How to Make Forest Plot from Odds ratios

Forest Plots using ggplot2

What is a Forest Plot?

A forest plot for odds ratios is a graphical representation used in meta-analyses to show the strength of the effects of treatment across several scientific studies, or in research studies to display odds ratios across several different outcomes. It's a type of plot that allows researchers to compare the results of different studies at a glance, providing a visual summary of the findings.

Here is what it includes:

Odds Ratios (ORs): The OR is a measure of effect size, describing the strength of association or non-independence between two binary data values. It's used in this context to compare the odds of a certain event occurring in a treatment group versus a control group.

Plot Structure:

- The x-axis of a forest plot typically represents the scale of the effect (in this case, the odds ratios), often on a logarithmic scale to accommodate the wide range of values and to symmetrically display ratios below and above 1.
- Each study included in the meta-analysis is represented by a row in the plot. A square marker shows the odds ratio for that study, with the size of the square often reflecting the weight of the study in the meta-analysis (usually related to the study's sample size or precision).
- Horizontal lines through the squares represent the confidence intervals (CIs) for each study's odds ratio, indicating the precision of the estimate. Wider intervals suggest less precision, while narrower intervals indicate greater precision.

Overall Estimate: The plot also includes a diamond at the bottom, representing the pooled odds ratio from the combined studies. The width of the diamond reflects the confidence interval for this pooled estimate, offering a summary measure of the effect across all included studies.

Interpretation:

- An odds ratio of 1 indicates no difference between the treatment and control groups.
- An odds ratio greater than 1 suggests that the event is more likely to occur in the treatment group.

- An odds ratio less than 1 implies that the event is less likely to occur in the treatment group.

Heterogeneity: Forest plots also allow researchers to assess heterogeneity among the studies' results. Significant variation in the odds ratios or confidence intervals across studies might suggest variability in the effects of treatment, which could be due to differences in study populations, interventions, or methodologies.

In summary, forest plots for odds ratios provide a concise and comprehensive way to visualize the effectiveness of a treatment across multiple studies, facilitating comparison and synthesis of research findings in meta-analyses.

Forest Plot Example

For this example, we will be analyzing the odds of thromboembolic events post-COVID-19 diagnosis for patients who were prescribed daily aspirin at baseline using the dataset "m.data."

The first step I take to any data analysis or visualization task is to install the necessary packages and call on their libraries.

For our Forest Plot, we will be using the following packages:

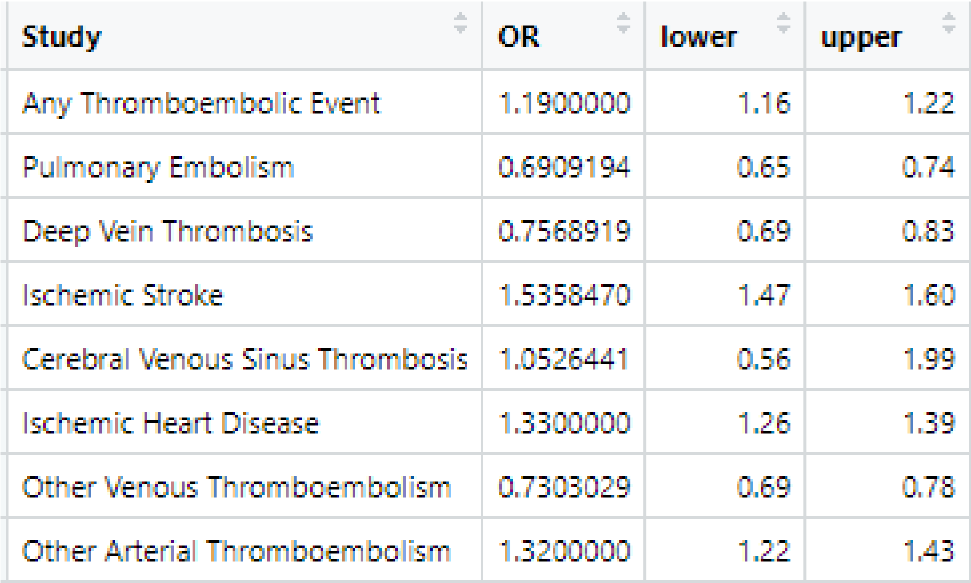
```
#Install dependencies
#install.packages("ggplot2")
#install.packages("dplyr")

library(ggplot2)

Warning: package 'ggplot2' was built under R version 4.2.3

library(dplyr)

Warning: package 'dplyr' was built under R version 4.2.3


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
```

For this first Forest Plot, we will be analyzing the uni-variate outcomes of patients who were prescribed aspirin at baseline and their odds of thrombotic events later on using the dataset "m.data."

To do this, we first need to extract all of the Odds Ratios and Confidence Intervals for all of our outcomes of interest.

```
# Define the glm models using aspirin cohort
AnyLogit <- glm(AnyThromboembolicEvent ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

PEogit <- glm(PulmonaryEmbolism ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

DVTLogit <- glm(DeepVeinThrombosis ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

IStrokeLogit <- glm(IschemicStroke ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

HStrokeLogit <- glm(HemorrhagicStroke ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

CVSTLogit <- glm(CerebralVenousSinusThrombosis ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

IHDLogit <- glm(IschemicHeartDisease ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

OVTLogit <- glm(OtherVenousThromboembolism ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

OATLogit <- glm(OtherArterialThromboembolism ~ Aspirin_Baseline, data = m.data, family = binomial(link = "logit"))

# Create a list of the glm models
models <- list(AnyLogit, PEogit, DVTLogit, IStrokeLogit, HStrokeLogit, CVSTLogit, IHDLogit, OVTLogit, OATLogit)

#Extract ORs and CIs
or<-sapply(models, function(model) exp(coef(model)[2]))

ci <- sapply(models, function(model) {coef_ci <- confint(model)
      lower <- coef_ci[2, "2.5 %"]
      upper <- coef_ci[2, "97.5 %"]
      c(lower, upper)
      })

forest_data$OR_CI <- with(forest_data, paste(round(OR, 2), " (", round(lower, 2), "-", round(upper, 2), ")", sep = ""))
```

Using this code, we want to have a new dataset, ‘forest_data’ that is formatted as follows:



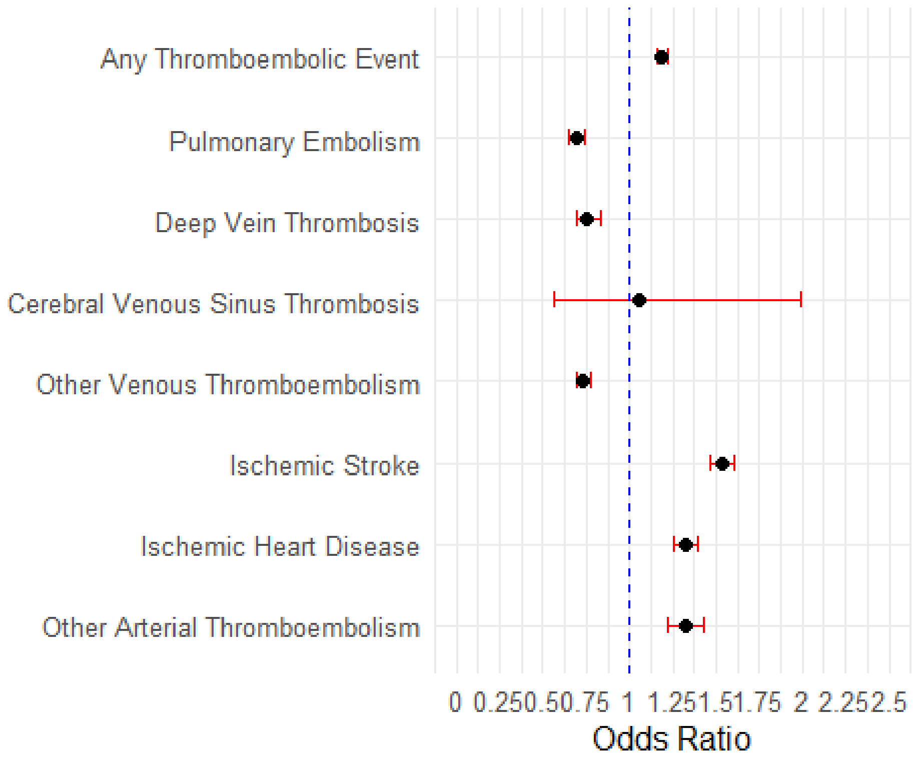
Study	OR	lower	upper
Any Thromboembolic Event	1.1900000	1.16	1.22
Pulmonary Embolism	0.6909194	0.65	0.74
Deep Vein Thrombosis	0.7568919	0.69	0.83
Ischemic Stroke	1.5358470	1.47	1.60
Cerebral Venous Sinus Thrombosis	1.0526441	0.56	1.99
Ischemic Heart Disease	1.3300000	1.26	1.39
Other Venous Thromboembolism	0.7303029	0.69	0.78
Other Arterial Thromboembolism	1.3200000	1.22	1.43

The next step will be to reorder the outcomes to how we want them displayed in the plot.

```
#Reorder the levels of the forest plot
forest_data$Study<-factor(forest_data$Study, levels = c(
  "Other Arterial Thromboembolism",
  "Ischemic Heart Disease",
  "Ischemic Stroke",
  "Other Venous Thromboembolism",
  "Cerebral Venous Sinus Thrombosis",
  "Deep Vein Thrombosis",
  "Pulmonary Embolism",
  "Any Thromboembolic Event"))
```

Now we can plot our forest plot using the ‘ggplot2’ library:

```
library(ggplot2)

#Create forest plot using the 'ggplot' package
ggplot(forest_data, aes(x = OR, y = Study)) +
      geom_errorbar(aes(xmin = lower, xmax = upper), width = 0.2, color = "red") +
      geom_point(size = 2) +
      geom_vline(xintercept = 1, linetype = "dashed", color = "blue") +
      scale_x_continuous(breaks = seq(0, 2.5, 0.25), limits = c(0, 2.5), labels = seq(0, 2.5, 0.25)) +
      labs(x = "Odds Ratio", y = "") +
      theme_minimal()+
      theme(text = element_text(size = 12))
```



Lattice Forest Plots

A lattice forest plot, while similar in purpose to a traditional forest plot for visualizing results across multiple studies or analyses, incorporates an additional layer of complexity or stratification into the visualization. This added dimension allows for the comparison of odds ratios (or other effect sizes) across different subgroups or categories within the studies, enhancing the depth of analysis. Here's how a lattice forest plot differs from a traditional forest plot:

- **Stratification/Subgroup Analysis**: The key feature of a lattice forest plot is its ability to display results stratified by one or more variables or subgroups within the studies. This could involve categorizing studies by specific characteristics such as age groups, genders, disease severity, or other relevant factors. The plot visually separates these subgroups, providing a clear comparison across different categories.

- **Visual Structure**: In a lattice forest plot, each subgroup or category may have its own mini forest plot within the larger plot structure. This means that for each category, there could be a series of squares (representing individual study estimates) and horizontal lines (confidence intervals) similar to a standard forest plot. These are arranged in a manner that visually distinguishes between the subgroups while maintaining a cohesive overall structure.

- **Interpretation and Analysis**: By incorporating subgroups into the plot, a lattice forest plot enables a more nuanced analysis of how effect sizes (e.g., odds ratios) may vary across different populations or conditions. This can be crucial for understanding the heterogeneity in effects that might not be apparent from a traditional forest plot. Analysts can identify patterns or differences in outcomes that are specific to certain subgroups, which can inform more tailored recommendations or highlight areas for further research.

- **Complexity and Clarity**: While offering richer information, lattice forest plots can also be more complex to interpret. The clarity of presentation is crucial to ensure that the additional layers of information enhance rather than obscure the insights. Proper labeling, spacing, and organization are essential to maintain readability and effectiveness in conveying the findings.

In summary, the main difference between a traditional forest plot and a lattice forest plot lies in the latter's ability to stratify and compare results across different subgroups within the included studies. This added dimension provides a more detailed view of the data, allowing for deeper insights into the variability and applicability of treatment effects across diverse populations or conditions.

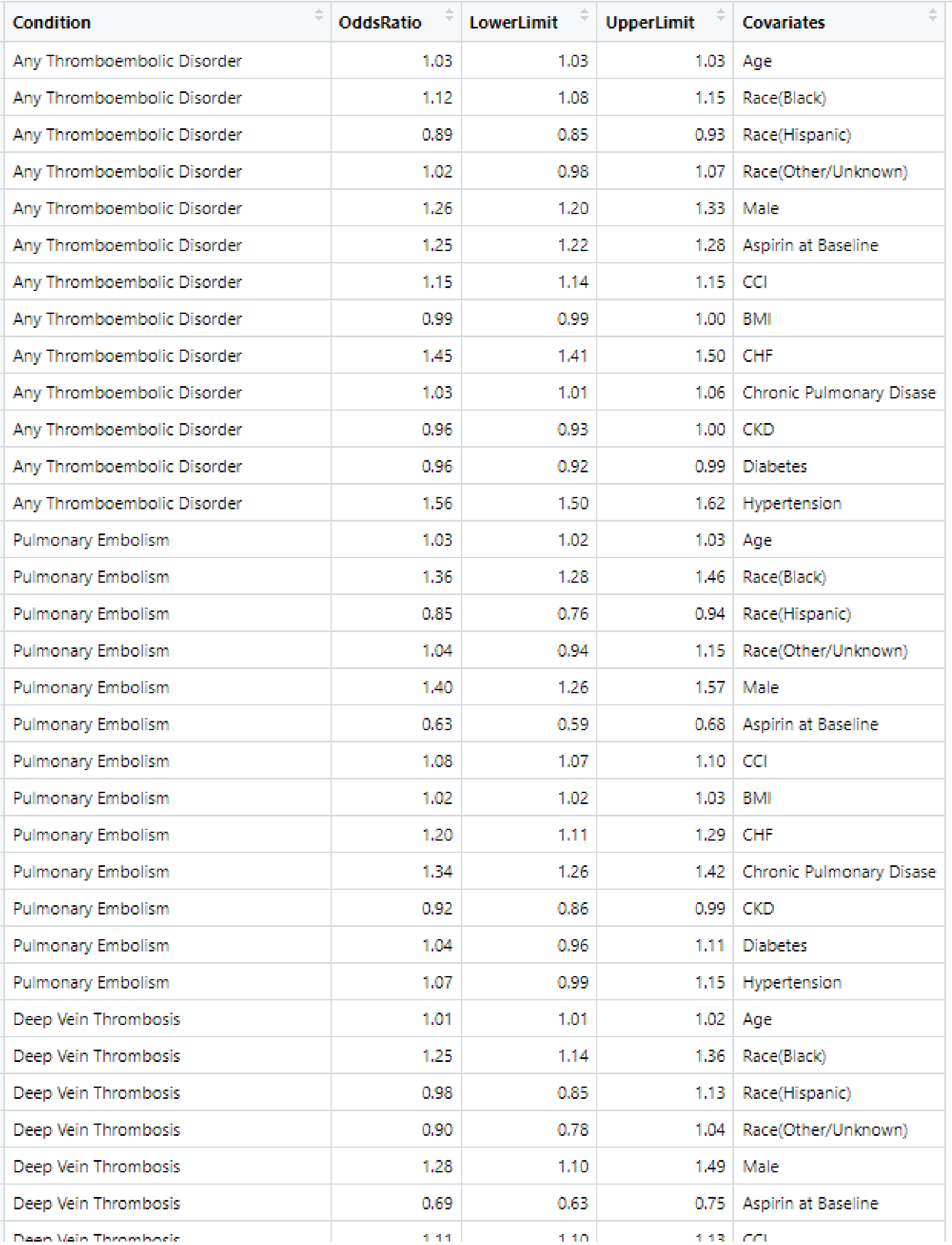
Lattice Plot Example

For lattice plots compared to the forest plot above, we are going to follow very similar steps. However, this time we are able to capture the ORs of multiple conditions using mutlivariate regression analysis and again capturing our ORs and 95% confidence intervals (as seen above).

Below is an example of one of the regression models we will use, and this would be repeated for each of the thromboembolic conditions.

```
Any Thromboembolic Event
   summary(mylogit3<- glm(AnyThromboembolicEvent~
                          Age
                          + factor(Race)
                          + Male
                          + Aspirin_Baseline
                          + CCIScore
                          + BMI
                          + CongestiveHeartFailure
                          + ChronicPulmonaryDisease
                          + RenalDisease
                          ,data = m.data
                          ,family = "binomial"(link="logit")))
      vif(mylogit3)
      glm.ratio(mylogit3, DIGITS = 2)
```

Once we have ran all of our models and captured the Condition name and the OR (95% CIs) for each of our covariates, our dataset should look like the figure below before plotting our lattice plot.



Condition	OddsRatio	LowerLimit	UpperLimit	Covariates
Any Thromboembolic Disorder	1.03	1.03	1.03	Age
Any Thromboembolic Disorder	1.12	1.08	1.15	Race(Black)
Any Thromboembolic Disorder	0.89	0.85	0.93	Race(Hispanic)
Any Thromboembolic Disorder	1.02	0.98	1.07	Race(Other/Unknown)
Any Thromboembolic Disorder	1.26	1.20	1.33	Male
Any Thromboembolic Disorder	1.25	1.22	1.28	Aspirin at Baseline
Any Thromboembolic Disorder	1.15	1.14	1.15	CCI
Any Thromboembolic Disorder	0.99	0.99	1.00	BMI
Any Thromboembolic Disorder	1.45	1.41	1.50	CHF
Any Thromboembolic Disorder	1.03	1.01	1.06	Chronic Pulmonary Disase
Any Thromboembolic Disorder	0.96	0.93	1.00	CKD
Any Thromboembolic Disorder	0.96	0.92	0.99	Diabetes
Any Thromboembolic Disorder	1.56	1.50	1.62	Hypertension
Pulmonary Embolism	1.03	1.02	1.03	Age
Pulmonary Embolism	1.36	1.28	1.46	Race(Black)
Pulmonary Embolism	0.85	0.76	0.94	Race(Hispanic)
Pulmonary Embolism	1.04	0.94	1.15	Race(Other/Unknown)
Pulmonary Embolism	1.40	1.26	1.57	Male
Pulmonary Embolism	0.63	0.59	0.68	Aspirin at Baseline
Pulmonary Embolism	1.08	1.07	1.10	CCI
Pulmonary Embolism	1.02	1.02	1.03	BMI
Pulmonary Embolism	1.20	1.11	1.29	CHF
Pulmonary Embolism	1.34	1.26	1.42	Chronic Pulmonary Disase
Pulmonary Embolism	0.92	0.86	0.99	CKD
Pulmonary Embolism	1.04	0.96	1.11	Diabetes
Pulmonary Embolism	1.07	0.99	1.15	Hypertension
Deep Vein Thrombosis	1.01	1.01	1.02	Age
Deep Vein Thrombosis	1.25	1.14	1.36	Race(Black)
Deep Vein Thrombosis	0.98	0.85	1.13	Race(Hispanic)
Deep Vein Thrombosis	0.90	0.78	1.04	Race(Other/Unknown)
Deep Vein Thrombosis	1.28	1.10	1.49	Male
Deep Vein Thrombosis	0.69	0.63	0.75	Aspirin at Baseline
Deep Vein Thrombosis	1.11	1.10	1.13	CCI

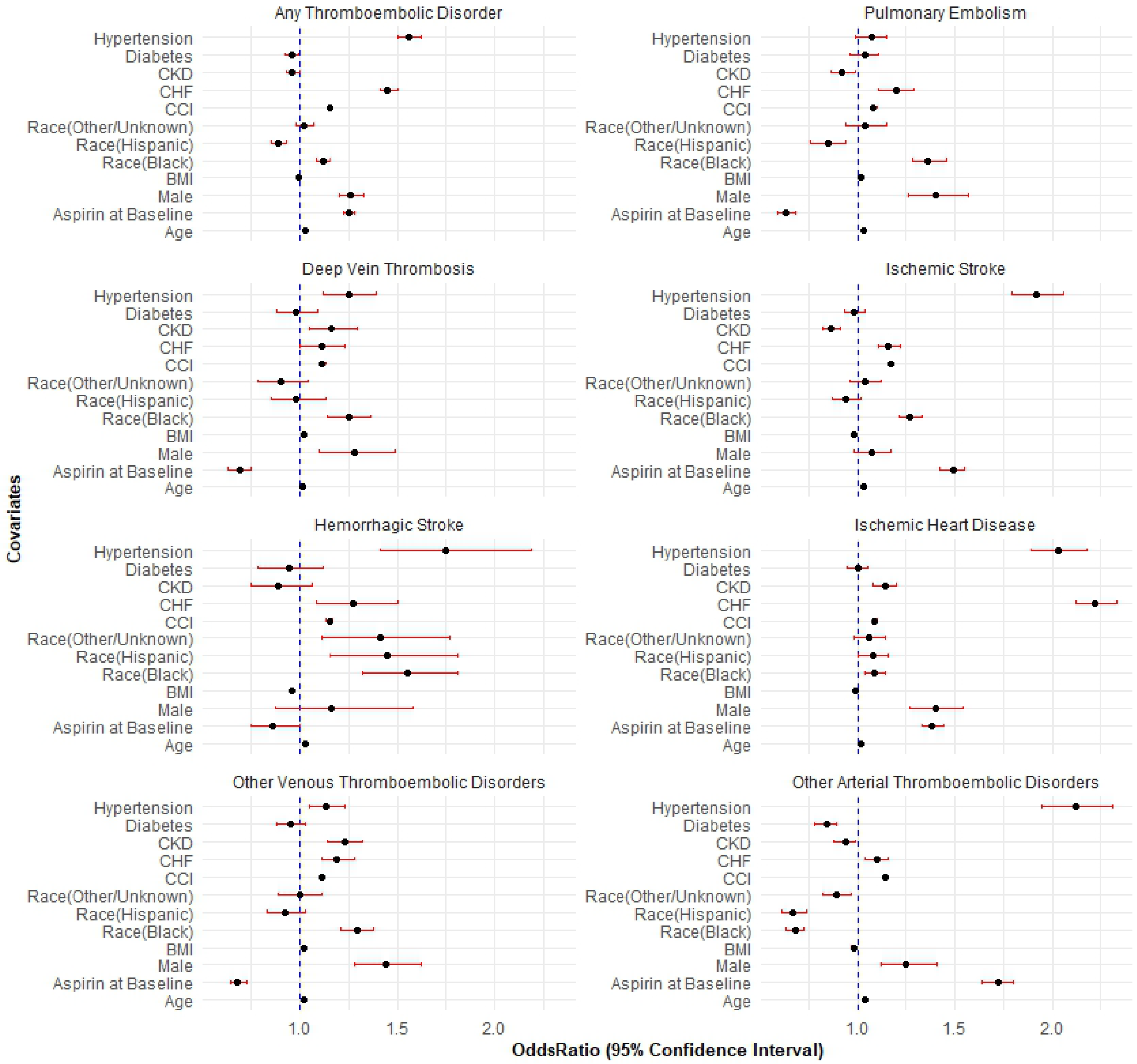
```
#Reorder covariates
OR_data$Covariates <- factor(OR_data$Covariates,levels = c("Age", "Aspirin at
Baseline", "Male", "BMI","Race(Black)", "Race(Hispanic)","Race(Other/Unknown)
","CCI", "CHF", "Chronic Pulmonary Disease", "CKD", "Diabetes", "Hypertension
"))
```

```
#Reorder primary conditions
OR_data$Condition <- factor(OR_data$Condition,levels = c("Any Thromboembolic
Disorder", "Pulmonary Embolism", "Deep Vein Thrombosis","Ischemic Stroke","Ce
rebral Venous Sinus Thrombosis", "Ischemic Heart Disease","Other Venous Throm
boembolic Disorders", "Other Arterial Thromboembolic Disorders"))
```

And again, we can re-order our conditions AND our covariates so that they are the same across all conditions.

Now we can plot our lattice plot using the ‘ggplot2’ library:

```
p2 = ggplot(data=na.omit(OR_data),
                 aes(x = Covariates,
                     y = OddsRatio, ymin = LowerLimit, ymax = UpperLimit ))+
      geom_hline(yintercept =1, linetype="dashed", color = "blue")+
      xlab('Covariates')+ ylab("OddsRatio (95% Confidence Interval)")+
      geom_errorbar(aes(ymin=LowerLimit, ymax=UpperLimit),width=0.2,color = "
red")+       geom_point(size=1.25)+
      facet_wrap(~Condition,
                   strip.position="top",
                   nrow=4,
                   ncol=2,
                   scales = "free_y") +
      theme_minimal() +
      theme(legend.position="none") +
      theme(plot.title=element_text(size=8,face="bold"),
              axis.text.y=element_text(size=7,face="plain"),
              axis.text.x=element_text(size=7,face="plain"),
              axis.title=element_text(size=8,face="bold")) +
        coord_flip()

p2
```



In summary, this is a working example of how to obtain forest plots and lattice plots using a healthcare study example. I look forward to hearing what you all think!

Here are some more references if this topic interests you:

- https://datascienceplus.com/lattice-like-forest-plot-using-ggplot2-in-r/
- http://zevross.com/blog/2019/04/02/easy-multi-panel-plots-in-r-using-facet_wrap-and-facet_grid-from-ggplot2/
- https://3.85.142.75/blog/2014/08/04/beautiful-plotting-in-r-a-ggplot2-cheatsheet-3/