Assignment 2 - Data Wrangling

John DeForest

July 17, 2024

1: Generate and Summarize Random Data

```
library(ggplot2)
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(tidyr)
library(ggpubr)


set.seed(103)
n=10000
#Age = uniform dis between 18 and 35
#InfantSex = 'Female' or 'Male' with equal prob
# Gluscose1 = normal: if InfantSex Male, mean 85 w sd 6, if female, mean 80 w sd 6
# Glucose2 = normal, if infantSex male, mean 165 w sd 9, if female, mean 155 w sd 9
# Diagnosis = 'Gestational Diabetes' if either Glucose1 is>95 or Glucose2 >180, ='Healthy' otherwise
randpreggos = data.frame('SubjectID' = 1:n,
                         'Age' = runif(n, 18, 35),
                         #InfantSex = 'Female' or 'Male' with equal prob
                         'InfantSex' = sample(c('Female', 'Male'), n, replace = TRUE))
randpreggos$Glucose1 = ifelse(randpreggos$InfantSex == 'Male',
                              rnorm(n, mean = 85, sd = 6),
                              rnorm(n, mean = 80, sd = 6))
randpreggos$Glucose2 = ifelse(randpreggos$InfantSex == 'Male',
                              rnorm(n, mean = 165, sd = 9),
                              rnorm(n, mean = 155, sd = 9))
randpreggos$Diagnosis = ifelse(randpreggos$Glucose1 > 95 | randpreggos$Glucose2 > 180,
                               'Gestational Diabetes', 'Healthy')
randpreggos$Diagnosis <- as.factor(randpreggos$Diagnosis)
```

```r
#^enforces conv to factor, so Diagnosis is a factor variable

print(head(randpreggos))
```

```
##   SubjectID      Age InfantSex Glucose1 Glucose2 Diagnosis
## 1         1 21.67101    Female 84.24059 153.6634   Healthy
## 2         2 19.07386      Male 82.86041 159.9009   Healthy
## 3         3 26.87105    Female 77.65629 140.4010   Healthy
## 4         4 26.56149      Male 89.75866 165.4398   Healthy
## 5         5 20.04826    Female 79.60097 163.3738   Healthy
## 6         6 19.48368    Female 90.62183 168.8398   Healthy
```

```r
# Use the summary() function to print summaries of your data for male infants and female infants sepa
#rately (hint: you will need to subset your data to do this) diagnosis should appear summarized as a fa
# variable
summary(randpreggos[randpreggos$InfantSex == 'Male', ])
```

```
##    SubjectID          Age          InfantSex            Glucose1
##  Min.   :   2   Min.   :18.01   Length:5043        Min.   : 66.06
##  1st Qu.:2516   1st Qu.:22.40   Class :character   1st Qu.: 80.84
##  Median :5003   Median :26.66   Mode  :character   Median : 84.89
##  Mean   :5003   Mean   :26.59                      Mean   : 84.95
##  3rd Qu.:7472   3rd Qu.:30.73                      3rd Qu.: 89.12
##  Max.   :9999   Max.   :34.99                      Max.   :109.07
##     Glucose2                   Diagnosis
##  Min.   :131.6   Gestational Diabetes: 456
##  1st Qu.:159.0   Healthy             :4587
##  Median :165.0
##  Mean   :165.1
##  3rd Qu.:171.3
##  Max.   :211.0
```

```r
summary(randpreggos[randpreggos$InfantSex == 'Female', ])
```

```
##    SubjectID          Age          InfantSex            Glucose1
##  Min.   :    1   Min.   :18.00   Length:4957        Min.   : 57.64
##  1st Qu.: 2490   1st Qu.:22.22   Class :character   1st Qu.: 76.01
##  Median : 4997   Median :26.58   Mode  :character   Median : 80.12
##  Mean   : 4998   Mean   :26.55                      Mean   : 80.03
##  3rd Qu.: 7542   3rd Qu.:30.80                      3rd Qu.: 84.07
##  Max.   :10000   Max.   :35.00                      Max.   :100.03
##     Glucose2                   Diagnosis
##  Min.   :122.9   Gestational Diabetes:  32
##  1st Qu.:149.0   Healthy             :4925
##  Median :155.1
##  Mean   :155.1
##  3rd Qu.:161.2
##  Max.   :191.9
```

```r
#Generate a boxplot showing the distribution of Glucose1
#(Fasting Glucose) on the y axis, Diagnosis on
```

```r
#the x axis, and color the plot by InfantSex.
#Define a unique color palette for your boxplot with the colors of
#your choosing.

library(ggplot2)
ggplot(randpreggos, aes(x = Diagnosis, y = Glucose1, fill = InfantSex)) +
  geom_boxplot() +
  scale_fill_manual(values = c("lightblue", "pink")) + #swaggy colors
  labs(
    title = "Distribution of Fasting Glucose by Diagnosis and Infant Sex",
    x = "Diagnosis",     # x-axis label
    y = "Fasting Glucose (mg/dL)",  # y-axis label
  )
```

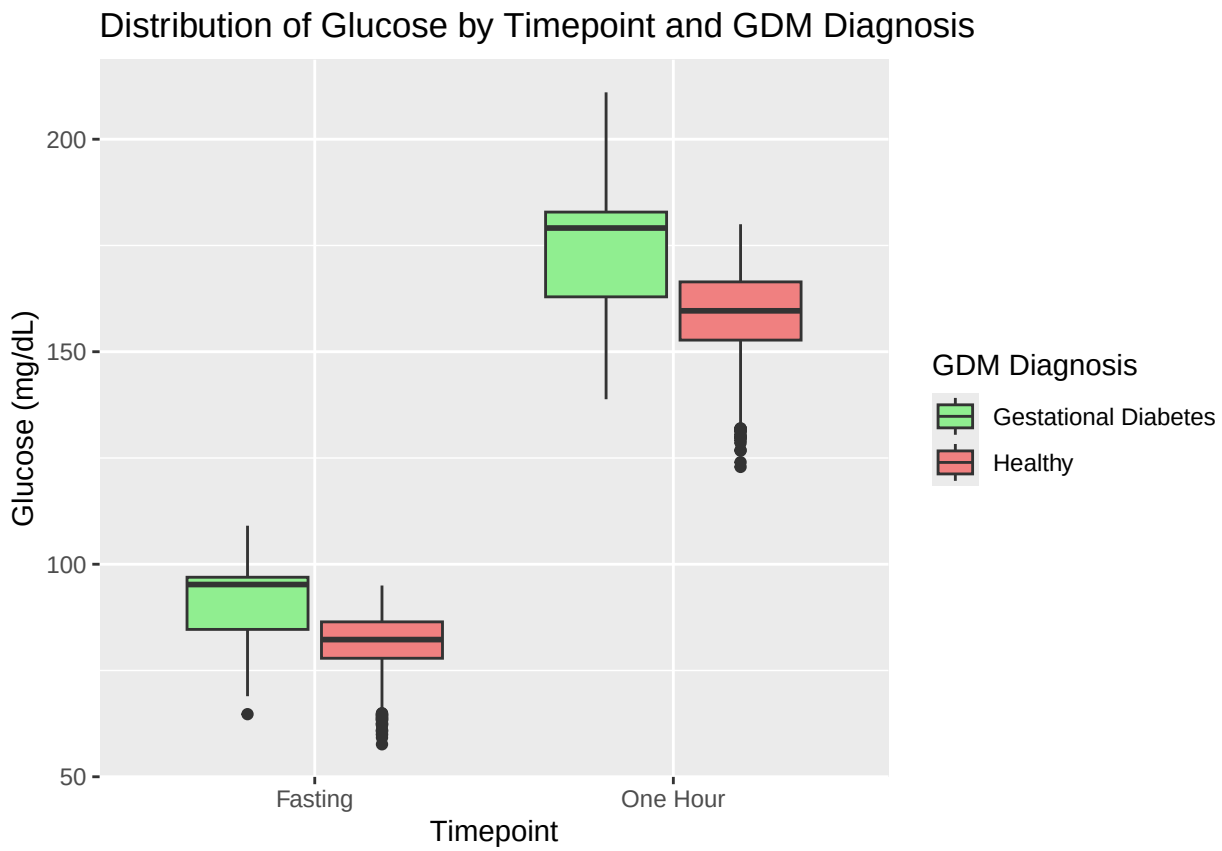
2:Plot The Distribution of Glucose Measures by Timepoint and Diagnosis

```r
#Convert your dataset into a long format data
#frame with a single Glucose variable and a
#separate variable designating Timepoint
#(Fasting or One Hour). Print the entries for
#Subject 1
library(tidyr)
randpreggos_long <- randpreggos %>%
  pivot_longer(cols = c(Glucose1, Glucose2),
               names_to = "Timepoint",
               values_to = "Glucose") %>% #single Glucose variable: (1)
```

```
  dplyr::mutate(Timepoint = ifelse(Timepoint == "Glucose1", "Fasting", "One Hour"))
  #weird mutate not recognized. fixed.
print(randpreggos_long[randpreggos_long$SubjectID == 1, ])
```

```
## # A tibble: 2 x 6
##   SubjectID   Age InfantSex Diagnosis Timepoint Glucose
##       <int> <dbl> <chr>     <fct>     <chr>       <dbl>
## 1         1  21.7 Female    Healthy   Fasting      84.2
## 2         1  21.7 Female    Healthy   One Hour    154.
```

```
#Generate a boxplot of the distribution of
#Glucose (y axis) for all subjects where
#Timepoint is on the x-axis and the plot is
#colored by Diagnosis. Set a color palette that
#is unique from the color scheme used in Part 1.
#Define axis titles, x axis labels, and legend
# title using ggplot2:
# x axis title: "Timepoint",
# x axis labels: "Baseline" and "One Hour",
#y axis title: "Glucose (mg/dL)",
#legend title: "GDM Diagnosis"
ggplot(randpreggos_long, aes(x = Timepoint, y = Glucose, fill = Diagnosis)) +
  geom_boxplot() +
  scale_fill_manual(values = c("lightgreen", "lightcoral")) + #new fire colors lesgo
  labs(
    title = "Distribution of Glucose by Timepoint and GDM Diagnosis",
    x = "Timepoint",  # x-axis label
    y = "Glucose (mg/dL)",  # y-axis label
    fill = "GDM Diagnosis"  # legend title
  ) +
  scale_x_discrete(labels = c("Fasting", "One Hour"))
```

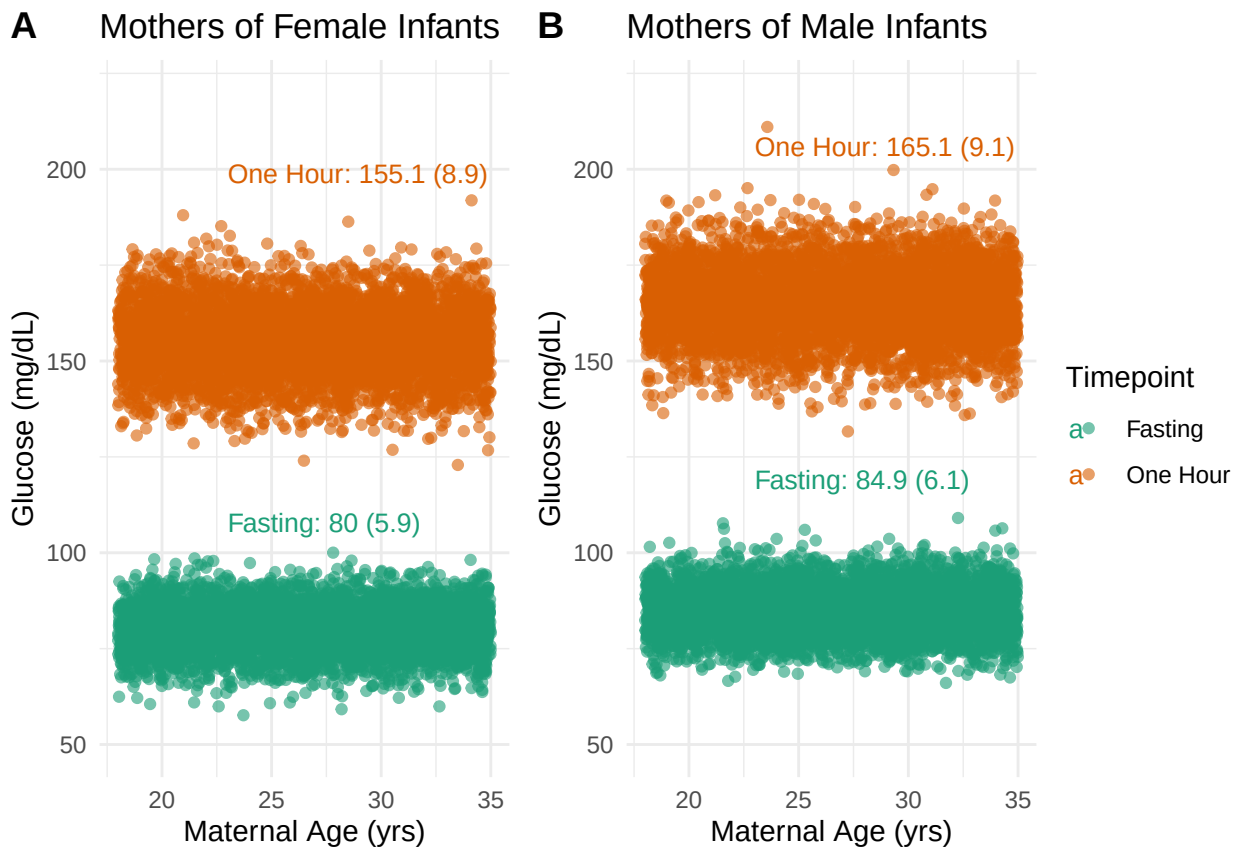


3: Plot the Distribution of Glucose Measures by Maternal Age

```
#Generate a scatter plot with maternal age plotted
#on the x axis (labeled "Maternal Age (yrs)")
#and glucose on the y axis
#(labeled "Glucose (mg/dL)". Color points by
#Time point (Baseline or One Hour) and
#define a new color palette than those used in
#parts 1 and 2. Separate into two plots by infant sex. For each
#plot, add text with the
#mean (standard deviation) fasting and 1 hr
#glucose in the same colors used for the color
#palette.
#Define the location of the text such
#that it is legible does not overlap with any
#points (note: this should be plotted as an
#annotation, not a plot title). Figures should
#have titles identifying "Mothers of Female
#Infants" and "Mothers of Male Infants"
#respectively. Finally, use the ggarrange()
#function in ggpubr or a similar function for
#arranging multiple plots of your choosing to
#produce a single figure with both plots
#adjacent, labeled as figures A and B, respectively.
#ggplot(randpreggos)
```

```r
#library(ggplot2)
#library(dplyr)
#library(ggpubr)

#new color palette woot woot
new_colors <- c("Fasting" = "#1b9e77", "One Hour" = "#d95f02")  # green-orange type shi ong

#stats mean/sd annotatin' for plots
annot_stats <- randpreggos_long %>%
  group_by(InfantSex, Timepoint) %>%
  dplyr::summarize(
    mean_glucose = round(mean(Glucose), 1),
    sd_glucose = round(sd(Glucose), 1),
    .groups = "drop"
  ) %>%
  dplyr::mutate( #mean (sd) annot for ea timepoint
    label = paste0(Timepoint, ": ", mean_glucose, " (", sd_glucose, ")")
  )

#split M/F data into sep dataframes via filter
female_data <- randpreggos_long %>% filter(InfantSex == "Female")
male_data   <- randpreggos_long %>% filter(InfantSex == "Male")

#^ea annotations no overlap allow'd
female_annot <- annot_stats %>% filter(InfantSex == "Female") %>%
  dplyr::mutate(x = 23, y = ifelse(Timepoint == "Fasting", 108, 199))
  #TODO test x/y vals for no overlap

male_annot <- annot_stats %>% filter(InfantSex == "Male") %>%
  dplyr::mutate(x = 23, y = ifelse(Timepoint == "Fasting", 119, 206))

#plots separated by infantSex:
#FEMALE PLOT
plot_female <- ggplot(female_data, aes(x = Age, y = Glucose, color = Timepoint)) +
  geom_point(alpha = 0.6) +
  scale_color_manual(values = new_colors) +
  labs(
    title = "Mothers of Female Infants",
    x = "Maternal Age (yrs)",
    y = "Glucose (mg/dL)",
    color = "Timepoint"
  ) +
  #TODO check x/y vals for no overlap
  geom_text(data = female_annot, aes(x = x, y = y, label = label, color = Timepoint),
            hjust = 0, inherit.aes = FALSE, size = 3.5) +
  theme_minimal()+
  coord_cartesian(ylim = c(50, 220)) #to align w MALE plot y axis


#now MALE
plot_male <- ggplot(male_data, aes(x = Age, y = Glucose, color = Timepoint)) +
  geom_point(alpha = 0.6) +
  scale_color_manual(values = new_colors) +
```

```
  labs(
    title = "Mothers of Male Infants",
    x = "Maternal Age (yrs)",
    y = "Glucose (mg/dL)",
    color = "Timepoint"
  ) +
  geom_text(data = male_annot, aes(x = x, y = y, label = label, color = Timepoint),
            hjust = 0, inherit.aes = FALSE, size = 3.5) +
  theme_minimal() +
  coord_cartesian(ylim = c(50, 220)) #for alignment of y axis vals w female

#arrange plots side-o w ggarrange/ A/B labels
final_plot <- ggarrange(plot_female, plot_male,
                        labels = c("A", "B"),
                        ncol = 2,
                        common.legend = TRUE,
                        legend = "right")
print(final_plot) #show dat shi'
```



4:Generate Wide Table of Summary Statistics

```
#Generate a wide format table summarizing the age, fasting glucose,
#and one hour glucose of all subjects by both disease status and infant sex.
#Your table should have 4 rows in the following order:
#Healthy & Female,Gestational Diabetes & Female,
#Healthy & Male, Gestational Diabetes & Male
```

```r
#and should summarize mean age,
#mean and sd fasting glucose, and mean and sd one hour glucose.

#library(dplyr)

# Force factor levels to control order
randpreggos_ordered <- randpreggos_long %>%
  dplyr::mutate(
    Diagnosis = factor(Diagnosis, levels = c("Healthy", "Gestational Diabetes")),
    InfantSex = factor(InfantSex, levels = c("Female", "Male"))
  ) %>%
  group_by(Diagnosis,InfantSex) %>% #first sort column is diagnosis, then infantSex (but doesnt enforce
  dplyr::summarize(
    mean_age = round(mean(Age), 1),
    mean_glucose1 = round(mean(Glucose[Timepoint == "Fasting"]), 2),
    sd_glucose1 = round(sd(Glucose[Timepoint == "Fasting"]), 1),
    mean_glucose2 = round(mean(Glucose[Timepoint == "One Hour"]), 2),
    sd_glucose2 = round(sd(Glucose[Timepoint == "One Hour"]), 1),
    .groups = "drop"
  ) %>%
  arrange(InfantSex, Diagnosis) # first sort top by infantSex Female, then by diagnosis

print(randpreggos_ordered,width=Inf) #would cut off last col
```

```
## # A tibble: 4 x 7
##   Diagnosis            InfantSex mean_age mean_glucose1 sd_glucose1
##   <fct>                <fct>        <dbl>         <dbl>       <dbl>
## 1 Healthy              Female        26.6          80.0         5.8
## 2 Gestational Diabetes Female        25.2          91.5         9.1
## 3 Healthy              Male          26.6          84.3         5.5
## 4 Gestational Diabetes Male          26.9          91.3         7.7
##   mean_glucose2 sd_glucose2
##           <dbl>       <dbl>
## 1          155.         8.8
## 2          163.        15.3
## 3          164.         8.2
## 4          174.        12
```