Outcomes_Main.Rmd

Jonathan Reardon

```r
library(reticulate)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(reshape2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
use_python("/usr/local/bin/python3")
```

```python
# import necessary libraries
import json
from collections import Counter
from pprint import pprint
from matplotlib import pyplot as plt
import pandas as pd
plt.style.use('ggplot')

# import dataset
with open('/home/jon/json/Batch1.json') as f:
    data=json.load(f)

####################################################
### GET STRAND LABELS AND KEYS FROM TOP OUTER LAYER
####################################################

def get_strand_info():
    '''
    a function that returns
    a dict containing strand labels
    and corresponding attribute ids
    '''
    strands={}
    for counter, element in enumerate(data["CodeSets"][0]["Attributes"]["AttributesList"]):
        attribute_name=(data["CodeSets"][0]["Attributes"]["AttributesList"][counter]["AttributeName"])
        attribute_id=(data["CodeSets"][0]["Attributes"]["AttributesList"][counter]["AttributeId"])
        strands.update( {attribute_id:attribute_name} )
    return strands

######################################
### DISPLAY STRAND SUMMARY INFORMATION
######################################

def get_strand_summary():
    '''
    A function that produces a basic
    summary of strand study counts
    and a graph to display them
    '''
    global counts, strand_title
    strand_overview=[]
    for element in range(len(data["References"])):
        for key, value in strands.items():
            for section in range(len(data["References"][element]["Codes"])):
                if key == data["References"][element]["Codes"][section]["AttributeId"]:
                    a=(data["References"][element]["ItemId"])
                    b=(data["References"][element]["Title"])
                    strand_overview.append([value, key, a, b])

    strand_title=[]
    for element in strand_overview:
        strand_title.append(element[0])

    counts = Counter(strand_title)
    pprint(counts)

#######################################
### GET THE ID FOR OUR STRAND OF CHOICE
#######################################

def get_strand_value(strand_label):
    '''
    A function that takes in a
    strand name and returns
    the strand ID
    '''
    for key, value in strands.items():
        if value == strand_label:
            return key, value

######################################################
###  GET EFFECT SIZE INFO FROM STRAND SPECIFIC STUDIES
######################################################
```

```python
def get_data(strand_key, strand_value, outcome_choice):
    '''
    A function that accepts a strand id and a variable of
    interest and returns a list of that id and the variable
    values.
    '''
    outcome_studies=[]

    # iterate over each section of 'references'
    for section in range(len(data["References"])):
        # iterate over each study within each section of 'references'
        for study in range(len(data["References"][section]["Codes"])):
            # check each study to see if strand id is present
            if strand_id[0] == data["References"][section]["Codes"][study]["AttributeId"]:
                if "Outcomes" in data["References"][section]:
                    if data["References"][section]["Outcomes"][0]["OutcomeText"] == outcome_choice:
                        outcome_id=((data["References"][section]["Outcomes"][0]["OutcomeId"]))
                        outcome_type=(data["References"][section]["Outcomes"][0]["ShortTitle"])
                        outcome_text=(data["References"][section]["Outcomes"][0]["OutcomeText"])
                        SMD=(data["References"][section]["Outcomes"][0]["SMD"])
                        SESMD=(data["References"][section]["Outcomes"][0]["SESMD"])
                        year=(data["References"][section]["Year"])
                        intervention=(data["References"][section]["Outcomes"][0]["InterventionText"])
                        CIupperSMD=(data["References"][section]["Outcomes"][0]["CIUpperSMD"])
                        CIlowerSMD=(data["References"][section]["Outcomes"][0]["CILowerSMD"])
                        outcome_studies.append([strand_key, strand_value, outcome_id, outcome_text, outc

    # display number of studies found within selected strand
    print('Number of studies within strand {}: {} with Primary Outcome'.format(strand_value, len(outcom

    pd.set_option('display.max_rows', 15)
    pd.set_option('display.max_columns', 15)

    # convert data list to pandas dataframe for viewing
    df_primary = pd.DataFrame(outcome_studies, columns=['AttributeId', 'Strand', 'OutcomeId', 'OutcomeTy

    # round SMD and SESMS and CIupperSMD and CIlowerSMD to two decimal points
    df_primary.loc[:, "SMD"] = df_primary["SMD"].astype(float).round(2)
    df_primary.loc[:, "SESMD"] = df_primary["SESMD"].astype(float).round(2)
    df_primary.loc[:, "CIupperSMD"] = df_primary["CIupperSMD"].astype(float).round(2)
    df_primary.loc[:, "CIlowerSMD"] = df_primary["CIlowerSMD"].astype(float).round(2)

    return df_primary

strands = get_strand_info()
get_strand_summary()
```

```
## Counter({'Oral language interventions': 138,
##          'Feedback': 114,
##          'Peer tutoring': 109,
##          'Teaching assistants': 62,
##          'Small group tuition': 30,
##          'One to one tuition': 10,
##          'Phonics': 6,
##          'Digital technology': 4,
##          'Metacognition and self-regulation': 4,
##          'Parental engagement': 1,
##          'Extending school time': 1,
##          'Reducing class size': 1})
```

```
strand_id = get_strand_value("Oral language interventions")
oral_lang = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Oral language interventions: 89 with Primary Outcome
```

```
strand_id = get_strand_value("Feedback")
feedback = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Feedback: 89 with Primary Outcome
```

```
strand_id = get_strand_value("Peer tutoring")
peer_tut = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Peer tutoring: 94 with Primary Outcome
```

```
strand_id = get_strand_value("Teaching assistants")
teaching_assist = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Teaching assistants: 42 with Primary Outcome
```

```
strand_id = get_strand_value("Small group tuition")
small_group = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Small group tuition: 26 with Primary Outcome
```

```
strand_id = get_strand_value("One to one tuition")
one_to_one = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand One to one tuition: 9 with Primary Outcome
```

```
strand_id = get_strand_value("Phonics")
phonics = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Phonics: 6 with Primary Outcome
```

```
strand_id = get_strand_value("Digital technology")
digital_tech = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Digital technology: 3 with Primary Outcome
```

```
strand_id = get_strand_value("Metacognition and self-regulation")
metacog_self_reg = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Metacognition and self-regulation: 3 with Primary Outcome
```

```
strand_id = get_strand_value("Parental engagement")
parental_engage = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Parental engagement: 1 with Primary Outcome
```

```
strand_id = get_strand_value("Extending school time")
extending_school = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Extending school time: 1 with Primary Outcome
```

```
strand_id = get_strand_value("Reducing class size")
reduce_class = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Reducing class size: 1 with Primary Outcome
```

```
all_strands=pd.concat([oral_lang, feedback, peer_tut, teaching_assist, small_group,
                       one_to_one, phonics, digital_tech, metacog_self_reg,
                       parental_engage, extending_school, reduce_class]).drop_duplicates().reset_index(d
```

```
# convert pandas dataframe to R data frame
all_strands_df <- data.frame(py$all_strands)

all_strands_df$Intervention <- as.character(all_strands_df$Intervention)
all_strands_df$Intervention[all_strands_df$Intervention==""] <- NA
all_strands_df$Intervention <- as.factor(all_strands_df$Intervention)

mean_SMD <- mean(all_strands_df$SMD, na.rm=TRUE)
mean_SESMD <- mean(all_strands_df$SESMD, na.rm=TRUE)

mean_SMD
```

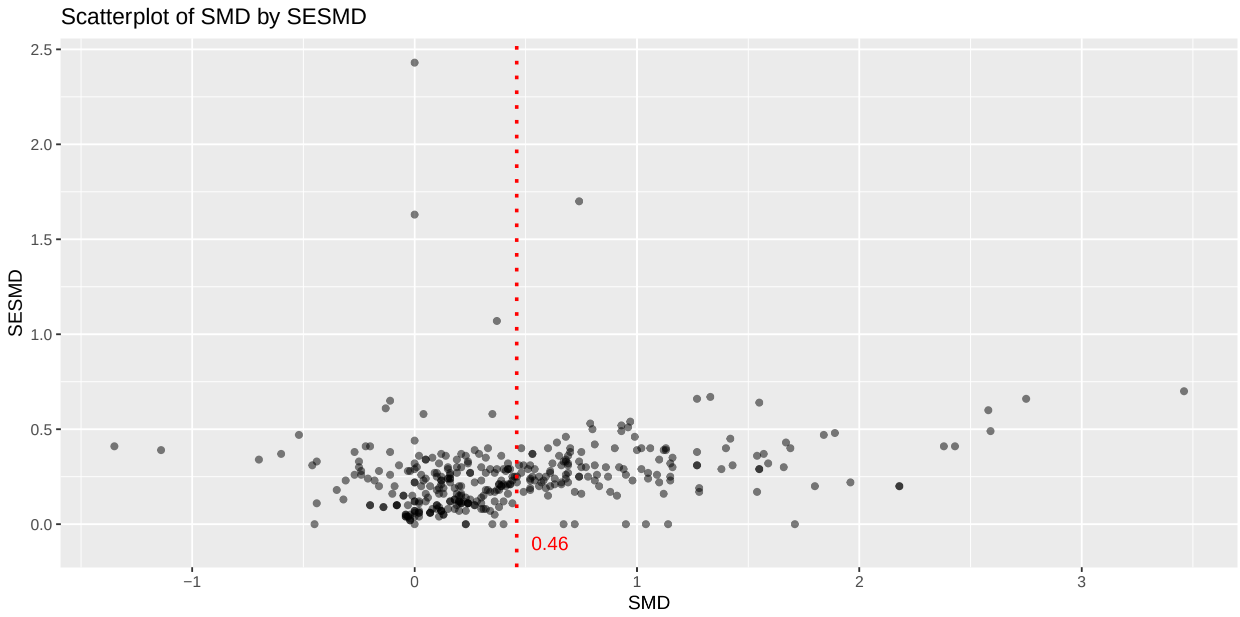
```
## [1] 0.4588462
```

```
mean_SESMD
```

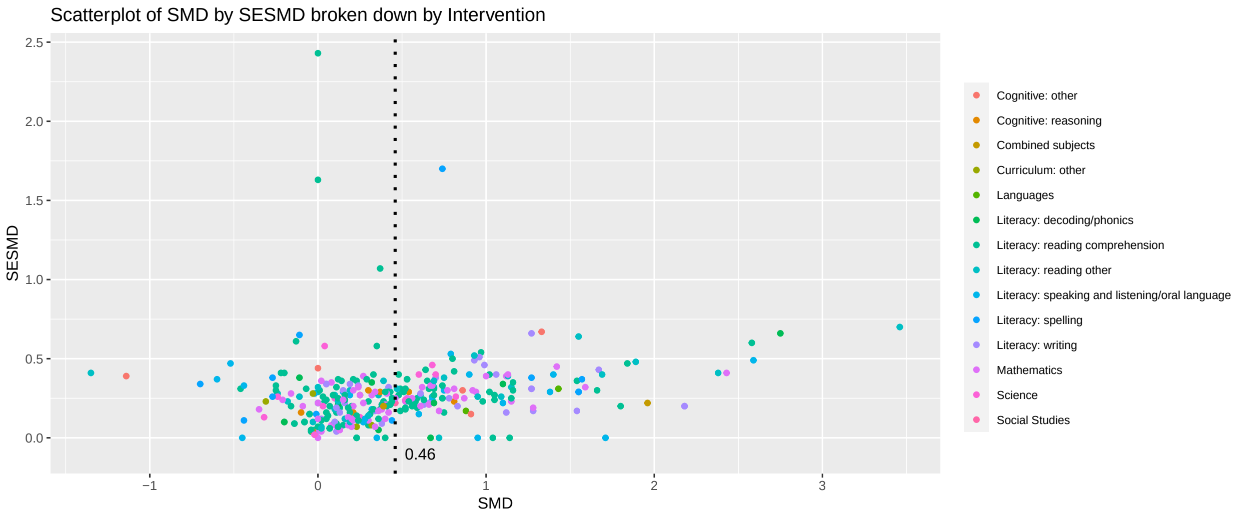
```
## [1] 0.2553994
```

```
# view dataframe
View(all_strands_df)
```

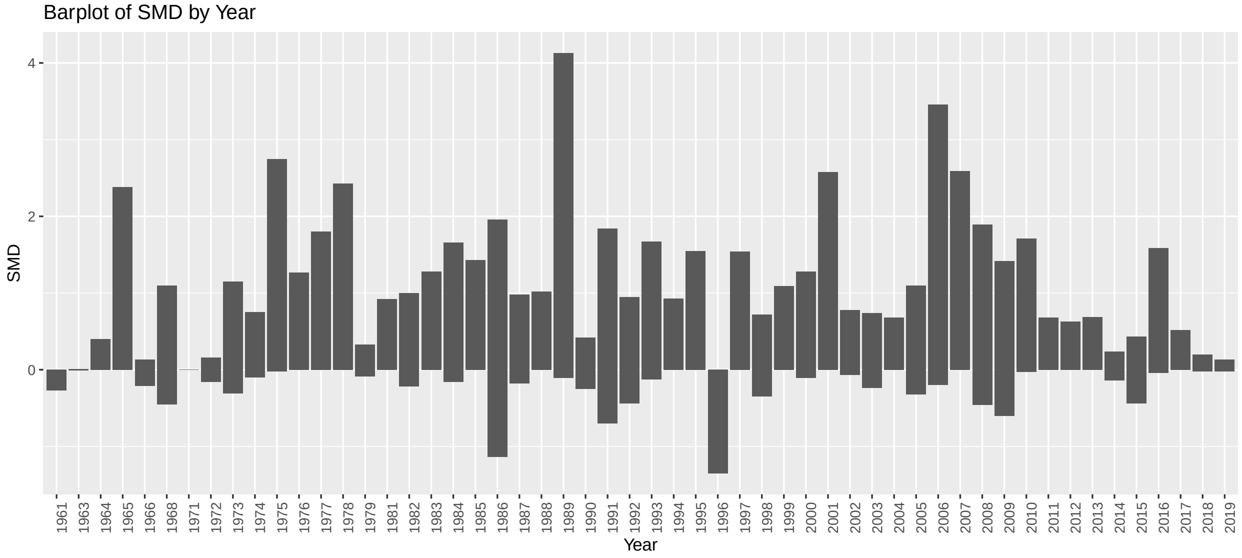
```
ggplot(data=subset(all_strands_df, !is.na(Intervention)), aes(SMD, SESMD)) + geom_point(alpha=.5, na.rm=
    theme_grey() +
    geom_vline(xintercept=mean_SMD, linetype="dotted", color="red", size=1) +
    theme(legend.title = element_text(color = "blue", size = 5),
          legend.text = element_text(color = "red", size = 5)) +
    annotate(geom="text", x=mean_SMD+.15, y=-.1, label=round(mean_SMD, 2), color="red") +
    ggtitle("Scatterplot of SMD by SESMD")
```



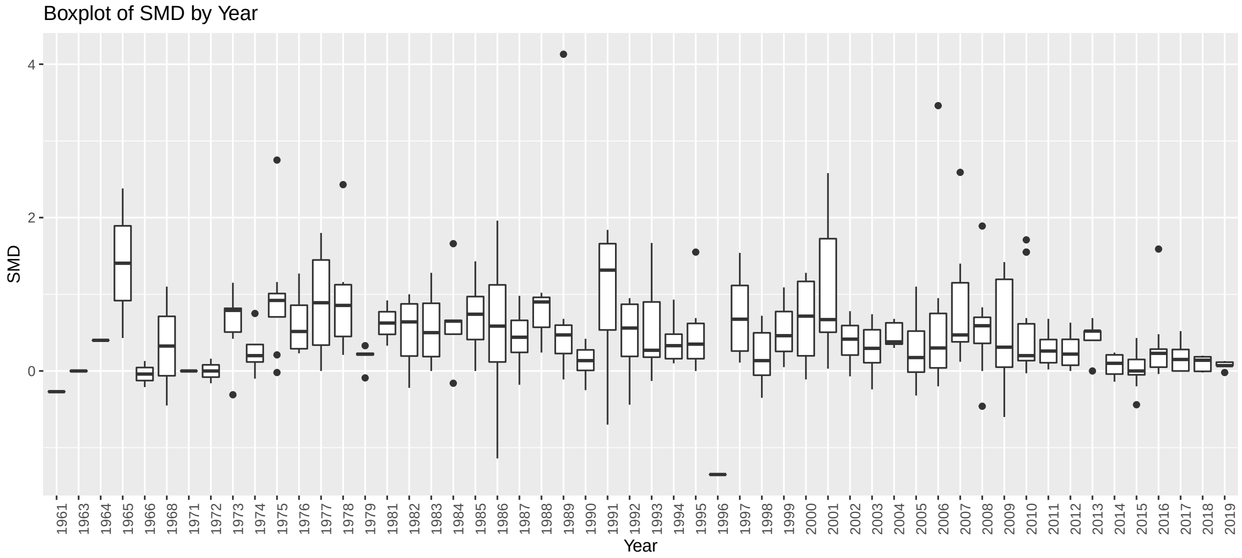
```
ggplot(data=subset(all_strands_df, !is.na(Intervention)), aes(SMD, SESMD, color=Intervention)) + geom_p
    theme_grey() +
    geom_vline(xintercept=mean_SMD, linetype="dotted", color="black", size=1) +
    theme(legend.title = element_text(color = "black", size = 10),
          legend.text = element_text(color = "black", size = 8)) +
    theme(legend.position="right") +
    guides(fill=guide_legend(nrow=5, byrow=TRUE)) +
    theme(legend.title=element_blank()) +
    annotate(geom="text", x=mean_SMD+.15, y=-.1, label=round(mean_SMD, 2), color="black") +
    ggtitle("Scatterplot of SMD by SESMD broken down by Intervention")
```



```
ggplot(data=all_strands_df, aes(Year, SMD)) +
    geom_bar(position="dodge", stat="identity", na.rm=TRUE) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    ggtitle("Barplot of SMD by Year")
```



```
ggplot(data=all_strands_df, aes(Year, SMD)) +
    geom_boxplot() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    ggtitle("Boxplot of SMD by Year")
```



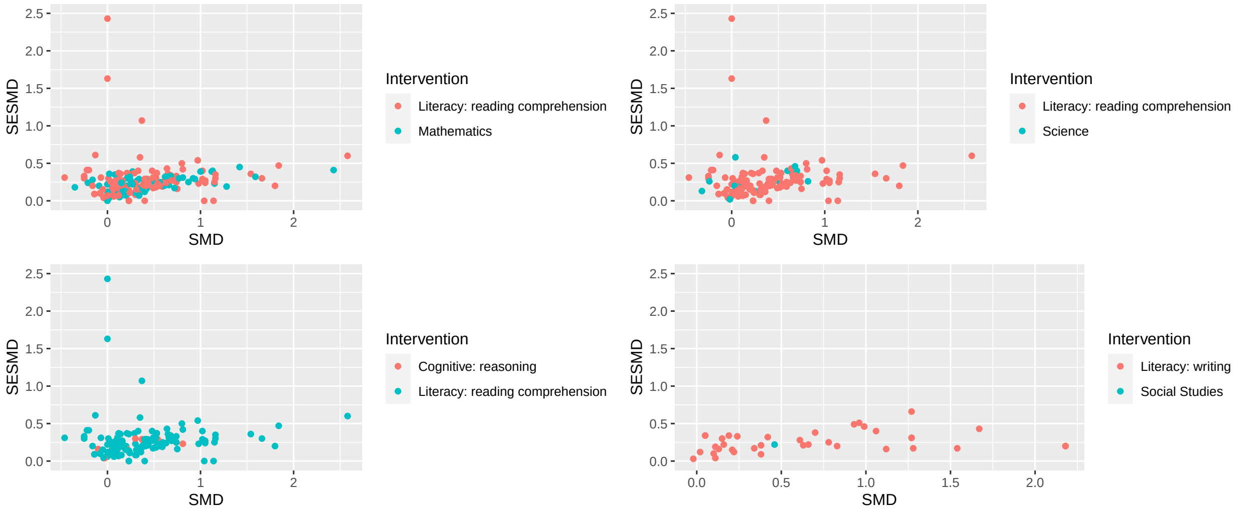
```
all_strands_df %>%
  filter(Intervention=="Mathematics" | Intervention=="Literacy: reading comprehension") %>%
  ggplot(., aes(SMD, SESMD, color=Intervention)) + geom_point(alpha=1, na.rm=TRUE) +
  ylim(0, 2.5) -> p1

all_strands_df %>%
  filter(Intervention=="Science" | Intervention=="Literacy: reading comprehension") %>%
  ggplot(., aes(SMD, SESMD, color=Intervention)) + geom_point(alpha=1, na.rm=TRUE) +
  ylim(0, 2.5) -> p2

all_strands_df %>%
  filter(Intervention=="Cognitive: reasoning" | Intervention=="Literacy: reading comprehension") %>%
  ggplot(., aes(SMD, SESMD, color=Intervention)) + geom_point(alpha=1, na.rm=TRUE) +
  ylim(0, 2.5) -> p3

all_strands_df %>%
  filter(Intervention=="Literacy: writing" | Intervention=="Social Studies") %>%
  ggplot(., aes(SMD, SESMD, color=Intervention)) + geom_point(alpha=1, na.rm=TRUE) +
  ylim(0, 2.5) -> p4

grid.arrange(p1, p2, p3, p4, ncol=2)
```



```
strand_id = get_strand_value("Feedback")
feedback = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Feedback: 89 with Primary Outcome
```

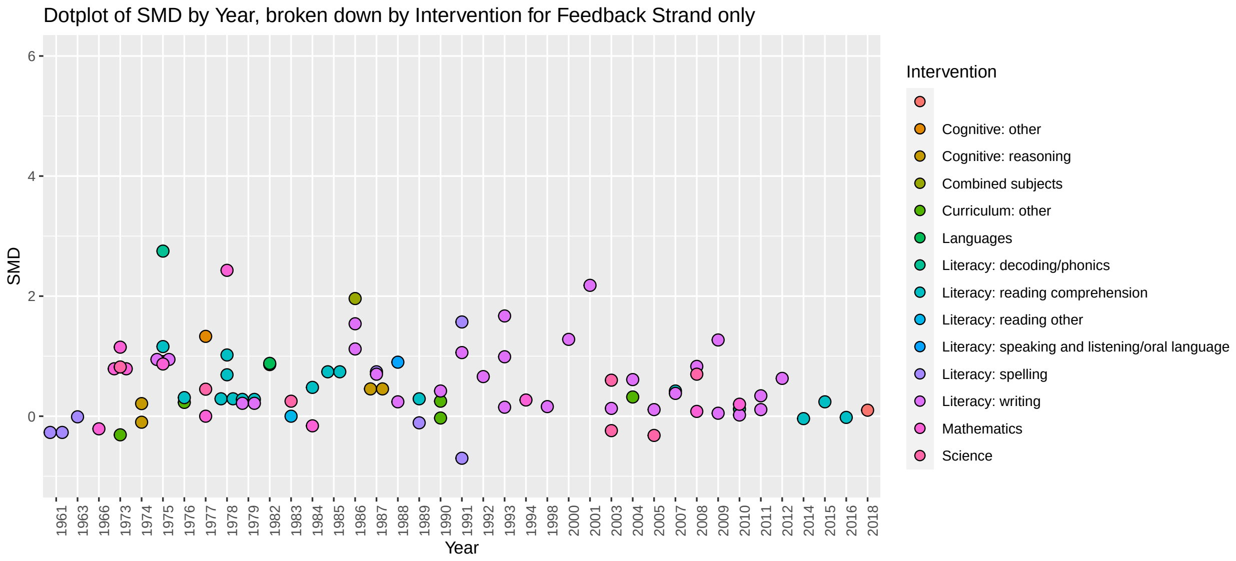
```
strand_id = get_strand_value("Oral language interventions")
oral_lang = get_data(strand_id[0], strand_id[1], "Primary outcome")
```

```
## Number of studies within strand Oral language interventions: 89 with Primary Outcome
```

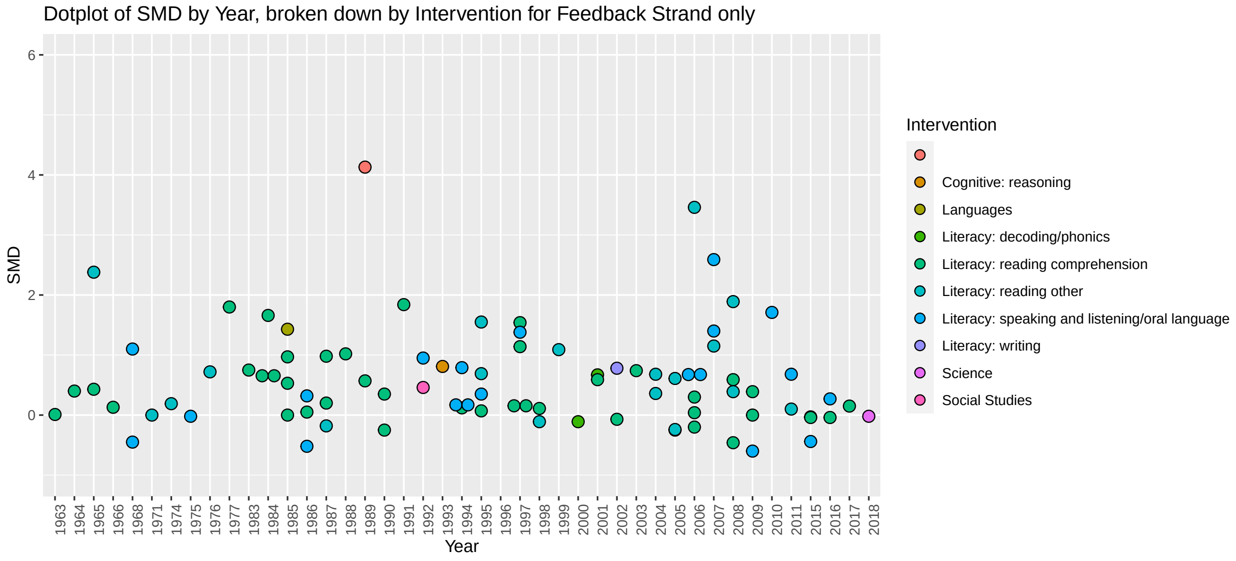
```
feedback_df <- data.frame(py$feedback)
oral_lang <- data.frame(py$oral_lang)

View(feedback_df)
View(oral_lang)

filter(feedback_df, !is.na(Intervention)) %>%
  ggplot(aes(fill=Intervention, y=SMD, x=Year)) +
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1, binwidth=.2, na.rm=TRUE) +
  ggtitle("Dotplot of SMD by Year, broken down by Intervention for Feedback Strand only") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylim(-1,6)
```



```r
filter(oral_lang, !is.na(Intervention)) %>%
  ggplot(aes(fill=Intervention, y=SMD, x=Year)) +
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1, binwidth=.2, na.rm=TRUE) +
  ggtitle("Dotplot of SMD by Year, broken down by Intervention for Feedback Strand only") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ylim(-1,6)
```


Dotplot of SMD by Year, broken down by Intervention for Feedback Strand only
SMD
Year
Intervention
Cognitive: reasoning
Languages
Literacy: decoding/phonics
Literacy: reading comprehension
Literacy: reading other
Literacy: speaking and listening/oral language
Literacy: writing
Science
Social Studies