Himalayan Expeditions

Srishti Mehra

9/30/2020

Low-Oxygen Statistics

We will construct a 95% confidence interval for the mean highest elevation for expeditions represented by the data set. We will evaluate assumptions for being able to run any statistical computation and see it's implications. We will also run the 95% confidence interval by hand and with the help of a statistical software to get experience with both methods.

```
#Getting expeditions data
expeditions <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/da
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   expedition_id = col_character(),
##   peak_id = col_character(),
##   peak_name = col_character(),
##   year = col_double(),
##   season = col_character(),
##   basecamp_date = col_date(format = ""),
##   highpoint_date = col_date(format = ""),
##   termination_date = col_date(format = ""),
##   termination_reason = col_character(),
##   highpoint_metres = col_double(),
##   members = col_double(),
##   member_deaths = col_double(),
##   hired_staff = col_double(),
##   hired_staff_deaths = col_double(),
##   oxygen_used = col_logical(),
##   trekking_agency = col_character()
## )
```

```
#Cleaning the data
# Highpoint of 0 is most likely missing value
# - they may have not started the expedition for some reason
highpoint_metres = ifelse(expeditions$highpoint_metres == 0, NA,
                          expeditions$highpoint_metres)
print(length(highpoint_metres))
```

```
## [1] 10364
```

```r
#There is no need of the NA values in the calculation of the mean so we will remove them
cleaned_highpoint_meters <- na.omit(highpoint_metres)
print(length(cleaned_highpoint_meters))
```

```
## [1] 9950
```

a. There are two assumptions we need to be able to draw t-based confidence intervals: the points must be i.i.d and normally distributed

```r
library("tidyverse")
```

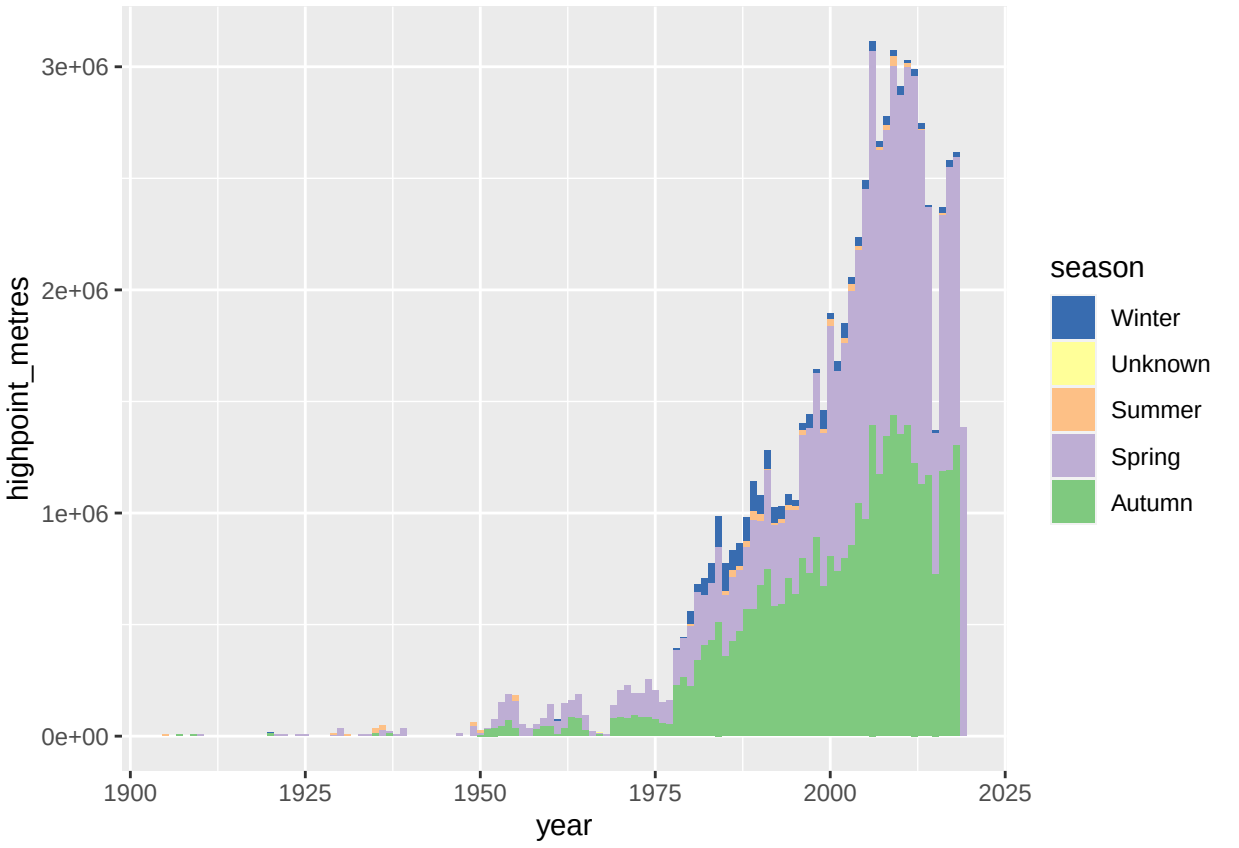
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

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

```r
ggplot(data = expeditions, mapping = aes(x=year, y=highpoint_metres, fill=season)) +
  geom_bar(stat="identity", position = position_stack(reverse=TRUE)) +
  guides(fill = guide_legend(reverse=TRUE))+
  scale_fill_brewer(palette = "Accent")
```

```
## Warning: Removed 414 rows containing missing values (position_stack).
```



The above graph shows us that they are not absolutely i.i.d because:

They are not identically distributed:

1. The number of expeditions were somewhat increasing till the 2010s and have declined since. It seems that history of expeditions influences how many are attempted after them.
2. They have also not been identically distributed across seasons with spring and autumn being more popular than others.
3. These popularities also seem influenced from lessons learned in past - possibly of climbing conditions.

Independence cannot be guaranteed either since the number of concurrent expeditions at a time could affect:

1. Crowding at summit, like it happened last year at Mount Everest:

https://www.nationalgeographic.com/adventure/2019/05/everest-season-deaths-controversy-crowding

This derailed anticipated plans causing some to not make it to the top. Thus some expeditions were dependent on others.

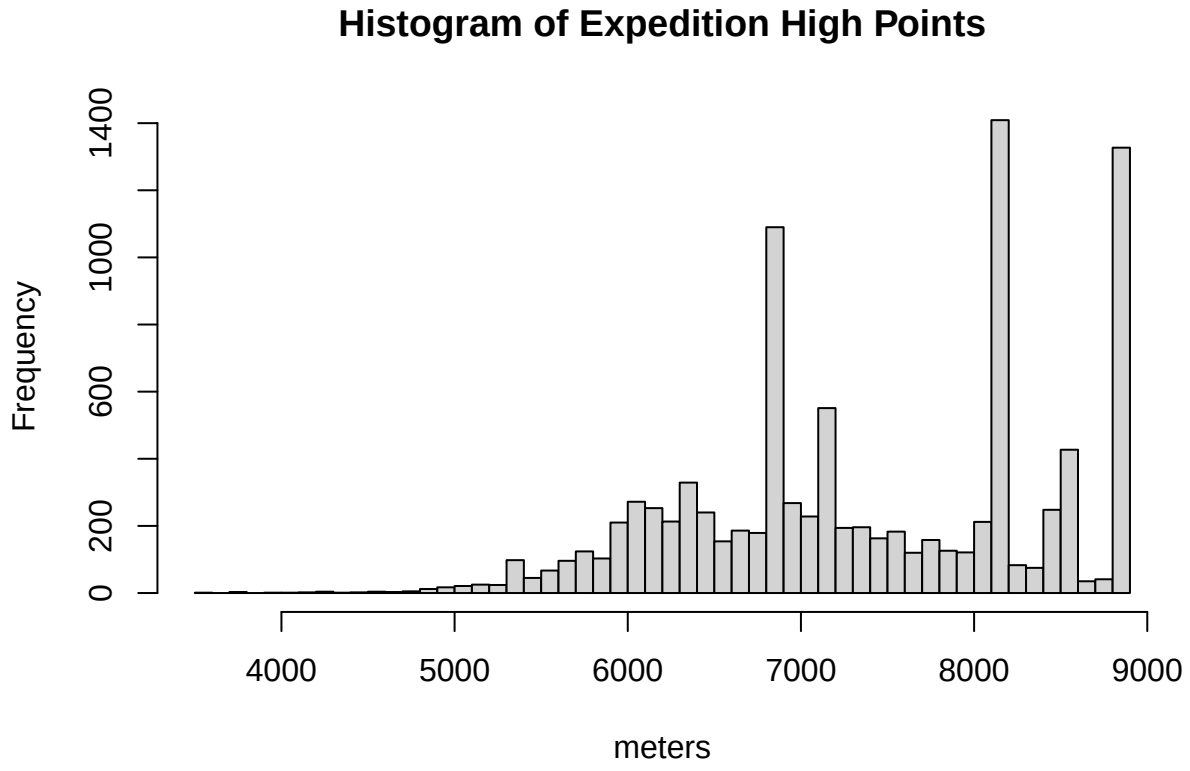
2. If multiple expeditions are attempting a climb, they could share knowledge of route, climate conditions, and resources making it more likely to achieve summit. Thus defying independence.

Figure 1: Overcrowing at Mount Everest in 2019

Since the assumption for identically and independent distribution is not met, we must be cautious with the conclusions we draw. The inability to satisfy this assumption hampers us from generalizing any conclusion we get from this sample to the general population. While we can

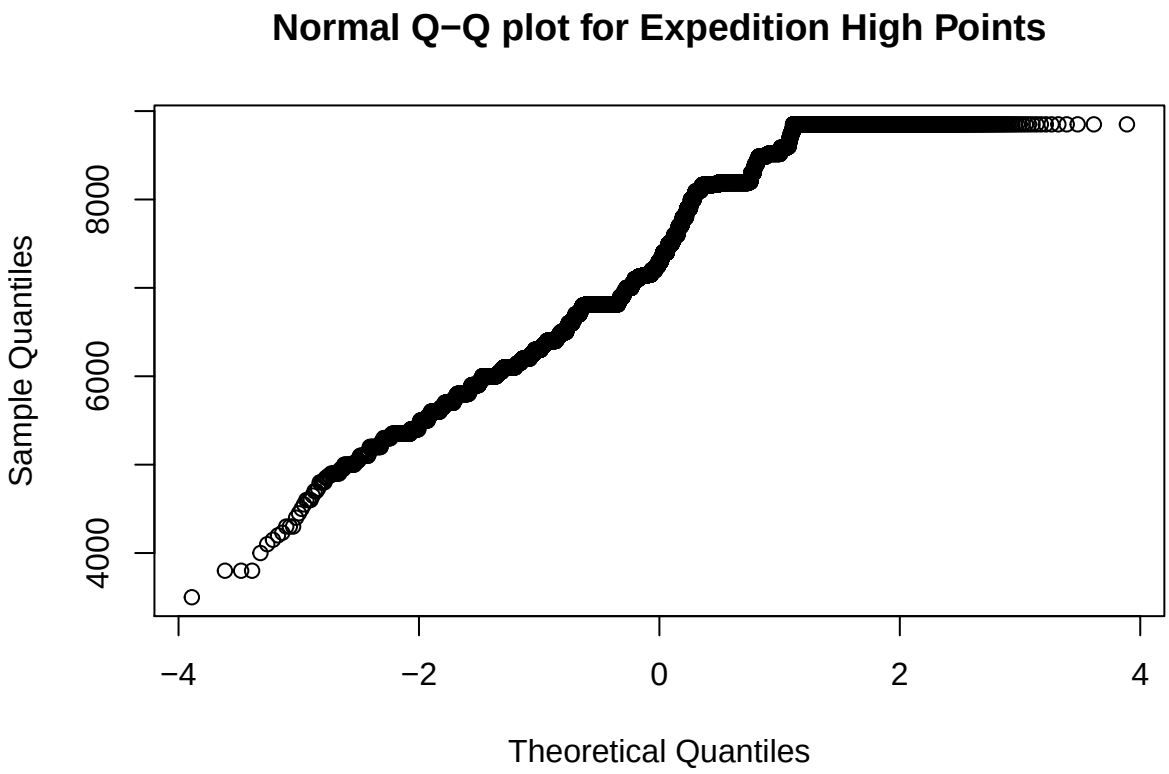
Now obeserving the assumption of normality:

```
#Plotting histogram to see if the distribution is normal
hist(cleaned_highpoint_meters, breaks = 50,
     main = "Histogram of Expedition High Points",
     xlab = "meters")
```




```
#There seem to be some outliers to normal but those that a t-distribution's robustness can handle
```

```
#Performing one more normality check
qqnorm(cleaned_highpoint_meters,
       main = "Normal Q-Q plot for Expedition High Points")
```



Due to the deviations from normality in the distribution of Expedition High Points we will compute t-based Confidence Intervals for it's mean

Computing the 95% confidence interval

By hand

```
mean_hp = mean(cleaned_highpoint_meters)
sd_hp = sd(cleaned_highpoint_meters)
n = length(cleaned_highpoint_meters)

#For a 95% confidence level
t_critical_value=qt(0.975,df=n-1)

left_bound<- mean_hp - t_critical_value*sd_hp/sqrt(n)
right_bound<- mean_hp + t_critical_value*sd_hp/sqrt(n)


#Confidence Level
cat(sprintf("Confidence Interval (using t-distribution) is [%f,%f]",
            left_bound, right_bound))
```

```
## Confidence Interval (using t-distribution) is [7389.024443,7428.823195]
```

By using statistical software

```r
#On original data
#we do not need to provide the cleaned data here because t.test handles NAs and Os
t.test(expeditions$highpoint_metres)
```

```
##
##  One Sample t-test
##
## data:  expeditions$highpoint_metres
## t = 729.82, df = 9949, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  7389.024 7428.823
## sample estimates:
## mean of x
##  7408.924
```

```r
t.test(expeditions$highpoint_metres, conf.level = .95)$conf.int
```

```
## [1] 7389.024 7428.823
## attr(,"conf.level")
## [1] 0.95
```

```r
#If we were still to verify with clean data, we see that we get the same confidence intervals
t.test(cleaned_highpoint_meters)
```

```
##
##  One Sample t-test
##
## data:  cleaned_highpoint_meters
## t = 729.82, df = 9949, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  7389.024 7428.823
## sample estimates:
## mean of x
##  7408.924
```

```r
t.test(cleaned_highpoint_meters, conf.level = .95)$conf.int
```

```
## [1] 7389.024 7428.823
## attr(,"conf.level")
## [1] 0.95
```