Market Analysis of Bird-Based Tourism Using ebird Database

eBird is an online database of bird observation trips (checklists) providing scientists and naturalists with information about bird distribution and abundance.

Questions:

1. Which metric can we use to measure the popularity of bird watching in different counties?

2. How can we design successful birding tours?

3. What demographic/ economic factors contribute to the popularity of bird watching?

```
# Import libraries. Read files.
library(auk)
library(tidycensus)
library(tidyverse)
library(readxl)
library(urbnmapr)
library(gridExtra)
library(ggplot2)
library(corrplot)
library(modelr)
library(ranger)
library(gbm)
library(caret)
library(pdp)
library(lubridate)
library(ggmap)
library(gridExtra)
```

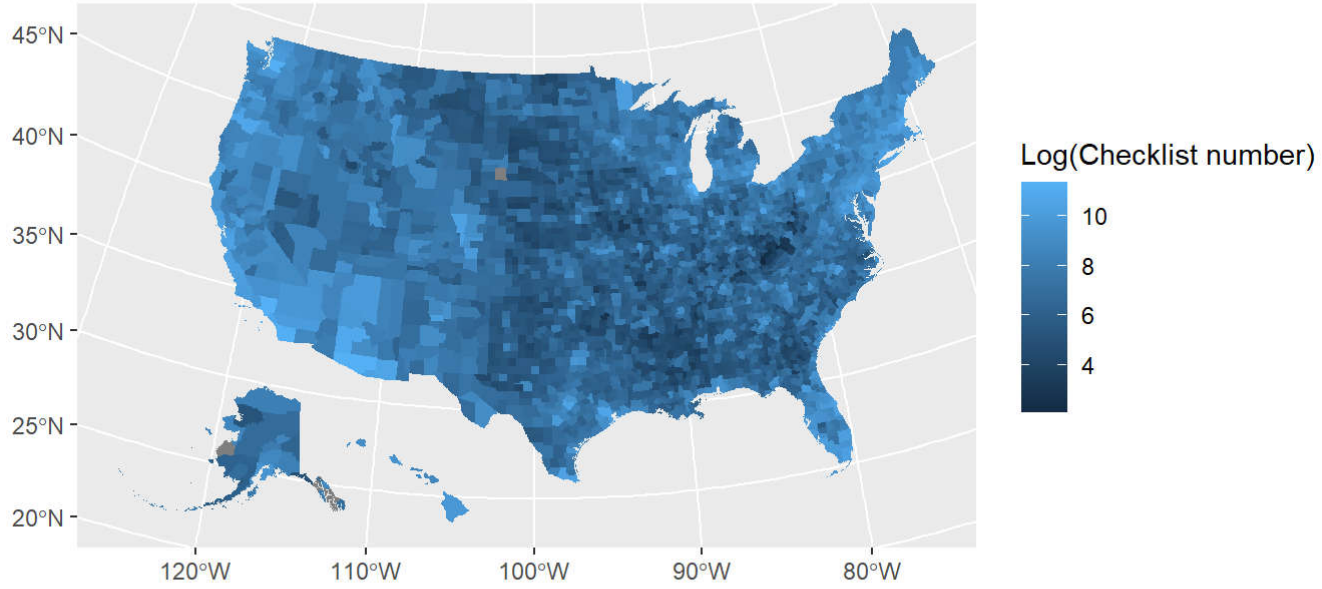
1. Which metric can we use to measure the popularity of bird watching in different counties?

1.1 Get the dataset

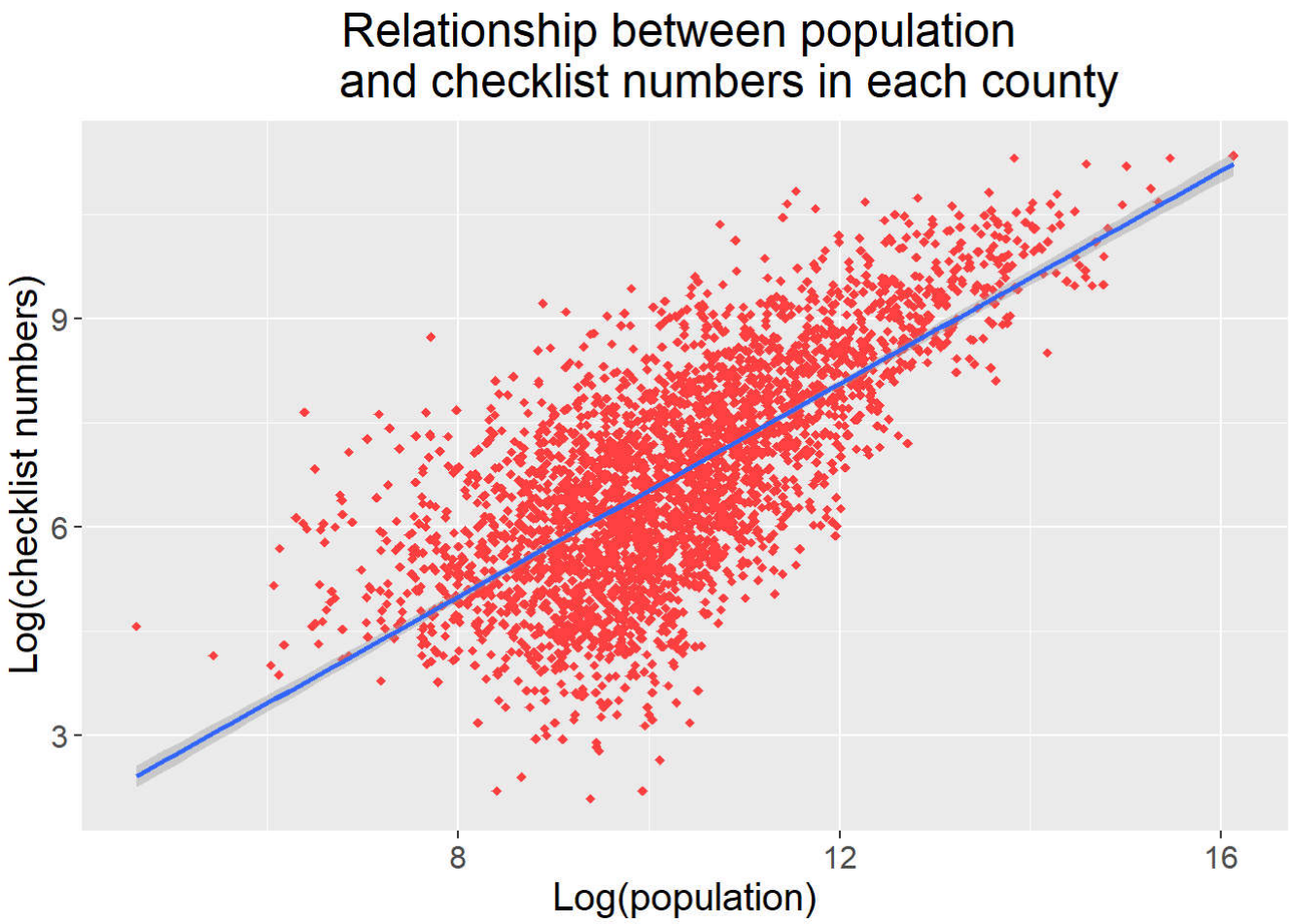
```
#read and filter checklists for 2018-2020 https://ebird.org/science/download-ebird-data-products
checklists_filters <- auk_sampling('F:/ebd_sampling_relMar-2020/ebd_sampling_relMar-2020.txt', s
ep = "\t") %>%
                       auk_date(date = c("2018-01-01", "2019-12-31")) %>% auk_country("United Sta
tes") %>% auk_filter(file = 'filtered_checklist.txt')
checklists_raw <- read_sampling(checklists_filters,unique = TRUE)

checklists_raw <- checklists_raw %>%
  select(checklist_id,state,county,county_code,observation_date,time_observations_started)

#get county level population, gender, age, race, citizen, income from US Census Bureau  https://
www.census.gov/data/developers/data-sets/ACS-supplemental-data.html
census_raw <- get_acs(geography = "county",
              variables = c(population = "B01003_001", male='B01001_002',female='B01001_026',med
ian_age='B01002_001',
                            white='B02001_002',black='B02001_003',native='B02001_004',asian='B02
001_005',pacific='B02001_006'                                                              ,nonecit
izen='B05001_006',incomepercapita='B19301_001'),
                            year = 2018, output = "wide") %>% select(-ends_with("M"))
```

1.2 Data cleaning, preprocessing

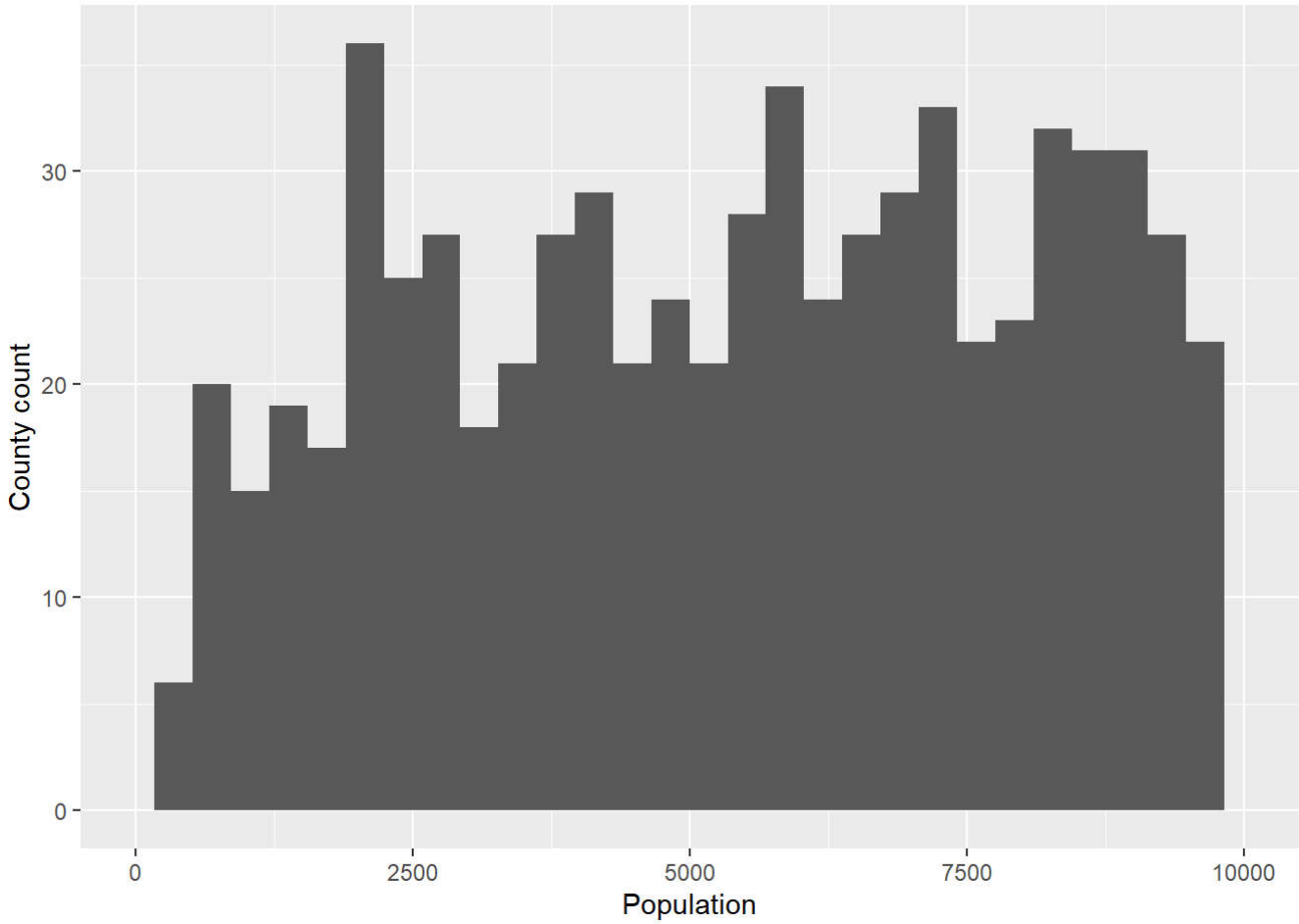
```
checklist_county_num <- checklists_raw %>%
  group_by(state,county,county_code) %>% count(name='checklist_n') %>% ungroup() %>%  drop_na()

#Feature transformation
census_raw <- census_raw %>%
  mutate(male=maleE/populationE, female=femaleE/populationE, white=whiteE/populationE,     black
=blackE/populationE,native=nativeE/populationE,asian=asianE/populationE, pacific=pacificE/popula
tionE,otherrace=1-white-black-native-asian-pacific,citizen=1-nonecitizenE/populationE) %>%
  select(-(maleE:femaleE),-(whiteE:nonecitizenE))

#Employment statistics from Department of Labor, Education level from USDA
unemployment_raw <- read_xlsx('D:/ebird/data/birding_market/bls_laucnty18_labor.xlsx') %>%
  unite("FIPS", c('State FIPS','county FIPS'),sep = '') %>%
  mutate(unemployment_p=`Unemployment rate(%)`) %>%
  select(-'Unemployment rate(%)')

#https://www.ers.usda.gov/data-products/county-level-data-sets/
education_raw <- read_xls('D:/ebird/data/birding_market/Education_USDA.xls') %>%
  mutate(FIPS=`FIPS Code`) %>% select(-`FIPS Code`)

census_raw <- census_raw %>%
  inner_join(unemployment_raw,by = c("GEOID" = "FIPS")) %>%
  inner_join(education_raw,by = c("GEOID" = "FIPS")) %>%
  mutate(county_fips=substr(GEOID, start = 3, stop = 5)) %>%
  separate(NAME,c(NA,'state'),sep=', ') %>%
           select(-c(County,'Area name'))

#https://github.com/birderboone/birderboone.github.io/blob/master/ebirdSpeciesModeled.csv
species_n <- read.csv('D:/ebird/data/birding_market/ebirdSpecies.csv')

checklist_county_num <- checklist_county_num %>% separate(county_code,c(NA,NA,'county_fips'),sep
='-')

census_ebird_raw <- left_join(checklist_county_num,census_raw, by=c("state",'county_fips')) %>%
  mutate(county=tolower(county)) %>%
  left_join(species_n, by=c('State'='state','county'))
```




```
ggplot(census_ebird_raw %>% drop_na(), aes(x=log(populationE), y=log(checklist_n))) +
  geom_point(shape=18, color="brown1")+
  geom_smooth(method=lm)+
  labs(title="Relationship between population
       and checklist numbers in each county",
       x="Log(population)", y = "Log(checklist numbers)")+
  theme(plot.title = element_text(hjust = 0.5),text = element_text(size=15))
```



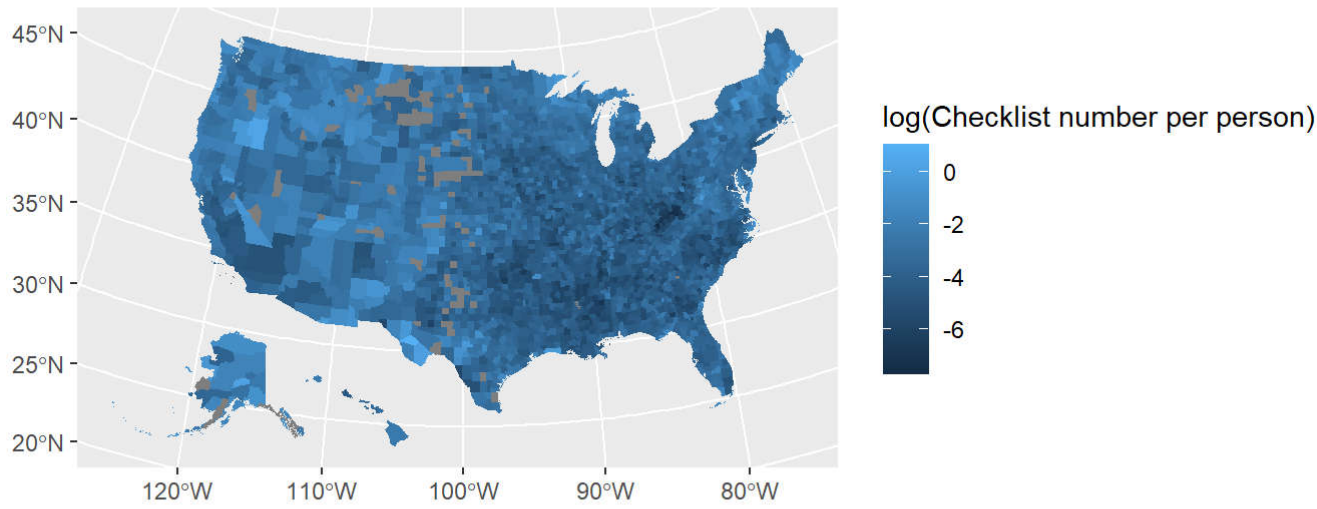

```
ggplot(census_ebird_raw, aes(x=populationE)) +
  geom_histogram()+
  xlim(0, 10000)+
  ylab("County count")+
  xlab("Population")
```




```
census_ebird_filtered <- census_ebird_raw %>%
  filter(populationE>=2000)
```

1.3 Use checklists per person as a measurement for the popularity of bird watching in each county

```
census_ebird_filtered <- census_ebird_filtered %>%
  mutate(checklist_pps=checklist_n/populationE)

counties_sf %>% left_join(census_ebird_filtered %>%
                          select(state,county_fips,checklist_pps), by = c("state_name"='state'
,'county_fips')) %>%
  ggplot() +
  geom_sf(mapping = aes(fill = log(checklist_pps)),
          color = NA, size = 0.05) +
  labs(fill = "log(Checklist number per person)")
```



2. Where should we start the business and how to design a successful product?

2.1 Select the candidate county

```
census_ebird_filtered_des <- readRDS("D:/ebird/data/birding_market/import_rawdata/census_ebird_f
iltered.rds")
census_ebird_filtered_des <- census_ebird_filtered_des %>% filter(populationE>=10000) %>%   #Pop
ulation more than 10000
  select(state,county,checklist_n,populationE,totalSpSeen,checklist_pps) %>%
  arrange(desc(checklist_pps))
census_ebird_filtered_des
```

state <chr>	county <chr>	checklist_n <int>	populationE <dbl>	totalSpSeen <int>	checklist_pps <dbl>
New Mexico	los alamos	12556	18356	280	0.6840270211
Arizona	santa cruz	31558	46584	425	0.6774428988
California	mono	8415	14174	365	0.5936926767
Virginia	northampton	6507	11957	412	0.5442000502
Illinois	menard	6599	12367	241	0.5335974772

state <chr>	county <chr>	checklist_n <int>	populationE <dbl>	totalSpSeen <int>	checklist_pps <dbl>
New Mexico	socorro	8769	17000	405	0.5158235294
New York	tompkins	50968	102962	330	0.4950175793
Massachusetts	nantucket	5439	11101	365	0.4899558598
Maine	hancock	25075	54541	327	0.4597458792
New Jersey	cape may	42552	93705	418	0.4541059709

1-10 of 2,443 rows Previous **1** 2 3 4 5 6 … 245 Next

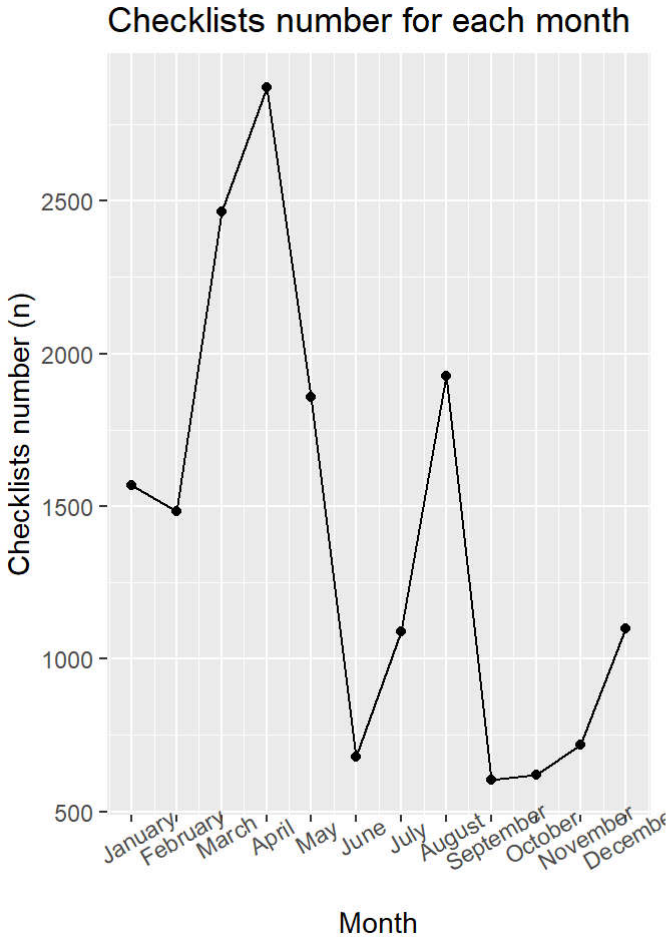
2.2 Read the checklist information from Arizona Santa Cruz county

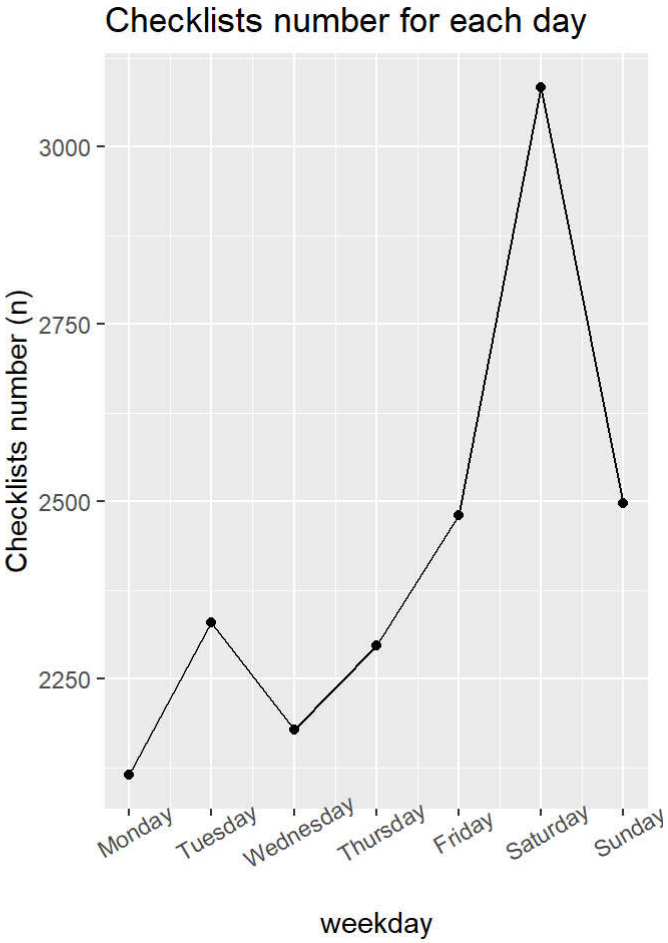
```
checklists_sc <- checklists_raw %>%
  filter(state=='Arizona',county=='Santa Cruz') %>%
  select("checklist_id","observation_date","time_observations_started","duration_minutes", "effo
rt_distance_km",'latitude','longitude') %>%
  distinct()
```

```
#Convert hh:mm:ss to decimal time
time_to_decimal <- function(x) {
  x <- hms(x, quiet = TRUE)
  hour(x) + minute(x) / 60 + second(x) / 3600
}

checklists_sc <- checklists_sc %>%
  mutate(time_observations_started = time_to_decimal(time_observations_started),
  day_of_year = yday(observation_date), month_of_year=month(observation_date), weekday=wday(obse
rvation_date)) %>%
  drop_na()
```

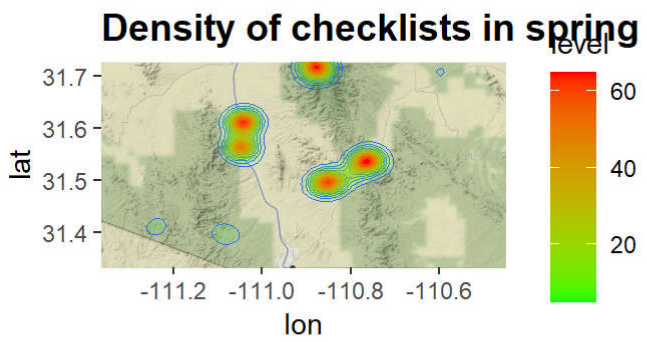
2.3 People’s favorite months and days for bird watching

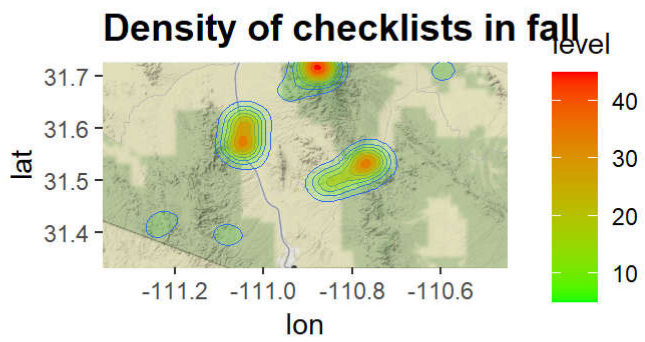
```
# summarize data by hourly bins
month_n <- checklists_sc %>%
  group_by(month_of_year) %>%
  summarise(n_checklists = n())

month_fig <- ggplot(month_n) +
  aes(x = month_of_year, y = n_checklists) +
  geom_line() +
  geom_point() +
  labs(x = "Month",
       y = "Checklists number (n)",
       title = "Checklists number for each month") +
  scale_x_continuous(breaks = 1:12,labels = c('January', 'February', 'March', 'April', 'May', 'J
une', 'July', 'August', 'September', 'October', 'November', 'December'))+
  theme(axis.text.x = element_text(angle = 30))

weekday_n <- checklists_sc %>%
  group_by(weekday) %>%
  summarise(n_checklists = n()) %>%
  ungroup() %>%
  mutate(weekday=weekday-1) %>%
  mutate(weekday=ifelse(weekday==0, 7,weekday))

weekday_fig <- ggplot(weekday_n) +
  aes(x = weekday, y = n_checklists) +
  geom_line() +
  geom_point() +
  labs(y = "Checklists number (n)",
       title = "Checklists number for each day") +
  scale_x_continuous(breaks = 1:7,labels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Frida
y", "Saturday", "Sunday")) +
    theme(axis.text.x = element_text(angle = 30))

# combine
grid.arrange(month_fig, weekday_fig, nrow=1)
```


Checklists number for each month
Checklists number (n)
2500
2000
1500
1000
500
January
February
March
April
May
June
July
August
September
October
November
Month


Checklists number for each day
Checklists number (n)
3000
2750
2500
2250
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
weekday

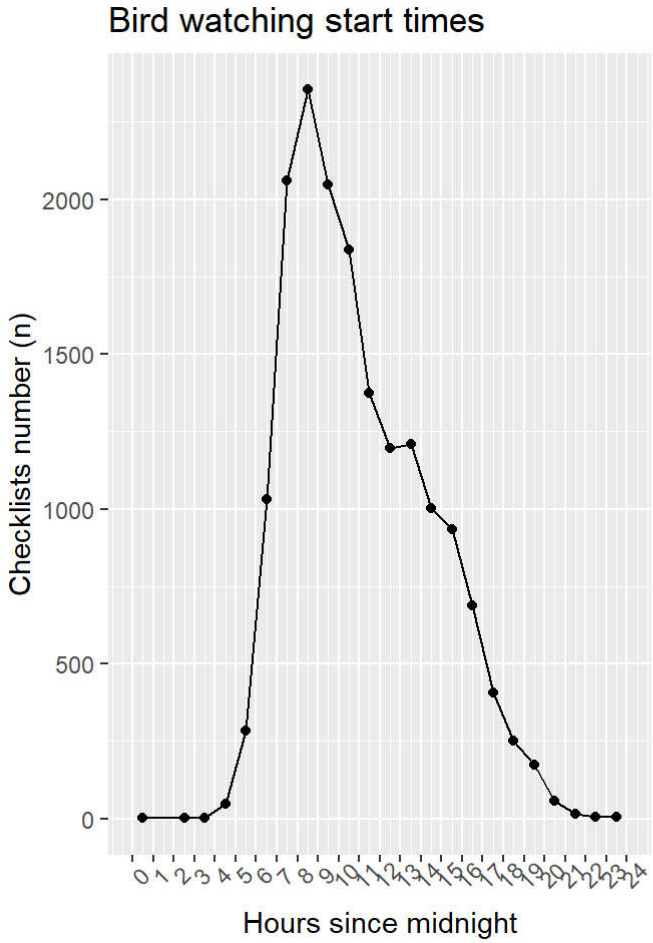
```
#Latitude and longitude of sf found through https://www.latlong.net/
sc_map <- get_stamenmap(bbox = c(left = -111.363808, top=31.727364, bottom=31.331352, right=-11
0.448482), zoom = 10)

checklist_spring <- checklists_sc %>% filter(month_of_year%in%c(3,4,5))

map_spring <- ggmap(sc_map) +
  #contour
  geom_density2d(data = checklist_spring, aes(x = longitude, y = latitude), size = 0.3)+
  #fill in the color according to density
  stat_density2d(data = checklist_spring, aes(x = longitude, y = latitude, fill = ..level.., alp
ha = ..level..), geom = "polygon") +
  scale_fill_gradient(low = "green", high = "red") +
  scale_alpha(range = c(0.1, 1), guide = FALSE)+
  theme(plot.title = element_text(size = 14, face = "bold"))+
  labs(title = "Density of checklists in spring")

checklist_fall <- checklists_sc %>% filter(month_of_year%in%c(7,8,9))

map_fall <- ggmap(sc_map) +
  #contour
  geom_density2d(data = checklist_fall, aes(x = longitude, y = latitude), size = 0.3)+
  #fill in the color according to density
  stat_density2d(data = checklist_fall, aes(x = longitude, y = latitude, fill = ..level.., alpha
= ..level..), geom = "polygon") +
  scale_fill_gradient(low = "green", high = "red") +
  scale_alpha(range = c(0.1, 1), guide = FALSE) +
  theme(plot.title = element_text(size = 14, face = "bold"))+
  labs(title = "Density of checklists in fall" )

grid.arrange(map_spring, map_fall, nrow=1)
```

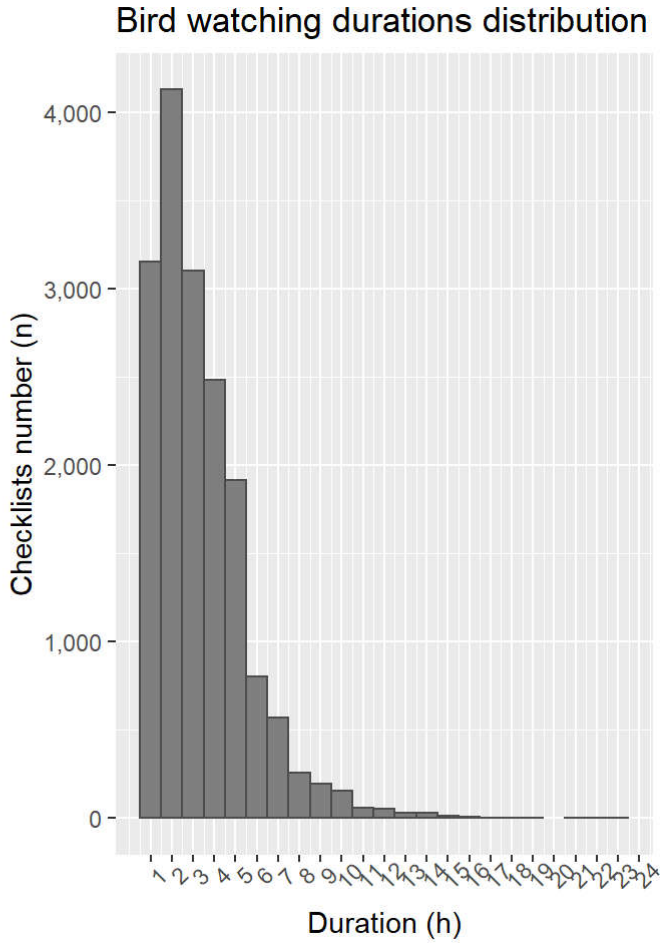




2.4 People's preference for start time and trip duration

```
breaks <- 0:24
labels <- breaks[-length(breaks)] + diff(breaks) / 2
starttime_n <- checklists_sc %>%
  mutate(startt = cut(time_observations_started,
                       breaks = breaks,
                      labels = labels,
                       include.lowest = TRUE),
         startt = as.numeric(as.character(startt))) %>%
  group_by(startt) %>%
  drop_na() %>%
  summarise(n_checklists = n())

starttime_fig <- ggplot(starttime_n) +
  aes(x = startt, y = n_checklists) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = 0:24, limits = c(0, 24)) +
  labs(x = "Hours since midnight",
       y = "Checklists number (n)",
       title = "Bird watching start times")+
  theme(axis.text.x = element_text(angle = 45))


breaks <- 0:24
labels <- 1:24
duration_n <- checklists_sc %>%
  mutate(duration_h=duration_minutes/36) %>%
  filter(duration_h<=24) %>%
  mutate(duration = cut(duration_h,
                      breaks = breaks,
                      label = labels,
                      include.lowest = TRUE),
         duration = as.numeric(as.character(duration))) %>%
  drop_na() %>%
  group_by(duration) %>%
  summarise(n_checklists = n())

# histogram
duration_fig <- ggplot(duration_n) +
  aes(x = duration, y = n_checklists) +
  geom_col(width = mean(diff(breaks)), color = "grey30", fill = "grey50") +
  scale_x_continuous(breaks = 1:24) +
  scale_y_continuous(labels = scales::comma) +
  labs(x = "Duration (h)",
       y = "Checklists number (n)",
       title = "Bird watching durations distribution")+
  theme(axis.text.x = element_text(angle = 45))

# combine
grid.arrange(starttime_fig, duration_fig, nrow=1)
```

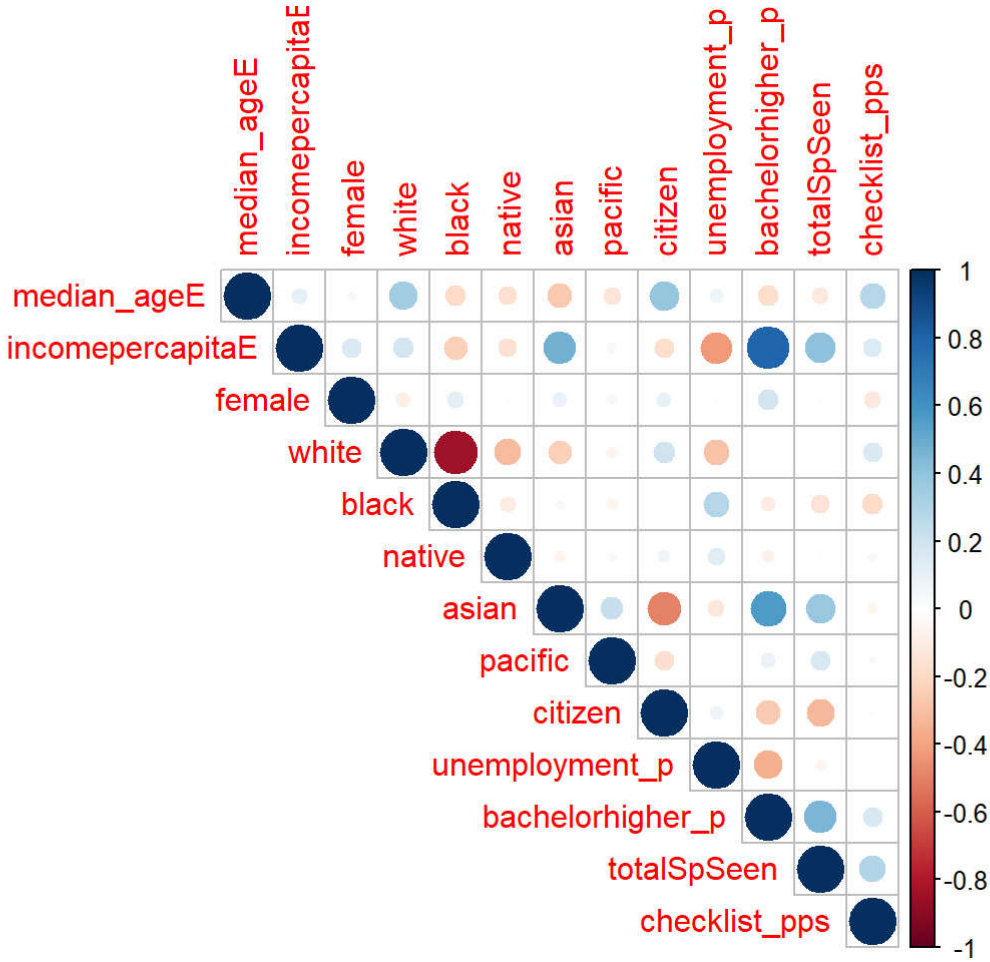




3. Which demographic/ economic factors contribute to the popularity of bird watching? The demographic features for our potential customers (The demographic features of ebird users are not provided by the database).

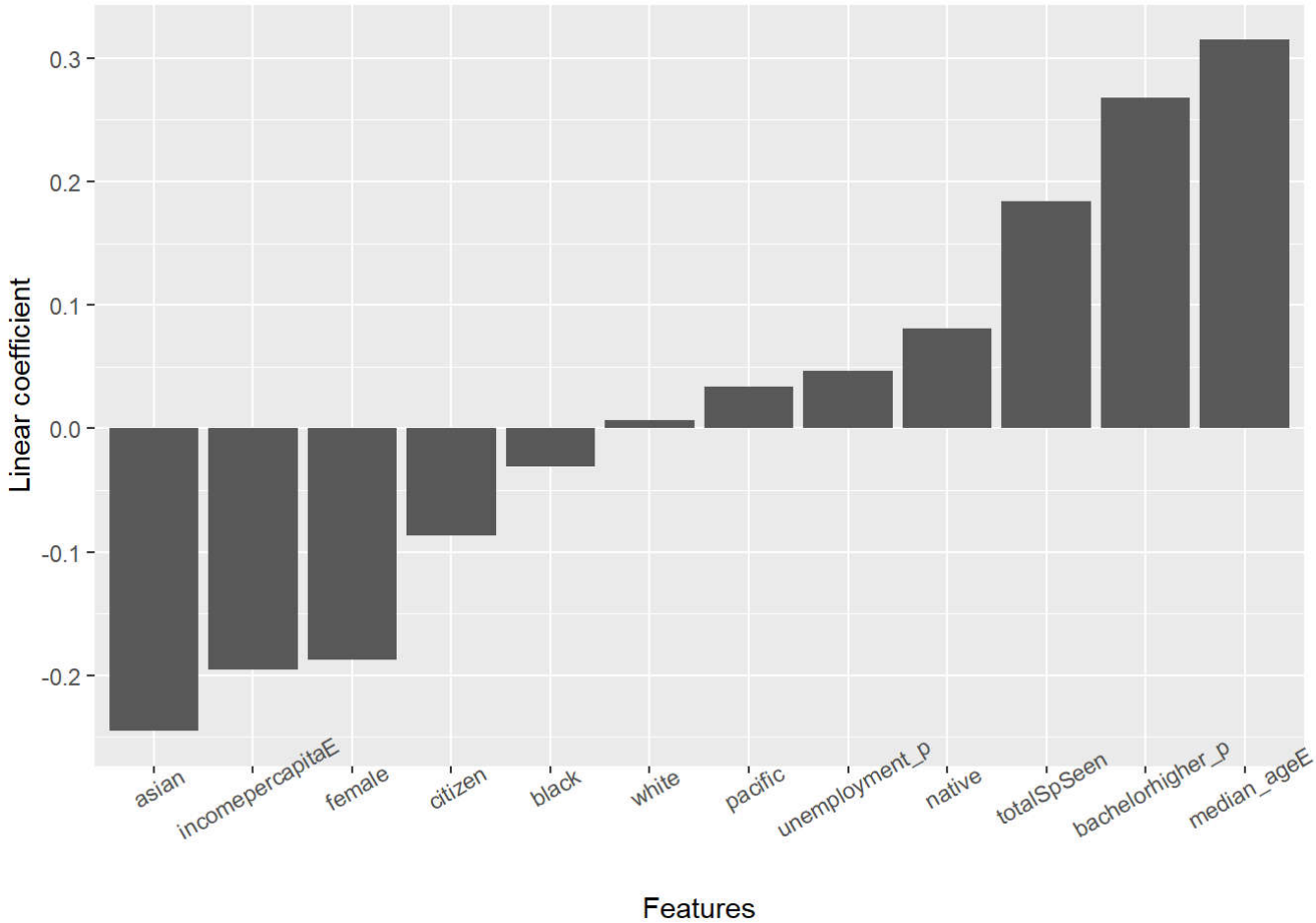
3.1 The Pearson correlation between checklist number per person and other features

```
census_ebird_filtered <- census_ebird_filtered %>%
  select(-c('county','State',"state","county_fips","checklist_n","GEOID","populationE","male","o
therrace","Lesshighschool_p","highschool_p",  "college_p","checklists","modeledTotal","speciesDi
fference" )) %>%
  filter(checklist_pps<=1) %>%
  drop_na()

feature_cor=cor(census_ebird_filtered, method = "pearson")
corrplot(feature_cor, type="upper")
```



3.2 Use linear regression to predict the popularity of bird watching in each county.

```
normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x)))}
census_ebird_nor <- census_ebird_filtered %>% mutate_at(vars(median_ageE:totalSpSeen), normaliz
e)

#Use 80% samples to fit the model, use another 20% to test the model
set.seed(1)
in.trn <- sample.int(nrow(census_ebird_nor), 0.8*nrow(census_ebird_nor))
census_ebird_nor_trn <- census_ebird_nor[ in.trn,]
census_ebird_nor_tst <- census_ebird_nor[-in.trn,]

linearMod <- lm(checklist_pps ~ ., data=census_ebird_nor_trn)

linear_coeff <- as.data.frame(sort(linearMod$coefficients[-1]))
colnames(linear_coeff) <- 'coeff'
linear_coeff$feature <- rownames(linear_coeff)

ggplot(data=linear_coeff, aes(x=reorder(row.names(linear_coeff),coeff), y=coeff))+
  geom_bar(stat="identity")+
  theme(axis.text.x = element_text(angle = 30))+
  ylab("Linear coefficient")+
  xlab("Features")
```




```
error=matrix(0,nrow=2,ncol=2, dimnames = list(c('linear','GBM'), c('training error','test error'
)))
error[1,1]=sqrt(sum((linearMod$residuals)^2)/length(linearMod$residuals))
```

```
tstPred <- predict(linearMod, census_ebird_nor_tst)

linearMod_result <- as.data.frame(cbind(actuals=census_ebird_nor_tst$checklist_pps, predicteds=t
stPred))

rmse_linear <- sqrt(mean((linearMod_result$actuals - linearMod_result$predicteds)^2))
error[1,2]=rmse_linear
sprintf('The RMSE for linear model on test data is %f',rmse_linear)
```

```
## [1] "The RMSE for linear model on test data is 0.062296"
```

3.3 Use Gradient Boosting Machine to predict the popularity of bird watching in each county. Find the optimized parameters

```
set.seed(66)
# train GBM model, find the range of iteration
gbm.fit <- gbm(
  formula = checklist_pps ~ .,
  distribution = "gaussian",
  data = census_ebird_nor_trn,
  n.trees = 10000,  #total number of train
  interaction.depth = 3,  #tree height, determine the number of splits
  shrinkage = 0.01,   #Learning rate
  cv.folds = 5,     #5 fold cross validation
  n.cores = NULL,
  verbose = FALSE  #don't print progress and performance indicators
  )
gbm.perf(gbm.fit, plot.it = FALSE,method = "cv")
```

```
## [1] 2100
```

```
hyper_grid <- expand.grid(
  shrinkage = c(.005,0.01, .1),
  interaction.depth = c(1,5,10),
  n.minobsinnode = c(5, 10, 15),   # minimum number of observations in tree terminal nodes
  bag.fraction = c(0.65,0.8,0.9),
  optimal_trees = 0,               # a place to dump results
  min_RMSE = 0                     # a place to dump results
)


# grid search
for(i in 1:nrow(hyper_grid)) {

  set.seed(66)

  gbm.tune <- gbm(
    formula = checklist_pps ~ .,
    distribution = "gaussian",
    data = census_ebird_nor_trn,
    n.trees = 2000,
    interaction.depth = hyper_grid$interaction.depth[i],
    shrinkage = hyper_grid$shrinkage[i],
    n.minobsinnode = hyper_grid$n.minobsinnode[i],
    bag.fraction = hyper_grid$bag.fraction[i],
    train.fraction = .75,
    n.cores = NULL,
    verbose = FALSE
  )

  hyper_grid$optimal_trees[i] <- which.min(gbm.tune$valid.error)
  hyper_grid$min_RMSE[i] <- sqrt(min(gbm.tune$valid.error))
}

hyper_grid %>%
  dplyr::arrange(min_RMSE) %>%
  head(10)
```

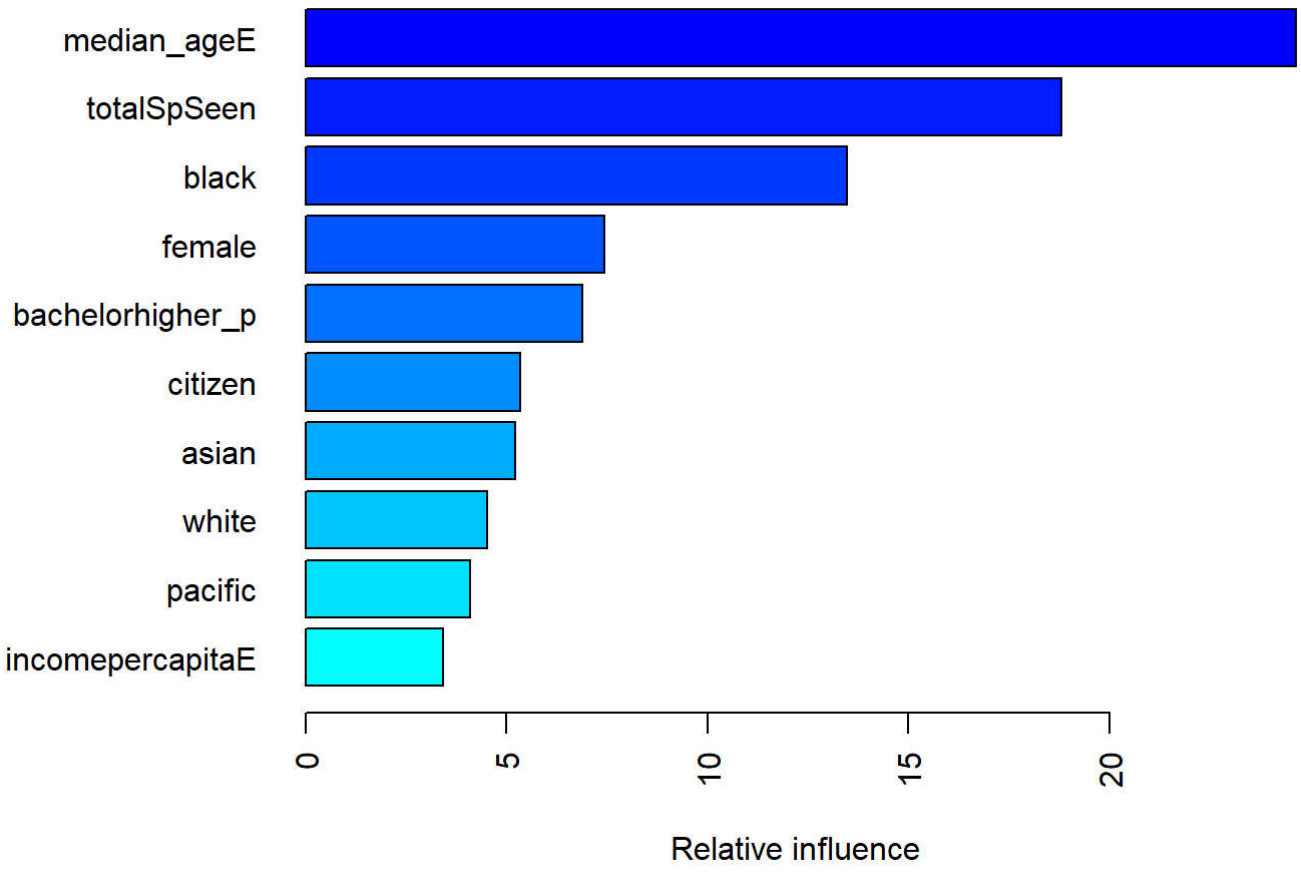
	shrinkage <dbl>	interaction.depth <dbl>	n.minobsinnode <dbl>	bag.fraction <dbl>	optimal_trees <dbl>	min_RMSE <dbl>
1	0.010	10	5	0.65	1136	0.05815085
2	0.010	10	5	0.80	1306	0.05817272
3	0.010	5	5	0.80	1896	0.05833896
4	0.010	10	5	0.90	1864	0.05861208
5	0.010	5	15	0.90	1989	0.05871387
6	0.100	5	15	0.65	249	0.05878527
7	0.100	10	15	0.65	146	0.05878953
8	0.005	10	5	0.65	1999	0.05880478

	shrinkage <dbl>	interaction.depth <dbl>	n.minobsinnode <dbl>	bag.fraction <dbl>	optimal_trees <dbl>	min_RMSE <dbl>
9	0.005	10	5	0.90	1936	0.05881260
10	0.100	10	15	0.80	103	0.05884034

1-10 of 10 rows

3.4 Train the final GBM model, identify the most relevant variables

```
set.seed(66)
# train GBM model
gbm.fit.final <- gbm(
  formula = checklist_pps ~ .,
  distribution = "gaussian",
  data = census_ebird_nor_trn,
  n.trees = 1136,
  interaction.depth = 10,
  shrinkage = 0.01,
  n.minobsinnode = 5,
  bag.fraction = .65,
  train.fraction = .75,
  n.cores = NULL,
  verbose = FALSE
  )

par(mar = c(5, 8, 1, 1))
summary(
  gbm.fit.final,
  cBars = 10,
  method = relative.influence, # also can use permutation.test.gbm
  las = 2
  )
```

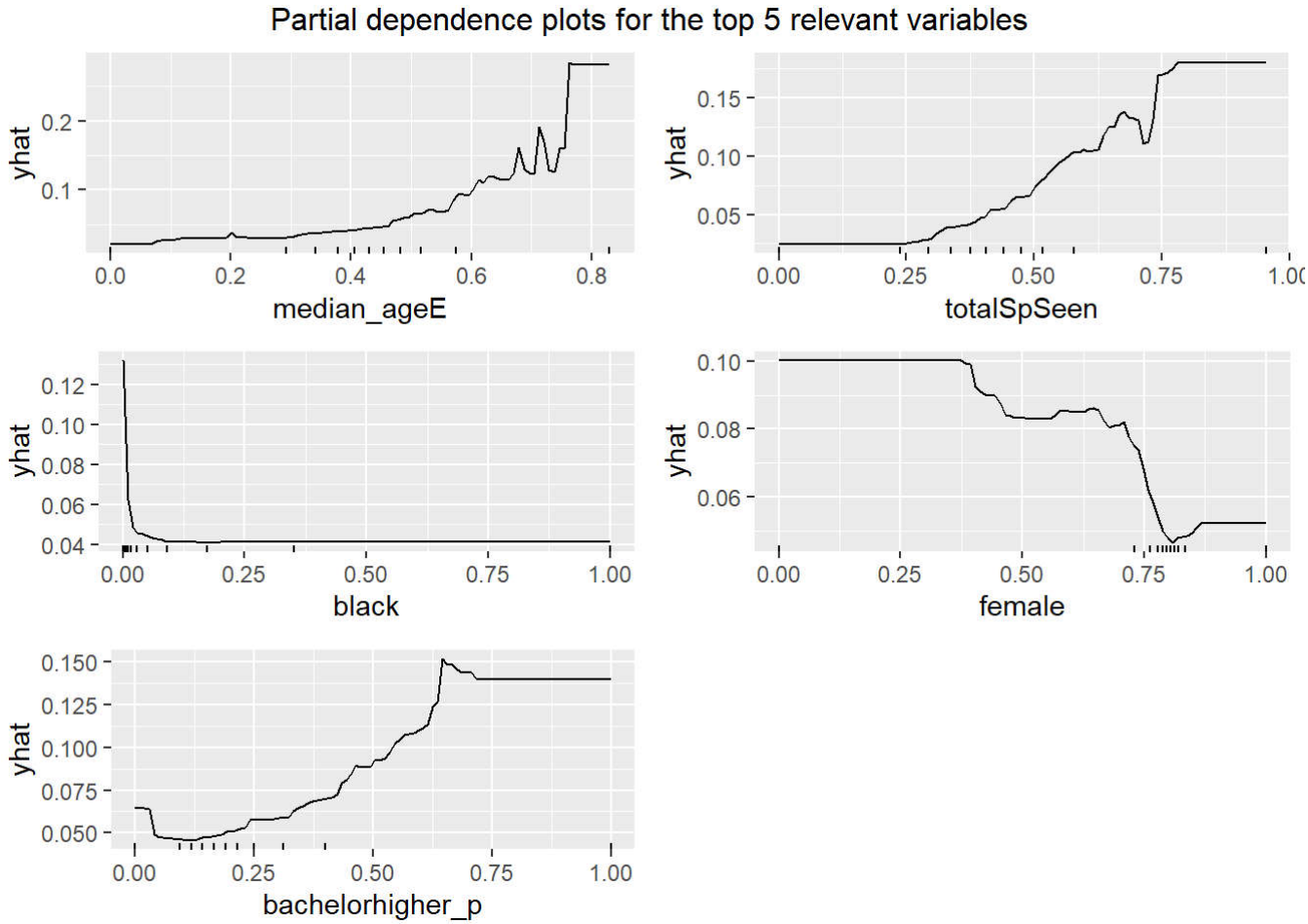


	var <fctr>	rel.inf <dbl>
median_ageE	median_ageE	24.641328
totalSpSeen	totalSpSeen	18.801576
black	black	13.479613
female	female	7.435628
bachelorhigher_p	bachelorhigher_p	6.878394
citizen	citizen	5.348987
asian	asian	5.232043
white	white	4.521422
pacific	pacific	4.099578
incomepercapitaE	incomepercapitaE	3.420782

1-10 of 12 rows Previous **1** 2 Next

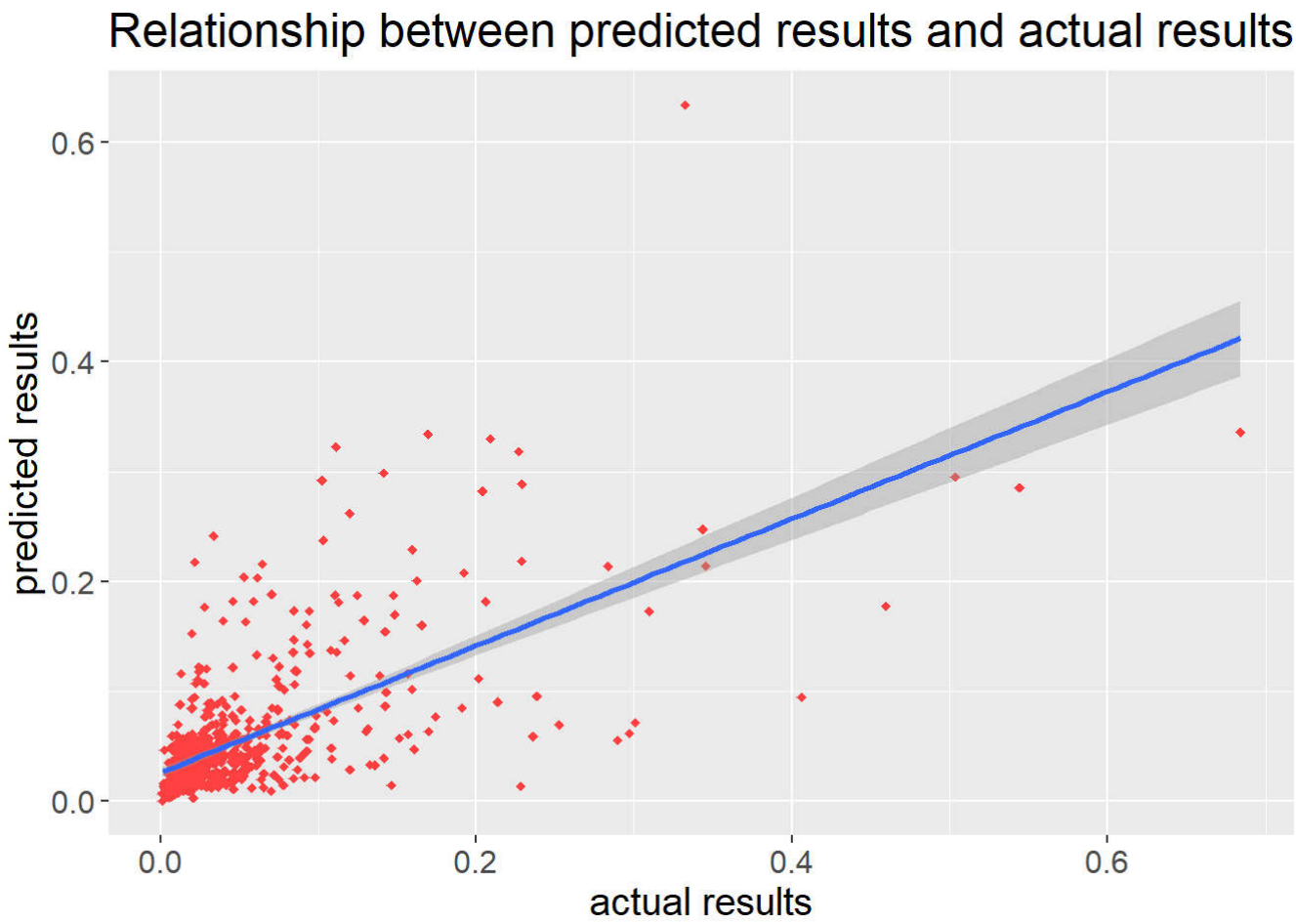
```
error[2,1] <- sqrt(sum((gbm.fit.final$fit-census_ebird_nor_trn$checklist_pps)^2)/length(census_ebird_nor_trn$checklist_pps))
```

```
fig1=gbm.fit.final %>%
  partial(
    pred.var = "median_ageE",
    n.trees = gbm.fit.final$n.trees,
    grid.resolution = 100
    ) %>%
  autoplot(rug = TRUE, train = census_ebird_nor_trn)

fig2=gbm.fit.final %>%
  partial(
    pred.var = "totalSpSeen",
    n.trees = gbm.fit.final$n.trees,
    grid.resolution = 100
    ) %>%
  autoplot(rug = TRUE, train = census_ebird_nor_trn)

fig3=gbm.fit.final %>%
  partial(
    pred.var = "black",
    n.trees = gbm.fit.final$n.trees,
    grid.resolution = 100
    ) %>%
  autoplot(rug = TRUE, train = census_ebird_nor_trn)

fig4=gbm.fit.final %>%
  partial(
    pred.var = "female",
    n.trees = gbm.fit.final$n.trees,
    grid.resolution = 100
    ) %>%
  autoplot(rug = TRUE, train = census_ebird_nor_trn)

fig5=gbm.fit.final %>%
  partial(
    pred.var = "bachelorhigher_p",
    n.trees = gbm.fit.final$n.trees,
    grid.resolution = 100
    ) %>%
  autoplot(rug = TRUE, train = census_ebird_nor_trn)

grid.arrange(fig1, fig2,fig3,fig4,fig5, top="Partial dependence plots for the top 5 relevant var
iables")
```



3.5 Test the final GBM model on test data

```
gbm_pred <- predict(gbm.fit.final, n.trees = gbm.fit.final$n.trees, census_ebird_nor_tst)
error[2,2]=RMSE(gbm_pred, census_ebird_nor_tst$checklist_pps)

gbmMod_result <- as.data.frame(cbind(actuals=census_ebird_nor_tst$checklist_pps, predicteds=gbm_
pred))

ggplot(gbmMod_result,aes(x=actuals, y=predicteds)) +
  geom_point(shape=18, color="brown1")+
  geom_smooth(method=lm)+
  labs(title="Relationship between predicted results and actual results",
       x="actual results", y = "predicted results")+
  theme(plot.title = element_text(hjust = 0.5),text = element_text(size=15))
```




```
print(error)
```

```
##        training error test error
## linear     0.07157802 0.06229614
## GBM        0.03768957 0.05569177
```

Conclusions:

1. Checklist number per person is a good metric for measuring popularity of bird watching.

2. Santa Cruz, AZ is a potential place to start the business. People prefer 1-3 hours trips in Spring on Saturday.

3. Our potential customers are more likely to be older man with higher education level.