KoreanHealth

2024-03-09

Loading in Data

```r
# Tilde represents user's home directory
setwd("~/Documents/GitHub/GEE_lifestyleEffectsOnHypertension")

dat_01 <- read.csv(file = "KoreanHealthRecords/follow_01_data.csv")
dat_02 <- read.csv(file = "KoreanHealthRecords/follow_02_data.csv")
dat_03 <- read.csv(file = "KoreanHealthRecords/follow_03_data.csv")
dat_04 <- read.csv(file = "KoreanHealthRecords/follow_04_data.csv")
dat_05 <- read.csv(file = "KoreanHealthRecords/follow_05_data.csv")
```

Removing column name prefixes and merging csv files

```r
# Replace everything before first underscore with empty string
names(dat_01) <- sub(".*?_", "", names(dat_01))
names(dat_02) <- sub(".*?_", "", names(dat_02))
names(dat_03) <- sub(".*?_", "", names(dat_03))
names(dat_04) <- sub(".*?_", "", names(dat_04))
names(dat_05) <- sub(".*?_", "", names(dat_05))

# bind_rows automatically matches columns by name
merged_df <- bind_rows(dat_01, dat_02, dat_03, dat_04, dat_05)

# Spot check merge, choose random id/row and compare to original excel sheet data
# merged_df[28, ] #follow_02 file. correct.
# merged_df[2029, ] #follow_03 file. correct.
# merged_df[3145, ] #follow_04 file. correct.
```

Save merged dataset to csv

```r
write.csv(merged_df, 'merged_df.csv')

# Cleaning up environment
rm(dat_01)
rm(dat_02)
rm(dat_03)
rm(dat_04)
rm(dat_05)
```

Cleaning data

```r
clean_df <- merged_df

# Change values in SMOKE
clean_df <- clean_df %>%
  mutate(SMOKE = case_when(
    SMOKE %in% c(66666, 77777, 99999) ~ NA_real_,
    TRUE ~ SMOKE
  ))

# Change values in DRINK
clean_df <- clean_df %>%
  mutate(DRINK = case_when(
    DRINK %in% c(66666, 77777, 99999) ~ NA_real_,
    TRUE ~ DRINK
  ))

# Change values in EXER
clean_df <- clean_df %>%
  mutate(EXER = case_when(
    EXER == 1 ~ 0, # no exercise
    EXER == 2 ~ 1, # yes exercise
    EXER %in% c(66666, 77777, 99999) ~ NA_real_
  ))

# Change values in HTN
clean_df <- clean_df %>%
  mutate(HTN = case_when(
    HTN == 1 ~ 0,
    HTN == 2 ~ 1,
    HTN %in% c(66666, 77777, 99999) ~ NA_real_
  ))


# Change values in PULSE
clean_df <- clean_df %>%
  mutate(PULSE = case_when(
  PULSE %in% c(66666, 77777, 99999) ~ NA_real_,
  TRUE ~ PULSE
  ))

# Change values in SBP
clean_df <- clean_df %>%
  mutate(SBP = case_when(
  SBP %in% c(66666, 77777, 99999) ~ NA_real_,
  TRUE ~ SBP
  ))


# Create categorical proxy for SBP
clean_df <- clean_df %>%
  mutate(SBP_CAT = case_when(
    SBP <= 119 ~ "Healthy",
    SBP >= 120 & SBP <= 139 ~ "Pre-hypertension",
    SBP >= 140 ~ "Hypertension"
  ))

# Create categorical proxy for AGE
# Divisions from incremental increase in hypertension from CDC
clean_df <- clean_df %>%
  mutate(AGE_CAT = case_when(
    AGE < 18 ~ "Children under 18",
    AGE >= 18 & AGE <= 39 ~ "Adults 18 to 39",
    AGE >= 40 & AGE <= 59 ~ "Adults 40-59",
    AGE >= 60 ~ "Adults over 60"
  ))


# Change values in EDATE into Date datatype
clean_df <- clean_df %>%
  mutate(EDATE = as.Date(paste0(EDATE, "01"), format = "%Y%m%d"
  ))

# Clean ID values for geeglm function
clean_df <- clean_df %>%
  mutate(ID = sub(".*?_.*?_", "", ID))

# Change DRINK to categorical data
clean_df$DRINK <- factor(clean_df$DRINK, levels = c(
  '1','2','3'
))

# Change SMOKE to categorical data
clean_df$SMOKE <- factor(clean_df$SMOKE, levels = c(
  '1','2','3'
))

# Change EXER to categorical data, for graphing
clean_df$EXER <- factor(clean_df$EXER, levels = c(
  '0','1'
))

# Change SBP_CAT to categorical data
clean_df$SBP_CAT <- factor(clean_df$SBP_CAT, levels = c(
  'Healthy','Pre-hypertension','Hypertension'
))


# Sort by ID and then EDATE
clean_df <- clean_df[
  with(clean_df, order(ID, EDATE)),
]



# Must run after df is sorted:
```

```
# SEX - Fill in missing data within same ID based on first value
clean_df <- clean_df %>%
  group_by(ID) %>%
  fill(SEX, .direction = 'down') %>%
  ungroup

# EDU - Fill in missing data within same ID based on first value
clean_df <- clean_df %>%
  group_by(ID) %>%
  fill(EDU, .direction = 'down') %>%
  ungroup

# EDU - combine 2 bachelors categories and clean EDU
clean_df <- clean_df %>%
  mutate(EDU = case_when(
    EDU %in% c(66666, 77777, 99999) ~ NA_real_,
    EDU == 5 ~ 4,
    TRUE ~ EDU
  ))
clean_df <- clean_df %>%
  mutate(EDU = case_when(
    EDU == 6 ~ 5,
    TRUE ~ EDU
  ))

# Create year proxy for EDATE, for spaghetti plot
clean_df$YEAR <- clean_df$EDATE
clean_df$YEAR <- format(clean_df$YEAR, format="%Y")

# Change EDU to categorical data, for graphing
# 1 is male
clean_df$EDU <- factor(clean_df$EDU, levels = c(
  '1','2', '3', '4', '5'
))

# Change SEX to categorical data, for graphing
# 1 is male
clean_df$SEX <- factor(clean_df$SEX, levels = c(
  '1','2'
))
```

Subset dataframe

```
# Create a new subset dataframe to run through the model
model_df <- subset(clean_df, select = c(
  'ID', 'EDATE', 'YEAR', 'DRINK', 'SMOKE', 'EXER', 'SBP', 'SBP_CAT', 'AGE', 'AGE_CAT', 'SEX', 'EDU', 'PU
))

# Exclude observations where any field contains NA
model_df <- na.omit(model_df)

# Checking frequencies of each value in each column
# col_names <- c('DRINK', 'SMOKE', 'EXER', 'SBP')
# lapply(model_df[col_names], function(x) table(x, useNA = "ifany"))
```

Save model dataset to csv

```
write.csv(model_df, 'model_df.csv')
```

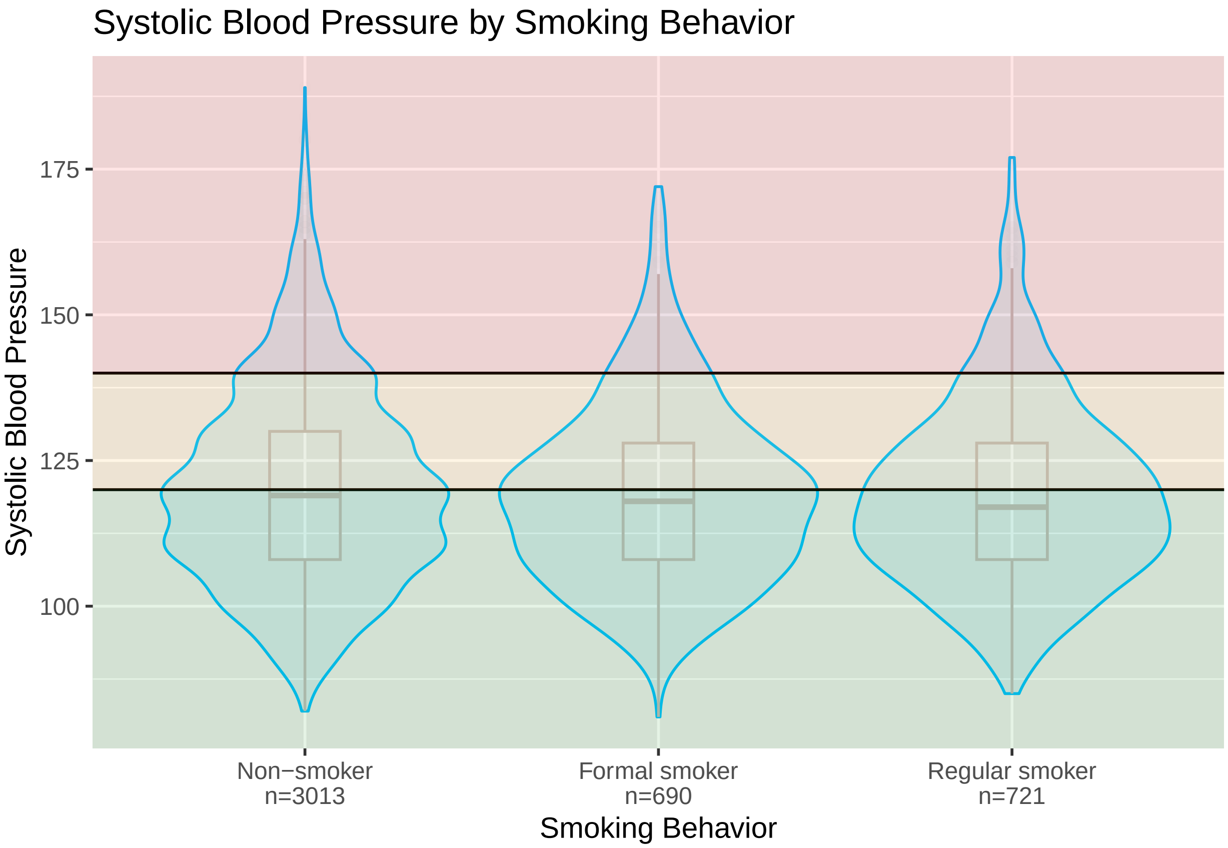
EDA and viz

Violin plots of SBP over SMOKE, DRINK, EXER

Violin plots describe distribution of the data when data set gets too large for a jitter option to represent clearly.

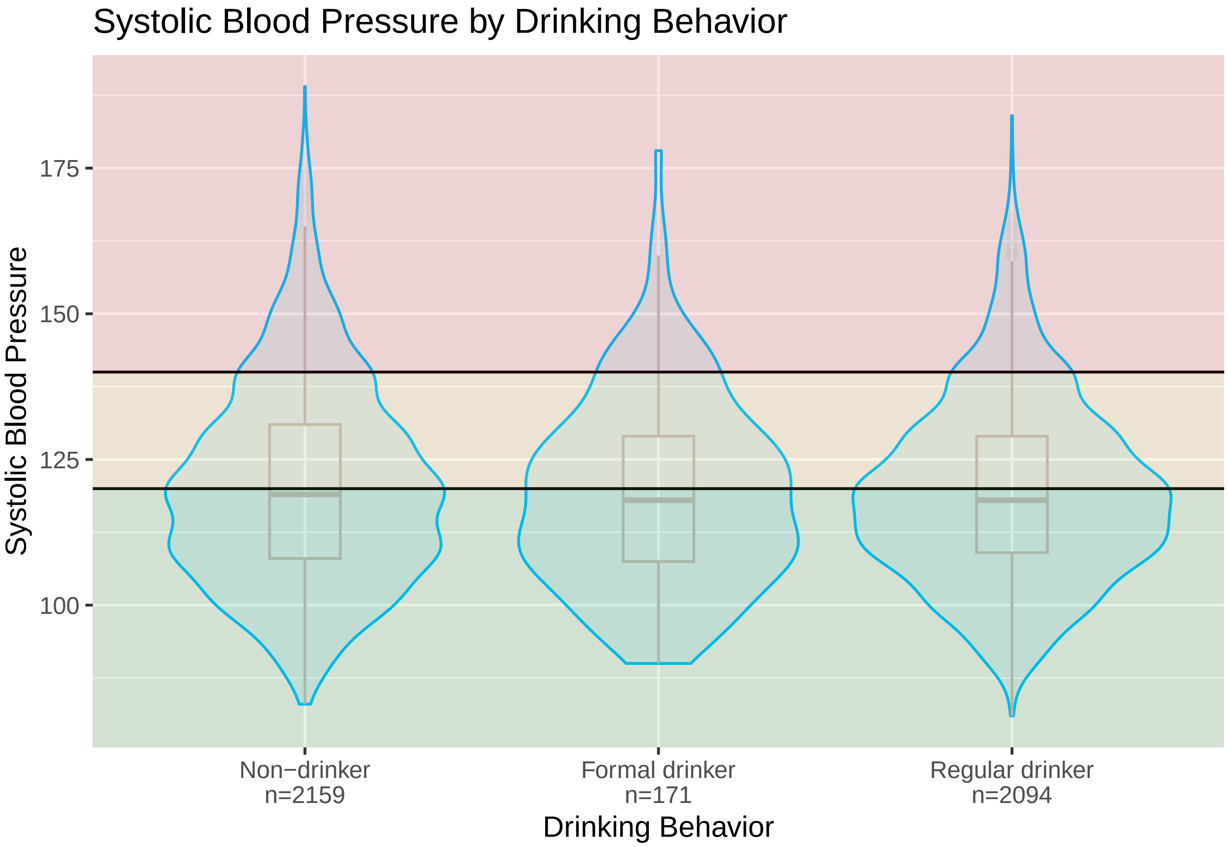
```
# Calculating n for SMOKE
smoke_sum <- model_df %>%
  group_by(SMOKE) %>%
  tally()

# Calculating n for DRINK
drink_sum <- model_df %>%
  group_by(DRINK) %>%
  tally()

# Calculating n for EXER
exer_sum <- model_df %>%
  group_by(EXER) %>%
  tally()

# Calculating n for SEX
sex_sum <- model_df %>%
  group_by(SEX) %>%
  tally()

# Calculating n for EDU
edu_sum <- model_df %>%
  group_by(EDU) %>%
  tally()

# Calculating n for AGE_CAT
age_sum <- model_df %>%
  group_by(AGE_CAT) %>%
  tally()

# set cutoff lines for SBP
sbp_cutoff <- data.frame(yintercept=c(120, 140), Lines=c('Healthy', 'At Risk'))

# Violin plot of SBP over SMOKE
sbp_smoke_v <- ggplot(model_df, aes(x = SMOKE, y = SBP)) +
    geom_violin(color="deepskyblue", fill='deepskyblue', alpha=.09) +
    geom_boxplot(width=0.2, color='grey', alpha=.02) +
    scale_x_discrete(labels = c(
      paste0('Non-smoker', '\n', 'n=', smoke_sum[1,2]),
      paste0('Formal smoker', '\n', 'n=', smoke_sum[2,2]),
      paste0('Regular smoker', '\n', 'n=', smoke_sum[3,2])
      )) +
    labs(x = 'Smoking Behavior', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Smoking Behavior")

sbp_smoke_v + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



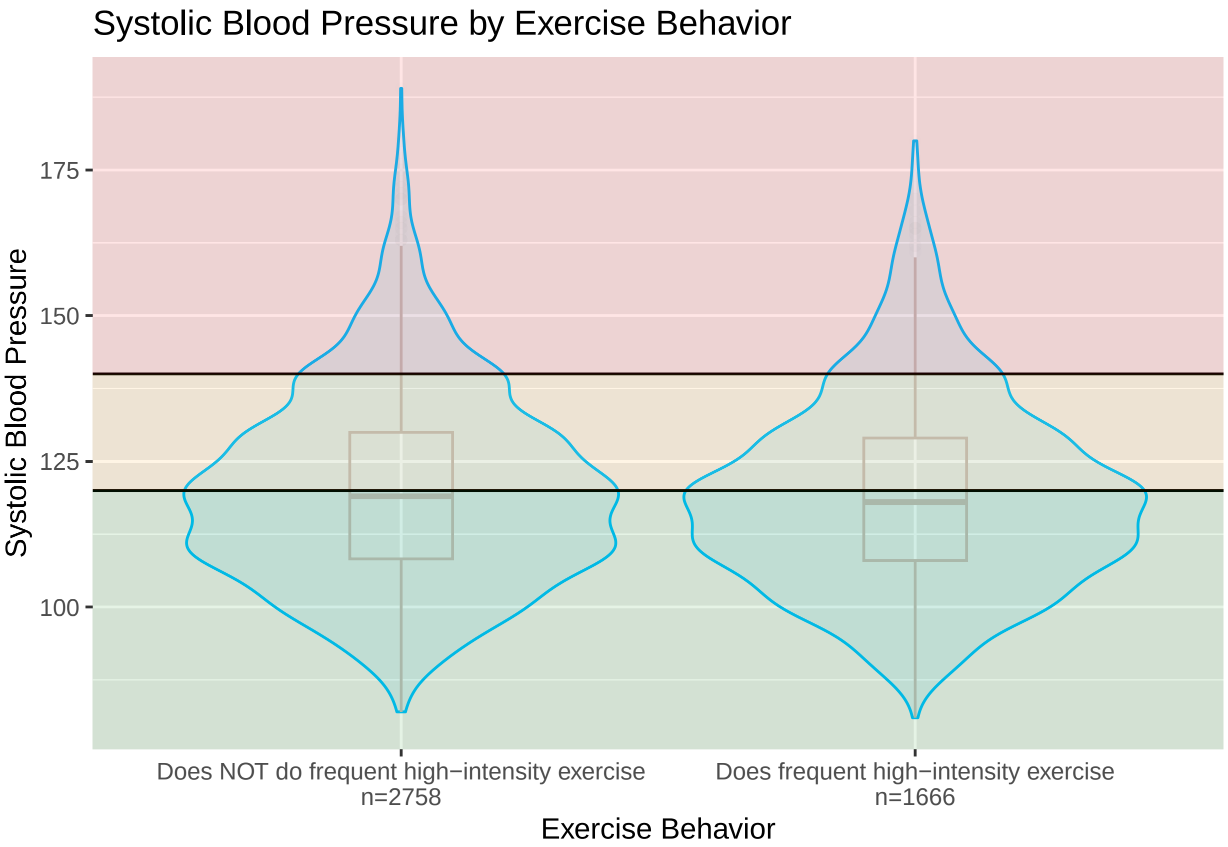
```
# Violin plot of SBP over DRINK
sbp_drink_v <- ggplot(model_df, aes(x = DRINK, y = SBP)) +
    geom_violin(color="deepskyblue", fill='deepskyblue', alpha=.09) +
    geom_boxplot(width=0.2, color='grey', alpha=.02) +
    scale_x_discrete(labels = c(
      paste0('Non-drinker', '\n', 'n=', drink_sum[1,2]),
      paste0('Formal drinker', '\n', 'n=', drink_sum[2,2]),
      paste0('Regular drinker', '\n', 'n=', drink_sum[3,2])
      )) +
    labs(x = 'Drinking Behavior', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Drinking Behavior")

sbp_drink_v + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
          ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



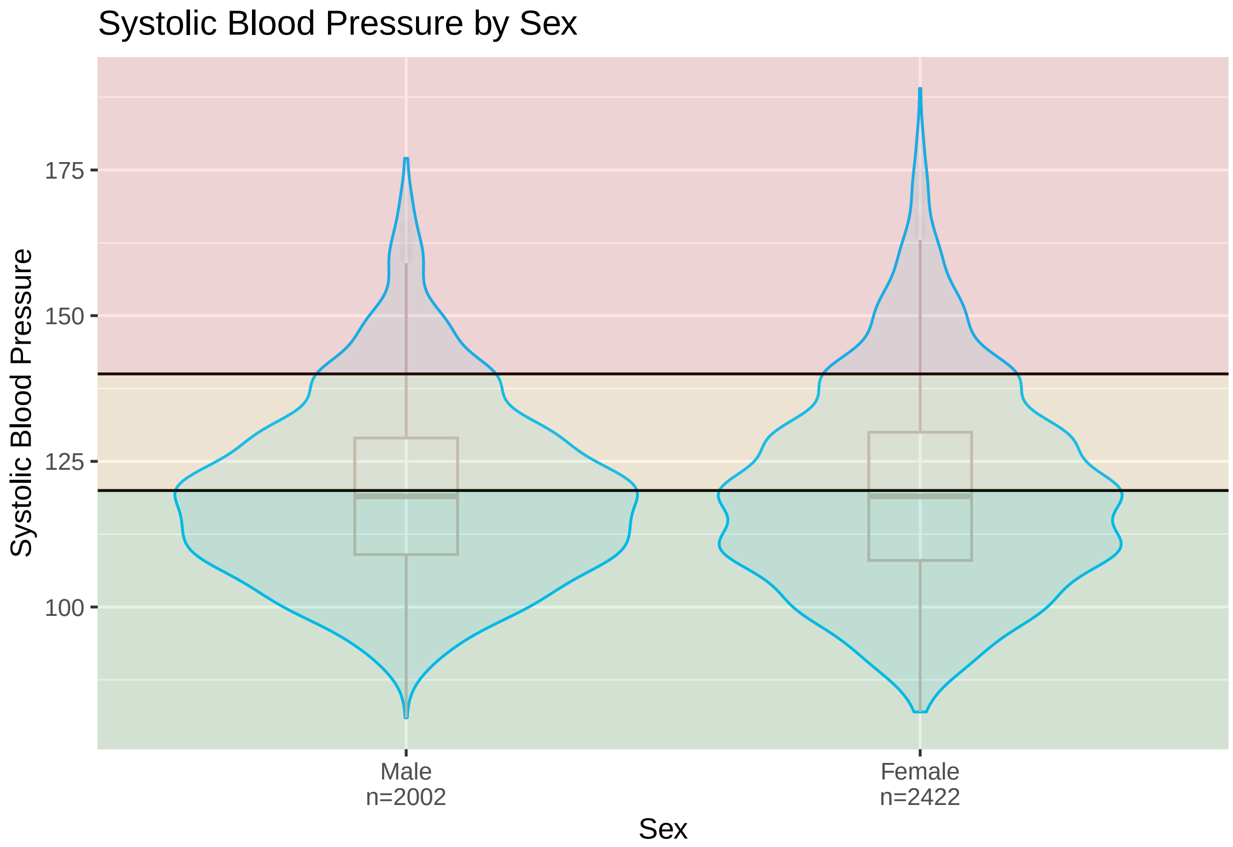
```
# Violin plot of SBP over EXER
sbp_exer_v <- ggplot(model_df, aes(x = EXER, y = SBP)) +
    geom_violin(color="deepskyblue", fill='deepskyblue', alpha=.09) +
    geom_boxplot(width=0.2, color='grey', alpha=.02) +
    scale_x_discrete(labels = c(
      paste0('Does NOT do frequent high-intensity exercise', '\n', 'n=', exer_sum[1,2]),
      paste0('Does frequent high-intensity exercise', '\n', 'n=', exer_sum[2,2])
      )) +
    labs(x = 'Exercise Behavior', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Exercise Behavior")

sbp_exer_v + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
          ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



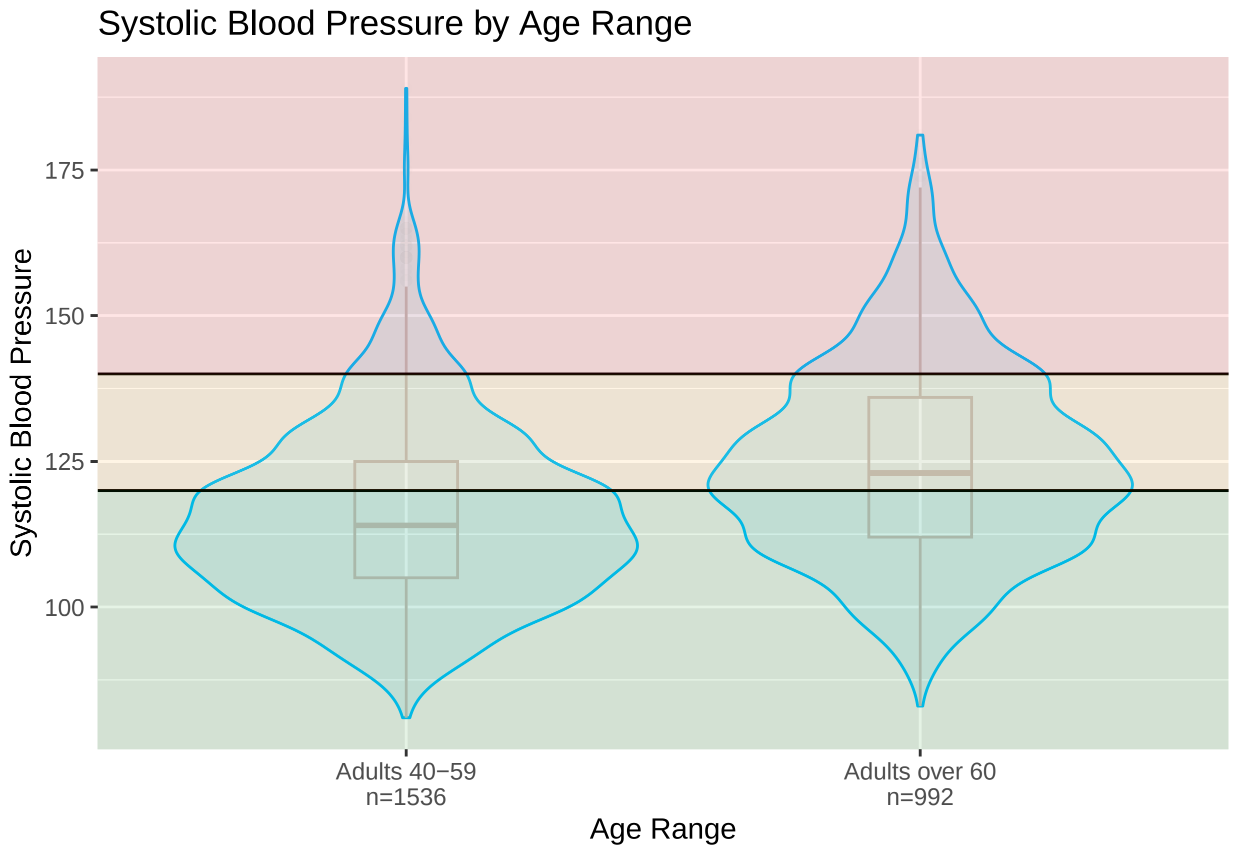
```
# Violin plot of SBP over SEX
sbp_sex_v <- ggplot(model_df, aes(x = SEX, y = SBP)) +
    geom_violin(color="deepskyblue", fill='deepskyblue', alpha=.09) +
    geom_boxplot(width=0.2, color='grey', alpha=.02) +
    scale_x_discrete(labels = c(
      paste0('Male', '\n', 'n=', sex_sum[1,2]),
      paste0('Female', '\n', 'n=', sex_sum[2,2])
      )) +
    labs(x = 'Sex', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Sex")

sbp_sex_v + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



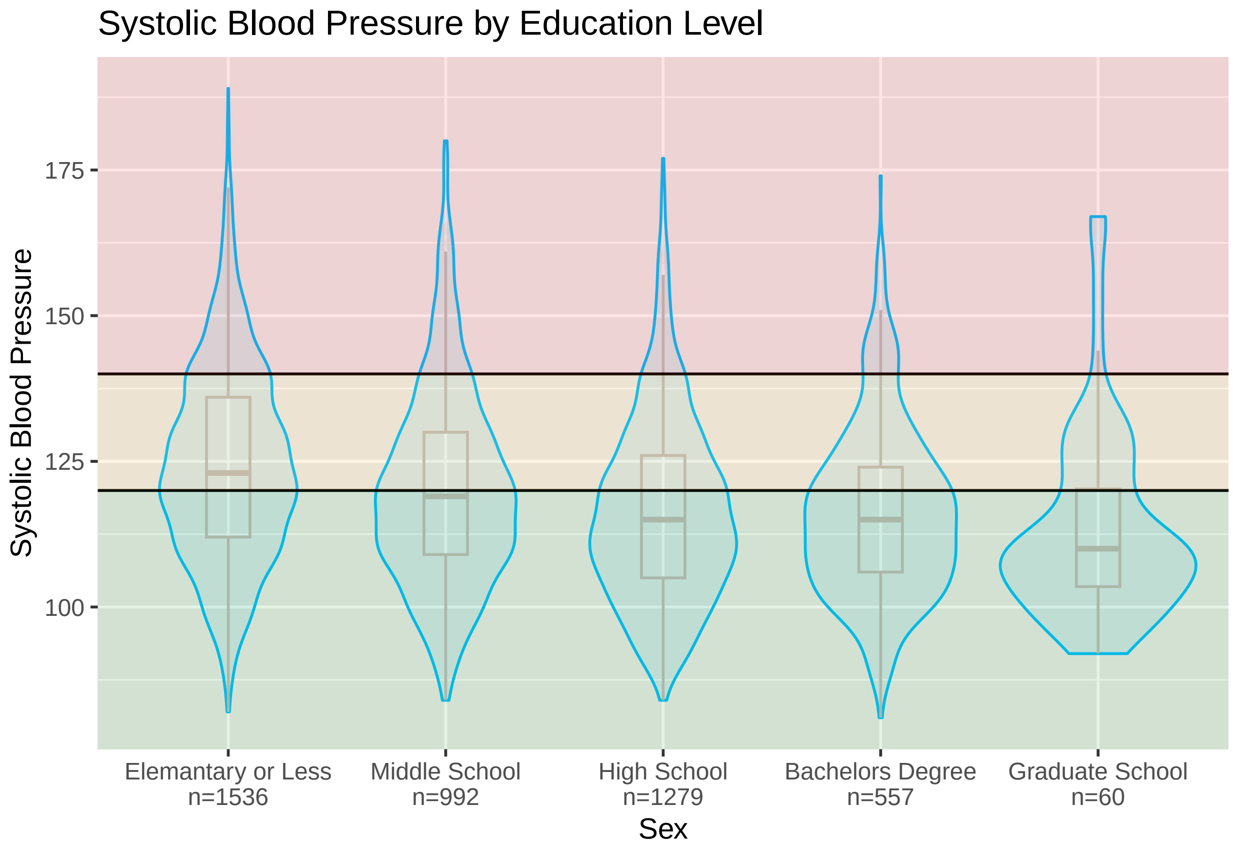
```
# Violin plot of SBP over AGE
sbp_age_v <- ggplot(model_df, aes(x = AGE_CAT, y = SBP)) +
    geom_violin(color="deepskyblue", fill='deepskyblue', alpha=.09) +
    geom_boxplot(width=0.2, color='grey', alpha=.02) +
    scale_x_discrete(labels = c(
      paste0('Adults 40-59', '\n', 'n=', edu_sum[1,2]),
      paste0('Adults over 60', '\n', 'n=', edu_sum[2,2])
      )) +
    labs(x = 'Age Range', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Age Range")

sbp_age_v + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



```
# Violin plot of SBP over EDU
sbp_edu_v <- ggplot(model_df, aes(x = EDU, y = SBP)) +
    geom_violin(color="deepskyblue", fill='deepskyblue', alpha=.09) +
    geom_boxplot(width=0.2, color='grey', alpha=.02) +
    scale_x_discrete(labels = c(
      paste0('Elemantary or Less', '\n', 'n=', edu_sum[1,2]),
      paste0('Middle School', '\n', 'n=', edu_sum[2,2]),
      paste0('High School', '\n', 'n=', edu_sum[3,2]),
      paste0('Bachelors Degree', '\n', 'n=', edu_sum[4,2]),
      paste0('Graduate School', '\n', 'n=', edu_sum[5,2])
      )) +
    labs(x = 'Sex', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Education Level")

sbp_edu_v + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

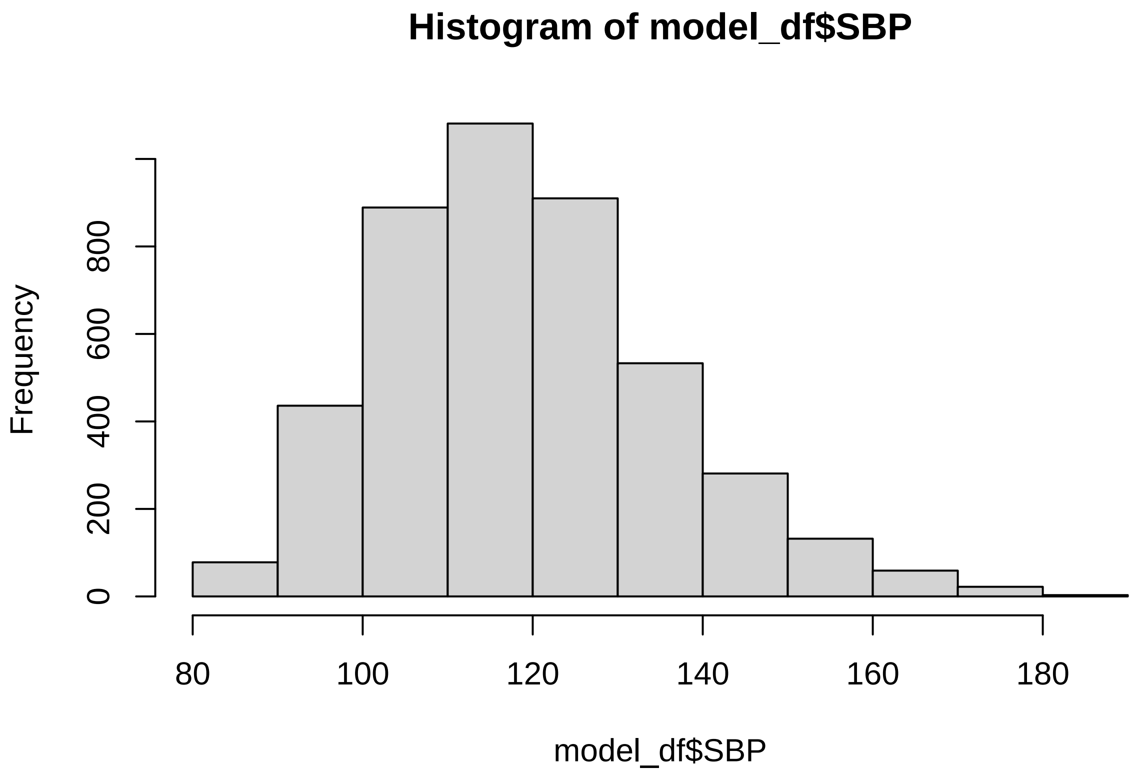
```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



```r
# SBP Median, Mean, Quartiles of Categorical SMOKE
smoke_summary <- aggregate(SBP~SMOKE, data=model_df, summary)

# SBP Median, Mean, Quartiles of Categorical DRINK
drink_summary <- aggregate(SBP~DRINK, data=model_df, summary)

# SBP Median, Mean, Quartiles of Categorical EXER
exer_summary <- aggregate(SBP~EXER, data=model_df, summary)

# SBP Median, Mean, Quartiles of Categorical AGE
age_summary <- aggregate(SBP~AGE_CAT, data=model_df, summary)

# SBP Median, Mean, Quartiles of Categorical SEX
sex_summary <- aggregate(SBP~SEX, data=model_df, summary)

# SBP Median, Mean, Quartiles of Categorical EDU
edu_summary <- aggregate(SBP~DRINK, data=model_df, summary)
```

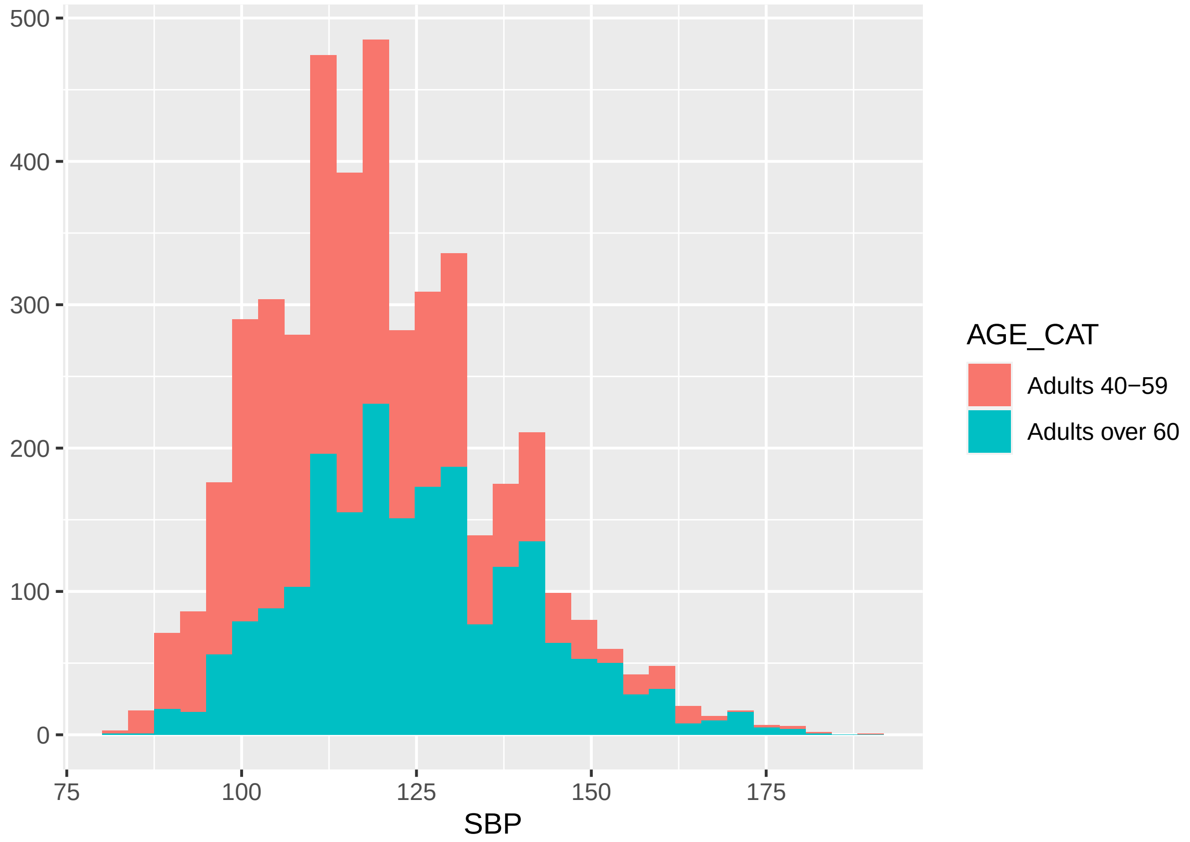
Histograms of SBP over SMOKE, DRINK, EXER

```r
# Histogram of SBP
hist(model_df$SBP)
```



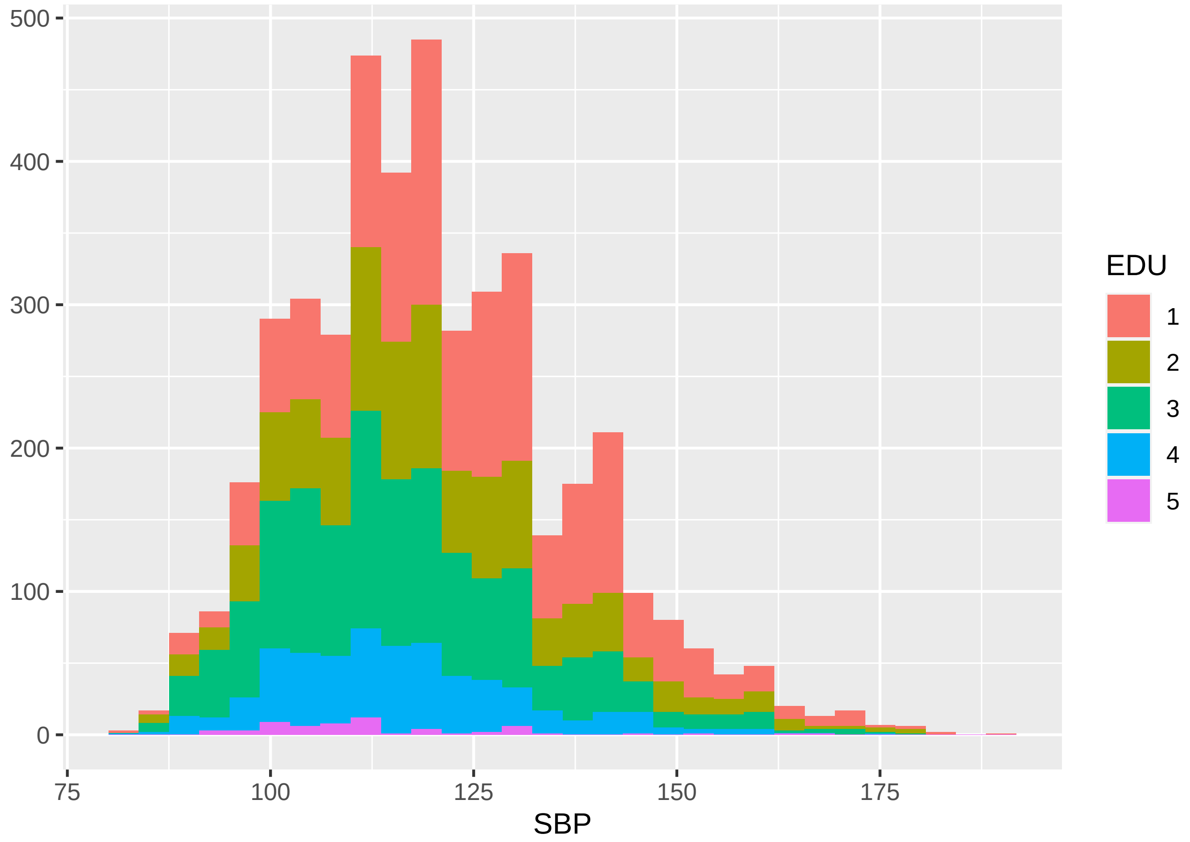
```
# Histogram of SBP by AGE_CAT
qplot(x=SBP,
      fill=AGE_CAT,
      data=model_df,
      geom = c('histogram'))
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



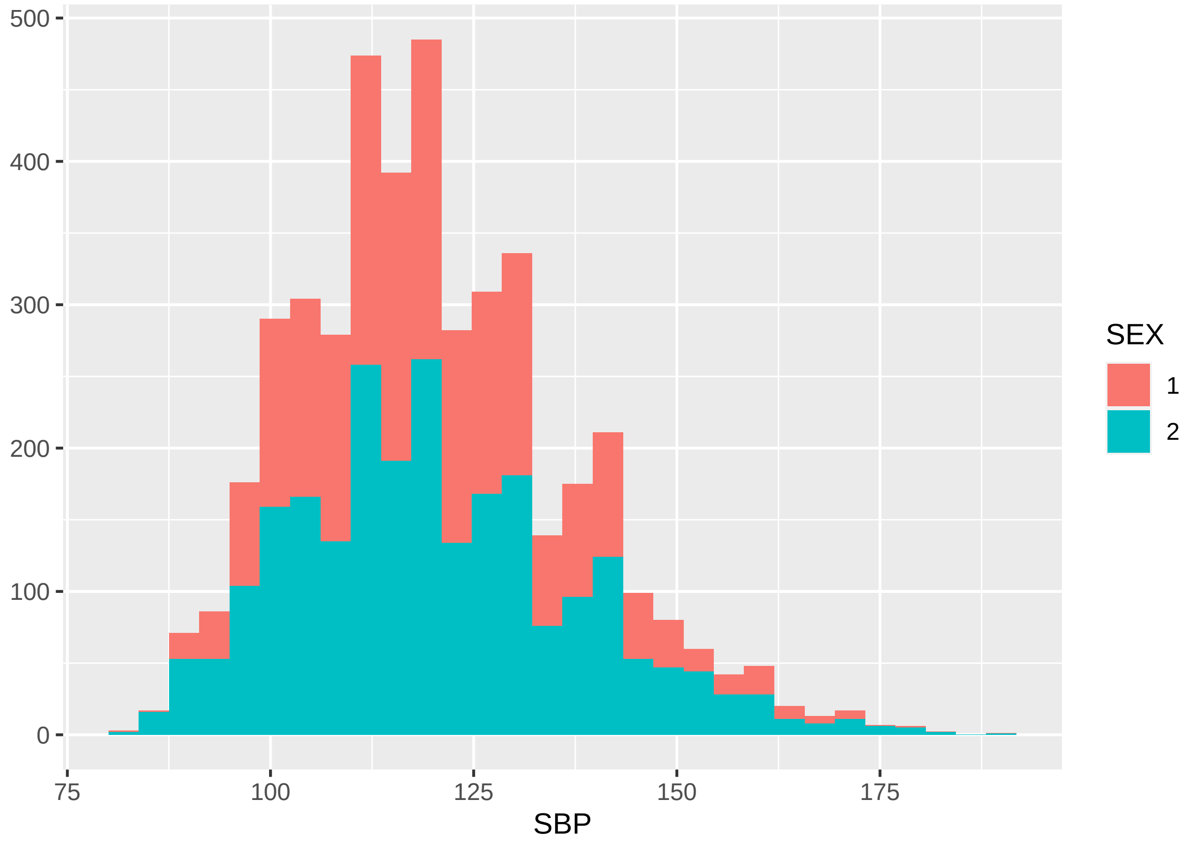
```
# Histogram of SBP by EDU
qplot(x=SBP,
      fill=EDU,
      data=model_df,
      geom = c('histogram'))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



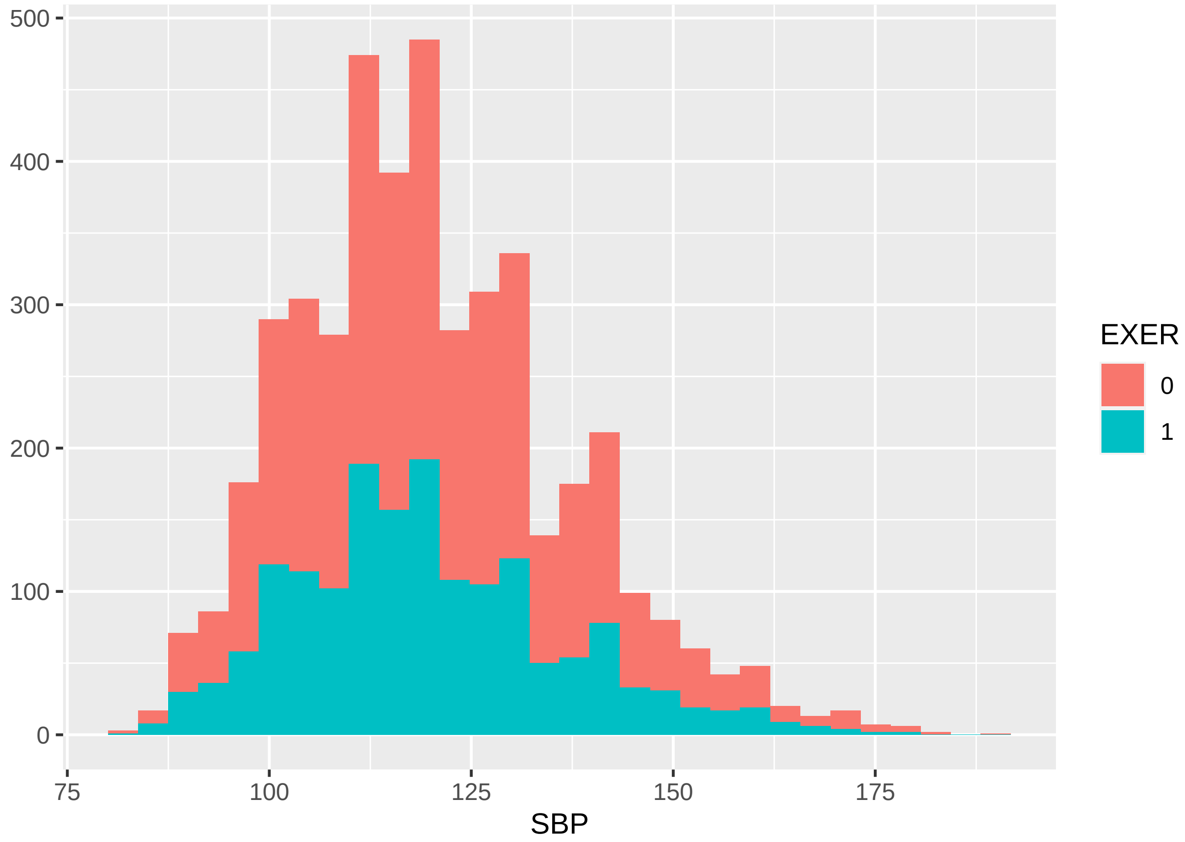
```r
# Histogram of SBP by SEX
qplot(x=SBP,
      fill=SEX,
      data=model_df,
      geom = c('histogram'))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



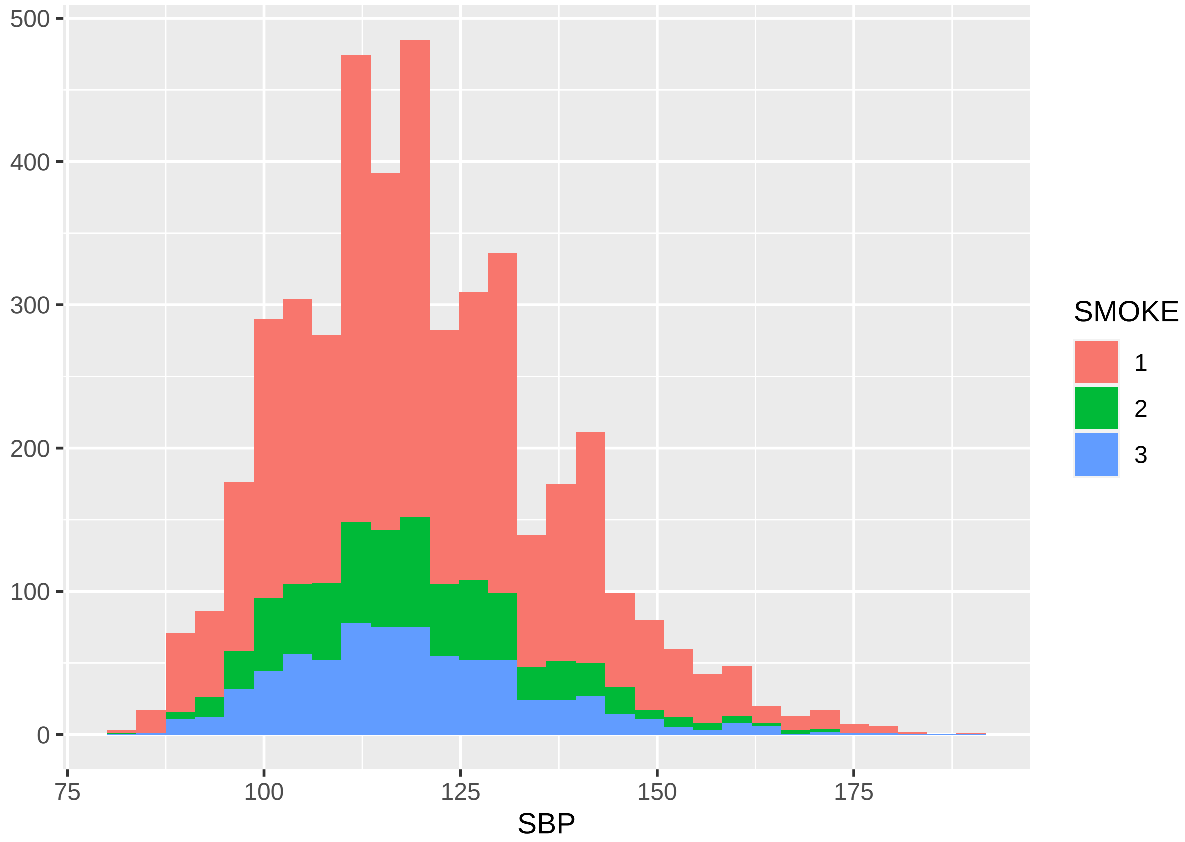
```
# Histogram of SBP by EXER
qplot(x=SBP,
      fill=EXER,
      data=model_df,
      geom = c('histogram'))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



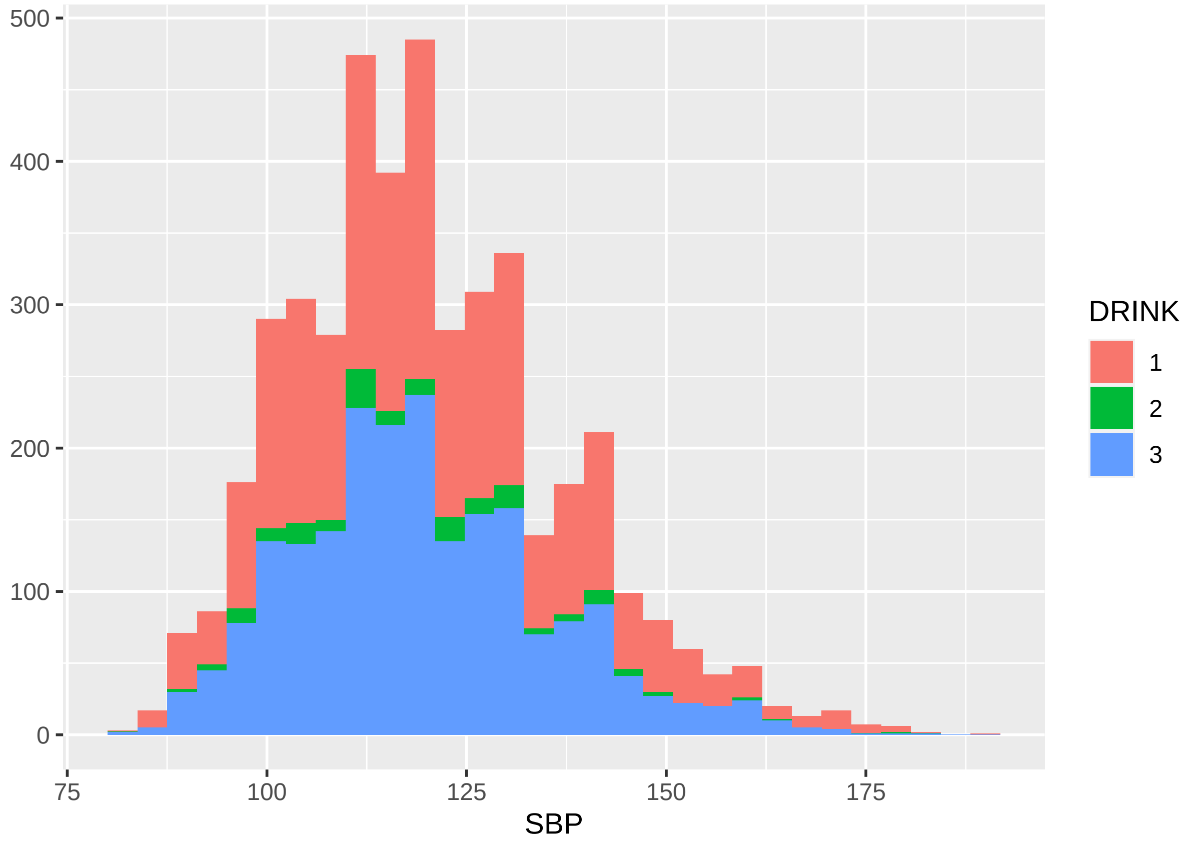
```
# Histogram of SBP by SMOKE
qplot(x=SBP,
      fill=SMOKE,
      data=model_df,
      geom = c('histogram'))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# Histogram of SBP by DRINK
qplot(x=SBP,
      fill=DRINK,
      data=model_df,
      geom = c('histogram'))
```

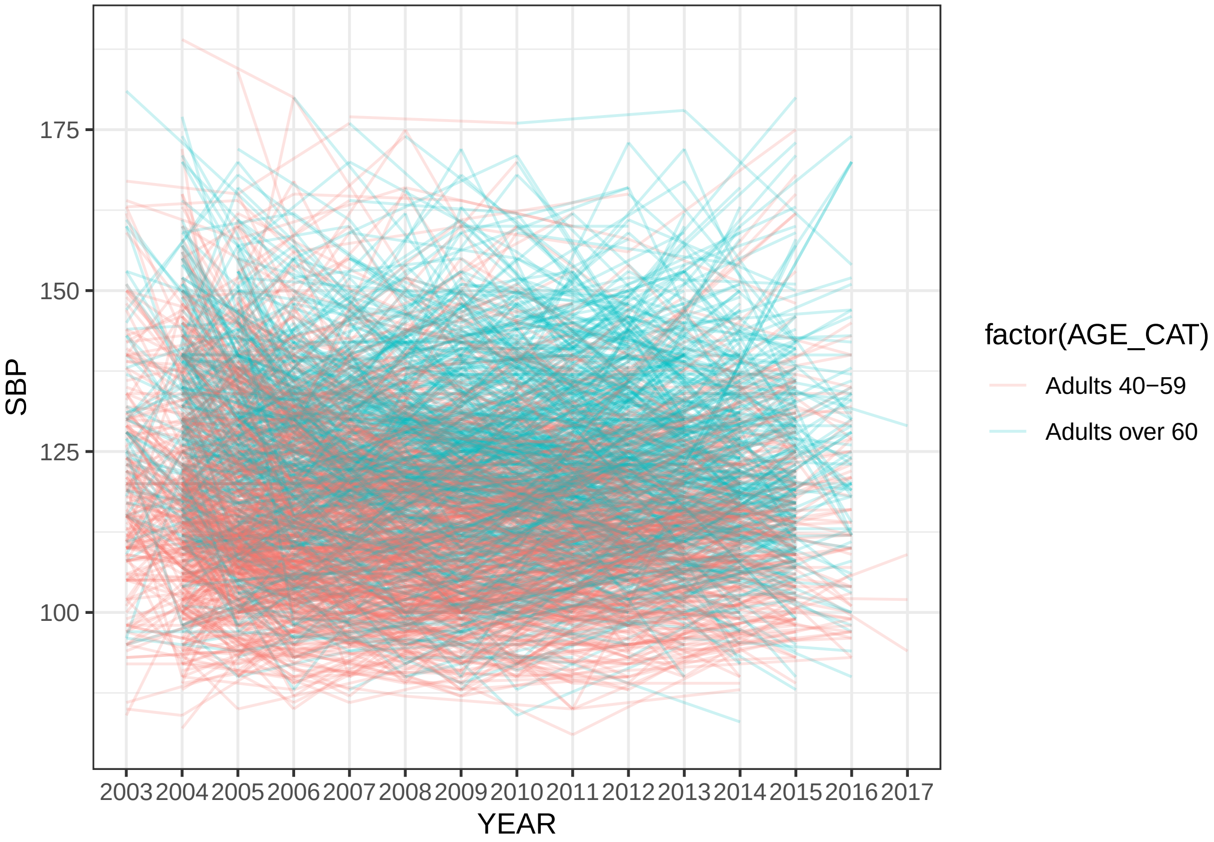
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



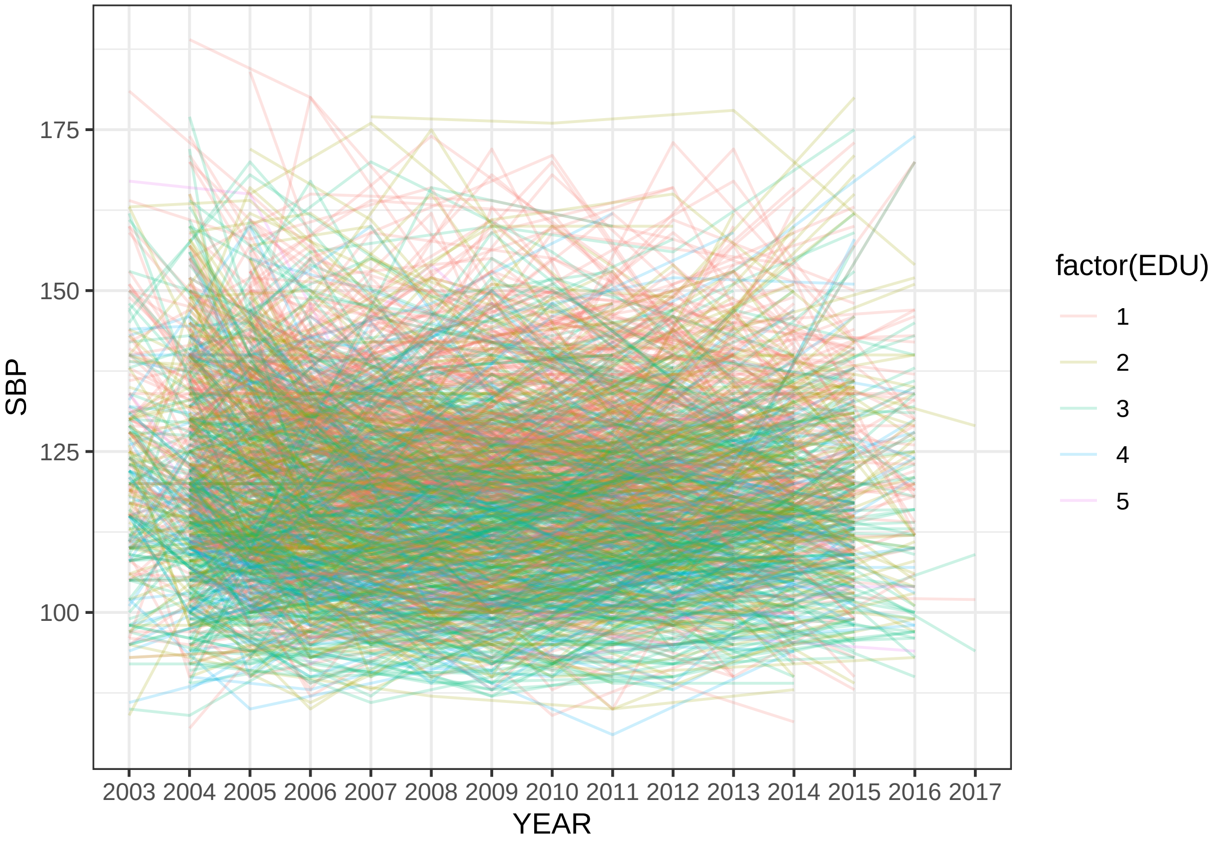
Evolution over time plots

Spaghetti and faceted area plots - not very informative

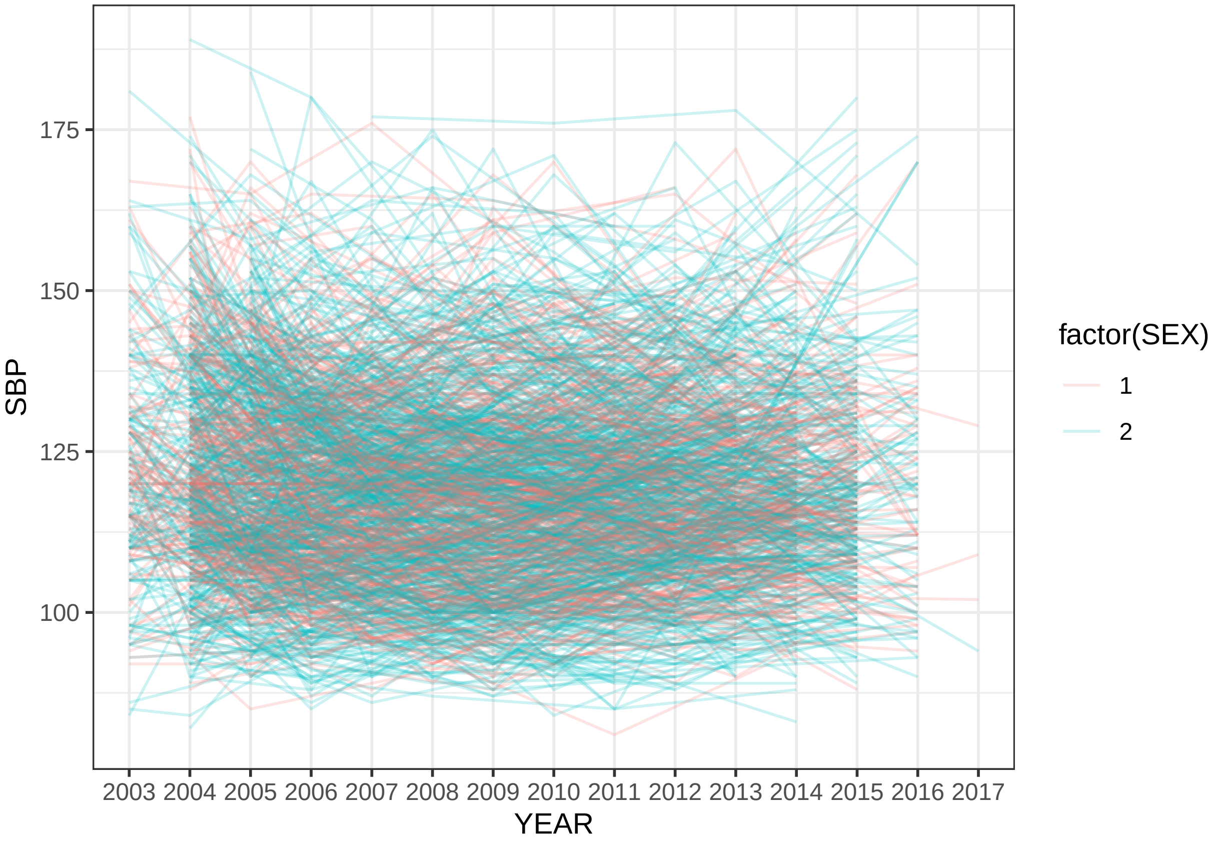
```r
# --- Spaghetti plot of SBP overtime by AGE_CAT ---
sbp_age_s <- ggplot(model_df, aes(YEAR, SBP, colour = factor(AGE_CAT), group = ID)) +
            geom_line(alpha=0.2) +
            theme_bw()
sbp_age_s
```



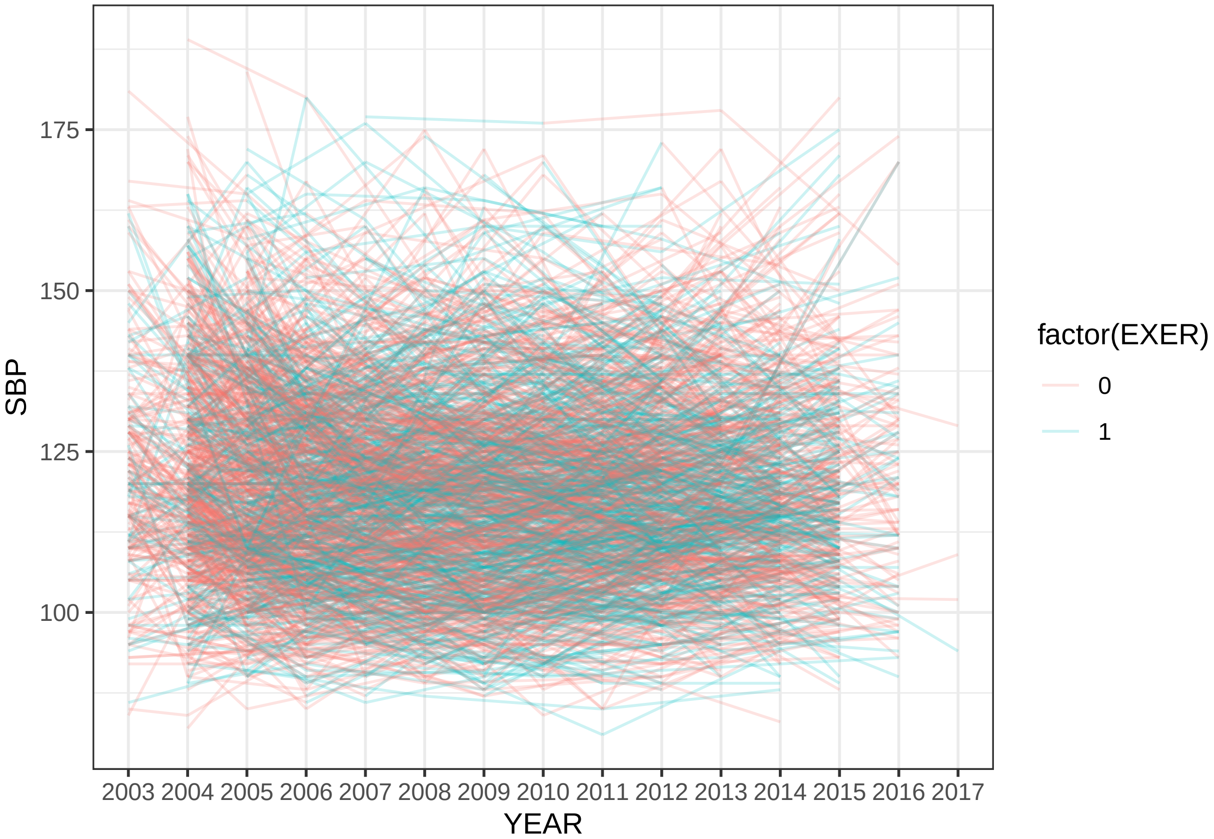
```
# --- Spaghetti plot of SBP overtime by EDU ---
colors = okabe_ito(5)
sbp_edu_s <- ggplot(model_df, aes(YEAR, SBP, colour = factor(EDU), group = ID)) +
            geom_line(alpha=0.2) +
            theme_bw() +
            scale_fill_manual(values=colors)
sbp_edu_s
```



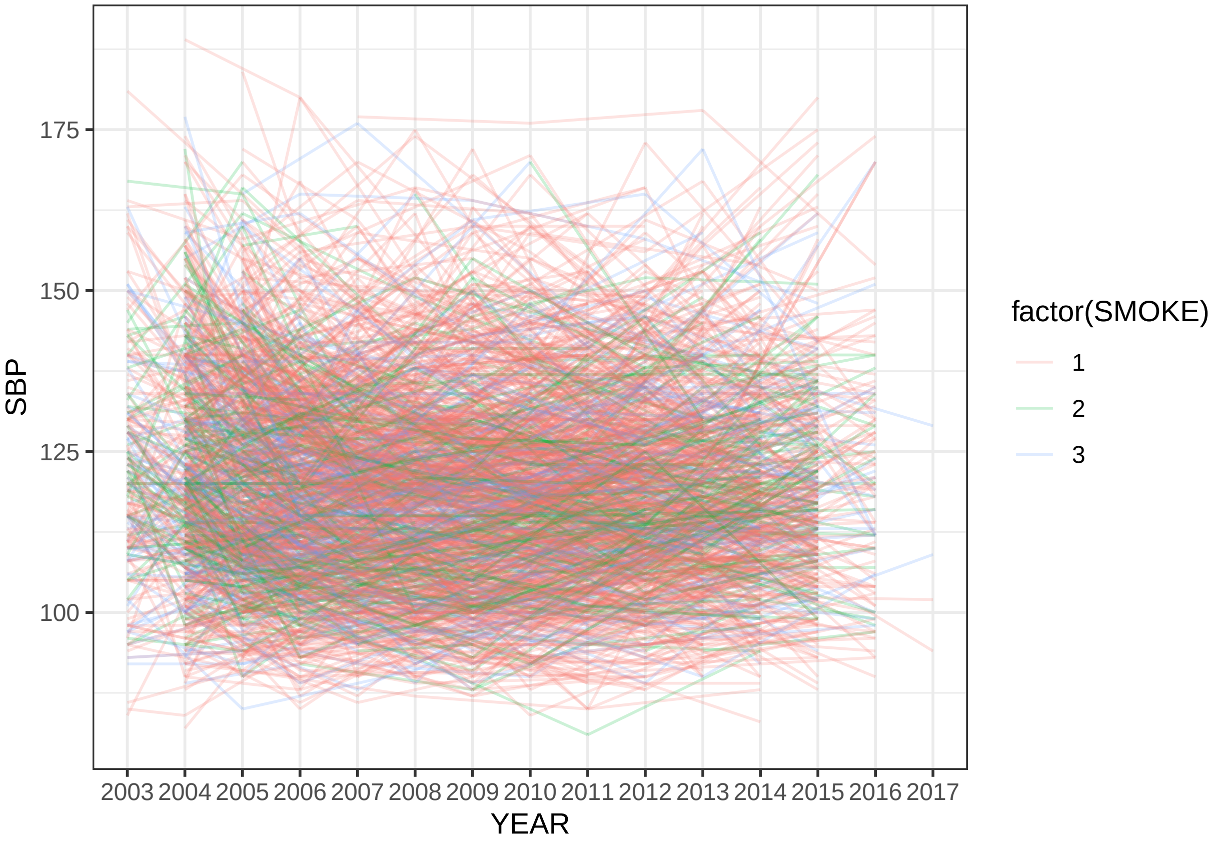
```
# --- Spaghetti plot of SBP overtime by SEX ---
sbp_sex_s <- ggplot(model_df, aes(YEAR, SBP, colour = factor(SEX), group = ID)) +
            geom_line(alpha=0.2) +
            theme_bw() +
            scale_fill_manual(values=colors)
sbp_sex_s
```



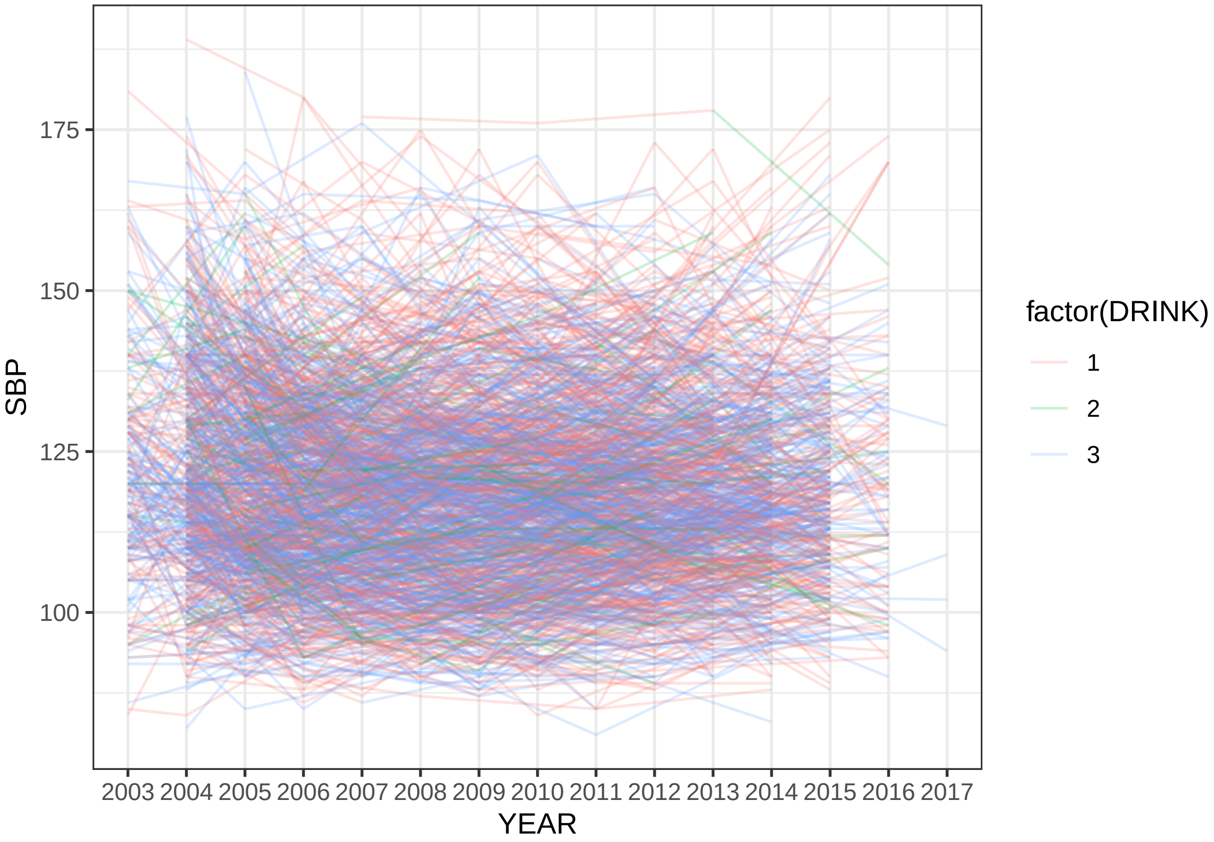
```
# --- Spaghetti plot of SBP overtime by EXER ---
sbp_exer_s <- ggplot(model_df, aes(YEAR, SBP, colour = factor(EXER), group = ID)) +
            geom_line(alpha=0.2) +
            theme_bw() +
            scale_fill_manual(values=colors)
sbp_exer_s
```



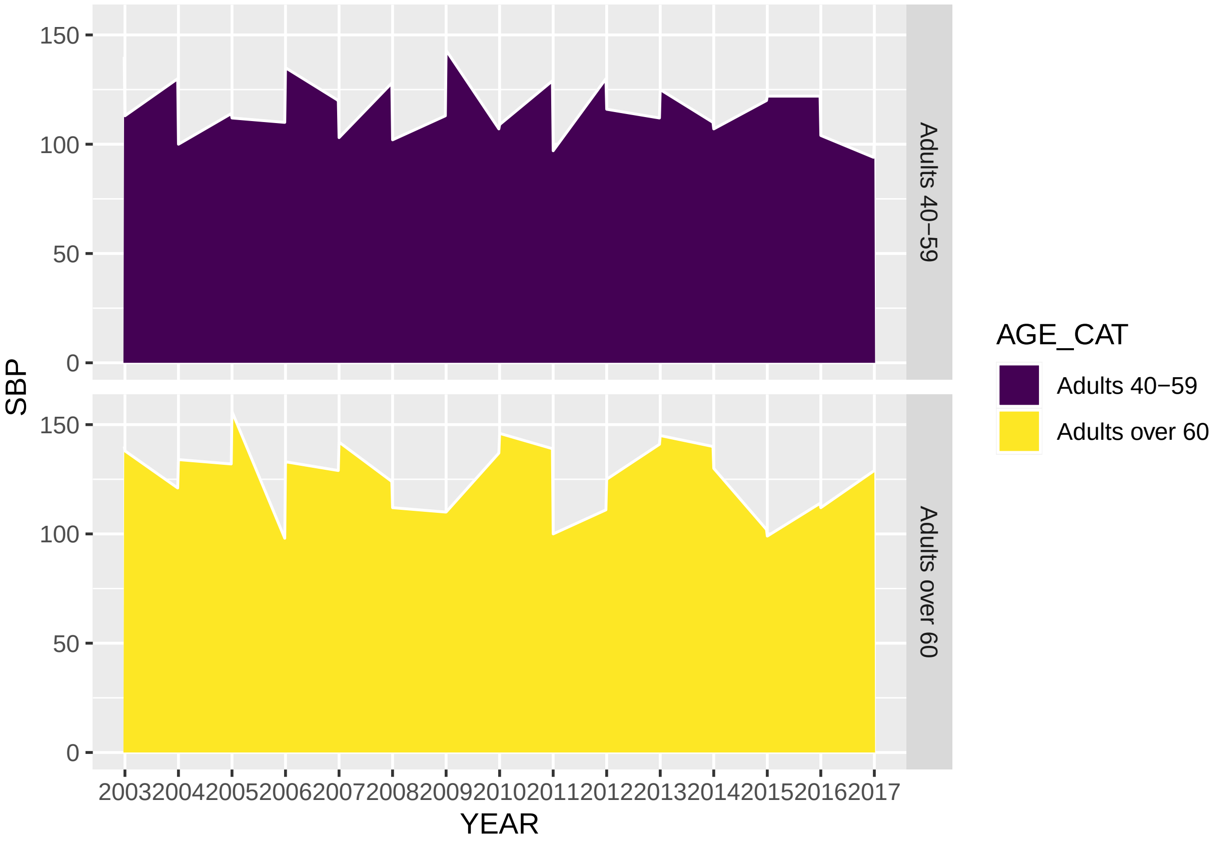
```
# --- Spaghetti plot of SBP overtime by SMOKE ---
sbp_smoke_s <- ggplot(model_df, aes(YEAR, SBP, colour = factor(SMOKE), group = ID)) +
            geom_line(alpha=0.2) +
            theme_bw() +
            scale_fill_manual(values=colors)
sbp_smoke_s
```



```
# --- Spaghetti plot of SBP overtime by DRINK ---
sbp_drink_s <- ggplot(model_df, aes(YEAR, SBP, colour = factor(DRINK), group = ID)) +
            geom_line(alpha=0.2) +
            theme_bw() +
            scale_fill_manual(values=colors)
sbp_drink_s
```



```
# --- Faceted Area plot of SBP overtime by AGE_CAT ---
sbp_age_facet <- model_df %>%
  ggplot(aes(YEAR, SBP, group = AGE_CAT, fill = AGE_CAT)) +
  geom_area(color='white') +
  scale_fill_viridis(discrete = TRUE) +
  facet_grid(AGE_CAT ~.)
sbp_age_facet
```

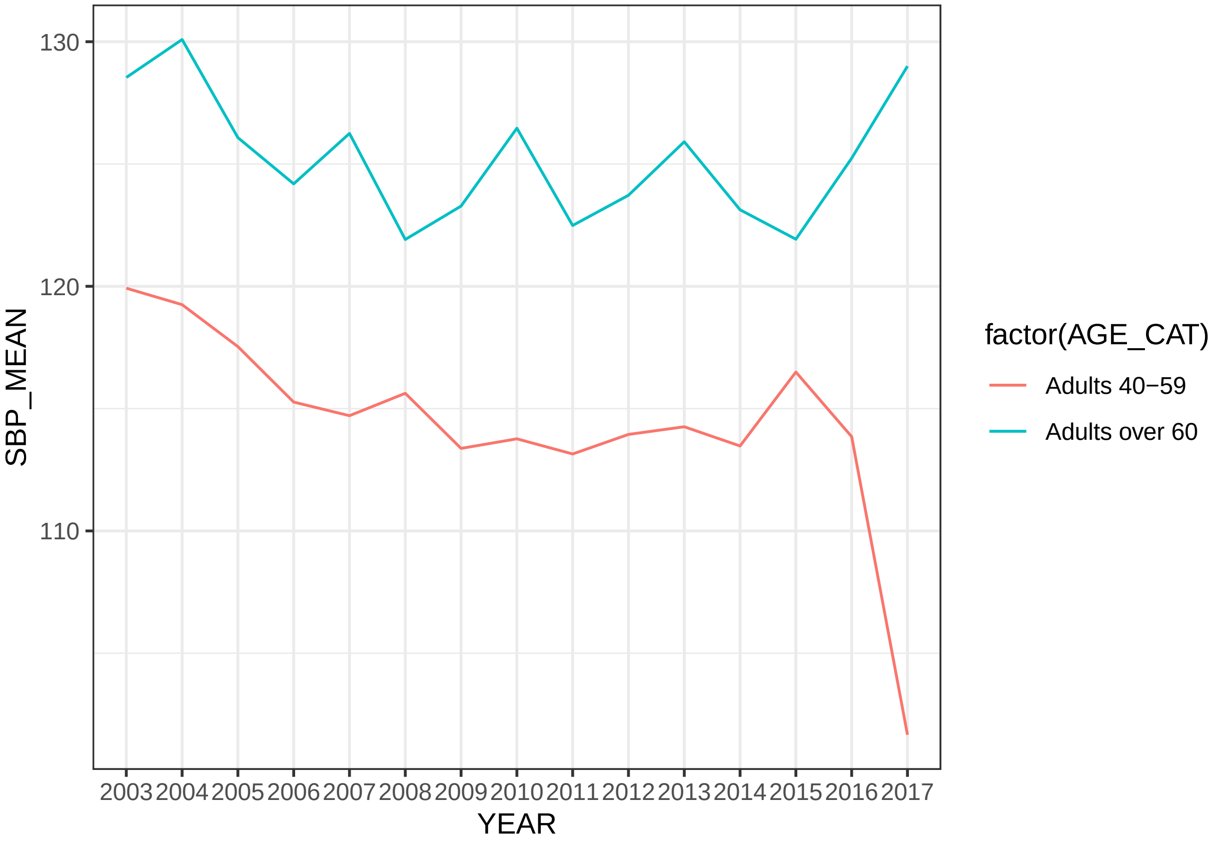


Evolution of SBP mean and medians by predictor category over time

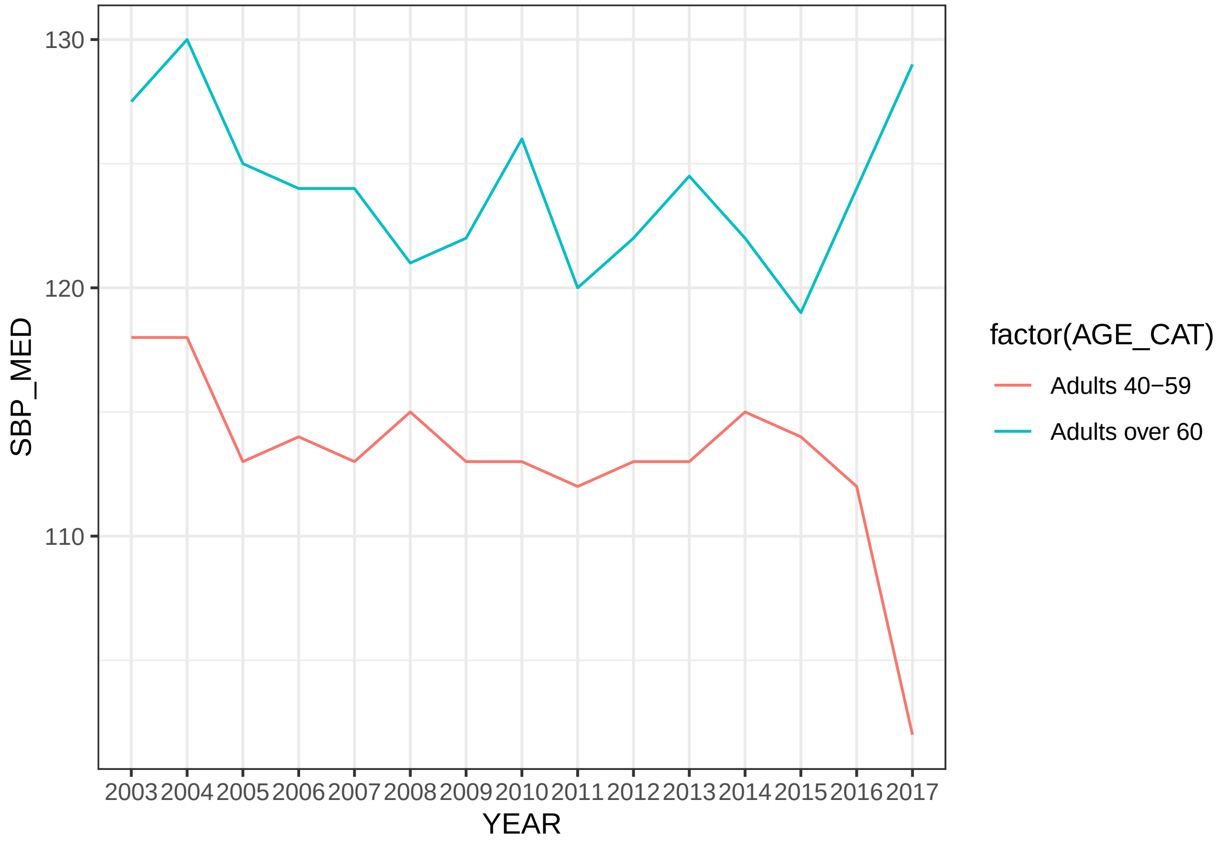
```r
# --- Creating new df with SBP mean and median by AGE category ---
sbp_age_stats <- model_df %>%
  group_by(YEAR, AGE_CAT) %>%
  summarize(SBP_MEAN = mean(SBP),
            SBP_MED = median(SBP)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```r
# Line graph of SBP_MEAN over time based on age group
sbp_age_avg_line <- ggplot(sbp_age_stats, aes(YEAR, y = SBP_MEAN, colour = factor(AGE_CAT), group = AGE_
            geom_line() +
            theme_bw()
sbp_age_avg_line
```



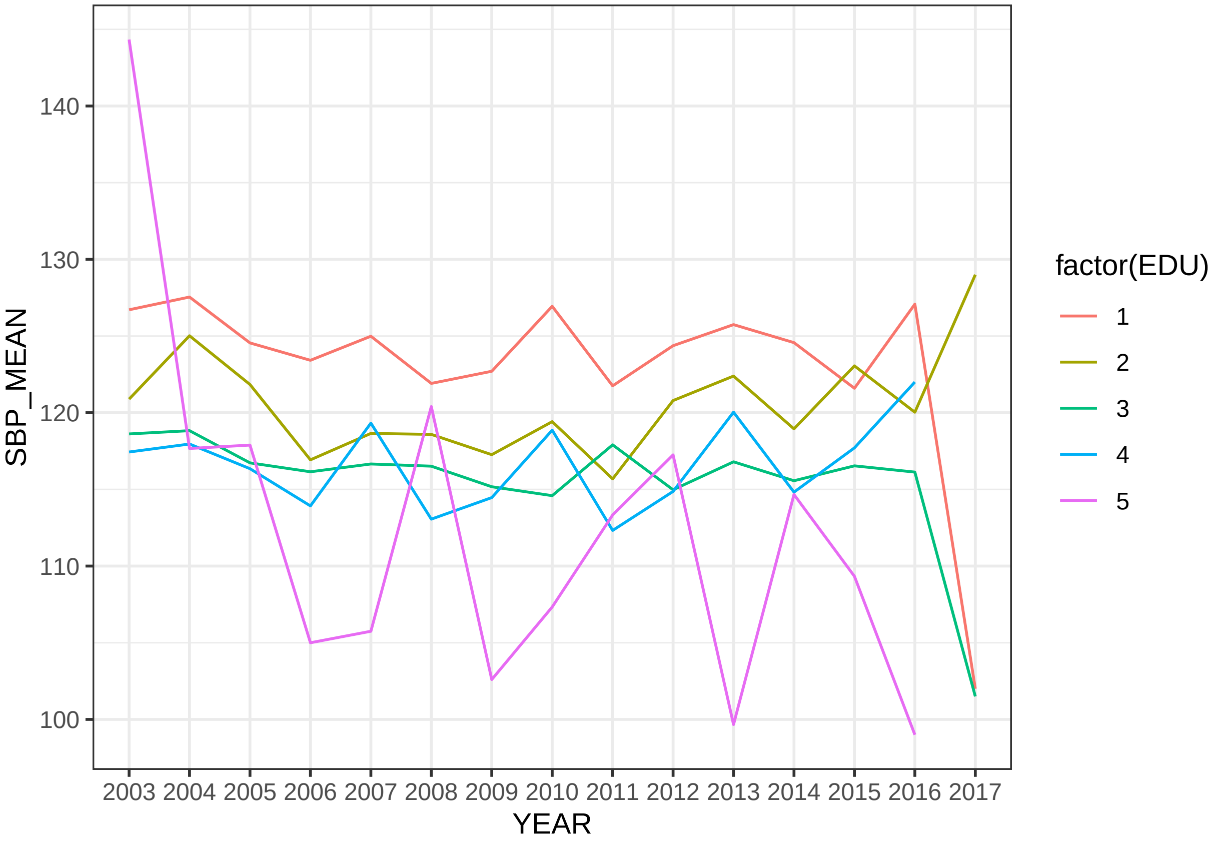
```
# Line graph of SBP_MED over time based on age group
sbp_age_med_line <- ggplot(sbp_age_stats, aes(YEAR, y = SBP_MED, colour = factor(AGE_CAT), group = AGE_C
            geom_line() +
            theme_bw()
sbp_age_med_line
```



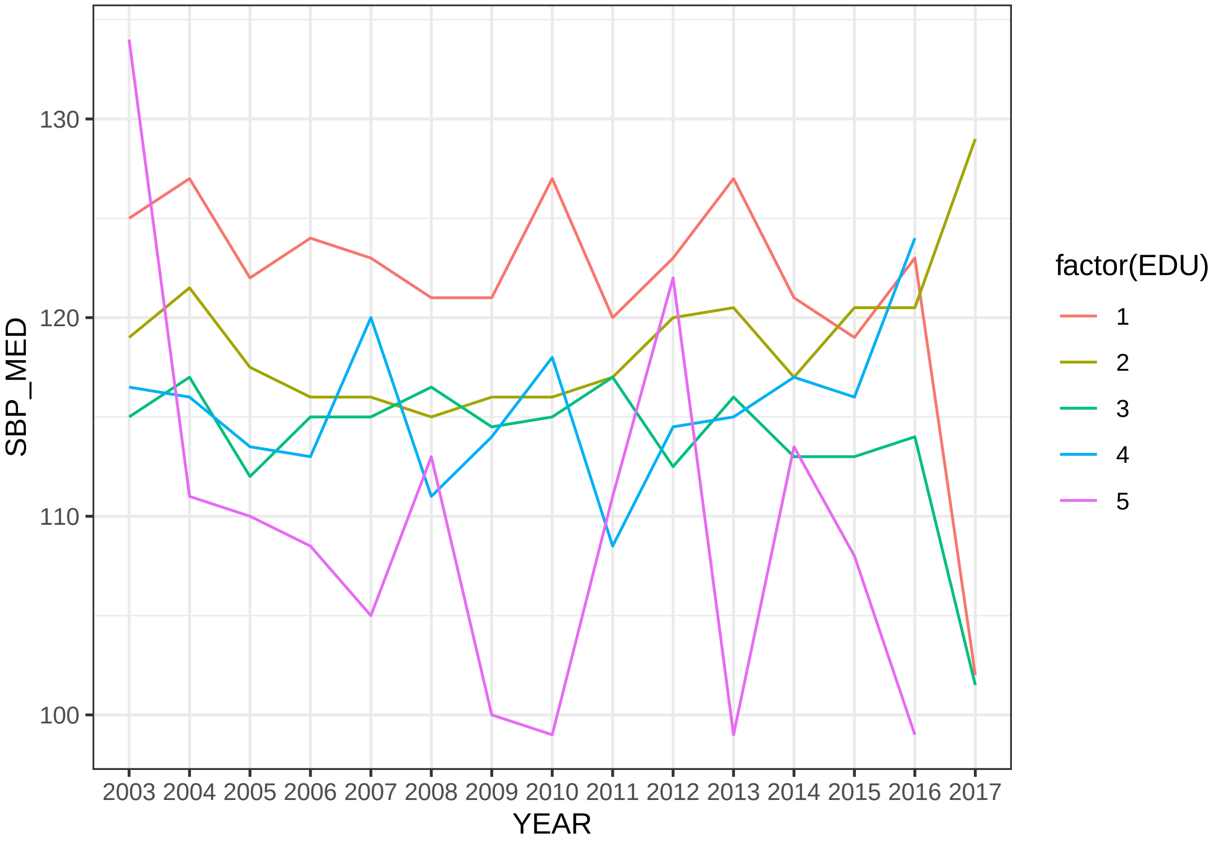
```
# 2002 Two off-duty U.S. servicemen accidentally kill two South Korean middle-school girls while drivin
# 2003 denuclearization conversations b/t U.S. and Korea, North Korea withdraws from these
# 2008 financial crisis in Korea
# 2011 Seoul floods

# --- Creating new df with SBP mean and median by EDU category ---
sbp_edu_stats <- model_df %>%
  group_by(YEAR, EDU) %>%
  summarize(SBP_MEAN = mean(SBP),
            SBP_MED = median(SBP)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```
# Line graph of SBP_MEAN over time based on EDU group
sbp_edu_avg_line <- ggplot(sbp_edu_stats, aes(YEAR, y = SBP_MEAN, colour = factor(EDU), group = EDU)) +
            geom_line() +
            theme_bw()
sbp_edu_avg_line
```



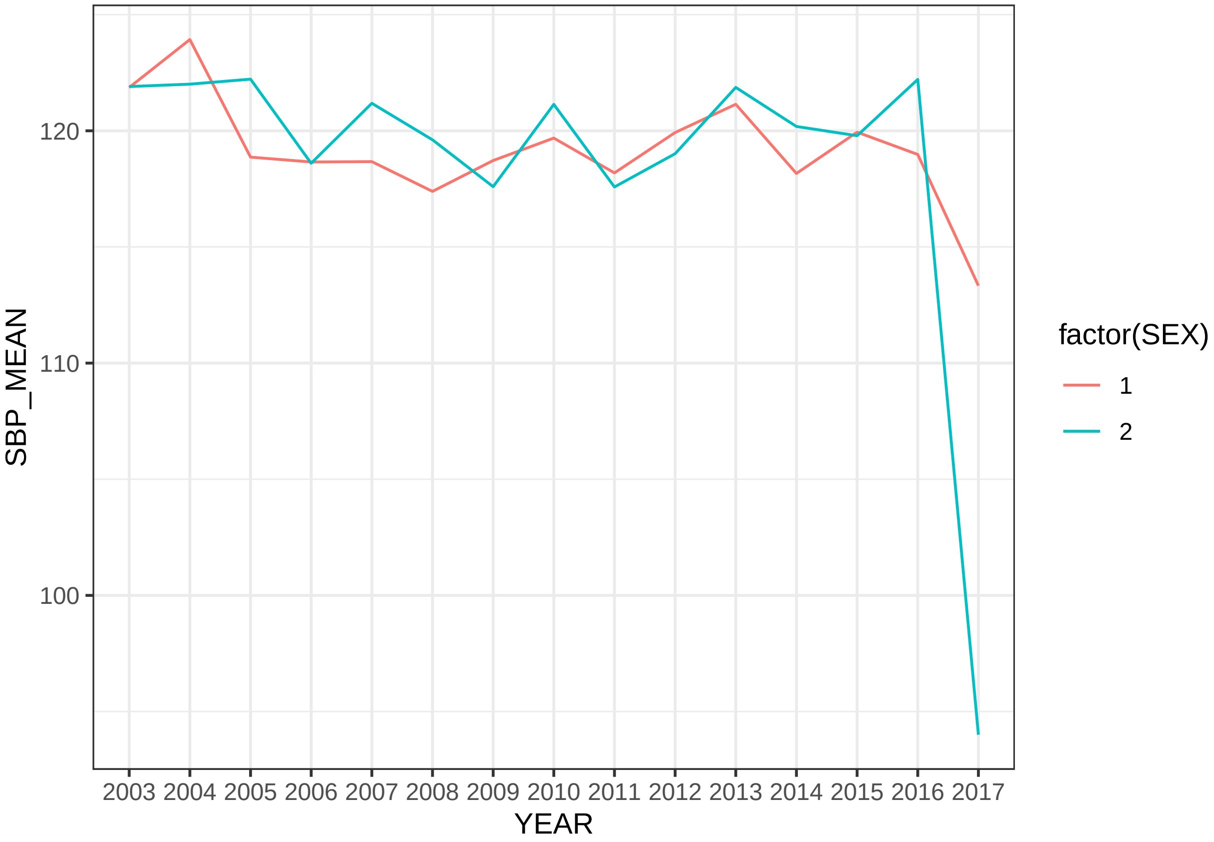
```
# Line graph of SBP_MED over time based on EDU group
sbp_edu_med_line <- ggplot(sbp_edu_stats, aes(YEAR, y = SBP_MED, colour = factor(EDU), group = EDU)) +
            geom_line() +
            theme_bw()
sbp_edu_med_line
```



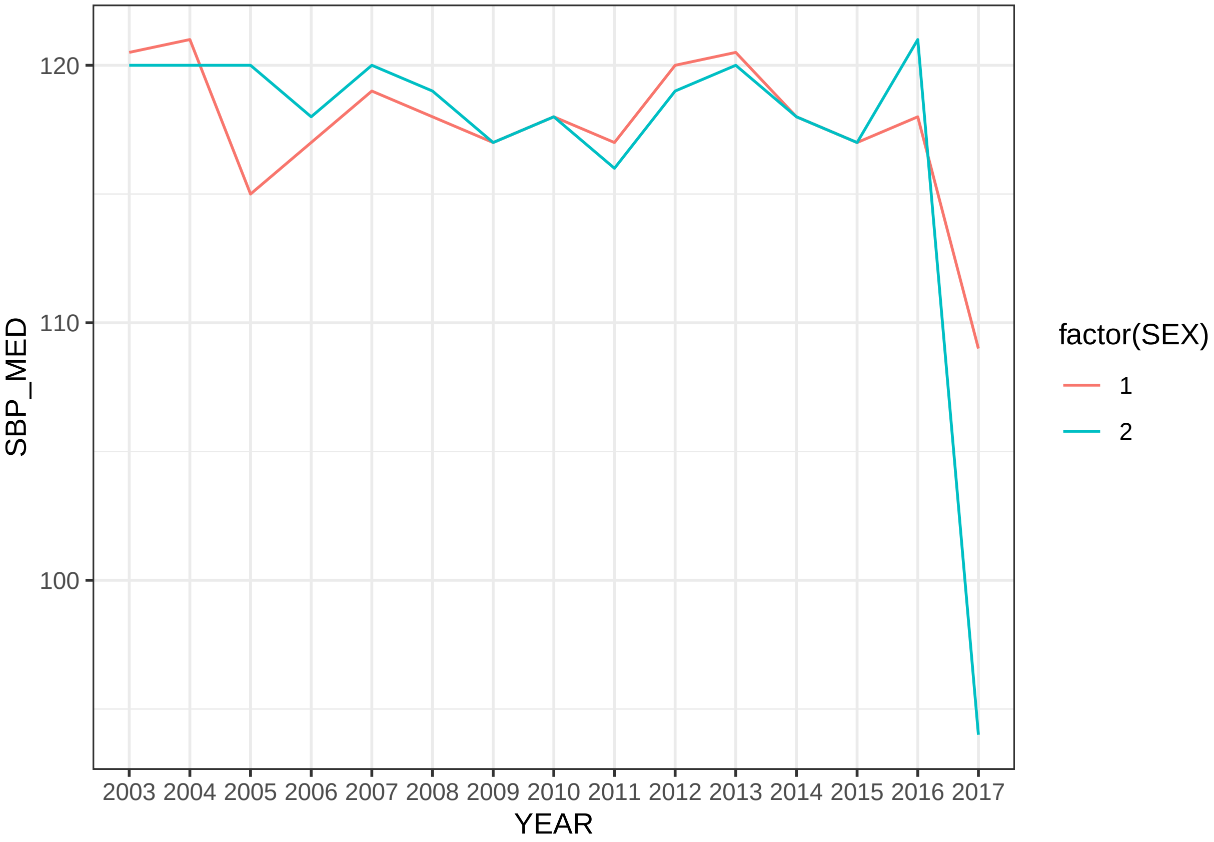
```
# --- Creating new df with SBP mean and median by SEX category ---
sbp_sex_stats <- model_df %>%
  group_by(YEAR, SEX) %>%
  summarize(SBP_MEAN = mean(SBP),
            SBP_MED = median(SBP)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```
# Line graph of SBP_MEAN over time based on SEX group
sbp_sex_avg_line <- ggplot(sbp_sex_stats, aes(YEAR, y = SBP_MEAN, colour = factor(SEX), group = SEX)) +
            geom_line() +
            theme_bw()
sbp_sex_avg_line
```



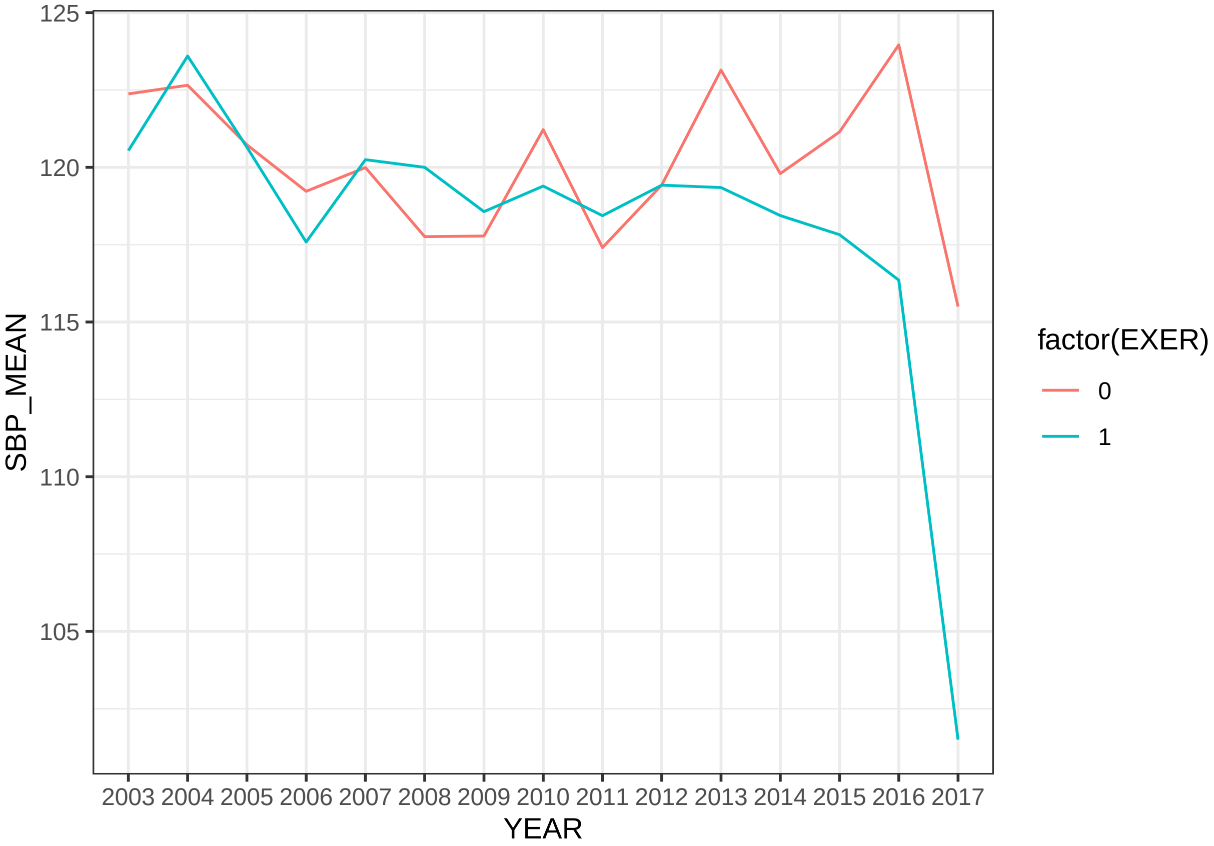
```
# Line graph of SBP_MED over time based on SEX group
sbp_sex_med_line <- ggplot(sbp_sex_stats, aes(YEAR, y = SBP_MED, colour = factor(SEX), group = SEX)) +
            geom_line() +
            theme_bw()
sbp_sex_med_line
```



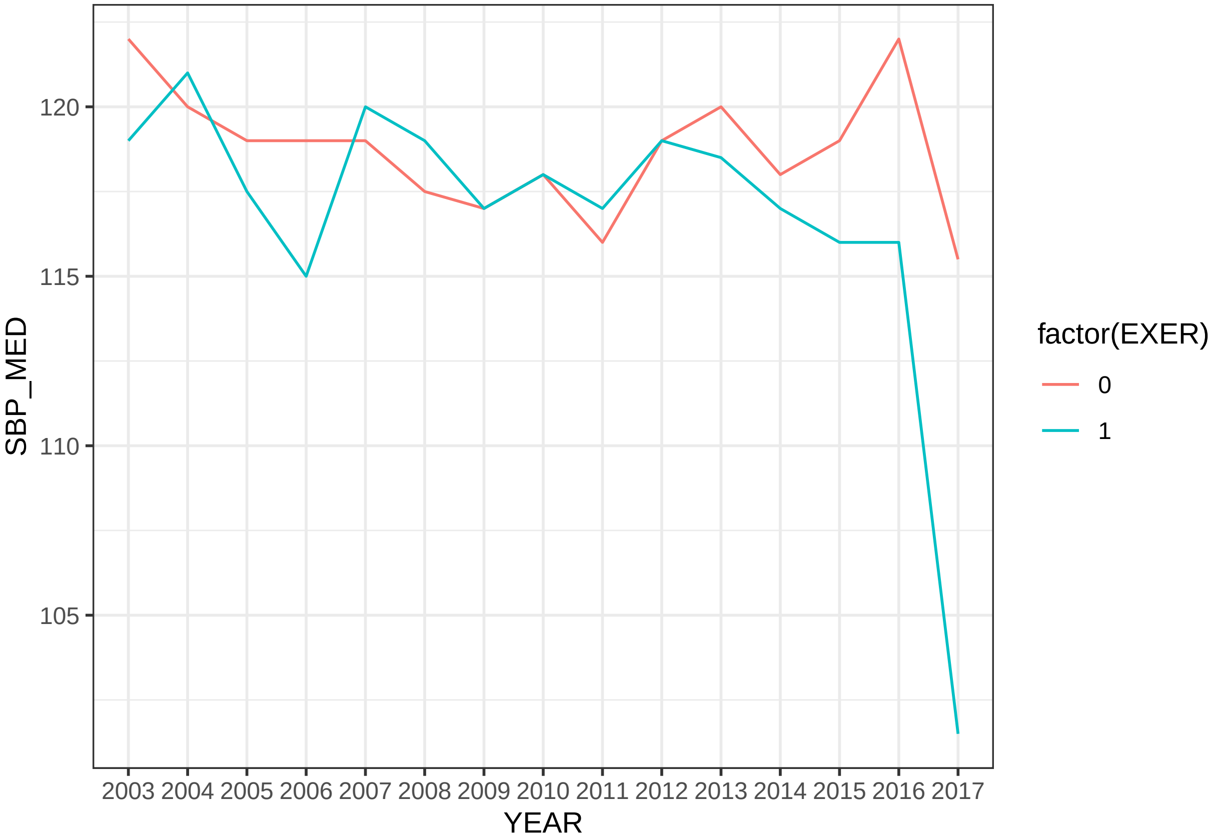
```
# --- Creating new df with SBP mean and median by EXER category ---
sbp_exer_stats <- model_df %>%
  group_by(YEAR, EXER) %>%
  summarize(SBP_MEAN = mean(SBP),
            SBP_MED = median(SBP)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```
# Line graph of SBP_MEAN over time based on EXER group
sbp_exer_avg_line <- ggplot(sbp_exer_stats, aes(YEAR, y = SBP_MEAN, colour = factor(EXER), group = EXER
            geom_line() +
            theme_bw()
sbp_exer_avg_line
```



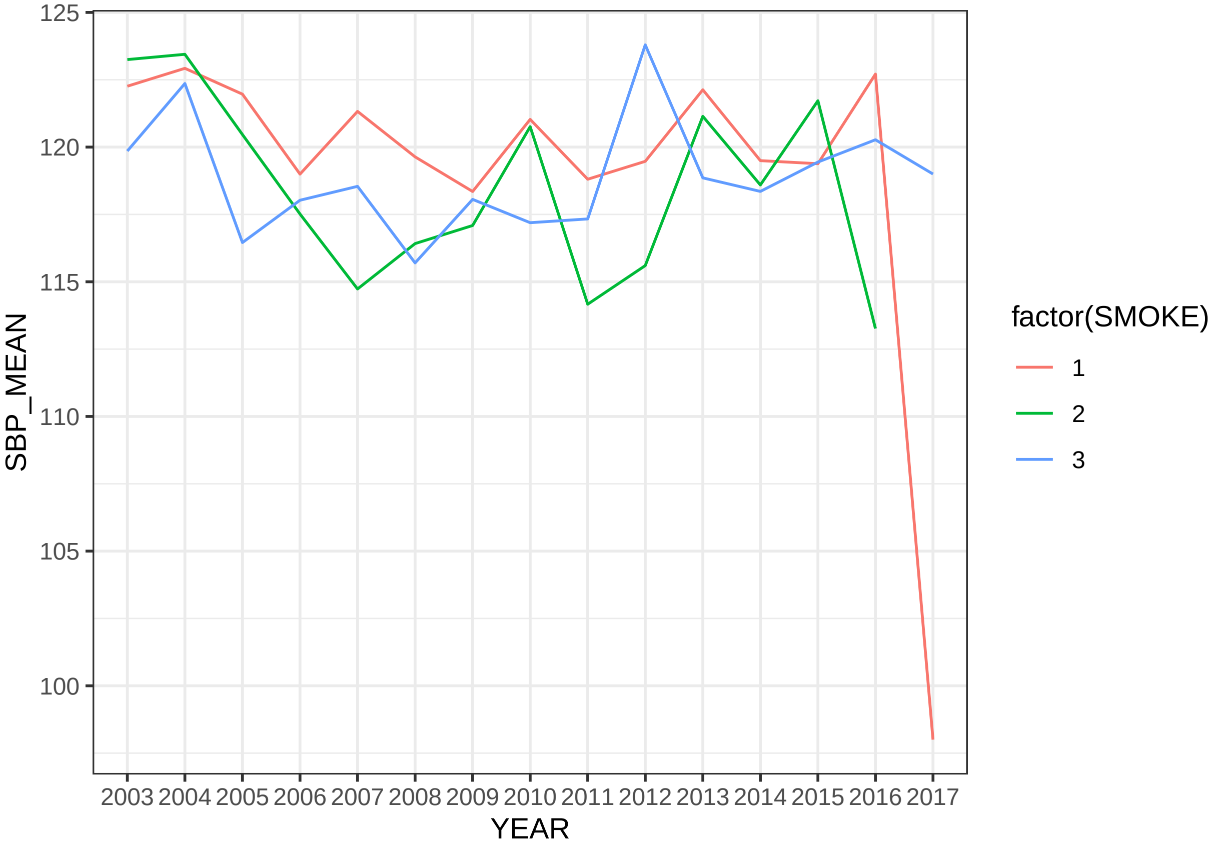
```
# Line graph of SBP_MED over time based on EXER group
sbp_exer_med_line <- ggplot(sbp_exer_stats, aes(YEAR, y = SBP_MED, colour = factor(EXER), group = EXER)
            geom_line() +
            theme_bw()
sbp_exer_med_line
```



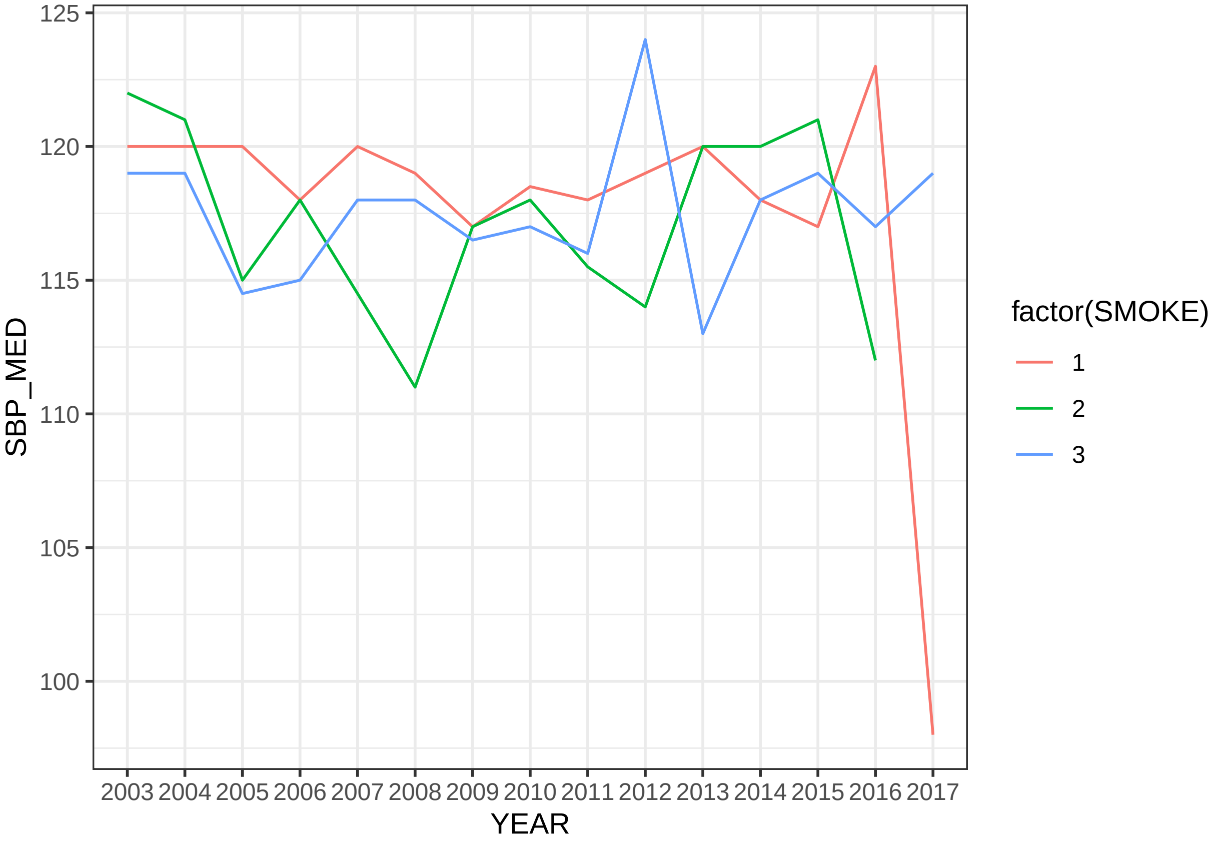
```
# --- Creating new df with SBP mean and median by SMOKE category ---
sbp_smoke_stats <- model_df %>%
  group_by(YEAR, SMOKE) %>%
  summarize(SBP_MEAN = mean(SBP),
            SBP_MED = median(SBP)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```
# Line graph of SBP_MEAN over time based on SMOKE group
sbp_smoke_avg_line <- ggplot(sbp_smoke_stats, aes(YEAR, y = SBP_MEAN, colour = factor(SMOKE), group = S
            geom_line() +
            theme_bw()
sbp_smoke_avg_line
```



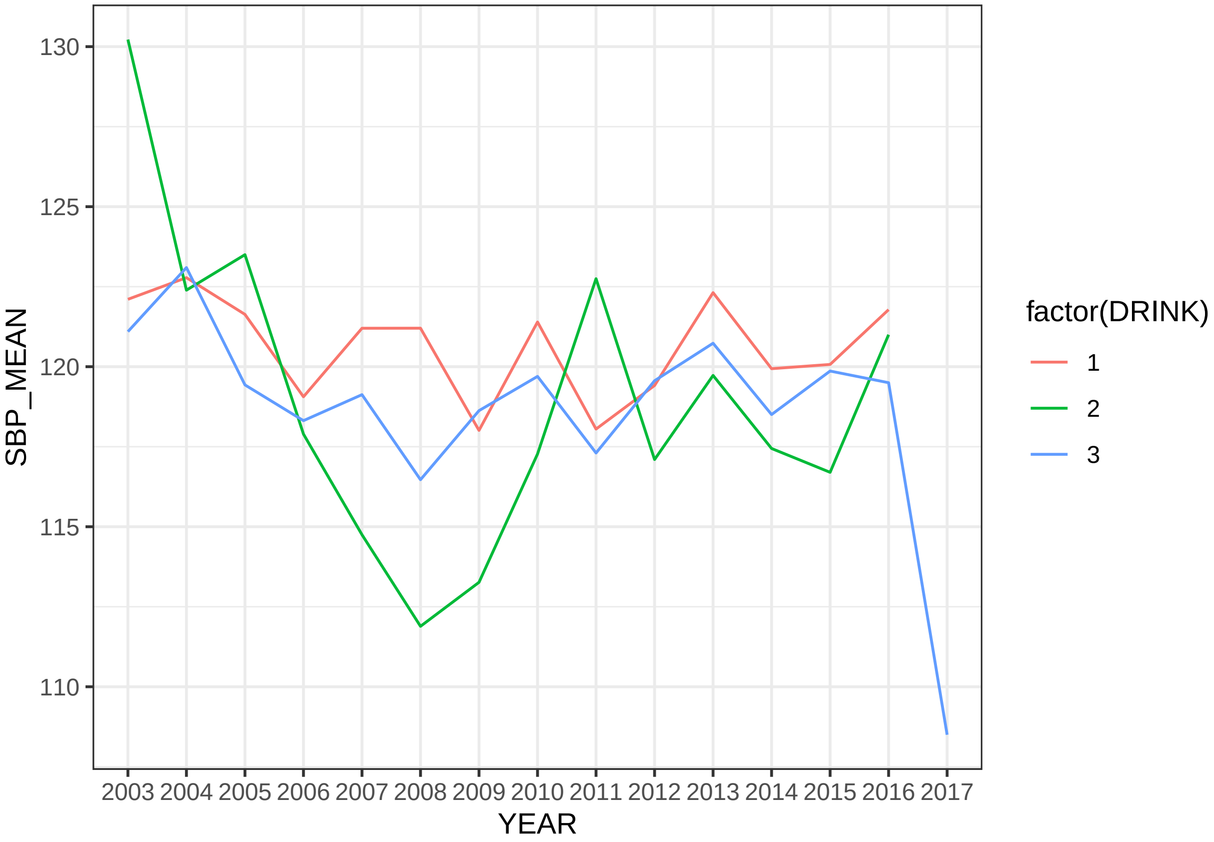
```r
# Line graph of SBP_MED over time based on SMOKE group
sbp_smoke_med_line <- ggplot(sbp_smoke_stats, aes(YEAR, y = SBP_MED, colour = factor(SMOKE), group = SMO
            geom_line() +
            theme_bw()
sbp_smoke_med_line
```



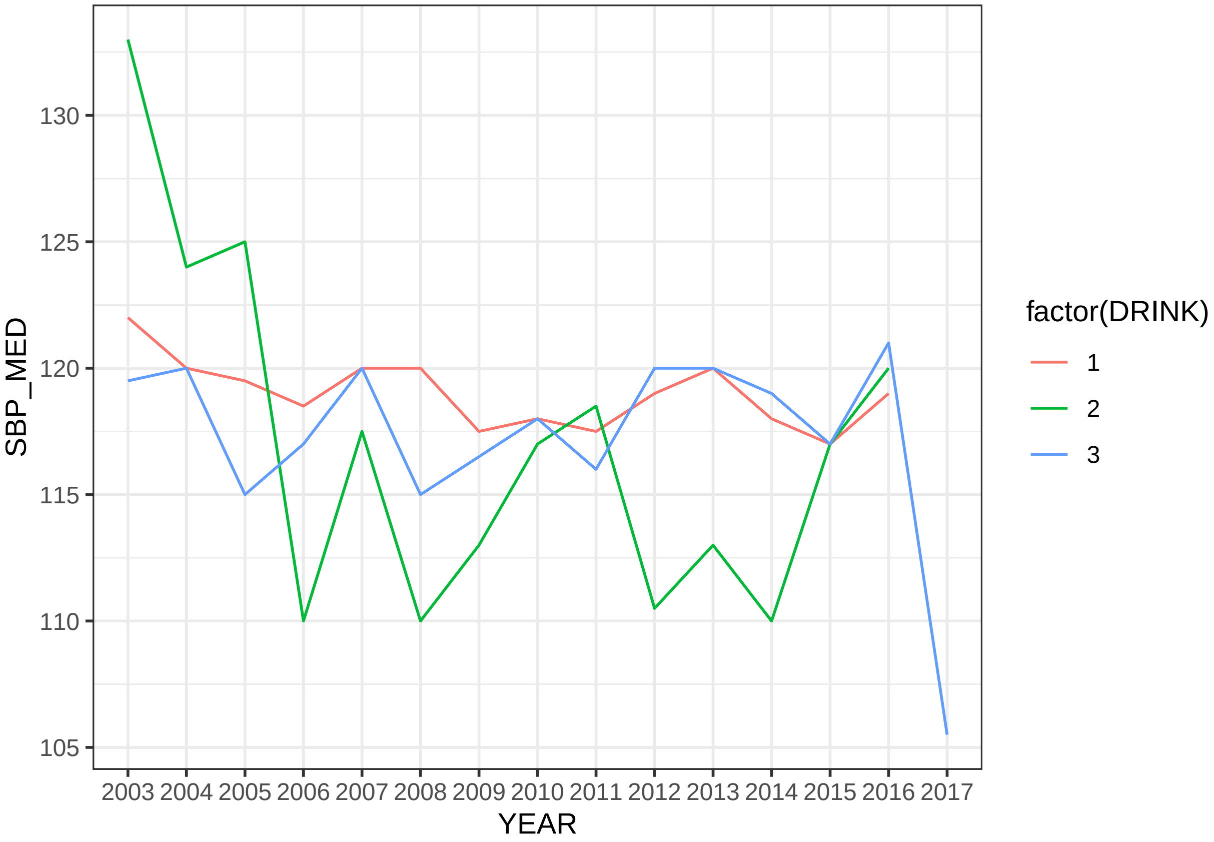
```
# --- Creating new df with SBP mean and median by DRINK category ---
sbp_drink_stats <- model_df %>%
  group_by(YEAR, DRINK) %>%
  summarize(SBP_MEAN = mean(SBP),
            SBP_MED = median(SBP)) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```
# Line graph of SBP_MEAN over time based on DRINK group
sbp_drink_avg_line <- ggplot(sbp_drink_stats, aes(YEAR, y = SBP_MEAN, colour = factor(DRINK), group = DI
            geom_line() +
            theme_bw()
sbp_drink_avg_line
```



```
# Line graph of SBP_MED over time based on DRINK group
sbp_drink_med_line <- ggplot(sbp_drink_stats, aes(YEAR, y = SBP_MED, colour = factor(DRINK), group = DRI
            geom_line() +
            theme_bw()
sbp_drink_med_line
```



GEE models & model comparisons using QIC

```
# How DRINK, SMOKE, EXER effect Systolic
# Model assuming exchangeable correlation structure
m1_e <- geeglm(SBP ~ DRINK + SMOKE + EXER + AGE + SEX + EDU,
                 id = ID, corstr = 'exchangeable', data = model_df)

# Model assuming ar1 correlation structure
m1_a <- geeglm(SBP ~ DRINK + SMOKE + EXER + AGE + SEX + EDU,
                 id = ID, corstr = 'ar1', data = model_df)

# Model assuming independence correlation structure
m1_i <- geeglm(SBP ~ DRINK + SMOKE + EXER + AGE + SEX + EDU,
                 id = ID, corstr = 'independence', data = model_df)

# Model assuming unstructured correlation structure
m1_u <- geeglm(SBP ~ DRINK + SMOKE + EXER + AGE + SEX + EDU,
                 id = ID, corstr = 'unstructured', data = model_df)
```

Model comparisons using QIC

```
# Exchangeable
QIC(m1_e)
```

```
##          QIC         QICu     Quasi Lik          CIC       params         QICC
## 1145587.1533 1145564.0519 -572770.0259      23.5507      12.0000 1145587.5251
```

```r
# ar1
QIC(m1_a)
```

```
##           QIC          QICu     Quasi Lik           CIC        params
## 1135773.41024 1135750.19849 -567863.09924      23.60588      12.00000
##          QICC
## 1135773.78205
```

```r
# independence
QIC(m1_i)
```

```
##           QIC          QICu     Quasi Lik           CIC        params
## 1121801.25275 1121773.88683 -560874.94342      25.68296      12.00000
##          QICC
## 1121801.57112
```

```r
# unstructured
QIC(m1_u)
```

```
##           QIC          QICu     Quasi Lik           CIC        params
## 1141489.03412 1141467.21676 -570721.60838      22.90868      12.00000
##          QICC
## 1141490.07742
```

Independent correlation structure has the lowest QIC, and thus is the best working correlation structure.

- Exchangeable QIC: 1,145,587
- ar1 QIC: 1,135,773
- Independent QIC: 1,121,801
- Unstructured QIC: 1,141,489

GEE model output with independent correlation structure

```r
summary(m1_i)
```

```
##
## Call:
## geeglm(formula = SBP ~ DRINK + SMOKE + EXER + AGE + SEX + EDU,
##     data = model_df, id = ID, corstr = "independence")
##
##  Coefficients:
##             Estimate  Std.err    Wald Pr(>|W|)
## (Intercept) 99.48069  3.22018 954.370  < 2e-16 ***
## DRINK2      -0.96103  1.44495   0.442 0.505990
## DRINK3       0.91377  0.80781   1.280 0.257984
## SMOKE2      -0.44765  1.12266   0.159 0.690085
## SMOKE3      -0.70812  1.18872   0.355 0.551376
## EXER1        0.66078  0.64392   1.053 0.304804
## AGE          0.40403  0.04242  90.714  < 2e-16 ***
## SEX2        -1.97140  1.09844   3.221 0.072699 .
## EDU2        -1.86133  1.18303   2.475 0.115636
## EDU3        -4.52897  1.18656  14.569 0.000135 ***
## EDU4        -5.48791  1.51836  13.064 0.000301 ***
## EDU5        -9.69253  2.94618  10.823 0.001002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

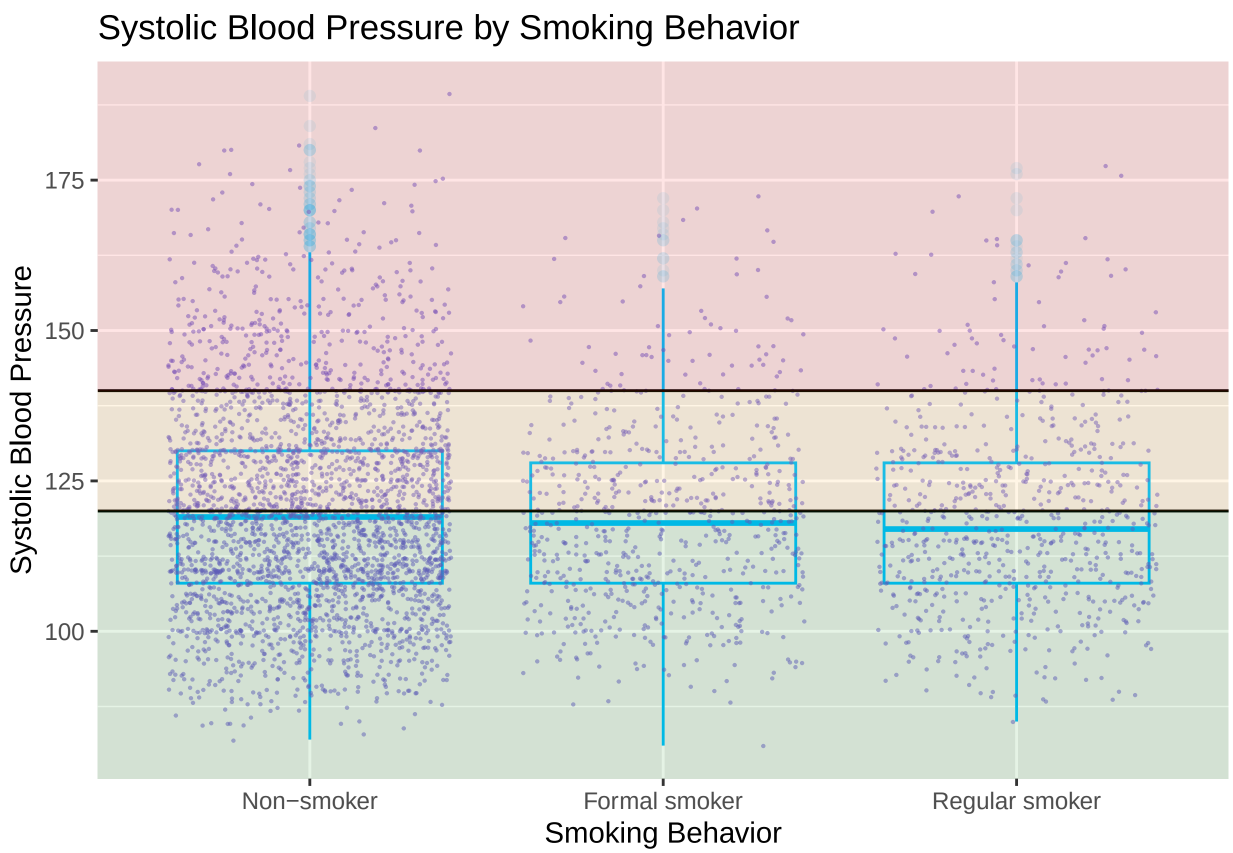
```
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)    253.6    8.05
## Number of clusters:   993  Maximum cluster size: 5
```

—Experimenting with other visualizations—

Box and Jitter Plots of SBP over SMOKE, DRINK, EXER – Violin Plots are more informative

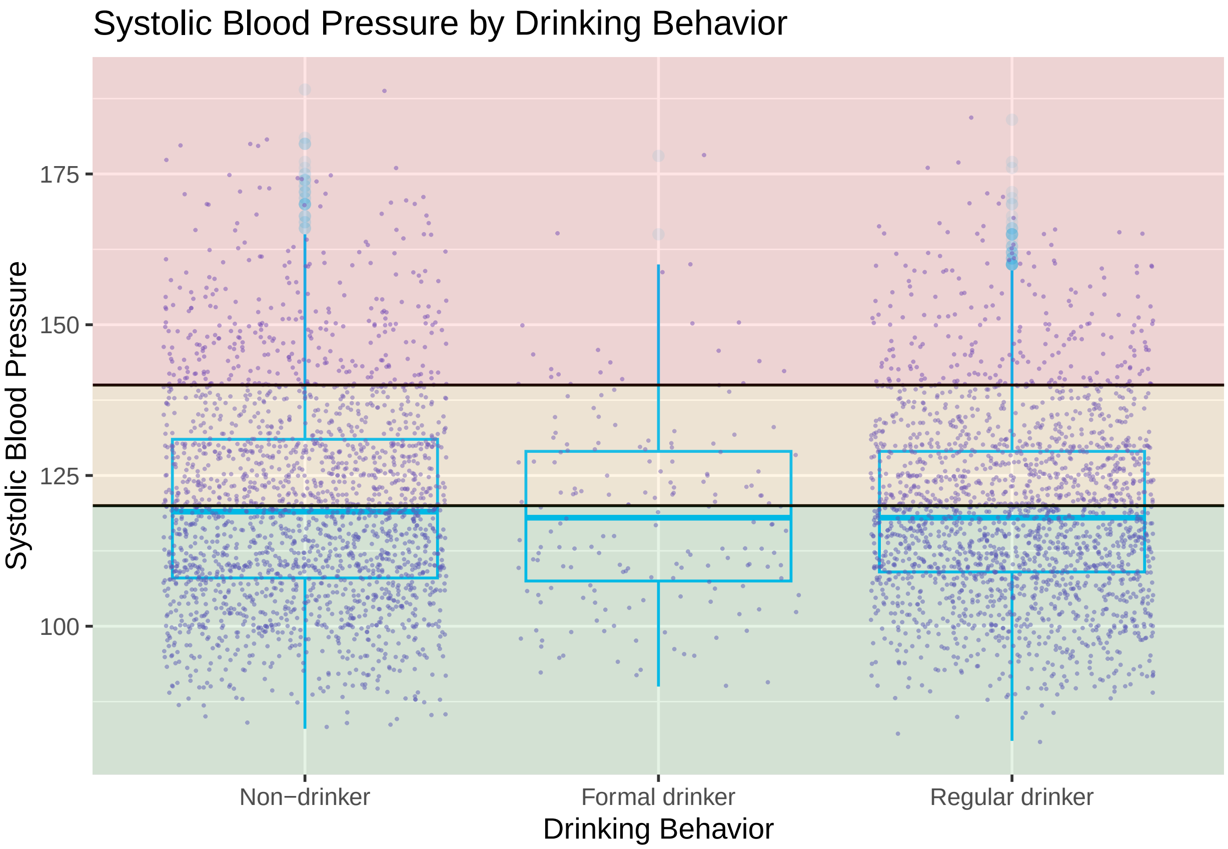
```r
# set cutoff lines for SBP
sbp_cutoff <- data.frame(yintercept=c(120, 140), Lines=c('Healthy', 'At Risk'))

# Scatter plot of SBP over SMOKE
sbp_smoke_g <- ggplot(model_df, aes(x = SMOKE, y = SBP)) +
    geom_boxplot(color="deepskyblue", alpha=.09) +
    geom_jitter(color="slateblue", size=0.1, alpha=0.5) +
    scale_x_discrete(labels = c('Non-smoker', 'Formal smoker', 'Regular smoker')) +
    labs(x = 'Smoking Behavior', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Smoking Behavior")

sbp_smoke_g + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



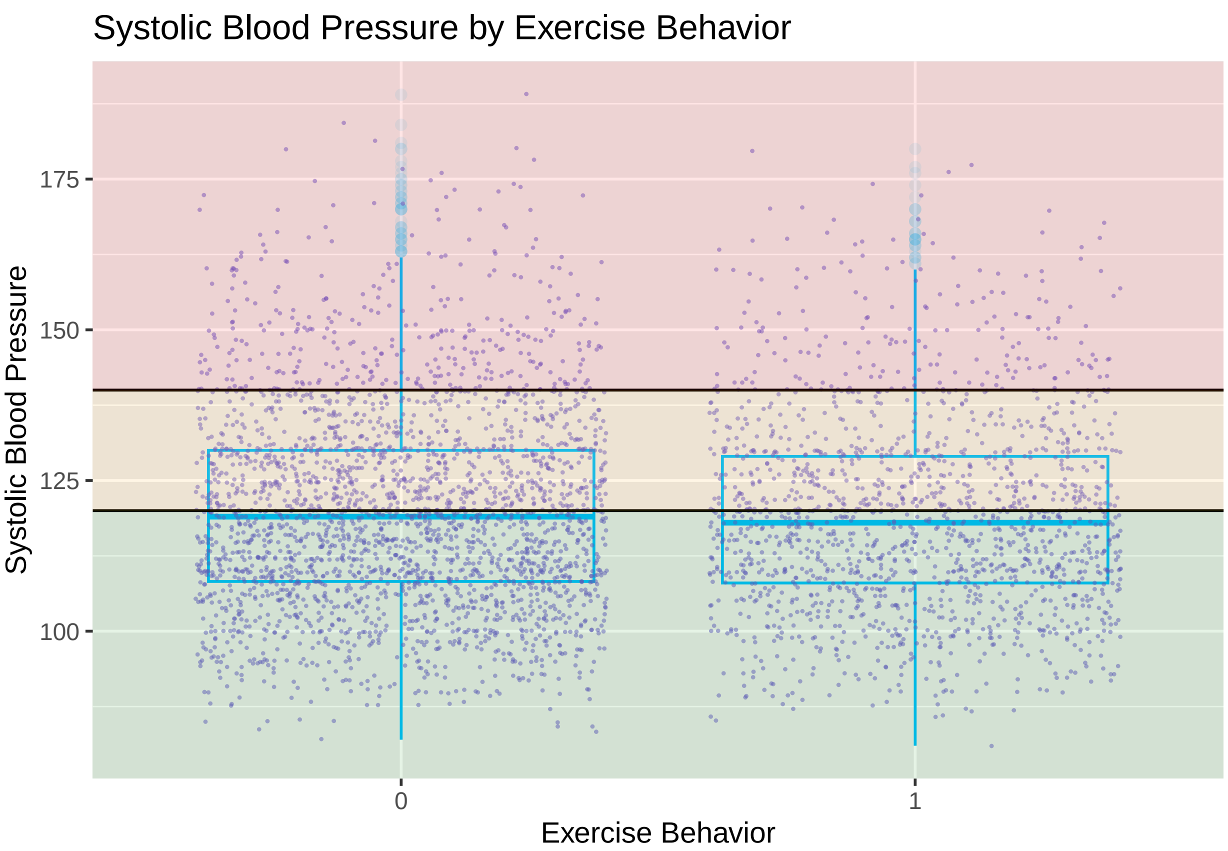
```
# Scatter plot of SBP over DRINK
sbp_drink_g <- ggplot(model_df, aes(x = DRINK, y = SBP)) +
    geom_boxplot(color="deepskyblue", alpha=.09) +
    geom_jitter(color="slateblue", size=0.1, alpha=0.5) +
    scale_x_discrete(labels = c('Non-drinker', 'Formal drinker', 'Regular drinker')) +
    labs(x = 'Drinking Behavior', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Drinking Behavior")

sbp_drink_g + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



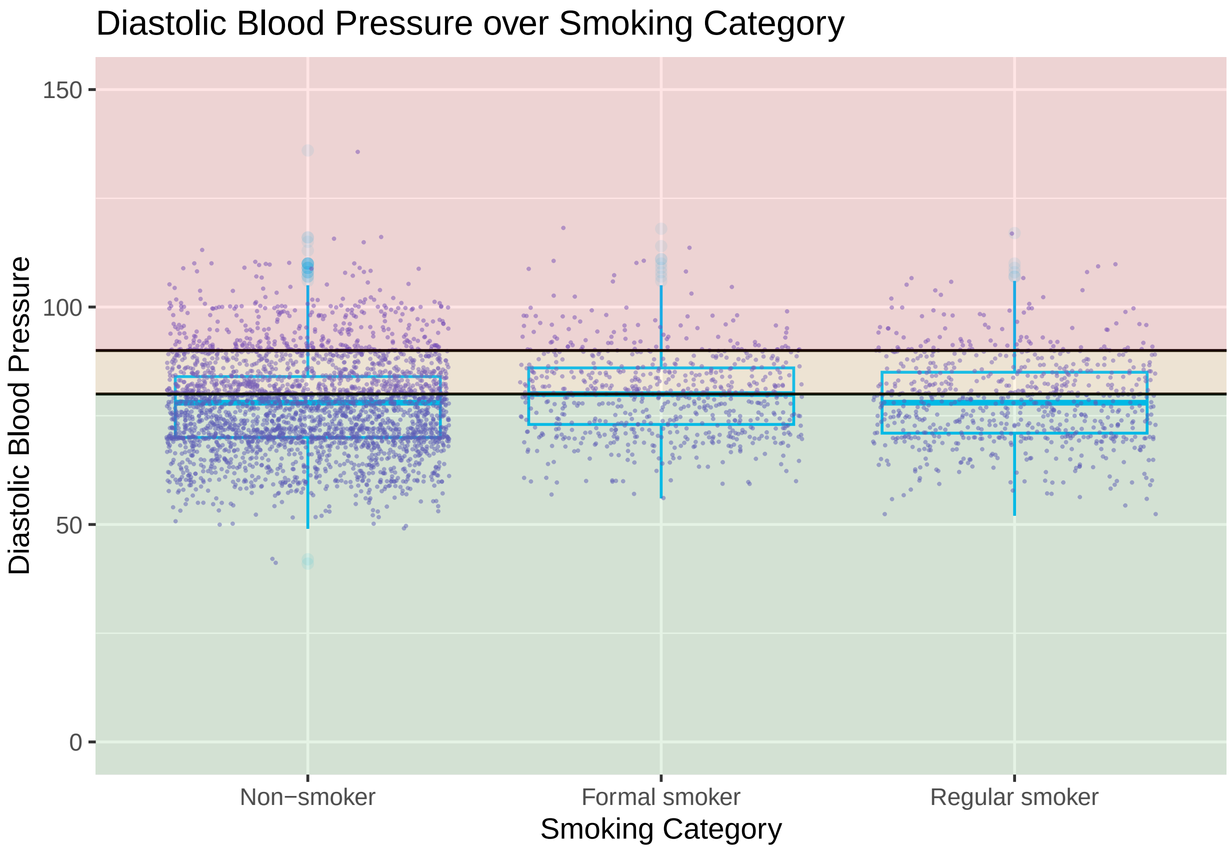
```r
# Scatter plot of SBP over EXER
sbp_exer_g <- ggplot(model_df, aes(x = EXER, y = SBP)) +
    geom_boxplot(color="deepskyblue", alpha=.09) +
    geom_jitter(color="slateblue", size=0.1, alpha=0.5) +
#    scale_x_discrete(labels = c('Does frequent high intensity exercise', 'Does NOT do frequent high in
    labs(x = 'Exercise Behavior', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure by Exercise Behavior")

sbp_exer_g + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```



```
# Out oc curiosity:
# Scatter plot of DBP over SMOKE
dbp_g <- ggplot(data=subset(clean_df, !is.na(SMOKE)), aes(x = SMOKE, y = DBP)) +
    geom_boxplot(color="deepskyblue", alpha=.09) +
    geom_jitter(color="slateblue", size=0.1, alpha=0.5) +
    ylim(0, 150) +
    scale_x_discrete(labels = c('Non-smoker', 'Formal smoker', 'Regular smoker')) +
    labs(x = 'Smoking Category', y = "Diastolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Diastolic Blood Pressure over Smoking Category")

dbp_cutoff <- data.frame(yintercept=c(80, 90), Lines=c('Healthy', 'At Risk'))

dbp_g + geom_hline(aes(yintercept=yintercept, line=Lines), dbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 80, 90),
           ymax = c(80, 90, Inf), fill = c("green4", "orange", "red"), alpha = .1, color = NA)
```

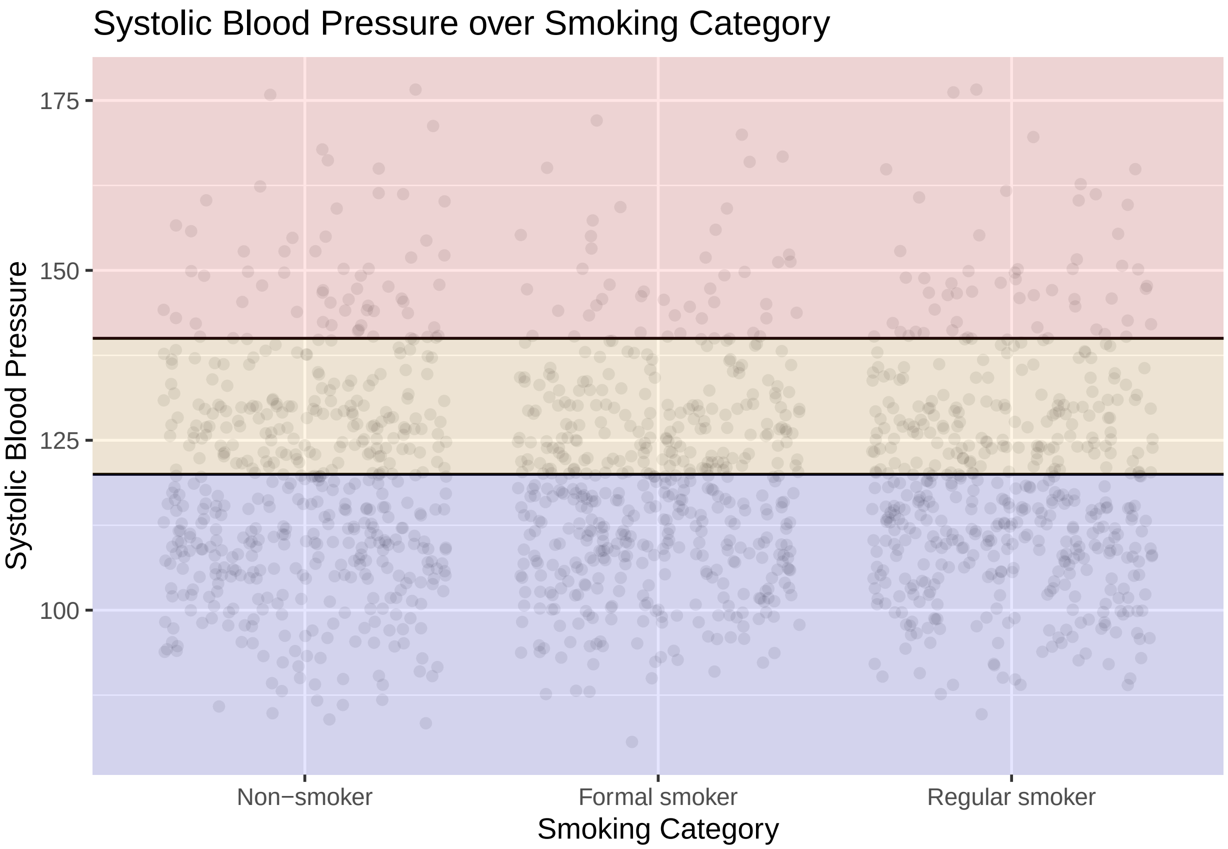
```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), dbp_cutoff):
## Ignoring unknown aesthetics: line

## Warning: Removed 3 rows containing non-finite values (`stat_boxplot()`).

## Warning: Removed 3 rows containing missing values (`geom_point()`).
```

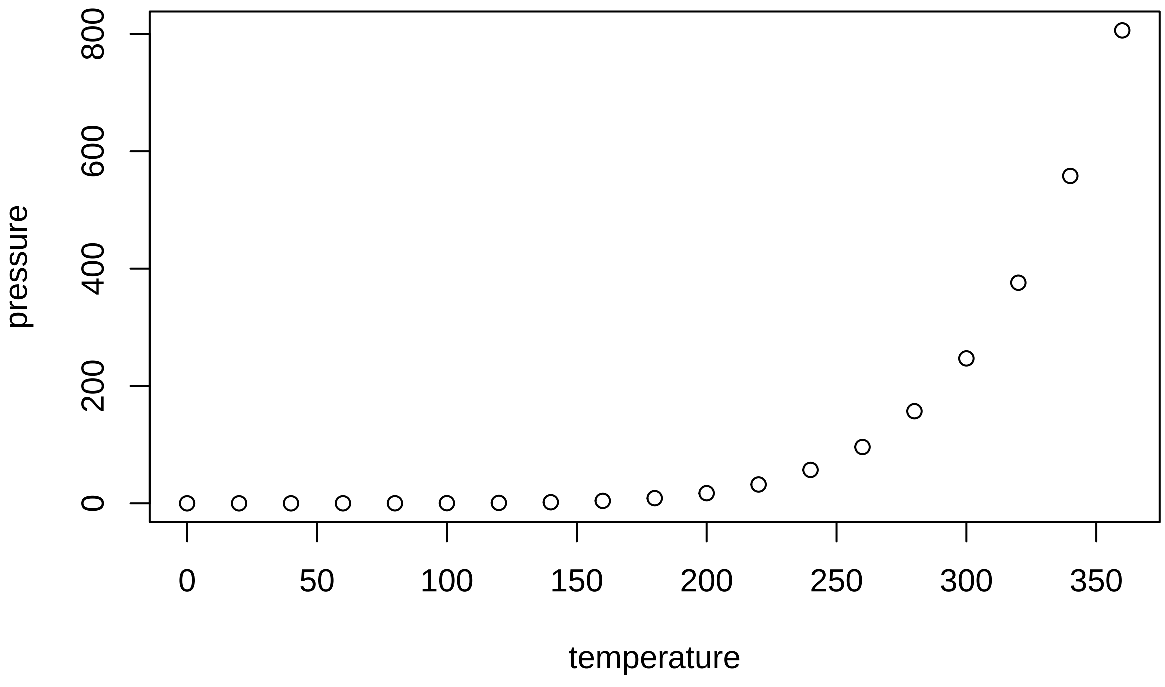


Scatter plot of SBP over SMOKE with equal n in each category - just experimenting

```
# Randomly sample 500 observations from each category in SMOKE
smokeran_df <- model_df %>% group_by(SMOKE) %>% slice_sample(n=500)

sbpran_g <- ggplot(smokeran_df, aes(x = SMOKE, y = SBP)) +
    geom_jitter(alpha=0.08) +  # Add points
    scale_x_discrete(labels = c('Non-smoker', 'Formal smoker', 'Regular smoker')) +
    labs(x = 'Smoking Category', y = "Systolic Blood Pressure") +  # Labels for x and y axes
    ggtitle("Systolic Blood Pressure over Smoking Category")

# Adding SBP cutoff lines. Healthy, At Risk, Hypertension
sbpran_cutoff <- data.frame(yintercept=c(120, 140), Lines=c('Healthy', 'At Risk'))

sbpran_g + geom_hline(aes(yintercept=yintercept, line=Lines), sbp_cutoff) +
  annotate("rect", xmin = -Inf, xmax = Inf, ymin = c(-Inf, 120, 140),
           ymax = c(120, 140, Inf), fill = c("blue", "orange", "red"), alpha = .1, color = NA)
```

```
## Warning in geom_hline(aes(yintercept = yintercept, line = Lines), sbp_cutoff):
## Ignoring unknown aesthetics: line
```





Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.