

HPAM 7660 Data Assignment 4

Tyler LaBattes

March 4, 2024

The 3 essential components of data visualization are data, geom, and aes. Data is the dataset containing the variables that will be used. Geom is the geometric form you will use to display the variables. Aes is for the aesthetic attributes of the graph such as x/y, size, and shape. These 3 components are used to specify the way variables will be used to form a visualization.

Below I am loading the packages I will use for this assignment.

```
load("~/data_4.RData")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

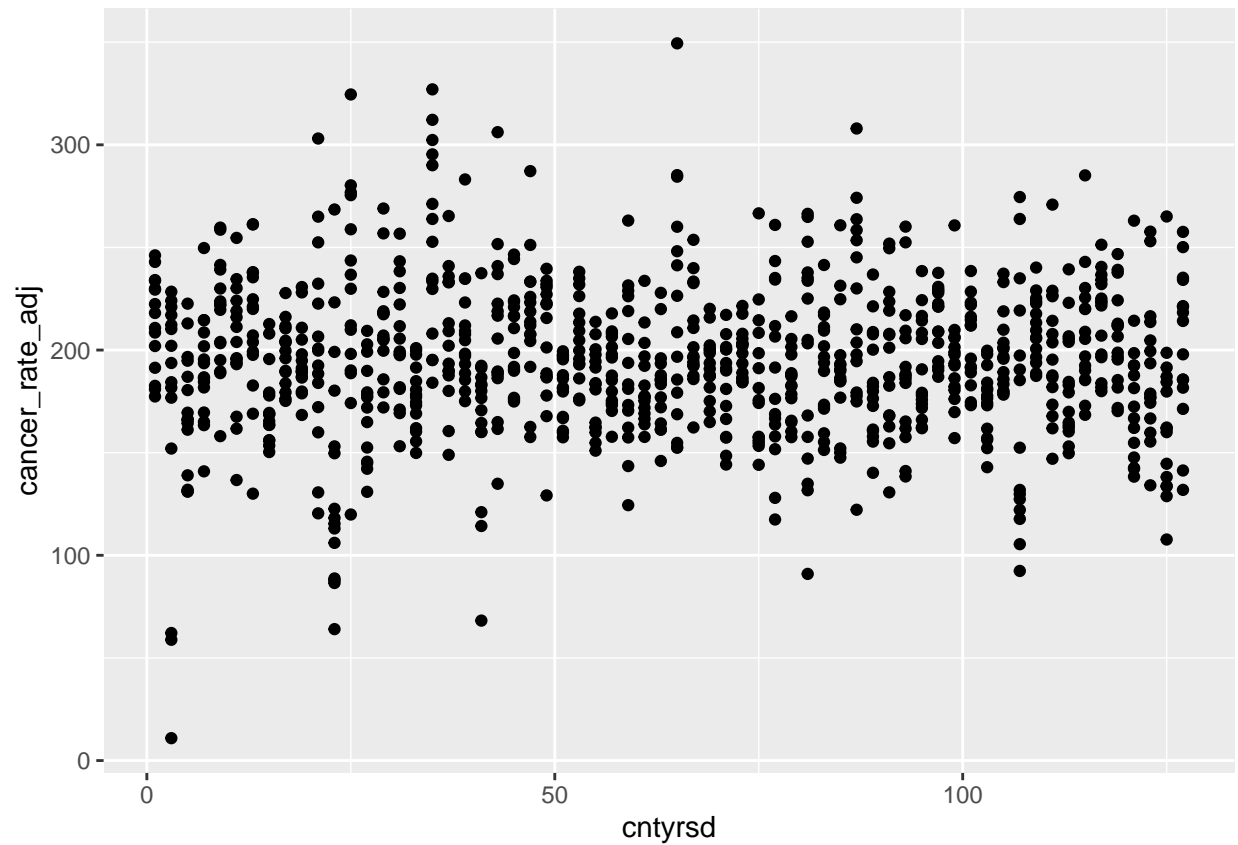
## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(knitr)
library(ggplot2)
```

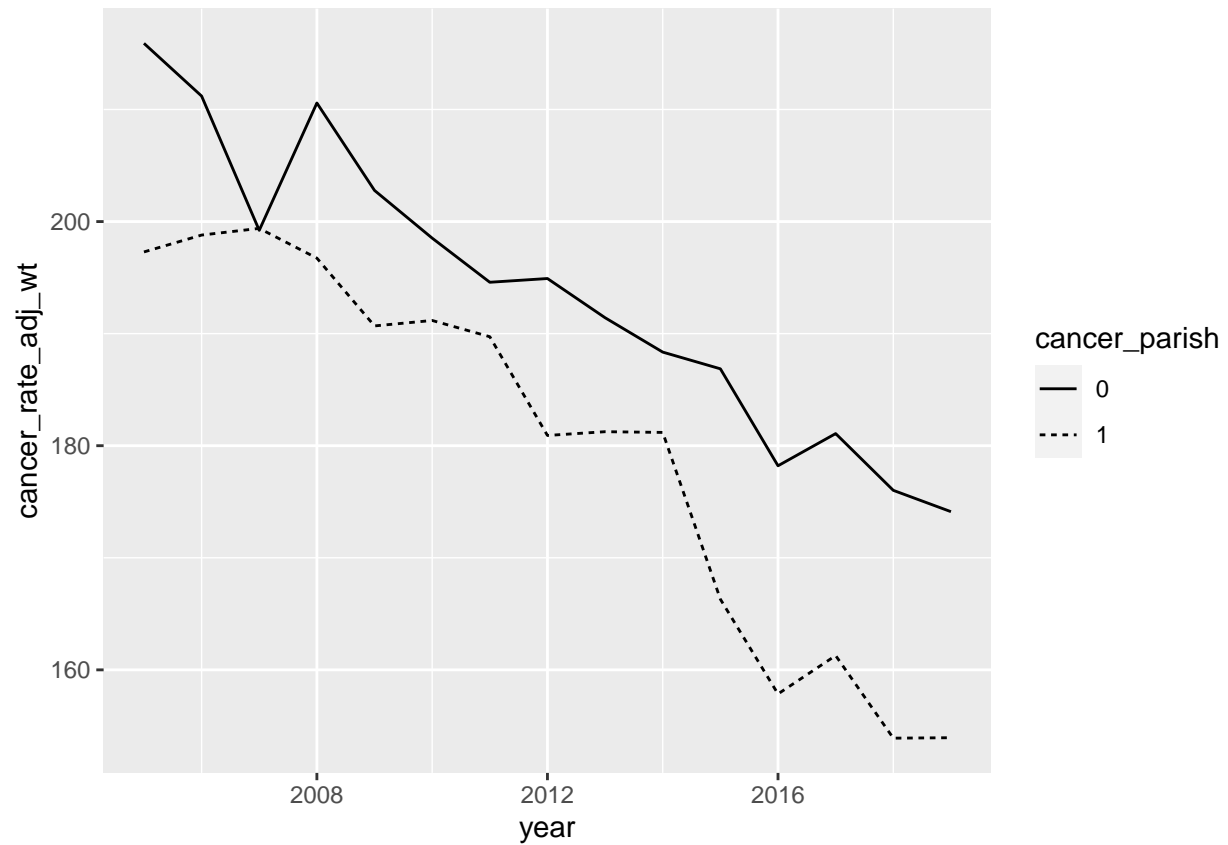
In the code below I created a scatterplot for cancer rates by county in the year 2019.

```
filter(year == 2019) %>%
ggplot(data = parish_rates, mapping = aes(x = cntyrscd, y = cancer_rate_adj)) +
  geom_point()
```



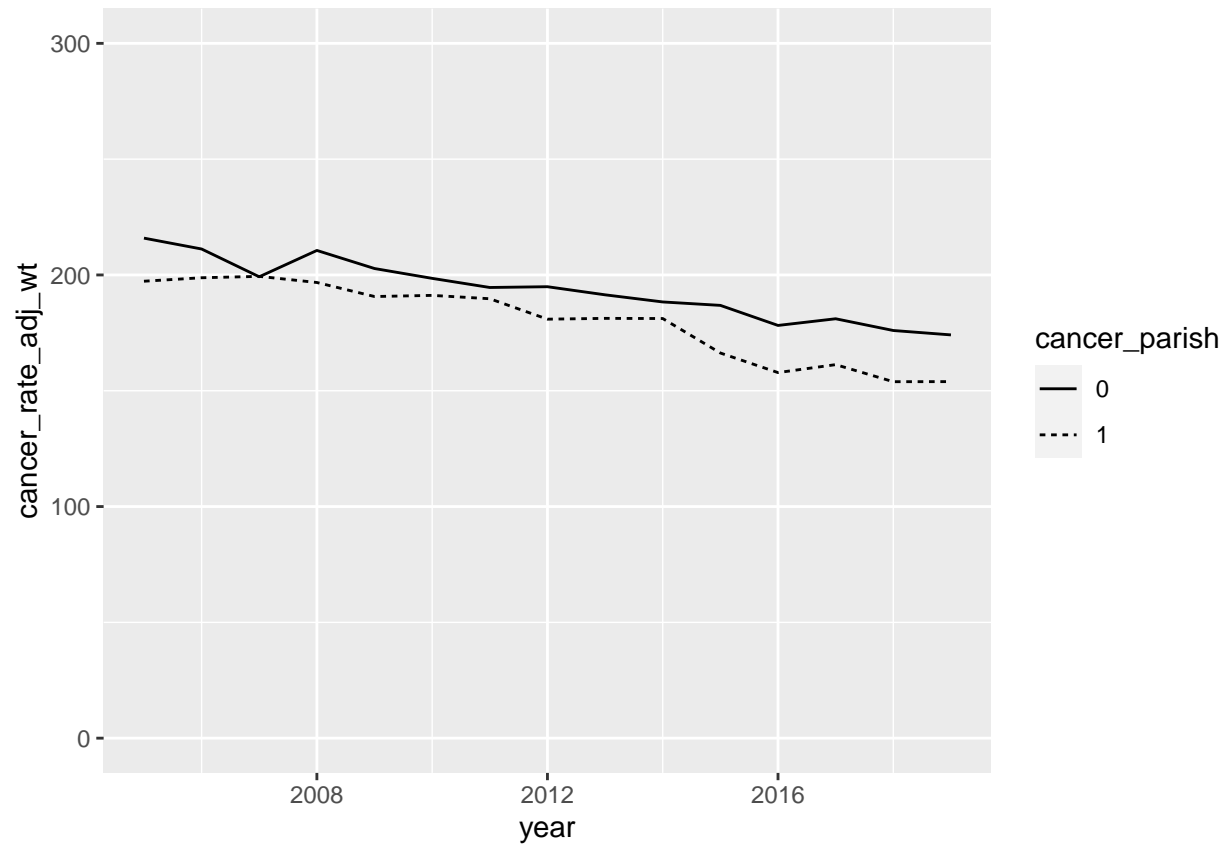
In the code below, I created a linegraph for the cancer rates per year from 2005 to 2019. I group the parishes by cancer alley parishes and non-cancer alley parishes to show the difference in cancer rates between the two groups.

```
cancer_alley_rates$cancer_parish <- factor(cancer_alley_rates$cancer_parish)
ggplot(data = cancer_alley_rates, mapping = aes(x = year, y = cancer_rate_adj_wt, linetype = cancer_parish)) +
  geom_line()
```



I added a y axis value range of 0 to 300 to the graph in the code below.

```
ggplot(data = cancer_alley_rates, mapping = aes(x = year, y = cancer_rate_adj_wt, linetype = cancer_parish))  
  geom_line() +  
  scale_y_continuous(limits = c(0, 300))
```



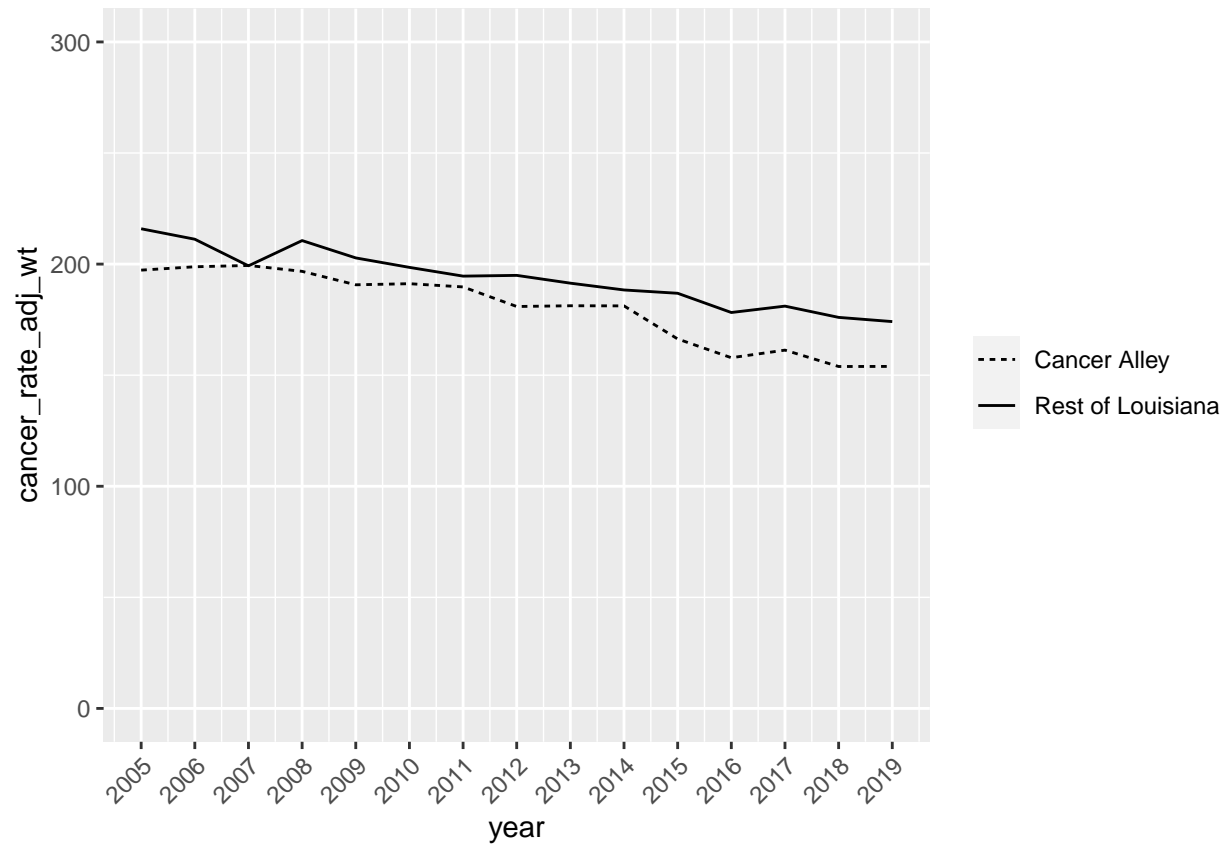
Below I added some changes to the x axis variable by showing each year in the data and also edited the format of the years so that they would not overlap.

```
ggplot(data = cancer_alley_rates, mapping = aes(x = year, y = cancer_rate_adj_wt, linetype = cancer_parish)) +
  geom_line() +
  scale_y_continuous(limits = c(0, 300)) +
  scale_x_continuous(breaks = seq(2005, 2019, by = 1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Below I added labels to the legend to specify what each line represents.

```
ggplot(data = cancer_alley_rates, mapping = aes(x = year, y = cancer_rate_adj_wt, linetype = cancer_parish)) +
  geom_line() +
  scale_y_continuous(limits = c(0, 300)) +
  scale_x_continuous(breaks = seq(2005, 2019, by = 1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_linetype_discrete(name = NULL, labels = c("Rest of Louisiana", "Cancer Alley"), guide = guide_legend())
```



Finally, I am adding a title and y axis label to the linegraph.

```
ggplot(data = cancer_alley_rates, mapping = aes(x = year, y = cancer_rate_adj_wt, linetype = cancer_pari
  geom_line() +
  scale_y_continuous(limits = c(0, 300)) +
  scale_x_continuous(breaks = seq(2005, 2019, by = 1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_linetype_discrete(name = NULL, labels = c("Rest of Louisiana", "Cancer Alley"), guide = guide_le
  labs(title = "Cancer Mortality Rate Comparison", y = "Cancer Deaths per 100,000 Residents", x = NULL)
```

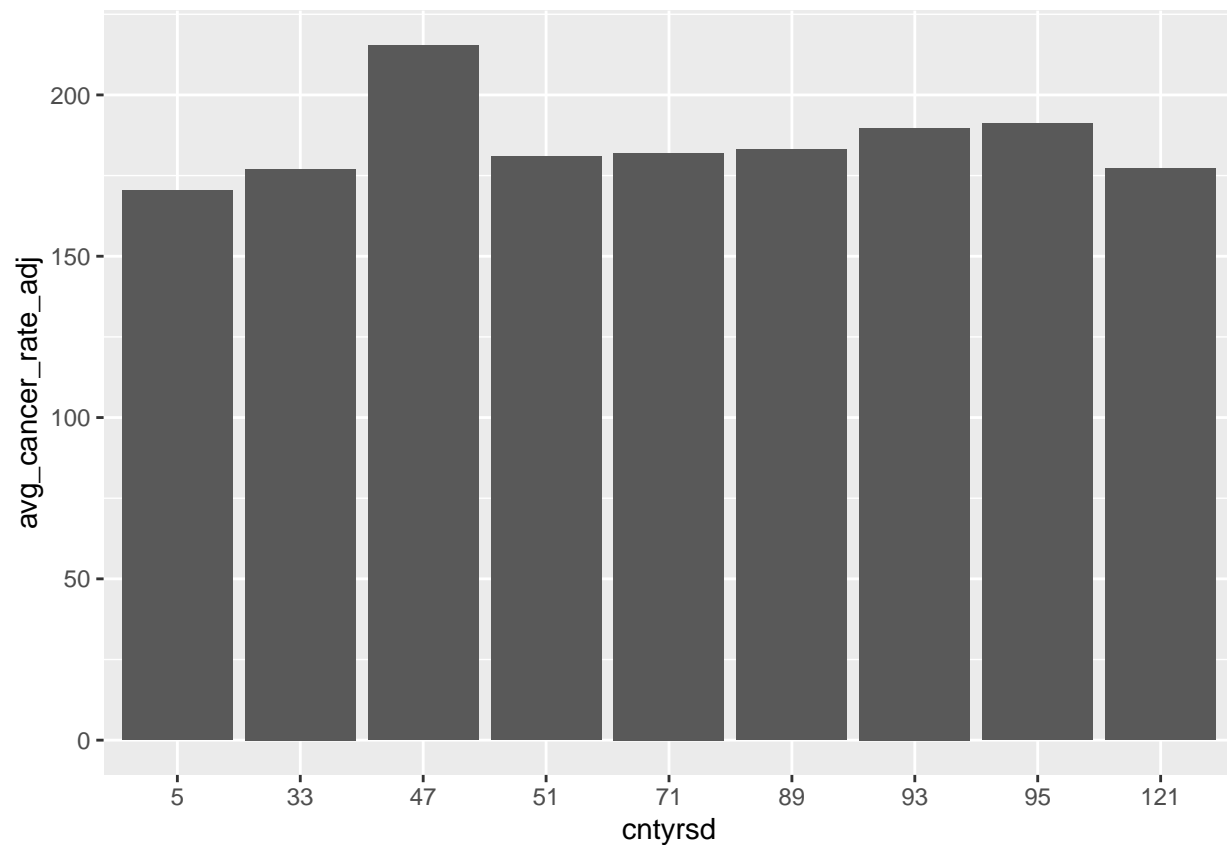


Below, I am creating a bar graph for the average cancer rate from 2005-2019 for each parish in cancer alley. First, I set the “county” variable at factor so that it would not be seen as a continuous variable. Next, I created a new data frame that include only cancer alley parishes, and I also calculated the cancer rates for the years into one average for each parish. Then, I created a bargraph using only the cancer alley parishes and the average cancer rates from 2005-2019 for those parishes.

```
parish_rates$cntyrstd <- factor(parish_rates$cntyrstd)
parish_rates_ca <- filter(parish_rates, cancer_parish == 1)
parish_rates_ca <- parish_rates_ca %>%
  summarize(avg_cancer_rate_adj = mean(cancer_rate_adj, na.rm = TRUE))
```

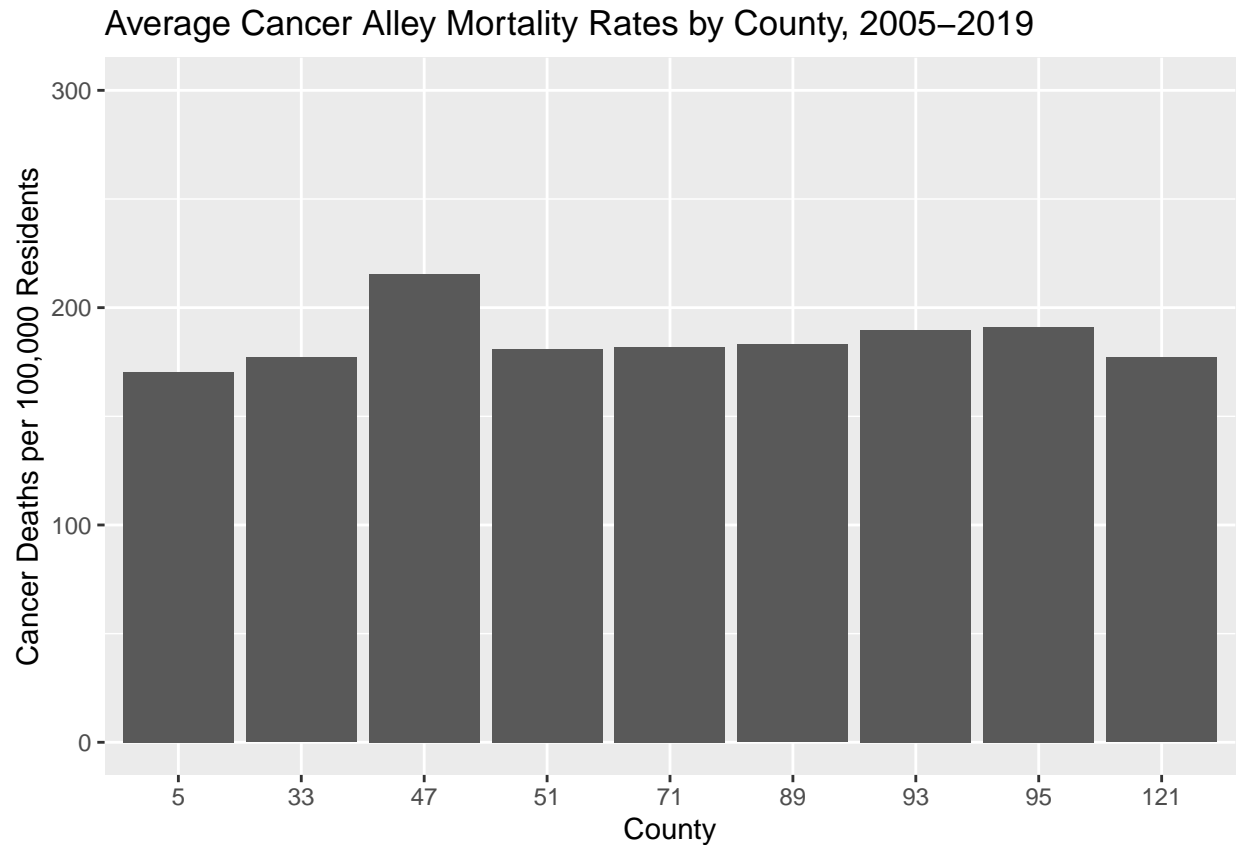
```
## 'summarise()' has grouped output by 'cntyrstd'. You can override using the
## '.groups' argument.
```

```
ggplot(data = parish_rates_ca, mapping = aes(x = cntyrstd, y = avg_cancer_rate_adj)) +
  geom_col()
```



Below I added a y axis value scale of 0-300, and I also added labels for the x axis, y axis, and title.

```
ggplot(data = parish_rates_ca, mapping = aes(x = cntyrstd, y = avg_cancer_rate_adj)) +
  geom_col() +
  scale_y_continuous(limits = c(0, 300)) +
  labs(title = "Average Cancer Alley Mortality Rates by County, 2005-2019", y = "Cancer Deaths per 100,
```

Finally, I added x axis labels for each county to replace the county codes that were shown.

```
ggplot(data = parish_rates_ca, mapping = aes(x = cntyrscd, y = avg_cancer_rate_adj)) +
  geom_col() +
  scale_y_continuous(limits = c(0, 300)) +
  labs(title = "Average Cancer Alley Mortality Rates by County, 2005-2019", y = "Cancer Deaths per 100,000 Residents") +
  scale_x_discrete(labels = c("5" = "Acension",
                              "33" = "East Baton Rouge",
                              "47" = "Iberville",
                              "51" = "Jefferson",
                              "71" = "Orleans",
                              "89" = "St. Charles",
                              "93" = "St. James",
                              "95" = "St. John the Baptist",
                              "121" = "West Baton Rouge")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Average Cancer Alley Mortality Rates by County, 2005–2019

