Company Growth

```
knitr::opts_chunk$set(message = FALSE, warning = FALSE)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```
library(vroom)
```

```
Attaching package: 'vroom'

The following objects are masked from 'package:readr':

    as.col_spec, col_character, col_date, col_datetime, col_double,
    col_factor, col_guess, col_integer, col_logical, col_number,
    col_skip, col_time, cols, cols_condense, cols_only, date_names,
    date_names_lang, date_names_langs, default_locale, fwf_cols,
    fwf_empty, fwf_positions, fwf_widths, locale, output_column,
    problems, spec
```

```
library(zoo)
```

```
Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```r
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```r
library(splines)
library(prophet)
```

```
Loading required package: Rcpp
Loading required package: rlang

Attaching package: 'rlang'

The following objects are masked from 'package:purrr':

    %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
    flatten_raw, invoke, splice
```

```r
library(yardstick)
```

```
Attaching package: 'yardstick'

The following object is masked from 'package:forecast':

    accuracy

The following object is masked from 'package:vroom':

    spec

The following object is masked from 'package:readr':

    spec
```

```r
data <- vroom('CompanyGrowth.csv')
```

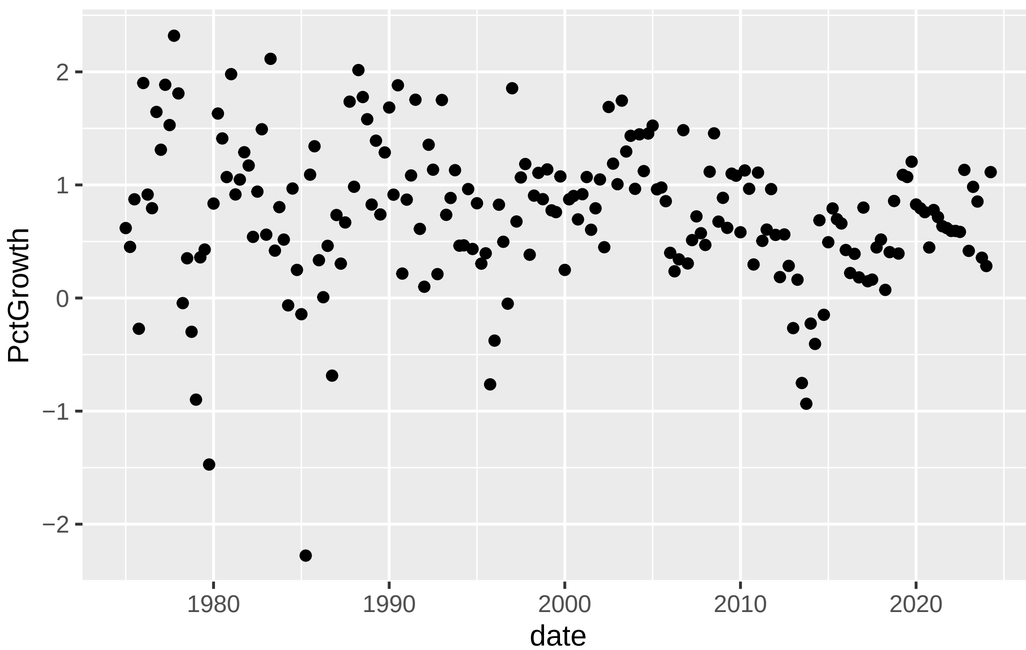
```
Rows: 198 Columns: 7
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (7): PctGrowth, Income, Production, Savings, Unemployment, Year, Qrtr

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.
```

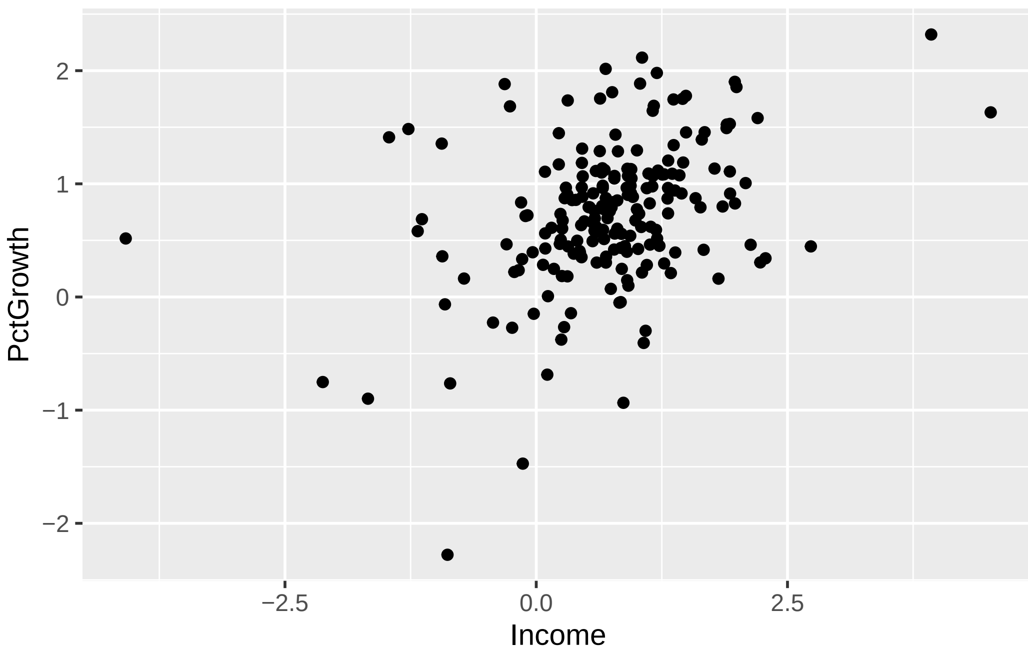
```r
data <- data %>%
  mutate(date = as.Date(as.yearqtr(paste(Year, Qrtr, sep = "-"))))
```

EDA

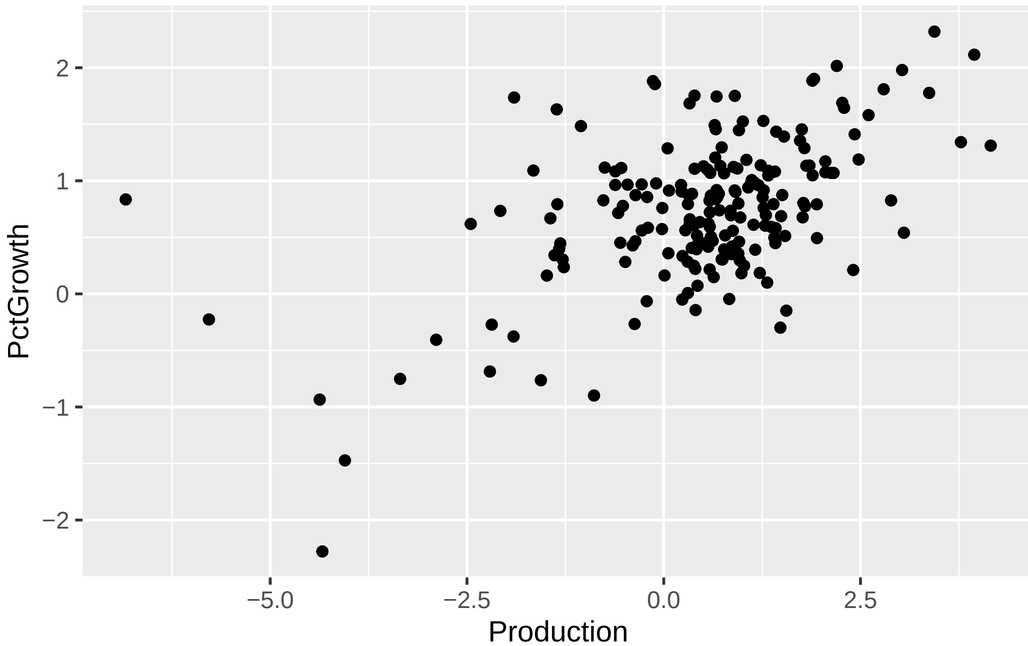
```r
# growth stays about the same throughout
ggplot(data=data, aes(x=date, y=PctGrowth)) +
  geom_point()
```



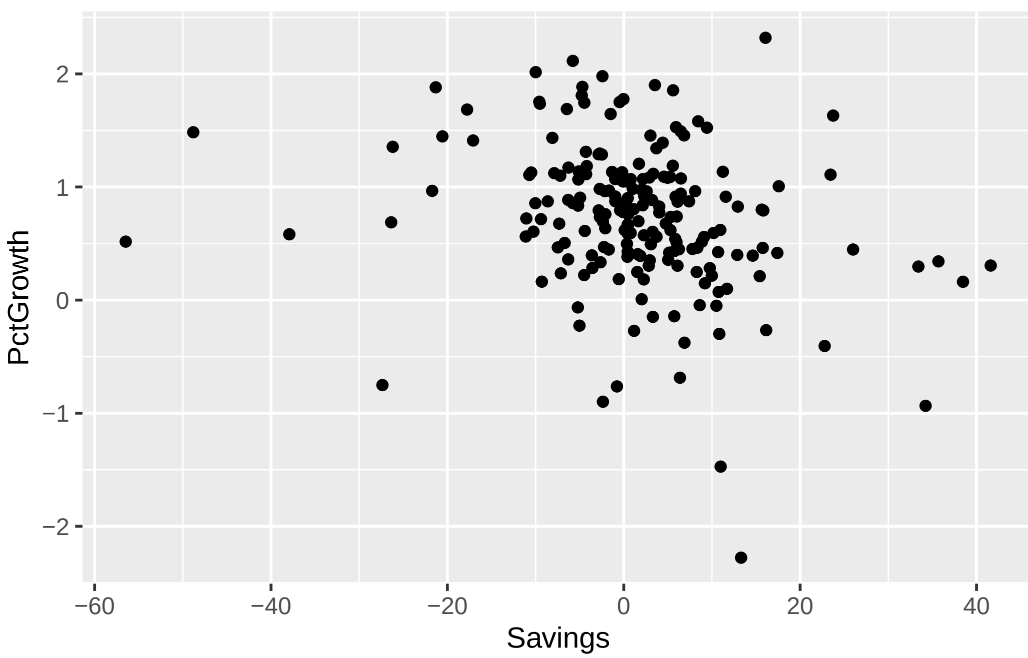
```r
# Change in Income
ggplot(data=data, aes(x=Income, y=PctGrowth)) +
  geom_point()
```



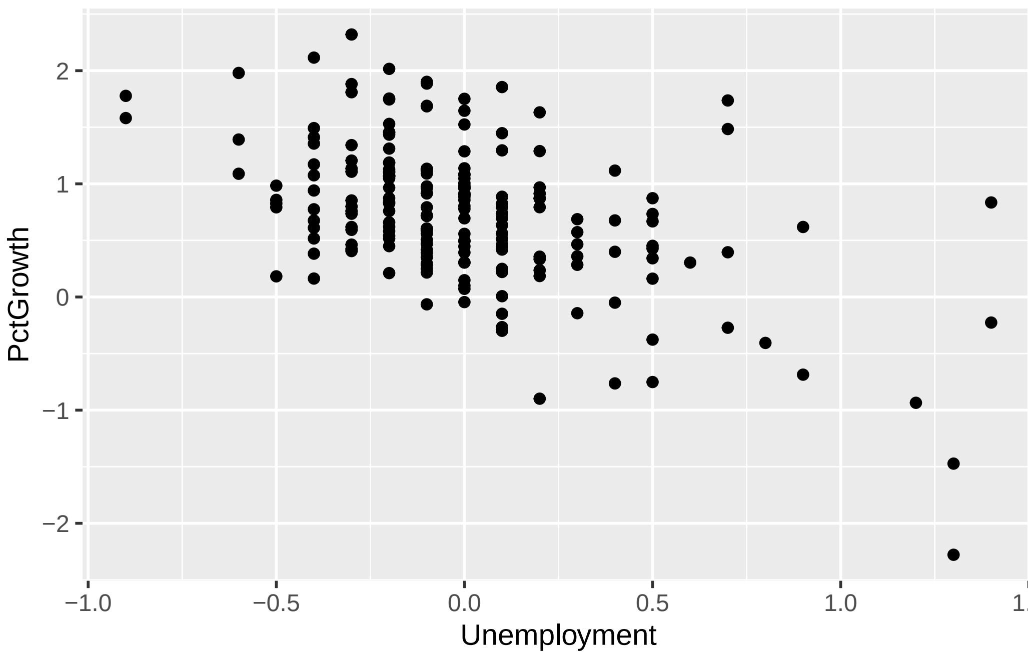

```r
# Change in Production
ggplot(data=data, aes(x=Production, y=PctGrowth)) +
  geom_point()
```




```r
# Change in Savings
ggplot(data=data, aes(x=Savings, y=PctGrowth)) +
  geom_point()
```



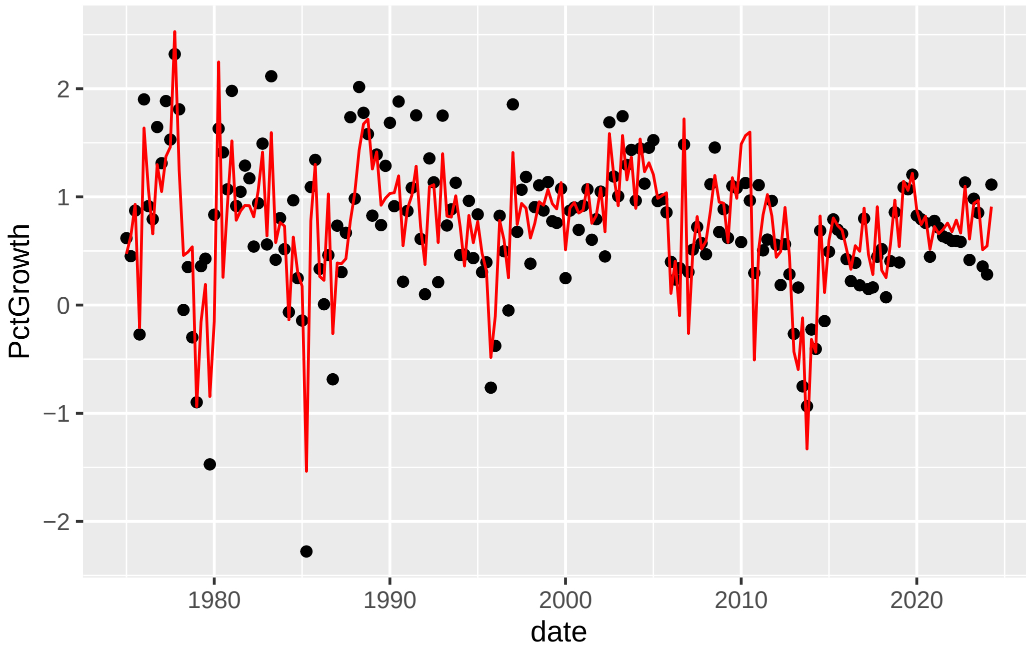
```r
# Change in Unemployment
ggplot(data=data, aes(x=Unemployment, y=PctGrowth)) +
  geom_point()
```



1. Describe the model you are currently using, how it performs for predictions and what the 2025 projections are under this model. Be sure to describe how you obtained the 2025 predictions. Company Growth Forecasting Problem Background Growth = 0 + 1Income + 2Production + 3Savings + 4Unemployment + i i iid N (0, 2)

```r
# basic linear model
growth.lm <- lm(PctGrowth ~ Income + Production + Savings + Unemployment, data=data)

data %>%
  ggplot(aes(x = date, y = PctGrowth)) +
  geom_point() +
  geom_line(aes(y = growth.lm$fitted.values), col = 'red')
```




```r
# lm models for each predictor
new_data <- data.frame(Year = 2025,
                       Qrtr = c(1,2,3,4))
income.lm <- lm(Income~Year+Qrtr, data=data)
income_lm_preds <- predict(income.lm, newdata = new_data)
production.lm <- lm(Production~Year+Qrtr, data=data)
production_lm_preds <- predict(production.lm, newdata = new_data)
savings.lm <- lm(Savings~Year+Qrtr, data=data)
savings_lm_preds <- predict(savings.lm, newdata = new_data)
unemployment.lm <- lm(Unemployment~Year+Qrtr, data=data)
unemployment_lm_preds <- predict(unemployment.lm, newdata = new_data)

# predict growth
growth_new_data <- data.frame(Income = income_lm_preds,
                              Production = production_lm_preds,
                              Savings = savings_lm_preds,
                              Unemployment = unemployment_lm_preds)
lm_preds <- predict(growth.lm, growth_new_data)
lm_preds
```

```
        1         2         3         4
0.6223706 0.6520897 0.6818088 0.7115279
```

```r
## how well does our model predict
# r^2
lm_r2 <- summary(growth.lm)$r.squared
lm_r2
```

```
[1] 0.7682829
```

```r
# rmse
lm_rmse <- sqrt(mean(residuals(growth.lm)^2))
lm_rmse
```

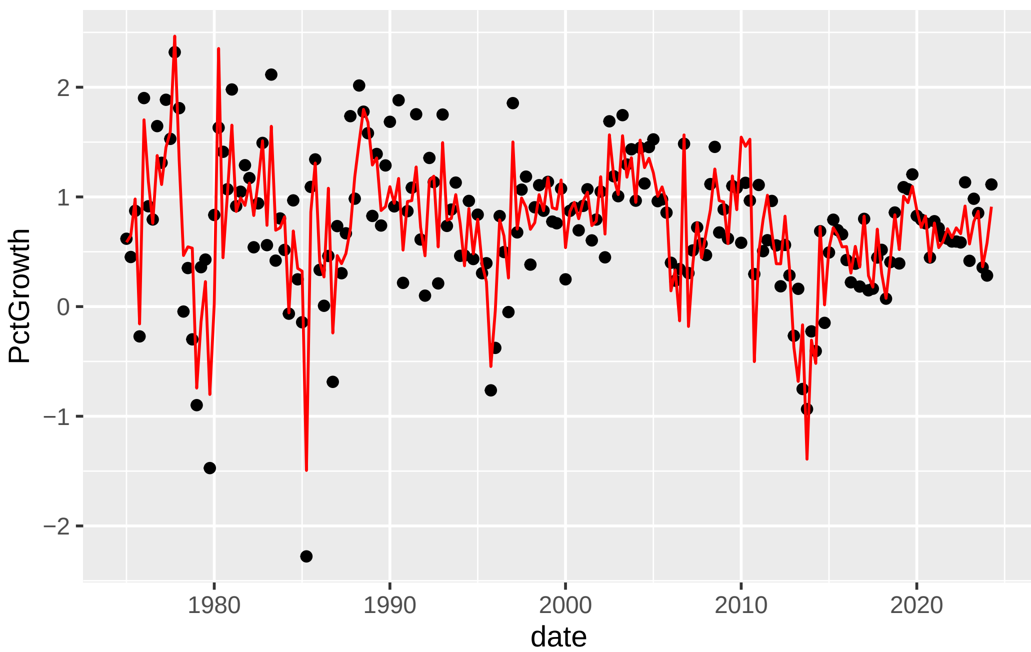
```
[1] 0.3062718
```

2. Describe at least 2 alternative models that could be used to do the projections and emphasize how they differ from the current method.

```r
# new linear model accounting for date features and maybe with spline and what not
data_date_features <- data %>%
  mutate(Year = year(date),
         Month = month(date, label = TRUE),
         DOW = wday(date, label = TRUE),
         Numeric_date = decimal_date(date))

spline_model <- lm(PctGrowth ~ bs(Numeric_date, knots = c(1975, 1990, 2005, 2020),
                                  degree = 3) +., data = data_date_features)

data_date_features %>%
ggplot(aes(x = date, y = PctGrowth)) +
geom_point() +
geom_line(aes(y = spline_model$fitted.values), col = 'red')
```




```
future_dates <- seq(from = as.yearqtr("2025 Q1"), to = as.yearqtr("2025 Q4"), by = 1/4)
future_dates <- as.Date(as.yearqtr(future_dates), frac = 0)
future_features <- tibble(date = future_dates) %>%
  mutate(Year = year(date),
    Month = month(date, label = TRUE),
    DOW = wday(date, label = TRUE),
    Numeric_date = decimal_date(date),
    Qrtr = quarter(date, with_year = FALSE, fiscal_start = 1),
    Income = income_lm_preds,
    Production = production_lm_preds,
    Savings = savings_lm_preds,
    Unemployment = unemployment_lm_preds)

spline_preds <- predict(spline_model, newdata = future_features)
spline_preds
```

```
        1         2         3         4
0.7421330 0.6718305 0.7534381 0.7538485
```

```
spline_r2 <- summary(spline_model)$r.squared
spline_r2
```

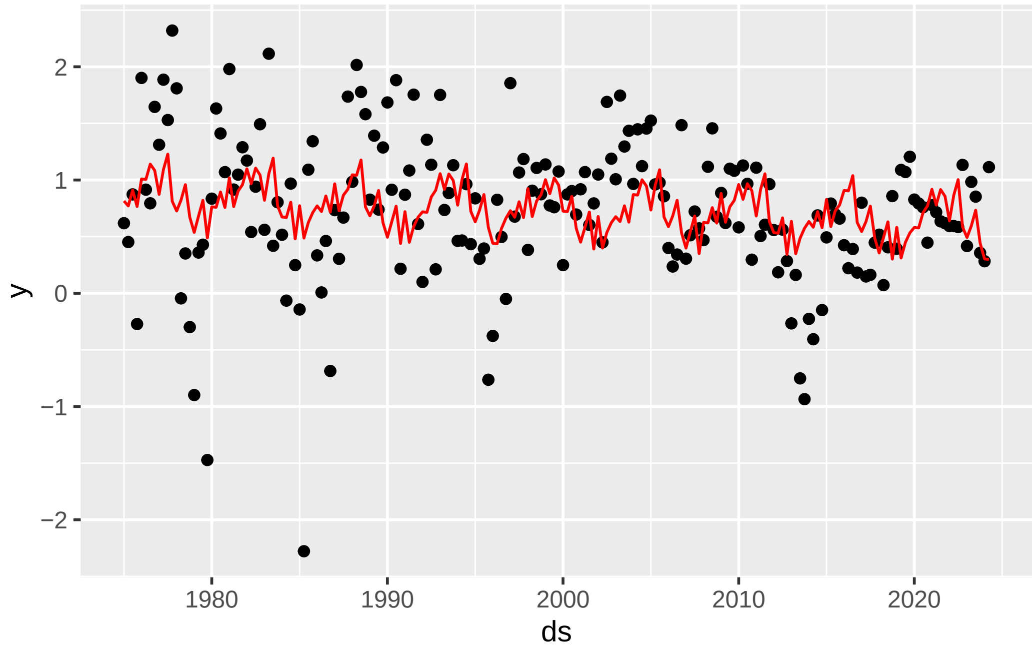
```
[1] 0.7843894
```

```r
spline_rmse <- sqrt(mean(residuals(spline_model)^2))
spline_rmse
```

```
[1] 0.2954357
```

```r
# facebook prophet (maybe or maybe not this one because it won't account for other varia
prophet_df <- data %>%
  rename(y = PctGrowth, ds = date) %>%
  dplyr::select(y, ds)
prophet_model <- prophet_model <- prophet(prophet_df, yearly.seasonality = TRUE,
                                          weekly.seasonality = TRUE,
                                          daily.seasonality = TRUE)
prophet_fitted <- predict(prophet_model) %>% pull(yhat)

prophet_df %>%
  ggplot(aes(x = ds, y = y)) +
  geom_point() +
  geom_line(aes(y = prophet_fitted), col = 'red')
```




```r
prophet_future <- future_features %>%
  rename(ds = date) %>%
  dplyr::select(ds)
prophet_preds <- predict(prophet_model, prophet_future)$yhat
```

```r
prophet_preds
```

```
[1] 0.5905687 0.5327727 0.6693263 0.5297125
```

```r
prophet_r2 <- 1 - sum((prophet_df$y - prophet_fitted)^2) /
  sum((prophet_df$y - mean(prophet_df$y))^2)
prophet_r2
```

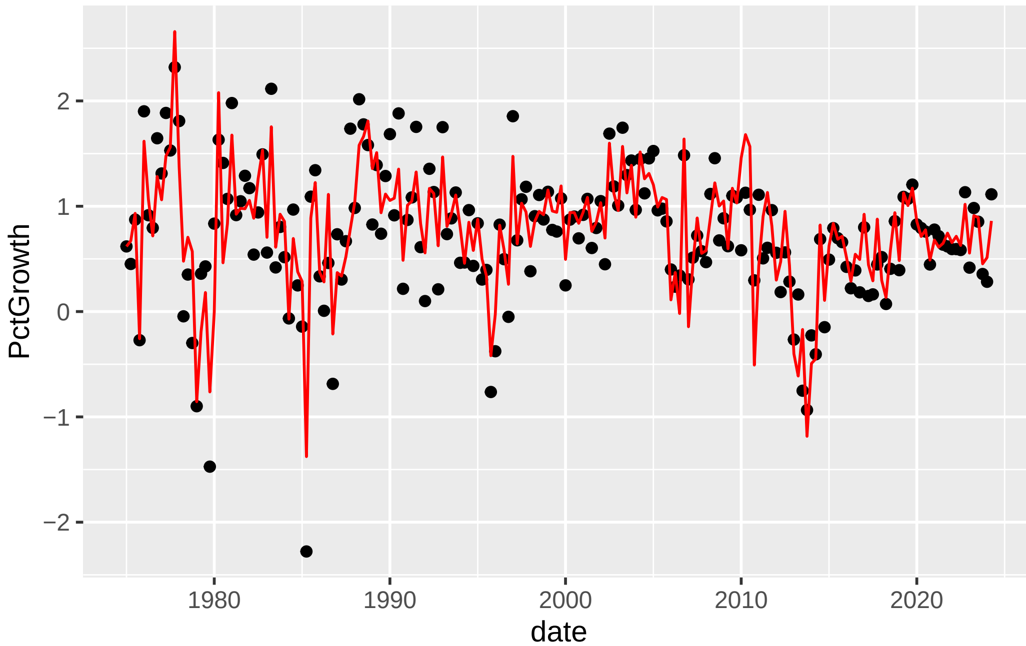
```
[1] 0.1085342
```

```r
prophet_rmse <- sqrt(mean((prophet_df$y - prophet_fitted)^2))
prophet_rmse
```

```
[1] 0.6007314
```

```r
# ARIMA
## Define response as time series object with a frequency S
ts_data <- ts(data=data$PctGrowth, frequency=4)
X_noInt <- model.matrix(growth.lm)[,-1]
arima_model <- auto.arima(y=ts_data,xreg=X_noInt)

data %>%
  ggplot(aes(x = date, y = PctGrowth)) +
  geom_point() +
  geom_line(aes(y = fitted(arima_model)), col = 'red')
```




```
x_future <- model.matrix(growth.lm, data = data)[,-1]
arima_preds <- forecast(arima_model, xreg = x_future, level = 95)$mean


arima_r2 <- cor(fitted(arima_model), data$PctGrowth)^2
arima_r2
```

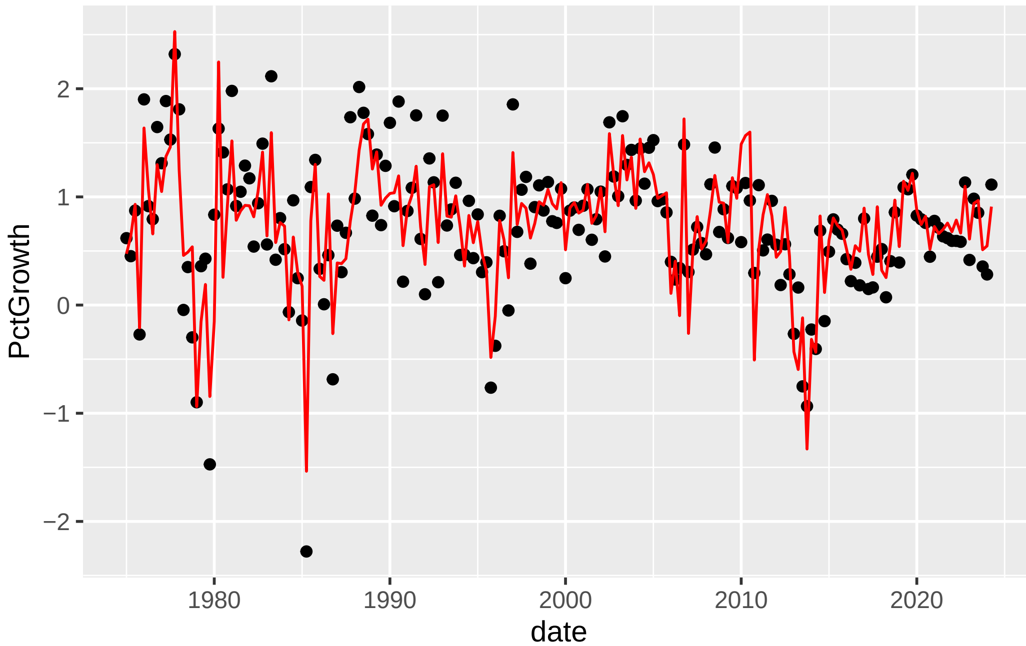
```
[1] 0.7777633
```

```
arima_rmse <- rmse_vec(data$PctGrowth, fitted(arima_model))
arima_rmse
```

```
[1] 0.3011327
```

3. Compare these 2 new methods to the current method in terms of ability to predict and the 2025 projections. Be sure to describe how you obtained the 2025 predictions. (predict for all variables, not just the response)

```
# Basic lm model
data %>%
  ggplot(aes(x = date, y = PctGrowth)) +
  geom_point() +
  geom_line(aes(y = growth.lm$fitted.values), col = 'red')
```




```r
print(paste('R2 value:', lm_r2))
```

```
[1] "R2 value: 0.768282945629248"
```

```r
print(paste('RMSE:', lm_rmse))
```

```
[1] "RMSE: 0.306271757028289"
```

4. The board is particularly interested to know how Income, Production, Savings and Unemployment relate to company growth. Use your preferred method (among the 3 or more you consider here) to illustrate this relationship.