SPM_2

Reagan Costello-White

6/16/2021

1) Import and clean data

Load packages, import Excel files generated by ART reports in Service Point (One for each fiscal year)

Load excel file of demographics data for all clients, created with report writer

```
library(tidyverse)
```

Add a column for fiscal year, change variable names to snake_case

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(readxl)
library(rmarkdown)

SPM2_FY15 <- read_excel("data/SPM2_FY15.xls",
                        sheet = "Tab B - First Exit Detail") %>%
  as_tibble("SPM2_FY15.xls") %>%
  clean_names() %>%
  mutate(FY = 15)

SPM2_FY16 <- read_excel("data/SPM2_FY16.xls",
                        sheet = "Tab B - First Exit Detail")%>%
  as_tibble("SPM2_FY16.xls") %>%
  clean_names() %>%
  mutate(FY = 16)

SPM2_FY17 <- read_excel("data/SPM2_FY17.xls",
                        sheet = "Tab B - First Exit Detail")%>%
  as_tibble("SPM2_FY17.xls") %>%
  clean_names()%>%
  mutate(FY = 17)

SPM2_FY18 <- read_excel("data/SPM2_FY18.xls",
                        sheet = "Tab B - First Exit Detail")%>%
  as_tibble("SPM2_FY18.xls") %>%
  clean_names()%>%
  mutate(FY = 18)

SPM2_FY19 <- read_excel("data/SPM2_FY19.xls",
                        sheet = "Tab B - First Exit Detail")%>%
  as_tibble("SPM2_FY19.xls") %>%
  clean_names()%>%
  mutate(FY = 19)

SPM2_FY20 <- read_excel("data/SPM2_FY20.xls",
                        sheet = "Tab B - First Exit Detail")%>%
  as_tibble("SPM2_FY20.xls") %>%
  clean_names()%>%
  mutate(FY = 20)

SPM2_FY21 <- read_excel("data/SPM2_FY21.xls",
                        sheet = "Tab B - First Exit Detail")%>%
  as_tibble("SPM2_FY21.xls") %>%
  clean_names()%>%
  mutate(FY = 21)

demographics_FY13_FYTD21 <- read_csv("data/demographics_FY13_FYTD21.csv") %>%
    as_tibble("SPM2_FY21.xls") %>%
  clean_names()
```

```
##
```

```
## -- Column specification --------------------------------------------------------
## cols(
##   client_id = col_double(),
##   last_name = col_character(),
##   first_name = col_character(),
##   entry_date = col_character(),
##   exit_date = col_character(),
##   client_location_always_choose_va_502_unless_directed_otherwise = col_character(),
##   total_monthly_income = col_double(),
##   current_locality = col_character(),
##   date_of_birth = col_character(),
##   ethnicity = col_character(),
##   primary_race = col_character()
## )
```

2) Merge & Prepare Data Files

Merge data of SPM2 for all fiscal years into one dataframe

Join this data frame with the client demographic information

```
d_all_FY <-  bind_rows(SPM2_FY15, SPM2_FY16, SPM2_FY17, SPM2_FY18, SPM2_FY19, SPM2_FY20, SPM2_FY21)
d_all_raw <- left_join(d_all_FY, demographics_FY13_FYTD21, by = c("client_uid" = "client_id"))
d_all <-  d_all_raw %>%
  group_by(client_uid) %>%
  arrange(client_uid) %>%
  mutate(race_f = factor(primary_race),
         proj_type_f = factor(proj_type),
         provider_f = factor(provider)) %>%
  mutate(race_fc = fct_collapse(race_f,
                                White = "White (HUD)",
                                "Black or African American"= "Black or African American (HUD)",
                                Unknown = c("Client refused (HUD)", "Client doesn't know (HUD)",
                                            "Data not collected (HUD)"),
                                Asian = "Asian (HUD)",
                                "Native Hawaiian or Pacific Islander" = "Native Hawaiian or Other Pacif
                                "American Indian or Alaska Native" =  "American Indian or Alaska Native
  mutate(race_3 = fct_collapse(race_f,
                               White = "WHite (HUD)",
                               "Black or African American" = "Black or African American (HUD)",
                               Unknown = c("Client refused (HUD)", "Client doesn't know (HUD)",
                                           "Data not collected (HUD)"),
                               Other = c( "Native Hawaiian or Other Pacific Islander (HUD)",
                                          "American Indian or Alaska Native (HUD)", "Asian (HUD)"))) %>%
    mutate(race_2 = fct_collapse(race_f,
                               White = "White (HUD)",
                               "Black or African American" = "Black or African American (HUD)",
                               "Other or Unknown" = c("Client refused (HUD)", "Client doesn't know (HUD
                                           "Data not collected (HUD)", "Native Hawaiian or Other Pacific
                                          "American Indian or Alaska Native (HUD)", "Asian (HUD)"))) %>%
```

```
  mutate(race_n = as.numeric(race_fc)) %>%
  mutate(end_date = as.Date(end_date, format = '%Y/%m/%d')) %>%
  mutate(days_to_reappear = replace_na(days_to_reappear, 0)) %>%
  mutate(days_dichot = case_when(is.na(days_to_reappear) ~ "No",
                                 days_to_reappear == 0 ~ "No",
                                 days_to_reappear > 0 ~ "Yes")) %>%
  mutate(start_date = ymd(start_date),
         end_date = ymd(end_date)) %>%
           mutate(year = year(end_date),
                  month = month(end_date)) %>%
           mutate(y_m = paste(year, month, sep = "-")) %>%
           relocate(client_uid, end_date, days_to_reappear, days_dichot,
                    race_fc, race_n) %>%
           slice(1) %>%
  ungroup
```

Change variable types, collapse factors, and filter one row per client

3) Summary Tables

To create barplots with categorical variables, we first need to create a table of summary statistics

```
library(gt)
sum_days_dichot <- d_all %>%
  group_by(days_dichot, race_fc) %>%
  select(client_uid, FY, race_fc, race_f, month, year, days_dichot, days_to_reappear)%>%
  na.omit()%>%
  summarise(n = n_distinct(client_uid),
            min = min(days_to_reappear, na.rm = TRUE),
            max = max(days_to_reappear, na.rm = TRUE),
            mean = mean(days_to_reappear, na.rm = TRUE),
            median = median(days_to_reappear, na.rm = TRUE),
            sd = sd(days_to_reappear, na.rm = TRUE),
            sem = sd/sqrt(n()),
            upper_ci = mean + (1.96 * sem),
            lower_ci = mean - (1.96 * sem))

gt_days_dichot <- gt(sum_days_dichot) %>%
  tab_header(title = "Days to Return to Homelessness") %>%
  fmt_number(columns = 7:10, decimals = 2) %>%
  fmt_number(columns = 6, decimals = 2) %>%
  cols_width(upper_ci ~ px(100),
             days_dichot ~ px(100),
             lower_ci ~ px(100),
             sd ~ px(120),
             n ~ px(50),
             mean ~ px(100),
             median ~ px(100),
```

```
            days_dichot ~ px(200)) %>%
  cols_align(align = "center") %>%
  cols_label(race_fc = "Race",
             mean = "Mean",
             median = "Median",
             min = "Min",
             max = "Max",
             sd = "S. Dev",
             sem = "SEM",
             upper_ci = "Upper",
             lower_ci = "Lower") %>%
  tab_spanner(label = "95% Confidence Intervals",
              columns = c(upper_ci, lower_ci))

gt_days_dichot
```

Days to Return to Homelessness

Race	n	Min	Max	Mean	Median	S. Dev	SEM	95% Confidence Intervals	
								Upper	Lower
No									
Black or African American	912	0	0	0.00	0.00	0.00	0.00	0.00	0.0000
White	938	0	0	0.00	0.00	0.00	0.00	0.00	0.0000
American Indian or Alaska Native	8	0	0	0.00	0.00	0.00	0.00	0.00	0.0000
Asian	9	0	0	0.00	0.00	0.00	0.00	0.00	0.0000
Unknown	9	0	0	0.00	0.00	0.00	0.00	0.00	0.0000
Native Hawaiian or Pacific Islander	3	0	0	0.00	0.00	0.00	0.00	0.00	0.0000
Yes									
Black or African American	285	1	1093	433.71	395.00	285.46	16.91	466.85	400.5668
White	321	1	1066	336.06	277.00	275.35	15.37	366.18	305.9402
American Indian or Alaska Native	9	162	937	549.33	599.00	284.16	94.72	734.99	363.6800
Asian	1	382	382	382.00	382.00	NA	NA	NA	NA

This first Table looks people who returned to homelessness compared to those who did not

```
library(gt)
sum_FY <- d_all %>%
  group_by(FY) %>%
  select(client_uid, FY, race_fc, race_f, month, year, days_dichot, days_to_reappear)%>%
  na.omit()%>%
  summarise(n = n_distinct(client_uid),
            min = min(days_to_reappear, na.rm = TRUE),
            max = max(days_to_reappear, na.rm = TRUE),
            mean = mean(days_to_reappear, na.rm = TRUE),
            median = median(days_to_reappear, na.rm = TRUE),
            sd = sd(days_to_reappear, na.rm = TRUE),
            sem = sd/sqrt(n()),
            upper_ci = mean + (1.96 * sem),
```

```
            lower_ci = mean - (1.96 * sem))

gt_FY <- gt(sum_FY) %>%
  tab_header(title = "Days to Return to Homelessness") %>%
  fmt_number(columns = 7:10, decimals = 2) %>%
  fmt_number(columns = 5, decimals = 2) %>%
  cols_width(upper_ci ~ px(100),
             lower_ci ~ px(100),
             sd ~ px(120),
             n ~ px(50),
             mean ~ px(100),
             median ~ px(100)) %>%
  cols_align(align = "center") %>%
  cols_label(FY = "FY",
             mean = "Mean",
             median = "Median",
             min = "Min",
             max = "Max",
             sd = "S. Dev",
             sem = "SEM",
             upper_ci = "Upper",
             lower_ci = "Lower") %>%
  tab_spanner(label = "95% Confidence Intervals",
              columns = c(upper_ci, lower_ci))


gt_FY
```

Days to Return to Homelessness

								95% Confidence Intervals	
FY	n	Min	Max	Mean	Median	S. Dev	SEM	Upper	Lower
15	72	0	764	49.56	0	160.10	18.87	86.54	12.57
16	328	0	1093	75.38	0	210.91	11.65	98.21	52.56
17	363	0	1066	117.99	0	231.82	12.17	141.84	94.14
18	570	0	1008	100.48	0	228.02	9.55	119.20	81.76
19	674	0	1036	94.74	0	216.68	8.35	111.10	78.38
20	488	0	1041	91.31	0	206.69	9.36	109.65	72.97

We can also look at a table by Fiscal year

```
sum_FY <- d_all %>%
  group_by(FY) %>%
  select(client_uid, FY, race_fc, race_f, month, year, days_dichot, days_to_reappear)%>%
  na.omit()%>%
  summarise(n = n_distinct(client_uid),
            min = min(days_to_reappear, na.rm = TRUE),
            max = max(days_to_reappear, na.rm = TRUE),
            mean = mean(days_to_reappear, na.rm = TRUE),
```

```
            median = median(days_to_reappear, na.rm = TRUE),
            sd = sd(days_to_reappear, na.rm = TRUE),
            sem = sd/sqrt(n()),
            upper_ci = mean + (1.96 * sem),
            lower_ci = mean - (1.96 * sem))

gt_FY <- gt(sum_FY) %>%
  tab_header(title = "Days to Return to Homelessness") %>%
  fmt_number(columns = 7:10, decimals = 2) %>%
  fmt_number(columns = 5, decimals = 2) %>%
  cols_width(upper_ci ~ px(100),
             lower_ci ~ px(100),
             sd ~ px(120),
             n ~ px(50),
             mean ~ px(100),
             median ~ px(100)) %>%
  cols_align(align = "center") %>%
  cols_label(FY = "FY",
             mean = "Mean",
             median = "Median",
             min = "Min",
             max = "Max",
             sd = "S. Dev",
             sem = "SEM",
             upper_ci = "Upper",
             lower_ci = "Lower") %>%
  tab_spanner(label = "95% Confidence Intervals",
              columns = c(upper_ci, lower_ci))


gt_FY
```

Summary by Race

Summary Table By Month

```
d_month <- d_all %>%
  group_by(year, month) %>%
  mutate(mean_y_m = mean(days_to_reappear, na.rm = TRUE),
         median_y_m = median(days_to_reappear, na.rm = TRUE))

sum_month <- d_month %>%
  group_by(year, month) %>%
  select(client_uid, days_to_reappear, FY, race_fc, race_f, month, year, y_m, mean_y_m, median_y_m)%>%
  na.omit()%>%
  summarise(n = n_distinct(client_uid),
            min = min(days_to_reappear, na.rm = TRUE),
            max = max(days_to_reappear, na.rm = TRUE),
            mean = mean(days_to_reappear, na.rm = TRUE),
            median = median(days_to_reappear, na.rm = TRUE),
```

```
            sd = sd(days_to_reappear, na.rm = TRUE),
            sem = sd/sqrt(n()),
            upper_ci = mean + (1.96 * sem),
            lower_ci = mean - (1.96 * sem)) %>%
  arrange(year, month)
```

This creates a table with 69 rows

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

```
gt_month <- gt(sum_month) %>%
  tab_header(title = "Mean and Median Days to Return to Homelessness by Month") %>%
  fmt_number(columns = 7:10, decimals = 2) %>%
  fmt_number(columns = 6, decimals = 2) %>%
  cols_width(upper_ci ~ px(100),
             lower_ci ~ px(100),
             n ~ px(50),
             mean ~ px(100),
             median ~ px(100),
             year ~ px(200))%>%
  cols_align(align = "center") %>%
  cols_label(mean = "Mean",
             median = "Median",
             min = "Min",
             max = "Max",
             sd = "S. Dev",
             sem = "SEM",
             upper_ci = "Upper",
             lower_ci = "Lower",
             year = "Race") %>%
  tab_spanner(label = "95% Confidence Intervals",
              columns = c(upper_ci, lower_ci))

gt_month
```

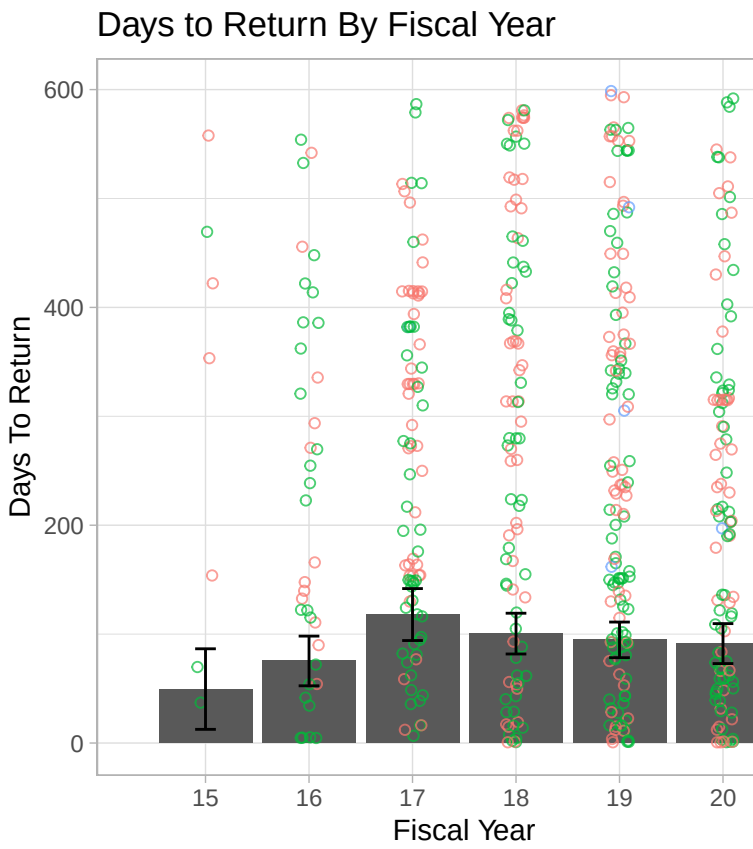
4) Plots

To get a better picture of what is going on, we can look at a graph

```
f_all <- d_all %>%
  filter(between(days_to_reappear, 1, 600))

ggplot(data = sum_FY, aes(FY, mean)) +
  geom_col()+
  scale_x_discrete(limits=c(15,16,17,18,19,20,21))+
  geom_point(data = f_all, aes(FY, days_to_reappear, color = race_2),
             position = position_jitter(width = 0.1),
             shape = 1,
             alpha = .7) +
```

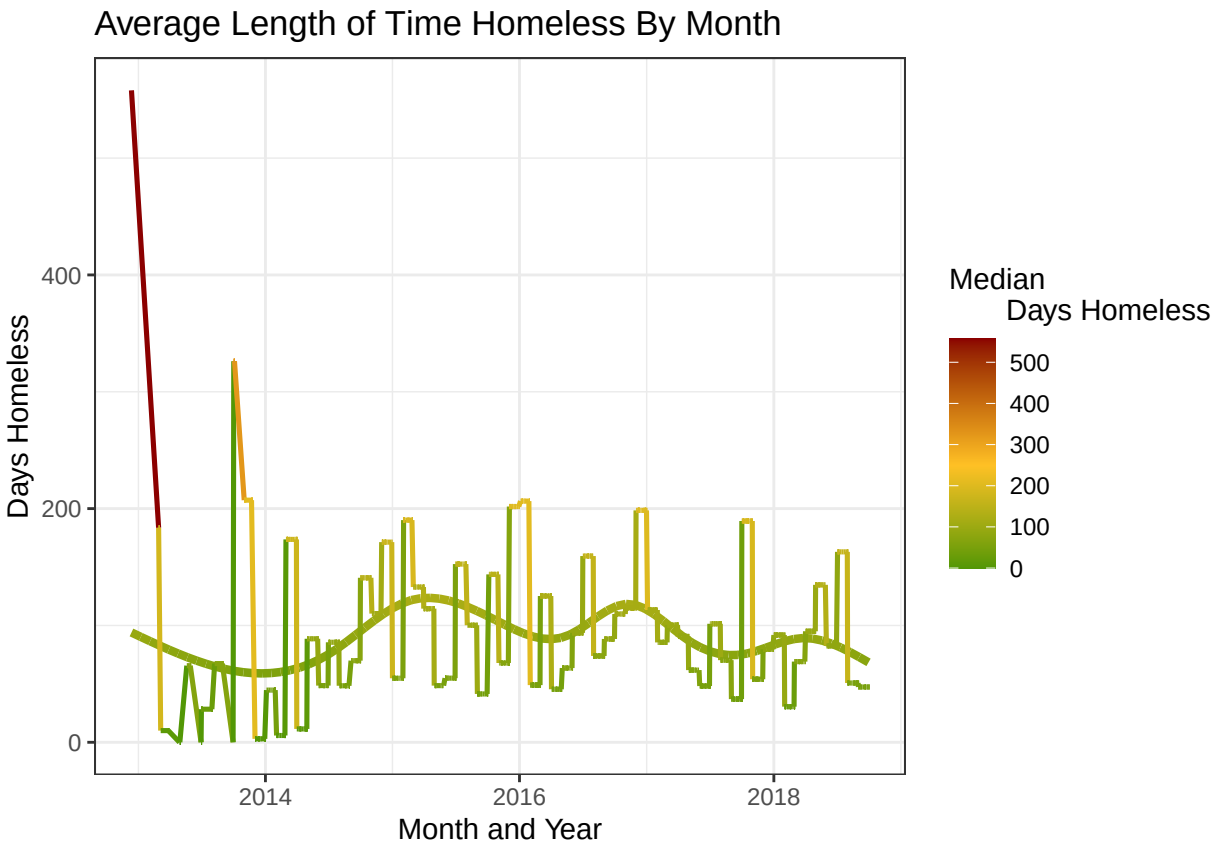
```
  geom_errorbar(aes(ymin = lower_ci, ymax = upper_ci,
                    width = 0.2, )) +
  labs(title = "Days to Return By Fiscal Year",
       x = "Fiscal Year",
       y = "Days To Return",
       col = "Primary Race") +
  theme_light()
```



First let's look at Days to Reappear by Fiscal Year
We can also look at each month

```
f_all <- d_month %>%
  select(client_uid, days_to_reappear, FY, race_fc, race_n, mean_y_m, median_y_m, y_m, year, month, end
  filter(between(days_to_reappear, 0, 600)) %>%
  na.omit()

ggplot(data = f_all, aes(end_date, mean_y_m, color = mean_y_m)) +
  geom_smooth(aes(color=..y..), size=1.5, se=FALSE)  +
  geom_line(size = 0.9)+
  scale_colour_gradient2(low = "green4", mid = "goldenrod1" , high = "darkred",
                         midpoint = 250) +
  labs(title = "Average Length of Time Homeless By Month",
       x = "Month and Year",
       y = "Days Homeless",
       color = "Median
       Days Homeless") +
  theme_bw()
```


Average Length of Time Homeless By Month
Days Homeless
400
200
0
2014
2016
2018
Month and Year
Median
Days Homeless
500
400
300
200
100
0