HM07_final

Sedreh

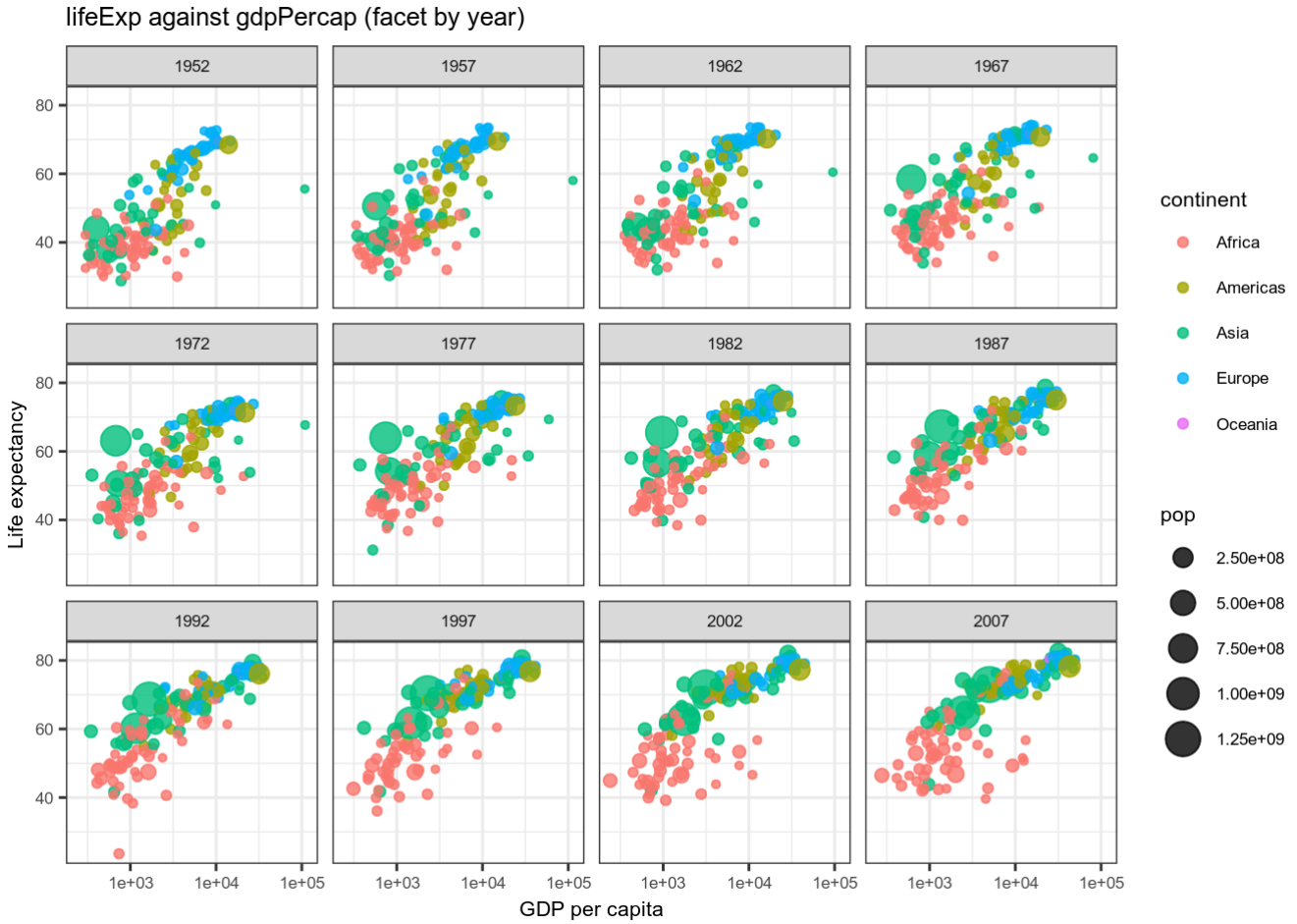
5/10/2019

```
library(tidyr)
library(ggplot2)
library(ggthemes)
library(gapminder)
library(gganimate)
theme_set(theme_bw())
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

```
#let's see lifeExp against gdpPercap
p <- ggplot(gapminder, aes(x = gdpPercap, y=lifeExp, size = pop, colour = continent)
  ) +
  geom_point(show.legend = TRUE, alpha = 0.5) +
  scale_x_log10() +
  labs(x = "GDP per capita", y = "Life expectancy") +
  ggtitle("lifeExp against gdpPercap")
  #theme(text = element_text(size = 8))
p
```


```
#lifeExp against gdpPercap ( see all of our datafacetting by year)

p <- ggplot(gapminder,aes(x=gdpPercap, y=lifeExp,color=continent,size=pop))+
  geom_point(alpha = 0.8)+
  facet_wrap(~year)+
  scale_x_log10() +
  labs(x = "GDP per capita", y = "Life expectancy")+
  ggtitle("lifeExp against gdpPercap (facet by year)")+
  theme(text = element_text(size = 8))
p
```




```
#I found some beautiful function and I just wanted practice it in this homework!!!

# Transition_time() function. The transition length between the states will be set to correspond to the actual time difference between them.
# Label variables: frame_time. Gives the time that the current frame corresponds to.
#
# library(gapminder)
#
# p <-ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
#   geom_point(alpha = 0.7, show.legend = TRUE) +
#   scale_colour_manual(values = country_colors) +
#   scale_size(range = c(2, 12)) +
#   scale_x_log10() +
#   facet_wrap(~continent) +
#   # Here comes the gganimate specific bits
#   labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
#   transition_time(year) +
#   ease_aes('linear')
# p
```

```
############ Data: Airquality, transform, plot all measures by time ###############
#understand the data
# The air quality dataset, which pertains to the daily air quality measurements in New York from May to September 1973. This dataset consists of more than 100 observations on 6 variables i.e. Ozone(mean parts per billion), Solar.R(Solar Radiation), Wind(Average wind speed), Temp(maximum daily temperature in Fahrenheit), Month(month of observation) and Day(Day of the month)
```

```
library(datasets)
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
sum(is.na(airquality))
```

```
## [1] 44
```

```
#we should clean data! it contains missing values
airquality_clean <- na.omit(airquality)
#also we can do it inside ggplot like this: (geom_point(na.rm=TRUE))
head(airquality_clean)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 7    23     299  8.6   65     5   7
## 8    19      99 13.8   59     5   8
```

```
sum(is.na(airquality_clean))
```

```
## [1] 0
```

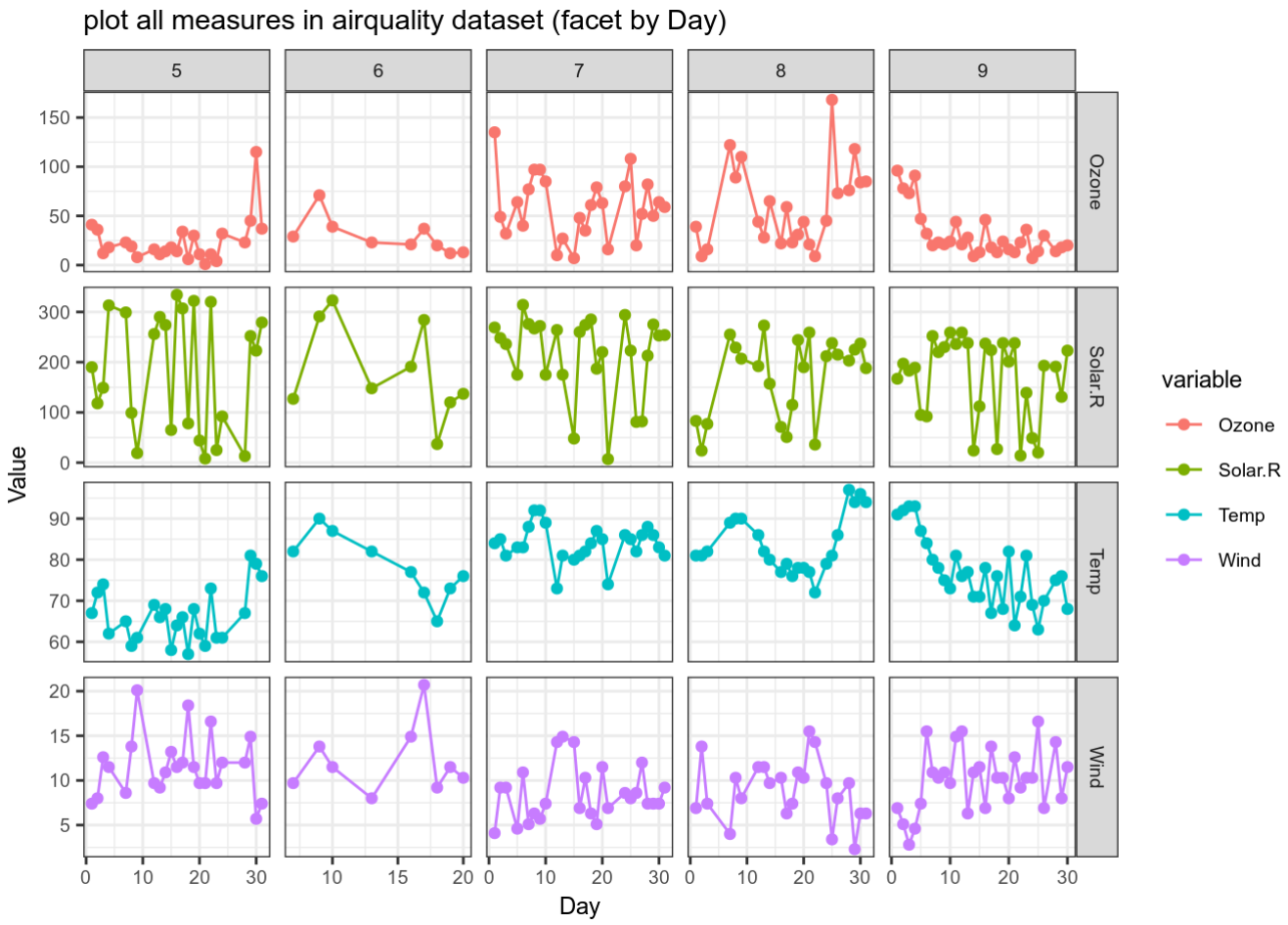
```
# In “wide” (“multivariate”) format, the three environmental parameters, ozone, wind
 and temp appear as separate variables. In “long ” (“univariate”) format, they appear
as different levels of the variable parameter. tidyr is a very powerful package for c
onverting between long (univariate) and wide (multivariate) formats(also we can use r
eshape2).
# In wide format, categorical data is always grouped. You can think of it as a summar
y of long data. It is easier to read and interpret as compared to long format.
# In long vertical format, every row represents an observation belonging to a particu
lar category.

# for examole:
#   This is a long format:
# Product | Attribute | Value
# A | Height | 10
# A | Width | 5
# A | Weight | 2
# B | Height | 20
# B | Width | 10
#
# The same data is a wide format would be:
# Product | Height | Width | Weight
# A | 10 | 5 | 2
# B | 20 | 10 | NA

#In R, tidyr and dplyr are mostly used for such transformations.
############################
#we can move multiple columns into a single column (making the data long and skinny)
 by "melting" multiple columns.
# airquality_long <- reshape(data=airquality, varying=1:4, v.names="Measure",
#                            timevar="Dimension", times=names(airquality)[1:4],
#                            idvar="Measure ID", direction="long")

# data = dataframe that we want to convert
# varying = columns in the wide format that correspond to a single column in the long
format
# timevar = name of new variable that differentiates multiple observations from the s
ame individual
# idvar = variable in our dataset that identifies multiple records from the same indi
vidual
# direction = “wide” if you’re going from long to wide and “long” if you’re going fro
m wide to long


#########################
#second method
airquality_long <- gather(airquality_clean, variable, value, -Month, -Day)

# key= provides a name for the new variable that is created by gathering together sev
eral
# variables from the previous data frame
# value= provides a name for the new variable that accompanies the keying variable
head(airquality_long)
```

```
##   Month Day variable value
## 1     5   1    Ozone    41
## 2     5   2    Ozone    36
## 3     5   3    Ozone    12
## 4     5   4    Ozone    18
## 5     5   7    Ozone    23
## 6     5   8    Ozone    19
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
p <- ggplot(airquality_long, aes(x = Day, y= value, color= variable, fill=variable))
+ geom_point() +
  geom_line() +
  facet_grid(variable ~ Month, scales = "free")+
  labs(x = "Day", y = "Value")+
  ggtitle("plot all measures in airquality dataset (facet by Day)") +
  theme(text = element_text(size = 9))
p
```



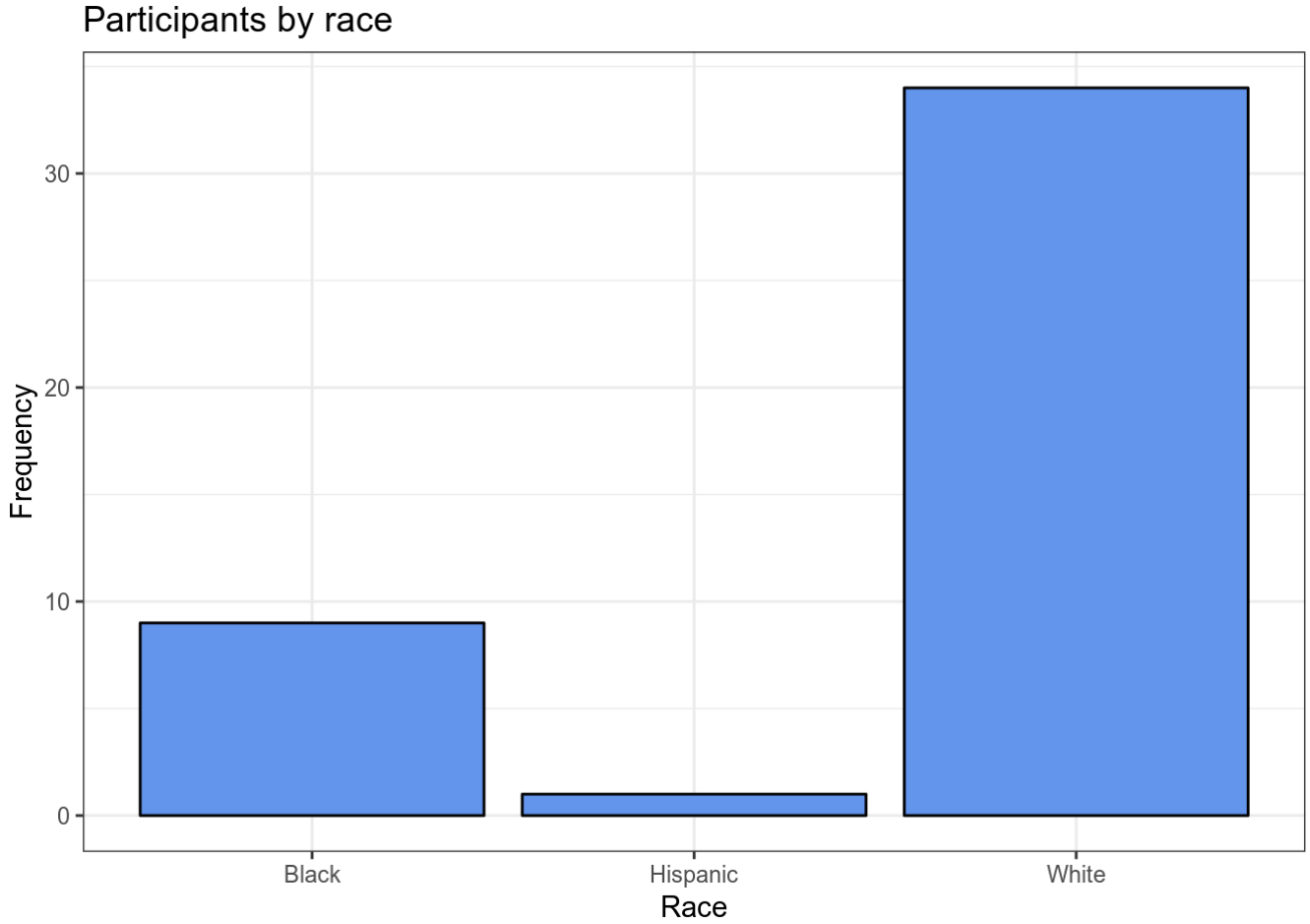

```
##################3333Distributional plot#######################
marriage <- data(Marriage, package = "mosaicData")
head(Marriage)
```

```
##   bookpageID  appdate ceremonydate delay     officialTitle person      dob
## 1   B230p539 10/29/96      11/9/96    11    CIRCUIT JUDGE   Groom  4/11/64
## 2   B230p677 11/12/96     11/12/96     0 MARRIAGE OFFICIAL  Groom   8/6/64
## 3   B230p766 11/19/96     11/27/96     8 MARRIAGE OFFICIAL  Groom  2/20/62
## 4   B230p892  12/2/96      12/7/96     5          MINISTER  Groom  5/20/56
## 5   B230p994  12/9/96     12/14/96     5          MINISTER  Groom 12/14/66
## 6  B230p1209 12/26/96     12/26/96     0 MARRIAGE OFFICIAL  Groom  2/21/70
##        age     race prevcount prevconc hs college dayOfBirth        sign
## 1 32.60274    White         0     <NA> 12       7      102.0       Aries
## 2 32.29041    White         1  Divorce 12       0      219.0         Leo
## 3 34.79178 Hispanic         1  Divorce 12       3       51.5      Pisces
## 4 40.57808    Black         1  Divorce 12       4      141.0      Gemini
## 5 30.02192    White         0     <NA> 12       0      348.5 Saggitarius
## 6 26.86301    White         1     <NA> 12       0       52.5      Pisces
```

```
sum(is.na(Marriage))
```

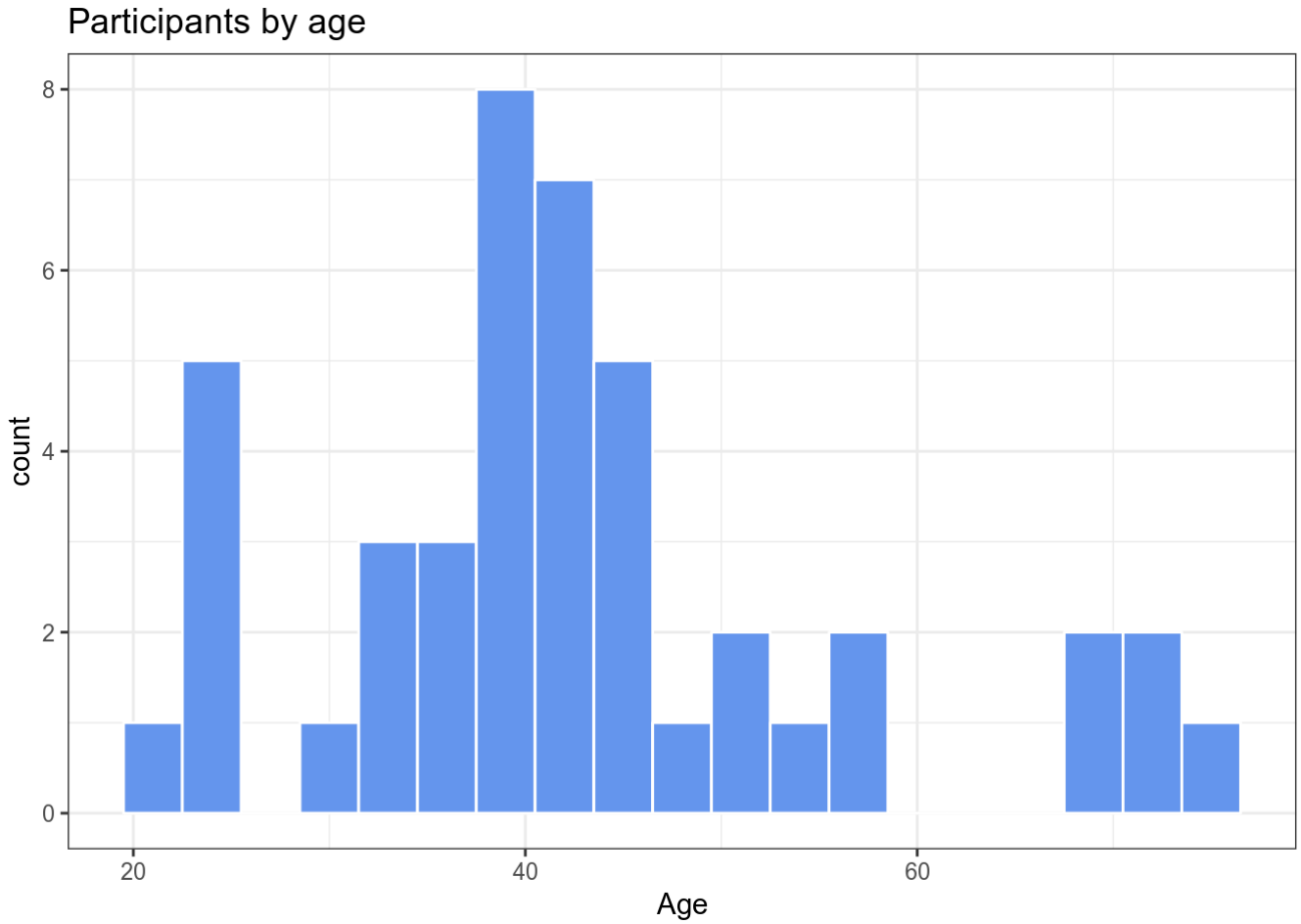
```
## [1] 58
```

```
# plot the distribution of race with modified colors and labels
Marriage <- na.omit(Marriage)
ggplot(Marriage, aes(x = race)) +
  geom_bar(fill = "cornflowerblue",
           color="black") +
  labs(x = "Race",
       y = "Frequency",
       title = "Participants by race")
```



```
#The majority of participants are white, followed by black, with very few Hispanics.
```

```
#plot the ages of the wedding participants
ggplot(Marriage, aes(x = age)) +
  geom_histogram(fill = "cornflowerblue",
                 color = "white", binwidth = 3) +
  labs(title="Participants by age",
       x = "Age")
```




```
#Most participants appear betwen the age of 30 and 60 and a much smaller group in the
ir later seventies.
```

```
#Load dataset:
mammals <- read.csv ("/home/sedreh/ITMO/semester2/Statistic-R/7/mammals.csv")
#view(data)
head(mammals)
```

```
##                  Species   BodyWt BrainWt NonDreaming Dreaming TotalSleep
## 1        Africanelephant 6654.000  5712.0          NA       NA        3.3
## 2 Africangiantpouchedrat    1.000     6.6         6.3      2.0        8.3
## 3              ArcticFox    3.385    44.5          NA       NA       12.5
## 4   Arcticgroundsquirrel    0.920     5.7          NA       NA       16.5
## 5          Asianelephant 2547.000  4603.0         2.1      1.8        3.9
## 6                 Baboon   10.550   179.5         9.1      0.7        9.8
##   LifeSpan Gestation Predation Exposure Danger
## 1     38.6       645         3        5      3
## 2      4.5        42         3        1      3
## 3     14.0        60         1        1      1
## 4       NA        25         5        2      3
## 5     69.0       624         3        5      4
## 6     27.0       180         4        4      4
```

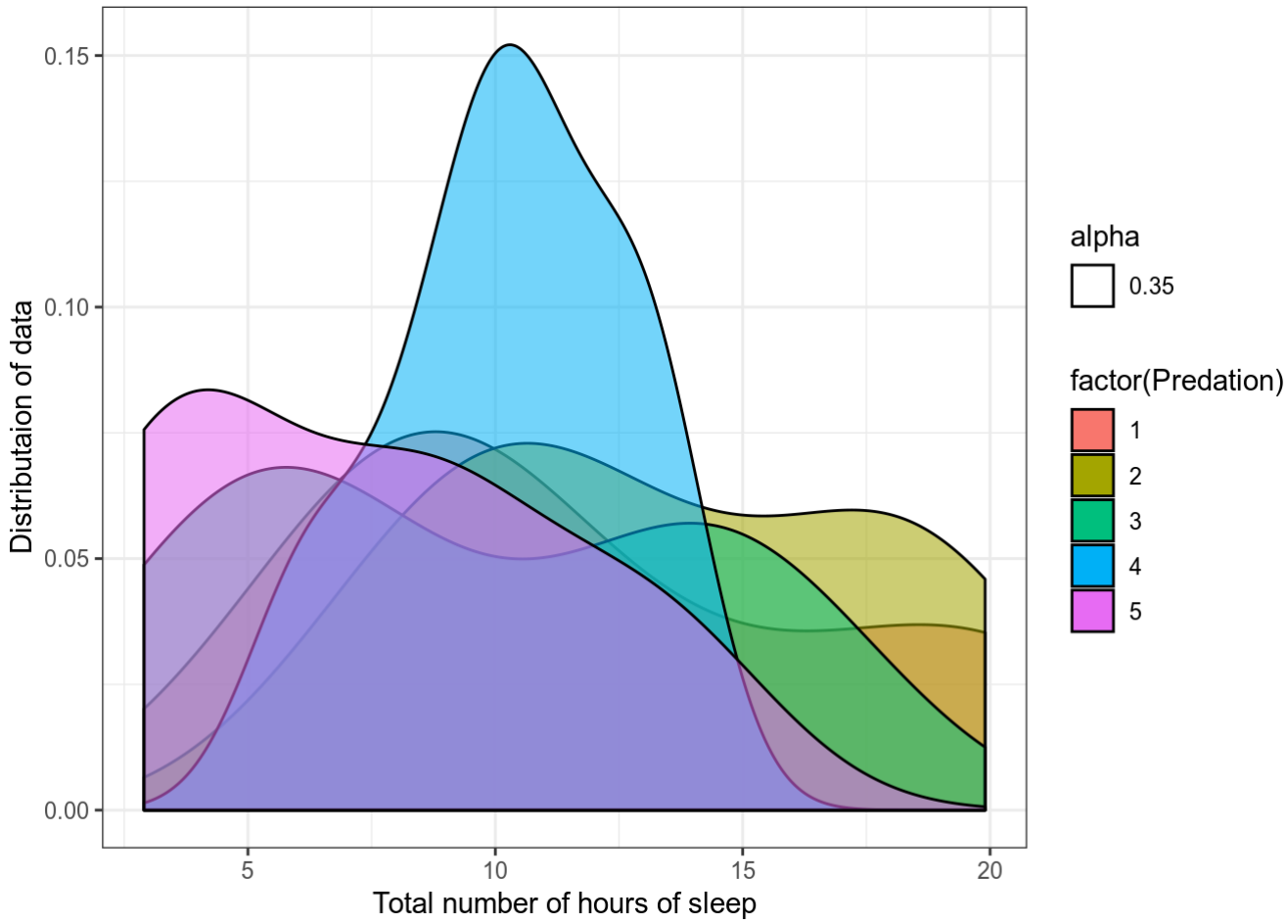
```
sum(is.na(mammals))
```

```
## [1] 38
```

```
mammals <- na.omit(mammals)
p <- ggplot(mammals, aes(TotalSleep)) +
  geom_density(aes( fill = factor(Predation), adjust= 1.5, alpha = 0.35))+
  labs (x = "Total number of hours of sleep", y= "Distributaion of data")+
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: adjust
```

```
p
```




```
#This data set includes data for 39 species of mammals distributed over 13 orders. Th
e data were used for analyzing the relationship between constitutional and ecological
factors and sleeping in mammals. Two qualitatively different sleep variables (dreamin
g and non dreaming) were recorded. Constitutional variables such as life span, body w
eight, brain weight and gestation time were evaluated. Ecological variables such as s
everity of predation, safety of sleeping place and overall danger were inferred from
 field observations in the literature
```

```
#Distributional plot
#Load dataset:
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans  drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(… f        18    29 p     comp…
## 2 audi         a4      1.8  1999     4 manua… f        21    29 p     comp…
## 3 audi         a4      2    2008     4 manua… f        20    31 p     comp…
## 4 audi         a4      2    2008     4 auto(… f        21    30 p     comp…
## 5 audi         a4      2.8  1999     6 auto(… f        16    26 p     comp…
## 6 audi         a4      2.8  1999     6 manua… f        18    26 p     comp…
```

```
sum(is.na(mpg))
```

```
## [1] 0
```

```
#cyl is numric data. we need to change numeric data to a factor.
p <- ggplot(mpg, aes(cty)) +
  geom_density(aes( fill = factor(cyl), adjust= 1.5, alpha = 0.85))+
  labs (title = "City Mileage grouped by number of cylinders", x = "city Milage", y=
"Distributaion of data")+
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: adjust
```

```
p
```