Day 1 - R and regression review

Introduction

> ChatGPT: What is time series analysis?
>
> The study of time series is the study of statistical techniques used to analyzed time-ordered data points. The goal is to identify patterns, trends, and relationships within the data over time. This type of analysis is crucial in various fields, including finance, economics, environmental science, and engineering. Key components and objectives of time series analysis include: trend analysis, understanding seasonality, forecasting, and decomposition.

The study of serially correlated data builds upon the basic framework of linear models. The purpose of today's lab/lecture is to ensure everyone is both proficient with the software required for this course and with linear models and regression.

A quick note on separating your response from the rest of the text in assignments: when possible, please use callout boxes to separate your responses from the rest of the text, such as:

> Here is the answer!

R, RStudio, and TinyTex

1. Install R and RStudio. R and RStudio may be installed for free at https://cran.r-project.org/ and https://posit.co/downloads/, respectively.

   > Note
   >
   > RStudio is an integrated development environment (IDE) that we will use to interact with the statistical software R. Seldom will we interact with the program R directly.

2. RStudio (but really, Pandoc) allows us to render a combination of plain text, code, and output into a variety of file types using a single reproducible document known as a Quarto document (.qmd file). While RStudio is capable of rendering documents into numerous file types, we will generally focus on HTML, Word, and PDF in this class.

   You will be expected to turn in a .qmd file and .pdf file for all assignments in this course - any other file type will not be graded. In order to render to a .pdf file, RStudio relies upon a LaTeX distribution. There are a variety of LaTeX distributions available, but I

recommend using `tinytex`, as it is made specifically for use with R. To install `tinytex`, run the following code chunk:

```
install.packages("tinytex")
tinytex::install_tinytex()
```

> **!** STOP
>
> Before continuing, click the render button at the top of the *editor* panel and ensure that you are able to render this document to a .pdf file.

Combining languages

3. When creating .qmd files, you can use a variety of different languages to produce desired effects. For example, you may use Markdown to typeset the .qmd file (in fact, the `[]()` shortcut used to hyperlink a website *is* Markdown). You are also able to use other languages to typeset the document, which depend on what type of document you are rendering. For example, when rendering to .pdf, you may use LaTeX commands; when rendering to .html, you may use HTML code.

> **!** STOP
>
> Copy and paste the entire YAML header (which begins with `---` as the first line of this document, and ends with `---` on line 23 of this document) into a separate R script (under the file tab) so that you have it for later.

Render this document first as a .pdf file by clicking the render button. Which languages rendered as boldface?

- Markdown: **The quick brown fox jumps over the lazy dog.**
- LaTeX: **The quick brown fox jumps over the lazy dog.**
- HTML: The quick brown fox jumps over the lazy dog.

Next, render the document as a .html file by replacing

```
format:
 pdf:
 fontsize: 12pt
 geometry:
   - inner=1.5cm
   - outer=1.5cm
   - top=2.5cm
   - bottom=2.5cm
 include-in-header:
```

```
  - text: |
      \addtokomafont{disposition}{\rmfamily}
      \usepackage{fancyhdr, lastpage, framed, caption}
      \captionsetup[figure]{labelformat=empty}
      \pagestyle{fancyplain}
      \fancyhf{}
      \lhead{\fancyplain{}{STAT 0116: Introduction to Statistical Science}}
      \rhead{\fancyplain{}{Stratton - Day 2}}
      \fancyfoot[RO, LE] {page \thepage\ of \pageref{LastPage}}
      \thispagestyle{plain}
```

with

```
format: html
```

in the YAML header and clicking render. Which languages rendered as boldface?

> Only the Markdown renders in both! In this class, we will generally prefer .pdf documents, so LaTeX or Markdown are permitted.

Learning LaTeX, Markdown, or HTML is not the focus of this class. However, you will be exposed to a fair amount of each language, and each of these languages are marketable with employers. I encourage you to play around with typesetting documents in RStudio!

> **!** STOP
>
> Replace the updated YAML header with the original code so that this document renders to .pdf. Ensure that the document renders to .pdf before continuing. Note that the YAML header is sensitive to indentation. If the document will not render to .pdf, please notify me.

Regression review

To guide our review of regression, we will use average annual temperature measurements collected in Vermont since 1850, courtesy of Berkeley Earth, and curated by Data World.

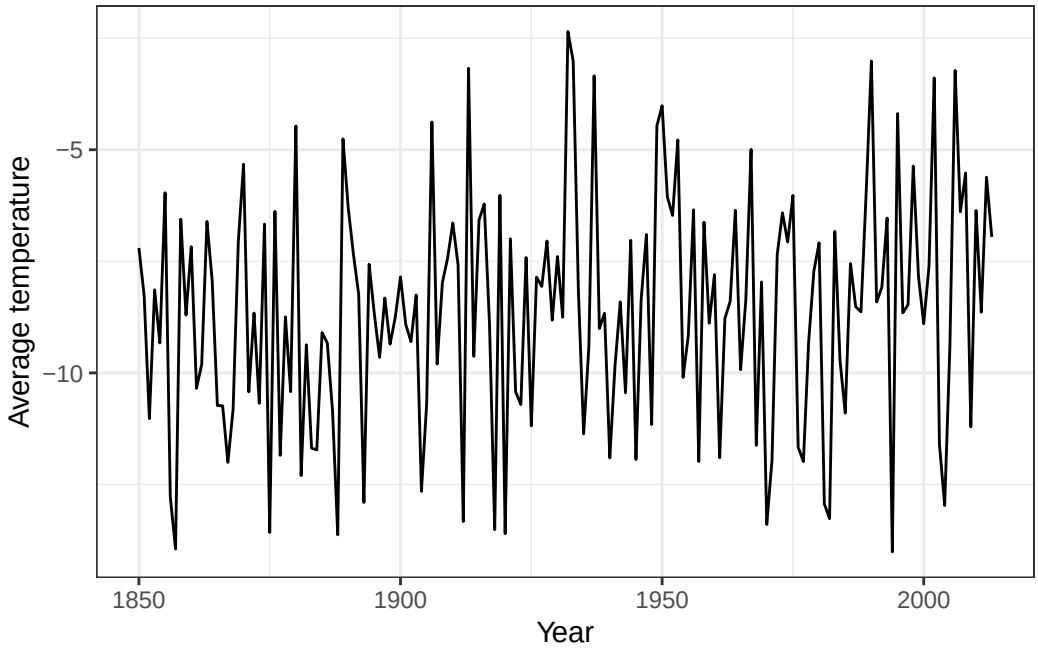
1. Pull the `vt_temps.csv` file into R, create new variables denoting year and month, and create a new object called `vt_jan_temps` that contains only measurements from January. Be sure your working directory is set to the source file location.

> **i** Note
>
> The `lubridate` package is very useful for manipulating `date` objects in R.

```r
# library packages
library(tidyverse);library(lubridate)

# read in data
vt_temps <- readr::read_csv("vt_temps.csv")
vt_temps <- read.csv("vt_temps.csv")

# create year and month variables
vt_jan_temps <- vt_temps |>
  mutate(
    dt = as.Date(dt),
    year = year(dt),
    month = month(dt)
  ) |>
  filter(month == 1)
```

2. Plot the average January temperature in Vermont by `dt`. Comment on the plot. Are there any noticeable trends?

```r
vt_jan_temps |>
  ggplot(aes(x = dt, y = AverageTemperature)) +
  geom_line() +
  theme_bw() +
  labs(
    x = "Year",
    y = "Average temperature"
  )
```



> Potentially a moderate, positive trend, but it is a very noisy signal with some cyclical patterns.

3. Fit a simple linear regression model with the average temperature as a response and the year as an explanatory variable. Interpret the slope coefficient. Is there evidence of a linear trend?

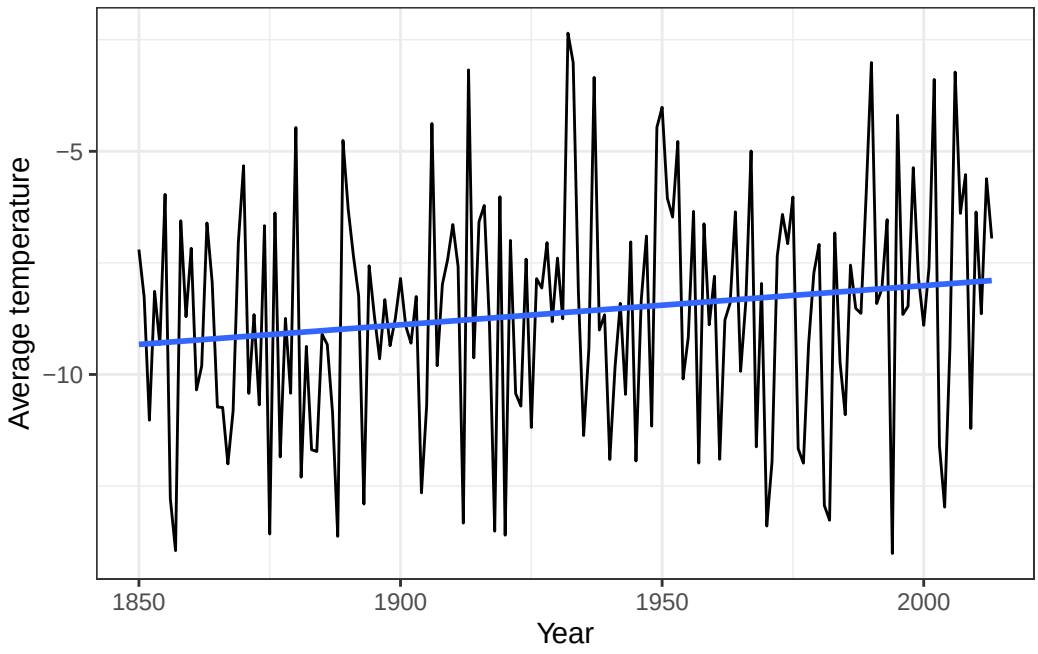
```
jan_lm <- lm(AverageTemperature ~ year, vt_jan_temps)
summary(jan_lm)
```

```
Call:
lm(formula = AverageTemperature ~ year, data = vt_jan_temps)

Residuals:
   Min     1Q Median     3Q    Max
-5.948 -1.622  0.056  1.642  6.253

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -25.579262   8.234796  -3.106  0.00224 **
year          0.008785   0.004262   2.061  0.04088 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.584 on 162 degrees of freedom
Multiple R-squared:  0.02556,    Adjusted R-squared:  0.01954
F-statistic: 4.248 on 1 and 162 DF,  p-value: 0.04088
```

> For each additional year, there is an estimated 0.00879 centigrade increase in the mean average temperature, on average. There is moderate evidence (p-value of 0.041) of a linear association between average temperature and year.

4. Plot the estimated regression model on the temperature time series. Does it appear to be a good fit to the data?

```
vt_jan_temps |>
  ggplot(aes(x = year, y = AverageTemperature)) +
  geom_line() +
  geom_smooth(method = "lm", se = F) +
  # geom_abline(aes(intercept = coef(jan_lm)[1], slope = coef(jan_lm)[2])) +
  theme_bw() +
  labs(
    x = "Year",
    y = "Average temperature"
  )
```

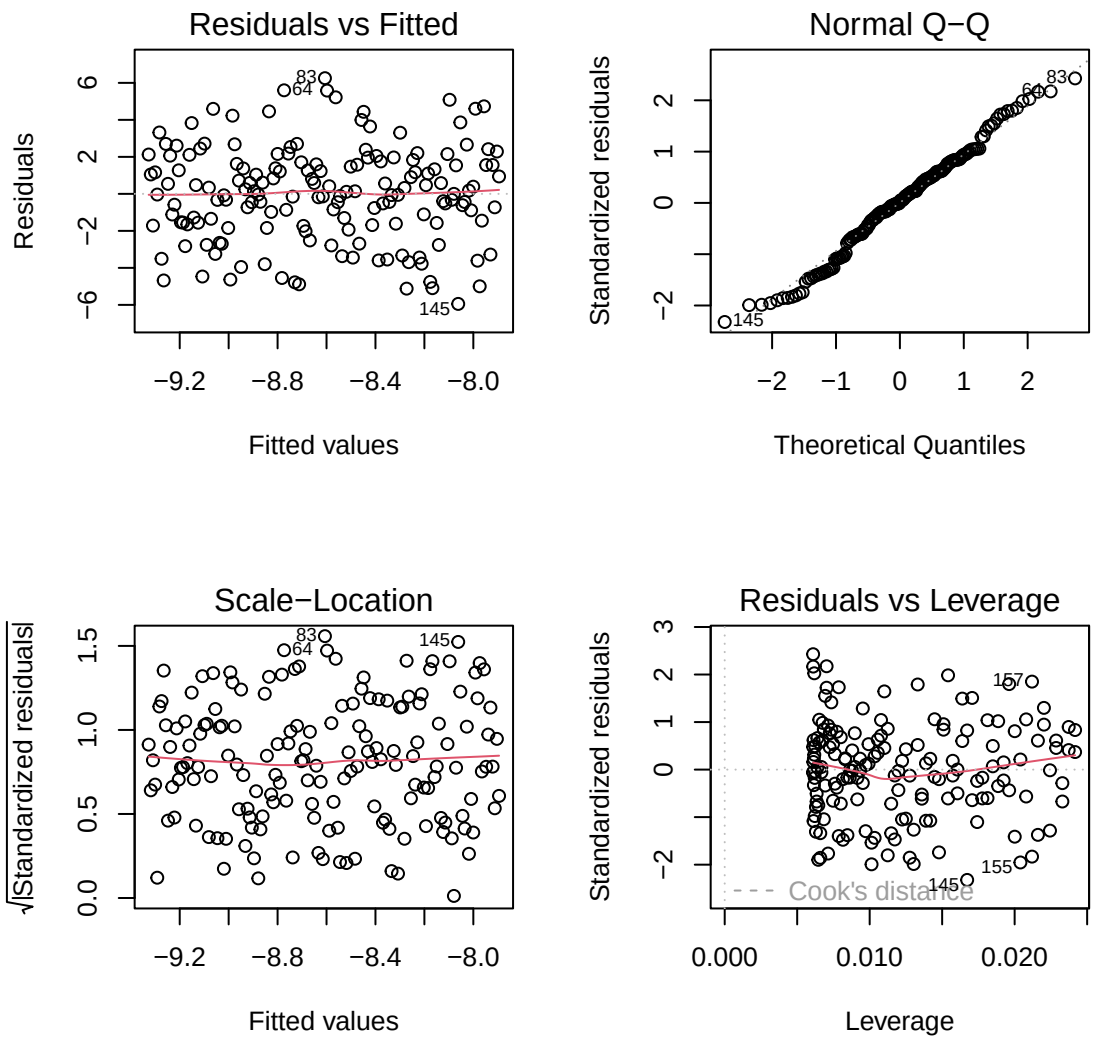


> Absolutely not. It captures the general trend okay, but does not address the "spikey" nature of the time series.

5. Assess the assumptions of the linear regression model; be sure to reference appropriate diagnostic plots. Are all the assumptions reasonably satisfied?

- *Independence:* We assume that the average temperature measurements from each time point are independent of one another. This assumption is almost certainly violated, as we expect temperatures from sequential months to be more similar than temperatures from months further apart in time.
- *Linearity:* We assume linearity of the residuals. The residuals vs fitted values plot presents as a random cloud of points with little-to-no leftover structure or trend, suggesting that this assumption is reasonably satisfied.
- *Constant variance:* We assume that the residuals have constant variance. We see approximately even vertical spread of the residuals for all fitted values in the residuals vs fitted values plot, suggesting that this assumption is reasonably satisfied.
- *Normality:* We assume that the residuals are normally distributed. There are no significant (in the english language sense, rather than statistical sense) departures from normality in the normal Q-Q plot, as the points follow the theoretical 1-1 quantile line fairly well. Again, this assumption appears reasonably satisfied.

```
par(mfrow = c(2,2))
plot(jan_lm)
```



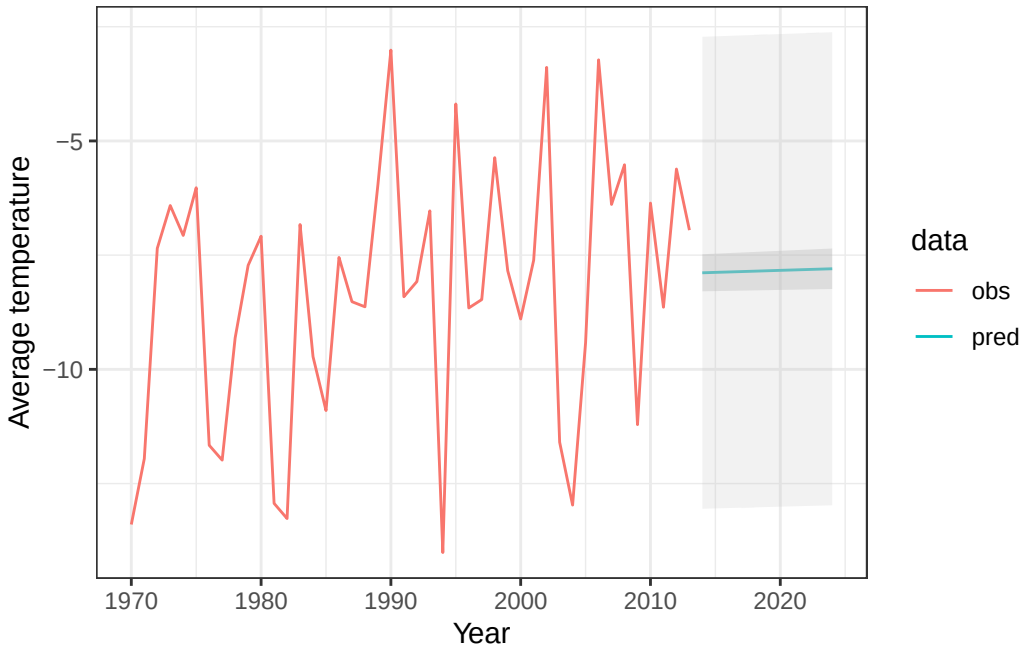

```
par(mfrow = c(1,1))
```

6. Using the linear regression model, predict the average January temperature for the years 2014 to 2024. Be sure to include a 95% prediction interval for each year. Comment on the quality of the prediction.

```
# get prediction intervals directly with R
pred <- predict(
  jan_lm,
  newdata = tibble(year = 2014:2024),
  interval = "prediction",
  level = 0.95
) |> as_tibble()
```

```r
# alternatively, calculate by hand with SEs (in case you're curious)
# pred_ <- predict(jan_lm, newdata = tibble(year = 2014:2024), se.fit = T)
# pred <- tibble(
#   fit = pred_$fit,
#   lwr = pred_$fit -
#     qt(.975, jan_lm$df.residual) *
#     sqrt(pred_$residual.scale^2 + pred_$se.fit^2),
#   upr = pred_$fit +
#     qt(.975, jan_lm$df.residual) *
#     sqrt(pred_$residual.scale^2 + pred_$se.fit^2)
# )

# create table for plotting
predict_tbl <- tibble(
  year = 2014:2024,
  AverageTemperature = pred$fit,
  lwrSE = pred$fit - predict(
    jan_lm, newdata = tibble(year = 2014:2024), se.fit = T
  )$se.fit,
  uprSE = pred$fit + predict(
    jan_lm, newdata = tibble(year = 2014:2024), se.fit = T
  )$se.fit,
  lwr95 = pred$lwr,
  upr95 = pred$upr
)

# plot prediction with intervals
# I have filtered to after 1970 to improve visibility of the prediction
bind_rows(
  vt_jan_temps %>% mutate(data = "obs"),
  predict_tbl %>% mutate(data = "pred")
) |>
  filter(year >= 1970) |>
  ggplot(aes(x = year, y = AverageTemperature)) +
  geom_line(aes(col = data, group = data)) +
  geom_ribbon(
    aes(x = year, ymin = lwrSE, ymax = uprSE), fill = "gray", alpha = 0.4
  ) +
  geom_ribbon(
    aes(x = year, ymin = lwr95, ymax = upr95), fill = "gray", alpha = 0.2
  ) +
  theme_bw() +
  labs(
```

```
    x = "Year",
    y = "Average temperature"
  )
```



Noticing a pattern here. The prediction is extremely poor, and does not capture the nuance and cyclical detail of the time series. The take away here is that classic linear regression is **not** well-suited to handle time series problems, both due to the poor prediction and violation of the independence assumption.