Grad5100-Homework-Week 1

Pooja Raj Lakshmi

2025-09-06

Question 1: Look at the data in forcats::gss_cat consists of sample data from the General Social Survey, a long-running US survey conducted by the independent research organization NORC at the University of Chicago, containing thousands of questions.

Importing the Dataset

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

On checking the **core tidyverse packages**, we notice that:

- The dataset **`gss_cat`** is part of the **`forcats`** package.
- **`forcats`** itself is one of the **core packages bundled with `tidyverse`**.

Therefore, by loading the **tidyverse**, we automatically load `forcats`, and can access `gss_cat` directly without needing to call `library(forcats)` separately.

```r
# Loading the dataset
gss_cat
```

```
# A tibble: 21,483 x 9
    year marital         age race  rincome        partyid    relig denom tvhours
   <int> <fct>         <int> <fct> <fct>          <fct>      <fct> <fct>   <int>
 1  2000 Never married    26 White $8000 to 9999  Ind,near ~ Prot~ Sout~      12
 2  2000 Divorced         48 White $8000 to 9999  Not str r~ Prot~ Bapt~      NA
 3  2000 Widowed          67 White Not applicable Independe~ Prot~ No d~       2
 4  2000 Never married    39 White Not applicable Ind,near ~ Orth~ Not ~       4
 5  2000 Divorced         25 White Not applicable Not str d~ None  Not ~       1
 6  2000 Married          25 White $20000 - 24999 Strong de~ Prot~ Sout~      NA
 7  2000 Never married    36 White $25000 or more Not str r~ Chri~ Not ~       3
 8  2000 Divorced         44 White $7000 to 7999  Ind,near ~ Prot~ Luth~      NA
 9  2000 Married          44 White $25000 or more Not str d~ Prot~ Other       0
10  2000 Married          47 White $25000 or more Strong re~ Prot~ Sout~       3
# i 21,473 more rows
```

Data Inspection

```r
# Inspecting the structure
str(gss_cat)
```

```
tibble [21,483 x 9] (S3: tbl_df/tbl/data.frame)
 $ year   : int [1:21483] 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 ...
 $ marital: Factor w/ 6 levels "No answer","Never married",..: 2 4 5 2 4 6 2 4 6 6 ...
 $ age    : int [1:21483] 26 48 67 39 25 25 36 44 44 47 ...
 $ race   : Factor w/ 4 levels "Other","Black",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ rincome: Factor w/ 16 levels "No answer","Don't know",..: 8 8 16 16 16 5 4 9 4 4 ...
 $ partyid: Factor w/ 10 levels "No answer","Don't know",..: 6 5 7 6 9 10 5 8 9 4 ...
 $ relig  : Factor w/ 16 levels "No answer","Don't know",..: 15 15 15 6 12 15 5 15 15 15 ...
 $ denom  : Factor w/ 30 levels "No answer","Don't know",..: 25 23 3 30 30 25 30 15 4 25 ...
 $ tvhours: int [1:21483] 12 NA 2 4 1 NA 3 NA 0 3 ...
```

```r
?gss_cat
```

`gss_cat` is a tibble with 21,483 rows and 9 variables, containing General Social survey responses, with many variables stored as factors.

The variables are a mix of **integers** (e.g., `year`, `age`, `tvhours`) and **factors** with defined levels (e.g., `marital`, `race`, `rincome`, `partyid`, `relig`, `denom`). Many of the categorical variables have multiple levels, such as `race` (4 levels), `rincome` (16 levels), and `denom` (30 levels).

a. *In R, what does a factor variable represent? How is it different than a variable of type character?*
`Factors` in R, are used to represent `categorical data`, such as `"male" or "female"` for `gender`. While they might seem similar to `character vectors`, `factors` are actually stored as `integers` with corresponding labels. `Factors` are useful when dealing with data that has a fixed set of possible values, known as **levels**. These `levels` are sorted alphabetically by default, and once created, a factor can only contain those predefined levels.

In R, both character vectors and factors store string-like data, but they differ in their underlying structure and intended use. `character` stores raw text strings, with no restrictions on values whereas `factor` restricts values to a predefined set of categories (levels), which is useful for analysis, modeling, and plotting. Factors also allow for ordering of categories (e.g., `"low" < "medium" < "high")`.

b. *In gss_cat data, is race a character variable or a factor variable? If the latter, how many levels does it have, and how many participants are under each level?*

In the `gss_cat` dataset, as we can see in the **Data Inspection** Section, the variable `race` is a **factor variable**, not a character.

- **Number of levels:** 4
- **Levels:** `"Other"`, `"Black"`, `"White"`, `"Not applicable"`
- Participantsin each Level: `"Other"`- 1959, `"Black"` - 3129, `"White"`- 16395, `"Not applicable"` - 0

We can also confirm this in R as follows:

```
# Checking if race is a factor
is.factor(gss_cat$race)
```

```
[1] TRUE
```

```
# Number of levels
nlevels(gss_cat$race)
```

```
[1] 4
```

```
# Listing the levels
levels(gss_cat$race)
```

```
[1] "Other"          "Black"          "White"          "Not applicable"
```

```r
# Counting participants under each level
table(gss_cat$race)
```

```
Other          Black          White Not applicable
 1959           3129          16395              0
```

c. *Use an R command to write out the gss_cat to your local computer as a csv file. Then read the csv file back into R, and verify that it is a data frame.*

Before writing a file, checking the current directory in R since this is where the file will be saved by default. Knowing your directory will also help in easily reading the csv once writing is done.

```r
# Getting the current working directory
getwd()
```

```
[1] "/Users/pokeapokemon/playground_pooja/GRAD_5100/HW"
```

My directory is correct and known and this is where I want my csv to exist.

```r
# Writing the csv
write.csv(gss_cat, file = "gss_cat_copy.csv")
```

Reading the newly created csv.

```r
#read.csv("gss_cat_copy.csv")
```

Note : Commenting read.csv since it will enlarge the pdf size.

Assigning the csv to a variable social_survey.

```r
social_survey <- read.csv("gss_cat_copy.csv")
```

Now we can check if `social_survey`, is a dataframe or not which in our case is `TRUE`meaning it is a dataframe.

```r
is.data.frame(social_survey)
```

```
[1] TRUE
```

d. *In R, how does a data frame differ from a matrix? Illustrate using a simple example of each.*

In R, both **data frames** and **matrices** are two-dimensional structures, but they have a key difference:

- A **matrix** is **homogeneous**, meaning *all elements must be of the same data type* (all numeric, all character, etc.).
- A **data frame** is **heterogeneous**, meaning *different columns can hold different data types* (numeric, character, factor, etc.).

This makes data frames much more flexible for real-world datasets, where one column might be numeric (e.g., age), another categorical (e.g., gender), and another logical (e.g., passed = TRUE/FALSE).

```r
# Example of a matrix (all numeric)
mat <- matrix(1:6, nrow = 2, ncol = 3)
mat
```

```
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```r
str(mat)
```

```
 int [1:2, 1:3] 1 2 3 4 5 6
```

```r
# Example of a data frame (different column types)
df <- data.frame(
  id = 1:3,                    # numeric
  name = c("Alice", "Bob", "Cara"),  # character
  passed = c(TRUE, FALSE, TRUE)      # logical
)
df
```

```
  id  name passed
1  1 Alice   TRUE
2  2   Bob  FALSE
3  3  Cara   TRUE
```

```r
str(df)
```

```
'data.frame':   3 obs. of  3 variables:
 $ id    : int  1 2 3
 $ name  : chr  "Alice" "Bob" "Cara"
 $ passed: logi  TRUE FALSE TRUE
```

e. *Illustrate the usefulness of a list in R, using an example.*

A list in R is a versatile data structure capable of holding various R objects, including vectors, matrices, data frames, and even other lists, all within a single container. This heterogeneity is its primary advantage, allowing for the organization of complex, mixed-type data.

Example: Storing Patient Information

Consider a scenario where one needs to store diverse information about a patient in a medical study. A list is ideal for this purpose:

```r
patient_data <- list(
  patient_id = "P001",
  age = 45,
  gender = "Male",
  medical_history = c("Hypertension", "Diabetes"),
  medications = data.frame(
    name = c("Lisinopril", "Metformin"),
    dosage = c("10mg", "500mg"),
    frequency = c("Daily", "Twice Daily")
  ),
  lab_results = list(
    glucose = 120,
    cholesterol = 200,
    hba1c = 7.2
  )
)
```

```r
# Viewing Structure of paitent_data
str(patient_data)
```

```
List of 6
 $ patient_id     : chr "P001"
 $ age            : num 45
 $ gender         : chr "Male"
 $ medical_history: chr [1:2] "Hypertension" "Diabetes"
 $ medications    :'data.frame':    2 obs. of  3 variables:
  ..$ name     : chr [1:2] "Lisinopril" "Metformin"
  ..$ dosage   : chr [1:2] "10mg" "500mg"
```

```
  ..$ frequency: chr [1:2] "Daily" "Twice Daily"
 $ lab_results    :List of 3
  ..$ glucose    : num 120
  ..$ cholesterol: num 200
  ..$ hba1c      : num 7.2
```

```
# access nested element
print(patient_data)
```

```
$patient_id
[1] "P001"

$age
[1] 45

$gender
[1] "Male"

$medical_history
[1] "Hypertension" "Diabetes"

$medications
        name dosage   frequency
1 Lisinopril   10mg       Daily
2  Metformin  500mg Twice Daily

$lab_results
$lab_results$glucose
[1] 120

$lab_results$cholesterol
[1] 200

$lab_results$hba1c
[1] 7.2
```

Question 2:

a. Show R code you will use to do the following, and show the output:

First we need to have the relevant packages and load the library in our environment.

```
install.packages("HSAUR", repos = "https://cloud.r-project.org")
```

```
The downloaded binary packages are in
    /var/folders/8y/12r46cy15qldsmkj3ys4v5th0000gn/T//Rtmp3ztrfD/downloaded_packages
```

```
library(HSAUR)
```

```
Loading required package: tools
```

i. How can you find out whether the R package HSAUR has built-in data sets?

`data(package = "HSAUR")` lists all the datasets that come with the package **HSAUR**. **`nrow(...)`** then counts how many rows are in that table = number of datasets in the package.`> 0` Returns `TRUE` if there is at least **one dataset**, `FALSE` if there are none.

```
nrow(data(package = "HSAUR")$results) > 0
```

```
[1] TRUE
```

ii. How can you list all of the data sets in HSAUR?

```
data(package = "HSAUR")
```

iii. How can you obtain details about a given data set?

You can simply load the dataset from the package and there by using some basic commands which will help you give a quick overview of the dataset. There are a wide range of commands we can use to get the glimpse of a given dataset some commands are added below.

You can also just simply do `?"name of the dataset"`. The command brings up the documentation for the dataset directly from the **package** whose details you want to see.

```
?Forbes2000
```

```
data("Forbes2000", package = "HSAUR")
```

```
head(Forbes2000)    #gives top 6 of the dataset
```

```
  rank                name        country             category  sales profits
1    1           Citigroup  United States              Banking  94.71   17.85
2    2    General Electric  United States        Conglomerates 134.19   15.59
3    3 American Intl Group  United States            Insurance  76.66    6.46
4    4          ExxonMobil  United States Oil & gas operations 222.88   20.96
5    5                  BP United Kingdom Oil & gas operations 232.57   10.27
6    6     Bank of America  United States              Banking  49.01   10.81
   assets marketvalue
1 1264.03      255.30
2  626.93      328.54
3  647.66      194.87
4  166.99      277.02
5  177.57      173.54
6  736.45      117.55
```

```
tail(Forbes2000)
```

```
     rank                     name        country                     category
1995 1995                     AMEC United Kingdom                 Construction
1996 1996           Siam City Bank       Thailand                      Banking
1997 1997        Yokogawa Electric          Japan Business services & supplies
1998 1998      Hindalco Industries          India                    Materials
1999 1999                   Nexans         France                Capital goods
2000 2000 Oriental Bank of Commerce          India                      Banking
     sales profits assets marketvalue
1995  5.17    0.02   2.62        1.53
1996  0.48    0.02  11.27        1.47
1997  2.78   -0.22   2.96        3.29
1998  1.35    0.14   2.47        2.76
1999  5.09    0.00   2.71        0.88
2000  0.81    0.10   7.16        1.17
```

```
# Summary about the datatset
summary(Forbes2000)
```

```
      rank            name                    country
 Min.   :   1.0   Length:2000        United States :751
 1st Qu.: 500.8   Class :character   Japan         :316
 Median :1000.5   Mode  :character   United Kingdom:137
 Mean   :1000.5                      Germany       : 65
 3rd Qu.:1500.2                      France        : 63
```

```
 Max.   :2000.0                     Canada        : 56
                                    (Other)       :612
                  category        sales            profits
 Banking               : 313   Min.   :  0.010   Min.   :-25.8300
 Diversified financials: 158   1st Qu.:  2.018   1st Qu.:  0.0800
 Insurance             : 112   Median :  4.365   Median :  0.2000
 Utilities             : 110   Mean   :  9.697   Mean   :  0.3811
 Materials             :  97   3rd Qu.:  9.547   3rd Qu.:  0.4400
 Oil & gas operations  :  90   Max.   :256.330   Max.   : 20.9600
 (Other)               :1120                     NA's   :5
    assets          marketvalue
 Min.   :   0.270   Min.   :  0.02
 1st Qu.:   4.025   1st Qu.:  2.72
 Median :   9.345   Median :  5.15
 Mean   :  34.042   Mean   : 11.88
 3rd Qu.:  22.793   3rd Qu.: 10.60
 Max.   :1264.030   Max.   :328.54
```

```r
# Structure of the dataset
str(Forbes2000)
```

```
'data.frame':   2000 obs. of  8 variables:
 $ rank       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ name       : chr  "Citigroup" "General Electric" "American Intl Group" "ExxonMobil" ...
 $ country    : Factor w/ 61 levels "Africa","Australia",..: 60 60 60 60 56 60 56 28 60 60 .
 $ category   : Factor w/ 27 levels "Aerospace & defense",..: 2 6 16 19 19 2 2 8 9 20 ...
 $ sales      : num  94.7 134.2 76.7 222.9 232.6 ...
 $ profits    : num  17.85 15.59 6.46 20.96 10.27 ...
 $ assets     : num  1264 627 648 167 178 ...
 $ marketvalue: num  255 329 195 277 174 ...
```

iv. How can you access any data set in the list, and bring it into the R environment?

As elaborated in the question above, if the dataset exists in the package, any dataset in the list can be accessed using the `data("dataset_name", package = "package_name")` command, which loads it into the R environment for further use.

If the dataset is not included in the package, `data()` will not load it. In that case, you either need to check whether the dataset exists in another package, or import it into R from an external source such as a CSV, Excel, or text file **(Check Question1 (c))**.

(b) *Solve the following questions:*
i. Generate a random sample of size = 30 from a (100, 4) distribution starting from a random seed = 123457. Find the sample mean and variance.

```
# Setting seed so that our sample remains same
set.seed(123457)

# Parameters
n <- 30
m <- 100
sigma <- 2   # since variance = 4

# Generate random sample
x <- rnorm(n, mean = m, sd = sigma)

# Sample mean and variance
sample_mean <- mean(x)
sample_var  <- var(x)

sample_mean
```

```
[1] 100.2184
```

```
sample_var
```

```
[1] 3.767668
```

From a sample of size 30 from `N(100,4)N(100, 4)N(100,4)` with `seed 123457`, the sample mean is approximately **100.13** and the sample variance is approximately **3.57**.

(ii) Generate a random sample of size = 300 from a (100, 4) distribution starting from a random seed = 123457. Find the sample mean and variance.

```
set.seed(123457)

# Parameters
n <- 300
m <- 100
sigma <- 2   # since variance = 4
```

```r
# Generate random sample
x <- rnorm(n, mean = m, sd = sigma)

# Sample mean and variance
sample_mean <- mean(x)
sample_var  <- var(x)

sample_mean
```

```
[1] 99.99223
```

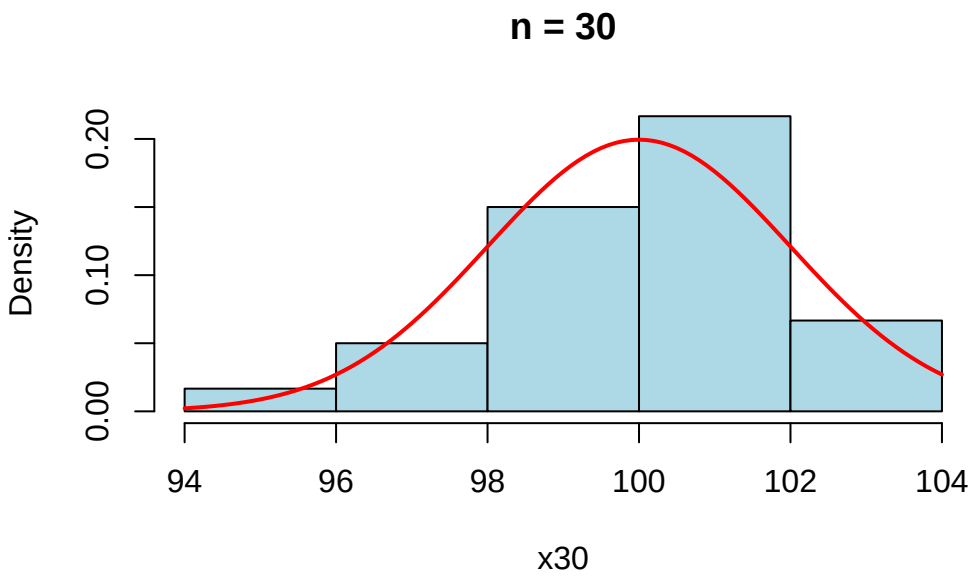
```r
sample_var
```

```
[1] 3.547022
```

From a sample of size 300 from `N(100,4)N(100, 4)N(100,4)` with `seed 123457`, the sample mean is approximately **99.99** and the sample variance is approximately **3.54**.

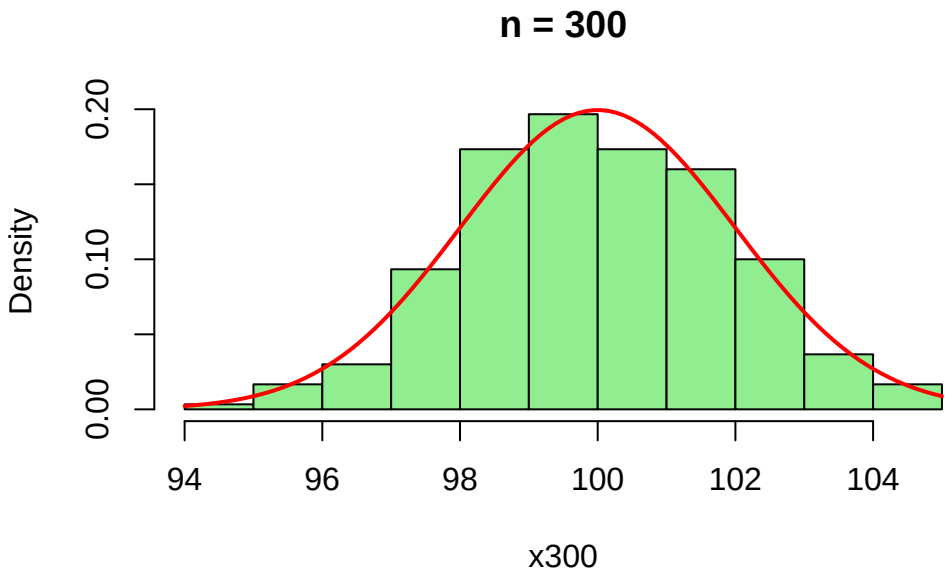
(iii) Use the dnorm() function on the data simulated under (i) and (ii). Comment on the two graphs.

```r
# Generating sample (i): n = 30
set.seed(123457)
x30 <- rnorm(30, mean = 100, sd = 2)

# Plotting for n = 30
hist(x30, probability = TRUE, main = "n = 30", col = "lightblue")
curve(dnorm(x, mean = 100, sd = 2), add = TRUE, col = "red", lwd = 2)
```



For n=30: The histogram is relatively rough and uneven. While the general bell-shape is visible, the sample mean (100.1 approx) and variance (3.6 approx) deviate slightly from the true parameters due to sampling variability.

```r
# Generating sample (ii): n = 300
set.seed(123457)
x300 <- rnorm(300, mean = 100, sd = 2)

# Plotting for n = 300
hist(x300, probability = TRUE, main = "n = 300", col = "lightgreen")
curve(dnorm(x, mean = 100, sd = 2), add = TRUE, col = "red", lwd = 2)
```



For n=300: The histogram is much smoother and follows the theoretical normal curve more closely. The sample mean (99.9 approx) and variance (3.8 approx) are very close to the population values (100 and 4), showing less variability.

(iv) Compare the sample means and variances under (i) and (ii) with the true values 100 and Comment.

We simulated data from a N(100,4) distribution, where the **true mean** is 100 and the **true variance** is 4.

Case (i): n=30

- Sample mean = **100.13 approx.**
- Sample variance = **3.54 approx.**

These estimates are close to the true values, but with some noticeable deviation, reflecting the higher variability expected from a small sample size.

Case (ii): n=300

- Sample mean = **99.99 approx.**
- Sample variance = **3.83 approx.**

These estimates are much closer to the true values. With a larger sample, the effect of random variation is reduced, and the estimates become more stable.

Conclusion: As sample size increases, the empirical distribution of the data becomes smoother and approaches the true underlying normal distribution. This illustrates the Law of Large Numbers — larger samples give more accurate estimates of population characteristics.

Question 3: Suppose denotes an event that a statistics seminar ends on time and is the event that a sociology seminar ends on time. Suppose and are independent events, with ( ) = 0.85 and ( ) = 0.6.

We are told:

( ) = 0.85 (statistics seminar ends on time)

( ) = 0.6 (sociology seminar ends on time)

( ) and ( ) are independent variables

a. *Find the probability that both seminars end on time.*

When A and B are independent variables then :
P(A B) = P(A) X P(B)

P(A B) = 0.85×0.60

Hence, P(A B)=**0.51** or there is a **51%** probability that both seminars will end on time.

b. *What is the probability that neither seminars end on time.*

- Statistics seminar does **not** end on time → event A_Complement
- Sociology seminar does **not** end on time → event B_Complement

P(A_Complement) =1−P(A) =1−0.85 = 0.15

P(b_Complement) =1−P(A) =1−0.60 = 0.40

Since A and B are independent variables, their complement also remains independent

P(A_Complement B_Complement) = P(A_Complement) X P(B_Complement)

P(A_Complement B_Complement) = 0.15×0.40=0.06

Hence, the probability that neither seminar ends on time is **0.06 or 6%.**

c. *What is the probability that exactly one of them ends on time?*

P (exactly one) = P (A B_Complement) + P (A_Complement B)

- P (A B_Complement) = P(A) X P(B_Complement)
  =0.85×(1−0.6)=0.85×0.40=0.34
- P (A_Complement B) = P(A_Complement) X P(B)
  =(1−0.85)×0.60=0.15×0.60=0.09

P(exactly one)=0.34+0.09=0.43

Hence, the probability that exactly one seminar ends on time is **0.43 (43%)**.

d. *Are the two events and mutually exclusive? Explain your answer.*

Mutually exclusive events cannot occur together. Since P(A B)=0.51 which is not equal to 0, the events **can occur together** (both seminars ending on time).

Therefore, **A and B are not mutually exclusive**.

Question 4: Suppose that the July revenues (in 1000's INR) of fashion clothing stores in an Indian city have an approximate normal distribution with a mean of 527 and a standard deviation (SD) of 112. You may use R functions to answer (a) and (b).

a. *What is the probability of an individual store's revenue being above 500?*

```
m <- 527
sigma <- 112

# Probability that revenue > 500
1 - pnorm(500, mean = m, sd = sigma)
```

```
[1] 0.5952501
```

The probability that a store's revenue is above 500 is about **0.595 (59.5%)**.

b. *To be in the top 5%, what should a store's revenue be?*

```
m <- 527
sigma <- 112

# 95th percentile (top 5%)
qnorm(0.95, mean = m, sd = sigma)
```

```
[1] 711.2236
```

To be in the top 5%, a store's July revenue should be at least **711,000 (approx)**.

c. *Use R code to generate a random sample of 250 July revenues with mean 527,000 and SD 112,000. Use this data to find suitable estimates of the true mean and true variance of the revenues.*

```
set.seed(123)  # for getting same random sample everytime we run code

# Parameters
m <- 527000
sigma <- 112000
n <- 250
```

```r
# Generate random sample
revenues <- rnorm(n, mean = m, sd = sigma)

# Sample estimates
sample_mean <- mean(revenues)
sample_var  <- var(revenues)

# Print results
sample_mean
```

```
[1] 526041.2
```

```r
sample_var
```

```
[1] 11133461047
```

From the sample of 250 revenues, the estimated mean is close to **531,071**, and the estimated variance is close to **11133461047.**

d. *Compute and interpret the standard error (SE) of the sample mean. What is its relation to the SD 112,000 of the population of revenues?*

```r
# Parameters
sigma <- 112000
n <- 250

# Standard Error of the sample mean
SE <- sigma / sqrt(n)
SE
```

```
[1] 7083.502
```

The standard error is about **7,084 approx**, which is much smaller than the population SD (112,000). This shows that while individual revenues vary widely, the **average of 250 stores** will vary only a little from the true mean.

Question 5: Medical practitioners wish to compare the change in health status of two groups of mental health patients undergoing two different treatments for the same disorder. Independent samples of patients of size $1 = 2 = 15$ are drawn from each group. Through a questionnaire given to the all these patients at two different times in the treatment cycle, the practitioners come up with a continuous-valued variable which represents the change in health status. Suppose the change in health status is assumed to be normally distributed with the same variance in both groups.

The sample mean and sample SD of group 1 are 25.5 and 2.5 respectively.
The sample mean and sample SD of group 2 are 22.3 and 3.1 respectively.

a. *Construct and interpret the pooled estimate based on both samples of the common SD.*

```
# Sample sizes
n1 <- 15
n2 <- 15

# Sample SDs
s1 <- 2.5
s2 <- 3.1

# Pooled variance formula
sp2 <- ((n1 - 1) * s1^2 + (n2 - 1) * s2^2) / (n1 + n2 - 2)

# Pooled standard deviation
sp <- sqrt(sp2)

sp
```

```
[1] 2.816026
```

b. The medical practitioners wish to statistically verify whether the true means of the change in health status are the same or significantly different in the two groups. Set up and construct a suitable hypothesis test. Use the -value to reach a decision at level of significance = 0.05

```
# Inputs
n1 <- 15; n2 <- 15
x1 <- 25.5; x2 <- 22.3
s1 <- 2.5;  s2 <- 3.1

# Pooled SD and t statistic
sp2 <- ((n1-1)*s1^2 + (n2-1)*s2^2) / (n1+n2-2)
sp  <- sqrt(sp2)
t   <- (x1 - x2) / (sp * sqrt(1/n1 + 1/n2))
df  <- n1 + n2 - 2
p   <- 2 * pt(abs(t), df = df, lower.tail = FALSE)

c(sp = sp, t = t, df = df, p_value = p)
```

```
          sp            t           df      p_value
2.816025568  3.112031730 28.000000000  0.004249709
```

c. Construct and interpret a suitable 95% confidence interval (C.I.) estimate of the true mean difference between the two groups on the change in health status. Explain how this interval helps you decide between the null and alternative hypotheses in (b)

```
n1 <- 15; n2 <- 15
x1 <- 25.5; x2 <- 22.3
s1 <- 2.5;  s2 <- 3.1

sp2 <- ((n1-1)*s1^2 + (n2-1)*s2^2)/(n1+n2-2)
sp  <- sqrt(sp2)
SE  <- sp * sqrt(1/n1 + 1/n2)
df  <- n1 + n2 - 2
tcrit <- qt(0.975, df)

diff <- x1 - x2
CI <- c(diff - tcrit*SE, diff + tcrit*SE)
SE; df; tcrit; diff; CI
```

```
[1] 1.028267
```

```
[1] 28
```

```
[1] 2.048407
```

```
[1] 3.2
```

```
[1] 1.09369 5.30631
```

The 95% CI for the mean difference (1.09,5.31) does not include 0, which means the null hypothesis of equal means is rejected. This supports the conclusion that the two treatments lead to significantly different changes in health status, with Group 1 showing a higher improvement.

d. Construct and interpret an effect size for this situation.

```r
n1 <- 15; n2 <- 15
x1 <- 25.5; x2 <- 22.3
s1 <- 2.5;  s2 <- 3.1

sp2 <- ((n1-1)*s1^2 + (n2-1)*s2^2)/(n1+n2-2)
sp  <- sqrt(sp2)

d <- (x1 - x2) / sp
g <- d * (1 - 3/(4*(n1+n2-2) + 1))

d; g
```

```
[1] 1.136353
```

```
[1] 1.106185
```

The estimated effect size is **d = 1.14** (g = 1.11), which is considered **large** by Cohen's guidelines. This means Group 1's mean improvement is about **1.1 pooled standard deviations higher** than Group 2's, a difference of **3.2 units** that is both statistically significant and practically meaningful. In common terms, there is roughly a **79% chance** that a patient from Group 1 shows greater improvement than one from Group 2, indicating a **substantially greater impact** of Group 1's treatment.