RA_test_ARSHAD_AZAD

arshad

11/01/2022

```
library(tidyverse, warn.conflicts = F)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

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

```
library(haven)
library(naniar)
library(ggplot2)
library(ggpubr)
```

Part 1

1. Importing

```
# a. Importing 'New Variables.csv' as new_variables

new_variables <- read.csv("~/Desktop/RA_test/coding_challenge/Part 1/New Variables.csv")


# b. Importing 'Main Dataset.dta' as main_dataset

main_dataset <- read_dta("~/Desktop/RA_test/coding_challenge/Part 1/Main Dataset.dta")

## merging new_variables with main_dataset

main_dataset <- main_dataset %>% left_join(new_variables, by = "uniqueid")

# c. Importing 'New Observations.dta' as new_observations
```

```
new_observations <- read_dta("~/Desktop/RA_test/coding_challenge/Part 1/New Observations.dta")

## adding new_observation to main_dataset

main_dataset <- bind_rows(main_dataset, new_observations)
```

2. Quality checks

a. average and median values of time spent surveying for completed surveys

```
main_dataset$surveytime <- as.numeric(main_dataset$surveytime)

avg_time <- mean(main_dataset$surveytime[main_dataset$survey_complete == 1])
paste0("average time spent is ", avg_time)
```

```
## [1] "average time spent is 7201.99555555556"
```

```
median_time <- median(main_dataset$surveytime[main_dataset$survey_complete == 1])
paste0("median time spent is ", median_time)
```

```
## [1] "median time spent is 7165"
```

b. survey time variations among surveyors

```
df_survey_time <- main_dataset %>% select(surveyor, survey_complete, surveytime) %>% filter(survey_compl

avg_time_for_surveyors <- unique(paste0("Surveyor ", df_survey_time$surveyor, " spent ", df_survey_time$

avg_time_for_surveyors
```

```
##  [1] "Surveyor Benjamin spent 7242.25196850394 seconds."
##  [2] "Surveyor Peter spent 7241.18421052632 seconds."
##  [3] "Surveyor Anna spent 7204.74166666667 seconds."
##  [4] "Surveyor Mary spent 7283.44202898551 seconds."
##  [5] "Surveyor John spent 7154.63076923077 seconds."
##  [6] "Surveyor Caroline spent 7058.62068965517 seconds."
##  [7] "Surveyor Grace spent 7416.13333333333 seconds."
##  [8] "Surveyor David spent 7163.47552447552 seconds."
##  [9] "Surveyor Joseph spent 7168 seconds."
## [10] "Surveyor Sam spent 7161.70652173913 seconds."
## [11] "Surveyor Jane spent 6855.86666666667 seconds."
```

c. Duplicate 'hhid'.

```
## First we make a table with frequecies for each hhid
n_occur <- data.frame(table(main_dataset$hhid))

## Following gives the hhid with more than 1 entries
n_occur[n_occur$freq > 1,]
```

```
## [1] Var1 Freq
## <0 rows> (or 0-length row.names)
```

This subsets and then finds the 'uniqueid' for which 'hhid' are repeated, i.e, duplicate 'hhid' exists.

```
main_dataset[main_dataset$hhid %in% n_occur$Var1[n_occur$Freq > 1],]$uniqueid
```

```
## [1] 597 598 641 642 679 680
```

We can resolve the duplicate id problem as follows. Make a composite 'hhid2' from 'hhid' and 'uniqueid'. This gives a unique 'hhid2'.

```
main_dataset$hhid2 <- paste0(main_dataset$uniqueid,main_dataset$hhid)

length(main_dataset$hhid2)
```

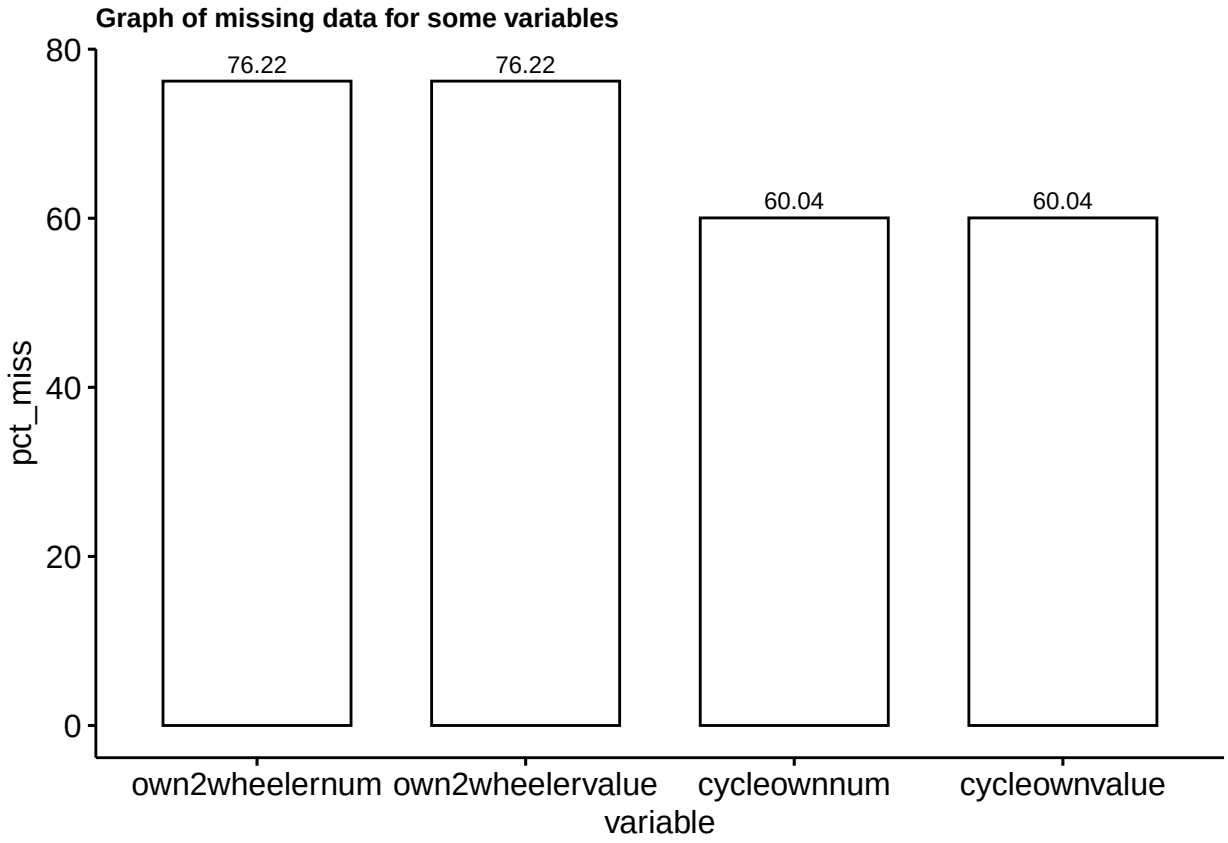
```
## [1] 1001
```

d. Missing value plot

```
# We select following 4 variables.

missing_data <- main_dataset %>% select(cycleownnum, cycleownvalue, own2wheelernum, own2wheelervalue) %

missing_data
```

```
## # A tibble: 4 x 3
##   variable         n_miss pct_miss
##   <chr>             <int>    <dbl>
## 1 own2wheelernum      763     76.2
## 2 own2wheelervalue    763     76.2
## 3 cycleownnum         601     60.0
## 4 cycleownvalue       601     60.0
```

```
missing_data_plot <- ggbarplot(missing_data, x = "variable", y = "pct_miss", add = "none") + stat_summa
             vjust = -0.5) + ggtitle("Graph of missing data for some variables") + theme(plot.title = e

missing_data_plot
```



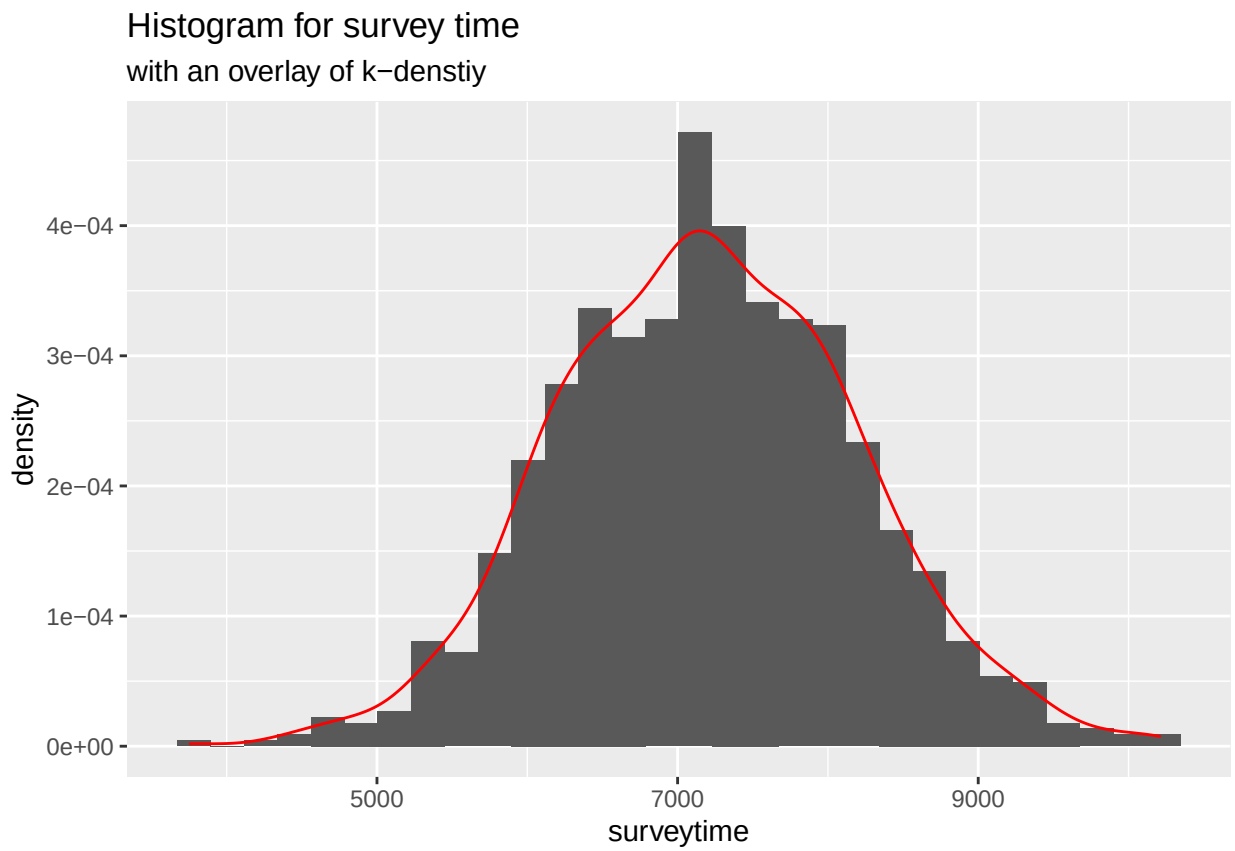

```
ggsave("missing_data_plot.png")
```

```
## Saving 6.5 x 4.5 in image
```

e. histograms and k-density plots

```
ggplot(main_dataset, aes(surveytime)) +
  geom_histogram(aes(y=..density..)) +
  geom_density(col = "red") +
  ggtitle("Histogram for survey time", subtitle = "with an overlay of k-denstiy")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

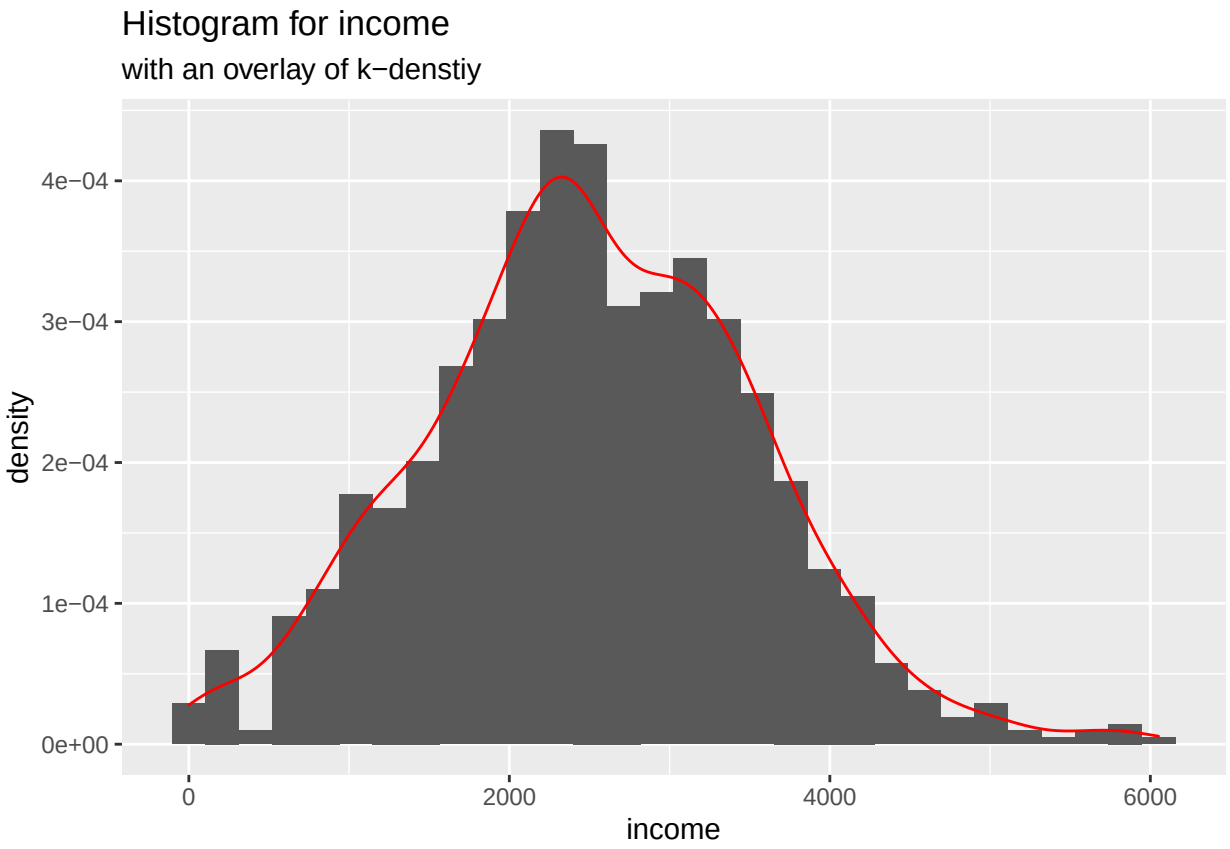



```
ggsave("hist_survey_time.png")
```

```
## Saving 6.5 x 4.5 in image
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
ggplot(main_dataset, aes(income)) +
  geom_histogram(aes(y=..density..)) +
  geom_density(col = "red") +
  ggtitle("Histogram for income", subtitle = "with an overlay of k-denstiy")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggsave("hist_income.png")
```

```
## Saving 6.5 x 4.5 in image
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

3. Cleaning

a. removing personally identifiable information

```
main_dataset$surveyor <- str_replace_all(main_dataset$surveyor, c("Benjamin" = "1", "Peter" = "2", "Anna

main_dataset$surveyor <- as.numeric(main_dataset$surveyor)

unique(main_dataset$surveyor)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11
```

b. recoding missing values

```
## for variable 'burglaryyn'
main_dataset$burglaryyn <- str_replace_all(main_dataset$burglaryyn, c("-997" = "Refuse to answer", "-99

unique(main_dataset$burglaryyn)
```

```
## [1] "2"                "1"                "Refuse to answer" "Don't Know"
## [5] NA
```

```
## for variable 'vandalismyn'
main_dataset$vandalismyn <- str_replace_all(main_dataset$vandalismyn, c("-999" = "Don't Know"))
unique(main_dataset$vandalismyn)
```

```
## [1] "2"          "Don't Know" "1"           NA
```

```
## for variable 'trespassingyn'
main_dataset$trespassingyn <- str_replace_all(main_dataset$trespassingyn, c("-997" = "Refuse to answer"
unique(main_dataset$trespassingyn)
```

```
## [1] "2"                "1"                "Refuse to answer" NA
```

Part 2

1.) Crop Insurance Project

```
# Loading data

farmer_ind <- read_dta("~/Desktop/RA_test/coding_challenge/Part 2/A/farmer_ind.dta")
income <- read_dta("~/Desktop/RA_test/coding_challenge/Part 2/A/income.dta")
mkt_prices <- read_dta("~/Desktop/RA_test/coding_challenge/Part 2/A/mkt_prices.dta")


## merging data

farmer_full <- farmer_ind %>%
  left_join(income, by = "farmer_id") %>%
  left_join(mkt_prices, by = "crop_id")
```

a). Calculating income of cocoa farmers and others

```
income_cocoa_farmers <- mean(farmer_full$income[farmer_full$crop_id == 8])

income_cocoa_farmers
```

```
## [1] 4586.661
```

```r
income_other_farmers <- mean(farmer_full$income[farmer_full$crop_id != 8])

income_other_farmers
```

```
## [1] 4586.661
```

Both the values are numerically same.

```r
## checking for the statistical difference
diff_mean <- t.test(farmer_full$income[farmer_full$crop_id == 8], farmer_full$income[farmer_full$crop_i

diff_mean
```

```
##
##  Welch Two Sample t-test
##
## data:  farmer_full$income[farmer_full$crop_id == 8] and farmer_full$income[farmer_full$crop_id != 8]
## t = 0, df = 454.24, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -177.5542  177.5542
## sample estimates:
## mean of x mean of y
##  4586.661  4586.661
```

A p-value of 1 confirms that the difference is statistically zero. The income of farmers who grow cocoa have income statistically equal to those who do not. This provides no statistical support for or against the growing of cocoa. Based on the data no suggestions can be provided to pursue or go against the growing of cocoa.

b). Estimating the market price of squash.

For each farmer, we have the information on their quantity of a crop sold, their total income, and the market price of the all crops sans squash. We estimate the price of the squash as follows.

We estimate P_8 by calculating residual income. Residual income is the income of a farmer from the sale of the crops sans squash.

$$\sum_{1}^{8} P_j \, x_{ij} = M_i$$

or,

$$P_8 x_{i8} + \sum_{1}^{8} P_j \, x_{ij} = M_i$$

or,

$$M_i - \sum_{1}^{8} P_j \, x_{ij} = P_8 x_{i8}$$

or,

$$M_i^* = P_8 x_{i8}$$

or,

$$M_i^* = \hat{P}_8 x_{i8} + e_i$$

We then have the estimate $\hat{P}_8$ due to linear regression.

```
## replacing '-99' with NA for variable 'sale_amt'


farmer_full$sale_amt <- str_replace_all(farmer_full$sale_amt, c("-99" = "NA"))
farmer_full$sale_amt <- as.numeric(farmer_full$sale_amt)
```

```
## Warning: NAs introduced by coercion
```

A large number of farmers do not have 'sale amount' data for several crops. Removing the farmers or removing the individual observations of crops is not feasible. We impute the missing value by the average 'sale amount' or corresponding crops.

```
## creating a varible 'avg_sale_amt' as follows.
farmer_full <- farmer_full %>% group_by(crop_id) %>% mutate(avg_sale_amt = mean(sale_amt, na.rm = T))
```

- step 1: estimate residual income We estimate the farmers residual M_i^* as follows. 'farmer_res' is the data set that contains all the information about crop prices and quantity sold sans squash. After value imputation of the missing values, we calculate residual income as shown above.

```
## farmers residual

farmer_res <- farmer_full %>% select(farmer_id, crop_id, sale_amt, mkt_price, income) %>% filter(crop_id
```

- step 2: We then regress residual income on the sale amount for squash.

Eventually, we have the estimate $\hat{P}_8$

2.) Vitamin supplement Project

```
# Loading data

vitamins <- read_dta("~/Desktop/RA_test/coding_challenge/Part 2/B/vitamins.dta")
```

a. Average solve time

Variable 'time' in data is of format 'MM.SS'. We extract the minute and second part, multiply the minute part with 60 and add the second part. We then have the 'time' variable in seconds.

```
vitamins$ch <- as.character(vitamins$time)
vitamins[c('min', 'sec')] <- str_split_fixed(vitamins$ch, '\\.', 2)

vitamins$sec <- as.numeric(vitamins$sec)
vitamins$sec[is.na(vitamins$sec)] <- 0
vitamins$min <- as.numeric(vitamins$min)

vitamins$tot_sec <- vitamins$min * 60 + vitamins$sec

avg_time <- mean(vitamins$tot_sec)
paste0("average solve time is ", avg_time)
```

```
## [1] "average solve time is 234.228"
```

b. Statistical difference between the solve times of treatment and control groups.

```
### different avg times
avg_time_treatment <- mean(vitamins$tot_sec[vitamins$treat == 1])
paste0("average solve time for treatment group is ", avg_time_treatment)
```

```
## [1] "average solve time for treatment group is 230.254166666667"
```

```
avg_time_control <- mean(vitamins$tot_sec[vitamins$treat == 0])
paste0("average solve time for control group is ", avg_time_control)
```

```
## [1] "average solve time for control group is 237.896153846154"
```

We check the statistical difference of means of different groups. Due to unequal sizes of the groups of treament and control, and uneuqal 'sd', a two sample welch-test is conducted.

```
diff_test_time <- t.test(vitamins$tot_sec[vitamins$treat == 1], vitamins$tot_sec[vitamins$treat == 0])

diff_test_time
```

```
##
##  Welch Two Sample t-test
##
## data:  vitamins$tot_sec[vitamins$treat == 1] and vitamins$tot_sec[vitamins$treat == 0]
## t = -1.6358, df = 462.27, p-value = 0.1026
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.822183   1.538209
## sample estimates:
## mean of x mean of y
##  230.2542  237.8962
```

We fail to reject the null hypothesis in favor of the alternate hypothesis. Hence, we conclude that true difference in means is equal to 0. The treatment group does not fare well statistically in the test compared to the control group, i.e, we find no evidence of impact of vitamin supplement on the analytical ability. Also, a p-value greater than 0.1 shows that at the 95% or 90% significance level the mathematical difference is likely to be a fluke.

c. Better supplement

It is likely that 'A' and 'a' refer to the same supplement. Similarly for 'D', 'B', and 'C'. Replacing them appropriately and capitalizing the rest.

```
supp_list <- unique(vitamins$supplement)
supp_list
```

```
##  [1] "B"  ""   "A"  "C"  "c"  "d"  "3"  "2"  "D"  "b"  "a"  "n"  "r"  " a"
```

```r
vitamins$supplement <- str_replace_all(
  vitamins$supplement, c("c" = "C", "d" = "D",
                         "b" = "B", "a" = "A",
                         "n" = "N", "r" = "R",
                         " A" = "A"))
```

Since control group were not provided any supplement, their column is blank for supplements.

```r
vitamins %>% filter(treat == 1) %>% group_by(supplement) %>% summarise(tot_time_grp = mean(tot_sec)) %>%
```

```
## # A tibble: 1 x 2
##   supplement tot_time_grp
##   <chr>             <dbl>
## 1 R                   191
```

'R' is the better supplement.