Text Mining of Letters of Intent

Claudia Abreu Lopes, Thepan Ravindran, Fui-Ching Lam and Evangelia Berdou

27/5/2020

Objective

The objective of this project is to develop a method and analytical tools to automatically extract information from batches of research proposals submitted as part the final exam of the TDR MOOC on Implementation Research (IR). The findings consist of mapping the implementation research needs through interactive information products for public access (method, code, maps, and dashboards) to promote IR as an essential prerequisite to effective public health interventions

1 Corpus Preparation

1.1 Import LoIs in pdf format into a dataframe

```
rm(list = ls())

# Upload LoIs batches
loi_raw_1 <- readtext::readtext("Batch1")
loi_raw_2 <- readtext::readtext("Batch2")
loi_raw_3 <- readtext::readtext("Batch3")
loi_raw_4 <- readtext::readtext("Batch4")
loi_raw_5 <- readtext::readtext("Batch5")
loi_raw_6 <- readtext::readtext("Batch6")

# Add batch number to LoIs
loi_raw_1$batch_number <- 1
loi_raw_2$batch_number <- 2
loi_raw_3$batch_number <- 3
loi_raw_4$batch_number <- 4
loi_raw_5$batch_number <- 5
loi_raw_6$batch_number <- 6

# Add serial number to LoIs
loi_raw_1$serial_number <- paste("loi_", 1:nrow(loi_raw_1), sep = "")
loi_raw_2$serial_number <- paste("loi_", 1:nrow(loi_raw_2), sep = "")
loi_raw_3$serial_number <- paste("loi_", 1:nrow(loi_raw_3), sep = "")
loi_raw_4$serial_number <- paste("loi_", 1:nrow(loi_raw_4), sep = "")
loi_raw_5$serial_number <- paste("loi_", 1:nrow(loi_raw_5), sep = "")
loi_raw_6$serial_number <- paste("loi_", 1:nrow(loi_raw_6), sep = "")

# Combine all batches in one dataset
loi_raw <- rbind(loi_raw_1, loi_raw_2, loi_raw_3, loi_raw_4,
```

```
    loi_raw_5, loi_raw_6)

# Remove LoIs batches
rm(loi_raw_1, loi_raw_2, loi_raw_3, loi_raw_4, loi_raw_5, loi_raw_6)
```

1.2 Clean LoIs text

Change text to lower case, remove reference list, punctuation and weblinks.

```
# loi_raw$text <- replace_non_ascii(loi_raw$text) #Convert to
# ASCII
loi_raw$text <- tolower(loi_raw$text)  #Change to lowercase
loi_raw$text <- gsub("References.+", "", loi_raw$text)  #Remove reference list
loi_raw$text <- gsub("[[:punct:]]+", " ", loi_raw$text)  #Remove punctuation
loi_raw$text <- gsub("http[[:alnum:][:punct:]]*", "", loi_raw$text)  #Remove punctuation
loi_raw$text <- gsub("http[^[:space:]]*", "", loi_raw$text)  #Remove URL/Hyperlink
loi_raw$text <- gsub("(f|ht)tp\\S+\\s*", "", loi_raw$text)  #Remove URL/Hyperlink
```

2 Text Mining Diseases

2.1 Compile list of Neglected Tropical Diseases

The list of Neglected Tropical Diseases was obtained from the WHO website: http://www.who.int/neglected_diseases/diseases/summary/en/ (accessed in December 2019)

```
# Create regular expressions for neglected tropical diseases
diseases_list <- "\\bhiv\\b|\\btb\\b|tuberc[uo]los[ei]s*|malaria|influenzae|influenz[ae]
|dengue|rab[gi]es|trachoma|\\byaws\\b|lepr[eo]sys*|chagas|leishmani[oa]ses|taeniasis
|neurocysticercos[ei]s|dracuncul[oi]as[ei]s|echinococcos[ei]s|onchocerc[oi]as[ei]s
|schistosomias[ei]s|mycetoma|buruli\\s+ulcer|human\\s+immunodeficiency\\s+viruses
|trypanosomiasis|sleeping\\s+sickness|guinea\\s+worm|foodborne\\s+trematodiases
|lymphatic\\s+filariasis|river\\s+blindness|soil\\s+transmitted\\s+helminthiases"
```

2.2 Extract diseases from the text

This code extract diseases words above and expressions based on detection techniques (including flowcharts and image captions). LoIs with diseases not detected were manually inputted and a random sample of LoIs were manually validated. Since several diseases can be mentioned in the LoIs, including literature review, only diseases with frequencies above the average number of mentions per LoIs were selected (detecting diseases that were a focus on the LoIs).

```
# Extract list of diseases
loi_raw$diseases <- str_extract_all(loi_raw$text, diseases_list)

# Fill empty cells with 'other'
loi_raw$diseases <- str_replace_all(loi_raw$diseases, "character\\(0\\)",
    "other")

# Create long dataset with one disease per row
loi_diseases <- cSplit(loi_raw, "diseases", ",", "long")
loi_diseases$diseases <- str_replace_all(loi_diseases$diseases,
    "\\btb\\b", "tuberculosis")
```

```
loi_diseases$diseases <- str_replace_all(loi_diseases$diseases,
    "\\binfluenzae", "influenza")
loi_diseases$diseases <- str_replace_all(loi_diseases$diseases,
    "\\btuberculosi", "tuberculosis")
loi_diseases$diseases <- str_replace_all(loi_diseases$diseases,
    "\\s+", " ")
loi_diseases$diseases <- str_replace_all(loi_diseases$diseases,
    "c\\(", "")
loi_diseases$diseases <- str_replace_all(loi_diseases$diseases,
    "\\)", "")
loi_diseases$diseases <- str_replace_all(loi_diseases$diseases,
    "\"", "")

# Collapse long dataset to obtain disease frequencies per
# LoIs
loi_count_1 <- loi_diseases
loi_count_1$tally <- 1
loi_count_1 <- aggregate(tally ~ doc_id + serial_number + batch_number +
    diseases, loi_count_1, sum)

# Add variables for each diseases with frequencies
loi_column_1 <- cast(loi_count_1, doc_id + serial_number + batch_number ~
    diseases)

# Delete mentions of other diseases and create dummy
# variables
loi_column_1$average_disease <- rowMeans(loi_column_1[, 4:22],
    na.rm = T)
bool_disease <- ifelse(loi_column_1[, 4:22] >= loi_column_1$average_disease,
    1, NA)
loi_column_1[, 4:22] <- bool_disease
```

2.3 Visualise diseases in LoIs

Each LoIs was classified into one or more diseases and the percentage of LoIs that focused on that disease calculated. The code below produces a bar chart of percentages for each disease.

```
viz_disease <- as.data.frame(prop.table(table(loi_count_1$diseases)) *
    100)

viz_disease$Var1 <- str_replace_all(viz_disease$Var1, "character\\(0",
    "unidentified")

names(viz_disease)[names(viz_disease) == "Freq"] <- "Percentage"
names(viz_disease)[names(viz_disease) == "Var1"] <- "Group"
viz_disease <- subset(viz_disease, viz_disease$Group != "unidentified")

g1 <- ggplot(viz_disease, aes(x = reorder(Group, Percentage),
    y = Percentage)) + geom_bar(stat = "identity", fill = "indianred2") +
    theme_minimal() + coord_flip() + theme(plot.title = element_blank(),
    axis.title.x = element_blank(), axis.title.y = element_blank())

print(g1)
```

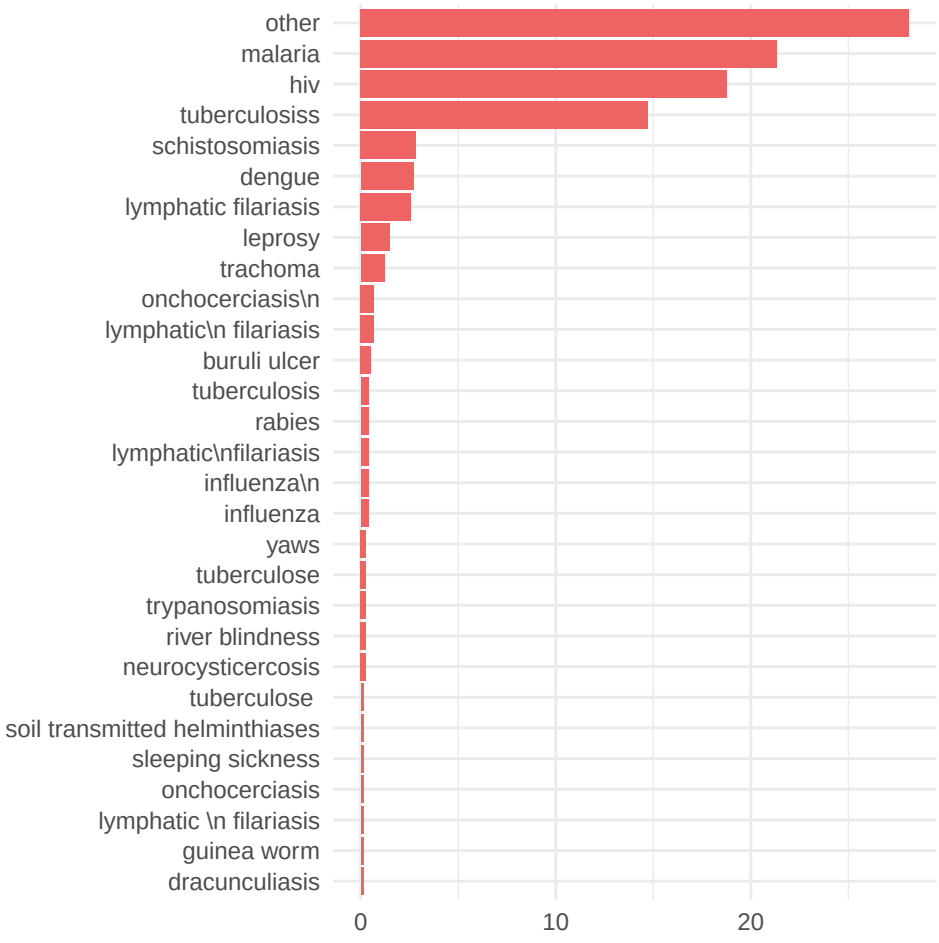

Figure 1: Percentage of LoIs focusing on the disease

3 Text Mining Countries

The list of countries and income groups was obtained from the World Bank website: https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lending-groups

3.1 Compile a list of countries, WHO regions and income groups

Compile a list of countries were classified according to the WHO region and World Bank income group.

```
# Import file with Country, WHO Region and WB Income Group
country_wb_who <- read.csv("Country List.csv")

head(country_wb_who[, -1])
```

```
##   country_corrected      who_region     wb_income_group
## 1       afghanistan        WHO_EMRO          Low income
## 2           albania        WHO_EURO Upper middle income
## 3           algeria        WHO_AFRO Upper middle income
## 4    american samoa NA (WHO Region) Upper middle income
## 5           andorra        WHO_EURO         High income
## 6            angola        WHO_AFRO Lower middle income
```

```
# Create country list
country_list <- country_wb_who[, 2]
country_list <- paste(country_list, collapse = "|")
country_list <- str_replace_all(country_list, "\\\\s+", " ")
country_list <- str_replace_all(country_list, "gambie", "gambia")
```

3.2 Extract countries from the text

This code extract country words above based on detection techniques (including flowcharts and image captions). LoIs with countries not detected were manually inputted and a random sample of LoIs were manually validated. Since several countries can be mentioned in the LoIs, including in the literature review, only countries requencies above the average number of mentions per LoIs were selected (detecting countries that were a focus on the LoIs).

```
# Extract country from LoIs
loi_country <- loi_raw
loi_country$country <- str_extract_all(loi_raw$text, country_list)
loi_country$diseases <- NULL

# Create long dataset with one country per row
loi_country <- cSplit(loi_country, "country", ",", "long")
loi_country$country <- str_replace_all(loi_country$country, "c\\(",
    "")
loi_country$country <- str_replace_all(loi_country$country, "\\)",
    "")
loi_country$country <- str_replace_all(loi_country$country, "\"",
    "")

# Collapse long dataset into LoIs
loi_count_2 <- loi_country
loi_count_2$tally <- 1
loi_count_2 <- aggregate(tally ~ doc_id + serial_number + batch_number +
```

```
    country, loi_count_2, sum)

# Add variables for each country with counts
loi_column_2 <- cast(loi_count_2, doc_id + serial_number + batch_number ~
    country)
names(loi_column_2)[names(loi_column_2) == "character(0"] <- "unidentified"

# Delete mentions of other countries and create dummy
# variables
loi_column_2$average_country <- rowMeans(loi_column_2[, 4:50],
    na.rm = T)
bool_country <- ifelse(loi_column_2[, 4:50] >= loi_column_2$average_country,
    1, NA)
bool_country <- loi_column_2[, 4:50] * bool_country[, 1:47]
loi_column_2[, 4:50] <- bool_country
```

3.3 Classify countries in WHO regions and World Bank income groups

The countries extracted were classified into WHO regions and World Bank income groups.

```
# Merge country with WHO regions and WB income group in long
# dataset
loi_count_2$country <- str_replace_all(loi_count_2$country, "character\\(0",
    "unknown")
loi_count_3 <- merge(loi_count_2, country_wb_who, all.x = TRUE,
    by = "country")
loi_count_3$tally <- 1

## Collapse long dataset into LoIs - WHO_Region
loi_count_3_1 <- aggregate(tally ~ doc_id + serial_number + batch_number +
    who_region, loi_count_3, sum, na.action = na.pass)

## Collapse long dataset into LoIs - WB Income Group
loi_count_3_2 <- aggregate(tally ~ doc_id + serial_number + batch_number +
    wb_income_group, loi_count_3, sum, na.action = na.pass)

## Add variables for each WHO_Region with counts
loi_column_3 <- cast(loi_count_3_1, doc_id + serial_number +
    batch_number ~ who_region, na.action = na.pass)

## Add variables for each WB Income Group with counts
loi_column_4 <- cast(loi_count_3_2, doc_id + serial_number +
    batch_number ~ wb_income_group, na.action = na.pass)
```

3.4 Visualise countries, WHO regions and World Bank income groups in LoIs

Each LoIs was classified into one or more countries and the percentage of LoIs that focused on that country calculated. The code below produces a bar chart of percentages for each country, WHO region and World Bank income group.

```
viz_country <- as.data.frame(prop.table(table(loi_count_2$country)) *
    100)
viz_country$Var1 <- str_replace_all(viz_country$Var1, "character\\(0",
```

```
    "unidentified")

names(viz_country)[names(viz_country) == "Freq"] <- "Percentage"
names(viz_country)[names(viz_country) == "Var1"] <- "Group"

viz_country <- subset(viz_country, viz_country$Group != "unidentified")

g2 <- ggplot(viz_country, aes(x = reorder(Group, Percentage),
    y = Percentage)) + xlab("Country") + ylab("Percentage") +
    geom_bar(stat = "identity", fill = "turquoise4") + theme_minimal() +
    expand_limits(y = c(0, 15), x = c(0, 0)) + coord_flip() +
    theme(plot.title = element_blank(), axis.title.x = element_blank(),
        axis.title.y = element_blank())

print(g2)
```

```
viz_region <- as.data.frame(prop.table(table(loi_count_3$who_region)) *
    100)

viz_region$Var1 <- str_replace_all(viz_region$Var1, "character\\(0",
    "unidentified")

names(viz_region)[names(viz_region) == "Freq"] <- "Percentage"
names(viz_region)[names(viz_region) == "Var1"] <- "Group"

viz_region <- subset(viz_region, viz_region$Group != "NA (WHO Region)")

g3 <- ggplot(viz_region, aes(x = reorder(Group, Percentage),
    y = Percentage)) + xlab("WHO Region") + ylab("Percentage") +
    geom_bar(stat = "identity", fill = "tan2") + theme_minimal() +
    coord_flip() + theme(plot.title = element_blank(), axis.title.x = element_blank(),
    axis.title.y = element_blank())

print(g3)
```

```
viz_income <- as.data.frame(prop.table(table(loi_count_3$wb_income_group)) *
    100)

viz_income$Var1 <- str_replace_all(viz_income$Var1, "character\\(0",
    "unidentified")

names(viz_income)[names(viz_income) == "Freq"] <- "Percentage"
names(viz_income)[names(viz_income) == "Var1"] <- "Group"
viz_income <- subset(viz_income, viz_income$Group != "NA (Income Group)")

g4 <- ggplot(viz_income, aes(x = reorder(Group, Percentage),
    y = Percentage)) + xlab("WB Income Group") + ylab("Percentage") +
    geom_bar(stat = "identity", fill = "darkseagreen") + theme_minimal() +
    coord_flip() + theme(plot.title = element_blank(), axis.title.x = element_blank(),
    axis.title.y = element_blank())

print(g4)
```

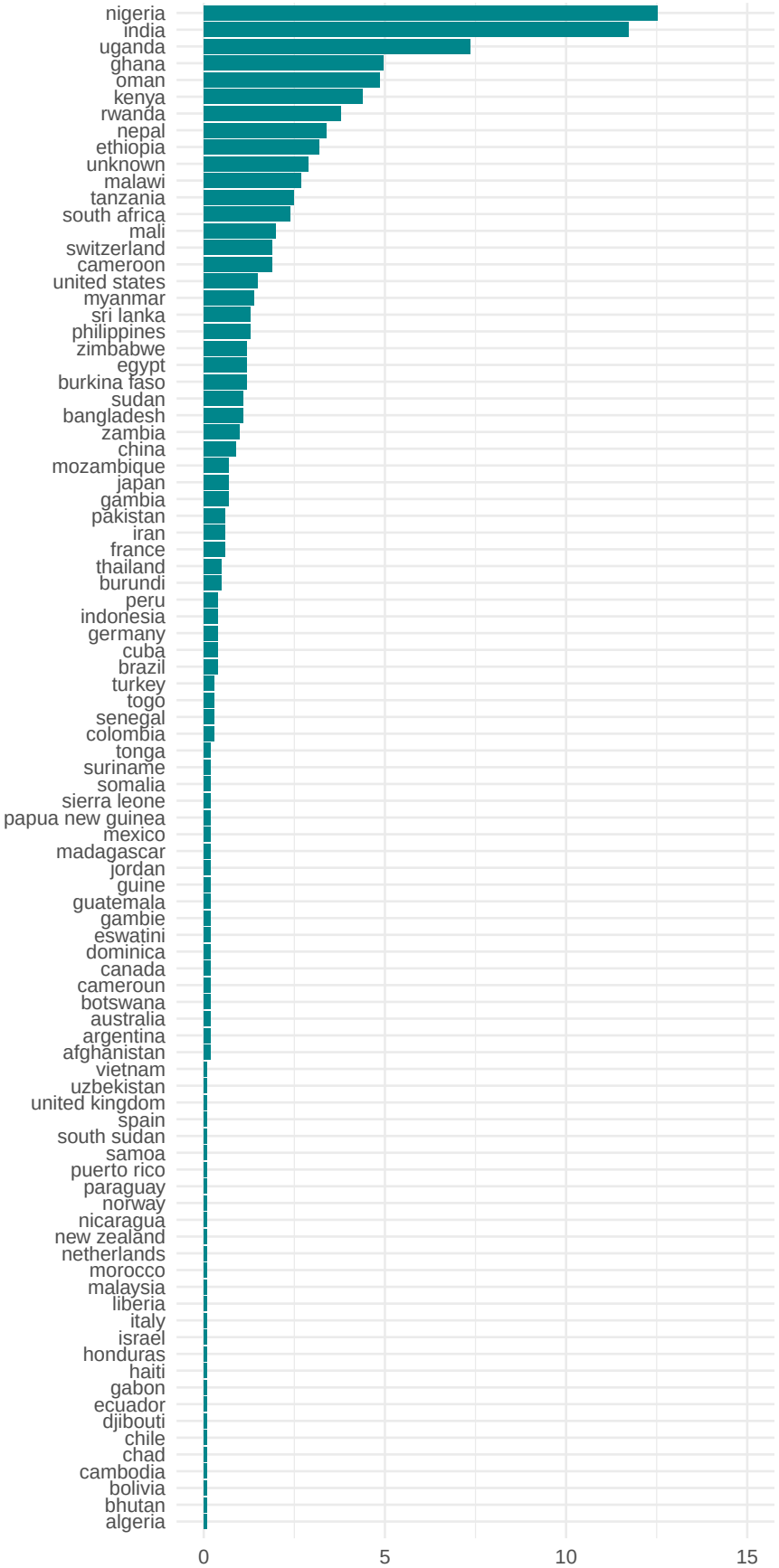

Figure 2: Percentage of LoIs focusing on the country

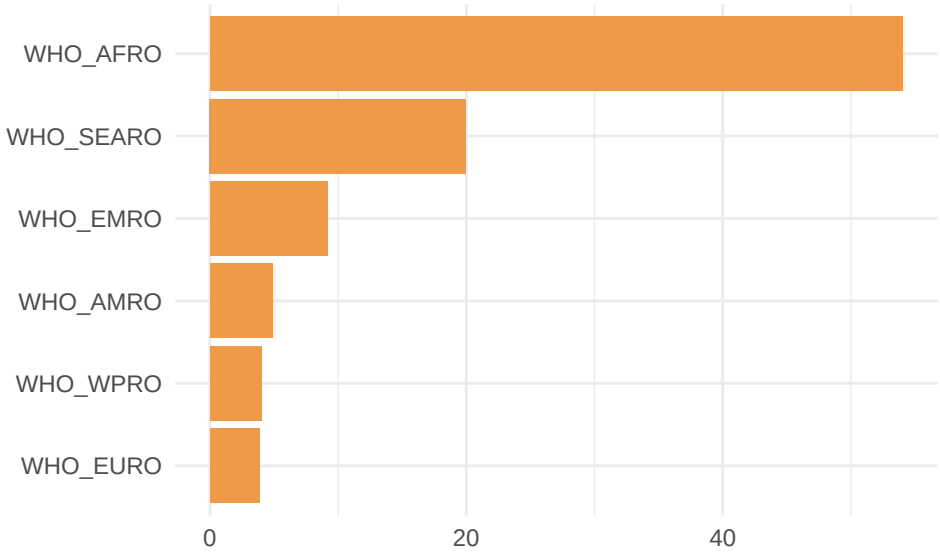

Figure 3: Percentage of LoIs focusing on the WHO region

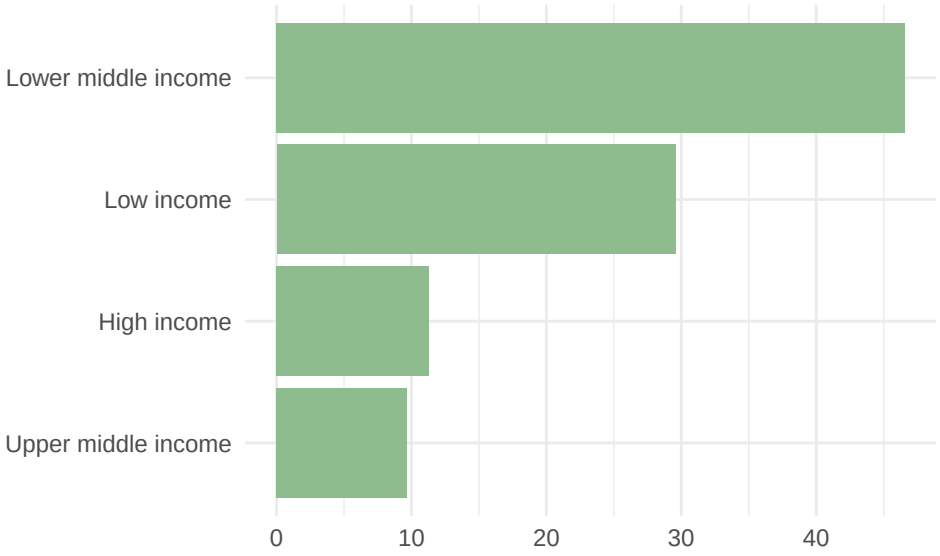

Figure 4: Percentage of LoIs focusing on the World Bank income group

3 Text Mining Research Methods

3.1 Compile a list of research methods

A glossary of qualitative and quantitative methods based on the WHO TDR's Toolkit and the MOOC transcript was developed to serve as a guide to extract terms and expressions.

```
# Upload lexicon for Research Methods
method <- read.csv("Research Methods.csv")

head(method)
```

```
##       General_Terms Research_Designs         Data_sources
## 1       Methodology      Descriptive      Routine program
## 2            Design    Observational Routine intervention
## 3   Research Design      Exploratory Routine service data
## 4 Research Strategy  Cross-sectional      Routine records
## 5 Research Approach      Comparative       Treatment logs
## 6           Methods       Analytical    Treatment records
##          Qualitative_Methods            Quantitative_Methods
## 1     Qualitative interviews                          Survey
## 2      Individual interviews            Survey questionnaire
## 3 Semi-structured interviews               Structured survey
## 4        In-depth interviews             Face-to-face survey
## 5          Expert interviews                    Phone survey
## 6   Key informant interviews Online self-administered survey
```

3.2 Extract research methods from text

3.2.1 Extract general research terms from text

Extract general terms for research methods from the LoIs using detection techniques.

```
# General Terms

# Creating a list for - General Terms
general_terms <- tolower(method[, 1])
general_terms <- as.data.frame(general_terms)
general_terms <- general_terms[rowSums(general_terms == "") !=
    ncol(general_terms), ]
general_terms <- paste(general_terms)
general_terms <- str_replace_all(general_terms, " ", "\\\\s+")

# Extracting General Terms from LoIs
loi_method <- loi_raw
loi_method$general_terms <- str_extract_all(loi_raw$text, general_terms)
loi_method$diseases <- NULL

# Create long dataset with one 'General Term' per row
loi_method <- cSplit(loi_method, "general_terms", ",", "long")
loi_method$general_terms <- str_replace_all(loi_method$general_terms,
    "c\\(", "")
loi_method$general_terms <- str_replace_all(loi_method$general_terms,
    "\\)", "")
loi_method$general_terms <- str_replace_all(loi_method$general_terms,
```

```
    "\"", "")

# Collapse long dataset into LoIs
loi_count_method_1 <- loi_method
loi_count_method_1$tally <- 1
loi_count_method_1 <- aggregate(tally ~ doc_id + serial_number +
    batch_number + general_terms, loi_count_method_1, sum)

# Add variables for each 'General Terms' with counts
loi_column_method_1 <- cast(loi_count_method_1, doc_id + serial_number +
    batch_number ~ general_terms)
names(loi_column_method_1)[names(loi_column_method_1) == "character(0"] <- "unidentified"
```

3.2.2 Extract research design terms from text

Extract research design terms from the LoIs using detection techniques.

```
# Research Designs

# Creating a list for - Research Designs
research_designs <- tolower(method[, 2])
research_designs <- as.data.frame(research_designs)
research_designs <- research_designs[rowSums(research_designs ==
    "") != ncol(research_designs), ]
research_designs <- paste(research_designs)
research_designs <- str_replace_all(research_designs, " ", "\\\\s+")

# Extracting Research Designs from LoIs
loi_method_2 <- loi_raw
loi_method_2$research_designs <- str_extract_all(loi_raw$text,
    research_designs)
loi_method_2$diseases <- NULL

# Create long dataset with one 'Research Design' per row
loi_method_2 <- cSplit(loi_method_2, "research_designs", ",",
    "long")
loi_method_2$research_designs <- str_replace_all(loi_method_2$research_designs,
    "c\\(", "")
loi_method_2$research_designs <- str_replace_all(loi_method_2$research_designs,
    "\\)", "")
loi_method_2$research_designs <- str_replace_all(loi_method_2$research_designs,
    "\"", "")

# Collapse long dataset into LoIs
loi_count_method_2 <- loi_method_2
loi_count_method_2$tally <- 1
loi_count_method_2 <- aggregate(tally ~ doc_id + serial_number +
    batch_number + research_designs, loi_count_method_2, sum)


# Add variables for each 'Research Design' with counts
loi_column_method_2 <- cast(loi_count_method_2, doc_id + serial_number +
    batch_number ~ research_designs)
names(loi_column_method_2)[names(loi_column_method_2) == "character(0"] <- "unidentified"
```

3.2.3 Extract data sources terms from text

Extract data sources terms from the LoIs using detection techniques.

```
# Data Sources

# Creating a list for - Data Sources
data_sources <- tolower(method[, 3])
data_sources <- as.data.frame(data_sources)
# data_sources <- data_sources[1:26,]
data_sources <- data_sources[rowSums(data_sources == "") != ncol(data_sources),
    ]
data_sources <- paste(data_sources, collapse = "|")
data_sources <- str_replace_all(data_sources, "\\\\s+", " ")

# Extracting Data Sources from LoIs
loi_method_3 <- loi_raw
loi_method_3$data_sources <- str_extract_all(loi_raw$text, data_sources)
loi_method_3$diseases <- NULL

# Create long dataset with one 'Data Source' per row
loi_method_3 <- cSplit(loi_method_3, "data_sources", ",", "long")
loi_method_3$data_sources <- str_replace_all(loi_method_3$data_sources,
    "c\\(", "")
loi_method_3$data_sources <- str_replace_all(loi_method_3$data_sources,
    "\\)", "")
loi_method_3$data_sources <- str_replace_all(loi_method_3$data_sources,
    "\"", "")

# Collapse long dataset into LoIs
loi_count_method_3 <- loi_method_3
loi_count_method_3$tally <- 1
loi_count_method_3 <- aggregate(tally ~ doc_id + serial_number +
    batch_number + data_sources, loi_count_method_3, sum)

# Add variables for each 'Data Source' with counts
loi_column_method_3 <- cast(loi_count_method_3, doc_id + serial_number +
    batch_number ~ data_sources)
names(loi_column_method_3)[names(loi_column_method_3) == "character(0"] <- "unidentified"
```

3.2.4 Extract qualitative methods from text

Extract qualitative methods terms from the LoIs using detection techniques.

```
# Qualitative Methods

# Creating a list for - Qualitative Methods
qual_methods <- tolower(method[, 4])
qual_methods <- as.data.frame(qual_methods)
qual_methods <- qual_methods[1:25, ]
# qual_methods <-
# qual_methods[rowSums(qual_methods=='')!=ncol(qual_methods),
```

```
# ]
qual_methods <- paste(qual_methods, collapse = "|")
qual_methods <- str_replace_all(qual_methods, "\\\\s+", " ")

# Extracting Qualitative Methods from LoIs
loi_method_4 <- loi_raw
loi_method_4$qual_methods <- str_extract_all(loi_raw$text, qual_methods)
loi_method_4$diseases <- NULL

# Create long dataset with one 'Qualitative Methods' per row
loi_method_4 <- cSplit(loi_method_4, "qual_methods", ",", "long")
loi_method_4$qual_methods <- str_replace_all(loi_method_4$qual_methods,
    "c\\(", "")
loi_method_4$qual_methods <- str_replace_all(loi_method_4$qual_methods,
    "\\)", "")
loi_method_4$qual_methods <- str_replace_all(loi_method_4$qual_methods,
    "\"", "")

# Collapse long dataset into LoIs
loi_count_method_4 <- loi_method_4
loi_count_method_4$tally <- 1
loi_count_method_4 <- aggregate(tally ~ doc_id + serial_number +
    batch_number + qual_methods, loi_count_method_4, sum)

# Add variables for each 'Data Source' with counts
loi_column_method_4 <- cast(loi_count_method_4, doc_id + serial_number +
    batch_number ~ qual_methods)
names(loi_column_method_4)[names(loi_column_method_4) == "character(0"] <- "unidentified"

# Replace FGD
loi_column_method_4[, 9] <- rowSums(loi_column_method_4[, c("focus group discussion",
    "fgd")], na.rm = TRUE)
loi_column_method_4[, 8] <- NULL
```

3.2.5 Extract quantitive methods terms from text

Extract quantitative methods from the LoIs using detection techniques.

```
# Quantitative Methods

# Creating a list for - Quantitative Methods
quan_methods <- tolower(method[, 5])
quan_methods <- as.data.frame(quan_methods)
quan_methods <- quan_methods[rowSums(quan_methods == "") != ncol(quan_methods),
    ]
quan_methods <- paste(quan_methods, collapse = "|")
quan_methods <- str_replace_all(quan_methods, "\\\\s+", " ")

# Extracting Quantitative Methods from LoIs
loi_method_5 <- loi_raw
loi_method_5$quan_methods <- str_extract_all(loi_raw$text, quan_methods)
loi_method_5$diseases <- NULL

# Create long dataset with one 'Quantitative Methods' per row
```

```
loi_method_5 <- cSplit(loi_method_5, "quan_methods", ",", "long")
loi_method_5$quan_methods <- str_replace_all(loi_method_5$quan_methods,
    "c\\(", "")
loi_method_5$quan_methods <- str_replace_all(loi_method_5$quan_methods,
    "\\)", "")
loi_method_5$quan_methods <- str_replace_all(loi_method_5$quan_methods,
    "\"", "")

# Collapse long dataset into LoIs
loi_count_method_5 <- loi_method_5
loi_count_method_5$tally <- 1
loi_count_method_5 <- aggregate(tally ~ doc_id + serial_number +
    batch_number + quan_methods, loi_count_method_5, sum)

# Add variables for each 'Quantitative Method' with counts
loi_column_method_5 <- cast(loi_count_method_5, doc_id + serial_number +
    batch_number ~ quan_methods)
names(loi_column_method_5)[names(loi_column_method_5) == "character(0"] <- "unidentified"

# Add Research Methods extraction variables into the full
# dataset
loi_all_methods <- merge(merge(merge(loi_column_method_1, loi_column_method_2),
    merge(loi_column_method_3, loi_column_method_4)), loi_column_method_5)
```

3.3 Visualise research methods in LoIs

Each LoIs was classified into one or more research method. The code below produces a bar chart of percentages for each terms and expression of research methods, grouped into general terms, research designs, data sources, quantitative and qualitative methods.

```
# Research Methods - Create a dataset for visualization
viz_general_terms <- as.data.frame(prop.table(table(loi_count_method_1$general_terms)) *
    100)

viz_general_terms$Var1 <- str_replace_all(viz_general_terms$Var1,
    "character\\(0", "unidentified")

names(viz_general_terms)[names(viz_general_terms) == "Freq"] <- "Percentage"
names(viz_general_terms)[names(viz_general_terms) == "Var1"] <- "Research_Method"

viz_general_terms <- subset(viz_general_terms, viz_general_terms$Research_Method !=
    "unidentified")

g6 <- ggplot(viz_general_terms, aes(x = reorder(Research_Method,
    Percentage), y = Percentage)) + xlab("General Terms") + ylab("Percentage") +
    geom_bar(stat = "identity", fill = "indianred2") + theme_minimal() +
    coord_flip() + theme(plot.title = element_blank(), axis.title.x = element_blank(),
    axis.title.y = element_blank())


print(g6)
```

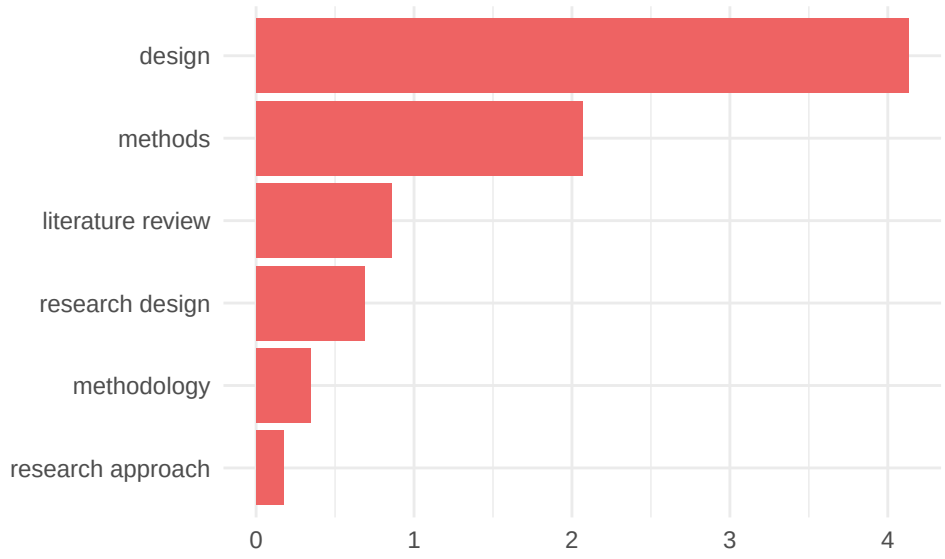

Figure 5: Percentage of LoIs mentioning method

```
viz_research_designs <- as.data.frame(prop.table(table(loi_count_method_2$research_designs)) *
    100)

viz_research_designs$Var1 <- str_replace_all(viz_research_designs$Var1,
    "character\\(0", "unidentified")

names(viz_research_designs)[names(viz_research_designs) == "Freq"] <- "Percentage"
names(viz_research_designs)[names(viz_research_designs) == "Var1"] <- "Research_Method"

viz_research_designs <- subset(viz_research_designs, viz_research_designs$Research_Method !=
    "unidentified")

g7 <- ggplot(viz_research_designs, aes(x = reorder(Research_Method,
    Percentage), y = Percentage)) + xlab("Research Design") +
    ylab("Percentage") + geom_bar(stat = "identity", fill = "turquoise4") +
    theme_minimal() + coord_flip() + theme(plot.title = element_blank(),
    axis.title.x = element_blank(), axis.title.y = element_blank())

print(g7)
```

```
viz_data_sources <- as.data.frame(prop.table(table(loi_count_method_3$data_sources)) *
    100)

viz_data_sources$Var1 <- str_replace_all(viz_data_sources$Var1,
    "character\\(0", "unidentified")

names(viz_data_sources)[names(viz_data_sources) == "Freq"] <- "Percentage"
names(viz_data_sources)[names(viz_data_sources) == "Var1"] <- "Research_Method"

viz_data_sources <- subset(viz_data_sources, viz_data_sources$Research_Method !=
    "unidentified")
```

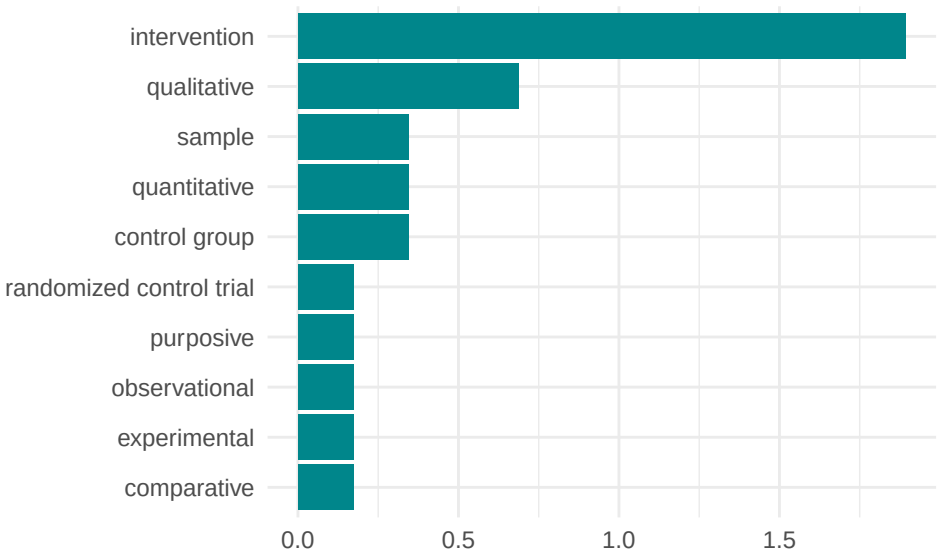

Figure 6: Percentage of LoIs focusing on the research design

```
g8 <- ggplot(viz_data_sources, aes(x = reorder(Research_Method,
    Percentage), y = Percentage)) + xlab("Data Sources") + ylab("Percentage") +
    geom_bar(stat = "identity", fill = "tan2") + theme_minimal() +
    coord_flip() + theme(plot.title = element_blank(), axis.title.x = element_blank(),
    axis.title.y = element_blank())

print(g8)
```

```
viz_qual_methods <- as.data.frame(prop.table(table(loi_count_method_4$qual_methods)) *
    100)

viz_qual_methods$Var1 <- str_replace_all(viz_qual_methods$Var1,
    "character\\(0", "unidentified")

names(viz_qual_methods)[names(viz_qual_methods) == "Freq"] <- "Percentage"
names(viz_qual_methods)[names(viz_qual_methods) == "Var1"] <- "Research_Method"

viz_qual_methods <- subset(viz_qual_methods, viz_qual_methods$Research_Method !=
    "unidentified")

g9 <- ggplot(viz_qual_methods, aes(x = reorder(Research_Method,
    Percentage), y = Percentage)) + xlab("Qualitative Methods") +
    ylab("Percentage") + geom_bar(stat = "identity", fill = "darkseagreen") +
    theme_minimal() + coord_flip() + theme(plot.title = element_blank(),
    axis.title.x = element_blank(), axis.title.y = element_blank())

print(g9)
```

```
viz_quan_methods <- as.data.frame(prop.table(table(loi_count_method_5$quan_methods)) *
    100)

viz_quan_methods$Var1 <- str_replace_all(viz_quan_methods$Var1,
    "character\\(0", "unidentified")
```

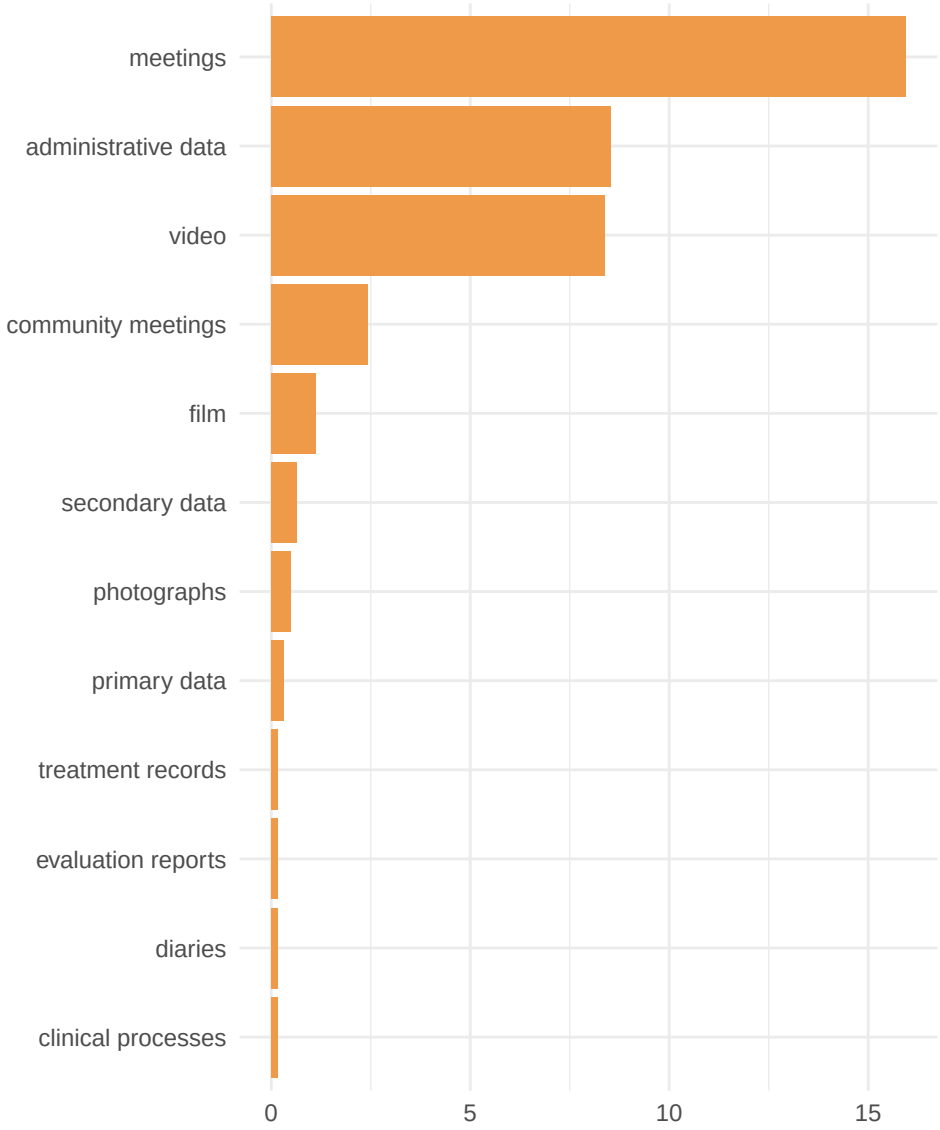

Figure 7: Percentage of LoIs focusing on the data sources

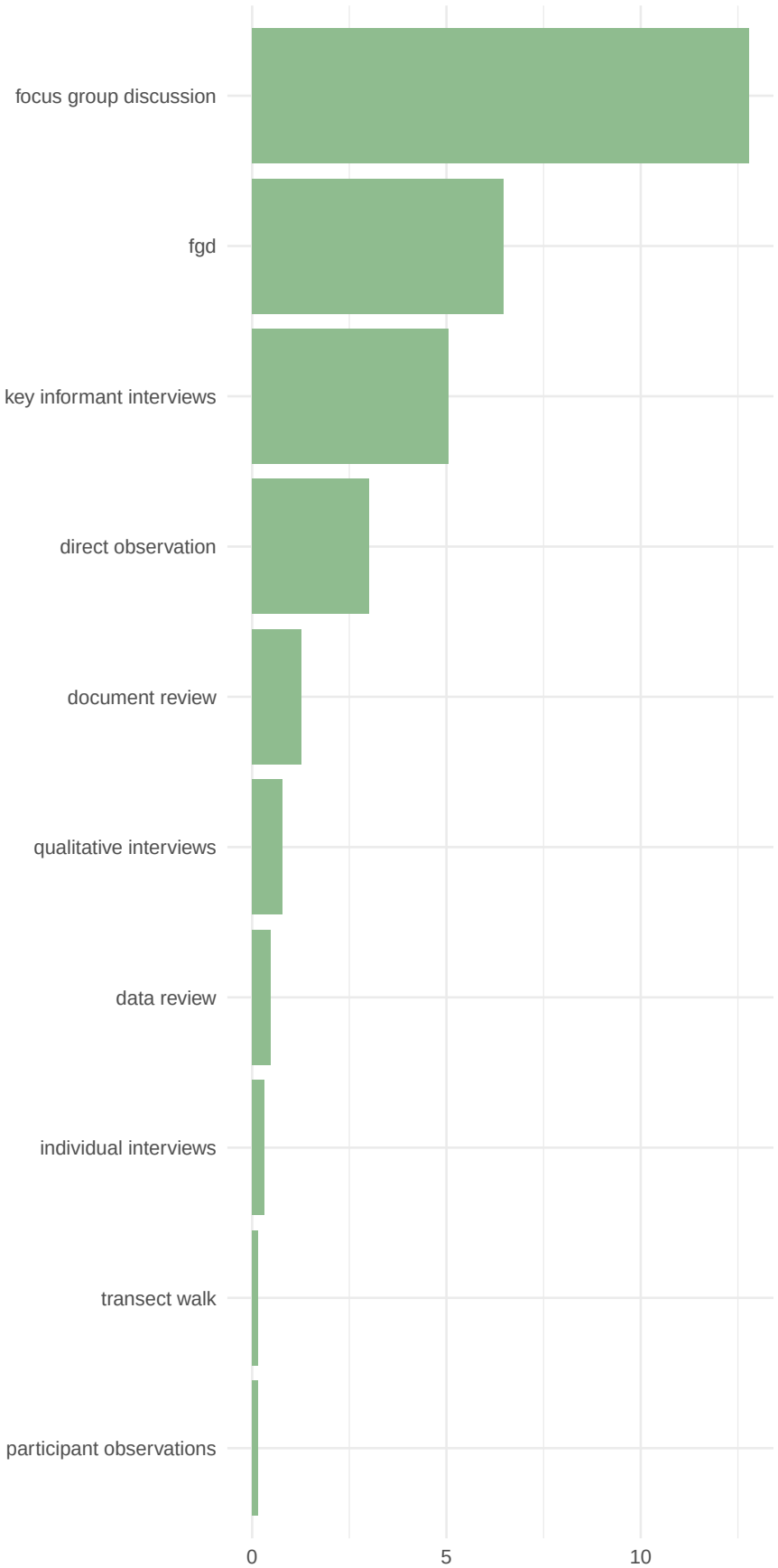

Figure 8: Percentage of LoIs focusing on the qualitative method

```r
names(viz_quan_methods)[names(viz_quan_methods) == "Freq"] <- "Percentage"
names(viz_quan_methods)[names(viz_quan_methods) == "Var1"] <- "Research_Method"

viz_quan_methods <- subset(viz_quan_methods, viz_quan_methods$Research_Method !=
    "unidentified")

g10 <- ggplot(viz_quan_methods, aes(x = reorder(Research_Method,
    Percentage), y = Percentage)) + xlab("Quantitative Methods") +
    ylab("Percentage") + geom_bar(stat = "identity", fill = "steelblue") +
    theme_minimal() + coord_flip() + theme(plot.title = element_blank(),
    axis.title.x = element_blank(), axis.title.y = element_blank())

print(g10)
```

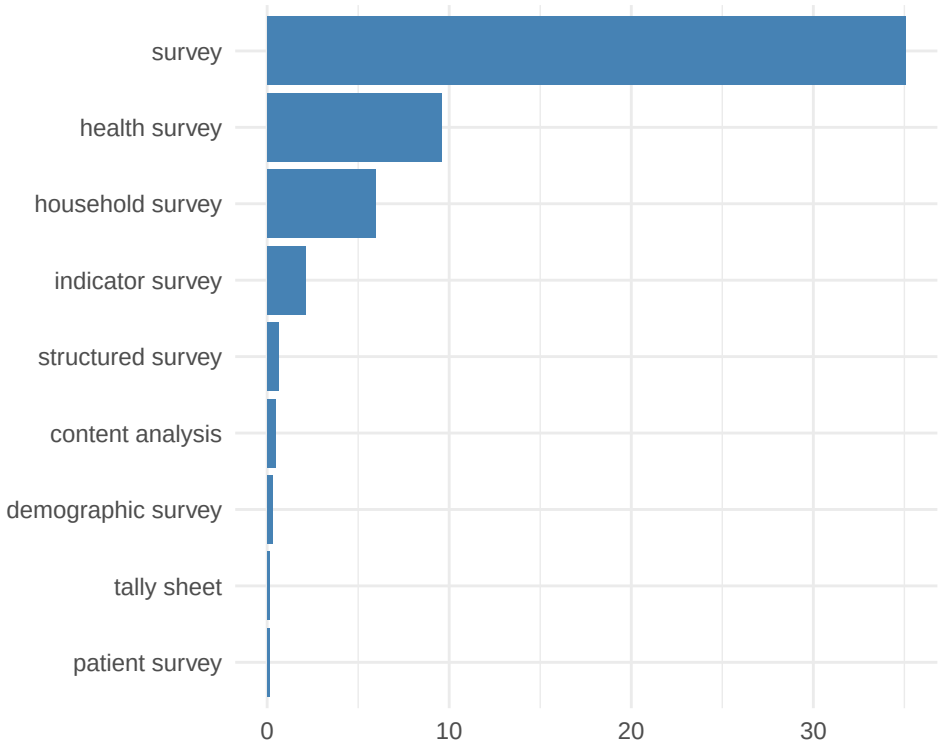

Figure 9: Percentage of LoIs focusing on the quantitative method

4 Text Mining Implementation Research Strategies

4.1 Compile list of IR Strategies

Identify a glossary of 73 IR strategies from the literature Powell et al (2015) and manually scanned the transcript of the MOOC to select IR strategies. The 73 IR strategies identified in the literature were matched with those identified in the MOOC transcript. This was done by two independent researchers that agree on 76% of the classification of IR strategies. The remaining 24% of strategies were discussed with a third researcher to reach a consensus.

```r
# Upload lexicon for IR Strategies
ir_strategies <- read.csv("IR Strategies_Global.csv")
```

```
head(ir_strategies[-2, ])
```

```
##   Page                  MOOC_Strategy  Literature_Strategy
## 1   66                Advocacy speech Inform local leaders
## 3   NA   Community clinic partnership    Build a coalition
## 4   52    Community designed strategy Community engagement
## 5   NA    Community directed strategy    Tailor strategies
## 6   NA         Community directedness    Tailor strategies
## 7   51 Community medicine distributor Involve local actors
```

```
# Creating a list for IR Strategies
ir_strategies_mooc <- tolower(ir_strategies[, 2])
ir_strategies_mooc <- as.data.frame(ir_strategies_mooc)
ir_strategies_mooc <- ir_strategies_mooc[rowSums(ir_strategies_mooc ==
    "") != ncol(ir_strategies_mooc), ]
ir_strategies_mooc <- paste(ir_strategies_mooc, collapse = "|")
ir_strategies_mooc <- str_replace_all(ir_strategies_mooc, "\\\\s+",
    " ")
```

4.2 Extract IR Strategies from text

Extract IR strategies based on terms and expressions identified in the previous step.

```
# Extracting IR strategies from LoIs
loi_strategies <- loi_raw
loi_strategies$mooc_strategy <- str_extract_all(loi_raw$text,
    ir_strategies_mooc)
loi_strategies$diseases <- NULL

# Create long dataset with one 'IR Strategy' per row
loi_strategies <- cSplit(loi_strategies, "mooc_strategy", ",",
    "long")
loi_strategies$mooc_strategy <- str_replace_all(loi_strategies$mooc_strategy,
    "c\\(", "")
loi_strategies$mooc_strategy <- str_replace_all(loi_strategies$mooc_strategy,
    "\\)", "")

# Collapse long dataset into LoIs
loi_count_strategies <- loi_strategies
loi_count_strategies$tally <- 1
loi_count_strategies <- aggregate(tally ~ doc_id + serial_number +
    batch_number + mooc_strategy, loi_count_strategies, sum)

# Add variables for each 'IR Strategy' with counts
loi_column_strategies <- cast(loi_count_strategies, doc_id +
    serial_number + batch_number ~ mooc_strategy)
names(loi_column_strategies)[names(loi_column_strategies) ==
    "character(0"] <- "unidentified"
```

4.3 Visualise IR Strategies in LoIs

Each LoIs was classified into one or more IR strategies. The code below produces a bar chart of percentages for each terms and expression of IR strategies.

```r
# Create dataset for visualisations
viz_strategies <- as.data.frame(prop.table(table(ir_strategies$Literature_Strategy)) *
    100)

viz_strategies$Var1 <- str_replace_all(viz_strategies$Var1, "character\\(0",
    "unidentified")

names(viz_strategies)[names(viz_strategies) == "Freq"] <- "Percentage"
names(viz_strategies)[names(viz_strategies) == "Var1"] <- "Literature_Strategy"

g11 <- ggplot(viz_strategies, aes(x = reorder(Literature_Strategy,
    Percentage), y = Percentage)) + xlab("IR Strategy") + ylab("Percentage") +
    geom_bar(stat = "identity", fill = "darkseagreen") + theme_minimal() +
    coord_flip() + theme(plot.title = element_blank(), axis.title.x = element_blank(),
    axis.title.y = element_blank())

print(g11)
```

5 Text Mining Implementation Research Outcomes

5.1 Compile list of IR Oucomes

List implementation research outcomes.

```r
# Upload lexicon for IR Outcomes

outcomes <- read.csv("Outcomes.csv")
outcomes$other_terms <- tolower(outcomes$other_terms)

as.matrix(unique(outcomes$Outcomes))
```

```
##       [,1]
##  [1,] "Acceptability"
##  [2,] "Adoption"
##  [3,] "Appropriateness"
##  [4,] "Cost"
##  [5,] "Feasibility"
##  [6,] "Fidelity"
##  [7,] "Penetration"
##  [8,] "Sustainability"
##  [9,] "Unidentified Outcome"
```

```r
head(outcomes)
```

```
##          Outcomes   other_terms
## 1   Acceptability  satisfaction
## 2        Adoption        uptake
## 3        Adoption   utilization
## 4        Adoption   utilisation
## 5 Appropriateness     relevance
## 6 Appropriateness compatibility
```

```r
# Creating a list for IR Outcomes
other_terms <- tolower(outcomes[, 2])
```

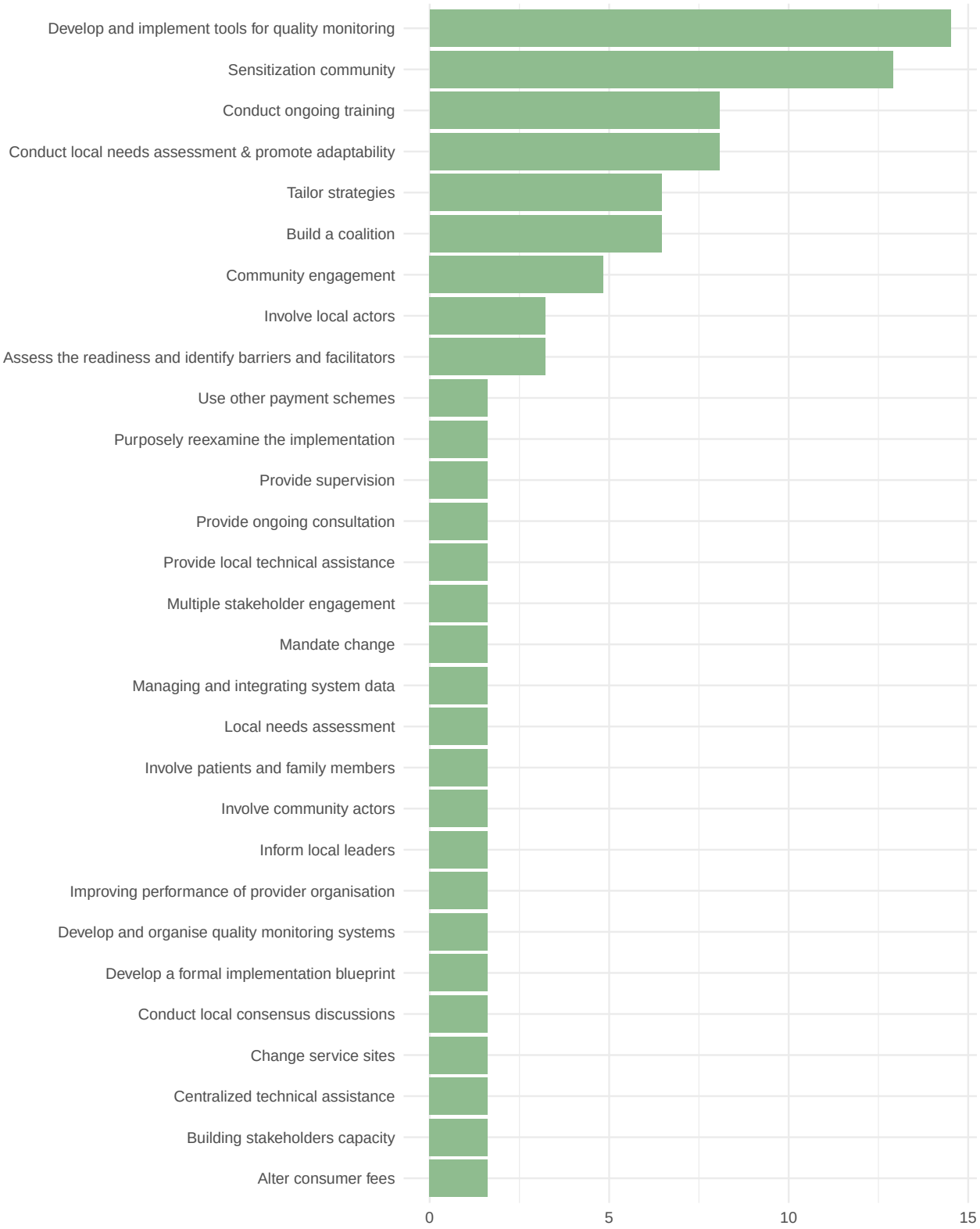

Figure 10: Percentage of LoIs using IR strategy

```
other_terms <- as.data.frame(other_terms)
other_terms <- other_terms[rowSums(other_terms == "") != ncol(other_terms),
    ]
other_terms <- paste(other_terms, collapse = "|")
other_terms <- str_replace_all(other_terms, "\\\\s+", " ")
loi_outcome <- loi_raw
loi_outcome$other_terms <- str_extract_all(loi_raw$text, other_terms)
loi_outcome$diseases <- NULL
```

5.2 Extract IR Outcomes from text

Extract IR outcomes based on expressions identified in the previous step.

```
# Create long dataset with one 'IR Strategy' per row
loi_outcome <- cSplit(loi_outcome, "other_terms", ",", "long")
loi_outcome$other_terms <- str_replace_all(loi_outcome$other_terms,
    "c\\(", "")
loi_outcome$other_terms <- str_replace_all(loi_outcome$other_terms,
    "\\)", "")
loi_outcome$other_terms <- str_replace_all(loi_outcome$other_terms,
    "\"", "")
loi_outcome$other_terms <- str_replace_all(loi_outcome$other_terms,
    "character\\(0", "character\\(0\\)")

# Collapse long dataset into LoIs
loi_outcome <- merge(loi_outcome, outcomes, by = "other_terms")
loi_count_outcome <- loi_outcome
loi_count_outcome$tally <- 1
loi_count_outcome <- aggregate(tally ~ doc_id + serial_number +
    batch_number + Outcomes, loi_count_outcome, sum)

# Add variables for each 'IR Outcome' with counts
loi_column_outcome <- cast(loi_count_outcome, doc_id + serial_number +
    batch_number ~ Outcomes)
```

5.3 Visualise IR Outcomes in LoIs

Each LoIs was classified into one or more IR outcomes The code below produces a bar chart of percentages for each terms and expression of IR outcomes

```
# Create dataset for visualisations
viz_outcomes <- as.data.frame(prop.table(table(loi_count_outcome$Outcomes)) *
    100)
names(viz_outcomes)[names(viz_outcomes) == "Var1"] <- "IR_Outcomes"
names(viz_outcomes)[names(viz_outcomes) == "Freq"] <- "Percentage"

viz_outcomes$IR_Outcomes <- str_replace_all(viz_outcomes$IR_Outcomes,
    "character\\(0", "unidentified")
viz_outcomes <- subset(viz_outcomes, viz_outcomes$IR_Outcomes !=
    "Unidentified Outcome")

g12 <- ggplot(viz_outcomes, aes(x = reorder(IR_Outcomes, Percentage),
    y = Percentage)) + xlab("IR Outcomes") + ylab("Percentage") +
```

```
    geom_bar(stat = "identity", fill = "steelblue") + theme_minimal() +
    coord_flip() + theme(plot.title = element_blank(), axis.title.x = element_blank(),
    axis.title.y = element_blank())

print(g12)
```

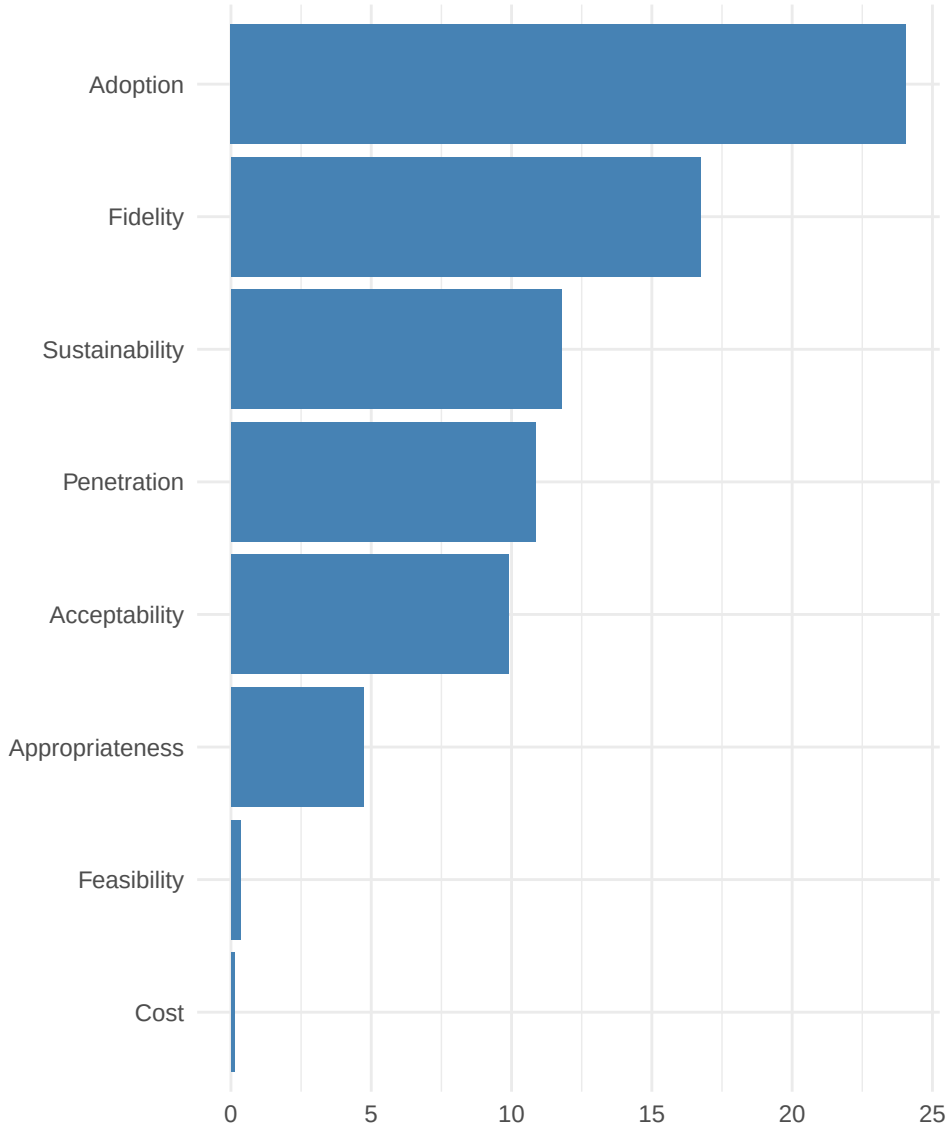

Figure 11: Percentage of LoIs focusing on the IR outcome