# Create a blank visualization

```r
votes <- read.csv("votes.csv")
ggplot()
```
# Supply data

```r
votes <- read.csv("votes.csv")
ggplot(votes)
```
# Add first geometry

votes <- read.csv("votes.csv")

ggplot(votes) + geom_col()
# Add x and y aesthetics

```r
votes <- read.csv("votes.csv")

ggplot(votes, aes(x = candidate, y = votes)) +
  geom_col()
```
# Adjust y scale

```r
votes <- read.csv("votes.csv")

ggplot(votes, aes(x = candidate, y = votes)) +
  geom_col() +
  scale_y_continuous(limits = c(0, 250))
```
# Add labels

votes <- read.csv("votes.csv")

ggplot(votes, aes(x = candidate, y = votes)) +
  geom_col() +
  scale_y_continuous(limits = c(0, 250)) +
  labs(
    x = "Candidate",
    y = "Votes",
    title = "Election Results"
)

# Add fill aesthetic mapping for geom_col

```r
votes <- read.csv("votes.csv")

ggplot(votes, aes(x = candidate, y = votes)) +
  geom_col(aes(fill = candidate)) +
  scale_y_continuous(limits = c(0, 250)) +
  labs(
    x = "Candidate",
    y = "Votes",
    title = "Election Results"
  )
```
# Use viridis scale to design for color blindness

votes <- read.csv("votes.csv")

ggplot(votes, aes(x = candidate, y = votes)) +
  geom_col(aes(fill = candidate)) +
  scale_fill_viridis_d("Candidate") +
  scale_y_continuous(limits = c(0, 250)) +
  labs(
    x = "Candidate",
    y = "Votes",
    title = "Election Results"
  )
# Adjust ggplot theme

```r
votes <- read.csv("votes.csv")

ggplot(votes, aes(x = candidate, y = votes)) +
  geom_col(aes(fill = candidate)) +
  scale_fill_viridis_d("Candidate") +
  scale_y_continuous(limits = c(0, 250)) +
  labs(
    x = "Candidate",
    y = "Votes",
    title = "Election Results"
  ) +
  theme_classic()
```
# Save file

```r
votes <- read.csv("votes.csv")
p <- ggplot(votes, aes(x = candidate, y = votes)) + geom_col(aes(fill = candidate)) + scale_fill_viridis_d("Candidate") + scale_y_continuous(limits = c(0, 250)) + labs(x = "Candidate", y = "Votes", title = "Election Results") + theme_classic()

ggsave("votes.png", plot = p, width = 1200, height = 900, units = "px")
```

# Introduce geom_point

```r
load("candy.RData")

# Introduce geom_point

ggplot()
  candy,
  aes(x = price_percentile, y = sugar_percentile)
)
  geom_point()
```
# Add labels and theme

```r
load("candy.RData")

ggplot(candy,
aes(x = price_percentile, y = sugar_percentile)
) +
  geom_point() +
  labs(
    x = "Price",
    y = "Sugar",
    title = "Price and Sugar"
  ) +
  theme_classic()
```
# Introduce `geom_jitter`

ggplot(candy, aes(x = price_percentile, y = sugar_percentile)) +
  geom_jitter() +
  labs(x = "Price",
       y = "Sugar",
       title = "Price and Sugar") +
  theme_classic()
# Introduce size and color aesthetic

ggplot(candy, aes(x = price_percentile, y = sugar_percentile)) + geom_jitter(color = "darkorchid", size = 2) + labs(x = "Price", y = "Sugar", title = "Price and Sugar") + theme_classic()
# Introduce point shape and fill color

ggplot(candy, aes(x = price_percentile, y = sugar_percentile)) +
  geom_jitter(color = "darkorchid", fill = "orchid", shape = 21, size = 2) +
  labs(x = "Price", y = "Sugar", title = "Price and Sugar") +
  theme_classic()
# Visualize with geom_point

```r
load("anita.RData")

ggplot(anita, aes(x = timestamp, y = wind)) + geom_point()
```
# Introduce geom_line

```r
load("anita.RData")
ggplot(anita, aes(x = timestamp, y = wind)) +
  geom_line()
```
# Combine geom_line and geom_point

load("anita.RData")

ggplot(anita, aes(x = timestamp, y = wind)) +
  geom_line() +
  geom_point(color = "deepskyblue4")
# Experiment with geom_line and geom_point aesthetics

```r
load("anita.RData")

ggplot(anita, aes(x = timestamp, y = wind)) +
  geom_line(
    linetype = 1,
    linewidth = 0.5
  ) +
  geom_point(
    color = "deepskyblue4",
    size = 2
  )
```

# Add labels and adjust theme

```r
load("anita.RData")

ggplot(anita, aes(x = timestamp, y = wind)) +
  geom_line(
    linetype = 1,
    linewidth = 0.5
  ) +
  geom_point(
    color = "deepskyblue4",
    size = 2
  ) +
  labs(
    y = "Wind Speed (Knots)",
    x = "Date",
    title = "Hurricane Anita"
  ) +
  theme_classic()
```
# Add horizontal line to demarcate hurricane status

```r
load("anita.RData")

ggplot(anita, aes(x = timestamp, y = wind)) +
  geom_line(
    linetype = 1,
    linewidth = 0.5
  ) +
  geom_point(
    color = "deepskyblue4",
    size = 2
  ) +
  geom_hline(
    linetype = 3,
    yintercept = 64
  ) +
  labs(
    y = "Wind Speed (Knots)",
    x = "Date",
    title = "Hurricane Anita"
  ) +
  theme_classic()
```