

R Programming Hw4

Amy Fox

October 24, 2018

```
knitr::opts_chunk$set(message = FALSE)
```

Load necessary packages

```
library(readr)
library(tidyr)
library(dplyr)
library(forcats)
library(broom)
library(purrr)
library(ggplot2)
library(knitr)
```

Read in homicide csv

```
homicides <- read_csv("../data/homicide-data.csv")
```

See first few lines of data to see what's there

```
head(homicides)
```

```
## # A tibble: 6 x 12
##   uid      reported_date victim_last victim_first victim_race victim_age
##   <chr>          <int> <chr>      <chr>        <chr>      <chr>
## 1 Alb-000001    20100504 GARCIA     JUAN          Hispanic    78
## 2 Alb-000002    20100216 MONTOYA   CAMERON       Hispanic    17
## 3 Alb-000003    20100601 SATTERFIELD VIVIANA       White       15
## 4 Alb-000004    20100101 MENDIOLA   CARLOS        Hispanic    32
## 5 Alb-000005    20100102 MULA      VIVIAN        White       72
## 6 Alb-000006    20100126 BOOK      GERALDINE     White       91
## # ... with 6 more variables: victim_sex <chr>, city <chr>, state <chr>,
## #   lat <dbl>, lon <dbl>, disposition <chr>
```

Unite the city and state columns to make a general location column

Remove Tulsa, AL because does not exist and skews data

```
homicides <- homicides %>%
  unite(col = "location", c("city", "state"),
        sep = ", ") %>%
  filter(location != "Tulsa, AL")
```

Key for myself

unsolved = closed without arrest or open/no arrest

Create new dataframe with unsolved cases info

Select only necessary columns

Make a new True/False column for unsolved cases

Group by the location and summarize the number of unsolved and the total cases

```
unsolved_df <- homicides %>%
  select(location, disposition) %>%
  mutate(unsolved = disposition != "Closed by arrest") %>%
```

```
group_by(location) %>%
  summarise(total_cases = n(), unsolved = sum(unsolved))

head(unsolved_df)
```

```
## # A tibble: 6 x 3
##   location      total_cases unsolved
##   <chr>          <int>     <int>
## 1 Albuquerque, NM      378      146
## 2 Atlanta, GA         973      373
## 3 Baltimore, MD      2827     1825
## 4 Baton Rouge, LA      424      196
## 5 Birmingham, AL       800      347
## 6 Boston, MA          614      310
```

Create new dataframe with only data for Baltimore
 Perform proportion test on Baltimore cases
 Tidy prop_test data

```
Baltimore_df <- unsolved_df %>%
  filter(location == "Baltimore, MD")

Baltimore_prop_test <- prop.test(x= Baltimore_df$unsolved,
                                n = Baltimore_df$total_cases)
```

```
#print output of prop.test
Baltimore_prop_test
```

```
##
## 1-sample proportions test with continuity correction
##
## data:  Baltimore_df$unsolved out of Baltimore_df$total_cases, null probability 0.5
## X-squared = 239.01, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.6275625 0.6631599
## sample estimates:
##      p
## 0.6455607
```

```
# print tidied prop.test
tidied_Baltimore_prop_test <- tidy(Baltimore_prop_test)
tidied_Baltimore_prop_test
```

```
##   estimate statistic      p.value parameter  conf.low conf.high
## 1 0.6455607    239.011 6.461911e-54          1 0.6275625 0.6631599
##                                     method alternative
## 1 1-sample proportions test with continuity correction two.sided
```

```
#pull out estimate proportion and confidence intervals
tidied_Baltimore_prop_test$estimate
```

```
## [1] 0.6455607
```

```
tidied_Baltimore_prop_test$conf.low
```

```
## [1] 0.6275625
```

```
tidied_Baltimore_prop_test$conf.high
```

```
## [1] 0.6631599
```

Create new column with prop.test of each city using map2

Create new column with tidied proptest data

Unnest data -> tidy data from list to df

Reorder location by estimate

Plot cities according to the estimate for unsolved cases showing the 95% confidence interval

Change x axis from decimal to percent

Add labels

(Must change fig.width to fig_width for PDF)

```
tidy_cities_prop <- unsolved_df %>%
  mutate(my_prop_test = map2(unsolved, total_cases, prop.test),
         tidy_prop_test = map(my_prop_test, tidy)) %>%
  unnest(tidy_prop_test, .drop = TRUE) %>%
  mutate(location = factor(location, levels = location[order(estimate)]))

tidy_cities_prop %>%
  ggplot(aes(estimate, location)) +
  geom_point(color = "white") +
  geom_errorbarh(aes(xmin = conf.low,
                    xmax = conf.high,
                    height = 0), color = "white") +
  scale_x_continuous(labels= scales::percent) +
  ggtitle("Unsolved homicides by city", subtitle = "Bars show 95% confidence interval") +
  xlab("Percent of homicides that are unsolved")+
  ylab("") +
  theme_dark()
```

Unsolved homicides by city

Bars show 95% confidence interval

