Carson Slater

6376 Homework 8 Solutions

```
source("../helpers.R")
```

Preliminary Citations

For computing the kernel density estimator, Theo pointed out to me the `lims` argument in the `kde2d()` package. This enabled me to include more of the domain in the plot for 3(j). Otherwise, my limits for my plot would have been from $[-1, 1]$ on both the x and y axes.

Question 1

I have reread/peruised Trefethen and Bau lectures 24-28, focusing on lecture 27.

Question 2

(a)

The projection matrix $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Here $\mathbf{P} \in \mathbb{R}^{n \times n}$. It is idempotent ($\mathbf{P}^2 = \mathbf{P}$), and symmetric ($\mathbf{P} = \mathbf{P}'$).

(b)

Here, we have that

$$\mathbf{X}\beta = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

This yields

$$\beta = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

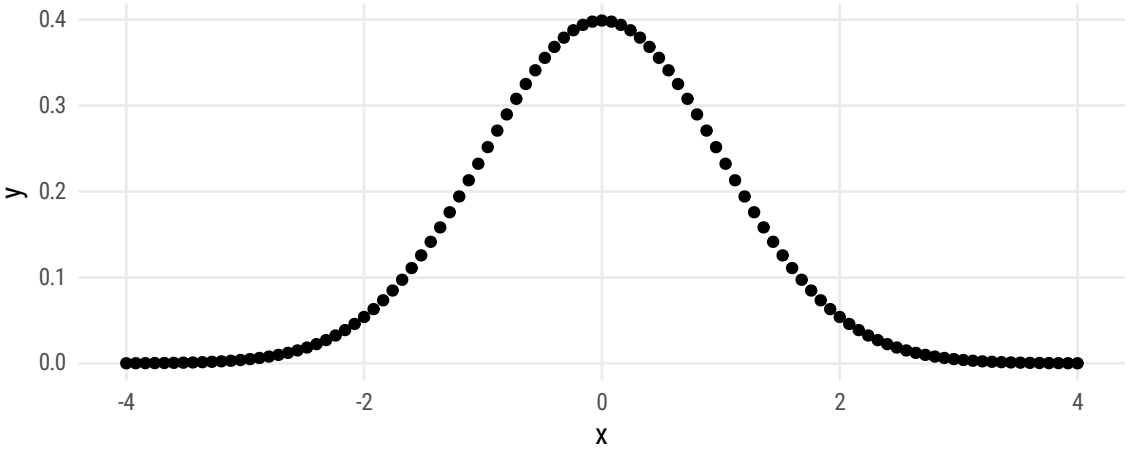
(c)

$\mathbf{P}\mathbf{y}$ are the fitted values of the model, and $(\mathbf{I}_n - \mathbf{P})\mathbf{y}$ are the residuals of the model.

Question 3

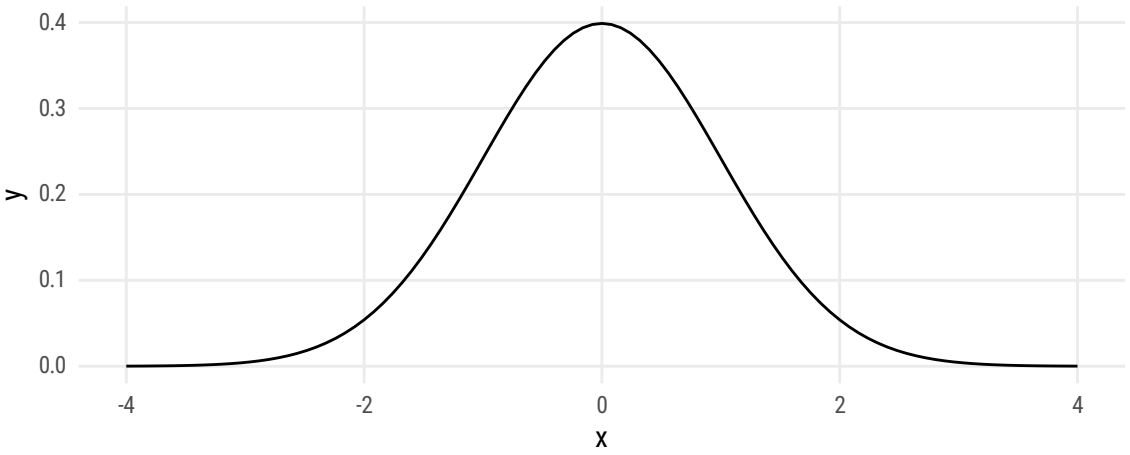
(a)

```
# Create the points
x <- seq(-4, 4, length.out = 101)
y <- dnorm(x)
df <- tibble(x, y)

df |> ggplot(aes(x, y)) +
  geom_point()
```



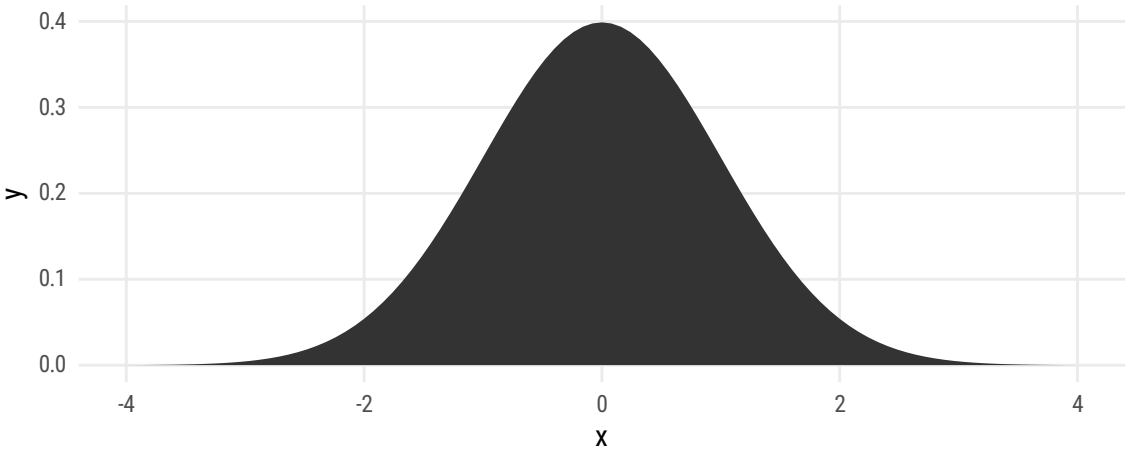
(b)

```
df |> ggplot(aes(x,y)) +
  geom_line()
```



(c)

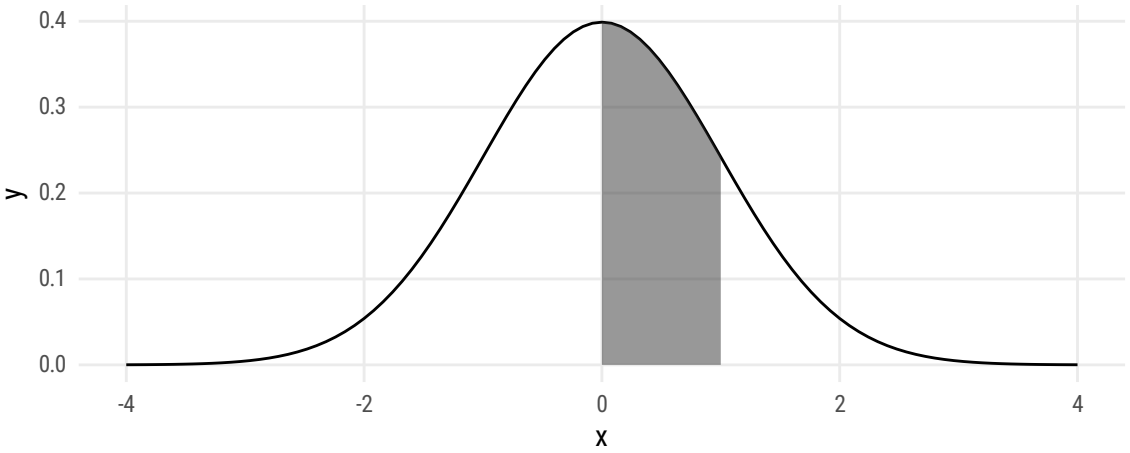
```
df |> ggplot(aes(x,y)) +
  geom_polygon()
```



(d)

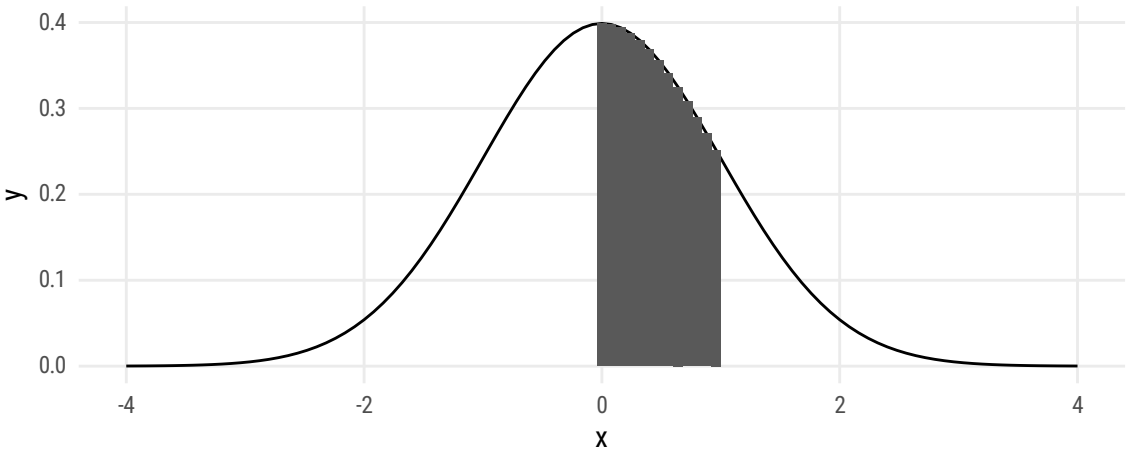
```
df2 <- subset(df, x >= 0 & x <= 1)
df2 <- rbind(c(0,0), df2, c(1, dnorm(1)), c(1,0))

ggplot() +
  geom_line(data = df, aes(x,y)) +
```

```
  geom_polygon(data = df2, aes(x, y), alpha = 0.5)
```



(e)

```
first <- function(x) x[1]
diff <- df$x |> diff() |> first()
df2 <- subset(df, x >= 0 & x <= 1) # rebuild df2 for not geom_poly()
df3 <- transform(df2, 'xmin' = x - diff/2,
                 'xmax' = x + diff/2,
                 'ymin' = 0,
                 'ymax' = y)

# for the sake of the polygon geom in the prior
# exercise (3d), I had to add row 15 in df2 to make the
# plot, but for this one needed to remove it
df3 <- df3[-15]

ggplot() +
  geom_line(data = df, aes(x,y)) +
  geom_rect(data = df3, aes(xmin = xmin,
                            xmax = xmax,
                            ymin = ymin,
                            ymax = ymax))
```



(f)

```
df3 <- transform(df3, 'area' = diff*y)

# Riemann sum - got help from Theo with aggregate()
aggregate(area ~ 1, data = df3, FUN = sum)
```

```
     area
1 0.357367
```

(g)

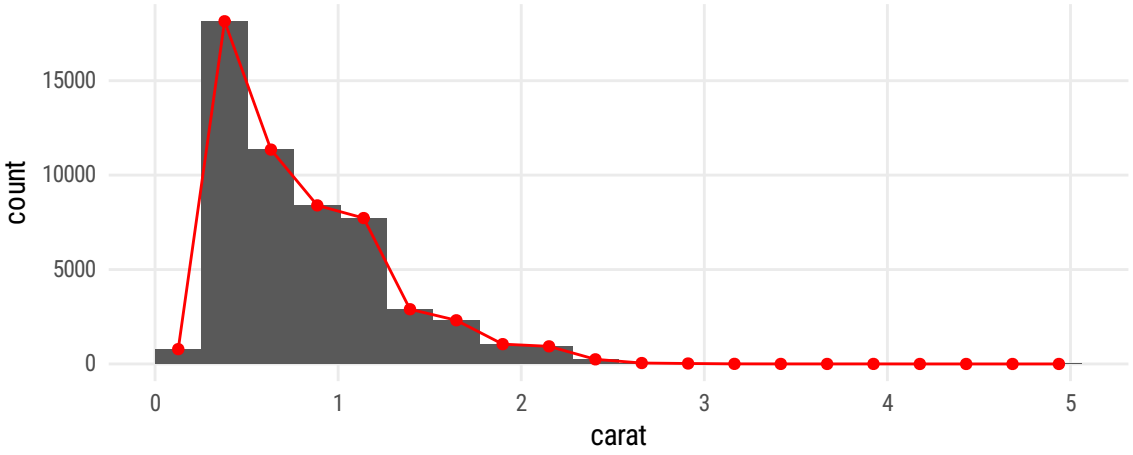
```
# Area under the curve
pnorm(1) - pnorm(0)
```

```
[1] 0.3413447
```

This Riemann sum gets within two percent of the true probability. Increasing the number of rectangles from 15 should generate an approximation numerically closer to the true probability.

(h)

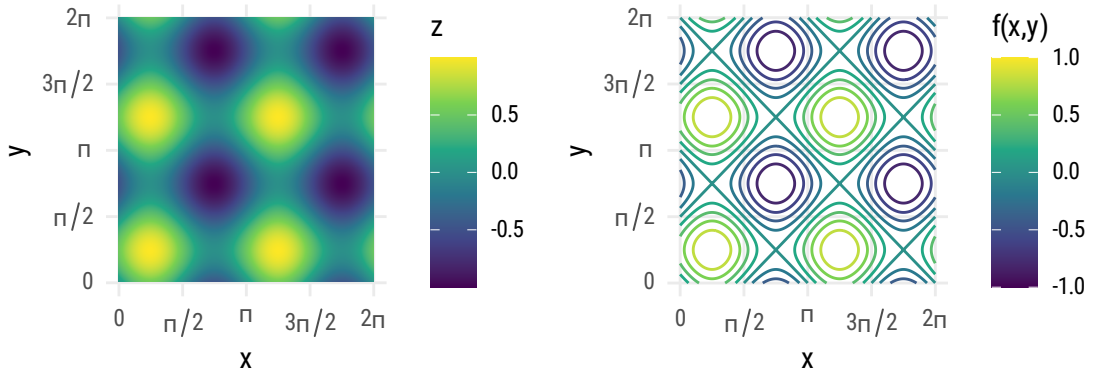
```
ggplot(data = diamonds, aes(carat)) +
  geom_histogram(bins = 20, boundary = 0) +
  geom_point(aes(y = after_stat(count)),
             stat = "bin",
             bins = 20,
             color = "red",
             boundary = 0) +
  geom_line(aes(y = after_stat(count)),
            stat = "bin",
            bins = 20,
            color = "red",
            boundary = 0)
```



(i)

```
x <- seq(0, 2*pi, by = 0.01)
grid <- expand.grid(x = x, y = x)
grid <- transform(grid, "z" = (sin(2*x) + sin(2*y))/2)

pi_axes <- c(0, expression(pi/2), expression(pi), expression(3*pi/2),
             expression(2*pi))

p1 <- grid |> ggplot(aes(x, y)) +
  geom_raster(aes(fill = z)) +
```

```
  scale_color_continuous(type = "viridis") +
  scale_x_continuous(breaks = seq(0, 2*pi, by = pi/2), labels = pi_axes) +
  scale_y_continuous(breaks = seq(0, 2*pi, by = pi/2), labels = pi_axes)

p2 <- grid |> ggplot(aes(x, y)) +
  geom_contour(aes(z = z, color = after_stat(level))) +
  scale_color_continuous(limits = c(-1, 1), type = "viridis") +
  labs(color = "f(x,y)") +
  scale_x_continuous(breaks = seq(0, 2*pi, by = pi/2), labels = pi_axes) +
  scale_y_continuous(breaks = seq(0, 2*pi, by = pi/2), labels = pi_axes)

p1 + p2
```



(j)

```
df <- data.frame(x = -1:1, y = c(-1, 1, -1))

# compute KDE
dens_df <-MASS::kde2d(df$x,
                      df$y,
                      n = 1500,
                      lims = c(c(-2,2),
                               c(-2,2))
                      )

dens_df2 <- expand.grid(x = dens_df$x, y = dens_df$y) |>
  transform(density = as.vector(dens_df$z))
```

```
dens_df2 |> ggplot(aes(x, y)) +
  geom_raster(aes(fill = density)) +
  geom_point(data = df, aes(x, y), color = "red", size = 2)
```