ia530_finalProject

Michael Gilbert/Amy Moyer

3/29/2021

1 Import Data

Import Data and set up main theme for graphs

```
final_data <- read_csv(here('Data','final_data.csv'))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   month_end = col_date(format = ""),
##   ment_health_no_good_18_24 = col_double(),
##   genhlth_no_good_18_24 = col_double(),
##   ment_health_no_good_25on = col_double(),
##   genhlth_no_good_25on = col_double(),
##   divorced_widowed_separated = col_double(),
##   avg_close_apple_stock = col_double(),
##   avg_close_atandt_stock = col_double(),
##   avg_close_verizon_stock = col_double(),
##   x0_24_suicides = col_double(),
##   x25on_suicides = col_double(),
##   population_num_million = col_double(),
##   unemp_per = col_double(),
##   personal_save_rate = col_double(),
##   x0_24_suicides_per_thous = col_double(),
##   x25on_suicides_per_thous = col_double()
## )
```

```
# contains both variable and pretty names
all_names_cleaned <- read_csv(here('Data','all_names_cleaned.csv'))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   all_names = col_character(),
##   clean_name = col_character()
## )
```

```
main_theme <- theme(panel.grid = element_blank(),
                    panel.background = element_blank(),
                    plot.title = element_text(size=10),
                    plot.subtitle = element_text(size=8),
                    axis.title.x = element_text(size=8),
                    axis.title.y = element_text(size=8),
                    axis.text.x = element_text(size=8),
                    axis.text.y = element_text(size=8))

# filter out unneeded variables
final_data_2 <- final_data %>%
  dplyr::select(-c(population_num_million, x0_24_suicides, x25on_suicides))
```

2 Describe Data

Below are the descriptions of the data used to try and understand suicide rates for adolescents (0-24) and adults (25+)

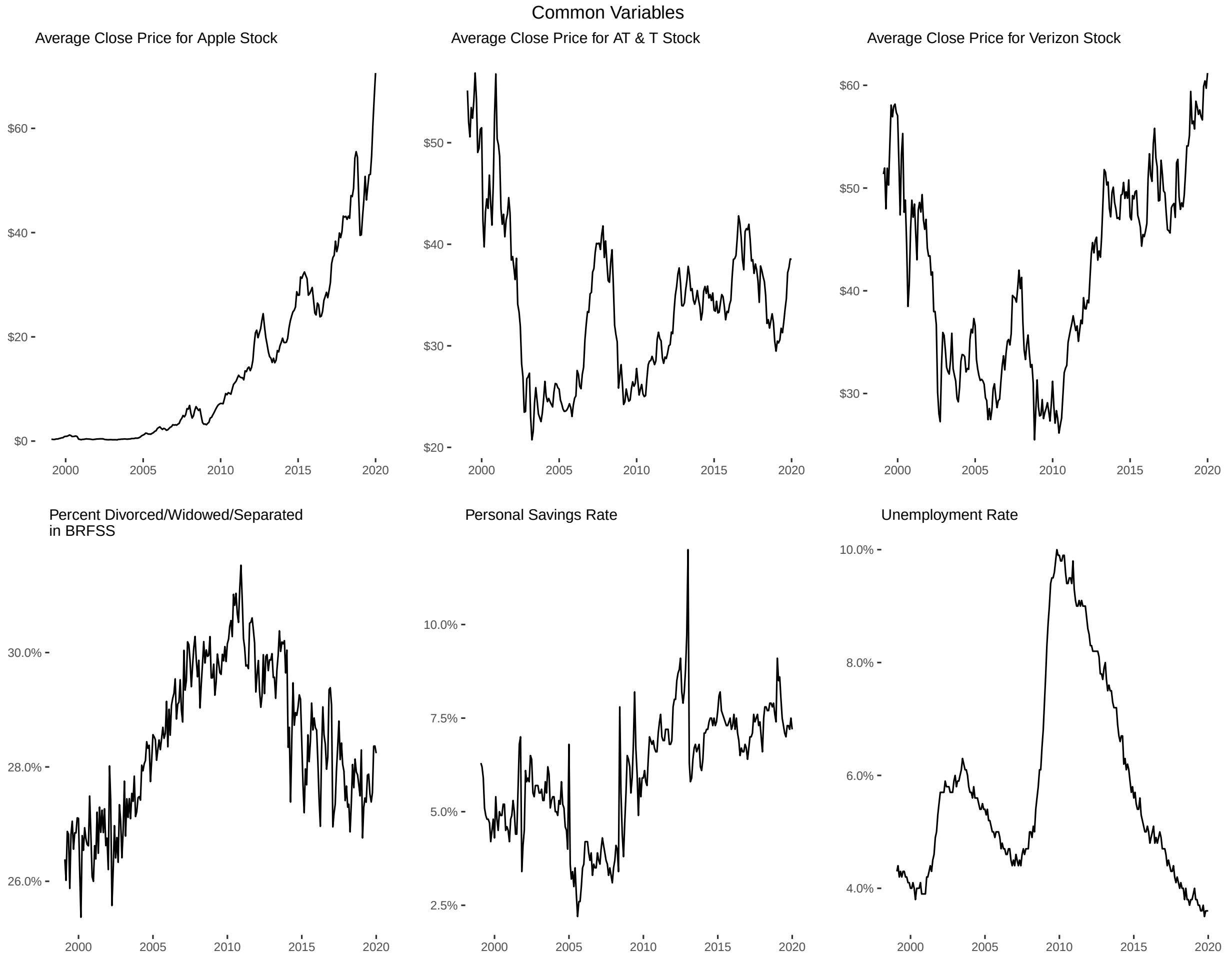
- Month End (month_end): the last day of the month
- Avg Close Apple Stock (avg_close_apple_stock): Avg close price for the month for apple. Proxy variable for social media/internet usage
- Avg Close AT&T Stock (avg_close_atandt_stock): Avg close price for the month for AT&T. Proxy variable for social media/internet usage
- Avg Close Verizon Stock (avg_close_verizon_stock): Avg close price for the month for verizon. Proxy variable for social media/internet usage
- Divorced/Widowed/Separated (divorced_widowed_separated): This comes from the Behavioral Risk Factor Surveillance System (BRFSS) from the CDC. This is the percentage of respondents between that responded Divorced, Widowed or Separated to the following question "Are you: (marital status)". Other responses included Married, Never Married, A member of an unmarried couple or refused.
- 24 and Under General Health Not Good (genhlth_no_good_18_24): This comes from the Behavioral Risk Factor Surveillance System (BRFSS) from the CDC. This is the percentage of respondents between the age 18 and 24 that responded Poor to the following question "Would you say that in general your health is:". Other responses included Excellent, Very Good, Good, Fair and Don't Know/Not Sure, Refused.
- 25 and Over General Health Not Good (genhlth_no_good_25on): Same as above except for age 25+
- 24 and Under Mental Health Not Good (ment_health_no_good_18_24): This comes from the Behavioral Risk Factor Surveillance System (BRFSS) from the CDC. This is the percentage of respondents between the age 18 and 24 that responded 1 or more (up to 30) to the following question: "Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days was your mental health not good?"
- 25 and Over Mental Health Not Good (ment_health_no_good_25on): Same as above except for age 25+
- Personal Savings Rate (personal_save_rate): personal savings rate from St Louis Fed
- Unemployment Rate (unemp_per): unemployment from department of labor
- 24 and Under Suicides Per Thousand (x0_24_suicides_per_thous): number of suicides per thousand people in the US between 0 and 24 from CDC
- 25 and Over Suicides Per Thousand (x25on_suicides_per_thous): number of suicides per thousand people in the US 25 and on from CDC

3 Graph Common Variables

For this project we will create two models. One for adolescents and one for adults. A number of variables from above will be used in both models, while variables that can be split by age, will be separated for each model. Below is a graph of the variables in common between the two models.

```
##### Common Variables

temp_list <- list()

# Apple Stock Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=avg_close_apple_stock)) +
  geom_line() +
  labs(title=’Average Close Price for Apple Stock’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = dollar_format())

temp_list <- append(temp_list, list(temp_plot))

# AT & T Stock Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=avg_close_atandt_stock)) +
  geom_line() +
  labs(title=’Average Close Price for AT & T Stock’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = dollar_format())

temp_list <- append(temp_list, list(temp_plot))

# Verizon Stock Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=avg_close_verizon_stock)) +
  geom_line() +
  labs(title=’Average Close Price for Verizon Stock’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = dollar_format())

temp_list <- append(temp_list, list(temp_plot))

# Divorced/Widowed/Separated Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=divorced_widowed_separated)) +
  geom_line() +
  labs(title=’Percent Divorced/Widowed/Separated \nin BRFSS’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Personal Savings Rate Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=personal_save_rate)) +
  geom_line() +
  labs(title=’Personal Savings Rate’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))
```

```
# Unemployment Rate Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=unemp_per)) +
  geom_line() +
  labs(title=’Unemployment Rate’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

grid <- grid.arrange(grobs=temp_list, nrow=2,top=’Common Variables’)
```

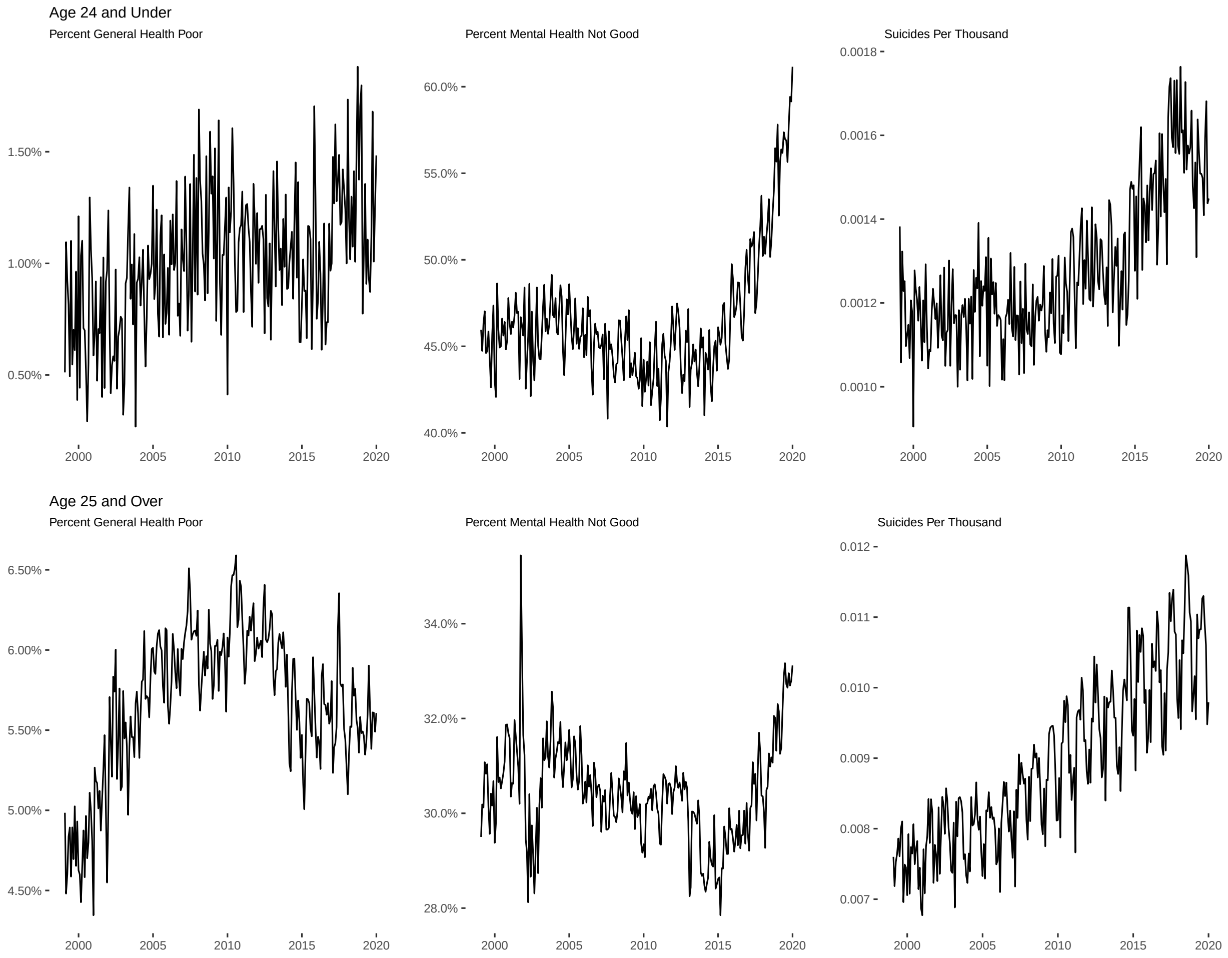


```
ggsave(here(’Plots’, ’common_variables.png’),plot=grid,dpi=600, width = 11.5, height = 9, units=’in’)
```

4 Graph Age Specific Variables

```
temp_list <- list()

# Gen Health No Good
```

```
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=genhlth_no_good_18_24)) +
  geom_line() +
  labs(title=’Age 24 and Under’,subtitle=’Percent General Health Poor’ ,x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Ment Health No Good
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=ment_health_no_good_18_24)) +
  geom_line() +
  labs(title=’’, subtitle=’Percent Mental Health Not Good’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Per 1000 Suicides Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=x0_24_suicides_per_thous)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

# Gen Health No Good
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=genhlth_no_good_25on)) +
  geom_line() +
  labs(title=’Age 25 and Over’,subtitle=’Percent General Health Poor’ ,x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Ment Health No Good
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=ment_health_no_good_25on)) +
  geom_line() +
  labs(title=’’, subtitle=’Percent Mental Health Not Good’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Per 1000 Suicides Graph
temp_plot <- ggplot(final_data_2, aes(x=month_end, y=x25on_suicides_per_thous)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

grid <- grid.arrange(grobs=temp_list, nrow=2)
```



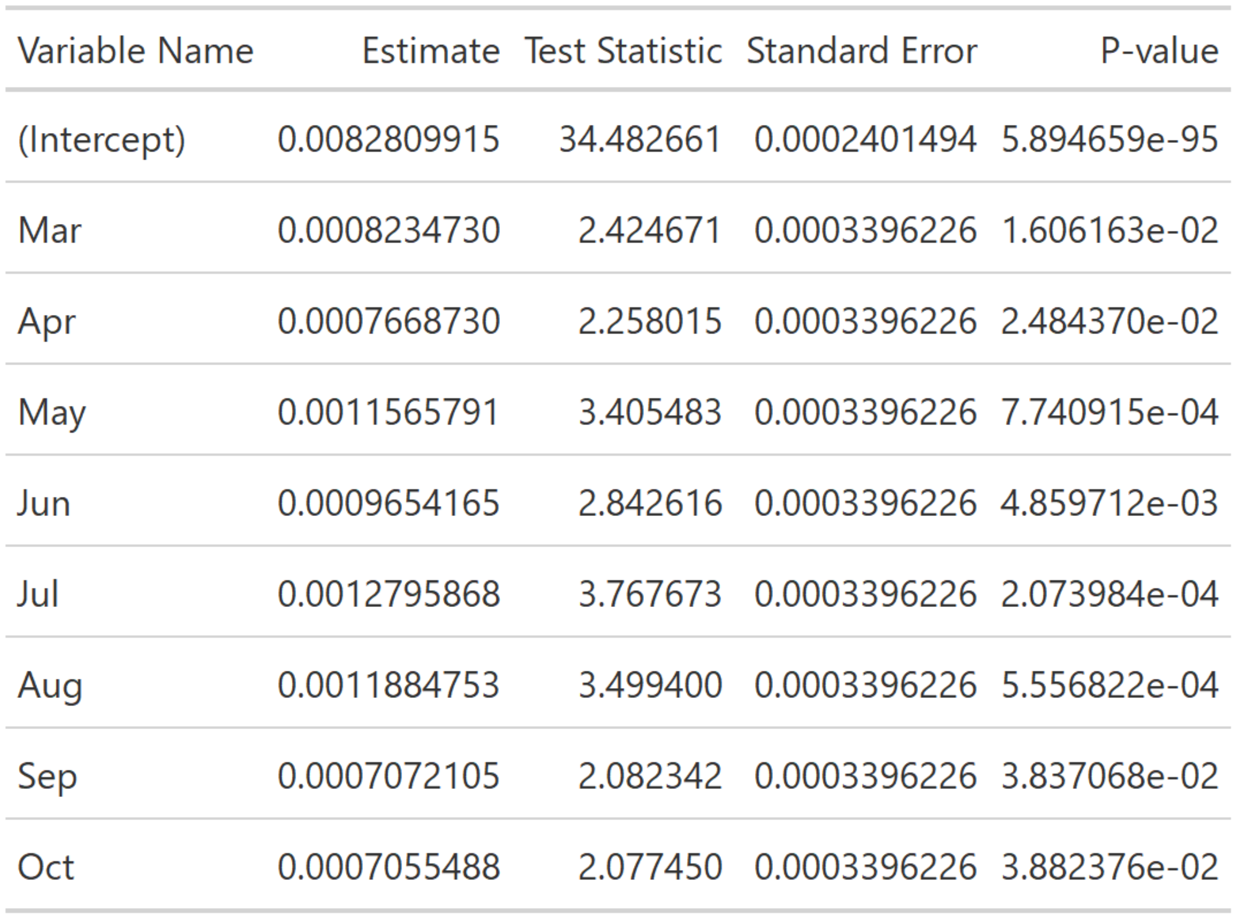
```r
ggsave(here('Plots', 'age_variables.png'),plot=grid,dpi=600, width = 11.5, height = 9, units='in')
```

5 Create Seasonal Dummy Variables and Test for Seasonality

Next we decided to test to see if there was seasonality. We only felt that it made sense to test for seasonality in the mental health and suicide variables. We only provide one sample below which seemed to be the most seasonal (the suicide rate for the 25 and above). All variables were fit, but only those that were significant (p-value below 0.05) are shown in the table

```r
# Create month dummy variables

final_data_2 <- final_data_2 %>%
  mutate(month_num=month(month_end),
         month_name=month.abb[month_num])

final_data_2 <- dummy_cols(final_data_2, select_columns = c('month_name'))

names(final_data_2)[16:27] <- gsub('month_name_','',names(final_data_2)[16:27])

# Convert data to time-series and fit dummy regressions
```

```
ts_final_data <- ts(final_data_2, start=c(1999,1), frequency=12)

x24under_suicide_lm <- lm(x0_24_suicides_per_thous ~
                            Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,
                          data=ts_final_data)

x25over_suicide_lm <- lm(x25on_suicides_per_thous ~
                           Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,
                         data=ts_final_data)

x24under_menthlth_lm <- lm(ment_health_no_good_18_24 ~
                             Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,
                           data=ts_final_data)

x25over_menthlth_lm <- lm(ment_health_no_good_25on ~
                            Jan+Feb+Mar+Apr+May+Jun+Jul+Aug+Sep+Oct+Nov,
                          data=ts_final_data)

x25over_suicide_coef <- coef(summary(x25over_suicide_lm))

x25over_suicide_coef_tibble <- bind_cols(var_name=rownames(x25over_suicide_coef),
                                         as_tibble(x25over_suicide_coef)) %>%
  clean_names()

x25over_suicide_coef_pretty_table <- x25over_suicide_coef_tibble %>%
  filter(pr_t<=0.05) %>%
  rename(’P-value’ = pr_t,
         ’Estimate’=estimate,
         ’Standard Error’=std_error,
         ’Test Statistic’=t_value,
         ’Variable Name’=var_name) %>%
  dplyr::select(’Variable Name’,
                ’Estimate’,
                ’Test Statistic’,
                ’Standard Error’,
                ’P-value’) %>%
  gt()


gt::gtsave(x25over_suicide_coef_pretty_table ,here(’Plots’, ’x25over_suicide_coef_coef.png’))
```



Variable Name	Estimate	Test Statistic	Standard Error	P-value
(Intercept)	0.0082809915	34.482661	0.0002401494	5.894659e-95
Mar	0.0008234730	2.424671	0.0003396226	1.606163e-02
Apr	0.0007668730	2.258015	0.0003396226	2.484370e-02
May	0.0011565791	3.405483	0.0003396226	7.740915e-04
Jun	0.0009654165	2.842616	0.0003396226	4.859712e-03
Jul	0.0012795868	3.767673	0.0003396226	2.073984e-04
Aug	0.0011884753	3.499400	0.0003396226	5.556822e-04
Sep	0.0007072105	2.082342	0.0003396226	3.837068e-02
Oct	0.0007055488	2.077450	0.0003396226	3.882376e-02

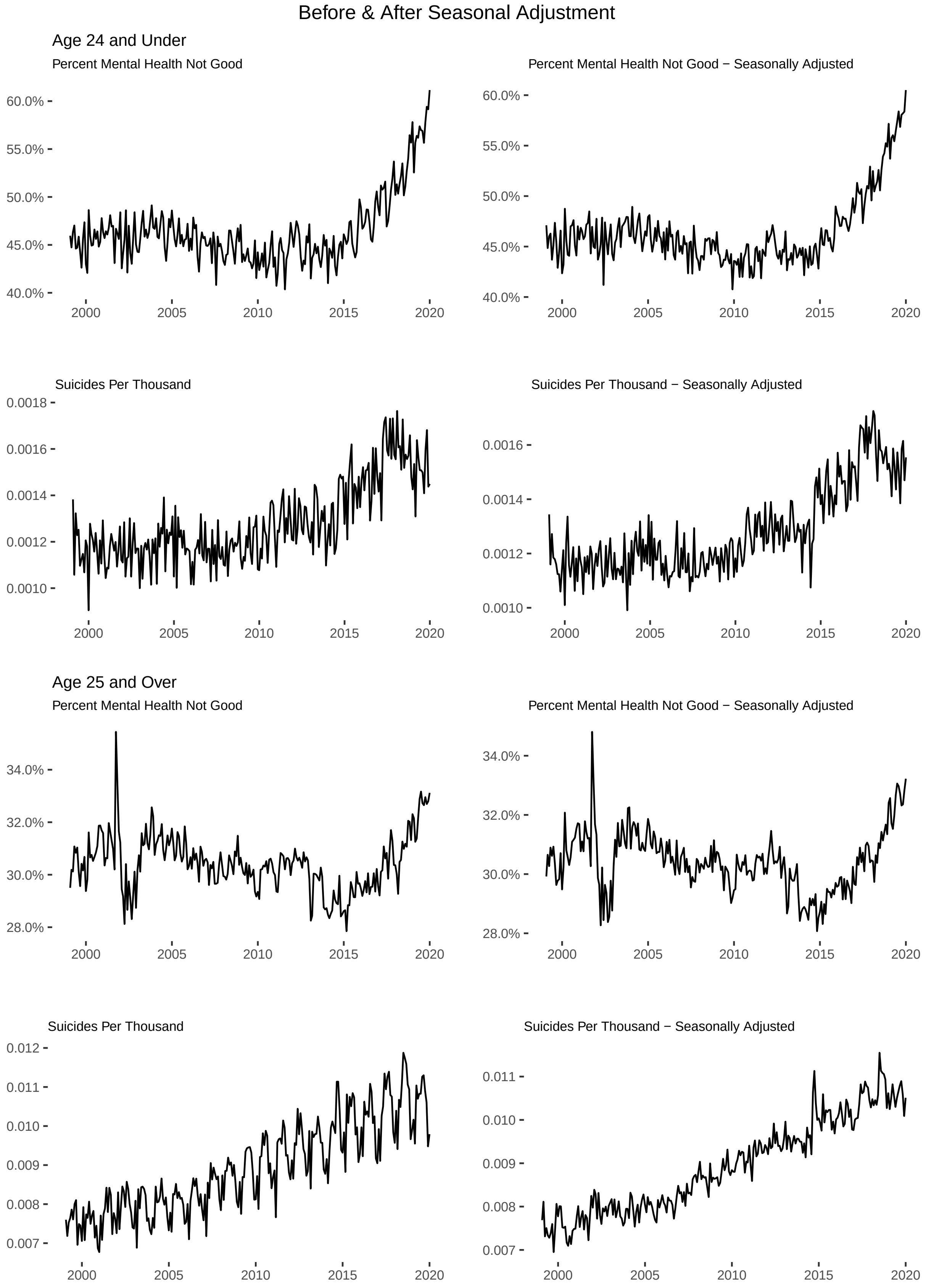
6 Seasonally adjust the four variables that we tested above

```r
# Remove seasonal component from suicide variables

x24under_suicide_decomp <- decompose(ts_final_data[,'x0_24_suicides_per_thous'])
x24under_suicide_sa <- x24under_suicide_decomp$x - x24under_suicide_decomp$seasonal

x25over_suicide_decomp <- decompose(ts_final_data[,'x25on_suicides_per_thous'])
x25over_suicide_sa <- x25over_suicide_decomp$x - x25over_suicide_decomp$seasonal

# Remove seasonal component from mental health variables

x24under_ment_hlth_decomp <- decompose(ts_final_data[,'ment_health_no_good_18_24'])
x24under_ment_hlth_sa <- x24under_ment_hlth_decomp$x - x24under_ment_hlth_decomp$seasonal

x25over_ment_hlth_decomp <- decompose(ts_final_data[,'ment_health_no_good_25on'])
x25over_ment_hlth_sa <- x25over_ment_hlth_decomp$x - x25over_ment_hlth_decomp$seasonal

final_data_3 <- bind_cols(final_data_2,
                             x24under_suicide_sa=x24under_suicide_sa,
                             x25over_suicide_sa=x25over_suicide_sa,
                             x24under_ment_hlth_sa=x24under_ment_hlth_sa,
                             x25over_ment_hlth_sa=x25over_ment_hlth_sa)

ts_final_data <- ts(final_data_3, start=c(1999,1), frequency=12)
```

6b Graph Seasonally Adjusted Variables

```
## Graph After Seasonal
temp_list <- list()


## 24 and Under
# Percent Mental Health Not Good Before
temp_plot <- ggplot(final_data_3, aes(x=month_end, y=ment_health_no_good_18_24)) +
  geom_line() +
  labs(title=’Age 24 and Under’, subtitle=’Percent Mental Health Not Good’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Percent Mental Health Not Good After
temp_plot <- ggplot(final_data_3, aes(x=month_end, y=x24under_ment_hlth_sa)) +
  geom_line() +
  labs(title=’’, subtitle=’Percent Mental Health Not Good - Seasonally Adjusted’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Suicides Per Thousand Before
temp_plot <- ggplot(final_data_3, aes(x=month_end, y=x0_24_suicides_per_thous)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

# Suicides Per Thousand After
temp_plot <- ggplot(final_data_3, aes(x=month_end, y=x24under_suicide_sa)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand - Seasonally Adjusted’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

## 25 and Over
# Percent Mental Health Not Good Before

temp_plot <- ggplot(final_data_3, aes(x=month_end, y=ment_health_no_good_25on)) +
  geom_line() +
  labs(title=’Age 25 and Over’, subtitle=’Percent Mental Health Not Good’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Percent Mental Health Not Good After

temp_plot <- ggplot(final_data_3, aes(x=month_end, y=x25over_ment_hlth_sa)) +
  geom_line() +
  labs(title=’’, subtitle=’Percent Mental Health Not Good - Seasonally Adjusted’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Suicides Per Thousand Before

temp_plot <- ggplot(final_data_3, aes(x=month_end, y=x25on_suicides_per_thous)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

# Suicides Per Thousand After
temp_plot <- ggplot(final_data_3, aes(x=month_end, y=x25over_suicide_sa)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand - Seasonally Adjusted’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

grid <- grid.arrange(grobs=temp_list, nrow=4,top=’Before & After Seasonal Adjustment’)
```

```
## Don’t know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don’t know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



Before & After Seasonal Adjustment

Age 24 and Under

Percent Mental Health Not Good

Percent Mental Health Not Good − Seasonally Adjusted

Suicides Per Thousand

Suicides Per Thousand − Seasonally Adjusted

Age 25 and Over

Percent Mental Health Not Good

Percent Mental Health Not Good − Seasonally Adjusted

Suicides Per Thousand

Suicides Per Thousand − Seasonally Adjusted

```r
ggsave(here('Plots', 'ba_season_adjust.png'),plot=grid,dpi=600, width = 8, height = 11, units='in')
```

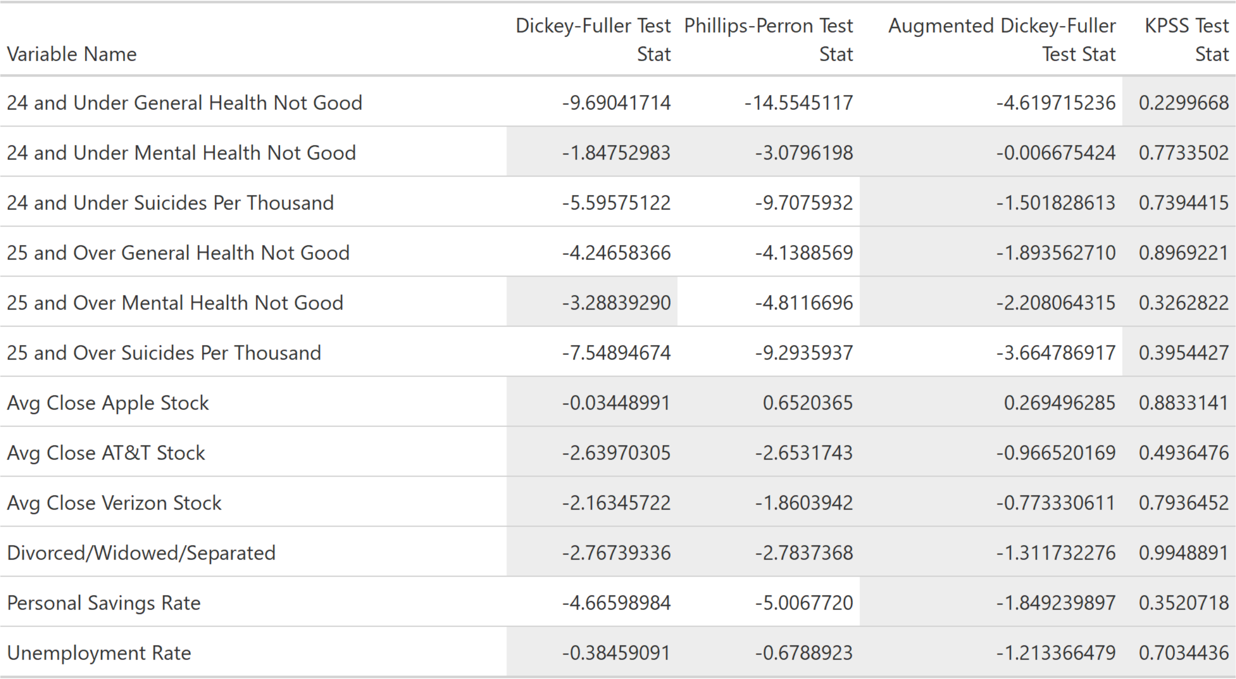
7 Test for Stationarity

Below we tested all variables for stationarity as they all appeared to increase over time (see above).

The table shown below highlights in grey, test statistics where the test statistics result in the variable being non-stationarity

```r
# variables we will test for stationarity
stationarity_vars <- c('genhlth_no_good_18_24',
                       'genhlth_no_good_25on',
                       'divorced_widowed_separated',
                       'avg_close_apple_stock',
                       'avg_close_atandt_stock',
                       'avg_close_verizon_stock',
                       'unemp_per',
                       'personal_save_rate',
                       'x24under_suicide_sa',
                       'x25over_suicide_sa',
                       'x24under_ment_hlth_sa',
                       'x25over_ment_hlth_sa')

# vectors to hold the test statistics
df_testStat <- c()
df_critValue <- c()
pp_testStat <- c()
pp_critValue <- c()
adfgls_testStat <- c()
adfgls_critValue <- c()
kpss_testStat <- c()
kpss_critValue <- c()


for(v in stationarity_vars){
  # dickey-fuller test
  temp_df <-ur.df(ts_final_data[,v], type=c('trend'), selectlags='BIC')
  df_testStat <- c(df_testStat, temp_df@teststat[1])
  df_critValue <- c(df_critValue, temp_df@cval[1,2])

  # phillips-perron test
  temp_pp <-ur.pp(ts_final_data[,v], type=('Z-tau'), model=c('trend'))
  pp_testStat <- c(pp_testStat, temp_pp@teststat[1])
  pp_critValue <- c(pp_critValue, temp_pp@cval[1,2])

  # augmented dickey-Fuller
  temp_adfgls <- ur.ers(ts_final_data[,v], type='DF-GLS', model='trend')
  adfgls_testStat <- c(adfgls_testStat, temp_adfgls@teststat[1])
  adfgls_critValue <- c(adfgls_critValue, temp_adfgls@cval[1,2])

  # kpss test
  temp_kpss <- ur.kpss(ts_final_data[,v], type=c('tau'))
  kpss_testStat <- c(kpss_testStat, temp_kpss@teststat[1])
  kpss_critValue <- c(kpss_critValue, temp_kpss@cval[1,2])
}

# create new tibble with stats
stationarity_stats <- tibble(stationarity_vars,
                             df_testStat,
                             df_critValue,
                             pp_testStat,
                             pp_critValue,
                             adfgls_testStat,
                             adfgls_critValue,
                             kpss_testStat,
                             kpss_critValue)

# add columns to compare the test stats to the critical values.  Include 1
# if the variables is stationary and 0 if not.  For kpss the null and
# alternative are different

stationarity_stats <- stationarity_stats %>%
  mutate(df_result = if_else(abs(df_testStat)>abs(df_critValue),1,0),
         pp_result = if_else(abs(pp_testStat)>abs(pp_critValue),1,0),
         adfgls_result = if_else(abs(adfgls_testStat)>abs(adfgls_critValue),1,0),
         kpss_result = if_else(abs(kpss_testStat)>abs(kpss_critValue),0,1),
         total_stationary=df_result+pp_result+adfgls_result+kpss_result)

##  create table with stationarity stats

alpha_1 <- 0.4

station_pretty <- stationarity_stats %>%
  left_join(all_names_cleaned, by=c(’stationarity_vars’=’all_names’)) %>%
  dplyr::arrange(clean_name) %>%
  dplyr::select(’Variable Name’=clean_name,
         ’Dickey-Fuller Test Stat’=df_testStat,
         ’Phillips-Perron Test Stat’=pp_testStat,
         ’Augmented Dickey-Fuller Test Stat’=adfgls_testStat,
         ’KPSS Test Stat’=kpss_testStat,
         df_result,
         pp_result,
         adfgls_result,
         kpss_result) %>%
  gt()  %>%
  tab_style(
    style = cell_fill(color = "lightgray", alpha=alpha_1),
    locations = cells_body(
      columns = ’Dickey-Fuller Test Stat’,
      rows = df_result==0)
  ) %>%
  tab_style(
    style = cell_fill(color = "lightgray", alpha=alpha_1),
    locations = cells_body(
      columns = ’Phillips-Perron Test Stat’,
      rows = pp_result==0)
  ) %>%
  tab_style(
    style = cell_fill(color = "lightgray", alpha=alpha_1),
    locations = cells_body(
      columns = ’Augmented Dickey-Fuller Test Stat’,
      rows = adfgls_result==0)
  ) %>%
  tab_style(
    style = cell_fill(color = "lightgray", alpha=alpha_1),
    locations = cells_body(
      columns = ’KPSS Test Stat’,
      rows = kpss_result==0)
  ) %>%
  cols_hide(columns=c(’df_result’,
                      ’pp_result’,
                      ’adfgls_result’,
                      ’kpss_result’)) %>%
  cols_width(
    ’Variable Name’ ~ px(400)
  )



gt::gtsave(station_pretty ,here(’Plots’, ’stationarity_test_results.png’))
```

```
## Warning: The ‘.dots‘ argument of ‘group_by()‘ is deprecated as of dplyr 1.0.0.
```



Variable Name	Dickey-Fuller Test Stat	Phillips-Perron Test Stat	Augmented Dickey-Fuller Test Stat	KPSS Test Stat
24 and Under General Health Not Good	-9.69041714	-14.5545117	-4.619715236	0.2299668
24 and Under Mental Health Not Good	-1.84752983	-3.0796198	-0.006675424	0.7733502
24 and Under Suicides Per Thousand	-5.59575122	-9.7075932	-1.501828613	0.7394415
25 and Over General Health Not Good	-4.24658366	-4.1388569	-1.893562710	0.8969221
25 and Over Mental Health Not Good	-3.28839290	-4.8116696	-2.208064315	0.3262822
25 and Over Suicides Per Thousand	-7.54894674	-9.2935937	-3.664786917	0.3954427
Avg Close Apple Stock	-0.03448991	0.6520365	0.269496285	0.8833141
Avg Close AT&T Stock	-2.63970305	-2.6531743	-0.966520169	0.4936476
Avg Close Verizon Stock	-2.16345722	-1.8603942	-0.773330611	0.7936452
Divorced/Widowed/Separated	-2.76739336	-2.7837368	-1.311732276	0.9948891
Personal Savings Rate	-4.66598984	-5.0067720	-1.849239897	0.3520718
Unemployment Rate	-0.38459091	-0.6788923	-1.213366479	0.7034436

8 Adjust for Stationarity

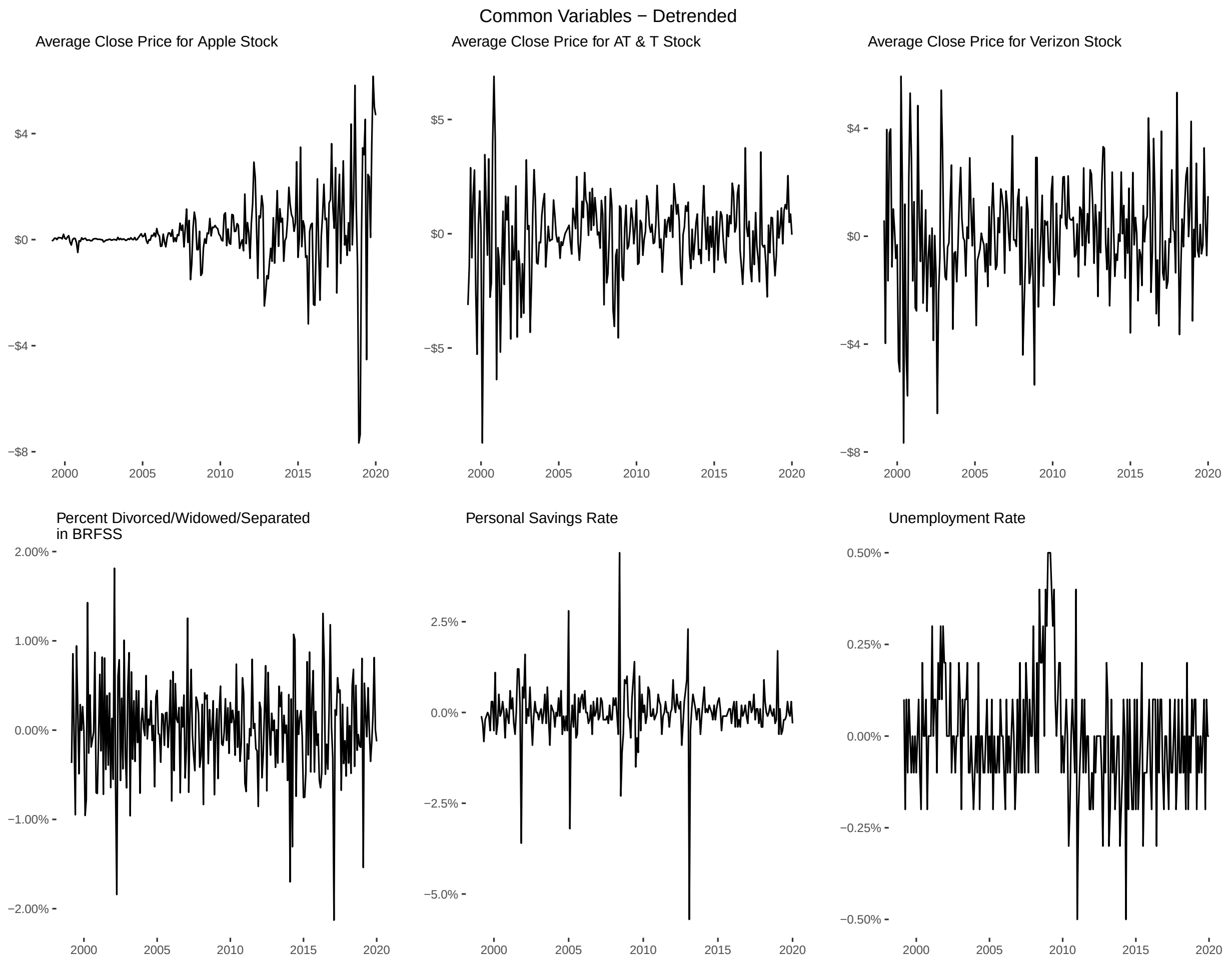
Next we adjusted all variables for stationarity

```
nvar <- nrow(stationarity_stats)
nobs <- nrow(ts_final_data)

# remove the first variable because we adjusted for seasonality
final_data_4 <- final_data_3 %>%
  dplyr::select(month_end) %>%
  filter(month_end!=ymd(’1999-01-31’))

for(i in 1:nvar){

  # If all four tests say that the data is stationary, don’t adjust
  if(stationarity_stats[i,’total_stationary’]==4){
    var_name <- stationarity_vars[i]

    final_data_4 <- final_data_4 %>%
      mutate(!!var_name:=ts_final_data[2:nobs,stationarity_vars[i]])
  } else {
    var_name <- paste(stationarity_vars[i],’_noTrend’,sep=’’)
    final_data_4 <- final_data_4 %>%
      mutate(!!var_name:=diff(ts_final_data[,stationarity_vars[i]]))
  }
}
```

8b Graph After All Adjustments

```
## Graph Final Variables

##### Common Variables

temp_list <- list()

# Apple Stock Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=avg_close_apple_stock_noTrend)) +
  geom_line() +
  labs(title=’Average Close Price for Apple Stock’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = dollar_format())

temp_list <- append(temp_list, list(temp_plot))

# AT & T Stock Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=avg_close_atandt_stock_noTrend)) +
  geom_line() +
  labs(title=’Average Close Price for AT & T Stock’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = dollar_format())
```

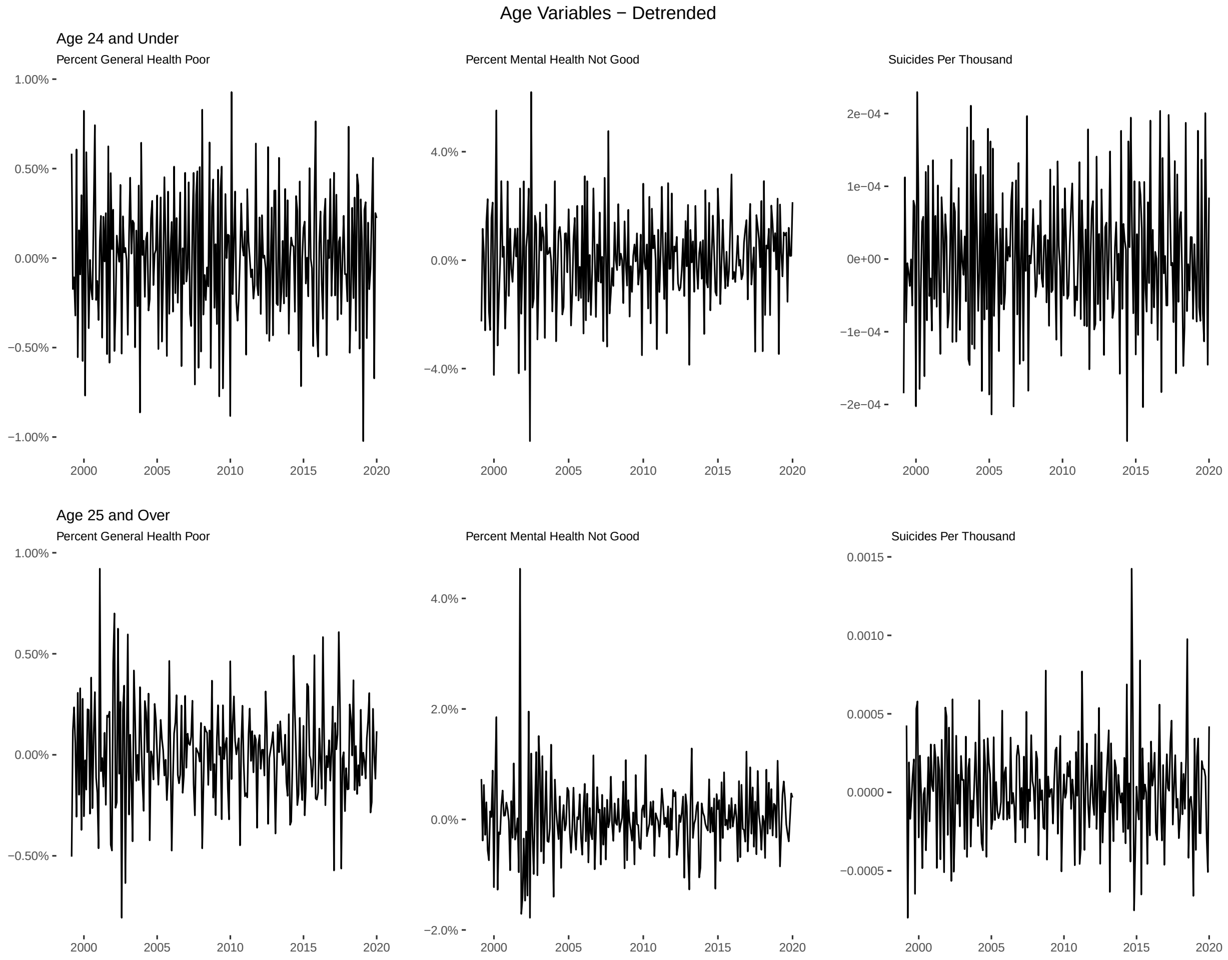
```
temp_list <- append(temp_list, list(temp_plot))

# Verizon Stock Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=avg_close_verizon_stock_noTrend)) +
  geom_line() +
  labs(title=’Average Close Price for Verizon Stock’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = dollar_format())

temp_list <- append(temp_list, list(temp_plot))

# Divorced/Widowed/Separated Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=divorced_widowed_separated_noTrend)) +
  geom_line() +
  labs(title=’Percent Divorced/Widowed/Separated \nin BRFSS’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Personal Savings Rate Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=personal_save_rate_noTrend)) +
  geom_line() +
  labs(title=’Personal Savings Rate’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Unemployment Rate Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=unemp_per_noTrend)) +
  geom_line() +
  labs(title=’Unemployment Rate’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

grid <- grid.arrange(grobs=temp_list, nrow=2,top=’Common Variables - Detrended’)
```



```
ggsave(here('Plots', 'common_variables_detrended.png'),plot=grid,dpi=600, width = 11.5, height = 9, uni

##### Age Variables

temp_list <- list()

# Gen Health No Good
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=genhlth_no_good_18_24_noTrend)) +
  geom_line() +
  labs(title='Age 24 and Under',subtitle='Percent General Health Poor' ,x='', y='') +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Ment Health No Good
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=x24under_ment_hlth_sa_noTrend)) +
  geom_line() +
  labs(title='', subtitle='Percent Mental Health Not Good', x='', y='') +
  main_theme +
  scale_y_continuous(labels = percent)
```

```r
temp_list <- append(temp_list, list(temp_plot))

# Per 1000 Suicides Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=x24under_suicide_sa_noTrend)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

# Gen Health No Good
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=genhlth_no_good_25on_noTrend)) +
  geom_line() +
  labs(title=’Age 25 and Over’,subtitle=’Percent General Health Poor’ ,x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Ment Health No Good
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=x25over_ment_hlth_sa_noTrend)) +
  geom_line() +
  labs(title=’’, subtitle=’Percent Mental Health Not Good’, x=’’, y=’’) +
  main_theme +
  scale_y_continuous(labels = percent)

temp_list <- append(temp_list, list(temp_plot))

# Per 1000 Suicides Graph
temp_plot <- ggplot(final_data_4, aes(x=month_end, y=x25over_suicide_sa_noTrend)) +
  geom_line() +
  labs(title=’’, subtitle=’Suicides Per Thousand’, x=’’, y=’’) +
  main_theme

temp_list <- append(temp_list, list(temp_plot))

grid <- grid.arrange(grobs=temp_list, nrow=2,top=’Age Variables - Detrended’)
```

```
## Don’t know how to automatically pick scale for object of type ts. Defaulting to continuous.
## Don’t know how to automatically pick scale for object of type ts. Defaulting to continuous.
```



```
ggsave(here('Plots', 'age_variables_detrended.png'),plot=grid,dpi=600, width = 11.5, height = 9, units=
```

9 Fit Models

Next, after adjusting for seasonality and stationarity we decided to fit two VAR models. First we fit one VAR for 24 and under and graphed the impulse response function for all variables against suicide rates for 24 and under. We did the same for 25 and over.

```
# create theme for the VAR plots
main_theme <- theme(panel.grid = element_blank(),
                    panel.background = element_blank(),
                    plot.title = element_text(size=9),
                    plot.subtitle = element_text(size=8),
                    axis.title.x = element_text(size=8),
                    axis.title.y = element_text(size=8),
                    axis.text.x = element_text(size=8),
                    axis.text.y = element_text(size=8))

# grab only the variables for adolescents (24 and under)
x24_under <- final_data_4 %>%
  dplyr::select(-c(genhlth_no_good_25on_noTrend,
            x25over_suicide_sa_noTrend,
            x25over_ment_hlth_sa_noTrend))

# convert to time-series and the VAR models
x24_under_ts <- ts(x24_under)
x24_under_var <- VAR(x24_under_ts, lag.max = 12, ic = 'AIC')

x24under_names <- names(x24_under)


######################

# tibbles to hold the impulse response functions as well as the lower and
# upper bounds
lower <- tibble(month=1:13)
irf <- tibble(month=1:13)
upper <- tibble(month=1:13)



for(name in x24under_names){

  if(name!='month_end'){

  # when we fit the final models maybe do a 1000 runs
  temp_irf <- irf(x24_under_var,impulse=c(name), response=c("x24under_suicide_sa_noTrend"), n.ahead=12,

  lower <- bind_cols(lower, !!name:=temp_irf$Lower[[1]])
  upper <- bind_cols(upper, !!name:=temp_irf$Upper[[1]])
  irf <- bind_cols(irf, !!name:=temp_irf$irf[[1]])

  }
  print(name)
}
```

```
## [1] "month_end"
## [1] "genhlth_no_good_18_24_noTrend"
## [1] "divorced_widowed_separated_noTrend"
## [1] "avg_close_apple_stock_noTrend"
## [1] "avg_close_atandt_stock_noTrend"
## [1] "avg_close_verizon_stock_noTrend"
## [1] "unemp_per_noTrend"
## [1] "personal_save_rate_noTrend"
## [1] "x24under_suicide_sa_noTrend"
## [1] "x24under_ment_hlth_sa_noTrend"
```

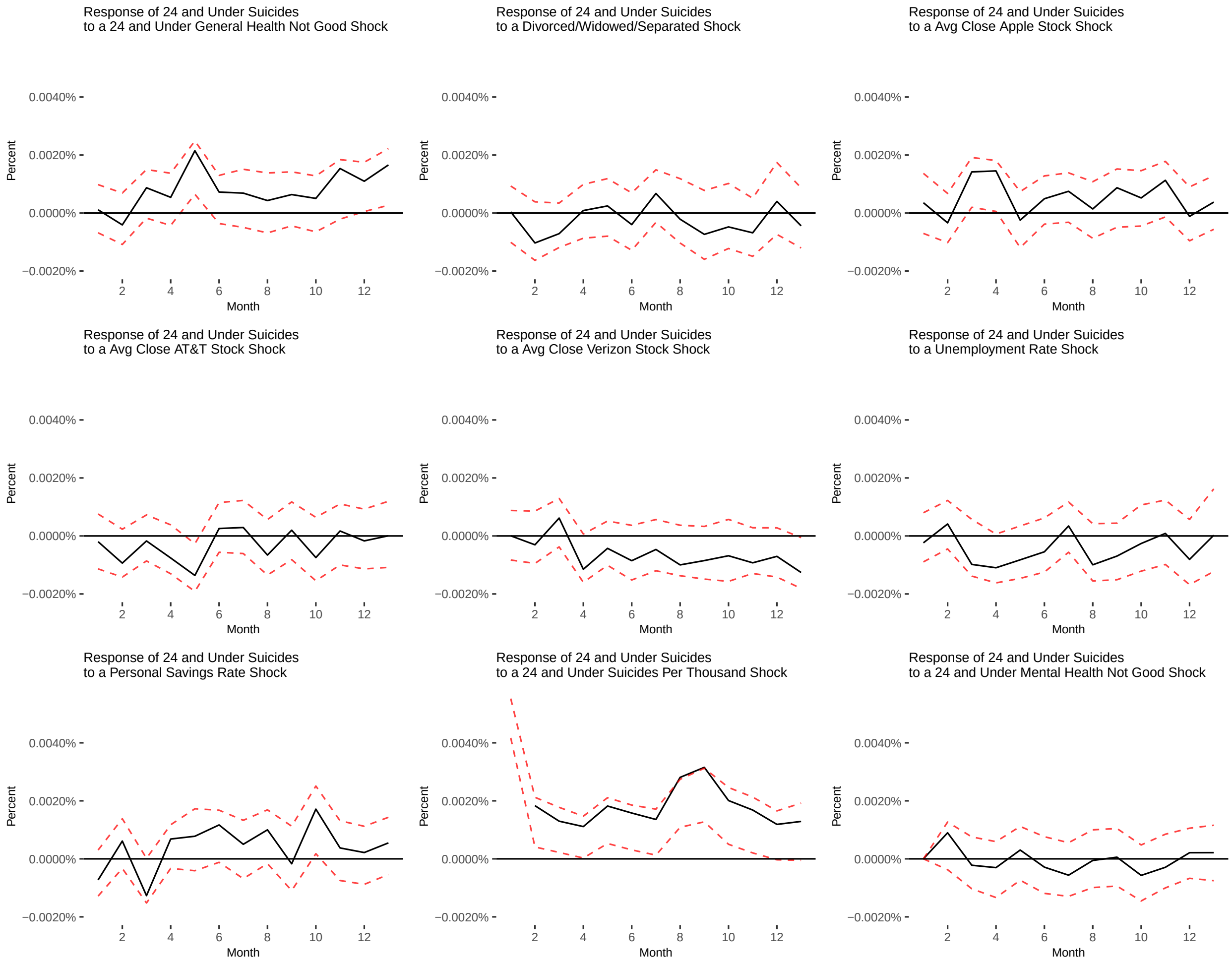
```
# get minimum and maximum for lower and upper bounds
min_lower <- min(lower[,2:ncol(lower)])
max_upper <- max(upper[,2:ncol(upper)])

temp_list <- list()
```

```r
for(name in x24under_names){

  # get the nice names for graphing
  clean_name <- as.character(all_names_cleaned %>% filter(all_names==name) %>% dplyr::select(clean_name)

  if(name!=’month_end’){


    temp_plot <- ggplot(data=lower, aes_string(x=’month’, y=name)) +
      geom_line(color=’red’, alpha=0.75, linetype = "dashed") + #lower bound
      geom_line(data=irf, aes_string(x=’month’, y=name)) + # irf
      geom_line(data=upper, aes_string(x=’month’, y=name), color=’red’, alpha=0.75, linetype = "dashed")
      geom_hline(yintercept=0) +
      scale_x_continuous(breaks=c(2,4,6,8,10,12)) +
      scale_y_continuous(labels = percent, limits=c(min_lower,max_upper)) +
      main_theme +
      labs(x=’Month’,
          y=’Percent’,
          title=paste(’Response of 24 and Under Suicides \nto a ’,clean_name,’ Shock’,sep=’’))

    temp_list <- append(temp_list, list(temp_plot))

  }
  print(name)
}
```

```
## [1] "month_end"
## [1] "genhlth_no_good_18_24_noTrend"
## [1] "divorced_widowed_separated_noTrend"
## [1] "avg_close_apple_stock_noTrend"
## [1] "avg_close_atandt_stock_noTrend"
## [1] "avg_close_verizon_stock_noTrend"
## [1] "unemp_per_noTrend"
## [1] "personal_save_rate_noTrend"
## [1] "x24under_suicide_sa_noTrend"
## [1] "x24under_ment_hlth_sa_noTrend"
```

```r
grid <- grid.arrange(grobs=temp_list, nrow=3)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



Response of 24 and Under Suicides
to a 24 and Under General Health Not Good Shock

Response of 24 and Under Suicides
to a Divorced/Widowed/Separated Shock

Response of 24 and Under Suicides
to a Avg Close Apple Stock Shock

Response of 24 and Under Suicides
to a Avg Close AT&T Stock Shock

Response of 24 and Under Suicides
to a Avg Close Verizon Stock Shock

Response of 24 and Under Suicides
to a Unemployment Rate Shock

Response of 24 and Under Suicides
to a Personal Savings Rate Shock

Response of 24 and Under Suicides
to a 24 and Under Suicides Per Thousand Shock

Response of 24 and Under Suicides
to a 24 and Under Mental Health Not Good Shock

```
ggsave(here(’Plots’, ’24_and_under_VAR_shocks.png’),plot=grid,dpi=600, width = 11.5, height = 9, units=
```

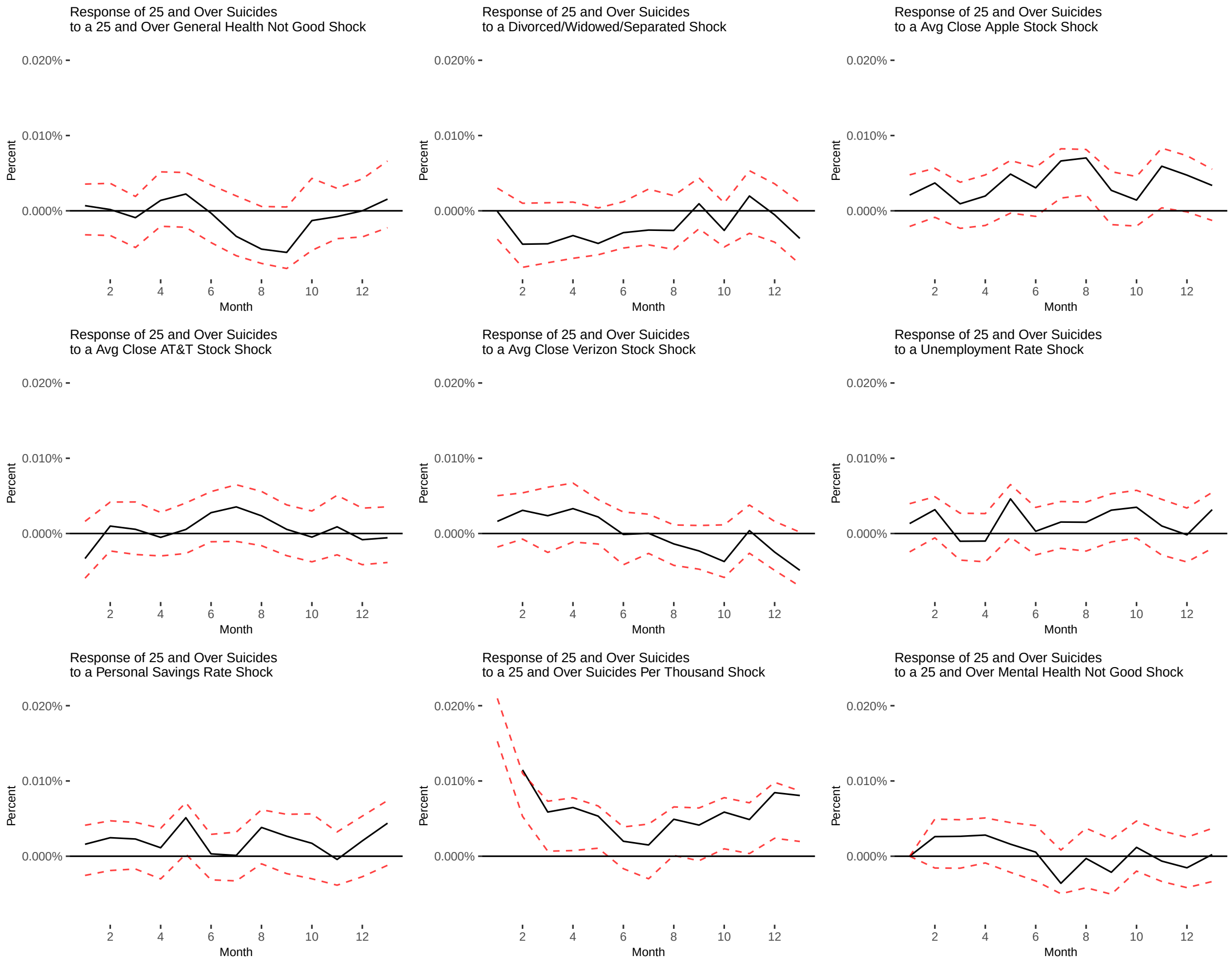
After this we did the same for the 25 and over

```
x25_over <- final_data_4 %>%
  dplyr::select(-c(genhlth_no_good_18_24_noTrend,
                   x24under_suicide_sa_noTrend,
                   x24under_ment_hlth_sa_noTrend))

x25_over_ts <- ts(x25_over)
x25_over_var <- VAR(x25_over_ts, lag.max = 12, ic = ’AIC’)
x25_over_names <- names(x25_over)

######################


lower <- tibble(month=1:13)
irf <- tibble(month=1:13)
upper <- tibble(month=1:13)

for(name in x25_over_names){
  if(name!='month_end'){

    temp_irf <- irf(x25_over_var,impulse=c(name), response=c("x25over_suicide_sa_noTrend"), n.ahead=12,

    lower <- bind_cols(lower, !!name:=temp_irf$Lower[[1]])
    upper <- bind_cols(upper, !!name:=temp_irf$Upper[[1]])
    irf <- bind_cols(irf, !!name:=temp_irf$irf[[1]])

  }
}

min_lower <- min(lower[,2:ncol(lower)])
max_upper <- max(upper[,2:ncol(upper)])

temp_list <- list()

for(name in x25_over_names){

  clean_name <- as.character(all_names_cleaned %>% filter(all_names==name) %>% dplyr::select(clean_name

  if(name!='month_end'){

    temp_plot <- ggplot(data=lower, aes_string(x='month', y=name)) +
      geom_line(color='red', alpha=0.75, linetype = "dashed") +
      geom_line(data=irf, aes_string(x='month', y=name)) +
      geom_line(data=upper, aes_string(x='month', y=name), color='red', alpha=0.75, linetype = "dashed"
      geom_hline(yintercept=0) +
      scale_x_continuous(breaks=c(2,4,6,8,10,12)) +
      scale_y_continuous(labels = percent, limits=c(min_lower,max_upper)) +
      main_theme +
      labs(x='Month',
           y='Percent',
           title=paste('Response of 25 and Over Suicides \nto a ',clean_name,' Shock',sep=''))

    temp_list <- append(temp_list, list(temp_plot))

  }
  print(name)
}
```

```
## [1] "month_end"
## [1] "genhlth_no_good_25on_noTrend"
## [1] "divorced_widowed_separated_noTrend"
## [1] "avg_close_apple_stock_noTrend"
## [1] "avg_close_atandt_stock_noTrend"
## [1] "avg_close_verizon_stock_noTrend"
## [1] "unemp_per_noTrend"
## [1] "personal_save_rate_noTrend"
## [1] "x25over_suicide_sa_noTrend"
## [1] "x25over_ment_hlth_sa_noTrend"
```

```
grid <- grid.arrange(grobs=temp_list, nrow=3)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```


Response of 25 and Over Suicides to a 25 and Over General Health Not Good Shock

Response of 25 and Over Suicides to a Divorced/Widowed/Separated Shock

Response of 25 and Over Suicides to a Avg Close Apple Stock Shock

Response of 25 and Over Suicides to a Avg Close AT&T Stock Shock

Response of 25 and Over Suicides to a Avg Close Verizon Stock Shock

Response of 25 and Over Suicides to a Unemployment Rate Shock

Response of 25 and Over Suicides to a Personal Savings Rate Shock

Response of 25 and Over Suicides to a 25 and Over Suicides Per Thousand Shock

Response of 25 and Over Suicides to a 25 and Over Mental Health Not Good Shock

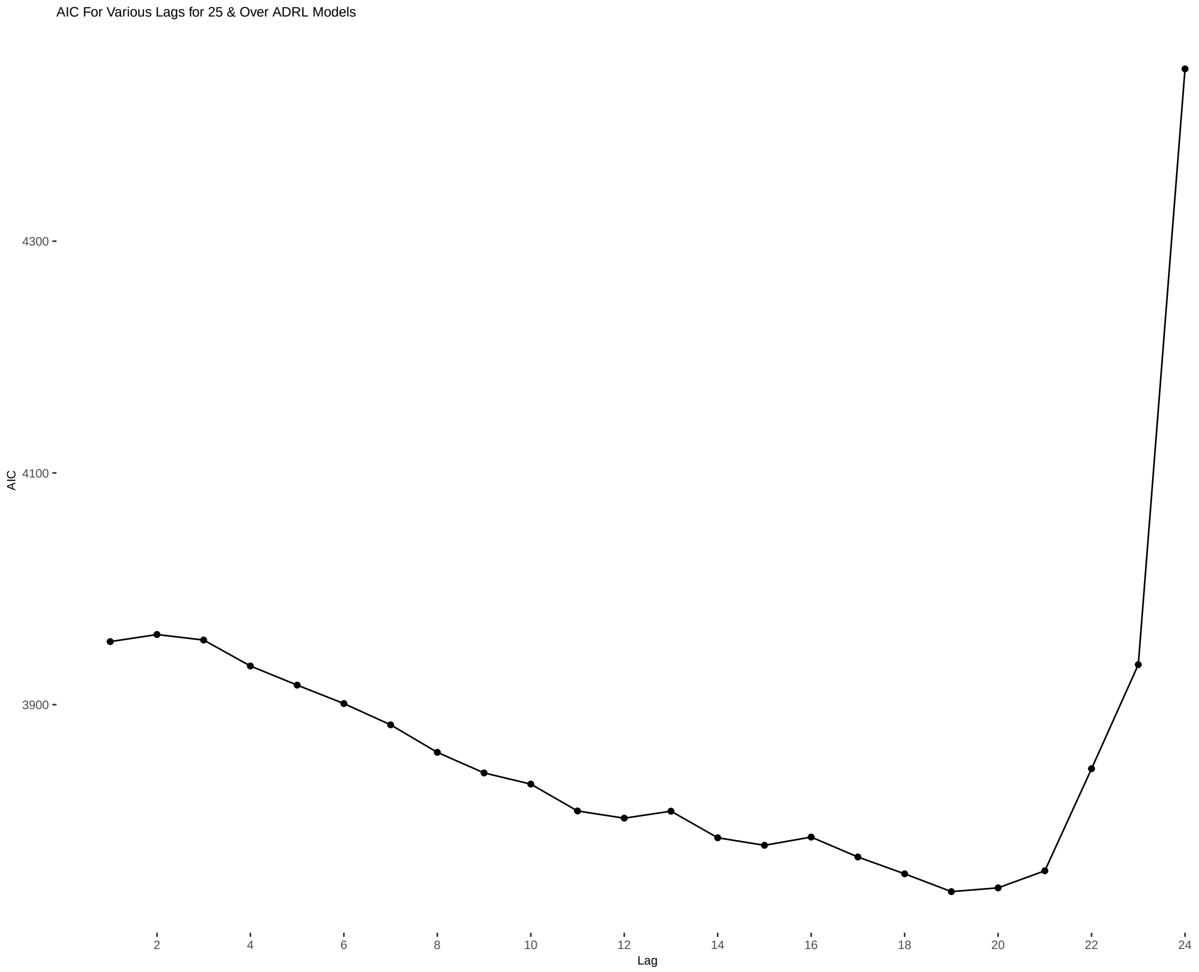
```
ggsave(here('Plots', '25_and_over_VAR_shocks.png'),plot=grid,dpi=600, width = 11.5, height = 9, units='
```

10 Fit ARIMA Models and Compare the AIC For Various Lags

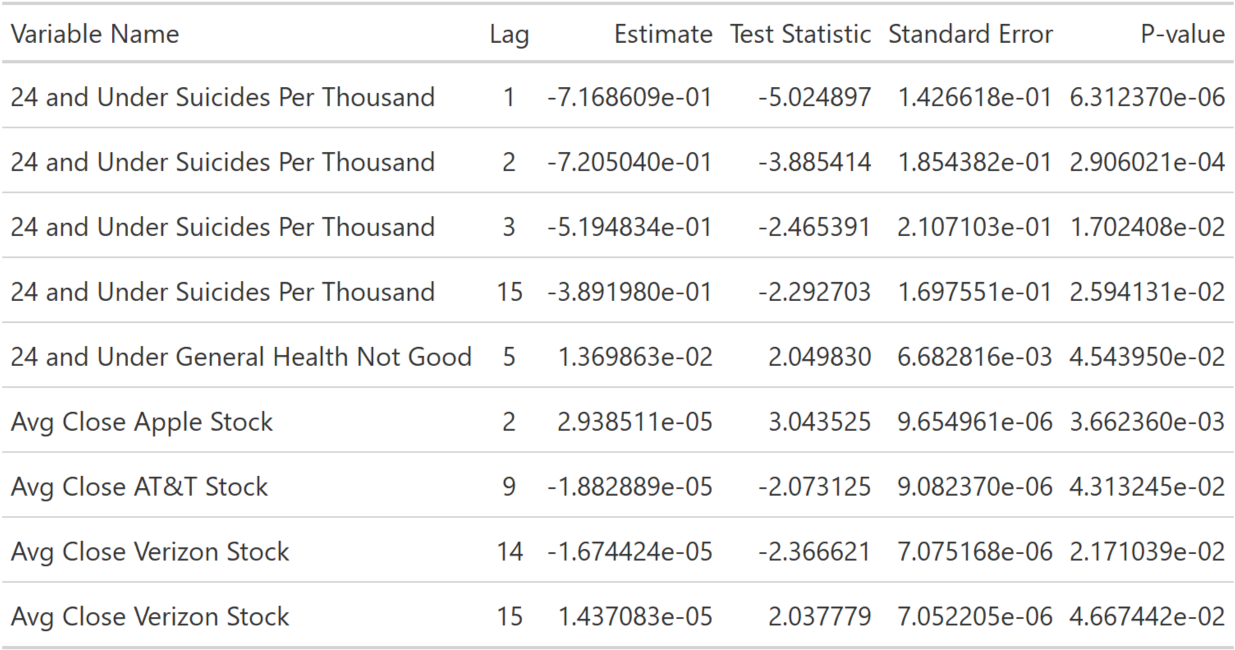
Next, we fit ARIMA models with 24 lag variables each and graphed the AIC for each set of lags.

First the 24 and under

```
max_lag <- 24

ardl_aic <- c()
ardl_models <- c()

for(i in 1:max_lag){
  temp_ARDL <- dynlm(x24under_suicide_sa_noTrend ~
                      L(x24under_suicide_sa_noTrend, 1:i) +
                      L(genhlth_no_good_18_24_noTrend, 0:i) +
                      L(divorced_widowed_separated_noTrend, 0:i) +
                      L(avg_close_apple_stock_noTrend, 0:i) +
                      L(avg_close_atandt_stock_noTrend, 0:i) +
                      L(avg_close_verizon_stock_noTrend, 0:i) +
                      L(unemp_per_noTrend, 0:i) +
                      L(personal_save_rate_noTrend, 0:i) +
                      L(x24under_ment_hlth_sa_noTrend, 0:i)
                    , data = x24_under_ts)

  ardl_models <- append(ardl_models, list(temp_ARDL))
  ardl_aic <- append(ardl_aic, abs(AIC(temp_ARDL)))
}

ardl_aic_tibble <- tibble(lag=1:max_lag, aic=ardl_aic)

aic_plot <- ggplot(ardl_aic_tibble, aes(x=lag, y=aic)) +
  geom_line() +
  geom_point() +
  main_theme +
  scale_x_continuous(breaks=seq(from=2, to=max_lag, by =2)) +
  labs(title='AIC For Various Lags for 25 & Over ADRL Models', x='Lag', y='AIC')

ggsave(here('Plots', 'aic_graph_24under.png'),plot=aic_plot,dpi=600, width = 11.5, height = 9, units='i

aic_plot
```



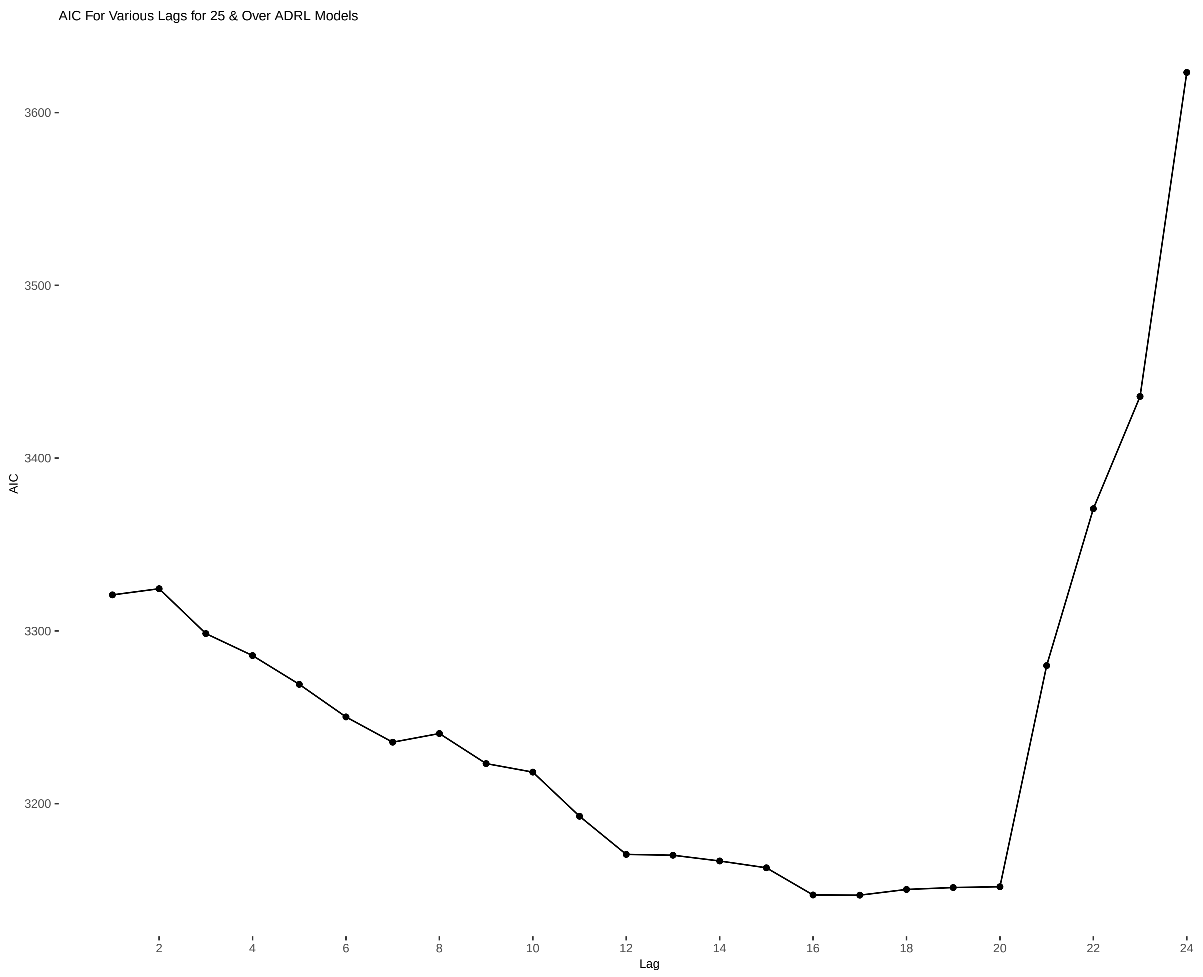
```
best_model_coef <- coef(summary(ardl_models[[which.min(ardl_aic)]]))

best_model_tibble <- bind_cols(var_name=rownames(best_model_coef), as_tibble(best_model_coef)) %>%
  clean_names() %>%
  mutate(Lag =  rep(seq(0,which.min(ardl_aic)),9))

best_model_pretty_table <- best_model_tibble %>%
  mutate(pretty_var_name =  gsub(’L\\(’,’’,gsub(’,.*’,’’,var_name))) %>%
  left_join(all_names_cleaned, by=c(’pretty_var_name’=’all_names’)) %>%
  mutate(clean_name = if_else(is.na(clean_name),’Intercept’, clean_name)) %>%
  filter(pr_t<=0.05) %>%
  rename(’P-value’ = pr_t,
         ’Estimate’=estimate,
         ’Standard Error’=std_error,
         ’Test Statistic’=t_value,
         ’Variable Name’=clean_name) %>%
  dplyr::select(’Variable Name’,
                ’Lag’,
                ’Estimate’,
                ’Test Statistic’,
                ’Standard Error’,
                ’P-value’) %>%
  gt()

gt::gtsave(best_model_pretty_table,here('Plots', '24under_armaModel_coef.png'))
```



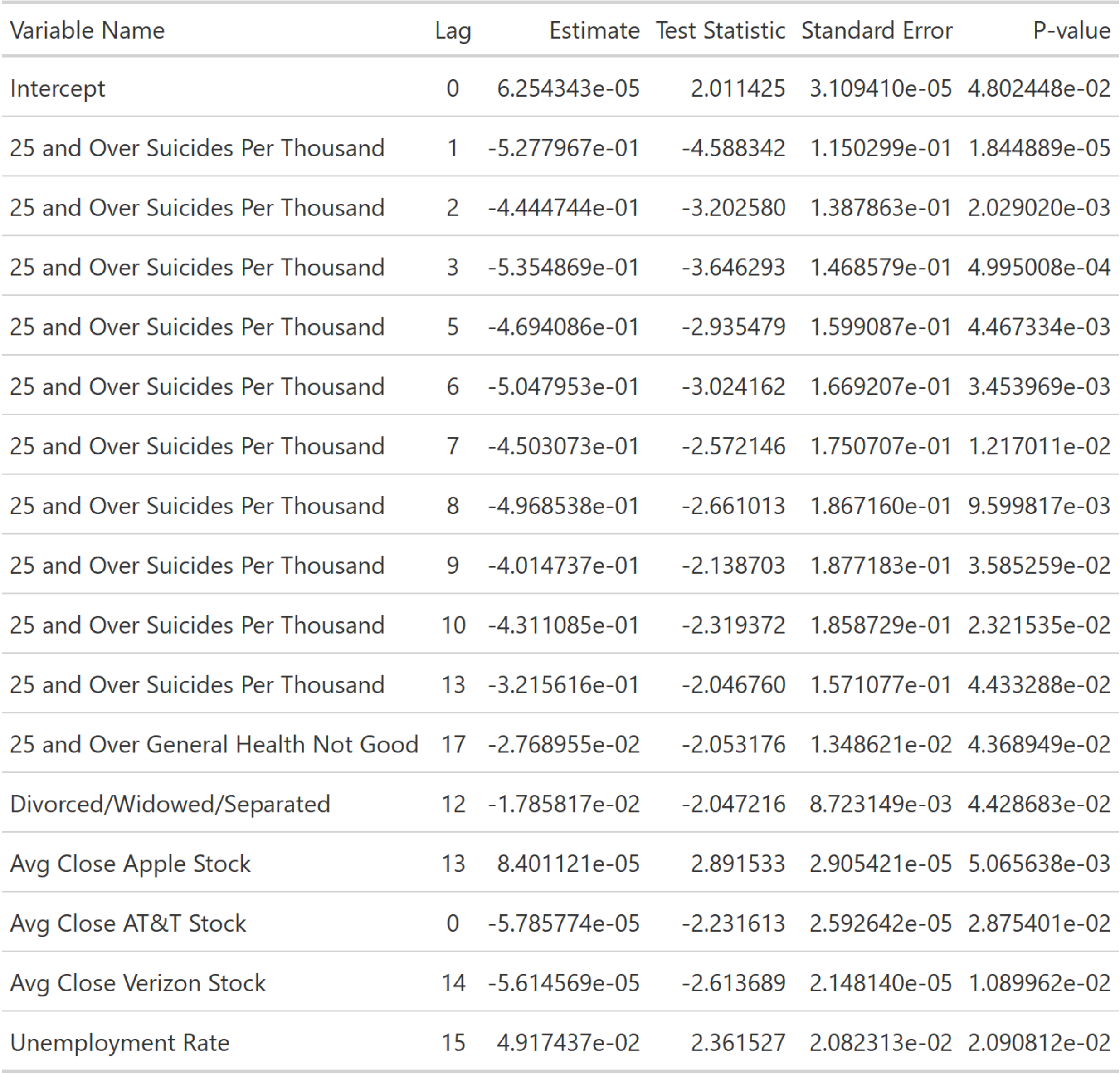
Variable Name	Lag	Estimate	Test Statistic	Standard Error	P-value
24 and Under Suicides Per Thousand	1	-7.168609e-01	-5.024897	1.426618e-01	6.312370e-06
24 and Under Suicides Per Thousand	2	-7.205040e-01	-3.885414	1.854382e-01	2.906021e-04
24 and Under Suicides Per Thousand	3	-5.194834e-01	-2.465391	2.107103e-01	1.702408e-02
24 and Under Suicides Per Thousand	15	-3.891980e-01	-2.292703	1.697551e-01	2.594131e-02
24 and Under General Health Not Good	5	1.369863e-02	2.049830	6.682816e-03	4.543950e-02
Avg Close Apple Stock	2	2.938511e-05	3.043525	9.654961e-06	3.662360e-03
Avg Close AT&T Stock	9	-1.882889e-05	-2.073125	9.082370e-06	4.313245e-02
Avg Close Verizon Stock	14	-1.674424e-05	-2.366621	7.075168e-06	2.171039e-02
Avg Close Verizon Stock	15	1.437083e-05	2.037779	7.052205e-06	4.667442e-02

Next we fit for 25 and over

```
max_lag <- 24

ardl_aic <- c()
ardl_models <- c()

for(i in 1:max_lag){
  temp_ARDL <- dynlm(x25over_suicide_sa_noTrend ~
                       L(x25over_suicide_sa_noTrend, 1:i) +
                       L(genhlth_no_good_25on_noTrend, 0:i) +
                       L(divorced_widowed_separated_noTrend, 0:i) +
                       L(avg_close_apple_stock_noTrend, 0:i) +
                       L(avg_close_atandt_stock_noTrend, 0:i) +
                       L(avg_close_verizon_stock_noTrend, 0:i) +
                       L(unemp_per_noTrend, 0:i) +
                       L(personal_save_rate_noTrend, 0:i) +
                       L(x25over_ment_hlth_sa_noTrend, 0:i)
                     , data = x25_over_ts)

  ardl_models <- append(ardl_models, list(temp_ARDL))
  ardl_aic <- append(ardl_aic, abs(AIC(temp_ARDL)))
}

ardl_aic_tibble <- tibble(lag=1:max_lag, aic=ardl_aic)

aic_plot <- ggplot(ardl_aic_tibble, aes(x=lag, y=aic)) +
  geom_line() +
  geom_point() +
  main_theme +
  scale_x_continuous(breaks=seq(from=2, to=max_lag, by =2)) +
  labs(title=’AIC For Various Lags for 25 & Over ADRL Models’, x=’Lag’, y=’AIC’)

ggsave(here(’Plots’, ’aic_graph_25over.png’),plot=aic_plot,dpi=600, width = 11.5, height = 9, units=’in

aic_plot
```



```
best_model_coef <- coef(summary(ardl_models[[which.min(ardl_aic)]]))

best_model_tibble <- bind_cols(var_name=rownames(best_model_coef), as_tibble(best_model_coef)) %>%
  clean_names() %>%
  mutate(Lag =  rep(seq(0,which.min(ardl_aic)),9))

best_model_pretty_table <- best_model_tibble %>%
  mutate(pretty_var_name =  gsub(’L\\(’,’’,gsub(’,.*’,’’,var_name))) %>%
  left_join(all_names_cleaned, by=c(’pretty_var_name’=’all_names’)) %>%
  mutate(clean_name = if_else(is.na(clean_name),’Intercept’, clean_name)) %>%
  filter(pr_t<=0.05) %>%
  rename(’P-value’ = pr_t,
         ’Estimate’=estimate,
         ’Standard Error’=std_error,
         ’Test Statistic’=t_value,
         ’Variable Name’=clean_name) %>%
  dplyr::select(’Variable Name’,
                ’Lag’,
                ’Estimate’,
                ’Test Statistic’,
                ’Standard Error’,
                ’P-value’) %>%
  gt()

gt::gtsave(best_model_pretty_table,here(’Plots’, ’25over_armaModel_coef.png’))
```



Variable Name	Lag	Estimate	Test Statistic	Standard Error	P-value
Intercept	0	6.254343e-05	2.011425	3.109410e-05	4.802448e-02
25 and Over Suicides Per Thousand	1	-5.277967e-01	-4.588342	1.150299e-01	1.844889e-05
25 and Over Suicides Per Thousand	2	-4.444744e-01	-3.202580	1.387863e-01	2.029020e-03
25 and Over Suicides Per Thousand	3	-5.354869e-01	-3.646293	1.468579e-01	4.995008e-04
25 and Over Suicides Per Thousand	5	-4.694086e-01	-2.935479	1.599087e-01	4.467334e-03
25 and Over Suicides Per Thousand	6	-5.047953e-01	-3.024162	1.669207e-01	3.453969e-03
25 and Over Suicides Per Thousand	7	-4.503073e-01	-2.572146	1.750707e-01	1.217011e-02
25 and Over Suicides Per Thousand	8	-4.968538e-01	-2.661013	1.867160e-01	9.599817e-03
25 and Over Suicides Per Thousand	9	-4.014737e-01	-2.138703	1.877183e-01	3.585259e-02
25 and Over Suicides Per Thousand	10	-4.311085e-01	-2.319372	1.858729e-01	2.321535e-02
25 and Over Suicides Per Thousand	13	-3.215616e-01	-2.046760	1.571077e-01	4.433288e-02
25 and Over General Health Not Good	17	-2.768955e-02	-2.053176	1.348621e-02	4.368949e-02
Divorced/Widowed/Separated	12	-1.785817e-02	-2.047216	8.723149e-03	4.428683e-02
Avg Close Apple Stock	13	8.401121e-05	2.891533	2.905421e-05	5.065638e-03
Avg Close AT&T Stock	0	-5.785774e-05	-2.231613	2.592642e-05	2.875401e-02
Avg Close Verizon Stock	14	-5.614569e-05	-2.613689	2.148140e-05	1.089962e-02
Unemployment Rate	15	4.917437e-02	2.361527	2.082313e-02	2.090812e-02