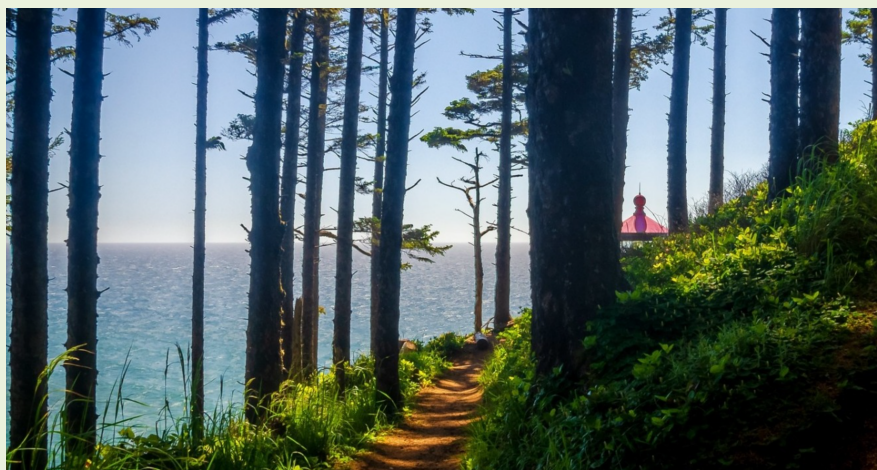


BIOSTATS 640 – Introduction to R
Fall 2023

<https://people.umass.edu/biep640w/webpages/demonstrations.html>



Source: <https://nature.desktopnexus.com/get/2496307/?t=pc3pdp642uonpprag38pgatu46453d46a5eb6c>

12
Forest Plots
December 1, 2023

Dataset used
hersdata.Rdata

		Page
1.	The Heart and Estrogen/Progestin Replacement Study (HERS): hersdata.Rdata	2
2.	Introduction to Forest Plots	3
3.	Univariate Distribution, by Group	6
4.	Multiple Predictor Normal Theory Regression	11
5.	Multiple Predictor Logistic Regression	13
6.	Introduction to {forestploter}	16

Packages used: **tidyverse**, **Hmisc**, **summarytools**, **ggplot2**, **gridExtra**, **sjPlot**, **forestploter**

1. Introduction to The Heart and Estrogen/progestin Replacement Study (HERS)

[hersdata.Rdata](#)

In this illustration, we will work with a larger subset of the HERS study.

Source: Hulley S, Grady D, Bush T, Furberg C, Herrington D, Riggs B and Vittinghoff E (1998). Randomized trial of estrogen plus progestin for secondary prevention of heart disease in postmenopausal women. The Heart and Estrogen/progestin Replacement Study. *Journal of the American Medical Association*, **280**(7), 605-613.

In the HERS study, Hulley et al. (1998) sought to determine if exercise, a modifiable behavior, might lower the risk of diabetes in non-diabetic women who were at risk of developing the disease. The question is a complex one because there are many risk factors for diabetes. Moreover, the type of woman who chooses to exercise may be related in other ways to risk of diabetes, apart from the fact of her exercise habit. For example, women who exercise regularly are typically younger and have lower body mass index (BMI); these characteristics also confer a risk benefit with respect to diabetes. Finally, the benefit of exercise may be mediated through a reduction of body mass index. Vittinghoff, Glidden, Shiboski and McCulloch (2005) consider portions of this data in their 2005 text, *Regression Methods in Biostatistics: Linear, Logistic, Survival and Repeated Measures Models* (Springer).

The subset **hersdata.Rdata** has n=2,763 observations on 37 variables. We will be using the following 4 variables.

Data Dictionary

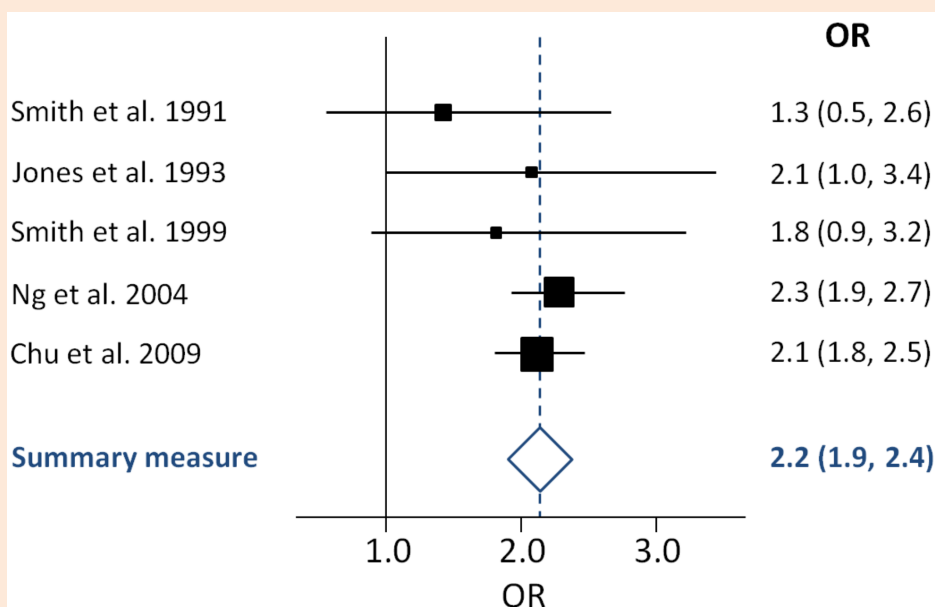
Position	Variable	Variable Label	Type	Codes	Missing data
	age	Age in years	numeric	Range: [44, 79]	None
	physact	Physical Activity compared to women of same age	character	"much less active" "somewhat less active" "about as active" "somewhat more active" "much more active"	None
	glucose	Fasting Glucose, mg/dl	numeric	Range: [67, 294]	None
	diabetes	Diabetes	character	"no" "yes"	None

2. Introduction to Forest Plots

Forest plots, also called **blobbograms** (*who came up with that!!*) were developed for use in meta-analyses.

A meta-analysis forest plot is a visual summary of the findings of the multiple studies together (e.g. a side-by-side plot of the separate "study-specific" odds ratios together with 95% CI) together with a meta-analysis pooled summary.

Example 1: Forest Plot - Meta Analysis

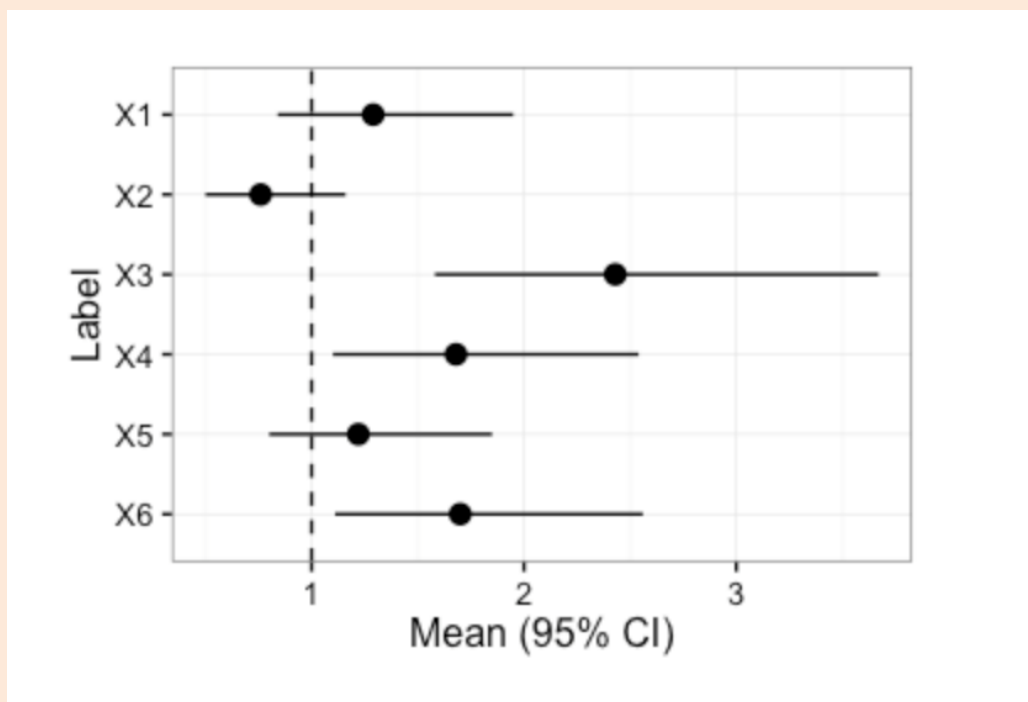


(Source: https://en.wikipedia.org/wiki/Forest_plot#/media/File:Generic_forest_plot.png)

Lucky for us, **forest plots** can also be used for many other data visualizations! These include, but are not limited to:

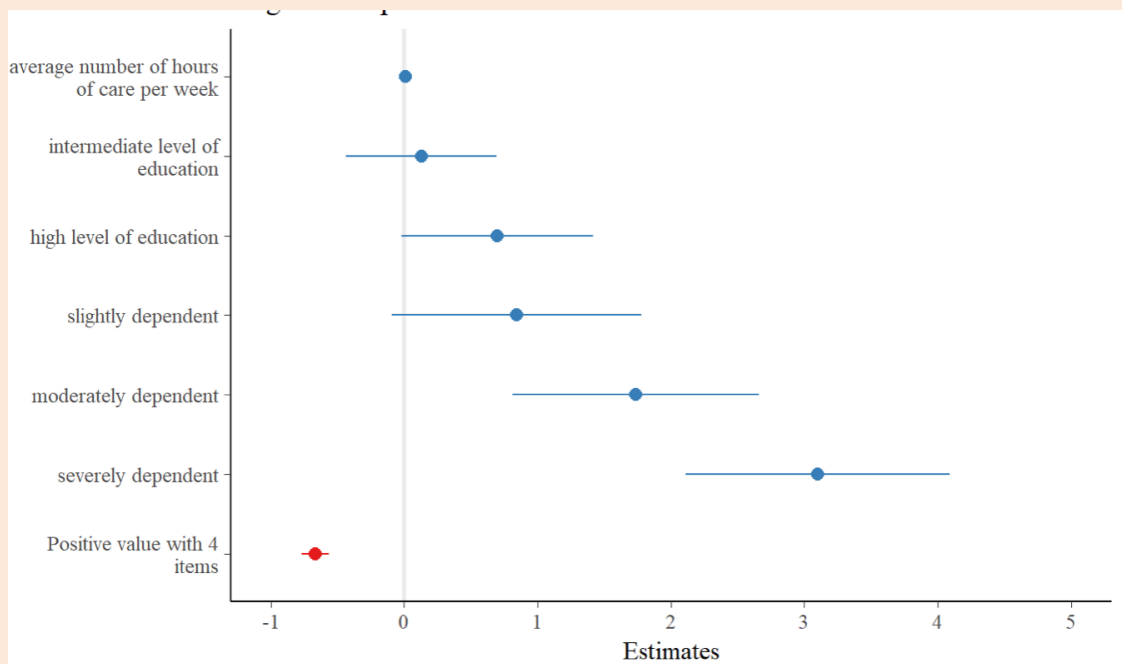
- Univariate distribution, by group;
- Multiple predictor normal theory regression;
- Multiple predictor logistic regression ; and
- Multiple predictor survival analysis

Example 2: Forest Plot - Univariate Distribution, by Group



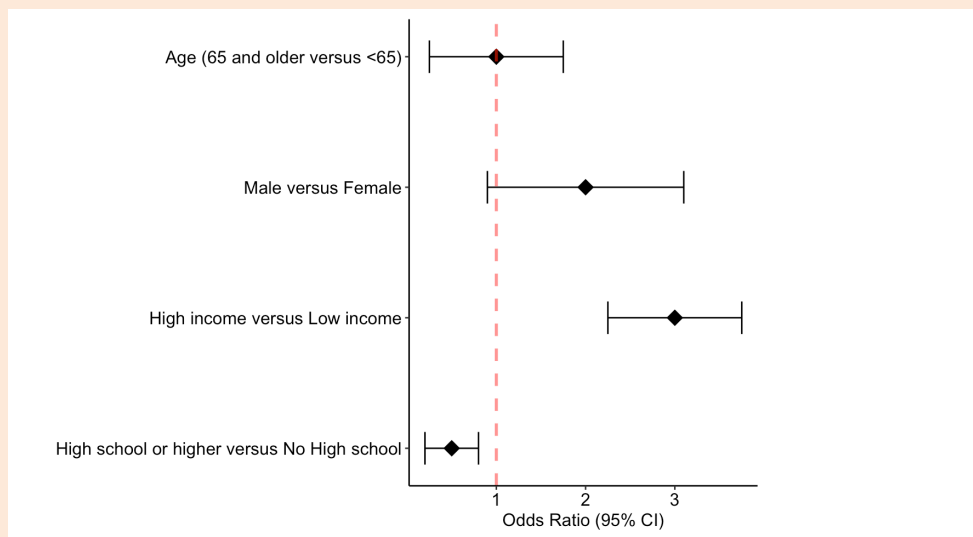
(Source: <https://stackoverflow.com/questions/38062650/forest-plot-for-a-beginner-simple-example-using-ggplot2-edited>)

Example 3: Forest Plot - Multiple Predictor Normal Theory Regression



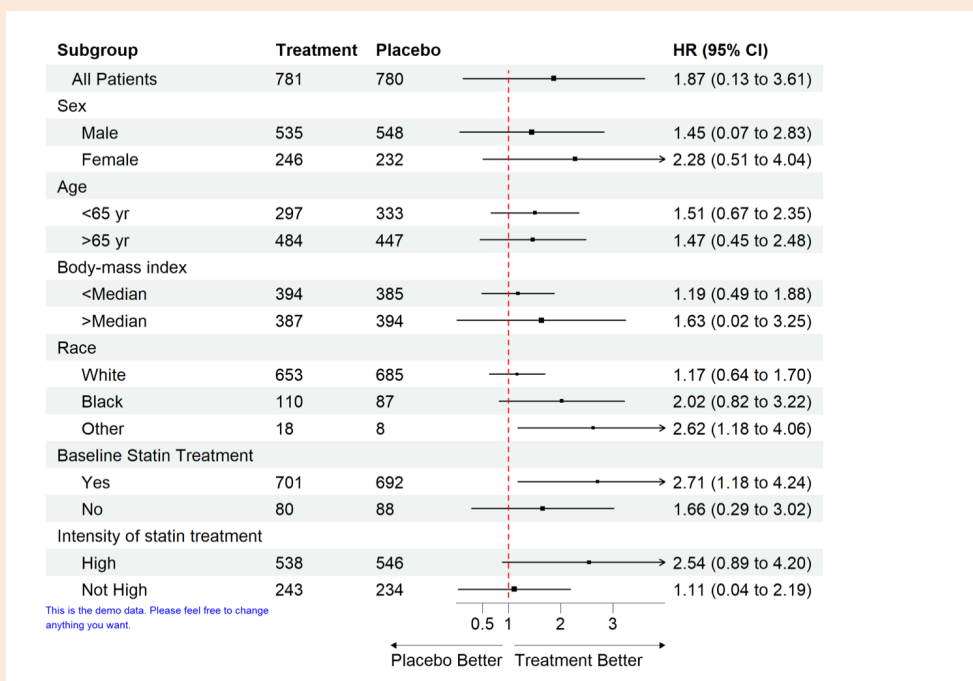
(Source: http://www.strengejacke.de/sjPlot/reference/plot_model.html)

Example 4: Forest Plot - Multiple Predictor Logistic Regression



(Source: https://rpubs.com/mbounthavong/forest_plots_r)

Example 5: Forest Plot - Multiple Predictor Survival Analysis - Clinical Trial of Intervention with Covariates



(Source: <https://github.com/adayim/forestploter/blob/main/man/figures/README-example-1.png>)

3. Univariate Distribution, by Group

```
load R data. inspect structure
load(file="hersdata.Rdata")
hersdata <- as.data.frame(hersdata)
#str(hersdata) #str() to inspect dataframe structure. Remove leading hashtag to execute.

convert character to factor, set order of display
library(tidyverse)
library(Hmisc)
library(summarytools)

mydata1 <- hersdata %>%
  select(physactf, glucose)

mydata1 <- mydata1 %>%
  mutate(physactf = factor(physactf)) %>% # factor() to convert character to factor
  mutate(physactf = recode_factor(physactf, # recode_factor() to set desired order
    "much less active" = "Much less active", # KEY: "old" = "new"
    "somewhat less active" = "Somewhat less active",
    "about as active" = "About as active",
    "somewhat more active" = "Somewhat more active",
    "much more active" = "Much more active"))

Hmisc::label(mydata1$physactf) <- "Comparative Physical Activity" # Label() in {Hmisc} to set variable labels in plot
freq(mydata1$physactf) # freq() in {summarytools}

## Frequencies
## mydata1$physactf
## Label: Comparative Physical Activity
## Type: Factor
##
```

	Freq	% Valid	% Valid Cum.	% Total	% Total Cum.
Much less active	197	7.13	7.13	7.13	7.13
Somewhat less active	503	18.20	25.33	18.20	25.33
About as active	919	33.26	58.60	33.26	58.60
Somewhat more active	838	30.33	88.93	30.33	88.93
Much more active	306	11.07	100.00	11.07	100.00
<NA>	0			0.00	100.00
Total	2763	100.00	100.00	100.00	100.00

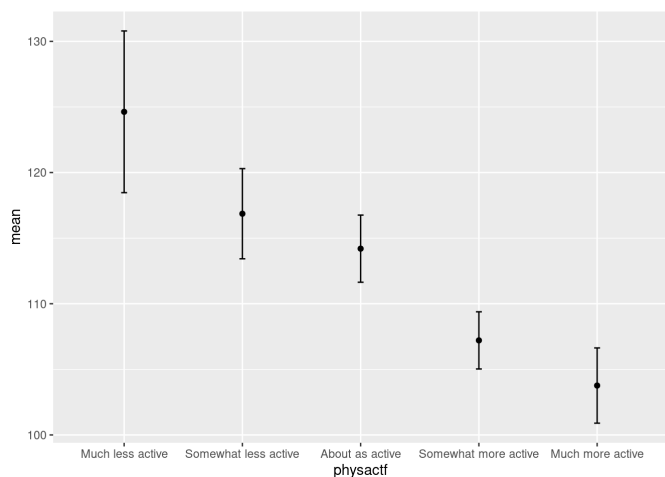
```
##
## -----
##      Much less active    197      7.13      7.13      7.13      7.13
##      Somewhat less active 503     18.20     25.33     18.20     25.33
##      About as active     919     33.26     58.60     33.26     58.60
##      Somewhat more active 838     30.33     88.93     30.33     88.93
##      Much more active    306     11.07    100.00     11.07    100.00
##      <NA>                 0          0.00      0.00     100.00
##      Total               2763    100.00    100.00    100.00    100.00

Univariate distribution, by group - Method 1: {ggplot2} of means +/- 95% CI, BASIC

library(tidyverse)
library(ggplot2)

# create plotdata1 = dataframe for plotting
plotdata1 <- mydata1 %>%
  group_by(physactf) %>% # for each level of physactf, get
  summarise(
    n = sum(!is.na(glucose)), # sample size (complete observations only)
    mean = mean(glucose, na.rm=TRUE), # mean (remove missings)
    sd = sd(glucose, na.rm=TRUE), # standard deviation (remove missings)
    se = sd/sqrt(n), # standard error
    ci95se = 1.96*se, # 1.96 * standard error
    lowerci = mean - ci95se, # lower CI value
    upperci = mean + ci95se # upper CI value

# basic plot
ggplot(data=plotdata1) +
  aes(x=physactf) + # x = group variable that defines levels
  aes(y=mean) +
  geom_errorbar(aes(ymin=lowerci, ymax=upperci), width=.05) + # plot CI limits
  geom_point() # plot mean
```



Univariate distribution, by group - Method 1: {ggplot2} means +/- 95% CI, with AESTHETICS

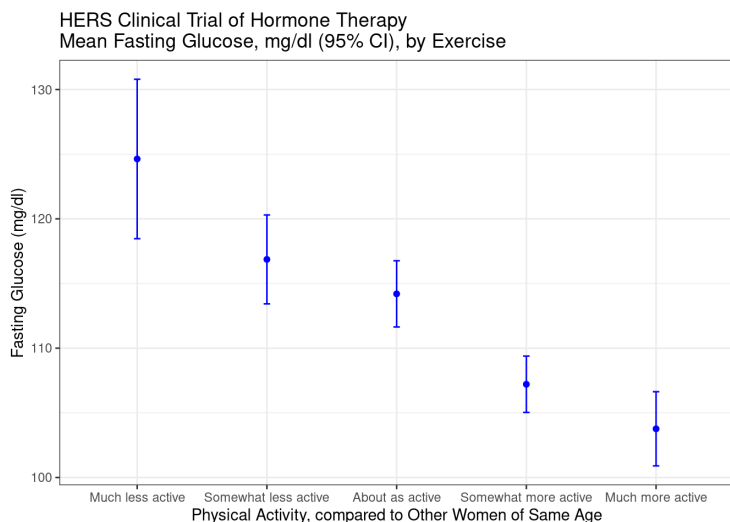
```
library(tidyverse)
```

```
library(ggplot2)
```

Data for plotting is the same as above: plotdata1

```
ggplot(data=plotdata1) +
  # required layers
  aes(x=physactf) +
  aes(y=mean) +
  geom_errorbar(aes(ymin=lowerci, ymax=upperci), width=.05, color="blue") +
  geom_point(color="blue") +

  # add aesthetics
  ggtitle("HERS Clinical Trial of Hormone Therapy\nMean Fasting Glucose, mg/dl (95% CI), by Exercise") +
  xlab("Physical Activity, compared to Other Women of Same Age")+
  ylab("Fasting Glucose (mg/dl)") +
  theme_bw()
```



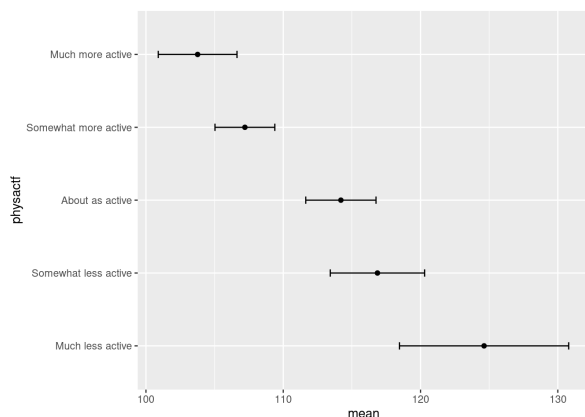
Univariate distribution by group - Method 2: {ggplot2} forest plot, BASIC

```
library(tidyverse)
library(ggplot2)

# Data for plotting is the same as above: plotdata1

ggplot(data=plotdata1) +
  aes(y=physactf) +
  aes(x=mean, xmin=lowerci, xmax=upperci) +
  geom_point() +
  geom_errorbarh(height=.1)
```

For horizontal error bars, y = Levels of grouping var
For horizontal error bars, x = mean +/- 95% CI of outcome



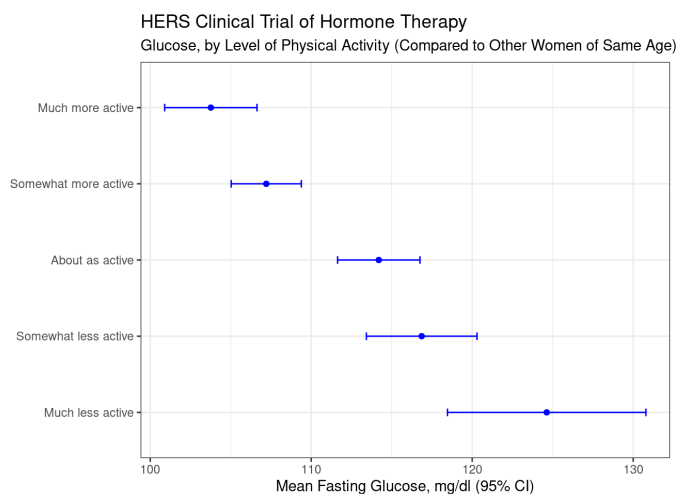
Univariate distribution by group - Method 2: {ggplot2} forest plot, with AESTHETICS

```
library(tidyverse)
library(ggplot2)

# Data for plotting is the same as above: plotdata1
ggplot(data=plotdata1) +
  # required layers
  aes(y=physactf) +
  aes(x=mean, xmin=lowerci, xmax=upperci) +
  geom_point(color="blue") +
  geom_errorbarh(color="blue",height=.1) +

  # add aesthetics
  xlab("Mean Fasting Glucose, mg/dl (95% CI)") +
  ylab(" ") +
  labs(title = "HERS Clinical Trial of Hormone Therapy",
        subtitle = "Glucose, by Level of Physical Activity (Compared to Other Women of Same Age)") +
  theme_bw()
```

For horizontal CI, y = Levels of grouping var
For horizontal CI, x = mean +/- 95% CI of outcome



Univariate distribution by group - Method 2: {ggplot2} and {gridExtra} plot, with AESTHETICS + TABLE

```
library(tidyverse)
library(ggplot2)
library(gridExtra)

# REQUIRED PRELIMINARY: To produce an accompanying table, must create a new dataframe for plotting: plotdata2
plotdata2 <- mydata1 %>%
  group_by(physactf) %>%
  summarise(
    n = sum(!is.na(glucose)),
    mean = mean(glucose, na.rm=TRUE),
    sd = sd(glucose, na.rm=TRUE),
    se = sd/sqrt(n),
    ci95se = 1.96*se,
    lowerci = mean - ci95se,
    upperci = mean + ci95se) %>%
  mutate(lower = as.character(round(lowerci,digits=1)),
    upper = as.character(round(upperci,digits=1)),
    CI = str_c(lower, " ", upper))

# for each level of physactf, get:
# get sample size (complete observations only)
# mean (remove missings)
# standard deviation (remove missings)
# standard error
# 1.96 * standard error
# Lower CI value
# upper CI value
# Needed for table: # digits to show
# Needed for table: CI limits as character
# Needed for table: format display of CI

## STEP 1: create forest plot display of mean +/- CI
myforestplot <- ggplot(data=plotdata2) +
  aes(y = physactf) +
  aes(x = mean) +
  geom_point(shape = 18, size = 3, color="blue") +
  geom_errorbarh(aes(xmin = lowerci, xmax = upperci), height = 0.10, color="blue") +
  xlab("Mean Fasting Glucose, mg/dl (95% CI)") +
  ylab(" ") +
  theme_bw() +
  theme(panel.border = element_blank(),
    panel.background = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "black"),
    axis.text.y = element_text(size = 12, colour = "black"),
    axis.text.x.bottom = element_text(size = 12, colour = "black"),
    axis.title.x = element_text(size = 12, colour = "black"))

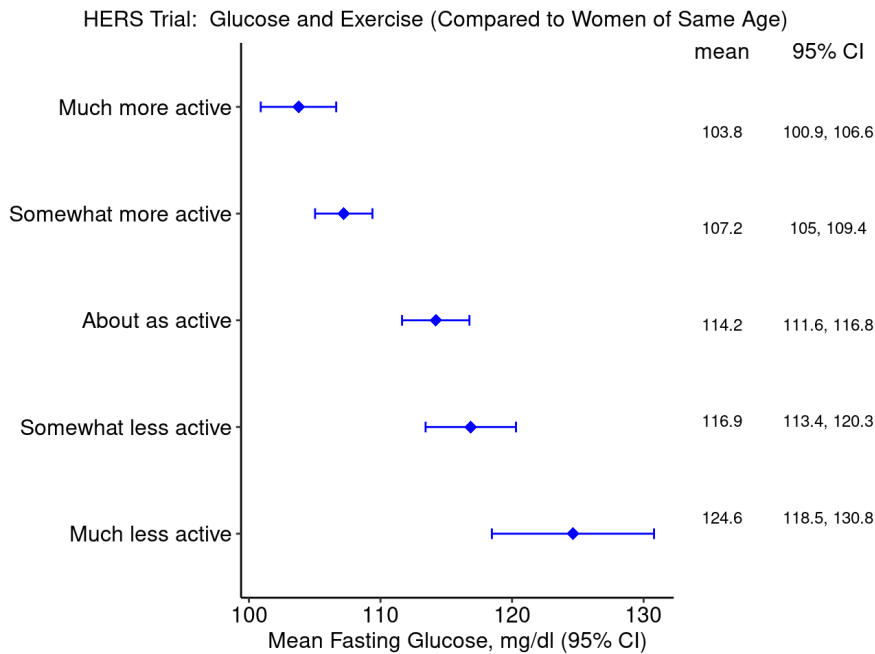
# For horizontal CI, y = Levels of grouping var
# For horizontal CI, x = mean +/- CI of outcome

## STEP 2: Create the table-base palette
table_base <- ggplot(plotdata2) +
  aes(y=physactf) +
  ylab(NULL) +
  xlab(" ") +
  theme(plot.title = element_text(hjust = 0.5, size=12),
    axis.text.x = element_text(color="white", hjust = -3, size = 25), ## This is used to help with alignment (Yikes!)
    axis.line = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks = element_blank(),
    axis.title.y = element_blank(),
    legend.position = "none",
    panel.background = element_blank(),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.background = element_blank())

## mean table
mymean_table <- table_base +
  labs(title = "space") +
  geom_text(aes(y = physactf, x = 1, label = sprintf("%0.1f", round(mean, digits = 1))), size = 3.2) +
  ggtitle("mean")
```

```
## 95% CI table
myci_table <- table_base +
  geom_text(aes(y = physactf, x = 1, label = CI), size = 3.2) +
  ggtitle("95% CI")

lay <- matrix(c(1,1,1,1,1,1,1,1,1,2,3,3), nrow = 1) # Carol to research and explain later
grid.arrange(myforestplot, mymean_table, myci_table,
  layout_matrix = lay,
  top="HERS Trial: Glucose and Exercise (Compared to Women of Same Age)")
```



4. Multiple Predictor Normal Theory Regression

```
multiple predictor regression: fit model using explicit 0/1 design vars for physactf (REFERENT = "about as active")
library(tidyverse)
library(Hmisc)

# create dataframe for multiple predictor regression
mydata2 <- hersdata %>%
  select(glucose, age, physact)

mydata2 <- mydata2 %>%
  mutate(age10 = age/10, na.rm=TRUE) %>%
  mutate(physactf = factor(physact)) %>%
  mutate(physactf = recode_factor(physactf,
    "much less active" = "Much less active",
    "somewhat less active" = "Somewhat less active",
    "about as active" = "About as active",
    "somewhat more active" = "Somewhat more active",
    "much more active" = "Much more active")) %>%

  # design vars for activity (reference = "about as active")
  mutate(Iactive_muchless = ifelse(physactf=="Much less active",1,0)) %>%
  mutate(Iactive_somewhatless = ifelse(physactf=="Somewhat less active",1,0)) %>%
  mutate(Iactive_somewhatmore = ifelse(physactf=="Somewhat more active",1,0)) %>%
  mutate(Iactive_muchmore = ifelse(physactf=="Much more active",1,0)) %>%

  select(glucose,age10,physactf,
    Iactive_muchless, Iactive_somewhatless,Iactive_somewhatmore, Iactive_muchmore ) %>%

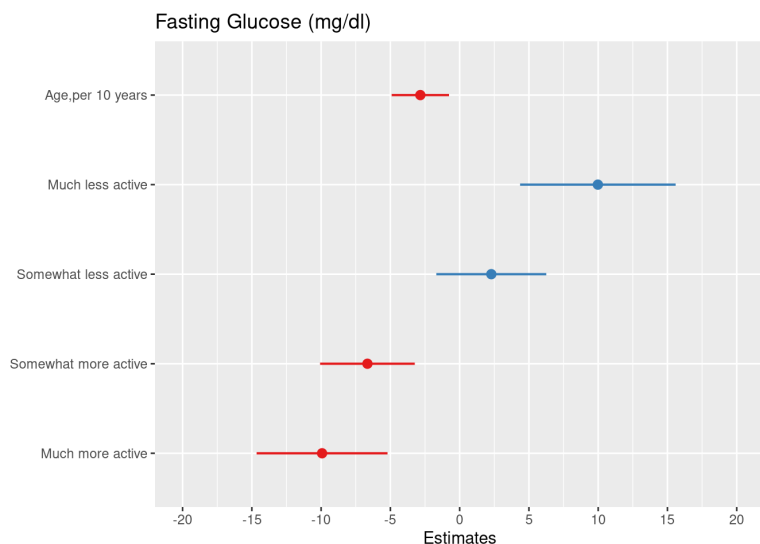
  na.omit()

Hmisc::label(mydata2$physactf) <- "Comparative Physical Activity"
Hmisc::label(mydata2$age10) <- "Age,per 10 years"
Hmisc::label(mydata2$glucose) <- "Fasting Glucose (mg/dl)"
Hmisc::label(mydata2$Iactive_muchless) <- "Much less active"
Hmisc::label(mydata2$Iactive_somewhatless) <- "Somewhat less active"
Hmisc::label(mydata2$Iactive_somewhatmore) <- "Somewhat more active"
Hmisc::label(mydata2$Iactive_muchmore) <- "Much more active"

myfit <- lm(glucose ~ age10 +
  Iactive_muchless + Iactive_somewhatless + Iactive_somewhatmore + Iactive_muchmore,
  data=mydata2)
summary(myfit)
## Call:
## lm(formula = glucose ~ age10 + Iactive_muchless + Iactive_somewhatless +
##   Iactive_somewhatmore + Iactive_muchmore, data = mydata2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -89.599 -20.192 -11.261   3.726 191.590
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept)    133.056       7.119   18.690 < 0.000000e+000 ***
## age10           -2.838       1.056   -2.687    0.007245 **
## Iactive_muchless    9.977       2.863    3.485    0.000500 ***
## Iactive_somewhatless 2.285       2.024    1.129    0.258971
## Iactive_somewhatmore -6.657       1.743   -3.819    0.000137 ***
## Iactive_muchmore   -9.931       2.410   -4.121    0.0000388 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.4 on 2757 degrees of freedom
## Multiple R-squared:  0.02593,    Adjusted R-squared:  0.02417
## F-statistic: 14.68 on 5 and 2757 DF,  p-value: 0.000000e+00003114
```

multiple predictor normal theory regression - {sjPlot} forest plot of betas and CI: BASIC

```
library(sjPlot)
plot_model(myfit) # plot_model in {sjPlot}
```



Note: A limitation of this forest plot is that it doesn't tell the reader that the referent phsyactf is "about as active" So, how might you communicate this? Change labels of the design vars so as to be more clear? Other?

multiple predictor normal theory regression - {sjPlot} forest plot of betas and CI: with AESTHETICS

```
library(sjPlot)
plot_model(myfit,
  show.values=TRUE,
  value.offset = .2,
  vline.color = "black",
  title = "Multiple Linear Regression Model of Fasting Glucose (mg/dl)",
  axis.title="Beta (95% CI)") # show betas
                             # position beta values just above the estimate
                             # show reference null line at beta=0
```



5. Multiple Predictor Logistic Regression

```

fit multiple predictor logistic regression using explicit 0/1 design vars for physactf (REFERENT = "about as active")
library(tidyverse)
library(Hmisc)

# create dataframe for multiple predictor Logistic regression
mydata3 <- hersdata %>%
  select(diabetes, age, physact)

mydata3 <- mydata3 %>%
  mutate(age10 = age/10, na.rm=TRUE) %>%
  mutate(physactf = factor(physact)) %>%
  mutate(physactf = recode_factor(physactf,
    "much less active" = "Much less active",
    "somewhat less active" = "Somewhat less active",
    "about as active" = "About as active",
    "somewhat more active" = "Somewhat more active",
    "much more active" = "Much more active")) %>%

  # design vars for activity (reference = "about as active")
  mutate(Iactive_muchless = ifelse(physactf=="Much less active",1,0)) %>%
  mutate(Iactive_somewhatless = ifelse(physactf=="Somewhat less active",1,0)) %>%
  mutate(Iactive_somewhatmore = ifelse(physactf=="Somewhat more active",1,0)) %>%
  mutate(Iactive_muchmore = ifelse(physactf=="Much more active",1,0)) %>%

  # 0/1 outcome for Logistic regression
  mutate(diabetes01 = ifelse(diabetes=="yes",1,0)) %>%

  select(diabetes01, age10, physactf,
    Iactive_muchless, Iactive_somewhatless, Iactive_somewhatmore, Iactive_muchmore ) %>%

  na.omit()

Hmisc::label(mydata3$physactf) <- "Comparative Physical Activity"
Hmisc::label(mydata3$age10) <- "Age, per 10 years"
Hmisc::label(mydata3$Iactive_muchless) <- "Much less active"
Hmisc::label(mydata3$Iactive_somewhatless) <- "Somewhat less active"
Hmisc::label(mydata3$Iactive_somewhatmore) <- "Somewhat more active"
Hmisc::label(mydata3$Iactive_muchmore) <- "Much more active"

mylogistic <- glm(diabetes01 ~ age10 +
  Iactive_muchless + Iactive_somewhatless + Iactive_somewhatmore + Iactive_muchmore,
  data=mydata3,
  family=binomial)

summary(mylogistic)

## Call:
## glm(formula = diabetes01 ~ age10 + Iactive_muchless + Iactive_somewhatless +
##   Iactive_somewhatmore + Iactive_muchmore, family = binomial,
##   data = mydata3)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1973  -0.7925  -0.6581   1.2477   1.9109
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.27703    0.44469  -0.623   0.533307
## age10        -0.11078    0.06618  -1.674   0.094168 .
## Iactive_muchless  0.82216    0.16171   5.084 0.00000369 ***
## Iactive_somewhatless 0.42149    0.11952   3.527 0.000421 ***

```

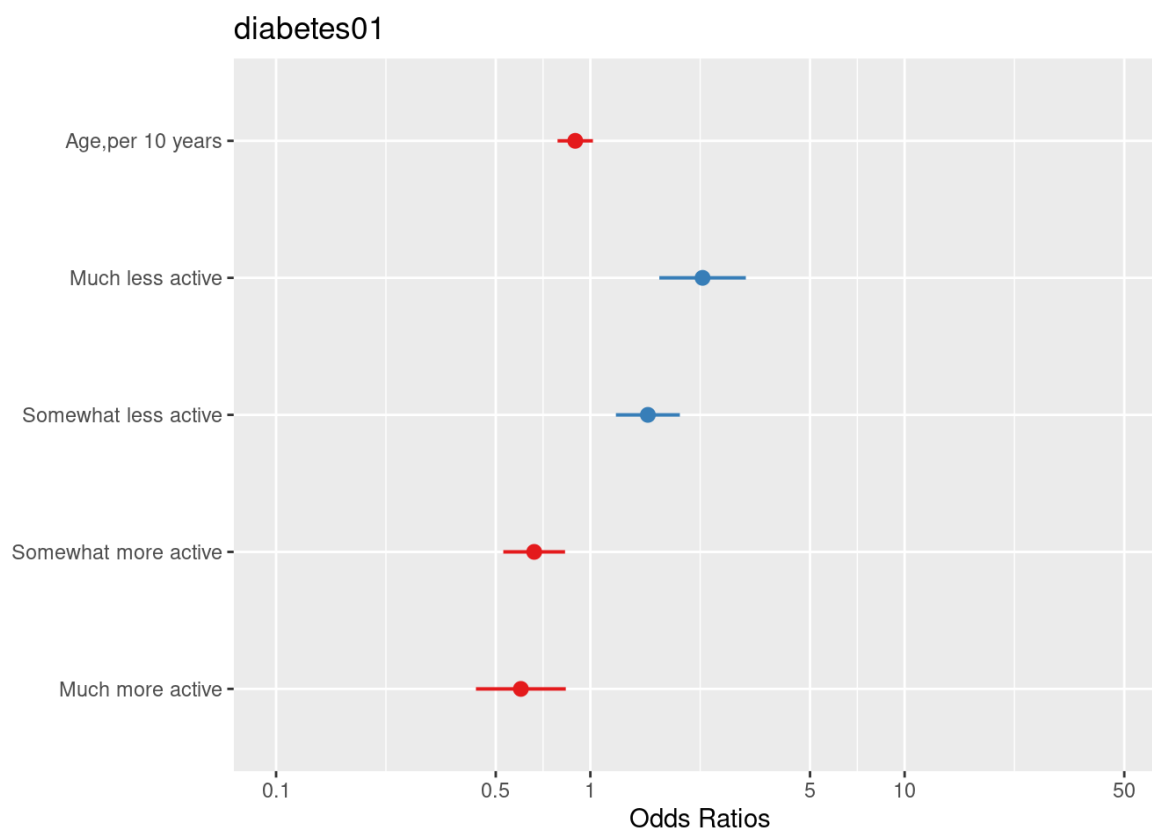
```
## Iactive_somewhatmore -0.41159    0.11541 -3.566    0.000362 ***
## Iactive_muchmore    -0.50914    0.16793 -3.032    0.002431 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 3192.8  on 2762  degrees of freedom
## Residual deviance: 3097.1  on 2757  degrees of freedom
## AIC: 3109.1
##
## Number of Fisher Scoring iterations: 4

exp(cbind(OR = coef(mylogistic), confint(mylogistic)))      # coef(MODEL) and confint(MODEL) to show OR's and 95% CI
##              OR      2.5 %    97.5 %
## (Intercept)    0.7580347 0.3163075 1.8089752
## age10          0.8951383 0.7862682 1.0192502
## Iactive_muchless 2.2754095 1.6558410 3.1232801
## Iactive_somewhatless 1.5242284 1.2054237 1.9261512
## Iactive_somewhatmore 0.6625938 0.5278417 0.8300102
## Iactive_muchmore 0.6010115 0.4291451 0.8298138
```

multiple predictor logistic regression - {sjPlot} forest plot of OR's and 95% CI's: BASIC

```
library(sjPlot)
```

```
plot_model(mylogistic)
```



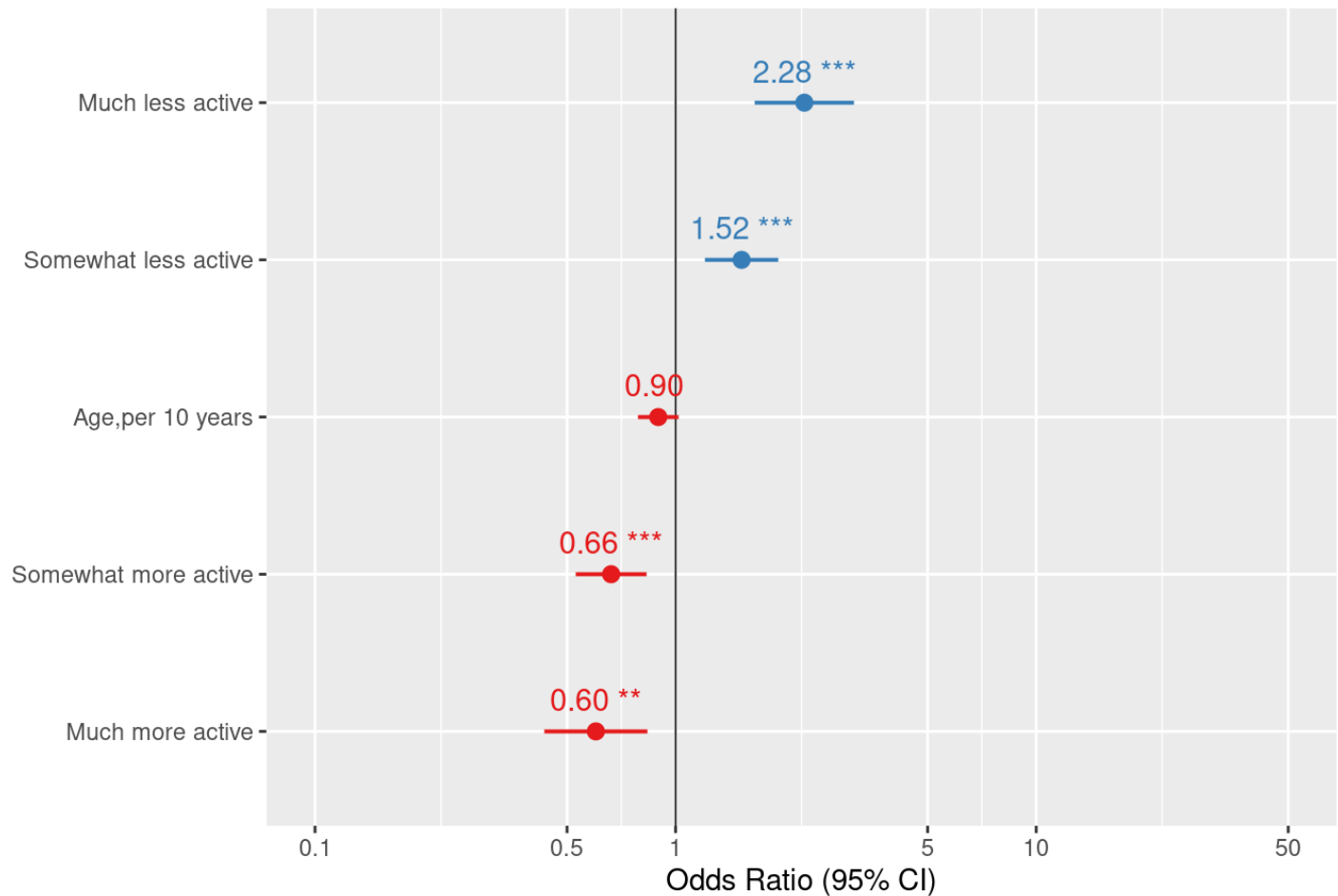
Here, too. A limitation of this forest plot is that it doesn't tell the reader that the referent phsyactf is "about as active" Additional refinement of R coding is needed so that the plot is more clear to the reader!

multiple predictor logistic regression - forest plot of OR's and 95% CI's: with AESTHETICS

```
library(sjPlot)
```

```
plot_model(mylogistic,
  show.values=TRUE,           # show OR's
  value.offset = .2,          # position OR values just above the estimate
  sort.est = TRUE,            # sort them (nice!)
  vline.color = "black",      # show null line at OR = 1
  title = "Multiple Logistic Regression of Event of Diabetes",
  axis.title="Odds Ratio (95% CI)")
```

Multiple Logistic Regression of Event of Diabetes



6. Introduction to {forestploter}

Introduction to {forestploter}

Step 1. create mystats = dataframe w desired statistics

```
library(tidyverse)

# convert physact to factor. set order
mydata <- hersdata %>%
  # convert physact to factor. set order.
  mutate(physactf = factor(physact)) %>%
  mutate(physactf = recode_factor(physactf,
    "much less active" = "Much less active",
    "somewhat less active" = "Somewhat less active",
    "about as active" = "About as active",
    "somewhat more active" = "Somewhat more active",
    "much more active" = "Much more active"))
  # factor( ) to convert character to factor
  # recode_factor() to set desired order of levels
  # "old" = "new"

# Keep needed columns
mydata <- mydata %>%
  select(physactf, glucose)

# get statistics
mystats <- mydata %>%
  group_by(physactf) %>%
  summarise(
    n = sum(!is.na(glucose)),
    mean = mean(glucose, na.rm=TRUE),
    sd = sd(glucose, na.rm=TRUE),
    se = sd/sqrt(n),
    ci95se = 1.96*se,
    lowerci = mean - ci95se,
    upperci = mean + ci95se)
  # for each level of physactf, get
  # sample size (complete observations only)
  # mean (remove missings)
  # standard deviation (remove missings)
  # standard error
  # 1.96 * standard error
  # Lower CI value
  # upper CI value

mystats

## # A tibble: 5 × 8
##   physactf      n mean    sd    se ci95se lowerci upperci
##   <fct>    <int> <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 1 Much less active    197  125.  44.1  3.14   6.16    118.    131.
## 2 Somewhat less active  503  117.  39.3  1.75   3.44    113.    120.
## 3 About as active    919  114.  39.6  1.31   2.56    112.    117.
## 4 Somewhat more active  838  107.  32.2  1.11   2.18    105.    109.
## 5 Much more active   306  104.  25.6  1.46   2.87    101.    107.
```

Introduction to {forestploter}

Step 2. Required Preliminary - create df_fig = dataframe for plotting

```
library(tidyverse)
library(ggplot2)
library(forestploter)

# Step 2a. Create mystats$plotcol = blank column to hold plot
mystats$plotcol <- paste(rep(" ", 20), collapse = " ")

# Step 2b. Create mystats$`(95% CI)` = new var to hold both mean and CI
mystats$`(95% CI)` <- ifelse(is.na(mystats$se), "",
                             sprintf("%.2f (%.2f - %.2f)",
                                     mystats$mean, mystats$lowerci, mystats$upperci))

# Step 2c. Create df_fig = subset of the dataframe mystats
# Source dataframe = mystats
# New dataframe = df_fig
# KEY: df_fig will have ALL the rows of mystats
# KEY: df_fig will have the FOLLOWING selected columns of mystats. These MUST be in the following specified order
# df_fig COLUMN 1 = physactf (source: mystats column 1)
# df_fig COLUMN 2 = plotcol (source: mystats column 9- this is the blank column that will have plot!)
# df_fig COLUMN 3 = mean and 95% CI (source: mystats column 10)
df_fig <- mystats[,c(1,9,10)]

df_fig

## # A tibble: 5 × 3
##   physactf      plotcol      `(95% CI)`
##   <fct>      <chr>      <chr>
## 1 Much less active "      " 124.63 (118.47...
## 2 Somewhat less active "      " 116.86 (113.43...
## 3 About as active "      " 114.20 (111.64...
## 4 Somewhat more active "      " 107.21 (105.03...
## 5 Much more active "      " 103.76 (100.90...
```

Introduction to {forestploter}

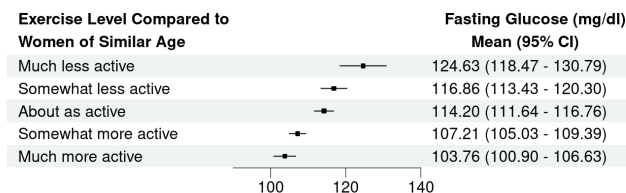
Step 3. Basic plot

```
library(ggplot2)
library(forestploter)

# colnames(DATAFRAME) to set desired column labels.
# colnames(DATAFRAME) <- c("Label for column 1", "Label for column 2", "Label for column 2") w labels in quotes.
# note - you may have to fiddle with spacing to line things up
colnames(df_fig) <- c("Exercise Level Compared to\nWomen of Similar Age",
                     "",
                     "Fasting Glucose (mg/dl)\nMean (95% CI)" ) # NOTE the forced spacing here!

p <- forest(df_fig,
            est=mystats$mean,
            lower=mystats$lowerci,
            upper=mystats$upperci,
            ci_column=2,
            xlim=c(90,140),
            ticks_at=c(100,120,140))

p
```



Not bad. Column 1 above = levels of physact, nicely labeled. Column 2 = plot. Column 3 = mean and 95% CI. You can also see that I juggled the placement of column headings by fooling with `colnames(df_fig)` on previous page.

Introduction to {forestploter}

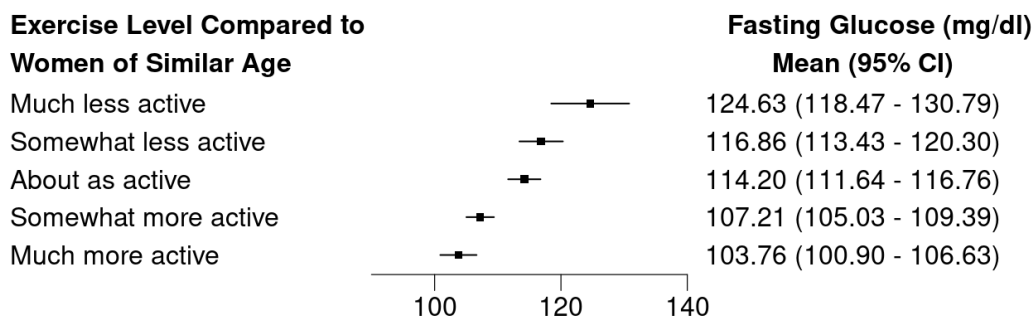
Step 4. Basic + aesthetics

```
library(tidyverse)
library(ggplot2)
library(forestploter)

# add a nice background
# theme
tm <- forest_theme(core=list(bg_params=list(fill=c("white"))),
                    summary_col="black")

# plot
p <- forest(df_fig,
            est=mystats$mean,
            lower=mystats$lowerci,
            upper=mystats$upperci,
            ci_column=2,
            xlim=c(90,140),
            ticks_at=c(100,120,140),
            theme=tm)

p
```



Here, I edited some of the options of `forest()` in `{forestploter}` to obtain a prettier display.