DSCI/MATH 530 RLab Four

Overview

This lab will support finding confidence intervals for the population mean within R and also their interpretation. Also we some cover model checking. For those interested the EXTRA CREDIT is an example of a more modern stats approach where we leverage computation to gain some insight not available by analytic formulas.

Exercise 6.12

```
set.seed(123)
CI=matrix(nrow=25,ncol=2)
sample_means=numeric(25)
for (i in 1:25){
  x1=rnorm(20,50,6)
  sample_means[i]=mean(x1)
}
se=6/sqrt(20)
sample_means
```

```
##  [1] 50.84974 49.69246 50.63891 49.28050 52.25057 47.84372 49.76990 48.95998
##  [9] 51.00134 49.19865 50.35690 49.76996 49.86055 51.25689 52.36966 49.76593
## [17] 50.30130 49.55444 50.17152 49.12012 48.72822 51.23950 49.78155 51.80738
## [25] 51.61886
```

```
for (i in 1:25){
  me=qt(.975,df=19)*se
  lower=sample_means[i]-me
  upper=sample_means[i]+me
  CI[i,]=c(lower,upper)
}
CI
```

```
##           [,1]     [,2]
##  [1,] 48.04166 53.65783
##  [2,] 46.88437 52.50054
##  [3,] 47.83082 53.44700
##  [4,] 46.47241 52.08858
##  [5,] 49.44248 55.05865
##  [6,] 45.03563 50.65180
##  [7,] 46.96182 52.57799
##  [8,] 46.15190 51.76807
##  [9,] 48.19325 53.80943
## [10,] 46.39057 52.00674
## [11,] 47.54881 53.16498
## [12,] 46.96187 52.57804
```

```
## [13,] 47.05247 52.66864
## [14,] 48.44880 54.06498
## [15,] 49.56157 55.17774
## [16,] 46.95784 52.57401
## [17,] 47.49322 53.10939
## [18,] 46.74636 52.36253
## [19,] 47.36343 52.97960
## [20,] 46.31204 51.92821
## [21,] 45.92014 51.53631
## [22,] 48.43142 54.04759
## [23,] 46.97347 52.58964
## [24,] 48.99930 54.61547
## [25,] 48.81078 54.42695
```

```
 c_mu=sum(CI[,1]<=50&CI[,2]>=50)
 c_mu
```

```
## [1] 25
```

```
 c_mu=sum(CI[,1]<=53&CI[,2]>=53)
 c_mu
```

```
## [1] 11
```

```
 CI=matrix(nrow=25,ncol=2)
sample_means=numeric(25)
for (i in 1:25){
  x1=rnorm(20,50,6)
  sample_means[i]=mean(x1)
}
se=6/sqrt(100)
sample_means
```

```
##  [1] 48.34760 51.51303 50.60611 49.75584 48.50842 48.37536 48.79349 50.21736
##  [9] 49.02926 49.09521 50.38998 50.48453 52.43894 50.73219 49.13056 50.11246
## [17] 48.55058 51.84371 52.27873 50.02220 51.45695 50.74784 48.71139 48.08310
## [25] 50.42494
```

```
for (i in 1:25){
  me=qt(.975,df=99)*se
  lower=sample_means[i]-me
  upper=sample_means[i]+me
  CI[i,]=c(lower,upper)
}
CI
```

```
##           [,1]     [,2]
##  [1,] 47.15707 49.53813
##  [2,] 50.32250 52.70356
##  [3,] 49.41558 51.79664
##  [4,] 48.56531 50.94637
```

```
##  [5,] 47.31789 49.69895
##  [6,] 47.18483 49.56589
##  [7,] 47.60296 49.98402
##  [8,] 49.02683 51.40789
##  [9,] 47.83873 50.21979
## [10,] 47.90468 50.28574
## [11,] 49.19945 51.58051
## [12,] 49.29400 51.67507
## [13,] 51.24841 53.62947
## [14,] 49.54166 51.92272
## [15,] 47.94003 50.32109
## [16,] 48.92193 51.30299
## [17,] 47.36005 49.74111
## [18,] 50.65318 53.03424
## [19,] 51.08820 53.46926
## [20,] 48.83167 51.21273
## [21,] 50.26642 52.64748
## [22,] 49.55731 51.93837
## [23,] 47.52086 49.90192
## [24,] 46.89257 49.27363
## [25,] 49.23441 51.61547
```

```
c_mu=sum(CI[,1]<=50&CI[,2]>=50)
c_mu
```

```
## [1] 13
```

```
c_mu=sum(CI[,1]<=53&CI[,2]>=53)
c_mu
```

```
## [1] 3
```

Increasing the sample size decreases the size of the confidence interval.

Exercise 6.23 The *alpha*-risk of this rule is the probability of committing a Type I error, which is the probability of rejecting H0 when H0 is true. In this case, the α-risk is 5% as the test is performed at the 5% level of significance.

```
n <- 9
alpha <- 0.05
type1_errors_mu0 <- 0
type2_errors_mu0 <- 0
type1_errors_mu1 <- 0
type2_errors_mu1 <- 0
for (i in 1:100) {
  x <- rnorm(n, mean = 0, sd = 1)

  # Perform a one-sample t-test
  t_test_result_mu0 <- t.test(x, mu = 1, alternative = "greater")

  # Increment the counter for Type I errors if H0 is rejected
  if (t_test_result_mu0$p.value < alpha) {
    type1_errors_mu0 <- type1_errors_mu0 + 1
  }
  else {
    type2_errors_mu0 <- type2_errors_mu0 + 1
  }
  x <- rnorm(n, mean = 1, sd = 1)

  # Perform a one-sample t-test
  t_test_result_mu1 <- t.test(x, mu = 1, alternative = "greater")

  # Increment the counter for Type I errors if H0 is rejected
  if (t_test_result_mu1$p.value < alpha) {
    type1_errors_mu1 <- type1_errors_mu1 + 1
  }
  else {
    type2_errors_mu1 <- type2_errors_mu1 + 1
  }
}
prop_type1_errors_mu0 <- type1_errors_mu0 / 100
prop_type2_errors_mu1 <- type2_errors_mu1 / 100
prop_type1_errors_mu0
```

```
## [1] 0
```

```
prop_type2_errors_mu1
```

```
## [1] 0.94
```

Submission

Please submit through gradescope. Complete this assignment using R Markdown and converting to pdf. *Make sure each problem is on a separate page.* Use `\newpage` before each problem to insert a page break.

You can also refer to the .Rmd file for this assignment to have an example of this markdown document. Note however, that this version is setup to also use Latex for some math and conversion directly to pdf.

1. Stats 101!

This problem is to make sure everyone has the skill to use R and interpret a well defined statistical inference. Also it is real data and, if you live close to Boulder, might be interesting.

1(a)

Load the snowfall data from the text file `BoulderSnowfall.txt`. I am also creating a handy subset with just the annual fall/winter season snowfall for 1961 through 2020.

```
BoulderSnowfall<-
       read.table("BoulderSnowfall.txt", header=TRUE )
# note subscripts based on the names to make this easier to
# follow year is also column 14 in the data object.
year<-    BoulderSnowfall[ ,'Year']
snowAllYears<- BoulderSnowfall[, 'Sep.Jun']
snow<- snowAllYears[year >= 1961]
```

Create a 95% confidence interval (CI) for mean annual snowfall based on this subset. Do this "by hand", that is, use the R functions `mean`, `sd` and `qnorm` to get the intermediate computations and assemble.

Carefully interpret this CI in words.

```
sm=mean(snow)
n=length(snow)
ssd=sd(snow)
sse=ssd/sqrt(n)
alpha=.05
df=n-1
tscore=qt(p=alpha/2,df=df,lower.tail=FALSE)
error=tscore*sse
lower=sm - error
upper=sm+error
lower
```

```
## [1] 79.23243
```

```
upper
```

```
## [1] 93.17757
```

We are 95% confident that the tru emean will lie in this interval. ## 1(b) Here are two classic and common errors in interpreting a 95% confidence interval for the population mean (aka μ)

- We are 95% confident the interval contains 95% of the observations.
- We are 95% confident the interval contains the sample mean.

Explain why each of these is *not* correct. We are talking about the true mean. WSe are 95% confident the true mean lies in this interval. The sample mean will of course be in the interval.

1(c) Some details about data and R.

- What is the purpose of `header=TRUE` in the code that reads in the data? What is the class of the object 'BoulderSnowfall
- Explain why the useful text in the data file is ignored when the file is read.

Header true gets the data column labels.

```
class(BoulderSnowfall)
```

```
## [1] "data.frame"
```

2 Model checking

We know from the central limit theorem that the sample mean will follow a normal distribution with the expectd value being the population mean and we can estimate σ using the sample standard deviation. This justifies the CI in general terms from **1**. However, it is always a good idea to look at the data to check. For example weather data can include **999** as a missing value – obviously goofing up sample statistics if missed.

2(a)

Are there any missing observations in these annual snow totals?

```
sum(snow==999)
```

```
## [1] 0
```

```
sum(is.na(snow))
```

```
## [1] 0
```

2(b)

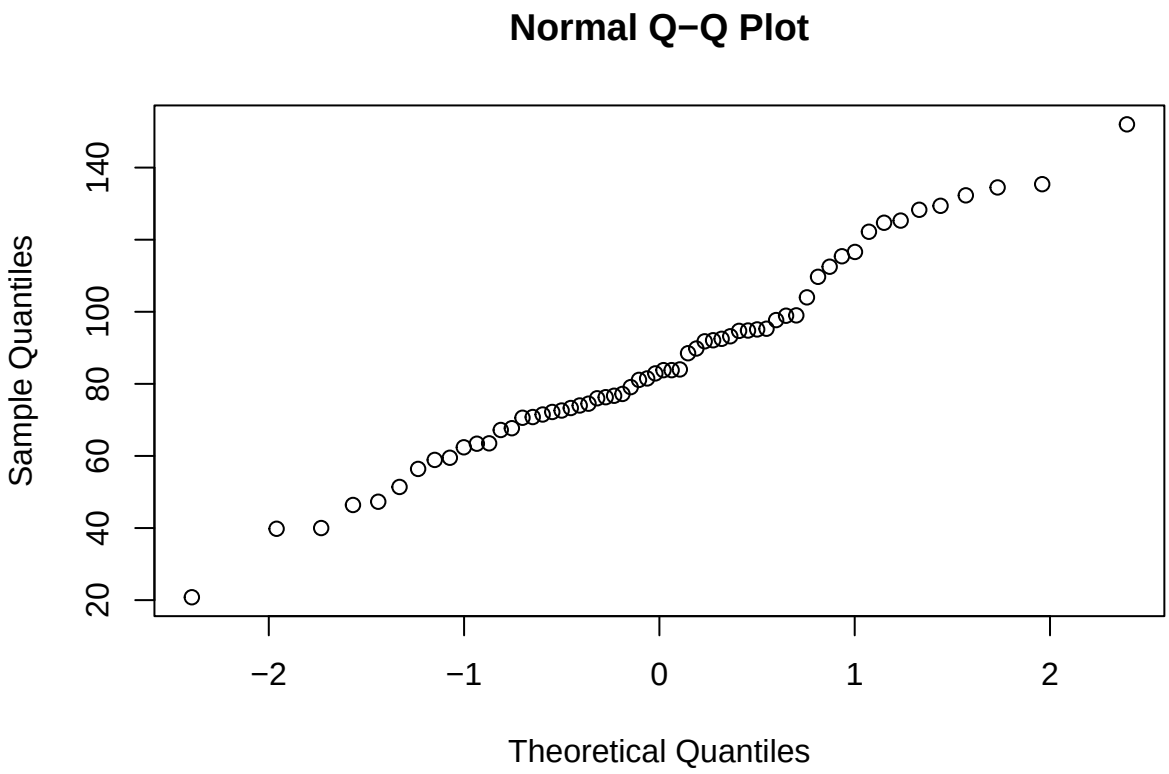
Create a stem and leaf plot (i.e. `stem( snow)`) of these data qnd comment on the shape of the distribution. (For a small data set this is better than making a histogram.)

```
stem(snow)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##    2 | 1
##    4 | 0067169
##    6 | 0234781122334566779
##    8 | 1234449022335555899
##   10 | 40357
##   12 | 25589255
##   14 | 2
```

The dtata is mostly clustered around 60 and 80 ## 2(c) Create a probability plot of the data, `qqnorm( snow)` and interpret this plot. Use wording and terminology so that someone without a statistics course can understand the point of this method.

```
qqnorm(snow)
```

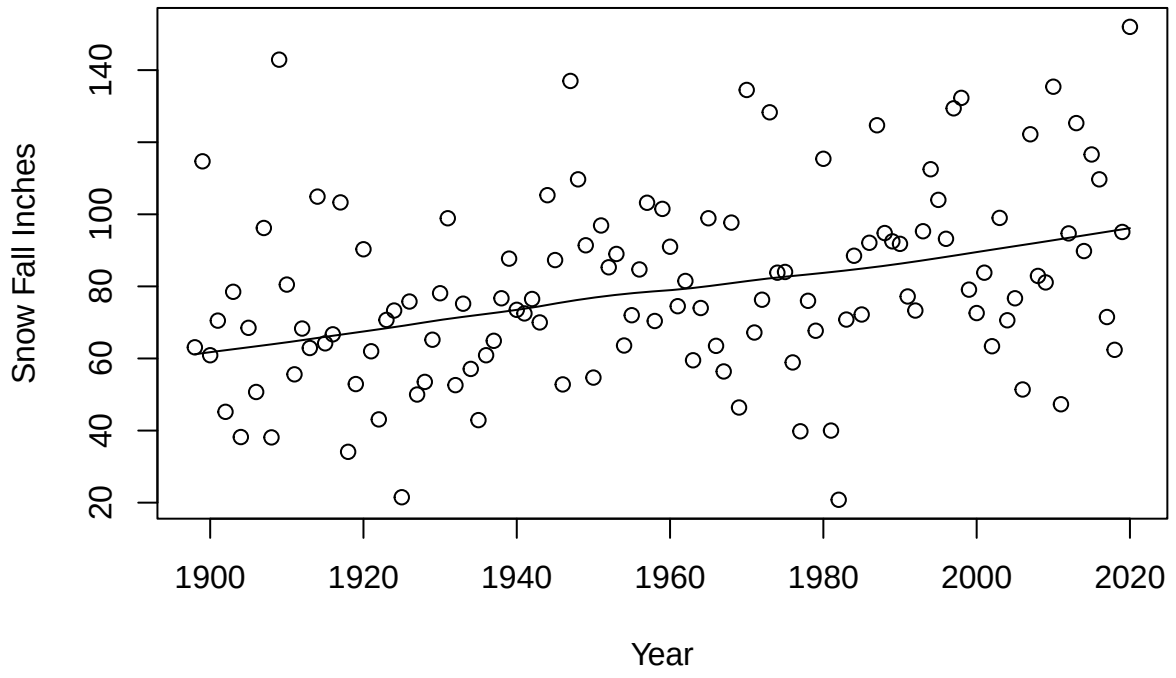


This data is mostly normally distributed. It contains no outliers.

3 Time series plot of the data.

This feature came as a surprise to me. Make a scatterplot of the full set of annual data against the year variable. When time is on the X axis this is called a times series plot. (Carefully label your plot and make it presentation quality.) Explain why inference for the population mean is not appropriate using the complete data set (1898- 2020).

```
scatter.smooth(year,snowAllYears,xlab="Year",ylab="Snow Fall Inches")
```



The data is not normal.

4 EXTRA CREDIT

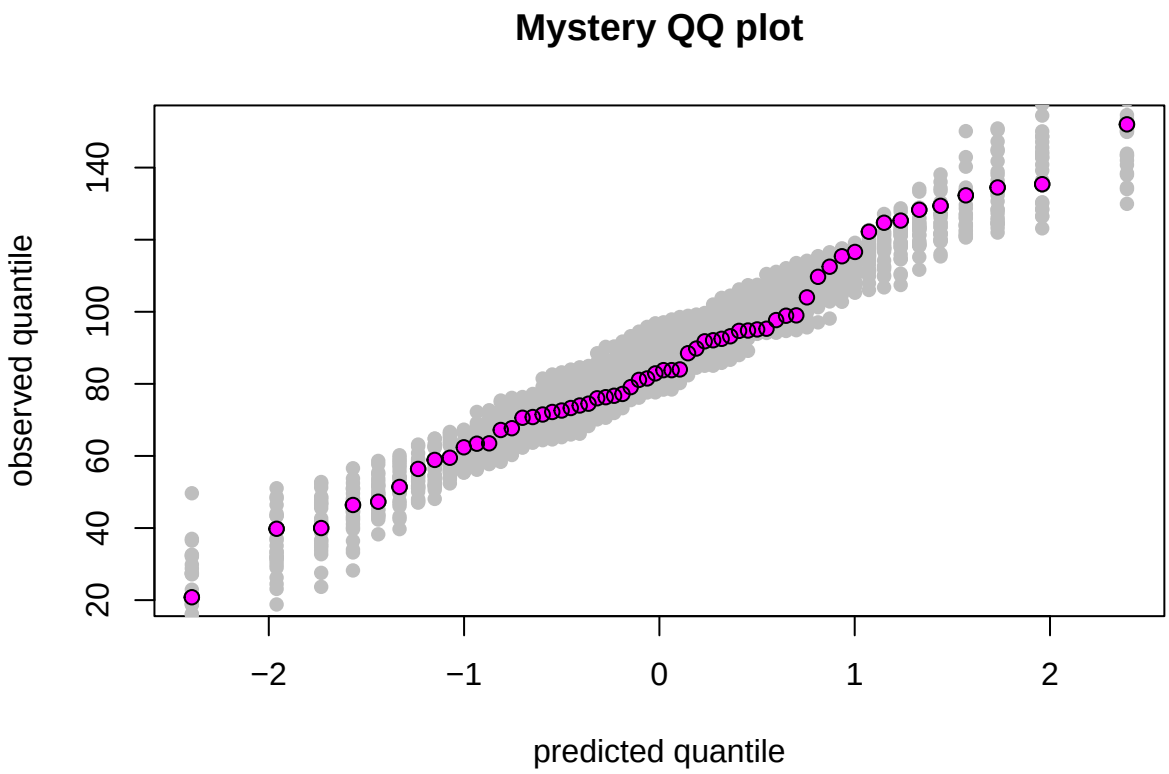
4(a)

The following code creates the QQ plot above "by hand" so that more stuff can be added to it. Explain the point of the **for** loop and how this figure should be interpreted.

```
N<- length( snow)
frac<- ((1:N) - .5)/N
Z<- sort( snow )
ZPerfect<- qnorm( frac)

plot( ZPerfect, Z, xlab="predicted quantile",
     ylab="observed quantile")

set.seed(333)
mu<- mean(snow)
sigma<- sd( snow)
for( k in 1:25){
  Zsim<- sort(
          rnorm( N, mean=mu, sd=sigma )
          )
  points(ZPerfect, Zsim, col="grey", pch=16 )
}
abline( 0,1, col="black", lwd=2)
points( ZPerfect, Z, col="magenta",pch=16 )
points( ZPerfect, Z ) # add a pleasing black outline to points
title("Mystery QQ plot")
```



The point of the for loop is to go over the data and sor ti tinto the normal distribution. This graph is to be interpreted by making inferences into the normality of the data.

4(b)

Within this problem is buried a hypothesis test! What "null hypothesis" (aka H_0) is being considered from a graphical/subjective point of view?

The hypotheis stest is whether the data is normally distributed with true mean equal to mu.