Milestone 3: Clean dataset with descriptive statistics for relevant data elements

Team 20

#1. Subset rows & columns

```
#air quality
air_qual<- read_csv("air_qual.csv",
                    col_names = TRUE, col_types = NULL, na = c("", "NA"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   `California County` = col_character(),
##   ZIP = col_double(),
##   `Avg PM2.5 Per ZIP` = col_double()
## )
```

```
ca_air_qual <- rename(air_qual, county = "California County",
                      zip = "ZIP",
                      avg_PM2.5_per_zip = "Avg PM2.5 Per ZIP")
ca_air_qual[,3] <- round(ca_air_qual$avg_PM2.5_per_zip, 3)
head(ca_air_qual)
```

```
## # A tibble: 6 x 3
##   county    zip avg_PM2.5_per_zip
##   <chr>   <dbl>             <dbl>
## 1 Alameda 94501              8.70
## 2 Alameda 94502              8.70
## 3 Alameda 94536              8.94
## 4 Alameda 94538              9.44
## 5 Alameda 94539              9.46
## 6 Alameda 94541              8.70
```

```
dim(ca_air_qual)
```

```
## [1] 1377    3
```

```
#mort data
mort_by_zip <- read_csv("mort_by_zip.csv",na = c("", "NA"))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   Year = col_double(),
##   ZIP_Code = col_double(),
##   Geography_Type = col_character(),
```

```
##   Strata = col_character(),
##   Strata_Name = col_character(),
##   Cause = col_character(),
##   Cause_Desc = col_character(),
##   Count = col_double(),
##   Annotation_Code = col_double(),
##   Annotation_Desc = col_character()
## )
```

```
mort_by_zip <- select(mort_by_zip, "Year", "ZIP_Code", "Cause", "Cause_Desc", "Count")
mort_by_year <- filter(mort_by_zip, between(Year, 2009, 2018) )
mort_by_cause <- filter(mort_by_year, Cause == "CLD")
head(mort_by_cause)
```

```
## # A tibble: 6 x 5
##    Year ZIP_Code Cause Cause_Desc                          Count
##   <dbl>    <dbl> <chr> <chr>                               <dbl>
## 1  2009    90001 CLD   Chronic lower respiratory diseases    NA
## 2  2009    90002 CLD   Chronic lower respiratory diseases    NA
## 3  2009    90003 CLD   Chronic lower respiratory diseases    18
## 4  2009    90004 CLD   Chronic lower respiratory diseases    NA
## 5  2009    90005 CLD   Chronic lower respiratory diseases    NA
## 6  2009    90006 CLD   Chronic lower respiratory diseases    NA
```

```
dim(mort_by_cause)
```

```
## [1] 26640     5
```

#2. Creating 2+ new variables (and subsetting by pollution level)

Identify and create a "critical" category of PM2.5 measurements (the highest quantile), and subset those zipcode observations.

```
#grouping all zipcodes
air_qual_grouped_zips <- ca_air_qual %>%
  group_by(zip,county) %>%
  summarise(avg_PM_per_zip = mean(avg_PM2.5_per_zip))
```

```
## `summarise()` has grouped output by 'zip'. You can override using the `.groups` argument.
```

```
quantile(air_qual_grouped_zips$avg_PM_per_zip, c(.10,.25,.5,.75,.90),
         na.rm = TRUE)
```

```
##     10%     25%     50%     75%     90%
##  6.0448  7.8600  9.5360 11.8400 12.8900
```

```
#75th percentile is 11.84
#90th percentile is 12.89

#creating new binary column to indicate whether or not the zip has a critical
#value of PM2.5 pollution

air_qual_grouped_zips <- air_qual_grouped_zips %>%
  mutate(critical_level_pollution =  case_when(avg_PM_per_zip >= 11.84~1))

air_qual_grouped_zips$critical_level_pollution[is.na(air_qual_grouped_zips$critical_level_pollution)] <-

#creating a subset of zip codes that have PM2.5 above critical value
air_qual_by_75_percent <- air_qual_grouped_zips %>%
  filter(critical_level_pollution == 1)
```

```
#remove duplicates of zip code
duplicated(air_qual_by_75_percent$zip)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
length(duplicated(air_qual_by_75_percent$zip))
```

```
## [1] 344
```

```
length(unique(air_qual_by_75_percent$zip))
```

```
## [1] 342
```

```
air_qual_by_75_percent<-air_qual_by_75_percent[!duplicated(air_qual_by_75_percent$zip), ]
```

Creating a *total count* of CLDRM for the 2009-2018 period.

```
mort_grouped_zip <- mort_by_cause %>%
  group_by(ZIP_Code, Cause) %>%
  summarise(sum_count = sum(Count))
```

```
## `summarise()` has grouped output by 'ZIP_Code'. You can override using the `.groups` argument.
```

```
head(mort_grouped_zip)
```

```
## # A tibble: 6 x 3
## # Groups:   ZIP_Code [6]
##   ZIP_Code Cause sum_count
##      <dbl> <chr>     <dbl>
## 1    90001 CLD          NA
## 2    90002 CLD          NA
## 3    90003 CLD          NA
## 4    90004 CLD          NA
## 5    90005 CLD          NA
## 6    90006 CLD          NA
```

```
dim(mort_grouped_zip)
```

```
## [1] 2664    3
```

subset of mortality data to match pollution data

```
mort_matched_zips <- mort_grouped_zip %>%
  filter(ZIP_Code %in% air_qual_by_75_percent$zip)

mort_matched_zips
```

```
## # A tibble: 342 x 3
## # Groups:   ZIP_Code [342]
##    ZIP_Code Cause sum_count
##       <dbl> <chr>     <dbl>
##  1    90001 CLD          NA
##  2    90002 CLD          NA
##  3    90003 CLD          NA
##  4    90004 CLD          NA
##  5    90005 CLD          NA
```

```
##  6    90006 CLD          NA
##  7    90007 CLD          NA
##  8    90008 CLD         151
##  9    90010 CLD          NA
## 10    90011 CLD          NA
## # ... with 332 more rows
```

#3. Cleaning data Creating new data set with mortality and pollution columns for all zip codes with a critical level of PM2.5 pollution + died from CLD

```
final_data <- cbind(air_qual_by_75_percent, mort_matched_zips)
final_data <- final_data %>%
  select( county, zip, avg_PM_per_zip, sum_count)

#cleaning
final_data$sum_count[is.na(final_data$sum_count)] <- NA
final_data$avg_PM_per_zip <- round(final_data$avg_PM_per_zip, 2)
final_data$zip <-as.character(final_data$zip)
head(final_data)
```

```
## # A tibble: 6 x 4
## # Groups:   zip [6]
##   county      zip   avg_PM_per_zip sum_count
##   <chr>       <chr>          <dbl>     <dbl>
## 1 Los Angeles 90001           12.1        NA
## 2 Los Angeles 90002           12.0        NA
## 3 Los Angeles 90003           12.1        NA
## 4 Los Angeles 90004           12.9        NA
## 5 Los Angeles 90005           12.9        NA
## 6 Los Angeles 90006           12.9        NA
```

Table 1: Data Dictionary for CA CLD mortality per zip code given PM2.5 air pollution rate

Variable Name	Data Type	Description
zip	Character	California zip codes
avg_PM_per_zip	Numeric	Average PM2.5 per zip code
sum_count	Numeric	Total count of CLDRM for the 2009-2018 period
critical_level_pollution	Numeric	Binary variable to indicate whether or not the zip has a critical value of PM2.5 po
county	Character	California county names

#4. Data dictionary

```
data_dictionary <- data.frame("Variable name" = c("zip", "avg_PM_per_zip",
                                                  "sum_count",
                                                  "critical_level_pollution",
                                                  "county"),
                              "Data type" = c("Character", "Numeric", "Numeric",
                                              "Numeric", "Character"),
                              "Description" = c("California zip codes",
                              "Average PM2.5 per zip code",
                              "Total count of CLDRM for the 2009-2018 period",
                              "Binary variable to indicate whether or not the zip has a critical value
                              "California county names"))

kable(data_dictionary, col.names = c("Variable Name", "Data Type", "Description"),
      caption = "Data Dictionary for CA CLD mortality per
      zip code given PM2.5 air pollution rate")
```

#5. One or more tables with descriptive statistics for 4 data elements

```
#table of average pollution & counts CLDRM

myvars <- c("avg_PM_per_zip", "sum_count")

tab_1 <- CreateTableOne(vars = myvars, data = final_data,)

tab_1
```

```
##
##                               Overall
##   n                             342
##   avg_PM_per_zip (mean (SD))  13.33 (1.77)
##   sum_count (mean (SD))      208.68 (104.33)
```

```
#table of average pollution and deaths per county

avgpm_mort_county<-final_data %>%
  group_by(county) %>%
  summarise(avg_pm25=mean(avg_PM_per_zip), death = sum(sum_count, na.rm = T))
avgpm_mort_county <- avgpm_mort_county %>% arrange(desc(avg_pm25))

kable(avgpm_mort_county)
```

county	avg_pm25	death
Kern	17.57091	1981
Kings	15.86400	295
Tulare	15.55706	1093
Fresno	15.24297	583
Imperial	13.23000	0
San Diego	13.15778	312
Riverside	12.97318	1177
San Joaquin	12.94824	867
Merced	12.91000	213
Stanislaus	12.80882	1066
San Bernardino	12.67483	2220
Madera	12.66600	0
Los Angeles	12.21759	2714
Orange	12.05000	0