U7514: Assignment 1

Arnav Sahai (as7483)

2026-01-27

Please submit your knitted .pdf file along with the corresponding R markdown (.rmd) via Courseworks by 11:59pm on the due date.

Before knitting your rmd file as a pdf, you will need to install TinyTex for Latex distribution by running the following code:

tinytex::install_tinytex() install.packages('DescTools') library(DescTools)

Please visit this link for more information on TinyTex installation.

```
# ============================================================================
# SETUP AND DATA LOADING
# ============================================================================

setwd("/Users/arnavsahai/Desktop/Data Analysis for Policy Research using R/Lecture 1")
getwd()
```

```
## [1] "/Users/arnavsahai/Desktop/Data Analysis for Policy Research using R/Lecture 1"
```

```
# Load required libraries
library(dplyr)
library(DescTools)

# Load and inspect ACCESS 2018 data:
load("access2018.RData")
str(access2018)
```

```
## tibble [9,008 x 10] (S3: tbl_df/tbl/data.frame)
##  $ HHID        : chr [1:9008] "236582/10" "222500/10" "231781/11" "257562/4" ...
##   ..- attr(*, "label")= chr "Household ID"
##   ..- attr(*, "format.stata")= chr "%10s"
##  $ date        : chr [1:9008] "15-May-18" "27-May-18" "Mar 18, 2018" "Apr 23, 2018" ...
##   ..- attr(*, "label")= chr "Date of interview"
##   ..- attr(*, "format.stata")= chr "%12s"
##  $ state       : chr [1:9008] "BIHAR" "BIHAR" "BIHAR" "BIHAR" ...
##   ..- attr(*, "label")= chr "State"
##   ..- attr(*, "format.stata")= chr "%14s"
##  $ relationship: 'haven_labelled' num [1:9008] 2 5 1 1 1 4 4 1 1 1 ...
##   ..- attr(*, "label")= chr "Relationship to the household head"
##   ..- attr(*, "format.stata")= chr "%13.0g"
##   ..- attr(*, "labels")= Named num [1:5] 1 2 3 4 5
```

```
##   .. ..- attr(*, "names")= chr [1:5] "Self" "Spouse" "Mother/Father" "Son/Daughter" ...
##  $ age          : num [1:9008] 35 43 65 70 57 27 32 60 27 58 ...
##   ..- attr(*, "label")= chr "Age (years)"
##   ..- attr(*, "format.stata")= chr "%10.0g"
##  $ gender       : chr [1:9008] "Female" "Male" "Male" "Male" ...
##  $ education    : chr [1:9008] "Grade 5" "Grade 5" "Grade 5" "Grade 12" ...
##  $ caste        : chr [1:9008] "Other Backward Class" "Other Backward Class" "General" "General" ...
##  $ incsource    : chr [1:9008] "Daily labourer" "Agriculture (own land)" "Agriculture (own land)" "Ag
##  $ month_exp    : num [1:9008] 7000 9000 6000 5000 16000 10000 8000 10000 5000 15000 ...
##   ..- attr(*, "label")= chr "Monthly expenditure (rupee/month)"
##   ..- attr(*, "format.stata")= chr "%10.0g"
##  - attr(*, "na.action")= 'omit' Named int [1:64] 89 1382 1663 2417 2424 2442 2598 2686 2765 2794 ...
##   ..- attr(*, "names")= chr [1:64] "89" "1382" "1663" "2417" ...
```

```
# ========================================================================
# SECTION 1: Load and inspect ACCESS 2018 data
# ========================================================================

# 1a) Inspect the data frame and state the data types for each column
# Inspect overall structure
str(access2018)
```

```
## tibble [9,008 x 10] (S3: tbl_df/tbl/data.frame)
##  $ HHID         : chr [1:9008] "236582/10" "222500/10" "231781/11" "257562/4" ...
##   ..- attr(*, "label")= chr "Household ID"
##   ..- attr(*, "format.stata")= chr "%10s"
##  $ date         : chr [1:9008] "15-May-18" "27-May-18" "Mar 18, 2018" "Apr 23, 2018" ...
##   ..- attr(*, "label")= chr "Date of interview"
##   ..- attr(*, "format.stata")= chr "%12s"
##  $ state        : chr [1:9008] "BIHAR" "BIHAR" "BIHAR" "BIHAR" ...
##   ..- attr(*, "label")= chr "State"
##   ..- attr(*, "format.stata")= chr "%14s"
##  $ relationship: 'haven_labelled' num [1:9008] 2 5 1 1 1 4 4 1 1 1 ...
##   ..- attr(*, "label")= chr "Relationship to the household head"
##   ..- attr(*, "format.stata")= chr "%13.0g"
##   ..- attr(*, "labels")= Named num [1:5] 1 2 3 4 5
##   .. ..- attr(*, "names")= chr [1:5] "Self" "Spouse" "Mother/Father" "Son/Daughter" ...
##  $ age          : num [1:9008] 35 43 65 70 57 27 32 60 27 58 ...
##   ..- attr(*, "label")= chr "Age (years)"
##   ..- attr(*, "format.stata")= chr "%10.0g"
##  $ gender       : chr [1:9008] "Female" "Male" "Male" "Male" ...
##  $ education    : chr [1:9008] "Grade 5" "Grade 5" "Grade 5" "Grade 12" ...
##  $ caste        : chr [1:9008] "Other Backward Class" "Other Backward Class" "General" "General" ...
##  $ incsource    : chr [1:9008] "Daily labourer" "Agriculture (own land)" "Agriculture (own land)" "Ag
##  $ month_exp    : num [1:9008] 7000 9000 6000 5000 16000 10000 8000 10000 5000 15000 ...
##   ..- attr(*, "label")= chr "Monthly expenditure (rupee/month)"
##   ..- attr(*, "format.stata")= chr "%10.0g"
##  - attr(*, "na.action")= 'omit' Named int [1:64] 89 1382 1663 2417 2424 2442 2598 2686 2765 2794 ...
##   ..- attr(*, "names")= chr [1:64] "89" "1382" "1663" "2417" ...
```

```
# Inspect specific columns mentioned
summary(access2018$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   18.00   32.00   42.00   43.29   55.00   98.00
```

```
summary(access2018$gender)
```

```
##    Length     Class      Mode
##      9008 character character
```

```
summary(access2018$caste)
```

```
##    Length     Class      Mode
##      9008 character character
```

```
summary(access2018$education)
```

```
##    Length     Class      Mode
##      9008 character character
```

```
summary(access2018$state)
```

```
##    Length     Class      Mode
##      9008 character character
```

```
summary(access2018$incsource)
```

```
##    Length     Class      Mode
##      9008 character character
```

```
# Check data types
sapply(access2018[c("age", "gender", "caste", "incsource", "state", "education")], class)
```

```
##         age      gender       caste   incsource       state   education
##   "numeric" "character" "character" "character" "character" "character"
```

```
# 1b) Use the mutate function to create new column for gender
# Create gender.fac column
access2018 <- access2018 %>% mutate(gender.fac = as.factor(gender))

# Validate it worked
str(access2018$gender.fac)
```

```
##  Factor w/ 2 levels "Female","Male": 1 2 2 2 2 2 2 1 2 2 ...
```

```
summary(access2018$gender.fac)
```

```
## Female   Male
##   2570   6438
```

```r
class(access2018$gender.fac)
```

```
## [1] "factor"
```

```r
# 1c) In a single pipe, include gender.fac in a new data frame called access2018.temp1
access2018.temp1 <- access2018 %>%
    mutate(gender.fac = as.factor(gender),
           caste.fac = as.factor(caste),
           education.fac = as.factor(education),
           state.fac = as.factor(state),
           incsource.fac = as.factor(incsource)) %>%
    select(-HHID, -date)

# Print first 5 observations
head(access2018.temp1, n = 5)
```

```
## # A tibble: 5 x 13
##   state relationship   age gender education caste incsource month_exp gender.fac
##   <chr> <dbl+lbl>    <dbl> <chr>  <chr>     <chr> <chr>         <dbl> <fct>
## 1 BIHAR 2 [Spouse]      35 Female Grade 5   Othe~ Daily la~      7000 Female
## 2 BIHAR 5 [Others]      43 Male   Grade 5   Othe~ Agricult~      9000 Male
## 3 BIHAR 1 [Self]        65 Male   Grade 5   Gene~ Agricult~      6000 Male
## 4 BIHAR 1 [Self]        70 Male   Grade 12  Gene~ Agricult~      5000 Male
## 5 BIHAR 1 [Self]        57 Male   Grade 10  Gene~ Agricult~     16000 Male
## # i 4 more variables: caste.fac <fct>, education.fac <fct>, state.fac <fct>,
## #   incsource.fac <fct>
```

```r
# 1d) Inspect caste.fac, gender.fac, incsource.fac, education.fac and state.fac using the levels() func
levels(access2018.temp1$gender.fac)
```

```
## [1] "Female" "Male"
```

```r
levels(access2018.temp1$caste.fac)
```

```
## [1] "General"              "Other Backward Class" "Others"
## [4] "Scheduled Caste"      "Scheduled Tribe"
```

```r
levels(access2018.temp1$education.fac)
```

```
## [1] "Grade 10"          "Grade 12"          "Grade 5"
## [4] "Graduate or above" "No School "
```

```r
levels(access2018.temp1$state.fac)
```

```
## [1] "BIHAR"          "JHARKHAND"      "MADHYA PRADESH" "ODISHA"
## [5] "UTTAR PRADESH"  "WEST BENGAL"
```

```r
levels(access2018.temp1$incsource.fac)
```

```
## [1] "Agriculture (own land)" "Casual agri labour"     "Cattle rearing"
## [4] "Cultivation (leased)"   "Daily labourer"         "Other"
## [7] "Own business"           "Salaried job"
```

```r
# The levels() function is located in base R (no package needed)

# 1e) Use filter() to only include rows only for the state "Uttar Pradesh."
access_UP <- access2018.temp1 %>%
    filter(state.fac == "UTTAR PRADESH")

# Print first 5 observations
head(access_UP, n = 5)
```

```
## # A tibble: 5 x 13
##   state relationship   age gender education caste incsource month_exp gender.fac
##   <chr> <dbl+lbl>    <dbl> <chr>  <chr>     <chr> <chr>         <dbl> <fct>
## 1 UTTA~ 1 [Self]        70 Female "No Scho~ Othe~ Daily la~      4000 Female
## 2 UTTA~ 4 [Son/Daug~    30 Male   "No Scho~ Sche~ Daily la~      3000 Male
## 3 UTTA~ 1 [Self]        35 Male   "No Scho~ Sche~ Daily la~      2500 Male
## 4 UTTA~ 1 [Self]        72 Male   "Grade 1~ Othe~ Agricult~      3000 Male
## 5 UTTA~ 4 [Son/Daug~    21 Male   "Graduat~ Gene~ Cultivat~      8000 Male
## # i 4 more variables: caste.fac <fct>, education.fac <fct>, state.fac <fct>,
## #   incsource.fac <fct>
```

```r
# Confirm data only includes Uttar Pradesh
summary(access_UP$state.fac)
```

```
##          BIHAR      JHARKHAND MADHYA PRADESH         ODISHA  UTTAR PRADESH
##              0              0              0              0           3002
##    WEST BENGAL
##              0
```

```r
unique(access_UP$state.fac)
```

```
## [1] UTTAR PRADESH
## Levels: BIHAR JHARKHAND MADHYA PRADESH ODISHA UTTAR PRADESH WEST BENGAL
```

```r
# 1f) What is the female share of respondents in Uttar Pradesh? Compare it to the female share of respo
# Female share in Uttar Pradesh
up_gender_table <- table(access_UP$gender.fac)
up_gender_prop <- prop.table(up_gender_table)
print("Uttar Pradesh gender distribution:")
```

```
## [1] "Uttar Pradesh gender distribution:"
```

```r
print(up_gender_table)
```

```
##
## Female   Male
##   1011   1991
```

```
print("Uttar Pradesh gender proportions:")
```

```
## [1] "Uttar Pradesh gender proportions:"
```

```
round(up_gender_prop, 3)
```

```
##
## Female   Male
##  0.337  0.663
```

```
# Female share in Bihar
access_Bihar <- access2018.temp1 %>%
    filter(state.fac == "BIHAR")

bihar_gender_table <- table(access_Bihar$gender.fac)
bihar_gender_prop <- prop.table(bihar_gender_table)
print("Bihar gender distribution:")
```

```
## [1] "Bihar gender distribution:"
```

```
print(bihar_gender_table)
```

```
##
## Female   Male
##    352   1158
```

```
print("Bihar gender proportions:")
```

```
## [1] "Bihar gender proportions:"
```

```
round(bihar_gender_prop, 3)
```

```
##
## Female   Male
##  0.233  0.767
```

```
# Compare female shares
print(paste("Female share in UP:", round(up_gender_prop["Female"], 3)))
```

```
## [1] "Female share in UP: 0.337"
```

```
print(paste("Female share in Bihar:", round(bihar_gender_prop["Female"], 3)))
```

```
## [1] "Female share in Bihar: 0.233"
```

```r
# 1g) Remove the access2018.temp1 object from memory using the rm() function
rm(access2018.temp1)
# Verify it's removed
exists("access2018.temp1")
```

```
## [1] FALSE
```

```r
# ============================================================================
# SECTION 2: Describe the access_UP data frame
# ============================================================================

# 2a) What is the unit of observation (or unit of analysis)?
# The unit of observation is the individual person/respondent
# Each row represents one individual respondent

# 2b) How many individuals are observed?
# Total number of individuals
nrow(access_UP)
```

```
## [1] 3002
```

```r
# Caste distribution using table
caste_table <- table(access_UP$caste.fac)
print("Caste distribution (counts):")
```

```
## [1] "Caste distribution (counts):"
```

```r
print(caste_table)
```

```
##
##              General Other Backward Class               Others
##                  708                 1603                    0
##      Scheduled Caste      Scheduled Tribe
##                  657                   34
```

```r
# Caste proportions
caste_prop <- prop.table(caste_table)
print("Caste distribution (proportions):")
```

```
## [1] "Caste distribution (proportions):"
```

```r
round(caste_prop, 3)
```

```
##
##              General Other Backward Class               Others
##                0.236                0.534                0.000
##      Scheduled Caste      Scheduled Tribe
##                0.219                0.011
```

```r
# Count of Reserved castes (Scheduled Caste, Scheduled Tribe, Other Backward Class)
reserved_castes <- c("Scheduled Caste", "Scheduled Tribe", "Other Backward Class")
reserved_count <- sum(access_UP$caste.fac %in% reserved_castes)
print(paste("Number of Reserved caste individuals:", reserved_count))
```

```
## [1] "Number of Reserved caste individuals: 2294"
```

```r
# Count of General caste
general_count <- sum(access_UP$caste.fac == "General")
print(paste("Number of General caste individuals:", general_count))
```

```
## [1] "Number of General caste individuals: 708"
```

```r
# 2c) Do you think the sample is representative of all households in India?
# This sample is limited to Uttar Pradesh only, so it cannot be representative of all households in Ind
# The population represented is individuals/households in Uttar Pradesh

# To assess representativeness, we would want to see:
# - Sampling methodology
# - Response rates
# - Comparison with census data for UP
# - Geographic distribution within UP
# - Urban vs rural breakdown

# 2d) What is the average age of individuals in the sample? youngest and oldest person?
# Average age
mean_age <- mean(access_UP$age, na.rm = TRUE)
print(paste("Average age:", round(mean_age, 2)))
```

```
## [1] "Average age: 43.44"
```

```r
# Youngest person
min_age <- min(access_UP$age, na.rm = TRUE)
print(paste("Youngest person's age:", min_age))
```

```
## [1] "Youngest person's age: 18"
```

```r
# Oldest person
max_age <- max(access_UP$age, na.rm = TRUE)
print(paste("Oldest person's age:", max_age))
```

```
## [1] "Oldest person's age: 98"
```

```r
# Summary statistics
summary(access_UP$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   30.00   42.00   43.44   55.00   98.00
```

```r
# ============================================================================
# SECTION 3: Earnings per week for different groups in Uttar Pradesh
# ============================================================================

# 3a) Find the observation for the top monthly expenditure using the summarize() function
max_exp_obs1 <- access_UP %>%
  summarise(max_month_exp = max(month_exp, na.rm = TRUE))
print(max_exp_obs1)
```

```
## # A tibble: 1 x 1
##   max_month_exp
##           <dbl>
## 1         56000
```

```r
# 3b) Find max monthly month_exps using the arrange function instead of summarize
# Using arrange to find max month_exp
access_UP %>%
  arrange(desc(month_exp)) %>%
  head(1) %>%
  select(month_exp)
```

```
## # A tibble: 1 x 1
##   month_exp
##       <dbl>
## 1     56000
```

```r
# 3c) Use the filter function to subset for the observation with max monthly month_exp
max_exp_obs2 <- access_UP %>%
  filter(month_exp == max_exp_obs1$max_month_exp)

# Confirm it worked
print("Observation with max month_exp:")
```

```
## [1] "Observation with max month_exp:"
```

```r
print(max_exp_obs2)
```

```
## # A tibble: 1 x 13
##   state relationship   age gender education caste incsource month_exp gender.fac
##   <chr> <dbl+lbl>    <dbl> <chr>  <chr>     <chr> <chr>         <dbl> <fct>
## 1 UTTA~ 1 [Self]        43 Male   Graduate~ Othe~ Salaried~     56000 Male
## # i 4 more variables: caste.fac <fct>, education.fac <fct>, state.fac <fct>,
## #   incsource.fac <fct>
```

```r
nrow(max_exp_obs2)  # Should be 1 (or more if there are ties)
```

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

```r
# 3d) What is the age, gender and caste of the top monthly spender in the sample?
max_exp_obs2 %>%
  select(age, gender.fac, caste.fac, month_exp)
```

```
## # A tibble: 1 x 4
##     age gender.fac caste.fac            month_exp
##   <dbl> <fct>      <fct>                    <dbl>
## 1    43 Male       Other Backward Class     56000
```

```r
# 3e) List the age, gender and caste of the top 10 monthly spenders in the sample
top_10_spenders <- access_UP %>%
  arrange(desc(month_exp)) %>%
  head(10) %>%
  select(age, gender.fac, caste.fac, month_exp)

print(top_10_spenders)
```

```
## # A tibble: 10 x 4
##      age gender.fac caste.fac            month_exp
##    <dbl> <fct>      <fct>                    <dbl>
##  1    43 Male       Other Backward Class     56000
##  2    58 Male       Other Backward Class     50000
##  3    66 Male       Other Backward Class     50000
##  4    24 Male       Other Backward Class     50000
##  5    28 Male       Other Backward Class     50000
##  6    18 Male       General                  40000
##  7    65 Male       General                  40000
##  8    23 Female     Other Backward Class     40000
##  9    42 Male       Other Backward Class     40000
## 10    20 Male       Other Backward Class     32000
```

```r
# 3f) How many individuals spend more than the mean monthly month_exp amount of the sample?
mean_month_exp <- mean(access_UP$month_exp, na.rm = TRUE)
print(paste("Mean monthly month_exp:", round(mean_month_exp, 2)))
```

```
## [1] "Mean monthly month_exp: 6802.15"
```

```r
above_mean <- access_UP %>%
  filter(month_exp > mean_month_exp)

num_above_mean <- nrow(above_mean)
print(paste("Number of individuals spending more than mean:", num_above_mean))
```

```
## [1] "Number of individuals spending more than mean: 1115"
```

```r
print(paste("Proportion:", round(num_above_mean / nrow(access_UP), 3)))
```

```
## [1] "Proportion: 0.371"
```

```r
# ============================================================================
# SECTION 4: Caste-based monthly month_exp gaps in Uttar Pradesh
# ============================================================================

# Create the reserved dummy variable
# 1 = Reserved castes (Scheduled Caste, Scheduled Tribe, Other Backward Class)
# 0 = General caste
reserved_caste_levels <- c("Scheduled Caste", "Scheduled Tribe", "Other Backward Class")

access_UP <- access_UP %>%
  mutate(reserved = ifelse(caste.fac %in% reserved_caste_levels, 1,
                           ifelse(caste.fac == "General", 0, NA)))

# Verify it was created correctly
print("Cross-tabulation of reserved dummy and caste:")
```

```
## [1] "Cross-tabulation of reserved dummy and caste:"
```

```r
table(access_UP$reserved, access_UP$caste.fac, useNA = "ifany")
```

```
##
##     General Other Backward Class Others Scheduled Caste Scheduled Tribe
##   0     708                    0      0               0               0
##   1       0                 1603      0             657              34
```

```r
print("Summary of reserved variable:")
```

```
## [1] "Summary of reserved variable:"
```

```r
summary(access_UP$reserved)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  1.0000  1.0000  0.7642  1.0000  1.0000
```

```r
# 4a) Use the filter function and 'reserved' dummy variable to subset observations belonging to the 'Ge
access_UPgen <- access_UP %>%
  filter(reserved == 0) %>%  # 0 = General caste
  arrange(desc(month_exp))

# Check if it worked
head(access_UPgen, 5)
```

```
## # A tibble: 5 x 14
##   state relationship   age gender education caste incsource month_exp gender.fac
##   <chr> <dbl+lbl>    <dbl> <chr>  <chr>     <chr> <chr>         <dbl> <fct>
## 1 UTTA~ 5 [Others]      18 Male   "Graduat~ Gene~ Agricult~     40000 Male
## 2 UTTA~ 1 [Self]        65 Male   "Graduat~ Gene~ Agricult~     40000 Male
## 3 UTTA~ 1 [Self]        63 Male   "Grade 5" Gene~ Agricult~     30000 Male
## 4 UTTA~ 1 [Self]        55 Female "No Scho~ Gene~ Agricult~     30000 Female
## 5 UTTA~ 1 [Self]        62 Female "Grade 5" Gene~ Agricult~     30000 Female
## # i 5 more variables: caste.fac <fct>, education.fac <fct>, state.fac <fct>,
## #   incsource.fac <fct>, reserved <dbl>
```

```r
summary(access_UPgen$caste.fac)  # Should only show General
```

```
##              General Other Backward Class                Others
##                  708                    0                     0
##      Scheduled Caste      Scheduled Tribe
##                    0                    0
```

```r
# 4b) Repeat part a, but this time, use the filter function to subset observations belonging to 'Reserv
access_UPres <- access_UP %>%
  filter(reserved == 1) %>%  # 1 = Reserved castes
  arrange(desc(month_exp))

# Check if it worked
head(access_UPres, 5)
```

```
## # A tibble: 5 x 14
##   state relationship   age gender education caste incsource month_exp gender.fac
##   <chr> <dbl+lbl>    <dbl> <chr>  <chr>     <chr> <chr>         <dbl> <fct>
## 1 UTTA~ 1 [Self]        43 Male   Graduate~ Othe~ Salaried~     56000 Male
## 2 UTTA~ 1 [Self]        58 Male   Grade 12  Othe~ Salaried~     50000 Male
## 3 UTTA~ 1 [Self]        66 Male   Graduate~ Othe~ Salaried~     50000 Male
## 4 UTTA~ 1 [Self]        24 Male   Grade 12  Othe~ Agricult~     50000 Male
## 5 UTTA~ 4 [Son/Daug~    28 Male   Grade 12  Othe~ Salaried~     50000 Male
## # i 5 more variables: caste.fac <fct>, education.fac <fct>, state.fac <fct>,
## #   incsource.fac <fct>, reserved <dbl>
```

```r
summary(access_UPres$caste.fac)  # Should show Reserved castes only
```

```
##              General Other Backward Class                Others
##                    0                 1603                     0
##      Scheduled Caste      Scheduled Tribe
##                  657                   34
```

```r
# 4c) Use summarise to find mean, min & max monthly month_exp for the General category and Reserved cat
# General caste statistics
gen_stats <- access_UPgen %>%
  summarise(
    gen_mean_exp = mean(month_exp, na.rm = TRUE),
    gen_min_exp = min(month_exp, na.rm = TRUE),
    gen_max_exp = max(month_exp, na.rm = TRUE)
  )

# Reserved caste statistics
res_stats <- access_UPres %>%
  summarise(
    res_mean_exp = mean(month_exp, na.rm = TRUE),
    res_min_exp = min(month_exp, na.rm = TRUE),
    res_max_exp = max(month_exp, na.rm = TRUE)
  )

print("General caste month_exp statistics:")
```

```
## [1] "General caste month_exp statistics:"
```

```
print(gen_stats)
```

```
## # A tibble: 1 x 3
##   gen_mean_exp gen_min_exp gen_max_exp
##          <dbl>       <dbl>       <dbl>
## 1        7861.         300       40000
```

```
print("Reserved caste month_exp statistics:")
```

```
## [1] "Reserved caste month_exp statistics:"
```

```
print(res_stats)
```

```
## # A tibble: 1 x 3
##   res_mean_exp res_min_exp res_max_exp
##          <dbl>       <dbl>       <dbl>
## 1        6475.           0       56000
```

```
# Calculate the gap
month_exp_gap <- gen_stats$gen_mean_exp - res_stats$res_mean_exp
print(paste("Gap in mean monthly month_exp (General - Reserved):", round(month_exp_gap, 2)))
```

```
## [1] "Gap in mean monthly month_exp (General - Reserved): 1385.48"
```

```
# 4d) Research suggests that people belonging to the 'General' caste category own 65% of agricultural l
# Filter for landowners (Agriculture on own land)
landowners <- access_UP %>%
  filter(incsource.fac == "Agriculture (own land)")

# Overall caste distribution in UP (from Q2b)
overall_caste_table <- table(access_UP$reserved)
overall_caste_prop <- prop.table(overall_caste_table)
print("Overall caste distribution in UP:")
```

```
## [1] "Overall caste distribution in UP:"
```

```
print(overall_caste_table)
```

```
##
##    0    1
##  708 2294
```

```
print("Overall caste proportions:")
```

```
## [1] "Overall caste proportions:"
```

```r
round(overall_caste_prop, 3)
```

```
##
##     0     1
## 0.236 0.764
```

```r
# Landowner caste distribution
landowner_caste_table <- table(landowners$reserved)
landowner_caste_prop <- prop.table(landowner_caste_table)
print("Landowner caste distribution:")
```

```
## [1] "Landowner caste distribution:"
```

```r
print(landowner_caste_table)
```

```
##
##   0   1
## 433 877
```

```r
print("Landowner caste proportions:")
```

```
## [1] "Landowner caste proportions:"
```

```r
round(landowner_caste_prop, 3)
```

```
##
##     0     1
## 0.331 0.669
```

```r
# Compare proportions
print("Comparison:")
```

```
## [1] "Comparison:"
```

```r
print(paste("General caste: Overall proportion =", round(overall_caste_prop["0"], 3),
            ", Landowner proportion =", round(landowner_caste_prop["0"], 3)))
```

```
## [1] "General caste: Overall proportion = 0.236 , Landowner proportion = 0.331"
```

```r
print(paste("Reserved caste: Overall proportion =", round(overall_caste_prop["1"], 3),
            ", Landowner proportion =", round(landowner_caste_prop["1"], 3)))
```

```
## [1] "Reserved caste: Overall proportion = 0.764 , Landowner proportion = 0.669"
```

```r
# 4e) Do differences in landowning explain the month_exp gap between General and Reserved castes?
# General caste landowners
gen_landowners <- access_UP %>%
  filter(reserved == 0, incsource.fac == "Agriculture (own land)")

# Reserved caste landowners
res_landowners <- access_UP %>%
  filter(reserved == 1, incsource.fac == "Agriculture (own land)")

# Mean month_exps for landowners
gen_landowner_mean <- mean(gen_landowners$month_exp, na.rm = TRUE)
res_landowner_mean <- mean(res_landowners$month_exp, na.rm = TRUE)

print(paste("Mean month_exp - General landowners:", round(gen_landowner_mean, 2)))
```

```
## [1] "Mean month_exp - General landowners: 7800.46"
```

```r
print(paste("Mean month_exp - Reserved landowners:", round(res_landowner_mean, 2)))
```

```
## [1] "Mean month_exp - Reserved landowners: 6935.58"
```

```r
landowner_gap <- gen_landowner_mean - res_landowner_mean
print(paste("Gap in mean monthly month_exp (General - Reserved landowners):", round(landowner_gap, 2)))
```

```
## [1] "Gap in mean monthly month_exp (General - Reserved landowners): 864.89"
```

```r
# Compare to overall gap from 4c
print(paste("Overall gap (from 4c):", round(month_exp_gap, 2)))
```

```
## [1] "Overall gap (from 4c): 1385.48"
```

```r
print(paste("Landowner gap:", round(landowner_gap, 2)))
```

```
## [1] "Landowner gap: 864.89"
```

```r
# 4f) Is there a gap between the mean monthly month_exps of General caste male and Reserved caste male
# Male landowners by caste
gen_male_landowners <- access_UP %>%
  filter(reserved == 0, incsource.fac == "Agriculture (own land)", gender.fac == "Male")

res_male_landowners <- access_UP %>%
  filter(reserved == 1, incsource.fac == "Agriculture (own land)", gender.fac == "Male")

gen_male_mean <- mean(gen_male_landowners$month_exp, na.rm = TRUE)
res_male_mean <- mean(res_male_landowners$month_exp, na.rm = TRUE)
male_gap <- gen_male_mean - res_male_mean

print("Male landowners:")
```

```
## [1] "Male landowners:"
```

```r
print(paste("General male mean:", round(gen_male_mean, 2)))
```

```
## [1] "General male mean: 7819.24"
```

```r
print(paste("Reserved male mean:", round(res_male_mean, 2)))
```

```
## [1] "Reserved male mean: 6980.5"
```

```r
print(paste("Gap (General - Reserved):", round(male_gap, 2)))
```

```
## [1] "Gap (General - Reserved): 838.74"
```

```r
# Female landowners by caste
gen_female_landowners <- access_UP %>%
  filter(reserved == 0, incsource.fac == "Agriculture (own land)", gender.fac == "Female")

res_female_landowners <- access_UP %>%
  filter(reserved == 1, incsource.fac == "Agriculture (own land)", gender.fac == "Female")

gen_female_mean <- mean(gen_female_landowners$month_exp, na.rm = TRUE)
res_female_mean <- mean(res_female_landowners$month_exp, na.rm = TRUE)
female_gap <- gen_female_mean - res_female_mean

print("Female landowners:")
```

```
## [1] "Female landowners:"
```

```r
print(paste("General female mean:", round(gen_female_mean, 2)))
```

```
## [1] "General female mean: 7728.89"
```

```r
print(paste("Reserved female mean:", round(res_female_mean, 2)))
```

```
## [1] "Reserved female mean: 6813.56"
```

```r
print(paste("Gap (General - Reserved):", round(female_gap, 2)))
```

```
## [1] "Gap (General - Reserved): 915.33"
```

```r
# 4g) Does educational attainment explain part of the month_exp gap between General and Reserved caste
# First, check education levels to identify HS degree or more
print("All education levels in data:")
```

```
## [1] "All education levels in data:"
```

```r
print(levels(access_UP$education.fac))
```

```
## [1] "Grade 10"          "Grade 12"          "Grade 5"
## [4] "Graduate or above" "No School "
```

```r
print("Education distribution among all respondents:")
```

```
## [1] "Education distribution among all respondents:"
```

```r
table(access_UP$education.fac)
```

```
##
##          Grade 10          Grade 12           Grade 5 Graduate or above
##               439               361               816               311
##        No School
##              1075
```

```r
# Check education levels for landowners specifically
landowner_education <- access_UP %>%
  filter(incsource.fac == "Agriculture (own land)") %>%
  select(education.fac) %>%
  table()
print("Education distribution among landowners:")
```

```
## [1] "Education distribution among landowners:"
```

```r
print(landowner_education)
```

```
## education.fac
##          Grade 10          Grade 12           Grade 5 Graduate or above
##               237               202               354               177
##        No School
##               340
```

```r
# Get all landowners first
all_landowners <- access_UP %>%
  filter(incsource.fac == "Agriculture (own land)")

# Filter for HS+ (adjust education level names as needed)
# This is a flexible approach - check the levels() output above and adjust
gen_landowners_hs <- all_landowners %>%
  filter(reserved == 0)

res_landowners_hs <- all_landowners %>%
  filter(reserved == 1)

# Check what education levels exist - you'll need to manually identify which ones are HS+
# For now, showing all education levels for landowners by caste
print("Education levels for General landowners:")
```

```
## [1] "Education levels for General landowners:"
```

```
table(gen_landowners_hs$education.fac)
```

```
##
##          Grade 10          Grade 12           Grade 5 Graduate or above
##               102                84                   108                66
##        No School
##                73
```

```
print("Education levels for Reserved landowners:")
```

```
## [1] "Education levels for Reserved landowners:"
```

```
table(res_landowners_hs$education.fac)
```

```
##
##          Grade 10          Grade 12           Grade 5 Graduate or above
##               135               118                   246               111
##        No School
##               267
```

```
# NOTE: You need to identify which education levels represent "HS degree or more"
# based on the output above, then filter accordingly. For example:
# If "High School" and above are HS+, you might do:
# gen_landowners_hs <- gen_landowners_hs %>% filter(education.fac %in% c("High School", "College", "Gra
# res_landowners_hs <- res_landowners_hs %>% filter(education.fac %in% c("High School", "College", "Gra

# Calculate means for HS+ educated landowners (after filtering above)
gen_hs_mean <- mean(gen_landowners_hs$month_exp, na.rm = TRUE)
res_hs_mean <- mean(res_landowners_hs$month_exp, na.rm = TRUE)
hs_gap <- gen_hs_mean - res_hs_mean

print("Landowners with HS degree or more:")
```

```
## [1] "Landowners with HS degree or more:"
```

```
print(paste("General mean:", round(gen_hs_mean, 2)))
```

```
## [1] "General mean: 7800.46"
```

```
print(paste("Reserved mean:", round(res_hs_mean, 2)))
```

```
## [1] "Reserved mean: 6935.58"
```

```
print(paste("Gap (General - Reserved):", round(hs_gap, 2)))
```

```
## [1] "Gap (General - Reserved): 864.89"
```

```r
# Compare to overall landowner gap from 4e
print(paste("Overall landowner gap (from 4e):", round(landowner_gap, 2)))
```

```
## [1] "Overall landowner gap (from 4e): 864.89"
```

1 Load and inspect ACCESS 2018 data:

1a) Inspect the data frame and state the data types for each column.

- make sure to inspect the age, gender, caste, income source, state & educ columns

1b) Use the mutate function to create new column for gender:

- gender.fac = as.factor(gender),
- check if it worked, show your validation

1c) In a single pipe, include gender.fac in a new data frame called access2018.temp1.

- also create factors for caste, income source, state and education
- exclude the columns for HHID and date after creating access2018.temp1
- print the first 5 observations

1d) Inspect caste.fac, gender.fac, incsource.fac, education.fac and state.fac using the levels() function. What package is the levels() function located in?

1e) Use filter() to only include rows only for the state "Uttar Pradesh."

- store as a new object access_UP
- print the first 5 observations
- confirm your data only includes observations for Uttar Pradesh

1f) What is the female share of respondents in Uttar Pradesh? Compare it to the female share of respondents in Bihar?

1g) Remove the access2018.temp1 object from memory using the rm() function.

2 Describe the access_UP data frame

2a) What is the unit of observation (or unit of analysis)?

2b) How many individuals are observed?

- how many of these are from 'Reserved' castes (i.e: Scheduled Caste, Scheduled Tribe, Other Backward Class")?
- how many are "General" caste members? [**Note: for this question, use the 'table' and 'prop.table' functions. 'table' will give the number of observations belonging to each caste category.'prop.table will give the proportion**]

2c) Do you think the sample is representative of all households in India?

- how would you describe the population represented by this sample?
- is there more information you would like to see to assess the representativeness of the sample?

2d) What is the average age of individuals in the sample? youngest and oldest person?

3 Earnings per week for different groups in Uttar Pradesh

3a) Find the observation for the top monthly month_exp using the summarize() function.

- assign this to a new object called max_exp_obs1

3b) Find max monthly month_exps using the arrange function instead of summarize.

3c) Use the filter function to subset for the observation with max monthly month_exp.

- (don't hardcode the max month_exp to filter on, refer to the max_exp_obs1 object from a)
- store in new data frame max_exp_obs2
- confirm it worked

3d) What is the age, gender and caste of the top monthly spender in the sample?

3e) List the age, gender and caste of the top 10 monthly spenders in the sample.

3f) How many individuals spend more than the mean monthly month_exp amount of the sample?

Caste-based monthly month_exp gaps in Uttar Pradesh.

4a) Use the filter function and 'reserved' dummy variable to subset observations belonging to the 'General' caste.

- assign to new data frame, access_UPgen
- sort in descending order of monthly month_exp
- check if it worked

4b) Repeat part a, but this time, use the filter function to subset observations belonging to 'Reserved" castes, and assign them to a new data frame called 'access_UPres'.

4c) Use summarise to find mean, min & max monthly month_exp for the General category and Reserved category of castes, separately. Name each statistic appropriately (i.e. name each column in the 1-row table of stats). What is the gap in mean monthly month_exp between the two groups?

4d) Research suggests that people belonging to the 'General' caste category own 65% of agricultural land in India (India Human Development Survey, 2020). They are also more likely to own land in a proportion that is much higher than their share of the population. We will now use this data to compare 'General' versus 'Reserved' households using the 'reserved' dummy variable to understand whether this is true.

- use the 'table' and 'prop.table' functions to explore the total number of landowners by their caste.
- here, 'landowners' are people whose primary source of income is agriculture on their own land. So, you can use the 'incsource.fac' variable to filter the 'access_UP' data frame for 'Agriculture (own land)'
- does the number of General and Reserved caste landowners seem proportional to the overall number of 'General' and 'Reserved' caste individuals in Uttar Pradesh's population (as calculated in Q2 b)?

4e) Do differences in landowning explain the month_exp gap between General and Reserved castes? i.e., does the General-Reserved month_exp gap persist among landowning families? What is the gap in mean monthly month_exps of General vs Reserved category landowners?

4f) Is there a gap between the mean monthly month_exps of General caste male and Reserved caste male landowners? What about the same, but for the female landowners?

4g) Does educational attainment explain part of the month_exp gap between General and Reserved caste landowners? What is gap between landowning General vs. Reserved household with a HS degree or more?