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)
library(tseries)
```

```
data <- datasets::co2

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

EDA

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

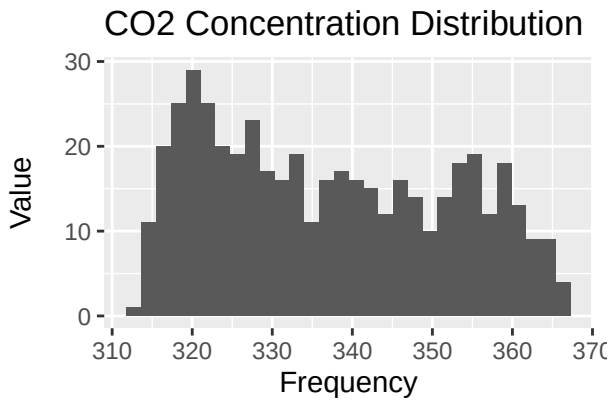
The CO2 data set consists of 468 observations. Each observation represents the monthly total atmospheric concentration of CO2, measured in parts per

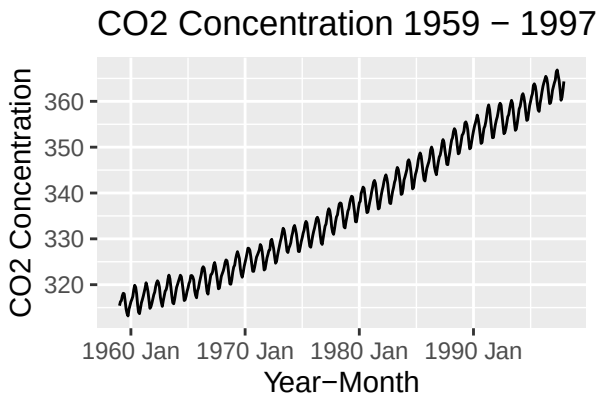
million (ppm) and collected at the Mauna Loa Observatory in Hawaii. The data ranges from January 1959 to December 1997. The data is originally sourced from the Scripps institute and was collected as part of the Scripps CO2 Program. Observations for February, March, and April 1964 were unavailable so the values in the data et were generated via linear interpolation between the observations for January and May 1964.

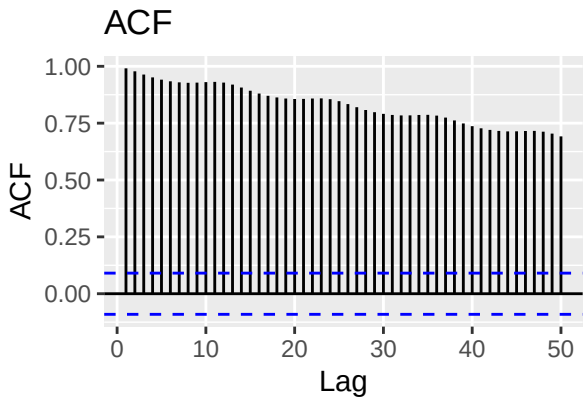
2. Investigation of trend, seasonal and irregular elements

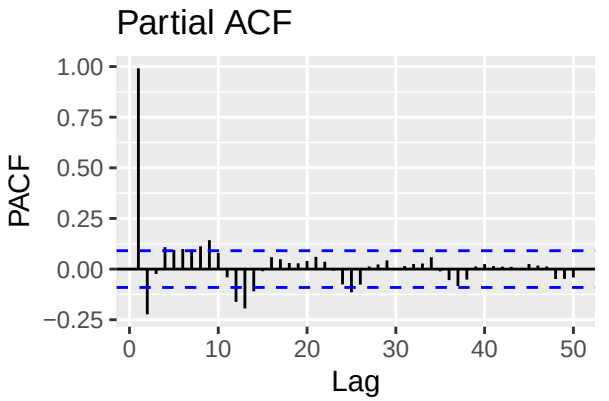
```
# Distribution
dist <- ggplot(data, aes(x = value)) +
  geom_histogram() +
  ylab("Value") +
  xlab("Frequency") +
  ggtitle("CO2 Concentration Distribution")

# 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

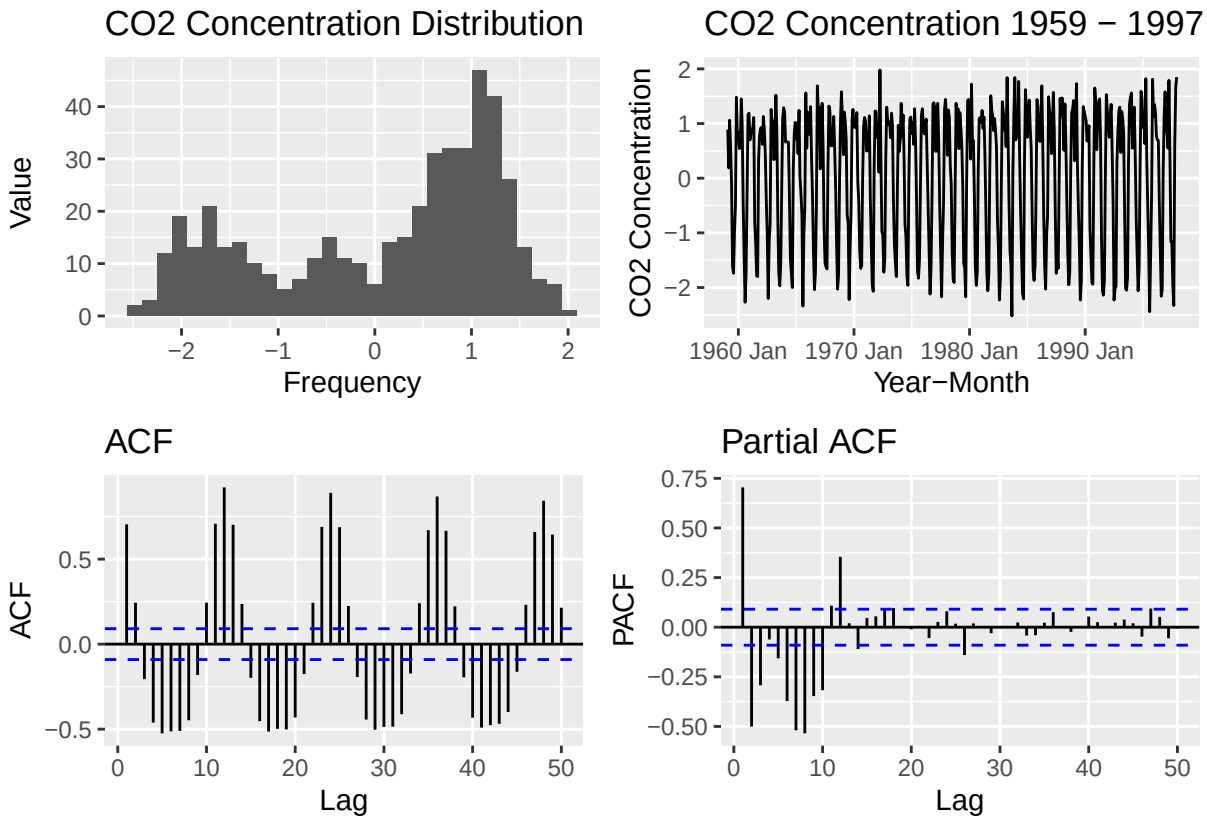
```
data_diff <- data
data_diff$value_diff <- difference(data_diff$value)
# Distribution
dist <- ggplot(data_diff, aes(x = value_diff)) +
  geom_histogram() +
  ylab("Value") +
  xlab("Frequency") +
  ggtitle("CO2 Concentration Distribution")
```

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

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

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

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

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

```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```

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

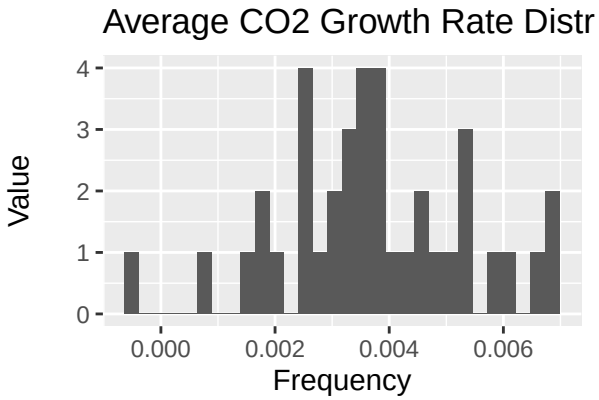


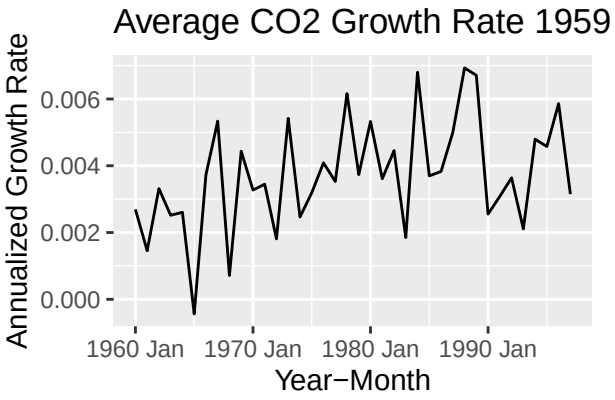
3. Trends in levels and growth rates should be discussed (long-run growth rate as annualized averages)

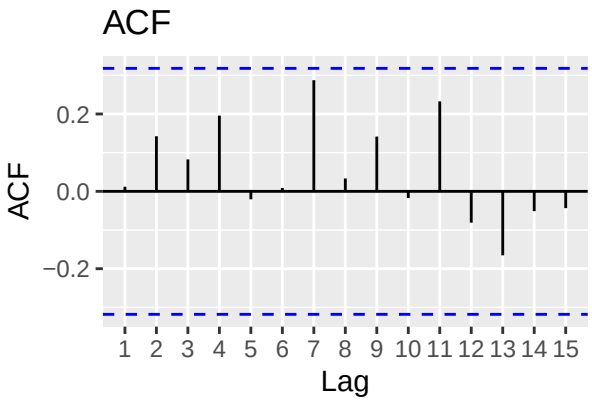
```
data <- data %>% mutate(month = as.factor(month(index)))

annual_growth <- data %>% filter(month == 1) %>%
  mutate(co2_diff = value - lag(value),
         growth_rate = co2_diff / lag(value) ) %>%
  na.omit()

# Distribution
dist <- ggplot(annual_growth, aes(x = growth_rate)) +
  geom_histogram() +
  ylab("Value") +
```

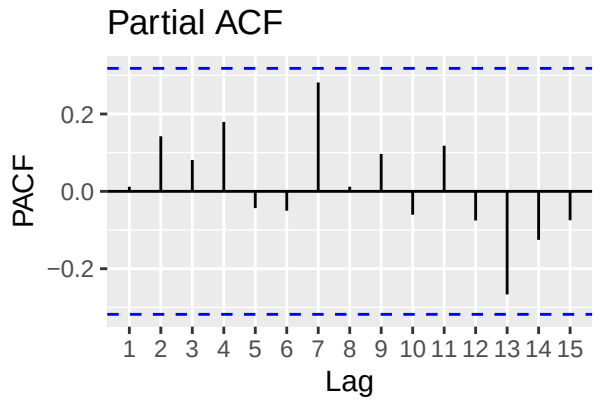
```
  xlab("Frequency") +
  ggtitle("Average CO2 Growth Rate Distribution")

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

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

# PACF
pacf <- ggPacf(annual_growth$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
```

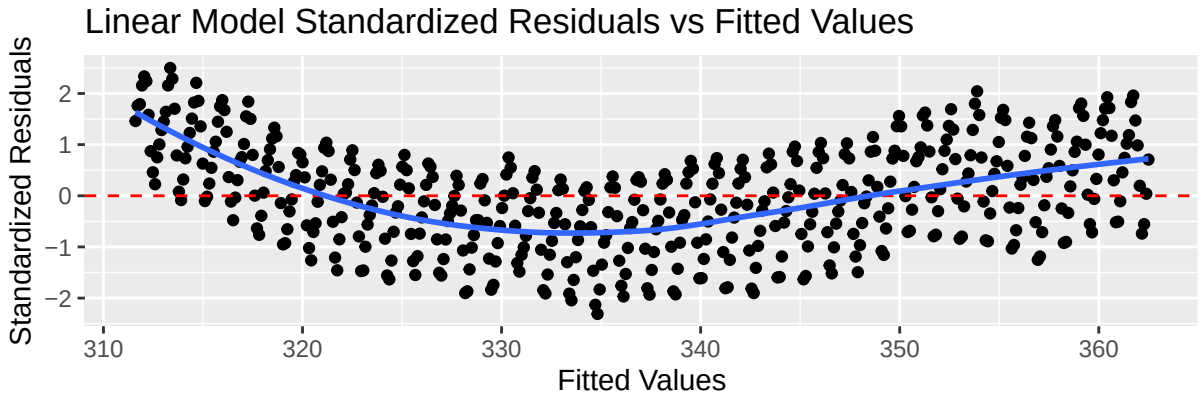
```
quad_model <- lm(value ~ I(month_since_start^2) , data=data)
quad_model
```

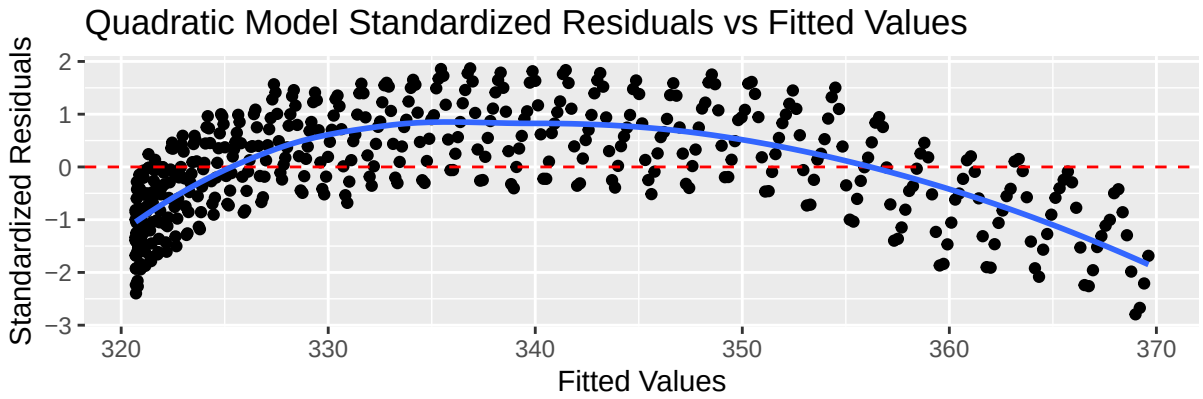
```
##
## Call:
## lm(formula = value ~ I(month_since_start^2), data = data)
##
## Coefficients:
##            (Intercept)  I(month_since_start^2)
##              3.207e+02               2.234e-04
```

```
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) +
```

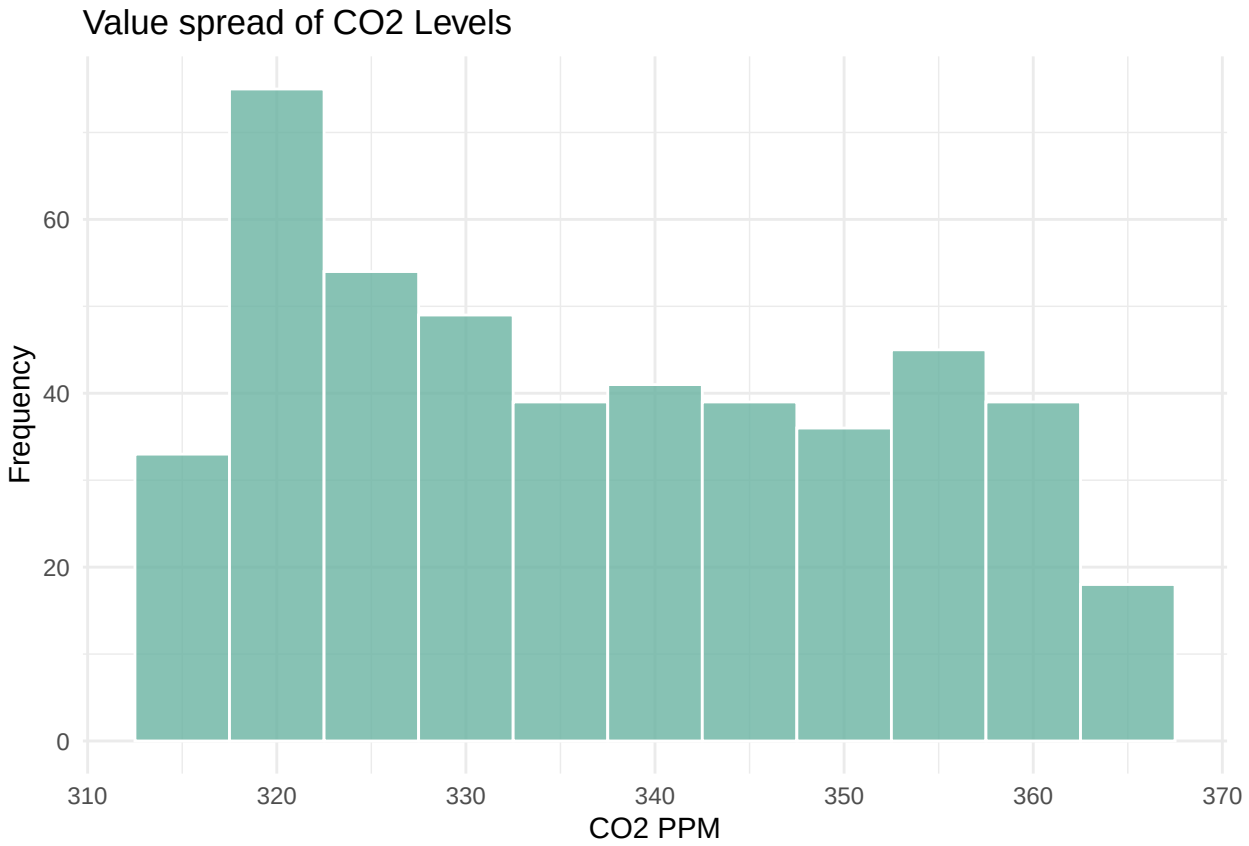
```
  labs(title = "Linear Model Standardized Residuals vs Fitted Values",
       x = "Fitted Values", y = "Standardized Residuals")

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

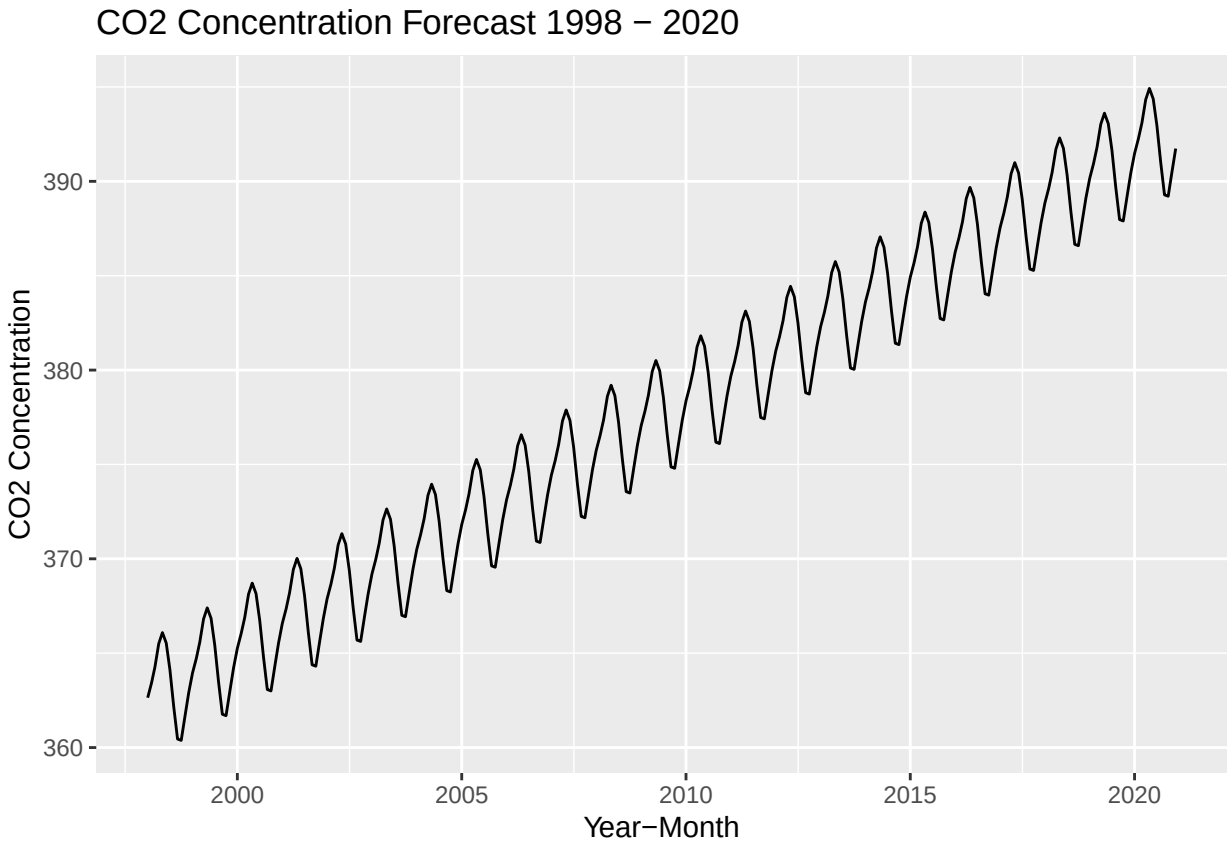
```
value_hist
```



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 that the variance of the residuals is constant over time.

```
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)))

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

```r
forecast <- data.frame(predict = predict(lm_seasonal_model, newdata = ext_data))

forecast$index <- ext_data$index

ggplot(forecast, aes(x=index, y=predict)) +
  geom_line() +
  ylab("CO2 Concentration") +
  xlab("Year-Month") +
  ggtitle("CO2 Concentration Forecast 1998 - 2020")
```



Part 3a

```
adf_result <- adf.test(data$value)

kpss_result <- kpss.test(data$value)

paste("ADF Test p-value",adf_result$p.value)
```

```
## [1] "ADF Test p-value 0.22692616692272"
```

```
paste("KPSS Test p-value",kpss_result$p.value)
```

```
## [1] "KPSS Test p-value 0.01"
```

Both the ADF & KPSS tests determined that the series was non-stationary and needed to be differenced.

```
data_diff <- data %>% mutate(value_diff = difference(value)) %>% na.omit()

diff_adf_result <- adf.test(data_diff$value_diff)

diff_kpss_result <- kpss.test(data_diff$value_diff)

paste("ADF Test p-value",diff_adf_result$p.value)
```

```
## [1] "ADF Test p-value 0.01"
```

```
paste("KPSS Test p-value",diff_kpss_result$p.value)
```

```
## [1] "KPSS Test p-value 0.1"
```

Now both tests conclude that the differenced series is stationary and modeling can continue on the differenced series.

```
model_ma.bic<-data %>%
model(ARIMA(value ~ 1 + pdq(0,0:1,0:3) + PDQ(0,0:1,0:3), ic="bic", stepwise=F, greedy=F,
            order_constraint = p + q + P + Q <= 10))
```

```
## Warning: Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
```

```
model_ma.bic %>%
report()
```

```
## Series: value
## Model: ARIMA(0,1,1)(0,1,1)[12] w/ poly
```

```
##
## Coefficients:
##           ma1     sma1  constant
##       -0.3539  -0.8563    0.0021
## s.e.   0.0498   0.0254    0.0015
##
## sigma^2 estimated as 0.08558:  log likelihood=-85.12
## AIC=178.24   AICc=178.33   BIC=194.72
```

```
model_ar.bic<-data %>%
model(ARIMA(value ~ 1 + pdq(0:3,0:1,0) + PDQ(0:3,0:1,0), ic="bic", stepwise=F, greedy=F,
            order_constraint = p + q + P + Q <= 10))
```

```
## Warning: Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
```

```
model_ar.bic %>%
report()
```

```
## Series: value
## Model: ARIMA(3,1,0)(3,1,0)[12] w/ poly
##
## Coefficients:
##           ar1      ar2      ar3     sar1     sar2     sar3  constant
##       -0.3678  -0.1561  -0.1112  -0.6756  -0.4821  -0.2333    0.0089
## s.e.   0.0469   0.0500   0.0474   0.0477   0.0529   0.0480    0.0146
##
## sigma^2 estimated as 0.09642:  log likelihood=-106.97
## AIC=229.94   AICc=230.27   BIC=262.91
```

```
model_full.bic<-data %>%
model(ARIMA(value ~ 1 + pdq(0:3,0:1,0:3) + PDQ(0:3,0:1,0:3), ic="bic", stepwise=F, greedy=F,
            order_constraint = p + q + P + Q <= 10))
```

```
## Warning: Model specification induces a quadratic or higher order polynomial trend.
## This is generally discouraged, consider removing the constant or reducing the number of differences.
```

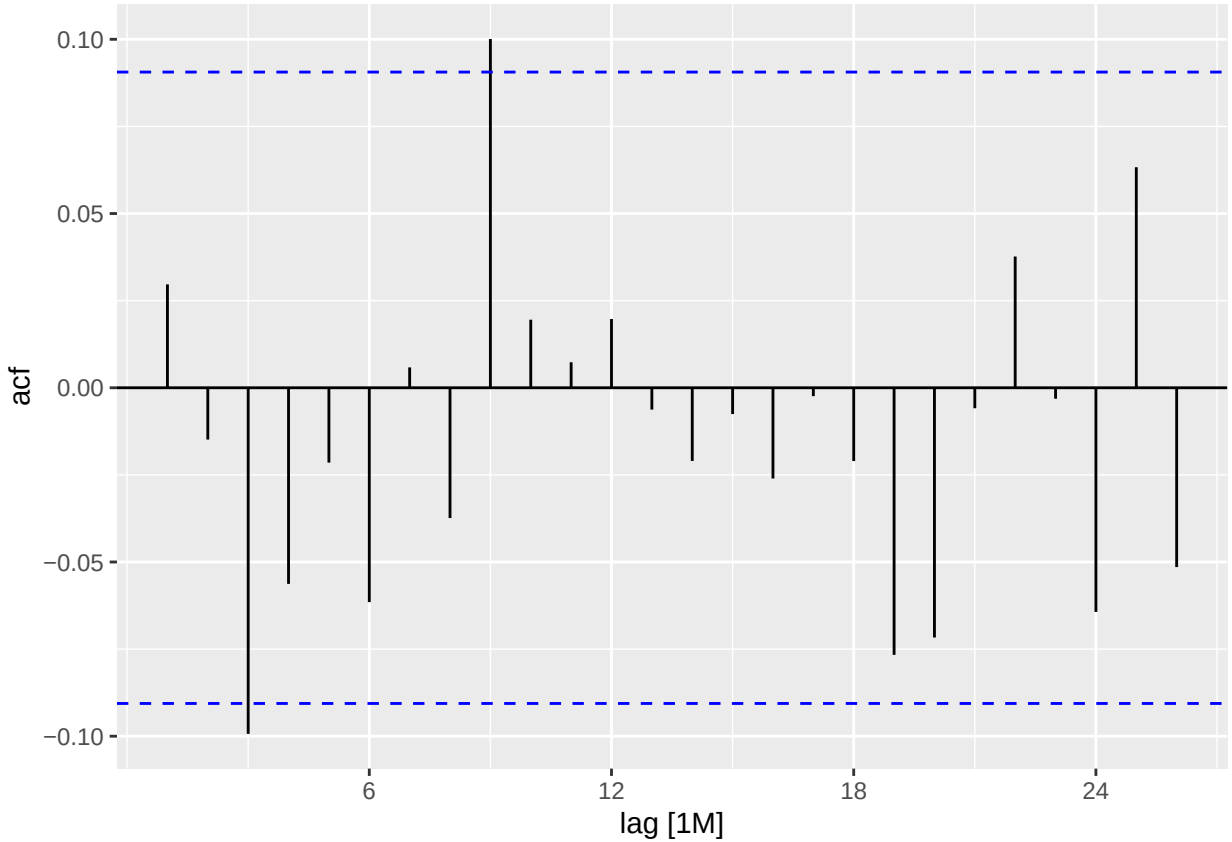
```
model_full.bic %>%
report()
```

```
## Series: value
## Model: ARIMA(0,1,1)(1,1,2)[12] w/ poly
```

```
##
## Coefficients:
##           ma1     sar1     sma1     sma2  constant
##       -0.3521  -0.5363  -0.2842  -0.4984    0.0033
## s.e.   0.0501   0.5606   0.5440   0.4621    0.0023
##
## sigma^2 estimated as 0.08579:  log likelihood=-84.63
## AIC=181.26   AICc=181.45   BIC=205.98
```

The final model was selected using the BIC scores with the ultimate winner being the SARIMA (0,1,1) (0,1,1) [12] model. BIC score was our preferred way of selecting a model due to its inherent penalty on adding additional terms leading to a well fitted and simpler model. Finding the ideal model is done through grid search of various parameters as we employed three simultaneous searches to see the most variety of different model specialties. All searches included non-zero difference terms based on the our EDA while one model focused on AR terms, another MA terms and one finally on both at the same time.

```
model_ma.bic %>%
augment() %>%
ACF(.resid) %>%
autoplot()
```



From simple inspection of the residual ACF plot is hard to conclude if the full covariance structure has been removed from the data. However, it seems likely as lags rarely fall outside the 95% confidence interval around 0 correlation.

```
resid.ts<-model_ma.bic %>%
augment() %>%
pull(.resid) %>%
as.ts()

Box.test(resid.ts, lag = 10, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
```

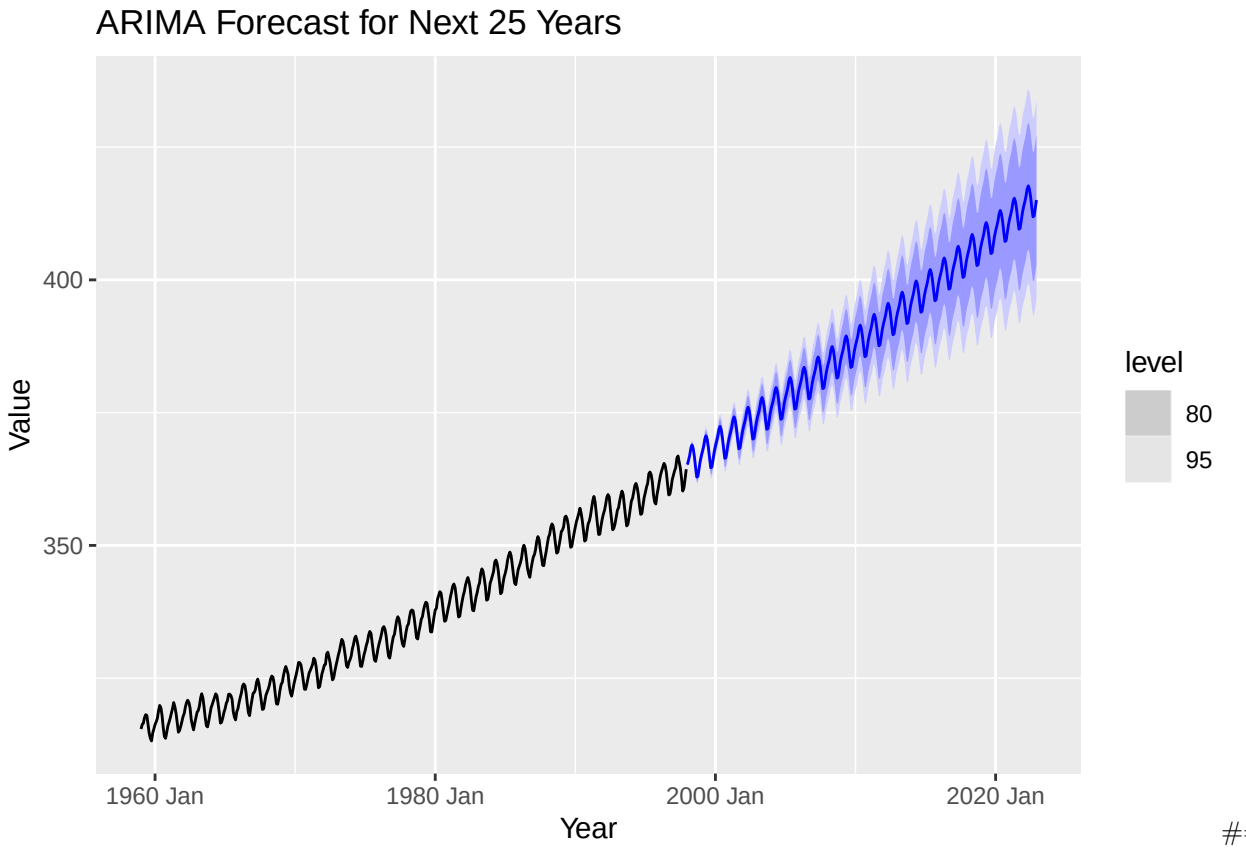
```
## data:  resid.ts
## X-squared = 14.376, df = 10, p-value = 0.1565
```

Having a p-value greater than .05 from the Box-Ljung test implies we fail to reject the null hypothesis that residuals are randomly distributed. Thus the residuals do not exhibit significant autocorrelation giving greater confidence that the model found is appropriate for this data.

```
model_ma.forecasts<-forecast(model_ma.bic, h=25*12)

autoplot(model_ma.forecasts) +
  autolayer(data, series = "Historical Data", colour = "black") +
  xlab("Year") +
  ylab("Value") +
  ggtitle("ARIMA Forecast for Next 25 Years")
```

```
## Plot variable not specified, automatically selected `.vars = value`
```

```
## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: `series`
```



Part 4a

```
model_ma.forecast_2100 <- forecast(model_ma.bic, h=103*12)

forecast_values <- model_ma.forecast_2100$value

i <- 1

f_level_420 <- list()
f_level_420_n <- list()
f_level_420_f <- list()

f_level_500 <- list()
f_level_500_n <- list()
```

```r
f_level_500_f <- list()

l_level_420 <- list()
l_level_420_n <- list()
l_level_420_f <- list()

l_level_500 <- list()
l_level_500_n <- list()
l_level_500_f <- list()

for (f in forecast_values) {

  mu <- as.numeric(f[1])
  sigma <- as.numeric(f[2])

  if (mu >= 420) { # first at 420
     f_level_420 <- c(f_level_420, i)
  }

  if (mu + (2*sigma) >= 420) { # first at 420 (near term)
     f_level_420_n <- c(f_level_420_n, i)
  }

  if (mu - (2*sigma) >= 420) { # first at 420 (far term)
     f_level_420_f  <- c(f_level_420_f, i)
  }

  if (mu < 420) { # last at 420
     l_level_420 <- c(l_level_420, i)
  }

  if (mu + (2*sigma) < 420) { # last at 420 (near term)
     l_level_420_n <- c(l_level_420_n, i)
  }

  if (mu - (2*sigma) < 420) { # last at 420 (far term)
     l_level_420_f <- c(l_level_420_f, i)
  }
```

```
  if (mu >= 500) { # first at 500
     f_level_500 <- c(f_level_500, i)
  }

  if (mu + (2*sigma) >= 500) { # first at 500 (near term)
     f_level_500_n <- c(f_level_500_n, i)
  }

  if (mu - (2*sigma) >= 500) { # first at 500 (far term)
     f_level_500_f <- c(f_level_500_f, i)
  }


  if (mu < 500) { # last at 500
     l_level_500 <- c(l_level_500, i)
  }

  if (mu + (2*sigma) < 500) { # last at 500 (near term)
     l_level_500_n <- c(l_level_500_n, i)
  }

  if (mu - (2*sigma) < 500) { # last at 500 (far term)
     l_level_500_f <- c(l_level_500_f, i)
  }


  i <- i + 1

}

f_420_n <- model_ma.forecast_2100$index[f_level_420_n[[1]]]
f_420 <- model_ma.forecast_2100$index[f_level_420[[1]]]
f_420_f <- model_ma.forecast_2100$index[f_level_420_f[[1]]]

paste('Forecated mean first time at 420 ppm :',f_420)
```

```
## [1] "Forecated mean first time at 420 ppm : 2023 May"
```

```r
paste('95% Range from:',f_420_n, 'to',f_420_f)
```

```
## [1] "95% Range from: 2017 May to 2036 May"
```

```r
l_420_n <- model_ma.forecast_2100$index[l_level_420_n[[length(l_level_420_n)]]+1]
l_420 <- model_ma.forecast_2100$index[l_level_420[[length(l_level_420)]]+1]
l_420_f <- model_ma.forecast_2100$index[l_level_420_f[[length(l_level_420_f)]]+1]

paste('Forecated mean last time at 420 ppm :',l_420)
```

```
## [1] "Forecated mean last time at 420 ppm : 2025 Nov"
```

```r
paste('95% Range from:',l_420_n, 'to',l_420_f)
```

```
## [1] "95% Range from: 2018 Dec to 2039 Nov"
```

```r
f_500_n <- model_ma.forecast_2100$index[f_level_500_n[[1]]]
f_500 <- model_ma.forecast_2100$index[f_level_500[[1]]]
f_500_f <- model_ma.forecast_2100$index[f_level_500_f[[1]]]

paste('Forecated mean first time at 500 ppm :',f_500)
```

```
## [1] "Forecated mean first time at 500 ppm : 2053 Mar"
```

```r
paste('95% Range from:',f_500_n, 'to',f_500_f)
```

```
## [1] "95% Range from: 2040 Apr to 2078 May"
```

```r
l_500_n <- model_ma.forecast_2100$index[l_level_500_n[[length(l_level_500_n)]]+1]
l_500 <- model_ma.forecast_2100$index[l_level_500[[length(l_level_500)]]+1]
l_500_f <- model_ma.forecast_2100$index[l_level_500_f[[length(l_level_500_f)]]+1]

paste('Forecated mean last time at 500 ppm :',l_500)
```

```
## [1] "Forecated mean last time at 500 ppm : 2053 Dec"
```

```r
paste('95% Range from:',l_500_n, 'to',l_500_f)
```

```
## [1] "95% Range from: 2041 Oct to 2080 Nov"
```

```r
last_index <- length(model_ma.forecast_2100$index)
last_year <- last_index-11
```

```r
prediction_2100 <- (mean(model_ma.forecast_2100$value[last_index]) + mean(model_ma.forecast_2100$value[last_year]))/2

paste('2100 C02 ppm prediction:',round(prediction_2100,2))
```

```
## [1] "2100 C02 ppm prediction: 674.66"
```

Confidence is low in all of these predictions due to the widening variance of the predictions over time. Despite the strong cyclical patterns forecasting this far out in time is an incredibly hard task to do accurately.