Unit 6 Homework: Tests and and Confidence Intervals

w203: Statistics for Data Science

Low-Oxygen Statistics

The file `expeditions.csv` contains data about 10,000 climbing expeditions in the Himalayan Mountains of Nepal. The data was compiled by the Himalayan Database and published in csv format on Tidy Tuesday.

First, navigate to https://github.com/rfordatascience/tidytuesday/tree/master/data/2020/2020-09-22 to read some basic information about the data and examine the codebook.

The variable `highpoint_metres` represents the highest elevation reached by each expedition. Your task is to test whether the mean highest elevation is above 7400 meters.

a. Using the documentation about the data, your background knowledge, and the data itself, assess whether the assumptions underlying a valid t-test are met. If plots are useful to make this argument, include them; if numeric statements are useful to make this argument, use them.

A one sample test of means compares the mean of a sample to a pre-specified value and tests for a deviation from that value.

one samples independent t-test assume the following characteristics about the data:

(1)Independence of the observations. Each subject should belong to only one group. There is no relationship between the observations in each group. (2) No significant outliers in the two groups (3) Normality. the data for each group should be approximately normally distributed. (4) Homogeneity of variances. the variance of the outcome variable should be equal in each group.

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

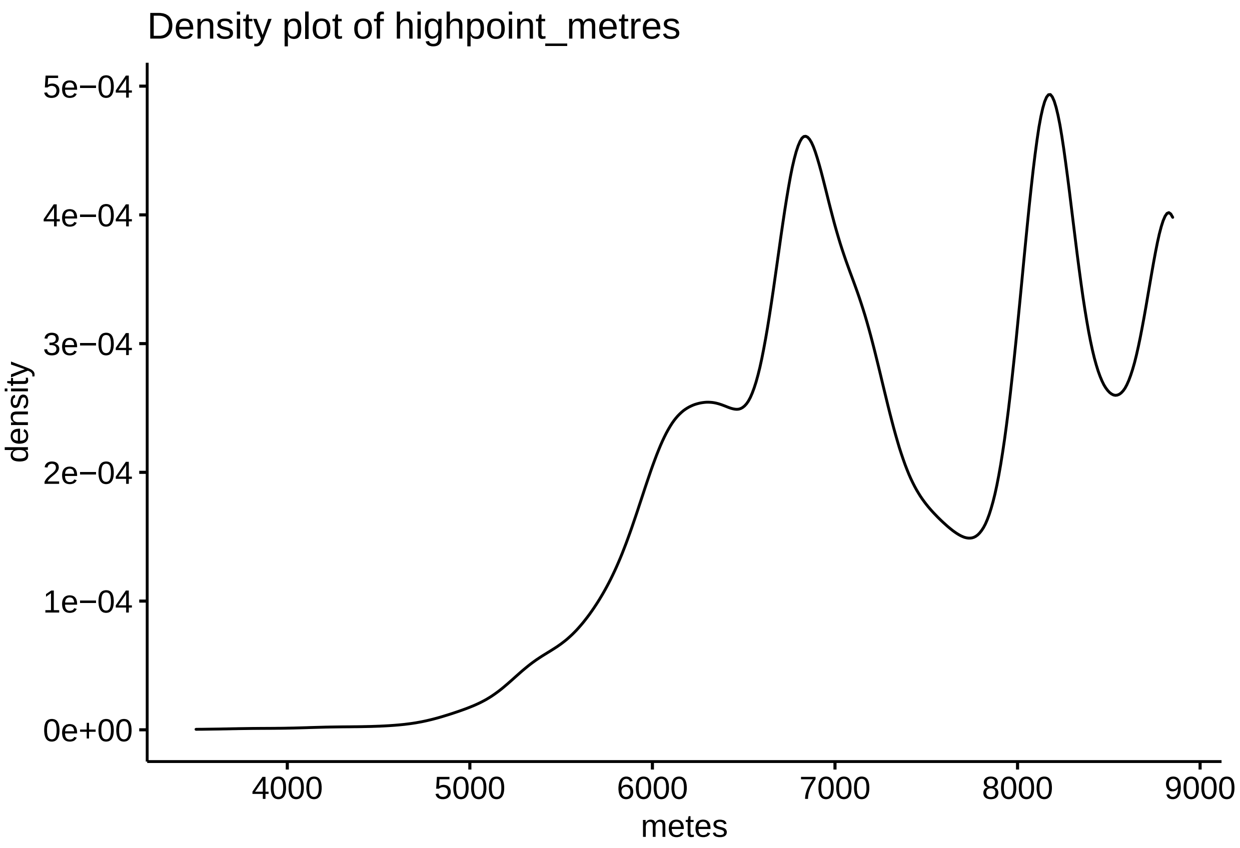
```
library(ggpubr)
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(dplyr)


# Load the data
e_read<-read.csv('expeditions.csv', header=TRUE)
e<-e_read%>%drop_na(highpoint_metres)
metres<-e$highpoint_metres

#data("genderweight", package = "datarium")
# Show a sample of the data by group
#set.seed(123)
#data()
#head(e, 6)
#metres %>% sample_n_by(group, size = 2)

set.seed(1234)
sample_n(e, 10)
```

```
##    expedition_id peak_id   peak_name year season basecamp_date highpoint_date
## 1      EVER12122    EVER     Everest 2012 Spring    2012-04-28     2012-05-25
## 2      ANN113301    ANN1 Annapurna I 2013 Autumn    2013-09-21     2013-10-09
## 3      AMAD11304    AMAD  Ama Dablam 2011 Autumn    2011-10-17     2011-10-25
## 4      EVER13188    EVER     Everest 2013 Spring    2013-04-18     2013-05-22
## 5      EVER11301    EVER     Everest 2011 Autumn    2011-09-11     2011-10-06
## 6      EVER17181    EVER     Everest 2017 Spring    2017-04-18     2017-05-21
## 7      EVER90105    EVER     Everest 1990 Spring    1990-03-28     1990-05-16
## 8      TUKU86301    TUKU     Tukuche 1986 Autumn    1986-10-12     1986-10-22
## 9      PUMO71101    PUMO      Pumori 1971 Spring          <NA>     1971-04-19
## 10     ANN178301    ANN1 Annapurna I 1978 Autumn    1978-08-26     1978-10-15
##    termination_date                     termination_reason highpoint_metres
## 1        2012-05-29     Accident (death or serious injury)             8445
## 2        2013-10-11                    Success (main peak)             8091
## 3        2011-10-27                    Success (main peak)             6814
## 4        2013-05-26                    Success (main peak)             8850
## 5        2011-10-08       Bad weather (storms, high winds)             7700
## 6        2017-05-28                    Success (main peak)             8850
## 7        1990-05-20 Illness, AMS, exhaustion, or frostbite             8200
## 8              <NA>                    Success (main peak)             6920
## 9              <NA>       Bad weather (storms, high winds)             5740
## 10       1978-10-24                    Success (main peak)             8091
##    members member_deaths hired_staff hired_staff_deaths oxygen_used
## 1        1             0           1                  0        TRUE
## 2        2             0           0                  0       FALSE
## 3        4             0           1                  0       FALSE
## 4        3             0           2                  0        TRUE
## 5        1             0           1                  0       FALSE
## 6        4             0           2                  0        TRUE
## 7        6             0           3                  0       FALSE
## 8       50             0           0                  0       FALSE
## 9        5             0           0                  0       FALSE
## 10      13             2           6                  0        TRUE
##                              trekking_agency
## 1                             Asian Trekking
## 2                         Royal Orchid Treks
## 3                 Sherpa Shangri-La Trekking
## 4                         Seven Summit Treks
## 5  Bochi Bochi Treks (Asian Trekking permit)
## 6                        Rolwaling Excursion
## 7                                Nepal Himal
## 8                  Royal Nepalese Army (RNA)
## 9                                       <NA>
## 10                           Mountain Travel
```

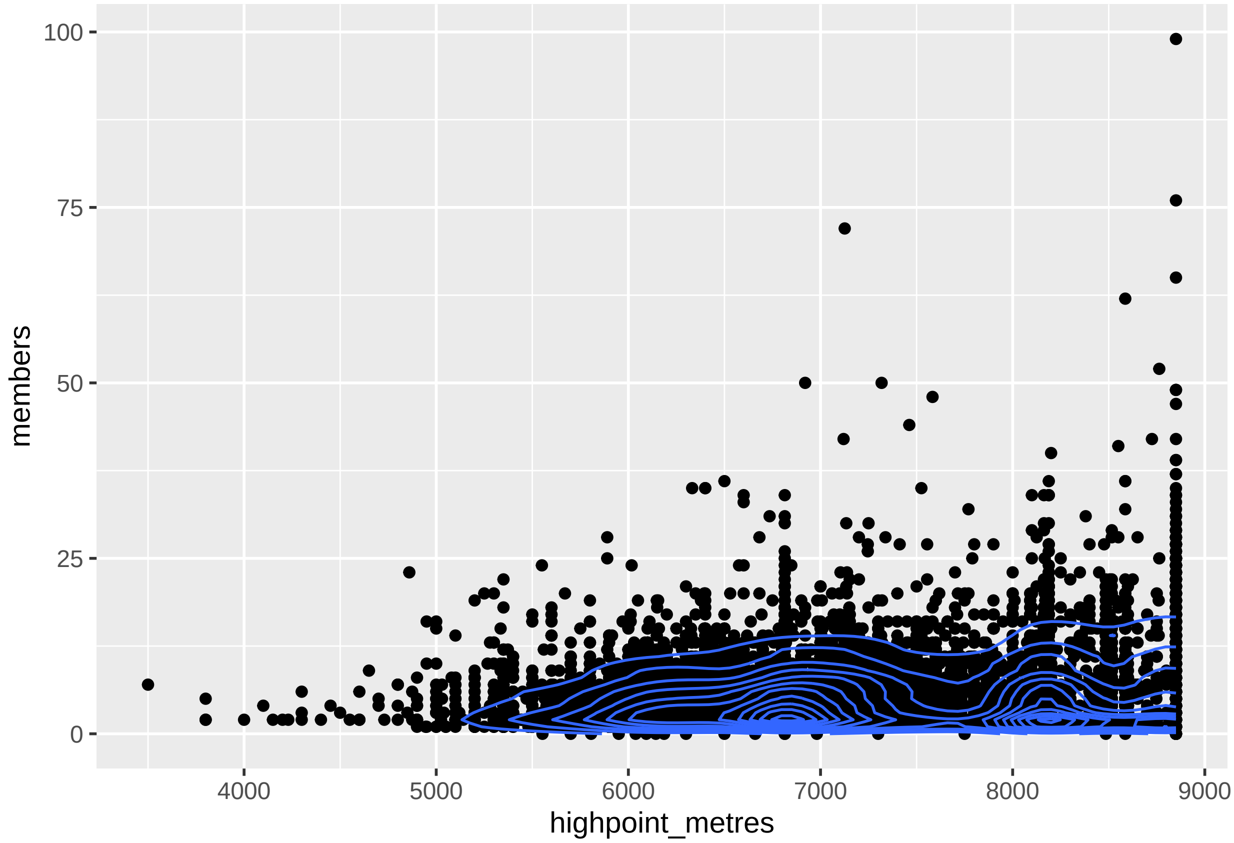
```
library("ggpubr")
ggdensity(metres,
          main = "Density plot of highpoint_metres",
          xlab = "metes")
```



```
mean_value <-mean(metres)
print(mean_value)
```

```
## [1] 7408.924
```

```
ggplot(e, aes(x = highpoint_metres, y = members))+
  geom_point()+
  stat_density2d()
```



b. Provide an argument for why you should conduct a two-tailed test in this case, even though your personal interest is primarily in whether the mean is higher than 7400.

I use a two-tailed test because I care whether the mean is greater than or less than the target value, i.e. 7400. Two-tailed tests can test for effects in both directions. When performing a two-tailed test, I split the significance level percentage between both tails of the distribution. I do not use one-tailed test because it is only justified if we have a specific prediction about the direction of the t-test, Or if we completely uninterested in the possibility that the opposite outcome could be true.

c. Compute the t-statistic by plugging in the values from the data manually into the formula. A *great* solution would write a function (perhaps called `t_statistic`) that takes arguments and returns a value. However, writing a function isn't necessary for a full solution. Feel free to use functions `mean()`, `sd()`, and `sqrt()`.

```
t_statistic <- function(highpoint_metres, mean_highest_elevation) {
   t <- (mean(highpoint_metres)-mean_highest_elevation)/
     (sd(highpoint_metres)/sqrt(length(highpoint_metres)))
   cat("t = ", t)
}
```

```
d<- e$highpoint_metre
h<-7400

t<-t_statistic(d, h)
```

```
## t =  0.8790473
```

d. Using `qt()`, compute the t-critical value for a two-tailed test.

```
#df is degree of freedom

df<-length(metres)-1

df
```

```
## [1] 9949
```

```
#compute the t-critical value for a left-tailed test
#t_critical_left_tailed<-qt(p=.05, df, lower.tail=TRUE)
#cat("t_critical_left_tailed=", t_critical_left_tailed)
#compute the t-critical value for a right-tailed test
#t_critical_right_tailed<-qt(p=.05, df, lower.tail=FALSE)
#cat("t_critical_right_tailed=", t_critical_right_tailed)

#compute the t-critical value for a two-tailed test
t_critical_two_tailed<-qt(p=.05/2, df, lower.tail=FALSE)
cat("t_critical_two_tailed=", t_critical_two_tailed)
```

```
## t_critical_two_tailed= 1.960202
```

When perform a two-tailed test, there will be two critical values. In this case, the T critical values are 1.960202 and -1.960202. Thus, if the test statistic is less than -1.960202 or greater than 1.960202, the results of the test are statistically significant.

e. Compute the p-value for your two-tailed test. You may use the `pt()` function.

```
t<-0.879043

p_value<-2*pt(-abs(t),df=length(metres)-1)

p_value
```

```
## [1] 0.3793992
```

f. Explain what your rejection decision should be in two ways.

When perform a two-tailed test, there will be two critical values. In this case, the T critical values are 1.960202 and -1.960202. Thus, if the test statistic is less than -1.960202 or greater than 1.960202, the results of the test are statistically significant.

f. Confirm that your work is correct, by running the `t.test` command.

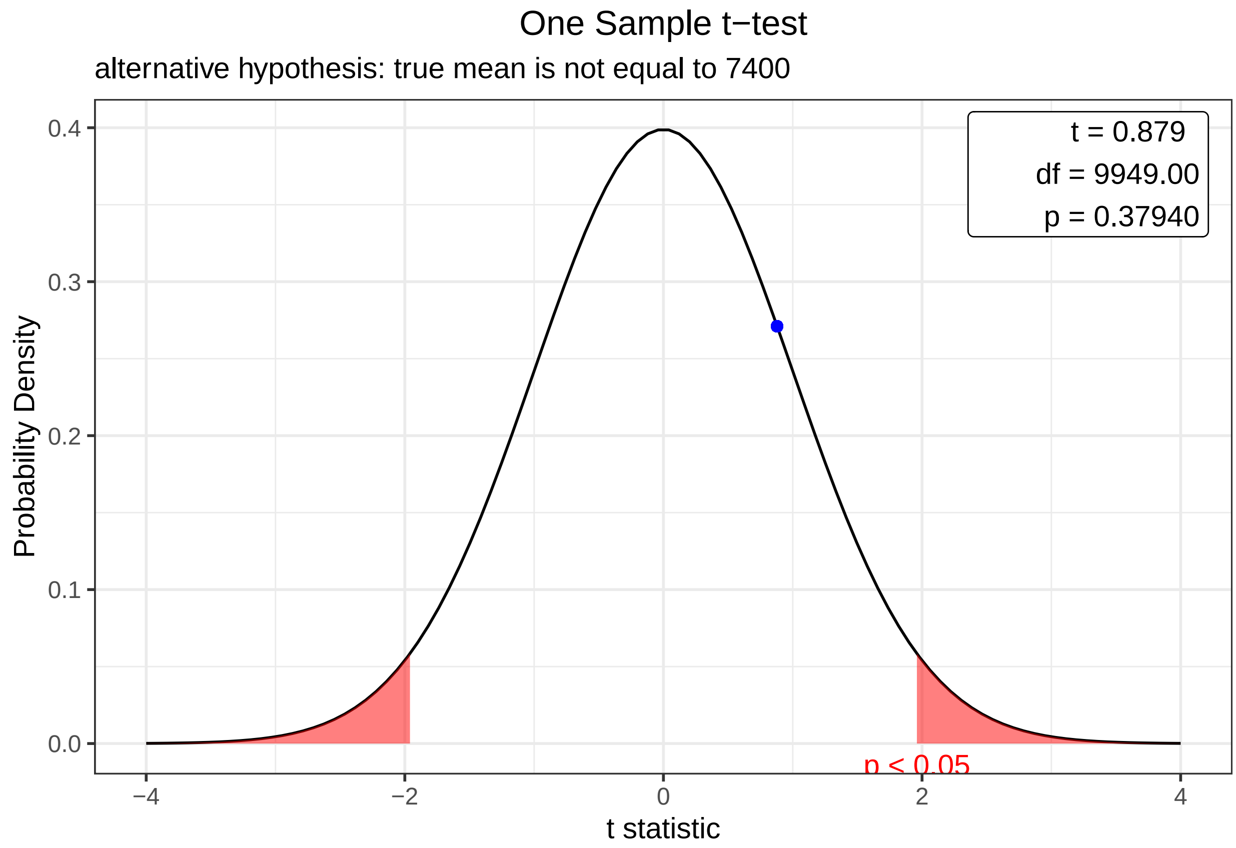
```
#require(moonBook)
require(webr)
```

```
## Loading required package: webr
```

```
t.test(e$highpoint_metres, mu=7400, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  e$highpoint_metres
## t = 0.87905, df = 9949, p-value = 0.3794
## alternative hypothesis: true mean is not equal to 7400
## 95 percent confidence interval:
##  7389.024 7428.823
## sample estimates:
## mean of x
##  7408.924
```

```
plot(t.test(e$highpoint_metres,mu=7400))
```



g. Evaluate the practical significance of your result.