

Displaying analytic results

Abhijit Dasgupta

BIOF 339

Comparing groups

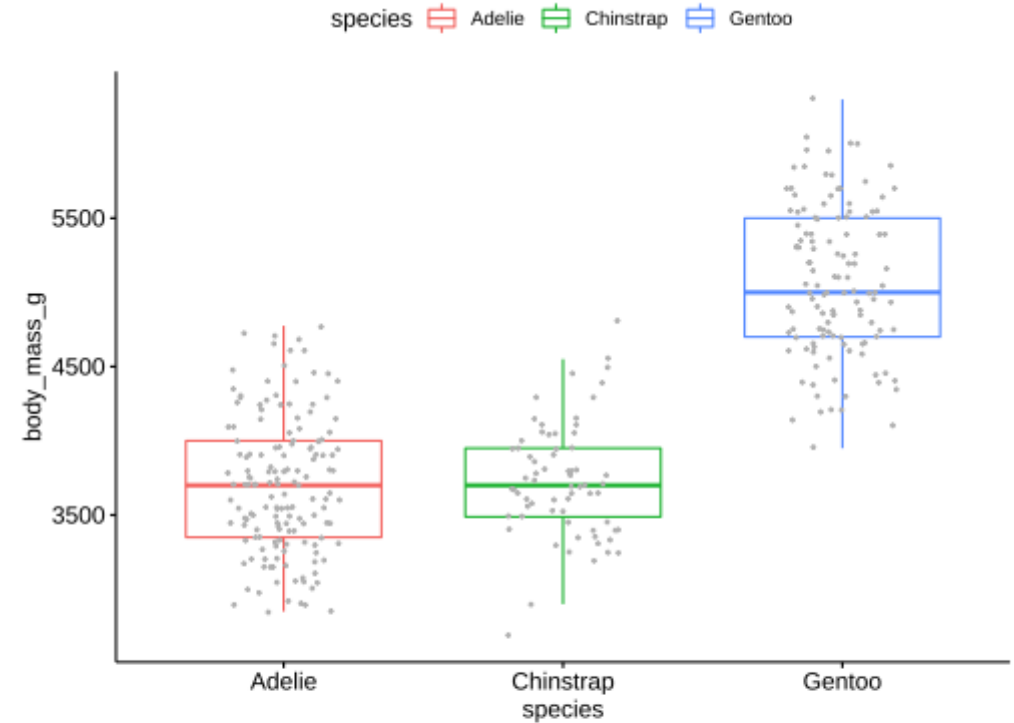
The `ggpubr` package

The `ggpubr` package, which extends `ggplot2` functionality, has several functions that allow the computation and visualization of different statistical analyses

Under the hood, it's just fancy application of R for statistical tests and then translating the results to ggplot geometries.

The ggpubr package

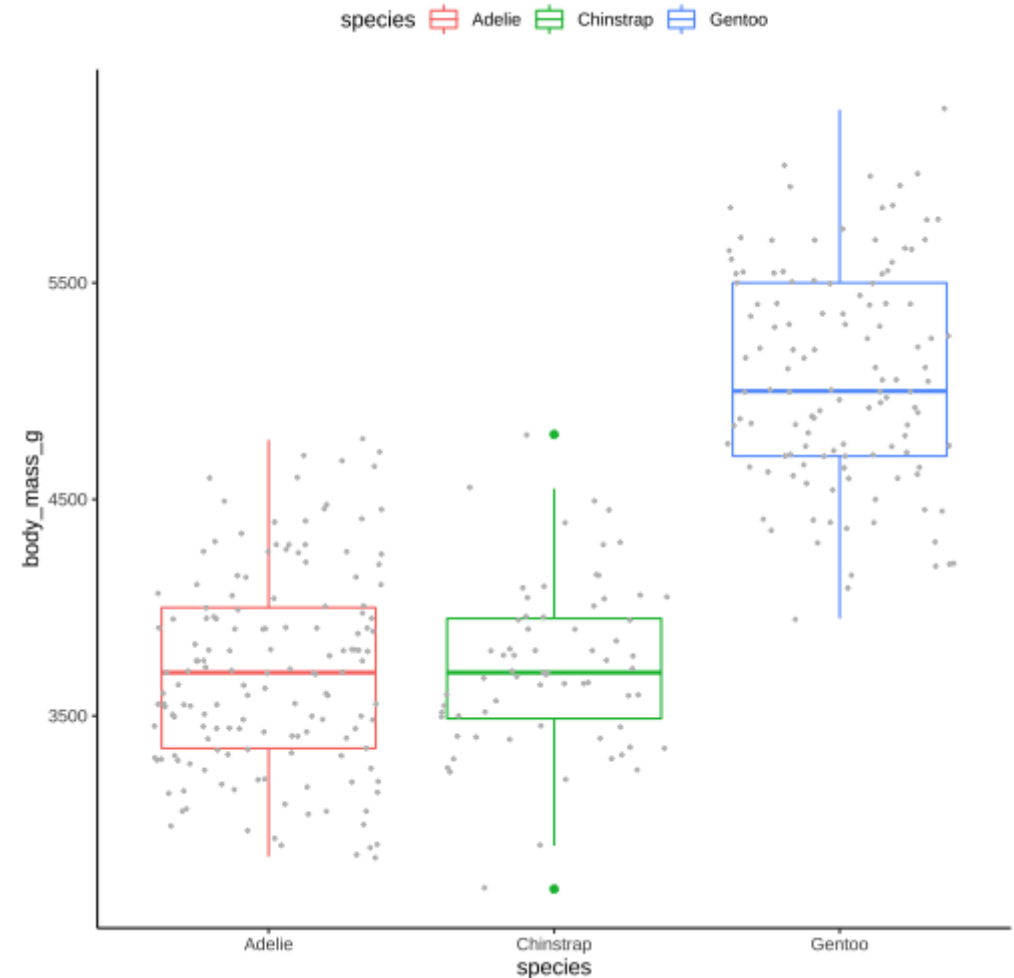
```
plt <- ggboxplot(penguins,
  x = 'species',
  y = 'body_mass_g',
  color = 'species',
  add='jitter',
  add.params = list(color='grey', size=0.5))
summary.stats <- penguins %>%
  select(body_mass_g, species) %>%
  group_by(species) %>%
  get_summary_stats(type='common')
plt2 <- ggsummarytable(
  summary.stats, x = 'species', y = c('n', 'median', 'iqr')
) +
  theme_minimal()+
  theme(panel.grid=element_blank(),
        axis.text.x = element_blank())+
  labs(x='', y='')
ggarrange(plt, plt2, ncol=1, heights=c(3,1))
```



	Adelie	Chinstrap	Gentoo
n	151	68	123
median	3700	3700	5000
iqr	650	462	800

The ggpubr package

```
(plt <- ggplot(penguins,  
             aes(x=species,  
                 y=body_mass_g,  
                 color=species))+  
  geom_boxplot()+  
  geom_jitter(color='grey', size=0.5)+  
  theme_classic()+  
  theme(legend.position = 'top'))
```



The ggpubr package

```
(plt <- ggplot(penguins,  
             aes(x=species,  
                 y=body_mass_g,  
                 color=species))+  
  geom_boxplot()+  
  geom_jitter(color='grey', size=0.5)+  
  theme_classic()+  
  theme(legend.position = 'top'))
```

```
summary.stats <- penguins %>%  
  select(body_mass_g, species) %>%  
  group_by(species) %>%  
  get_summary_stats(type='common')  
(plt2 <- ggsummarytable(  
  summary.stats, x = 'species',  
  y = c('n', 'median', 'iqr'),  
  color='species'  
  ) +  
  theme_minimal()+  
  theme(panel.grid=element_blank(),  
        axis.text.x = element_blank()+  
  labs(x='', y=''))
```

	Adelie	Chinstrap	Gentoo
n	151	68	123
median	3700	3700	5000
iqr	650	462	800

species

- a Adelie
- a Chinstrap
- a Gentoo

The ggpubr package

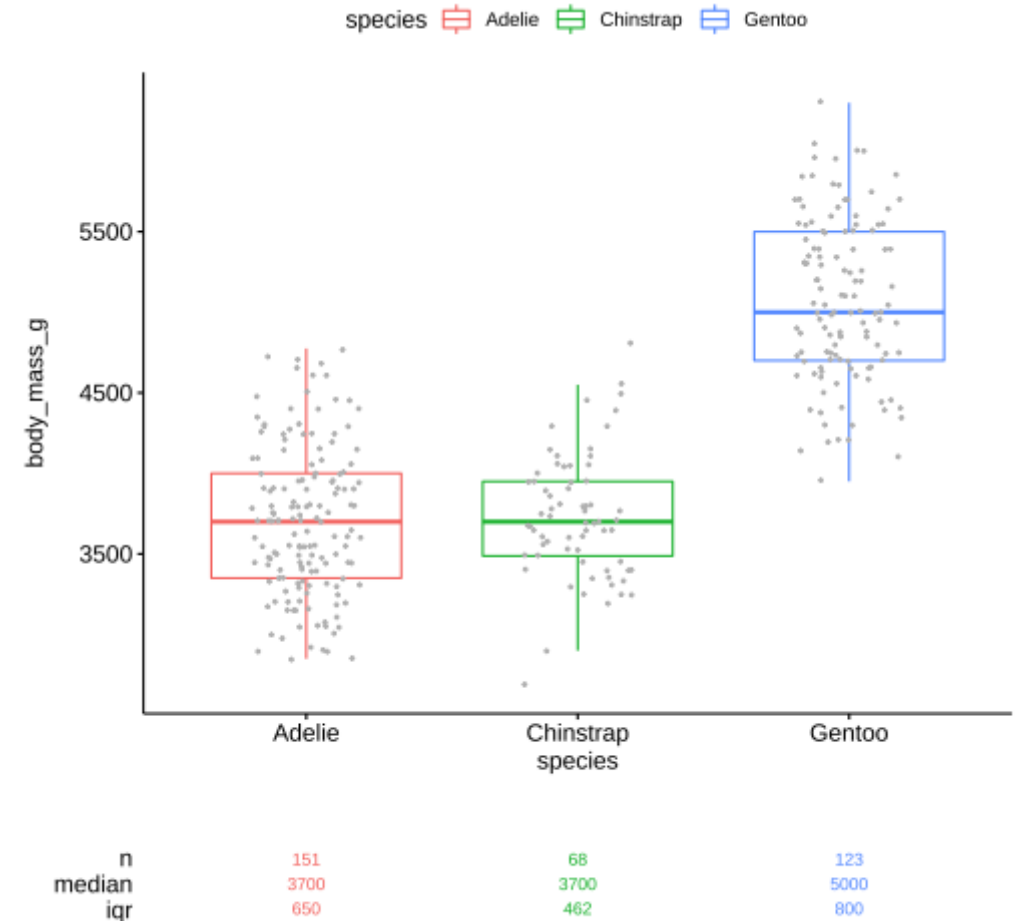
```
(plt <- ggplot(penguins,  
             aes(x=species,  
                 y=body_mass_g,  
                 color=species))+  
  geom_boxplot()+  
  geom_jitter(color='grey', size=0.5)+  
  theme_classic()+  
  theme(legend.position = 'top'))
```

```
summary.stats <- penguins %>%  
  select(body_mass_g, species) %>%  
  group_by(species) %>%  
  get_summary_stats(type='common')  
(plt2 <- ggsummarytable(  
  summary.stats, x = 'species',  
  y = c('n', 'median', 'iqr'),  
  color='species'  
  ) +  
  theme_minimal()+  
  theme(panel.grid=element_blank(),  
        axis.text.x = element_blank()+  
        labs(x='', y=''))
```

```
ggarrange(plt, plt2, ncol=1, heights = c(4,1))
```

The ggpubr package

```
(summ_plt <- ggsummarystats(  
  penguins,  
  x = 'species',  
  y = 'body_mass_g',  
  ggfunc = ggboxplot, add='jitter',  
  color='species',  
  add.params=list(color='grey', size=0.5)  
))
```



The ggpubr package

Scatter plots

```
ggscatter(penguins,  
          x = 'bill_length_mm',  
          y = 'body_mass_g',  
          color='grey',  
          add='reg.line',  
          add.params=list(color='blue'))  
labs(x = 'Bill length (mm)',  
     y = 'Body mass (g)') +  
stat_cor( label.x=30, label.y=6000,  
          label.sep='\n') +  
facet_wrap(~species)
```

The ggpubr package

Add tables to a graphic

```
library(ggsci) # Themes for science journals
dens_plt <- ggplot(penguins,
                  aes(x = body_mass_g))+
  geom_density(aes(fill=species))+
  scale_fill_jco(alpha=0.3)+ # Journal of Clinical
  theme_classic() +
  theme(axis.text.y = element_blank(),
        legend.position='top') +
  labs(y="")

stable <- desc_statby(penguins,
                    measure.var='body_mass_g',
                    grps='species')
stable <- stable[,c('species', 'length', 'mean', 'sd')]
stable_plt <- ggtexttable(stable, rows=NULL)

ggarrange(dens_plt, stable_plt,
          ncol=1, heights=c(2,1))
```

The ggpubr package

Add tables to a graphic

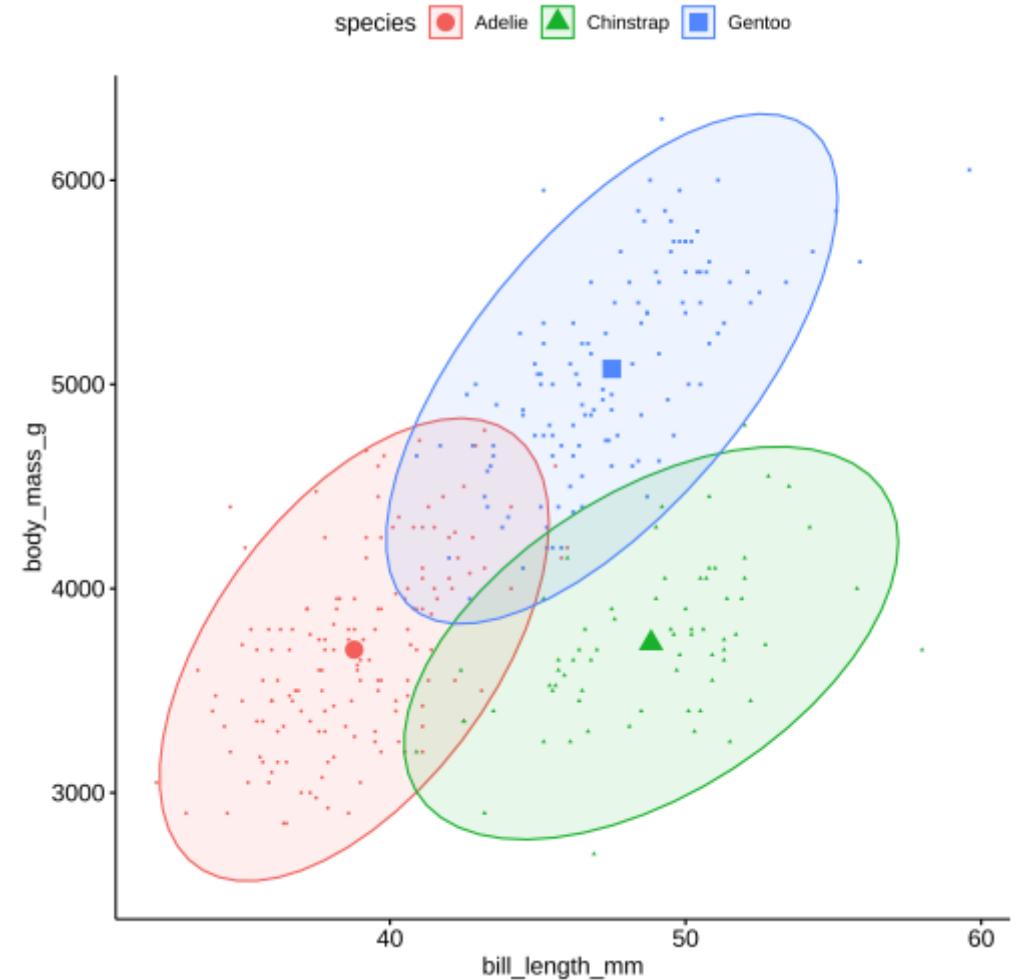
```
library(ggsci)
dens_plt <- ggplot(penguins,
                  aes(x = body_mass_g)) +
  geom_density(aes(fill=species)) +
  scale_fill_jco(alpha=0.3) +
  theme_classic() +
  theme(axis.text.y = element_blank(),
        legend.position='top') +
  stat_central_tendency(
    aes(color=species),
    type='mean',
    geom='line', linetype=2
  ) +
  scale_color_jco() +
  labs(y = "")

stable <- desc_statby(penguins,
                    measure.var='body_mass_g',
                    grps='species')
stable <- stable[,c('species', 'length', 'mean', 'sd')]
stable_plt <- ggtexttable(stable, rows=NULL)

ggarrange(dens_plt, stable_plt,
          ncol=1, heights=c(2,1))
```

The ggpubr package

```
ggscatter(penguins,  
  x='bill_length_mm',  
  y='body_mass_g',  
  color='species',  
  shape='species',  
  size=0.5,  
  ellipse=TRUE)+  
  stat_mean(aes(color=species, shape=species), size=
```



Modeling results

```
library(gtsummary)
theme_gtsummary_compact()
tbl_regression(m1,
  label = list(
    bill_length_mm ~ 'Bill length (mm)',
    bill_depth_mm ~ 'Bill depth (mm)',
    flipper_length_mm ~ 'Flipper length (mm)',
    species ~ 'Species',
    island ~ 'Island',
    sex ~ 'Sex',
    year ~ 'Year')
) %>%
  add_global_p() %>%
  bold_p(t = 0.05) %>%
  bold_labels() %>%
  italicize_levels()
```

preserve3f46698eb29c82dc

Modeling results

Putting together multiple models

```
(gt_r1 <- glm(response ~ trt + grade, trial,  
             family = binomial) %>%  
  tbl_regression(exponentiate = TRUE))
```

preservece78e1b69e63de5b

Modeling results

Putting together multiple models

```
(gt_r2 <- coxph(Surv(ttdeath, death) ~ trt + grade,  
               trial) %>%  
  tbl_regression(exponentiate = TRUE))
```

preservea8e8d46170a799d7

Modeling results

Putting together multiple models

```
(gt_t1 <- trial[c("trt", "grade")] %>%  
  tbl_summary(missing = "no") %>%  
  add_n() %>%  
  modify_header(stat_0 ~ "**n (%)**") %>%  
  modify_footnote(stat_0 ~ NA_character_))
```

preserve3e4bd5fb1314577c

Modeling results

Putting together multiple models

```
tbl_merge(  
  list(gt_t1, gt_r1, gt_r2),  
  tab_spanner = c(NA_character_, "**Tumor Response**", "**Time to Death**")  
)
```

preservebab918c1ae4dd6d

Modeling results

The finalfit package

```
explanatory <- setdiff(names(penguins), 'body_mass_g')
dependent <- 'body_mass_g'
t2 <- penguins %>%
  finalfit::finalfit(dependent, explanatory, metrics=TRUE)
knitr::kable(t2[[1]], row.names = F)
```

Dependent: body_mass_g		unit	value	Coefficient (univariable)	Coefficient (multivariable)
species	Adelie	Mean (sd)	3700.7 (458.6)	-	-
	Chinstrap	Mean (sd)	3733.1 (384.3)	32.43 (-100.37 to 165.22, p=0.631)	-282.54 (-457.22 to -107.86, p=0.002)
	Gentoo	Mean (sd)	5076.0 (504.1)	1375.35 (1264.91 to 1485.80, p<0.001)	890.96 (606.55 to 1175.36, p<0.001)
island	Biscoe	Mean (sd)	4716.0 (782.9)	-	-
	Dream	Mean (sd)	3712.9 (416.6)	-1003.11 (-1149.16 to -857.07, p<0.001)	-21.18 (-136.05 to 93.69, p=0.717)

Modeling results

Other packages

- [sjPlot](#) ([link](#))
- [stargazer](#)

Utility packages

The function `broom::tidy` takes results from [many](#) models and transforms them into *tidy datasets*, which can be formatted and output using a variety of other packages in the R ecosystem. Some popular and effective packages are

- [gt](#): The grammar of tables ([link](#))
- [flextable](#): Creating tables that can work with RMarkdown/HTML as well as the [officeverse](#), which creates Microsoft Word and Powerpoint files from R and RMarkdown ([link](#))
- [pixiedust](#) is designed to format the results from the `broom::tidy` ([link](#))
- [xtable](#) and [stargazer](#) are targeted towards \LaTeX and PDF outputs

This slide differs from the video. It was updated in 2021

Model results

Plotting results

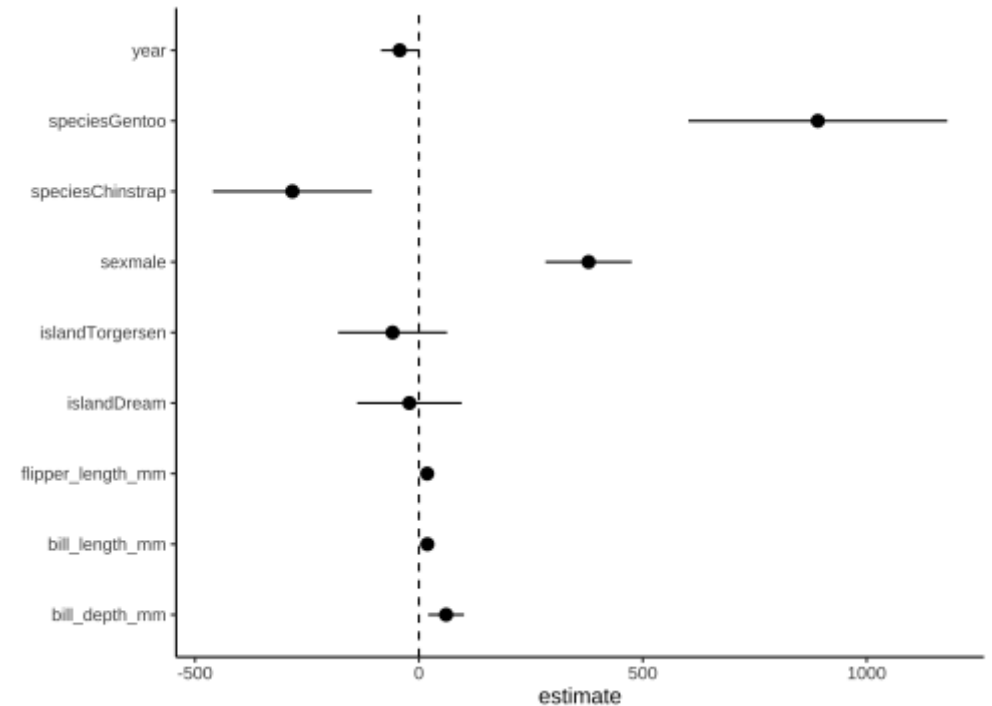
```
out <- broom::tidy(m1) %>%  
  slice(-1) %>%  
  mutate(lcb = estimate - 2*std.error,  
         ucb = estimate + 2*std.error) %>%  
  select(term, estimate, lcb, ucb)  
knitr::kable(out, digits = 2) %>%  
  kable_styling()
```

term	estimate	lcb	ucb
speciesChinstrap	-282.54	-460.12	-104.96
speciesGentoo	890.96	601.83	1180.08
islandDream	-21.18	-137.96	95.60
islandTorgersen	-58.78	-180.48	62.93
bill_length_mm	18.96	4.74	33.19
bill_depth_mm	60.80	20.79	100.80
flipper_length_mm	18.50	12.25	24.76
sexmale	378.98	282.83	475.13
year	-42.78	-84.68	-0.89

Model results

Plotting results

```
(plt1 <- out %>%  
  ggplot(aes(x = term, y = estimate,  
             ymin = lcb, ymax = ucb))+  
  geom_pointrange()+  
  geom_hline(yintercept=0, linetype=2)+  
  xlab('') + geom_vline(xintercept=0, linetype=2)+  
  coord_flip() + theme_classic())
```

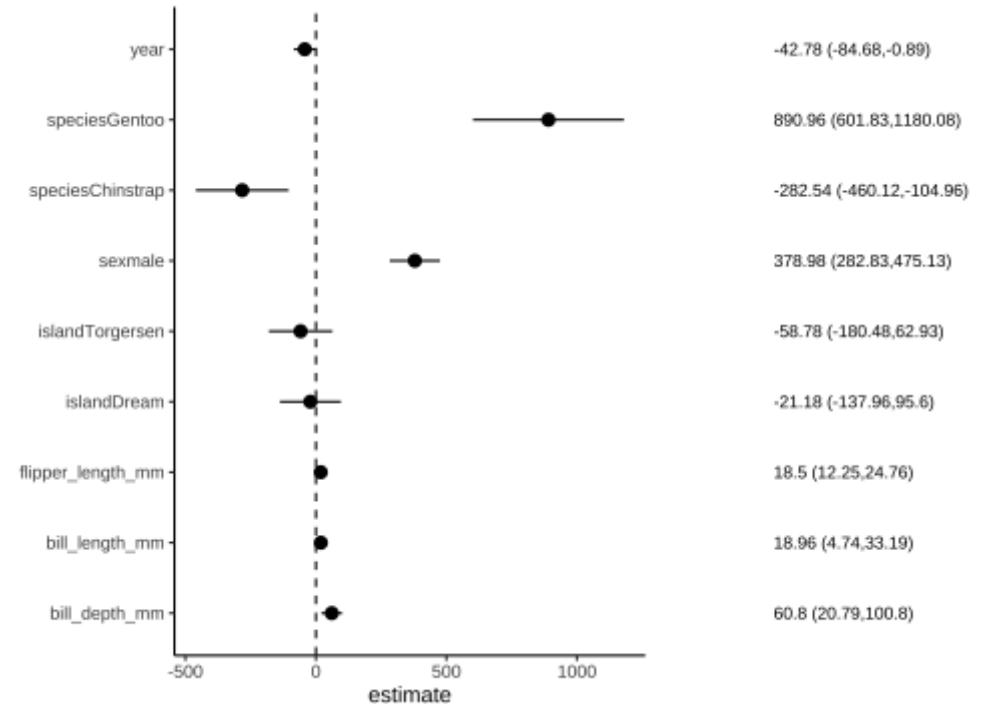


Model results

Plotting results

```
(plt1 <- out %>%  
  ggplot(aes(x = term, y = estimate,  
            ymin = lcb, ymax = ucb))+  
  geom_pointrange()+  
  geom_hline(yintercept=0, linetype=2)+  
  xlab('') + geom_vline(xintercept=0, linetype=2)+  
  coord_flip() + theme_classic())
```

```
out <- out %>%  
  mutate(across(-term, round, 2)) %>%  
  mutate(ci = glue::glue('{estimate} ({lcb},{ucb})')  
  plt2 <- ggplot(out, aes(x = term, y = 0))+  
  geom_text(aes(label=ci), size=3, hjust=0)+  
  coord_flip()+theme_void() + scale_y_continuous(lin  
  ggpubr::ggarrange(plt1, plt2, nrow=1, widths=c(2,1),
```

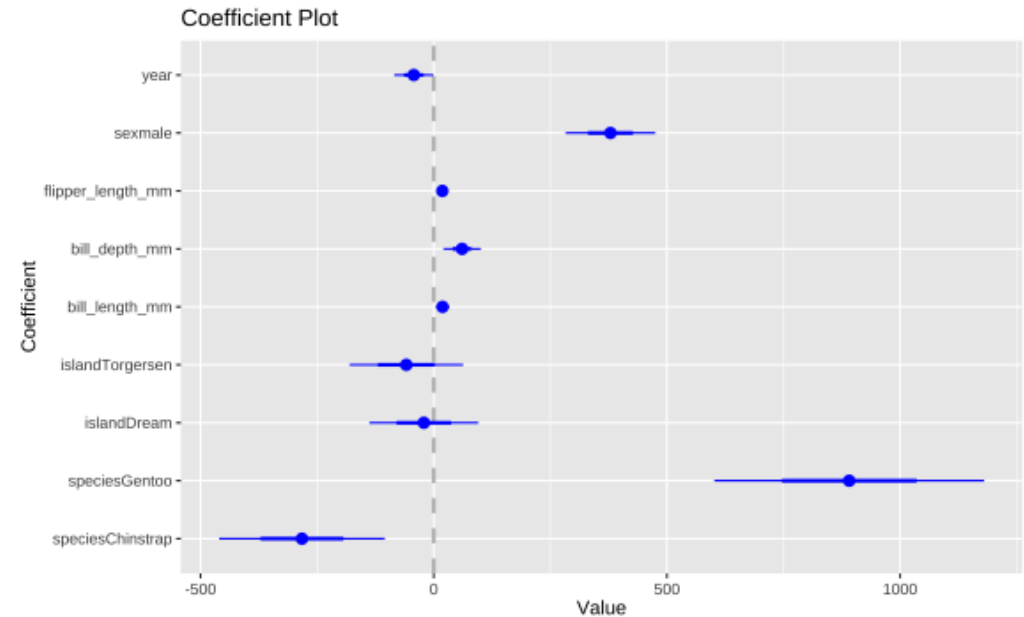


Model results

Plotting results

We can use the `coefplot` package to make a similar plot

```
pacman::p_load("coefplot")  
coefplot(m1, intercept=FALSE)
```



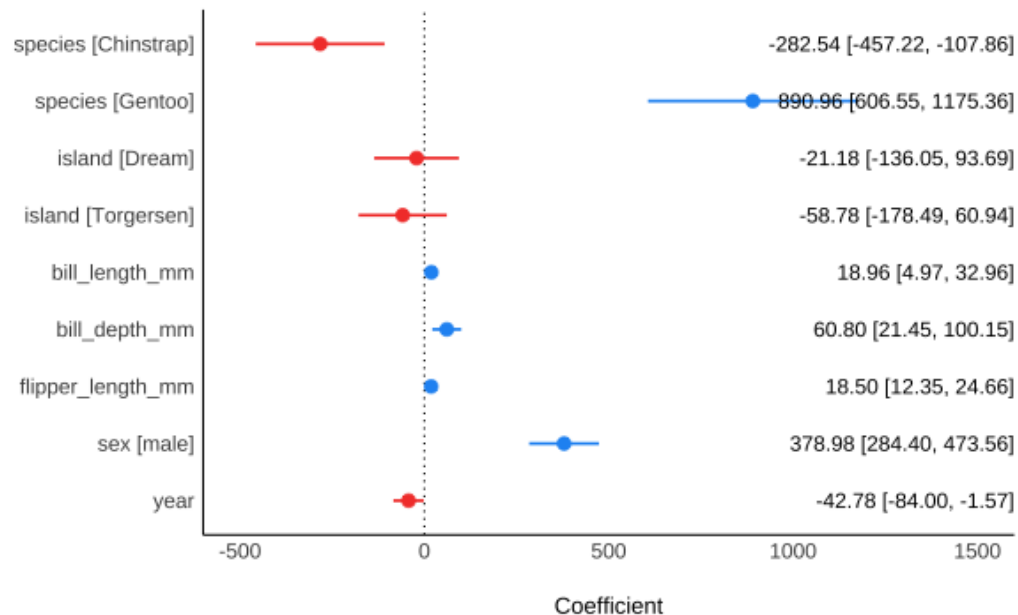
This slide differs from the video. It was updated in 2021

Model results

Plotting results

We can also use the `see` package

```
pacman::p_load(see, parameters)  
plot(model_parameters(m1), show_labels = TRUE)
```



This slide differs from the video. It was updated in 2021

Model results

Survival analysis

```
library(survival)
library(survminer)
pbc <- pbc %>% mutate(trt = as.factor(trt), stage =
m <- survfit(Surv(time, status==2)~trt, data=pbc)
ggsurvplot(m, pval=TRUE, risk.table=T)
```

Model results

Survival analysis

```
library(survival)
library(survminer)
pbc <- pbc %>% mutate(trt = as.factor(trt), stage =
m <- survfit(Surv(time, status==2)~trt, data=pbc)
ggsurvplot(m, pval=TRUE, risk.table=T)
```

```
m2 <- coxph(Surv(time, status==2) ~ trt, data = pbc)
gtsummary::tbl_regression(m2, exponentiate=T)
```

preservefa538ab9eac99275

Model results

Survival analysis

```
m3 <- coxph(Surv(time, status==2) ~ trt + sex +
            age + stage, data=abc)
out <- broom::tidy(m3) %>%
  mutate(lcb = estimate - 2*std.error,
         ucb = estimate + 2*std.error) %>%
  mutate(across(c(estimate, lcb, ucb), exp))
gtsummary::tbl_regression(m3, exponentiate = T)
```

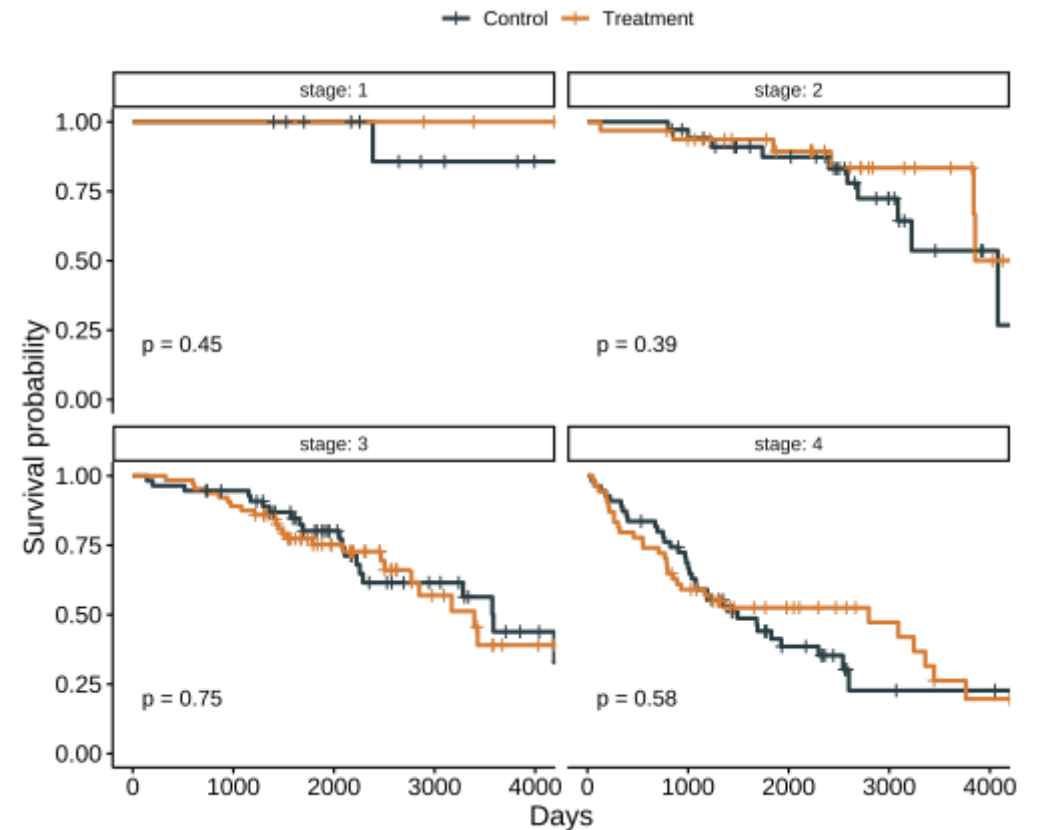
preserve5d634018ef9b25bf

```
out %>%
  ggplot(aes(x = term, y = estimate,
            ymin = lcb, ymax=ucb))+
  geom_pointrange()+
  geom_hline(yintercept=1, linetype=2)+
  coord_flip()+
  theme_classic()
```

Model results

Survival analysis

```
ggsurvplot(  
  m, pval = TRUE, risk.table = FALSE,  
  palette="jama",  
  xlab = "Days",  
  legend.title="",  
  facet.by = "stage",  
  legend.labs=c("Control", "Treatment")  
)
```



This slide differs from the video and was updated in 2021