EC421ProblemSet2

Samuel Marsh

2025-05-25

1. Loading packages & data.

```
pacman::p_load(tidyverse, here, lmtest, ggplot2, dplyr, tibble, tidyr)

data_life = read.csv('/Users/sammarsh/Desktop/R DATASETS/EC 421 R/data-life.csv')
```

2. Taking a peek at the data to see what we are working with.

```
glimpse(data_life)
```

```
## Rows: 72
## Columns: 8
## $ year       <int> 1952, 1953, 1954, 1955, 1956, 1957, 1958, 1959, 1960, 1961,…
## $ exp_female <dbl> 71.50, 71.91, 72.70, 72.80, 72.95, 72.74, 73.00, 73.31, 73.…
## $ exp_male   <dbl> 65.54, 65.80, 66.57, 66.58, 66.61, 66.36, 66.60, 66.75, 66.…
## $ exp_pop    <dbl> 68.38, 68.72, 69.50, 69.56, 69.64, 69.41, 69.67, 69.90, 69.…
## $ pop        <dbl> 157.4926, 160.1462, 162.9683, 165.8723, 168.8577, 171.9068,…
## $ gdp        <dbl> 3.67341, 3.89218, 3.90549, 4.25478, 4.49353, 4.74039, 4.812…
## $ cpi        <dbl> 26.56667, 26.76833, 26.86500, 26.79583, 27.19083, 28.11333,…
## $ inf        <dbl> 2.2843943, 0.7590966, 0.3611232, -0.2574601, 1.4741098, 3.3…
```

```
max(data_life$year)
```

```
## [1] 2023
```

```
sum(is.na(data_life))
```

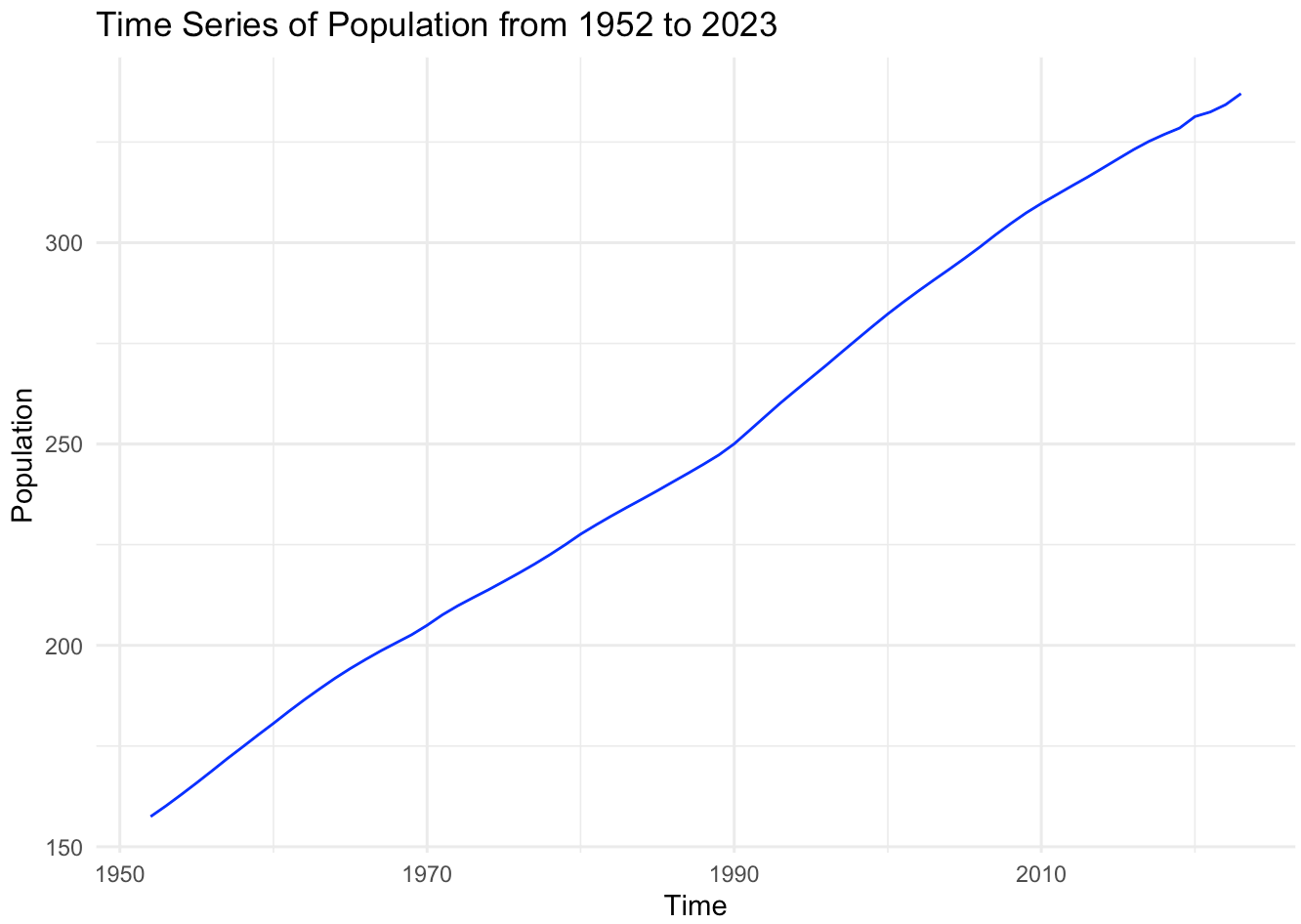
```
## [1] 0
```

- It looks like the data covers from 1952 through 2023.
- It appears to be ordered by year with 8 columns and 72 rows
- There are 0 missing values in any rows.

3. Now creating 5 time series plots.

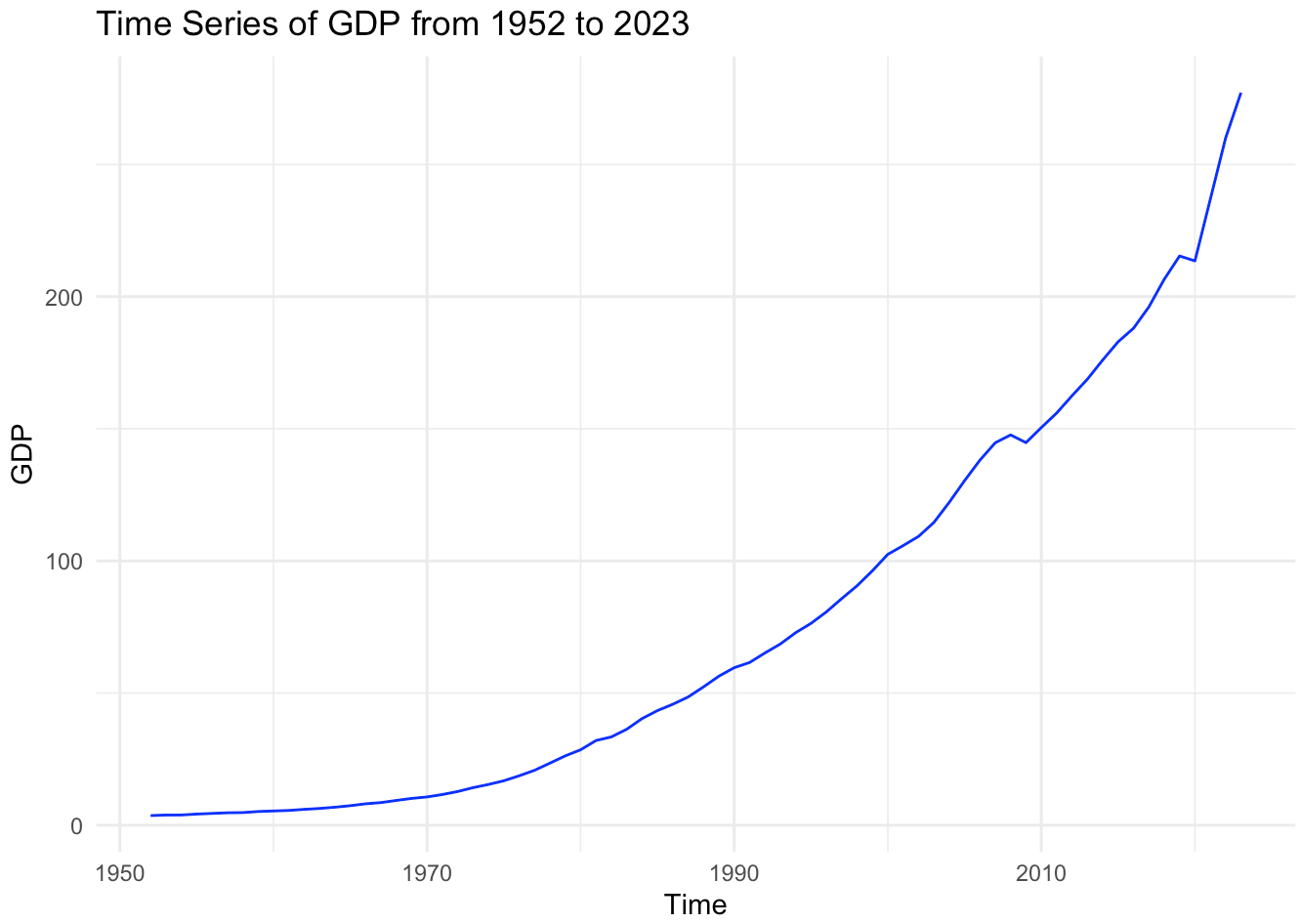
- a. population (pop)

```
ggplot(data_life, aes(x = year, y = pop)) +
  geom_line(color = 'blue') +
  labs(title = "Time Series of Population from 1952 to 2023",
       x = "Time",
       y = "Population") +
  theme_minimal()
```



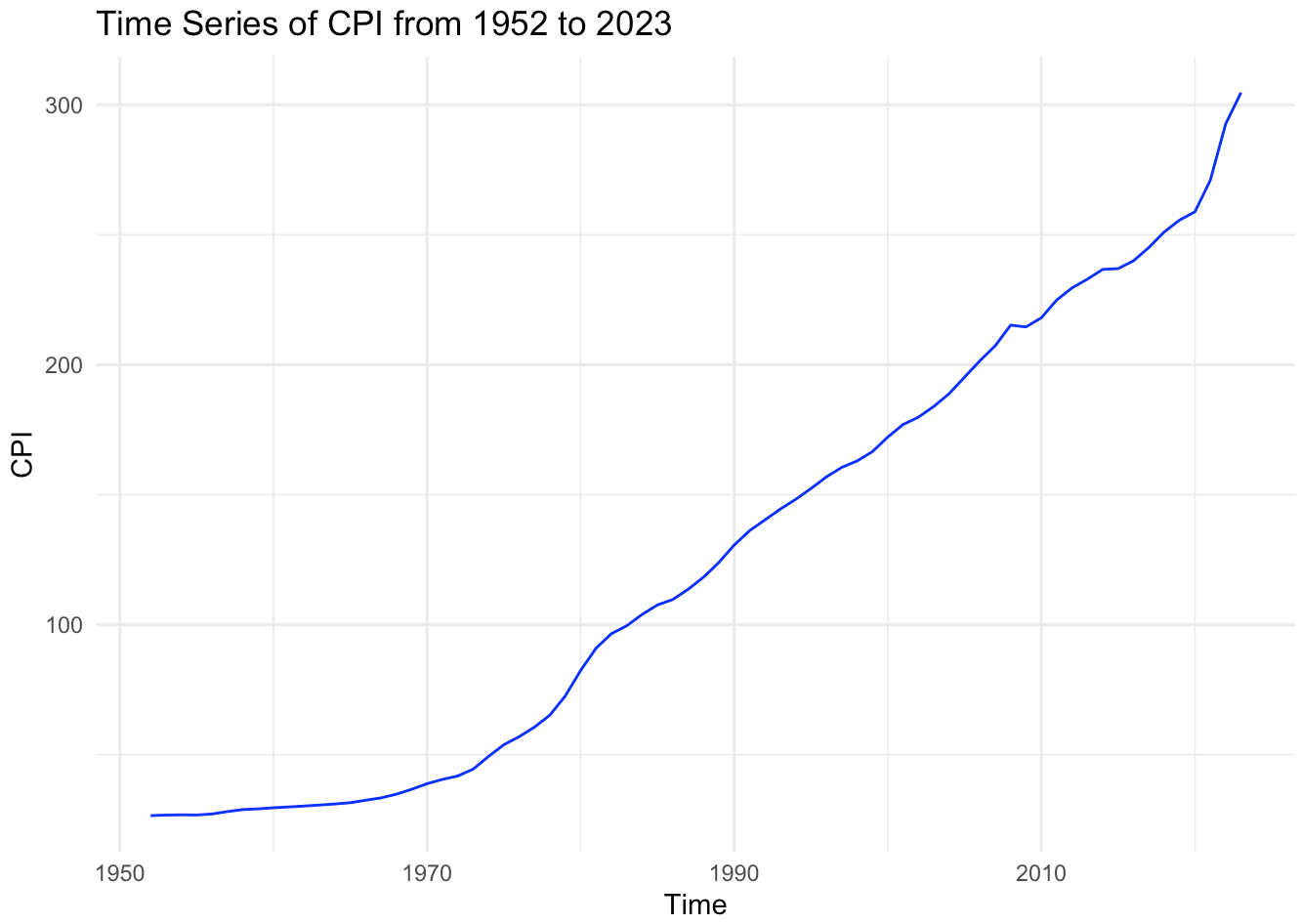
- b. GDP (gdp)

```
ggplot(data_life, aes(x = year, y = gdp)) +
  geom_line(color = 'blue') +
  labs(title = "Time Series of GDP from 1952 to 2023",
       x = "Time",
       y = "GDP") +
  theme_minimal()
```



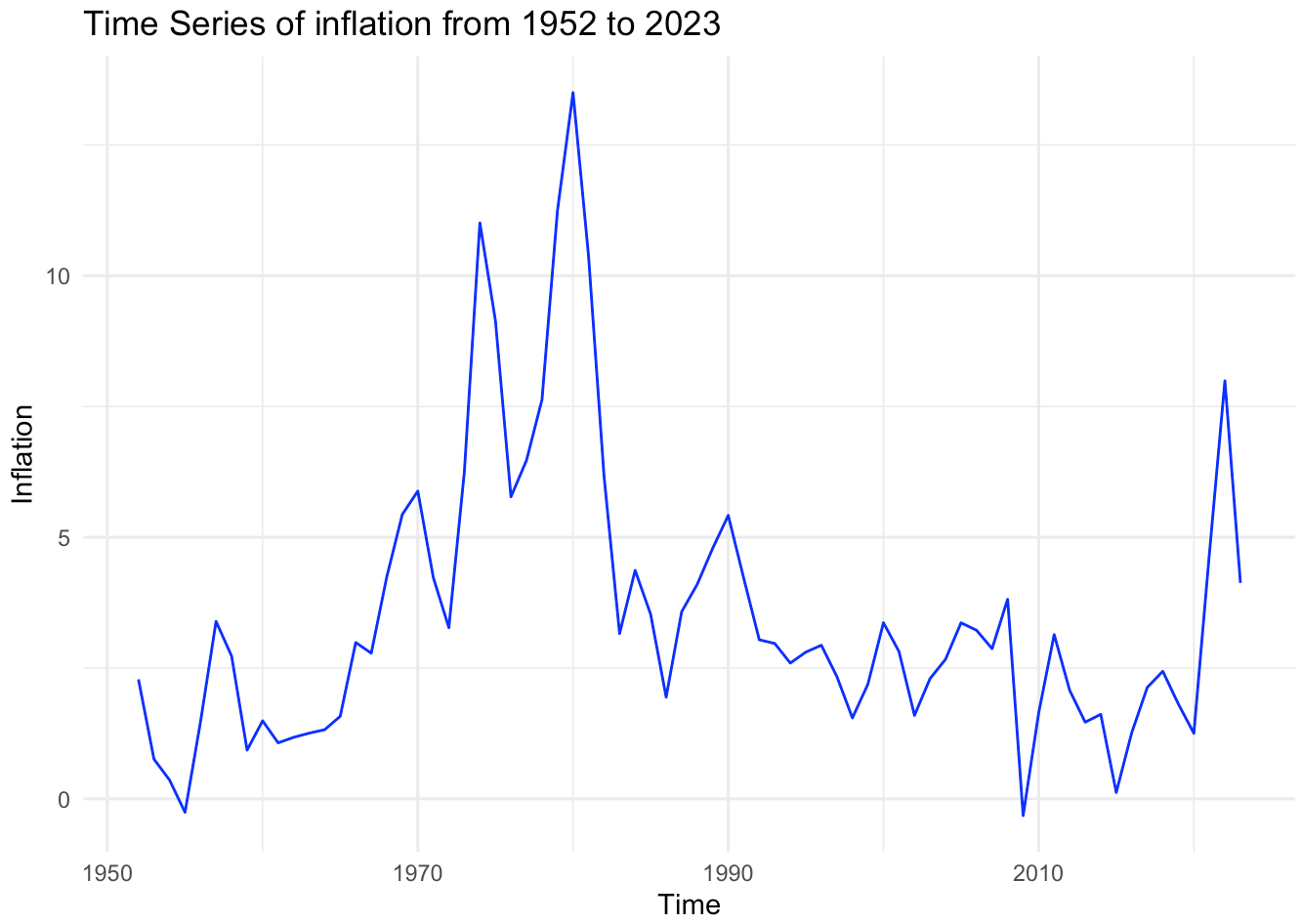
- c. CPI (cpi)

```
ggplot(data_life, aes(x = year, y = cpi)) +
  geom_line(color = 'blue') +
  labs(title = "Time Series of CPI from 1952 to 2023",
       x = "Time",
       y = "CPI") +
  theme_minimal()
```



- d. inflation rate (inf)

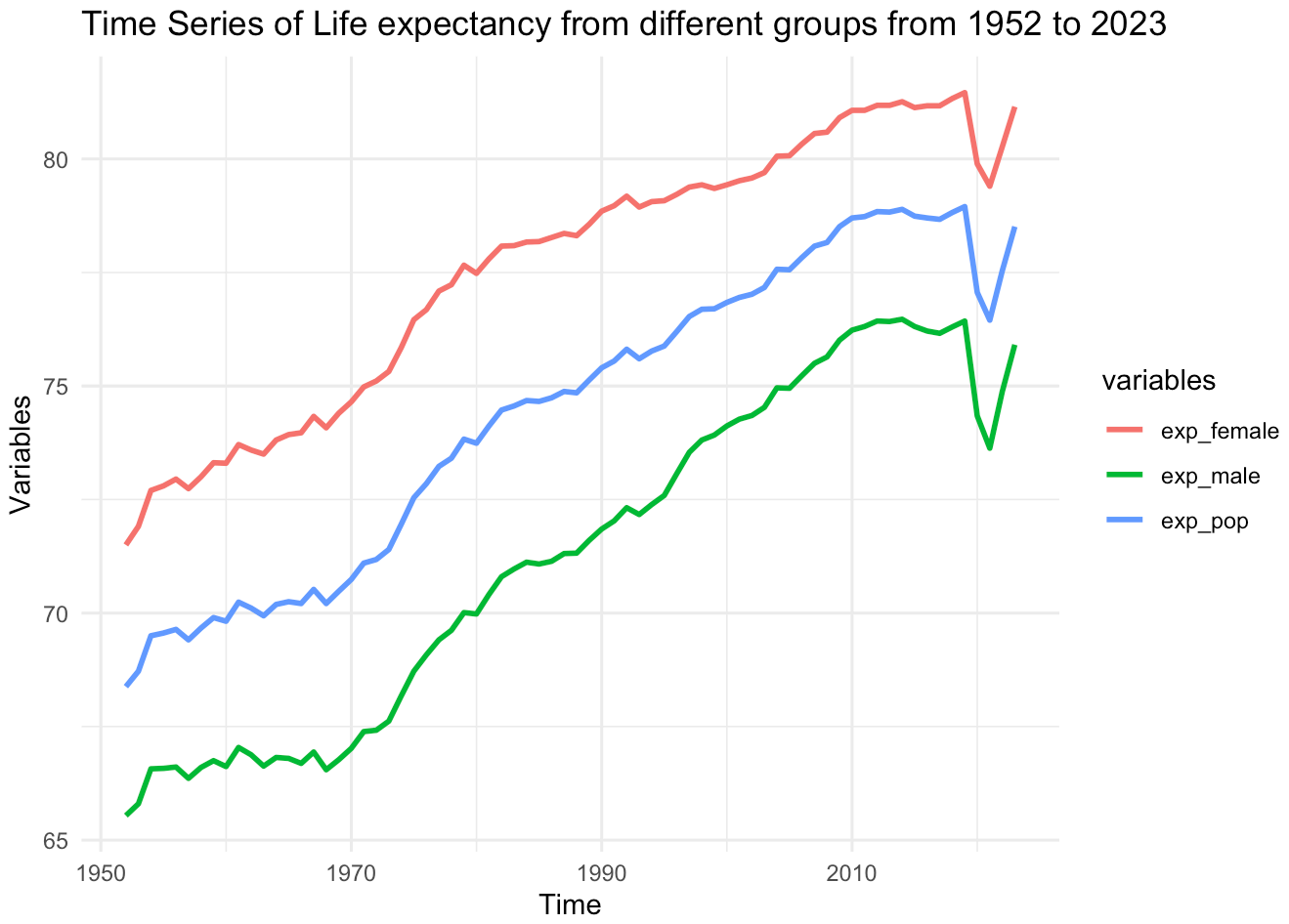
```
ggplot(data_life, aes(x = year, y = inf)) +
  geom_line(color = 'blue') +
  labs(title = "Time Series of inflation from 1952 to 2023",
       x = "Time",
       y = "Inflation") +
  theme_minimal()
```



- e. female, male, and population life expectancy (exp_female, exp_male, and exp_pop) all on the same plot.

```
ggplot(data_life, aes(x = year)) +
  geom_line(aes(y = exp_female, color = 'exp_female'), size = 1) +
  geom_line(aes(y = exp_male, color = 'exp_male'), size = 1) +
  geom_line(aes(y = exp_pop, color = 'exp_pop'), size = 1) +
  labs(title = "Time Series of Life expectancy from different groups from 1952 to 2023",
       x = "Time",
       y = "Variables",
       color = "variables") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



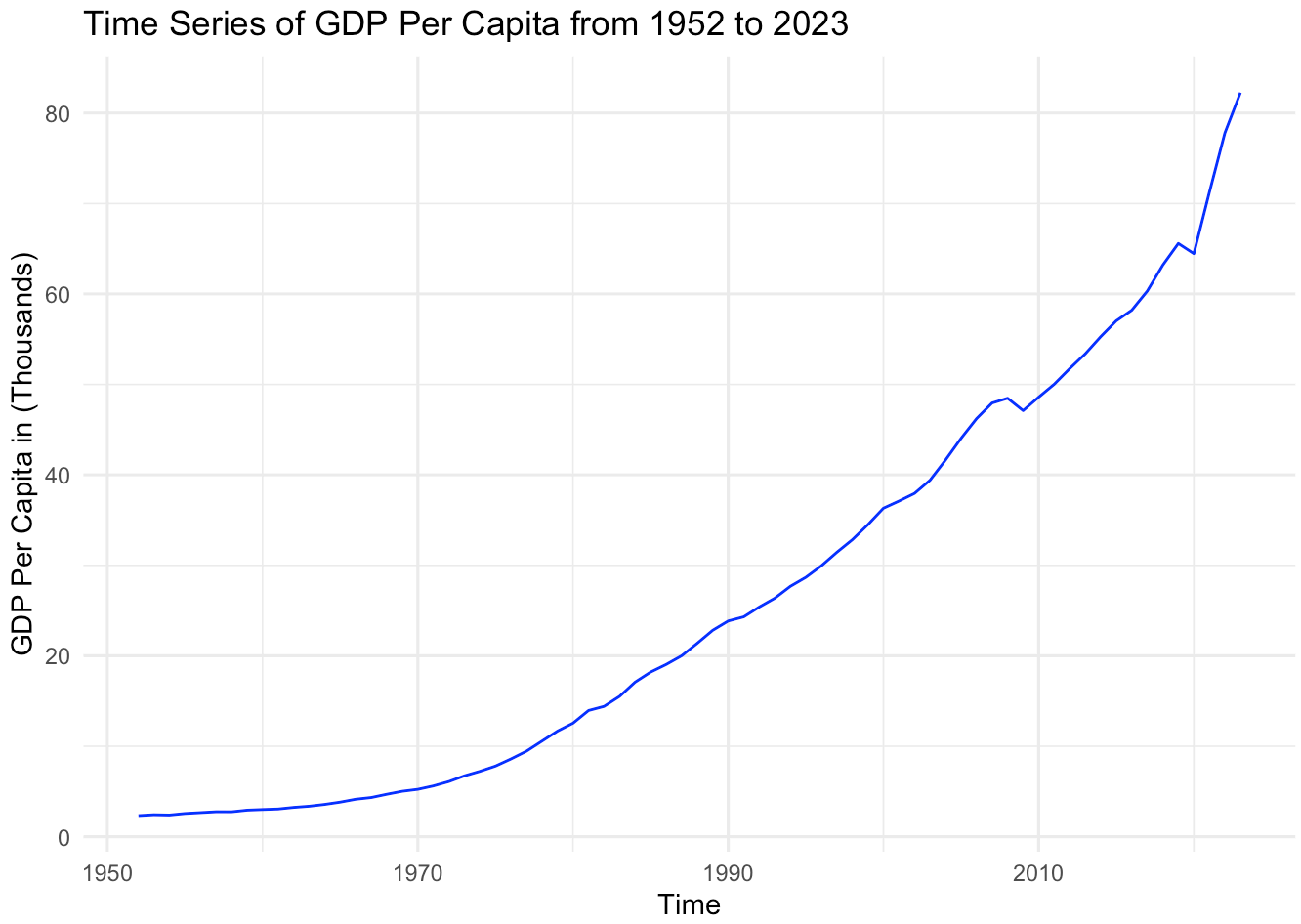
4. Autocorrelation occurs when variables are subject to similar “shocks” which makes them correlated over time. This is not exclusive to being between variables, and can happen with variables being correlated with their own past values. The final graph, with exp_female, exp_male, and exp_pop, appears to suggest that there is some autocorrelation due to its smooth upward trend, minus 2020 which has a dip, likely as a result of covid. This trend is similar to population, gdp, and cpi charts, which indicates potential autocorrelation as well.

5. Missing from this dataset is gdp per capital, which we will call gdppc and create by mutating the data to add a variable. I;ll then add a time series plot similar to the others.

```
data_life = data_life %>%
  mutate(gdppc = (gdp/pop)*100)

tibble(data_life)
```

```
## # A tibble: 72 × 9
##     year exp_female exp_male exp_pop   pop   gdp   cpi    inf gdppc
##    <int>      <dbl>    <dbl>   <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>
##  1  1952       71.5     65.5    68.4  157.  3.67  26.6  2.28   2.33
##  2  1953       71.9     65.8    68.7  160.  3.89  26.8  0.759  2.43
##  3  1954       72.7     66.6    69.5  163.  3.91  26.9  0.361  2.40
##  4  1955       72.8     66.6    69.6  166.  4.25  26.8 -0.257  2.57
##  5  1956       73.0     66.6    69.6  169.  4.49  27.2  1.47   2.66
##  6  1957       72.7     66.4    69.4  172.  4.74  28.1  3.39   2.76
##  7  1958       73       66.6    69.7  175.  4.81  28.9  2.73   2.75
##  8  1959       73.3     66.8    69.9  178.  5.22  29.2  0.932  2.93
##  9  1960       73.3     66.6    69.8  181.  5.42  29.6  1.49   3.00
## 10  1961       73.7     67.0    70.2  184.  5.62  29.9  1.07   3.06
## # ℹ 62 more rows
```

```
ggplot(data_life, aes(x = year, y = gdppc)) +
  geom_line(color = 'blue') +
  labs(title = "Time Series of GDP Per Capita from 1952 to 2023",
       x = "Time",
       y = "GDP Per Capita in (Thousands)") +
  theme_minimal()
```



6. The trend is very similar, with an identical dip around 2020. I would say that the concerns for autocorrelation were not fully addressed. It looks like there is still some issues although the curve has smoothed out quite a bit.

7. Starting by estimating a static model, life expectancy regressed on gdp per capita and inflation rate

```
q7staticmodel = lm(exp_pop ~ gdppc + inf, data = data_life)
summary(q7staticmodel)
```

```
##
## Call:
## lm(formula = exp_pop ~ gdppc + inf, data = data_life)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4619 -0.9387  0.6278  1.1218  1.6466
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 70.448648   0.369797 190.506   <2e-16 ***
## gdppc        0.137931   0.007816  17.648   <2e-16 ***
## inf          0.102839   0.064017   1.606    0.113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.451 on 69 degrees of freedom
## Multiple R-squared:  0.8194, Adjusted R-squared:  0.8142
## F-statistic: 156.6 on 2 and 69 DF,  p-value: < 2.2e-16
```

8. Interpreting the results, it looks like both gdppc and inflation have positive effects on life expectancy of the population, with another thousand in gdp resulting in a .138 year increase in lifespan and with inflation resulting in a .103 years increase on lifespan. The P values support gdppc, but not inf as having significant effects.

9. gdppc and inflation make sense as variables, except the inflation rate is just a function of cpi, and using real gdp would eliminate that variable. Using gdppc was good, but leaves out the real gdp, so while the variables make sense, they may not account for the simply increasing inflation rates. It might also be good to add population into the rate, which indicates how gdppc might be increasing/decreasing compared to gdp. These two are likely connected.

10. Adding population into the regression.

```
q10staticmodel = lm(exp_pop ~ gdppc + inf + pop, data = data_life)
summary(q10staticmodel)
```

```
##
## Call:
## lm(formula = exp_pop ~ gdppc + inf + pop, data = data_life)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0826 -0.1546  0.1965  0.4483  1.1022
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 53.271438   1.139716  46.741  < 2e-16 ***
## gdppc       -0.082509   0.014926  -5.528 5.56e-07 ***
## inf         -0.031115   0.031894  -0.976    0.333
## pop          0.094100   0.006168  15.257  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6952 on 68 degrees of freedom
## Multiple R-squared:  0.9592, Adjusted R-squared:  0.9574
## F-statistic: 532.6 on 3 and 68 DF,  p-value: < 2.2e-16
```

11. Including population drastically changed the results from question 7's regression. It looks now like high gdppc and inflation have a negative impact on life expectancy, with population having a positive effect, of .094 years per million added. I think we have exposed an endogeniety, or omitted variable bias, which indicates that the population size has the largest effect on life expectancy, however high inflation and gdppc have higher negative effects.

12. Estimating a model that interacts with time by adding a lagged variable for gdppc and inflation.

```
q12staticmodel = lm(exp_pop ~ gdppc + inf + lag(gdppc, 1) + lag(inf,1), data = data_lif
e)
summary(q12staticmodel)
```

```
##
## Call:
## lm(formula = exp_pop ~ gdppc + inf + lag(gdppc, 1) + lag(inf,
##     1), data = data_life)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2250 -0.8477  0.0976  1.1329  1.8906
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   70.20084    0.36231 193.760  < 2e-16 ***
## gdppc         -0.35172    0.17917  -1.963  0.05386 .
## inf            0.05818    0.10672   0.545  0.58746
## lag(gdppc, 1)  0.50978    0.18682   2.729  0.00814 **
## lag(inf, 1)    0.13777    0.09930   1.387  0.16997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.334 on 66 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.8471, Adjusted R-squared:  0.8378
## F-statistic:  91.4 on 4 and 66 DF,  p-value: < 2.2e-16
```

13. Estimating the total effect of gdppc on life expectancy is basically the summed effect of the lagged and static variables.

```
total_effect13 = -0.35172 + 0.50978
print(total_effect13)
```

```
## [1] 0.15806
```

the total effect is .15806 years added per thousand added of gdp per capita.

14. The assumptions for OLS to be unbiased in 8 and 10 are less than that of 12. In 8 and 10, we need to assume that there is linearity, no presence of endogeniety, homeoskedasticity, or multicolinarity. In 12, we also need to have no autocorrelation, which in time series regressions shows up when the variables are correlated across time. This causes massive issues in dynamic models, such as 12.

15. We can expect some autocorrelation among our variables like exp_pop, gdp, and gdppc, where the two are connected through both being related to the health situating in any given year. The omission of real gdp may also cause some endogeniety problems, because inflation may lead to increasing values for everything regardless.

16. In regression 12 in particular there are some serious autocorrelation issues, especially between the errors for variables. This leads to inefficient OLS and unreliable standard errors.

17. We will use model (12)'s residuals to test for first order autocorrelation. I'll outline the steps then follow them.

- 1. Estimating the model
- 2. Record the residuals from our OLS regression

- 3. Regress residuals on an intercept, the explanatory variables, and lagged residuals.
- 4. F (or LM) test for ρ1=ρ2=0

```
# 1. Estimate Model
autocorr_test_reg = lm(exp_pop ~ gdppc + inf + lag(gdppc, 1) + lag(inf, 1), data = data_
life)
# 2. Record residuals
data_life$e = c(NA, residuals(autocorr_test_reg))
# 3. Regress residuals on an intercept, ev, and lagged results.
q17reg_p3 <- lm(
  e ~ lag(gdppc, 1) + lag(inf, 1) + lag(e, 1),
  data = data_life
)
# 4. F test
waldtest(q17reg_p3, c("lag(e)", "lag(e, 1)"))
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): terms specified that
## are not in the model: "lag(e)"
```

```
## Wald test
##
## Model 1: e ~ lag(gdppc, 1) + lag(inf, 1) + lag(e, 1)
## Model 2: e ~ lag(gdppc, 1) + lag(inf, 1)
##   Res.Df Df      F    Pr(>F)
## 1     66
## 2     67 -1 201.46 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

18. Living with autocorrelation is complex. There are several answers, including, (1) Misspecification (2), Serial Robust Standard Errors (Newey-West), (3)FLGS.

- a. Misspecification is similar to heterosekedasticity. Essentially, the model is specified for variables that are not correct, so the errors are not consistent, making them correlated through time.
- b. Newey West standard errors are errors that are robust to specifically serial standard errors being correlated. We did not derive these in class and won't be (beyond this class's scope)
- c. feasible generalized least squares (FGLS) gives us efficient and consistent standard errors in the presence of autocorrelation. We do this in a couple steps: **(1)** Estimate the original (untransformed) model; save residuals, **(2)** Estimate ρ: Regress residuals on their lags (no intercept), **(3)** Estimate the transformed model, plugging in ^ρ for ρ.

19. Adding a lagged variable for the outcome variable in model (12).

```
q19LaggedOutReg = lm(exp_pop ~ gdppc + inf + lag(gdppc) + lag(inf) + lag(exp_pop), data
= data_life)
summary(q19LaggedOutReg)
```

```
##
## Call:
## lm(formula = exp_pop ~ gdppc + inf + lag(gdppc) + lag(inf) +
##     lag(exp_pop), data = data_life)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44121 -0.12359  0.05142  0.13162  0.64140
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.62411    2.07352   0.783  0.43632
## gdppc         0.13814    0.04521   3.055  0.00326 **
## inf          -0.03863    0.02562  -1.508  0.13643
## lag(gdppc)   -0.14205    0.04871  -2.916  0.00486 **
## lag(inf)      0.06544    0.02378   2.752  0.00767 **
## lag(exp_pop)  0.97807    0.02955  33.101  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3181 on 65 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9914, Adjusted R-squared:  0.9908
## F-statistic:  1505 on 5 and 65 DF,  p-value: < 2.2e-16
```

20. Testing (19) for autocorrelation. This follows the same process as q (18), so I will not list out the steps again.

```
autocorr_test_reg_20 = lm(exp_pop ~ gdppc + inf + lag(gdppc, 1) + lag(inf, 1) + lag(exp_
pop, 1), data = data_life)
data_life$e = c(NA, residuals(autocorr_test_reg_20))
q17reg_p3 <- lm(
  e ~ lag(gdppc, 1) + lag(inf, 1) + lag(exp_pop,1 ) + lag(e, 1),
  data = data_life
)

waldtest(q17reg_p3, c("lag(e)", "lag(e, 1)"))
```

```
## Warning in modelUpdate(objects[[i - 1]], objects[[i]]): terms specified that
## are not in the model: "lag(e)"
```

```
## Wald test
##
## Model 1: e ~ lag(gdppc, 1) + lag(inf, 1) + lag(exp_pop, 1) + lag(e, 1)
## Model 2: e ~ lag(gdppc, 1) + lag(inf, 1) + lag(exp_pop, 1)
##   Res.Df Df      F  Pr(>F)
## 1     65
## 2     66 -1 3.7366 0.05759 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p value is well over .05, so we can reject the null hypothesis that there is no first order autocorrelation.

21. OLS being autocorrelated, as a result of a lagged outcome variable, results in OLS being biased for the coefficients as well as no longer the most efficient estimated. Because OLS is biased for β, we have to say it is also biased for the disturbances υ_t.

22. I think it does make sense to include a lagged variable because the likelihood of continued progressive health policy over years is high, suggesting that past policy would affect the future of life expectancy.

23. CPI, and inflation as a result, seems to suggest that the variables are non stationary. I can see this because of the persistent upward trend. GDP and population, as well as life expectancy all violate this trend as well.

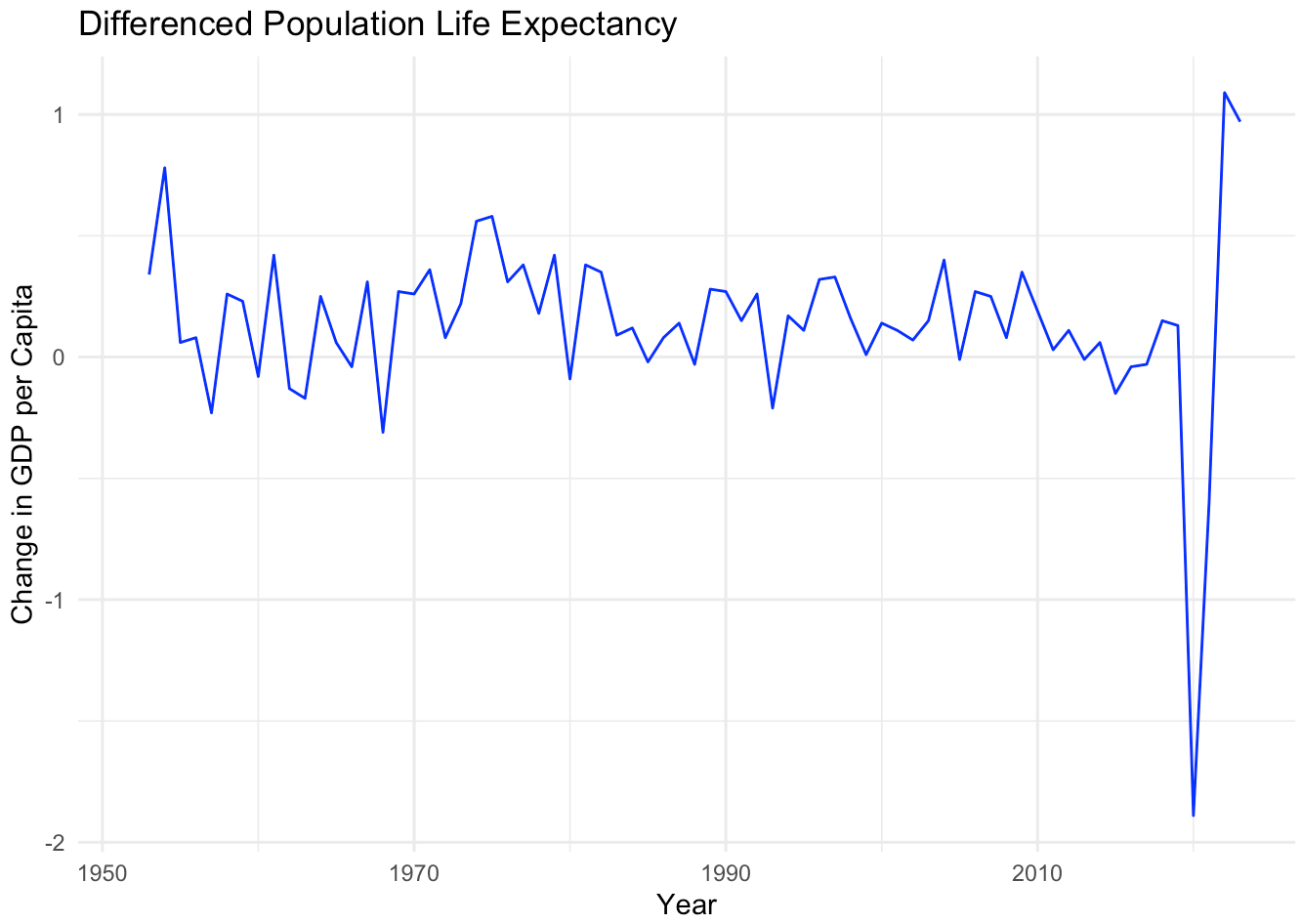
24. Non-stationarity in our data causes a lot of problems. We may get spurious results. We also cannot trust OLS. In order to combat this there are a number of tests to find out what the solutions may be, including calculating the disturbance from the year prior to pull ourselves back to “good behavior”.

25. CPI variable may not be stationary, but Inflation as a variable may have some hope of being stationary. This is because the inflation formula includes a built in ‘lagged’ variable,

26. We need to mutate the variables to create “differenced” versions of exp_pop, gdppc, and inf. We will do this by taking the differences of each with this formula: $\delta x_t = x_t - x_{t-1}$. I will name these with the phrase “diff” ahead of them.

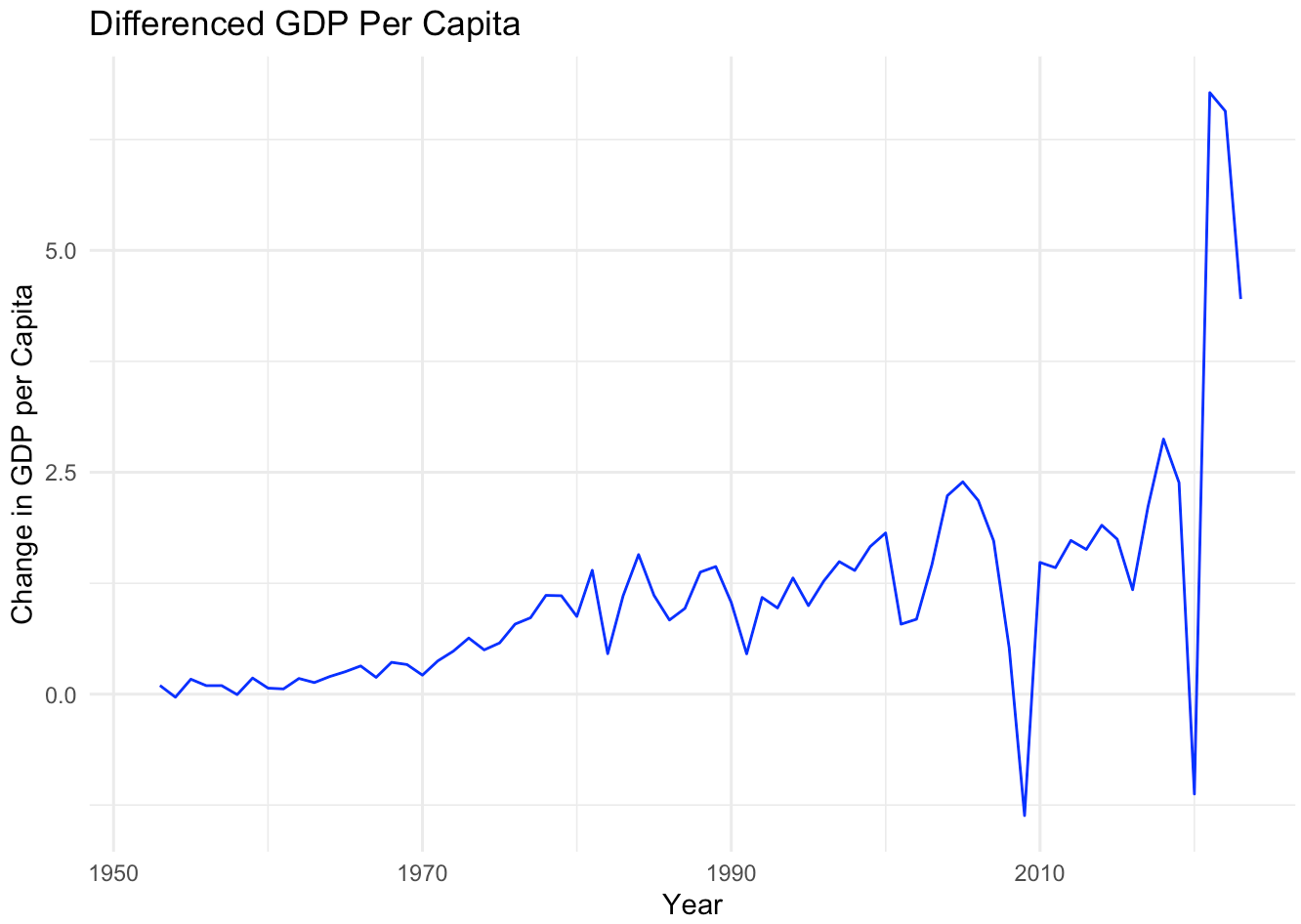
```
data_life = data_life %>%
  mutate(
    diff_exp_pop = exp_pop - lag(exp_pop, 1),
    diff_gdppc = gdppc - lag(gdppc, 1),
    diff_inf = inf - lag(inf,1)
  )

ggplot(data_life, aes(x = year, y = diff_exp_pop)) +
  geom_line(color = 'blue') +
  labs(title = "Differenced Population Life Expectancy",
       x = "Year",
       y = "Change in GDP per Capita") +
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



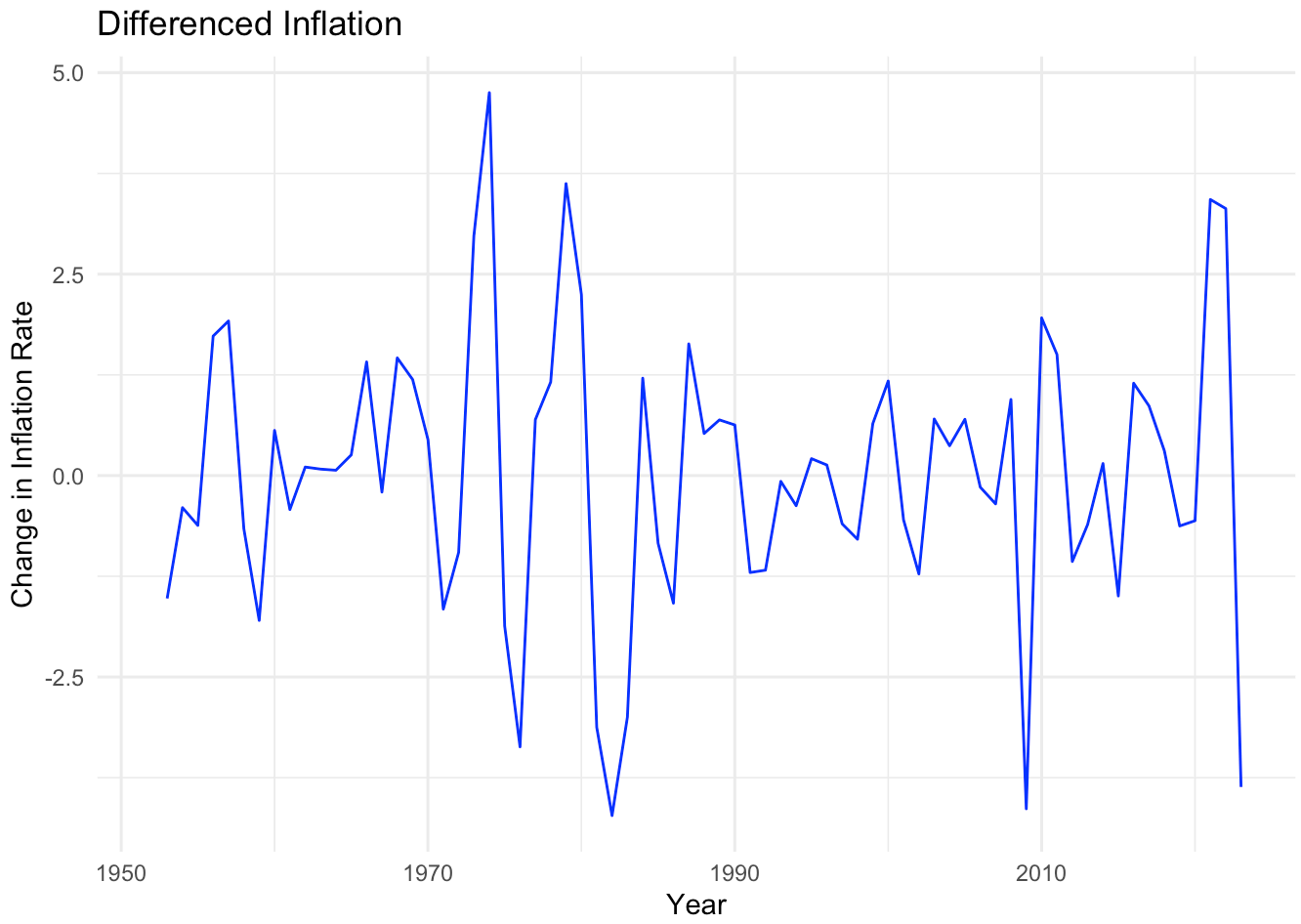

```
ggplot(data_life, aes(x = year, y = diff_gdppc)) +
  geom_line(color = 'blue') +
  labs(title = "Differenced GDP Per Capita",
       x = "Year",
       y = "Change in GDP per Capita") +
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```




```
ggplot(data_life, aes(x = year, y = diff_inf)) +
  geom_line(color = 'blue') +
  labs(title = "Differenced Inflation",
       x = "Year",
       y = "Change in Inflation Rate" ) +
  theme_minimal()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



Examining these time series charts I think that using a differenced variable did help with the stationarity problem. The data looks like it is much more centered around 0, with the exception of GDP Per capita, which still indicates a slight an upward trend.

27. We will now estimate the model from question 7 again with the differenced form of the variables.

```
q27diff_model = lm(diff_exp_pop ~ diff_gdppc + diff_inf, data = data_life)
summary(q27diff_model)
```

```
##
## Call:
## lm(formula = diff_exp_pop ~ diff_gdppc + diff_inf, data = data_life)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89811 -0.13091  0.00876  0.17063  0.70219
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06556    0.05559   1.179   0.2424
## diff_gdppc   0.06935    0.03308   2.096   0.0398 *
## diff_inf    -0.03668    0.02469  -1.485   0.1421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3489 on 68 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.07381,    Adjusted R-squared:  0.04657
## F-statistic:  2.71 on 2 and 68 DF,  p-value: 0.07375
```

28. Interpreting these results yields again different results. It looks like the only significant result is the diff_gdppc, where a 1000 dollar increase in gdppc results in a .06935 year increase in expected lifespan. Inf is unfortunately not a high enough enough to suggest a significant effect.

29. Accounting for differenced variables increased our p value from ‘pretty much’ zero up to around .039, which is significant at the 5% level, still. It looks like the overall the differentiated model (27) is addressing autocorrelation, while regression in (7) is likely spurious, or unfounded.

30. The ‘best’ model we estimated is going to be the one with stationarity accounted for, the lm from question 27. This is because it gets rid of a problem that is present in all our other models.