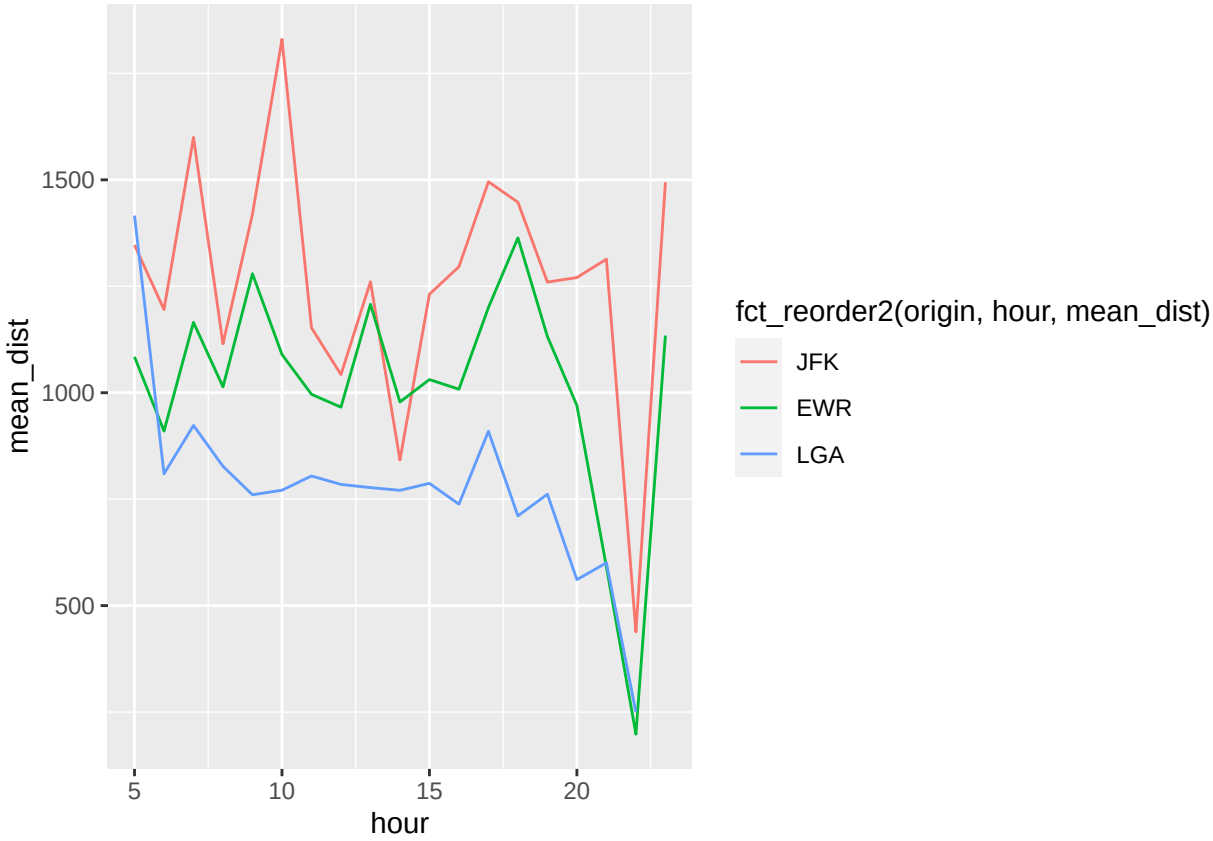
Change author to your name and save your file as HW7_YOURNAMEHERE.Rmd to your Submit folder.

For the following problems, we will use the not_cancelled dataset.

1. Create a plot with 3 lines - 1 per origin airport - where each line connects the mean distance of flights by hour. Be sure the legend reflects the order of origin airports at the latest hour (in terms of mean distance). What conclusions can you draw from the plot? (Hint: you might need two variables in your `group_by` statement, and you should consider `fct_reorder2`) You can see what your graph should look like in homework/images/hw7_hour_dist_origin.png

```
not_cancelled%>%
  group_by(origin, hour)%>%
  summarise(mean_dist = mean(distance))%>%
  ggplot()+
  geom_line(aes(x = hour, y = mean_dist, col = fct_reorder2(origin, hour, mean_dist)))
```

```
## `summarise()` has grouped output by 'origin'. You can override using the
## `.groups` argument.
```



2. This problem will investigate flights going to O'Hare airport.

a. Filter not_cancelled to include only flights going to O'Hare (dest = ORD). Create a table showing the proportion of flights which arrive more than 10 minutes late (arr_delay > 10) for each carrier. Arrange the table from largest to smallest proportion using "arrange".

```
not_cancelled|>
  filter(dest == "ORD")|>
  group_by(carrier)|>
  summarise(proportion = (sum(arr_delay > 10))/n())|>
  arrange(desc(proportion))
```

```
## # A tibble: 7 x 2
##   carrier proportion
##   <chr>        <dbl>
## 1 OO           1
## 2 EV           0.5
## 3 B6           0.351
## 4 MQ           0.347
## 5 9E           0.339
## 6 UA           0.267
## 7 AA           0.220
```

b. The two national carriers, United (UA) and American (AA) have a lower proportion delay than the regional carriers (all the others). Use fct_collapse to create two categories "National" and "Regional" and find the proportion late in each group.

```
not_cancelled|>
  filter(dest == "ORD")|>
  group_by(carrier)|>
  mutate(carrier = fct_collapse(carrier, "National" = c("UA", "AA"), "Regional" = c("OO", "EV", "B6", "
  summarise(proportion = (sum(arr_delay > 10))/n())
```

```
## Warning: Unknown levels in `f`: UA, AA, OO, EV, B6, MQ
```

```
## Warning: Unknown levels in `f`: UA, OO, EV, B6, MQ, 9E
```

```
## Warning: Unknown levels in `f`: UA, AA, OO, EV, MQ, 9E
```

```
## Warning: Unknown levels in `f`: UA, AA, OO, B6, MQ, 9E
```

```
## Warning: Unknown levels in `f`: UA, AA, OO, EV, B6, 9E
```

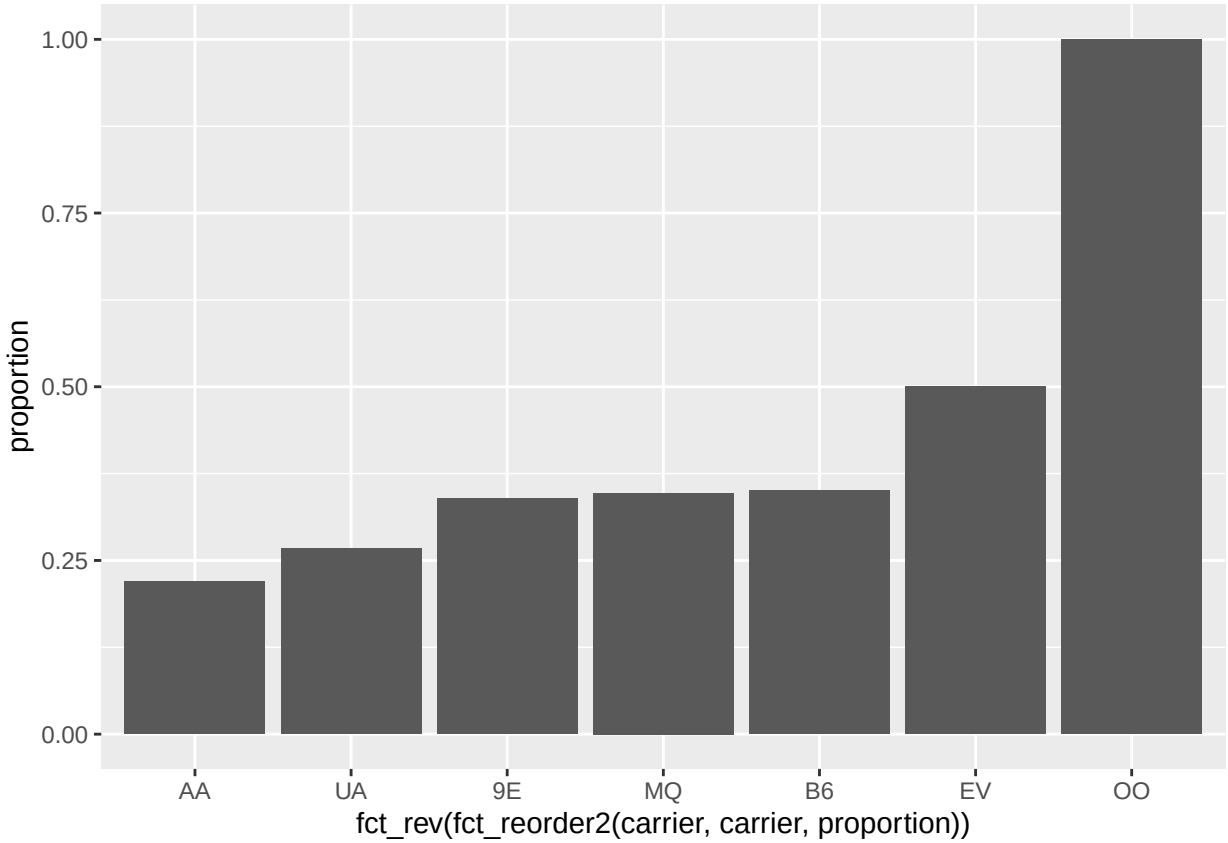
```
## Warning: Unknown levels in `f`: UA, AA, EV, B6, MQ, 9E
```

```
## Warning: Unknown levels in `f`: AA, OO, EV, B6, MQ, 9E
```

```
## # A tibble: 2 x 2
##   carrier  proportion
##   <fct>         <dbl>
## 1 Regional      0.346
## 2 National      0.245
```

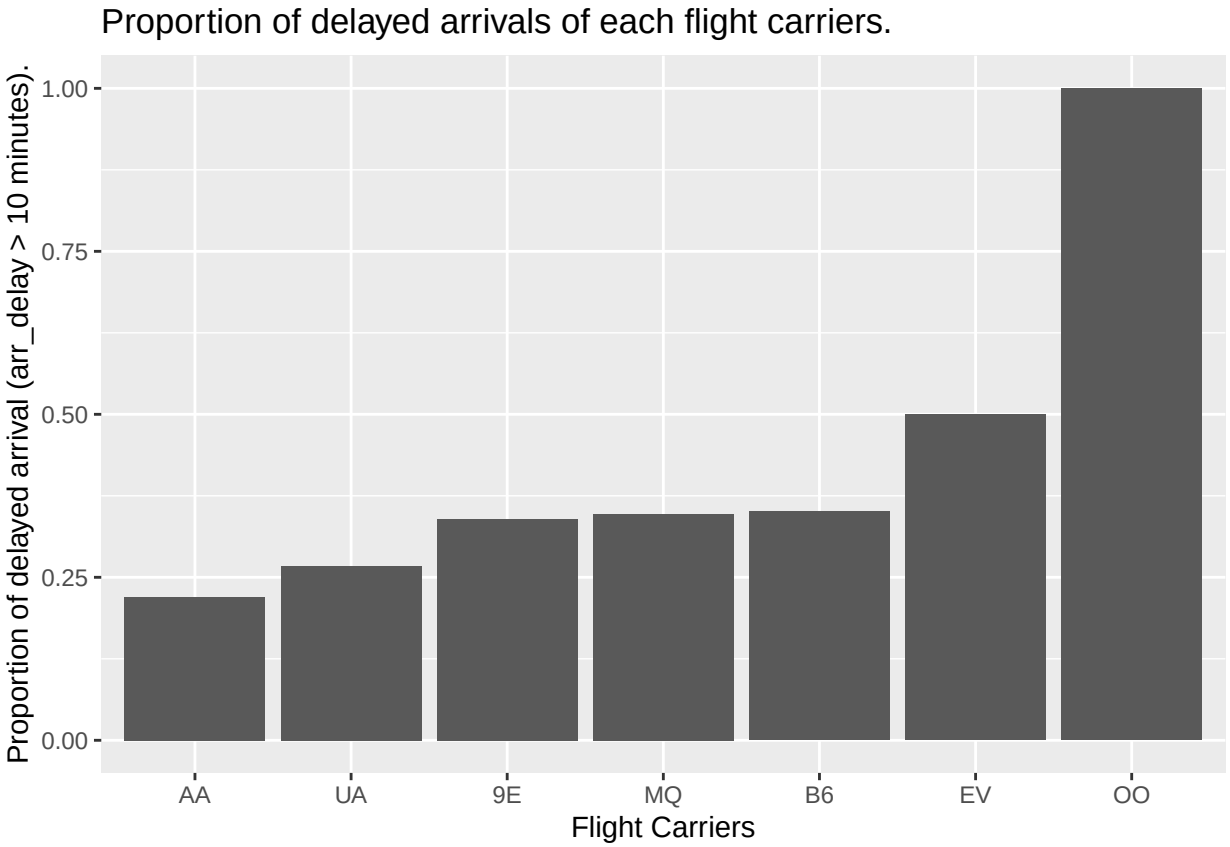
c. Rather than creating a table of the information in part a., lets make a graph using geom_col. Have one bar for each carrier, and let the y axis indicate the proportion late. See example in homework/images/hw7_ord_late_by_carrier.png.

```
not_cancelled|>
  filter(dest == "ORD")|>
  group_by(carrier)|>
  summarise(proportion = (sum(arr_delay > 10))/n())|>
  ggplot()+
  geom_col(aes(fct_rev(fct_reorder2(carrier, carrier, proportion)), proportion))
```



d. Add a title and labels to your graph above.

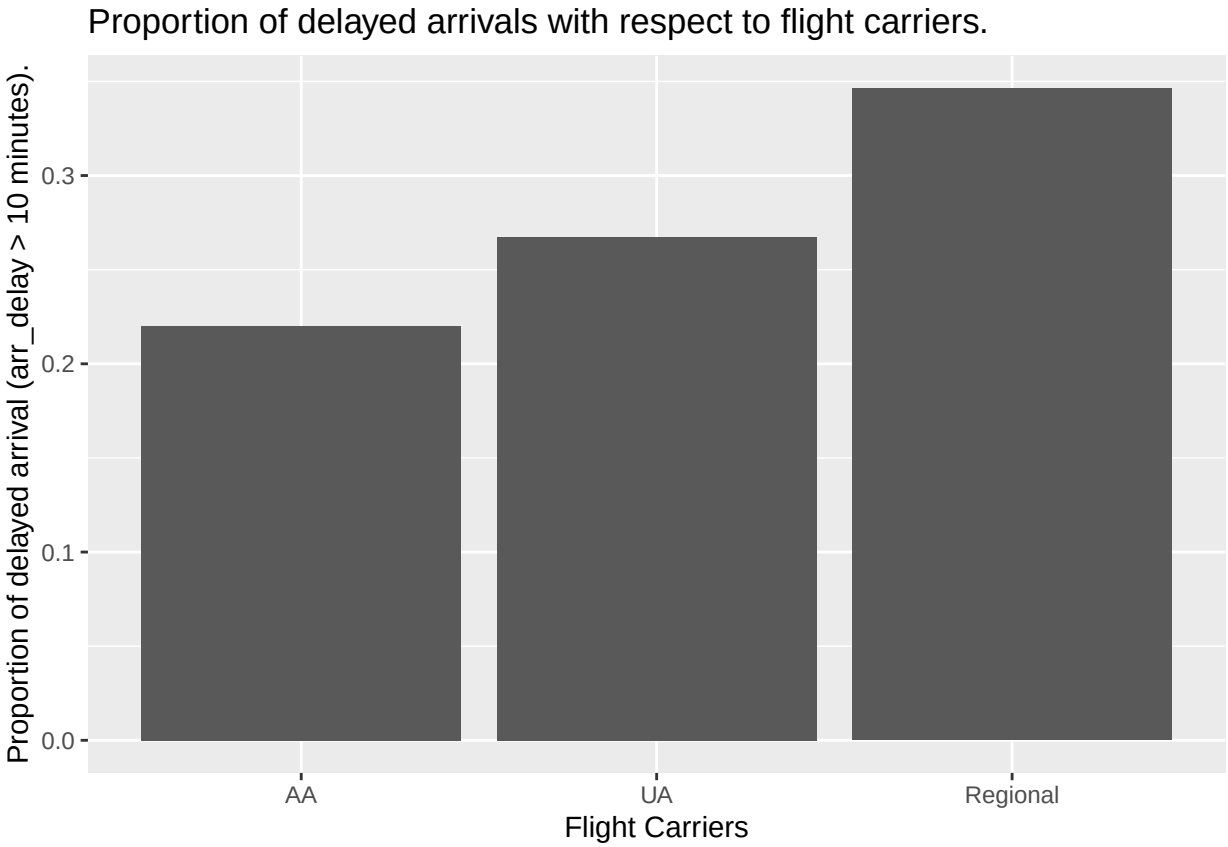
```
not_cancelled|>
  filter(dest == "ORD")|>
  group_by(carrier)|>
  summarise(proportion = (sum(arr_delay > 10))/n())|>
  ggplot()+
  geom_col(aes(fct_rev(fct_reorder2(carrier, carrier, proportion)), proportion))+
labs(x = "Flight Carriers",
       y = "Proportion of delayed arrival (arr_delay > 10 minutes).",
       title = "Proportion of delayed arrivals of each flight carriers.")
```



e. Make a graph similar to above that has categories AA, UA, and "Regional". (Hint: fct_lump, fct_recode).

```
not_cancelled|>
  filter(dest == "ORD")|>
  group_by(carrier)|>
  mutate(carrier = fct_collapse(carrier, "Regional" = c ("9E", "B6", "EV", "MQ", "OO")))|>
  summarise(proportion = (sum(arr_delay > 10))/n())|>
  ggplot()+
  geom_col(aes(fct_rev(fct_reorder2(carrier, carrier, proportion)), proportion))+
labs(x = "Flight Carriers",
       y = "Proportion of delayed arrival (arr_delay > 10 minutes).",
       title = "Proportion of delayed arrivals with respect to flight carriers.")
```

```
## Warning: Unknown levels in `f`: B6, EV, MQ, OO
## Warning: Unknown levels in `f`: 9E, B6, EV, MQ, OO
## Warning: Unknown levels in `f`: 9E, EV, MQ, OO
## Warning: Unknown levels in `f`: 9E, B6, MQ, OO
## Warning: Unknown levels in `f`: 9E, B6, EV, OO
## Warning: Unknown levels in `f`: 9E, B6, EV, MQ
## Warning: Unknown levels in `f`: 9E, B6, EV, MQ, OO
```



For these problems, we will use the gss_cat dataset

3. In this problem, we look at the gss_cat data. Examine the code below. Explain what each line does.

```
by_age <- gss_cat %>%       # 1
  filter(!is.na(age)) %>%   # 2
  count(age, marital) %>%   # 3
  group_by(age) %>%         # 4
  mutate(prop = n / sum(n)) # 5

by_age
```

```
## # A tibble: 351 x 4
## # Groups:   age [72]
##      age marital           n    prop
##    <int> <fct>         <int>   <dbl>
##  1    18 Never married    89 0.978
##  2    18 Married           2 0.0220
##  3    19 Never married   234 0.940
##  4    19 Divorced          3 0.0120
##  5    19 Widowed           1 0.00402
##  6    19 Married          11 0.0442
##  7    20 Never married   227 0.904
##  8    20 Separated         1 0.00398
##  9    20 Divorced          2 0.00797
## 10    20 Married          21 0.0837
```

```
## # ... with 341 more rows
## # i Use 'print(n = ...)' to see more rows
```

Line 1: creates a new variable called by_age that will hold the outcome of the pipe

Line 2: filters out any NA values in the age variable (column).

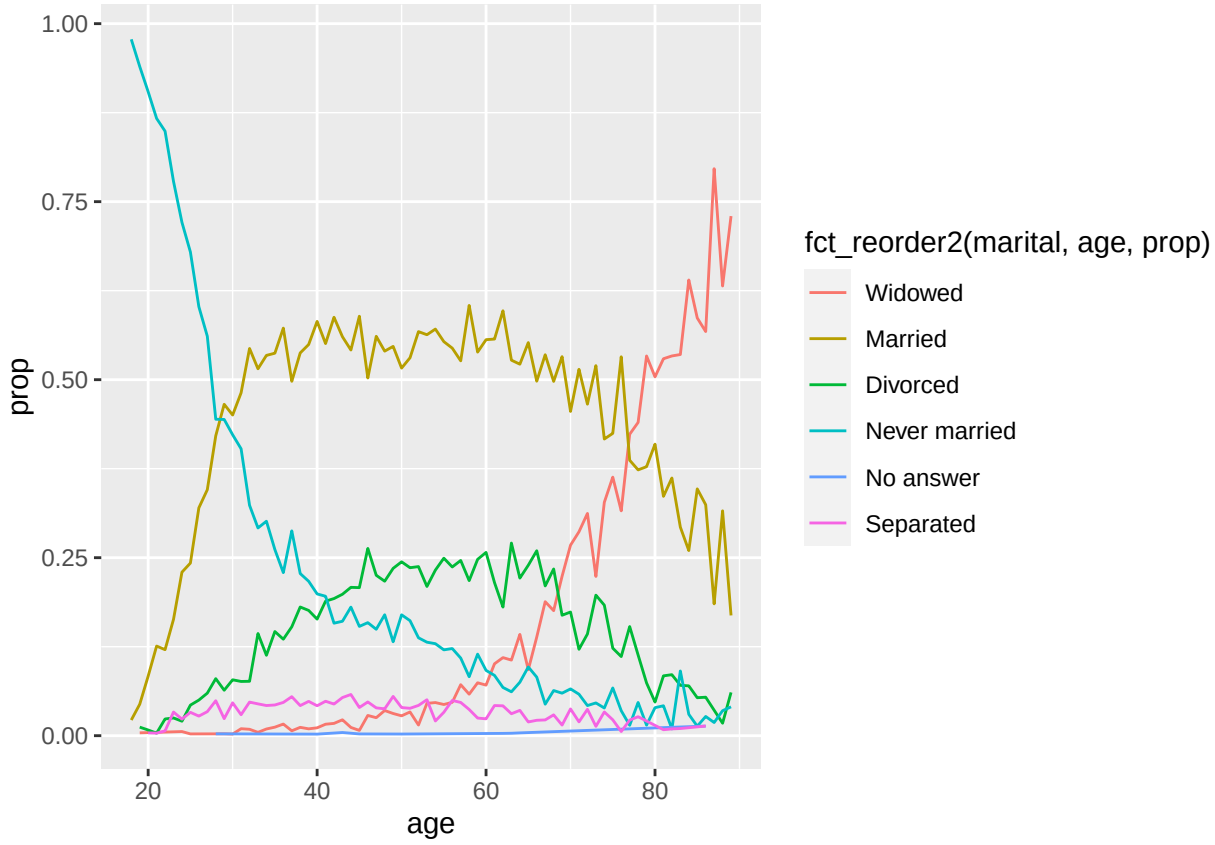
Line 3: counts the frequency of all possible combinations of age and marital status.

Line 4: groups the dataset by age.

Line 5: creates the prop variable that stores the proportion of different marital status in each age group.

4. Use fct_reorder2 to reorder the factor levels in the ggplot code below.

```
by_age %>%
  ggplot(aes(age, prop, colour = fct_reorder2(marital, age, prop))) +
  geom_line(na.rm = TRUE)
```



5. Using the gss_cat data again, create a table that includes:

- 4 columns: marital status, median hours of tv, mean hours of tv, and number of people
- Excludes people who did not answer marital status
- Combines separated and divorced into one category
- Is arranged by mean tvhours
- Be sure the number of people EXCLUDES anyone who did not answer the question of TV hours.

```
gss_cat|>
  select(marital, tvhours)|>
  drop_na()|>
  filter(marital != "No answer")|>
```

```
  mutate(marital = fct_collapse(marital, "Seperated" = c("Separated", "Divorced")))|>
  group_by(marital)|>
  summarise(med_tv = median(tvhours), mean_tv = mean(tvhours), n())|>
  arrange(desc(mean_tv))
```

```
## # A tibble: 4 x 4
##   marital       med_tv mean_tv `n()`
##   <fct>          <dbl>   <dbl> <int>
## 1 Widowed            3    3.91  1000
## 2 Seperated          2    3.17  2161
## 3 Never married      2    3.11  2995
## 4 Married            2    2.65  5172
```

6. You can copy your code from above as a start. Instead of creating a table, this time we will create a boxplot of the tvhours variable. (Hint: be sure to delete any group_by and summarize from above!). Order the marital status categories according to **mean** tvhours.

This question is very vague and does not make sense.

```
gss_cat|>
  select(marital, tvhours)|>
  drop_na()|>
  filter(marital != "No answer")|>
  mutate(marital = fct_collapse(marital, "Seperated" = c("Separated", "Divorced")))|>
  mutate(marital = fct_reorder(marital, tvhours, .fun = mean))|>
  ggplot()+
  geom_boxplot(aes(marital, tvhours))
```