Reproducability Report

Christopher Keith Yurris

2022-09-27

Summary

The project was fully reproducible with no issues, besides the visuals of the figures.

Data Sources

This report is using data from the Statistics Canada 2016 Census Public Use Microdata File.

Replicator’s Computing Environment

Software:

Edition Windows 10 Home Version 21H2 Installed on 3/10/2021 OS build 19044.2006 Experience Windows Feature Experience Pack 120.2212.4180.0 R version 4.1.2 (2021-11-01) ##Hardware: Processor Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz Installed RAM 8 GB System type 64-bit operating system, x64-based processor

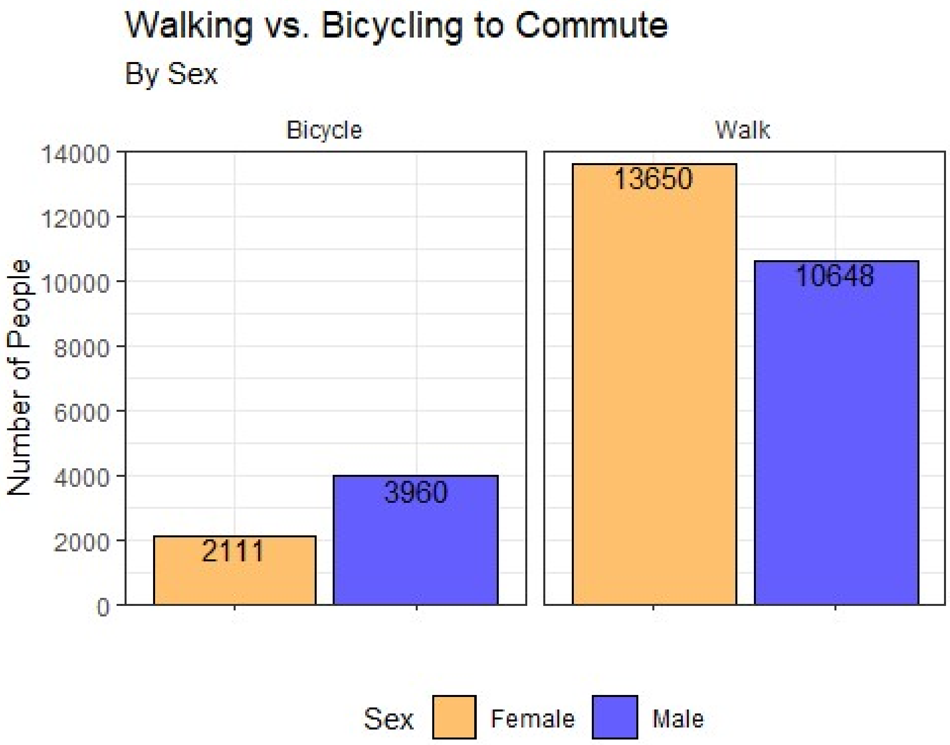
1. Load Dataset and choose the variables

```
#Load dataset and choose the variables:
census <- read_csv("pumf-98M0001-E-2016-individuals_F1.csv") %>%
select(AGEGRP,
                                                                          PR,

MODE,

PWLEAVE,

PWDUR,

DIST,

TotInc,

NOC16,

PWPR,
VisMin,

Sex,

POB) %>%
  #Remove the missing values/"not available" and inclusion only the MODE
Bike=1 e Walked=2.
  subset((MODE == 1 |
           MODE == 7) &

TotInc !=  88888888 &

PWPR != 88 &

TotInc != 99999999 &

DIST != 9 &

DIST != 8 &

PWDUR != 9 &

NOC16 != 99 &

NOC16 != 88 &

VisMin != 99 &

VisMin != 88 &

POB != 88) %>%
  #Recode of variables: MODE, Sex
  mutate(Modality =
           dplyr::recode(
             MODE,
             "1" ='Bicycle' ,
             "7" = 'Walk' ),

           Sexe = dplyr::recode(
             Sex,
             "1"='Female',
             "2"='Male'))

## Rows: 930421 Columns: 141
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (141): PPSORT, WEIGHT, WT1, WT2, WT3, WT4, WT5, WT6, WT7, WT8, WT9,
WT10...
```

```
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

2. Data Visualization - Use ggplot and geom_bar for graphics.

```
as.data.frame(table(census$Modality, census$Sexe)) %>%  ggplot(aes(x = Var2,
                                                                   y = Freq,
                      fill = Var2)) +#Fill as sex
  facet_wrap(~Var1)+
geom_col(position = "dodge", color = "black") + labs(y = "Number of People",
                  fill = "Sex", title = "Walking vs. Bicycling to Commute",
                  subtitle = "By Sex", x = "") + scale_y_continuous(expand =
c(0,0), limits = c(0, 14000), breaks = seq(0, 14000, 2000)) + theme_bw() +
scale_fill_manual(values = c("#ffc06e", "#645eff")) + geom_text(aes(label =
Freq, y = Freq - 400)) + theme(legend.position = "bottom",

panel.spacing = unit(.5, "lines"),
        strip.background = element_blank(),
        strip.placement = "outside",
          axis.text.x=element_blank())
```



3. Descriptive Statistics.

```
#avg income

mean(census$TotInc)

## [1] 45747.74

#median income

median(census$TotInc)

## [1] 30000

#Function to calculate for each category:


calcincome <- function(x){
  summary <- census %>% group_by_(x) %>% summarise("Median Income" =
median(TotInc),

"Average Income" = mean(TotInc))

return(summary)
}




map(c("Modality",
      "Sexe",
      "NOC16",
      "PR",
      "VisMin"), calcincome)

## Warning: `group_by_()` was deprecated in dplyr 0.7.0.
## ℹ Please use `group_by()` instead.
## ℹ See vignette('programming') for more help

## Warning: `group_by_()` was deprecated in dplyr 0.7.0.
## ℹ Please use `group_by()` instead.
## ℹ See vignette('programming') for more help

## Warning: `group_by_()` was deprecated in dplyr 0.7.0.
## ℹ Please use `group_by()` instead.
## ℹ See vignette('programming') for more help

## Warning: `group_by_()` was deprecated in dplyr 0.7.0.
## ℹ Please use `group_by()` instead.
## ℹ See vignette('programming') for more help
## Warning: `group_by_()` was deprecated in dplyr 0.7.0.
## ℹ Please use `group_by()` instead.
## ℹ See vignette('programming') for more help

## [[1]]
## # A tibble: 2 × 3
##   Modality `Median Income` `Average Income`
##   <chr>              <dbl>            <dbl>
## 1 Bicycle            39000           56541.
## 2 Walk               28000           43051.
##
## [[2]]
## # A tibble: 2 × 3
##   Sexe   `Median Income` `Average Income`
##   <chr>            <dbl>            <dbl>
## 1 Female           28000           37804.
## 2 Male             33000           54318.
##
## [[3]]
## # A tibble: 30 × 3
##    NOC16 `Median Income` `Average Income`
##    <dbl>           <dbl>            <dbl>
##  1     1           89000          155031.
##  2     2           81000          102652.
##  3     3           32000           41701.
##  4     4           40500           68601.
##  5     5           64000           90310.
##  6     6           42000           48804.
##  7     7           37000           42341.
##  8     8           32000           34983.
##  9     9           76000           96952.
## 10    10           53000           57380.
## # … with 20 more rows
##
## [[4]]
## # A tibble: 11 × 3
##       PR `Median Income` `Average Income`
##    <dbl>           <dbl>            <dbl>
##  1    10           24000           40147.
##  2    11           26000           27817.
##  3    12           29000           43598.
##  4    13           25000           32243.
##  5    24           28000           38471.
##  6    35           28000           46660.
##  7    46           30000           38943.
##  8    47           38000           48454.
##  9    48           36000           60451.
## 10    59           34000           48908.
## 11    70           34500           55526.
##
```

```
## [[5]]
## # A tibble: 13 × 3
##    VisMin `Median Income` `Average Income`
##     <dbl>           <dbl>            <dbl>
##  1      1           26000           42207.
##  2      2           27000           44570.
##  3      3           22000           27665.
##  4      4           28000           31794.
##  5      5           23000           30526.
##  6      6           24000           43845.
##  7      7           26000           40002.
##  8      8           24000           41779.
##  9      9           23000           36942.
## 10     10           35000           50679.
## 11     11           23000           32088.
## 12     12           24500           35578.
## 13     13           31000           47218.
```

4. Multivariate analyses: run GLM with modality as the outcome variable and total income as the predictor variable. First run a model with no controls, then run a model with all of the controls (all of the variables in the data that were selected in step 1).

```
model1 <-
  glm(as.factor(MODE) ~ TotInc, #OUtcome variable is a dummy between active
transport and non-active transport. Outcome variable is total income
    data = census,
              family = binomial(link = logit))



#Add controls of sex, province, occupation, distance of commute, visible
minority status, and the province of work
model2 <- glm(as.factor(MODE) ~ TotInc +
              as.factor(Sex) +
              as.factor(PR) +
              as.factor(NOC16) +
              as.factor(DIST) +
              as.factor(VisMin) +
              as.factor(PWPR),
    data = census,
              family = binomial(link = logit))

screenreg(list(model1, model2),
          custom.coef.map = list("TotInc" = "Income"),
          custom.gof.rows = list("Controls" = c("No", "Yes")))
```

```
##
## =========================================
##                  Model 1        Model 2
## -----------------------------------------
## Income               -0.00 ***      -0.00
##                      (0.00)         (0.00)
## -----------------------------------------
## Controls             No            Yes
## AIC               30251.69       26353.10
## BIC               30268.33       26935.58
## Log Likelihood  -15123.85      -13106.55
## Deviance          30247.69       26213.10
## Num. obs.         30369          30369
## =========================================
## *** p < 0.001; ** p < 0.01; * p < 0.05
```