DPSS 2023 Capstone

Gabriel Toscano

2023-07-28

Contents

Loading Dependencies

```
library(dplyr)
library(ggplot2)
library(tidyverse)
library(sf)
library(viridis)
library(grid)
#load car package for plotting fitted lm models
library(car)
```

Q1: Data Overview

(1.1) Load the UNHCR_Survey data.

```
un_survey <- read.csv('UNHCR_Survey2.csv')
```

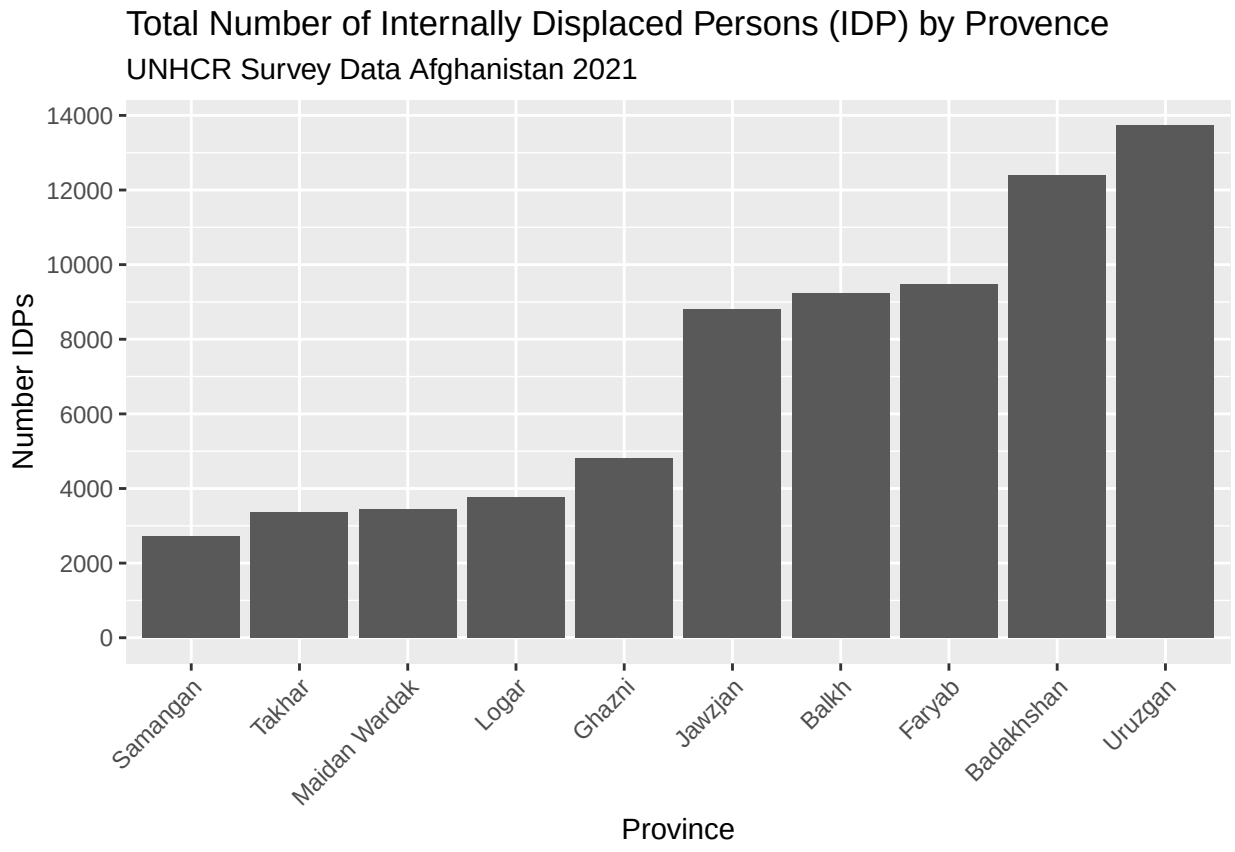
(1.2) Top 10 provinces by number of IDP

The UNHCR_Survey data is already filtered to only IDP records

```
top_10_provinces <- un_survey %>%
  dplyr::select(province, total_members) %>%
  group_by(province) %>%
  summarize(num_idp = sum(total_members)) %>%
  arrange(-num_idp) %>%
  slice(1:10)

knitr::kable(top_10_provinces)
```

province	num_idp
Uruzgan	13730
Badakhshan	12383
Faryab	9471
Balkh	9234
Jawzjan	8806
Ghazni	4817
Logar	3773
Maidan Wardak	3440
Takhar	3376
Samangan	2721

```
  ggplot(top_10_provinces) +
  geom_col(aes(x = reorder(province, num_idp), y = num_idp)) +
    labs(
    title = "Total Number of Internally Displaced Persons (IDP) by Provence",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Province",
    y = "Number IDPs"
  ) +  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
    scale_y_continuous(breaks = seq(0, 20000, by = 2000))
```



Q2: Common Challenges Analysis

(For Question 2, you need to select five of the six challenges (2.1 - 2.6) to analyze) In this part, to investigate the distribution of numerical features, you may consider using visual representations such as histograms or box plots. You also need to calculate the statistical measurements such as the mean, median, mode, standard deviation, and range to understand the distribution of numerical features. To investigate the distribution of categorical features, you may consider using a bar graph or any other appropriate methods.

(2.1) life-sustaining sectoral needs

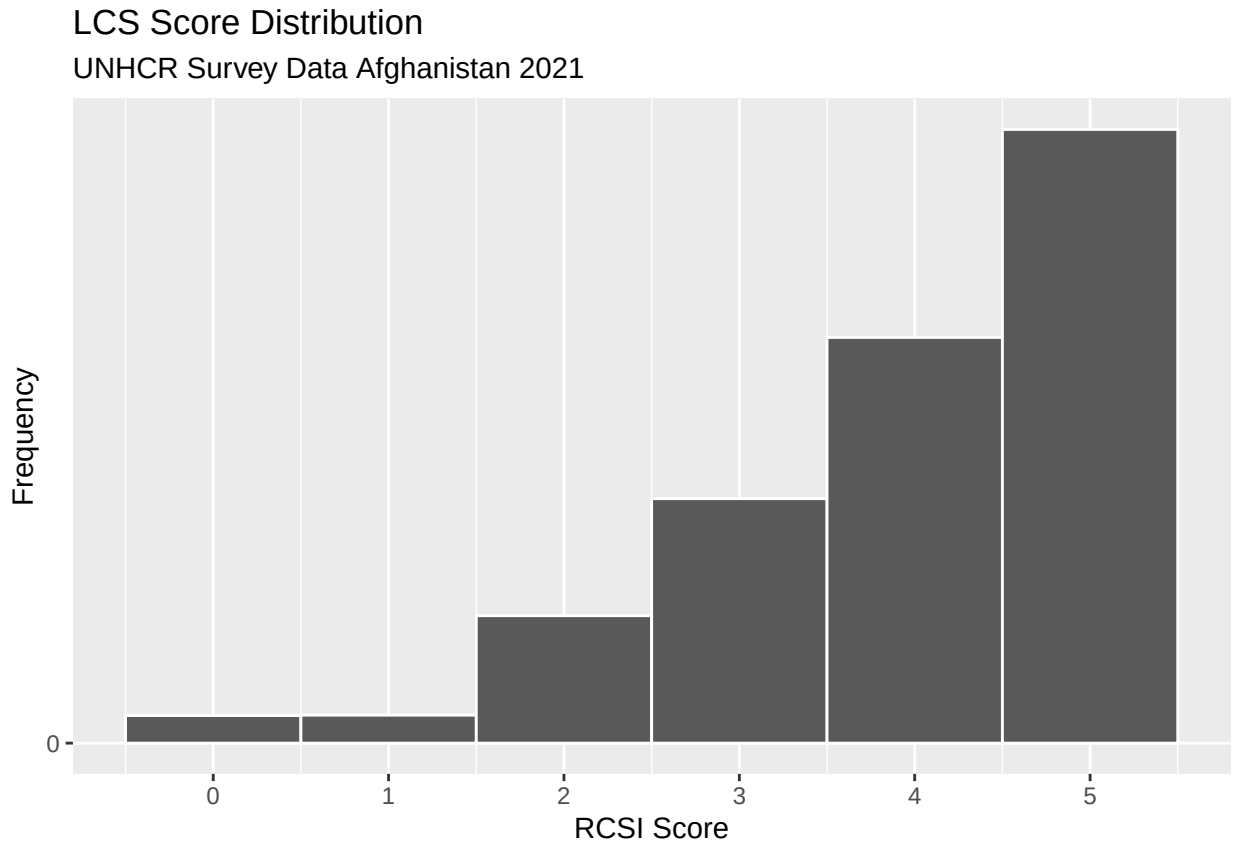
(2.1.1) Analyze and visualize the distribution of lcs score. lcs_score = calc_sold_asset + calc_seek_employment + calc_working + calc_borrow + calc_medical_attention.

```
un_survey <- un_survey %>%
  mutate(lcs_score = calc_sold_asset + calc_seek_employment + calc_working + calc_borrow + calc_medical_

summary(un_survey$lcs_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   3.000   4.000   3.945   5.000   5.000    4271
```

```
ggplot(un_survey, aes(x=lcs_score)) +
 geom_histogram(color="white",binwidth = 1) +
```

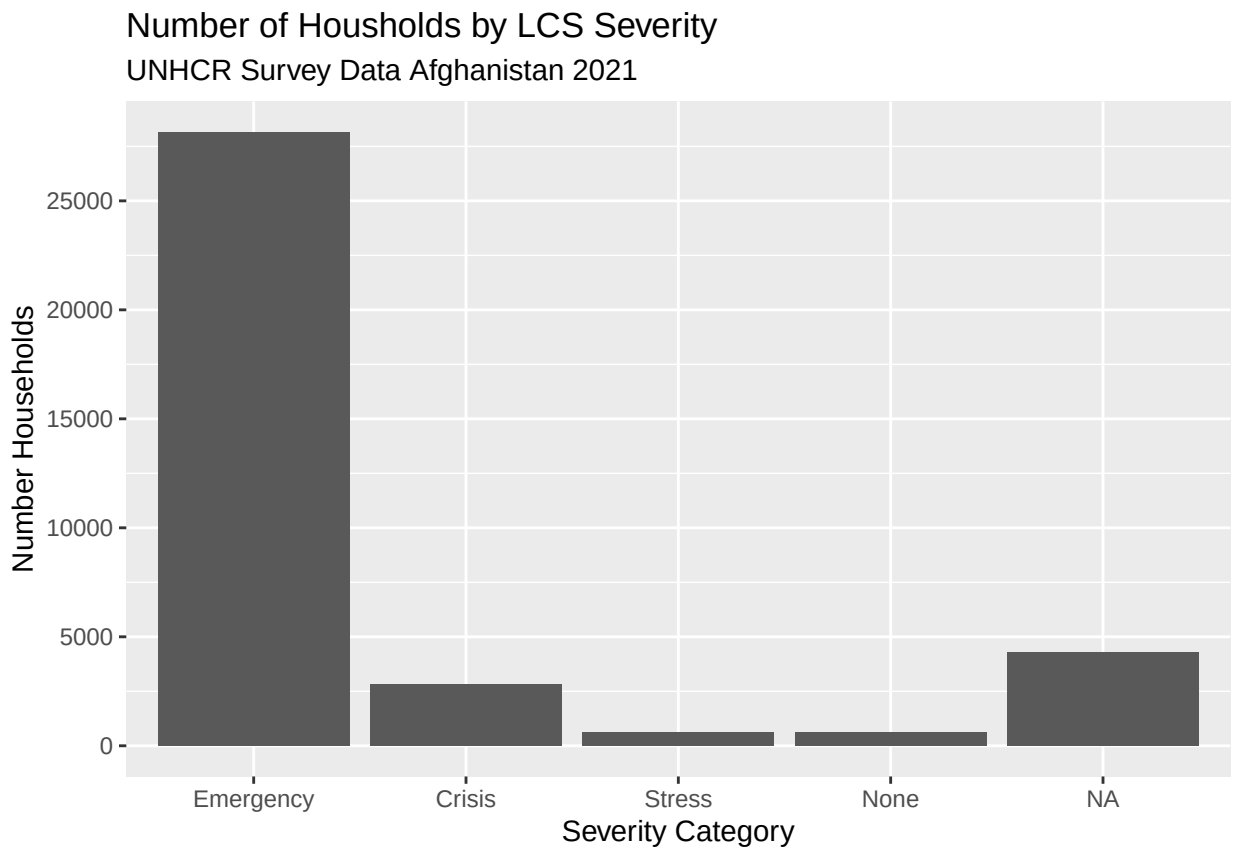
```
    labs(
    title = "LCS Score Distribution",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "RCSI Score",
    y = "Frequency"
  ) +
    scale_y_continuous(breaks = seq(0, 1500, by = 2000)) +
    scale_x_continuous(breaks = seq(0, 6, by=1))
```



(2.1.2) How severe are their life-sustaining sectoral needs under threat? How many interviewees are in emergency, crisis or stress status?

```
#Creating the lcs_severity variable
un_survey <- un_survey %>%
  mutate(lcs_severity = case_when(
                        lcs_score == 0 ~ "None",
                        lcs_score == 1 ~ "Stress",
                        lcs_score == 2 ~ "Crisis",
                        lcs_score >= 3 ~ "Emergency"
                        ))

#Total number of households by lcs_severity
lcs_severity_household_count <- un_survey %>%
  group_by(lcs_severity) %>%
```

```
  summarize(count = n()) %>%
  arrange(-count)

ggplot(lcs_severity_household_count) +
geom_col(aes(x = reorder(lcs_severity, -count), y = count)) +
    labs(
    title = "Number of Housholds by LCS Severity",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Severity Category",
    y = "Number Households" ) +
  scale_y_continuous(breaks = seq(0, 30000, by = 5000))
```

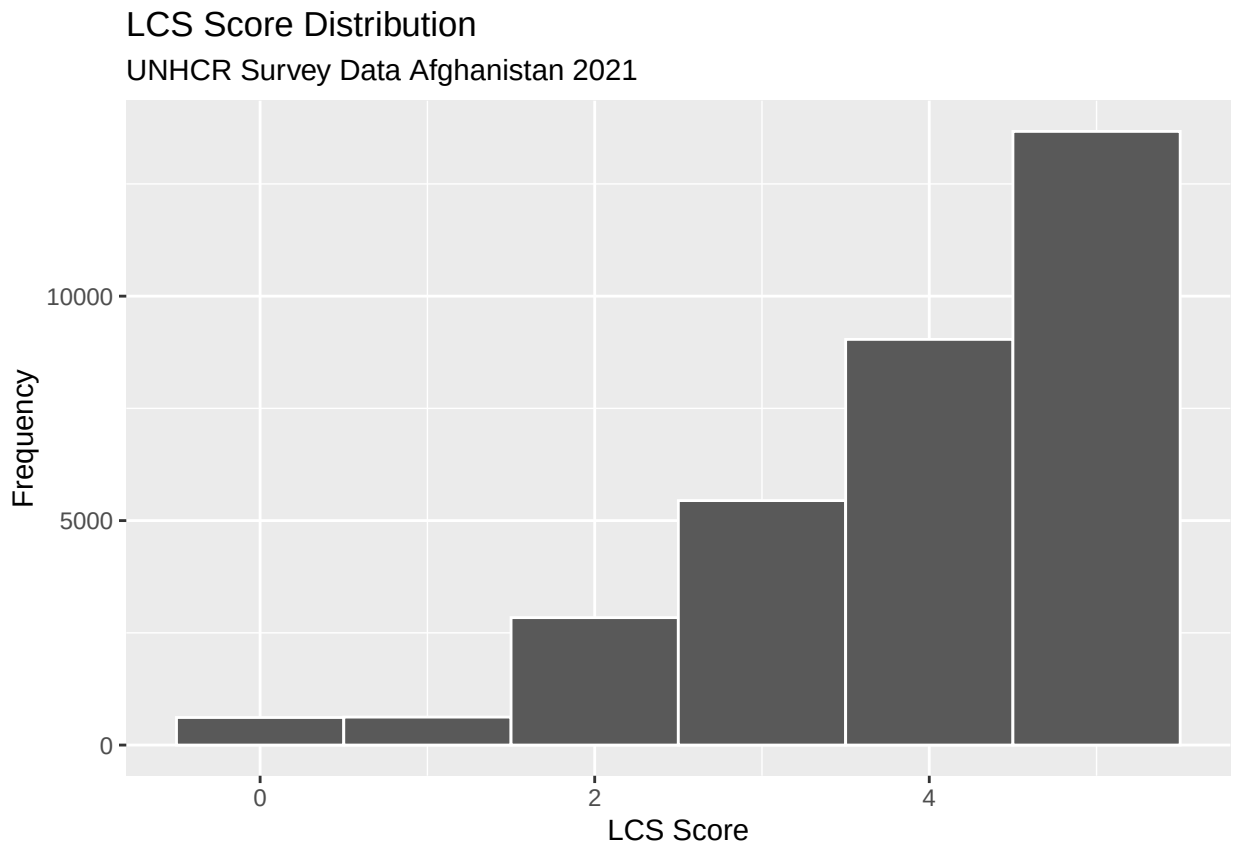



```
knitr::kable(lcs_severity_household_count)
```

lcs_severity	count
Emergency	28153
NA	4271
Crisis	2839
Stress	623
None	614

```
ggplot(un_survey, aes(x=lcs_score)) +
 geom_histogram(colour="white", binwidth = 1) +
    labs(
    title = "LCS Score Distribution",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "LCS Score",
    y = "Frequency"
  )
```

```
## Warning: Removed 4271 rows containing non-finite values (‘stat_bin()‘).
```



(2.2) Food insecurity

How is the rcsi_score distributed? Analyze the distribution of rcsi_score. Is it left-skewed or right-skewed? rcsi_score = ${less_food} + ${borrow_food} * 2 + ${limit_mealtime} + ${restrict_consumption} * 3 + ${reduce_meal}. In the survey data, the rcsi_score= foodcoping1 + (foodcoping2 * 2) + foodcoping3 + (foodcoping4 * 3) + foodcoping5

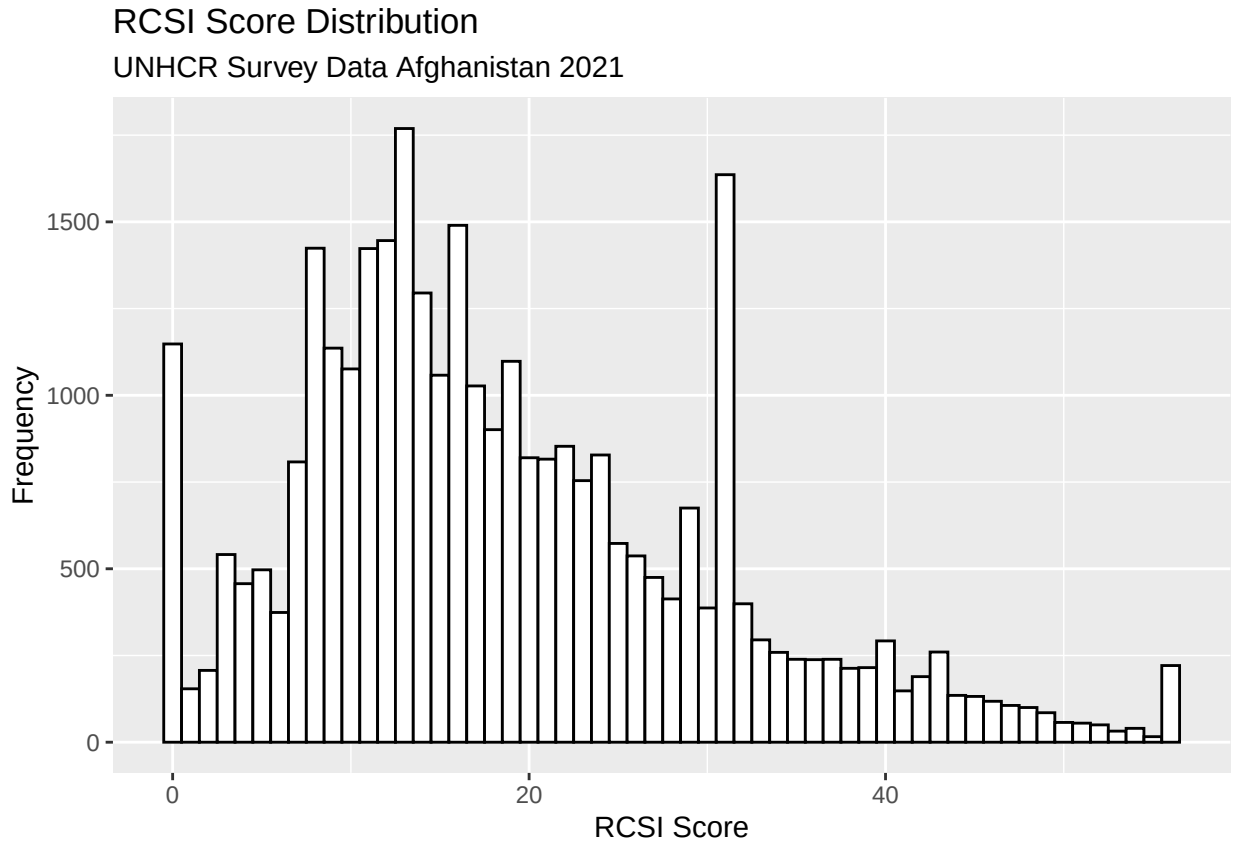
RCSI is left skewed in the survey data

```
#calculate RCSI Score
un_survey <- un_survey %>%
  mutate(rcsi_score = food_coping1 + (food_coping2 * 2) + food_coping3 + (food_coping4 * 3) + food_copi
```

```r
# Distribution Summary
summary(un_survey$rcsi_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   11.00   16.00   18.81   26.00   56.00    4271
```

```r
#Plot RCSI distribution
ggplot(un_survey, aes(x=rcsi_score)) +
 geom_histogram(colour="black", fill="white", binwidth = 1) +
    labs(
    title = "RCSI Score Distribution",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "RCSI Score",
    y = "Frequency"
  )
```



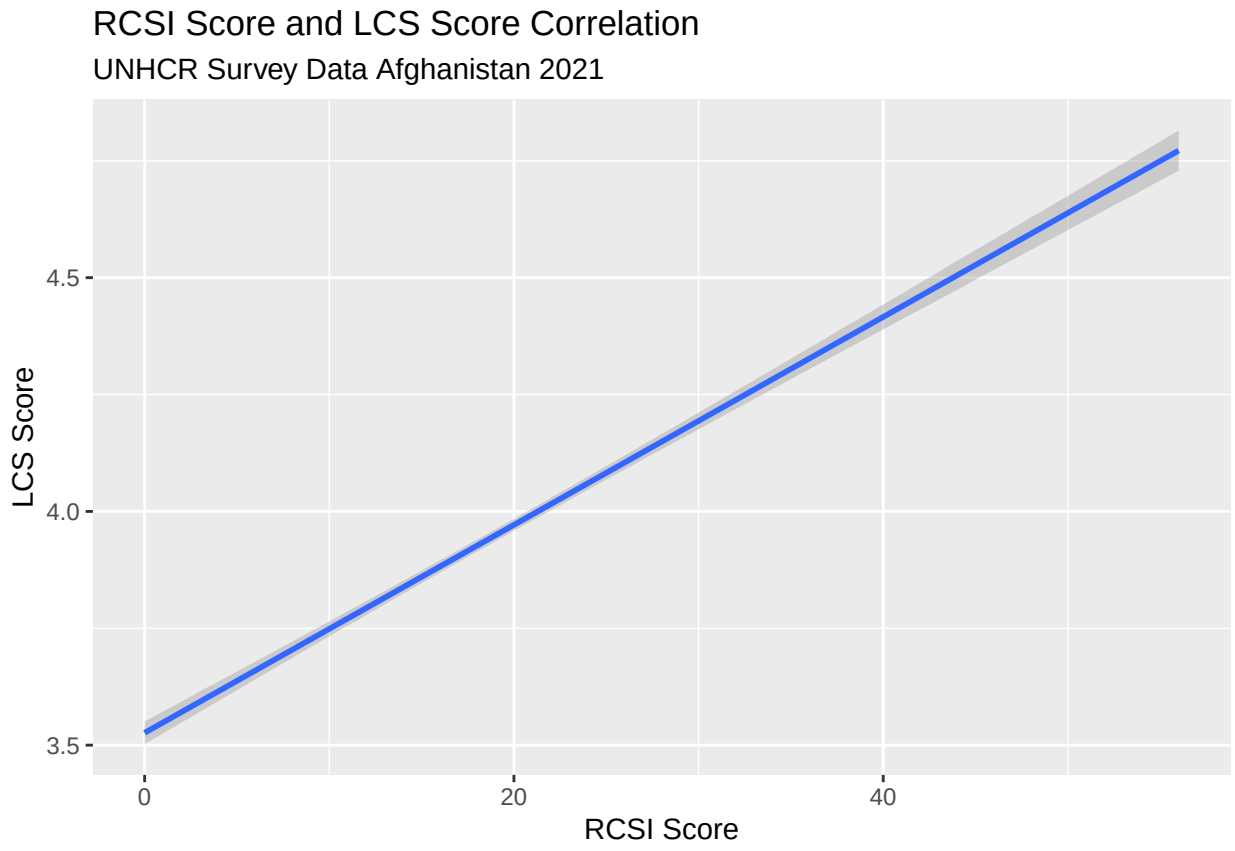
What is the correlation between rcsi_score and lcs_score?

```r
cor.test(un_survey$lcs_score, un_survey$rcsi_score, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  un_survey$lcs_score and un_survey$rcsi_score
```

```
## t = 39.595, df = 32227, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2049470 0.2257694
## sample estimates:
##       cor
## 0.2153827
```

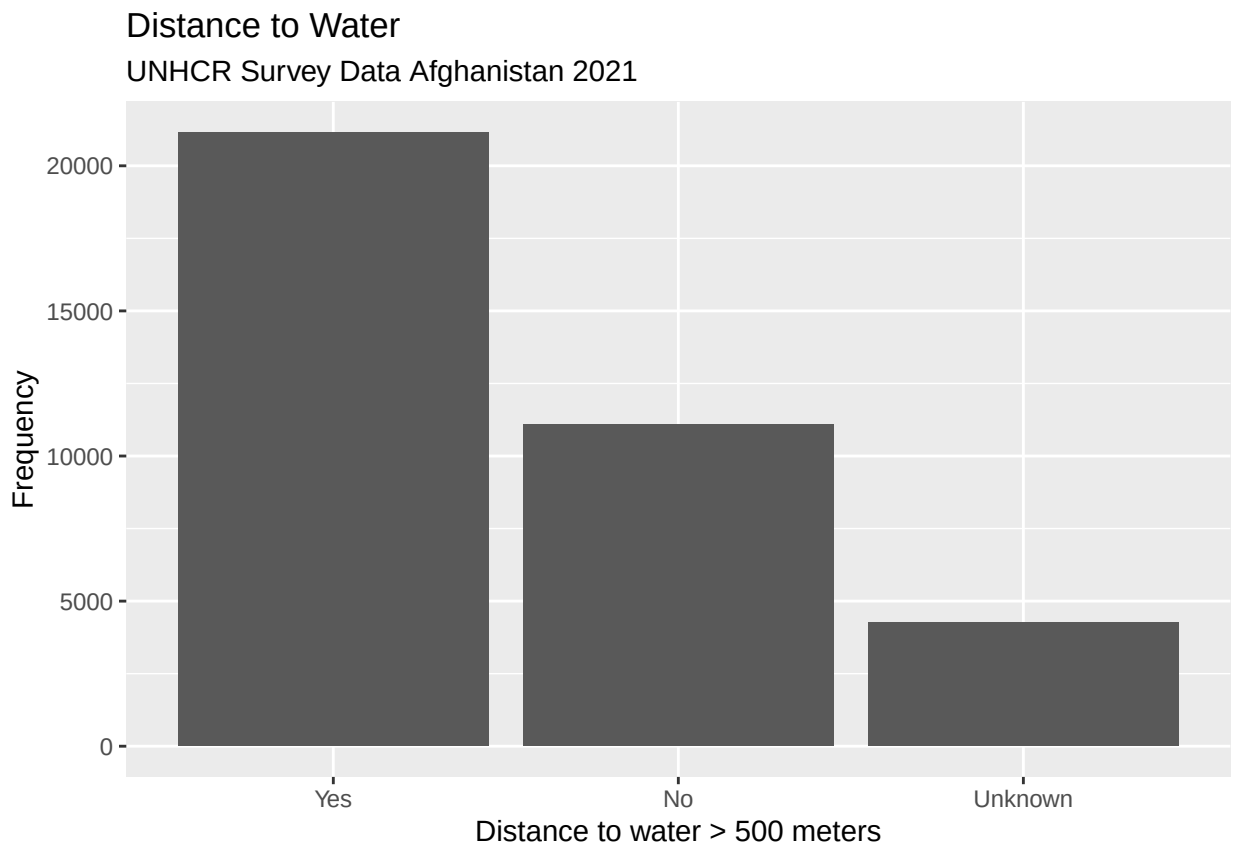
```r
un_survey %>% ggplot() +
  geom_smooth(mapping = aes(x = rcsi_score, y = lcs_score), method=lm) +
    labs(
    title = "RCSI Score and LCS Score Correlation",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "RCSI Score",
    y = "LCS Score"
  )
```



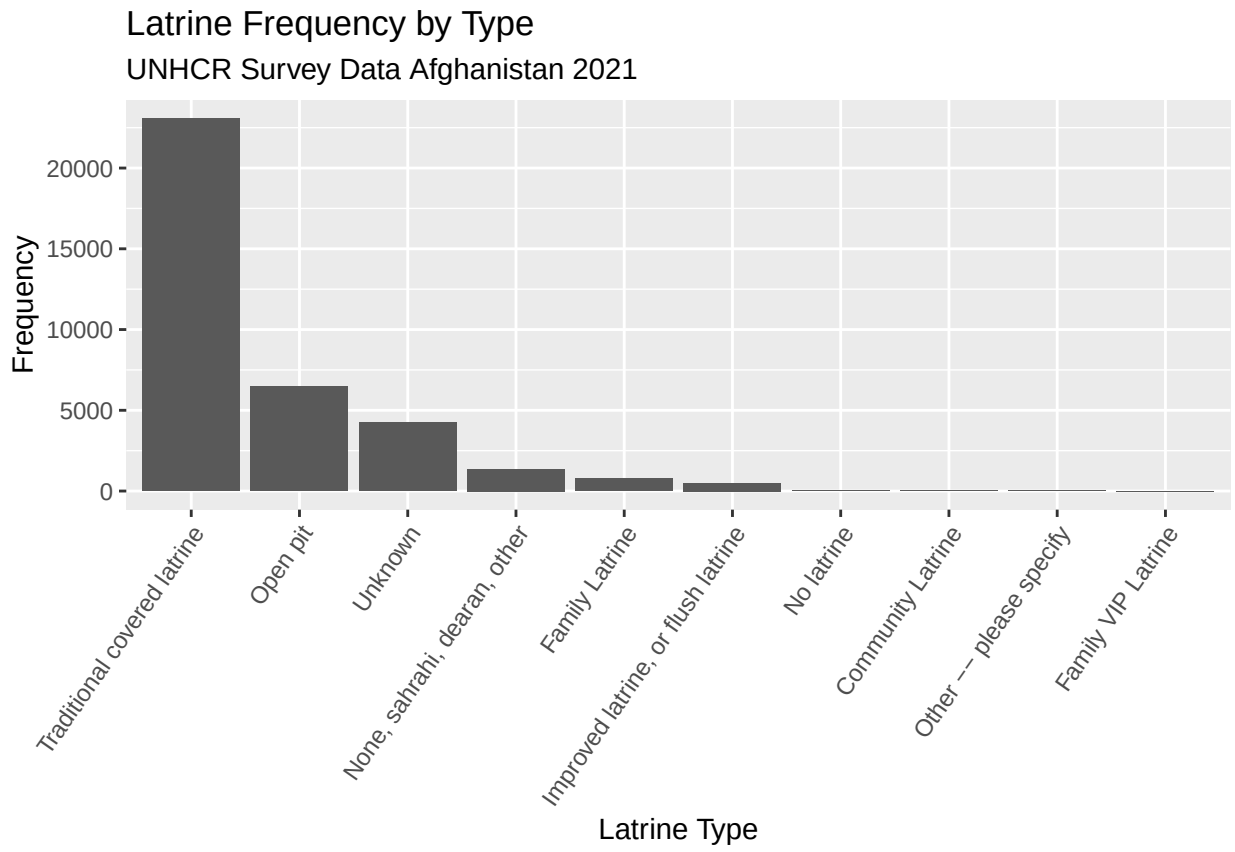
(2.3) Water Safety

For water safety analysis, you can consider two variables: "distance_water" (binary feature indicating whether the house is more than 500 meters from water) and "latrine" (the type of latrine the household has access to). Analyze the distribution of "distance_water" to understand how many households are located more than 500 meters from water sources. Additionally, examine the distribution of "latrine" to identify the types of latrines accessible to the households.

```r
water_safety <- un_survey %>%
  dplyr::select(distance_water) %>%
  drop_na() %>%
  group_by(distance_water) %>%
  summarize(count = n())


#Updating blank values to unknown
water_safety$distance_water[water_safety$distance_water == ""] <- "Unknown"

latrine <- un_survey %>%
  dplyr::select(latrine) %>%
  group_by(latrine) %>%
  summarise(count = n())

#Updating latrine names for graph
latrine$latrine[latrine$latrine == ""] <- "Unknown"
latrine$latrine[latrine$latrine == "None (open field, bush) or sahrahi, dearan (area inside or outside

ggplot(water_safety) +
  geom_col(aes(x = reorder(distance_water, -count), y = count)) +
    labs(
    title = "Distance to Water",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Distance to water > 500 meters",
    y = "Frequency"
  )
```




```
ggplot(latrine) +
  geom_col(aes(x = reorder(latrine, -count), y = count)) +
    labs(
    title = "Latrine Frequency by Type",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Latrine Type",
    y = "Frequency"
  ) +  theme(axis.text.x = element_text(angle = 55, vjust = 1, hjust=1))
```

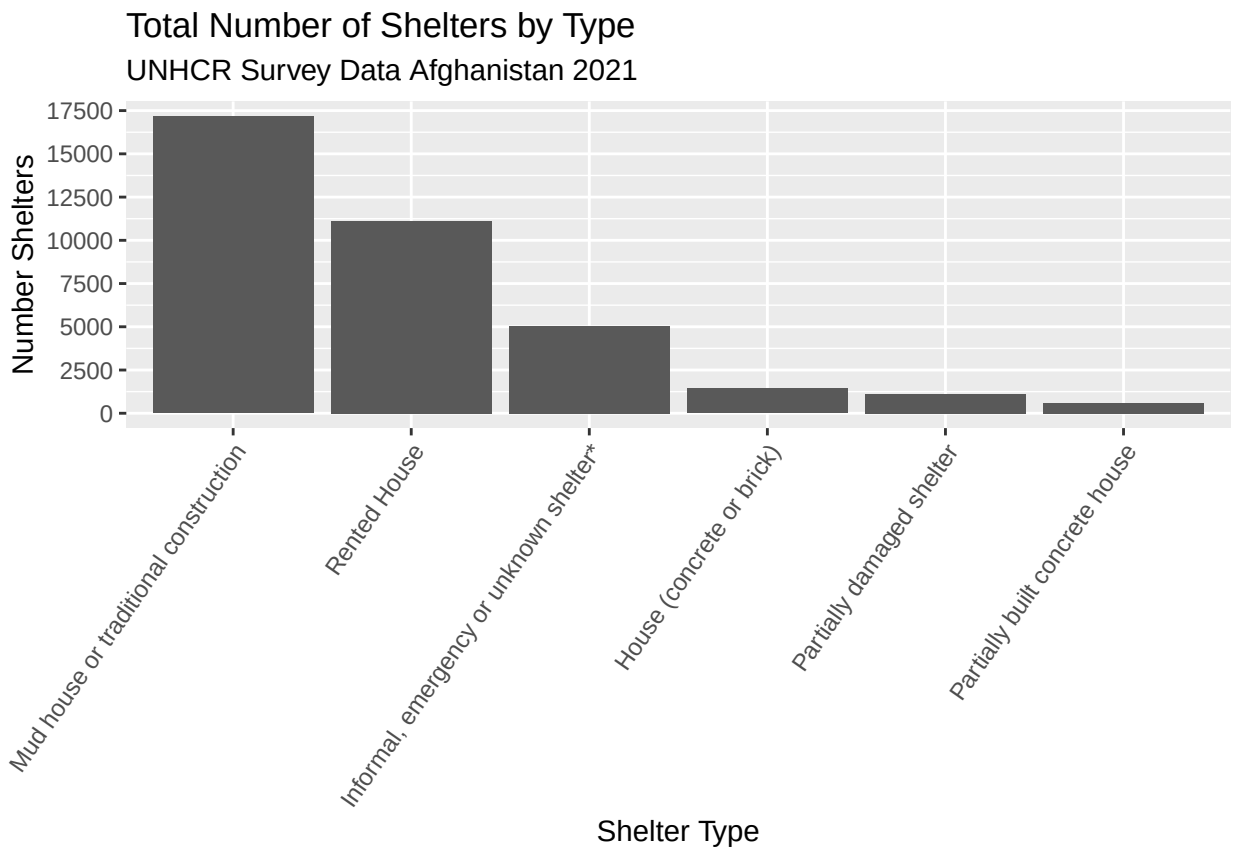


(2.4) Shelter and Living Conditions:

(2.4.1) Focus on the 'shelter' and 'shelter_number' and 'tenure'. Analyze the frequency of the different types of shelter, the number of people living in different types of shelter, and the distribution of how the tenure is collected.Shelter denotes the type of shelter. Shelter number denotes how many people live in the shelter. Tenure denotes the state of occupation (ownership, renting, etc).

```
num_shelter_type <- un_survey %>%
  dplyr::select(shelter) %>%
  group_by(shelter) %>%
  summarize(count = n()) %>%
  arrange(-count)

#bundle the lowest frequency cats, and in write-up talk about how they were grouped!!
bundle_sum <- 0
shelter_to_bundle <- c("Tent or Emergency shelter","Makeshift shelter", "Collective centre", "Open spac
for(i in 1:length(num_shelter_type$shelter)){
  curr_shelter <- num_shelter_type$shelter[i]
  if (curr_shelter %in% shelter_to_bundle) {
    bundle_sum <- bundle_sum + num_shelter_type$count[i]
  }
}

#Remove rows that are part of our bundled, and then add the new bundled row
num_shelter_type_bundled <- subset(num_shelter_type, !num_shelter_type$shelter %in% shelter_to_bundle)
```

```
  add_row(shelter = "Informal, emergency or unknown shelter*", count = bundle_sum)

  ggplot(num_shelter_type_bundled) +
  geom_col(aes(x = reorder(shelter, -count), y = count)) +
    labs(
    title = "Total Number of Shelters by Type",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Shelter Type",
    y = "Number Shelters"
  ) +  theme(axis.text.x = element_text(angle = 55, vjust = 1, hjust=1)) +
     scale_y_continuous(breaks = seq(0, 20000, by = 2500))
```

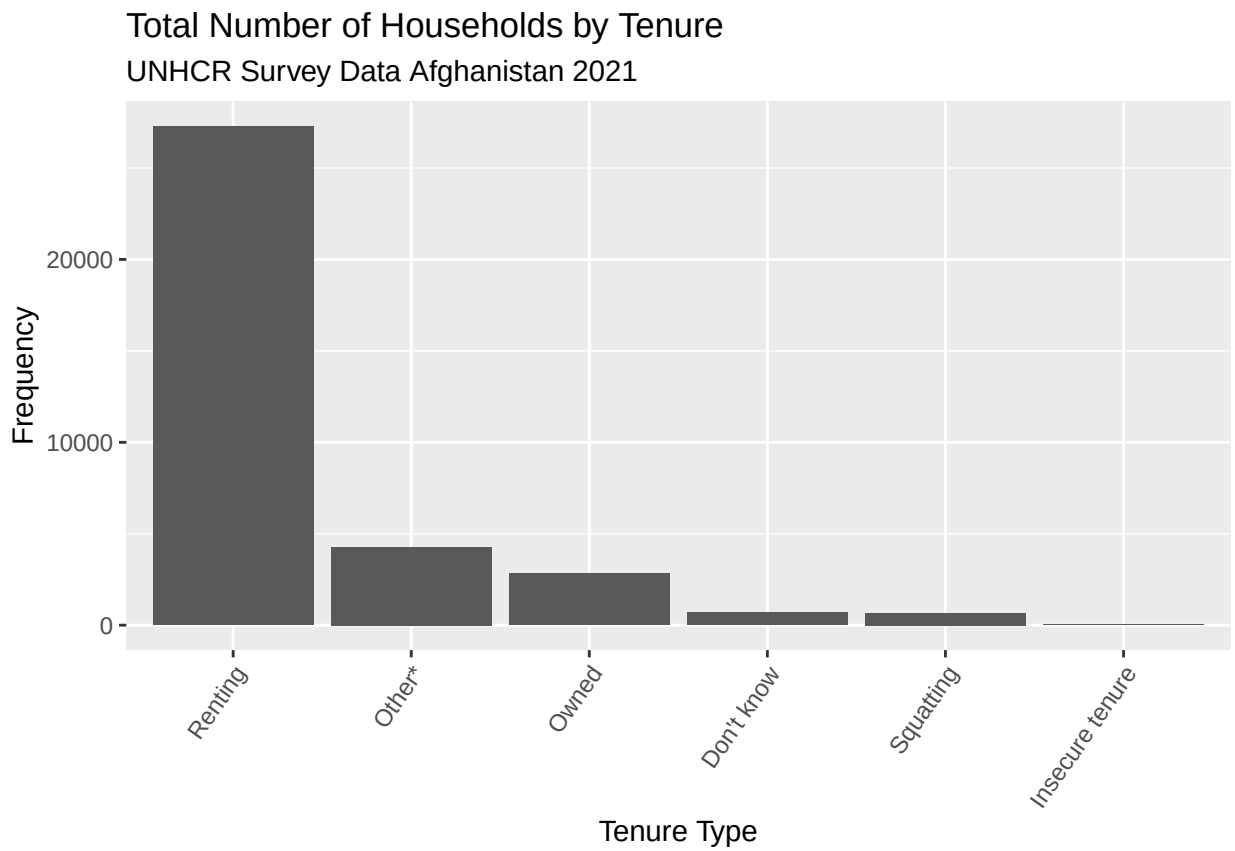


The categories (Tent or Emergency shelter,Makeshift shelter, Collective centre, Open space or no shelter, " ") were bundled as: (Informal, emergency or unknown shelter)

shelter	count
Mud house or traditional construction	17172
Rented House	11107
House (concrete or brick)	1441
Partially damaged shelter	1120
Partially built concrete house	606
Informal, emergency or unknown shelter*	5054

Distribution of how the tenure is collected.

```r
num_tenure_household <- un_survey %>%
  dplyr::select(tenure) %>%
  group_by(tenure) %>%
  summarize(count = n()) %>%
  arrange(-count)

#Shortening row names for graphing
bundle_sum = 0
tenure_to_bundle <- c("Other", "")
for(i in 1:length(num_tenure_household$tenure)){
  curr_tenure <- num_tenure_household$tenure[i]

  #add to the bundle sum
  if (curr_tenure %in% shelter_to_bundle) {
    bundle_sum <- bundle_sum + num_tenure_household$count[i]

  #change names
  }
  if(curr_tenure == "Insecure tenure status as rent, hosted, iSETs, labour exchanged, etc..)"){
    num_tenure_household$tenure[i] <- "Insecure tenure"
  }
  if(curr_tenure == "Squatting, as occupation without permission, including collective centres"){
    num_tenure_household$tenure[i] <- "Squatting"
  }
  if(curr_tenure == "Owned (land deed, customary tenure document, letter of permission from government a
    num_tenure_household$tenure[i] <- "Owned"
  }
}

#Bundling all blank, other categories
num_tenure_bundled <- subset(num_tenure_household, !num_tenure_household$tenure %in% tenure_to_bundle) %
  add_row(tenure = "Other*", count = bundle_sum) %>%
  arrange(-count)

  ggplot(num_tenure_bundled) +
  geom_col(aes(x = reorder(tenure, -count), y = count)) +
    labs(
    title = "Total Number of Households by Tenure",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Tenure Type",
    y = "Frequency"
  ) +  theme(axis.text.x = element_text(angle = 55, vjust = 1, hjust=1))
```



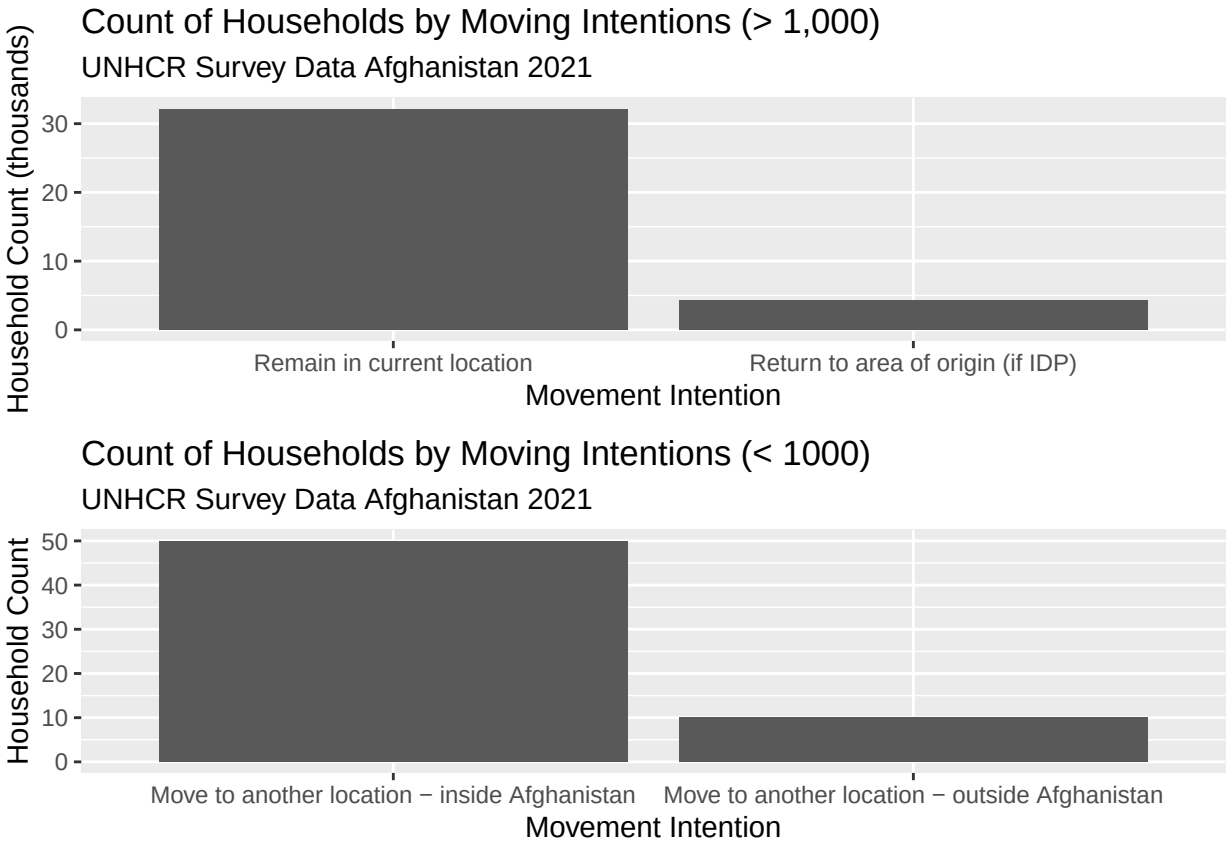
The categories ("Other", " ") were bundled as: (Other*)

tenure	count
Renting	27254
Other*	4271
Owned	2831
Don't know	678
Squatting	674
Insecure tenure	28

(2.4.2) Analyze the distribution of "hh_intentions_movements" which is the household's current movement intentions. Investigate the frequency or proportion of interviewees with different movement intentions. Visualize the distribution using barplot or pie chart to provide a clear representation of the different movement intentions.

```
movement_intentions <- un_survey %>%
  dplyr::select(hh_intentions_movements) %>%
  group_by(hh_intentions_movements) %>%
  summarize(count = n())

top_movement_plot <- movement_intentions %>%
  arrange(-count) %>%
  slice(1:2) %>%
  ggplot() +
  geom_col(aes(x = reorder(hh_intentions_movements, -count), y = count/1000)) +
```

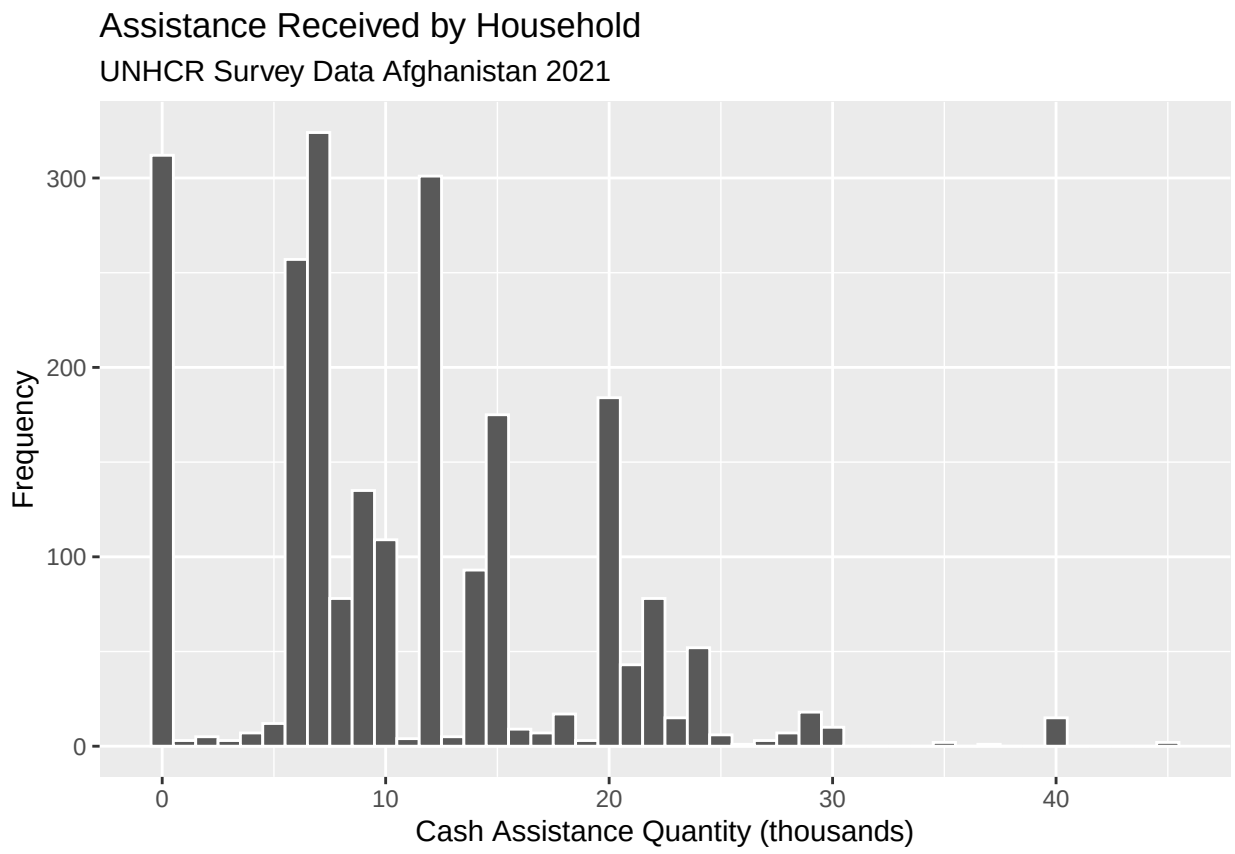
```
    labs(
    title = "Count of Households by Moving Intentions (> 1,000)",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Movement Intention",
    y = "Household Count (thousands)"
  )

bottom_movement_plot <- movement_intentions %>%
  arrange(-count) %>%
  slice(3:4) %>%
  ggplot() +
  geom_col(aes(x = reorder(hh_intentions_movements, -count), y = count)) +
    labs(
    title = "Count of Households by Moving Intentions (< 1000)",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Movement Intention",
    y = "Household Count"
  )


# Create a new page
grid.newpage()
# Next push the vissible area with a layout of 2 columns and 2 row using pushViewport()
pushViewport(viewport(layout = grid.layout(2,1)))
print(top_movement_plot, vp = viewport(layout.pos.row = 1, layout.pos.col = 1))
print(bottom_movement_plot, vp = viewport(layout.pos.row = 2, layout.pos.col = 1))
```



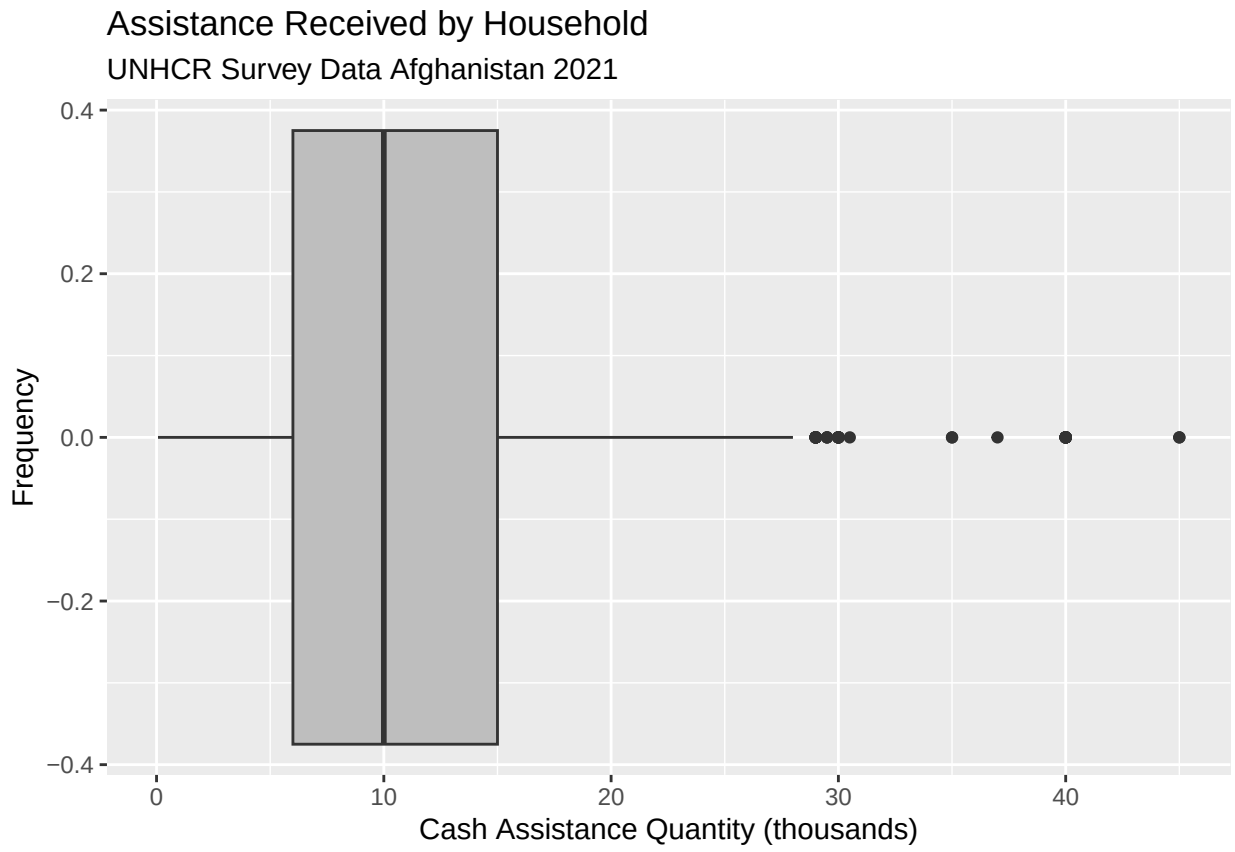
(2.6) Assistance Received By Household

For households that have received assistance with clear cash assistance received ("assistance = Yes" and cash_qtty not null), calculate statistical measures on the distribution of "cash_qtty" such as the mean, 25th percentile, 75th percentile, minimum, and maximum values. This analysis will provide insights into the average amount and the spread of cash assistance received by the households.

```
received_assistance <- un_survey %>%
  dplyr::select(assistance, cash_qtty) %>%
  filter(assistance == "Yes" & !is.na(cash_qtty))


received_assistance %>% ggplot(aes(x=cash_qtty/1000)) +
 geom_histogram(color="white" , binwidth = 1) +
    labs(
    title = "Assistance Received by Household",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Cash Assistance Quantity (thousands)",
    y = "Frequency"
  )
```




```
received_assistance %>% ggplot( aes(x=cash_qtty/1000)) +
    geom_boxplot(fill="gray") +
    labs(
    title = "Assistance Received by Household",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Cash Assistance Quantity (thousands)",
    y = "Frequency"
  )
```



Q3: Geography of Displacement

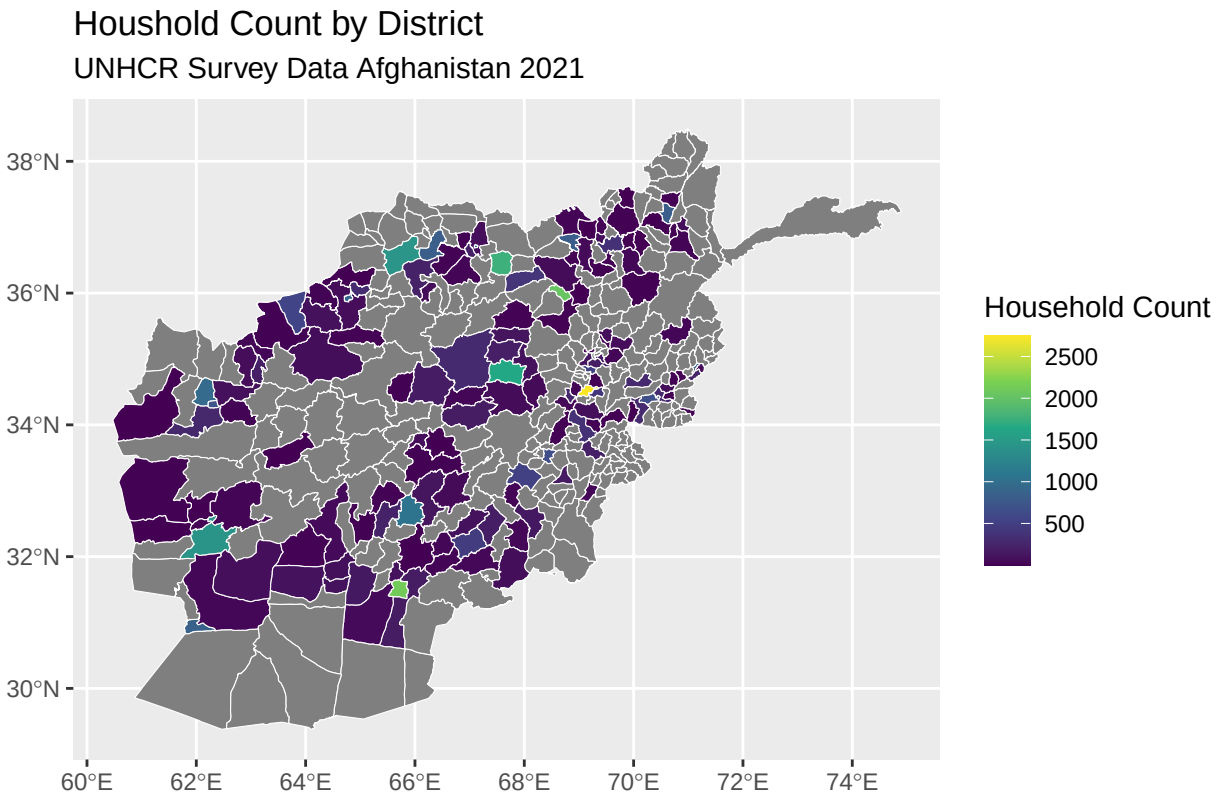
(3.1) Create a geographical map of IDP districts.

```
# Reading in the shapefile. Note the geometry column, this is what stores the coordinates that make up
geography_gdf <- st_read("district398_FixGEo.shp")
```

```
## Reading layer 'district398_FixGEo' from data source
##   '/Users/gabrieltoscano/Desktop/DPSS 2023 Code/Capstone/district398_FixGEo.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 398 features and 3 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 60.47208 ymin: 29.37706 xmax: 74.88945 ymax: 38.49079
## Geodetic CRS:  WGS 84
```

```
#count records (rows) for each district (DISTID)
district_count <- un_survey %>%
  dplyr::select(DISTID) %>%
  group_by(DISTID) %>%
  summarize(count = n())

geo_count_merge <- merge(geography_gdf, district_count, by = "DISTID", all.x = TRUE)
```

```
ggplot() +
  geom_sf(data = geo_count_merge, aes(fill = count), color="white") +
    scale_fill_viridis_c(name = "Household Count", breaks = seq(0, 3000, 500))+   labs(
    title = "Houshold Count by District",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
  )
```




```
#looking at this plot and the data, there are a total of 398 districts but only 149 have data
```

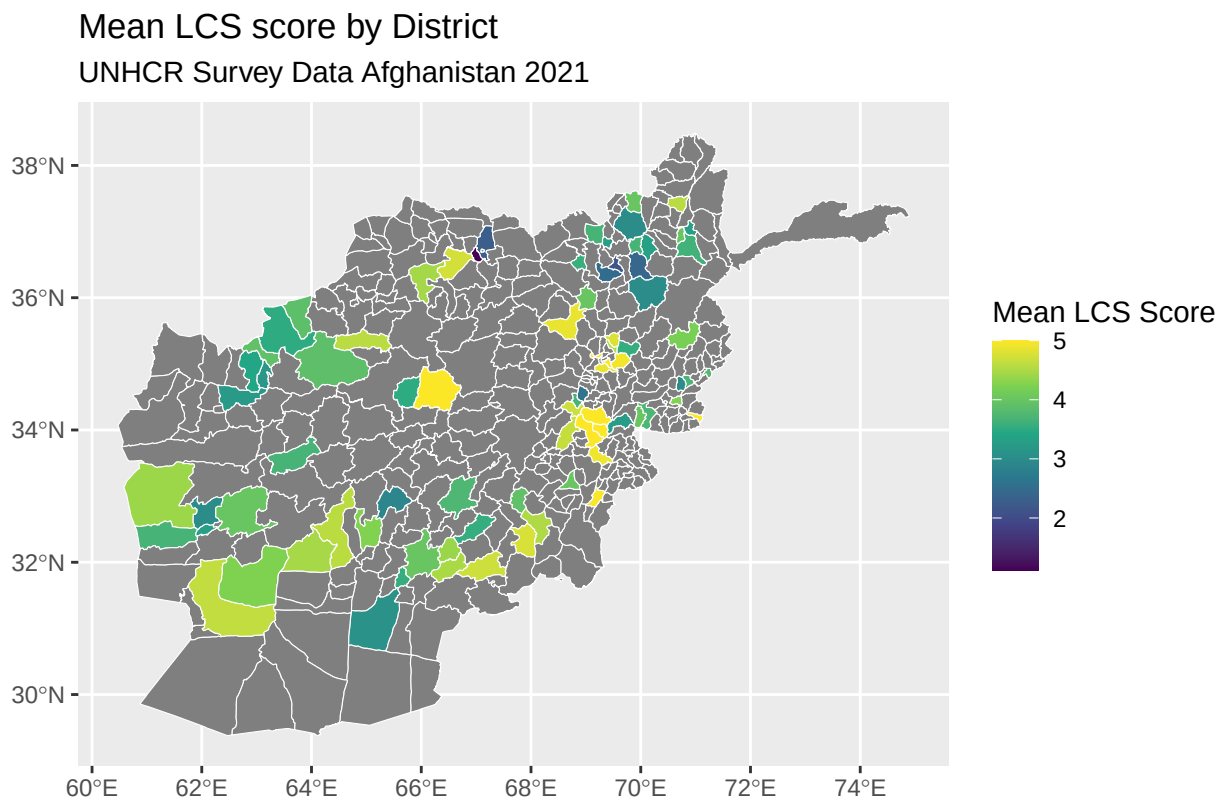
(3.2) From Question 2, pick three preferred challenge areas from these 5 options

(hint: calculate the average for numeric variables, and most frequent category for categorical variables and use either the average or most frequent category to visualize the map): 1.LSC Score 2.RCSI score 3.'shelter_number' , 'hh_intentions_movements'

Looking at LCS scores by district, it's clear that most are under high levels of duress related to life-sustaining sectoral needs. The mean for most districts is 3 or above.

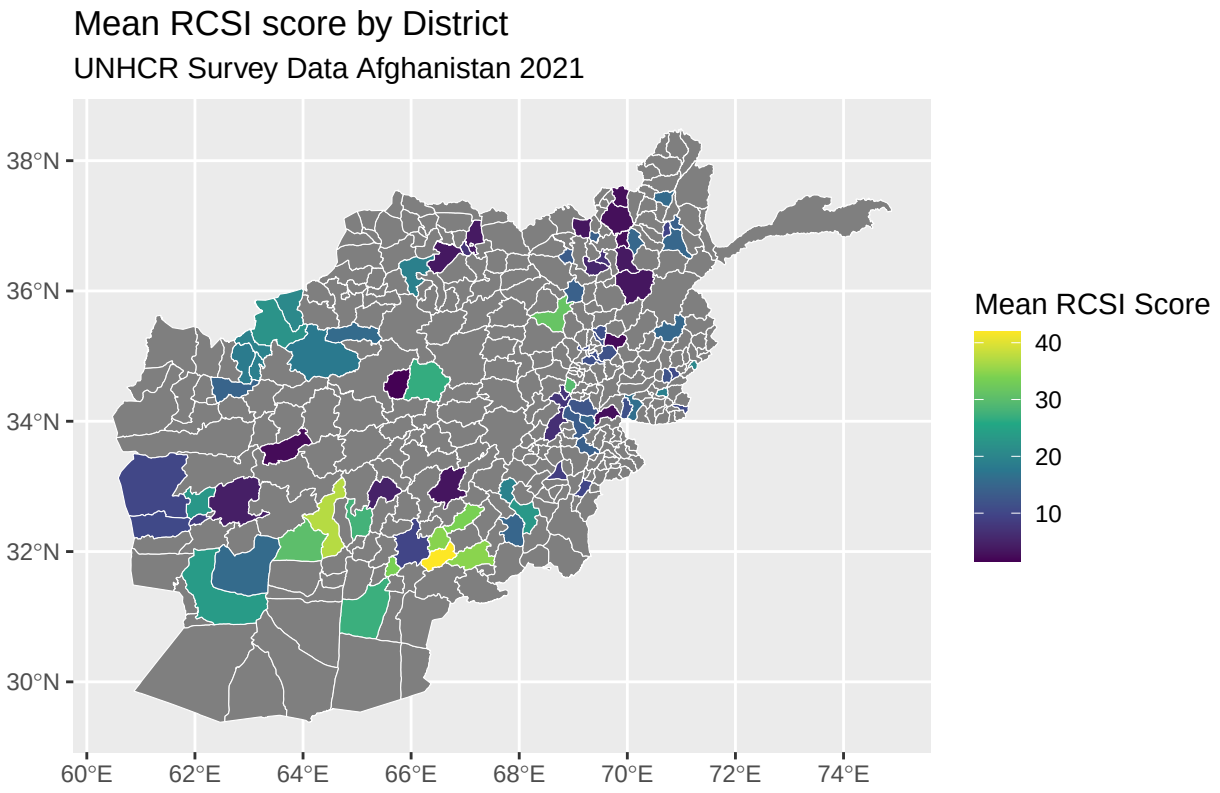
```
district_lcs_avg <- un_survey %>%
  dplyr::select(DISTID, lcs_score) %>%
  group_by(DISTID) %>%
  summarize(avg_lcs = mean(lcs_score))

geo_lcs_count_merge <- merge(geography_gdf, district_lcs_avg, by = "DISTID", all.x = TRUE)
```

```
ggplot() +
  geom_sf(data = geo_lcs_count_merge, aes(fill = avg_lcs), color="white") +
    scale_fill_viridis_c(name = "Mean LCS Score") +
  labs(
    title = "Mean LCS score by District",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
  )
```



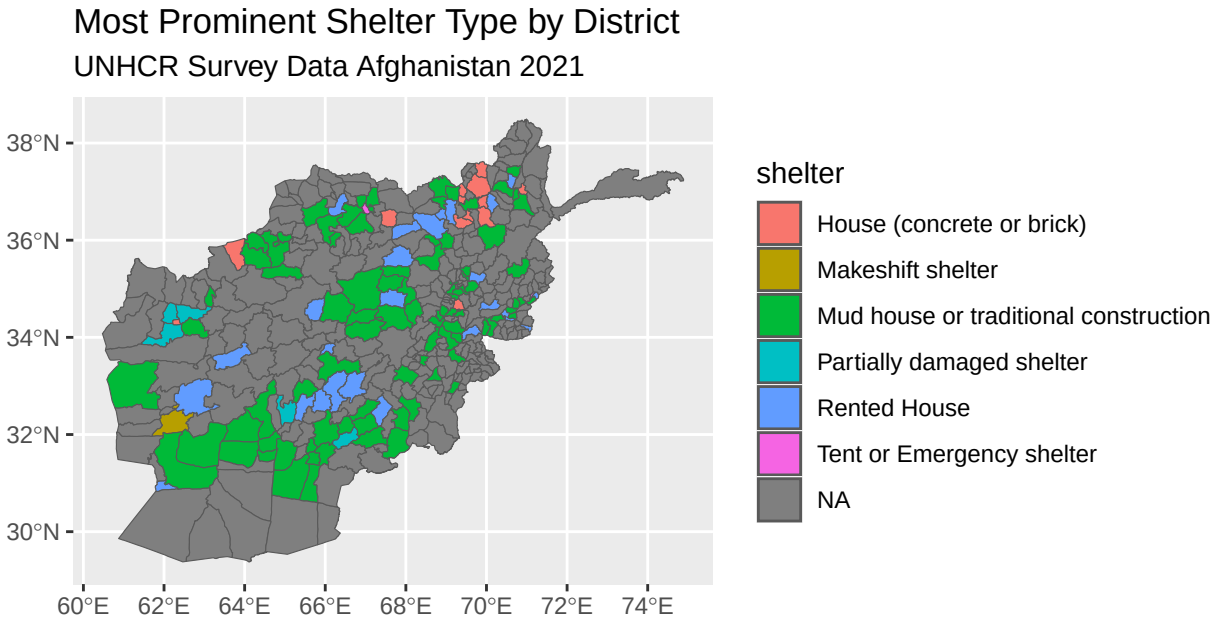
The map of mean RCSI scores, a metric for food insecurity, shows severe food insecurity clusters in several districts. Other clusters show relatively low levels of food insecurity, particularly in northeast districs.

```
district_rcsi_avg <- un_survey %>%
  dplyr::select(DISTID, rcsi_score) %>%
  group_by(DISTID) %>%
  summarize(avg_rcsi = mean(rcsi_score))

geo_rcsi_merge <- merge(geography_gdf, district_rcsi_avg, by = "DISTID", all.x = TRUE)

ggplot() +
  geom_sf(data = geo_rcsi_merge, aes(fill = avg_rcsi), color="white") +
    scale_fill_viridis_c(name = "Mean RCSI Score") +
      labs(
    title = "Mean RCSI score by District",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
  )
```



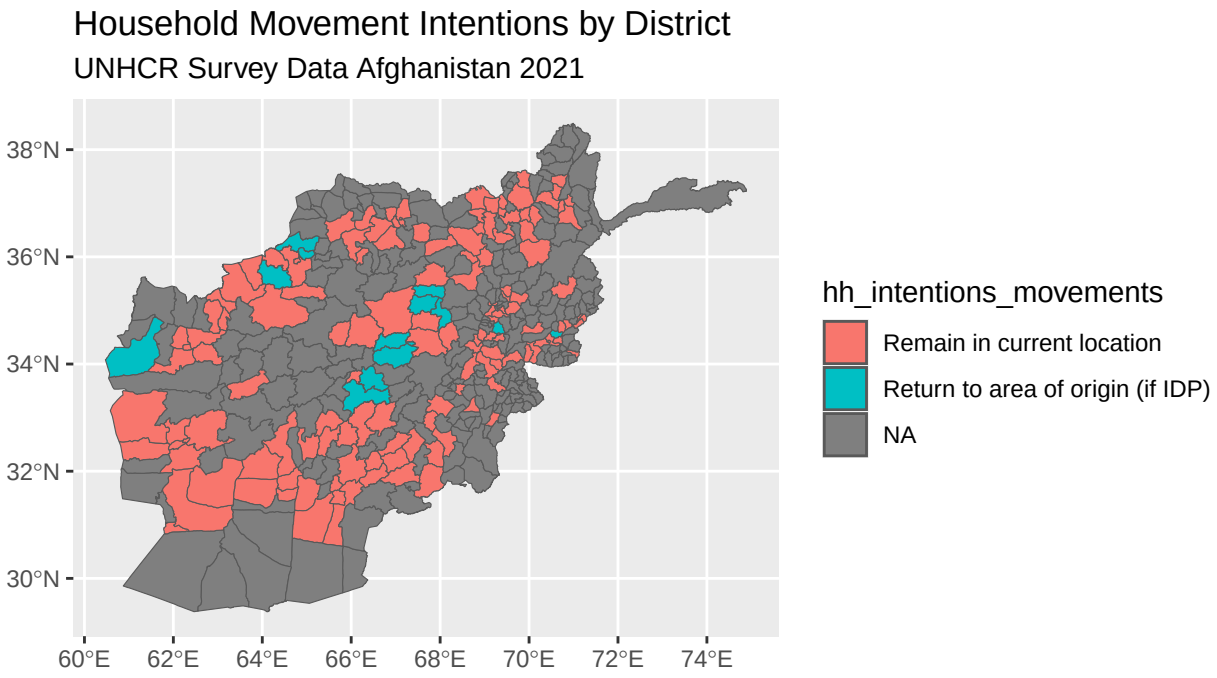
Graphing most frequent category for the type of shelter by district, we can infer urbanization and infrastructural needs based on the kinds of constructions that are most prominent. For the majority of regions, with the exception of the northeastern districts, the most prominent shelter remains traditional and mudhouse constructions.

```
#finding the largest category of shelter by district
district_shelter <- un_survey %>% dplyr::select(DISTID, shelter, shelter_number) %>%
  group_by(DISTID, shelter) %>%
  summarise(count = sum(shelter_number)) %>%
  ungroup() %>%
  drop_na() %>%
  group_by(DISTID, shelter) %>%
  summarise(count = max(count, na.rm = FALSE)) %>%
  group_by(DISTID) %>%
  filter(count == max(count, na.rm=TRUE))

geo_shelter_merge <- merge(geography_gdf, district_shelter, by = "DISTID", all.x = TRUE)

# plot
ggplot() + geom_sf(data = geo_shelter_merge, aes(fill = shelter))+
  labs(
    title = "Most Prominent Shelter Type by District",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
  )
```



An overwhelming majority of IDPs report no intention to return to area of origin. Further analysis might inquire into the conditions in districts where there is a higher level of desire to return and understand IDPs motivations.

```
hh_intentions <- un_survey %>%
  dplyr::select(DISTID, hh_intentions_movements) %>%
  group_by(DISTID, hh_intentions_movements) %>%
  summarize(count = n()) %>%
  group_by(DISTID) %>%
  filter(count == max(count, na.rm=TRUE))

geo_hh_intentions_merge <- merge(geography_gdf, hh_intentions, by = "DISTID", all.x = TRUE)

# plot
ggplot() + geom_sf(data = geo_hh_intentions_merge, aes(fill = hh_intentions_movements))+
  labs(
    title = "Household Movement Intentions by District",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
  )
```



Q4: Linear Model

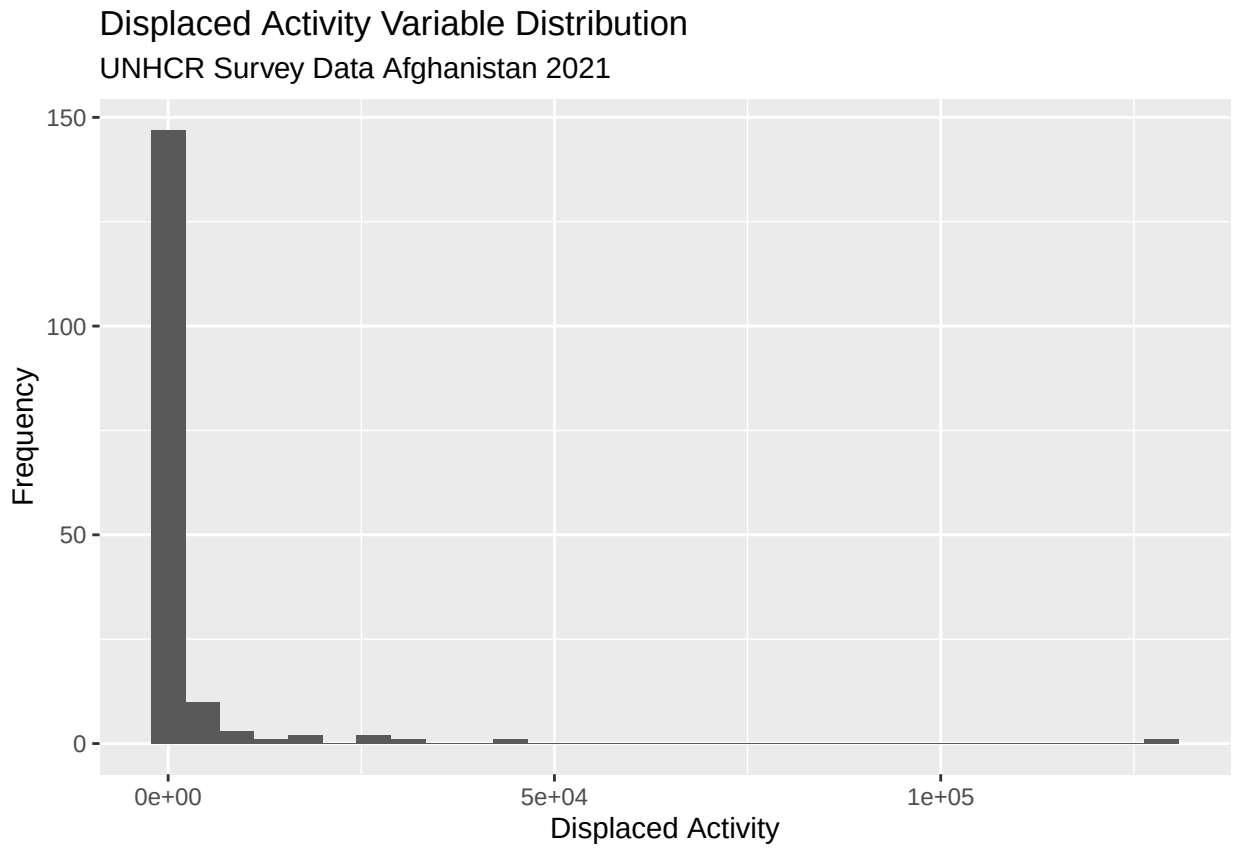
For this part, you will need to combine the datasets: 'features.csv' and the 'displaced_activity.csv'. The key to join is the 'DISTID' feature which represents the unique ID of each district.

(4.1) Construct a Linear Regression or Lasso Regression model

set your response variable to be the displaced_activity. (Hint: You will need to consider all of the factors in the 'features.csv' in your model. Some of them are: urban classification, number of local markets, nighttime light output, military base locations, agricultural activity, religiosity, district healthcare infrastructure, various types of aid (electricity, microfinance, road development, water supplies), as well as geographic features (e.g., ruggedness,water source access, wheat suitability).

```
library(caret)
library(MASS)


#reading files
activity_df <- read_csv('displaced_activity-1.csv') %>%
    rename(DISTID = distid )
features_df <- read_csv("Features-1.csv")


#merging dataframes for linear modeling
```

```r
lm_data <- merge(activity_df, features_df, by="DISTID") %>%
  na.omit() %>%
  dplyr::select(c("DISTID","unique_devices","displaced_persons", "urban", "displaced_activity", "nightli

#Checking distribution of outcome variable
ggplot(lm_data, aes(x=displaced_activity)) +
    geom_histogram() +
  labs(
    title = "Displaced Activity Variable Distribution",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Displaced Activity",
    y = "Frequency")
```




```r
#Strong Covariance between displaced activity and displaced persons, removing displaced persons as a va
cov(lm_data$displaced_activity, lm_data$displaced_persons)
```

```
## [1] 19198335
```

```r
lm_data <- lm_data %>%
  dplyr::select(-displaced_persons)
```

Fitting Linear Models Stepwise Approach The Stepwise approach uses 10-fold cross validation to test every combination of variables and estimate the average prediction error of each of the models. The

RMSE statistical metric is used to compare the models and to automatically choose the best one, where best is defined as the model that minimize the RMSE and maximize the R-squared.

```
##-----------------------------
# Stepwise Approach
##-----------------------------
# Set seed for reproducibility
set.seed(123)
train.control <- trainControl(method = "cv", number = 10)
# Train the model
step.model <- train(displaced_activity ~., data = lm_data,
                    method = "leapBackward",
                    tuneGrid = data.frame(nvmax = 1:10),
                    trControl = train.control
                    )

#See models' results
step.model$results
```

```
##    nvmax     RMSE  Rsquared      MAE   RMSESD RsquaredSD     MAESD
## 1      1 3331.438 0.7426522 1509.006 2485.744  0.2500737  877.9162
## 2      2 3374.398 0.8048275 1353.507 2621.275  0.1705098  973.8021
## 3      3 3145.704 0.6965187 1497.826 2504.958  0.3050172  961.2746
## 4      4 3121.200 0.6910571 1576.210 2372.086  0.2971460  894.7063
## 5      5 3172.320 0.6771536 1626.288 2372.062  0.3003247  945.1770
## 6      6 3203.401 0.6785001 1663.146 2328.952  0.2985405  911.7096
## 7      7 3272.676 0.6735012 1718.968 2350.851  0.2929966  939.8296
## 8      8 3304.006 0.6740568 1757.798 2342.066  0.2904544  953.2483
## 9      9 3330.643 0.6722944 1765.245 2368.850  0.2910937  973.9041
## 10    10 3353.066 0.6650677 1786.964 2409.195  0.2921561 1007.5517
```

```
summary(step.model$finalModel)
```

```
## Subset selection object
## 22 Variables  (and intercept)
##                       Forced in Forced out
## DISTID                    FALSE      FALSE
## unique_devices            FALSE      FALSE
## urban                     FALSE      FALSE
## nightlight                FALSE      FALSE
## shareagri                 FALSE      FALSE
## district_hospital         FALSE      FALSE
## foodmarket_time           FALSE      FALSE
## microfinance_aid          FALSE      FALSE
## news_tvset                FALSE      FALSE
## road_aid                  FALSE      FALSE
## ruggedness                FALSE      FALSE
## project_any               FALSE      FALSE
## watersource_time          FALSE      FALSE
## suitability_rw_wheat      FALSE      FALSE
## affected_opium            FALSE      FALSE
## watersupply_aid           FALSE      FALSE
## madrassa                  FALSE      FALSE
```

```
## total_markets            FALSE      FALSE
## anybase                  FALSE      FALSE
## religious_involved       FALSE      FALSE
## electricity_aid          FALSE      FALSE
## project_agriculture      FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: backward
##          DISTID unique_devices urban nightlight shareagri district_hospital
## 1  ( 1 ) " "    "*"            " "   " "        " "       " "
## 2  ( 1 ) " "    "*"            " "   "*"        " "       " "
## 3  ( 1 ) " "    "*"            "*"   "*"        " "       " "
## 4  ( 1 ) " "    "*"            "*"   "*"        " "       " "
##          foodmarket_time microfinance_aid news_tvset road_aid ruggedness
## 1  ( 1 ) " "             " "               " "        " "      " "
## 2  ( 1 ) " "             " "               " "        " "      " "
## 3  ( 1 ) " "             " "               " "        " "      " "
## 4  ( 1 ) " "             " "               " "        " "      " "
##          project_any watersource_time suitability_rw_wheat affected_opium
## 1  ( 1 ) " "         " "              " "                  " "
## 2  ( 1 ) " "         " "              " "                  " "
## 3  ( 1 ) " "         " "              " "                  " "
## 4  ( 1 ) " "         " "              " "                  " "
##          watersupply_aid madrassa total_markets anybase religious_involved
## 1  ( 1 ) " "             " "      " "           " "     " "
## 2  ( 1 ) " "             " "      " "           " "     " "
## 3  ( 1 ) " "             " "      " "           " "     " "
## 4  ( 1 ) " "             " "      " "           "*"     " "
##          electricity_aid project_agriculture
## 1  ( 1 ) " "             " "
## 2  ( 1 ) " "             " "
## 3  ( 1 ) " "             " "
## 4  ( 1 ) " "             " "
```

```
# Get the coefficients of the best performing model
coef(step.model$finalModel, 3)
```

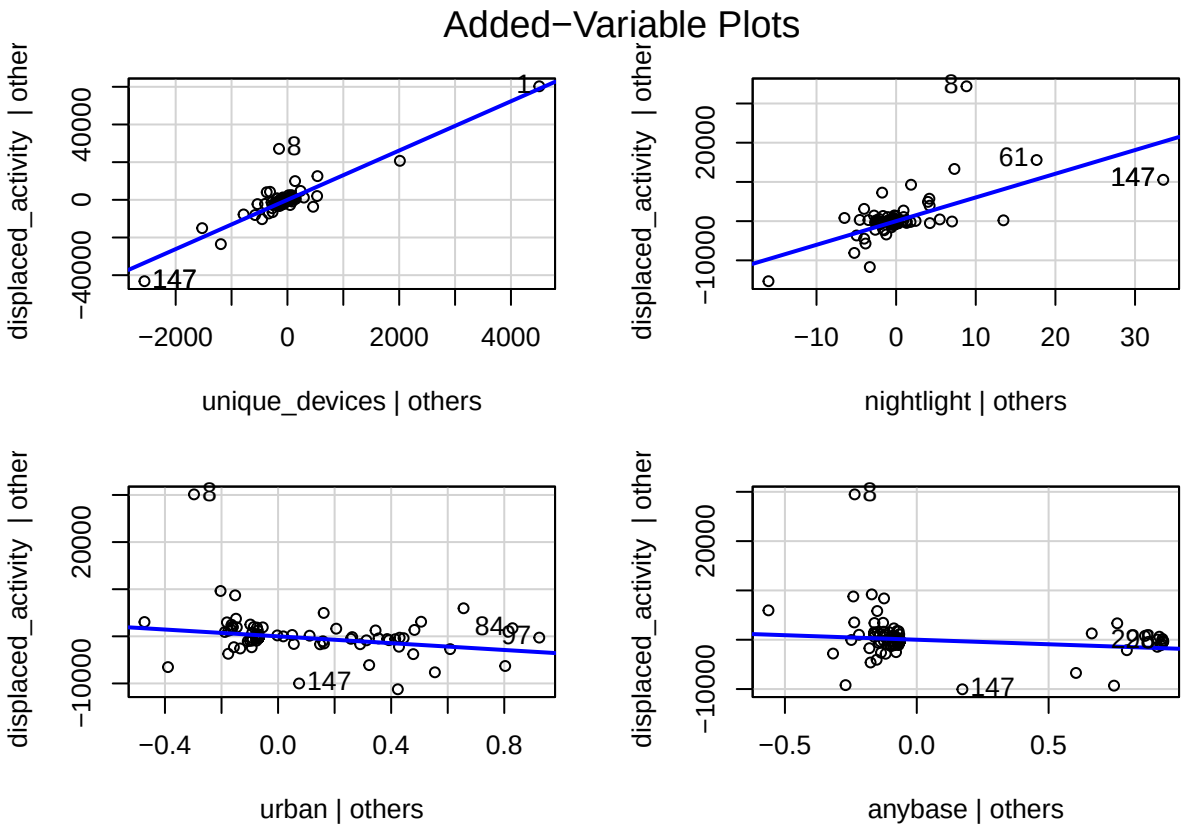
```
##    (Intercept) unique_devices          urban     nightlight
##      374.23435       12.99901    -3924.37569      582.48714
```

```
#using our best fit model from step-wise regression, we
#get our final summary using a single linear model
#There's  covarience here with nighlight and urban variables, which may introduce bias
lm_best_step <-  lm(displaced_activity~ unique_devices + nightlight + urban + anybase, data = lm_data)
summary(lm_best_step)
```

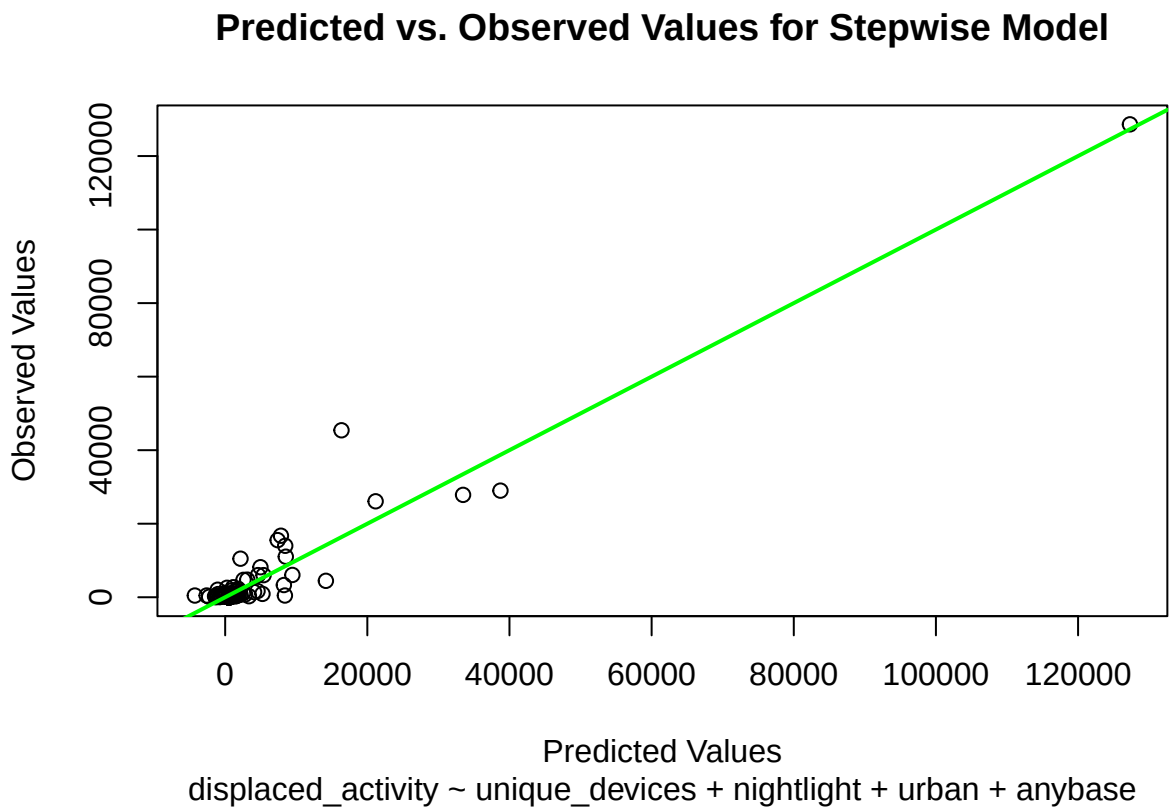
```
##
## Call:
## lm(formula = displaced_activity ~ unique_devices + nightlight +
##     urban + anybase, data = lm_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9731.8  -547.6  -484.9   118.9 29046.4
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      494.0867   272.3094   1.814  0.07145 .
## unique_devices    13.0810     0.5077  25.764  < 2e-16 ***
## nightlight       604.1393    63.5732   9.503  < 2e-16 ***
## urban          -3618.0027  1147.5194  -3.153  0.00192 **
## anybase        -1840.1204   850.2625  -2.164  0.03191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3156 on 163 degrees of freedom
## Multiple R-squared:  0.9231, Adjusted R-squared:  0.9212
## F-statistic: 489.1 on 4 and 163 DF,  p-value: < 2.2e-16
```

```r
#produce added variable plots
avPlots(lm_best_step)
```




```r
# Plot Model predicted values and actual values
quant_plot <- plot(predict(lm_best_step), lm_data$displaced_activity,
     xlab = "Predicted Values",
     ylab = "Observed Values")
title(main = "Predicted vs. Observed Values for Stepwise Model",
      sub = "displaced_activity ~ unique_devices + nightlight + urban + anybase")
abline(a = 0, b = 1, lwd=2,
       col = "green")
```



Theory-based Approach Some research suggests that factors outside of our data, such as violence are highly predictive of internal displacement in Afghanistan. Studies in Yemen and Syria show shelter, access to food, water and healthcare to be strongly correlated with rates of internal displacement.

See: Huynh, B., & Basu, S. (2020). Forecasting Internally Displaced Population Migration Patterns in Syria and Yemen. Disaster Medicine and Public Health Preparedness, 14(3), 302-307. doi:10.1017/dmp.2019.73

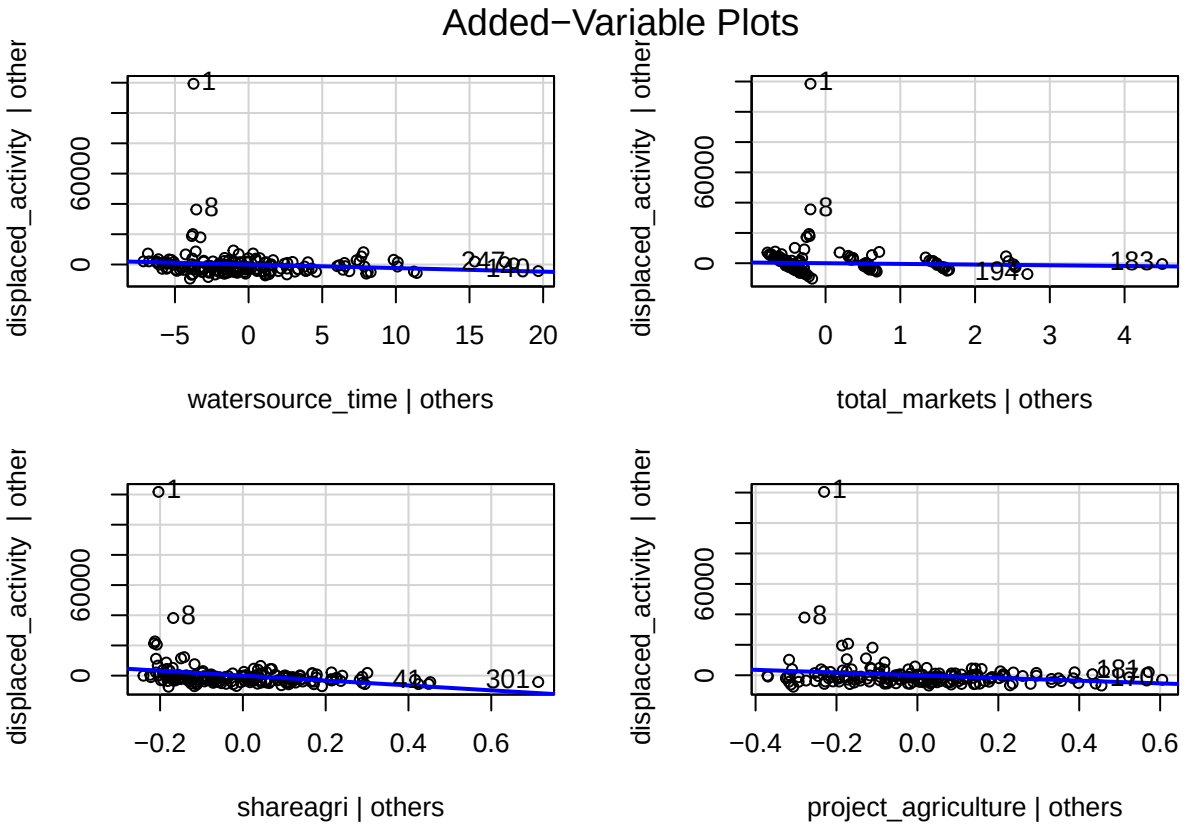
Tai, X.H., Mehra, S. & Blumenstock, J.E. Mobile phone data reveal the effects of violence on internal displacement in Afghanistan. Nat Hum Behav 6, 624–634 (2022). https://doi.org/10.1038/s41562-022-01336-4

```
##-----------------------------
# Theory-based Approach
##-----------------------------

lm_qual <- lm(displaced_activity ~ watersource_time + total_markets + shareagri + project_agriculture, d
summary(lm_qual)
```

```
##
## Call:
## lm(formula = displaced_activity ~ watersource_time + total_markets +
##     shareagri + project_agriculture, data = lm_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -10305  -3963  -1648   1441 118421
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           11176.4     2135.3   5.234 5.05e-07 ***
## watersource_time       -236.5      168.8  -1.401  0.16315
## total_markets          -454.3      967.0  -0.470  0.63910
## shareagri            -16249.2     5144.2  -3.159  0.00189 **
## project_agriculture   -8890.8     4093.3  -2.172  0.03130 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10730 on 163 degrees of freedom
## Multiple R-squared:  0.1108, Adjusted R-squared:  0.08893
## F-statistic: 5.075 on 4 and 163 DF,  p-value: 0.000702
```

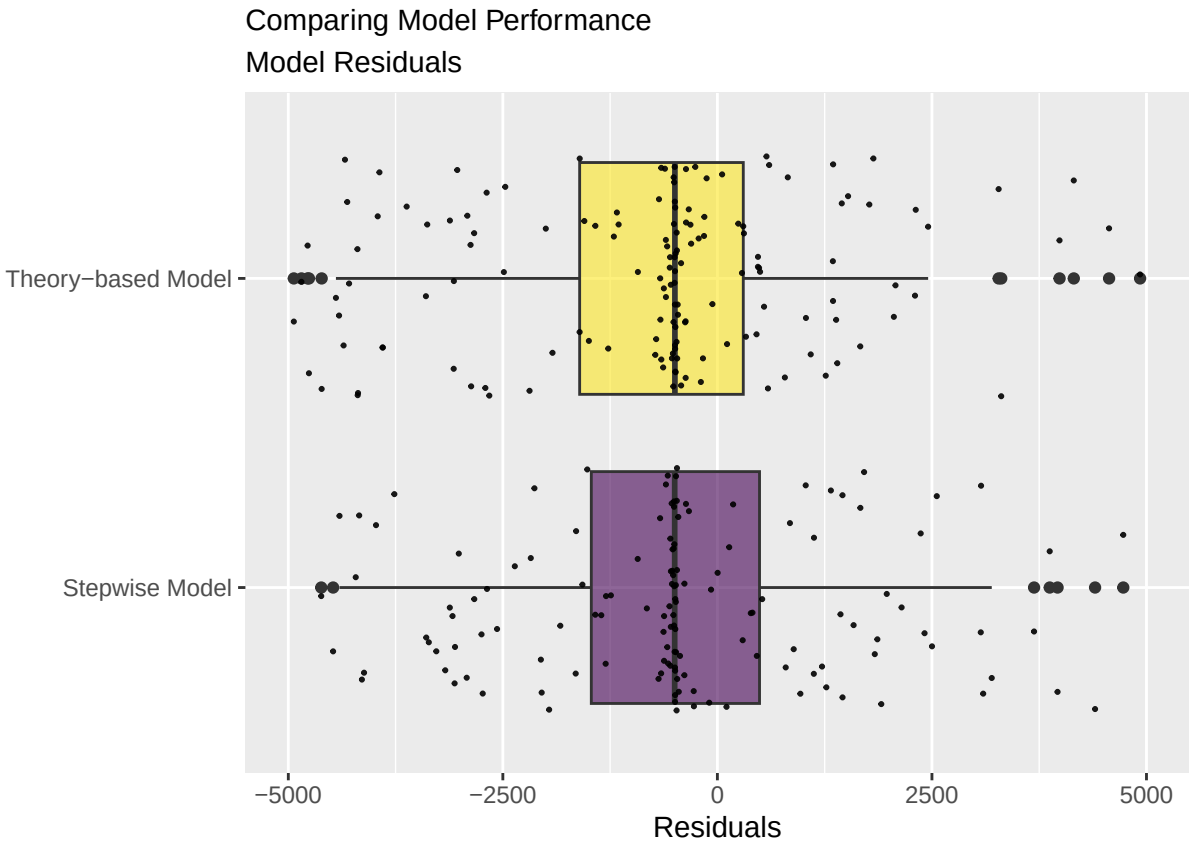
```
#produce added variable plots
avPlots(lm_qual)
```



(4.2) Apply appropriate methods to evaluate the accuracy of your model.

Residuals: The residuals represent the differences between the actual observed values (displaced_activity) and the predicted values from the model. They provide a measure of how well the model fits the data. In this case, the residuals range from -9731.8 to 29046.4, indicating that the model predictions can deviate from the actual values by a substantial amount.

```r
model_names <- c("Stepwise Model", "Theory-based Model")
# Combine the residuals and model identifiers into a data frame
residuals_data <- data.frame(
  Model = factor(rep(model_names)),
  Residuals = c(lm_best_step$residuals, lm_qual$residuals)
)

residuals_data %>% ggplot( aes(x=Model, y=Residuals, fill=Model)) +
    geom_boxplot() +
    scale_fill_viridis(discrete = TRUE, alpha=0.6) +
    geom_jitter(color="black", size=0.4, alpha=0.9) +
    theme(
      legend.position="none",
      plot.title = element_text(size=11)
    ) +
    ggtitle("Comparing Model Performance", subtitle = "Model Residuals") +
    xlab("") +
    ylim(-5000, 5000) +
    coord_flip()
```




```r
confint(lm_best_step)
```

```
##                     2.5 %      97.5 %
## (Intercept)      -43.62207  1031.79546
## unique_devices    12.07844    14.08355
```

```
## nightlight       478.60617   729.67247
## urban          -5883.92280 -1352.08264
## anybase        -3519.06954  -161.17118
```

Given key metrics(R-squared, Standard Errors,F-tats, and P-Values), the model resulting from the stepwise approach is more precise and therefore will be the benchmark for analysis.

$displaced_activity = 494.0867 + 13.0810 * unique_devices + 604.1393 * nightlight - 3618.0027 * urban - 1840.1204 * anybase$

```
summary(lm_best_step)
```

```
##
## Call:
## lm(formula = displaced_activity ~ unique_devices + nightlight +
##     urban + anybase, data = lm_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9731.8  -547.6  -484.9   118.9 29046.4
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      494.0867   272.3094   1.814  0.07145 .
## unique_devices    13.0810     0.5077  25.764  < 2e-16 ***
## nightlight       604.1393    63.5732   9.503  < 2e-16 ***
## urban          -3618.0027  1147.5194  -3.153  0.00192 **
## anybase        -1840.1204   850.2625  -2.164  0.03191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3156 on 163 degrees of freedom
## Multiple R-squared:  0.9231, Adjusted R-squared:  0.9212
## F-statistic: 489.1 on 4 and 163 DF,  p-value: < 2.2e-16
```

Coefficients: Holding other predictors constant, predictor variable coefficients represent the expected change in the outcome variable for a one-unit change in that predictor. All the predictors in our best fit linear model have statistically significant coefficients siince their Pr(>|t|) values are lower than 0.05.

Residual Standard Error (RSE): 3156 on 163 degrees of freedom The RSE value provides an estimate of the standard deviation of the residuals. In this case, a value of 3156 means that, on average, the model's predictions deviate from the actual data points by approximately 3156 units. This suggests lack of precision in its predictions and unexplained variability. Considering the scale of the outcome variable, which ranges from 0 to 128621, an RSE of 3156 may not necessarily be considered too high, however. The non-normal distribution of our outcome variable may explain the large magnitude of the RSE. A more predictive model and precise estimation may require additional data points or the normalization of the distribution of outcome variables within our dataset by imposing an upper and lower bound.

R-squared and Adjusted R-squared: Multiple R-squared: 0.9231, Adjusted R-squared: 0.9212 These values indicate that approximately 92.31% of the variance in the dependent variable (displaced_activity) can be explained by the predictor variables in the model.The R-squared value shows the proportion of the total variability in the outcome variable that is explained by the model's predictors. A high R-squared value suggests that the model captures a significant portion of the data's variability.

F-statistic and p-value:F-statistic: 489.1 on 4 and 163 DF, p-value: < 2.2e-16. This indicates that the overall model is statistically significant and at least one of the predictor variables has a significant relationship with the outcome variable.

Conclusion: The linear model seems to have a reasonably good fit to the data, as indicated by the high R-squared value (92.31%) and the statistically significant F-statistic. The model's coefficients are all statistically significant, suggesting that each predictor variable (unique_devices, nightlight, urban, and anybase) has a meaningful impact on the outcome variable (displaced_activity).

As with any linear model, validation may require analysis of how well the model can predict new data.

(4.3) What can we do to assist IDP populations? List three factors that may have the most impact on the level of displaced activity. Explain why. $displaced_activity = 494.0867 + 13.0810 * unique_devices + 604.1393 * nightlight - 3618.0027 * urban - 1840.1204 * anybase$

If our assumption is that displaced_activity should be reduced, a model-based approach suggests that policies should encourage urbanization and the construction of military bases while decreasing unique devices and nighlight.

The three factors that have the most impact are the existence of bases, urbanization, and the amount of nightlight. Since urbanization and amount of nighlight are often related, covarience between these factors may bias our results. Urbanization in particular, might lead to lower levels of displaced activity since it's usually accompanied by higher governmental and civic society presence, as well as higher security and infrastructure. Moreover, urban zones tend to provide more opportunities for work and education which may also contribute to lower levels of displaced activity.

If we are interested in both improving the situation of already displaced persons and preventing, research suggests that access to health, water, food along with physical safety may led to more desirable outcomes. However, these factors are outside the scope of this current study.

See: Tai, X.H., Mehra, S. & Blumenstock, J.E. Mobile phone data reveal the effects of violence on internal displacement in Afghanistan. Nat Hum Behav 6, 624–634 (2022). https://doi.org/10.1038/s41562-022-01336-4

Huynh, B., & Basu, S. (2020). Forecasting Internally Displaced Population Migration Patterns in Syria and Yemen. Disaster Medicine and Public Health Preparedness, 14(3), 302-307. doi:10.1017/dmp.2019.73

(4.4) Based on your selected most influential factors in 4.3, list five districts where people need the most support on these factors.

Given the model analysis, the selection should target places with the least urbanization and anybase index, and the highest nighlight index.

```
#Find the least urbanized districts (most influential) INCREASE
lowest_urban <- features_df %>%
  dplyr::select(DISTID, urban) %>%
  arrange(urban)

#Find the districts with the least bases INCREASE
lowest_bases <- features_df %>%
  dplyr::select(DISTID, anybase) %>%
  arrange(anybase)

#Find the districts with the most nighlight (least influential) DECREASE
highest_nightlight <- features_df %>%
  dplyr::select(DISTID, nightlight) %>%
  arrange(-nightlight)
```

```r
#put all data frames into list
df_list <- list(lowest_urban, lowest_bases, highest_nightlight)

#merge all data frames in list
intervention_dist <- df_list %>% reduce(full_join, by='DISTID') %>%
  drop_na() %>%
  group_by(urban, anybase, nightlight) %>%
  arrange(-nightlight) %>%
  filter(urban == 0 & anybase == 0) %>%
  ungroup()

intervention_dist_shortlist <- intervention_dist %>%
  slice(1:5)

knitr::kable(intervention_dist_shortlist)
```

DISTID	urban	anybase	nightlight
110	0	0	11.980822
107	0	0	9.670153
2002	0	0	8.293878
102	0	0	7.305234
310	0	0	6.343576

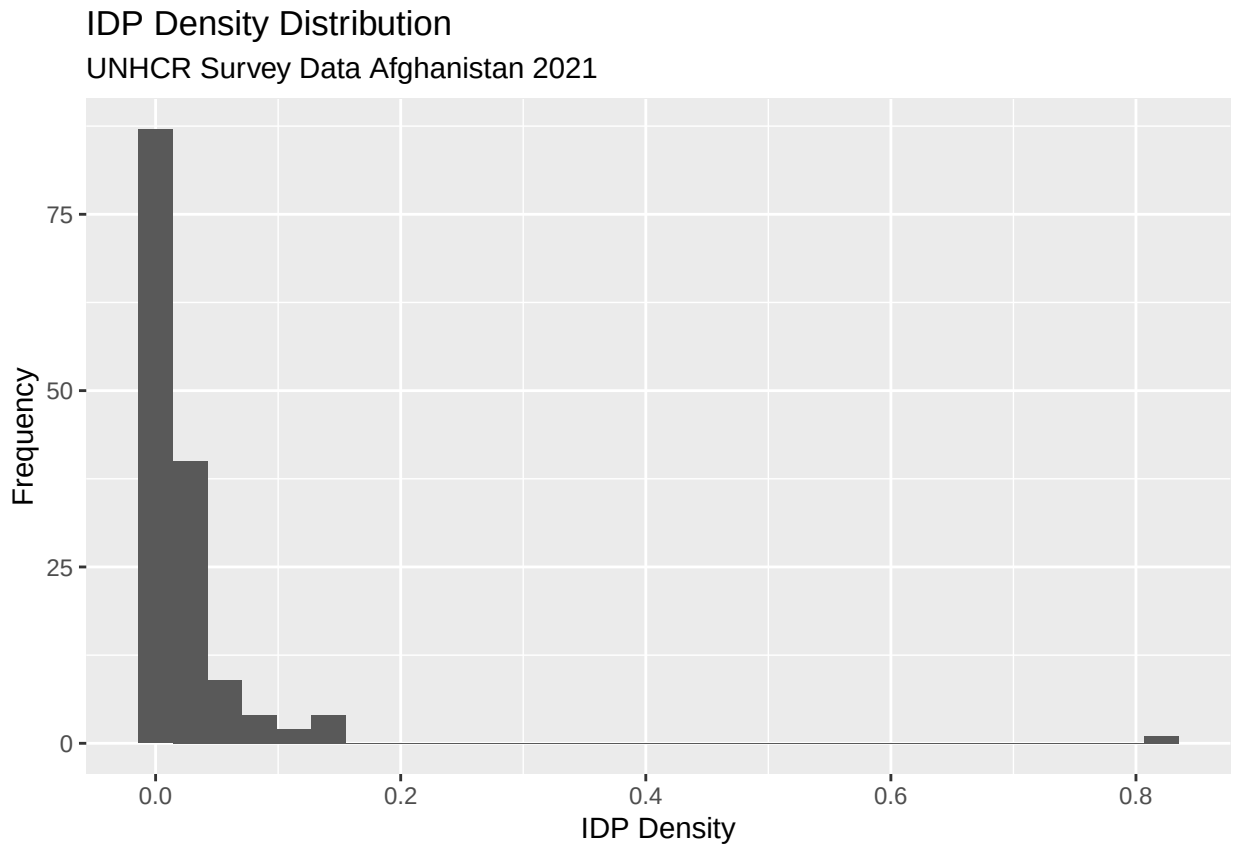
Q5: Over/under Sampling Study

Use population data to study sampling. Determine whether the UNHCR survey in each region is oversampled or undersampled. You may draw a histogram or Kernel Density Estimate (KDE) plot to check the distribution of Density of IDP among different regions. Hint: Density of IDP = Count of IDP (UNHCR_Survey.csv) / total population (population.csv).

```r
population <- read_csv("Population.csv") %>%
  rename(DISTID = distid)

all_dist_population <- population %>%
  summarize(population = sum(total_population))

dist_survey_sample <- un_survey %>%
  dplyr::select(total_members) %>%
  drop_na() %>%
  summarize(survey_population = sum(total_members))
#Population 25354500
#Surveyed population 262740
#Representative sample size = 2401 with a 2% margin of error at 95% confidence level
#Since Survey sample size is 262740, we can say that this is a representative sample size with a 0.19%
```

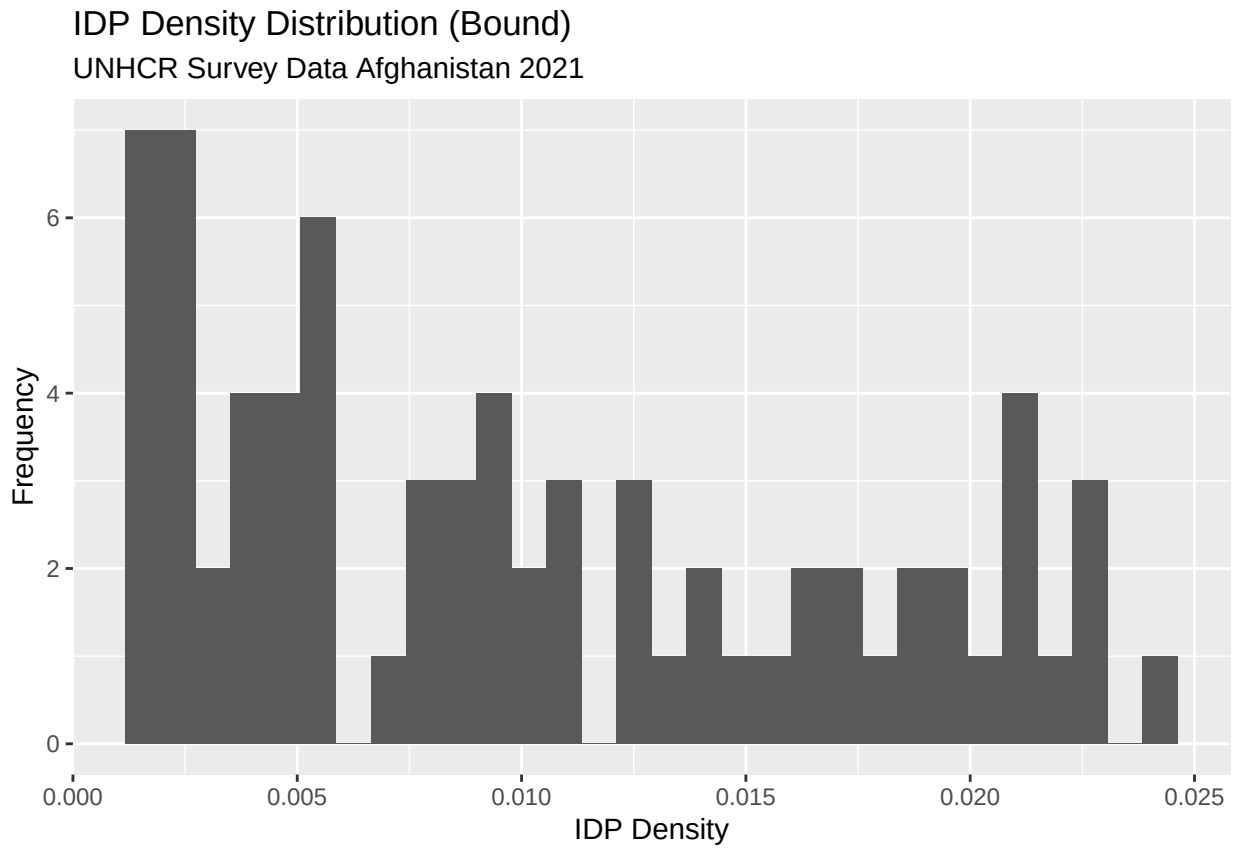
```r
#Dropped NA's prior to sum so we keep as many district info as possible
dist_survey_sample <- un_survey %>%
  dplyr::select(DISTID, total_members) %>%
  drop_na() %>%
  group_by(DISTID) %>%
```

```
  summarise(survey_count = sum(total_members))

dist_survey_sample <- merge(population, dist_survey_sample, by="DISTID") %>%
  mutate(idp_density = survey_count / total_population) %>%
  mutate(bound = "unbound")


#Extreme outliers and a skew
ggplot(dist_survey_sample, aes(x=idp_density)) +
    geom_histogram() +
  labs(
    title = "IDP Density Distribution",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "IDP Density",
    y = "Frequency")
```




```
#Since we don't have a normal distribution, we'll use the IQR to identify outliers by calculating upper
iqr <- 0.0249496 - 0.0014913
# Define the normal data range with lower limit as Q1-0.0001*IQR and upper limit as Q3+0.0001*IQR.
lower_bound = 0.0014913-0.0001*iqr
upper_bound = 0.0249496+0.0001*iqr


dist_survey_sample_bound <- dist_survey_sample %>%
  filter(idp_density <= upper_bound & idp_density >= lower_bound) %>%
```

```
  mutate(type = "bound")

ggplot(dist_survey_sample_bound, aes(x=idp_density)) +
    geom_histogram() +
  labs(
    title = "IDP Density Distribution (Bound)",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "IDP Density",
    y = "Frequency")
```




```
#Comparing IDP Densities
#True distribution
summary(dist_survey_sample$idp_density)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000885 0.0014913 0.0084831 0.0255687 0.0249496 0.8205512
```

```
# sd(dist_survey_sample$idp_density)

#Bound distribution
summary(dist_survey_sample_bound$idp_density)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001507 0.004021 0.008483 0.010097 0.016341 0.024179
```

```
# sd(dist_survey_sample_bound$idp_density)
```

```
#we'll consider any districts under the lower bound as undersampled, and any over the upper_bound as ov
dist_survey_oversampled <- dist_survey_sample %>%
  dplyr::select(DISTID, idp_density) %>%
  filter(idp_density >= upper_bound)

dist_survey_undersampled <- dist_survey_sample %>%
  dplyr::select(DISTID, idp_density) %>%
  filter(idp_density <= lower_bound)
```

Which places are over/under sampled in the survey? Oversampled Districts

```
knitr::kable(dist_survey_oversampled)
```

DISTID	idp_density
105	0.0533917
110	0.0563636
401	0.0487212
501	0.0298929
601	0.0305647
613	0.0285807
801	0.0276077
802	0.0418958
1001	0.1499074
1015	0.0630189
1101	0.0895827
1301	0.0709528
1507	0.8205512
1701	0.0544589
1708	0.1431111
1801	0.0783949
1813	0.0403239
1905	0.0608022
2101	0.0884686
2201	0.1161566
2401	0.0388464
2402	0.0512760
2407	0.0269649
2501	0.1446064
2502	0.0338028
2503	0.0297792
2509	0.0288048
2601	0.0995486
2605	0.0374747
2801	0.1506057
2803	0.0287819
2804	0.0277882

DISTID	idp_density
2807	0.0698283
2901	0.0257202
2903	0.0500816
3301	0.0294536
3302	0.0288060

Undersampled Districts

```
knitr::kable(dist_survey_undersampled)
```

DISTID	idp_density
302	0.0003221
407	0.0008237
504	0.0013675
506	0.0012121
608	0.0004703
804	0.0004193
818	0.0006693
1013	0.0009740
1102	0.0003053
1115	0.0003150
1118	0.0007568
1203	0.0010646
1205	0.0010569
1207	0.0007018
1208	0.0003137
1209	0.0002359
1309	0.0003125
1402	0.0007042
1405	0.0011790
1407	0.0001456
1606	0.0003095
1608	0.0008520
1609	0.0012387
1803	0.0013763
1808	0.0000885
1907	0.0009957
2004	0.0012923
2009	0.0003492
2107	0.0014760
2108	0.0003344
2303	0.0003463
2403	0.0005687
2604	0.0013962
2704	0.0007122
2709	0.0014151
3402	0.0002432
3409	0.0002786

Question 6: Brief write-up

Based on your analysis, write a brief write-up (no more than 250 words) to summarize your valuable findings. In your write-up, you should provide at least three recommendations on how to support the internally displaced population in Afghanistan.

Utilizing 2021 UNHRC survey data, this study describes the situation of IDPs in Afghanistan and prescribes policy recommendations based on statistical data analysis. Given the total district population (25,354,500) and our surveyed population (26,2740), the survey sample is representative with a 0.19% margin of error at a 95% confidence level.

By studying distributions and relationships between variables related to IDP populations, food/water security, and life-sustaining needs, we try to describe the situation of IDPs at a district level. For example, LCS and RSCI scores are positively correlated. Among the ten districts with the highest IDP presence, there is a five-fold difference between the lowest and highest IDP populations. Future policy and humanitarian efforts should consider targeting these high-need areas.

The study then provides a multivariate linear regression model—-a generalized statistical approach for predicting the effects of policy changes. In order to decrease 'displaced_activity' the model-based approach suggests policies that:

1. Encourage urbanization, particularly in districts with the highest IDP populations
2. Proliferate and strengthen security forces (i.e. bases)
3. Decrease nightlight

It should be noted that since urbanization and the amount of nightlight are often related, covariance between these (and other) factors may bias our results.

If we are interested in both improving the situation of already displaced persons and preventing increases in IDPs, research suggests that access to health, water, and food along with physical safety may lead to more desirable outcomes. However, these factors are outside the scope of this current study.

Bonus Points: Healthcare Barrier Study (2+8=10 points) Note: This is optional

(1) Sum all the healthcare_barriers variables from healthcare_barriers1 to healthcare_barriers12 to calculate the total healthcare barrier.

```
un_survey_2 <- un_survey %>%
  dplyr::select(DISTID, healthcare_barriers1, healthcare_barriers2, healthcare_barriers3, healthcare_ba

barrier_list <- c("healthcare_barriers1", "healthcare_barriers2", "healthcare_barriers3", "healthcare_b

#replace all NA's with zeros
un_survey_2 <- un_survey_2 %>%
  mutate_all(~replace_na(., 0))

#add all healthcare barrier row values
un_survey_2 <- un_survey_2 %>%
  mutate(healthcare_barriers = rowSums(un_survey_2[,barrier_list])) %>%
  dplyr::select(DISTID, healthcare_barriers)
```

(2) Analyze the impact of the healthcare barrier on IDP (Internally Displaced Persons). Investigate the relationship between the level of healthcare barriers and the frequency of IDP among different districts. Will a lower healthcare barrier lead to a decrease in the number of IDP in that district?

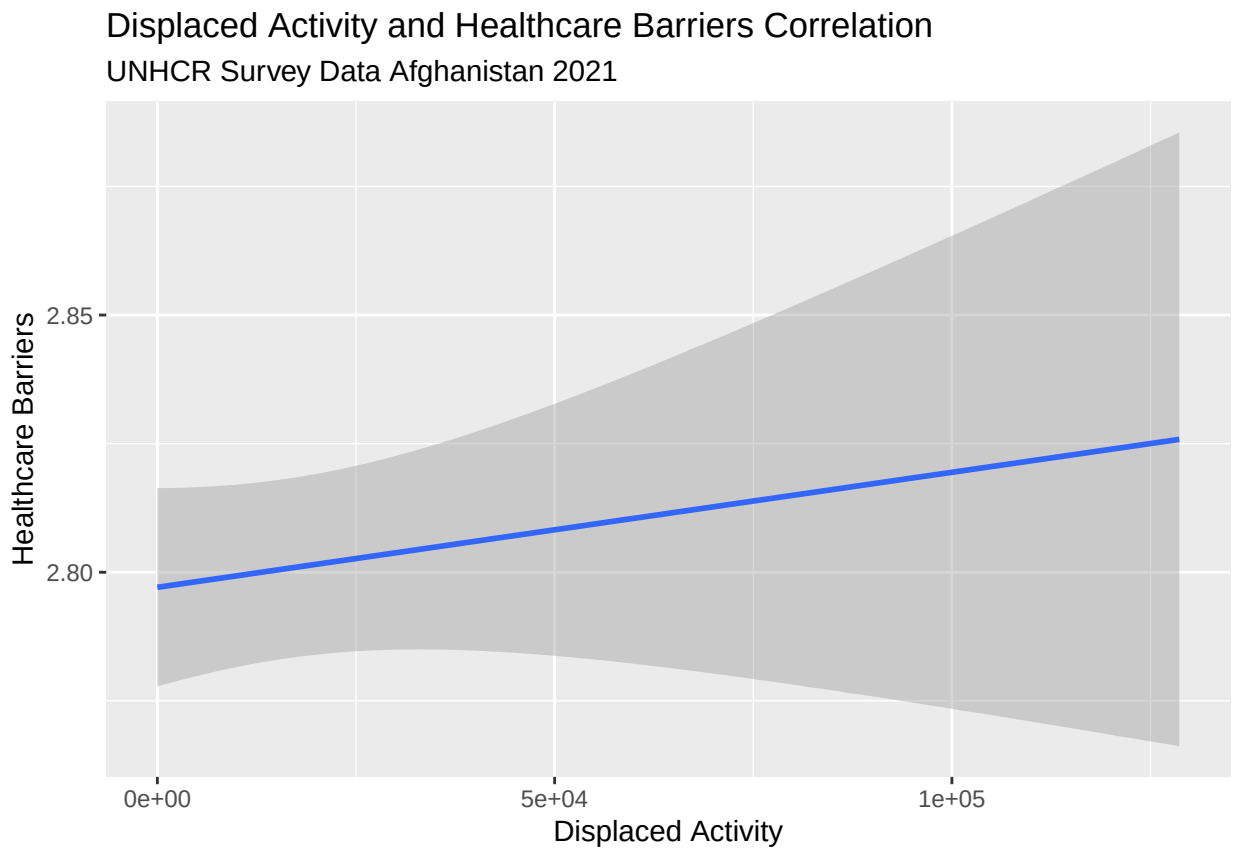
There is a positive correlation between displaced activity and healthcare barriers. However, utilizing healtcare variables within a linear model, it appears that healthcare barriers are not as predictive (Coeff: 97.58, P-val: 0.387, Rquared 2.182e-05) as the effects of variables previously discussed.

```
healthcare_model_data <- merge(lm_data, un_survey_2, by="DISTID")

cor.test(healthcare_model_data$healthcare_barriers, healthcare_model_data$displaced_activity, method =
```

```
##
##  Pearson's product-moment correlation
##
## data:  healthcare_model_data$healthcare_barriers and healthcare_model_data$displaced_activity
## t = 0.8645, df = 34258, p-value = 0.3873
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.005918689  0.015258958
## sample estimates:
##         cor
## 0.004670658
```

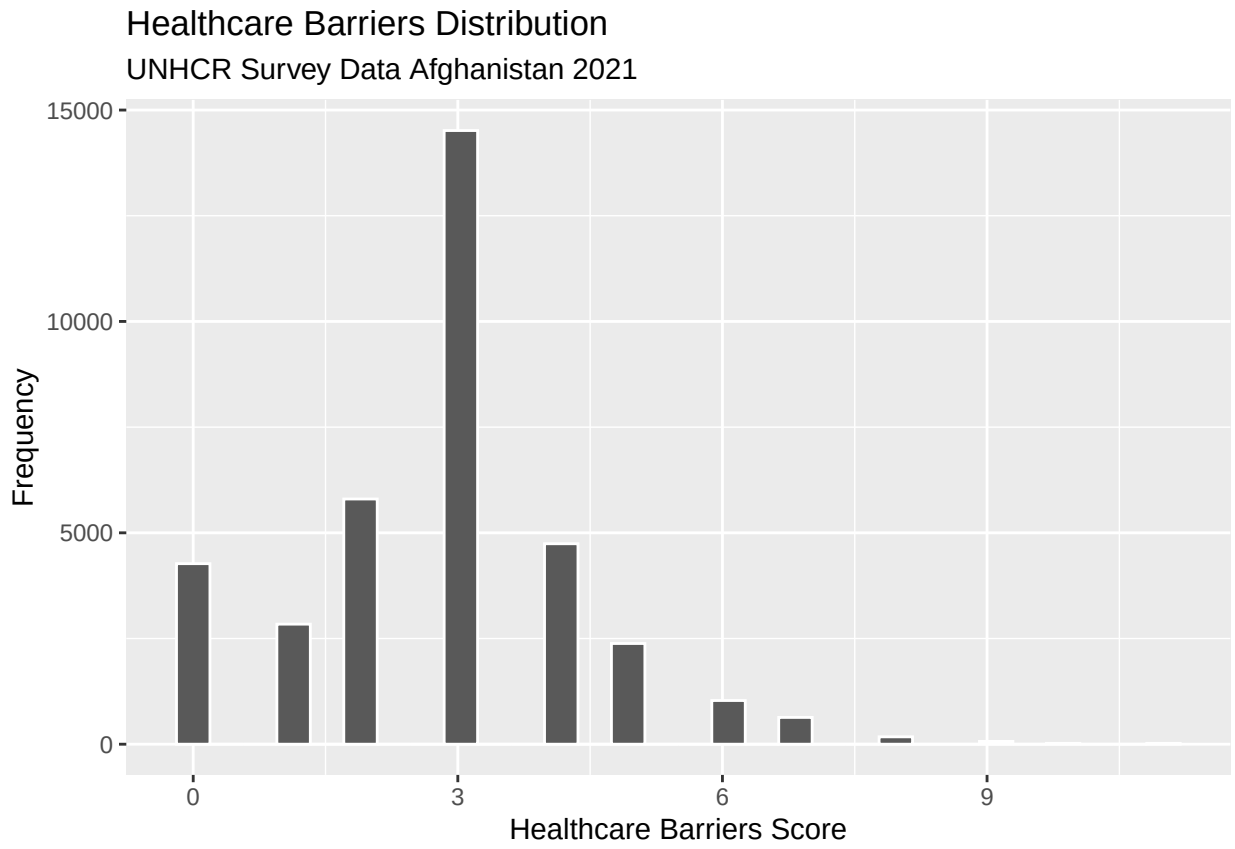
```
healthcare_model_data %>% ggplot() +
  geom_smooth(mapping = aes(x = displaced_activity, y = healthcare_barriers), method=lm) +
    labs(
    title = "Displaced Activity and Healthcare Barriers Correlation",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Displaced Activity",
    y = "Healthcare Barriers"
  )
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(un_survey_2, aes(x=healthcare_barriers)) +
 geom_histogram(color="white") +
    labs(
    title = "Healthcare Barriers Distribution",
    subtitle = "UNHCR Survey Data Afghanistan 2021",
    x = "Healthcare Barriers Score",
    y = "Frequency"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
healthcare_model <- lm(displaced_activity ~ healthcare_barriers, data=healthcare_model_data)
summary(healthcare_model)
```

```
##
## Call:
## lm(formula = displaced_activity ~ healthcare_barriers, data = healthcare_model_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -16637 -15683 -14176  -5089 112797
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         15823.83     366.84  43.136   <2e-16 ***
## healthcare_barriers    97.58     112.87   0.864    0.387
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34450 on 34258 degrees of freedom
## Multiple R-squared:  2.182e-05,  Adjusted R-squared:  -7.375e-06
## F-statistic: 0.7474 on 1 and 34258 DF,  p-value: 0.3873
```

```r
geo_count_merge <- merge(geography_gdf, healthcare_model_data, by = "DISTID", all.x = TRUE)

ggplot() +
```

```
geom_sf(data = geo_count_merge, aes(fill = healthcare_barriers), color="white") +
  scale_fill_viridis_c(name = "Healthcare Barrier Score", breaks = seq(0, 11, 1)) +
labs(
  title = "Healthcare Score by District",
  subtitle = "UNHCR Survey Data Afghanistan 2021",
)
```