

# SPM1

Reagan Costello-White

6/18/2021

## System Performance Measure 1: Length of Time Homeless

### 1) Import SPM Data

```
library(readxl)
library(tidyverse)
library(janitor)
SPM1_FYTD15 <- read_excel("data/SPM1_FYTD15.xls",
  sheet = "Tab B - Detail") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(client_uid = as.numeric(client_uid),
    FY = 15)
SPM1_FYTD16 <- read_excel("data/SPM1_FYTD16.xls",
  sheet = "Tab B - Detail") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(FY = 16)
SPM1_FYTD17 <- read_excel("data/SPM1_FYTD17.xls",
  sheet = "Tab B - Detail") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(FY = 17)
SPM1_FYTD18 <- read_excel("data/SPM1_FYTD18.xls",
  sheet = "Tab B - Detail") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(FY = 18)
SPM1_FYTD19_a <- read_excel("data/SPM1_FYTD19.xls",
  sheet = "Tab B - Detail") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(FY = 19)
SPM1_FYTD19_b <- read_excel("data/SPM1_FYTD19.xls",
  sheet = "Tab B - Detail(1)") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(FY = 19)
SPM1_FYTD20 <- read_excel("data/SPM1_FYTD20.xls",
```

```

    sheet = "Tab B - Detail") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(FY = 20)
SPM1_FYTD21 <- read_excel("data/SPM1_FYTD21.xls",
  sheet = "Tab B - Detail") %>%
  clean_names() %>%
  as_tibble() %>%
  mutate(FY = 21)

```

## 2) Import Demographic Data

```

library(readr)
demographics_FY13_FYTD21 <- read_csv("data/demographics_FY13_FYTD21.csv")

```

## 3) Merge and Clean Data

```

library(tidyverse)
library(lubridate)
SPM1_FYTD19_a <- SPM1_FYTD19_a %>%
  select(-overlap_likely_hmi_error) ##### This variable is not in the second sheet of the FY19 report
SPM1_FYTD19 <- rbind(SPM1_FYTD19_a, setNames(SPM1_FYTD19_b, names(SPM1_FYTD19_a))) %>%
  mutate(date_move_in = ymd_hms(date_move_in)) ##### Only FY19 has date_move_in as a numeric variable
d_all_FY <- bind_rows(SPM1_FYTD15, SPM1_FYTD16, SPM1_FYTD17,
  SPM1_FYTD18, SPM1_FYTD19, SPM1_FYTD20, SPM1_FYTD21)
d_all <- left_join(d_all_FY, demographics_FY13_FYTD21, by = c("client_uid" = "client_id")) %>%
  group_by(client_uid) %>%
  slice(1) %>%
  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 Pacific Islander (HUD)",
    "American Indian or Alaska Native" = "American Indian or Alaska Native (HUD)"))
  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 Islander (HUD)",
        "American Indian or Alaska Native (HUD)", "Asian (HUD)")) %>%
mutate(race_n = as.numeric(race_fc)) %>%
mutate(entry_adj_base = ymd(entry_adj_base),
       exit_adj_base = ymd(exit_adj_base),
       adjusted_cutoff = ymd(adjusted_cutoff)) %>%
mutate(entry_exit_interval = interval(entry_adj_base, exit_adj_base),
       tran_lot_days = time_length(entry_exit_interval, "day")) %>%
mutate(tran_lot_sum = sum(tran_lot)) %>%
mutate(LOT_h = total_es_sh_th) %>%
mutate(end_date_ymd = as.Date(end_date, format = '%Y/%m/%d')) %>%
replace_na(list(end_date_ymd = today())) %>%
mutate(year = year(end_date_ymd),
       month = month(end_date_ymd)) %>%
mutate(y_m = paste(year, month, sep = "-")) %>%
relocate(client_uid, entry_adj_base, exit_adj_base, entry_exit_interval,
         tran_lot, tran_lot_days, total_es_sh, total_es_sh_th, LOT_h)
summary(d_all$race_2)

```

#### 4) Inspect Data

```

library(gt)
gt_all <- head(d_all)
gt_all

```

```

## # A tibble: 6 x 43
## # Groups:   client_uid [6]
##   client_uid entry_adj_base exit_adj_base entry_exit_interval
##         <dbl> <date>         <date>         <Interval>
## 1         3 2015-10-08      2015-10-09      2015-10-08 UTC--2015-10-09 UTC
## 2        33 2013-10-01      2014-04-05      2013-10-01 UTC--2014-04-05 UTC
## 3        37 2016-06-23      2016-06-23      2016-06-23 UTC--2016-06-23 UTC
## 4        42 2019-07-28      2019-07-28      2019-07-28 UTC--2019-07-28 UTC
## 5        59 2019-01-10      2019-01-11      2019-01-10 UTC--2019-01-11 UTC
## 6        61 2018-01-23      2018-02-10      2018-01-23 UTC--2018-02-10 UTC
## # ... with 39 more variables: tran_lot <dbl>, tran_lot_days <dbl>,
## #   total_es_sh <dbl>, total_es_sh_th <dbl>, LOT_h <dbl>, unique_id <chr>,
## #   trans_type <chr>, trans_id <dbl>, provider <chr>, proj_type <chr>,
## #   date_move_in <dtm>, start_date <dtm>, end_date <dtm>, x12 <lgl>,
## #   adjusted_cutoff <date>, overlap_likely_hmi_error <lgl>, FY <dbl>,
## #   last_name <chr>, first_name <chr>, entry_date <chr>, exit_date <chr>,
## #   client_location_always_choose_va_502_unless_directed_otherwise <chr>,
## #   total_monthly_income <dbl>, current_locality <chr>, date_of_birth <chr>,
## #   ethnicity <chr>, primary_race <chr>, race_f <fct>, proj_type_f <fct>,
## #   provider_f <fct>, race_fc <fct>, race_3 <fct>, race_2 <fct>, race_n <dbl>,
## #   tran_lot_sum <dbl>, end_date_ymd <date>, year <dbl>, month <dbl>, y_m <chr>

```

## 5) Summarize Data

```
library(gt)
sum_FY <- d_all %>%
  group_by(FY) %>%
  select(client_uid, LOT_h, FY, race_fc, race_f, total_es_sh_th, end_date_ymd, month, year)%>%
  na.omit()%>%
  summarise(n = n_distinct(client_uid),
            min = min(LOT_h, na.rm = TRUE),
            max = max(LOT_h, na.rm = TRUE),
            mean = mean(LOT_h, na.rm = TRUE),
            median = median(LOT_h, na.rm = TRUE),
            sd = sd(LOT_h, 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 = "Length of Time Homeless by Fiscal Year") %>%
  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),
            FY ~ px(200)) %>%
  cols_align(align = "center") %>%
  cols_label(FY = "Fiscal Year",
            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
```

Fiscal Year	n	Min	Max	Mean	Median	S. Dev	SEM	95% Confidence Intervals	
								Upper	Lower
15	414	1	932	80.90	51.5	98.93	4.86	90.43	71.37
16	1036	1	306	45.94	25.0	53.75	1.67	49.21	42.66
17	743	1	322	37.88	22.0	42.97	1.58	40.97	34.79
18	1338	1	249	28.07	11.0	40.89	1.12	30.26	25.87
19	1328	1	327	28.54	10.0	45.55	1.25	30.99	26.09

20	676	1	473	36.94	14.0	54.95	2.11	41.08	32.80
21	250	1	358	65.83	19.0	88.19	5.58	76.76	54.90

```

library(gt)
d_all <- d_all %>%
  group_by(year, month) %>%
  mutate(mean_y_m = mean(LOT_h, na.rm = TRUE),
         median_y_m = median(LOT_h, na.rm = TRUE))

sum_month <- d_all %>%
  group_by(year, month) %>%
  select(client_uid, LOT_h, FY, race_fc, race_f, total_es_sh_th, end_date_ymd, month, year, y_m, mean_y_m,
         na.omit()) %>%
  summarise(n = n_distinct(client_uid),
            min = min(LOT_h, na.rm = TRUE),
            max = max(LOT_h, na.rm = TRUE),
            mean = mean(LOT_h, na.rm = TRUE),
            median = median(LOT_h, na.rm = TRUE),
            sd = sd(LOT_h, na.rm = TRUE),
            sem = sd/sqrt(n()),
            upper_ci = mean + (1.96 * sem),
            lower_ci = mean - (1.96 * sem))

gt_month <- gt(sum_month) %>%
  tab_header(title = "Length of Time Homeless by Month") %>%
  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),
            mean ~ px(200)) %>%
  cols_align(align = "center") %>%
  cols_label(year = "Year",
            month = "Month",
            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_month

```

---

## Length of Time Homeless by Month

95% Confidence Intervals

---

Month	n	Min	Max	Mean	Median	S. Dev	SEM	Upper	Lower
2014									
4	1	202	202.00	202.00000	202.00	NA	NA	NA	NA
5	3	157	283.00	229.33333	248.00	65.04	37.55	302.93	155.732499
6	4	18	232.00	129.25000	133.50	92.44	46.22	219.84	38.660056
7	3	51	183.00	132.66667	164.00	71.36	41.20	213.42	51.914561
8	4	46	168.00	107.50000	108.00	49.92	24.96	156.42	58.581735
9	2	32	370.00	201.00000	201.00	239.00	169.00	532.24	-130.240000
10	36	3	343.00	64.63889	33.00	72.29	12.05	88.25	41.024687
11	26	13	277.00	74.61538	46.00	60.34	11.83	97.81	51.420510
12	29	11	318.00	90.79310	67.00	70.10	13.02	116.31	65.278521
2015									
1	26	6	225.00	78.73077	57.00	64.16	12.58	103.39	54.069361
2	28	2	145.00	57.57143	58.00	44.25	8.36	73.96	41.180508
3	25	2	322.00	70.08000	50.00	74.48	14.90	99.28	40.881938
4	44	2	565.00	95.29545	61.50	96.75	14.59	123.88	66.708119
5	22	6	381.00	109.77273	40.00	126.27	26.92	162.54	57.005844
6	34	3	189.00	49.35294	51.50	40.35	6.92	62.92	35.790092
7	29	11	409.00	68.72414	47.00	76.27	14.16	96.48	40.965318
8	29	12	476.00	76.17241	41.00	114.32	21.23	117.78	34.562565
9	29	1	265.00	65.31034	52.00	60.60	11.25	87.37	43.252895
10	138	1	270.00	53.66667	32.00	58.50	4.98	63.43	43.906530
11	78	1	268.00	50.75641	32.50	54.57	6.18	62.87	38.646229
12	65	1	174.00	43.75385	29.00	43.22	5.36	54.26	33.246360
2016									
1	71	1	212.00	40.80282	24.00	45.90	5.45	51.48	30.127193
2	72	1	227.00	42.59722	30.50	47.93	5.65	53.67	31.527040
3	63	1	286.00	49.41270	25.00	61.22	7.71	64.53	34.294441
4	100	1	271.00	68.27000	51.00	69.69	6.97	81.93	54.610521
5	119	1	233.00	44.48739	20.00	56.74	5.20	54.68	34.292925
6	99	1	191.00	41.35354	25.00	45.88	4.61	50.39	32.315903
7	75	1	850.00	47.98667	18.00	103.42	11.94	71.39	24.581146
8	90	1	203.00	29.77778	17.50	37.74	3.98	37.58	21.980115
9	78	1	409.00	33.41026	14.50	59.84	6.78	46.69	20.130204
10	63	1	932.00	78.28571	35.00	131.75	16.60	110.82	45.752201
11	49	1	309.00	50.20408	33.00	58.57	8.37	66.60	33.804750
12	60	1	204.00	42.41667	27.00	46.65	6.02	54.22	30.613532
2017									
1	74	1	242.00	47.37838	31.50	47.87	5.56	58.29	36.471707
2	59	1	232.00	43.28814	25.00	45.65	5.94	54.94	31.639873
3	60	1	137.00	30.40000	15.00	34.33	4.43	39.09	21.714560
4	63	1	178.00	35.92063	18.00	43.51	5.48	46.67	25.175355
5	70	1	143.00	44.48571	30.50	41.44	4.95	54.19	34.776650
6	55	1	182.00	43.32727	25.00	41.20	5.56	54.22	32.438061
7	58	1	167.00	43.75862	31.00	38.23	5.02	53.60	33.918942
8	80	1	86.00	26.90000	24.00	19.61	2.19	31.20	22.603365
9	79	1	95.00	13.12658	9.00	14.50	1.63	16.32	9.929937
10	41	1	204.00	34.97561	16.00	46.45	7.25	49.19	20.756837
11	27	1	128.00	21.51852	14.00	30.17	5.81	32.90	10.138620
12	31	1	188.00	28.38710	11.00	41.05	7.37	42.84	13.935516

2018

1	103	1	248.00	48.08738	31.00	50.18	4.94	57.78	38.395919
2	178	1	249.00	54.58989	28.50	64.93	4.87	64.13	45.051177
3	162	1	178.00	25.87037	7.00	38.42	3.02	31.79	19.953659
4	123	1	163.00	28.47967	13.00	37.20	3.35	35.05	21.906079
5	117	1	126.00	19.88034	8.00	27.89	2.58	24.93	14.827359
6	137	1	119.00	18.72263	8.00	24.97	2.13	22.90	14.541315
7	121	1	181.00	22.86777	9.00	31.77	2.89	28.53	17.206980
8	140	1	85.00	19.71429	12.50	20.17	1.70	23.06	16.373258
9	151	1	109.00	11.67550	6.00	17.23	1.40	14.42	8.927695
10	119	1	326.00	43.26050	19.00	56.54	5.18	53.42	33.101584
11	130	1	327.00	42.07692	12.50	77.00	6.75	55.31	28.840535
12	95	1	303.00	32.14737	11.00	53.96	5.54	43.00	21.296103

2019

1	102	1	265.00	36.05882	19.00	47.28	4.68	45.23	26.884212
2	108	1	230.00	32.75926	10.00	48.66	4.68	41.94	23.582807
3	117	1	176.00	29.26496	12.00	37.87	3.50	36.13	22.402569
4	109	1	152.00	23.45872	4.00	37.52	3.59	30.50	16.415725
5	103	1	146.00	22.15534	5.00	34.66	3.42	28.85	15.460897
6	94	1	116.00	24.14894	12.00	29.15	3.01	30.04	18.256845
7	118	1	99.00	22.05932	11.00	24.20	2.23	26.43	17.693404
8	116	1	242.00	19.01724	8.00	28.96	2.69	24.29	13.746196
9	104	1	136.00	11.13462	7.00	16.12	1.58	14.23	8.035692
10	100	1	344.00	42.27000	20.00	61.65	6.17	54.35	30.186212
11	97	1	290.00	32.42268	9.00	60.46	6.14	44.45	20.391315
12	83	1	473.00	25.53012	6.00	58.58	6.43	38.13	12.928322

2020

1	102	1	274.00	34.27451	9.00	56.17	5.56	45.17	23.374439
2	57	1	233.00	38.15789	10.00	60.46	8.01	53.85	22.462644
3	46	1	195.00	41.23913	12.00	56.80	8.37	57.65	24.824737
4	18	1	200.00	37.83333	11.50	50.83	11.98	61.32	14.349524
5	19	1	154.00	47.15789	23.00	50.58	11.60	69.90	24.414731
6	36	1	129.00	54.61111	39.00	48.50	8.08	70.45	38.768951
7	30	1	91.00	32.60000	22.50	27.93	5.10	42.60	22.603890
8	41	1	68.00	25.68293	23.00	20.86	3.26	32.07	19.297065
9	33	1	296.00	31.54545	14.00	54.04	9.41	49.98	13.106452
10	33	1	327.00	46.75758	17.00	63.89	11.12	68.56	24.959788
11	34	1	312.00	71.61765	31.00	96.10	16.48	103.92	39.315914
12	38	1	299.00	50.63158	25.50	69.69	11.31	72.79	28.472066

2021

1	35	1	358.00	35.11429	13.00	63.32	10.70	56.09	14.134864
2	26	1	235.00	41.42308	11.00	64.52	12.65	66.22	16.624105
3	34	1	205.00	41.00000	12.50	55.76	9.56	59.74	22.258346
4	30	1	178.00	46.30000	26.00	54.39	9.93	65.76	26.836742
5	21	1	151.00	69.71429	8.00	71.34	15.57	100.23	39.199809
6	34	35	350.00	180.35294	159.50	93.20	15.98	211.68	149.023649

```
library(gt)
sum_race <- d_all %>%
  group_by(race_fc) %>%
```

```

select(client_uid, LOT_h, FY, race_fc, race_f, total_es_sh_th, end_date_ymd, month, year)%>%
na.omit()%>%
summarise(n = n_distinct(client_uid),
          min = min(LOT_h, na.rm = TRUE),
          max = max(LOT_h, na.rm = TRUE),
          mean = mean(LOT_h, na.rm = TRUE),
          median = median(LOT_h, na.rm = TRUE),
          sd = sd(LOT_h, na.rm = TRUE),
          sem = sd/sqrt(n()),
          upper_ci = mean + (1.96 * sem),
          lower_ci = mean - (1.96 * sem))%>%
arrange(desc(n))

gt_race <- gt(sum_race) %>%
  tab_header(title = "Length of Time Homeless by Client Race") %>%
  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),
            race_fc ~ 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_race

```

Length of Time Homeless by Client Race

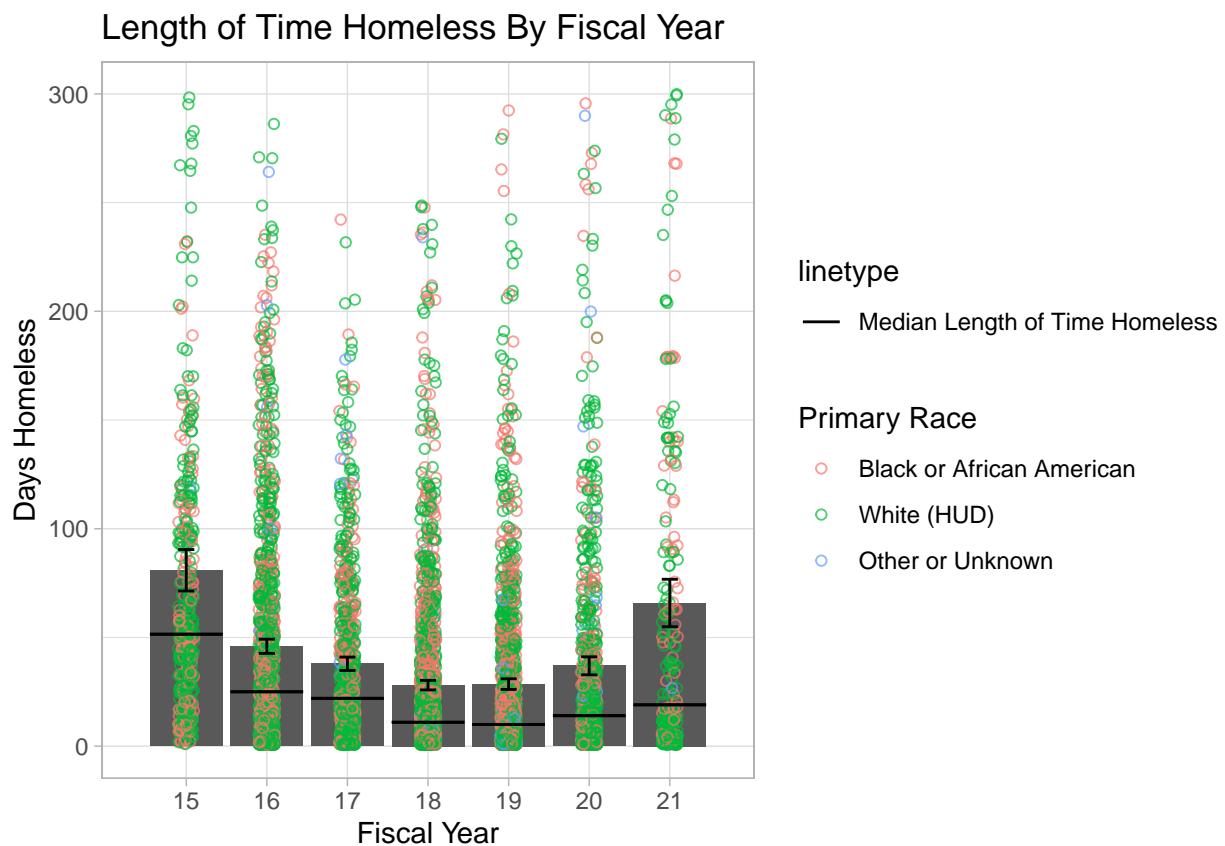
Race	n	Min	Max	Mean	Median	S. Dev	SEM	95% Confidence Intervals	
								Upper	Lower
White	3462	1	850	38.72	16	57.26	0.97	40.63	36.82
Black or African American	2207	1	932	39.61	20	55.88	1.19	41.94	37.28
American Indian or Alaska Native	53	1	290	49.42	25	67.83	9.32	67.68	31.15
Unknown	36	1	234	34.39	11	54.92	9.15	52.33	16.45
Asian	14	1	122	21.79	6	33.73	9.01	39.45	4.12
Native Hawaiian or Pacific Islander	13	1	132	35.69	11	44.91	12.46	60.10	11.28



## 6) Plot Data

```
f_all <- d_all %>%
  filter(between(LOT_h, 1, 300))%>%
  drop_na(race_3)

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, LOT_h, 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, )) +
  geom_errorbar(aes(ymax = median, ymin = median, linetype = "Median Length of Time Homeless")) +
  labs(title = "Length of Time Homeless By Fiscal Year",
       x = "Fiscal Year",
       y = "Days Homeless",
       col = "Primary Race") +
  theme_light()
```



```
ggsave("LOT_homeless x FY .jpg")
```

```
f_all <- d_all %>%
  select(client_uid, LOT_h, FY, race_fc, race_n, mean_y_m, end_date_ymd, median_y_m)%>%
  filter(between(LOT_h, 0, 300)) %>%
  na.omit()

ggplot(data = f_all, aes(end_date_ymd, median_y_m, color = median_y_m)) +
  geom_smooth(aes(color=..y..), size=1.5, se=FALSE) +
  geom_line(size = 0.9)+
  scale_x_date(date_labels = "%y",
               date_breaks = "1 year")+
  scale_colour_gradient2(low = "green4", mid = "goldenrod1", high = "darkred",
                        midpoint=100) +
  labs(title = "Average Length of Time Homeless By Month",
       x = "Month and Year",
       y = "Days Homeless",
       color = "Median
Days Homeless") +
  theme_bw()
```

