Assignment 6: Gas Furnace

Joshua Goldberg

May, 23 2019

Data

Input gas rate - this is the independent variable

Output gas CO2 % - this is the dependent variable that needs to be forecast

```
gas_data <- read_csv("Gas Furnace Dataset.csv") %>%
  janitor::clean_names()

gas_data %>% head()
```

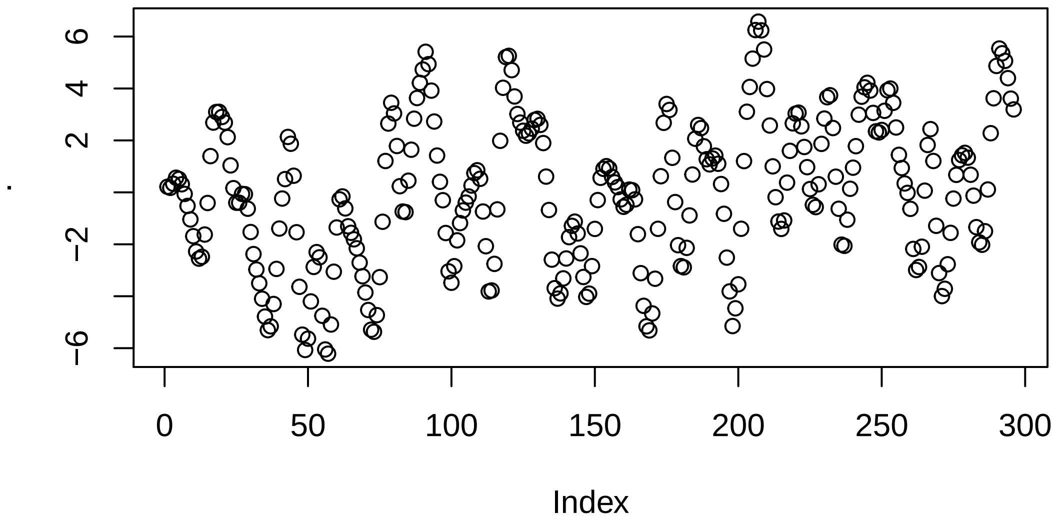
```
## # A tibble: 6 x 2
##   input_gas_rate outlet_gas_co2_percentage
##            <dbl>                     <dbl>
## 1         -0.109                      53.8
## 2          0                          53.6
## 3          0.178                      53.5
## 4          0.339                      53.5
## 5          0.373                      53.4
## 6          0.441                      53.1
```

Tasks

1 Linear regression

Use linear regression model - plot the ACF - what can you conclude?

```
model_formula <- formula(outlet_gas_co2_percentage ~ input_gas_rate)
lm_mod <- lm(model_formula, gas_data)
residuals(lm_mod) %>% plot()
```

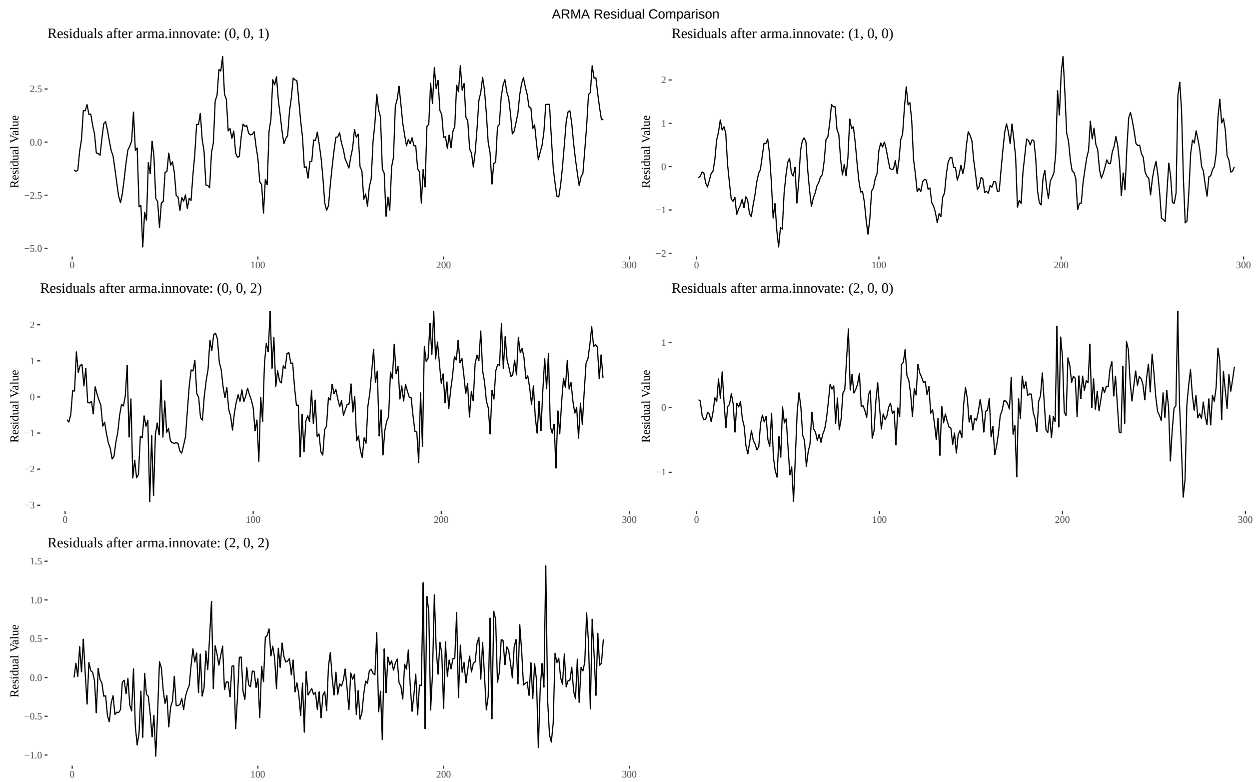


2-6 ARIMA

Use ARIMA models for the residuals. Adjust the Input gas rate and Output CO2 % with the AR/MA coefficient. Combine with the linear regression model. Plot the residuals.

```
arma_innovate <- function(.data, .formula, .params) {
  lm_model <- lm(.formula, .data)
  resids <- lm_model %>% residuals()
  arima_model <- Arima(resids, .params)
  data_vecs <- map(gas_data, ~ .x)
  new_data <- map_dfc(data_vecs, ~ arma.innovation(.x, arima_model))
  lm_mod_adj <- lm(.formula, new_data)
  list(model = lm_mod_adj, residuals = lm_mod_adj %>% residuals())
}

params <-
  list(
    "(0, 0, 1)" = c(0, 0, 1),
    "(1, 0, 0)" = c(1, 0, 0),
    "(0, 0, 2)" = c(0, 0, 2),
    "(2, 0, 0)" = c(2, 0, 0),
    "(2, 0, 2)" = c(2, 0, 2)
  )

innovate_results <- map(params, ~ arma_innovate(gas_data, model_formula, .x))
models <- map(innovate_results, "model")
resids <- map(innovate_results, "residuals") %>%
  map( ~ tibble::enframe(.x, name = NULL))

resid_plots <- map2(resids, (resids %>% names), function(resid, name) {
  plot <- resid %>%
    mutate(row = row_number()) %>%
    ggplot(aes(row, value)) +
    geom_line() +
    labs(title = glue::glue("Residuals after arma.innovate: {name}"),
         x =  NULL,
```

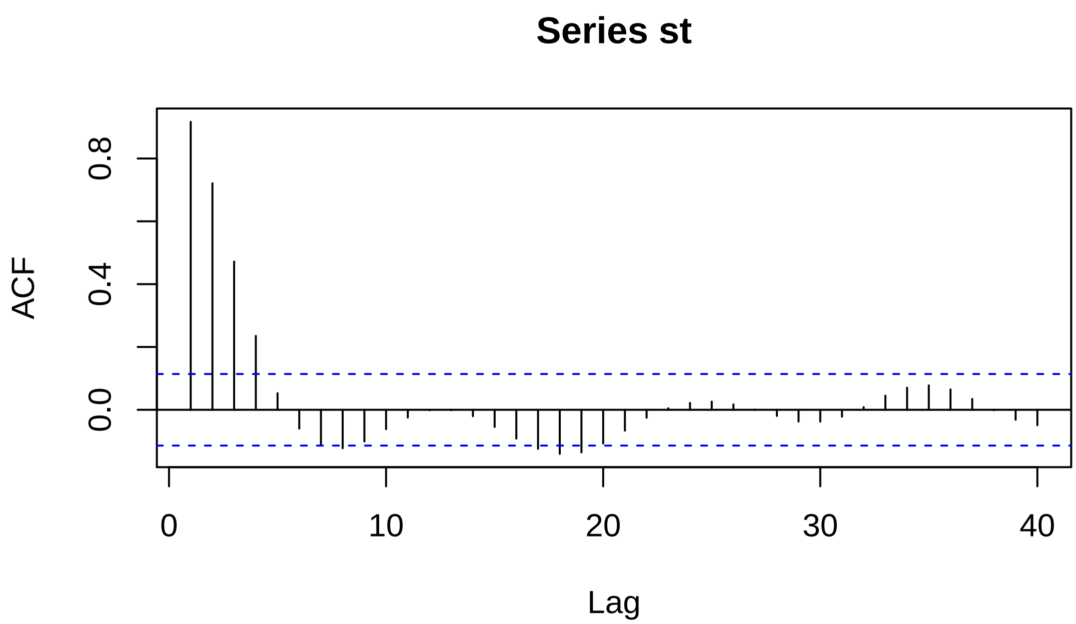
```r
          y = "Residual Value")
})

gridExtra::grid.arrange(resid_plots$`(0, 0, 1)`,
                        resid_plots$`(1, 0, 0)`,
                        resid_plots$`(0, 0, 2)`,
                        resid_plots$`(2, 0, 0)`,
                        resid_plots$`(2, 0, 2)`, ncol = 2,
                        top = "ARMA Residual Comparison")
```



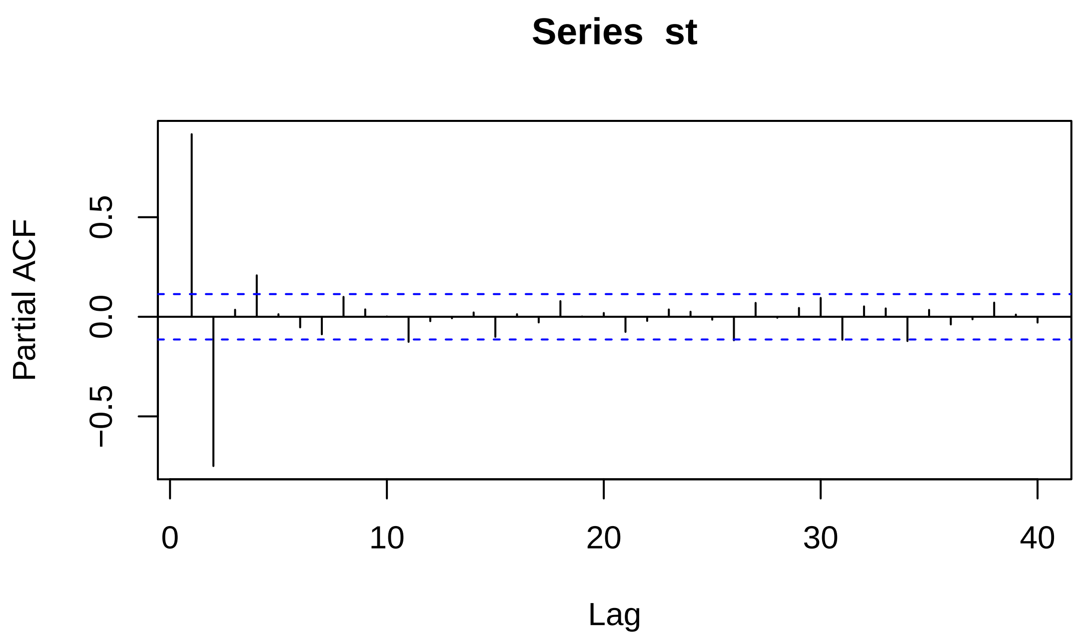
7 ARFIMA

Use fractional ARIMA model (aka ARFIMA) for the output gas CO2% - plot the residuals, acf and pacf plots of the model. You can use an R package like fracdiff – be careful to determine which lag to choose when executing this test.

```r
y <- gas_data$outlet_gas_co2_percentage
# get the fractional d
d <- fracdiff::fracdiff(y)

# do the fractional difference
st <- fracdiff::diffseries(y, d$d)
acf(st, lag = 40)
```



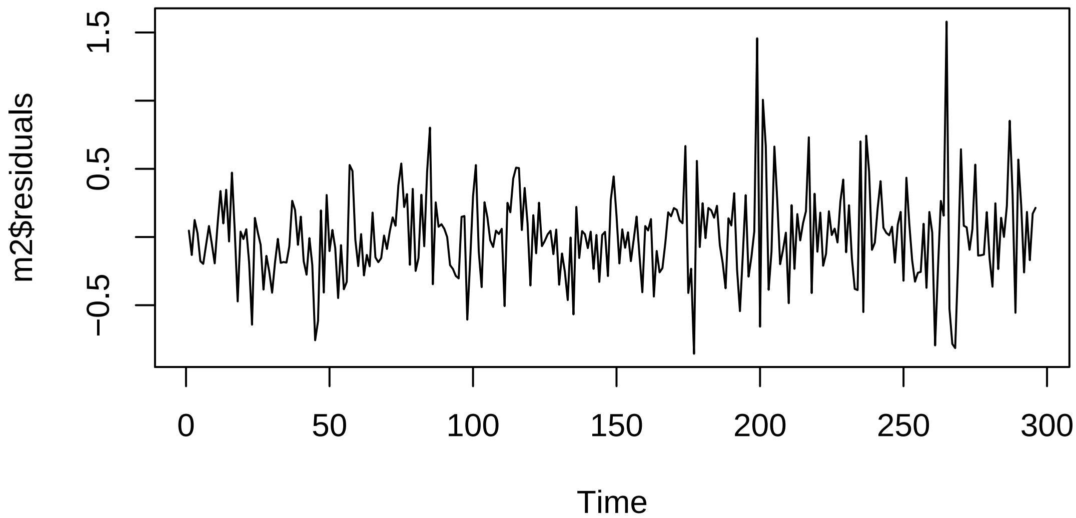

```
pacf(st, lag = 40)
```




```
# now the TS is stationary, run ARIMA
m1 <- auto.arima(st)
AIC(m1)
```

```
## [1] 198.6697
```

```
# does the above (fractional difference + ARIMA) in 1 step
m2 <- forecast::arfima(y)

plot(m2$residuals)
```




```
# you can compare the models and choose the one with lower AIC
AIC(m2)
```

```
## [1] 197.8829
```

```
arfima_models <- list(m1, m2)

best_arfima <- arfima_models[which(c(AIC(m1), AIC(m2)) == max(AIC(m1), AIC(m2)))][[1]]
```

8 Durbin-Watson and Box-Ljung tests

Perform summaries, Durbin-Watson and Box-Ljung tests for each model and build table to compare AICs, BICs and p-vaules for each test across the ARIMA and ARFIMA models.

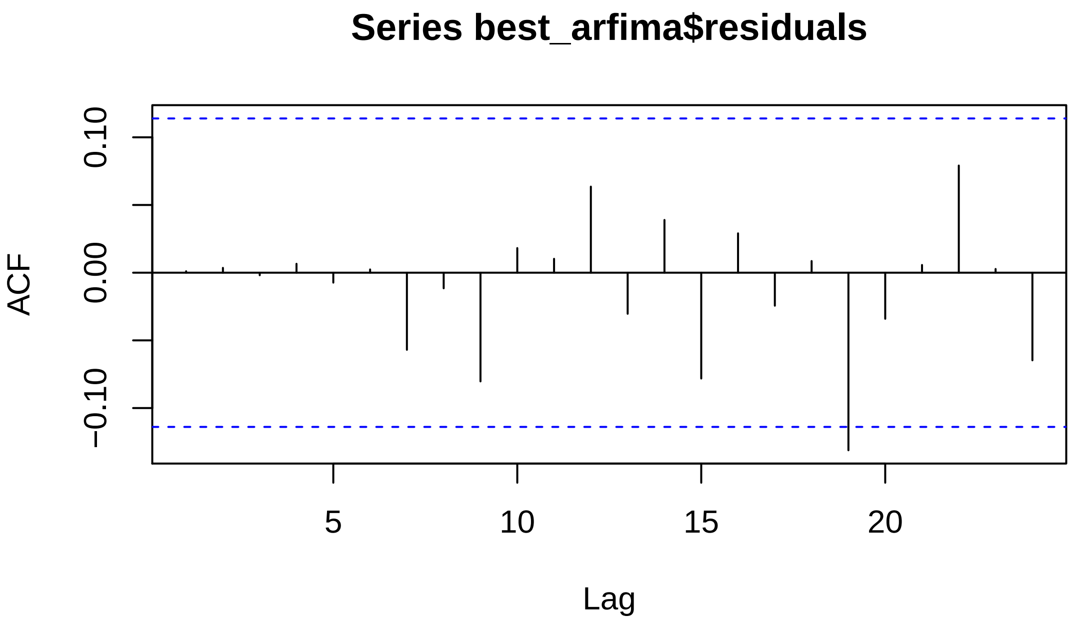
```
all_models <- c(models, arfima = list(best_arfima))
dw_p_values <- map_dbl(c(models, list(lm(best_arfima$residuals ~ 1))), ~ dwtest(.x)$p.value)
box_p_values <-
  map_dbl(all_models,
          ~ Box.test(.x %>% residuals(), lag = 10, type = "L")$p.value)
aics <- map_dbl(all_models, ~ AIC(.x))
bics <- map_dbl(all_models, ~ BIC(.x))

models %>%
  enframe("parameter", "model") %>%
  bind_rows(
    tibble(
      parameter = arimaorder(best_arfima) %>% str_c(collapse = ", "),
      model = list(best_arfima))) %>%
  mutate(dw_p_value = dw_p_values,
         box_p_value = box_p_values,
         aic = aics,
         bic = bics) %>%
  select(-model) %>%
  knitr::kable(format = "latex")
```

parameter	dw_p_value	box_p_value	aic	bic
(0, 0, 1)	0.0000000	0.0000000	1141.4688	1152.4367
(1, 0, 0)	0.0000000	0.0000000	652.9356	663.9965
(0, 0, 2)	0.0000000	0.0000000	796.6360	807.6040
(2, 0, 0)	0.0000000	0.0000000	359.8765	370.9273
(2, 0, 2)	0.0000130	0.0000000	256.5856	267.5536
2, 0, 4	0.4887582	0.9776214	198.6697	224.5022

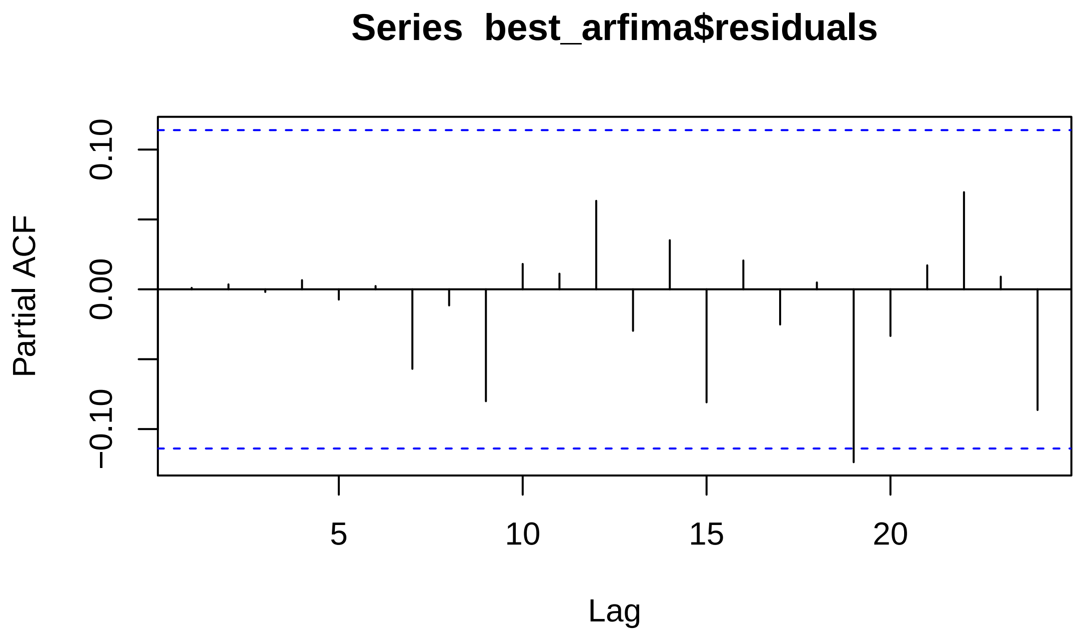
9 Best model

Based on ACF plots and test results, which ARIMA model gives the best result in terms of residuals being close to white noise? ARFIMA performs the best from an AIC and BIC perspective, most likely driven by the differencing. However, if we do `auto.arima()`, we may find better parameters to feed into `arma.innovation()`.

```
acf(best_arfima$residuals)
```




```
pacf(best_arfima$residuals)
```


Series best_arfima$residuals
Partial ACF
0.10
0.00
−0.10
5
10
15
20
Lag