DATA2002

Regression

Garth Tarr

THE UNIVERSITY OF SYDNEY

Learning and prediction

Simple linear regression

Inference

In-sample performance

Module 4: learning and prediction

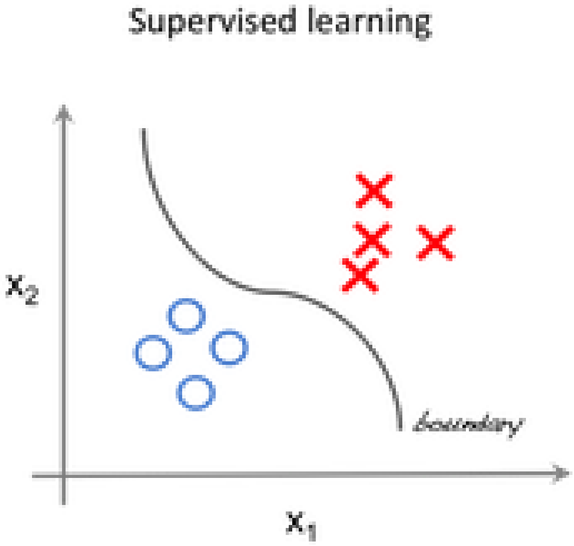
Types of learning

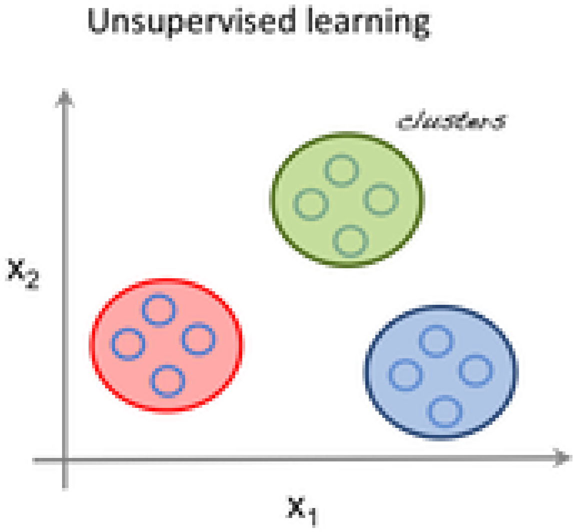
Supervised learning

- We have knowledge of class labels or values.
- Goal: train a model using known class labels to predict class or value label for a new data point.

Unsupervised learning

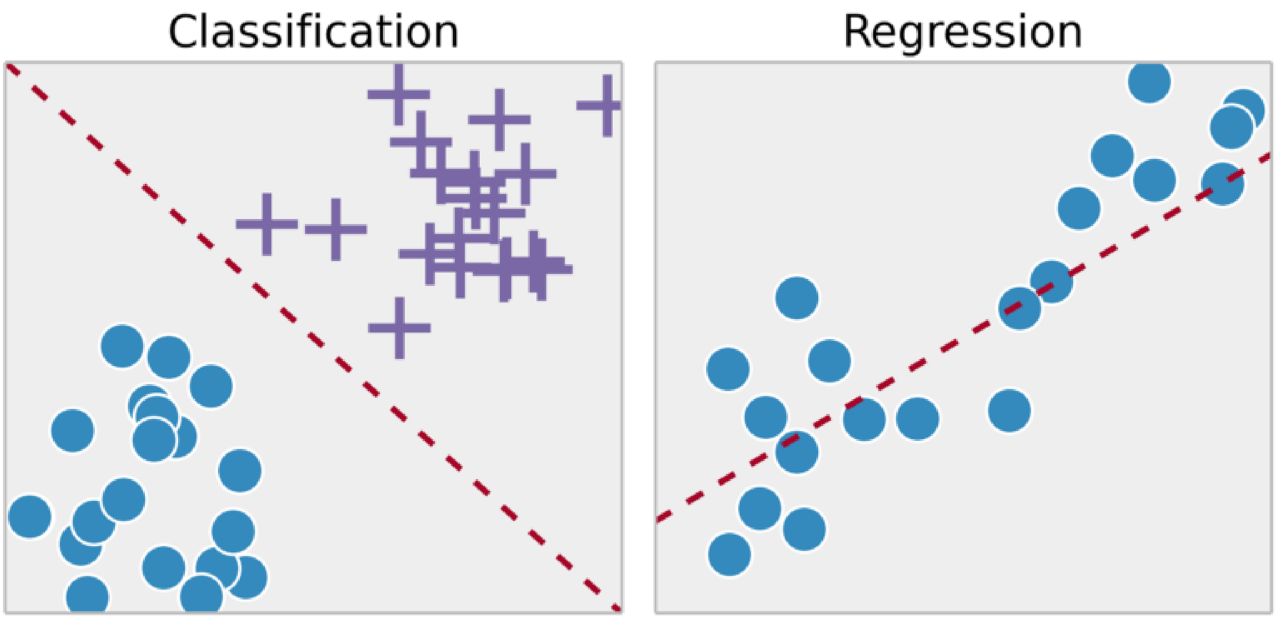
- No knowledge of output class or value – data is unlabelled.
- Goal: determine data patterns/groupings.





Supervised learning

Supervised learning can be further broken down into two main classes: **classification** and **regression**.



Classification maps inputs to an output label (e.g. decision trees, nearest neighbour, logistic regression, naive bayes, support vector machines, artificial neural networks, and random forests)

Regression maps inputs to a continuous output

Regression

Air pollution

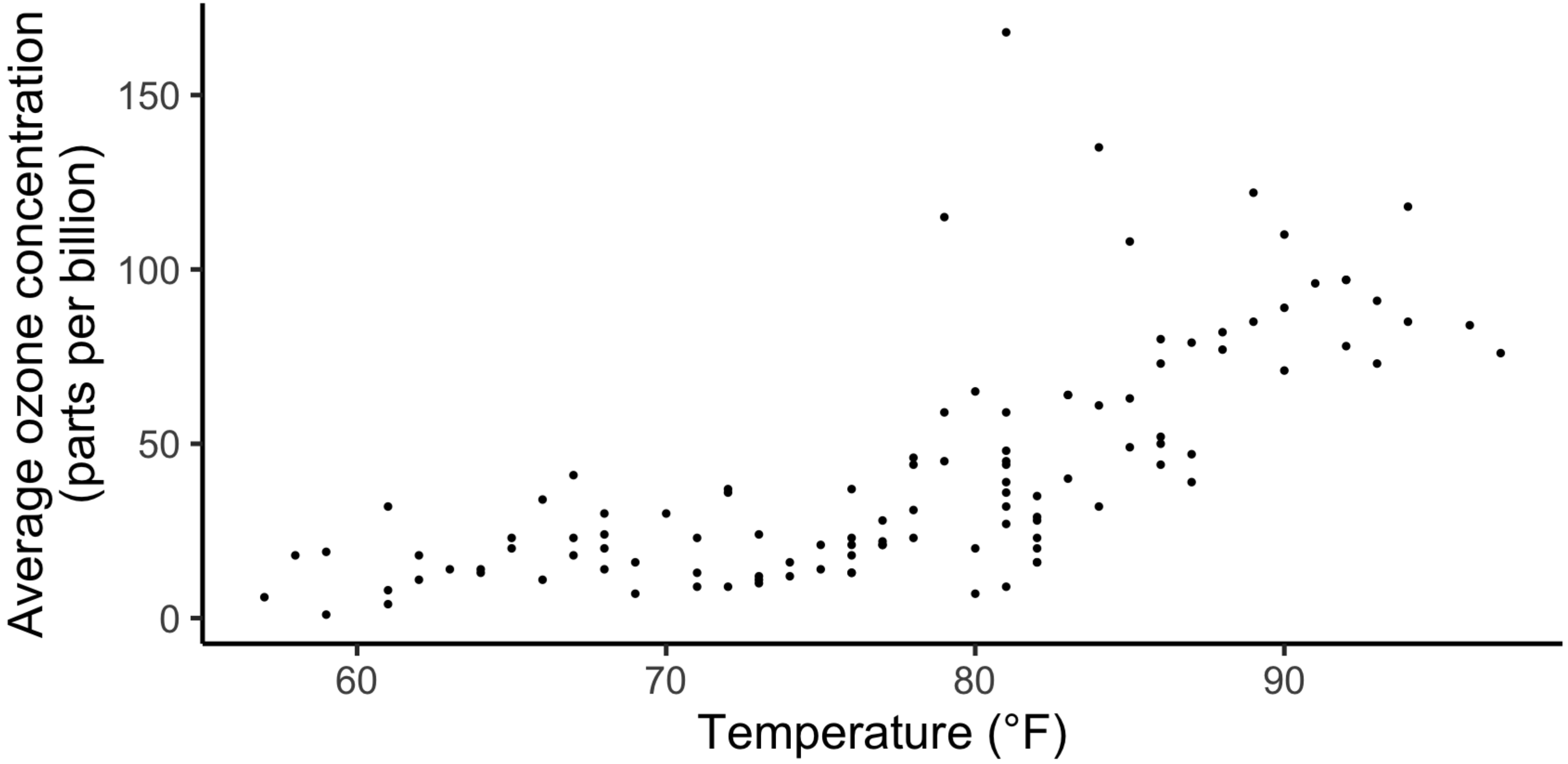
The data frame **environmental** has four environmental variables `ozone`, `radiation`, `temperature` and `wind` taken in New York City from May to September of 1973.

```
library(tidyverse)
data(environmental, package = "lattice")
# ?environmental
glimpse(environmental)
```

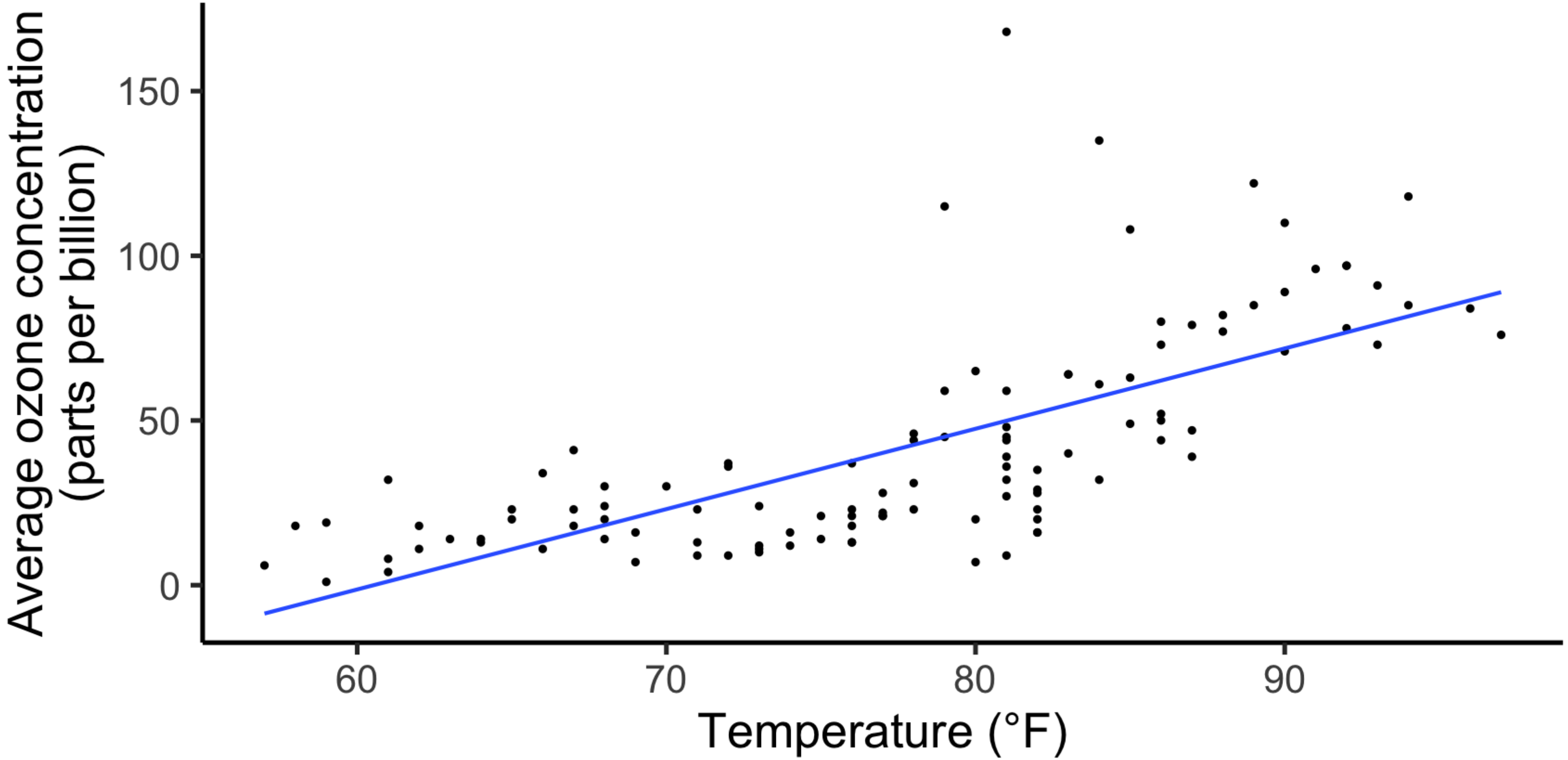
```
## Rows: 111
## Columns: 4
## $ ozone       <dbl> 41, 36, 12, 18, 23, 19, 8, 16, 11, 14,…
## $ radiation   <dbl> 190, 118, 149, 313, 299, 99, 19, 256, …
## $ temperature <dbl> 67, 72, 74, 62, 65, 59, 61, 69, 66, 68…
## $ wind        <dbl> 7.4, 8.0, 12.6, 11.5, 8.6, 13.8, 20.1,…
```

We'd like to assess whether the maximum daily temperature has an influence on average ozone concentration.

```r
ggplot(environmental, aes(x = temperature, y = ozone)) +
  geom_point() + theme_classic(base_size = 26) +
  labs(x = "Temperature (°F)", y = "Average ozone concentration\n(parts per billion)")
```



```r
ggplot(environmental, aes(x = temperature, y = ozone)) +
  geom_point() + theme_classic(base_size = 26) +
  labs(x = "Temperature (°F)", y = "Average ozone concentration\n(parts per billion)") +
  geom_smooth(method = "lm", se = FALSE)
```



Simple linear regression

A **simple linear regression** model aims to predict an outcome variable, Y, using a single predictor variable x,

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

for $i = 1, 2, \ldots, n$ where n is the number of observations (rows) in the data set.

This is just the equation of a straight line (like $y = mx + b$) plus some additional variation,

- β_0 is the population intercept parameter
- β_1 is the population slope parameter
- ε_i is the error term and typically assumed to follow $N(0, \sigma^2)$

Hence,

$$Y_i \sim N(\beta_0 + \beta_1 x_i,\ \sigma^2).$$

Fitting a straight line by least squares

How to estimate β_0 and β_1? We aim to **minimise the sum of squared residuals**.

- What's a residual?

$$r_i = y_i - \hat{y}_i$$

  where $\hat{y}_i$ is the fitted value, the value we predict for the ith observation given the ith predictor value:

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

- The estimated intercept ($\hat{\beta}_0$) and estimated slope ($\hat{\beta}_1$) are found by solving the following optimisation problem:

$$\mathbf{argmin}_{\beta_0,\beta_1} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2.$$

- Closed form solutions exist for $\hat{\beta}_0$ and $\hat{\beta}_1$.
- R does this for us with the `lm()` function (short for linear model).

```r
lm1 = lm(ozone ~ temperature,
         data = environmental)
lm1
```

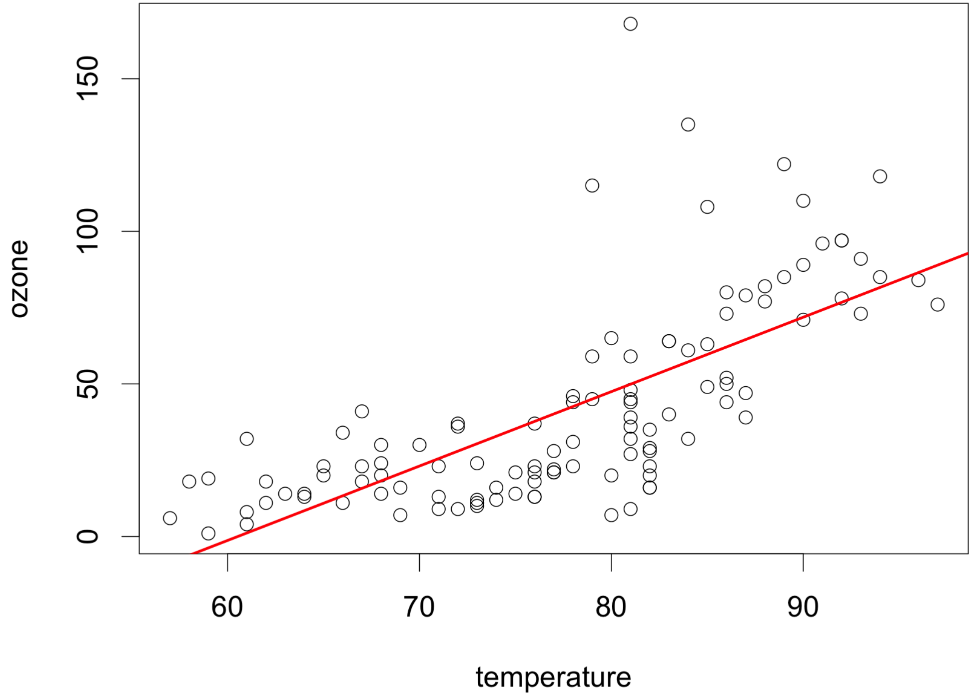
```
##
## Call:
## lm(formula = ozone ~ temperature, data = environmer
##
## Coefficients:
## (Intercept)  temperature
##    -147.646        2.439
```

Our estimated model is:

$$\widehat{\text{ozone}} = -147.646 + 2.439 \times \text{temperature}$$

Using base graphics:

```r
par(cex = 2)
plot(ozone~temperature, data = environmental)
abline(lm1, lwd = 3, col = "red")
```



Fitted values and residuals

The fitted values ($\hat{y}$) are obtained by plugging the observed predictor (x) values into our estimated model, $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$.

```
environmental = environmental %>%
  mutate(
    fitted = -147.646 + 2.439 * temperature
  )
```

The residuals are the differences between the observed outcome variable (y) and the value the estimated model predicts for that observation (the fitted value, $\hat{y}$),

$$r_i = y_i - \hat{y}_i \ .$$

```
environmental = environmental %>%
  mutate(resid = ozone - fitted)
```

An easier alternative is to extract the residuals and fitted values from the `lm1` object directly:

```
environmental = environmental %>%
  mutate(
    resid = lm1$residuals,
    fitted = lm1$fitted.values
  )
```

Alternatively we could have used the `augment()` function from the **broom** package to do this:

```
broom::augment(lm1) %>% glimpse()
```

```
## Rows: 111
## Columns: 8
## $ ozone       <dbl> 41, 36, 12, 18, 23, 19, 8, 16
## $ temperature <dbl> 67, 72, 74, 62, 65, 59, 61, 6
## $ .fitted     <dbl> 15.774291, 27.969841, 32.8480
## $ .resid      <dbl> 25.225709, 8.030159, -20.8480
## $ .hat        <dbl> 0.02066883, 0.01236793, 0.010
```

The `lm` object

What other hidden treasures does the `lm1` object hold?

```
names(lm1)
```

```
##  [1] "coefficients"  "residuals"
##  [3] "effects"       "rank"
##  [5] "fitted.values" "assign"
##  [7] "qr"            "df.residual"
##  [9] "xlevels"       "call"
## [11] "terms"         "model"
```

E.g. we can extract the coefficients:

```
lm1$coefficients
```

```
## (Intercept) temperature
##  -147.64607     2.43911
```

Or we can use the `tidy()` function from the **broom** package:

```
lm1 %>% broom::tidy()
```

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic
##   <chr>          <dbl>     <dbl>     <dbl>
## 1 (Intercept)  -148.      18.8       -7.87
## 2 temperature     2.44     0.239     10.2
## # … with 1 more variable: p.value <dbl>
```

Linear regression assumptions

There are 4 assumptions underling our linear regression model:

1. **Linearity** - the relationship between Y and x is linear
2. **Independence** - all the errors are independent of each other
3. **Homoskedasticity** - the errors have constant variance $\mathrm{Var}(\varepsilon_i) = \sigma^2$ for all $i = 1, 2, \ldots, n$
4. **Normality** - the errors follow a normal distribution

The last three can be written succinctly as $\varepsilon_i \sim$ iid $N(0, \sigma^2)$.

Assumption 1: linearity

- Violations to the linearity assumption are very serious, it means your predictions are likely to be **systematically wrong**

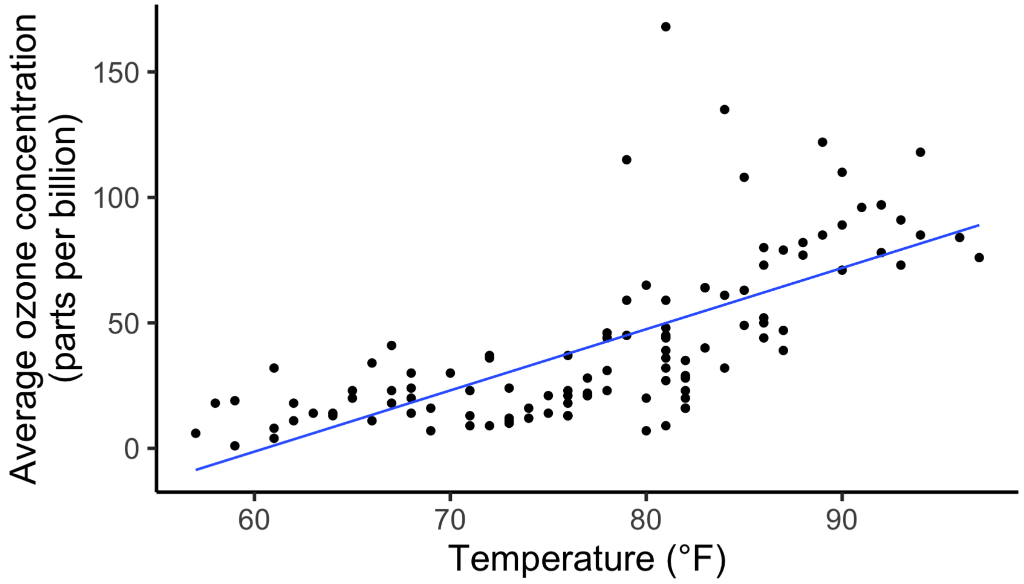
Checking for linearity

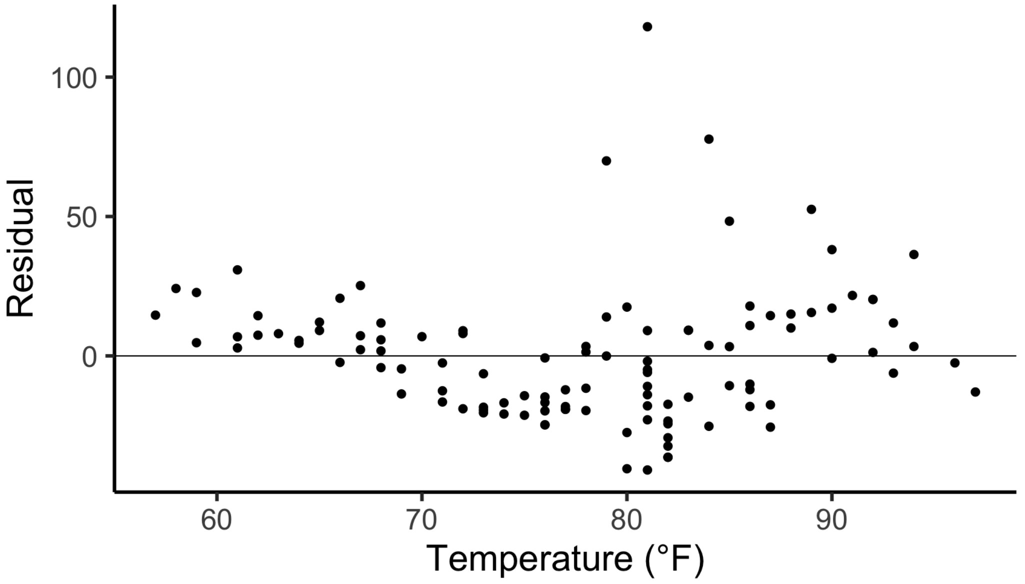
1. Before running the regression: plot y against x and look to see if the relationship is approximately linear
2. After running the regression: look at a plot of the residuals against x
   - Residuals should be symmetrically distributed above and below zero
   - A curved pattern in the residuals is evidence for non-linearity, i.e. for some values of x the model regularly overestimates y while in other regions the model regularly underestimates y

Assumption 1: linearity

```r
p1 = environmental %>% ggplot() +
  aes(x = temperature, y = ozone) +
  geom_point(size = 3) +
  theme_classic(base_size = 30) +
  labs(x = "Temperature (°F)",
       y = "Average ozone concentration\n(parts
  geom_smooth(method = "lm", se = FALSE)
p1
```

```r
p2 = environmental %>% ggplot() +
  aes(x = temperature, y = resid) +
  geom_point(size = 3) +
  theme_classic(base_size = 30) +
  labs(x = "Temperature (°F)",
       y = "Residual") +
  geom_hline(yintercept = 0)
p2
```

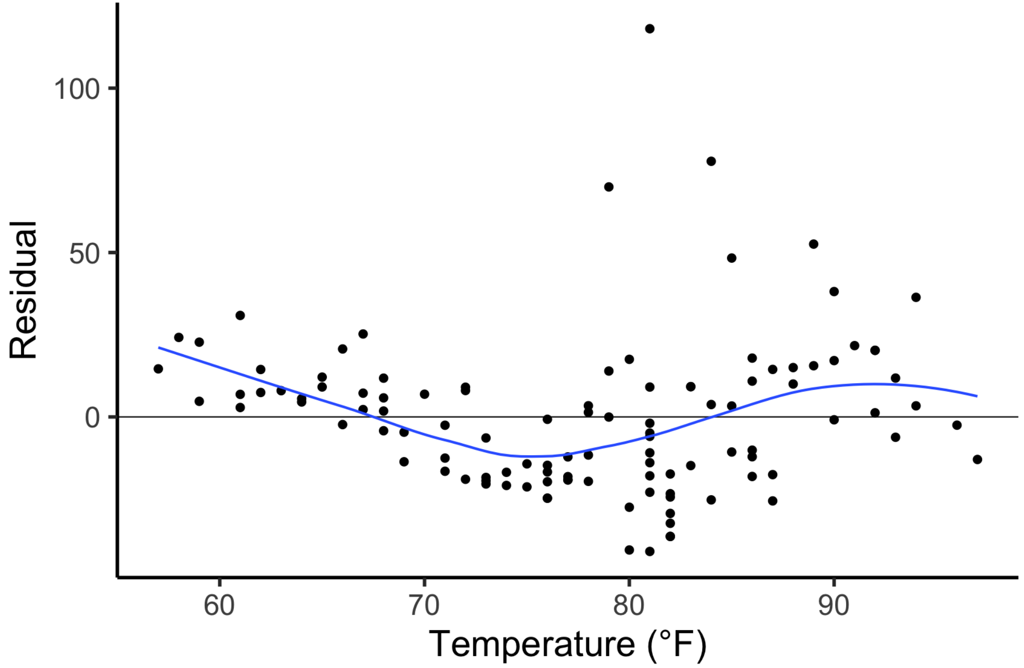




Assumption 1: linearity

In the plot below the residuals are above zero for low temperatures, then they go below zero and end up again above zero for high temperatures (as highlighted by the local smoothing curve).

```
p2 + geom_smooth(method = "loess", se = FALSE)
```



This means that we **underestimate** the ozone level for low and high temperatures and **overestimate** the ozone level at moderate temperatures.

Our predictions are **systematically wrong** for certain ranges of temperature.

If the linearity assumption fails, there's not much point checking the other assumptions because it's not an appropriate prediction model.

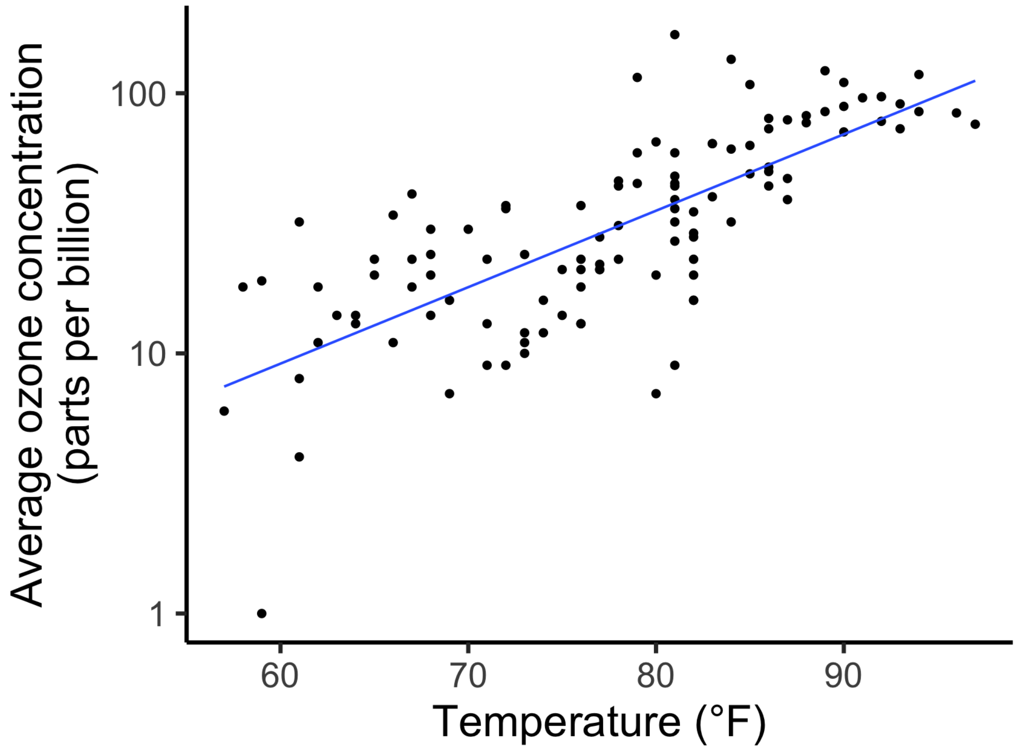
Transformations

If we see a non-linear relationship between y and x we might be able to transform the data so that we have a linear relationship between the transformed variable(s).

What if we considered the log of ozone concentration?

```
p1 = ggplot(environmental,
            aes(x = temperature,
                y = ozone)) +
  geom_point(size = 3) +
  scale_y_log10() +
  theme_classic(base_size = 36) +
  labs(x = "Temperature (°F)",
       y = "Average ozone concentration\n(parts
  geom_smooth(method = "lm", se = FALSE)
```

```
p1
```



```r
environmental = environmental %>%
  mutate(lozone = log(ozone))
lm2 = lm(lozone ~ temperature, data = environmental)
lm2
```

```
##
## Call:
## lm(formula = lozone ~ temperature, data = environmental)
##
## Coefficients:
## (Intercept)  temperature
##    -1.84852      0.06767
```

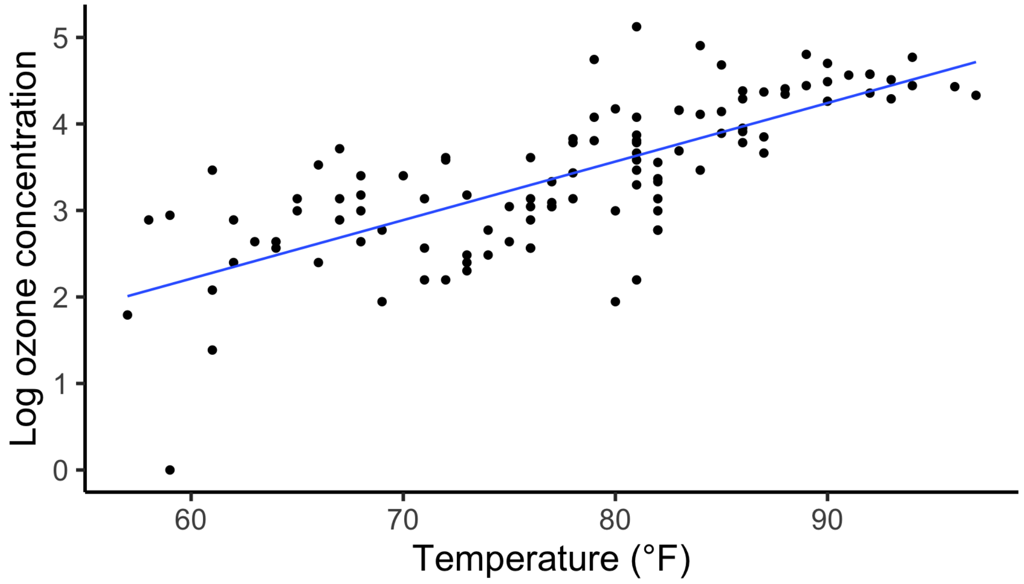
Now the fitted model is:

$$\log(\widehat{\text{ozone}}) = -1.84852 + 0.06767 \times \text{temperature}$$

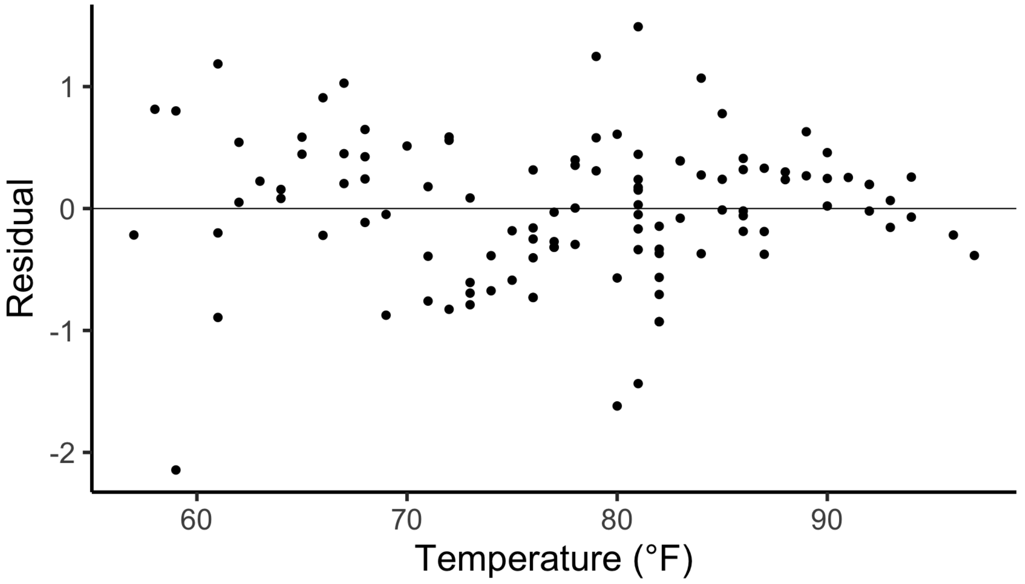
```r
environmental = environmental %>%
  mutate(
    lfitted = lm2$fitted.values,
    lresid = lm2$residuals
  )
```

Assumption 1: linearity

```
p1 = ggplot(environmental, aes(x = temperature,
                               y = lozone)) +
  geom_point(size = 3) +
  theme_classic(base_size = 30) +
  labs(x = "Temperature (°F)",
       y = "Log ozone concentration") +
  geom_smooth(method = "lm", se = FALSE)
p1
```




```
p2 = ggplot(environmental, aes(x = temperature,
                               y = lresid)) +
  geom_point(size = 3) +
  theme_classic(base_size = 30) +
  labs(x = "Temperature (°F)",
       y = "Residual") +
  geom_hline(yintercept = 0)
p2
```



Assumption 2: independence

The assumption of independence between the errors is usually dealt with in the experimental design phase - before data collection.

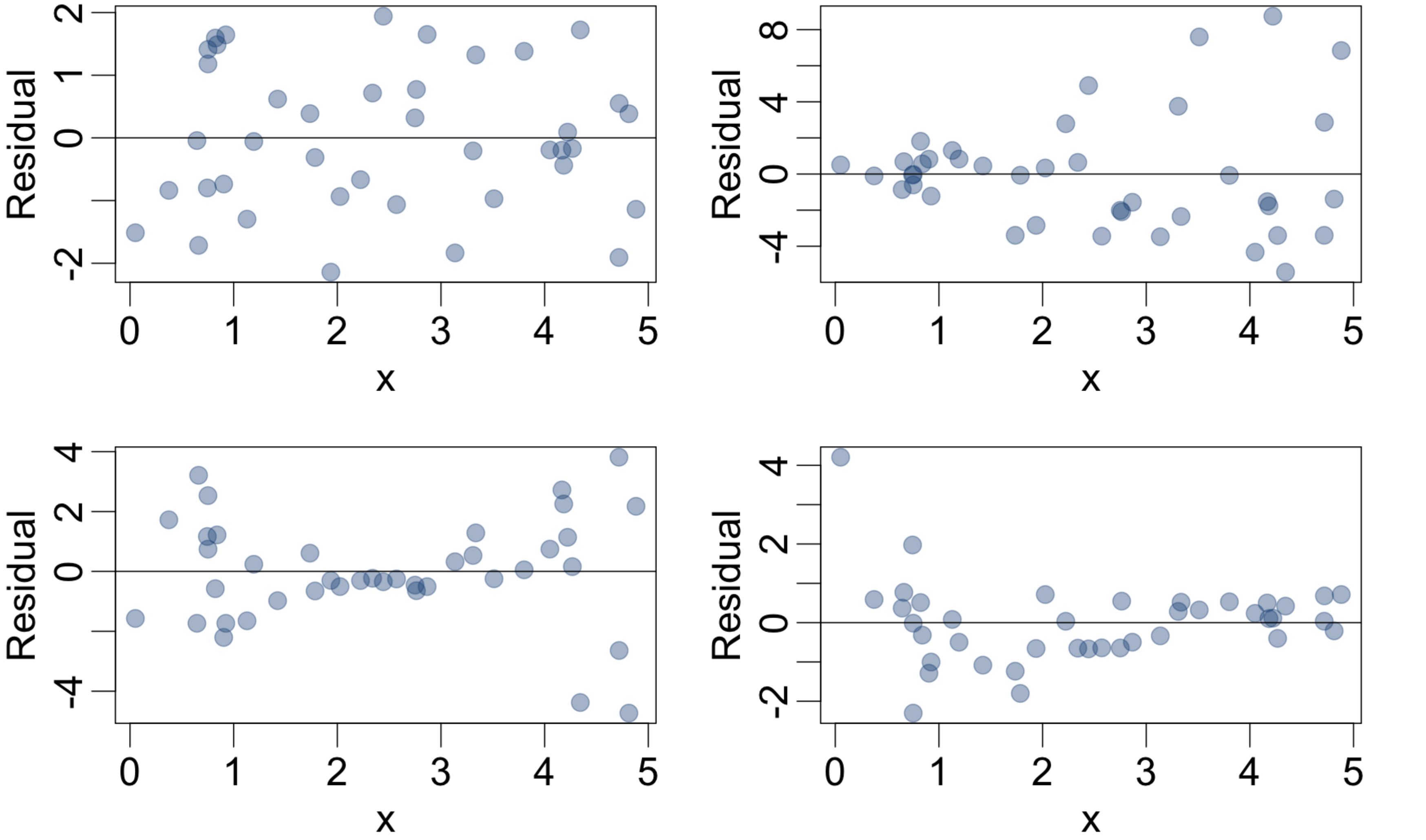
- You aim to design the experiment so that the observations are not related to one another.
- If you don't have a random sample, your estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ may be biased.
- Violations of independence often arise in time series data where observations are measured on the same subject through time and therefore may be related to one another. This is beyond the scope of DATA2002.

In the environmental data, there may be dependence that we haven't accounted as it is a time series data set (though we don't know which days they were taken on and if the records were sequential).

Assumption 3: homoskedasticity

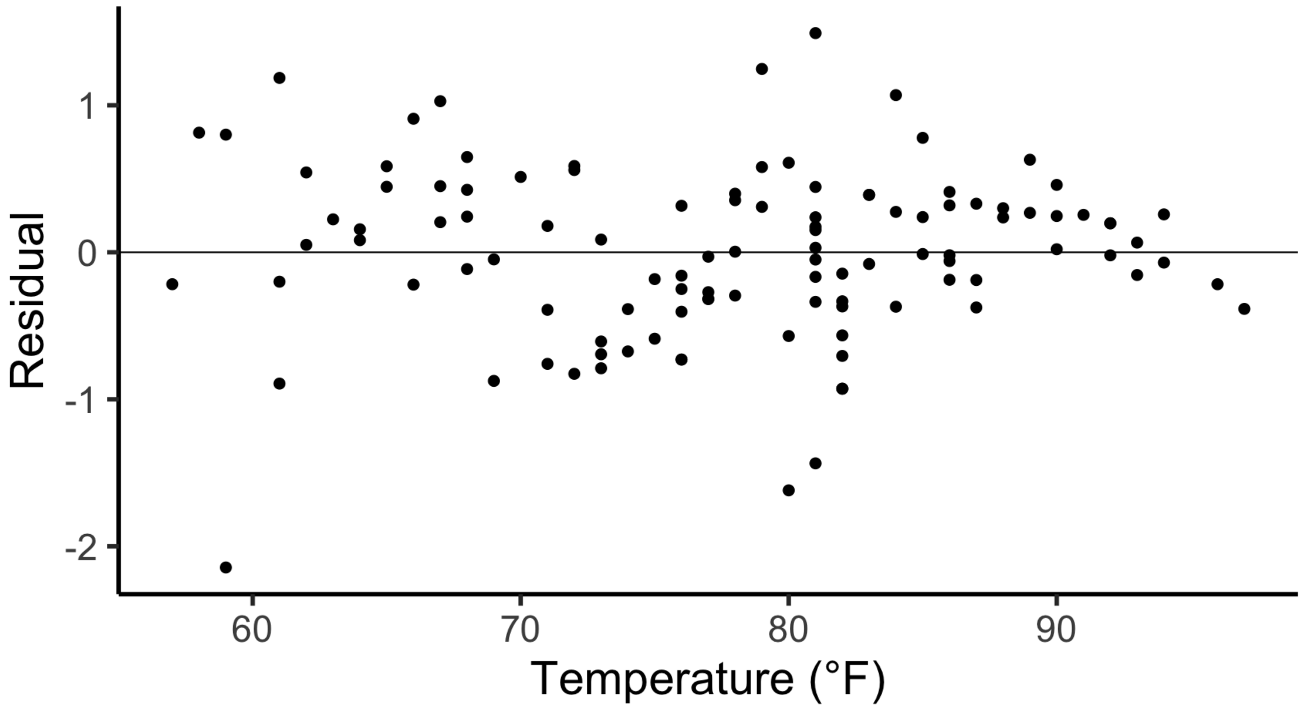
- Homoskedasticity (homo: same, skedasticity: spread)
- Constant error variance is important to ensure the hypothesis tests to give valid results.
- Violations of homoskedasticity, called **heteroskedasticity**, make it difficult to estimate the "true" standard deviation of the errors, resulting in confidence intervals that are too wide or too narrow.
- Heteroskedasticity may also have the effect of giving too much weight to small subset of the data (namely the subset where the error variance was largest) when estimating coefficients.
- You can check for homoskedasticity in plots of residuals versus x. If it appears the residuals are getting more spread-out, that is evidence of heteroskedasticity

Assumption 3: checking for homoskedasticity



Assumption 3: homoskedasticity

```
p2
```



The spread looks reasonably constant over the range of temperature values.

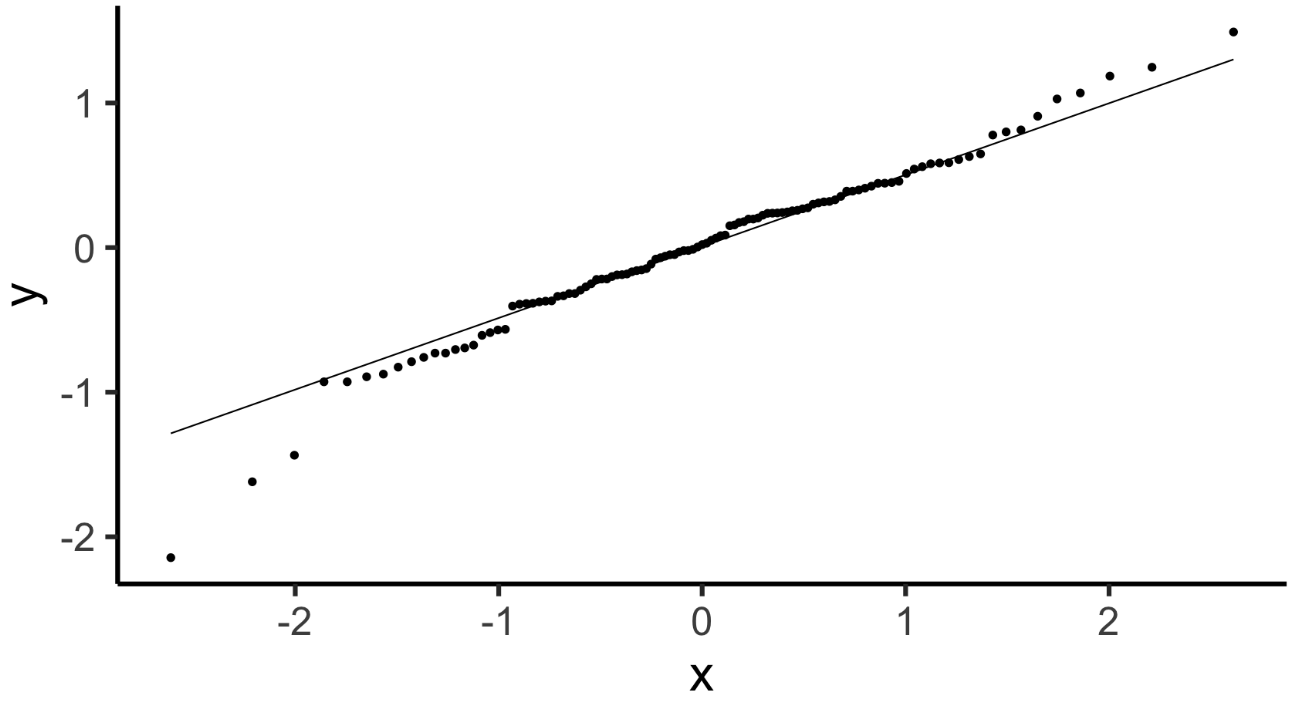
In the region above 85°F, the spread might be somewhat smaller than the spread in the region below 85°F but it's nothing to get too worried about.

Assumption 4: normality

- Violations of normality of the errors can compromise our inferences. The calculation of confidence intervals may be too wide or narrow and our conclusions from our hypothesis tests may be incorrect.
- The best way to check (visually) for normality is a QQ plot.
- In some cases, the problem may be due to one or two outliers. Such values should be scrutinised closely: are they genuine, are they explainable, are similar events likely to occur again in the future.
- Sometimes the extreme values in the data provide the most useful information.

Assumption 4: normality

```
environmental %>% ggplot() +
  aes(sample = lresid) +
  geom_qq(size = 2) + geom_qq_line()
```



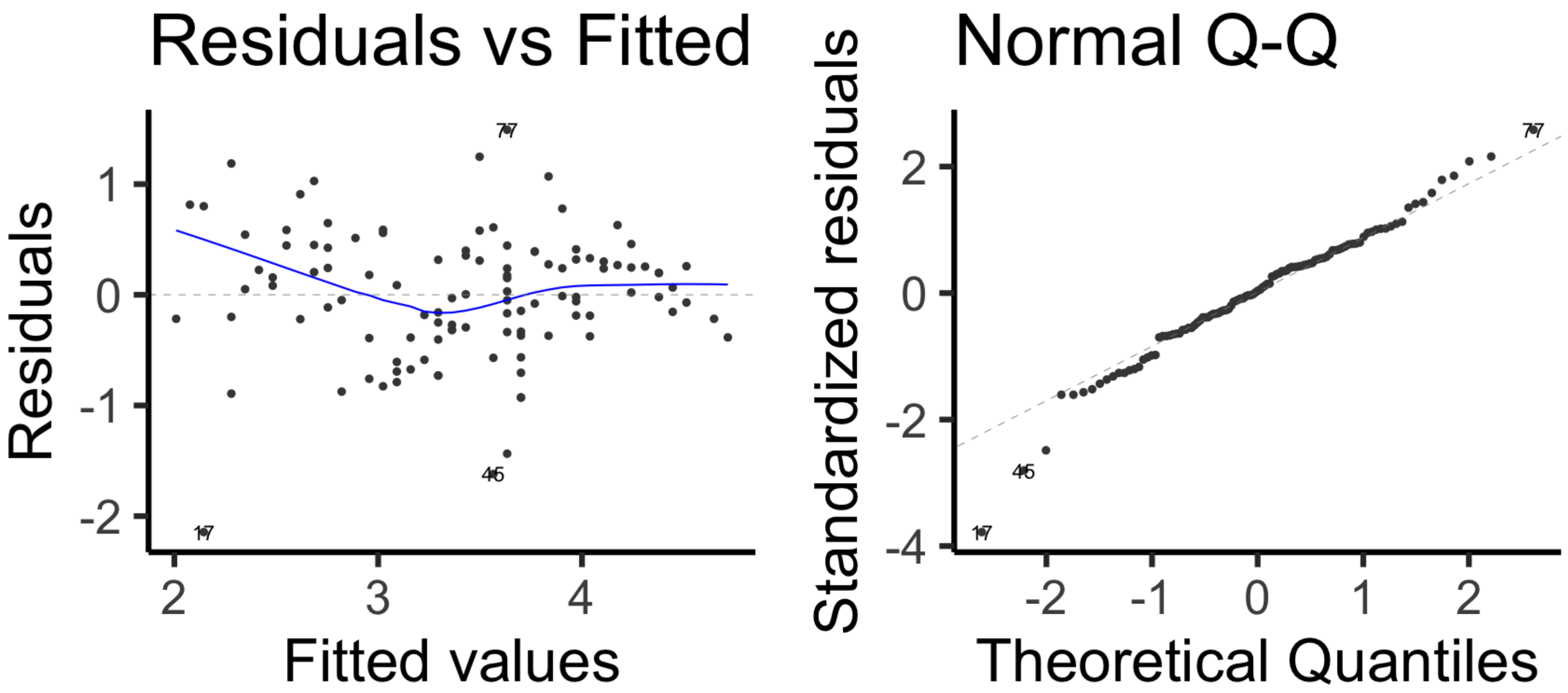
Apart from three points in the lower tail, the majority of the points lie quite close to the diagonal line in the QQ plot. Hence, the normality assumption for the residuals is reasonably well satisfied.

Additionally, we have quite a large sample size so we can also rely on the central limit theorem to give us approximately valid inferences.

Autoplot

The **ggfortify** package provides an `autoplot()` method for `lm` objects.

```
library(ggfortify)
autoplot(lm2, which = 1:2)
```



Inference in regression models

Inference

Recall our simple linear regression population model:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i.$$

Typically, we are interested in hypotheses of the form, $H_0\colon\ \beta_1 = 0$ vs $H_1\colon\ \beta_1 \neq 0$ or $\beta_1 > 0$ or $\beta_1 < 0$

To do this we use a t-test:

$$T = \frac{\hat{\beta}_1 - \beta_1}{\mathrm{SE}(\hat{\beta}_1)} \sim t_{n-2}$$

where $\hat{\beta}_1$ and $\mathrm{SE}(\hat{\beta}_1)$ are given in the R output.

Inference

```
summary(lm2)
```

```
## 
## Call:
## lm(formula = lozone ~ temperature, data = environmental)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.14417 -0.32555  0.02066  0.34234  1.49100 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.848518   0.455080  -4.062  9.2e-05 ***
## temperature  0.067673   0.005807  11.654  < 2e-16 ***
## ---
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.5804 on 109 degrees of freedom
## Multiple R-squared:  0.5548,    Adjusted R-squared:  0.5507 
## F-statistic: 135.8 on 1 and 109 DF,  p-value: < 2.2e-16
```

Nicer model output

```r
sjPlot::tab_model(lm2, show.ci = FALSE)
```

	lozone	
Predictors	*Estimates*	*p*
(Intercept)	-1.85	**<0.001**
temperature	0.07	**<0.001**
Observations	111	
R^2 / R^2 adjusted	0.555 / 0.551	

```r
# install.packages("equatiomatic")
library(equatiomatic)
extract_eq(lm2)
```

$$\operatorname{lozone} = \alpha + \beta_{1}(\operatorname{temperature}) + \epsilon$$

```r
extract_eq(lm2, use_coefs = TRUE)
```

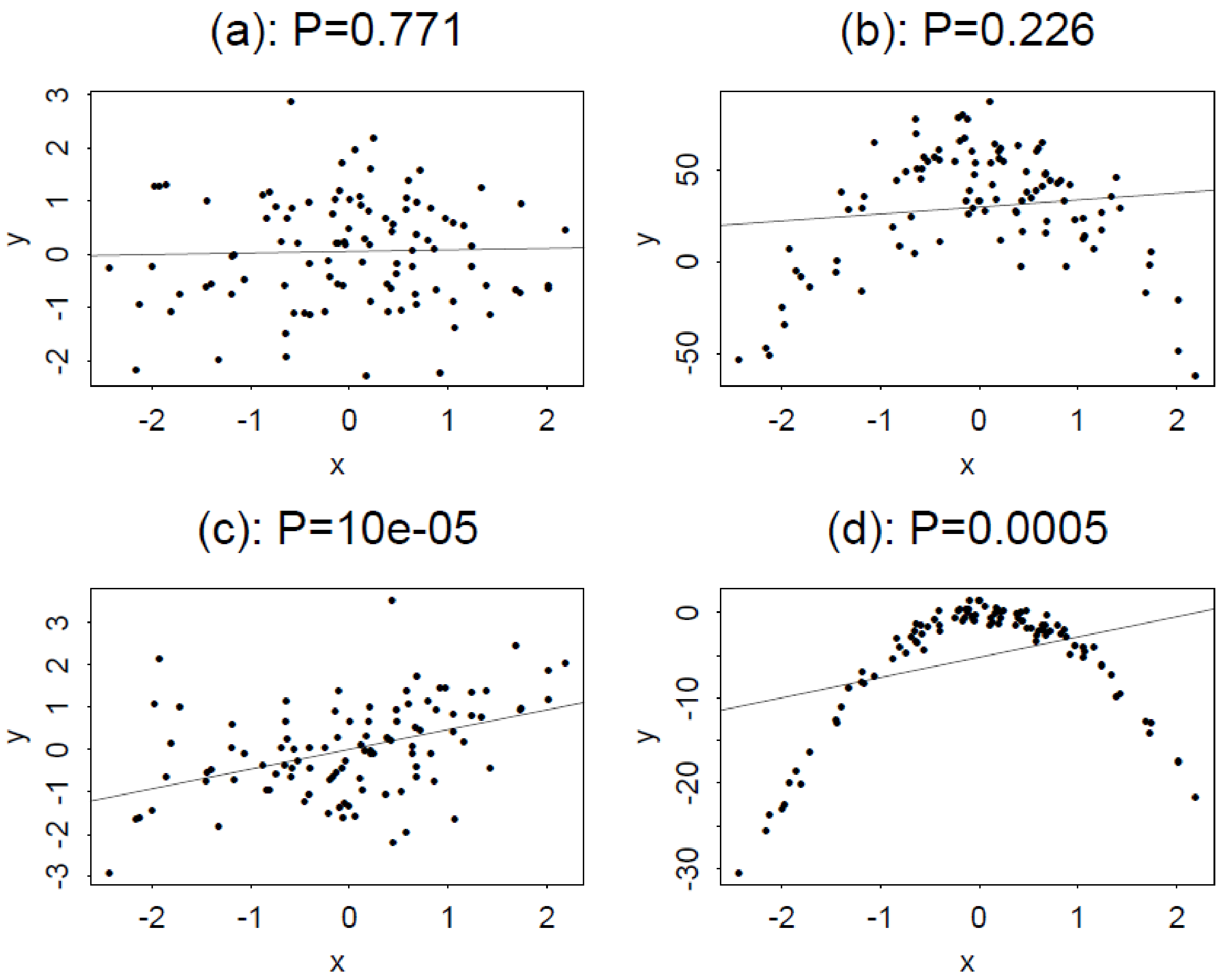
$$\operatorname{\widehat{lozone}} = -1.85 + 0.07(\operatorname{temperature})$$

Testing for the significance of the slope parameter β_1

The workflow to test the significance of β_1 (i.e. $\beta_1 = 0$) and hence the regression model are

- **Hypothesis:** H_0: $\beta_1 = 0$ vs H_1: $\beta_1 > 0$, $\beta_1 < 0$, $\beta_1 \neq 0$
- **Assumptions:** The residuals ε_i are iid $N(0, \sigma^2)$ and there is a linear relationship between y and x.
- **Test statistic:** $T = \dfrac{\hat{\beta}_1}{\mathrm{SE}(\hat{\beta}_1)} \sim t_{n-2}$ under H_0.
- **Observed test statistic:** t_0 (from R)
- **p-value:** $P(t_{n-2} \geq t_0)$ for H_1: $\beta_1 > 0$,
- $P(t_{n-2} \leq t_0)$ for H_1: $\beta_1 < 0$;
- $2P(t_{n-2} \geq |t_0|)$ for H_1: $\beta_1 \neq 0$.
- **Conclusion:** Reject H_0 if the p-value is less than the level of significance, α.

p-values mean nothing if you haven't looked at your data!



Recall our population model: $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$.

- **Hypothesis:** $H_0\colon\ \beta_1 = 0$ vs $H_1\colon\ \beta_1 \neq 0$
- **Assumptions:** The residuals ε_i are iid $N(0, \sigma^2)$ and there is a linear relationship between y and x (checked previously).
- **Test statistic:** $T = \dfrac{\hat{\beta}_1}{\mathrm{SE}(\hat{\beta}_1)} \sim t_{n-2}$ under H_0.
- **Observed test statistic:** $t_0 = \dfrac{0.0677}{0.00581} = 11.65$
- **P-value:** $2P(t_{109} \geq 11.95) < 0.0001$
- **Decision:** There is very strong evidence in the data to indicate a linear relationship between temperature and the logarithm of ozone concentration.

```
lm2 %>% broom::tidy()
```

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic  p
##   <chr>          <dbl>     <dbl>     <dbl>
## 1 (Intercept)  -1.85     0.455       -4.06 9.
## 2 temperature   0.0677   0.00581     11.7  7.
```

CI for regression coefficients

$100(1-\alpha)\%$ confidence intervals can be constructed for regression coefficients in the usual way:

$$\hat{\beta}_1 \pm t^\star \times \text{SE}(\hat{\beta}_1)$$

where $t^\star$ is the $\alpha/2$ quantile from a t distribution with $n-2$ degrees of freedom.

```r
# summary(lm2)$coefficients %>% round(4)
lm2 %>% broom::tidy() %>%
  knitr::kable(digits = 4)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-1.8485	0.4551	-4.0620	1e-04
temperature	0.0677	0.0058	11.6539	0e+00

```r
qt(0.025, df = 109) %>% round(3)
```

```
## [1] -1.982
```

Plugging in these values

$$0.0677 \pm 1.982 \times 0.0058 = (0.056, 0.079)$$

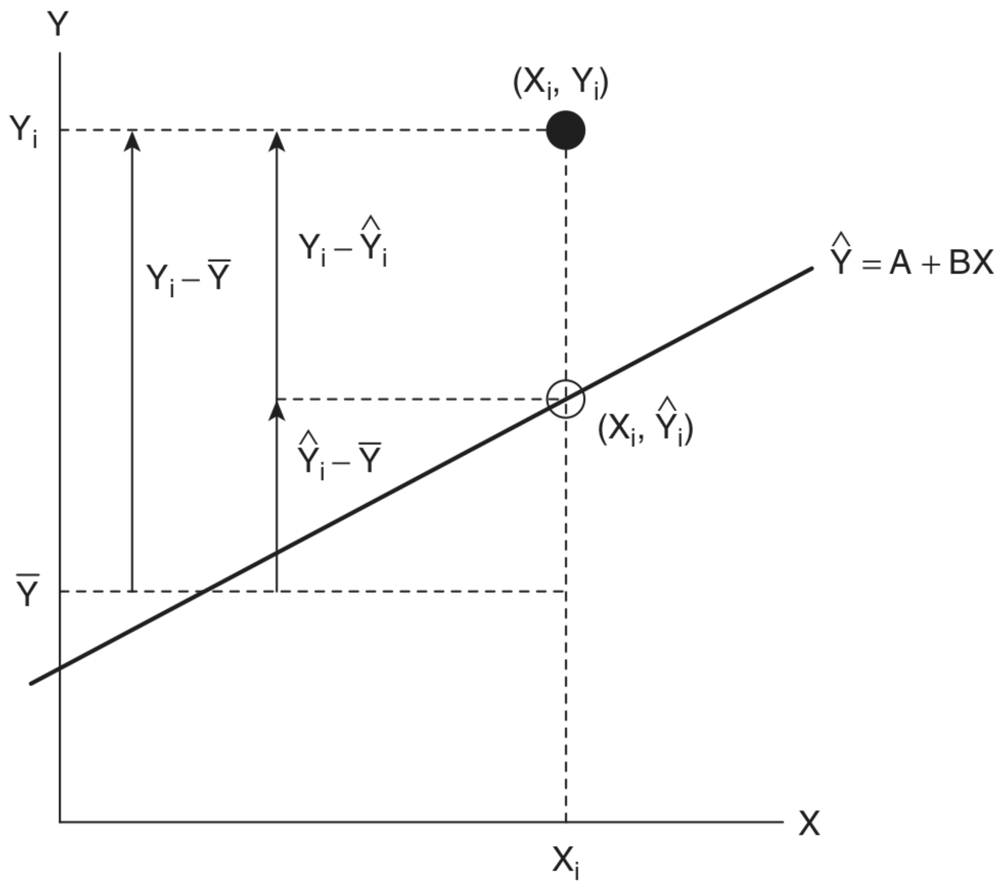
Or we can use the `confint()` function

```r
confint(lm2) %>% round(3)
```

```
##              2.5 % 97.5 %
## (Intercept) -2.750 -0.947
## temperature  0.056  0.079
```

In-sample performance

Decomposing the error



$$\underbrace{\sum_{i=1}^{n}(y_i - \bar{y})^2}_{SST_o} = \underbrace{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}_{SST} + \underbrace{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}_{SSR}$$

where

- SST_o is the **total** variation in Y
- SST is the sum of squares **explained** by the regression line
- $SSR = SST_o - SST$ is the variation in Y remain **unexplained**

Image source: Fox (2016; Figure 5.5 p. 91)

Coefficient of determination r^2

The square of correlation coefficient r^2 called the **coefficient of determination** measures the proportion of *total* variation in Y *explained* by the linear regression model:

It is "one minus the proportion of variation not explained by the model":

$$r^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{SSR}{SST_o}.$$

Hence the **coefficient of determination** r^2 measures the strength of the linear relationship between x and y by the percentage of variation in y explained by the linear regression model in x.

```r
summary(lm2)
```

```
##
## Call:
## lm(formula = lozone ~ temperature, data = environmental)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.14417 -0.32555  0.02066  0.34234  1.49100
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.848518   0.455080  -4.062  9.2e-05 ***
## temperature  0.067673   0.005807  11.654  < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5804 on 109 degrees of freedom
## Multiple R-squared:  0.5548,	Adjusted R-squared:  0.5507
## F-statistic: 135.8 on 1 and 109 DF,  p-value: < 2.2e-16
```

The r^2 in the ozone example is 0.5548.

Interpretation: We can say that temperature explains 55% of the observed variation in the logarithm of ozone concentration.

Can we do better if we use more variables to help explain the logarithm of ozone concentration?

References

- This module will largely follow a few chapters from Baumer, Kaplan, and Horton (2017).
- It is available on Canvas through the Reading List tab. You can download the relevant chapters.
    - II: Statistics and Modeling
        - Chapter 8 Statistical learning and predictive analytics
        - Chapter 9 Unsupervised learning
    - IV: Appendix E Regression modeling [freely available from the book website]

Baumer, B. S., D. T. Kaplan, and N. J. Horton (2017). *Modern Data Science with R*. Boca Raton: Chapman and Hall/CRC. URL: https://mdsr-book.github.io/index.html.

Fox, J. (2016). *Applied regression analysis and generalized linear models*. 3rd ed. Thousand Oaks, California: SAGE.