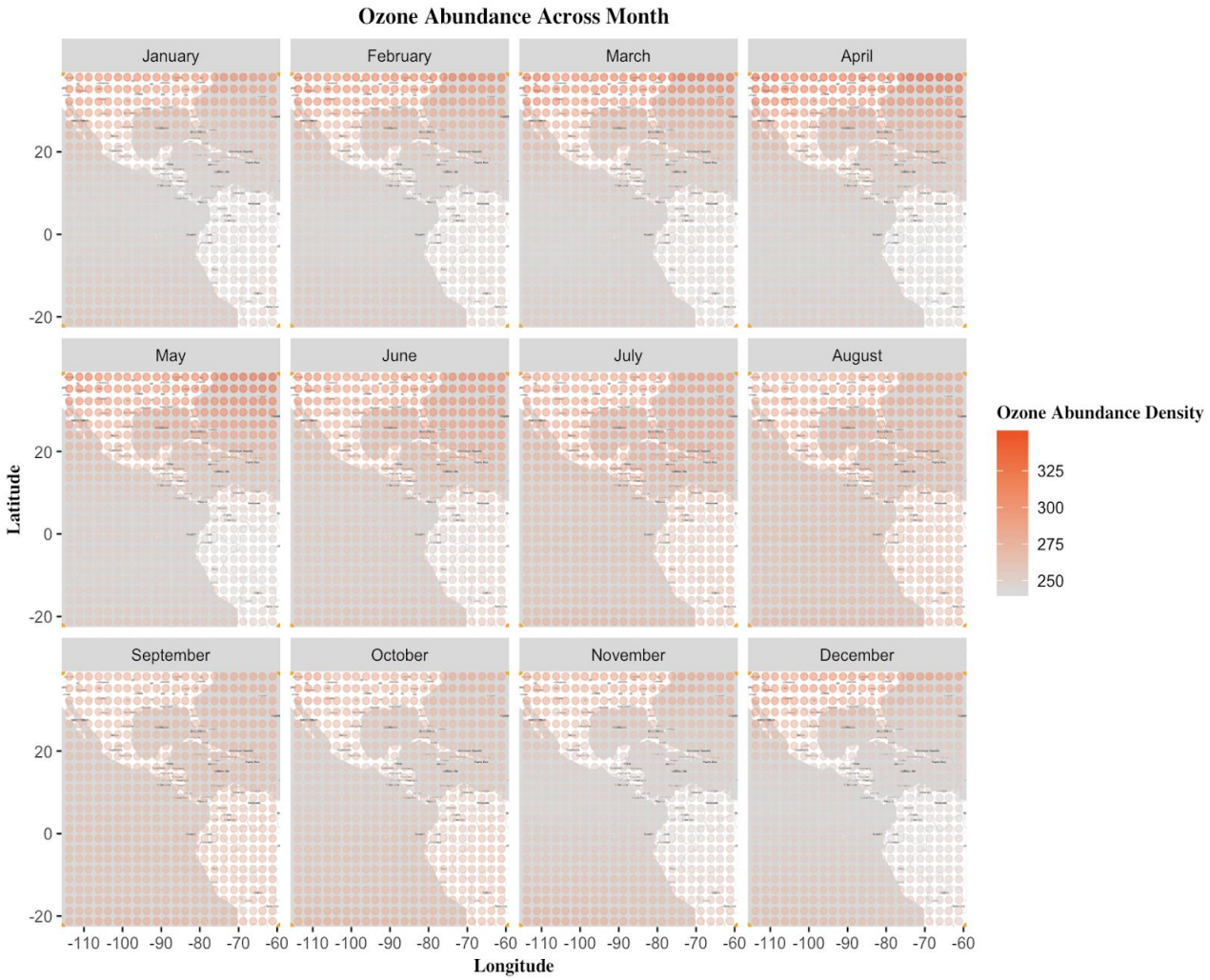
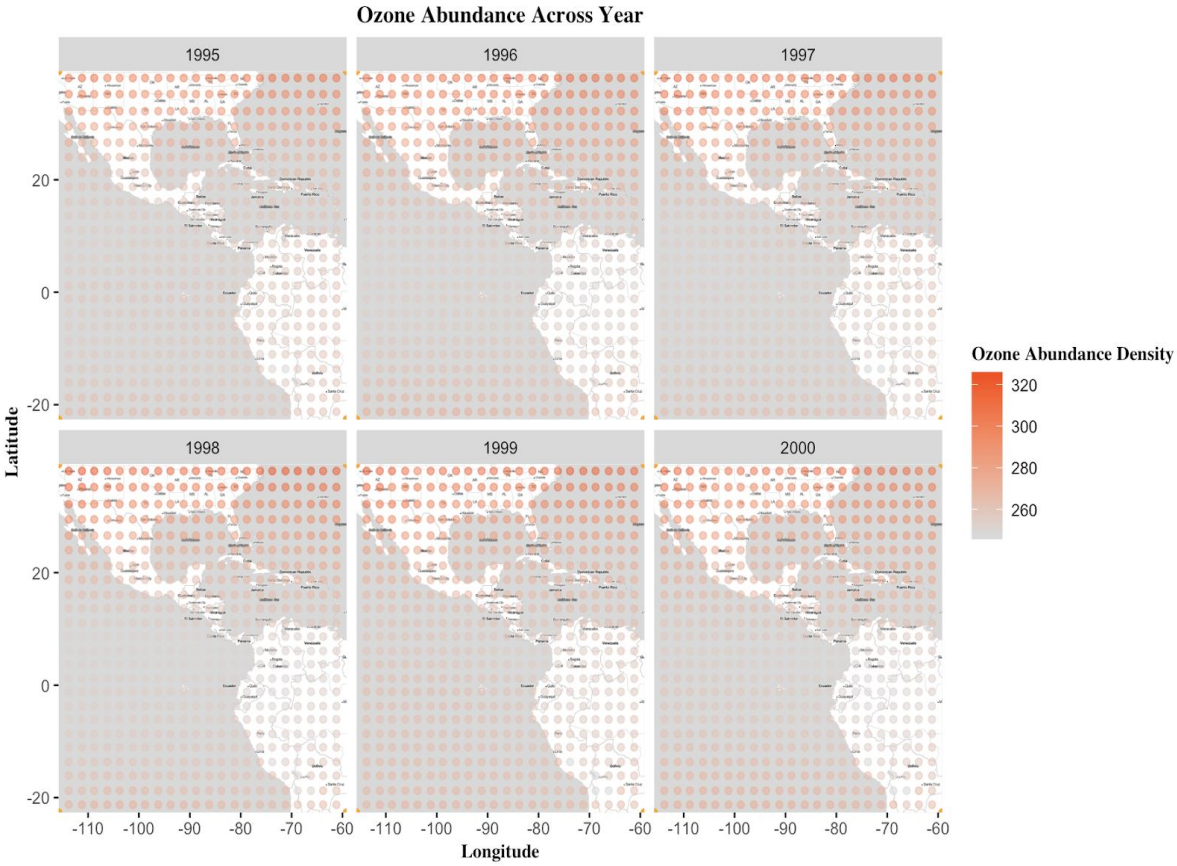
Jisu Kim (jk5nc), Timothy Kim (tsk8va), Olivia Ryu (hr2ad)
STAT 3280-001
Professor Tianxi Li
13 March 2020

Homework 2

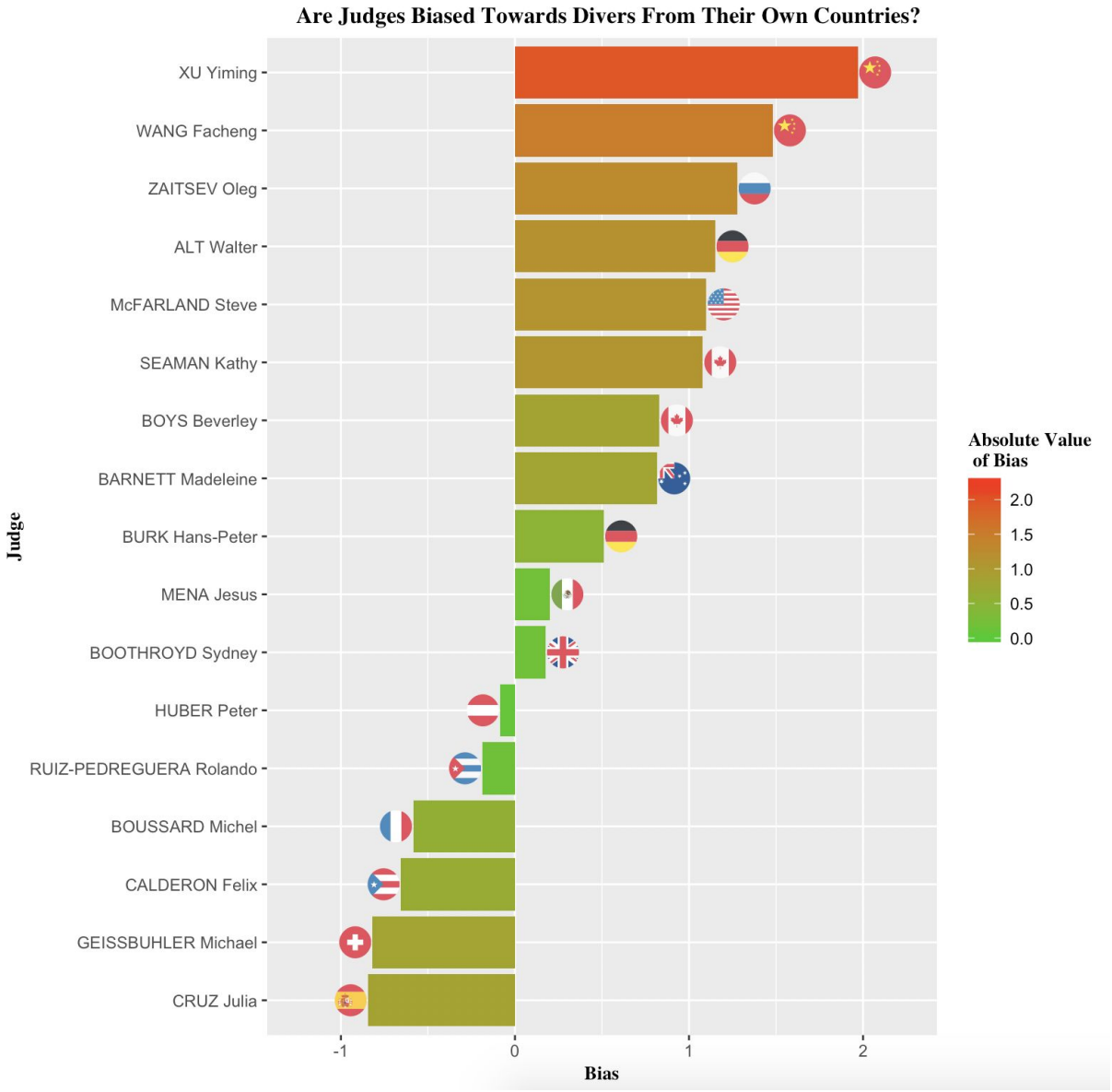
Problem 1.1



Problem 1.2


Ozone Abundance Across Year
1995
1996
1997
1998
1999
2000
Latitude
20
0
-20
-110
-100
-90
-80
-70
-60
Longitude
Ozone Abundance Density
320
300
280
260

Problem 2



NOTE: Bias was computed for each judge by subtracting the average of scores given to foreign divers from the average of scores given to divers of the same nationality. Judges have been removed for lack of sufficient data; some were missing scores given to home country divers while others were missing scores given to foreign country divers.

CONCLUSION: Due to the considerable distance some of the judges deviate from the ideal non-biased value of zero, we concluded that some judges **are** biased towards divers of the same nationality. Specifically, the judges from China (XU Yiming, WANG Facheng) and Russia (ZAITSEV Oleg) seemed to be the "most biased," while the Danish judge, HUBER Peter, was the "least biased." We were interested to notice some judges actually have a negative bias, meaning they are more critical of divers from their own country.

Appendix 1.1

```
# Problem 1
# We will be analyzing Ozone Abundance

# Problem 1.1 - All twelve monthly averages for ozone measurements

# Create a new df with lat, long, ozone data
example <- scan("../Files/cloudhigh1.txt",what="",sep="\t")
long <- trimws(example[6])
long <- strsplit(long, " ")
long <- long[[1]]
long <- long[long != ""]

lat <- vector()
for(i in 8:31){
  lat<- c(lat, substr(example[i], 2, 5))
}
for(i in 12:19){
  lat[i] <- substr(lat[i],1, nchar(lat[i])-1)
}
for(i in 16:24){
  lat[i] <- paste0("-",lat[i])
}
for(i in 1:24){
  long[i] <- substr(long[i],1, nchar(long[i])-1)
  long[i] <- paste0("-",long[i])
}

long <- as.numeric(long)
lat <- as.numeric(lat)

longitudes <- vector()
latitudes <- vector()
for(i in 1:24){
  for(j in 1:24){
    longitudes <- c(longitudes, long[i])
    latitudes<- c(latitudes, lat[j])
  }
}

jan <- data.frame(latitudes, longitudes, "January",
(GridTimeSeries[[01]]$X$ozone + GridTimeSeries[[13]]$X$ozone +
GridTimeSeries[[25]]$X$ozone + GridTimeSeries[[37]]$X$ozone +
GridTimeSeries[[49]]$X$ozone + GridTimeSeries[[61]]$X$ozone) / 6)
feb <- data.frame(latitudes, longitudes, "February",
(GridTimeSeries[[02]]$X$ozone + GridTimeSeries[[14]]$X$ozone +
```

```
GridTimeSeries[[26]]$X$ozone + GridTimeSeries[[38]]$X$ozone +
GridTimeSeries[[50]]$X$ozone + GridTimeSeries[[62]]$X$ozone) / 6)
mar <- data.frame(latitudes, longitudes, "March",
(GridTimeSeries[[03]]$X$ozone + GridTimeSeries[[15]]$X$ozone +
GridTimeSeries[[27]]$X$ozone + GridTimeSeries[[39]]$X$ozone +
GridTimeSeries[[51]]$X$ozone + GridTimeSeries[[63]]$X$ozone) / 6)
apr <- data.frame(latitudes, longitudes, "April",
(GridTimeSeries[[04]]$X$ozone + GridTimeSeries[[16]]$X$ozone +
GridTimeSeries[[28]]$X$ozone + GridTimeSeries[[40]]$X$ozone +
GridTimeSeries[[52]]$X$ozone + GridTimeSeries[[64]]$X$ozone) / 6)
may <- data.frame(latitudes, longitudes, "May", (GridTimeSeries[[05]]$X$ozone
+ GridTimeSeries[[17]]$X$ozone + GridTimeSeries[[29]]$X$ozone +
GridTimeSeries[[41]]$X$ozone + GridTimeSeries[[53]]$X$ozone +
GridTimeSeries[[65]]$X$ozone) / 6)
jun <- data.frame(latitudes, longitudes, "June", (GridTimeSeries[[06]]$X$ozone
+ GridTimeSeries[[18]]$X$ozone + GridTimeSeries[[30]]$X$ozone +
GridTimeSeries[[42]]$X$ozone + GridTimeSeries[[54]]$X$ozone +
GridTimeSeries[[66]]$X$ozone) / 6)
jul <- data.frame(latitudes, longitudes, "July", (GridTimeSeries[[07]]$X$ozone
+ GridTimeSeries[[19]]$X$ozone + GridTimeSeries[[31]]$X$ozone +
GridTimeSeries[[43]]$X$ozone + GridTimeSeries[[55]]$X$ozone +
GridTimeSeries[[67]]$X$ozone) / 6)
aug <- data.frame(latitudes, longitudes, "August",
(GridTimeSeries[[08]]$X$ozone + GridTimeSeries[[20]]$X$ozone +
GridTimeSeries[[32]]$X$ozone + GridTimeSeries[[44]]$X$ozone +
GridTimeSeries[[56]]$X$ozone + GridTimeSeries[[68]]$X$ozone) / 6)
sep <- data.frame(latitudes, longitudes, "September",
(GridTimeSeries[[09]]$X$ozone + GridTimeSeries[[21]]$X$ozone +
GridTimeSeries[[33]]$X$ozone + GridTimeSeries[[45]]$X$ozone +
GridTimeSeries[[57]]$X$ozone + GridTimeSeries[[69]]$X$ozone) / 6)
oct <- data.frame(latitudes, longitudes, "October",
(GridTimeSeries[[10]]$X$ozone + GridTimeSeries[[22]]$X$ozone +
GridTimeSeries[[34]]$X$ozone + GridTimeSeries[[46]]$X$ozone +
GridTimeSeries[[58]]$X$ozone + GridTimeSeries[[70]]$X$ozone) / 6)
nov <- data.frame(latitudes, longitudes, "November",
(GridTimeSeries[[11]]$X$ozone + GridTimeSeries[[23]]$X$ozone +
GridTimeSeries[[35]]$X$ozone + GridTimeSeries[[47]]$X$ozone +
GridTimeSeries[[59]]$X$ozone + GridTimeSeries[[71]]$X$ozone) / 6)
dec <- data.frame(latitudes, longitudes, "December",
(GridTimeSeries[[12]]$X$ozone + GridTimeSeries[[24]]$X$ozone +
GridTimeSeries[[36]]$X$ozone + GridTimeSeries[[48]]$X$ozone +
GridTimeSeries[[60]]$X$ozone + GridTimeSeries[[72]]$X$ozone) / 6)

names <- c("latitudes", "longitudes", "month", "ozone")

colnames(jan) <- names;
colnames(feb) <- names;
```

```
colnames(mar) <- names;
colnames(apr) <- names;
colnames(may) <- names;
colnames(jun) <- names;
colnames(jul) <- names;
colnames(aug) <- names;
colnames(sep) <- names;
colnames(oct) <- names;
colnames(nov) <- names;
colnames(dec) <- names;

all.months <- rbind(jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov,
dec)

# Plotting the 12 months' data

library(ggplot2)
library(sf)
library(ggmap)
library(viridis)

# Create the map background
place <- 'central america'
(google <- get_googlemap(place, zoom = 5))
ggmap(google)
bbox_centralamerica <- c(left= -115.8, right = -59.2, top = 37.2, bottom=
-22.5)

# Plot the points over the map background
ggmap(get_stamenmap(bbox_centralamerica, maptype='toner-lite', zoom = 5)) +
  geom_point(data=all.months,
             aes(x=longitudes, y=latitudes, color=all.months$ozone, alpha=0))
+
  geom_point(color='orange', shape=20, size=3, stroke=FALSE) +
  scale_color_gradient(high='orangered', low='grey85') +
  guides(alpha=F) +

  facet_wrap(vars(month), ncol=4) +
  labs(title="Ozone Abundance Across Month",
        x="Longitude", y="Latitude", color="Ozone Abundance Density",
        alpha="Ozone Abundance Density") +
  theme(plot.title=element_text(family="Times", face="bold", size=12,
hjust=0.5),
        axis.title=element_text(family="Times", face="bold", size=10),
        legend.title=element_text(family="Times", face="bold", size=10))
```

Appendix 1.2

```
# Problem 1.2 - All six yearly averages for ozone measurements

y1 <- data.frame(latitudes, longitudes, "1995", (GridTimeSeries[[01]]$X$ozone
+ GridTimeSeries[[02]]$X$ozone + GridTimeSeries[[03]]$X$ozone +
GridTimeSeries[[04]]$X$ozone + GridTimeSeries[[05]]$X$ozone +
GridTimeSeries[[06]]$X$ozone + GridTimeSeries[[07]]$X$ozone +
GridTimeSeries[[08]]$X$ozone + GridTimeSeries[[09]]$X$ozone +
GridTimeSeries[[10]]$X$ozone + GridTimeSeries[[11]]$X$ozone +
GridTimeSeries[[12]]$X$ozone) / 12)
y2 <- data.frame(latitudes, longitudes, "1996", (GridTimeSeries[[13]]$X$ozone
+ GridTimeSeries[[14]]$X$ozone + GridTimeSeries[[15]]$X$ozone +
GridTimeSeries[[16]]$X$ozone + GridTimeSeries[[17]]$X$ozone +
GridTimeSeries[[18]]$X$ozone + GridTimeSeries[[19]]$X$ozone +
GridTimeSeries[[20]]$X$ozone + GridTimeSeries[[21]]$X$ozone +
GridTimeSeries[[22]]$X$ozone + GridTimeSeries[[23]]$X$ozone +
GridTimeSeries[[24]]$X$ozone) / 12)
y3 <- data.frame(latitudes, longitudes, "1997", (GridTimeSeries[[25]]$X$ozone
+ GridTimeSeries[[26]]$X$ozone + GridTimeSeries[[27]]$X$ozone +
GridTimeSeries[[28]]$X$ozone + GridTimeSeries[[29]]$X$ozone +
GridTimeSeries[[30]]$X$ozone + GridTimeSeries[[31]]$X$ozone +
GridTimeSeries[[32]]$X$ozone + GridTimeSeries[[33]]$X$ozone +
GridTimeSeries[[34]]$X$ozone + GridTimeSeries[[35]]$X$ozone +
GridTimeSeries[[36]]$X$ozone) / 12)
y4 <- data.frame(latitudes, longitudes, "1998", (GridTimeSeries[[37]]$X$ozone
+ GridTimeSeries[[38]]$X$ozone + GridTimeSeries[[39]]$X$ozone +
GridTimeSeries[[40]]$X$ozone + GridTimeSeries[[41]]$X$ozone +
GridTimeSeries[[42]]$X$ozone + GridTimeSeries[[43]]$X$ozone +
GridTimeSeries[[44]]$X$ozone + GridTimeSeries[[45]]$X$ozone +
GridTimeSeries[[46]]$X$ozone + GridTimeSeries[[47]]$X$ozone +
GridTimeSeries[[48]]$X$ozone) / 12)
y5 <- data.frame(latitudes, longitudes, "1999", (GridTimeSeries[[49]]$X$ozone
+ GridTimeSeries[[50]]$X$ozone + GridTimeSeries[[51]]$X$ozone +
GridTimeSeries[[52]]$X$ozone + GridTimeSeries[[53]]$X$ozone +
GridTimeSeries[[54]]$X$ozone + GridTimeSeries[[55]]$X$ozone +
GridTimeSeries[[56]]$X$ozone + GridTimeSeries[[57]]$X$ozone +
GridTimeSeries[[58]]$X$ozone + GridTimeSeries[[59]]$X$ozone +
GridTimeSeries[[60]]$X$ozone) / 12)
y6 <- data.frame(latitudes, longitudes, "2000", (GridTimeSeries[[61]]$X$ozone
+ GridTimeSeries[[62]]$X$ozone + GridTimeSeries[[63]]$X$ozone +
GridTimeSeries[[64]]$X$ozone + GridTimeSeries[[65]]$X$ozone +
GridTimeSeries[[66]]$X$ozone + GridTimeSeries[[67]]$X$ozone +
GridTimeSeries[[68]]$X$ozone + GridTimeSeries[[69]]$X$ozone +
GridTimeSeries[[70]]$X$ozone + GridTimeSeries[[71]]$X$ozone +
GridTimeSeries[[72]]$X$ozone) / 12)
```

```
names <- c("latitudes", "longitudes", "year", "ozone")

colnames(y1) <- names;
colnames(y2) <- names;
colnames(y3) <- names;
colnames(y4) <- names;
colnames(y5) <- names;
colnames(y6) <- names;

all.years <- rbind(y1, y2, y3, y4, y5, y6)

# Create the map background
place <- 'central america'
(google <- get_googlemap(place, zoom = 5))
ggmap(google)
bbox_centralamerica <- c(left= -115.8, right = -59.2, top = 37.2, bottom=
-22.5)

# Plot the points over the map background
ggmap(get_stamenmap(bbox_centralamerica, maptype='toner-lite', zoom = 5)) +
  geom_point(data=all.years,
             aes(x=longitudes, y=latitudes, color=all.years$ozone, alpha=0)) +
  geom_point(color='orange', shape=20, size=3, stroke=FALSE) +
  scale_color_gradient(high='orangered', low='grey85') +
  guides(alpha=F) +

  facet_wrap(vars(year), ncol=3) +
  labs(title="Ozone Abundance Across Year",
       x="Longitude", y="Latitude", color="Ozone Abundance Density",
       alpha="Ozone Abundance Density") +
  theme(plot.title=element_text(family="Times", face="bold", size=12,
hjust=0.5),
        axis.title=element_text(family="Times", face="bold", size=10),
        legend.title=element_text(family="Times", face="bold", size=10))
```

Appendix 2

```
# Problem 2

# Run only once
install.packages("devtools")
devtools::install_github("rensa/ggflags")

diving <- read.csv("Diving2000.csv")

judge.info <- unique(diving[c("judge", "jcountry")])
rownames(judge.info) <- 1:25

biases <- list(1:25)

for(i in 1:25)
{
  j <- as.character(judge.info[i,]$judge)
  c <- as.character(judge.info[i,]$jcountry)

  temp <- diving[which(diving$judge == j), ]
  domestic <- data.frame(temp[which(temp$dcountry == c), ])
  foreign <- data.frame(temp[which(temp$dcountry != c), ])

  biases[i] <- mean(domestic$score) - mean(foreign$score)
}

countries <- c("cu", "ca", "fr", "pr", "es", "cn", "de", "us",
               "mx", "ru", "gb", "au", "de", "ch", "at", "cn", "ca")

offsets <- c(-.1, .1, -.1, -.1, -.1, .1, .1, .1,
             .1, .1, .1, .1, .1, -.1, -.1, .1, .1)

judge.bias <- cbind(judge.info, unlist(biases), abs(unlist(biases)))
judge.bias <- na.omit(judge.bias)
judge.bias <- cbind(judge.bias, countries, offsets)
colnames(judge.bias) <- c("Judge", "Country", "Bias",
                          "Abs.Bias", "Code", "Offset")

# Plot the Bias
library(ggflags)
library(ggplot2)

ggplot(data=judge.bias, aes(x=reorder(Judge, Bias), y=Bias, fill=Abs.Bias)) +
  geom_bar(stat='identity') +
  geom_flag(aes(country=as.character(Code)), size=7,
y=judge.bias$Bias+judge.bias$Offset) +
```

```
  scale_fill_gradient(name="Absolute Value \n of Bias", high='red1',
low='green3', limits=c(0, 2.25)) +
  coord_flip() + expand_limits(y=c(-1.25, 2.25)) +
  labs(title="Are Judges Biased Towards Divers From Their Own Countries?",
       x="Judge", y="Bias") +
  theme(plot.title=element_text(family="Times", face="bold", size=12,
hjust=0.65),
        axis.title=element_text(family="Times", face="bold", size=10),
        legend.title=element_text(family="Times", face="bold", size=10))
```

Goal: Evaluate nationality bias from judges
Answer: Are judges more biased toward divers from their own countries? If so, which judges?
Approach Ideas:

- Group judges by their countries, find the average score given to divers (grouped by country)
- Average difficulty vs average score
- Select one diver, analyze highest scores and see if judges giving out highest scores are from native country

Algorithm for finding judge bias score:

1. Get list of all judge names
2. For each judge name:
    a. Get subset of Diving2000 dataset to only entries for that judge
    b. For each diver in the subset:
        i. If diver's country is same as judge's country, add to "domesticSum"
        ii. If diver's country is not same as judge's country, add to "foreignSum"
    c. Divide domesticSum and foreignSum by counts to get mean score
3. Plot domesticSum minus foreignSum
    a. X-axis: judge name
    b. Y-axis: average score difference
        i. Look for Positive values that signify bias
        ii. Look for highest positive value