Figures

```
library(tidyverse)
library(MASS, exclude = "select")
library(marginaleffects)
library(broom)
library(car)
library(survival)
```

Figure 1: Packages

```
   Year Seed Final4 Izzo
1  2003    7      0    1
2  2007   10      0    0
3  1992   16      0    0
4  2002    3      0    0
5  1986    6      0    0
6  1994    7      0    0
7  2003    1      0    0
8  1985    3      0    0
9  1987    3      0    0
10 1993    8      0    0
11 2004    7      0    0
12 1990   13      0    0
13 1992   15      0    0
14 2003   12      0    0
15 1998    9      0    0
16 1999    9      0    0
17 2001    5      0    0
18 2007   15      0    0
19 2000    8      0    0
20 1986    4      0    0
```

Figure 2: Izzo data (20 randomly chosen rows)

```r
izzo.1 <- glm(Final4 ~ Seed + Izzo, data = FinalFourIzzo, family = "binomial")
summary(izzo.1)
```

```
Call:
glm(formula = Final4 ~ Seed + Izzo, family = "binomial", data = FinalFourIzzo)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.02776    0.19967   0.139  0.88942
Seed        -0.58809    0.05909  -9.953  < 2e-16 ***
Izzo         2.32441    0.73971   3.142  0.00168 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 778.06  on 1663  degrees of freedom
Residual deviance: 519.56  on 1661  degrees of freedom
AIC: 525.56

Number of Fisher Scoring iterations: 8
```

Figure 3: Izzo data logistic regression

```
new <- datagrid(model = izzo.1, Seed = c(1, 5, 9, 13), Izzo = c(0, 1))
cbind(predictions(izzo.1, new)) %>%
  select(Seed, Izzo, estimate, conf.low, conf.high)
```

```
  Seed Izzo     estimate     conf.low   conf.high
1    1    0 0.3634724918 0.2963916353 0.436320736
2    1    1 0.8537203338 0.5735880449 0.962008512
3    5    0 0.0515307373 0.0370854692 0.071186657
4    5    1 0.3570333767 0.1174754651 0.698471533
5    9    0 0.0051427167 0.0024183140 0.010902808
6    9    1 0.0501821692 0.0112579158 0.196888042
7   13    0 0.0004915945 0.0001470338 0.001642276
8   13    1 0.0050017290 0.0008634866 0.028408518
```

Figure 4: Izzo data predictions

```
plot_predictions(izzo.1, condition = c("Seed", "Izzo"))
```

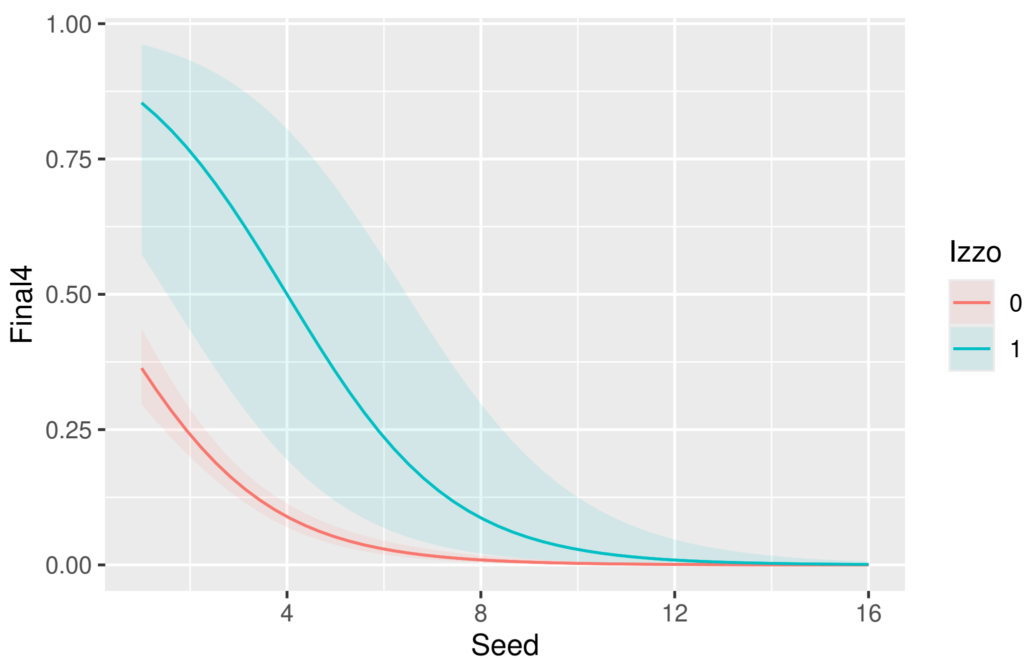


Figure 5: Graph of predictions for Izzo data

```
     Task Report  Time
1  Visual Verbal  5.74
2  Visual Visual 28.15
3  Visual Verbal  6.68
4  Visual Visual 15.85
5  Verbal Visual  8.44
6  Verbal Visual 13.16
7  Visual Verbal  5.88
8  Verbal Verbal 11.37
9  Verbal Verbal 18.28
10 Verbal Visual 14.69
```

Figure 6: Brain side data (10 randomly chosen rows)

```
ggplot(VisualVerbal, aes(x = Task, y = Time, fill = Report)) + geom_boxplot()
```

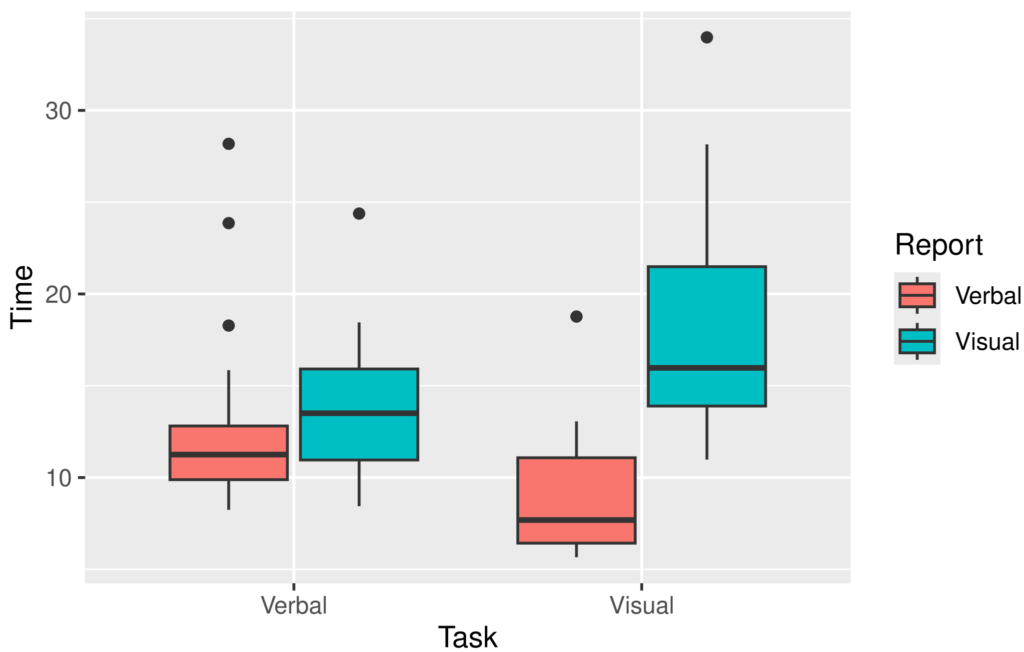


Figure 7: Brain side boxplots

```r
vis.1 <- aov(log(Time) ~ Task * Report, data = VisualVerbal)
summary(vis.1)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
Task         1  0.033  0.0335    0.33    0.567    
Report       1  3.136  3.1360   30.92 3.83e-07 ***
Task:Report  1  1.963  1.9630   19.36 3.49e-05 ***
Residuals   76  7.708  0.1014                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 8: Brain side ANOVA

```r
VisualVerbal %>% filter(Task == "Verbal") -> verbals
verbals.1 <- aov(log(Time) ~ Report, data = verbals)
summary(verbals.1)
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
Report       1  0.068 0.06839   0.722  0.401
Residuals   38  3.599 0.09470               
```

Figure 9: Brain side further analysis part 1

```r
VisualVerbal %>% filter(Task == "Visual") -> visuals
visuals.1 <- aov(log(Time) ~ Report, data = visuals)
summary(visuals.1)
```

```
            Df Sum Sq Mean Sq F value  Pr(>F)    
Report       1  5.031   5.031   46.52 4.3e-08 ***
Residuals   38  4.109   0.108                    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 10: Brain side further analysis part 2

	Loc	Var	Y1	Y2
1	UF	M	81.0	80.7
2	UF	S	105.4	82.3
3	UF	V	119.7	80.4
4	UF	T	109.7	87.2
5	UF	P	98.3	84.2
6	W	M	146.6	100.4
7	W	S	142.0	115.5
8	W	V	150.7	112.2
9	W	T	191.5	147.7
10	W	P	145.7	108.1
11	M	M	82.3	103.1
12	M	S	77.3	105.1
13	M	V	78.4	116.5
14	M	T	131.3	139.9
15	M	P	89.6	129.6
16	C	M	119.8	98.9
17	C	S	121.4	61.9
18	C	V	124.0	96.2
19	C	T	140.8	125.5
20	C	P	124.8	75.7
21	GR	M	98.9	66.4
22	GR	S	89.0	49.9
23	GR	V	69.1	96.7
24	GR	T	89.3	61.9
25	GR	P	104.1	80.3
26	D	M	86.9	67.7
27	D	S	77.1	66.7
28	D	V	78.9	67.4
29	D	T	101.8	91.8
30	D	P	96.0	94.1

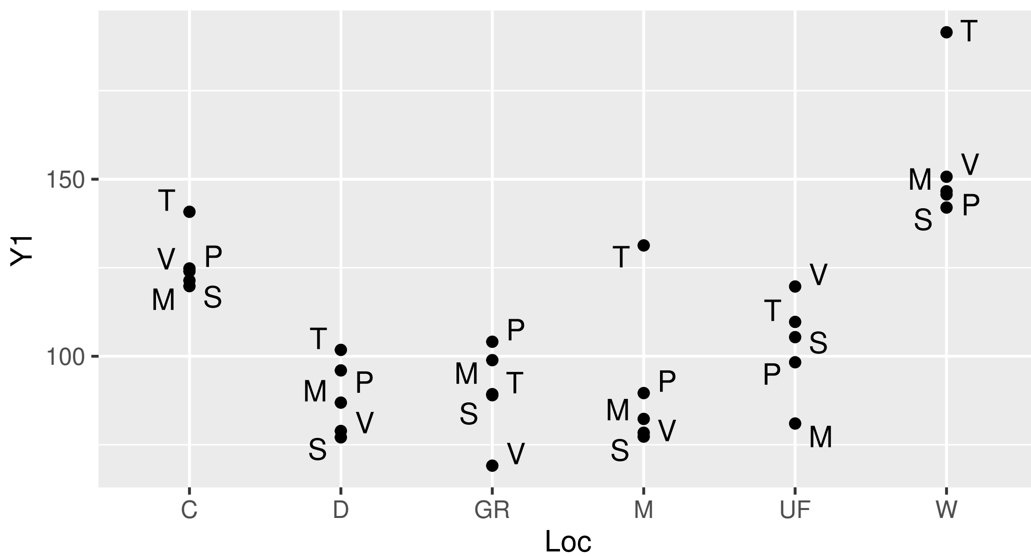
Figure 11: Barley yield data (all)

```
immer %>% select(starts_with("Y")) %>% as.matrix() -> y
immer.1 <- manova(y ~ Var + Loc, data = immer)
summary(immer.1)
```

```
          Df  Pillai approx F num Df den Df    Pr(>F)    
Var        4 0.64205    2.364      8     40   0.03469 *  
Loc        5 1.50658   12.213     10     40 2.543e-09 ***
Residuals 20                                             
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 12: Barley yield MANOVA

```
ggplot(immer, aes(x = Loc, y = Y1, label = Var)) + geom_point() +
  geom_text_repel()
```



```
ggplot(immer, aes(x = Loc, y = Y2, label = Var)) + geom_point() +
  geom_text_repel()
```

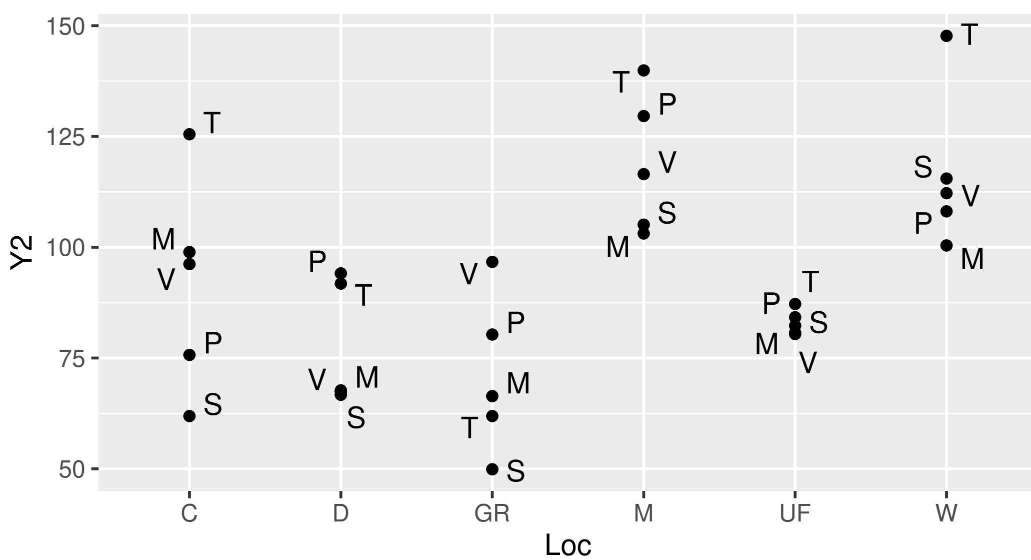


Figure 13: Barley yield plots

	id	grip.type	replicate	UBP
1	c_1	classic	1	168.2084
2	c_1	classic	2	161.4141
3	c_1	classic	3	163.2345
4	c_2	classic	1	155.9429
5	c_2	classic	2	168.5388
6	c_2	classic	3	166.3163
7	c_3	classic	1	162.6191
8	c_3	classic	2	157.8030
9	c_3	classic	3	171.6529
10	c_4	classic	1	165.1400
11	c_4	classic	2	164.9525
12	c_4	classic	3	158.2008
13	m_1	modern	1	160.0739
14	m_1	modern	2	161.2383
15	m_1	modern	3	166.7635
16	m_2	modern	1	161.8334
17	m_2	modern	2	162.7900
18	m_2	modern	3	157.5793
19	m_3	modern	1	165.2248
20	m_3	modern	2	162.7804
21	m_3	modern	3	159.7632
22	m_4	modern	1	160.3049
23	m_4	modern	2	168.5381
24	m_4	modern	3	164.4688
25	i_1	integrated	1	166.7134
26	i_1	integrated	2	173.0319
27	i_1	integrated	3	173.2537
28	i_2	integrated	1	165.4825
29	i_2	integrated	2	166.0498
30	i_2	integrated	3	170.5794
31	i_3	integrated	1	174.8182
32	i_3	integrated	2	166.8222
33	i_3	integrated	3	165.2776
34	i_4	integrated	1	174.8661
35	i_4	integrated	2	173.0058
36	i_4	integrated	3	165.1532

Figure 14: Ski grip data (all)

```
grip %>%
  group_by(grip.type, replicate) %>%
  summarize(mean_ubp = mean(UBP)) %>%
  ggplot(aes(x = replicate, y = mean_ubp,
             colour = grip.type, group = grip.type)) +
    geom_point() + geom_line()
```

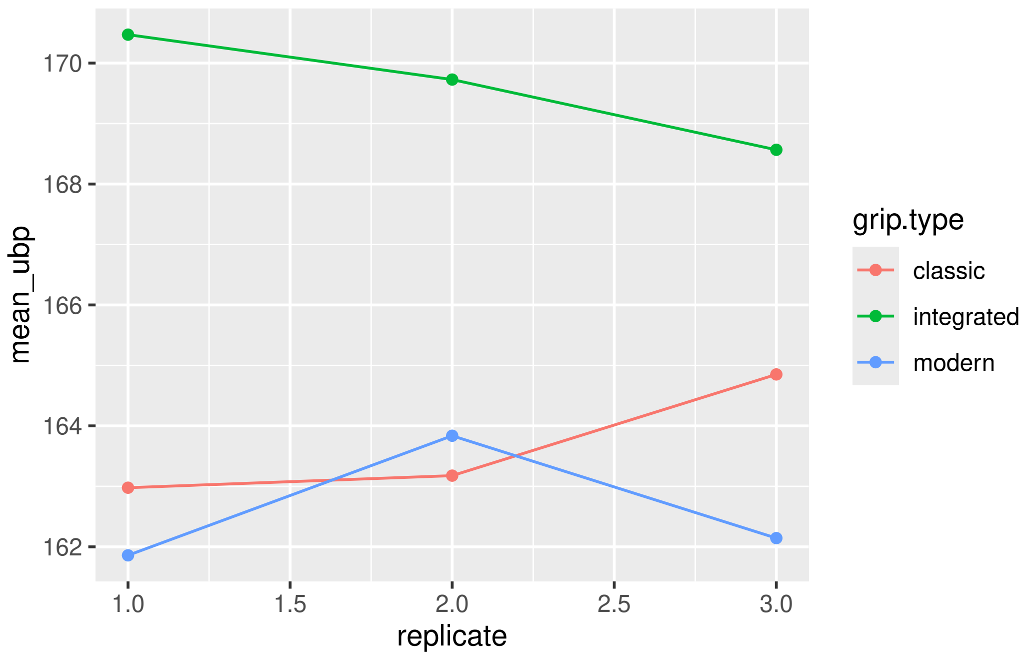


Figure 15: Ski grip interaction plot

```
ggplot(grip, aes(x = replicate, y = UBP, colour = grip.type, group = id)) +
  geom_point() + geom_line()
```

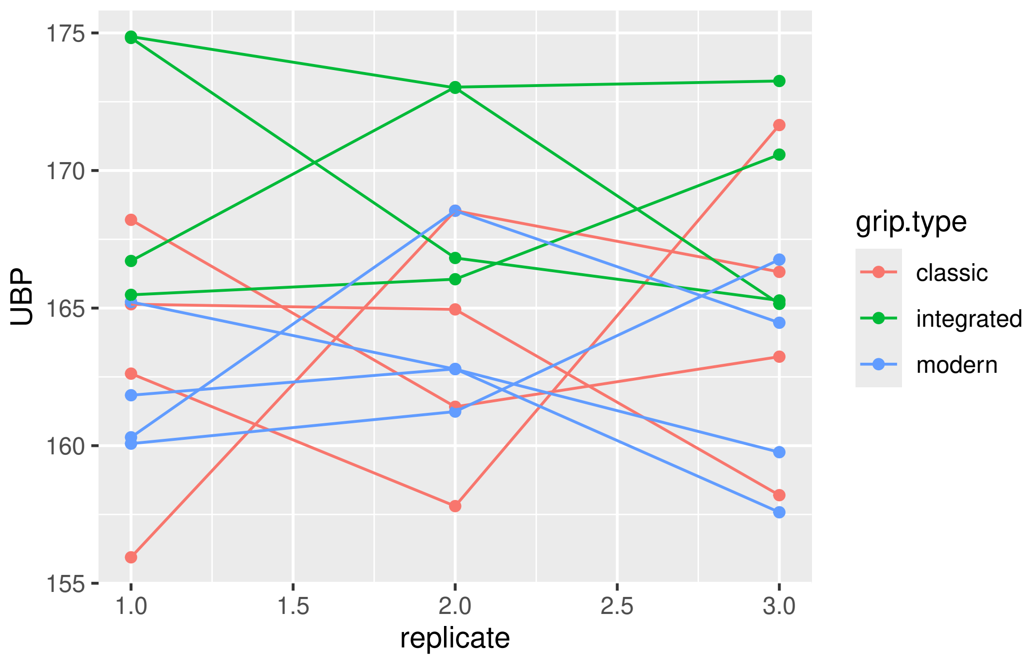


Figure 16: Ski grip spaghetti plot

```
grip %>% pivot_wider(names_from = replicate, values_from = UBP) -> grip_wide
```

Figure 17: Ski grip code

```
grip_wide %>%
  select(`1`:`3`) %>%
  as.matrix() -> y
grip.1a <- lm(y ~ grip.type, data = grip_wide)
times <- colnames(y)
times.df <- data.frame(times = factor(times))
grip.1 <- Manova(grip.1a, idata = times.df, idesign = ~ times)
```

Figure 18: Ski grip analysis code

```
Univariate Type II Repeated-Measures ANOVA Assuming Sphericity

                Sum Sq num Df Error SS den Df    F value    Pr(>F)
(Intercept)     983547      1    52.50      9 1.6861e+05 < 2.2e-16 ***
grip.type          339      2    52.50      9 2.9073e+01 0.0001182 ***
times                2      2   458.88     18 3.0600e-02 0.9698719
grip.type:times     23      4   458.88     18 2.2970e-01 0.9181327
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Mauchly Tests for Sphericity

                Test statistic p-value
times                   0.9088 0.68214
grip.type:times         0.9088 0.68214


Greenhouse-Geisser and Huynh-Feldt Corrections
 for Departure from Sphericity

                 GG eps Pr(>F[GG])
times           0.91642     0.9616
grip.type:times 0.91642     0.9057

                  HF eps Pr(>F[HF])
times           1.139118  0.9698719
grip.type:times 1.139118  0.9181327
```

Figure 19: Ski grip analysis output

Player	position	FGPct	FG3Pct	FTPct	Rebounds	Steals	Fouls
Rudy Gay	forward	0.455	0.359	0.858	5.867647	1.0441176	2.308823
Jamal Crawford	guard	0.396	0.327	0.901	1.937500	0.9218750	1.687500
J.J. Redick	guard	0.477	0.437	0.901	2.141026	0.5000000	1.717949
Robin Lopez	centre	0.535	0.000	0.772	6.677966	0.2711864	2.067797
Greivis Vasquez	guard	0.408	0.379	0.758	2.634146	0.5609756	2.158537
Brandon Jennings	guard	0.401	0.360	0.839	2.536585	1.0731707	1.560976
Zach Randolph	forward	0.487	0.350	0.765	10.521127	0.9718310	2.464789
Tony Allen	guard	0.495	0.345	0.627	4.444444	2.0476190	2.634921
Al Jefferson	centre	0.481	0.400	0.655	8.430769	0.7230769	2.138462
Jeff Teague	guard	0.460	0.343	0.862	2.520548	1.7123288	1.904110
Derrick Rose	guard	0.405	0.280	0.813	3.156863	0.7058824	1.235294
Andre Drummond	centre	0.514	0.000	0.389	13.463415	0.8902439	3.475610
Luol Deng	forward	0.469	0.355	0.761	5.222222	0.9027778	1.513889
Jeff Green	forward	0.430	0.332	0.833	4.205128	0.6794872	1.884615
Norris Cole	guard	0.412	0.313	0.716	2.120000	0.7600000	1.653333

Figure 20: NBA 2015 data (15 randomly selected rows)

```
y <- with(nba, cbind(FGPct, FG3Pct, FTPct, Rebounds, Steals, Fouls))
nba.2 <- manova(y ~ position, data = nba)
summary(nba.2)
```

```
           Df  Pillai approx F num Df den Df    Pr(>F)
position    2 0.72575   15.852     12    334 < 2.2e-16 ***
Residuals 171
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 21: NBA 2015 MANOVA

```
nba.1 <- lda(position ~ FGPct + FG3Pct + FTPct + Rebounds + Steals +
               Fouls, data = nba)
nba.1
```

```
Call:
lda(position ~ FGPct + FG3Pct + FTPct + Rebounds + Steals + Fouls,
    data = nba)

Prior probabilities of groups:
   centre   forward     guard
0.1379310 0.3563218 0.5057471

Group means:
            FGPct    FG3Pct     FTPct Rebounds    Steals    Fouls
centre  0.5214167 0.2160417 0.6998750 8.642295 0.6699533 2.614796
forward 0.4553065 0.3293065 0.7515161 5.941030 0.9615395 2.228547
guard   0.4277500 0.3467841 0.7935682 3.518978 1.1223494 2.085363

Coefficients of linear discriminants:
                LD1        LD2
FGPct    -6.7149807 17.4822905
FG3Pct    1.1377356 -5.7597904
FTPct     0.3022703  3.2938611
Rebounds -0.4652831 -0.4747002
Steals    1.4300714 -0.1587456
Fouls    -0.1288587  0.9834488

Proportion of trace:
   LD1    LD2
0.9537 0.0463
```

Figure 22: NBA 2015 discriminant analysis

```
p <- predict(nba.1)
d <- cbind(nba, p)
with(d, table(obs = position, pred = class))
```

```
         pred
obs       centre forward guard
  centre      18       6     0
  forward      9      31    22
  guard        1       4    83
```

Figure 23: NBA 2015 further discriminant analysis

```
ggplot(d, aes(x = x.LD1, y = x.LD2, colour = position)) + geom_point()
```

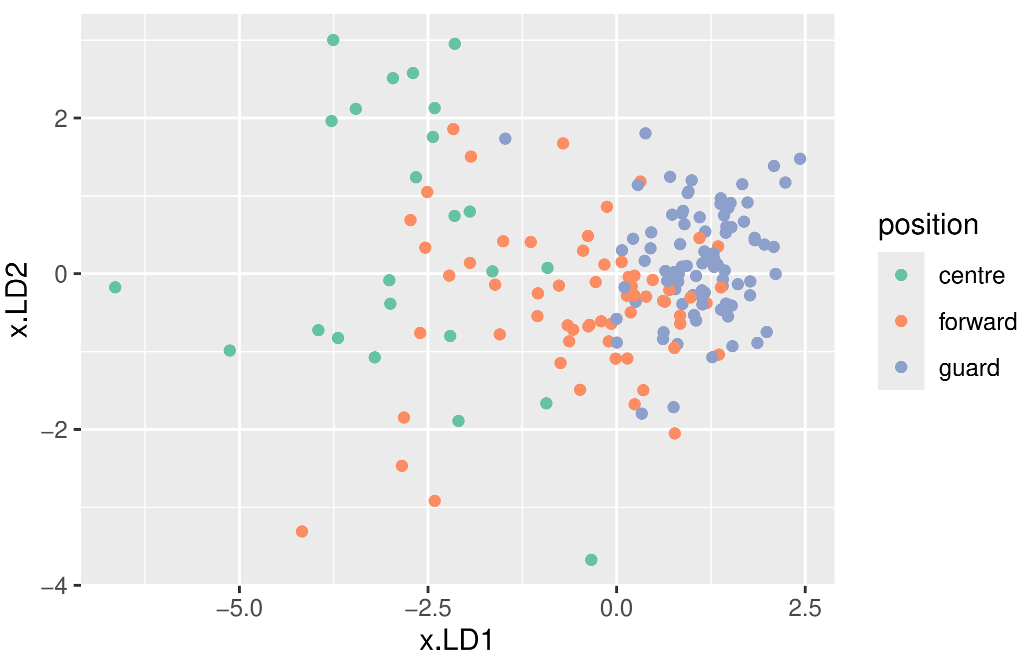


Figure 24: NBA 2015 discriminant analysis plot

	Player	position	class	p.centre	p.forward	p.guard
25	Jose Calderon	guard	guard	0.000	0.171	0.829
173	Thaddeus Young	forward	guard	0.001	0.329	0.670
103	Kawhi Leonard	forward	guard	0.000	0.349	0.651
57	Danilo Gallinari	forward	guard	0.000	0.275	0.724
105	Jeremy Lin	guard	guard	0.000	0.061	0.939
13	Nicolas Batum	forward	forward	0.001	0.647	0.353
167	Andrew Wiggins	guard	guard	0.001	0.354	0.645
140	Iman Shumpert	guard	guard	0.000	0.133	0.867
168	Deron Williams	guard	guard	0.000	0.169	0.831
141	Marcus Smart	guard	guard	0.000	0.064	0.936

Note: The columns with names starting with `p.` originally started with `posterior.` The column `p.centre`, for example, was originally called `posterior.centre`. I changed this so that the table would fit on the page.

Figure 25: NBA 2015 posterior probabilities (selected)

	Sl_No	L500	L1000	L2000	L4000	R500	R1000	R2000	R4000
1	47	5	0	10	70	-5	5	15	40
2	14	5	15	5	60	5	5	0	50
3	55	15	20	10	60	20	20	0	25
4	66	-10	0	5	60	-10	-5	0	65
5	71	0	10	40	60	-5	0	25	50
6	75	0	-10	0	60	15	0	5	50
7	28	-5	-5	-5	55	-5	5	10	70
8	50	-5	0	10	55	-10	0	5	50
9	67	5	10	40	55	0	5	30	40
10	98	10	10	15	55	0	0	5	75
11	18	5	0	0	50	10	10	5	65
12	27	0	0	5	50	5	0	5	40
13	73	0	5	45	50	0	10	15	50
14	34	-10	-10	-10	45	-10	-10	5	45
15	35	-5	10	20	45	-5	10	35	60

Figure 26: Hearing data (15 selected rows)

```r
hearing %>% select(-Sl_No) -> hearing0
hearing.1 <- princomp(hearing0, cor = TRUE)
hearing.1
```

```
Call:
princomp(x = hearing0, cor = TRUE)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
1.9821719 1.2721328 0.9875853 0.6832146 0.5831723 0.5620420 0.4473378 0.3930313

 8  variables and  100 observations.
```

```r
hearing.1$loadings
```

```
Loadings:
      Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
L500   0.401  0.317  0.158  0.328         0.446  0.329  0.546
L1000  0.421  0.225         0.482 -0.379               -0.623
L2000  0.366 -0.239 -0.470  0.282  0.439        -0.526  0.186
L4000  0.281 -0.474  0.430  0.161  0.350 -0.417  0.427
R500   0.343  0.386  0.259 -0.488  0.498  0.195 -0.159 -0.343
R1000  0.411  0.232        -0.372 -0.351 -0.614         0.361
R2000  0.312 -0.317 -0.563 -0.391 -0.111  0.265  0.478 -0.147
R4000  0.254 -0.514  0.426 -0.159 -0.396  0.366 -0.414

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
Cumulative Var  0.125  0.250  0.375  0.500  0.625  0.750  0.875  1.000
```

Figure 27: Hearing data principal components

```
ggscreeplot(hearing.1)
```

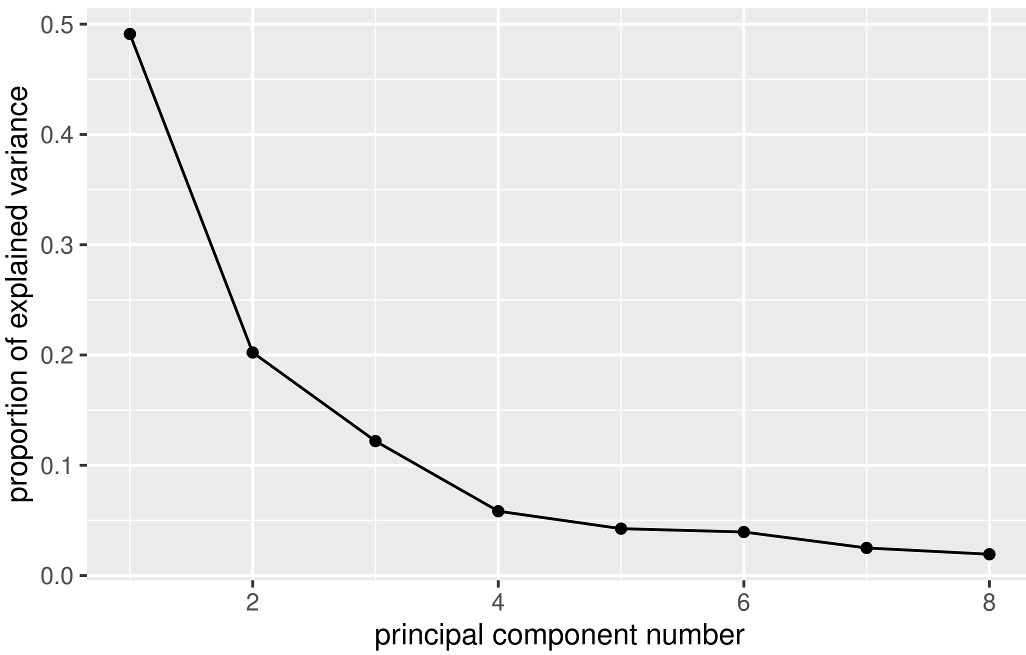


Figure 28: Hearing data screeplot

```
cbind(subject = hearing$Sl_No, hearing.1$scores) %>%
  as_tibble() -> hearing_scores
ggplot(hearing_scores, aes(x = Comp.1, y = Comp.2, label = subject)) +
  geom_text()
```

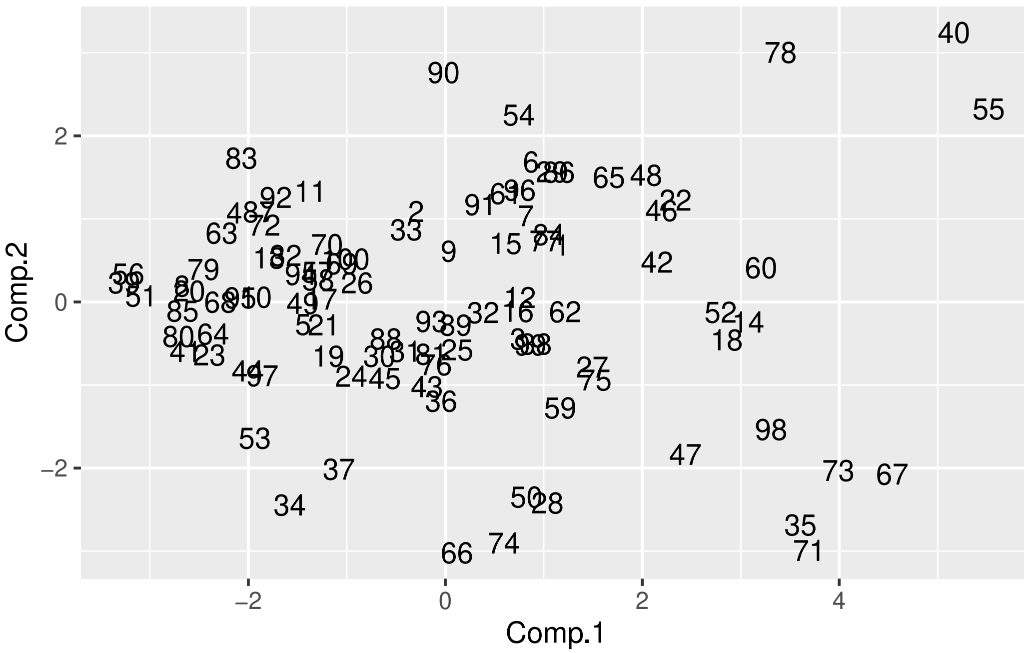


Figure 29: Hearing data principal component scores graph

```
# A tibble: 5 x 9
  percentile  L500 L1000  L2000  L4000  R500 R1000 R2000 R4000
  <chr>      <dbl> <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
1 0%           -10   -10 -10     -10     -10   -10   -10   -10
2 25%          -10    -5  -5       8.75  -10    -5    -5     5
3 50%           -5     0   0      20      -5     0     0    15
4 75%            5     5   6.25   35       0     5     5    30
5 100%          15    20  45      70      25    20    35    75
```

Figure 30: Hearing data summary, showing the 0th, 25th, 50th, 75th, and 100th percentiles (min, Q1, median, Q3, max) of each variable

```
# A tibble: 20 x 4
   age     left_side right_side     n
   <fct>   <chr>     <chr>      <int>
 1 (15,30] no        no          1913
 2 (15,30] no        yes           41
 3 (15,30] yes       no           149
 4 (15,30] yes       yes           63
 5 (30,40] no        no          2226
 6 (30,40] no        yes           48
 7 (30,40] yes       no           190
 8 (30,40] yes       yes           84
 9 (40,50] no        no          2262
10 (40,50] no        yes           40
11 (40,50] yes       no           148
12 (40,50] yes       yes           70
13 (50,70] no        no          1974
14 (50,70] no        yes           31
15 (50,70] yes       no           113
16 (50,70] yes       yes           60
17 (70,90] no        no           671
18 (70,90] no        yes           10
19 (70,90] yes       no            55
20 (70,90] yes       yes           38
```

Figure 31: Chest pain data

```r
chest.1 <- glm(n ~ age * left_side * right_side, data = chest, family = "poisson")
drop1(chest.1, test = "Chisq")
```

```
Single term deletions

Model:
n ~ age * left_side * right_side
                          Df Deviance    AIC    LRT Pr(>Chi)
<none>                          0.0000 175.45
age:left_side:right_side  4   6.6877 174.13 6.6877   0.1533
```

```r
chest.2 <- update(chest.1, . ~ . - age:left_side:right_side)
drop1(chest.2, test = "Chisq")
```

```
Single term deletions

Model:
n ~ age + left_side + right_side + age:left_side + age:right_side +
    left_side:right_side
                     Df Deviance     AIC    LRT Pr(>Chi)
<none>                      6.69  174.13
age:left_side         4    19.52  178.97  12.83  0.01211 *
age:right_side        4     7.71  167.15   1.02  0.90674
left_side:right_side  1   983.32 1148.76 976.63  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

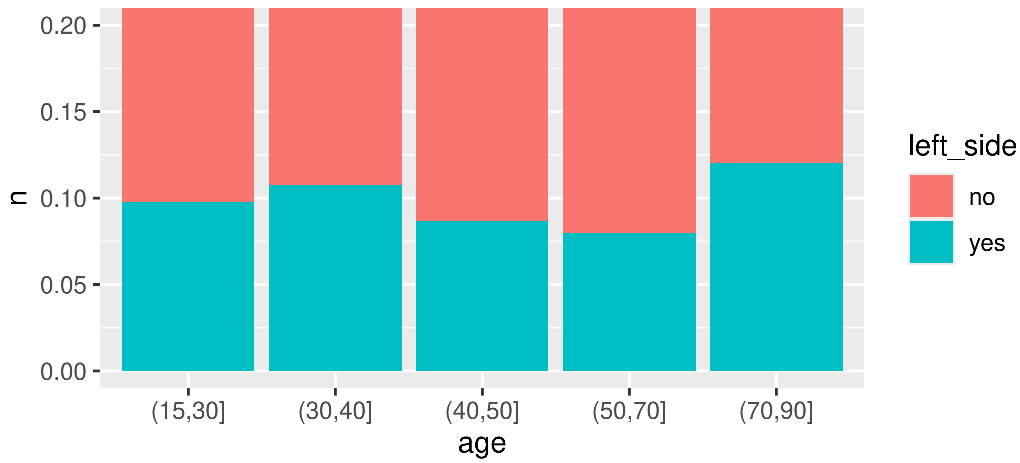
```r
chest.3 <- update(chest.2, . ~ . - age:right_side)
drop1(chest.3, test = "Chisq")
```

```
Single term deletions

Model:
n ~ age + left_side + right_side + age:left_side + left_side:right_side
                     Df Deviance     AIC    LRT  Pr(>Chi)
<none>                      7.71  167.15
age:left_side         4    26.33  177.78  18.63 0.0009308 ***
left_side:right_side  1   990.13 1147.58 982.42 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 32: Chest pain model fitting

```
ggplot(chest, aes(x = age, y = n, fill = left_side)) +
  geom_col(position = "fill") +
  coord_cartesian(ylim = c(0, 0.2))
```



Note: the `coord_cartesian` is used to truncate the y-scale, as on Worksheet 12.

Figure 33: Chest pain graph 1

```
ggplot(chest, aes(x = left_side, y = n, fill = right_side)) +
  geom_col(position = "fill")
```

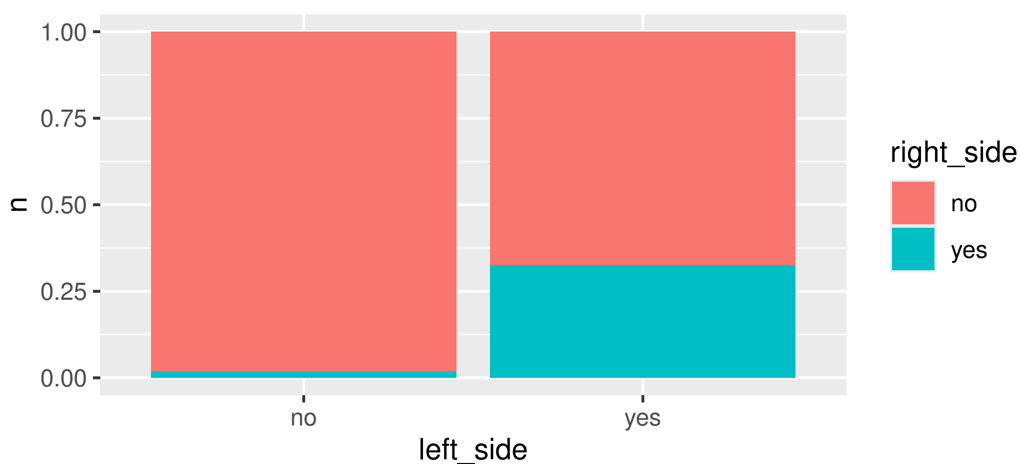


Figure 34: Chest pain graph 2