Week 3
Relationships & Comparisons

Data Visualization for Social Scientists

Lecturer: Jeffrey Ziegler, PhD
Teaching Fellow: Shekhar Kedia

ASDS - Trinity College Dublin

Spring 2026

Road map for today

- Today:
  - Dual y-axes in ggplot()

Road map for today

- Today:
  - Dual y-axes in ggplot()
  - Combining plots with `patchwork`

Road map for today

- Today:
  - Dual y-axes in ggplot()
  - Combining plots with `patchwork`
  - Correlations, scatterplot matrices, and correlograms

Road map for today

- Today:
  - Dual y-axes in ggplot()
  - Combining plots with `patchwork`
  - Correlations, scatterplot matrices, and correlograms
  - Regression, predictions, and marginal effectsd plot them with `ggplot2`

Road map for today

- Today:
  - Dual y-axes in ggplot()
  - Combining plots with `patchwork`
  - Correlations, scatterplot matrices, and correlograms
  - Regression, predictions, and marginal effectsd plot them with `ggplot2`
- By next week, please...
  - Problem set #3

Legal dual y-axes

- Safe when both axes measure same underlying quantity

Legal dual y-axes

- Safe when both axes measure same underlying quantity
  - Ex: counts vs %, Fahrenheit vs Celsius, pounds vs kilograms

Legal dual y-axes

- Safe when both axes measure same underlying quantity
  - Ex: counts vs %, Fahrenheit vs Celsius, pounds vs kilograms
- In ggplot2: use `sec.axis` inside `scale_y_continuous()`

Ex: Dual y-axes: Fahrenheit & Celsius

- Historical weather for Dublin: Met Éireann
- Variables include temperature, humidity, wind speed, precipitation, etcvspace*.25cm

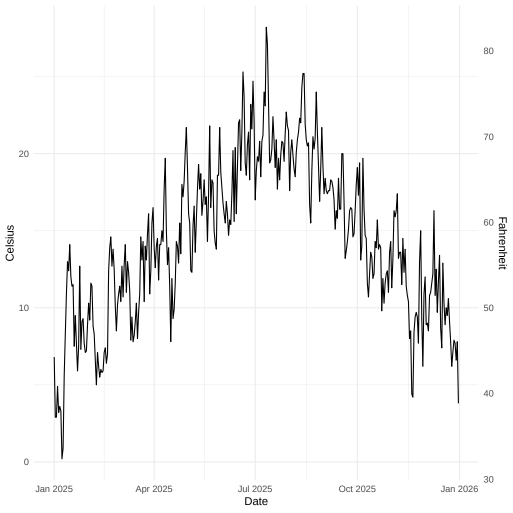
Ex: Dual y-axes: Fahrenheit & Celsius

- Historical weather for Dublin: Met Éireann
- Variables include temperature, humidity, wind speed, precipitation, etcvspace*.25cm
- C to F conversion: $\mathrm{F} = (\mathrm{C}) \times \frac{9}{5}) + 32$

Ex: Max temp by day in 2025 for Dublin

```
# we'll only look at 2025
DUB_weather <- DUB_weather[30317:nrow(DUB_weather),]
CORK_weather <- CORK_weather[23012:nrow(CORK_weather),]
IRE_weather <- rbind(subset(DUB_weather, select=-c(g_rad)), CORK_weather)
# need to alter date variable to not be character
IRE_weather$date <- dmy(DUB_weather$date)
```

```
ggplot(IRE_weather[IRE_weather$station=="Dublin",], aes(x = date, y = maxtp)) +
  geom_line() +
  scale_y_continuous(sec.axis = sec_axis(trans = ~ (. * 9/5) + 32, name = "Fahrenheit")) +
  labs(x = "Date", y = "Celsius") +
  theme_minimal()
```



Combining plots with patchwork

- Alternative to dual axes: separate, aligned plots

Combining plots with patchwork

- Alternative to dual axes: separate, aligned plots
- `patchwork` combines ggplots
  - To use `patchwork`, we need to (1) save our plots as objects and (2) add them together with +

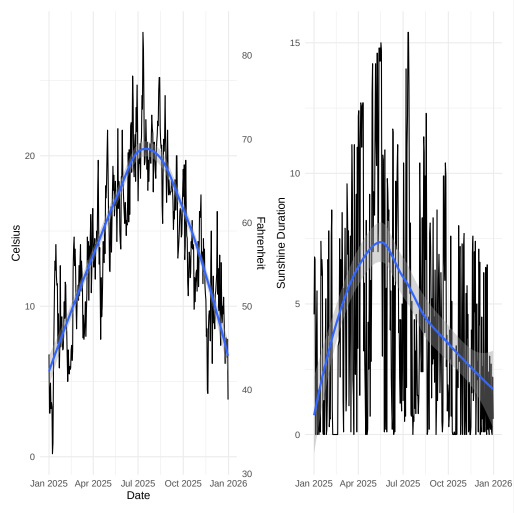
Combining plots with patchwork

- Alternative to dual axes: separate, aligned plots
- `patchwork` combines ggplots
    - To use `patchwork`, we need to (1) save our plots as objects and (2) add them together with `+`
- Supports layouts, relative heights, and more

Ex: Relationship of temp & sunshine duration

```
sun_plot <- ggplot(IRE_weather[IRE_
      weather$station=="Dublin",],
      aes(x = date, y = sun)) +
  geom_line() +
  geom_smooth() +
  labs(x = NULL, y = "Sunshine
      Duration") +
  theme_minimal()
```

```
# library(patchwork)
temp_plot + sun_plot
```



Comparing time series

- Many-group (i.e., country, city) line charts can become difficult to read

Comparing time series

- Many-group (i.e., country, city) line charts can become difficult to read
- Strategies:

Comparing time series

- Many-group (i.e., country, city) line charts can become difficult to read
- Strategies:
  - Highlight a few focal groups

Comparing time series

- Many-group (i.e., country, city) line charts can become difficult to read
- Strategies:
  - Highlight a few focal groups
  - Use small multiples (facets)

- Many-group (i.e., country, city) line charts can become difficult to read
- Strategies:
  - Highlight a few focal groups
  - Use small multiples (facets)
  - Use map-based facet layouts with `geofacet`
    - `geofacet` arranges country facets in map-like grids

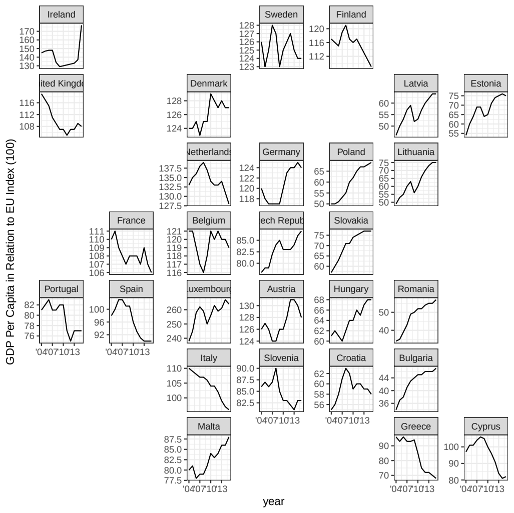
Comparing time series

- Many-group (i.e., country, city) line charts can become difficult to read
- Strategies:
  - Highlight a few focal groups
  - Use small multiples (facets)
  - Use map-based facet layouts with `geofacet`
    - `geofacet` arranges country facets in map-like grids
    - Preserves some geographic structure while showing small multiples

Comparing time series

- Many-group (i.e., country, city) line charts can become difficult to read
- Strategies:
  - Highlight a few focal groups
  - Use small multiples (facets)
  - Use map-based facet layouts with `geofacet`
    - `geofacet` arranges country facets in map-like grids
    - Preserves some geographic structure while showing small multiples
    - Useful for regional or global comparisons

Geofacet example

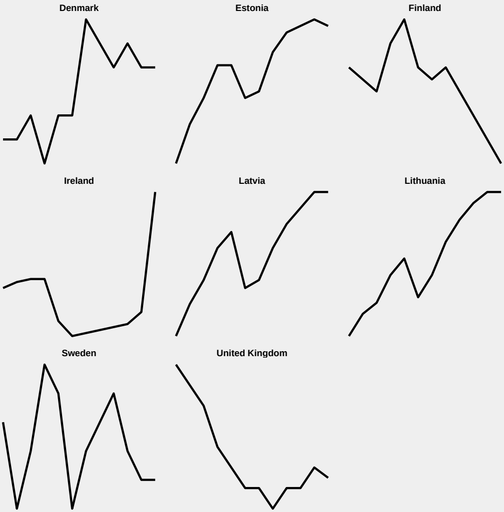



```r
# library("geofacet")
ggplot(eu_gdp, aes(year, gdp_pc)) +
  geom_line() +
  facet_geo(~ name, grid = "eu_
      grid1", scales = "free_y") +
  scale_x_continuous(labels =
      function(x) paste0("'", substr
      (x, 3, 4))) +
  ylab("GDP Per Capita in Relation
      to EU Index (100)") +
  theme_bw()
```

Ex: Small multiples - EU GDP in Northern Europe

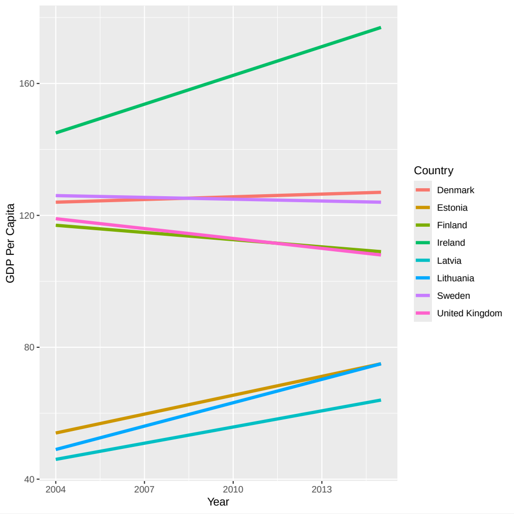
```r
# create "Northern Europe" indicator based on UN geoscheme
eu_gdp$north_europe <- ifelse(eu_gdp$name %in% c("Denmark", "Estonia", "Finland", "Ireland", "Latvia", "Lithuania", "Norway", "Sweden", "United Kingdom"), 1, 0)
```

```r
ggplot(data = eu_gdp |> filter(north_europe == 1), aes(x = year, y = gdp_pc)) +
  geom_line(linewidth = 1) +
  facet_wrap(vars(name), scales = "free_y", nrow = 3) +
  theme_void() +
  theme(strip.text = element_text(face = "bold"))
```



Slopegraphs

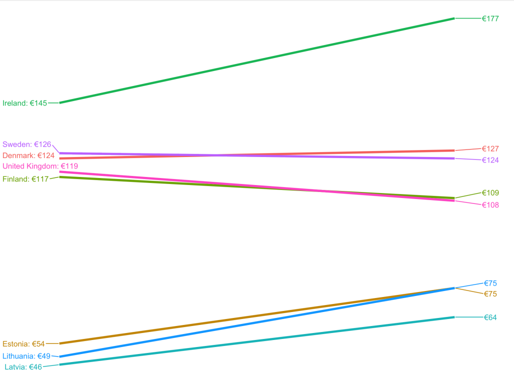
Show only change in GDP per capita between two time periods




```r
north_europe_gdp <- eu_gdp |>
    filter(north_europe == 1) |>
    filter(year %in% c(2004, 2015)) |>
  mutate(label_first = ifelse(year == 2004, paste0(name, ": ", label_dollar(prefix = "   ")(round(gdp_pc))), NA),
         label_last = ifelse(year == 2015, label_dollar(prefix = "   ")(round(gdp_pc, 0)), NA))
ggplot(north_europe_gdp, aes(x = year, y = gdp_pc, group = name, color = name)) +
  geom_line(linewidth = 1.5) +
  labs(y="GDP Per Capita", x="Year", color="Country")
```

Labeling selected points with ggrepel

```r
ggplot(north_europe_gdp, aes(x = year, y = gdp_pc, group = name, color = name)) +
  geom_line(linewidth = 1.5) +
  geom_text_repel(aes(label = label_first), direction = "y", nudge_x = -1, seed = 1234) +
  geom_text_repel(aes(label = label_last), direction = "y", nudge_x = 1, seed = 1234) +
  guides(color = "none") +
  theme_void()
```



Design considerations

- Use sorting and consistent scales to clarify comparisons

Design considerations

- Use sorting and consistent scales to clarify comparisons
- Prefer small multiples over overloaded single panels

Design considerations

- Use sorting and consistent scales to clarify comparisons
- Prefer small multiples over overloaded single panels
- Highlight only a few key labels; avoid label clutter

Design considerations

- Use sorting and consistent scales to clarify comparisons
- Prefer small multiples over overloaded single panels
- Highlight only a few key labels; avoid label clutter
- Use log scales when data span orders of magnitude

Exploring correlations

- Use `GGally::ggpairs()` for scatterplot matrices

Exploring correlations

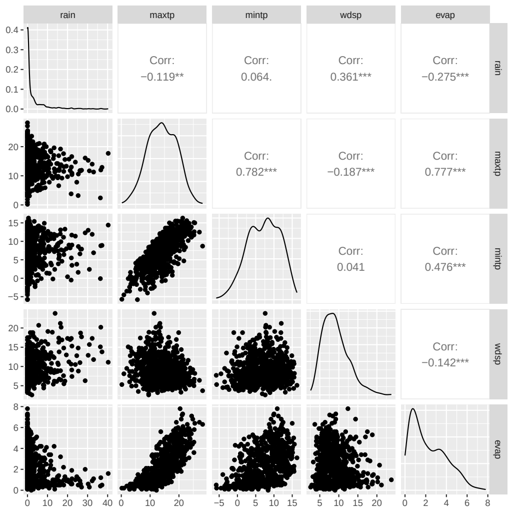
- Use `GGally::ggpairs()` for scatterplot matrices
- Visualize relationships among several variables at once

Exploring correlations

- Use `GGally::ggpairs()` for scatterplot matrices
- Visualize relationships among several variables at once
- Good for exploration; often too dense for publication

Scatterplot matrix with GGPAIRS

```r
# library(GGally)
weather_cor <- IRE_weather |> select(rain, maxtp, mintp, wdsp, evap)
```



Correlation patterns

- High vs low temperature: very strong positive correlation ($r \approx 0.78$)

Correlation patterns

- High vs low temperature: very strong positive correlation ($r \approx 0.78$)
- High temp vs evaporation: very strong positive correlation ($r \approx 0.77$)

Correlation patterns

- High vs low temperature: very strong positive correlation ($r \approx 0.78$)
- High temp vs evaporation: very strong positive correlation ($r \approx 0.77$)
- Wind speed vs rain: moderate positive correlation

Correlation patterns

- High vs low temperature: very strong positive correlation ($r \approx 0.78$)
- High temp vs evaporation: very strong positive correlation ($r \approx 0.77$)
- Wind speed vs rain: moderate positive correlation
- Wind speed vs temperature: minor negative correlation

Correlation patterns

- High vs low temperature: very strong positive correlation ($r \approx 0.78$)
- High temp vs evaporation: very strong positive correlation ($r \approx 0.77$)
- Wind speed vs rain: moderate positive correlation
- Wind speed vs temperature: minor negative correlation
- Little or no correlation for some variable pairs (e.g., wind speed vs min temp)

Correlograms

- Heatmaps of correlation coefficients

Correlograms

- Heatmaps of correlation coefficients
- More suitable for publication than full scatterplot matrices

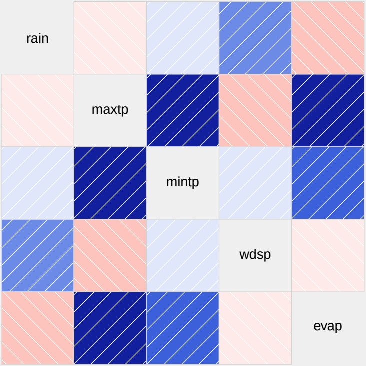
Correlograms

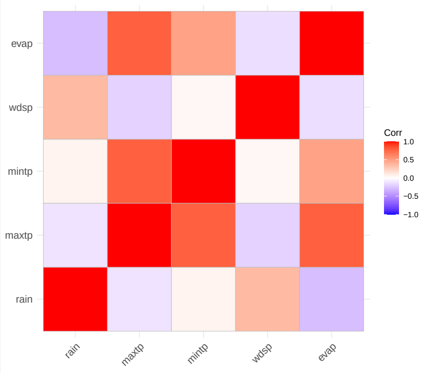
- Heatmaps of correlation coefficients
- More suitable for publication than full scatterplot matrices
- Depending on package, need to compute a correlation matrix and tidy it

Ex: Correlogram with weather

```r
# library(corrgram)
corrgram(weather_cor)
```

```r
# library(ggcorrplot)
ggcorrplot(cor(weather_cor))
```





Simple regression: idea

- Outcome: Daily high temperature

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

```
# run our "simple" regression w/ max temp as outcome
model_simple <- lm(maxtp ~ rain, data = IRE_weather)
# library(broom)
tidy(model_simple, conf.int = TRUE)
```

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

```r
# run our "simple" regression w/ max temp as outcome
model_simple <- lm(maxtp ~ rain, data = IRE_weather)
# library(broom)
tidy(model_simple, conf.int = TRUE)
```

```
# A tibble: 2 × 7
term        estimate std.error statistic p.value conf.low conf.high
<chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1 (Intercept) 14.5       0.204      70.9  0          14.1      14.9
2 rain       -0.0997    0.0307     -3.24 0.00124   -0.160   -0.0393
```

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

```r
# run our "simple" regression w/ max temp as outcome
model_simple <- lm(maxtp ~ rain, data = IRE_weather)
# library(broom)
tidy(model_simple, conf.int = TRUE)
```

```
# A tibble: 2 × 7
term        estimate std.error statistic p.value conf.low conf.high
<chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1 (Intercept) 14.5        0.204      70.9  0          14.1      14.9
2 rain        -0.0997     0.0307     -3.24 0.00124    -0.160    -0.0393
```

We can interpret these coefficients like so:

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

```r
# run our "simple" regression w/ max temp as outcome
model_simple <- lm(maxtp ~ rain, data = IRE_weather)
# library(broom)
tidy(model_simple, conf.int = TRUE)
```

```
# A tibble: 2 × 7
  term        estimate std.error statistic p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1 (Intercept)  14.5       0.204       70.9 0          14.1     14.9
2 rain         -0.0997    0.0307      -3.24 0.00124   -0.160   -0.0393
```

We can interpret these coefficients like so:

- Intercept:

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

```r
# run our "simple" regression w/ max temp as outcome
model_simple <- lm(maxtp ~ rain, data = IRE_weather)
# library(broom)
tidy(model_simple, conf.int = TRUE)
```

```
# A tibble: 2 × 7
  term        estimate std.error statistic p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1 (Intercept)  14.5       0.204      70.9  0          14.1     14.9
2 rain         -0.0997    0.0307     -3.24 0.00124   -0.160   -0.0393
```

We can interpret these coefficients like so:

- Intercept: Average temperature when there's no rain is 14.5°C

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

```r
# run our "simple" regression w/ max temp as outcome
model_simple <- lm(maxtp ~ rain, data = IRE_weather)
# library(broom)
tidy(model_simple, conf.int = TRUE)
```

```
# A tibble: 2 × 7
  term        estimate std.error statistic p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1 (Intercept)  14.5       0.204      70.9  0          14.1     14.9
2 rain         -0.0997    0.0307     -3.24 0.00124   -0.160   -0.0393
```

We can interpret these coefficients like so:

- Intercept: Average temperature when there's no rain is 14.5°C
- β_{rain}:

Simple regression: idea

- Outcome: Daily high temperature
- Predictor: Rain

```
# run our "simple" regression w/ max temp as outcome
model_simple <- lm(maxtp ~ rain, data = IRE_weather)
# library(broom)
tidy(model_simple, conf.int = TRUE)
```

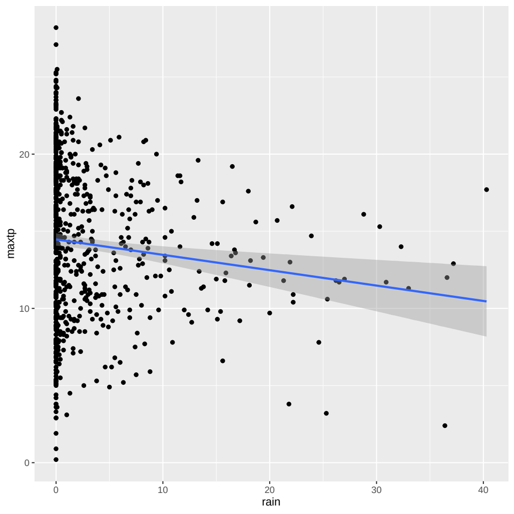
```
# A tibble: 2 × 7
term        estimate std.error statistic p.value conf.low conf.high
<chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1 (Intercept) 14.5       0.204      70.9 0          14.1     14.9
2 rain        -0.0997    0.0307     -3.24 0.00124   -0.160   -0.0393
```

We can interpret these coefficients like so:

- Intercept: Average temperature when there's no rain is 14.5°C
- β_{rain}:↑ 1mm of rain is associated with a 0.0997° decrease in max temp, on average

Visualizing simple regression

```r
ggplot(IRE_weather,
       aes(x = rain, y = maxtp)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Multiple regression

- Add predictors: humidity, moon phase, rain probability, wind speed, pressure, cloud cover

Multiple regression

- Add predictors: humidity, moon phase, rain probability, wind speed, pressure, cloud cover
- Hard to visualize directly in original space

Multiple regression

- Add predictors: humidity, moon phase, rain probability, wind speed, pressure, cloud cover
- Hard to visualize directly in original space
- Use coefficient plots and predicted values

Multiple regression

```r
# run our "complex" regression w/ max temp as outcome
model_complex <- lm(maxtp ~ rain + mintp + wdsp + evap + station,
                    data = IRE_weather)
tidy(model_complex, conf.int = TRUE)
```

Multiple regression

```r
# run our "complex" regression w/ max temp as outcome
model_complex <- lm(maxtp ~ rain + mintp + wdsp + evap + station,
                    data = IRE_weather)
tidy(model_complex, conf.int = TRUE)
```

```
# A tibble: 6 × 7
  term          estimate std.error statistic   p.value conf.low conf.high
  <chr>            <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)     7.70      0.267      28.8  3.38e-122   7.17      8.22
2 rain            0.0406    0.0136      2.98 3.00e-  3   0.0138    0.0674
3 mintp           0.625     0.0192     32.6  2.87e-144   0.587     0.663
4 wdsp           -0.216     0.0228     -9.46 4.16e- 20  -0.261    -0.171
5 evap            1.51      0.0521     28.9  8.80e-123   1.40      1.61
6 stationDublin   0.805     0.145       5.56 3.79e-  8   0.521     1.09
```

Multiple regression

```r
# run our "complex" regression w/ max temp as outcome
model_complex <- lm(maxtp ~ rain + mintp + wdsp + evap + station,
                    data = IRE_weather)
tidy(model_complex, conf.int = TRUE)
```

```
# A tibble: 6 × 7
  term          estimate std.error statistic   p.value conf.low conf.high
  <chr>            <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)     7.70      0.267      28.8  3.38e-122   7.17      8.22
2 rain            0.0406    0.0136      2.98 3.00e-  3   0.0138    0.0674
3 mintp           0.625     0.0192     32.6  2.87e-144   0.587     0.663
4 wdsp           -0.216     0.0228     -9.46 4.16e- 20  -0.261    -0.171
5 evap            1.51      0.0521     28.9  8.80e-123   1.40      1.61
6 stationDublin   0.805     0.145       5.56 3.79e-  8   0.521     1.09
```

- β_{rain}: On average, ↑ 1mm of rain is associated with a 0.0406°C increase in max temp, holding all else constant

Multiple regression

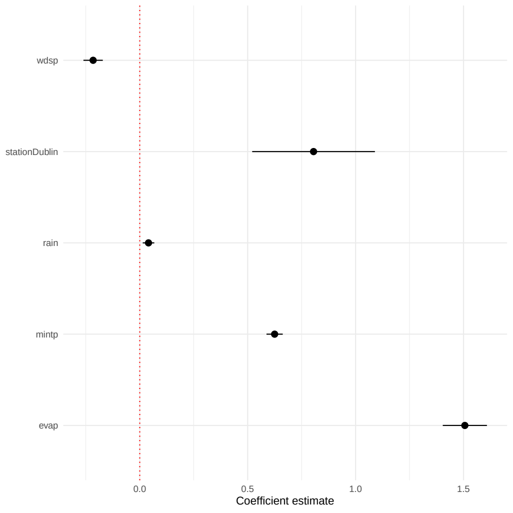
```r
# run our "complex" regression w/ max temp as outcome
model_complex <- lm(maxtp ~ rain + mintp + wdsp + evap + station,
                    data = IRE_weather)
tidy(model_complex, conf.int = TRUE)
```

```
# A tibble: 6 × 7
  term          estimate std.error statistic   p.value conf.low conf.high
  <chr>            <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)     7.70      0.267      28.8  3.38e-122   7.17      8.22
2 rain            0.0406    0.0136      2.98 3.00e-  3   0.0138    0.0674
3 mintp           0.625     0.0192     32.6  2.87e-144   0.587     0.663
4 wdsp           -0.216     0.0228     -9.46 4.16e- 20  -0.261    -0.171
5 evap            1.51      0.0521     28.9  8.80e-123   1.40      1.61
6 stationDublin   0.805     0.145       5.56 3.79e-  8   0.521     1.09
```

- β_{rain}: On average, $\uparrow$ 1mm of rain is associated with a 0.0406°C increase in max temp, holding all else constant
- β_{Dublin}: On average, Dublin has a 0.805°C higher max temp than Cork, holding all else constant

Coefficient plot

```r
ggplot(tidy(model_complex, conf.int = TRUE) |> filter(term != "(Intercept)"),
       aes(x = estimate, y = term)) +
  geom_vline(xintercept = 0, color = "red", linetype = "dotted") +
  geom_pointrange(aes(xmin = conf.low, xmax = conf.high)) +
  labs(x = "Coefficient estimate", y = NULL) +
  theme_minimal()
```



Using marginaleffects

- `predictions()` for predicted values with CIs

Using marginaleffects

- `predictions()` for predicted values with CIs
- `slopes()` for marginal effects (slopes)

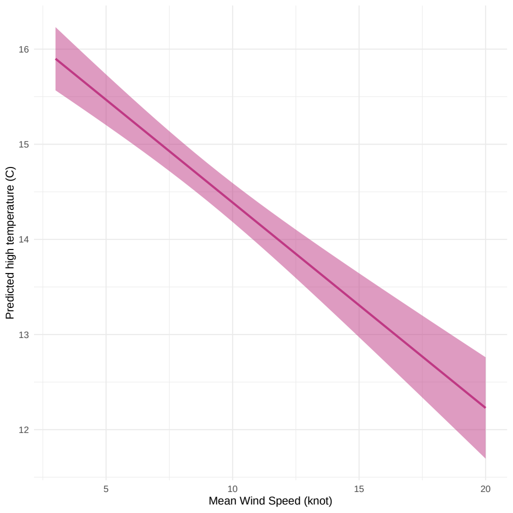
Using marginaleffects

- `predictions()` for predicted values with CIs
- `slopes()` for marginal effects (slopes)
- `datagrid()` constructs scenarios, holding other variables at typical values

MARGINALEFFECTS - WIND

```r
# library(marginaleffects)
# Calculate predictions across a range of windSpeed
predicted_values_easy <- predictions(model_complex,
  newdata = datagrid(wdsp = seq(3, 20, 0.5))
)
```

```r
ggplot(predicted_values_easy, aes(x = wdsp, y = estimate)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high),
              fill = "#BF3984", alpha = 0.5) +
  geom_line(linewidth = 1, color = "#BF3984") +
  labs(x = "Mean Wind Speed (knot)", y = "Predicted high temperature (C)") +
```



Nonlinear and interaction model

You could:

- Add quadratic term for wind speed

Nonlinear and interaction model

You could:

- Add quadratic term for wind speed
- Add interaction: wind speed $\times$ station

Nonlinear and interaction model

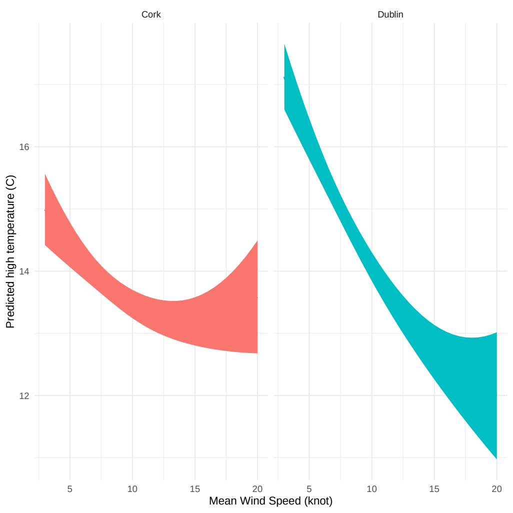
You could:

- Add quadratic term for wind speed
- Add interaction: wind speed $\times$ station

```r
model_wild <- lm(maxtp ~ rain + mintp + wdsp + evap + station + I(wdsp^2) + wdsp:station,
      data = IRE_weather)

predicted_values_wild <- predictions(
  model_wild,
  newdata = datagrid(
    wdsp = seq(3, 20, 0.5),
    station = c("Cork", "Dublin")))
```

MARGINALEFFECTS - INTERACTIONS




```r
ggplot(predicted_values_wild, aes(x = wdsp, y = estimate)) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high, fill = station)) +
  geom_line(aes(color = station), linewidth = 1) +
  labs(x = "Mean Wind Speed (knot)", y = "Predicted high temperature (C)") +
  theme_minimal() +
  guides(fill = "none", color = "none") +
  facet_wrap(vars(station), nrow = 1)
```

Wrap up

- Dual y-axes are safe when measuring the same quantity in different units

Wrap up

- Dual y-axes are safe when measuring the same quantity in different units
- `patchwork` simplifies multi-plot layouts

Wrap up

- Dual y-axes are safe when measuring the same quantity in different units
- `patchwork` simplifies multi-plot layouts
- Correlograms give compact views of many correlations

Wrap up

- Dual y-axes are safe when measuring the same quantity in different units
- `patchwork` simplifies multi-plot layouts
- Correlograms give compact views of many correlations
- `broom` + `marginaleffects` streamline predictions and marginal effects

Class business

- Read required (and suggested) online materials

Class business

- Read required (and suggested) online materials
- Problem set #3 is up on GitHub