Advanced Epidemiological Analysis

Andreas Neophytou and G. Brooke Anderson

2021-04-05

Contents

Chapter 1

Overview

This is a the coursebook for the Colorado State University course ERHS 732, Advanced Epidemiological Analysis. This course provides the opportunity to implement theoretical expertise through designing and conducting advanced epidemiologic research analyses and to gain in-depth experience analyzing datasets from the environmental epidemiology literature.

1.1 License

Click on the **Next** button (or navigate using the links in the table of contents) to continue.

Chapter 2

Course information

This is a the coursebook for the Colorado State University course ERHS 732, Advanced Epidemiological Analysis. This course provides the opportunity to implement theoretical expertise through designing and conducting advanced epidemiologic research analyses and to gain in-depth experience analyzing datasets from the environmental epidemiology literature. This course will complement the student's training in advanced epidemiological methods, leveraging regression approaches and statistical programming, providing the opportunity to implement their theoretical expertise through designing and conducting advanced epidemiologic research analyses. During the course, students will gain in-depth experience analyzing two datasets from the environmental epidemiology literature—(1) time series data with daily measures of weather, air pollution, and cardiorespiratory outcomes in London, England and (2) a dataset with measures from the Framingham Heart Study. Additional datasets and studies will be discussed and explored as a supplement.

This class will utilize a variety of instructional formats, including short lectures, readings, topic specific examples from the substantive literature, discussion and directed group work on in-course coding exercises putting lecture and discussion content into practice. A variety of teaching modalities will be used, including group discussions, student directed discussions, and in-class group exercises. It is expected that before coming to class, students will read the required papers for the week, as well as any associated code included in the papers' supplemental materials. Students should come to class prepared to do statistical programming (i.e., bring a laptop with statistical software, download any datasets needed for the week etc). Participation is based on in-class coding exercises based on each week's topic. If a student misses a class, they will be expected to complete the in-course exercise outside of class to receive credit for participation in that exercise. Students will be required to do mid-term and final projects which will be presented in class and submitted as a written write-up describing the project.

Prerequisites for this course are:

- ERHS 534 or ERHS 535 and
- ERHS 640 and
- STAR 511 or STAT 511A or STAT 511B

2.1 Course learning objectives

The learning objectives for this proposed course complement core epidemiology and statistics courses required by the program and provide the opportunity for students to implement theoretical skills and knowledge gained in those courses in a more applied setting.

Upon successful completion of this course students will be able to:

1. List several possible statistical approaches to answering an epidemiological research questions. (*Knowledge*)
2. Choose among analytical approaches learned in previous courses to identify one that is reasonable for an epidemiological research question. (*Application*)
3. Design a plan for cleaning and analyzing data to answer an epidemiological research question, drawing on techniques learned in previous and concurrent courses. (*Synthesis*)
4. Justify the methods and code used to answer an epidemiological research question. (*Evaluation*)
5. Explain the advantages and limitations of a chosen methodological approach for evaluating epidemiological data. (*Evaluation*)
6. Apply advanced epidemiological methods to analyze example data, using a regression modeling framework. (*Application*)
7. Apply statistical programming techniques learned in previous courses to prepare epidemiological data for statistical analysis and to conduct the analysis. (*Application*)
8. Interpret the output from statistical analyses of data for an epidemiological research question. (*Evaluation*)
9. Defend conclusions from their analysis. (*Comprehension*)
10. Write a report describing the methods, results, and conclusions from an epidemiological analysis. (*Application*)
11. Construct a reproducible document with embedded code to clean and analyze data to answer an epidemiological research question. (*Application*)

2.2 Meeting time and place

[To be determined]

2.3 Class Structure and Expectations

- **Homework/preparation:** Every two weeks we will focus on a different topic. It is expected that *before* coming to class, students will read the required papers for the week, as well as any associated code included in the papers' supplemental materials. Students should come to class prepared to prepared to do statistical programming (i.e., bring in a laptop with statistical software, download any datasets needed for the week).
- **In-class schedule:**
  - Topic overview: Each class will start with a vocabulary quiz on a select number of the words from the chapter's vocabulary list.
  - Discussion of analysis and coding points: Students and faculty will be divided into small groups to discuss the chapter and think more deeply about the content. This is a time to bring up questions and relate the chapter concepts to other datasets and/or analysis methods you are familiar with.
  - Group work: In small groups, students will work on designing an epidemiological analysis for the week's topic and developing code to implement that analysis. Students will use the GitHub platform to work collaboratively during and between class meetings.
  - Wrap-up: We will reconvene as one group at the end to discuss topics that came up in small group work and to outline expectations for students before the next meeting.

2.4 Course grading

Assessment Components	Percentage of Grade
Midterm written report	30
Midterm presentation	15
Final written report	30
Final presentation	15
Participation in in-course exercises	10

2.5 Textbooks and Course Materials

Readings for this course will focus on peer-reviewed literature that will be posted for the students in the class. Additional references that will be useful to students throughout the semester include:

- Garrett Grolemund and Hadley Wickham, *R for Data Science*, O'Reilly, 2017. (Available for free online at https://r4ds.had.co.nz/ and in print through most large book sellers.)
- Miguel A. Hernan and James M. Robins, *Causal Inference: What If*, Boca Raton: Chapman & Hall/CRC, 2020. (Available for free online at https://cdn1.sph.harvard.edu/wp-content/uploads/sites/1268/2021/01/ciwha

tif_hernanrobins_31jan21.pdf with a print version anticipated in 2021.)

- Francesca Dominici and Roger D. Peng, *Statistical Methods for Environmental Epidemiology with R*, Springer, 2008. (Available online through the CSU library or in print through Springer.)

Chapter 3

Time series / case-crossover study designs

3.1 Reading

The readings for this week are:

- Vicedo-Cabrera et al. (2019), with supplemental material available to download by clicking http://links.lww.com/EDE/B504
- Armstrong et al. (2014), with supplemental material available at https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/1471-2288-14-122#Sec13

3.2 Time series data

[Exploring time series data with daily measurements of health outcomes and environmental exposures]

Andreas section

Brooke's section

Example datasets are available as part of the supplemental material for both of the articles in this chapter's readings. For Vicedo-Cabrera et al. (2019), the example data are available as the file "lndn_obs.csv". These data are saved in a csv format, and so they can be read into R using the `read_csv` function from the `readr` package (part of the tidyverse). For example, you can use the following code to read in these data, assuming you have saved them in a "data" subdirectory of your current working directory:

```
library(tidyverse) # Loads all the tidyverse packages, including readr
obs <- read_csv("data/lndn_obs.csv")
obs
```

```
## # A tibble: 8,279 x 14
##    date        year month   day   doy dow     all all_0_64 all_65_74 all_75_84
##    <date>     <dbl> <dbl> <dbl> <dbl> <chr> <dbl>    <dbl>     <dbl>     <dbl>
##  1 1990-01-01  1990     1     1     1 Mon     220       38        38        82
##  2 1990-01-02  1990     1     2     2 Tue     257       50        67        87
##  3 1990-01-03  1990     1     3     3 Wed     245       39        59        86
##  4 1990-01-04  1990     1     4     4 Thu     226       41        45        77
##  5 1990-01-05  1990     1     5     5 Fri     236       45        54        85
##  6 1990-01-06  1990     1     6     6 Sat     235       48        48        84
##  7 1990-01-07  1990     1     7     7 Sun     231       38        49        96
##  8 1990-01-08  1990     1     8     8 Mon     235       46        57        76
##  9 1990-01-09  1990     1     9     9 Tue     250       48        54        96
## 10 1990-01-10  1990     1    10    10 Wed     214       44        46        62
## # ... with 8,269 more rows, and 4 more variables: all_85plus <dbl>,
## #   tmean <dbl>, tmin <dbl>, tmax <dbl>
```

This example dataset shows many characteristics that are common for datasets for time series studies in environmental epidemiology. General characteristics of time series data for environmental epidemiology studies are:

- Observations are given at an aggregated level. For example, instead of individual observations for each person in London, the `obs` data give counts of deaths throughout London. The level of aggregation is often determined by geopolitical boundaries, for example counties of ZIP codes in the US.
- Observations are given at regularly spaced time steps over a period. In the `obs` dataset, the time step is day. Typically, values will be provided continuously over that time period, with observations for each time step. Occasionally, however, the time series data may only be available for particular seasons (e.g., only warm season dates for an ozone study), or there may be some missing data on either the exposure or health outcome over the course of the study period.
- Daily observations are given for the health outcome, for the environmental exposure of interest, and for potential time-varying confounders. In the `obs` dataset, the health outcome is mortality (from all causes; sometimes, the health outcome will focus on a specific cause of mortality or other health outcome). Counts are given for everyone in the city for each day (`all` column), as well as for specific age categories (`all_0_64` for all deaths among those up to 64 years old, and so on). The exposure of interest in the `obs` dataset is temperature, and three metrics of this are included (`tmean`, `tmin`, and `tmax`). Day of the week is one time-varying factor that could be a confounder, or at least help explain variation in the outcome (mortality). This is included through the `dow` variable in the `obs` data.

Sometimes, you will also see a marker for holidays included as a potential time-varying confounder, or other exposure variables (temperature is a potential confounder, for example, when investigating the relationship between air pollution and mortality risk).

- Multiple metrics of an exposure and / or multiple health outcome counts may be included for each time step. In the `obs` example, three metrics of temperature are included (minimum daily temperature, maximum daily temperature, and mean daily temperature). Several counts of mortality are included, providing information for specific age categories in the population.

When working with time series data, it is helpful to start with some exploratory data analysis. The following applied exercise will take you through some of the questions you might want to answer through this type of exploratory analysis. In general, the `lubridate` package is an excellent tool for working with date data in R (although, in the example code above, we mostly used tools from base R). You may find it worthwhile to explore this package some more. There is a helpful chapter in Wickham and Grolemund (2016), https://r4ds.had.co.nz/dates-and-times.html, as well as a cheatsheet at https://evoldyn.gitlab.io/evomics-2018/ref-sheets/R_lubridate.pdf. For visualizations, if you are still learning techniques in R, two books you may find useful are Healy (2018) (available online at https://socviz.co/) and Chang (2018) (available online at http://www.cookbook-r.com/Graphs/).

Applied: Exploring time series data

Read the example time series data in R and explore it to answer the following questions:

1. What is the study period for the example `obs` dataset? (i.e., what dates / years are covered by the time series data?)
2. Are there any missing dates within this time period?
3. Are there seasonal trends in the exposure? In the outcome?
4. Are there long-term trends in the exposure? In the outcome?
5. Is the outcome associated with day of week? Is the exposure associated with day of week?

Based on your exploratory analysis in this section, talk about the potential for confounding when these data are analyzed to estimate the association between daily temperature and city-wide mortality. Is confounding by seasonal trends a concern? How about confounding by long-term trends in exposure and mortality? How about confounding by day of week?

Applied exercise: Example code

1. **What is the study period for the example `obs` dataset? (i.e., what dates / years are covered by the time series data?)**

In the `obs` dataset, the date of each observation is included in a column called `date`. The data type of this column is “Date”—you can check this by using the `class` function from base R:

```
class(obs$date)
```

```
## [1] "Date"
```

Since this column has a "Date" data type, you can run some mathematical function calls on it. For example, you can use the `min` function from base R to get the earliest date in the dataset and the `max` function to get the latest.

```
min(obs$date)
```

```
## [1] "1990-01-01"
```

```
max(obs$date)
```

```
## [1] "2012-08-31"
```

You can also run the `range` function to get both the earliest and latest dates with a single call:

```
range(obs$date)
```

```
## [1] "1990-01-01" "2012-08-31"
```

2. Are there any missing dates within this time period?

There are a few things you should check to answer this question. First (and easiest), you can check to see if there are any `NA` values within any of the observations in the dataset. The `summary` function will provide a summary of the values in each column of the dataset, including the count of missing values (NAs) if there are any:

```
summary(obs)
```

```
##       date                 year          month             day
##  Min.   :1990-01-01   Min.   :1990   Min.   : 1.000   Min.   : 1.00
##  1st Qu.:1995-09-01   1st Qu.:1995   1st Qu.: 3.000   1st Qu.: 8.00
##  Median :2001-05-02   Median :2001   Median : 6.000   Median :16.00
##  Mean   :2001-05-02   Mean   :2001   Mean   : 6.464   Mean   :15.73
##  3rd Qu.:2006-12-31   3rd Qu.:2006   3rd Qu.: 9.000   3rd Qu.:23.00
##  Max.   :2012-08-31   Max.   :2012   Max.   :12.000   Max.   :31.00
##       doy             dow                 all           all_0_64
##  Min.   :  1.0   Length:8279        Min.   : 81.0   Min.   : 9.0
##  1st Qu.: 90.5   Class :character   1st Qu.:138.0   1st Qu.:27.0
##  Median :180.0   Mode  :character   Median :157.0   Median :32.0
##  Mean   :181.3                      Mean   :160.2   Mean   :32.4
##  3rd Qu.:272.0                      3rd Qu.:178.0   3rd Qu.:37.0
##  Max.   :366.0                      Max.   :363.0   Max.   :64.0
##    all_65_74       all_75_84       all_85plus         tmean
##  Min.   : 6.00   Min.   : 17.00   Min.   : 17.00   Min.   :-5.503
##  1st Qu.:23.00   1st Qu.: 41.00   1st Qu.: 39.00   1st Qu.: 7.470
```

```
##  Median :29.00   Median : 49.00   Median : 45.00   Median :11.465
##  Mean   :30.45   Mean   : 50.65   Mean   : 46.68   Mean   :11.614
##  3rd Qu.:37.00   3rd Qu.: 58.00   3rd Qu.: 53.00   3rd Qu.:15.931
##  Max.   :70.00   Max.   :138.00   Max.   :128.00   Max.   :29.143
##       tmin             tmax
##  Min.   :-8.940   Min.   :-3.785
##  1st Qu.: 3.674   1st Qu.:10.300
##  Median : 7.638   Median :14.782
##  Mean   : 7.468   Mean   :15.058
##  3rd Qu.:11.438   3rd Qu.:19.830
##  Max.   :20.438   Max.   :37.087
```

Based on this analysis, all observations are complete for all dates included in the dataset.

However, this does not guarantee that every date between the start date and end date of the study period are included in the recorded data. Sometimes, some dates might not get recorded at all in the dataset, and the `summary` function won't help you determine when this is the case.

There are a few alternative explorations you can do. First, you can check the number of days between the start and end date of the study period, and then see if the number of observations in the dataset is the same:

```
# Calculate number of days in study period
obs %>%             # Using piping (%>%) throughout to keep code clear
  pull(date) %>%    # Extract the `date` column as a vector
  range() %>%       # Take the range of dates (earliest and latest)
  diff()            # Calculate time difference from start to finish of study
```

```
## Time difference of 8278 days
```

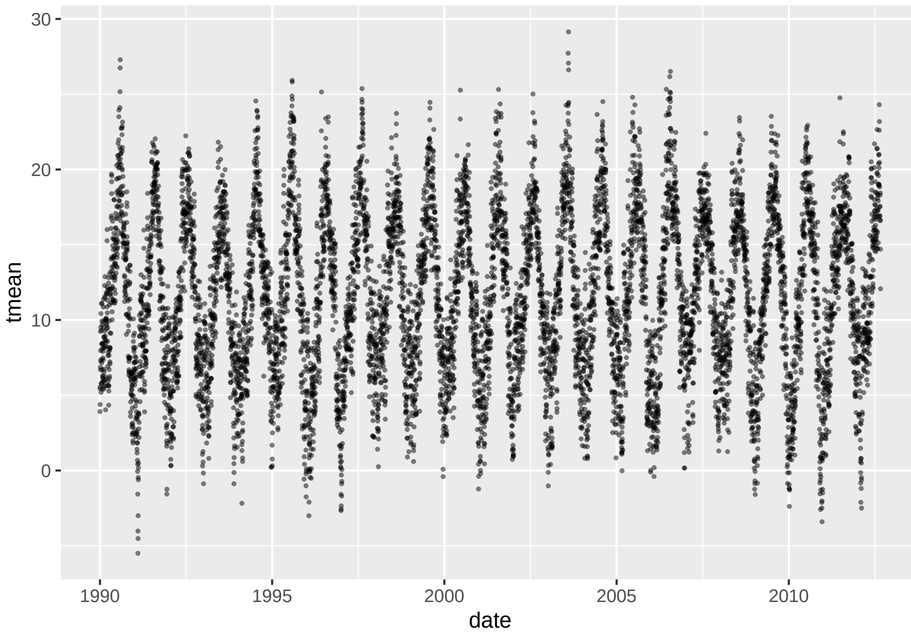
```
# Get number of observations in dataset---should be 1 more than time difference
obs %>%
  nrow()
```

```
## [1] 8279
```

3. Are there seasonal trends in the exposure? In the outcome?

You can use a simple plot to visualize patterns over time in both the exposure and the outcome. For example, the following code plots a dot for each daily temperature observation over the study period. The points are set to a smaller size (`size = 0.5`) and plotted with some transparency (`alpha = 0.5`) since there are so many observations.

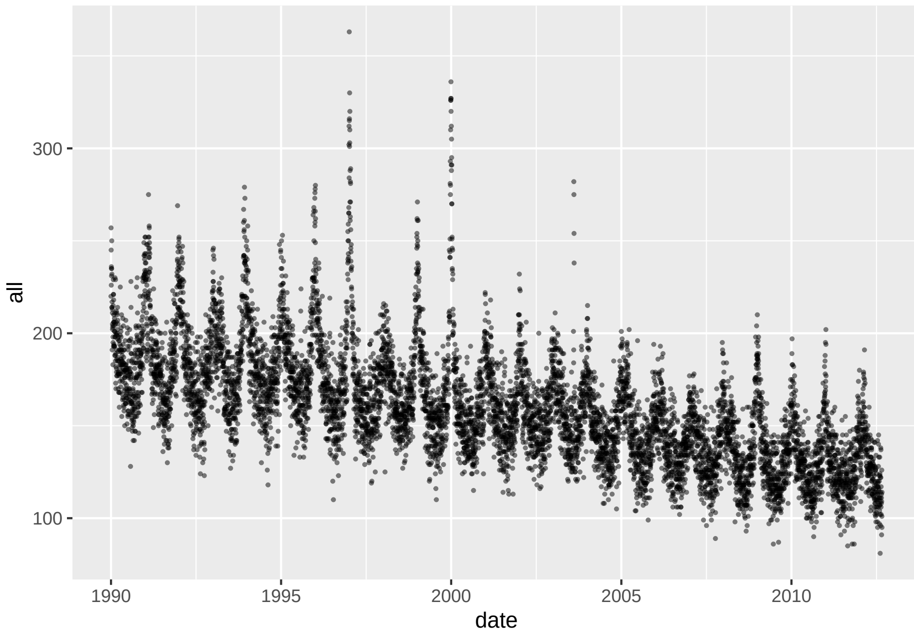
```
ggplot(obs, aes(x = date, y = tmean)) +
  geom_point(alpha = 0.5, size = 0.5)
```



There is clear evidence here of a strong seasonal trend in mean temperature, with values typically lowest in the winter and highest in the summer.

You can plot the outcome variable in the same way:

```
ggplot(obs, aes(x = date, y = all)) +
  geom_point(alpha = 0.5, size = 0.5)
```

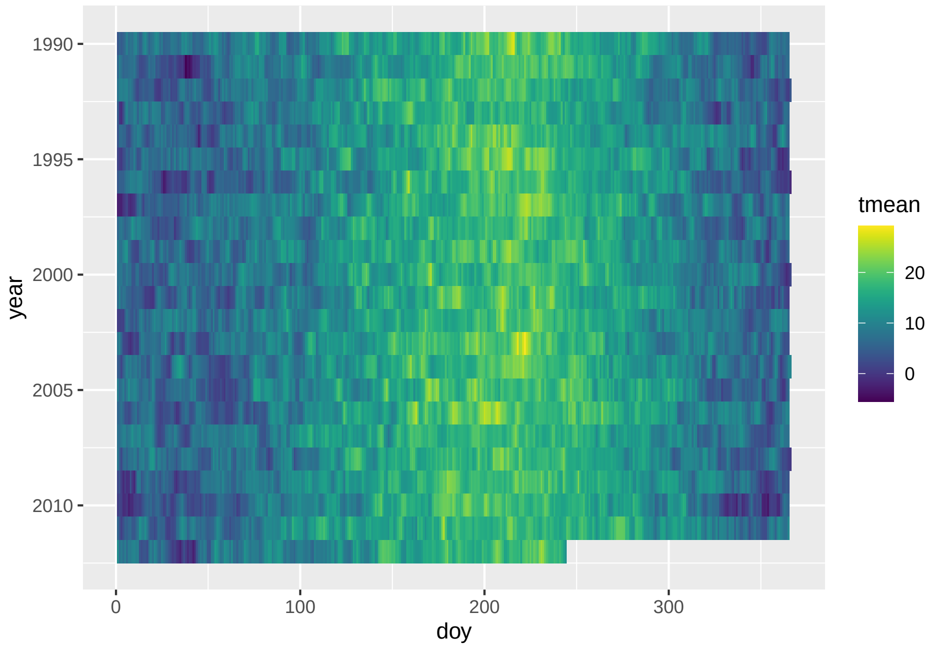


Again, there are seasonal trends, although in this case they are inversed. Mortality tends to be highest in the winter and lowest in the summer. Further, the seasonal pattern is not equally strong in all years—some years it has a much higher winter peak, probably in conjunction with severe influenza seasons.

Another way to look for seasonal trends is with a heatmap-style visualization, with day of year along the x-axis and year along the y-axis. This allows you to see patterns that repeat around the same time of the year each year (and also unusual deviations from normal seaonsal patterns).

For example, here's a plot showing temperature in each year, where the observations are aligned on the x-axis by time in year. We've reversed the y-axis so that the earliest years in the study period start at the top of the visual, then later study years come later—this is a personal style, and it would be no problem to leave the y-axis as-is. We've used the `viridis` color scale for the fill, since that has a number of features that make it preferable to the default R color scale, including that it is perceptible for most types of color blindness and be printed out in grayscale and still be correctly interpreted.

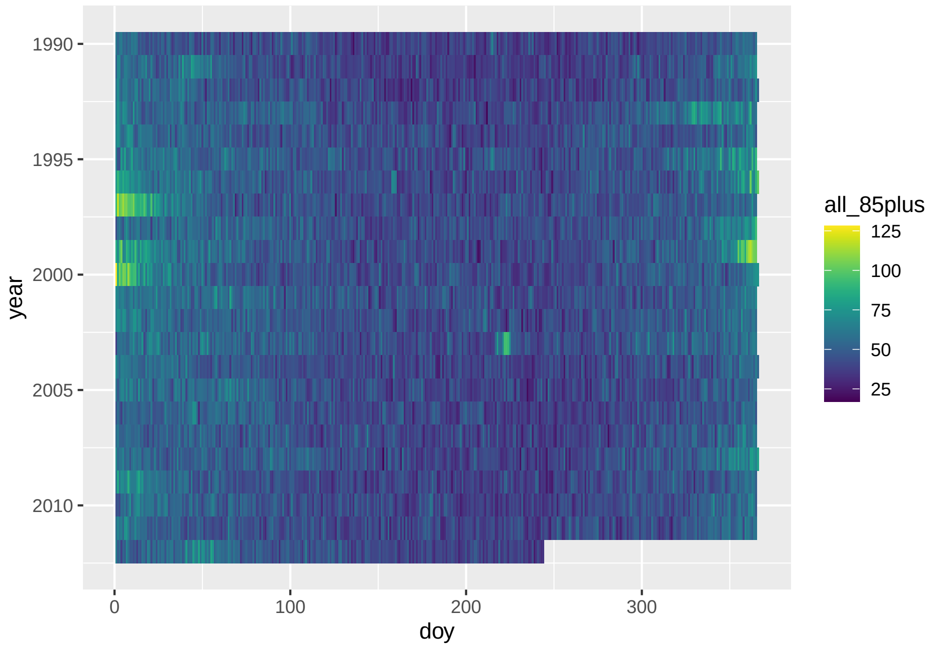
```
library(viridis)
ggplot(obs, aes(x = doy, y = year, fill = tmean)) +
  geom_tile() +
  scale_y_reverse() +
  scale_fill_viridis()
```



From this visualization, you can see that temperatures tend to be higher in the summer months and lower in the winter months. “Spells” of extreme heat or cold are visible—where extreme temperatures tend to persist over a period, rather than randomly fluctuating within a season. You can also see unusual events, like the extreme heat wave in the summer of 2003, indicated with the brightest yellow in the plot.

We created the same style of plot for the health outcome. In this case, we focused on mortality among the oldest age group, as temperature sensitivity tends to increase with age, so this might be where the strongest patterns are evident.

```
ggplot(obs, aes(x = doy, y = year, fill = all_85plus)) +
  geom_tile() +
  scale_y_reverse() +
  scale_fill_viridis()
```

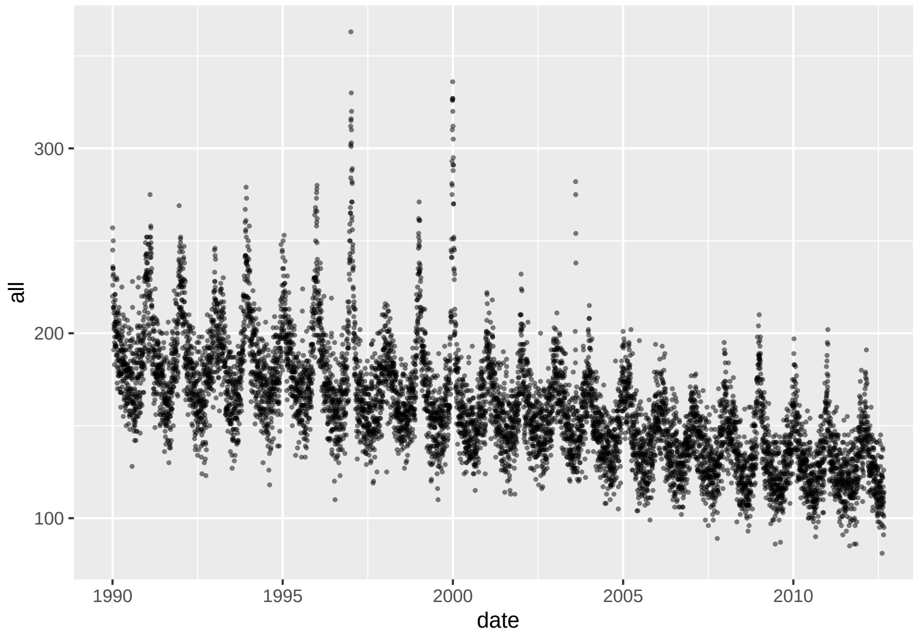


For mortality, there tends to be in increase in the winter compared to the summer. Some winters have stretches with particularly high mortality—these are likely a result of seasons with strong influenza outbreaks. You can also see on this plot the impact of the 2003 heat wave on mortality among this oldest age group.

4. Are there long-term trends in the exposure? In the outcome?

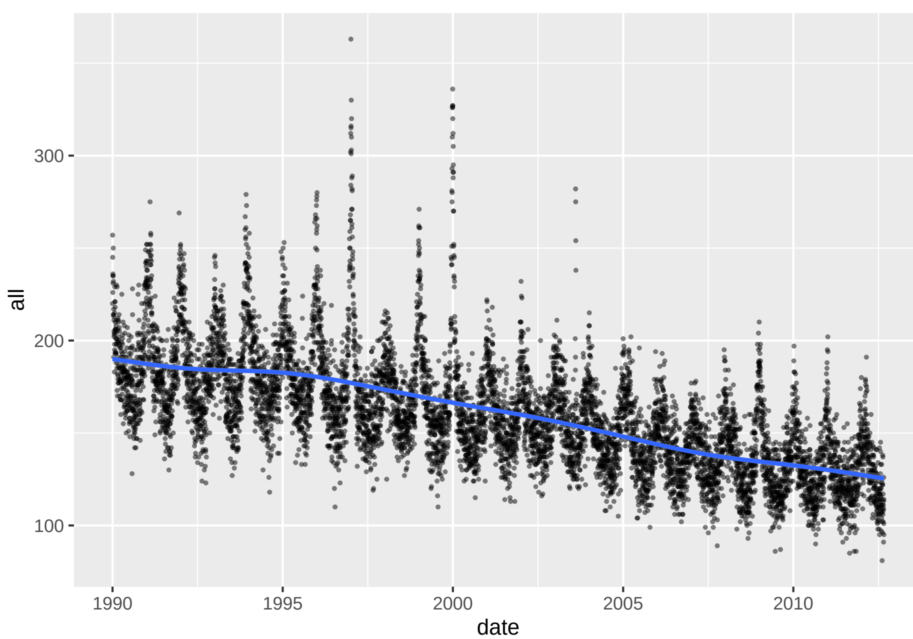
Some of the plots we created in the last section help in exploring this question. For example, the following plot shows a clear pattern of decreasing daily mortality counts, on average, over the course of the study period:

```
ggplot(obs, aes(x = date, y = all)) +
  geom_point(alpha = 0.5, size = 0.5)
```



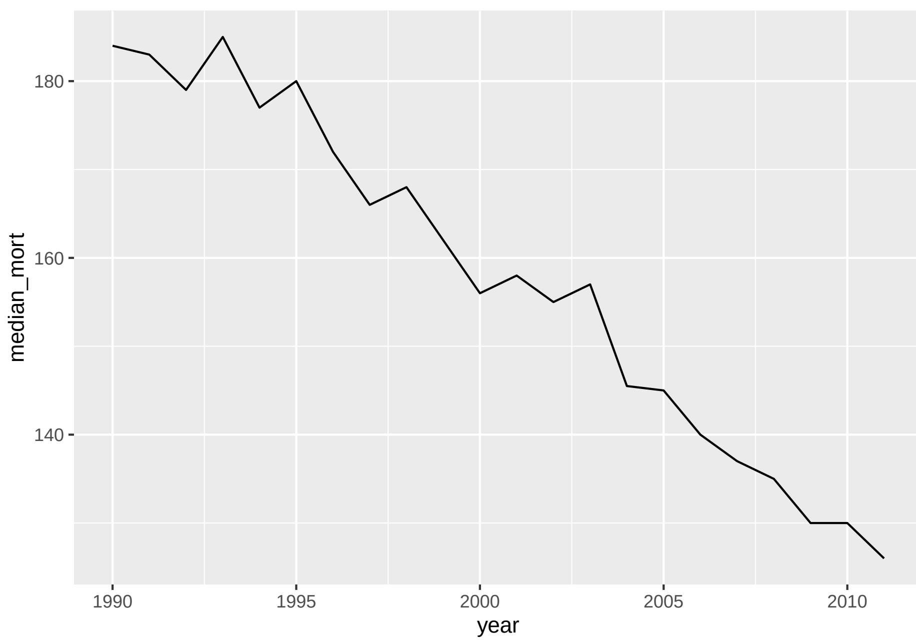
It can be helpful to add a smooth line to help detect these longer-term patterns, which you can do with `geom_smooth`:

```
ggplot(obs, aes(x = date, y = all)) +
  geom_point(alpha = 0.5, size = 0.5) +
  geom_smooth()
```



You could also take the median mortality count across each year in the study period, although you should take out any years without a full year's worth of data before you do this, since there are seasonal trends in the outcome:

```
obs %>%
  group_by(year) %>%
  filter(year != 2012) %>% # Take out the last year
  summarize(median_mort = median(all)) %>%
  ggplot(aes(x = year, y = median_mort)) +
  geom_line()
```



5. **Is the outcome associated with day of week? Is the exposure associated with day of week?**

The data already has day of week as a column in the data (`dow`). However, this is in a character data type, so it doesn't have the order of weekdays encoded (e.g., Monday comes before Tuesday). This makes it hard to look for patterns related to things like weekend / weekday.

```
class(obs$dow)
```

```
## [1] "character"
```

We could convert this to a factor and encode the weekday order when we do it, but it's even easier to just recreate the column from the `date` column. We used the `wday` function from the `lubridate` package to do this—it extracts weekday as a factor, with the order of weekdays encoded (using a special "ordered" factor type):

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
obs <- obs %>%
  mutate(dow = wday(date, label = TRUE))

class(obs$dow)
```

```
## [1] "ordered" "factor"
```

```
levels(obs$dow)
```

```
## [1] "Sun" "Mon" "Tue" "Wed" "Thu" "Fri" "Sat"
```

We looked at the mean, median, and 25th and 75th quantiles of the mortality counts by day of week:

```
obs %>%
  group_by(dow) %>%
  summarize(mean(all),
            median(all),
            quantile(all, 0.25),
            quantile(all, 0.75))
```

```
## # A tibble: 7 x 5
##   dow   `mean(all)` `median(all)` `quantile(all, 0.25)` `quantile(all, 0.75)`
## * <ord>       <dbl>         <dbl>                 <dbl>                 <dbl>
## 1 Sun          156.           154                  136                    173
## 2 Mon          161.           159                  138                    179
## 3 Tue          161.           158                  139                    179
## 4 Wed          160.           157                  138.                   179
## 5 Thu          161.           158                  139                    179
## 6 Fri          162.           159                  141                    179
## 7 Sat          159.           156                  137                    178
```

Mortality tends to be a bit higher on weekdays than weekends, but it's not a dramatic difference.

We did the same check for temperature:

```
obs %>%
  group_by(dow) %>%
  summarize(mean(tmean),
            median(tmean),
```

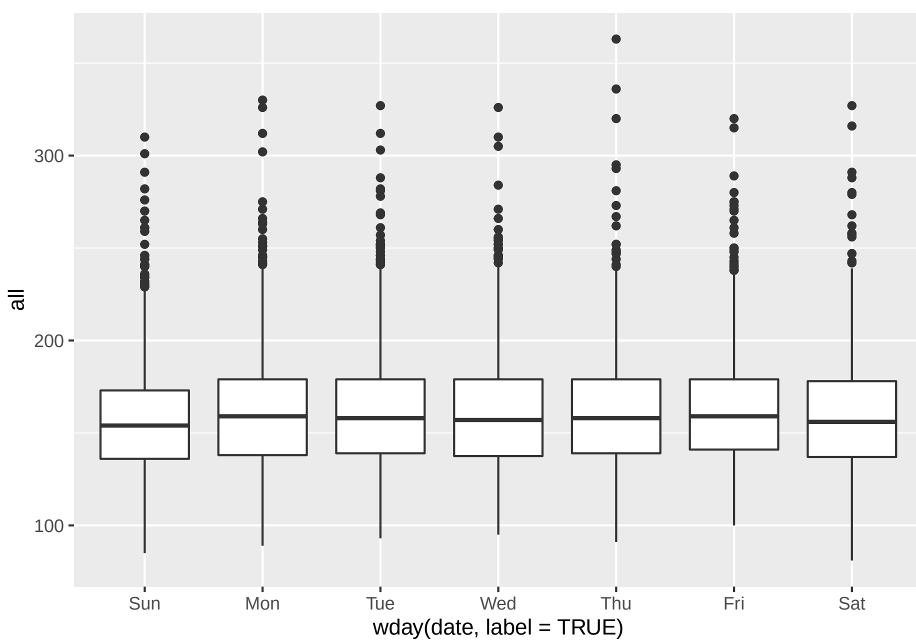
```
            quantile(tmean, 0.25),
            quantile(tmean, 0.75))
```

```
## # A tibble: 7 x 5
##   dow   `mean(tmean)` `median(tmean)` `quantile(tmean, 0.2~ `quantile(tmean, 0.~
## * <ord>         <dbl>           <dbl>                 <dbl>                <dbl>
## 1 Sun            11.6            11.3                  7.48                 15.9
## 2 Mon            11.6            11.4                  7.33                 15.8
## 3 Tue            11.5            11.4                  7.48                 15.9
## 4 Wed            11.7            11.7                  7.64                 16.0
## 5 Thu            11.6            11.5                  7.57                 16.0
## 6 Fri            11.6            11.6                  7.41                 15.8
## 7 Sat            11.6            11.5                  7.53                 15.9
```

In this case, there does not seem to be much of a pattern by weekday.

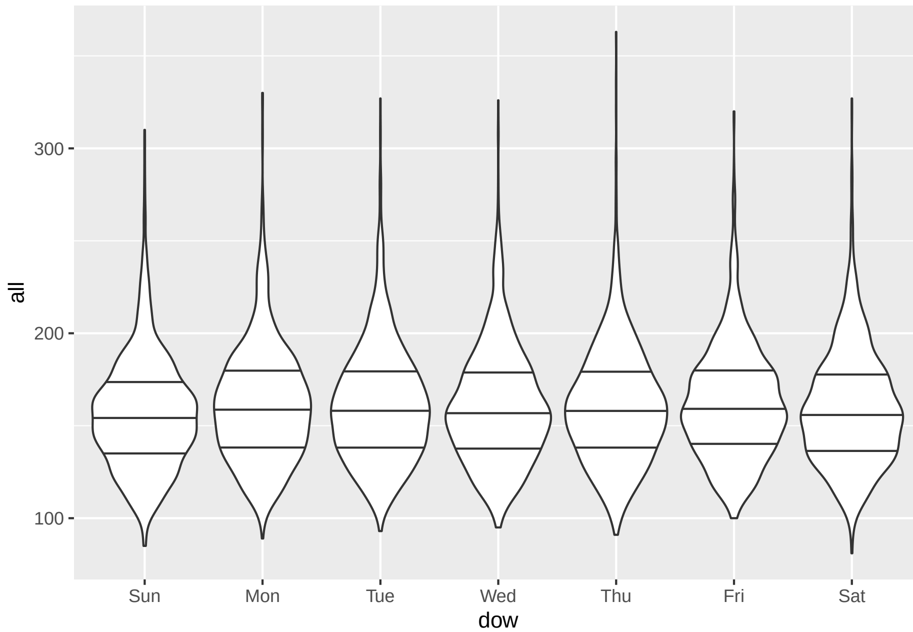
You can also visualize the association using boxplots:

```
ggplot(obs, aes(x = wday(date, label = TRUE), y = all)) +
  geom_boxplot()
```



You can also try violin plots—these show the full distribution better than boxplots, which only show quantiles.

```
ggplot(obs, aes(x = dow, y = all)) +
  geom_violin(draw_quantiles = c(0.25, 0.5, 0.75))
```



3.3 Fitting models

[Fitting models under time series and case-crossover study designs]

Andreas section

Brooke's section

3.4 Chapter vocabulary

Each class will start with a vocabulary quiz on a select number of the words from the chapter's vocabulary list. The vocabulary words for this chapter are:

- time-series study design
- case-crossover study design
- exposure
- health outcome
- confounder
- study period
- seasonal trends
- long-term trends

Chapter 4

Generalized linear models

4.1 Splines in GLMs

[Using splines to model non-linear associations in a GLM]

4.2 Cross-basis functions in GLMs

[Using a cross-basis to model an exposure's association with the outcome in two dimensions (dimensions of time and exposure level)]

Chapter 5

Natural experiments

5.1 Interrupted time series

[Interrupted time series assessing effects of policy/intervention in specific point in time]

5.2 Difference-in-differences

[Difference-in differences application for intervention introduced in one point in time]

Chapter 6

Risk assessment

[Predict expected heat-related mortality under a climate change scenario]

Chapter 7

Longitudinal cohort study designs

7.1 Coding a survival analysis

7.2 Handling complexity

7.2.1 Multi-level exposure

7.2.2 Recurrent outcome

7.2.3 Time-varying coeffificents

7.2.4 Using survey results

[e.g., NHANES]

Chapter 8

Some approaches for confounding

8.1 Inverse probability weighting

8.2 Propensity scores

[Modeling for weights/propensity scores, involves machine learning]

Chapter 9

Mixed models

[Using a mixed modeling framework to help analyze repeated measures]

Chapter 10

Instrumental variables

Chapter 11

Causal inference

Bibliography

Armstrong, B. G., Gasparrini, A., and Tobias, A. (2014). Conditional Poisson models: a flexible alternative to conditional logistic case cross-over analysis. *BMC Medical Research Methodology*, 14(1):122.

Chang, W. (2018). *R graphics cookbook: practical recipes for visualizing data.* O'Reilly Media.

Healy, K. (2018). *Data visualization: a practical introduction.* Princeton University Press.

Vicedo-Cabrera, A. M., Sera, F., and Gasparrini, A. (2019). Hands-on tutorial on a modeling framework for projections of climate change impacts on health. *Epidemiology*, 30(3):321–329.

Wickham, H. and Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data.* " O'Reilly Media, Inc.".