Data Science Concepts and Analysis

Week 7: Linear models

- Review of the simple linear model
- Multiple linear regression: linear models with many variables
- Case study: urban tree cover

This week: multiple linear regression

Objective: extend the simple linear model to multiple explanatory variables.

- **Review of the simple linear model**
    - The simple linear model in matrix form
    - Estimation and uncertainty quantification
    - Parameter interpretation
- **Multiple linear regression**
    - The linear model in matrix form
    - Estimation and uncertainty quantification
- **Case study: urban tree cover**
    - Background
    - Model 1: summer temperatures, tree cover, and income
        - model fitting calculations, step by step
        - model visualization
        - interpretation of results
    - Model 2: adding population density
        - categorical variable encodings
        - results and interpretation

Review of the simple linear model

- The simple linear model in matrix form
- Estimation and uncertainty quantification
- Parameter interpretation

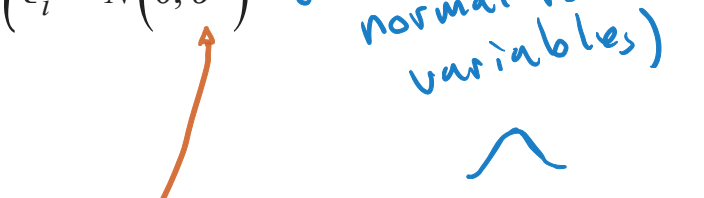
The simple linear model

The **simple linear model** describes a quantitative variable y as a linear function of another variable x and a random error for n observations:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \qquad \begin{cases} i = 1, \ldots, n \\ \epsilon_i \sim N(0, \sigma^2) \end{cases}$$

(indexes observations)

(errors are normal random variables)



- y_i is the ***response variable***
- x_i is the ***explanatory variable***
- ϵ_i is the ***error***
- $\beta_0, \beta_1, \sigma^2$ are the ***model parameters***
  - β_0 is the ***intercept***
  - β_1 is the ***coefficient***
  - σ^2 is the ***error variance***

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad \Longleftrightarrow \quad \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$y = X\beta + \varepsilon$$

Estimation

The parameters β_0, β_1 are estimated by **ordinary least squares** (OLS), which are best under many conditions.

The OLS estimates have a closed form:

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y}$$

An estimate of the error variance is:

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right)^2 = \frac{1}{n-2}\underbrace{\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)'\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)}$$

measure of residual variation

Uncertainty quantification

Model **uncertainty** can be thought of in terms of the ***variation in parameter estimates***: estimates that vary a lot from sample to sample are less certain.

The variances and covariances of the estimates have a closed form:

$$\mathbf{V} = \begin{bmatrix} \text{var}\hat{\beta}_0 & \text{cov}\left(\hat{\beta}_0, \hat{\beta}_1\right) \\ \text{cov}\left(\hat{\beta}_0, \hat{\beta}_1\right) & \text{var}\hat{\beta}_1 \end{bmatrix} = \sigma^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}$$

This matrix is estimated by plugging in $\hat{\sigma}^2$ (the estimate) for σ^2:

$$\hat{\mathbf{V}} = \begin{bmatrix} \hat{v}_{11} & \hat{v}_{12} \\ \hat{v}_{21} & \hat{v}_{22} \end{bmatrix} = \hat{\sigma}^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}$$

The square roots of the diagonal elements give estimated standard deviations, which are known as *standard errors*:

$$\text{SE}(\hat{\beta}_0) = \sqrt{\hat{v}_{11}} \qquad \text{and} \qquad \text{SE}(\hat{\beta}_1) = \sqrt{\hat{v}_{22}}$$

For about 95% of samples, estimates will vary within about 2 standard errors.

Parameter interpretations

Parameter interpretations are based on the observation that under the model:

$$\mathrm{E}y_i = \mathrm{E}\left(\beta_0 + \beta_1 x_i + \epsilon_i\right) = \beta_0 + \beta_1 x_i + \mathrm{E}\left(\epsilon_i\right) = \beta_0 + \beta_1 x_i$$

- (Intercept) [at $x_i = 0$] the mean [response variable] is estimated to be [$\hat{\beta}_0$ units].

$$x_i = 0 \quad \Longrightarrow \quad \mathrm{E}y_i = \beta_0 + \beta_1(0) = \beta_0$$

- (Slope) Every [one-unit increase in x_i] is associated with an estimated change in mean [response variable] of [$\hat{\beta}_1$ units].

$$\text{increment } x \quad \Longrightarrow \quad \beta_0 + \beta_1(x_i + 1) = \beta_0 + \beta_1 x_i + \beta_1 = \mathrm{E}y_i + \beta_1$$

Example from last time

Last time we fit a simple linear model to the SEDA math gap data. 'Fitting' the model means computing the estimates.

In [3]:
```
(scatter + line + band).properties(width = 300, height = 200)
```

Out[3]:

In [4]:
```
coef_summary
```

Out[4]:

	estimate	standard error
intercept	-1.356170	0.130697
log median income	0.121057	0.011843

Multiple linear regression

- The linear model in matrix form
- Estimation and uncertainty quantification

Extending the simple linear model

The simple linear model was:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \begin{cases} i = 1, \ldots, n \\ \epsilon_i \sim N\left(0, \sigma^2\right) \end{cases} \qquad \text{(simple linear model)}$$

It's called 'simple' because it only has a single explanatory variable x_i.

The **linear model** is a direct extension of the simple linear model to $p-1$ variables $x_{i1}, \ldots, x_{i,p-1}$:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i,p-1} + \epsilon_i \qquad \begin{cases} \epsilon_i \sim N(0, \sigma^2) \\ i = 1, \ldots, n \end{cases} \qquad \text{(linear model)}$$

Other names: this is sometimes also called the *multiple regression model* or *multiple linear model*.

The linear model in matrix form

The linear model is often written observation-wise in *indexed* form as above: i indexes the observations. However, it's much more concise in matrix form:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

This is shorthand for:

$$\mathbf{y}: \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n\times 1} = \mathbf{X}: \begin{bmatrix} 1 & x_{11} & \cdots & x_{1,p-1} \\ 1 & x_{21} & \cdots & x_{2,p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{n,p-1} \end{bmatrix}_{n\times p} \times \beta: \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}_{p\times 1} + \epsilon: \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}_{n\times 1}$$

Carrying out the arithmetic on the right-hand side:

Good news!

Estimation and uncertainty quantification are **exactly the same as in the simple linear model**.

The OLS estimates are:

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \vdots \\ \hat{\beta}_{p-1} \end{bmatrix} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y}$$

An estimate of the error variance is:

$$\hat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \cdots - \hat{\beta}_{p-1}x_{i,p-1}\right)^2 = \frac{1}{n-p}\ \left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)'\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)$$

measure of residual variation

The simple linear model was the special case with $p = 2$.

Uncertainty quantification

In the case of the multiple linear model, the variances and covariances are a $p \times p$ (instead of 2×2) matrix:

$$\mathbf{V} = \begin{bmatrix} \mathrm{var}\hat{\beta}_0 & \mathrm{cov}\left(\hat{\beta}_0, \hat{\beta}_1\right) & \cdots & \mathrm{cov}\left(\hat{\beta}_0, \hat{\beta}_{p-1}\right) \\ \mathrm{cov}\left(\hat{\beta}_0, \hat{\beta}_1\right) & \mathrm{var}\hat{\beta}_1 & \cdots & \mathrm{cov}\left(\hat{\beta}_1, \hat{\beta}_{p-1}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}\left(\hat{\beta}_0, \hat{\beta}_{p-1}\right) & \mathrm{cov}\left(\hat{\beta}_1, \hat{\beta}_{p-1}\right) & \cdots & \mathrm{var}\hat{\beta}_{p-1} \end{bmatrix} = \sigma^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}$$

This matrix is again estimated by plugging in $\hat{\sigma}^2$ (the estimate) for σ^2:

$$\hat{\mathbf{V}} = \begin{bmatrix} \hat{v}_{11} & \hat{v}_{12} & \cdots & \hat{v}_{1p} \\ \hat{v}_{21} & \hat{v}_{22} & \cdots & \hat{v}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{v}_{p1} & \hat{v}_{p2} & \cdots & \hat{v}_{pp} \end{bmatrix} = \hat{\sigma}^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}$$

The square roots of the diagonal elements give *standard errors*:

Even better news!

All the computations in `sklearn` are exactly the same.

So let's have a look at some examples.

Case study: urban tree cover

- Background
- Model 1: summer temperatures, tree cover, and income
    - model fitting calculations, step by step
    - model visualization
    - interpretation of results
- Model 2: adding population density
    - categorical variable encodings
    - results and interpretation

Urban tree cover data

The following are data on urban tree cover in the San Diego area:

In [6]:
```
trees.head(4)
```

Out[6]:

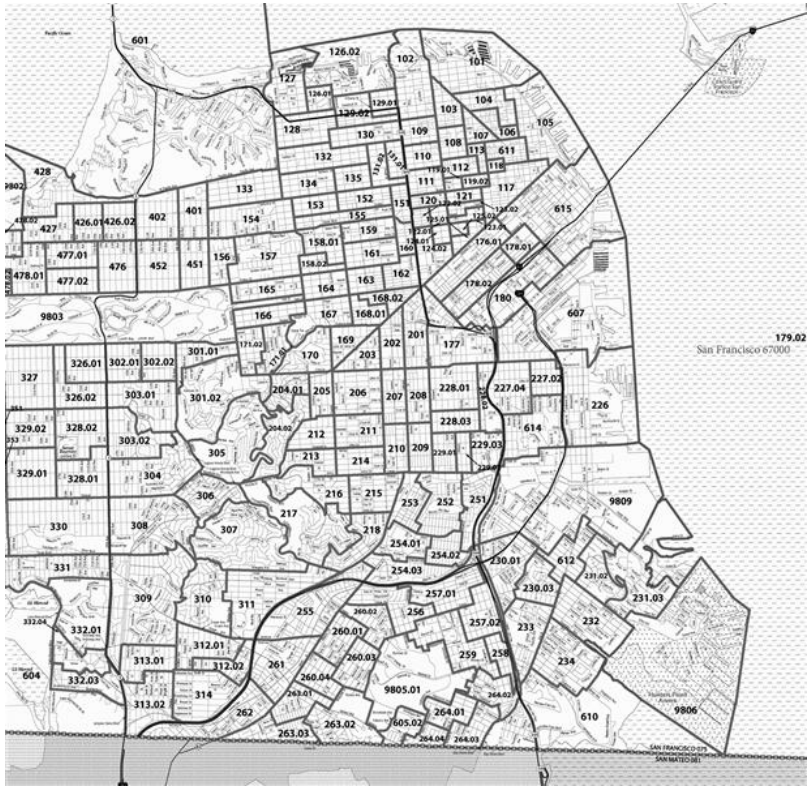
	Name	census_block_GEOID	tree_cover	mean_summer_temp	mean_income	income_level	pop_density
11013	San Diego, CA	6.073010e+13	0.158259	31.986364	40951	medium	high
18997	San Diego, CA	6.073020e+13	0.013488	34.068851	51502	high	very low
8786	San Diego, CA	6.073010e+13	0.052844	37.098611	28454	low	low
11163	San Diego, CA	6.073000e+13	0.217973	33.213636	40229	medium	medium

Source: McDonald RI, Biswas T, Sachar C, Housman I, Boucher TM, Balk D, et al. (2021) The tree cover and temperature disparity in US urbanized areas: Quantifying the association with income across 5,723 communities. PLoS ONE 16(4): e0249715. doi:10.1371/journal.pone.0249715.

Observational units

The **observational units** are census blocks.

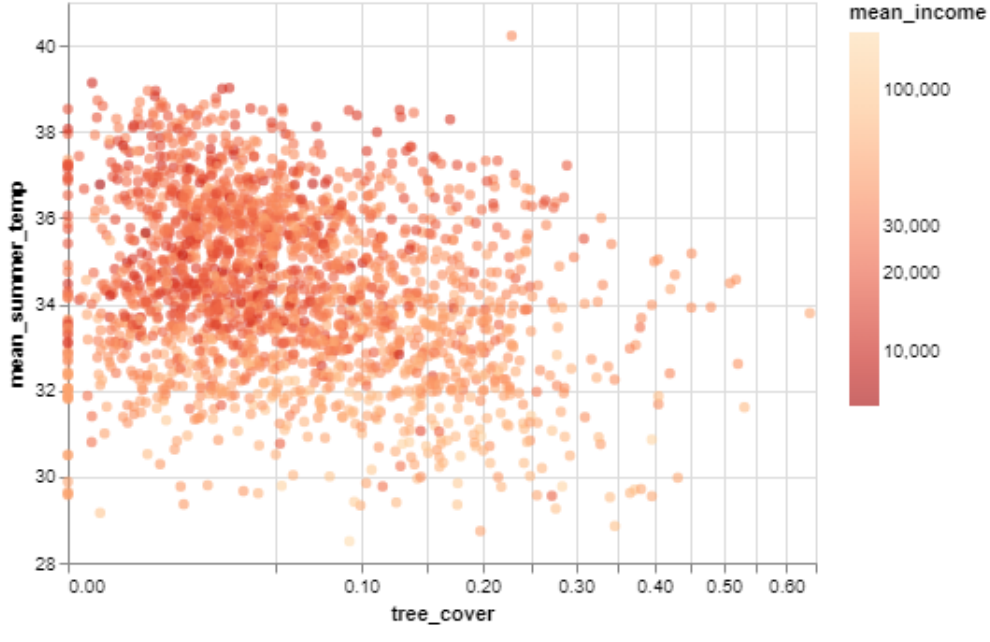
Census blocks are pretty small, about the size of a city block. To get an idea of the geographic scale, here's a map of census tracts in San Francisco. Each tract contains several census blocks:



The data comprise observations on a random sample of 1,998 census blocks in San Diego.

Tree cover and summer temperatures

Tree canopy mitigates temperature in urban areas.



For this reason, urban forestry projects have been advocated as a strategy for mitigating the effects of climate change.

Modeling objectives

Here we'll try to quantify the association between temperature, tree cover, income, and population density using multiple regression.

Let's start with a MLR model with just two explanatory variables, tree cover and income:

$$\underbrace{\text{temp}_i}_{y_i} = \beta_0 + \beta_1 \underbrace{\text{cover}_i}_{x_{i1}} + \beta_2 \underbrace{\text{income}_i}_{x_{i2}} + \epsilon_i \qquad i = 1, \ldots, 1998$$

In [8]:
```
y = trees.mean_summer_temp.values
x_df = trees[['tree_cover', 'mean_income']]
x_df.head(3)
```

Out[8]:

	tree_cover	mean_income
11013	0.158259	40951
18997	0.013488	51502
8786	0.052844	28454

Explanatory variable matrix

We'll need to create an explanatory variable matrix of the form:

$$\mathbf{X} = \begin{bmatrix} 1 & \text{cover}_1 & \text{income}_1 \\ 1 & \text{cover}_2 & \text{income}_2 \\ \vdots & \vdots & \vdots \\ 1 & \text{cover}_{1998} & \text{income}_{1998} \end{bmatrix}$$

So we just need to add a column of ones to `x_df`:

In [9]:
```
# add column of ones (for intercept)
x_mx = add_dummy_feature(x_df, value = 1)

# preview
x_mx[0:3]
```

value for a constant column

data frame to modify

Out[9]:
```
array([[1.00000000e+00, 1.58258747e-01, 4.09510000e+04],
       [1.00000000e+00, 1.34878570e-02, 5.15020000e+04],
       [1.00000000e+00, 5.28437900e-02, 2.84540000e+04]])
```

Model fitting

Since we've added an intercept column, the fitting module should be configured **not** to fit an intercept separately.

In [10]:
```
# fit first model
mlr = lm.LinearRegression(fit_intercept = False)
mlr.fit(x_mx, y)
mlr.coef_
```

Out[10]:
```
array([ 3.65559763e+01, -3.14810456e+00, -5.02075353e-05])
```

Error variance estimate

Next we'll calclulate $\hat{\sigma}^2$. For this we need the fitted values:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$$

(fitted)

And residuals:

(resid)

$$\mathbf{e} = \left(\mathbf{y} - \mathbf{X}\hat{\beta}\right) = \mathbf{y} - \hat{\mathbf{y}} = y - X\hat{\beta}$$

In [11]:
```
# fitted values and residuals
fitted = mlr.predict(x_mx)
resid = y - fitted
```

Then the error variance estimate is:

$$\hat{\sigma}^2 = \frac{1}{n-p}\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)'\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right) = \frac{1}{n-p}\mathbf{e}'\mathbf{e} = \frac{n-1}{n-p}S_e^2$$

$S_e^2 = \frac{1}{n-1}\sum_{i=1}^{n}(e_i - \bar{e})^2$ ($\bar{e} = 0$)

sample variance of $e_1, \ldots, e_n$, i.e., of the residuals

$= \frac{1}{n-p}\sum_{i=1}^{n} e_i^2$

In [12]:
```
# error variance estimate
n, p = x_mx.shape
sigmasqhat = resid.var()*(n - 1)/(n - p)
sigmasqhat
```

S_e^2 $\quad \frac{n-1}{n-p}$

Uncertainty quantification

Lastly, we'll compute the coefficient estimate standard errors:

$$\sqrt{\hat{v}_{jj}} \quad \text{from} \quad \hat{\mathbf{V}} = \hat{\sigma}^2 \left(\mathbf{X}'\mathbf{X} \right)^{-1}$$

In [13]:
```
xtx = x_mx.transpose().dot(x_mx) # X'X
xtxinv = np.linalg.inv(xtx) # (X'X)^{-1}
vhat = sigmasqhat*xtxinv # V
vhat
```

Out[13]:
```
array([[ 6.72240283e-03, -6.63125605e-03, -1.25915361e-07],
       [-6.63125605e-03,  2.23049324e-01, -2.70769667e-07],
       [-1.25915361e-07, -2.70769667e-07,  3.80729809e-12]])
```

In [14]:
```
coef_se = np.sqrt(vhat.diagonal()) # (v_jj)^{1/2}
coef_se
```

Out[14]:
```
array([8.19902606e-02, 4.72280980e-01, 1.95122989e-06])
```

Quality of fit: R^2

There are many metrics that measure fit quality. The most common is the **proportion of variation explained by the model**, which is ***denoted by*** R^2:

$$R^2 = \frac{\text{reduction in variation}}{\text{total variation}} = \frac{\tilde{\mathbf{y}}'\tilde{\mathbf{y}} - \mathbf{e}'\mathbf{e}}{\tilde{\mathbf{y}}'\tilde{\mathbf{y}}} \qquad \text{where} \qquad \tilde{\mathbf{y}} = \mathbf{y} - \bar{\mathbf{y}}$$

$\Rightarrow \hat{\tilde{y}}'\hat{\tilde{y}} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$

i.e. variation in response

In [15]:
```
# 'by hand'
y_ctr = y - y.mean()
(y_ctr.transpose().dot(y_ctr) - resid.transpose().dot(resid))/(y_ctr.transpose().dot(y_ctr))
```

Out[15]:
```
0.30684982283103096
```

In [16]:
```
# using sklearn.metrics.r2_score
r2_score(y, fitted)
```

Out[16]:
```
0.3068498228310309
```

Interpretation: 30% of variation in mean summer temperature is explained by tree cover and income.

Reporting model fit

Now we've computed relevant quantities -- estimates and standard errors -- but we have yet to report these in an organized fashion.

Typically, these are displayed in a table.

In [17]:
```
mlr_summary = pd.DataFrame(
    {'estimate': np.append(mlr.coef_, sigmasqhat),
     'standard error': np.append(coef_se, float('nan'))},
    index = ['interept', 'cover', 'income', 'error variance']
)

mlr_summary
```

Out[17]:

	estimate	standard error
interept	36.555976	0.081990
cover	-3.148105	0.472281
income	-0.000050	0.000002
error variance	2.729421	NaN

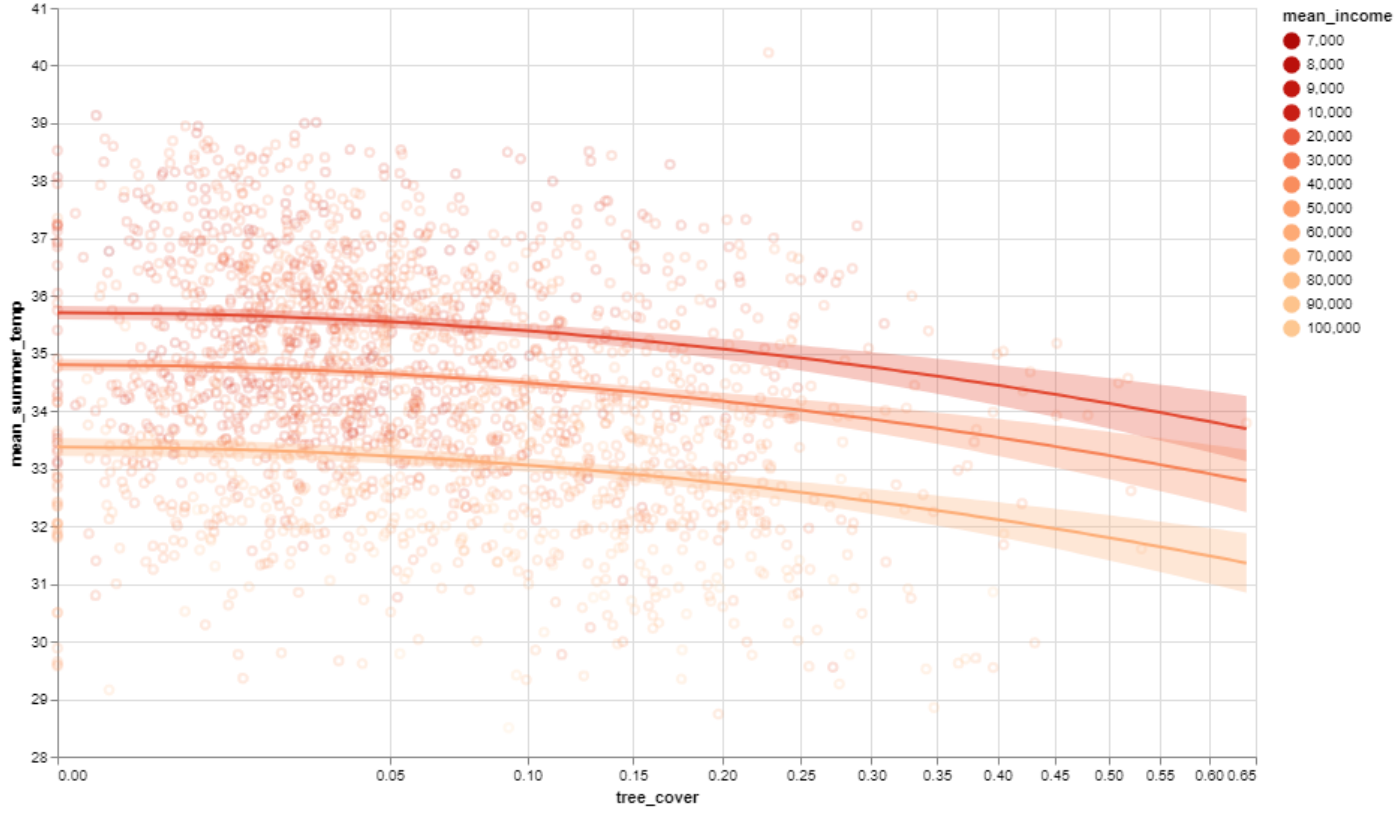
Along with R^2:

In [18]:
```
r2_score(y, fitted)
```

Out[18]:
```
0.3068498228310309
```

Visualization

Here are trend lines and error bands for three values of income (10th, 50th, and 90th percentiles).



The error bands represent the variation of the lines -- how much they might change if we sampled different census blocks. (**Not** the variation of temperatures.)

Interpretation

So what have we learned from this exercise?

In [20]:
```
mlr_summary
```

Out[20]:

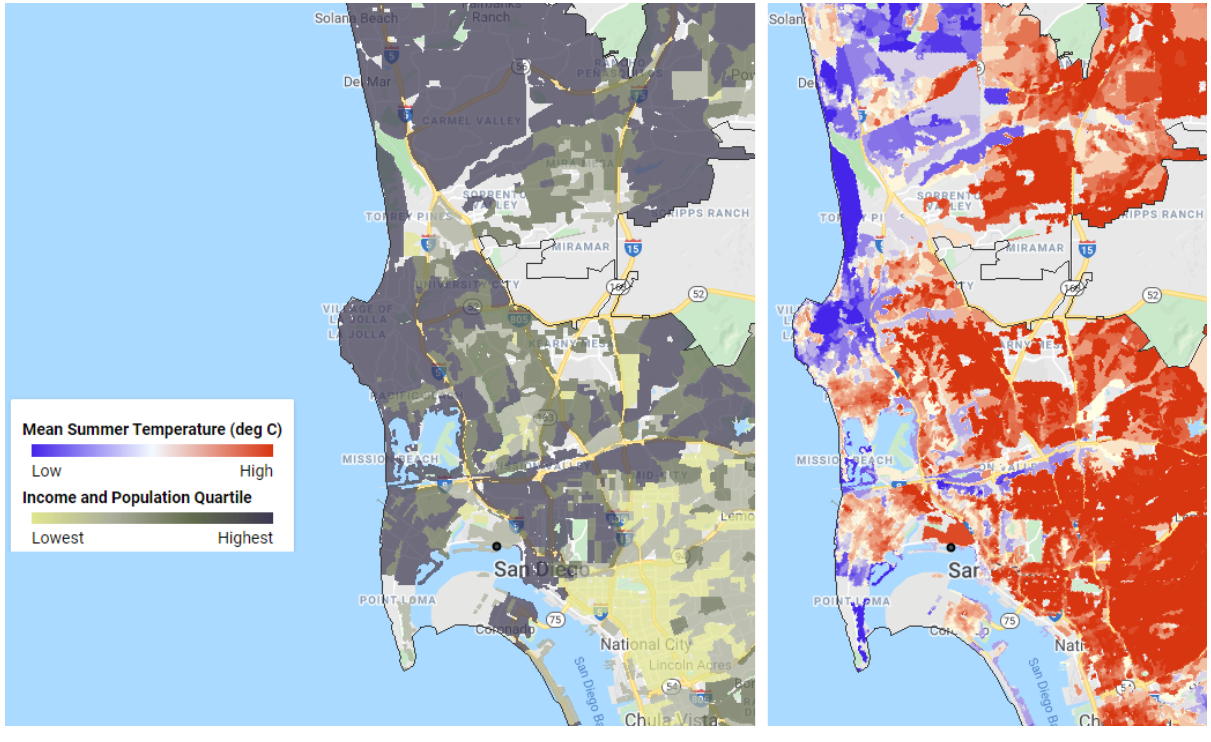
	estimate	standard error
interept	36.555976	0.081990
cover	-3.148105	0.472281
income	-0.000050	0.000002
error variance	2.729421	NaN

- Among census blocks in San Diego, a 1% increase in tree canopy cover is associated with a decrease in average summer temperatures of 3.15 degrees Celsius, after accounting for mean income of the census block.
- Among census blocks in San Diego, a $10K increase in mean income is associated with a decrease in average summer temperatures of 0.5 degrees Celsius, after accounting for tree canopy cover.
- There's still lots of unexplained local variation in average summer temperatures; low R^2.

Explaining associations

There are various mechanisms by which tree canopy reduces temperatures: providing shade; improving air quality. But what explains the weaker association between income and temperature?

Well, one possibility is that property values are higher near the ocean.



So income may simply be a proxy for distance to the ocean; and temperatures are cooler near the ocean. This is an excellent example of **confounding**!

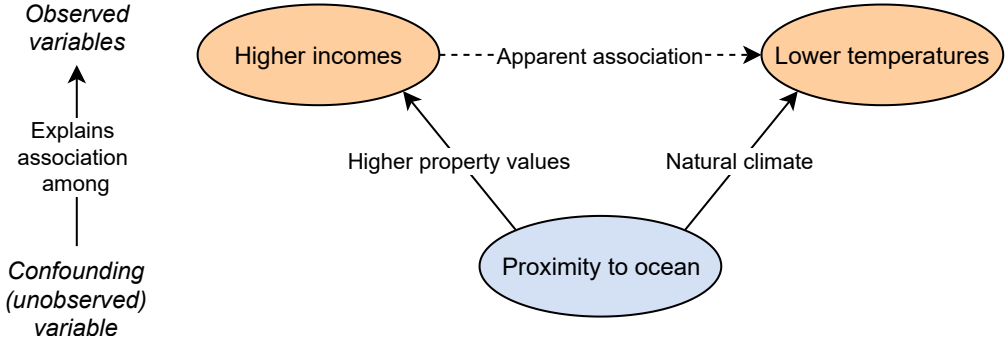
Confounding

In statistical jargon, we'd say that proximity to the ocean is a *confounding factor*:

Confounding

In statistical jargon, we'd say that proximity to the ocean is a *confounding factor*:

- it is correlated with higher incomes (the explanatory variable);
- it is also correlated with lower temperatures (the response);
- and so it 'explains away' the apparent association.



This kind of phenomenon only affects interpretation; *we should still keep income in the model*.

MLR with categorical variables

What if we also incorporated the population density factor (a categorical variable)?

In [21]:
```
trees.head(4)
```

Out[21]:

	Name	census_block_GEOID	tree_cover	mean_summer_temp	mean_income	income_level	pop_density
11013	San Diego, CA	6.073010e+13	0.158259	31.986364	40951	medium	high
18997	San Diego, CA	6.073020e+13	0.013488	34.068851	51502	high	very low
8786	San Diego, CA	6.073010e+13	0.052844	37.098611	28454	low	low
11163	San Diego, CA	6.073000e+13	0.217973	33.213636	40229	medium	medium

Well, it might be natural to think we could just append this column as another variable x_{i3}:

$$\underbrace{\text{temp}_i}_{y_i} = \beta_0 + \beta_1 \underbrace{\text{cover}_i}_{x_{i1}} + \beta_2 \underbrace{\text{income}_i}_{x_{i2}} + \beta_3 \underbrace{\text{density}_i}_{x_{i3}} + \epsilon_i \qquad i = 1, \ldots, 1998$$

But this doesn't quite make sense, because the *values* of density_i would be *words*! So...

$$\beta_3 \times \text{low} = \ ?$$

So we'll need to represent the categorical variable differently to include it in the model.

Indicator variable encoding

The solution to this issue is to **encode each level** of the categorical variable using an ***indicator***: a function whose value is zero or one to indicate a condition.

If we want to indicate whether a census block is of low population density, we can use the indicator:

$$I(\text{density} = \text{low}) = \begin{cases} 1 \text{ if population density is low} \\ 0 \text{ otherwise} \end{cases}$$

We can encode the levels of `pop_density` using a collection of indicators:

In [22]:
```
density_encoded = pd.get_dummies(trees.pop_density, drop_first = True)
pd.concat([trees[['pop_density']], density_encoded], axis = 1).head(4)
```

Out[22]:

	pop_density	low	medium	high
11013	high	0	0	1
18997	very low	0	0	0
8786	low	1	0	0
11163	medium	0	1	0

This captures all the information about the categorical variable in quantitative terms.

The MLR model with indicators

The model with the encoded population density variable is:

$$\underbrace{\text{temp}_i}_{y_i} = \beta_0 + \underbrace{\beta_1 \text{cover}_i}_{x_{i1}} + \underbrace{\beta_2 \text{income}_i}_{x_{i2}} + \underbrace{\beta_3 \text{low}_i}_{x_{i3}} + \underbrace{\beta_4 \text{med}_i}_{x_{i4}} + \underbrace{\beta_5 \text{high}_i}_{x_{i5}} + \epsilon_i \qquad i = 1, \ldots, 1998$$

The *effect* of doing this is to allow the model to have different intercepts for each population density group.

$$\text{density} = \text{very low} \implies \text{Etemp}_i = \underbrace{\beta_0}_{\text{intercept}} + \beta_1 \text{cover}_i + \beta_2 \text{income}_i \qquad \text{density} = \text{low} \implies \text{Etemp}_i$$

In matrix form

The explanatory variable matrix $\mathbf{X}$ for this full model ('full' because it includes all variables) will be of the form:

$$\mathbf{X} = \begin{bmatrix} 1 & \text{cover}_1 & \text{income}_1 & \text{low}_1 & \text{med}_1 & \text{high}_1 \\ 1 & \text{cover}_2 & \text{income}_2 & \text{low}_2 & \text{med}_2 & \text{high}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \text{cover}_{1998} & \text{income}_{1998} & \text{low}_{1998} & \text{med}_{1998} & \text{high}_{1998} \end{bmatrix}$$

Here's everything to the right of the dashed partition:

In [23]:
```
x_full_df = pd.concat([x_df, density_encoded], axis = 1)
x_full_df.head(4)
```

Out[23]:

	tree_cover	mean_income	low	medium	high
11013	0.158259	40951	0	0	1
18997	0.013488	51502	0	0	0
8786	0.052844	28454	1	0	0
11163	0.217973	40229	0	1	0

Fit summary

The remaining calculations are all the same as before. (You can see the source code for these slides if you're interested in reviewing the details.)

Here is the model fit summary:

In [25]:
```
# print
mlr_full_summary
```

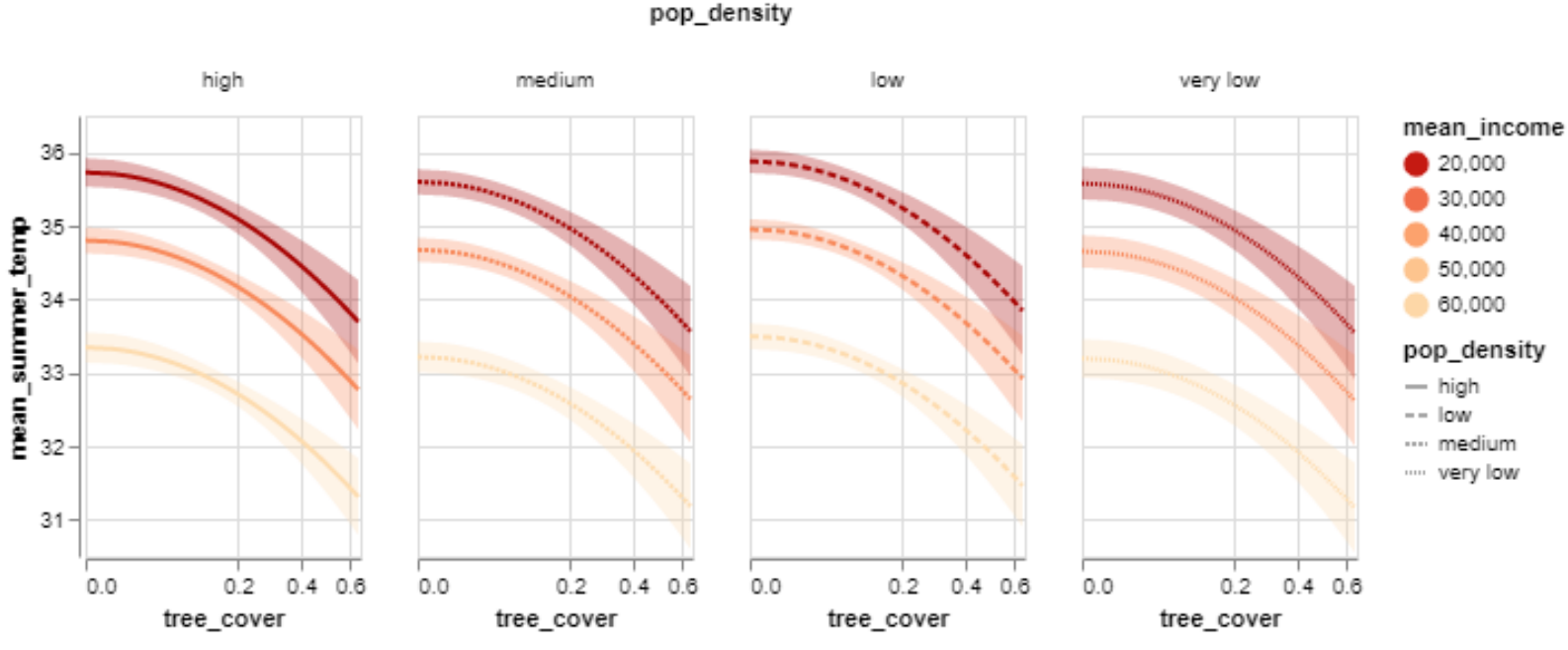
Out[25]:

	estimate	standard error
intercept	36.593462	0.114941
cover	-3.177783	0.492391
income	-0.000051	0.000002
low density	0.151946	0.095099
medium density	-0.128064	0.107888
high density	-0.149409	0.132829
error variance	2.719294	NaN

- Estimates for the cover and income coefficients are about the same.
- The population density variable changes the intercept by about ± 0.15 degrees Celsius, depending on the density level.
- So the association between temperature and population density appears negligible.

Model visualization

The estimated trends and error bands for the same three income levels and each level of population density are shown below, without the data scatter:



Comments on scope of inference

The data in this case study are from a *random sample* of census blocks in the San Diego urban area.

They are therefore representative of *all* census blocks in the San Diego urban area (population).

So the results are generliazable, meaning:

- The model approximates the *actual* associations between summer temperatures, tree cover, income, and population density in the region.

Summary

This week focused on extending the simple linear model to multiple explanatory variables.

- The **linear model** represents a quantitative response variable y_i as a linear function of several explanatory variables and a random error:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i,p-1} + \epsilon_i$$

- When represented in matrix form $\mathbf{y} = \mathbf{X}\beta + \epsilon$, all calculations are the same as in the simple linear model.
    - OLS estimates: $\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y}$
    - Error variance: $\hat{\sigma}^2 = \frac{1}{n-p}\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)'\left(\mathbf{y} - \mathbf{X}\hat{\beta}\right)$
    - Uncertainty quantification: $\hat{\mathbf{V}} = \hat{\sigma}^2\left(\mathbf{X}'\mathbf{X}\right)^{-1}$