Course 2 Section 3.4 - INFERENCE

Jiaying Wu

15/10/2020

```
# Load tidyverse
library(tidyverse)

# Load lubridate
library(lubridate)

# load broom
library(broom)

# Read CO2 data and apply some pre-processing
CO2.spo <- read_csv(
  "https://raw.githubusercontent.com/datascienceprogram/ids_course_data/master/daily_merge_co2_spo.csv"
  col_names = c("date", "time", "day", "decdate", "n", "flg", "co2"), skip = 69) %>%
  filter(flg == 0) %>%
  mutate(date = ymd(date))
```

```
## Warning: 2 parsing failures.
## row col  expected    actual
##   1  -- 7 columns 1 columns 'https://raw.githubusercontent.com/datascienceprogram/ids_course_data/mas
##   2  -- 7 columns 1 columns 'https://raw.githubusercontent.com/datascienceprogram/ids_course_data/mas
```

```
# Create variable day0 (a rescaling of day)
CO2.spo <- CO2.spo %>%
  mutate(day0 = day - min(day))

# Add predictions, residuals, etc. to the training data
co2_fit <- lm(co2~day0, data=CO2.spo)
```

Give it a go!

Q1.Try to add a quadratic term (by squaring day0), or more, to the model to improve the fit. While you're doing this, you may want to centre the day0 values, or even standardise them, to get a nice quadratic form.

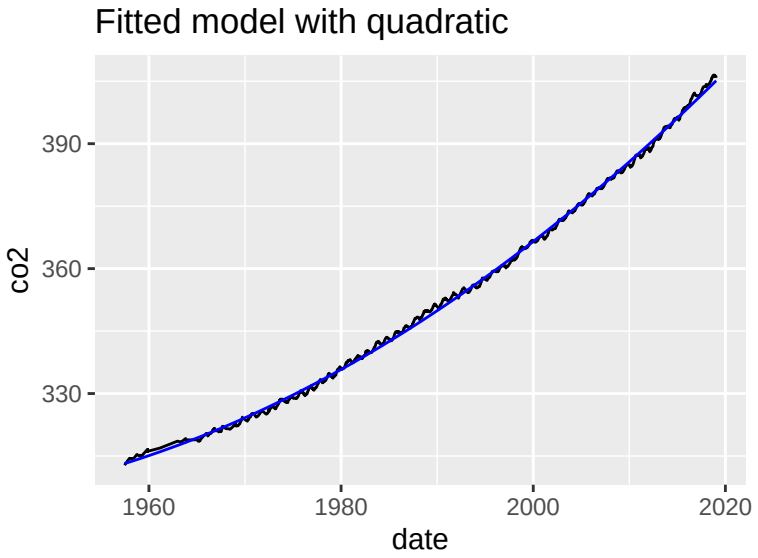
```
# Fit second model that includes day squared as an explanatory variable
co2_fit2 <- lm(co2~day0+I(day0^2), data=CO2.spo)

# Tidy output of fitted model
tidy(co2_fit2)
```

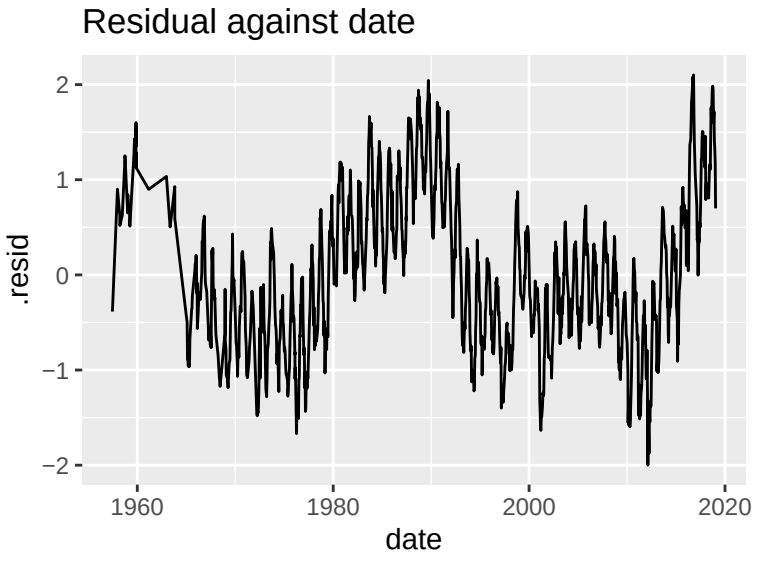
```
## # A tibble: 3 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  3.13e+2  1.01e- 1     3096.        0
## 2 day0         1.97e-3  1.80e- 5      109.        0
## 3 I(day0^2)    9.42e-8  7.13e-10      132.        0
```

```
# Append fitted and residual value into CO2.spo (training data)
co2_model2 <- augment(co2_fit2, CO2.spo)

# Plot fitted model over the data
ggplot(co2_model2, aes(x=date, y=co2)) +
  geom_line() +
  geom_line(aes(y=.fitted), colour="blue") +
  labs(title = "Fitted model with quadratic")
```




```
# Plot residual against date
ggplot(co2_model2, aes(x=date, y=.resid)) +
  geom_line() +
  labs(title = "Residual against date")
```



Q2.Predict CO2 at another location

Step 1: Download the data

```
# Read CO2 from Point Barrow data and apply some data pre-processsing
CO2.ptb <- read_csv(
  "https://raw.githubusercontent.com/datascienceprogram/ids_course_data/master/daily_merge_co2_ptb.csv"
  col_names = c("date", "time", "day", "decdate", "n", "flg", "co2"), skip = 69) %>%
  mutate(lat = -90.0, lon = 0, stn = "ptb") %>%
  filter(flg == 0) %>%
  mutate(date = ymd(date))
```

Step 2: Create a variable, and more

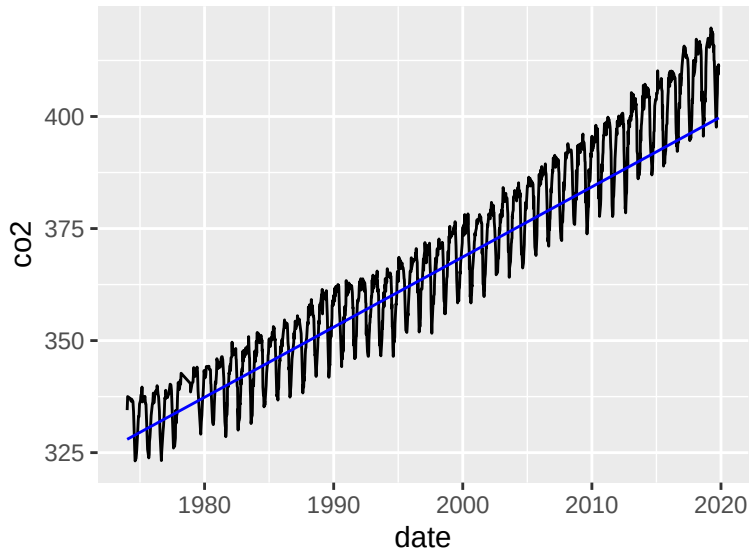
```
CO2.ptb <- CO2.ptb %>% mutate(day0 = day - min(CO2.spo$day))
```

Step 3: Fit new data

```
co2_model_ptb <- augment(co2_fit, newdata=CO2.ptb)
```

Step 4: Plot and overlay

```
ggplot(co2_model_ptb, aes(x=date, y=co2)) +
  geom_line() +
  geom_line(aes(y=.fitted), colour="blue")
```


400
375
350
325
co2
1980
1990
2000
2010
2020
date