End Semester Examination

1940223, Pranav Gopalkrishna

2021-04-16

# Preliminary steps

setwd("~/Documents/Study/computerScience/programming/r/rAssignments")  
data = read.csv("data/weatherAustralia.csv")  
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.6.2

##   
## Attaching package: 'dplyr'

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

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# **Data**

This data has measures of the temperatures, wind movement and predictions of rain in Australia, for the past 10 years. My variable of interest is wind gust speed, which can be an important factor in predicting storms and rains.

head(data, 5)

## Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir  
## 1 01/12/08 Albury 13.4 22.9 0.6 NA NA W  
## 2 02/12/08 Albury 7.4 25.1 0.0 NA NA WNW  
## 3 03/12/08 Albury 12.9 25.7 0.0 NA NA WSW  
## 4 04/12/08 Albury 9.2 28.0 0.0 NA NA NE  
## 5 05/12/08 Albury 17.5 32.3 1.0 NA NA W  
## WindGustSpeed WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am  
## 1 44 W WNW 20 24 71  
## 2 44 NNW WSW 4 22 44  
## 3 46 W WSW 19 26 38  
## 4 24 SE E 11 9 45  
## 5 41 ENE NW 7 20 82  
## Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm  
## 1 22 1007.7 1007.1 8 NA 16.9 21.8  
## 2 25 1010.6 1007.8 NA NA 17.2 24.3  
## 3 30 1007.6 1008.7 NA 2 21.0 23.2  
## 4 16 1017.6 1012.8 NA NA 18.1 26.5  
## 5 33 1010.8 1006.0 7 8 17.8 29.7  
## RainToday RainTomorrow  
## 1 No No  
## 2 No No  
## 3 No No  
## 4 No No  
## 5 No No

# Preparing the population

# Getting rid of null values for the chosen field  
windgustspeeds = c()  
locations = c()  
popl\_size = length(data$WindGustSpeed)  
popl\_size

## [1] 15076

for (i in c(1:popl\_size))  
{  
 flag = 0  
 if(!is.na(data$WindGustSpeed[i]))  
 {  
 windgustspeeds = c(windgustspeeds, data$WindGustSpeed[i])  
 locations = c(locations, data$Location[i])  
 }  
}  
popl = data.frame(windgustspeeds, locations)  
head(popl, 10)

## windgustspeeds locations  
## 1 44 1  
## 2 44 1  
## 3 46 1  
## 4 24 1  
## 5 41 1  
## 6 56 1  
## 7 50 1  
## 8 35 1  
## 9 80 1  
## 10 28 1

# Checking if NA values are still there  
sum(is.na(windgustspeeds))

## [1] 0

# # Resetting population size popl\_size = length(windgustspeeds) # Population total and mean popl\_total = sum(windgustspeeds) popl\_mean = popl\_total/popl\_size

# Stratification based on location # Dividing population into strata. summary(data$Location)

## Albury BadgerysCreek Cobar CoffsHarbour Moree   
## 3040 3009 3009 3009 3009

We can see we have five locations. We shall divide the data by the locations, since wind gust speeds are more likely to be more homogenously distributed for a single location than for multiple locations, considering the current lack of information about these locations. Hence, we have five strata, as follows…

# stratum1 = filter(popl, popl$locations == 1) stratum2 = filter(popl, popl$locations == 2) stratum3 = filter(popl, popl$locations == 3) stratum4 = filter(popl, popl$locations == 4) stratum5 = filter(popl, popl$locations == 5)

# Using optimum allocation

Optimum allocation is designed to get most efficient sample sizes for estimating population parameters. Hence, we use this method.

# Using optimum allocation to find the sample size for each stratum  
sample\_size = 1000  
strata = c(stratum1, stratum2, stratum3, stratum4, stratum5)  
N = c()  
S = c()  
denominator\_sum = 0  
for(i in c(1:5))  
 {  
 Nh = length(strata[2 \* i - 1]$windgustspeeds)  
 Sh = sd(strata[2 \* i - 1]$windgustspeeds)  
 denominator\_sum = denominator\_sum + Nh \* Sh \* Sh  
 N = c(N, Nh)  
 S = c(S, Sh \* Sh)  
}  
print(denominator\_sum)

## [1] 2145761

print(N)

## [1] 3008 2933 2967 2650 2932

print(S)

## [1] 178.9759 164.8594 115.6067 152.1413 128.8163

n = c()  
for(i in c(1:5))  
{  
 x = N[i] \* S[i] \* sample\_size / denominator\_sum  
 n = c(n, x)  
}  
print(n)

## [1] 250.8944 225.3432 159.8524 187.8935 176.0166

# Sampling

# # Taking samples sample1 = sample(strata[2 \* 1 - 1]$windgustspeeds, n[1], replace = FALSE) sample2 = sample(strata[2 \* 2 - 1]$windgustspeeds, n[2], replace = FALSE) sample3 = sample(strata[2 \* 3 - 1]$windgustspeeds, n[3], replace = FALSE) sample4 = sample(strata[2 \* 4 - 1]$windgustspeeds, n[4], replace = FALSE) sample5 = sample(strata[2 \* 5 - 1]$windgustspeeds, n[5], replace = FALSE) Point estimation

# Estimating population mean and total  
ybar1 = mean(sample1)  
ybar2 = mean(sample2)  
ybar3 = mean(sample3)  
ybar4 = mean(sample4)  
ybar5 = mean(sample5)  
  
ybarh = c(ybar1, ybar2, ybar3, ybar4, ybar5)  
print(ybarh)

## [1] 34.10000 34.17333 38.57862 36.73262 40.60795

**# Estimated mean**  
ybarst = sum(Nh \* ybarh) / popl\_size  
print(ybarst)

## [1] 37.2707

**# Estimated population total**  
est\_popl\_total = ybarst \* popl\_size  
print(est\_popl\_total)

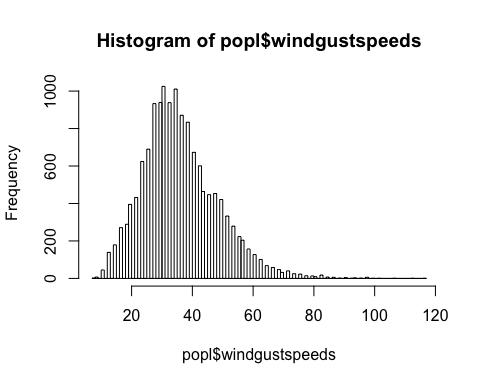
## [1] 540052.5

# Interval estimation

# Confidence intervals  
# Finding the bound on error of estimation for ybarst through computation...  
ybarstList = c()  
max = 10000  
for(i in c(1:max))  
{  
 sample1 = sample(stratum1$windgustspeeds, n[1], replace = FALSE)  
 sample2 = sample(stratum2$windgustspeeds, n[2], replace = FALSE)  
 sample3 = sample(stratum3$windgustspeeds, n[3], replace = FALSE)  
 sample4 = sample(stratum4$windgustspeeds, n[4], replace = FALSE)  
 sample5 = sample(stratum5$windgustspeeds, n[5], replace = FALSE)  
   
 ybar1 = mean(sample1)  
 ybar2 = mean(sample2)  
 ybar3 = mean(sample3)  
 ybar4 = mean(sample4)  
 ybar5 = mean(sample5)  
 ybarh = c(ybar1, ybar2, ybar3, ybar4, ybar5)  
   
 ybarst = sum(Nh \* ybarh) / N  
 ybarstList = c(ybarstList, ybarst)  
}

Finding the variance of the 10000 ybarst values…

var\_ybarst = var(ybarstList)  
  
# Checking if population can be said to follow normal distribution...  
hist(popl$windgustspeeds, breaks = 100)



# It is negatively skewed, hence cannot be said to follow normal distribution.  
  
# Given that we need the 95% confidence interval, bound on error of estimate, B, is 0.05.  
B = 0.05  
# Using Chebychev's inequality, we have that  
# P((X - mu) ≥ k\*sigma) ≤ 1/k^2 ... (1)  
# where X is the variable, mu is its mean, sigma is its standard deviation, and k is the number of standard deviations from the mean.  
# Applying to estimated population mean and the given bound on error of estimation, we get  
# # P((ybarst - popl\_mean) ≥ B) ≤ 1/k^2 ... (2)  
# We do this because we want to find k, that is, the number of standard deviations required so that the error of estimation i.e. p - P is under B i.e. the bound on error of estimation.  
# Comparing (1) and (2), we see that  
# B = k \* sigma, where sigma is the standard deviation of the variable.  
# But the variable is estimated population proportion, i.e. p.  
# Hence, sigma = sqrt(V(ybarst))  
# Hence, B = k \* sqrt(V(ybarst))  
# Hence, k = B / sqrt(V(ybarst))  
  
# Variance of ybarst is given by var\_ybarst  
  
# Finding k  
k = B / sqrt(var\_ybarst)  
print(k)

## [1] 0.005670691

# Confidence intervals

## Population mean

# Hence, the 95% confidence interval for estimate of Ybar i.e. population mean is  
c(ybarst - k\*sqrt(var\_ybarst), ybarst + k\*sqrt(var\_ybarst))

## [1] 178.4208 182.9845 180.8870 202.5312 183.0469 178.5208 183.0845 180.9870  
## [9] 202.6312 183.1469

I.e. (202.6312, 183.1469)

There is 95% chance of the estimation lying in this interval.

## Population total

# Now, variance of estimated population total is given by population size times variance of proportion.  
# Hence, V(N\_est) = N \* V(ybarst)  
# Similarly the 95% confidence interval for estimate of A i.e. population total is  
c(est\_popl\_total - k\*popl\_size\*sqrt(var\_ybarst), est\_popl\_total + k\*popl\_size\*sqrt(var\_ybarst))

## [1] 539328 540777

I.e. c(539328, 540777)

There is 95% chance of the estimation lying in this interval.