

# AnalysisNote9R

February 24, 2023

As we generated dataset for biome plot with Python, we plot them on R

```
[1]: install.packages("remotes")
remotes::install_github("valentinitnelav/plotbiomes")
```

```
/var/folders/g4/3cvylr8j6mq822n672x35vn40000gr/T//RtmpvWHh72/downloaded_packages
```

Skipping install of 'plotbiomes' from a github remote, the SHA1 (d098e2b7) has not changed since last install.

Use `force = TRUE` to force installation

```
[18]: install.packages('sp')
```

```
/var/folders/g4/3cvylr8j6mq822n672x35vn40000gr/T//Rtmp4tVbIN/downloaded_packages
```

```
[2]: #install.packages('maptools')
library(plotbiomes)
library(ggplot2)
library(raster)
library(maptools)
library(sp)
```

We can use the bilow dataset as boudaries. If we can check whether they are in the boudaries or not the TS and RA data can estimate their biome\_id. Use sp library In addition, we can check the outliers using get\_outliers function

```
[3]: data=read.csv("Biome_distributions.csv") # distributions of RA and TS
data_null=read.csv("Real_animal_hex_biome.csv") # distributions of RA _null
data$Annu_Prec=data$Annu_Prec/10 # convert mm to cm
data_null$Annu_Prec=data_null$Annu_Prec/10 # convert mm to cm
head(data)
```

		hex_index <chr>	Annu_Mean_Temp <dbl>	Annu_Prec <dbl>	Species <chr>	index <chr>
A data.frame: 6 × 5	1	815f3fffffffff	25.954997	208.1	anteater	TS
	2	81667fffffffff	27.524673	277.4	anteater	TS
	3	81807fffffffff	27.020096	243.9	anteater	TS
	4	81817fffffffff	25.155792	150.0	anteater	TS
	5	818bbfffffffff	24.461034	177.3	anteater	TS
	6	8126bfffffffff	3.876704	36.9	anteater	RA

```
[4]: head(data_null)
```

		Lat <dbl>	Lng <dbl>	hex_index <chr>	Annu_Mean_Temp <dbl>	Annu_Prec <dbl>
A data.frame: 6 × 5	1	78.30223	38.199745	81003fffffffff	-7.57941452	17.60132
	2	76.28928	73.828666	8100bfffffffff	-12.43920027	25.26250
	3	71.98744	29.055703	81013fffffffff	-0.01703325	52.35956
	4	76.25412	4.274271	81017fffffffff	-7.03445589	37.73640
	5	71.52423	51.709609	8101bfffffffff	-5.43794250	31.00000
	6	79.19902	-108.398309	81023fffffffff	-18.00474525	12.26207

```
[5]: Biome_class=function(data){
  df1=Whittaker_biomes[Whittaker_biomes$biome_id==1, ]
  pol1.x=df1$temp_c
  pol1.y=df1$prec_cm

  df2=Whittaker_biomes[Whittaker_biomes$biome_id==2, ]
  pol2.x=df2$temp_c
  pol2.y=df2$prec_cm

  df3=Whittaker_biomes[Whittaker_biomes$biome_id==3, ]
  pol3.x=df3$temp_c
  pol3.y=df3$prec_cm

  df4=Whittaker_biomes[Whittaker_biomes$biome_id==4, ]
  pol4.x=df4$temp_c
  pol4.y=df4$prec_cm

  df5=Whittaker_biomes[Whittaker_biomes$biome_id==5, ]
  pol5.x=df5$temp_c
  pol5.y=df5$prec_cm

  df6=Whittaker_biomes[Whittaker_biomes$biome_id==6, ]
  pol6.x=df6$temp_c
  pol6.y=df6$prec_cm

  df7=Whittaker_biomes[Whittaker_biomes$biome_id==7, ]
  pol7.x=df7$temp_c
  pol7.y=df7$prec_cm
}
```

```

df8=Whittaker_biomes[Whittaker_biomes$biome_id==8, ]
pol8.x=df8$temp_c
pol8.y=df8$precip_cm

df9=Whittaker_biomes[Whittaker_biomes$biome_id==9, ]
pol9.x=df9$temp_c
pol9.y=df9$precip_cm

category=c()
X=data$Annu_Mean_Temp
Y=data$Annu_Prec # we need cm, not mm
for (i in 1:nrow(data)){
  X=data$Annu_Mean_Temp[i]
  Y=data$Annu_Prec[i]
  if (point.in.polygon(X, Y, pol1.x, pol1.y)==1){
    category=c(category, 1)
  }
  else
    if (point.in.polygon(X, Y, pol2.x, pol2.y)==1){
      category=c(category, 2)
    }
  else if (point.in.polygon(X, Y, pol3.x, pol3.y)==1){
    category=c(category, 3)
  }
  else if (point.in.polygon(X, Y, pol4.x, pol4.y)==1){
    category=c(category, 4)
  }
  else if (point.in.polygon(X, Y, pol5.x, pol5.y)==1){
    category=c(category, 5)
  }
  else if (point.in.polygon(X, Y, pol6.x, pol6.y)==1){
    category=c(category, 6)
  }
  else if (point.in.polygon(X, Y, pol7.x, pol7.y)==1){
    category=c(category, 7)
  }
  else if (point.in.polygon(X, Y, pol8.x, pol8.y)==1){
    category=c(category, 8)
  }
  else if (point.in.polygon(X, Y, pol9.x, pol9.y)==1){
    category=c(category, 9)
  }
  else{
    category=c(category, 10)
  }
}

```

```

}
#print(category)
data$biome_id=as.factor(category)
return(data)
}
data=Biome_class(data)
head(data)

```

A data.frame: 6 × 6

	hex_index <chr>	Annu_Mean_Temp <dbl>	Annu_Prec <dbl>	Species <chr>	index <chr>	biome_id <fct>
1	815f3fffffffff	25.954997	208.1	anteater	TS	1
2	81667fffffffff	27.524673	277.4	anteater	TS	4
3	81807fffffffff	27.020096	243.9	anteater	TS	1
4	81817fffffffff	25.155792	150.0	anteater	TS	1
5	818bbfffffffff	24.461034	177.3	anteater	TS	1
6	8126bfffffffff	3.876704	36.9	anteater	RA	5

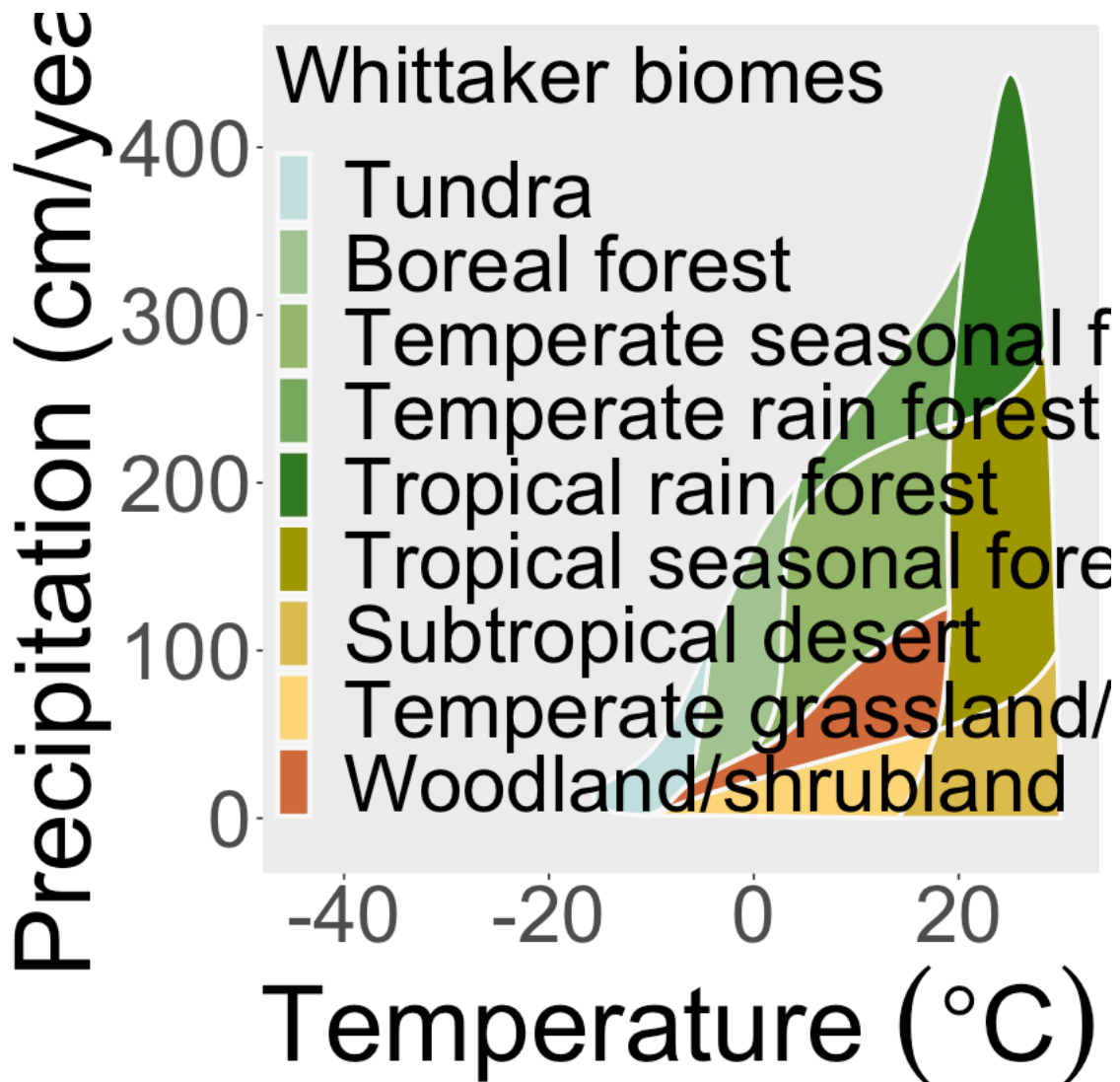
```

[45]: g=whittaker_base_plot()+theme(legend.position="none")+ylim(-10, 450)
g=g+geom_point(data=data_null, aes(Annu_Mean_Temp, Annu_Prec),
               size=4, colour='black', alpha=0.0, shape=4) # null data
g=g+labs(x='Annual mean temperature (°C)', y='Precipitation (cm/year)')
g=g+theme(
  legend.justification = c(0, 1), # pick the upper left corner of the legend
  ↪ box and
  legend.position = c(0, 1), # adjust the position of the corner as relative
  ↪ to axis
  legend.background = element_rect(fill = NA), # transparent legend background
  legend.box = "horizontal", # horizontal arrangement of multiple legends
  legend.spacing.x = unit(0.5, units = "cm"), # horizontal spacing between
  ↪ legends
  legend.text=element_text(size=36),
  legend.title=element_text(size=36),
  axis.text = element_text(size = 36),
  axis.title = element_text(size = 48),
  panel.grid = element_blank() # eliminate grids
)
#
#g=g+guides(color = guide_legend(nrow = 6))
ggsave(paste0('./Biome/Biome_example.pdf'), width = 18, height = 12)
g

```

Scale for **y** is already present.

Adding another scale for **y**, which will replace the existing scale.



```
[6]: BiomePlot=function(target){
  g=whittaker_base_plot()+theme(legend.position="none")

  g=g+geom_point(data=data_null, aes(Annu_Mean_Temp, Annu_Prec),
    size=4, colour='black', alpha=0.0, shape=4) # null data

  d=data[data$index=='RA', ]
  g=g+geom_point(data=d[d$Species==target, ],
    aes(Annu_Mean_Temp, Annu_Prec),
    shape = 1,
    stroke = 2, # acts as the thickness of the boundary line
    colour = 'black', # acts as the color of the boundary line
```

```

        size = 9)
g=g+geom_point(data=d[d$Species==target, ],
               aes(Annu_Mean_Temp, Annu_Prec),
               shape = 1,
               stroke = 1, # acts as the thickness of the boundary line
               colour = '#8da0cb', # acts as the color of the boundary line
               size = 8)
g=g+geom_point(data=data[data$Species==target, ],
               aes(Annu_Mean_Temp, Annu_Prec, color=index, shape=index,
↪size=index),
               )+ scale_color_manual(values=c('#8da0cb',
↪'#fc8d62'))+scale_size_manual(values=c(8,8))
g=g+ylim(-10, 450)
# theme(axis.title = element_text(size = 20), axis.text= element_text(size
↪= 16))
# labs(x='Annual mean temperature (°C)', y='Precipitation (cm/
↪year)')+guides(color = guide_legend(nrow = 6), legend.position=c(0.4, 1.
↪12))

d=data[data$index=='TS', ]
g=g+geom_point(data=d[d$Species==target, ],
               aes(Annu_Mean_Temp, Annu_Prec),
               shape = 2,
               stroke = 3, # acts as the thickness of the boundary line
               colour = 'gray95', # acts as the color of the boundary line
               size = 9)
g=g+geom_point(data=d[d$Species==target, ],
               aes(Annu_Mean_Temp, Annu_Prec),
               shape = 17,
               stroke = 1, # acts as the thickness of the boundary line
               colour = '#fc8d62', # acts as the color of the boundary line
               size = 8)
#g=g+theme_bw()+ggtitle(target) +
g=g+theme(
  #legend.justification = c(0, 1), # pick the upper left corner of the
↪legend box and
  #legend.position = c(0, 1), # adjust the position of the corner as
↪relative to axis
  #legend.background = element_rect(fill = NA), # transparent legend
↪background
  #legend.box = "horizontal", # horizontal arrangement of multiple legends
  #legend.spacing.x = unit(0.5, units = "cm"), # horizontal spacing
↪between legends
  #legend.text=element_text(size=36),
  #legend.title=element_text(size=36),
  axis.text = element_text(size = 36),

```

```

    axis.title = element_blank(),
    #plot.title=element_text(size=64),
    panel.grid = element_blank() # eliminate grids
  )
  ggsave(paste0('./Biome/Biome_', target, '.pdf'), width = 18, height = 12)
  return (g)
}

```

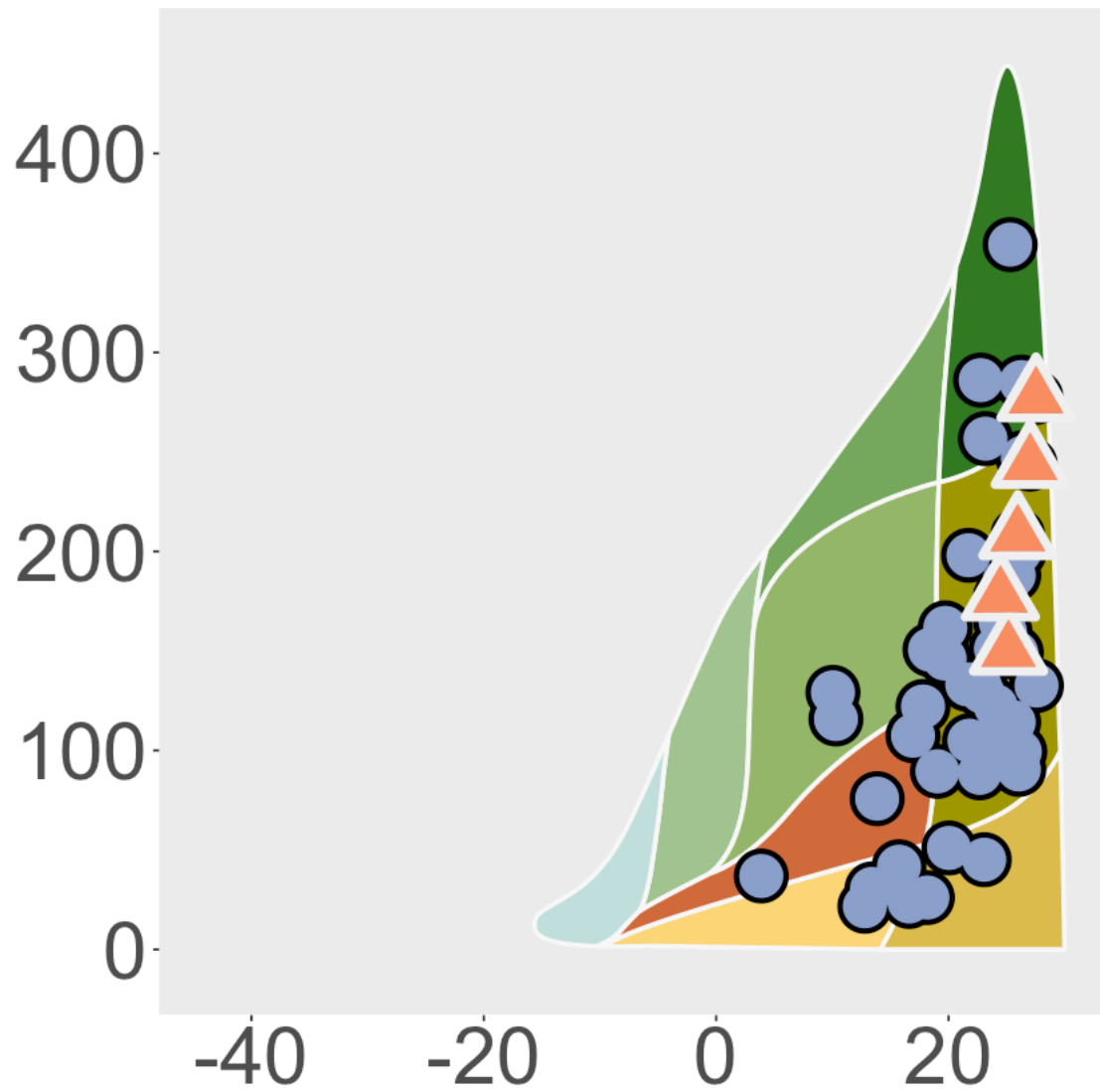
```
[7]: unique(data$Species)
```

1. 'anteater' 2. 'badger' 3. 'ground squirrel' 4. 'hawk' 5. 'mink' 6. 'mouse' 7. 'opossum' 8. 'owl'  
9. 'porcupine' 10. 'rabbit' 11. 'raccoon' 12. 'rat' 13. 'raven' 14. 'skunk' 15. 'spider' 16. 'wren'

```
[8]: s=unique(data$Species)
for (i in 1:length(s)){
  target=s[i]
  g=BiomePlot(target)
  print(g)
}

```

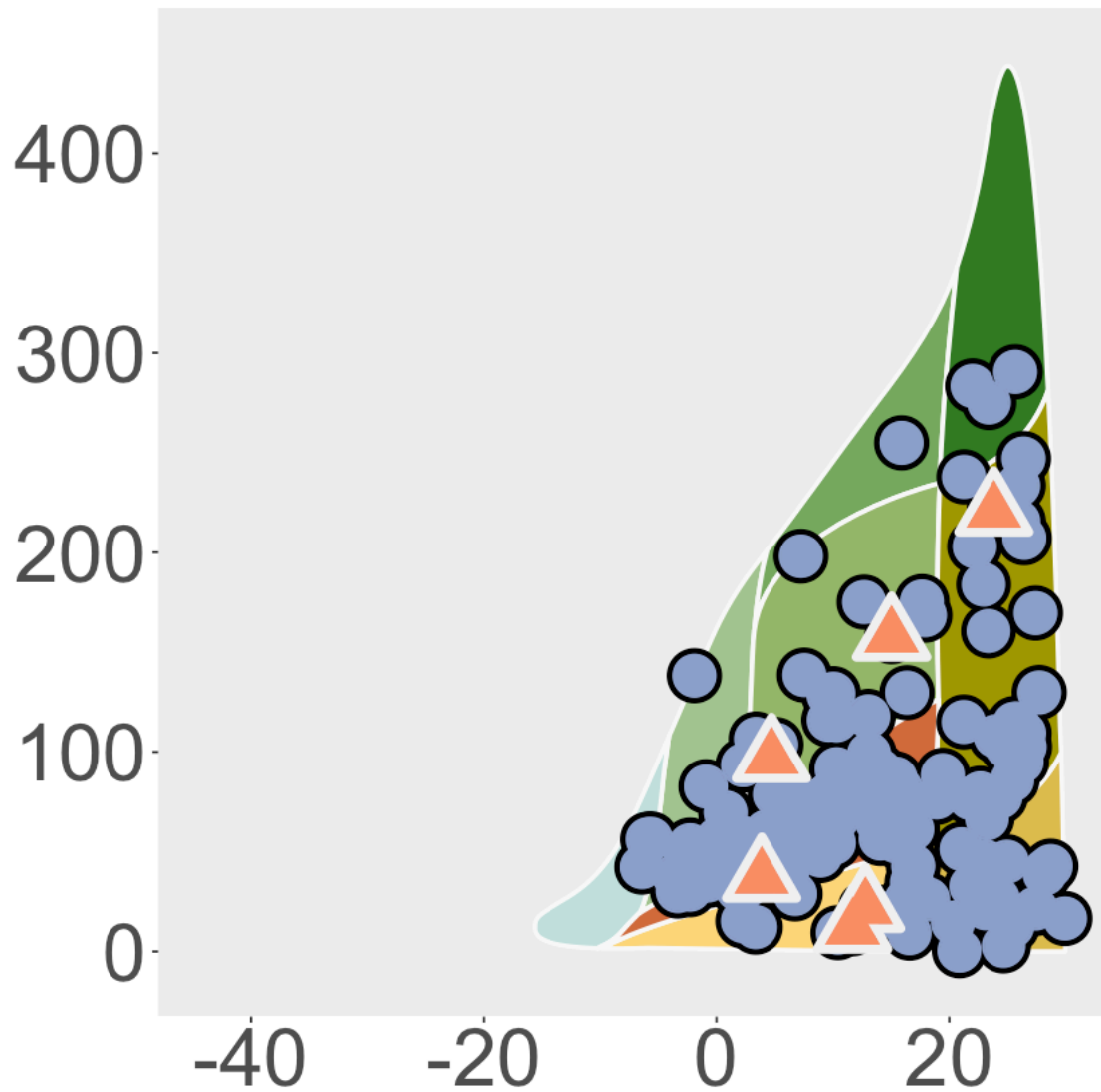
Scale for y is already present.  
Adding another scale for y, which will replace the existing scale.  
Scale for y is already present.  
Adding another scale for y, which will replace the existing scale.  
Scale for y is already present.  
Adding another scale for y, which will replace the existing scale.



Scale for `y` is already present.

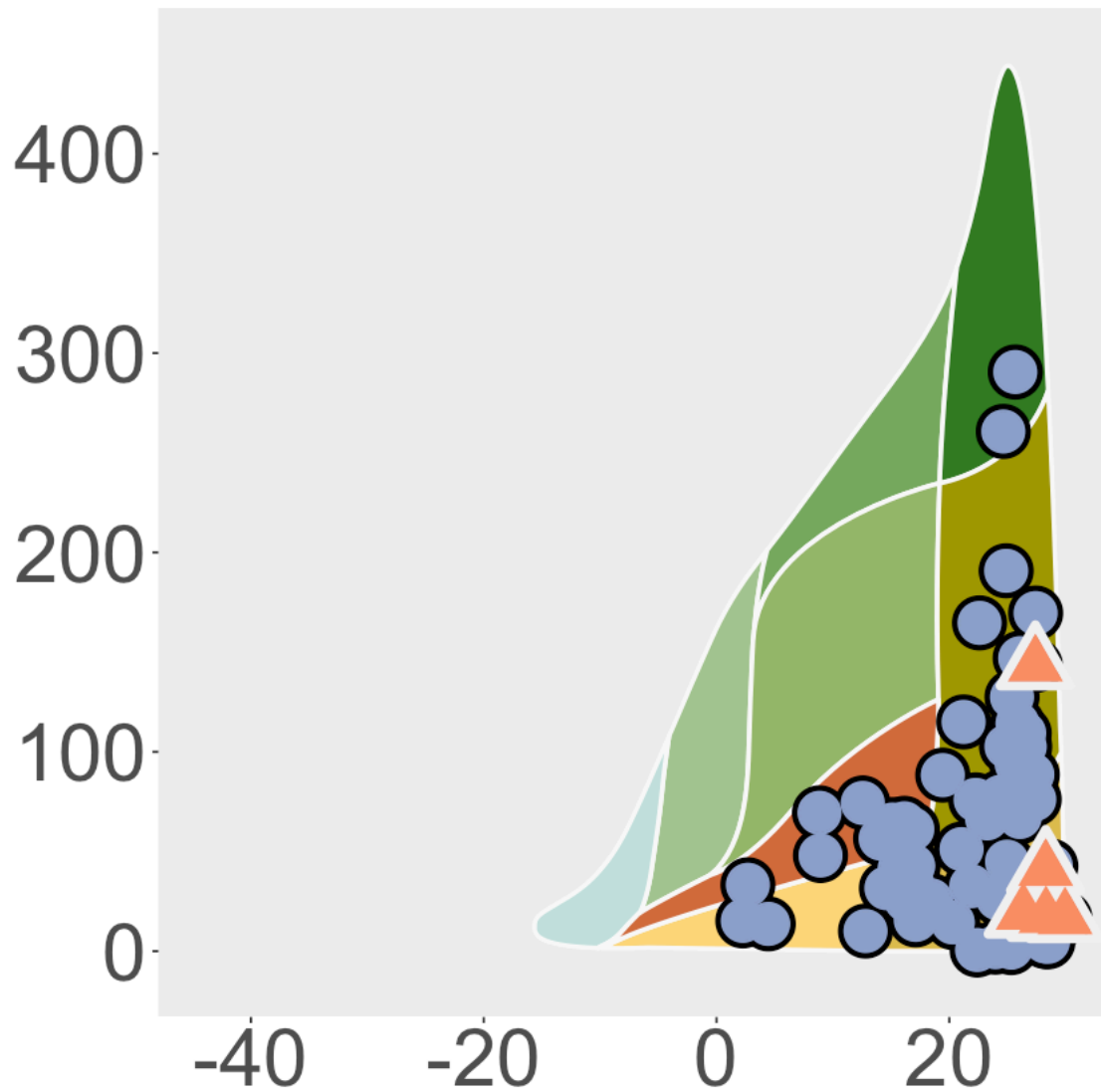
Adding another scale for `y`, which will replace the existing scale.





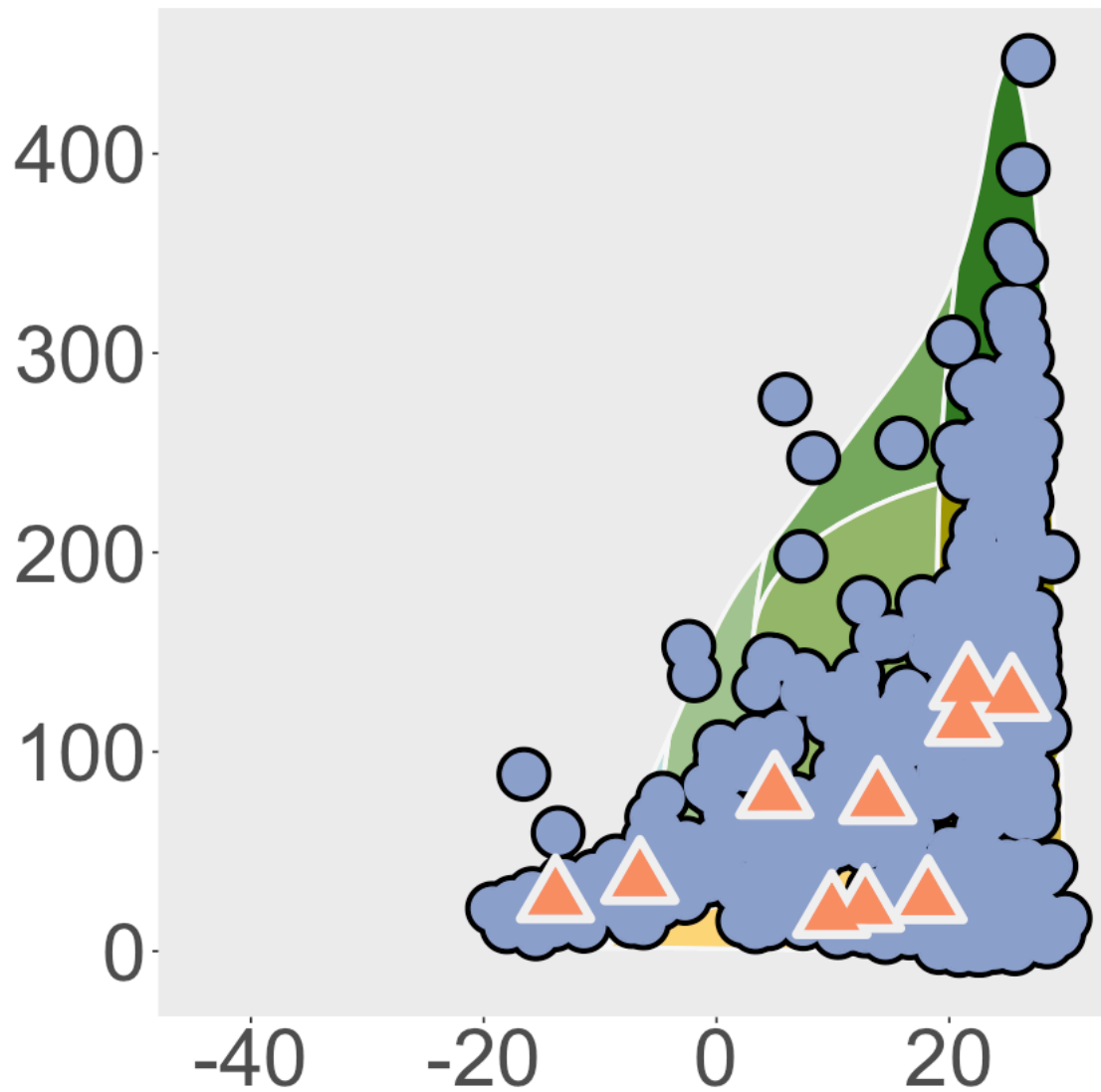
Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.



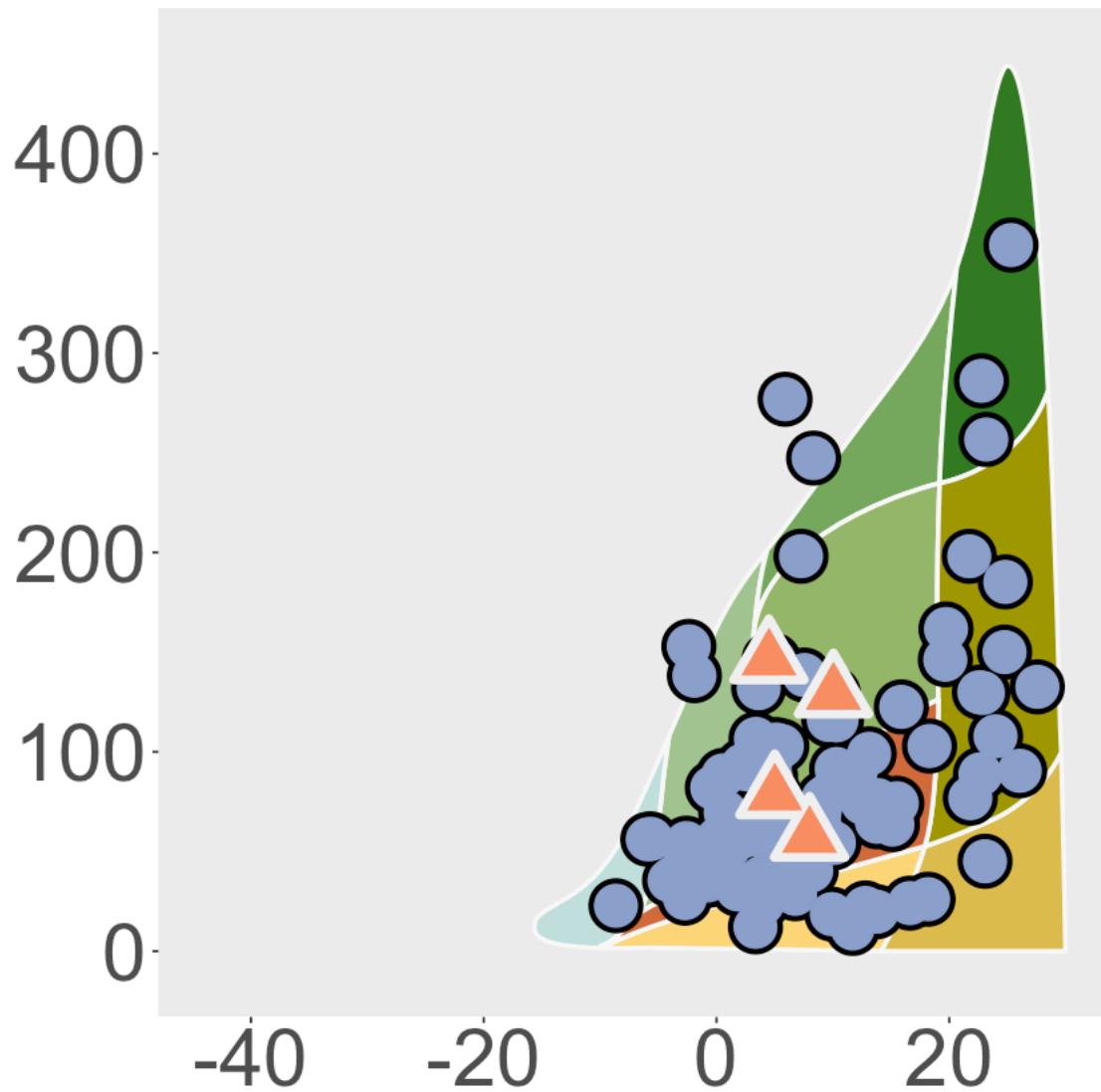
Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.



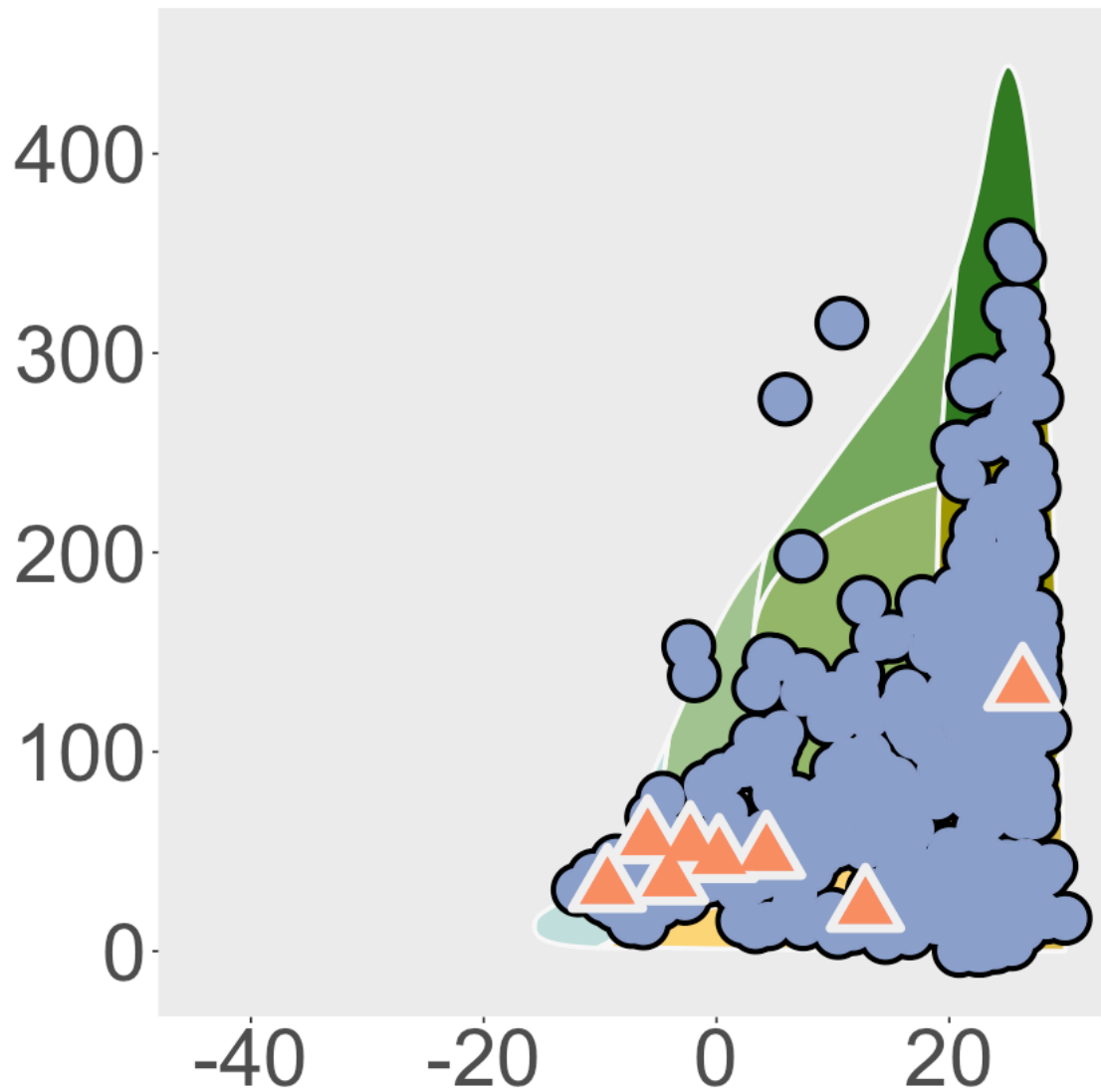
Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.



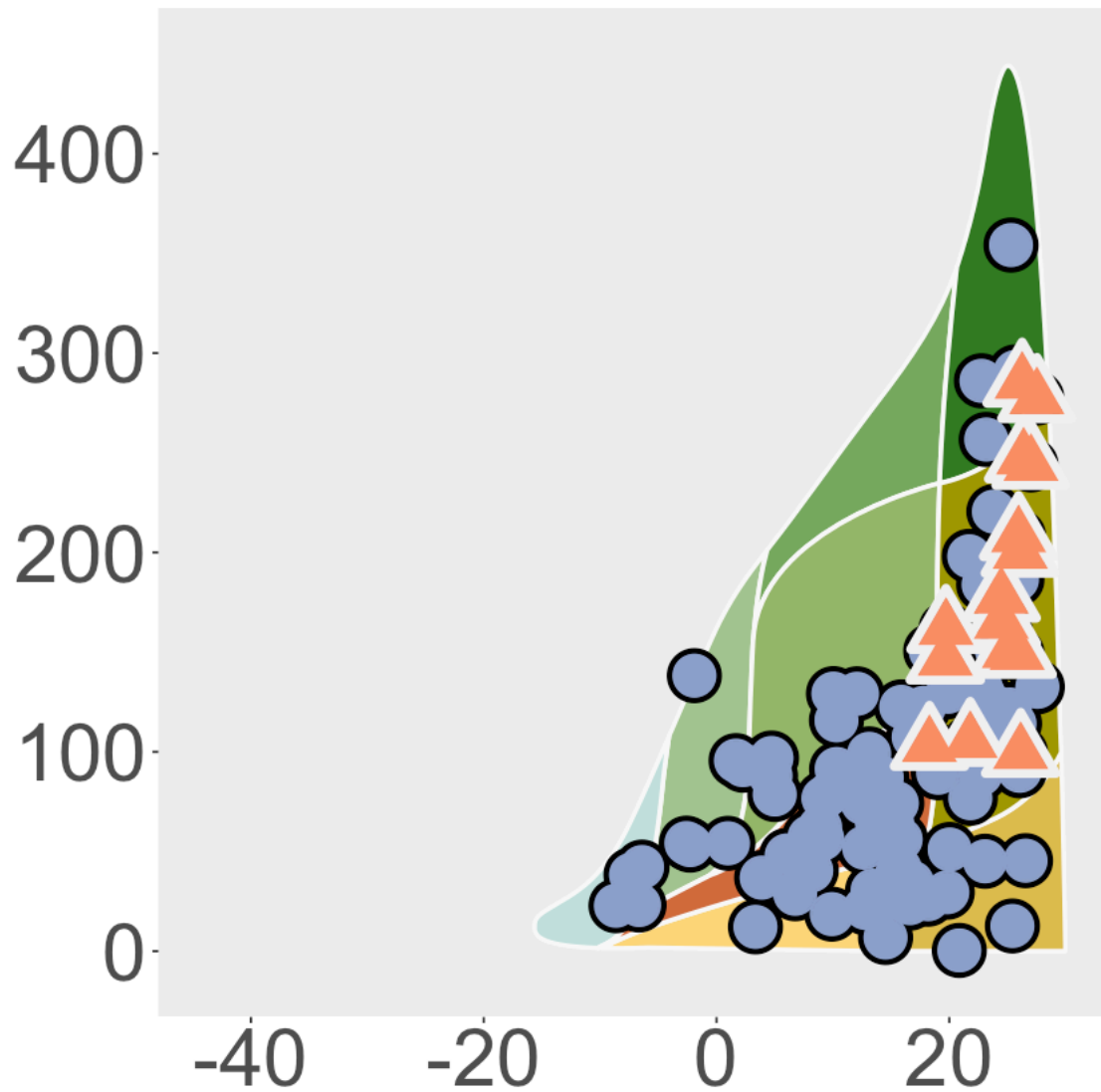
Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.



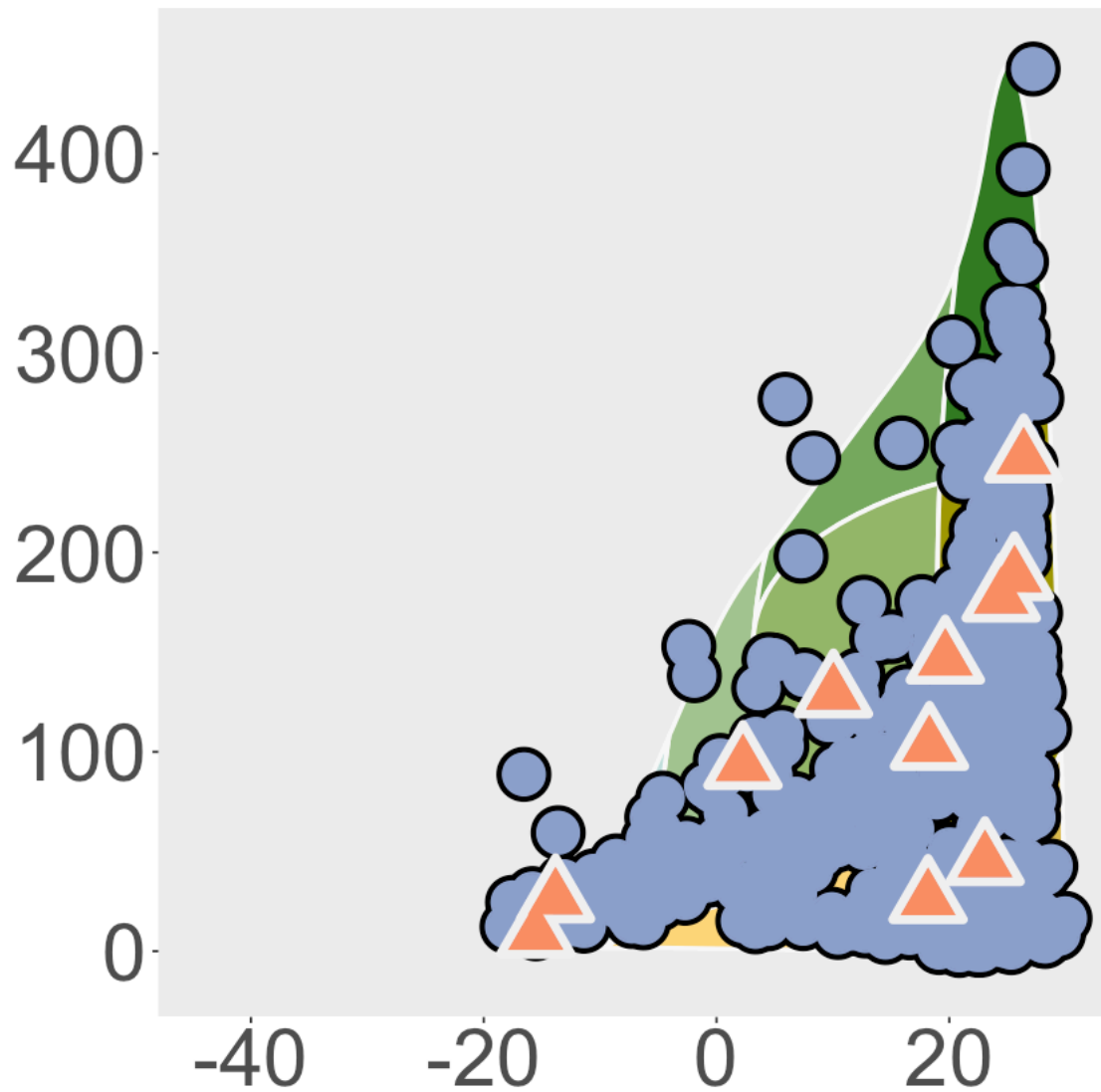
Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.



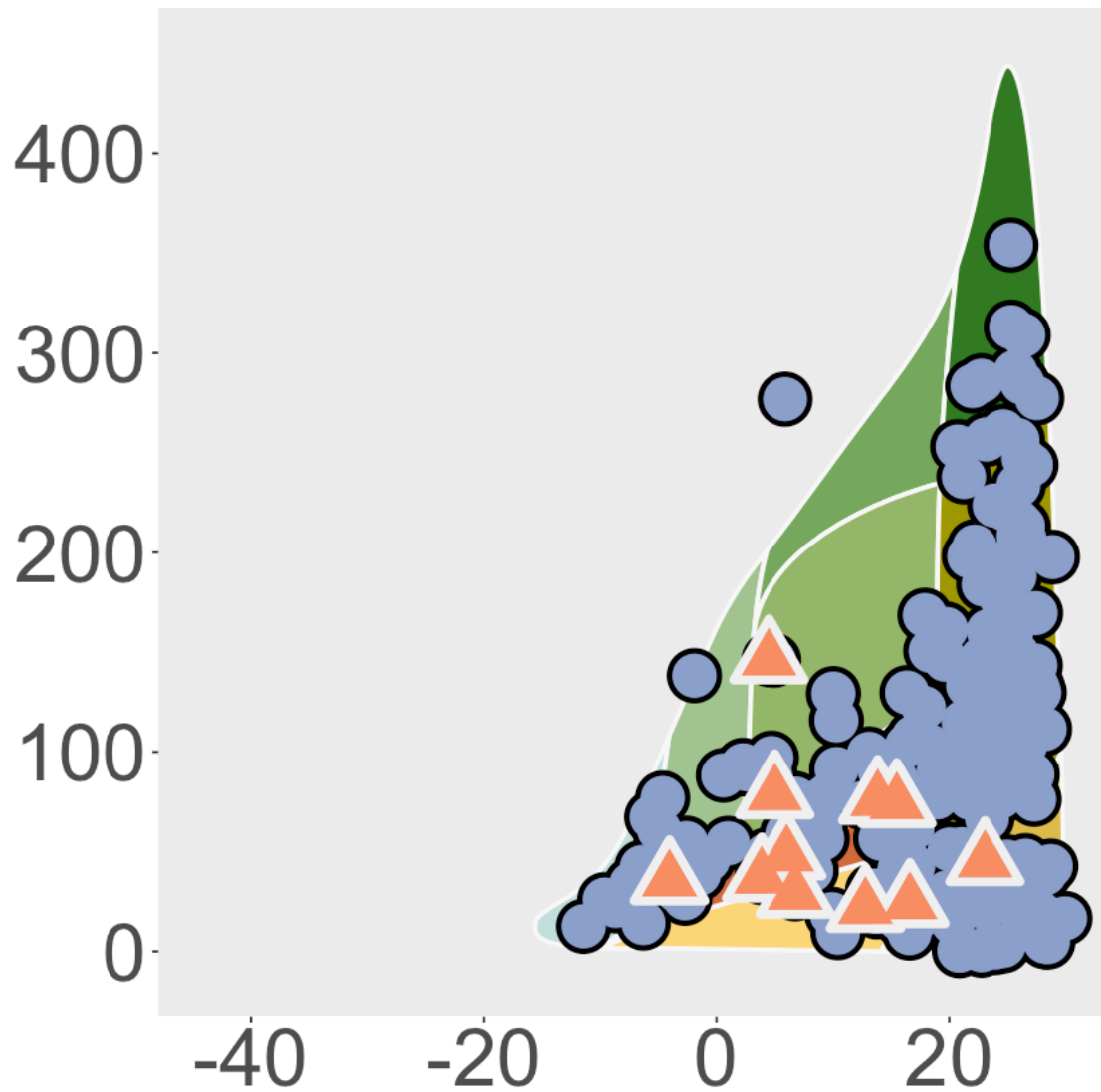
Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.



Scale for `y` is already present.

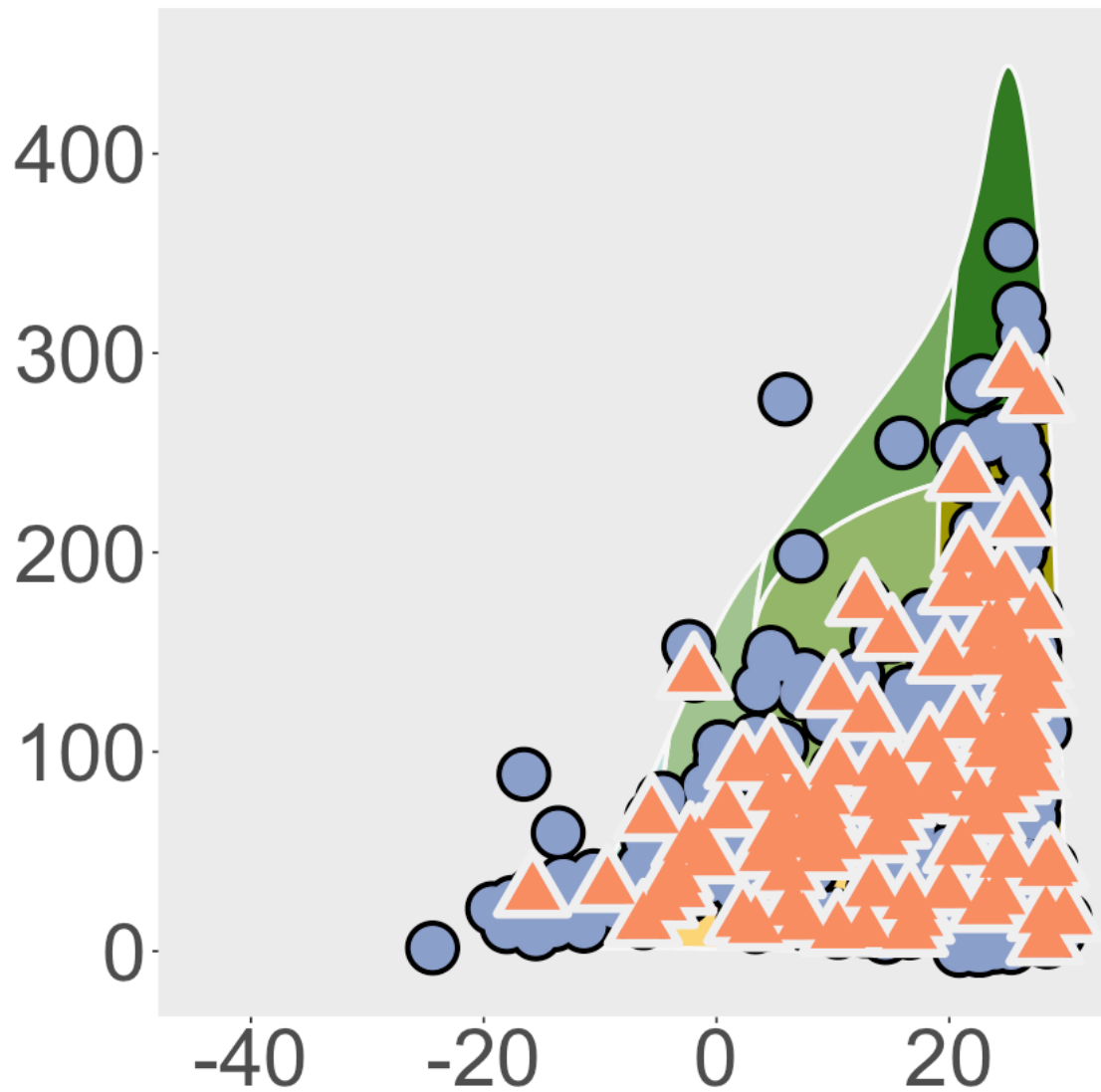
Adding another scale for `y`, which will replace the existing scale.



Scale for `y` is already present.

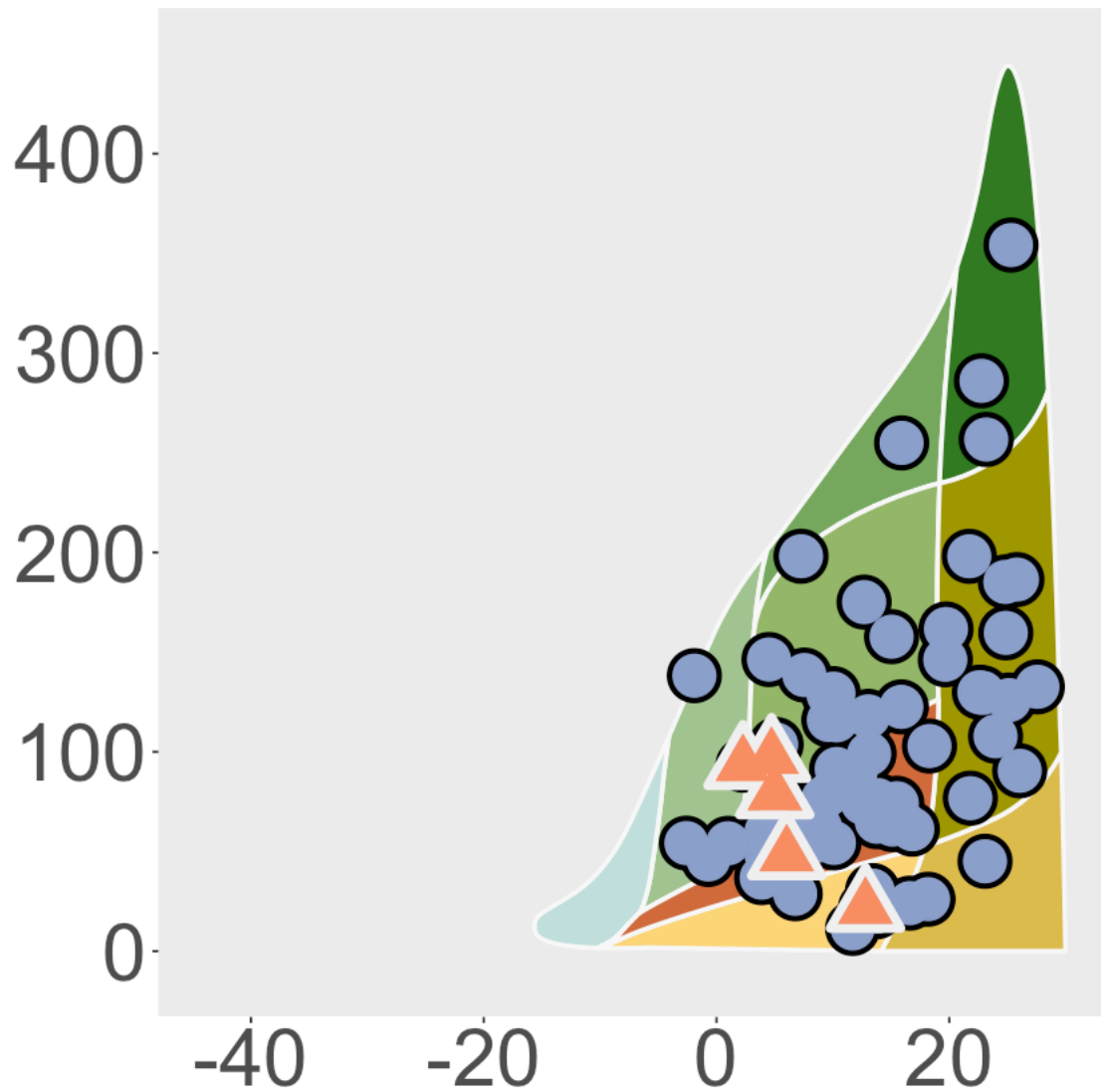
Adding another scale for `y`, which will replace the existing scale.



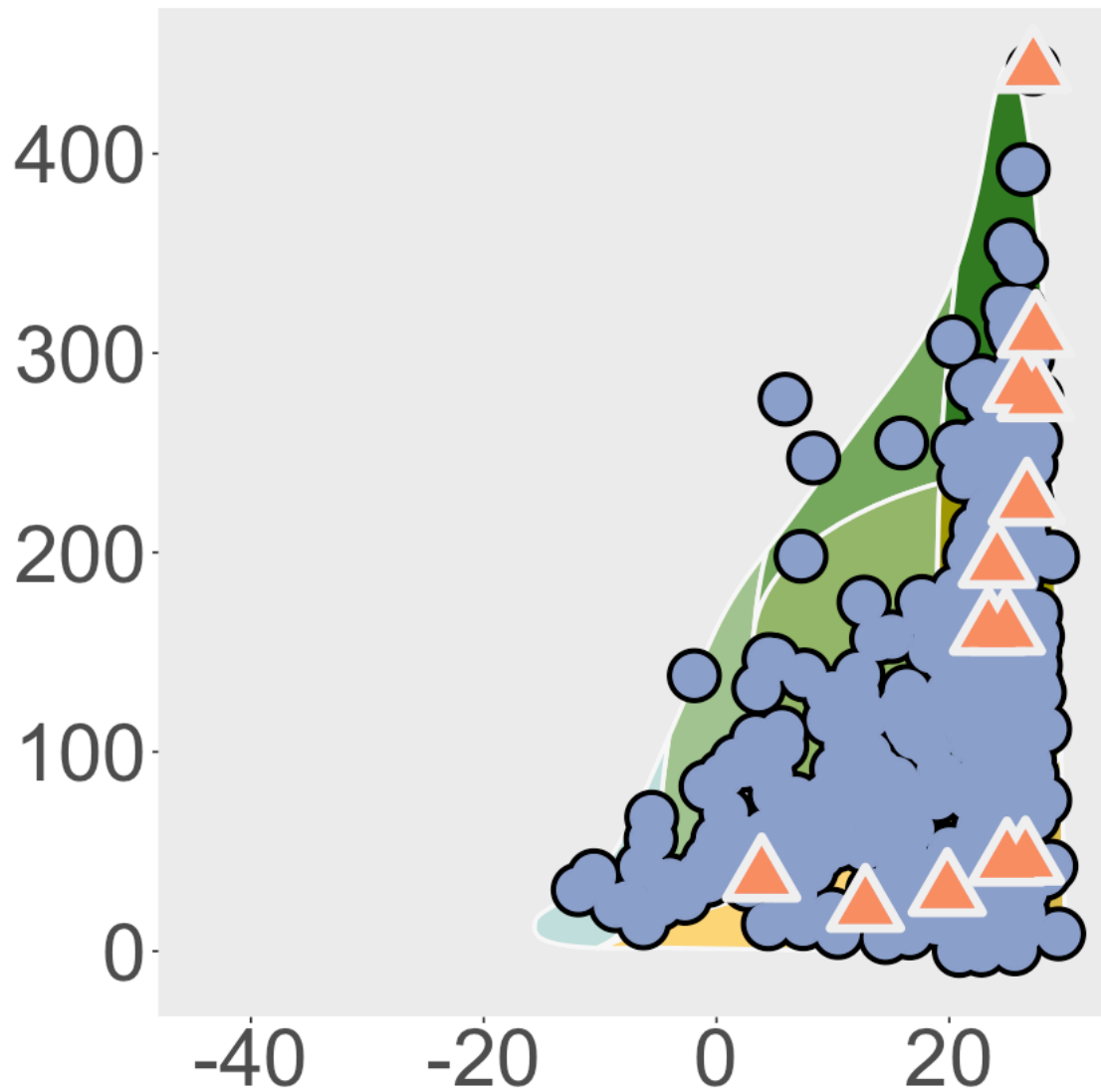


Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

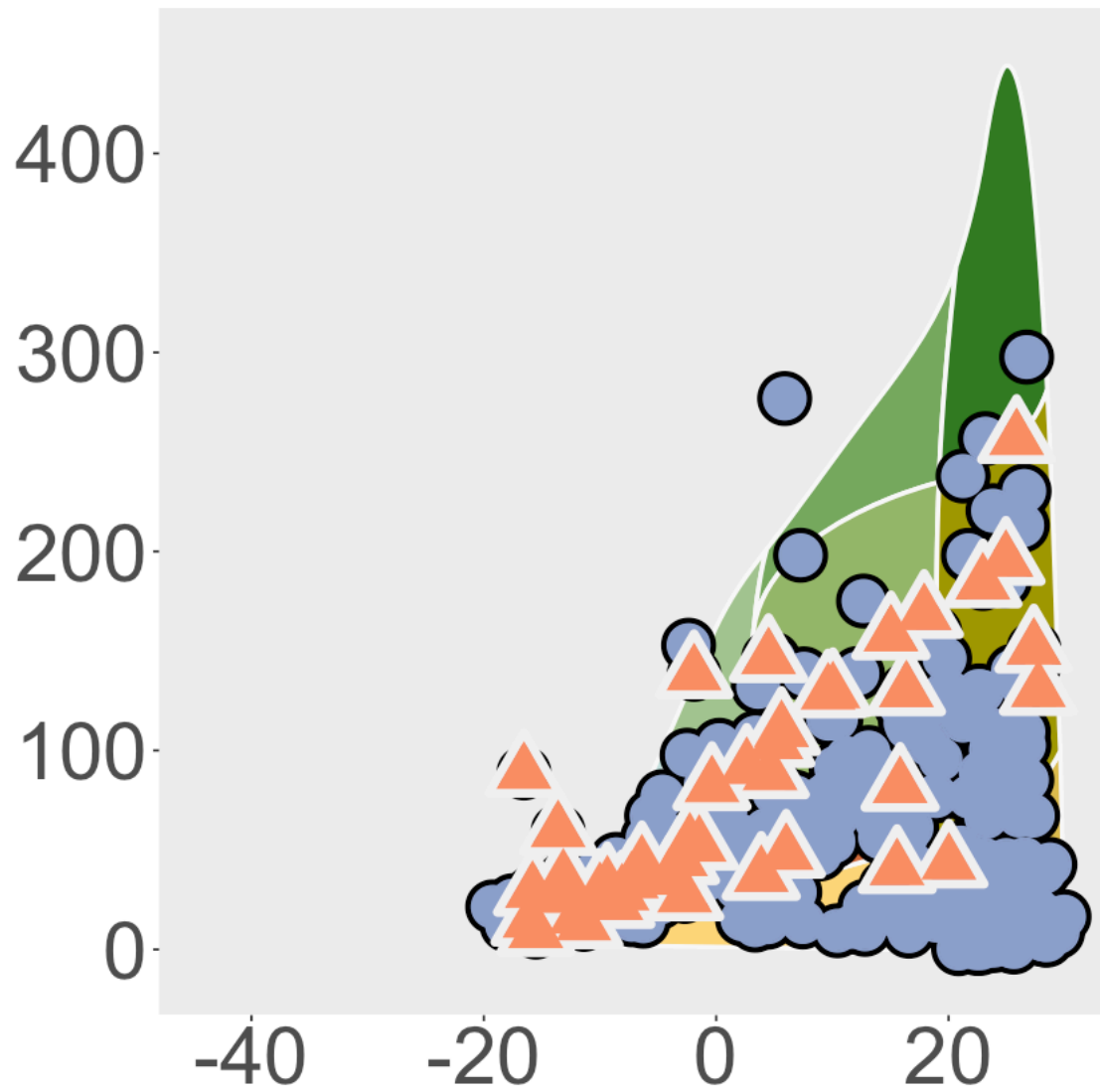


Scale for `y` is already present.  
Adding another scale for `y`, which will replace the existing scale.



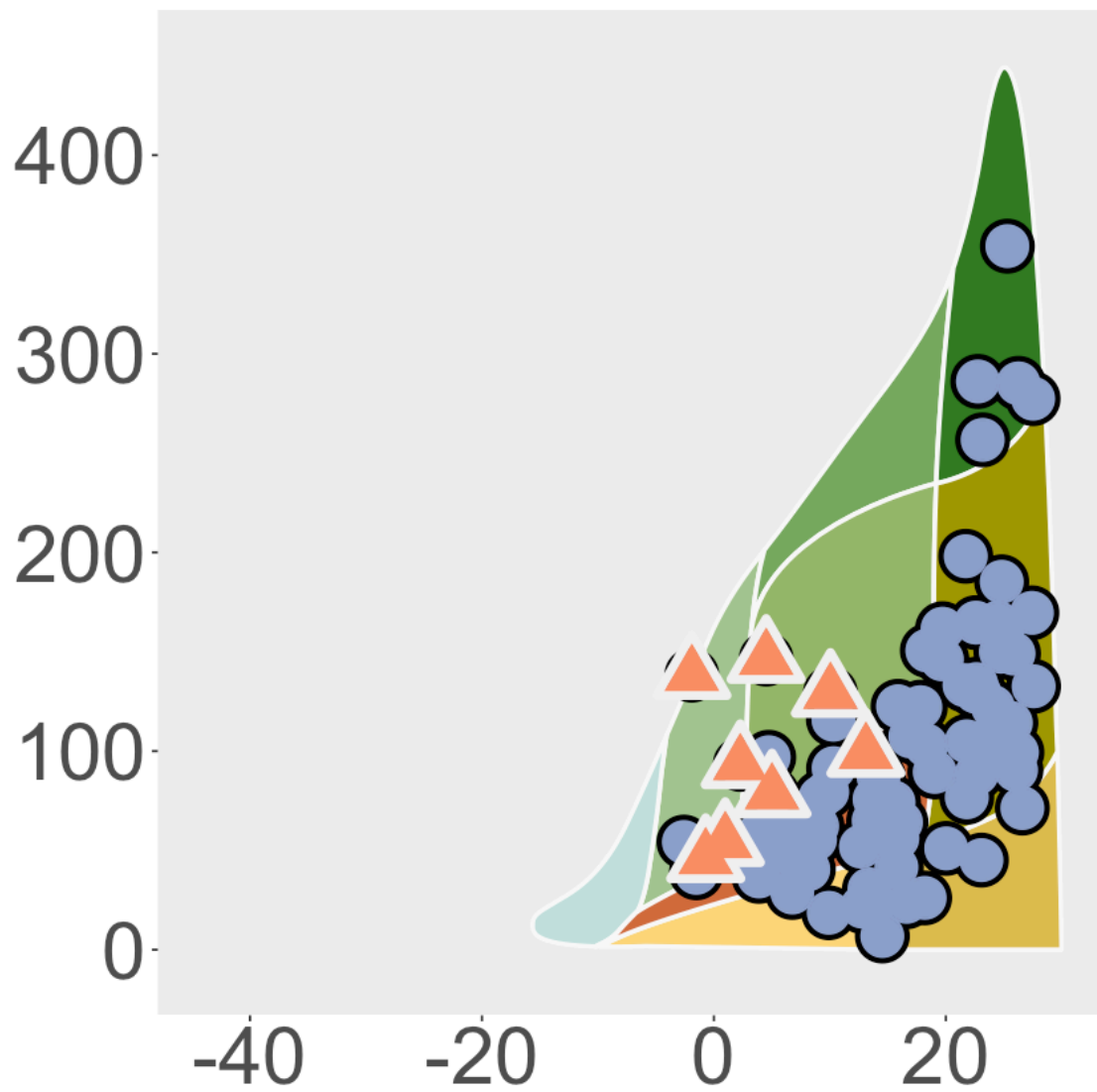
Scale for `y` is already present.

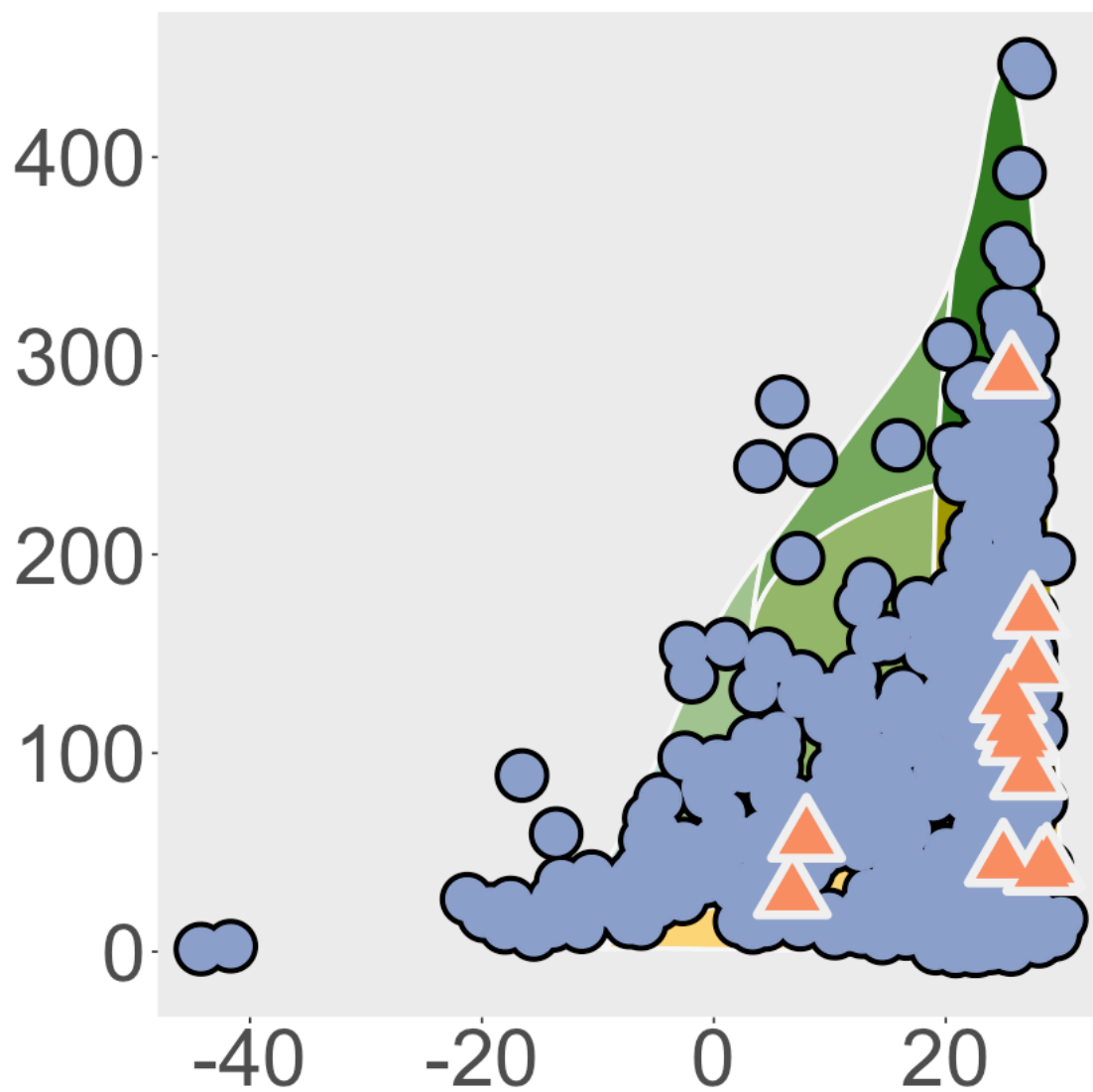
Adding another scale for `y`, which will replace the existing scale.

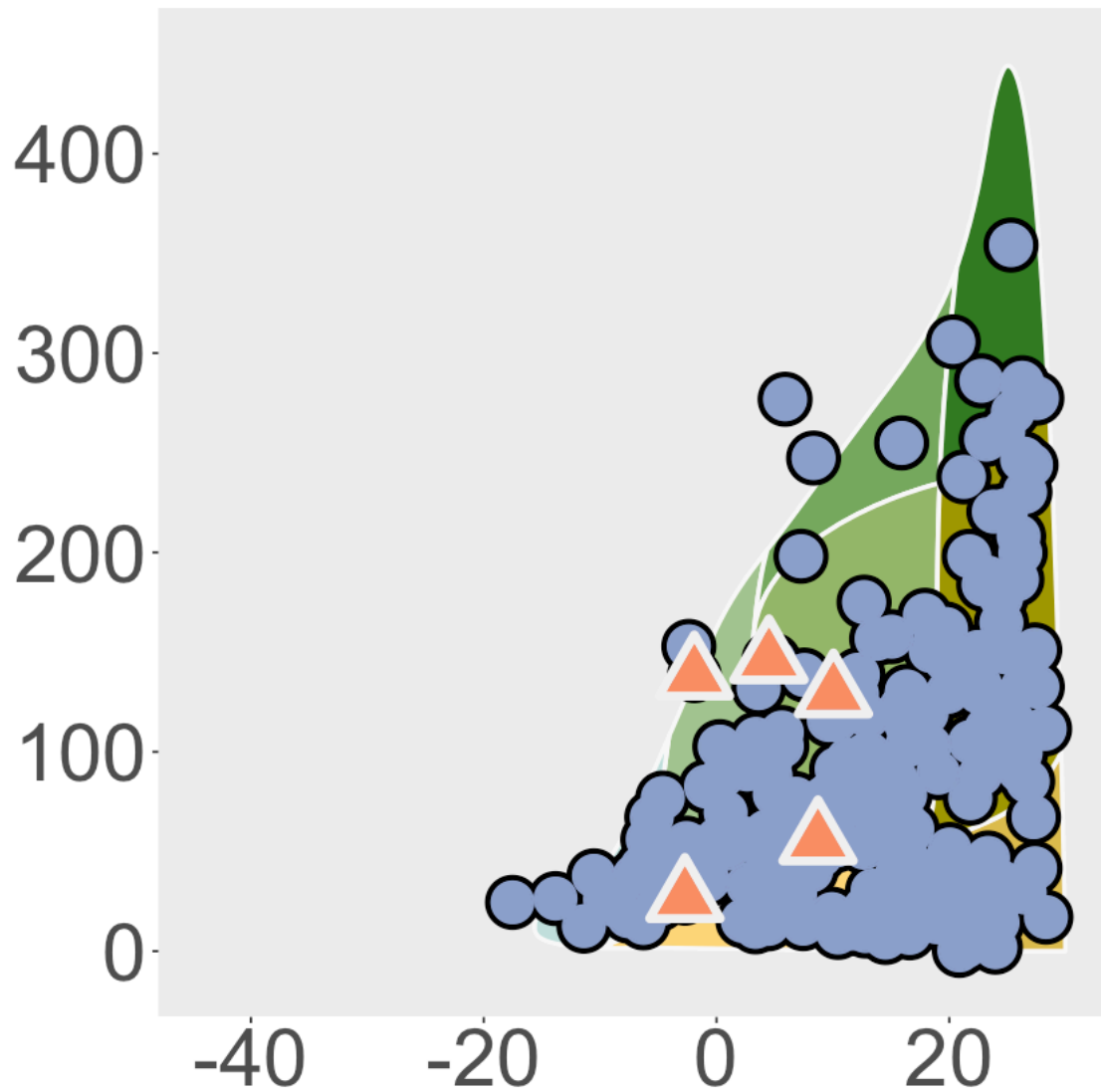


Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.







```
[6]: BiomeClass=function(target){
  g=ggplot(data[data$Species==target, ], aes(y=biome_id, color=index,
  ↪fill=index))+ geom_bar(width=0.4)

  g=g+ylim(0, 10)+scale_y_discrete(breaks = c("1","2","3","4","5","6", "7",
  ↪"8", "9", "10"),
    labels=c("Tropical seasonal forest/savanna",
              "Subtropical desert",
              "Temperate rain forest",
              "Tropical rain forest",
              "Woodland/shrubland",
              "Tundra",
              "Boreal forest",
```

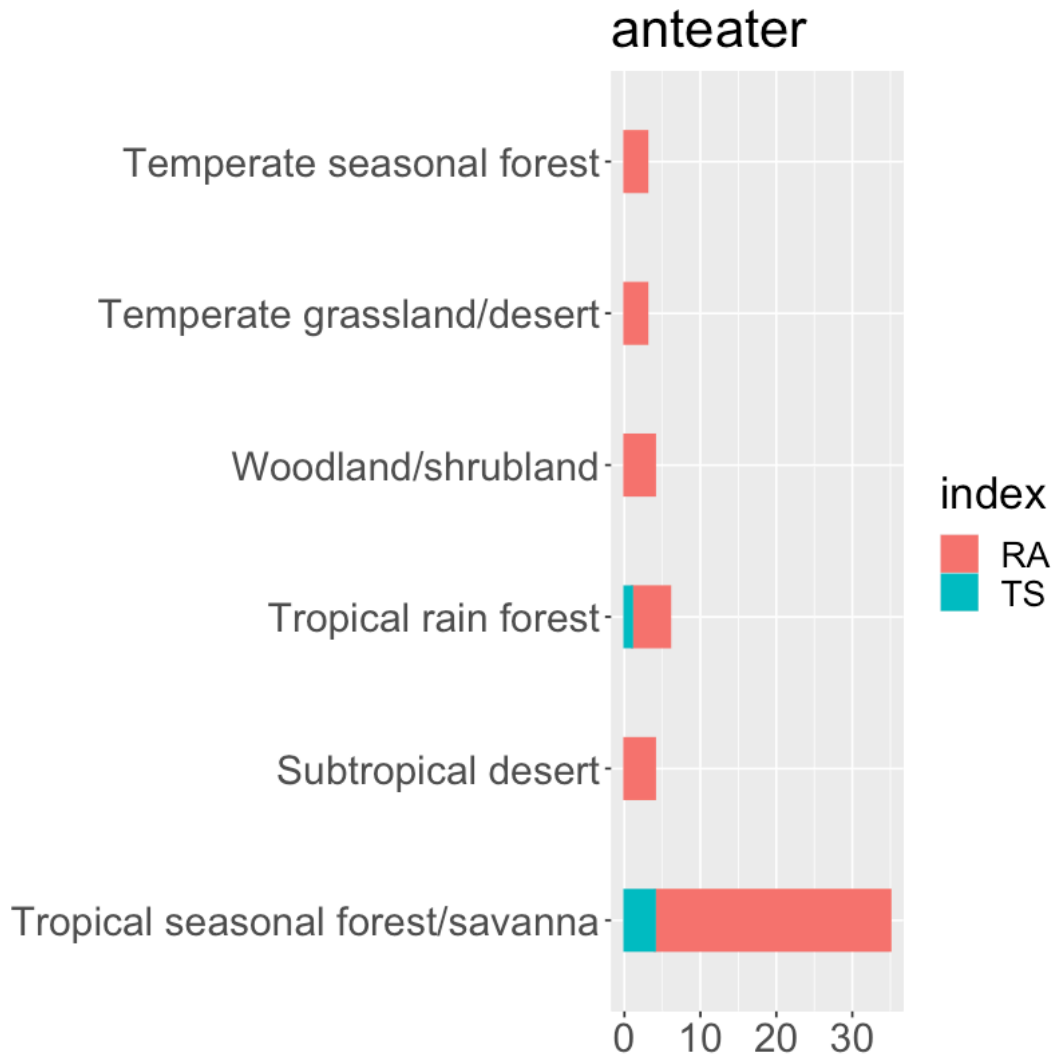
```

                                "Temperate grassland/desert",
                                "Temperate seasonal forest", "Outlier"))
  g=g+xlab("")+ylab("")+ggtitle(target)+theme(axis.text = element_text(size = 18), text=element_text(size=20))
  ggsave(paste0('BiomeClass_', target, '.pdf'))
  return (g)
}
s=unique(data$Species)
for (i in 1:length(s)){
  target=s[i]
  g=BiomeClass(target)
  print(g)
}

```

Scale for y is already present.  
 Adding another scale for y, which will replace the existing scale.  
 Saving 7 x 7 in image  
 Scale for y is already present.  
 Adding another scale for y, which will replace the existing scale.  
 Saving 7 x 7 in image  
 Scale for y is already present.  
 Adding another scale for y, which will replace the existing scale.  
 Saving 7 x 7 in image

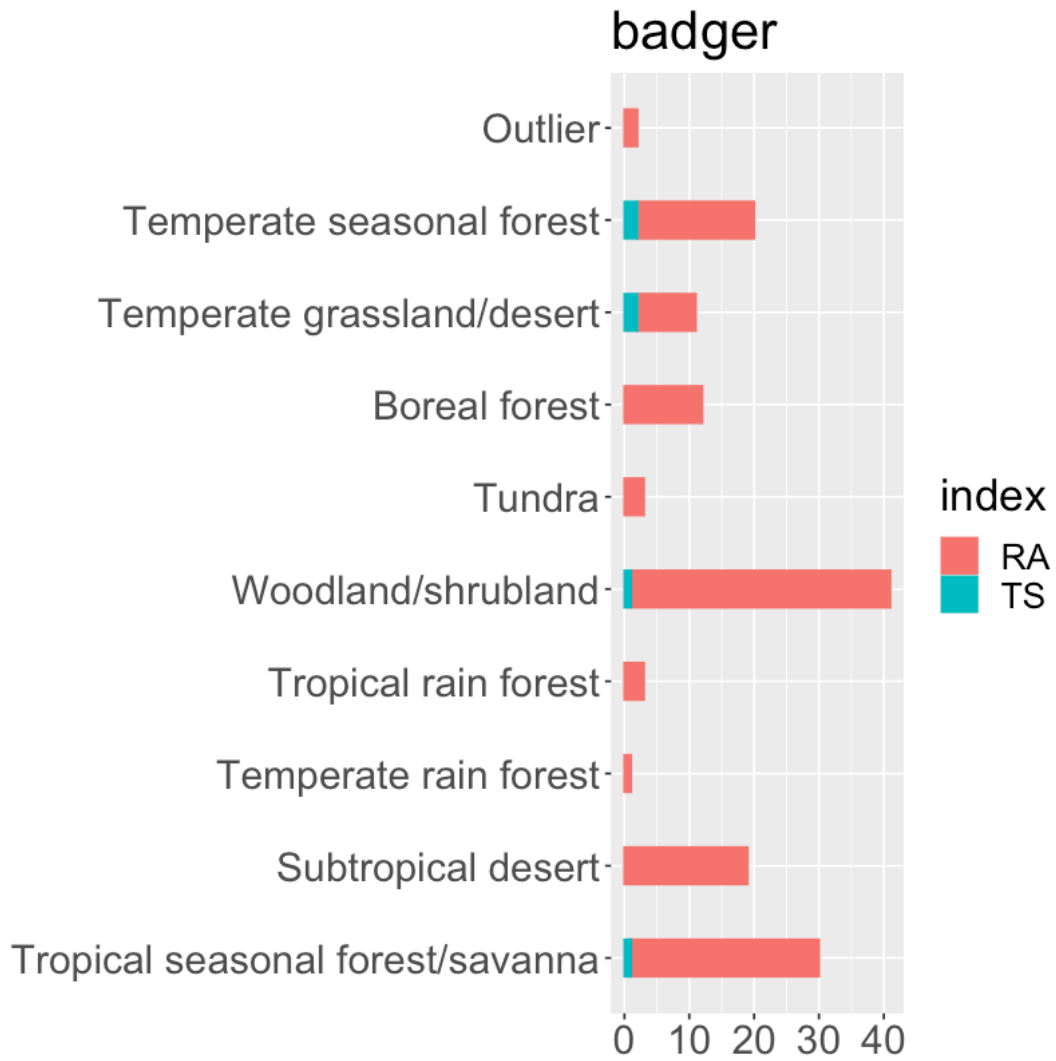




Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

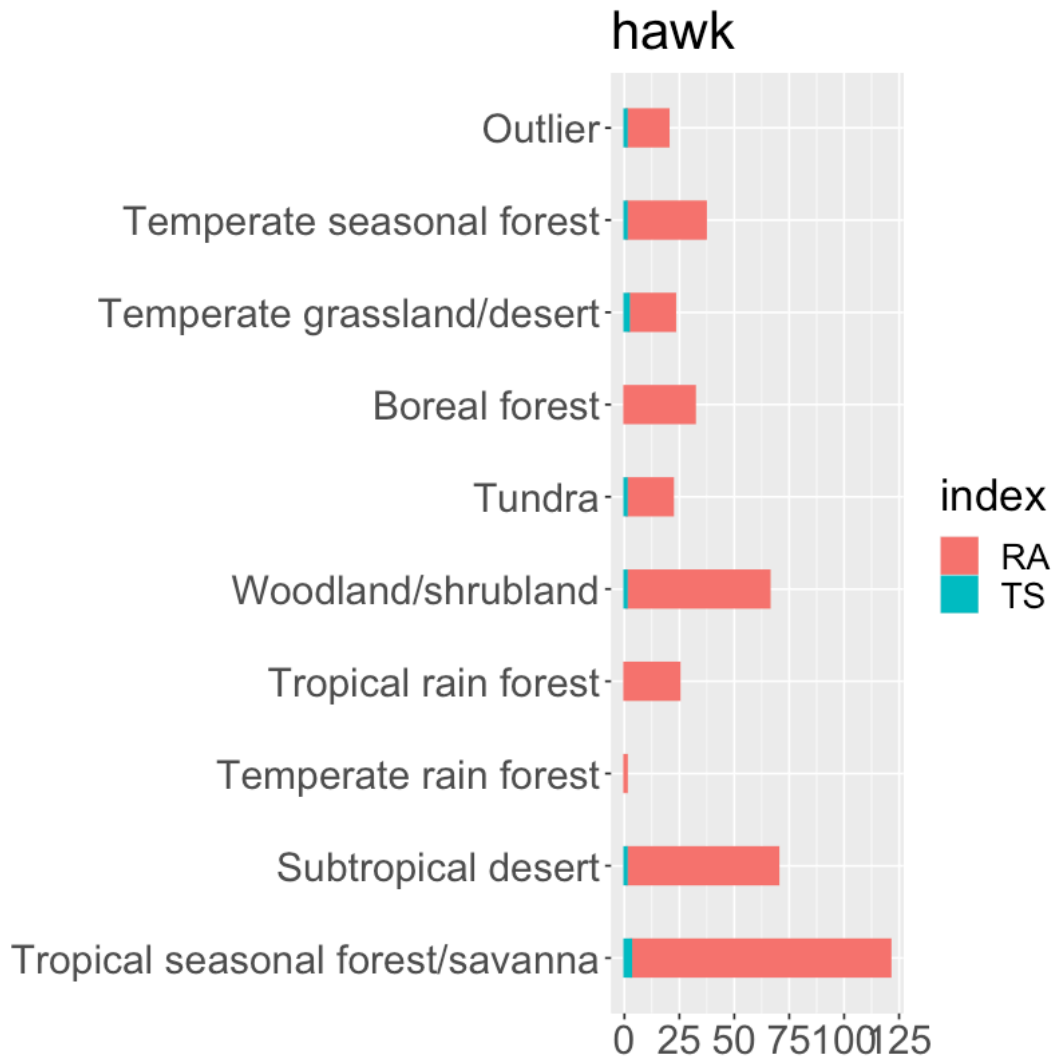
Saving 7 x 7 in image



Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

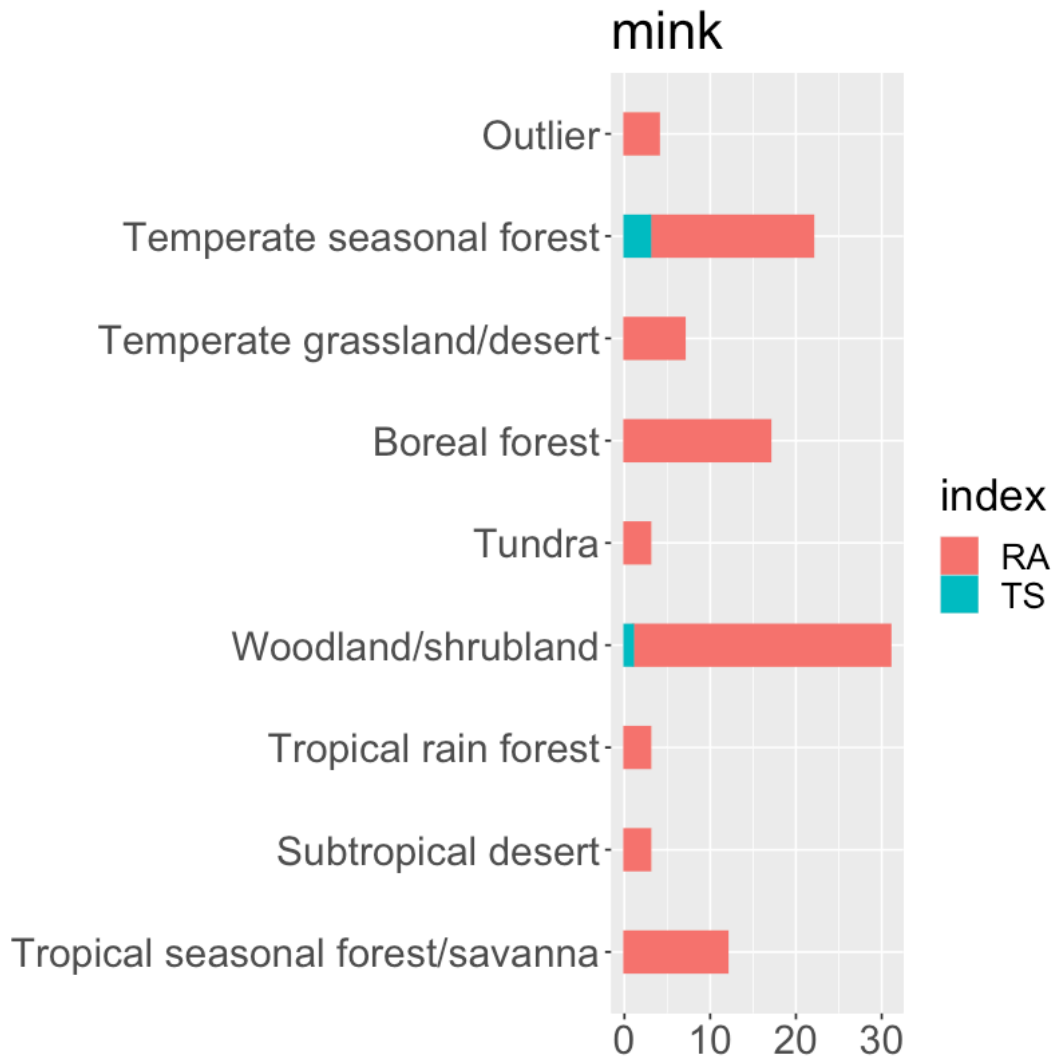
Saving 7 x 7 in image



Scale for y is already present.

Adding another scale for y, which will replace the existing scale.

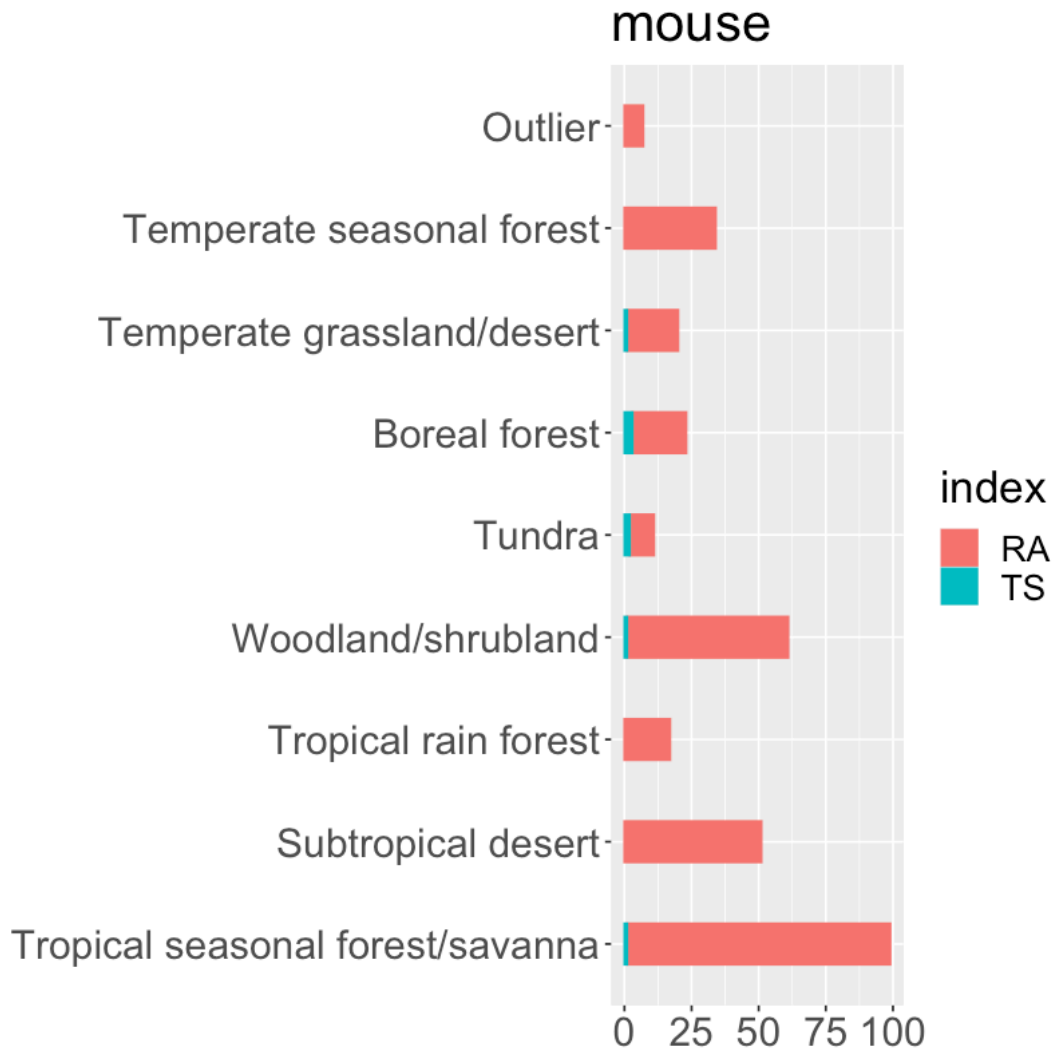
Saving 7 x 7 in image



Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

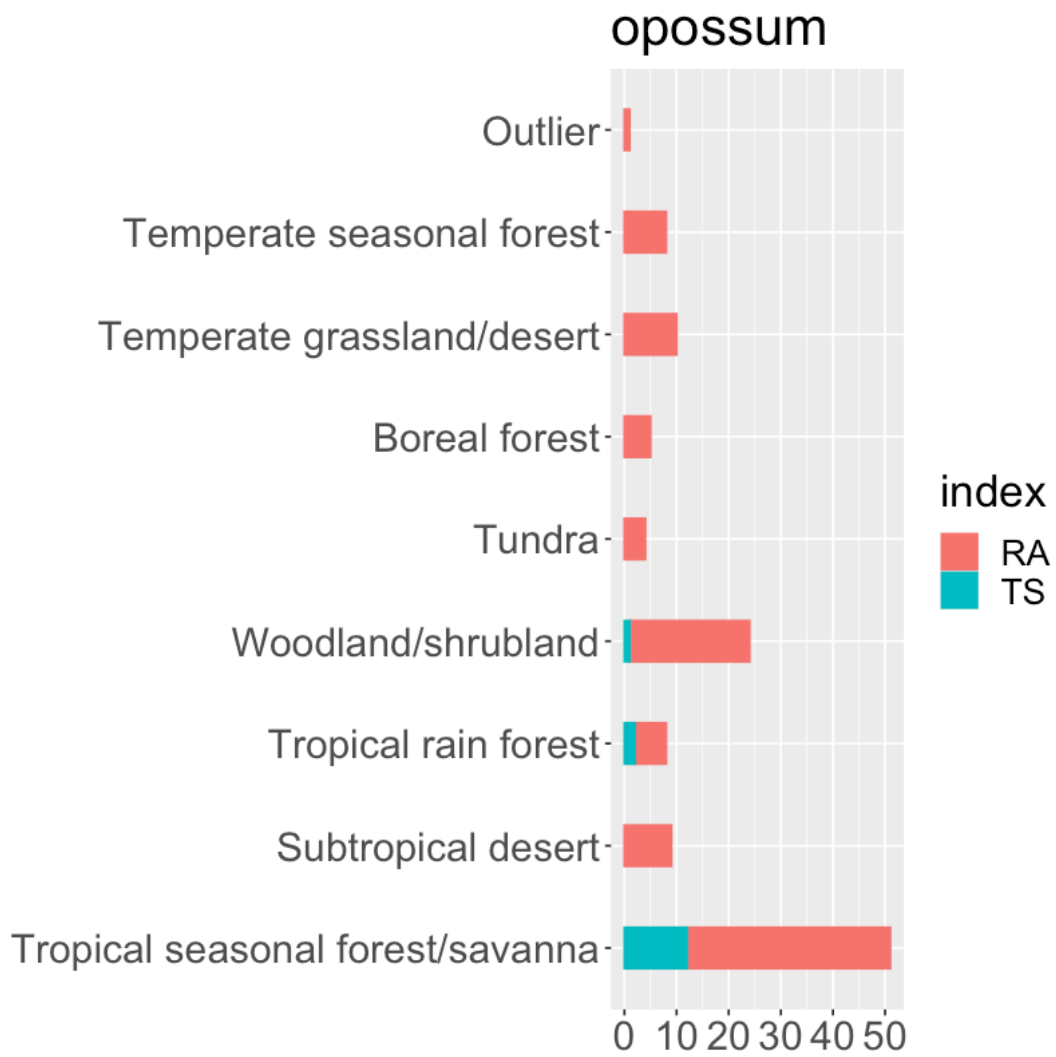
Saving 7 x 7 in image



Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

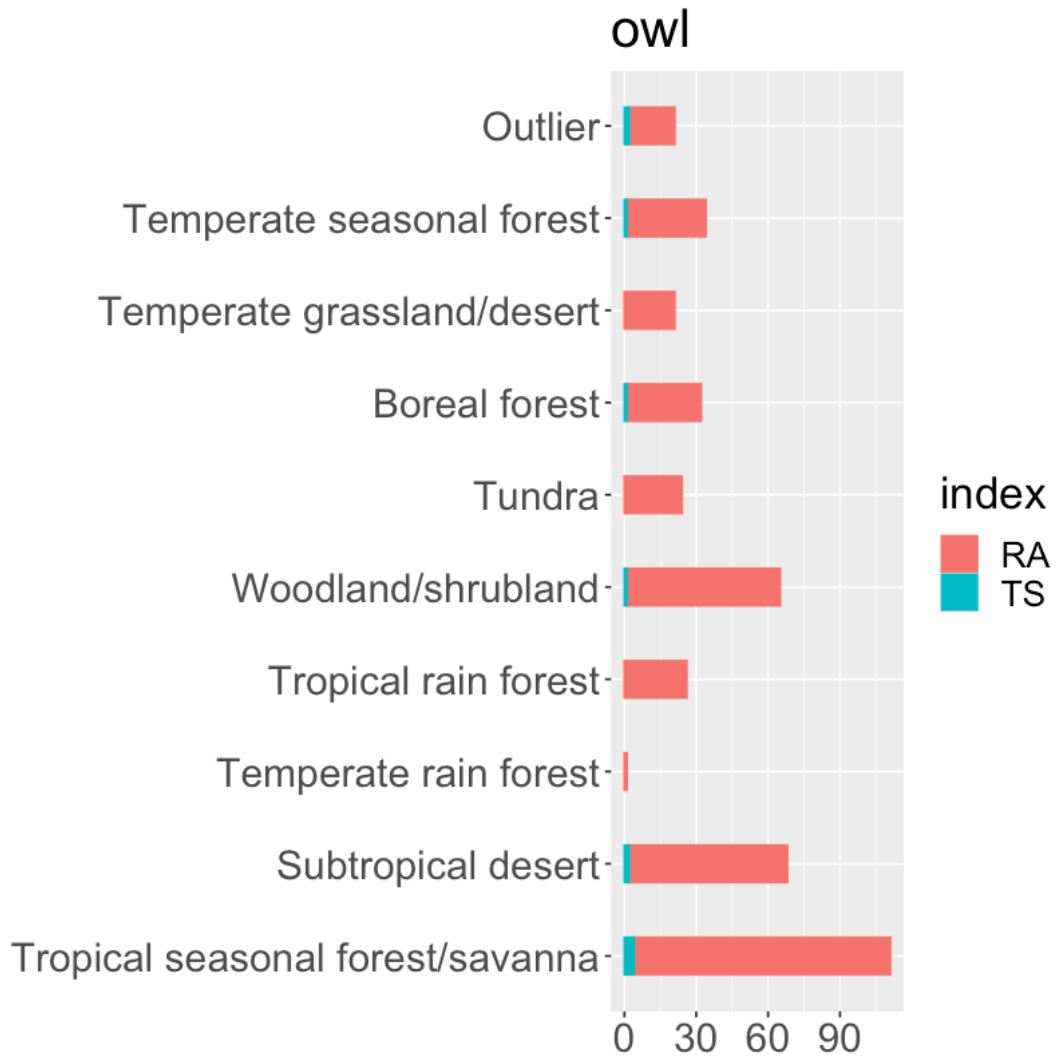
Saving 7 x 7 in image



Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

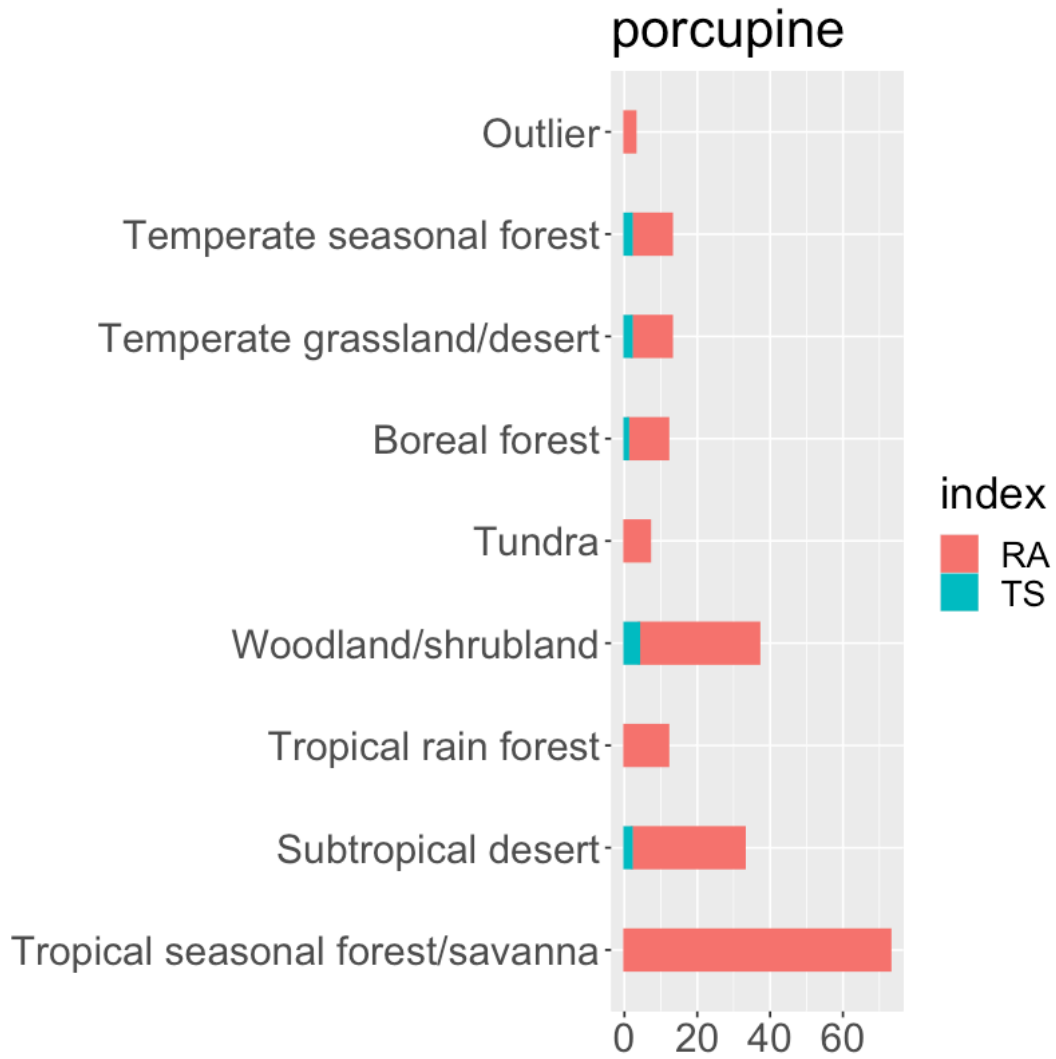
Saving 7 x 7 in image



Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

Saving 7 x 7 in image

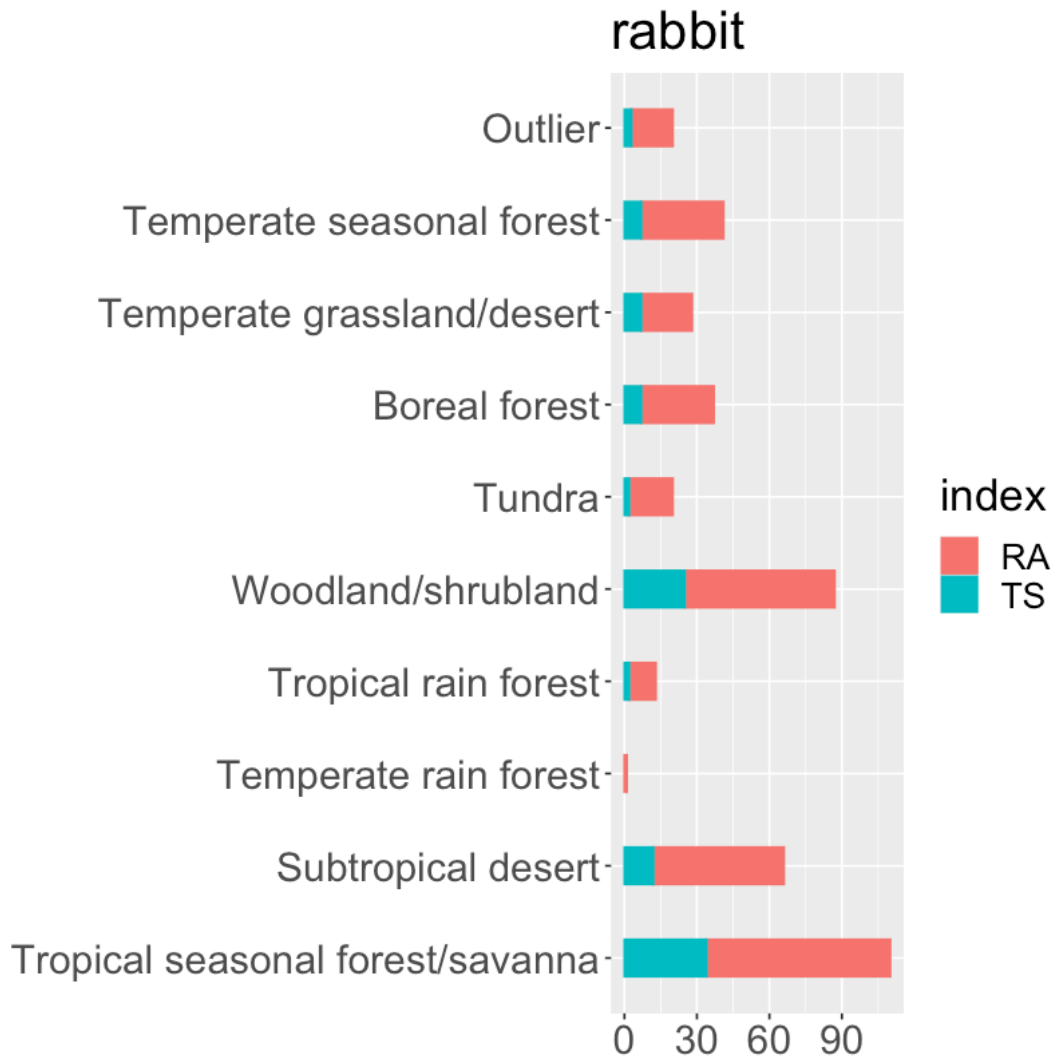


Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

Saving 7 x 7 in image

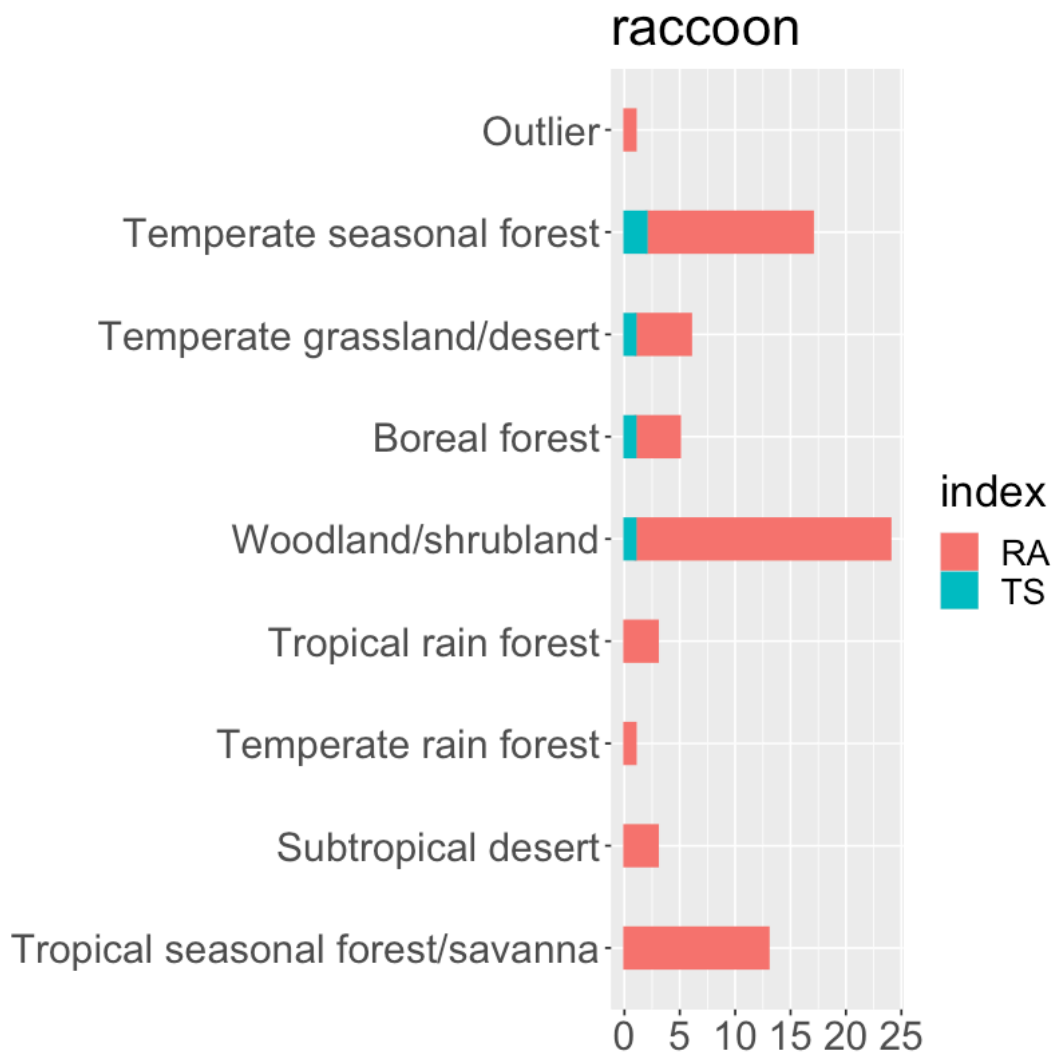




Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

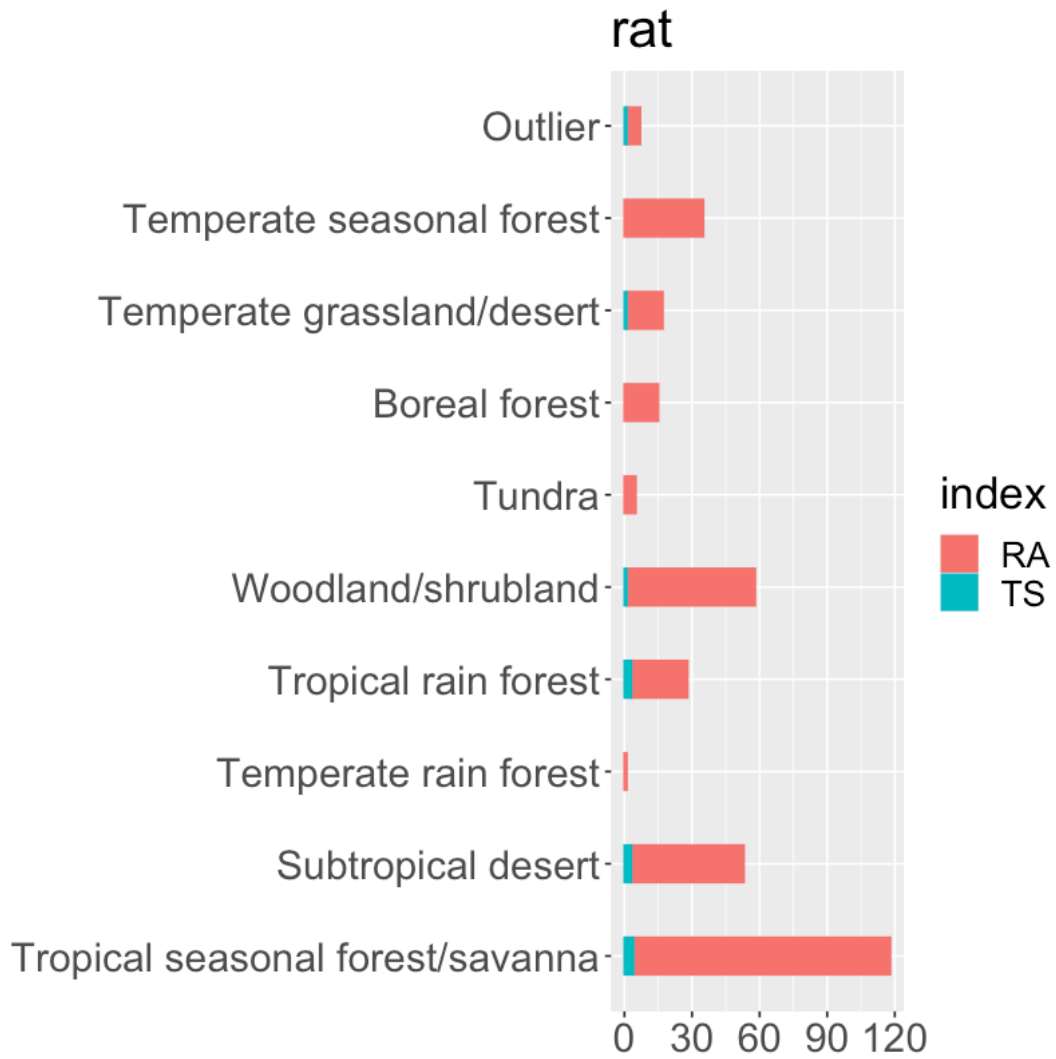
Saving 7 x 7 in image



Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

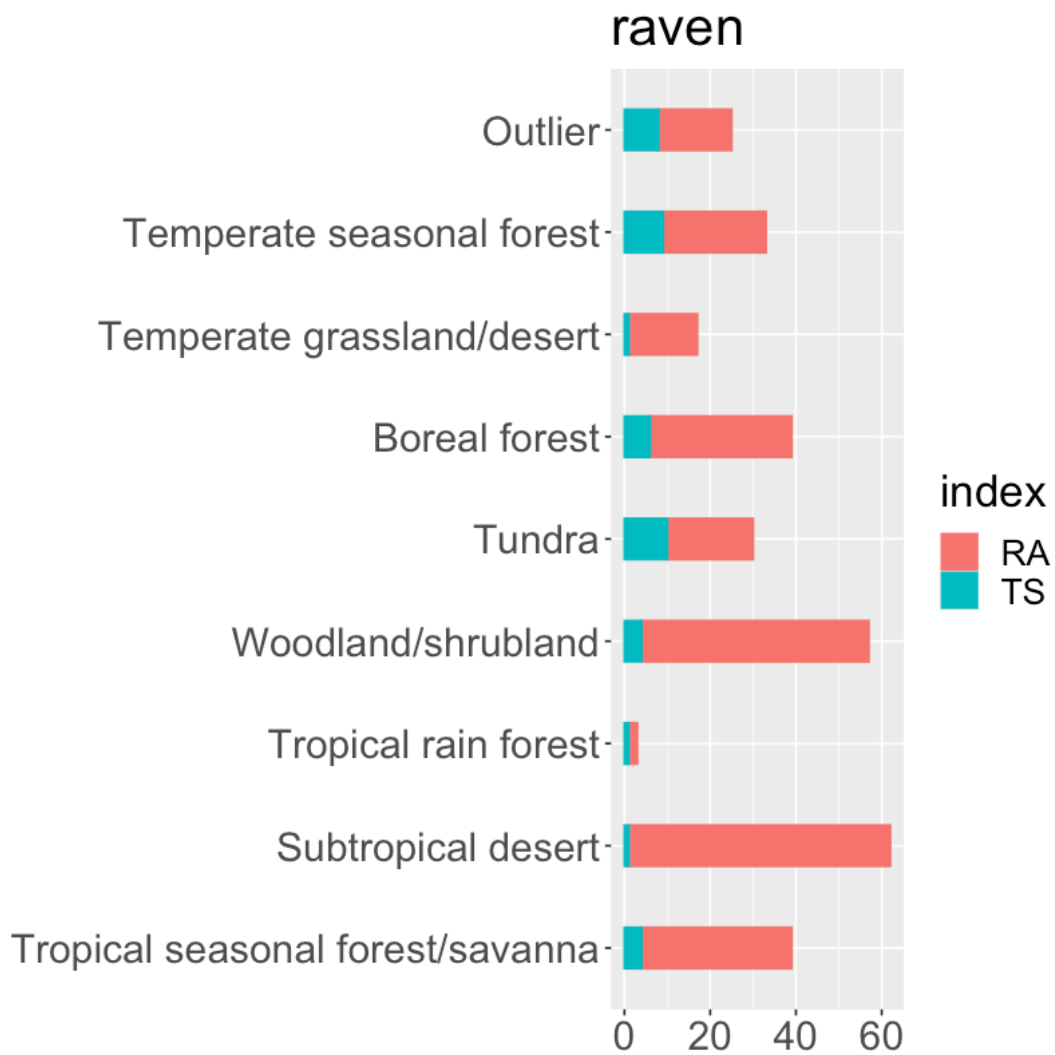
Saving 7 x 7 in image



Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

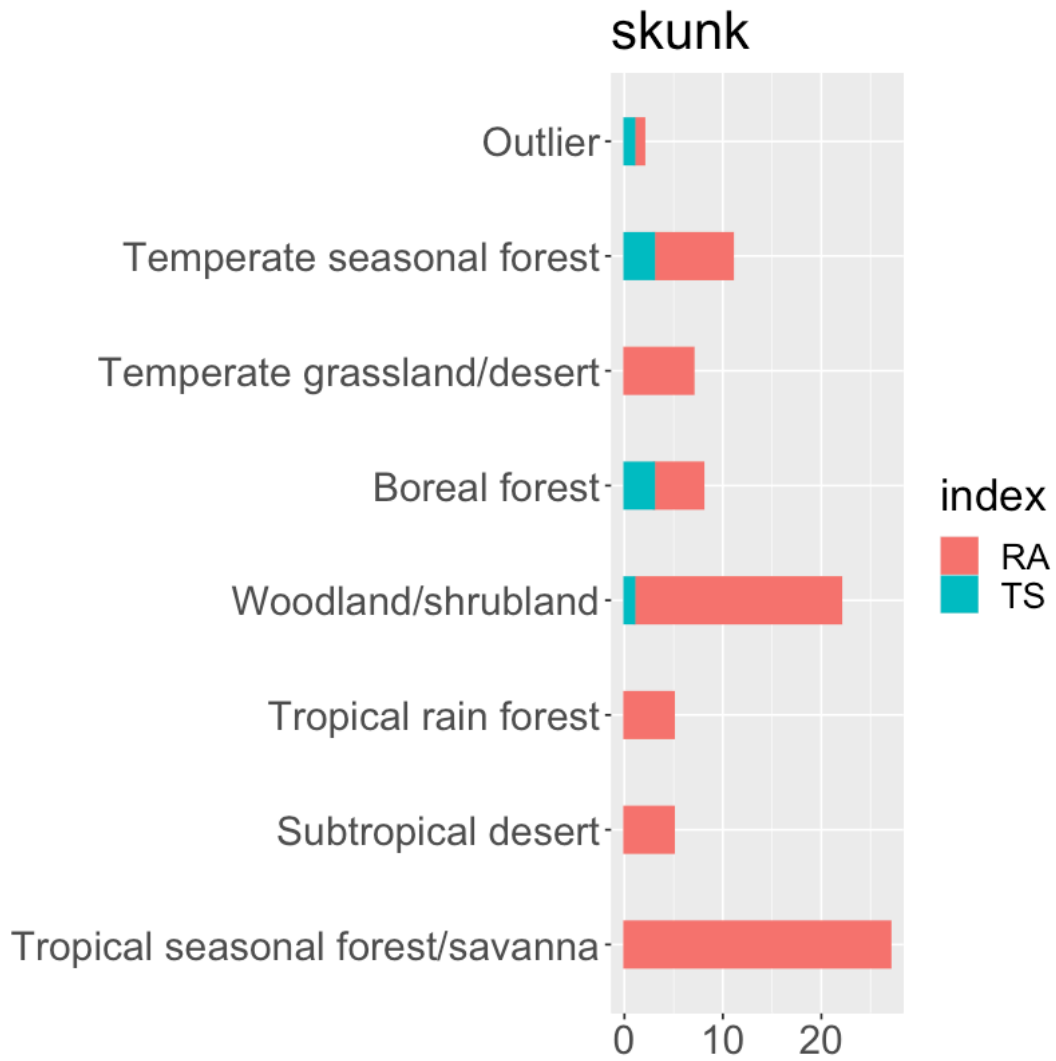
Saving 7 x 7 in image

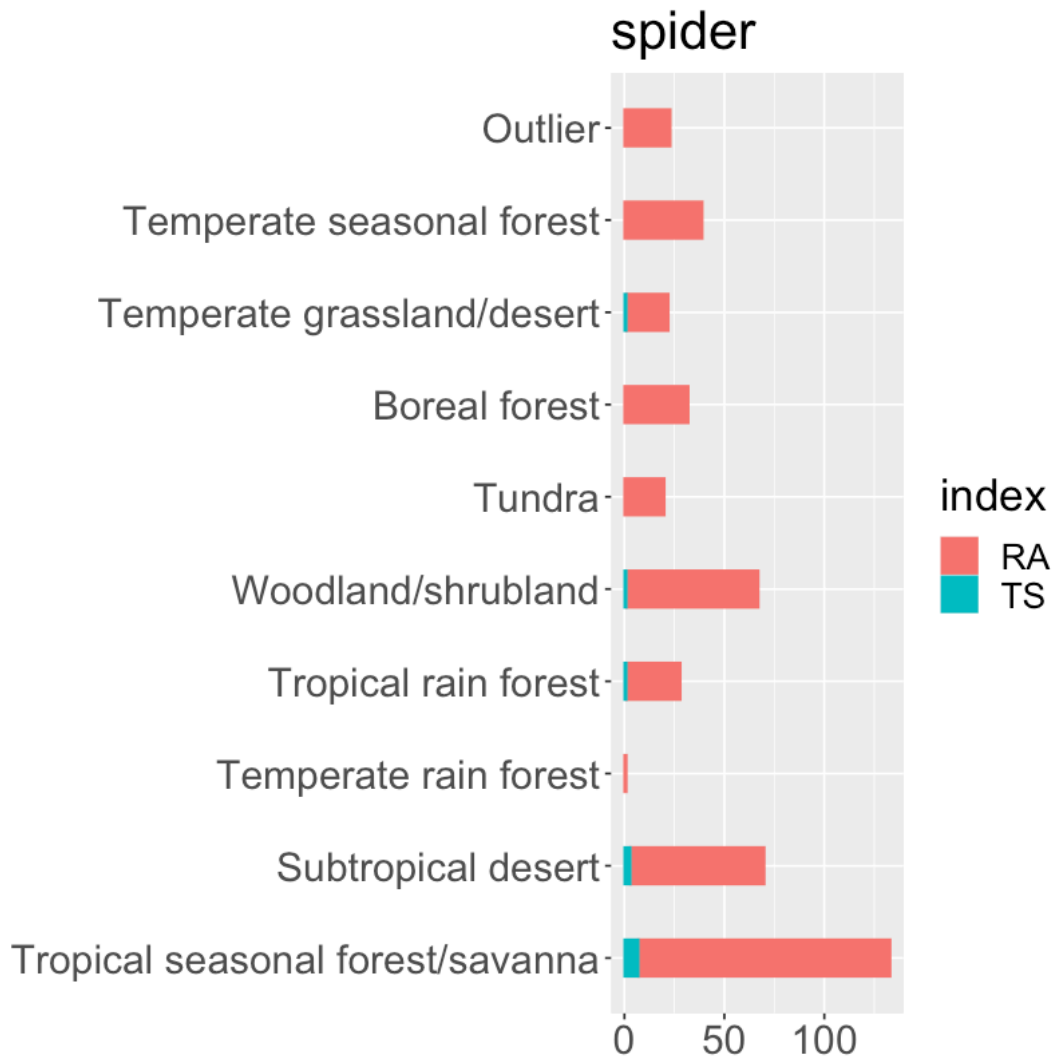


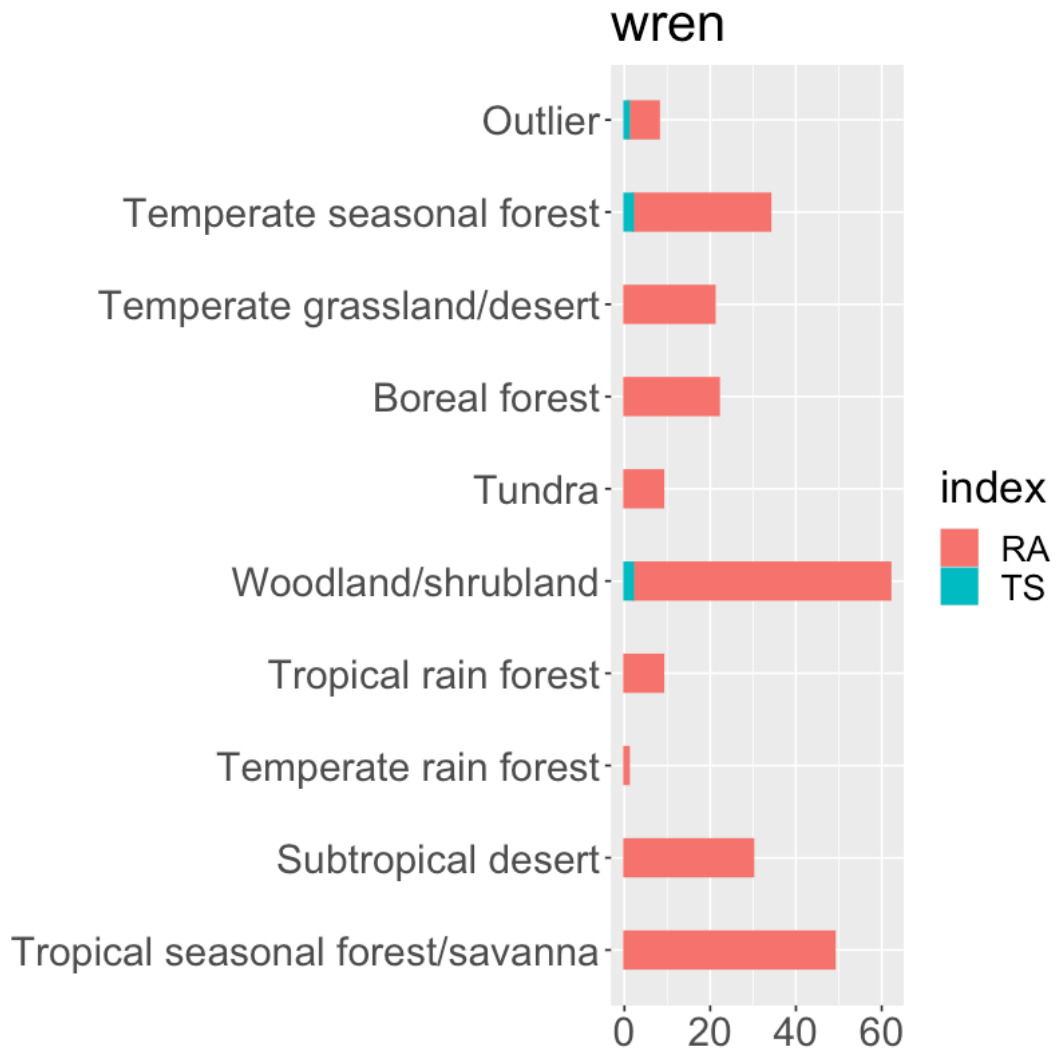
Scale for `y` is already present.

Adding another scale for `y`, which will replace the existing scale.

Saving 7 x 7 in image







```
[9]: install.packages('tidyr')  
library(dplyr)  
library(tibble)  
library(tidyr)
```

(dependency) 'purrr', 'stringr'

/var/folders/g4/3cvylr8j6mq822n672x35vn40000gr/T//RtmpK8jzbW/downloaded\_packages

```

: 'dplyr'

'package:raster'      :

intersect, select, union

'package:stats'      :

filter, lag

'package:base'      :

intersect, setdiff, setequal, union

: 'tidyr'

'package:raster'      :

extract

```

```

[33]: target='badger'
      print(target)
      d=data[data$Species== target, ]

      d_RA=d[d$index=='RA', ]
      freq_RA=c(nrow(d_RA[d_RA$biome_id==1,]),nrow(d_RA[d_RA$biome_id==2,]),
                nrow(d_RA[d_RA$biome_id==3,]), nrow(d_RA[d_RA$biome_id==4,]),
                nrow(d_RA[d_RA$biome_id==5,]), nrow(d_RA[d_RA$biome_id==6,]),
                nrow(d_RA[d_RA$biome_id==7,]),nrow(d_RA[d_RA$biome_id==8,]),
                nrow(d_RA[d_RA$biome_id==9,]),nrow(d_RA[d_RA$biome_id==10,]))
      freq_RA=freq_RA/sum(freq_RA)
      d_TS=d[d$index=='TS', ]
      freq_TS=c(nrow(d_TS[d_TS$biome_id==1,]),nrow(d_TS[d_TS$biome_id==2,]),
                nrow(d_TS[d_TS$biome_id==3,]), nrow(d_TS[d_TS$biome_id==4,]),
                nrow(d_TS[d_TS$biome_id==5,]), nrow(d_TS[d_TS$biome_id==6,]),
                nrow(d_TS[d_TS$biome_id==7,]),nrow(d_TS[d_TS$biome_id==8,]),
                nrow(d_TS[d_TS$biome_id==9,]),nrow(d_TS[d_TS$biome_id==10,]))
      drop=c()
      for (i in 1:10){

```



```

    if (freq_TS[i]==0 && freq_RA[i]==0){
      drop=c(drop, i)
    }
  }

  if(length(drop)>0){
    chisq.test(freq_TS[-c(drop)], p=freq_RA[-c(drop)])
  } else{
    chisq.test(freq_TS, p=freq_RA)
  }
}

```

```

[1] "badger"
[1] 0.213235294 0.139705882 0.007352941 0.022058824 0.294117647 0.022058824
[7] 0.088235294 0.066176471 0.132352941 0.014705882
[1] 1 0 0 0 1 0 0 2 2 0

```

```

Warning message in chisq.test(freq_TS, p = freq_RA):
"          "

```

Chi-squared test for given probabilities

```

data: freq_TS
X-squared = 10.459, df = 9, p-value = 0.3146

```

```

[11]: pvals=c()
for (i in 1:length(s)){
  target=s[i]
  print(target)
  d=data[data$Species== target, ]

  d_RA=d[d$index=='RA', ]
  freq_RA=c(nrow(d_RA[d_RA$biome_id==1,]),nrow(d_RA[d_RA$biome_id==2,]),
    nrow(d_RA[d_RA$biome_id==3,]), nrow(d_RA[d_RA$biome_id==4,]),
    nrow(d_RA[d_RA$biome_id==5,]), nrow(d_RA[d_RA$biome_id==6,]),
    nrow(d_RA[d_RA$biome_id==7,]),nrow(d_RA[d_RA$biome_id==8,]),
    nrow(d_RA[d_RA$biome_id==9,]),nrow(d_RA[d_RA$biome_id==10,]))
  freq_RA=freq_RA/sum(freq_RA)
  d_TS=d[d$index=='TS', ]
  freq_TS=c(nrow(d_TS[d_TS$biome_id==1,]),nrow(d_TS[d_TS$biome_id==2,]),
    nrow(d_TS[d_TS$biome_id==3,]), nrow(d_TS[d_TS$biome_id==4,]),
    nrow(d_TS[d_TS$biome_id==5,]), nrow(d_TS[d_TS$biome_id==6,]),
    nrow(d_TS[d_TS$biome_id==7,]),nrow(d_TS[d_TS$biome_id==8,]),
    nrow(d_TS[d_TS$biome_id==9,]),nrow(d_TS[d_TS$biome_id==10,]))
  drop=c()
  for (i in 1:10){

```

```

    if (freq_TS[i]==0 && freq_RA[i]==0){
      drop=c(drop, i)
    }
  }

  if(length(drop)>0){
    ans=chisq.test(freq_TS[-c(drop)], p=freq_RA[-c(drop)], B=1000000)
  } else{
    ans=chisq.test(freq_TS, p=freq_RA, B=1000000)
  }
  print(ans)
  pvals=c(pvals, ans$p.value)
}
#print(pvals)
p.adjust(pvals, 'fdr')

```

```
[1] "anteater"
```

```
Warning message in chisq.test(freq_TS[-c(drop)], p = freq_RA[-c(drop)], B =
1e+06):
```

```
"          "
```

```
Chi-squared test for given probabilities
```

```
data: freq_TS[-c(drop)]
```

```
X-squared = 2.1613, df = 5, p-value = 0.8264
```

```
[1] "badger"
```

```
Warning message in chisq.test(freq_TS, p = freq_RA, B = 1e+06):
```

```
"          "
```

```
Chi-squared test for given probabilities
```

```
data: freq_TS
```

```
X-squared = 10.459, df = 9, p-value = 0.3146
```

```
[1] "ground squirrel"
```

```
Warning message in chisq.test(freq_TS[-c(drop)], p = freq_RA[-c(drop)], B =
1e+06):
```

```
"          "
```

```
Chi-squared test for given probabilities
```

```
data: freq_TS[-c(drop)]
```

X-squared = 12.362, df = 5, p-value = 0.03015

[1] "hawk"

Warning message in chisq.test(freq\_TS, p = freq\_RA, B = 1e+06):  
" "

Chi-squared test for given probabilities

data: freq\_TS

X-squared = 7.2834, df = 9, p-value = 0.6076

[1] "mink"

Warning message in chisq.test(freq\_TS[-c(drop)], p = freq\_RA[-c(drop)], B = 1e+06):  
" "

Chi-squared test for given probabilities

data: freq\_TS[-c(drop)]

X-squared = 8.4219, df = 8, p-value = 0.3934

[1] "mouse"

Warning message in chisq.test(freq\_TS[-c(drop)], p = freq\_RA[-c(drop)], B = 1e+06):  
" "

Chi-squared test for given probabilities

data: freq\_TS[-c(drop)]

X-squared = 30.349, df = 8, p-value = 0.0001833

[1] "opossum"

Warning message in chisq.test(freq\_TS[-c(drop)], p = freq\_RA[-c(drop)], B = 1e+06):  
" "

Chi-squared test for given probabilities

data: freq\_TS[-c(drop)]

X-squared = 15.817, df = 8, p-value = 0.04507

[1] "owl"

```
Warning message in chisq.test(freq_TS, p = freq_RA, B = 1e+06):  
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS  
X-squared = 6.7772, df = 9, p-value = 0.6603
```

```
[1] "porcupine"
```

```
Warning message in chisq.test(freq_TS[-c(drop)], p = freq_RA[-c(drop)], B =  
1e+06):  
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS[-c(drop)]  
X-squared = 13.996, df = 8, p-value = 0.08187
```

```
[1] "rabbit"
```

```
Warning message in chisq.test(freq_TS, p = freq_RA, B = 1e+06):  
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS  
X-squared = 13.847, df = 9, p-value = 0.1279
```

```
[1] "raccoon"
```

```
Warning message in chisq.test(freq_TS[-c(drop)], p = freq_RA[-c(drop)], B =  
1e+06):  
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS[-c(drop)]  
X-squared = 5.338, df = 8, p-value = 0.7209
```

```
[1] "rat"
```

```
Warning message in chisq.test(freq_TS, p = freq_RA, B = 1e+06):  
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS
X-squared = 10.105, df = 9, p-value = 0.342
```

```
[1] "raven"
```

```
Warning message in chisq.test(freq_TS[-c(drop)], p = freq_RA[-c(drop)], B =
1e+06):
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS[-c(drop)]
X-squared = 42.418, df = 8, p-value = 1.13e-06
```

```
[1] "skunk"
```

```
Warning message in chisq.test(freq_TS[-c(drop)], p = freq_RA[-c(drop)], B =
1e+06):
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS[-c(drop)]
X-squared = 31.23, df = 7, p-value = 5.639e-05
```

```
[1] "spider"
```

```
Warning message in chisq.test(freq_TS, p = freq_RA, B = 1e+06):
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS
X-squared = 7.2243, df = 9, p-value = 0.6138
```

```
[1] "wren"
```

```
Warning message in chisq.test(freq_TS, p = freq_RA, B = 1e+06):
"          "
```

Chi-squared test for given probabilities

```
data: freq_TS
X-squared = 11.057, df = 9, p-value = 0.2718
```

```
1. 0.826407277491224 2. 0.547254771603764 3. 0.120586727741319 4. 0.75462742100524
5. 0.572183609206447 6. 0.000977835645210114 7. 0.144233655586388 8. 0.75462742100524
```

9. 0.21831921738013 10. 0.2923016655192 11. 0.768978825056247 12. 0.547254771603764  
13. 1.80855445880415e-05 14. 0.000451153640564889 15. 0.75462742100524 16. 0.543624840132482

We can see thee species (mouse, ravem and skunk) that differ in distributions between real animals and tricksters

[ ]: