

Supplementary document - The Performance of Hypotension Prediction Index May Be Overestimated Due to Selection Bias

Johannes Enevoldsen and Simon Tilma Vistisen

Contents

Setup	1
Simulation of the selection problem	1
Creating a logistic regression model with only one relevant predictor	4
A logistic regression with a single independent variable will return predictions ranked identically to the independent variable.	9
Create figure 3	10

Setup

```
library(pROC)
library(tidyverse)
# install.packages("faux") # used to simulate data

# A few extra packages are used to create the simulation figure. We load these in
# the relevant section.

theme_set(theme_minimal(base_size = 12))
```

Simulation of the selection problem

OBS: This simulation does not attempt to create realistic data or to accurately preset how Hypotension Prediction Index may have been developed.

We simulate 1000 paired observations, to show what the data selection described in the model-development paper does to the performance of a model that uses MAP as a predictor.

The specific parameters of the simulation are simply chosen to place most data in a relevant interval: between 55 and 95 mmHg. The correlation coefficient ($r = 0.6$) is chosen to give MAP a modest predictive power, similar to what is presented by Davies *et al* (<http://doi.org/10.1213/ANE.0000000000004121>) However, any reasonable distribution of data, and any correlation coefficient between 0 and ~0.9, will convey the intended message.

```
set.seed(123)
correlated_map <- faux::rnorm_multi(n = 1000,
                                   mu = c(75, 75),
                                   sd = c(10, 10),
                                   r = 0.6,
                                   # "map_predictor" represents the current map value
                                   # "map_outcome" represents what map is 5 minutes from now.
                                   varnames = list("map_predictor", "map_outcome")
                                   )
```

We filter the simulated data to remove observations where the predictor MAP is below 65 mmHg. This reduces the AUC of all analyses, but the difference between the biased and unbiased selections remain clear.

We also remove a few extreme points simply to reduce white space in the plots. Again, this simulation is not intended to mimic real data!

```
d <- filter(correlated_map,
            # We remove points where map is below 65 in the predictor
            map_predictor > 65,
            # Remove extreme points
            map_predictor < 105,
            map_outcome > 45,
            map_outcome < 105
            )
```

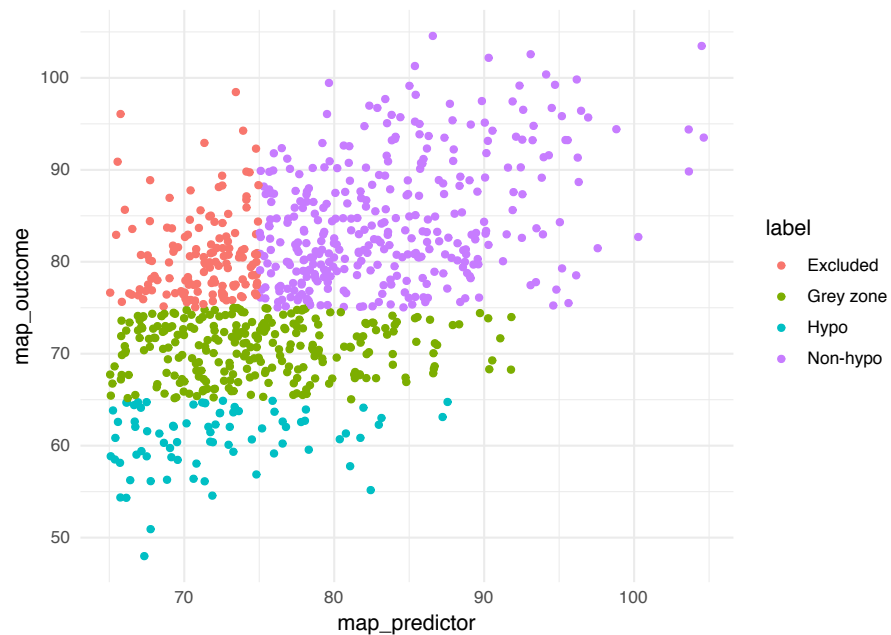
We label the relevant zones of the data. Each row is the combination of an event (hypotensive or non-hypotensive) and the sample used to predict the event (map_predictor).

```
d <- mutate(d,
            # If map_outcome is < than 65, this is represented as a hypotensive event
            hypo = map_outcome < 65,
            # If map_outcome is between 65 and 75, this is a greyzone event.
            grey = map_outcome >= 65 & map_outcome < 75,
            # If map_outcome and map_predictor is >= 75, this is a nonevent.
            nonhypo = map_outcome >= 75 & map_predictor >= 75
            )
```

Here is a plot to show the applied labels.

```
ggplot(d, aes(map_predictor, map_outcome, color = case_when(hypo ~ "Hypo",
                                                             grey ~ "Grey zone",
                                                             nonhypo ~ "Non-hypo",
                                                             TRUE ~ "Excluded")) +

  geom_point() +
  labs(color = "label")
```



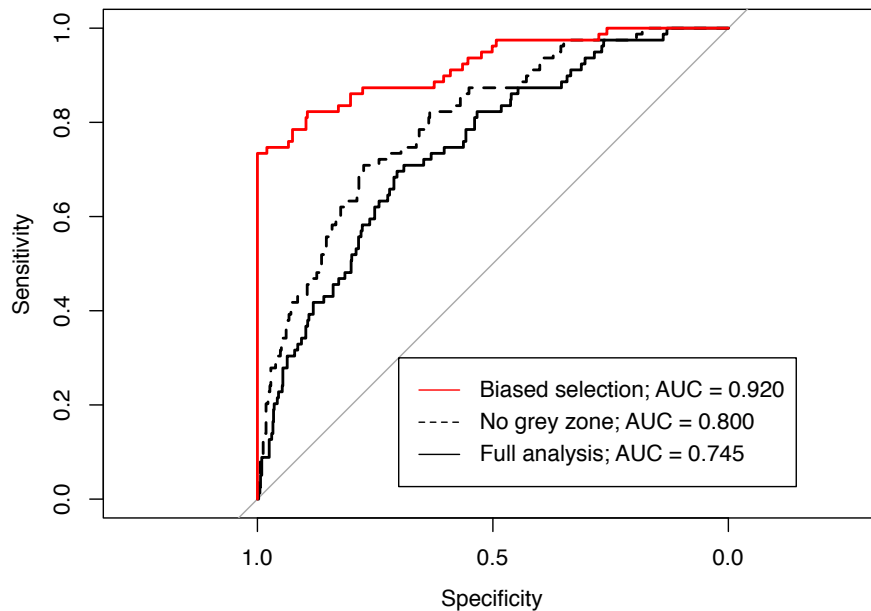
Now we can make the ROC analyses.

```
# With all data
roc_analysis_full <- roc(hypo~map_predictor, data = d)

# Excluding grey zone
roc_analysis_grey <- roc(hypo~map_predictor, data = filter(d, !grey))

# Include only data that is either event or non-event per model-development definition
roc_analysis_biased <- roc(hypo~map_predictor, data = filter(d, hypo | nonhypo))

plot(roc_analysis_full)
plot(roc_analysis_grey, lty = 2, add = TRUE)
plot(roc_analysis_biased, col = "red", add = TRUE)
legend(
  x = 0.7, y = 0.3,
  legend = c(
    sprintf("Biased selection; AUC = %.3f", auc(roc_analysis_biased)),
    sprintf("No grey zone; AUC = %.3f", auc(roc_analysis_grey)),
    sprintf("Full analysis; AUC = %.3f", auc(roc_analysis_full))
  ),
  lty = c(1, 2, 1),
  col = c("red", "black", "black")
)
```



In the end of this document we paste the remaining code used to create the paper's figure 3.

Creating a logistic regression model with only one relevant predictor

Here is a simple demonstration of how a logistic regression based prediction model could get its entire predictive ability from a single predictor, despite including multiple predictors.

In this example, we simulate a scenario, where only MAP is a relevant predictor of hypotension. The remaining variables are random noise. In reality, it is of course entirely possible that HPI is also relevantly influenced by other variables than MAP.

We generate data identically to presented above, only this time we generate 5000 observations.

```
library(pROC)

set.seed(1)
correlated_map_large <-
  faux::rnorm_multi(
    n = 5000,
    mu = c(75, 75),
    sd = c(10, 10),
    r = 0.6,
    varnames = list("map_predictor", "map_outcome")
  ) |>
  as_tibble() |>
  filter(map_predictor > 65,
         map_predictor < 105,
         map_outcome > 45,
```

```

    map_outcome < 105)

d_large <- mutate(
  correlated_map_large,
  hypo = map_outcome < 65,
  grey = map_outcome >= 65 & map_outcome < 75,
  nonhypo = map_outcome >= 75 & map_predictor >= 75
)

d_model <- filter(d_large, hypo | nonhypo)

n_obs <- nrow(d_model)

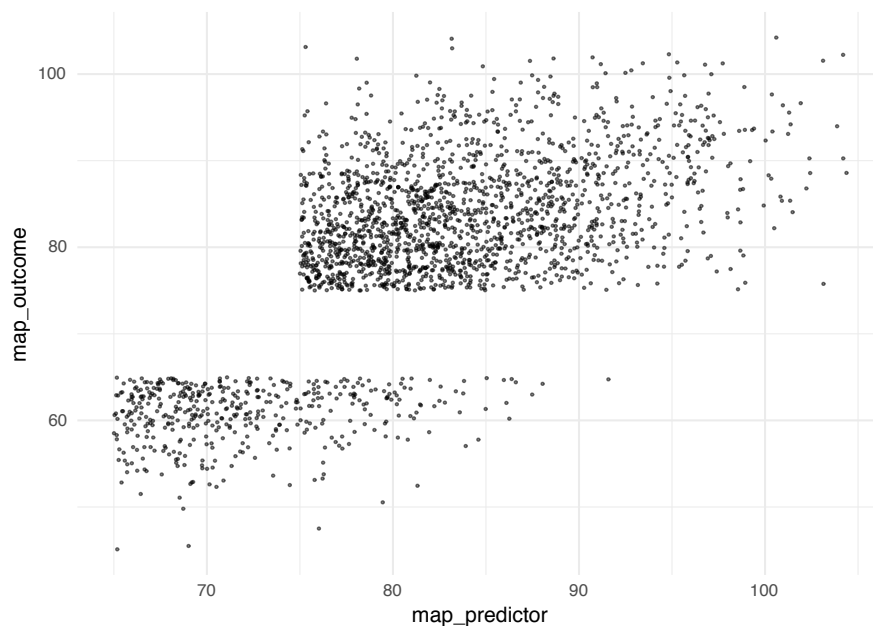
```

After applying the biased data selection, we have 2209 observations.

```

ggplot(d_model, aes(map_predictor, map_outcome)) +
  geom_point(size = 0.5, alpha = 0.5)

```



In addition to MAP, we will add 9 variables that convey no information about the risk of future hypotension. We simply draw these independently from a normal distribution. We use a standard deviation of 10, to put the random variables on the same scale as MAP.

```

d_model <- mutate(d_model,
  var1 = rnorm(n = n_obs, sd = 10),
  var2 = rnorm(n = n_obs, sd = 10),
  var3 = rnorm(n = n_obs, sd = 10),
  var4 = rnorm(n = n_obs, sd = 10),
  var5 = rnorm(n = n_obs, sd = 10),

```

```

var6 = rnorm(n = n_obs, sd = 10),
var7 = rnorm(n = n_obs, sd = 10),
var8 = rnorm(n = n_obs, sd = 10),
var9 = rnorm(n = n_obs, sd = 10),
.after = hypo)

head(d_model)

```

```

## # A tibble: 6 x 14
##   map_predictor map_outcome hypo   var1   var2   var3   var4   var5   var6
##         <dbl>      <dbl> <lgl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1         76.2        62.6 TRUE  -8.04    9.57  -12.9    3.65    6.89  -12.7
## 2         75.0        60.0 TRUE -10.6   -2.35   -2.16   -9.96    1.65   -7.57
## 3         84.0        94.5 FALSE -10.4   -0.0817 -0.483   0.951   10.6    1.68
## 4         73.2        62.1 TRUE -11.9    11.8    19.3    9.97  -14.5  -19.2
## 5         78.9        84.3 FALSE  -5.00    0.372   -8.92  -16.2   10.9    9.23
## 6         78.8        81.5 FALSE  -5.25  -11.3   -15.3   19.4   -0.637  -3.54
## # ... with 5 more variables: var7 <dbl>, var8 <dbl>, var9 <dbl>, grey <lgl>,
## #   nonhypo <lgl>

```

We split the data into a training set and a test set (50:50).

```

set.seed(1)
sample_i <- sample(1:n_obs, size = floor(n_obs/2))
d_train <- d_model[sample_i,]
d_test <- d_model[-sample_i,]

```

We fit a logistic regression to the test set.

```

fake_hpi_model <- glm(hypo ~ map_predictor + var1 + var2 + var3 + var4 + var5 + var6 +
  var7 + var8 + var9, data = d_train, family = binomial())

gtsummary::tbl_regression(fake_hpi_model) |>
  gtsummary::as_gt() |>
  gt::as_latex()

```

Characteristic	log(OR) ¹	95% CI ¹	p-value
map_predictor	-0.50	-0.57, -0.43	<0.001
var1	-0.03	-0.05, 0.00	0.030
var2	0.00	-0.03, 0.02	0.8
var3	0.01	-0.01, 0.04	0.3
var4	0.01	-0.02, 0.03	0.5
var5	0.00	-0.02, 0.02	0.9
var6	-0.03	-0.05, 0.00	0.023
var7	0.01	-0.02, 0.03	0.5

var8	0.00	-0.03, 0.02	0.8
var9	0.01	-0.01, 0.03	0.4

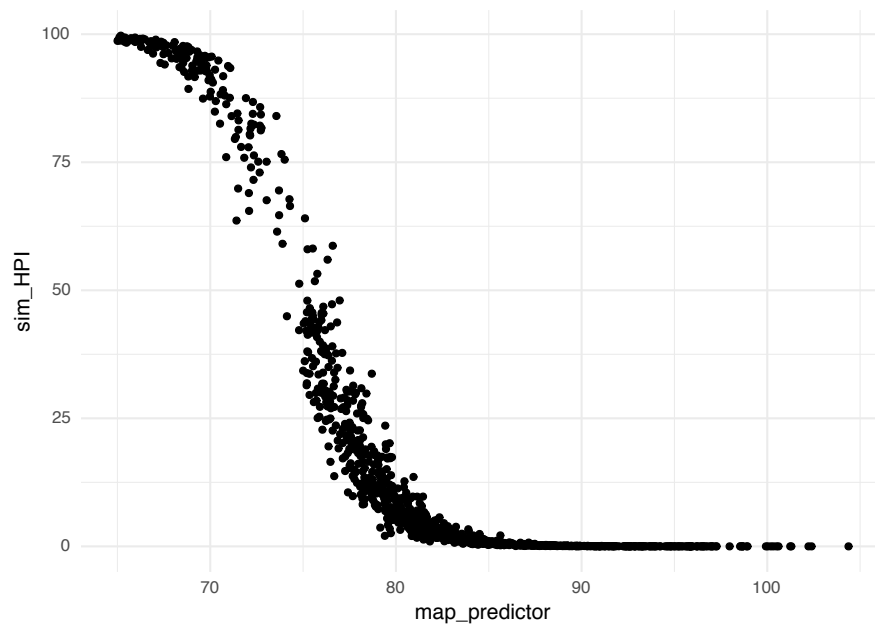
¹OR = Odds Ratio, CI = Confidence Interval

We can see that the model correctly identifies that only MAP is a relevant predictor of hypotension in this simulated data set.

HPI is the predictions from a logistic regression times 100. We calculate these predictions for each observation in the training set and plot it against the predictor MAP.

```
d_train$sim_HPI <- predict(fake_hpi_model, type = "response") * 100

ggplot(d_train, aes(map_predictor, sim_HPI)) +
  geom_point()
```



We use a ROC analysis to see how well this simulated HPI can discriminate future hypotension from future nonhypotension.

```
(roc_hpi_train <- roc(hypo ~ sim_HPI, data = d_train))
```

```
##
## Call:
## roc.formula(formula = hypo ~ sim_HPI, data = d_train)
##
## Data: sim_HPI in 879 controls (hypo FALSE) < 225 cases (hypo TRUE).
## Area under the curve: 0.9377
```

Now, we use the same fitted model to create predictions on the test set.

```
d_test$sim_HPI <- predict(fake_hpi_model, newdata = d_test, type = "response") * 100

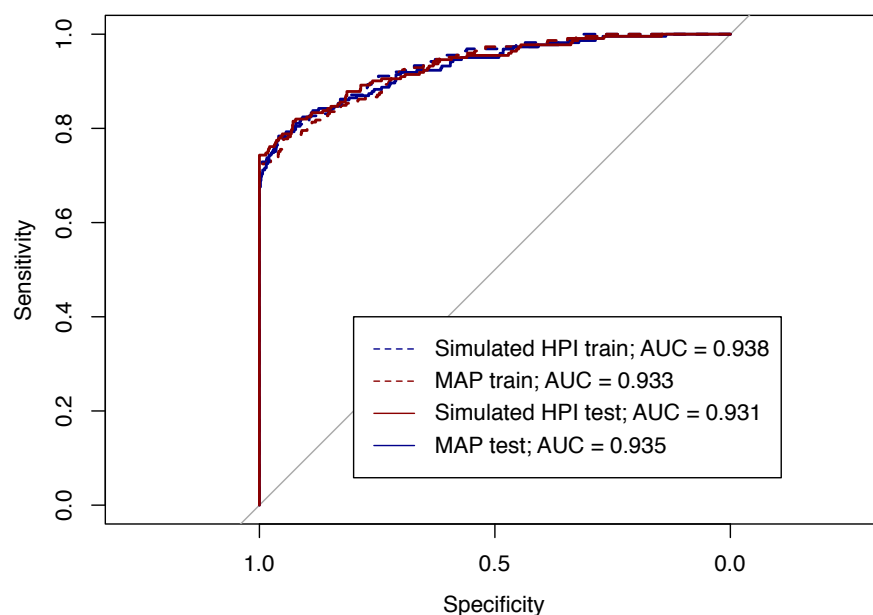
(roc_hpi_test <- roc(hypo ~ sim_HPI, data = d_test))

##
## Call:
## roc.formula(formula = hypo ~ sim_HPI, data = d_test)
##
## Data: sim_HPI in 883 controls (hypo FALSE) < 222 cases (hypo TRUE).
## Area under the curve: 0.9314
```

We also make the same ROC analyses for MAP directly, and plot all four ROC curves.

```
roc_map_train <- roc(hypo ~ map_predictor, data = d_train)
roc_map_test <- roc(hypo ~ map_predictor, data = d_test)

{
  plot(roc_hpi_train, col = "darkblue", lty = 2)
  plot(roc_map_train, col = "darkred", lty = 2, add = TRUE)
  plot(roc_hpi_test, col = "darkblue", add = TRUE)
  plot(roc_map_test, col = "darkred", add = TRUE)
  legend(x = 0.8, y = 0.4,
        legend = c(sprintf("Simulated HPI train; AUC = %.3f", auc(roc_hpi_train)),
                    sprintf("MAP train; AUC = %.3f", auc(roc_map_train)),
                    sprintf("Simulated HPI test; AUC = %.3f", auc(roc_hpi_test)),
                    sprintf("MAP test; AUC = %.3f", auc(roc_map_test))),
        lty = c(2,2,1,1),
        col = c("darkblue", "darkred", "darkred", "darkblue"))
}
```



A logistic regression with a single independent variable will return predictions ranked identically to the independent variable.

As a last example, we make a logistic regression with MAP as the only independent variable. Here, the ROC analysis will be identical to the ROC analysis using MAP directly. A ROC analysis relies only on the ranks of the predictor, and ranks are not changed by a logistic regression with one independent variable (with the exception that they can be reversed, which also does not impact the ROC analysis result). This can also be appreciated from the logistic regression formula:

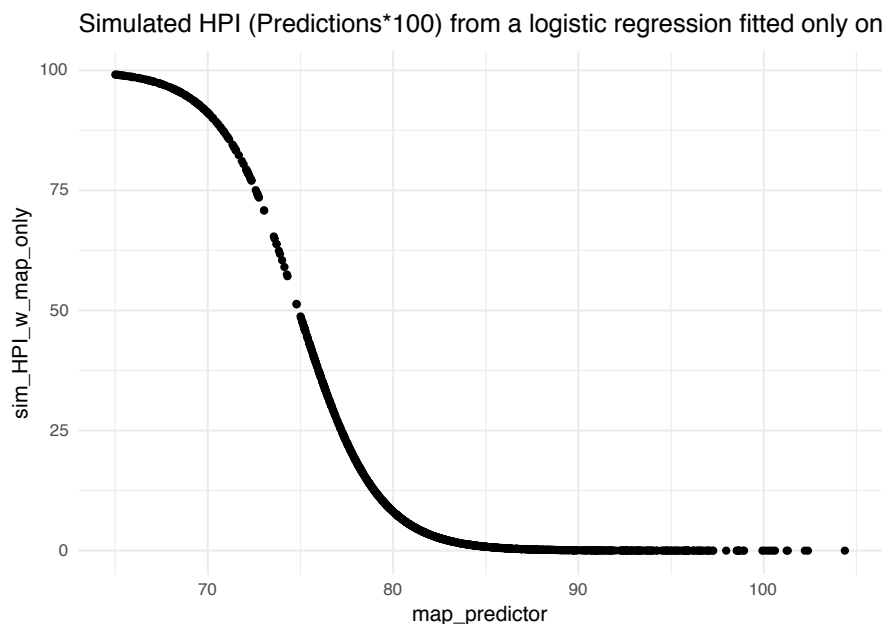
$$HPI_{map\ only} = 100 * \frac{1}{1 + e^{-(\beta_0 + \beta_1 MAP)}}$$

The linear part of a logistic regression ($\beta_0 + \beta_1 MAP$) multiplies the independent variable with a constant, β_1 and adds a constant, β_0 . This can reverse the ranks (if $\beta_1 < 0$) but not otherwise change them. The logisic link function also preserves the ranks (the same does multiplying by 100 to get the final HPI score).

```
map_model <- glm(hypo ~ map_predictor, data = d_train, family = binomial())

d_train <- mutate(d_train, sim_HPI_w_map_only = predict(map_model, type = "response") * 100)

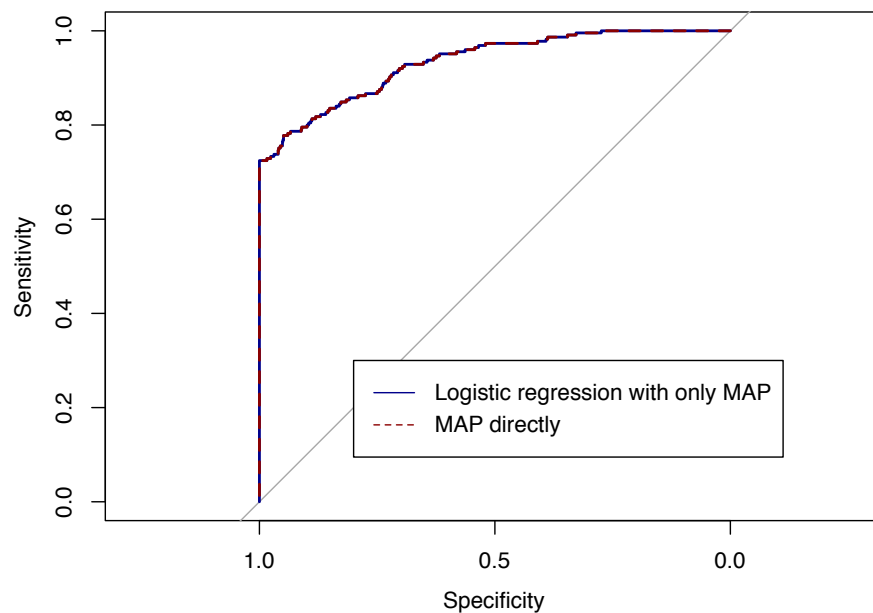
ggplot(d_train, aes(map_predictor, sim_HPI_w_map_only)) +
  geom_point() +
  labs(title = "Simulated HPI (Predictions*100) from a logistic regression fitted only on MAP")
```



Here, we compare the ROC analysis from this simulated HPI based only on MAP with the ROC analysis using MAP directly.

```
roc_map_model <- roc(hypo ~ sim_HPI_w_map_only, data = d_train)

{
  plot(roc_map_model, col = "darkblue")
  plot(roc_map_train, col = "darkred", lty = 2, add = TRUE)
  legend(
    x = 0.8,
    y = 0.3,
    legend = c("Logistic regression with only MAP",
               "MAP directly"),
    lty = c(1, 2),
    col = c("darkblue", "darkred")
  )
}
```



Create figure 3

```
library(ggrepel)
library(patchwork)

# Set plot theme
# Plot theme
# Theme for plot
theme_set(
  theme_minimal(base_size = 11) +
  theme(
    plot.title = ggtext::element_textbox_simple(
```

```

        size = 10,
        vjust = 0,
        minheight = unit(3, "lines"), width = unit(2, "inches")
    ),
    axis.title = element_text(size = 9.5),
    axis.text = ggplot2::element_text(size = ggplot2::rel(0.8)),
    panel.grid.major = ggplot2::element_line(
        color = "#454545",
        size = 0.3,
        linetype = "dotted"
    ),
    panel.grid.minor = ggplot2::element_blank()
))

plot_colors <- c("#2A6EBB", "#bb612a")

# First we extract all combinations of sensitivity, specificity and thresholds for
# all ROC analyses
roc_data_full <- as.data.frame(roc_analysis_full[c("sensitivities",
                                                  "specificities",
                                                  "thresholds")]) |>
  arrange(desc(specificities), sensitivities)

roc_data_grey <- as.data.frame(roc_analysis_grey[c("sensitivities",
                                                  "specificities",
                                                  "thresholds")]) |>
  arrange(desc(specificities), sensitivities)

roc_data_biased <- as.data.frame(roc_analysis_biased[c("sensitivities",
                                                      "specificities",
                                                      "thresholds")]) |>
  arrange(desc(specificities), sensitivities)

# We extract the rows with thresholds closest to 70, 75 and 80
# This is used for labeling the ROC curves.
roc_label_filter <- function(roc_data) {
  roc_data |>
    filter(
      row_number() == which.min(abs(thresholds - 70)) |
      row_number() == which.min(abs(thresholds - 75)) |
      row_number() == which.min(abs(thresholds - 80))
    )
}

roc_labels_full <- roc_label_filter(roc_data_full)
roc_labels_grey <- roc_label_filter(roc_data_grey)
roc_labels_biased <- roc_label_filter(roc_data_biased)

```

```

# Make the 3 ROC panels
roc_plot_full <- ggplot(roc_data_full, aes(1-specificities, sensitivities)) +
  geom_step() +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "grey") +
  geom_label_repel(aes(label = glue::glue("MAP[now] < {round(thresholds, 0)}")),
    parse = TRUE,
    size = 3.1,
    box.padding = 0.1,
    point.size = 0.1,
    segment.color = "#555555",
    position = position_nudge_repel(y = -0.2, x=0.2),
    ylim = c(0.15, 1),
    direction = "y",
    label.size = NA,
    #vjust = 1,
    hjust = 0,
    family = "Helvetica",
    data = roc_labels_full
  ) +
  annotate("label",
    x = 1, y = 0,
    label = sprintf("AUC = %.2f", auc(roc_analysis_full)),
    hjust = 1, vjust = -0,
    label.size = NA,
    family = "Helvetica",
    #label.size = NA,
    size = 3.5) +
  labs(x = "1 - specificity", y = "Sensitivity") +
  coord_equal()

roc_plot_grey <- ggplot(roc_data_grey, aes(1-specificities, sensitivities)) +
  geom_step() +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "grey") +
  geom_label_repel(aes(label = glue::glue("MAP[now] < {round(thresholds, 0)}")),
    parse = TRUE,
    size = 3.1,
    box.padding = 0.1,
    point.size = 0.1,
    segment.color = "#555555",
    direction = "y",
    position = position_nudge_repel(y = -0.2, x=0.2),
    ylim = c(0.15, 1),
    label.size = NA,
    #vjust = 1,
    hjust = 0,
    family = "Helvetica",
    data = roc_labels_grey
  )

```

```

) +
  annotate("label",
    x = 1, y = 0,
    label = sprintf("AUC = %.2f", auc(roc_analysis_grey)),
    hjust = 1, vjust = -0,
    label.size = NA,
    family = "Helvetica",
    #label.size = NA,
    size = 3.5) +
  labs(x = "1 - specificity", y = "Sensitivity") +
  coord_equal()

roc_plot_biased <- ggplot(roc_data_biased, aes(1-specificities, sensitivities)) +
  geom_step() +
  geom_abline(slope = 1, intercept = 0, linetype = 2, color = "grey") +
  geom_label_repel(aes(label = glue::glue("MAP[now] < {round(thresholds, 0)}")),
    parse = TRUE,
    size = 3.1,
    box.padding = 0.1,
    point.size = 0.3,
    segment.color = "#555555",
    position = position_nudge_repel(y = -0.1, x=0.2),
    ylim = c(0.15, 1),
    direction = "y",
    label.size = NA,
    #vjust = 1,
    hjust = 0,
    family = "Helvetica",
    data = roc_labels_biased
  ) +
  annotate("label",
    x = 1, y = 0,
    label = sprintf("AUC = %.2f", auc(roc_analysis_biased)),
    hjust = 1, vjust = -0,
    label.size = NA,
    family = "Helvetica",
    size = 3.5) +
  labs(x = "1 - specificity", y = "Sensitivity") +
  coord_equal()

# Make the 3 scatter plots
color_scale <- scale_color_manual(values = plot_colors,
                                   labels = c("Non-hypotensive events", "Hypotensive events"))

labels <- d |>
  mutate(label = ifelse(hypo, "Hypotensive events", "Non-hypotensive events")) |>
  group_by(label) |>

```

```

summarise(across(starts_with("map"), .fns = mean), hypo = first(hypo))

scatter_plot_full <- ggplot(d, aes(map_predictor, map_outcome, color = hypo)) +
  geom_point(size = 0.7, show.legend = FALSE) +
  geom_hline(yintercept = 65, color = "grey") +
  scale_y_continuous(breaks = seq(55, 95, by = 10), expand = expansion(add = c(0,5))) +
  scale_x_continuous(breaks = seq(65, 95, by = 10)) +

  labs(x = "MAP<sub>now</sub> (predictor) [mmHg]", y = "MAP in 5 minutes (outcome) [mmHg]", color = "") +
  ggtitle("***A** Full analysis") +
  color_scale +
  coord_equal() +
  theme(axis.title.x = ggtext::element_markdown())

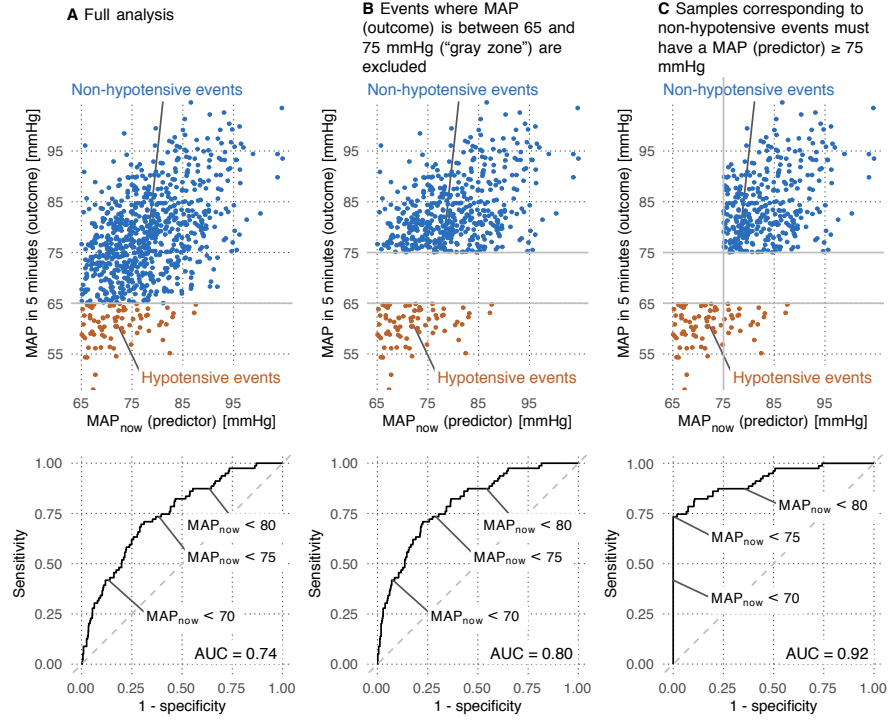
scatter_plot_full_w_labs <- scatter_plot_full
scatter_plot_full_w_labs$layers <- c(
  geom_label_repel(aes(label = label),
    size = 3.5,
    box.padding = 0,
    point.size = 0.3,
    point.padding = 0,
    min.segment.length = 0,
    segment.color = "#555555",
    position = position_nudge_repel(y = c(-100, 42), x = c(100, -100)),
    label.size = NA,
    #vjust = 1,
    force = 1000,
    hjust = 0,
    family = "Helvetica",
    data = labels,
    show.legend = FALSE
  ),
  scatter_plot_full$layers
)

scatter_plot_grey <- scatter_plot_full_w_labs %>% filter(d, !grey) +
  geom_hline(yintercept = 75, color = "grey") +
  ggtitle("***B** Events where MAP (outcome) is between 65 and 75 mmHg (‘gray zone’) are excluded")

scatter_plot_biased <- scatter_plot_grey %>% filter(d, hypo | nonhypo) +
  geom_vline(xintercept = 75, color = "grey") +
  ggtitle("***C** Samples corresponding to non-hypotensive events must have a MAP (predictor) ≥ 75 mmHg")

# Combine plots
scatter_plot_full_w_labs + scatter_plot_grey + scatter_plot_biased +
  roc_plot_full + roc_plot_grey + roc_plot_biased +
  plot_layout(ncol = 3, byrow = TRUE, guides = "collect")

```



```
ggsave("figs/figure2_simulation.pdf", device = cairo_pdf, width = 9, height = 7)
```