Awesome Paper

true true true true true true

2021-03-06

Abstract

This is the abstract.
It consists of two paragraphs.

0.1 Setup

0.1.1 Install and Attach

`installr` provides `require2`, this will install a package if it is missing and library it. Unfortunately, `intall` is a package too, so you cannot use `require2` on it.

```
if(!require(installr))install.packages("installr")
library(installr)

## https://rstudio.github.io/distill/tables.html

# Provides support for HTML tables in Rmarkown
require2(rmarkdown)
require2(kableExtra)

# Allows animations and intractable HTML plots
require2(plotly)

require2(Rmisc)
require2(devtools)
require2(xtable)
require2(printr)
require2(stargazer)
require2(DT)
require2(xfun)
require2(psych)
require2(lmtest)
require2(sandwich)
require2(huxtable)
require2(jtools)
require2(tutorial)
require2(car)
require2(olsrr)
require2(broom)
require2(multcomp)
require2(zoo)
```

```r
require2(sandwich)
require2(dynlm)
require2(orcutt)
require2(pdfetch)
require2(rticles)


# Contains ggplot2, dpyr, and much more
require2(tidyverse)

# Replaces `paste`
require2(glue)

# Import data from files
require2(readr)
require2(readxl)

# Mange dates
# require2(lubridate)

# Download files
require2(curl)

# Download and import EPI
require2(epidata)

# Adjust currency values for inflation
require2(priceR)
```

0.1.2 Set up the Knitted table

This will automatically detect if the document is being knited and apply the provided table formatting function or `rmarkdown::paged_table` if not provided. If `nhead` or `ntail` it will call the `head` or `tail` function respectively and limit the data. On `0`, it will ignore it. The default is to create a paginated table on overflow so all the data is accessible but does not take the entire screen.

```r
kblstyle=function(data){
    kableExtra::kable_styling(kableExtra::kbl(data))
}

innerDisp=function(tbl, style){
    ## If the code is kniting
    if(isTRUE(getOption('knitr.in.progress'))){
        return(style(tbl))
    }
    ## Otherwise just return the raw tibble to be formatted by RStudio
    return(tbl)
}

disp=function(tbl, nhead=10, ntail=0, style=FALSE, styleHTML=paged_table, stylePDF=kblstyle){
    if(nhead!=0)tbl=head(tbl, n=nhead)
    if(ntail!=0)tbl=tail(tbl, n=ntail)
```

```
    if(is.function(style))
        return(innerDisp(tbl, style))
    if(knitr::is_html_output())
      return(innerDisp(tbl, styleHTML))
    return(innerDisp(tbl, stylePDF))
}
```

```
disp(mtcars)
mtcars%>%disp()
# Only output the first 20
mtcars%>%disp(nhead = 20)
# Only output the last 10
mtcars%>%disp(ntail = 10)
# Override to use the standard kbl function
mtcars%>%disp(style = function(t){
    kbl(t)%>%
            style()
})
```

```
ggdisp=function(gg){
    if(
        isTRUE(getOption('knitr.in.progress'))
        &&
        !knitr::is_html_output()
    )   return(gg)
    ggplotly(gg)
}
```

```
g=mtcars%>%
    ggplot(aes(mpg, disp))+
    geom_point()
ggdisp(g)
```

1 Introduction

2 Literature Review

(**Keating2019?**)

Here are two sample references: (Pais, 2011; **bob?**).

3 Theoretical Analysis

$$y = \beta_0 + \beta_1 x$$

4 Empirical Analysis

4.1 Data

4.2 Import Data

4.2.1 About the Data

We have two sources of data, one from U.S. Bureau of Labor Statistics (BLS) and the majority of data from Economic Policy Institute (EPI).

BLS maintains a data set called `cpsaat`, this data summaries the wage earnings per type of job, based on race and gender. To access the data in R we use a `curl_download` to retrieve the `.xlsx` file off the internet. To read the file we use the function `readxl::read_excel`.

EPI hosts a lot of data on wage statistics including, minimum wage, the participation, and earnings of each race, gender, education level, and much more. Due to the way EPI presents the data, it cannot be downloaded with `curl`. Instead, I have accessed the data with the package `epidata`, this simple package interfaces with EPI so that you don't have to manually download the data. EPI does not contain individual observations for wage, instead it provides 2 summarizations of the data grouped by race, age, gender, and education. This is the median, 50% of people make more and 50% of people make less than this value. The other one is mean, or they call average, this is the sum of wages added up and divided by the amount.

$$\bar{x} = \frac{\sum_{i=0}^{n-1} x_i}{n}$$

To reduce the effect of the highest earners we will be using the median, like they use in the housing market as a high outlier will only add one rather than a lot more.

4.2.2 Import cpsaat Data

Make sure we have internet and if not abort if not

```
if(!curl::has_internet())quit()
```

`cpsaat` data is provided online at bls.gov. As it is a direct link we can download it and save it to a temporary file and process the data with `readxl::read_excel()`

```
## Create a temp file name/location
tmp <- tempfile()
## Download cpsaat data
curl_download("https://www.bls.gov/cps/cpsaat11.xlsx", destfile = tmp)

## Import cpsaat
cpsaat11 <- read_excel(
        tmp,
        col_names = c(
            "Occupation",
            "Total",
            "Women",
            "White",
            "Black/African American",
            "Asian",
            "Hispanic/Latino"
        ),
        na = "-",
```

```
        col_types = c(
            Occupation="text",
            Total="numeric",
            "Women"="numeric",
            "White"="numeric",
            "Black/African American"="numeric",
            "Asian"="numeric",
            "Hispanic/Latino"="numeric"
        ),
        skip = 7
    )%>%
    drop_na(Occupation)
## Remove temp file and var
file.remove(tmp)
```

```
[1] TRUE
```

```
rm(tmp)
```

4.2.3 Import EPI Data

Get the data at EPI. As there is no direct link avalable we cannot use `curl`, instead there is a package that we can use to access the data, `epidata`. This will download data in the background.

```
Labor_force_participation <- epidata::get_labor_force_participation_rate(by = "gr")

Medianaverage_hourly_wages <- epidata::get_median_and_mean_wages(by = "gr")

Minimum_wage <- epidata::get_minimum_wage()
```

4.3 Clean Data

As with most data, it will have to be cleaned. This includes pivoting the tibble into a longer tibble, as it will work better for `ggplot2`. This current format is called wide format as it has many columns. To fix this we can convert it into long format, as there are many rows, with `pivot_longer`. When we do this sometimes the new column we create contains more than one value, to remedy this issue we can use `seperate` and mutate if necessary to get the values in the right column. Another inconsistancy we should be aware of is that the currency values are in different years, not a large difference, but something that should be corrected.

4.3.1 Clean cpsaat11

```
cpsaat11%>%disp()
```

Occupation	Total	Women	White	Black/African American	Asi
Total, 16 years and over	147795	46.8	78.0	12.1	6
Management, professional, and related occupations	63644	51.7	78.7	9.7	8
Management, business, and financial operations occupations	27143	44.6	81.7	8.8	6
Management occupations	18564	40.4	83.4	8.0	5
Chief executives	1669	29.3	88.0	4.3	5
General and operations managers	1057	30.5	84.4	7.1	4
Legislators	25	NA	NA	NA	N
Advertising and promotions managers	56	52.1	80.5	14.7	3
Marketing managers	554	60.7	84.1	5.5	7
Sales managers	521	30.9	87.6	5.8	4

```
cpsaat11=cpsaat11%>%
    pivot_longer(-c(Occupation, Total), names_to = "Race", values_to = "Percentage")
```

Looks fine.

4.3.2 Clean Labor_force_participation

```
Labor_force_participation%>%disp()
```

date	all	women	men	black	black_women	black_men	hispanic	hispanic_women	hispanic_men
1978-01-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-02-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-03-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-04-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-05-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-06-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-07-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-08-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-09-01	NA	NA	NA	NA	NA	NA	NA	NA	NA
1978-10-01	NA	NA	NA	NA	NA	NA	NA	NA	NA

```
Participation=Labor_force_participation%>%
    pivot_longer(-date, names_to = "Race", values_to = "Participation", values_drop_na = T)%>%
    separate(Race, into = c("Race", "Gender"))
```

```
Warning: Expected 2 pieces. Missing pieces filled with ‘NA‘ in 3036 rows [1, 2,
3, 4, 7, 10, 13, 14, 15, 16, 19, 22, 25, 26, 27, 28, 31, 34, 37, 38, ...].
```

```
Participation=Participation%>%
    filter(grepl("women|men", Race, ignore.case = T))%>%
    mutate(
        Gender=Race,
        Race=NA_character_
    )%>%
    union(
        Participation%>%
```

```
            filter(!grepl("women|men", Race, ignore.case = T))
    )
Participation%>%
    filter(!is.na(Race))%>%
    disp()
```

date	Race	Gender	Participation
1978-12-01	all	NA	0.634
1978-12-01	black	NA	0.617
1978-12-01	black	women	0.535
1978-12-01	black	men	0.718
1978-12-01	hispanic	NA	0.633
1978-12-01	hispanic	women	0.470
1978-12-01	hispanic	men	0.812
1978-12-01	white	NA	0.635
1978-12-01	white	women	0.499
1978-12-01	white	men	0.785

```
rm(Labor_force_participation)
```

4.3.3 Clean Medianaverage__hourly__wages

```
Medianaverage_hourly_wages%>%disp()
```

date	median	average	men__median	men__average	women__median	women__average	white__median	white__a
1973	17.27	20.09	20.97	23.55	13.19	15.13	17.94	
1974	16.93	19.72	20.71	23.13	13.00	14.86	17.51	
1975	16.94	19.77	21.04	23.12	13.18	15.06	17.44	
1976	16.90	19.99	20.70	23.35	13.30	15.42	17.53	
1977	16.92	19.88	20.90	23.36	13.20	15.24	17.48	
1978	17.07	19.92	21.20	23.47	13.25	15.29	17.71	
1979	16.79	20.10	21.07	23.66	13.35	15.49	17.45	
1980	16.68	19.70	20.87	23.19	13.29	15.33	17.39	
1981	16.50	19.59	20.43	23.05	13.36	15.33	16.98	
1982	16.43	19.76	20.44	23.24	13.21	15.59	17.25	

```
Wages=Medianaverage_hourly_wages%>%
    pivot_longer(-date, names_to = "Race", values_to = "Wage", values_drop_na = T)%>%
    separate(Race, into = c("Race", "Gender", "Summary"), fill = "left")

## Race is in the wrong location sometimes
Wages=Wages%>%
    filter(!grepl("women|men", Gender, ignore.case = T))%>%
    mutate(
        Race=Gender,
        Gender=NA_character_
    )%>%
    union(
```

```
        Wages%>%
            filter(grepl("women|men", Gender, ignore.case = T))
    )
## No need to keep the Average and Median split up
Wages=Wages%>%
    pivot_wider(names_from = Summary, values_from = Wage)
rm(Medianaverage_hourly_wages)
```

4.3.4 Clean Minimum_wage

This data has data in terms of 2018, the other data is in 2019 USD. As it will be easiest and the latest data, we will be using 2019. Although small, there will be a difference and we need to adjust for inflation. The package `priceR` allows us to convert those monetary values into other ones using online inflation data.

```
Minimum_wage%>%disp()
```

date	federal_minimum_wage_nominal_dollars	federal_minimum_wage_real_x_2018_dollars	average_wages_of_
2018	7.25	7.25	
2017	7.25	7.43	
2016	7.25	7.59	
2015	7.25	7.68	
2014	7.25	7.70	
2013	7.25	7.83	
2012	7.25	7.94	
2011	7.25	8.11	
2010	7.25	8.37	
2009	7.25	8.51	

```
##adjust for inflation to get to common 2019
Minimum_wage=Minimum_wage%>%
    mutate(
        Min2019=priceR::adjust_for_inflation(
            federal_minimum_wage_real_x_2018_dollars,
            2018,
            "US",
            2019
        )
    )
```

```
Generating URL to request all 297 results
Retrieving inflation data for US
Generating URL to request all 61 results
```

```
Minimum_wage=Minimum_wage%>%
    rename(MinCur=federal_minimum_wage_nominal_dollars)%>%
    select(Min2019, MinCur, date)
```

4.3.5 Fix inconsistant case

As the data was imported with `epidata`, the colum names have been changed from what the csv has. So we need to fix that to conform to consistency.

```r
Wages=Wages%>%
    rename(
        Date=date,
        Median=median,
        Average=average
    )

Participation=Participation%>%
    rename(Date=date)

Minimum_wage=Minimum_wage%>%
    rename(Date=date)
```

4.4 Export the data as .csv files

To backup our data we will export the cleaned tibbles.

```r
if(!dir.exists("../data"))dir.create("../data")

cpsaat11%>%
    write_csv("../data/cpsaat11.csv")

Minimum_wage%>%
    write_csv("../data/Minimum_wage.csv")

Participation%>%
    write_csv("../data/Participation.csv")

Wages%>%
    write_csv("../data/Wages.csv")
```

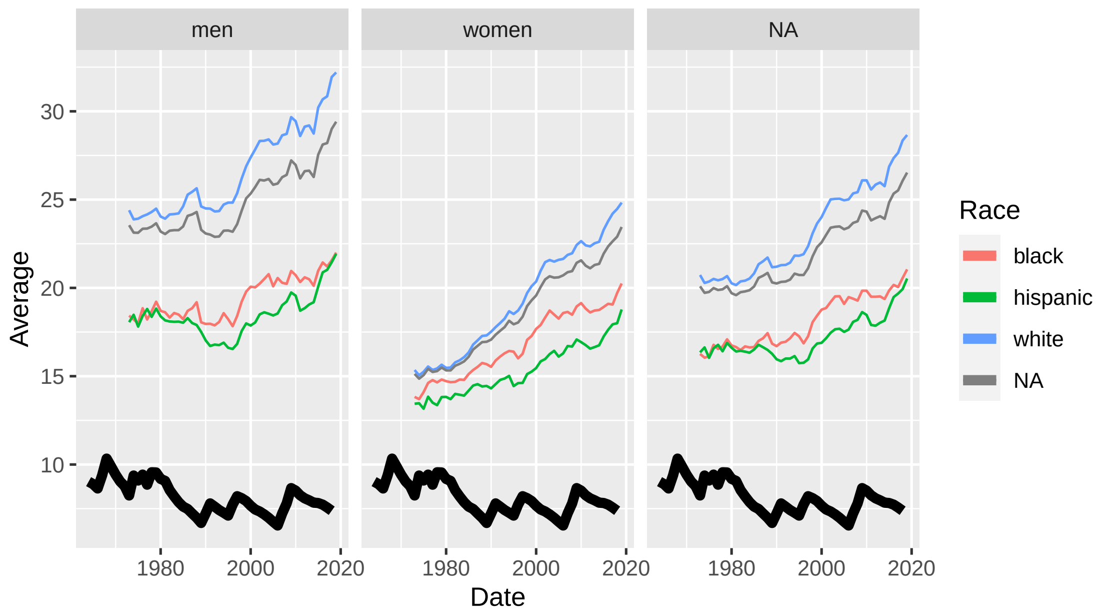
4.5 Methodology

4.6 Results

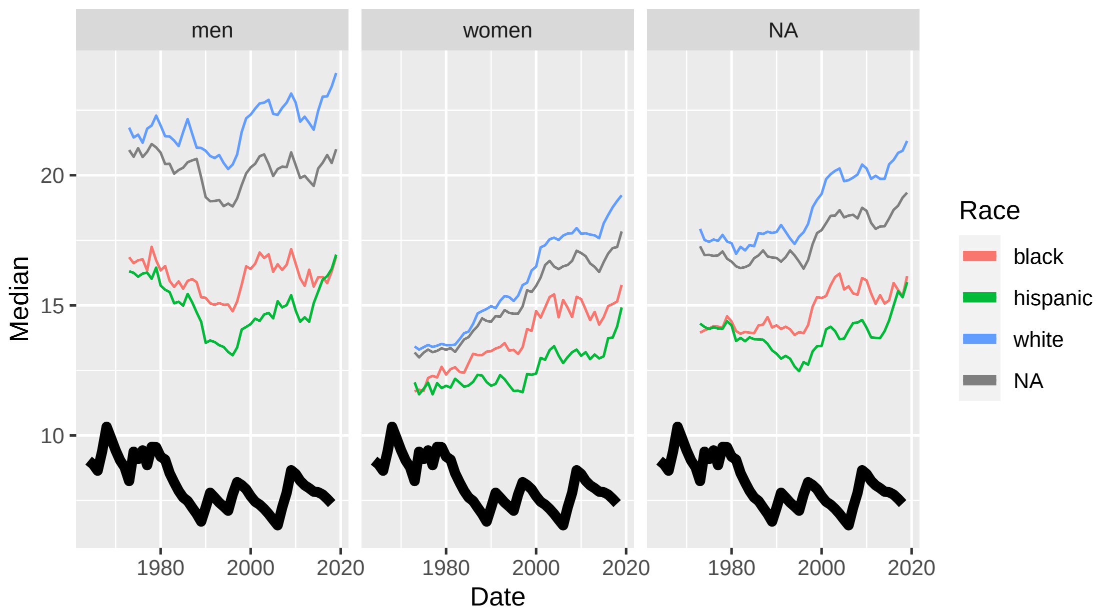
4.7 Wage over Time by Race and Gender

4.7.1 Average and Medium Wage over Time by Race and Gender

```r
g=Wages%>%
    ggplot(aes(col=Race, x=Date))+
    geom_line(aes(y=Average))+
    geom_line(aes(y=Min2019, col=NULL), data=Minimum_wage, size=2)+
    facet_wrap(~Gender)
ggdisp(g)
```




```
g=Wages%>%
    ggplot(aes(col=Race, x=Date))+
    geom_line(aes(y=Median))+
    geom_line(aes(y=Min2019, col=NULL), data=Minimum_wage, size=2)+
    facet_wrap(~Gender)
ggdisp(g)
```



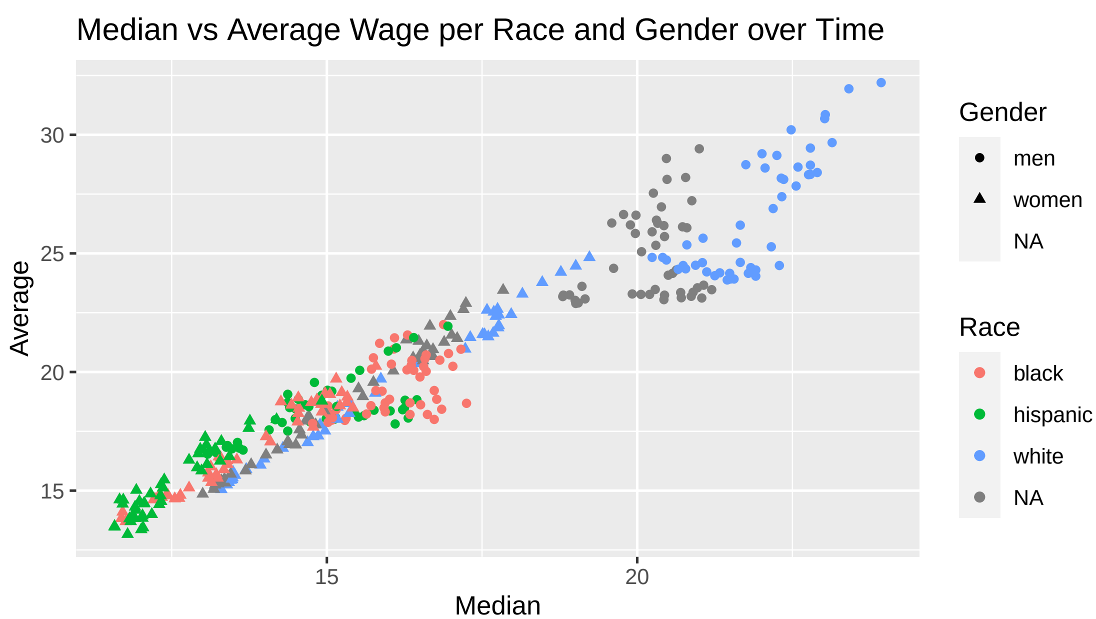
4.7.2 Scatter Plot over Time

```
g=Wages%>%
    ggplot()+
    geom_point(aes(x=Median, y=Average, col=Race, shape=Gender, frame=Date))+
    ggtitle("Median vs Average Wage per Race and Gender over Time")
```

```
Warning: Ignoring unknown aesthetics: frame
```

```
ggdisp(g)
```

```
Warning: Removed 188 rows containing missing values (geom_point).
```



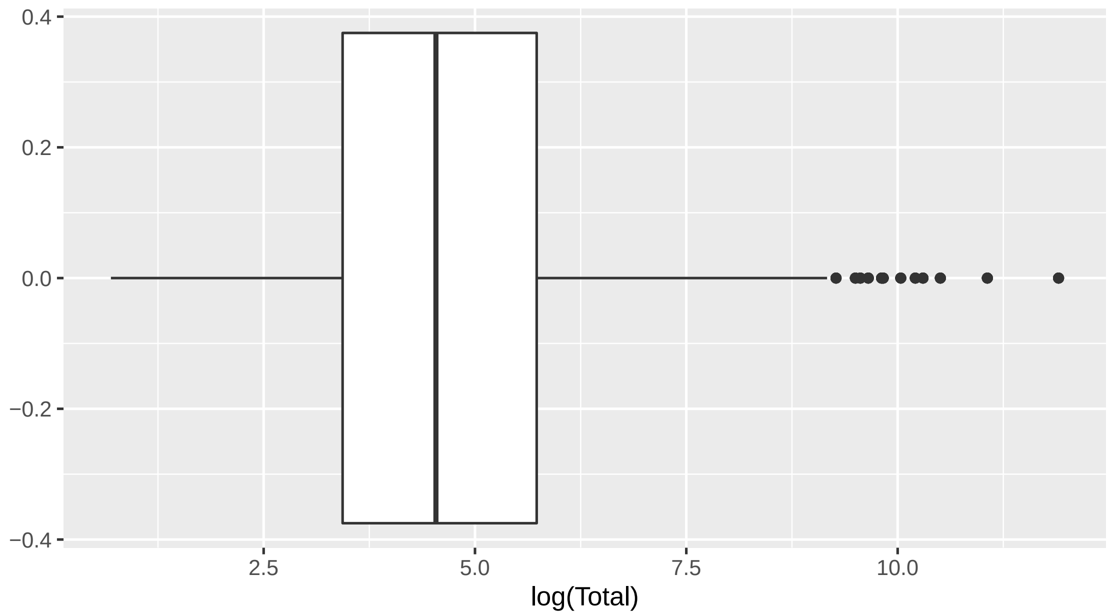
4.8 Wages according to Jobs

4.8.1 Sumarise data according to income of jobs

This data is currently unusable as there is only one opservation per type of job, we don't have over time statistics. We do however, have a snapshot of the diverse earnings, we don't care what the job is, but the average wage of each race per earning bracket.

```
cpsaat11%>%
    ggplot(aes(x=log(Total)))+
    geom_boxplot()
```

```
Warning: Removed 10 rows containing non-finite values (stat_boxplot).
```




```r
# Generate the percentiles
se=quantile(log(cpsaat11$Total), seq(0, 1, by=.1), na.rm=T)

# Add outlyers
se["200%"]=Inf

# break into groups and drop NAs
d=cpsaat11%>%
    drop_na(Percentage)%>%
    group_by(gr=cut(Total, breaks=exp(se)), Race)

# Summarize the data and remove women as it is not a race
# This is so it add up to 100% or so
d=d%>%
    summarise(Percentage=mean(Percentage), Total=mean(Total))%>%
    filter(Race!="Women")
d
```

```
# A tibble: 32 x 4
# Groups:   gr [8]
   gr       Race                   Percentage Total
   <fct>    <chr>                       <dbl> <dbl>
 1 (40,60]  Asian                        3.72  53.7
 2 (40,60]  Black/African American      10.9   53.7
 3 (40,60]  Hispanic/Latino             16.9   53.7
 4 (40,60]  White                       82.1   53.7
 5 (60,93]  Asian                        8.60  74.6
 6 (60,93]  Black/African American      13.1   74.6
 7 (60,93]  Hispanic/Latino             14.1   74.6
 8 (60,93]  White                       74.6   74.6
 9 (93,131] Asian                        5.88 110.
10 (93,131] Black/African American      11.9  110.
# ... with 22 more rows
```

4.8.2 Is there missing data

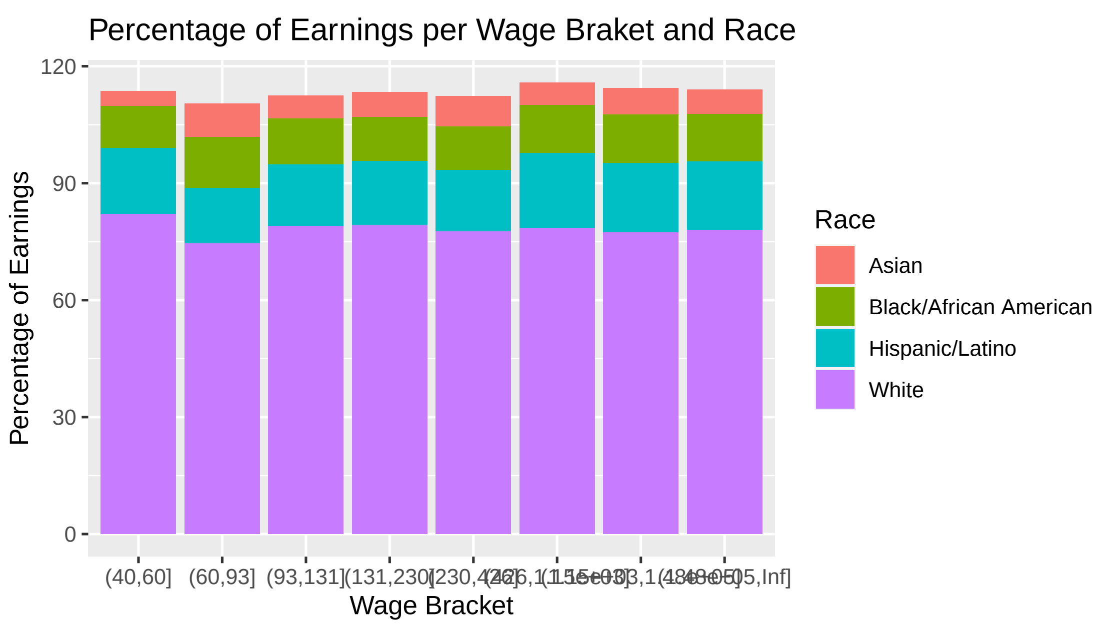
```
cpsaat11%>%
    drop_na(Percentage)%>%
    filter(Total<30)
```

Occupation	Total	Race	Percentage

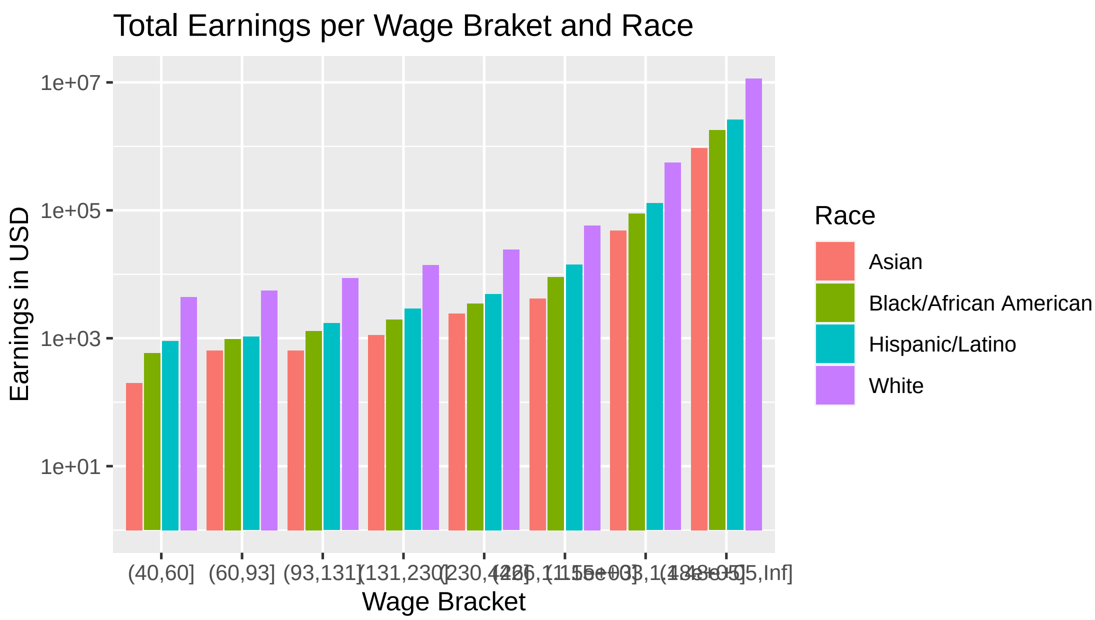
No, we just have a lack of observations for poor paying jobs.

4.8.3 Graph

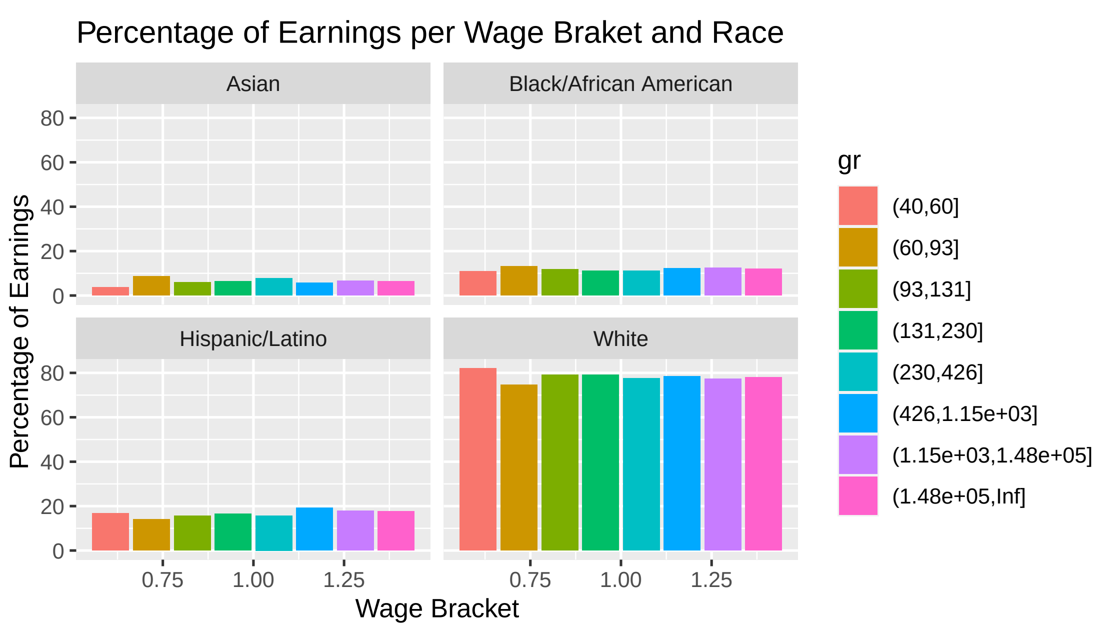
```
g=d%>%
    ggplot(aes(fill=Race, y=Percentage, x=gr))+
    geom_col()+
    xlab("Wage Bracket")+
    ylab("Percentage of Earnings")+
    ggtitle("Percentage of Earnings per Wage Braket and Race")
ggdisp(g)
```




```
g=d%>%
    ggplot(aes(fill=Race, y=Percentage*Total, x=gr))+
    geom_col(position = "dodge2")+
    scale_y_log10()+
    xlab("Wage Bracket")+
    ylab("Earnings in USD")+
    ggtitle("Total Earnings per Wage Braket and Race")
ggdisp(g)
```

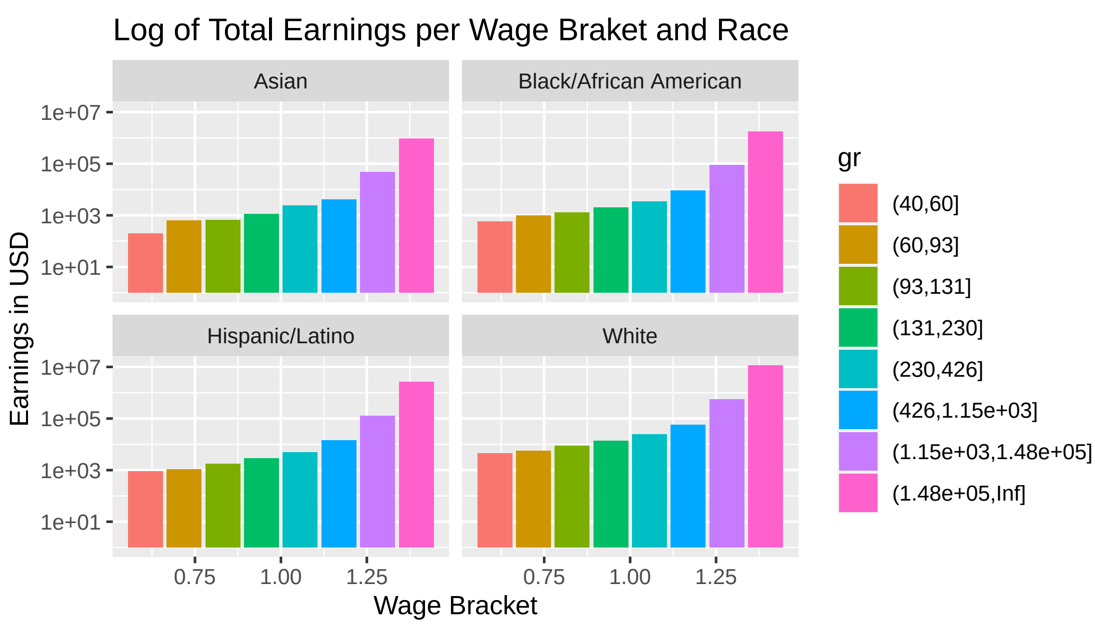



```
g=d%>%
    ggplot(aes(fill=gr, x=1, y=Percentage))+
    geom_col(position = "dodge2")+
    facet_wrap(~Race)+
    xlab("Wage Bracket")+
    ylab("Percentage of Earnings")+
    ggtitle("Percentage of Earnings per Wage Braket and Race")
ggdisp(g)
```

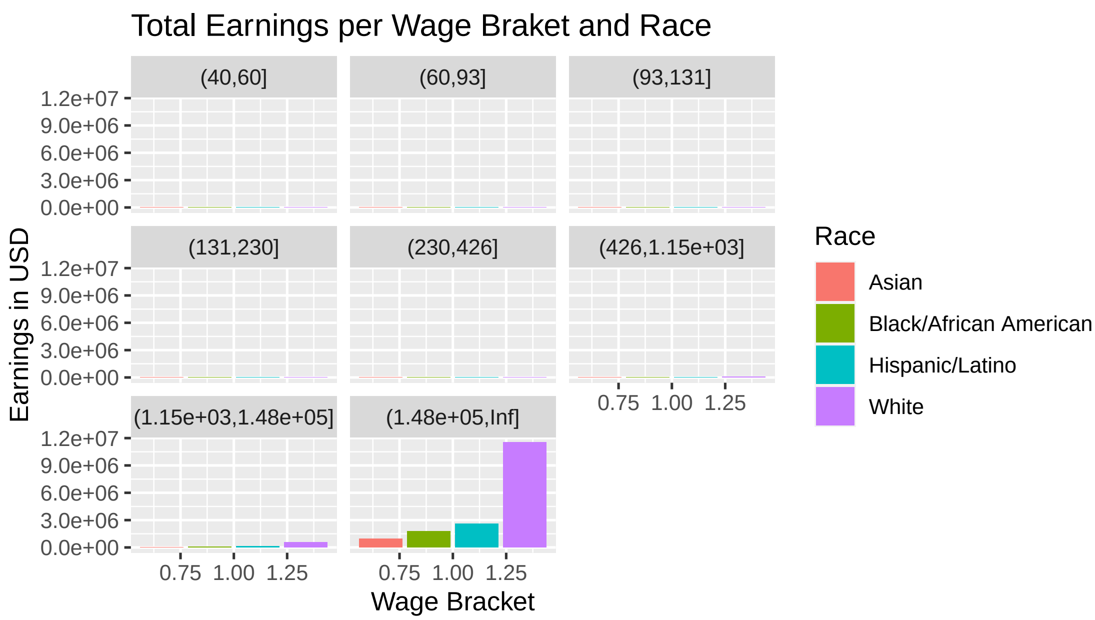



```
g=d%>%
    ggplot(aes(fill=gr, x=1, y=Percentage*Total))+
```

```
    geom_col(position = "dodge2")+
    facet_wrap(~Race)+
    scale_y_log10()+
    xlab("Wage Bracket")+
    ylab("Earnings in USD")+
    ggtitle("Log of Total Earnings per Wage Braket and Race")
ggdisp(g)
```




```
g=d%>%
    ggplot(aes(fill=Race, x=1, y=Percentage*Total))+
    geom_col(position = "dodge2")+
    facet_wrap(~gr)+
    xlab("Wage Bracket")+
    ylab("Earnings in USD")+
    ggtitle("Total Earnings per Wage Braket and Race")
ggdisp(g)
```



5 Conclusion

6 References

Pais, J., 2011. Socioeconomic background and racial earnings inequality: A propensity score analysis. Social Science Research 40, 37–49. doi:10.1016/j.ssresearch.2010.06.016