Multivariate Data

EES 4891/5891

Probability & Statistics for Geosciences

Jonathan Gilligan

Class #18: Thursday, March 20 2025

Setting Up

Setting Up

- Accept the GitHub Classroom assignment at https://classroom.github.com'/

Learning Goals

Learning Goals

Multivariate Data

Multivariate Data

- Univariate data:
  - Observations have measurements of one variable:
    - You collect a bunch of water samples and measure pH.
- Multivariate data:
  - Each observation measures multiple variables:
    - pH, salinity, dissolved oxygen
  - Time or date can be a variable:
    - Date and atmospheric CO_2 concentration
  - *Bivariate* data has 2 variables per observation
- Multivariate data let us examine relationships among different variables
  - Is there a relationship between pH and dissolved oxygen?

- Example: atmospheric CO_2 vs. global average temperature
  - Monthly CO_2 and temperature measurements

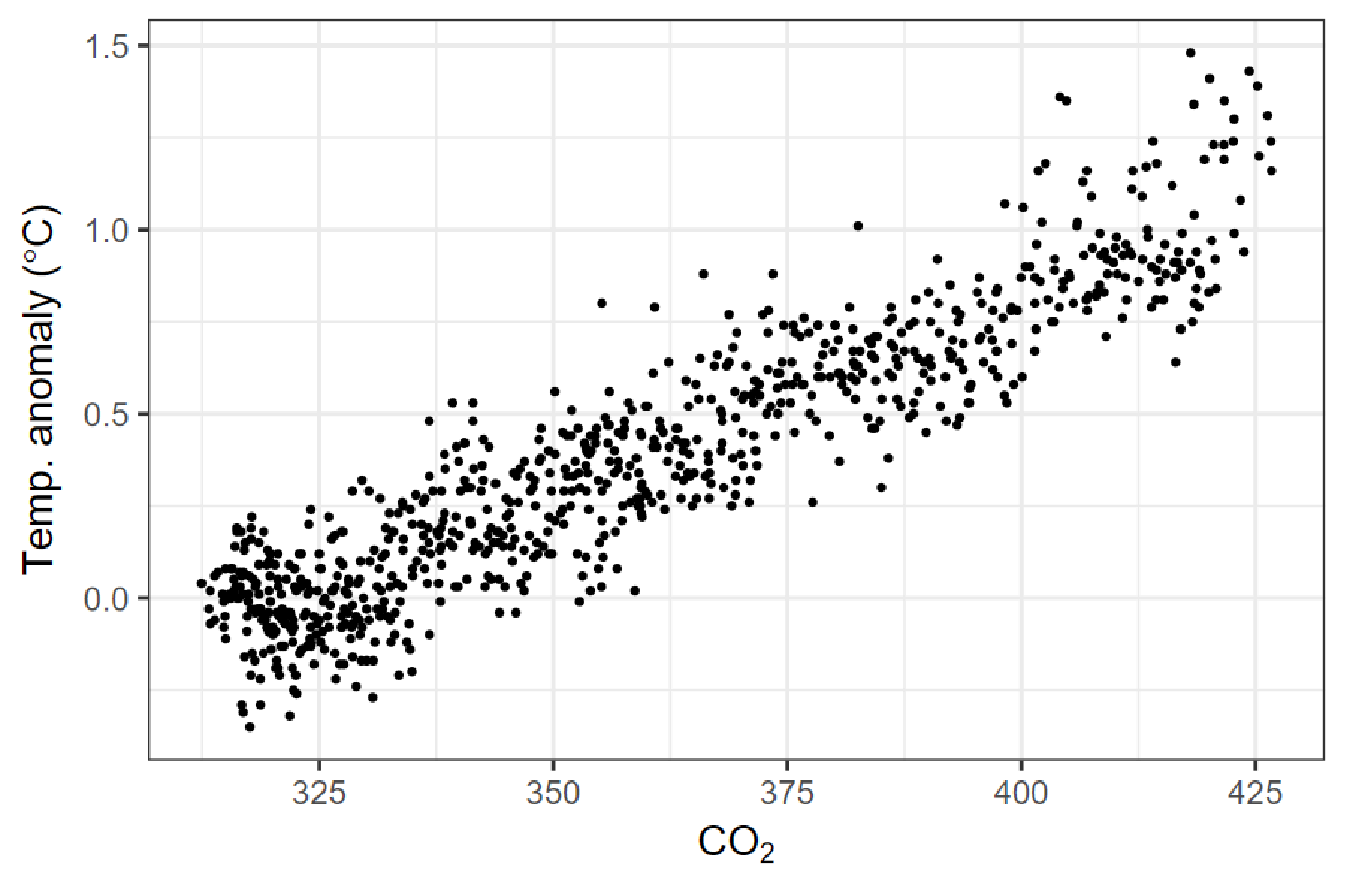
```
co2_t <- read_rds("co2_temp.rds")
glimpse(co2_t)
```

```
## Rows: 799
## Columns: 5
## $ year  <dbl> 1958, 1958, 1958, 1958, …
## $ month <ord> mar, apr, may, jun, jul,…
## $ time  <dbl> 1958.208, 1958.292, 1958…
## $ temp  <dbl> 0.08, 0.01, 0.06, -0.09,…
## $ co2   <dbl> 315.71, 317.45, 317.51, …
```

Plotting Bivariate Data

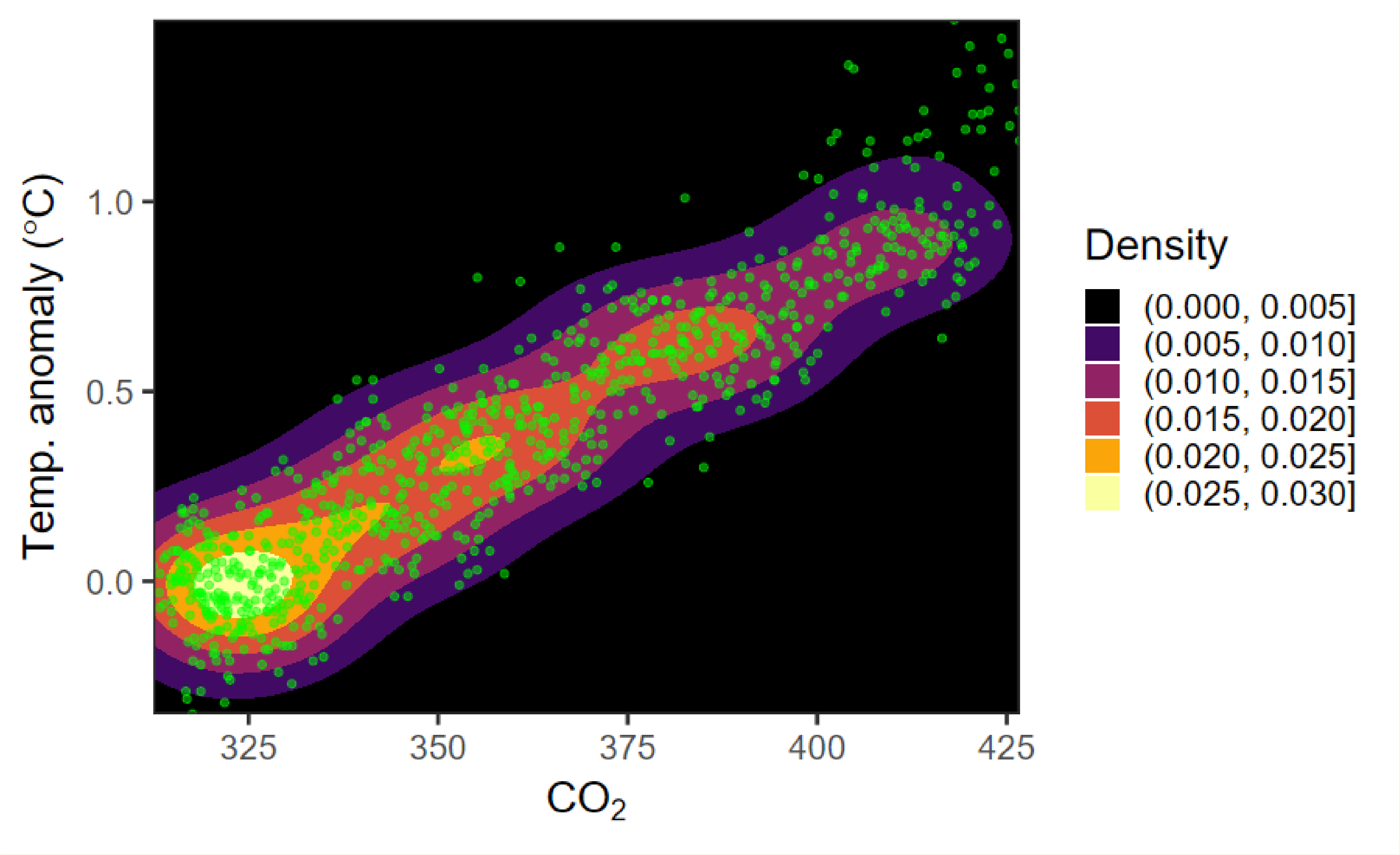
- Scatterplot

```
ggplot(co2_t, aes(x = co2, y = temp)) +
  geom_point() +
  labs(x = expression(CO[2]),
       y = expression(paste("Temp. anomaly (",
                            degree * C, ")"))
       )
```



- Kernel Density Plots

```
ggplot(co2_t, aes(x = co2, y = temp)) +
  geom_density_2d_filled() +
  geom_point(color = "green", alpha = 0.5) +
  scale_fill_viridis_d(option = "inferno", name =
        "Density") +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0)) +
  labs(x = expression(CO[2]),
       y = expression(paste("Temp. anomaly (", degree *
        C, ")")))
```



Joint & Marginal Distributions

- Multivariate data: N observations of p variables

$$X = \begin{bmatrix} (x_{1,1}, x_{1,2}, x_{1,3}, \ldots, x_{1,p}), \\ (x_{2,1}, x_{2,2}, x_{2,3}, \ldots, x_{2,p}), \\ \ldots, \\ (x_{N,1}, x_{N,2}, x_{N,3}, \ldots, x_{N,p}) \end{bmatrix}$$

  - Joint density: A point in p-dimensional space

$$f(x_1, x_2, x_3, \ldots, x_p)$$

- Simple case: Bivariate data

$$X = [(x_{1,1}, x_{1,2}), (x_{2,1}, x_{2,2}), \ldots, (x_{N,1}, x_{N,2})]$$

  - Joint density: a point on a plane, $f(x_1, x_2)$

- Marginal Density
  - The density of one variable, averaged over the oth

$$f_1 = \int_{x_2=-\infty}^{\infty} f(x_1, x_2)\,\mathrm{d}x_2$$
$$f_2 = \int_{x_1=-\infty}^{\infty} f(x_1, x_2)\,\mathrm{d}x_1$$

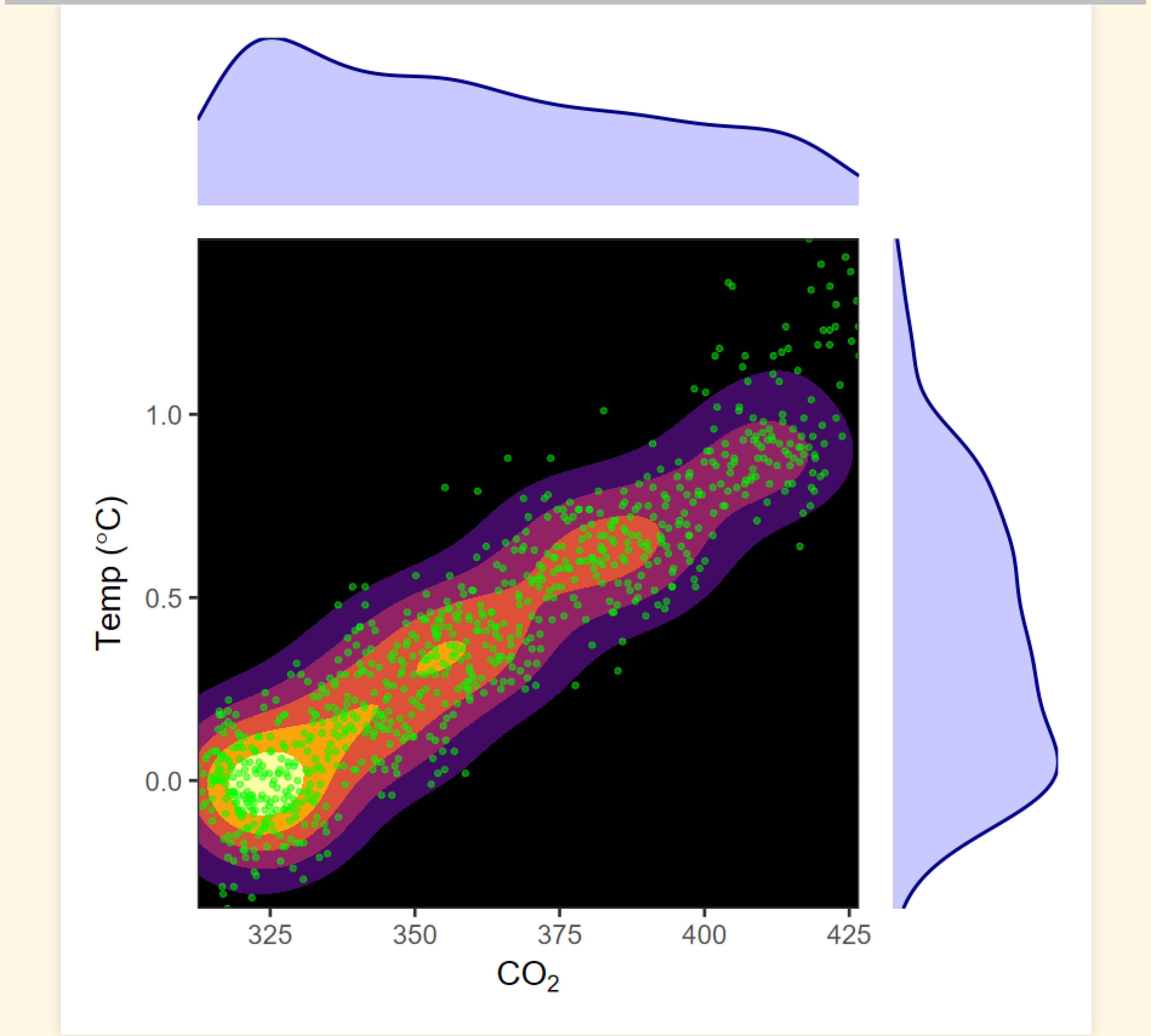
  - For p variables: The density of one variable, averag over all $p-1$ others:

$$f_1 = \int_{x_2=-\infty}^{\infty} \int_{x_3=-\infty}^{\infty} \ldots \int_{x_p=-\infty}^{\infty} f(x_1, x_2, x_3, \ldots, x_p)\,\mathrm{d}x_2\,\mathrm{d}x_3 \ldots \mathrm{d}x$$
$$f_2 = \int_{x_1=-\infty}^{\infty} \int_{x_3=-\infty}^{\infty} \ldots \int_{x_p=-\infty}^{\infty} f(x_1, x_2, x_3, \ldots, x_p)\,\mathrm{d}x_1\,\mathrm{d}x_3 \ldots \mathrm{d}x$$
$$\ldots$$
$$f_p = \int_{x_1=-\infty}^{\infty} \int_{x_2=-\infty}^{\infty} \ldots \int_{x_{p-1}=-\infty}^{\infty} f(x_1, x_2, x_3, \ldots, x_p)\,\mathrm{d}x_1\,\mathrm{d}x_2 \ldots \mathrm{d}$$

Plotting Marginal Density

```r
library(patchwork)

pxy <- ggplot(co2_t, aes(x = co2, y = temp)) +
  geom_density_2d_filled()  +
  geom_point(color = "green", alpha = 0.5) +
  scale_fill_viridis_d(option = "inferno", name =
        "Density",
                       guide = "none") +
  labs(x = expression(CO[2]),
       y = expression(paste("Temp (",
                            degree * C, ")"))) +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0))


px <- ggplot(co2_t, aes(x = co2)) +
  geom_density(linewidth = 1, color = "darkblue",
               fill = "blue", alpha = 0.2) +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(limits = c(0, NA), expand = c(0,0)) +
  theme_void()

py <- ggplot(co2_t, aes(x = temp)) +
  geom_density(linewidth = 1, color = "darkblue",
               fill = "blue", alpha = 0.2) +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(limits = c(0, NA), expand = c(0,0)) +
  theme_void() + coord_flip()
```

```r
px + plot_spacer() + pxy + py +
  plot_layout(ncol = 2, nrow = 2, widths = c(4, 1),
              heights = c(1, 4))
```



Conditional Distributions, Covariance, and Correlation

Conditional Distributions

- If the components $x_1, x_2, \ldots, x_p$ are independent, then the joint distribution is just the product of the marginal distributions:

$$f(x_1, x_2, \ldots, x_p) = f_1(x_1) f_2(x_2) \cdots f_p(x_p)$$

- But it's frequently the case that the variables are not independent.
- When variables are not independent, we use *conditional* distributions
  - For bivariate data (x_1, x_2),

$$\begin{aligned} g_2(x_1 | X_2 = x_2) &= f(x_1, x_2), \qquad & x_1 \text{ variable, } x_2 \text{ fixed} \\ g_1(x_2 | X_1 = x_1) &= f(x_1, x_2), \qquad & x_1 \text{ fixed, } x_2 \text{ variable} \end{aligned}$$

Covariance

- From the marginal distributions, we get means and variances:

$$\begin{aligned}
\mu_1 &= \int_{-\infty}^{\infty} f_1(x_1)x_1 \,\mathrm{d}x_1 \\
\sigma_1^2 &= \int_{-\infty}^{\infty} f_1(x_1)(x_1 - \mu_1)^2 \,\mathrm{d}x_1 \\
\mu_2 &= \int_{-\infty}^{\infty} f_1(x_2)x_2 \,\mathrm{d}x_2 \\
\sigma_2^2 &= \int_{-\infty}^{\infty} f_1(x_2)(x_2 - \mu_2)^2 \,\mathrm{d}x_2 \\
&\dots
\end{aligned}$$

- We get the *covariance* from the joint distribution:

$$\begin{aligned}
\mathrm{Cov}(X_1, X_2) =& E[(X_1 - E(X_1))(X_2 - E(X_2)) \\
=& E(X_1 X_2) - E(X_1)E(X_2) \\
=& \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1, x_2)x_1 x_2 \,\mathrm{d}x_1 \,\mathrm{d}x \\
& - \mu_1\mu_2 \\
\mathrm{Cov}(X_1, X_1) =& \mathrm{Var}(X_1) = \sigma_1^2
\end{aligned}$$

Covariance and Correlation

- The covariance has units, and this can be annoying, so we define *correlation* as a dimensionless quantity:

$$\rho_{X_i,X_j} = \mathrm{Cov}\left(\frac{X_i - \mu_i}{\sigma_i}, \frac{X_j - \mu_j}{\sigma_j}\right)$$

  - This is the Pearson correlation coefficient, which measures correlations assuming a linear relationship between the variables.
  - The Spearman correlation coefficient accounts for nonlinear correlations by comparing the *rank* of the data points, similar to a Q-Q plot.

- Correlation coefficients vary from -1 to $+1$.
  - $\rho = \pm 1$: The two variables have a perfectly linear relationship

Covariance Matrix

- We can represent the covariance of a *p*-dimensional multivariate data using a covariance matrix Σ:

$$\Sigma_{i,j} = \mathrm{Cov}(X_i, X_j)$$

$$\Sigma = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) & \cdots & \mathrm{Cov}(X_1, X_p) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \mathrm{Cov}(X_2, X_3) & \cdots & \mathrm{Cov}(X_2, X_p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_p, X_1) & \mathrm{Cov}(X_p, X_2) & \mathrm{Cov}(X_p, X_3) & \cdots & \mathrm{Var}(X_p) \end{pmatrix}$$

  - For bivariate data,

$$\Sigma = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) \end{pmatrix}$$

Covariance Matrix

- Covariance Matrix:

$$\Sigma = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

  - $\mathrm{Cov}(X_i, X_j) = \mathrm{Cov}(X_j, X_i)$, so the matrix is symmetric.

- Correlation matrix is like the covariance matrix, but with Pearson correlation coefficients instead of covariances:

$$R_{i,j} = \rho(X_i, X_j)$$

$$R = \begin{pmatrix} 1 & \rho(X_1, X_2) & \cdots \\ \rho(X_2, X_1) & 1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

  - Remember that $\rho(X_i, X_i) = 1$ because any variable is perfectly correlated with itself.

Multivariate Normal Distribution

Multivariate Normal Distribution

- Multivariate normal distribution:
  - Vector of p variables $x_1, x_2, \ldots, x_p$, where
    - Each x_i is normally distributed,
    - Covariances between variables defined by a covariance matrix.
- Start with bivariate normal ($p = 2$)
  - (X_1, X_2) follow a bivariate normal distribution if

$$X_1 \sim \mathcal{N}(\mu_1, \sigma_1)$$
$$X_2 \sim \mathcal{N}(\mu_2, \sigma_2)$$

- If X_1 and X_2 are *independent*, then $\Sigma_{1,2} = \Sigma_{2,1} = 0$ and $f(x_1, x_2) = f_1(x_1) f_2(x_2)$
- If X_1 and X_2 are not independent, then the formula for $f(x_1, x_2)$ involves complicated linear algebra with the covariance matrix.
  - Fortunately, we can use R to do the calculations, so we don't have to, using the package `mvtnorm`

Multivariate Normal in R

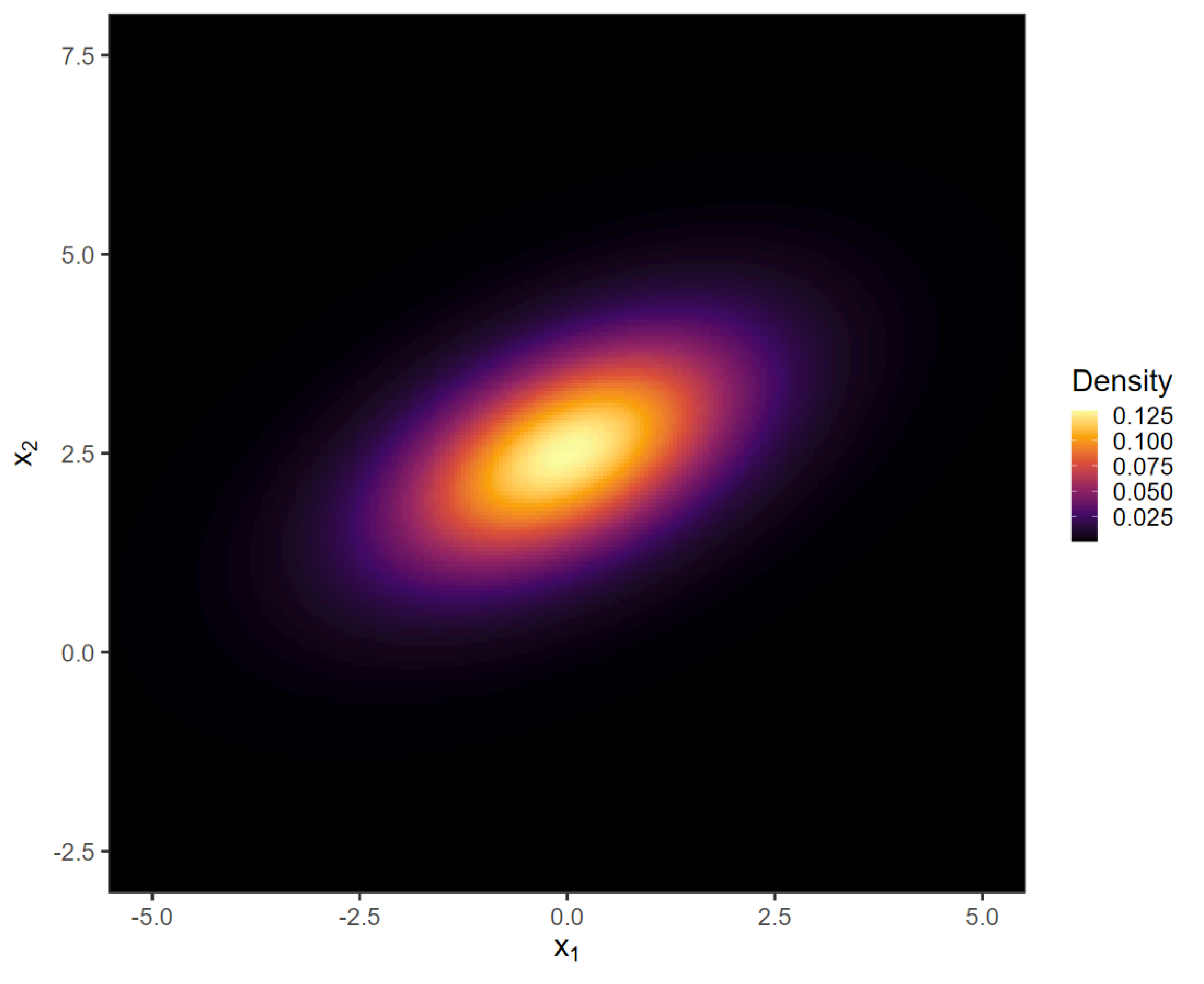
- `matrix()` takes a vector of data and turns it into a two-dimensional matrix.
- `expand.grid()` takes multiple vectors and creates a data frame containing all combinations.
- `dmvnorm()` takes an $n \times p$ matrix `x`, where each row has $x_1, x_2, \ldots, x_p$, a vector of means $\mu_1, \mu_2, \ldots, \mu_p$, and a covariance matrix, and returns a vector of probability densities.

```
# if necessary, install.packages("mvtnorm")
library(mvtnorm)

mu <- c(0, 2.5)
sigma <- matrix(c(2, 0.7, 0.7, 1), ncol = 2)

show(sigma)
```

```
##      [,1] [,2]
## [1,]  2.0  0.7
## [2,]  0.7  1.0
```

```
dens <- expand.grid(x1 = seq(-5.5, 5.5, 0.05),
                    x2 = seq(-3, 8, 0.05)) |>
  mutate(z  = dmvnorm(x = matrix(c(x1, x2), ncol = 2),
                      mean = mu, sigma = sigma))
```

```
ggplot(dens, aes(x = x1, y = x2)) +
  geom_raster(aes(fill = z)) +
  scale_fill_viridis_c(option = "inferno", name="Density") +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0)) +
  labs(x = expression(x[1]), y = expression(x[2]))
```

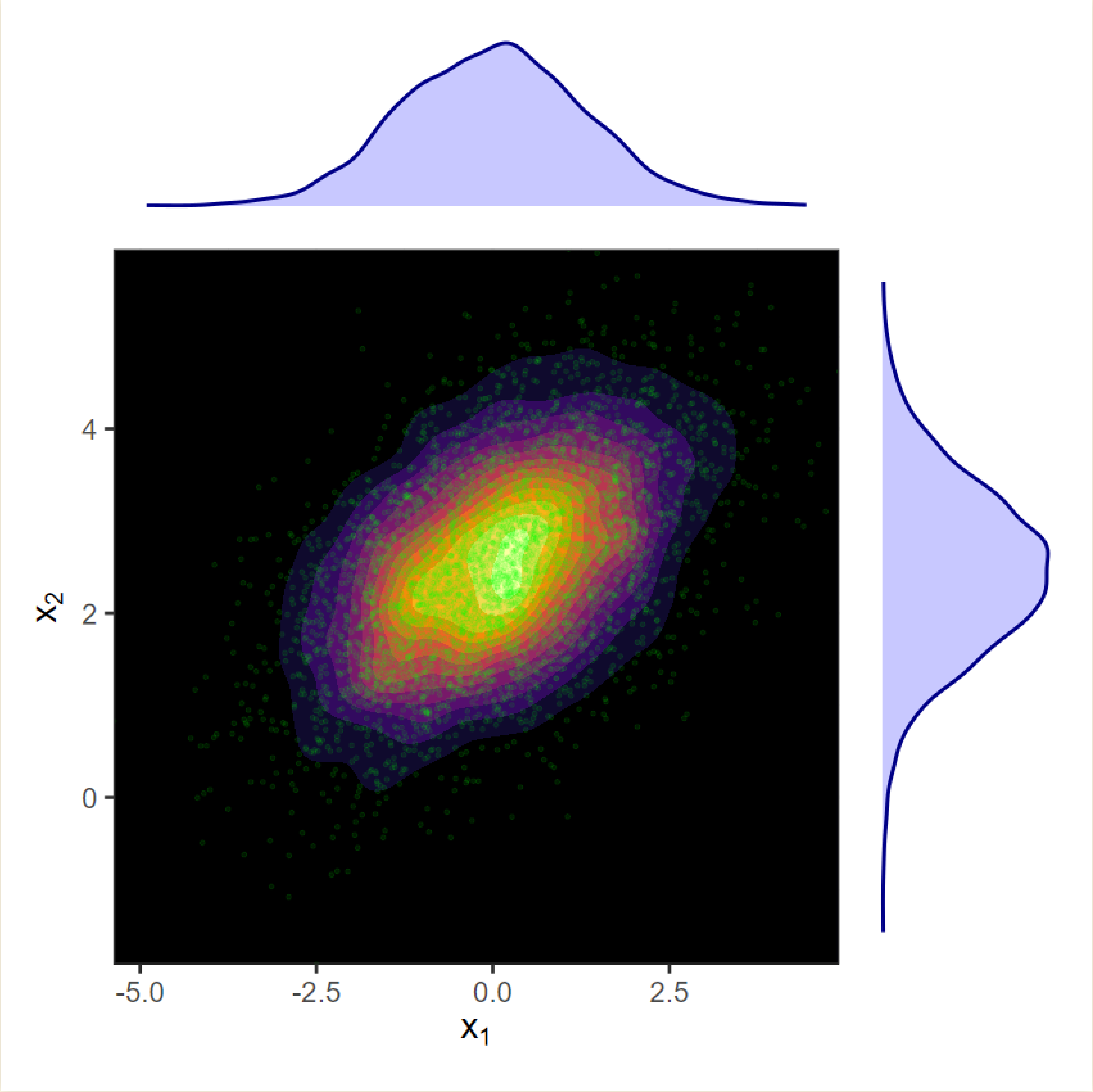


Plotting Multivariate Normal Data

- `rmvnorm()` generates an ***n*** × ***p*** matrix with ***n*** random samples from a ***p***-dimensional multivariate normal.

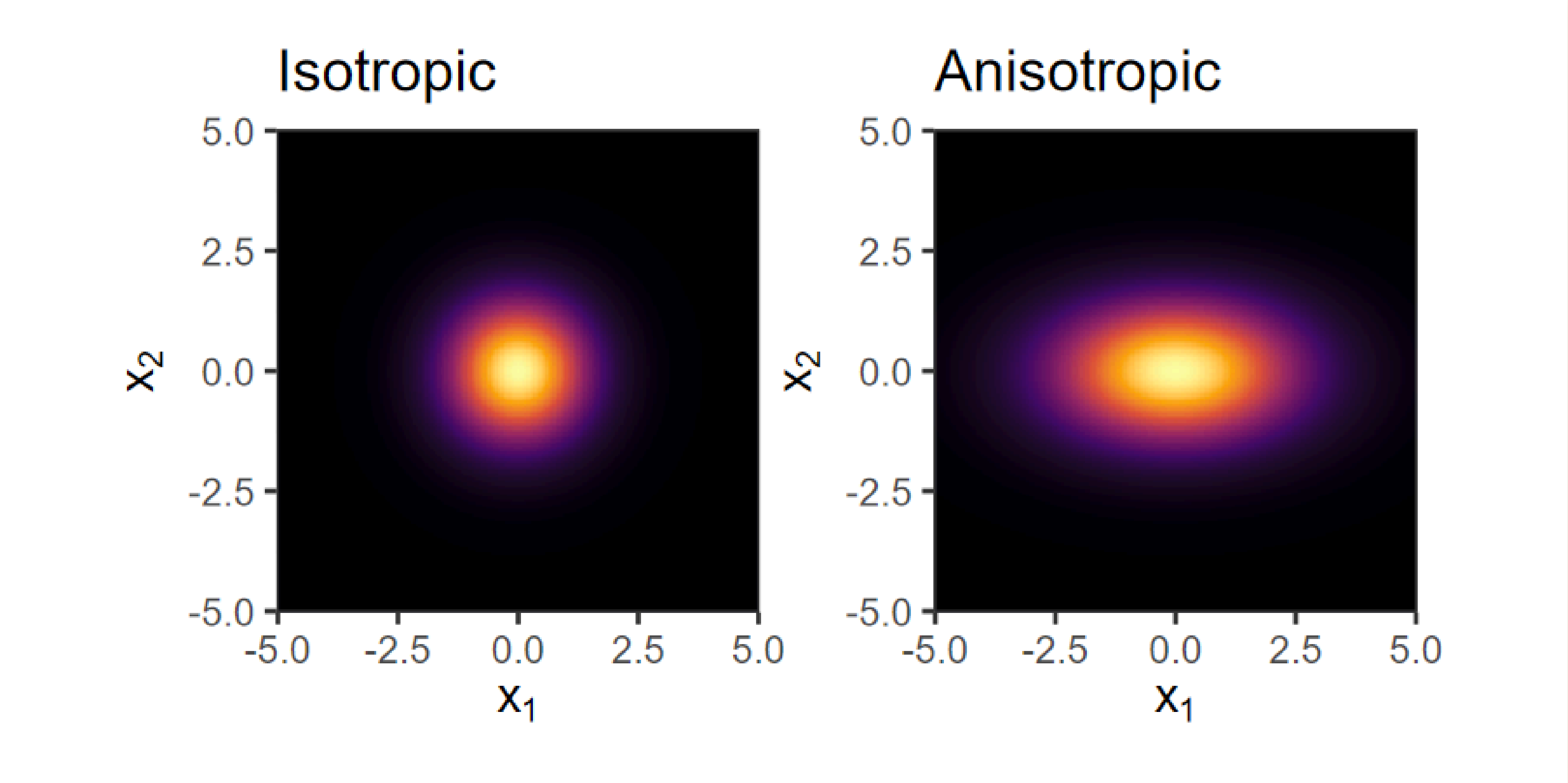
```
set.seed(123456)
samples <- rmvnorm(5000, mu, sigma) |> as_tibble()
names(samples) <- c("x1", "x2")

pxy <- ggplot(samples, aes(x = x1, y = x2)) +
  geom_density_2d_filled() +
  geom_point(color = "green", alpha = 0.1, size = 1) +
  scale_fill_viridis_d(option = "inferno", guide = "none")
         +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0)) +
  labs(x = expression(x[1]), y = expression(x[2]))

px <- ggplot(samples, aes(x = x1)) +
  geom_density(linewidth = 1, color = "darkblue",
               fill = "blue", alpha = 0.2) +
  theme_void()

py <- ggplot(samples, aes(x = x2)) +
  geom_density(linewidth = 1, color = "darkblue",
               fill = "blue", alpha = 0.2) +
  theme_void() + coord_flip()
```

```
px + plot_spacer() + pxy + py +
  plot_layout(ncol = 2, widths = c(4, 1), heights = c(1,
        4))
```



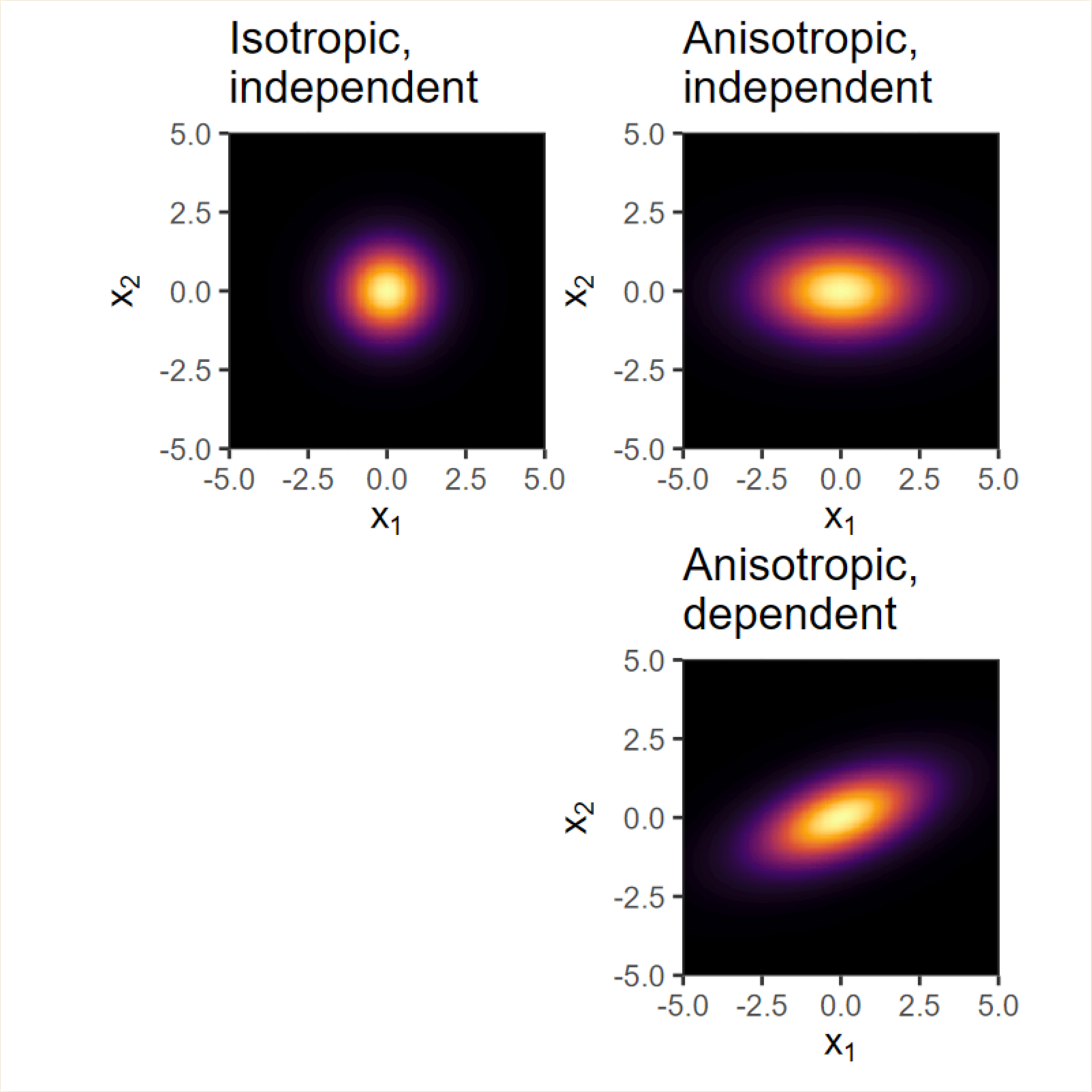
Describing MV Normal Distributions

- Isotropy: Does the distribution look the same if you rotate it about its center?



- Dependence: Does knowing one variable tell you about another?
  - In general: *If* variables are independent, *then* $\text{Cov} = 0$
  - For MVN distributions: *If* $\text{Cov} = 0$, *then* variables are independent

- All *isotropic* distributions are *independent*, but *anisotropic* distributions can be *dependent* or *independent*

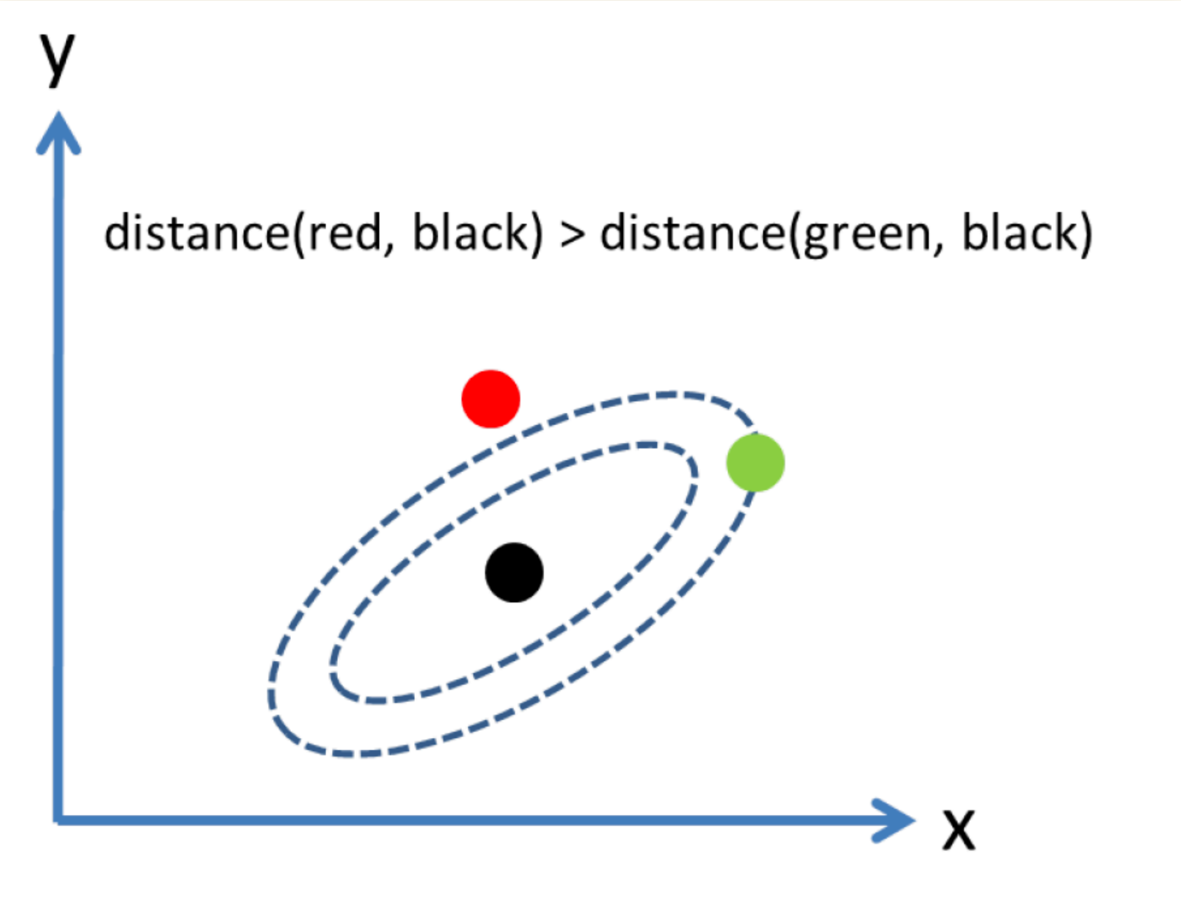


Measuring Distances

- How far is a point from the center of the distribution μ?
  - Euclidean distance is “as the crow flies”

$$d(x) = \sqrt{\sum_{i=1}^{p} (x_i - \mu_i)^2}$$

  - Mahalanobis distance accounts for anisotropies due to the covariance matrix



- *Isotropic, independent* distributions: Mahalanobis distance = Euclidean distance.
- *Anisotropic, independent* distributions: Mahalanobis distance is what the Euclidean distance would be, if you divided each coordinate by its standard-deviation:

$$d(x) = \sqrt{\sum_{i=1}^{p} \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2}$$

- *Anisotropic, dependent* distributions: Mahalanobis distance is complicated to calculate
  - Use `mahalanobis()` in R

```
x <- rmvnorm(5, mean = mu, sigma = sigma)
d <- mahalanobis(x, center = mu, cov = sigma)
```

Estimating Parameters of an MV Normal

- For each variable x_i, we estimate the mean just as for a regular normal:

$$\hat{\mu}_i = \textbf{mean}(x_i) = \frac{1}{n} \sum_{k=1}^{n} x_{k,i}$$

- Estimating covariance similar to estimating variance:

$$\hat{\Sigma}_{i,j} = \frac{1}{n-1} \sum_{k=1}^{n} \left(x_{k,i} - \hat{\mu}_i\right) \left(x_{k,j} - \hat{\mu}_j\right)$$

  - This estimate only works if $n \gg p$

- We can estimate the covariance using R:

```
set.seed(12345)
mu <- c(3, 5)
sigma <- matrix(c(3, 0.7, 0.7, 1), ncol =
        2)
x <- rmvnorm(100, mean = mu, sigma = sigma)
        |>
  as_tibble()
names(x) <- c("x1", "x2")
glimpse(x, width = 40)
```

```
## Rows: 100
## Columns: 2
## $ x1 <dbl> 4.188032, 2.694286, 3.56210…
## $ x2 <dbl> 5.837873, 4.533699, 3.40368…
```

```
cov(x)
```

```
##           x1        x2
## x1 3.0217064 0.7152096
## x2 0.7152096 1.2367679
```

Temperature vs. CO$_2$

- Covariance:

```
df  <- select(co2_t, co2, temp)
cov_co2_t <- cov(df)
kable(cov_co2_t, digits = 2)
```

	co2	temp
co2	1014.78	10.80
temp	10.80	0.14

- Pearson Correlation:

```
cor_co2_t_pearson <- cor(df, method = "pearson")
kable(cor_co2_t_pearson, digits = 2)
```

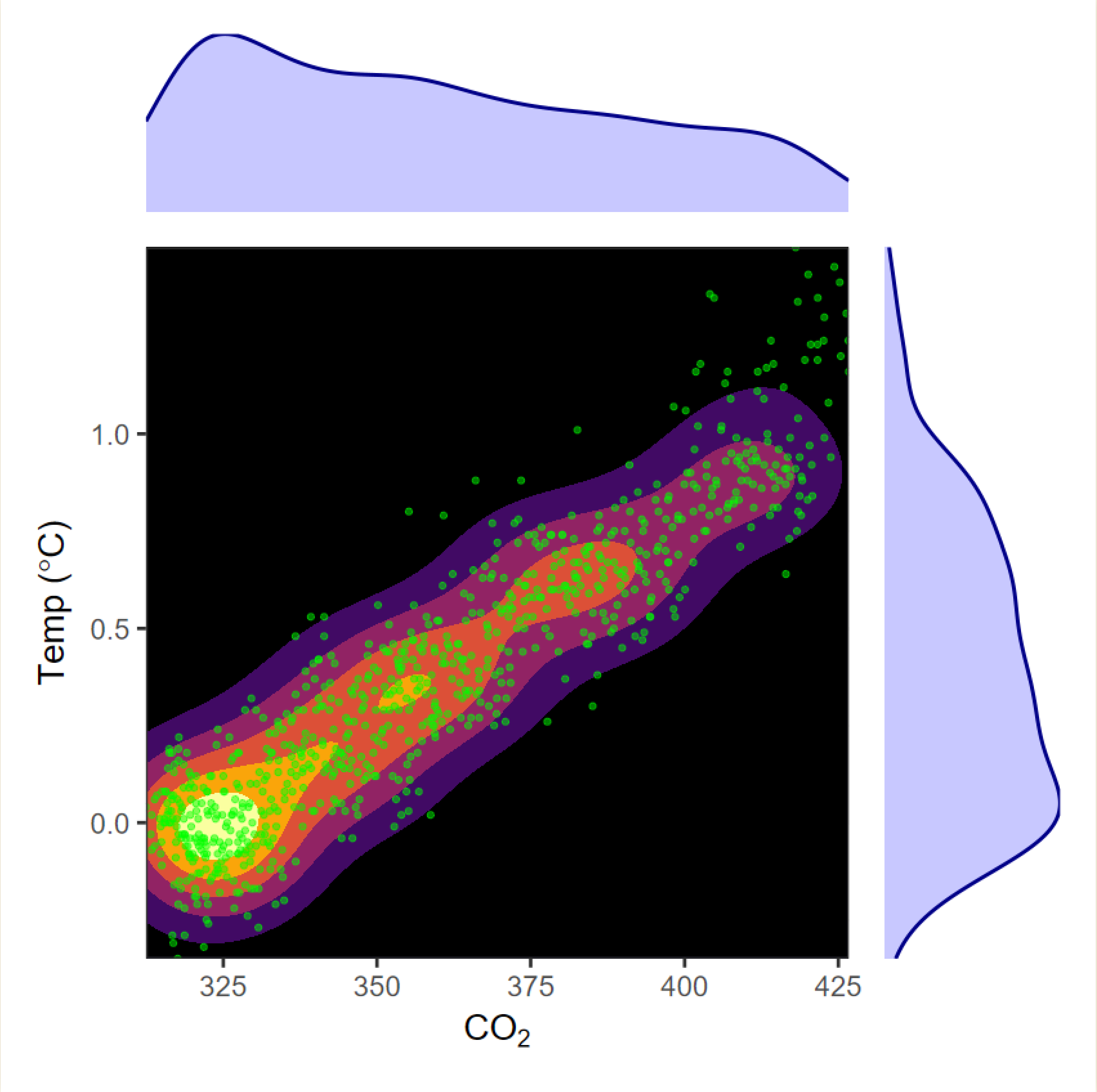
	co2	temp
co2	1.00	0.92
temp	0.92	1.00

- Spearman Correlation:

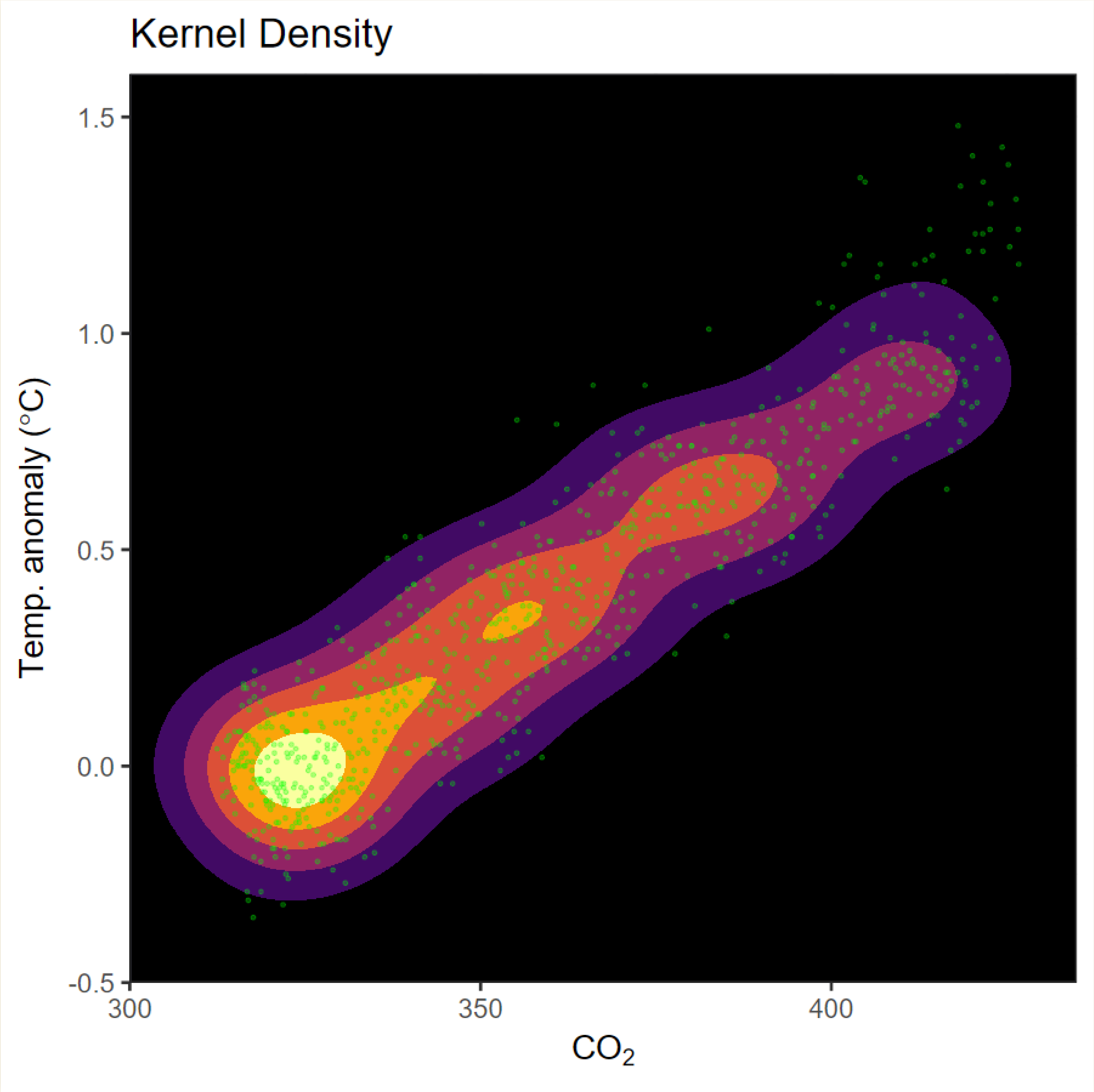
```
cor_co2_t_spearman <- cor(df, method = "spearman")
kable(cor_co2_t_spearman, digits = 2)
```

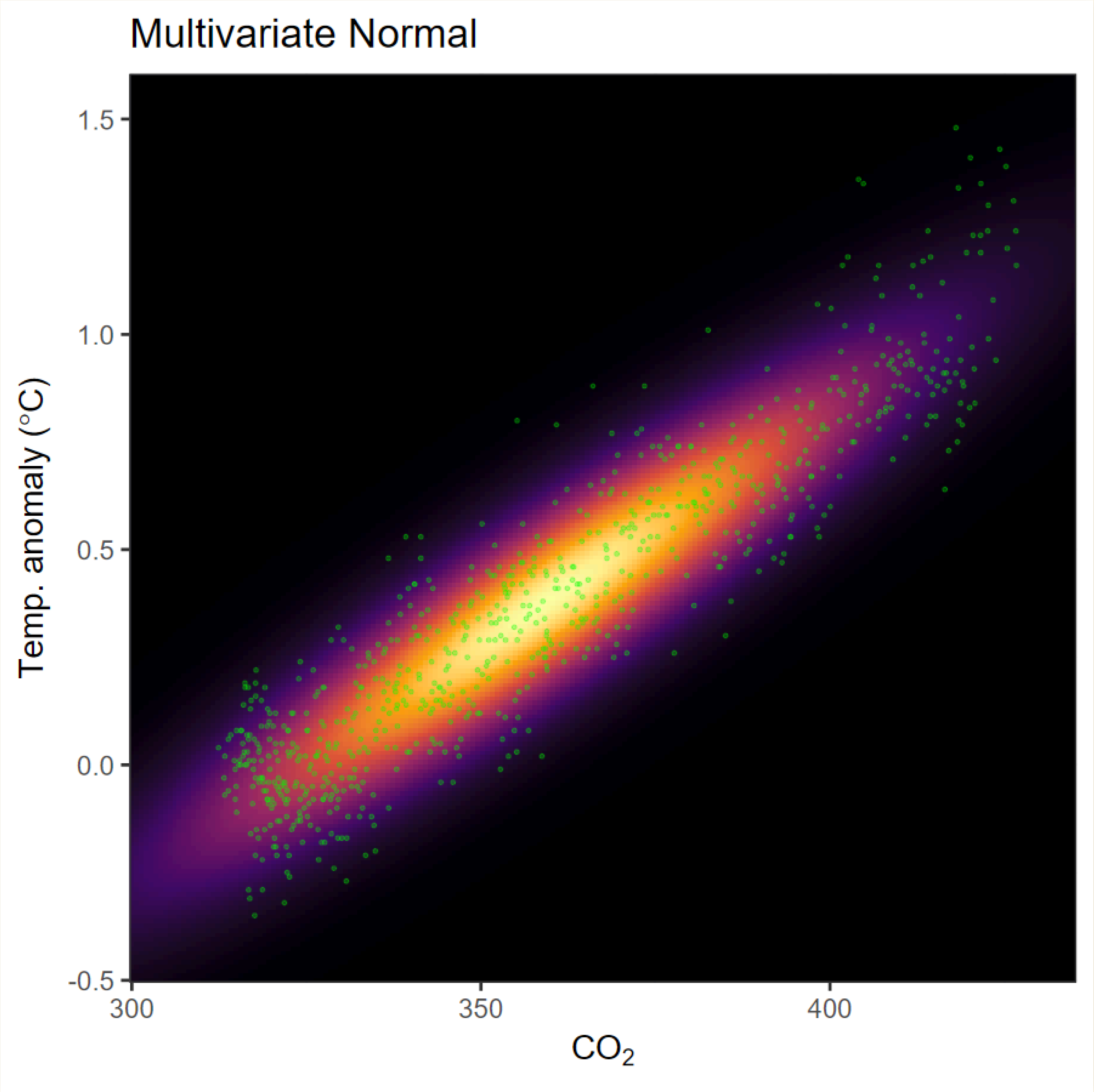
	co2	temp
co2	1.00	0.92
temp	0.92	1.00

```
px + plot_spacer() + pxy + py +
  plot_layout(ncol = 2, nrow = 2, widths = c(4, 1),
                        heights = c(1, 4))
```



Kernel Density vs. Multivariate Normal





Multivariate Central Limit Theorem

- Univariate Central Limit Theorem:
  - Consider a set of M experiments, in which each experiment takes n samples of a variable x, with $X \sim \mathcal{D}$, for some distribution $\mathcal{D}$
  - For each experiment, $\overline{x}$ is the mean of the n observations of x from that experiment
  - As $n \to \infty$, the distribution of $\overline{x}$ approaches a normal distribution with
    - Mean μ approaching $E(\mathcal{D})$
    - Variance σ approaching

$$\frac{1}{n}\mathrm{Var}(\mathcal{D})$$

- Multivariate central limit theorem:
  - M experiments, each taking n observations of p variables $x = (x_1, x_2, \ldots, x_p)$, with $X \sim \mathcal{M}$ for some multivariate distribution $\mathcal{M}$
  - As $n \to \infty$, the distribution of the mean $\overline{x} = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_p})$ approaches a multivariate normal distribution with

$$\mu = (\mu_1, \mu_2, \ldots, \mu_p) \to E(\mathcal{M})$$
$$\Sigma = (\Sigma_{i,j}) \to \frac{1}{n}\Sigma(\mathcal{M})$$

  - So regardless what multivariate distribution your data come from,
    - If n is large, $\overline{x} = (\overline{x_1}, \overline{x_2}, \ldots, \overline{x_p})$ will follow a multivariate normal.