EDA 1997 CO2 Data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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 become errors
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(patchwork)

library(lubridate)

library(tsibble)
```

```
##
```

```
## Attaching package: 'tsibble'
##
## The following object is masked from 'package:lubridate':
##
##     interval
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, union
```

```
library(feasts)
```

```
## Loading required package: fabletools
```

```
library(forecast)
```

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

```
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following object is masked from 'package:tsibble':
##
##     index
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(nycflights13)
library(blsR)

library(Matrix)
```

```
##
## Attaching package: 'Matrix'
```

```
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following object is masked from 'package:tsibble':
##
##     key
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(stats)
library(fable)
```

```
data <- datasets::co2

data <- data %>%
  as_tsibble(data)
```

EDA

1. Description of how, where, and why the data is generated

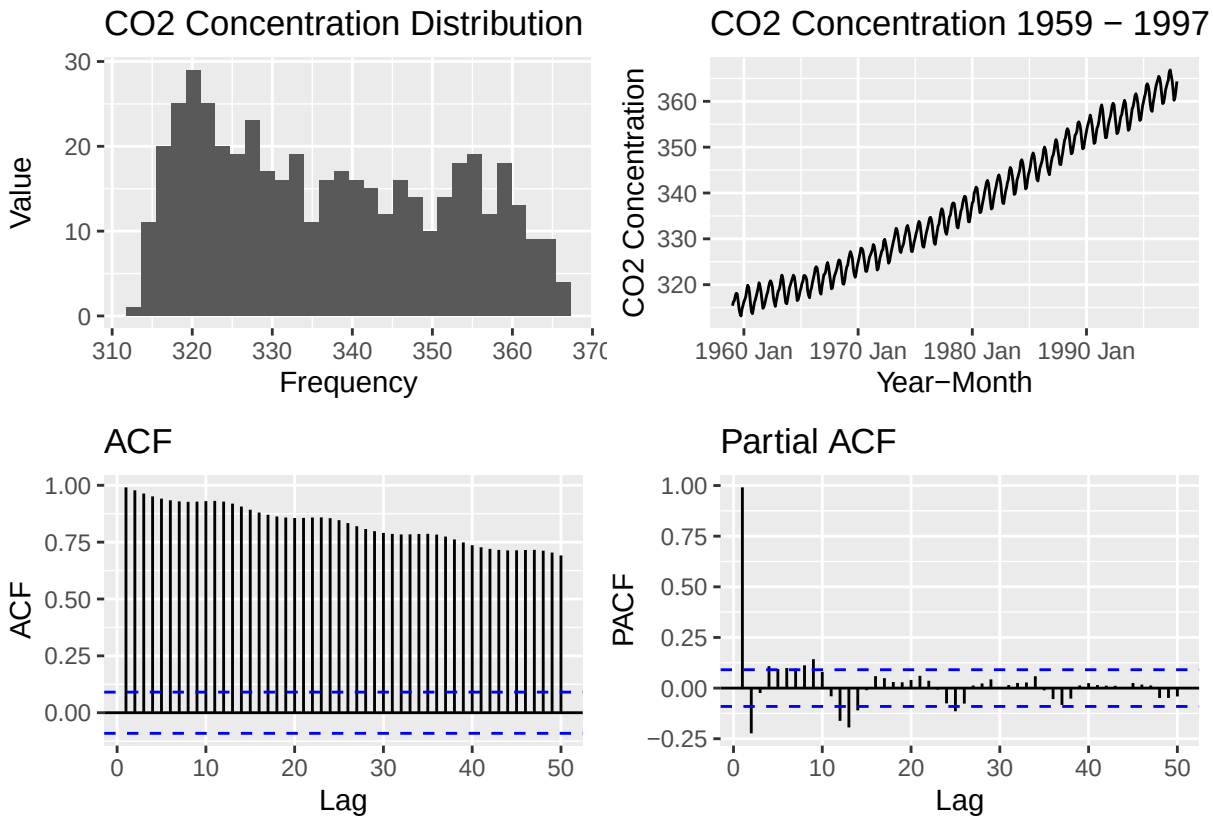
The CO2 data set consists of 468 observations. Each observation represents the monthly total atmospheric concentration of CO2, measured in parts per million (ppm) and collected at the Mauna Loa Observatory in Hawaii. The data ranges from January 1959 to December 1997. The data is originally

sourced from the Scripps institute and was collected as part of the Scripps CO2 Program. Observations for February, March, and April 1964 were unavailable so the values in the data et were generated via linear interpolation between the observations for January and May 1964.

2. Investigation of trend, seasonal and irregular elements

```
# Distribution
dist <- ggplot(data, aes(x = value)) +
  geom_histogram() +
  ylab("Value") +
  xlab("Frequency") +
  ggtitle("CO2 Concentration Distribution")

# Time series data
time_series <- ggplot(data, aes(x=index, y=value)) +
  geom_line() +
  ylab("CO2 Concentration") +
  xlab("Year-Month") +
  ggtitle("CO2 Concentration 1959 - 1997")

# ACF
acf <- ggAcf(data$value, lag.max = 50) +
  ggtitle("ACF")

# PACF
pacf <- ggPacf(data$value, lag.max = 50) +
  ggtitle("Partial ACF")

(dist | time_series) / (acf | pacf)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



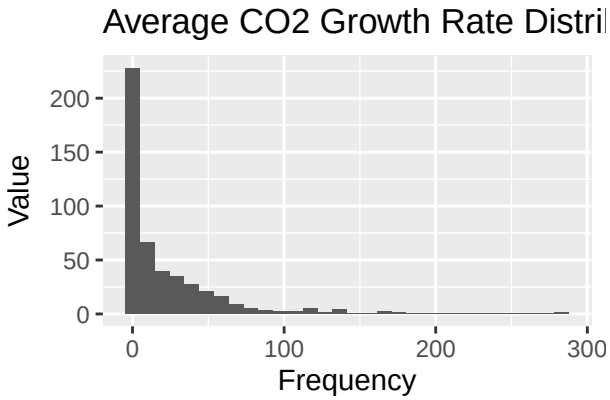
- Non-stationarity but variance stationary? i.e., variance is relatively constant overtime
- Clear from time series plot, ACF, and PACF that there is monthly seasonality present in the data
- Time series and ACF show evidence of a trend in the data: 1. continuously increasing at a consistent rate and 2. Slow decay in ACF

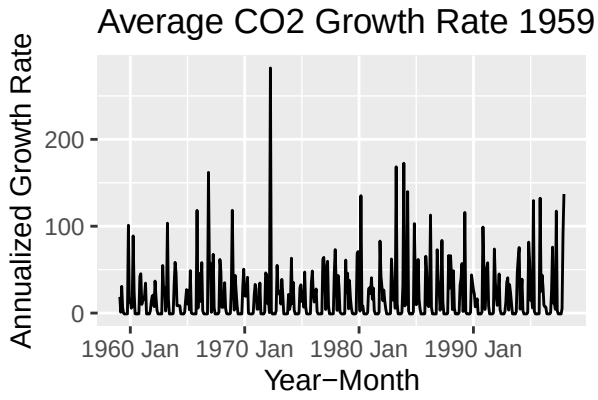
3. Trends in levels and growth rates should be discussed (long-run growth rate as annualized averages)

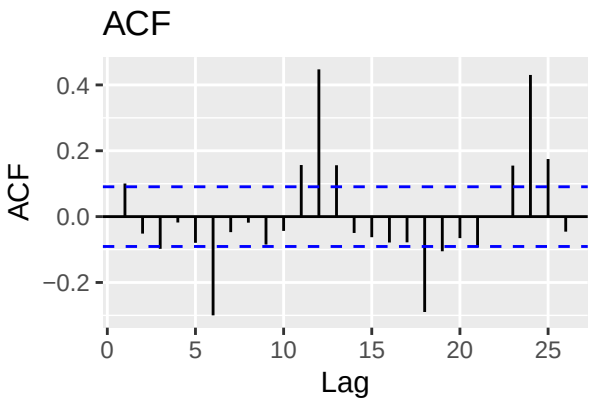
```
annual_growth <- data %>%
  mutate(co2_diff = value - lag(value),
         monthly_growth_rate = co2_diff / value * 100,
         annualized_growth_rate = (1 + monthly_growth_rate)^12 - 1) %>%
  na.omit()

# Distribution
dist <- ggplot(annual_growth, aes(x = annualized_growth_rate)) +
```

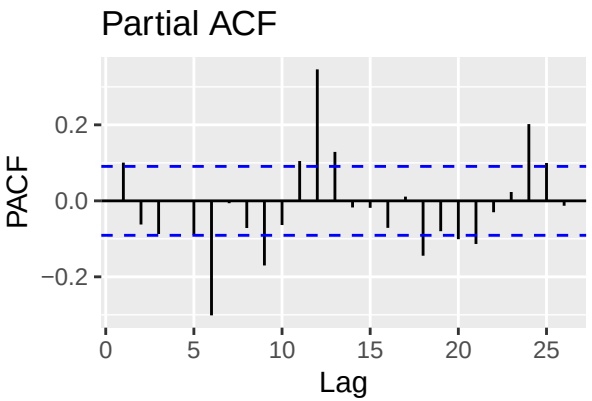
```r
  geom_histogram() +
  ylab("Value") +
  xlab("Frequency") +
  ggtitle("Average CO2 Growth Rate Distribution")

# Time series data
time_series <- ggplot(annual_growth, aes(x=index, y=annualized_growth_rate)) +
  geom_line() +
  ylab("Annualized Growth Rate") +
  xlab("Year-Month") +
  ggtitle("Average CO2 Growth Rate 1959 - 1997")

# ACF
acf <- ggAcf(annual_growth$annualized_growth_rate) +
  ggtitle("ACF")

# PACF
pacf <- ggPacf(annual_growth$annualized_growth_rate) +
  ggtitle("Partial ACF")

(dist | time_series) / (acf | pacf)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```









- Growth rates follow a white noise process

i.e., shows no autocorrelation

Next steps

- How has the seasonal cycle of CO2 concentrations changed between 1958 and 1997? Is there an identifiable pattern that will persist into the future?
- Is atmospheric CO2 concentration predictable?

De-trending? Seasonal adjustment? Differencing?

KSPSS test

Part 2a

```
data$month_since_start <- as.numeric(index(data) - min(index(data))) + 1

lm_model <- lm(value ~ month_since_start, data=data)
lm_model
```

```
##
## Call:
## lm(formula = value ~ month_since_start, data = data)
##
## Coefficients:
##       (Intercept)  month_since_start
##           311.503              0.109
```

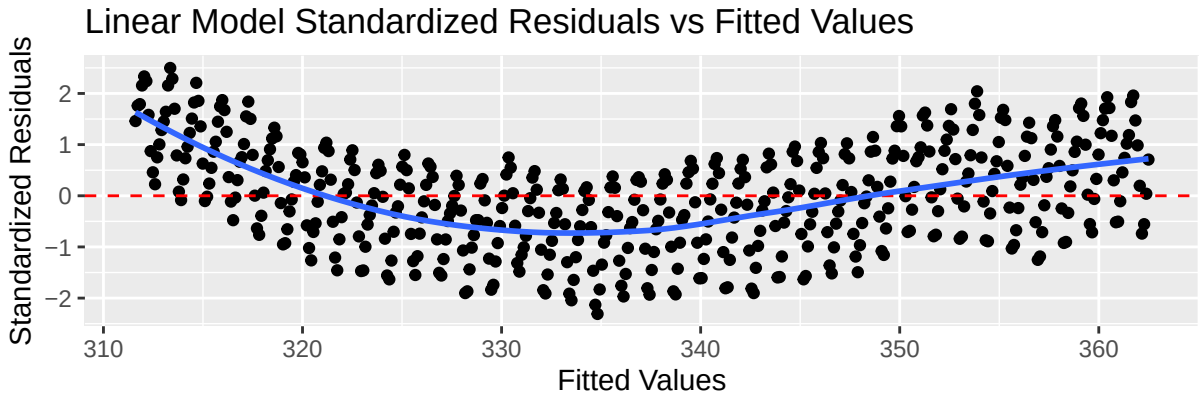
```
quad_model <- lm(value ~ I(month_since_start^2) , data=data)
quad_model
```

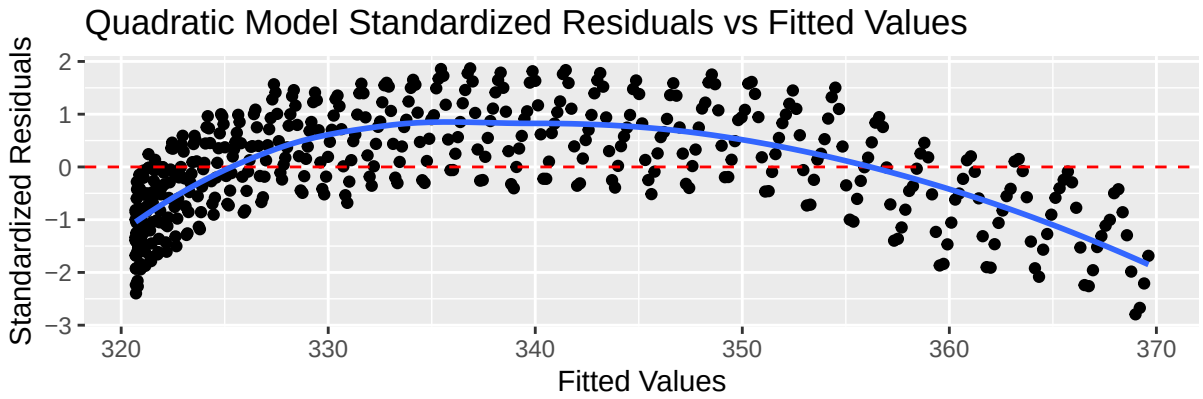
```
##
## Call:
## lm(formula = value ~ I(month_since_start^2), data = data)
##
## Coefficients:
##            (Intercept)  I(month_since_start^2)
##              3.207e+02               2.234e-04
```

```
lm_residual_data <- data.frame(
  predicted_values = fitted(lm_model),
  residuals = rstandard(lm_model)
)

quad_residual_data <- data.frame(
  predicted_values = fitted(quad_model),
  residuals = rstandard(quad_model)
)

lm_resid_plot <- ggplot(lm_residual_data, aes(x = predicted_values, y = residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_smooth(se = FALSE) +
```

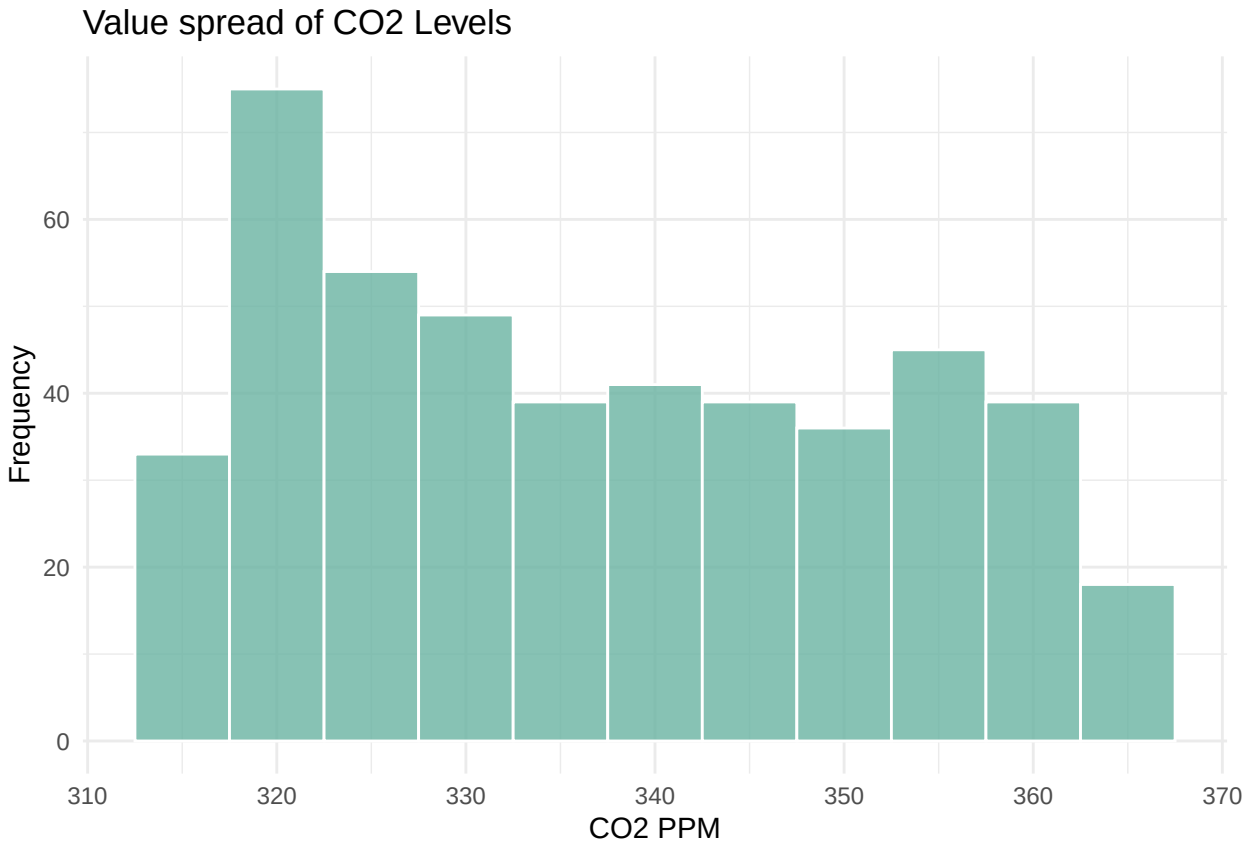
```r
  labs(title = "Linear Model Standardized Residuals vs Fitted Values",
       x = "Fitted Values", y = "Standardized Residuals")

quad_resid_plot <- ggplot(quad_residual_data, aes(x = predicted_values, y = residuals)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_smooth(se = FALSE) +
  labs(title = "Quadratic Model Standardized Residuals vs Fitted Values",
       x = "Fitted Values", y = "Standardized Residuals")

value_hist <- ggplot(data, aes(x = value)) +
  geom_histogram(binwidth = 5, fill = "#69b3a2", color = "white", alpha = 0.8) +
  labs(title = "Value spread of CO2 Levels",
       x = "CO2 PPM",
       y = "Frequency"
       ) +
  theme_minimal() +
  theme(legend.position = "top")

(lm_resid_plot / quad_resid_plot)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```


Linear Model Standardized Residuals vs Fitted Values
Standardized Residuals
2
1
0
−1
−2
310
320
330
340
350
360
Fitted Values


Quadratic Model Standardized Residuals vs Fitted Values
Standardized Residuals
2
1
0
−1
−2
−3
320
330
340
350
360
370
Fitted Values

```
value_hist
```



The linear model produces residuals that are homoskedastic however the linearity assumption is not verified as the mean residual value deviates from zero. The quadratic model has similar homoskedastic errors but is similarly plagued by the lack of a linear relationship between the fitted and residual values. Taking a logarithm is not the appropriate decision for the data given the spread of response values. Having a small range from ~300 to ~400 with values relatively uniformly distributed makes little sense to condense by taking a logarithm.

```
data <- data %>% mutate(month = as.factor(month(index)))

lm_seasonal_model <- lm(value ~ month_since_start + month, data=data)

index <- seq(from = ymd("1998-01-01"), to = ymd("2020-12-01"), by = "1 month")

ext_data <- data.frame(index = index) %>% as_tsibble(index=index) %>% mutate(month = as.factor(month(index)))
```

```r
ext_data$month_since_start <- seq(max(data$month_since_start)+1,max(data$month_since_start)+(23*12))

forecast <- predict(lm_seasonal_model, newdata = ext_data)
```

Part 3a