Lab 09: T tests and ANOVA

Your name and student ID

today's date

Run this chunk of code to load the autograder package!

Instructions

- Due date: Thursday, August 3rd at 10:00pm PST with 2 hour grace period.
- Late penalty: 50% late penalty if submitted within 24 hours of due date, no marks for assignments submitted thereafter.
- This assignment is graded on **correct completion**, all or nothing. You must pass all public tests and submit the assignment for credit.
- Submission process: Follow the submission instructions on the final page. Make sure you do not remove any `\newpage` tags or rename this file, as this will break the submission.

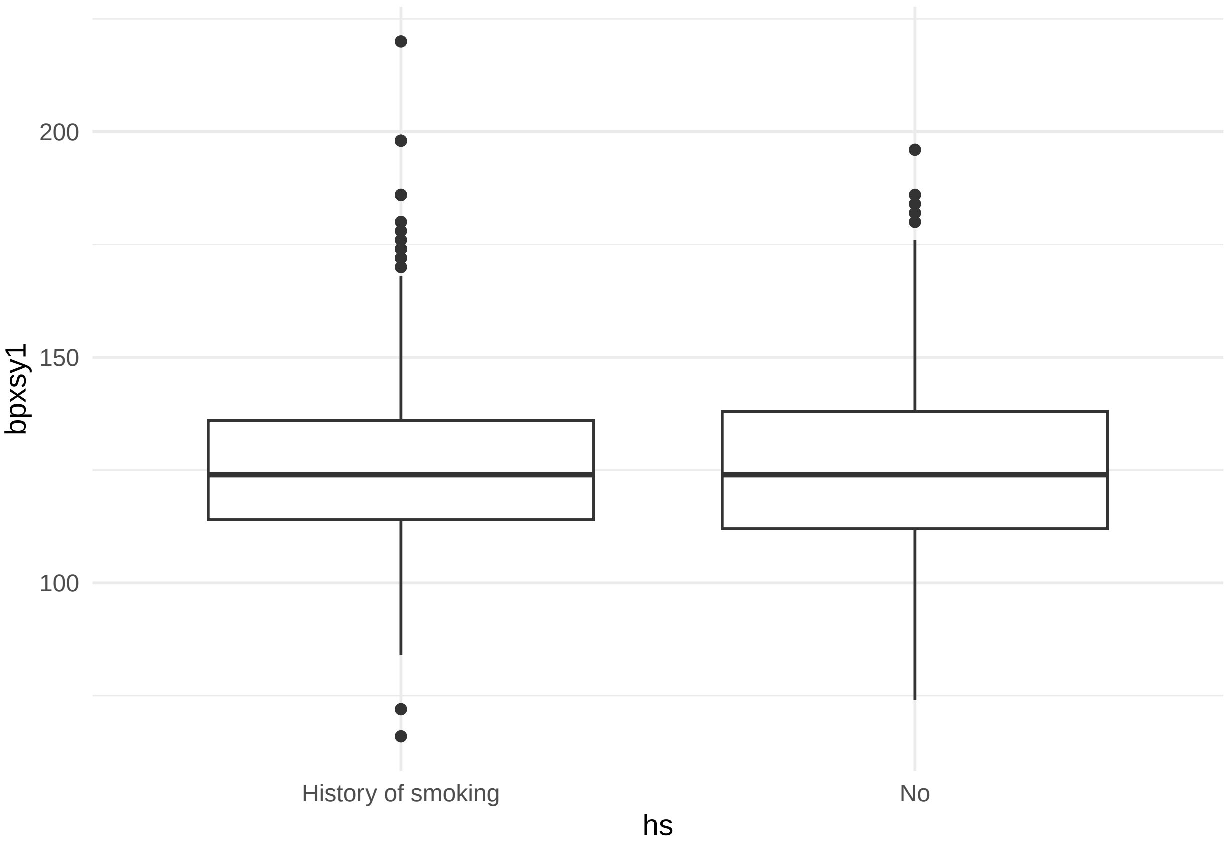
Part 1: T tests and NHANES

The NHANES is a large national survey conducted by the CDC. We will look at a reduced set of data from the NHANES for this lab.

```
##
## Attaching package: 'readr'

## The following objects are masked from 'package:testthat':
##
##     edition_get, local_edition

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:testthat':
##
##     matches

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
## Rows: 2503 Columns: 40
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (27): agegroup, gender, military, born, citizen, drinkscat, bmicat, sys1...
## dbl (13): ridageyr, drinks, bmxwt, bmxht, bmxbmi, bpxpls, bpxsy1, bpxsy2, bp...
##
## 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.
```

1. [1 point] We are interested in looking at the systolic blood pressure, bpxsy, by history of smoking, hs. Start by generating an appropriate box plot to look at these data.

```
plot1 <- ggplot(nhanes, aes(x= hs, y = bpxsy1))+ geom_boxplot()+ theme_minimal()
plot1
```



```
nhanes$bpxdi1
```

```
##    [1]   90   86   80   80   72   84   78   58   72   56   66   70   74   46   98   70   66   62
##   [19]   68   68   52   66   80   76   62   70   88   68   82   58   68   74   72   64   60   64
##   [37]   52   64   72   58   74   52   94   72   90   66   76   78   66   68   64   68   68   54
##   [55]   70   48   86   74   58   80   66   52   52   64   66   70   64   70   66   66   72   90
##   [73]   84   58   68   70   74   52   76   80   94   68   78   86   62   72   68   82   58    0
##   [91]   78   48   72   68   76   70   66   80   82   72   76   80   76   48   86   58   48   62
##  [109]   60   84   68   74   56   66   88   52   92   56   72   52   72   70   68   68   76   62
##  [127]   66   56   74   70   76   68   52   78   76   58   52   64   74   76   66   80   72   86
##  [145]   74   78   90   66   66   92   72   72   62   66   64   56   66   62   72   78   68   68
##  [163]  64  88  70  58  88  68  84  66  72  64  72  86  64  60  54  50  74  86
##  [181]  70  88  54  78  66  74  68  74  60  74  62  58  64  74  72  80  74  80
##  [199]  76  76  88  68  64  62  66  80  76  78  86  56  76  62  76  94  54  64
##  [217]  70  74  72  72  42  88  74  68  70  64  82  70  58  50  92  72  66  70
##  [235]  54  52  70  52  72  56  76  68  66  78  76  74  62  78  76  56  76  72
##  [253]  74  70  76  96  70  70  84  78  62  48  66  62  72  74  82  52  64  74
##  [271]  72  66  72  76  98  70  82  52  72  58  76  62  62  70  56  68  72  74
##  [289]  64  84  66  72  74  64  64  78  72  76  68  54  68  74  94  74  68  66
##  [307]  84  70  62  68  82  86  58  64  58  74  68  98  70  76  72  72  76  82
##  [325]  66  78  56  60  50  56  50  78  74  60  62  68  64  74  60  66  86  50
##  [343]  68  54  84  58  68  82  74  72  88  56  64  70  86  86  58  76  52  74
##  [361]  76  48  74  78  56  78  54  90  86  66  90  80  62  62  64  58  80  86
##  [379]  42  56  64  76  66  84  56  78  58  90  54  78  64  88  68  76  62  76
##  [397]  58  60  50  78  56  60  48  62  58  66  72  68  66  52  66  86  88  62
##  [415]  54  64  68  64  80  72  70  62  62  76  74  66  64  74  72  78  48  56
##  [433]  84  66  68  52  62  64  88  72  64  70  64  60  52  84  64  90  70  86
##  [451]  48  58  68  68  70  74  58  54  70  44  74  80  72  86  70  92  74  66
##  [469]  54  82  80  50  70  84  60  30  66  84  78  78  74  60  72  66  62  86
##  [487]  54  94  54 106  60  84  86  72  60  70  72  82  76  52  74  52   0  78
##  [505]  82  34  56  98   0  66  72  76  82  74  72  56  78  84  56  76  74  84
##  [523]  76  66  58  78  62  70  50  60  74  92  58  74  86  70  82  78  86  98
##  [541]  76  58  74  68  92  56  70  76  68  74  64  58  64  68  72  70  76  76
##  [559]  64 106  62  68  46  62  88  74  96  66  82  52  86  86  84  70  72  68
##  [577]  78  72  82  74  60  78  70  80  54  68  78  44  54  64 112  76  76  66
##  [595]  74  88  70   0  70  82  70  80  76  74  64  58  78  72  72  78  78  60
##  [613]  58  74  64  88  58  74  76  74  68  72  76  76  78   0  64  76  72  66
##  [631]  94  80  66  74  50  72  48  84  74  72  62  74  88  58  62  42  56  72
##  [649]  56  76  86  66  48  74  36  72  64   0  56  80  72   0  64  70  62  50
##  [667]  92  76  80  70  80  84  56  52  76  66  64  70  68  68  60  64  62  72
##  [685]  64  64  68  64  70 104  68  78  72  66  90  56  82  72 122  70  62  76
##  [703]  66  72  66 100  68  58  72  78  66  56  68  58  78  72  64  76  58  72
##  [721]  86  80  74  50  74  76  76  60  82  78  72  70  74  80   0  70  82  80
##  [739]  50  72  84  76  74  74  54  84  68  78  54  72  40  76  76  80  74  90
##  [757]  56  80  58  68  82  44  70  70  70  68  82  72  98  76  64  54  82  70
##  [775]  76  68  74  76  66  62  80  76  82  60  72  58  66  84  76  72  72  48
##  [793]  70  82  82  70  58  60  66  84  68  80  58  76  70  78  64  66  66  72
##  [811]  44  70  64  66  72  68  72  62  68  96  58  66  62  62  78  68  28  66
##  [829]  68  70  82  70  60  56  80  74  74  50  66  58  80  74  94  62  84  74
##  [847]  58  68  74  72  76  68  48  72  76  80  72  64  86  72  60  52  76  66
##  [865]  70  56  60  54  52  94  96  46  56  70  78  76  90  62  74  82  76  60
##  [883]  74  44  78  66  60 100  66  80  60  62  60  64  78  86  44  62  66  58
##  [901]  80  74  74  78  58  78  58  80  72  76  56  84  72  78  60  64  66  82
##  [919]  66  78  88  48  58  82  44  74  86  78  72  68  80  72  80  78  88  68
##  [937]  68  82  70  68  62  72  72  66  84  68  74  80  86  86  80  70  74  84
##  [955]  38  80  76  84  58  64  72  78  26  72  80  74  78  56  74  82  68  78
##  [973]  72  64  68  64  78  78  74  64  72  78  64  66  90  74  70  80  70  66
##  [991]  56  68  62  56  74  84  66  56  64  58  76  68  58  72  82  66  74  58
## [1009]  62  72  74  78 100  78  62  40  72  66  60  82  66  84  86  84  74  56
## [1027]  78  82  72  64  72  62  64  70  68  54  78  88  60  74  52  56   0  74
## [1045]  80  52  34  76  62  72  56  98  74  60  84  68  78  68  68  68   0  74
## [1063] 100  70  72  56  76  88  78  72  52  82  62  70  60  60  62  76  74  66
## [1081]  64  76  68  70  70  58  52  80  60  70  50   0  74  58  54  74  74  84
## [1099]  78  84  60  82  64  76  74  68  72  82  72  78  76  74  54  78  82  66
## [1117]  90  94  68  68  66  58  58  88  46  70  82  66  68  64  56  82  70  72
## [1135]  62  88  78  86  66  52  66  72  90  66  72  82  74  70  76  70  72  84
## [1153]  70  48  60  88  92  68  64  76  86  64  74  80  78  70  82  58  76  46
## [1171]  86  82  66  78  78  60  86  70
```

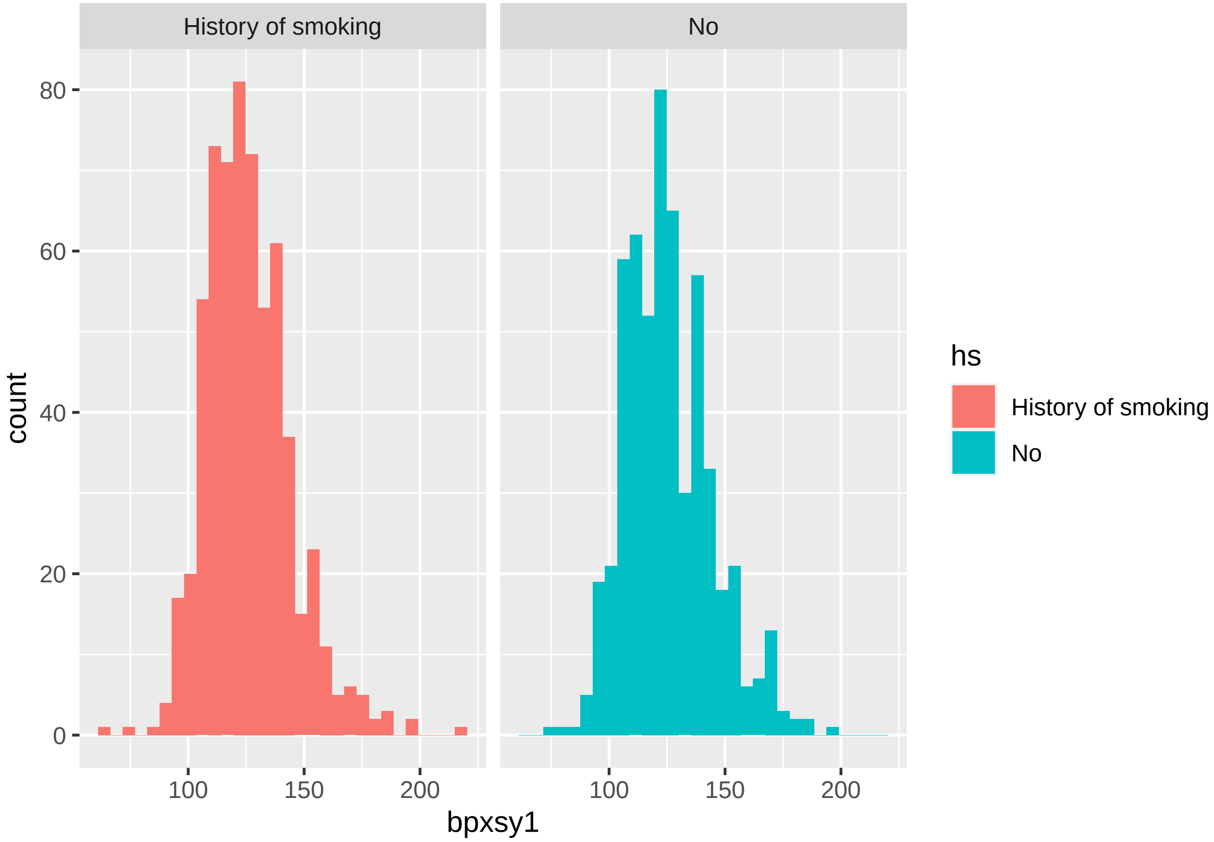
```
. = ottr::check("tests/p1.R")
```

```
##
## All tests passed!
```

2. [1 point] **Now generate a set of faceted histograms that show the same data.**

```
plot2 <- ggplot(nhanes, aes(x = bpxsy1)) + geom_histogram(aes(fill = hs)) +
         facet_wrap(~hs)
plot2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
. = ottr::check("tests/p2.R")
```

```
##
## All tests passed!
```

3. [1 point] **Summarize the means and standard deviations of the systolic blood pressurea for each category of `hs`. Assign `p3` to a dataframe with the mean systolic blood pressures assigned to `mean_bp` and the standard deviations assigned to `sd_bp`.**

```
p3 <- nhanes %>% group_by(hs)%>% summarize (mean_bp = mean(bpxsy1),
                                            sd_bp = sd(bpxsy1) )
p3
```

```
## # A tibble: 2 x 3
##   hs                 mean_bp sd_bp
##   <chr>                <dbl> <dbl>
## 1 History of smoking    126.  18.6
## 2 No                    126.  18.7
```

```
. = ottr::check("tests/p3.R")
```

```
## 
## All tests passed!
```

4. Do we meet the all of the assumptions to run a two-sample t-test? Why or why not?

1. The observation are independent. 2, Mean (group) is normally distributed.
2. The sample(group) variance needs to be equal.

5. State the null and alternative hypotheses in the context of this question.

Ho : The mean SBP of smokers is equal to the mean SBP of non-smokers. Ha : The mean SBP of smokers is not equal to the mean SBP of non-smokers.

6. [1 point] Use an R function to test if the variability gives enough evidence to reject the null hypothesis of no difference between mean blood pressure by smoking history.

```r
p6 <- t.test(bpxsy1 ~ hs, data = nhanes)
p6
```

```
##
##  Welch Two Sample t-test
##
## data:  bpxsy1 by hs
## t = 0.23094, df = 1161.9, p-value = 0.8174
## alternative hypothesis: true difference in means between group History of smoking and group No is no
## 95 percent confidence interval:
##  -1.883164  2.385630
## sample estimates:
## mean in group History of smoking                  mean in group No
##                         126.1260                          125.8748
```

```r
. = ottr::check("tests/p6.R")
```

```
##
## All tests passed!
```

7. Use these results to interpret your p-value in the context of this question. Do you reject or fail to reject the null hypothesis?

Under the null hypothesis, we have 81.74% of chance of seeing a difference between our two sample is 0.2512. There we would fail to reject the null hypothesis and not conclude that there is a significance difference between the SBP of smokers vs non-smokers.

Repeat your analysis above without using the `t.test()` function.

8. [1 point] First calculate the test statistic by hand. Do not round and assign this value to `t_stat`.

```
# this code gives you the number of smokers in the dataset
n_s <- nrow(nhanes %>% filter(hs == 'History of smoking'))
n_s
```

```
## [1] 619
```

```
# this code gives you the number of non-smokers in the dataset
n_ns <- nrow(nhanes %>% filter(hs == 'No'))
n_ns
```

```
## [1] 559
```

```
# calculate your test statistic. You can make more objects if you wish.
t_stat <- 0.2512/ sqrt((18.56617^2 / n_s) + (18.71515^2/ n_ns))
t_stat
```

```
## [1] 0.2309112
```

```
. = ottr::check("tests/p8.R")
```

```
##
## All tests passed!
```

9. [1 point] Now compare your test statistic to a t-distribution with df = 558 and calculate the p-value. This is an approximation using the smaller of the two sample sizes - 1.

```
p_value <- pt(0.2309112, df = 558, lower.tail = FALSE) *2
p_value
```

```
## [1] 0.8174684
```

```
. = ottr::check("tests/p9.R")
```

```
##
## All tests passed!
```

10. [1 point] **Finally, construct a 99% confidence interval for these data. Interpret the interval in the context of this question and decide whether or not to reject the null hypothesis.**

```
CV <- qt(0.005, df = 558, lower.tail = FALSE)

SE <- sqrt((18.56617^2 / n_s) + (18.71515^2/ n_ns))

lowerbound <- 0.2512 - CV * SE
upperbound <- 0.2512 + CV * SE
conf_int <- c(lowerbound, upperbound)
conf_int
```

```
## [1] -2.560568  3.062968
```

Our 99% conifdence interval for mean difference of SBP in smoker and non-smokers is (-2.560568 3.062968)

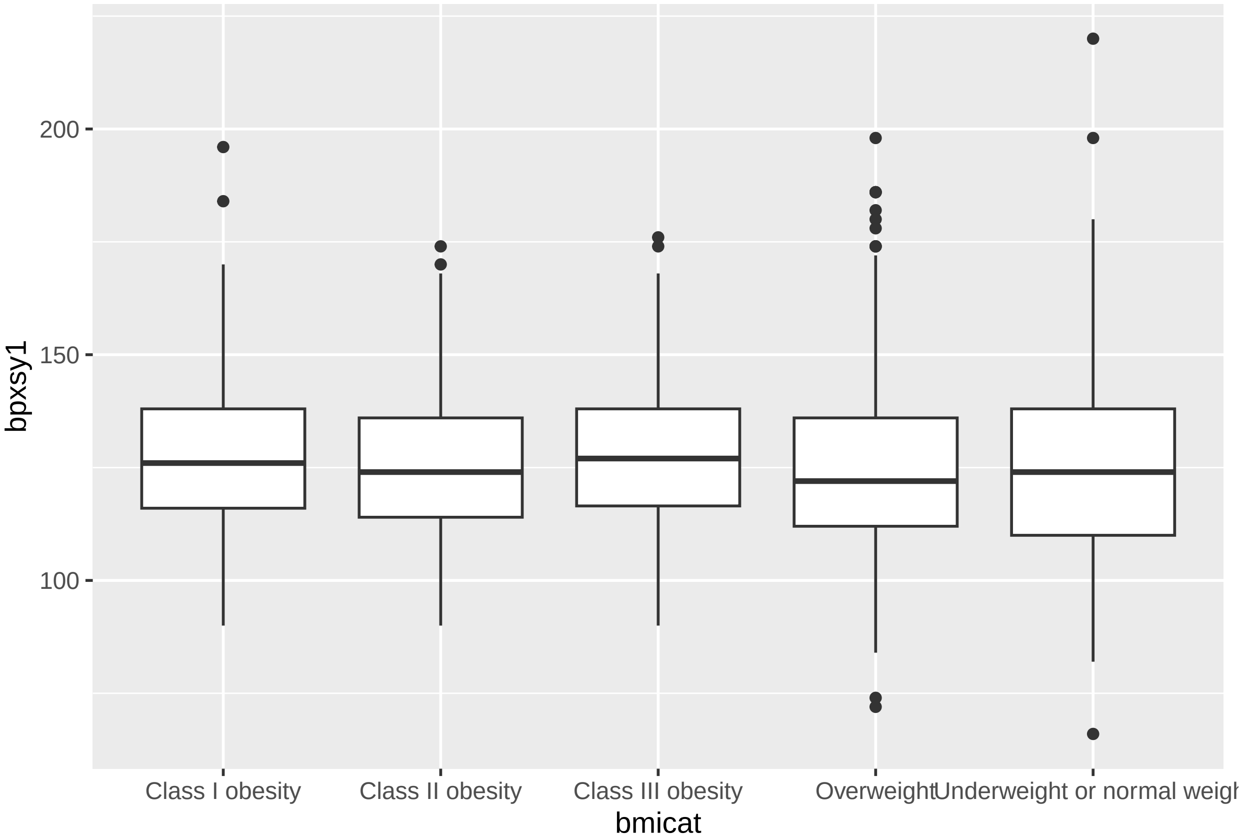
```
. = ottr::check("tests/p10.R")
```

```
##
## All tests passed!
```

Part 2: ANOVA

11. [1 point] We are interested in looking at the systolic blood pressure, `bpxsy1`, by BMI category, `bmicat`. Generate an appropriate box plot to visualize these data.

```r
plot11 <- ggplot(nhanes, aes(x= bmicat, y = bpxsy1)) + geom_boxplot()
plot11
```



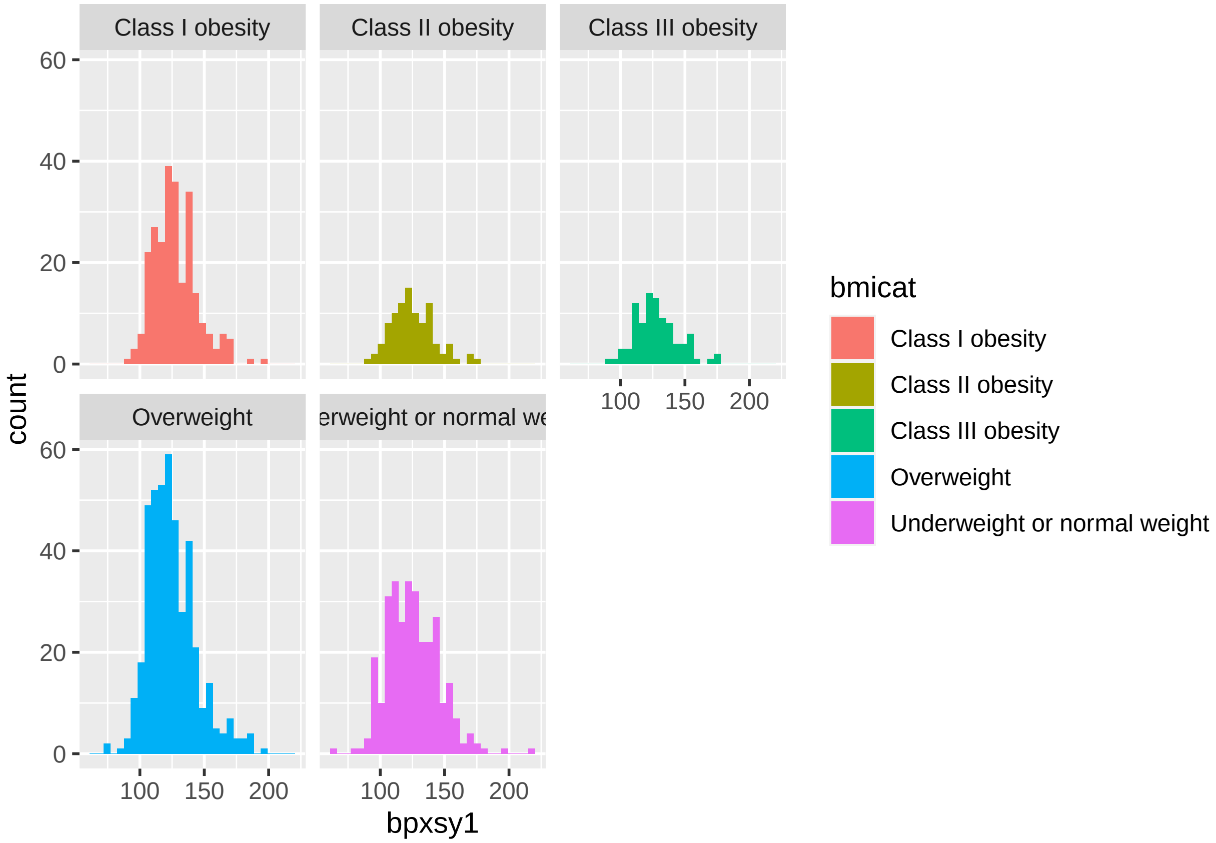
```r
. = ottr::check("tests/p11.R")
```

```
##
## All tests passed!
```

12. [1 point] Now generate a set of faceted histograms that show the same data. It might be useful to assign a fill color to each category.

```
plot12 <- ggplot(nhanes, aes(x = bpxsy1)) + geom_histogram(aes(fill = bmicat))+
          facet_wrap(~bmicat)
plot12
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
. = ottr::check("tests/p12.R")
```

```
##
## All tests passed!
```

13. [1 point] Summarize the means and standard deviations of the outcome for each BMI category. Assign `p13` to a dataframe with the mean systolic blood pressure assigned to `mean_bp` and the standard deviation assigned to `sd_bp`.

```r
p13 <- nhanes %>% group_by(bmicat) %>% summarize(mean_bp = mean(bpxsy1),
                                                  sd_bp = sd(bpxsy1))
p13
```

```
## # A tibble: 5 x 3
##   bmicat                         mean_bp sd_bp
##   <chr>                            <dbl> <dbl>
## 1 Class I obesity                   128.  17.0
## 2 Class II obesity                  126.  16.9
## 3 Class III obesity                 128.  17.0
## 4 Overweight                        125.  19.0
## 5 Underweight or normal weight      125.  20.3
```

```r
. = ottr::check("tests/p13.R")
```

```
## 
## All tests passed!
```

14. [1 point] Use an R function to test whether there is evidence to reject the null hypothesis of no difference between mean blood pressure by BMI category.

```
p14 <- aov(bpxsy1 ~ bmicat, data = nhanes)
tidy(p14) # tidy displays your output. It lives in the `broom` package
```

```
## # A tibble: 2 x 6
##   term         df   sumsq meansq statistic p.value
##   <chr>     <dbl>   <dbl>  <dbl>     <dbl>   <dbl>
## 1 bmicat        4   1651.   413.      1.19   0.314
## 2 Residuals  1173 406837.   347.     NA     NA
```

```
. = ottr::check("tests/p14.R")
```

```
##
## All tests passed!
```

15. [1 point] Conduct a Tukey's HSD test using these data. What can you conclude assuming a standard error rate of 5%?

```
p15 <- TukeyHSD(p14)
tidy(p15)
```

```
## # A tibble: 10 x 7
##    term   contrast             null.value estimate conf.low conf.high adj.p.value
##    <chr>  <chr>                     <dbl>    <dbl>    <dbl>     <dbl>       <dbl>
##  1 bmicat Class II obesity-C~           0  -2.09      -8.19      4.01       0.883
##  2 bmicat Class III obesity-~           0   0.638     -5.61      6.89       0.999
##  3 bmicat Overweight-Class I~           0  -2.60      -6.63      1.43       0.396
##  4 bmicat Underweight or nor~           0  -2.18      -6.51      2.16       0.646
##  5 bmicat Class III obesity-~           0   2.73      -4.74     10.2        0.856
##  6 bmicat Overweight-Class I~           0  -0.510     -6.25      5.23       0.999
##  7 bmicat Underweight or nor~           0  -0.0871    -6.04      5.87       1.00
##  8 bmicat Overweight-Class I~           0  -3.24      -9.13      2.66       0.562
##  9 bmicat Underweight or nor~           0  -2.81      -8.92      3.29       0.716
## 10 bmicat Underweight or nor~           0   0.423     -3.38      4.22       0.998
```

Based on the Tukey's HSD test results and a standard error rate of 5%, we can conclude that there is statistically significant differences between the group means we have compared. Therefore, we fail to reject the null hypothesis of no difference between mean blood pressure by BMI category.

```
. = ottr::check("tests/p15.R")
```

```
##
## All tests passed!
```

Submission

For assignments in this class, you'll be submitting using the **Terminal** tab in the pane below. In order for the submission to work properly, make sure that:

1. Any image files you add that are needed to knit the file are in the `src` folder and file paths are specified accordingly.
2. You **have not changed the file name** of the assignment.
3. The file knits properly.

Once you have checked these items, you can proceed to submit your assignment.

1. Click on the **Terminal** tab in the pane below.
2. Copy-paste the following line of code into the terminal and press enter.

```
cd; cd ph142-su23/lab/lab09; python3 turn_in.py
```

3. Follow the prompts to enter your Gradescope username and password.
4. If the submission is successful, you should see "Submission successful!" appear as the output. **Check your submission on the Gradescope website to ensure that the autograder worked properly and you received credit for your correct answers. If you think the autograder is incorrectly grading your work, please post on Ed!**
5. If the submission fails, try to diagnose the issue using the error messages–if you have problems, post on Ed under the post "Datahub Issues".

The late policy will be strictly enforced, **no matter the reason**, including submission issues, so be sure to submit early enough to have time to diagnose issues if problems arise.

END