

# Ordinal logistic model on large, classified windows data

Ruth Gómez Graciani

## Contents

<b>Prepare the data</b>	<b>1</b>
<b>Descriptive statistics</b>	<b>5</b>
Variable scalation (optional) . . . . .	10
<b>Basic inversions model</b>	<b>10</b>
<b>All dataset together without colors category</b>	<b>12</b>
<b>Centromeres vs telomeres models</b>	<b>13</b>
Centromeres & arms . . . . .	13
Arms . . . . .	14
Telomeres . . . . .	15
<b>Without some values</b>	<b>15</b>
Without 3+ . . . . .	15
Without 2 . . . . .	16
Without 1 . . . . .	17
Without 0 . . . . .	18
<b>Bingo! 1 categories are doing something...</b>	<b>19</b>
Without per-region (Color) outliers . . . . .	22
Without per-category (InvCategory) outliers . . . . .	25
Without all outliers . . . . .	27
Manually remove group 1 outlier . . . . .	28
<b>Which is this point?</b>	<b>29</b>
<b>Wihtout some chromosomes</b>	<b>32</b>
<b>Multiply the numbers</b>	<b>33</b>
cM/Mb *10 . . . . .	33
cM/Mb *100 . . . . .	34
cM/Mb *1000 . . . . .	35
cM/Mb *10000 . . . . .	36

## Prepare the data

First, we obtain the density distribution, and local minima and maxima for the recombination map.

# CEUSpence\_COzones\_0.03\_1000000

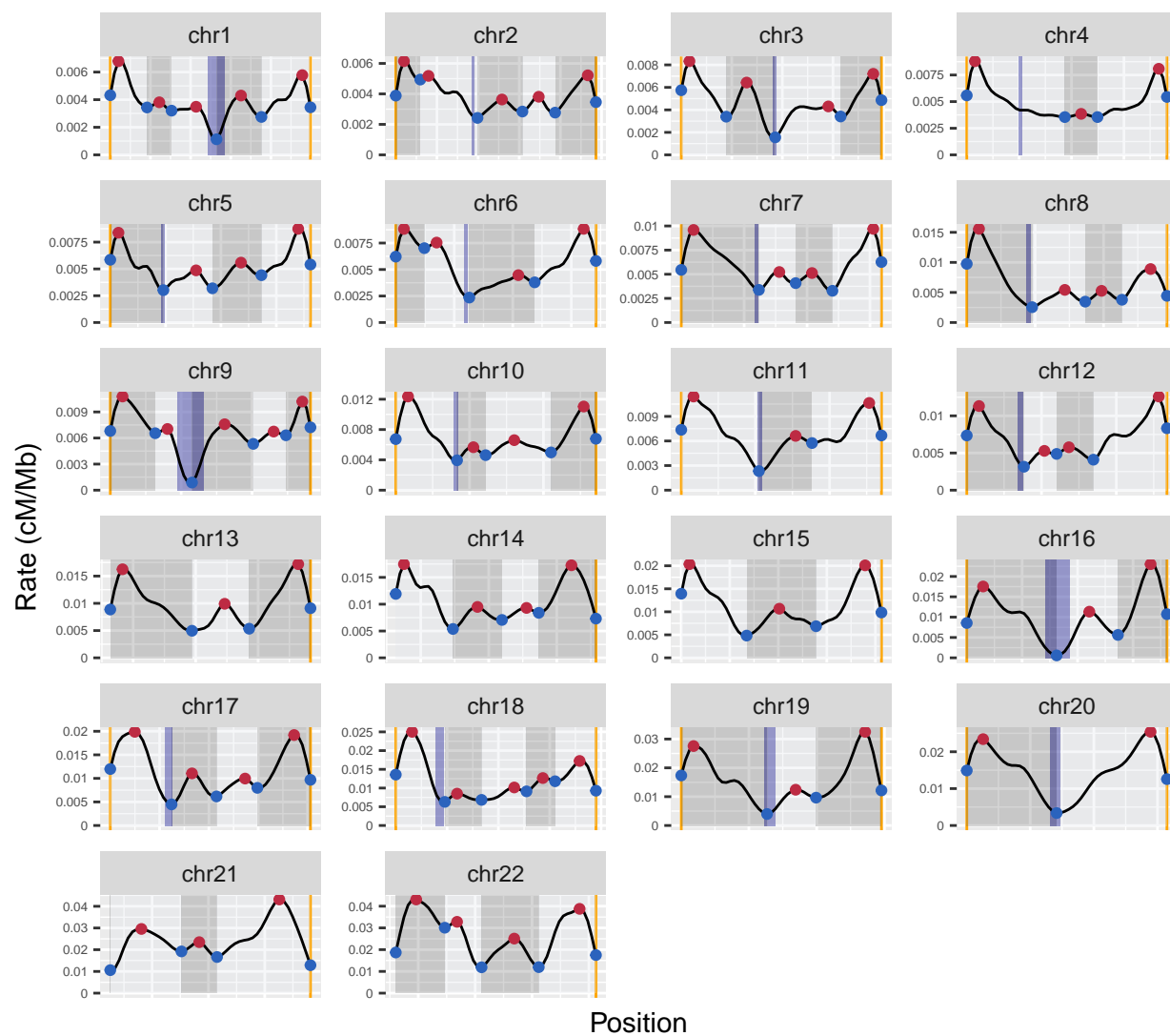


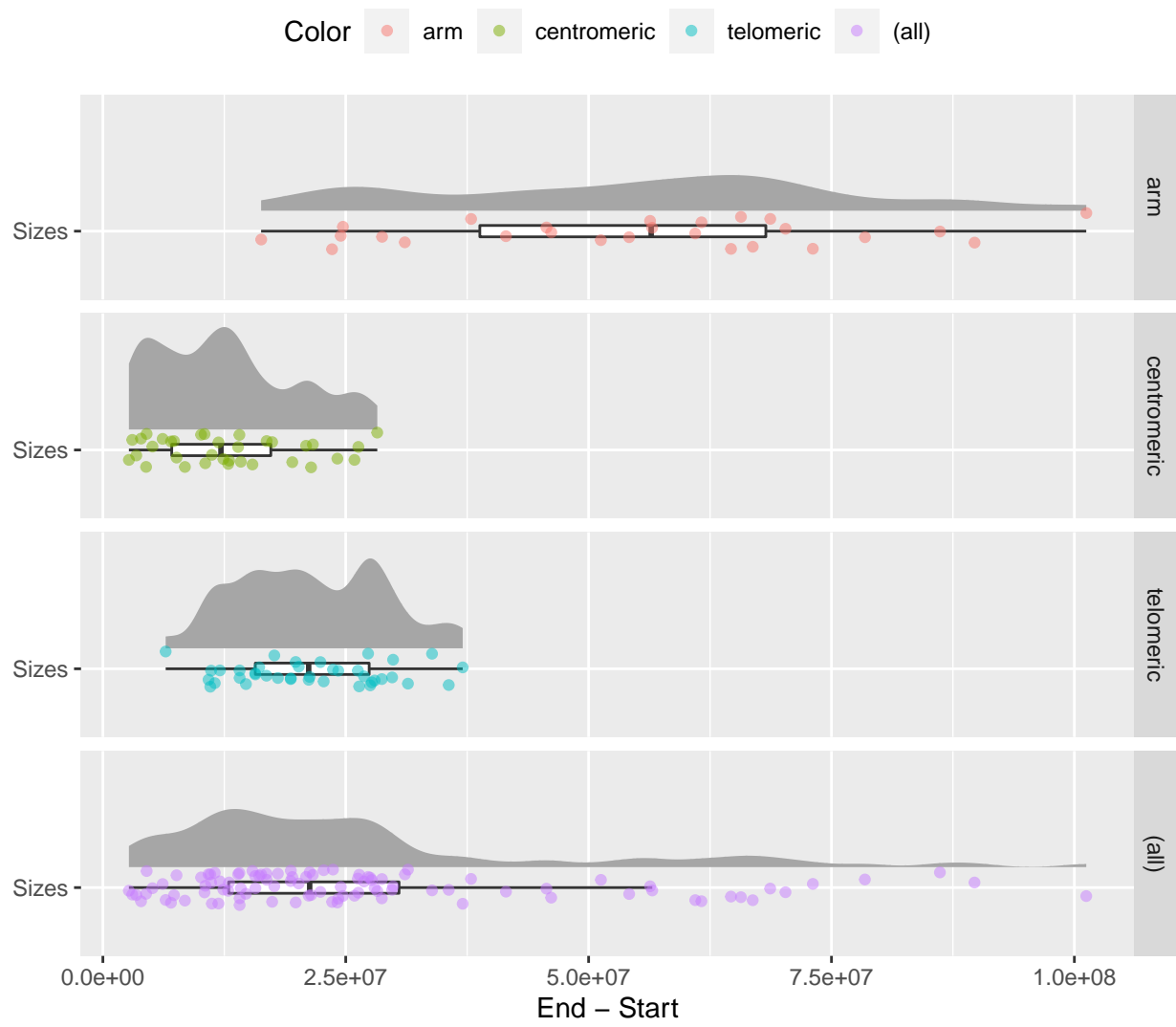
Figure 1: Crossover zones; centromeres in blue, workspace limits in orange.

Next, we define telomeric regions as the space between the chromosome start to the next local minimum, or between the chromosome end to the previous local minimum. We also define centromeric regions as the space between two local maxima that contains the centromere. When the local maximum delimiting a centromeric region is the same as the peak from the corresponding telomeric region (see chr1, chr5, chr7, chr8, etc.), the limit between the telomeric and centromeric regions is defined as the center point between the local maximum corresponding to the telomeric peak and the local minimum corresponding to the centromere valley. These categories will be represented as the “Color” variable in this analysis.

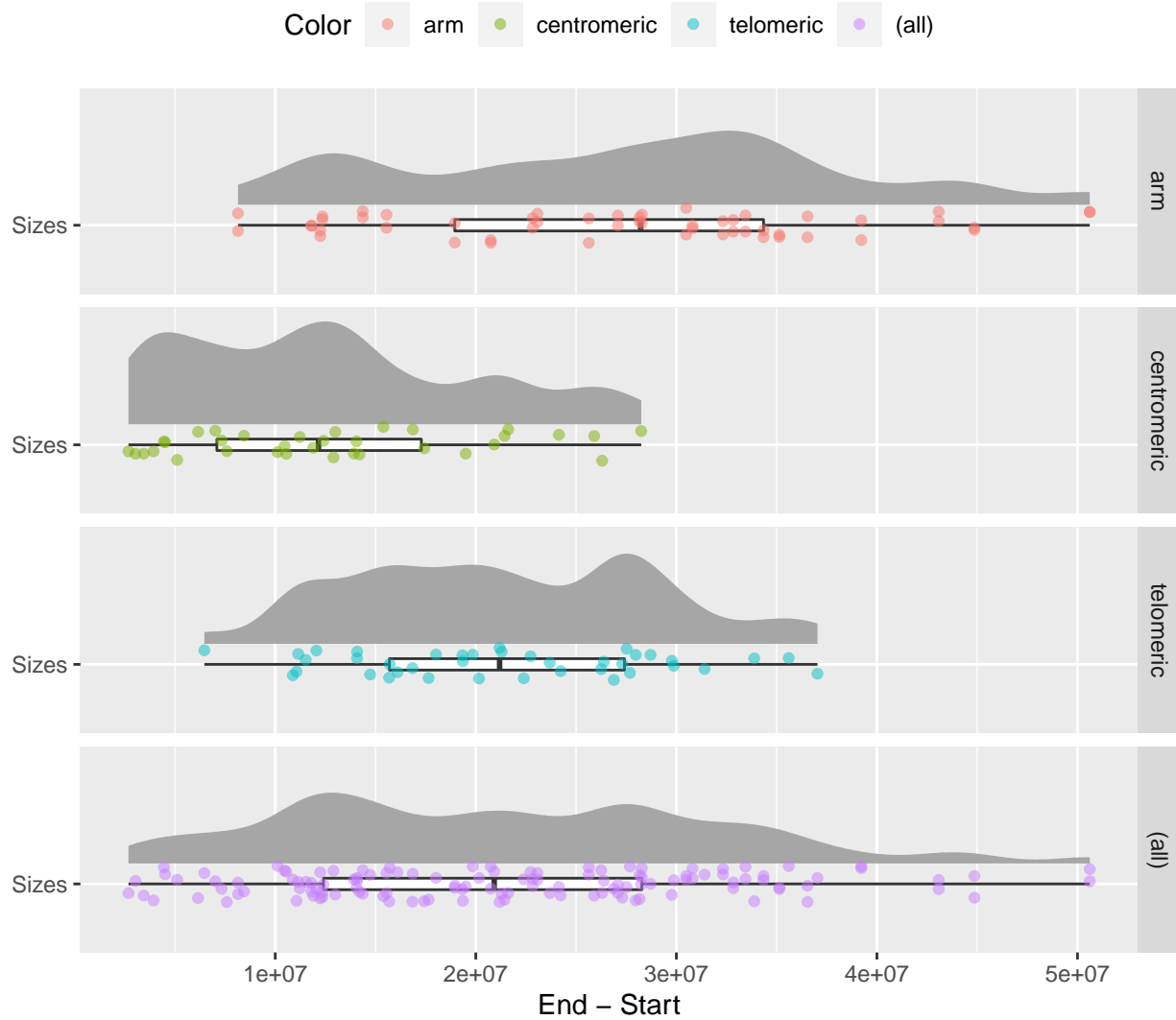


Figure 2: Color-coded windows for telomeric, centromeric and arm categories.

## Distribution



## Distribution



## Descriptive statistics

Raw data:

Chromosome	Start	End	Color	invCenters	NHCenters	NAHRCenters	Length.Mb	AllRepCounts	WAvgRate.cM/Mb	GenomType
chr10	60683	23750219	telomeric	3	2	1	23.689536	428	0.0144805	A
chr10	23750219	39146059	centromeric	0	0	1	15.395840	880	0.0079079	A
chr10	116172416	35524372	telomeric	1	1	0	19.351956	266	0.0140434	A
chr10	42369508	47478464	centromeric	0	0	0	5.108956	954	0.0068103	A
chr11	87267	29960849	telomeric	2	1	1	29.873582	920	0.0119573	A
chr11	29960849	51577746	centromeric	0	0	0	21.616897	596	0.0060984	A

For each window, I calculated the number of total inversions, NH inversions, and NAHR inversions, the window length in Mb, number of repeats and the average recombination rate in cM/Mb.

I want to perform Ordinal Logistic Regressions on different subsets of the data. The assumptions of the Ordinal Logistic Regression are as follow:

1. The dependent variable is ordered.
2. One or more of the independent variables are either continuous, categorical or ordinal.
3. No multi-collinearity.
4. Proportional odds.

I show the data distributions in the figure below. The inversion counts have only a number of possible options, so they can be considered an ordinal variable. The independent variables are continuous and categorical, so assumptions 1 and 2 are satisfied

## Distribution of variables

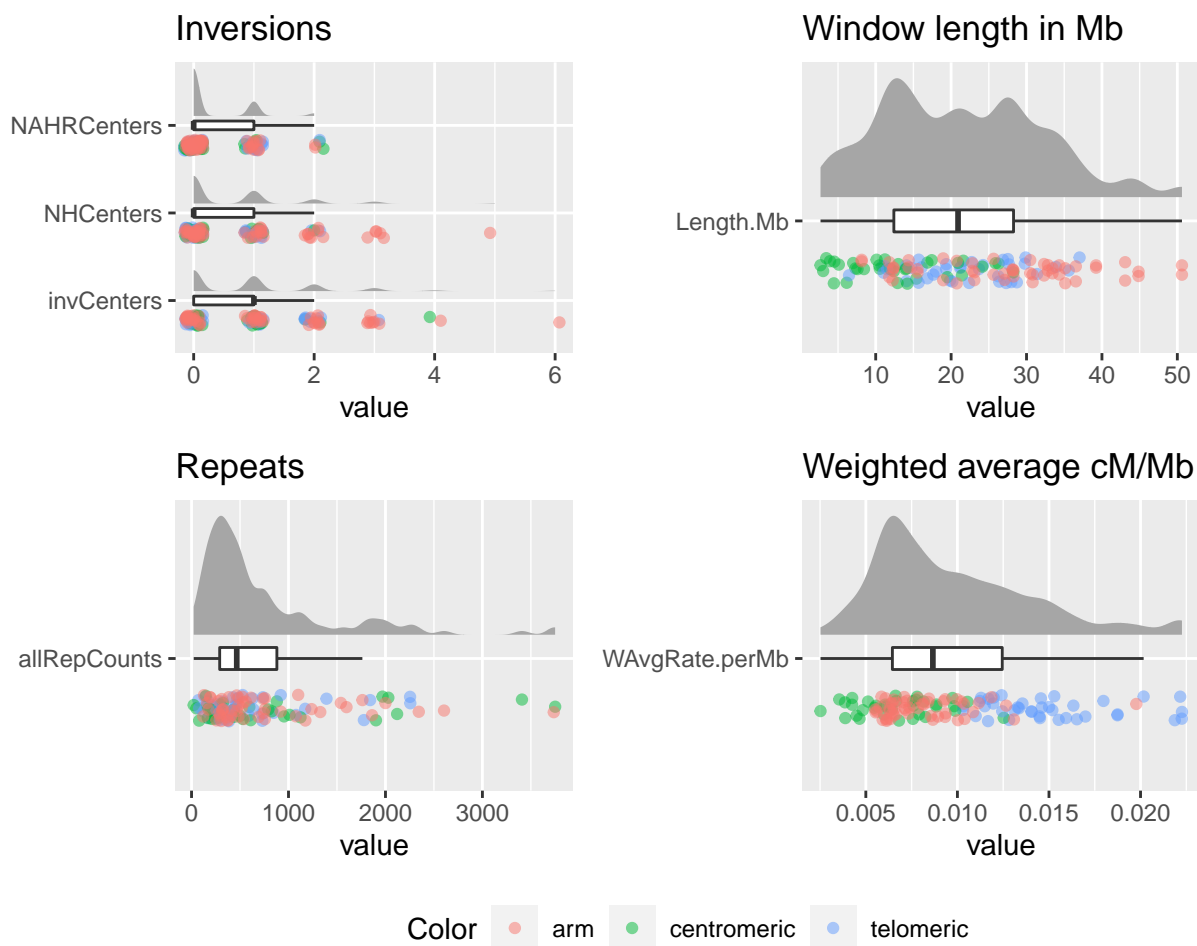


Figure 3: Distribution of variables.

We see that some categories have low number of cases, so I will make a “3 or more” category when relevant.

Table 2: Original counts

CountGroups	invCenters	NHCenters	NAHRCenters
0	56	74	92
1	40	34	28
2	18	11	5

CountGroups	invCenters	NHCenters	NAHRCenters
3	8	5	NA
4	2	NA	NA
5	NA	1	NA
6	1	NA	NA

Table 3: New counts

CountGroups	invCategory	NHCategory	NAHRCategory
0	56	74	92
1	40	34	28
2	18	11	5
3+	11	6	NA

With these groups, I visualize the relationships between dependent and independent variables.

## Differences in each chromosomal variable between inversion count groups

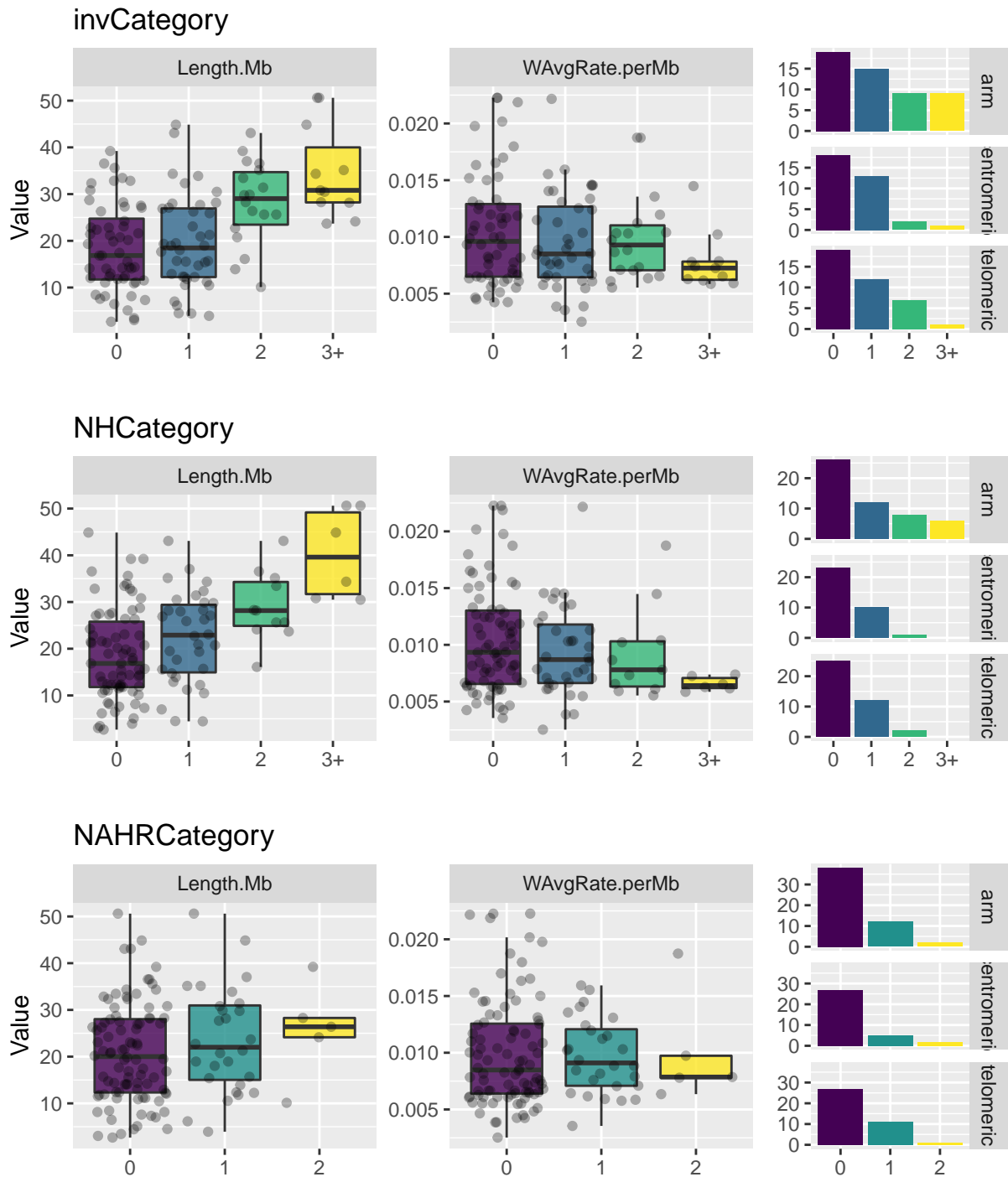
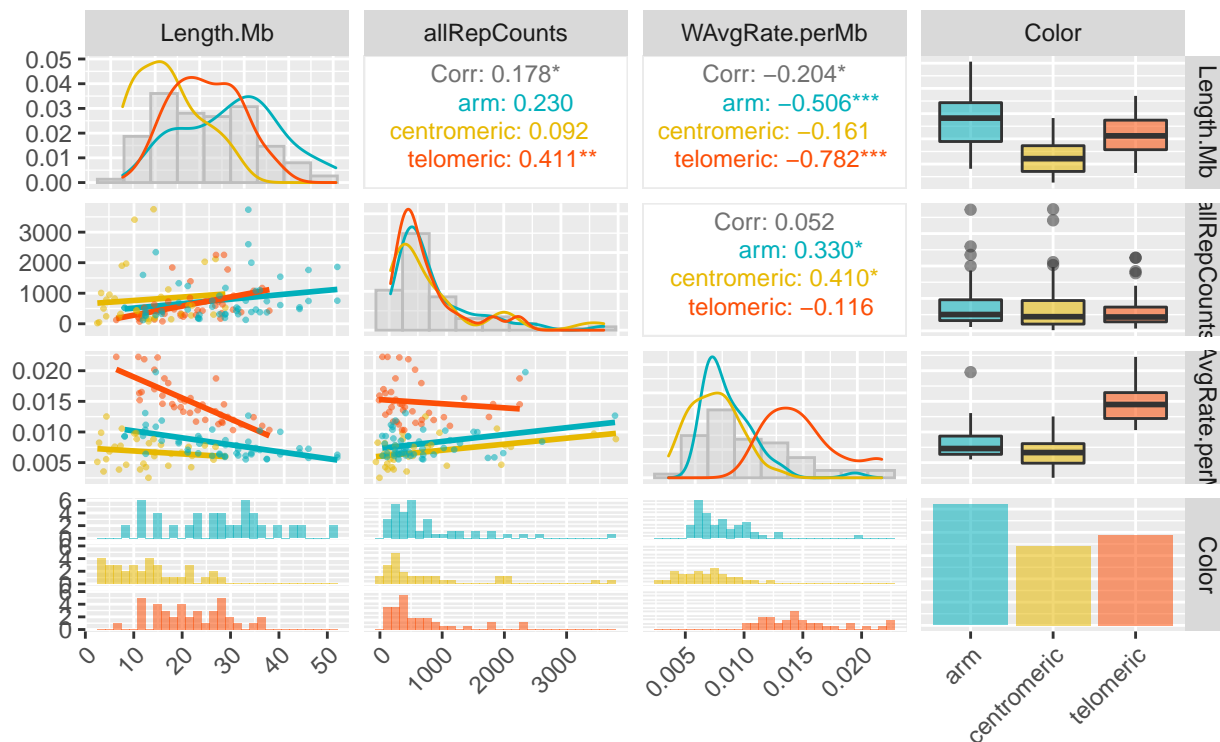


Figure 4: Potential effect of independent variables on the different types of inversions.



Finally, I will test assumption number 3, no multi-collinearity between independent variables.

### Pearson correlation



### Spearman correlation

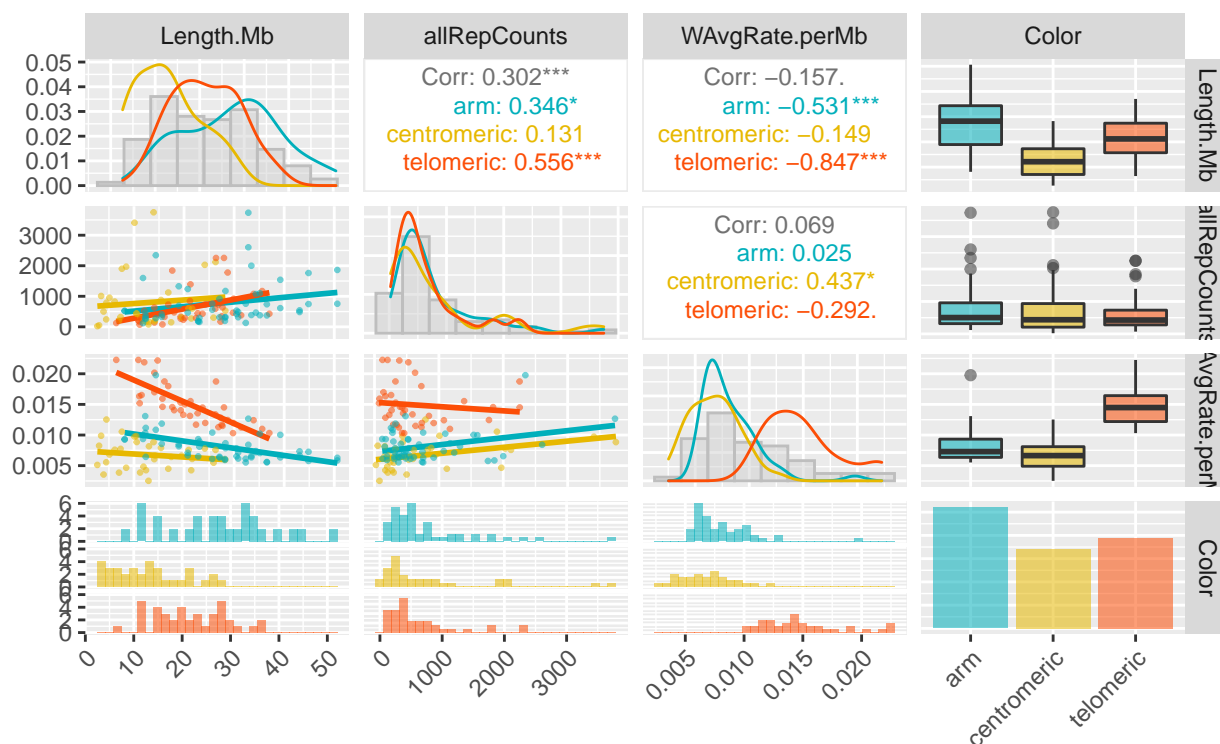


Figure 5: Correlations between variables.

We see that our three variables are significantly correlated, but this does not confirm multi-collinearity. I perform a variance inflation factor test on the corresponding linear model to further check the multi-collinearity.

	GVIF	Df	GVIF^(1/(2*Df))
Length.Mb	2.272644	1	1.507529
allRepCounts	1.232203	1	1.110046
Color	5.610644	2	1.539052
WAvgRate.perMb	4.011360	1	2.002838

The general rule of thumbs for VIF test is that if the VIF value is greater than 10, then there is multi-collinearity, so we can say that the third assumption (no multi-collinearity) is satisfied.

The proportional odds assumption will be tested for each model that we fit in the following analyses.

## Variable scalation (optional)

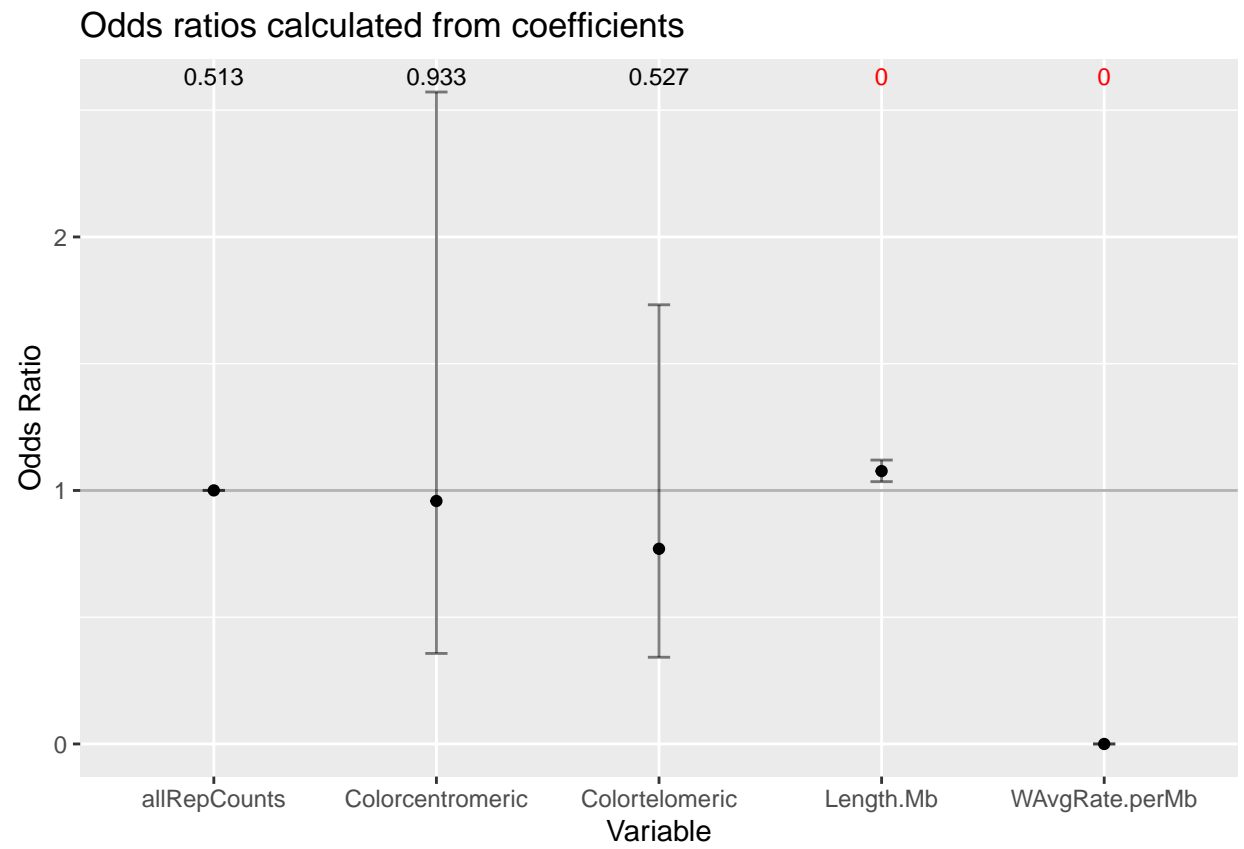
Standardized coefficients are useful in our case to compare effects of predictors reported in different units. The most straightforward way is using the Agresti method of standardization, applied with the `scale()` function.

	Length.Mb	Length.Mb.Scaled	allRepCounts	allRepCounts.Scaled	WAvgRate.perMb	WAvgRate.perMb.Scaled
Min.	2.680877	-1.7498964	22.000	-0.9723187	0.0025248	-1.6620166
1st Qu.	12.405960	-0.8498356	290.000	-0.6083294	0.0064492	-0.7741887
Median	20.918330	-0.0620119	466.000	-0.3692916	0.0086610	-0.2738265
Mean	21.588363	0.0000000	737.904	0.0000000	0.0098713	0.0000000
3rd Qu.	28.276348	0.6189760	880.000	0.1929904	0.0124489	0.5831374
Max.	50.611965	2.6861476	3750.000	4.0909359	0.0222621	2.8031981

Once the model is fitted, we can use the `sd` to transform scaled coefficients to natural coefficients and viceversa.

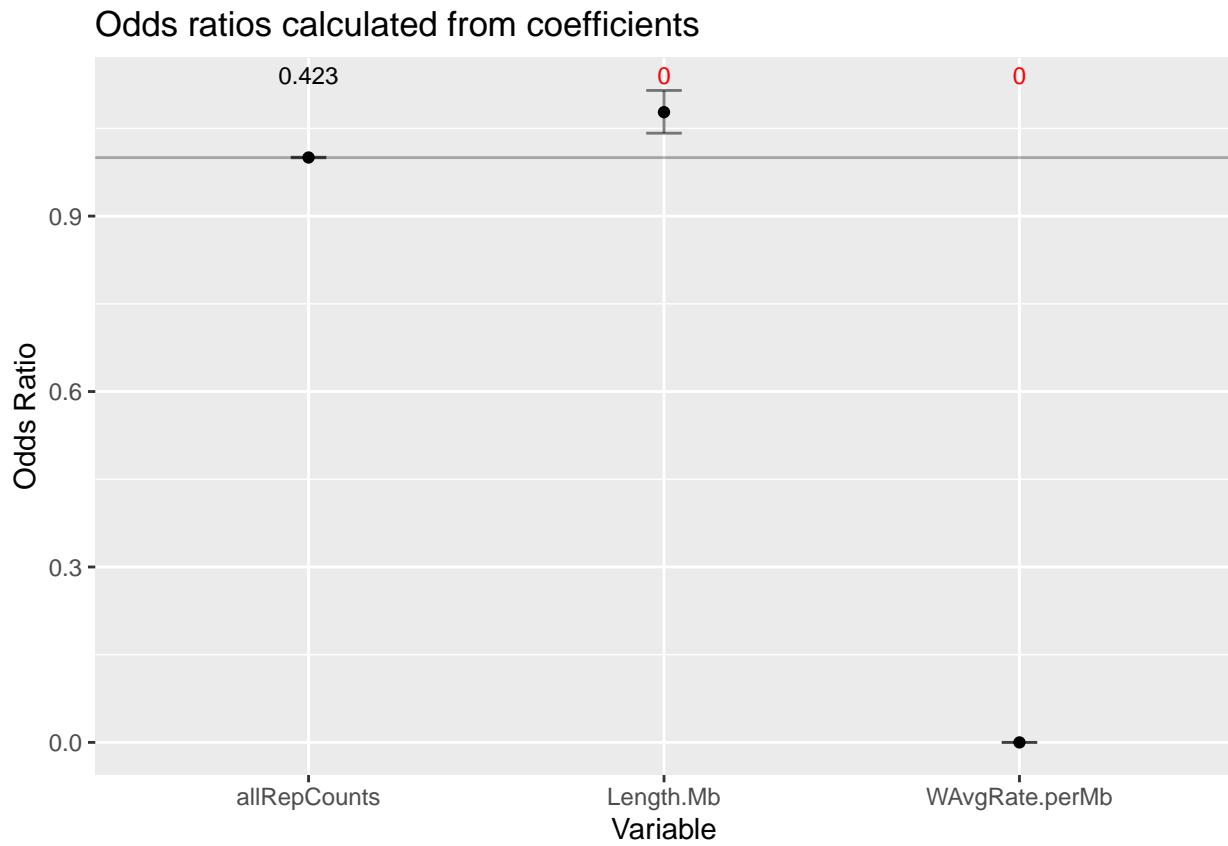
## Basic inversions model

	Value	Std. Error	t value	p value
Length.Mb	0.0737271	0.0201554	3.6579369	0.0002542
allRepCounts	0.0001784	0.0002728	0.6539988	0.5131126
Colorcentromeric	-0.0424398	0.5035783	-0.0842765	0.9328366
Colortelomeric	-0.2615772	0.4138051	-0.6321266	0.5273042
WAvgRate.perMb	-20.4600426	0.0075202	-2720.6640308	0.0000000
0 1	1.0831908	0.5879727	1.8422466	0.0654391
1 2	2.7165527	0.6382078	4.2565330	0.0000208
2 3+	4.0236666	0.7141154	5.6344767	0.0000000



## All dataset together without colors category

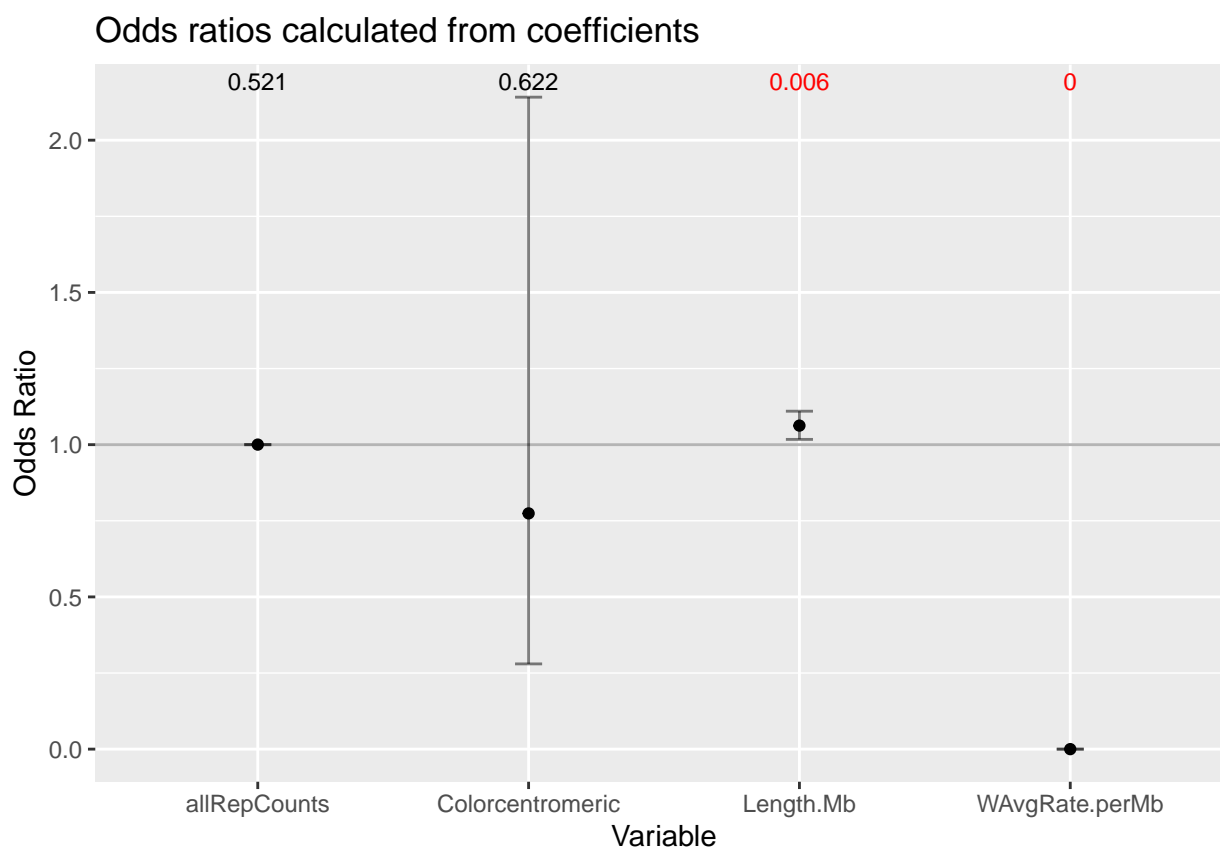
	Value	Std. Error	t value	p value
Length.Mb	0.0748453	0.0173033	4.3254994	0.0000152
allRepCounts	0.0002178	0.0002716	0.8018725	0.4226267
WAvgRate.perMb	-42.4941805	0.0044931	-9457.7004110	0.0000000
0 1	1.0247472	0.4191767	2.4446666	0.0144986
1 2	2.6632238	0.4914423	5.4191992	0.0000001
2 3+	4.0057595	0.5917357	6.7695084	0.0000000



## Centromeres vs telomeres models

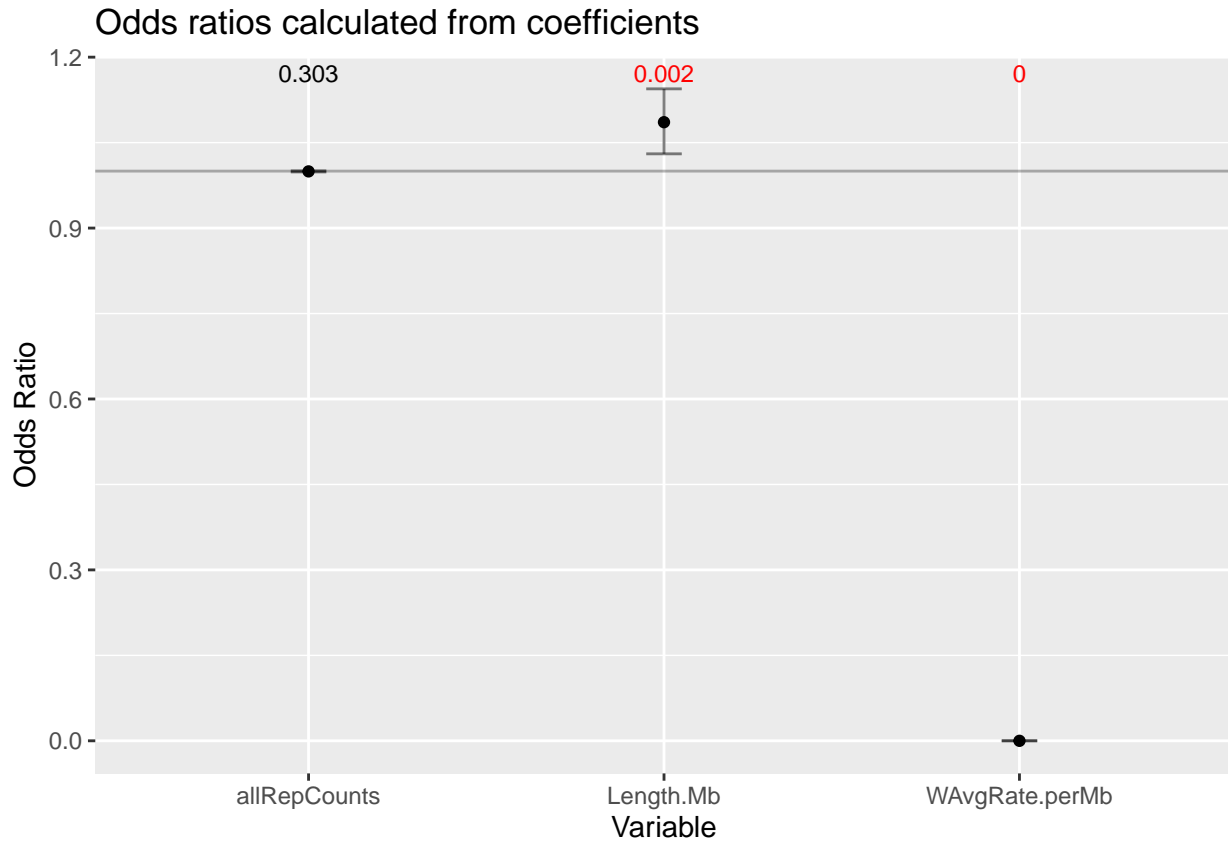
### Centromeres & arms

	Value	Std. Error	t value	p value
Length.Mb	0.0607367	0.0222281	2.7324270	0.0062870
allRepCounts	0.0001975	0.0003074	0.6424220	0.5205992
Colorcentromeric	-0.2558071	0.5190072	-0.4928779	0.6220989
WAvgRate.perMb	-70.9254147	0.0078320	-9055.9069961	0.0000000
0 1	0.4021982	0.6483600	0.6203316	0.5350395
1 2	2.0349075	0.7057819	2.8831958	0.0039366
2 3+	3.0992051	0.7809907	3.9682998	0.0000724



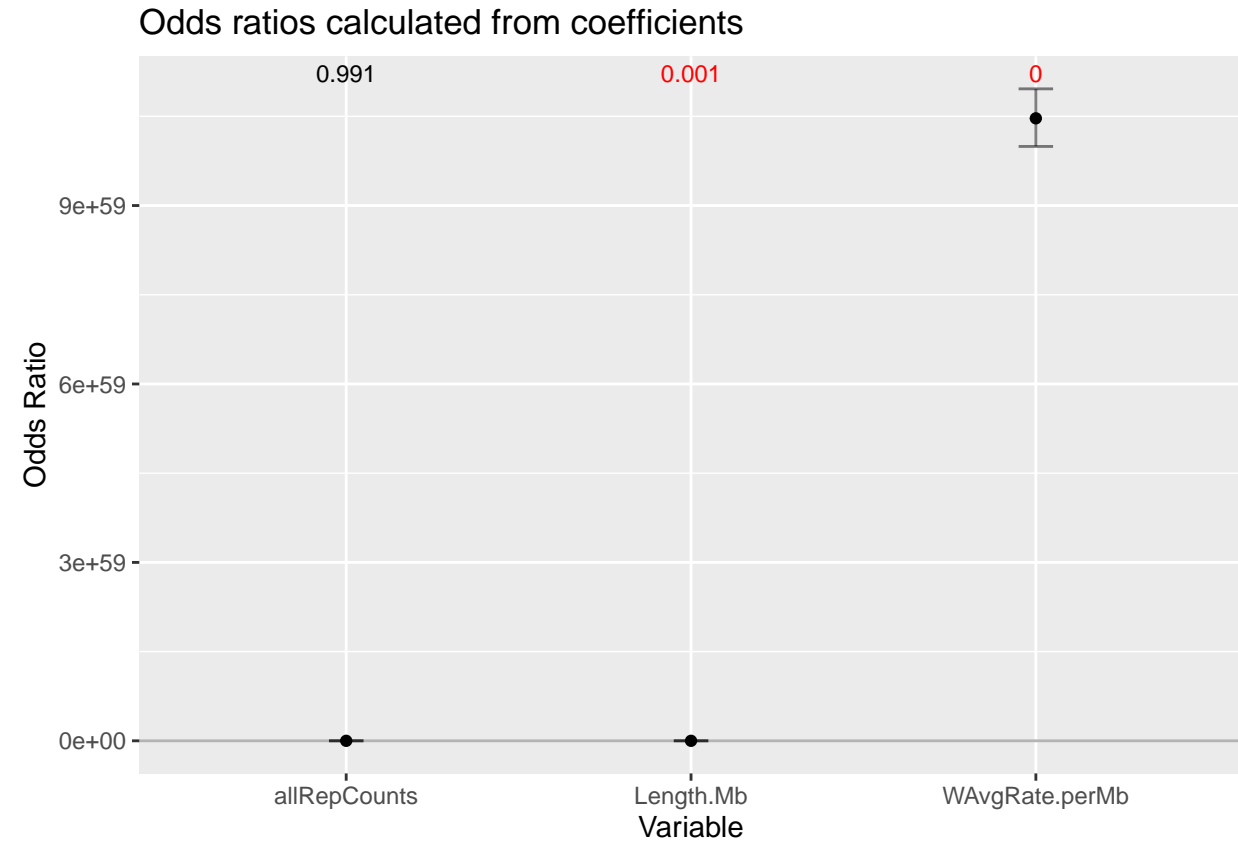
## Arms

	Value	Std. Error	t value	p value
Length.Mb	0.0823750	0.0267929	3.074511e+00	0.0021085
allRepCounts	-0.0004984	0.0004835	-1.030656e+00	0.3027021
WAvgRate.perMb	-85.7120834	0.0076066	-1.126809e+04	0.0000000
0 1	0.4604720	0.7357741	6.258334e-01	0.5314242
1 2	1.9100213	0.8097604	2.358749e+00	0.0183367
2 3+	3.0216266	0.8984374	3.363202e+00	0.0007704



## Telomeres

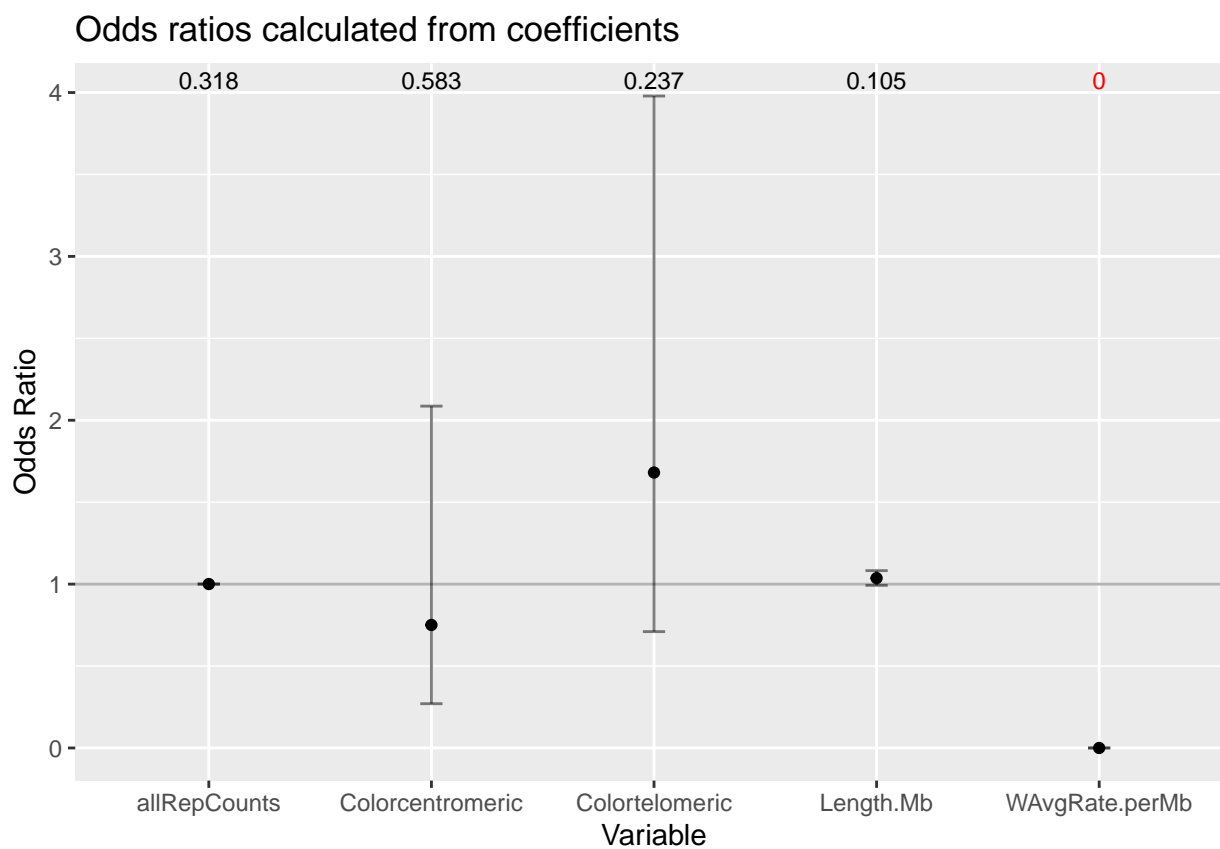
	Value	Std. Error	t value	p value
Length.Mb	0.1684136	0.0488775	3.4456263	0.0005697
allRepCounts	0.0000070	0.0006143	0.0114539	0.9908613
WAvgRate.perMb	138.2007194	0.0236048	5854.7706739	0.0000000
0 1	5.5689339	1.0379514	5.3653126	0.0000001
1 2	7.2537166	1.1844030	6.1243652	0.0000000
2 3+	9.7362012	1.5787288	6.1671143	0.0000000



## Without some values

### Without 3+

	Value	Std. Error	t value	p value
Length.Mb	0.0357966	0.0220799	1.621229e+00	0.1049686
allRepCounts	0.0003083	0.0003091	9.975525e-01	0.3184964
Colorcentromeric	-0.2866251	0.5214320	-5.496885e-01	0.5825331
Colortelomeric	0.5193225	0.4396308	1.181270e+00	0.2374956
WAvgRate.perMb	-92.7126645	0.0079481	-1.166482e+04	0.0000000
0 1	0.0287219	0.6213614	4.622420e-02	0.9631316
1 2	1.8601967	0.6788796	2.740098e+00	0.0061421

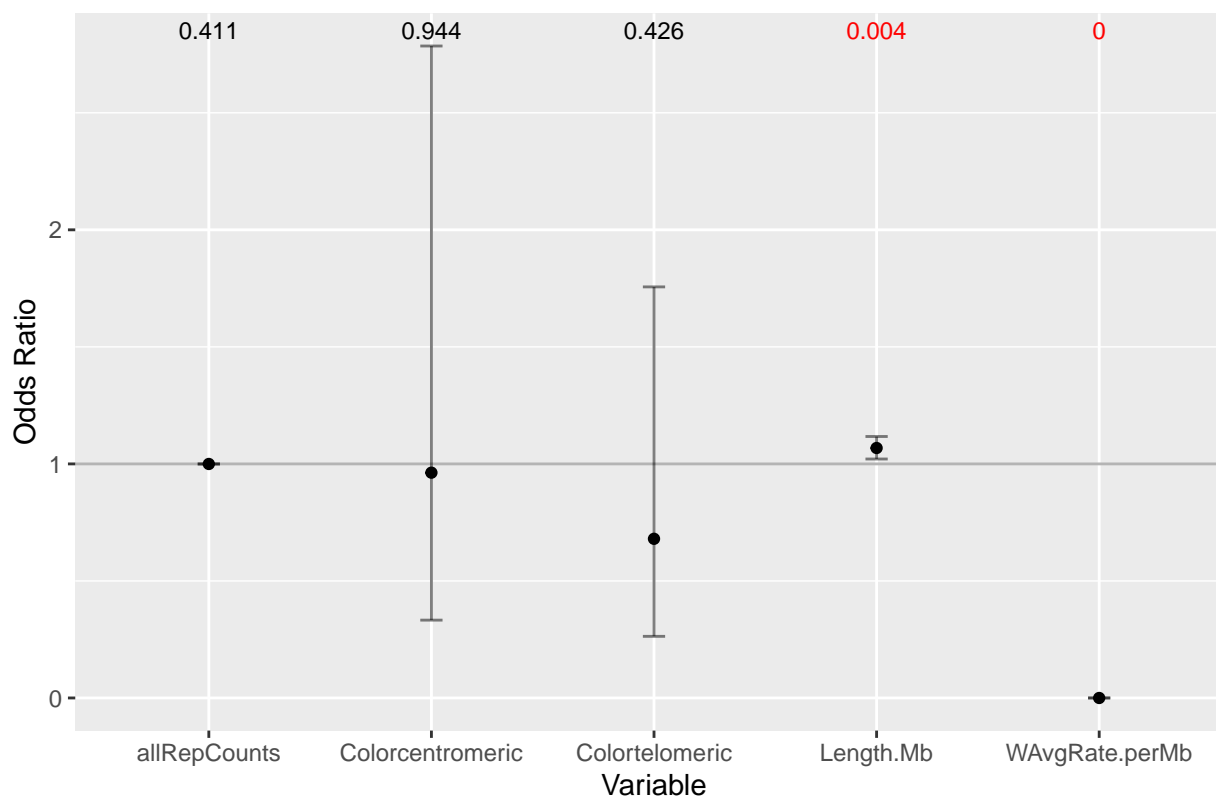


## Without 2

	Value	Std. Error	t value	p value
Length.Mb	0.0657398	0.0229874	2.8598215	0.0042388
allRepCounts	-0.0003079	0.0003747	-0.8217789	0.4112028
Colorcentromeric	-0.0381814	0.5421341	-0.0704280	0.9438530
Colortelomeric	-0.3856099	0.4840778	-0.7965865	0.4256912
WAvgRate.perMb	-29.3894076	0.0078837	-3727.8528784	0.0000000
0 1	0.7605662	0.6374277	1.1931803	0.2327988
1 3+	3.0714713	0.7312693	4.2001920	0.0000267

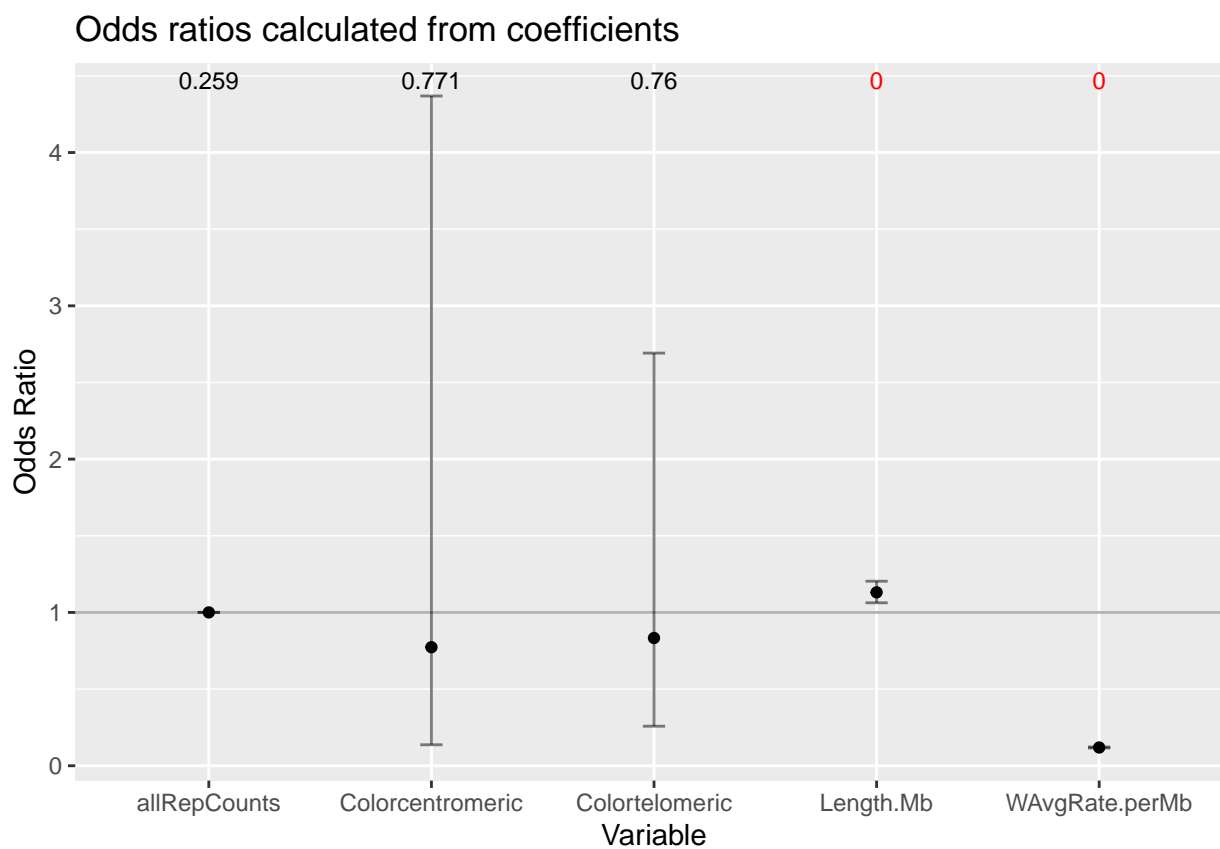


### Odds ratios calculated from coefficients



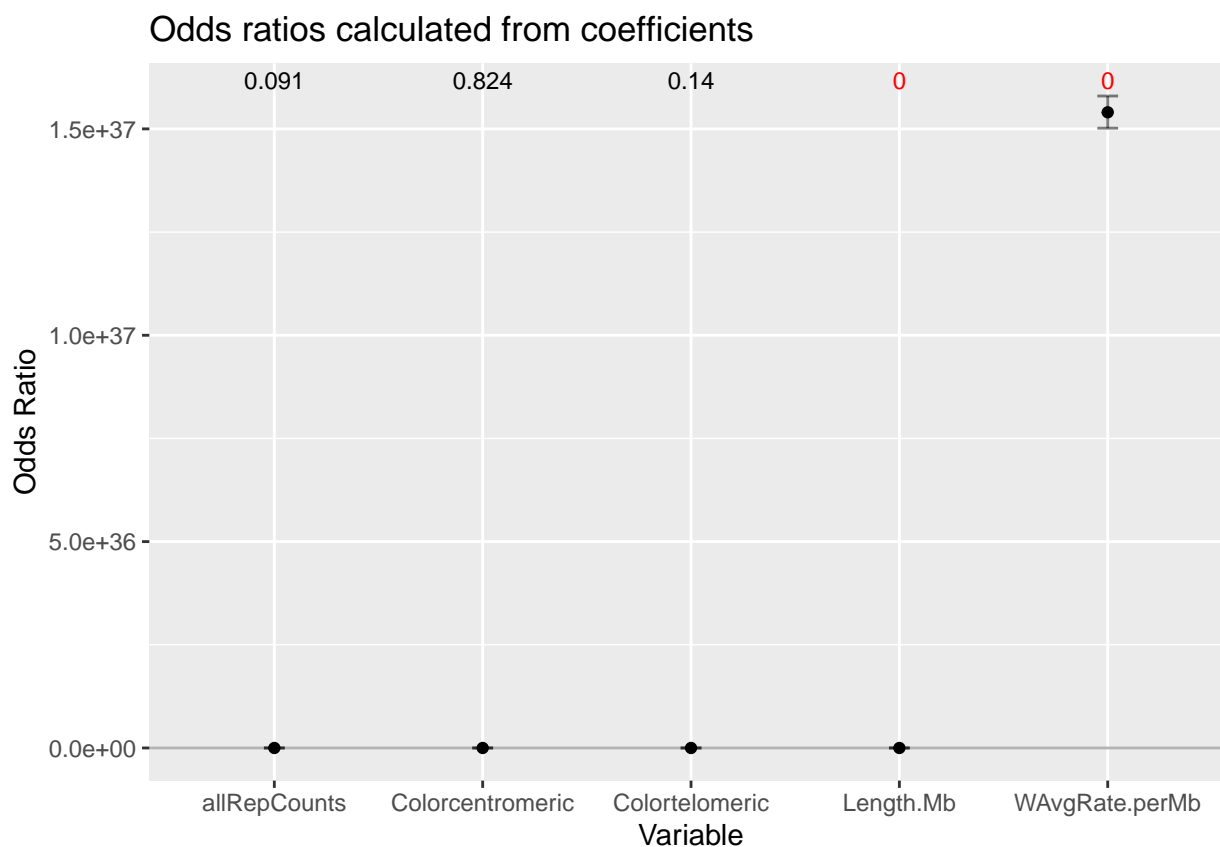
### Without 1

	Value	Std. Error	t value	p value
Length.Mb	0.1232740	0.0317454	3.8832083	0.0001031
allRepCounts	0.0003903	0.0003458	1.1285186	0.2591010
Colorcentromeric	-0.2570092	0.8834121	-0.2909278	0.7711065
Colortelomeric	-0.1823910	0.5982558	-0.3048713	0.7604642
WAvRate.perMb	-2.1272076	0.0137193	-155.0526408	0.0000000
0 2	3.8626769	1.0597724	3.6448175	0.0002676
2 3+	5.5240181	1.1521950	4.7943431	0.0000016



## Without 0

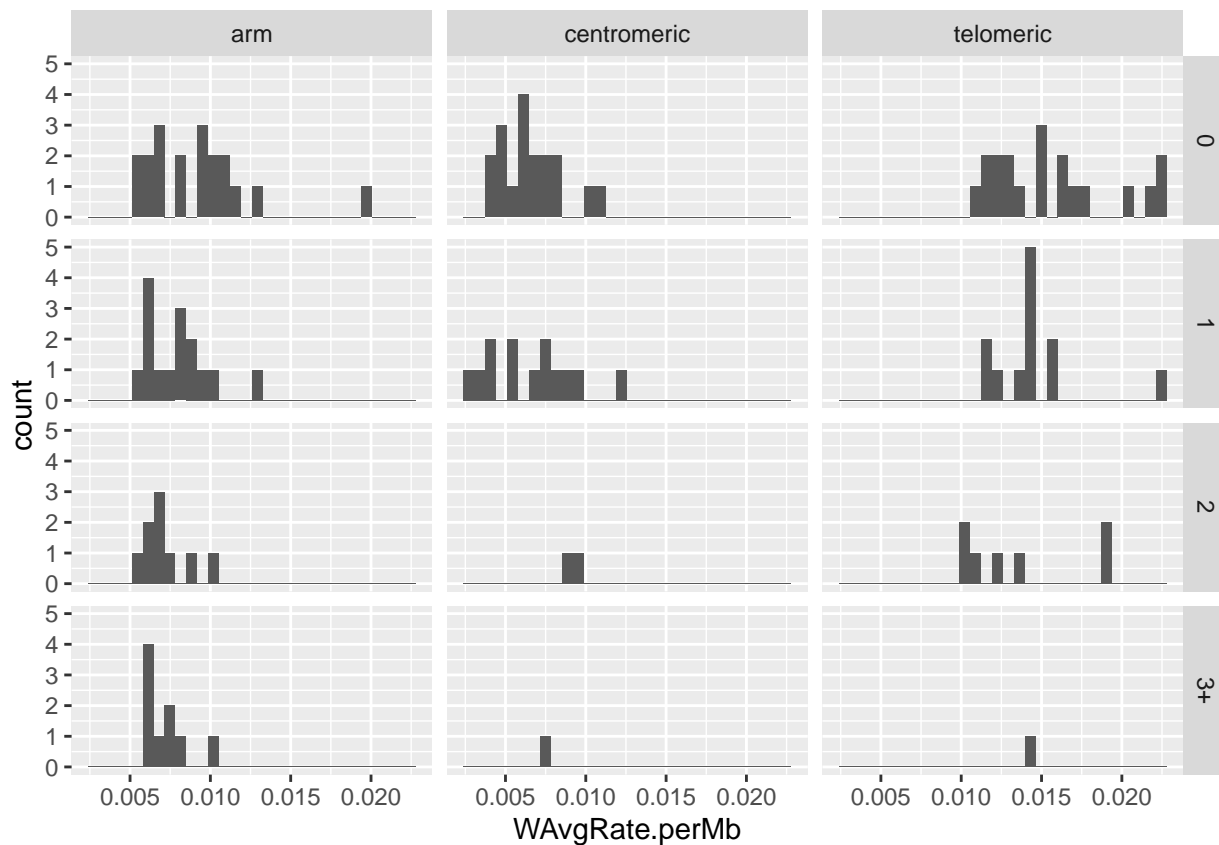
	Value	Std. Error	t value	p value
Length.Mb	0.1242950	0.0342084	3.6334649	0.0002796
allRepCounts	0.0007091	0.0004198	1.6890466	0.0912105
Colorcentromeric	-0.2214517	0.9960831	-0.2223225	0.8240628
Colortelomeric	-0.9174297	0.6209592	-1.4774395	0.1395579
WAvRate.perMb	85.6275888	0.0129082	6633.5801778	0.0000000
1 2	4.4949442	1.1303102	3.9767350	0.0000699
2 3+	6.3083182	1.2421986	5.0783490	0.0000004



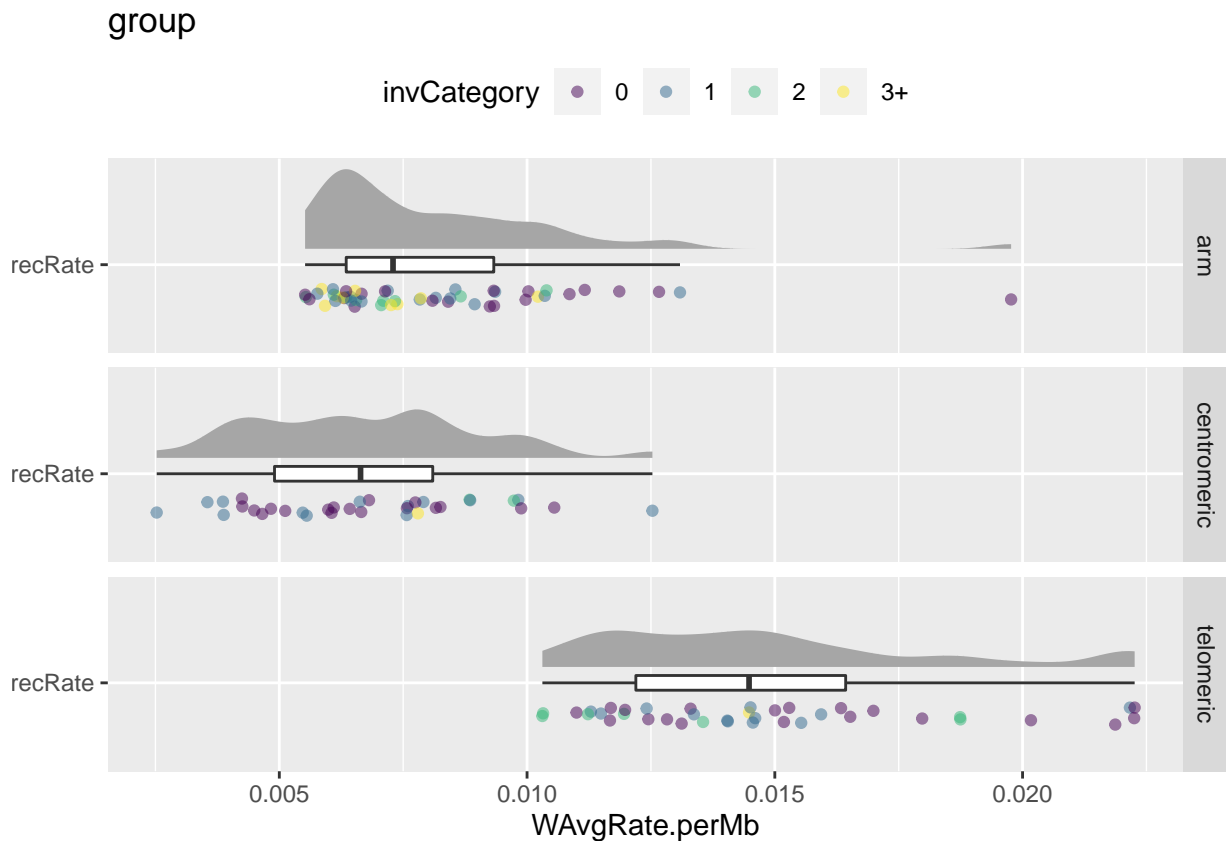
**Bingo! 1 categories are doing something...**

```
load(file= "data/winRegions.RData")

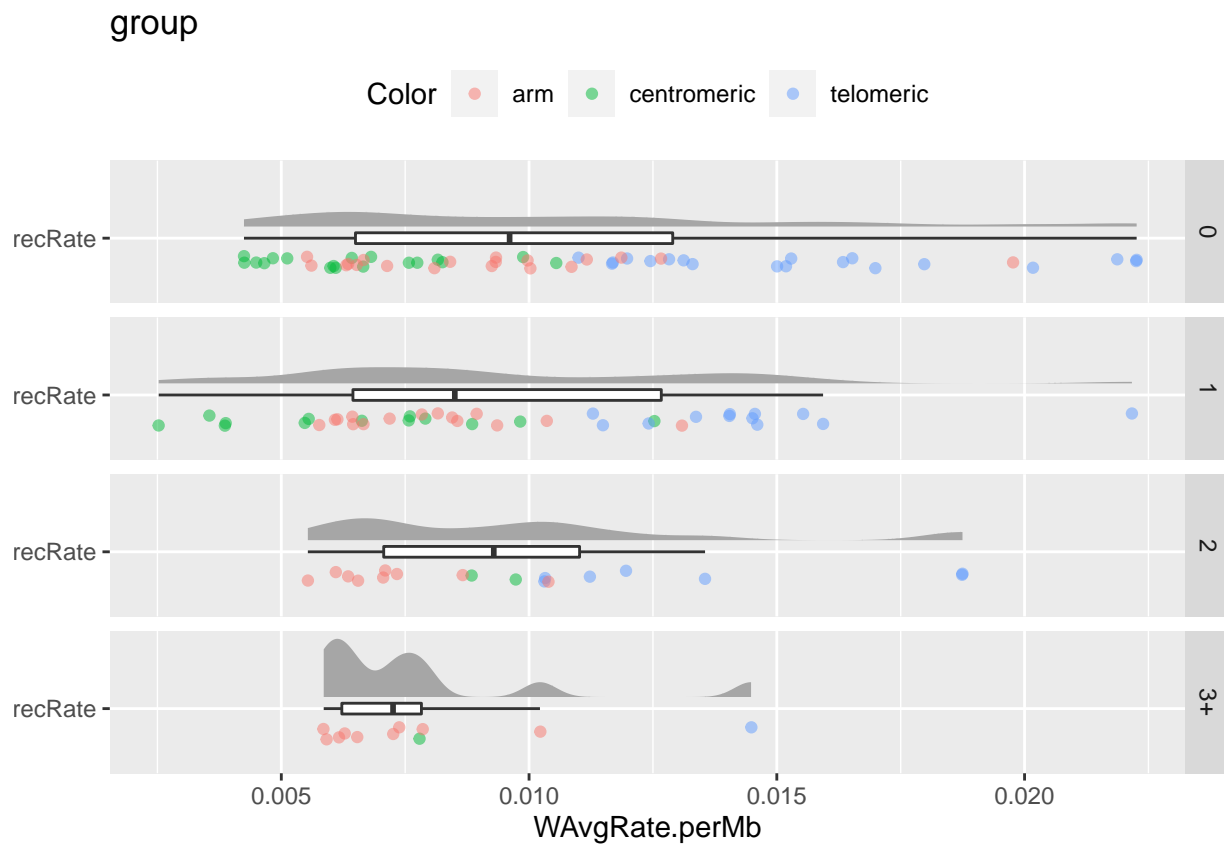
ggplot(winRegions)+geom_histogram(aes(x = WAvRate.perMb))+
  facet_grid(invCategory ~ Color)
```



```
ggplot(winRegions, aes(x = "recRate", y = WAvgRate.perMb))+
  # Half violin
  ggdist::stat_halfeye(adjust = .5, width = .6, .width = 0, justification = -.2, point_colour = NA)
  # Boxplot
  geom_boxplot(width = .1, outlier.shape = NA) +
  # Points
  gghalves::geom_half_point_panel(side = "l", range_scale = .6, alpha = .5, aes(color = invCategory))
  # scale_color_manual(values = c(rep("#3c7ae7",11),rep("#89b23e",11) ))+
  # Adjust coordinates
  coord_flip()+facet_grid(Color ~ .)+
  # coord_flip( xlim = c(1.3, NA))+
  # Adjust labels
  theme(axis.title.y = element_blank(), legend.position = "top")+
  # Title
  ggtitle("group")
```



```
ggplot(winRegions, aes(x = "recRate", y = WAvgRate.perMb))+
  # Half violin
  ggdist::stat_halfeye(adjust = .5, width = .6, .width = 0, justification = -.2, point_colour = NA)
  # Boxplot
  geom_boxplot(width = .1, outlier.shape = NA) +
  # Points
  gghalves::geom_half_point_panel(side = "l", range_scale = .6, alpha = .5, aes(color = Color))+
  # scale_color_manual(values = c(rep("#3c7ae7",11),rep("#89b23e",11) ))+
  # Adjust coordinates
  coord_flip()+facet_grid(invCategory ~ .)+
  # coord_flip( xlim = c(1.3, NA))+
  # Adjust labels
  theme(axis.title.y = element_blank(), legend.position = "top")+
  # Title
  ggtitle("group")
```

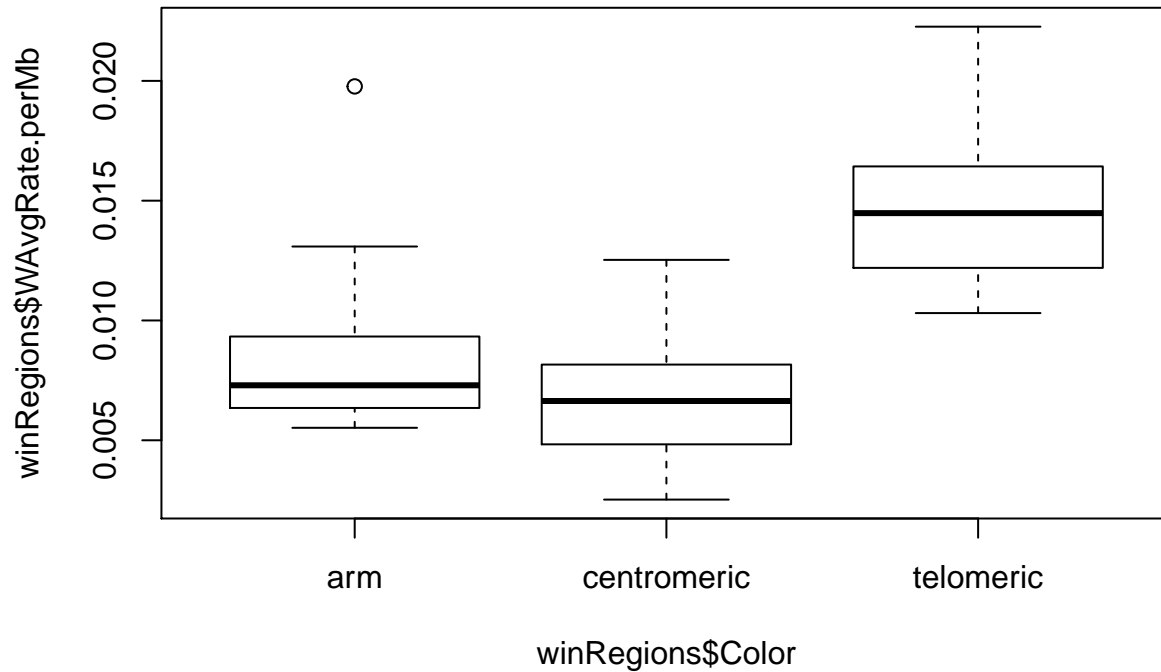


I will repeat the analysis without outliers to look for influential points.

### Without per-region (Color) outliers

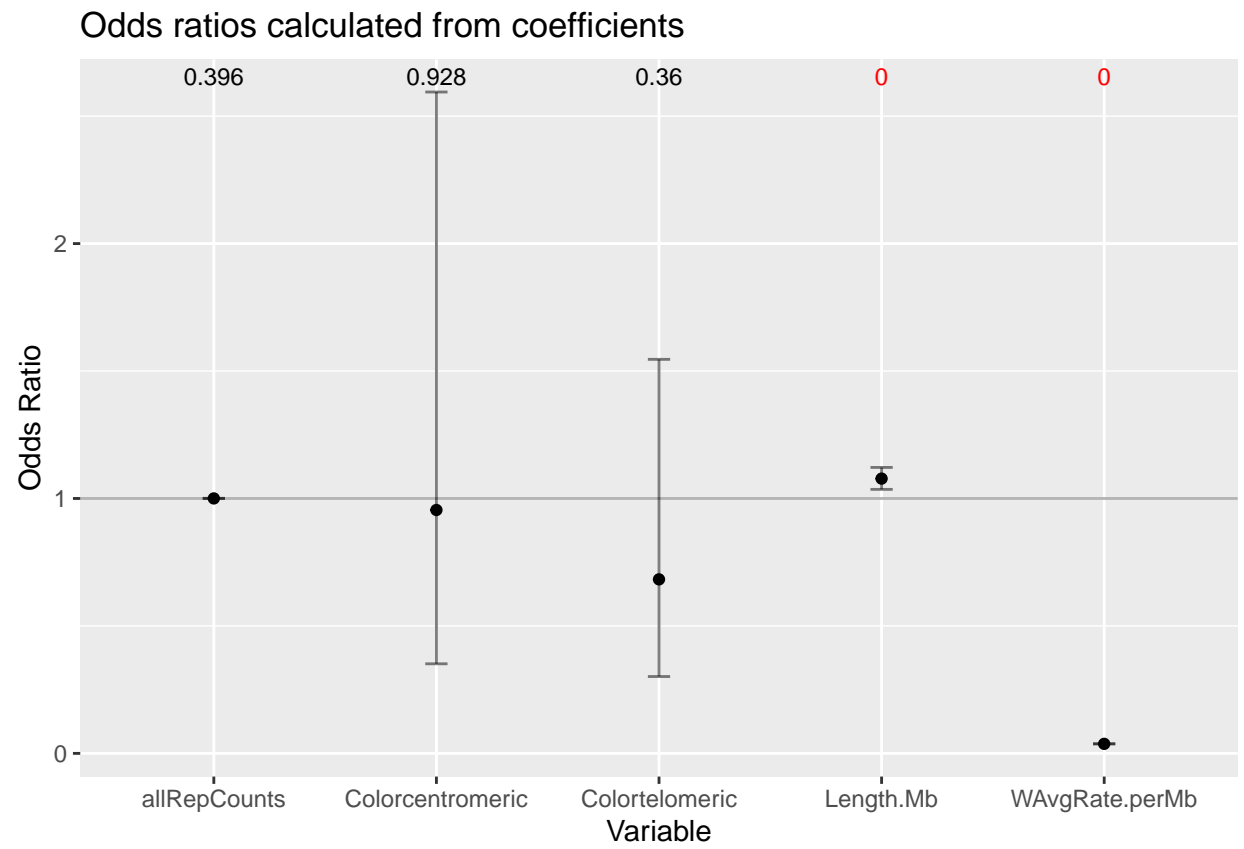
```
load(file= "data/winRegions.RData")

outvalue<-boxplot(winRegions$WAvgRate.perMb ~ winRegions$Color)$out
```



```
winRegions<-winRegions[!(winRegions$WAvgRate.perMb %in% outvalue),]
```

	Value	Std. Error	t value	p value
Length.Mb	0.0750515	0.0203656	3.6852054	0.0002285
allRepCounts	0.0002364	0.0002785	0.8488965	0.3959389
Colorcentromeric	-0.0462419	0.5100767	-0.0906567	0.9277654
Colortelomeric	-0.3817157	0.4170139	-0.9153548	0.3600054
WAvgRate.perMb	-3.2871793	0.0072546	-453.1177931	0.0000000
0 1	1.2684207	0.5930473	2.1388190	0.0324503
1 2	2.9027449	0.6421521	4.5203384	0.0000062
2 3+	4.2414116	0.7194241	5.8955647	0.0000000

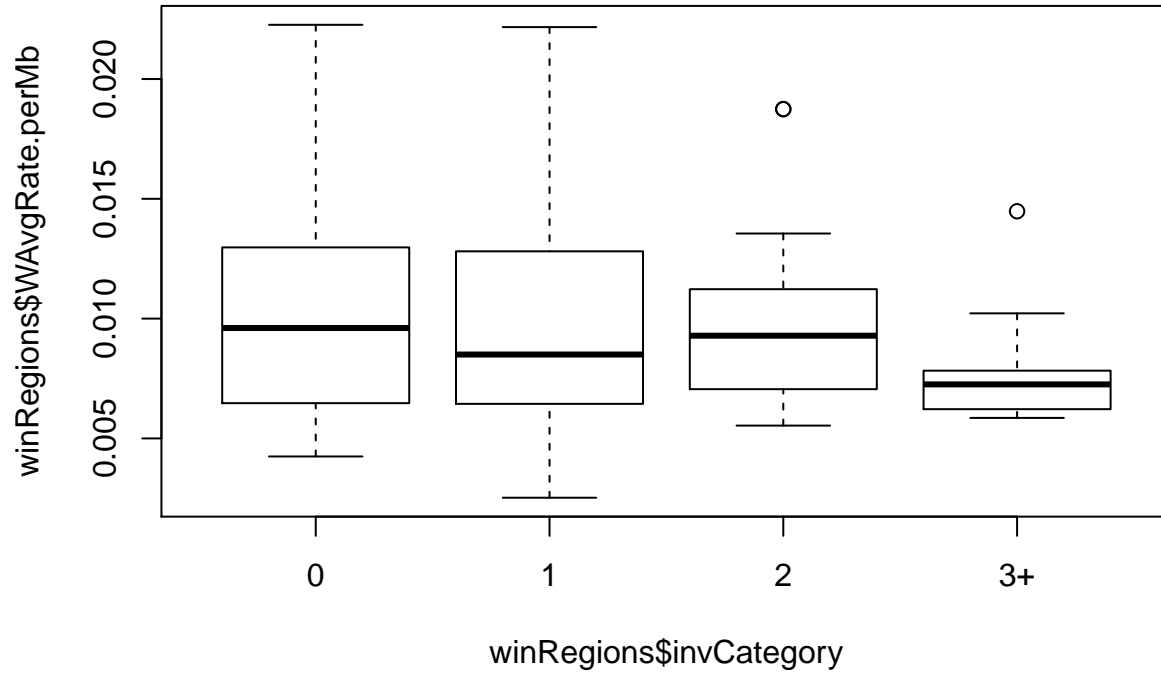




## Without per-category (InvCategory) outliers

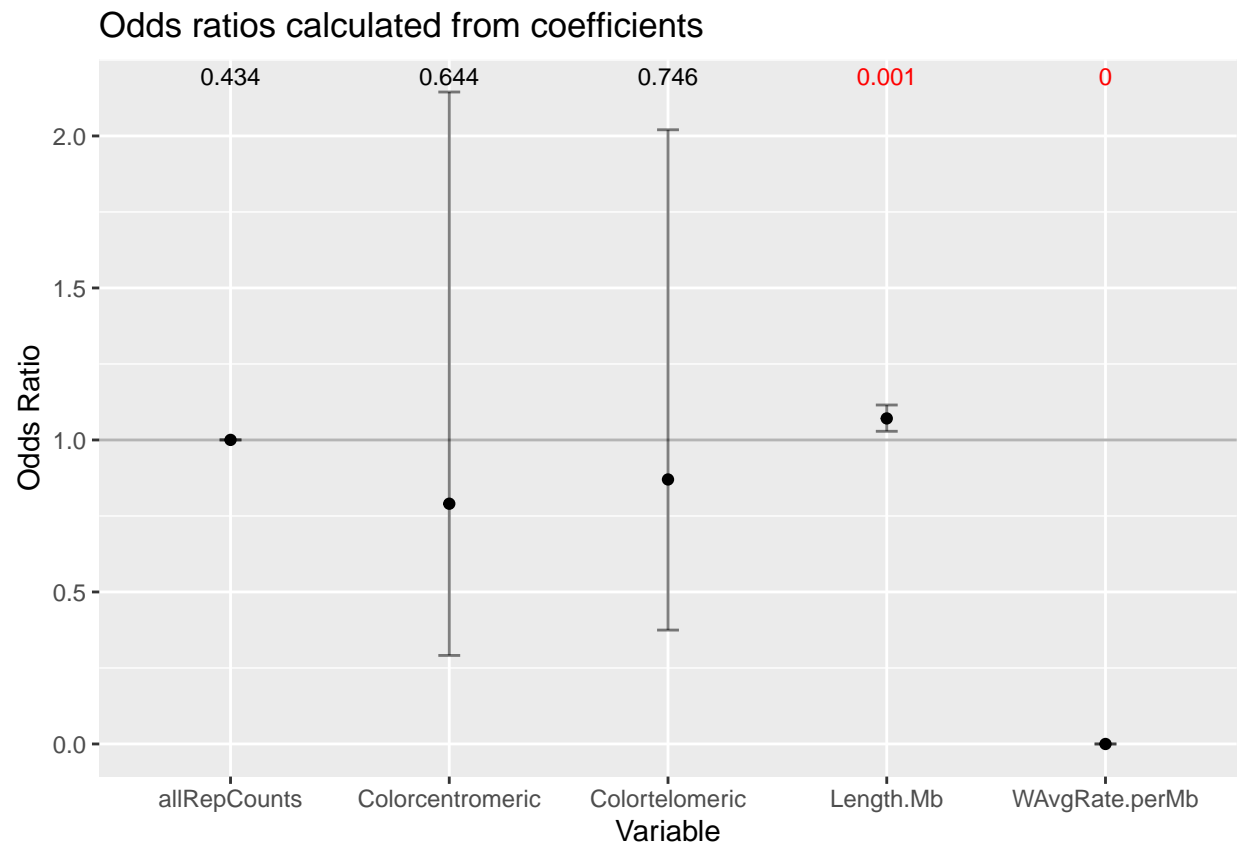
```
load(file= "data/winRegions.RData")
```

```
outvalue<-boxplot(winRegions$WAvgRate.perMb ~ winRegions$invCategory)$out
```



```
winRegions<-winRegions[!(winRegions$WAvgRate.perMb %in% outvalue),]
```

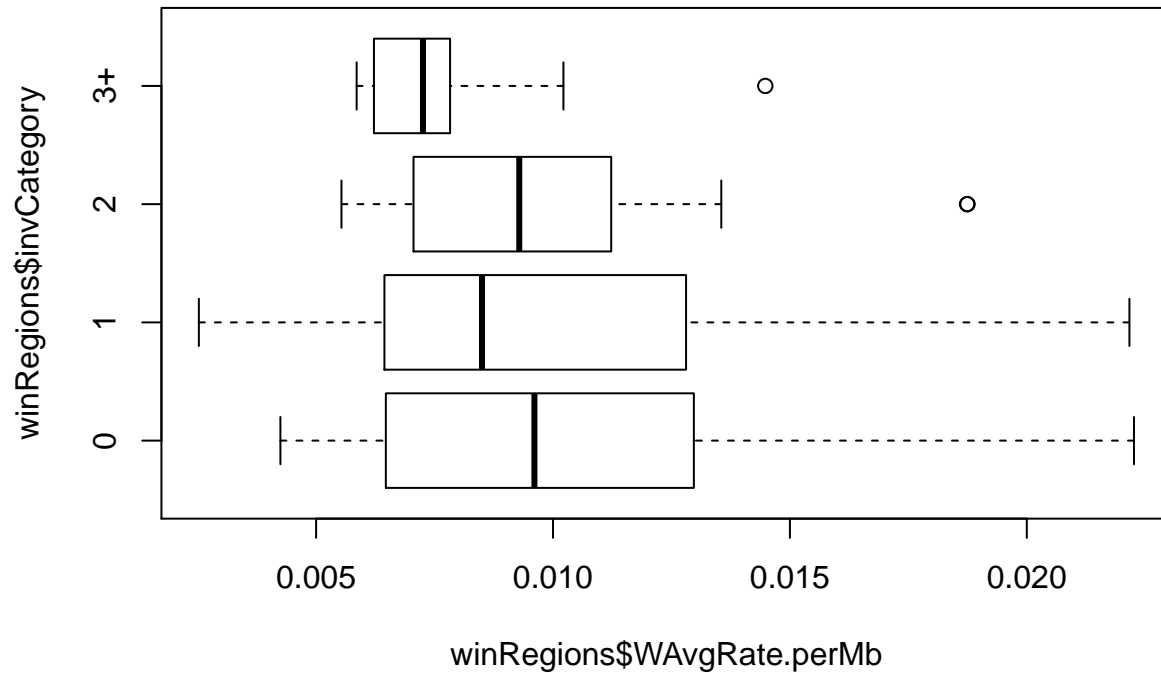
	Value	Std. Error	t value	p value
Length.Mb	0.0684941	0.0205945	3.325843e+00	0.0008815
allRepCounts	0.0002199	0.0002812	7.819999e-01	0.4342146
Colorcentromeric	-0.2352583	0.5092974	-4.619271e-01	0.6441336
Colortelomeric	-0.1392175	0.4298412	-3.238812e-01	0.7460280
WAvgRate.perMb	-82.0451558	0.0075305	-1.089501e+04	0.0000000
0 1	0.4400045	0.5997809	7.336087e-01	0.4631873
1 2	2.1706342	0.6546799	3.315566e+00	0.0009146
2 3+	3.4892504	0.7366953	4.736355e+00	0.0000022



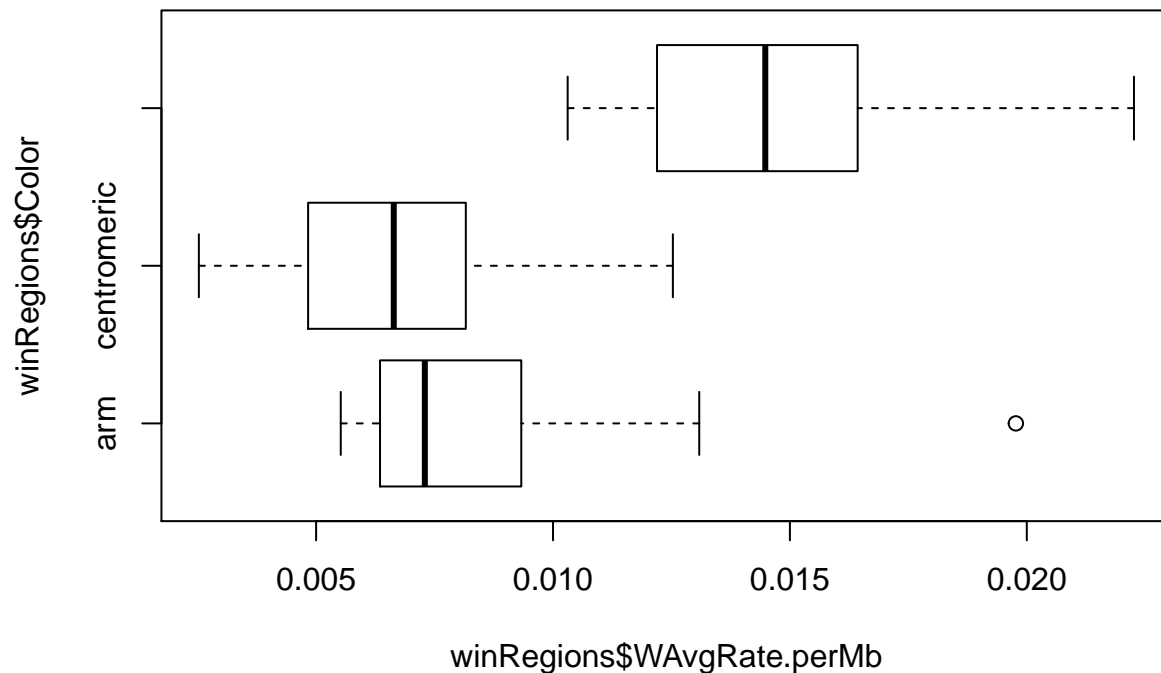
## Without all outliers

```
load(file= "data/winRegions.RData")
```

```
outvalue_A<-boxplot(winRegions$WAvGRate.perMb ~ winRegions$invCategory, horizontal = TRUE)$out
```



```
outvalue_B<-boxplot(winRegions$WAvGRate.perMb ~ winRegions$Color, horizontal = TRUE)$out
```

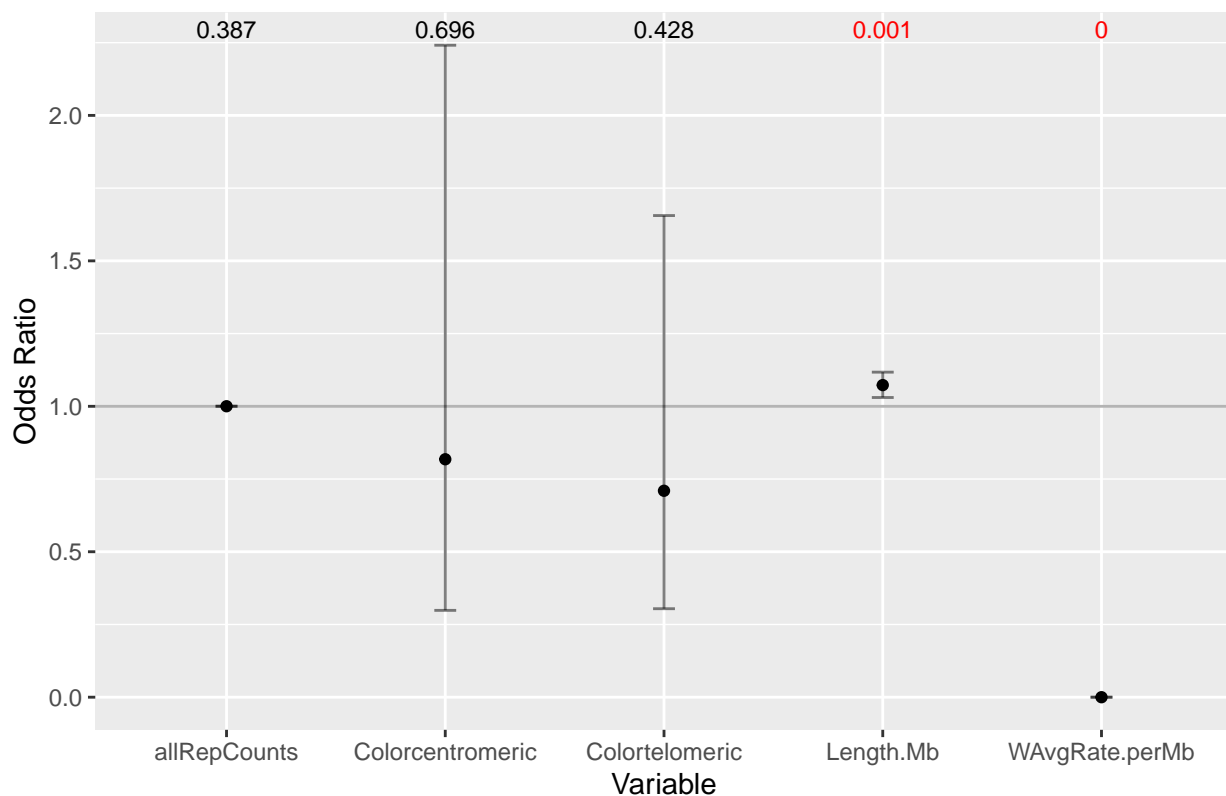


```
winRegions<-winRegions[!(winRegions$WAvGRate.perMb %in% outvalue_A),]  
winRegions<-winRegions[!(winRegions$WAvGRate.perMb %in% outvalue_B),]
```

```
save(yVar,xVars,myFormula,xVarsVector,samples , winRegions, file= "data/analysisParams.RData")
```

	Value	Std. Error	t value	p value
Length.Mb	0.0703901	0.0207286	3.3957920	0.0006843
allRepCounts	0.0002483	0.0002869	0.8653041	0.3868719
Colorcentromeric	-0.2009656	0.5142903	-0.3907629	0.6959725
Colortelomeric	-0.3429690	0.4322469	-0.7934562	0.4275121
WAvgRate.perMb	-52.8849659	0.0072844	-7260.0654212	0.0000000
0 1	0.7096547	0.6022166	1.1784044	0.2386354
1 2	2.4455977	0.6559532	3.7283115	0.0001928
2 3+	3.7692352	0.7380748	5.1068470	0.0000003

Odds ratios calculated from coefficients



## Manually remove group 1 outlier

```
load(file= "data/winRegions.RData")

outvalue<-max(winRegions[winRegions$invCategory == "1", "WAvgRate.perMb"])

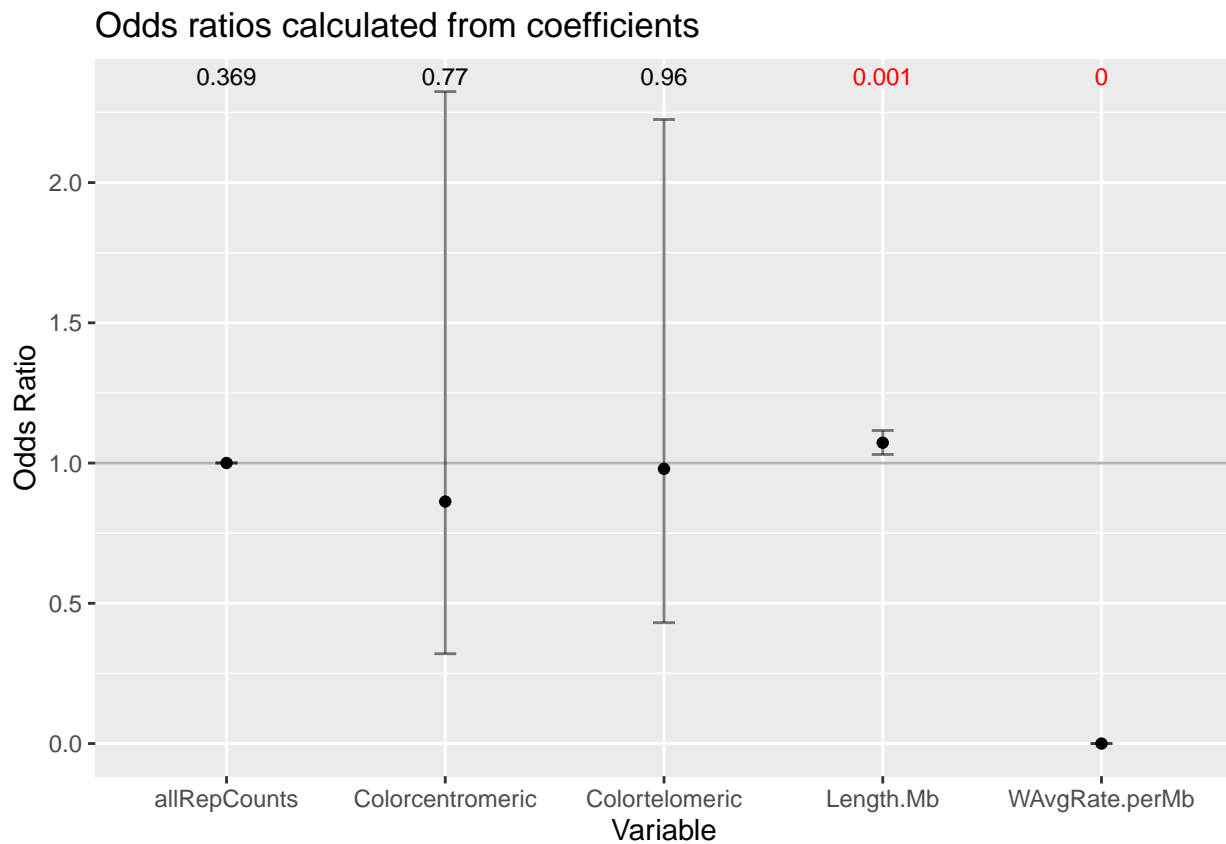
winRegions<-winRegions[!(winRegions$WAvgRate.perMb %in% outvalue),]
winRegions[winRegions$WAvgRate.perMb == outvalue,]
```

```
## [1] Chromosome          Start                End
```

```
## [4] Color                invCenters          NHCenters
## [7] NAHRCenters           Length.Mb          allRepCounts
## [10] WAvgRate.perMb        ChromType          invCategory
## [13] NHCategory            NAHRCategory       plotgroup
## [16] Length.Mb.Scaled      allRepCounts.Scaled WAvgRate.perMb.Scaled
## <0 rows> (or 0-length row.names)

save(yVar,xVars,myFormula,xVarsVector,samples , winRegions, file= "data/analysisParams.RData")
```

	Value	Std. Error	t value	p value
Length.Mb	0.0699748	0.0203525	3.4381454	0.0005857
allRepCounts	0.0002456	0.0002732	0.8990661	0.3686175
Colorcentromeric	-0.1475520	0.5056204	-0.2918236	0.7704215
Colortelomeric	-0.0209640	0.4186967	-0.0500697	0.9600668
WAvgRate.perMb	-65.3042648	0.0074218	-8799.0176031	0.0000000
0 1	0.6980202	0.5948266	1.1734852	0.2406013
1 2	2.3049403	0.6446221	3.5756457	0.0003494
2 3+	3.6414174	0.7219612	5.0437854	0.0000005



Which is this point?

```
load(file= "data/winRegions.RData")

# Setup
```

```

pacman::p_load(ggplot2)
name <- "CEUSpence_CDzones_0.03_1000000"
divChroms <- paste0("data/divideChromosomes/", name, ".")

# PATHS
windowsFile <- paste0(divChroms, "/windows.txt")
densityFile <- paste0(divChroms, "/densities.txt")
extremesFile <- paste0(divChroms, "/extremes.txt")
centroFile <- paste0(divChroms, "/workspace.txt")

# FILES
windows <- read.table(windowsFile, header = T, sep = "\t")
windows$Color <- "a"
windows[as.numeric(rownames(windows)) %% 2 == 1, "Color"] <- "b"
windows$Chromosome <- factor(windows$Chromosome, levels = paste(rep("chr", 23), as.character(c(c(1:22), "X", "Y"))))

density <- read.table(densityFile, header = T, sep = "\t")
density$Chromosome <- factor(density$Chromosome, levels = paste(rep("chr", 23), as.character(c(c(1:22), "X", "Y"))))

extremes <- read.table(extremesFile, header = T, sep = "\t")
extremes$Chromosome <- factor(extremes$Chromosome, levels = paste(rep("chr", 23), as.character(c(c(1:22), "X", "Y"))))

centromeres <- read.table(centroFile, header = T, sep = "\t")
centromeres$Chromosome <- factor(centromeres$Chromosome, levels = paste(rep("chr", 23), as.character(c(c(1:22), "X", "Y"))))
armLimits <- centromeres[grepl("cen", centromeres$chromID, invert = T), ]
centromeres <- centromeres[grepl("cen", centromeres$chromID), ]
starts <- armLimits[grepl("p", armLimits$chromID), c("Start", "Chromosome")]
ends <- armLimits[grepl("q", armLimits$chromID), c("End", "Chromosome")]

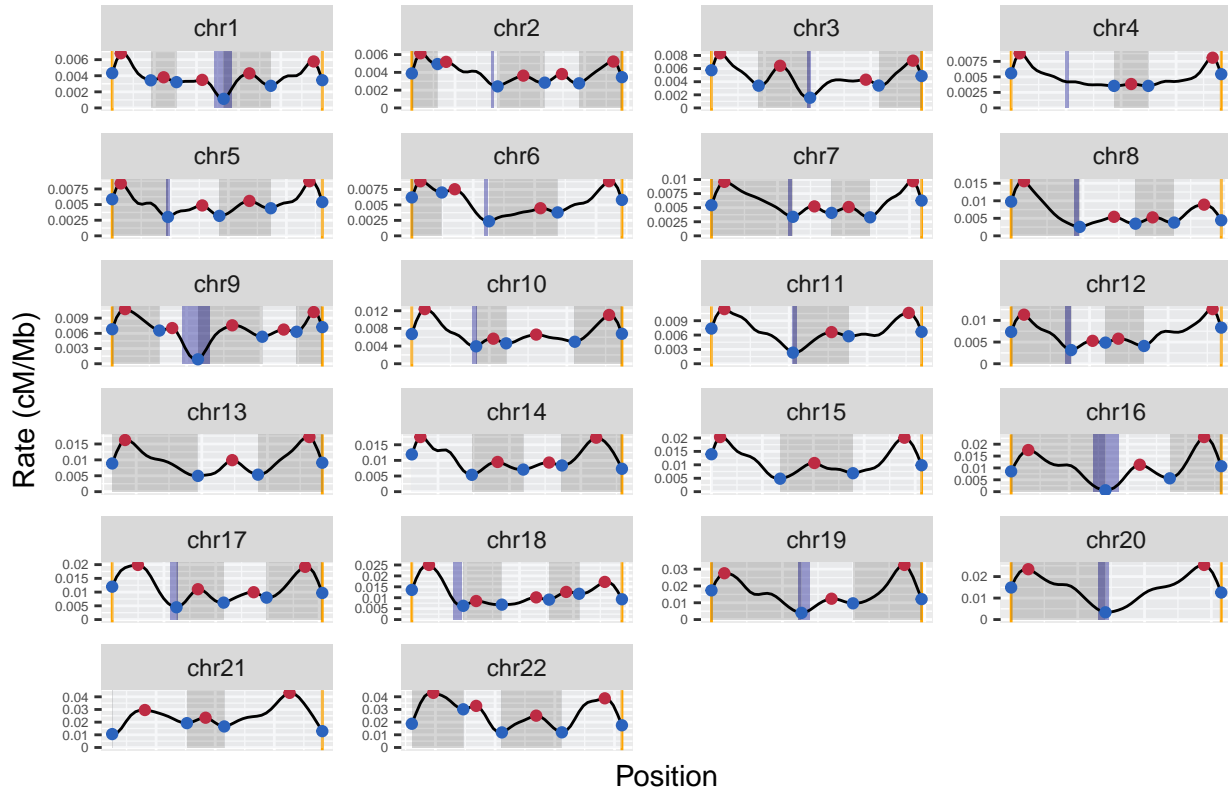
d_rowlist <- c()
c_rowlist <- c()
for (c in unique(starts$Chromosome)) {
  cStart <- min(extremes[extremes$Chromosome == c, "pos"])
  starts <- starts[!(starts$Chromosome == c & starts$Start < cStart), ] # chromosome start will be fixed
  windows[windows$Chromosome == c & windows$Start < cStart, "Start"] <- cStart # window start may be affected
  d_rowlist <- c(d_rowlist, rownames(density[density$Chromosome == c & density$pos < cStart, ])) # some density values
  c_rowlist <- c(c_rowlist, rownames(centromeres[centromeres$Chromosome == c & centromeres$Start < cStart, ]))
}
density <- density[!(rownames(density) %in% d_rowlist), ]
centromeres <- centromeres[!(rownames(centromeres) %in% c_rowlist), ]

# PLOT
ggplot()+
  geom_rect(data = centromeres, aes(xmin = Start, xmax = End, ymin = 0, ymax = Inf), fill = "blue", alpha = 0.5)+
  geom_vline(data = starts, aes(xintercept = Start), color = "orange")+geom_vline(data = ends, aes(xintercept = End), color = "orange")+
  geom_rect(data = windows, aes(xmin = Start, xmax = End, fill = Color, ymin = 0, ymax = Inf), alpha = 0.5)+
  geom_line(data = density, aes(x = pos, y = val))+
  geom_point(data = extremes, aes(x = pos, y = val, color = Type))+
  facet_wrap("Chromosome", scales = "free", ncol = 4)+
  scale_fill_manual(values = c("#737373", "#e1e5eb"), guide = "none")+
  scale_color_manual(values = c("#bd2b43", "#2b63bd"), guide = "none")+

```

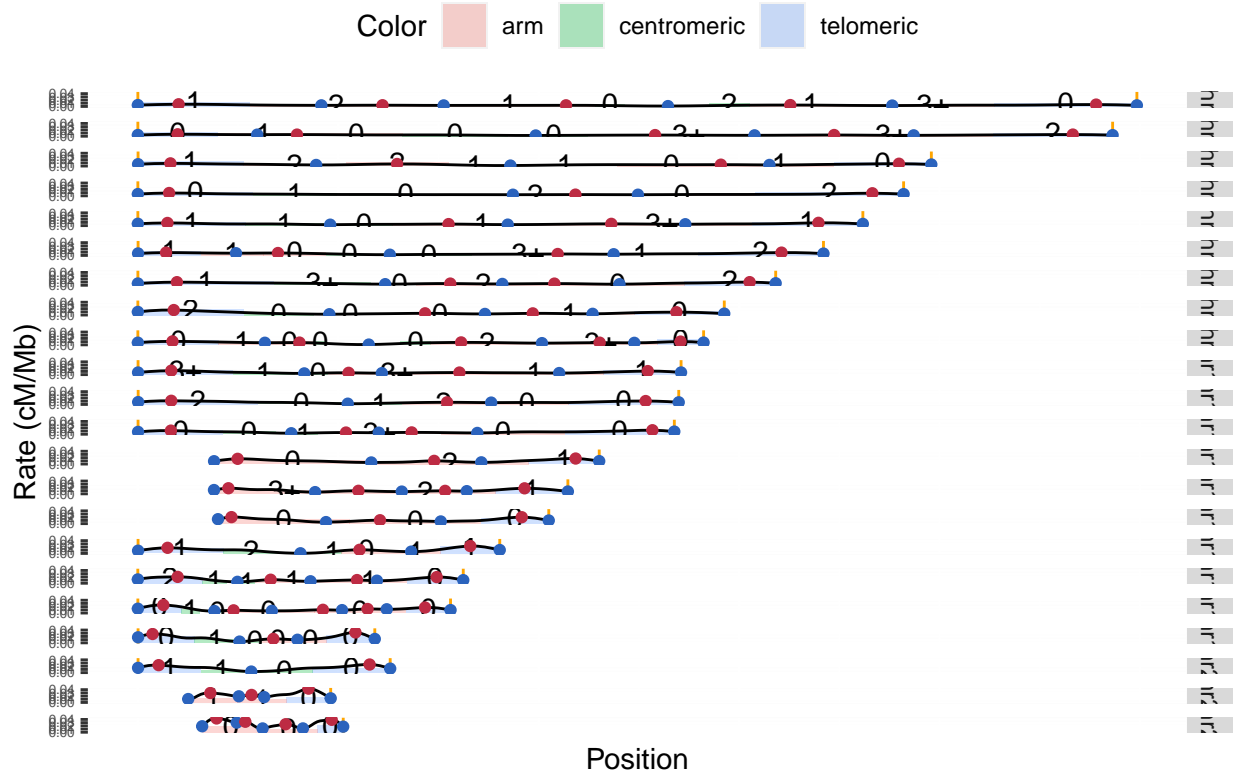
```
scale_y_continuous(labels=function(x)x*1000000)+
  theme(axis.ticks.x = element_blank(),axis.text.x = element_blank(),axis.text.y = element_text(
  ylab("Rate (cM/Mb)")+xlab("Position")+ggtitle(name)
```

## CEUSpence\_COzones\_0.03\_1000000



```
ggplot()+
  # geom_rect(data= centromeres, aes(xmin = Start, xmax = End, ymin = 0, ymax = Inf), fill = "b
  geom_vline(data=starts, aes(xintercept = Start), color = "orange")+geom_vline(data=ends, aes(xin
  geom_rect(data=winRegions, aes(xmin = Start, xmax = End, fill = Color, ymin = 0, ymax = WAvgrate
  geom_line(data=density,aes(x = pos, y = val * 1000000 ))+
  geom_text(data = winRegions, aes(x = (Start+End)/2 , y = WAvgrate.perMb, label = invCategory))+
  geom_point(data=extremes, aes(x = pos, y =val * 1000000 , color = Type))+
  facet_grid("Chromosome", scales = "free_x", space = "free_x")+
  # scale_fill_manual(values=c("#737373", "#e1e5eb"), guide="none")+
  scale_color_manual(values = c("#bd2b43", "#2b63bd"), guide = "none")+
  # scale_y_continuous(labels=function(x)x*1000000)+
  theme(axis.ticks.x = element_blank(),axis.text.x = element_blank(),axis.text.y = element_text(
  ylab("Rate (cM/Mb)")+xlab("Position")+ggtitle(name)
```

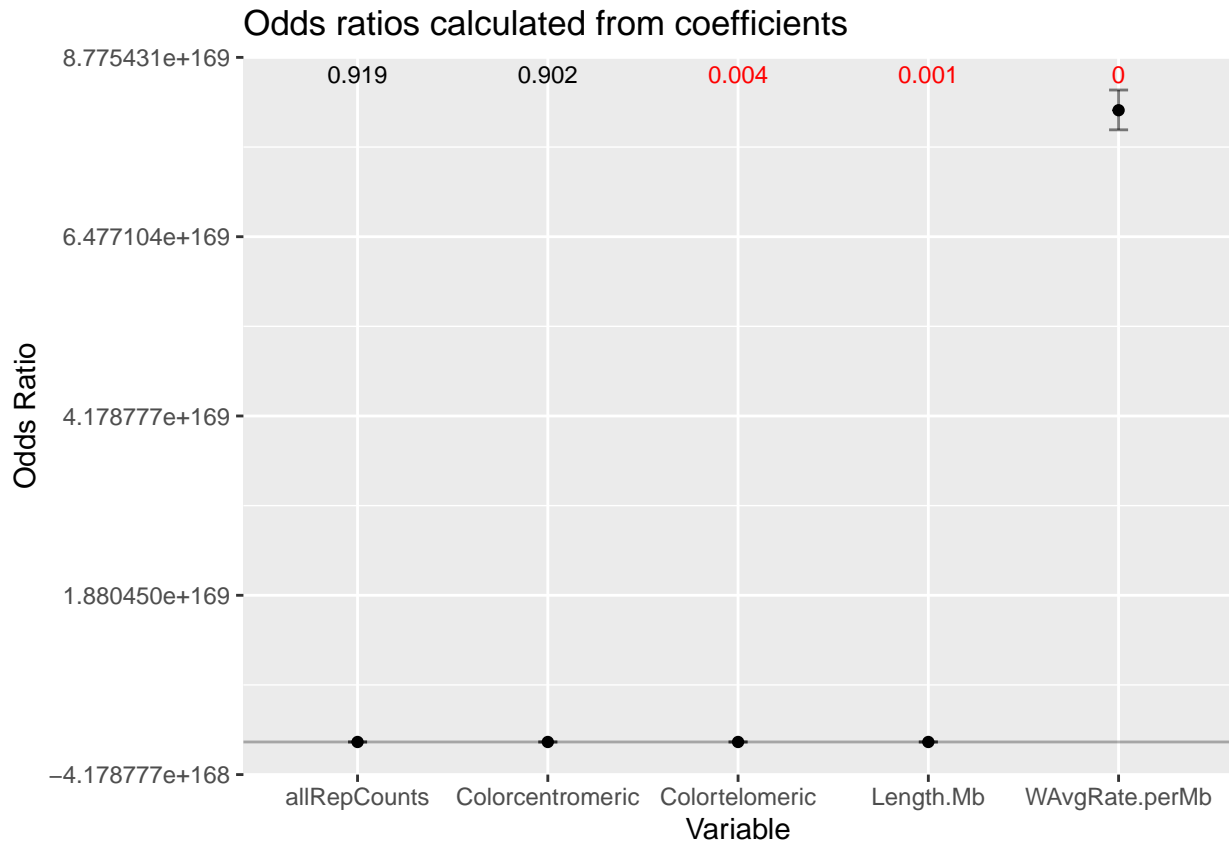
## CEUSpence\_COzones\_0.03\_1000000



Without some chromosomes

	Value	Std. Error	t value	p value
Length.Mb	0.1747090	0.0505233	3.4579886	0.0005442
allRepCounts	-0.0000613	0.0006051	-0.1013287	0.9192896
Colorcentromeric	0.1413892	1.1510196	0.1228382	0.9022352
Colortelomeric	-2.2741922	0.7817295	-2.9091803	0.0036238
WAvgRate.perMb	391.2285330	0.0160786	24332.3073241	0.0000000
0 1	6.3872174	1.3625175	4.6878060	0.0000028
1 2	8.0249932	1.5046694	5.3333931	0.0000001
2 3+	9.1366160	1.5889662	5.7500378	0.0000000





Tested unsuccessfully:

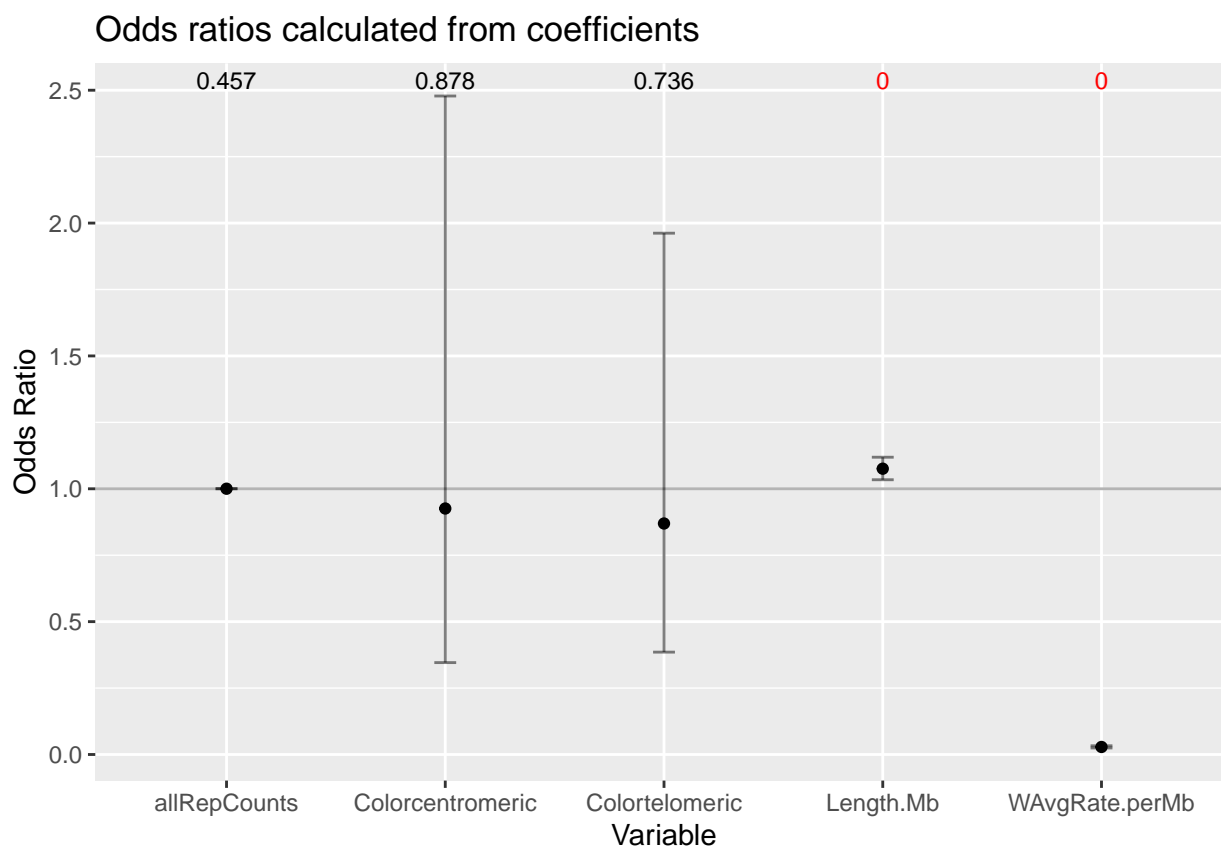
- no Chr22
- no Chr21
- no Chr19
- chroms 1,2,3,4,5
- chroms 6,7,8,9,10

So it seems something about the data...

## Multiply the numbers

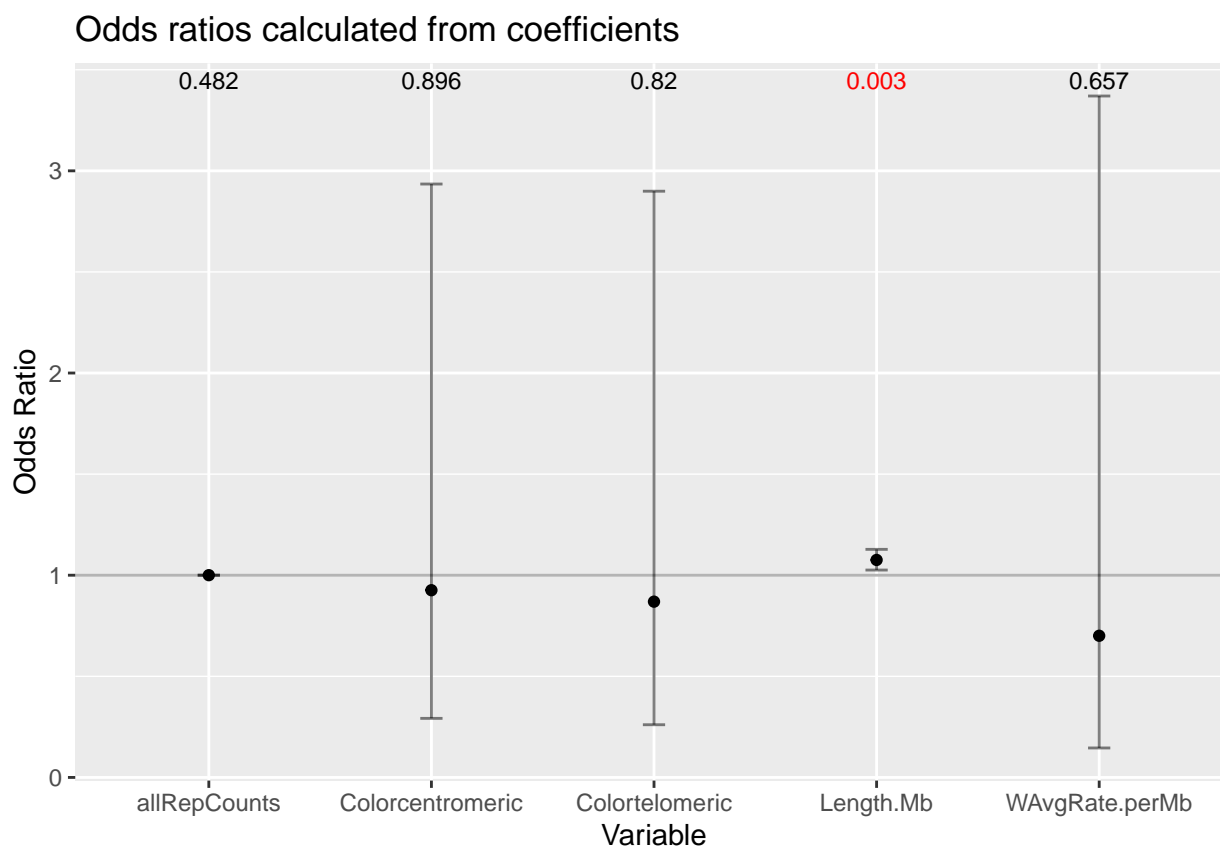
cM/Mb \*10

	Value	Std. Error	t value	p value
Length.Mb	0.0728497	0.0201091	3.6227291	0.0002915
allRepCounts	0.0002031	0.0002730	0.7437002	0.4570579
Colorcentromeric	-0.0771374	0.5023924	-0.1535401	0.8779723
Colortelomeric	-0.1399662	0.4152548	-0.3370611	0.7360709
WAvgRate.perMb	-3.5621851	0.0743385	-47.9184552	0.0000000
0 1	0.9713314	0.5817022	1.6698087	0.0949572
1 2	2.5961332	0.6326116	4.1038344	0.0000406
2 3+	3.9276278	0.7102449	5.5299626	0.0000000



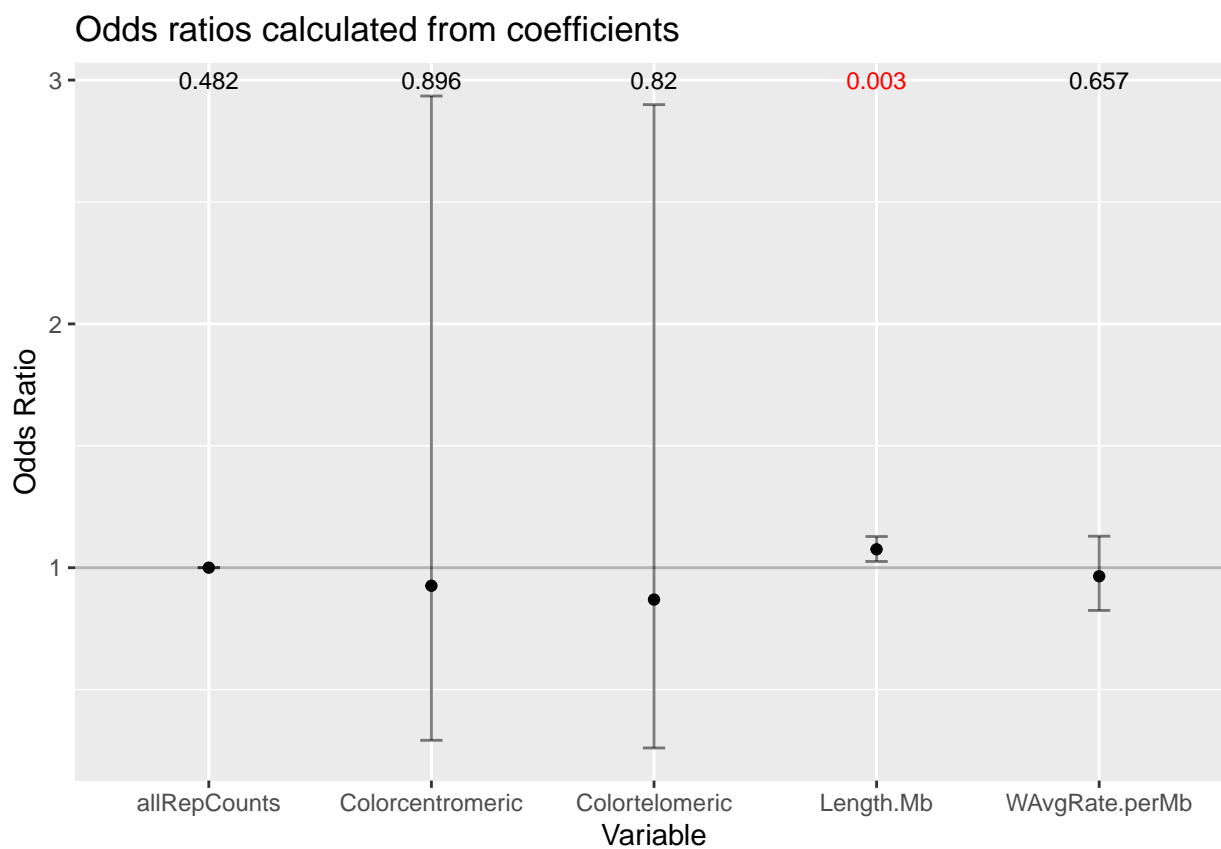
cM/Mb \*100

	Value	Std. Error	t value	p value
Length.Mb	0.0728569	0.0242608	3.0030712	0.0026727
allRepCounts	0.0002030	0.0002889	0.7026144	0.4822961
Colorcentromeric	-0.0769941	0.5885796	-0.1308134	0.8959230
Colortelomeric	-0.1401181	0.6145913	-0.2279858	0.8196573
WAvgRate.perMb	-0.3558995	0.8014672	-0.4440599	0.6569992
0 1	0.9717801	1.1144136	0.8720103	0.3832027
1 2	2.5965576	1.1354513	2.2868067	0.0222071
2 3+	3.9280475	1.1857060	3.3128344	0.0009236



cM/Mb \*1000

	Value	Std. Error	t value	p value
Length.Mb	0.0728566	0.0242608	3.0030590	0.0026728
allRepCounts	0.0002030	0.0002889	0.7026109	0.4822983
Colorcentromeric	-0.0769918	0.5885793	-0.1308095	0.8959260
Colortelomeric	-0.1401134	0.6145917	-0.2279780	0.8196634
WAvRate.perMb	-0.0355908	0.0801473	-0.4440671	0.6569941
0 1	0.9717631	1.1144154	0.8719935	0.3832119
1 2	2.5965472	1.1354544	2.2867913	0.0222080
2 3+	3.9280298	1.1857092	3.3128104	0.0009236



cM/Mb \*10000

	Value	Std. Error	t value	p value
Length.Mb	0.0728569	0.0242611	3.0030340	0.0026730
allRepCounts	0.0002030	0.0002889	0.7026239	0.4822902
Colorcentromeric	-0.0770008	0.5885540	-0.1308305	0.8959094
Colortelomeric	-0.1401096	0.6146278	-0.2279584	0.8196786
WAvgRate.perMb	-0.0035591	0.0080205	-0.4437485	0.6572244
0 1	0.9717729	1.1146403	0.8718264	0.3833031
1 2	2.5965559	1.1357978	2.2861075	0.0222480
2 3+	3.9280357	1.1860982	3.3117289	0.0009272

